

Afstudeer verslag

“Koppeling Google Docs met OpenIMS”



Datum : 11-03-2011
Plaats : Nieuwegein (Utrecht)

Auteur : Danny van Tol
Studentnr : 20057842
Opleiding : Informatica

DE HAAGSE
HOGESCHOOL

Begeleider(s) : Dhr. Okan Zor / Mevr. Gerda In 't Veld
Functie : Docenten (afstudeerbegeleiders)

OpenSesame ICT
de kracht van eenvoud

Opdrachtgever: Dhr. Nico de Vries
Functie : Directie
Begeleider: Dhr. Eric Diks
Functie : Software ontwikkelaar

Afstudeerverslag versie :Final

Document revisie geschiedenis

Versie beheer

Revisie datum	Auteur	Versie
06-04-2011	Danny van Tol	1.0
07-04-2011	Danny van Tol	1.1
08-04-2011	Danny van Tol	1.2
25-05-2011	Danny van Tol	1.3
07-06-2011	Danny van Tol	1.4
16-06-2011	Danny van Tol	1.5
17-06-2011	Danny van Tol	1.6
22-06-2011	Danny van Tol	1.7
11-07-2011	Danny van Tol	1.8
12-07-2011 / 09-09-2011	Danny van Tol	Am Final
09-09-2011 / 23-09-2011	Danny van Tol	Final

Tabel 1: Versie beheer

Distributielijst

Naam	Instantie	Rol/Functie	Begindatum	Einddatum
Eric Diks	OpenSesame ICT	Begeleider (intern)	11-03-2011	15-07-2011
Nico de Vries	OpenSesame ICT	Opdrachtgever	11-03-2011	15-07-2011
Okan Zor	Haagse Hogeschool	Begeleider (school)	11-03-2011	15-07-2011
Gerda In 't Veld	Haagse Hogeschool	Begeleider (school)	11-03-2011	15-07-2011
Administratië	Haagse Hogeschool	Distributie	11-03-2011	15-07-2011

Tabel 2: Distributielijst

Management Samenvatting

Met de koppeling tussen Google Docs en OpenIMS is het document management systeem van OpenSesame ICT bruikbaar geworden. De doelstelling was een koppeling te realiseren tussen Google Docs en OpenIMS om de documenten die in het document management systeem van OpenIMS staan, te laten synchroniseren met de documenten in Google Docs en visa versa.

Met het oude document management systeem van OpenIMS konden alleen documenten beheerd en opgeslagen worden, maar niet real-time aangepast worden, wat de unieke functionaliteit Google Docs wel heeft.

Door de integratie van Google Docs met OpenIMS heeft OpenSesame ICT een mooi en tevens gratis alternatief om haar klanten een omgeving aan te bieden waarmee zij nu de mogelijkheid hebben om de documenten die zij bezitten door meerdere werknemers tegelijk te laten bewerken. Door de synchronisatie functionaliteit zullen de bewerkingen die gedaan zijn in Google Docs ook up-to-date blijven in het document management systeem van OpenIMS.

Voorwoord

Dit afstudeerverslag is geschreven ter afsluiting van de opleiding Informatica aan de Haagse Hogeschool. Dit verslag beschrijft hoe ik mijn afstudeeropdracht bij OpenSesame ICT in Nieuwegein heb uitgevoerd.

Voor dit project heb ik me de afgelopen vier maanden mogen verdiepen in een omgeving met nog twee stagiaires. We hebben naast hard werken ook een gezellige tijd met elkaar gehad, en dat maakt dat het een waardevolle periode is geweest.

Tot slot wil ik graag mijn interne begeleider Eric Diks bedanken voor de nuttige hulp en zijn behulpzame houding. Tevens wil ik ook de begeleiders van school, Dhr. Okan Zor en Mevr. Gerda In 't Veld bedanken voor de bijdrage aan mijn project door middel van gesprekken en de adviezen.

Nieuwegein, 15 juli 2011

Inhoudsopgave

1	Inleiding.....	7
2	Achtergronden	8
2.1	De organisatie.....	8
3	Planning	9
3.1	Globale planning.....	9
4	De afstudeeropdracht.....	10
4.1	Probleemstelling.....	10
4.2	Doelstelling.....	10
4.3	Onderzoeksvragen.....	10
4.4	Scope van het afstudeerproject	10
5	Onderzoeksfase	11
5.1	Google Docs.....	11
5.2	OpenIMS.....	13
5.2.1	Producten en diensten van OpenSesame ICT	13
5.2.2	DMS architectuur van OpenIMS.....	14
6	Aanpak op basis van de onderzoekresultaten.....	18
6.1	Ontwikkelmethodiek DSDM	18
6.2	Gebruikte hulpmiddelen en ontwikkeltools	21
6.3	Gebruikte software libraries.....	22
6.3.1	Het Zend Framework.....	22
6.3.2	Gdata library van Zend	23
7	Ontwerpfase (fase 4)	25
7.1	MoSCoW prioritering.....	25
7.2	Timeboxes.....	25
7.3	Functioneel systeemontwerp	26
7.4	Technisch systeemontwerp	29
8	Ontwikkelfase (fase 4)	31
8.1	Authenticatieproces Google Docs (1ste increment).....	31
8.2	Upload functionaliteit (2de increment).....	32
8.3	Download functionaliteit (3de increment)	32
8.4	Update functionaliteit (4de increment).....	33
8.5	Delete functionaliteit (5de increment)	33
8.6	Synchronisatie functionaliteit (6de increment).....	34
8.7	Verloop Timeboxes (incrementen).....	36

9	Implementatiefase (fase5).....	37
9.1	Oplevering producten.....	37
9.2	Handleiding.....	37
10	Evaluaties	38
10.1	Productevaluatie	38
10.2	Procesevaluatie	38
10.3	Beroepstaken.....	39
10.4	Zelfreflectie.....	41
10.5	Conclusie.....	41
11	Woordenlijst.....	42
12	Referentielijst	43
13	Bijlage.....	44

1 Inleiding

De door OpenSesame ICT ontwikkelde OpenIMS is een server product dat geheel in PHP geschreven is. Dit product dient als bouwsteen voor haar losse producten. Deze losse producten zijn; portaal, document, content, formulier en email management. Deze op service gerichte producten werken op een geïntegreerde manier met elkaar samen en vullen ieder een eigen functie in het systeem.

Binnen één van deze producten worden documenten en content bestanden beheerd door een management systeem, document management systeem (DMS) en content management systeem (CMS) waarin documenten gemanaged worden door gebruikers die verantwoordelijk zijn voor bijvoorbeeld het goedkeuren of wijzigen van documenten.

In het document management systeem van OpenIMS kunnen documenten opgeslagen, bekeken en gewijzigd worden. Met dit documenten systeem zijn gebruikers niet in staat om opgeslagen documenten in real-time met meerdere gebruikers te bewerken. Mede hierdoor is het document management systeem binnen OpenIMS minder functioneel als de medewerkers van OpenSesame ICT zouden willen. De behoefte is om de real-time editing functionaliteit die Google Docs heeft, samen te laten werken met het documentensysteem van OpenIMS voor extra functionaliteit van documentenbeheer.

2 Achtergronden

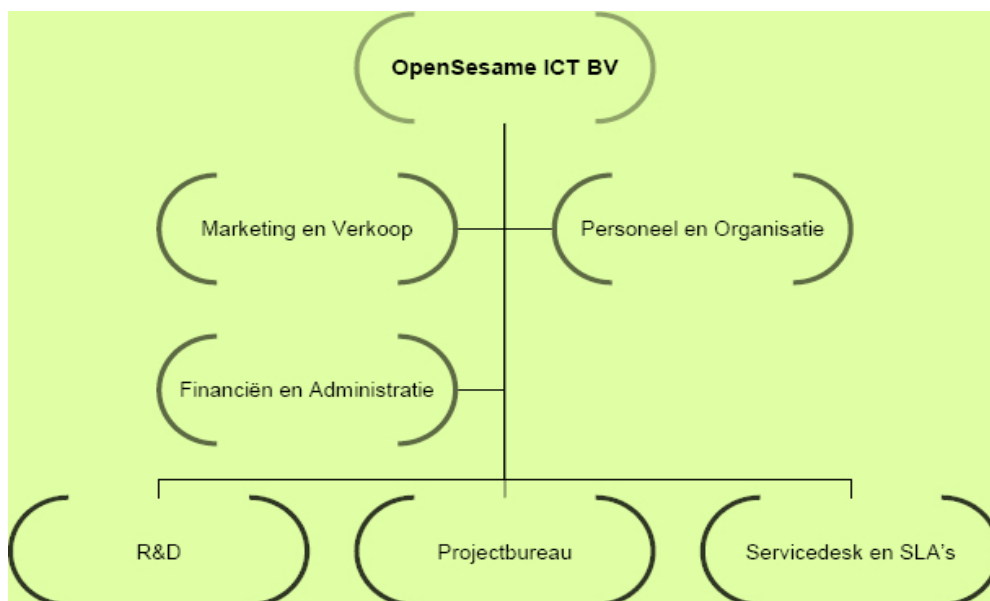
Het afstudeerproject die ik als student bij OpenSesame ICT op de afdeling Research en Development heb uitgevoerd, heeft in de periode van 11 maart tot en met 15 juli 2011 plaatsgevonden. Onderzocht moest worden welke mogelijkheden er zijn voor een koppeling van OpenIMS met Google Docs.

2.1 De organisatie

OpenSesame ICT is een ICT dienstverlener die opgericht is in 2003 door drie man, de huidige drie directeuren. Het is nu uitgegroeid tot een middelgroot bedrijf. OpenSesame ICT richt zich voornamelijk op het verder ontwikkelen en implementeren van de productlijn OpenIMS. Het is een pakket dat zelf door OpenSesame ICT is bedacht en ontwikkeld, het is tevens het vlaggenschip van de organisatie. OpenIMS is gebaseerd op het Open Source platform, wat inhoudt dat het volledig gebouwd is met 'open' technieken die in veel gevallen gratis toegankelijk zijn voor iedereen.

Verder levert OpenSesame ICT ook andere diensten aan zoals advies, ontwikkeling, implementatie, integratie en support voor OpenIMS. De klant koopt dus niet alleen het product, maar ook de implementatie of integratie met hun huidige producten en nazorg van het systeem. Dit gebeurt doormiddel van support en wordt opgenomen in de SLA's (*Service Level Agreements*), oftewel contracten.

De organisatie bestaat op dit moment uit zeven afdelingen met 30 tot 35 werknemers die hieronder te zien zijn in het organigram. Het bedrijf is nog sterk groeiende en er komen regelmatig nieuwe werknemers bij.



Figuur 1: Toon de verschillende afdelingen aan binnen OpenSesame ICT BV.

De meeste mensen binnen OpenSesame ICT zijn ontwikkelaars. Op de afdeling Research en Development zitten dan zo'n 22 vaste medewerkers. De afdeling Marketing en verkoop 3, de afdeling Personeel en Organisatie 2, de afdeling Financien en Administratie 3, de afdeling Projectbureau 2 en de afdeling Servicedesk en SLA's ook 2 personen.

Gebieden waarin OpenSesame ICT actief is, zijn de overheid, gezondheidszorg, onderwijs en de zakelijke dienstverlening. Klanten van OpenSesame ICT zijn onder andere:

- De Nederlandse Spoorwegen
- Fuji Film
- HGM (Hulpverlening Gelderland Midden)
- Bouwfonds
- Deventer Ziekenhuis
- Nederlandse Overheid (waarvan een aantal ministeries)

Één van de belangrijkste pakketten die OpenSesame ICT naast OpenIMS gebruikt is het relationship management pakket SugarCRM.

SugarCRM is een product uit Amerika dat sinds 2004 op de markt is. Het product is een Open source Customer Relationship Management (CRM) software pakket om gegevens van klanten te kunnen beheren.

Het is een webbased pakket zodat het altijd en overal beschikbaar kan zijn, mits er een internet verbinding beschikbaar is. Doordat deze SugarCRM applicaties overal beschikbaar kunnen zijn is het bijvoorbeeld mogelijk om via Google, je mail, agenda, en documenten online te delen met anderen. Dit is een handige functionaliteit, want sales- en marketingmedewerkers zijn vaak, en soms ook lang, onderweg. Via een browser worden gegevens makkelijk toegankelijk gemaakt, zodat een gebruiker op elke plek (met internetverbinding) en op ieder tijdstip gebruik kan maken van de diensten van SugarCRM.

Het doel van de organisatie OpenSesame ICT als geheel

Het doel van OpenSesame ICT is het bieden van IT oplossingen op basis van Open Source Software. Ze maken gebruik van Open Source omdat dit beter aan te passen is, aan de wens van de gebruiker of klant. De openheid van broncode maakt het altijd mogelijk de software aan te passen aan bestaande systemen, tools, platformen, databases et. cetera. Buiten de openheid van Open Source is het voor de klant ook mogelijk om derden in te schakelen en niet bij één bedrijf te blijven hangen. Tevens heeft de klant als voordeel dat zij geen hoge licentiekosten hoeven te betalen in tegenstelling tot gesloten software zoals Office en Windows pakketten.

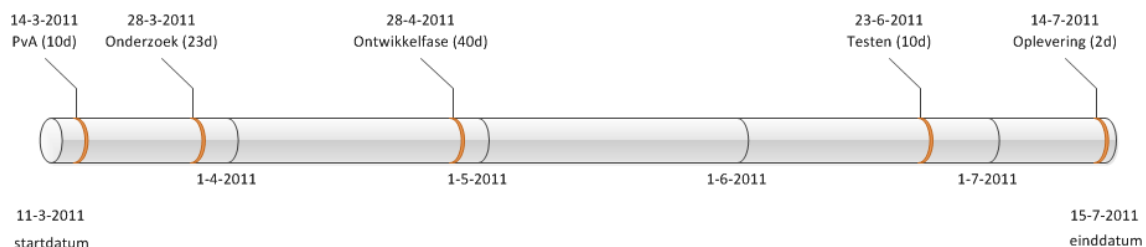
Diensten van OpenSesame ICT

OpenSesame ICT levert kennis en ervaring op basis van standaard oplossingen voor het structureren van alle ongestructureerde informatie in een organisatie. Dit betekent praktische oplossingen op het gebied van onder andere Webcontent, Document, Capture, CRM en Workflow Management.

Voor het realiseren van deze oplossingen maakt OpenSesame ICT zo veel mogelijk gebruik van beschikbare Open Source Software, die waar nodig 'geconfigureerd' wordt naar de specifieke situatie die de klant wil. Hierdoor kunnen oplossingen snel geïmplementeerd worden en blijven de kosten beheersbaar. [\[bron\]](#)

3 Planning

3.1 Globale planning



Figuur 2: Globale planning afstudeerproject

4 De afstudeeropdracht

4.1 Probleemstelling

Google Docs is een omgeving waarin documenten in de cloud kunnen worden beheerd met een eenvoudige online tekstverwerker en spreadsheet. Google Docs heeft geen document meta-data zoals in OpenIMS wel het geval is. De bedoeling van dit project is om onderzoek te doen naar de mogelijkheden van interfacing van OpenIMS met Google Docs, om vervolgens deze twee systemen te koppelen en daarmee samen te laten werken. Functioneel wordt hierbij gedacht aan een synchronisatie tussen deze twee systemen, en het over en weer beschikbaar stellen van functionaliteiten.

OpenSesame ICT heeft een documenten management systeem in het OpenIMS pakket. Dit documentensysteem heeft echter zijn beperkingen. Bestanden kunnen geüpload, bekeken en gewijzigd worden. Dit kan echter maar door één persoon tegelijk (één kant op). Er is geen functie beschikbaar zoals in Google Docs, waarbij meerdere mensen in staat zijn om tegelijkertijd een document te bekijken en of te wijzigen, zonder het risico dat er informatie verloren kan gaan.

Tot op heden heeft niemand binnen dit bedrijf nog iets gedaan aan de integratie tussen het OpenIMS documenten systeem en Google Docs. Het is hierdoor van belang dat er onderzoek wordt gedaan of Google Docs met OpenIMS kan interfacen, en zo ja, welke middelen hiervoor beschikbaar en benodigd zijn.

4.2 Doelstelling

Doelstelling voor de student is aan het einde van de afstudeerperiode inzicht te hebben in de werking van de afzonderlijke producten (Google Docs en het OpenIMS DMS), en de mogelijkheden voor interfacing tussen deze twee systemen te definiëren en te realiseren.

Met deze doelstelling wil ik bereiken dat het product ook echt waardevol is voor het bedrijf en de klanten van OpenSesame ICT.

4.3 Onderzoeksvragen

Voor het project zijn een tweetal onderzoeksvragen opgesteld welke centraal staan in dit afstudeertraject:

- Onderzoek doen naar de bestaande functionaliteit van Google Docs. Wat is het, hoe werkt het, en hoe is het opgebouwd?
- Wat is, en wat kan het OpenIMS document management systeem? Welke functies binnen OpenIMS zijn beschikbaar, en hoe kan deze functionaliteit met Google Docs samenwerken, en zo ja, hoe?

4.4 Scope van het afstudeerproject

Met de scope van het project wordt een grens vastgelegd waarbinnen dit project plaatsvindt:

- (1) Dit project betreft alleen documenten beheer en geen andere bestanden.
- (2) Het OpenIMS document management systeem moet zijn huidige functionaliteit blijven behouden.
- (3) Nieuwe functionaliteit van Google Docs moet samenwerken met de bestaande DMS functionaliteit van OpenIMS
- (4) Synchronisatie van de functionaliteit vindt alleen plaats tussen OpenIMS en Google Docs en geen derde of andere software pakketten.

5 Onderzoeksfase

In deze fase wordt onderzoek gedaan aan de hand van de onderzoeksvragen. In dit verslag zullen zowel Google Docs als OpenIMS apart worden onderzocht.

Het eerste waar onderzoek naar gedaan zal worden is Google Docs. Bij Google Docs is het van belang hoe de structuur is opgebouwd, wat het kan, en wat je ervoor nodig hebt. Alle informatie wat betreft Google Docs, zal van internet moeten komen. Binnen het bedrijf OpenSesame ICT heeft niemand nog iets inhoudelijk gedaan met Google Docs.

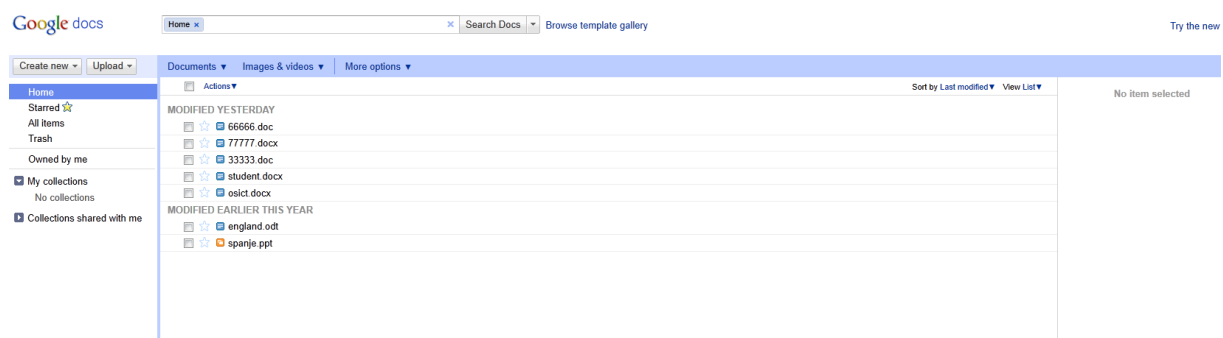
Als tweede moet onderzoek gedaan worden naar OpenIMS. Niet het hele OpenIMS pakket zal hierbij onderzocht worden. Dit onderzoek zal zich beperken tot het documenten systeem (DMS) van OpenIMS. Dit is namelijk het essentiële onderdeel (serverproduct) waarmee gewerkt zal worden. Ook hierbij geldt hetzelfde als voor Google Docs. Hoe het werkt het documenten systeem van OpenIMS, wat is het precies, wat kan je ermee, en hoe is de structuur opgebouwd. Voor informatie hierover kan ik terecht op de website van OpenSesame zelf, het intranet, andere verslagen, handleidingen, of indien iets niet te vinden is, bij de mensen van OpenSesame ICT zelf.

Aan de hand van de onderzoeksresultaten zal vervolgens een te hanteren ontwikkelmethodiek, en ontwikkeltool(s) gekozen worden. Na een methodiek gekozen te hebben, zal deze methodiek en de daarbij horende technieken toegepast worden op het afstudeerproject. Dit zal met name betrekking hebben op het ontwerp en de bouw van de applicatie.

5.1 Google Docs

Wat is Google Docs?

Google Docs is een gratis online dienst die Google aanbiedt. Met Google Docs is het mogelijk om opgemaakte documenten, spreadsheets en presentaties online op te slaan. Deze dienst kun je vergelijken met de betaalde programma's als Word van Microsoft Office. De tekstverwerker van Google Docs heette vroeger Writely, en is tegenwoordig bekend als Google Docs editor. Zie figuur 3 om een visueel beeld te krijgen hoe Google Docs eruit ziet:



In figuur 3 zie je de Google Docs omgeving met een aantal documenten

Wat heb je er voor nodig?

Om van Google Docs gebruik te kunnen maken, heb je een Google account nodig. De aangemaakte documenten en/ of spreadsheets worden online opgeslagen en kunnen ook met andere Google gebruikers worden gedeeld en real-time tegelijk worden bewerkt, waarbij je direct elkaars wijzigingen op het scherm kan zien.

Wat kan je er eigenlijk mee?

Met Google Docs is het mogelijk om bestanden in de cloud te beheren in een real-time omgeving. Dit houdt in dat je met verschillende personen op een afzonderlijke werkplek je documenten online kunt aanmaken, bekijken en wijzigen.

Gmail account gebruikt voor dit project

Doordat OpenSesame een gratis oplossing wil voor haar klanten, is in het geval van het project Google Docs en OpenIMS gekozen voor een Google email account. Tevens was het mogelijk om alle gevraagde functionaliteit te realiseren zonder Google Apps benodigd was. Op dit moment is het mogelijk om met elk willekeurig Google account dit project uit te voeren.

De structuur van Google Docs

Het Data Protocol van Google Docs is gebaseerd op ReST (zie bijlage: 1.1) technologie voor het lezen, schrijven en wijzigen van informatie op het web. [\[bron\]](#)

Veel diensten van Google bieden toegang tot externe gegevens en functionaliteit via API's die de Google Data-protocol gebruiken. Het protocol ondersteunt momenteel twee primaire vormen van toegang:

- **AtomPub XML:** Informatie wordt verzonden als een verzameling van Atom-items, met behulp van het standaard Atom syndicatie formaat om de data en HTTP communicatie te kunnen verwerken. Denk hierbij aan functies als GET, POST en DELETE om entries te verkrijgen, aan te maken, of te verwijderen. Het Google Data Protocol gebruikt AtomPub tevens voor het verwerken van queries, verificatie en batch verzoeken. [\[bron\]](#)
- **JSON:** Informatie wordt verzonden als JSON objecten welke de aanwezigheid controleren van Atom.

Als je direct op het Google protocol wilt programmeren, kun je gebruik maken van HTTP requests als GET en POST. Met deze aanroepen wordt Google data verstuurd in de vorm van zogeheten datafeeds. Deze datafeeds zijn simpelweg gestructureerde lijsten welke data bevatten. Voorheen was het primaire formaat van de feed AtomPub XML*, maar nu is dat JSON*, als alternatief bestandsformaat.

Als de voorkeur niet uitgaat naar het schrijven van code met directe HTTP aanroepen, is het ook mogelijk om programma's te schrijven met één van de programmeertalen die beschikbaar zijn in de set van cliënt libraries via de Google API. Op deze manier wordt de code meer op een conceptuele manier geschreven met gebruikmaking van klassen en functies van een specifieke programmeertaal.

Aangezien OpenIMS een webbased product is, welke hoofdzakelijk geschreven is in PHP, is voor dit project gebruik gemaakt van de PHP cliënt library. De PHP list API is beschikbaar in protocol versie 1. Deze versie is 'deprecated' wat inhoudt dat deze verouderd is. In de planning van Google zal de PHP API in 2012 geüpdate worden. Doordat er weinig code functionaliteit van de PHP list API zelf beschikbaar is, maakt deze gebruik van een aparte library genaamd, Gdata. Gdata is ontwikkeld door Zend Frameworks en biedt functionaliteit welke te gebruiken is in combinatie met de bestaande code functies van de PHP API. Meer over de PHP API en het gebruik hiervan is te lezen in *H6.3 Gebruikte software libraries*.

Google heeft naast PHP library nog een drietal andere talen als Java, Python en .NET. Deze zijn uitgebracht in protocol versie 2 en zijn meer up-to-date met meer standaard functionaliteiten.

NOTE * Zie *Bijlage 1.1* voor meer informatie over de Google Protocol bindings als ReST, XML, Atom en JSON

5.2 OpenIMS

De afkorting 'OpenIMS' staat voor Open Informatie Management Server. Het is een platform onafhankelijk product, wat wil zeggen dat het op alle besturingssystemen kan draaien (Windows, Mac, Unix). De Core van het product bestaat uit een PHP backend dat draait op een Apache web server.

Het doel van OpenIMS is structuur bieden in het beheer en onderhoud van niet structurele data. Deze ontstaan tijdens werkprocessen of projecten waarin een groep gebruikers zich bezig houdt met het uitvoeren van taken waarin documenten, webpagina's en e-mail berichten onderling worden geproduceerd, uitgewisseld of gebruikt.

Veel informatie staat in de meeste gevallen decentraal op verschillende bron locaties, als websites, intranet, file servers of op lokale harde schijven en in e-mail accounts. Deze 'ongestructureerde' informatie wordt met het product OpenIMS beheerd, zodat alle gebruikers over dezelfde actuele informatie kunnen beschikken.

5.2.1 Producten en diensten van OpenSesame ICT

OpenIMS (ECM):

OpenIMS is een van de eerste Informatie Management Servers (IMS) en is door OpenSesame ICT ontwikkeld. In plaats van separate oplossingen voor Content Management, Document Management, Kennis Management en E-mail Archivering biedt OpenIMS een geïntegreerde oplossing voor al deze terreinen. Voordeel van deze integratie is dat het hergebruik van documenten enorm bevordert wordt, en dat er een centrale opslag is voor alle mogelijke vormen van informatie. OpenIMS heeft vijf serverproducten. Dit zijn: DMS, CMS, BPMS, EMS en PS. Deze serverproducten zullen in deze paragraaf aan de orde komen.

Document management (DMS):

OpenIMS Document Management Server is volledig ontwikkeld op basis van internettechnologie. Met 'replicatie' is het mogelijk om documenten offline te bewerken en later te synchroniseren. OpenIMS komt het beste tot zijn recht in een online omgeving, maar is dermate efficiënt dat zelfs op een laptop de performance uitstekend is. Hierdoor kan het systeem vanaf iedere werkplek, binnen of buiten de organisatie, worden ingezet.

Capture management (BPMS):

Deze specialisatie is te omschrijven als het slim omzetten van papieren documenten, zoals de post die dagelijks ontvangen wordt naar een digitaal formaat. Over het algemeen is dit PDF of het archiefwaardige (toekomstige) PDF/A formaat.

Niet alleen het omzetten naar een digitaal formaat maar ook alle relevante informatie uit het papieren document herkennen en beschikbaar maken voor IT systemen is een specialiteit van OpenSesame ICT. OpenSesame ICT is in staat om NAW gegevens, bedragen, bankrekeningnummers, kenmerken en andere relevante gegevens uit papieren documenten te halen en gestructureerd beschikbaar te maken.

Web content management (CMS):

Bij het OpenIMS content management systeem kan gebruik worden gemaakt van bijvoorbeeld Microsoft Office voor het beheren van content (tekst, tabellen, berekeningen, presentaties en dergelijke) en Dreamweaver of Frontpage voor het beheren van de vormgeving.

E-mail management (EMS)

Binnen organisaties worden contacten steeds vaker met behulp van e-mail onderhouden. OpenIMS Email Management Systeem beschikt over een eenvoudige zoek faciliteit en kan worden ingezet om alle informatie binnen het E-mail Management Systeem te doorzoeken. De zoekmachine van OpenIMS beschikt over vergelijkbare faciliteiten als die van de populaire zoekmachine Google, maar dan binnen de OpenIMS omgeving.

Portaal management (PS):

Een Portaal kan worden ingezet om informatie (lees: documenten, bedrijfsapplicaties, nieuws, e-mail en dergelijke) efficiënt beschikbaar te stellen aan diegenen die daar behoefte aan hebben.

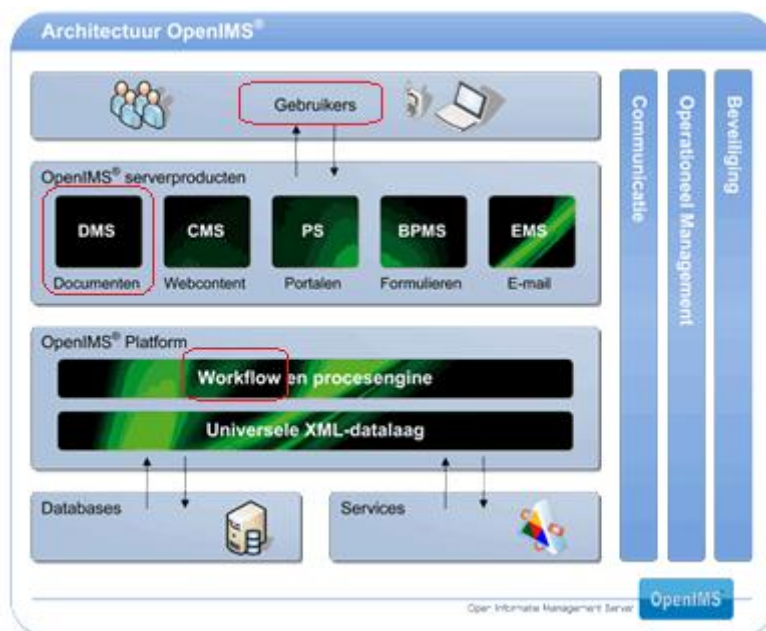
Het Portaal kan fungeren als een centrale toegang waarmee, gebruikmakend van LDAP technologie, een zogenaamde Single Sign On oplossing kan worden gerealiseerd. Met behulp van dit principe hoeft maar één keer te worden ingelogd waarna alle onderliggende systemen kunnen worden bereikt zonder telkenmale opnieuw te moeten inloggen. [\[bron\]](#)

Structuur OpenIMS platform

In figuur 2 is te zien hoe het OpenIMS platform zorgt voor de onderlinge communicatie tussen deze onderdelen. Het bestaat uit een workflow en proces engine, die gegevens opslaan via een universele XML datalaag. Data wordt binnen MySQL databases opgeslagen en diensten worden met behulp van SOAP (Simple Object Access Protocol) aan boven liggende lagen geboden. OpenIMS verzorgt in elk stadium de communicatie, operationeel management en beveiliging binnen het systeem.

Tijdens dit project is alleen van het DMS gebruik gemaakt als serverproduct. Alle andere serverproducten zijn beschikbaar na de OpenIMS installatie, maar deze hadden geen betrekking bij dit afstudeerproject. Naast het DMS serverproduct is alleen de workflow gebruikt de voor het controleren van de status van een document. In figuur 2 is in rood aangegeven wat direct gebruikt is tijdens dit project.

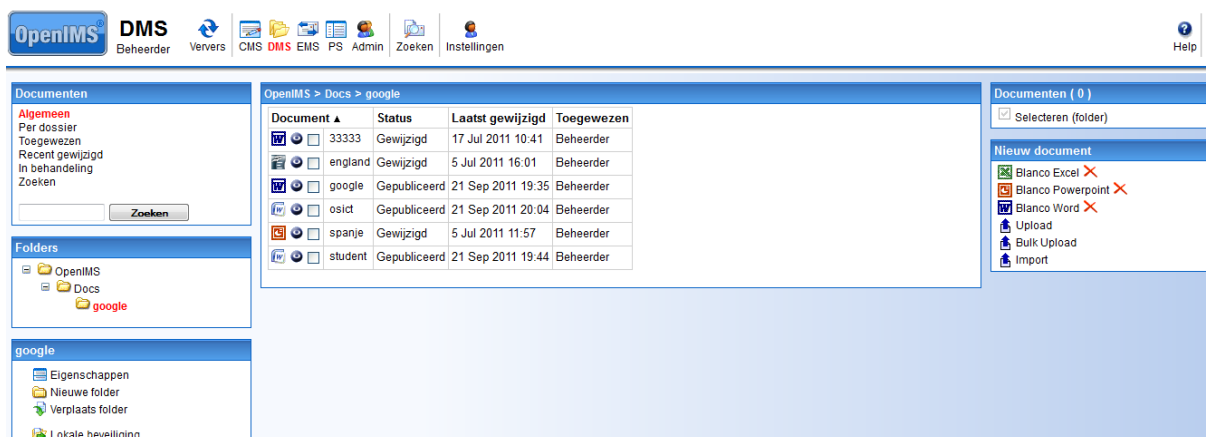
De fysieke database, de XML tussenlaag en services zijn niet direct, maar op de achtergrond gebruikt bij de ontwikkelfase van het eindproduct.



Figuur 4: een overzicht van de diverse OpenIMS serverproducten.

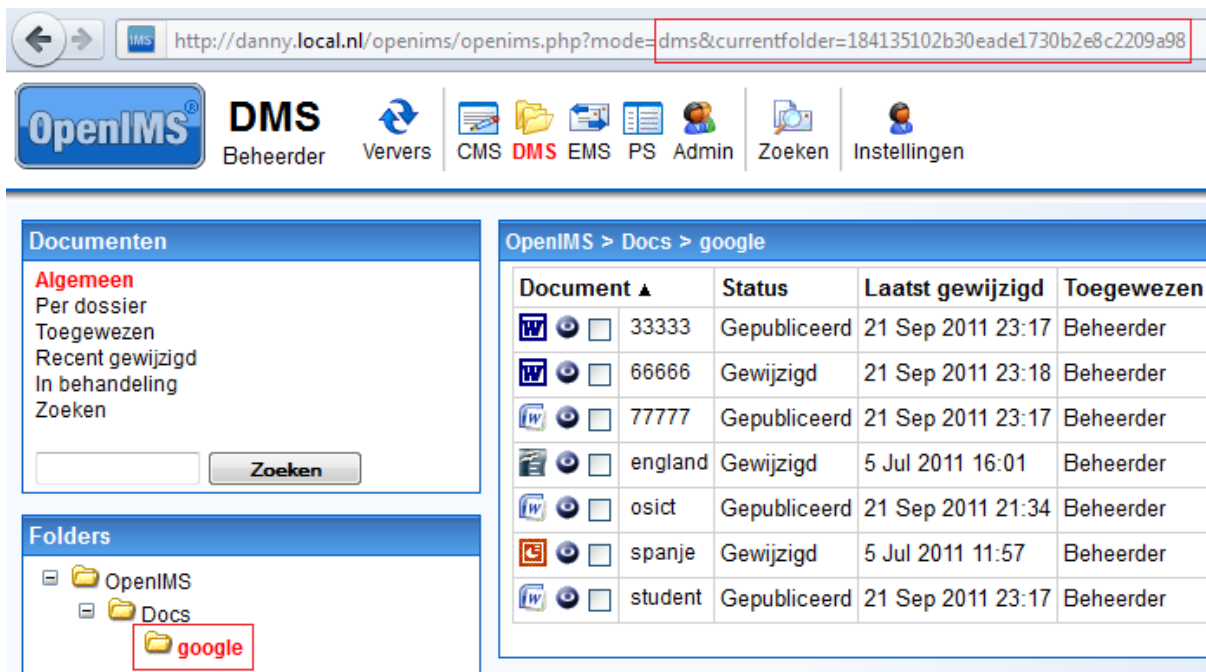
5.2.2 DMS architectuur van OpenIMS**Meta data opvragen van documenten**

Voor deze afstudeeropdracht waarin documentenbeheer centraal staat, is het relevant om DMS binnen OpenIMS verder toe te lichten. Zoals al eerder is benoemd, is het DMS één van de vijf serverproducten binnen OpenIMS. Het DMS ondersteunt documenten van diverse content types (extensions). Zie Figuur 5 van de OpenIMS DMS omgeving met een aantal documenten.



In figuur 5 zie je het DMS van OpenIMS.

De architectuur van dit subsysteem is zo ontworpen dat alle documenten en alle folders een eigen unieke ID hebben. Een document wordt opgeslagen in een subfolder van een folder welke het document ID als naam heeft. De reden dat het bestand wordt opgeslagen in een subfolder is omdat er van één document meerdere versies kunnen bestaan. De naam van een subfolder heeft eveneens een uniek ID (document-versie ID). Bij elke wijziging aan een document, zal de nieuwere gewijzigde versie in een nieuwe subfolder geplaatst worden. In figuur 6 zijn drie folders aangemaakt. De OpenIMS folder wordt niet gebruikt. Dit is de rootfolder met de naam root. De andere twee folders, Docs en Google worden wel gebruikt. Deze folder hebben allebei hun unieke ID. Dit is te zien m.b.v. de rode omlijningen.



Figuur 6 toont aan dat de folder Google (subfolder van Docs) een unieke ID heeft.

Dit systeem werkt voor documenten precies hetzelfde. Als je een document in het DMS aanklikt, is aan het einde van de URL string een document ID af te lezen.

Alle bestaande documenten in het OpenIMS DMS worden ook fysiek opgeslagen op de harde schijf van de server. De naam van het bestand heet standaard 'document' met de file extensie (docx, pdf, ppt, ...). Als een document altijd 'document' (plus extensie) als bestandsnaam heeft, dan moet het originele bestand ook ergens opgeslagen worden. Hiervoor maakt OpenIMS gebruik van een MySQL database. In deze database zijn op XML gebaseerde entries opgeslagen voor elk document dat aangemaakt is. Deze XML entries bevatten meta-data zoals de originele naam van het bestand, een document ID, een timestamp, de workflow status van een document. De workflow status van een document kan in de 'new' status zijn als een document net is aangemaakt, of de status 'edit-publish' hebben als een document gewijzigd is. De XML entry bevat zelfs een sub-entry voor elke versie (history) van het document welke gebonden is, aan het specifieke document-versie ID. In figuur 4 wordt een aantal meta-data gegevens in het geel en blauw aangegeven. Wat dit precies is, zal later uitgelegd worden.

Om een XML-entry van een document te verkrijgen als een PHP-array, heeft OpenIMS een functie ontwikkeld. Deze functie heet 'MB_Load'. De parameters, die nodig zijn voor deze functie, zijn de naam van de MySQL-tabel en het document ID. De afbeelding hieronder toont wat MB_Load terug geeft voor data met een voorbeeld-document.

Om deze gegevens te verkrijgen heeft OpenIMS een zogeheten 'codetester' ontworpen. Met deze codetester is het mogelijk om binnen het OpenIMS pakket PHP code uit te voeren. Hierdoor is het mogelijk om zowel standaard PHP als door OpenSesame ICT ontwikkelde PHP functies als MB_Load uit te voeren in de OpenIMS omgeving zelf.

```
MB_Load('ims_objects', 'da76438dc2995941dc84a6dd583d8ad7')

Array {
    [objecttype] => document
    [preview] => yes
    [published] => no
    [stage] => 1
    [directory] => da76438dc2995941dc84a6dd583d8ad7
    [shorttitle] => Afstudeer-verslag-DvT.docx
    [filename] => document.docx
    [allocto] => admin
    [workflow] => edit-publish
    [executable] => winword.exe
    [history] => Array
        (
            [11b1028c47194401db0c480a7e87ef44] => Array
                (
                    [type] => new
                    [when] => 1304950145
                    [author] => admin
                    [server] => FRESHINSTALL
                    [http_host] => localhost
                    [remote_addr] => 127.0.0.1
                    [server_addr] => 127.0.0.1
                )
        )
    )
}
```

Figuur 7: Voor een visueel beeld hoe en welke metadata gegeven wordt bij het opvragen met MB_Load.

In figuur 7 is de meta-data af te lezen die als een array teruggegeven wordt met het aanroepen van de "MB_Load" functie. De XML entry in het geel geeft het (ID) van de directory, de originele naam van het bestand (shorttitle) die is opgeslagen in de database, en de (filename) van het document wat standaard als document.docx wordt opgeslagen op de harde schijf. De blauwe markerings geven de sub-entries aan. De history geeft aan dat het om een array gaat. De lange string met nummers is het (ID) van het document, het (type) geeft aan dat het om een nieuw aangemaakt document gaat die nog niet gewijzigd is, en (when) is een timestamp wanneer een document is aangemaakt

Het lokaliseren van bestand(-en)

Om van de huidige versie van een document de fysieke locatie te achterhalen, is de functie 'FILES_Filelocation' binnen OpenIMS ontwikkeld. Deze functie heeft dezelfde parameters nodig als de functie 'MB_Load'. Op deze manier is het bestand toegankelijk, en kan eventueel met een extra functie de content uitgelezen worden als een PHP string.

Het opslaan van bestand(-en)

Voor het opslaan van een document in het OpenIMS DMS, zal er een nieuwe XML-entry aan de MySQL database toegevoegd moeten worden, en het nieuwe document zal op de juiste plaats op de harde schijf opgeslagen dienen te worden. Om een document op de harde schijf op te slaan, kan de functie 'N_Write_File' van OpenIMS gebruikt worden. Deze functie heeft twee parameters. Een string parameter voor de content van een document en een parameter om de filelocatie aan te geven waar het bestand opgeslagen dient te worden. Deze locatie moet gegenereerd worden door het gebruik van een nieuwe document id. De functie welke de filelocatie van een nieuw document genereert, is 'FILES_PreviewFileLocation' genaamd.

6 Aanpak op basis van de onderzoekresultaten

In dit hoofdstuk wordt de aanpak besproken van het project. Hierin zal de ontwikkelmethodiek centraal staan, maar ook zaken als welke andere hulpmiddelen en tools, libraries enz. er gebruikt zijn.

6.1 Ontwikkelmethodiek DSDM

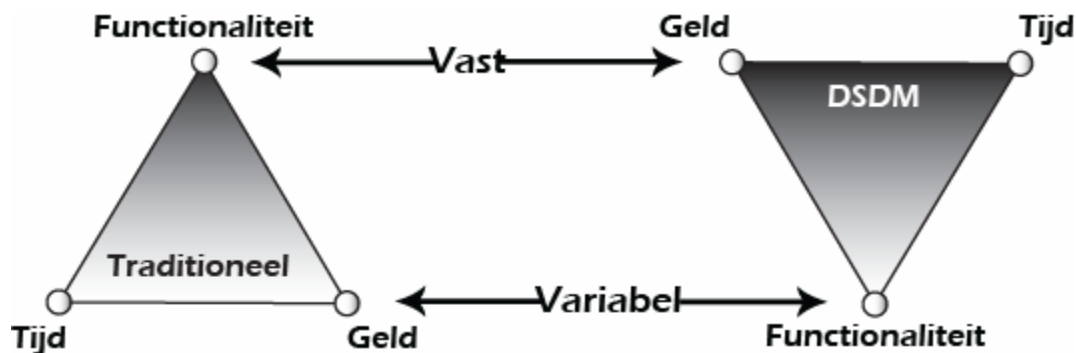
Aan de hand van het onderzoek en de resultaten hiervan, is gekozen voor DSDM als ontwikkelmethodiek. Waarom deze methodiek gekozen is, zal later duidelijk worden.

Wat is DSDM?

DSDM is een iteratieve, incrementele en agile ontwikkelmethode gebaseerd op Rapid Application Development (RAD) en werd halverwege de jaren negentig uitgebracht. De overeenkomst met RUP is dat het een raamwerk is en niet een methode die kant-en-klaar toegepast kan worden. Het raamwerk is een verzameling best-practices waaruit gekozen kan worden. Anders dan bijvoorbeeld RUP, is DSDM niet door een specifiek bedrijf bedacht, maar door een non-profit organisatie genaamd het DSDM consortium.

Het belangrijkste uitgangspunt bij DSDM is dat een product op tijd en binnen budget opgeleverd moet worden. Bij DSDM ligt de focus op projectmanagement. Daarom kan deze methode ook gecombineerd worden met XP en RUP. [[bron](#)]

Uitgangspunten van DSDM



Figuur 8: toont het verschil tussen traditionele methoden als de Waterval en DSDM.

In traditionele methoden wordt zo snel mogelijk vastgezet welke functionaliteit gemaakt gaat worden. De hoeveelheid tijd en geld die hiervoor nodig zijn, hangen af van de hoeveelheid functionaliteit die nodig is. Bij DSDM is de driehoek omgedraaid. Als eerste wordt de hoeveelheid tijd en geld vastgezet. Vervolgens wordt binnen die tijd en binnen het budget zoveel mogelijk functionaliteit gerealiseerd. [[bron](#)]

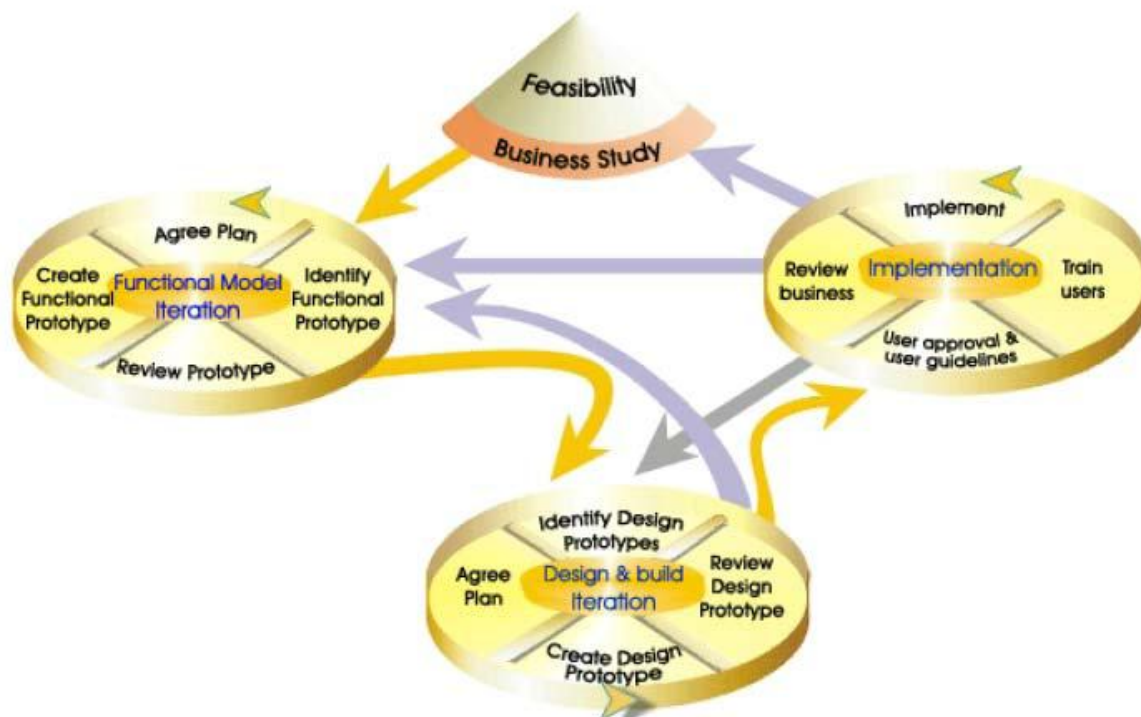
DSDM hanteert negen uitgangspunten:

1. Actieve betrokkenheid van de gebruiker
2. Beslissingsbevoegdheid van het team
3. Regelmatige levering van producten
4. **Geschiktheid voor het bedrijfsdoel (relevant voor dit project)**
5. **Incrementele/ iteratieve ontwikkeling (relevant voor dit project)**
6. **Veranderingen zijn omkeerbaar (relevant voor dit project)**
7. Vereisten worden op hoog niveau bevroren
8. **Testen gebeurt tijdens de gehele levenscyclus (wordt apart testrapport) (relevant voor dit project)**
9. Medewerking en samenwerking is noodzaak

Zoals aangegeven op de vorige pagina, zijn niet alle DSDM uitgangspunten gekozen als zijnde relevant voor dit project. De redenen van de punten die niet gekozen zijn, zijn de volgende:

1. De afstudeerder is de enige actieve betrokkene tijdens dit project die hierdoor geen kennis hoeft over te dragen. Tevens is een simpele handleiding gemaakt wat de gebruiker na oplevering van het product moet weten.
2. Er is geen speciale groep voor het nemen van beslissingen. Dit doet de afstudeerder enkel in samenspraak met de opdrachtgever.
3. Hoewel er incrementeel gewerkt wordt tijdens dit project, is regelmatige levering van producten niet aan de orde. De deelproducten zijn onderdeel van het gehele product, en hoeven niet afzonderlijk opgeleverd te worden.
7. Vereisten zullen tijdens dit project niet bevroren worden. Aan alle vereisten dient voldaan te worden zoals aangegeven in de MoSCoW lijst.
9. Het is de bedoeling dat de afstudeerder zijn opdracht alleen uitvoert. De noodzaak van samenwerking is dus beslist niet een noodzaak.

Stappenplan DSDM



Figuur 9: DSDM fasen.

In Figuur 9 is te zien hoe DSDM werkt. Eerst wordt er bekeken of DSDM een geschikte methode is voor het project dat uitgevoerd gaat worden. Dan krijg je een Business Study fase waarin een projectorganisatie wordt opgericht die zich o.a. bezig gaan houden met de planning, architectuur voor de software, en requirements. Vervolgens krijg je "Functional Model iteration" voor de ontwerpfase, de "Design en Build Iteration" voor het iteratief en/ of incrementeel bouwen van de software, en als laatste de "Implementation" fase voor het opleveren en overdragen van het eindproduct.

Het DSDM-proces bestaat uit de volgende vijf fasen:

1. *Feasibility Study*
In deze fase wordt gekeken of DSDM een geschikte methode is voor de situatie.
2. *Business Study*
In deze fase wordt de projectorganisatie opgericht, er wordt een planning op grote lijnen gemaakt, er vindt een risico inventarisatie plaats, de architectuur voor de software wordt bepaald en een lijst met eisen en randvoorwaarden op hoog niveau wordt vastgesteld.
3. *Functional Model Iteration*
Deze fase kan uit één of meerdere iteraties bestaan afhankelijk van de complexiteit van het project. In deze fase wordt gewerkt aan functionele prototypes die bedoeld zijn om de gewenste functionaliteit duidelijk te krijgen.
4. ***Design and Build Iteration (relevant voor dit project)***
Tijdens deze fase wordt de software in één of meerder iteraties, of in geval van dit project incrementele fases ontworpen en gebouwd. Dit zal in combinatie met UML gedaan worden.
5. ***Implementation (relevant voor dit project)***
De term implementatie is verwarrend, omdat hiermee vaak het 'programmeerwerk' wordt bedoeld. In dit geval gaat het om het in gebruik nemen van de software. De overdracht naar OpenSesame ICT in dit geval.

Omdat DSDM in dit geval in combinatie met UML wordt gebruikt, is alleen gekozen om de fasen "Design and Build Iteration" voor het ontwerp en de bouw van de applicatie, en de "Implementation" fase voor het opleveren en overdragen van het product. Het gebruik van UML zal tijdens de ontwerp en ontwikkelfase gebruikt worden. Net als DSDM is gekozen omdat het vertrouwd is, en gebruikt is tijdens projecten, is UML van RUP dit eveneens. Tijdens de opleiding Informatica is RUP (UML) vaak gebruikt tijdens projecten voor het maken van functionele en technische ontwerpen.

Hoewel de andere fasen niet gebruikt worden op de manier zoals DSDM ze gebruikt, zitten er toch onderdelen tussen die gebruikt worden tijdens dit project. Denk hierbij aan planning, het bepalen van de software op basis van onderzoek en het vaststellen van eisen middels de MoSCoW methode. Ook het gebruik van timeboxing komt aan de orde voor de haalbaarheid (feasibility) van het succesvol uitvoeren van het project wat betreft de tijdsduur.

Tools voor DSDM

DSDM is een methode die toolonafhankelijk is. Dat betekent uiteraard niet dat er geen tools worden gebruikt, maar dat het met alle mogelijke tools te combineren is. Je zit niet vast aan bepaalde tools, wat enige mate van vrijheid geeft.

De technieken van DSDM

De techniek die het meeste geassocieerd wordt met DSDM is timeboxen in combinatie met MoSCoW. Bij de start van een increment zijn tijd en budget vastgelegd, dit wordt een timebox genoemd. De volgende stap is om de gewenste functionaliteit in kaart te brengen en de prioriteiten daarvan vast te leggen. MoSCoW is een manier om deze prioriteiten aan te geven. MoSCoW staat voor Must-have, Should-have, Could-have en Won't-have-this-time. Binnen een timebox wordt ervoor gezorgd dat de functionaliteiten met de hoogste prioriteit als eerste gemaakt worden, dit zijn de Must-haves. Als er tijd over is wordt er gewerkt aan de Should-haves enz.

Deze essentiële technieken zoals MoSCoW prioritering en timeboxing zullen dan ook gebruikt worden tijdens dit project.

Testen in DSDM

Hoewel het bij DSDM gebruikelijk is om gedurende de gehele levenscyclus te testen (zie: uitgangspunt 8 op blz. 18), is in samenspraak met de opdrachtgevers van school besloten, om hier een apart test plan en testrapport voor op te stellen. Voor het testen zijn twee beroepstaken opgegeven die elk apart beoordeeld zullen worden.

Waarom DSDM als ontwikkelmethodiek voor dit afstudeerproject?

De reden dat ik voor DSDM als ontwikkelmethodiek heb gekozen, en geen andere Agile methodiek, is omdat DSDM een ICT-project flexibeler kan maken door het systeem of project op te delen in zelfstandige eenheden. Hierdoor wordt het aanbrengen van tussentijdse veranderingen voor de ontwikkelaar eenvoudiger. Aangezien in dit project ook functionaliteit incrementeel ontwikkeld is, leek mij DSDM een goede methodiek om te gebruiken.

Een andere reden is de flexibiliteit van DSDM waardoor ik deze methodiek kan combineren met UML wat onderdeel is van Rational Unified Processing (RUP). RUP (UML) is tijdens de opleiding een aantal blokken het vaste stramien geweest tijdens de ontwikkeling van software, en is dus vertrouwd en bekend om mee te werken. Een derde reden is de vrijheid om zelf een keuze te maken welke tools ik kan en wil gebruiken om mijn doelen te realiseren.

Hoewel er meer ontwikkelmethodieken voor het incrementeel en iteratief ontwikkelen zoals bijvoorbeeld IAD, is de keuze toch op DSDM gevallen omdat deze ontwikkelmethodiek ook tijdens school projecten en tijdens mijn stageperiode gebruikt is. Buiten het feit dat het met deze ontwikkelmethodiek vertrouwd werken is, is de combinatie met UML ook zeer wenselijk, aangezien deze modeleringstaal centraal staat in de opleiding Informatica. Het werken met UML is daardoor ook bekend, wat de keuze makkelijker maakte.

NOTE * Budget speelt tijdens dit project geen rol. Wel is de tijd en de functionaliteit die gerealiseerd wordt binnen een bepaalde tijd van belang voor de borging van het op te leveren product. Hiervoor is het begrip timeboxing gebruikt voor het tijdig opleveren van verschillende functionaliteiten of artefacten.

6.2 Gebruikte hulpmiddelen en ontwikkeltools

NetBeans IDE

Netbeans is in dit project gebruikt als ontwikkelomgeving. Dit softwarepakket is een hulpmiddel om software te ontwikkelen, te compileren, te debuggen, en te implementeren. NetBeans is geschreven in Java, maar ondersteund ook een groot aantal andere talen. Tevens is er ook een grote hoeveelheid aan modules beschikbaar om NetBeans IDE uit te breiden.

UMLet

UMLet is gebruikt voor het ontwerpen van de modellen. Door middel van een eenvoudige syntax is het mogelijk om goede UML diagrammen te maken.

Office2010

Office2010 is gebruikt voor de verslaglegging van alle documentatie, het maken van de presentatie, en het gebruik van Visio voor een tijdsplanning.

Windows 7

Hoewel OpenIMS zowel op Linux als op Windows draait, is gekozen voor een Windows omgeving, omdat de andere gebruikte tools ook goed met Windows compatibel zijn er hierdoor minder kans is op complicaties.

6.3 Gebruikte software libraries

6.3.1 Het Zend Framework

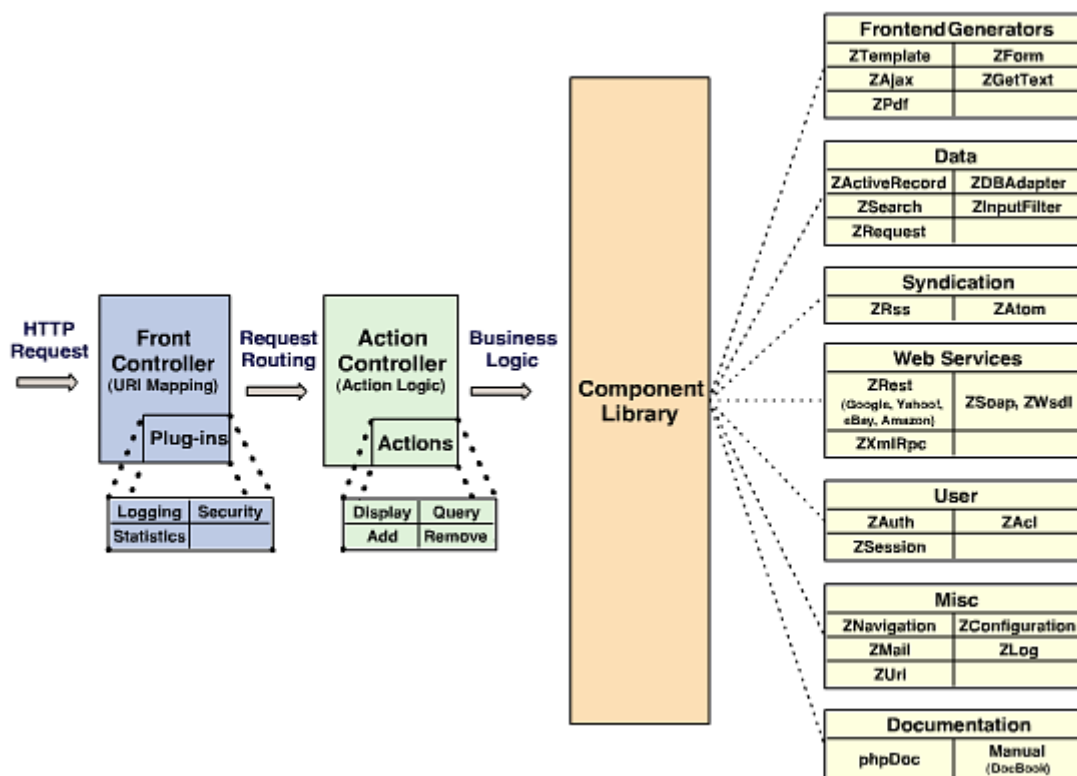
Het Zend Framework is een Open Source Framework, ontwikkeld in object georiënteerde PHP5 code door Zend Technologies (de makers van PHP). Het PHP Framework is gemaakt met de gedachte om het gebruik ervan zo simpel mogelijk te houden. Zend is het bedrijf achter de PHP engine, hetgeen suggereert dat ze heel goed zouden moeten weten wat er in het Zend Framework moet zitten.

Zend Framework streeft naar volledige dekking zodat ontwikkelaars niet het gehele internet hoeven af te zoeken naar oplossingen voor generieke problemen.

Zend Framework heeft ondersteuning voor meerdere databases, database objects, caching, validatie, autorisatie model en modules. Dit Framework heeft echter geen ondersteuning voor Ajax (Asynchroon Javascript en XML), en kent geen template engine.

Eenvoud en object-oriënteerde best practices van Zend

Het Zend Framework is tevens gebaseerd op eenvoud en object-oriënteerde best practices. Het is er op gericht om moderne Web 2.0 applicaties en web services te bouwen die veilig en betrouwbaar zijn. Zend Framework biedt ook ondersteuning aan algemeen beschikbare interfaces van leveranciers zoals Google, en Yahoo!

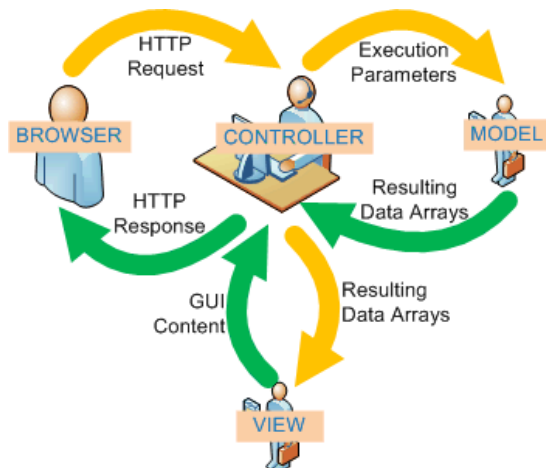


Zie Figuur 10: Een overzicht van het Framework en de diverse beschikbare componenten.

In figuur 10 is tevens in stappen te zien hoe de web services aangeroepen kunnen worden. Eerst worden er HTTP verzoeken gedaan, aan de Front Controller voor het loggen en de beveiliging van URI's (internet-protocolelementen verwant aan URL en URN). De Front controller doet op haar beurt weer verzoeken aan de Action controller voor het uitvoeren van acties als het toevoegen en verwijderen van een componenten. De Action controller gebruikt de Component Library met web services als REST en het Syndication format Atom.

Structuur Zend

Het Zend PHP framework is gebaseerd op het PHP MVC design patroon (Model-View-Controller). De verschillende componenten van het Zend PHP framework kunnen ook afzonderlijk gebruikt worden.



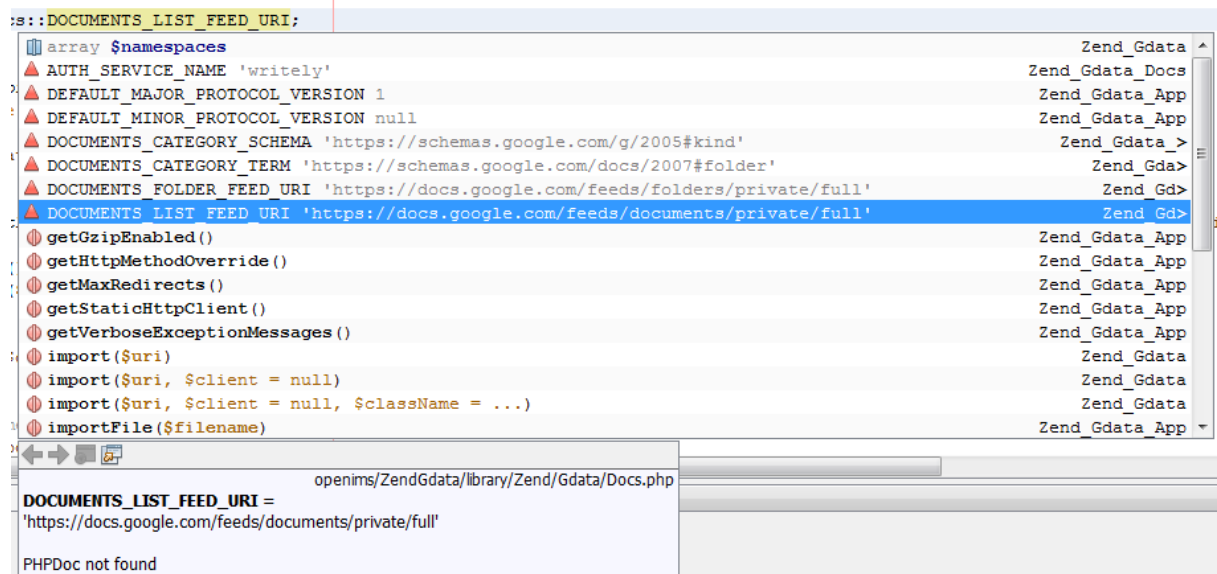
Figuur 11: In dit plaatje de ontwerpkeuze van Zend (MVC).

Model-view-controller (of MVC) is een ontwerppatroon ("design pattern") dat het ontwerp van complexe toepassingen opdeelt in drie eenheden met verschillende verantwoordelijkheden: datamodel (model), datapresentatie (view) en applicatielogica (controller). Het scheiden van deze verantwoordelijkheden bevordert de leesbaarheid en herbruikbaarheid van code. Het maakt ook dat bijvoorbeeld veranderingen in de gebruikersinterface niet direct invloed hebben op het datamodel en vice versa. Doordat Zend gebruik maakt van deze toepassing is de code goed leesbaar en gemakkelijker om mee te werken.

6.3.2 Gdata library van Zend

De Zend_Gdata component is een PHP 5-interface speciaal gemaakt door de makers van Zend Frameworks voor het verkrijgen van toegang tot Google Gegevens middels PHP. De Zend_Gdata component ondersteunt het toegang verkrijgen van services zoals het Atom Publishing Protocol (AtomPub).

Gdata is als aparte package te downloaden via de website van Zend. Na het downloaden van deze package moeten de bestanden uitgepakt en ergens op de harde schijf geplaatst worden. Vervolgens moet er via een regel in de php.ini file naar deze library verwezen worden, zodat de functies ook daadwerkelijk te bereiken zijn. In mijn geval is gebruik gemaakt van NetBeans. Dit is een hele goede en duidelijke ontwikkeltool, omdat direct met een aanroep te zien is, of de functies beschikbaar zijn. Dit is visueel aangetoond in figuur 12. [\[bron\]](#)



Figuur 12: Gdata library van Zend wordt aangeroepen met NetBeans IDE. Beschikbare feeds zijn zichtbaar.

Gebruik van de Gdata library

De Google Docs List API staat toe dat cliënt applicaties documenten kunnen uploaden naar Google Docs en zet deze in een lijst in de vorm van Google Data API ("Gdata") feeds. Vervolgens kan met een cliënt-applicatie een lijst van documenten, van een bepaalde gebruiker, en de inhoud van een bestaand document opgevraagd worden.

Om een lijst van Google documenten van een gebruiker te verkrijgen kan de functie `'getDocumentListFeed()'` van de Docs service gebruikt worden. Deze service zal een zogeheten `Zend_Gdata_Docs_DocumentListFeed` object teruggeven met een lijst van documenten van de Google User die zich heeft geauthentiseerd. Een voorbeeld hiervan is te zien in figuur 13.

```
1. $service = Zend_Gdata_Docs::AUTH_SERVICE_NAME;
2. $client = Zend_Gdata_ClientLogin::getHttpClient($user, $pass, $service);
3. $docs = new Zend_Gdata_Docs($client);
4. $feed = $docs->getDocumentListFeed();
```

Figuur 13: Het gebruik van de Google Documents List API in combinatie met de Gdata library.

Het resulterende `Zend_Gdata_Docs_DocumentListFeed` object vertegenwoordigt de reactie van de server. Deze feed bevat een lijst van `Zend_Gdata_Docs_DocumentListEntry` objecten (`$feed->entries`), die elk afzonderlijk een Google-document vertegenwoordigen. [\[bron\]](#)

7 Ontwerpfase (fase 4)

Op basis van interviews met de opdrachtgever(-s), is allereerst een lijst opgesteld met wensen en eisen. Vervolgens zijn deze wensen en eisen met behulp van de MoSCoW methode onderverdeeld. Na het opstellen van de lijst met eisen is tevens een akkoord gegeven door de opdrachtgever en begeleider van OpenSesame ICT.

7.1 MoSCoW prioritering

*** Must have ***
- Authenticatie functionaliteit
- Het downloaden van documenten naar Google Docs
- Het uploaden van documenten naar OpenIMS
- Update functionaliteit van de documenten en de inhoudelijke content
- Het verwijderen van documenten
- Het synchroniseren van de documenten tussen deze twee systemen

*** Should have ***
- Het opslaan van documenten in sitecollection(s); ook wel folders genoemd
- Het koppelen van een sitecollection aan een folderid van OpenIMS
- Synchronisatie functionaliteit aanpassen om ook documenten in folders te syncen

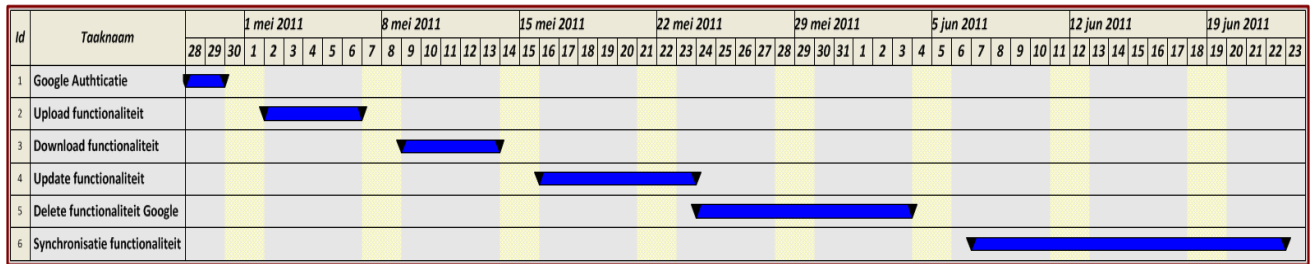
*** Could have ***
- Folder management. Het uploaden en downloaden een folder
- Het automatisch afhandelen van document(en) en folder(s) share rechten. Het uitdelen van rechten is mogelijk, indien dit handmatig wordt gedaan in Google Docs
- Het updaten van share rechten
- Het verwijderen van share rechten
- Document revisie (het ophalen van de history van document(en))

*** Won't have this time, but would like in the future ***
- Synchronisatie met een derde softwarepakket
- Mogelijkheid om spreadsheets te up en downloaden
- Meerdere authenticatie opties (Authsub, OAuth)

7.2 Timeboxes

In figuur 11 zijn de timeboxes aangegeven van de verschillende incrementen van de ontwikkelfase. Zo worden alle belangrijke incrementen van de ontwikkelfase onderverdeeld in verschillende timeboxes waarbij alle functionaliteit samenkomt in 'Timebox 6'. Voor het verloop van de timeboxes (zie §8.7), waar aangegeven wordt of de timeboxes behaald zijn binnen de geplande tijd, of dat hier uitloop heeft plaatsgevonden. Volgens planning zijn er 40 dagen voor de ontwikkelfase. Van 28-04-2011 t/m 22-06-2011 (weekenden zijn hierbij niet meegerekend). [\[bron\]](#)

Alleen bij de ontwikkelfase is gekozen voor een gedetailleerde planning. De reden hiervan is dat bij de ontwikkelfase het risico vele malen hoger ligt op uitloop, als de andere onderdelen in de planning. Het gebruik van timeboxes vereist overigens specifieke datums voor een tijdige overgang naar een volgend geplande timebox.



Figuur 14: Toont een chantt-chart diagram met de zes timeboxes. Aan de linkerkant staat de omschrijving van elke timebox, en rechts is in het blauw een gedetailleerde planning (tijdlijn) aangeduid op welke dag een timebox begint en eindigt.

Zoals in figuur 14 te zien is, duurt niet elke timebox even lang. Voor elke timebox staat een aantal dagen. Bij de einddatum van elke timebox moet er begonnen worden aan een nieuwe timebox, ook al is de vorige timebox nog niet geheel afgerond. Een timebox die niet is afgerond, kan later alsnog afgerond worden indien er nog tijd over is.

7.3 Functioneel systeemontwerp

Om het algehele synchronisatie proces werkend te kunnen krijgen, moesten er aparte functies worden geschreven om deze later in de synchronisatie functionaliteit toe te kunnen voegen. Deze aparte onderdelen (functies) zijn uitgewerkt in use-case scenarios. De volgende use-cases zullen behandeld worden;

[1] Het autoriseren bij gmail via CliëntLogin. Dit is nodig om überhaupt bij de documenten te komen om vervolgens enige andere functie uit te kunnen voeren.

[2] Het uploaden van één of meerdere documenten. Deze functionaliteit upload een bestand vanuit Google Docs naar OpenIMS.

[3] Het downloaden van één of meerdere documenten. Deze functie wordt gebruikt om het document vanuit OpenIMS in Google Docs te plaatsen.

[4] Het updaten van één of meerdere documenten. Dit is nodig om na elke wijziging(en) de documenten up-to-date te houden.

[5] Het verwijderen van documenten uit Google Docs. Deze functie is nodig indien een actor zijn of haar document(en) niet meer wil bewaren.

[6] Het verwijderen van documenten uit OpenIMS. Deze functie is nodig indien een actor zijn of haar document(en) niet meer wil bewaren.

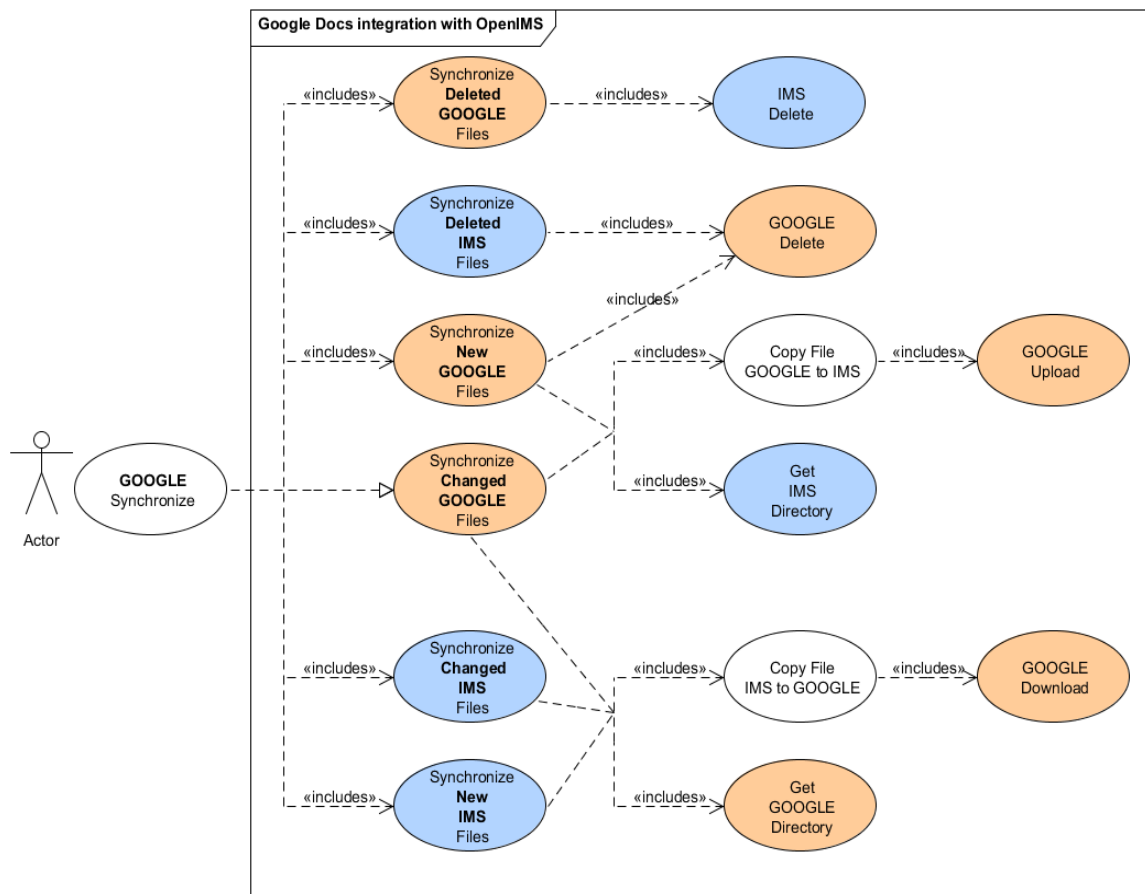
[7] De synchronisatie is het algehele proces met alle hierboven genoemde functies bij elkaar. De synchronisatie functionaliteit controleert de status en update, verwijdert, of voegt documenten toe zodat de inhoud aan documenten in OpenIMS inherent blijft aan die van Google Docs en visa versa (zie: Figuur 10).

Punt 7 is het algehele synchronisatie proces. Alle andere sub-functies zijn ontworpen om de synchronisatie functionaliteit te realiseren. Om die reden is alleen punt 7 als functioneel ontwerp opgenomen als meest relevante. Hiervoor is gekozen om zo het document overzichtelijk te houden. Alle afzonderlijke sub-functies zijn ook beschreven met een use-case, use-case diagram en sequencediagram. Zie *Bijlage 2 van dit document*.

Use-case Synchronisatie:

Deze Use-case beschrijft het synchronisatieproces. Ervan uitgaande dat een actor de synchronisatie start en de folder-state-maps hun werk gedaan hebben (zie paragraaf 8.6), kunnen er verschillende functies aangeroepen worden, afhankelijk van de wijzigingen die in OpenIMS of Google Docs hebben plaatsgevonden. Als er documenten in Google Docs zijn verwijderd, dan zal dit bij het synchroniseren herkend worden, en het bestand ook uit OpenIMS verwijderd worden. Hetzelfde geldt voor een nieuwe en/ of gewijzigde documenten. Ook andersom, van OpenIMS naar Google Docs zal het volgens deze procedure werken.

Als een document nieuw is aangemaakt in Google Docs, wordt de upload functie aangeroepen, om het document te uploaden naar OpenIMS. Als een document nieuw is aangemaakt in OpenIMS, wordt de download functie aangeroepen, om het document naar Google Docs te downloaden. De "Get IMS Directory" en "Get Google Directory" zorgen ervoor dat de documenten in een desbetreffende folder ook te bereiken zijn voor het uploaden en downloaden naar beide systemen (zie: Figuur 15). [\[bron\]](#)



Figuur 15: Synchronisatie proces weergeven in een Use-case.

Use-case beschrijving Synchronisatie:

Deze Use-case beschrijving beschrijft alle stappen die worden gedaan bij het uitvoeren van de synchronisatie. Hoewel de synchronisatie door middel van een batch bestand kan worden uitgevoerd, kan dit ook handmatig worden gedaan. In deze beschrijving is ervoor gekozen dat een klant of medewerker van OpenSesame ICT als actor fungeert, en de synchronisatie handmatig start.

De eerste vier stappen (1 – 4) is het verkrijgen van de huidige en laatste status van de folders van zowel OpenIMS als Google Docs. In de stappen 6 tot en met 11 wordt nagegaan of dat er documenten uit OpenIMS of Google verwijderd zijn, of er eventueel nieuwe documenten zijn aangemaakt, of dat er documenten gewijzigd zijn. In de laatste vier stappen (13 – 15) wordt de huidige status van de folders verkregen en opgeslagen om zowel OpenIMS als Google Docs weer up-to-date te krijgen voor een volgende synchronisatie die uitgevoerd kan worden. In figuur 16 zijn deze stappen terug te vinden in een Use-case beschrijving.

ID	7
Use-case	Google Synchronize
Actor	Klant, Medewerker osict (tester)
Pre-conditie	De klant en of medewerker wil dat de documenten in OpenIMS gelijk blijven aan die van Google Docs en visa versa.
Werkwijze	[1] Authenticeren bij Google [2] Verkrijgen laatste folder-state-map van de Google-repository folder [3] Verkrijgen laatste folder-state-map van de OpenIMS-DMS folder [4] Verkrijgen huidige folder-state-map van de Google-repository folder [5] Verkrijgen huidige folder-state-map van de OpenIMS-DMS folder [6] Synchroniseer verwijderde Google-repository documenten [7] Synchroniseer verwijderde OpenIMS documenten [8] Synchroniseer nieuwe OpenIMS documenten [9] Synchroniseer nieuwe Google-repository documenten [10] Synchroniseer gewijzigde OpenIMS documenten [11] Synchroniseer gewijzigde Google-repository documenten [12] Verkrijgen huidige folder-state-map van de Google-repository folder [13] Opslaan huidige folder-state-map van de Google-repository folder [14] Verkrijgen huidige folder-state-map van de OpenIMS-DMS folder [15] Opslaan huidige folder-state-map van de OpenIMS-DMS folder
- 80% scenario	De actor voert de synchronisatie uit om de documenten in beide systemen up-to-date te houden.
- 20% scenario	De actor kan er ook voor kiezen de synchronisatie uit te laten voeren door een batch component in OpenIMS aan te maken en deze elke 5 minuten te laten draaien.
Post conditie	De documenten zijn gesynchroniseerd en indien wijzigingen zijn gemaakt, succesvol geupdate in beide systemen.

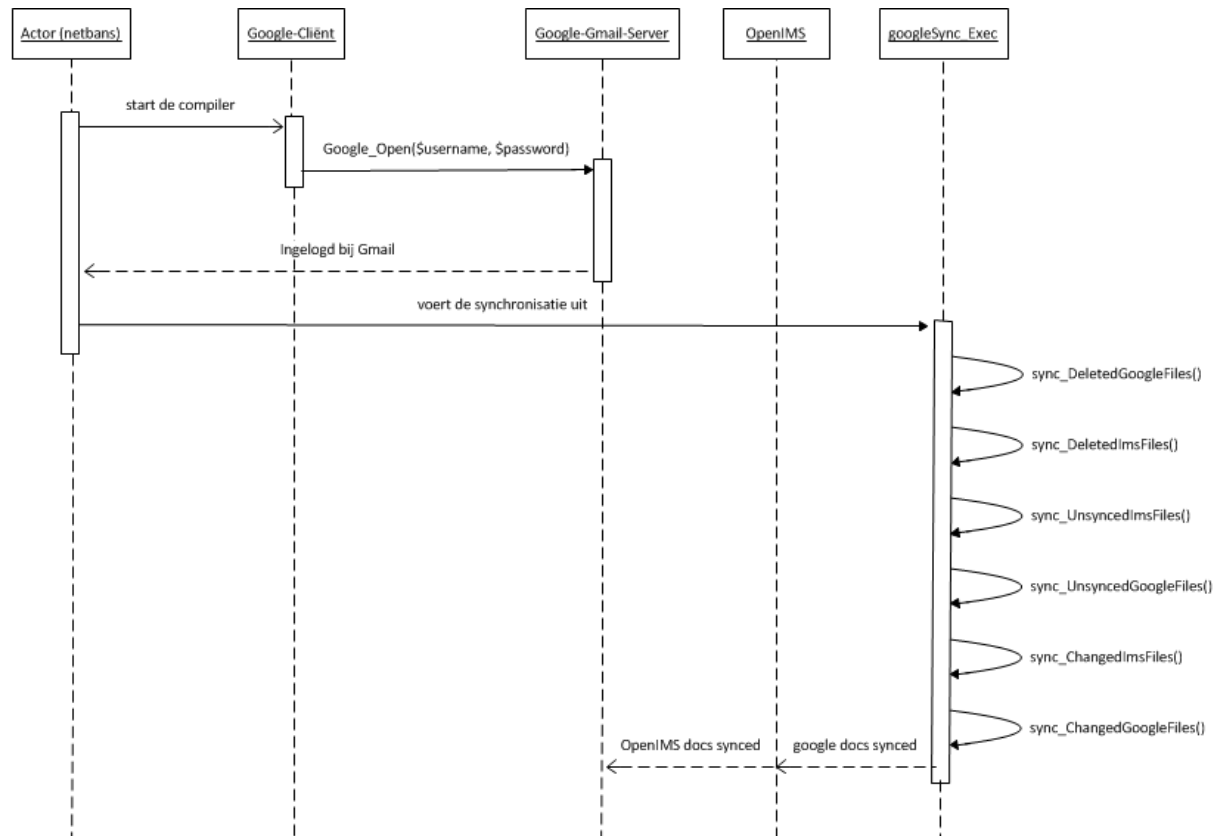
Figuur 16: De Use-case beschrijving van de synchronisatie functionaliteit.

7.4 Technisch systeemontwerp

In het Technisch systeemontwerp komen de technische details aan de orde. Het technisch ontwerp bevat o.a. sequence diagrammen. [\[bron\]](#)

Sequence diagram synchronisatieproces

[7] Sequence: De synchronisatie (algehele proces met alle gedefinieerde functies)

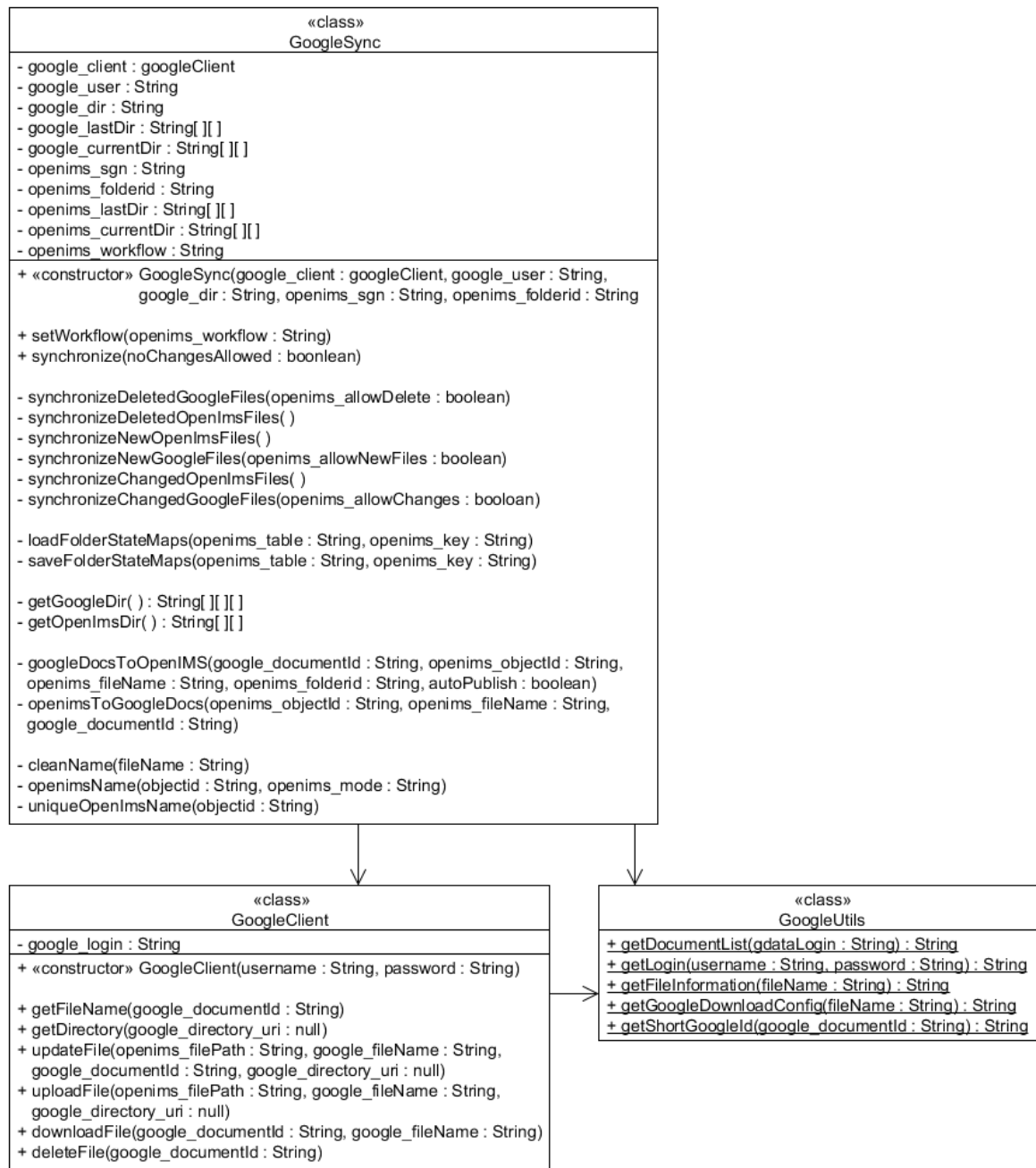


De functies hieronder aangegeven worden aangeroepen voor het uitvoeren van de synchronisatie:

- (1) `sync_DeletedGoogleFiles($google_lastDir, $google_currentDir, $sgn, $openims_lastDir, $openims_currentDir, $openims_thespecs);`
- (2) `sync_DeletedImsFiles($google, $google_dir, $google_lastDir, $google_currentDir, $openims_lastDir, $openims_currentDir);`
- (3) `sync_UnsyncedImsFiles($google, $google_dir, $google_lastDir, $google_currentDir, $sgn, $openims_currentDir);`
- (4) `sync_UnsyncedGoogleFiles($google, $google_dir, $google_lastDir, $google_currentDir, $sgn, $openims_folderid, $openims_lastDir, $openims_currentDir, $openims_thespecs);`
- (5) `sync_ChangedImsFiles($google, $google_dir, $google_lastDir, $google_currentDir, $sgn, $openims_lastDir, $openims_currentDir);`
- (6) `sync_ChangedGoogleFiles($google, $google_dir, $google_lastDir, $google_currentDir, $sgn, $openims_folderid, $openims_lastDir, $openims_currentDir, $openims_thespecs);`

Figuur 17: Synchronisatie proces weergegeven in een sequencediagram.

Class diagram algehele synchronisatie functionaliteit, zie figuur 18



Figuur 18: class diagram van alle classes en haar functies.

8 Ontwikkelfase (fase 4)

In dit hoofdstuk worden de verschillende incrementen (sub-producten) beschreven, die tijdens de ontwikkelfase zijn programmeert. Er zijn zeven van deze incrementen beschreven waarbij de eerste zes een onderdeel zijn van het zevende increment waarbij alle programmatuur samen komt tot één eindproduct.

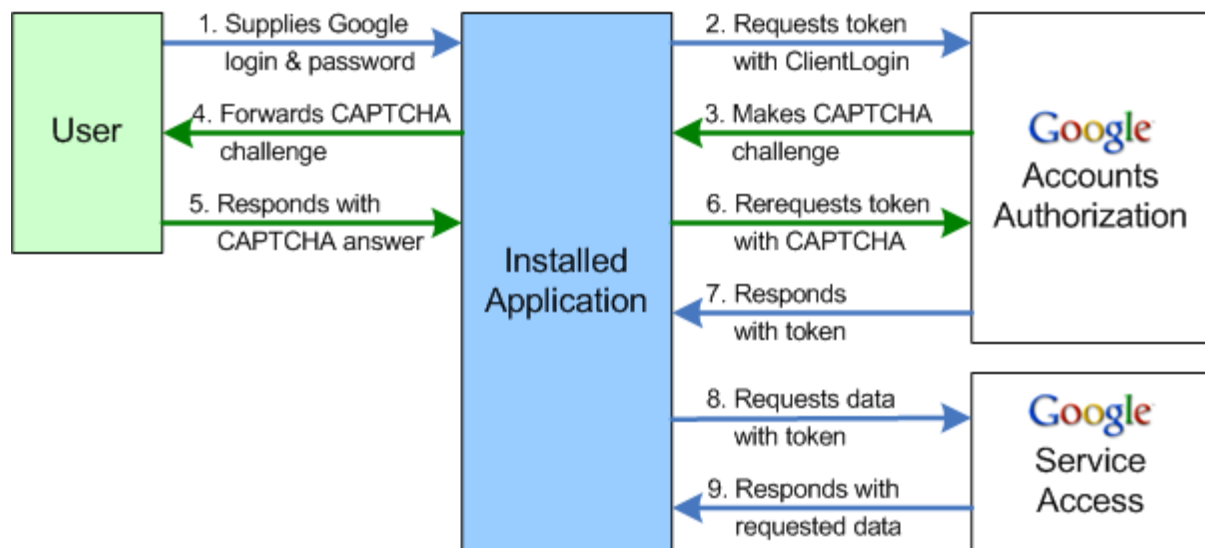
Allereerst komt de Google authenticatie aan bod, vervolgens het uploaden, het downloaden, het updaten, en het verwijderen van documenten. Als laatste, en zevende increment is de synchronisatie functionaliteit beschreven waarin alle eerder genoemde functionaliteit in terug komt. [\[bron\]](#)

8.1 Authenticatieproces Google Docs (1ste increment)

Het eerste onderdeel van de ontwikkelfase was de authenticatie bij Gmail. De PHP API ondersteunt twee authenticatiemogelijkheden. CliëntLogin en AuthSub. Buiten deze twee opties, is er ook nog OAuth. Deze wordt niet door de PHP API ondersteund en is dus niet bruikbaar.

Er moest een keuze gemaakt worden in het gebruik van CliëntLogin of AuthSub. Omdat met AuthSub na elke aanvraag autorisatie verleend moet worden tot de Google documenten via de browser, was dit geen optie. De bedoeling is dat de synchronisatie automatisch op de achtergrond met behulp van een batch-bestand geladen wordt, daarom was de enige optie de authenticatie met behulp van CliëntLogin.

Met CliëntLogin wordt éénmalig de account gegevens van het Google mail account in de code gezet, waarna er altijd toegang is, en daarmee dus aanvragen gedaan kunnen worden, zonder dat er elke keer eerst toestemming verleend dient te worden.



Figuur 19: CliëntLogin autorisatie procedure.

Clientlogin: (zie: Figuur 19)

- De gebruikersnaam en het paswoord worden verzonden naar Google over een HTTPS (beveiligde) verbinding.
- Het is mogelijk dat Google een CAPTCHA terugstuurt waarop de gebruiker moet antwoorden voordat er een sleutel wordt ontvangen.
- Google antwoordt met een sleutel waarna alle volgende aanvragen moeten gebeuren.

Doordat de gegevens via een HTTPS verbinden lopen is er sprake van encryptie. Google bepaald in deze zelf wanneer er een CAPTCHA vereist is (bijv. als Google de gebruiker van spam zal verdenken). Voor zover bekeken kan worden, hoeft niet elke aanvraag “gesigned” te worden zoals bij OAuth wel het geval is (zie: *Bijlage 1.2*). De verkregen sleutel bij ClientLogin is enkel geldig voor een bepaalde tijd, afhankelijk van de gebruikte API.

Buiten ClientLogin zijn er nog twee andere authenticatie-mogelijkheden. Hoewel deze niet geschikt zijn voor dit project, zijn ze wel onderzocht. Voor een overzicht hiervan zie: *Bijlage 1.2*.

8.2 Upload functionaliteit (2de increment)

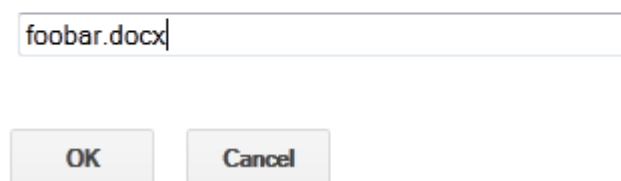
Upload functionaliteit betreft het uploaden van bestanden van Google naar OpenIMS. Met behulp van de DOCUMENTS_LIST_FEED_URI van Zend_Gdata_Docs wordt een lijst met documenten opgehaald. Vervolgens worden deze bestanden met een functie, genaamd “uploadFile”, geupload naar een aangegeven folder in het OpenIMS DMS. De functie “uploadFile” heeft overigens vier parameters die meegegeven worden, namelijk:

\$openims_filePath (Om het OpenIMS pad mee te geven waar de bestanden gedropt dienen te worden).
\$Google_fileName (De bestandsnaam van Google om te weten over welk bestand het gaat).
\$Google_mimetype (De mimetype, om naast de bestandsnaam ook de content type mee te kunnen geven).
\$Google_directory_uri (De Google lijst met feeds waar de bestanden opgehaald moeten worden).

Het is echter wel belangrijk, wanneer er bestanden van Google naar OpenIMS overgezet worden, dat deze bestanden een extensie hebben. Bij het aanmaken van een Google document in Google Docs is het van belang dat de gebruiker een extensie meegeeft. Als dit niet gedaan wordt, zal OpenIMS dit zien als een bestand zonder extensie, en dit bestand niet kunnen openen. Het is namelijk in Google Docs ook mogelijk om een bestand “foobar” (als voorbeeld) te noemen in plaats van bijvoorbeeld: “foobar.docx”. (zie: Figuur 20).

Rename Document

Enter a new document name:



Figuur 20: Het meegeven van een extensie bij het aanmaken van een nieuw Google document. Op deze manier is het mogelijk om geüploade bestanden te openen in OpenIMS.

8.3 Download functionaliteit (3de increment)

De download functionaliteit “download bestanden” van OpenIMS naar Google Docs is `downloadFile()` en heeft twee parameters.

\$Google_documentId (omdat alle documenten in Google Docs een uniek id hebben).

\$Google_fileName (om de bestandsnaam mee te geven welke bij een uniek id hoort).

8.4 Update functionaliteit (4de increment)

De functie update functionaliteit betreft het updaten van documenten nadat er wijzigingen hebben plaatsgevonden.

Update Google Docs

Het updaten van Google werkt in dit opzicht iets anders als het updaten van documenten in OpenIMS. Doordat de PHP API geen enkele ondersteuning biedt op het gebied van updaten, is bekeken of Zend dit wel biedt. Na de code van de Zend Gdata library bekeken te hebben, en het zoeken op internet is gebleken dat Zend geen ondersteuning voor het updaten van documenten (content) geen code heeft ontwikkeld. Doordat het teveel tijd zou gaan kosten om in de Gdata library van Zend te gaan programmeren, voor deze functionaliteit, is er in samenspraak met de opdrachtgever voor gekozen om dit anders aan te pakken. Om een document met de inhoudelijke content te updaten wordt het oude document verwijderd en een nieuwe aangemaakt in Google Docs als de update wordt uitgevoerd.

De volgende functie toont dit aan:

```
// for updating documents. New update functionality will be edit in new PHP Document
List API 2012.
Function updateFile($openims_filePath,$Google_fileName, $Google_documentId, $Google_directory_uri =
null) {
    $this->deleteFile($Google_documentId);
    return $this->uploadFile($openims_filePath, $Google_fileName);
}
```

Hier is te zien dat de deleteFile functie een Google_documentId meekrijgt zodat bekend is wel document verwijderd is uit Google Docs, en vervolgens de nieuwe gewijzigde versie vanuit OpenIMS weer geüpload wordt naar Google Docs.

Update OpenIMS

OpenIMS gebruikt tijdens het updaten de functie “MB_Flush” welke de gegevens zal flushen waarna de functie “getGoogleDir” aangeroepen wordt om de nieuwe gewijzigde documenten op te halen. De “getGoogleDir” roept op haar beurt weer de functie \$item->getUpdated()->getText() aan om de gewijzigde content te kunnen updaten.

8.5 Delete functionaliteit (5de increment)

Voor het verwijderen van documenten gebruiken Google en OpenIMS ieder haar eigen functies.

Google Docs verwijderen

Google Docs gebruikt de standaard delete functie “deleteFile” van de PHP API. Het unieke “docid” wordt meegegeven om te kunnen bepalen wel docid niet meer bestaat na het synchroniseren.

OpenIMS verwijderen

OpenIMS heeft een eigen functie die hiervoor ontwikkeld is. Dit is de functie “IMS_Delete”. Na het uitvoeren van deze functie wordt de “MB_Flush” aangeroepen voor het flushen van de database.

8.6 Synchronisatie functionaliteit (6de increment)

In deze paragraaf zal de synchronisatie uiteengezet worden. Er wordt ingegaan op de metadata van zowel OpenIMS als Google, over waar en hoe deze data wordt opgehaald, en waar deze data uit bestaat om bestanden met elkaar te kunnen vergelijken. Dit is immers essentieel voor de synchronisatie, omdat het belangrijk is te weten wat de huidige status is, en of er tijdens het uitvoeren van een nieuwe synchronisatie wijzigingen zijn aangebracht.

Google synchronisatie

De definitie van de synchronisatie behandelt in dit geval altijd twee wegen. Een bron is een map in het DMS van OpenIMS en het andere is de root directory van Google Docs. In het OpenIMS DMS kan de gebruiker dus zelf een folder kiezen welke hij of zij wil laten synchroniseren. In Google staan de documenten standaard in de hoofddirectory.

De synchronisatie tussen Google en OpenIMS is 'bi directioneel'. Dit houdt in dat, als er wijzigingen aangebracht zijn in een folder van OpenIMS, deze veranderingen ook in Google plaats zullen vinden. Andersom zullen de wijzigingen in Google documenten ook aangepast worden in OpenIMS. Een derde mogelijkheid is dat zowel in Google als OpenIMS hetzelfde bestand gewijzigd is sinds de laatste synchronisatie. Dit is een geval van twijfel. De versie van het bestand dat in OpenIMS gewijzigd is, is dominant, wat inhoudt dat deze dan overruled wordt door dat van Google Docs. De keuze om OpenIMS dominant aan Google Docs te laten zijn, is omdat het OpenIMS DMS de basis blijft van de gebruiker voor documentenbeheer. Om problemen te voorkomen bij het overschrijven van bestanden is het verstandig dat als er meerdere gebruikers tegelijk documenten willen wijzigen, zij dit in Google Docs doen, aangezien Google Docs real-time editing heeft.

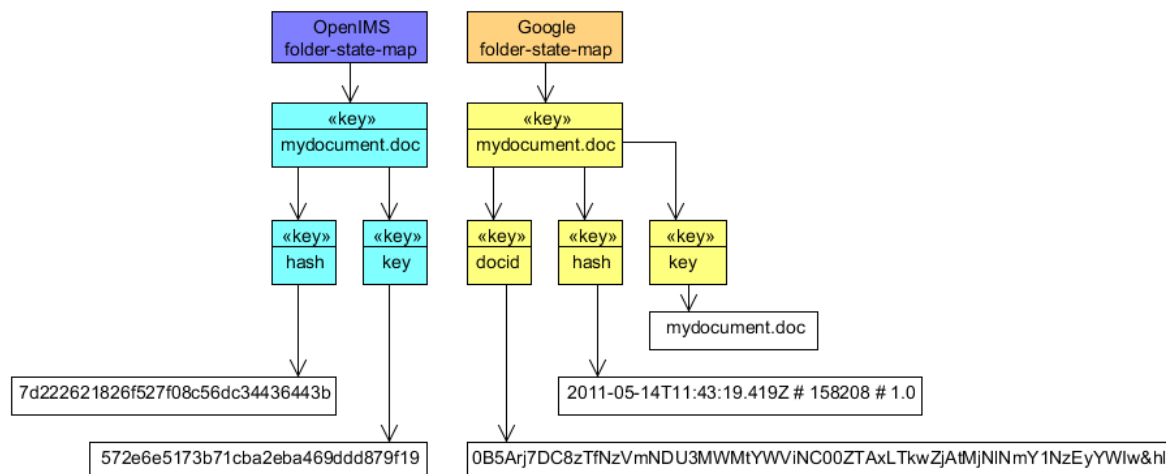
Folder-state-map algemeen

Om de synchronisatie goed te laten verlopen, is het nodig om metadata op te slaan met de status van elke laatste synchronisatie. Deze metadata wordt gebruikt om verschil te herkennen van de metadata welke is aangemaakt vóór de huidige synchronisatie. Deze metadata bevat inhoud over welke bestanden er in OpenIMS en welke in Google opgeslagen zijn. Na elke synchronisatie zal dan zowel de folder van OpenIMS als de root directory van Google Docs gelijk aan elkaar zijn. Als er bijvoorbeeld nieuwe bestanden zijn aangemaakt in een folder of bestanden zijn verwijderd, zal de synchronisatie samenstelling de metadata dan ook niet veranderen. Op basis van de laatste synchronisatie en de huidige kunnen de verschillen namelijk bepaald worden.

De synchronisatie functionaliteit is zo gemaakt dat alleen OpenIMS zich bewust is van de samengestelde metadata. Om dit doel te bereiken, moet deze metadata dus in OpenIMS opgeslagen worden. Deze samengestelde metadata wordt "folder-state-map" genoemd. Nadat een synchronisatie proces klaar is, is de laatste stap het opslaan van zowel de OpenIMS folder als de root directory van Google. Beide folder-state-maps van een gesynchroniseerde samenstelling zijn gebonden aan een unieke index om zo de samenstelling te behouden.

Een unieke index is aangemaakt voor elk gesynchroniseerde samenstelling. Deze index bevat het unieke id van de OpenIMS folder, en het id van de Google root directory.

Om meer inzicht te krijgen wat een folder-state-map is, is het belangrijk om te weten dat er een vermelding voor elk document van de gerelateerde map is. De naam van het document, inclusief de file-extensie, is de index van een item voor een document. Een folder-state-map van Google is niet precies even gestructureerd als een folder-state-map van een OpenIMS map. Figuur 21 toont een folder-state-map van Google (oranje) en één van een OpenIMS map (blauw) met van elk document van OpenIMS hash en key informatie, en van Google de hash, een key en een uniek document id.



Figuur 21: Overzicht van de metadata van OpenIMS en Google Docs.

Folder-state-map Google

De folder-state-map van Google heeft drie subitems: De eerste is de naam "key" en de waarde hiervan is altijd de naam van het document met extensie. De tweede vermelding is de "hash". De hash bevat een gegenereerde hash waarde. Dit is een string die bestaat uit de volgende eigenschappen van een bestand, namelijk: laatste datum van wijziging, de grootte van het bestand, de grootte van de inhoud (content stream lengte), en als laatste een versie nummer. De metadata is van een specifiek document. De hash van het bestand is uniek. Dit maakt het dus mogelijk om de verschillen tussen twee bestanden te bepalen. Met andere woorden: als het bestand gewijzigd wordt, dan zal ook de hash waarde veranderen.

Folder-state-map OpenIMS

De folder-state-map van een OpenIMS heeft dezelfde subitems (op docid na) als Google. De waarde van de hash heeft ook dezelfde functie. De waarde van de sleutel is echter anders. Het is in dit geval niet de volledige naam van het bestand, maar het is het document-id van een OpenIMS document. Dat klinkt logisch aangezien het OpenIMS DMS het mogelijk maakt om meerdere bestanden in een folder te hebben die dezelfde naam hebben. Om het verschil dus te bepalen, wordt dit aan de hand van een uniek document id gedaan. Ook bij Google is het mogelijk om twee dezelfde bestanden met dezelfde naam te hebben doordat ze ieder uniek zijn met een eigen id.

Voor het genereren van folder-state-maps zijn functies ontwikkeld voor zowel Google als voor OpenIMS. De Google variant is genaamd "getGoogleDir" en de OpenIMS variant "getIMSDir".

De synchronisatie

Er zijn om precies te zijn zes functies welke de core werkzaamheden uitvoeren van de synchronisatie (zie: Figuur 22). In principe zijn er drie basisfuncties die beide beschikbaar zijn voor "OpenIMS naar Google" en "Google naar OpenIMS". Deze zijn voor nieuwe, gewijzigde en verwijderde bestanden. Zoals het Use-case diagram (7.3) en het sequence-diagram (7.4) al aangaven, is de synchronisatie PHP functie hieronder weergegeven.

Zoals al eerder is vermeld, is de synchronisatie functie de meest belangrijke. De eerste stap is het laden van de folder-state-maps van beide systemen (regel regel 8). In dit geval zijn er view folder-state-maps geladen, elk voor de laatste en de huidige status. Vervolgens worden de zes sub(functies) aangeroepen (regel 10-15). Deze

manipuleren de folder-state-maps (welke onderdeel zijn van de GoogleSync class), zodat uiteindelijk de nieuwe status opgeslagen kan worden met de functie "saveFolderStateMaps (regel 19).

```
Class:      GoogleSync
Functie:    synchronize

1  // this is main function which includes all sub-functions for synchronization.
2  function synchronize($openims_allowChanges=true) {
3      $openims_key = "Google User: $this->Google_user" . " # " . $this->Google_dir
4          . " # " . $this->openims_sgn . " # " . $this->openims_folderId;
5      $openims_table = "ims_" . $this->openims_sgn . "_Googlesyncdir";
6
7      N_Log("Google", "Sync started - " . $openims_key);
8      $this->loadFolderStateMaps($openims_table, $openims_key);
9
10     $this->synchronizeDeletedGoogleFiles($openims_allowChanges);
11     $this->synchronizeDeletedOpenImFiles();
12     $this->synchronizeNewOpenImFiles();
13     $this->synchronizeNewGoogleFiles($openims_allowChanges);
14     $this->synchronizeChangedOpenImFiles();
15     $this->synchronizeChangedGooleFiles($openims_allowChanges);
16
17     $this->saveFolderStateMaps($openims_table, $openims_key);
18     N_Log("Google", "Sync completed - " . $openims_key);
19 }
```

Figuur 22: Synchronisatie functie in PHP.

Overigens is het zinvol om in het kort uit te leggen wat deze zes sub-synchronisatie functies doen:

De functie "synchronizeDeletedGoogleFiles" werkt als volgt: de laatste folder-state-maps wordt vergeleken met de huidige. Als een specifiek bestand in beide systemen aanwezig is, en deze verwijderd wordt in Google, zal dit bestand ook uit OpenIMS verwijderd worden. De functie "synchronizeDeletedOpenImFiles" werkt precies andersom. Als een bestand uit de OpenIMS folder verwijderd is, zal ook dit bestand uit Google Docs verwijderd worden.

De functie "synchronizeChangedGoogleFiles" werkt in principe hetzelfde als "synchronizeDeletedGoogleFiles". Als een bestand in Google is gewijzigd, zal deze gewijzigde versie gedownload worden, en de oude versie die in OpenIMS aanwezig is vervangen. Deze zelfde methode werkt ook weer andersom met de functie "synchronize-ChangedOpenIMSFiles".

De functie "synchronizeNewGoogleFiles" spreekt voor zich. Het kopieert nieuw aangemaakte bestanden in Google naar OpenIMS en de functie "synchronizeNewOpenImFiles" zal nieuw aangemaakte bestanden van OpenIMS naar Google Docs kopiëren.

8.7 Verloop Timeboxes (incrementen)

Timeboxes	Start datum	Geplande Einddatum	Status verloop	Dagen exact	Betreft
1 ^{ste} increment	28-04-2011	29-04-2011 (2 dagen)	Volgens planning	3 dagen	Authenticatie
2 ^{de} increment	02-05-2011	06-05-2011 (5 dagen)	Volgens planning	5 dagen	Upload
3 ^{de} increment	07-05-2011	13-05-2011 (6 dagen)	2 dagen extra	8 dagen	Download
4 ^{de} increment	16-05-2011	23-05-2011 (6 dagen)	Volgens planning	6 dagen	Update (Beide)
5 ^{de} increment	24-05-2011	06-06-2011 (9 dagen)	Volgens planning	5 dagen	Delete Google
6 ^{de} increment	07-06-2011	22-06-2011 (12 dagen)	2 dagen ingelopen	10 dagen	Synchronisatie

De twee dagen die uitlopen zijn op twee timeboxes, is ingelopen op een andere timebox. De einddatum (22-06-2011) kon hierdoor volgens planning behaald worden. In totaal waren 40 dagen ingepland voor de ontwikkelfase. Weekenden zijn hier overigens niet meegerekend, alleen werkdagen.

9 Implementatiefase (fase5)

9.1 Oplevering producten

In het OpenIMS DMS van OpenSesame ICT heeft iedere werknemer, maar ook stagiaires en afstudeerders een eigen privé map om zijn of haar gemaakte documenten en/ of code te uploaden. Op deze manier is er een extra back-up aanwezig en kan men na vertrek van de stagiair of afstudeerder nog steeds bij het opgeleverde materiaal (zie: Figuur 23)

Hoofdfolder > Prive > Danny van Tol > Afstuderen > google code					
Document ▲	Status	Versie	Laatst gewijzigd	Toegewezen	
 shortmanual-gdocs-ims	Gewijzigd	0.6	8 Jul 2011 15:41	Danny van Tol	
 uuse googledocs	Nieuw	0	1 Jun 2011 15:00	Danny van Tol	
 uuse googledocs sync	Gewijzigd	0.1	3 Jun 2011 11:58	Danny van Tol	
 uuse googledocs sync oo	Nieuw	0	7 Jul 2011 15:33	Danny van Tol	
 uuse googledocs sync oo test	Nieuw	0	7 Jul 2011 15:32	Danny van Tol	

Figuur 23: OpenIMS DMS van OpenSesame ICT. Hier staat de Google code + handleiding geüpload.

9.2 Handleiding

Bij de oplevering van de code is ook een korte handleiding geschreven hoe Google Docs werkend te krijgen met OpenIMS (zie: Figuur 24). Dit is gedaan voor de ontwikkelaars van OpenSesame ICT, indien zij dit voor henzelf, of een klant operationeel willen hebben.

```
/*
 * Handleiding door: Danny van Tol
 * Datum: 01-06-2011
 * Versie: 0.3
 * Bedrijf OpenSesame ICT BV.
 */

Setup Gdata library van het PHP Zend Framework
-----
- Download de (nieuwste) Zend Gdata package vanaf deze url: http://framework.zend.com/download/gdata
- Pak deze zip uit en voeg deze toe in de openims map. ik heb de naam van de hoofdfolder "zendgdata-versienr" veranderd in "zendgdata"
- Edit vervolgens de volgende regel in de php.ini file bij "Paths en directories": include_path = ".;c:\openims\openims\zendgdata\library"
- Dit voorbeeld is voor een windows installatie.. voor unix zal je een ander path op moeten geven.
- Vergeet tevens na deze aanpassingen niet Apache even te restarten!

Extra info:
 * Indien je met NetBeans IDE werkt, is het van belang dat je een project aanmaakt, anders zal de library niet geladen worden!

Google Docs oo code met openims
-----
(1) *** Als de Zend Gdata library goed is ingeladen, is het van belang dat de volgende aanpassingen gedaan worden in de code ***
- Pas in "uuse_googledocs_sync_oo_test.php" het volgende aan:
- Pas het folder id van de map die je in openims wilt gebruiken: eg. $openims_folderid = "184135102b30eade1730b2e8c2209a98";
- Pas de credentials aan van Gmail met je eigen gebruikersnaam en wachtwoord bij de variabelen: $username en $password

Extra info:
 * Deze installatie zou op elke willekeurige openims installatie moeten werken. Dit zal echter eerst getest moeten worden.

(2) *** Batch aanmaken voor autosync ***
- Om OpenIMS en Google Docs automatisch te laten synchroniseren, maak je een batch component aan, waarin je de volgende
regel toevoegd: uuse("googledocs_sync_oo_test.php"). Sla het batch bestand op met een tijdsperiode voor het uitvoeren van een synchronisatie.
- In mijn geval voer ik de synchronisatie elke minuut uit. Dit is voldoende tijd om van OpenIMS naar Google en visa versa de documenten te updaten.

Extra info:
 * Om direct de documenten te laten synchroniseren, kun je de functie uuse("googledocs_sync_oo_test.php") ook direct aanroepen in de codetester.

Google Docs procedurele code met openims
-----
(1) *** Als de Zend Gdata library goed is ingeladen, is het van belang dat de volgende aanpassingen gedaan worden in de code ***
- In de eerste functie van het bestand "uuse_googledocs_sync.php" genaamd: GoogleSync_Exec() zet je de username en passwd van het gmail account van de klant.
eg. $openims_specs['username'] = "ict.opensesame"; -> $openims_specs['username'] = "gmail_user_account";
eg. $openims_specs['password'] = "current_customer_password"; -> $openims_specs['password'] = "new_customer_password";
- In dezelfde file/ functie pas je even het folderid aan, van de folder waarin je de documenten binnen openims wilt hebben.
eg. $openims_specs['openims_folderid'] = "184135102b30eade1730b2e8c2209a98"; -> $openims_specs['openims_folderid'] = "new_folder_id";

Extra info:
 * Deze installatie zou op elke willekeurige openims installatie moeten werken. Dit zal echter eerst getest moeten worden.

(2) *** Batch aanmaken voor autosync ***
- Om OpenIMS en Google Docs automatisch te laten synchroniseren, maak je een batch component aan, waarin je de twee volgende
regels toevoegd: uuse("googledocs_sync.php"); en daaronder de functie die de sync uitvoerd: GoogleSync_Exec();
- In mijn geval voer ik de synchronisatie elke minuut uit. Dit is voldoende tijd om van openims naar Google en visa versa de documenten te updaten.

Extra info:
 * Om direct de documenten te laten synchroniseren, kun je uuse("googledocs_sync.php"); en de GoogleSync_Exec(); functies ook direct aanroepen in de codetester
```

Figuur 24: Korte handleiding met informatie hoe Google Docs met OpenIMS aan de praat te krijgen.

10 Evaluaties

Dit hoofdstuk bevat een terugblik van alle processen die uitgevoerd zijn tijdens het afstuderen. Er zal geëvalueerd worden over het product, het proces, en de beroepstaken die op een bepaald niveau behaald moesten worden. In dit hoofdstuk zal de afstudeerder tevens zijn mening geven over bepaalde aspecten.

10.1 Productevaluatie

De op te leveren software moest in stappen ontwikkeld worden om tot een goed eindproduct te komen. Het doel van een koppeling van OpenIMS met Google Docs moest ik opdelen in meerdere doelen, die zich vertaald hebben in meerdere incrementen, die gezamenlijk ingezet zou worden om een tweeweg synchronisatie functionaliteit te programmeren.

Voor het tijdig opleveren van alle software met de gevraagde functionaliteit is gebruik gemaakt van de MoS-CoW methode i.c.m. timeboxing. Hoewel er op een paar timeboxes uitloop was, had dit verder geen gevolgen voor het eindproduct. Bij andere timeboxes ben ik namelijk weer iets ingelopen waardoor ik volgens planning kon op tijd kon starten aan de synchronisatie software.

Evaluerend over alle deelproducten, en het opgeleverde eindproduct ben ik hier goed tevreden over. De functionaliteiten die de opdrachtgever en begeleider van OpenSesame ICT voor ogen hadden, zitten in het eindproduct.

10.2 Procesevaluatie

Door transparante communicatie met de opdrachtgever en begeleider van OpenSesame is het project goed verlopen. OpenSesame is een vrij platte organisatie waar mensen elkaar dingen kunnen vragen. Hoewel ze van afstudeerders een grote mate van zelfstandigheid verwachten, zijn ze bereid te luisteren als je echt met inhoudelijke vragen komt, waarbij je geen beschikking hebt tot enige vorm van informatie via internet of andere informatiebronnen.

Om de paar weken werd een evaluatiegesprek gehouden met de opdrachtgever en begeleider over de status van het project. Tijdens deze gesprekken werd bekeken of ik nog steeds op de juiste weg zat, en niet teveel afgedwaald was. Eventuele knelpunten of onduidelijkheden konden tijdens dit gesprek ook besproken worden, zodat de opdrachtgever en begeleider ook mee konden denken. Tijdens de besprekingen heb ik direct notities gemaakt, zodat ik geen informatie zou vergeten.

Organisatorisch was alles goed geregeld om goed te kunnen werken. Ik zat met twee andere stagiairs in een aparte ruimte waar we onze eigen werkplek hadden. Er was een goede internet verbinding, en er kon geprint worden. Om up-to-date te blijven, en te communiceren, kreeg ik (zoals iedere stagiair/ afstudeerder) een eigen e-mail adres van het bedrijf tot mijn beschikking. Om een back-up van mijn bestanden te behouden kreeg ik ook een eigen privéfolder met diskruimte in het DMS van OpenSesame ICT. Tevens is er een intranet beschikbaar waarop je kan inloggen voor andere benodigde informatie.

Gezien al deze punten heb ik weinig tot geen hinder ondervonden in het werkproces. Doordat OpenSesame ICT het hele jaar door afstudeerders (en ook stagiairs) in dienst hebben, weten zij hier goed mee om te gaan. Ze zijn flexibel, en geven je een mate van vrijheid, maar verwachten wel een hoge mate van zelfstandigheid hebt, en dat je je werk goed en serieus doet. Voor mij was dit een fijne manier om te werken.

10.3 Beroepstaken

Tijdens de opleiding Informatica zijn een aantal beroepstaken (ook wel competenties) behandeld. Tijdens het afstuderen heb ik aan moeten tonen aan dat ik deze beroepstaken op voldoende niveau heb uitgevoerd. Voor- dat ik begon met afstuderen, heb ik bij het invullen van de afstudeer-template aan moeten geven dat ik 2 beroepstaken op het hoogste niveau, niveau(4) moet behalen.

Bij iedere beroepstaak zal ik een korte toelichting geven over de inhoudt, en of ik naar mijn mening hier voldoende in ben geslaagd deze op niveau te behalen.

- 1.1 Selecteren, methoden, technieken en tools (niv 3)

- o Deze beroepstaak betreft het selecteren en adviseren over methoden en technieken om een systeem te ontwikkelen, beheren en te testen. Het selecteren van ontwikkeltools die een onderdeel (gaan) vormen van de ontwikkel-,beheer-, test- en productieomgeving. Het opzetten van de ontwikkel-, beheer-, test- en productieomgeving. Kunnen werken met de geselecteerde methoden binnen de gerealiseerde omgeving.

- Ben ik hier in geslaagd?

- Tijdens dit afstudeerproject is gebruikt gemaakt van de DSDM ontwikkelmethode en technieken als (MoSCoW, Timeboxing) welke onderdeel zijn van DSDM. Door de flexibiliteit van DSDM heb ik UML van RUP gebruikt om deze twee te combineren. Deze methoden en technieken zijn gebruik vanaf de ontwerpfase tot de bouw van de applicatie, om tot een goed eindproduct te komen. Naar mijn mening heb ik deze beroepstaak zeker op voldoende niveau behaald, door deze ontwikkelmethodieken en technieken op een juiste wijze toe te passen tijdens mijn afstudeerproject.

- 3.2 Ontwerpen systeemdeel (niv 4)

- o inhoud competentie

- Het beschrijven van systeemdelen (subsystemen, componenten, modules), hun onderliggende structuur en het gedrag in detail, zodanig dat bouwen van het systeemdeel mogelijk is. Ontwerpen (grafische) user interface.

- Ben ik hier in geslaagd?

- Voor het ontwerpen van de systeemdelen is gebruik gemaakt van UML als modeleringstaal. Dit is toegepast bij zowel functioneel ontwerp voor het maken van Use-case diagrammen en beschrijvingen, en bij het technisch ontwerp voor het klassendiagram en alle sequencediagrammen. Voor zowel het gehele proces als de sub-producten (incrementen) zijn functionele en technische ontwerpen opgesteld. Door gebruik te maken van de juiste technieken als UML, die ook tijdens de opleiding Informatica centraal staat wat betreft software ontwerp vind ik dat ik deze competentie zeker op voldoende niveau heb uitgevoerd.

- 3.3 Bouwen applicatie (niv 4)

- o inhoud competentie

- Verfijnen en/of transformeren van het (detail)ontwerp van de systeemdelen (subsystemen, componenten, modules) van een applicatie. Bouwen en documenteren van de systeemdelen. Systeemdelen samenstellen tot een werkende applicatie.

- Ben ik hier in geslaagd?

- Bij de ontwikkelfase zijn er systeemdelen (incrementen) ontwikkeld voor het realiseren van het eindproduct. Alle must-haves die opgesteld waren in samenspraak met de opdrachtgever c.q. externe begeleider zijn nagekomen, en zitten in het eindproduct. Bij de bouw van de applicatie zijn kleine aanpassingen gedaan met betrekking tot de ontwerpmodellen voor een goede werking. De applicatie is zowel procedureel als object-

georiënteerd geschreven waarbij een keuze gemaakt kan worden welke versie men wil gebruiken. Doordat alle functionaliteit aanwezig is, en op de manier werk zoals gewenst is, is mijn mening dat ik deze competentie zeker op voldoende niveau is behaald.

- 3.4 Initiëren en plannen van het testproces (niv 3)

o inhoud competentie

Opstellen testplan, waaronder het vaststellen van de uitgangspunten van testen, de teststrategie, de testsoorten, de testomgeving en de te gebruiken testmethoden, -ontwerptechnieken en hulpmiddelen.

- Ben ik hier in geslaagd?

Voor de testfase is een testplan opgesteld. Dit testplan heeft alle hierboven genoemde punten. Daarnaast heb ik een aantal kopjes gebruikt van de IEEE 829-structuur. Dit is de standaard voor software test documentatie. Deze competentie vind ik op voldoende niveau behaald, aangezien het testplan is opgesteld met de inhoud die daarvoor vastgesteld is.

- 3.5 Uitvoeren van en rapporteren over het testproces (niv 3)

o inhoud competentie

Opstellen testscript, waaronder het specificeren van de testgegevens, de benodigde startsituatie en de uitvoerverwachtingen per testgebied. Uitvoeren testplan en opstellen rapportage.

- Ben ik hier in geslaagd?

Het testrapport is uitgevoerd op basis van de vooraf gedefinieerde testtechnieken uit het testplan. De resultaten c.q. uitvoerverwachtingen zijn opgenomen in het testrapport. Voor het uitvoeren van de tests m.b.t. de testtechnieken is gebruik gemaakt van de testmethode, TMap. TMap is ook veelvuldig gebruikt tijdens mijn opleiding Informatica voor het opstellen van tests. Ik ben van mening, dat ook deze competentie op voldoende niveau is afgerond.

10.4 Zelfreflectie

Als ik terugblik op mijn afstudeerperiode, vind ik zelf, dat ik veel geleerd heb. Ik heb meer ervaring op het gebied van object-georiënteerd programmeren in PHP, het zelfstandig werken, het kiezen van een ontwikkelmethodiek en technieken die ik dit project zou gaan gebruiken, het documenteren en daarmee verantwoorden van keuzes en beslissingen, en het werken in een professionele omgeving bij OpenSesame ICT.

Vooraf aan het project had ik mezelf voorgenomen zo snel mogelijk te werken en alles keurig stipt in een logboek bij te houden. Dit ging de eerst paar weken redelijk goed. Toen de onderzoeksfase begon, kwam er zoveel informatie op me af dat het heel lastig was om dit nog steeds goed bij te houden. De weken daarna heb ik wat minder uitgebreid mijn logboek bijgehouden door de hoeveel informatie. Ik wilde voor mijzelf dan ook selectief zijn, en echt alleen de informatie die nuttig en relevant was, gebruiken voor het afstudeerverslag. Ik heb wederom gemerkt dat het discipline vereist om een logboek goed bij te houden.

Om goed bij te houden wat ik gedaan heb, en welke werkzaamheden nog verricht moesten worden wat betreft het product en de documentatie, heb ik checklists op papier bijgehouden. Met markerings stiften afgestreept welke taken verricht waren, en welke nog gedaan moesten worden. Dit vond ik een goede manier van werken om zo controle over het project te behouden.

Tijdens de ontwikkelfase heb ik het programmeren af en toe afgewisseld met verslaglegging. Hoewel programmeren een leuke bezigheid is, en er voldoening is, als iets goed werkt, is het ook moeilijk om hele dagen achtereen de concentratie erbij te houden. Bij problemen of lastige functies die geprogrammeerd moesten worden was het soms hoofdbreken. Als het dan laat in de middag was, was het lastig om de concentratie erbij te houden. In dit geval ben ik vaak de volgende dag in de ochtend fris begonnen, waardoor problemen vaak veel sneller gevonden werden.

Gezien de hoeveelheid werk van het op te leveren product en alle verslaglegging vind ik dat ik niet getreuzeld heb. Ik heb gedurende het hele afstudeertraject hard doorgewerkt en weinig tijd verloren door andere bezigheden. Hiermee wilde ik voor het bedrijf OpenSesame ICT en mijzelf een goed product neerzetten, waar ik nu met een tevreden gevoel op kan terugkijken.

10.5 Conclusie

Een geslaagd afstudeerproject waarbij veel ervaring is opgedaan wat betreft programmeren, documenteren, en zelfstandig werken in een professionele omgeving. Ik heb het bij OpenSesame ICT erg naar mijn zin gehad. Ik heb een leuke opdracht uit mogen voeren, wat goed aansluit bij de opleiding Informatica.

Ik wil dan ook graag de externe begeleider, Dhr. Eric Diks en opdrachtgever, Dhr. Nico de Vries van OpenSesame ICT bedanken voor het vertrouwen en de hulp tijdens mijn afstuderen.

Ook de begeleiders van school, Dhr. Okan Zor en Mevr. Gerda In 't Veld voor de nuttige feedback voor verbeteringen aan het verslag en de tijd die aan mij besteed is.

11 Woordenlijst

Nr.	Afkorting	Betekenis
1	DSDM	Dynamic Systems Development Method
2	RAD	Rapid Application Development
3	JSON	JavaScript Object Notation
4	ReST	Representational state transfer
5	AtomPub	Atom Publishing Protocol
6	CMS	Content Management Systeem
7	DMS	Document Management Systeem
8	ECM	Enterprise Content Management
9	PS	Portal System
10	SLA	Service Level Agreement
11	XML	Extensible Markup Language
12	HTTP	Hypertext Transfer Protocol
13	PHP	Hypertext Preprocessor
14	OpenIMS	Open Informatie Management Systeem
15	SugarCRM	Sugar Cliënt Relatie Management
16	API	Application Programming Interface
17	Captcha	Completely automated public Turing-test to tell computers and humans apart
18	MVC	Model View Controller
19	LDAP	Lightweight Directory Access Protocol
20	Ajax	Asynchronous Javascript And XML
21	URI	Uniform Resource Identifier
22	URL	Uniform Resource Locator
23	URN	Uniform Resource Name
24	IEEE	Institute of Electrical and Electronic Engineers

12 Referentielijst

Nr.	Standaard	Onderwerp	Website / boek / handleiding
1	Google	API home	http://code.Google.com/intl/nl/apis/documents/
2	Google	PHP API	http://code.Google.com/intl/nl/apis/documents/Docs/1.0/developers_guide_php.html
3	Google	Protocol	http://code.Google.com/intl/nl/apis/documents/Docs/3.0/developers_guide_protocol.html
4	Google	Libs Downloads	http://code.Google.com/intl/nl/apis/gdata/Docs/client-libraries.html
5	Google	API forum	http://groups.Google.com/group/Google-documents-list-api
6	Zend	Gdata downloads	http://framework.zend.com/manual/en/zend.gdata.html
7	Zend	Gdata libs info	http://framework.zend.com/apidoc/core/
8	Zend	Reference guide	http://framework.zend.com/manual/en/zend.gdata.youtube.html
9	Zend	Zend Gdata	http://framework.zend.com/manual/en/zend.gdata.Docs.html
10	OpenIMS	Handleiding	OpenSesame ICT BV, Technisch en Functioneel Beheer handleiding, OpenIMS 4.2, Nieuwegein, 2007
11	OpenIMS	Intranet	http://intranet.osict.com/
12	OpenIMS	Opslag Docs	http://openims/dms
13	OpenIMS	Info algemeen	http://www.openims.com/
14	DSDM	Wiki	http://en.wikipedia.org/wiki/Dynamic_Systems_Development_Method
15	DSDM	Boek	Pearson Education Benelux., DSDM Consortium., DSDM – De methode in de praktijk 2e editie, Jennifer Stapleton, United Kingdom, 2003
16	DSDM	Home	http://www.dsdm.org/
17	ReST	Wiki	http://en.wikipedia.org/wiki/Representational_state_transfer
18	ReST	Introduction	http://www.oio.de/public/xml/rest-webservices.htm
19	ReST	Tut. ReST/PHP	http://rest.elkstein.org/2008/02/using-rest-in-php.html
20	Atom	Wiki	http://nl.wikipedia.org/wiki/Atom_%28bestandsformaat%29
21	Atom	Specificaties	http://www.atomenabled.org/developers/protocol/atom-protocol-spec.php
22	Atom	Protocol	http://bitworking.org/projects/atom/rfc5023.html
23	Atom	AtomPub	http://www.slideshare.net/benramsey/web-service-design-with-atompub
24	Atom	ASF	http://tools.ietf.org/html/rfc4287#section-1
25	JSON	Home	http://www.json.org/
26	JSON	Wiki	http://nl.wikipedia.org/wiki/JSON
27	JSON	PHP/ JSON man	http://php.net/manual/en/book.json.php
28	JSON	JSON+REST	http://blog.feedly.com/2009/03/03/jsonrest-vs-xmlsoap/
29	UMLet	UML	http://www.umlet.com/
30	UML	Sequence	http://en.wikipedia.org/wiki/Sequence_diagram
31	PHP	Home	http://www.php.net/
32	Google	Apps accounts	http://code.Google.com/intl/nl-NL/apis/apps/start.html

13 Bijlage

1 Bijlage van de onderzoeksfase

1.1 Google Protocol bindings

Hoewel het Google Docs protocol niet rechtstreeks gebruikt is bij het ontwikkelen van de software, maar gebruik is gemaakt van de PHP API, heb ik wel onderzoek gedaan naar dit protocol. Het is belangrijk en kenmerken namelijk de structuur van Google Documenten. Hoewel ze niet rechtstreeks gebruikt zijn, hebben ze op de achtergrond een rol gespeeld tijdens dit project. In de volgende koppen zullen de Google Docs bindings ReST, Atom, XML en JSON aan de orde komen.

ReST

REST gebaseerde diensten (Representational State Transfer) bieden de kans om op eenvoudige wijze de real-time-uitvoering van connectiviteit te realiseren. Hoewel het acroniem (de afkorting) misschien lastig is, gebruikt REST het World Wide Web dat iedereen kent om toegang te krijgen tot informatie. In plaats van de data in enveloppen (zoals SOAP-XML) te stoppen, en je heel veel moeite moet doen om gegevens te bereiken, doet REST dat met een simpele URL. REST kan uit de voeten met verschillende formaten zoals XML, HTML, net zoals het web dat tegenwoordig kan.

Het feit dat REST gebruik maakt van de eigenschappen die het World Wide Web succesvol hebben gemaakt en de wetenschap dat die technologie beschikbaar is, makkelijk en open voor iedereen, is het zeker een model om in overweging te nemen wanneer het aankomt op real-time connectiviteit.

In plaats van XML (SOAP) gebruikt Google vandaag de dag JSON in combinatie met REST. Hier zijn een aantal redenen voor te noemen:

1. JSON is simpelweg eenvoudiger te begrijpen en te lezen dan XML en XML-schema's. Bij beide onderwerpen zal een voorbeeld volgen.
2. Een tweede punt is dat de browsers van nu een grote hoeveelheid JSON veel efficiënter kunnen consumeren dan een grote hoeveelheid XML. De kloof wordt tevens steeds groter om te consumeren, doordat de nieuwste browsers versies nu eigen, veilige ondersteuning voor het coderen en decoderen van JSON aanbieden. [\[bron\]](#)

Atom(ASF)

Atom (Syndicate Format) is een soort webfeed, net als RSS (Really Simple Syndication). Webfeeds zijn voor gebruikers erg handig omdat zij ervoor zorgen dat content van websites automatisch aan hen wordt gepresenteerd als deze wordt aangepast of toegevoegd. De gebruiker hoeft dus niet steeds zelf de website te bezoeken. Een feed heeft een datastructuur van een XML of JSON (zoals bij Google in beide gevallen) formaat. Nieuw aangemaakte Google documenten bijvoorbeeld worden in een aantal regels samengevat en vergezeld van een titel en deeplink waar de gebruiker vervolgens meer informatie kan lezen. [\[bron\]](#)

Atom is een open standaard die zowel een XML-gebaseerde websyndicatie voor onder meer weblog, nieuwssite en webmail levert als een HTTP-gebaseerd protocol voor bewerken op afstand van Atom-based weblogs. De ontwikkeling van Atom werd ingegeven door onvolkomenheden van het RRS 2.0 formaat en daaraan verwante XML-RPC APIs. Atom is nu een officieel door de Internet Engineering Task Format voorgestelde standaard (zie: Figuur 25).

```
XML Data
<?xml version='1.0' encoding='utf-8'?>
<feed xmlns='http://www.w3.org/2005/Atom'
      xmlns:openSearch='http://a9.com/-/spec/opensearch/1.1/'
      xmlns:gAcl='http://schemas.Google.com/acl/2007'
      xmlns:gd='http://schemas.Google.com/g/2005'
      gd:etag='W/"CUUNSXYyfCp7ImA9WxRVGUo."'>

  <id>https://Docs.Google.com/feeds/acl/private/full/document%3Adocument_id</id>
    <updated>2008-09-03T22:03:04.733Z</updated>
    <category scheme='http://schemas.Google.com/g/2005#kind'
term='http://schemas.Google.com/acl/2007#accessRule' />
    <title type='text'>Document Permissions</title>
    <link rel='http://schemas.Google.com/g/2005#feed'
type='application/atom+xml'

href='https://Docs.Google.com/feeds/acl/private/full/document%3Adocument_id' />
    <link rel='http://schemas.Google.com/g/2005#post'
type='application/atom+xml'

href='https://Docs.Google.com/feeds/acl/private/full/document%3Adocument_id' />
    <link rel='self' type='application/atom+xml'

href='https://Docs.Google.com/feeds/acl/private/full/document%3Adocument_id' />
    <openSearch:totalResults>3</openSearch:totalResults>
    <openSearch:startIndex>1</openSearch:startIndex>
</feed>
```

Figuur 25: Een voorbeeld van een Google Atom feed.

XML

XML is een standaard van het World Wide Web Consortium voor de syntaxis van formele markuptalen waarmee men gestructureerde gegevens kan weergeven in de vorm van platte tekst. Deze representatie is zowel machineleesbaar als leesbaar voor de mens. Het XML-formaat wordt gebruikt om gegevens op te slaan (zoals in het OpenDocument-formaat) en om gegevens over het internet te versturen (zoals in AJAX).

XML-talen gebruiken zogenaamde elementen en attributen om gegevens te structureren. De XML-specificatie definieert de syntaxis van elementen, attributen en de andere structuren die in XML-bestanden kunnen voorkomen. De XML-specificatie legt echter geen namen vast voor deze elementen en attributen, precies omdat deze keuze afhangt van het doel van het XML-bestand (zie: Figuur 26). [\[bron\]](#)

XML Data

```
<?xml version="1.0" encoding="UTF-8" ?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:openSearch="http://a9.com/-/spec/opensearchrss/1.0/"
      xmlns:gd="http://schemas.Google.com/g/2005"
      xmlns:gCal="http://schemas.Google.com/gCal/2005">
  <id>...</id>
  <updated>2006-11-12T21:25:30.000Z</updated>
  <title type="text">Google Developer Events</title>
  <subtitle type="text">The calendar contains information about up-
coming
  developer conferences at which Google will be speaking, along
with
  other developer-related events.</subtitle>
  <link rel="http://schemas.Google.com/g/2005#feed"
        type="application/atom+xml" href="..." />
  <link rel="self" type="application/atom+xml" href="..." />
  <author>
    <name>Google Developer Calendar</name>
    <email>developer-calendar@Google.com</email>
  </author>
```

Figuur 26: Voorbeeld XML syntaxis.

JSON

JSON staat voor JavaScript Object Notation en is een deelverzameling van de programmeertaal JavaScript. Het wordt gebruikt voor het uitwisselen van datastructuren, met name in webapplicaties die asynchroon gegevens ophalen van de webserver zoals AJAX.

De eenvoud van JSON heeft geleid tot een grote populariteit ervan, met name als een alternatief voor XML. Omdat gegevens worden uitgewisseld in de vorm van JavaScript-expressies, kunnen de gegevens worden ingelezen in een JavaScript-applicatie door simpelweg de JSON-expressie te evalueren; er is dus geen aparte parser voor nodig. Voor een groot aantal andere programmeertalen zijn functiebibliotheken beschikbaar voor het lezen en schrijven van JSON-expressies (zie: Figuur 27). [bron]

JSON Data

```
{
  "version": "1.0",
  "encoding": "UTF-8",
  "feed": {
    "xmlns": "http://www.w3.org/2005/Atom",
    "xmlns$openSearch": "http://a9.com/-/spec/opensearchrss/1.0/",
    "xmlns$gd": "http://schemas.Google.com/g/2005",
    "xmlns$gCal": "http://schemas.Google.com/gCal/2005",
    "id": {"$t": "..."},
    "updated": {"$t": "2006-11-12T21:25:30.000Z"},
    "title": {
      "type": "text",
      "$t": "Google Developer Events"
    },
    "subtitle": {
      "type": "text",
      "$t": "The calendar contains information about upcoming de-
developer
conferences at which Google will be speaking, along with
other
developer-related events."
    }
  }
}
```

Figuur 27: voorbeeld JSON syntax.

1.2 Google Apps

Google Apps is het gebundelde software pakket van Google met de belangrijkste Office toepassingen zoals Gmail, Docs, en spreadsheets. Het pakket is in 2006 geïnstalleerd en wordt gezien als een grote concurrent voor Microsoft Office dat sinds het midden van de jaren 90 dominantie had op de Office software markt.

Google Apps is een pakket Google-toepassingen welke essentiële services samengebracht hebben voor bedrijven. Het is een gehoste service waarmee bedrijven, scholen en instellingen verschillende Google-producten kunnen gebruiken, waaronder Gmail, Google Agenda, Google Documenten, en Google Talk via een uniek domeinnaam (bijvoorbeeld www.uwbedrijf.nl). [bron]

Bij het registreren van je eigen domeinnaam bij Google Apps krijgt iedereen binnen het bedrijf o.a. de volgende services aangeboden:

- Een persoonlijk e-mailadres (gebruiker@example.com)
- Hulpprogramma's voor tekstverwerking
- Spreadsheets en presentaties
- Een gedeeld agendasysteem

- Hulpprogramma's voor het maken van webpagina's en sites voor het bedrijf
- Toegang tot een flexibel intranetsysteem

Wat is het verschil tussen het Google account en het Google Apps-account ?

Hoewel je met beide accounts toegang kunt krijgen tot een aantal van dezelfde Google-producten, zijn dit toch verschillende accounts. Een Google-account is een centraal aanmeldingssysteem waarmee je toegang hebt tot verschillende (gratis) Google-producten voor consumenten, zoals Gmail, Google Discussiegroepen, Google Shopping List, Picasa, Webgeschiedenis, iGoogle, en Google Checkout, die worden beheerd door Google. Met Google Apps heeft men toegang tot producten die mogelijk worden gemaakt door Google maar worden beheerd door uw eigen organisatie.

Wat is het verschil tussen Google Apps Standard Edition, Premier Edition, en Education Edition ?

Google heeft drie verschillende Apps versies: Google Apps Standard Edition, Google Apps Premier Edition en Google Apps education edition. De Premier edition kost US Dollar 50 p/j en biedt 10gb schrijfruimte, 2gb telefonische support en geen verplichte advertenties. De education edition is gratis voor erkende opleidingen maar biedt geen uptime garanties.

Figuur: 28 toont in een diagram met een overzicht van de beschikbare en bruikbare services van de Google Apps accounts:

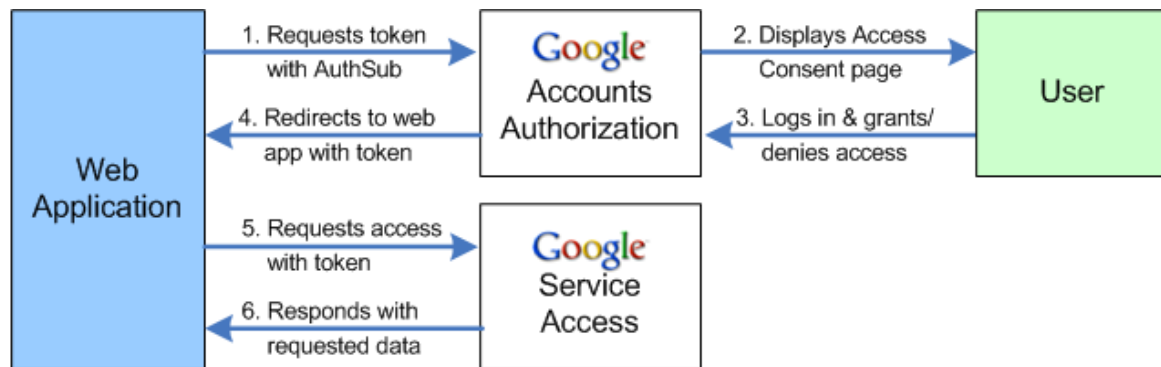
	Google Apps Google Apps	Google Apps Google Apps for Business	Google Apps Google Apps for Education*
Provisioning API		✓	✓
Reporting API		✓	✓
Email Migration API		✓	✓
Email Settings API		✓	✓
Single Sign-On service		✓	✓
Federated Login Service	✓	✓†	✓†
Email gateway interface		✓	✓
Admin Settings API		✓	✓
Shared Contacts Data API		✓	✓
User Profiles Data API	✓	✓	✓
Contacts Data API	✓	✓	✓
Calendar Data API	✓	✓	✓
Calendar Resources API		✓	✓
Documents List Data API	✓	✓	✓
Spreadsheets Data API	✓	✓	✓
2-legged OAuth Google Data access		✓	✓
Gmail Atom feed	✓	✓	✓

Figuur 28: De beschikbare opties van drie verschillende Google Apps accounts.

1.3 Andere authenticatie methoden die bekeken zijn

AuthSub autorisatie proces:

Authenticatie met AuthSub is een opeenvolging van interacties tussen de volgende drie entiteiten: De webapplicatie, Google services, en de gebruiker(s). Om hier een visueel beeld bij te vormen is in Figuur 29 een diagram weergegeven:



Figuur 29: AuthSub authenticatie proces.

De stappen die uitgevoerd worden zijn als volgt gedefinieerd:

1. Als de web applicatie toegang moet hebben met de gebruikers Google service, doet AuthSub een oproep aan het Google autorisatie proxy-service.
2. De autorisatie service reageert met inachtneming van een "Access-Request-pagina". Deze Google pagina vraagt de gebruiker toegang te verlenen tot de Google service. Deze kan overigens ook geweigerd worden. Als de gebruiker nog niet ingelogd is op zijn of haar Google account, zal de pagina eerst vermelden aan de gebruikers om in te loggen alvorens de toestemming al dan niet te verlenen.
3. De Google gebruiker bepaald zelf of hij de web applicatie toegang zal verlenen of deze zal weigeren. In het geval de gebruiker zal weigeren, wordt men doorverwezen naar een Google pagina in plaats van terug te keren naar de web applicatie zelf.
4. Als de gebruiker toegang verleent, zorgt de Google service ervoor dat de gebruiker terug geleid wordt naar de web applicatie. De redirect bevat een token die goed is voor eenmalig gebruik. Dit token kan echter ingewisseld word door een zogeheten "long-lived" token.
5. De web applicatie verstuurd een verzoek naar de Google service met gebruikmaking van een autorisatie token om zich voor te doen als een gebruiker.
6. Als de Google service het token herkent, zal deze de opgevraagde gegevens leveren. [\[bron\]](#)

OAuth autorisatie proces:

OAuth is een open standaard verificatieprotocol waarmee beheerders toegang kunnen krijgen tot basisgegevens van gebruikers zonder de gebruikersaccounts te hoeven openen. Gebruikers hoeven hun gebruikersnaam en wachtwoord niet af te geven. OAuth werkt als de "mastersleutel" die via een gebruikerstoken beperkte toegang krijgt tot gegevens. Indien nodig kunnen gebruikers de mastersleutel intrekken. [\[bron\]](#)

Met Google Apps Professional en Education hebben beheerders toegang tot een tweezijdige OAuth. Hiermee kunnen beheerders van het domein zich bij het gebruiken van elke gegevens-API van Google in feite voordoen als een gebruiker, zonder dat ze toegang hebben tot de aanmeldingsgegevens van de gebruiker. Als tweezijdige OAuth is ingeschakeld, kan elke domeinbeheerder bijvoorbeeld de Google Agenda-feed of de Google Documenten List-feed weergeven en/of aanpassen voor elke gebruiker van het domein.

In figuur 30 is een model weergegeven van OAuth:



Figuur 30: OAuth autorisatie proces.

1. Met OAuth zijn er drie typen tokens: Request token, authorized token en het access token. Het doel is het bekomen van een access token.
2. Eerst wordt er een verzoek gedaan naar Google om een RequestToken te verkrijgen.
3. Dit RequestToken kan ingehuild worden door een AuthorizedRequestToken als de gebruiker de applicatie toestaat om andere gebruikers toe te laten tot zijn Google account. Deze manier lijkt op de manier zoals Facebook toegang geeft aan applicaties.
4. Vervolgens moet het AuthorizedRequestToken ingehuild worden voor een AccessToken. Dit AccessToken is in principe "oneindig lang" geldig.
5. Elk volgend verzoek naar Google moet "gesigned" worden. (een hash van alle header variabelen).

Verschil van AuthSub ten opzichte van OAuth

- Met AuthSub heb je een geen customer key en secret nodig, met OAuth is dit wel het geval.
- AuthSub is gemakkelijker om mee te werken.
- Bij AuthSub kan de sessie verlopen.
- OAuth krijgt een access en secret token en connect elke keer totdat de gebruiker de pagina aanroept.
- AuthSub is alleen beschikbaar binnen de Google API, OAuth is ook verkrijgbaar in andere API's zoals Google zelf, maar ook Twitter bijv.

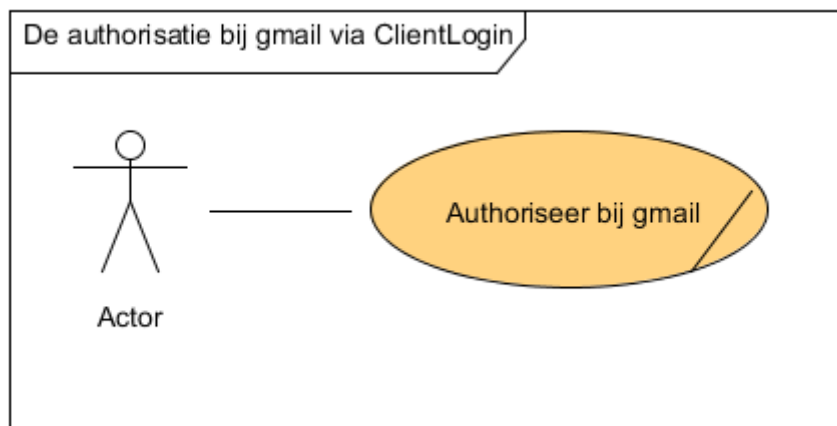
2 Bijlage van de ontwerpfase

Tijdens de ontwerpfase zijn er afzonderlijke functies geprogrammeerd die benodigd waren voor het algehele synchronisatie proces. Van elke functie is een use-case, use-case beschrijving en sequence digram gemaakt. Om verwarring te voorkomen zijn van deze functies de use-cases, use-case beschrijvingen, en sequence diagrammen apart opgenomen in de bijlage.

Use-case diagrammen + beschrijvingen

[1] Het autoriseren bij Gmail

Use-case:

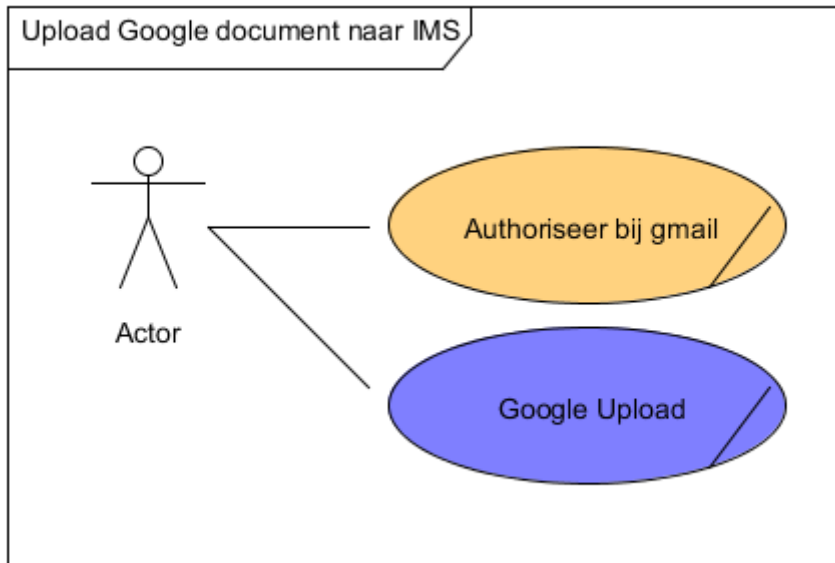


Use-case beschrijving:

ID	1
Use-case	Authentiseren bij Gmail
Actor	Tester / Ontwikkelaar
Pre-conditie	De actor wil zich autoriseren.
- 80% scenario	De actor is ingelogd bij Google.
Werkwijze	[1] De actor authenticiseert zich met zijn of haar gebruikersnaam en wachtwoord, om bij zijn Google documenten te komen.
- 20% scenario	De actor voert een onjuiste gebruikersnaam of wachtwoord in.
Post conditie	Met het juist invoeren van het Gmail gebruikersnaam en wachtwoord is de actor ingelogd, en zijn de documenten bereikbaar.

[2] Het uploaden van een document

Use-case:

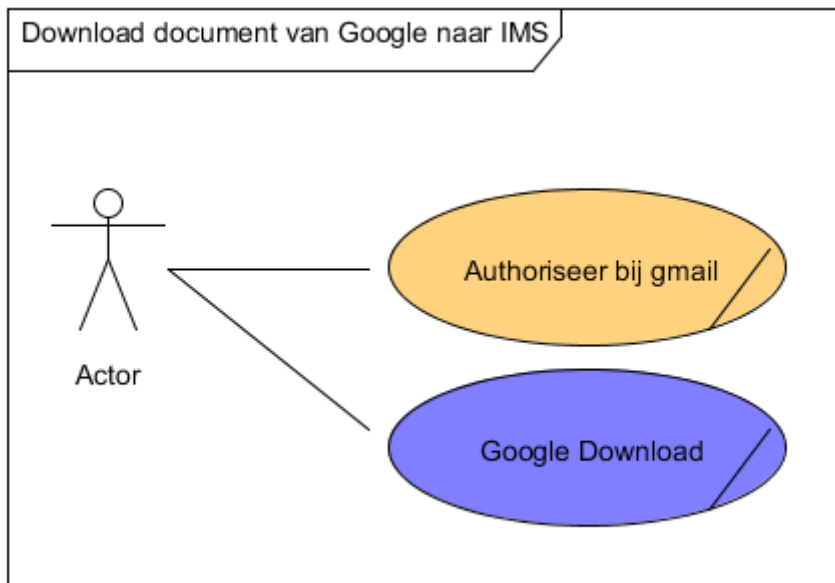


Use-case beschrijving:

ID	2
Use-case	Google Upload
Actor	Tester / Ontwikkelaar
Pre-conditie	De actor wil een document uploaden.
- 80% scenario	Documenten worden geüpload vanuit Google Docs naar OpenIMS
Werkwijze	[1] De actor upload een document vanuit OpenIMS naar Google Docs.
- 20% scenario	Een document met dezelfde bestandsnaam reeds aanwezig in OpenIMS kan als apart document geüpload worden. Google Docs ondersteund evenals OpenIMS, meerdere bestanden met dezelfde bestandsnaam. Deze documenten met dezelfde bestandsnaam zullen niet overschreven worden.
Post conditie	Het document vanuit Google zal geüpload zijn naar OpenIMS.

[3] Het downloaden van een document

Use-case:



Use-case beschrijving:

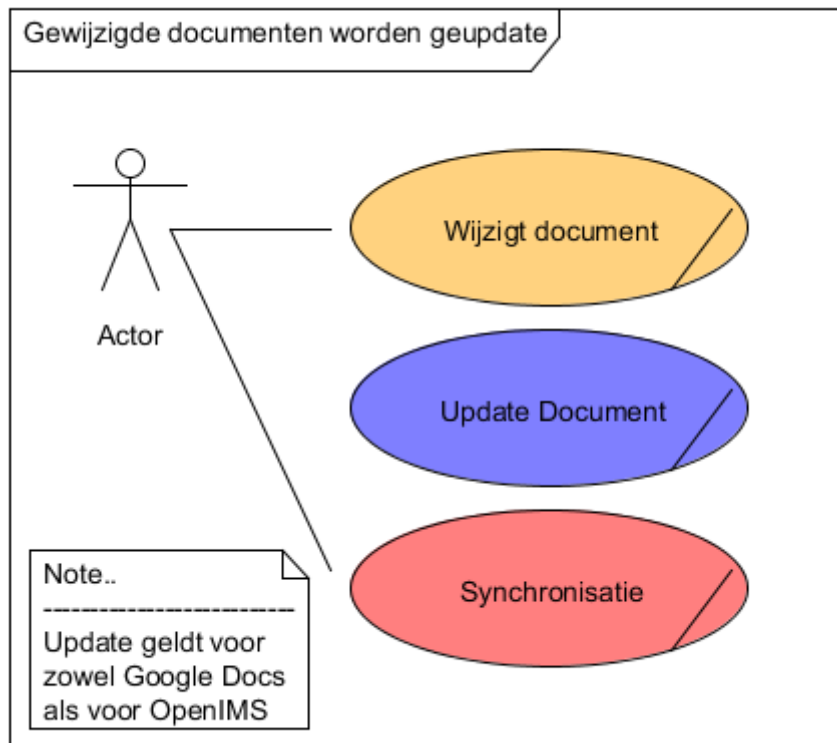
Use-case beschrijving:

ID	3
Use-case	Google Download
Actor	Tester / Ontwikkelaar
Pre-conditie	De actor wil het document van OpenIMS naar Google Docs hebben.
- 80% scenario	Het document wordt gedownload van OpenIMS en vervolgens op de Google server geplaatst.
Werkwijze	[1] De actor download het OpenIMS document, welke vervolgens in Google Docs geplaatst zal worden.
- 20% scenario	Een document met dezelfde bestandsnaam reeds aanwezig in Google Docs kan als apart document gedownload worden. Google Docs ondersteund evenals OpenIMS, meerdere bestanden met dezelfde bestandsnaam. Deze documenten met dezelfde bestandsnaam zullen niet overschreven worden.
Post conditie	Het document is nu aanwezig in Google Docs.

Download bestanden van OpenIMS naar Google docs

[4] Het updaten van een document

Use-case:

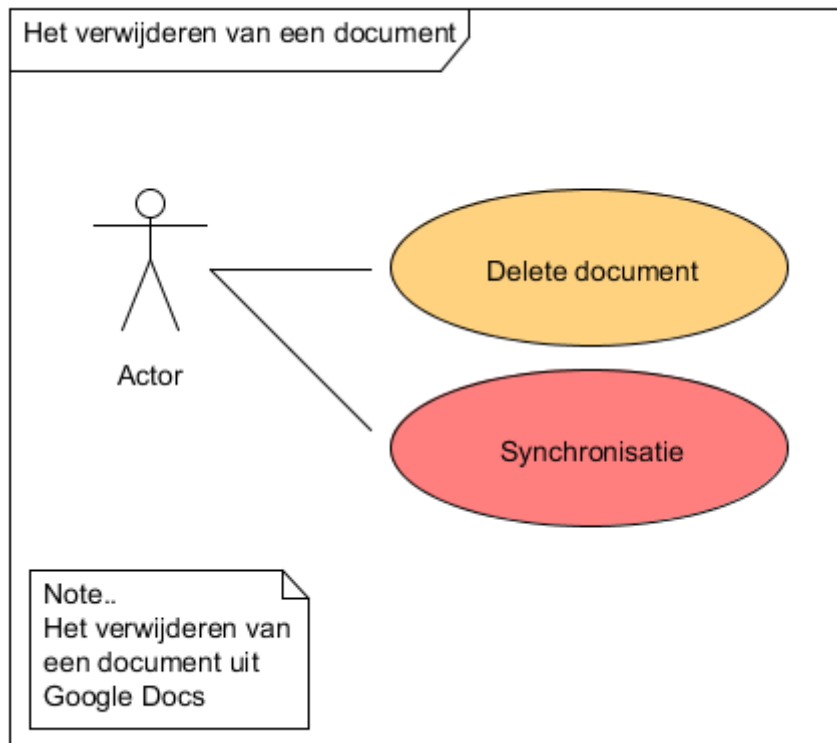


Use-case beschrijving:

ID	4
Use-case	Update document (Google / OpenIMS)
Actor	Tester / Ontwikkelaar
Pre-conditie	De actor wil een check of de update functie naar behoren werkt.
- 80% scenario	Het gewijzigde document zal geüpdatet zijn na het synchroniseren.
Werkwijze	[1] De actor wijzigt een document, en voert vervolgens de synchronisatie handmatig uit.
- 20% scenario	Een Update van document(en) door middel van een synchronisatie kunnen zowel handmatig als via batch gedaan worden. In beide gevallen krijg je hetzelfde resultaat. Het voordeel van een handmatige synchronisatie is, dat de document direct geüpdatet worden. Met een batch wordt dit elke 5 minuten gedaan. *
Post conditie	Na het wijzigen en synchroniseren zullen de document(en) in beide systemen geüpdatet zijn, en daarmee inherent aan elkaar zijn.

[5] Het verwijderen van een document uit Google Docs

Use-case:

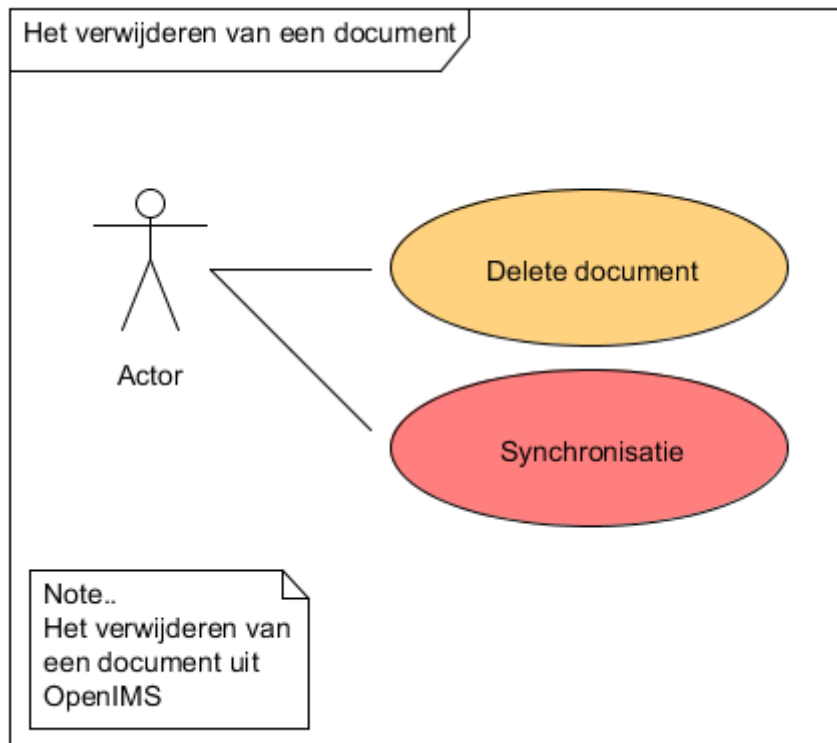


Use-case beschrijving:

ID	5
Use-case	Delete document
Actor	Tester / Ontwikkelaar
Pre-conditie	De actor doet een check of de delete functie naar behoren werkt.
- 80% scenario	Het reeds aanwezige document zal uit Google Docs verwijderd zijn.
Werkwijze	[1] De actor verwijderd het document uit Google Docs.
- 20% scenario	Het document is verwijderd uit Google, maar nog reeds aanwezig in OpenIMS. Indien bij het uitvoeren van de synchronisatie de delete functie van OpenIMS niet aanwezig is, of niet goed werkt, dan zal het document weer naar Google teruggezet worden. Dit wordt gedaan om de lijst met documenten inherent aan elkaar te laten zijn. *
Post conditie	Na het verwijderen is het document niet meer in Google aanwezig. Na het synchroniseren is het document eveneens verwijderd uit OpenIMS.

[6] Het verwijderen van een document uit OpenIMS

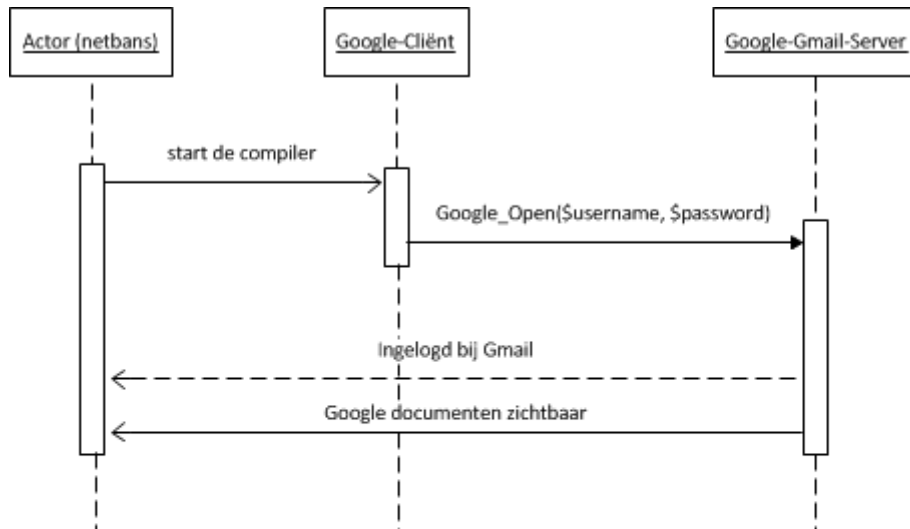
Use-case:



Use-case beschrijving:

ID	6
Use-case	Delete document
Actor	Tester / Ontwikkelaar
Pre-conditie	De actor wil testen of het document ook daadwerkelijk uit OpenIMS verwijderd word.
- 80% scenario	Het reeds aanwezige document zal uit OpenIMS verwijderd zijn.
Werkwijze	[1] De actor verwijderd het document uit OpenIMS.
- 20% scenario	Het document is verwijderd uit OpenIMS, maar nog reeds aanwezig in Google. Indien bij het uitvoeren van de synchronisatie de delete functie van Google Docs niet aanwezig is, of niet goed werkt, dan zal het document weer naar OpenIMS teruggezet worden. Dit wordt gedaan om de lijst met documenten inherent aan elkaar te laten zijn. *
Post conditie	Na het verwijderen is het document niet meer in OpenIMS aanwezig. Na het synchroniseren is het document eveneens verwijderd uit Google Docs.

[1] Sequence: Autorisatie Googlemail

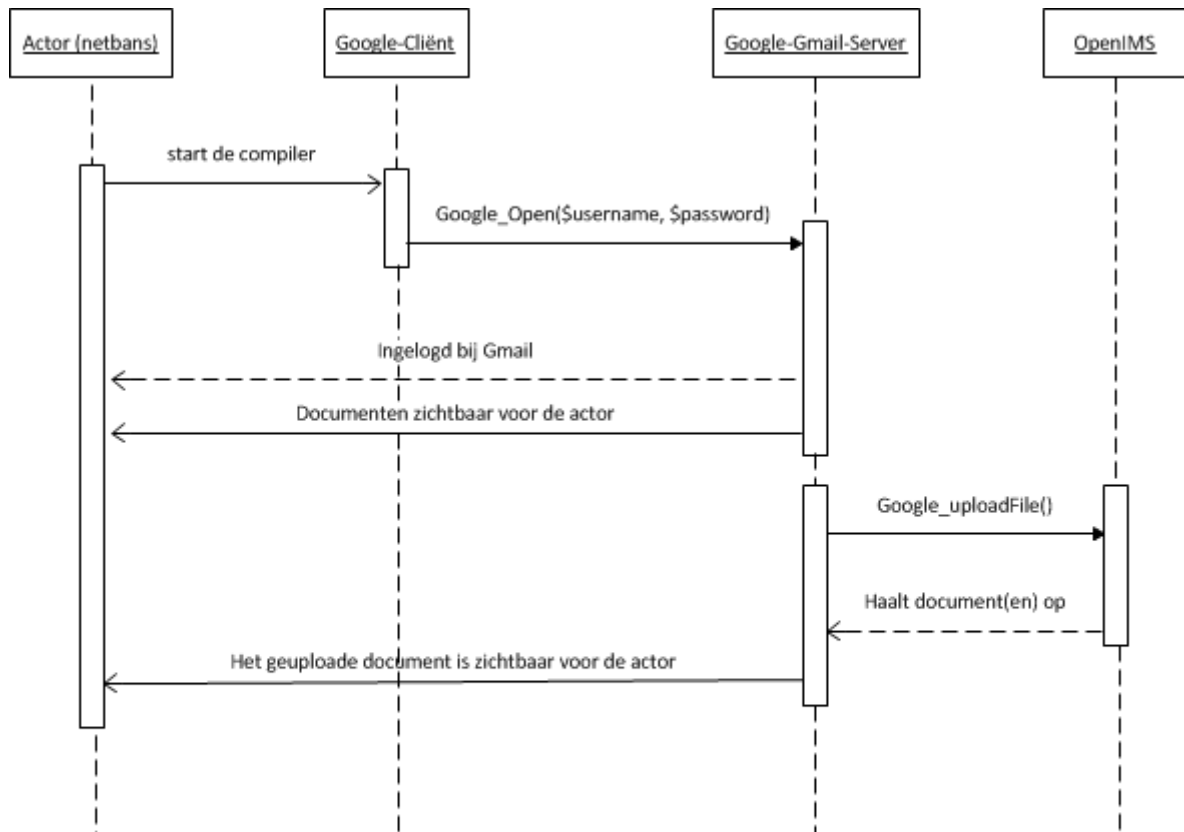


De functie hieronder aangegeven wordt aangeroepen voor autorisatie bij Google

```

function Google_Open($username, $password) {
return Zend_Gdata_ClientLogin::getHttpClient($username, $password, "writely");
}
  
```

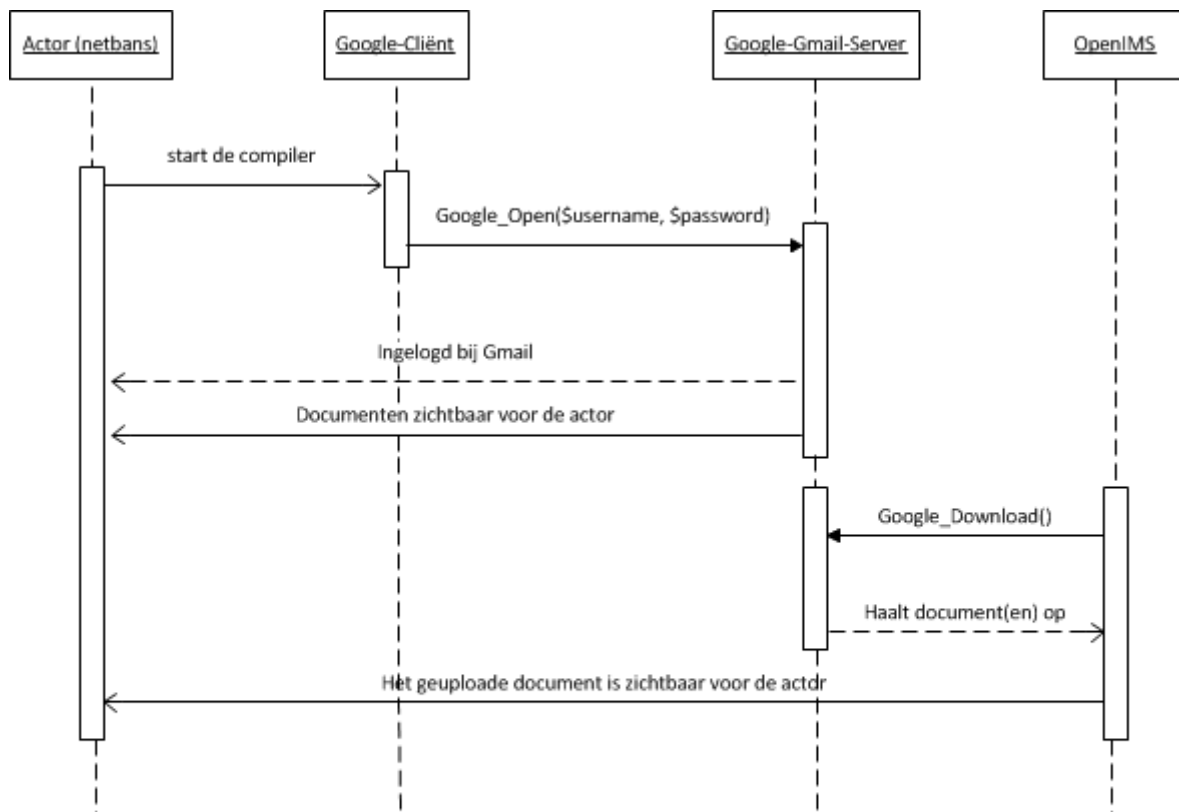
[2] Sequence: Uploaden Google document naar OpenIMS



De volgende functie wordt aangeroepen voor het uploaden van een document:

```
function Google_UploadFile($google, $openims_filePath, $google_fileName)
```

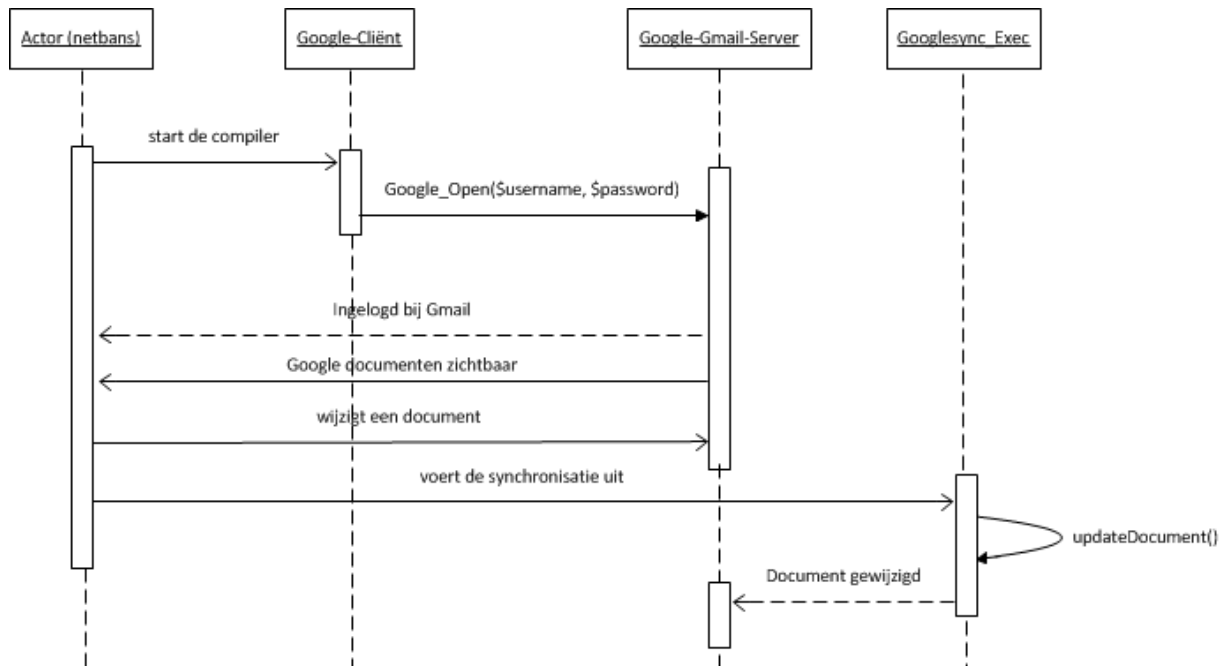
[3] Sequence: Het downloaden van Google documenten naar OpenIMS



De volgende functie wordt aangeroepen voor het downloaden van een document:

```
function Google_Download($google, $google_documentId, $google_fileName)
```

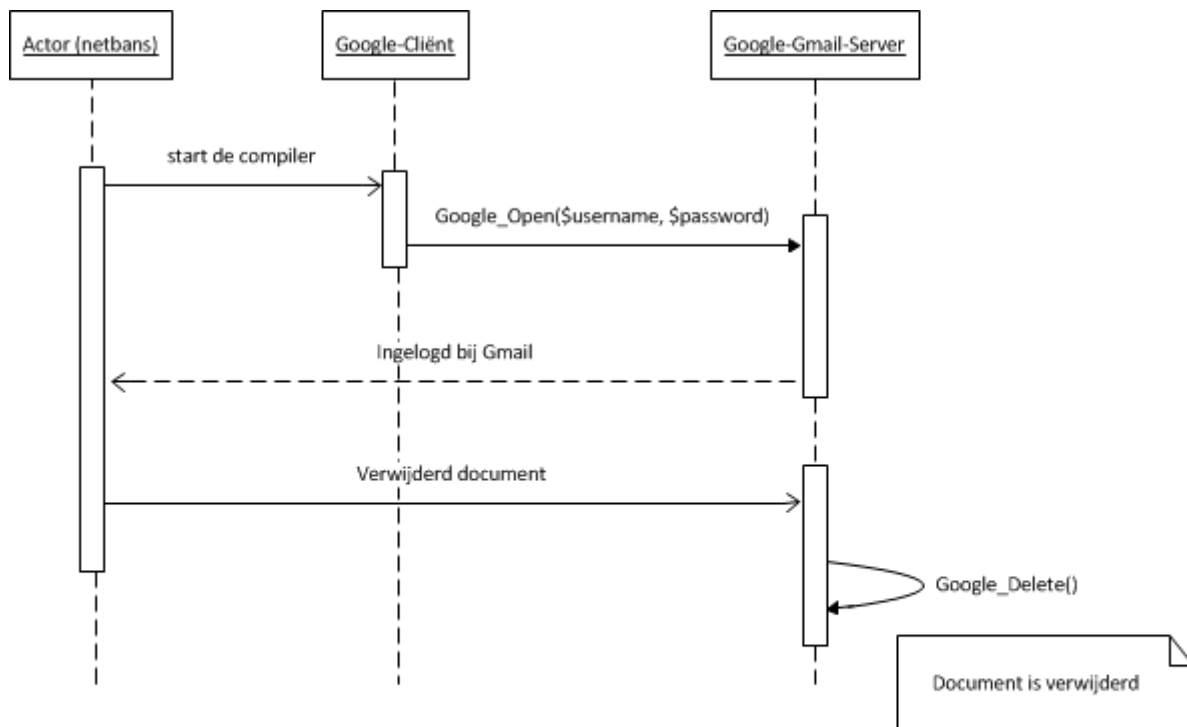
[4] Sequence: Het updaten van document(en)



De functie hieronder aangegeven wordt aangeroepen voor het updaten van een document na het wijzigen van document(en).

```
function Google_UpdateFile($google, $openims_filePath, $google_fileName, $google_documentId)
```

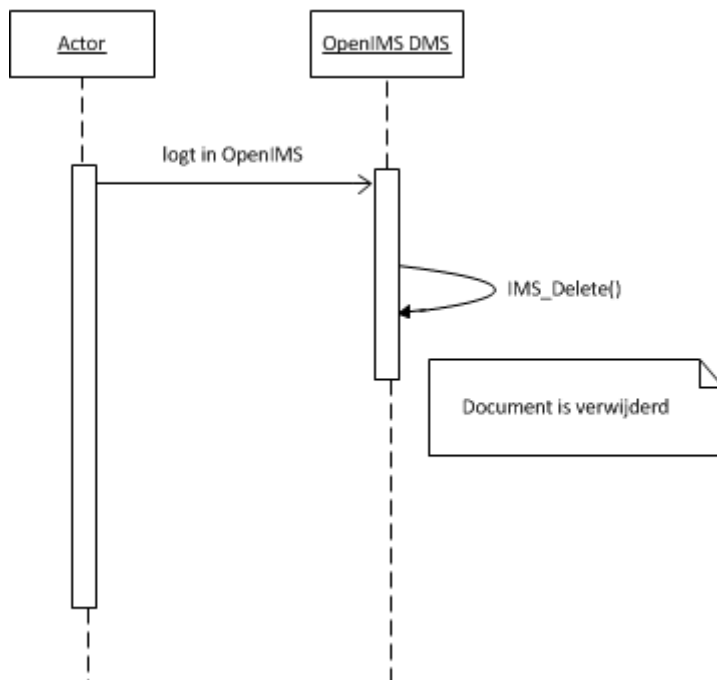
[5] Sequence: Het verwijderen van een Google document



De functie hieronder aangegeven wordt aangeroepen voor het verwijderen van een Google document:

```
function Google_Delete($google, $google_documentId)
```

[6] Sequence: Het verwijderen van een OpenIMS document



Plan van Aanpak

“Het realiseren van een koppeling van Google Docs met OpenIMS”



Datum : 15-03-2011
Plaats : Nieuwegein (Utrecht)

Auteur : Danny van Tol
Studentnr : 20057842

PvA versie : Final

Document revisie geschiedenis

Versie beheer

Revisie datum	Auteur	Versie	Omschrijving
15-03-2011	Danny van Tol	1.0	Opmaak document sub(koppen)
16-03-2011	Danny van Tol	1.1	Achtergrond bedrijf, planning
17-03-2011	Danny van Tol	1.2	Inleiding, probleem, doelstelling
19-03-2011	Danny van Tol	1.3	Probleem, doelstelling aangevuld
13-09-2011	Danny van Tol	Final	Plan van Aanpak afgerond

Tabel 1: Versie beheer

Distributielijst

Naam	Instantie	Rol/Functie	Datum
Eric Diks	OpenSesame ICT	Begeleider	11-07-2011
Nico de Vries	OpenSesame ICT	Opdrachtgever	11-07-2011
Okan Zor	Haagse Hogeschool	Begeleider (extern)	11-07-2011
Gerda in 't Veld	Haagse Hogeschool	Begeleider (extern)	11-07-2011
Administratie	Haagse Hogeschool	Distributie	11-07-2011

Tabel 2: Distributielijst

Inhoudsopgave

1	Inleiding.....	66
2	Achtergrond bedrijf	67
2.1	Het bedrijf OpenSesame ICT BV	67
2.2	OpenIMS	67
2.3	De afdelingen binnen OpenSesame ICT.....	69
2.4	Plaats van de student binnen de organisatie	69
3.	Probleemomschrijving	70
4	Doelstellingen.....	71
4.1	Doelstelling afstudeeropdracht	71
4.2	Doelstelling en competenties student.....	71
5	Afstudeeropdracht	72
5.1	beschrijving van de opdracht.....	72
5.2	Onderzoeksvragen	72
5.3	Scope van de opdracht	72
6	Projectplanning	73
6.1	Globale planning	73
6.1	Randvoorwaarden	73
6.2	Mijlpalen (op te leveren producten)	73
7	Gebruikte ontwikkelmethode DSDM	74
7.1	DSDM Fasen.....	74
7.2	Technieken DSDM (Timeboxing en MoSCoW).....	74
8	Kosten en baten	75
9	Projectrisico's	75
10	Literatuur	75

1 Inleiding

De door OpenSesame ICT ontwikkelde OpenIMS is een server product dat geheel in PHP geschreven is. Dit product dient als bouwsteen voor haar losse producten. Deze losse producten zijn; portaal, document, content, formulier en email management. Deze op service gerichte producten werken op een geïntegreerde manier met elkaar samen en vullen ieder een eigen functie in het systeem.

Binnen één van deze producten worden documenten en content bestanden beheerd door een management systeem, afgekort met (DMS) document management systeem en (CMS) content management systeem waarin documenten gemanaged worden door gebruikers die verantwoordelijk zijn voor bijvoorbeeld het goedkeuren of wijzigen van documenten.

In het document management systeem van OpenIMS kunnen documenten opgeslagen, bekeken en gewijzigd worden. Met dit documenten systeem zijn gebruikers niet in staat om opgeslagen documenten in real-time met meerdere gebruikers te bewerken. Mede hierdoor is het document management systeem binnen OpenIMS minder functioneel als zij willen. De behoefte is om de functionaliteit die Google Docs heeft, samen te laten werken met het documentensysteem van OpenIMS voor extra functionaliteit van documentenbeheer.

2 Achtergrond bedrijf

In dit hoofdstuk zal kennis gemaakt worden met het bedrijf OpenSesame ICT waar deze afstudeeropdracht wordt uitgevoerd.

2.1 Het bedrijf OpenSesame ICT BV

Het Nederlandse softwarebedrijf OpenSesame is in 2003 opgericht door de huidige drie directeuren. Het is ontstaan door een overname onderhandeling van Transiciel SA door CapGemini, waarbij OpenSesame als zelfstandig onderdeel verder ging. Inmiddels is het bedrijf uitgegroeid tot een middelgroot bedrijf, dat zich in Nieuwegein heeft gevestigd.

OpenSesame ICT presenteert zich met het product OpenIMS, een pakket dat door het bedrijf zelf ontworpen en ontwikkeld is met Microsoft Producten. Dit product is doorontwikkeld, waarbij de functionele scope is uitgebreid. Daarbij is ook de Microsoft technologie vervangen voor een platformonafhankelijke technologie en zijn vele Open Standaarden geïmplementeerd, waaronder de Open Source technologie LAMP (Linux, Apache, MySQL en PHP). Het grote voordeel hiervan is dat OpenIMS ook op OSX en Microsoft georiënteerde systemen kan draaien.

Naast het ontwikkelen van software als kernactiviteit, levert OpenSesame ook andere diensten zoals ontwikkeling, implementatie, integratie en beheer van de OpenIMS standaard software. Al dan niet in combinatie met andere (specialistische) pakketten zoals SugarCRM, EasyData OCR, en PDF converters.

Gebieden waarin OpenSesame ICT actief is, zijn de overheid, gezondheidszorg, onderwijs en de zakelijke dienstverlening;

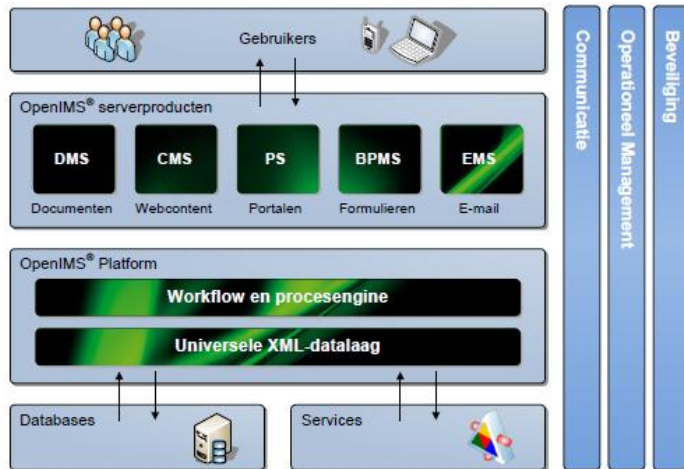
- De Nederlandse Spoorwegen
- Fuji Film
- HGM (Hulpverlening Gelderland Midden)
- GTI Zuidoost
- ASM
- Kinder- en jeugdpsychiatrie Oost Nederland
- Deventer Ziekenhuis
- Nederlandse Overheid (waarvan een aantal ministeries)

2.2 OpenIMS

Het belangrijkste product van OpenSesameICT is OpenIMS, wat staat voor Open Informatie Management Server. Dit is het platform waarop alle producten draaien en dat grotendeels met open - Source technologie gemaakt is, waaronder PHP en Apache server.

OpenIMS is een volledige Object Georiënteerd web-gebaseerde server oplossing, dat ingezet kan worden voor het beheren en onderhouden van niet- structurele informatie. Dit zijn werkprocessen of projecten waarin grote hoeveelheden documenten, webpagina's en e-mailberichten worden geproduceerd en uitgewisseld. Deze informatie staat vaak op websites, intranet, file- servers , lokale harde schijven en in e- mailberichten. Met OpenIMS kan ervoor gezorgd worden dat bijna alle belangrijke informatie binnen iedere organisatie kan worden gestructureerd en beheerd worden zodat er altijd over de juiste en actuele informatie wordt beschikt.

In het onderstaande figuur staat weergegeven hoe OpenIMS is opgebouwd en is te zien hoe de afzonderlijke componenten communiceren met het Platform. Van ieder product wordt een nadere beschrijving gegeven.



Figuur 1 : OpenIMS structuur

DMS : Document Management Server is ontwikkeld om alle denkbare documenten te repliceren op een data-server. Documenten worden gesynchroniseerd met de documenten op de server, waardoor iedereen met dezelfde document werkt.

CMS : Content Management Server voor het beheren van content (tekst, tabellen, berekeningen, presentaties en dergelijke). Voor het aanpassen van content op bijvoorbeeld het intranet of de website van een organisatie.

PS : Portal Server is het platform waarmee informatie uit de verschillende applicaties beschikbaar gesteld kan worden. Er zijn verschillende portals beschikbaar zoals to-do-lijsten, laatst bekeken dossiers, direct zoeken in het Document Management Systeem of de toegewezen webcontent uit het Content Management Systeem.

BPMS : Bedrijfs Proces Management Server, een management-gereedschap om workflows zoals de communicatie in een werkproces te stroomlijnen. Voorbeelden hiervan zijn activiteiten zoals het verzamelen en versturen van informatie zoals documentatie die als één geheel moet worden afgehandeld.

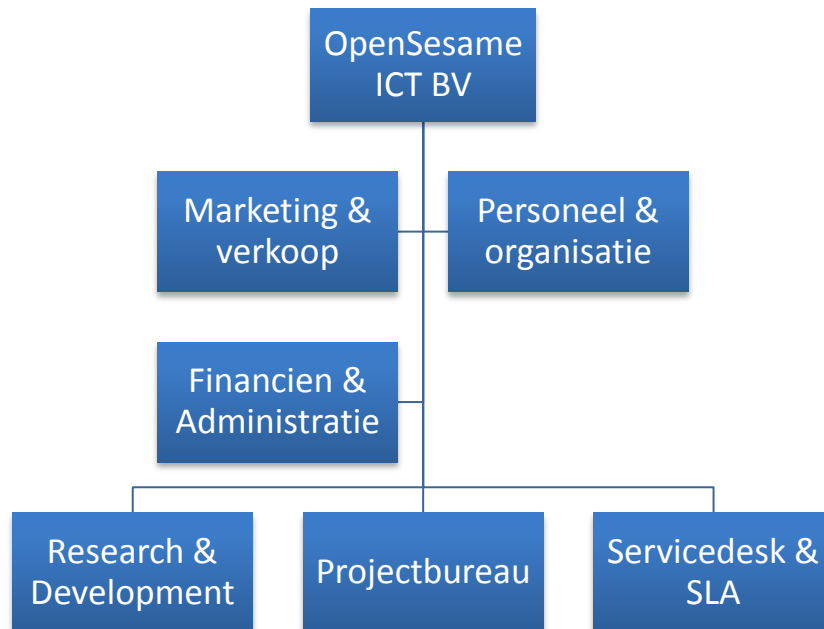
EMS : E-mail Management Server, voor het archiveren van mail berichten binnen een organisatie. Berichten worden in deze tool geïndexeerd en met bijlagen in het IMS-systeem opgeslagen. Dat kan handig zijn wanneer er bijvoorbeeld in projecten de communicatie vanuit verschillende persoonlijke mail-accounts gebeurt.

Voorbeelden waarbij OpenIMS is ingezet zijn;

- Beheer van website, portal, intranet en/of extranet.
- Het met behulp van een web browser beheren en ontsluiten van diverse bedrijfsinformatie zoals offertes, contracten, wetstukken, projectplannen, tekeningen en andere documenten, eenvoudige CRM functionaliteiten en andere formuliergestuurde gegevensverwerking.
- Kwaliteitsmanagement (ISO certificering, HKZ-model).

2.3 De afdelingen binnen OpenSesame ICT

De plaats waar de opdracht wordt uitgevoerd is OpenSesame ICT. Dit is een projectorganisatie waar gebruikelijk de meeste werknemers op kantoor werkzaam zijn. Afgestudeerden of stagiaires vallen (in de meeste gevallen) onder de afdeling Research & Development afdeling. Hieronder zijn de verschillende afdelingen binnen het bedrijf te zien in een organigram:



2.4 Plaats van de student binnen de organisatie

Binnen het kantoor van OpenSesame ICT is er voor stagiaires en afgestudeerden een speciale afgezonderde ruimte gereserveerd. Waar studenten een werkplek, een desktop computer en het gebruik van de kantoor omgeving wordt aangeboden, om zich te ontwikkelen met de opdracht die hij met zijn afstudeer begeleider(s) is overeengekomen. De student voert zijn werkzaamheden uit, volbrengt zijn opdracht en ontvangt daarvoor een maandelijkse vergoeding.

3. Probleemomschrijving

Google levert middels Google Docs een omgeving waarin online in the Cloud documenten kunnen worden beheerd met een eenvoudige online tekstverwerker en spreadsheet. Typische Document Management System functionaliteit als metadata en workflow is in Google Docs nauwelijks beschikbaar. Insteek van het project is om de (on)mogelijkheden van interfacing met Google Docs te bepalen en vervolgens een koppeling tussen OpenIMS DMS en Google Docs te realiseren. Functioneel kan hierbij gedacht worden aan tweeweg synchronisatie van bestanden en het over en weer beschikbaar stellen van functionaliteiten.

OpenSesame ICT heeft een DMS in het OpenIMS pakket. Dit DMS heeft echter zijn beperkingen. In het DMS kunnen bestanden geupload en bewerkt worden. Dit kan echter maar door één persoon tegelijk (één kant op). Er is geen functie beschikbaar zoals in Google Docs, waarbij meerdere mensen in staat zijn om één document te wijzigen, zonder het risico dat er informatie verloren kan gaan.

OpenSesame ICT wil daarom een uitbreiding van het DMS waarbij er een koppeling komt, tussen OpenIMS en Google Docs, zodat men ook in OpenIMS in staat is om deze functionaliteit te gebruiken. Tot op heden is binnen het bedrijf nog niets gedaan met een koppeling van het DMS en Google Docs.

4 Doelstellingen

Binnen mijn afstudeer project zijn er bepaalde doelstellingen. De doelstellingen zijn onderverdeeld in doelstellingen van zo wel de afstudeeropdracht zelf, als de doelstelling van de student. Bij de doelstellingen van de student kan men denken aan bepaalde competenties die de student wil behalen.

4.1 Doelstelling afstudeeropdracht

De doelstelling aan het einde van de afstudeerperiode is een helder beeld te hebben wat Google Docs kan, welke mogelijkheden er zijn voor een koppeling van Google Docs met het OpenIMS DMS. De mogelijke toepasbare functionaliteit ook daadwerkelijk te realiseren. De koppeling moet ervoor zorgen dat werknemers binnen het bedrijf in de Cloud real-time hun documenten kunnen editen en of wijzigen.

4.2 Doelstelling en competenties student

Hieronder een aantal competenties (beroepstaken) weergegeven welke aan de orde komen tijdens het afstudeer traject:

1.1 Selecteren, methoden, technieken en tools

Voor het uitvoeren van de afstudeeropdracht zal gebruik worden gemaakt van methoden, technieken en tools. Gedurende het afstudeertraject zal gebruik worden gemaakt van DSDM. Hieronder vallen o.a. de DSDM fasen zelf, en de MoSCoW methode voor het opstellen van de requirements. Bij de must have requirements zal gebruik worden gemaakt van timeboxing voor de tijdige oplevering en realisatie van het product.

Daarnaast zijn er een aantal vastgestelde technieken/ talen die gebruikt gaan worden tijdens het afstuderen. Dit is voor het overgrote deel met: PHP en JavaScript.

3.2 Ontwerpen systeemdeel (software)

Alvorens ontwikkeld kan worden aan het product, dient een ontwerp te worden gemaakt. Het is van belang dat bij het ontwerp ook eigen ideeën (inbreng) in het ontwerp wordt meegenomen. In overleg met de begeleiders kunnen zij een goedkeuring geven om het ontwerp uit te werken in praktijkvorm. Hier dient tevens rekening gehouden te worden met de bestaande software architectuur.

3.3 Bouwen applicatie

Dit betreft het ontwikkelen van een koppeling tussen OpenIMS en Google Docs. Hierbij zullen eigen ideeën een bepalende rol meespelen. In overleg kunnen mijn ideeën meegenomen voor het ontwerp en bouw van nieuwe functionaliteit(en). Hiervoor zullen eerst de mogelijkheden onderzocht moeten worden. Indien mogelijk zal er met het ontwikkelproces ook gebruik worden gemaakt van een externe bibliotheek (Gdata) en de Google Docs API.

3.4 Initiëren en plannen van het testproces

Om te testen of het gemaakte product aan de verwachting van de opdrachtgever(s) voldoet. Hoe en met welke methodiek getest wordt, wordt nog nader bepaald in het plan van aanpak.

3.5 Uitvoeren van en rapporteren over het testproces

Bij het uitvoeren van tests zullen deze vastgelegd worden in een testrapport. Het testrapport wordt opgeleverd bij de overdracht van alle producten aan het einde van de afstudeerperiode.

5 Afstudeeropdracht

5.1 beschrijving van de opdracht

Het ontwikkelen van een koppeling met OpenIMS en Google Docs. De bedoel is dat OpenIMS en Google bestanden uit kunnen wisselen met behulp van een tweeweg synchronisatie. Daarmee heeft het OpenIMS documenten systeem (DMS) van OpenSesame ICT extra functionaliteit. Middels Google Docs kunnen documenten namelijk real-time gewijzigd worden. Dit houdt in dat meerdere gebruikers tegelijk een document kunnen wijzigen, zonder er gegevens verloren gaan. Het is dan ook van essentieel belang dat er o.a. documenten geüpload, gedownload, geüpdatet kunnen worden waarna vervolgens met deze functionaliteit tweeweg synchronisatie functionaliteit gebouwd kan worden om beide systemen equivalent aan elkaar te houden.

5.2 Onderzoeksvragen

Voor het project zijn een tweetal onderzoeksvragen opgesteld welke centraal staan in dit afstudeertraject:

- Onderzoek doen naar de bestaande functionaliteit van Google Docs. Wat is het, hoe werkt het, en hoe het is opgebouwd?
- Wat is, en kan het OpenIMS document management systeem? Welke functies binnen OpenIMS zijn beschikbaar, en hoe kan deze functionaliteit met Google Docs samenwerken?

5.3 Scope van de opdracht

Het afbakenen van het project zal ook deels geregeld worden met de gebruikte ontwikkelmethode DSDM. Voor het opstellen van de requirements zal gebruik worden gemaakt van de MoSCoW methode. Hierdoor kan een gedeelte van het te ontwikkelen product vastgelegd worden wat als basisfunctionaliteit gerealiseerd moet worden.

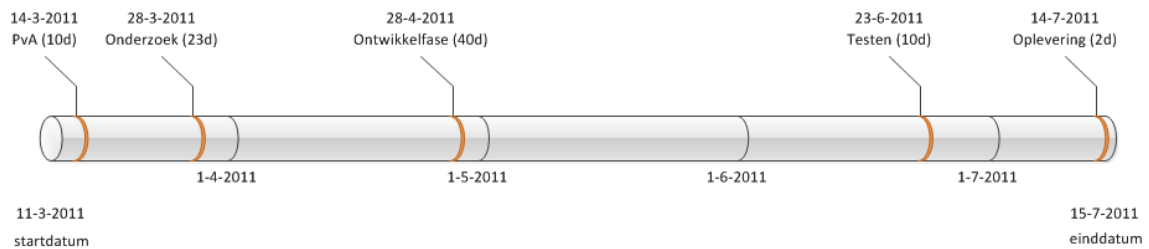
Met de scope van het project wordt een grens vastgelegd waarbinnen dit project plaatsvindt. Dit zijn de volgende punten;

- (5) Dit project betreft alleen documenten beheer en geen andere bestanden.
- (6) Het OpenIMS document management systeem moet zijn oude functionaliteit blijven behouden.
- (7) Nieuwe functionaliteit van Google Docs moet samenwerken met de bestaande DMS functionaliteit van OpenIMS
- (8) Synchronisatie functionaliteit vindt alleen plaats tussen OpenIMS en Google Docs en geen derde of andere software pakketten.

6 Projectplanning

In de projectplanning staan welke activiteiten gedurende het afstudeerproject ondernomen worden en welke eind- producten daarvan het resultaat zijn.

6.1 Globale planning



Figuur 1: Globale planning afstudeerproject

6.1 Randvoorwaarden

De planning is gemaakt op basis van de volgende randvoorwaarde waaraan moet worden voldaan;

- Het afstudeerproject zal lopen vanaf 11 maart t/m 15 juli 2011. In deze periode zal het afstudeerproject moeten worden afgerond.
- De totale afstudeeropdracht zal circa 820 studiebelastinguren voor de student opleveren.
- Ondersteuning en volledige beschikking tot documentatie en software van OpenSesame ICT.
- Zowel de bedrijfs- als de docent begeleider zijn betrokken met het afstudeertraject en bieden ondersteuning waar nodig.

6.2 Mijlpalen (op te leveren producten)

Product	Start	Deadline	
Plan van aanpak	15-03-2011	25-03-2011	
Afstudeer-verslag	26-03-2011	23-09-2011	(inlever datum)
Product (ontwikkelfase)	28-04-2011	15-07-2011	(einddatum bij OpenSesame ICT)
Testrapport	11-09-2011	23-09-2010	(inlever datum)

7 Gebruikte ontwikkelmethode DSDM

Om gestructureerd te kunnen werken, zal er volgens een ontwikkelmethodiek gewerkt worden. In dit geval lijkt DSDM mij de beste keuze. Deze ontwikkelmethodiek heb ik namelijk op school ook gebruikt tijdens projecten, en is mij daardoor vertrouwd. Tevens zitten er goede technieken bij als MoSCoW voor het prioriteren van de requirements en timeboxing voor het op tijd afleveren van een product binnen een bepaalde tijd. Normaal gesproken speelt ook budget hierbij een rol, maar doordat het een afstudeeropdracht betreft is dit niet het geval.

7.1 DSDM Fasen

Het DSDM-proces bestaat uit vijf fasen:

6. *Feasibility Study*
In deze fase wordt gekeken of DSDM een geschikte methode is voor de situatie.
7. *Business Study*
In deze fase wordt de projectorganisatie opgericht, er wordt een planning op grote lijnen gemaakt, er vindt een risico inventarisatie plaats, de architectuur voor de software wordt bepaald en een lijst met eisen en randvoorwaarden op hoog niveau wordt vastgesteld.
8. *Functional Model Iteration*
Deze fase kan uit één of meerdere iteraties bestaan afhankelijk van de complexiteit van het project. In deze fase wordt gewerkt aan functionele prototypes die bedoeld zijn om de gewenste functionaliteit duidelijk te krijgen.
9. *Design and Build Iteration*
Tijdens deze fase wordt de software in één of meerder iteraties, of in geval van dit project incrementele fases ontworpen en gebouwd. Dit zal in combinatie met UML gedaan worden.
10. *Implementation*
De term implementatie is verwarrend, omdat hiermee vaak het 'programmeerwerk' wordt bedoeld. In dit geval gaat het om het in gebruik nemen van de software. De overdracht naar OpenSesame ICT in dit geval.

Niet alle fasen zullen gebruikt worden tijdens dit project, omdat deze fasen niet relevant zijn voor dit afstudeerproject. De dikgedrukte fasen zullen wel gebruikt worden in combinatie met UML van RUP. Door de flexibiliteit van DSDM kunnen deze twee ontwikkelmethodieken met elkaar gecombineerd worden.

7.2 Technieken DSDM (Timeboxing en MoSCoW)

De techniek die het meeste geassocieerd wordt met DSDM is timeboxen en combinatie met MoSCoW. Bij de start van een increment zijn tijd en budget vastgelegd, dit wordt een timebox genoemd. De volgende stap is om de gewenste functionaliteit in kaart te brengen en de prioriteiten daarvan vast te leggen. MoSCoW is een manier om deze prioriteiten aan te geven. MoSCoW staat voor Must-have, Should-have, Could-have en Won't-have-this-time. Binnen een timebox wordt gezorgd dat de functionaliteiten met de hoogste prioriteit als eerste gemaakt worden, dit zijn de Must-haves. Als er tijd over is wordt er gewerkt aan de Should-haves enz.

Deze essentiële technieken zoals MoSCoW prioritering en timeboxing zullen dan ook gebruikt worden tijdens dit project.

8 Kosten en baten

Per week werkt de student gemiddeld zo'n 40 uur aan het project. Dit houdt in dat er aan het einde van de afstudeerperiode zo'n 820 uur aan het project gewerkt is. Als resultaat dient een product te zijn ontwikkeld, wat er toe een bijdragen een waarde te vertegenwoordigen voor het bedrijf OpenSesame ICT. De student ontvangt tevens een vergoeding voor het afstuderen.

Daar tegenover staat dat de student zijn kennis op verschillende terreinen binnen het bedrijfsleven vergroot zal zijn. De student zal tevens beloond worden met 30 ECTS, mits het afstuderen niet behaald word.

9 Projectrisico's

Bedreiging	Tegenmaatregel
Uitloop van activiteit(en)	Regelmatig bijstellen van de planning (gebruik van timeboxing)
Verlies van data (hard falen)	Back-up maken van alle source code en documentatie
Bronnen niet bereikbaar	Informatie verkrijgen via begeleider(s)

10 Literatuur

Onderwerp	Referentie / Naslagwerk
Informatie over het bedrijf	http://www.osict.com/
Documentatie over OpenIMS	http://Intranet.osict.com/
Google Docs API	http://code.google.com/intl/nl/apis/documents/
Zend Framework	http://framework.zend.com/manual/en/learning.quickstart.intro.html/
PHP help (beschikbare functies)	http://www.PHP.net/
Overige informatie	http://www.google.nl/
PHP boek NL	Cursus PHP5
PHP boek EN	Development with PHP (professional)

Testplan

“Afstudeeropdracht: Koppeling Google Docs met OpenIMS”



Datum : 18-09-2011

Plaats : Lelystad

Auteur : Danny van Tol

Studentnr : 20057842

Opleiding : Informatica

Testplan versie : Final

Versiebeheer

Versie	Datum	Auteur	Omschrijving
0.1	18-09-2011	Danny van Tol	Opmaak document
0.2	19-09-2011	Danny van Tol	Inhoudelijk (sub)koppen beschreven
0.3	20-09-2011	Danny van Tol	Inhoud aangepast
0.4	20-09-2011	Danny van Tol	Inhoudsopgave aangepast, inhoud aangepast
Final	21-09-2011	Danny van Tol	Document afgerond als Final

Tabel 1: versiebeheer

Distributielijst

Naam	Instantie	Rol/Functie	Begindatum	Einddatum
Eric Diks	OpenSesame ICT	Begeleider (intern)	11-03-2011	23-09-2011
Nico de Vries	OpenSesame ICT	Opdrachtgever	11-03-2011	23-09-2011
Okan Zor	Haagse Hogeschool	Begeleider (school)	11-03-2011	23-09-2011
Gerda In 't Veld	Haagse Hogeschool	Begeleider (school)	11-03-2011	23-09-2011
Administratie	Haagse Hogeschool	Distributie	11-03-2011	23-09-2011

Tabel 2: Distributielijst

Inhoudsopgave

1	Inleiding.....	79
1.1	Doel van dit document	79
1.2	Referenties	79
2	Opdrachtformulering.....	80
2.1	Binnen de scope	80
2.2	Buiten de scope (afbakening)	80
2.3	Randvoorwaarden	80
3	Testbasis.....	81
4	Teststrategie.....	82
4.1	Samenvatting productrisico-analyse	82
4.2	Testtechnieken	84
4.3	Verdeling van de testinspanning	84
5	Testaanpak	85
5.1	Methoden en Testtechnieken	85
5.2	Testproducten	85
5.3	Opleiding (kennis)	85
6	Gebruikte testtool(s) en hulpmiddelen	86
6.1	Testtools	86
6.2	hulpmiddelen	86

1 Inleiding

Dit testplan is geschreven naar aanleiding van mijn afstudeeropdracht bij OpenSesame ICT BV. Het is bedoelt als leidraad voor het op te leveren testrapport. Het bevat onderdelen als de opdrachtformulering, de scope (wat wel testen, wat niet), het kiezen van een testbasis et. cetera.

1.1 Doel van dit document

Dit testplan beschrijft de testaanpak die de student hanteert. Het besteedt aandacht aan: testscope, testorganisatie, testtypen, testhulpmiddelen, testtooling en testomgeving.

1.2 Referenties

Nr.	Onderwerp	Bron
1	Testtechnieken	http://nl.wikipedia.org/wiki/Testplan
2	TMap	http://en.wikipedia.org/wiki/Test_Management_Approach
3	Testen algemeen	Boek: TestGoal (educatief gericht)
4	Testen algemeen	Boek: Testen volgens TMap 2 ^e druk

2 Opdrachtformulering

Elke testtechniek heeft weer een ander doel bij het testen van een systeem. Het algemene doel van het testen van mijn afstudeerproject is, of het opgeleverde product voldoet aan de gestelde kwaliteitseisen van de opdrachtgever c.q. begeleider van OpenSesame ICT.

Daarnaast is het belangrijk dat bugs en onlogische zaken snel opgemerkt worden. Door het testen kan de planning eventueel aangepast worden en bepaalde zaken worden herzien en waar nodig verbeterd.

De testinspanningen binnen het ontwikkelteam hebben de volgende doelen:

- Een bijdrage leveren aan de kwaliteit van het eindproduct.
- Meetbaar maken van de kwaliteit van het eindproduct.
- Beoordelen van de geschiktheid voor oplevering van het eindproduct.

2.1 Binnen de scope

Binnen de scope van dit testtraject zal het testen zich alleen richten tot de ontwikkelde functionaliteit van de systemen waar tijdens het afstuderen mee gewerkt is. Dit is het documenten systeem (DMS) van OpenIMS, en Google Docs waarmee on the cloud documenten beheerd kunnen worden.

2.2 Buiten de scope (afbakening)

Met betrekking tot het uitvoeren van de tests zal ik mij beperken tot het testen van het eindproduct. Dit houdt in dat niet alle subonderdelen apart getest gaan worden, maar alleen het eindproduct (de algehele synchronisatie). De reden hiervan is dat in een later stadium van het afstudeerproject ervoor is gekozen om niet volgens de manier van DSDM te testen (gehele levenscyclus door), maar een apart testplan en een apart testrapport op te stellen.

De subonderdelen zijn overigens wel getest via de DSDM methode (gedurende de ontwikkelfase) op een correcte werking. De voorbereiding, en resultaten zijn alleen niet opgenomen in het testplan c.q. testrapport.

2.3 Randvoorwaarden

Om van start te kunnen met het testen is het belangrijk dat er aan een aantal voorwaarden is voldaan:

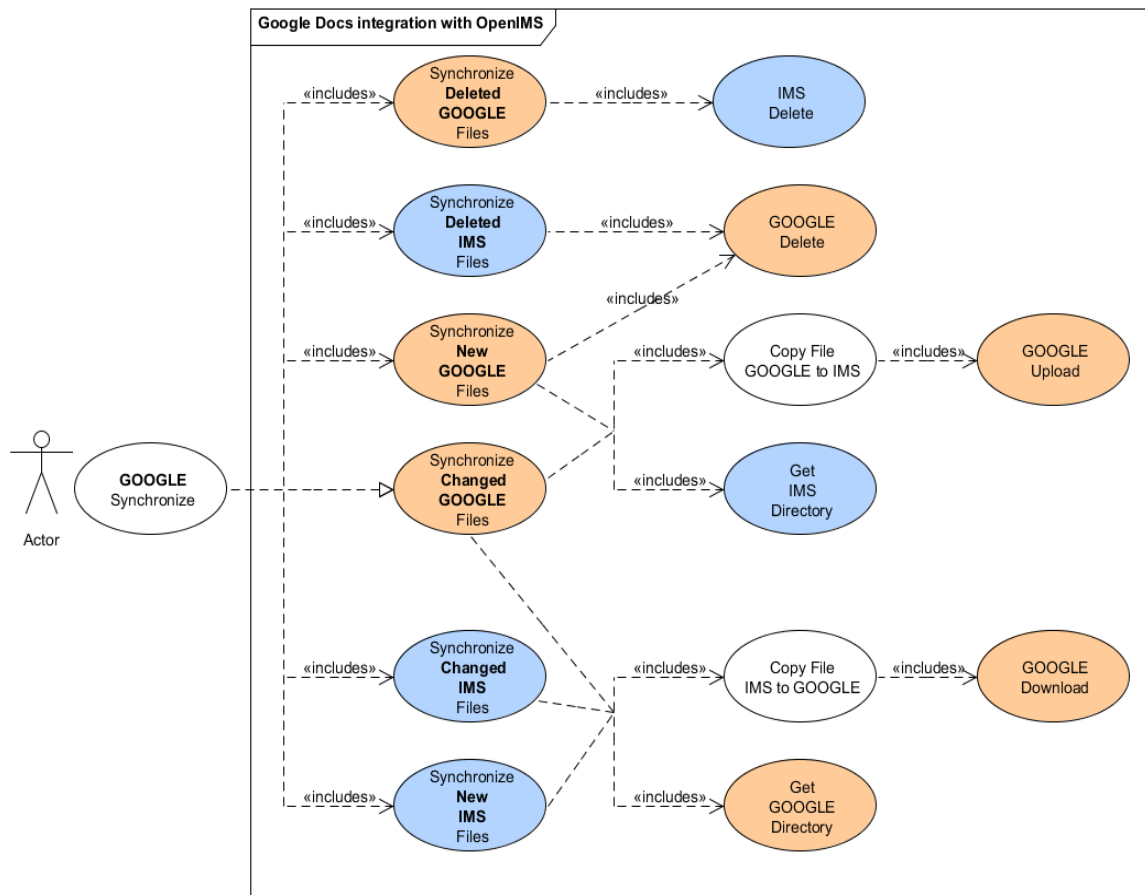
- Het product moet functionaliteiten bevatten voor uitvoerbaarheid van tests
- De resources (in dit geval code) moet aanwezig zijn om te kunnen testen
- Er dient voldoende tijd te zijn om beschikbaar te zijn om te kunnen testen
- Geschikte hardware is nodig om de tests uit te kunnen voeren

Bij het ontbreken van één van deze punten, is de tester niet in staat om überhaupt te kunnen testen.

3 Testbasis

Het volgende werkproduct wordt gebruikt als basis voor de testgevallen:

- USE-CASE MODEL (SYNCHRONISATIE FUNCTIONALITEIT)
- CODE (INDIEN NODIG)



Figuur 1: Het Use-case model van de synchronisatie functionaliteit

De reden dat het Use-case model gekozen als testbasis gekozen is, is omdat alle functionaliteit van het eind-product hierin opgesloten zit, en het dus volstaat bij het uitvoeren van de Reallife, en Error Guessing tests.

Voor de Programma interfacetest zal gebruik worden gemaakt van de code indien nodig.

4 Teststrategie

De testopdracht bestaat uit het testen van de software die ontwikkeld is tijdens mijn afstudeerperiode. Dit betreft een tweeweg synchronisatie functionaliteit van het OpenIMS documenten systeem (DMS) met Google Docs. Van beide kanten uit, zal getest worden of de document overdracht bij nieuwe, gewijzigde, en verwijderde documenten goed zal werken. Dit betreft van OpenIMS naar Google Docs, en van Google Docs naar OpenIMS.

De volgende testobjecten zullen als volgt getest worden:

- 1) Getest wordt of bij het nieuw aangemaken van een document in OpenIMS, dit document ook als nieuw overgeplaatst wordt naar Google Docs.
- 2) Als een document gewijzigd wordt in OpenIMS, deze wijzigingen eveneens aangepast zijn in het document op de Google Docs server.
- 3) Als een document verwijderd wordt uit OpenIMS, het document na het uitvoeren van de synchronisatie, ook verwijderd is uit Google Docs.
- 4) Het ophalen van de OpenIMS directory (getIMSdir).

Ditzelfde geldt ook aan de kant van Google Docs:

- 1) Getest wordt of bij het nieuw aanmaken van een document in Google Docs, ook nieuw document aangeemaakt wordt in OpenIMS.
- 2) Of een gewijzigd document in Google Docs, ook gewijzigd wordt in OpenIMS.
- 3) Als een document verwijderd wordt uit Google Docs, het document na het uitvoeren van de synchronisatie, ook verwijderd is uit OpenIMS.
- 4) Het ophalen van de Google Docs directory (getGOOGLEdir).

Daarnaast zal ook getest worden op dingen als vreemde tekens in de naam van een document, de hoofdlettergevoeligheid, en de content types (extensies) van documenten.

4.1 Samenvatting productrisico-analyse

Per onderdeel is een risico-analyse gedaan waarbij drie risicoklassen zijn gedefinieerd: Hoog, middel en laag.

Kenmerk / - onderdeel	Risicoklasse	Argumentatie
Gebruiksvriendelijkheid	Hoog	Gebruiker moet er met tevredenheid mee kunnen werken.
Performance	Midden	Wel belangrijk maar heeft geen overhead.
Beveiliging	Laag	De gebruiker dient vertrouwelijk met login gegevens te zijn.
Beheerbaarheid	Laag	Beheer is minder van belang. Sync kan via batch bestand.

Betrouwbaarheid	Hoog	Het moet doen waar het voor gemaakt is.
Uitbreidbaarheid	Hoog	In de toekomst moet men dit uit kunnen breiden.

4.2 Testtechniek

Binnen het testtraject worden de volgende testtechnieken toegepast.

Testtechniek	Beschrijving	Testbasis	Doel
1) Real-life test	Met deze testtechniek is het niet de bedoeling extreme situaties te testen, maar het normale gebruik. Er wordt gekeken naar aspecten als: bruikbaarheid, continuïteit en performance.	Alle	Het ontdekken van fouten die te maken hebben met de uiteindelijke "omvang" van het informatiesysteem.
2) Programma Interfacetest	Er wordt geverifieerd of de programma's c.q. modules na integratie nog steeds correct functioneren.	Programma – code of ontwerp	Testen van de interfaces tussen de verschillende programma's c.q. modules: het controleren of de verschillende programma's de gegevensstromen alle op dezelfde manier interpreteren.
3) Error Guessing Test	Error Guessing staat voor het raden van fouten.	Alle	Doel is om te raden waar het eventueel fout kan gaan binnen het systeem.

Aanvullende literatuur: Testen volgens Tmap 2^e druk

4.3 Verdeling van de testinspanning

Kenmerk / - onderdeel	Risicoklasse	Systeemtest	Acceptatietest	Regressietest	Productietest
[Zie ook §4.1]	[Zie ook §4.1]	[testinspanning ●, ●● of ●●●]	[testinspanning ●, ●● of ●●●]	[testinspanning ●, ●● of ●●●]	[testinspanning ●, ●● of ●●●]
Gebruiksvriendelijkheid		●●	●●	●●	●●●
Performance		●●	●	●	●
Beveiliging		●	●	●	●
Beheerbaarheid		●	●	●	●
Betrouwbaarheid		●●●	●●●	●●	●●
Uitbreidbaarheid		●●●	●●●	●●	●●●

5 Testaanpak

De in §4.2 aangegeven testtechnieken zullen gebruikt worden voor het testen van de synchronisatie functionaliteit. De resultaten zullen van de tests worden beschreven in het testrapport. Om het visueel begrijpelijker te maken zal naast het verzamelen van resultaten, zoveel mogelijk gebruik worden gemaakt van afbeeldingen en screenhots.

5.1 Methoden en Testtechnieken

Voor het uitvoeren van de tests en het gebruikmaken van de gekozen testtechnieken, wordt gebruik gemaakt van TMap als methode. TMap is gedurende mijn opleiding Informatica het meest gebruikt tijdens blokken, en daardoor het beste bekend en het fijnst om mee te werken.

TMap biedt technieken voor de volgende onderwerpen:

- analyseren van productrisico's
- ontwerpen van tests
- toetsen van producten.

5.2 Testproducten

Werkproduct	Toelichting
Testplan	Een Testplan draagt bij aan een goede organisatie om te kunnen testen. Het is een voorbereiding op de uit te voeren tests.
Testrapport	In het Testrapport komen de resultaten van de tests te staan.

5.3 Opleiding (kennis)

Voor het uitvoeren van de testopdrachten is voldoende kennis. Tijdens de opleiding informatica is bij een aantal blokken (waaronder I-1, I-2, I-3 en 1-4) testen aan bod gekomen. Aan het begin van mijn opleiding werd er getest met TMap, later is dit overgegaan op TestGoal en TMap Next. De reden dan TMap gekozen is voor het uitvoeren van de tests is omdat TestGoal teveel gericht is op de educatieve kant.

6 Gebruikte testtool(s) en hulpmiddelen

Dit hoofdstuk beschrijft de testtools en hulpmiddelen die gebruikt gaan worden bij het uitvoeren van de tests.

6.1 Testtools

Voor het uitvoeren van acceptatietests en error guessing wordt gebruik gemaakt van een webbrowser. De volgende versies van webbrowsers worden ondersteund:

- Mozilla Firefox 6.0.x of hoger
- Microsoft Internet Explorer 8 of hoger

6.2 hulpmiddelen

Als hulpmiddel bij het testen is het boek TMap. Aan de hand van dit boek worden de genoemde tests uitgevoerd. De tests zullen uitgevoerd worden op een eigen laptop waarop de software en benodigde data aanwezig is om te testen. Alle resultaten van de tests zullen vastgelegd worden in het testrapport.

Testrapport versie : Final

Versiebeheer

Versie	Datum	Auteur	Omschrijving
0.1	21-09-2011	Danny van Tol	Opmaak, Inhoud toegevoegd
0.2	22-09-2011	Danny van Tol	Inhoud toegevoegd
Final	23-09-2011	Danny van Tol	Final testrapport

Tabel 1: versiebeheer

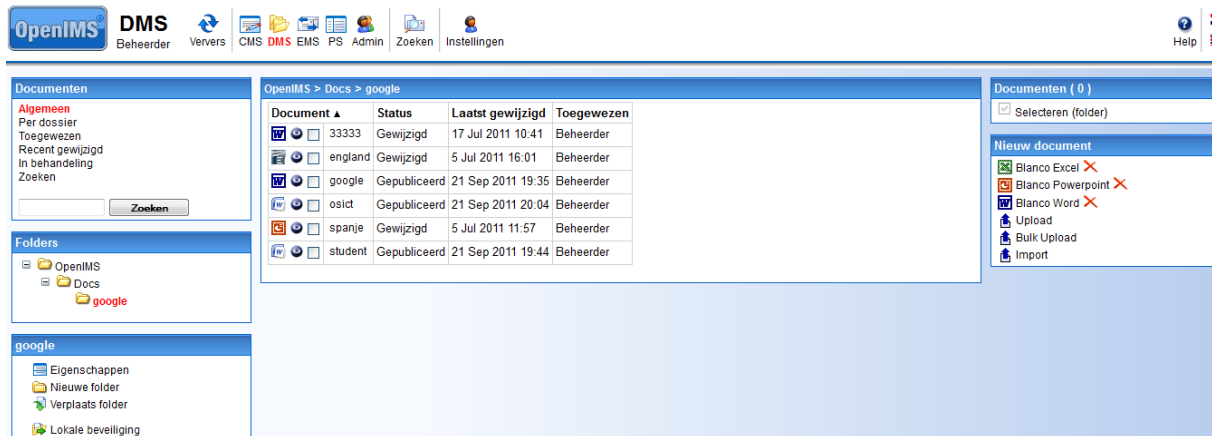
Distributielijst

Naam	Instantie	Rol/Functie	Begindatum	Einddatum
Eric Diks	OpenSesame ICT	Begeleider (intern)	11-03-2011	23-09-2011
Nico de Vries	OpenSesame ICT	Opdrachtgever	11-03-2011	23-09-2011
Okan Zor	Haagse Hogeschool	Begeleider (school)	11-03-2011	23-09-2011
Gerda In 't Veld	Haagse Hogeschool	Begeleider (school)	11-03-2011	23-09-2011
Administratie	Haagse Hogeschool	Distributie	11-03-2011	23-09-2011

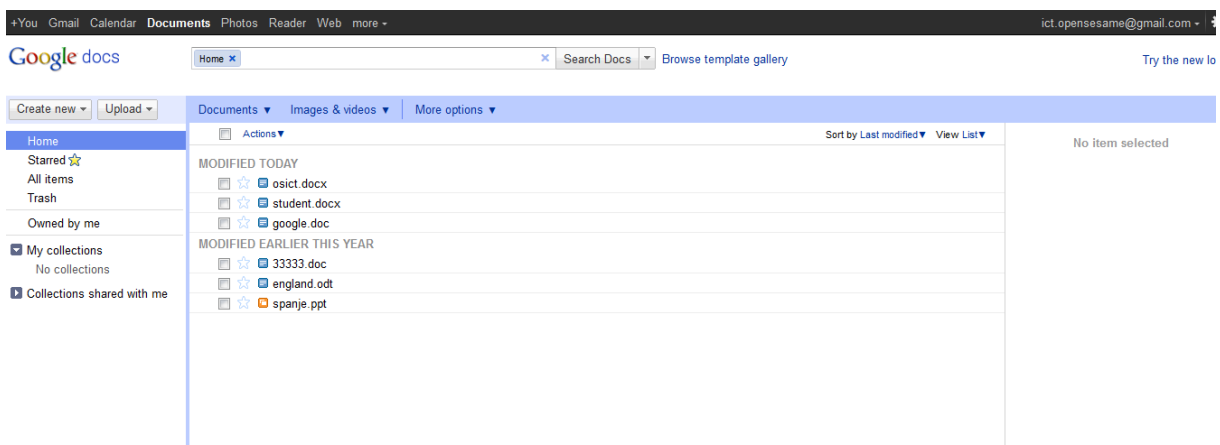
Tabel 2: Distributielijst

In dit testrapport worden drie testspecificatietechnieken uiteengezet voor het testen van OpenIMS met Google Docs. Als leidraad voor dit rapport is gebruik gemaakt van het vooropgestelde testplan.

Voorafgaand aan de tests is het wellicht nuttig om een visueel beeld te hebben van de testomgevingen. In de afbeeldingen hieronder zie je de omgevingen van het OpenIMS DMS, Google Docs, en de codetester van OpenIMS.



Figuur 1: OpenIMS documenten omgeving



Figuur 2: Google Docs omgeving



Figuur 3: Handmatig uitvoeren van de synchronisatie met de codetester van OpenIMS

Testontwerptechniek: Real-life test

Algemeen

Het doel van de real-life test is fouten te ontdekken die te maken hebben met de uiteindelijke “omvang” van het informatiesysteem. De test moet dus zoveel mogelijk de werkelijke situatie nabootsen ten aanzien van o.a. het aantal verwerkingen, het aantal gebruikers en de belasting van het systeem in de loop van de tijd.

Er wordt gekeken naar aspecten als: bruikbaarheid, continuïteit en performance. Gekeken wordt of de normale hoeveelheden in een acceptabele snelheid afgehandeld kunnen worden. Er wordt met name gekeken of de omgeving, en de resources (benodigde middelen) een normale belasting aan kunnen. Dit dus in tegenstelling tot een stresstest waarbij de grens wordt opgezocht en het systeem wordt belast totdat het kapot gaat.

Fouten die met real-life testen gevonden worden, houden doorgaans verband met het resourcegebruik van het systeem. In het geval van dit project kunnen dit de volgende zaken zijn:

- De responsetijden zijn onvoldoende
- De snelheid van de batchverwerking is onvoldoende
- De Google server is te traag
- De OpenIMS server is te traag
- Internetverbinding zelf is traag

Werkwijze

Om een real-life test uit te voeren, moeten de volgende stappen ondernomen worden:

- Profielschets systeemgebruik
- Specificeren testgevallen
- Testuitvoering
- Resultaat, Conclusie

1) Het maken van een profielschets van het systeemgebruik

Een profielschets geeft op hoofdlijnen aan welke acties (functies) gedurende een bepaalde periode worden uitgevoerd en het aantal *actieve* gebruikers. Hierbij kan gedacht worden aan een aantal dagdelen waarop een maximaal, gemiddeld, of minimaal verbruik is van het systeem. In de tabel hieronder is één dag verdeeld in een aantal dagdelen. Deze dagdelen zijn in de vorm van tijd geschetst. De verschillende activiteiten zijn ook een dagdeel gekoppeld als het “Inloggen” als men aankomt op het werk, het intensieve gebruik, normale gebruik, de lunchpauze, en het uitloggen van medewerkers als ze weer naar huis gaan.

Voor deze test zijn alleen de werknemers van OpenSesame ICT genomen als leidraad voor het gebruik van deze functionaliteit. Op basis van de hoeveelheid medewerkers die bij OpenSesame ICT werken en het OpenIMS DMS of Google Docs dagelijks gebruiken, is een realistisch gemiddelde geschat. Als het om klanten gaat is deze test niet realistisch te staven, omdat niet bekend is hoeveel klanten het systeem gebruiken.

profiel schets systeemgebruik van 1 dag bij OpenSesame ICT:

Aantal actieve gebruikers	Tijdstip van de dag	Activiteit	Minimaal gebruik	Gemiddeld gebruik	Intensief gebruik
3 - 6	08.00 – 09.00	Inloggen	X		
8	09.00 – 11.00	Intensief gebruik			X
5	11.00 – 12.00	Normaal gebruik		X	
1	12.00 – 13.00	Lunch	X		
8	13.00 – 16.00	Intensief gebruik			X
5	16.00 – 17.00	Normaal gebruik		X	
3 - 6	17.00 – 17.30	Uitloggen	X		

Niet alle personen zullen tegelijk uitloggen. Vandaar hier een gemiddelde is genomen van 3 tot 6 personen als gemiddelde van personen die tegelijkertijd uit zouden loggen.

2) specificeren testgevallen

In deze stap is het nader detailleren van de te testen onderdelen van systeemgebruik en het specificeren van de testgevallen voor de real-life test. Ook voor dit geldt dat de werkelijke situatie zoveel mogelijk nagebootst dient te worden.

Voor het nader detailleren van de te testen onderdelen van systeemgebruik dienen de volgende activiteiten te worden uitgevoerd:

Identificeren gebruikersgroepen

- Project en Organisatie
- Ontwikkelaars
- Administratief medewerkers

Overzicht taken per gebruikersgroep

Project en Organisatie – Het managen van projecten, relaties onderhouden met klanten. Het verkopen van producten en diensten. Opdrachten binnenhalen voor de ontwikkelaars, rapporten opstellen, nieuwe producten presenteren.

Ontwikkelaars – Software ontwikkelen, het schrijven van documentatie, bijhouden van urenregistratie, melden van incidenten.

Administratief medewerkers – Ondersteunen voorbereiden en organiseren van het werk van de directie, het beantwoorden van vragen van de baas over organisatorische punten, het bijhouden van de afsprakenagenda, het bijhouden van correspondentie of in- en uitgaande postmappen, het uittypen van brieven, verslagen, offertes, het beantwoorden van telefoon, mail en fax, het archiveren van stukken, het contracteren van klanten, het notuleren van gesprekken of vergaderingen en het organiseren van bijeenkomsten, congressen e.d.

Bepaal de frequentie per taak per tijdseenheid (alleen relevante taken voor deze test)

Het betreft hier niet het nauwkeurig bepalen van de frequentie, het schatten hoe vaak een taak per tijdseenheid wordt uitgevoerd door welke gebruikersgroep. Normaal gesproken dienen gegevens verkregen te worden uit voorgaande systemen met vergelijkbare functionaliteit of door in het huidige systeem bij te houden hoe vaak een bepaalde taak wordt uitgevoerd. Gezien de privacy belangen van OpenSesame ICT was het niet mogelijk deze gegevens te verkrijgen. Dit zou gemonitord of gelogd moeten worden. Daarom is ervoor gekozen om een zo goed mogelijke en reële schatting te maken.

Het bepalen de relatieve frequentie van OpenSesame ICT

De relatieve frequentie is het aantal keer dat een bepaalde taak in een bepaalde tijdseenheid wordt uitgevoerd in verhouding tot het aantal keer dat alle taken in die tijdseenheid worden uitgevoerd, uitgedrukt in een percentage.

3) Testuitvoering

Omdat het om relatief weinig aantallen betreft, zijn de berekeningen niet met een testtool gedaan. Deze tests zijn gedaan op reële schattingen en niet op basis van langdurige monitoring of het loggen van gegevens. De taken worden in dit geval per dag berekend.

praktijkgebruik:

Relatieve frequentie op basis van intensief gebruik:

Taken	Frequentie/Per dag	Relatieve frequentie
Aanmaken documenten	50	0,33
Wijzigen documenten	70	0,46
Verwijderen documenten	30	0,20
Totaal taken	150	1,0

De testgevallen worden gemaakt met in overeenstemming met de taken en de relatieve frequentie. In dit geval wordt uitgegaan van intensief gebruik van het systeem met 8 mensen. Dit betekent dat de relatieve frequentie van "Aanmaken documenten" 0,33 is. Dit betekent dus dat $(0,33 \times 100) = 33\%$ van de testgevallen betrekking hebben op het aanmaken van nieuwe documenten, 46% betrekking hebben op het wijzigen van documenten, en 20% betrekking heeft op het verwijderen van documenten.

Relatieve frequentie op basis van gemiddeld gebruik:

Taken	Frequentie/ Per dag	Relatieve frequentie
Aanmaken documenten	25	0,27
Wijzigen documenten	45	0,50
Verwijderen documenten	20	0,22
Totaal taken	90	1,0

In dit geval wordt uitgegaan van gemiddeld gebruik van het systeem met 5 mensen. Dit betekent dat de relatieve frequentie van "Aanmaken documenten" 0,27 is. Dit betekent dus dat $(0,27 \times 100) = 27\%$ van de testgevallen betrekking hebben op het aanmaken van nieuwe documenten, 50% betrekking hebben op het wijzigen van documenten, en 22% betrekking heeft op het verwijderen van documenten.

Relatieve frequentie op basis van minimaal gebruik:

Taken	Frequentie/Per dag	Relatieve frequentie
Aanmaken documenten	10	0,22
Wijzigen documenten	20	0,44
Verwijderen documenten	15	0,33
Totaal taken	45	1,0

In dit geval wordt uitgegaan van laag gebruik van het systeem met 3 mensen. Dit betekent dat de relatieve frequentie van "Aanmaken documenten" 0,33 is. Dit betekent dus dat $(0,22 \times 100)$ 22% van de testgevallen betrekking hebben op het aanmaken van nieuwe documenten, 44% betrekking hebben op het wijzigen van documenten, en 33% betrekking heeft op het verwijderen van documenten.

4) Resultaat

Op basis van deze resultaten is het ook mogelijk om berekeningen te doen met andere taken en aantallen. Je kan per uur, per maand of per jaar berekeningen maken. Dit vereist dan wel een goede monitoring en het loggen van de activiteiten voor een bepaald tijdsduur van de OpenIMS server. Daarbij komt het privacy aspect van de werknemers kijken, waardoor deze test niet altijd toepasbaar is. Bij OpenSesame ICT is het loggen en monitoren van deze gegevens niet toegestaan. Enkel als het gegevens van jezelf betreft.

Performance:

Voor het testen van de performance kan in OpenIMS eenvoudig gebruik worden gemaakt van de codetester. De codetester berekend namelijk automatisch het aantal milliseconden van het verwerken van data tijdens het synchroniseren. Zie figuur 3 voor een afbeelding van de codetester.

Hieronder zijn een aantal performancetests gedaan m.b.v. de codetester. Dit betreft het aanmaken, wijzigen en verwijderen van documenten van zowel OpenIMS en Google Docs. De responsetijd zal d.m.v. afbeeldingen aangetoond worden.

“Aanmaken van een nieuw document in OpenIMS”

--- Voor ---

N_Log: google: Sync started - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

N_Log: google_actions: 3: COPY NEW OpenIMS->Google Testdoc.doc

N_Log: google: Sync completed - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

--- Na ---

Benodigde tijd: **2570.1 ms**

Figuur 4 toont de tijd in ms met het aanmaken van een nieuw document in OpenIMS

“Wijzigen van een document in OpenIMS”

--- Voor ---

N_Log: google: Sync started - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

N_Log: google_actions: 5: COPY CHANGED OpenIMS->Google Testdoc.doc

N_Log: google: Sync completed - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

--- Na ---

Benodigde tijd: **3305.9 ms**

Figuur 5 toont de tijd in ms met het wijzigen van een document in OpenIMS

“Verwijderen van een document in OpenIMS”

--- Voor ---

N_Log: google: Sync started - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

N_Log: google_actions: 2: PROPAGATE DELETE OpenIMS->Google # filename: Testdoc.doc

N_Log: google: Sync completed - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

--- Na ---

Benodigde tijd: **1316.7 ms**

Figuur 6 toont de tijd in ms met het verwijderen van een document in OpenIMS

“Aanmaken van een Google document”

--- Voor ---

N_Log: google: Sync started - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

N_Log: google_actions: 4: COPY NEW Google->OpenIMS Google.doc

N_Log: google: Sync completed - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

--- Na ---

Benodigde tijd: **2559.3 ms**

Figuur 7 toont de tijd in ms met het aanmaken van een nieuw document in Google Docs

“Wijzigen van een Google document”

--- Voor ---

N_Log: google: Sync started - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

N_Log: google_actions: 6: COPY CHANGED Google->OpenIMS Googledoc.doc

N_Log: google: Sync completed - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

--- Na ---

Benodigde tijd: **2701.2 ms**

Figuur 8 toont de tijd in ms met het wijzigen van een document in Google Docs

“Verwijderen van een Google document”

--- Voor ---

N_Log: google: Sync started - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

N_Log: google_actions: 1: PROPAGATE DELETE Google->OpenIMS Googledoc.doc

N_Log: google: Sync completed - Google User: ict.opensesame # danny_sites # 184135102b30eade1730b2e8c2209a98

--- Na ---

Benodigde tijd: **692.0 ms**

Figuur 9 toont de tijd in ms met het verwijderen van een document in Google Docs

Overzicht performancetest:

OpenIMS		Google Docs	
Actie	Aantal ms	Actie	Aantal ms
Aanmaken document	2570.1ms	Aanmaken document	2559.3ms
Wijzigen document	3305.9ms	Wijzigen document	2701.2ms
Verwijderen document	1316.7ms	Verwijderen document	692.0ms

4) Resultaat, Conclusie

In de overzichtstabel zien we dat OpenIMS op twee punten trager is als Google Docs. Als het gaat om het wijzigen en verwijderen van documenten is Google sneller. Bij het aanmaken van een document zit een miniem verschil tussen de beide systemen. OpenIMS is daar een fractie sneller in als Google.

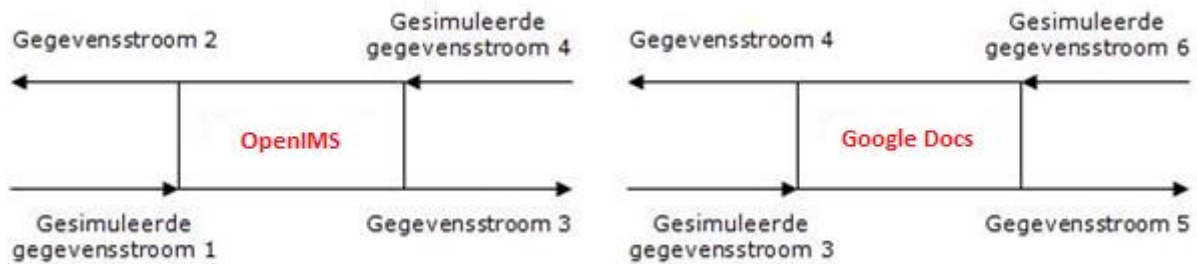
Deze eenmalige test wil uiteraard niet zeggen dat deze waardes altijd zo zijn. Het kan voorkomen dat de ene keer de server van OpenIMS sneller is als die van Google. Andersom kan dit ook het geval zijn. Dit kan verschillende redenen hebben zoals drukke en rustige tijdstippen (pieken en dalen) van gebruikers.

Testontwerptechniek: Programma Interfacetest

Algemeen

De programma interfacetest test de interfaces tussen programma's. Tijdens deze test worden een aantal programma's geïntegreerd tot één programma. Als blijkt dat dit programma goed werkt dan is aangetoond dat de interface tussen de programma's afzonderlijk ook goed werkt. Voordat er met een programma interfacetest gestart kan worden, moeten alle 'te integreren' programma's onafhankelijk van elkaar goed kunnen functioneren.

Globaal werkt de programma interfacetest als volgt:



Figuur 10 toont de gegevensstroom tussen beide programma's

Bovenstaande programma's OpenIMS en Google Docs zijn, aangestuurd d.m.v. gesimuleerde gegevensstromen, er vanuit gaande dat alle functionaliteiten van beide programma's OpenIMS en Google Docs los van elkaar goed werken. Tijdens de programma interfacetest worden OpenIMS en Google Docs geïntegreerd tot één programma waarbij de gesimuleerde invoerstroom 3 voor Google Docs vervangen wordt door de uitvoerstroom 3 van OpenIMS en de gesimuleerde invoerstroom 4 van OpenIMS vervangen wordt door de uitvoerstroom 4 van Google Docs. Het kan uiteraard ook zo zijn dat alleen stroom 3 of 4 aanwezig is i.p.v. allebei (zie figuur 10).



Figuur 11 toont de gegevensstroom als één werkend programma

Er wordt vervolgens nagegaan of OpenIMS en Google Docs, werkend als één programma, correct functioneert. De invoer vanuit andere programma's wordt nog steeds gesimuleerd (zie figuur 11 stroom 1 en 6) en de uitvoer naar een eventueel derde programma wordt nog steeds kunstmatig afgetapt (zie figuur 11 stroom 2 en 5). Als aangetoond is dat OpenIMS en Google Docs samengevoegd tot één programma goed kan functioneren, dan zouden er eventueel nog meer programma's aan toegevoegd kunnen worden. In dit geval is dit niet van belang en volstaat het dat OpenIMS met Google Docs goed met elkaar overweg kan als zijne één werkend programma.

Werkwijze

Een aantal stappen die ondernomen moeten worden m.b.t het gebruik van de programma interfacetest zijn de volgende:

- Identificeren gegevensstromen (testsituaties)
- Bepalen testgevallen
- Vaststellen initiële gegevensverzameling
- Opstellen testscript
- Testuitvoering
- Resultaat, Conclusie

1) Identificeren gegevensstromen (testsituaties)

Tijdens deze stap moeten de gegevensstromen (zowel de directe als indirecte) die bij de integratie betrokken zijn worden geïdentificeerd.

De volgende gegevensstromen (zowel direct als indirect) zijn geïdentificeerd	
1	Authenticeren bij Google
2	Verkrijgen laatste folder-state-map van de Google-repository folder
3	Verkrijgen laatste folder-state-map van de OpenIMS-DMS folder
4	Verkrijgen laatste folder-state-map van de OpenIMS-DMS folder
5	Verkrijgen huidige folder-state-map van de OpenIMS-DMS folder
6	Synchroniseer verwijderde Google-repository documenten
7	Synchroniseer verwijderde OpenIMS documenten
8	Synchroniseer nieuwe OpenIMS documenten
9	Synchroniseer nieuwe Google-repository documenten
10	Synchroniseer gewijzigde OpenIMS documenten
11	Synchroniseer gewijzigde Google-repository documenten
12	Verkrijgen huidige folder-state-map van de Google-repository folder
13	Opslaan huidige folder-state-map van de Google-repository folder
14	Opslaan huidige folder-state-map van de Google-repository folder
15	Opslaan huidige folder-state-map van de OpenIMS-DMS folder

2) Vaststellen initiële gegevensverzameling

Om fysieke testgevallen, dus acties op het zendende programma en de controles op het ontvangende programma te kunnen uitvoeren, zijn bepaalde gegevens nodig. De bedoeling is vanwege beheer en hergebruik argumenten vooraf eenmalig te definiëren. Omdat het bij OpenIMS en Google Docs gaat om bestand of document overdracht gaat, en niet om een programma waarbij veel invoerwaardes van toepassing zijn, is het niet nodig dit vooraf te definiëren. Het enige wat vooraf gedefinieerd kan worden is de bestandsnaam van een document dat nieuw aangemaakt word.

3) Bepalen testgevallen

De testgevallen zijn bepaald in de teststrategie van het testplan. Dit betreft de volgende testgevallen:

Van OpenIMS

- Het aanmaken van nieuwe documenten

- Het wijzigen van documenten
- Het verwijderen van documenten
- Het ophalen van de OpenIMS directory

Van Google Docs

- Het aanmaken van nieuwe documenten
- Het wijzigen van documenten
- Het verwijderen van documenten
- Het ophalen van de Google directory

4) Opstellen testscript

In het testscript wordt vastgelegd in welke volgorde de fysieke testgevallen moeten worden uitgevoerd. Hierbij is optimale efficiëntie van optimaal belang. In dit geval is het alleen belangrijk dat de OpenIMS en Google directory als eerste opgehaald worden, de rest is niet van belang aangezien ze hetzelfde doel dienen. Het belangrijkste is dat eerst de directory's opgehaald worden zodat er überhaupt document aangemaakt, gewijzigd en verwijderd worden. In onderstaand tabel staat de volgorde aangegeven:

Volgorde van fysieke testgevallen
Het ophalen van de OpenIMS directory
Het ophalen van de Google directory
OpenIMS
Het aanmaken van nieuwe documenten
Het aanmaken van nieuwe documenten
Het verwijderen van documenten
Google Docs
Het aanmaken van nieuwe documenten
Het aanmaken van nieuwe documenten
Het verwijderen van documenten

5) Testuitvoering

Voor een fysiek testgeval kan een testuitvoerformulier opgesteld worden. Hierop staan de gegevens van het testgeval vermeld. In dit geval zullen de gegevens vermeld worden in dit testrapport.

7) Resultaat, Conclusie

Deze test is meer geschikt voor het toepassen op systemen waar invoerwaardes van toepassing zijn. Ook systemen die met databases werken, zijn hier meer voor geschikt wat betreft controle van in- en uitvoergegevens. Ondanks deze test niet uitermate geschikt was voor dit project, heb ik er toch van kunnen leren

Testontwerptechniek: Error Guessing

Doel:

Het doel van Error Guessing is het uitvoeren van aanvullende testen of het verkrijgen van een subjectief kwaliteitsbeeld.

Beschrijving:

Error Guessing is in wezen ongestructureerd testen, het gissen naar fouten. De waarde ervan ligt in het onverwachte: er worden tests uitgevoerd die anders niet aan bod komen.

Bij Error Guessing speelt de ervaring van de tester een belangrijke rol. De essentie van Error Guessing is, dat de tester op zoek gaat naar die gevallen die altijd moeilijkheden veroorzaken. De tester wordt hierbij niet gesteund of gehinderd door testscripts, checklist et cetera. Hij of zij bepaalt zelf welke testgevallen onderzocht gaan worden. Verder kan hij of zij testen wat relevant lijkt.

Voor de test wordt een maximale duur vastgesteld. Ook in geval van Error Guessing er is dus sprake van timeboxing.

Toepasbaarheid:

Error Guessing is zeer breed toepasbaar. Het is een waardevolle aanvulling op de gestructureerde testsoorten. De techniek Error Guessing wordt met name gebruikt bij het black-box testen en kan zich in principe richten op alle mogelijke kwaliteitsattributen.

De enige startvoorwaarde voor Error Guessing is, dat een goed begrip moet bestaan van het te testen systeem, maar ook enige mate van stabiliteit van het te testen systeem is gewenst. Error Guessing is dan ook bij uitstek geschikt voor de eindfase van het testproces.

Voorwaarden:

- De tester is bekend met de inhoud en vorm van het te toetsen object.
- De tester is bekend met de inhoud van de andere tests binnen het testtraject zodat dubbele tests voorkomen worden.

Rapportage:

Normaal gesproken worden de bevindingen van deze test vastgelegd in een zogeheten bevindingenformulier. Dit formulier wordt aan de test coördinator gegeven. Aangezien dit een afstudeerproject betreft, zal in dit geval de bevindingen in het testrapport zelf worden opgenomen.

Error Guessing bevindingen synchronisatie functionaliteit:

- 1) Als twee gebruikers in OpenIMS hetzelfde document wijzigen en vervolgens synchroniseren met Google Docs, dan gaat van één document de gegevens verloren. Diegene die het laatst doorkomt met synchroniseren wint.

2) Als een document in Google gewijzigd wordt en datzelfde document ook in OpenIMS gewijzigd wordt, dan krijgt het document van OpenIMS voorrang op dat van Google Docs. De synchronisatie heeft namelijk geen “file merging” functionaliteit.

4) Nieuw aangemaakte documenten in OpenIMS zonder contenttype worden niet naar Google Docs overgeplaatst worden. Hier is een controle op gemaakt.

5) Als bij het aanmaken van een nieuw Google document geen contenttype wordt meegegeven in de bestandsnaam, dan is het document niet te openen in OpenIMS. OpenIMS kan alleen documenten en andere bestanden lezen met een extensie.

6) Bestanden met een andere extensie als .doc, .docx, .ppt, .odt, .pdf, worden niet overgezet om alleen deze extensies ondersteund worden.

Referenties

Nr.	Onderwerp	Bron
1	TMap	http://en.wikipedia.org/wiki/Test_Management_Approach
2	Testen algemeen	Boek: Testen volgens TMap 2 ^e druk

Afstudeerplan

Informatie afstudeerder en gastbedrijf *(structuur niet wijzigen)*

Afstudeerblok: 2011-1.2 (start uiterlijk 9 mei 2011)
Startdatum uitvoering afstudeeropdracht: 11-03-2011
Einddatum uitvoering afstudeeropdracht: 08-07-2011
Inleverdatum afstudeerdossier volgens jaarrooster: 23 september 2011

Studentnummer: 20057842
Achternaam: dhr. van Tol
Voorletters: D.P.J.
Roepnaam: Danny
Adres: Kamp 3074
Postcode: 8225 GP
Woonplaats: Lelystad
Telefoonnummer: -
Mobiel nummer: 06-10839026
Privé emailadres: danny@tendra.nl

Opleiding: Informatica
Locatie: Den Haag / Delft / Zoetermeer
Variant: voltijd

Naam studieloopbaanbegeleider: Rianne Bechet
Naam begeleider/examinator: -
Naam expert/examinator: -

Naam bedrijf: OpenSesame ICT
Afdeling bedrijf: Development
Bezoekadres bedrijf: Meerwal 13
Postcode bezoekadres: 3432 ZV
Postbusnummer: -
Postcode postbusnummer: -
Plaats: Nieuwegein
Telefoon bedrijf: 030 60 35 640
Telefax bedrijf: 030 60 32 946
Internetsite bedrijf: www.opensesame.nl

Achternaam opdrachtgever: dhr. Nico de Vries
Voorletters opdrachtgever: N.
Titulatuur opdrachtgever: drs. Ing.
Functie opdrachtgever: Directie
Doorkiesnummer opdrachtgever: -
Email opdrachtgever: nico.de.vries@opensesameict.com

Achternaam bedrijfsmentor: dhr. Eric Diks
Voorletters bedrijfsmentor: E.
Titulatuur bedrijfsmentor: drs.
Functie bedrijfsmentor: Programmeur
Doorkiesnummer bedrijfsmentor: -
Email bedrijfsmentor:

Doorkiesnummer afstudeerder: -
Functie afstudeerder (deeltijd/duaal): Student voltijd: Afstuderen

Titel afstudeeropdracht:

Onderzoek en ontwikkelen van een koppeling tussen OpenIMS en Google Docs

Opdrachtschrijving

1. Bedrijf

OpenSesame ICT B.V. is een Nederlands softwarebedrijf dat op basis van Open Source technologie bedrijfsmatige IT oplossingen biedt.

Het is een klein bedrijf met circa 30 mensen in dienst. De meeste van hen zijn voornamelijk bezig met software ontwikkeling. OpenSesame ICT is een bedrijf met een platte hiërarchie. Dat wil zeggen dat iedereen kan meedenken, en wat te vertellen heeft over de producten (software) dat ontwikkeld wordt. Ze vinden het belangrijk bij opensesame dat de afstudeerder zoveel mogelijk meedenkt en zijn ideeën geeft bij het uitvoeren van de opdracht.

Zij ontwikkelen en implementeren een product genaamd OpenIMS. OpenIMS is een op open source technologie gebaseerd webbased platform voor het beheren van ongestructureerde informatie (stromen). Denk hierbij aan werkprocessen of projecten waarin grote hoeveelheden documenten, webpagina's en e-mail berichten worden geproduceerd en uitgewisseld. Deze documenten staan vaak op fileservers, lokale harde schijven en in e-mail berichten. Met OpenIMS kan deze belangrijke informatie gestructureerd organiseren en beheren worden waardoor altijd over de juiste en actuele informatie kan worden beschikt. OpenIMS is gebaseerd op PHP (LAMP) en bevat ook enige Java, Delphi en VB.NET voor integratietoepassingen.

2. Probleemstelling

Google levert middels Google Docs een omgeving waarin online in the cloud documenten kunnen worden beheerd met een eenvoudige online tekstverwerker en spreadsheet. Typische Document Management System functionaliteit als metadata en workflow is in Google Docs nauwelijks beschikbaar. Insteek van het project is om de (on)mogelijkheden van interfacing met Google Docs te bepalen en vervolgens een koppeling tussen OpenIMS DMS en Google Docs te realiseren. Functioneel kan hierbij gedacht worden aan tweeweg synchronisatie van bestanden en het over en weer beschikbaar stellen van functionaliteiten.

OpenSesame ICT heeft een dms in het OpenIMS pakket. Dit dms heeft echter zijn beperkingen. In het dms kunnen bestanden geupload en bewerkt worden. Dit kan echter maar door één persoon tegelijk (één kant op). Er is geen functie beschikbaar zoals in Google docs, waarbij meerdere mensen in staat zijn om één document te bewerken/wijzigen, zonder het risico dat er informatie verloren kan gaan.

OpenSesame ICT wil daarom een uitbreiding van het dms waarbij er een koppeling komt, tussen OpenIMS en Google docs, zodat men ook in OpenIMS in staat is om deze functionaliteit te gebruiken. Tot op heden is binnen het bedrijf nog niets gedaan met een koppeling van het DMS en Google docs.

3. Doelstelling van de afstudeeropdracht

De doelstelling aan het einde van de afstudeerperiode is dan ook om het dms van OpenIMS gekoppeld te hebben aan Google docs, zodat de werknemers binnen het bedrijf in staat zijn om met meerdere personen documenten kunnen beheren/ aanpassen, zonder zich zorgen te maken of documenten overschreven worden door anderen, en hierdoor belangrijke informatie verloren gaat.

4. Resultaat

Het resultaat is, dat men binnen het bedrijf veel meer gebruik zal gaan maken van het dms om documenten gezamenlijk te beheren en aan te passen. Mede doordat deze functionaliteit(en) nog missen is het dms van OpenIMS een opslag media waar nu minder mee gedaan wordt als men eigenlijk zou willen.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

Plan van aanpak schrijven:

Hierin zullen verschillende onderdelen in beschreven worden als: literatuuronderzoek, het vaststellen van requirements (MoSCoW), beschrijving knelpunten in de huidige situatie, beschikbare documentatie en software, dat gebruikt zal worden. Evaluatiegesprekken (doel, met wie?) (Tijdplanning: 10 dagen).

Onderzoeken:

- Onderzoek doen naar de bestaande functionaliteit van Google Docs. Hoe het werkt en hoe het is opgebouwd.
- Onderzoek welke API's er beschikbaar zijn om de koppeling tussen OpenIMS en Google docs te realiseren.
- Onderzoek bestaande huidige functionaliteit en technologie OpenIMS DMS. Bestuderen hoe de code is opgebouwd en wat het doet. (Tijdplanning: 15 dagen).

* In deze fase zullen de requirements opgesteld worden, welke in het PvA worden opgenomen.

Ontwerp, bouw (product ontwikkelen)

Na onderzoek wordt er bepaalt welke (on)mogelijkheden er zijn voor implementatie. Hiervoor zal in overleg een ontwerp gemaakt worden (UML) waarin de koppelingen mogelijkheden uiteen gezet zullen worden. Bij het ontwerp zal tevens een advies worden gegeven als ook eigen ideeën alvorens de implementatie zal plaatsvinden. Voor het ontwikkelen van het product zal gedurende de stage gebruik worden gemaakt van Php, en JavaScript. (Tijdplanning: 35 dagen).

Testen

Nadat het ontwerp- ontwikkel proces zal er getest worden of de basis functionaliteit van het systeem naar behoren werkt. Voor dit testproces wordt (nog niet definitief) testgoal gebruikt. (Tijdplanning: 9 dagen).

Overdracht

Na het testen en goedkeuring van de opdrachtgever(s), zal een overdracht plaatsvinden van alle op te leveren producten aan het bedrijf. Dit zijn o.a. de technische documentatie en de code die van belang zijn voor het bedrijf. (Tijdplanning: 2 dagen).

Afstudeer dossier in orde maken

Alle benodigde documentatie voor school in orde maken. Gedurende de afstudeerperiode zal een logboek bijgehouden worden om alle informatie te verantwoorden in het verslag. (Tijdplanning (14 dagen).

6. Op te leveren (tussen)producten

- Onderzoeksrapport OpenIMS DMS / Google Docs
- Code (koppeling OpenIMS DMS / Google Docs)
- Technische documentatie (incl. Handleiding)
- Testrapport

* In de afstudeerperiode zullen tussentijdse evaluatiegesprekken zijn. In deze gesprekken wordt aangetoond aan de begeleiders hoe het afstuderen verloopt, hoe het gaat met het product en de verslaglegging (zowel

school als bedrijf). Als het nodig is kunnen er ook andere dingen besproken worden die op dat moment van toepassing zijn voor het afstuderen.

7. Te demonstreren competenties en wijze waarop

Analyseren en adviseren:

1.1 Selecteren, methoden, technieken en tools (niv 3)

Voor het uitvoeren van de afstudeeropdracht zal gebruik worden gemaakt van methoden, technieken en tools. Gedurende het afstudeertraject zal gebruik worden gemaakt van DSDM. Hieronder vallen o.a. de DSDM fasen zelf, en de MoSCoW methode voor het opstellen van de requirements. Bij de must have requirements zal gebruik worden gemaakt van timeboxing voor de tijdige oplevering en realisatie van het product.

Daarnaast zijn er een aantal vastgestelde technieken/ talen die gebruikt gaan worden tijdens het afstuderen. Dit is voor het overgrote deel met: Php en JavaScript.

Software ontwerp

3.2 Ontwerpen systeemdeel (niv 4)

Alvorens ontwikkeld kan worden aan het product, dient een ontwerp te worden gemaakt. Het is van belang dat bij het ontwerp ook eigen ideeën (inbreng) in het ontwerp wordt meegenomen. In overleg met de begeleiders kunnen zij een goedkeuring geven om het ontwerp uit te werken in praktijkvorm. Hier dient tevens rekening gehouden te worden met de bestaande software architectuur.

3.3 Bouwen applicatie (niv 4)

Dit betreft het ontwikkelen van een koppeling tussen OpenIMS en Google Docs. Hierbij zullen eigen ideeën een bepalende rol meespelen. In overleg kunnen mijn ideeën meegenomen voor het ontwerp en bouw van nieuwe functionaliteit(en). Hiervoor zullen eerst de mogelijkheden onderzocht moeten worden. Er zal bij het ontwikkelproces ook gebruik worden gemaakt van design patterns en de Google Docs API.

3.4 Initiëren en plannen van het testproces (niv 3)

Om te testen of het gemaakte product aan de verwachting van de opdrachtgever(s) voldoet. Hoe en met welke methodiek getest wordt, wordt nog nader bepaald in het plan van aanpak.

3.5 Uitvoeren van en rapporteren over het testproces (niv 3)

Bij het uitvoeren van tests zullen deze vastgelegd worden in een testrapport. Het testrapport wordt opgeleverd bij de overdracht van alle producten aan het einde van de afstudeerperiode.

Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: Danny van Tol

Periode: 11-03-2011 t/m 15-07-2011

Bedrijf c.q. instelling: OpenSesameICT

Bedrijfsmentor: Eric Diks

Plaats: Nieuwegein

Datum: 16 september 2011

1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeer-opdracht?

Ja.

2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?

Goed.

3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

Allemaal goed (in bold).

- | | |
|---|---|
| • Qua verantwoordelijkheid | goed / voldoende / matig / onvoldoende |
| • Qua zelfstandigheid | goed / voldoende / matig / onvoldoende |
| • Qua planmatig werken | goed / voldoende / matig / onvoldoende |
| • Qua creativiteit | goed / voldoende / matig / onvoldoende |
| • Qua productiviteit | goed / voldoende / matig / onvoldoende |
| • Qua samenwerken met collega's | goed / voldoende / matig / onvoldoende |
| • Qua draagvlakontwikkeling | goed / voldoende / matig / onvoldoende |
| • Qua inspelen op bedrijfscultuur | goed / voldoende / matig / onvoldoende |
| • Qua rekening houden met de specifieke context van het bedrijf | goed / voldoende / matig / onvoldoende |
| • Qua het op gang brengen van de nodige veranderingen | goed / voldoende / matig / onvoldoende |

4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?

Goed.

5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?

Goed.

6. Bent u tevreden over het opgeleverde (eind)product?

- **In hoeverre heeft u gekregen wat is afgesproken?**

Volledig.

- **In hoeverre voldoet het (eind)product aan uw verwachtingen?**

Volledig.

- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**

Goed en Goed.

- **Wat gebeurt er met het opgeleverde (eind)product?**

Komt in product bij voldoende klant belangstelling.

- **Kunt u direct met het opgeleverde product aan de slag?**

Bijna, verdere finetuning zal noodzakelijk zijn, ook door verdere ontwikkeling bij Google.

7. Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?

Nee.

8. Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats (graag met toelichting)?

Zeker, wij doen dit al jaren, tot tevredenheid van zowel de student als OpenSesame.

Een nuttige en plezierige kruisbestuiving van kennis en vaardigheden!

Bespreking concept	Tussentijds assessment	Eerste beoordeling
--------------------	-------------------------------	--------------------

Formulier tussentijds assessment

Student: Danny van Tol

Studentnummer: 20057842

Datum: 12-09-2011

eerste / tweede TTA:

Tijdens het tussentijds assessment is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	Ja	
b	Het afstudeerdossier is beschikbaar op Blackboard	Ja	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen	Ja	
d	Het goedgekeurde afstudeerplan is aanwezig		nee
e	Het plan van aanpak is aanwezig	Ja	
f	Reeds geleverd commentaar is aanwezig	Ja	
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken	Ja	
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	ja	

Beroepstaken op afgesproken niveau uitgevoerd?		O	V
1	1.1		V
2	3.2		V
3	3.3		V
4	3.4	O	
5	3.5 : nog niet verwerkt in het verslag, wel mondeling toegelicht	O	V
6			

7			
8			

Overige beoordelingscriteria

O V

Aanpak

V

Passend

	V
--	----------

Theoretisch verantwoord

--	--

Producten

Tussenproducten

	V
--	----------

Eindproduct

O	V
----------	----------

Reflectie

Inzicht in eigen functioneren: nog niet beschreven in verslag

	V
--	----------

Inzicht in eigen leerproces

	V
--	----------

Toelichting:

Advies:

Afstudeerdossier inleveren op 23-09-2011

Aankruisen welke beslissing de student heeft genomen

Afstudeerdossier wordt op afgesproken datum ingeleverd

Inleverdatum: 23-09-2011

De afstudeerperiode wordt met 30% verlengd

Nieuwe einddatum:

eerstvolgende inleverdatum volgens het jaarrooster

Inleverdatum:

Student stopt met de afstudeeropdracht

Naam begeleider/examinator: Okan Zor

Naam expert/examinator: Gerda In 't Veld

Datum: 13-09-2011

Dit formulier wordt door de expert/examinator digitaal ingevuld, waarna de begeleider/examinator het per email verstuurt naar de student met een cc naar de coördinator van ICT & Media @ Work (A.M.Schipper@hhs.nl). Het formulier dient door de student te worden opgenomen in het afstudeerdossier.