

AFSTUDEERVERSLAG

AEOLUS AGENDA



Interactieve agenda voor het
projectmanagementsysteem Aeolus

Student	Reyo Alwar Stallenberg
Studentnummer	20024731
Opdrachtgever	Connectholland, Rotterdam
Opleiding	Informatica en Informatiekunde
Afstudeerrichting	Vormgeving en ontwerp van interactie
Afstudeerperiode	2006-2.2 – 2007-1.1
Examinator	H. Mastenbroek
Examinator	J.J. van Beijnum
Bedrijfsmentor	J. Soeterbroek
Plaats en datum	Rotterdam, maart 2007

Referaat

Stallenberg, R.A., Eindverslag, Aeolus Agenda, Rotterdam 2006.

Dit verslag is een verantwoording van een 5 maanden durende afstudeerperiode van een student aan de Haagse Hogeschool, sector Informatica en Informatiekunde, opleiding Vormgeving en Ontwerp van Interactie (VIA).

De opdracht bestond uit het maken van een interactieve online agenda voor het projectmanagementsysteem Aeolus van het bedrijf Connectholland. Tevens moest deze agenda als losstaande module binnen het framework Windmill van Connectholland kunnen opereren.

In dit verslag worden de verrichte werkzaamheden beschreven en de gekozen aanpak die ten grondslag lag aan de opdracht. Ook worden de ontstane problemen beschreven en de gekozen oplossingen gemotiveerd.

Descriptoren

- Connectholland
- Webdesign
- Webmarketing
- PHP 5
- XSL
- XML
- HTML
- Javascript
- AJAX
- Projectmanagement
- Modulaire systemen



Everything should be made as simple as possible,
but not simpler.

Albert Einstein



Voorwoord

Alles zou zo eenvoudig mogelijk gemaakt moeten worden, maar niet eenvoudiger. Dit moet naar mijn idee het motto zijn achter elk software ontwikkeltraject. Want daarin worden vaak ingewikkelde oplossingen bedacht voor eenvoudig op te lossen problemen. Er wordt vaak vanuit computerlogica beredeneerd en ontwikkeld, in plaats vanuit de logica van de wereld om ons heen. Daarom heb ik deze opdracht als een uitdaging gezien om van deze gewoonte af te wijken. Maar ook als een uitdaging om mijn kennis en vaardigheden te vergroten. Naar mijn idee is dit project daarin geslaagd. Mede door de geweldige ondersteuning van mijn collega's.

Daarbij wil ik in het bijzonder mijn bedrijfsmentoren J. Soeterbroek en R. Rademaker bedanken voor hun steun in het doorlopen van dit project. Ook gaat mijn dank uit voor de hulp die ik heb gekregen bij het tot stand komen van dit verslag van mijn examinatoren op de Haagse Hogeschool, ing J.J. van Beijnum en H. Mastenbroek. Tot slot wil ik mijn tweelingbroer G. Stallenberg bedanken voor zijn steun op alle fronten.

Ik hoop dat u dit verslag met plezier leest en dat het een goed inzicht geeft in het door mij uitgevoerde project en het proces dat ik daarin heb doorlopen.

Reyo Stallenberg,
Rotterdam, maart 2007



Inhoud

1. Inleiding	7
1.1 Introductie.....	7
1.2 Beschrijving inhoud	7
2. Organisatie	9
2.1 Historie.....	9
2.2 Organisatiestructuur	10
2.3 Projectinrichting.....	11
3. Opdracht	12
3.1 Inleiding.....	12
3.2 Opdrachtgever	12
3.3 Uitvoerende	12
3.4 Probleemstelling.....	13
3.5 Uitgangspunten en randvoorwaarden	13
3.5.1 Benodigde Software	13
3.5.2 Benodigde hardware.....	13
3.5.3 Benodigde documentatie	13
3.6 Doelstellingen van de opdracht	13
3.7 Na te streven ideeën.....	14
3.8 Werkzaamheden	14
3.9 Methoden, technieken en nadrukken	15
3.9.1 Methoden.....	15
3.9.2 Technieken.....	15
3.9.3 Nadrukken.....	15
3.10 Resultaten	16
4. Oriëntatie en planning	17
4.1 Inleiding.....	17
4.2 Oriëntatie	17
4.2.1 Beginsituatie.....	18
4.2.2 Definitiestudie.....	19
4.2.3 Fasering van het project	21
4.3 Projectplan en pilots.....	22
4.3.1 Inleiding	22
4.3.2 Pilots	22
4.3.3 Doelstellingen van de op te leveren producten	23
4.4 Planning	24
4.5 Analyse	25
4.5.1 Strategie.....	25
4.5.2 Omvang	25
4.6 Ontwerp.....	26
4.6.1 Structuur	26
4.6.2 Skelet	28
4.6.3 Voorkant	29



5. Proces en uitvoering	30
5.1 Inleiding	30
5.2 Analysefase	30
5.2.1 Beginsituatie	30
5.2.2 Probleem 1: Structuur Aeolus	32
5.2.3 Probleem 2: Verwerken Database Gegevens	34
5.2.4 Inventarisatie van mogelijkheden	36
5.3 Ontwerpfase	37
5.4 Ontwikkelfase	40
5.4.1 Project, ticket en taak structuur	40
5.4.2 JSON	41
5.4.3 Frontend	44
6. Evaluatie	47
6.1 Inleiding	47
6.2 Procesevaluatie	47
6.3 Productevaluatie	48
7. Bijlagen	50
7.1 Geraadpleegde literatuur	50
7.2 Afkortingenlijst	51
7.3 Relevante internetadressen	54
7.4 Concept grafisch ontwerp agenda	55
7.5 Database ontwerp nieuwe structuur	56
7.6 Definitiestudie	57
7.7 Pilotplan	72



1. Inleiding

1.1 Introductie

Ter afsluiting van mijn studie Vormgeving en ontwerp van interactie bij de sector Informatica & Informatiekunde aan de Haagse Hogeschool heb ik een afstudeerstage gelopen bij Connectholland, een bureau voor interactieve marketing en e-business, in Rotterdam.

Doel van het afstuderen

Het leerdoel staat in het leerplan als volgt beschreven: De student kan, met behulp van de vakliteratuur zelfstandig (een deel van) een automatiseringsproject uitvoeren, dat aansluit bij de leerstof die na de stage is behandeld.

Doel van het verslag

Dit verslag moet duidelijk maken welk traject tijdens dit halfjaar is gevolgd om tot het eindproduct te komen. Uit dit verslag zal blijken welke werkwijze gevolgd is. Ook zal er duidelijk worden welke beslissingen er op bepaalde momenten genomen zijn en wat hier de reden voor was. Aan de hand van dit verslag zullen de examinatoren en gecommitteerde de uitvoering van de opdracht beoordelen.

1.2 Beschrijving inhoud

Hoofdstuk 2: Organisatie

Om een goed beeld te geven van het kader waarbinnen het afstudeerproject heeft plaatsgevonden, wordt het bedrijf waarbinnen de afstudeeropdracht is uitgevoerd beschreven. Hierbij komen de historie, de organisatiestructuur en de inrichting van de door het bedrijf uitgevoerde projecten aan bod.

Hoofdstuk 3: Opdracht

Aan het begin van de afstudeerperiode is een omschrijving vastgesteld van de opdracht die door de afstudeerder uitgevoerd zou worden. In dit hoofdstuk is deze omschrijving opgenomen. Hierin wordt slechts de basis van waaruit de opdracht is aangevangen en uitgevoerd besproken. Resultaten komen hier nog niet aan bod. Op deze manier wordt voor de rest van dit verslag duidelijk wat de grondslag is van keuzes die zijn gemaakt en resultaten die zijn geboekt.



Hoofdstuk 4: Oriëntatie en planning

In dit hoofdstuk wordt beschreven hoe de afstudeerder zich heeft georiënteerd op de opdracht. De planning van de uit te voeren werkzaamheden en de doelstellingen van de op te leveren producten worden besproken. Om aan te geven welke keuzes er zijn gemaakt tijdens de oriëntatie en waarom deze keuzes zijn gemaakt wordt er ook een gedetailleerde beschrijving van de situatie aan het begin van het afstuderen gegeven. De producten die zijn vervaardigd in de oriëntatiefase worden verantwoord.

Hoofdstuk 5: Proces en uitvoering

In dit hoofdstuk wordt beschreven hoe de uitvoering van de opdracht is verlopen en welke oplossingsmethoden zijn toegepast. Er wordt weergegeven wat de werkzaamheden inhielden en er wordt gemotiveerd welke keuzes zijn gemaakt gedurende de uitvoering van de opdracht. Met de nadruk op hoe en waarom deze keuzes zijn gemaakt. De gekozen aanpak en de producten worden verantwoord.

Hoofdstuk 6: Evaluatie

In dit hoofdstuk worden het proces en de producten geëvalueerd. Er wordt gereflecteerd op de procesgang en op de opgeleverde producten.

De waarde van het proces dat doorlopen is en de producten die zijn gemaakt wordt geschat.



2. Organisatie

2.1 Historie

Connectholland is een bureau voor interactieve marketing en e-business

Zij staan garant voor creatieve en doeltreffende internetconcepten en technische internettoepassingen. Hun heldere en gestructureerde werkwijze zorgen ervoor dat hun werk binnen tijd en budget wordt opgeleverd.

Connectholland maakt, implementeert en verkoopt internetoplossingen van hoge kwaliteit. Hun ervaring en kennis zorgen voor de overbrugging van het gat tussen business, marketing en technologie. Connectholland heeft enthousiaste medewerkers die andersdenkend, intelligent, nuchter, gepassioneerd en altijd online zijn. Kwaliteit, klantgerichtheid, service en innovativiteit zijn de door het bedrijf zelf geformuleerde kernwaarden.

Dankzij deze werkwijze rekenen zij onder andere Artihove Art Centre, Shipmate Vlaggen, Volker Wessels Stevin, GGD Rotterdam, OBR-Rotterdam, Havenbedrijf Rotterdam, Ministerie van Verkeer en Waterstaat, Gemeente Rotterdam en Syntens tot hun vaste opdrachtgevers.

Het jonge bedrijf Connectholland is opgericht in 2000 in Rotterdam en heeft daar een vestiging.



2.2 Organisatiestructuur

Connectholland is een typische lijnorganisatie. Een lijnorganisatie is de meest eenvoudige organisatiestructuur, waarbij iedere medewerker één directe baas heeft, op de hoogste man na. Het aantal lagen tussen de werkvloer en de directeur heet de “depth of control”. De “depth of control” bij Connectholland is zeer gering, wat er voor zorgt dat er een informele sfeer heerst.

Connectholland heeft vier onderdelen, te weten:

Management

Naast het adviseren van klanten over nieuwe websites en het vernieuwen van oude websites doet het management het personeelsbeleid en de administratie. Een van de belangrijkste taken van het management is de projectbeheersing.

Vormgeving

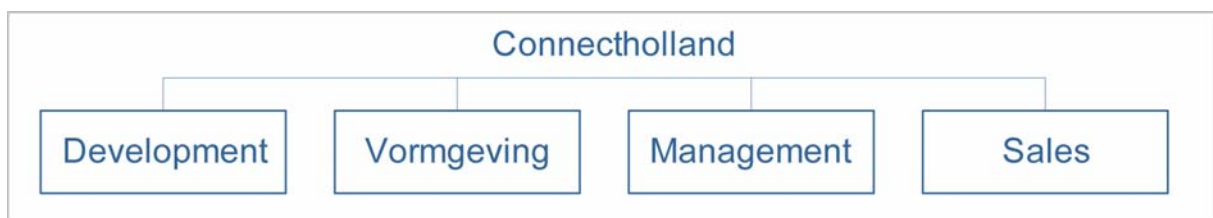
Het vormgevingsteam ontwerpt het uiterlijk van de websites die Connectholland maakt. Dit is het deel van de website dat de bezoeker daadwerkelijk ziet. En doet dus het interaction-design en navigation-design.

Development

Het developmentteam ontwerpt de techniek achter een website en werkt dit uit. Omdat developers zelf ontzettend veel bezig zijn met software en daarbij veel kennis opdoen doen zij ook een soort intuïtief interaction-design. Zij beoordelen vaak op basis van hun kennis en expertise of het interaction-design volledig is of niet en werken dat indien nodig bij.

Sales

Het salesteam zorgt dat er voldoende werk is om Connectholland draaiende te houden. Zij halen nieuwe opdrachten binnen en blazen oude projecten nieuw leven in door contacten aan te halen.



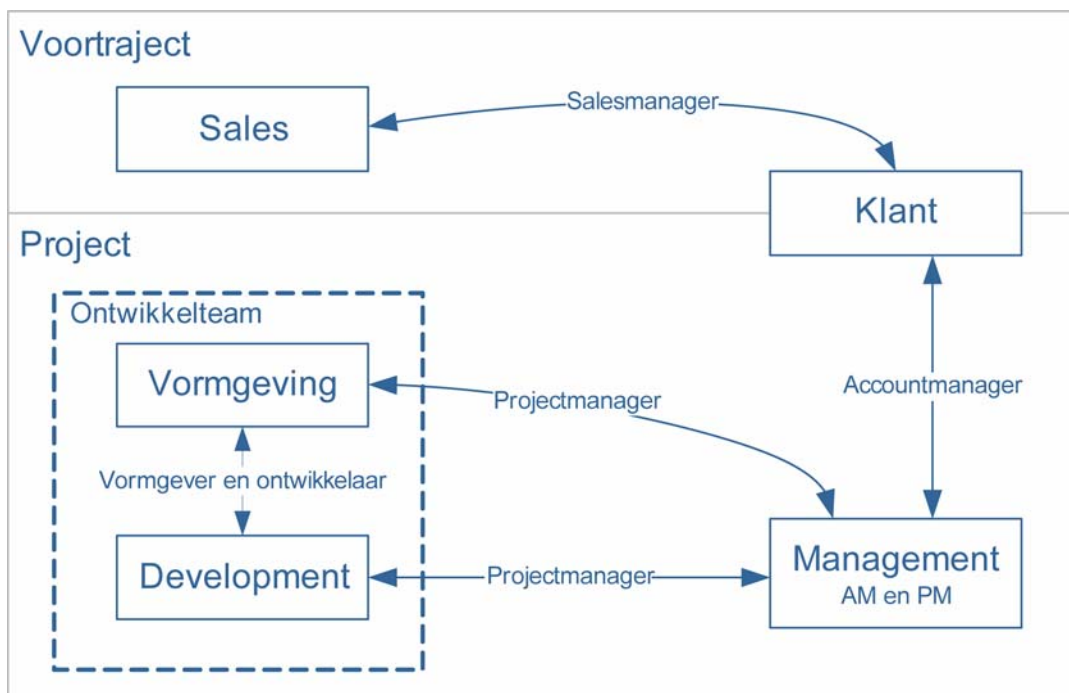
Figuur 1: Organigram Connectholland



2.3 Projectinrichting

Connectholland werkt op basis van projecten. Binnen die projecten zijn er verschillende rollen, weergegeven in Figuur 2. Een klant die een project bij Connectholland heeft lopen communiceert met de accountmanager over de gang van zaken binnen het project. Dat zijn dus de kosten, de plannen, de voortgang, het resultaat en dergelijke. Vaak is de accountmanager ook de projectmanager, maar deze rol kan ook vervuld worden door een vormgever of ontwikkelaar. De projectmanager is verantwoordelijk voor het verloop en de voortgang van het project en communiceert daarover met het ontwikkelteam en indien van toepassing met de accountmanager. In samenspraak worden er plannen gemaakt en uitgevoerd. Tevens houdt de projectmanager toezicht op het verloop van het project. Een accountmanager is de directe contactpersoon van een klant (de beheerder van zijn account). De meeste communicatie tussen de klant en Connectholland verloopt dan ook via de accountmanager, maar soms ook via de projectmanager of direct met Vormgeving of Development.

Alleen in uitzonderlijke gevallen wordt deze taak enkel door de accountmanager vervuld omwille van duidelijkheid tegenover de klant.



Figuur 2: Projectinrichting Connectholland



3. Opdracht

3.1 Inleiding

Connectholland heeft een eigenhandig ontwikkeld projectmanagementsysteem. In dit systeem kan het verloop van een project worden vastgelegd en worden bijgehouden om op die manier meer controle te verkrijgen over het bedrijfsproces. Daarmee kan dan ook een reële weergave worden verkregen van de voor het project benodigde resources. Het originele systeem werkt in combinatie met Google Calendar (calendar.google.com) voor tijdsplanning en -registratie en met het open-source Customer Relationship Management pakket sugarCRM (www.sugarcrm.com) om klantgegevens in op te slaan. In dit projectmanagementsysteem werden projecten onderverdeeld in tickets en eventueel subtickets. Deze tickets bevatten informatie over wat er gedaan moest worden. Dit zal dan worden uitgevoerd door de betreffende medewerkers.

Aanleiding tot deze opdracht was dat Connectholland graag het eigen projectmanagementsysteem wil ontkoppelen van externe systemen, zodat het een transporteerbaar zelfstandig systeem werd. Tevens wilde Connectholland binnen hun framework¹ Windmill een agenda module kunnen aanbieden.

3.2 Opdrachtgever

De opdrachtgever van het project Aeolus Agenda is Connectholland bv.

Naam:	J. Soeterbroek, Management
Adres:	Westerstraat 42 3016 DH Rotterdam
E-mail:	j.soeterbroek@connectholland.nl
Telefoonnummer:	010 436 50 50
Faxnummer:	010 436 90 80

3.3 Uitvoerende

De uitvoerende van het project Aeolus Agenda is Reyo Stallenberg.

Naam:	Reyo Alwar Stallenberg
Adres:	Vinkenstraat 273 3036XP Rotterdam
E-mail:	reyostallenberg@gmail.com
Telefoonnummer:	06 17 18 64 29
School adres:	Johanna Westerdijkplein 75 2521 EN Den Haag
Telefoonnummer:	070 445 88 88

¹ Een framework is een geheel van softwarecomponenten dat gebruikt kan worden bij het programmeren van toepassingen. Bij Connectholland is een framework ontwikkeld in PHP met de naam Windmill. Met de componenten uit dit framework wordt een website opgebouwd.



3.4 Probleemstelling

Hoe kan het projectmanagementsysteem Aeolus losgekoppeld worden van een extern kalendersysteem?

3.5 Uitgangspunten en randvoorwaarden

3.5.1 Benodigde Software

De volgende software is gebruikt:

- Quanta+
- Subversion versie beheer
- SugarCRM
- Mozilla Firefox
- Microsoft Internet Explorer
- PHP
- MySQL
- Propel
- Openoffice.org
- Apache webserver
- Windmill

3.5.2 Benodigde hardware

De volgende hardware is gebruikt:

- PC
- Randapparatuur, zoals muis en toetsenbord
- Server

3.5.3 Benodigde documentatie

De volgende basisdocumentatie stond tot beschikking van de uitvoerende:

- Briefingdocument
- Conceptvormgeving
- Handleiding Windmill

3.6 Doelstellingen van de opdracht

Aeolus loskoppelen van een externe applicatie en daarmee een zelfstandig product creëren. Een neven doelstelling was een losstaande agenda hebben binnen het framework Windmill van Connectholland.



3.7 Na te streven ideeën

De na te streven ideeën voor dit project waren:

- Drag-and-drop² agenda zoals in bestaande agendabeheer systemen
Zoals Microsoft Outlook, Mozilla Sunbird en Google Calendar.
- Een module die gegevens kan weergeven volgens Calendar v2.0 specificaties
- Het nastreven van standaards zoals vastgelegd door onder andere W3C en ISO, maar wel met het oog op marktaandeel van bijvoorbeeld internet browsers
- DRY³ code schrijven
- Uitbreidbare code schrijven

3.8 Werkzaamheden

De concrete werkzaamheden binnen dit project omvatten de in deze paragraaf beschreven punten. In het kader van de afstudeeropdracht zijn de volgende activiteiten verricht:

- Opstellen van een plan van aanpak
- Definiëren systeemeisen
- Opstellen definitiestudie
- Ontwerp maken van de Graphical User Interface
- Opstellen van pilotplan en pilotontwikkelplan
- Bouwen applicatie
- Testen applicatie
- Implementatie applicatie

² Drag-and-drop is een bewerking die inhoudt dat er met de muis een onderdeel op het scherm wordt vastgepakt en naar een andere plaats wordt gesleept.

³ Zie bijlage 7.2: Afkortingenlijst



3.9 Methoden, technieken en nadrukken

In deze paragraaf zijn de methoden, technieken en nadrukken behorende bij de opdracht opgenomen. Hierbij gaat het voornamelijk om de analyse methoden en de afbeeldingstechnieken die zijn gebruikt in het voortraject van de opdrachttuitvoering.

3.9.1 Methoden

Er is tijdens het afstuderen voornamelijk gebruik gemaakt van informatie uit het boek “The Elements of User Experience”. Daarin staat duidelijk beschreven hoe je een software ontwikkelproject goed in kan richten.

In de originele opdrachtschrijving stond dat er onder andere gebruik gemaakt zou worden van IAD (Iterative Application Design), een projectbeheersingsmethodiek voor het ontwikkelen van software. Omdat IAD vooral bedoeld is als projectbeheersingsmethodiek voor grote projecten is er niet of nauwelijks gebruik gemaakt van IAD. Met name omdat deze methodiek niet binnen het bedrijf gebruikt wordt waardoor de bijbehorende documentatie als relatief onnuttig kan worden beschouwd voor gebruik in doorontwikkeling. Er zijn wel een aantal zaken aan de hand van IAD gedaan, zoals het doen van een definitiestudie en het kiezen van pilots voor het project.

3.9.2 Technieken

De volgende technieken zijn gebruikt:

- UML
- Interviewtechnieken
- Technieken behorend bij IAD
- Afbeeldingstechnieken behorend bij The Elements of User Experience

3.9.3 Nadrukken

De nadruk van dit systeem lag op het vlak van usability. Het systeem moest voldoen aan het Internet Calendaring and Scheduling Core Object Specification (RFC 2445 van IMC zie: <http://www.imc.org/rfc2445>). Op die wijze wordt aan een standaard voldaan. Door gebruik te maken van deze standaard wordt data-uitwisseling mogelijk tussen verschillende applicaties.



3.10 Resultaten

De volgende producten zijn opgeleverd:

- Definitiestudie (met plan van aanpak)
- Pilotontwikkelplan
- Een werkende agenda



4. Oriëntatie en planning

4.1 Inleiding

In dit hoofdstuk wordt beschreven hoe ik me heb georiënteerd op de opdracht en hoe ik ben gekomen tot het maken van bepaalde keuzes. Om een reëel beeld te kunnen geven van die keuzes en ze te kunnen verantwoorden wordt er eerst uiteengezet wat de situatie was van het systeem bij aanvang van het project. Daarna worden de doelstellingen van de op te leveren producten gegeven en wat de uiteindelijke planning is geworden om deze doelstellingen binnen de duur van het project te kunnen behalen. Afsluitend wordt het projectplan uiteengezet, daarin komt naar voren hoe ik mijn werkzaamheden heb gefaseerd en er wordt aangegeven voor welke pilots (deelproducten) is gekozen en waarom.

4.2 Oriëntatie

In deze fase heb ik uitgebreid bekeken wat er nodig is om dit project te laten slagen. Wat er al bestond en uitgebreid moest worden, wat er nieuw ontwikkeld moest worden en welke bronnen daarvoor nodig waren. Ik heb een inventarisatie van de beginsituatie gemaakt en de doelstellingen van de op te leveren producten bepaald. Zodat ik met die doelstellingen kon bepalen of het project geslaagd was.

Tijdens mijn oriëntatie heb ik ter begeleiding voornamelijk gebruik gemaakt van het boek *The Elements of User Experience* (zie: 7.1: Geraadpleegde literatuur) omdat dit in een aantal beknopte stappen aangeeft voor welke strategie je kiest en wat de omvang is van wat je wilt gaan doen. Tevens wordt er dan vastgelegd wat de omvang van het product is en welke structuur het product krijgt. Daarnaast wordt het skelet (de indeling) en de voorkant (het grafische ontwerp) van het product bepaald.



4.2.1 *Beginsituatie*

Toen ik bij Connectholland aan deze opdracht begon was het projectmanagementsysteem Aeolus al gemaakt door een andere programmeur. En ik moest daar ook zelf mee gaan werken om mijn project vast te leggen. Dit was handig, want dan kon ik vanzelf inventariseren hoe Aeolus werkte en waar eventueel problemen zaten. In Aeolus konden gegevens over projecten worden vastgelegd in tickets.

Zoals:

- Titel
- Omschrijving van de werkzaamheden
- Klant
- Tarief
- Uitvoerende
- Verantwoordelijken
- Deadline
- Status
- Prioriteit
- Begrote uren
- Geschatte uren

Een ticket kreeg een uniek nummer, het ticketnummer. Als er werkzaamheden voor werknemers van Connectholland ingepland moesten worden, dan kon dat in Google Calendar door een afspraak in te plannen in de planningsagenda van bijvoorbeeld de uitvoerende met in de titel dat ticketnummer. Op dezelfde manier konden uren geregistreerd worden in de registratie agenda van de betreffende werknemer.

Twee keer per dag vroeg Aeolus de informatie van Google Calendar op om dat op te slaan in de database die bij Aeolus hoort. Die gegevens waren daarmee ook beschikbaar binnen het systeem om bijvoorbeeld uit te zoeken of de geregistreerde uren overeenkomen met de begrote uren.



4.2.2 Definitiestudie

In de oriëntatiefase heb ik ook een definitiestudie gedaan. Dit is een stap uit de projectbeheersingsmethodiek IAD (Iterative Application Design), waarin de kenmerken van een project worden vastgelegd. Ik heb in de opdrachtomschrijving vastgelegd dat ik slechts daar gebruik zou maken van onderdelen uit IAD waar ik dat nodig achtte. Ik heb in de oriëntatiefase besloten een definitiestudie te maken omdat ik daarmee goed vast kon leggen hoe het project zou gaan verlopen.

De definitiestudie van IAD schrijft voor om onder andere de volgende zaken bij aanvang van een project vast te leggen:

- Plan van Aanpak
 - Inhoud en omvang van het project
 - Aanpak van het project
 - Projectinrichting
 - Kwaliteitsborging
 - Gedetailleerde planning
- Systeemeisen
- Systeemconcept
- Technische voorzieningen
- Verdere ontwikkeling

Deze zaken heb ik in de definitiestudie vastgelegd omdat ik dan een goede afbakening kon maken van het project (zie bijlage 7.6). De voor mij belangrijkste punten die daaruit naar voren kwamen waren vooral de fasering van het project (zie paragraaf 4.2.3) en de indeling van de uitvoering in pilots (zie paragraaf 4.3.2). Deze zaken waren naar mijn idee belangrijk omdat ze een rode lijn creëerden voor het project en het te bewandelen pad uitstippelden.

Tijdens de definitiestudie kwam ik er achter dat Aeolus niet overeenkwam met de projectinrichting bij Connectholland. In paragraaf 2.3 heb ik laten zien hoe de projectinrichting van Connectholland is. Klanten hebben vaak meerdere projecten bij Connectholland en in Aeolus was het zo ingericht dat klanten meerdere tickets hadden bij Connectholland. Een ticket is echter geen project, het is slechts een deel van een project, maar er was niet vastgelegd bij welk project die tickets hoorden. Er was dus geen verzameling van tickets bij een project.



Dat dit niet werd vastgelegd vond ik heel onhandig, omdat op die manier nooit een makkelijke verzameling van gegevens over een project te verkrijgen was. Er moest gezocht worden op klantnaam en dan moest degene die informatie over een project zocht maar bekijken, aan de hand van de omschrijvingen die bij de tickets stonden, welk ticket bij welk project hoorde.

Het probleem met projecten was echter niet interessant voor mijn project, ik moest een agenda maken die tijden plant en registreert die bij een ticket horen. Wat wel interessant was voor mijn project was dat het ook niet mogelijk was om te plannen en registreren voor één bepaalde werkzaamheid, want die werkzaamheden stonden in de omschrijving van een ticket. Dit zou betekenen dat er altijd gepland of geregistreerd zou gaan worden op een verzameling van werkzaamheden.

Het was naar mijn idee onwenselijk dat er werd gepland en geregistreerd op een verzameling van werkzaamheden, want daarmee verlies je informatie over onder andere de voortgang van en de knelpunten bij een project. Terwijl een projectmanagementsysteem er mede op gericht moet zijn om juist dat soort informatie vast te leggen.

Voor dit probleem waren een aantal oplossingen. Ik kon de agenda maken zonder rekening te houden met dit probleem en het eenvoudig houden voor mijzelf. Ik kon het probleem voorleggen aan mijn opdrachtgever en voorstellen dat een collega Aeolus op dit punt zou verbeteren. Of ik kon dit probleem meenemen in mijn proces en zelf een nieuwe structuur voor het projectmanagementsysteem ontwerpen. Deze laatste oplossing had mijn voorkeur omdat ik graag meerwaarde voor mijn project wilde creëren. Ik wilde niet gewoon iets dat al bestaat kopiëren en hiermee dus in feite alleen maar de plaats van de agenda ten opzichte van Aeolus veranderen. Samen met mijn opdrachtgever heb ik besloten dat de laatste oplossing inderdaad de beste was. En omdat ik toch al ingrijpend dingen zou gaan veranderen hebben we ook besloten om het eerder genoemde probleem met het verzamelen van tickets als projecten mee te nemen in mijn ontwerp.

In hoofdstuk 5 is terug te lezen hoe ik het ontwerpen en invoeren van deze nieuwe structuur in mijn proces heb weten te integreren.



4.2.3 Fasering van het project

Het leek mij handig om een fasering aan te brengen in het project. Daarmee wordt een project overzichtelijker. Om die fasering in het project te kunnen aanbrengen heb ik gebruik gemaakt van het boek *The Elements of User Experience*. In dit boek staat duidelijk beschreven hoe je een project goed kan inrichten, welke stappen er doorlopen moeten worden om tot een geslaagd eindproduct te komen. Het boek kent een indeling in vijf stappen: strategy; scope; structure; skeleton en surface. Maar ik heb een andere indeling gemaakt aan de hand van deze vijf. Voornamelijk omdat in deze vijf stappen de stap “uitvoering van het project” ontbrak. Met alleen plannen maken voor een project dat moet worden uitgevoerd zal dat project nooit slagen.

Voor het faseren van de uitvoering heb ik gebruik gemaakt van het idee uit het boek *Iterative Application Design*, waarin een project wordt opgedeeld in pilots (deelproducten). Dan leg je in feite vast uit welke deelproducten het gehele product bestaat, om op die manier meer controle te verkrijgen over het verloop van het project. Er wordt namelijk ook in aangegeven wat een dergelijke pilot allemaal behelst en hoe de pilots elkaar dienen op te volgen in het verloop van het project.

Samen met de oriëntatiefase en de evaluatiefase zijn dit stappen die ik uit deze boeken geabstraheerd heb:

1. Oriëntatie Hierin wordt er kennis gemaakt met het project. Er wordt gesnuffeld aan de verschillende mogelijkheden die er zijn om dit project succesvol te laten verlopen.
2. Analyse Hierin wordt geanalyseerd wat een handige strategie (strategy) is achter het project en er wordt vastgelegd wat de omvang (scope) is.
3. Ontwerp De structuur (structure) van het product wordt bepaald. Er wordt een skelet (skeleton) gemaakt voor het product en er wordt een voorkant (surface) ontworpen.
4. Uitvoering De uitvoering van het project zal worden opgedeeld in pilots. De bepaling daarvan zal naar voren moeten komen in de analyse en ontwerp fases.
5. Evaluatie Er wordt teruggeblikt op het verloop van het project, het proces. En er wordt gekeken naar de waarde van de producten.

Nadat ik deze fasering had vastgelegd was ik in feite al begonnen met de analysefase. En kon ik beginnen met het schrijven van een echt plan voor mijn project, inclusief een gedetailleerde planning. In hoofdstuk 5 wordt beschreven hoe ik deze fases daadwerkelijk heb doorlopen.



4.3 Projectplan en pilots

4.3.1 Inleiding

Tijdens de definitiestudie heb ik vastgelegd met welke pilots ik aan de slag kon. Het zijn vrij grote pilots geworden, dit komt voornamelijk doordat ik nog geen concreet beeld had van wat er allemaal moest gebeuren. Ik wist in ieder geval dat de drie pilots die ik in de definitiestudie had vastgelegd gedaan moesten worden. Dan zou later, tijdens het schrijven van een pilotplan (zie bijlage 7.7) wel uitkomen wat er precies gedaan moest worden. Deze indeling bestond in ieder geval uit brokken die zelfstandig uitgevoerd konden worden en dat leek mij belangrijk, omdat het project zo gestructureerder werd. Daardoor hoef je je niet op alles tegelijk te focussen. De pilots worden verder belicht in dit hoofdstuk. Daarnaast worden de doelstellingen van de op te leveren producten vastgelegd en wordt uitgelegd waarom deze doelstellingen zijn gekozen.

4.3.2 Pilots

In paragraaf 4.2.2 heb ik reeds uitgelegd wat me ertoe heeft bewogen om deze pilotindeling te kiezen. Deze indeling is mede tot stand gekomen door een keuze te maken uit de verschillende oplossingen die naar voren kwamen uit een inventarisatie van alle problemen. Tevens zal er in paragraaf 5.4 verder worden aangegeven welk proces ik heb doorlopen om deze pilots af te ronden. Voor welke problemen ik ben komen te staan en welke oplossingen ik heb gekozen.

Project, ticket en taak structuur

In deze pilot heb ik een nieuwe structuur aangebracht in Aeolus. Deze pilot was nodig om de managementinformatie, die onder andere aan de hand van de agenda, uit Aeolus gehaald kan worden meer accuraat te maken. Er kan namelijk na het doorlopen van deze pilot informatie uit Aeolus gehaald worden over één bepaalde werkzaamheid en niet alleen maar over een verzameling van werkzaamheden.

Tevens was deze pilot nodig om de gehele structuur van Aeolus meer overeen te laten komen met de projectinrichting zoals geschetst in paragraaf 2.3 en reeds uitgelegd in paragraaf 4.2.2.

JSON

In deze pilot heb ik gezorgd dat het framework Windmill ook gegevens in Javascript Object Notation (JSON⁴) kan maken en aan de internet browser van de gebruiker kan sturen. Deze pilot was nodig om gemakkelijker gegevens uit de database te kunnen verwerken met Javascript. Dit verwerken moet gebeuren in en door de internet browser van de gebruiker.

⁴ Zie bijlage 7.2: Afkortingenlijst



Frontend

In deze pilot heb ik gezorgd dat de gegevens die uit de database worden opgevraagd gerepresenteerd worden op de gebruiker zijn beeldscherm. Daarnaast heb ik mogelijk gemaakt dat er allemaal acties met die gegevens mogelijk zijn, zoals opzoeken, toevoegen en verwijderen. Deze pilot is de spil in het agenda project. Het is de vertaling van de database naar een voor de gebruiker begrijpelijke structuur.

4.3.3 Doelstellingen van de op te leveren producten

Na de oriëntatie en analyse fase kon ik in de ontwerpfase ook de doelstellingen van de op te leveren producten vastleggen, doordat ik een concreter beeld had gekregen van het project en een plan had gemaakt voor het verloop van het project.

De volgende doelstellingen heb ik toen vastgelegd:

- Integreren van de agenda binnen Aeolus
- Vereenvoudigen van planning en registratie
- Vergroten betrouwbaarheid, doordat werkzaamheden op taakniveau worden gepland en geregistreerd
- Vergroten overzichtelijkheid, doordat er niet tussen vensters hoeft te worden gewisseld.
- Vergroten overzichtelijkheid, doordat alle benodigde informatie ook op de agenda weergegeven kan worden, zoals titel en omschrijving.

Mede door de bovenstaande doelstellingen zal het gebruikersgemak worden vergroot. Het systeem zal eenvoudiger worden doordat er binnen Aeolus “shortcuts” gecreëerd kunnen worden naar de agenda. Het beoogde resultaat van het project is een volledige vervanging van de agenda van Google Calendar waarin deze doelstellingen zijn behaald.



4.4 Planning

Voor aanvang van het project heb ik een ruwe schets van de planning voor het project gemaakt. Onderstaande tabel is de ruwe schets van die planning. Daarin is voor een aantal producten die ik van te voren dacht te gaan maken een opleverdatum vastgelegd.

November	December	Januari	Februari	Maart
13	4	1	5 Pilot	5 Pilot
20 Plan van aanpak	11	8	12	12
27	18	15 Pilotplan	19 Pilot	19 Procesverslag
	25 Definitiestudie	22	26	
		29		

Na aanvang van het project, ongeveer in de vijfde week, heb ik een gedetailleerdere planning gemaakt aan de hand van de in de oriëntatiefase vastgestelde fases en pilots. Deze gedetailleerde planning is in de onderstaande tabel weergegeven per projectweek.

Week	Fase	Pilot	Werkzaamheid
1	1		Oriëntatie op de werking van het huidige systeem
2	2		Plan van aanpak opstellen
3	2		Fouten in het huidige systeem inventariseren
4	2 / 3		Oriëntatie op de werking die het toekomstige systeem moet hebben
5	2 / 3		Definitiestudie opstellen
6	3	1	Database ontwerp maken voor het nieuwe systeem
7	4	1	Omzetten van gegevens uit oude database naar nieuwe
8	4	1	Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
9	4	1	Verwerken van testresultaten
10	4	1	Omzetten naar nieuwe structuur voor eindgebruiker
11	4	2	Generator van Propel ⁵ ombouwen
12	4	2	Javascript schrijven om gegevens op te vragen en te verwerken
13	4	2	Javascript schrijven om gegevens in de pagina te verwerken/plaatsen
14	4	3	Agendapagina maken waarop gegevens gepresenteerd kunnen worden
15	4	3	Gegevens uit database presenteren op agenda pagina
16	5		Eindverslag afronden
17	4	3	Gegevens aanpasbaar maken
18	5		Vorbereiden presentatie
19	5		
20			

⁵ Propel is een Object Relational Mapping (ORM) framework voor PHP5, zie paragraaf 5.4.2



4.5 Analyse

In deze paragraaf wordt beschreven hoe ik de gekozen strategie achter dit project heb bepaald en hoe ik heb bepaald welke omvang het project zou hebben. Hiermee wordt duidelijk welke doelstellingen dit project heeft en wat de omvang is van het project.

4.5.1 Strategie

Bij het vastleggen van de doelen heb ik als leidraad de opdrachtomschrijving gebruikt. Maar ook de in de oriëntatiefase ontdekte problemen heb ik meegenomen in de doelstellingen.

Een doel van de agenda is om op een zeer eenvoudige wijze planningen te maken en uren te registreren binnen het projectmanagementsysteem Aeolus. Hierbij zal gebruik worden gemaakt van AJAX⁶ methodiek. Dit verkleint de wachttijd en de gebruiker krijgt onder andere daardoor het gevoel dat zijn acties ogenblikkelijk plaatsvinden. Op die manier kan er dan direct vanuit de overzichten met taken worden gepland en worden geregistreerd. Los daarvan moet de agenda ook zelfstandig kunnen werken, aangezien het een zelfstandige module moet worden die in andere projecten of websites van klanten terecht moet kunnen komen.

4.5.2 Omvang

Bij de bepaling van de omvang worden de functionele specificaties en de vereisten vastgelegd voor het toekomstige systeem. Mede omdat je dan weet wat je wel en wat je niet aan het maken bent. Ik heb daarom een aantal functionele eisen vastgelegd.

Functionele eisen:

- De gegevens uit de agenda moeten wijzigbaar zijn door middel van drag-and-drop functionaliteit, als het gaat om het wijzigen van tijdstippen.
- Planning en registratie van uren wordt in een weekoverzicht weergegeven.
- Er moet gezocht kunnen worden in afspraken
- De code is DRY geschreven
- De code is uitbreidbaar, voor toepassing in andere systemen

⁶ Zie bijlage 7.2: Afkortingenlijst



4.6 Ontwerp

In de ontwerpfase heb ik aan de hand van diagrammen bepaald wat de structuur is van de agenda en het achterliggende skelet, daarnaast heb ik het grafisch ontwerp van het systeem gemaakt (de voorkant). Dit zijn diagrammen en ontwerpfasen die worden omschreven in het boek *The Elements of User Experience*. Met de diagrammen en ontwerpen in deze paragraaf wordt inzichtelijk gemaakt via welke weg de gebruiker aan zijn gegevens kan komen en hoe die gegevens dan gepresenteerd gaan worden.

Bij het ontwerpen heb ik echter alleen de zaken die echt met de agenda te maken hadden vastgelegd. Ik heb dus geen ontwerpen gemaakt voor de nieuwe structuur in Aeolus. Deze heb ik gewoon gestoeld op de huidige structuur, waardoor het niet nodig was om uitgebreid te ontwerpen.

4.6.1 Structuur

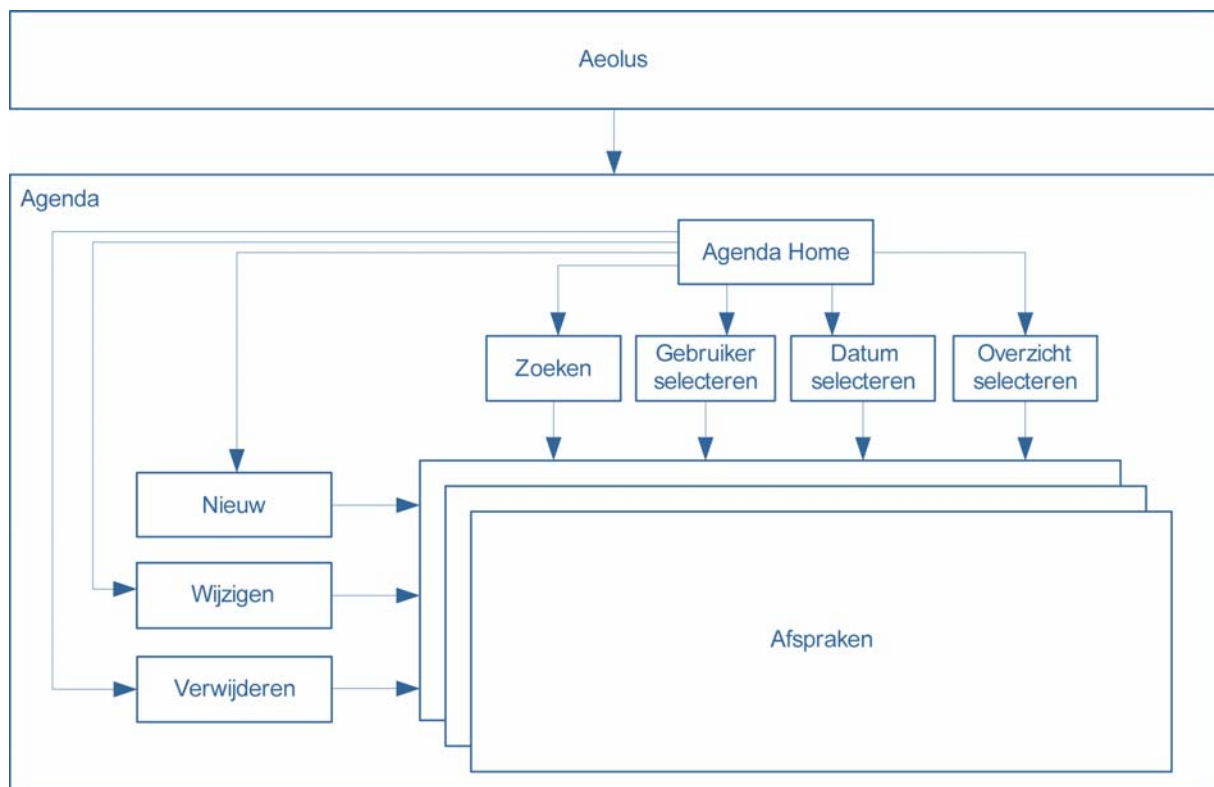
Bij het vastleggen van de structuur, het navigation-design, heb ik eerst geïnventariseerd welke acties er mogelijkerwijs door de gebruiker zouden moeten worden gedaan.

Daarnaast heb ik geïnventariseerd welke acties de gebruiker mogelijkerwijs zou willen doen.

Daarbij heb ik onder andere gekeken welke gebruikersacties er mogelijk zijn in Google Calendar en welke daarvan van toepassing waren op de agenda van Aeolus. Ik ben er hierbij van uit gegaan dat dit ook geldt voor de agenda als losstaande module binnen het framework, dit zal bij implementatie in een andere omgeving blijken. Door de open structuur waarin ik de agenda heb geprogrammeerd zullen dergelijke aanpassingen eenvoudig zijn.

De gebruikersacties die van toepassing waren zijn: nieuwe afspraken toevoegen aan jouw agenda; afspraken wijzigen of verwijderen; zoeken in bestaande afspraken en een datum selecteren die je wilt zien. Bovendien kan je in Google Calendar meerdere agenda's laten zien in één overzicht. Je kunt dan kiezen welke agenda's er weergegeven worden en deze worden gemixt op bijvoorbeeld een weekoverzicht. Dit is ingewikkeld om te programmeren en daarom heb ik ervoor gekozen om slechts twee agenda's per overzicht weer te geven, namelijk per werknemer zijn planningsagenda en registratieagenda. De acties die ik heb gekozen om te gebruiken in de agenda van Aeolus zijn weergegeven in de figuur op de volgende pagina.





Figuur 3: Flowdiagram van de agenda

De bovenstaande figuur geeft aan hoe de gebruiker bij zijn gegevens kan en het geeft aan welke acties een gebruiker kan ondernemen om bijvoorbeeld aanpassingen te maken aan de agenda.

Gegevens selecteren

- Zoeken:** De gebruiker kan doormiddel van het invoeren van zoektermen een overzicht verkrijgen met afspraken waarvan de gegevens voldoen aan zijn zoektermen
- Gebruiker selecteren:** De gebruiker kan een agenda van een andere gebruiker bekijken
- Datum selecteren:** De gebruiker kan een datum selecteren die hij in het huidige overzicht te zien wil krijgen
- Overzicht selecteren:** De gebruiker kan de vorm van het overzicht bepalen, zo kan er bijvoorbeeld een week- of een maandoverzicht gekozen worden

Gegevens aanpassen

De gebruiker kan de gegevens op het overzicht met afspraken aanpassen. Mogelijke acties zijn een nieuwe afspraak toevoegen en een bestaande afspraak wijzigen of verwijderen.



4.6.2 Skelet

Bij het vastleggen van het skelet (of ook wel de plaatsing van besturingselementen op de pagina) heb voornamelijk de indeling die Google Calendar hanteert aangehouden. Het skelet is een voorfase van het grafische ontwerp. Deze indeling was overigens een indeling die ik in bijna alle onderzochte voorbeelden tegenkwam tijdens mijn inventarisatie van mogelijkheden (zie paragraaf 5.2.4). Iedereen binnen het bedrijf gebruikt in Google Calendar voornamelijk het weekoverzicht om in te plannen en te registreren en niet een maandoverzicht of een dagoverzicht. Dat heeft mij ertoe doen besluiten eerst een weekoverzicht te maken. Maar daarbij wel rekening houdend met het feit dat er ook eenvoudig een maandoverzicht te realiseren moet zijn.



Figuur 4: Wireframe van de applicatie

In het bovenstaande wireframe is de indeling van de pagina bepaald. Deze indeling is gebaseerd op de indeling van Aeolus. In Aeolus is de navigatie opgenomen in de bovenste balk van de pagina. Daaronder is een balk waarin de gebruiker zoektermen in kan vullen om gegevens op te vragen. Dit heb ik in mijn indeling zo gelaten om eenduidigheid te behouden en verwarring te voorkomen.

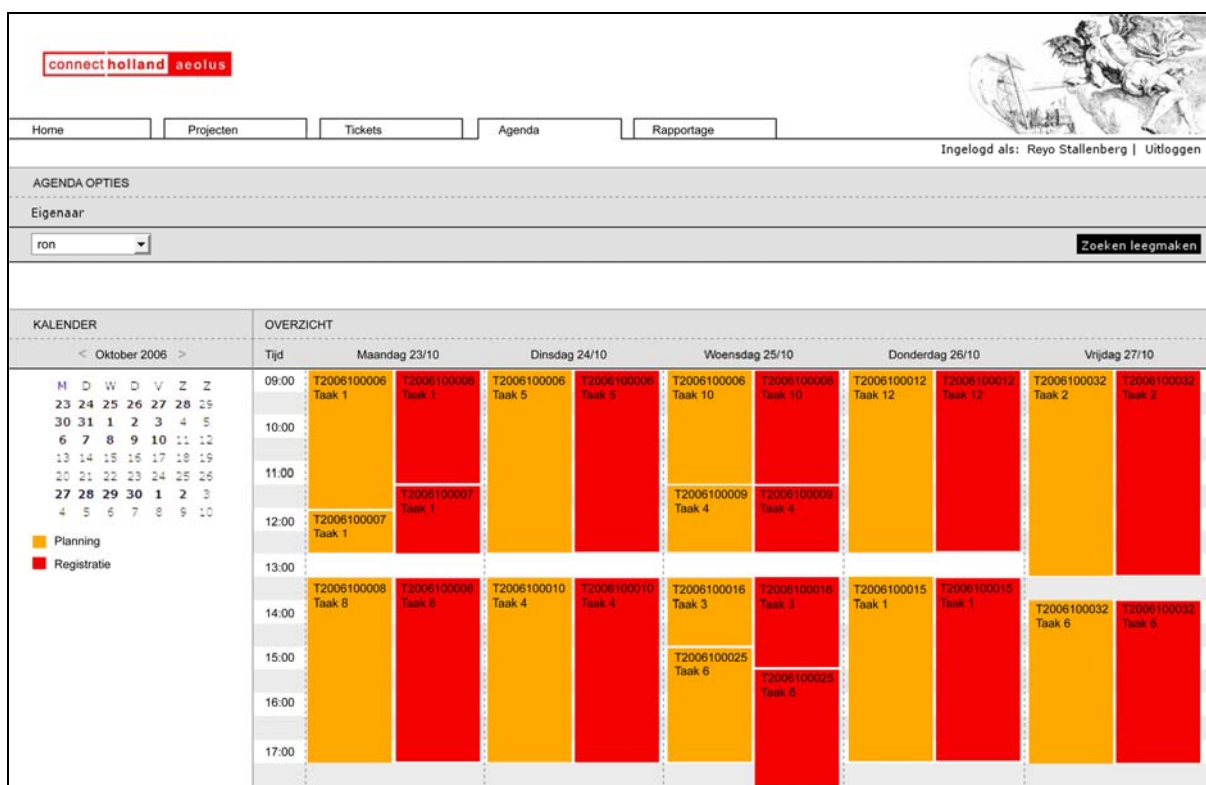


Daarna waren er nog drie elementen over die een plaats op de pagina moesten krijgen. Het afspraken overzicht element was het belangrijkste element. De andere elementen zijn de weg naar dat overzicht toe. Daarom heb ik die andere elementen als navigatie gezien. Omdat in de meeste gevallen navigatie links op de pagina wordt geplaatst heb ik daar ook voor gekozen.

4.6.3 Voorkant

Aan de hand van het in Figuur 4 weergegeven wireframe heb ik een conceptueel grafisch ontwerp gemaakt (zie Figuur 5). Ik heb deze vormgeving zelf gemaakt, aan de hand van de vormgeving die al bestond in Aeolus. Daarna heb ik hem laten goedkeuren door de vormgeefster van Aeolus. Aangezien er al gepland bleek te staan om de gehele vormgeving van Aeolus opnieuw te maken is er verder niets veranderd aan dit concept. Als de vormgeving van Aeolus nagelopen wordt, wordt de vormgeving van de agenda ook nagelopen en eventueel opnieuw gemaakt.

Deze vormgeving is voornamelijk gemaakt om een feeling te creëren en een beeld te kunnen geven van hoe het resultaat er uiteindelijk uit zou kunnen zien.



Figuur 5: Conceptuele vormgeving Agenda



5. Proces en uitvoering

5.1 Inleiding

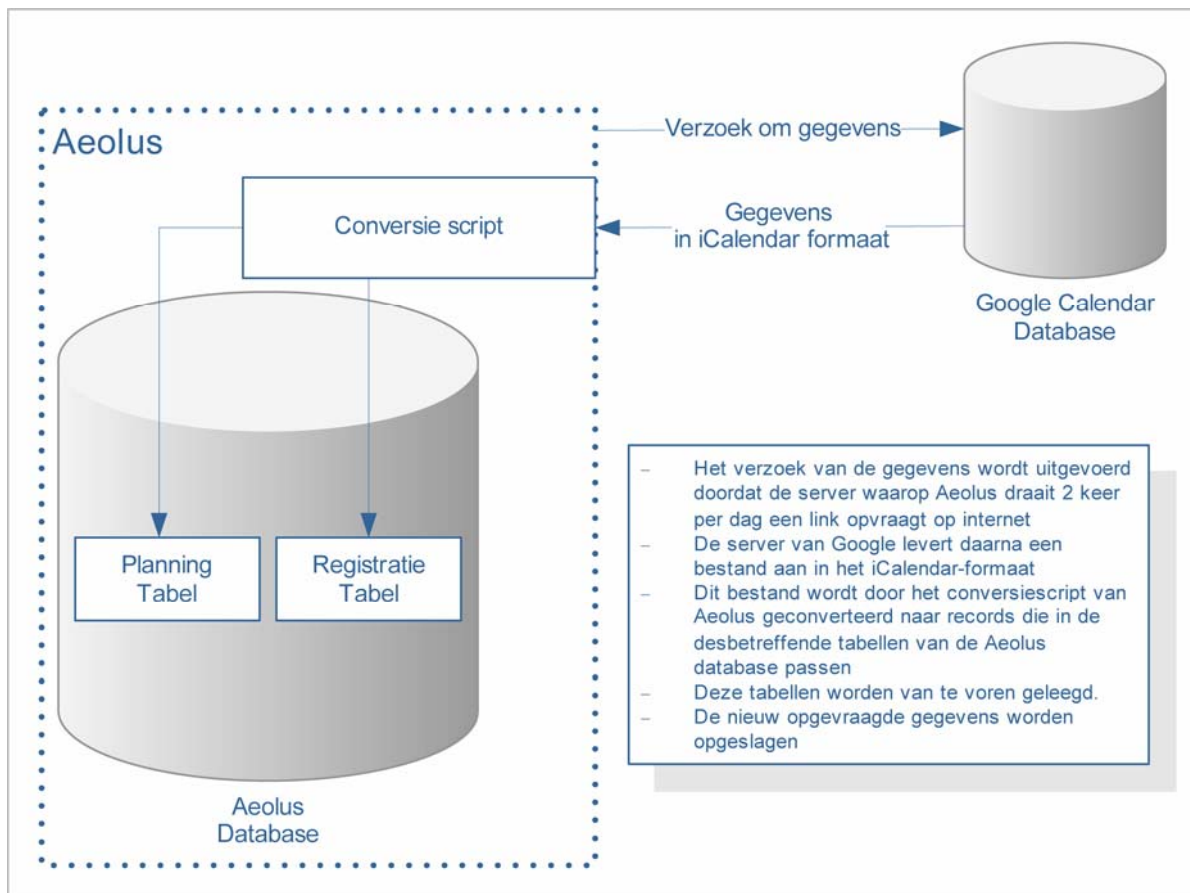
Aan de hand van de fasering van de opdracht worden in dit hoofdstuk de werkzaamheden besproken behorende bij de opdracht. Er wordt in aangegeven op welke problemen ik ben gestuit gedurende het proces en welke oplossingen ik heb gekozen. Daarnaast wordt beschreven waarom ik die oplossing heb gekozen.

5.2 Analysefase

5.2.1 Beginsituatie

Tijdens mijn definitiestudie ging ik uitzoeken hoe de registratie- en planningsgegevens in het projectmanagementsysteem Aeolus terecht kwamen. Deze gegevens werden geplaatst in een online agenda van Google (Google Calendar). Daarvoor moest elke werknemer twee agenda's aanmaken in dat systeem, één voor zijn planning en één voor zijn urenregistratie. In Google Calendar is het mogelijk om deze agenda's te delen met andere gebruikers. Dus elke werknemer moest die agenda's delen met de projectmanagers, zodat deze voor hen de planning kon maken. Tevens kunnen deze agenda's worden uitgelezen met andere agenda programma's, zoals Microsoft Outlook, mits men over een link beschikt naar de betreffende agenda. Dit is mogelijk doordat die link een link is naar een bestand in een gestandaardiseerd formaat (het iCalendar-formaat). Aeolus maakt gebruik van deze links om de gegevens uit de agenda's van de werknemers uit te lezen en te converteren naar de juiste vorm om op te kunnen slaan in de database van Aeolus. Om dit te kunnen doen diende de werknemer als titel voor zijn planning of registratie een nummer op te nemen dat overeenkomt met het nummer van het ticket waarvoor gepland of geregistreerd wordt. De koppeling tussen Aeolus en Google Calendar is op de volgende pagina schematisch weergegeven.





Figuur 6: Schematische weergave van koppeling tussen Aeolus en Google Calendar



5.2.2 Probleem 1: Structuur Aeolus

Om mijn planning te maken en mijn urenregistratie te doen moest ik ook gaan werken met Aeolus in combinatie met Google Calendar. Dus voor dit project is een ticket aangemaakt in Aeolus met deze vorm.

Ticketnummer	2006060096	Klant	Connectholland	Begroting	0
		Contactpersoon	J. Soeterbroek	Eigen schatting	NNB
Datum ontvangen	13-11-2006				
Leeftijd ticket	124 dagen	Titel			
Deadline	23-03-2007	Ontwikkeling Agenda voor Aeolus			
		Omschrijving			
Accountmanager	Jeroen	- oriëntatie op project - analyse - inventarisatie maken - ontwerp - structuur, skelet en voorkant ontwerpen - uitvoering - pilots uitvoeren			
Projectmanager	Ron				
Eigenaar	Reyo				
Status	Gepland				
Prioriteit					

Figuur 7: Weergave van een ticket in het oude Aeolus

Mijn planning en urenregistratie kon ik dus doen voor het hele project en niet voor een onderdeel ervan. Dus bijvoorbeeld niet voor het derde punt uit de omschrijving. Daarmee ging er dus verloren wat er precies is gedaan en op welk moment. Er werd alleen vastgelegd aan welk ticket er gewerkt diende te worden. En dan kon later worden vastgelegd aan welk ticket er gewerkt was. Maar niet wat er precies gedaan moet worden. En later niet precies wat er gedaan is. Dit kwam doordat de taken die bij een ticket horen als een lijstje in de omschrijving van dat ticket werden geplaatst. In Google Calendar is niet te registreren welk onderdeel van dat lijstje gedaan dient te worden of gedaan is, want de koppeling loopt via het ticketnummer.



De manier van gegevenskoppeling zoals weergegeven in Figuur 6 is foutgevoelig, omdat de gebruiker zelf het nummer moet kopiëren en plakken tussen de verschillende systemen. Als daar een fout in wordt gemaakt, dan is in feite de registratie verloren gegaan. Daarnaast is dit project ontstaan omdat de opdrachtgever het onwenselijk vond om met meerdere systemen tegelijk te werken om één doel te bereiken. Dit kan verwarrend werken en verkleint het overzicht. De wens van de opdrachtgever was ook om direct vanuit een ticket te kunnen plannen en registreren. Doordat het plannen en registreren in een toepassing van een derde partij geschiedt, was het in principe niet mogelijk om dit te doen. Google heeft daarvoor een oplossing, want via een API⁷ van Google kan dat weer wel. Als de gegevenskoppeling via die API was opgezet, was deze wens al vervuld. Dit was niet het geval en aangezien de wens was om te beschikken over een eigen agenda verviel de oplossing om het via die API te doen.

Een belangrijk doel in dit project was dat de informatie die wordt opgeslagen over urenplanning en -registratie ook specifiek is. Hierdoor wordt duidelijk wat er van bijvoorbeeld een programmeur wordt verwacht, nadat de planning gemaakt is. Ook wordt uit Aeolus achteraf duidelijk wat er feitelijk is gedaan en hoelang dat heeft geduurd. Deze doelen stelde ik mijzelf omdat ik vind dat een toepassing zoals de agenda die ik moest maken wel een toegevoegde waarde moet hebben ten opzichte van de oude situatie. Het moet niet een kopie zijn van wat er al was.

Het maken van een dergelijke kopie is heel erg veel werk en het enige wat je er mee oplost is dat je twee toepassingen samenvoegt. Daarom moest er op het niveau van werkzaamheden kunnen worden geregistreerd, want je hebt niets aan een systeem dat informatie over het verloop van een project dient te verschaffen maar dat niet doet.

De oplossing voor dit probleem leek simpel: de lijstjes die nu worden aangemaakt bij een ticket in de omschrijving moesten er uit verdwijnen en moesten worden vervangen voor taken. In feite een weinig ingrijpende verandering, maar op deze manier is een onderdeel van een ticket, een taak, wel uniek identificeerbaar geworden voor het systeem.

Met dit probleem en de bedachte oplossing ging ik naar mijn opdrachtgever. Ik heb gevraagd of het gewenst was dat ik een meerwaarde creëerde voor mijn project. In dat overleg bracht ik ook naar voren dat er andere dingen aan de structuur van Aeolus niet zo waren als gewenst. De opdrachtgever was het hiermee eens en samen hebben we besloten dat ik het oplossen van deze problemen mee zou nemen in mijn project.

⁷ Zie bijlage 7.2: Afkortingenlijst



5.2.3 *Probleem 2: Verwerken Database Gegevens*

Bij het vastleggen van de strategie achter de agenda (zie paragraaf 4.5.1) is bepaald dat de agenda een interactieve toepassing zou worden die met gebruik van de Ajax methodiek zorgt dat gegevens asynchroon worden opgevraagd en opgeslagen. Waardoor de gebruiker niet hoeft te wachten op het opvragen of versturen van een hele pagina, maar slechts op een gedeelte van die pagina. Ondertussen kan de gebruiker dan nog wijzigingen aanbrengen in een ander deel van de pagina. Of de gebruiker kan andere gegevens opvragen.

Het is al mogelijk met het framework Windmill om asynchroon gegevens op te vragen of te versturen. De opgevraagde gegevens worden dan teruggegeven in de vorm van XML⁸ of HTML⁸. Daarna moeten deze gegevens met behulp van een Javascript geplaatst worden in de pagina van de gebruiker. Als de gegevens in de vorm van XML zijn moet er in dat Javascript ook een conversie van XML naar HTML plaatsvinden.

Bij het verwerken van XML of HTML maakt Javascript gebruik van het Document Object Model⁹ (DOM), wat een W3C-specificatie is. Helaas is het niet zo dat iedere producent van een internet browser zich aan deze specificatie houdt. Daardoor is het vaak heel veel werk om een manier in Javascript te vinden die in de meeste internet browsers hetzelfde resultaat geeft. Bovendien is het in Javascript moeilijk om te testen of het resultaat in alle gevallen zo zal zijn als verwacht. Door beperkte foutopsporingsmogelijkheden in veel browsers.

Voor het bovengenoemde probleem is een oplossing. In Javascript is er een notatievorm voor objecten die wat structuur betreft veel weg heeft van XML, Javascript Object Notation (JSON). Een object in deze notatievorm is in één keer in te laden in het script dat er iets mee moet doen. Er hoeft dus maar een heel kort stuk code geschreven te worden om dat voor elkaar te krijgen. Terwijl bij het verwerken van XML alles moet worden uitgetypt. Dit komt doordat in deze notatie de eigenschappen van het object al vastliggen. Om XML te verwerken moet er bepaald worden welk stukje XML je wilt hebben en in welke variabele je dat stukje wilt opslaan. Daarna kan je bepalen wat je er mee wilt doen. In het voorbeeld van de agenda zou je het daarna in de pagina van de gebruiker kunnen plaatsen.

⁸ Zie bijlage 7.2: Afkortingenlijst

⁹ DOM is een W3C-specificatie en objectgeoriënteerde benadering van gestructureerde documenten zoals HTML-, XHTML- en XML-documenten, waarvan je gebruik kan maken in Javascript en andere toepassingen als je dergelijke bestanden wilt uitlezen



Om JSON te verwerken moet het dus alleen worden ingeladen. Dat JSON simpeler is in Javascript komt doordat het objecten van de programmeertaal zelf zijn. Daardoor zijn zij eenvoudiger te manipuleren met Javascript. Bovendien is hierdoor de kans dat een internet browser goed om kan gaan met JSON groter.

De keuze waarvoor ik kwam te staan was dus om veel, moeilijker onderhoudbare, ingewikkelde code te schrijven om een in verhouding tot JSON groot XML-document uit te lezen. Of om beknopte, makkelijker onderhoudbare, accurate code te schrijven om een voor de scripttaal eigen object mee in te laden. Dan moest ik wel een ingewikkelde aanpassing maken in het framework. Want het framework moet dan niet langer een XML-document aanleveren, maar een bestand dat JSON bevat. Aangezien het framework al in een vrij vroeg stadium XML aanmaakt van de data die uit de database is gehaald moet er veel worden gedaan om het aanleveren van een bestand met JSON mogelijk te maken.

Over dit probleem en de mogelijke oplossing ben ik gaan overleggen met de senior developer. Hij vond het verstandig om te kiezen voor JSON in plaats van het uitlezen van XML. Omdat daarmee een oplossing werd gecreëerd die in de toekomst goed van pas kan komen, want er komen steeds meer toepassingen met Ajax, waarbij het niet gewenst is om XML-documenten uit te lezen. In dat overleg heeft hij uitgelegd hoe ik deze aanpassing kon maken in het framework. In mijn beschrijving van de uitvoering van de pilot JSON (in paragraaf 5.4.2) leg ik uit hoe ik dat heb gedaan.



5.2.4 Inventarisatie van mogelijkheden

Tijdens het inventariseren van de problemen ben ik me ook gaan verdiepen in online agenda's. Ik heb een vergelijkend onderzoek gedaan. Als grote voorbeeld had ik al Google Calendar en ik heb ook wel eens gewerkt met de agenda van Microsoft Outlook (die overigens niet online is).

Daarnaast zijn er andere voorbeelden, zoals:

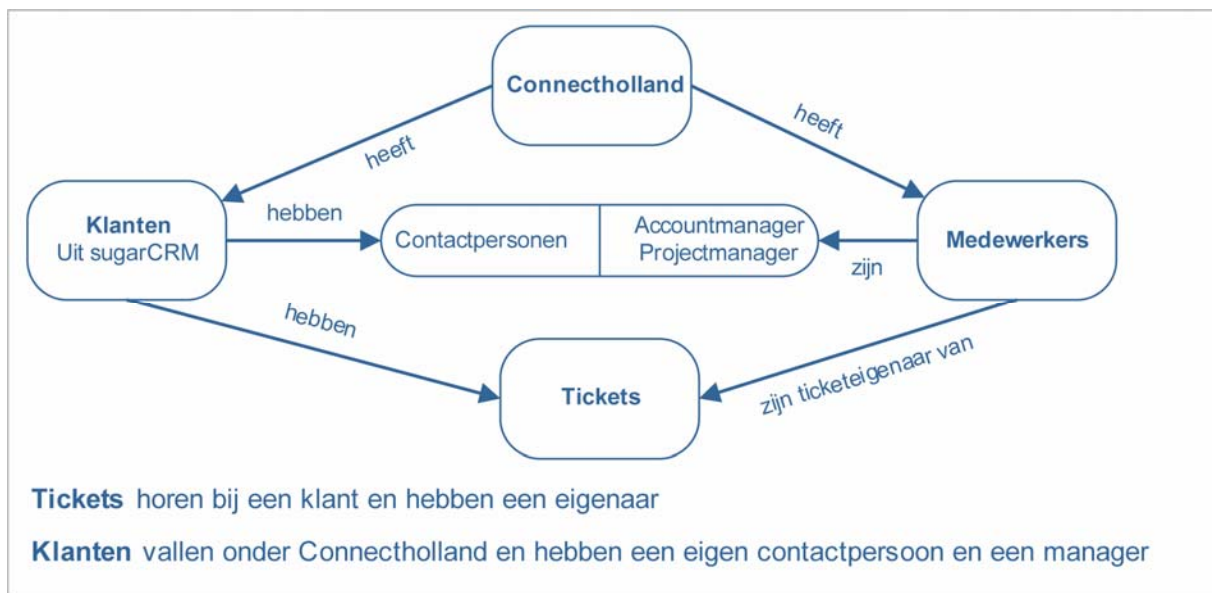
- 30boxes _____ 30boxes.com
- Calendarhub _____ www.calendarhub.com
- Trumba _____ www.trumba.com
- Kiko _____ www.kiko.com
- Yahoo! Calendar _____ calendar.yahoo.com
- Spongecell _____ spongecell.com
- Microsoft Calendar _____ calendar.msn.com

Ik heb gekeken wat deze verschillende toepassingen wel of niet konden en wat er handig aan was of niet. Dit heb ik niet gedocumenteerd, omdat het vooral ter oriëntatie van wat er mogelijk was was. En er was in de opdrachtomschrijving aangegeven dat de voornaamste wens was om een aan Google Calendar gelijkende agenda te maken. Bovendien leken alle agenda's sterk op elkaar, er waren maar zeer kleine verschillen. Zoals een dubbelklik actie bij de ene agenda, waarbij er bij de andere maar één keer geklikt moest worden op een item. Daarnaast is iedereen die met Aeolus werkt gewend om ook met Google Calendar te werken, waar ik overigens nooit klachten over heb gehoord. Dus als de nieuwe toepassing daar op lijkt is het voor iedereen veel makkelijker om te leren, omdat ze in principe de werking al kennen. Daarnaast is dit project een intern project. Wat ervoor zorgt dat er vanzelf feedback komt als er dingen die men anders zou willen.



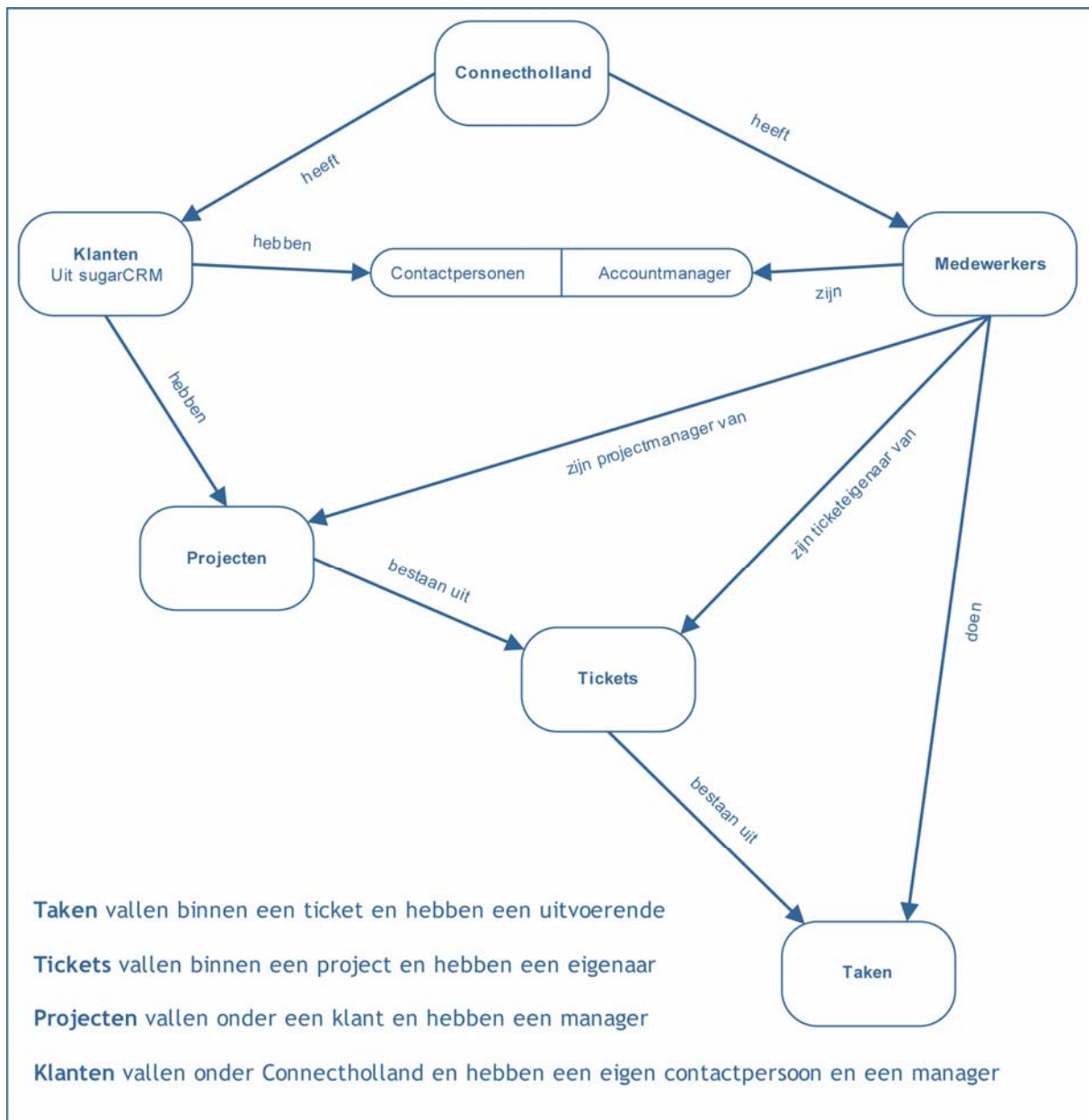
5.3 Ontwerpfase

Op basis van het overleg over de structuur van Aeolus ben ik een nieuwe structuur gaan ontwerpen, dit heb ik gedaan door eerst de huidige structuur te omvatten in een schema (zie Figuur 8) en daarna dit schema om te bouwen naar een schema waarin de voorgenomen veranderingen voorkomen (zie Figuur 9). Deze schema's heb ik gemaakt omdat het mij een goed inzicht kon verschaffen in de huidige structuur en daarop volgend hoe de nieuwe plannen daar dan precies in pasten.



Figuur 8: Structuur van Aeolus bij aanvang van het project



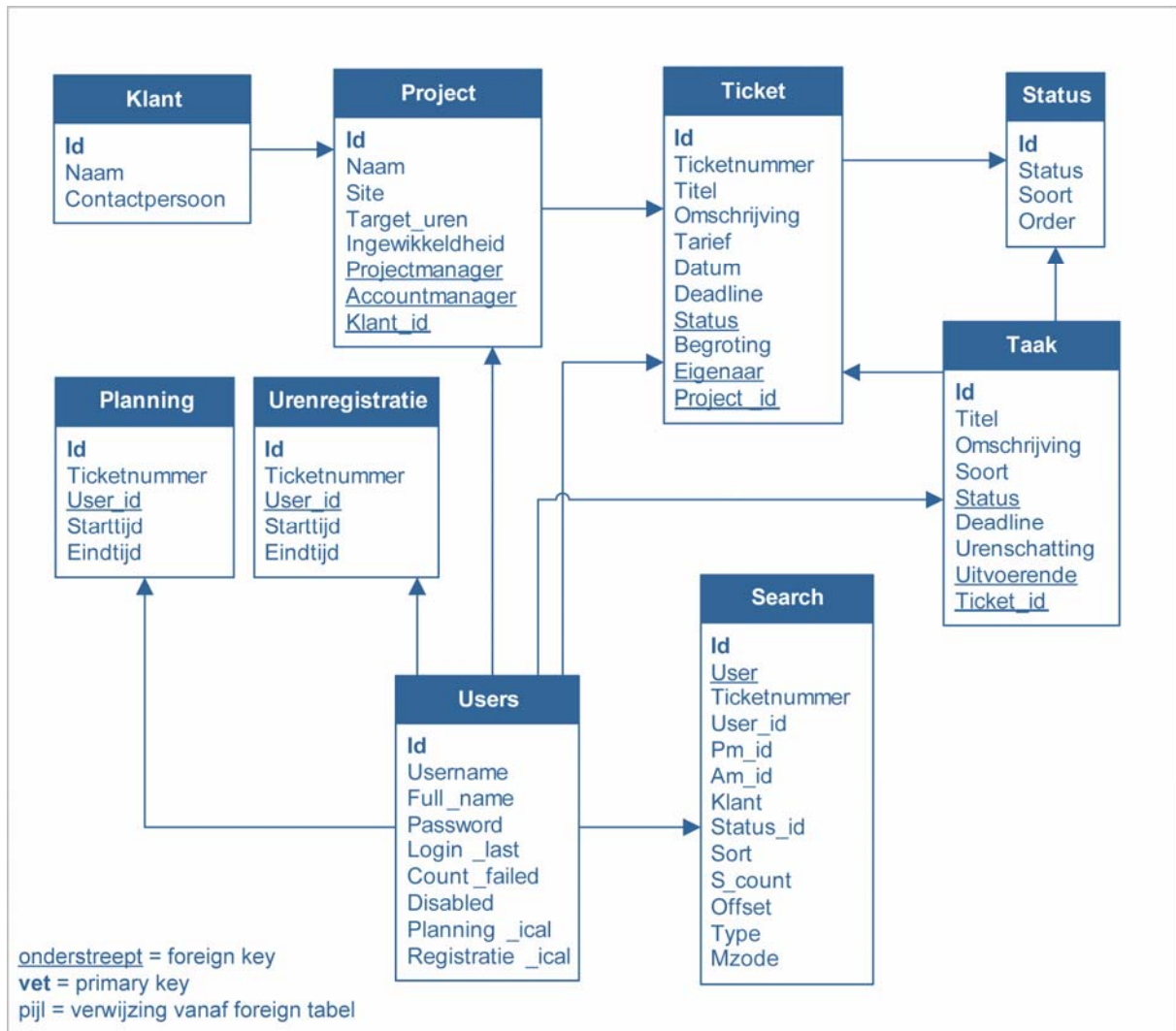


Figuur 9: Goedgekeurd voorstel nieuwe structuur van Aeolus
De project, ticket en taak structuur

In de nieuwe structuur zijn subtickets verdwenen, omdat in het oude systeem tickets en subtickets feitelijk hetzelfde waren. Een subticket bevatte hetzelfde nummer als het ticket waar het bij hoorde, maar dan met als toevoeging een letter aan het einde. Subtickets vielen daarmee dus onder bepaalde tickets, maar een ticket bestond niet uit een verzameling van gegevens uit een aantal subtickets. Een ticket had zijn eigen gegevens. Hierdoor ontstond er een moeilijk te begrijpen brei van gegevens. Als er uren waren gepland of geregistreerd op een ticket dan werd er in het totaaloverzicht van uren die bij dat ticket hoorde niet de uren uit de subtickets opgenomen. Dan is de vraag hoeveel uur er dan eigenlijk is gewerkt aan een ticket moeilijk te beantwoorden.



Mijn voorstel voor de nieuwe structuur werd goedgekeurd door de opdrachtgever. De nieuwe structuur moest vertaald worden naar een nieuw databaseschema, omdat de gegevens die bij projecten en taken horen uiteraard wel opgeslagen moeten kunnen worden. Ik heb daarop op basis van het oude databaseschema een nieuw database ontwerp gemaakt die is weergegeven in de onderstaande figuur.



Figuur 10: Database ontwerp nieuwe structuur



5.4 Ontwikkelfase

In deze paragraaf beschrijf ik welke werkzaamheden ik heb verricht om de verschillende pilots uit te voeren en af te ronden en welke oplossingen en aanpak ik heb gekozen voor problemen die ik tegenkwam.

5.4.1 Project, ticket en taak structuur

Tijdens de uitvoering van deze pilot heb ik veel werk verricht. Aeolus is een toepassing van het framework Windmill, waarbij er voor deze speciale toepassing modules zijn geschreven die een uitbreiding zijn op dat framework. Zo is er bij Aeolus bijvoorbeeld een ticketmodule. Deze module is een uitbreiding van de basismodule van het framework en regelt de output van de database records uit de tabel *Ticket* in XML, zodat dit met XSL omgezet kan worden naar HTML en dan verstuurd kan worden. XSL is een standaard voor het omzetten van de informatie in een XML-document naar een ander formaat zoals bijvoorbeeld HTML. Vanwege bijvoorbeeld het verdwijnen van subtickets in de nieuwe structuur en het toevoegen van taken en projecten aan die nieuwe structuur moesten er modules aangepast, herschreven en nieuw gemaakt worden.

De output van deze modules is een XML-document, dat ik daarna moest omzetten naar HTML. Dit omzetten gebeurt met XSL. Aan de XSL die zorgt dat tickets op een mooie manier in HTML worden gerepresenteerd moest weinig worden aangepast, want aan een ticket was weinig veranderd. Maar de XSL die zorgt dat projecten en taken op een mooie manier in HTML worden gerepresenteerd was er nog niet. Dus die moest ik gaan maken. Daarmee kwam ik voor een keuze te staan. Ofwel de XSL van de tickets gebruiken en aanpassen voor projecten en taken, ofwel een XSL bestand maken dat zowel tickets als projecten en taken van XML om kan zetten naar HTML.

Voor dat laatste heb ik toen gekozen. Op die manier werd mijn XSL DRY, in plaats van een product van knippen en plakken. Daardoor heb ik heel goed XSL leren begrijpen en leren schrijven. Tevens heb ik een waardevolle bijdrage geleverd aan het framework. Want op het moment dat er meerdere databaserecords op een pagina in een tabel moeten worden weergegeven kan nu iedere programmeur gebruik maken van mijn XSL. De opmaak van die tabel wordt dan verder geregeld in een extern CSS-bestand.



Tijdens de uitvoering van deze pilot heb ik ook nog de volgende werkzaamheden verricht: een javascript voor het direct bewerken van taken geschreven en een javascript voor het asynchroon opvragen van de formulieren behorende bij projecten en tickets gemaakt. Voor deze scripts heb ik geen gebruik gemaakt van JSON, maar van de mogelijkheid om HTML op te vragen. Dit was een betere oplossing omdat iedere keer dat er binnen dit script gegevens asynchroon werden opgevraagd het ging om één stuk van de pagina dat moest worden aangepast. Daardoor hoeft er met de HTML niets meer te gebeuren behalve kopiëren en dat is makkelijker dan het verwerken van JSON.

Deze pilot ging op een gegeven moment uitlopen, mede doordat ik de omvang ervan niet goed bepaald had. Er kwamen steeds nieuwe verzoeken bij. Daarom heb ik ondersteuning gevraagd en gekregen. De verzoeken die nog open stonden zijn door mij en een andere persoon uitgevoerd. Nieuwe verzoeken, omdat dingen niet goed werkte of omdat men het anders wilde, zijn meer niet op mijn bordje terecht gekomen. Tevens heb ik hulp gevraagd bij de conversie van de database gegevens uit de oude database naar de nieuwe database, omdat deze conversie moeilijk is, maar wel goed moet gebeuren.

5.4.2 JSON

De keuze voor JSON in plaats van XML-documenten heb ik al uitgelegd. Met de kleinere JSON documenten win je snelheid, zeker als de gegevens over een netwerk moeten. Daarnaast is JSON gemakkelijker in Javascript te verwerken dan XML en in de agenda moeten meerdere items op hetzelfde moment worden aangepast. Dus er moest JSON uit het framework komen. Uit overleg met mijn collega's kwam naar voren dat er twee manieren waren om JSON te verkrijgen. Eén manier was om de XML, die het framework maakt, met XSL om te zetten naar JSON. Een andere manier was om een stap eerder, nog voordat het framework XML heeft geproduceerd, de stap naar JSON te maken.

Het eerste was voor mij eenvoudiger dan het tweede, maar toch heb ik gekozen om de tweede optie uit te voeren. Met name omdat daar in minder stappen hetzelfde resultaat wordt bereikt. Minder stappen betekent dat er minder plaatsen zijn waar het fout kan gaan. Minder stappen betekent ook sneller en aangezien de agenda veel gegevens zou moeten kunnen bevatten is snelheid gewenst. Om een voorbeeld te noemen: een werkweekoverzicht met bijvoorbeeld per half uur een planning-item en een registratie-item op werkdagen betekent toch al 160 databaserecords die op de pagina moeten komen.



Om in een eerder stadium het framework JSON te laten maken moest ik aanpassingen maken in de generator van Propel. Propel is een Object Relational Mapping (ORM) framework voor PHP5, waar Connectholland binnen het framework gebruik van maakt. Hiermee kunnen gegevens uit een relationele database als objecten beschikbaar worden gemaakt binnen PHP5. Voordat dit kan moeten er eerst vanuit het databaseschema PHP Classes worden gegenereerd die onder andere de conversie van database records naar PHP objecten doen. Met die gegenereerde PHP Classes kan Propel deze objecten maken en worden er aan die objecten functionaliteit (methods) toegevoegd. Daarnaast kan je als je dat wilt aanpassingen maken aan de scripts die de PHP Classes genereren, waardoor je je eigen methods aan de door Propel gemaakte Classes kan toevoegen. Dit was al gedaan om toe te voegen dat deze objecten omgezet kunnen worden naar een XML-representatie. Wat ik moest doen was het mogelijk maken dat deze objecten omgezet kunnen worden naar JSON. Daarvoor moest ik de generator van Propel aanpassen.

De bovenstaande aanpassingen heb ik aangebracht. Daarbij kon ik een beetje afkijken bij de method die het mogelijk maakt om de Propel objecten om te zetten naar XML. Wat mijn werk eenvoudiger heeft gemaakt.

Het framework van Connectholland is een modulaair systeem geschreven in PHP, waarin bij elke website uitbreidingen kunnen worden gemaakt op die modules. Ik had voor mijn agenda een dergelijke uitbreiding kunnen maken. Met deze uitbreiding zou ik dan het stuk code waarin normaalgesproken XML wordt gemaakt overschrijven en vervangen voor code dat JSON maakt. Dit is een eenvoudige en snel te maken oplossing, maar er wordt dan een belangrijk punt over het hoofd gezien. Het feit dat er door mij mogelijk is gemaakt dat er JSON wordt geproduceerd door het framework is natuurlijk niet alleen te gebruiken voor mijn agenda. Er zijn andere toepassingen te bedenken. Als die toepassingen bestaan uit modules die zowel JSON als XML moeten kunnen maken, dan moeten er in al die modules uitzonderingen worden gemaakt. Het zou handiger zijn als er maar één kleine aanpassing gemaakt moet worden om JSON te maken in plaats van XML of beide tegelijkertijd.



Net op het moment dat ik met dit probleem worstelde kwam er een oplossing. Er was binnen het framework net een mogelijkheid gemaakt om plugins te maken. Een plugin binnen het framework is een laag bovenop het framework Windmill waarin allerlei Classes en methods die standaard uitgevoerd worden kunnen worden overschreven. Op die manier kan je dan dus een andere 'uitvoering' van het framework Windmill maken. Dus bijvoorbeeld een uitvoering die gewone tekst aan de gebruiker verstuurt. Maar dus ook een uitvoering die JSON aan de gebruiker verstuurt. Dit kon ik mooi gebruiken om in een vroegtijdig stadium de Propel objecten om te zetten naar JSON. Helaas lukte mij dat niet, omdat ik niet kon vinden wat ik allemaal precies moest overschrijven in het framework Windmill. Degene die mij daarmee kon helpen was op dat moment op vakantie. Daarom heb ik voor mijn agenda module niet die plugin geschreven, maar de voor mij eenvoudige snelle oplossing toegepast. Die plugin kan dan ook later worden gemaakt. De enige voorwaarde is dan dat de manier waarop de gegevens uit de agenda module komen precies hetzelfde uit het framework Windmill komen, op het moment dat de plugin wel is gemaakt. In het geval dat niet aan deze voorwaarde kan worden voldaan zal de agenda moeten worden aangepast.

Voor het verwerken van de gegevens in het Javascript dat alle interactie met de gebruiker regelt heb ik gebruik gemaakt van de Javascript API Prototype. Met deze API wordt het veel eenvoudiger om met Javascript te werken. Zo zorgt deze API er bijvoorbeeld voor dat sommige functies die Javascript wel zouden moeten werken volgens het Document Object Model, maar dat niet doen in een bepaalde internet browser, worden uitgevoerd volgens de niet standaard manier van die internet browser op het moment dat jij in je script wel de standaard manier gebruikt. Hierdoor hoeft je geen uitzonderingen te maken voor bepaalde internet browsers. Daarnaast voegt deze API extra mogelijkheden toe om dingen te doen met een HTML-pagina, die niet standaard in Javascript tot je beschikking staan. Je zou ze wel zelf kunnen maken, maar in dit geval heeft iemand dat al voor je gedaan. Het opvragen van gegevens van de server via de AJAX methode wordt ook vereenvoudigd door deze API. Daarom wordt hiervan bij Connectholland in veel gevallen gebruik gemaakt. Voor mij was dat in dit geval ook de beste keuze, want het gebruik ervan zou en heeft me veel werk uit handen genomen.

Het Javascript dat alle interactie met de gebruiker regelt vraagt op het moment dat de gebruiker de agenda bezoekt alle agendagegevens op die bij de huidige week horen. Op het moment dat die gegevens aankomen worden ze in het overzicht geplaatst en wordt er geregeld dat de gebruiker er dingen mee kan doen, zoals ze verplaatsen of aanpassen. Het Javascript dat dat regelt heb ik ontwikkeld in de pilot Frontend van mijn ontwikkelfase.



5.4.3 Frontend

Om een weekoverzicht te maken met afspraken heb ik de volgende dingen gedaan. Ik heb een module binnen het framework Windmill gemaakt, waarbij allerlei dingen over het overzicht in te stellen zijn.

Deze dingen zijn:

- het aantal dagen in het overzicht
bijvoorbeeld alleen werkdagen of een hele week
- de starttijd van de dag
bijvoorbeeld alles vanaf 8 uur 's morgens
- de eindtijd van de dag
bijvoorbeeld alles tot 18 uur 's avonds
- de begindag van de week
bijvoorbeeld maandag of zondag
- het aantal delen waarin een uur wordt opgedeeld
bijvoorbeeld per kwartier of per half uur

Deze instellingen komen terecht in een XML-document. Daarna heb ik een XSL-bestand geschreven die een tabel maakt aan de hand van dat XML-document. De aanpak met in te stellen waardes voor het overzicht heb ik gekozen omdat het mogelijk moet zijn om een ander soort overzicht te kunnen kiezen, want er staat niet vast dat het overzicht in een andere website hetzelfde moet zijn als in Aeolus. In Figuur 11 is een voorbeeld weergegeven van een weekoverzicht met de volgende instellingen:

- aantal dagen: 7
- starttijd: 8 uur
- eindtijd: 18 uur
- begindag van de week: maandag
- aantal delen van een uur: 2



Overzicht: 15 januari 2007 – 21 januari 2007

	Maandag 15/1	Dinsdag 16/1	Woensdag 17/1	Donderdag 18/1	Vrijdag 19/1	Zaterdag 20/1	Zondag 21/1
8u							
9u							
10u							
11u							
12u							
13u							
14u							
15u							
16u							
17u							

Figuur 11: Voorbeeld van een weekoverzicht

De keuze om het maken van het overzicht op deze manier aan te pakken heb ik gemaakt omdat alle HTML bij Connectholland wordt gemaakt met behulp van XSL. Hierdoor is dit stukje voor doorontwikkeling geschikt. Ik heb de XSL ook zo geschreven dat deze overschreven kan worden door andere XSL. Hierdoor zijn er nog wijzigingen te maken op het moment dat de agenda bijvoorbeeld in een andere website wordt geplaatst.

In het voorbeeld in Figuur 11 zijn 140 cellen die een tijdstip voorstellen. Zeven dagtitels en een titel met de begindatum en de einddatum van het overzicht. De informatie over het tijdstip dat een cel voor moet stellen is opgeslagen op de achtergrond. In het voorbeeld zijn dus in het totaal 150 variabele gegevens geplaatst. Aan de hand van die gegevens dienen de planning- en registratie-items in het overzicht geplaatst te worden nadat ze zijn opgevraagd.

Eerder heb ik al genoemd dat het best mogelijk zou zijn dat in een dergelijk overzicht veel van deze gegevens als afspraken geplaatst moeten worden. In dit voorbeeld kunnen 140 afspraken worden geplaatst als er maar één afspraak per cel wordt geplaatst. Daarmee ontstaat een probleem op het moment dat bijvoorbeeld de volgende week moet worden getoond. Het zou dan dus kunnen voorkomen dat er 140 afspraken moeten worden verwijderd, 150 variabele gegevens aangepast moeten worden in het overzicht en weer 140 nieuwe afspraken geplaatst moet worden. Dit moet allemaal worden uitgerekend door de computer van de gebruiker en zou er toe kunnen leiden dat de gebruiker moet gaan wachten totdat deze berekening klaar is.



Voor dit mogelijke probleem heb ik een oplossing bedacht. Ik heb één spil gemaakt waar alles in het overzicht om draait, dit is het absolute begin van de eerste dag in het overzicht. In het voorbeeld is dat maandag 15 januari om 0 uur. De informatie over het tijdstip dat een cel moet voorstellen heb ik relatief gemaakt aan deze spil. Op het moment dat er een nieuw overzicht wordt opgevraagd moet dus alleen die spil worden gewijzigd, de zeven dagtitels en de overzichtstitel. Daarnaast wordt er gezorgd dat de afspraken die op het volgende of vorige overzicht moeten staan van tevoren opgevraagd zijn. Voor deze afspraken wordt dan dat relatieve tijdstip uitgerekend, hierdoor moeten de oude afspraken alleen maar worden verwijderd en de afspraken die klaar staan worden geplaatst op het moment dat de gebruiker een nieuw overzicht opvraagt. Daarna wordt er voor elke afspraak die is geplaatst vastgelegd wat de gebruiker ermee kan doen.

De gebruiker kan een afspraak op zijn agenda verplaatsen naar een andere dag of de tijdstippen van een afspraak aanpassen door middel van drag-and-drop functionaliteiten. Of de gebruiker kan een formulier opvragen waarmee alle gegevens van een afspraak aanpasbaar zijn. Waardoor een afspraak bijvoorbeeld verplaatst zou kunnen worden naar een andere week.



6. Evaluatie

6.1 Inleiding

In deze fase wordt het verloop van het project onder de loep genomen en wordt hetgeen ontwikkeld is onderzocht aan de hand van testen met de gebruiker.

6.2 Procesevaluatie

In deze paragraaf wordt mijn procesgang geëvalueerd. Waarom heb ik bepaalde keuzes gemaakt en hoe ben ik tot die keuze gekomen. Ik zal dit aan de hand van een aantal vragen beantwoorden.

Deze vragen zijn:

- Waarom heb ik de dingen zo gedaan als ik ze heb gedaan?
- Waarover ben ik tevreden of ontevreden als het gaat om hoe ik dit proces heb doorlopen?
- Waarom ben ik tevreden of ontevreden en als ik ontevreden ben hoe zou ik het dan volgende keer aanpakken?

De fasering die ik heb aangebracht in mijn project was voor mij heel handig. Ik heb dit gedaan omdat ik geen goede procesbewaker ben. Dus ik had een duidelijke fasering nodig zodat ik een houvast had binnen het project. Mede door die fasering is mijn project goed verlopen. Ik had echter een strakkere planning moeten maken aan het begin van het project om de fasering beter af te bakenen. Dat ik dit niet heb gedaan heeft er mede toe geleid dat de eerste pilot is uitgelopen.

Als je geen goede procesbewaker bent is dat problematisch als je wel perfectionistisch bent ingesteld. Er valt altijd wel iets te verbeteren aan wat je hebt gemaakt en daar kan je tot in het oneindige mee doorgaan. Tenzij je op tijd ziet dat deze instelling tegen de voortgang van je proces werkt. Ik heb dit bij het ontwikkelen van de eerste pilot te laat gezien, maar ik heb daar lering uit getrokken en gezorgd dat dit bij de rest van het project niet meer is gebeurd. De zaken die nog niet zijn afgerond van pilot 3 stonden ook zo gepland.

Over de ontwikkelingsfase ben ik zeer tevreden, ondanks dat ik niet alles tot in de puntjes had vastgelegd is deze fase toch zeer voorspoedig verlopen. Ik ben duidelijk meer een doener dan een denker. Onder het doen bedenk ik wel oplossingen voor problemen die ik onderweg tegen kom. Je kunt toch niet van te voren alles bedenken. Achteraf gezien heb ik te inefficiënt tijd besteed aan de fases voor de uitvoering. Met ook nog eens te weinig resultaat.



De volgende keer dat ik in een software ontwikkel project aan de slag ga zal ik de volgende dingen anders doen. Ik zal meer gaan overleggen met andere mensen in het beginstadium. Zij kunnen mij dan aanvullen in de gebreken die ik heb in het maken van goede concrete plannen en ik kan daar dan van leren. Daarnaast zal ik eerder van start gaan met het daadwerkelijke ontwikkelen, doordat ik zorg dat ik wel efficiënt met mijn tijd om ga tijdens het maken van plannen.

6.3 *Productevaluatie*

In deze paragraaf worden de resultaten van mijn project geëvalueerd. Wat is er goed of slecht aan de producten die ik heb gemaakt en waarom is dat zo. Welke suggesties ter verbetering kan ik bedenken.

Zoals ik al aangaf in de procesevaluatie ben ik niet een goede plannen maker. Door dit gebrek vind ik de waarde van mijn plannen ook niet erg groot. De definitiestudie is veel te beknopt geweest en het pilotplan had veel gedetailleerder gekund. Dan waren de pilots waarschijnlijk veel kleiner en meer behapbaar geworden en was het makkelijker geweest om een gedetailleerde planning te maken.

Wel ben ik tevreden over de producten van de ontwerpfase. Deze zijn juist handiger als ze beknopt zijn. En aangezien er aan dit project weinig strenge eisen zaten was het ook niet nodig om heel gedetailleerd elk element op de pagina en elke mogelijke actie vast te leggen.

Als productevaluatie van de nieuwe structuur in Aeolus heb ik een groepsgesprek georganiseerd waarin ten eerste door mij het idee achter de nieuwe structuur werd uitgelegd aan de hand van een voorbeeld. Daarna konden alle deelnemers vragen stellen over de dingen die ze niet begrepen. Dit was een informeel gesprek waarbij ik aantekeningen heb gemaakt van de opmerkingen die men gaf, deze zijn verwerkt in de plannen voor de doorontwikkeling door iemand anders. In de week dat de nieuwe structuur werd ingevoerd heb ik alleen maar positieve reacties gekregen. Iedereen was zeer enthousiast over de nieuwe structuur. Project- en accountmanagers waren blij dat ze de gegevens die ze nodig hadden uit het systeem konden krijgen en de rest was blij dat ze een overzicht hadden met taken die ze konden gaan doen. Als de agenda echt af is zal ik wederom aan de hand van een groepsgesprek proberen informatie te verkrijgen over de kwaliteit ervan.



Ik denk dat de kwaliteit van de producten die ik tot nu toe heb gemaakt in de ontwikkelfase hoog is, doordat er goed is nagedacht over wat ik heb gemaakt, het zit naar mijn idee logisch en eenvoudig in elkaar. Het is open geprogrammeerd en uitbreidbaar gemaakt. Daarmee wordt doorontwikkeling en het toepassen in een ander systeem vergemakkelijkt.

Het is jammer dat ik mijn producten niet echt heb getest met eindgebruikers. Dit heb ik voornamelijk niet gedaan omdat het een intern project is, waardoor het gebruik ervan het testen is geworden. Er kan in een directe lijn feedback aan mij worden gegeven en dan kan het product alsnog worden aangepast. De agenda is bovendien nog niet te testen met eindgebruikers, omdat hij op dit moment nog niet klaar is voor gebruik.

Alles bij elkaar ben ik trots op wat ik heb gemaakt. Ik denk dat het een waardevolle bijdrage is geweest aan de ontwikkeling van het bedrijf en de ontwikkeling van mij persoonlijk.



7. Bijlagen

7.1 Geraadpleegde literatuur

Beginning Ajax with PHP: From Novice to Professional

Lee Babin, Apress, New York, 2007 (ISBN 1-59059-667-6)

Javascript Phrasebook

Christian Wenz, Sams Publishing, Indianapolis, 2006 (ISBN 0-672-32880-1)

XSLT 2.0 Programmer's Reference (3rd edition)

Michael Kay, Wiley Publishing Inc., 2004 (ISBN 0-764-56909-0)

The Element of User Experience: User-Centered Design for the Web

Jesse James Garreth, New Riders, Indianapolis, 2003 (ISBN 0-7357-1202-6)

UML in 24 uur

Joseph Schmuller, Sams Publishing, Indianapolis, 1999 (ISBN 90-395-1344-9)

IAD Het evolutionair ontwikkelen van informatiesystemen

R.J.H. Tolido, Acedemic Service, Schoonhoven, 1996 (ISBN 90-395-0401-6)



7.2 Afkortingenlijst

Basis van de hier volgende informatie is afkomstig van www.wikipedia.org en aangepast aan de toepassing van de afkortingen binnen dit verslag.

IETF: Internet Engineering Task Force

De Internet Engineering Task Force is een grote, open, internationale gemeenschap van netwerkontwerpers, - operators, -leveranciers en -onderzoekers die zich bezig houdt met de evolutie van de internetarchitectuur en de soepele werking van het internet. Deelname staat open voor alle geïnteresseerden.

AJAX: Asynchronous Javascript And XML

AJAX is een term voor het ontwerp van interactieve webpagina's waarin asynchroon gevraagde gegevens worden opgehaald van de webserver. Daardoor hoeven dergelijke pagina's niet in hun geheel verversd te worden. Een dergelijke pagina is te vergelijken met een applicatie die in de internet browser draait. De term is op 18 februari 2005 door Jesse James Garrett gelanceerd.

API: Application Programming Interface

Een Application Programming Interface is een verzameling definities op basis waarvan een computerprogramma kan communiceren met een ander programma of onderdeel, meestal in de vorm van bibliotheken. Vaak vormen API's de scheiding tussen verschillende lagen van abstractie, zodat applicaties op een hoog niveau van abstractie kunnen werken en het minder abstracte werk uitbesteden aan andere API's.

CSS: Cascading Style Sheets

Cascading Style Sheets is een techniek voor de stijl (vormgeving) van webpagina's en is een W3C-specificatie (zie W3C). De informatie over de vormgeving wordt toegevoegd aan de HTML-code van het document. Die informatie kan in het document zelf staan, maar ook in een extern document dat wordt geïmporteerd. Een dergelijk apart geïmporteerd document wordt ook wel stylesheet genoemd. Een stylesheet biedt de mogelijkheid inhoud en vormgeving van een document van elkaar te scheiden en op die manier een consistente vormgeving over meerdere documenten te bereiken.

DRY: Don't Repeat Yourself

De filosofie achter DRY benadrukt dat informatie niet zou moeten worden gedupliceerd, omdat de verdubbeling moeilijkheden die samenhangen met veranderingen verhoogt, de duidelijkheid kan verminderen, en leidt tot een grotere kans op inconsistentie.



DOM: Document Object Model

Het Document Object Model is een objectgeoriënteerde benadering van gestructureerde documenten zoals HTML-, XHTML- en XML-documenten. Dit is een W3C-specificatie (zie W3C). Een essentiële voorwaarde voor interactiviteit van (X)HTML-documenten is dat de onderdelen (en de eigenschappen van elk onderdeel) van het document afzonderlijk benaderd kunnen worden. Alleen op deze manier kunnen de eigenschappen van dat onderdeel aangepast worden.

IAD: Iterative Application Design

Dit is een ontwikkelmodel beschreven in het boek "IAD, het evolutionair ontwikkelen van informatiesystemen". Het beschrijft een model voor het op een iteratieve manier ontwikkelen van informatiesystemen. IAD propageert een cyclische aanpak waarin het informatiesysteem tot stand komt via een aantal snel uitgevoerde iteraties van de complete ontwikkelcyclus.

IMC: Internet Mail Consortium

Dit is een consortium dat informatie verschaft over alle internet email standaarden en technologieën. Door dit consortium worden ook rapporten voorbereidt ter toevoeging aan de Request for Comments (zie RFC) van de Internet Engineering Task Force (zie IETF).

ISO: International Organization for Standardization

De International Organization for Standardization is een internationale organisatie die normen vaststelt. De organisatie is een samenwerkingsverband van nationale standaardisatieorganisaties in 156 landen, de Nederlandse 'dochter' is het NEN (Nederlands Normalisatie-instituut).

JSON: JavaScript Object Notation

Dit is een subset van de programmeertaal Javascript. Het wordt gebruikt voor het uitwisselen van datastructuren, met name in webapplicaties die asynchroon gegevens ophalen van de webserver (zie Ajax).

ORM: Object-Relational Mapping

Dit is een programmeertechniek om gegevens tussen onverenigbare typesystemen in databases en objectgeoriënteerde programmeertalen om te zetten. Daarmee wordt een "virtuele objecten database" gecreëerd dat dan binnen de object objectgeoriënteerde programmeertaal kan worden gebruikt.



PHP 5: PHP: Hypertext Preprocessor (versie 5)

PHP is een scripttaal, die bedoeld is om op webserver's dynamische webpagina's te creëren. PHP is in 1994 ontworpen door Rasmus Lerdorf, een senior software engineer bij IBM. Destijds was de taal duidelijk geïnspireerd door Larry Walls Perl. Met name sinds versie 5 is de mogelijkheid om in PHP objectgeoriënteerd te programmeren sterk verbeterd. De naam is een recursief acroniem.

RFC: Request for Comments

Dit zijn documenten waarin nieuwe standaarden, methodes, innovaties en technologieën met betrekking tot het internet worden voorgesteld of beschreven. Eenieder mag een RFC document maken en eenieder mag erop reageren. De Internet Engineering Task Force (zie IETF) neemt sommige van deze voorstellen over als internet standaard.

W3C: World Wide Web Consortium

Het World Wide Web Consortium is een organisatie die de webstandaarden voor het World Wide Web ontwerpt, zoals HTML, XHTML, XML en CSS. Het wordt geleid door Tim Berners-Lee, de originele bedenker van het HTTP protocol en HTML, waar het Web oorspronkelijk en nog steeds grotendeels op gebaseerd is.

XML: eXtensible Markup Language

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

XSLT: eXtensible Stylesheet Language Transformations

XSLT is een standaard voor het omzetten van de informatie in een XML-document (zie XML) naar een ander formaat, of een anders gestructureerd XML-document, zoals XHTML (zie (X)HTML) of nieuwe XSL, aangezien dit zelfs ook uit XML bestaat.

(X)HTML: (eXtended) HyperText Markup Language

HyperText Markup Language is een taal voor de opmaak van documenten. HTML wordt vooral gebruikt op het internet, om webpagina's te tonen. XHTML is een taal die de functionaliteit heeft van HTML, maar een striktere syntaxis.



7.3 Relevante internetadressen

Connectholland

<http://www.connectholland.nl/>

Google Calendar

<http://calendar.google.com/>

iCalendar Specificaties

<http://www.ietf.org/rfc/rfc2445.txt>

<http://www.kanzaki.com/docs/ical/>

PHP

<http://www.php.net/>

Prototype API

<http://www.prototypejs.org/>

Scriptaculous Javascript Library

<http://script.aculo.us/>

W3C

<http://www.w3.org/>

XSL specificaties: <http://www.w3.org/TR/xsl/>

CSS specificaties: <http://www.w3.org/TR/REC-CSS2/>

XML specificaties: <http://www.w3.org/TR/xml/>

HTML specificaties: <http://www.w3.org/TR/html401/>



7.4 Concept grafisch ontwerp agenda

connect holland aeolus

Home

Projecten

Tickets

Agenda

Rapportage

Ingelogd als: Reyo Stallenberg | Uithloggen

AGENDA OPTIES

Eigenaar

ron

Zoeken leegmaken

KALENDER

<

Oktober 2006

>

M

D

W

D

V

Z

Z

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

1

2

3

4

5

6

7

8

9

10

Planning

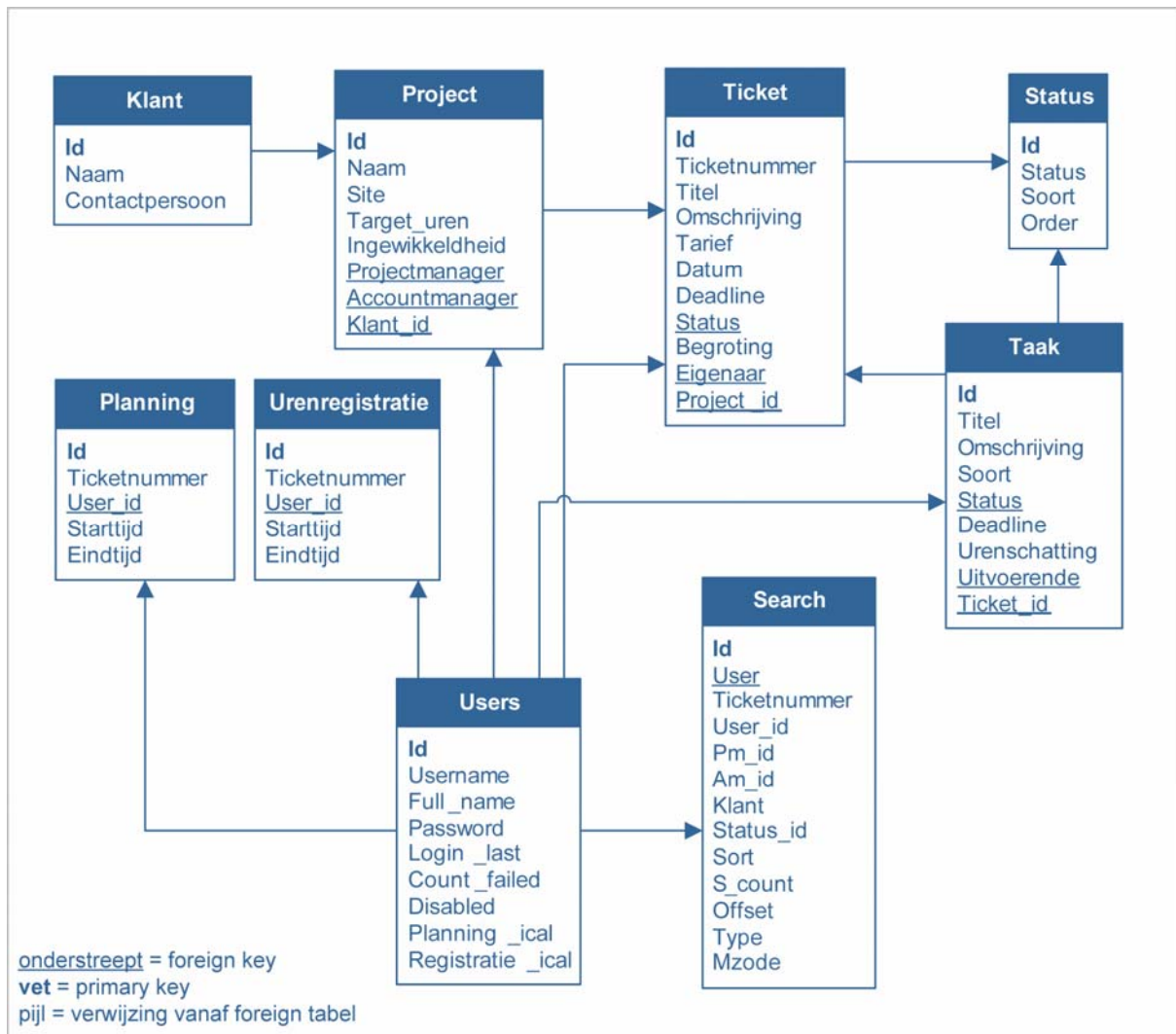
Registratie

OVERZICHT

Tijd	Maandag 23/10	Dinsdag 24/10	Woensdag 25/10	Donderdag 26/10	Vrijdag 27/10
09:00	T2006100006 Taak 1	T2006100006 Taak 5	T2006100006 Taak 10	T2006100012 Taak 12	T2006100032 Taak 2
10:00					
11:00					
12:00	T2006100007 Taak 1		T2006100009 Taak 4		
13:00					
14:00	T2006100008 Taak 8	T2006100010 Taak 4	T2006100016 Taak 3	T2006100015 Taak 1	T2006100032 Taak 6
15:00			T2006100025 Taak 6		
16:00					
17:00					



7.5 Database ontwerp nieuwe structuur



7.6 Definitiestudie

Definitiestudie

Aeolus Agenda



Inhoud

1. Inleiding	3
2. Plan van Aanpak	4
2.1. Inhoud en omvang.....	4
2.2. Aanpak.....	4
2.3. Projectinrichting	6
2.4. Kwaliteitsborging	6
2.5. Gedetailleerde planning	7
3. Systeemeisen	8
4. Systeemconcept	11
5. Technische voorzieningen.....	12
6. Verdere ontwikkeling.....	14



1. Inleiding

Deze definitiestudie gaat over het ontwikkelen van een Agenda module binnen het Windmill Framework van Connectholland. Deze Agenda module moet dan toegepast worden op het projectmanagementsysteem Aeolus. Het is de bedoeling dat er in dit projectmanagementsysteem planning en registratie van uren mogelijk wordt. Dit gebeurt nu door deze registratie plaats te laten vinden in Google Calendar (zie <http://calendar.google.com>) en deze informatie daarna te synchroniseren met de database achter Aeolus, deze situatie is niet gewenst. Mede doordat het niet zeker is of deze dienst van Google kostenvrij zal blijven.

In de fase definitiestudie, die binnen Iterative Application Development (IAD) valt, worden de doelen van de agenda module geanalyseerd. Hierdoor komen eisen evenals beperkingen naar boven. Van tevoren moet er dus gekeken worden naar het systeem. Deze definitiestudie beschrijft het plan van aanpak, de systeemeisen, het systeemconcept, de technische voorzieningen en de pilotindeling. De definitiestudie is, evenals alle andere onderdelen van het IAD ontwikkelmodel, iteratief van aard, waardoor het kan zijn dat er bijstellingen plaats vinden na een ontwikkelcyclus. Maar het zal waarschijnlijk niet een enorm iteratieve ontwikkeling worden. In ieder geval wordt hiermee de mogelijkheid om latere aanpassingen en toevoegingen te doen open gehouden. Maar Connectholland is wat structuur betreft niet ingericht op het ontwikkelen via de weg van IAD, waardoor het niet persé nodig is om erg uitgebreide documentatie vast te leggen om een nieuwe iteratie slag te kunnen maken. In ieder geval wordt hiermee de mogelijkheid om later aanpassingen en toevoegingen te doen open gehouden.



2. Plan van Aanpak

In het plan van aanpak worden de inhoud en omvang van de opdracht duidelijk en wordt er bepaald wat de uit te voeren werkzaamheden zijn om de opdracht tot een goed eind te brengen. Tevens wordt beschreven wat de personele bezetting is, welke middelen er zijn en hoe de financiën worden beheerst. Welke mogelijkheden er zijn om componenten her te gebruiken of herbruikbaar te maken. De reikwijdte van het project en de te verwachten resultaten worden vastgelegd.

Volgens de structuur van IAD dienen in dit hoofdstuk de inhoud, de omvang, de aanpak, de projectinrichting en de kwaliteitsborging van dit project weergegeven te worden. Maar omdat deze zaken in de opdrachtschrijving reeds zijn vastgesteld en omdat de gegevens uit die opdrachtschrijving vrijwel geheel overgenomen zullen worden in het eindverslag, zullen in deze definitiestudie slechts relevante aanvullingen op de opdrachtschrijving worden opgenomen.

2.1. Inhoud en omvang

Geen aanvullingen

2.2. Aanpak

Standaards, richtlijnen en procedures

Als standaard en richtlijn voor de code geldt de Quality Assurance (QA) van Connectholland. Dit is een richtlijn voor de opmaak en indeling van de code. Deze heeft tot doel om code leesbaar en onderhoudbaar te maken.

Fasering

De indeling van de fasering van dit project is tot stand gekomen met behulp van de richtlijnen beschreven in het boek *"The Elements of User Experience"*.

1. Oriëntatiefase

In deze fase wordt er uitgebreid bekeken wat er nodig is om dit project te laten slagen. Wat bestaat er reeds en moet worden uitgebreid? Wat moet er nieuw ontwikkeld worden? En welke bronnen zijn daarvoor nodig.

2. Analysefase

In deze fase wordt er gekeken wat de strategie wordt achter dit project en welke omvang het zal hebben. Op basis van die gegevens wordt de planning aangescherpt.

2.1 Strategie

In dit onderdeel zullen de doelen van de applicatie vastgesteld worden en de gebruikersbehoefte bepaald worden.

2.2 Omvang

In dit onderdeel worden de functionele specificaties vastgesteld.



3. Ontwerpfase

In deze fase wordt aan de hand van diagrammen bepaald wat de structuur is van de agenda en het achterliggende skelet.

3.1 Structuur

De structuur van de applicatie zal worden vastgelegd in een interactie ontwerp, dat met behulp van UML verduidelijkt zal worden.

3.2 Skelet

Van belang is hoe, bijv. in welke volgorde, de in de applicatie beschikbare informatie gepresenteerd zal worden. Daartoe zal een informatie ontwerp gemaakt worden. Tevens is van belang welke elementen er gepresenteerd worden. Daartoe zal een interface ontwerp gemaakt worden.

3.3 Voorkant

Op basis van de structuur en het skelet zal een visueel ontwerp gemaakt moeten worden.

4. Ontwikkelfase

In deze fase worden de in de eerdere fases vastgelegde stappen uitgevoerd.

5. Evaluatiefase

In deze fase wordt hetgeen ontwikkeld is onderzocht aan de hand van testen met de uiteindelijke gebruikers.

Tevens wordt het verloop van het project onder de loep genomen.

5.1 Gebruikerstesten

De gebruikerstesten zullen niet zeer uitgebreid zijn. Mede doordat dit een intern project is, waardoor de eindgebruikers gemakkelijk tijdens het gebruiken hun feedback op het systeem kunnen geven. Het zal neerkomen op een workshop waarin het product en de werking ervan kan worden toegelicht en waar iedereen zijn mening en dergelijke kwijt kan.

5.2 Projectverloop

Er zal uitgebreid worden gekeken naar de gang van het project en de communicatie er omheen.

Activiteiten

In dit gedeelte worden de activiteiten die bij elke fase horen gedetailleerder beschreven.

Planning van de pilots

- Pilot Project, Ticket, Taak structuur
Week 6 t/m 10
- Pilot JSON
Week 11 t/m 13
- Pilot Frontend
Week 14 t/m 16



Hergebruik

Er wordt op een dusdanige manier ontwikkeld dat alle geschreven code in losse stukjes binnen het Framework her te gebruiken zijn. Mede doordat bij de indeling van de code rekening wordt gehouden met de structuur van het Framework en doordat er wordt voldaan aan de Quality Assurance van Connectholland.

2.3. Projectinrichting

Personele bezetting

Het hele project wordt door één persoon gerealiseerd. Met behulp van informatie afkomstig van de opdrachtgever en met behulp van kennis aanwezig op de afdeling development. Hiervoor zal tijd nodig zijn van allerlei mensen binnen het bedrijf om informatie te verschaffen, workshops te houden e.d.

Financiën

Financiën spelen nauwelijks, of geen rol, omdat het hier om het vervullen van een afstudeerstage gaat wordt met een stage vergoeding volstaan.

2.4. Kwaliteitsborging

Het gemaakte product zal kunnen worden getoetst op bruikbaarheid tijdens de acceptatieworkshop.

Doordat voor de ontwikkeling van het systeem de IAD methode wordt gevolgd, zal het eindproduct ook aan de eisen die deze methode stelt voldoen. Het product zal een hogere waarde hebben doordat de fasen iteratief doorlopen zullen worden.

Risicofactoren

Aan dit project zijn een aantal risico's verbonden. Deze zullen niet uitgebreid worden behandeld, omdat dit een tijdrovende analyse is. Er zal puntsgewijs een aantal risico's worden opgesomd, met mogelijke oplossingen.

Risico: *De hoeveelheid werk is te groot om de opleverdatum te halen.*
Oplossing: *In de voorbereidende fase van het project zullen (essentiële) onderdelen worden gekozen die wij in de applicatie stoppen en wanneer deze af moeten zijn, zogenaamde milestones. Als het goed is moeten de essentiële onderdelen sowieso op tijd af komen.*



2.5. Gedetailleerde planning

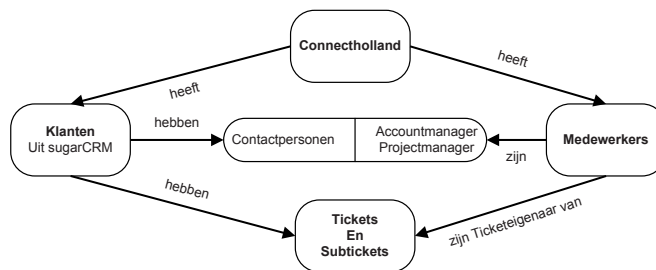
Datum	Fase	Pilot	Werkzaamheid
Week 1	1	0	Oriëntatie op de werking van het huidige systeem
Week 2	1	0	Plan van aanpak opstellen
Week 3	1	0	Fouten in het huidige systeem inventariseren
Week 4	1	0	Oriëntatie op de werking die het toekomstige systeem moet hebben
Week 5	2	0	Definitiestudie opstellen
Week 6	3	1	Database ontwerp maken voor nieuwe systeem
Week 7	4	1	Omzetten van gegevens uit oude database naar nieuwe
Week 8	4	1	Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
Week 9	4	1	Verbeteren van eventuele fouten
Week 10	4	1	Omzetten naar nieuwe structuur voor eindgebruiker
Week 11	4	2	Generator van Propel ombouwen
Week 12	4	2	Ontvangst van gegevens in javascript mogelijk maken
Week 13	4	2	Verwerken van ontvangen gegevens in javascript mogelijk maken
Week 14	4	3	Agendapagina maken waarop gegevens gepresenteerd kunnen worden
Week 15	4	3	Gegevens uit database presenteren op agenda pagina
Week 16	4	3	Gegevens aanpasbaar maken.

Fase 5 zal plaatsvinden in het eindverslag en voor sommige stukken zelfs na het afstuderen.



3. Systeemeisen

Om de systeemeisen te bepalen is er geen workshop gehouden. Er is een brainstorm gehouden met de opdrachtgever. Hieruit kwam naar voren dat de structuur van het huidige systeem ontoereikend was als structuur voor een projectmanagement systeem binnen Connectholland. En dat het tevens ongewenst was om gegevens van de planning en registratie aan die structuur te koppelen.



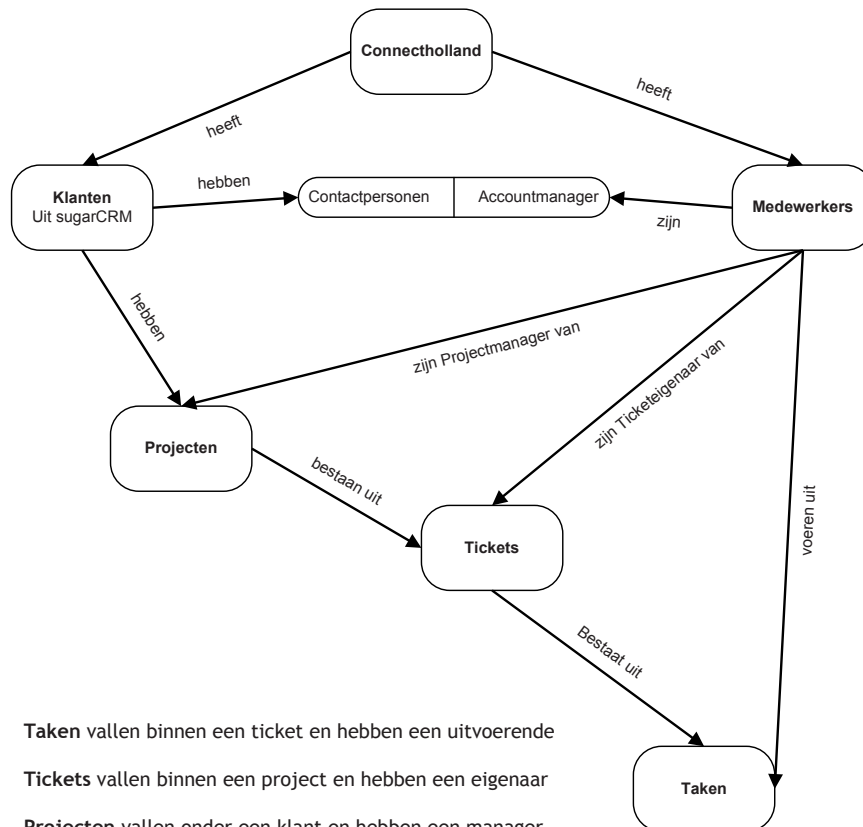
Tickets horen bij een klant en hebben een eigenaar

Klanten vallen onder Connectholland en hebben een eigen contactpersoon en een manager

Figuur 1: Huidige structuur



Daartoe is er een nieuwe structuur ontworpen.



Taken vallen binnen een ticket en hebben een uitvoerende

Tickets vallen binnen een project en hebben een eigenaar

Projecten vallen onder een klant en hebben een manager

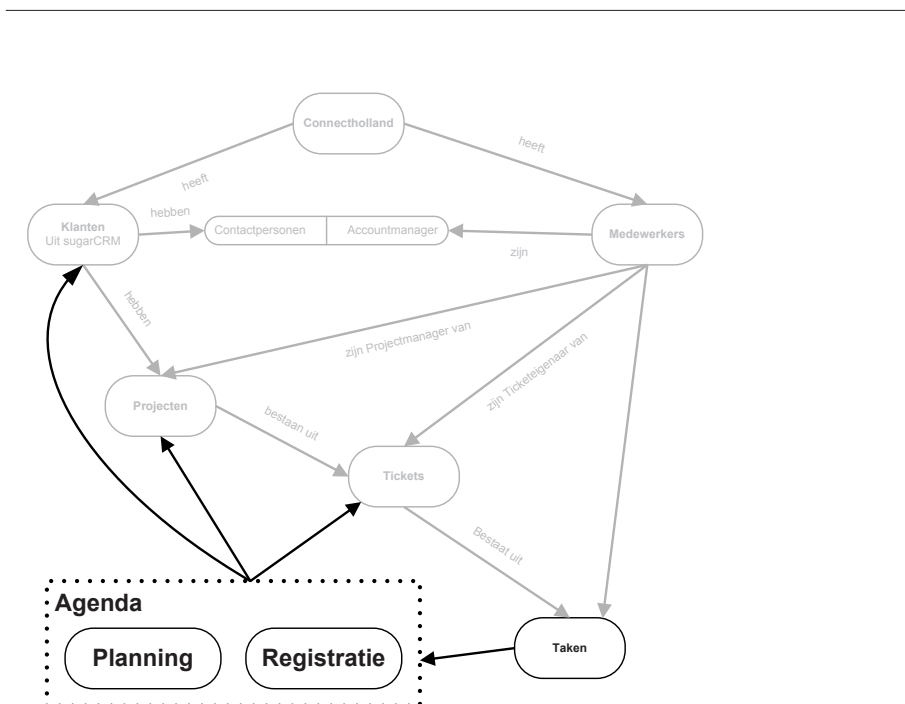
Klanten vallen onder Connectholland en hebben een eigen contactpersoon en een manager

Figuur 2: Nieuwe structuur

Conclusie

De opdrachtgever vindt het belangrijk dat de bovenstaande zaken in het systeem worden opgenomen. En dat de Agenda aan dit systeem wordt gekoppeld op de plaats zoals is weergegeven in Figuur 3 op de volgende pagina.





Figuur 3: Plaats van Agenda module binnen systeem



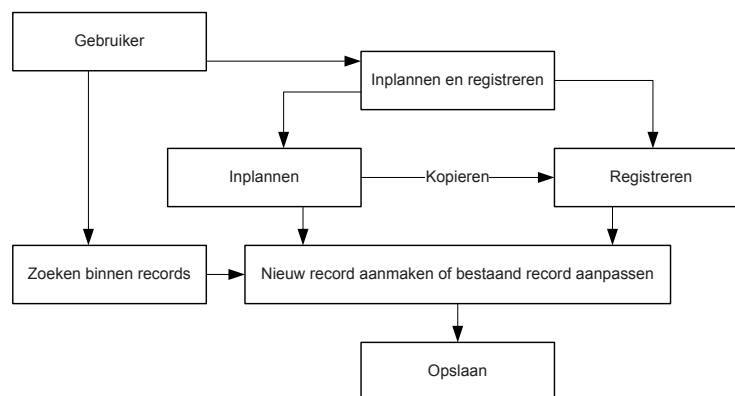
4. Systeemconcept

Taakanalyse

De taakanalyse houdt in dat de taken van de opdrachtgever/gebruikers worden weergegeven. Dit doen we door een beschrijving hiervan te maken en de problemen en oplossingen te vermelden die het toekomstige systeem gaat bieden. Hieronder vind u schematisch weergegeven hoe de relevante onderdelen eruit komen te zien.

Agenda

Schematische weergave van wat er binnen de agenda mogelijk moet zijn:



Figuur 4: Systeemschets



5. Technische voorzieningen

Technische architectuur

In het systeemconcept is er rekening mee gehouden dat er gewerkt zal worden met PHP, Javascript in combinatie met JSON en XML in combinatie met XSL. De realisatie van de Agenda is daar ook op gebaseerd.

Technologische basis

Bij het ontwikkelen van de website zal er mogelijk gebruik gemaakt worden van software zoals:

- Crimson editor
- TortoiseSVN
- Sugar CRM
- Mozilla Firefox
- Microsoft Internet Explorer
- PHP
- XML
- XSL
- Xpath
- CSS
- Javascript
- (X)HTML
- MySQL
- Propel
- Openoffice.org
- Apache webserver
- Windmill

De basis systeemeisen voor de gebruikers om de website te kunnen openen en te gebruiken zijn:

- Een computer,
- Een netwerkverbinding,
- Een webbrowser zoals Microsoft Internet Explorer of Mozilla Firefox.

Systeemconfiguratie

Om de agenda module doeltreffend en snel te laten werken is het nodig om snel gegevens uit de database op te kunnen vragen en deze daarna eenvoudig te verwerken. Aangezien de agenda module aan de voorkant werkt met javascript is het nodig om de gegevens uit de database terug te krijgen in JavaScript Object Notation (JSON). Derhalve dient het Framework zo te worden aangepast dat dit mogelijk wordt. Met JSON is het eenvoudiger om de gegevens uit de database uiteindelijk te verwerken in de view van de gebruiker.



Onderzoek haalbaarheid technische aspecten

Er is reeds een plugin systeem in het Framework aanwezig, waardoor het implementeren van een losse functionaliteit zoals het terugleveren van JSON zeer eenvoudig te implementeren is. Mede doordat het systeem nu al zo is ingericht dat er na een verzoek van de gebruiker gegevens worden teruggeleverd. De enige verandering is e representatie van die gegevens.

Definitie technische veranderingen

Er kunnen in een later stadium zeker veranderingen optreden, zowel in de database als op de frontend zelf. Mede doordat op dit moment niet duidelijk is hoe de database achter de module er uit moet zien voor een losstaande agenda, aangezien er nu wordt gewerkt met het doel om de agenda te implementeren in één systeem.

Vaststelling herbruikbare componenten

1. De agenda module moet los kunnen opereren binnen het Framework Windmill en dus niet afhankelijk zijn van de toepassing binnen het Projectmanagementsysteem Aeolus.
2. De verbinding met de database die JSON teruggeeft moet ook in andere modules bruikbaar zijn.



6. Verdere ontwikkeling

Hieronder een overzicht van de verschillende pilots tot nu toe. Daarna zullen de pilots één voor één uitgelegd worden

- Pilot Project, Ticket, Taakstructuur
- Pilot JSON
- Pilot Frontend

Pilot Project, Ticket, Taakstructuur

Naam van de pilot: *Project, Ticket, Taakstructuur*

Doel: Het ombouwen van de structuur van het systeem om accuratere gegevens te kunnen verkrijgen en het vereenvoudigen van het plaatsen van de agenda module mogelijk maken.

Programma voor bouwen: Quanta Editor en Mozilla Firefox

Richtlijnen: -

Kosten: -

Tijdsplanning:

- Week 1:* Oriëntatie op de werking van het huidige systeem
- Week 2:* Fouten in het huidige systeem inventariseren
- Week 3:* Database ontwerp maken voor nieuwe systeem
- Week 4:* Omzetten van gegevens uit oude database naar nieuwe
- Week 5:* Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
- Week 6:* Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
- Week 7:* Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
- Week 8:* Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
- Week 9:* Verbeteren van eventuele fouten
- Week 10:* Omzetten naar nieuwe structuur voor eindgebruiker

Pilot JSON

Naam van de pilot: *JSON*

Doel: Het implementeren van aanvragen naar de server die Javascript Object Notation (JSON) teruglevert. Dit moet bovenop Propel worden gedaan. Propel is een systeem dat gegevens uit databases aan PHP geeft als objecten in plaats van tekst.

Programma voor bouwen: Php, Phing, Propel, Quanta Editor en Mozilla Firefox

Richtlijnen: -

Kosten: -

Tijdsplanning:

- Week 11:* Generator van Propel ombouwen
- Week 12:* Ontvangst van gegevens in javascript mogelijk maken
- Week 13:* Verwerken van ontvangen gegevens in javascript mogelijk maken



Pilot Frontend

Naam van de pilot: *Frontend*

Doel: Het maken van de GUI en de interactie met de GUI.

Programma voor bouwen: Quanta Editor en Mozilla Firefox

Richtlijnen: Er wordt gekeken naar de stijl van het huidige systeem, op een dergelijke manier dat de module wat uiterlijk betreft binnen het huidige systeem niet uit de toon valt.

Kosten: -

Tijdsplanning:

Week 14: Agendapagina maken waarop gegevens gepresenteerd kunnen worden

Week 15: Gegevens uit database presenteren op agenda pagina

Week 16: Gegevens aanpasbaar maken.



7.7 Pilotplan

Pilotplan

Aeolus Agenda



Inhoud

1. Inleiding.....	3
2. Pilots.....	4
2.1. Project, Ticket en Taakstructuur	4
2.2. JSON.....	5
2.3. Frontend.....	6
3. Verloop pilots	7



1. Inleiding

Het pilotplan is een stap uit de IAD ontwikkelmodel. Dit is een ontwikkelmodel beschreven in het boek "IAD, het evolutionair ontwikkelen van informatiesystemen". Het beschrijft een model voor het op een iteratieve manier ontwikkelen van informatiesystemen. IAD propageert een cyclische aanpak waarin het informatiesysteem tot stand komt via een aantal snel uitgevoerde iteraties van de complete ontwikkelcyclus. Daarbinnen hoort een opdeling van het project in delen, zogehete pilots. In dit plan wordt beschreven voor welke pilots ik heb gekozen in de oriëntatiefase van mijn project.

Er wordt per pilot beschreven wat de aanleiding is tot het kiezen van deze pilot. Het doel van de pilot wordt aangegeven, net als de programma's voor het bouwen, de richtlijnen en de tijdsplanning.

Dit plan is gemaakt om een kort en bondig overzicht te geven van de deelproducten van het project en daarmee voldoet het niet aan alles wat IAD voorschrijft, want het was slechts mijn bedoeling om een eenvoudige afbakening van het project te maken.



2. Pilots

2.1. Project, Ticket en Taakstructuur

Aanleiding

De aanleiding voor deze pilot is dat de managementinformatie, die onder andere aan de hand van de agenda, uit Aeolus gehaald kan worden niet accuraat was. Dit doordat er niet op taakniveau geregistreerd en gepland kon worden.

Doel

Het ombouwen van de structuur van het systeem om accuratere gegevens te kunnen verkrijgen en het vereenvoudigen van het plaatsen van de agenda module mogelijk maken. Er kan na het doorlopen van deze pilot informatie uit Aeolus gehaald worden over één bepaalde werkzaamheid en niet alleen maar over een verzameling van werkzaamheden.

Programma voor bouwen

- Quanta Editor
- Mozilla Firefox

Richtlijnen

De geschreven code moet voldoen aan de Quality Assurance van Connectholland.

Tijdsplanning

- Week 1:** Oriëntatie op de werking van het huidige systeem
- Week 2:** Fouten in het huidige systeem inventariseren
- Week 3:** Database ontwerp maken voor nieuwe systeem
- Week 4:** Omzetten van gegevens uit oude database naar nieuwe
- Week 5:** Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
- Week 6:** Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
- Week 7:** Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
- Week 8:** Ombouwen frontend om aan eisen van de nieuwe structuur te voldoen
- Week 9:** Verbeteren van eventuele fouten
- Week 10:** Omzetten naar nieuwe structuur voor eindgebruiker



2.2. JSON

Aanleiding

Database gegevens zijn in het framework van Connectholland vooral in het xml-formaat, deze gegevens worden dan meestal geconverteerd naar HTML met gebruik van XSL. Maar de agenda module moet gegevens asynchroon uit de database verkrijgen, deze gegevens worden in het xml-formaat verstuurd. Daardoor moet er met een Javascript in de internet browser van de gebruiker nog een conversie plaatsvinden. Het is gemakkelijker om deze conversie automatisch plaats te laten vinden. Dit kan als de gegevens niet in xml-formaat, maar in JavaScript Object Notation (JSON) worden verstuurd naar het Javascript in de internet browser van de gebruiker.

Doel

Het implementeren van aanvragen naar de server die Javascript Object Notation (JSON) teruglevert. Dit moet bovenop Propel worden gedaan. Propel is een systeem dat gegevens uit databases aan PHP geeft als objecten in plaats van tekst. Met deze pilot wordt gezorgd dat het framework ook gegevens in Javascript Object Notation kan maken en aan de internet browser van de gebruiker kan sturen.

Programma voor bouwen

- Php
- Phing
- Propel
- Quanta Editor
- Mozilla Firefox

Richtlijnen

De gegevens die worden verstuurd moeten voldoen aan de JavaScript Object Notation.

Tijdsplanning

Week 11: Generator van Propel ombouwen

Week 12: Ontvangst van gegevens in javascript mogelijk maken

Week 13: Verwerken van ontvangen gegevens in javascript mogelijk maken



2.3. Frontend

Aanleiding

Nadat alle gegevens afhandeling is vastgelegd (in de vorige pilot), moeten deze gegevens uiteraard gepresenteerd worden aan de gebruiker. Daarnaast moet de gebruiker de gegevens kunnen aanpassen. Daarvoor moet het mogelijk zijn om gegevens op te zoeken, te wijzigen, toe te voegen en te verwijderen.

Doel

Het maken van de GUI en de interactie met de GUI.

Programma voor bouwen

- Quanta Editor
- Mozilla Firefox

Richtlijnen

Er wordt gekeken naar de stijl van het huidige systeem, op een dergelijke manier dat de module wat uiterlijk betreft binnen het huidige systeem niet uit de toon valt. De werking van de agenda moet nagenoeg het zelfde zijn als die van Google Calendar.

Tijdsplanning

Week 14: Agendapagina maken waarop gegevens gepresenteerd kunnen worden

Week 15: Gegevens uit database presenteren op agenda pagina

Week 16: Gegevens aanpasbaar maken.



3. Verloop pilots

De pilots zijn zo dat ze allen afzonderlijk kunnen worden begonnen en in een willekeurige volgorde, maar ze moeten wel in de volgorde van nummering eindigen. Of ook wel, pilot 3 is nutteloos zonder dat pilot 1 en 2 af zijn, maar pilot 3 kan voordat 1 en 2 af zijn wel al af zijn. Dit komt doordat pilot 3 het uiteindelijke product is en de andere pilots dienen ter ondersteuning daarvan. Ik heb er toch voor gekozen om deze pilots sequentieel uit te gaan voeren.

