

Interactief Roosteren

Afstudeerscriptie



Versie : 2.0
Datum : 24 mei 2005
Naam : Yannick Haacke
Studentnummer : 1128185
Eerste examiner : Maarten Schnerr

Samenvatting

Dit document beschrijft het project Interactief Roosteren. Dit project is binnen de Hogeschool van Amsterdam in opdracht van Jan Broos uitgevoerd door Yannick Haacke, als afstuderend Information Engineer. Het project was gericht op het verbeteren van incidentele roostering en de roosterpublicatie binnen de HvA. De probleemstelling is tweeledig. Ten eerste is er binnen de HvA geen geschikte, uniforme manier om de roosters te publiceren, ten tweede is er geen (technische) mogelijkheid om wijzigingen op het rooster te kunnen administreren. Na onderzoek (interviews, literatuur) heb ik besloten dat de beste oplossing voor de HvA een webbased Roosterportal is.

Na de nodige voorbereiding (Plan van Aanpak, Onderzoeksrapport, Functioneel Ontwerp) heb ik deze Roosterportal uiteindelijk gerealiseerd. Het product is ontwikkeld op een bestaande Content Management Server van de HvA.

Tijdens de ontwikkeling heb ik verschillende keuzes gemaakt aangaande de functionaliteit van het product en de wensen van de eindgebruikers. De verantwoording van deze keuzes wordt uiteengezet in hoofdstuk 4.

Concluderend kan ik twee zaken vaststellen. Ten eerste is het zaak dat de HvA zich sowieso bezig gaat houden met het verbeteren van het roosterproces, ook het proces na de initiële roostering (incidentele roostering en publicatie). Ten tweede kan ik vaststellen dat er na dit project een applicatie beschikbaar is die de problemen inzake het roosterproces kan helpen oplossen. Het "proof-of-concept" proces is succesvol doorlopen.

Voorwoord

Dit is mijn afstudeerscriptie. Bijna een half jaar ben ik als Information Engineer werkzaam geweest bij de Hogeschool van Amsterdam. In opdracht van de afdeling Strategische Informatievoorziening en Jan Broos in het bijzonder heb ik me dit half jaar bezig gehouden met het ontwikkelen van een Roosterportal. Ik heb erg plezierig gewerkt dankzij de leuke collega's en de structurele begeleiding van Jan Broos. Hiervoor wil ik Jan dan ook graag erg bedanken.

Ik ben erg blij met de opdracht die ik heb mogen uitvoeren. Omdat ik de opdracht vanaf de grond af aan in mijn eentje heb mogen doen heb ik de kans gehad om alle facetten van een Information Engineer aan bod te laten komen. Van de inventarisatiefase tot aan de uitvoering, inclusief ontwikkeling. Aangezien het mijn afstudeerstage is, is het erg belangrijk om te kunnen laten zien wat ik allemaal heb opgestoken gedurende de afgelopen drieëneenhalf jaar. Ik denk dat het gelukt is om al deze dingen te laten zien. Ik ben dan ook zeer tevreden met het eindresultaat.

Vanuit de Hogeschool van Utrecht heb ik begeleiding mogen ontvangen van Maarten Schnerr. Deze begeleiding heb ik als zeer prettig ervaren. Samen met de nodige opbouwende opmerkingen heb ik de samenwerking als zeer fijn en ontspannen ervaren. Ik ben blij dat we het ook over andere dingen dan mijn project hebben kunnen hebben. Mijn dank gaat hiervoor uit naar Maarten Schnerr.

Yannick Haacke

Mei 2005

Inhoudsopgave

Samenvatting	2
Voorwoord.....	3
1. Inleiding	5
2. Opdracht	6
2.1 . Organisatiebeschrijving	6
2.2. Probleemstelling.....	7
2.3. Opdrachtomschrijving	7
2.4. Projectkader	7
2.5. Uitgangssituatie	8
2.6. Draagvlak	9
3. Voorbereiding.....	10
3.1. Activiteiten	10
3.2. Op te leveren producten.....	■ ■
4. Verantwoording.....	i 2
4.1. Aanpak.....	12
4.2. Techniek.....	12
4.3. Functionaliteit.....	13
5. Uitvoering en analyse	16
5.1. Voorbereiding	16
5.2. Uitvoering	16
5.3. Terugkoppeling	17
5.4. Implementatie.....	17
6. Conclusies en aanbevelingen.....	19
7. Evaluatie	20
8. Figuurindex	21
9. Geraadpleegde literatuur	21
Bijlagen	
1. Plan van Aanpak	
2. Functioneel Ontwerp	
3. Sitemap	
4. Database ontwerp	

1. Inleiding

Het project Interactief Roosteren is binnen de Hogeschool van Amsterdam uitgevoerd op de afdeling Strategische Informatievoorziening. Dit is een afdeling die op ICT gebied de lange termijnplanning in de gaten houdt. Dit houdt in dat medewerkers op deze afdeling zich richten op de stroomlijning van processen en programmatuur. Omdat de afdeling nog niet zo lang bestaat is hij nog niet terug te vinden in het organogram van de HvA (zie figuur 1, pagina 6). Strategische Informatievoorziening is een stafafdeling die direct rapporteert aan het College van Bestuur. Een uitgebreide organisatiebeschrijving is terug te vinden in paragraaf 2.1.

Het project (en de problemen waaruit het project voortgevloeid is) speelde eigenlijk al langer, maar er waren niet genoeg resources om het project te kunnen starten. Dit is de reden dat het mogelijk is geweest dat ik het project helemaal alleen heb mogen doen.

2. Opdracht

Dit hoofdstuk zal naast de opdrachtomschrijving ook de probleemstelling en de uitgangssituatie bevatten. De uitgangssituatie beschrijf ik qua techniek, processen en mensen. Maar allereerst geef ik een beschrijving van de organisatie waarbinnen het project wordt uitgevoerd: de Hogeschool van Amsterdam.

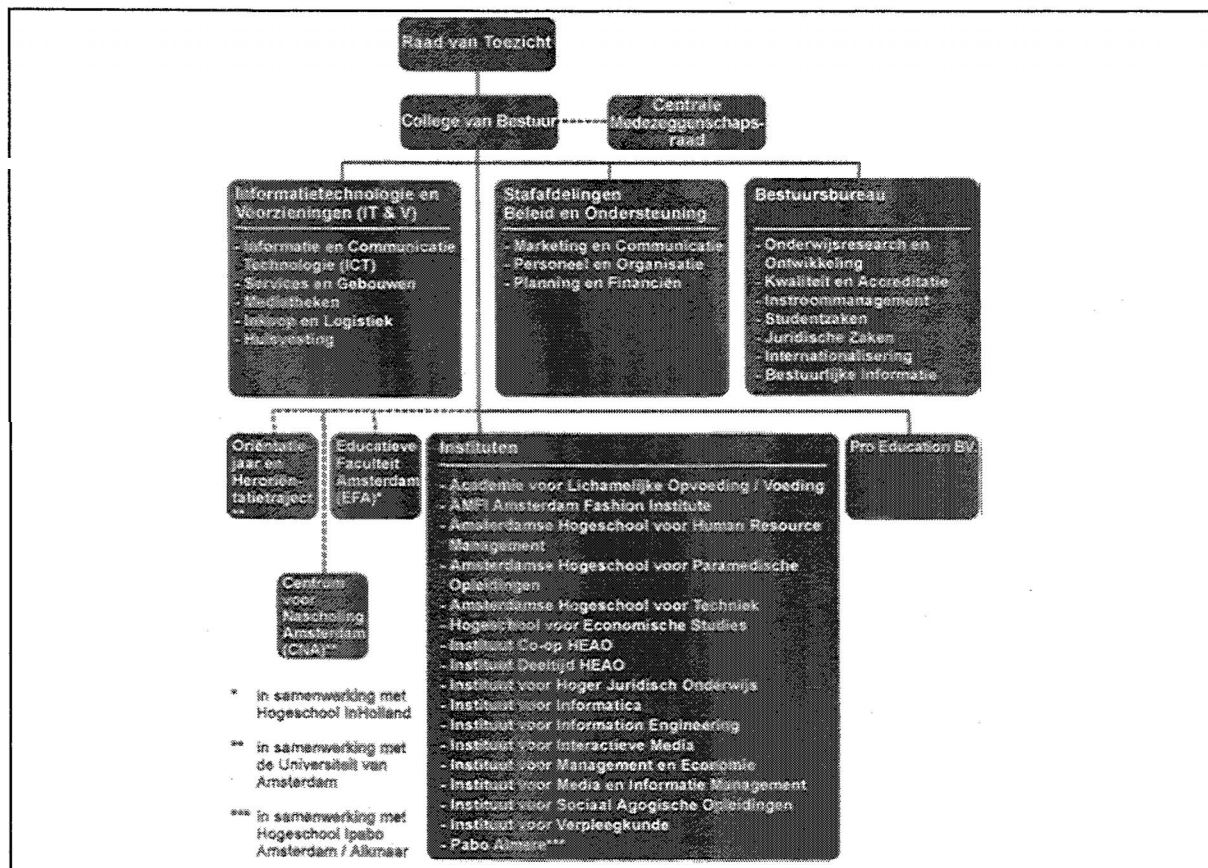
2.1. Organisatiebeschrijving

Het onderstaande organogram geeft een overzicht van de structuur van de Hogeschool van Amsterdam. De genoemde instituten bestaan elk weer uit verschillende opleidingen. Hoeveel dat er zijn verschilt per instituut.

De organisatiestructuur van de HVA schommelt eigenlijk tussen twee structuren: de divisiestructuur en de professionele organisatie. De divisiestructuur komt duidelijk naar voren, aangezien de instituten verantwoording over het aantal studenten en het bestede geld dienen af te leggen aan het college van bestuur. De organisatie is afhankelijk van deze gegevens in verband met de subsidie vanuit de overheid.

Tegelijkertijd komt ook de professionele organisatie naar voren, aangezien de instituten zelf verantwoordelijk zijn voor de uitvoering van het primaire proces (onderwijs). Met dit primaire proces houdt het college van bestuur zich niet bezig.

Ook de communicatie naar buiten verloopt niet via de top van de organisatie, de instituten communiceren zelf direct naar de buitenwereld.



Figuur 1: Organogram van de Hogeschool van Amsterdam

2.2. Probleemstelling

De probleemstelling voor dit project bestaat uit twee delen:

1. De huidige publicatievorm van de roosters is niet geschikt om roosterwijzigingen bruikbaar op te leveren.
2. Instituten hebben geen technische mogelijkheid om roosterwijzigingen door te voeren in het rooster, ze gebruiken ieder eigen systemen voor de instituutsroostering.

2.3. Opdrachtomschrijving

Vanuit de hiervoor genoemde twee problemen zijn mijn doelstellingen voor het project vast te stellen. In eerste instantie heb ik deze als volgt vastgesteld in vijf punten:

1. Voer onderzoek uit om erachter te komen hoe instituten / docenten / studenten hun rooster het liefst gepubliceerd zien (via mail, webcalendar, het Roxen CMS, etc.).
2. Onderzoek de integratiemogelijkheden van de in gebruik zijnde versie van GP-Untis en onderzoek de mogelijkheden die de nieuwste versie van GP-Untis biedt.
3. Realiseer een toekomstvaste en flexibele oplossing, op basis van de huidige technische mogelijkheden en pakketten.
4. Onderzoek de mogelijkheden voor een systeem waarmee instituten roosterwijzigingen terug kunnen koppelen naar GP-Untis (waar vanuit weer efficiënt gepubliceerd kan worden).
5. Realiseer deze eventuele oplossing.

Na mijn eerste onderzoeksfase bleek echter dat de technische mogelijkheden van het roosterprogramma GP-Untis te beperkt waren om een terugkoppelingssysteem te kunnen realiseren. Na nader overleg en onderzoek is gebleken dat de beste oplossing één systeem zou zijn, waarin zowel de publicatie als de "terugkoppeling" (roosterwijzigingen en publicatie van deze wijzigingen) wordt verzorgd. Mijn definitieve opdrachtomschrijving luidt daarom als volgt:

1. Voer onderzoek uit om erachter te komen hoe instituten / docenten / studenten hun rooster het liefst gepubliceerd zien (via mail, webcalendar, het Roxen CMS, etc.).
2. Realiseer een toekomstvaste en flexibele oplossing op basis van de huidige technische mogelijkheden en pakketten, die voorziet in initiële publicatie van de roosters, het bijhouden van wijzigingen en de publicatie van die wijzigingen.

2.4. Projectkader

Gezien de kosten van verschillende in gebruik zijnde softwarepakketten binnen de Hogeschool van Amsterdam, is binnen het projectkader vastgesteld dat een uiteindelijke oplossing gebaseerd dient te worden op bestaande systemen. Dit houdt in dat er keuze is uit de volgende systemen binnen de HvA:

- Roostering
 - GP-Untis (zie www.untis.nl)
- Publicatie
 - Sun Webcalendar (zie <http://www.sun.com/software/>)
 - Roxen Content Management Systeem (zie www.roxen.com)
 - Standaard e-mail
 - Standaard webpagina

2.5. Uitgangssituatie

De uitgangssituatie binnen de Hogeschool van Amsterdam is enorm verdeeld. Het huidige roosterproces verloopt momenteel in drie fases:



Figuur 2: Fasering roosterproces

In de derde fase wordt het rooster door het roosterbureau geleverd aan de verschillende instituten. De vorm waarin dat gebeurt verschilt per instituut. De meeste instituten krijgen de roosters in een digitale vorm aangeleverd, dat kan in een Excel-sheet zijn, maar dat kan ook in een ander formaat. Een aantal instituten maakt gebruik van een binnen de HvA ontwikkeld programma genaamd ROOS, wat kan voorzien in roostering. Dit programma is ontwikkeld in het databaseprogramma Filemaker Pro en werkt stand-alone, andere instituten hebben dus geen inzicht in dit programma. Naast deze digitale vormen zijn er ook instituten die de roosters zelfs nog op papier beheren.

In welke vorm het rooster ook wordt beheerd, een aantal problemen is op alle instituten van toepassing:

1. Niemand anders dan de instituutstroosteraars heeft inzicht in de nieuwste versie van de roosters.
2. Er is geen uniforme manier van publiceren van de initiële roosters.
3. Er is geen uniforme manier van attenderen van docenten / studenten op roosterwijzigingen.

2.6. Draagvlak

Hoewel de HvA als organisatie zoals eerder gezegd beschouwd kan worden als een mengeling tussen een organisatie met een divisiestructuur en een professionele organisatie, kunnen de medewerkers van de HvA, net als in veel andere organisaties, qua vastberadenheid inzake hun manier van werken wel vergeleken worden met medewerkers uit een machineorganisatie. Uit de interviews valt op te maken dat ze een manier van werken ontwikkelen en vervolgens moeilijk over zijn te halen een andere manier van werken aan te leren. Met deze eigenschap zal rekening moeten worden gehouden tijdens het ontwikkelen en tijdens de implementatie. Uiteraard probeer ik qua functionaliteit zo veel mogelijk te voldoen aan de wensen van de eindgebruikers door interviews af te nemen, maar uiteindelijk zal ik een manier moeten vinden om de eindgebruikers enthousiast te maken voor het product. Ik denk dat ik voor dit doel een aantal functies heb gerealiseerd, deze zal ik nader benoemen in het hoofdstuk Uitvoering.

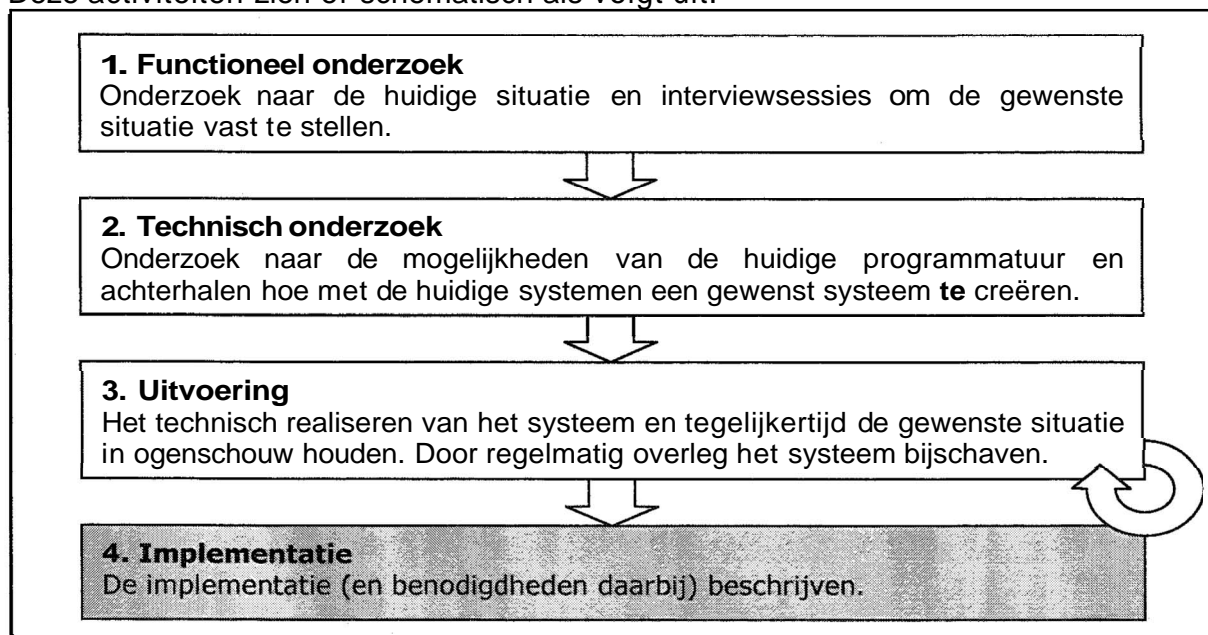
3. Voorbereiding

Dit hoofdstuk zal globaal beschrijven hoe ik van plan wat het project uit te voeren, de gedetailleerde beschrijving hiervan is terug te vinden in mijn Plan van Aanpak (zie bijlage 1). Daarnaast vindt u in dit hoofdstuk welke producten aan het eind van het project worden opgeleverd.

3.1. Activiteiten

De activiteiten die naar aanleiding van deze opdrachtomcchrijving en uitgangssituatie zullen worden verricht binnen dit project zullen alle facetten van Information Engineering bevatten. Allereerst wordt een onderzoek uitgevoerd naar de wens van **de** klant (studenten en docenten), hiervoor zullen interviewsessies worden gehouden met relevante personen. Vervolgens worden (binnen het projectkader) de technische mogelijkheden onderzocht om te kunnen voldoen aan de wensen van de klant. Naar aanleiding van dit onderzoek wordt een advies uitgebracht aan de opdrachtgever. Na goedkeuring van dit advies **zal** uiteindelijk ook het product zelf ontwikkeld worden. Ten **slotte** zal worden gekeken naar de afronding van het project, bestaande uit het maken van handleidingen en beschrijven van de implementatie. Gezien de beperkingen van het project (er is geen **tijd/geld** om mensen op te leiden of nieuwe mensen aan te nemen) zal het project beschouwd worden als een "**proof of concept**". Dit houdt in dat het product gereed gemaakt wordt voor productie en de implementatie wel gepland kan worden, maar dat het product niet daadwerkelijk geïmplementeerd zal worden tijdens mijn project. In een vervolgtraject kan wel overgegaan worden tot implementatie.

Deze activiteiten zien er schematisch als volgt uit:



Figuur 3: Fasering project

Een complete beschrijving van alle activiteiten, tevens geplaatst in een planning is terug te vinden in mijn Plan van Aanpak (zie bijlage 1).

3.2. Op te leveren producten

Op basis van de opdrachtomschrijving en de beschreven activiteiten zal ik aan het einde van het project de volgende producten opleveren:

- Roosterportal
 - De webapplicatie die voldoet aan de volgende eisen:
 - Initiële publicatie van de roosters (vanuit GP-Untis)
 - Mogelijkheid tot wijzigen van de lessen
 - Mogelijkheid tot publiceren van de wijzigingen (headlines)
 - Gepersonaliseerde informatievoorziening (alleen informatie zien die jou aangaat, als student of docent)
- Documentatie
 - Handleidingen per rol binnen de portal:
 - Beheerder
 - Docent
 - Student
 - Overige documentatie:
 - Plan van Aanpak
 - Onderzoeksrapport (incl. interviewverslagen)
 - Functioneel ontwerp
 - Technisch ontwerp
 - Implementatieplan

Al deze producten beslaan samen het complete project **en** voorzien in voldoende informatie voor een geslaagde implementatie van **de Roosterportal**.

4. Verantwoording

In dit hoofdstuk zal ik de keuzes die ik tijdens het project gemaakt heb verantwoorden. Ik zal toelichten waarom ik bepaalde keuzes heb gemaakt ten aanzien van aanpak, techniek en functionaliteit.

4.1. Aanpak

Om te beginnen de manier waarop ik het project aangepakt heb. Ik heb het project uitgevoerd middels de projectmatige aanpak, zoals ik die al sinds het begin van mijn studie gebruikt heb voor alle projecten. Dat wil zeggen dat ik ben begonnen met het schrijven van een Plan van Aanpak, waarin ik het project beschrijf en afkader en waarin ik ook een planning en fasering opneem. Voor een aantal vereisten heb ik gebruik gemaakt van andere technieken die ik tijdens mijn studie geleerd heb. Zo heb ik de functionele eisen voor het project opgesteld aan de hand van de MoSCoW prioritering, afkomstig uit de Dynamic Systems Development Method (DSDM) manier van werken. Deze prioritering is terug te vinden in het Functioneel Ontwerp (zie bijlage 2). In dit Functioneel Ontwerp zijn ook voorbeeldschermen te vinden van de applicatie. Deze zijn niet geheel identiek aan het eindresultaat, maar ze geven een goed beeld.

Ook tijdens de ontwikkelfase binnen mijn project heb ik gewerkt volgens de DSDM methode. Dat wil zeggen dat ik gewerkt heb met een evolutionair prototype en onder invloed van gebruikersparticipatie. Dankzij deze iteratieve, incrementele manier van ontwikkelen heb ik snel een werkend prototype kunnen opleveren, waarop gebruikers (en de opdrachtgever) hun ideeën los hebben kunnen laten. Hierdoor heb ik een snelle manier van werken gecreëerd, terwijl ik me geen zorgen hoefde te maken over het uiteindelijke eindresultaat, aangezien de eindgebruikers er continu hun wensen in hebben kunnen laten verwerken.

4.2. Techniek

Allereerst was mijn plan om de initiële publicatie te laten verlopen via een nieuw te ontwikkelen **website**, vanuit een GP-Untis bestand. Vervolgens zouden wijzigingen via een **website** geautomatiseerd teruggekoppeld worden naar GP-Untis, waarvan de publicatie dan weer, ook geautomatiseerd, plaats zou vinden naar de **website**. Na onderzoek bleek dit niet haalbaar, omdat GP-Untis niet geschikt is voor roosterwijzigingen. Het maken van een eerste rooster op basis van alle aangeleverde gegevens is geen probleem, maar zodra er een wijziging wordt doorgevoerd, wordt het gehele rooster opnieuw **berekend**. Dit kan bijvoorbeeld tot gevolg hebben dat door een roosterwijziging op dinsdag de lessen van donderdag verplaatst worden naar vrijdag. Dat is uiteraard geen werkbare situatie.

De oplossing hiervoor is het bijhouden van een eigen database van de lessen, die wel gevuld wordt vanuit GP-Untis. Op deze manier kunnen de wijzigingen ook in deze eigen database bijgehouden worden. Om deze reden heb ik gekozen voor één totaaloplossing die zowel de publicatie als de wijzigingen voor zijn rekening neemt.

Toen moest ik op zoek naar een basis om het systeem op te bouwen. Lettend op de beschikbare systemen (zie paragraaf 2.4 Projectkader), bleek deze keuze snel

gemaakt. Één systeem voldeed reeds aan alle eisen: het Roxen Content Management Systeem. Dit CMS beschikt over een eigen (MySQL) database en server side scripting. Met behulp van server side scripting is het mogelijk om op een laag niveau te programmeren binnen webpagina's waardoor het mogelijk is om dynamische pagina's te ontwikkelen. Tevens beschikt Roxen over een module die koppeling mogelijk maakt met de LDAP server¹ van de HvA, hierdoor hoeft het systeem geen eigen database van gebruikers en wachtwoorden bij te houden. Wanneer we dit alles in ogenschouw nemen, blijkt dat het Roxen CMS de beste keus is om de Roosterportal op te bouwen.

4.3. Functionaliteit

Een aantal functionaliteiten van het systeem lag vooraf al vast. Uiteraard moest het mogelijk zijn om roosters die gemaakt zijn in het roosterprogramma van de HvA (GP-Untis) te kunnen publiceren. Dit heb ik mogelijk gemaakt door middel van een importeerfunctie die bestanden vanuit GP-Untis rechtstreeks kan importeren in de eigen database van het systeem. Natuurlijk moet het dan ook mogelijk zijn om deze lessen te wijzigen, en om lessen te verwijderen en nieuwe lessen toe te voegen. Naast deze basisfunctionaliteit heb ik een aantal dingen bedacht om het systeem beter werkbaar / geschikter / leuker te maken:

Lokaties

Omdat de verschillende lokaties van de HvA (5 in totaal) ieder hun eigen versie van GP-Untis draaien is het voor het overzicht beter om op iedere locatie een losse versie van de Roosterportal te laten draaien. Dit heeft qua performance ook voordelen aangezien er minder gebruikers (tegelijktijd) op het systeem zullen werken. Het systeem is volledig voorbereid op het draaien op verschillende lokaties. Door in de voorkeuren van de site aan te geven dat het systeem op een bepaalde locatie draait, wordt automatisch de interface aangepast en ook de lestijden zullen zich automatisch aanpassen aan die locatie.

Rollen

Door drie "rollen" binnen het systeem te creëren (beheerder, docent, student) is het gehele systeem beter te beveiligen en is het systeem ook beter personaliseerbaar. De gehele Roxen-intranetsite is beveiligd op basis van de LDAP login van de gebruikers, op basis van deze login wordt bepaald welke rol de gebruiker krijgt. Aan de hand van deze rol krijgt de gebruiker een gepersonaliseerde website te zien. Allereerst krijgen gebruikers op de homepage alleen headlines (zie onder) te zien die voor hen bedoeld zijn. Dit heeft als groot voordeel dat gebruikers niet worden overspoeld met informatie waar ze toch niets aan hebben. Ook kunnen docenten en studenten dankzij de personalisatie op de knop "Mijn rooster" klikken, waarna ze zonder verdere handelingen hun eigen rooster voor de lopende week te zien krijgen. Zo hoeven ze niet iedere keer hun eigen docent- of klascode en het huidige weeknummer op te zoeken. Een compleet overzicht van de website en welke mogelijkheden de drie rollen hebben is terug te vinden in de sitemap (zie bijlage 3).

¹ Een LDAP server is een server die alle gegevens van gebruikers, zoals gebruikersnamen en wachtwoorden, bevat. De server wordt gebruikt wanneer gebruikers in willen loggen op een werkstation.

Headlines

Dit zijn korte nieuwsberichten die een beheerder aan kan maken via de site. Deze berichten kunnen bijvoorbeeld informatie bevatten over roosterwijzigingen, het uitvallen van docenten, etc. Als een beheerder een headline aanmaakt, selecteert hij gelijk voor wie de headline bedoeld is. Dat kan op basis van de rol van de gebruikers (docent, student), maar ook nog gedetailleerder op basis van docent- of klascode. Door bijvoorbeeld 4 docenten en 6 klassen te selecteren wordt de betreffende headline alleen zichtbaar voor die specifieke gebruikers. Daarnaast geeft de beheerder ook nog aan op welke dagen de headline zichtbaar dient te zijn (d.m.v. een begin- en einddatum). Dit alles zorgt ervoor dat gebruikers niet worden overspoeld met informatie. Ze krijgen alleen informatie te zien die ook daadwerkelijk voor hen bedoeld is.

RSS

De laatste 20 gepersonaliseerde headlines zijn niet alleen via de site op te vragen. Ze zijn ook beschikbaar via een zogenaamde RSS feed:

Als je elke dag naar dezelfde websites surft om te kijken of er iets nieuws op staat, kun je overwegen een softwareprogramma te kopen/downloaden die al dat dagelijkse zoeken en surfen uit handen neemt. Een RSS reader verzamelt koppen en intro's van je favoriete websites op één pagina, in overzichtelijke categorieën. Als er iets interessants bij zit, dan klik je simpelweg op de kop of intro, die je naar de betreffende website brengt.²

Met behulp van de RSS feed kunnen gebruikers dus, zonder dat ze naar de website hoeven, in één oogopslag de laatste 20 headlines volgen. Dit hoeft overigens niet per se op een PC te zijn. Er bestaan ook RSS readers voor bijvoorbeeld PocketPC's en mobiele telefoons. Hierdoor kunnen gebruikers dus eigenlijk zonder enige moeite op de hoogte blijven. Ik denk persoonlijk dat dit enorm bij zal dragen aan het al dan niet slagen van de Roosterportal. Als gebruikers iedere keer naar de website moeten surfen en in moeten loggen om te kijken of er wijzigingen zijn, dan kan deze drempel uiteindelijk te hoog worden, met als gevolg dat de gebruikers de portal niet meer gaan gebruiken. Dat is dankzij de RSS feed gelukkig niet het geval. Als de gebruikers uitgebreidere informatie willen dan de headlines (een rooster bijvoorbeeld) kunnen ze alsnog naar de website surfen.

Printen

Uiteraard zijn er mensen die hun rooster toch het liefste op papier hebben om op te vouwen en in hun agenda te stoppen. Om deze reden heb ik de mogelijkheid gemaakt om opgevraagde roosters af te drukken. Het rooster verschijnt dan in een printvriendelijke opmaak. Doordat mensen hun weekrooster zelf uit kunnen printen, in tegenstelling tot wanneer ze aan het begin van een blok hun rooster voor de komende 10 weken krijgen, zijn mensen er zeker van dat ze de laatste mogelijke versie (met alle doorgevoerde wijzigingen) in hun bezit hebben. Lessen die gewijzigd zijn sinds het begin van het blok worden bovendien gemarkeerd op de uitdraai, zodat snel zichtbaar is dat de les veranderd is.

² Bron: <http://www.hccmagazine.nl/>

Webcalendar

Opgevraagde roosters kunnen met één druk op de knop tevens worden geëxporteerd naar een zogenaamd iCalendar bestand. Dit is een tekstbestand in een speciale opmaak dat direct geïmporteerd kan worden in de webcalendar, waar alle medewerkers en studenten van de HvA toegang toe hebben. De webcalendar is een webbased kalender waar iedereen zijn eigen afspraken in kan plaatsen. Bovendien kan iedereen via de webcalendar zien wanneer iemand anders beschikbaar is. Wanneer mensen hun rooster snel in de webcalendar kunnen plaatsen wordt het eenvoudig om afspraken om lessen heen te plannen.

Gebruikers zouden deze functionaliteit ook kunnen gebruiken omdat de webcalendar gesynchroniseerd kan worden met een PDA, zo kunnen de **lessen** dus automatisch in de digitale agenda gezet worden.

5. Uitvoering en analyse

In dit hoofdstuk worden de concrete werkzaamheden benoemd die ik in de verschillende fases van het project heb gedaan. Tevens wordt daarbij benoemd hoe goed ik denk dat de werkzaamheden verlopen zijn en welke invloed ze op het project gehad hebben. Ten slotte wordt in dit hoofdstuk ingegaan op de implementatie van het project. Hoewel ik tijdens het schrijven van deze scriptie nog niet toe was aan de implementatie, kan ik wel een aantal facetten benoemen dat van belang is tijdens de implementatie.

5.1. Voorbereiding

De voorbereiding is begonnen met het schrijven van het Plan van Aanpak. Dit is in eerste instantie voorspoedig verlopen, de opdrachtgever was, op wat kleine aanmerkingen na, direct tevreden met de eerste versie. Naar aanleiding van dit Plan van Aanpak kon ik toen aan de slag met de interviewsessies. Deze interviews leverden twee belangrijke gegevens op:

- Veel docenten staan niet echt open voor een nieuw systeem, aangezien het huidige systeem voor hen min of meer goed werkt. Ze zien geen verbetering in een centraal punt voor alle roosters.
- De webcalendar, in eerste instantie één van de speerpunten van de opdrachtgever, blijkt niet veel gebruikt te worden en wordt ook niet gezien als een handige tool voor het beheren van de roosters.

Naar aanleiding van deze interviews was het dus zaak om het Plan van Aanpak aan te passen en het project te richten op het ontwikkelen van één centraal systeem, niet de webcalendar. En bovendien, nog belangrijker, dient er functionaliteit te komen in dat systeem die zoveel meer biedt dan de huidige oplossingen dat docenten bereid zullen zijn over te stappen.

Ik denk dat de interviews erg belangrijk zijn geweest voor het project. De opdrachtgever had een bepaald idee bij een roostersysteem, maar dat bleek niet overeen te komen met een systeem dat de eindgebruikers zouden willen gebruiken. Dankzij de interviews komt er uiteindelijk een systeem dat voldoet aan de wensen van de eindgebruikers.

5.2. Uitvoering

De gehele Roosterportal is ontwikkeld op het Roxen Content Management Systeem. Dit systeem beschikt, zoals eerder genoemd, over server **side** scripting. Deze manier van ontwikkelen komt overeen met die van ontwikkelen met behulp van bijvoorbeeld PHP of ASP (veelgebruikte server **side** scripting talen op het Internet). Het PHP en ASP had ik zelf al ervaring, maar met Roxen had ik totaal geen ervaring. Ik kende **het** hele pakket niet zelfs. Om deze reden was het belangrijk om mezelf eerst deze 'programmeertaal' aan te leren. Een voordeel is wel dat de Roxentaal (RXML) gebaseerd is op XML, een metataal die ik al geleerd heb tijdens mijn minor E-Business. Op basis van mijn XML kennis heb een vrij vlotte start kunnen maken met het ontwikkelen binnen Roxen. Nadat ik de basis van RXML onder de knie had, ben ik me bezig gaan houden met het ontwerpen van de database die als basis zou dienen voor het systeem. Deze database moet, naast alle lessen uiteraard, ook de

headlines bevatten evenals de gegevens om headlines te kunnen personaliseren. De complete indeling van de database, inclusief ERD is terug te vinden in de **database-indeling** (zie bijlage 4).

5.3. **Terugkoppeling**

Tijdens de ontwikkeling heb ik verschillende malen contact gehad met docenten, studenten en andere mensen om het prototype op dat moment te beoordelen. De aanwijzingen die zij mij op dat moment gaven zijn erg belangrijk geweest tijdens de ontwikkeling. Zonder deze input komt er uiteindelijk een systeem dat voldoet aan mijn eigen wensen of, met een beetje geluk, aan de wensen van de opdrachtgever. Een aantal kleine, maar belangrijke, functies is dankzij deze feedback tot stand gekomen. De mogelijkheid tot het ongedaan maken van een import van lessen, of de manier waarop wijzigingen van lessen worden bijgehouden zijn voorbeelden van functionaliteit die ik zelf in eerste instantie niet geïmplementeerd zou hebben omdat ik er simpelweg niet aan gedacht had.

5.4. **Implementatie**

Zoals eerder gezegd was ik tijdens het schrijven van deze scriptie nog **niet** toe aan de implementatie van het project. Desalniettemin wil ik toch de belangrijkste facetten benoemen die van belang zijn tijdens het implementeren van het systeem. Deze punten samen vormen nog geen volledig **implementatieplan**, maar geven toch inzicht in waar op gelet dient te worden wanneer het systeem geïmplementeerd gaat worden:

- **ICT Infrastructuur**

Hoewel de **ICT** hardware een ondergeschikte rol heeft, is deze niet minder belangrijk. Tijdens het ontwikkelen heb ik gewerkt op een losstaande ontwikkelserver, die snel genoeg was voor de Roosterportal. Tegelijkertijd heb ik echter ook gemerkt dat de productieserver van Roxen (de server waar de applicatie uiteindelijk op gaat draaien) eigenlijk niet voldoet aan de eisen die eraan gesteld worden. De performance is niet genoeg om er goed op te kunnen werken. Als mijn systeem uiteindelijk op deze server moet gaan draaien, dan zal de performance omhoog moeten, omdat mensen anders af gaan zien van het gebruik ervan, aangezien het geheel gewoon niet snel genoeg werkt.

Naast deze hardware-kwestie dient er ook een SSL certificaat aangevraagd te worden voor de LDAP koppeling. Een SSL certificaat is een manier om gegarandeerd een beveiligde verbinding te creëren tussen de Roxen server en de LDAP server. Dit is uiteraard een vereiste, aangezien er gebruikersnamen en wachtwoorden tussen deze twee servers verstuurd gaan worden.

Ook zal de ICT Infrastructuur ervoor moeten zorgen dat de Roosterportal gemakkelijk te bereiken is, door een gemakkelijk internetadres te verzorgen, bijvoorbeeld <http://roosters.hva.nl>.

- **Opleiden**

Om de Roosterportal zonder problemen in gebruik te nemen zal een aantal mensen bekend gemaakt dienen te worden met het systeem en de manier van werken. Ten eerste de locatieroosteraars, deze moeten voortaan alle

roosters exporteren naar het formaat dat de Roosterportal kan importeren. Dit is op zich geen probleem, aangezien dit een ingebouwde functie is van GP-Untis. Deze geëxporteerde bestanden dienen echter wel door deze roosteraars op de Roxen server geplaatst te worden op de juiste locatie, zodat ze door het systeem gevonden kunnen worden. Dit zal hen geleerd moeten worden.

Daarna zullen de belangrijkste gebruikers opgeleid dienen te worden. De instituutsroosteraars. Deze roosteraars krijgen binnen de Roosterportal de "beheerdersrol". Dit houdt in dat zij de lessen moeten importen en beheren. En daarnaast houden zij ook de headlines bij. Dit is in principe geen ander werk dan dat ze nu doen, maar ze moeten het wel op een heel andere manier gaan doen uiteraard. Om deze manier van werken onder de knie te krijgen zal er eerst geoefend dienen te worden.

Alle rollen binnen de Roosterportal (beheerde, docent, student) krijgen overigens een eigen handleiding waarin staat beschreven hoe hun persoonlijke functies binnen het systeem werken.

- Voorlichting / promotie

Tot slot zullen de mensen natuurlijk ook te weten moeten komen dat de Roosterportal bestaat. Door posters op te hangen, nieuwsberichten te plaatsen op intranetsites en bijvoorbeeld emailberichten rond te sturen breng je mensen op de hoogte van het bestaan van **en** de voordelen van de Roosterportal.

Uit deze opsomming blijkt wel dat een implementatie niet één, twee, drie te realiseren is. Er dient goed gepland te worden en er moet van tevoren gezorgd worden voor voldoende ondersteuning. Zowel qua hardware als qua mensen.

6. Conclusies en aanbevelingen

Op basis van mijn opgedane kennis tijdens het project kan ik een aantal conclusies trekken. Deze zijn niet allemaal van toepassing op mijn specifieke project, maar mijns inziens wel van belang voor de Hogeschool van Amsterdam.

De incidentele roostering gedurende het lesblok dient veranderd te worden

In de huidige vorm loopt men tegen een aantal problemen aan bij het bijhouden van roosterwijzigingen. Ieder instituut heeft zijn eigen werkwijze ontwikkeld, wat de overzichtelijkheid niet ten goede komt. Door de verschillende manieren van bijhouden moeten de verschillende instituten de roosters ook in verschillende formaten aangeleverd krijgen, wat de locatieroosteraars weer extra werk oplevert. Met een uniforme manier van het bijhouden van roosterwijzigingen kunnen ten eerste de locatieroosteraars efficiënter werken. Ten tweede is er voor de uniforme manier van roosteren een draaiboek op te stellen zodat bij ziekte iemand anders het werk van de instituutstroosteraar over kan nemen. Dit is namelijk erg lastig als iedereen zijn eigen werkwijze ontwikkelt.

De roosterpublicatie dient verbeterd te worden

Omdat de instituten geen uniforme manier van incidentele roostering hebben is het ook niet mogelijk om één publicatiesysteem te realiseren. Voor de mensen die op zoek zijn naar een specifiek rooster is het lastig als roosters verspreid staat over verschillende locaties (al dan niet online). Het mooiste zou één webpagina zijn waar alle roosters te vinden zijn. Dat dit haalbaar is binnen de HvA blijkt uit het feit dat een aantal instituten reeds een min of meer geautomatiseerde intranetpagina heeft waar de roosters te vinden zijn. Er dient alleen nog voor gezorgd te worden dat alle instituten dit realiseren. En dan het liefst gecentraliseerd uiteraard.

Beide problemen zijn op te lossen door gebruik te maken van de door mij ontwikkelde Roosterportal. Mocht het, om welke reden ook, niet mogelijk zijn om deze Roosterportal te implementeren, dan is het aan te bevelen deze twee problemen toch aan te pakken. Door verbeteringen te realiseren op deze punten, wordt het gehele roosterproces gestroomlijnd, waardoor het mogelijk wordt de roosters beter te beheren. Centralisatie is een belangrijke stap hierin. Ook de mogelijkheid om gestandaardiseerde processen te definiëren zodat een willekeurig persoon met behulp van een draaiboek het roosterwerk op zich kan nemen is erg belangrijk. En uiteindelijk is de eindgebruiker altijd gebaat bij een betere publicatie van de roosters.

7. Evaluatie

Nu het project ten einde loopt wordt steeds duidelijker welke vorm het eindresultaat krijgt. Terugkijkend op de vooraf gestelde doelen kan ik concluderen dat aan alle producteisen is voldaan. Dat geeft mij uiteraard ook een voldaan gevoel.

Ik denk dat het product grotendeels een resultaat is van de prettige werksfeer waarin ik heb gewerkt. Wanneer ik ook maar hulp nodig had of iemand wilde spreken of interviewen, er werd altijd tijd voor me gemaakt. In de meeste gevallen zelfs direct. Dit is erg prettig, aangezien mijn project het enige werk is dat ik te doen heb gehad. Als ik vastliep was er ook geen ander werk te doen, maar gelukkig is dit dus niet voorgekomen.

Ik denk dat ik veel geleerd heb tijdens het project. Niet alleen op technisch gebied (de Roxen XML taal), maar ook over mezelf. Ik weet nu dat ik alle facetten beheers die van een **Information Engineer** verwacht worden. Ik heb van begin tot eind mijn eigen project onder controle gehad. Ik had alle onderdelen van een project al eens ingevuld, als onderdeel van een projectteam tijdens mijn studie, maar nu weet ik dus dat ik alles ook zelf kan.

Het eindproduct is in principe direct implementeerbaar (op het regelen van een snellere server en een SSL certificaat na), en kan dus als geslaagd worden beschouwd. Ik kan alleen maar hopen dat het uiteindelijk ook daadwerkelijk in gebruik genomen gaat worden.

8. **Figuurindex**

Figuur 1: Organogram van de Hogeschool van Amsterdam	6
Figuur 2: Fasering roosterproces	8
Figuur 3: Fasering project	10

9. **Geraadpleegde literatuur**

1. Bots, R.T.M. & Jansen, W. (2001).
 Organisatie en Informatie
 Groningen: Wolters-Noordhoff, vijfde druk
2. Severijnen, O. & Westbroek, R. (1999).
 Netwerk, Basisboek Professionele bedrijfscommunicatie
 Zuthpen: Thieme, eerste druk
3. Network Communication Design (2000).
 NCD HTML Design Guide
 <http://www.ncdesicin.org/>: Version 5.0 - PDF Edition
4. DSDM Consortium Nederland
 <http://www.dsdm.nl/>

Bijlage 1: Plan van Aanpak

Interactief Roosteren

Plan van Aanpak

1. Inleiding 2. Doelstelling 3. Scope 4. Organisatie 5. Risico's 6. Middelen 7. Tijdenlijn 8. Conclusie 9. Bijlagen



Versie : 1.2
Datum : 29 maart 2005
Naam : Yannick Haacke

Inhoudsopgave

1. Samenvatting.....	3
2. Opdracht.....	4
2.1. Aanleiding.....	4
2.2. Opdrachtgever.....	4
2.3. Opdrachtnemer.....	4
2.4. Probleemstelling / Doelstelling.....	4
2.5. Werkwijze	5
3. Beheersaspecten	7
3.1. Scope.....	7
3.1.1. Business-eisen.....	7
3.1.2. Levenscyclus processen	7
3.1.3. Functionaliteit.....	8
3.1.4. Fasering	8
3.2. Tijd.....	9
3.3. Middelen.....	9
4. Deliverables	10
5. Indices	11
5.1. Figuurindex	11
5.2. Tabelindex	11
Bijlage i : Planning.....	12

1. Samenvatting

Dit document beschrijft de aanpak en fasering van het project 'Interactief Roosteren', dat zal worden uitgevoerd door Yannick Haacke.

Dit project, uitgevoerd in opdracht van de Hogeschool van Amsterdam, zal zich richten op het oplossen van de problemen die de Hogeschool van Amsterdam op dit moment heeft met het roosteren. Momenteel zijn de roosters voor instituten, docenten en studenten niet goed of helemaal niet toegankelijk. Daarnaast worden roosterwijzigingen vaak niet doorgevoerd in het rooster, waardoor het overzicht verdwijnt.

~~Het~~ project zal twee hoofdonderdelen bevatten, te weten:

- Het vinden van een optimale publicatievorm voor de roosters.
- Het vinden van een gebruikersvriendelijke manier voor het terugkoppelen van roosterwijzigingen naar het roostersysteem (GP-Untis).

Beide onderdelen zullen bestaan uit twee fases, een onderzoeksfase en een programmeetfase. In de onderzoeksfase zal ik niet alleen de technische mogelijkheden onderzoeken, maar ook de wensen van de eindgebruikers achterhalen door middel van interviews. Als ik uiteindelijk duidelijk op papier heb wat de wensen zijn en hoe het technisch gerealiseerd kan worden dan ga ik de volgende fase in, het programmeren.

2. Opdracht

2.1. Aanleiding

Binnen de Hogeschool van Amsterdam wordt al lange tijd gesproken over het roosterproces. Dit verloopt niet optimaal, maar men heeft er ook niet echt een oplossing voor. De belangrijkste problemen zijn dat het rooster, nadat het opgesteld is in GP-Untis, niet up-to-date gehouden wordt op één centrale plaats. En, als gevolg hiervan, worden mensen niet goed op de hoogte gehouden van roosterwijzigingen. Binnen de Hogeschool van Amsterdam is de afdeling Strategische Informatievoorziening (SI) belast met dit probleem, maar de afdeling heeft op het moment weinig tijd om aandacht te geven aan het probleem. Daarom is het voor mij als stagiair een uitgelezen mogelijkheid voor een afstudeerproject.

2.2. Opdrachtgever

Hogeschool van Amsterdam

Dhr. J. W. Broos, Sr. Project Manager

Adres: Weesperzijde 190, kamer A2.18
1097 DZ Amsterdam

Tel. 020-5952966
E-mail: j.w.s.broos@hva.nl

2.3. Opdrachtnemer

Yannick Haacke, Student Information Engineering, Hogeschool van Utrecht

Adres: Ringweg Koppel 24D
3813 BB Amersfoort

Tel. 06-18812561
E-mail: y.haacke@hva.nl

2.4. Probleemstelling / Doelstelling

De probleemstelling voor dit project bestaat uit twee delen:

1. De huidige publicatievorm van de roosters is niet geschikt om roosterwijzigingen bruikbaar op te leveren.
2. Instituten hebben geen technische mogelijkheid om roosterwijzigingen door te voeren in het rooster, ze gebruiken ieder eigen systemen voor de instituutsroostering.

Vanuit deze twee problemen zijn mijn doelstellingen voor het project vast te stellen. Deze zijn onder te verdelen in vijf punten:

1. Voer onderzoek uit om erachter te komen hoe instituten / docenten / studenten hun rooster het liefst gepubliceerd zien (via mail, webcalendar, het Roxen-CMS?, etc.).
2. Onderzoek de integratiemogelijkheden van de in gebruik zijnde versie van GP-Untis en onderzoek de mogelijkheden die de nieuwste versie van GP-Untis biedt.
3. Realiseer een toekomstvaste en flexibele oplossing, op basis van de huidige technische mogelijkheden en pakketten.
4. Onderzoek de mogelijkheden voor een systeem waarmee instituten roosterwijzigingen terug kunnen koppelen naar GP-Untis (waar vanuit weer efficiënt gepubliceerd kan worden).
5. Realiseer deze eventuele oplossing.

Na mijn eerste onderzoeksfase bleek echter dat de technische mogelijkheden van het roosterprogramma GP-Untis te beperkt waren om een terugkoppelingssysteem te kunnen realiseren. Na nader overleg en onderzoek is gebleken dat de beste oplossing één systeem zou zijn, waarin zowel de publicatie als de "terugkoppeling" (roosterwijzigingen en publicatie van deze wijzigingen) verzorgd wordt. Mijn definitieve opdrachtomschrijving luidt daarom als volgt:

1. Voer onderzoek uit om erachter te komen hoe instituten / docenten / studenten hun rooster het liefst gepubliceerd zien (via mail, webcalendar, het Roxen CMS, etc.).
2. Realiseer een toekomstvaste en flexibele oplossing op basis van de huidige technische mogelijkheden en pakketten, die voorziet in initiële publicatie van de roosters, het bijhouden van wijzigingen en de publicatie van die wijzigingen.

2.5. Werkwijze

Mijn werkwijze zal ik verduidelijken aan de hand van de hierboven genoemde vijf punten:

1. Het onderzoek zal voor het grootste deel bestaan uit het houden van interviews. Door de mensen te interviewen die de roosters gaan gebruiken kan ik een goed idee krijgen van hoe ze de roosters het liefst zien. Door goed te luisteren naar de wensen van de gebruikers zal de acceptatie van een nieuw systeem veel groter zijn. Aan de hand van de interviews ga ik onderzoeken hoe deze wensen te vertalen zijn naar de technische mogelijkheden van de huidige systemen. Zo zal ik bijvoorbeeld onderzoek doen naar de importmogelijkheden van webcalendar en de exportmogelijkheden van GP-Untis.
2. Naast interviews zal ik ook onderzoek doen naar de mogelijkheden van GP-Untis. Wellicht kan het pakket veel meer dan wat er op dit moment mee gedaan wordt. Door meer functies te gebruiken kan mensen werk uit handen genomen worden. En wellicht bevat de nieuwste versie van GP-Untis wel meer handige functies dan de versie die in gebruik is binnen de Hogeschool

- van Amsterdam. Tijdens dit onderzoek zal ik vooral literatuur over GP-Untis raadplegen. Daarnaast zal ik met de mensen spreken die GP-Untis gebruiken.
3. Tijdens het ontwikkeling van de oplossing zal ik me in eerste instantie richten op een werkend prototype. Als dit er eenmaal is, zal ik dit doorontwikkelen naar een goed ontworpen, gebruikersvriendelijk systeem.

3. Beheersaspecten

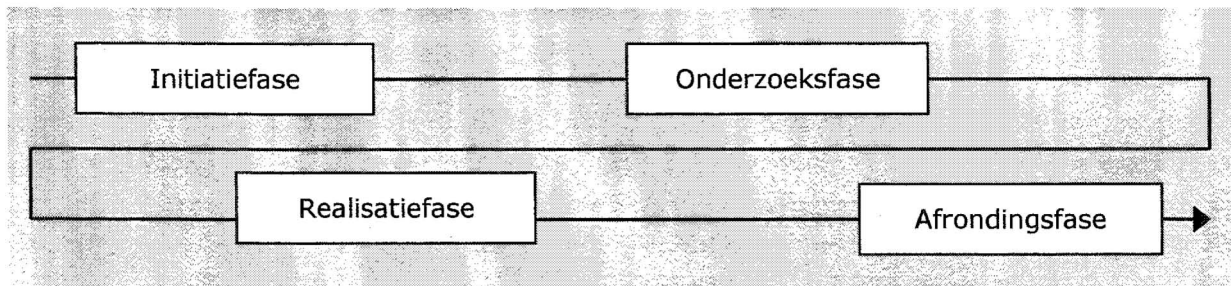
3.1. Scope

3.1.1. Business-eisen

Uiteindelijk resultaat dient een toekomstvaste en flexibele oplossing te zijn voor het roosterprobleem. Dit dient gerealiseerd te worden door middel van gedegen onderzoek. De oplossing dient gerealiseerd te worden met behulp van de huidige technische mogelijkheden en pakketten binnen de Hogeschool van Amsterdam.

3.1.2. Levenscyclus processen

Om het project te doorlopen heb ik gekozen voor een lineaire aanpak. Dit betekent dat fases opeenvolgend doorlopen worden. De fases zijn weergegeven in figuur 1, de pijl van links naar rechts geeft de volgorde weer:

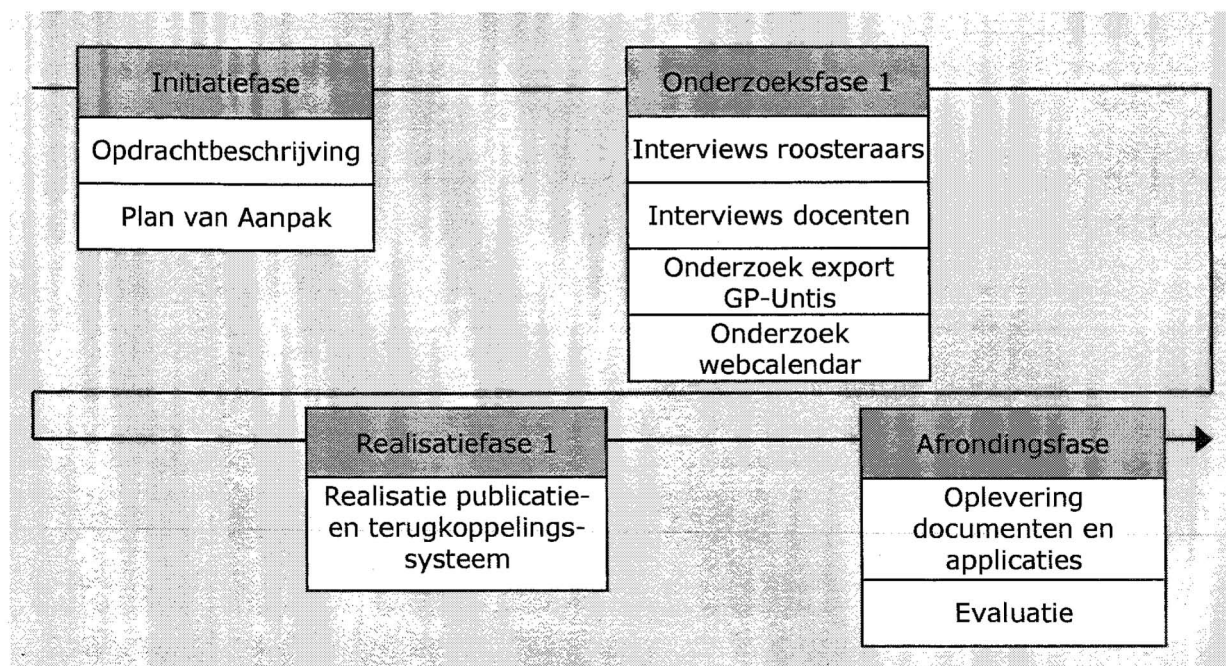


Figuur 1: Lineaire fasering

Omdat niet alles te voorzien is, kan het lineair doorlopen van de fases tijdelijk losgelaten worden. Een dergelijke situatie zou zich voor kunnen doen wanneer in een realisatiefase blijkt dat bepaalde informatie nog ontbreekt. Dan kan besloten worden om een extra interview (uit een van de onderzoeksfases) te doen, terwijl het project zich in de realisatiefase verder ontwikkelt.

Aan het einde van iedere fase ligt een beslismoment. Op zo'n moment wordt de afgelopen fase geëvalueerd en wordt vooruitgeblikt naar de komende fase. Deze beslismomenten worden ook benut om de taken (die in de initiatiefase grofweg zijn omschreven) verfijnd uit te werken en vast te leggen. Op dat moment zal er namelijk *wél* het benodigde inzicht zijn om operationele beslissingen te kunnen nemen.

De praktische invulling van deze fases wordt weergegeven in figuur 2. Hiermee wordt aangegeven hoe de te verwachten levenscyclus van het project eruit ziet. In hoofdstuk 3.1.4. is deze fasering uitgezet in de tijd.



Figuur 2 : Lineaire fasering in levenscyc-usprocessen

3.1.3. Functionaliteit

De uiteindelijke oplossing zal moeten voldoen aan de volgende eisen:

- Publicatie
 - Het individuele rooster voor een instituut zal up-to-date uitleesbaar moeten zijn met 'één druk op de knop'.
 - Het individuele rooster voor een docent zal up-to-date uitleesbaar moeten zijn met 'één druk op de knop'.
 - Het individuele rooster voor een student zal up-to-date uitleesbaar moeten zijn met 'één druk op de knop'.
- Terugkoppeling
 - Een docent zal een roosterwijziging zelf door moeten kunnen voeren in GP-Untis.
 - Een student moet vrije lokalen kunnen reserveren.

3.1.4. Fasering

Initiatiefase – Week 1 t/m 2

Activiteiten	Opdracht beschrijven en afbakenen
Resultaat	Plan van Aanpak

Tabel 1: Fasering – Initiatiefase

Onderzoeksfase – Week 3 t/m 7

Activiteiten	Interviews roosteraars
	Interviews docenten
	Onderzoek GP-Untis export mogelijkheden
	Onderzoek webcalendar import mogelijkheden
	Onderzoek GP-Untis import mogelijkheden
	Onderzoek naar logische opbouw / interface voor terugkoppelsysteem
Resultaat	Interviewverslagen
	Onderzoeksrapport "Roosterportal"

Tabel 2: Fasering – Onderzoeksfase

Realisatiefase – Week 8 t/m 18

Activiteiten	Programmeren koppeling tussen GP-Untis en publicatievorm
	Programmeren terugkoppelsysteem voor roosters
	Testen van oplossing
	Oplossing terugkoppelen naar eindgebruikers
Resultaat	Roosterportal

Tabel 3: Fasering – Realisatiefase

Afrondingsfase – Week 19

Activiteiten	Evaluatie eindproducten
Resultaat	Beoordeling eindproducten

Tabel 4: Fasering – Afrondingsfase

3.2. Tijd

Het project zal lopen van maandag 14 februari 2005 (projectweek 1) tot en met vrijdag 24 juni 2005 (projectweek 19). Er is ruimte voor eventuele uitloop, uiterlijk tot en met vrijdag 8 juli 2005 (projectweek 21).

3.3. Middelen

Voor een goede uitvoering van het project zal ik gebruik gaan maken van een aantal mensen c.q. middelen:

- Projectbegeleider vanuit de Hogeschool van Amsterdam (dhr. J. Broos)
- Afstudeerbegeleider vanuit de Hogeschool van Utrecht (dhr. M. Schnerr)
- Relevante medewerkers aan de Hogeschool van Amsterdam
 - Roosteraars / docenten
- Een werkplek
- Toegang tot een server voor de applicatieontwikkeling

4. Deliverables

De op te leveren producten van dit project zijn:

- Plan van Aanpak
- Onderzoek
 - Onderzoeksrapport "Roosterportal"
- Ontwikkeling
 - Documenten
 - Functioneel ontwerp Publicatiesysteem
 - Technisch ontwerp Publicatiesysteem
 - Systemen
 - Roosterportal

5. Indices

5.1. Figuurindex

Figuur 1: Lineaire fasering	7
Figuur 2: Lineaire fasering in levenscyclusprocessen.....	8

5.2. Tabelindex

Tabel 1: Fasering – Initiatiefase	8
Tabel 2: Fasering – Onderzoeksfase.....	9
Tabel 3: Fasering – Realisatiefase.....	9
Tabel 4: Fasering – Afrondingsfase.....	9

Bijlage 1: Planning

Week	1 (14 feb)	2 (21 feb)	3 (28 feb)	4 (7 mrt)
Interactief Roosteren	Plan van Aanpak	Plan van Aanpak Interviews voorbereiden	Interviews voorbereiden Interviews afnemen	Interviews afnemen Onderzoek GP-Untis Onderzoek webcalendar
HvU				
Overig				

Week	5 (14 mrt)	6 (21 mrt)	7 (28 mrt)	8 (4 apr)
Interactief Roosteren	Interviews afnemen Interviews uitwerken Onderzoek webcalendar	Onderzoek Roosterportal	Onderzoek Roosterportal	Programmeren Testen
HvU	14 maart overleg in Amersfoort			
Overig		21 maart Maarten Schnerr in Amsterdam		

Week	9 (11 apr)	10 (18 apr)	11 (25 apr)	12 (2 mei)
Interactief Roosteren	Programmeren Testen	Programmeren Testen Overleggen met gebruikers	Programmeren Testen	Programmeren Testen Overleggen met gebruikers
HvU				
Overig				

Week	13 (9 mei)	14 (16 mei)	15 (23 mei)	16 (30 mei)
Interactief Roosteren	Programmeren Testen	Programmeren Testen Overleggen met gebruikers	Programmeren Testen Overleggen met gebruikers 25 mei scriptie inleveren	Programmeren Testen Overleggen met gebruikers
HvU				
Overig				

Week	17 (6 jun)	18 (13 jun)	19 (20 jun)	20 (27 jun)
Interactief Roosteren	Programmeren Testen	Programmeren Testen Overleggen met gebruikers Afstudeerzittingen	Evaluatie eindproducten	
HvU			Afstudeerzittingen	
Overig				Evt. uitloopweek

Week	21 (4 jul)
Interactief Roosteren	
HvU	
Overig	Evt. uitloopweek

Bijlage 2: Functioneel Ontwerp

Interactief Roosteren

Functioneel Ontwerp Publicatiesysteem



Versie : 1.1
Datum : 29 maart 2005
Naam : Yannick Haacke

Inhoudsopgave

1. Samenvatting	3
2. Functionele eisen.....	4
3. Interfaceontwerp	5
4. Indices	12
4.1. Figuurindex.....	12
4.2. Tabelindex.....	12

1. Samenvatting

Dit document zal informatie bevatten over de functionele eisen van de te ontwikkelen roosterportal in het kader van het project Interactief Roosteren. Naast de functionele eisen van het systeem die concreet geformuleerd worden in hoofdstuk 2, bevat dit document ook afbeeldingen van de toekomstige user interface.

Deze schematische weergaven van de interface moeten inzicht geven in hoe de functionaliteit van het systeem bereikbaar zal zijn voor de gebruiker en hoe het uiterlijk vormgegeven zal worden. De schermafbeeldingen zijn te vinden in hoofdstuk 3.

Het doel van dit document is op een heldere manier vastleggen welke functionaliteiten het te bouwen systeem gaat bevatten, Dit document dient als basis voor de ontwikkeling en het ontwerp van de roosterportal.

2. Functionele eisen

De functionele eisen per gebruikersrol zijn in tabel 1 hieronder te vinden. De derde kolom in de tabel geeft de prioriteit van de eis aan, gebaseerd op het MoSCoW principe.

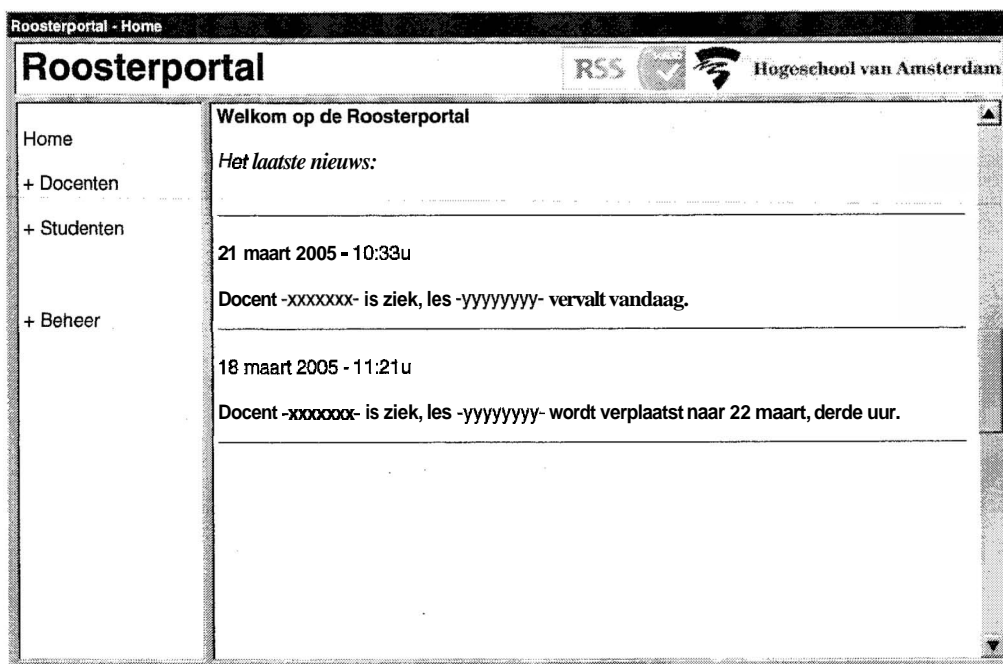
MoSCoW is een eenvoudig maar krachtig prioriteitstellingsprincipe. De eisen aan het systeem worden gecategoriseerd in Must have, Should have, **Could** have, en Want to have but won't have this time. 'M' eisen moeten worden gerealiseerd en vormen de 'Minimal Usable SubseT'. 'S' eisen moeten ook worden gerealiseerd, maar er is eventueel een work-around realiseerbaar. 'C' eisen moeten worden gerealiseerd wanneer er voldoende tijd is, maar zouden eventueel kunnen vervallen zonder de toepasbaarheid van het systeem aan te tasten. 'W' eisen zijn belangrijke eisen die echter niet horen bij de doelstellingen van het huidige increment.

Rol	Eisen	MoSCoW
Beheerder	GP-Untis ResIS export importeren	M
	Per les wijzigingen doorvoeren (nieuw, wijzig, verwijder)	M
	Ziekmelding	M
	Export naar gewenst formaat (webcalendar)	S
Docent	Lessen van zichzelf kunnen zien	M
	Lessen van klassen in kunnen zien	M
	Ziekmelding	S
	Eigen lessen wijzigen (nieuw, wijzig, verwijder)	S
Student	Lessen van de eigen klas inzien	M
	Lessen van andere klassen inzien	S
	Lessen van docenten in kunnen zien	S
Overig	Headline systeem om wijzigingen bekend te maken op de homepage	S
	RSS feed voor roosterwijzigingen	S
	Logfile events bijhouden	C
	Lokalenbeheer (inzicht in vrije lokalen)	W

Tabel 1: Functionele eisen geprioriteerd naar het MoSCoW principe

3. Interfaceontwerp

De onderstaande afbeeldingen geven een globaal overzicht van de gewenste interface van de roosterportal. Op basis van deze interface dient alle functionaliteit op eenvoudige wijze beschikbaar te zijn. Let wel dat deze afbeeldingen een schematische weergave van het toekomstige uiterlijk zijn. Ze zijn bedoeld om een idee te geven van hoe de functionaliteit gescheiden wordt door het menu en welke pagina's uiteindelijk beschikbaar zullen zijn.



Figuur 1: Interface – Roosterportal – Home

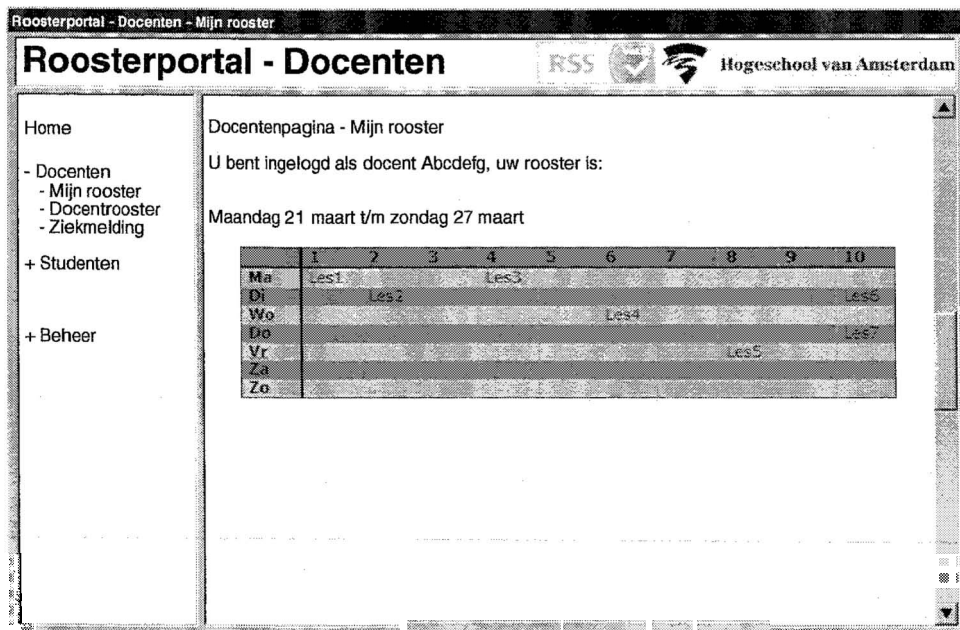
Op deze beginpagina is direct het laatste nieuws te lezen in de vorm van de headlines. Deze headlines bevatten de laatste ziekmeldingen, lokaalwijzigingen, etc.

Links is het menu te vinden. Het menu is opgedeeld in drie categorieën, te weten:

- Docenten
- Studenten
- Beheer

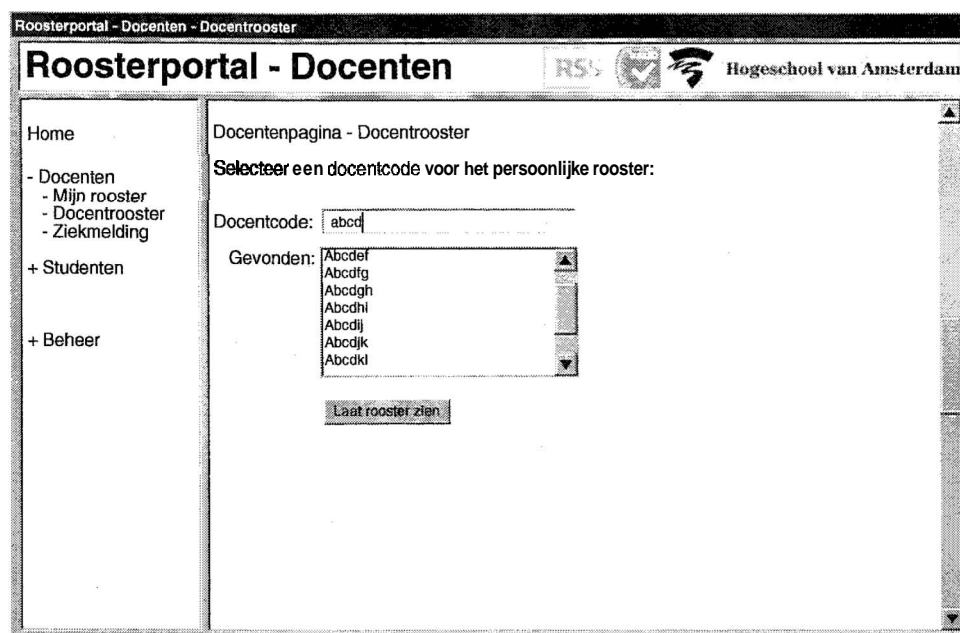
Door deze scheiding kan ieder type gebruiker snel de op hem/haar van toepassing zijnde functies vinden. Bovenaan komt de eventuele link naar de RSS feed¹ die de laatste headlines aan zal bieden.

¹ Kijk voor meer informatie over RSS feeds op:
<http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>



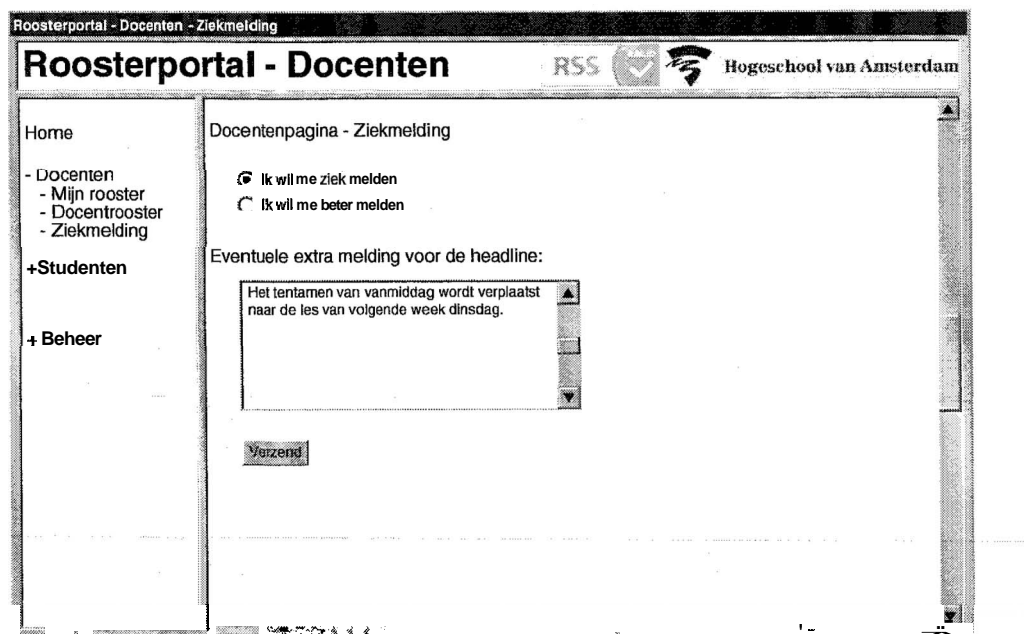
Figuur 2: Interface – Docenten – Mijn rooster

Aan de hand van de ingevoerde gebruikersnaam wordt automatisch het juiste rooster opgehaald uit de database. De docent kan op deze manier snel zijn/haar eigen rooster raadplegen.



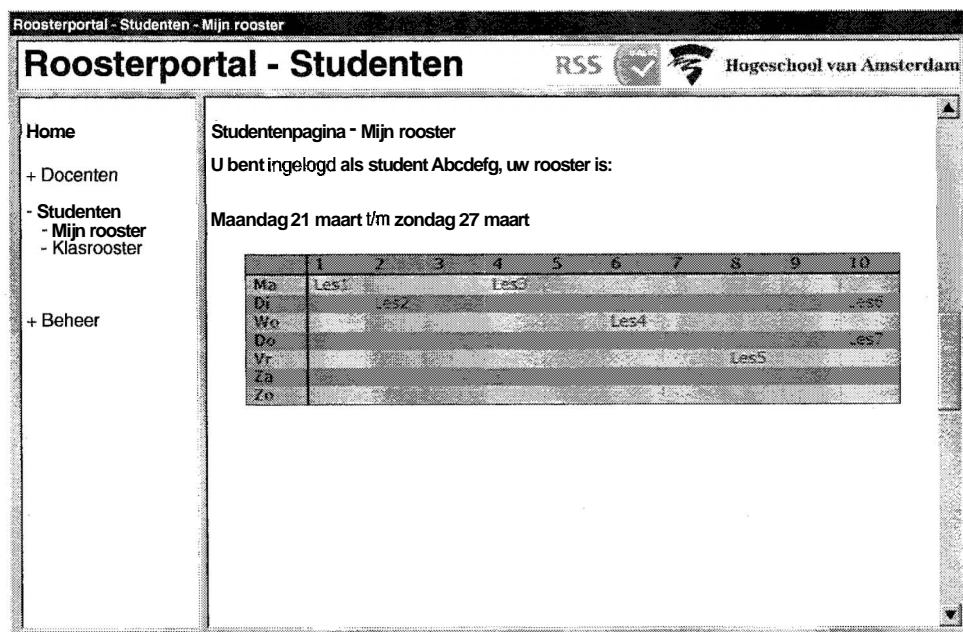
Figuur 3: Interface – Docenten – Docentenrooster

Door te beginnen met het typen van een docentcode worden in de listbox alle mogelijke bestaande docentcodec getoond, gefilterd op de gegeven invoer. Zo is het rooster van een willekeurige docent snel te vinden.



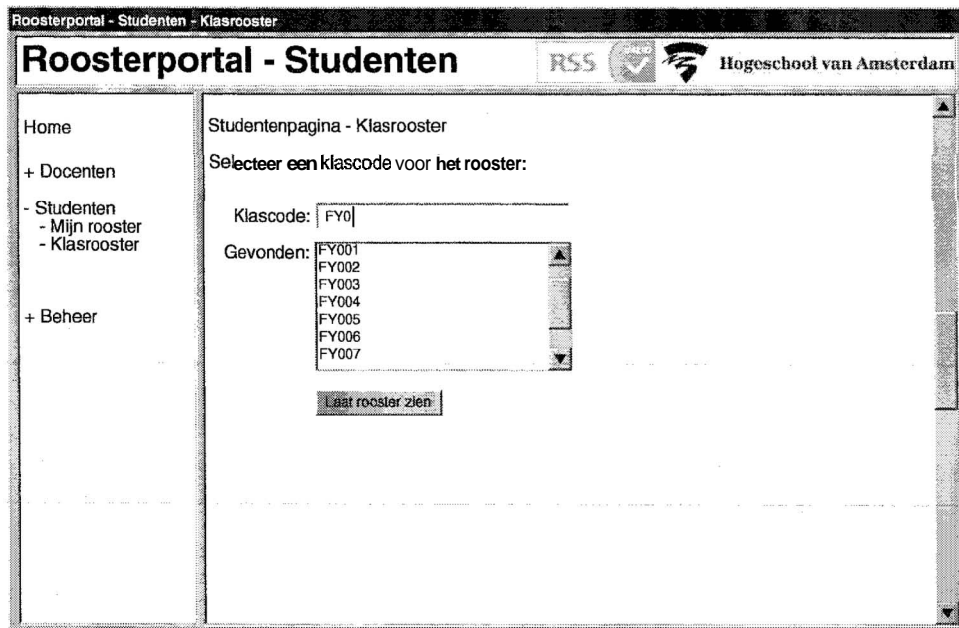
Figuur 4: Interface – Docenten – Ziekmelding

Via dit formulier kan een docent zichzelf ziek en beter melden. Naar aanleiding van de ziekmelding wordt automatisch een headline gecreëerd. De gevolgen voor het rooster (lesuitval) dienen verder onderzocht te worden. Wellicht dient de roosteraar een email te ontvangen over de ziekmelding, waarna de hij/zij de les laat vervallen.



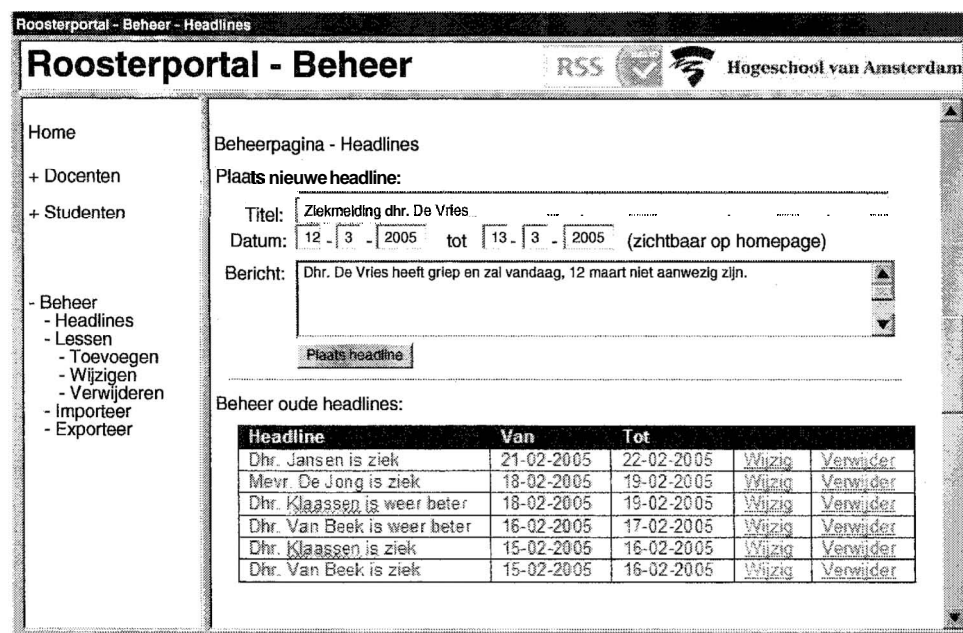
Figuur 5: Interface – Studenten – Mijn rooster

Aan de hand van de ingevoerde gebruikersnaam wordt de klascode bepaald, waarna automatisch het juiste rooster opgehaald wordt uit de database. De student kan op deze manier snel zijn/haar eigen rooster raadplegen.



Figuur 6: Interface – Studenten – Klasrooster

Door te beginnen met het typen van een klascode worden in de listbox alle mogelijke bestaande klascodes getoond, gefilterd op de gegeven invoer. Zo is het rooster van een willekeurige klas snel te vinden.





Figuur 7: Interface – Beheer – Headlines

Via dit formulier kan een beheerder zelf headlines toevoegen, wijzigen en verwijderen. Doordat de headlines een begin- en einddatum hebben, verdwijnen ze automatisch van de homepage, ze kunnen niet vergeten worden.

Roosterportal - Beheer - Lessen toevoegen

Roosterportal - Beheer

RSS   Hogeschool van Amsterdam

Home

+ Docenten

+ Studenten

- Beheer
 - Headlines
 - Lessen
 - Toevoegen
 - Wijzigen
 - Verwijderen
 - Importeer
 - Exporteer

Beheerpagina - Lessen toevoegen

Docentcode:

Datum: - -

HvA Lesuur:

Vakcode:

Lokaal:



Klascode:

Figuur 8: Interface – Beheer – Lessen toevoegen

Hier kan een beheerder handmatig een les toevoegen. Deze les zal automatisch opgenomen worden in de betreffende roosters van docenten en studenten.

Roosterportal - Beheer - Lessen wijzigen

Roosterportal - Beheer

RSS   Hogeschool van Amsterdam

Home

+ Docenten

+ Studenten

- Beheer
 - Headlines
 - Lessen
 - Toevoegen
 - Wijzigen
 - Verwijderen
 - Importeer
 - Exporteer

Beheerpagina - Lessen wijzigen

Docentcode:

Datum: - -

HvA Lesuur:

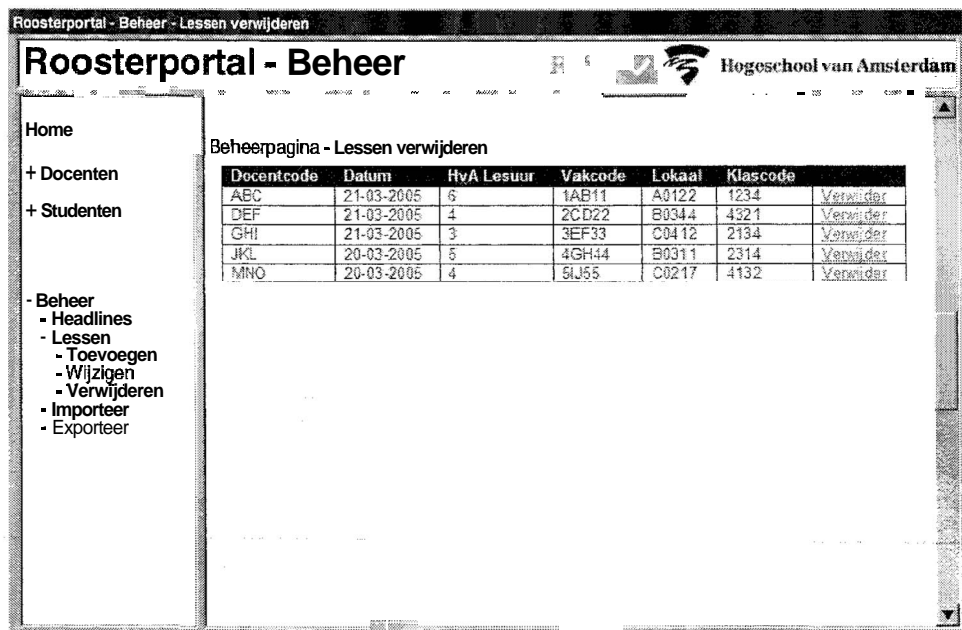
Vakcode:

Lokaal:

Klascode:

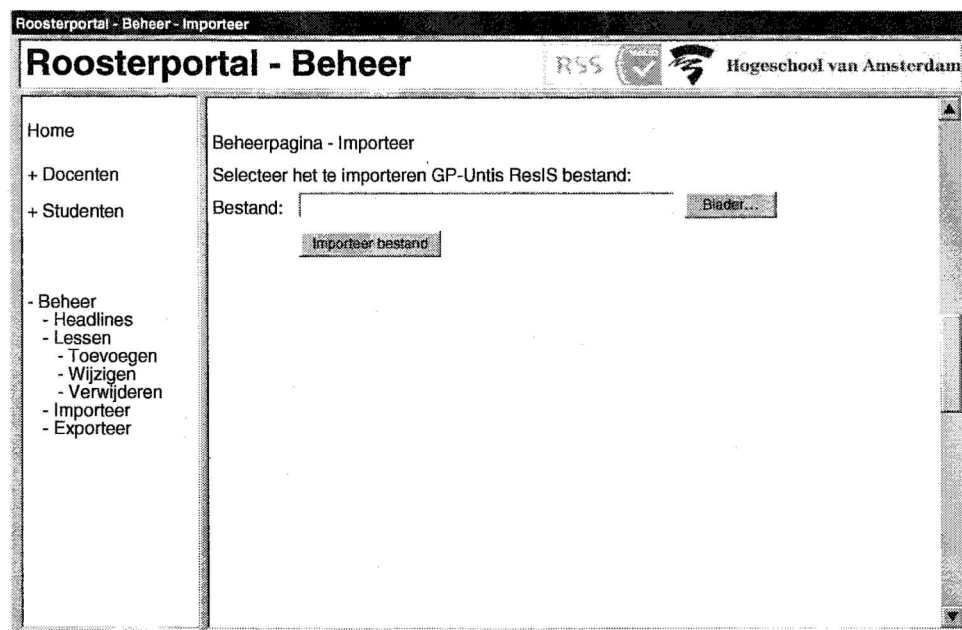
Figuur 9: Intetface – Beheer – Lessen wijzigen

Bestaande lessen kunnen gewijzigd (verplaatst) worden via deze pagina. Voorafgaand aan dit venster is er een zoekformulier om de juiste les te kunnen vinden.



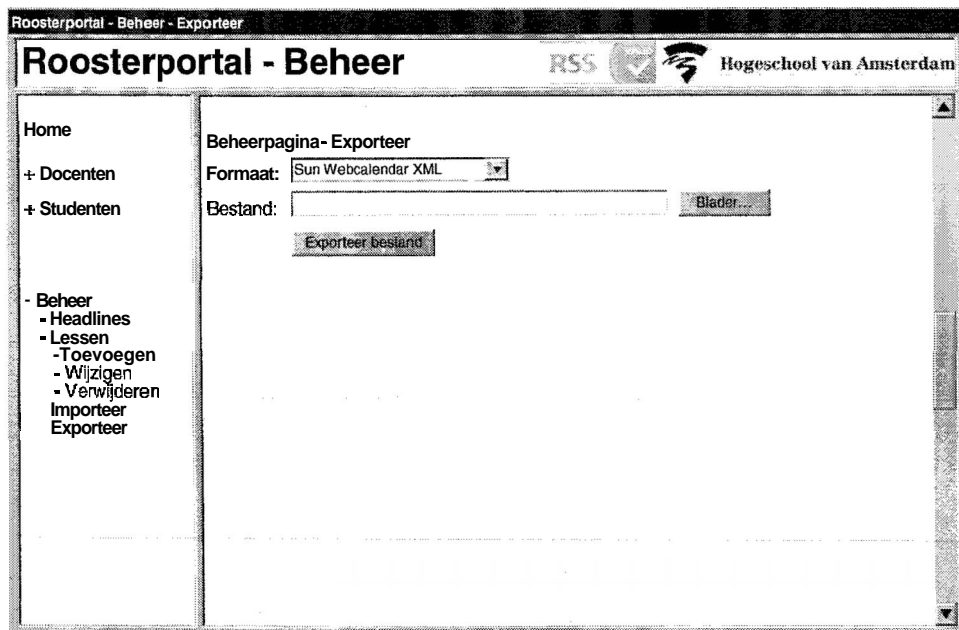
Figuur 10: Interface – Beheer – Lessen verwijderen

Via dit overzicht kunnen lessen verwijderd (vervallen) worden. Deze actie kan niet ongedaan gemaakt worden. Wel kan er een nieuwe les toegevoegd worden.



Figuur 11: Interface – Beheer – Importeer

Via deze pagina kan een beheerder de ResIS export van GP-Untis importeren in het systeem. Geïmporteerde gegevens worden direct zichtbaar via de portal.



Figuur 12: Interface – Beheer – Exporteer

Ten slotte kan een beheerder via deze pagina de gegevens uit het systeem exporteren naar een geschikt formaat om bijvoorbeeld te importeren in de Sun Webcalendar.

4. Indices

4.1. Figuurindex

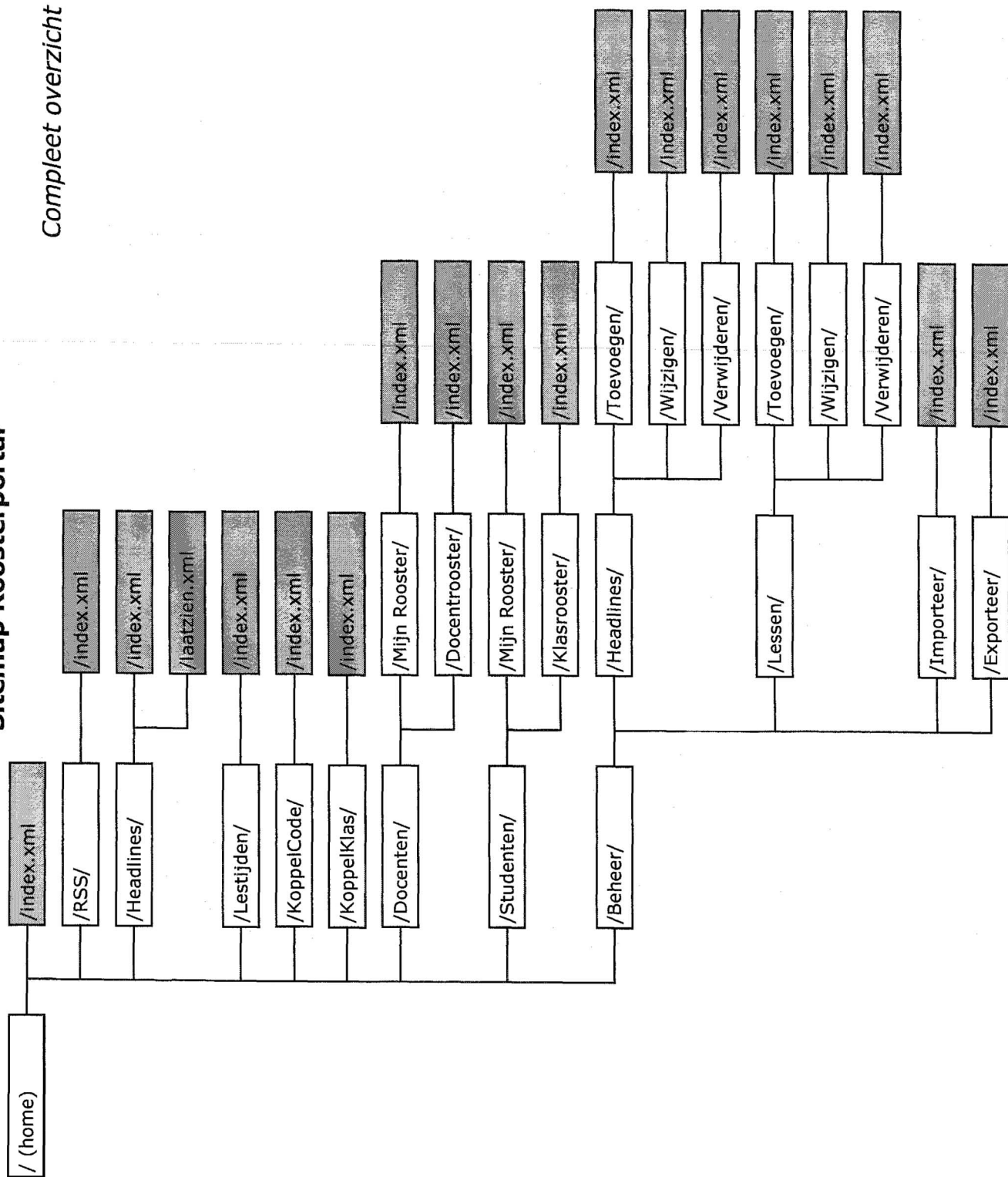
Figuur 1: Interface _ Roosterportal _ Home	5
Figuur 2: Interface _ Docenten _ Mijn rooster	6
Figuur 3: Interface _ Docenten _ Docentenrooster	6
Figuur 4: Interface - Docenten - Ziekmelding	7
Figuur 5: Interface - Studenten - Mijn rooster	7
Figuur 6: Interface - Studenten - Klasrooster	8
Figuur 7: Interface - Beheer - Headlines.....	8
Figuur 8: Interface - Beheer - Lessen toevoegen.....	9
Figuur 9: Interface - Beheer - Lessen wijzigen.....	9
Figuur 10: Interface - Beheer - Lessen verwijderen	10
Figuur 11: Interface - Beheer - Importeer.....	10
Figuur 12: Interface - Beheer - Exporteer	11

4.2. Tabelindex

Tabel 1: Functionele eisen geprioriteerd naar het MoSCoW principe	4
---	---

Bijlage 3: Sitemap

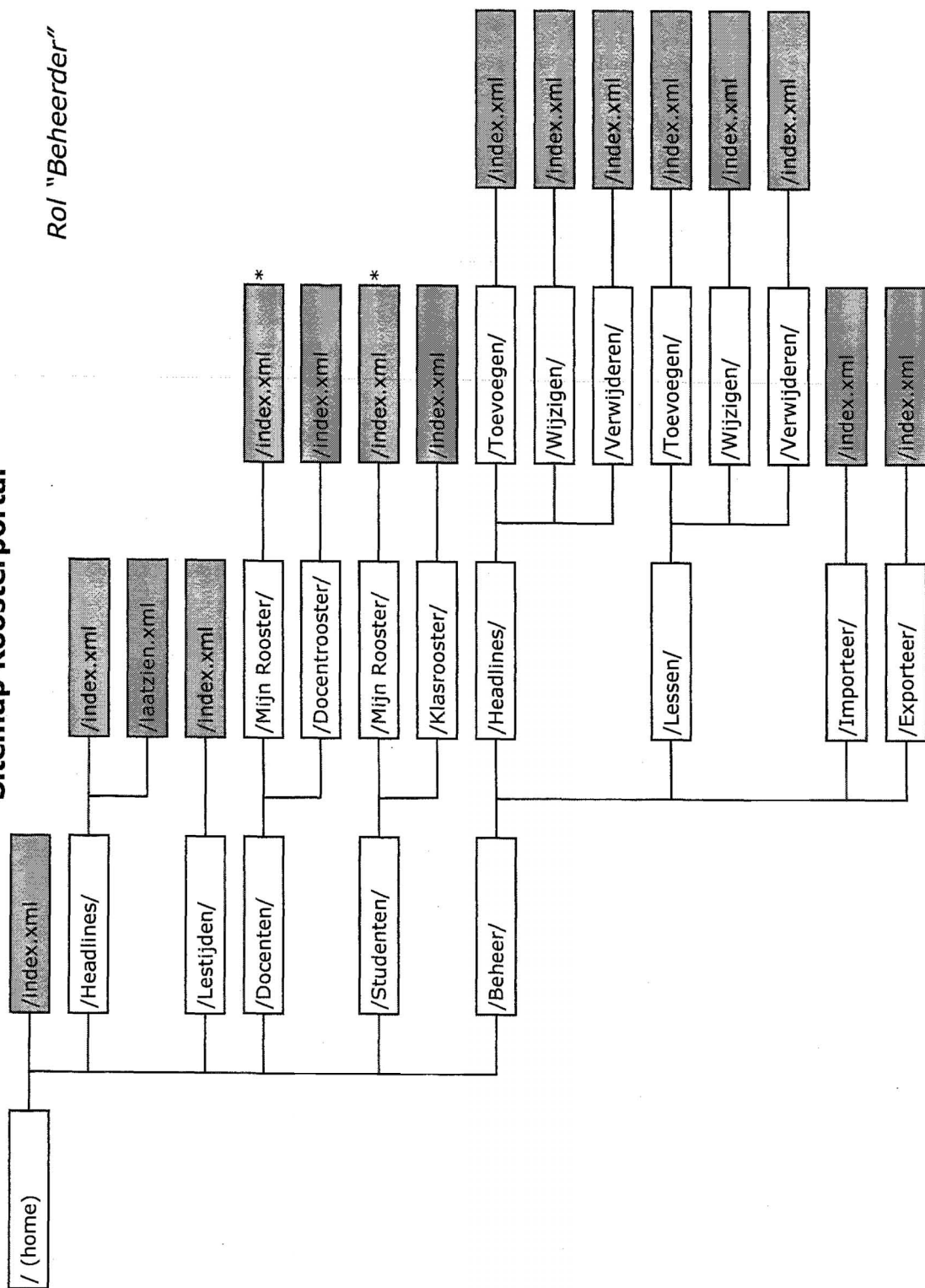
Sitemap Roosterportal



Compleet overzicht

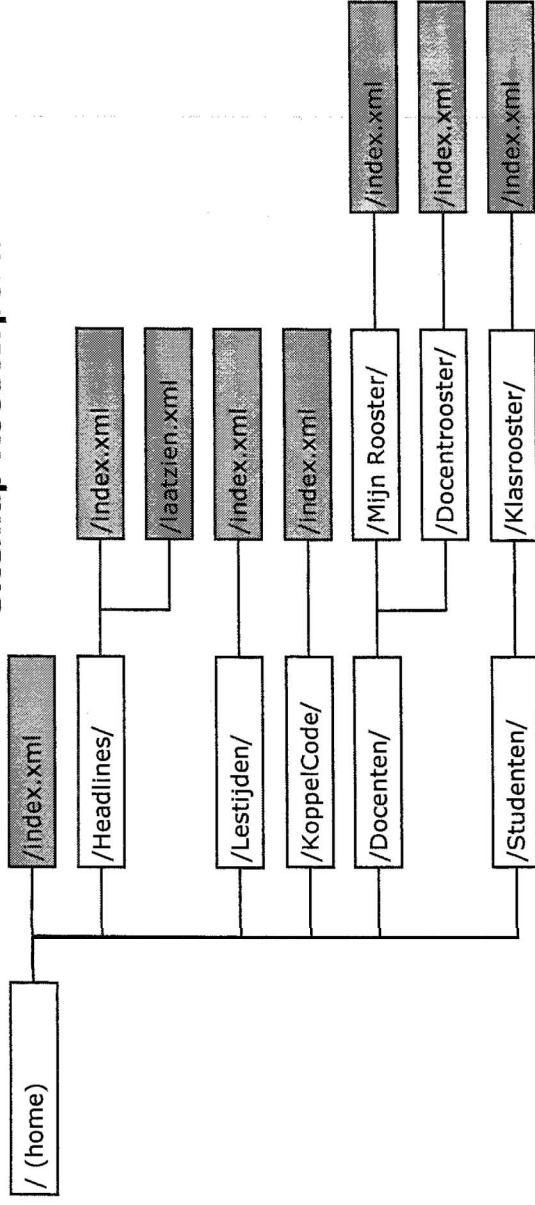
Sitemap Roosterportal

Rol "Beheerder"



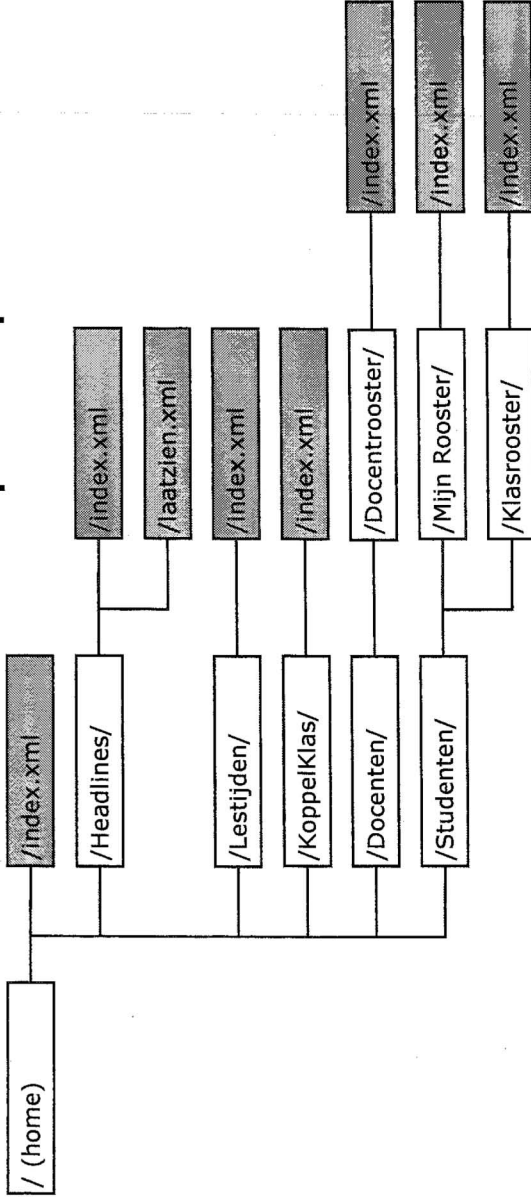
* Wel zichtbaar, geen functionaliteit

Sitemap Roosterportal



Rol "Docent"

Sitemap Roosterportal



Rol "Student"

Bijlage 4: Database-indeling

Database-indeling Roosterportal

lessen	
SQL: CREATE TABLE lessen(id DOUBLE AUTO_INCREMENT PRIMARY KEY, docent VARCHAR(10), datum DATE, week INTEGER, uur INTEGER, vak VARCHAR(25), klas VARCHAR(10), lokaal VARCHAR(6), gewijzigd DATETIME, import VARCHAR(250))	
Veld	Type
id	DOUBLE, AUTO_INCREMENT, PRIMARY KEY
docent	VARCHAR (10)
datum	DATE
week	INTEGER
uur	INTEGER
vak	VARCHAR (25)
klas	VARCHAR (10)
lokaal	VARCHAR (6)
gewijzigd	DATETIME
import	VARCHAR (250)

headlines	
SQL: CREATE TABLE lessen(id DOUBLE AUTO_INCREMENT PRIMARY KEY, titel VARCHAR(200), datumvan DATE, datumtot DATE, bericht TEXT, gewijzigd DATETIME)	
Veld	Type
id	DOUBLE, AUTO_INCREMENT, PRIMARY KEY
titel	VARCHAR (200)
datumvan	DATE
datumtot	DATE
bericht	TEXT
gewijzigd	DATETIME

headlinerechten	
SQL: CREATE TABLE headlinerechten(id DOUBLE, type VARCHAR(1), user VARCHAR(100))	
Veld	Type
id	DOUBLE
type	VARCHAR (1)
user	VARCHAR (100)

studentklas	
SQL: CREATE TABLE studentklas(student VARCHAR(50) PRIMARY KEY, klas VARCHAR(50))	
Veld	Type
student	VARCHAR (50), PRIMARY KEY
klas	VARCHAR (50)

docentcode	
SQL: CREATE TABLE docentcode(docent VARCHAR(50) PRIMARY KEY, code VARCHAR(50))	
Veld	Type
docent	VARCHAR (50), PRIMARY KEY
code	VARCHAR (50)

