

Afstudeerverslag

*Ontwikkeling van een E-learning management
systeem*

Student:
Ewout Pronk
Januari 2004

99007299

Colofon

Naam: Ewout Pronk
Studentnummer: 99007299
Afstudeerperiode: 1 september 2003 t/m 9 januari 2004
Opleiding: Informatica & informatiekunde, Haagse Hogeschool
Afstudeerrichting: Vormgeving en Ontwerp van Interactie
Afstudeerblok: 2003-2.1
Project: Ontwikkeling van een E-Learning Management Systeem
Stagebedrijf: Metrix b.v.
bedrijfsmentor: dhr. M. Maas
Examinatoren: mevr. E.B. Graven, dhr. L.C.M. van Koppen

Metrix b.v.
Laan van nieuw oost indië 16
2593 BT Den Haag

Telefoon: +31 (0)70 383 41 14
Telefax: +31 (0)70 383 58 30

Haagse Hogeschool
Sector Informatica

Bezoekadres: Johanna Westerdijkplein 75
2521 EN Den Haag
postadres:
Postbus 19070
2500 CB Den Haag

Telefoon: +31 (0)70 445 84 00
Telefax: +31 (0)70 445 84 05

Referaat

Pronk, E.L. afstudeerverslag, ontwikkeling van een E-learning management systeem, Metrix B.V. te Den Haag, januari 2004.

Dit afstudeerverslag is geschreven in het kader van het afstuderen voor de opleiding Informatica & informatiekunde, afstudeervariant vormgeving en ontwerp van interactie, aan de Haagse Hogeschool te Den Haag.

De auteur geeft in dit verslag zijn ervaringen en problemen weer die zijn opgedaan, de daarbij gevonden oplossingen en gemaakte keuzes in het kader van het afstuderen in de periode 1 september 2003 tot 9 januari 2004.

Dit verslag dient om de examinatoren, de rijksgecommitteerde en andere lezers van dit document inzicht te geven in de omvang van de opdracht en de diepgang waarmee de verschillende aspecten van de opdracht zijn uitgewerkt. Daarnaast geeft dit verslag inzicht in de wijze waarop de producten tot stand zijn gekomen en de uiteindelijke opgeleverde producten aan het bedrijf.

De afstudeeropdracht heeft betrekking op het ontwikkelen van een E-learning management systeem met behulp van Microsoft visual basic 6.0, ASP, en DHTML.

Descriptoren:

- Afstudeerverslag
- Metrix B.V.
- Afstuderen
- Haagse Hogeschool
- Vormgeving en ontwerp van interactie
- Microsoft visual basic
- Active server pages
- ASP
- DHTML
- Javascript
- Dynamic link library
- E-learning
- Leren

Voorwoord

In de periode van 1 september 2003 tot en met 9 januari 2004 heb ik mijn afstudeerstage vervuld bij het bedrijf Metrix B.V.

In dit voorwoord wil ik graag van de gelegenheid gebruik maken om iedereen te bedanken die mij hebben geholpen tijdens deze opdracht.

Het vinden van een afstudeerplek en een daarbij behorende geschikte opdracht is geen makkelijke klus. Meerdere malen heb ik zelfs gedacht een goede afstudeeropdracht op zak te hebben waarna het door verschillende omstandigheden toch nog afketste. Daarom wil ik allereerst Metrix B.V. en haar medewerkers, in het bijzonder mijn begeleider Michon Maas hartelijk danken voor de leuke opdracht en de goede begeleiding die ik heb mogen ontvangen.

Verder gaat mijn dank ook uit naar mijn collega Ruben Maas die mij ook ter ondersteuning was bij moeilijke vraagstukken. Tevens wil ik mijn examinatoren mevr. E.B. Graven en dhr. L.C.M. van Koppen bedanken voor ondersteuning en sturing tijdens dit project

Den Haag, Januari 2004

Inhoudsopgave

Inleiding	6
2 Opstartfase	8
2.1 Afstudeerbedrijf	8
2.2 Opdrachtsomschrijving	9
2.3 Ontwikkelmethode	11
2.4 Op te leveren producten	13
2.5 Te gebruiken technieken	13
2.6 Planning	14
3 Definitiestudie	16
3.1 Planning	16
3.2 Inwinnen van informatie	16
3.3 Beschouwen doelen en beperkingen systeem	18
3.4 Pilotplan	28
4 Pilot student	32
4.1 Definitie	32
4.2 Planning	32
4.3 Workshop	33
4.4 Pilotontwikkelplan	34
4.5 Realisatie	40
4.6 Test	43
4.7 Aanpassingen	44
5 Pilot docent	45
5.1 Definitie	45
5.2 Planning	45
5.3 Workshop	46
5.4 Pilotontwikkelplan	46
5.5 Realisatie	48
5.6 testen	53
5.7 Aanpassingen	54
6 Herstel en oplevering	55
6.1 Aanpak	55
6.2 Uitvoering	55
6.3 Afronding	57
7 Evaluatie proces	59
7.1 Evaluatie opstartfase	59
7.2 Evaluatie fase plan van aanpak	59
7.3 Evaluatie fase definitiestudie	60
7.4 Evaluatie fase pilotontwikkeling	61
7.5 Evaluatie fase herstel en oplevering	62
8 Evaluatie product	63
8.1 Evaluatie documenten	63
8.2 Evaluatie applicatie Keylearning	63
Bronnenlijst	66
Woordenlijst	67

1 Inleiding

Het afstuderen is een belangrijk onderdeel van de opleiding Informatica en informatiekunde, opleidingsvariant Vormgeving en Ontwerp van Interactie (VIA) aan de Haagse Hogeschool te Den Haag. In dit document wordt een compleet verslag gegeven van het afstuderen. Hierbij wordt niet enkel een beschrijving gegeven van de gebeurtenissen, ook worden afwegingen, gemaakte keuzes en oplossingsmethoden beschreven. Doelstelling van dit afstudeerverslag is de examinatoren en de rijksgecommitteerde voldoende inzicht te geven in:

- De omvang van de opdracht
- De diepgang waarmee verschillende aspecten van de opdracht zijn uitgewerkt.
- De wijze waarop producten tot stand komen.
- De uiteindelijke opgeleverde producten voor het bedrijf.

Het verslag is opgedeeld in drie delen. Naast deze drie delen zijn de bijlagen van dit verslag in een apart document opgenomen.

Deel 1: Opstartfase en definitiestudie

Deel 2: Bouwfase (pilotontwikkeling en afronding bouwfase)

Deel 3: Evaluatie (product en proces)

Mijn opdracht luidt globaal als volgt: "ontwerp een laagdrempelig leer management systeem dat zich ten aanzien van de prijsstelling aan de onderkant van de E-Learning markt gaat positioneren. De doelgroep zijn kleine en middelgrote organisaties met de behoefte om leermateriaal en activiteiten te managen en de voortgang te bewaken." Vanaf nu zal in dit verslag het "learning management systeem" worden afgekort tot LMS of Keylearning. Dit is ook de voorlopige naam van het systeem.

Keylearning wordt ontwikkeld in opdracht van het bedrijf Metrix B.V. Metrix B.V. maakt service management software. Dit is software om bedrijven die zich bezig houden met service activiteiten (reparaties en dergelijke) te ondersteunen. Metrix maakt hiervoor gebruik van twee applicaties. Enerzijds is dit de Metrix applicatie voor grote bedrijven. En anderzijds Rachel dat is gemaakt voor het midden- en kleinbedrijf (MKB). Bij wijze van proef wil Metrix zich meer gaan richten op het MKB. Metrix wil zijn producten uitbreiden met een learning management systeem. Metrix heeft zelf al onderzoek gedaan en daaruit blijkt dat er een markt voor is.

Waarom heb ik voor deze opdracht gekozen? Ik wilde een opdracht waarbij niet alleen aandacht wordt besteed aan vormgeving, maar waarin ook iets met databases gedaan moet worden plus een programmeergedeelte. Ik had natuurlijk ook voor een opdracht kunnen kiezen met alleen vormgeving, maar dan is de vraag: Heeft zo'n opdracht voldoende diepgang voor het afstuderen? Verder is het zo dat ik zelf later niet alleen het vormgevingsaspect wil doen maar ook iets met programmeren. Toen deze opdracht voorhanden kwam, was de keuze voor deze "complete" opdracht snel gemaakt.

Dit verslag bestaat uit een aantal delen. In het eerste hoofdstuk "opstartfase" wordt de eerste aanzet gegeven over het kader waarin de opdracht zich bevindt. Tevens wordt in dit hoofdstuk het "plan van aanpak" besproken. In het hoofdstuk "Definitiestudie" ga ik in op keuzen die ik neem in de fase waarin de doelen en beperkingen van het systeem in kaart worden gebracht. In het hoofdstukken "pilot student" en "pilot docent" wordt een toelichting gegeven over de fase waarin de aparte bouwdelen van het systeem zijn ontwikkeld. In het hoofdstuk "herstel en oplevering" wordt de afronding van dit project beschreven en in de hoofdstukken "productevaluatie" en "procesevaluatie" volgt er een evaluatie waarin kritisch wordt teruggekeken op het proces en de producten van dit project.

Deel I Opstartfase en definitiestudie

2 Opstartfase

In dit hoofdstuk wordt besproken hoe de opstartfase van het project is verlopen. Het hoofdstuk is onderverdeeld in de volgende paragrafen:

- Afstudeerbedrijf
- Opdrachtomschrijving
- De ontwikkelmethoden
- Op te leveren producten
- Te gebruiken technieken
- Planning

Alvorens de opdracht uit te diepen volgt hieronder een paragraaf welke een beeld geeft van de activiteiten van Metrix alsmede de plaats van de opdracht en afstudeerder.

2.1 Afstudeerbedrijf

2.1.1 Het bedrijf Metrix

Metrix is opgericht in 1980 door de heren Larry Laux, Rob Oldenburgh en Jim Elliot. Aanvankelijk was het opgericht als een automatiseringsbedrijf bestaande uit de bovengenoemde ervaren programmeurs. In de periode 1980 tot en met 1983 hebben ze echter een toepassing opgeleverd, die gericht was op "field service management". Dat wil zeggen een applicatie die werknemers "managed" die bij klanten werkzaam zijn.

Deze applicatie werd geschreven in de programmeertaal basic voor de hp250. In die tijd was er nog sprake van dat een programmeertaal afhankelijk is van een bepaald type computer. In 1987 werd een tweede versie op de markt gebracht die ook de service activiteiten van repair centres kon managen. Deze versie werd geschreven met behulp van cobol voor de hp3000. Door de laatste versie heeft het bedrijf tot aan begin van de jaren negentig kunnen uitgroeien tot een organisatie met ongeveer 60 werknemers. Voor die tijd heeft Metrix zich nog beziggehouden met het bouwen van een extra toepassing, die eveneens met behulp van de programmeertaal Cobol is geschreven. Met deze applicatie werd verdere integratie met de ingenieur en het Metrix systeem verwezenlijkt. Dit omdat deze toepassing de ingenieur in staat stelt om op afstand met het systeem te communiceren met behulp van een palmtop.

In het begin van de jaren negentig is Metrix overgestapt op vierde generatie programmeertalen, met als resultaat "Metrix 4^e". Tot op de dag van vandaag wordt "Metrix 4^e" verkocht aan klanten en worden uitbreidingen en aanpassingen op dit product gedaan.

Metrix is één van de marktleiders in het maken en ondersteunen van Fieldservice management producten. Dit zijn producten voor het web maar ook stand-alone applicaties. Dit beslaat alle aspecten van fieldservice management voor bedrijven met service activiteiten. Zoals reparatiecentra voor medische producten, elektronica, telecommunicatie en dergelijke.

Huidige werkzaamheden van Metrix Europe:

- Het leveren van productondersteuning bij bestaande klanten
- Het werven van nieuwe klanten
- Het ontwikkelen van nieuwe applicaties voor het MKB

Huidige werkzaamheden van Metrix USA:

- Het verder verbeteren van eProduct service
- Het leveren van productondersteuning bij bestaande klanten
- Het werven van nieuwe klanten
- Het ontwikkelen van nieuwe (niet nader genoemde) applicaties

2.1.2 Plaats opdracht binnen Metrix

De opdracht heeft de titulatuur "proof of concept". Dit wil zeggen dat Metrix er vanuit gaat dat het te ontwikkelen systeem commercieel uit te buiten is maar Metrix is hier nog niet 100% van overtuigd. Hier ligt een mooie kans voor een afstudeerder om een eerste aanzet te geven om het product te lanceren. De opdracht is binnen Metrix uniek omdat Metrix zich tot nu toe puur heeft gericht op het bouwen en het leveren van ondersteuning op CRM producten. Omdat Metrix niet langer op één paard wil wedden wordt deze nieuwe weg ingeslagen. Eén Metrix medewerker heeft een E-learning achtergrond en bij hem is het idee geboren. Behalve het op de markt brengen voor het MKB is Keylearning ook voor Metrix zelf een aanvulling omdat deze ook gebruikt kan worden om mensen intern te scholen. Omdat medewerkers van Metrix zich regelmatig bij klanten bevinden beschikt Metrix over informatie dat ook bij bestaande klanten vraag is naar een applicatie zoals Keylearning.

2.1.3 Plaats afstudeerder binnen Metrix

De afstudeerder is werkzaam op de afdeling development, deze afdeling is klein omdat de meeste ontwerpactiviteiten in de Verenigde Staten plaatsvinden. Behalve twee reguliere medewerkers is er naast de afstudeerder (mijzelf) ook nog een andere afstudeerder werkzaam aan een ander product, "keytask".

2.2 Opdrachtsomschrijving

De beschrijving van de opdracht is beknopt. De volledige opdrachtsomschrijving is te vinden in de bijlage.

2.2.1 Aanleiding van de opdracht

Bij wijze van proef wil Metrix een learning management system (LMS) maken voor het MKB. Naast de eigen organisatie van Metrix dient het product geschikt te zijn voor aanbieders van cursussen, bedrijven in het MKB met leeractiviteiten voor eigen medewerkers en reïntegratiebedrijven. Het LMS dient vanaf iedere computer met een internet aansluiting en Microsoft Internet Explorer (minimaal versie 5.5) te benaderen te zijn. Dit voor zowel diegene die de course (cursus, leertraject) volgt (student) alsmede voor diegene die de cursus onderhoudt (docent).

2.2.2 Totstandkoming van de opdrachtsomschrijving

Over de manier waarop ik het project ga aanpakken, wordt ik door de opdrachtgever vrijgelaten. De opdrachtgever/contactpersoon heeft geen voorkeur voor een ontwikkelmethode en wil mij zo weinig mogelijk beperkingen opleggen. Zijn motto is: "Je bent een bijna afgestudeerde H.B.O.'er en daarom heb ik er veel vertrouwen in dat je in

staat bent de juiste ontwikkelmethode te kiezen en mij te adviseren over de reden waarom de methode is gekozen. Het maakt mij niet uit welke methode je kiest, zolang het resultaat maar goed is en te realiseren is binnen de kaders die door ons en de opleiding gesteld worden". De opdrachtgever stelt mij verder geen beperkingen en ik ben hierdoor vrij om een scala aan methoden en technieken te gebruiken die mij het beste liggen en het meest geschikt zijn voor de opdracht en het te behalen resultaat.

Om voor de lezer duidelijk te stellen wat het probleem/doel is van de opdracht volgt hieronder een opsomming van de kernpunten van de opdrachtschrijving.

2.2.3 Probleemstelling

Voor bedrijven binnen het MKB is het moeilijk een LMS te vinden dat zowel qua prijs alsmede qua functionaliteit aan de behoefte voldoet. Vooral voor bedrijven waar leren een secundair proces is, is het moeilijk een passend product te vinden.

Het op maat laten maken van een applicatie is vaak te duur voor MKB'ers evenals de huidige systemen op de markt. Het aanleveren van cursusmateriaal wordt ook als probleem ervaren. Courses worden veelal op papier aangeleverd. Versies zijn hierdoor vaak niet up to date en distributie van cd's en diskettes is lastig.

2.2.4 Doelstelling

De afstudeerder gaat een laagdrempelig LMS ontwikkelen dat past binnen de behoefte van organisaties die in bepaalde mate leeractiviteiten aanbieden aan het eigen personeel of externen. De applicatie zal onder de noemer "KeyLearning" worden ontwikkeld en zal zich qua prijsstelling aan de onderkant van de markt van learning management systems gaan positioneren. Het betreft een standaard applicatie, die slechts op het gebied van vormgeving aangepast hoeft te worden aan de stijl van het bedrijf waar het geheel geïmplementeerd gaat worden. De applicatie is een raamwerk voor het managen, monitoren en beschikbaar maken van leermiddelen waarmee de leeractiviteiten uitgevoerd kunnen worden. Organisaties kunnen hun eigen leermaterialen gebruiken als onderdelen van het leertraject. De leeractiviteiten kunnen naar deze content verwijzen als deze centraal op een (web)server wordt geplaatst.

De doelgroepen zijn kleine en middelgrote organisaties met de behoefte om leermiddelen en activiteiten te managen en de voortgang te bewaken. Te denken valt aan:

- Afdelingen van grotere organisaties met eigen leeractiviteiten
- Scholen in het primaire en voortgezet onderwijs
- Kleine opleidingsbedrijven
- Kleine en middelgrote organisaties met eigen leeractiviteiten

Door het systeem tweetalig te maken (Engels en Nederlands) behoort een internationale werkomgeving en een wereldwijd afzetgebied tot de mogelijkheden.

Overigens zal het LMS eerst bij Metrix intern worden ingevoerd als proef, om vervolgens in een later stadium buiten bestek van deze opdracht op de markt te worden gebracht.

2.2.5 Resultaat

Het te bouwen systeem valt uiteen in twee modules:

Een studentmodule:

De studentmodule is beveiligd door middel van verificatie van gebruikersnaam en password. Als de student ingelogd is, bestaat de mogelijkheid om openstaande en reeds afgeronde leertrajecten te bekijken. Van deze leertrajecten kan de student de activiteiten uitvoeren en hierbij gebruik maken van bijbehorende leermiddelen. De student kan met de manager van het leertraject (een docent) communiceren door middel van een forum

of prikbord. Verder kan de student zijn persoonlijke gegevens wijzigen indien dit gewenst is. De student kan zijn eigen voortgang bewaken door interpretatie van zijn eigen resultaten. Indien gewenst mag de student zich zelf inschrijven voor leertrajecten en ter ondersteuning van het proces kan de student documenten uploaden en downloaden.

Een docentmodule:

Voor een docent is het mogelijk om gebruikers, leertrajecten, activiteiten, leermiddelen en het forum te beheren. Verder dient de docent te kunnen communiceren door middel van een prikbord of forum. Docenten kunnen via het LMS beoordelingen geven aan activiteiten van studenten. Per activiteit, leertraject en gebruiker dienen rapporten gecreëerd te kunnen worden. Dit is bedoeld voor managementrapportage maar ook om de voortgang te bewaken en problemen te signaleren.

Voor beide modulen dient een intuïtieve user interface ontwikkeld te worden, zodat het in eerste instantie nog niet nodig is om een gebruikershandleiding te vervaardigen.

2.3 Ontwikkelmethode

Na het vaststellen van de probleem- en doelstelling moet er een belangrijke keuze in dit project worden genomen, namelijk het kiezen van een ontwikkelmethode. Metrix had bij vorige projecten reeds met IAD gewerkt en is daar zeer over te spreken. Ik heb zelf ook een lichte voorkeur voor IAD omdat er door mij in het verleden ook goede resultaten mee geboekt zijn. Een goed alternatief voor IAD is RAD. Om na te gaan welke methode het beste werkt heb ik beiden nauwkeurig geanalyseerd. Ter verduidelijking zijn hieronder deze resultaten te vinden:

IAD werkt als:

- Er onduidelijke of snel wijzigende systeemeisen zijn.
- Er een aantal bijna gelijksoortige applicaties worden ontwikkeld, waarbij hergebruik met name tot een stijging van de productiviteit zou kunnen leiden.
- Applicaties een grafische gebruikers-interfaces (GUI's) hebben, waarbij de specificaties alleen goed kunnen worden gevalideerd via werkende prototypen.
- Een organisatie veel waarde hecht aan een hoge mate van gebruikersparticipatie in het ontwikkelproces.

IAD werkt niet als:

- Het korte-termijnresultaat voorop staat, want dit kan leiden tot mindere kwaliteit van de architectuur en de documentatie.
- De klant niet kan leven met het veelvuldig nemen van (kleinere) beslissingen.
- Het niet mogelijk is om constant van prioriteit te wisselen.

RAD werkt als:

- De applicatie standalone is.
- Er al gebruik kan worden gemaakt van bestaande class libraries (API's).
- De performance niet kritisch is.
- Het systeem in verschillende modules kan worden gemaakt.
- De technologie meer dan een jaar oud is.

RAD werkt niet als:

- De applicatie samen moet kunnen werken met bestaande programma's.
- Er niet of nauwelijks gebruik kan worden gemaakt van plugin componenten.
- RAD wordt vervangen door QADAD (Quick And Dirty Application Development).
- RAD wordt gebruikt voor het maken van operating systems of games.

Uiteindelijk heb ik besloten om gebruik te maken van IAD op basis van de volgende argumenten:

- IAD werkt goed als er GUI's (Grafical User Interfaces) gebruikt worden. Motivatie: Een intuïtieve gebruikers interface is een belangrijk onderdeel van de opdracht.
- IAD is een methode waarbij gebruikersparticipatie uitermate goed ingepast kan worden in bijvoorbeeld workshops of testen. Dit komt door het gebruik van workshops waarbij een grote input van beoogde gebruikers wordt gegenereerd.
- Het kiezen van een onbekende methode is risicovol, omdat het zeker aan het begin meer tijd kost om deze onder de knie te krijgen. Tijd die ik in dit project wellicht te kort kan komen.
- IAD is een methode die prima werkt met de methode GUIDE. Dit weet ik uit eigen ervaring omdat ik in het verleden zeer goede resultaten met deze combinatie heb behaald.

Natuurlijk valt nooit meer te zeggen of een andere methode wellicht beter uit zou pakken maar de slogan "never change a winning team" was voor mij en de doelstelling van de opdracht belangrijker dan "wie niet waagt wie niet wint".

Na de keuze voor de methode volgt logischerwijs de keuze voor een ontwikkelstrategie. IAD is echter een methode die veel varianten kent. Eén van de belangrijke zaken die de slagingskansen van het project bepaalt betreft het kiezen van een iteratiestrategie.

IAD onderscheidt vier strategieën:

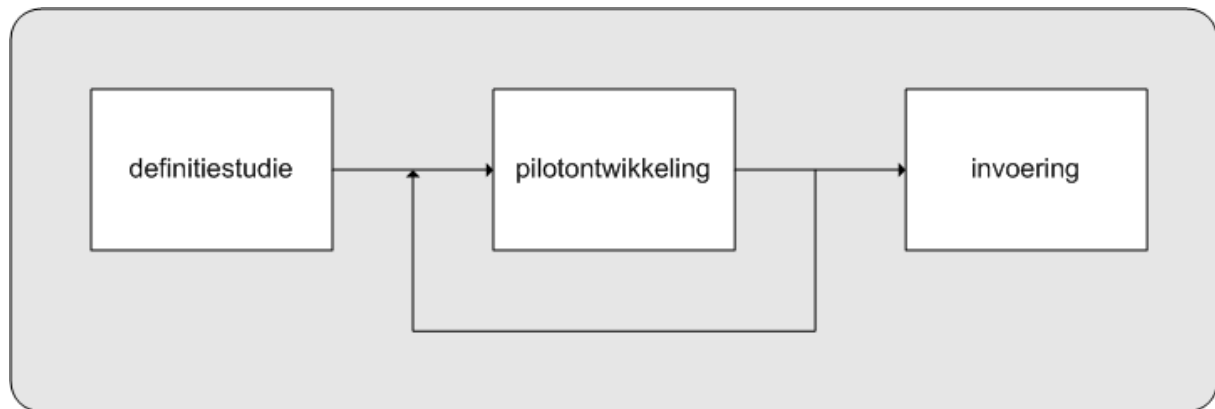
- Evolutionair ontwikkelen
- Incrementeel opleveren
- Incrementeel ontwikkelen
- Big-bang invoeren

Allereerst heb ik evolutionair ontwikkelen bestudeerd. Hierbij worden definitiestudie, pilotontwikkeling en invoering steeds herhaald. Deze strategie is echter niet helemaal geschikt. Het is een methode die vaak gebruikt wordt in omstandigheden waarmee ik in dit project niet mee te maken heb (onzekere functionele systeemeisen, snel veranderende projectomgeving, hoog risico en voortdurend evoluerend informatiesysteem). Bovendien is het bij deze strategie niet mogelijk om na de fase invoering weer opnieuw te beginnen met de fase definitiestudie zoals deze strategie voorschrijft. Hier is simpelweg de tijd niet voor.

Incrementeel opleveren is een methode waarbij pilotontwikkeling en invoering steeds worden herhaald. Op zich een goed te gebruiken strategie. Echter speelt hier weer een beperking in tijd. Er is geen tijd om het systeem in te voeren en vervolgens weer te beginnen aan de pilotontwikkeling. Bij een project met meer tijd is dit een perfecte methode.

Big-bang invoeren past helemaal niet. Voor deze methode heb je echt een aanzienlijke langere tijd nodig en is vooral gericht op organisaties die niet veranderingsgereed zijn. Dit terwijl Metrix een organisatie is waarbij innovatie hoog in het vaandel staat.

Incrementeel ontwikkelen blijkt het beste te passen bij de eigenschappen van dit project. Bovendien wordt incrementeel ontwikkelen "Rad binnen IAD" genoemd omdat het een snelle ontwikkelingsmethode is. Zoals al eerder genoemd is dit project zeer tijdgebonden. Deze strategie maakt de slagingskans groter:



Figuur 2.1 Incrementele Ontwikkeling

Deze methode is qua tijd naar mijn inschatting te doen in de periode die staat voor het project. Ook heeft deze methode een aantal voordelen die perfect passen bij een project zoals ik die uitvoer namelijk:

- Het systeem kan goed van tevoren worden gespecificeerd
- De omvang van het systeem is goed te overzien
- Een stapsgewijze invoering is niet gewenst of zelfs onmogelijk

2.4 Op te leveren producten

Nu de strategie vastligt kan ik aan de hand van IAD bepalen welke producten er worden opgeleverd. Dit zijn de producten: plan van aanpak, definitiestudie en pilotontwikkelplannen. Het plan van aanpak is bedoeld om ontwikkelaar en opdrachtgever inzicht te geven in het project, in de definitiestudie wordt het verdere verloop definitief vastgelegd en in de pilotontwikkelplannen wordt concreet gemaakt wanneer er wat gebouwd moet worden.

Reeds in de opstartfase heb ik ervoor gekozen om voor alle op te leveren producten precies vast te leggen welke eisen er aan de documenten worden gesteld. Dit is voor zowel de opdrachtgever als voor mijzelf controleerbaar. De opdrachtgever weet wat hij kan verwachten en ik kan op deze manier beter plannen, doordat alles in kleine stukjes wordt verdeeld. Eén nadeel kan natuurlijk wel zijn dat wanneer je iets om een bepaalde reden niet maakt, de opdrachtgever je hierop afrekent.

2.5 Te gebruiken technieken

Voor wat betreft technieken kies ik voor standaardtechnieken. Hiermee bedoel ik de technieken die ik op school heb geleerd. Het zijn technieken die over het algemeen in elk project inpasbaar zijn. Ik had natuurlijk ook kunnen kiezen voor een scala nieuwe technieken. Maar ik vind dat je beter technieken kan gebruiken waarmee je al vertrouwd bent en in het verleden resultaat opgeleverd hebben, dan nieuwe die misschien alleen maar meer tijd kosten.

Ik heb voor de techniek objectmodellering gekozen omdat dit de meest gangbare techniek is als je gebruik maakt van objectoriëntatie. Ik maak gebruik van interviewtechniek omdat je hiermee in een korte tijd, door het stellen van gerichte vragen, veel informatie

kan achterhalen. Als planningstechniek kies ik voor time-boxing, doordat je het project in kleine delen opdeelt kun je betrouwbaar plannen. Bovendien blijkt dat time-boxing goed om een iteratieve ontwikkelcyclus te beheren. Met betrekking tot databasemodelling heb ik gekozen voor een relationeel databasemodel. Dit omdat dit model in één oogopslag weergeeft wat de opbouw van de database is. Ik heb niet gekozen voor een visueel databasemodel omdat een visuele interpretatie van de database ook herleidt kan worden uit het objectmodel. De keuze voor een interview in plaats van een workshop zal later in dit verslag bij paragraaf 3.2 worden toegelicht.

2.6 Planning

Plannen is een ingewikkeld proces. Ineens ligt er een planning waarvan je op het moment van opstellen niet 100% zeker weet of het realiseerbaar is. In deze paragraaf zal ik beschrijven hoe ik tot mijn planning gekomen ben.

Voor ik kan overgaan tot plannen heb ik moeten wachten tot ik de juiste planning van de afstudeerperiode van school aangeleverd kreeg. Toen ik deze ontving, bleek de tijd die voor het afstuderen beschikbaar is slechts 19 weken te zijn (althans op papier). Er van uitgaande dat ik geen vrije dagen opneem hou je 91 ($19 * 5$, – 4 feestdagen) werkdagen over. Dit betekent dus 18 weken voor het maken van de opdracht plus het maken van het afstudeerverslag. Aangezien het afstudeerverslag tussen de werkzaamheden door wordt gemaakt hoeft hier verder geen tijd voor ingepland te worden. Het is wel zo dat de laatste week genaamd “uitloop” geheel in het teken staat van het leggen van de laatste hand aan het afstudeerverslag.

Bij het maken van de planning kwam ik erachter dat het maken van een gebruikershandleiding geen haalbare kaart was.

Het maken van een gebruikershandleiding is een tijdrovende klus. In een gesprek met de opdrachtgever ben ik tot de conclusie gekomen dat er beter wat meer tijd kan worden gestoken in het bouwen van meer functionaliteit. Als ik ervoor zou kiezen om gebruikershandleidingen te maken dan zou dit ten koste gaan van de bouwtijd voor de applicatie. Het komt er dan op neer dat je een handleiding hebt voor een applicatie die niet af is, en dit is natuurlijk een situatie die je wil vermijden.

Daarnaast speelt een intuïtieve gebruikersinterface een belangrijke rol in de opdracht. Een handleiding zou hierdoor geen noodzakelijk goed dienen te zijn.

Ter verduidelijking de globale- en detailplanning van het afstudeerproject:

2.6.1 Globale planning

Activiteiten \ Weken	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Maken Plan van aanpak																		
Maken Defenitiestudie																		
Maken Pilotontwikkelplan 'student'																		
Bouwen pilot 'student'																		
Maken Pilotontwikkelplan 'Docent'																		
Bouwen pilot 'Docent'																		
Test & Herstel & invoering																		
Uitloop																		

Figuur 2.2 Globale plannig

2.6.2 Detail planning

Activiteiten \ Weken	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Kalenderweken	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	1	2
	Plan van aanpak																	
opstellen plan van aanpak																		
	Definitiestudie																	
stel plan van aanpak op																		
definieer ontwikkelscenario																		
bereid workshop voor																		
voer workshop uit																		
evalueer workshop																		
definieer systeemeisen																		
bepaal systeemconcept																		
beschouw technische structuur																		
beschouw organisatorische inrichting																		
stel pilotplan op																		
	pilot ontwikkeling pilot student																	
stel plan van aanpak op																		
bereid pilotplan workshop voor																		
voer workshop uit																		
evalueer workshop																		
stel pilotontwikkelplan op																		
bouw pilot																		
test pilot																		
evalueer test																		
pas pilot aan																		
	pilot ontwikkeling pilot docent																	
stel plan van aanpak op																		
bereid pilotplan workshop voor																		
voer workshop uit																		
evalueer workshop																		
stel pilotontwikkelplan op																		
bouw pilot																		
test pilot																		
evalueer test																		
pas pilot aan																		
	test & herstel & invoering																	
koppeling pilots																		
laatste mogelijkheid tot aanpassing																		
bereid test voor																		
test systeem																		
evalueer test																		
invoering/opstellen "to do list"																		
	uitloop																	
uitloop																		

Figuur 2.3 Detail planning

In tegenstelling tot vorige projecten heb ik wat minder tijd ingepland voor de definitiestudie. In het verleden heb ik hier vaak wat teveel tijd aan besteed. Omdat voor dit systeem de specificaties vrij helder zijn heb ik deze beslissing genomen. De nadruk bij dit project ligt op het bouwen van functionaliteit. De fase "test, herstel & invoering" is voor het koppelen van de pilots en de puntjes op de "I" zetten. Er zijn altijd nog wel zaken die gedaan moeten worden alvorens een pilot kan worden opgeleverd. Ook kan in deze fase worden hersteld wat in de test als negatief is beoordeeld. Na deze herstelfase kan het systeem opgeleverd worden of er wordt een "to do" list opgesteld. De invoering zal dan worden uitgesteld.

De pilots kennen overigens slechts één iteratie, meer hierover is te vinden in paragraaf 3.4

3 Definitiestudie

Binnen de fase definitiestudie worden de doelen en beperkingen van het systeem in kaart gebracht. Tevens zal in deze fase worden aangeduid hoe het verdere verloop van het project eruit zal komen te zien. In de eerste paragraaf zal de planning voor de fase definitiestudie worden doorgenomen, in de paragraaf *inwinnen van informatie* zal ik nader verklaren hoe ik aan de input voor het schrijven van de definitiestudie ben gekomen. In de paragraaf *beschouwen doelen en beperkingen systeem* zal dieper worden ingegaan op het systeemconcept en de organisatorische inrichting van het systeem. In de paragraaf *pilotplan* zal worden toegelicht hoe het pilotplan tot stand is gekomen.

3.1 Planning

De fase definitiestudie bestaat uit 2,5 werkweek (12 werkdagen), zoals beschreven in het globale plan van aanpak:

Activiteiten \ dagen	1	2	3	4	5	6	7	8	9	10	11	12
opstellen plan van aanpak												
definieer ontwikkelscenario												
voorbereiding workshop												
uitvoering workshop												
evaluatie workshop												
in kaart brengen huidige situatie												
definieer systeemeisen												
bepaal systeemconcept												
beschouw technische structuur												
beschouw organisatorische inrichting												
stel pilotplan op												

Figuur 3.1 planning fase definitiestudie

3.2 Inwinnen van informatie

De definitiestudie begint over het algemeen met het inwinnen van informatie. Ik heb hiervoor een aantal E-learning pakketen bekeken. Dit met als doel om de huidige situatie zo goed mogelijk in kaart te brengen en om van alle systemen de voor- en nadelen af te wegen. Probleem echter is dat deze systemen (uiteeraard) niet zomaar te downloaden zijn. Ik ben mij dus gaan verdiepen in de systeemspecificaties van de verschillende systemen welke wel gemakkelijk te verkrijgen zijn. Alvorens ik deze ben gaan bestuderen moest ik natuurlijk wel criteria hebben om onderzoek naar te doen. Ik heb gekozen voor de twee belangrijkste criteria waaraan Keylearnnig moet voldoen, te weten: laagdrempeligheid en prijsstelling

Ik heb van de volgende systemen specificaties bekeken:

- Lotus LearningSpace
- Docent
- Whizzdom
- Netg

De reden dat ik juist deze applicaties heb onderzocht is omdat dit bekende E-learning pakketten zijn waar ik voldoende informatie over kon vinden zonder dat de applicatie zelf te moeten raadplegen.

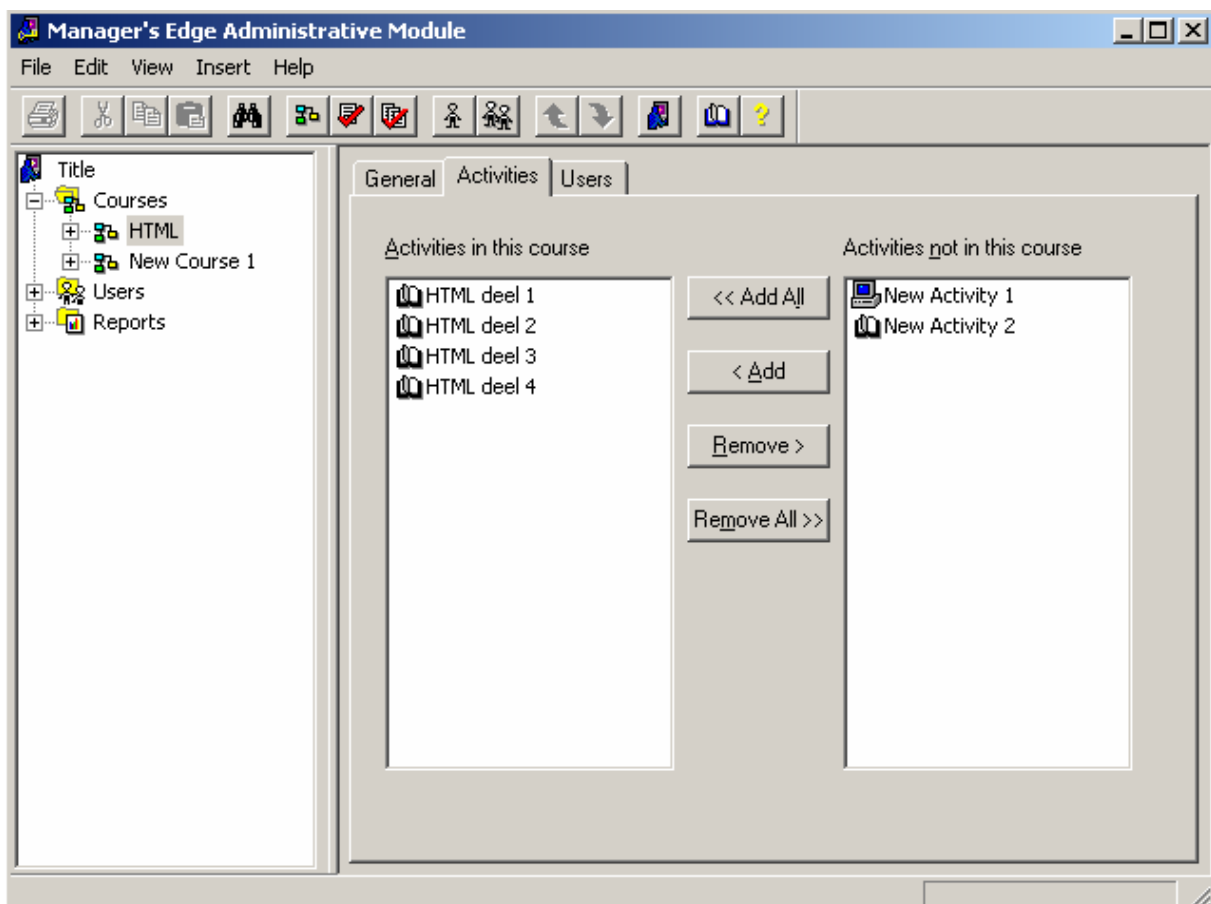
Het blijkt dat deze systemen allemaal te groot en te complex zijn voor de vier doelgroepen (zie paragraaf 3.3) van deze applicatie. Zoals u wellicht kunt begrijpen, heb ik de gebruiksvriendelijkheid niet kunnen beoordelen aan de hand van specificaties.

De systemen *Blackboard* en *Managers Edge* heb ik in de praktijk getest. Blackboard is in principe een goed product, maar is te duur voor de vier doelgroepen. Verder is het zo dat Blackboard over het algemeen niet als laagdrempelig wordt beschouwd. Ook blijkt bij navraag over grote systemen zoals Blackboard geen draagvlak te zijn onder alle gebruikers. Eigenlijk is dit ook hetgeen wat bij de Haagse Hogeschool gebeurt. Men probeert Blackboard te promoten maar er worden geen leermaterialen op gezet. Het is speculeren waardoor dit komt maar het zou best kunnen zijn dat deze applicatie teveel opties heeft waardoor het gebruik voor docenten als te complex wordt ervaren en het draagvlak laag is.

Als ik hiernaar onderzoek had willen doen had mij dat een aantal weken gekost om uit te zoeken wat de reden hiervoor precies is. Ik heb dit achterwege gelaten omdat grote organisaties als de Haagse hogeschool sowieso niet tot de doelgroep horen.

Het product *Managers Edge* zit qua navigatie en gebruiksvriendelijkheid zeer goed in elkaar. De navigatie door middel van een treeview is overzichtelijk en spreekt mij zeer aan. Verder blijkt uit onderstaande figuur het gemak waarmee activiteiten kunnen worden toegevoegd aan courses. Dit kan door middel van buttons en door middel van dubbelklikken.

Voor de gekozen Keylearning doelgroepen die Keylearning is het systeem echter iets te uitgebreid. Verder is een groot nadeel van dit systeem dat het geen webfunctionaliteit heeft. Het biedt verder wel een goede kijk op E-learning in de praktijk.



Figuur 3.2 managers edge

Verder is het zo dat binnen Metrix door één van de medewerkers veel informatie over E-learning heeft verzameld. De belangrijkste standaarden met betrekking tot E-learning zijn *AICC* en *SCORM*. AICC is een veelgebruikte standaard uit de luchtvaartindustrie. Het is echter zo dat deze standaard te uitgebreid is voor een relatief klein programma zoals

Keylearning. Ook is de standaard verouderd en niet geschikt voor het web. Scorm betekent Sharable Content Object Reference Model en is een applicatieprofiel (referentiemodel) dat is ontwikkeld door ADL. ADL, Advanced Distributed Learning, is een Amerikaans overheidsinitiatief van het ministerie van defensie met als doel het promoten en stimuleren van het ontwikkelen van on-line leermaterialen en software voor onderwijs en training. Uitwisselbaarheid van onderwijsmateriaal is een primaire doelstelling. Om dit te bewerkstelligen heeft men het SCORM model ontwikkeld. SCORM is een verzameling van onderwijstechnologische specificaties van diverse organisaties (bijvoorbeeld IMS) die samen zijn gebracht in een referentiemodel dat het mogelijk moet maken om voor ontwerpers van E-learning software om content te kunnen uitwisselen tussen producten van diverse leveranciers. Voor deze standaard geldt eigenlijk dat deze te complex is om deze in Keylearning toe te passen. Deze ingewikkelde functies hebben in eerste instantie niet de hoogste prioriteit in het systeem. Wellicht kan er in een latere versie van het systeem alsnog rekening worden gehouden met deze standaarden.

De grootste bron van informatie wordt gerealiseerd in de workshop van de fase definitiestudie. De meest ideale situatie is beoogde gebruikers binnen het MKB te vragen voor een workshop. Dit bleek organisatorisch niet haalbaar. Zelfs binnen Metrix bleek het lastig te zijn geschikte mensen te vinden door de volle agenda's van deze personen. Er werd al snel duidelijk dat er weinig animo is om een workshop te organiseren met alle beoogde gebruikers. Dit is erg jammer want het was voor mij een gewenste situatie geweest omdat er gewerkt wordt met de ontwikkelmethode IAD. Dit omdat IAD zo een grote gebruikersparticipatie heeft. Opdrachtgever is echter wel bereid om één persoon af te vaardigen voor een uitgebreid interview. Deze persoon is een expert op E-learning gebied. In de voorbereiding heb ik een lange lijst met vragen voorbereid om te stellen. De geïnterviewde heeft hier echter ook een hoop aanvullingen op en uiteindelijk is het een soort dialoog geworden waarin de huidige situatie, de systeemeisen en het systeemconcept wordt blootgelegd.

3.3 Beschouwen doelen en beperkingen systeem

Binnen deze paragraaf zal worden ingegaan op de doelen en beperkingen van het systeem. Alvorens ik kan beginnen met de ontwikkeling dient er nauwkeurig in kaart te worden gebracht hoe de huidige situatie in elkaar zit en waaraan het systeem moet voldoen om aan de doelstelling van de opdracht tegemoet te komen. Aan de hand van de resultaten is het systeemconcept en de technische architectuur opgesteld.

3.3.1 Huidige situatie

De huidige situatie is moeilijk weer te geven. Dit omdat er meerdere doelgroepen voor deze applicatie zijn. Toch is het nut van de huidige situatie cruciaal, omdat met behulp van deze activiteit de problemen in kaart kunnen worden gebracht en dat is toch datgene waar ik achter wil komen. Mede aan de hand van de huidige situatie zal ik een systeemconcept kunnen samenstellen. Naar aanleiding van het interview met de opdrachtgever kunnen er vier doelgroepen worden getypeerd, te weten:

- Afdelingen van grotere organisaties met eigen leeractiviteiten
- Scholen in het primaire en voortgezet onderwijs
- Kleine opleidingsbedrijven
- Kleine- en middelgrote organisaties met eigen leeractiviteiten

U kunt zich afvragen waarom er voor deze doelgroepen gekozen is. Deze doelgroepen zijn over het algemeen bedrijven (of afdelingen van bedrijven) met een kleiner budget. Meer hierover kunt u verderop in deze paragraaf vinden.

Ter verduidelijking vindt u hieronder voor iedere situatie een beschrijving:

Afdelingen van grotere organisaties met eigen leeractiviteiten

Voor deze doelgroep heb ik een klein onderzoek gedaan bij T-mobile klantenservice b.v. De afdelingen met eigen leeractiviteiten maken over het algemeen gebruik van E-learning om medewerkers intern bij te scholen. De reden voor het intern scholen heeft vooral te maken met kosten. In enkele bedrijven speelt ook geheimhouding een rol. Omdat deze afdelingen onderdeel uitmaken van grote organisaties maken deze bedrijven over het algemeen gebruik van krachtige pakketten zoals : " Lotus LearningSpace, Docent, Whizzdom, Netg". Er is een grote mate van herbruikbaarheid met betrekking tot activiteiten/materialen. Als een activiteit voor een bepaalde cursus wordt gebruikt dan kan deze ook bij een andere cursus worden gebruikt. Bijvoorbeeld de cursus "opleiding medewerker klantenservice" zit de activiteit "communicatie deel 1". Deze activiteit kan bijvoorbeeld ook voorkomen in de cursus "succesvol presenteren" De grote mate van uitwisselbaarheid van cursusmateriaal scheelt in de kosten. Het nadeel kan wel zijn dat er weinig draagvlak is binnen sommige afdelingen zodat de inhoud leeg blijft. Over het algemeen beschikken bijvoorbeeld customer contact afdelingen over een "trainer/coach". Die enerzijds zorgt voor actieve begeleiding tijdens de cursus maar ook de voortgang bewaakt, beoordeelt en feedback geeft. Vaak kan men stellen dat het cursusbeheer geautomatiseerd verloopt. Het cursusmateriaal is (nog) niet in alle gevallen geautomatiseerd en is ook niet altijd noodzakelijk. Denk bijvoorbeeld aan een schriftelijke toets of specifiek boek. Bij een schriftelijke toets kunnen deelnemers deze schriftelijk maken en de docent kan deze in het betreffende systeem als voltooid afvinken. Het deelnemersgedeelte bestaat dan uit een soort overzicht van gevolgde en nog niet gevolgde leeractiviteiten. Dit overzicht wordt dan weer gebruikt als een soort trainingspaspoort bij interne sollicitaties en functioneringsgesprekken.

Scholen in het primaire en voortgezet onderwijs

In deze tak is op dit moment een kentering zichtbaar. Van jongs af aan wordt er geleerd met computers om te gaan en ook automatisering speelt een steeds grotere rol in onze samenleving. Bij de meeste scholen staat E-learning nog in de kinderschoenen. Veel scholen geven op dit moment computerlessen en hebben vaak een "open leercentrum". Dit is een grote computerruimte waar de leerlingen een aantal uur per week verplicht moeten doorbrengen. Bijvoorbeeld om via internet zaken op te zoeken over een opdracht die gemaakt moet worden of een opdracht uitwerken in een tekstverwerkingsprogramma. Het alleen hebben van een computerlokaal waar computerles gegeven wordt is niet de definitie die voor deze opdracht aan E-learning is gegeven. 93% procent van de docenten maakt gebruik van computers bij de lessen. "Echte" E-learning waarbij het leerproces wordt beheerd wordt op dit moment slechts bij 11% van de scholen gebruikt. Het gaat hierbij dan vaak om standaardoplossingen die worden aangeboden door allerlei softwareontwikkelingsbedrijven (Three ships met N@tschool, Netg, Blackboard) waarbij het aanbod groot is maar de overzichtelijkheid klein. Aangezien het om grote uitgaven gaat en een behoorlijk implementatieproces zijn scholen hier voorzichtig mee.

Verder speelt er een organisatorisch probleem, bij een E-learning systeem waarbij het mogelijk is om courses te volgen en te beheren zal de docent een andere functie krijgen dan waar hij oorspronkelijk voor is aangenomen. Nu is het bekend dat vernieuwingen in bedrijven vaak leiden tot weerstand. Voor deze docenten is het daadwerkelijk gebruiken van het E-learning systeem een (te) grote stap. Dit is één van de redenen dat E-learning in het onderwijs nog in de kinderschoenen staat.

Kleine opleidingsbedrijven

Er zijn in Nederland enorm veel kleine opleidingsbedrijven. Over het algemeen zijn deze bedrijven gespecialiseerd in een bepaald type opleidingen. Deze groep is onder te verdelen in drie subgroepen. De bedrijven die nog geen gebruik maken van E-learning, de bedrijven die een oplossing op maat hebben (voor veel geld) en bedrijven die gebruik maken van een duur standaardpakket zoals whizzdom. Over het algemeen zijn dit cursussen voor particulieren, danwel individuele cursisten die op verzoek van hun werkgever een cursus volgen. Afronding geschiedt vaak door middel van het halen van een diploma of certificaat, welke aan de werkgever getoond kan worden.

Kleine- en middelgrote organisaties met eigen leeractiviteiten

In deze bedrijven zijn de courses in eigen beheer. Dit doen bedrijven als zij veel courses moeten geven (besparing in de kosten) of wanneer er sprake is van geheimhouding. Bij sommige bedrijven is de "docent" niet altijd een fulltime trainer. Het kan bijvoorbeeld ook een manager zijn die de trainingen verzorgt. Zoals ook genoemd bij paragraaf 5.1 zijn deze courses vaak bedoeld om de medewerkers bij te scholen of een nieuwe medewerker klaar te maken voor zijn/haar taken. Omdat het hier meestal om kleine courses gaat met een geringe complexiteit is een uitgebreide (en dus te dure) applicatie niet nodig. Dit zijn echter de software pakketten die deze doelgroep afneemt omdat het aanbod zo beperkt is (Whizzdom, docent).

Ik heb in samenspraak met de opdrachtgever gekozen om mij op deze doelgroepen te richten. Hoewel de doelgroepen verschillend zijn, is het gebruik van de applicatie door de doelgroepen vrijwel hetzelfde. Er zijn ook een aantal groepen die niet binnen de reikwijdte van het systeem vallen, zoals HBO instellingen, Universiteiten en grote opleidingsbedrijven. Dan rijst natuurlijk de vraag, waarom juist deze doelgroepen? Het heeft vooral te maken met vraag en aanbod op de markt. Bij de hogescholen en universiteiten is er veel aanbod en leren vormt hier het primaire proces waardoor budgetten hiervoor groter zijn. Bij de doelgroepen voor het te bouwen systeem is er juist weinig aanbod (met een goede prijs-kwaliteitverhouding) maar wel veel vraag. Ook hebben grote bedrijven vaak al langlopende contracten en is het niet mogelijk om een uitgebreide applicatie aan te bieden voor een prijs waarvoor Keylearning verkocht gaat worden. Het was dus "of/of" en volgens Metrix liggen bij de vier gekozen doelgroepen de meeste kansen.

De applicatie moet zoveel mogelijk aan de eisen van de doelgroepen voldoen. Om deze doelgroepen over te halen om het systeem aan te schaffen moet het allereerst laagdrempelig zijn voor de gebruikers en tevens een uitdagend lage prijsstelling hebben. De laagdrempeligheid van het systeem zal verder worden uitgewerkt in de volgende paragraaf.

3.3.2 Systeemeisen

Na het in kaart brengen van de huidige situatie worden de systeemeisen gedefinieerd. Hiermee worden de behoeften van de toekomstige gebruikers weergegeven en daarnaast vormen de systeemeisen de basis voor alle ontwerpactiviteiten. De systeemeisen zijn opgesteld aan de hand van het interview dat is afgenomen met de opdrachtgever. Ook na dit interview zijn er nog kleine aanpassingen heen en weer gemaakt waardoor eventuele toekomstige iteraties worden voorkomen.

Zoals IAD voorschrijft zijn de eisen gespecificeerd op drie niveaus, te weten *basis*, *comfort* en *luxé*. In samenspraak met de opdrachtgever heb ik besloten om alle eisen die met functionaliteit te maken hebben de hoogste prioriteit te geven. De eisen met betrekking tot hoe een functionaliteit te benaderen hebben over het algemeen de status *comfort* gekregen en maken van rapporten hebben de status *luxé* gekregen. Hoewel de

comforteisen ook belangrijk zijn heb ik gekozen om de prioriteit bij de functionaliteit te leggen. Als ik mij bijvoorbeeld in eerste instantie op de eis "veelgebruikte functies dienen vanuit het menu te benaderen zijn" zou richten, dan zou er een schitterend mooi menu zijn, maar de gerelateerde functies zijn dan nog niet gebouwd. Dit is geen gewenste situatie. Sommige functionaliteiten staan ook bij *comfort*. Dit omdat binnen de functionaliteit ook nog een prioritering is. Het maken van rapporten is een luxe-eis geworden. De reden hiervan is dat het maken van goede rapporten complex is. Het maken van standaard rapporten is natuurlijk mogelijk met HTML maar voor de wat uitgebreidere rapporten dient er gebruik te worden gemaakt van een extern webobject zoals crystal reports. De verantwoordelijkheid hiervoor ligt bij de afnemer van het product die zelf beslist of hij deze module aan wil schaffen. Te zijner tijd zal moeten worden gekeken naar de haalbaarheid hiervoor maar het heeft op dit moment niet de hoogste prioriteit.

De eisen waarin ik kritische beslissingen heb moeten nemen zal ik hieronder uitdiepen. Een complete lijst met eisen is na deze beschrijving te vinden.

Ik heb ervoor gekozen om het systeem optimaal te maken voor 1024x768. Natuurlijk zijn de meeste andere resoluties ook te gebruiken maar ik heb ervoor gekozen om hem voor de meest gebruikte resolutie te optimaliseren.

Om de twee rollen in de applicatie (docent en student) zo goed mogelijk gescheiden te houden zullen er twee aparte modules worden gemaakt. Je hebt dus te maken met twee hoofdschermen. Het is in principe ook mogelijk om één hoofdscherm te maken waarbij de studenten geen toegang tot docentfunctionaliteit hebben door middel van knoppen "disabled" zijn. Omdat de docentmodule veel functies bevat, zou het studentgedeelte zeer veel functies hebben die niet te benaderen maar wel te zien zijn. Dit is natuurlijk frustrerend voor een gebruiker. Vandaar dat ik niet voor deze optie gekozen heb.

De hoofdpagina moet in maximaal 10 seconden geladen worden bij een modale (ISDN) verbinding. Ik heb dit niet wetenschappelijk onderzocht maar collega's bij Metrix deelden in ieder geval mijn mening hierover. 10 seconden lijkt lang maar ik heb hiervoor gekozen omdat je dan ook de mogelijkheid hebt om bijvoorbeeld JAVA applets te gebruiken. Alleen als Java applets een toegevoegde waarde hebben zullen deze gebruikt worden aangezien de aanpasbaarheid van een JAVA applet te wensen over laat. Verder is het zo dat het internet steeds sneller wordt. Als het laden van een pagina nu tien seconden duurt, dan zal dit over een paar jaar wellicht gehalveerd zijn.

Een andere keuze die ik heb gemaakt is om de applicatie alleen geschikt te maken voor Internet Explorer. Op zich is dit een beperking aan de gebruikerskant maar het is een verrijking aan de ontwikkeltkant. Het is namelijk zo dat het aanspreken van DHTML objecten via javascript meer wordt ondersteund door Internet Explorer dan welke browser dan ook. Aangezien de Keylearning applicatie hier zeker gebruik van gaat maken heb ik voor Internet Explorer gekozen. Internet explorer heeft een marktaandeel van bijna 95% waardoor de gevolgen aan de gebruikerskant wel te overzien zijn. Groot voordeel is natuurlijk ook versiebeheer. Omdat er niet een aparte versie voor Internet Explorer en de andere browsers onderhouden en ontwikkeld hoeft te worden, scheelt dit in de kosten.

Om een goede indruk te krijgen van het systeem dat gebouwd gaat worden volgt op de volgende bladzijde een opsomming van de systeemeisen:

Basissysteem-eisen

- Het systeem moet een grafische weergave kunnen geven van gevolgde courses en nog te volgen courses voor een student. (basis)
- Een gebruiker moet ingelogd zijn om van het systeem gebruik te maken. (basis)
- Een gebruiker moet "docentrechten" hebben om in het docentgedeelte te komen. (basis)
- Een course heeft 0 of meer activiteiten. (basis)
- Een student kan in meerdere groepen zitten. (basis)
- Een groep heeft 0 of meer studenten. (basis)
- Een activiteit kan in meerdere courses voorkomen. (basis)
- Een student moet de mogelijkheid hebben om courses te volgen en eventueel de status hiervan te wijzigen (indien permissie hiertoe). (basis)
- Het systeem moet laagdrempelig zijn, en moet ook als zodanig in een test worden beoordeeld (basis)
- Veel gebruikte functies dienen vanuit het menu te benaderen te zijn. (comfort)
- Een student dient een rapport te kunnen genereren van zijn eigen resultaten (luxé)
- Een student moet de mogelijkheid hebben zijn paswoord te wijzigen (indien permissie). (luxé)
- Een activiteit kan pas worden opgestart als aan de startvoorwaarden (prerequisites) is voldaan. (comfort)
- Als de docent setByAdmin heeft geactiveerd voor een taak, dan kan alleen een docent aangeven dat deze taak is voltooid. (comfort)
- Als een activiteit is voltooid, dan wordt hiervan de dag en tijd genoteerd. (comfort)
- Het aantal malen dat een activiteit door een student mag worden geprobeerd alvorens een succesvol resultaat te halen kan door de docent worden beperkt. (comfort)
- Een docent kan studenten aanmaken, wijzigen en verwijderen (basis)
- Een docent kan activiteiten aanmaken, wijzigen en verwijderen (basis)
- Een docent kan courses aanmaken, wijzigen en verwijderen (basis)
- Een docent kan aan een course activiteiten linken en weer verwijderen. (basis)
- Een docent kan rapporten genereren per course, activiteit en per gebruiker. (luxé)

Interface-eisen

- Het systeem moet bestaan uit twee aparte hoofdschermen (student en docent) (basis)
- De schermen moeten passen in een resolutie van 1024X768. (basis)

Integriteits-eisen

- Een student mag pas worden verwijderd als deze geen open courses heeft. (comfort)
- Een activiteit mag alleen worden verwijderd als deze niet aan een course meer is gelinkt. (comfort)

Performance-eisen

- De laadtijd van een pagina mag niet meer dan 10 seconden in beslag nemen uitgaande van een 64k isdn verbinding. (basis)

Operationele-eisen

- Het systeem dient tweetalig (Nederlands, Engels) te zijn. Om deze reden moeten alle gebruikte labels, boodschappen en error messages centraal zijn opgeslagen. (luxe)
- De functionaliteit binnen de data laag en business laag moet volledig herbruikbaar zijn. (comfort)
- Het systeem moeten kunnen worden gehost vanuit Microsoft Internet Information Server. (basis)
- Het systeem moet kunnen draaien in Internet Explorer 5.5 en hoger. (basis)
- Eenvoudige rapporten kunnen met HTML worden gegenereerd, complexere rapporten dienen te worden gemaakt door crystal report webobject (zelf door de klant aan te schaffen). (comfort)

Figuur 3.3 systeemeisen

3.3.3 Systeemconcept

Het systeemconcept bestaat uit een aantal onderdelen. Allereerst de gebruikersgroepen. Voor elk van deze gebruikersgroepen wordt in kaart gebracht welke taken door het systeem dienen te worden geautomatiseerd.

Actoren

Allereerst heb ik de actoren van het systeem gespecificeerd. Hiervoor had ik twee opties. twee actoren (docent/student) of drie actoren (beheerder/docent/student). In de situatie met 2 actoren is de docent tevens de beheerder van het systeem. In de situatie met drie actoren is dit een apart persoon met meer rechten dan een docent. Deze persoon houdt zich dan bezig met passwordmanagement, invoeren van courses en activiteiten.

Ik heb gekozen voor 2 actoren (student en docent) om de volgende reden(en). Vanwege de laagdrempeligheid zal het systeem niet te complex worden, beheersfuncties kunnen prima door een docent gedaan worden. Tevens is het zo dat wanneer er een aparte beheersfunctie is de docent erg weinig te doen heeft behalve resultaten van studenten invoeren. Tevens is hij sterk afhankelijk van de beheerder voor het plaatsen van zijn leermateriaal. Hij heeft hierdoor niet alle touwtjes in handen en beperkte flexibiliteit. Als een docent weinig te doen heeft en beperkt wordt in zijn handelen zal hij zeer waarschijnlijk de applicatie niet met plezier gebruiken en deze situatie is ongewenst.

Taakdiagrammen

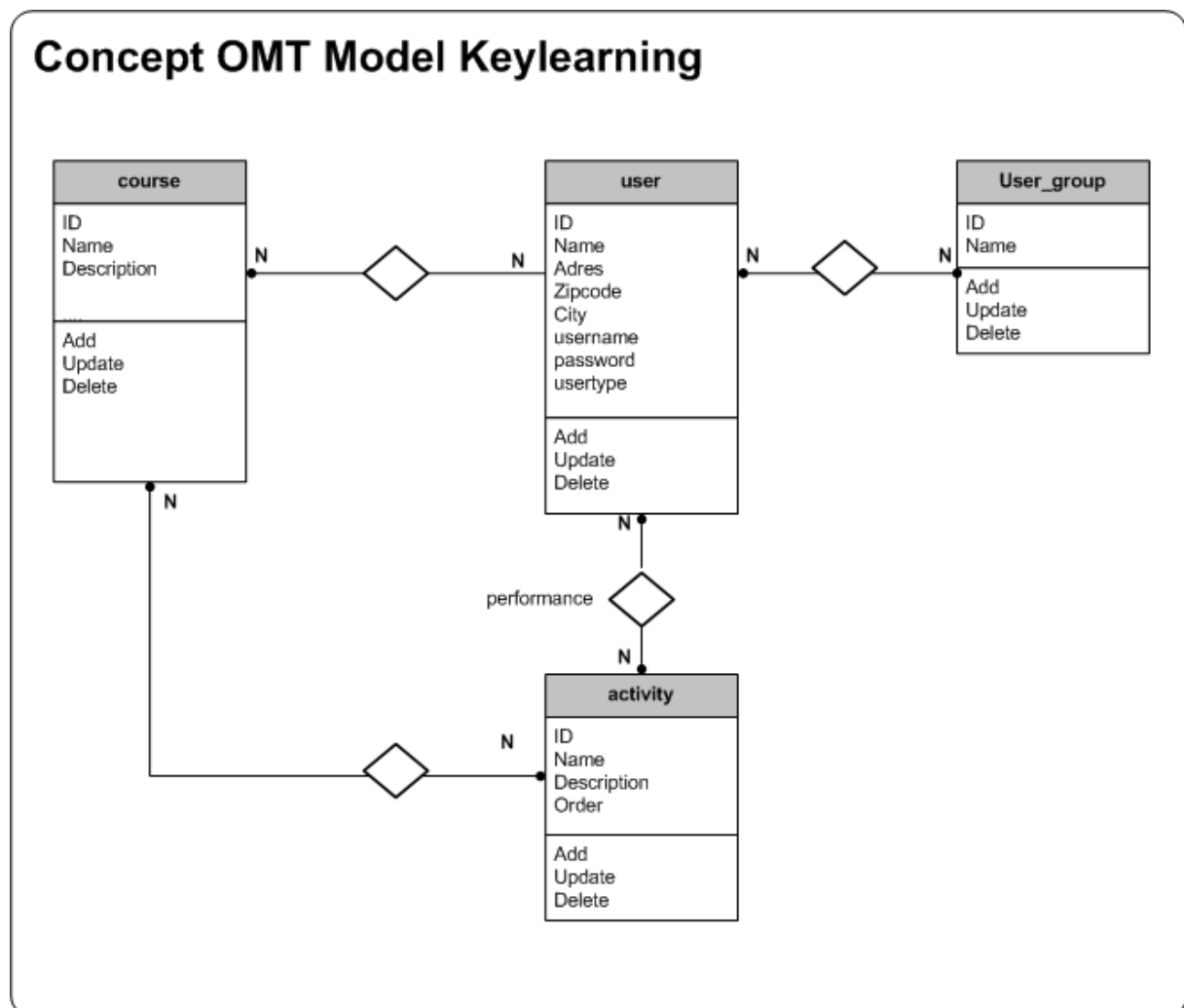
Ik heb voor de meest voorkomende taken taakdiagrammen gemaakt. De informatie van de scenario's is vergaard uit het interview bij aanvang van de definitiestudie. De taken die door de gebruiker worden gedaan zijn niet erg complex en om deze reden zijn de taakdiagrammen op een globaal niveau uitgewerkt. In de fase pilotontwikkeling zal specifiekere worden ingegaan op de taken van de gebruikers. Dit door middel van use cases, procesmodellen en navigatieschema's.

Objecten

Na het analyseren van de actoren en de taakdiagrammen heb ik de objecten kunnen vastleggen.

Het vastleggen van objecten is zeer belangrijk in het concept, daarom heb ik reeds in deze fase ervoor gekozen om een eerste versie van het objectmodel te maken. Het is zo dat het in een learnmanagement systeem draait om vier objecten:

- Course (een set activities)
- Activity (een taak die onderdeel is van course)
- User (een gebruiker)
- User_group (een groep met 0 of meer gebruikers)



Figuur 3.4 concept OMT model

Ik heb ervoor gekozen om activity geen eigenschap te laten zijn van het object course. Als ik dit zou doen zou ik het systeem een beperking opleggen. Een course zou in dat geval een collectie activities bevatten. Een activity kan dan slechts tot één course behoren. Ik heb gekozen voor een handige optie. In het geval van het geschetste objectmodel is een activity geen onderdeel van course maar een opzichzelfstaand iets. Activity A kan nu voorkomen in course X, maar ook in course Y. Hergebruik van een activiteit is hierdoor dus mogelijk.

Concreet gezegd: in course "opleiding medewerker frontoffice" zit de activiteit "lees blz. 10 van het boek communicatie voor beginners". Deze activity kan nu ook voorkomen in de course "opleiding medewerker E-mail team".

Eigenlijk heb ik precies hetzelfde gedaan bij de objecten user en user_group. Een user kan in dit geval behoren tot meerdere groups en een group kan uiteraard meerdere users bevatten.

Meer over de objecten en de daaruit vervaardigde database vindt u in de twee pilotontwikkelplannen in de bijlagen of later in dit verslag onder hoofdstuk 4

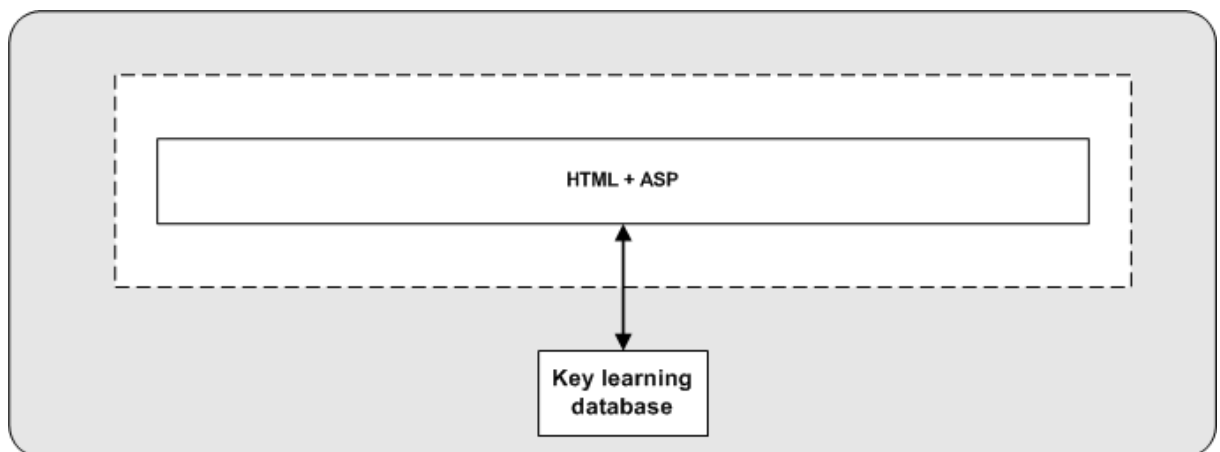
Technische architectuur

Het tweede onderdeel van het systeemconcept bestaat uit de beschrijving van de technische architectuur. Deze technische architectuur heeft als doel de fundamenteën van het te ontwikkelen systeem te beschrijven. Er wordt hier gekozen voor een manier waarop het systeem zal worden opgebouwd. Tevens wordt de technische architectuur gebruikt om de verschillende pilots samen te stellen, omdat uit de technische structuur de verschillende systeemonderdelen herkend kunnen worden.

Reeds in de opdrachtomschrijving was bekend dat de applicatie via het web zou moeten draaien. Voor het web waren er voor mij twee opties om het systeem te bouwen, te weten:

1. De applicatie niet onderverdelen in lagen

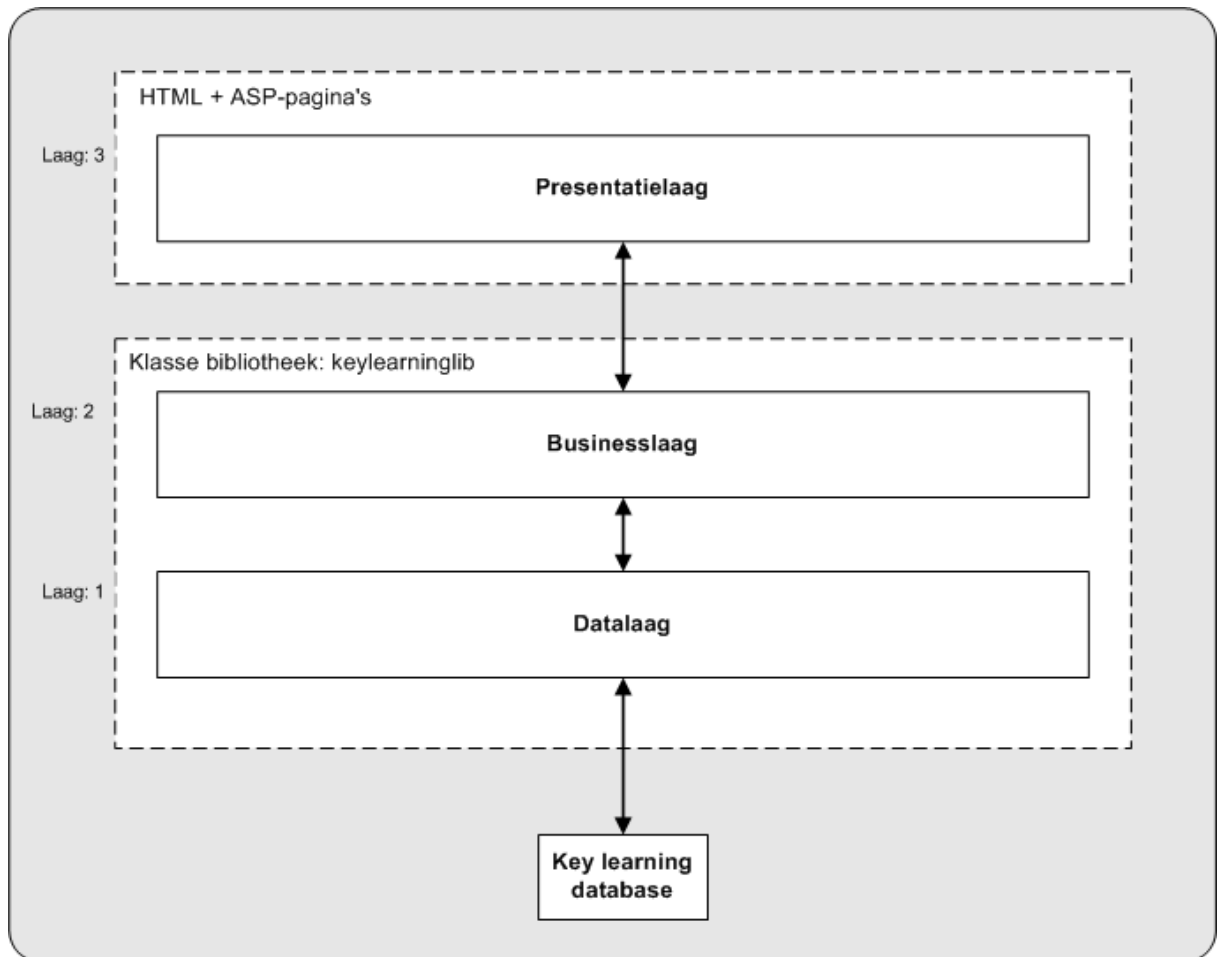
Bij de variant wordt de functionaliteit gemaakt in een "server side scripttaal" en de presentatielaag door middel van HTML en Javascript. Dit heeft als voordeel dat je de gehele applicatie kunt schrijven in één tool (frontpage, notepad).



Figuur 3.5 applicatie niet onderverdeelt in lagen

2. De applicatie onderverdelen in drie lagen

Bij de variant wordt de functionaliteit geschreven in een 4^{de} generatie programmeertaal. Deze wordt dan onderverdeeld in een data laag en een business laag. De koppeling van de business met de presentatielaag gebeurt in dit geval in de server side scripting taal. De presentatielaag wordt geschreven in HTML en Javascript. Dit heeft als voordeel dat de code beter te herbruiken is, beter te onderhouden en makkelijker aan te passen is.



Figuur 3.6 applicatie onderverdeelt in lagen

Onderhoudbaarheid, herbruikbaarheid en aanpasbaarheid zijn zeer belangrijk. Dit omdat in tegenstelling tot andere producten het met software mogelijk (of zelfs noodzakelijk) is aanpassingen of onderhoud uit te voeren op je bestaande programmatuur. Dit om de kwaliteit van je systeem te waarborgen. Doordat je de programmatuur openstelt voor aanpasbaarheid en onderhoudbaarheid wordt de tijd en kosten die daaraan zijn gerelateerd geminimaliseerd. Herbruikbaarheid vindt zijn voordeel bij de ontwikkeling van nieuwe programmatuur. Waarom iets gaan ontwikkelen wat al reeds bestaat?

Als ontwikkelomgeving heb ik gekozen voor microsoft visual studio 6.0. Een goed alternatief is visual basic 7 dat zich in het .Net platform bevindt. Microsoft .NET is weliswaar recenter, maar heeft voor het ontwikkelen van Keylearning weinig toegevoegde waarde. Bovendien is de syntax en denkwijze van .NET totaal anders. Aangezien ik wel wat ervaring heb met visual basic was de keuze gemakkelijk. Door te programmeren in een taal die je kent boek je wat tijdswinst. Verder is het zo dat Metrix nog geen licentie had voor microsoft .net. In visual basic wordt het werken met objecten

ondersteund en is het door het gebruik van modules mogelijk om de code gestructureerd te houden. Door het ontwikkelen van procedures en methodes die vaker door de applicatie heen te gebruiken zijn heeft dit systeem ook een grote mate van herbruikbaarheid.

Als server side scripting taal is er gekozen voor ASP. Een alternatief is PHP. Omdat de data en businesslaag worden ontwikkeld in microsoft visual basic heb ik gekozen voor ASP. Omdat ASP ook een microsoft product is kunnen de lagen met elkaar communiceren. Sterker nog, het is de enige mogelijkheid om vanuit ASP de objecten van de data- en businesslaag (lees dll's) aan te spreken.

Database

Voor het data management systeem heb ik gekozen voor msSQL server. Een goed alternatief is msAcces. Ik heb gekozen voor msSQL omdat msSQL over het algemeen een beter performance heeft dan access. Verder is msSQL wat makkelijker in gebruikersbeheer. Omdat msSQL over tcp/ip draait kan vanaf iedere plek verbinding worden gemaakt.

Drie lagen model

Zoals eerder gezegd bestaat het systeem uit drie lagen met elk hun eigen functie. Om het model te begrijpen zal ik de functies van de drie verschillende lagen beschrijven.

De Business- en data laag bevinden zich in een klassenbibliotheek, genaamd Keylearninglib. Deze klassenbibliotheek bestaat op haar beurt uit een DLL-bestand (Dynamic Link Library). Een DLL-bestand wordt onder het Microsoft Windows platform gebruikt om referenties te kunnen leggen naar de objecten binnen die klassenbibliotheek. Op deze wijze hebben ontwikkelaars de mogelijkheid reeds bestaande programmatuur te hergebruiken.

Data laag

Deze laag heeft als doel de integriteit van het systeem te bewaken. Indien er schrijfacties worden uitgevoerd op de database, dan worden deze schrijfacties gecontroleerd zodat er geen inconsistentie binnen de database optreedt. Ik zal een voorbeeld geven om dit te verduidelijken:

Wanneer een user uit het systeem wordt verwijderd, dan controleert deze laag eerst of deze user nog openstaande activiteiten heeft. Is dit het geval dan wordt de schrijfactie geannuleerd.

Hierdoor voorkomt de data laag dat er records in de database staan die niet meer van toepassing zijn.

Business laag

Deze laag heeft als doel de systeemfunctionaliteit aan de bovenliggende laag aan te bieden. Deze systeemfunctionaliteit beschikt over allerlei procedures die het mogelijk maken de gegevens te overhandigen aan een bovenliggende laag. In de procesbeschrijving van de beiden pilots (de hoofdstukken vier en vijf) zal dieper worden ingegaan op de inhoud van de businesslaag.

Presentatielaag

De presentatielaag heeft als doel de gegevens van het systeem te tonen aan de gebruiker. In het geval van Keylearning bestaat de presentatielaag uit tal van internet pagina's, die door middel van ASP (Active Server Pages) kunnen communiceren met de businesslaag.

De interactie met de gebruiker zal gebeuren met behulp van JavaScript. JavaScript is een zogeheten Client-Side script, dat wil zeggen dat het script wordt uitgevoerd door de browser. De presentatielaag moet ook worden gekoppeld aan de businesslaag. Om dit te bewerkstelligen maak ik zoals eerder genoemd gebruik van ASP. Naast de te gebruiken programmeertalen zal de opmaaktaal HTML gebruikt worden om de gebruikers interface te realiseren.

3.3.4 Organisatorische inrichting

De volgende stap is het in kaart brengen van de organisatorische inrichting. Bij de invoering van een Learning Management System zal de inrichting van het werk totaal anders worden. Bij een bedrijf met weinig computerervaring moeten de medewerkers allereerst geschoold worden. Hoewel Keylearning laagdrempelig is, is er toch een redelijke kennis van computers en de nieuwe manier van werken vereist. Omdat de inrichting van het werk anders wordt zal dit gaan stuiten op weerstand. Vooral voor een docent heeft dit grote gevolgen, Als een docent zijn lessen via Keylearning zal aanbieden zal dat voelen alsof hem zijn werk wordt ontnomen. Terwijl het eigenlijk zo is dat zijn werk anders wordt ingericht. Keylearning zal zo ontwikkeld moeten worden dat een docent zich zo comfortabel voelt om mee te werken en dat er een draagvlak voor deze applicatie wordt gecreëerd.

Het laagdrempelig maken van een systeem is lastig te definiëren. Naar mijn mening moet er vooral gelet worden op het feit dat de applicatie niet te complex is/wordt. Verder is het taak van het management van het bedrijf om te kijken of de applicatie voldoende wordt gebruikt, als dit niet het geval is, dienen de medewerkers hiertoe te worden aangespoord.

3.4 Pilotplan

De laatste activiteit van de fase definitiestudie is het opstellen van het pilotplan. Het pilotplan heeft als doel in kaart te brengen hoe het verdere verloop van dit project eruit komt te zien. Eenmaal in de fase pilotontwikkeling wordt er daadwerkelijk gebouwd aan het systeem. Alvorens ik daar aan ga beginnen ga ik het te ontwikkelen systeem opdelen in zogeheten pilots (systeemdelen). Het totaal van de pilots vormt op haar beurt weer de Keylearning applicatie.

3.4.1 Verdeling Pilots

Reeds bij het maken van de opdrachtomschrijving was bekend dat de applicatie uit één kan worden gezet in twee modulen. Er zal een studentgedeelte komen en een docentgedeelte. Het is dan ook een logische stap om deze twee zaken als aparte pilots te ontwikkelen. Nu kunnen de volgende pilots onderscheiden worden:

Pilot student

Binnen de pilot "student" zal het gedeelte van de pilot worden gemaakt waar courses kunnen worden gevolgd en waar door de gebruiker de resultaten kunnen worden bekeken. In deze pilot wordt begonnen met het ontwerp van de objectlibrary (datalaag),

vervolgens zal de database worden ontworpen, hierna de businesslaag en tenslotte de presentatielaag.

Pilot docent

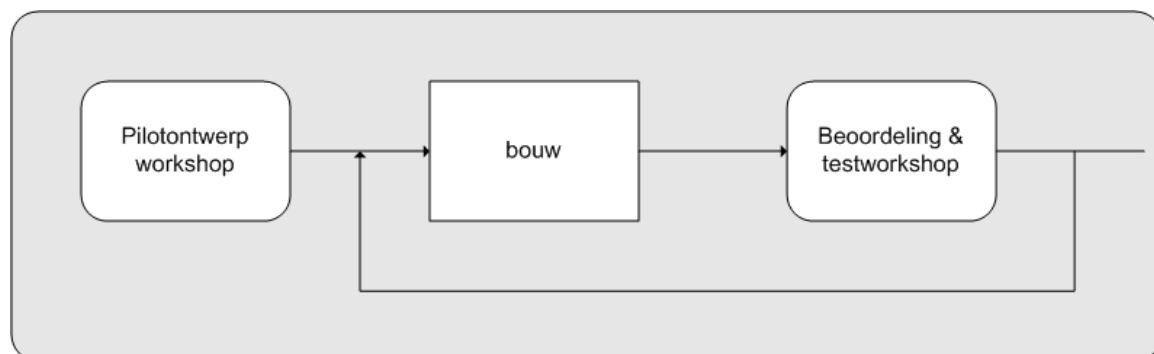
In de pilot "docent" kunnen alle beheertaken worden uitgevoerd van de applicatie. Dat wil zeggen user management, course management, activity management en rapporten. In deze pilot wordt begonnen met het ontwerp van de database, vervolgens zal de data laag worden ontworpen, hierna de businesslaag en tenslotte de presentatielaag. Aangezien een groot gedeelte van de database van de pilot "student" kan worden gebruikt (er is tenslotte maar één database) zal dit gedeelte in verhouding niet zoveel tijd beslag nemen.

De vraag is nu welke pilot eerst? In principe maakt het voor het te bouwen systeem niet zo heel veel uit. Beide pilots zullen af moeten zijn om een compleet systeem te hebben. Pilot "docent" bevat meer functionaliteit (en is dus ook complexer) dan de pilot "student". Omdat ik voor aanvang van dit project al een tijd niet meer geprogrammeerd heb leek het mij handiger om te beginnen met de gemakkelijkste pilot, in dit geval de pilot "student".

De pilot "docent" heeft wel de hoogste prioriteit. Dit omdat dit nu eenmaal de grootste pilot is. Daarom heb ik besloten de module student niet te laten uitlopen. Als hij volgens de planning de fase pilot "student" klaar had moeten zijn dan wordt de pilot afgerond. Ongeacht of hij al af is of niet. De rest van deze pilot kan dat in de fase test, herstel en invoering worden vervaardigd. Als ook blijkt dat de pilot "docent" niet afkomt heb ik ervoor gekozen om zoveel mogelijk aan de presentatielaag te bouwen. Dit omdat Metrix na afloop wat demo's wil geven met het systeem. Als ik bijvoorbeeld alleen de data- en de businesslaag af zou maken en niet meer aan de presentatielaag zou toekomen dan valt er weinig te laten zien in een demo. Dit wil dus zeggen dat als ik tijdens het ontwikkelen van de pilot docent het gevoel krijg dat ik deze niet afkrijg, ik me zal moeten richten op louter ontwerp van de presentatielaag.

3.4.2 Pilotontwikkelstrategie

Voor ontwikkelstrategie heb ik gekozen voor "iteratie van bouw & beoordeling & test" en niet voor "volledige iteratie binnen de fase". Dit omdat er binnen de gestelde tijd geen tijd voor de betrokkene(n) is om nogmaals een workshop bij te wonen. Ter illustratie een afbeelding van deze variant:



Figuur 3.7 iteratie van bouw & beoordeling & test

Bij deze variant wordt de ontwerp workshop slechts één keer uitgevoerd. Aangezien deze pilots te overzien zijn is dit niet een al te groot probleem.

Ik heb besloten om voor beide pilots slechts één iteratie toe te staan, de pilots worden éénmaal getest en hierna kunnen de pilots worden aangepast. Alternatieven zijn er genoeg. Je kunt kiezen voor twee, drie of oneindig veel iteraties. Wat de kwaliteit van het product wellicht verbetert. Kiezen voor meer iteraties past beter bij projecten waarbij de factor tijd een ondergeschikte rol speelt. Ik heb voor één iteratie gekozen omdat je anders lang in de iteratie blijft steken. Eventuele aanpassingen kunnen in de herstel fase worden uitgevoerd.

3.4.3 Planning

Na inschatten van de verschillende bouwonderdelen ben ik tot de volgende planning gekomen:

- Pilotontwikkelpun student 2 weken
- Pilot student 3,5 weken
- Pilotontwikkelpun docent 2,5 weken
- Pilot docent 4 weken

Ik heb ervoor gekozen om voor de pilot student minder tijd uit te trekken dan voor de pilot docent omdat deze kleiner is. Dit geldt zowel voor de bouw van de pilots alsmede voor de pilotontwikkelpunnen.

Voor de prioritering van de pilotdelen geldt dezelfde prioritering zoals in de fase definitiestudie bij de systeemeisen vermeld is.

3.4.4 Testen

In eerste instantie leek het mij handig om de applicatie te testen door de één persoon binnen Metrix. In dit geval zou dat de bedrijfsmentor zijn omdat hij alle ins en outs kent van de opdracht. Ik ben hier op teruggekomen omdat het handiger is om te testen met "objectieve" personen. Omdat Metrix zelf het systeem ook gaat gebruiken is het meest voor de hand liggende om de medewerkers van dit bedrijf te laten testen. De pilots worden op het intranet geplaatst en hierop zal feedback worden gegeven door de medewerkers. Een meer gedetailleerde beschrijving over het proces testen is te vinden in paragraaf 4.5 en 5.5

Deel II Bouwfase

4 Pilot student

Zoals eerder beschreven zal de bouwfase van de applicatie onderverdeeld worden in twee pilots. Een pilot is een coherente subset van het uiteindelijke beoogde informatiesysteem die als zelfstandige, bruikbare eenheid kan worden ingevoerd in de organisatie. Omdat reeds in de fase “plan van aanpak” bekend werd dat het systeem op te delen is in twee onderdelen is de keuze voor twee aparte pilots logisch.

In de eerste paragraaf komt de *definitie* aan de orde, in de volgende paragraaf *planning* zal de planning nader onder de loep worden genomen. De paragraaf *workshop* beschrijft hoe de workshop verlopen is. In de paragraaf *pilotontwikkelplan* zal ik de totstandkoming van dit document verklaren. De paragraaf *realisatie* beschrijft het bouwgedeelte van de pilot en in de paragraaf *testen* zal de test van deze pilot worden toegelicht. Tot slot komen in de paragraaf *aanpassingen* de aanpassingen aan bod die ik na de test heb uitgevoerd.

4.1 Definitie

De definitie van de pilot student is als volgt: “De pilot student is beveiligd door middel van verificatie van gebruikersnaam en password. Als de student ingelogd is, is het mogelijk om openstaande en reeds afgeronde leertrajecten te bekijken. Van deze leertrajecten kan de student de activiteiten uitvoeren en hierbij gebruik maken van bijbehorende leermiddelen. De student kan met de manager van het leertraject (een docent) communiceren door middel van een forum of prikbord. Verder kan de student zijn persoonlijke gegevens wijzigen indien dit gewenst is. De student kan zijn eigen voortgang bewaken door interpretatie van zijn eigen resultaten. Indien gewenst mag de student zich zelf inschrijven voor leertrajecten en ter ondersteuning van het proces kan de student documenten uploaden en downloaden”

4.2 Planning

Op basis van de planning uit het plan van aanpak, zou deze pilot van het begin tot het eind zes weken in beslag nemen:

Activiteit per week	1	2	3	4	5	6
Stel plan van aanpak op						
Bereid workshop voor						
Voer workshop uit						
Evalueer workshop						
Stel pilotontwikkelplan op						
Bouw pilot						
Test pilot						
Evalueer test						
Pas pilot aan						

Figuur 4.1 planning pilot student

Zoals eerder gemeld heb ik reeds in de definitiestudie besloten om voor de pilots slechts één iteratie toe te staan. De pilot wordt dus eerst gebouwd en vervolgens door testgebruikers van Metrix getest. Hierna is er voor mij nog de gelegenheid voor het maken van aanpassingen. Als er zaken uit de test komen die niet binnen de gestelde tijd aangepast kunnen worden dan worden deze activiteiten verschoven naar de fase “test, herstel en invoering”

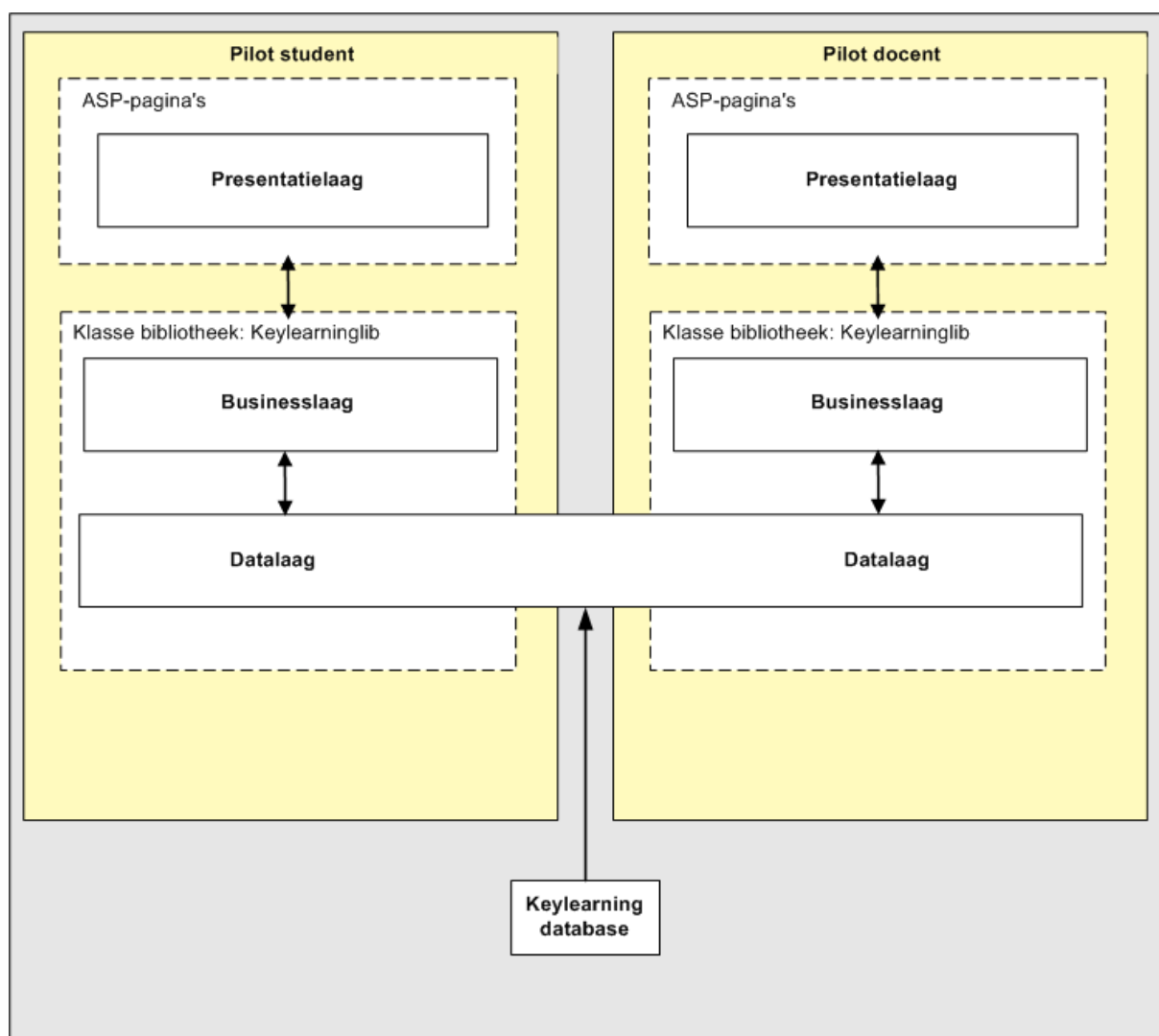
4.3 Workshop

Voorafgaand aan het bouwen is er een workshop. Althans IAD schrijft voor dat er een workshop moet worden gehouden om specificaties nog helderder en specifiekere te krijgen dan zij reeds in de definitiestudie al waren. Jammer genoeg was het door gebrek aan beschikbare mensen niet mogelijk om een workshop te organiseren. Ik had zelf het idee dat een sessie met meerdere mensen de meeste input zou genereren. In plaats van een workshop is er een interview gepland met mijn bedrijfsmentor dhr. Maas. In dit interview kwam vooral ter sprake wat het systeem uiteindelijk moet doen en hoe het schermontwerp eruit moet komen te zien. Dit kwam vooral ook door de actieve houding van de geïnterviewde, behalve een interview is ook flink gediscussieerd over de (on)mogelijkheden van het systeem. Na dit gesprek heb ik ook nog twee medewerkers van Metrix los van elkaar kunnen onderwerpen aan een kort interview. Hier was ik content over omdat deze met een andere bril naar de te bouwen pilot kunnen kijken dan opdrachtgever en ontwikkelaar.

4.4 Pilotontwikkelplan

In het pilotontwikkelplan wordt vastgelegd *hoe* de pilot gebouwd gaat worden. De meeste input komt uit het eerder beschreven interview met dhr. Maas. Na het plan van aanpak is de eerste activiteit het vastleggen van de objecten. Omdat ik reeds in de definitiestudie een begin had gemaakt met het ontwerp van deze objecten begon ik niet helemaal bij af maar kon ik hierop voortborduren.

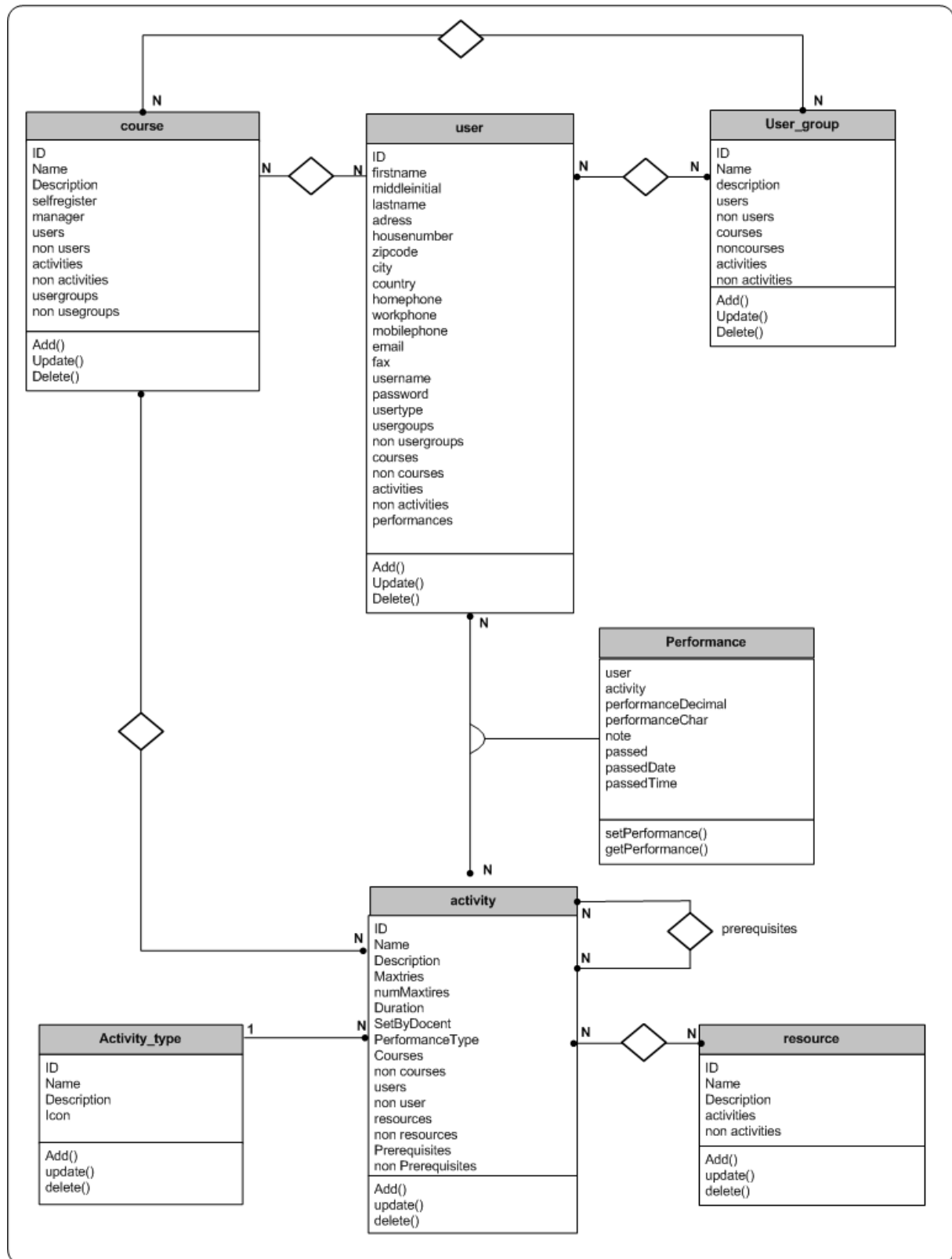
Op dit moment kwam ik erachter dat in de planning een denkfout heb gemaakt. Ik heb over het hoofd gezien dat alvorens er aan de pilot student wordt begonnen het noodzakelijk is om de objecten voor de gehele applicatie vast te leggen, je werkt tenslotte ook maar uit één database. Het is zo dat alle objecten in beide pilots gebruikt worden. Ook de meeste checks die de integriteit bewaken kunnen voor beide pilots gebruikt gaan worden. De data laag bestaat nu juist uit deze twee onderdelen (vastleggen objecten en integriteitschecks). Om deze reden heb ik besloten om de bouw van de objecten, de database en de data laag voor beide pilots te maken. Ter verduidelijking een afbeelding van de relatie van de pilots tot elkaar:



Figuur 4.2 relatie pilots tot elkaar

Dit heeft wel als gevolg dat de pilot student een week langer in beslag neemt. Omdat ik in de fase "pilotontwikkeling docent" niet opnieuw deze zaken hoeft te doen zal er daar waarschijnlijk wat tijd worden ingelopen.

Om een goed inzicht te krijgen in de opbouw van het te bouwen systeem volgt hieronder een afbeelding van het objectmodel:



Figuur 4.3 objectmodel Keylearning

De klassenbibliotheek (objectlibrary) is duidelijk uitgebreider dan die in de fase definitiestudie is vastgelegd. Op dat moment had ik nog niet voldoende inzicht om alle objecten vast te leggen. Dit besef kwam pas tijdens de interviews en test van deze pilot. Tijdens de fase definitiestudie had ik nog geen rekening gehouden met leermiddelen en had ik het performance object nog niet vastgelegd. Ter verduidelijking volgt hieronder een omschrijving van alle objecten:

User object:

Het user object is de gebruiker van het systeem. Dit kan zowel een student als een docent zijn.

Usergroup object:

Het usergroup object bevat een group (student) gebruikers van het systeem. Deze group kan dan worden toegevoegd aan een course.

Course object:

Een course is een verzameling activiteiten die een user kan volgen (leertraject). Indien activiteiten met succes zijn afgerond is de course "passed"

Activity object:

Een activity is een opdracht voor een user. Activities worden ondergebracht in een of meerdere courses.

Activitytype object:

Een activitytype is het type van de activiteit. Bijvoorbeeld een leesopdracht of een toets.

Resource object:

Een resource is een leermiddel wat kan worden gebruikt ter ondersteuning van een activity.

Performance object:

In dit object worden de prestaties bijgehouden. Ieder performanceobject heeft een unieke userid en een unieke activityid.

Voor een uitgebreidere beschrijving van de objecten wil ik u verwijzen naar de pilotontwikkelplannen in de bijlagen.

Natuurlijk zijn deze objecten in zekere mate afhankelijk van elkaar:

- Een user kan gekoppeld worden aan 0 of meerdere usergroups (en andersom)
- Een user kan gekoppeld zijn aan 0 of meerdere courses (en andersom)
- Een user kan gekoppeld zijn aan 0 of meerdere activiteiten (en andersom)
- Een usergroup kan gekoppeld zijn aan 0 of meerdere courses (en andersom)
- Een activity heeft 1 of meer resources
- Een activity heeft 1 activitytype
- Een activitytype heeft 0 of meer activiteiten
- Een combinatie van een user en activity heeft 1 unieke performance

Na het ontwerpen van de objectlibrary heb ik een start gemaakt met het bouwen van de database. Dit is een kwestie van de objecten redeneren naar een database. De manier

hoe je hier relaties vastlegt is zeer belangrijk voor het te bouwen systeem. Dit omdat het tijdens de bouwfase lastig is om nog aanpassingen te maken aan de dataaag. Hieronder ziet u het relationeel model van de database:

```
Sysuser (id, username, lastname, middleinitial, firstname, adress,
houzenumber, zipcode, city, country, homephone, workphone, mobilephone,
email, fax, password, usertype)

Usergroup (id, name, description)

Sysuser_usergroup (sysuserid, groupid)
Foreign key sysuserid references "id" in sysuser. Nulls not allowed
Foreign key groupid references "id" in usergroup. Nulls not allowed

Course (id, name, description, selfregister, manager)
Foreign key manager references "id" in sysuser. Nulls allowed

Course_sysuser (sysuserid, courseid)
Foreign key sysuserid references "id" in sysuser. Nulls not allowed
Foreign key courseid references "id" in sysuser. Nulls not allowed

Course_usergroup (courseid, usergroupid )
Foreign key courseid references "id" in sysuser. Nulls not allowed
Foreign key usergroupid references "id" in sysuser. Nulls not allowed

Activity (id, name, description, maxtries, nummaxtries, setbydocent,
performancetype, activitytype)
Foreign key activitytype references "id" in activitytype. Nulls allowed

Performance (userid, activityid, performancedecimal, performarmancechar,
performanceboolean, description, passeddate, passedchar)
Foreign key userid references "id" in sysuser. Nulls not allowed
Foreign key activityid references "id" in activity. Nulls not allowed

Activity_course (activityid, courseid)
Foreign key activityid references "id" in activity. Nulls not allowed
Foreign key courseid references "id" in course. Nulls not allowed

Prerequisites (first, next)
Foreign key first references "id" in activity. Nulls not allowed
Foreign key next references "id" in activity. Nulls not allowed

Activitytype (id, name, description, icon)

Resource (id, name, description)

Activity_resource (activityid, resourceid)
Foreign key activityid references "id" in activity. Nulls not allowed
Foreign key resource references "id" in resource. Nulls not allowed
```

Figuur 4.4 relationeel databasemodel Keylearning

Om te onderzoeken of wat ik heb ontworpen ook juist is heb ik model door enkele personen laten beoordelen (begeleider, collega's en afstudeergenoot). Ik kwam hierbij tot de conclusie dat hetgeen ik ontworpen heb inderdaad te rechtvaardigen is. Ik kreeg van deze personen wel een aantal vragen over het objectmodel welke ook bij de lezer kunnen rijzen:

Waarom maar één unieke performance voor een combinatie activityid/userid?

Ik heb ervoor gekozen om slechts één performance in het systeem bij te houden. Een alternatief zou zijn om ook "oude" performances van een gebruiker in het systeem bij te houden. Om voor de gebruiker het systeem zo simpel (laagdrempelig) te houden heb ik niet voor deze optie gekozen.

Waarom heeft een activiteit voor een user de status "passed" en niet een course niet?

Ik heb er bewust voor gekozen om een course geen status "afgerond" mee te geven. Als ik dit wel had gedaan dan had er in de tussentabel van "user" en "course" een veld passed gestaan. Als bijvoorbeeld user X in usergroep Y zit en de usergroep Y volgt course A dan bestaat de relatie "user" "course" niet eens, ondanks dat de gebruiker natuurlijk wel ingeschreven is. Ik kan om die reden de status passed niet in de database bijhouden. Omdat ik toch wil bijhouden of een gebruiker een course heeft afgerond heb ik een functie geschreven die voor user X checkt of hij alle activiteiten van course A heeft afgerond. Indien dit geval is wordt deze status door de businesslaag doorgegeven aan de presentatielaag.

Waarom zijn docent en student geen aparte objecten?

Ik heb de keuze gemaakt maar één object te gebruiken. Docent en student zijn beide entiteiten met dezelfde eigenschappen. Door middel van usertype worden beide typen onderscheiden. Dit heeft als gevolg dat een docent ook ingepland kan worden voor een course, bijvoorbeeld door een andere docent. Dit kan zelfs gezien worden als een voordeel.

Waarom kan een activiteit meerdere leermiddelen hebben, is dit nodig?

Het kan zijn dat om een bepaalde activiteit te completeren je uit drie verschillende boeken passages moet lezen. Dit wordt dan voor de overzichtelijkheid als aparte leermiddelen gespecificeerd. Ook heeft het als voordeel dat al deze leermiddelen in meerdere activiteiten kunnen worden gebruikt.

Een user is gekoppeld aan courses en aan activiteiten, een usergroep heeft alleen courses, hoe kan dat?

Dit komt omdat in de relatie tussen activiteit en user (de performance tabel) het resultaat wordt bijgehouden. Een usergroep heeft natuurlijk ook activiteiten maar deze hoeven niet in de database te worden bijgehouden. Deze worden met een SQL query opgevraagd. Een usergroep heeft namelijk courses en deze courses hebben activiteiten. Dit volstaat.

Wat zijn Prequisites?

Prequisites zijn startvoorwaarden van activiteiten. Bijvoorbeeld als je de toets van een bepaalde course wil doen dan zal je eerst alle andere activiteiten moeten doen. Pas dan wordt de activiteit voor een gebruiker vrijgegeven.

Nadat ik de database heb vastgelegd ben ik mij bezig gaan houden met de vraag wat wil een student allemaal met de applicatie doen. In de workshop werd nog eens bevestigd dat dit de volgende drie zaken te zijn:

- Activiteit uitvoeren
- Resultaat bekijken
- Rapport genereren

Ik ben deze zaken gedetailleerder gaan uitwerken door middel van een use case. Dit om de functionaliteit, schermontwerp en het navigatieschema te bepalen. Een use case is een scenario van een activiteit in het systeem. Ik had deze extra stap nodig omdat mijn taakdiagrammen uit de fase definitiestudie (te) globaal zijn. Ik heb meerdere personen van Metrix een test laten doen door hun gedachtegang over hoe deze drie zaken uitgevoerd moeten worden op te schrijven. Uiteindelijk bleek dat alle personen ongeveer dezelfde gedachtegang hadden. Als voorbeeld volgt hieronder een use case:

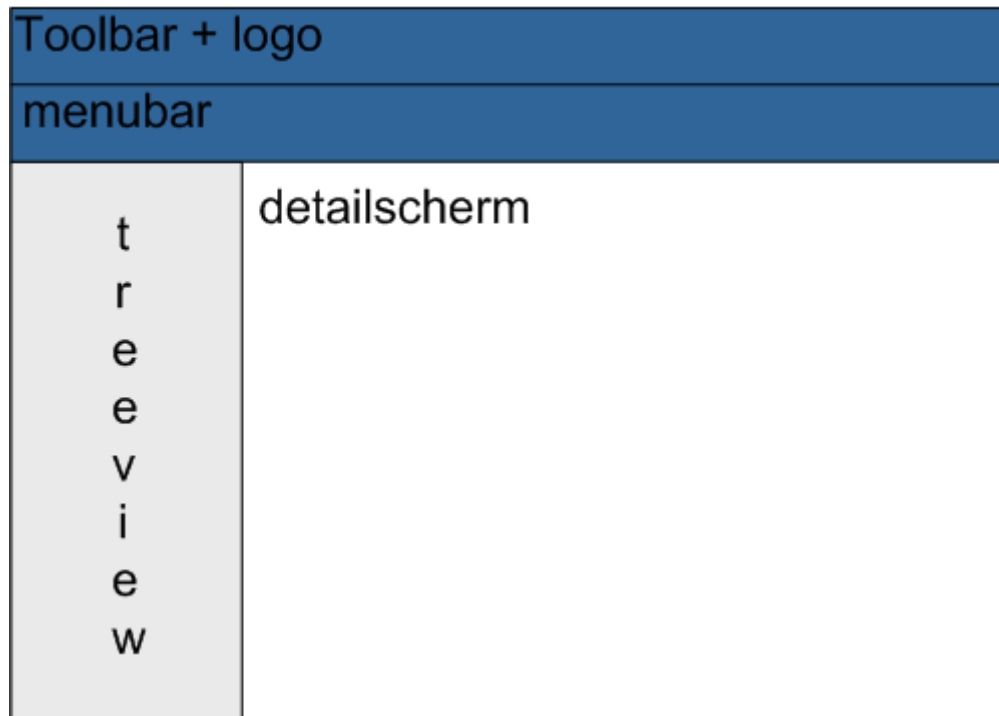
Use case Activiteit uitvoeren

Naam	Activiteit uitvoeren
Samenvatting	De gebruiker kan een activiteit uit een gerelateerde course uitvoeren
Actoren	Student
Aanname	De activiteit is daadwerkelijk aan de student toegewezen
Beschrijving	Na inloggen opent de gebruiker een course, na het openen van deze course klikt de gebruiker een activiteit aan, na het uitvoeren van deze activiteit, mailt de gebruiker zijn uitwerking naar de docent. Indien de activiteit dit toelaat mag de gebruiker ook zelf het resultaat terugkoppelen, bijvoorbeeld voor een leesopdracht.
Uitzonderingen	Geen
Resultaat	Gebruiker heeft activiteit uitgevoerd en resultaat is eventueel vastgelegd

Figuur 4.5 user case activiteit uitvoeren

Alle gebruikers hebben de behoefte om vanuit een course een activiteit te benaderen. Om deze reden heb ik ervoor gekozen dat gebruikers via een course naar een activiteit kunnen navigeren.

Aan de hand van deze gegevens heb ik het navigatieschema en het voorlopige schermontwerp opgebouwd. Dit zijn de laatste activiteiten van het pilotontwikkelplan. Omdat uit mijn onderzoek bleek dat gebruikers het prettig vinden om via een course te navigeren heb ik gekozen voor een treeview dat resulteerde in het volgende globale schermontwerp:



Figuur 4.5 globaal schermontwerp pilot student

Aan de linkerkant is gekozen voor een treeview met de openstaande en reeds gevolgde courses. Zo weet de gebruiker altijd precies waar hij zit. De overige functies voor een student zoals het wijzigen van een password en het genereren van het rapport zijn te benaderen vanuit het menu. Om het systeem zo overzichtelijk mogelijk te houden heb ik gekozen om alles te tonen in het hoofdscherm en wordt er nimmer gebruik gemaakt van pop-ups. Voor het kleurontwerp en lettertypen had ik mij te houden aan de style van de Metrix producten. Aan de ene kant is dit natuurlijk wel jammer. Maar gelukkig zijn er nog heel veel andere zaken vorm te geven. Nu het definitieve ontwerp vastligt kan ik aan de slag met de realisatie van de pilot welke in de volgende paragraaf wordt beschreven.

4.5 Realisatie

4.5.1 Datalaag

Zoals eerder in dit verslag al aan de orde kwam leek het mij het handigst om één data laag voor de beide pilots te ontwikkelen. Hoewel het vastleggen van de objecten en de bijbehorende eigenschappen niet erg complex is, is het wel een tijdrovende klus. Voor ieder object moeten de eigenschappen worden beschreven. Ook moet de koppeling van de objecten worden vastgelegd. Zo moet van een course de activiteiten geladen kunnen worden. Voor het vastleggen van gerelateerde objecten had ik twee mogelijkheden:

1. relaties tussen objecten vaststellen tijdens het laden van het object

voorbeeld:

Course heeft behalve heeft als eigenschap een collectie "activities" . Deze collectie bevat activity objecten die tot de course behoren. Als de course wordt geladen dan worden ook deze activiteiten geladen

2. relaties tussen objecten vaststellen op een moment naar keuze door het aanroepen van een functie

voorbeeld:

Course heeft als eigenschap een collectie "activities" . Deze collectie bevat activity objecten die tot de course behoren. Als de course geladen wordt blijft de eigenschap activities leeg. Pas als de functie getRelatedObject wordt aangeroepen wordt de eigenschap gevuld.

In eerste instantie heb ik gekozen voor optie één, het leek mij wel handig om gelijk de gerelateerde objecten in een collectie hebben. Naarmate ik meer objecten kreeg kwam ik hiermee echter in de knoei. Allereerst is zo iets niet goed voor de performance van het systeem omdat er een sneeuwbaaleffect ontstaat. Als bijvoorbeeld een instantie van het object course wordt geladen, worden automatisch de activiteiten geladen, vervolgens wordt van iedere activiteit alle resources geladen. Verder is het zo dat je in een "deadlock" terecht kan komen. Als ik van een user alle courses laad, vervolgens laad ik van al deze courses weer alle users enzovoort. Dit kan eeuwig duren. Toen ik tot deze conclusie ben gekomen ben ik overgestapt op optie twee. Verder heb ik in de data laag nog een aantal voor de hand liggende methoden om de integriteit van de database in de gaten te houden opgeslagen.

4.5.2 Businesslaag

In de businesslaag heb ik gekozen voor zoveel mogelijk herbruikbare methoden. Het maakt je code weliswaar wat complexer om te maken, maar uiteindelijk levert het wel minder code op die bovendien goed gestructureerd is. Om de objecten herbruikbaar te maken is het belangrijk deze te nummeren, dus:

User = 1
Usergroup = 2
Course = 3
Activity = 4
Performance = 5
Resource = 6
Activitytype = 7

Om de manier waarop herbruikbaarheid wordt toegepast te tonen volgen hieronder voorbeelden van twee herbruikbare functies:

GetObjectById(objectId As Variant, objtype As Long) As Object

Deze functie geeft een object terug door middel van het geven van een ID van een object en het bijbehorende type. Als je aan deze functie de waarden 1 en 3 aanbiedt zal deze een course object teruggeven die correspondeert met een courseId die 1 bevat. Dit werkt natuurlijk een stuk handiger in plaats van voor ieder object een aparte module te maken (getUserById, GetActivityById)

GetRelatedObject(parentIndex As Long, childIndex As Long, parentId As Long, retrieveType As Boolean, Optional orderBy As String) As Collection

Deze functies geeft een collectie objecten terug die gerelateerd zijn (of juist niet gerelateerd zijn) aan een bepaald object, deze collectie kan eventueel gesorteerd worden. Een tweetal voorbeelden om dit te verduidelijken:

```
GetRelatedObject(4,6,2,true,"resource_name")
```

Deze functie geeft een collectie resource objecten terug van de activity waarvan de ID 2 is, deze collectie wordt gesorteerd op de naam van de resource.

```
GetRelatedObject(1,3,1,false)
```

Deze functie geeft een collectie course objecten terug die **niet** gerelateerd zijn aan een user waarvan de id 1 is.

De functie GetRelatedObject is een functie die in de pilot zeer vaak wordt gebruikt, en deze is natuurlijk een stuk handiger ten opzichte van voor ieder object een aparte functie te schrijven (getUsersByCourse, get CoursesByUser, getActivitiesByUser enzovoort). Na vastleggen van de businesslaag is het zaak om de presentatielaag te ontwerpen. Natuurlijk is het zo, dat deze modules al ontworpen zijn tijdens de pilot student, maar ook uitstekend bruikbaar zijn voor de pilot docent.

4.5.3 Presentatielaag

Voor de presentatielaag ben ik begonnen met het maken van de treeview. Op internet zijn genoeg voorbeelden te vinden van standaard treeviews. Oorspronkelijk heb ik zo een standaard treeview ingebouwd in de pilot. Ik kwam erachter dat deze standaardoplossing niet de juiste was. De treeview gaf af en toe vreemde foutmeldingen. Ondanks mijn redelijke kennis van programmeren bevat de treeview lastige javascriptcode waar ik niet helemaal uitkwam. De treeview bevatte een enorme lap javascriptcode waar ik de zin en de tijd niet voor had om uit te zoeken waar de fout zat. Hierop heb ik zelf mijn treeview gemaakt met zo weinig mogelijk code. Doordat de applicatie alleen geschikt hoeft te zijn voor Internet Explorer is een treeview met DHTML gemakkelijk te realiseren. Bijna alle navigatie vindt ook plaats vanuit de treeview. Klik je op een course dan zie je in het detailscherm alles over de course, klik je op een activiteit dan zie je in het detailscherm alles over de activiteit. Andere functies zoals het wijzigen van het password en het generen een rapport zijn te benaderen vanuit het menu. Dit menu is gemaakt in de style van Metrix en was reeds binnen Metrix voorhanden. Ik heb het niet zelf ontworpen.

Iconen heb ik zelf uitgekozen. Alvorens ik deze heb ingevoerd, heb ik aan medewerkers binnen Metrix gevraagd waarmee ze de iconen associeerde. Op deze manier heb ik onderzocht of mijn iconen duidelijk genoeg zijn. Met betrekking tot kleurgebruik en lettertype ben ik gebonden aan de eisen van Metrix. Ik heb mij behalve de plaatsing weinig kunnen uitleven. Aan de ene kant in dit natuurlijk wel jammer maar aan de andere kant is het natuurlijk logisch daar het product door Metrix zal worden gebruikt en op de markt gebracht gaat worden. Bovendien ben ik wel tevreden met het kleurgebruik van Metrix en heb dus ook geen reden gehad om een voorstel te doen voor een wijziging.

4.6 Test

4.6.1 Opzet

Zoals eerder gespecificeerd mag de pilot maximaal één maal een iteratie ondergaan. Dit moet dan blijken uit een gebruikerstest. De pilot is bij Metrix intern getest. In eerste instantie had ik besloten om de pilots te laten testen door mijn begeleider. Na een gesprek met mijn begeleider ben ik tot de conclusie gekomen dat het beter is om "objectieve" personen te kiezen voor een test en niet iemand die direct bij de opdracht betrokken is.

Een test waarbij gebruik wordt gemaakt van toekomstige gebruikers die niet betrokken zijn bij het ontwerpen en bouwen van een systeem geeft informatie die meer waarheidsgetrouw is. Dit omdat de opdrachtgever een expert is. Een expert kan onbewust dezelfde expertise verwachten van 'gewone' gebruikers.

Hieruit vloeide het idee voort om de pilot door medewerkers van Metrix te laten testen. Hiervoor is de pilot op het intranet geplaatst. Nu is het ook belangrijk om een betrouwbare test te maken. Om een test betrouwbaar te maken is het goed om een grote doelgroep te benaderen. Alle medewerkers van Metrix zijn enthousiast en willen graag medewerking verlenen. Hierdoor had ik een testgroep van negen personen tot mijn beschikking. Vooral om de intuïtieviteit van de applicatie te bepalen is het handig om de gebruikers op te delen in bepaalde groepen. Bij Metrix is er één persoon van het niveau beginner, twee van het niveau medium en zes hebben het niveau expert. De test bestaat enerzijds uit een aantal opdrachten en anderzijds uit een aantal gesloten vragen. Het testformulier is te vinden in de bijlagen.

In de test wordt de gebruikers niet altijd rechtstreeks gevraagd wat zij van de applicatie vinden. De kans is namelijk groot dat dan een niet waarheidsgetrouw antwoord zal worden gegeven. In plaats daarvan wordt de gebruiker op een indirecte manier gevraagd wat hij of zij van de applicatie vindt. De testpersoon wordt gevraagd om tijdens de opdrachten hardop te denken en deze gedachtegang op te schrijven. Dit staat ook wel bekend als de "think aloud" methode. Hierdoor wordt een goed inzicht verkregen in hoe een gebruiker denkt over de navigatie in het systeem en over de gebruiksvriendelijkheid.

Bij sommige vragen is het moeilijker om een open vraag te stellen en zit er niks anders op om dit in gesloten vorm te doen. Bijvoorbeeld in hoeverre zij het systeem als laagdrempelig beschouwen en in hoeverre zij collega's aansporen tot het gebruik van de applicatie. Zodoende heb ik uit de test conclusies kunnen trekken tot wat er aangepast moet worden.

Behalve het testen van de gebruikerskant dient natuurlijk ook de data en businesslaag getest te worden. Dit is iets wat de ontwikkelaar zelf kan. Je geeft dan aan iedere functie een aantal waarden mee en je kijkt of hetgeen wat de functie teruggeeft hetgeen is wat je verwacht. Als dit niet het geval is pas je de functie aan en ga je verder met de volgende procedure. Dit is een manier van testen waarmee je tijdens het ontwikkelen al mee bezig bent. Programmeren is eigenlijk constant testen van hetgeen wat je gebouwd hebt. Ik heb hiervoor een test applicatie gemaakt die vanuit visual basic de objectlibrary aanspreekt. Ik heb hiervoor gekozen omdat je anders dit vanuit ASP moet doen en dit programmeert veel lastiger omdat je na iedere wijziging de DLL opnieuw moet compileren. De resultaten van het testen van de data- en businesslaag zijn echter niet zinvol om hier uit te werken.

4.6.2 Resultaat

Uit de gebruikerstest bleek dat gebruikers nadat er een actie heeft plaatsgevonden een melding missen. Als een activiteit is afgerond dan wil de gebruiker een melding krijgen dat de activiteit werkelijk is afgerond. Hoewel iconen dit ook aangeven wil een gebruiker meer bevestiging krijgen. Omdat gebruikers moesten opschrijven wat zij dachten kwam ik erachter dat sommige testpersonen veel moeite hadden met de manier hoe de treeview werkt. Zoals bekend staat een item van een treeview uit een plaatje met daarachter tekst, bijvoorbeeld:

⊕ informaticastudenten

Om het item “uit te klappen” ben ik persoonlijk geneigd om op het item zelf te klikken en niet op het plaatje. Uit de test bleek dat sommige gebruikers liever op het ⊕ symbool klikken. Verder kwam naar voren dat gebruikers na herladen van het scherm graag de treeview willen zien in de staat waarin deze was voor het herladen. Als een student een activiteit als voltooid wil zetten moet de treeview herladen worden en verschijnt er na de activiteit een vinkje. De treeview is na herladen helemaal dichtgeklapt. Dit wordt door de gebruiker als zeer onprettig ervaren.

Uit de test blijkt dat de gebruikers de pilot als intuïtief en laagdrempelig ervaren. Hier ben ik zeer content mee, omdat dit mijn gedachtegang over hoe een intuïtieve en laagdrempelige interface gemaakt dient te worden bevestigde. Ook reageren gebruikers positief over het gebruik van de applicatie en het aansporen van collega's hiervan. Natuurlijk moet dit laatste in de praktijk blijken maar het in eerste instantie natuurlijk positief.

4.7 Aanpassingen

Na inventarisatie bleek dat mij drie dingen te doen staan:

1. Toevoegen dat een gebruiker ook op het plaatje naast de treeview kan klikken.
2. Items in de treeview de status (open of dicht) aan de url meegeven.
3. Een pop-up met een bevestiging als een student zijn gegevens heeft gewijzigd of wanneer een activiteit is voltooid.

Hoewel ik eerder stelde dat ik geen gebruik van pop-ups ging maken, bleek uit de test dat gebruikers een “melding” graag in een pop-up willen zien. Omdat de mening van testgebruikers minstens zo zwaar is als van de ontwikkelaar heb ik ervoor gekozen om deze wijziging door te voeren.

Alleen aanpassing twee kost wat extra programmeerwerk. Aanpassing één en drie heb ik in een half uur kunnen realiseren. Aanpassing twee heeft me een halve dag gekost.

5 Pilot docent

Na de pilot student volgt de ontwikkeling van de pilot docent. Een aantal zaken van deze pilot zijn reeds uitgevoerd. De database is al vervaardigd en een deel van de objectlibrary is reeds al gemaakt. Ook is er zoals eerder geïllustreerd een grote overlap met de data laag van de pilot student. Bepaalde zaken van dit hoofdstuk zullen minder uitgebreid aan de orde komen omdat voor deze aspecten hetzelfde geldt als in de pilot student.

In de eerste paragraaf komt de *definitie* aan de orde, in de volgende paragraaf *planning* zal de planning nader onder de loep worden genomen. De paragraaf *workshop* beschrijft hoe de workshop verlopen is. In de paragraaf *pilotontwikkelplan* zal ik de totstandkoming van dit document verklaren. De paragraaf *realisatie* beschrijft het bouwgedeelte van de pilot en in de paragraaf *testen* zal de test van deze pilot worden toegelicht. Tot slot komen in de paragraaf *aanpassingen* de aanpassingen aan bod die ik na de test heb uitgevoerd.

5.1 Definitie

De definitie van de pilot docent is de volgende:

“binnen de pilot docent is het mogelijk om gebruikers, leertrajecten, activiteiten, leermiddelen en het forum te beheren. Verder dient de docent te kunnen communiceren door middel van een prikbord of forum. Docenten kunnen via het LMS beoordelingen geven aan activiteiten van studenten. Per activiteit, leertraject en gebruiker dienen rapporten gecreëerd te kunnen worden. Dit is bedoeld voor managementrapportage maar ook om voortgang te bewaken en om problemen te signaleren.”

5.2 Planning

De pilot docent neemt zeven weken in beslag. Aangezien de pilot vrij uitgebreid is, is drie weken voor de bouwfase vrij kort. Doordat de pilot student een week is uitgelopen moet ook deze tijd nog ingelopen gaan worden. Ter illustratie vindt u hieronder de planning van de pilot docent.

Activiteit per dag	1	2	3	4	5	6	7
Stel plan van aanpak op							
Bereid workshop voor							
Voer workshop uit							
Evalueer workshop							
Stel pilotontwikkelplan op							
Bouw pilot							
Test pilot							
Evalueer test							
Pas pilot aan							

Figuur 5.1 planning pilot docent

5.3 Workshop

Na het maken van het plan van aanpak is het tijd voor de workshop. Zoals bekend is ook deze workshop omgedoopt tot interview. Het onderwerp van het interview is vooral op welke wijze er genavigeerd moet worden in het systeem. De specificaties hoeven niet verder uitgediept te worden omdat deze in de fase definitiestudie al zijn vastgelegd. Dhr. Maas is zeer tevreden over hoe de navigatie van de pilot student eruit ziet en wil deze voortzetten. Er komt naar voren dat het systeem vooral snel moet reageren. Als een course aan een aantal activiteiten gekoppeld wordt, dan moet dit met een paar muisklikken geregeld kunnen. Verder heeft het de voorkeur om niet constant, als er een actie uitgevoerd wordt, het hele scherm te herladen. Dit is een moeilijke opgave om met HTML te realiseren. De opdrachtgever wil het liefst dat de gebruiker niet door heeft dat het systeem herladen wordt. Als een activiteit wordt toegevoegd aan een course dan mag dit direct naar de database worden doorgevoerd maar de treeview moet dan in dezelfde staat blijven. De focus dient ook op hetzelfde object terugkomen, zodat de gebruiker niet door heeft dat de wijzigingen realtime worden doorgevoerd naar de database. Omdat de opdrachtgever vooral functionaliteit wil tonen staat dit voorop in de te bouwen pilot. In overleg met dhr. Maas is er een vernieuwde prioritering afgesproken, deze wijkt in delen af van die in de definitiestudie en is als volgt:

1. hoofdschermen maken voor user, usergroup, course, activity, en resource
2. navigatieobject maken (treeview)
3. intelligente koppeling maken (voor entiteiten aan elkaar te koppelen)
4. performance kunnen toevoegen, verwijderen en wijzigen
5. wijzigschermen maken voor user, usergroup, course, activity, en resource
6. inlogscherms maken
7. rapporten creëren

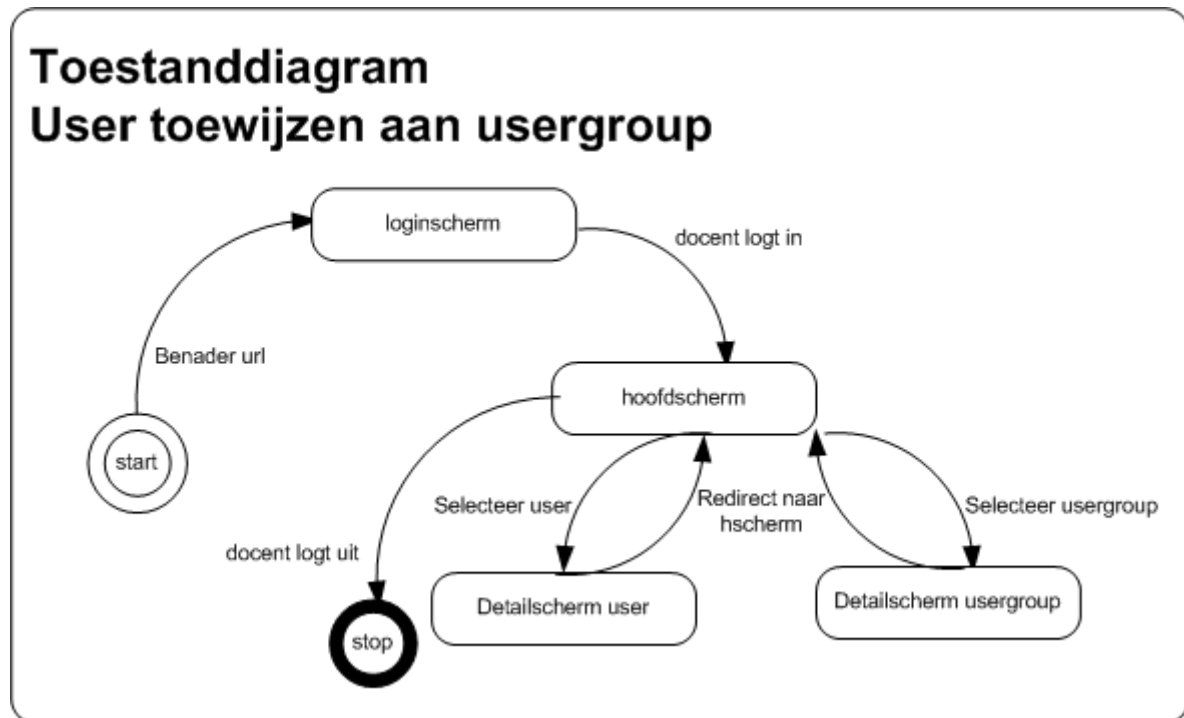
De reden hiervan is dat de opdrachtgever op korte termijn een demonstratie wil geven bij een bedrijf wat al producten van Metrix afneemt. De bovengenoemde zaken worden hiervoor belangrijk geacht. De reden dat rapporten in deze prioritering onderaan staan is het feit dat de verwachting is dat het maken van een rapportageobject gezien de complexiteit veel tijd in beslag neemt en afhankelijk is van de kwaliteit en opbouw van de rest van het systeem.

5.4 Pilotontwikkelplan

Nu er genoeg input is gekomen uit de workshop kan ik aan de slag met het pilotontwikkelplan. Ik heb als voordeel dat ik in de pilot student al een aantal zaken heb gemaakt die ik nu kan overslaan of hergebruiken. De database en het objectmodel zijn al klaar. Het toeval wil dat mij dit precies een week tijdwinst oplevert. Dit was nou precies de week die ik eerder "kwijt" was geraakt. De eerste activiteit die ik uitvoer is het vastleggen van de taken van de gebruiker van de pilot docent. Uit de workshop kwam naar voren dat een docent hoofdzakelijk deze activiteiten uitvoert:

- Courses, activiteiten en user aanmaken
- User toewijzen/verwijderen van usergroup
- Activiteit toewijzen/verwijderen van course
- Users of usergroups toewijzen/verwijderen van course
- Activiteit van een student beoordelen
- Rapport genereren

Net als in de pilot student beschreven is, heb ik met hulp van de medewerkers use cases gemaakt. Wat mij opviel is dat medewerkers entiteiten vanuit verschillende invalshoeken benaderen. Een activiteit toevoegen aan een course, wordt door de ene persoon benadert via course en door de andere vanuit activiteit. Dit geldt overigens voor bijna alle use cases en is dus een belangrijk punt om rekening mee te houden. Na deze use cases ben ik toestandsdiagrammen te maken. Ter illustratie vindt u hieronder een voorbeeld van een toestandsdiagram:

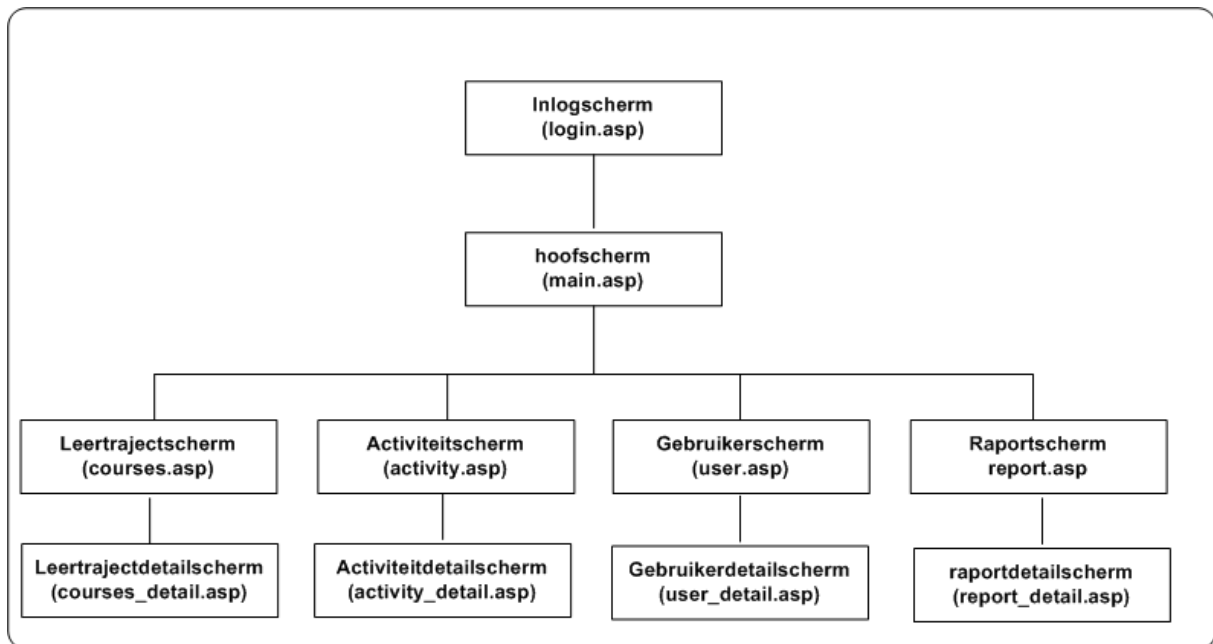


Figuur 5.2 toestandsdiagram user toewijzen aan usergroup

Na het vastleggen van de “check” ben ik mij gaan bezig houden met de manier hoe er genavigeerd moet gaan worden in het systeem. Op de term “check” zal ik later in dit verslag terugkomen.

Ik heb ervoor gekozen om aparte detailschermen te ontwerpen, de reden hiervan is dat je anders te veel informatie te zien krijgt. Een alternatief is het laten scrollen van een gebruiker. Het blijkt dat gebruikers hier niet de voorkeur aan geven. Het plaatsen van meer content in het hoofdscherm door kleinere ruimten tussen de items, ofwel kleinere lettergrootten, zorgt echter voor een onrustige opbouw van het scherm. Zoals de naam van het scherm al aangeeft gaat het slechts om details die niet altijd nodig zijn. Het hoofdscherm van een item bestaat uit de belangrijkste eigenschappen van het item plus de “intelligente koppeling” welke de entiteiten vanuit één scherm aan elkaar kan koppelen. Zoals eerder genoemd wil de opdrachtgever de manier van navigeren vanuit de pilot student voortzetten. Dit is in principe mogelijk met een paar aanpassingen. Meer over het schermontwerp en over de intelligente koppeling vindt u onder het kopje van de presentatielaag.

Voor de volledigheid vindt u hieronder het navigatieschema van de pilot docent:



Figuur 5.3 navigatieschema pilot docent

5.5 Realisatie

5.5.1 Dataaag

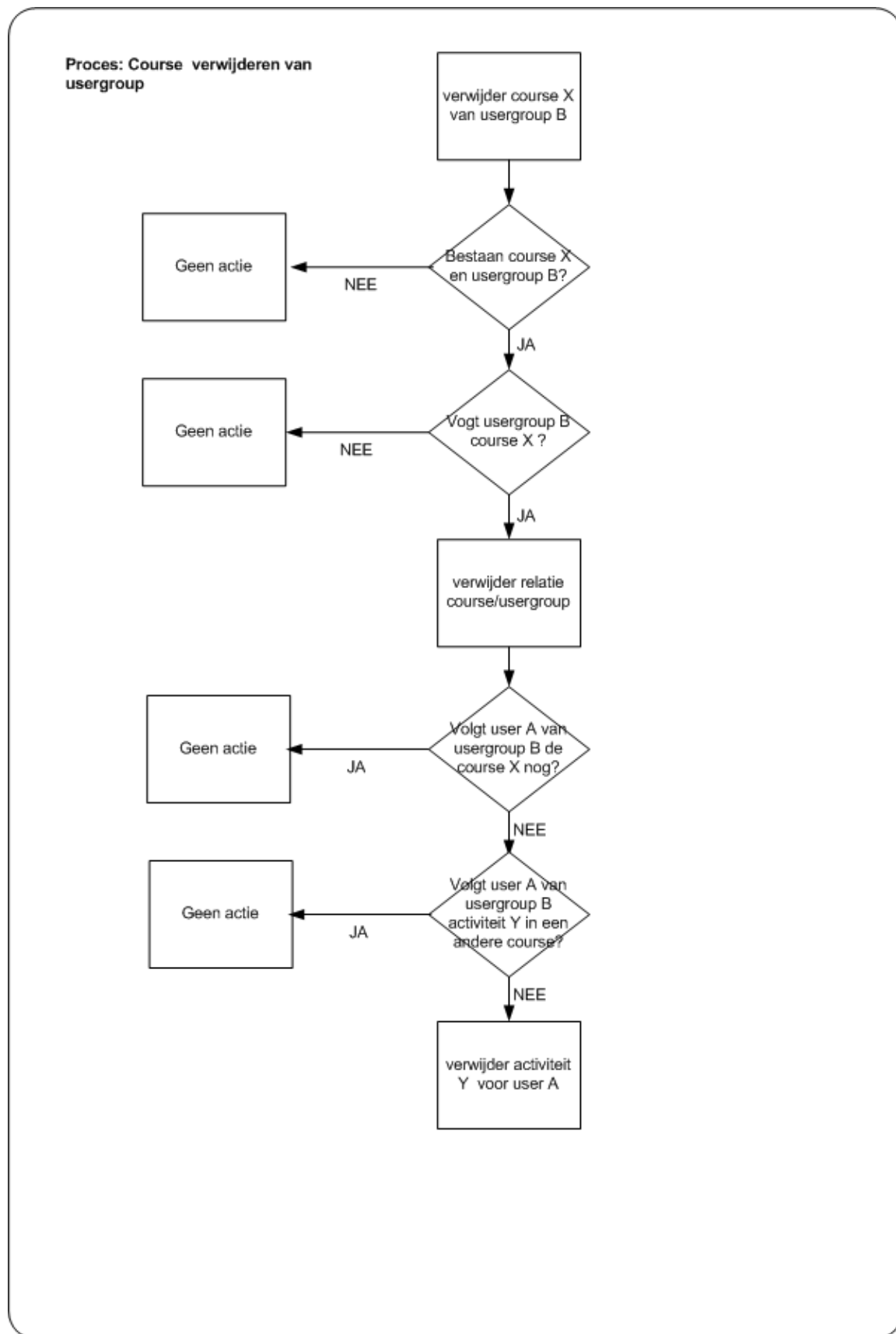
Na het maken van het pilotontwikkelplan is het tijd voor de bouwfase. Zoals eerder aangeven moet er in het systeem entiteiten aan elkaar gekoppeld worden. Alvorens zo een koppeling mag plaatsvinden dienen er een aantal controles uitgevoerd te worden. Er zijn een aantal kritische punten wanneer zo een “check” moet plaatsvinden, te weten:

- User toevoegen aan een usergroup
- User verwijderen van een usergroup
- Course toevoegen aan usergroup
- Course verwijderen van usergroup
- Course toevoegen aan user
- Course verwijderen van user
- Activiteit toevoegen aan user
- Activiteit verwijderen van user

Omdat het resultaat van een gebruiker vastligt in de relatie tussen user en activity (de tabel performance) dient er in deze tabel records te worden toegevoegd en verwijderd. Het niet uitvoeren van zo een check lijdt onherroepelijk tot inconsistentie in de database.

De relatie tussen user en activity mag/moet alleen bestaan als een gebruiker in een course zit waarin de activiteit is toegewezen. De acht bovenstaande zaken hebben hier allemaal invloed op. Als je bijvoorbeeld een course van een usergroup verwijderd moet je controleren of de user de course of activiteiten van de course nog elders op de user van toepassing zijn. Deze checks zijn gemakkelijk op papier te zetten maar vrij complex om te programmeren.

Om dit te verduidelijken volgt hieronder het schema van zo een check:



Figuur 5.4 check course verwijderen van usergroup

5.5.2 Businesslaag

Zoals eerder beschreven in het gedeelte pilot student maak ik ook in deze pilot veel gebruik van de eerder genoemde herbruikbare functies. Natuurlijk dienen er ook vele nieuwe functies te worden toegevoegd. Deze functies had ik reeds tijdens het opstellen van het pilotontwikkelplan vastgelegd. Tijdens deze fase kwam ik er achter dat een aantal functies ontbraken. Voor het opvragen van objecten heb ik de eerder beschreven functie `getRelatedObject` tot mijn beschikking. Met deze functie is het echter alleen mogelijk om objecten vanuit een ander object op te vragen. Bijvoorbeeld vanuit het object `user` de gerelateerde objecten opvragen. Ik had dus nog een functie nodig om vanuit de presentatielaag alle objecten van een bepaalde entiteit op te vragen. Om herbruikbaarheid van de functies in de gaten te houden heb ik gekozen voor de functie `getObjects(index)`. Deze functie geeft een collectie objecten terug naar aanleiding van het indexnummer. Deze belangrijke functie heb ik uiteraard ook in het pilotontwikkelplan aangepast. Belangrijk in de businesslaag is de beveiliging. Het is belangrijk de objectlibrary goed te beveiligen. Ik heb ervoor gekozen om deze controles op de businesslaag en niet in de presentatielaag uit te voeren.

Dit om te voorkomen dat een slimme student de businesslaag en haar functies via de docentmodule kan benaderen. Iedere gebruiker die inlogt heeft een unieke sessie. Aan de hand van deze sessieID kan de gebruiker worden herleidt. Deze gebruiker heeft weer rechten die in de objectlibrary worden vastgelegd. Hiermee kan weer worden bepaald of een gebruiker de functie wel of niet mag uitvoeren. Als ik ervoor had gekozen om deze controle op de presentatielaag uit te voeren zou de applicatie niet goed beveiligd zijn omdat er genoeg programma's te downloaden zijn om direct een objectlibrary aan te spreken.

5.5.3 Presentatielaag

Na het bouwen van de businesslaag ben ik aangekomen bij het ontwerp van de presentatielaag. Zoals eerder genoemd wordt de style van de pilot student voortgezet. Het scherm bestaat dan ook uit een menubalk, een treeview en een contentscherm waarin alle "actie" plaatsvindt. De treeview uit de pilot student is bruikbaar, maar moet (weer) aangepast worden. Deze dient namelijk een extra niveau te hebben, gebruikers in het systeem kunnen in dat geval worden onderverdeeld in gebruikersgroepen.

Een van de belangrijkste zaken waarvoor denkwerk moet worden verricht is ontwerp van de intelligente koppeling. De intelligente koppeling is het gedeelte van de applicatie waar de entiteiten van het systeem op een slimme en overzichtelijke manier aan elkaar moeten worden gekoppeld. Deze intelligente koppeling komt voor in het userscherm (koppeling met `usergroup`, `course`), het `usergroupscherm` (koppeling met `user`, `course`), het `coursescherm` (koppeling met `activity`, `user`, `usergroup`), het `activitiescherm` (koppeling met `course`, `prerequisites`, `resources`) het `resourcescherm` (koppeling met `activity`).

Voor de opbouw van deze koppeling zijn er een aantal mogelijkheden:

1. De koppeling in het detailscherm van de entiteit door middel van een listbox
2. De koppeling door middel van het lijsten in een apart window
3. De koppeling door middel van dubbelkliklijsten in het detailscherm van de activiteit.

Bij de eerste optie heb je als nadeel dat je het overzicht mist van de huidige situatie. Als bijvoorbeeld student "Jan Jansen" moet worden toegevoegd aan de gebruikersgroep "informaticastudenten" wordt er genavigeerd naar de student Jan Jansen, vervolgens wordt in dit scherm via een dropdownlistbox gekozen voor de juiste usergroep en wordt er op de button "voeg toe" gedrukt. Dit is zondermeer mogelijk maar in dit geval mis je het overzicht. Je kunt niet in één oogopslag zien of deze user al gekoppeld aan de usergroep en aan welke usergroups deze user nog meer gekoppeld is.

Bij de tweede optie staan alle gerelateerde en niet gerelateerde activiteiten in een apart window. In het bovengenoemde voorbeeld moet vanuit het hoofdscherm van de entiteit op een button worden gedrukt. In dit aparte window staan alle koppelingen als lijsten gespecificeerd, door middel van dubbelklikken kunnen entiteiten indien gewenst worden toegevoegd en verwijderd. Dit heeft als voordeel dat je de informatie op een hele overzichtelijke manier aan de gebruiker presenteert. Een nadeel is dat je die overzichtelijkheid een beetje te niet doet doordat je deze informatie in een pop-up opent.

De derde optie was voor mij de juiste. Ik heb besloten om deze koppeling in het hoofdscherm te plaatsen. Dit heeft als voordeel dat de gebruiker de zaken die hij het meest doet vanuit het hoofdscherm van de entiteit kan doen. Eventuele details staan in een apart scherm welke te benaderen vanuit het hoofdscherm. Ter illustratie ziet u het contentscherm van een activiteit met de bijbehorende intelligente koppeling:

The screenshot displays the 'HTML I deel 1' activity interface. At the top, there's a header bar with the title 'HTML I deel 1' and a 'type:' dropdown set to 'toets'. Below this, a form contains fields for 'Naam:' (filled with 'HTML I deel 1') and 'omschrijving:' (filled with 'Een introductie in html'). A 'type:' dropdown is also present. The main content area is divided into several sections: 'Resultaat' with radio buttons for 'decimaal' and 'tekstueel' (selected), and checkboxes for 'aantal pogingen:' (selected 'ja') and 'afronding:' (selected 'door student'). Below this, there are four panels: 'Leertrajecten:' containing 'HTML voor dummies'; 'Overige leertrajecten:' containing 'HTML voor gevorderden javascript voor gev.'; 'Startvoorwaarden' (empty); and 'Overige activiteiten' containing a list of related activities like 'HTML I deel 2', 'HTML I deel 3', 'toets HTML voor dummie', 'HTML deel 4', 'HTML deel 5', 'HTML deel 6', and 'toets HTML voor gevorde'. At the bottom, there are two more panels: 'Leermiddelen:' containing 'wat is HTML?', 'wat is HTML? (2)', and 'wat is HTML (3)?'; and 'Overige leermiddelen:' containing a list of topics like 'wat zijn hyperlinks?', 'document over frames', 'wat zijn frames?', 'wat zijn tabellen?', 'wat zijn formulieren?', 'webdesign in de praktijk', 'webdesign in de praktijk eindtoets HTML', and 'eindtoets HTML'. Blue circular arrows indicate the intelligent linking between these different sections.

Figuur 5.5 activiteitscherm met intelligente koppeling

Omdat ik de details in een apart scherm plaats heb ik als voordeel dat alle belangrijke zaken van deze activiteit in één scherm te zien zijn. Je kunt precies zien welke startvoorwaarden deze activiteit (niet) heeft, in welke courses deze activiteit zit en welke leermiddelen deze activiteit heeft. Door middel van het dubbelklikken op een item zal deze worden verwijderd of worden toegevoegd.

Zoals eerder uit de use cases naar voren kwam zal de ene gebruiker vanuit het activiteitscherm een resource aan een activiteit willen koppelen en de andere wil dit doen vanuit het resourcescherm. Door deze intelligente koppeling is dit beide mogelijk. Dit geldt natuurlijk ook voor de benadering van de andere activiteiten.

Als er een koppeling moet worden gewijzigd zal deze wijziging uiteraard naar de database moeten worden doorgevoerd. Ik had hier twee mogelijkheden tot mijn beschikking:

1. alle wijzigingen van een lijst doorvoeren als de lijst focus verliest;
2. iedere wijziging direct doorvoeren.

Bij optie één komt het op de gebruiker over alsof de items al zijn gewijzigd als hij erop klikt. Dit is slechts schijn want alle wijzigingen worden door middel van javascript bijgehouden en pas daadwerkelijk doorgevoerd als het object de focus verliest. Er zijn echter nadelen aan optie één. Als de gebruiker in de treeview een ander item aanklikt wordt het javascript event "onfocusout" afgevuurd, in dit geval moet de gebruiker even wachten voordat alle wijzigingen zijn doorgevoerd, dit kan door een gebruiker als zeer vervelend worden ervaren, verder is het zo dat als het systeem van de gebruiker zou vastlopen de wijzigingen nooit worden doorgevoerd en dat al het werk van de gebruiker voor niets is geweest. Ook wanneer de gebruiker de pagina in het geheel wegklikt wordt het "onfocusout" in het geheel niet uitgevoerd. Op basis van deze informatie heb ik besloten om optie twee te gebruiken. In dit geval moeten wijzigingen dus direct worden doorgevoerd. Dit direct doorvoeren moet aan een aantal eisen voldoen omdat het anders niet zinvol is, te weten:

- Na een wijziging dient niet het hele scherm ververs te worden
- Het direct doorvoeren moet over een modale internetverbinding mogelijk zijn en een acceptabele performance hebben

Omdat ik hiermee rekening moest houden heb ik gekozen voor I-frames. I-frames worden ondersteund door microsoft internet explorer en is eigenlijk een HTML pagina in een HTML pagina. Het geheel wordt geïllustreerd door de onderstaande afbeelding.



Figuur 5.6 I-frames

Als een item verandert door middel van een dubbelklik dan wordt de database actie uitgevoerd en het I-frame herladen. Als een course wordt toegevoegd worden er in chronologische volgorde worden de volgende acties uitgevoerd:

1. Gebruiker dubbelklikt op leertraject.
2. Iframe wordt omgeleid naar een ASP pagina waar de update wordt aangeboden aan de objectlibrary.
3. Businesslaag voert functie `updateUserUsergroup` uit en spreekt functie `CheckupdateUserUsergroup` van de data laag aan.
4. Data laag voert check uit en voegt indien nodig een "performance" toe in de database aan een gebruiker die nu gerelateerd is aan deze activiteit.
5. Database geeft resultaat van update terug aan data laag.
6. Data laag geeft resultaat terug aan businesslaag.
7. Businesslaag geeft resultaat terug aan presentatielaag.
8. I-frame wordt teruggeleid naar de ASP pagina waar de gekoppelde course te zien is, in dit geval met de doorgevoerde wijziging.
9. Gebruiker ziet wijziging in I-frame van content scherm.

Dit zijn heel wat acties maar uit een test bleek dat over een modale internetverbinding dit geheel maximaal $\frac{3}{4}$ seconde(!) in beslag neemt. Er mag geconcludeerd worden dat dit acceptabel is.

Ik ben erg lang bezig geweest met het werkend maken van het uploaden van bestanden. Het ASP script wat bij Metrix al werkend was, kreeg ik voor Keylearning niet aan de praat. Ik kreeg ook geen foutmelding en ik had geen idee waaraan het lag. Na veel speuren en navraag bij Metrix kwam ik erachter dat mijn Norton virusscanner dit tegenhield. Als je Norton antivirus niet opstart werkte het wel.

5.6 testen

5.6.1 Opzet

Na het afronden van de bouwfase is het tijd om de docentmodule te testen. Omdat ik uit de manier van testen bij de pilot student veel input heb gekregen zie ik geen reden om van deze strategie af te wijken. De test van deze pilot is iets uitgebreider omdat de pilot qua functionaliteit ook uitgebreider is. Nadat ik de pilot on-line heb gezet heb ik de deelnemers uitgenodigd. Het testen van de pilots gaf in de praktijk twee nadelen waarover ik tijdens het ontwerp niet had nagedacht. Als de deelnemers bij een bepaalde opdracht vastliepen konden zij niet verder met de volgende opdracht. Dit omdat de opdrachten aan elkaar gekoppeld zijn. Tijdens het testen heb ik met deze deelnemers contact gehad. Verder is het zo dat er uit één database wordt gewerkt, als deelnemer X iets wijzigt dan is dit ook van toepassing op deelnemer Y. Ik heb hiervoor een schema gemaakt wanneer welke deelnemer met de test aan de slag kon. Dit had wel als gevolg dat ik een paar dagen vertraging op heb gelopen bij het testen van deze pilots. In deze dagen heb ik een flink gedeelte van het document geschreven wat nu voor u ligt.

5.6.2 Resultaat

Uit de test bleek dat de meeste gebruikers zeer tevreden zijn over de manier waarop de pilot docent werkt. Het toevoegen van een resultaat van een student bleek voor sommige deelnemers niet te werken en verder bleek dat de treeview wat kleine kinderziektes vertoonde. Wat mij zeer tevreden stelt is het feit dat de gebruikers de applicatie hoog laten scoren op het gebied van usability aspecten en laagdrempeligheid. Verder willen de meeste deelnemers collega's aansporen tot gebruik. Dit zie ik als een bevestiging van de

laagdrempeligheid en hieruit kun je de conclusie trekken dat de applicatie prettig is om mee te werken.

5.7 Aanpassingen

De veranderingen die ik aan de hand deze pilot moest doen waren slechts een aantal kleine aanpassingen in javascript. Deze aanpassingen kon ik snel realiseren. Natuurlijk zijn er nog een heleboel zaken welke na deze pilot gedaan dienen te worden alvorens het systeem ingevoerd kan worden. Dit vindt plaats in de volgende fase, "test, herstel en invoering"

6 Herstel en oplevering

Na de fase pilotontwikkeling breekt er een nieuwe fase aan. Het eerste wat u hier aan op kan vallen is de naam. In de planning werd deze fase namelijk de fase “test, herstel en invoering” genoemd. Later vond ik “herstel en oplevering” beter passen. Alvorens ik beschrijf hoe het proces is verlopen zal ik aangeven waarom de naam van de fase aangepast is.

Allereerst is het zo dat de pilots niet gekoppeld hoeven te worden zoals ik eerder in de planning had aangegeven. Ik had hier in planning tijd voor vrijgemaakt toen ik nog in de veronderstelling was dat ik de pilots helemaal los van elkaar kon ontwikkelen. Zoals eerder genoemd maak ik gebruik van één objectlibrary. De koppeling van de data en businesslaag van de pilots hoeft er dus niet te komen, deze is reeds aanwezig. De test die in deze fase plaats zal moeten vinden is overbodig. Voor elke pilot weet ik doormiddel van de test in de fase pilotontwikkeling precies wat er nog moest gebeuren. Als hier nu nog twijfel over bestaat dan zou dat betekenen dat mijn test niet goed is geweest. Dit is niet het geval. In plaats van een test zal er direct begonnen worden met het maken van aanpassingen. Hierna wordt in samenwerking met de opdrachtgever geïnventariseerd wat er nog is blijven liggen. Vervolgens zal een “to do list” worden opgesteld om vast te leggen wat er gedaan moet worden om de applicatie compleet te maken. Dit kan zowel door mij persoonlijk worden gedaan als door iemand anders, maar in ieder geval buiten het bestek van deze opdracht. Zoals in de opdrachtsomschrijving genoemd zal de applicatie niet direct in het bedrijfsleven worden ingevoerd. Ik vind hierom het woord invoering niet goed gekozen, in plaats hiervan gaat het om oplevering van het systeem. Met herstel wordt niet alleen een mogelijkheid tot aanpassing bedoeld maar ook het maken van kleine uitbreidingen

6.1 Aanpak

Ik ben begonnen met inventariseren van wat er nog gedaan moet worden om het systeem geheel functioneel te maken. Ik heb hiervoor met de opdrachtgever de systeemeisen nog eens doorgelopen en deze vergeleken met de applicatie die tot nu toe was gebouwd. Aan de hand hiervan heb ik een voorlopige “to do list” opgesteld. Aan de lijst is ook een prioritering meegegeven. De zaken die het belangrijkste zijn staan bovenaan. De lijst moest vervolgens van boven naar onder worden afgewerkt.

6.2 Uitvoering

Voor het uitvoeren van deze werkzaamheden heb ik ongeveer 1,5 week tot mijn beschikking. In deze paragraaf zal ik uitleggen wat ik in deze 1,5 week heb gedaan en hoe dit tot stand is gekomen.

Probleem 1, Rapporten te globaal:

Een aantal HTML rapporten voor docenten zijn te globaal, zo is het mogelijk om rapporten te maken per gebruiker en per activity maar nog niet per usergroup. Bij de performance per gebruiker of per activity wordt er nog geen gemiddeld cijfer uitgerekend. Verder is het zo dat het rapport nog enigszins uitgelijnd moet worden omdat deze nog niet optimaal uit de printer komt.

Ik ben begonnen met mogelijkheid tot berekening van gemiddelden, dit was slechts een kleine aanpassing aan de businesslaag. Vervolgens heb ik het genereren van een rapport met betrekking tot een usergroup toegevoegd. Nu was het ook nog zaak om het geheel “printable” te maken. Hiervoor heb ik de items

wat dichter op elkaar moeten plaatsen en heb ik de lettergrootte verkleind. Al met al heb ik één dag gedaan over het aanpassen van de standaard rapport functionaliteit.

Probleem 2, menu nog niet af:

Het menu is nog niet helemaal af, niet alle items zijn toegevoegd. Bovendien is het handiger als de menu items dynamisch zijn en uit de database komen. Dit menu is reeds werkend bij Metrix maar moet nog wel worden ingevoerd in de database om deze werkend te maken.

Ik ben begonnen met het toevoegen van een aantal tabellen in de database om het menu dynamisch te maken. Hierna heb ik alle missende menuitems toegevoegd en heb ik het menu uitgebreid getest. Het aanpassen van het menu heeft mij één dag gekost.

Probleem 3, password management voor student:

Een student heeft nog niet de mogelijkheid om een password te wijzigen. Het geheel is wel te vinden in de presentatielaag maar wijzigingen worden nog niet doorgevoerd. Ook moet, indien de docent dit wil, de student worden gedwongen tot het wijzigen van een password door middel van het geven van een geldigheidsduur aan het password.

Ik ben begonnen met het doorvoeren van het wijzigen van een password in de database. Hiervoor heb ik wat extra checks moeten toevoegen aan de businesslaag. Verder kan een docent een geldigheidsduur meegeven aan een password. Als een student met een verlopen password probeert in te loggen, zal dit geweigerd worden tot hij zijn password heeft gewijzigd. Deze aanpassing had ik in 1,5 werkdag gerealiseerd.

Probleem 4, aanpassing toevoegen startvoorwaarden:

Door een fout in mijn code bleek het mogelijk te zijn om aan een activiteit dezelfde activiteit als startvoorwaarde mee te geven. Een gebruiker kan dan bijvoorbeeld activiteit 10 niet afronden omdat een docent (al dan niet per ongeluk) activiteit 10 als startvoorwaarde heeft meegegeven. De gebruiker zit hierdoor vast in zijn leertraject.

Ik heb dit aangepast. Dit betrof een aanpassing in de businesslaag die ik in een paar uur heb kunnen bewerkstelligen.

Probleem 5, problemen datum en tijd formaat:

Ik heb problemen ondervonden met het opslaan en uitlezen van datum en tijd velden van de database. Hierdoor was het mij nog niet gelukt om de datum en tijd mee te geven aan een activiteit die een gebruiker van het systeem voltooid.

Een kleine aanpassing in de data laag bleek de oplossing te zijn. Omdat het toevoegen van de juiste datum en tijd op mijn computer wel werkte maar op de centrale server niet heb ik hier wel wat onderzoek voor moeten doen. Deze aanpassing heeft mij twee werkdagen gekocht.

Probleem 6, koppeling prikboard aan Keylearning:

Het prikboard was tot nu toe nog niet volledig functioneel. Een standaard ASP forum vinden is niet al te moeilijk maar deze code moet wel gekoppeld worden aan Keylearning. Een standaard forum is natuurlijk niet uitgewerkt in drie lagen. Ik heb geprobeerd het standaard forum over te zetten naar een drie lagen model. Dit bleek echter lastiger dan ik dacht, na anderhalve dag kwam ik erachter dat dit geen doen is. Dit probleem zal verschoven moeten worden naar de definitieve "to do list". Het trachten op te lossen van dit probleem heeft mij ongeveer 1,5 dag gekost.

6.3 Afronding

De eerder genoemde voorlopige "to do list" kan nu worden omgezet in een definitieve "to do list". Het belangrijkste winstpunt is wel dat de applicatie ook zonder deze punten in de "to do list" kan werken. De applicatie is dus in te voeren in het bedrijfsleven maar nog niet volledig functioneel zoals in de eisen is gespecificeerd. Voor de volledigheid volgt hieronder een opsomming van de "to do list":

Koppeling standaard prikboard aan Keylearning	Zoals eerder te lezen was is de koppeling van een standaard prikboard aan Keylearning nog niet gelukt. Er zijn nu twee mogelijkheden. Alsnog proberen een standaard prikboard aan de drie lagen filosofie te hangen of een geheel nieuw prikboard ontwerpen in de stijl van Keylearning.
Aanpassen stijl standaard prikboard aan Keylearning stijl	Indien er alsnog besloten wordt een standaard prikboard te "verbouwen" dan is het nodig om deze aan de Keylearning stijl aan te passen
Genereren uitgebreide rapporten met extern object	Het moet met de Keylearning applicatie mogelijk zijn om rapporten te genereren. Hierbij kan men bijvoorbeeld denken aan een extern webobject als Seagate Crystal Reports. Zo'n object dient door de klant aan te worden geschaft. Natuurlijk moet dit wel door Metrix worden getest. Alvorens je dit kan doen moet eerst crystal reports worden aangeschaft.
Ontwikkelen systeemspecificaties	Om een (potentiële) klant goed inzicht te geven in wat Keylearning doet is het nodig om systeem specificaties te ontwikkelen
Ontwerpen van een logo	Een logo is er nog niet. Bij een applicatie hoort een logo dus deze zal op korte termijn opgeleverd moeten worden.
Labels toevoegen aan database	De database zal moeten worden uitgebreid om de applicatie ook in het Engels uit te kunnen brengen.

Het opstellen van deze lijst was mijn laatste werk bij Metrix. Hoe en wanneer de opdracht afgerond wordt is het moment dat ik dit verslag schrijf nog niet bekend.

Deel III Evaluatie

7 Evaluatie proces

In hoofdstuk word er teruggekeken op het proces. Alle fasen in het project worden geevalueerd, te weten; *opstartfase*, *plan van aanpak definitiestudie*, *pilotontwikkeling* en tot slot *herstel en oplevering*.

7.1 Evaluatie opstartfase

De opdracht zoals ik deze in samenwerking met de opdrachtgever heb opgesteld sta ik nog volledig achter. Zoals uit dit verslag blijkt was de opdracht haalbaar. Ik vond het wel vervelend dat ik toen ik al begonnen was de opdrachtschrijving nog aangepast moest worden omdat deze natuurlijk ook precies aan de eisen die de Haagse Hogeschool stelt aan het afstuderen moet voldoen. Dit betrof vooral aanpassingen om de omvang en de diepgang van de opdracht zo goed mogelijk vast te leggen.

Over mijn oriënterende gesprekken met de opdrachtgever ben ik zeer tevreden. Vanaf het eerste moment had ik het gevoel dat het “klikte”. Wat ik hieraan erg fijn vond is dat de opdrachtgever nooit uit het oog verloor dat er ook vanuit school eisen worden gesteld aan de opdracht. Ook ben ik erg tevreden over de vrijheid die ik heb gekregen met betrekking tot het kiezen van ontwikkelmethode en techniek. Een nieuwe kijk op IAD is ook voor Metrix zeer nuttig gebleken. Verder was ik erg content met het feit dat deze opdracht veel vaardigheden van de afstudeerder vraagt (databases, programmeren en vormgeving).

7.2 Evaluatie fase plan van aanpak

De fase “plan van aanpak” heb ik ervaren als een onzekere fase. Achteraf blijkt wel dat enig denkwerk over de planning zeker loont. Op de week achterstand die ik had bij aanvang van de pilot student na heb ik de planning zo goed als helemaal nagekomen. Ik zal volgende keer wel een aparte fase maken voor het ontwikkelen van de objectlibrary en de database. Dit omdat ik dit nu onder de noemer “pilot student” heb gemaakt en dit natuurlijk niet helemaal de bedoeling.

Ik ben tevreden over het feit dat ik IAD als ontwikkelmethode had gekozen. Er kan natuurlijk gesteld worden dat het kiezen voor IAD een schijnkeuze is omdat ik tot nu alleen van IAD gebruik heb gemaakt. Dit is echter niet het geval. Ondanks dat ik in de opdrachtsomschrijving de voorlopige keuze voor IAD heb gemaakt heb ik in de fase plan van aanpak de methode IAD naast RAD gelegd en de voors en tegens gespecificeerd. Pas hierna ben ik tot de conclusie gekomen dat IAD de beste methode was. Wat ik minder goed bestudeerd heb, is welke keuze het beste is voor het bedrijf. Dit is namelijk een andere invalshoek. Ik heb bij Metrix moeite gehad om genoeg mensen op de been te krijgen voor testen en workshops. Ik zal in het vervolg ook beter in de gaten houden of een bedrijf ook geschikt is voor een methode zoals IAD.

Binnen IAD heb ik gekeken naar een ontwikkelstrategie. Ik ben tevreden over het feit dat ik heb gekozen voor “incrementele ontwikkeling”. De voor- en nadelen zoals IAD deze onderkent bleken inderdaad te kloppen. Voor een éénmans project voor de periode van nog geen vijf maanden volstaat deze methode. Voor grotere projecten kan er beter een andere methode worden gekozen omdat dan ook indien nodig passages van de definitiestudie opnieuw opgeleverd kunnen worden.

7.3 Evaluatie fase definitiestudie

Ten opzichte van al mijn andere projecten die ik heb uitgevoerd, heb ik besloten om dit maal minder tijd te besteden aan de definitiestudie. Ik moet eerlijk bekennen dat ik hierin een beetje te ver doorgeschoten ben. Ik heb de definitiestudie weliswaar in twaalf werkdagen afgekregen maar de laatste vijf dagen waren wel werkdagen van elf uur. Ik zal in volgende projecten hierin een middenweg moeten zoeken.

Het bestuderen van systeemspecificaties van de verschillende aanbieders heeft mij een goede kijk gegeven op het gebied van E-learning. Ik heb de positieve punten uit deze applicatie kunnen benutten en van de negatieve punten heb ik een hoop geleerd.

Het klinkt misschien wat vreemd maar ik ben achteraf blij dat mijn applicatie niet gestandaardiseerd is. De AICC standaard is ouderwets en niet geschikt voor Web applicaties en SCORM richt zich puur op uitwisselbaarheid met andere E-learning systemen, wellicht dat hier in de toekomst nog naar gekeken kan worden maar voorlopig wordt dit nog niet nodig geacht. In de praktijk wordt er ook door organisaties zelf maar weinig leermaterialen ontwikkelt die voldoen aan deze standaarden.

Waar ik nogal van baalde is het feit dat er geen mensen beschikbaar waren voor het uitvoeren van een workshop, omdat ik dit tijdig in de gaten had, heb ik van deze workshops gelukkig geen werk gemaakt. Ik zal ook nooit weten hoe deze dan verlopen waren en of dit waardevoller zou zijn dan een interview. Omdat ik heel veel ideeën had over het geven van een workshop, zoals dit ook op de opleiding is geleerd, vond ik het extra jammer. Aan de andere kant ben ik over de input die uit de interviews voortkwam zeer tevreden. De doelen en beperkingen van het systeem werden mij snel duidelijk.

Voor het typeren van de huidige situatie heb ik veel gehad aan mijn begeleider. Hij wist veel van E-learning en was daarom snel geneigd wat aan hem te vragen. Ook heb ik bij mij bijbaan in de avonduren (T-mobile klantenservice) een onderzoek gedaan naar E-learning. Ik heb één van de vier doelgroepen dus persoonlijk kunnen onderzoeken. Als ik meer tijd had gehad, had ik ook bij de andere drie doelgroepen een onderzoek kunnen doen. Niet dat deze informatie niet betrouwbaar is. Maar eigen onderzoek geeft je nu eenmaal een betere kijk op het geheel.

Over het opstellen van de systeemeisen ben ik tevreden. IAD eist dat systeemeisen geprioriteerd moeten worden en dat heeft mij wel geholpen. Ik kreeg hierdoor een zeer goed inzicht over in welke volgorde bepaalde activiteiten uitgevoerd moesten worden. Wel vind ik achteraf dat bepaalde eisen beter geformuleerd hadden moeten worden. Een voorbeeld: "De functionaliteit binnen de data- en businesslaag moet volledig herbruikbaar zijn". Ik zal in het vervolg woorden zoals: altijd, volledig en nooit vermijden omdat je met dit soort woorden jezelf in de nesten kan werken. De zaken die nog zijn blijven liggen zijn veelal luxe eisen.

Het kiezen voor twee in plaats van drie actoren heeft goed uitgepakt. Ik zal nooit weten hoe het was vergaan met "beheerder" als extra actor. Ik kan wel zeggen dat docenten de denkwijze van de applicatie goed doorhebben opdat een actor als beheerder niet nodig is. Dit is een punt wat de laagdrempeligheid van de applicatie bevestigt.

Over de taakdiagrammen ben ik niet geheel tevreden, deze waren te globaal uitgewerkt. De objecten heb ik er weliswaar uit kunnen herleiden maar ze waren te globaal om de navigatie van je systeem eruit te ontleiden. Met behulp van use cases in de fase pilotontwikkeling heb ik dit probleem kunnen oplossen.

Het drie lagen model is een goede keuze geweest. Door deze manier van programmeren is de code gemakkelijk opnieuw te gebruiken en uit te breiden. Als er in de toekomst een

client/server applicatie van Keylearning nodig is dan is deze gemakkelijk te maken, hetzelfde geldt voor gebruik van een ander database management systeem, dit omdat alle integriteitschecks zich in de data laag bevinden.

Het pilotplan had naar mijn mening beter gekund. Hoewel de opdrachtgever over het pilotplan tevreden was vond ik dat ik heb te weinig nagedacht over prioritering van de pilotdelen. Natuurlijk had ik de verschillende systeemeisen geprioriteerd. Maar dit zijn kleinere stukken dan pilotdelen. Een pilotdeel bevat over het algemeen een aantal systeemeisen. Hierdoor is de ontwerpfase wel eens wat chaotisch verlopen. Het is niet zo dat alles in het honderd liep maar het had zeker beter gekund. In het vervolg zal ik mij in de fase definitiestudie meer gaan verdiepen in de pilotdelen en de prioritering ervan.

7.4 Evaluatie fase pilotontwikkeling

Over de planning van de pilot student ben ik ontevreden. Ik had mij niet gerealiseerd dat ik voor het ontwikkelen van de objectlibrary meer tijd nodig had. Dit omdat ik deze direct voor beide pilots moest maken zoals eerder toegelicht. Door deze achterstand heb ik in deze pilot veel te maken gehad met tijdsdruk. Onder tijdsdruk blij ik moeilijk te presteren en kan ik minder geconcentreerd werken. Ik heb in deze periode veel lange dagen gemaakt. Eindconclusie is dat ik een week uitgelopen ben wat in het bedrijfsleven grote gevolgen kan hebben.

Over de planning van de pilot docent ben ik wel tevreden. Omdat ik in de pilot student al wat werk had verricht heb ik een week kunnen inlopen wat mij weer op schema bracht.

Over de workshops heb ik gemengde gevoelens, aan de ene kant vind ik het heel jammer dat ik mijn opgedane kennis over workshops niet in de praktijk heb kunnen brengen omdat de workshop niet door ging. Ik ben wel zeer content met het interview wat is gehouden. Dit omdat hier alsnog zeer veel informatie uit heb kunnen halen.

Over het pilotontwikkelplan ben ik tevreden. Uit het interview heb ik veel informatie gehaald om dit pilotontwikkelplan te maken. Over het samenstellen van de objecten ben ik zeer tevreden. Door het voorleggen van de objecten aan meerdere mensen heb ik de conclusie kunnen trekken dat mijn denkwijze hierover goed is geweest.

Over de realisatie van de pilots ben ik zeer tevreden. Ik heb in een korte tijd veel kunnen programmeren. Het meest tevreden ben ik over de grote mate van herbruikbaarheid van de pilots en de omvang van de hetgeen wat ik heb gebouwd.

Over de test ben ik zeer tevreden. Ik heb gebruik gemaakt van een combinatie van de wetenschappelijke hardop denk methode en een aantal directe vragen aan de gebruikers. Ik zal in het vervolg wel meer rekening houden met het feit dat de testgebruikers gebruik maken van één database. Zoals eerder beschreven liep dit nog wel eens mis.

7.5 Evaluatie fase herstel en oplevering

De fase herstel en oplevering is een fase geweest waarin er niet meer zo methodisch is gewerkt. Hoewel de opdrachtgever zeer tevreden is over deze fase ben ik minder tevreden. Dit omdat ik niet methodisch heb gewerkt. IAD onderkent namelijk geen fase om extra aanpassingen te doen. Dat is wel jammer want tijd voor een volledige iteratie was er niet meer. Maar er was nog wel tijd om wat kleine dingen uit te voeren. Hoewel je deze fase niet "methodisch" kunt noemen ben ik niet als een kip zonder kop allerlei dingen gaan programmeren. Ik heb mij gehouden aan de voorlopige "to do list" welke ik aan het begin van deze fase had opgesteld. Alleen over de activiteit "koppeling prikbord aan Keylearning" ben ik niet tevreden omdat deze activiteit onbegonnen werk was. Omdat dit wel een systeemeis was heb ik nog tevergeefs geprobeerd dit af te ronden maar helaas zonder succes. Over de overige activiteiten ben ik wel tevreden. Het moeilijke aan deze fase is dat je keuzes moet maken. Aan de ene kant moet er aan het verslag worden gewerkt en aan de andere kant moet de applicatie ook zo ver als mogelijk afgerond worden. Ik heb geprobeerd beide te combineren en vind dat ik daar redelijk is geslaagd ben.

8 Evaluatie product

In dit hoofdstuk zal ik kort ingaan op de kwaliteit van de verschillende producten, allereerst zal het plan van aanpak onder de loep worden genomen, hierna de definitiestudie, vervolgens de pilotontwikkelplannen student en docent en tot slot de Keylearning applicatie.

8.1 Evaluatie documenten

Over het plan van aanpak over het algemeen tevreden, de planning heeft mij voldoende houvast gegeven gedurende het project. Ondanks de korte tijd die ik had uitgetrokken voor het maken van de definitiestudie ben ik over het product "definitiestudie" tevreden. Belangrijk aan een definitiestudie is dat beide partijen op één lijn zitten over de manier waarop het systeem gebouwd gaat worden en wanneer deze gebouwd gaat worden. Dit was het geval en daarom ben ik er tevreden over. Als je specifiek naar de inhoud kijkt ben ik minder tevreden over de manier waarop de taakdiagrammen zijn uitgewerkt, omdat deze te globaal waren heb ik hier weinig aan gehad, dit zal ik in het vervolg uitgebreider doen. Over de beide pilotontwikkelplannen ben ik zeer tevreden. De producten pilotontwikkelplan "pilot student" en "pilot docent" zijn zeer compleet, ik ben tijdens de bouwfase dan ook vaak op deze producten teruggevallen.

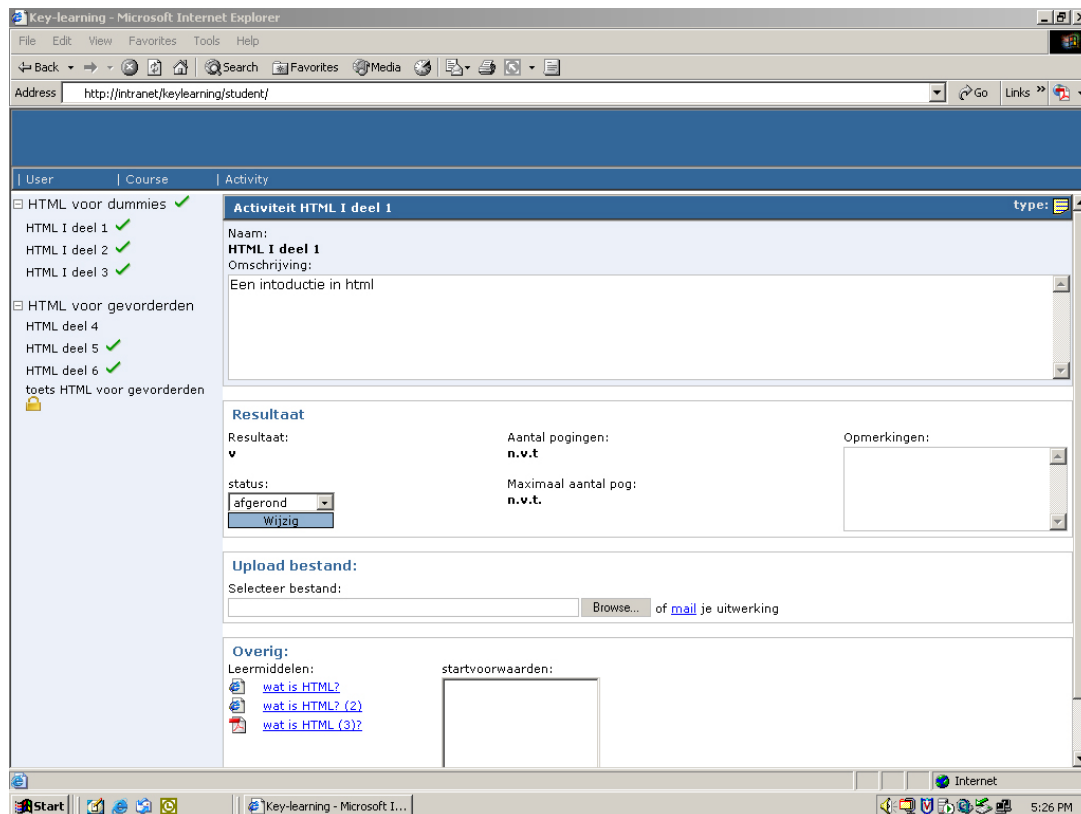
8.2 Evaluatie applicatie Keylearning

Tot slot zal ik in deze paragraaf het eindresultaat onder de loep nemen.

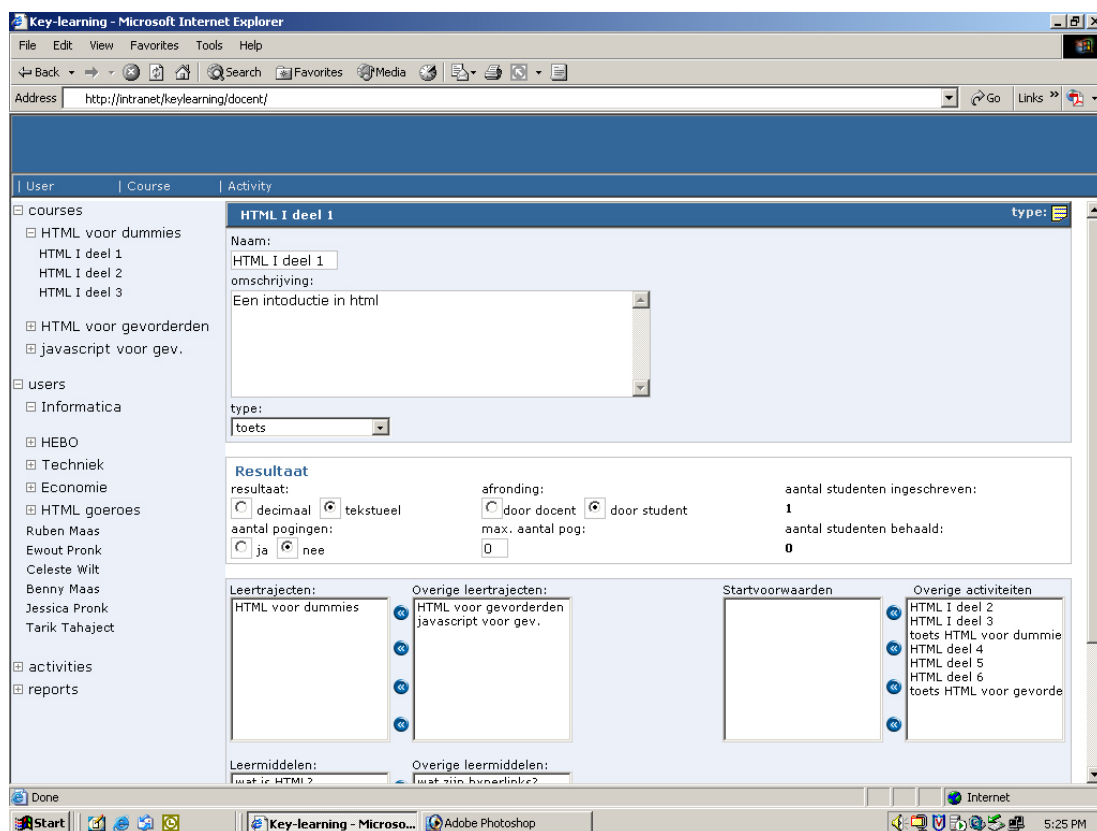
Over het algemeen genomen ben ik over het eindresultaat tevreden. Zoals eerder uit de test bleek is de applicatie door de gebruikers als zeer intuïtief en gebruiksvriendelijk ervaren. Ook ben ik zelf tevreden over de functionaliteit die ik heb opgeleverd in de tijd die er voor de periode gepland stond.

Over de vormgeving ben ik tevreden. Aangezien vormgeving nooit mijn sterkste kant is geweest, ben ik tevreden over kleurgebruik en schermopbouw. Vanwege tijdgesprek laat de vormgeving van de pilot docent nog enigszins te wensen over. Dit betreft vooral kleine zaken die nog uitgelijnd moeten worden.

Om u een indruk te geven van het te ontwikkelen systeem volgt op de volgende bladzijde een tweetal "screendumps" van de Keylearning applicatie.



Figuur 8.1 studentmodule



Figuur 8.2 docentmodule

In figuur 8.1 ziet u de studentmodule, links bevindt zich een treeview met de courses waaraan de student gerelateerd is en de bijbehorende activiteiten. De student kan via het menu andere functies benaderen zoals het wijzigen van gegevens en het uitdraaien van een rapport.

In figuur 8.2 ziet u de docentmodule, hierin bevindt zich aan de linkerkant een treeview waarin de docent tussen gebruikers activiteiten en courses kan navigeren. Per entiteit is het mogelijk om via de intelligente koppeling deze aan een andere entiteit te koppelen. Een docent kan in het performance scherm, welke via activiteit of via user te benaderen is, het resultaat voor een student invoeren. Ook dan de docent enkele globale rapporten uitdraaien.

Als ik kritisch naar het product kijk dan moet ik concluderen dat het nog niet gereed is voor invoering. Het product mist nog een aantal zaken zoals in de "to do list" van paragraaf 6.3 reeds te vinden was. Het product kan echter wel bij (potentiële) klanten van Metrix worden getoond in een demo.

9 Bronnenlijst

Literatuur

Redmond-Pyle, D & Moore, A. Graphical User Interface Design and Evaluation. Hemel Hempstead; Prentice Hall Europe, 1995

Rumbaugh, J e.a. Object Oriented Modeling and Design, New Jersey, Prentice Hall inc.

Tolido R.J.H. IAD, het evolutionair ontwikkelen van informatiesystemen. Schonehoven; Academic Service, 1997

Vandenbulcke, Databasesystemen voor de praktijk 6dr, Kluwer Bedrijfswetenschappen, 1997

Diverse auteurs, ASP data access, wrox press, 2000

Paul Wilton, beginning javascript, wrox press, 2000

Cliff Wooton, javascript, wrox press, 2001

Eric A. Smith, active server pages, IDG books, 2000

Petroutos Evangelos, Database programming with visual basic 6, sybex, 2000

Internet

DHTML library

<http://msdn.microsoft.com/workshop/author/dhtml/reference/objects.asp>

10 Woordenlijst

drie lagen model	Een strategie om een applicatie uit meerdere lagen te laten bestaan met als doel om de onderhoudbaarheid en herbruikbaarheid te optimaliseren.
.NET	Nieuw ontwikkelplatform van Microsoft.
Active server pages	Techniek om statische HTML te koppelen aan dynamische content zoals een database.
AICC	Een E-learning standaard in de luchtvaart industrie.
ASP	Afkorting van active server pages.
Browser	Een programma waarmee je internet pagina's kunt bezoeken.
DHTML	Dynamisch aanspreken van objecten in HTML.
DLL	Afkorting van Dynamic link library.
Dynamic link library	Een bestand wat functionaliteit bevat en kan worden aangesproken door andere applicaties en processen.
Field service management software	Software dat word gebruikt door bedrijven met service activiteiten met als doel de activiteiten te coördineren.
GUIDE	Graphical user interface design en Evaluation; ontwikkeltechnieken.
HTML	De programmeertaal (opmaakcode) van een webpagina.
IAD	Interactive application development, methodiek ontworpen door Cap Volmac om informatie systemen te ontwikkelen.
Javascript	Script taal (client-side).
LMS	Een learning management system. Een systeem om leertrajecten, gebruikers en activiteiten en managen en eventueel voortgang te bewaken.
Pilot	Een apart in te voeren bouweenheid van een applicatie.

proof of concept

Een experiment.

SCORM

Sharable Content Object Reference Model. Dit is een applicatieprofiel (referentiemodel) dat is ontwikkeld door ADL. ADL, Advanced Distributed Learning, is een Amerikaans overheidsinitiatief van het ministerie van defensie met als doel het promoten en stimuleren van het ontwikkelen van on-line leermaterialen en software voor onderwijs en training. Uitwisselbaarheid van onderwijsmateriaal is een primaire doelstelling. Om dit te bewerkstelligen heeft men het SCORM model ontwikkeld.

stand-alone

Programmatuur die zonder hulp van andere applicaties of computers kan werken

Visual basic

4^{de} generatie programmeertaal.