

Auteur	Niels Wijers
Studentnummer	2061798
Bedrijf	Relate4u
Adres	Esp 405
Postcode en plaats	5633 AJ Eindhoven
Telefoon	+31 (0)40 280 23 45
Fax	+31 (0)40 280 23 44
Website	www.relate4u.com
E-mail	info@relate4u.com

VOORWOORD

Deze scriptie is geschreven in het kader van de afstudeeropdracht als onderdeel van de opleiding ICT Media design aan de Fontys Eindhoven. Ik heb in eerste instantie gekozen om af te studeren bij Relate4u omdat de sfeer mij erg aansprak en de opdracht die ik zou gaan uitvoeren me een uitdaging en interessant leek. Bij aanvang van het project had ik al een beetje ervaring met Actionscript en het maken van online applicaties, maar ik heb door het werken aan deze opdracht heb ik hier de afgelopen periode enorm veel van op gestoken.

In dit boek beschrijf ik mijn ervaringen over de opdracht en het onderzoek dat ik uitgevoerd heb voor Relate4u. Relate4u is een middelgroot ontwerp en ICT bureau met verschillende grote klanten, zoals Philips, Adidas, Volvo en Klene. In de afgelopen 20 weken heb ik me bezig gehouden met ontwikkelen van een toolkit in Flash en Flex, waarin de vormgevers van Relate4u eenvoudig en snel internet campagnes kunnen opzetten.

Tijdens mijn stage heb ik veel plezier gehad, maar vooral ook veel geleerd op alle vlakken die mijn opleiding overlappen. Hiervoor wil ik graag alle medewerkers van Relate4u bedanken. Graag bedank ik in het bijzonder Mischa Zabransky. Mischa is de Artdirector binnen Relate4u en tevens mijn bedrijfsbegeleider. Ook bedank ik Thijs van Gerven en Mieke van Vucht voor de ondersteuning vanuit school en de feedback op dit verslag.

Kortom, het was een drukke en leuke stage en wens ik iedereen die dit leest veel leesplezier.

09-06-2008., Eindhoven

Niels Wijers

INHOUDSOPGAVE

Voorwoord	2
Samenvatting	7
Summary	8
Verklarende woordenlijst	9

Hoofdstuk 1. Inleiding 11

Hoofdstuk 2. Relate4u 13

2.1. Ontstaansgeschiedenis	14
2.2. Missie en visie	15
2.2.1 Simplify media complexity	15
2.2.2 Combineren van creatieve ontwerpen met oplossingen	15
2.2.3 Enthousiasme en specialisme	16
2.3. Organogram	16

Hoofdstuk 3. Doelstelling en probleemstelling 19

3.1. Doelstelling van de opdracht	20
3.2. Doelstelling binnen het onderzoek	21
3.3. Probleemstelling	21
3.4. Deelvragen op de probleemstelling	22
3.4.1 Microcampagnes	22
3.4.2 Toolkit structuur	22
3.4.3 Database connecties	23
3.4.4 Flash componenten	23

Hoofdstuk 4. Onderzoek 25

4.1. Activiteiten	26
4.2. Op te leveren producten	26

Hoofdstuk 5. Microcampagnes 29

5.1. Redactioneel en commercieel	30
5.2. Viral marketing	31
5.3. Consument en adverteerder	32
5.4. Conclusie	33

Hoofdstuk 6. Algemene webapplicatie architectuur 35

6.1. Model-view-controller	36
6.1.1 Wat	36
6.1.2 Hoe	36
6.1.3 Waarom	38
6.2. Data Access Object	38
6.2.1 Wat	38
6.2.2 Hoe	38
6.2.3 Verbergen van structuur	40
6.3. Conclusie	40

Hoofdstuk 7. Communicatie 41

7.1. RPC en realtime communications	42
7.1.1 HTTPServices	42
7.1.2 Webservices	43
7.1.3 Remote Objects	43
7.2. Conclusie	45

Hoofdstuk 8. Flash structuur 47

8.1. De view-laag	48
8.2. De model-laag	49
8.3. De controller-laag	50
8.4. De business laag	50
8.5. Volledige structuur	52
8.6. Frameworks	53
8.6.1 Cairngorm vs PureMVC	53
8.7. Conclusie	55

Hoofdstuk 9. Flash ui componenten 57

9.1. Verschillen Flash en Flex	58
9.2. Overerving	59
9.3. Custom events	59
9.4. Parameters	60
9.5. Metadata	60
9.6. Styling	61
9.7. Conclusie	61

Hoofdstuk 10. PCC microcampagne 63

10.1. PCC-Klantendag	64
10.1.1 Doelstellingen	64
10.2. Concept	65

Hoofdstuk 11. Concept- en product ontwikkeling 67

11.1. De voorbereiding	68
11.1.1 Vereiste data	68
11.1.2 Wireframes	68
11.1.3 Gekozen software en werkwijzen	71
11.1.4 Overzicht van de toolkit	72
11.2. Definitieve opzet	74
11.2.1 UML	74
11.3. Realisatie	75

Hoofdstuk 12. Conclusie en aanbevelingen 77

12.1. Conclusie	78
12.2. Aanbevelingen	79

Evaluatie 80

Literatuurlijst 81

Bijlage A. Plan van aanpak 83

SAMENVATTING

Ik zal als afstudeeropdracht in dienst van Relate4u een aantal onderzoeken uitvoeren en een basis leggen voor de ontwikkeling van een beheer systeem voor microcampagnes. De onderzoeken en analyses zullen bestaan uit een onderzoek naar wat virals zijn, een onderzoek naar een goede structuur voor de opbouw van Flash applicaties en uiteindelijk te komen tot een functioneel en grafisch ontwerp.

Aan de hand van deze onderzoeken ben ik nagegaan hoe je het beste een systeem op kunt zetten waarmee het mogelijk is eenvoudig een microcampagne op te zetten en welke gegevens daarvoor van belang zijn.

De onderzoeksmethode bestond veelal uit het verzamelen en analyseren van relevante artikelen met betrekking tot de te gebruiken technieken, virale marketing en microcampagnes. Er is gekozen voor Adobe Flash en Adobe Flex als het gaat om de uiteindelijke ontwikkeling van de applicatie. Ik kan concluderen dat het model view controller patroon een zeer goede manier van het ontwikkelen is voor applicaties binnen een Flash omgeving, maar ook voor daar buiten.

Tevens zou ik PureMVC willen aanbevelen. Dit volwassen Framework bied een zeer goede structuur van werken en leert je meer inzicht te krijgen grootschalige Flash applicaties goed op te bouwen.

SUMMARY

For my final graduation project in service of Relate4u, I will conduct a number of researches and form a foundation for the development of a management system for microcampagnes. The various researches and analyses will consist of research about what virals are, how to build a firm structure for Flash applications and finally, a functional and graphic design.

Through this research I've learned how to best set up a system which would make it possible to easily design a microcampaign and which data is most important to this end.

The research method was mainly focused on the collecting and analysing of relevant articles about which techniques to apply, viral marketing and microcampagnes. For the eventual development of the application, the programs Adobe Flash and Adobe Flex were chosen.

In conclusion, I have found that the model view controller pattern is a very efficient way to develop applications within, but also outside of a Flash environment. I also recommend the use of PureMVC. This matured Framework offers an extremely good work structure while also providing you with better insight into building large Flash applications correctly.

VERKLARENDE WOORDENLIJST

Adobe flex

Adobe Flex 2.0 is een clientapplicatie waarmee met behulp van Macromedia Flex Markup Language (MXML) cliëntinterfaces gebouwd kunnen worden en bepaald worden hoe ze gaan functioneren met de back-end.

Application Programming Interface (API)

Een Application Programming Interface (API) is een verzameling definities op basis waarvan een computerprogramma kan communiceren met een ander programma of onderdeel

Data Access Object (DAO)

Een Data Access Object is een object dat een abstracte interface voor een database verzorgt. Hij zorgt voor voor bewerk operaties van de database zonder aan de buitenwereld te laten zien wat hij doet

Model-view-controller (MVC)

Model-View-Controller is een structuur dat het ontwerp van complexe toepassingen opdeelt in drie eenheden met verschillende verantwoordelijkheden: Datamodel (Model), Datapresentatie (View) en Applicatielogica (Controller). Het scheiden van deze verantwoordelijkheden bevordert de leesbaarheid en herbruikbaarheid van code.

Rich Internet Application (RIA)

Een Rich Internet Application is een term die gebruikt wordt voor interactieve internet-applicaties, die het gevoel geven van een desktopprogramma

Software Development Kit (SDK)

Een Software Development Kit is een verzameling hulpmiddelen die handig zijn bij het ontwikkelen van computerprogramma's voor een bepaald besturingssysteem of type hardware, of voor het maken van software die een speciale techniek gebruikt.

Value Object (VO)

Een Value Object is een object dat een rij uit de database vertegenwoordigt

HOOFDSTUK 1.

Inleiding

Ik voer mijn afstudeer opdracht uit bij Relate4u te Eindhoven. Relate4u is een allround multi-media bedrijf waar momenteel 40 mensen in dienst zijn. Ze houden zich bezig met DTP, print-automatisering, websites en motion graphics. Mijn werkzaamheden zullen vooral in het verlengde liggen van de websites afdeling van Relate4u. De opdracht is een onderzoek naar een systeem waarmee eenvoudig en snel microcampagnes opgezet kunnen worden, hierna gaat dit systeem gerealiseerd worden.

Relate4u is op zoek naar een systeem, waarin men makkelijk en snel microcampagnes kan opzetten. Ook moet dit systeem klanten en adverteerders inzicht geven in de resultaten van haar campagnes. Mijn rol hier in is om een basis te leggen voor dit systeem.

Het boek is als volgt opgebouwd. In hoofdstuk 2 is gedetailleerde informatie te vinden over Relate4u. Hierna volgen de probleem- en doelstellingen en de opzet van het onderzoek. Hoofdstuk 5 tot en met 10 zijn onderzoeksresultaten en conclusies op de verschillende deelvragen. In hoofdstuk 11 is het functioneel ontwerp van het systeem te vinden. Als laatste is er een test case voor het eindproduct en een eindconclusie.

HOOFDSTUK 2.

Relate4u

In het komende hoofdstuk wordt verteld hoe Relate4u nu eigenlijk ontstaan is en wat er allemaal gebeurd is tot wat Relate4u tot op de dag van vandaag is. Dit wordt gevolgd door de diensten die Relate4u levert.

2.1. ONTSTAANSGESCHIEDENIS

Het begin van Relate4u start in het jaar 2000. Op dat moment bestond Relate4u nog niet. Op dat moment was er Paro. Paro is een digitale drukkerij die voornamelijk folders, leaflets en flyers drukte. Haar grootste klant was op dat moment Philips Medical Systems.

Philips Medical Systems had meer wensen. Een probleem was dat ze verschillende e-mailadressen hadden waar mensen naar konden e-mailen als ze vragen of problemen hadden. Zij wensten dat alles naar één e-mailadres verzonden zou worden en dat het vanuit daar verdeeld werd onder de verschillende medewerkers. Dit project heeft Paro aangenomen en realiseerde dit onder de naam Globic. Ze maakten één e-mailadres en een applicatie die de e-mails sorteerte op onderwerp en land. Vervolgens werd de email naar de juiste persoon doorgestuurd. Het was een geslaagd project, en het wordt tot op heden gebruikt.

De oplossing die Paro realiseerde viel ver buiten de service die Paro normaal leverde. Drukken was de corebusiness van Paro en zodoende ontstond er dus een aparte bedrijfstak. Deze bedrijfstak besloot Paro af te splitsen naar een nieuw bedrijf. Het bedrijf dat ontstond is Relate4u.

Het idee achter Relate4u was er op gericht om meer services aan te bieden dan alleen maar drukwerk. Relate4u richtte zich daarom op IT solutions. Kort na het ontstaan van Relate4u werd al snel duidelijk dat er ook een grote vraag was naar ontwerp en creatie. Dit zorgde er voor dat Relate4u al snel op gedeeld werd in 2 afdelingen, Een IT afdeling en een design studio.

De groei van Relate4u was in 4,5 jaar zo snel gegaan dat ze besloten hadden om naar een eigen pand te verhuizen. En in 2005 was het zo ver. Ze verhuisden naar een eigen bedrijvenpand op Esp in Eindhoven. Hierdoor kregen zij niet alleen een eigen plekje, maar ook een eigen identiteit.

2.2. MISSIE EN VISIE

2.2.1 SIMPLIFY MEDIA COMPLEXITY

De diensten van Relate4u omvatten meer dan die van een traditioneel reclamebureau, ICT bedrijf of printproductiebedrijf apart. Wat Relate4u biedt is de krachtige combinatie van deze 3, creatie, applicatie en productie. Relate4u biedt geïntegreerde media, communicatie en publicatie oplossingen, die de klanten in staat stellen om op diverse manieren, tijdstippen en via verschillende media en met hun klanten te communiceren. De producties en oplossingen van Relate4u zijn uitermate geschikt voor alle cross media, content management en publicatievraagstukken. Relate4u zet de vaak complexe structuur van processen om naar eenvoud voor klanten en gebruikers.

2.2.2 COMBINEREN VAN CREATIEVE ONTWERPEN MET OPLOSSINGEN

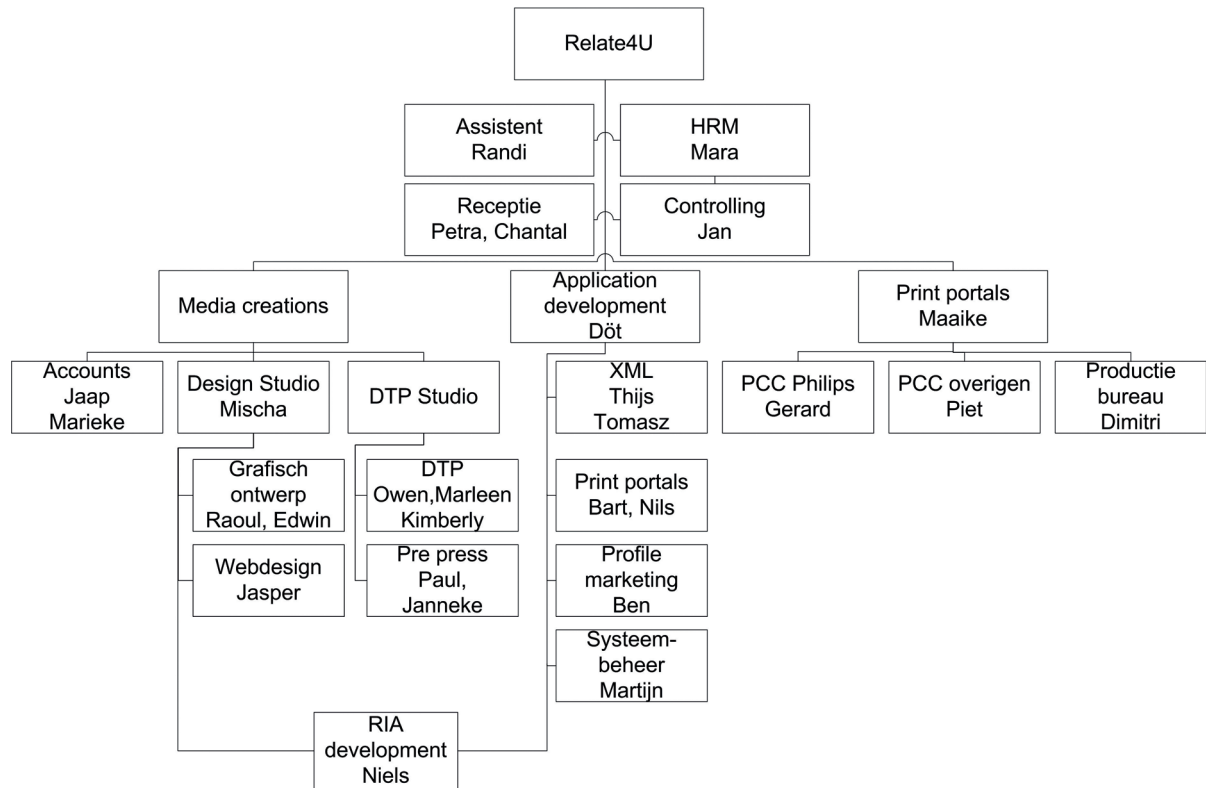
De alsmaar toenemende concurrentie en de steeds ingewikkelder wordende media, zijn feiten waar (grote) bedrijven mee leren moeten leven. Daarbij vragen nieuwe, complexe technologieën en alsmaar veranderende wensen en eisen van consumenten van een organisatie dat zij zich steeds aanpast aan de nieuwe omstandigheden in het huidige medialandschap. Met de middelen die Relate4u in huis heeft kunnen de efficiëntie en de armslag van een organisatie vergroten. Hiermee kan een bedrijf zich differentiëren van de concurrentie. Door creatief ontwerpen met applicatieontwikkeling te combineren, zijn wij in staat innovatieve XML, Print Portaal, Media en Marketingoplossingen te bieden voor organisaties, waar dan ook ter wereld.

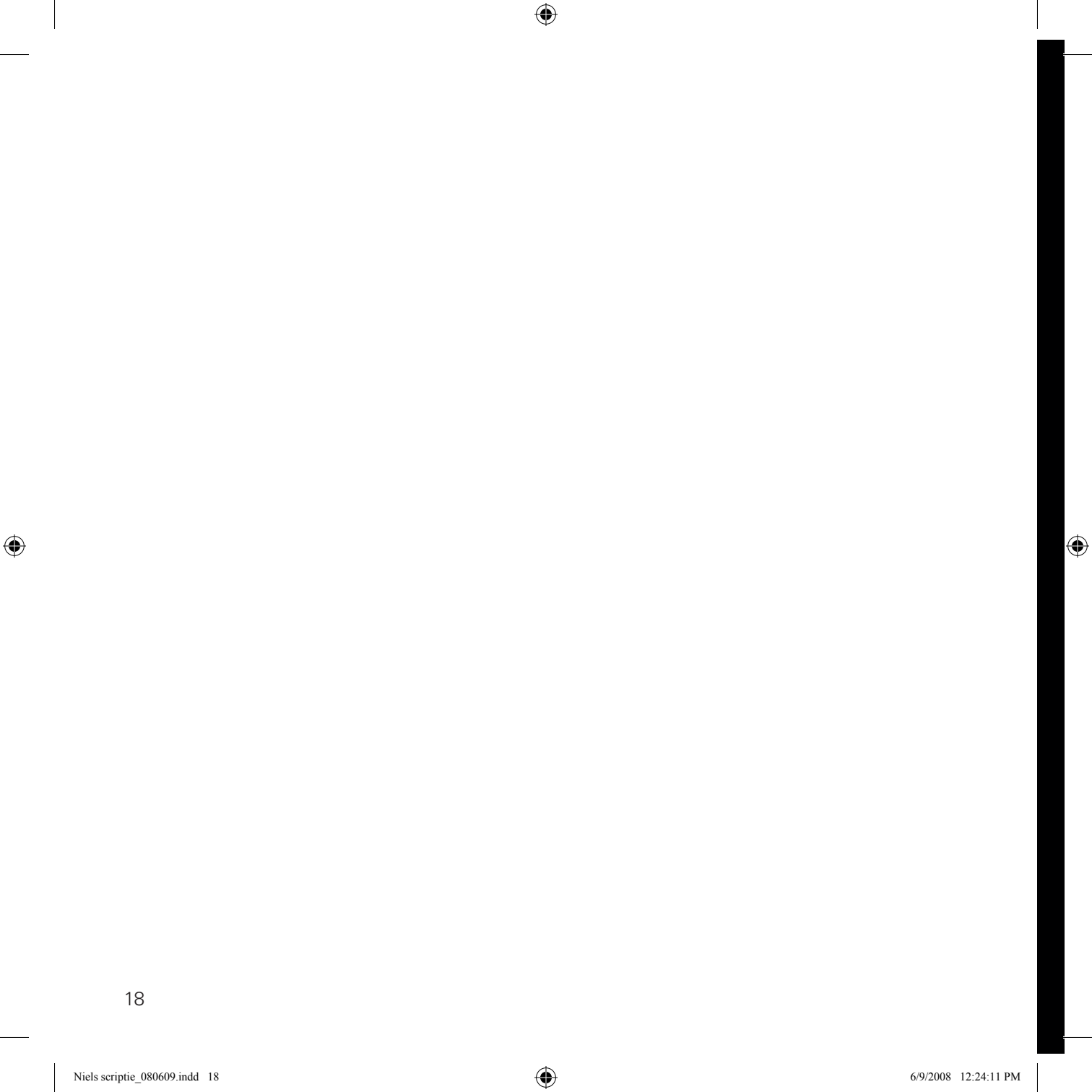
2.2.3 ENTHOUSIASME EN SPECIALISME

Relate4u bestaat uit een groep enthousiaste en ervaren mensen, met specialismen in grafisch design, marketing, print productie en webapplicaties. Het doel is eenvoudigere en betere bedrijfsoplossingen voor haar klanten te creëren. Van ontwerp en conceptontwikkeling tot creatie en productie. Elk project wordt benaderd vanuit het perspectief van onze klant. Relate4u zorgt er voor dat de technologie doet wat de klant verwacht

2.3. ORGANOGRAM

Rechts staat de organogram van Relate4u. Mijn werkplaats is op de Development afdeling, maar ik werk zowel samen met het Application development team als met de designstudio.





HOOFDSTUK 3.

Doelstelling en probleemstelling

Relate4u wil voor haar klanten mogelijk maken snel en goed koop microcampagnes aan te bieden. Dit hoofdstuk vertelt de doelstellingen van de opdracht. Na het analyseren van deze probleemstellingen levert dit een doelstelling op. Om tot een juiste conclusie te komen om de opdracht tot een succes te brengen zijn er subdoelstellingen opgesteld.

3.1. DOELSTELLING VAN DE OPDRACHT

Er moet een product komen voor vormgevers dat het mogelijk maakt om gemakkelijk een microcampagne te lanceren. Dit is precies wat Relate4u wil dit hebben, zodat ze snel geavanceerde microsites kan aanbieden aan haar klanten. Op het moment is er bij Relate4u nog geen kennis op dit gebied en wil het graag onderzoek doen om deze unieke mogelijkheid te exploreren.

Het is de bedoeling dat Viralz4u, genaamd Viralz4u, gemakkelijk te gebruiken is voor de vormgevers en een standaardwerkwijze en structuur bevat om deze kleine sites te realiseren. Een voorwaarde is natuurlijk wel dat het Flexibel moet zijn, zodat de vormgever alle vrijheid heeft de site te ontwerpen. Het kan dus niet de bedoeling zijn dat de vormgever vast zit aan bepaalde pagina's die hij moet creëren of bepaalde componenten die hij niet naar de stijl van de site kan aanpassen.

Vanuit het perspectief van de klanten zijn er ook verschillende dingen waar rekening mee gehouden moet worden. Relate4u wil haar klanten namelijk aanbieden om een bepaalde dataverrijking te verkrijgen, nadat de site online is en wordt gebruikt. Dit kan bijvoorbeeld zijn dat er een lijst komt met geregistreerde gebruikers, maar het kan nog veel verder gaan zoals data uit enquêtes, games of polls (stellingen). De klant kan inloggen op een achterliggend systeem om deze data ten aller tijden op te vragen.

Aangezien er vooraf niet duidelijk is welke klanten van Relate4u Viralz4u gaan gebruiken en welke wensen deze hebben, is het erg belangrijk dat de structuur van Viralz4u breed wordt opgezet en dat het voor een developer mogelijk is het altijd te kunnen uitbereiden.

3.2. DOELSTELLING BINNEN HET ONDERZOEK

De opzet van Viralz4u is heel breed en moet dus uitbreidbaar zijn, maar dit wil niet zeggen dat er voor Viralz4u geen grenzen zijn. Alle userinterfaces zullen draaien in Flash, de data worden opgeslagen in MySQL database en de server side technologie die gebruikt wordt is PHP. Zoals eerder verteld moet Viralz4u uitbreidbaar en zal het uiteindelijk niet uitmaken welke technologieën er gebruikt worden. Voor Viralz4u is nu gekozen voor deze technologieën, maar in de toekomst zouden er andere bij kunnen komen. Dit hangt af van de developers die hier later extensies voor zullen maken.

Viralz4u krijgt drie userinterfaces. Elke userinterface heeft een functie voor een bepaalde groep gebruikers. Er komt een userinterface voor de vormgevers, hierin kunnen zij projecten aanmaken. Er komt een userinterface voor de klant van Relate4u. Hierin kan men statistiek van de campagne site bekijken. De laatste userinterface is voor de daadwerkelijk gebruikers van de campagne website zelf. Uit deze laatste wordt de data gegenereerd die de klanten later kunnen terug zien.

3.3. PROBLEEMSTELLING

Er moet een toolkit komen dat aan deze wensen voldoet. Het woord toolkit is natuurlijk zeer abstract. Eigenlijk is het een verzameling van Flash componenten, databases, server-side scripts en natuurlijk een duidelijke manier van werken. De toolkit heeft een database, waarin alle gegevens worden opgeslagen. Deze data wordt gevuld met data van de campagne gebruikers en door de vormgevers als zij het project aanmaken of eventueel wijzigen. De opgeslagen data kunnen weer opgehaald worden door de klanten van Relate4u.

Het is niet dat deze toolkit van essentieel belang is voor Relate4u, maar deze toolkit zou ervoor kunnen zorgen dat Relate4u bepaalde diensten beter en sneller zou kunnen aanbieden. Ook zorgt het er voor dat verschillend werk niet meer dubbel gedaan hoeft te worden. Wat dus uiteindelijk geld bespaart.

De centrale vraag is dus. “Hoe maak je een toolkit waarmee eenvoudig micro-campagnes kunnen worden gemaakt en welke gegevens worden veelal in deze campagnes gebruikt”

3.4. DEELVRAGEN OP DE PROBLEEMSTELLING

Er zijn verschillende struikelblokken en onduidelijkheden die eerst onderzocht zullen moeten worden om tot het gewenste resultaat te kunnen komen. De volgende dingen zullen nog onderzocht moeten worden

3.4.1 MICROCAMPAGNES

Wat zijn microcampagnes? Welke data vragen ze in het algemeen? Dit zijn vragen die belangrijk zijn voor het aanmaken van de databases. Ook voor de dataflow binnen de applicaties is het prettig om een goed beeld te hebben welke data er nu daadwerkelijk zijn.

3.4.2 TOOLKIT STRUCTUUR

Er moet onderzocht worden welke structuur het beste is om dit hele product op te zetten. De toolkit zal volledig object georiënteerd opgezet worden. Een bekend patroon dat veelvuldig in objectgeoriënteerd programmeren voor komt is het model view controller (MVC) patroon. De toolkit zal zeker gebruikmaken van dit patroon.

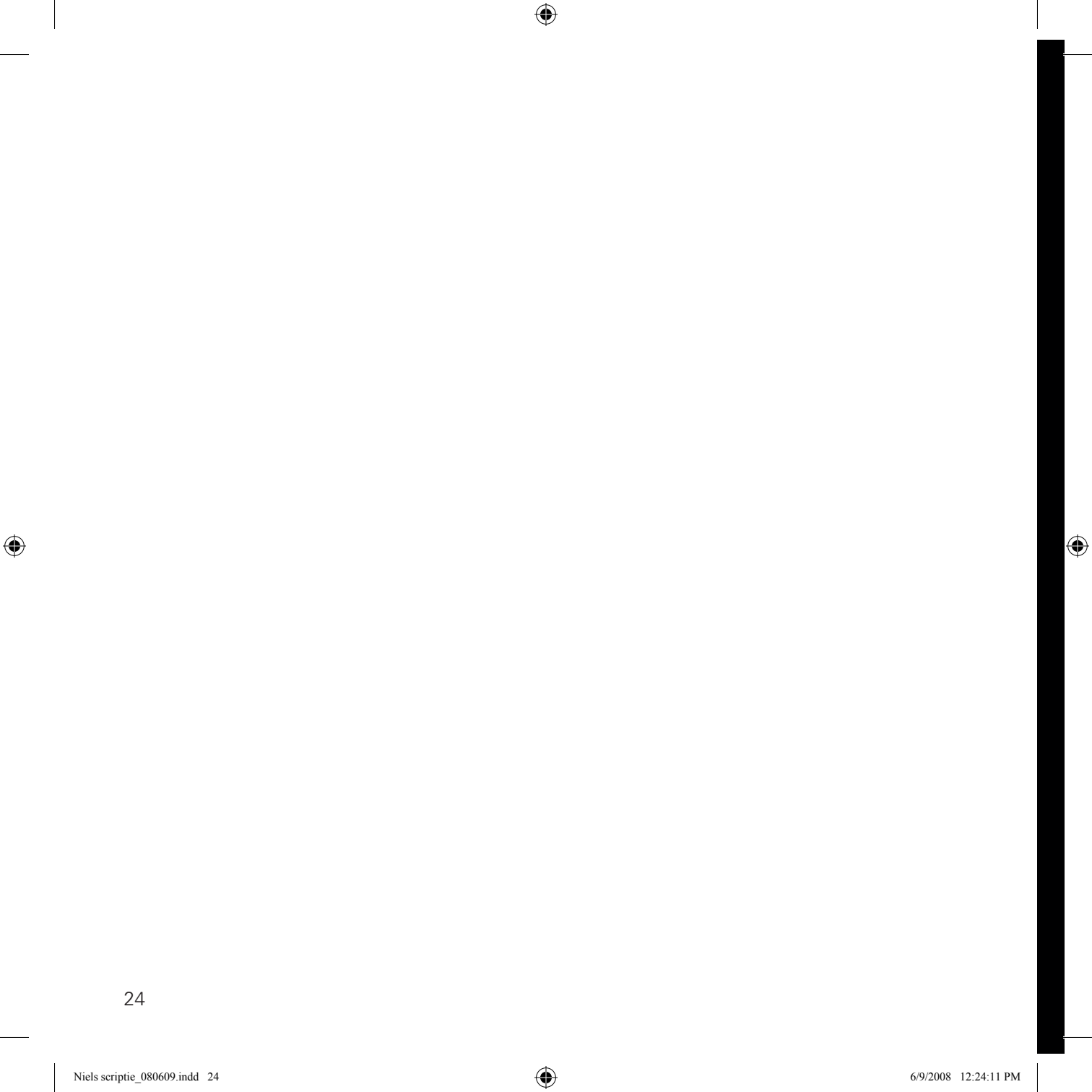
Er zijn verschillende mogelijkheden dit patroon toe te passen. Er kan voor gekozen worden dit patroon zelf te schrijven of er wordt gekozen voor een bestaand framework dat dit patroon toepast.

3.4.3 DATABASE CONNECTIES

De connectie tussen de databases en de applicaties gaan niet vanzelf. Welke mogelijkheden er zijn om deze verbinding mogelijk te maken zullen worden onderzocht.

3.4.4 FLASH COMPONENTEN

Er moet een manier worden gevonden waarop men componenten kan maken die door de vormgever gebruikt kunnen worden om de campagne site te kunnen maken. Ook moeten deze componenten te skinnen zijn, zodat de vormgever de componenten naar wens kan aanpassen.



HOOFDSTUK 4.

Onderzoek

Om tot een eindproduct te komen zullen er verschillende activiteiten moeten worden uitgevoerd. In dit hoofdstuk wordt beschreven welke stappen er doorlopen moeten worden om het eindproduct te realiseren en welke producten deze tussenstappen gaat opleveren

4.1. ACTIVITEITEN

Het opstellen van onderzoek over microcampagnes om na te gaan wat ze precies in houden, wat je er mee kunt bereiken en welke informatie ze vragen.

Het onderzoeken van mogelijkheden om een microcampagne te bouwen via Flash of Flex om na te gaan wat de beste en effectiefste manier is om dit te realiseren

Een usabilityonderzoek uitvoeren met de strekking: welke gegevens presenteert je en hoe kan een dergelijke applicatie het beste worden opgebouwd om vooral gebruiksvriendelijk en intuïtief in gebruik te zijn en tenslotte vertalen naar wireframes.

Een functioneel ontwerp realiseren met behulp van de gewonnen informatie uit voorgaande onderzoeken. Hiermee kan er samen met ontwikkelaars begonnen worden aan de bouw van de applicatie. Ik zal ook het grafische ontwerp en de implementatie hiervan voor mijn rekening nemen.

4.2. OPTE LEVEREN PRODUCTEN

- **Document: Onderzoek microcampagnes**
- **Document: Onderzoek van het maken van componenten**
Dit document is een set van resources hoe je het beste componenten kunt opzetten binnen Flash
- **Document: Structuur van de toolkit (UML)**
Dit document beschrijft de volledige structuur van de toolkit. Deze structuur zal de blauwdruk zijn door het hele project

- **Wireframes van userinterfaces.**

Lijntekeningen van de opbouw van de userinterfaces. Dit betreft de admin- en de client interface. De wireframes zullen later uitgewerkt worden in Flex.

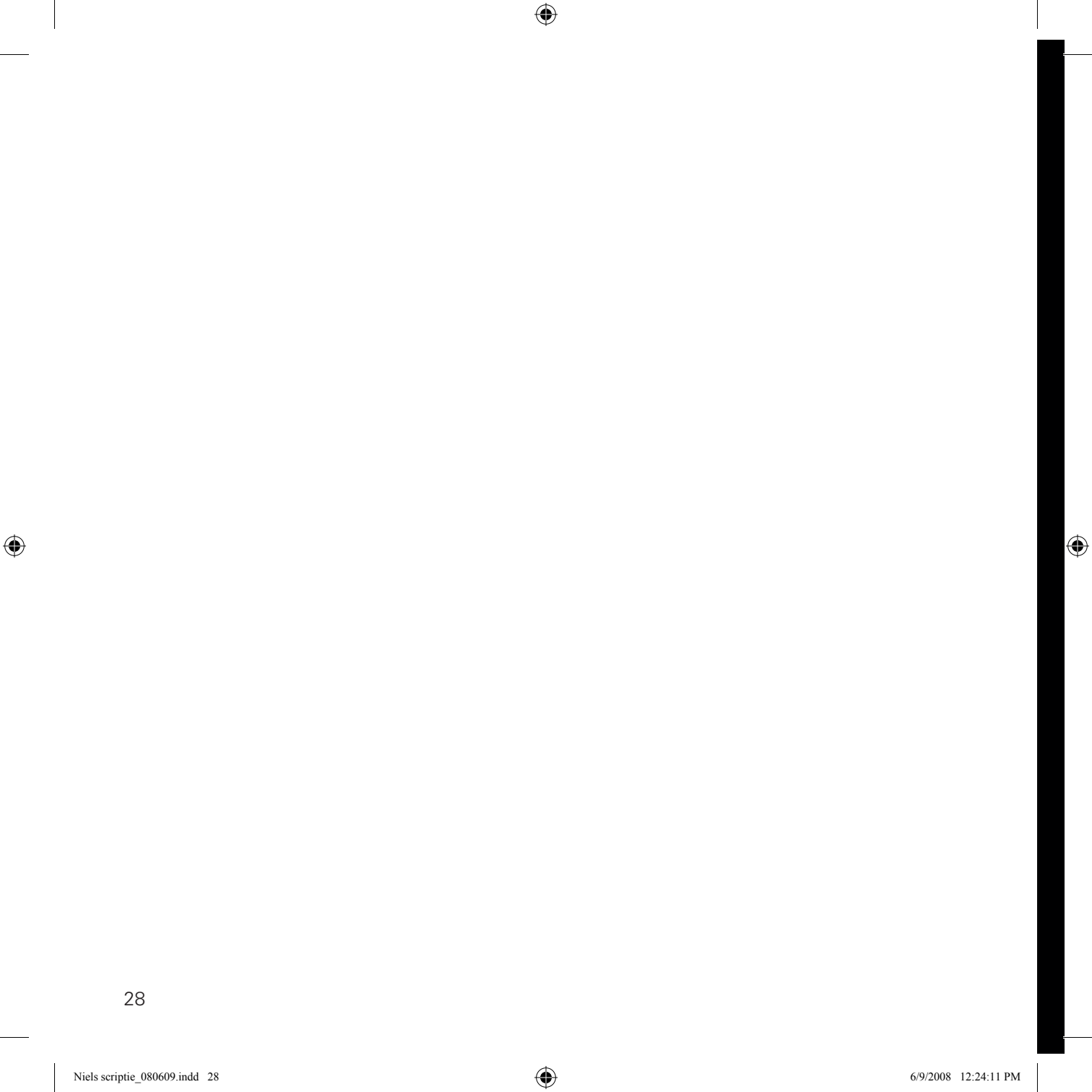
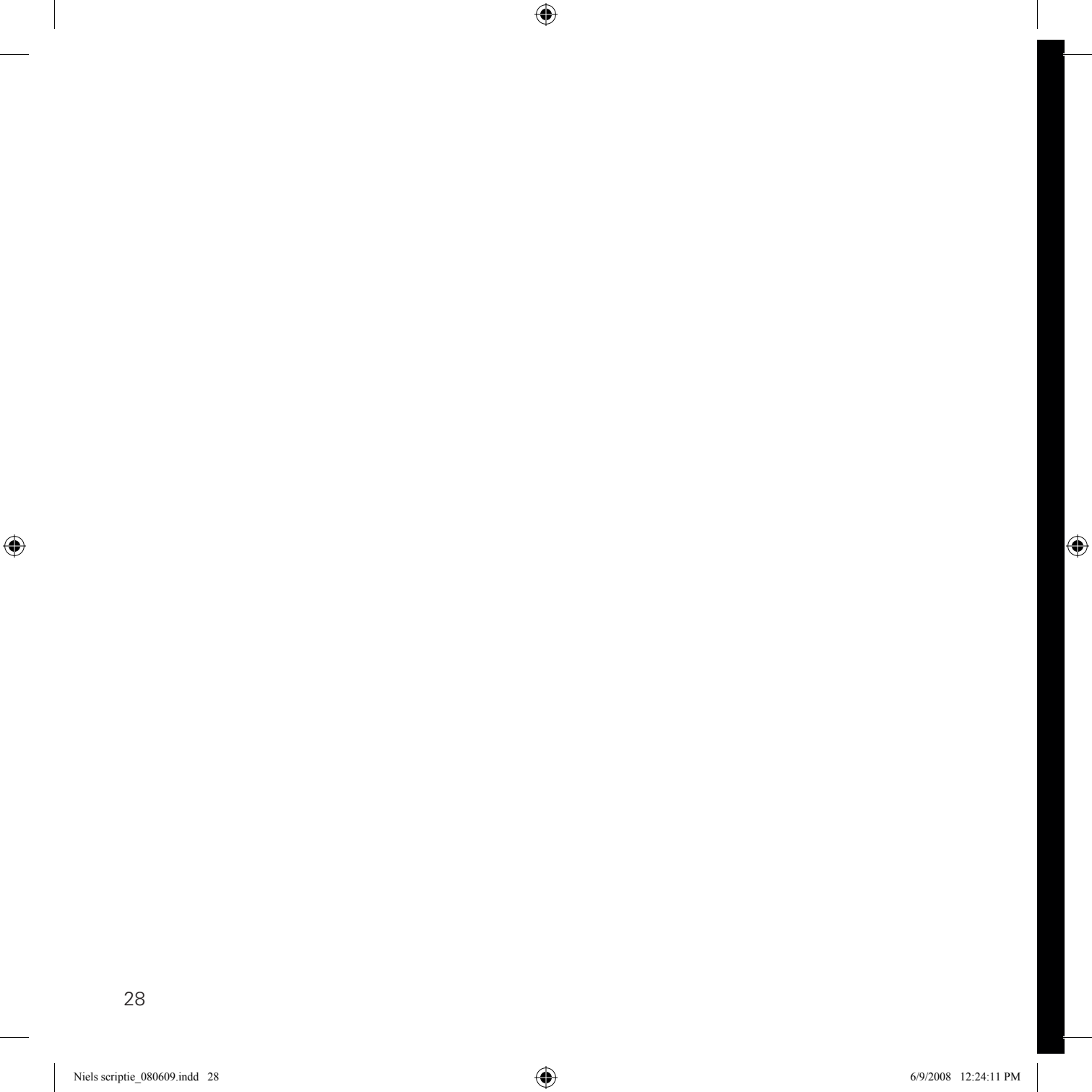
- **Database ontwerp**

Schema van de database. In dit schema zal alles te zien over informatie er opgeslagen gaat worden en welke relaties de tabellen met elkaar zullen hebben.

- **Document: ontwerpeisen voor aanvullende componenten**

Hierin staan de eisen waaraan aanvullende componenten moeten voldoen. Dit zorgt ervoor dat personen die geen kennis hebben van het project eenvoudig extensies kunnen maken om de functionaliteit van het eindproduct uit te bereiden.

- **Documentatie**



HOOFDSTUK 5.

Microcampagnes

Wat zijn microcampagnes eigenlijk en met welke redenen zijn ze interessant voor de adverteerders en bezoekers? Waarom komen bezoekers naar de site toe? Het doel is dat er een beter inzicht komt over microsites en -campagnes die we kunnen toepassen in de toolkit die gemaakt zal gaan worden.

5.1. REDACTIONEEL EN COMMERCIEEL

Een mini- of microsite is een kleine website waarop bedrijven specifieke informatie geven over een of meerdere producten of diensten. Vaak zijn deze kleine websites aangekleed met verschillend multimedia content, zoals bijvoorbeeld animaties, korte videoboodschappen of schema's die de prestaties van het product weergeven. Microsites worden vaak buiten de officiële site van het bedrijf geplaatst en hebben vaak ook hun eigen URL.. Er zijn 3 mogelijke vormen waarin microsites voor komen.

Redactioneel

Deze vorm bestaat vaak uit een website van een of meerdere pagina's, waarin bijvoorbeeld een vakantie of soort gelijk onderwerp besproken word. Deze pagina's geven in dit geval net iets meer informatie dan algemene paginas van het bedrijf of community.

Redactioneel – commercieel

In deze vorm probeert een bedrijf met redactionele informatie mensen naar de site te trekken. Kijk nou naar een bedrijf dat kerstartikelen verkoopt. Dit bedrijf maakt dan een website over kerst en bijvoorbeeld de historie er van. Het doel is natuurlijk meer artikelen te verkopen, maar ook is dit erg goed voor de trefwoorden binnen de zoekmachineoptimalisatie. Op deze manier kunnen ze meerdere bezoekers genereren voor de hoofdwebsite

Commercieel

In een commercieel perspectief geeft een bedrijf via een microsite diepere informatie over een product of dienst. Hierin worden bijvoorbeeld nieuwe technieken beschreven zoals bijvoorbeeld duurzame oplossingen voor een auto.

5.2. VIRAL MARKETING

De ouderwetse mond-tot-mond reclame heeft een nieuwe naam gekregen: Viral Marketing. Eigenlijk is het precies zoals vroeger alleen gaat het door de automatisering nu ongelooflijk veel sneller. Viral marketing is een strategie die prima past in het concept van microcampagnes.

Viral Marketing werkt op dezelfde wijze als een virus. De gebruiker communiceert de boodschap passief. Iedereen die de boodschap ontvangt kan weer zender worden van de boodschap.. Elk voordeel heeft zijn nadeel en andersom. Zo ook bij de viral marketing. Onderstaand een opsomming van deze voor en nadelen.

Voordelen

- Vergeleken met traditionele media kent virale marketing doorgaans een kosten-voordeel. Op een goedkope manier kun je snel veel mensen bereiken. Zo kun je bijvoorbeeld bij een e-mail automatisch onder ieder uitgaande e-mail een standaardtekst meesturen.
- Het mechanisme sluit goed aan bij specifieke doelgroepen zoals jongeren, die via de traditionele media moeilijker te bereiken zijn.
- De snelheid van de campagne wordt doorgaans als groot voordeel gezien. Traditionele media zoals televisie hebben een relatief lange aanlooperperiode nodig voordat consumenten de reclame voldoende herkennen. De boodschap verspreid zich niet alleen snel, je kunt de boodschap ook snel aanpassen. In het geval van drukwerk moet je bij wijzigende content het hele drukproces opnieuw opstarten.

Nadelen

- Het ontwerpen van virale campagnes is lastig en vereist specifieke vaardigheden. Wat in het verleden werkte, doet dat doorgaans in het heden al niet meer.
- Vaak is het bij een echt succesvolle campagne niet direct duidelijk dat het om een virale campagne gaat. “Ontdekking” van de commerciële boodschap kan in het ergste geval het succes van de campagne vroegtijdig beëindigen.
- De virale campagne onttrekt zich, eenmaal gelanceerd vaak aan de invloed van de onderneming. Waar bij een reclamefilm op televisie binnen enkele uren een campagne gestaakt kan worden, is dit bij virale campagnes meestal niet meer het geval.
- Sommige mensen stellen het ontvangen van filmpjes en andere uitingen niet op prijs. Het is mogelijk dat dit negatief uitstraalt op het merk of het imago van de fabrikant.

5.3. CONSUMENT EN ADVERTEERDER

Virale marketing en microcampagnes zijn interessant voor zowel de consument als de adverteerder. Beide groepen hebben interesses, maar welke zijn deze?

Waarom zijn microcampagnes interessant voor de consument?

- Ze bieden entertainment
- Je kunt er customized producten bestellen
- Je kunt er gedetailleerde informatie over een product of dienst op vinden
- Je kunt er prijzen winnen

Waarom zijn microcampagnes interessant voor de adverteerder?

- Je kunt informatie over personen of groepen winnen
- Je kunt er potentials mee bereiken
- Je kunt je naamsbekendheid vergroten
- Je kunt feedback winnen over eigen producten of diensten
- Je kunt er informatie over concurrenten winnen

Welke manieren zijn er

- Games
- Virals
- Maken van persoonlijke producten
- Friendcodes en Gifts
- Personen informatie over vrienden laten vertellen
- Polls
- Quiz waar je informatie uithaalt

Wat willen bedrijven te weten komen

- Persoonlijke informatie
- Doelgroep informatie
- Feedback van producten en/of service
- Nieuwe wensen betreft producten en/of diensten
- Geografische informatie

5.4. CONCLUSIE

Het maken van een goede microcampagne vraagt behoorlijk wat aandacht en na lanceren van de campagne is het niet goed te zien wat het resultaat nu daadwerkelijk is geweest. Door de informatie op te slaan die wordt ingegeven in de microsite wordt het interessant voor de adverteerder hier meer mee te gaan doen.

HOOFDSTUK 6.

Algemene webapplicatie architectuur

Om het makkelijk te houden is een rich internet application (RIA) een webapplicatie waarvan de client een interactieve interface is die het gevoel geeft van een desktop applicatie. Een client is programmaatje dat op de eigen computer draait en contact maakt met een server om gegevens op te halen of te verzenden. Een goede webapplicatie vraagt om een goede structuur. In dit hoofdstuk wordt behandeld wat hier de beste oplossing voor is.

6.1. MODEL-VIEW-CONTROLLER

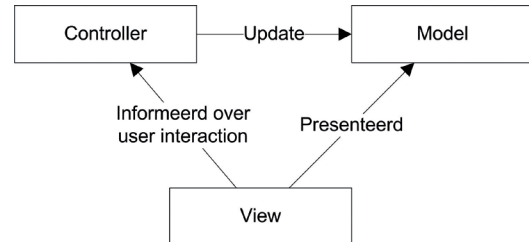
6.1.1 WAT

Het maken van een webapplicatie kan een complexe klus zijn als er geen rekening gehouden wordt met de structuur. Om het proces van het maken van een webapplicatie overzichtelijk te houden is het verstandig de verschillende taken binnen de webapplicatie te verdelen. Veelal worden dusdanige applicaties verdeeld in 3 lagen met elk haar eigen verantwoordelijkheden. Deze structuur wordt ook wel model-view-controller genoemd. De 3 lagen zijn datamodel (model), datapresentatie (view) en applicatielogica (controller).

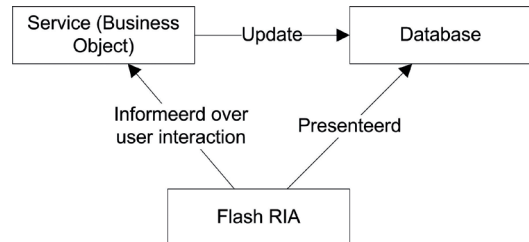
6.1.2 HOE

Naast het verdelen van de verschillende verantwoordelijkheden ligt er een complexe wisselwerking van berichten tussen deze lagen. De view is hetgeen wat de gebruiker ziet, Het komt er op neer dat elke view die aangepast moet worden bij veranderingen in het model, dit aan het model meedeelt door zich te registreren bij het model. Als een gebruiker nu een handeling verricht in de view (bijv. Een muisklik op een knop) schakelt deze een controller aan. Deze controller laat het model weten wat er moet gebeuren en verandert de data in het model. Als de data van het model verandert zijn laat deze het weten door een update-bericht te verzenden, ook wel events genoemd. De views die zich op dit bericht hebben geregistreerd zien dat het model geüpdate is en vernieuwen zich met de nieuwe gegevens uit het model.

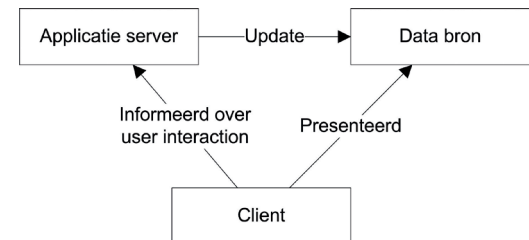
Het model slaat alle data op die de applicatie gebruikt. De view presenteert de data van het model visueel en ontvangt user input van de gebruiker, zoals input van muis en toetsenbord. De controller verwerkt de input die het krijgt van de view en bewerkt de data in het model.



In de context van een webapplicatie is het model de bron, de controller is de applicatie op de server en de view is de client.



Om te dit terug te koppelen naar Flash houdt dit het volgende in. De controller kan de vorm nemen van een dienst op de server die een database leest en bijwerkt. Uit eindelijk presenteert de client (die je maakt in Flash) de data.



6.1.3

WAAROM

Het verdelen van je applicatie in 3 lagen levert verschillende voordelen op. Het scheiden van deze verantwoordelijkheden zorgt er namelijk voor dat je overzicht houdt binnen je code. Ook is de code beter herbruikbaar, omdat het niet direct aan elkaar gekoppeld zit, maar geïnformeerd wordt door middel van events. In traditionele manieren van programmeren, waar alles in een script onder elkaar staat, is dit zeker niet het geval. Ook is het mogelijk verschillende views te maken om data te presenteren van het zelfde model. Dit houdt in dat alleen de views gemaakt worden en het model en de controller blijven gewoon intact.

6.2. DATA ACCESS OBJECT

6.2.1

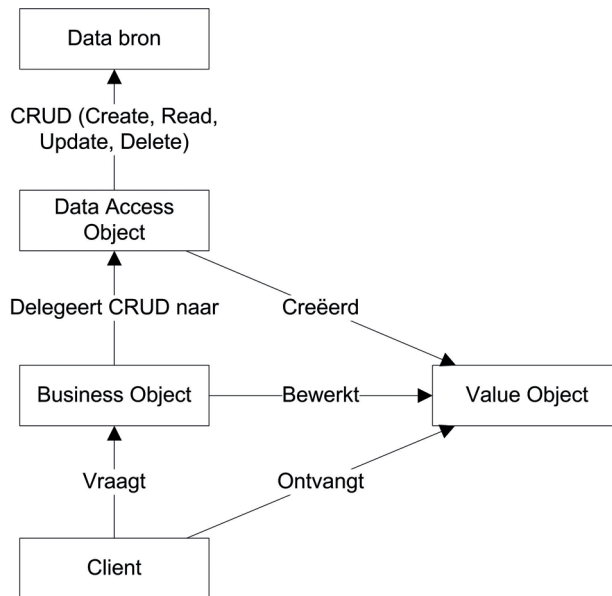
WAT

Vaak wordt er binnen een webapplicatie gebruik gemaakt van een database. Hiervoor is het Data Access Object (DAO) patroon ontworpen. Een DAO is een object dat een abstracte interface levert voor een database. Deze DOA's voeren zo genoemde CRUD operaties uit op de database. CRUD staat voor create, read, update en delete. Deze DOA's zetten hun resultaten om naar value objecten (VO). Een VO representeert een rij (ook wel een record genoemd) uit een database tabel.

6.2.2

HOE

De architectuur op de server kan verschillen van een eenvoudig script tot een zeer complexe architectuur. Door een dienst (ook wel service genoemd) te maken die de client alleen de functies laat gebruiken die het nodig heeft. Daardoor de rest van de structuur verbergt, kun je de communicatie van uit de client met de database vereenvoudigen. Dit principe wordt ook wel Application Programming Interface genoemd.



Een client krijgt dus verschillende functies tot zijn beschikking van de service die hij kan aanroepen. Een dienst op de server delegeert dan naar verschillende DAO's die op hun beurt dan een bepaalde verantwoordelijkheid op zich nemen voor het bewerken of verwerken van gegevens in de database. De dienst op de server ontvangt dan één of meerdere VO's en stuurt deze terug naar de client.

6.2.3

VERBERGEN VAN STRUCTUUR

Door gebruik te maken van een goede service wordt de achterliggende structuur verborgen gehouden voor de client. De client hoeft dus alleen maar API van de service te weten en verder is het niet belangrijk hoe deze aan zijn informatie komt. Dit heeft als groot voordeel dat de structuur van de service kan veranderen en de client gewoon gebruik blijft maken van de zelfde functies.

6.3. CONCLUSIE

Het opzetten van webapplicatie is tijdrovend en vraagt enig denk werk. Het is daarom niet aan te bevelen dit soort patronen toe te passen op kleine webapplicaties. In kleine applicaties is het vaak prima mogelijk om overzicht te bewaren zonder deze patronen. Voor middelgrote en grote applicaties is het wel erg verstandig om deze patronen te gebruiken. Ze zorgen dat je webapplicatie overzichtelijk blijft door de taken te verdelen en doordat je de code kan hergebruiken door er andere views voor te maken.

HOOFDSTUK 7.

Communicatie

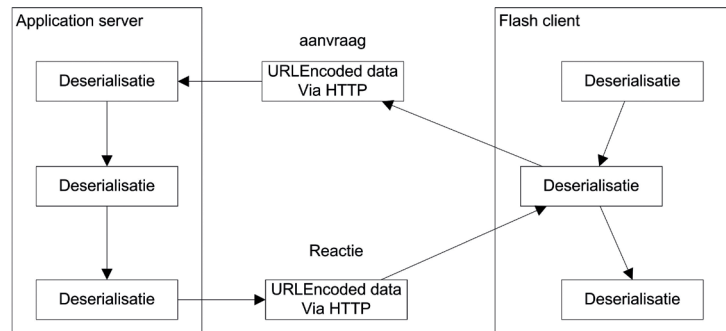
Het vorige hoofdstuk gaf aan dat de Flash-client verbinding maakt met de API van de service. Maar er is één probleem. Beiden zijn gemaakt in een andere taal en kunnen niet direct met elkaar communiceren. Om met elkaar te kunnen communiceren maken zij gebruik van een protocol. Een protocol is een afspraak hoe ze met elkaar moeten communiceren. Flash heeft verschillende manieren om te communiceren met een server en deze zullen besproken worden in het komende hoofdstuk.

7.1. RPC EN REALTIME COMMUNICATIONS

Flash beschikt over twee technologieën om verbinding te maken met een server. Remote procedure calls (RPC) en Realtime Communications. Realtime Communications wil zeggen dat er een direct contact met de server blijft bestaan. Dit wordt gebruikt voor multiplayer games en webcam verbindingen. Voor dit onderzoek maken we gebruik van RPC en wordt er verder niet meer ingegaan op Realtime Communications. RPC verbinding is een asynchrone verbinding met de server, dit wil zeggen, er wordt een aanvraag gedaan en na een tijd als de server klaar is stuurt hij gegevens terug. Flash beschikt over drie verschillende manieren om een RPC tot stand te brengen.

7.1.1 HTTP-SERVICES

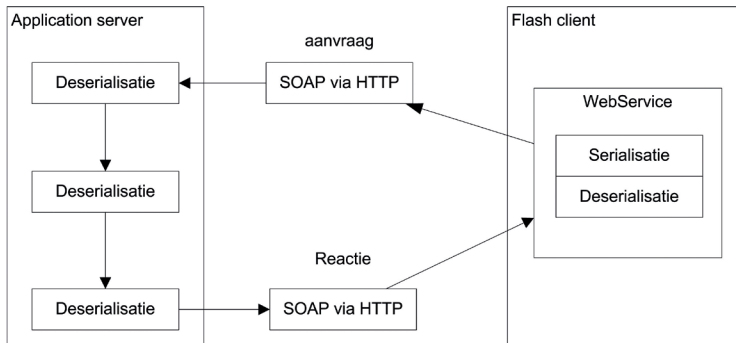
Een HTTP-Service maakt een aanvraag via het HTTP protocol. Meestal is dit POST of GET. Deze aanvragen kunnen verschillende content types bevatten. Zo kunnen het bijvoorbeeld simpele variabelen zijn, maar ook kunnen het complexe gegevensstructuren zijn zoals XML.



Dikwijls worden HTTPServices gebruikt voor simpele data en wordt er verbinding gemaakt met een eenvoudig script op de server. De waarde van aanvraag die je wilt doen wordt aangegeven in de link van de dienst.

De data die over en weer verzonden wordt en zijn gewoon kale teksten. Om dit bruikbaar te maken in de client en bij de dienst op de server moeten de data steeds vertaald worden naar de programmeertaal waarin ze gebruikt gaan worden.

7.1.2 WEBSERVICES



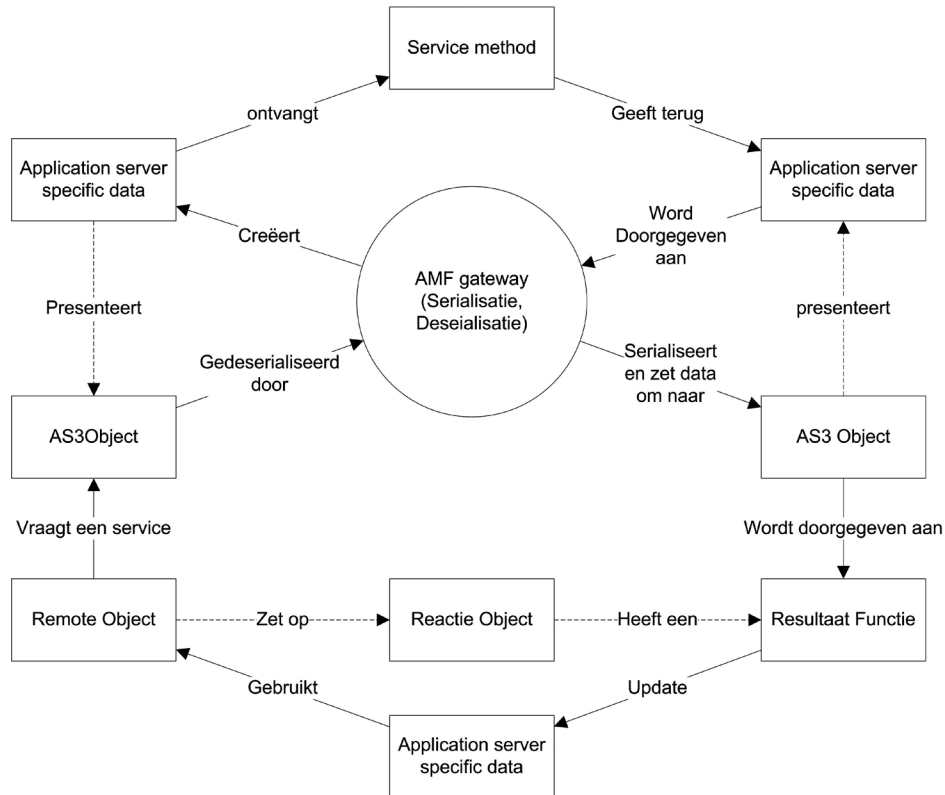
Webservices zijn hetzelfde als HTTP-Services. Alleen in dit geval worden er dan SOAP berichten verzonden en ontvangen over het HTTP protocol. De meeste Actionscript variabelen worden

binnen Flash automatisch omgezet naar XML /SOAP data. Dit proces wordt afgehandeld binnen Flash door de Webservice class in Flash.

7.1.3 REMOTE OBJECTS

De Remote object class in Flash is verantwoordelijk voor het verzenden en ontvangen van Actionscript Message Format (AMF). Dit wordt ook wel remotng genoemd. AMF is geen protocol, maar het is binair Actionscript. In dit geval wordt AMF verzonden

over het HTTP protocol, maar het zou net zo goed verzonden kunnen worden over realtime communication protocols. Uitwisselen van AMF is veel sneller dan welke andere manier van communicatie.



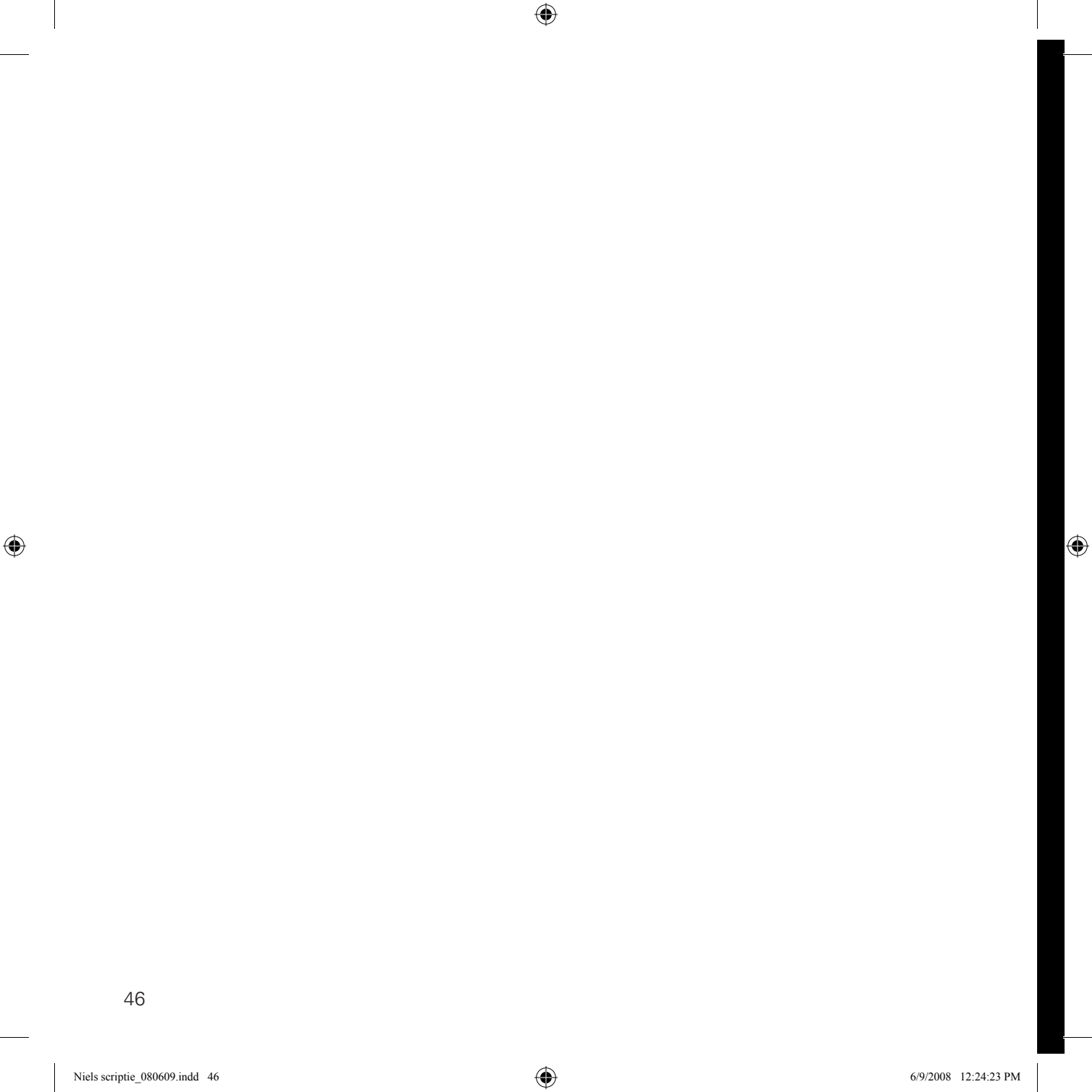
Behalve dat de Flash applicatie “Strongly typed” ontvangt, wat inhoud dat Flash binnen krijgt wat voor soort variabele het zijn, gebeurt het serialisatie/deserialisatie proces alleen op de server. Hier is een AMF gateway voornodig (ook wel remoting gateway genoemd).

Met remoting kan je zowel eenvoudige data, als complexe data versturen (net als een webservice), maar het is ok mogelijk handmatig een type toe te wijzen aan een variabele. Zodoende is het dus mogelijk een VO te verzenden dat in de client en op de server exact hetzelfde is.

Hieronder is een voorbeeld hoe dit er uit zou kunnen zien.

7.2. CONCLUSIE

Het gebruik van HTTPServices of webservices is goed om te gebruiken als het om kleinere projecten gaat. Je hoeft er namelijk geen gateway voor op te zetten om te communiceren. Voor grote projecten is het aan te raden om remoting te gebruiken. Het communiceren via remoting zorgt er namelijk voor dat de VO in je client en op je server hetzelfde zijn. Dat betekent dat altijd de juiste objecten over en weer worden gestuurd. Ook biedt AMF een groot voordeel omdat het binair is. Een binair bestand is vele malen kleiner dan een tekstbestand, Omdat het kleiner is kost het dus minder tijd om te versturen.



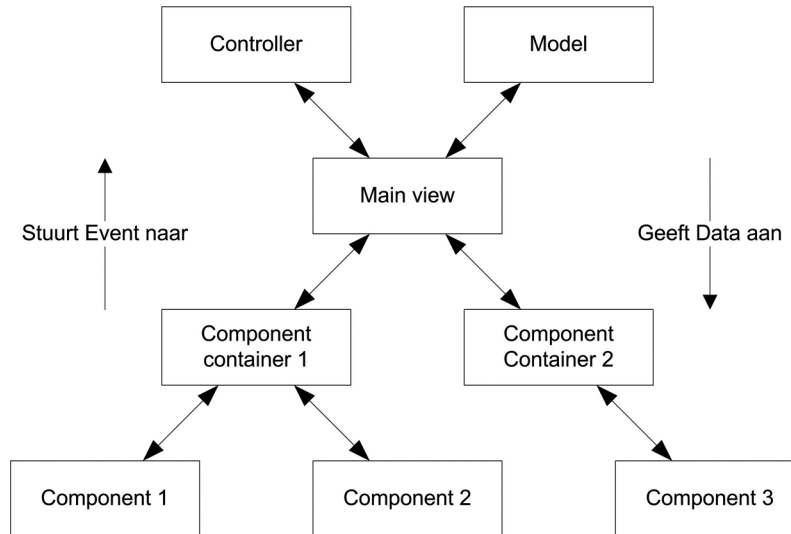
HOOFDSTUK 8.

Flash structuur

Zoals in hoofdstuk 6 verteld werd kan de Flash-client gezien worden als de view in het MVC patroon. Het beste is dat de Flash applicatie zo afhankelijk mogelijk is van services die de server aanbied. De enigste afhankelijkheid die de applicatie moet hebben is de Service API en het moet weten welke VO's hij kan versturen en ontvangen. Ook voor de Flash applicatie is het mogelijk om de architectuur op te delen in 3 lagen. Hoe ziet de structuur er van de Flash-client uit als het gebruik maar van MCV en een Service API?

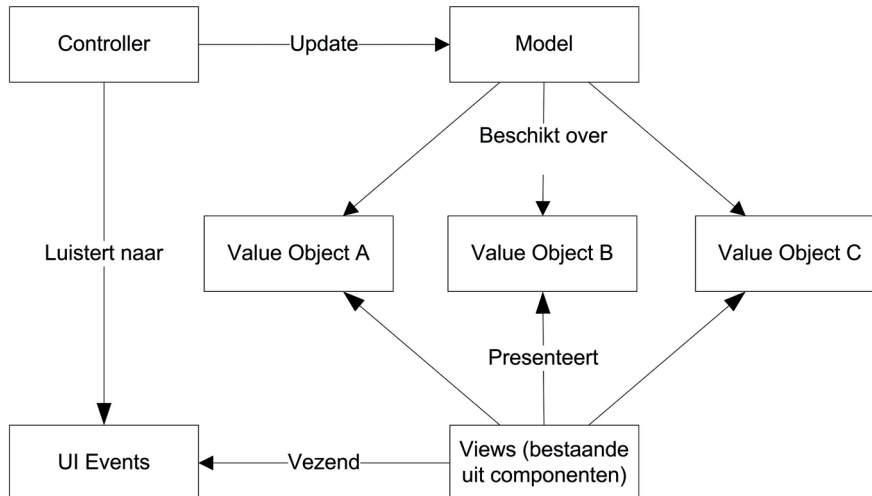
8.1. DE VIEW-LAAG

De view-laag in een Flash-applicatie die bestaat uit een hiërarchie van componenten die data weergeeft. De root van de applicatie representeert de top van de set componenten. De communicatie binnen de view-laag wordt bereikt door events binnen Flash. Dit zorgt voor communicatie zonder dat er afhankelijkheden ontstaan. Het doel is dus ook dat elk component dat data weergeeft afzonderlijk van elkaar te maken. Dit zorgt er voor dat een component opnieuw te gebruiken is en dit scheelt natuurlijk tijd.



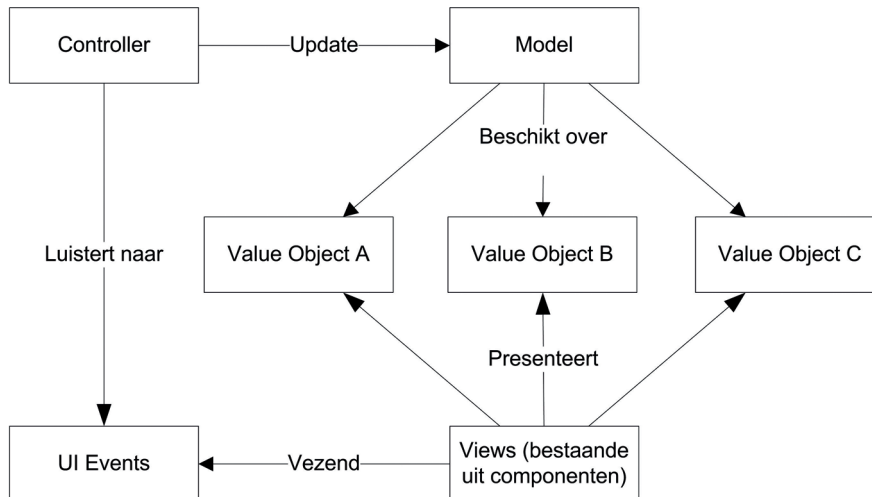
8.2. DE MODEL-LAAG

Deze laag is bedoeld om data op te slaan. Het slaat eigenlijk de staat van de applicatie op. Het heeft geen rekenkundige logica, alleen logica om op zoek te gaan naar bepaalde data. Simpel gezegd is het gewoon een collectie van variabelen die gedeclareerd zijn in de root van de applicatie. De model-laag bestaat voornamelijk uit VO's die op zijn tijd weergegeven worden in de view. Deze VO's zijn representaties van records in de database.



8.3. DE CONTROLLER-LAAG

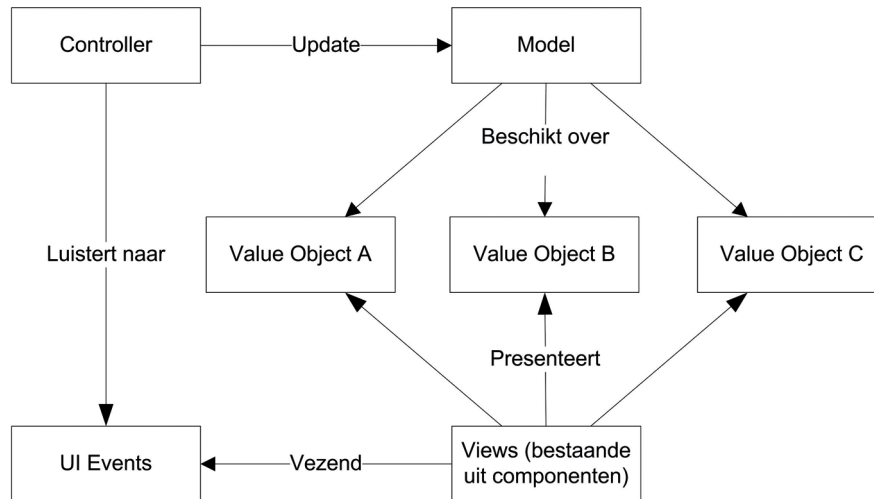
De controller is verantwoordelijk voor de logica en interactie tussen de model- en de view-laag. Als bijvoorbeeld de gebruiker op een knop druk in de view-laag, is de controller degene die hier op reageert. Meestal wordt dan de model-laag geüpdate.



8.4. DE BUSINESS LAAG

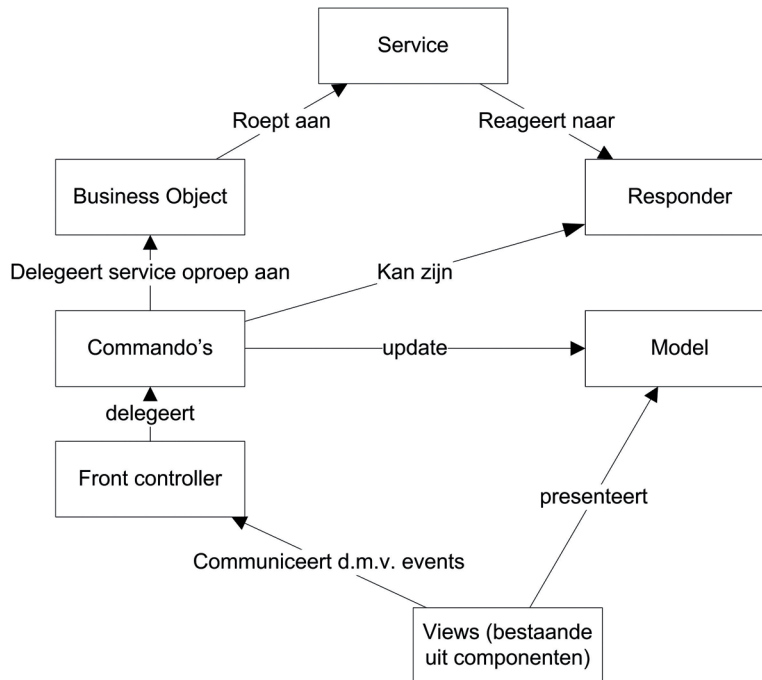
Een andere belangrijke taak voor de controller is het updaten van de model-laag in de web applicatie, zoals bijvoorbeeld de database. Een controller kan dan gebruik maken van een soort van RPC-component die verbinding maakt met de service op de server. Het is goed voor te stellen dat de service kan veranderen gedurende de ontwikkeling.

Om de Flash-client goed onderhoudbaar te houden is het verstandig om een business laag toe te voegen. Het doel van deze laag is om de API van de service te vertegenwoordigen.



8.5. VOLLEDIGE STRUCTUUR

Er zijn nog veel meer patronen die er bij kunnen komen, maar het bovenstaande is de globale manier van werken. Deze extra patronen verdelen de verschillende lagen in extra taken. Hieronder een schema van een voorbeeld waarin de structuur die verteld is enorm is uitgebreid.



8.6. FRAMEWORKS

Er zijn verschillende mogelijkheden om het MVC patroon toe te passen op een applicatie. Er kan gekozen worden om het patroon zelf te ontwikkelen, maar het is ook mogelijk om te kiezen voor een MVC framework. Op internet zijn verschillende MVC frameworks gratis beschikbaar om een Flash applicatie mee te maken. Op het moment zijn twee frameworks zeer populair en dusdanig volwassen om te gebruiken. Deze zijn Cairngorm van Adobe en PureMVC

Een framework is een verzameling software componenten dat gebruikt kan worden voor het programmeren van een applicatie. Een framework bestaat uit een verzameling classes en afspraken hoe deze met elkaar samenwerken. Het is eigenlijk een herbruikbaar ontwerp voor een applicatie. Ze bieden structuur en bepaalde werkwijzen om snel en zorgvuldig applicaties te maken.

Het maken van een applicatie zonder framework is een verleidelijk idee, maar is het realistisch? In de voorgaande hoofdstukken stond beschreven hoe de structuur van een dusdanige applicatie er uit zou zien. Het maken van een applicatie zonder framework vereist behoorlijk wat ervaring om het goed op te bouwen, daarom wordt er gekeken naar de volgende twee frameworks, Cairngorm van Adobe en PureMVC.

8.6.1 CAIRNGORM VS PUREMVC

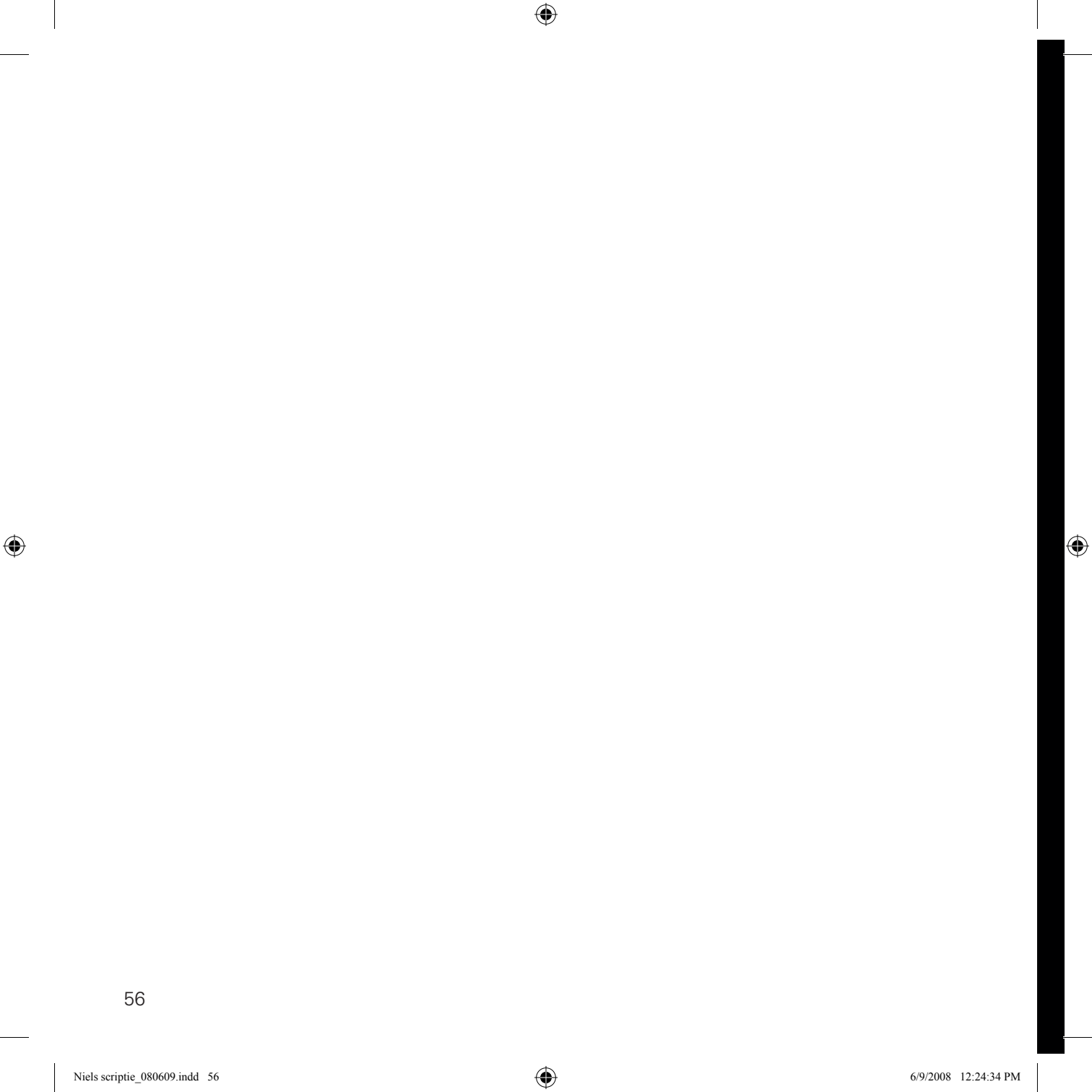
Cairngorm is waarschijnlijk het meeste gebruikte MVC framework voor Flex. PureMVC is geen Flex framework, maar een Actionscript 3 framework. Op het moment wordt er zelfs aangewerkt om PureMVC te vertalen naar verschillende andere talen zoals JAVA en C++ en nog veel meer. Net als Cairngorm gebruikt

PureMVC een controller / command techniek. Dit wil zeggen dat de berichtgeving in de applicatie tussen alle objecten rond gaan door middel van notifications, niet te verwarren met events. Bij events is het luisterend object verantwoordelijk voor het opvangen van het bericht. Bij notifications is het object dat het verstuurd verantwoordelijk. Cairngorm is bij de meeste Flex ontwikklers het meest bekend, dit komt vooral door de ondersteuning van Adobe. Hier onder een kleine vergelijking tussen beide frameworks.

Cairngorm	PureMVC
Flex framework	Actionscript 3 framework
Model	Model
View	View
Controller	Controller
Command	Command
CairngormEvent	Facade
Delegate	Proxy
Service	Mediator
	Observer
	Notification

8.7. CONCLUSIE

Vrijwel alle Flash schema's die je kan bedenken passen in dit schema. Wanneer er begonnen wordt aan een nieuw project (vooral wanneer een project zeer complex kan worden) bied dit een zeer goede richtlijn voor je Flash-client. Als deze richtlijnen eenmaal bij iemand bekend zijn kan hij gemakkelijk werk van andere lezen en zij het werk van hem. Dit betekend tijdwinst en meer productiviteit. Het is niet belangrijk om alle patronen uit deze werkwijze toe te passen in de applicatie. Vaak is maar een klein gedeelte bruikbaar in de ontwikkeling van de applicatie.



HOOFDSTUK 9.

Flash ui componenten

Een Userinterface Component is een grafisch object in Flash. Een Flash presenteert data. Deze componenten beschikken meestal functionaliteit om gegevens te laten zien of om gegevens te verzenden. Componenten worden gemaakt met als doel dat ze vaker gebruikt kunnen worden. Flash beschikt al over een aantal componenten, zoals bijvoorbeeld Button, TextInput, Scrollbar. Dit hoofdstuk verteld waar opgelet moet worden om unieke herbruikbare componenten te maken.

9.1. VERSCHILLEN FLASH EN FLEX

Bij het maken van Flash-componenten is het belangrijk dat verantwoordelijkheden van het component duidelijk zijn. Een goede planning is daarom erg belangrijk. Door van te voren goed te bepalen welke verantwoordelijkheden een component heeft kun je bepalen op welke manier het gemaakt moet worden en welke API deze laat zien voor de buitenwereld. Het afhandelen van data wordt intern geregeld. Dit zorgt er dan ook voor dat het component nooit afhankelijk is van de rest van de applicatie en dus steeds opnieuw gebruikt kan worden.

Er zijn verschillende dingen waarmee rekening gehouden moet worden voor de opbouw van een component. Als eerste moet er een keuze gemaakt worden in welke omgeving dit component gebruikt gaat worden. Dit kan Flash, Flex of in beide. Het probleem is dat de Software Development Kit (SDK is een verzameling hulpmiddelen waarmee programma's gemaakt worden) van Flex uitgebreider is dan die van Flash. Dit betekent dat componenten die voor Flex gemaakt worden niet altijd werken in Flash en andersom.

Het maken van een component voor de SDK's afzonderlijk levert weinig problemen op. Het maken van een component dat geschikt is voor beide is wat lastiger omdat beide SDK's verschillend zijn en omdat beide SDK's anders omgaan met plaatsen van het component op het scherm.

9.2. OVERERVING

Overerving is in OOP een veel gebruikte manier om nieuwe classes te maken. Overerving wil zeggen dat je een nieuw component maakt door de functionaliteit over te nemen van een bestaand component. Neem als voorbeeld een knop. Een knop heeft bijvoorbeeld een label en functies om het label op deze knop aan te passen. Door een nieuw component de functies van deze bestaande knop over te erven, wil dit zeggen dat de functies om het label aan te passen niet meer gemaakt hoeven worden. Overerving is geen vereiste voor het maken van een component. Elke SDK heeft zijn eigen bestaande componenten die gebruikt kunnen worden om van over te erven. Elke bestaande class die bestaat in de SDK kan de functionaliteit van worden overgenomen.

Flash en Flex hebben niet dezelfde SDK. Toch is het mogelijk een component voor beiden te maken. Dit is door middel van overerving van UIMovieClip. Deze class die bestond in Flex is teruggebracht naar Flash. Als van deze class wordt uitgegaan dan werkt het component in beide ontwikkel omgevingen.

9.3. CUSTOM EVENTS

Zoals verteld in hoofdstuk 6 zijn events, berichten waarop andere delen van de applicatie zich kunnen registreren zodat ze op de hoogte gebracht worden wanneer een bepaalde gebeurtenis plaats vindt. Componenten hebben ook events waarop geregistreerd kan worden. Een knop heeft bijvoorbeeld een klik event. Een nieuw component heeft waarschijnlijk ook nieuwe events. Events zijn erg belangrijk en er moet zeker over nagedacht worden tijdens de planning van het component.

9.4. PARAMETERS

Parameters zijn de variabelen van een component. Deze parameters zijn onderdeel van de API van het component. De parameters geven het component bepaalde waarden mee, zoals bijvoorbeeld de tekst in een label op een knop of de kleur van een component.

9.5. METADATA

Meta data zijn bepaalde waarden die je mee kunt geven aan een class, eigenschap of methode. Deze metadata zorgt ervoor bepaalde mogelijkheden nadat het component is gecompileerd en gebruikt gaat worden in de applicatie. Hieronder een paar voorbeelden van belangrijk metadata.

- **Bindable**

Als een eigenschap van een component Bindable is, wil dit zeggen dat alles in applicatie dat gebruik maakt van deze variabele ziet zelf ook update als deze gewijzigd wordt

- **Event**

Event metadata zorgt ervoor dat de events te zien zijn in de codehinting in de applicatie waarin component gebruikt wordt

- **Inspectable**

De inspectable tag zorgt ervoor dat net zoals bij events de parameters van het component te zien zijn bij codehinting van de applicatie. Ook komen deze para-

meters in de Graphical Userinterface van de applicatie waar in je werkt zodat je het component kunt instellen zonder code te gebruiken.

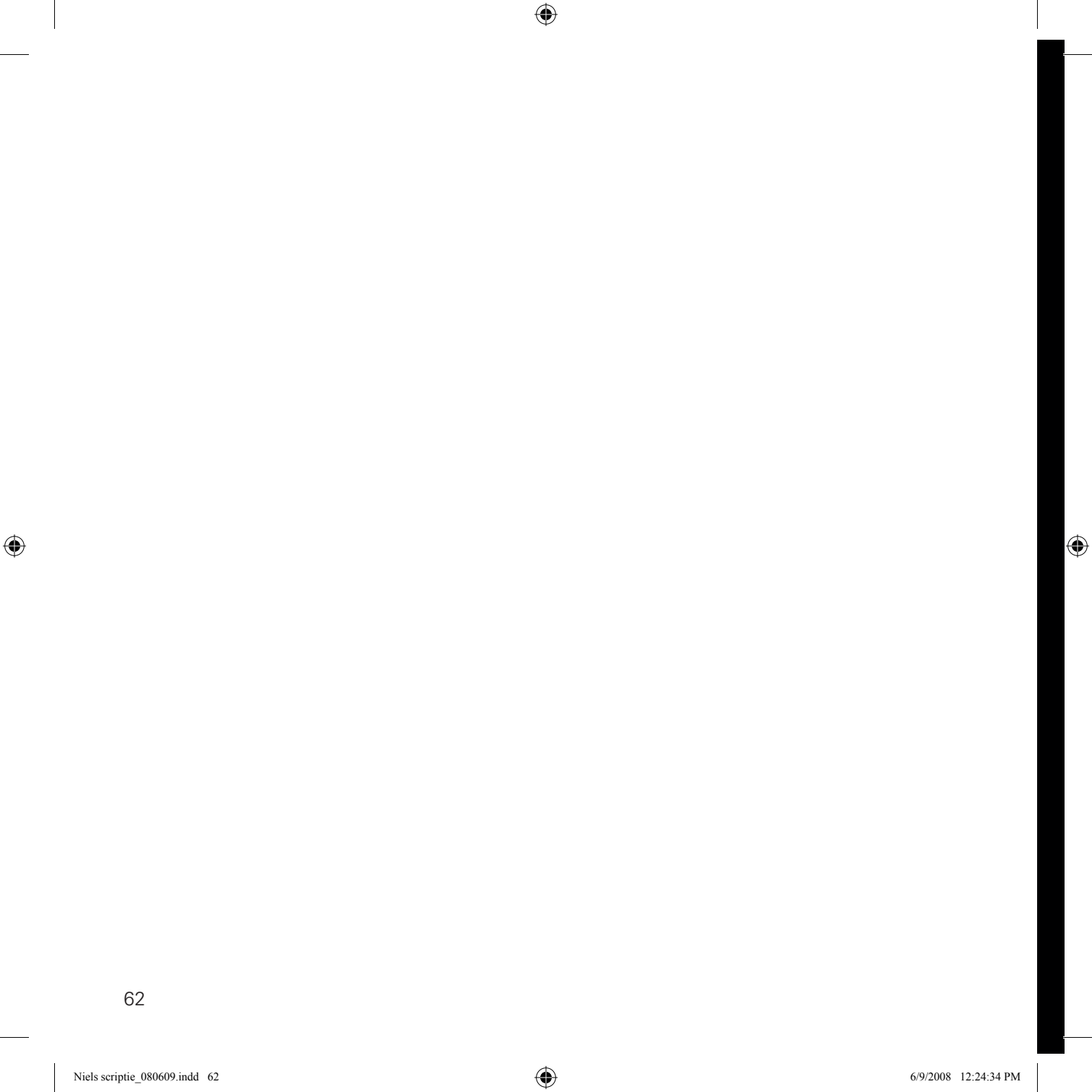
9.6. STYLING

Er zijn 2 manieren om componenten te stylen. De eerste manier is het stylen via Cascading Style Sheets (CSS). CSS is een techniek om de stijl van een website, applicatie en ook componenten in te stellen door middel van coding. Met stylen door middel van CSS is het mogelijk om kleuren, fonts en afmetingen in te stellen in een centraal bestand dat dan consequent wordt doorgevoerd in heel de applicatie.

De tweede mogelijkheid is doormiddel van skinning. Met skinnen bestaat er een soort van template van het component. De visuals in dit component zijn volledig aan te passen naar eigen wens en de stijl die past bij de vormgeving van de rest van de applicatie.

9.7. CONCLUSIE

Het maken van componenten kost wat extra tijd en omdat er verschillende manieren zijn om het op te lossen. Daarom is het belangrijk dat er van te voren duidelijk bepaald wat de verantwoordelijkheden zijn van het component. Het opzetten van een component dat gestijlt kan worden is heel verstandig. Dit component kun dan elke keer weer gebruikt worden in een andere applicatie en heeft het toch zijn eigen stijl



HOOFDSTUK 10.

PCC microcampagne

Viralz4u wordt getest met een eerste campagne. Het is een campagne voor een ander project van Relate4u, Print Competence Center (PCC). PCC is een print-automatiseringsprogramma. PCC zorgt voor een centraal punt waar bedrijven, met name Philips, bestanden kunnen sturen PCC zorgt er dan voor dat deze bestanden verspreid worden over drukkerijen over de hele wereld en vanuit daar geëxporteerd worden naar de juiste locatie. Met betrekking tot PCC wordt een klantendag georganiseerd voor een selecte groep mensen van Philips.

10.1. PCC-KLANTENDAG

Voor deze mensen op deze klantendag, maar ook zeker voor de mensen die niet worden uitgenodigd op deze dag wordt een viral campagne gerealiseerd. De viral zal de voordelen van de PCC-applicatie naar voren brengen. Het moet duidelijk worden dat PCC dé manier is om printopdrachten te verzorgen. Ook zal de viral een bepaalde experience over brengen. Deze viral is gericht aan de marcommers (marketing communicatie medewerkers) van Philips

10.1.1 DOELSTELLINGEN

Met de viral willen we de volgende doelstellingen bereiken:

Voor Relate4u

- De viral moet de voordelen van de PCC-applicatie naar voren moeten brengen. Er zijn 2 manieren waarop dit verteld gaat worden.
 - Algemene voordelen, sneller, beter, goedkoper. Denk aan een snel overzicht waarom PCC de applicatie is voor de betreffende afdeling
 - Technisch uitgebreid verhaal over alle mogelijkheden
- De viral moet bepaalde informatie op gaan leveren. Met deze informatie moet Relate4u, PCC sterker kunnen gaan positioneren, PCC beter gaan vermarkten, PCC verder ontwikkelen.

Voor Philips gebruikers

- De viral moet een bepaalde experience overbrengen. De viral moet leuk, creatief, interactief zijn. Zo leuk dat mensen het doorvertellen.

- De viral moet voor de gebruiker een persoonlijk voordeel opleveren. Denk aan bepaalde kaartjes met een naam erop gedrukt vanuit de memolio omgeving.
- De viral moet de mogelijkheid bieden informatie en ervaringen te verspreiden onder de gebruikers. Het is een streven de afdelingen onder elkaar meer met elkaar te binden.

10.2. CONCEPT

Het concept van deze testcase is nog in ontwikkeling, maar hier is de eerste opzet van de viral.

De marcommers krijgen via e-mail en uitnodiging om naar de PCC-klanten dag te komen. Op deze dag vertelt het PCC-team meer over PCC. De uitnodiging verwijst naar een interne website bij Philips, waarin de marcommers zich kunnen aanmelden voor deze dag. Voordat ze zich kunnen inschrijven krijgen ze een korte quiz. De quiz ziet er als volgt uit. Er zijn 2 animaties uitgebeeld. Links de PCC-gebruiker aan een bureau, rechts een niet PCC gebruiker aan een bureau. Tijdens de animatie worden voordelen van PCC naar voren gebracht in een vraagstelling. De marcommers moeten de vraag beantwoorden om verder te kunnen naar de volgende vraag. Tussen de vragen door speelt de animatie weer verder en stopt bij de volgende vraag. Gedurende de quiz verslechtert de situatie bij de niet PCC-gebruiker en wordt het zelfs chaotisch. De situatie bij de PCC-gebruiker blijft relaxt.

Do you like doing thing twice, tripple,...?

☐ yes

☐ no



HOOFDSTUK 11.

Concept- en product ontwikkeling

Om tot een product te komen moeten er verschillende voorontwerpen gemaakt worden. Dit hoofdstuk verteld de stappen die doorlopen zijn om tot juiste opzet te komen van Viralz4u.

11.1. DE VOORBEREIDING

11.1.1 VEREISTE DATA

De gegevens die binnen de toolkit gebruikt worden zijn natuurlijk heel verschillend. Dit is erg afhankelijk van de campagne. Elke campagne heeft zo zijn eigen gegevens die een bedrijf van haar klanten wil weten. Toch hebben deze campagnes een overeenkomst. Deze campagnes vragen om persoonlijke gegevens.

Persoonlijke marketing is een gebied waar Relate4u graag in verder gaat. Het verzamelen van persoonlijke gegevens voor klanten die later gebruikt kunnen worden voor het leveren van persoonlijke producten. Mede om deze reden is er besloten de toolkit zo algemeen mogelijk te houden en de basis van de toolkit te realiseren waarin de gebruiker centraal staat. Vanuit hier kan de toolkit aangevuld worden met specifieke dataopslag die op dat moment gewenst is.

Centraal staat dus de gebruiker. De gegevens die van de gebruiker worden opgeslagen is vrij algemene informatie. Dit betreft onder andere naam, lokale informatie, leeftijd en geslacht. Meer informatie komt aanbod in het gedeelte waar de opbouw van de database aan bod komt.

11.1.2 WIREFRAMES

Om een beter inzicht te krijgen hoe de interfaces er precies uitkomen te zien, zijn er 2 wireframes gemaakt die meer vertellen over de applicatie. Het eerste plaatje gaat over de administrator en over de client interface. Deze 2 zijn vrijwel identiek alleen heeft de administrator interface nog wat extra functionaliteit voor het aanmaken en wijzigen van campagnes.

Company
Logo

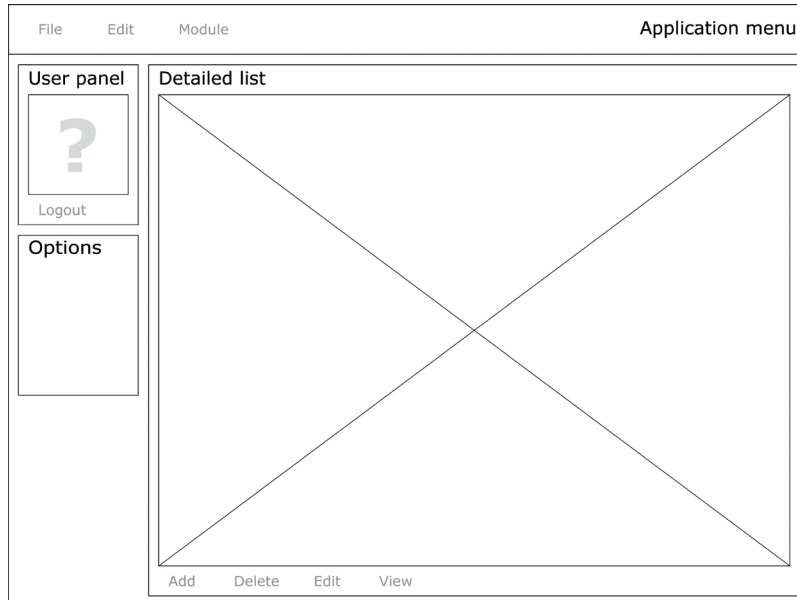
[Login](#) | [Register](#)

Header

Content

[Campaign](#) | [Tell a friend](#) | [Terms & conditions](#) | [Contact information](#)

Het tweede geeft in de wireframe een voorbeeld hoe de campaign interface er uit zou kunnen zien. Het geeft een voorbeeld in welk menu items praktisch kunnen zijn binnen een campagne.



11.1.3 GEKOZEN SOFTWARE EN WERKWIJZEN

Database: MySQL

De software die gekozen is om de database op te draaien is MySQL. Dit is standaard binnen Relate4u.

Server-side code: PHP

Voor de server side coding is gekozen voor PHP. Java is echter meest gangbare binnen Relate4u. Dit zou het later ook nog kunnen worden. Maar omdat de Java programmeurs het enorm druk hebben wordt de eerste versie in PHP gerealiseerd.

MVC Framework: PureMVC

PureMVC wordt het Actionscript framework waarin alle Flash interfaces opgebouwd zullen worden. Er is voor gekozen om niet zelf een framework te gaan bouwen, omdat bestaande frameworks een goede structuur en werkwijze bevatten om de toolkit te maken. Bovendien zijn deze al goed gedocumenteerd.

Cairngorm is afgefallen als framework, omdat dit framework zich alleen maar richt op Flex. Mocht er dus gekozen worden voor Cairngorm dan was het niet meer mogelijk om in Flash te ontwikkelen. Dit zou de vrijheid van het ontwikkelen van campagnes te veel beperken.

Remoting: Amfphp

Aangezien de server side coding PHP is moet er dus ook een remoting gateway komen die voor PHP gemaakt is. Amfphp is een remoting gateway die draait op PHP. Als straks de server side coding gemaakt gaat worden in Java dan zal er een andere

gateway bij moeten komen die geschikt is voor Java. Deze gateways kunnen altijd naast elkaar gebruikt blijven worden. Uiteindelijk is het niet belangrijk hoe de clients aan hun informatie komen. Het is belangrijk dát ze er aan komen.

Flash componenten: UIMovieClip

De werkwijze om een component te maken is in de meeste gevallen een component haar eigenschappen te laten overerven van de UIMovieClip class. Het maken van een component op deze manier zorgt er voor dat het component zowel in Flash als in Flex gebruikt kan worden.

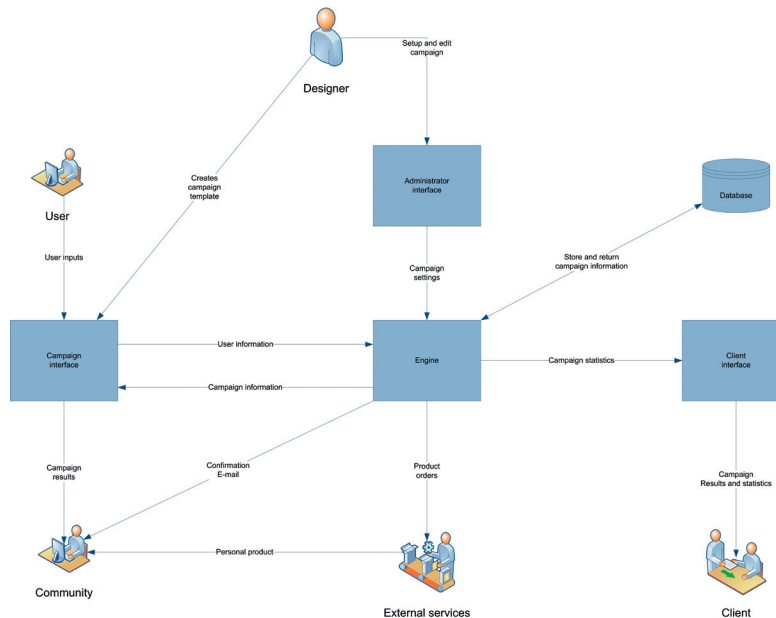
De andere manieren om componenten op te bouwen zullen ook nog gebruikt gaan worden. Dit hangt sterk af van de eisen van het component en in welke ontwikkel tool het gebruikt gaat worden. De componenten die alleen gebruikt worden in de client- en administrator-interface zullen vaak ontwikkeld worden binnen Flex.

11.1.4 OVERZICHT VAN DE TOOLKIT

Om een schematisch overzicht te krijgen hoe de toolkit in elkaar zit is dit diagram gemaakt. Dit diagram laat nog eens zien welke verantwoordelijkheden elke interface heeft. De administrator interface is het gedeelte waar een ontwerper van relate4u een campagne van maakt of eventueel wijzigt. De client interface is waar de klant van relate4u haar campagne statistieken kan op vragen. De gebruikers binnen een bepaalde community die de campagne willen bekijken komen binnen op de campaign interface. Hier geeft de gebruiker haar input en hier krijgt hij eventueel de feedback van de campagne.

De engine is het centrale gedeelte van de toolkit. Alle interfaces maken een connectie met dit gedeelte. De engine zorgt er vooral dat data naar de database wordt geschreven en dat data weer uit de database worden gehaald. Ook wordt de engine gebruikt om mailings en persoonlijke orders te versturen naar derden zoals een drukkerij of een multi player server. Als multiplayer server zorgt dit er voor dat er meerdere client interfaces met elkaar in contact staan, verschillende gebruiker tegen elkaar een spel kunnen spelen of met elkaar kunnen chatten.

Kijk voor het volledige schema in de digitale bijlage

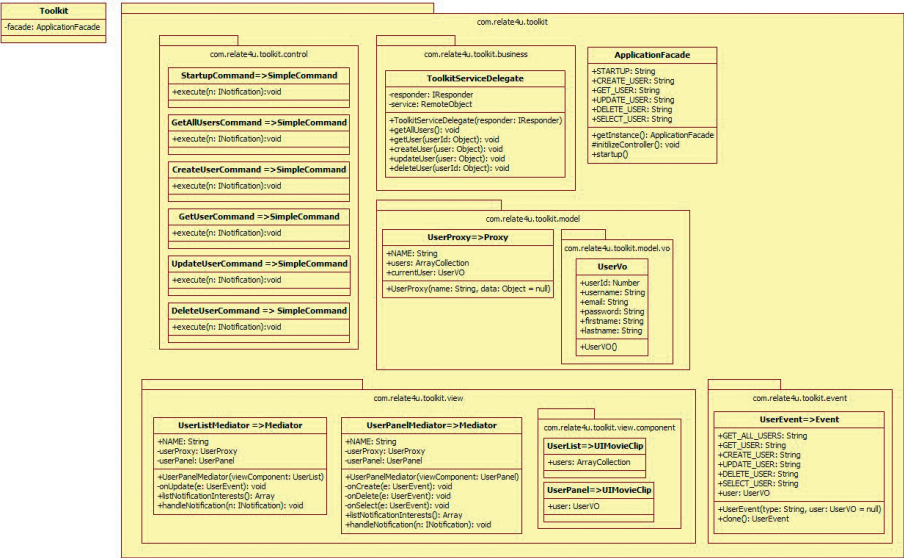


11.2. DEFINITIEVE OPZET

11.2.1 UML

Om tot een goed beeld te komen wat voor classes er allemaal geschreven moeten worden voor de toolkit wordt een UML class diagram gemaakt. Het maken van een goed UML schema kan bestaan uit meerdere stappen, waarin nog andere diagrammen gemaakt worden om tot een beter class diagram te komen,. Maar omdat er gebruik gemaakt wordt van het PureMVC framework wat al een goede structuur bevat is het maken van alleen een class diagram voldoende.

Het volgende schema is een class diagram van de toolkit. Dit zijn de classes die geprogrammeerd moeten worden om de toolkit te realiseren. Het schema is volledig te bekijken op de digitale bijlage.

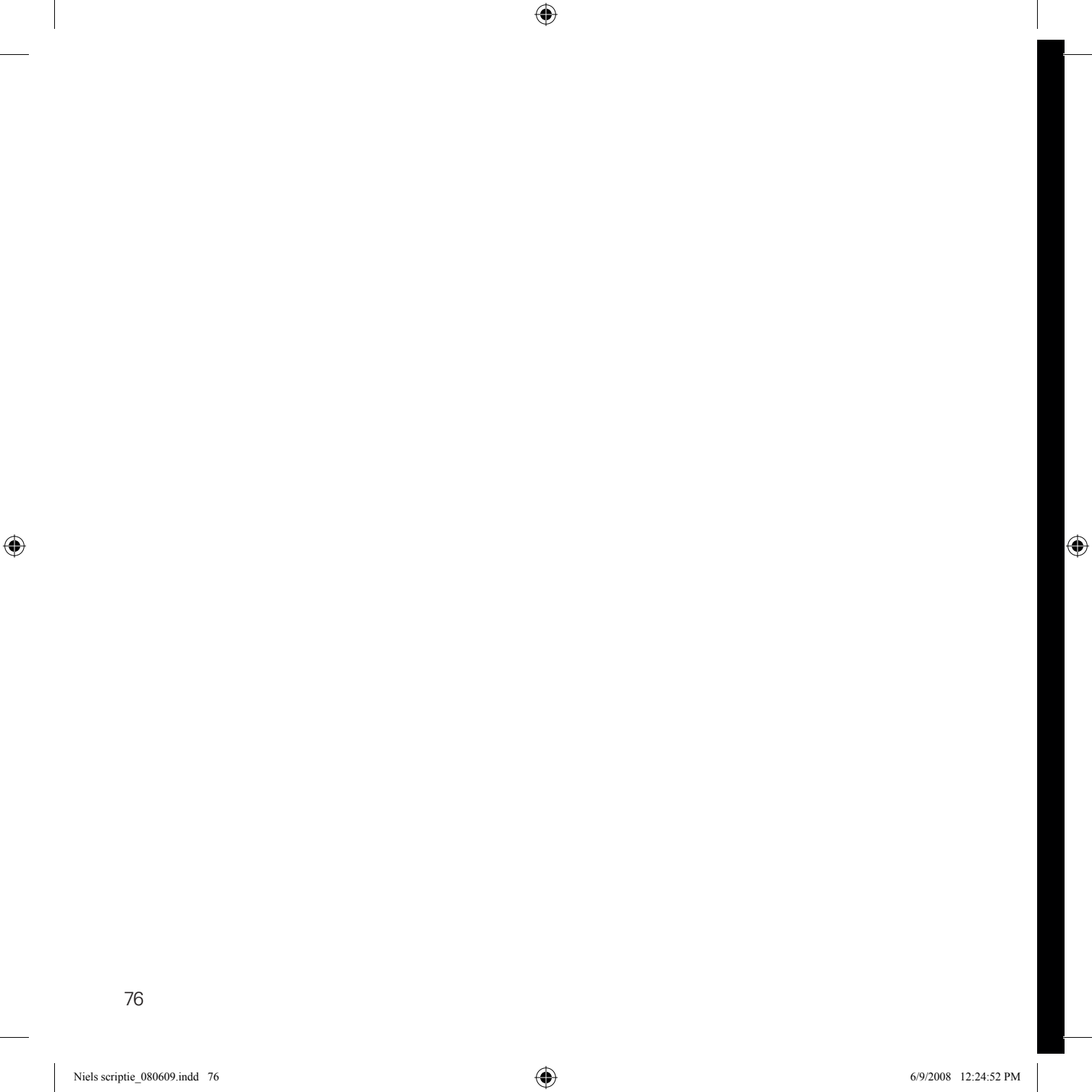


11.3. REALISATIE

Tijdens de afstudeeropdracht is er een ander project tussen gekomen. Dit betreft Philips Corporate Research Exhibition (CRE). CRE is een evenement intern bij Philips. Op dit evenement presenteren Philips onderzoekers nieuwe technieken aan andere afdelingen binnen Philips om deze eventueel toe te passen in bestaande producten. Relate4u maakt voor dit evenement alle interactieve media. Dit betreft de intranet site en de interactieve media die op het evenement draaien.

Deze opdracht heeft de vordering in de afstudeer opdracht zodanig vertraagt dat de realisatie van de toolkit uitgesteld moest worden. Dit zal in de komende weken na de stage uitgevoerd worden.

Alle uitkomsten van de onderzoeken zijn weltoegepast op de opdrachten die gedaan zijn voor Philips CRE



HOOFDSTUK 12.

Conclusie en aanbevelingen

12.1. CONCLUSIE

Om terug te komen op de centrale vraag.

“Hoe maak je een toolkit waarmee eenvoudig microcampagnes kunnen worden gemaakt en welke gegevens worden veelal in deze campagnes gebruikt”

Het komt er op neer dat microcampagnes veel verschillen ten op zichte van elkaar. Elke microcampagne gaat specifiek over een product of dienst en heeft haar eigen gegevens. Persoonlijke gegevens hebben deze campagnes als overeenkomst hebben. Dit staat daarom ook centraal in Viralz4u.

Voor het opzetten van een Rich Internet Application is het zeer verstandig om gebruik te maken. Dit patroon brengt een solide werkwijze om RIA's op te bouwen. Met deze werkwijze leer je efficiënter programmeren en je houdt meer overzicht over je code.

Om een goede connectie te creëren met externe facetten, zoals een database, moet er goed een afweging gemaakt worden wat voor soort communicatie je gaat gebruiken. Het is verstanding om AMF te gebruiken als communicatie. AMF zorgt er voor dat je objecten zowel server side als in de client blijven bestaan. Bovendien de data stroom is binair, wat veel kleiner en sneller is dan een normaal tekst formaat.

Het maken van Flash componenten is een taak die goed gepland moet worden. Eerst moet er goed bekeken worden wat de wensen en eisen zijn aan het component. Hierna kunnen afwegingen gemaakt worden op welke manier het component gemaakt gaat worden

12.2. AANBEVELINGEN

Ik wil graag aanbevelingen doen op een aantal zaken. Met name het gebruik van frameworks als je een applicatie gaat maken. Frameworks bieden een goede structuur en het zorgt dat je beter gaat programmeren. Het versneld de tijd voor het maken van een applicatie en je behoud overzicht. PureMVC is een zeer goed voorbeeld van een framework voor actionscript, maar ook voor PHP, JAVA en Javascript zijn er talloze frameworks die je werk zullen verbeteren.

EVALUATIE

Over het algemeen ben ik erg positief wat betreft de onderzoeken die zijn uitgevoerd. Van de onderzoeken, voornamelijk op het technische vlak, heb ik echt enorm veel geleerd. Over het tijdsverloop van het project ben ik wat minder tevreden. Rond de achtste week van de stage heb kreeg ik te horen dat Relate4U een groot evenement van Philips deed vormgeven. Ik heb gevraagd of ik bij een van deze vergaderingen mocht zitten. Vanaf dit moment kreeg ik taken naar mij toe die ze zelf niet of minder goed konden uitvoeren. Omdat op dat moment mijn opdracht nog op een laag pitje stond was dit een mooie kans om te laten zien wat ik kan.

Voor mijn planning was dit geen goede keuze. De taken werden groter en omdat ik de enigste was met de juiste expertise kon ik hier niet zomaar uitstappen. Van het maken van deze keuze heb ik absoluut geen spijt. Dit was voor mij een unieke kans om te werken met een grote klant als Philips. Ik kon mijn technische onderzoeken direct toe passen in de producten die ik voor Philips maakte. Ook stond ik in een rol waarin ik programmeurs met vormgevers liet communiceren en was ik dus een centraal punt. Ook heb ik later presentaties gedaan voor de managers van Philips, waar ik enorm veel van opgestoken heb.

Kortom voor het realiseren van Viralz4u kwam het niet ten goede, maar kijk er naar uit dit in de komende weken als nog te gaan uitvoeren. Alle producten zijn conceptueel en schematisch uitgewerkt en hoeven nu dus alleen nog maar gemaakt te worden.

LITERATUURLIJST

Artikelen

Wikipedia: Model-view-controller

<http://en.wikipedia.org/wiki/Model-view-controller>

Wikipedia: Data Access Object

http://en.wikipedia.org/wiki/Data_Access_Object

Wikipedia: Value Object

http://en.wikipedia.org/wiki/Data_Transfer_Object

Wikipedia: Application programming interface

<http://en.wikipedia.org/wiki/Api>

Wikipedia: Software development kit

http://en.wikipedia.org/wiki/Software_development_kit

Cairngorm vs puremvc quick comparison

<http://geekglue.blogspot.com/2007/10/cairngorm-vs-puremvc-quick-comparison.html>

Creating ActionScript 3.0 components in Flash

http://www.adobe.com/devnet/Flash/articles/creating_as3_components.html

Media

Online seminar: Vergelijkingen van MVC frameworks for Flex

https://admin.adobe.acrobat.com/_a300965365/p75214263/

Internetsites

<http://www.wikipedia.nl>
<http://www.wikipedia.com>
<http://www.adobe.com/devnet/>
<http://www.Flashfocus.nl>
<http://www.gotoandlearn.com>
<http://www.puremvc.org>
<http://labs.adobe.com/wiki/index.php/Cairngorm>
<http://livedocs.adobe.com>

Literatuur

Peter Hall, Chafic Kazoun & Joey Lott (2007), Programming Flex 2
Joey Lott & Danny Patterson(2006), Advanced ActionScript 3 with Design Patterns
Ralph Johnson (1994), Design Patterns Elements of Reusable Object-Oriented Software

Documenten

Niels Wijers en Mark van Hagen (2007) Onderzoek naar Virals

Bijlage A. Plan van aanpak

PROJECTOPDRACHT

PROJECTNAAM

Codenaam: Viralz4U

OPDRACHT NEMERS

Niels Wijers en Relate4U

OPDRACHT ORGANISATIE

Relate4U

OPDRACHT GEVER

Koen Luijten, Director

DE OPDRACHT

Er gaat een toolkit ontworpen worden binnen Flash/Flex, waarin het mogelijk is voor de designers binnen Relate4u gemakkelijk een microcampagne op te zetten. De toolkit bestaat uit een algemene basis die zowel aan de voorkant als aan de achterkant uitgebreid kan worden met componenten om de gebruiksvriendelijkheid en experience te verbreden. Aan de voorkant moet je denken aan componenten zoals het bekijken van data van bezoekers, invoegen van games en animatie enz. Aan de

achterkant komen er modules voor de klant om mails te versturen, gegevens op te halen van alle bezoekers enz. Ook is er de mogelijkheid om aanvragen door gebruikers bijvoorbeeld direct door te sturen naar de drukkerij.

De mogelijkheden voor uitbereiding van deze toolkit zijn eindeloos. In de eerste weken van de stage zal er onderzocht worden wat de minimale eisen van de toolkit zijn en hoe het opgezet gaat worden. Daarna wordt gekeken welke componenten gemaakt zullen worden.

Uiteindelijk zal de toolkit getest worden in een eerste microcampagne. Er zou bijvoorbeeld een microcampagne gemaakt kunnen worden om nieuwe Java programmeurs binnen te halen. De microcampagne kan bijvoorbeeld bestaan uit een kleine site waarin Java programmeurs geprikkeld worden voor Relate4u te kiezen. Ik ben verantwoordelijk voor deze microsite van begint tot het einde. Dit wil zeggen dat er eerst een onderzoek gedaan moet worden hoe het beste naar de programmeurs gecommuniceerd kan gaan worden. Uiteindelijk moet de site ontworpen worden en klaargemaakt zodat het in gezet kan gaan worden.

Of dit laatste het onderwerp van de testfase is, is nog niet bekend. Later in het project zal de het juiste onderwerp worden toegekend.

DOELSTELLING

Na het project moet het voor designers van Relate4u mogelijk zijn zonder / beperkte kennis van Actionscript snel microcampagnes / sites op te zetten. De microcampagnes / sites zijn interactief en zijn gekoppeld aan databases voor verder data verrijking en marketing doeleinden. De toolkit moet dus ingezet kunnen worden voor commercieel gebruik.

SUBDOELSTELLINGEN

Deze toolkit kan Relate4U de mogelijkheid geven nieuwe kanten en experiences van het internet op te zoeken en te vermarkten naar bestaande en nieuwe klanten.

EINDPRODUCT

Het eindproduct is een toolkit waarmee men snel microsites op kan zetten. Deze toolkit bestaat uit verschillende onderdelen

TOOLKIT ENGINE

Dit is het hart van de toolkit. De core van de toolkit zorgt er voor dat alle onderdelen van de toolkit met elkaar kunnen communiceren. De core maakt het dus mogelijk dat de Community en Client interface hun gegevens uit de database halen en kunnen opslaan. Ook zorgt de core ervoor dat informatie gestuurd wordt naar external service zoals een drukkerij of dat er een bevestigingsmail gestuurd wordt naar de gebruikers. Dit wordt gemaakt in Java.

CAMPAIGN INTERFACE

Dit gedeelte bestaat uit een aantal core componenten die vereist zijn voor een minimale opzet van een microsite. Deze componenten worden gemaakt in Flash en voor Flash. De designer hoeft deze alleen maar op de stage van Flash te slepen en de settings op te geven. Deze kan de designer skinnen en stylen voor de juiste sfeer in de campagne

CLIENT INTERFACE

De client interface is het backend van de toolkit. In dit gedeelte kan de klant inloggen om informatie op te halen betreft de campagne. Denk hierbij aan persoonlijke informatie van gebruikers, uitkomsten van polls en alle informatie die met de campagne gegenereerd word. Dit wordt gemaakt in Flex.

ADMINISTRATOR INTERFACE

Het gedeelte waar de designer inlogt om een campagne aan te maken, wijzigen en te verwijderen. Dit wordt gemaakt in Flex

DOCUMENTATIE

Viralz4u wordt gedocumenteerd zodat gebruikers, klanten, designers en vormgevers na de overdracht eenvoudig aan de slag kunnen met deze toolkit.

TEST CAMPAGNE (OPTIONEEL)

De eerste test die de toolkit zal moeten doorstaan.

TOOLKIT EXTENSIONS (OPTIONEEL)

Een set componenten die het mogelijk maakt de toolkit meer functionaliteit te geven

PRODUCT EISEN

TOOLKIT ENGINE

- Engine verwerkt data van Administrator interface en slaat deze op in database
- Engine verwerkt data van Campaign interface en slaat deze op in database
- Engine geeft campagne informatie door aan campagne, administrator en client interface
- Engine geeft product orders door aan external services
- Engine stuurt gebruikers bevestigingsemail.

ADMINISTRATOR INTERFACE

- Admin moet een campagne kunnen aanmaken
- Admin moet een campagne kunnen wijzigen
- Admin moet een campagne kunnen verwijderen
- Admin moet optionele componenten aan een campagne kunnen toevoegen

CAMPAIGN INTERFACE

- User moet zich kunnen aanmelden
- User moet zijn gegevens kunnen wijzigen
- User moet zich kunnen afmelden
- User moet zich in kunnen schrijven voor een nieuwsbrief
- User moet de site kunnen doorsturen naar vrienden
- User moet contact informatie kunnen bekijken
- User moet de voorwaarden kunnen bekijken
- User moet campagne- en bedrijfsinformatie kunnen bekijken
- Designer moet campaign interface op kunnen zetten aan de hand van componenten

CLIENT INTERFACE

- Client moet statistieken van campagne kunnen opvragen
- Client moet inlog gegevens kunnen wijzigen

DATABASE

- Slaat alle campagnes op
- Slaat alle gebruikers op per campagne
- Slaat alle campagne resultaten op per campagne.

PROJECTACTIVITEITEN

De projectactiviteiten zullen worden uitgevoerd in 4 verschillende fases. Hieronder de fases met daarin de verschillende taken per fase.

ANALYSEFASE

- Plan van aanpak
- Onderzoek microcampagnes
- Onderzoek opzetten van toolkit
- Onderzoek naar het maken van Flash componenten

ONTWERPFASE

- Structuur opzetten / uitdenken van de toolkit (UML)
- Wireframes maken van de userinterfaces
- Vaststellen van de basis gegevens van toolkit en ontwerp van de database
- Eisen opzetten voor componenten / extensions

REALISATIEFASE

- Uitwerken van de engine
- Database opzetten
- Uitwerken van de userinterfaces
- Uitwerken van campaign interface componenten voor Flash
- Documenteren van toolkit

TESTFASE

- Afhandeling project

- Uitwerken van test campagne (subproject)
- Terugkoppeling van het resultaat van de test campagne

PROJECTGRENZEN

Dit project heeft ook grenzen waarbinnen de productie plaats moet vinden. In overeenstemming met de opdrachtgever zijn de volgende grenzen opgesteld:

DE PRODUCTEN

OP TE LEVEREN PRODUCTEN

- Document: Stagewerk plan
- Document: Onderzoek microcampagnes
- Document: Onderzoek van het maken van componenten
Dit document is een set van resources hoe je het beste componenten kan opzetten binnen Flash
- Document: Structuur van de toolkit (UML)
Dit document beschrijft de volledige structuur van de toolkit. Deze structuur zal de blauwdruk zijn doorgaans het hele project
- Wireframes van userinterfaces.
Lijn tekeningen van de opbouw van de userinterfaces. Dit betreft de Admin en de client interface. De wireframes zullen later uitgewerkt worden in Flex.
- Database ontwerp
Schema van de database. In dit schema zal alle is te zien welke informatie er opgeslagen gaat worden en welke relaties de tabellen met elkaar zullen hebben.
- Document: ontwerp eisen voor aanvullende componenten
Hierin staan de eisen waaraan aanvullende componenten zich moeten voldoen. Dit zorgt ervoor dat personen die geen kennis hebben van het project eenvoudig extensies kunnen maken om de functionaliteit van het eindproduct uit te bereiden.
- Documentatie
- Eindproduct: Campagne toolkit
- Toolkit engine

- Campaign interface
- Client interface
- Administrator interface
- Toolkit core components
- Toolkit extensions

KWALITEIT

Om kwaliteit van het eindproduct en tussentijdse rapporten te waarborgen, houdt de groep meerdere malen per week contact. Verder wordt er per week een onofficiële vergadering gehouden.

Verder zullen we werken met de planning van dit plan van aanpak voor een degelijk projectverloop en zullen er deadlines worden gezet om tijdnood te voorkomen. Het project zal in fases verlopen, waarbij voornamelijk opdracht na opdracht zal worden afgewerkt, zie hiervoor hoofdstuk 5: Producten.

PROJECTORGANISATIE

ORGANISATIE

De rollen in de project groep zijn als volgt verdeelt:

PROJECT LEIDER

Niels Wijers

ARCHIVARIS

Niels Wijers

UITVOERENDE PROJECTLEDEN

Niels Wijers, Mischa Zabransky, Koen Luijten

WERKGROEOPLEIDER ANALYSEFASE

Niels Wijers

WERKGROEOPLEIDER ONTWERPFASE

Niels Wijers

WERKGROEOPLEIDER REALISATIEFASE

Niels Wijers

INFORMATIE

Communicatie zal binnen de groep meerdere keren per week verlopen.

Verder wordt er bij de wekelijkse vergadering voorgelegd aan de projectleider wat er is gedaan, wat er gedaan moet worden en hoe het staat met de planning.