

Afstudeerscriptie

Versie 1.0



Web-To-Print Module Concentrator

Afstudeerder:

Arno Becker, 1536748

Hogeschool Utrecht

Afstudeerbedrijf:

Direct-IT

Stage Periode:

01-02-2011 – 24-06-2011

INHOUDSOPGAVE

Voorwoord	4
1 Inleiding.....	5
1.1 Toelichting op de opdracht	5
1.1.1 Direct-IT.....	5
1.1.2 Concentrator	6
1.1.3 WebToPrint	6
2 Probleemstelling	7
2.1 WebToPrint	7
2.1.1 WebToPrint 1.0	7
2.1.2 WebToPrint 1.5	7
2.2 Opdracht Omschrijving	8
2.2.1 Probleemstelling	8
2.2.2 Doelstelling	9
2.2.3 Opdracht	9
3 Aanpak	10
3.1 De projectorganisatie.....	10
3.2 De Projectfasering.....	11
3.2.1 Featureonderzoek.....	13
3.2.2 Technologieonderzoek.....	13
3.2.3 Prototype Implementatie fase	13
3.2.4 Hoofd Implementatie fase	14
3.2.5 Testfase	14
3.2.6 Opleverfase	14
4 Uitwerking.....	15
4.1 Featureonderzoek.....	15
4.2 Technologieonderzoek.....	17
4.2.1 Designer	17
4.2.2 Printing.....	19
4.2.3 Dataformaten Designer.....	21
4.3 Prototype Implementatiefase	22
4.4 Hoofd Implementatiefase	24
4.4.1 Uitwerking Concentrator Module	24
4.4.2 Uitwerking Designer.....	25
4.5 Testfase	26

4.6 Opleverfase	26
5 Conclusies	27
5.1 Opgeleverde producten	27
5.1.1 De Designer	27
5.1.2 De Concentrator WebToPrint Module	30
5.1.3 De Concentrator Print Service	31
5.1.4 De Concentrator WebToPrint Administratie App	32
5.2 Project evaluatie	33
5.2 Wat ging goed	33
5.3 Wat kon beter	33
6 Bijlagen.....	34
6.1 MVVM UML Schema	34
6.2 MVC UML Schema.....	34
6.3 Promotiemateriaal	34
6.4 Plan Van Aanpak	34

VOORWOORD

Voor u ligt de afstudeerscriptie welke het resultaat is van de afstudeeropdracht die ik heb uitgevoerd als afsluiting van de opleiding Informatica aan de Hogeschool Utrecht. Deze afstudeeropdracht is uitgevoerd bij Diract-IT, gevestigd in Almere.

Om te beginnen wil ik André van Rijn bedanken voor het feit dat ik in een hele korte tijd een stageplek heb kunnen krijgen bij Diract. Mijn originele stageplek was plotseling verdwenen (commentaar van het bedrijf: “Er zijn helaas niet genoeg stageplekken beschikbaar”) en André kwam als redder in nood met een bericht dat ik wel een 2^e keer een stage kon lopen. André kende ik al sinds mijn vorige stage die ik van Augustus 2009 tot Januari 2010 heb gelopen. Omdat mijn vorige ervaring bij het bedrijf enorm positief was, heb ik dit aanbod ook zonder aarzelen geaccepteerd.

Het onderwerp van deze scriptie, de WebToPrint functie in de Concentrator, is eigenlijk een gedeeltelijke voortzetting van het werk wat ik in mijn eerste stage bij Diract heb gedaan. Tijdens mijn vorige stage heb ik aan de prijskaartjes tool gewerkt (Wat men eigenlijk de WebToPrint 1.0 kan noemen) en was het heel leuk om opnieuw met dit onderwerp bezig te zijn.

Verder wil ik Tim Zeeman, Stan Todorov, Casper Blokzijl en Duy Nguyen bedanken voor de immense ondersteuning tijdens het ontwikkelen van de WebToPrint module. Elk van mijn eindeloze stroom vragen werd beantwoord in een manier waarop ik uit mijn probleem kon komen, zelfs wanneer ze zelf een berg werk te doen hadden.

Tot slot wil ik mijn docent begeleiders Bernard Markus en Eelco Tienstra bedanken voor de begeleiding tijdens het project. De heer Markus moest helaas vervangen worden door de heer Tienstra wegens ziekte en ik wens hem daarom ook veel beterschap.

1 INLEIDING

Deze scriptie beschrijft het ontwerpen, bouwen en implementeren van een module voor de Concentrator service van Diract en het ontwerpen van een webbased programma om data direct vanuit de Concentrator tot documenten te verwerken.

1.1 Toelichting op de opdracht

In dit hoofdstuk zal een beschrijving van Diract worden gegeven, wat de Concentrator is en wat het doel van de opdracht is.

1.1.1 Diract-IT

De historie van Diract begint in 1997 met de kennismaking van de nu huidige eigenaren: André van Rijn en Mark Gerrits. In de jaren daaropvolgend hebben ze samengewerkt bij meerdere bedrijven, André vanuit operatie/commercie en Mark vanuit IT development. In 2006 zijn beide werkzaam voor de BAS groep en krijgen het idee om samen een software development bedrijf te starten: Producten, die ze in die jaren voor werkgevers hebben bedacht, zelf te gaan ontwikkelen en verkopen. BAS wilde rond dezelfde tijd de IT afdeling gaan outsourcen en dus word de gehele IT afdeling afsplitst in een losstaand bedrijf: Diract-IT (Diract = Direction & Action). De BAS groep blijft klant bij Diract, blijft in het pand van de BAS groep en deze behoudt zo de kennis en continuïteit.

Oktober 2007 is de formele start van het bedrijf en start met in totaal 5 mensen en één klant. Per april 2011 zijn dit 22 mensen en ruim 20 klanten en worden er bij DiractIT meerdere producten ontwikkeld, waaronder business intelligence pakketten voor managers om de resultaten van een bedrijf eenvoudig in kaart te brengen, de website en bijbehorende webshop van MyCom¹, Het Ceyenne warehouse management systeem wat in gebruik is bij BAS en magazijnen door heel Europa, een Point-Of-Sale systeem voor kassa's en uiteindelijk de Concentrator.

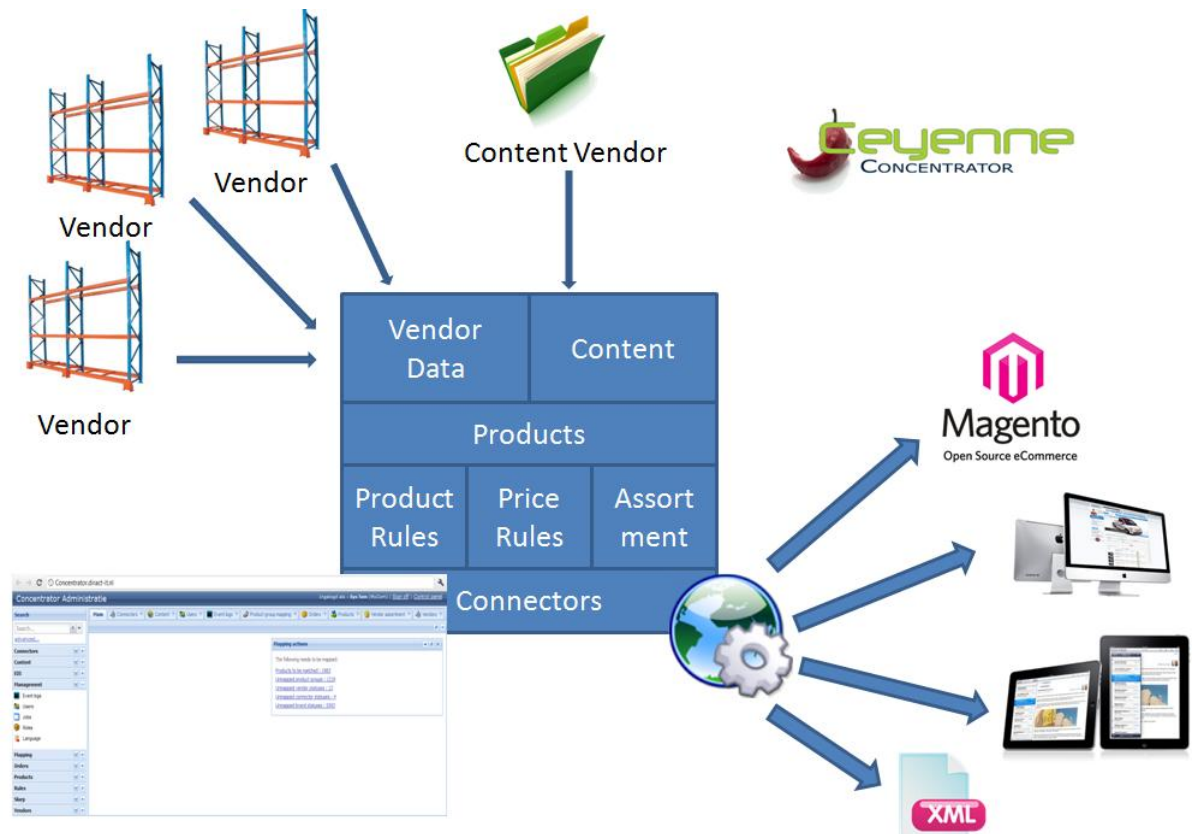


Figuur 1: Kantoor Diract in Almere

¹ MyCom is een electronica winkelketen, onderdeel van de BAS groep

1.1.2 Concentrator

De Concentrator is een software service die de productinformatie voor Ceyenne en verschillende andere pakketten bijhoudt. Dit gebeurt door het ophalen van informatie vanuit services zoals Icecat² en deze te verzamelen in een eigen database. Deze database bevat informatie over meer dan 100.000 producten en bevat gegevens als beschrijvingen, plaatjes, prijsgegevens, gerelateerde producten en specificaties. Deze data wordt dan verder gebruikt in producten zoals de MyCom webshop, om correcte en relevante informatie over producten weer te geven.



Figuur 2: De Concentrator

1.1.3 WebToPrint

De WebToPrint functie was toegevoegd aan de Concentrator in begin 2010, om MyCom en Vobis winkels de mogelijkheid te geven zelf prijskaartjes af te drukken. Tot dan werden deze kaartjes centraal afgedrukt en gedistribueerd naar winkels. Daardoor kostte het aanvragen van een nieuw kaartje meestal tussen de 2 en 3 weken. Hieruit was de vraag voor de WebToPrint functionaliteit ontstaan, want de winkels hadden bijvoorbeeld na acties of magazijnopruiming een overschot aan producten waarbij er geen prijskaartje voorhanden was en er met de hand iets gemaakt moest worden.

² Icecat.biz is een online database met producten en productinformatie

2 PROBLEEMSTELLING

Dit hoofdstuk behandelt de voorganger van het WebToPrint project. Hier uit is de vraag voor een nieuwe versie is ontstaan.

2.1 WebToPrint

Bij klanten van de Concentrator is er al lange tijd een vraag om direct data vanuit de Concentrator te kunnen verwerken naar afdrukken of eenvoudig verspreidbare nieuwsbrieven. Om dit te doen zijn er op dit moment twee mogelijkheden. De eerste optie is om gebruik te maken van de door Direct ontwikkelde Prijskaartjes tool, die informatie uit de Concentrator haalt om prijskaartjes voor de MyCom en Vobis winkels af te drukken. Het alternatief daarvoor is om manueel data kopiëren uit de Concentrator database naar documenten. Deze data wordt nu uit de database geëxporteerd naar bijvoorbeeld Excel lijsten en wordt vanuit daar manueel overgetypt naar de brochures. Helaas kost deze methode veel tijd en is vrij foutgevoelig.

2.1.1 WebToPrint 1.0

De eerste versie van het WebToPrint project was een simpele web-based prijskaartjes printing tool. Deze tool draaide in de browser, en liet de gebruiker direct vanuit de browser prints van prijskaartjes maken. Om deze prijskaartjes te maken werd een keuze uit een aantal voorgedefinieerde ontwerpen gemaakt en na invoer van een lijst van productnummers werden er dan een aantal prijskaartjes afgedrukt. Deze tool werd vooral in winkels gebruikt die een overvoorraad van een bepaald product hadden. Deze winkels hoefden dan niet via het centrale systeem prijskaartjes te bestellen, maar konden deze zelf afdrukken en direct gebruiken.

2.1.2 WebToPrint 1.5

De 2^e versie van WebToPrint was een aanpassing op het oude project, vanwege een aantal technische problemen in de winkels. Het originele project was geschreven in een combinatie van een ASP.NET server, javascript en HTML voor de pagina's en een Flash object om objecten binnen het prijskaartje (zoals de tekst, de marges en de grootte van het prijskaartje) aan te kunnen passen en uit te printen. Tijdens de grootschalige uitrol over de vestigingen van MyCom kwamen de winkeleigenaren er achter dat een aantal PC's geen Flash hadden, of enkel Internet Explorer 6, waarbij een aantal javascript fouten het programma onbruikbaar maakten. Versie 2 van het WebToPrint project verbeterde dit door PDF generatie toe te voegen, zodat winkels met computers zonder flash alsnog prijskaartjes konden maken, en daarna de PDF af konden drukken.

2.2 Opdracht Omschrijving

2.2.1 Probleemstelling

De Concentrator is een web-based beheerpakket voor het aanmaken, aanpassen en verwijderen van informatie over artikelen. Deze artikelinformatie wordt dan verder gebruikt in andere applicaties binnen het bedrijf van de gebruiker. De Concentrator bevat daarom informatie over meer dan 100.000 producten, en dit aantal groeit snel. Omdat dit pakket een gigantische hoeveelheid informatie bevat, die niet in andere pakketten aanwezig is, is er bij de gebruikers een vraag ontstaan om direct vanuit de applicatie prints van deze informatie te maken. De Concentrator extraheert data van verschillende bronnen en kan dit op slimme wijze met elkaar matchen. Hierdoor kunnen bedrijven hun data zeer consistent maken. Om die reden is de Concentrator een prima bron voor het invullen van brochures en andere product documenten. Op dit moment is de Concentrator in staat alle producten in verschillende formaten te exporteren, maar zijn design pakketten niet altijd geschikt voor het eenvoudig importen van data (zonder data te verliezen). Het zou een prima oplossing zijn wanneer Web to Print geïntegreerd is in de Concentrator Applicatie waardoor eenvoudig gebruik kan worden gemaakt van de beschikbare data en alle data kan worden gebruikt voor het ontwerpen van documenten (folders/webpagina's/brochures/e.d.)

Het is de bedoeling dat de Concentrator uitgebreid wordt met 2 modules: Een design module en een print module. In de design module kan een gebruiker in de webbrowser een ontwerp van de print maken (denk aan een lijst van artikelen in eigen huisstijl, of per artikel een A7 prijskaartje), en dit ontwerp op de server opslaan. In de 2e module (de print module) kan dan met een van deze ontwerpen, samen met een lijst van artikelen, een document gegenereerd worden. Dit document wordt dan bijvoorbeeld in een PDF op de server gemaakt, waarna de gebruiker deze kan downloaden, en printen of op zijn website voor klanten plaatsen.

Omdat nog niet zeker is wat de beste manier is om gebruikers dit te laten doen (het maken van een ontwerp, het opslaan van het ontwerp naar de server, het gebruiken van dit ontwerp op de server om een document te genereren), moet er eerst een onderzoek worden opgesteld. In dit onderzoek moet er worden gekeken naar verschillende frameworks voor het maken van layouts. (Een voorbeeld van zo'n framework is Adobe Rome). Mocht echter geen pakket aan de eisen voldoen, zal er ook aandacht besteed moeten worden aan de haalbaarheid van het zelf implementeren van de ontwerpmodule in bijvoorbeeld een combinatie van Flash/Silverlight, Javascript en CSS. Ook het te exporteren van het document bevat een aantal vraagstukken, vooral welk formaat de klanten het liefst willen ontvangen (een PDF lijkt voor de hand te liggen, maar er zijn genoeg andere bestandsformaten zoals Microsoft Word's doc/docx, ODF (open document format), of zelfs een simpel `ascii3 txt` bestand).

³ ASCII is vertaaltabel tussen getallen en leestekens. In deze context houdt het in dat het om een simpel tekstbestand gaat.

2.2.2 Doelstelling

De doelstelling van dit project is een systeem ontwerpen waarmee gebruikers zelf een ontwerp voor een document kunnen maken, dit kunnen linken met data uit de Concentrator en dit om kunnen zetten in een printbaar document.

Om dit te realiseren moet er eerst onderzoek worden gedaan naar de verschillende technologieën die gebruikt kunnen worden, moet er een onderzoek naar de precieze wensen van de klanten worden uitgevoerd, moet de module gemaakt en geïmplementeerd, en als laatste een publieke test om eventuele fouten eruit te halen.

2.2.3 Opdracht

De opdracht bestaat uit het onderzoeken van de gewenste oplossing en het realiseren hiervan. De Concentrator is op dit moment voornamelijk in gebruik bij klanten in de retail en distributie branche. Onderdeel van het onderzoek zal, in overleg met de klanten, bepalen van de gewenste functionaliteiten zijn en het vast stellen van de gebruikersinterface. Daarnaast moet er gekeken worden welke technologie het beste bij dit probleem past.

Deze opdracht zal worden opgesplitst in twee specifieke onderdelen: De module voor de Concentrator en de Designer, een programma wat in de browser draait en gebruikers in staat stelt de verschillende soorten documenten te ontwerpen.

De module voor de backend zal ontwikkeld worden op ASP.NET, met C# als programmeertaal. Deze taal wordt verder gebruikt in Silverlight en XNA en bied daardoor genoeg mogelijkheden. Als ontwikkelomgeving zal Visual Studio van Microsoft gebruikt worden, met versiebeheer in Git. De designer zal worden gemaakt worden in Adobe Rome, Adobe Flash of Silverlight, wat uitgezocht zal worden in een onderzoek voor het ontwikkelen begint. Communicatie tussen de client applicatie (de designer) en de backend (Concentrator) zal worden verzorgd met asynchrone communicatie over AJAX en vanuit daar zal de module intern met de Concentrator verbinding maken om informatie over producten op te halen. Om specifieke data zoals layouts op te slaan zal een MS-SQL server worden gebruikt. Als projectomgeving zal er een Agile/Extreme Programming methode gebruikt worden, waardoor om de week een voortgangsrapportage en om de maand een voortgangspresentatie/evaluatierapport word gegeven.

3 AANPAK

Dit hoofdstuk beschrijft de projectorganisatie, scope en planning.

3.1 De projectorganisatie

De projectorganisatie bestaat uit een opdrachtgever, een projectleider, een developer en een aantal Concentrator/WPF developers voor advies en hulp.

Opdrachtgever : André van Rijn (Financieel Directeur Direct)

Projectleider : Tim Zeeman (Lead developer Concentrator)

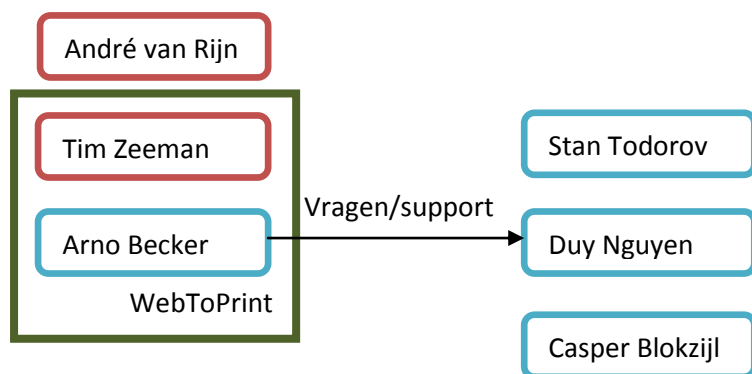
Developer : Arno Becker

External Advisors: Stan Todorov

 Duy Nguyen

 Casper Blokzijl

De projectleider is verantwoordelijk voor het opstellen van de eisen van het project, het definiëren van het globale ontwikkelproces, het sturen van de developer en de globale leiding van het project. De developer voor het ontwerpen, bouwen en implementeren van de developer en de externe adviseurs voor het leveren van informatie over de Concentrator en Windows Presentation Foundation⁴.



Figuur 3: Schema Project Organisatie
projectleider, developer, project

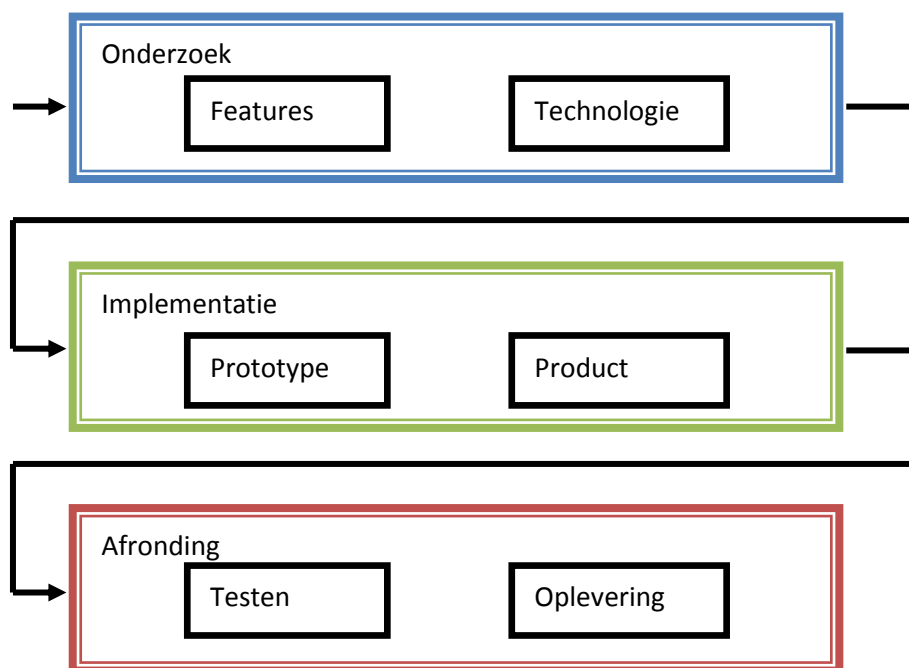
⁴ Ontwikkelomgeving voor het maken van web-based en desktop based applicaties.

3.2 De Projectfasering

Het project is opgedeeld in een aantal belangrijke fasen, en elke fase van het project werd opgedeeld in een aantal sub-fasen.

De hoofdfasen van het project zijn:

- Featureonderzoek
- Technologieonderzoek
- Prototype implementatie fase
- Hoofd implementatie fase
- Testfase
- Opleverfase



Figuur 4: De projectfasering

Elke fase bestaat uit een aantal subfasen, om het project simpeler te laten beheren en duidelijke doelen te stellen:

Startup:

In de startup werd gekeken naar de huidige staat van het project, de doelen voor de fase en het stellen van deadlines, milestones en discussiemomenten. De startup was kort, meestal één of twee dagen, en resulteerde in een to-do lijst voor de uitvoerende fase, een tijdsindeling voor de verschillende onderdelen en een bespreking met de projectleider over de voortgang van het project.

Uitvoerfase:

Tijdens de uitvoerfase werd de to-do lijst stapsgewijs afgewerkt. Om de paar dagen was er een informele en korte meeting met de projectleider over de voltooide gedeeltes van de to-do lijst, de aanpak van de volgende punten en of er misschien wijzigingen in de To-Do lijst aangebracht moesten worden. Door de werkdag door werden de gedane taken beschreven in het tijd managementsysteem zodat er later precies kon worden gekeken hoeveel tijd er aan welke taken werd besteed. Deze fase duurde gemiddeld 3 tot 4 weken, waarbij de hoofd implementatiefase eigenlijk de gehele stage doorging.

Reflectiefase:

In de reflectiefase werden de resultaten van de gedane taken opgesomd, een balans gemaakt van wat goed ging en wat niet en werd er gekeken naar punten die bij de to-do lijst van de volgende fase konden worden bijgeschreven.

Documentatiefase:

Tijdens de documentatiefase werden de notities van de uitvoerfase herschreven in een nieuw document. Wanneer er in de uitvoerfase is geprogrammeerd, werd de code voorzien van commentaar en eventueel geherstructureerd voor gemakkelijker onderhoud. Bij de overige fasen werden de resultaten in de scriptie verwerkt.

3.2.1 Featureonderzoek

Het featureonderzoek begon met het uitzoeken wat de wensen van de klanten waren. Als start werd er een brainstorm sessie georganiseerd met de ontwikkelaars van de Concentrator en de originele WebToPrint module. De ideeën uit deze brainstormsessie werden op papier gezet, en verwerkt tot een vragenlijst. Deze vragenlijst werd opgedeeld in twee gedeeltes: Een gedeelte met de features uit de brainstormsessie, met de vraag om punten aan features toe te kennen, die het meeste in overeenstemming kwamen met de wensen van de klant en een open gedeelte waarin klanten zelf features konden beschrijven. Deze vragenlijst was daarna verstuurd naar zowel de klanten die de extra features in het WebToPrint project hadden aangevraagd en bestaande klanten van de Concentrator werden geïnformeerd over dit project, om de interesse voor het project te pollen, en te kijken wat de wensen waren.

Uit de resultaten van deze vragenlijst is er daarna een MoSCoW document opgesteld, een document waarin alle requirements worden opgedeeld in 4 prioriteitsgroepen, de Must Haves, de Should Haves, de Could Haves en de Wants. Het project werd georganiseerd in een op Agile⁵ geïnspireerde omgeving, dus werden er features geïmplementeerd in de volgorde van deze MoSCoW lijst. Een ander voordeel van deze projectmethode was dat nieuwe features simpel in de MoSCoW lijst bijgevoegd konden worden zodat deze dan met de volgende iteratie meegenomen konden worden zonder extra planning.

3.2.2 Technologieonderzoek

Na het uitzoeken van een lijst met features moest worden uitgezocht welke technologieën er in het project gebruikt gingen worden. Bij Diract wordt er voornamelijk in Microsoft talen software ontwikkeld, maar omdat dit een vrije complexe applicatie moest worden die in de browser zou draaien, en daar bij Diract geen standaardpakket voor is, werd er besloten dat er een onderzoek naar de verschillende beschikbare technologieën uitgevoerd moest worden. Er moest worden beslist in welke programmeertaal/omgeving er ontwikkeld zou worden, hoe de applicatie zou worden vormgegeven en hoe de Concentrator het beste met de applicatie geïntegreerd kon worden. Het onderzoeken zelf gebeurde met het zoeken naar informatie over verschillende technologieën en daarvan een lijst te maken met de voor- en nadelen en aan welke voorwaarden ermee voldaan werden.

3.2.3 Prototype Implementatie fase

Om ervoor te zorgen dat de technologie gekozen in het vorige onderzoek voldeed aan de eisen van het project werd besloten om eerst een prototype te maken. Belangrijke onderdelen van zowel de Designer als de Concentrator module werden (gedeeltelijk) uitgewerkt om de performance en gemak van implementatie uit te testen. Het resultaat van de prototype-fase kon worden gebruikt om klanten een eerste indruk van de Designer te kunnen geven en om eventueel commentaar later in de hoofd implementatie fase mee te kunnen nemen. Tevens werd de opgedane kennis gebruikt om het uiteindelijk Plan van Aanpak te schrijven.

⁵ Agile, een projectorganisatiemethode waarin het project iteratief wordt ontwikkeld. Dit houdt in dat het project eerst in grote lijnen wordt opgezet, en daarna steeds dieper wordt uitgewerkt. Dit brengt als voordeel dat het project in een veel eerder stadium bekeken kan worden door klanten, en deze vroeg in het ontwikkelproces het project kunnen bijsturen.

3.2.4 Hoofd Implementatie fase

Nadat het prototype was afgerond werd er gekeken naar de verbeterpunten van het prototype. Op basis van deze verbeterpunten werd het originele MoSCoW document aangepast.

De uitvoerfase werd opgedeeld in blokken van 2 weken, waarin om de paar dagen een bespreking met de projectleider werd georganiseerd over de voortgang van het project, eventuele nieuwe ideeën van klanten en voor hulp met het ontwerpen van de applicatie.

3.2.5 Testfase

Ongeveer op driekwart van het ontwikkelproces van de applicatie, wanneer alle Must Haves en de meeste Could Haves waren afgerond, zou de applicatie als beta versie live gaan, om directe input van gebruikers te kunnen krijgen, en daarmee in de resterende ontwikkeltijd de applicatie zoveel mogelijk af kunnen stemmen op de wensen van deze gebruikers.

Het was belangrijk dat de applicatie intuïtief, toegankelijk en gemakkelijk te gebruiken voor de (niet technische) eindgebruikers was, dus moest er relatief veel zorg aan de indeling van de gebruikersinterface en workflow besteed worden.

3.2.6 Opleverfase

In de opleverfase werd er gewerkt aan de finishing touches van de applicatie. De feedback van de testfase van het programma werd gebruikt om de bruikbaarheid van het programma te verbeteren en eventuele laatste bugs op te lossen. Naast het afronden van de applicatie moest er documentatie voor de eindgebruikers geschreven worden.

De laatste stap in het opleveren van de applicatie was een lokale uitrol bij de BAS groep. Deze lokale versie draait bij Direct. Vanwege de nabijheid van de BAS gebruikers (hetzelfde gebouw als Direct) is dit een goede laatste test voordat de WebToPrint applicatie wordt uitgerold bij andere klanten.

4 UITWERKING

Dit hoofdstuk beschrijft de verschillende fases van de uitwerking van de opdracht

4.1 Featureonderzoek

Voor de featurelijst werden er e-mails gestuurd naar 3 bestaande klanten van de Concentrator, zowel naar 2 nieuwe klanten die interesse getoond hadden voor de WebToPrint functionaliteit. Deze vragenlijst bestond uit de lijst van features die tijdens een brainstormsessie waren bedacht, en een open gedeelte waar klanten zelf features in konden vullen. Het resultaat van de response op deze vragenlijst is de volgende MoSCoW lijst:

4.1.1 *Must Haves*

- De applicatie moet niet gebonden zijn aan bepaalde velden in de database: Data ophalen moet dynamisch via voorgedefinieerde parameters gebeuren.
- Een gebruiker moet zonder enige kennis van designsoftware elementen kunnen aanmaken, verplaatsen, kopiëren en plakken, vergroten/verkleinen, en verwijderen
- Een gebruiker moet voor het aanmaken van meerdere elementen een 'composite' aan kunnen maken: Een element dat bestaat uit meerdere andere elementen met door de gebruiker gedefinieerde stijlen en plaatsingen. Een voorbeeld van een composite is een prijskaartje: Eenmaal aanmaken, en dan simpel invoegen voor het aantal prijskaartjes dat de gebruiker nodig heeft.
- Gebruiker moet projecten kunnen aanmaken en inladen: Binnen een project bestaan dan documenten en voorgedefinieerde pagina's voor het snel aanmaken van documenten en het delen van stijlen tussen documenten voor een uniforme opmaak.
- Documenten moeten kunnen verwerkt worden tot een printbaar document formaat (PDF, DOC) op de server.

4.1.2 *Should Haves*

- Een gebruiker moet in de applicatie zijn document als een voorbeeld kunnen bekijken met echte data uit de Concentrator.
- Gebruikers moeten in kunnen loggen met hun Concentrator gebruikersnaam/wachtwoord en alleen projecten en documenten van die gebruiker te zien krijgen.
- Een gebruiker moet kunnen zoeken in de Concentrator en daarmee gemakkelijk producten vinden en in de Designer in kunnen voeren.
- Een administrator moet documenten, componenten en pagina's in de Management App van de Concentrator kunnen verwijderen.
- Een pagina moet een layout toegewezen kunnen krijgen, om componenten automatisch te kunnen rangschikken.

4.1.3 Could Haves

- Een gebruiker moet de resultaten van de Concentrator in tabulaire vorm (Comma Seperated Values, Excel) kunnen exporteren.
- Een gebruiker moet lijsten met producten als component in de Designer in kunnen voeren
- Een gebruiker moet een pagina direct vanuit de Designer als plaatje (JPG, PNG) kunnen exporteren
- Een gebruiker moet ook basisvormen (vierkanten, cirkels) aan pagina's toe kunnen voegen.

4.1.4 Wants

- Een gebruiker moet ook niet-Concentrator data in de Designer kunnen importeren/invullen voor gebruik in een document.

4.2 Technologieonderzoek

4.2.1 Designer

Het eerste technologieonderzoek begon met het zoeken naar een ontwikkelomgeving die voldeed aan de eisen voor de designer: Het moest een applicatie worden waarin gebruikers tot op de millimeter nauwkeurig elementen moeten kunnen plaatsen, het moet een directe verbinding met de Concentrator service kunnen draaien en het moet binnen een browser kunnen draaien. Het onderzoek werd gedaan door informatie over verschillende technieken te verzamelen en deze informatie samen te vatten in een lijst, waardoor er door het gemaakte overzicht gemakkelijk een keuze gemaakt kon worden.

Adobe Project Rome

- + Gemakkelijke designer voor het maken van designs.
- + Korte development: Het grootste gedeelte van het project (de designer) is al af.
- Geen directe import mogelijk: Er zou dan een tussenapplicatie moeten worden ontwikkeld om de templates van Rome met de data uit de Concentrator samen te voegen.
- Geen support meer vanuit Adobe.

Adobe Flash

- + Goede ondersteuning onder gebruikers, (bijna) iedereen heeft een flash plugin.
- + Print support direct uit de browser.
- Geen licentie meer voor ontwikkel omgeving, zou apart voor dit project aangeschaft moeten worden.
- Geen goede kennis van AS3⁶ bij Direct.l
- Designer moet helemaal vanaf de grond af aan opgebouwd worden.

Microsoft Silverlight

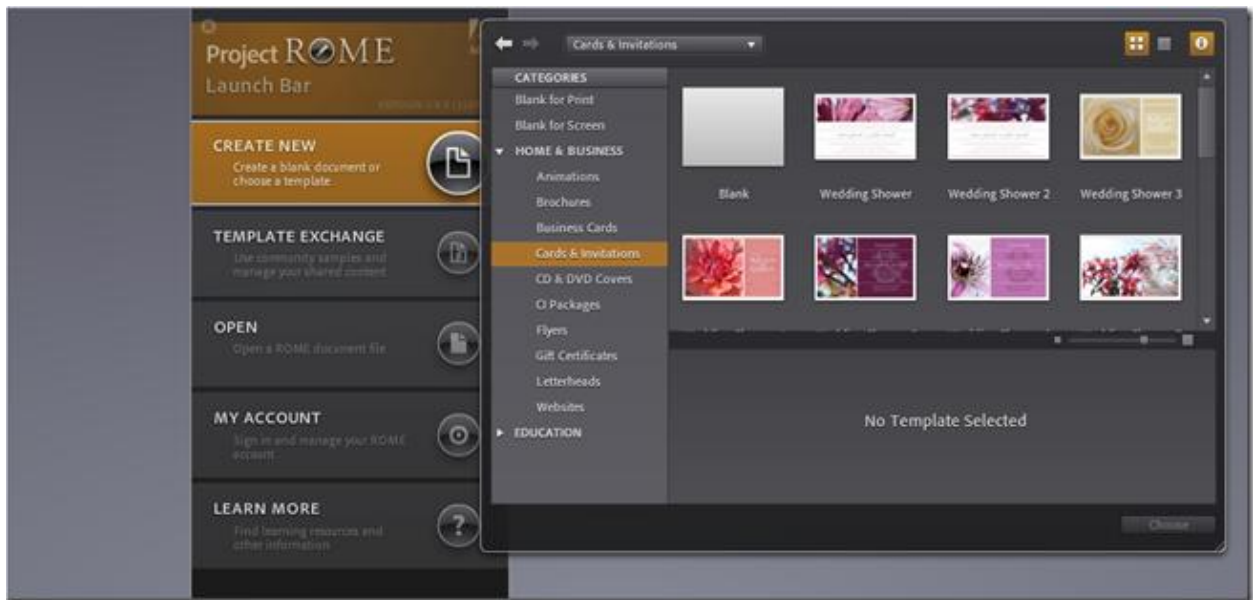
- + Werkt direct uit de browser.
- + Kan direct met de Communicator via SOAP werken.
- + Goede ontwikkel omgeving.
- Werkt alleen op Windows.

Microsoft Windows Presentation Foundation

- + Kan alles wat Silverlight kan, en meer.
- Werkt officieel alleen met Internet Explorer onder Windows 7, met Firefox en Internet Explorer (en niet officieel met Chrome) onder Windows XP.

⁶ Action Script 3, de programmeertaal in Flash

Op basis van deze mogelijkheden hebben we als technologie voor Web-Based WPF gekozen. Adobe Rome (zie figuur 5) is helaas uit gefaseerd door Adobe en krijgt geen ondersteuning meer. Adobe Flash zou een prima alternatief zijn, maar de programmeerkennis voor Flash producten bij Diract is gelimiteerd, en er zou dan een aparte licentie voor de Flash ontwikkelstudio moeten worden gekocht. Een ander minpunt van Flash is dat de ontwikkelomgeving is gespecialiseerd voor ontwerpers en dus minder geschikt om grote complexe applicaties in te schrijven.



Figuur 5: Adobe Rome

De keuze tussen Silverlight en WPF is wat gecompliceerder. WPF is een zogeheten superset van Silverlight, dat betekent dat alle functionaliteit in Silverlight ook in WPF zit, maar de mogelijkheden in WPF zijn uitgebreider. Het nadeel van WPF is wel dat het alleen op Windows platformen draait (Silverlight heeft een open-source Linux poort genaamd Mono, en draait ook op Linux en Apple systemen), en het eigenlijk een desktop applicatie is die binnen in de browser draait.

Omdat WPF meer mogelijkheden heeft om direct met andere servers te praten zonder een interface-laag te schrijven, de applicatie niet op andere besturingssystemen dan Windows hoeft te draaien en in het algemeen sneller is dan Silverlight is dan ook voor deze optie gekozen om de Designer in te ontwikkelen.

4.2.2 Printing

Omdat het doel van het project was om data vanuit de Concentrator en ontwerpen vanuit de Designer samen te voegen tot documenten, moest er eerst onderzocht worden naar wat voor documenten we dit gingen exporteren en hoe we die export zouden implementeren.

4.2.2.1 Bestandsformaat

Om te uit te zoeken welk bestandsformaat het beste bij de WebToPrint functionaliteit zou passen, is er een lijst opgesteld met verschillende documentformaten en bijbehorende plus- en minpunten.

Microsoft Word (.DOCX/.DOC)

- + De-facto standaard voor documenten binnen bedrijven
- + Leesbaar met gratis Reader van Microsoft
- + Kan gemakkelijk veranderd worden na export
- + Heeft simpele bestandsstructuur
- Layout problemen als het geopend word met een ander programma dan Microsoft Word

Adobe Portable Document Format (.PDF)

- + Goede ondersteuning onder gebruikers, (bijna) iedereen heeft een PDF reader
- + (gratis en betaalde) exporteerm modules voor ASP.NET zijn al ontwikkeld
- Grotere bestanden
- Niet veranderbaar door eindgebruikers

Open XML Paper Specification (.XPS)

- + Gratis lezer beschikbaar van Microsoft
- + Kleiner dan PDF
- Vergeten formaat, was bedoeld als antwoord op PDF maar is nooit populair geworden

Open Document Format (.ODF)

- + Openbaar in Word, OpenOffice en tal van andere opensource pakketten
- + Simpel bestandsformaat
- Geen grote gebruikersgroep

Omdat de bestanden van de WebToPrint functie waarschijnlijk later ook in nieuwsbrieven zal worden gebruikt, en met het oog op de grote hoeveelheid eindgebruikers die een PDF reader geïnstalleerd hebben, is de keuze al vrij snel op PDF gevallen. Het kan zijn dat er in een later stadium ook een exporter voor Microsoft Word bestanden word geschreven, maar dat zal in de MoSCoW “Could Have” lijst vallen.

4.2.2.2 Exporteer module

Nadat besloten was welk bestandsformaat er gebruikt ging worden moest er een besluit worden genomen over de module die het exporteren zou gaan doen. Deze module zou dan worden aangeroepen in de Concentrator, en stapsgewijs het document opbouwen met de data uit de Concentrator en het ontwerp uit de designer.

Er waren twee opties: Een module gemaakt door een derde partij, of een module die in-house ontwikkeld zou moeten worden.

Zelf een PDF module implementeren

- + Geen licentiekosten
- + Volledig aanpasbaar aan de behoeften van de applicatie
- Kost veel tijd om te implementeren
- Kan mogelijk foutieve PDF documenten genereren

iText library

- + Volledig geteste en werkende module
- Heeft licentie nodig

Report.Net library

- + Gratis
- + Simpele code
- GPL licentie
- Gebruikt eigen klassen voor waardetypes: Alle data moet eerst geconverteerd worden voordat het gebruikt kan worden

PDF Sharp

- + Gratis
- + MIT Licentie
- + Simpele code
- Implementeerd niet alle PDF functies

Sharp PDF

- + Gratis
- + Simpele code
- GPL licentie

De keuze om een module van een 3^e partij te kiezen was simpel: zelf een exporteer module schrijven zou te veel tijd kosten, zowel om te ontwikkelen als om te testen. Die optie viel daarom al snel af. De Sharp PDF en Report.NET modules vielen ook snel af, omdat deze alleen gratis te gebruiken waren onder een GPL licentie. Dat zou betekenen dat de WebToPrint ook onder de GPL licentie zou vallen, en daarmee de complete Concentrator. Dit zou betekenen dat de Concentrator installaties bij klanten geen WebToPrint module zou kunnen bevatten, terwijl WebToPrint juist voor dit doel zal worden ontwikkeld.

De iText module heeft een licentie nodig om voor commerciële producten te gebruiken, en na wat testprojecten met zowel de PDF Sharp en iText module was beslist dat de iText module het extra geld niet waard was. De PDF Sharp module was op het gebied van functionaliteit praktisch gelijk aan de iText module, en daarom is er uiteindelijk gekozen voor de PDF Sharp module.

4.2.3 Dataformaten Designer

Het laatste technisch onderzoek ging over het zoeken van een oplossing voor het versturen en opslaan van de ontwerpen van de Designer. De ontwerpen zelf moeten in de Designer in een aantal data-klassen worden opgeslagen, maar moeten worden omgezet naar een tekstformaat voor het versturen over het internet.

Voor het vertalen van objecten naar tekst zijn een aantal mogelijkheden, en daarom is er een klein onderzoek opgesteld om uit te zoeken welke methode het beste is.

eXtensible Markup Language (XML)

- + Ingebouwde functies in .Net
- + Gemakkelijk te implementeren
- Redelijke overhead
- Vertaling moet zelf geschreven worden

Javascript Object Notation (JSON)

- + Weinig overhead
- + Snel
- Parser niet beschikbaar in WPF

Interne .Net Serialization (ISerializable)

- + Ingebouwd in .net
- + Zeer snel
- + Geen overhead
- Niet plaintext (kan niet door mensen gelezen worden)

De interne .Net Serialization viel direct af, omdat het niet leesbaar is (en het is dus niet gemakkelijk uitzoekbaar als er iets fout gaat tijdens het wegschrijven/inladen).

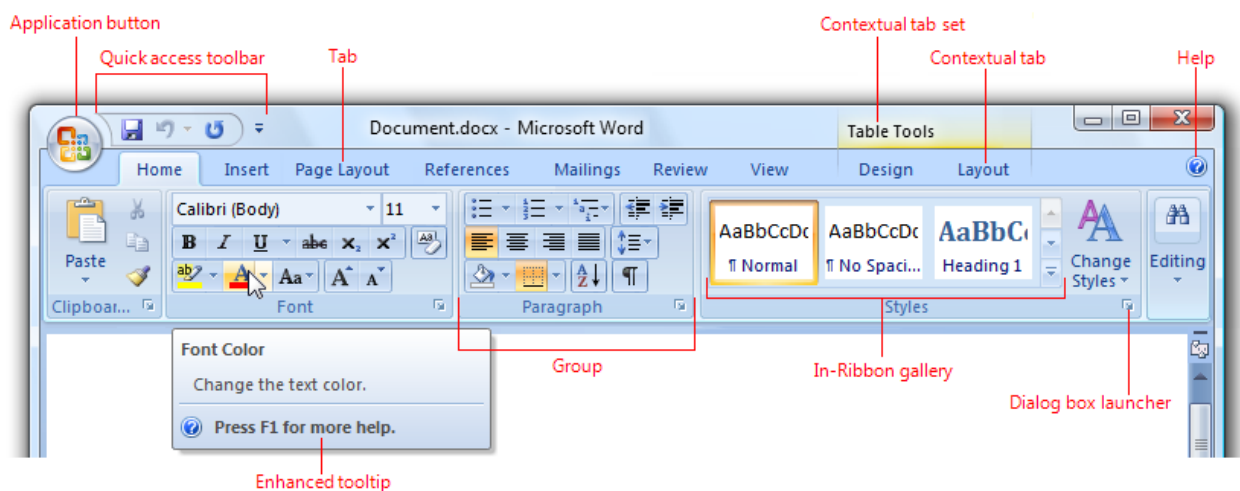
Daarna is er voor XML gekozen omdat de Concentrator service via SOAP⁷ werkt en het een beter idee leek om alles via hetzelfde protocol te versturen.

⁷ Simple Object Reference Protocol, een manier om data via XML te versturen

4.3 Prototype Implementatiefase

De prototype-implementatiefase was vooral om kennis te maken met WPF en de Concentrator, kijken hoe geschikt WPF was voor het ontwikkelen van de Designer en hoeveel tijd er ongeveer in het project zou gaan zitten.

Het eerste punt waar aandacht aan werd besteed was de User Interface. Omdat de applicatie vooral gericht ging worden op gebruikers die ervaring hebben met Microsoft Office was een van de eerste ideeën om een ribbon-achtige⁸ interface te gebruiken. Omdat het gebruik van een Ribbon in applicaties door Microsoft aan banden is gelegd, waren er 3 opties: De officiële Ribbon implementatie van Microsoft gebruiken, Zelf een Ribbon implementatie schrijven of een 3^e partij Ribbon implementatie gebruiken. Aan de laatste 2 keuzes waren licentiekosten verbonden voor het gebruik van de Ribbon, dus werd er gekozen voor de Microsoft Fluent Ribbon.



Figuur 6: Een Ribbon Interface

De ribbon zou de gebruiker voorzien van een herkenbare en duidelijke user interface zonder gecompliceerde menustructuren of oneindige hoeveelheden knoppen en tabbladen. De reactie van klanten was positief genoeg om te besluiten dit ook in de uiteindelijke versie van het programma te gebruiken.

De 2^e grote feature van het Designer pakket zijn de componenten. Componenten houden de toevoegbare, verplaatsbare en verwijderbare objecten op een pagina, zoals plaatjes, tekstvakken en titels. Om de ontwikkeltijd kort te houden is voor het slepen en schalen van deze componenten een opensource stuk code (met een vrije licentie) gebruikt.

Een ander belangrijk onderdeel was het 'model' gedeelte van de designer. Vroeg in de ontwikkeling is het besluit genomen om een basis 'Component' klasse te schrijven en elk ander object hiervan te laten overerven. Dit bracht als voordeel dat elk component in een ander component als kind gestopt kon worden, en er dus gemakkelijk een hiërarchie van componenten gebouwd kon worden.

⁸ Ribbon of Fluent Interface is de hoge balk bovenin elk Microsoft Office programma in Office 2007 en Office 2010, alswel als de Paint en Wordpad programma's in Microsoft Windows Vista en Windows.

Als laatste onderdeel van de Designer is het 'binding' gedeelte geïmplementeerd. Dit binding gedeelte geeft gebruikers de mogelijkheid om velden uit de database met velden in het ontwerp te koppelen. Het enige wat een gebruiker hoeft te doen na het opzetten van een binding is een product nummer invullen, en daarna worden de alle plaatjes en tekstvelden automatisch ingevuld met data uit de Concentrator.

Tim Zeeman en een aantal klanten waren genoeg onder de indruk van het prototype dat daarna de go-ahead werd gegeven voor het verder ontwikkelen van de WebToPrint in de hoofd implementatiefase.

4.4 Hoofd Implementatiefase

Nadat het prototype goedkeuring had gekregen werd er besloten om door te gaan met het uitwerken van de uiteindelijke versie van de Designer. Dit programma zou van de grond af aan opnieuw gebouwd worden om een aantal redenen:

- Het prototype was een snelle implementatie, er zat veel slordige code in.
- Tijdens het bouwen is er veel geleerd over WPF, en dingen hergebruiken zou ervoor zorgen dat veel foute stukken code overgenomen zouden worden.
- Dit zou een kans geven om een duidelijke structuur in het programma te brengen, en de applicatie zal daardoor beter onderhoudbaar zijn in de toekomst.

Tijdens de hoofd implementatiefase is de Concentrator Module tegelijk met de designer ontwikkeld, maar om dit duidelijk in dit verslag te verwerken zijn de implementaties opgedeeld in twee aparte deelhoofdstukken.

4.4.1 Uitwerking Concentrator Module

De uitwerking van de Concentrator Module begon met het ontwerpen van de database. Deze database zou de verschillende projecten, documenten, samengestelde objecten, stijlen en paginaontwerpen bevatten. Om het ophalen van data te versimpelen werd besloten om de XML data vanuit de Designer direct in een veld in de database te plaatsen, in plaats van de XML data uit de Designer te verwerken en alle losse componenten in aparte tabellen op te slaan.

Het volgende punt van de Concentrator Module was het implementeren van een manier om de binnenkomende data vanuit de Designer om te zetten in een PDF bestand. De PDF Sharp bibliotheek heeft een duidelijke documentatie en een simpele manier van programmeren, dus het duurde niet lang voordat er een volledige PDF exporteer functie was geïmplementeerd. Het enige probleem met deze functie is dat het genereren van deze documenten veel tijd kostte, soms wel een kwartier om een simpel document te verwerken. Om dat probleem te verhelpen is daarom besloten om een print queue te bouwen: Elke keer wanneer een gebruiker op de “Genereer PDF” knop in de Designer drukte, werd het document naar de Concentrator module gestuurd, die deze dan in de printqueue zette. De printqueue werd constant bekeken door een aparte service binnen de Concentrator, die nieuwe documenten begon te verwerken. De gebruiker kon dan in de Designer kijken hoeveel documenten er nog in de printqueue staan, en kon het document downloaden wanneer het klaar was. Dit zorgde ervoor dat er minder piekbelasting op de Concentrator was, en de gebruiker duidelijk zicht had waarom het genereren van een document lang duurde.

De rest van de Concentrator module bestond uit de webservice functies die de Designer in staat stelde om data uit de database aan te vragen, dit was relatief simpel te implementeren.

4.4.2 Uitwerking Designer

Voor het bouwen van de Designer moest er eerst een plan gemaakt worden over de manier waarop het programma gestructureerd kon worden. Er waren een aantal mogelijkheden:

- MVVM
Model View ViewModel, Dit is de geprefereerde manier om WPF applicaties te maken. De Model (de data) is verbonden met de View (wat de gebruiker ziet) via een ViewModel, die alle veranderingen opslaat en op alle data business rules toepast.
- MVC
Model View Controller. De gebruiker verandert dingen in de View, die daarna door een Controller wordt bijgehouden en dingen veranderd in de Model.

In eerste instantie was het de bedoeling om een MVVM applicatie te maken (bijlage 1 bevat het schema voor de applicatie met het MVVM patroon) , maar omdat in de Designer applicatie de View praktisch gelijk is aan de Model (een gebruiker sleept componenten heen en weer, en de positie, grootte en type van de componenten staat direct in het Model), is later besloten om hier toch een MVC applicatie van te maken (en dit patroon is beschreven in bijlage 2).

Het bouwen van de Designer begon met het herimplementeren van de User Interface. De Ribbon interface die uit het technisch onderzoek kwam voldeed prima als interface, maar de implementatie gaf een aantal onvoorziene problemen.

Een van de eerste problemen met het implementeren van deze Ribbon was het feit dat het pakket van Microsoft nog niet af was. Een van de features die vrij noodzakelijk voor het goed kunnen gebruiken van de Designer, de zogeheten in-ribbon gallery⁹, was niet aanwezig in de ribbon. Andere problemen met de Ribbon waren stijlfouten, defecte onderdelen en in sommige gevallen crashes van de Ribbon als iets niet op een bepaalde manier werd gebruikt. Al deze zaken repareren kostte aanzienlijk meer tijd dan origineel was ingepland voor het bouwen van de GUI, en nader onderzoek op de discussiefora van de Ribbon gaven aan dat Microsoft de Ribbon nooit echt heeft afgemaakt, en simpelweg bij versie 0.9 alle ontwikkelingen heeft stopgezet.

Nadat de interface in grote lijnen was opgezet, moest de zogeheten 'model' gebouwd worden. Dit model bevat alle data voor de applicatie, in het geval van de Designer de positie van alle elementen, de structuur tussen objecten, de maten van de pagina's en binding voor alle componenten.

Door complexere componenten, zoals pagina's en composiet componenten te laten overerven van normale componenten was het mogelijk om een hiërarchische structuur te vormen. Dit houdt in dat een component als kind aan elke andere component toe gevoegd kon worden. De gebruiker kan daardoor gemakkelijk groepen van componenten knippen, plakken en verplaatsen.

Tijdens het ontwikkelen werd er continue getest of het programma correct werkt, of het niet vast kon lopen en of alle fouten binnen het programma goed werden afgehandeld.

⁹ In-Ribbon Gallery is een lijst met dingen (font opmaak, maar kan bijvoorbeeld ook plaatjes) waarvan in de Ribbon voorbeelden worden gegeven.

4.5 Testfase

Nadat de meeste features van de Must en Should Have van de MoSCoW featurelijst waren geïmplementeerd, was het moment om een kleinschalige testfase uit te voeren. Dit werd gedaan door de WebToPrint bij de lokale Concentrator installatie te voegen, en deze door een aantal eindgebruikers te laten testen. Dit gaf de gebruikers een kans om met het programma te werken en eventueel commentaar te geven wat nog mee genomen kon worden in de laatste ronde development en gaf bij Direct de mogelijkheid om een goede documentatie voor het pakket te maken en om bugs in het programma op te lossen voordat het programma compleet uitgerold zou worden.

Uit deze testronden kwamen er een aantal belangrijke dingen naar boven, zoals kleine aanpassingen in de Ribbon om de workflow van het programma te verbeteren samen met een aantal kleine schoonheidsfoutjes in het programma.

4.6 Opleverfase

De laatste stap in het ontwikkeltraject van de WebToPrint module was het uitrollen van de applicatie naar de Concentrator installaties van de verschillende klanten. Deze uitrol betekende niet dat de ontwikkeling van de applicatie werd stopgezet, maar dat nu de meeste klanten met de applicatie konden werken die aangesloten waren op de Concentrator bij de BAS groep. Als laatste zouden de klanten aangesloten worden die een lokale Concentrator draaien, vooral omdat het bijwerken van deze installaties lastiger is en omdat sommige klanten bepaalde modules die de WebToPrint applicatie nodig heeft (zoals de Icecat Image download) niet hebben.

5 CONCLUSIES

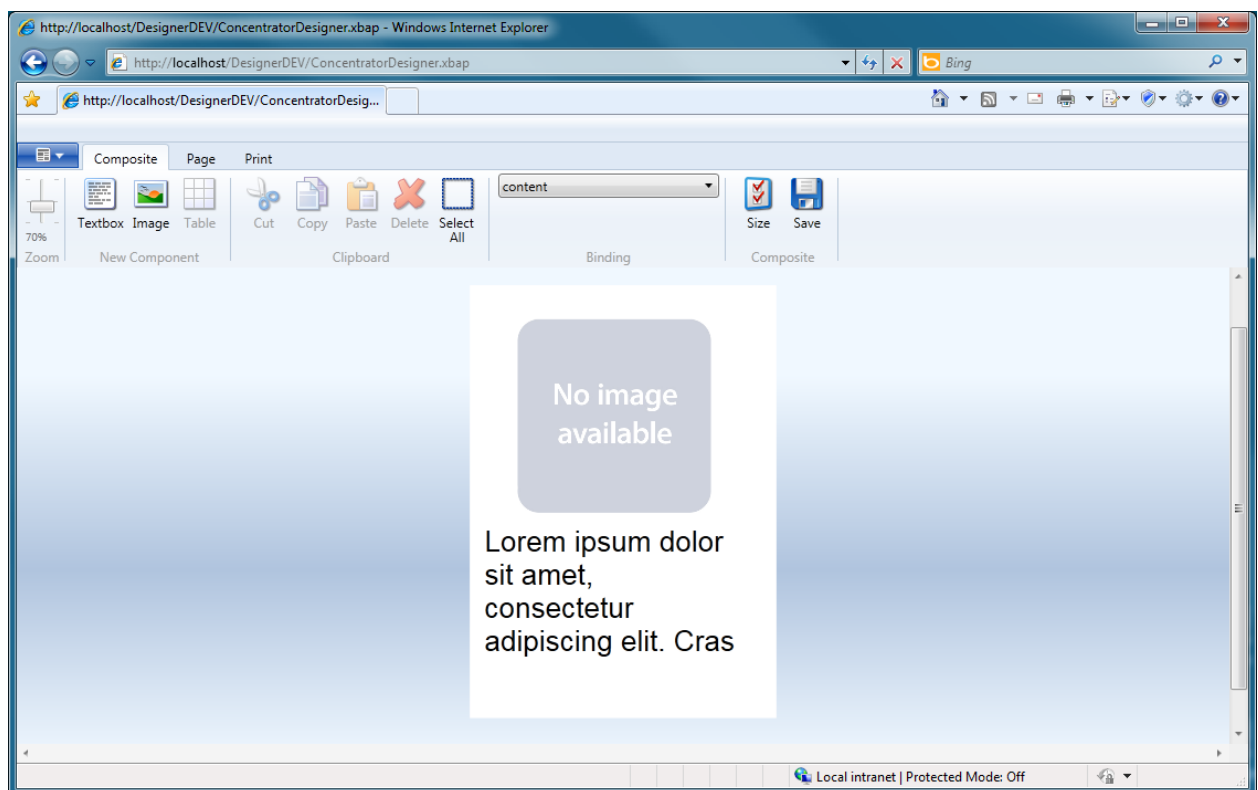
In dit hoofdstuk worden de verschillende opgeleverde producten besproken, en de evaluatie van het hele project.

5.1 Opgeleverde producten

5.1.1 De Designer

De WebToPrint Designer, het hoofdgedeelte van het product. Geschreven in C# met de Windows Presentation Foundation. Dit programma draait met de WPF plugin in de webbrowser en stelt een gebruiker in staat om document ontwerpen aan te maken, te verbinden met data uit de Concentrator, en uiteindelijk de opdracht aan de Concentrator Module te geven om er een print van te maken.

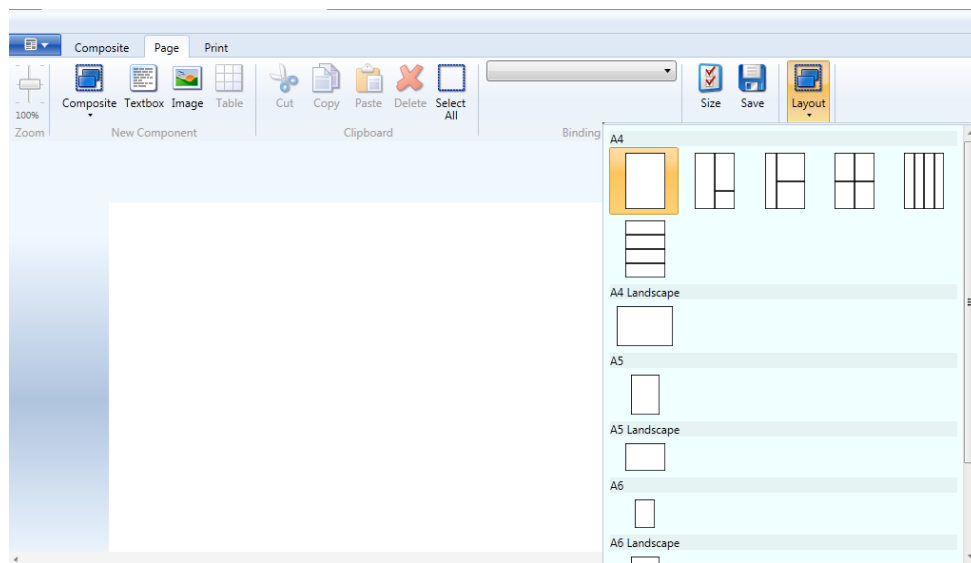
Om goed te kunnen begrijpen wat de Print Designer precies is, word hier een stapsgewijze uitleg over het maken van een document gegeven.



Figuur 7: Het maken van een Composiet component

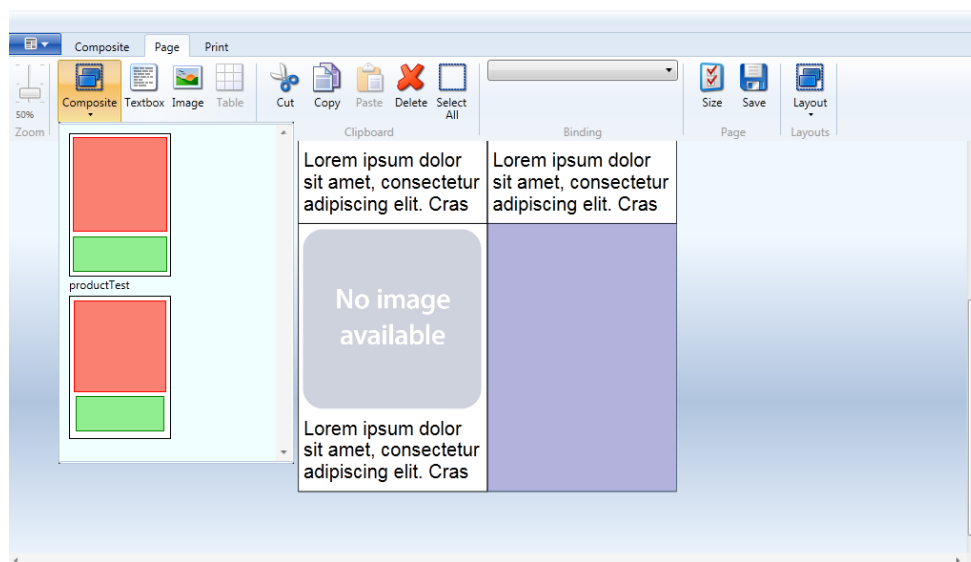
Om een document te maken moeten er eerst zogeheten composite elementen ontworpen worden. Dit gebeurt door het invoegen van plaatjes en tekstvelden. Na het toevoegen moeten deze onderdelen verbonden worden met de database, en dat gebeurt door in de Binding dropbox een datasource te selecteren. In het voorbeeld hierboven is dat de “content” datasource. Deze datasources worden gedefinieerd in de Concentrator Management App module (daarover later meer).

De volgende stap is het ontwerpen van een pagina. Deze pagina wordt later gebruikt om het document te vullen. Als eerste moet er een layout aan een pagina gekoppeld worden. Deze layout zorgt ervoor dat de alle componenten op de pagina goed ingedeeld en uitgelijnd zijn.



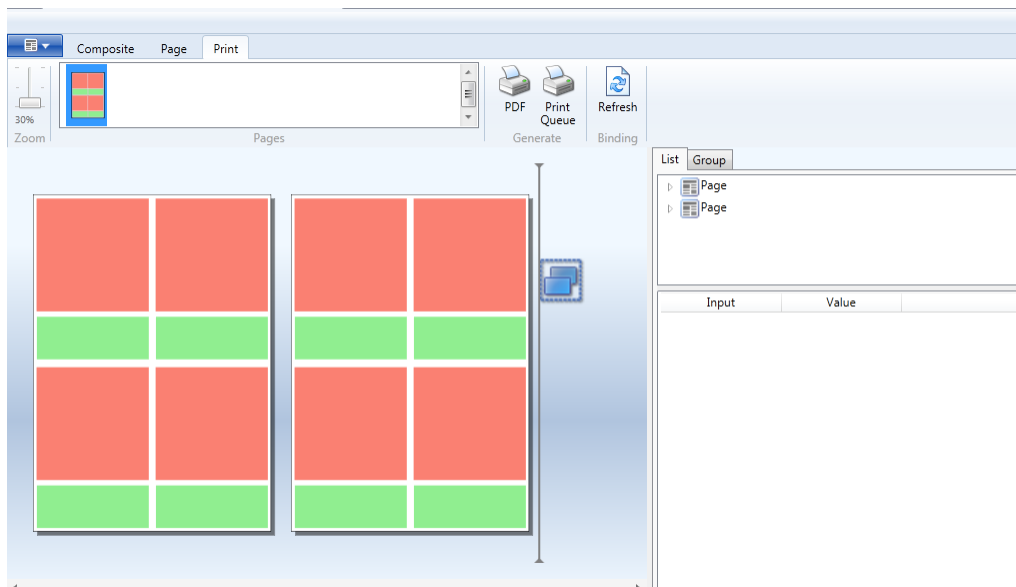
Figuur 8: Het kiezen van een layout

Nadat er een layout aan de pagina is toegevoegd, kan het net ontworpen composiet component aan de pagina toe gevoegd worden. Dit gebeurt door een 'veld' in het ontwerp te selecteren en daarna een van de opgeslagen composiete componenten uit de lijst te selecteren. Deze wordt daarna aan de pagina toegevoegd.



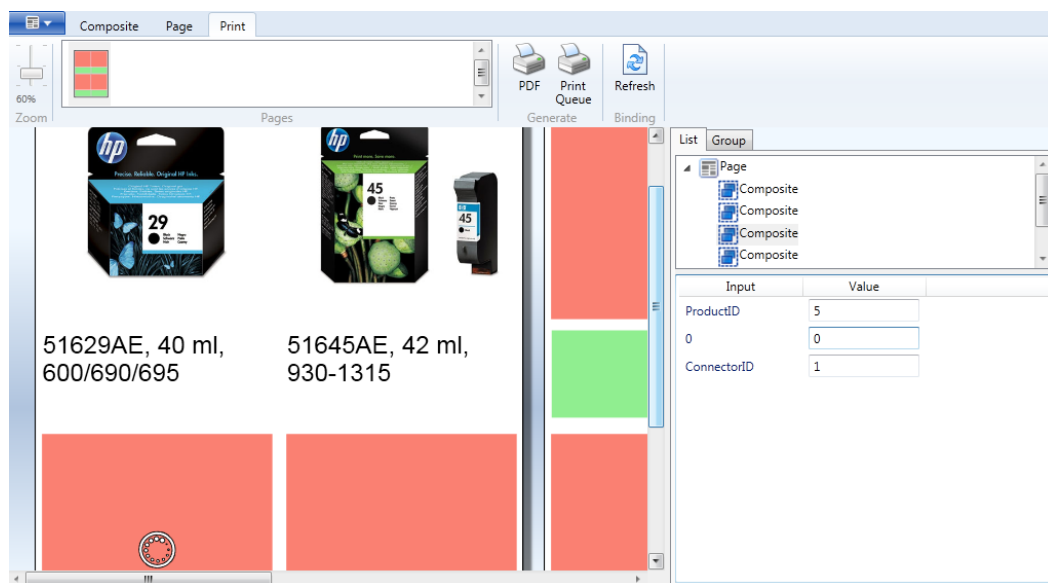
Figuur 9: Het plaatsen van een composiet component in een layout

Nadat deze pagina, samen met de layout en de ingevoegde componenten is opgeslagen, kan de gebruiker verder naar het Print tabblad. In deze mode kunnen de verschillende layouts in volgorde toegevoegd worden.



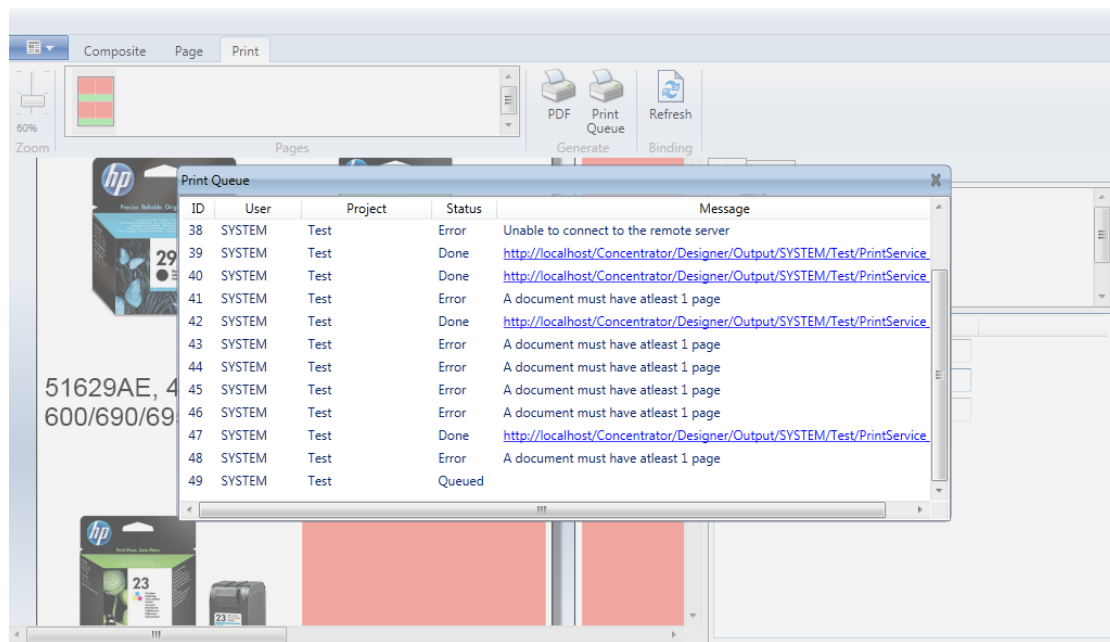
Figuur 10: Het toevoegen van pagina's in het document

Wanneer de benodigde pagina's zijn toegevoegd, hoeft de gebruiker alleen nog maar de productnummers in het scherm rechts in te voeren. Wanneer alle waarden zijn ingevuld, haalt de Designer de informatie uit de Concentrator, en laat deze als voorbeeld in de Print modus zien.



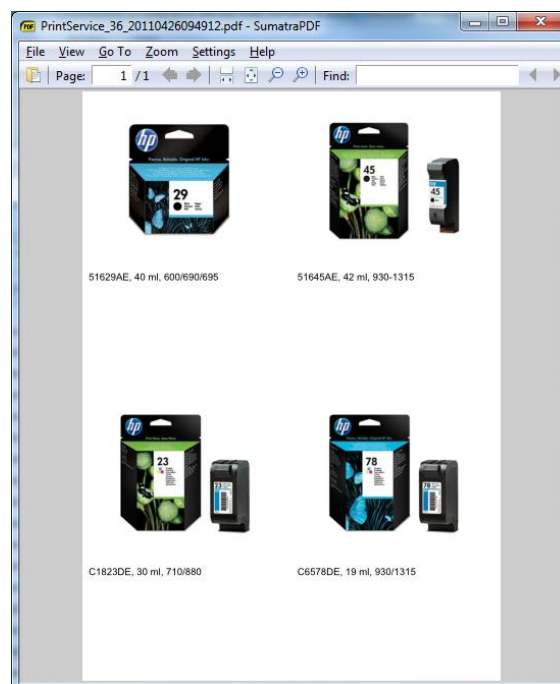
Figuur 11: De Designer laat de voorbeelden voor 2 producten zien, terwijl een 3^e word opgehaald.

De laatste stap in het maken van een document is de PDF output. Als een gebruiker op de PDF knop drukt word het huidige document opgestuurd naar de PrintService. Deze is constant bezig met het verwerken van data tot documenten. De plaats in de print wachtrij kan worden bekeken door op de Print Queue te drukken. Voltooide documenten kunnen ook via deze knop worden opgehaald.



Figuur 12: De printwachtrij, deze wordt automatisch om de 10 seconden bijgewerkt.

Nadat het verwerken is voltooid, komt er als bericht een link naar het document te staan. Dan is het een kwestie van het bestand openen en afdrucken.



Figuur 13: Na verwerken krijgt de gebruiker een PDF bestand.

5.1.2 De Concentrator WebToPrint Module

De Concentrator WebToPrint module is een webservice die de koppeling vormt tussen de Concentrator en de Designer. Deze koppeling wordt gevormd door een aantal functies via SOAP open te stellen, die de Designer dan aan kan roepen.

De WebToPrint module zorgt ook voor de authenticatie van gebruikers in de Designer, en zorgt ervoor dat de data die van de Designer komt geen fouten bevat.

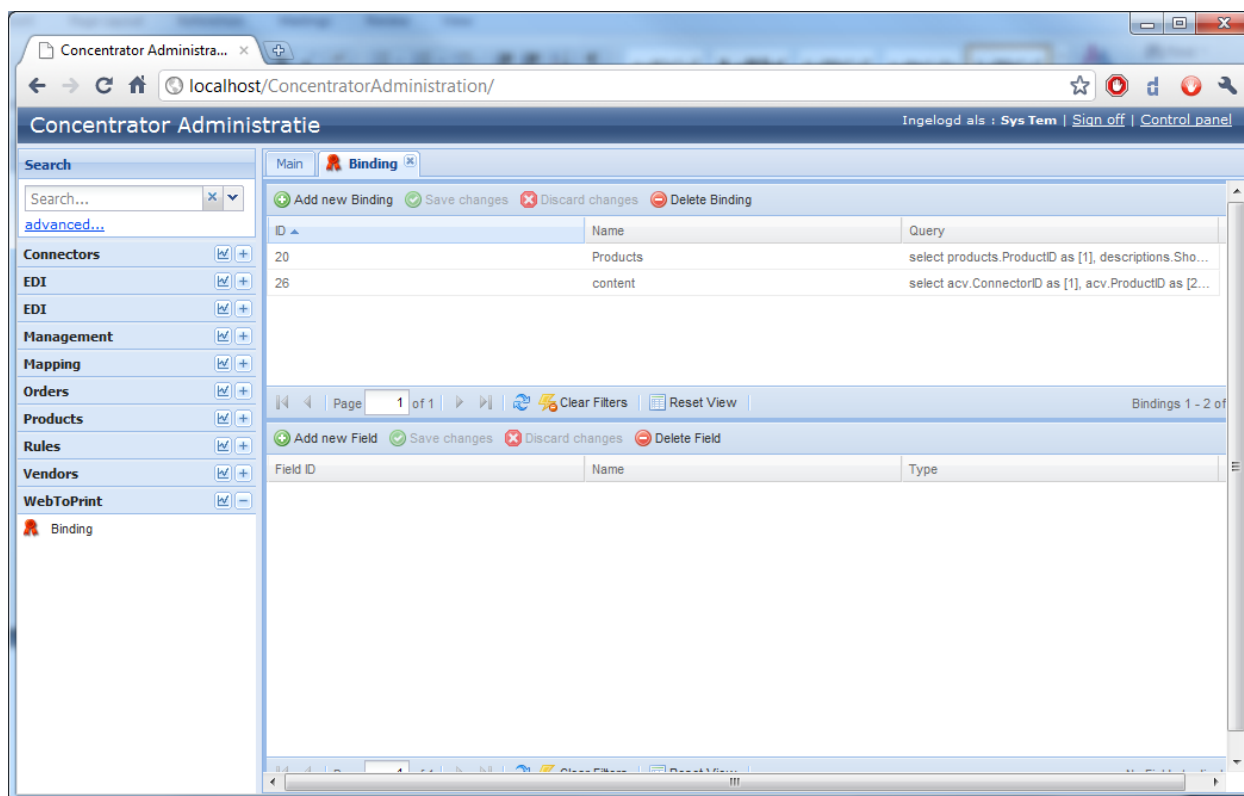
5.1.3 De Concentrator Print Service

De Concentrator Print Service is een plugin die elk kwartier door de Concentrator uitgevoerd wordt. Deze printservice neemt de opgeslagen documenten die in de printqueue staan, en verwerkt deze tot PDF documenten.

Door de print service los te koppelen van de WebToPrint module wordt de belasting op de server verdeeld. Het zorgt ervoor dat er geen conflicten ontstaan door bijvoorbeeld twee documenten die tegelijkertijd verwerkt worden. Door de PDF generatie op de Concentrator server te laten lopen kunnen ook veel grotere documenten verwerkt worden die anders te lang zouden duren als het in de Designer zelf verwerkt zou worden.

5.1.4 De Concentrator WebToPrint Administratie App

De Concentrator WebToPrint Administratie App is een onderdeel van de Concentrator Administratie pagina. Dit onderdeel kan gebruikt worden om projecten, documenten, pagina's en composiet elementen te beheren. De belangrijkste functie van de Administratie App is het toevoegen van datasources voor de Binding in de Designer. Deze datasources bestaan uit een query die de data ophaalt uit de Concentrator Database, en een aantal inputs en outputs waaruit de Designer de queries kan laten uitvoeren, en de terugkomende data kan verbinden met velden in het ontwerp.



Figuur 14: De WebToPrint Administratie App

Een van de handige features is dat de gebruiker tijdens het aanmaken alleen de query in hoeft te vullen: de WTP Administratie app zal zelf de inputs en outputs automatisch definiëren.

5.2 Project evaluatie

Als laatste onderdeel van dit scriptiedocument zal er een evaluatie van het project worden beschreven. Hoewel het project nog niet is afgerond ten tijde van het schrijven van dit document, zijn er wel een aantal bevindingen over het verloop van het project te beschrijven.

5.2 Wat ging goed

Een van de beste beslissingen van het project was het gebruik van de een ontwikkelmethode gebaseerd op Agile/Extreme Programming. Bij de start van het project waren nog niet alle voorwaarden en benodigdheden van het project bekend, en deze kwamen naar voren bij de verschillende onderzoeken. Omdat deze ontwikkelmethode is gebruikt kon er tijdens de ontwikkeling wensen van klanten in het ontwikkeltraject toegevoegd worden, en daardoor is het project ook beter afgestemd op die wensen.

En ander goed punt was het ontwikkelen van het prototype voordat de echte applicatie zou worden ontwikkeld. Dit gaf ons kans om bekend te worden met WPF, te kijken wat de mogelijkheden van WPF waren en te bekijken, te kijken of de Ribbon geschikt was voor onze doeleinden en te testen of de applicatie in alle browsers zou werken.

Als laatste de projectfasering, waardoor er bij elk beslissingsmoment werd gekeken wat er moest gebeuren, en er een to-do lijst werd opgesteld. Dit gaf duidelijke doelen tijdens het ontwikkelen en het bakende het project goed in verschillende onderdelen af.

5.3 Wat kon beter

Tijdens het technologieonderzoek werd er voor de Microsoft implementatie voor de Ribbon gekozen, terwijl we halverwege het project er achter kwamen dat een aantal grote onderdelen uit het Ribbon project niet af waren. Een grondiger onderzoek naar de verschillende Ribbons had kunnen voorkomen dat er veel tijd in het herimplementeren van deze onderdelen zou worden gestoken.

Daarnaast is op ongeveer driekwart van het project het team met de meeste kennis over WPF naar een andere locatie verhuisd, waardoor vragen over WPF lastiger gesteld konden worden.

6 BIJLAGEN

6.1 MVVM UML Schema

Omdat in eerste instantie was besloten om voor het prototype een MVVM structuur te ontwerpen, is hier een UML schema voor gemaakt.

6.2 MVC UML Schema

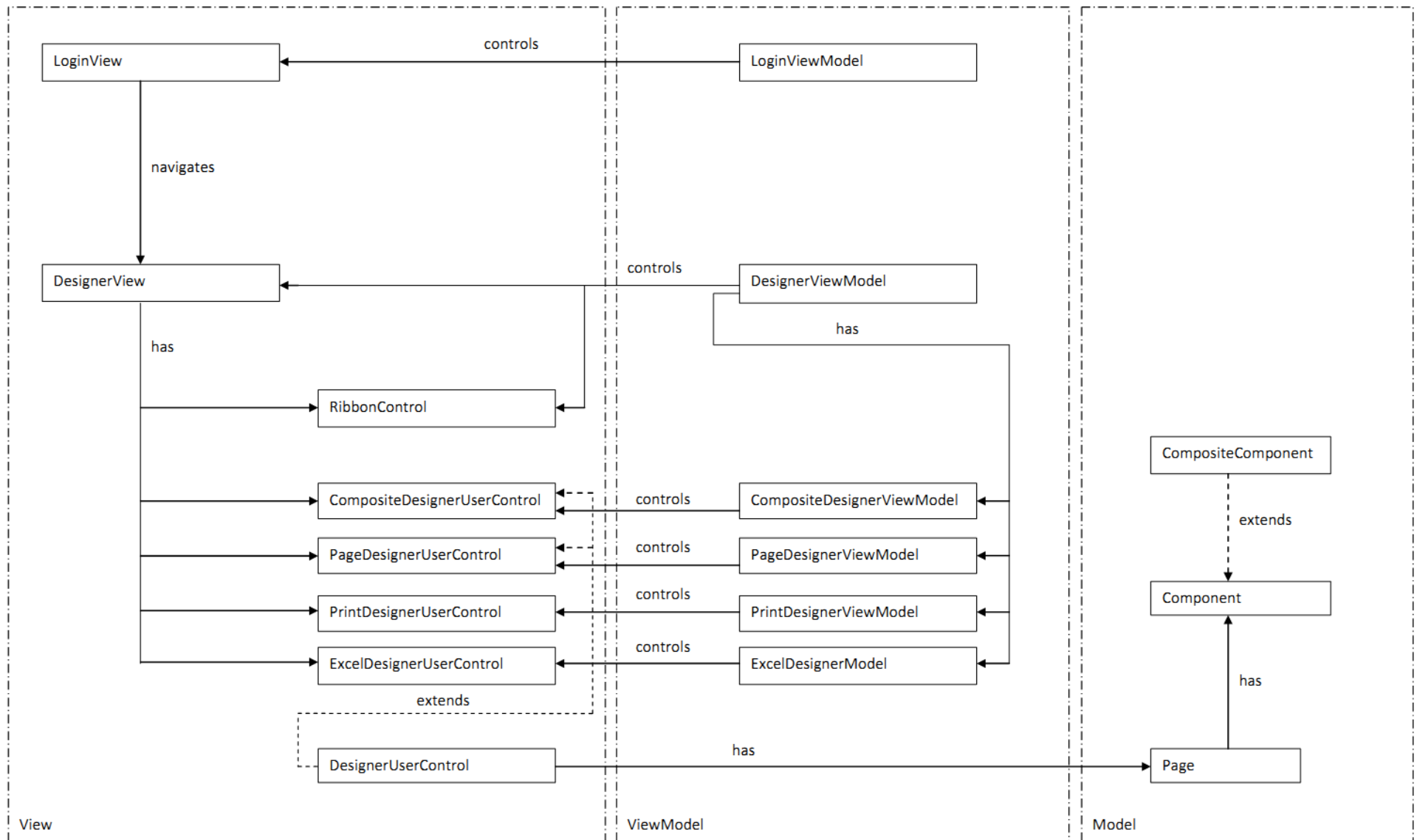
Tijdens het uitwerken van het prototype werd er besloten om over te stappen naar het MVC model, omdat dit een beter gestructureerd programma op zou leveren. Hiervan is ook een UML schema gemaakt.

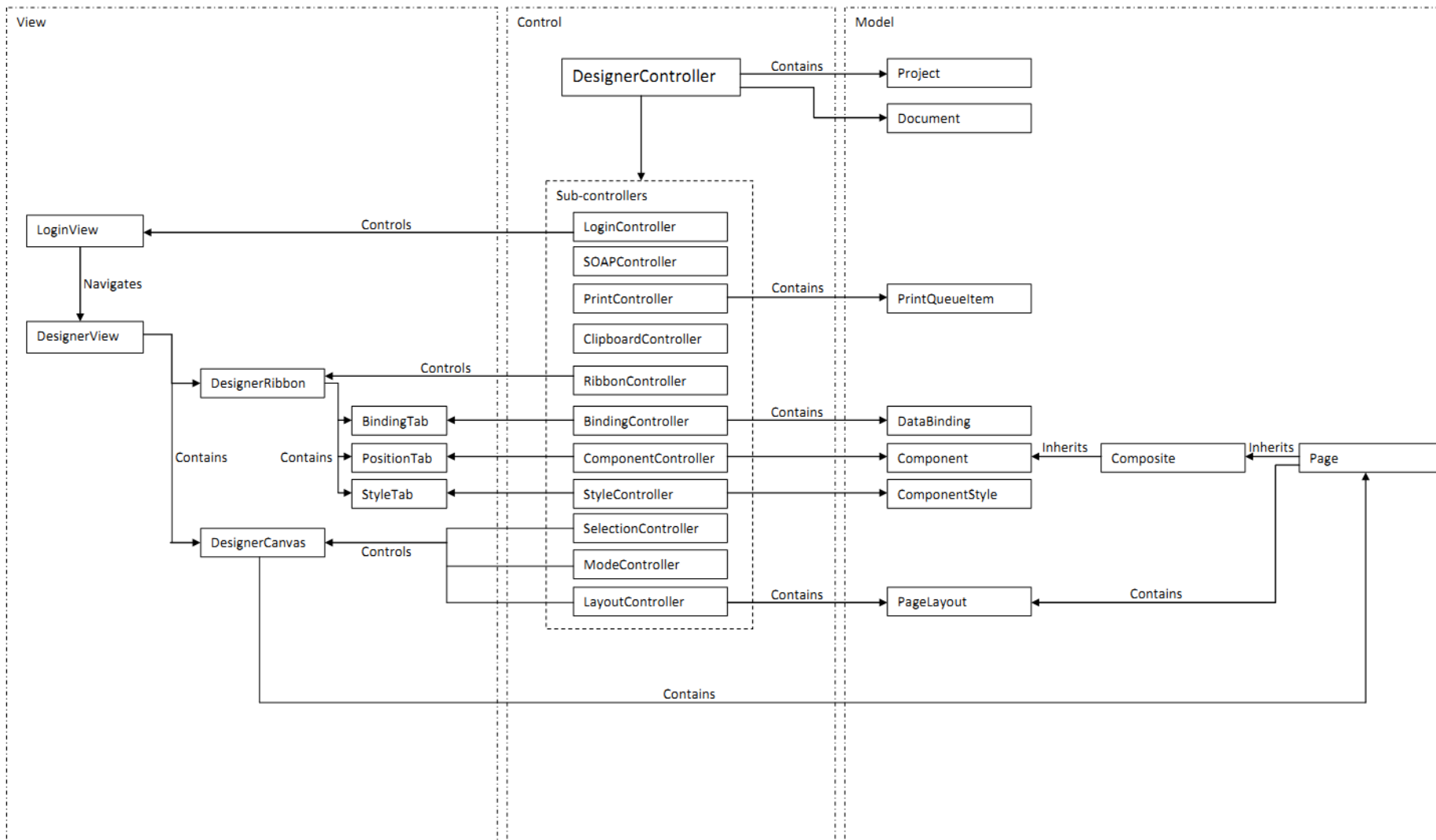
6.3 Promotiemateriaal

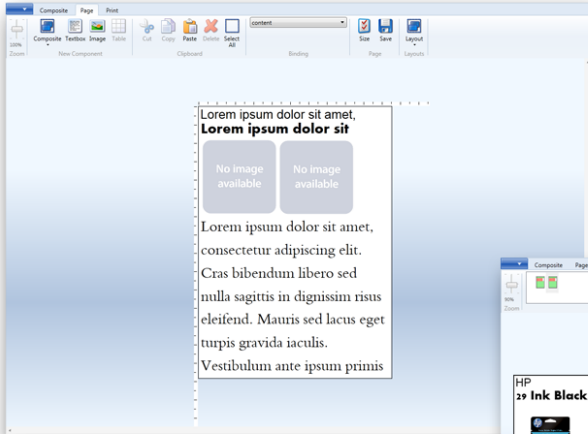
Om het product ook aan klanten te kunnen verkopen is er ook wat promotiemateriaal gemaakt om in powerpoint presentaties te zetten.

6.4 Plan Van Aanpak

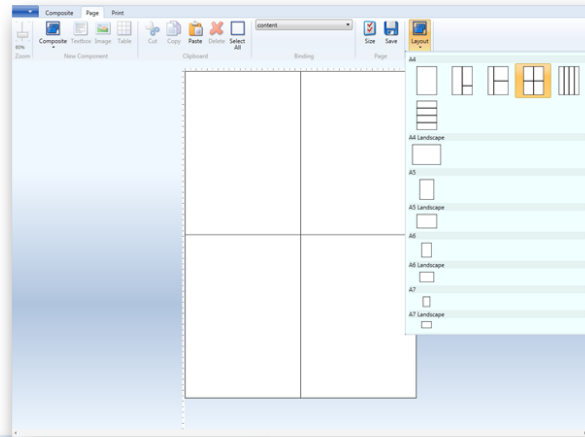
Het originele plan van aanpak wat is ingeleverd bij André van Rijn



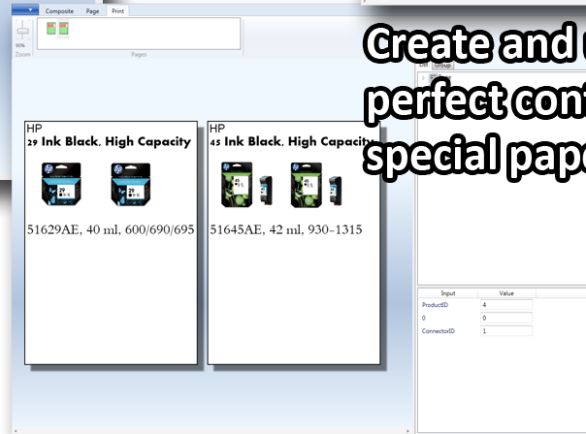




Easy template creation by drag 'n' drop content boxes



Create and use templates for perfect content alignment or special paper sizes



Retrieves content directly from Concentrator



Designing content was never this easy!

Plan van Aanpak

Versie 1.0



Web-To-Print Module Concentrator

Afstudeerder:

Arno Becker, 1536748

Hogeschool Utrecht

Afstudeerbedrijf:

Direct-IT

0.1 ALGEMENE GEGEVENS

Afstudeerder:

Arno Becker 1536748
Eekhoornstraat 26
1216 AS Hilversum
tel. : 06-24190417
e-mail: arno.becker@gmail.com

Afstudeerbedrijf:

Diract IT
Versterkerstraat 12
1322 AP Almere
tel.: 036-5458939
e-mail: info@diract-it.nl

Afstudeerproject:

Onderzoeken van en ontwerpen van een web-to-print module voor Concentrator.

Afstudeerbegeleiders:

Hogeschool Utrecht:
Dhr. B. Markus

Diract-IT:

Dhr. T. Zeeman
Ontwikkelaar Concentrator
Dhr. A. van Rijn
Financieel Directeur Diract

Afstudeerrichting

Informatica

0.2 INHOUDSOPGAVE

0.1 Algemene Gegevens	7
0.2 Inhoudsopgave	8
1 Introductie	9
1.1 Aanleiding	9
2 Opdracht Omschrijving	8
2.1 Probleemstelling	8
2.2 Doelstelling	9
2.3 Opdracht	9
3 Projectactiviteiten en producten	13
3.1 Technisch onderzoeksfase	13
3.2 Prototype designerfase	13
3.3 Prototype Concentrator module fase	14
3.4 Implementatiefase	14
3.5 Testfase/ gebruikresonderzoek fase	14
3.6 Opleverfase	14
4 Projectgrenzen en randvoorwaarden	15
4.1 Projectgrenzen	15
4.2 Randvoorwaarden	15
5 Kwaliteitsbewaking van de afstudeeropdracht	16
5.1 Kwaliteit	16
5.2 Terugkoppeling	16
6 Project Organisatie	17
6.1 Leden	17
7 Planning	18

1 INTRODUCTIE

1.1 Aanleiding

De Ceyenne Concentrator is een CMS voor het beheren van informatie die gebruikt wordt door een aantal andere producten van Direct, zoals het Ceyenne POS/WMS. Gebruikers van deze producten hebben aangegeven dat een optie om vanuit de Concentrator direct data te kunnen exporteren en kunnen printen welkom zou zijn.



,
voor de concentrator waarin administrators gedeeltes van de database open kunnen stellen voor de designer, een designer waarin gebruikers folders/productlijsten kunnen ontwerpen en een module voor de Concentrator die documenten gemaakt in de designer kunnen uitprinten.

2 OPDRACHT OMSCHRIJVING

2.1 Probleemstelling

De Concentrator is een web-based beheerpakket voor het aanmaken, aanpassen en verwijderen van artikelen die gebruikt worden in andere applicaties binnen het bedrijf van de gebruiker, en bevat op dit moment meer dan 100,000 items, en dit aantal groeit snel. Omdat dit pakket een gigantische hoeveelheid informatie bevat is er bij de gebruikers een vraag om direct vanuit de applicatie hardcopies van deze informatie te maken. De Concentrator extraheert data van verschillende bronnen en kan dit op slimme wijze met elkaar matchen hierdoor kunnen bedrijven hun data zeer consistent maken. Omdat de Concentrator consistente data bevat is dit een prima bron voor het opmaken van brochures en andere product documenten. Op dit moment is de Concentrator in staat alle producten in verschillende formaten te exporteren, maar zijn design pakketten niet altijd geschikt voor het eenvoudig importen van data (zonder data te verliezen). Het zou een prima oplossing zijn wanneer Web to Print geïntegreerd is in de Concentrator Applicatie waardoor eenvoudig gebruik kan worden gemaakt van de beschikbare data en alle data kan worden gebruikt voor het ontwerpen van documenten (folders/webpagina's/brochures/e.d.)

Het is de bedoeling dat de Concentrator uitgebreid wordt met 2 modules: Een design en een print module. In de design module kan een gebruiker in de webbrowser een ontwerp van de print maken (denk aan een lijst van artikelen in eigen huisstijl, of per artikel een A7 prijskaartje), en dit ontwerp op de server opslaan. In de 2e module (de print module) kan dan met een van deze ontwerpen, samen met een lijst van artikelen, een document gegenereerd worden. Dit document wordt dan bijvoorbeeld in een PDF op de server gemaakt, waarna de gebruiker deze kan downloaden, en printen of op zijn website voor klanten plaatsen.

Omdat nog niet zeker is wat de beste manier is om gebruikers dit te laten doen (het maken van een ontwerp, het opslaan van het ontwerp naar de server, het gebruiken van dit ontwerp op de server om een document te genereren), zal er eerst een onderzoek opgesteld moeten worden. In dit onderzoek moet worden gekeken naar verschillende frameworks voor het maken van layouts (een voorbeeld hiervan is Adobe Rome, maar mocht geen pakket aan de eisen voldoen, zal er ook aandacht besteed moeten worden aan de haalbaarheid van het zelf implementeren van de ontwerpmodule in bijvoorbeeld een combinatie van Flash/Silverlight, Javascript en CSS. Ook het te exporteren van het document bevat een aantal vraagstukken, vooral welk formaat de klanten het liefst willen ontvangen (een PDF lijkt voor de hand te liggen, maar er zijn zat andere formaten zoals Microsoft Word's doc/docx, ODF (open document format), of zelfs een simpel ascii txt bestand.)

2.2 Doelstelling

De doelstelling van dit project is een systeem ontwerpen waarmee gebruikers zelf een ontwerp voor een document kunnen maken, dit kunnen linken met data uit de Concentrator, en dit om kunnen zetten in een printbaar document.

Om dit te realiseren moet er eerst onderzoek worden gedaan naar de verschillende technologieën die gebruikt kunnen worden, moet er een onderzoek naar de precieze wensen van de klanten uitgevoerd worden, de module gemaakt en geïmplementeerd worden, en als laatste een publieke test om eventuele fouten eruit te halen.

2.3 Opdracht

De opdracht bestaat uit het onderzoeken van de gewenste oplossing en het realiseren hiervan. De Concentrator is op dit moment voornamelijk in gebruik bij klanten in de retail en distributie branche. Onderdeel van het onderzoek zal het in overleg met de klanten bepalen van de gewenste functionaliteiten zijn en het vast stellen van de gebruikersinterface. Daarnaast moet er gekeken worden welke technologie het beste bij dit probleem past.

De uitbreiding zal voor de backend op ASP.NET ontwikkeld worden, met C# als programmeertaal. Deze taal wordt verder gebruikt in Silverlight en XNA, en biedt daardoor genoeg mogelijkheden. Als ontwikkelomgeving zal Visual Studio van Microsoft gebruikt worden, met versiebeheer in Git. De designer (het ontwerpen van de layout van de prints) zal worden gemaakt in Adobe Fireworks, Adobe Flash of Silverlight, wat uitgezocht zal worden in een onderzoek voor het ontwikkelen begint. Communicatie tussen de client applicatie (de designer) en de backend (Concentrator) zal worden verzorgd met asynchrone communicatie over AJAX, en vanuit daar zal de module intern met de Concentrator verbinding maken om informatie over producten op te halen. Om specifieke data zoals layouts op te slaan zal een MS-SQL server worden gebruikt. Als projectomgeving zal er een Agile/Extreme Programming methode gebruikt worden, waardoor om de week een voortgangsrapportage en om de maand een voortgangspresentatie/evaluatierapport wordt gegeven.

3 PROJECTACTIVITEITEN EN PRODUCTEN

Het project zal worden opgedeeld in verschillende fases:

- 1) Onderzoek technieken
- 2) Prototype designer
- 3) Prototype Concentrator module
- 4) Designer ontwikkeling
- 5) Binding module
- 6) Koppeling Binding/Designer
- 7) Testen
- 8) Opleveren

3.1 Technisch onderzoeksfase

Doel: Uitzoeken welke technologieën het beste geschikt zijn voor de verschillende onderdelen van het project

Producten: Een idee van de mogelijkheden van verschillende talen/technieken

Betrokkenen: Student, Tim Zeeman

Tijd: ~2 weken

3.2 Prototype designerfase

Doel: Implementeren van die techniek voor een prototype van de Designer. Veel functionaliteit als mockup, om een indruk van de web-to-print module aan klanten te geven.

Producten: Prototype van de designer

Betrokkenen: Student, Tim Zeeman

Tijd: ~3 weken

3.3 Prototype Concentrator module fase

Doel: Implementeren van de plugin voor Concentrator, en implementeren van mockup van bindings.

Product: Prototype van de web-to-print module

Betrokkenen: Student, Tim Zeeman

Tijd: ~2 weken

3.4 Implementatiefase

Doel: Implementeren van de de web-to-print module

Product: Web-to-print module

Betrokkenen: Student, Tim Zeeman

Tijd: ~10 weken

3.5 Testfase/ gebruikresonderzoek fase

Doel: Eindgebruikers product laten testen, last-minute bugs fixen

Product: geen.

Betrokkenen: Student, Tim Zeeman, gebruikers web-to-print

Tijd: ~2 weken

3.6 Opleverfase

Doel: Project afronden, overdracht aan andere developers, code opschonen

Product: Web-To-Print module, gebruikershandleiding

Betrokkenen: Student, Tim Zeeman, gebruikers web-to-print

Tijd: ~2 weken

4 PROJECTGRENZEN EN RANDVOORWAARDEN

Door aan het project grenzen en voorwaarden toe te kennen kan worden afgebakend wat wel en wat niet tijdens de afstudeerperiode wordt meegenomen.

4.1 Projectgrenzen

- Het project start op 31 Januari 2011 en eindigt op 11 Juni 2011
- De opdracht die wordt uitgevoerd staat beschreven in paragraaf 2.3
- De opdracht wordt uitgevoerd binnen Direct IT
- De opdracht wordt uitgevoerd als uitbreiding op de Concentrator

4.2 Randvoorwaarden

- De designer moet web-based zijn
- Gebruiksvriendelijke interface voor niet technische gebruikers
- Naadloze integratie met huidige software (Concentrator)

5 KWALITEITSBEWAKING VAN DE AFSTUDEEROPDRACHT

5.1 Kwaliteit

Om de kwaliteit van het project te waarborgen worden de volgende stappen ondernomen:

- Tim Zeeman word met elke vordering van het project mondeling op de hoogte gesteld
- Voor vragen kan ik altijd bij een van de andere developers van DirectIT terecht, en voor specifieke vragen bij Stan Todorov of Tim Zeeman.
- Elke maandag is er een overleg met Tim gepland om de voortgang van het project, en of er nog specifieke problemen zijn.
- Later in het project zal er een feedback lus ontstaan tussen mij, Tim en de eindgebruikers.

5.2 Terugkoppeling

- Er word minimaal 2 keer per week overleg met Tim gepleegd over de voortgang van het project.
- Dagelijks worden er mondelinge voortgangsrapportages aan Tim of Stan gepleegd.

6 PROJECT ORGANISATIE

Het project heeft de volgende projectorganisatie

Projectleider: Tim Zeeman

IT ondersteuning: Stan Todorov

6.1 Leden

Tim Zeeman

e-mail: t.zeeman@direct-it.nl

Functie: Lead Developer Concentrator

Rol: Ondersteuning stage, projectvoortgang

Stan Todorov

e-mail: s.todorov@direct-it.nl

Functie: Software Developer Concentrator

Rol: Ondersteuning software development

7 PLANNING

Stage	Weeknummer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Datum		31-jan	7-feb	14-feb	21-feb	28-feb	7-mrt	14-mrt	21-mrt	28-mrt	4-apr	11-apr	18-apr	25-apr	2-mei	9-mei	16-mei	23-mei	30-mei	6-jun	13-jun	20-jun
Plan van aanpak opstellen																						
Technische onderzoeksfase																						
Uitzoeken framework designer																						
Uitzoeken framework web-to-print																						
Uitzoeken PDF printing																						
Prototype designerfase																						
Uitzoeken UI mogelijkheden																						
Uitzoeken componentmodel																						
Implementeren prototype																						
Prototype concentrator module fase																						
Uitzoeken verbinding designer-concentrator																						
Uitzoeken binding implementatie																						
Implementeren prototype																						
Implementatiefase																						
Implementeren Designer																						
Implementeren Concentrator Module																						
Implementeren Binding																						
Implementeren PDF printing/export																						
Implementeren save/load																						
Testfase																						
Opleverfase																						