

Bijlage G: Architectuur beschrijving

Exact Cloud Solutions

Architectuur beschrijving

Project External Documentmanagement SaaS' Integration

Opdrachtgever: Ronald Dähne – Director Technology Cloud Solutions
Begeleider: Ad van der Hoeven – Senior Research Engineer
Datum: 21 maart 2013
Versie: 2.0

= exact

And it all comes together.

Google

Microsoft®
V.NET

Auteur

E.M. Pekel
Student – 20064646
Informatica
Haagse Hogeschool – Den Haag

Begeleidend examiner: G.A. Mijharends
Tweede examiner: Dhr. B. Kuiper

Inleiding

In het kader van het afstudeerproject “External documentmanagement SaaS’ Integration” is deze architectuur opgesteld. De architectuurbeschrijving heeft als doel het in kaart brengen van de risico’s, het eenvoudig maken van uitbreidingen in de toekomst en maken van ontwerpbeslissingen voor de integratie van SaaS-oplossingen op het gebied van documentbeheer.

Dit rapport zal fungeren als basis voor de integratie van meerdere externe SaaS-oplossingen, mits de opdrachtgever na deze integratie wil doorgaan met het integreren van andere SaaS-oplossingen. Verschillende zaken die van essentieel belang zijn om een dergelijk systeem op te kunnen zetten worden behandeld. Algemeen gezegd zal dit rapport de architectuur van het systeem weergeven.

Inhoudsopgave

1.	Introductie en management	3
1.1	Korte introductie Exact Online	3
1.2	Doelstelling architectuurbeschrijving.....	3
1.3	Doelstelling systeem.....	3
1.4	Relevante bedrijfsprocessen	3
1.5	Positionering systeemdeel	3
1.6	Stakeholdersanalyse.....	4
1.7	Non-functionele requirements voor Google Integration	5
1.8	Concerns m.b.t. architectuur.....	5
2.	Scope	6
2.1	Activiteiten	6
3.	Functional Viewpoint	7
3.1	Concerns.....	7
3.2	Technische achtergrond Exact Online	7
3.3	Lagenmodel	7
3.4	Functionele capaciteiten	11
3.5	Perspectives.....	12
4.	Information Viewpoint	16
4.1	Concerns.....	16
4.2	Perspectives.....	16
4.3	Inrichting relatieboom bij Google Drive	19
5.	Development Viewpoint.....	21
5.1	Concerns.....	21
5.2	Libraries versus eigen HTTPsRequest opstellen	21
5.3	Eisen aan library	23
5.4	Perspectives.....	24
5.5	Design patterns	25
6.	Key scenario's.....	30
	Bibliography.....	35
	Bijlage 1: OAuth.....	36

1. Introductie en management

1.1 Korte introductie Exact Online

Exact Online is een Cloud-oplossing voor de financiële administratie voor midden- en kleinbedrijf. Omdat Exact Online in de Cloud staat, hebben klanten overal waar zij internettoegang hebben ook toegang tot hun financiële administratie.

Behalve de financiële administratie zijn er ook specifieke modules beschikbaar, bijvoorbeeld de samenstelling voor handel(logistiek, voorraadbeheer, CRM en boekhouding) en de samenstelling voor zakelijke dienstverlening(urenregistratie, facturatie, CRM en boekhouding).

Verder is Exact Online een multi-tenant applicatie, wat betekent dat een infrastructuur wordt gebruikt om meerdere klanten te bedienen.

1.2 Doelstelling architectuurbeschrijving

- Het samen brengen en documenteren van de business requirements, software requirements en technische requirements op een zodanige manier dat het voor iedereen begrijpelijk is.
- Adviseren en uiteenzetten van de meest gangbare architectuur voor de integratie tussen Exact Online en de externe SaaS-oplossingen.
- De opdrachtgever Ronald Dähne moet op basis van de architectuurbeschrijving een beslissing kunnen maken over het wel of niet doorgaan met het integreren van SaaS-oplossingen op het gebied van documentbeheer.
- Beschrijven van de meest gangbare architectuur en mogelijke alternatieven.
- Extern bedrijf Comsec moet op basis van de architectuur beschrijving een oordeel kunnen geven over de dekking van de beveiligingsrisico's

1.3 Doelstelling systeem

De belangrijkste functionaliteiten van het nieuwe systeemdeel zijn:

- het laten zien welke documenten in bijvoorbeeld Google Drive zijn gekoppeld aan een bepaalde klant of een bepaald project;
- Het kunnen inzien van die documenten.

Het positief verhogen van de userexperience is de kritieke succesfactor voor dit project.

1.4 Relevante bedrijfsprocessen

- Het inzien van de lijst documenten horend bij een bepaald project of bij een bepaalde klant/account
- Het inzien van de documentinhoud horend bij een bepaald project of bij een bepaalde klant/account
- Het aanmaken, aanpassen en verwijderen van documenten
- Het delen en versturen van documenten
- Het archiveren van documenten

1.5 Positionering systeemdeel

Het nieuwe systeemdeel wordt onderdeel van Exact Online en wordt beschikbaar voor gebruikers met modules Project en CRM in hun licentie. De interface krijgt dezelfde look and feel als de bestaande documentenfunctionaliteiten binnen Exact Online en/of van de externe SaaS.

Het nieuwe systeemdeel moet interacteren met externe systemen van andere bedrijven. Dit brengt een aantal beveiligingsrisico's met zich mee die afgevangen moeten worden en performance vraagstukken die tijdens het ontwerp moeten worden beantwoord. Verder zullen er statistieken, activity logs en error logs worden bijgehouden om het nieuwe systeemdeel te monitoren.

Als technische beperking geldt dat het nieuwe systeemdeel gebruik moet maken van de huidige SQL Server database.

1.6 Stakeholdersanalyse

Stakeholder	Belangen
Opdrachtgever	De nieuwe module zorgt voor een nieuw hoofdstuk in het cloud-tijdperk. Als deze module een succes wordt, opent dat deuren naar andere cloud-oplossingen. Dit is een project dat al een tijdje op stapel lag. De opdrachtgever wil de resultaten van dit project gebruiken om een betere beslissing te kunnen nemen over: gaan we in de toekomst vaker gebruik maken van andere cloud oplossingen om ons eigen product te versterken. Belangrijke punten zijn: usability, security, risico-analyse, eenmalige aanmelding
Project Team	Het kunnen linken met bijvoorbeeld Google Drive zorgt er voor dat het beheren van documenten bij projecten makkelijker wordt. Klanten hoeven niet meer documenten te uploaden en bij wijzigingen de nieuwe versie te uploaden. De documenten worden met de koppeling live beschikbaar bij een project. Hierdoor wordt de meerwaarde van het projectgebied in Exact Online vergroot. Er zijn mogelijk verschillende manieren om de relatiestructuur in te richten. Dit moet onderzocht worden.
Hosting Taskforce	Hoe het nieuwe systeemdeel wordt gebouwd, bepaald hoe groot de serverbelasting en de laadtijd van een pagina zal zijn. Zaak is om de serverbelasting zo laag mogelijk te houden. Het is van belang om activiteiten en errors te loggen en statistieken bij te houden. Op lange termijn zou de integratie een kostenbesparing kunnen opleveren voor Exact omdat er minder servers nodig zijn voor de opslag van de documenten. Belangrijke punten zijn: performance, security, monitoring, logging
System Team	Goed werkende module die eenvoudig te onderhouden en te beheren is. Hiervoor moeten de ICT-standaarden en richtlijnen worden gevolgd en hetzelfde framework worden gebruikt. Het eenvoudig kunnen toevoegen van andere SaaS-oplossingen met alle overeenkomende functies.

1.7 Non-functionele requirements voor Google Integration

Hoofdeigenschap	Kwaliteitseigenschap	Requirement
Efficiency	Snelheid	Gepaste reactie-, verwerkings- en doorgeeftijden bij bepaalde volumes. De laadtijd van een pagina mag door de integratie module met hooguit 90 milliseconden worden vertraagd.
Betrouwbaarheid	Bedrijfszekerheid	Serverbelasting minimaal houden en server down halen voorkomen. De serverbelasting mag met maximaal 10% toenemen door de integratie module.
Functionaliteit	Beveiligbaarheid	Voor uploaden controle op (kwaad kunnende) scripts.
Functionaliteit	Beveiligbaarheid	Beschermen van de gegevens bij uitwisseling met externe server.
Onderhoudbaarheid	Analyseerbaarheid	Het loggen van activiteiten en fouten in logs.
Onderhoudbaarheid	Testbaarheid	Het kunnen valideren van de beoogde werking van het nieuwe systeemdeel bij wijzigingen binnen dat systeemdeel.
Bruikbaarheid	Leerbaarheid	Binnen een uur te gebruiken door zelfde look and feel als de huidige oplossing.
Bruikbaarheid	Gebruiksgemak	Bij connectieproblemen met google foutmelding weergeven en verder gaan met de rest van de pagina laden.

1.8 Concerns m.b.t. architectuur

1. De gegevensuitwisseling tussen Exact Online en externe SaaS-oplossingen is beveiligd en voldoet aan de huidige beveiligingseisen die worden gehanteerd binnen Exact.
2. De laadtijd van een pagina wordt met hooguit 90 milliseconden vertraagd bij het ophalen van de documentenlijst.
3. Bij het uploaden naar de Exact Server wordt een controle gedaan op kwaad kunnende bestanden.
4. Activiteiten en foutmeldingen worden vastgelegd.
5. De beschikbaarheid van Exact Online wordt niet beïnvloed door de integratie.
6. Het nieuwe systeemdeel heeft dezelfde leercurve als het huidige documentengedeelte.
7. Het nieuwe systeemdeel is eenvoudig te onderhouden en uit te breiden
8. De rechten tot mappen en bestanden zoals ze binnen de externe SaaS-oplossing zijn ingericht, zijn binnen Exact Online hetzelfde ingericht.
9. Het systeem heeft de functionaliteiten zoals genoemd in het document "Requirements opstellen"

2. Scope

De architectuurbeschrijving moet voldoende informatie geven, zodat de opdrachtgever een beslissing kan nemen over het wel of niet verder gaan met het integreren van SaaS-oplossingen zoals Google Drive. Verder zal er bij het opstellen van de architectuurbeschrijving ook rekening worden gehouden met de concerns van de verschillende stakeholders. De huidige architectuur van Exact Online, die geen directe connectie heeft met het nieuwe systeemdeel, valt buiten de scope en wordt alleen globaal weergegeven. De focus van de beschrijving ligt op de integratie met externe SaaS-oplossingen zoals Google Drive. Hieronder een lijst met activiteiten die zullen leiden tot de architectuurbeschrijving.

2.1 Activiteiten

- Doel architectuurbeschrijving specificeren
- Scope architectuurbeschrijving bepalen
- Opnemen van requirements in architectuurbeschrijving
- Het opnemen van de concerns van de stakeholders in architectuurbeschrijving
- Bepalen viewpoints voor architectuurbeschrijving
- Het opstellen van het functional viewpoint
- Het opstellen van het information viewpoint
- Het opstellen van het development viewpoint
- Onderzoek OAuth
 - Wat is OAuth?
 - Wat zijn de verschillen tussen OAuth en OAuth2.0?
- Onderzoek asynchronous pattern
 - Wat is het asynchronous pattern?
 - Waarom toepasselijk in dit project?
 - Wat houdt het in voor dit project?
- Mappen en bestanden structuur bij externe SaaS-oplossing schetsen m.b.v. relatieboom
- Korte beschrijving factory pattern

3. Functional Viewpoint

Het functioneel viewpoint definieert de architectonische elementen die zorgen voor de functionaliteiten van het nieuwe systeemdeel. In dit viewpoint worden het nieuwe systeemdeel functionaliteiten en structuur gedocumenteerd inclusief de belangrijkste functionele elementen, hun verantwoordelijkheden en hun interacties onderling. Alles tezamen genomen zorgt deze view voor een abstract beeld over het nieuwe systeemdeel en welke functies deze dient te bevatten.

3.1 Concerns

De concerns die in dit viewpoint worden behandeld, zijn:

1. De gegevensuitwisseling tussen Exact Online en externe SaaS-oplossingen is beveiligd en voldoet aan de huidige beveiligingseisen die worden gehanteerd binnen Exact.
2. De laadtijd van een pagina wordt met hooguit 90 milliseconden vertraagd bij het ophalen van de documentenlijst.
3. De beschikbaarheid van Exact Online wordt niet beïnvloed door de integratie.
4. Het systeem heeft de functionaliteiten zoals genoemd in het document "Requirements opstellen"

3.2 Technische achtergrond Exact Online

Technische gezien is Exact Online volledig op basis van het Microsoft ASP.NET (Active Server Pages .NET) framework, waarbij gebruik wordt gemaakt van Visual Basic.NET (VB.NET) als server-side programmeertaal. Aan de client-side worden verschillende Javascript libraries gebruikt om de User Interface te manipuleren, denk hierbij aan AJAX¹ en jQuery². Voor de opslag van de data wordt een Microsoft SQL Server gebruikt. Verder ondersteunt Exact Online het gebruik van verschillende browser, te weten Microsoft Internet Explorer, Mozilla Firefox en Apple Safari. De omvang van Exact Online code, als in het aantal regels code, wordt geschat op ongeveer drie miljoen regels broncode. (Göritzer, 2011)

3.3 Lagenmodel

In deze paragraaf worden het oude en nieuwe lagenmodel van de architectuur van Exact Online besproken.

3.3.1 Oude model

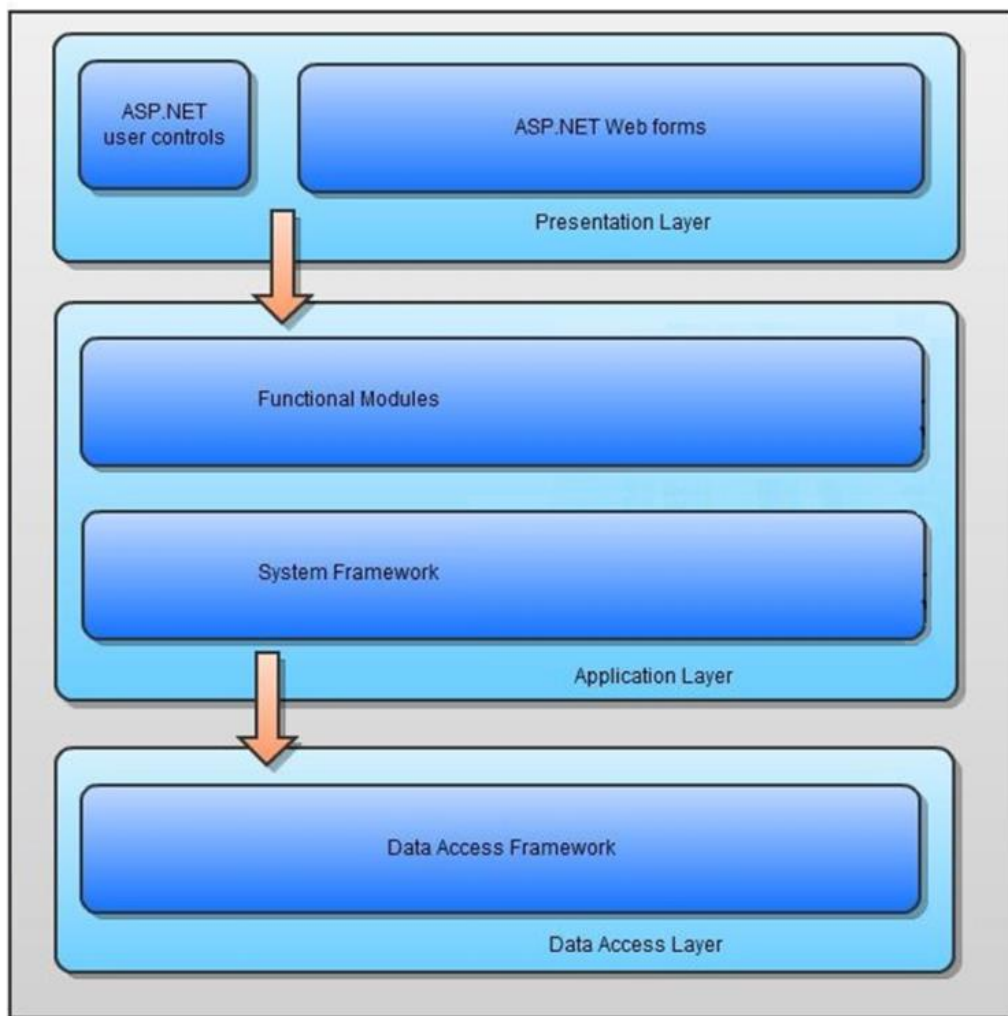
Ter illustratie is in figuur 3-1 de "oude" architectuur te zien van Exact Online. In het lagenmodel is de huidige gewenste situatie van Exact Online aan.

De architectuur bestaat uit drie lagen, namelijk de presentation layer, application layer en data access layer. Een van de belangrijkste kenmerken van een lagen architectuur is dat de componenten van een laag alleen kunnen interacteren componenten in dezelfde laag of de directe laag eronder. Op deze

¹ Asynchronous JavaScript And XML (AJAX) is een term voor het ontwerp van interactieve webpagina's waarin asynchroon gevraagde gegevens worden opgehaald van de webserver. Daardoor hoeven dergelijke pagina's niet in hun geheel ververs te worden.

² jQuery is een multi-browser JavaScript library die is ontworpen om de client-side scripting van HTML te vereenvoudigen. jQuery syntax is zo ontworpen dat het navigeren door een document, selecteren van DOM-elementen, animaties maken, verwerken events en de ontwikkeling van Ajax-toepassingen makkelijker wordt gemaakt.

manier is elke laag op een losse manier gekoppeld aan maar één andere laag, wat de onderhoudbaarheid van een applicatie ten goede komt. De praktijk is dat er veel legacy code is die de lagenstructuur niet gebruikt zoals geïllustreerd in figure 3-1 en beschreven in deze paragraaf.



3-1 Oude lagenmodel architectuur Exact Online (zelf opgesteld aan de hand van (Göritzer, 2011))

3.3.2 Presentation layer

De presentation layer bestaat uit een grote verzameling van ASP.NET Web Forms (opgenomen in de bestanden met de extensie .aspx), die de user interface van Exact Online representeren. Naast de statische HTML opmaak en JavaScript code bevatten deze bestanden ook de opmaak voor het definiëren van de server-side ASP.NET controls. Deze server-side ASP.NET controls worden dynamisch gerendered wanneer een gebruiker een pagina opvraagt. Daarnaast bevat een ASP.NET Web form van Exact Online een aanzienlijke hoeveelheid business logica specifiek afgestemd op het Web form. Dit is niet in overeenstemming met het code-behind model wat Microsoft adviseert, namelijk het scheiden van de business logica en de user interface code door ze in aparte bestanden te plaatsen.

Exact Online heeft verschillende redenen om te kiezen voor het niet scheiden van de business logica en

de user interface code. Een van de redenen is dat het naast elkaar hebben van de business logica en de user interface code de software developers zou helpen bij het krijgen van een compleet overzicht van de broncode zonder te maken krijgen met enige mogelijkheid van verschillende versies tussen de twee bestanden.

Een andere reden is dat code-behind model impliceert dat elk bestand met business logica van een ASP.NET Web form gecompileerd moet worden in een aparte Dynamic Link Library (DLL). Dit betekent dat er honderden extra DDL's verzonden moeten worden (namelijk één voor elke pagina) en dat bemoeilijkt het deployment proces significant. Daarom is een aanzienlijk deel van Exact Online's business logica deel van de presentation layer.

Daarnaast zijn de ASP.NET user controls (opgenomen in de bestanden met de extensie .ascx) welke pagina secties bevatten die regelmatig worden gebruikt door de ASP.NET Web forms. (Göritzer, 2011)

3.3.3 Application layer

De application layer kan ruwweg worden verdeeld in het system framework en een verzameling van de functional modules (zie figuur 3-1). Het system framework bestaat uit een verzameling van DDL's die de kernfunctionaliteit van de software omvat en wordt gebruikt door de gehele applicatie (bijvoorbeeld door de functional modules en de ASP.NET Web Forms).

Exact Online omvat verschillende functional modules (bijvoorbeeld salarisadministratie, logistiek en Human Resource management). Elk klant kan zijn eigen functional modules pakket samenstellen. De algemenere functionaliteiten van deze modules zijn geïmplementeerd in verschillende DDL's die onderdeel zijn van de application layer van Exact Online, oftewel de functional modules. (Göritzer, 2011)

3.3.4 Data access layer

De data access framework vormt de data access layer (zie figuur 3-1) binnen de lagenarchitectuur van Exact Online. Er wordt gebruik van het Microsoft ADO.NET (ActiveX Data Objects .NET) Entity Framework om toegang te krijgen tot de hosting en klanten databases en voor het uitvoeren CRUD (Create, Read, Update en Delete) bewerkingen in deze databases. (Göritzer, 2011)

3.3.5 Nieuwe model

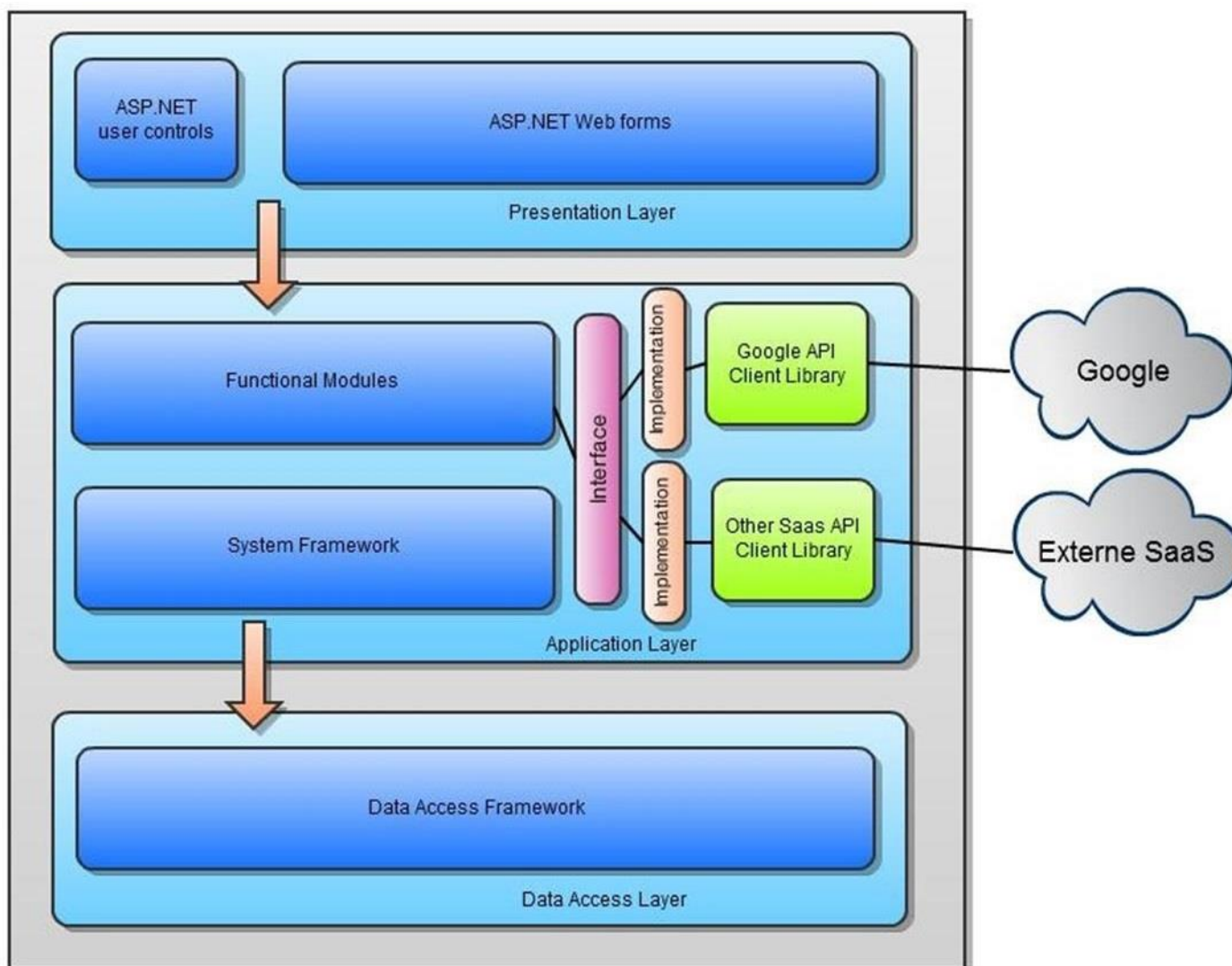
In figuur 3-2 is de nieuwe (gewenste) situatie van Exact Online te zien. De blauwe vakken geven de huidige gewenste situatie van Exact Online aan. Duidelijk is te zien (in vergelijking met figuur 3-1) dat er een interface, implementaties, de libraries en externe SaaS API bij zijn gekomen. Deze toevoegingen worden in deze paragraaf toegelicht. Het nieuwe systeemdeel External Document Management SaaS' Integration zal zich wel de lagenstructuur aanhouden.

De groene blokken zijn de libraries die beschikbaar zijn gesteld door de externe SaaS-oplossing. Een library is een verzameling code (functies/routines) die kan worden gebruikt door het eigen programma of functie. Het voordeel hiervan is dat er geen eigen code geschreven hoeft te worden voor bepaalde algemene bewerkingen. Er kunnen simpelweg functies aangeroepen worden die zich in de bibliotheek bevinden. (WikipediaNL, 2012)

Dit heeft ook voordelen wanneer er veranderingen zijn aan de externe SaaS-oplossingskant. In het geval dat de veranderingen dusdanig zijn dat een nieuwe library gebruikt moet worden, kan de huidige library worden verwijderd en de nieuwe geïmplementeerd. Het enige wat in een dergelijk geval aangepast moet

worden is de implementation. De implementations worden in het model aangegeven met oranjegekleurde blokken naast de libraries.

3-2 Nieuwe lagenmodel architectuur Exact Online (zelf samengesteld)



In paragraaf 3.3.3 is de application layer beschreven. De keuze om de libraries van de externe SaaS-en op te nemen in de application layer is dat in de application layer ook de DDL's van het System Framework en Functional modules zijn opgenomen.

Voor de implementations(oranje blokjes) staat een interface(paarse blokje). Deze interface beschrijft het gedrag van de implementations en is als het ware een contract tussen de verschillende implementations. De interface bepaalt hierin enkel welke gemeenschappelijke functies en properties die implementations moeten hebben. De interface heeft buiten de method signatures en properties feitelijk geen inhoud.

De implementations(oranje blokjes) implementeren de interface(paarse blokje) en moeten daardoor alle functies die beschreven zijn in de interface implementeren. Per implementation kan de invulling van de functies verschillen. Bijvoorbeeld alle implementations zullen een functie hebben in de trant

GetDocuments, bij de implementation van Google Drive zal deze functie anders zijn ingevuld dan de implementation van Dropbox. Echter het resultaat namelijk een lijst van documenten is in beide gevallen hetzelfde.

De implementation code heeft verschillende functies. Ten eerste het verzoek van de client(Exact Online) verwerken naar een request die afgehandeld kan worden door de library(groene blokjes). Hierop krijgt de implementation een antwoord, bijvoorbeeld een lijst met document objecten, en zet in implementation dan om naar een algemeen format(alle implementation gebruiken dit zelfde format). Op die manier kan de client de gegevens makkelijker behandelen. Kortgezegd is de implementation verantwoordelijk voor de communicatie tussen de code van Exact Online (via de interface) en de library van de externe SaaS-oplossing.

Het voordeel van het gebruik van een implementation is dat bij veranderingen in de library alleen de implementation aangepast zal moeten worden. Dit zorgt voor een korte ontwikkeltijd zowel in het begin als in latere stadia.

Het stuk code in Exact Online dat zorgt voor de aanroep van door de interface beschikbare functies hoeft na implementatie niet meer te worden aangepast, tenzij er iets verandert aan de visuele kant van de integratie. Men wil bijvoorbeeld zien wat de wijzigingsdatum is, in plaats van de aanmaakdatum, dan moet de Exact Online code wel worden aangepast.

Het toepassen van het Factory Pattern zorgt onder andere voor deze client code vriendelijkheid. Zie 5.5.2 voor meer over het Factory Pattern.

In de library zijn ook de functies en routines te vinden die de uiteindelijke communicatie met de externe SaaS-oplossing API uitvoeren. De library zet de response van de API om naar een hanteerbaarder format dan het origineel (JSON³). De implementation zet dit nieuwe format om naar een algemeen format wat door alle implementaties wordt gebruikt.

3.4 Functionele capaciteiten

De functionaliteiten die bij de integratie horen zijn in te delen in drie categorieën. Namelijk:

Accountmanagement – Het systeem moet...

- ...een gebruiker kunnen registreren voor het gebruik van een integratie van een externe SaaS.
 - ... de inloggegevens kunnen valideren m.b.v. de externe SaaS.
 - ... de authenticatiegegevens kunnen opslaan in de database.
 - ... de module kunnen activeren.
 - ... de module kunnen deactiveren.
 - ...kunnen controleren of een gebruiker aangemeld is voor de externe SaaS integratie.
-

Activiteiten – Het systeem moet...

- ...connectie kunnen leggen met een externe SaaS.
 - ...de documentenlijst horend bij een bepaald onderwerp kunnen ophalen.
 - ...een zoekopdracht vanuit Exact Online binnen de externe SaaS kunnen uitvoeren.
 - ... de gebruiker kunnen doorsturen naar het viewscherm van een bepaald document in de externe
-

³ JSON is a text-based open standard designed for human-readable data interchange.

SaaS.

- ... een document kunnen kopiëren van de externe SaaS om hem te archiveren in Exact Online.
- ... de gebruiker kunnen doorsturen naar het editscherf van een bepaald document in de externe SaaS.
- ... de gebruiker kunnen doorsturen naar de externe SaaS voor het aanmaken van een document.
- ... kunnen checken of er een map voor een bepaald onderwerp al bestaat in de externe SaaS.
- ... een map kunnen aanmaken voor een bepaald onderwerp.
- ... de gebruiker informeren in geval van een error.
- ... een documentenlijst kunnen tonen in de viewsection.

Systeembeheer – Het systeem moet...

- ... de acties van een gebruiker in de activitylog registreren.
- ... bij foutmeldingen deze ook registreren in de errorlog.
- ... statistieken over de module registreren.
- ... statistieken over de module laten tonen.

Samengevat heeft de integratie als belangrijkste functies het koppelen van documenten aan een bepaald onderwerp (project, account, etc.), het kunnen inzien van documenten en het beheren van de integratie. Daarnaast moeten alle activiteiten, foutmeldingen en statistieken worden bijgehouden en inzichtelijk worden gemaakt.

3.5 Perspectives

3.5.1 Security

OAuth2.0 is een populaire open standaard voor autorisatie. Het geeft de gebruikers de mogelijkheid om hun resources die bij een andere partij zijn opgeslagen beschikbaar te maken zonder dat ze hun inloggegevens hoeven op te geven. In plaats daarvan wordt een tijdelijke token gebruikt voor de communicatie tussen de partijen. Per integratie kan bepaald worden welke OAuth versie gebruikt kan/moet worden. (Wat is OAuth en wat zijn tokens nou precies en wat zijn de verschillen tussen OAuth en OAuth2.0, zie bijlage 1 voor meer informatie)

OAuth2.0 is een meer commerciële versie van OAuth1.0. Over welke qua veiligheid het beste zou zijn verschillen de meningen. Hieronder staat een overzicht van een aantal bekende Service Providers en welke OAuth versie ze gebruiken.

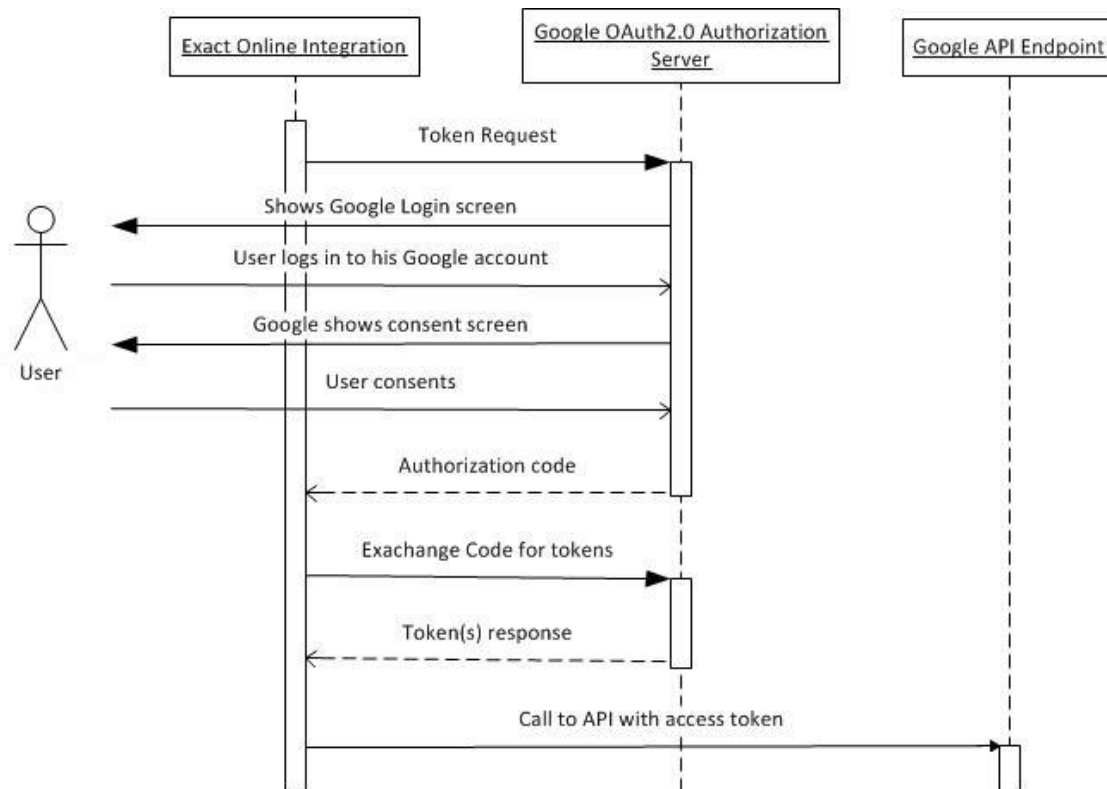
Service Provider	OAuth versie
Dropbox	1.0
Facebook	2.0
Foursquare	2.0
Google	2.0
Instagram	2.0
Microsoft (Hotmail, Messenger, Xbox)	2.0
LinkedIn	1.0a
MySpace	1.0a
Twitter	1.0a
Yahoo!	1.0a

Google schrijft OAuth2.0 voor als manier voor authenticatie en autorisatie. Er zijn nog wel andere manieren zoals ClientLogin en OAuth1.0, maar deze zijn gedeprecieerd door Google en deze mogelijkheden gaan er over onbepaalde tijd geheel uit. Google raad zelf OAuth2.0 aan en dit is ook de enige manier die door library, die zij ter beschikking stellen, wordt ondersteund.

OAuth2.0 houdt bij Google Drive in dat:

- Eerste login bij Google Drive (zie figuur 3-3):
 1. Exact Online doet een verzoek voor tokens.
 2. Google stuurt de gebruiker naar een login scherm(webbrowser).
 3. De gebruiker logt in op zijn Google account (webbrowser)
 4. Google laat een lijst zien van gegevens waartoe Exact Online toegang wil (webbrowser)
 5. De gebruiker geeft akkoord. (webbrowser)
 6. Google stuurt de gebruiker terug naar Exact Online met een autorisatiecode, te vinden in de code parameter van de URL.
 7. De integratie module binnen Exact Online doet een request voor het inwisselen van de autorisatiecode.
 8. Google stuurt credentials terug, welke o.a. een access token en een long-lived refresh token bevatten.
 9. De integratie module binnen Exact Online kan nu requests doen aan de Google API, voor bijvoorbeeld document gegevens, met de access token.

3-3 Illustratie van OAuth2.0 flow Google (aangepast diagram origineel bron: Google)



Wanneer een accesstoken is verlopen wordt er een nieuwe aangevraagd door middel van een HTTPsRequest. In dat HTTPsRequest wordt onder andere de refreshtoken meegestuurd.

De maatregelen op het gebied van security van de data en datastromen zijn te vinden in 4.2.1.

Alle maatregelen die getroffen worden ten aanzien van de security voldoen aan de beveiligingseisen zoals die gelden binnen Exact. De beveiligingseisen van Exact Online zijn zeer strikt en divers. Voor dit project om te voldoen aan deze beveiligingseisen, houdt vooral in dat het framework van Exact Online gebruikt moet worden zoals het bedoeld is. Collega's zullen hier regelmatig op controleren. Ook zal een extern bedrijf regelmatig een controle doen op het gebied van security zodat Exact kan garanderen dat hun producten voldoen aan de laatste beveiligingsregels en -mogelijkheden.

3.5.2 Evolution

In de toekomst zal er aan het nieuwe systeemdeel veel evolueren. Zo komen er externe SaaS-koppelingen bij, worden er koppelingen weggehaald of ze worden vernieuwd. Aan hetgeen er mee bereikt moet worden verandert niet veel. Het hoofddoel blijft het beschikbaar maken van documenten van een externe SaaS-oplossing in Exact Online. De functies die dit hoofddoel moeten realiseren zullen dan ook niet veel veranderen in de tijd, wat betekent dat het mogelijk is om deze functies te hergebruiken bij zowel nieuwe als vernieuwende koppelingen.

De richtlijnen die hiervoor gevolgd dienen te worden zijn terug te vinden in het Development Viewpoint.

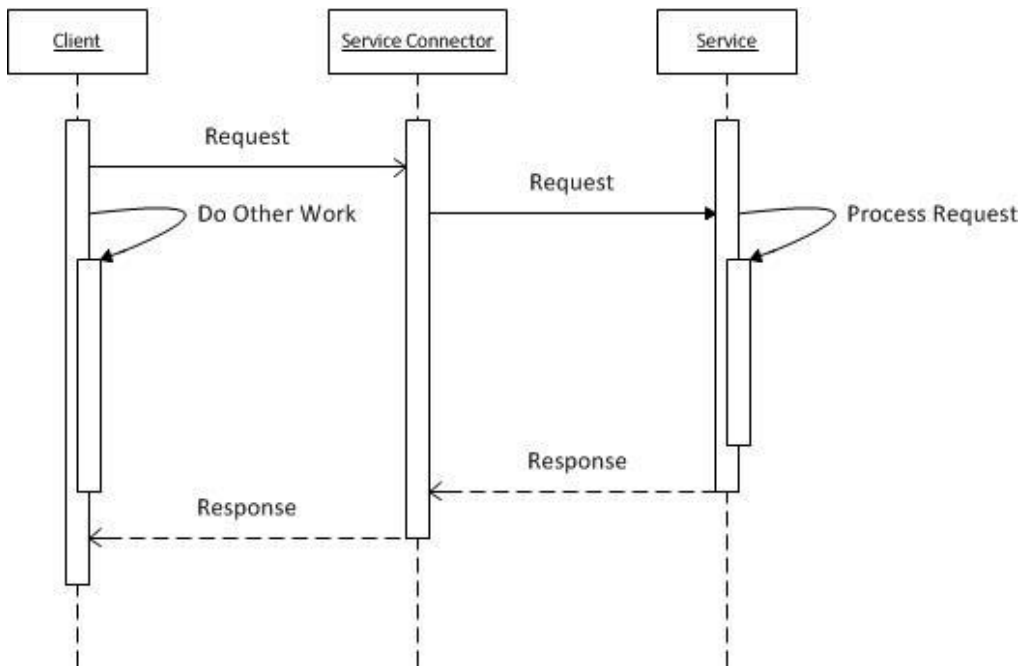
3.5.3 Performance and scalability

Om de performance zo hoog mogelijk te houden wordt er zo weinig mogelijk data opgehaald, door gebruik te maken van de filtermogelijkheden bij de externe SaaS-oplossingen. Hun servers zijn in de meeste gevallen veel krachtiger en kunnen de filtering sneller toepassen dan Exact Online zelf. Daarbij is de serverbelasting voor Exact Online ook veel lager, wanneer de filtering al heeft plaatsgevonden.

Het is mogelijk dat de externe SaaS-oplossing niet beschikbaar is of problemen ondervindt. Wanneer dit het geval is, moet dit niet de beschikbaarheid van Exact Online aantasten. Daarom wordt er een tijdslimiet verbonden aan de communicatie. Wanneer deze limiet wordt overschreden wordt er gestopt met de communicatie met de externe SaaS-oplossing. De gebruiker wordt hiervan op de hoogte gesteld en de pagina wordt normaal verder geladen. De gebruiker kan er dan eventueel voor kiezen om op een later tijdstip het nog eens te proberen.

Ondanks de maatregelen zoals het filteren van gegevens kan het zijn dat het ophalen van gegevens bij een externe SaaS tientallen, soms wel honderdtallen milliseconden in beslag kan nemen. Door deze gegevens asynchroon op te halen als taak op de achtergrond, kan de rest van de taken die horen bij de pagina worden uitgevoerd. Wanneer de uiteindelijke pagina wordt opgebouwd, kunnen de gegevens van de achtergrondtaak worden toegevoegd aan het geheel. Indien de achtergrond taak nog niet klaar is, moet daar op gewacht worden. Deze werkwijze is geïllustreerd in figuur 3-4.

Deze methode is sneller dan het moeten wachten tot de taak van gegevens ophalen bij de externe SaaS en daarna pas doorgaan met het laden van de rest van de pagina. De technische invulling van deze methode is te vinden in het Development Viewpoint, §5.5.1.



3-4 Globale weergave asynchrone werking (aangepast model origineel bron: <http://www.servicedesignpatterns.com>)

Om de performance te optimaliseren is het nodig om de communicatie tussen Exact Online en andere SaaS-oplossingen zo veel mogelijk te beperken. Zowel in grootte van de totale vraag door alleen op te vragen wat je gaat gebruiken, als door niet voor elk dingetje een aparte aanvraag naar de externe SaaS-oplossing te doen, maar te letten op wat te combineren valt. Het laatste zal slechts in een enkel geval mogelijk zijn, maar elke aanvraag is er één.

3.5.4 Availability

De beschikbaarheid Exact Online moet niet leiden onder eventuele connectieproblemen met een externe SaaS, daarom wordt er een tijdslimiet ingesteld en bij het overschrijden van de tijdslimiet wordt de connectie met de externe SaaS afgekapt. De gebruiker krijgt een melding te zien, dat het ophalen van de documentgegevens op dat moment niet mogelijk is en de vraag het later opnieuw te proberen. Deze tijdslimiet wordt toegepast in het asynchronous pattern. Meer over het asynchronous pattern in 5.5.1.

4. Information Viewpoint

4.1 Concerns

De concerns die in dit viewpoint worden behandeld, zijn:

1. De gegevensuitwisseling tussen Exact Online en externe SaaS-oplossingen is beveiligd en voldoet aan de huidige beveiligingseisen die worden gehanteerd binnen Exact.
2. De laadtijd van een pagina wordt met hooguit tientallen milliseconden vertraagd bij het ophalen van de documentenlijst.
3. Bij het uploaden naar de Exact Server wordt een controle gedaan op kwaad kunnende bestanden.
4. De rechten tot mappen en bestanden zoals ze binnen de externe SaaS-oplossing zijn ingericht, zijn binnen Exact Online hetzelfde ingericht.

4.2 Perspectives

4.2.1 Security

Tijdens de communicatie tussen Exact Online en een andere SaaS-oplossing wordt er gevoelige informatie verzonden. Gevoelige informatie waaraan gedacht moet worden is:

- Gebruikersinformatie afkomstig van het Google Account
- Namen projecten en klanten Exact Online
- Informatie over bestanden van Google Account
- Inhoud documenten van Google Account

Door alleen de informatie te sturen en op te vragen die echt nodig is, wordt het risico dat derden door onwettige activiteiten toegang hebben tot voor hun bruikbare informatie geminimaliseerd. Bijvoorbeeld: Voor het opstellen van een documentenlijst zijn er per document verschillende gegevens beschikbaar, denk aan titel, labels, grootte bestand, aanmaakdatum, gebruikersnaam van degene die het heeft aangemaakt, laatste wijzigingsdatum etc. Door alleen de gegevens op te vragen die ook echt worden gebruikt op dat moment voorkom je onnodige risico's. Het komt ook de performance van Exact Online ten goede. (zie stroom A in figuur 4-1)

Bij het archiveren van een document uit Google Drive naar Exact Online wordt de inhoud van een document wel opgehaald. Daarom vindt hierbij altijd een controle op de inhoud plaats voordat een bestand wordt geüpload naar de Exact Online Server. Hiervoor zijn al functies aanwezig binnen de huidige oplossing en deze zullen geheel of in ieder geval zoveel mogelijk worden gebruikt. (zie stroom B in figuur 4-1)

Voor functionaliteiten als het inzien en wijzigen van een document wordt er een nieuw scherm geopend met een redirect naar Google Drive, waar het document wordt geopend. De gebruiker kan hierdoor de voor hem bekende functionaliteiten van Google Drive gebruiken en het risico op informatieonderschepping wordt sterk verkleind doordat de inhoud van een document niet wordt opgehaald, althans niet door Exact Online. (zie stroom C in figuur 4-1)



4-1 Document(gegevens) stromen tussen Exact Online en Google Drive

Een andere belangrijke functionaliteit op het gebied van informatie security is het geven van rechten aan bepaalde gebruikers. Zo is het bij Google Drive mogelijk om rechten per map en zelfs per bestand te regelen, bijvoorbeeld:

- Hoofdgebruiker heeft recht op toegang tot alles (dit kan je veranderen maar is niet wenselijk)
- Medewerker A: heeft toegang tot ProjectA, ProjectB, ProjectE, etc.
- Medewerker B: heeft toegang tot Project A, ProjectC, ProjectD etc.

Er kan dus per gebruiker worden bepaald welke rechten hij heeft en deze rechten worden dan automatisch overgenomen binnen de integratie in Exact Online. (inrichting van de relatieboom wordt besproken in 4.3).

4.2.2 Regulation

Wanneer een gebruiker kiest voor het gebruik van de integratiemogelijkheden met externe SaaS-oplossingen, accepteert de gebruiker hiermee automatisch de eventuele privacy beperkingen die horen bij de externe SaaS-oplossingen.

Voor het vastleggen van gegevens van gebruikers gelden strenge regels. Daarom wordt er zo weinig mogelijk vastgelegd van de gebruiker en niets van de documenten, tenzij expliciet aangegeven (archiveren van een document). Van een gebruiker wordt in de database alleen het id/emailadres, het short-lived accesstoken and long-lived refreshtoken opgeslagen.

In de logs worden bepaalde activiteiten en errors anoniem opgeslagen en andere niet. In sommige gevallen moet je namelijk in staat zijn om een bepaalde situatie na te bootsen om een probleem op te lossen, in andere gevallen zijn statistieken genoeg.

Om gebruik te kunnen maken van een externe SaaS API moet er vaak een account worden geregistreerd. Een account dat de API access mogelijk maakt door het aanvragen van een clientid, clientsecret etc.

Hiermee bedoelen we dus niet het account van een gebruiker, maar een soort masteraccount, zonder toegang tot andere accounts. Hiervoor kan Exact een nieuw Google Account aanmaken of een account gebruiken van een medewerker. Dit laatste heeft de nodige risico's (bij het vertrek van de desbetreffende medewerker moeten de clientgegevens aangepast worden), daarom wordt er gekozen voor een apart account dat onder het beheer van het System Team valt of in de toekomst het team dat de verantwoordelijkheid heeft over de integraties.

4.2.3 Usability

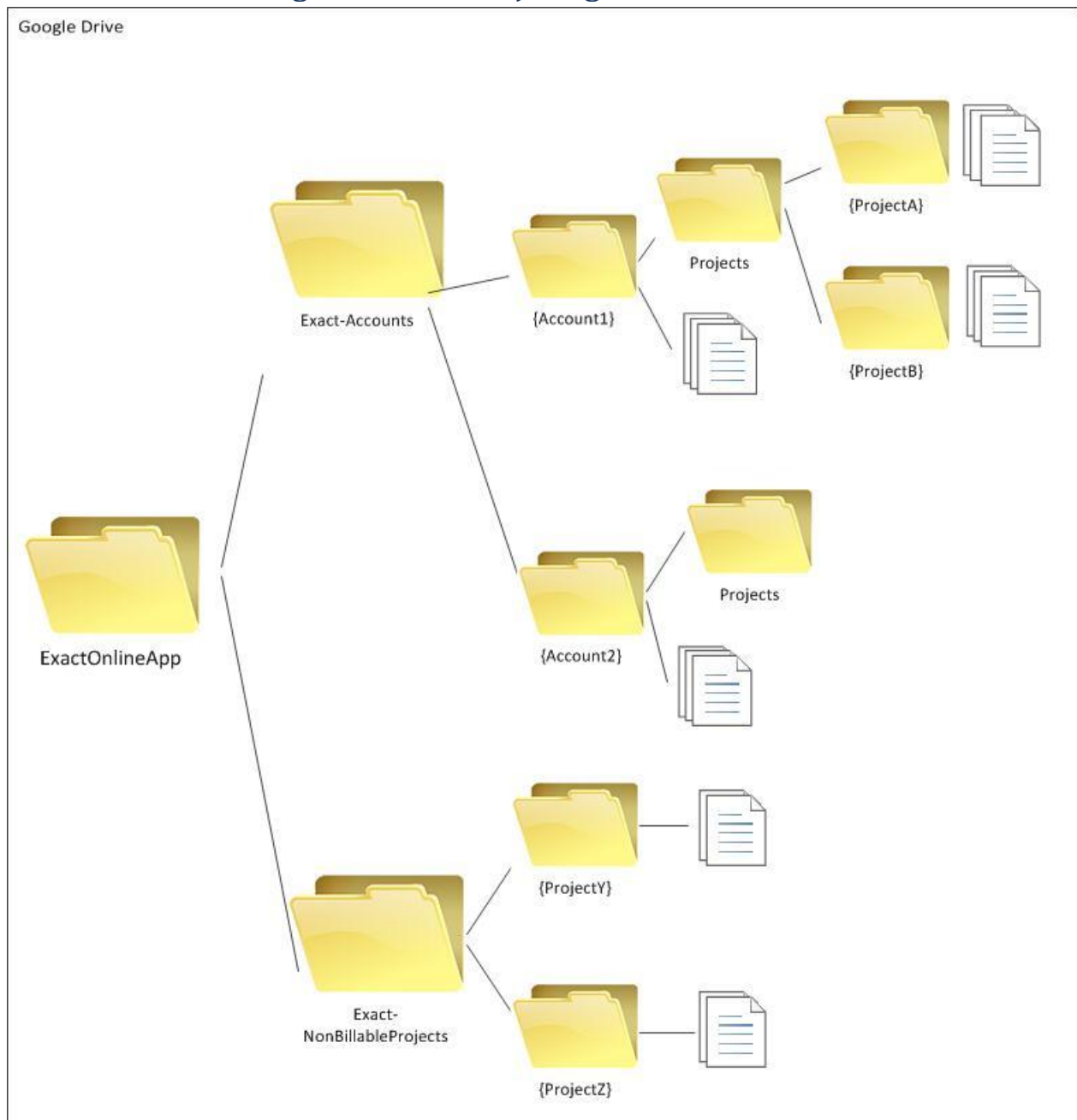
De informatie waarover een gebruiker wil beschikken moet up-to-date zijn, daarom wordt de informatie rechtstreeks(realtime) opgehaald bij de bron. Wanneer er iets verandert in de bron is deze nieuwe informatie meteen beschikbaar (het moet wel opnieuw opgehaald worden, er is geen continue verbinding).

Wanneer er gezocht wordt naar bepaalde informatie, wil je dat de meest relevante resultaten als eerste worden getoond. Dit verhoogt de usability voor de gebruiker. Door het gebruik van een sterke zoekmachine, bijv. Google, wordt het zoeken naar relevante informatie of naar dat ene bestand minder werkintensief. In de toekomst is het ook een mogelijkheid om de zoekfunctie centraal te maken, waardoor de zoekresultaten van Exact Online en de geïntegreerde SaaS-oplossingen kunnen worden samengevoegd.

4.2.4 Performance en Scalability

De gegevens die voor een gebruiker moeten worden opgeslagen, worden in de huidige gebruikerstabel toegevoegd. In januari waren er 106.571 gebruikers voor Nederland en België met in totaal bijna 1,5 miljoen usersettings(waar de integrationsettings onderdeel van zijn/worden). Een alternatief zou zijn geweest om de gegevens in een aparte tabel op te slaan, maar met het oog op de toekomst zou dit problemen kunnen opleveren. Daarbij beschikt de huidige database over de mogelijkheid om door het indexeren en het maken van een soort view, de gegevens snel op te kunnen halen.

4.3 Inrichting relatieboom bij Google Drive



4-2 Relatieboom mappenstructuur in GoogleDrive voor integration

Bij onderzoek naar Google Drive bleek het niet mogelijk om extra tags toe te voegen aan een bestand of map op een manier die voor de gebruiker ook buiten Exact Online te gebruiken is. Een andere optie was het gebruik van mappen om de relatiestructuur op te bouwen (zie figuur 4-2). Per map en zelfs per bestand kan je een andere gebruiker toegangsrechten toekennen. Om deze relatieboom te kunnen gebruiken binnen Exact Online is de voorwaarde dat de map “ExactOnlineApp” en in deze map, de mappen “Exact-NonBillableProjects” en “Exact-Accounts”, te allen tijde aanwezig is en dat een gebruiker ook rechten heeft tot deze mappen. Per accountmap onder “Exact-Accounts” is er ook altijd een map

“Projects” aanwezig. (Deze namen zullen in de toekomst taal/land afhankelijk worden).

Per map in projecten en accounts kunnen de rechten anders zijn. De rechten die kunnen worden toegekend zijn:

- Is eigenaar
- Mag bewerken
- Kan weergeven

Bij de aanmelding van een nieuw account worden alleen de mappen “ExactOnlineApp”, “Exact-NonBillableProjects”, “Exact-Accounts” aangemaakt. De gebruiker moet vervolgens per project en account aangeven of hiervoor een map moet worden aangemaakt (een map wordt aangemaakt op het moment dat een gebruiker een bestand aanmaakt en de account-/projectmap bestaat nog niet). Dit is ter voorkoming van het aanmaken van honderden mappen in Google Drive die niet of niet in de nabije toekomst gebruikt gaan worden. Deze maatregel komt zowel de performance als de userexperience ten goede. Daarnaast kan de gebruiker bestaande mappen plaatsen in de bovengenoemde mappenstructuur. Wanneer de map op de juiste plek staat en dezelfde naam heeft als het account in Exact Online, kan de integratiemodule binnen Exact Online deze documenten(gegevens) ophalen.

5. Development Viewpoint

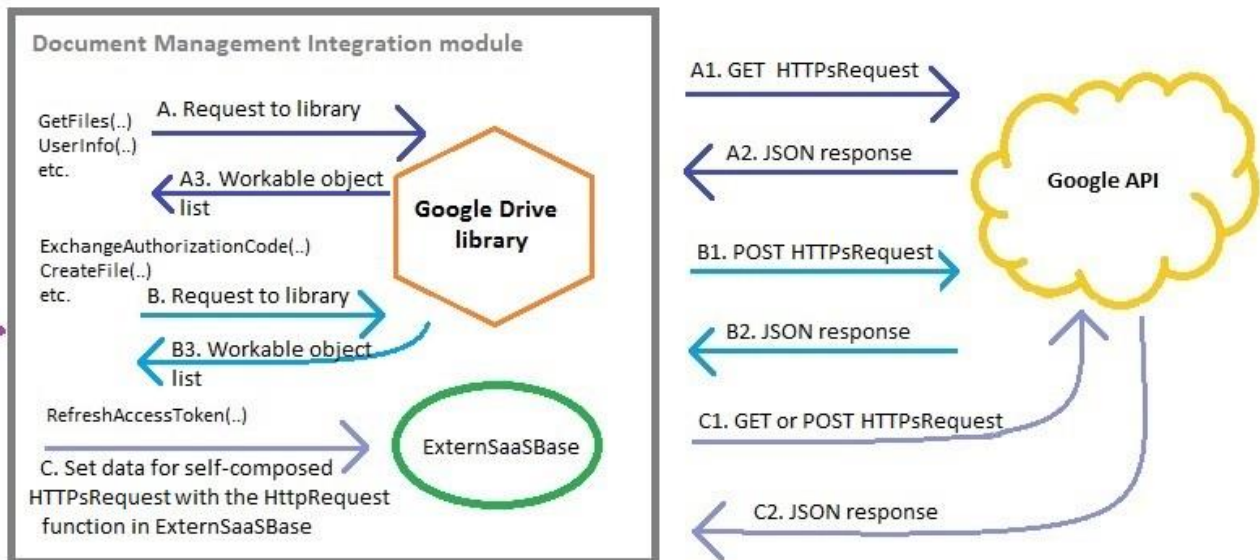
5.1 Concerns

De concerns die in dit viewpoint worden behandeld, zijn:

1. De gegevensuitwisseling tussen Exact Online en externe SaaS-oplossingen is beveiligd en voldoet aan de huidige beveiligingseisen die worden gehanteerd binnen Exact.
2. De laadtijd van een pagina wordt met hooguit tientallen milliseconden vertraagd bij het ophalen van de documentenlijst.
3. Bij het uploaden naar de Exact Server wordt een controle gedaan op kwaad kunnende bestanden.
4. Activiteiten en foutmeldingen worden vastgelegd.
5. De beschikbaarheid van Exact Online wordt niet beïnvloed door de integratie.
6. Het nieuwe systeemdeel heeft dezelfde leercurve als het huidige documentengedeelte.
7. Het nieuwe systeemdeel is eenvoudig te onderhouden en uit te breiden
8. Het systeem heeft de functionaliteiten zoals genoemd in het document "Requirements opstellen"

5.2 Libraries versus eigen HTTPsRequest opstellen

Voor zowel het gebruik van beschikbare libraries als het schrijven van de HTTPsRequest is iets te zeggen. Het voordeel van het gebruik beschikbare libraries is dat de functies van de library meteen beschikbaar zijn om te gebruiken in de implementatie. In het geval van de integratie van Google Drive neemt de library ook meteen de vertaling, van de geretourneerde data, van JSON⁴ naar een meer hanteerbare lijst met de documenten en hun gegevens voor zijn rekening. Een groot nadeel van een library is dat je gebonden bent aan de binnen de library beschikbare functies. . In figuur 5-1 worden de HTTPsRequests die door de library worden afgehandeld aangegeven met de letters A en B.



5-1 Schematische weergave van de drie verschillende HTTPsRequests en hun flow

⁴ JSON is a text-based open standard designed for human-readable data interchange.

Bij het gebruik van een HTTPsRequest heb je de vrijheid om het request naar de externe SaaS zelf vorm te geven. Het nadeel is wel dat de geretourneerde data ook met een zelf geschreven functie naar een meer hanteerbare lijst moet worden vertaald. Dit kost bij een kleine verandering, bijvoorbeeld in plaats van de creëerdatum de datum van de laatste wijziging aan een document, meer tijd in vergelijking met het gebruik van een library. In figuur 5-1 worden de HTTPsRequests die buiten de library (dus zelf) worden opgesteld en afgehandeld aangegeven met de letter C. Voor meer uitleg over figuur 5-1, zie bijlage H – Technical Documentation, paragraaf 1.5.

Een combinatie van het gebruik van de beschikbare library en het gebruik van HTTPsRequests is daarom voor dit project het beste. De basis functies zijn gedekt in de libraries en waar de library beperkt is, kan een HTTPsRequest worden gebruikt. In het geval van de integratie van Google Drive komt deze combinatie voor. Voor het ophalen van documenten zijn er vele functionaliteiten in de library aanwezig en is er dus geen noodzaak om daar zelf een HTTPsRequest voor op te stellen. De functie voor het refreshen van een accesstoken die in de Google Drive library beschikbaar is echter omslachtig en vraagt ook actie van de gebruiker. Door voor deze functie zelf een HTTPsRequest op te stellen gebeurt het refreshen van de accesstoken zonder dat de gebruiker dit merkt.

5.3 Eisen aan library

Wanneer er voor een integratie een library beschikbaar is, moet eerst worden beoordeeld of deze geschikt is voor gebruik binnen de integratie. De library wordt beoordeeld aan de hand van ISO 25010 (deze vervangt sinds 2011 ISO 9126). Er wordt gekeken naar de volgende punten:

- **Functional suitability:**
 - Functional completeness: De mate waarin de set van functies binnen een library alle gespecificeerde taken en gebruikersdoelen afdekt.
 - Functional correctness: De mate waarin een library de correcte resultaten met de gewenste mate van nauwkeurigheid levert.
 - Functional appropriateness: De mate waarin de functies van de library het bereiken van gespecificeerde taken en doelen mogelijk maken.
- **Performance efficiency:**
 - Time-behaviour: De mate waarin antwoord- en verwerkingstijden en doorvoersnelheid van de library, tijdens de uitvoer van zijn functies, voldoet aan de wensen.
 - Resource utilization: De mate waarin de hoeveelheid en type middelen die gebruikt worden door een library, tijdens de uitvoer van zijn functies, voldoet aan de wensen.
 - Capacity: De mate waarin de maximale limieten van een library voldoet aan de wensen.
- **Usability:**
 - Appropriateness recognisability: De mate waarin de developers kunnen herkennen of een library geschikt is voor hun behoeften.
 - Learnability: De mate waarin een developer zich nieuwe vaardigheden moet aanleren om de library effectiviteit en efficiëntie te kunnen gebruiken.
 - Operability: De mate waarin een library attributen heeft die het makkelijk maken om het te bedienen en beheersen.
 - Accessibility: De mate waarin de library gebruikt kan worden door developers met de meest uiteenlopende eigenschappen en mogelijkheden om een gespecificeerd doel te bereiken in een gespecificeerde developerscontext.
- **Reliability:**
 - Maturity: De mate waarin een library aan betrouwbaarheidsbehoeften voldoet onder normale werkomstandigheden.
- **Security:**
 - Confidentiality: De mate waarin een library er voor zorgt dat gegevens alleen toegankelijk zijn voor diegenen die geautoriseerd zijn.
 - Authenticity : De mate waarin bewezen kan worden dat de identiteit van een onderwerp of bron is zoals wordt beweerd.
De mate waarin een claim over de oorsprong of de auteur van de informatie verifieerbaar is.
- **Maintainability:**
 - Reusability: De mate waarin een library gebruikt kan worden in meer dan één systeem of bij het bouwen van een nieuw onderdeel.
 - Analysability: De mate waarin het mogelijk is om effectief en efficiënt de impact van een geplande verandering van de integratie met de library te beoordelen.

- Modifiability: De mate waarin een library effectief en efficiënt gewijzigd kan worden zonder fouten of kwaliteitsvermindering tot gevolg.
- Testability: De mate waarin effectief en efficiënt testcriteria vastgesteld kunnen worden voor een library en waarin tests uitgevoerd kunnen worden om vast te stellen of aan die criteria is voldaan.

5.4 Perspectives

5.4.1 Evolution

Het nieuwe systeemdeel zal in de toekomst evolueren. Er komen meer koppelingen met externe SaaS-oplossingen, er gaan er ook weer een aantal weg en externe SaaS-oplossingen kunnen ook veranderen. Het nadeel van externen in dit geval is dat je er geen invloed op hebt. Daarom is het zaak om de tijd die nodig is om te koppelen met een externe SaaS-oplossing zo kort mogelijk te houden.

Eén van de maatregelen die hiervoor wordt genomen is het gebruik van de bestaande libraries voor de communicatie met de API van de externe SaaS-oplossingen. Bij de meeste gevallen zijn deze gewoon voor handen en dit scheelt veel ontwikkeltijd. Daarbij gebruiken de nieuwe functionaliteiten dezelfde user interface als de rest van de Exact Online oplossing. Het enige wat nu nog ontwikkeld moet worden is een implementation om te communiceren met de library. Door een interface voor de implementations te zetten is het mogelijk om alle externe SaaS-oplossingen op dezelfde manier aan te roepen. De interface zorgt er dan voor dat de juiste externe SaaS-oplossing wordt gebruikt en het gewenste resultaat levert.

Voor en tijdens het beheer van de integraties zal er regelmatig een controle plaatsvinden of de beveiligingseisen die gelden voor alle Exact producten gewaarborgd zijn. Dit gebeurt door reviews door collega's en door een extern bedrijf met experts op het gebied van security.

5.4.2 Performance en Scalability

Door functies algemeen op te stellen zijn ze eenvoudig te hergebruiken bij nieuwe of vernieuwde integratie. Indien beschikbaar wordt de library zoals beschikbaar gesteld door de externe SaaS overgenomen. In de interface worden de volgende functienamen gebruikt:

- setupService
- checkRefreshAccessToken
- getAuthorizationUrl
- getFiles
- createFile
- showAll
- searchFiles
- revoked

Specifieke functies die nodig zijn bij een integratie om de library te kunnen gebruiken worden in de implementation geplaatst van die specifieke externe SaaS-oplossing. (zie figuur 3-2)

Om het uitbreiden en onderhoud van de integraties overzichtelijk te houden wordt de codeconventie gevolgd zoals die in Exact Online wordt gebruikt. De libraries zullen niet aangepast worden aan de codeconventie, maar worden gebruikt zoals ze zijn.

Ter voorkoming van lange laadtijd wordt er gebruik gemaakt van het asynchronous pattern (zie 5.5.1). Het ophalen van data bij de externe SaaS kan namelijk meer dan enkele tientallen milliseconden in beslag nemen. Daarom is ervoor gekozen om het ophalen van de data als “asynchronous task” uit te voeren zodat de laadtijd van een pagina zo kort mogelijk wordt gehouden.

Voor het uitvoeren van een “asynchronous task” is ook een tijdslimiet gesteld. Bij overschrijding van de tijdslimiet wordt het proces beëindigd en wordt de rest van de pagina geladen als gebruikelijk. Hierdoor kan de gebruiker wel de functionaliteiten van Exact Online blijven gebruiken. Aan de gebruiker wordt een melding getoond dat het op dat moment niet mogelijk is om de data van de externe SaaS op te halen en dat de gebruiker het op een later tijdstip opnieuw kan proberen.

De integraties kunnen een grote meerwaarde vormen voor de gebruikers. Alleen is het wel zaak dat de huidige functionaliteiten van Exact Online gewaarborgd blijven. De maatregelen die hiervoor genomen worden zijn: controleren of de externe SaaS beschikbaar is en het instellen van een tijdslimiet.

Bij het ophalen van data bij een externe SaaS wordt alleen opgehaald wat echt nodig is. Gebruik je voor een documentenlijst alleen de titel, id en een link naar de externe SaaS, dan worden ook alleen die gegevens opgehaald. Dit wordt bereikt door bij de externe SaaS al te filteren.

Wanneer een gebruiker een document wil inzien of wijzigen klikt hij/zij op de daarvoor bedoelde link. Het systeem opent dan een nieuw scherm die het document opent bij de externe SaaS. Hierdoor kan de gebruiker zijn document bekijken en bewerken zoals hij/zij gewend is. Daarbij komt dat het voor Exact meerdere voordelen oplevert, bijvoorbeeld dat de server van Exact niet belast hoeft te worden om de documentinhoud op te halen.

Wanneer een gebruiker een document vanuit een externe SaaS wil archiveren in Exact Online moet de documentinhoud wel worden opgehaald. Voor het controleren van de inhoud van een bestand zijn binnen Exact Online al functionaliteiten aanwezig, deze zullen voor de inhoudscontrole worden gebruikt.

Om te monitoren hoeveel gebruik er wordt gemaakt van een integratie en waar er functionaliteiten falen, worden er logs (activity- en errorlog) bijgehouden. Als de performance van een systeem sterk vermindert kan uit de logs worden afgeleid welke activiteiten hebben plaatsgevonden en welke fouten daarbij zijn opgetreden. Deze gegevens worden ook eventueel bijgehouden als statistieken in de database. Google houdt zelf ook statistieken over het gebruik van de Google Drive API door Exact Online. Deze statistieken zijn met het Google account dat is gebruikt om Exact Online te registreren als Google Drive API gebruiker, te bekijken.

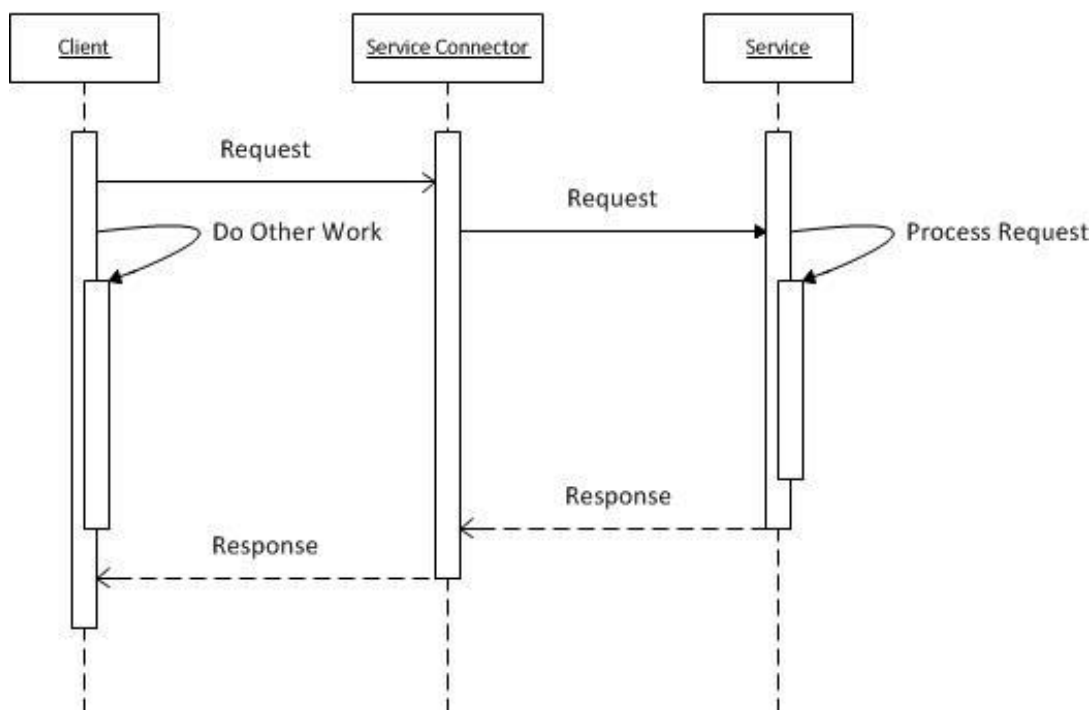
5.5 Design patterns

5.5.1 Asynchronous Pattern

Een asynchrone werking houdt in dat een proces onafhankelijk werkt van andere processen (zie figuur 5-2). In tegenstelling tot een synchrone werking waar processen elkaar opvolgen en afhankelijk zijn van elkaar om het gewenste resultaat te behalen.

Om lange laadtijd van pagina's te voorkomen, kan er voor gekozen worden om sommige taken asynchroon uit te voeren. In het geval van Exact Online worden deze taken dan op de achtergrond uitgevoerd en zodra een zogenaamde "asynchronous task" klaar is, wordt de aanvrager daarvan op de hoogte gebracht. Of een taak wordt in de achtergrond uitgevoerd en het resultaat wordt opgehaald zodra het hoofdproces het resultaat van deze "asynchronous task" nodig heeft.

Voor het uitvoeren van een "asynchronous task" is ook een tijdslimiet gesteld. Bij overschrijding van de tijdslimiet wordt het proces beëindigd en wordt de rest van de pagina geladen als gebruikelijk. Dit heeft als doel dat een pagina niet te lang of oneindig gaat zitten wachten op het afronden van een "asynchronous task". Er kan immers een probleem zijn bij de externe SaaS waardoor het niet mogelijk is om de documentgegevens op te halen op dat moment.



5-2 Globale weergave asynchrone werking (aangepast model origineel bron: <http://www.servicedesignpatterns.com>)

Een simpel voorbeeld van de implementatie van het task-based asynchronous pattern is het volgende (zie figuur 5-3):

```
static void Main() {
    // Construct a TaskCompletionSource and get its
    // associated Task
    TaskCompletionSource<int> tcs =
        new TaskCompletionSource<int>();
    Task<int> task = tcs.Task;

    // Asynchronously, call SetResult on TaskCompletionSource
    ThreadPool.QueueUserWorkItem(_ => {
        Thread.Sleep(1000); // Do something
        tcs.SetResult(123);
    });

    Console.WriteLine(
        "The operation is executing asynchronously...");
    task.Wait();

    // And get the result that was placed into the task by
    // the TaskCompletionSource
    Console.WriteLine("The task computed: {0}", task.Result);
}
```

5-3 codevoorbeeld implementatie Task-based Asynchronous Pattern

De taken die bij documentenintegratie dus moeten worden uitgevoerd zijn:

1. Het toevoegen van de “asynchronous task” voor het ophalen van de documentgegevens
2. Het eerste deel van de pagina opbouwen
3. Wachten tot de “asynchronous task” is afgerond
4. Resultaat “asynchronous task” ophalen
5. De rest van de pagina opbouwen

5.5.2 Factory method pattern

Om het implementeren van externe SaaS-oplossingen makkelijker te maken nu en in de toekomst wordt het factory method pattern gebruikt.

Een factory is een object voor het creëren van andere objecten. Het is een abstractie van een constructor en kan worden gebruikt voor toewijzen van de verschillende implementaties.

Het factory object heeft een methode voor elke implementatie, die hij in staat is om te creëren. Deze methoden accepteren parameters die definiëren hoe een object gecreëerd moet worden en retourneert het gecreëerde object naar de client.

De factory bepaalt welke soort implementatie(Google Drive, Dropbox etc.) gecreëerd moet worden en deze wordt dan ook binnen de factory gecreëerd. De factory retourneert alleen een abstracte pointer aan de client(Exact Online), hierdoor weet de client code niet welke soort implementatie gecreëerd is en wordt dus ook niet met deze details belast. De door factory gecreëerde implementatie is wordt door de client aangesproken door hun abstracte interface.

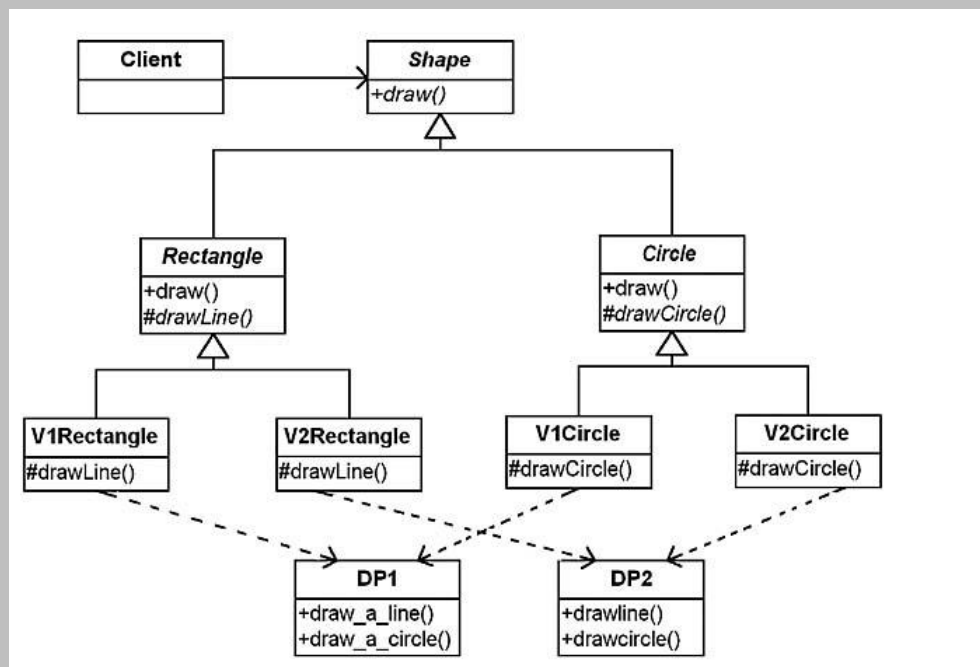
Voor het toevoegen van nieuwe implementaties hoeft de client code door het gebruik van de factory pattern dus niet aangepast te worden. Een nieuwe implementatie wordt toegevoegd aan de factory en de factory bepaalt vervolgens welke implementatie in welke situatie van toepassing is. Dit is de belangrijkste reden voor het gebruik van de factory pattern.

Een samenvatting van het factory method pattern is hieronder weergegeven(zie figuur 5-4). De daadwerkelijke implementatie van het factory method pattern is te vinden in de technische documentatie.

5-4 Factory Pattern volgens “Design Patterns Explained” (Shalloway & Trott, Design Patterns Explained, 2010)

The Factory Pattern: Key Features

Intent	You want to have families or sets of objects for particular clients (or cases)
Problem	Families of related objects need to be instantiated.
Solution	Coordinates the creation of families of objects. Gives a way to take the rules of how to perform the instantiation out of the client object that is using these created objects.
Participants and collaborators	The Factory defines the interface for how to create each member of the family of objects required. Typically, each family is created by having its own unique ConcreteFactory .
Consequences	The pattern isolates the rules of which objects to use from the logic of how to use these objects.
Implementation	Define an abstract class that specifies which objects are to be made. Then implement one concrete class for each family. Tables or files can also be used to accomplish the same thing.



Bron: (Shalloway & Trott, Design Patterns Explained, 2010)

6. Key scenario's

De key scenario's zijn afgeleid uit het document "Requirements opstellen" en kunnen vanwege de SCRUM methode afwijken van de uiteindelijke realisatie van de userstories.

Naam	Aanmelding module
Samenvatting	De gebruiker meldt zich aan voor de module Google Integration
Actor	Gebruiker
Aannamen	- De gebruiker heeft rechten voor het aanmelden voor de module Google Integration - De gebruiker bevindt zich in het scherm General -> General Setup -> My account data
Beschrijving	(1) De gebruiker geeft aan zijn Google Drive account te willen linken aan zijn Exact Online account (2) Het systeem opent de OAuth-pagina van Google Drive (3) Google Drive vraagt om toestemming voor externe toegang tot bepaalde gegevens (4) Google Drive geeft het inlogscherm weer (5) De gebruiker voert de inloggegevens van zijn Google Drive account in en klikt op inloggen (6) Google Drive valideert de inloggegevens (7) Google Drive geeft een lijst met gegevens weer waar Exact Online toegang toe wil hebben. (8) De gebruiker klikt op de button "Allow" (9) Google Drive stuurt de gebruiker terug naar Exact Online (10) Het systeem haalt de autorisatiecode uit de url (11) Het systeem wisselt de autorisatiecode in voor credentials (12) Het systeem registreert het Google Drive ID van de gebruiker samen met de refreshtoken in de database (13) Het systeem activeert de module (14) Het systeem registreert dit event in de activitylog
Uitzonderingen	[inlogfout] De inloggegevens die de gebruiker heeft ingevoerd komen niet voor bij Google Drive, error wordt geregistreerd in de errorlog [geenToestemming] De gebruiker gaat niet akkoord met de mogelijkheid met de externe toegang tot de voorgestelde gegevens [verbindingsprobleem] Er kan geen connectie gemaakt worden met Google Drive, er wordt een foutmelding getoond, error wordt geregistreerd in de errorlog
Resultaat	De module Google Integration is geactiveerd

Naam	Documentenlijst bepaald onderwerp (project/ account)
Samenvatting	Het systeem toont de documenten horend bij een bepaald onderwerp
Actor	Gebruiker
Aannamen	De gebruiker bevindt zich op de pagina van een bepaald onderwerp
Beschrijving	(1) Het systeem laadt het eerste deel van de pagina (2) Het systeem checkt of de gebruiker Google Integration gebruikt (3) Het systeem maakt verbinding met Google Drive (4) Het systeem haalt de basis gegevens van de documenten van het onderwerp op (5) Het systeem toont in de Google Integration viewsection de documentenlijst (6) Het systeem registreert dit event in de activitylog (7) Het systeem laadt de rest van de pagina
Uitzonderingen	[geenGI] Voor dat account is er geen Google Integration geactiveerd, de viewsection wordt niet aangemaakt [verbindingsprobleem] Er kan geen connectie gemaakt worden met Google Drive, de viewsection wordt getoond met een foutmelding, de rest van de pagina laadt wel, error wordt geregistreerd in de errorlog
Resultaat	De pagina wordt geladen met documentenlijst

Naam	Document zoeken
Samenvatting	De gebruiker voert zoekactie uit en systeem toont resultaten
Actor	Gebruiker
Aannamen	De gebruiker bevindt zich op een pagina met een Google Integration viewsection
Beschrijving	(1) De gebruiker voert een zoekterm in, in het algemene zoekveld boven aan de pagina en klikt op de button “Zoeken” naast het zoekveld. (2) Het systeem legt verbinding met Google Drive (3) Het systeem vraagt om de basisgegevens van de documenten die voldoen aan de zoekterm (4) Het systeem registreert dit event in de activitylog (5) Het systeem zoekt in de documenten opgeslagen binnen Exact Online (6) Het systeem voegt de zoekresultaten samen tot één lijst (7) Het systeem toont in een nieuw scherm de documentenlijst die voldoet aan de zoekterm
Uitzonderingen	[verbindingsprobleem] Er kan geen connectie gemaakt worden met Google Drive, een foutmelding verschijnt, error wordt geregistreerd in de errorlog [geenresultaat] Een foutmelding verschijnt met melding dat er geen documenten voldoen aan de zoekterm, error wordt geregistreerd in de errorlog
Resultaat	De documentenlijst die voldoet aan de zoekterm in een nieuw scherm

Naam	Document inzien
Samenvatting	De gebruiker kan een document inzien
Actor	Gebruiker
Aannamen	De gebruiker bevindt zich op een pagina met een Google Integration viewsection
Beschrijving	(1) De gebruiker klikt op de naam van het document (2) Het systeem stuurt de gebruiker door naar Google Drive naar de preview van het document in een nieuw scherm (3) Het systeem registreert dit event in de activitylog
Uitzonderingen	[verbindingsprobleem] In het nieuwe scherm zal Google Drive aangeven dat er geen verbinding kan worden gemaakt en waarom.
Resultaat	De gebruiker ziet de inhoud van het gevraagde document

Naam	Document archiveren
Samenvatting	Een document wordt gearchiveerd in Exact Online
Actor	Gebruiker
Aannamen	De gebruiker bevindt zich op een pagina met een Google Integration viewsection
Beschrijving	(1) De gebruiker klikt op de checkbox voor het document dat gearchiveerd moet worden (2) De gebruiker klikt op de archiveer-link boven in de Google Integration viewsection (3) Het systeem legt verbinding met Google Drive (4) Het systeem download een kopie van het document van Google Drive (5) Het systeem controleert het document op uploadvoorwaarden (6) Het systeem upload het document naar Exact Online (7) Het systeem registreert dit event in de activitylog (8) Het systeem herlaadt de pagina
Uitzonderingen	[verbindingsprobleem] Er kan geen verbinding gemaakt worden met Google Drive, er verschijnt een foutmelding, error wordt geregistreerd in de errorlog
Resultaat	Het document is gearchiveerd naar Exact Online

Naam	Document wijzigen
Samenvatting	Een gebruiker wijzigt een document vanuit Exact Online
Actor	Gebruiker
Aannamen	De gebruiker bevindt zich op een pagina met een Google Integration viewsection De gebruiker heeft rechten om het document te wijzigen
Beschrijving	(1) De gebruiker klikt op de naam van het document (2) Het systeem stuurt de gebruiker door naar Google Drive naar de editpagina van het document in een nieuw scherm (3) Het systeem registreert dit event in de activitylog (4) De gebruiker kan het document wijzigen en opslaan met behulp van de bekende Google Drive functionaliteiten (5) De gebruiker refreshed de pagina van Exact Online na het opslaan van het document (6) Het systeem laadt de documentenlijst horend bij het paginaonderwerp
Uitzonderingen	[verbindingsprobleem] Er kan geen verbinding gemaakt worden met Google Drive, er verschijnt een foutmelding, error wordt geregistreerd in de errorlog
Resultaat	Het document is gewijzigd

Naam	Document aanmaken
Samenvatting	Een gebruiker maakt een document aan vanuit Exact Online
Actor	Gebruiker
Aannamen	De gebruiker bevindt zich op een pagina met een Google Integration viewsection
Beschrijving	(1) De gebruiker klikt op de add-link boven in de Google Integration viewsection waar het document bij moet komen (2) Het systeem controleert of er voor dit onderwerp al een map is aangemaakt in Google Drive (2a) Geen map: Het systeem maakt een map aan in Google Drive (2b) Wel map: Het systeem doet niets (3) Het systeem registreert dit event in de activitylog (4) Het systeem stuurt de gebruiker door naar Google Drive naar de pagina voor een nieuw document (5) De gebruiker voegt inhoud aan het document toe, en slaat het document op (6) De gebruiker refreshed de pagina van Exact Online (7) Het systeem haalt de documentenlijst op
Uitzonderingen	[verbindingsprobleem] Er kan geen verbinding gemaakt worden met Google Drive, er verschijnt een foutmelding, error wordt geregistreerd in de errorlog
Resultaat	Het document is aangemaakt

Naam	Statistieken inzien
Samenvatting	De systeembeheerder kan de statistieken van de module Google Integration inzien
Actor	Systeembeheerder
Aannamen	De systeembeheerder bevindt zich op de monitorpagina
Beschrijving	(1) De systeembeheerder gaat naar de tab module Google Integration (2) Het systeem haalt de statistieken op
Uitzonderingen	[geenstatistieken] Er zijn geen statistieken beschikbaar
Resultaat	De systeembeheerder kan de statistieken inzien

Naam	Map aanmaken in Google Drive
Samenvatting	Nieuwe map aanmaken bij een bestaand onderwerp in Google Drive
Actor	Gebruiker
Aannamen	De gebruiker bevindt zich op de project-/accountpagina waarvoor nog geen map bestaat in Google Drive.
Beschrijving	(1) De gebruiker klikt op map aanmaken in de viewsection. (2) Het systeem maakt een map aan in Google Drive voor dat onderwerp. (3) Het systeem refreshed de pagina.
Uitzonderingen	[verbindingsprobleem] Er kan geen verbinding gemaakt worden met Google Drive, er verschijnt een foutmelding, error wordt geregistreerd in de errorlog
Resultaat	Er is een map aangemaakt voor een bepaald project of account in Google Drive.

Naam	Map naam wijzigen in Google Drive
Samenvatting	Wanneer een project/account naam wordt aangepast in Exact Online, wordt deze verandering ook doorgevoerd in Google Drive
Actor	Gebruiker
Aannamen	De gebruiker bevindt zich op de project-/accountpagina waarvoor al een map bestaat in Google Drive.
Beschrijving	(1) De gebruiker klikt op edit boven het overzicht van de project-/accountpagina. (2) Het systeem laat de editpagina voor het project/account zien. (3) De gebruiker wijzigt de gegevens en klikt op save. (4) Het systeem verwerkt de wijzigingen in Exact Online. (5) Het systeem wijzigt de naam van de map horend bij het onderwerp in Google Drive. (6) Het systeem stuurt de gebruiker naar het overzicht van het project/account.
Uitzonderingen	[interne error] Exact Online kan de wijzigingen niet doorvoeren [verbindingsprobleem] Er kan geen verbinding gemaakt worden met Google Drive, er verschijnt een foutmelding, error wordt geregistreerd in de errorlog
Resultaat	De naamwijziging van een project/account is ook doorgevoerd in Google Drive.

Bibliography

- Göritzer, R. (2011). *Using software change impact analysis*. Klagenfurt, Austria(Oostenrijk).
- Hammer, E. (2009, December 26). *The OAuth1.0 Guide - Terminology*. Retrieved November 6, 2012, from Hueniverse: <http://hueniverse.com/oauth/guide/terminology/>
- Hammer, E. (2010, Mei 15). *Introducing OAuth2.0*. Retrieved November 6, 2012, from Hueniverse: <http://hueniverse.com/2010/05/introducing-oauth-2-0/>
- Hammer, E. (2010, Mei 11). *The OAuth1.0 Guide - Introduction* . Retrieved November 6, 2012, from Hueniverse: <http://hueniverse.com/oauth/guide/intro/>
- Microsoft patterns and practices. (2009). *Microsoft Application Architecture Guide 2nd edition*. United States: Microsoft Press.
- Rozanski, N., & Woods, E. (2005). *Software Systems Architecture*. Upper Saddle River: Pearson Education, Inc.
- Shalloway, A., & Trott, J. R. (2010). *Design Patterns Explained*. Boston: Pearson Education Inc.
- Swart, N. d. (2010). *Handboek Requirements*. Delft: Eburon Business.
- Warmer, J., & Kleppe, A. (2004). *Praktisch UML*. Amsterdam: Pearson Education.
- WikipediaNL. (2012, juni 20). *Bibliotheek (informatica)*. Retrieved oktober 29, 2012, from Wikipedia: <http://nl.wikipedia.org/wiki/Library>

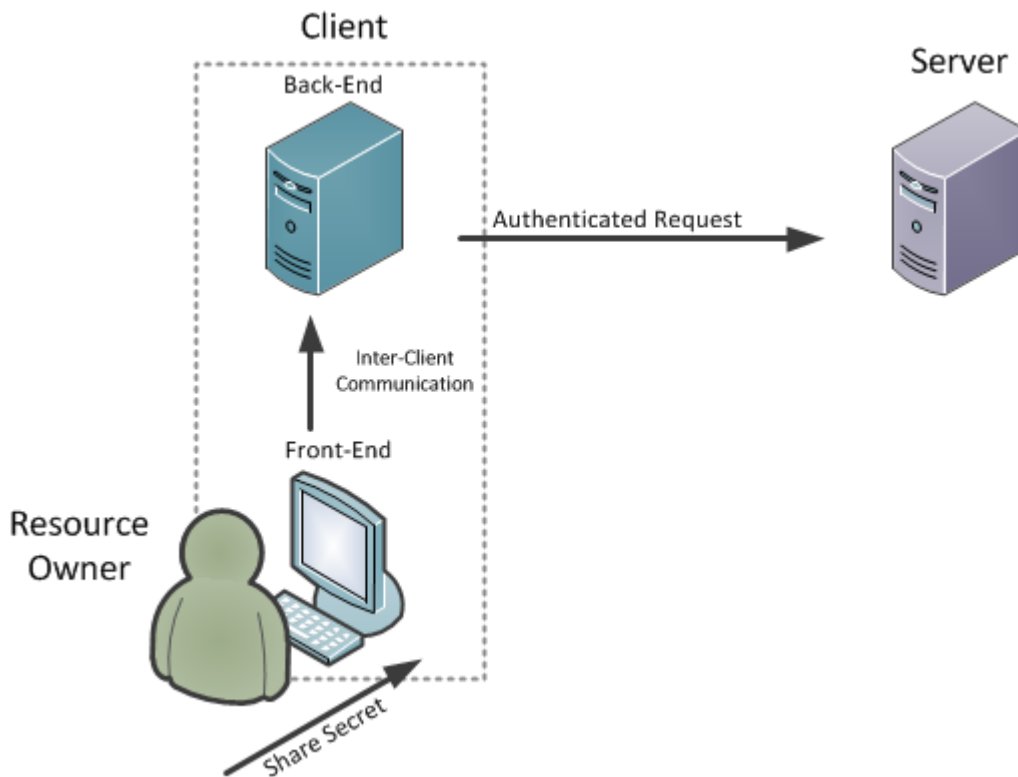
Bijlage 1: OAuth

1. Wat is OAuth?

Websites maken steeds vaker gebruik van gedistribueerde services en cloud-computing. Bijvoorbeeld het laten afdrukken bij een online fotostudio van je foto's die je bij een online service als picasa of Flickr hebt staan.

Het probleem dat ze met OAuth hebben opgelost is het in staat stellen van derden om de gegevens van een gebruiker op te halen zonder dat de gebruiker zijn gebruikersnaam en wachtwoord moet opgeven aan de derde partij. De gebruiker geeft de derde partij toestemming tot gelimiteerde toegang(tijd, scoop) tot zijn gegevens met behulp van OAuth.

Een voorbeeld: Een gebruiker (gegevenseigenaar) geeft toestemming aan een fotostudio(client) om zijn privéfoto's van een fotoalbum servicesite(server) op te halen. Het ophalen van deze foto's gebeurt zonder het delen van de gebruikersnaam en wachtwoord. In plaats daarvan vindt de authenticatie bij het fotoalbum servicesite(server) plaats en krijgt de fotostudio(client) een zogenaamde token voor gelimiteerde toegang tot de privéfoto's. (Hammer, The OAuth1.0 Guide - Introduction , 2010)



1-1 Web-Based-Client OAuth

2-legged, 3-legged, n-legged

Onder het aantal legs("poten") van een OAuth-verzoek verstaan we het aantal partijen dat betrokken is bij het OAuth-verzoek. In een simpele OAuth flow, zoals hierboven afgebeeld, hebben we te maken met een 3-legged, bestaande uit: een client, een server en een gegevenseigenaar. In het geval dat de client

ook de gegevenseigenaar is, is het een 2-legged constructie. (Hammer, The OAuth1.0 Guide - Terminology, 2009)

Credentials and tokens

OAuth onderscheidt drie soorten credentials: client credentials, tijdelijke credentials en token credentials. In de originele versie van de specificatie werden andere termen gebruikt, namelijk:

- Client credentials → consumer key en secret
- Tijdelijke credentials → request token en secret
- Token credentials → accesstoken en secret

De client credentials worden gebruikt om de client te authentifieren. Dit geeft de server toestemming om informatie te verzamelen over de clients die gebruik maken van de services, waardoor een aantal clients speciale behandeling kunnen krijgen (bijvoorbeeld: voorrang op andere, of meer informatie over clients die toegang zoeken tot beveiligde informatie). In sommige gevallen zijn de client credentials niet te vertrouwen en kunnen ze alleen worden gebruikt om informatie in te winnen.

Token credentials worden gebruikt in plaats van de gegevenseigenaar zijn gebruikersnaam en wachtwoord. In plaats van dat de gegevenseigenaar zijn credentials moet delen met de client, geeft hij/zij toestemming aan de server om een speciale soort van credentials, een token, aan de client te geven. Deze token representeert het toegangsniveau dat een gegevenseigenaar aan de client heeft gegeven. De client gebruikt deze token om toegang tot krijgen tot de beveiligde items/gegevens zonder dat de client het wachtwoord van de gegevenseigenaar nodig heeft.

Token credentials bevatten een tokenidentificeerder, vaak(maar niet altijd) is het een random combinatie van letters en nummers dat uniek en moeilijk te raden is, samen met een secret ter bescherming van de token tegen ongeautoriseerde partijen. Tokens hebben vaak een gelimiteerde scoop(welke gegevens/items heeft een client toegangstoestemming voor) en duur(hoe lang heeft een client toegang) en kan te allen tijde worden ingetrokken door de gegevenseigenaar.

Het OAuth autorisatieproces gebruikt ook een set van tijdelijke credentials welke worden gebruikt om een autorisatie verzoek te identificeren. Om verschillende soorten clients te kunnen bedienen(web-based, desktop, mobile etc.), bieden de tijdelijke credentials extra flexibiliteit en beveiliging.

Bij OAuth1.0, is het secretdeel van een set van credentials gedefinieerd als een gedeeld secret. Dit betekent dat zowel de client als de server beide toegang moeten hebben tot dezelfde secret.

De eerste versie van OAuth was gebaseerd op de protocollen Google Auth Sub, Yahoo BBAuth en Flickr API. Later kwam er een nieuwe versie uit OAuth1.0a die een zwakke plek in de beveiliging dichtte. (Hammer, The OAuth1.0 Guide - Terminology, 2009)

OAuth1.0a naar OAuth2.0

Na drie jaar ervaring met het OAuth protocol in de praktijk, was er genoeg ervaring opgedaan om het protocol te verbeteren. En wel op de volgende drie gebieden:

1. Authenticatie en handtekeningen
De meerderheid van de pogingen tot het implementeren van OAuth die mislukten, kwamen voort uit de complexiteit van de cryptografische requirements
2. Userexperience en alternatieve token uitgifte opties
OAuth bestaat uit twee hoofddelen, namelijk: een token verkrijgen door de gebruiker te vragen om toestemming voor bepaalde toegang en het gebruiken van tokens om toegang te hebben tot de beveiligde gegevens. De uitwisseling van deze tokens gebeurt volgens een bepaalde methode, ook wel flow genoemd. In het begin waren er drie flows: web-based applications, desktop clients en mobile or limited devices.
Deze werden in de tijd samengebracht tot één flow, wat er voor zorgde dat het goed werkte voor web-based applicaties, maar niet voor de andere implementaties.
3. Performance op schaal
Bij grotere serviceleveranciers kwam al snel naar voren dat OAuth niet goed te schalen is. Langlevende credentials zorgen dat de beveiliging minder is en zijn moeilijker te managen (en synchroniseren tussen data centers). Dit management probleem kwam ook voor wanneer er een aparte autorisatieserver en APIserver was.

Nieuw in OAuth2.0

1. Zes nieuwe flows:
 - User-agent flow → voor clients binnenin een user-agent (meestal een webbrowser)
 - Web Server flow → voor clients die deel uitmaken van een webserverapplicatie, die toegankelijk is via http aanvragen
 - Device Flow → geschikt voor clients die draaien op gelimiteerde apparaten, maar waar de eindgebruiker apart toegang heeft tot een browser op een andere computer of apparaat
 - Username and password flow → wordt gebruikt in het geval de gebruiker de client vertrouwd om zijn credentials te laten verwerken. Ook hier is het onwenselijk om de gebruikersnaam en het wachtwoord op te slaan. Deze flow is alleen toepasselijk bij een groot vertrouwen tussen de gebruiker en de client.
 - Client credentials flow → De client gebruikt zijn credentials om een access token te krijgen. Deze flow ondersteunt het zogeheten 2-legged scenario.
 - Assertion flow → de client presenteert een standaard voor het uitwisselen van authenticatie en autorisatie gegevens tussen twee partijen, zoals de SAML standaard, aan de autorisatieserver in ruil voor een accesstoken.
2. Bearer tokens
OAuth2.0 verzorgt een cryptografisch-vrije optie voor authenticatie welke gebaseerd is op een bestaande cookie authenticatie architectuur. In plaats van het sturen van getekende verzoeken met behulp van HMAC en token secrets, wordt de token zelf als secret gebruikt en over HTTPS

verstuurd. Dit maakt het mogelijk om API verzoeken te doen met behulp van cURL en andere simpele scripting tools.

3. Versimpelde handtekening

De ondertekeningssupport is aanzienlijk vereenvoudigd door het verwijderen van speciale parsing, encoding en het sorteren van parameters. Ook gebruikt het slechts één secret in plaats van twee.

4. Kortstondige(access) tokens met langdurige autorisaties

In plaats van het gebruik van een langdurige token, kan de server een kortstondige accesstoken en langdurige refreshtoken gebruiken. Dit geeft de clients de mogelijkheid om een nieuwe accesstoken te verkrijgen zonder dat de gebruiker daarvoor nodig is, maar de accesstokens blijven wel gelimiteerd.

5. Rolverdeling

OAuth2.0 scheidt de rollen:

- a) De rol van de autorisatie server die verantwoordelijk is voor het verkrijgen van de gebruikers autorisatie en het afhandelen van de tokens
- b) De rol voor het afhandelen van de API bevragingen op de gegevensserver

(Hammer, Introducing OAuth2.0, 2010)

2. OAuth2.0

In dit hoofdstuk wordt beschreven hoe OAuth2.0 protocol in elkaar zit en hoe het werkt. Voor het beschrijven van het OAuth2.0 protocol gebruik ik citaten uit een document die te vinden is op de OAuth Community website(zie bronnen). Ik heb deze citaten niet vertaald naar het Nederlands vanwege de vele Engelse termen met betrekking tot het OAuth2.0 protocol.

2.1.Roles

OAuth defines four roles:

1. Resource owner

An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user.

2. Resource server

The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens.

3. Client

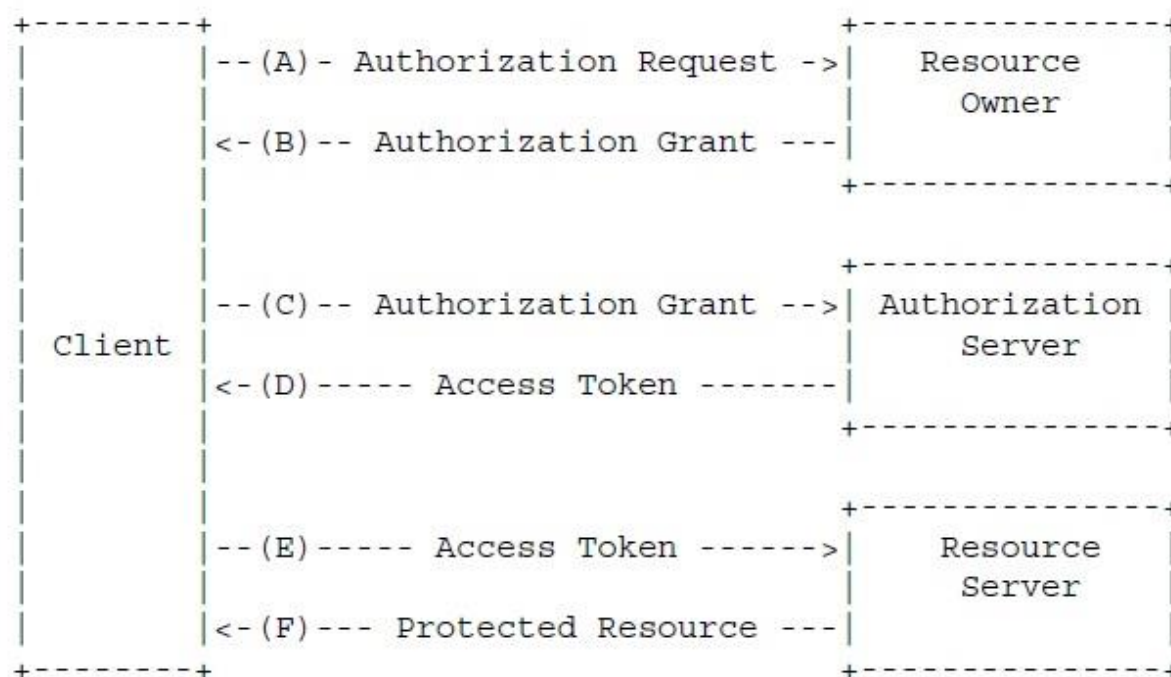
An application making protected resource requests on behalf of the resource owner and with its authorization. The term "client" does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices).

4. Authorization server

The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.

The interaction between the authorization server and resource server is beyond the scope of this specification. The authorization server may be the same server as the resource server or a separate entity. A single authorization server may issue access tokens accepted by multiple resource servers. (D. Hardt, 2012)

2.2.Protocol flow



2-1 Abstracte weergave OAuth2.0 flow (bron: (D. Hardt, 2012))

The abstract OAuth 2.0 flow illustrated in Figure 2-1 describes the interaction between the four roles and includes the following steps:

- (A) The client requests authorization from the resource owner. The authorization request can be made directly to the resource owner (as shown in figure 2-1), or preferably indirectly via the authorization server as an intermediary.
- (B) The client receives an authorization grant, which is a credential representing the resource owner's authorization, expressed using one of four grant types defined in this specification or using an extension grant type. The authorization grant type depends on the method used by the client to request authorization and the types supported by the authorization server.
- (C) The client requests an access token by authenticating with the authorization server and presenting the authorization grant.
- (D) The authorization server authenticates the client and validates the authorization grant, and if valid, issues an access token.
- (E) The client requests the protected resource from the resource server and authenticates by presenting the access token.
- (F) The resource server validates the access token, and if valid, serves the request.

The preferred method for the client to obtain an authorization grant from the resource owner (depicted in steps (A) and (B)) is to use the authorization server as an intermediary, which is illustrated in Figure 2-3 and described in paragraph 2.6. (D. Hardt, 2012)

2.3.Authorization Grant

An authorization grant is a credential representing the resource owner's authorization (to access its protected resources) used by the client to obtain an access token. This specification defines four grant types -- authorization code, implicit, resource owner password credentials, and client credentials -- as well as an extensibility mechanism for defining additional types. (D. Hardt, 2012)

2.3.1. Authorization Code

The authorization code is obtained by using an authorization server as an intermediary between the client and resource owner. Instead of requesting authorization directly from the resource owner, the client directs the resource owner to an authorization server (via its user-agent), which in turn directs the resource owner back to the client with the authorization code.

Before directing the resource owner back to the client with the authorization code, the authorization server authenticates the resource owner and obtains authorization. Because the resource owner only authenticates with the authorization server, the resource owner's credentials are never shared with the client.

The authorization code provides a few important security benefits, such as the ability to authenticate the client, as well as the transmission of the access token directly to the client without passing it through the resource owner's user-agent and potentially exposing it to others, including the resource owner. (D. Hardt, 2012)

2.3.2. Implicit

The implicit grant is a simplified authorization code flow optimized for clients implemented in a browser using a scripting language such as JavaScript. In the implicit flow, instead of issuing the client an authorization code, the client is issued an access token directly (as the result of the resource owner authorization). The grant type is implicit, as no intermediate credentials (such as an authorization code) are issued (and later used to obtain an access token).

When issuing an access token during the implicit grant flow, the authorization server does not authenticate the client. In some cases, the client identity can be verified via the redirection URI used to deliver the access token to the client. The access token may be exposed to the resource owner or other applications with access to the resource owner's user-agent.

Implicit grants improve the responsiveness and efficiency of some clients (such as a client implemented as an in-browser application), since it reduces the number of round trips required to obtain an access token. However, this convenience should be weighed against the security implications of using implicit grants, especially when the authorization code grant type is available. (D. Hardt, 2012)

2.3.3. Resource Owner Password Credentials

The resource owner password credentials (i.e., username and password) can be used directly as an authorization grant to obtain an access token. The credentials should only be used when there is a high

degree of trust between the resource owner and the client (e.g., the client is part of the device operating system or a highly privileged application), and when other authorization grant types are not available (such as an authorization code).

Even though this grant type requires direct client access to the resource owner credentials, the resource owner credentials are used for a single request and are exchanged for an access token. This grant type can eliminate the need for the client to store the resource owner credentials for future use, by exchanging the credentials with a long-lived access token or refresh token. (D. Hardt, 2012)

2.3.4. Client Credentials

The client credentials (or other forms of client authentication) can be used as an authorization grant when the authorization scope is limited to the protected resources under the control of the client, or to protected resources previously arranged with the authorization server. Client credentials are used as an authorization grant typically when the client is acting on its own behalf (the client is also the resource owner) or is requesting access to protected resources based on an authorization previously arranged with the authorization server. (D. Hardt, 2012)

2.4. Access Token

Access tokens are credentials used to access protected resources. An access token is a string representing an authorization issued to the client. The string is usually opaque to the client. Tokens represent specific scopes and durations of access, granted by the resource owner, and enforced by the resource server and authorization server.

The token may denote an identifier used to retrieve the authorization information or may self-contain the authorization information in a verifiable manner (i.e., a token string consisting of some data and a signature). Additional authentication credentials, which are beyond the scope of this specification, may be required in order for the client to use a token.

The access token provides an abstraction layer, replacing different authorization constructs (e.g., username and password) with a single token understood by the resource server. This abstraction enables issuing access tokens more restrictive than the authorization grant used to obtain them, as well as removing the resource server's need to understand a wide range of authentication methods.

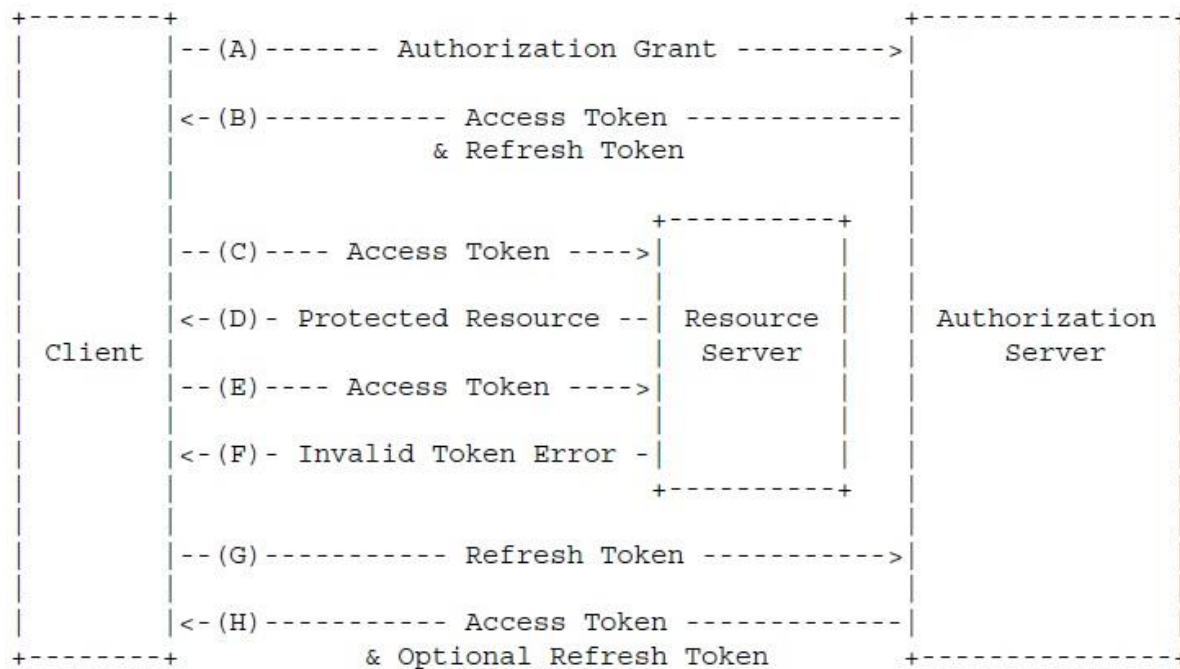
Access tokens can have different formats, structures, and methods of utilization (e.g., cryptographic properties) based on the resource server security requirements. Access token attributes and the methods used to access protected resources are beyond the scope of this specification. (D. Hardt, 2012)

2.5. Refresh Token

Refresh tokens are credentials used to obtain access tokens. Refresh tokens are issued to the client by the authorization server and are used to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope (access tokens may have a shorter lifetime and fewer permissions than authorized by the resource owner). Issuing a refresh token is optional at the discretion of the authorization server. If the

authorization server issues a refresh token, it is included when issuing an access token (i.e., step (D) in Figure 2-1).

A refresh token is a string representing the authorization granted to the client by the resource owner. The string is usually opaque to the client. The token denotes an identifier used to retrieve the authorization information. Unlike access tokens, refresh tokens are intended for use only with authorization servers and are never sent to resource servers.



2-2 Refreshing an expired access token (bron: (D. Hardt, 2012))

The flow illustrated in Figure 2-2 includes the following steps:

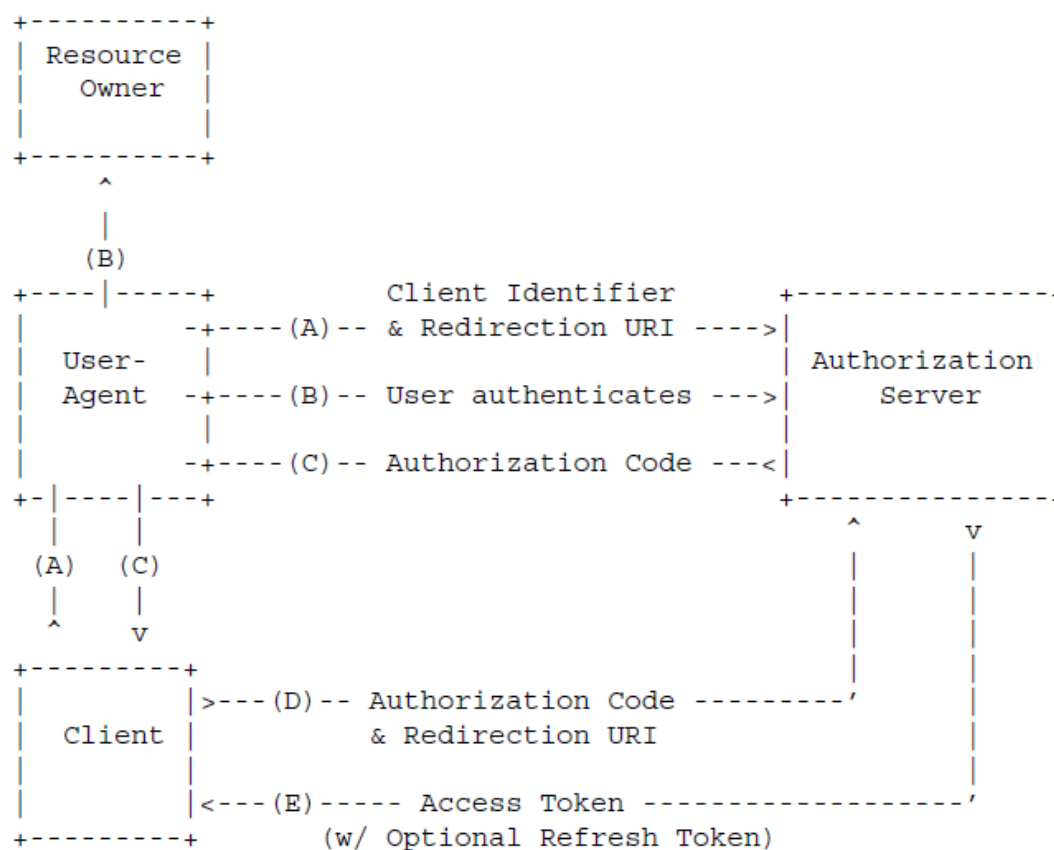
- (A) The client requests an access token by authenticating with the authorization server and presenting an authorization grant.
- (B) The authorization server authenticates the client and validates the authorization grant, and if valid, issues an access token and a refresh token.
- (C) The client makes a protected resource request to the resource server by presenting the access token.
- (D) The resource server validates the access token, and if valid, serves the request.
- (E) Steps (C) and (D) repeat until the access token expires. If the client knows the access token expired, it skips to step (G); otherwise, it makes another protected resource request.
- (F) Since the access token is invalid, the resource server returns an invalid token error.
- (G) The client requests a new access token by authenticating with the authorization server and presenting the refresh token. The client authentication requirements are based on the client type and on the authorization server policies.

(H) The authorization server authenticates the client and validates the refresh token, and if valid, issues a new access token (and, optionally, a new refresh token).

(D. Hardt, 2012)

2.6.Authorization Code Grant

The authorization code grant type is used to obtain both access tokens and refresh tokens and is optimized for confidential clients. Since this is a redirection-based flow, the client must be capable of interacting with the resource owner's user-agent (typically a web browser) and capable of receiving incoming requests (via redirection) from the authorization server.



Note: The lines illustrating steps (A), (B), and (C) are broken into two parts as they pass through the user-agent.

2-3 Authorization Code Flow (bron: (D. Hardt, 2012))

The flow illustrated in Figure 2-3 includes the following steps:

(A) The client initiates the flow by directing the resource owner's user-agent to the authorization endpoint. The client includes its client identifier, requested scope, local state, and a redirection URI to which the authorization server will send the user-agent back once access is granted (or denied).

- (B) The authorization server authenticates the resource owner (via the user-agent) and establishes whether the resource owner grants or denies the client's access request.
- (C) Assuming the resource owner grants access, the authorization server redirects the user-agent back to the client using the redirection URI provided earlier (in the request or during client registration). The redirection URI includes an authorization code and any local state provided by the client earlier.
- (D) The client requests an access token from the authorization server's token endpoint by including the authorization code received in the previous step. When making the request, the client authenticates with the authorization server. The client includes the redirection URI used to obtain the authorization code for verification.
- (E) The authorization server authenticates the client, validates the authorization code, and ensures that the redirection URI received matches the URI used to redirect the client in step (C). If valid, the authorization server responds back with an access token and, optionally, a refresh token.

Bronnen:

D. Hardt, E. (2012, Oktober). *Documentation - OAuth*. Opgeroepen op Oktober-Maart 2012-2013, van Website OAuth.net: <http://tools.ietf.org/html/rfc6749>

Hammer, E. (2009, December 26). *The OAuth1.0 Guide - Terminology*. Opgeroepen op November 6, 2012, van Hueniverse: <http://hueniverse.com/oauth/guide/terminology/>

Hammer, E. (2010, Mei 15). *Introducing OAuth2.0*. Opgeroepen op November 6, 2012, van Hueniverse: <http://hueniverse.com/2010/05/introducing-oauth-2-0/>

Hammer, E. (2010, Mei 11). *The OAuth1.0 Guide - Introduction*. Opgeroepen op November 6, 2012, van Hueniverse: <http://hueniverse.com/oauth/guide/intro/>

Bijlage H: Technical Documentation

Exact Cloud Solutions

Technical Documentation

Project External Document Management SaaS' Integration

Opdrachtgever: Ronald Dähne – Director Technology Cloud Solutions
Begeleider: Ad van der Hoeven – Senior Research Engineer
Datum: 21 Maart 2013
Versie: 2.0

= exact

And it all comes together.

Google

Microsoft®
V.NET

Auteur

E.M. Pekel
Student – 20064646
Informatica
Haagse Hogeschool – Den Haag

Begeleidend examiner: G.A. Mijnarends
Tweede examiner: Dhr. B. Kuiper

Preface

In this document the technical side of the External Document Management SaaS' Integration Project will be explained and illustrated. This document is initially for the developers who will work further on the integration module. The text is kept short and to the point, but gives all the information you will need as a developer.

First, the base will be explained and illustrated on the basis of the architecture, implementation of the factory pattern, the chosen folder structure at Google Drive, etc. After the base the functionalities of the project will be described.

Contents

<u>1.</u>	<u>INTRODUCTION</u>	<u>1</u>
1.1.	TECHNICAL BACKGROUND OF EXACT ONLINE	1
1.2.	ARCHITECTURE	1
1.2.1.	OLD LAYER ARCHITECTURE MODEL	1
1.2.2.	NEW LAYER ARCHITECTURE MODEL	4
1.3.	FACTORY PATTERN	6
1.4.	EXTERNSAASBASE	10
1.5.	HTTPSREQUEST	10
1.6.	SPLIT UP GOOGLECONNECTOR AND GOOGLESAA	13
1.7.	FOLDER STRUCTURE GOOGLE DRIVE	13
1.8.	BASE ACCOUNT	15
1.9.	STATISTICS	16
<u>2.</u>	<u>REGISTER TO MODULE</u>	<u>17</u>
2.1.	GENERAL INTEGRATION REGISTRATION	17
2.2.	RELATIONSHIPS TO OTHER REQUIREMENTS	18
2.3.	GOOGLE DRIVE INTEGRATION REGISTRATION	18
2.4.	TECHNICAL SIDE USER REGISTRATION	19
2.5.	RESULT SUCCESSFUL REGISTRATION GOOGLE DRIVE IMPLEMENTATION	21
<u>3.</u>	<u>DEACTIVATE INTEGRATION</u>	<u>22</u>
3.1.	GENERAL INTEGRATION DEACTIVATION	22
3.3.	STATE DIAGRAM STATUS INTEGRATION	23
3.4.	GOOGLE DRIVE INTEGRATION DEACTIVATION	23
3.5.	RESULT GOOGLE DRIVE INTEGRATION DEACTIVATION	23
<u>4.</u>	<u>ACTIVATE INTEGRATION</u>	<u>24</u>
4.1.	GENERAL INTEGRATION ACTIVATION	24
4.2.	RELATIONSHIPS TO OTHER REQUIREMENTS	24
4.3.	GOOGLE DRIVE INTEGRATION ACTIVATION	25
<u>5.</u>	<u>REVOKE REGISTRATION</u>	<u>26</u>
5.1.	GENERAL INTEGRATION REVOKE	26
5.2.	RELATIONSHIPS TO OTHER REQUIREMENTS	26
5.3.	GOOGLE DRIVE INTEGRATION REVOKE	27

6.	<u>DOCUMENT LIST</u>	28
6.1.	GENERAL INTEGRATION DOCUMENT LIST	28
6.2.	RELATIONSHIPS TO OTHER REQUIREMENTS	28
6.3.	GOOGLE INTEGRATION DOCUMENT LIST	29
6.4.	TECHNICAL SIDE DOCUMENT LIST	29
7.	<u>SEARCH FOR DOCUMENT(S)</u>	31
7.1.	GENERAL INTEGRATION SEARCH DOCUMENTS	31
7.2.	RELATIONSHIPS TO OTHER REQUIREMENTS	31
7.3.	GOOGLE INTEGRATION SEARCH DOCUMENTS	32
8.	<u>VIEW DOCUMENT CONTENT</u>	33
8.1.	GENERAL INTEGRATION VIEW DOCUMENT	33
8.2.	RELATIONSHIPS TO OTHER REQUIREMENTS	33
8.3.	GOOGLE INTEGRATION VIEW DOCUMENT	34
9.	<u>EDIT DOCUMENT CONTENT</u>	35
9.1.	GENERAL INTEGRATION EDIT DOCUMENT CONTENT	35
9.2.	RELATIONSHIPS TO OTHER REQUIREMENTS	35
9.3.	GOOGLE INTEGRATION EDIT DOCUMENT CONTENT	36
10.	<u>ADD NEW DOCUMENT</u>	37
10.1.	GENERAL INTEGRATION ADD NEW DOCUMENT	37
10.3.	GOOGLE INTEGRATION ADD NEW DOCUMENT	38
11.	<u>SHOW ALL DOCUMENTS</u>	39
11.1.	GENERAL INTEGRATION SHOW ALL DOCUMENTS	39
11.2.	RELATIONSHIPS TO OTHER REQUIREMENTS	39
11.3.	GOOGLE INTEGRATION SHOW ALL DOCUMENTS	39
	<u>LITERATURE</u>	40

1. Introduction

The aim for this chapter is to look at the technical side of the project. This chapter begins with the technical background of Exact Online in §1.1. After inventory Google Drive and Exact Online, the architecture of the old and the new desired situation where described. In §1.2.1 the old and in §1.2.2 the new layer architecture models are described. For more about the designed architecture, I recommend reading the architectural description of this project. In §1.2 it becomes clear that there're multiple implementations. To handle these implementations a factory pattern is used. The factory pattern makes it possible for the client to use all implementation like there the same and without knowing which implementation (Google Drive, Dropbox, etc.) is used on that moment.

All implementation have two kinds of functions in common, namely getting and setting of usersettings in the database and using self-composed HttpRequests. These have been put in their own class, namely ExternSaaSBase. In §1.4 a short list is given of the functions in this class.

The Google Drive implementation uses the self-composed HttpRequest for refreshing the access token, this is described in §1.5.

In first instants the Google Drive implementation couldn't be unittested. To make the Google Drive implementation unittestable, the Google drive implementation is split up in two part (§1.6).

In §1.7 I will explain why and how the folder structure is used in Google Drive for the Google Drive implementation.

Before Exact Online can use the Google Drive API it must be enabled for Exact Online. This is done with a regular Google account. How this works is explained in §1.8. With this account the statistics about the use of the integration can be monitored. This is described in §1.9.

1.1. Technical background of Exact Online

Technically seen, Exact Online is fully-based on the Microsoft ASP.NET (Active Server Pages .NET) framework, using the Visual Basic.NET (VB.NET) as the server-side programming language. On the client-side several JavaScript libraries are utilized for User Interface manipulation, for example Microsoft AJAX and jQuery. For data persistence, Microsoft's SQL Server is used. Furthermore, Exact Online officially supports multiple browsers, in particular Microsoft's Internet Explorer, Mozilla's Firefox and Apple's Safari. Regarding the size of Exact Online's codebase no exact number can be provided (in terms of lines of code), an estimation revealed that it consist approximately three million lines of source code. (Göritzer, 2011)

1.2. Architecture

In this section a description is giving about the old and new layer architecture model. For more details about the architecture and the decisions that were made regarding the architecture see document "Architectuur beschrijving - Project External Document Management SaaS' Integration".

1.2.1. Old layer architecture model

In the layer model (see figure 1-1) the old architecture situation is shown. The architecture can basically be described as a three-layered architecture consisting of a presentation layer, an application layer and a data access layer. One of the main characteristics of a layered architecture is that components of one layer only interact with components of the same layer, or components residing on the layer immediately

below. In that way, each layer is loosely coupled to only one different layer, which should result in a better maintainability of the application. (Göritzer, 2011) However Exact Online's design is not strictly layered, because there's a lot of legacy code and in most cases this legacy code isn't using the layered architecture as shown in figure 1-1 and described in this paragraph.

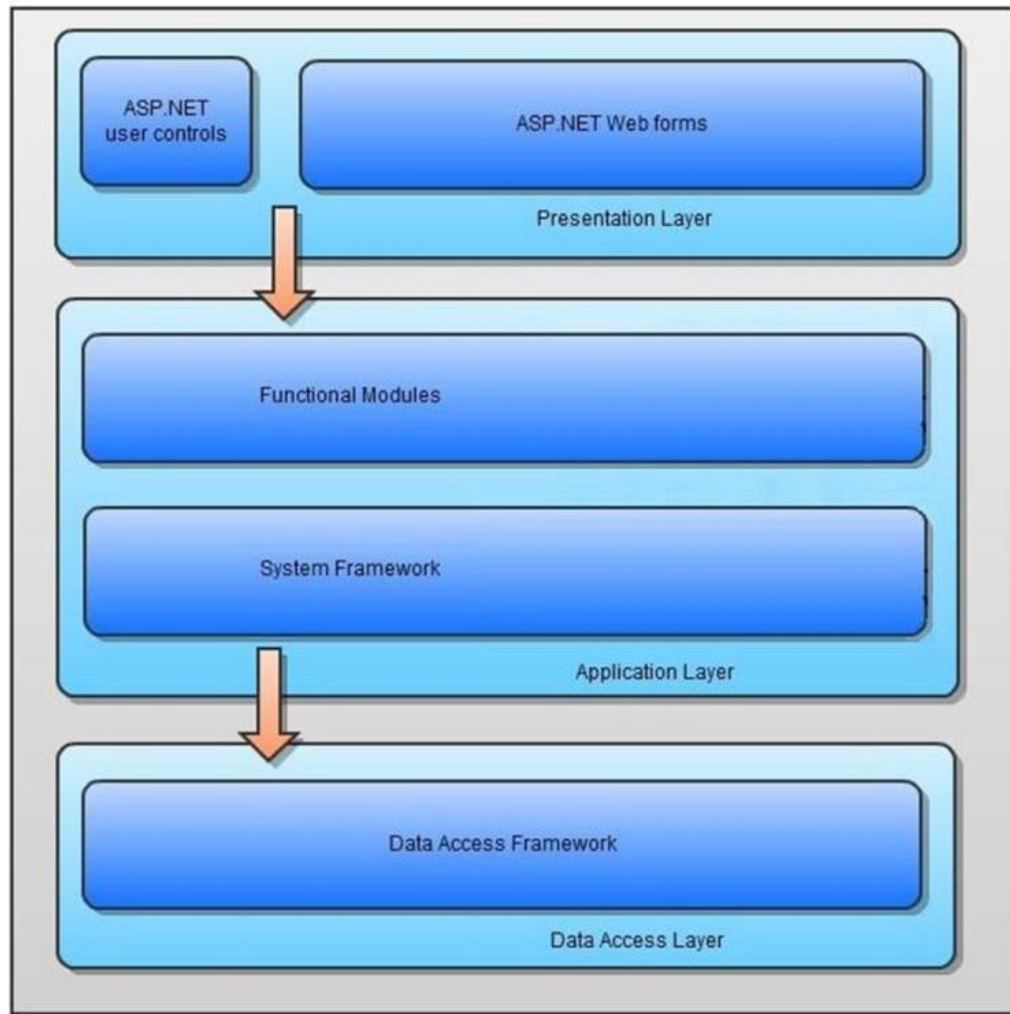
Presentation layer

The presentation layer comprises a large set of ASP.NET Web forms (contained in files with the extension .aspx) which represent the user interface of Exact Online. Besides static HTML markup and JavaScript source code, these files also contain markups defining server-side ASP.NET controls which are dynamically rendered when a user requests a page. In addition, an ASP.NET Web form of Exact Online usually comprises a considerable amount of business logic specifically tailored to the Web form. This concept differs from the code-behind model that Microsoft recommends, i.e., separating the business logic of an ASP.NET Web form from its user interface code by placing it in a different file.

This approach has been chosen for Exact Online due to several reasons. First, by having the business logic next to the user interface elements should help software engineers to get a complete view of the source code without having to deal with any possible differences between versions of the two files. Furthermore, using the code-behind model would imply that every file holding the business logic of an ASP.NET Web form would have to be compiled to a separate Dynamic Link Library (DLL).

Thus, the business logic would need to be deployed on the Web server in addition to the file containing the user interface code. This would mean shipping hundreds of additional DLLs (i.e., one for every page), and hence would significantly complicate the deployment process. Therefore, a considerable fraction of Exact Online's business logic is part of the presentation layer.

Additionally, ASP.NET user controls (contained in files with the extension .ascx) belong to the presentation layer which encapsulates page sections repeatedly used by the ASP.NET Web forms. (Göritzer, 2011)



1-1 Old layer architecture model of Exact Online (based on figure of (Göritzer, 2011))

Application layer

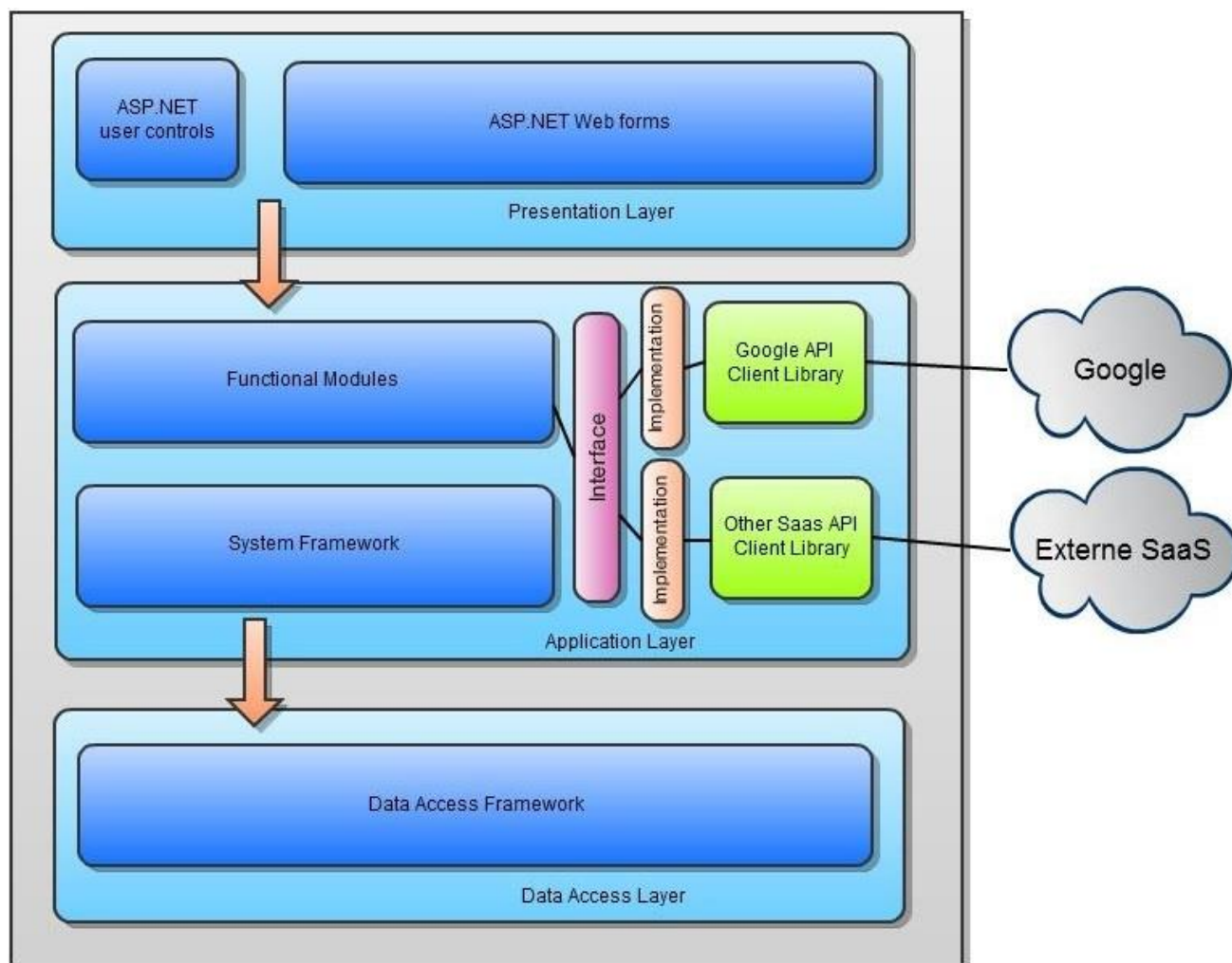
The application layer of Exact Online can basically be divided into the system framework and a set of functional modules. The system framework consists of a set of DLLs which embody the core functionality of the software and are used application-wide (e.g., by functional modules and ASP.NET Web forms). Exact Online incorporates various functional modules which can be independently subscribed by the customer (e.g., Payroll, Logistics or Human Resource Management). Next to the user interface implemented in ASP.NET Web forms together with page specific business logic, more general functionalities of these modules are implemented in several DLLs which are part of the application layer of Exact Online, i.e., the functional modules. (Göritzer, 2011)

Data access layer

The data access framework constitutes the data access layer within the logical architecture of Exact Online. It uses the Microsoft ADO.NET (ActiveX Data Objects .NET) Entity Framework to access the hosting and customer databases, and allows performing CRUD (create, read, update and delete) operations on them. (Göritzer, 2011)

1.2.2. New layer architecture model

In this section a brief description is giving of figure 1-2.



1-2 Layer model new architecture regarding External Document Management SaaS' Integration

In figure 1-2 the new architectural situation of Exact Online is shown. The blue blocks represent the current situation of Exact Online. The practice is that there is a lot of legacy code, that doesn't use the layer structure as intended. For parts regarding to this project and all future projects this layered structure will be maintained.

The green blocks are the libraries that are made available by the external SaaS solution. A library is a collection of code (functions / routines) that can be used by a program or function. The advantage of this is that there is need for writing code for general operations. The library contains functions who can execute these general operations (for example get information about the user' account).

The use of a library also has advantages when there are changes at the side of the external SaaS solution. In the event that the changes are such that a new library has to be used, the current library can be removed and the new implemented. In such a case only the code of the implementation need same changes. The implementations are orange blocks in the model next to the libraries.

In paragraph 1.2.1 the application layer is described. The reason for placing the libraries of the external SaaS' in the application layer is that the libraries of the System Framework en Functional modules also are placed here. For example a library like the one who is responsible for OAuth2.0 calls isn't developed by Exact Online, but is also placed in the application layer. In addition the libraries for the integration module are part of the integration module.

In front of the implementations (orange blocks) stands an interface (purple block). This interface describes the behavior of the implementations and is as it were a contract between the different implementations. An interface may only contain method signatures and constant declarations.

The implementations (orange blocks) implement the interface and must therefore implement all of the methods described in the interface. Per implementation the implemented functions' content differs. For example all implementations will have a function like GetDocuments, the content of this function in the Google Drive implementation will differ from the content of this function in the Dropbox implementation. However, the result namely a list of documents will be the same in both cases.

The implementation code has several functions. First, translating a call from the client (Exact Online) to a request that can be handled by the library (green blocks). The response on this request, for example a list of documents, must be converted in a common format (all implementations use the same format). This way the client can handle the data easily and in the same way for all the implementations. Briefly said, the implementation is the code that facilitates communication between the code of Exact Online (via the interface) and the library of the external SaaS solution.

To make it client code friendly to use this integration module and handling the implementations, a factory pattern is used (more about this see §1.3).

The piece of code in Exact Online that makes the integration fit in the current solution has two simple functions: call the function it needs of the implementation (the factory pattern takes care of which implementation is needed/asked) and present the result to the user. This means this piece of code only changes when there's need for changes on the visual side of the integration. For example: the creation date gets replaced by date of the last modification.

In the library there are also functions and routines for the communication with API of the external SaaS solution. The implementation takes care of calling the functions of the library it needs and translating the result into a data structure that is manageable the Exact Online code.

1.3.Factory pattern

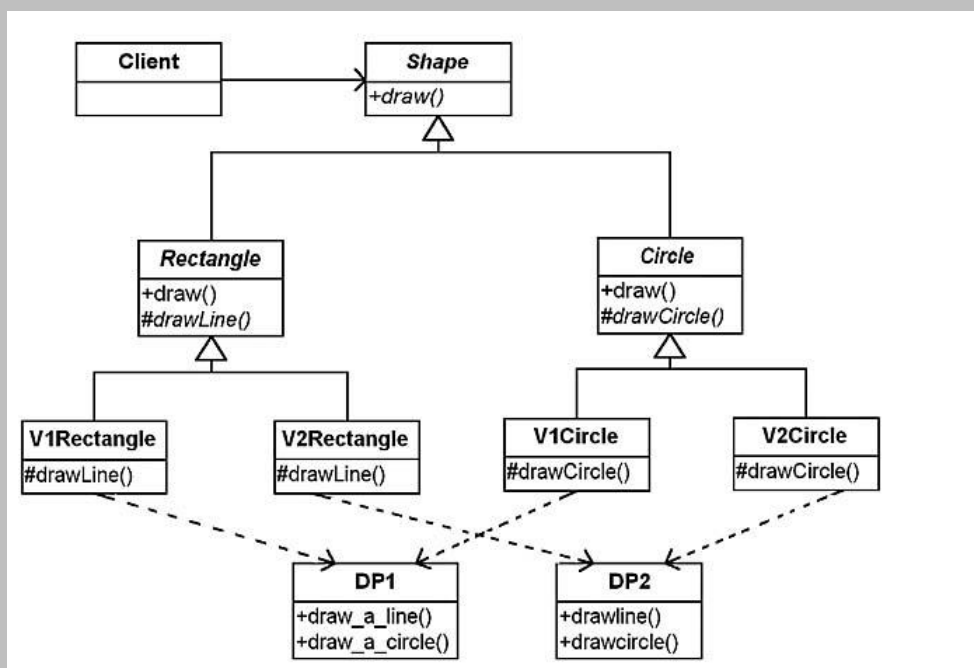
In this section a brief description is giving about what is a factory pattern and who is it used in this project.

In figure 1-3 is an overview of the intent of the factory pattern is, the kind of problem that can be solved with it, etc. like described in the book Design Patterns Explained.

1-3 Factory Pattern by “Design Patterns Explained” (Shalloway & Trott, 2010)

The Factory Pattern: Key Features

Intent	You want to have families or sets of objects for particular clients (or cases)
Problem	Families of related objects need to be instantiated.
Solution	Coordinates the creation of families of objects. Gives a way to take the rules of how to perform the instantiation out of the client object that is using these created objects.
Participants and collaborators	The Factory defines the interface for how to create each member of the family of objects required. Typically, each family is created by having its own unique ConcreteFactory .
Consequences	The pattern isolates the rules of which objects to use from the logic of how to use these objects.
Implementation	Define an abstract class that specifies which objects are to be made. Then implement one concrete class for each family. Tables or files can also be used to accomplish the same thing.



Bron: (Shalloway & Trott, 2010)

In object-oriented computer programming, a factory is an object for creating other objects. It is an abstraction of a constructor, and can be used to implement various allocation schemes.

The factory object has a method for every implementation it is capable of creating. These methods accept parameters defining how the implementation is created, and then return the created implementation. The factory determines the actual concrete type of implementation to be created, and it is here that the implementation is actually created. As the factory only returns an abstract pointer, the client(Exact Online) code does not know - and is not burdened by - the actual concrete type of the implementation which was just created. However, the type of a concrete implementation is known by the factory. In particular, this means: The client code has no knowledge whatsoever of the concrete type, not needing to include any header files or class declarations relating to the concrete type. The client code deals only with the abstract type. Objects of a concrete type are indeed created by the factory, but the client code accesses such objects only through their abstract interface.

Adding new implementations is done by modifying the factory code. This is significantly easier than modifying the client code to instantiate a new implementation, which would require changing every location in the code where the implementation is used.

The factory as it is used in this project is shown in figure 1-5. The code of Exact Online that makes calls to the integration module is presented by the client rectangle.

The first step for the client to use the factory and get the implementation for the integration functions is to call the GetImplementation function. A string of the externalSaaS type is needed for the factory to base the choice of which implementation it has to return on.

```
Private Function GetImplementation(externalSaaS As String) As IExternalSaaS
    Dim factory As New ExternalSaaSFactory
    Return factory.GetExternalSaaS(externalSaaS, env)
End Function
```

After an implementation is returned, the client can make calls to the interface. For example:

- implementation.GetFiles(DataLoader.DataSource.Name)
- implementation.SearchFiles(DataLoader.DataSource.Name, CStr(DocSearch.Value))

On this moment the client and the factory determine which implementation is needed on that moment. The reason the client and the factory work together on this, is that we wanted to leave the possibility of integrating more than one external SaaS (for example Google Drive and Dropbox) at a given time open. Another reason is that the client has a dedicated block (for showing the document lists) for every integration. See figure 1-4 for an example of a Google Drive block. If a user has an active Google Drive integration the block is visible. When the integration of Google Drive is not there or deactivated the block is not visible.

A possibility to exclude the client from the factory process is to say that a user may have a maximum of two implementations of external SaaS' at a given time. The factory then can create implementation1 and implementation2 and return the created implementations to the client.

The client can use implementation1 and implementation2 on the same matter as shown above.

= exact Administration : 1 - Customer Advanced Log off Help Search

Cockpits **General** **Financial** **CRM** **Sales** **Purchase**

Account: 3 - Groothandel sportfietsen Overview: Accounts > Account: 3 - Groothandel sportfietsen

Created: Customer Advanced 10-02-2012 11:11 Modified: Customer Advanced 10-02-2012 11:11

Edit **New** **Delete** **Deduplication** **Previous** **Next** **Close**

Groothandel sportfietsen

Postbus 9
2831AA Gouderak
Zuid-Holland, The Netherlands (Map / vCard)

Summary

There are no transactions.

The transaction overview shows you a summary of outstanding items, ongoing sales and purchases and open opportunities.

Activity stream

There are no activities.

The activity stream automatically shows you the last created tasks, appointments and opportunities.

Overview

General

Activities (0)

Documents (0)

Drive Docs

External References

Financial

Extra services

Drive Docs AddNew ShowAll

Search

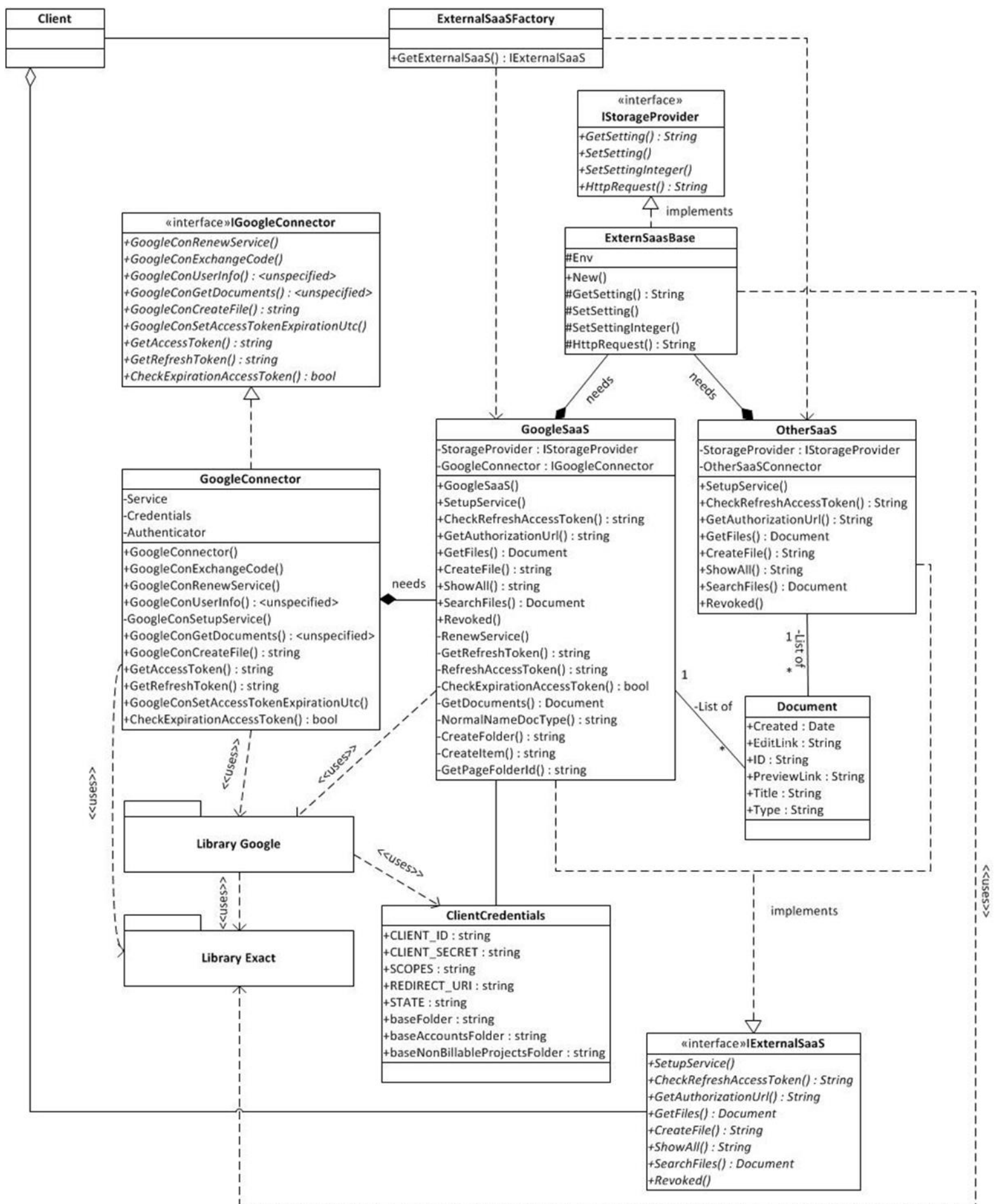
Subject	Type	Created
Sponsoroverleg voor wielerseizoen 2013	document	01/31/2013 11:44:06
2010-Q4	spreadsheet	01/31/2013 11:45:29
2010-Q3	spreadsheet	01/31/2013 11:45:16
2010-Q2	spreadsheet	01/31/2013 11:44:58
2010-Q1	spreadsheet	01/31/2013 11:44:41
PersoneelsOverleg 12-12-2012	document	01/31/2013 11:42:39
Groothandel doc	document	01/07/2013 15:22:56

1-4 Account card with Google Drive Docs block

The factory pattern returns an IExternalSaaS implementation. This means whether it is a Google Drive implementation or a Dropbox implementation etc., all the implementation have a standard list of functions.

The functions that are now available are:

- GetAuthorizationUrl (needed url for the registration of a user to a integration)
- SetupService (setup for the service that makes it possible to use the externalSaaS' library)
- CheckRefreshAccessToken (check if a accesstoken is expired, if so refresh the accesstoken)
- GetFiles (get document list for an account)
- CreateFile (add new document to the account folder at the externalSaaS)
- ShowAll (redirect the user to the account folder at the externalSaaS itself)
- SearchFiles (get a list of documents that correspond to the search query within the account folder)
- Revoked (when a user revokes the access of Exact Online at the external SaaS side or at Exact Online itself, the integration data will be delete from the database)



1-5 Class diagram of the whole integration module

This list of functions is expandable in the future, for example with the following functions:

- Archiving documents from and to Exact Online
- Account folder name changing when account name changes

All the implementations have two kinds of functions with the same content, namely the functions for manipulating the database and the function for handling HTTPrequests. These functions are put in a separate class, namely ExternSaaSBase(see figure 1-5). More about this in §1.5.

In figure 1-5 the Google Drive implementation “GoogleSaaS” has all the functions that the IExternalSaaS has. The private functions are the functions that are used to support the public functions. For example the public function “GetFiles”:

```
public List<Document> GetFiles(String account)
{
    String accountId = GetPageFolderId(account);
    if (accountId != "-1")
    {
        String q = "mimeType!='application/vnd.google-apps.folder' AND '" + accountId +
            "' in parents AND trashed = false";
        String fields = "items(alternateLink,createdDate,embedLink,id,mimeType,title)";
        return GetDocuments(q, fields);
    }
    else
        return new List<Document>();
}
```

The public function “GetFiles” uses two private functions, namely “GetPageFolderId” and “GetDocuments”. These functions are private and GoogleSaaS specific, meaning that Dropbox can have other private functions to support the public function “GetFiles”.

1.4.ExternSaaSBase

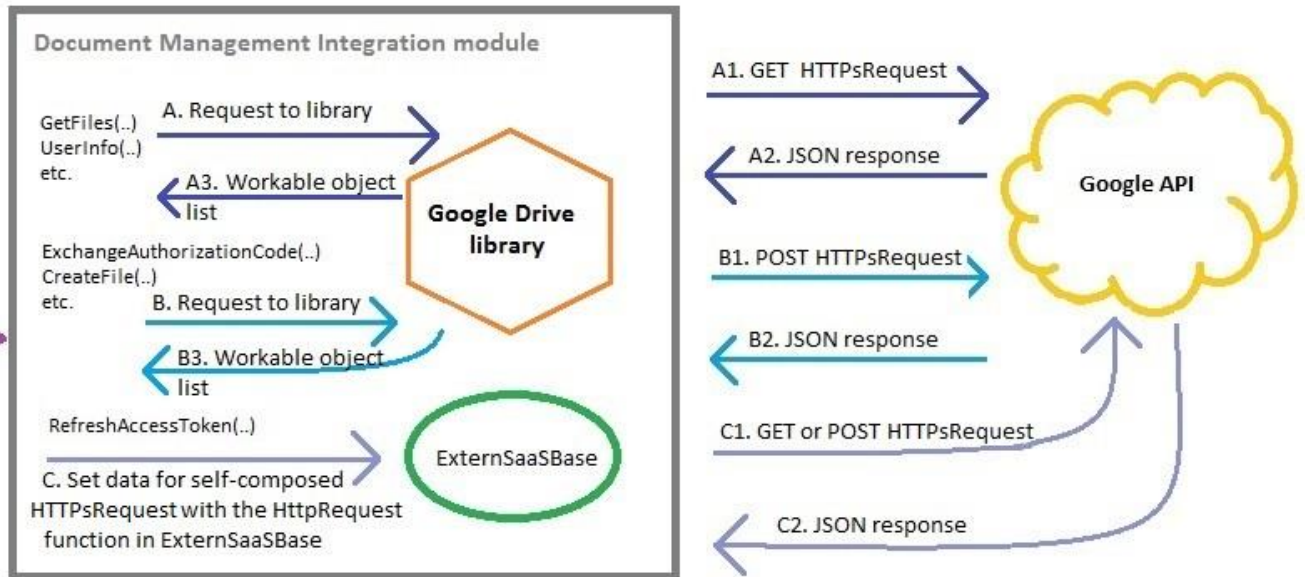
Every implementation uses the “ExternSaaSBase” for the getting and setting of usersettings from/to the database and for using self-composed HttpRequests. The ExternSaaSBase has 5 functions:

- New(create an instance of ExternSaaSBase)
- GetSetting(get usersetting from database as a string)
- SetSetting(set usersetting in database as string)
- SetSettingInteger(set usersetting in database as integer)
- HttpRequest(base function for a HTTPRequest, same input like: uri, parameters, host and method) more about this in §1.6.

1.5.HTTPsRequest

In this paragraph the HTTPSrequest used in the project is described. To fully understand this paragraph it is recommend to know what OAuth2.0 and tokens are. OAuth2.0 and tokens are explained in appendix 1 of the architectural description (“Architectuur beschrijving - Project External Document Management SaaS’ Integration”).

The library of Google Drive (and most other document management SaaS') is/are extensive. In the case of Google Drive the only function where a self-writing HTTPsRequest was needed, is the one to refresh the access token. For this HTTPsRequest (and all other HTTPsRequest to the Google API) the OAuth2.0 protocol has to be followed.



1-5 The types of HTTPsRequests with the Document Management Integration module

In figure 1-5 the types of HTTPsRequest are shown. Type A. are the GET HTTPsRequests which are send by the Google Drive library. For example the function GetFiles calls a function for getting the document list. The library then makes an HTTPsRequest based on the given data and sends the GET HTTPsRequest to the Google API (A1). The Google API responses with an object (file data) list in the format JSON¹ (A2). The Google Drive library deserializes this list from JSON to an object list that can easier be used in the Exact Online software (A3).

The second type (B) are the POST HTTPsRequest which also are send by the Google Drive library. For example the function CreateFile calls a function for creating a new document or new folder at the Google Drive account of the user. The library then makes an HttpRequest based on the given data and sends the POST HTTPsRequest to the Google API(B1). The Google API responses with the file data in the format JSON (B2). The Google Drive deserializes this list from JSON to an object that can easier be used by the Exact Online software.

And the last type (C) is/are the self-composed HTTPsRequests which are send by the ExternSaaSBase class. The self-composed HTTPsRequest can have as method GET or POST. The function for refreshing the access token uses the self-composed HTTPsRequest. The first step is to gather and set the required data(C). The ExternSaaSBase puts the HTTPsRequest together with the given data and sends it to the Google API (C1).

¹ JSON is a text-based open standard designed for human-readable data interchange.

The Google API responses with a new access token and information like the expiration date/time of the access token in JSON format(C2). The ExternSaaSBase gives this JSON data back to the RefreshAccessToken function without deserializing it. The RefreshAccessToken function is responsible for this.

For all above HTTPsRequests the OAuth2.0 protocol has to be followed. This means that the HTTPsRequest must contain the method, host, request-uri, HTTP/{version} and most importantly something for authorization. The last one can be two things: if you already have an access token (that isn't expired), the access token will be used (see figure 1-6 for an example). Or if you request an access token, you will need to send the ClientId, ClientSecret, refreshtoken, redirecturi and granttype next to the list above (see figure 1-7 for an example).

```
GET /oauth2/v1/userinfo HTTP/1.1
Authorization: Bearer 1/fFBGRNJrulFQd44AzqT3Zg
Host: googleapis.com
```

1-6 Example HTTPsRequest with access token as authorization to the Google API

```
POST /o/oauth2/token HTTP/1.1
Host: accounts.google.com
Content-Type: application/x-www-form-urlencoded

code=4/P7q7W91a-oMsCeLvIaQm6bTrgtp7&
client_id=8819981768.apps.googleusercontent.com&
client_secret={client_secret}&
redirect_uri=https://oauth2-login-demo.appspot.com/code&
grant_type=authorization_code
```

1-7 Example HTTPsRequest with the refreshtoken and client details as authorization to the Google API

1.6.Split up GoogleConnector and GoogleSaaS

The implementation of Google Drive is split in two parts:

- “GoogleConnector”: Handles the communication with the Google library. The Google library handles the communication with Google Drive.
- “GoogleSaaS”: Handling the client request, controls the GoogleConnector and translates the results of request to a format that the client can handle.

This split has as purpose to make GoogleSaaS unit testable. The GoogleConnector can now be stubbed² and/or mocked³.

1.7.Folder structure Google Drive

For the Google Drive folders and documents it wasn't possible to use same kind of tag, that the user could apply to a folder and document himself (without Exact Online). The alternative is using a folder structure. How the folder structure is used will be explained in this section.

When a user registers to the Google Drive implementation the three base folders will be created in the Google Drive account of the user. The three base folders are “ExactOnlineApp”, “Exact-Accounts” (within the “ExactOnlineApp” folder) and “Exact-NonBillableProjects” (within the “ExactOnlineApp” folder). The folder structure is shown in figure 1-8.

In Exact Online in the right corner of the Google Drive documents block is the button “AddNew” (for more details on this functions see chapter 10). When a user uses this function to add a new document, the system checks if the account folder already exists in the Google Drive account of the user. If this isn't the case the folder gets created in the base folder “Exact-Accounts”. Within the account folder another folder with the name “Projects” is created for the project folders and documents. (See figure 1-8)

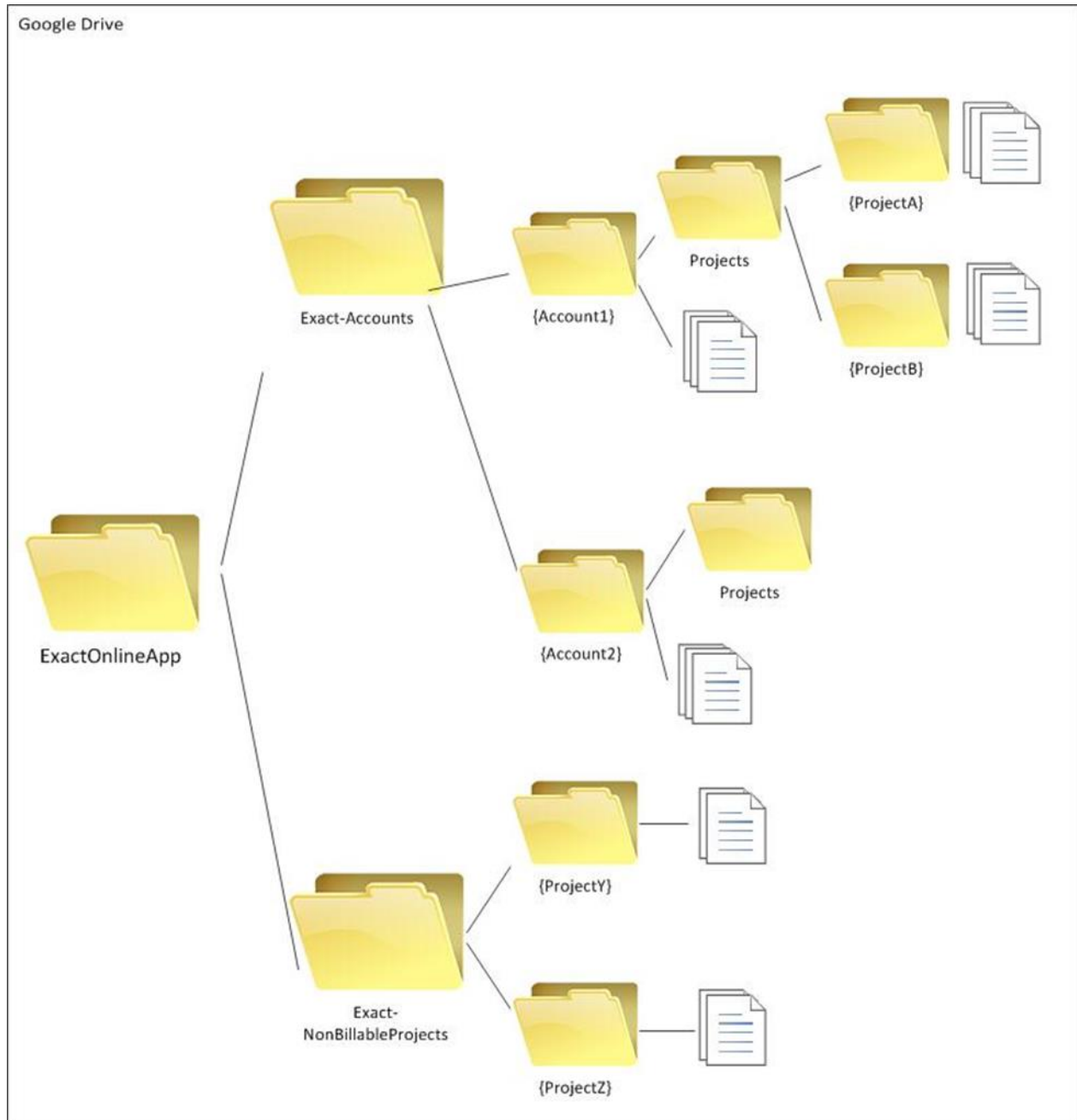
A user can also create an account folder him-/herself by creating it directly in his/hers Google Drive account. When the user does this, the user shouldn't modify the base folders. And the account folder name is must be exactly the same as the account name in Exact Online. When the account folder is created successfully the user can fill the folder with documents. These documents will be shown in Exact Online after a couple of minutes.

The folder “Exact-NonBillableProjects” is for the internal projects. These projects are not linked to an account, but are stand-alone projects. The project name within exact Online is the same as the folder name in the Google Drive account.

When a user revokes the access of Exact Online to his/hers Google Drive account, all folders and documents will subsist in the Google Drive account. In the future the user can register him-/herself again and use the still existing folder structure.

² Stubbing a method is all about replacing the method with code that returns a specified result (or in some cases raises a specified exception).

³ Mocking a method is all about asserting that a method has been called (perhaps with particular parameters).



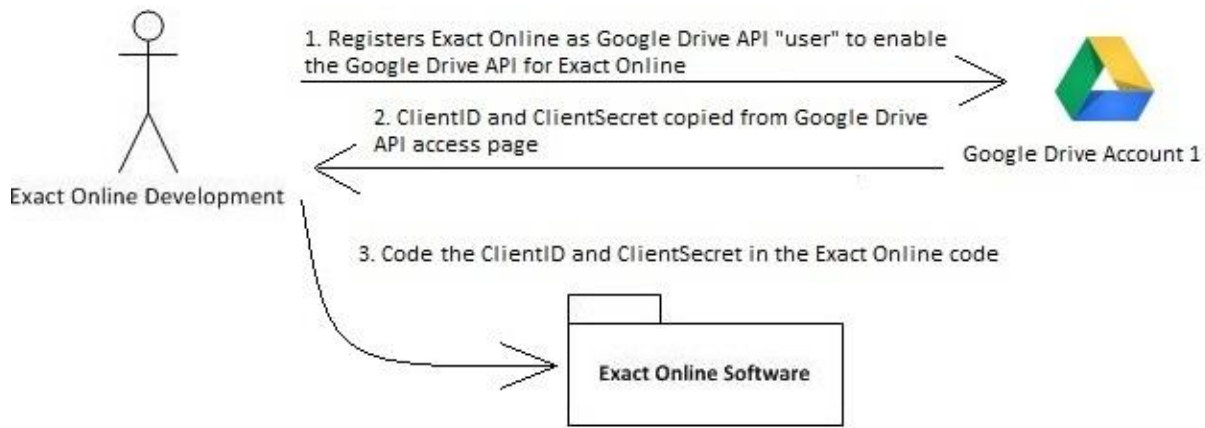
1-8 Folder structure for Google Drive

1.8.Base account

To get started with the Google Drive API, it must first be enabled. To enable the Google Drive API we need a regular Google account. This Google account will be managed by the development team, to avoid problems when employees are leaving the company how were using their personal Google account.

This Google account will be only used to enable the Google Drive API for Exact Online and look into the statistics. With this account we don't have access to the accounts of the Google Drive implementation users and there will not be any document storage except for testing purposes.

The process of enabling the Google Drive API is shown in figure 1-9.



1-9 Enabling the Google Drive API for Exact Online

Step 1: Login to the Google account en go to <https://code.google.com/apis/console/>. In the menu choose for the tab "API Access" and register the Exact Online as Google Drive API "user" to enable the Google Drive API for Exact Online.

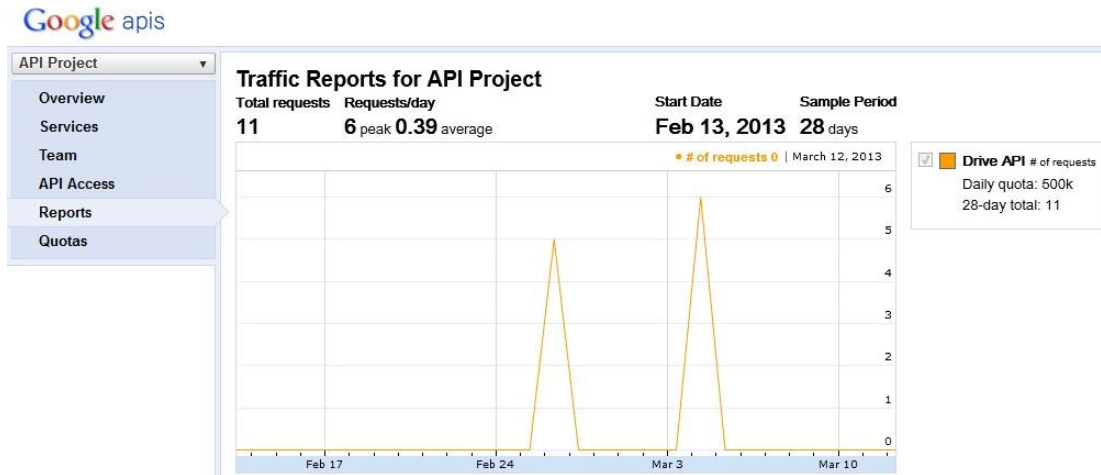
Step 2: After the registration (step 1) you can copy the ClientID and ClientSecret under the heading "Client ID for web applications". These can be found at the same page as step 1.

Step 3: The ClientID and ClientSecret now need to be coded in the Exact Online integration module as constants.

1.9.Statistics

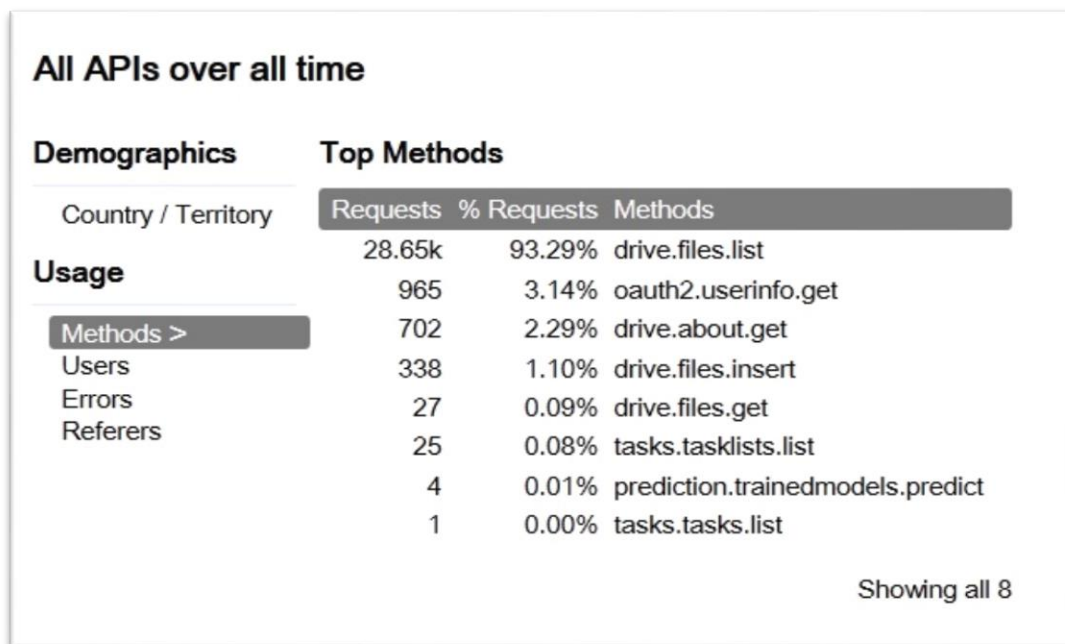
The Google account, that is managed by the development team and is used to enable the Google Drive API for Exact Online, can be used to show the statistics about the use of the Google Drive implementation in Exact Online. The statistics presented in two ways:

1. A traffic report for API project (in our case only the Drive API)



1-10 Traffic report about usage Google Drive API

2. All APIs over all time in numbers and percentages



1-11 All APIs over all time in numbers and percentages

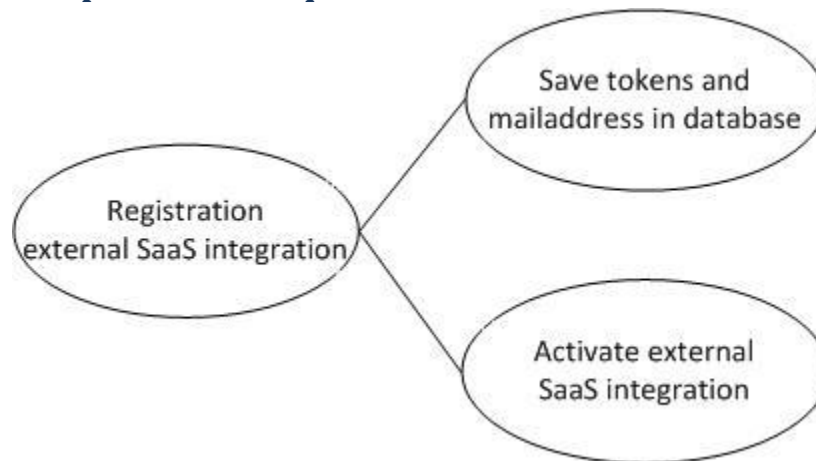
In figure 1-11 you can see that the usage can be shown filtered on methods, users, errors and referrers. This statistics can be used to see how much the Google Drive implementation is used and what the users use the most.

2. Register to module

2.1. General integration registration

Name	Registration Integration
Summary	The user registers him-/herself for an external SaaS integration
Actor	User
Assumptions	• The user has rights to register an external SaaS integration • The user is at the page General → User settings • The ExternalSaaSStatus for this user is nothing or nonRegistered
Description	(1) The user clicks on "Register the {SaaSname} integration" (2) The system creates the authorization url (3) The system redirect the user to the OAuth-page of the external SaaS with the authorization url (4) The user logs in on his SaaS Account (5) The external SaaS validates the login name and password combination (6) The External SaaS shows a list with areas Exact Online wants access to: a. View basic information about your account b. View and manage the files and documents in your account c. View your email address d. Perform these operations even when I'm not using the application (7) The user clicks on the button "Allow" (8) The external SaaS sends the user back to Exact Online (9) The system retrieves the authorization code from the redirecturl (10) The system exchange the authorization code for an access code and a refreshtoken (11)The system creates the folders ExactOnlineApp(in root), Exact-Account(in ExactOnlineApp) and Exact-NonBillableProjects(in ExactOnlineApp) (12) The system registers the registration and activation of the external SaaS Integration in the database under "{ExternalSaaS}Status" (13)The system registers the mail address under "{ExternalSaaS}UserId" and the tokens under "{ExternalSaaS}AccessToken" and "{ExternalSaaS}RefreshToken" in the database
Exceptions	[loginerror] The login of the user in his external SaaS account is not valid [noAccessGranted] The user doesn't except that Exact Online gets access to the user's documents, files and user information. The user will be redirect to Exact Online. [noConnection] There isn't a connection with the external SaaS, the viewsection will be shown with an error message, the rest of the page will be shown normally, the error will be registered in the errorlog
Result	The external SaaS Integration is activated and the credentials of the user are registered in the database.

2.2.Relationships to other requirements

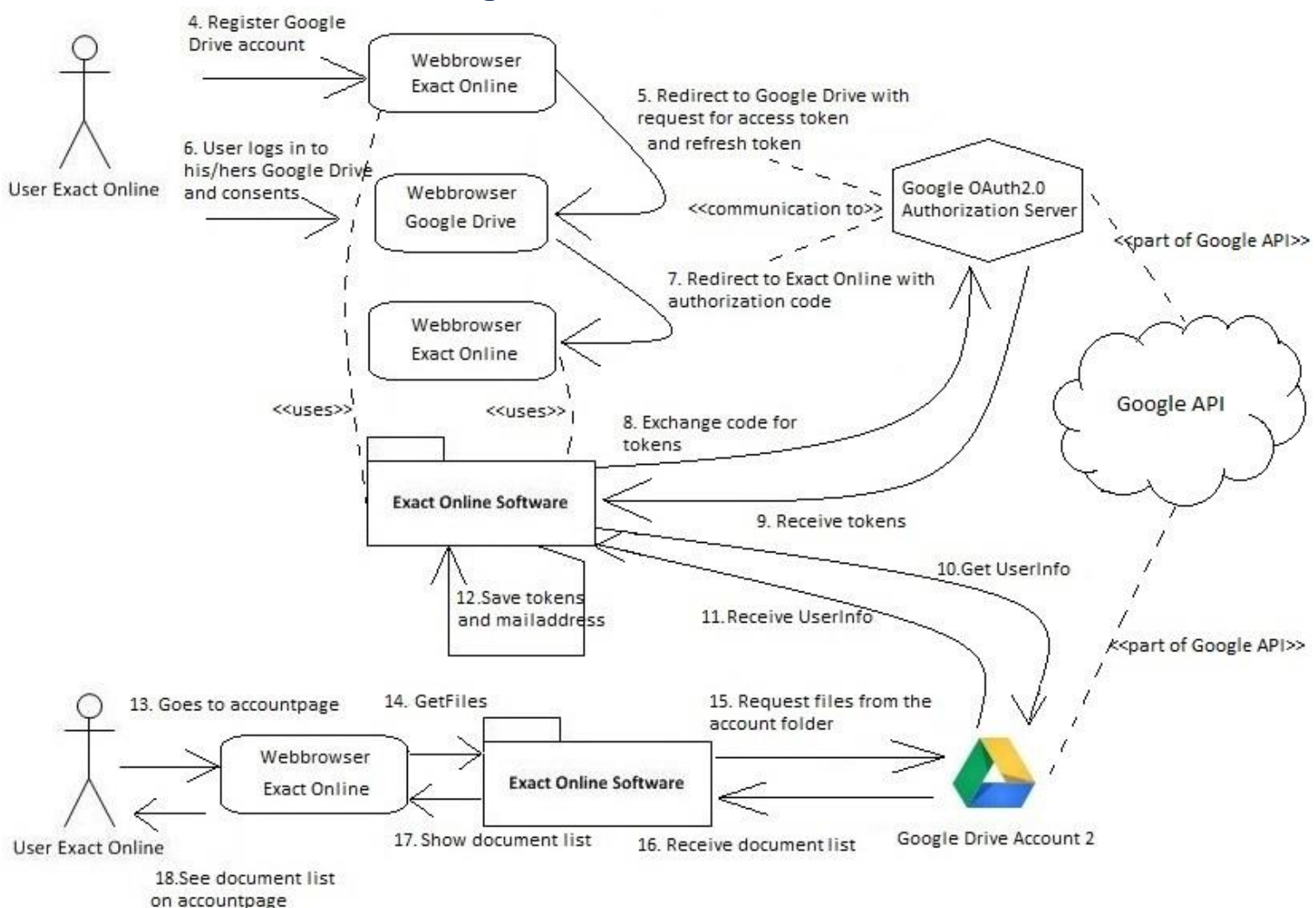


2.3.Google Drive integration registration

Name	Registration Google Drive Integration
Summary	The user registers himself for the Google Drive
Actor	User
Assumptions	• The user has rights to register to the Google Drive integration • The user is at the page General→User settings • The GoogleDriveStatus for this user is nothing or nonRegistered
Description	(1) The user clicks on “Register the Google Drive integration” (2) The system creates the authorization url (3) The system redirect the user to the OAuth-page of Google Drive with the authorization url (4) The user logs in on his Google Account (5) Google Drive validates the login name and password combination (6) Google Drive shows a list with areas Exact Online wants access to: a. View basic information about your account b. View and manage the files and documents in your Google Drive c. View your email address d. Perform these operations when I’m not using the application (7) The user clicks on the button “Allow” (8) Google Drive sends the user back to Exact Online (9) The system retrieves the authorization code from the redirecturl (10) The system exchange the authorization code for an access code and a refreshtoken (11)The system creates the folders ExactOnlineApp(in root), Exact-Account(in ExactOnlineApp) and Exact-NonBillableProjects(in ExactOnlineApp) (12) The system registers the registration and activation of the Google Drive Integration in the database (13)The system registers the GoogleUserId, GoogleAccessToken and GoogleRefreshToken in the database

Exceptions	[loginerror] The login of the user in his Google account is not valid [noAccessGranted] The user doesn't expect that Exact Online gets access to the user's documents, files and user information. The user will be redirect to Exact Online. [noConnection] There isn't a connection with Google, the viewsection will be shown with an error message, the rest of the page will be shown normally, the error will be registered in the errorlog
Result	The Google Integration is activated and the credentials of the user are registered in the database.

2.4. Technical side user registration



2-1 Continuation on figure 1-9, technical side of user registration

How the registration of the Google Drive implementation works is shown in figure 2-1. Each step will be explained further below. These steps correspond with the use case of paragraph 2.3. The steps 13 through 18 aren't part of the registration and will be explained in paragraph 6.4.

Step 4: The user clicks on “Register the Google Drive integration”

Step 5: Exact Online makes the authorization url and redirect the user to this url. The url goes to the Google Drive login screen.

Step 6: The user logs in on his/her Google Drive account. After a successful login, the user sees a (Google) screen with a list with the data where Exact Online wants access to on the users Google Drive account. The user clicks on “Allow” to grant Exact Online this access.

Step 7: Google redirects the user to Exact Online with an authorization code in the url.

Step 8: Exact Online catches the authorization code from the url and uses this code to request (self-composed HTTPsRequest) the Google OAuth2.0 Authorization Server for a access token and a refresh token.

Step 9: Exact Online receives the requested access token and refresh token.

Step 10: Exact Online sets up a service which enables the library to communicate with the Google API with authorized requests. The first request with this service is getting information of the user, in particular the mail address.

Step 11: Exact Online receives the user information.

Step 12: Exact Online saves the mail address, access token and refresh token in the Exact Online database and activates the Google Drive implementation for this user, by editing the GoogleDriveStatus in the Exact Online database.

2.5.Result successful registration Google Drive implementation

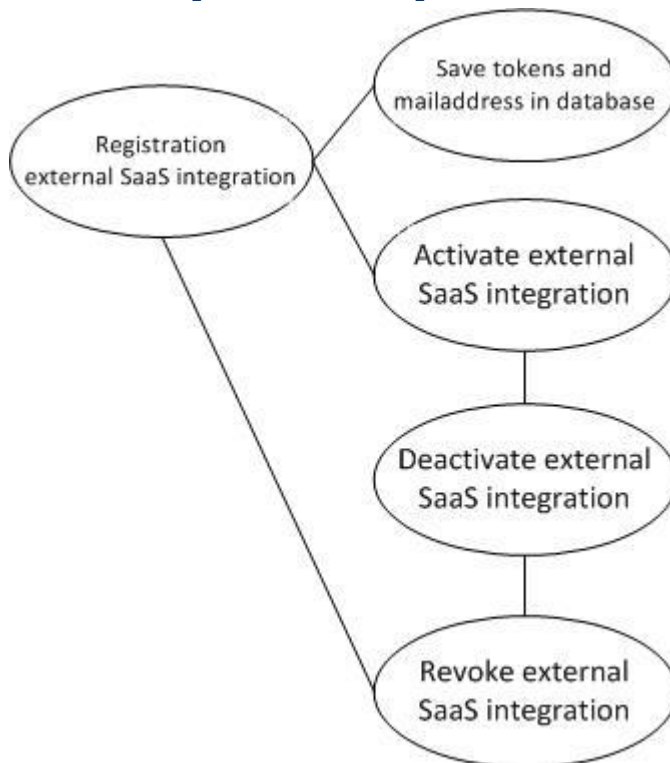
= exact		Cockpits	General	Financial	CRM	Sales	Purchase	Administration : 1 - Customer Advanced
Entry based on year/month instead of financial year/period								☐
Graph								
Animation								☒
Palette								Charcoal
Export								
Which office software are you using?								Microsoft Office
What to export								All pages
CSV export delimiter								Tab
Excel export with style formatting								☒
Dialog								
Show errors in popup screen								☒
Show options dialog when exporting or printing								☐
My Outlook								
Exchange server								☒ Microsoft Office 365 ☐ Microsoft Exchange 2010
User name								
Password								
Display: number of e-mail messages								3 <i>Maximum: 30</i>
Display: number of appointments								5 <i>Maximum: 30</i>
External SaaS Integration								
Google Integration status								active Deactivate Google Drive mailaddress@gmail.com
Dropbox Integration status								not yet available

3. Deactivate integration

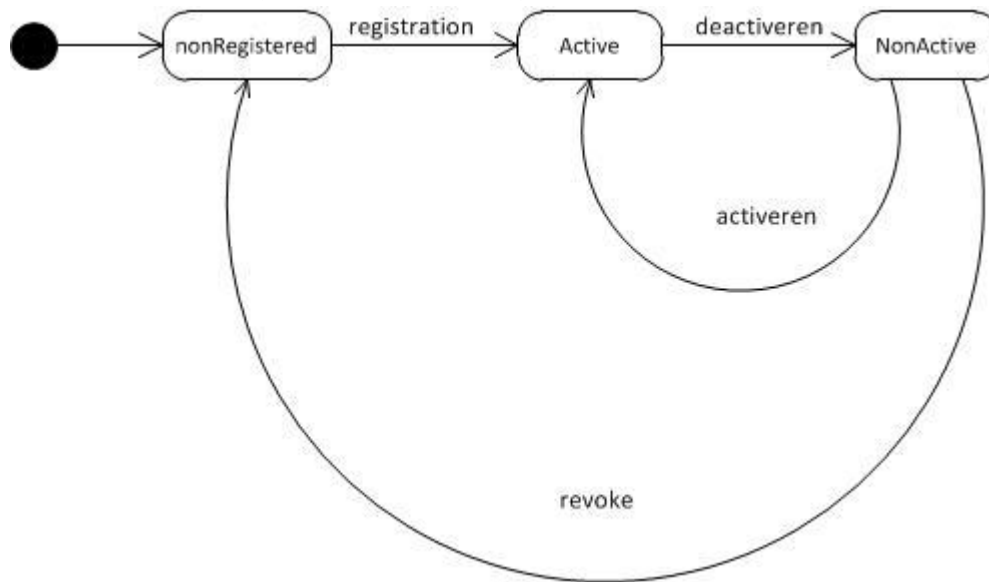
3.1.General integration deactivation

Name	Deactivate external SaaS integration
Summary	The user deactivates the external SaaS integration
Actor	User
Assumptions	The user is at the page General→User settings The ExternalSaaSStatus for this user is active
Description	(1) The user clicks on the button “Deactivate {SaaSname}” (2) The system updates the setting “{SaaSname}Status” in the database from “active” to “nonActive”
Exceptions	[dbfailure] Can’t update SaaS integration status, nothing changes
Result	The external SaaS integration is deactivated.

3.2.Relationships to other requirements



3.3.State diagram status integration



3.4.Google Drive integration deactivation

Name	Deactivate Google Drive integration
Summary	The user deactivates the Google Integration
Actor	User
Assumptions	The user is at the page General → User settings The GoogleDriveStatus for this user is active
Description	(1) The user clicks on the button “Deactivate Google Drive” (2) The system updates the setting “GoogleDriveStatus” in the database from “active” to “nonActive”
Exceptions	[dbfailure] Can’t update Google Integration status, nothing changes
Result	The Google integration is deactivated.

3.5.Result Google Drive integration deactivation

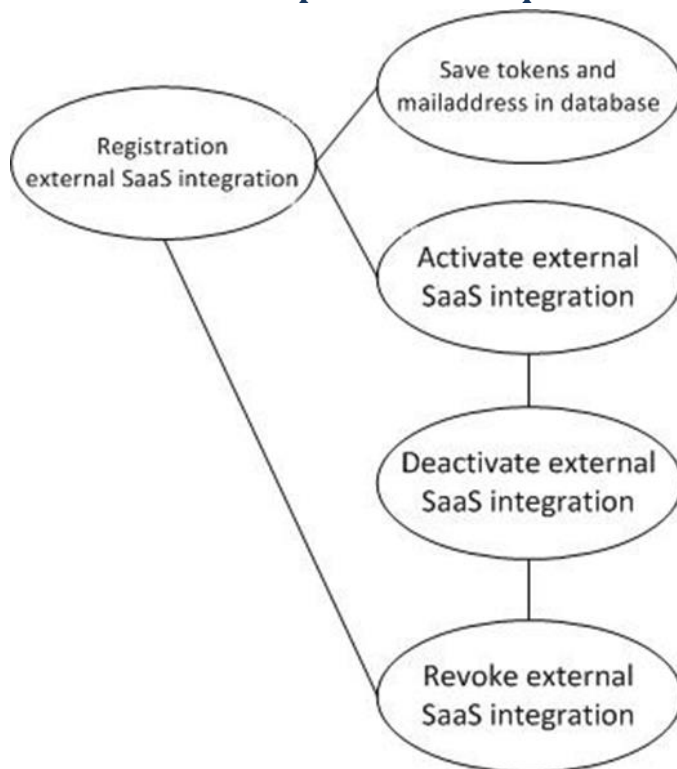
External SaaS Integration			
Google Integration status	nonActive	Activate GoogleDrive	Revoke Access to Google mailaddress@gmail.com
Dropbox Integration status	not yet available		

4. Activate integration

4.1.General integration activation

Name	Activate external SaaS integration
Summary	The user activates the external SaaS integration
Actor	User
Assumptions	The user is at the page General→User settings The ExternalSaaSStatus for this user is nonActive
Description	(1) The user clicks on the button “Activate {SaaSname}” (2) The system updates the setting “{SaaSname}Status” in the database from “nonActive” to “active”
Exceptions	[dbfailure] Can’t update SaaS integration status, nothing changes
Result	The external SaaS integration is activated.

4.2.Relationships to other requirements



4.3. Google Drive integration activation

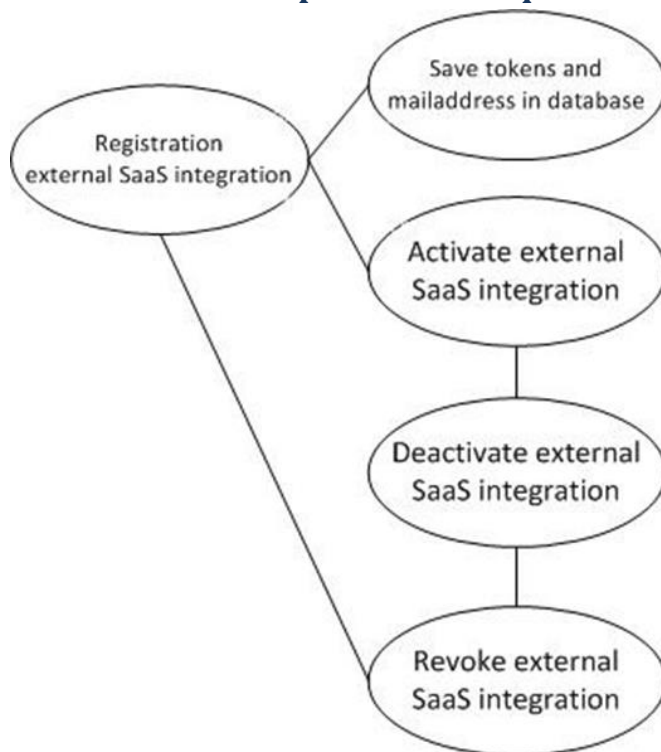
Name	Activate Google Drive integration
Summary	The user activates the Google Integration
Actor	User
Assumptions	The user is at the page General → User settings The GoogleDriveStatus for this user is nonActive
Description	(1) The user clicks on the button “Activate Google Drive” (2) The system updates the setting “GoogleDriveStatus” in the database from “nonActive” to “active”
Exceptions	[dbfailure] Can’t update Google Integration status, nothing changes
Result	The Google integration is activated.

5. Revoke registration

5.1.General integration revoke

Name	Revoke external SaaS integration
Summary	The user revokes the external SaaS integration
Actor	User
Assumptions	The user is at the page General→User settings The ExternalSaaSStatus for this user is nonActive
Description	(1) The user clicks on the button “Revoke Access to {SaaSName}” (2) The system updates the setting “{SaaSName}Status” in the database from “nonActive” to “nonRegistered” (3) The system updates the settings “{SaaSName}UserId”, “{SaaSName}RefreshToken” and “{SaaSName}AccessToken” in the database to “” (empty)
Exceptions	[dbfailure] Can’t update the SaaS integration database fields, nothing changes
Result	The external SaaS integration is revoked.

5.2.Relationships to other requirements



5.3.Google Drive integration revoke

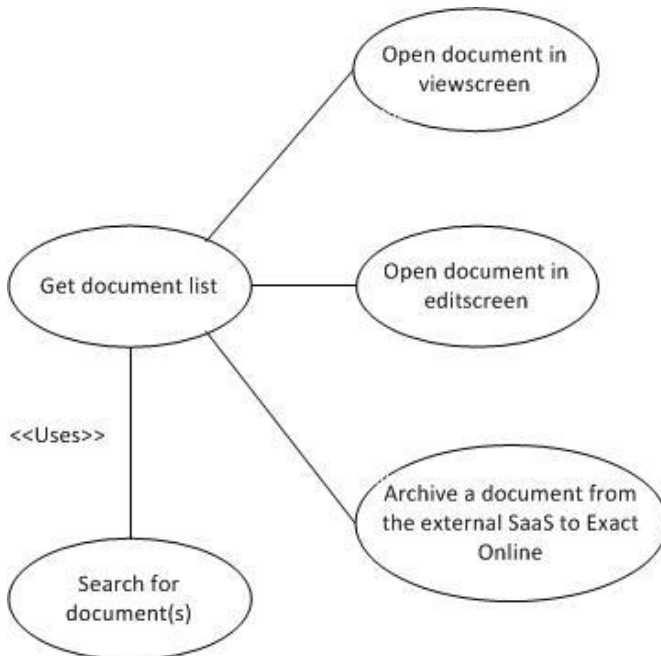
Name	Revoke Google Drive integration
Summary	The user revokes the external SaaS integration
Actor	User
Assumptions	The user is at the page General→User settings The ExternalSaaSStatus for this user is nonActive
Description	(4) The user clicks on the button “Revoke Access to {SaaSname}” (5) The system updates the setting “{SaaSname}Status” in the database from “nonActive” to “nonRegistered” (6) The system updates the settings “{SaaSname}UserId”, “{SaaSname}RefreshToken” and “{SaaSname}AccessToken” in the database to “” (empty)
Exceptions	[dbfailure] Can’t update the SaaS integration database fields, nothing changes
Result	The external SaaS integration is revoked.

6. Document list

6.1.General integration document list

Name	Document list concerning a particular topic(project, account)
Summary	The system shows a list of documents concerning a particular topic
Actor	User
Assumptions	The user is at the page of the particular topic
Description	(1) The system loads the first part of the page (2) The system checks if the user uses the {externalSaaSname} integration (3) The system makes an asynchronous task for requesting the document list a. The system connects with the external SaaS b. The system receives data about the documents concerning the particular topic (4) The system meanwhile continues with the second part of the page (5) The system waits for the asynchronous task to finish (6) The system retrieves the results of the asynchronous task (7) The system shows the document list in the {externalSaaSname} integration Viewsection (8) The system loads the rest of the page
Exceptions	[noESI] occurs when the user has no registration for the {externalSaaSname} integration, the section of the page and tasks concerning the integration will be skipped [noConnection]There isn't a connection with the external SaaS, the viewsection will be shown with an error message, the rest of the page will be shown normally, the error will be registered in the error log
Result	The page will be shown with the document list concerning the particular topic of that page

6.2.Relationships to other requirements



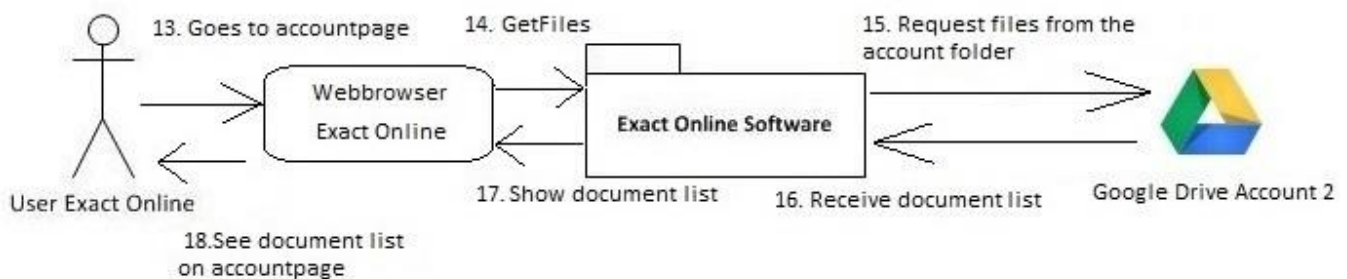
- E.M.Pekel

6.3. Google integration document list

The asynchronous task isn't implemented yet. On this moment the request for the list of documents gets done synchronal. The desired situation is shown below.

Name	Document list concerning a particular topic(project, account)
Summary	The system shows a list of documents concerning a particular topic
Actor	User
Assumptions	The user is at the page of the particular topic
Description	(1) The system loads the first part of the page (2) The system checks if the user uses Google Integration (3) The system makes an asynchronous task for requesting the document list - The system connects with Google Drive - The system receives data about the documents concerning the particular topic (4) The system meanwhile continues with the second part of the page (5) The system waits for the asynchronous task to finish (6) The system retrieves the results of the asynchronous task (7) The system shows the document list in the Google Integration Viewsection (8) The system loads the rest of the page
Exceptions	[noGI] occurs when the user has no registration for Google Integration, the section of the page and tasks concerning the integration will be skipped [noConnection] There isn't a connection with Google, the viewsection will be shown with an error message, the rest of the page will be shown normally, the error will be registered in the error log
Result	The page will be shown with the document list concerning the particular topic of that page

6.4. Technical side document list



6-1 Continues on figure 2-1, technical side of the document list functionality

How the document list functionality of the Google Drive implementation works is shown in figure 6-1. Each step will be explained further below.

Step 13: A user (with an active Google Drive implementation) goes to an account page.

Step 14: Next to gathering the standard account information, the Exact Online Software sends a request to the Document Management Integration module, in this case to the Google Drive implementation. The

request is for getting data about the documents in Google Drive that correspond to the account of the page. This request includes a “list” of the properties we want to have from a document.

Step 15: The Google Drive implementation puts together a HTTPsRequest and sends the request to the Google API.

Step 16: The Google API/Drive account response with an object list in the format JSON. This list gets deserialized by the Google Drive library to a more workable list.

Step 17: The Exact Online Software then makes the list of document data presentable to the user. And puts the account page together.

Step 18: The user now sees the account page include the Google Drive documents view section with the list of document data, links to the document, search field, AddNew-button and ShowAll-button.

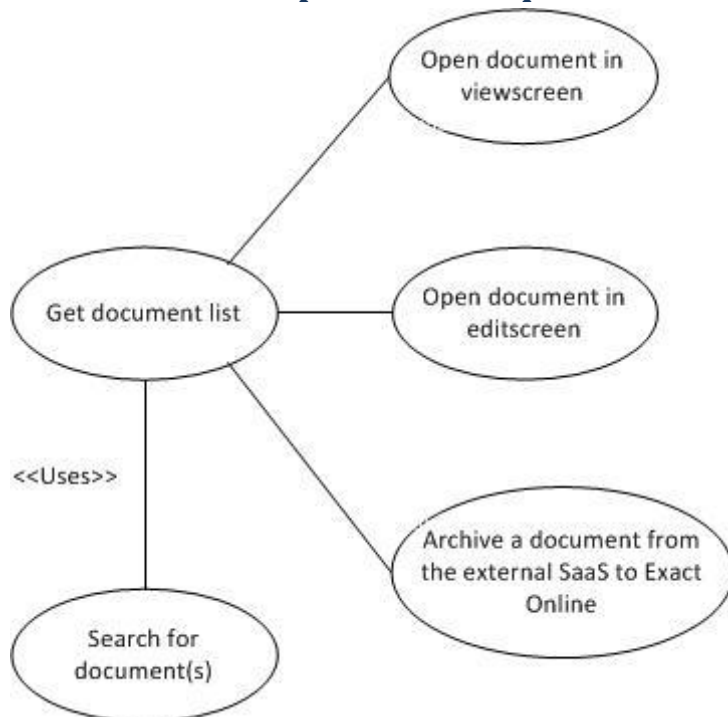
7. Search for document(s)

For the search for documents the use of the asynchronous pattern implementation is possible. The asynchronous task isn't implemented yet. On this moment the request for the list of documents gets done synchronal. The current situation is shown below.

7.1.General integration search documents

Name	Search for documents
Summary	A user searches for documents corresponding to the account or project of the page
Actor	User
Assumptions	The user is at an account or project page The user has an active registration for the external SaaS integration
Description	(1) The user types the word(s) to search for into search field and clicks on the button search. (2) The system connects with the external SaaS (3) The system receives data about documents that correspond to the search query. (4) The system shows the document list in the external SaaS Viewsection
Exceptions	[connectionError] It isn't possible to make connection to the external SaaS. An error message will appear and the error gets registered in the error log.
Result	The user sees a list of documents corresponding to his/hers search query.

7.2.Relationships to other requirements



7.3.Google integration search documents

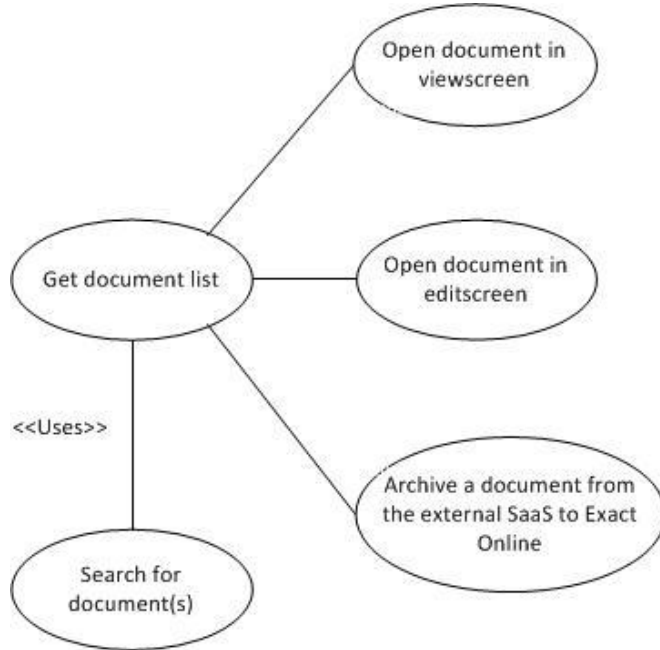
Name	Search for documents
Summary	A user searches for documents corresponding to the account or project of the page
Actor	User
Assumptions	The user is at an account or project page The user has an active registration for the Google Drive integration
Description	(1) The user types the word(s) to search for into search field and clicks on the button search. (2) The system connects with the Google Drive (3) The system receives data about documents that correspond to the search query. (4) The system shows the document list in the Google Integration Viewsection
Exceptions	[connectionError] It isn't possible to make connection to Google Drive. An error message will appear and the error gets registered in the error log.
Result	The user sees a list of documents corresponding to his/hers search query.

8. View document content

8.1.General integration view document

Name	View document content
Summary	A user views the content of a document
Actor	User
Assumptions	The user is at an account or project page The user has an active registration for the external SaaS integration
Description	(1) The user clicks on the name of the document he/she wants to view. (2) The system opens the document in a new window by the external SaaS.
Exceptions	[connectionError] It isn't possible to make connection to the external SaaS. An error message will appear and the error gets registered in the error log.
Result	The user views the content of a document.

8.2.Relationships to other requirements



8.3.Google integration view document

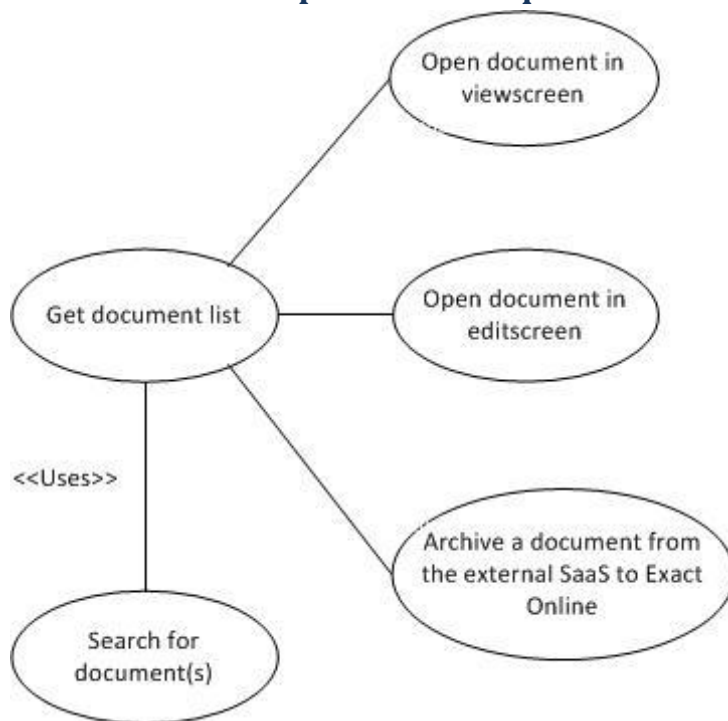
Name	View document content
Summary	A user views the content of a document
Actor	User
Assumptions	The user is at an account or project page The user has an active registration for the Google Drive integration
Description	(1) The user clicks on the name of the document he/she wants to view. (2) The system opens the document in a new window within the Google Drive environment.
Exceptions	[connectionError] It isn't possible to make connection to the Google Drive. An error message will appear and the error gets registered in the error log.
Result	The user views the content of a document.

9. Edit document content

9.1.General integration edit document content

Name	Edit document content
Summary	A user edits the content of a document
Actor	User
Assumptions	The user is at an account or project page. The user has an active registration for the external SaaS integration. The user has rights to edit the document it will click on.
Description	(1) The user clicks on the name of the document he/she wants to edit. (2) The system opens the document in a new window by the external SaaS. (3) The user edits the document in the external SaaS environment.
Exceptions	[connectionError] It isn't possible to make connection to the external SaaS. An error message will appear and the error gets registered in the error log. [noEditRights] The system opens the document in a new window by the external SaaS but the user can only view the content of the document.
Result	The user edits the content of a document.

9.2.Relationships to other requirements



9.3. Google integration edit document content

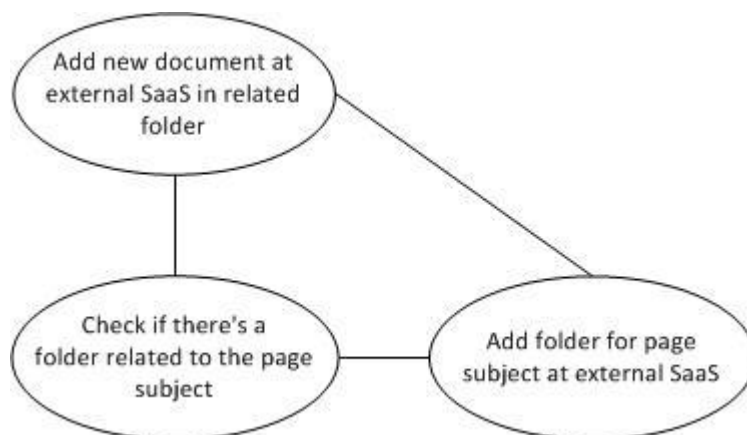
Name	Edit document content
Summary	A user edits the content of a document
Actor	User
Assumptions	The user is at an account or project page. The user has an active registration for the Google Drive integration. The user has rights to edit the document it will click on.
Description	(1) The user clicks on the name of the document he/she wants to edit. (2) The system opens the document in a new window by Google Drive. (3) The user edits the document in the Google Drive environment.
Exceptions	[connectionError] It isn't possible to make connection to Google Drive. An error message will appear and the error gets registered in the error log. [noEditRights] The system opens the document in a new window by Google Drive but the user can only view the content of the document.
Result	The user edits the content of a document.

10. Add new document

10.1. General integration add new document

Name	Add new document
Summary	The user adds a new document to the related external SaaS account/project folder from within Exact Online
Actor	User
Assumptions	The user is on the page of an account or project where the user wants to add a new document to.
Description	(1) The user clicks on the “Add new”-link in the right corner of the viewsection (2) The system checks if there’s a folder for the account/project of the page in the external SaaS (2a) No folder: The system creates a folder for the account/project in the external SaaS (2b) Folder exists: The system doesn’t have to create a folder (3) The system asks for the document’s title and filename (4) The user insert the document title and filename, and clicks “Create” (5) The system adds the new document in the related folder in the external SaaS (6) The system opens the document in a new window (7) The user adds content to the document and saves the document (8) The User refreshes the page account/project page in Exact Online (9) The system retrieves and shows the documentlist
Exceptions	[noConnection]There isn’t a connection with the external SaaS, the viewsection will be shown with an error message, the rest of the page will be shown normally, the error will be registered in the errorlog
Result	A new document is added to the corresponding folder in the external SaaS

10.2. Relationships to other requirements



10.3. Google integration add new document

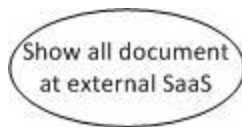
Name	Add new document
Summary	The user adds a new document to the related Google Drive account/project folder from within Exact Online
Actor	User
Assumptions	The user is on the page of an account or project where the user wants to add a new document to.
Description	(1) The user clicks on the “Add new”-link in the right corner of the viewsection (2) The system checks if there’s a folder for the account/project of the page in Google Drive (2a) No folder: The system creates a folder for the account/project in Google Drive (2b) Folder exists: The system doesn’t have to create a folder (3) The system asks for the document’s title and filename (4) The user insert the document title and filename, and clicks “Create” (5) The system adds the new document in the related folder in Google Drive (6) The system opens the document in a new window (7) The user adds content to the document and saves the document (8) The User refreshes the page account/project page in Exact Online (9) The system retrieves and shows the documentlist
Exceptions	[noConnection]There isn’t a connection with Google, the viewsection will be shown with an error message, the rest of the page will be shown normally, the error will be registered in the errorlog
Result	A new document is added to the corresponding folder in Google Drive

11. Show all documents

11.1. General integration show all documents

Name	Show all
Summary	Shows all documents in the external SaaS
Actor	User
Assumptions	The user is on the page of an account or project for which it wants to show the document list in the external SaaS
Description	(1) The user clicks on the “Show all”-link in the right corner of the viewsection (2) The system opens a new window with a redirect to the corresponding account/project folder in the external SaaS
Exceptions	[noConnection]There isn’t a connection with the external SaaS, the error will be registered in the errorlog [folderNotExists] If there is no corresponding folder for the account or project, it will send the user to the basefolder “ExactOnlineApp” in the external SaaS
Result	The user sees all the documents responding to the account/project in the external SaaS

11.2. Relationships to other requirements



11.3. Google integration show all documents

Name	Show all
Summary	Shows all documents in Google Drive
Actor	User
Assumptions	The user is on the page of an account or project for which I wants to show the document list in Google Drive
Description	(3) The user clicks on the “Show all”-link in the right corner of the viewsection (4) The system opens a new window with a redirect to the corresponding account/project folder in Google Drive
Exceptions	[noConnection]There isn’t a connection with Google, the error will be registered in the errorlog [folderNotExists] If there is no corresponding folder for the account or project, It will send the user to the basefolder “ExactOnlineApp” in Google Drive
Result	The user sees all the documents responding to the account/project in Google Drive.

Literature

Göritzer, R. (2011). *Using software change impact analysis*. Klagenfurt, Austria(Oostenrijk).

Shalloway, A., & Trott, J. R. (2010). *Design Patterns Explained*. Boston: Pearson Education, Inc.

Bijlage I: Test Documentation

Exact Cloud Solutions

Test plan and test scripts for Integration Tests

Project External Documentmanagement SaaS' Integration

Opdrachtgever: Ronald Dähne – Director Technology Cloud Solutions
Begeleider: Ad van der Hoeven – Senior Research Engineer
Datum: 25 January 2013
Versie: 1.0

= exact

And it all comes together.

Google

Microsoft®
V.NET

Auteur

E.M. Pekel
Student – 20064646
Informatica
Haagse Hogeschool – Den Haag

Begeleidend examiner: G.A. Mijharends
Tweede examiner: Dhr. B. Kuiper

Preface

In this document the design and test report will be presented. Because there's only one developer (the intern) that will set up and execute the tests, the subjects like management and test organization are left out.

1. Introduction

1.1 Objective of the development test plan

The objective of this Test Plan (TP) for the integration test is to inform all who are involved in the test process about the approach, the activities and the deliverables concerning the integration test for the Integration of Document Management SaaS' (IDMS).

1.2 Assignment

1.2.1 Client

The clients for the integration test are:

- Ronald Dähne: Product Development Director
- Ad van der Hoeven: Senior Research Engineer
- Robert Huberts: Senior Functional Designer

Every week of the testing process they will be briefed about the progression of the test process.

1.2.2 Supplier

The intern (Marlies Pikel) will set up, execute and document the whole integration testing process.

1.2.3 Assignment

The goal of the integration test on the IDMS module is:

- Short term:

Determine how the module meets the functional requirements, which have been agreed on with the stakeholders. After the testing process the functionalities which don't work as expected can be recoded.

Based on the test results, advice can be given about whether or not the module is ready to be merged to the development branch, where all employees of Cloud Solution can reach it and work on it.
- Long-term:

The integration test will be made suitable for use during and after maintenance on the IDSM module and the parts of Exact Online who make use of the IDMS module.

1.2.4 Scope

Within scope:

- The IDMS module itself:
 - Exact.ExternalSaaSdocFactory
 - Exact.ExternalSaaSdocument
 - Exact.GoogleModels
- The modules of Exact Online how directly make use of the IDMS module

Out of scope:

- The modules of Exact Online how not directly make use of the IDMS module.

1.2.5 Preconditions and assumptions

- The intern has access to the development environment on which the integration test can be run successfully;
- The technical documentation is finished;
- The IDMS module is “finished” (ready for the integration test) and code reviewed.

2. Test Basis

The test basis is formed by:

Product design:

Document Name	Version	Date	Author
Requirements opstellen			E.M. Pekel
Technical Documentation IDMS			E.M. Pekel

Norms and standards:

Source Name	Version	Date	Author
Testen volgens TMAP	Second print	2000	Martin Pol, Ruud Teunissen and Erik van Veenendaal

Manual:

Manual Name	Author
Manual for MS Visual Studio's standard test tools on the Microsoft website	others

Architecture:

Document Name	Version	Date	Author
Architectuur beschrijving			E.M. Pekel

Code:

The code of the IDMS module is a source of information itself.

3. Test Strategy Development Tests

The strategy of set up and execution of the integration test concerning the IDMS module will be to test all functionalities of the module. Testing the quality attributes requirements will not be included at this time, because the functionalities which have the most positive impact on meeting the quality attributes will be programmed by another development team on a later notice.

The functionalities will be tested from the implementation viewpoint. This means that the public functions will be called and the private functions will automatically tested as well. The public functions are:

- SetupService
- CheckRefreshAccessToken
- GetFiles
- SearchFiles
- ShowAll
- CreateFile
- Revoke

The functions as shown above will be tested on functionality and therefor (in the most cases) the two most executed paths will be tested.

The test technique that will be used for the integration test is algorithm testing. The purpose of integration testing is to test different scenarios that run through one or more systems to check that the different functions properly aligned. The algorithm tests can be performed as chain test to test the integration between Exact Online and Google Drive test.

Based on the test result a statement can be made about the extent to which the module meets the functional requirements, which have been agreed on with the stakeholders.

4. Approach

In this chapter the translation is being made from the 'what are we going to test' to the 'how are we going to test it'.

4.1 Execution Test Strategy

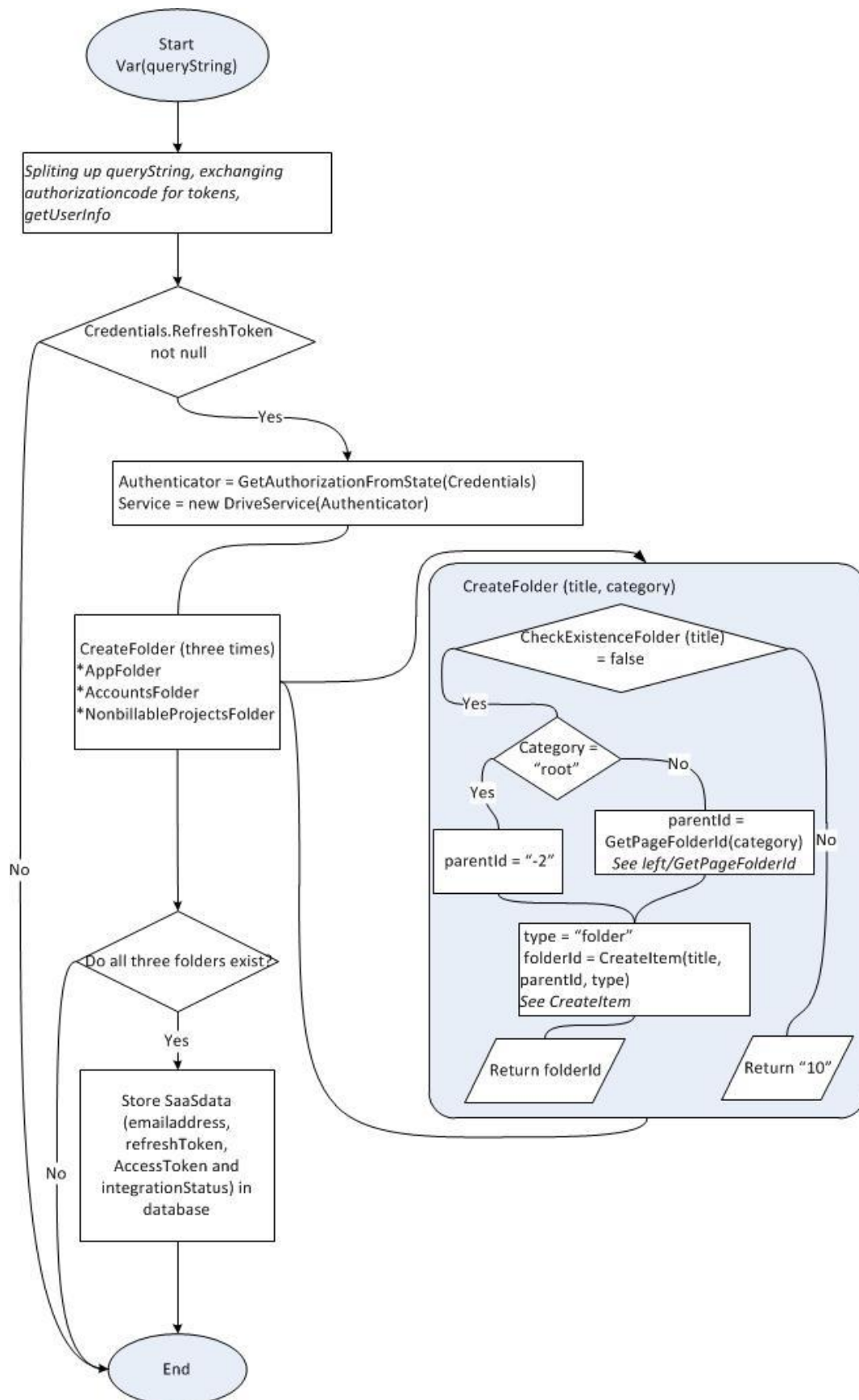
The intern will execute the Integration Test in the development environment. With the outcome of the integration test it is determined if the module is working correctly and according to the requirement. The focus is on the function of the processes that can be executed by the module when being used in Exact Online. The intern will report the outcomes of the integration test to the clients.

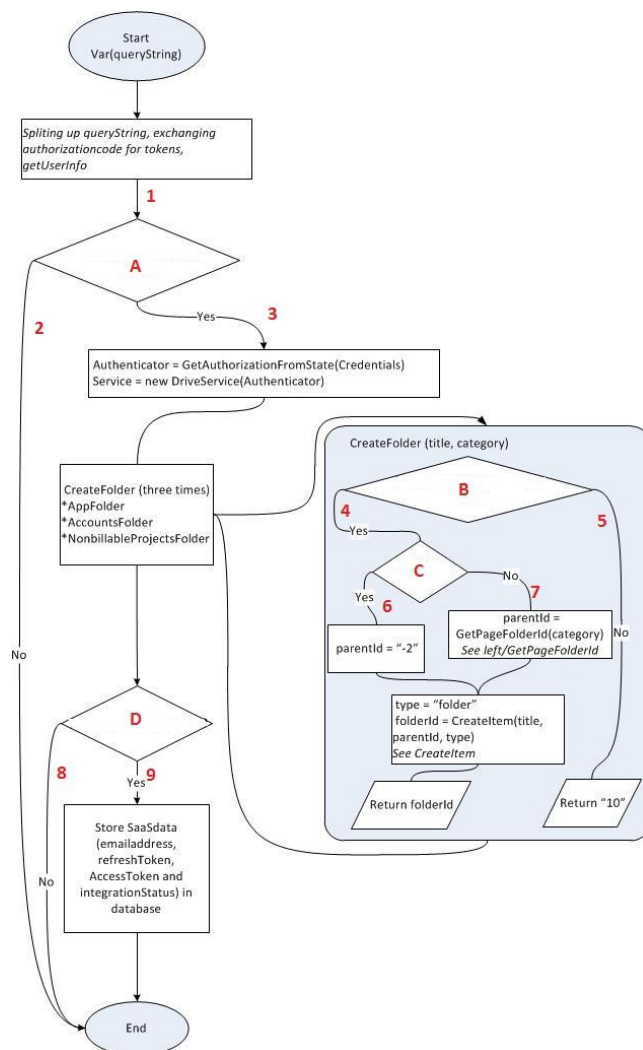
The process that will be followed for the algorithm test is:

1. Identifying the decision points (within the functionalities of the module)
2. Determining the test paths
3. Specifying the test cases
4. Determining the initial data
5. Compose the test scripts

5. Test SetupService

5.1 Identifying the decision points





5.2 Determining the test paths

Decision point	Path
Start	1
A	2(end), 3
B	4, 5
C	6, 7
D	8(end), 9(end)

SetupService BaseFolders don't exist: 1, 3, 4, 6, 4, 7, 4, 7, 9

RefreshToken null path: 1, 2

SetupService BaseFolders still exist path: 1, 3, 5, 5, 5, 9

Only the SetupService BaseFolders don't exist path and folders still exist path and SetupService BaseFolders still exist path are testable without code manipulating and these are the two paths that will occur the most. These will be the two which will be tested.

5.3 Specifying the test cases

Test case SetupService-BaseFolders-don't-exist path:

For the SetupService-BaseFolders-don't-exist path the Google Drive account must be checked. The base folders ("ExactOnlineApp", "Exact-Accounts", "Exact-NonBillableProjects") don't exist.

Test case SetupService-BaseFolders-still-exist path:

This path can best be tested directly after the SetupService-BaseFolders-don't-exist path (see above) is tested, because the base folders ("ExactOnlineApp", "Exact-Accounts", "Exact-NonBillableProjects") will exist when the SetupService-BaseFolders-don't-exist path test succeeded. Otherwise create the base folders yourself (use the exact names like above) in the root folder of the Google Drive account you would like to use during testing.

5.4 Determining the initial data

For the user "CustomerAdvanced" the following fields need to be empty (in the database):

- GoogleDriveStatus
- GoogleUserId
- GoogleRefreshToken
- GoogleAccessToken

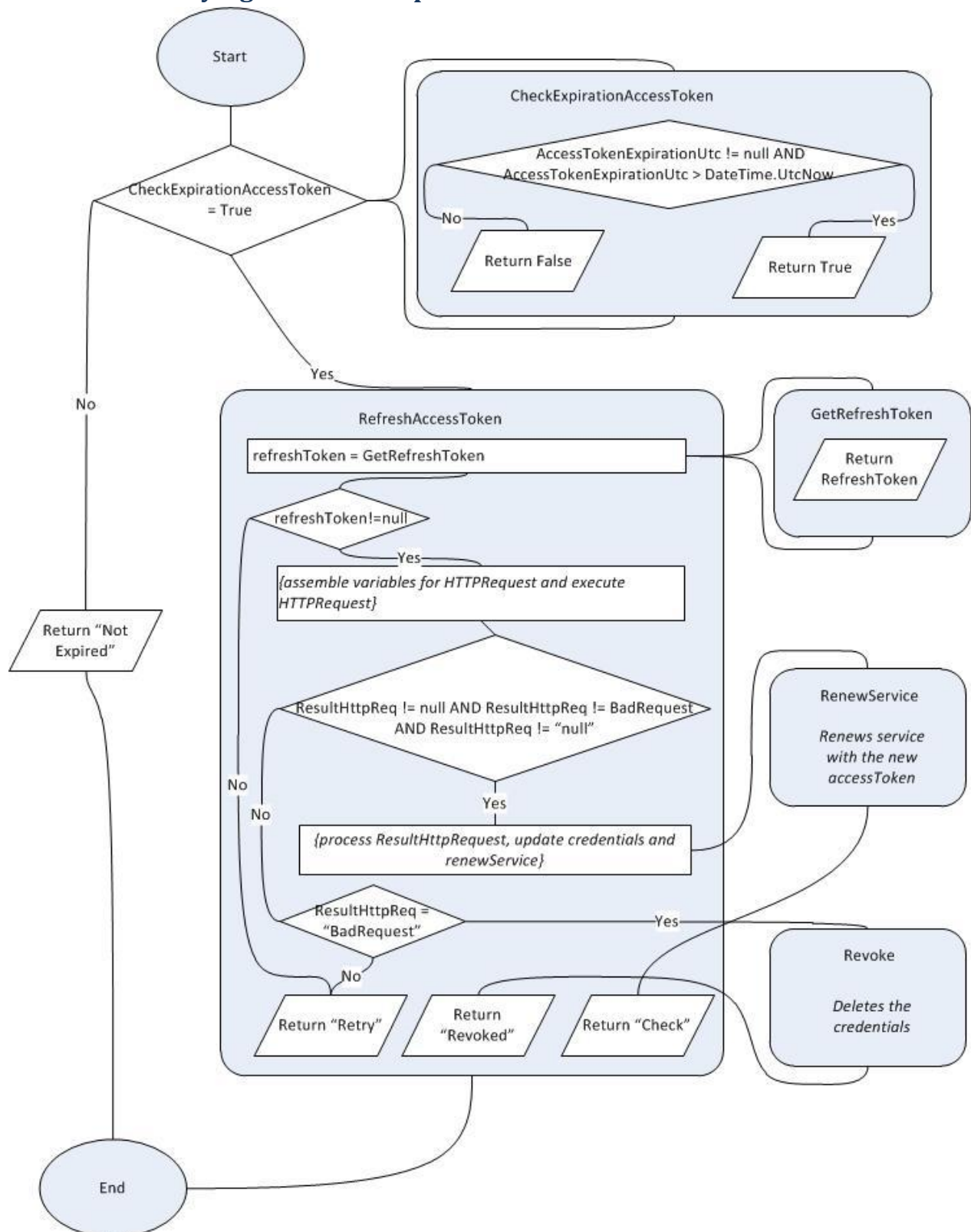
5.5 Compose the test scripts

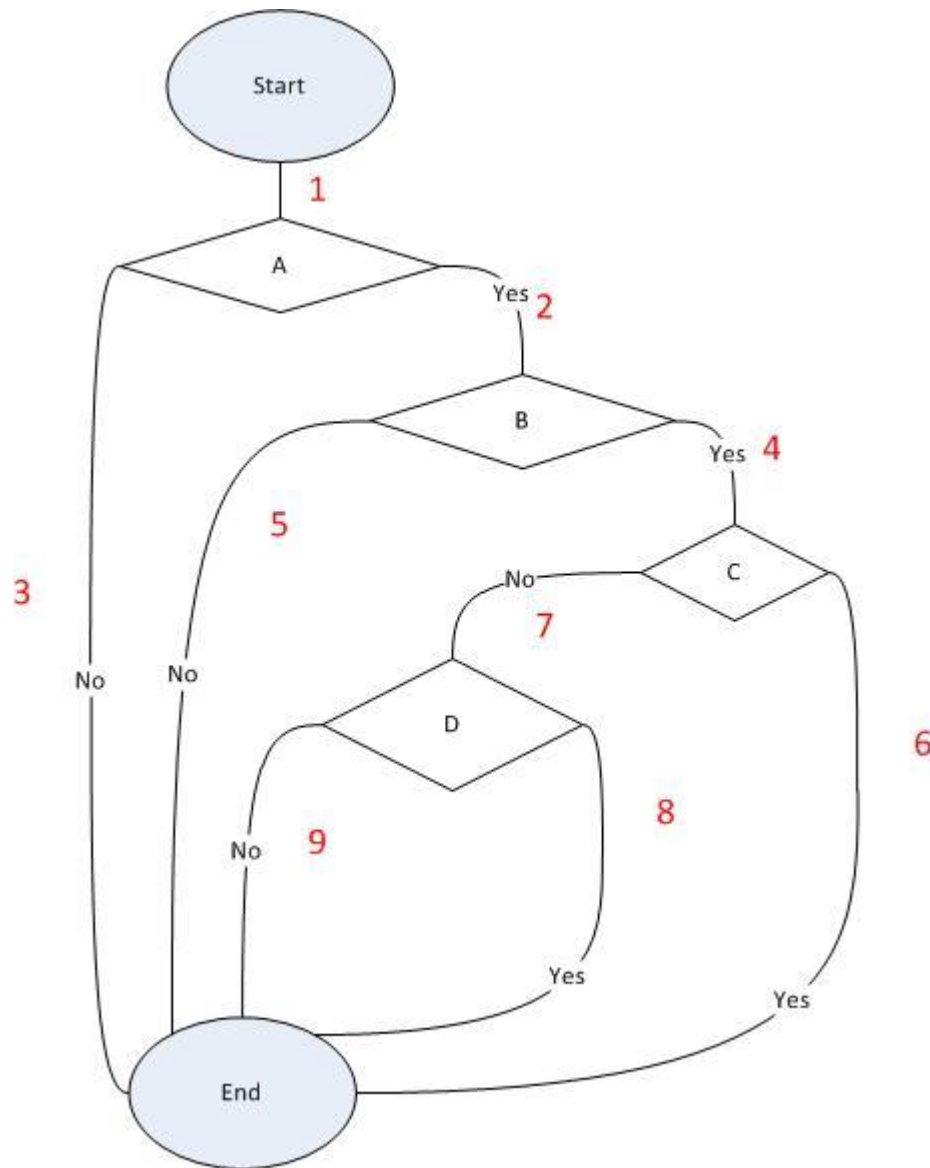
The result of the above steps can be found in the appendices, under the names:

- SetupService_NoBaseFolders_ServicelsSetUpAndFoldersAreCreated;
- SetupService_BaseFoldersAlreadyExist_ServicelsSetUpAndFoldersAreNotCreated.

6. Test CheckRefreshAccessToken

6.1 Identifying the decision points





6.2 Determining the test paths

Decision point	Path
Start	1
A	2, 3(end)
B	4, 5(end)
C	6(end), 7
D	8(end), 9(end)

Expired path: 1, 2, 4, 6

Not Expired path: 1, 3

Can't get refreshToken path: 1, 2, 5

Badrequest path: 1, 2, 4, 7, 8

NoResultHttpRequest path: 1, 2, 4, 7, 9

The Expired and Not-Expired paths are the paths mostly used, so these two will be tested.

6.3 Specifying the test cases

For both the Expired and the not-expired path the user ("CustomerAdvanced") needs to have an active GoogleIntegration.

6.4 Determining the initial data

For the user "CustomerAdvanced" the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (active)
- GoogleUserId → {emailaddress of google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

When these fields aren't filled, you have to register either through the webbrowser (like a real user would do) or by successfully perform one of the following tests:

- SetupService_NoBaseFolders_ServicelsSetUpAndFoldersAreCreated;
- SetupService_BaseFoldersAlreadyExist_ServicelsSetUpAndFoldersAreNotCreated.

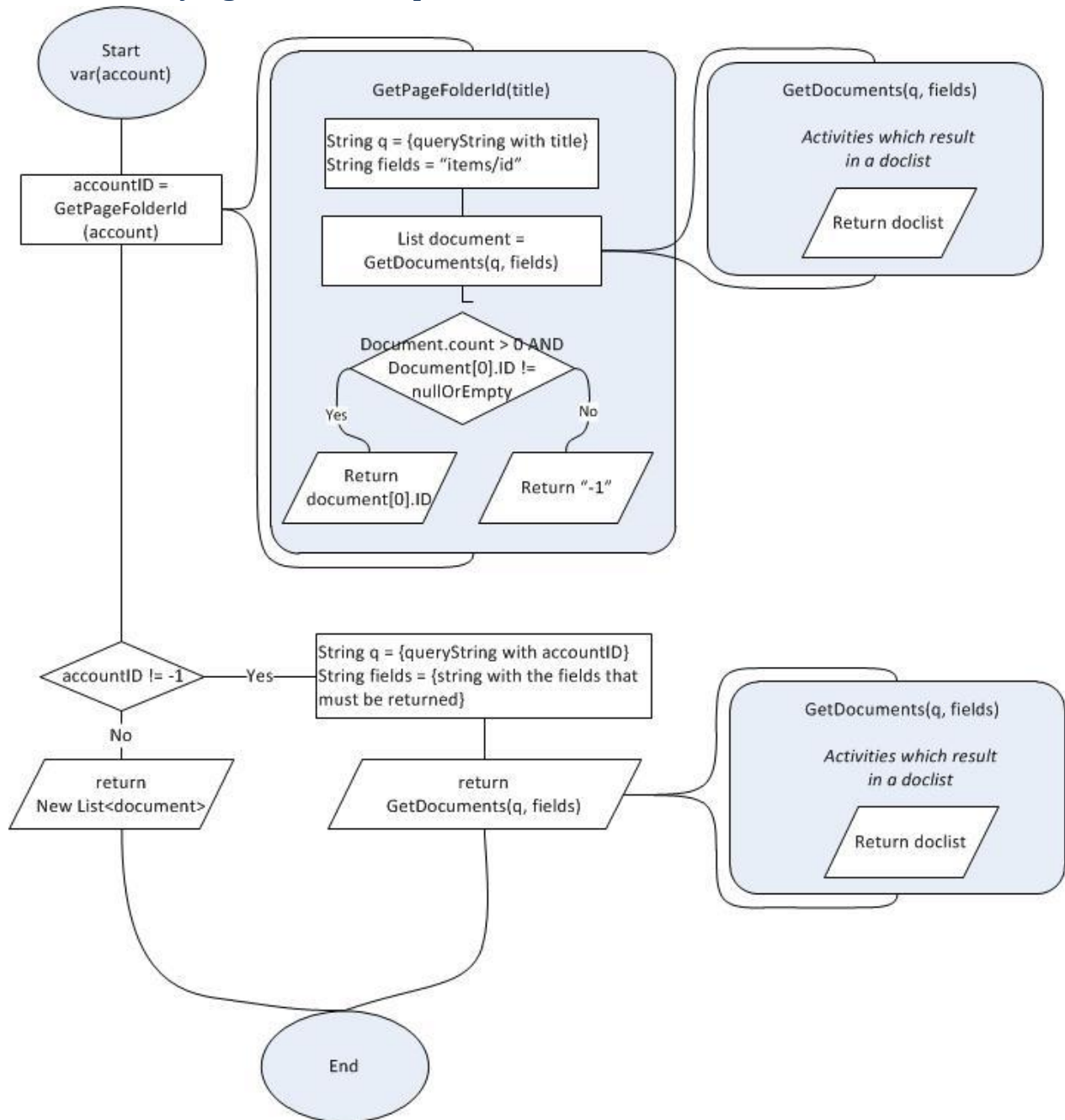
6.5 Compose the test scripts

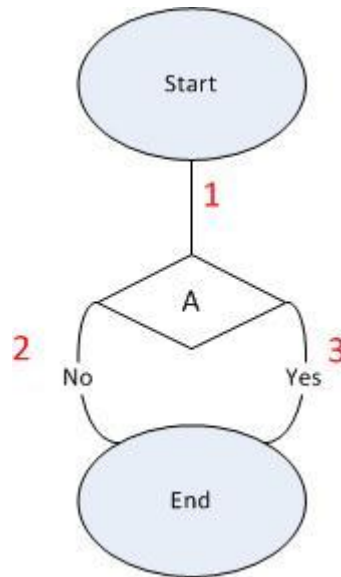
The result of the above steps can be found in the appendices, under the names:

- RefreshAccessToken_WithRefreshTokenFromDBAccessTokenExpired_AccessTokenReceivedCorrectly;
- RefreshAccessToken_WithRefreshTokenFromDBAccessTokenNotExpired_AccessTokenNotRefreshed.

7 Test GetFiles

7.1 Identifying the decision points





7.2 Determining the test paths

Decision point	Path
Start	1
A	2(end), 3(end)

Account-Folder-Exists path: 1, 2

Account-Folder-Doesn't-Exist path: 1,3

Both paths are used a lot, so both paths get tested.

7.3 Specifying the test cases

An active GoogleIntegration needs to be there for both paths.

Account-Folder-Exists path:

The folder "Growin Fietsen" must be in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts > Growin Fietsen) and within this folder there's at least one file.

Account-Folder-Doesn't-Exists path:

The folder "Not Existing AccountFolder" must not exist in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts).

7.4 Determining the initial data

For the user "CustomerAdvanced" the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (active)
- GoogleUserId → {email address of Google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

When these fields aren't filled, you have to register either through the webbrowser (like a real user would do) or by successfully perform one of the following tests:

- SetupService_NoBaseFolders_ServicesSetUpAndFoldersAreCreated;
- SetupService_BaseFoldersAlreadyExist_ServicesSetUpAndFoldersAreNotCreated.

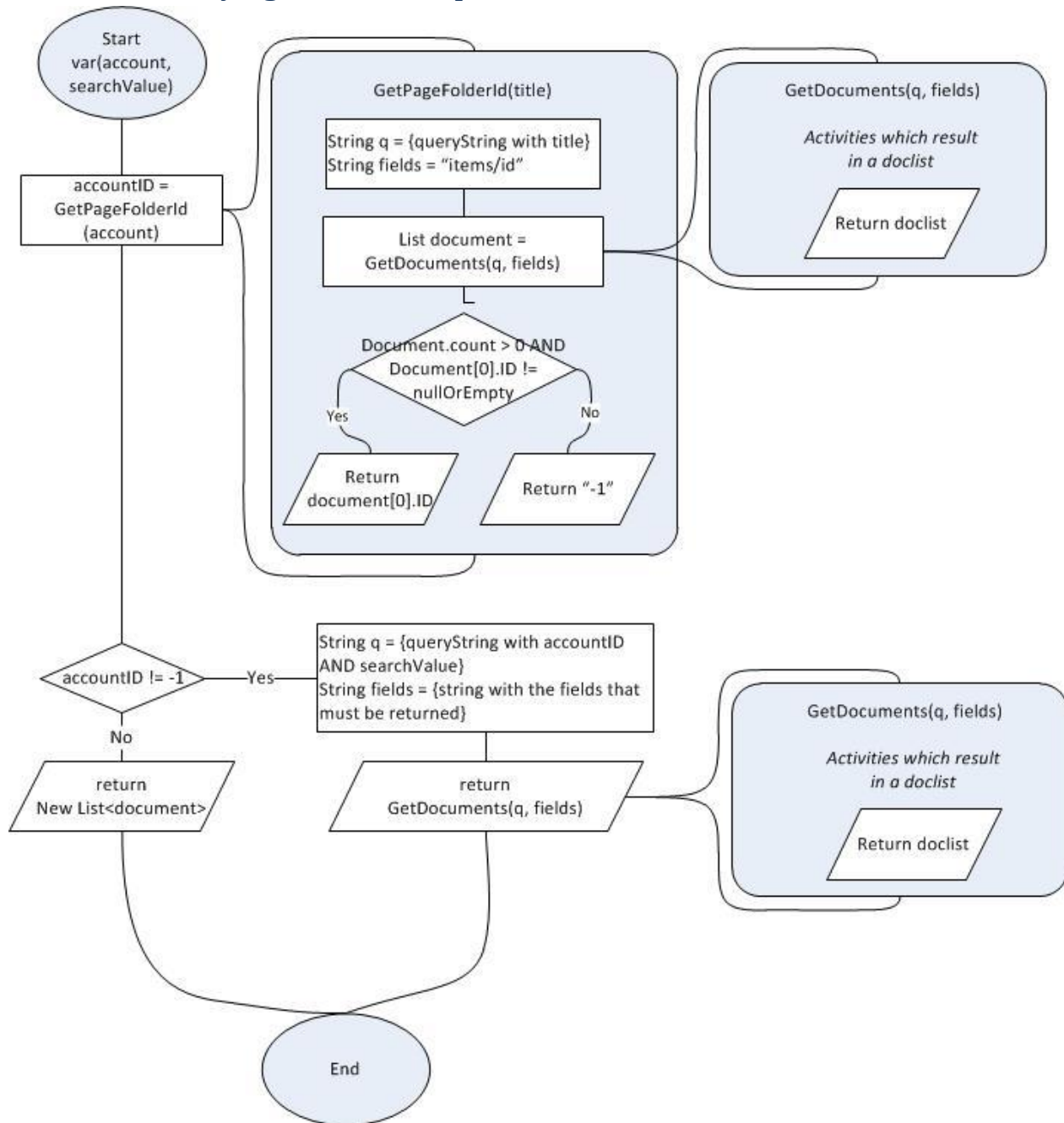
6.3 Compose the test scripts

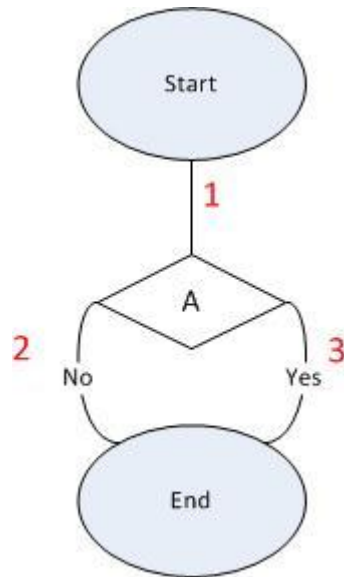
The result of the above steps can be found in the appendices, under the names:

- GetFiles_AccountFolderExists_FilledDocList;
- GetFiles_AccountFolderNotExist_EmptyDocList.

7. Test SearchFiles

7.1 Identifying the decision points





7.2 Determining the test paths

Decision point	Path
Start	1
A	2(end), 3(end)

Account-Folder-Exists path: 1, 2

Account-Folder-Doesn't-Exist path: 1, 3

The Account-Folder-Exists path will be used the most, but in this case the account-folder-doesn't-exist will be tested also.

7.3 Specifying the test cases

An active GoogleIntegration needs to be there for both paths.

Account-Folder-Exists path:

The folder "Growin Fietsen" must be in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts > Growin Fietsen) and within this folder there's at least one file with has in the title or content the word "Lorem". (It sometimes takes a few minutes for Google to update the document information)

Account-Folder-Doesn't-Exists path:

The folder "Not Existing AccountFolder" must not exist in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts).

7.4 Determining the initial data

For the user "CustomerAdvanced" the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (active)
- GoogleUserId → {email address of Google account}

- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

When these fields aren't filled, you have to register either through the webbrowser (like a real user would do) or by successfully perform one of the following tests:

- SetupService_NoBaseFolders_ServicesSetUpAndFoldersAreCreated;
- SetupService_BaseFoldersAlreadyExist_ServicesSetUpAndFoldersAreNotCreated.

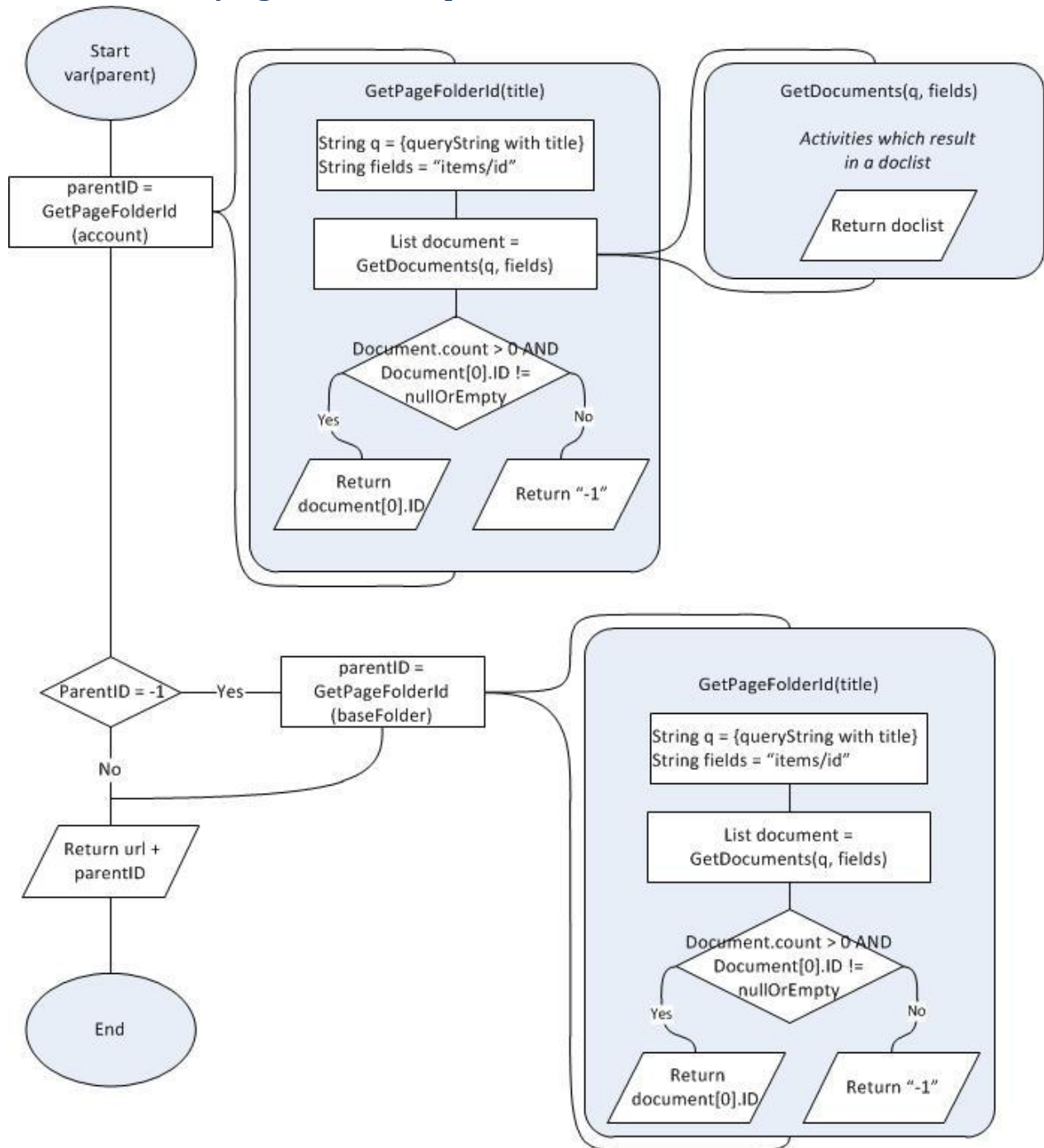
7.5 Compose the test scripts

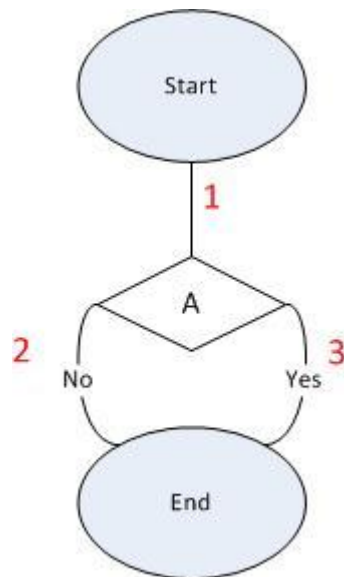
The result of the above steps can be found in the appendices, under the names:

- SearchFiles_AccountFolderExistsSearchValuePresent_FilledDocList;
- SearchFiles_AccountFolderNotExistsSearchValueAnything_EmptyDocList.

8. Test ShowAll

8.1 Identifying the decision points





8.2 Determining the test paths

Decision point	Path
Start	1
A	2(end), 3(end)

Account-Folder-Exists path: 1, 2

Account-Folder-Doesn't-Exist path: 1, 3

The Account-Folder-Exists path will be used the most, but in this case the account-folder-doesn't-exist will be tested also.

8.3 Specifying the test cases

An active GoogleIntegration needs to be there for both paths.

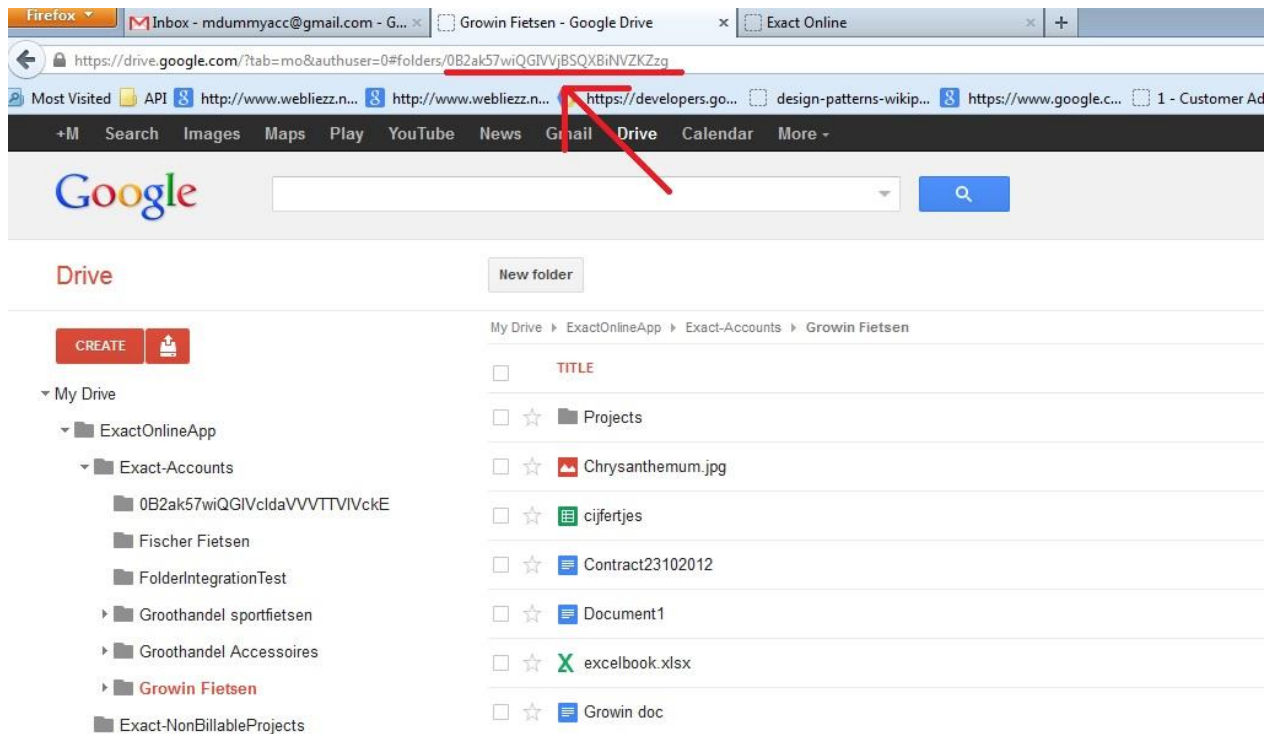
Account-Folder-Exists path:

The folder "Growin Fietsen" must be in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts > Growin Fietsen).

In de top of the test code file the following line must me edited:

`Private Const FolderIdExistingAccountFolder As String = ""`

To find the FolderId van the folder "Growin Fietsen" you have to go to your Google Drive Account and open the folder "Growin Fietsen". The folderID can now be found in the navigation bar.



Account-Folder-Doesn't-Exists path:

The folder "Not Existing AccountFolder" must not exist in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts).

In de top of the test code file the following line must me edited:

`Private Const FolderIdExactOnlineApp As String = ""`

To find the FolderId van the folder "ExactOnlineApp" you have to go to your Google Drive Account and open the folder "ExactOnlineApp". The folderID can now be found in the navigation bar.

8.4 Determining the initial data

For the user "CustomerAdvanced" the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (active)
- GoogleUserId → {email address of Google account}
- GoogleRefreshToken → {refreshToken}

- GoogleAccessToken → {accessToken}

When these fields aren't filled, you have to register either through the webbrowser (like a real user would do) or by successfully perform one of the following tests:

- SetupService_NoBaseFolders_ServicesSetUpAndFoldersAreCreated;
- SetupService_BaseFoldersAlreadyExist_ServicesSetUpAndFoldersAreNotCreated.

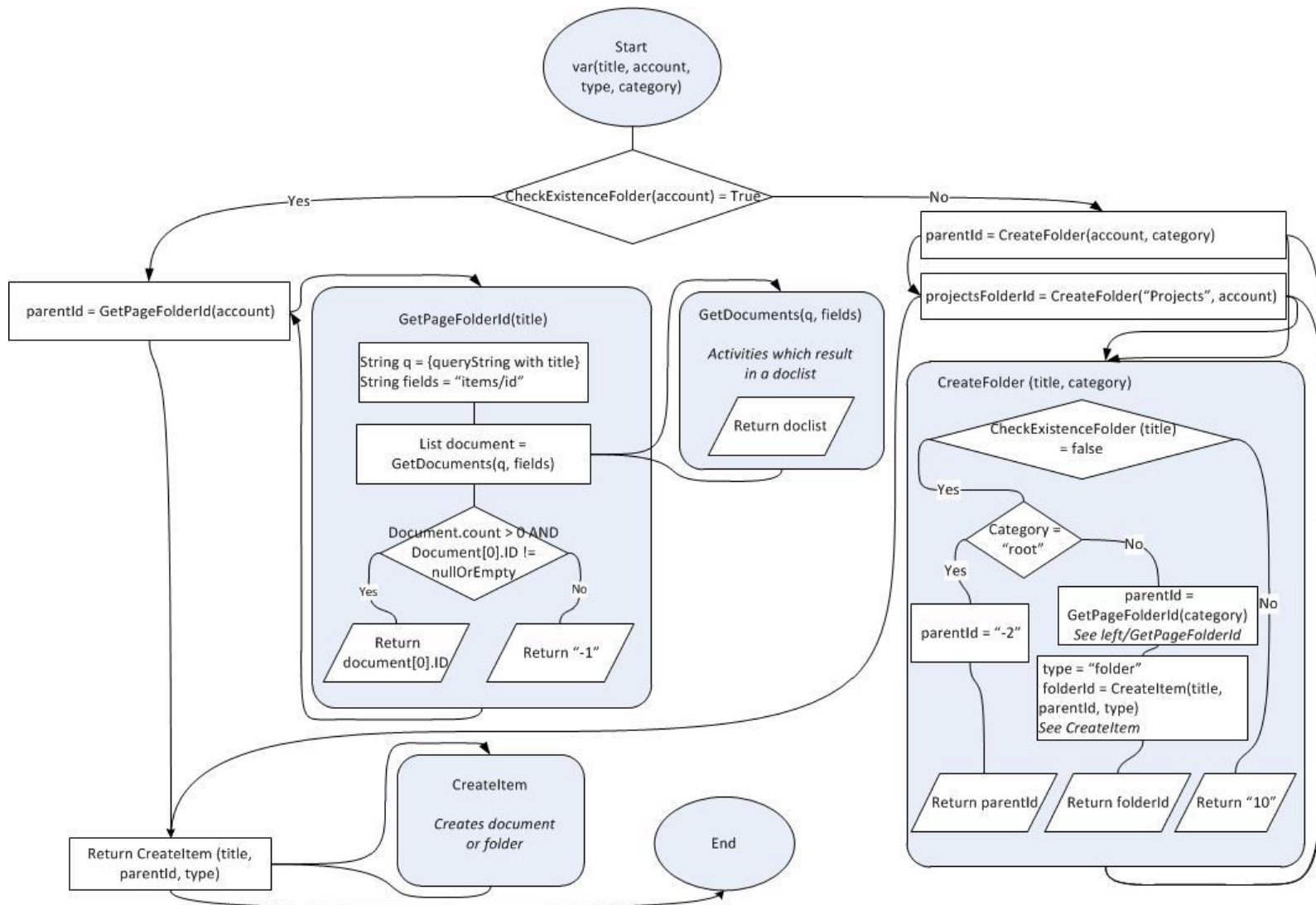
8.5 Compose the test scripts

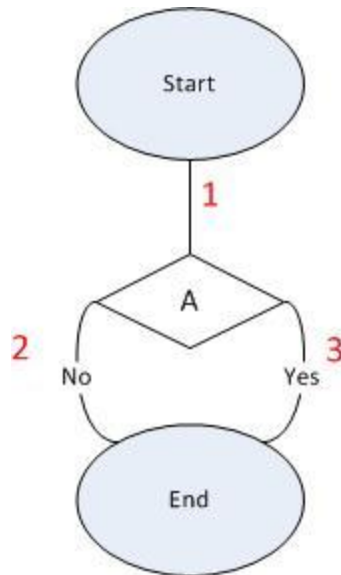
The result of the above steps can be found in the appendices, under the names:

- ShowAll_AccountFolderExists_UrlToAccountFolder;
- ShowAll_AccountFolderNotExists_UrlToAppFolder

9. Test CreateFile

9.1 Identifying the decision points





9.2 Determining the test paths

Decision point	Path
Start	1
A	2(end), 3(end)

Account-Folder-Already-Exists path: 1, 2

Account-Folder-Doesn't-Exist-Must-Be-Created path: 1, 3

9.3 Specifying the test cases

Account-Folder-Already-Exists path:

The folder "Growin Fietsen" must be in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts > Growin Fietsen). In the top of the test code file the following line must be edited:

```

Private Const ExistingAccountFolder As String = ""
Edit to: Private Const ExistingAccountFolder As String = "Growin Fietsen"

```

Account-Folder-Doesn't-Exist-Must-Be-Created:

The folder "FolderIntegrationTest" must not exist in the Google Drive account.

9.4 Determining the initial data

For the user "CustomerAdvanced" the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (active)
- GoogleUserId → {email address of Google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

When these fields aren't filled, you have to register either through the webbrowser (like a real user would do) or by successfully perform one of the following tests:

- SetupService_NoBaseFolders_ServicesSetUpAndFoldersAreCreated;
- SetupService_BaseFoldersAlreadyExist_ServicesSetUpAndFoldersAreNotCreated.

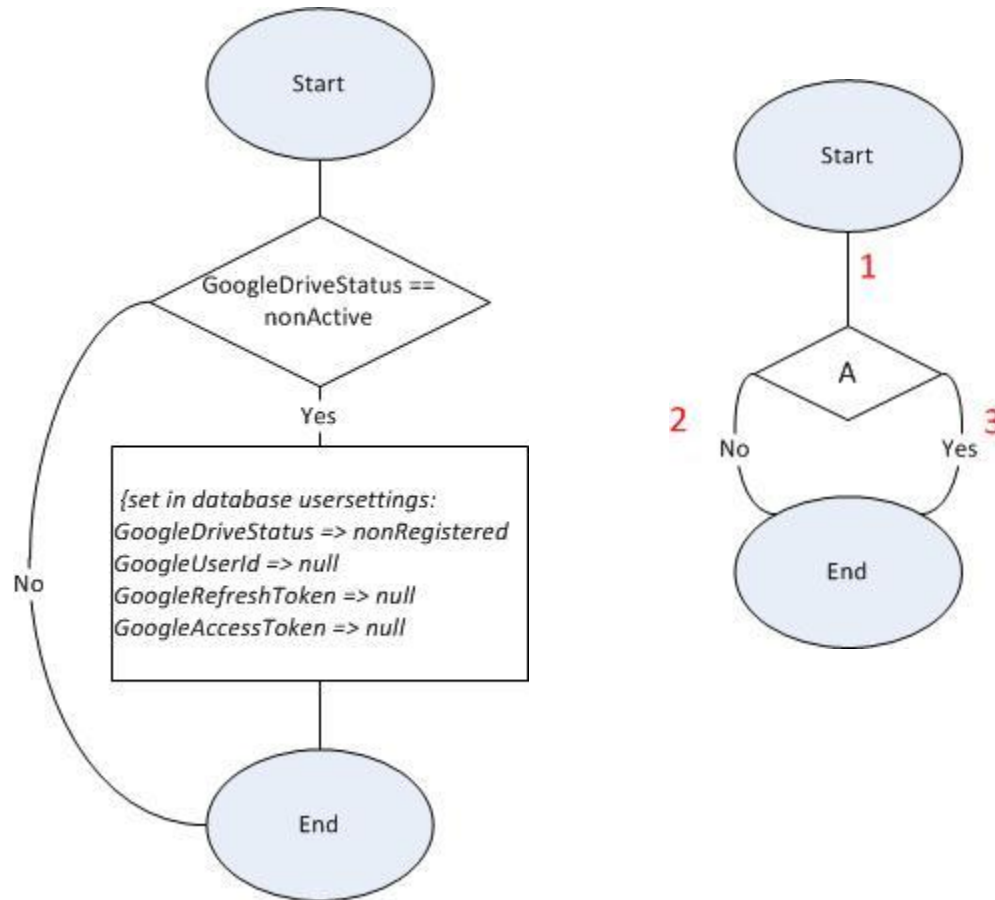
9.5 Compose the test scripts

The result of the above steps can be found in the appendices, under the names:

- CreateFile_AccountFolderExists_CreateFileInAccountFolder;
- CreateFile_NoAccountFolder_CreateFileAndAccountFolder.

11. Test Revoke

11.1 Identifying the decision points



11.2 Determining the test paths

Decision point	Path
Start	1
A	2(end), 3(end)

GoogleDriveStatus-nonActive path: 1, 3

GoogleDriveStatus-not-nonActive path: 1, 2

11.3 Specifying the test cases

GoogleDriveStatus-nonActive path:

The registration of the integration of Google needs to be nonActive, see 11.4.

GoogleDriveStatus-not-nonActive path:

The registration of the integration of Google needs to be Active, see 11.4.

11.4 Determining the initial data

GoogleDriveStatus-nonActive path:

For the user “CustomerAdvanced” the following fields need to be filled (in the database):

- GoogleDriveStatus → 1 (nonActive)
- GoogleUserId → {email address of Google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

When these fields aren’t filled, you have to register through the webbrowser (like a real user would do) and then deactivate in the usersettings. (Deactivating will be handled by the testcode).

GoogleDriveStatus-not-nonActive path:

For the user “CustomerAdvanced” the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (Active)
- GoogleUserId → {email address of Google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

When these fields aren’t filled, you have to register through the webbrowser (like a real user would do). (Activating will be handled by the testcode).

11.5 Compose the test scripts

The testscript for these tests can be found in the appendices under the name:

- Revoke_RevokedGoogleStatusNonActive_usersettingsDBemptyOrNonRegistered;
- Revoke_RevokedGoogleStatusActive_usersettingsStayTheSame.

Appendices

- A. Test script #1 SetupService_NoBaseFolders_ServiceIsSetUpAndFoldersAreCreated
- B. Test script #2 SetupService_BaseFoldersAlreadyExist_ServiceIsSetUpAndFoldersAreNotCreated
- C. Test script #3
RefreshAccessToken_WithRefreshTokenFromDBAccessTokenExpired_AccessTokenReceivedCorrectly
- D. Test script #4
RefreshAccessToken_WithRefreshTokenFromDBAccessTokenNotExpired_AccessTokenNotRefreshed
- E. Test script #5 GetFiles_AccountFolderExists_FilledDocList
- F. Test script #6 GetFiles_AccountFolderNotExist_EmptyDocList
- G. Test script #7 SearchFiles_AccountFolderExistsSearchValuePresent_FilledDocList
- H. Test script #8 SearchFiles_AccountFolderNotExistSearchValueAnything_EmptyDocList
- I. Test script #9 ShowAll_AccountFolderExists_UrlToAccountFolder
- J. Test script #10 ShowAll_AccountFolderNotExist_UrlToAppFolder
- K. Test script #11 CreateFile_AccountFolderExists_CreateFileInAccountFolder
- L. Test script #12 CreateFile_NoAccountFolder_CreateFileAndAccountFolder
- M. Test script #13 Revoke_RevokedGoogleStatusNonActive_usersettingsDBemptyOrNonRegistered
- N. Test script #14 Revoke_RevokedGoogleStatusActive_usersettingsStayTheSame

A. Test script #1

Identification

Identification (of test script):	SetupService_NoBaseFolders_ServiceIsSetUpAndFoldersAreCreated
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

1. Check the Google Drive account on whether or not the base folders ("ExactOnlineApp", "Exact-Accounts", "Exact-NonBillableProjects") exist. If they exist delete them.
2. Check if you aren't logged in to Exact Online, if so log out.
3. Check if you aren't logged in to a Google account, if so log out.
4. Go to the following url:
https://accounts.google.com/o/oauth2/auth?access_type=offline&client_id=703450183246-cdeht6oiilag000t46kjnsmlv0drvatb.apps.googleusercontent.com&redirect_uri=http://localhost:61422/&scope=https://www.googleapis.com/auth/drive+https://www.googleapis.com/auth/userinfo.email+https://www.googleapis.com/auth/userinfo.profile&response_type=code&approval_prompt=force&user_id=&state={'action:create'}
 Login in with:
 mdummyacc2@gmail.com, password: dummy12345
 Allow access to the Google Drive account.
 A page will appear with the message Unable to connect.
 Get the authorizationcode from the url in the browser address bar (the string after &code=) and paste it in the testcode:

```
Dim authCode As String = ""
```
5. You can execute the test now.

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/Control	Description		
1	Action	Executed test code		
2	Control	Basefolders now exist in the GoogleDrive account		

Scrubbing the test environment

After the test delete the authCode from the test code. End situation will look like this again:

```
Dim authCode As String = ""
```

B. Test script #2

Identification

Identification (of test script):	SetupService_BaseFoldersAlreadyExist_ServiceIsSetUpAndFoldersAreNotCreated
Execution date:	
Version test script:	1.0
Author:	Marlies Pikel (intern)

This test can best be tested directly after the SetupService-BaseFolders-don't-exist test is tested, because the base folders ("ExactOnlineApp", "Exact-Accounts", "Exact-NonBillableProjects") will exist when the SetupService-BaseFolders-don't-exist path test succeeded. Else create the base folders yourself (use the exact names like above) in the root folder of the Google Drive account you would like to use during testing.

Preparation of the initial situation

- Check the Google Drive account on whether or not the base folders ("ExactOnlineApp", "Exact-Accounts", "Exact-NonBillableProjects") exist. If they don't exist create them as followed:
 - Root > "ExactOnlineApp"
 - Root > ExactOnlineApp > "Exact-Accounts"
 - Root > ExactOnlineApp > "Exact-NonBillableProjects"
- Check if you aren't logged in to Exact Online, if so log out.
- Check if you aren't logged in to a Google account, if so log out.
- Go to the following url:

[https://accounts.google.com/o/oauth2/auth?access_type=offline&client_id=703450183246-cdeht6oiilag000t46kjnsmlv0drvatb.apps.googleusercontent.com&redirect_uri=http://localhost:61422/&scope=https://www.googleapis.com/auth/drive+https://www.googleapis.com/auth/userinfo.email+https://www.googleapis.com/auth/userinfo.profile&response_type=code&approval_prompt=force&user_id=&state={"action:create"}](https://accounts.google.com/o/oauth2/auth?access_type=offline&client_id=703450183246-cdeht6oiilag000t46kjnsmlv0drvatb.apps.googleusercontent.com&redirect_uri=http://localhost:61422/&scope=https://www.googleapis.com/auth/drive+https://www.googleapis.com/auth/userinfo.email+https://www.googleapis.com/auth/userinfo.profile&response_type=code&approval_prompt=force&user_id=&state={)

 - Login in with:


```
mdummyacc2@gmail.com, password: dummy12345
```

 Allow access to the Google Drive account.
 - A page will appear with the message Unable to connect.
 - Get the authorizationcode from the url in the browser address bar (the string after &code=) and paste it in the testcode:


```
Dim authCode As String = ""
```
- You can execute the test now.

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/ Control	Description		
1	Action	Executed test code		
2	Control	Basefolders exist in the GoogleDrive account (no doubles)		

Scrubbing the test environment

After the test delete the authCode from the test code. End situation will look like this again:

```
Dim authCode As String = ""
```

C. Test script #3

Identification

Identification (of test script):	RefreshAccessToken_WithRefreshTokenFromDBAccessTokenExpired_AccessTokenReceivedCorrectly
Execution date:	
Version test script:	1.0
Author:	Marlies Pikel (intern)

Preparation of the initial situation

For the user “CustomerAdvanced” the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (active)
- GoogleUserId → {emailaddress of google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/Control	Description		
1	Action	Executed test code		

Scrubbing the test environment

N/A

D. Test script #4

Identification

Identification (of test script):	RefreshAccessToken_WithRefreshTokenFromDBAccessTokenNotExpired_AccessTokenNotRefreshed
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

For the user “CustomerAdvanced” the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (active)
- GoogleUserId → {emailaddress of google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/ Control	Description		
1	Action	Executed test code		

Scrubbing the test environment

N/A

E. Test script #5

Identification

Identification (of test script):	GetFiles_AccountFolderExists_FilledDocList
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

1. The folder "Growin Fietsen" must be in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts > Growin Fietsen) and within this folder there's at least one file.
2. For the user "CustomerAdvanced" the following fields need to be filled (in the database):
 - GoogleDriveStatus → 2 (active)
 - GoogleUserId → {email address of Google account}
 - GoogleRefreshToken → {refreshToken}
 - GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/ Control	Description		
1	Action	Execute test code		

Scrubbing the test environment

N/A

F. Test script #6

Identification

Identification (of test script):	GetFiles_AccountFolderNotExist_EmptyDocList
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

1. The folder "Not Existing AccountFolder" must not exist in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts).
2. For the user "CustomerAdvanced" the following fields need to be filled (in the database):
 - GoogleDriveStatus → 2 (active)
 - GoogleUserId → {email address of Google account}
 - GoogleRefreshToken → {refreshToken}
 - GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/ Control	Description		
1	Action	Execute test code		

Scrubbing the test environment

N/A

G. Test script #7

Identification

Identification (of test script):	SearchFiles_AccountFolderExistsSearchValuePresent_FilledDocList
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

1. The folder "Growin Fietsen" must be in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts > Growin Fietsen) and within this folder there's at least one file with has in the title or content the word "Lorem". (It sometimes takes a few minutes for Google to update the document information)
2. For the user "CustomerAdvanced" the following fields need to be filled (in the database):
 - GoogleDriveStatus → 2 (active)
 - GoogleUserId → {email address of Google account}
 - GoogleRefreshToken → {refreshToken}
 - GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/Control	Description		
1	Action	Execute test code		

Scrubbing the test environment

N/A

H. Test script #8

Identification

Identification (of test script):	SearchFiles_AccountFolderNotExistsSearchValueAnything_EmptyDocList
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

1. The folder "Not Existing AccountFolder" must not exist in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts).
2. For the user "CustomerAdvanced" the following fields need to be filled (in the database):
 - GoogleDriveStatus → 2 (active)
 - GoogleUserId → {email address of Google account}
 - GoogleRefreshToken → {refreshToken}
 - GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/Control	Description		
1	Action	Execute test code		

Scrubbing the test environment

N/A

I. Test script #9

Identification

Identification (of test script):	ShowAll_AccountFolderExists_UrlToAccountFolder
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

1. The folder "Growin Fietsen" must be in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts > Growin Fietsen).
2. In de top of the test code file the following line must me edited:
`Private Const FolderIdExistingAccountFolder As String = ""`
 To find the FolderId van the folder "Growin Fietsen" you have to go to your Google Drive Account and open the folder "Growin Fietsen". The folderID can now be found in the navigation bar. (see Testplan for image)
3. For the user "CustomerAdvanced" the following fields need to be filled (in the database):
 - GoogleDriveStatus → 2 (active)
 - GoogleUserId → {email address of Google account}
 - GoogleRefreshToken → {refreshToken}
 - GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/Control	Description		
1	Action	Execute the test code		

Scrubbing the test environment

N/A

J. Test script #10

Identification

Identification (of test script):	ShowAll_AccountFolderNotExists_UrlToAppFolder
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

1. The folder "Not Existing AccountFolder" must not exist in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts).
2. In de top of the test code file the following line must me edited:
`Private Const FolderIdExactOnlineApp As String = ""`
 To find the FolderId van the folder "ExactOnlineApp" you have to go to your Google Drive Account and open the folder "ExactOnlineApp". The folderID can now be found in the navigation bar. (see Testplan for image)
3. For the user "CustomerAdvanced" the following fields need to be filled (in the database):
 - GoogleDriveStatus → 2 (active)
 - GoogleUserId → {email address of Google account}
 - GoogleRefreshToken → {refreshToken}
 - GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/Control	Description		
1	Action	Execute the test code		

Scrubbing the test environment

N/A

K. Test script #11

Identification

Identification (of test script):	CreateFile_AccountFolderExists_CreateFileInAccountFolder
Execution date:	
Version test script:	1.0
Author:	Marlies Pikel (intern)

Preparation of the initial situation

- The folder "Growin Fietsen" must be in the Google Drive account (My Drive > ExactOnlineApp > Exact-Accounts > Growin Fietsen). In de top of the test code file the following line must be edited:
`Private Const ExistingAccountFolder As String = ""`
 Edit to: `Private Const ExistingAccountFolder As String = "Growin Fietsen"`
- For the user "CustomerAdvanced" the following fields need to be filled (in the database):
 - GoogleDriveStatus → 2 (active)
 - GoogleUserId → {email address of Google account}
 - GoogleRefreshToken → {refreshToken}
 - GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/Control	Description		
1	Action	Execute the test code		
2	Control	Check if a new file is created in the "Growin Fietsen" folder with the name "CreateDocIntegrationTest2"		

Scrubbing the test environment

Delete the just created file from the "Growin Fietsen" folder in the Google Drive account.

L. Test script #12

Identification

Identification (of test script):	CreateFile_NoAccountFolder_CreateFileAndAccountFolder
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

1. The folder "FolderIntegrationTest" must not exists in the Google Drive account
2. For the user "CustomerAdvanced" the following fields need to be filled (in the database):
 - GoogleDriveStatus → 2 (active)
 - GoogleUserId → {email address of Google account}
 - GoogleRefreshToken → {refreshToken}
 - GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
Id	Action/ Control	Description	Pass/Fail	Remarks
1	Action	Execute test code		
2	Control	Check if a folder with the name "FolderIntegrationTest" is created with in there a file named "CreateDocIntegrationTest"		

Scrubbing the test environment

Delete the folder "FolderIntegrationTest" and the file named "CreateDocIntegrationTest".

M. Test script #13

Identification

Identification (of test script):	Revoke_RevokedGoogleStatusNonActive_usersettingsDBemptyOrNonRegistered
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

For the user “CustomerAdvanced” the following fields need to be filled (in the database):

- GoogleDriveStatus → 1 (nonActive)
- GoogleUserId → {email address of Google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/Control	Description		
1	Action	Execute test code		

Scrubbing the test environment

N/A

N. Test script #14

Identification

Identification (of test script):	Revoke_RevokedGoogleStatusActive_usersettingsStayTheSame
Execution date:	
Version test script:	1.0
Author:	Marlies Pekel (intern)

Preparation of the initial situation

For the user “CustomerAdvanced” the following fields need to be filled (in the database):

- GoogleDriveStatus → 2 (Active)
- GoogleUserId → {email address of Google account}
- GoogleRefreshToken → {refreshToken}
- GoogleAccessToken → {accessToken}

Actions and controls

			Logging	
			Pass/Fail	Remarks
Id	Action/ Control	Description		
1	Action	Execute test code		

Scrubbing the test environment

N/A

Bijlage J: User Guide

Exact

User guide integration Google Drive

Integration of Document Management SaaS'

Opdrachtgever: Ronald Dähne – Director Technology Cloud Solutions
Begeleider: Ad van der Hoeven – Senior Research Engineer
Datum: 7 februari 2012
Versie: 1.0



Auteur

E.M. Pekel
Student – 20064646
Informatica

Haagse Hogeschool – Den Haag

Begeleidend examiner: G.A. Mijharends

Tweede examiner: Dhr. B. Kuiper

Contents

1. Register integration Google Drive	2
2. Deactivate Google Drive integration	7
3. Activated Google Drive integration	9
4. See documents by account	11
5. Search documents	13
6. Add new document	14
7. See all document in Google Drive	16

1. Register integration Google Drive

1. When you are logged in to Exact Online, go to General in the menu and click on User settings. See red arrow in Figure 1.1.

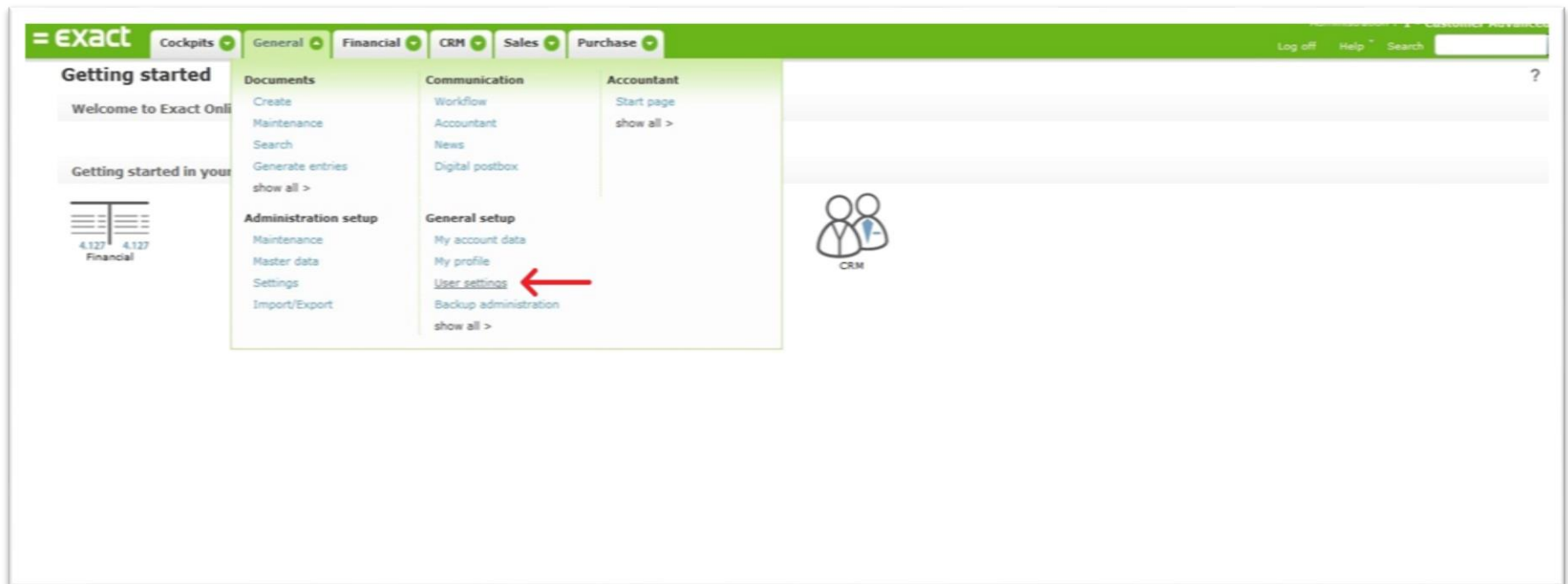


Figure 1.1.1 Homepage Exact Online after login

2. The user settings page will appear. See figure 1.2.

Administration : 1 - Customer Advanced

Log off Help Search

entry based on year/month instead of financial year/period ☐

Graph

Animation ☒

Palette Charcoal

Export

Which office software are you using? Microsoft Office

What to export All pages

CSV export delimiter Tab

Excel export with style formatting ☒

Dialog

Show errors in popup screen ☒

Show options dialog when exporting or printing ☐

My Outlook

Exchange server ☒ Microsoft Office 365 ☐ Microsoft Exchange 2010

User name

Password

Display: number of e-mail messages 3 Maximum: 30

Display: number of appointments 5 Maximum: 30

External SaaS Integration

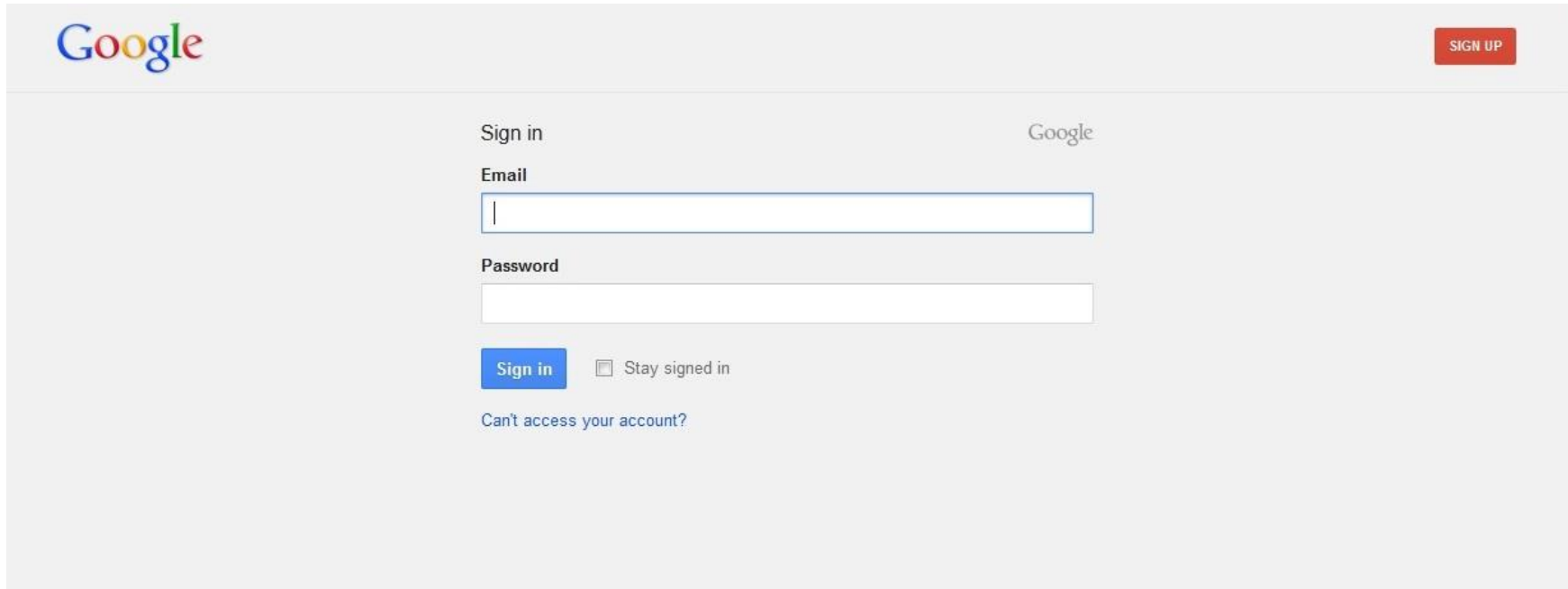
Google Integration status nonRegistered **Register your Google Drive**

Dropbox Integration status not yet available

Figure 1.2 User settings page in Exact Online

3. Click on the “Register your Google Drive” button. See figure 1.2.

4. You will be sent to Google to login in your Google Drive (if you're not logged in yet, if so go to step 5). See figure 1.3.

The image shows the Google Drive login page. At the top left is the Google logo. At the top right is a red 'SIGN UP' button. Below the logo, the text 'Sign in' is displayed. To the right of 'Sign in' is the word 'Google' in a smaller font. Below 'Sign in' is the label 'Email' followed by a text input field. Below the email field is the label 'Password' followed by a password input field. Below the password field is a blue 'Sign in' button. To the right of the 'Sign in' button is a checkbox labeled 'Stay signed in'. Below the 'Sign in' button and checkbox is a link that says 'Can't access your account?'.

Google

SIGN UP

Sign in Google

Email

Password

Sign in ☐ Stay signed in

[Can't access your account?](#)

Figure 1.3 Google Drive login page

5. For Exact Online to get access to your Google Drive account you have to click on “Allow access”. See figure 1.4.

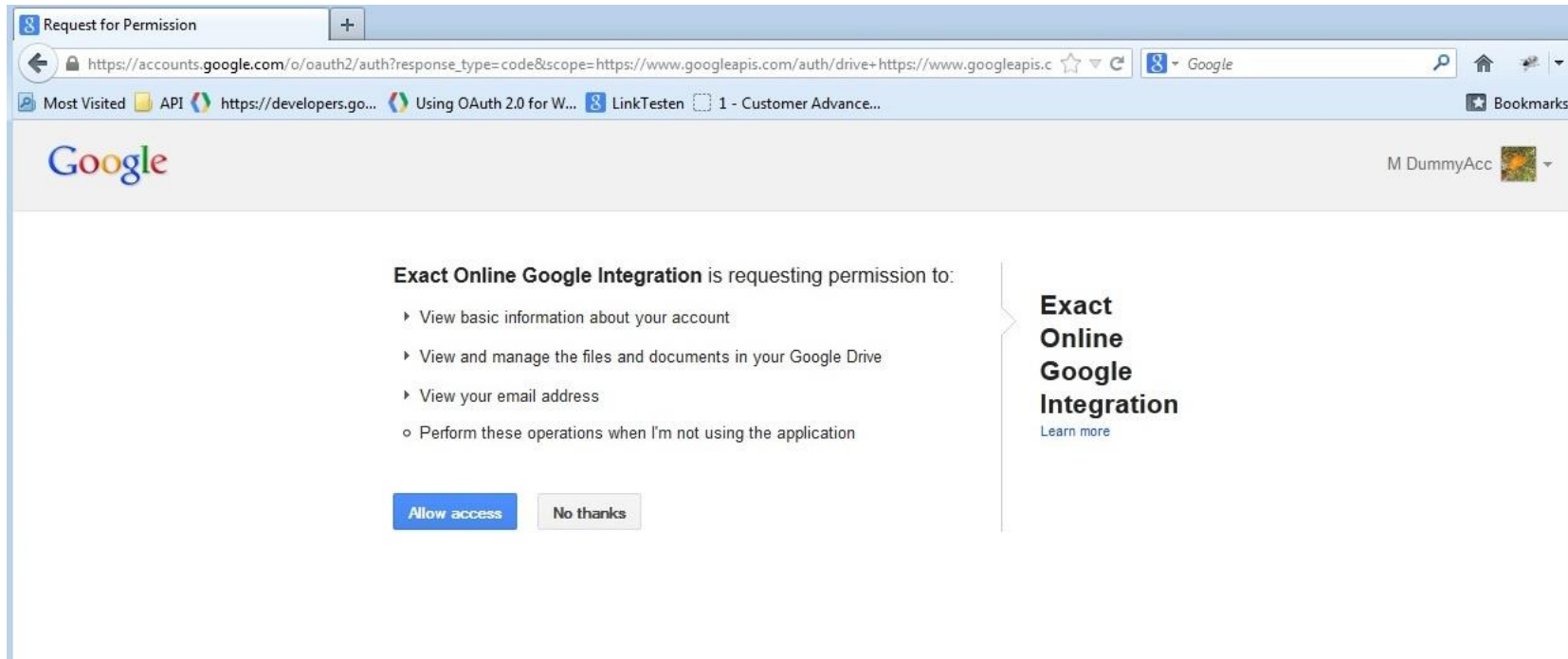


Figure 1.4 Request permission page to allow Exact Online access to your Google Drive account

6. You will be send back to Exact Online to the homepage.

7. Go to the user settings page again to see that your registration succeeded en the status of the integration with Google Drive is active. Behind the status the mail address of your account is shown. See figure 1.5.

The screenshot displays the 'exact' user settings interface. At the top, a green navigation bar includes the 'exact' logo and tabs for 'Cockpits', 'General', 'Financial', 'CRM', 'Sales', and 'Purchase'. The 'General' tab is selected. On the right of the bar, it shows 'Administration : 1 - Customer Advanced' along with 'Log off', 'Help', and a search bar.

The settings are organized into several sections:

- Graph:** Includes an 'Animation' checkbox (checked) and a 'Palette' dropdown menu set to 'Charcoal'.
- Export:** Contains options for 'Which office software are you using?' (Microsoft Office), 'What to export' (All pages), 'CSV export delimiter' (Tab), and 'Excel export with style formatting' (checked).
- Dialog:** Includes 'Show errors in popup screen' (checked) and 'Show options dialog when exporting or printing' (unchecked).
- My Outlook:** Features an 'Exchange server' selection (Microsoft Office 365 selected), fields for 'User name' and 'Password', and display counts for 'number of e-mail messages' (3) and 'number of appointments' (5), both with a 'Maximum: 30' limit.
- External SaaS Integration:** Shows the 'Google Integration status' as 'active' with a 'Deactivate Google Drive' button and the email 'mailaddress@gmail.com'. The 'Dropbox Integration status' is listed as 'not yet available'.

Figure 1.5 User settings page when the registration succeeded and the integration is activated

2. Deactivate Google Drive integration

1. When you are logged in to Exact Online, go to General in the menu and click on User settings. See red arrow in Figure 2.1.

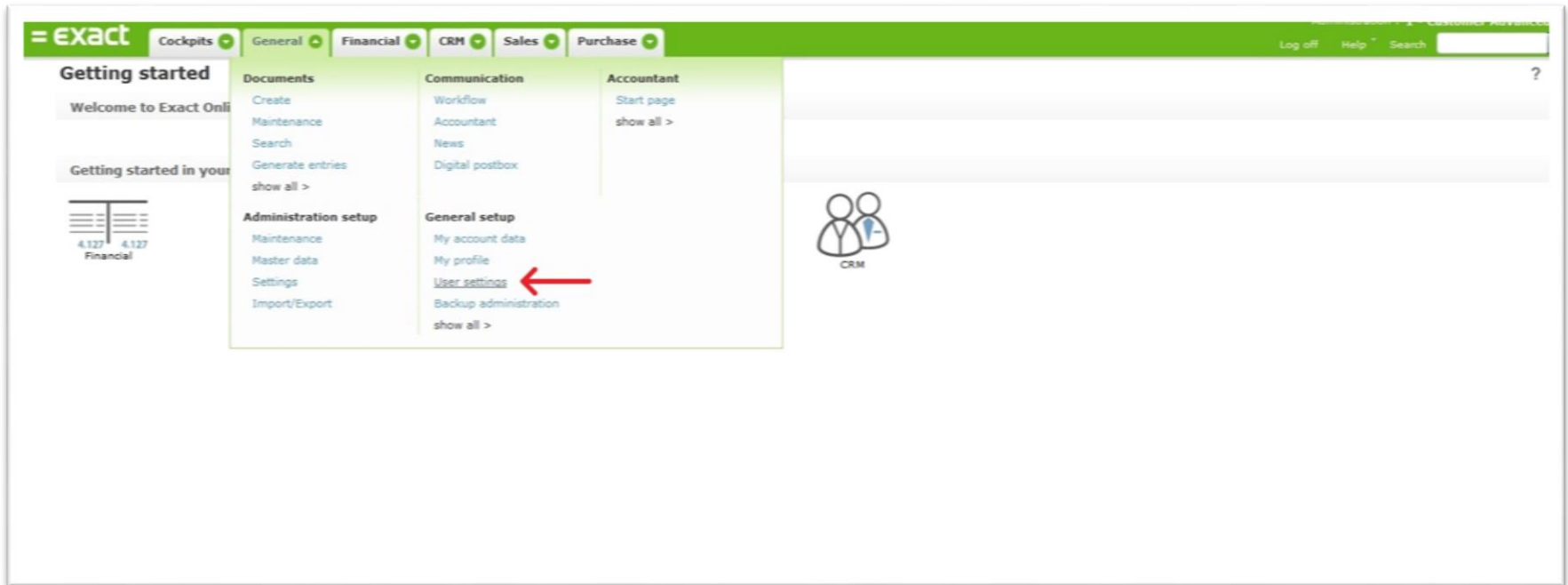


Figure 2.1 Homepage Exact Online after login

2. The user settings page will appear with the Google Drive status being “Active”. See figure 2.2.



Figure 2.2 Google Drive Integration status active on the user settings page

3. Click on the button “Deactivate Google Drive”
4. The status of the Google Drive Integration will be set to nonActive and the user settings page will refresh.

3. Activated Google Drive integration

1. When you are logged in to Exact Online, go to General in the menu and click on User settings. See red arrow in Figure 3.1.

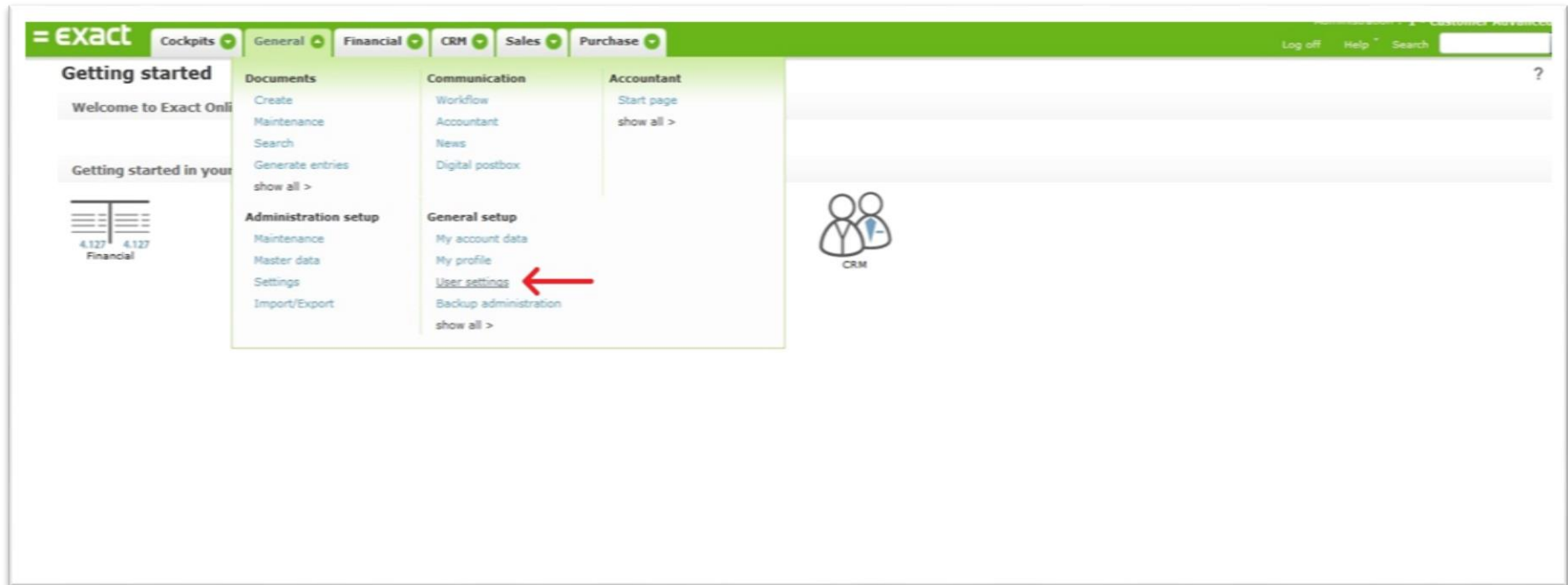


Figure 3.1 Homepage Exact Online after login

2. The user settings page will appear with the Google Drive status being “nonActive”. See figure 3.2.

External SaaS Integration			
Google Integration status	nonActive	Activate GoogleDrive	Revoke Access to Google mailaddress@gmail.com
Dropbox Integration status	not yet available		

Figure 3.2 Google Integration Drive status nonActive on the user settings page

3. Click on the “Activate Google Drive” button.
4. The status of the Google Drive Integration will be set to active and the user settings page will refresh.

4. See documents by account

When the integration is active and there's an account folder in Google Drive with the same name as the account in Exact Online, you can view your documents from Google Drive in the account card. For this follow the following steps:

1. Go to the account for which you want to view the Google Drive document.
2. In the account overview(see figure 4.1) there's a new block with the name Drive Docs, which contents up to fifty of your documents corresponding to the account. Or if you selected Drive Docs in the menu on the left you will see only the Drive Docs block. (see figure 4.2)

The screenshot shows the 'exact' online interface. The top navigation bar includes the 'exact' logo and several tabs: Cockpits, General, Financial, CRM, Sales, and Purchase. On the right side of the top bar, there are links for 'Log off', 'Help', and a 'Search' field. Below the top bar, the 'Activities' section is visible, followed by the 'Documents' section. The 'Drive Docs' section is highlighted, showing a search bar and a 'Search' button. Below this, a table lists documents with the following data:

Subject	Type	Created
Sponsoroverleg voor wielerseizoen 2013	document	01/31/2013 11:44:06
2010-Q4	spreadsheet	01/31/2013 11:45:29
2010-Q3	spreadsheet	01/31/2013 11:45:16
2010-Q2	spreadsheet	01/31/2013 11:44:58
2010-Q1	spreadsheet	01/31/2013 11:44:41
PersoneelsOverleg 12-12-2012	document	01/31/2013 11:42:39
Groothandel doc	document	01/07/2013 15:22:56

4.1 Documents in account overview

Cockpits
General
Financial
CRM
Sales
Purchase

Administration : 1 - Customer Advanced
Log off
Help
Search

Account: 3 - Groothandel sportfietsen

Overview: Accounts > Account: 3 - Groothandel sportfietsen

Created: Customer Advanced 10-02-2012 11:11 Modified: Customer Advanced 10-02-2012 11:11

Edit
New
Delete
Deduplication
Previous
Next
Close

Groothandel sportfietsen

Postbus 9
2831AA Gouderak
Zuid-Holland, The Netherlands ([Map](#) / [vCard](#))

Summary

There are no transactions.

The transaction overview shows you a summary of outstanding items, ongoing sales and purchases and open opportunities.

Activity stream

There are no activities.

The activity stream automatically shows you the last created tasks, appointments and opportunities.

Overview
General
Activities (0)
Documents (0)
Drive Docs
External References
Financial
Extra services

Drive Docs
AddNew | ShowAll

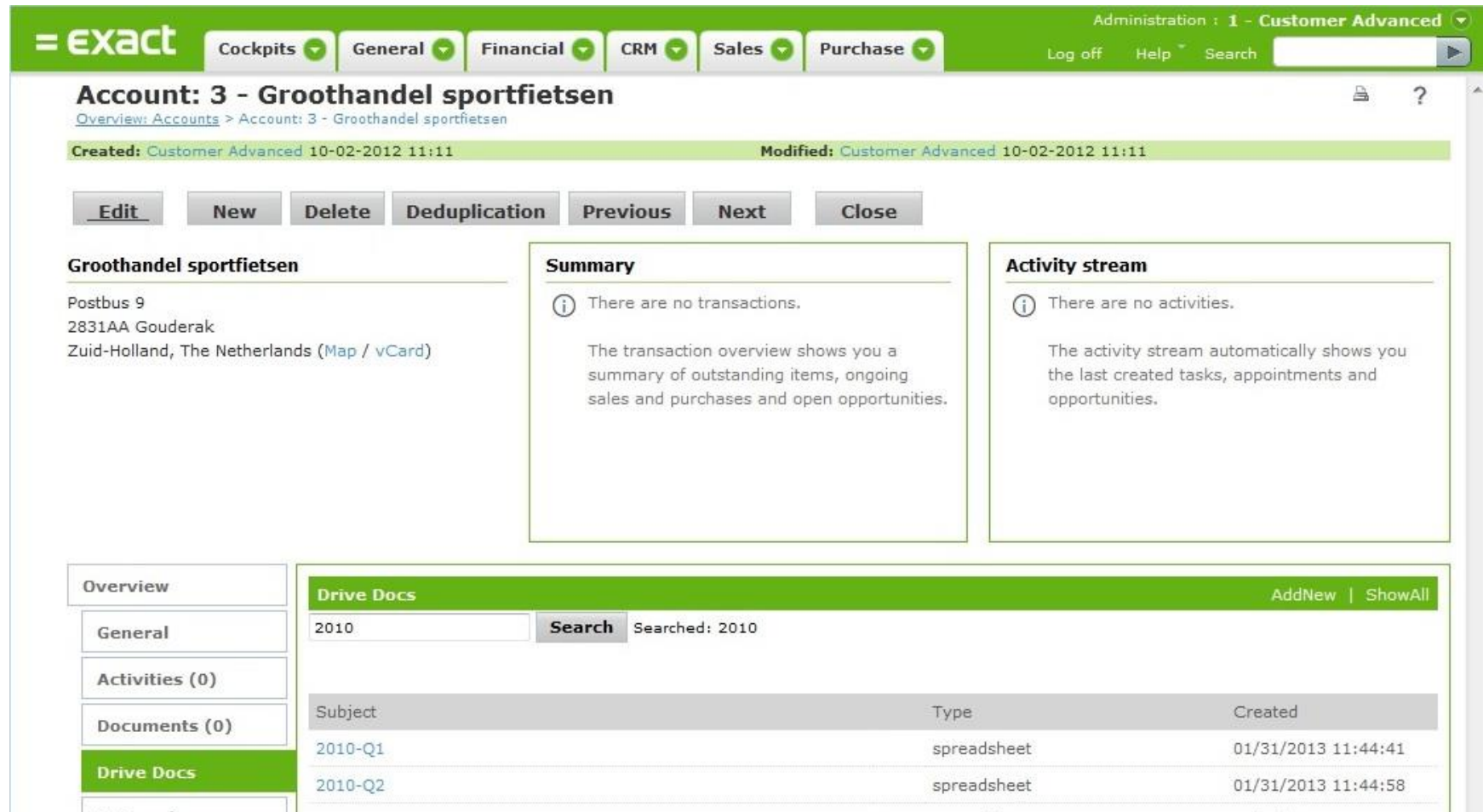
Search

Subject	Type	Created
Sponsoroverleg voor wielerseizoen 2013	document	01/31/2013 11:44:06
2010-Q4	spreadsheet	01/31/2013 11:45:29
2010-Q3	spreadsheet	01/31/2013 11:45:16
2010-Q2	spreadsheet	01/31/2013 11:44:58
2010-Q1	spreadsheet	01/31/2013 11:44:41
PersoneelsOverleg 12-12-2012	document	01/31/2013 11:42:39
Groothandel doc	document	01/07/2013 15:22:56

4.2 Drive docs block when selected in the menu on the left

5. Search documents

When you're on the account card (overview or the drive docs block) you can search for document in Google Drive within the represented folder. For example a search on "2010" (see figure 5.1)



exact Administration : 1 - Customer Advanced

Account: 3 - Groothandel sportfietsen

Overview: Accounts > Account: 3 - Groothandel sportfietsen

Created: Customer Advanced 10-02-2012 11:11 Modified: Customer Advanced 10-02-2012 11:11

[Edit](#) [New](#) [Delete](#) [Deduplication](#) [Previous](#) [Next](#) [Close](#)

Groothandel sportfietsen

Postbus 9
2831AA Gouderak
Zuid-Holland, The Netherlands ([Map](#) / [vCard](#))

Summary

There are no transactions.

The transaction overview shows you a summary of outstanding items, ongoing sales and purchases and open opportunities.

Activity stream

There are no activities.

The activity stream automatically shows you the last created tasks, appointments and opportunities.

Overview

General

Activities (0)

Documents (0)

Drive Docs

Drive Docs AddNew | ShowAll

2010 [Search](#) Searched: 2010

Subject	Type	Created
2010-Q1	spreadsheet	01/31/2013 11:44:41
2010-Q2	spreadsheet	01/31/2013 11:44:58

5.5.1 Search on "2010" in the accountfolder "Groothandel sportfietsen" with the results

6. Add new document

When you are on the account card that is need of a new document:

1. You can click on the AddNew button in the right corner of the Drive Docs block. See figure 6.1

The screenshot shows the 'exact' CRM interface. At the top, there's a navigation bar with tabs: Cockpits, General, Financial, CRM, Sales, and Purchase. The main header displays 'Account: 3 - Groothandel sportfietsen' with a lock icon and a help icon. Below this, it shows 'Created: Customer Advanced 10-02-2012 11:11' and 'Modified: Customer Advanced 10-02-2012 11:11'. A row of buttons includes Edit, New, Delete, Deduplication, Previous, Next, and Close.

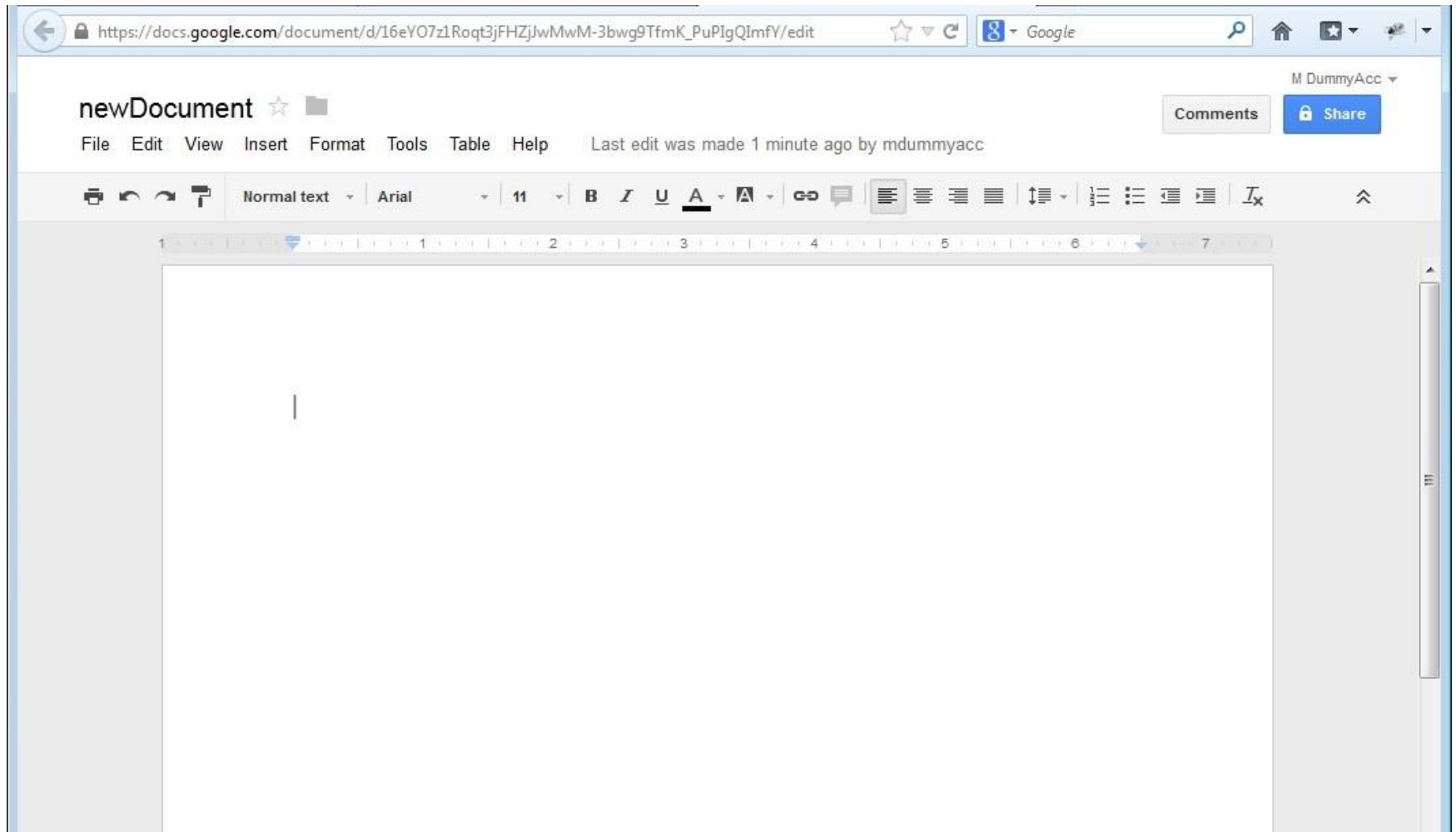
The account details for 'Groothandel sportfietsen' are listed: Postbus 9, 2831AA Gouderak, Zuid-Holland, The Netherlands (Map / vCard). To the right, there are two summary boxes: 'Summary' (stating 'There are no transactions') and 'Activity stream' (stating 'There are no activities').

At the bottom, there's a sidebar with navigation links: Overview, General, Activities (0), Documents (0), Drive Docs (highlighted), External References, Financial, and Extra services. The 'Drive Docs' block is expanded, showing a table of documents. A red arrow points to the 'AddNew' button in the top right corner of this block.

Subject	Type	Created
Sponsoroverleg voor wielerseizoen 2013	document	01/31/2013 11:44:06
2010-Q4	spreadsheet	01/31/2013 11:45:29
2010-Q3	spreadsheet	01/31/2013 11:45:16
2010-Q2	spreadsheet	01/31/2013 11:44:58
2010-Q1	spreadsheet	01/31/2013 11:44:41
PersoneelsOverleg 12-12-2012	document	01/31/2013 11:42:39
Groothandel doc	document	01/07/2013 15:22:56

6.1 Account card Drive Docs block with a red arrow pointing to AddNew button

2. A new window will open with the new document opened in Google Drive. Here you can edit the content and title of the new document. See figure 6.2.

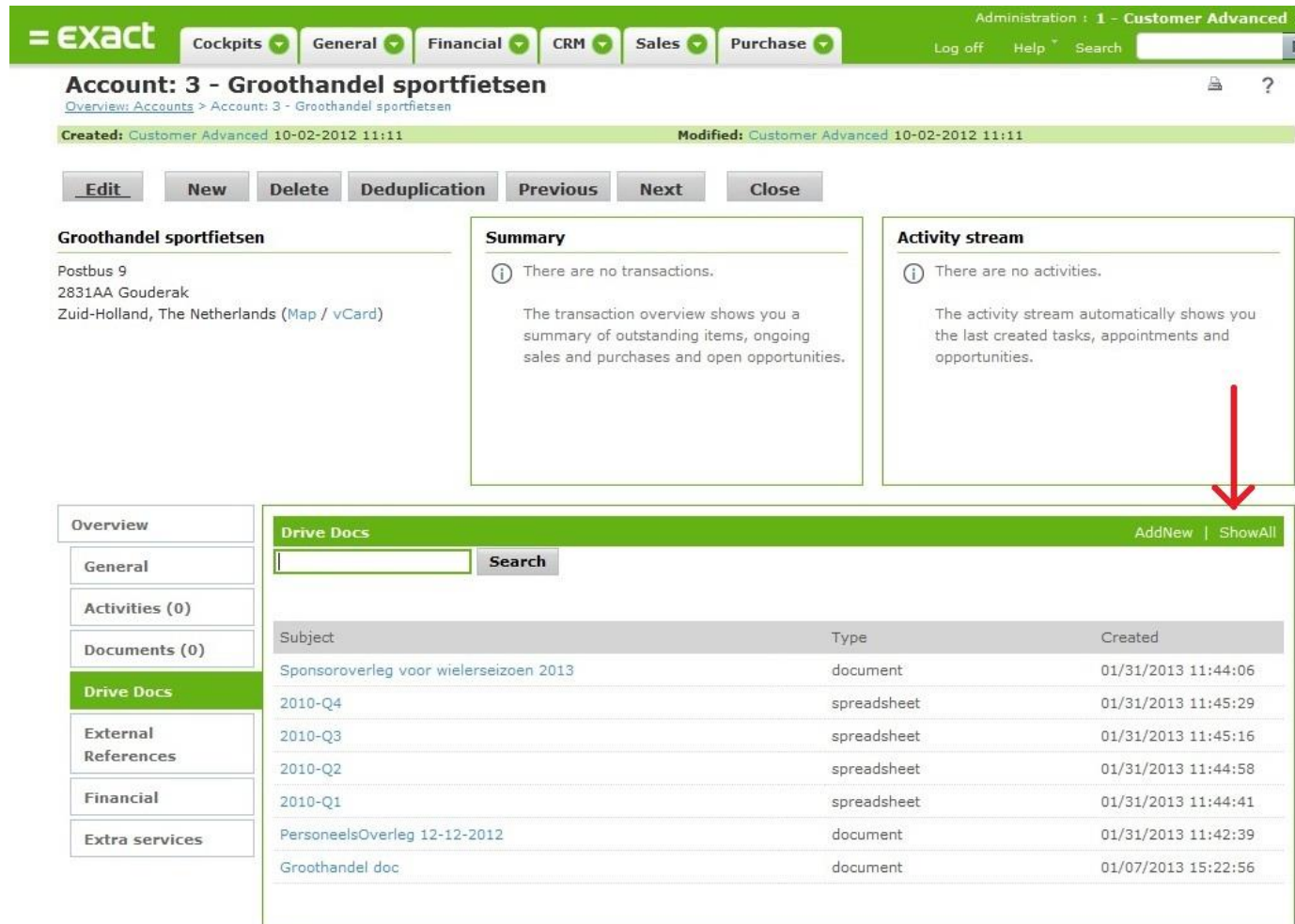


6.2 New document opened in Google Drive for editing

7. See all document in Google Drive

When you are on the account card and want to see more than fifty documents or just want to go to Google Drive to the account folder:

1. You can click on the ShowAll button in the right corner of the Drive Docs block. See figure 7.1



The screenshot shows the 'exact' CRM interface. At the top, there's a navigation bar with tabs for Cockpits, General, Financial, CRM, Sales, and Purchase. Below this, the account card for 'Groothandel sportfietsen' is displayed. The card includes a summary section with a message 'There are no transactions.' and an activity stream section with a message 'There are no activities.' A red arrow points to the 'ShowAll' button in the 'Drive Docs' block.

Account: 3 - Groothandel sportfietsen
 Overview: Accounts > Account: 3 - Groothandel sportfietsen
 Created: Customer Advanced 10-02-2012 11:11 Modified: Customer Advanced 10-02-2012 11:11

Groothandel sportfietsen
 Postbus 9
 2831AA Gouderak
 Zuid-Holland, The Netherlands (Map / vCard)

Summary
 There are no transactions.
 The transaction overview shows you a summary of outstanding items, ongoing sales and purchases and open opportunities.

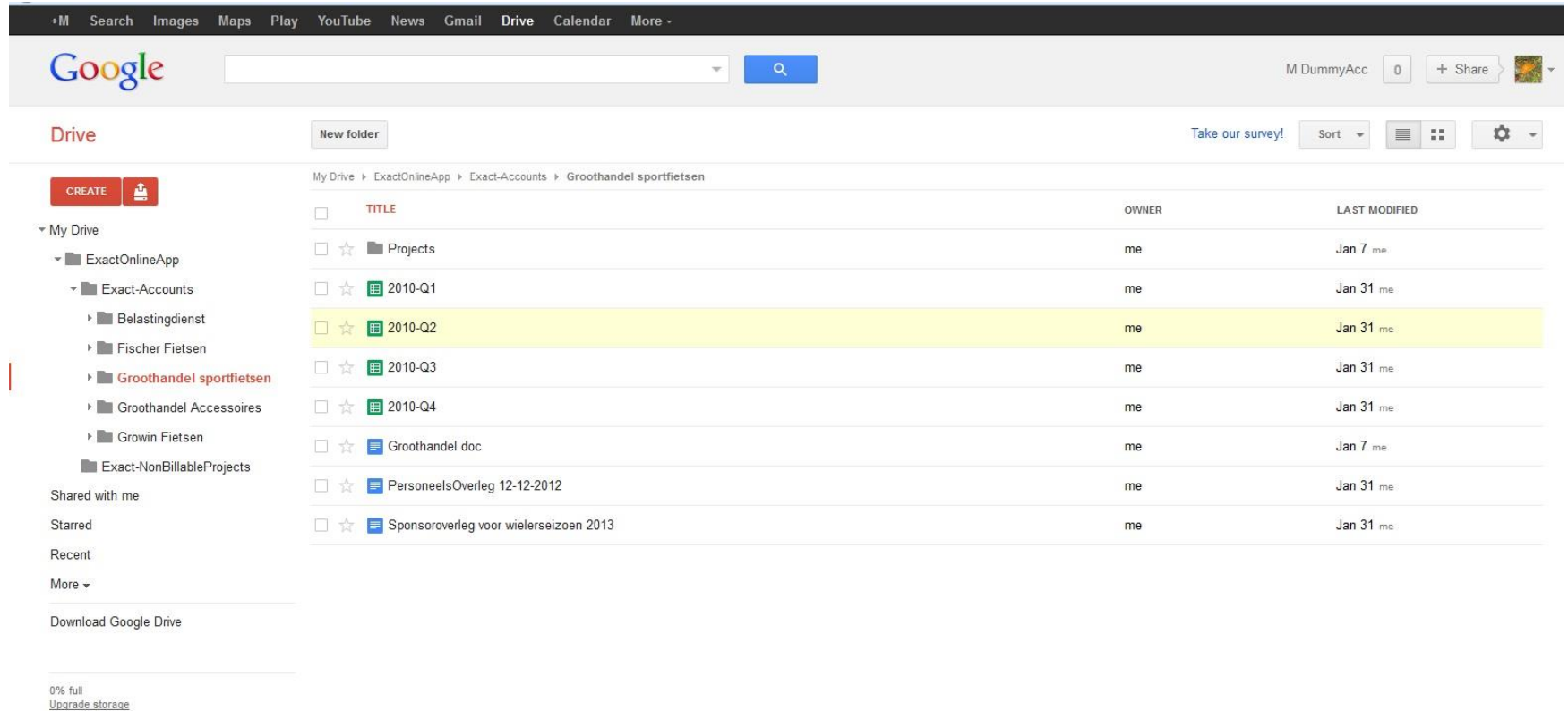
Activity stream
 There are no activities.
 The activity stream automatically shows you the last created tasks, appointments and opportunities.

Drive Docs AddNew | ShowAll

Subject	Type	Created
Sponsoroverleg voor wielerseizoen 2013	document	01/31/2013 11:44:06
2010-Q4	spreadsheet	01/31/2013 11:45:29
2010-Q3	spreadsheet	01/31/2013 11:45:16
2010-Q2	spreadsheet	01/31/2013 11:44:58
2010-Q1	spreadsheet	01/31/2013 11:44:41
PersoneelsOverleg 12-12-2012	document	01/31/2013 11:42:39
Groothandel doc	document	01/07/2013 15:22:56

7.1 Account card Drive Docs block with a red arrow pointing to the ShowAll button

2. A new window will open with the corresponding account folder in Google Drive. See figure 7.2.



7.2 Google Drive account folder in new window after clicking the “ShowAll” button

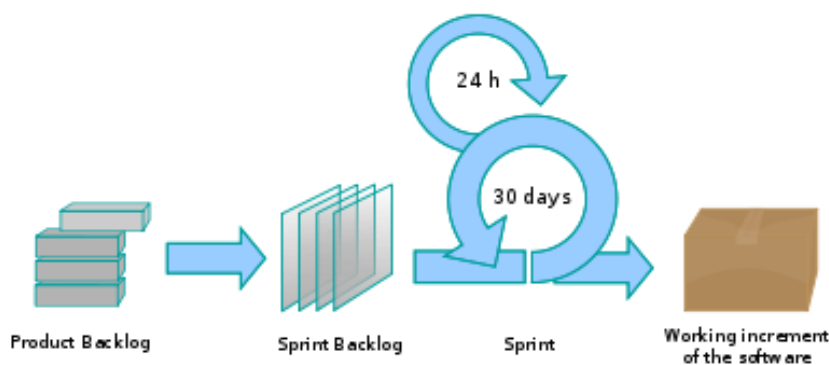
Bijlage K: De basis van SCRUM

De basis van SCRUM

SCRUM is een iteratieve en incrementele agile software ontwikkelings framework voor het managen van software projecten en producten of applicatie ontwikkeling. Software ontwikkeling met behulp van SCRUM gebeurt in kleine stukjes waarin iedere keer wordt voortgebouwd op de eerder gebouwde stukken (zie figuur 3-1). Het ontwikkelen van software in kleine stukjes bevordert de creativiteit en maakt het voor teams mogelijk snel te reageren op feedback en het kunnen veranderen van koers, indien nodig. Hierdoor wordt alleen gebouwd wat echt nodig is.

Een SCRUM project begint met het bepalen van de requirements, welke samen komen in de “backlog”. Een backlog is een lijst met de wensen en eisen voor het project met per userstory de waarde voor het project (prioritering) en hoeveel inspanning er nodig is om het te ontwikkelen. De requirements zijn niet volledig uitgewerkt zoals je in andere softwareontwikkelingsmethoden vaak ziet in een functioneel ontwerp. Het doel van de backlog is het documenteren van de gewenste functionaliteiten zodat iedereen in het project een idee heeft wat de scope van het project is. De backlog wordt opgesteld door de Product Owner.

De backlog is dynamisch en heeft regelmatige “grooming” nodig. Met grooming wordt bedoeld het toevoegen van details, schatting van de werkinspanning en de volgorde van de userstories op de backlog. Dit is een doorlopend proces waarbij de product owner en het ontwikkelteam samenwerken aan de details van de backlog items. Het ontwikkelteam is eindverantwoordelijk voor alle schattingen, maar de product owner mag het ontwikkelteam hierin wel sturen. Grooming is een activiteit die gedurende een sprint plaatsvindt voor de volgende sprint of twee sprints later en neemt gewoonlijk niet meer dan 10% van de capaciteit van het ontwikkelteam in beslag.



0-1 Korte weergave proces SCRUM (bron: [http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development)))

Het ontwikkelen van de software met behulp van SCRUM gebeurt in iteraties, die binnen SCRUM sprints worden genoemd. Een sprint is een korte periode, variërend van een week tot een maand, waarin een vooraf vastgesteld aantal functionaliteiten uit de backlog worden ontwikkeld. De functionaliteiten die voor een sprint zijn gepland vormen samen een sprintbacklog.

Een sprint begint altijd met een meeting van alle teamleden. In deze meeting wordt bepaald wat in een sprint gemaakt moet worden, hoe het team dit gaat aanpakken en wat het uiteindelijke doel van

de sprint is. Wanneer het hele team hierover overeenstemming heeft bereikt, kan de sprint beginnen.

Elke dag wordt er 's ochtends een stand-up meeting gehouden, die ten doel heeft alle teamleden een goed beeld van de voortgang van de huidige sprint te geven. Elke lid van het team vertelt dan in het kort wat hij/zij de vorige werkdag heeft gedaan, wat hij/zij die dag gaat doen en meldt eventuele problemen die hij/zij is tegengekomen. Verder worden er tijdens de stand-up eventuele review¹- en testtaken, die nog niet zijn opgepakt, verdeeld.

Aan het einde van een sprint wordt een sprint review gehouden. Tijdens de sprint review wordt bepaald wat afgerond is en wat de voortgang van het complete project is. Het team zal de afgeronde functionaliteiten presenteren en aantonen dat het voldoet aan de vooraf opgestelde criteria. Hierop wordt de backlog en sprint backlog aangepast.

Naast de sprint review wordt er ook een sprint retrospective meeting gehouden waarin wordt besproken wat er goed ging tijdens de sprint en wat moet veranderen/verbeteren voor de volgende sprint(s).

Als uiteindelijk alle sprints doorlopen zijn is het project klaar. De functionaliteiten die in de backlog staan zijn allemaal afgerond en de klant heeft het product gekregen waar om gevraagd was. Binnen sommige bedrijven wordt SCRUM ook gebruikt bij projecten die geen specifieke einddatum hebben.

¹ Het kritisch laten beoordelen van je werk door een ander teamlid om zo vroegtijdig fouten of ongewenste implementaties op te sporen en te voorkomen dat deze in productie komen.

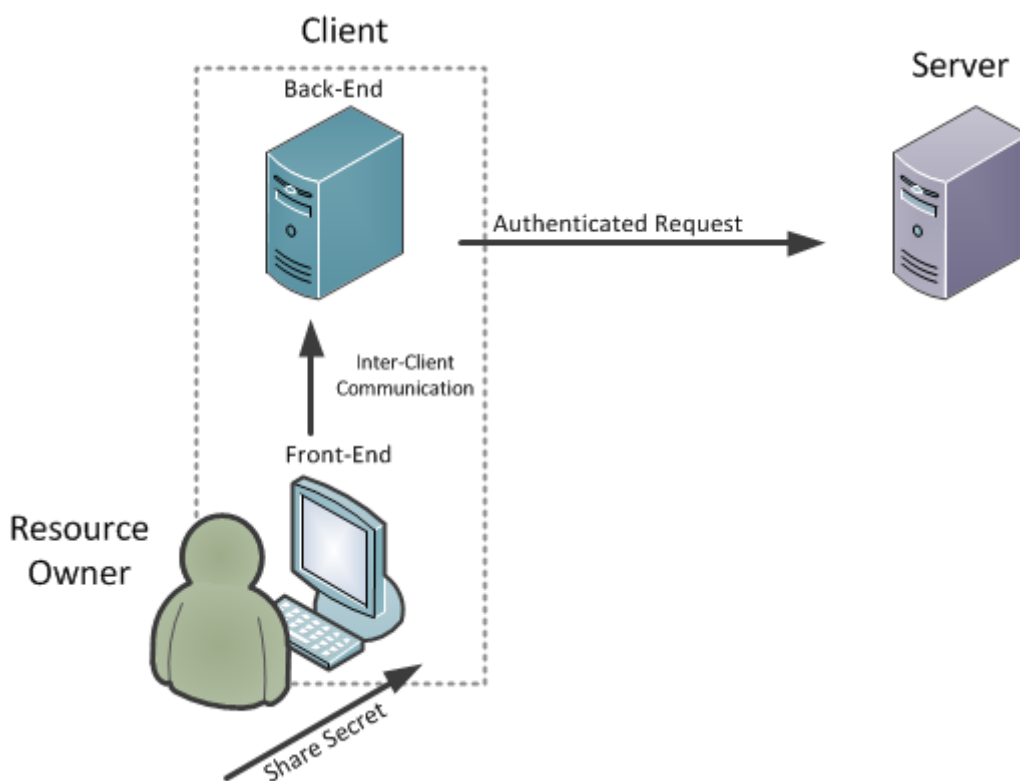
Bijlage L: Wat is OAuth

1. Wat is OAuth?

Websites maken steeds vaker gebruik van gedistribueerde services en cloud-computing. Bijvoorbeeld het laten afdrucken bij een online fotostudio van je foto's die je bij een online service als picasa of Flickr hebt staan.

Het probleem dat ze met OAuth hebben opgelost is het in staat stellen van derden om de gegevens van een gebruiker op te halen zonder dat de gebruiker zijn gebruikersnaam en wachtwoord moet opgeven aan de derde partij. De gebruiker geeft de derde partij toestemming tot gelimiteerde toegang(tijd, scoop) tot zijn gegevens met behulp van OAuth.

Een voorbeeld: Een gebruiker (gegevenseigenaar) geeft toestemming aan een fotostudio(client) om zijn privéfoto's van een fotoalbum servicesite(server) op te halen. Het ophalen van deze foto's gebeurt zonder het delen van de gebruikersnaam en wachtwoord. In plaats daarvan vindt de authenticatie bij het fotoalbum servicesite(server) plaats en krijgt de fotostudio(client) een zogenaamde token voor gelimiteerde toegang tot de privéfoto's. (Hammer, The OAuth1.0 Guide - Introduction , 2010)



1-1 Web-Based-Client OAuth

2-legged, 3-legged, n-legged

Onder het aantal legs("poten") van een OAuth-verzoek verstaan we het aantal partijen dat betrokken is bij het OAuth-verzoek. In een simpele OAuth flow, zoals hierboven afgebeeld (1-1), hebben we te maken met een 3-legged, bestaande uit: een client, een server en een gegevenseigenaar. In het geval dat de client ook de gegevenseigenaar is, is het een 2-legged constructie. (Hammer, The OAuth1.0 Guide - Terminology, 2009)

2. OAuth2.0

In dit hoofdstuk wordt beschreven hoe OAuth2.0 protocol in elkaar zit en hoe het werkt. Voor het beschrijven van het OAuth2.0 protocol gebruik ik citaten uit een document die te vinden is op de OAuth Community website(zie bronnen). Ik heb deze citaten niet vertaald naar het Nederlands vanwege de vele Engelse termen met betrekking tot het OAuth2.0 protocol.

2.1.Roles

OAuth defines four roles:

1. Resource owner

An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user.

2. Resource server

The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens.

3. Client

An application making protected resource requests on behalf of the resource owner and with its authorization. The term "client" does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices).

4. Authorization server

The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.

The interaction between the authorization server and resource server is beyond the scope of this specification. The authorization server may be the same server as the resource server or a separate entity. A single authorization server may issue access tokens accepted by multiple resource servers. (D. Hardt, 2012)

2.2.Protocol flow



2-1 Abstracte weergave OAuth2.0 flow (bron: (D. Hardt, 2012))

The abstract OAuth 2.0 flow illustrated in Figure 2-1 describes the interaction between the four roles and includes the following steps:

- (A) The client requests authorization from the resource owner. The authorization request can be made directly to the resource owner (as shown in figure 2-1), or preferably indirectly via the authorization server as an intermediary.
- (B) The client receives an authorization grant, which is a credential representing the resource owner's authorization, expressed using one of four grant types defined in this specification or using an extension grant type. The authorization grant type depends on the method used by the client to request authorization and the types supported by the authorization server.
- (C) The client requests an access token by authenticating with the authorization server and presenting the authorization grant.
- (D) The authorization server authenticates the client and validates the authorization grant, and if valid, issues an access token.
- (E) The client requests the protected resource from the resource server and authenticates by presenting the access token.
- (F) The resource server validates the access token, and if valid, serves the request.

The preferred method for the client to obtain an authorization grant from the resource owner (depicted in steps (A) and (B)) is to use the authorization server as an intermediary, which is illustrated in Figure 2-3 and described in paragraph 2.6. (D. Hardt, 2012)

2.3.Authorization Grant

An authorization grant is a credential representing the resource owner's authorization (to access its protected resources) used by the client to obtain an access token. This specification defines four grant types -- authorization code, implicit, resource owner password credentials, and client credentials -- as well as an extensibility mechanism for defining additional types. (D. Hardt, 2012)

2.3.1. Authorization Code

The authorization code is obtained by using an authorization server as an intermediary between the client and resource owner. Instead of requesting authorization directly from the resource owner, the client directs the resource owner to an authorization server (via its user-agent), which in turn directs the resource owner back to the client with the authorization code.

Before directing the resource owner back to the client with the authorization code, the authorization server authenticates the resource owner and obtains authorization. Because the resource owner only authenticates with the authorization server, the resource owner's credentials are never shared with the client.

The authorization code provides a few important security benefits, such as the ability to authenticate the client, as well as the transmission of the access token directly to the client without passing it through the resource owner's user-agent and potentially exposing it to others, including the resource owner. (D. Hardt, 2012)

2.3.2. Implicit

The implicit grant is a simplified authorization code flow optimized for clients implemented in a browser using a scripting language such as JavaScript. In the implicit flow, instead of issuing the client an authorization code, the client is issued an access token directly (as the result of the resource owner authorization). The grant type is implicit, as no intermediate credentials (such as an authorization code) are issued (and later used to obtain an access token).

When issuing an access token during the implicit grant flow, the authorization server does not authenticate the client. In some cases, the client identity can be verified via the redirection URI used to deliver the access token to the client. The access token may be exposed to the resource owner or other applications with access to the resource owner's user-agent.

Implicit grants improve the responsiveness and efficiency of some clients (such as a client implemented as an in-browser application), since it reduces the number of round trips required to obtain an access token. However, this convenience should be weighed against the security implications of using implicit grants, especially when the authorization code grant type is available. (D. Hardt, 2012)

2.3.3. Resource Owner Password Credentials

The resource owner password credentials (i.e., username and password) can be used directly as an authorization grant to obtain an access token. The credentials should only be used when there is a high degree of trust between the resource owner and the client (e.g., the client is part of the device operating system or a highly privileged application), and when other authorization grant types are not available (such as an authorization code).

Even though this grant type requires direct client access to the resource owner credentials, the resource owner credentials are used for a single request and are exchanged for an access token. This grant type can eliminate the need for the client to store the resource owner credentials for future use, by exchanging the credentials with a long-lived access token or refresh token. (D. Hardt, 2012)

2.3.4. Client Credentials

The client credentials (or other forms of client authentication) can be used as an authorization grant when the authorization scope is limited to the protected resources under the control of the client, or to protected resources previously arranged with the authorization server. Client credentials are used as an authorization grant typically when the client is acting on its own behalf (the client is also the resource owner) or is requesting access to protected resources based on an authorization previously arranged with the authorization server. (D. Hardt, 2012)

2.4. Access Token

Access tokens are credentials used to access protected resources. An access token is a string representing an authorization issued to the client. The string is usually opaque to the client. Tokens represent specific scopes and durations of access, granted by the resource owner, and enforced by the resource server and authorization server.

The token may denote an identifier used to retrieve the authorization information or may self-contain the authorization information in a verifiable manner (i.e., a token string consisting of some data and a signature). Additional authentication credentials, which are beyond the scope of this specification, may be required in order for the client to use a token.

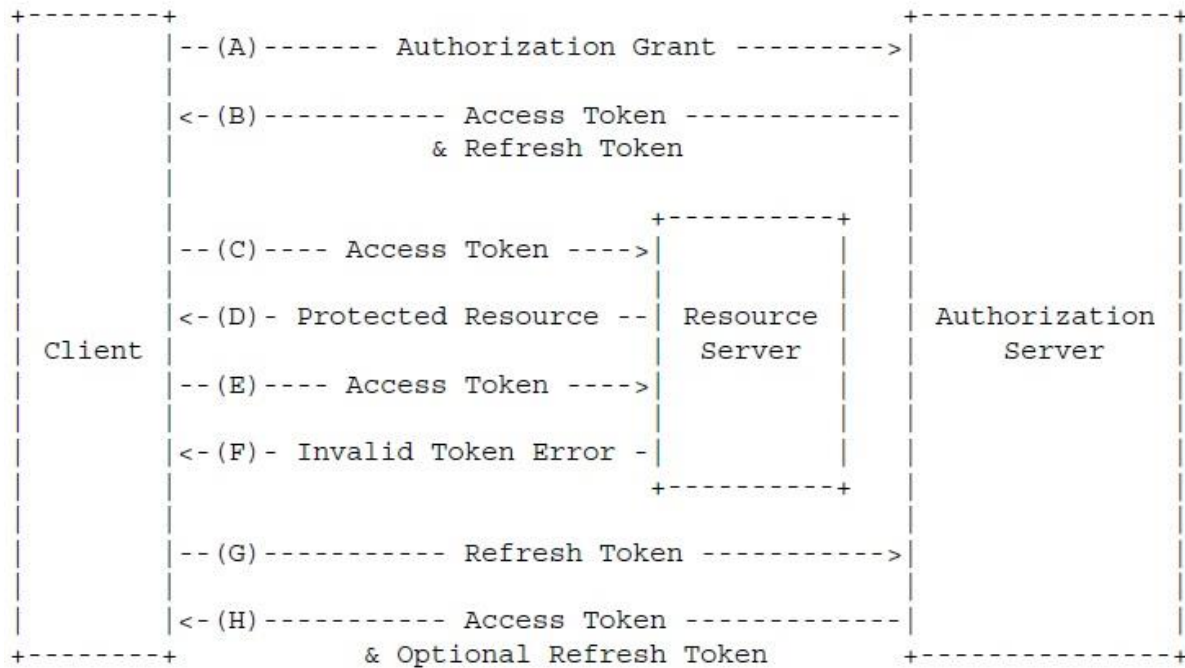
The access token provides an abstraction layer, replacing different authorization constructs (e.g., username and password) with a single token understood by the resource server. This abstraction enables issuing access tokens more restrictive than the authorization grant used to obtain them, as well as removing the resource server's need to understand a wide range of authentication methods.

Access tokens can have different formats, structures, and methods of utilization (e.g., cryptographic properties) based on the resource server security requirements. Access token attributes and the methods used to access protected resources are beyond the scope of this specification. (D. Hardt, 2012)

2.5. Refresh Token

Refresh tokens are credentials used to obtain access tokens. Refresh tokens are issued to the client by the authorization server and are used to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope (access tokens may have a shorter lifetime and fewer permissions than authorized by the resource owner). Issuing a refresh token is optional at the discretion of the authorization server. If the authorization server issues a refresh token, it is included when issuing an access token (i.e., step (D) in Figure 2-1).

A refresh token is a string representing the authorization granted to the client by the resource owner. The string is usually opaque to the client. The token denotes an identifier used to retrieve the authorization information. Unlike access tokens, refresh tokens are intended for use only with authorization servers and are never sent to resource servers.



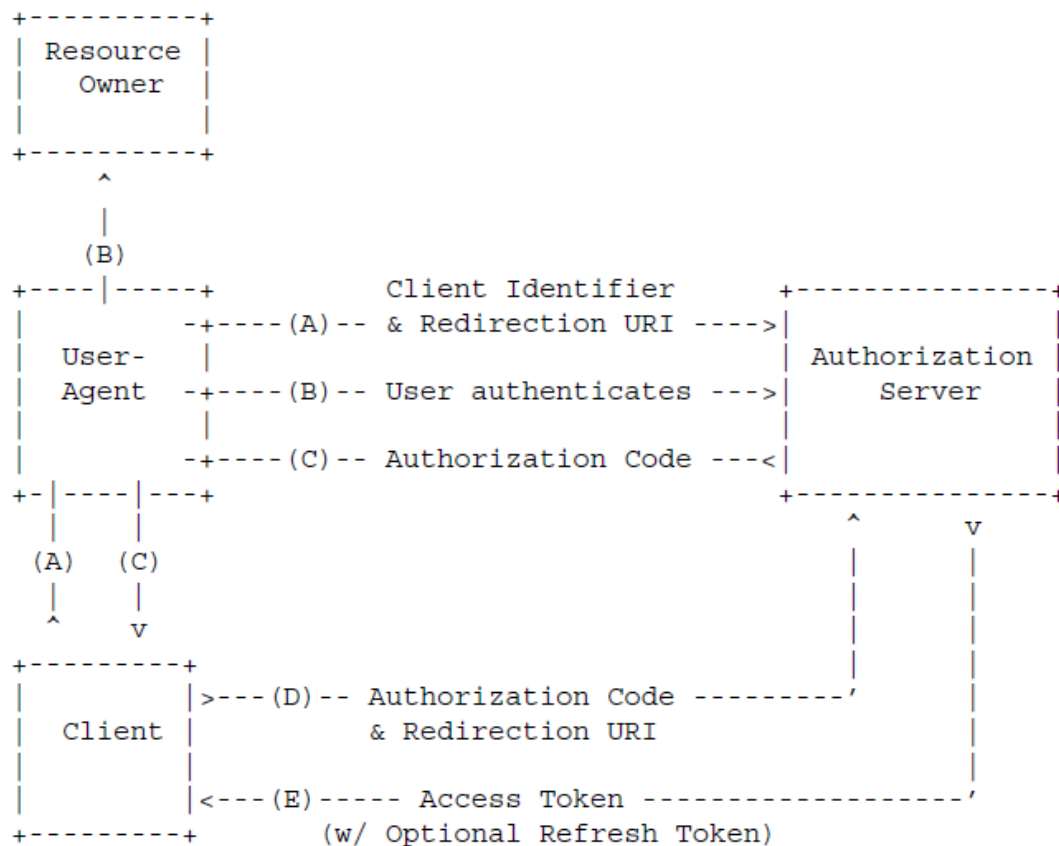
2-2 Refreshing an expired access token (bron: (D. Hardt, 2012))

The flow illustrated in Figure 2-2 includes the following steps:

- (A) The client requests an access token by authenticating with the authorization server and presenting an authorization grant.
 - (B) The authorization server authenticates the client and validates the authorization grant, and if valid, issues an access token and a refresh token.
 - (C) The client makes a protected resource request to the resource server by presenting the access token.
 - (D) The resource server validates the access token, and if valid, serves the request.
 - (E) Steps (C) and (D) repeat until the access token expires. If the client knows the access token expired, it skips to step (G); otherwise, it makes another protected resource request.
 - (F) Since the access token is invalid, the resource server returns an invalid token error.
 - (G) The client requests a new access token by authenticating with the authorization server and presenting the refresh token. The client authentication requirements are based on the client type and on the authorization server policies.
 - (H) The authorization server authenticates the client and validates the refresh token, and if valid, issues a new access token (and, optionally, a new refresh token).
- (D. Hardt, 2012)

2.6.Authorization Code Grant

The authorization code grant type is used to obtain both access tokens and refresh tokens and is optimized for confidential clients. Since this is a redirection-based flow, the client must be capable of interacting with the resource owner's user-agent (typically a web browser) and capable of receiving incoming requests (via redirection) from the authorization server.



Note: The lines illustrating steps (A), (B), and (C) are broken into two parts as they pass through the user-agent.

2-3 Authorization Code Flow (bron: (D. Hardt, 2012))

The flow illustrated in Figure 2-3 includes the following steps:

- (A) The client initiates the flow by directing the resource owner's user-agent to the authorization endpoint. The client includes its client identifier, requested scope, local state, and a redirection URI to which the authorization server will send the user-agent back once access is granted (or denied).
- (B) The authorization server authenticates the resource owner (via the user-agent) and establishes whether the resource owner grants or denies the client's access request.
- (C) Assuming the resource owner grants access, the authorization server redirects the user-agent back to the client using the redirection URI provided earlier (in the request or during client registration). The redirection URI includes an authorization code and any local state provided by the client earlier.
- (D) The client requests an access token from the authorization server's token endpoint by including the authorization code received in the previous step. When making the request, the client authenticates with the authorization server. The client includes the redirection URI used to obtain the authorization code for verification.
- (E) The authorization server authenticates the client, validates the authorization code, and ensures that the redirection URI received matches the URI used to redirect the client in step (C). If valid, the authorization server responds back with an access token and, optionally, a refresh token.

Bronnen:

D. Hardt, E. (2012, Oktober). *Documentation - OAuth*. Opgeroepen op Oktober-Maart 2012-2013, van Website OAuth.net: <http://tools.ietf.org/html/rfc6749>

Hammer, E. (2009, December 26). *The OAuth1.0 Guide - Terminology*. Opgeroepen op November 6, 2012, van Hueniverse: <http://hueniverse.com/oauth/guide/terminology/>

Hammer, E. (2010, Mei 15). *Introducing OAuth2.0*. Opgeroepen op November 6, 2012, van Hueniverse: <http://hueniverse.com/2010/05/introducing-oauth-2-0/>

Hammer, E. (2010, Mei 11). *The OAuth1.0 Guide - Introduction* . Opgeroepen op November 6, 2012, van Hueniverse: <http://hueniverse.com/oauth/guide/intro/>