

# **Genereren van naar wens in te richten videospelers**

**Document:**

Eindverslag v1.0

**Student:**

David Krijgsman, 20040036

**Bedrijfsmentoren:**

Maarten van Appel, Diederik van Huijstee

**Begeleidende docenten:**

Ellen Grummels

Jacob van der Linden

**Datum:**

8 januari 2008

David Krijgsman, 20040036  
Genereren van naar wens in te richten videospelers  
Eindverslag v1.0

## **Genereren van naar wens in te richten videospelers**

Wend Visual Communication

8 januari 2008

Amsterdam

G3 Afstuderen

Haagse Hogeschool

Dhr J.P. van der Linden

Mw E.P.H. Grummels

## REFERAAT

### **Genereren van naar wens in te richten videospelers**

David Krijgsman, 20040036

#### *Jaar van uitgave*

2007, Haagse Hogeschool

#### *Samenvatting*

Dit document beschrijft het proces dat is doorlopen bij het ontwikkelen van de Flash Videospeler Generator. Wend Visual Communication (Wend) heeft gemerkt dat een groot deel van haar huidige klanten video aanbiedt op hun websites. Hiervoor wordt Windows Media Player gebruikt. Deze videospeler is minder geschikt voor andere Operating Systems dan Windows. Wend had daarom de behoefte aan een gebruiksvriendelijke videospeler die browser en Operating System onafhankelijk op websites gebruikt kan worden. Bovendien is de vormgeving van Windows Media Player niet aan te passen waardoor de speler qua stijl, uitstraling en design niet overeen komt met de rest van de website. De Flash Videospeler Generator is een applicatie waarmee gebruikers zelf een videospeler kunnen vormgeven en op hun website kunnen plaatsen. De werkzaamheden die de student hiervoor heeft uitgevoerd zijn onder andere: onderzoek doen naar doelgroep, applicatie vormgeven, applicatie technisch ontwerpen, usability testen van vormgeving, applicatie ontwikkelen en usability testen van applicatie.

#### *Trefwoorden*

Videospeler, applicatie ontwikkeling, Jesse James Garret, The Elements of User Experience, UML, usability, Flash, PHP

## VOORWOORD

Graag bedank ik Wend voor de kans die ze mij hebben geboden en in het kader van mijn afstuderen deze opdracht aan mij toe te vertrouwen. Daarnaast een dankwoord richting de testpersonen die hun medewerking hebben verleend tijdens de verschillende onderzoeken die ik gedurende het project heb uitgevoerd. Tot slot bedank ik de personen uit mijn directe omgeving die me tijdens het project hebben bijgestaan.

David Krijgsman

Amsterdam, 8 januari 2008

## INHOUDSOPGAVE

1	INLEIDING	1
2	OPDRACHTOMSCHRIJVING	3
2.1	Opdrachtgever	3
2.2	Uitvoerende	3
2.3	Aanleiding	3
2.4	Probleemstelling	4
2.5	Doelstelling	4
2.6	Uitgangssituatie	4
3	AANPAK	5
3.1	Werkzaamheden	5
3.1.1	<i>Projectoriëntatie en projectorganisatie</i>	6
3.1.2	<i>Vorbereiding op ontwerp</i>	6
3.1.3	<i>Technisch ontwerp</i>	6
3.1.4	<i>Visueel ontwerp van iconen en schermen in verschillende stijlen</i>	6
3.1.5	<i>Usability testen van iconen en schermen</i>	7
3.1.6	<i>Visueel ontwerp uitwerken</i>	7
3.1.7	<i>Ontwikkelen van de applicatie</i>	7
3.1.8	<i>Usability testen van de gehele applicatie</i>	7
3.1.9	<i>Herzien van applicatie</i>	7
3.2	Methode	8
3.3	Technieken	10
3.3.1	<i>Projectoriëntatie en projectorganisatie</i>	11
3.3.2	<i>Vorbereiding op ontwerp</i>	14
3.3.3	<i>Technisch ontwerp</i>	17
3.3.4	<i>Visueel ontwerp van iconen en schermen in verschillende stijlen</i>	20
3.3.5	<i>Usability testen van iconen en schermen</i>	20
3.3.6	<i>Visueel ontwerp uitwerken</i>	20
3.3.7	<i>Ontwikkelen van de applicatie</i>	21
3.3.8	<i>Usability testen van de gehele applicatie</i>	21
3.3.9	<i>Herzien van applicatie</i>	22

3.4	Op te leveren producten	22
3.4.1	<i>Startdocument</i>	23
3.4.2	<i>Vorbereidend ontwerprapport</i>	24
3.4.3	<i>Technisch ontwerprapport</i>	24
3.4.4	<i>Visueel ontwerprapport van iconen en schermen</i>	25
3.4.5	<i>Testplan voor fase 'Usability testen van iconen en schermen'</i>	25
3.4.6	<i>Testresultaten en verwerking voor fase 'Usability testen van iconen en schermen'</i>	25
3.4.7	<i>Visueel ontwerprapport van alle schermen van de applicatie</i>	26
3.4.8	<i>Testplan voor fase 'Usability testen van de gehele applicatie'</i>	26
3.4.9	<i>Testresultaten en verwerking van fase 'Usability testen van de gehele applicatie'</i>	26
3.4.10	<i>Applicatie</i>	27
3.4.11	<i>Overdrachtsverslag</i>	27
4	PROJECTORIËNTATIE EN PROJECTORGANISATIE OPSTELLEN	28
4.1	Probleem- en doelstelling opstellen	28
4.2	Urenbegroting opstellen	28
4.3	Planning opstellen	29
4.4	Projectmanagement methode vastleggen	30
5	VOORBEREIDEN OP ONTWERP	32
5.1	Userneeds en siteobjectives opstellen	32
5.2	Doelgroepomschrijving opstellen	32
5.3	Moodboard ontwikkelen	34
5.4	Persona's ontwikkelen	35
5.5	Systeemeisen opstellen	37
6	TECHNISCH ONTWERP OPSTELLEN	39
6.1	(Kandidaat)klassen opstellen	39
6.2	Klassediagram opstellen	40
6.3	Usecases opstellen	41
6.4	Usecasediagram opstellen	43

6.5	Sequentiediagrammen opstellen	45
6.6	Database ontwerpen	46
6.7	Navigatiemodel opstellen	49
6.8	Bestandsstructuur opstellen	50
7	VISUEEL ONTWERP VAN ICONEN EN SCHERMEN OPSTELLEN	51
7.1	Wireframes opstellen	51
7.2	Iconen schetsen	53
7.3	Schermonderdelen schetsen	54
7.4	Schermonderdelen uitwerken in schermen	55
7.5	Iconen uitwerken	60
7.6	Videospeler stijlen ontwerpen	61
8	USABILITY TESTEN VAN ICONEN EN SCHERMEN	62
8.1	Doel bepalen	62
8.2	Testtaken opstellen	63
8.2.1	<i>Iconen herkennen</i>	63
8.2.2	<i>Iconen en betekenissen koppelen</i>	64
8.2.3	<i>Vragen over schermen</i>	65
8.3	Testresultaten verwerken	66
8.3.1	<i>Iconen herkennen</i>	66
8.3.2	<i>Iconen en betekenissen koppelen</i>	68
8.3.3	<i>Vragen over schermen</i>	71
9	VISUEEL ONTWERP UITWERKEN	74
9.1	Schetsen bijstellen	74
9.2	Iconen herontwerpen	74
9.3	Iconen testen	75
10	ONTWIKKELEN VAN DE APPLICATIE	76
10.1	Technieken kiezen	76
10.2	Basis van de applicatie leggen	76
10.3	Klassen schrijven	77
10.4	Pagina's ontwikkelen	79
10.5	Videospeler ontwikkelen	80

11	USABILITY TESTEN VAN DE GEHELE APPLICATIE	82
11.1	Doel bepalen	82
11.2	Testtaken opstellen	82
11.3	Vragenlijsten opstellen	83
11.4	Testinhoud verzorgen	84
11.5	Testresultaten verwerken	85
12	HERZIEN VAN DE APPLICATIE	87
12.1	Delen van de applicatie herontwikkelen	87
12.2	Overdrachtsverslag schrijven	89
13	EVALUATIE	91
13.1	Product	91
13.2	Proces	92
13.3	Conclusie	93
	FIGUREN INDEX	94
	LITERATUURLIJST	97

#### *EXTERNE BIJLAGEN*

I	STARTDOCUMENT	
II	VOORBEREIDEND ONTWERPRAPPORT	
III	TECHNISCH ONTWERPRAPPORT	
IV	VISUEEL ONTWERPRAPPORT VAN ICONEN EN SCHERMEN	
V	TESTPLAN VOOR 'USABILITY TESTEN VAN ICONEN EN SCHERMEN'	
VI	TESTRESULTATEN VAN 'USABILITY TESTEN VAN ICONEN EN SCHERMEN'	
VII	TESTPLAN VOOR 'USABILITY TESTEN VAN DE GEHELE APPLICATIE'	



VIII TESTRESULTATEN VAN 'USABILITY TESTEN VAN DE GEHELE APPLICATIE'

IX OVERDRACHTSVERSLAG

## 1 INLEIDING

In het kader van mijn afstuderen bij de opleiding Communication and Multimedia Design aan de Haagse Hogeschool heb ik een periode van 4 maanden gewerkt aan een project voor Wend. De opdracht betrof de ontwikkeling van een applicatie die de gebruiker in staat stelt een videospeler vorm te geven en deze op een website plaatsen. De beoogde doelstelling hierbij was Wend in staat te stellen een bredere doelgroep aan te spreken die een videospeler in de stijl van hun website wenst. Daarnaast moet er tijdswinst worden geboekt doordat gebruikers zelf hun videospeler vormgeven.

Het doel van dit document is het inzichtelijk maken van het proces dat ik tijdens het project heb doorlopen. Het is geschreven voor dhr. J.P. van der Linden en mevr. E.P.H. Grummels die als examinatoren vanuit de Haagse Hogeschool dat proces zullen beoordelen en voor de gecommitteerde die tijdens de examenzitting aanwezig zal zijn.

In hoofdstuk 2 wordt allereerst de opdracht nader toegelicht. Er wordt ingegaan op de betrokken partijen, de aanleiding en het probleem die aan de opdracht ten grondslag liggen. Vervolgens wordt de doelstelling gegeven die gedurende het project wordt nagestreefd.

In hoofdstuk 3 wordt er vooruit gekeken. Het project wordt in fases verdeeld en er wordt in gegaan op de wijze waarop ik de verschillende werkzaamheden gedurende het project wil aanpakken. Er wordt beschreven welke methode en technieken ik op relevantie en toepasbaarheid heb beoordeeld en ik motiveer waarom ik een bepaalde methode of techniek heb gekozen. Tot slot wordt beschreven welke producten ik op ga leveren.

Hoofdstuk 4 tot en met hoofdstuk 12 zijn beschrijvend. In chronologische volgorde beschrijf ik de werkzaamheden die ik heb uitgevoerd en de keuzes die ik daarbij heb gemaakt. Elk hoofdstuk vertegenwoordigt een fase van het project. De fases zijn achtereenvolgens: 'Projectoriëntatie en projectorganisatie', 'Vorbereiding op ontwerp', 'Technisch ontwerp', 'Visueel ontwerp van iconen en schermen', 'Usability testen van iconen en schermen', 'Visueel ontwerp ontwikkelen', 'Ontwikkelen van de applicatie', 'Usability testen van de gehele applicatie' en 'Herzien van de applicatie'.

In hoofdstuk 13 wordt er teruggekeken op het ontwikkelde product en het doorlopen proces.

Tot slot volgen de Figurenindex en Literatuurlijst.

Verschillende delen van dit document zijn in kaders geplaatst. Deze delen omschrijven de theorie van het betreffende onderwerp.

Alle afbeeldingen, tabellen etc. zijn volgens het format *fig. [hoofdstuknummer].[opeenvolgend nummer]* genummerd. Het opeenvolgende nummer wordt na elk hoofdstuk op 1 gezet. Een overzicht van figuren is te vinden in de Figurenindex waar de figuren per hoofdstuk in alfabetische volgorde zijn geplaatst.

## 2 OPDRACHTOMSCHRIJVING

In dit hoofdstuk zal ik de opdracht nader toelichten.

### 2.1 Opdrachtgever

Wend is al jaren bezig met het ontwikkelen van interactieve webapplicaties en websites. Wend werkt voor middelgrote en grote bedrijven. Ze kan onder andere Hema, Pearle en KLM onder haar klanten rekenen. Het bedrijf telt drie partners en beslaat in totaal ongeveer acht medewerkers. Wend is recentelijk verhuisd naar het businesspark te Amsterdam.

Vanuit Wend zullen de volgende twee personen mij tijdens het afstuderen begeleiden.

Maarten van Appel  
maarten@wend.nl

Diederik van Huijstee  
diederik@wend.nl

### 2.2 Uitvoerende

Het project zal worden uitgevoerd door:

David Krijgsman  
david@limoenmultimedia.nl

### 2.3 Aanleiding

Wend heeft gemerkt dat een groot deel van haar huidige klanten video aanbiedt op hun websites. Hiervoor wordt Windows Media Player gebruikt. Deze videospeler is minder geschikt voor andere Operating Systems dan Windows. Er is behoefte aan een gebruiksvriendelijke videospeler die browser en Operating System onafhankelijk op websites gebruikt kan worden. Bovendien is de vormgeving van Windows Media Player niet aan te passen waardoor de speler qua stijl, uitstraling en design niet overeen komt met de rest van de website.

## 2.4 Probleemstelling

Wend kan haar diensten slechts aan een beperkte doelgroep aanbieden omdat ze momenteel niet de mogelijkheid heeft om haar klanten een gebruiksvriendelijke browser en Operating System onafhankelijke videospeler aan te bieden. De vormgeving van de huidige gebruikte videospeler heeft een standaard vormgeving die niet overeen komt met de omgeving waarin de videospeler zal komen te staan. Het kost te veel tijd om voor elke klant een passende vormgeving te ontwikkelen en daar de videospelers op aan te passen.

## 2.5 Doelstelling

De doelstelling van het project is het mogelijk maken om een bredere doelgroep aan te spreken, een doelgroep die videospelers in de stijl van de rest van het product wenst. Daarnaast is het doel tijd te winnen; het ontwikkelen van een product dat gebruik maakt van een videospeler zal minder tijd kosten omdat de klant zelf de vormgeving van de videospeler zal maken en zelf de videospeler kan implementeren. Wend kan dit via de applicatie ook op een gemakkelijkere manier doen. Tevens speelt het aftroeuen van concurrenten een rol omdat Wend haar klanten een hogere mate van vrijheid in vormgeving biedt bij het gebruik van een videospeler.

## 2.6 Uitgangssituatie

Er is geen documentatie aanwezig die ik zou kunnen gebruiken voor het uitvoeren van de opdracht. Ook eventuele onderzoeksresultaten zijn niet aan de orde.

Wat betreft software maak ik gebruik van verschillende applicaties van Adobe zoals:

- Photoshop
- Dreamweaver
- Flash

Voor het documenteren van het project gebruik ik Microsoft Office:

- Word
- Excel
- Visio

### 3 AANPAK

In dit hoofdstuk zal ik de wijze waarop ik het project wil aanpakken beschrijven.

#### 3.1 Werkzaamheden

Het project is in de volgende fases ingedeeld.

- Projectoriëntatie en projectorganisatie
- Voorbereiding op ontwerp
- Technisch ontwerp
- Visueel ontwerp van iconen en schermen in verschillende stijlen
- Usability testen van iconen en schermen
- Visueel ontwerp uitwerken
- Ontwikkelen van de applicatie
- Usability testen van de gehele applicatie
- Herzien van de applicatie

Deze fases verhouden zich qua hoeveelheid werk als volgt.

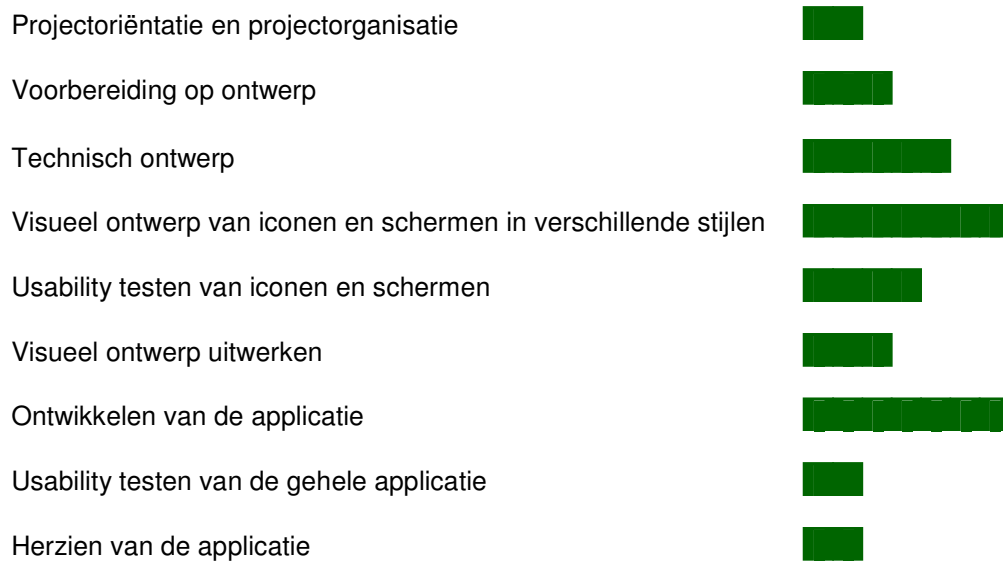


fig. 3.1 Faseverhouding

### *3.1.1 Projectoriëntatie en projectorganisatie*

In deze fase wordt het project geïnitieerd. Er wordt een startdocument opgesteld dat informatie en afspraken met betrekking tot het project vastlegt. Hierin komen onder andere probleem- en doelstelling, uit te voeren werkzaamheden en de te gebruiken methoden en technieken. Ook worden de afspraken met betrekking tot regelmaat van rapportage en rapportagetechniek vastgelegd. Naast de afspraken die zijn gemaakt kent het startdocument ook een dynamisch gedeelte, de planning. Deze zal gedurende het project telkens worden bijgehouden en waar nodig aangepast, zodat ik altijd weet waar ik in het project sta.

### *3.1.2 Voorbereiding op ontwerp*

Gedurende deze fase krijgt het project een inhoudelijke basis van waaruit verder gewerkt wordt. Er wordt informatie gewonnen over de aard van het project en de doelgroep. De systeemeisen worden opgesteld en ter goedkeuring aan de opdrachtgever voorgelegd.

### *3.1.3 Technisch ontwerp*

Tijdens deze fase wordt met de systeemeisen en de doelgroep in het achterhoofd het systeem technisch vormgegeven. Hierbij ligt de nadruk op de systeemeisen. Alle systeemeisen moeten in het technisch ontwerp verwerkt worden zodat niets over het hoofd wordt gezien en het systeem werkt zoals dat aanvankelijk is bedacht. De relatie van verschillende onderdelen van het systeem wordt duidelijk gemaakt en de mogelijkheden die de gebruiker heeft komen helder naar voren.

### *3.1.4 Visueel ontwerp van iconen en schermen in verschillende stijlen*

Voor het visueel ontwerp wordt informatie geput uit het technisch ontwerp, de systeemeisen en de doelgroepbeschrijving. Hierin speelt de doelgroepbeschrijving de belangrijkste rol. Bij elke stap die tijdens het visueel ontwerp wordt genomen wordt in ogenschouw genomen wat de reactie van de doelgroep zou zijn. Elke stap zal ook aan de opdrachtgever worden gepresenteerd en worden besproken. Er zullen verschillende concepten worden ontwikkeld; in de volgende fase wordt getest welke stijl of combinatie van stijlen het beste bij de doelgroep aansluit.

### *3.1.5 Usability testen van iconen en schermen*

Het product bestaat tot nu toe alleen uit afbeeldingen en schermen in verschillende stijlen. Gedurende deze fase zullen de verschillende stijlen aan de doelgroep worden getoetst. De stijl die het beste uit de bus komt zal verder worden uitgewerkt tot volledig ontwerp.

### *3.1.6 Visueel ontwerp uitwerken*

De stijl of combinatie van stijlen die het beste blijkt te werken zullen worden uitgewerkt en bijgeschaafd, totdat alle visuele aspecten van de applicatie vastliggen.

### *3.1.7 Ontwikkelen van de applicatie*

Tijdens deze fase wordt begonnen aan de bouw van het systeem. Omdat zowel het technisch ontwerp als het visueel ontwerp al vast ligt is dit een kwestie van toepassen. Het visueel ontwerp wordt vertaald naar de technieken die gebruikt worden voor de Videospeler Generator. Er zullen naar verwachting wel enkele zaken ten opzichte van het technisch ontwerp veranderen. Uit eigen ervaring weet ik dat er altijd dingen zijn die anders lopen dan verwacht. In deze fase zal het grootste gedeelte van de applicatie worden ontwikkeld.

### *3.1.8 Usability testen van de gehele applicatie*

Bij het usability testen wordt gekeken of de applicatie aansluit bij de verwachtingen van de gebruiker. Er worden antwoorden gezocht op vragen als 'Kan de gebruiker makkelijk zijn weg vinden?', 'Zijn de verschillende mogelijkheden duidelijk?', 'Is het systeem consistent?' en 'In hoeverre is de gebruiker tevreden over de applicatie?'.

### *3.1.9 Herzien van de applicatie*

Gedurende deze fase zal de applicatie worden afgemaakt. De resultaten van het usability testen spelen hierbij een belangrijke rol. Alle zaken waar de gebruikers moeite mee hadden of suggesties die de gebruikers hadden zullen worden meegenomen bij het ontwikkelen.

Verschillende reeds ontwikkelde onderdelen van het systeem zullen dus worden herzien om het geheel te laten aansluiten bij de verwachtingen van de gebruiker.



Omdat naar verwachting de applicatie niet 100% overeen komt met het aanvankelijke technische ontwerp zal de structuur van de applicatie nogmaals worden beschreven zodat eventuele vervolgon ontwikkelaars weten hoe het systeem in elkaar steekt.

### 3.2 Methode

Voor het project heb ik de volgende methoden overwogen.

- *Jesse James Garret, the elements of user experience (JJG)*

Deze methode beschrijft de ontwikkeling van een concept met behulp van een vijftal planes waarbij de gebruiker centraal staat. Elke plane kent een aantal stappen die uitgevoerd kunnen worden waarna tot de volgende plane wordt overgegaan. De planes overlappen elkaar wat een hoge flexibiliteit tot gevolg heeft. De planes zijn als volgt.

  - Strategy plane  
De doelstellingen van de website wordt hier vastgelegd (siteobjectives). Ook wordt beschreven wat de gebruikers uit website moeten kunnen halen (userneeds).
  - Scope plane  
Het project wordt afgebakend. Er wordt afgesproken wat er precies ontwikkeld gaat worden. Er worden systeemeisen opgesteld.
  - Structure plane  
De structuur van het product wordt duidelijk. De hiërarchie en onderlinge verhouding van de verschillende onderdelen worden beschreven.
  - Skeleton plane  
De website krijgt een indeling. De verschillende elementen worden met behulp van eenvoudige blokken op het scherm geplaatst.
  - Surface plane  
Dit is de laatste stap waarbij het geheel wordt uitgewerkt tot eindproduct.
- *Iterative Application Development (IAD)*

Bij deze methode wordt het project in kleine stappen opgedeeld. Elk deel is een pilot. De pilots kunnen gelijktijdig door verschillende teams worden ontwikkeld. Het ontwikkelen bestaat uit drie fases. Dit zijn de definitiestudie, pilot-ontwikkeling en invoering. Deze fases worden steeds herhaald. Elke cyclus levert een pilot op. Uiteindelijk worden de pilots bij elkaar gevoegd tot een eindproduct.

Deze methode wordt gekenmerkt door de grote betrokkenheid van de eindgebruikers en de opdrachtgever. Systeemeisen worden gedurende het ontwikkelproces aangepast.

- *Dynamic Systems Development Methodology (DSDM)*

Deze methode legt de ontwikkeling van een systeem vast met behulp van tijdsplanning. De duur van het project ligt vast. Systeemeisen worden van tevoren globaal vastgelegd en van prioriteit voorzien. Gedurende het project komen steeds meer gedetailleerde specificaties naar boven. Ook deze worden weer voorzien van prioriteit. De systeemeisen met de meeste prioriteit worden steeds als eerste ontwikkeld. DSDM kent de volgende fases.

- Feasibility study fase  
Tijdens deze fase wordt bepaald of het systeem geschikt is voor het bedrijf uitgaande van de gekozen technische realisatie, kosten en duur.
- Business study fase  
Gedurende deze fase worden globale systeemeisen opgesteld.
- Functional model iteration fase  
De systeemeisen worden in deze fase omgezet naar een functioneel model.
- System design/build fase  
In deze fase wordt een functioneel prototype uitgewerkt dat moet voldoen aan de systeemeisen.
- Implementation fase  
Het systeem wordt aan de eindgebruikers opgeleverd voor evaluatie. Als het systeem voldoet wordt het definitief opgeleverd, anders wordt een eerdere fase opnieuw uitgevoerd.

De eindgebruiker is gedurende het project zeer betrokken bij de ontwikkeling.

*fig. 3.2 Methoden, JJG, IAD en DSDM*

Jesse James Garret wordt op school onderwezen. IAD en DSDM zijn het gevolg van onderzoek op internet.

Ik heb voor dit project voor JJG gekozen. Deze methode legt de nadruk op de conceptontwikkeling waarbij de gebruiker centraal staat; IAD en DSDM richten zich meer op de globale inrichting van een project.

De doelgroep bestaat uit mensen met een gemiddeld internet- en computergebruik. Het zijn mensen die niet veel weten van het ontwikkelen van websites. Daarom is het vormgeven en

implementeren een relatief complexe handeling voor deze groep mensen. Het is dus van groot belang dat de applicatie gebruiksvriendelijk is en de gebruiker door de verschillende mogelijkheden heen loodst. JJG biedt alle handvatten om deze mate van gebruiksvriendelijkheid te implementeren.

IAD is beter geschikt voor projecten waar meer mensen mee bezig zullen zijn zodat er verschillende teams parallel aan pilots kunnen werken. De methode is veel meer gericht op samenwerken van projectleden wat bij mijn opdracht niet aan de orde is. Daarnaast is een van de principes van IAD dat de eindgebruiker zeer betrokken is. Ondanks dat ik op verschillende momenten de eindgebruiker bij het project zal proberen te betrekken zal ik de mate van betrokkenheid vereist bij het hanteren van IAD naar alle waarschijnlijkheid niet halen. De doelgroep bestaat uit middelgrote tot grote bedrijven waar mensen veelal druk zijn en weinig tijd zullen hebben om zich bezig te houden met de applicatie die ik ga bouwen. Op dit moment is het nog niet bekend in welke mate ik incentives kan bieden aan bijvoorbeeld testpersonen die hun medewerking verlenen.

Een van de basis principes van DSDM is een actieve deelname van de eindgebruikers. Zoals hierboven vermeld ben ik van mening dat ik niet op die mate van betrokkenheid kan rekenen. Daarnaast is ook bij deze methode samenwerking van belang wat bij dit project niet aan de orde is. Ook heeft Wend al een duidelijke applicatie voor ogen. Er zijn dus al veel systeemeisen bekend wat in contrast staat met DSDM waarbij de initiële eisen erg algemeen zijn opgesteld en gedurende het project steeds verder worden uitgewerkt.

Omdat JJG zich voornamelijk bezig houdt met het ontwikkelen van het product zal ik met betrekking tot de projectinrichting wel mijn toevlucht nemen tot een aantal technieken uit andere methoden, waaronder DSDM. Dit wordt in de volgende paragraaf nader toegelicht.

### **3.3 Technieken**

De te hanteren technieken zullen in deze paragraaf per fase worden besproken. Alternatieven van de gekozen technieken worden ook toegelicht waarna de motivatie voor de gekozen techniek wordt gegeven.

### 3.3.1 Projectoriëntatie en projectorganisatie

Deze paragraaf beschrijft de technieken die zullen worden gehanteerd bij de volgende werkzaamheden.

- Productgrootte inschatten
- Project managen
- Planning opstellen

#### *Productgrootte inschatten*

Om alle werkzaamheden in kaart te brengen zal er een Work Breakdown Structure (WBS) worden opgesteld. Elke activiteit die uit de WBS komt zal worden ingeschat op ontwikkeltijd. Hiervoor heb ik gekozen uit de volgende technieken.

- *Delphi (Boehm)*

Bij deze techniek schatten de groepsleden de grootte van verschillende producten in. Vervolgens worden gemiddelden berekend en er wordt gekeken waar grote verschillen zitten. Over de verschillen volgt een discussie totdat alle groepsleden het eens zijn met een schatting. Op deze manier worden er zinvolle schattingen gemaakt die door de hele groep gedragen worden. De methode is zeer geschikt als er weinig tot geen historische data voor handen is.

- *Fuzzy-Logic (Putnam)*

Bij deze techniek wordt er uitgegaan van beschikbare historische data. Er wordt gekeken hoeveel tijd de verschillende producten van project in het verleden kostten. Deze worden in een matrix gezet waaruit duidelijk wordt welke producten qua grootte vergelijkbaar zijn met het huidige product. De methode zorgt voor een grove schatting van de hoofdproducten en wordt gebruikt als er een beperkte hoeveelheid historische data beschikbaar is.

- *Standaard Componenten (Putnam)*

Deze techniek is gebaseerd op de beschikbaarheid van historische data, maar richt zich op de deelproducten van een hoofdproduct. Van de verschillende deelproducten wordt gekeken wat de ondergrens en wat de bovengrens is. Vervolgens wordt de volgende formule gehanteerd.

$$\text{ontwikkeltijd} = (\text{ondergrens} + (4 \times \text{geschatte grootte}) + \text{bovengrens}) / 6$$

Deze techniek is erg geschikt als er over veel historische data beschikt kan worden.

*fig. 3.3 Productgrootte inschattingstechnieken*

De opdracht bevat veel elementen die ik in voorgaande projecten al een keer heb uitgevoerd. Er is dus genoeg historische data beschikbaar waarop ik mijn inschatting kan baseren. Daarom is de techniek die ik heb gekozen, Standaard Componenten van Putnam voor dit project uitermate geschikt.

Delphi legt de nadruk op de inschattingen van verschillende groepsleden wat bij deze opdracht niet aan de orde is. Daarnaast is er historische data beschikbaar die ik bij het hanteren van deze techniek niet zou gebruiken.

Hoewel Fuzzy-Logic ook uitgaat van historische data gebeurt dit slechts op beperkte schaal. Alleen de hoofdproducten worden van een inschatting voorzien waardoor nog veel onbekend blijft. Een prima methode als er slechts een beperkte hoeveelheid historische data beschikbaar is maar omdat die voor mijn project wel beschikbaar is, is de techniek ongeschikt voor deze opdracht.

De inschatting van ontwikkeltijden zal ik dus berekenen met de Standaard Componenten formule in bovenstaand kader.

### *Projectmanagement*

Voor elk van deze ontwikkeltijden zal gedurende het project worden bijgehouden of ze nog kloppen met de werkelijkheid. Hiervoor wordt een schema opgesteld waaruit duidelijk wordt hoeveel tijd er voor een taak gepland staat, hoeveel tijd reeds aan een taak is besteed en hoeveel tijd er nog aan een taak te besteden is. Hierin kan ook worden aangegeven dat er meer of minder tijd is besteed dan van te voren was ingeschat. Een en ander wordt toegelicht met het volgende voorbeeld.

activiteit	gepland	besteed	te besteden
definiëren klassen	1	1	0
opstellen klassediagram	5	5	0
opstellen usecases	10	4	8
opstellen usecasediagram	3	0	3
opstellen sequentiediagrammen	8	0	9
<b>Totaal</b>	27	10	20

Uit het schema blijkt dat ik de eerste twee activiteiten hebben afgerond (de kolom 'te besteden' staat op 0). Ik ben nu bezig met het opstellen van usecases maar na 4 tijdseenheden schat ik dat deze activiteit niet 10 maar 12 tijdseenheden in beslag zal nemen. In de kolom 'te besteden' vul ik in plaats van 6 (10 gepland - 4 besteed) nu 8 in (12 gepland - 4 besteed). Op basis van deze kennis weet ik dat ook de laatste taak een tijdseenheid zal uitlopen waardoor de kolom 'te besteden' bij de laatste taak 9 wordt. Door de waarden van de kolommen 'besteed' en 'te besteden' bij elkaar op te tellen en te vergelijken met de waarden uit de kolom 'gepland' kan ik zien dat ik nu 3 tijdseenheden achterloop op schema. Van dit schema wordt een grafiek gemaakt die het geplande verloop van tijd tegen het werkelijke verloop van tijd uitzet.

*fig. 3.4 Projectmanagement techniek*

De grafiek geeft een overzichtelijk beeld van de status van het project. Doordat ik in een vroeg stadium weet of er wijzigingen in de planning gemaakt moeten worden kan ik dit ook in een vroeg stadium naar de opdrachtgever communiceren zodat er rekening mee gehouden kan worden.

#### *Planning opstellen*

Bij het kiezen voor een techniek voor het opstellen van de planning heb ik de volgende technieken beschouwd.

- *Pert diagram*  
 De verschillende activiteiten van het project worden tegen de tijd uitgezet, waarbij de nadruk wordt gelegd op de onderlinge afhankelijkheid van de deeltaken.
- *Gantt chart*  
 Bij deze techniek worden de activiteiten ook tegen de tijd uitgezet maar ligt de nadruk op het tijdsverloop van de activiteiten.

*fig. 3.5 Planningstechnieken*

Het verloop van het project is vrij logisch opgebouwd: onderzoek → ontwerp → ontwikkeling → overdracht. De werkzaamheden die moeten worden uitgevoerd volgen elkaar ook logisch op.

De onderlinge afhankelijkheid is voor mij daarom erg duidelijk. Daarom heb ik gekozen voor de Gantt chart. Deze techniek legt de nadruk op het tijdsverloop van de activiteiten. Aangezien de duur van het project vastligt, is de tijd die voor bepaalde activiteiten staat belangrijker dan de onderlinge afhankelijkheid die voor mij toch al duidelijk is.

In deze fase zal ik ook vraaggesprekken met de opdrachtgever voeren. Op basis van deze vraaggesprekken stel ik de userneeds (JJG, strategy plane) en de siteobjectives (JJG, strategy plane, in dit geval is het product een applicatie, niet zozeer een site maar ik hou de terminologie van JJG aan) op.

### 3.3.2 Voorbereiding op ontwerp

Deze paragraaf beschrijft de technieken die zullen worden gehanteerd bij de volgende werkzaamheden.

- Doelgroep omschrijven
- Systeemeisen opstellen

#### *Doelgroep omschrijven*

In deze fase beschrijf ik de doelgroep met behulp van de volgende variabelen.

- *Demografisch*

Hierbij wordt beschreven:

- Kennisniveau
- Grootte van het bedrijf
- Leeftijd

- *Geografisch*

Hierbij wordt beschreven:

- Land
- Regio

- *Gedragmatig*

Hierbij wordt beschreven:

- Mate van gebruik
- Technologische gerichtheid
- Loyaliteit

*fig. 3.6 Doelgroep variabelen*

De informatie hiervoor put ik uit de vraaggesprekken met de opdrachtgever. Daarnaast zal ik in deze fase mensen uit de doelgroep benaderen om gedurende het project als testpersoon te functioneren. Via e-mail zal direct een aantal vragen worden gesteld die bij de doelgroepanalyse zullen worden gebruikt.

Na een algemene beschrijving zal ik een tweetal persona's ontwikkelen die de doelgroep vertegenwoordigen.

- *Persona's*

Persona's zijn fictieve personen die de doelgroep vertegenwoordigen. Een persona heeft grof gezien twee doelstellingen.

- De eerste doelstelling heeft betrekking op de eigenschappen van de doelgroep die kunnen worden doorvertaald naar (functionele) systeemeisen of eisen met betrekking tot het ontwerp. Als een gebruiker bijvoorbeeld moeite heeft met het uitvoeren van bepaalde (technische) taken dan zou dat doorvertaald kunnen worden naar een helpfunctie die de gebruiker van de benodigde informatie voorziet.
- De tweede doelstelling heeft te maken met het levendig houden van het project. Door persona's op te stellen en deze te behandelen als echte mensen krijgt het project meer kleur en is het als ontwikkelaar makkelijker je in te leven in de doelgroep. Door 'droge' gegevens van doelgroeponderzoek een naam en gezicht te geven houdt je als ontwikkelaar gemakkelijker de doelgroep in het achterhoofd bij het ontwerpen en ontwikkelen van het product. Een voorbeeld hiervan is bijvoorbeeld het beschrijven van de gezinssituatie. Dat gegeven hoeft niet per sé informatie te geven over hoe de applicatie beter ontwikkeld kan worden maar zorgt er wel voor dat de persona meer gaat leven en als persoon wordt gezien.

*fig. 3.7 Persona's*

Beide persona's zal ik gebruiken als mentaal model dat aan de basis staat voor elke beslissing die ik neem die van invloed is op de wijze waarop de gebruiker het product gebruikt. Bij de persona's is onderscheid te maken tussen de primaire persona en de secundaire persona. Elke persona representeert een deel van de doelgroep.



Om de sfeer die de applicatie moet uitstralen weer te geven zal ik een moodboard maken. Dit moodboard is gebaseerd op de doelgroepomschrijving en de sfeer van het Content Management Systeem (CMS) dat Wend gebruikt. Veel klanten van Wend maken al gebruik van dit CMS, zodat dezelfde stijl als vertrouwd door gebruikers zal worden ervaren.

Door deze twee zaken te combineren krijg ik een goede indruk van de richting waarin ik moet zoeken om tot een geslaagd ontwerp te komen.

### *Systeemeisen opstellen*

Tevens zal ik op basis van de vraaggesprekken de systeemeisen (JJG, scope plane) opstellen. Deze systeemeisen moeten van prioriteit worden voorzien om te kunnen bepalen of de applicatie alle benodigde functionaliteit bevat. Hoewel Jesse James Garrett in zijn boek het prioriteren van systeemeisen benadrukt biedt hij hier geen concrete handvatten voor. Ik heb daarom technieken beschouwd uit andere methoden.

- *MoSCoW (Dynamic System Development Method)*

Er zijn veel verschillende definities voor de betekenis van de hoofdletters in MoSCoW, maar ze komen allemaal op hetzelfde neer. De definitie die ik zal hanteren is de volgende.

- **Must have**, essentiële eisen, zonder deze eisen is het systeem waardeloos.
- **Should have**, belangrijke eisen, maar andere oplossingen die ook aan de eis voldoen mogen ook.
- **Could have**, minder belangrijke eisen die toegevoegde waarde hebben maar slechts als extra dienen.
- **Won't have but would like in the future**, de minst belangrijke eisen waarvan de ontwikkeling nu niet in de planning wordt opgenomen, maar die wel van nut zijn.

- *Basis, Comfort, Luxe (Iterative Application Development)*

Bij deze techniek worden de systeemeisen in drie categorieën verdeeld.

- Basis
- Comfort
- Luxe

*fig. 3.8 Prioritering van systeemeisen*

Ik heb hierbij gekozen voor de MoSCoW techniek omdat meer gespecificeerd wordt weergegeven waar de categorieën voor staan. Daarnaast wordt er concreet aangegeven welke systeemeisen er in moeten (Must have) en welke systeemeisen er sowieso niet in komen (Won't have). Met name deze laatste categorie is voor dit project goed te gebruiken. De applicatie heeft

veel potentie en is gemakkelijk uit te breiden. Het is daarom belangrijk om het project in te kaderen maar ook om goede ideeën die de applicatie nu niet zullen gaan halen wél op te schrijven zodat het systeem later met die functionaliteiten uitgebreid kan worden.

De systeemeisen worden ter goedkeuring aan de opdrachtgever voorgelegd. Het prioriteren van systeemeisen zorgt ervoor dat zowel de opdrachtgever als ik duidelijk voor ogen hebben hoe de applicatie gaat werken, welke mogelijkheden het systeem zal bieden en dat de mate van importantie van de verschillende onderdelen duidelijk is.

### 3.3.3 Technisch ontwerp

Bij het technisch ontwerpen van applicaties wordt veel gebruik gemaakt van Unified Modeling Language (UML). UML kent verschillende technieken om verschillende aspecten van het technisch ontwerp weer te geven. Daarnaast zijn er nog andere technieken die hiervoor gebruikt kunnen worden die niet onder UML vallen. Ik heb de volgende technieken beschouwd.

- *Klassen en klassediagram (UML)*

Met het bepalen van klassen worden concrete elementen en eigenschappen van het systeem vastgelegd. Door ze daarna in een diagram te plaatsen wordt de onderlinge relatie duidelijk. Een klassediagram zorgt voor structuur en overzicht.

- *Usecases en usecasediagram (UML)*

Om de verschillende mogelijkheden die de gebruiker heeft in beeld te brengen en toe te lichten worden usecases gemaakt. Het diagram geeft wederom inzicht in onderlinge relaties maar ook welke actoren (gebruikers, beheerders etc.) over bepaalde mogelijkheden beschikken.

- *Sequentiediagrammen (UML)*

Sequentiediagrammen beschrijven hoe de verschillende functionaliteiten technisch in elkaar steken. Er wordt beschreven welke objecten worden aangesproken en hoe de flow van informatie zich zal gedragen. Hierbij wordt gebruik gemaakt van zowel reeds opgestelde usecases als van reeds opgestelde klassen. Sequentiediagrammen koppelen deze twee onderdelen aan elkaar. Bij sequentiediagrammen ligt de nadruk in tegenstelling tot bij communicatiediagrammen op tijd.

- *Communicatiediagrammen (UML)*  
Een communicatiediagram geeft vrijwel dezelfde informatie weer als een sequentiediagram. Hier ligt de nadruk echter op de samenwerking tussen objecten.
- *Toestandsdiagrammen (UML)*  
Een toestandsdiagram beschrijft de toestanden die objecten kunnen aannemen en welke mogelijkheden daarbij horen. Daarnaast wordt weergegeven welke events welke transities veroorzaken. Een event is een gebeurtenis die een transitie (= toestandsovergang) veroorzaakt.
- *Activiteitsdiagrammen (UML)*  
Een activiteitsdiagram heeft enige overlap met een toestandsdiagram maar in plaats van het weergeven van objecten die bepalen handelingen kunnen uitvoeren worden alleen de handelingen (activiteiten) weergegeven. Zo wordt de onderlinge relatie en volgorde tussen de activiteiten duidelijk.
- *Programme Structure Diagrams (PSD)*  
Een PSD is bij uitstek geschikt om code vorm te geven. Het diagram kent verschillende vormen om de verschillende programmeer mogelijkheden als statements, if-statements en while loops weer te geven.

*fig. 3.9 UML en andere technieken voor het beschrijven van een technisch ontwerp*

Ik heb hierbij gekozen voor een combinatie van klassen en klassediagram, usecases en usecasediagram en sequentiediagram.

Een klassediagram geeft op globale wijze weer hoe de verschillende objecten met elkaar communiceren. Het diagram geeft structuur en overzicht. Door de relaties en objecten van tevoren vast te leggen heb ik de vrijheid om zonder de beperkingen van een programmeertaal het systeem te bedenken. Dit scheelt veel tijd tijdens het ontwikkelen, omdat het slechts een kwestie van toepassen is; de structuur ligt immers al vast.

Het gebruik van usecases stelt me in staat de reeds opgestelde systeemeisen te vertalen naar handelingen die binnen het systeem plaatsvinden. Door alle systeemeisen in usecasediagrammen te verwerken weet ik zeker dat er geen functionaliteit vergeten wordt. Daarnaast wordt duidelijk welke objecten voor verschillende handelingen moeten worden aangesproken. Usecasediagrammen vormen een tussenstap tussen de beschrijvende tekst van systeemeisen en technische werking en uitvoering van systeemeisen waardoor ik niet direct het

gehele technische ontwerp hoeft te overzien maar toch in staat ben om de systeemeisen aan het technisch ontwerp te koppelen.

Sequentiediagrammen zijn daarna de logische stap omdat deze diagrammen de stappen uit de usecases koppelen aan de klassen waar ze betrekking op hebben. Zoals hierboven vermeld zou hiervoor communicatiediagrammen ook een optie zijn. Aangezien sequentiediagrammen naast de onderlinge samenwerking ook duidelijk de volgorde van handelingen weergeeft heb ik gekozen voor sequentiediagrammen. Deze volgorde wordt ook in de usecases aangegeven waardoor de usecases beter op sequentiediagrammen aansluiten.

Toestandsdiagrammen kunnen ook zeer waardevol zijn, maar zijn gericht op een klein onderdeel van het systeem waarbij relaties met andere objecten niet direct duidelijk zijn. Bij het technisch ontwerp ben ik op zoek naar harmonie tussen de objecten, een solide basis van waaruit ik kan gaan programmeren. Het gebruik van toestandsdiagrammen is voor dit project te specifiek en past beter binnen complexe systemen waarbij objecten veel toestanden aan kunnen nemen.

Activiteitsdiagrammen beschrijven hoe verschillende handelingen worden uitgevoerd. Hier komen de objecten helemaal niet meer aan bod; slechts de activiteiten worden weergegeven. Dit is goed toepasbaar voor handelingen die te complex zijn voor sequentiediagrammen. Het weglaten van de objecten kan dan een stuk overzichtelijker werken, maar de handelingen binnen dit project zijn nog overzichtelijk weer te geven in sequentiediagrammen.

Met PSD's kun je als ontwikkelaar over de exacte werking van de code nadenken zonder dat je de code met code hoeft te beschrijven. Mijn ervaring is echter dat PSD's en uiteindelijke code vaak ver uit elkaar liggen waardoor de PSD's geen toegevoegde waarde hebben. De oorzaak hiervan is dat PSD's niet (gemakkelijk) te testen zijn. Het zijn slechts diagrammen die later worden omgezet naar code. Tools die PSD's omzetten naar 'echte' code hanteren hun eigen programmeer conventies wat eventuele wijzigingen in een later stadium lastiger maakt.

Op basis van het klassediagram wordt de database ontworpen. De klassen in het klassediagram vormen vrijwel rechtstreeks de tabellen in de database.

Ook wordt tijdens het technisch ontwerp een navigatiemodel of architecture diagram (JJG, structure plane) opgesteld, een boomstructuur van de verschillende mogelijkheden die de gebruiker heeft. Deze boomstructuur wordt afgeleid uit de usecases. Het navigatiemodel maakt duidelijk welke functionaliteiten waaronder vallen en hoe de gebruiker daar komt.

### *3.3.4 Visueel ontwerp van iconen en schermen in verschillende stijlen*

Allereerst zullen op basis van de systeemeisen, de usecases en het navigatiemodel een aantal wireframes (JJG, skeleton plane) worden gemaakt. De wireframes geven aan waar de verschillende elementen van de applicatie komen te staan. Hierbij vraag ik me bij het positioneren van de verschillende elementen op het scherm telkens af hoe de persona's erover denken, waar ze de verschillende elementen verwachten en hoe ze zullen reageren. Vervolgens worden deze wireframes uitgewerkt tot schetsen en uiteindelijk gedigitaliseerd.

Bij het gebruik van wireframes zal ik de verschillende onderdelen weergeven als blokken. Het maken van wireframes stelt me zo in staat om na te denken over de logica achter de positie van elementen zonder te worden beïnvloed door de uiterlijke kenmerken van een element.

Door de wireframes daarna naar schetsen te vertalen kan ik bepalen welke richting ik met de vormgeving op wil. Door dit eerst op papier te doen heb ik geen last van technische beperkingen of complexiteiten. Ook hierbij worden de persona's veelvuldig gebruikt.

### *3.3.5 Usability testen van iconen en schermen*

Om te onderzoeken of ik met het ontwerp op de goede weg ben zal ik het testen. Daarbij wordt er gekeken naar in hoeverre iconen en schermen bruikbaar zijn en aansluiten bij de doelgroep.

Tijdens de eerste fase van usability testing zal de variabele 'herkenbaarheid' worden geoperationaliseerd. De applicatie bestaat op dit moment nog alleen uit een aantal schermen, afbeeldingen en iconen. Deze iconen worden aan de testpersoon voorgelegd met de vraag of hij er een associatie of betekenis bij kan geven. Daarna worden de iconen in de context geplaatst waarna de gebruiker gevraagd wordt een aantal taken uit te voeren.

De test wordt via internet uitgevoerd. Er is dus geen direct contact tussen ontwikkelaar en testpersoon.

### *3.3.6 Visueel ontwerp uitwerken*

Na het testen zal de stijl worden gekozen die het beste uit de bus komt. Deze wordt vervolgens verder uitgewerkt. Alle mogelijke schermen zullen tijdens deze fase worden vormgegeven.

Gedurende deze fase worden ook templates voor de videospelers ontwikkeld. Er zullen verschillende videospelers worden ontworpen. Dit zijn de stijlen waaruit de gebruiker zal kunnen kiezen.

### *3.3.7 Ontwikkelen van de applicatie*

Tijdens de eerste ontwikkelfase wordt het systeem gebouwd aan de hand van het technisch en visueel ontwerp uit eerdere fases. Tijdens de ontwerpfases is alles uitgedacht, gemodelleerd en vormgegeven zodat deze fase slechts een kwestie van toepassen is.

### *3.3.8 Usability testen van de gehele applicatie*

Voor het testen van de usability van de gehele applicatie zullen de volgende variabelen worden geoperationaliseerd.

- Efficiëntie
- Effectiviteit
- Tevredenheid
- Leerbaarheid

Voor de te operationaliseren variabelen worden testtaken geschreven. De gebruiker moet een taak uitvoeren en met behulp van verschillende technieken en methoden wordt gemeten of de applicatie voldoet aan de verwachtingen van de gebruiker.

Bij deze test zal ik aanwezig zijn en gedurende de uitvoering zaken observeren en noteren.

Effectiviteit is een ja/nee kwestie. De gebruiker is in staat de taak wel of niet uit te voeren. Efficiëntie wordt met behulp van het meten van tijd en aantal keer klikken vastgesteld. Er wordt een standaardwaarde gesteld en het uiteindelijke resultaat wordt vergeleken met die standaardwaarde. Door vergelijkbare taken op te stellen is te meten of de gebruiker leert van de applicatie, er wordt gemeten of de gebruiker minder tijd nodig heeft voor het uitvoeren van de taak dan de eerste keer. De tevredenheid kan worden gemeten door de gebruikers achteraf vragen te stellen over het gebruik van de applicatie.

Bij het uitvoeren van de usability test worden de volgende technieken gebruikt.

- *Cognitive walkthrough*  
De gebruiker wordt gewezen op het te volgen pad en er wordt gekeken of problemen zich voor doen.
- *Think Aloud*  
Ik zal de gebruikers bij het uitvoeren van de tests vragen hardop te denken. Hierdoor krijg ik inzicht in de zaken die zich in het hoofd van de gebruiker afspelen en kan ik beter beoordelen waar de gebruiker moeite mee heeft of wat de gebruiker juist gemakkelijk en handig vindt werken.
- *Software Usability Measurement Inventory (SUMI)*  
Dit is een gestandaardiseerde vragenlijst die al ontelbare keren in de praktijk is toegepast en inzicht geeft in de effectiviteit, tevredenheid en leerbaarheid van een applicatie.

fig. 3.10 Usability meettechnieken

Naast de cognitive walkthrough wordt de gebruiker bij verschillende testtaken ook 'losgelaten'. Hij krijgt een taak te volbrengen en moet zelf zijn weg zoeken.

Bij het ontwikkelen en beoordelen van de applicatie zal ook rekening worden gehouden met Nielsen's usability heuristics.

### 3.3.9 Herzien van de applicatie

De bevindingen van de usability testing fase spelen in deze tweede ontwikkelfase de grootste rol. De resultaten zullen worden nagelopen en de applicatie dient te worden herzien op de punten waar de gebruiker moeite mee heeft. Daarnaast zal de applicatie definitief worden afgemaakt. Hiervoor worden geen speciale technieken gebruikt.

Tot slot wordt de huidige werking van het systeem beschreven voor eventuele vervolgontwikkelaars.

## 3.4 Op te leveren producten

Gedurende het project zal ik de volgende producten opleveren.

- Startdocument
- Voorbereidend ontwerprapport
- Technisch ontwerprapport
- Visueel ontwerprapport van iconen en schermen
- Testplan voor fase 'Usability testen van iconen en schermen'
- Testresultaten en verwerking van fase 'Usability testen van iconen en schermen'
- Visueel ontwerprapport van alle schermen van de applicatie
- Testplan voor fase 'Usability testen de gehele applicatie'
- Testresultaten en verwerking van fase 'Usability testen de gehele applicatie'
- Applicatie
- Overdrachtsverslag

#### *3.4.1 Startdocument*

Dit rapport vormt de basis van het project. Alle relevante informatie, afspraken en planning worden hierin opgesteld. Het is een dynamisch document wat wil zeggen dat er gedurende het project zaken gewijzigd kunnen worden. Dit geldt met name voor de planning. Het document bevat de volgende elementen.

- Probleemstelling
- Doelstelling
- Werkzaamheden
- Methode en technieken
- Rapportage techniek
- Lijst met op te leveren producten
- Doorloopkalender
- Planning
- Urenbegroting



### *3.4.2 Voorbereidend ontwerprapport*

Voordat ik kan beginnen met ontwerpen moet ik weten wat ik ga ontwerpen en voor wie ik ga ontwerpen. In dit rapport worden die twee zaken vastgelegd. Met name de systeemeisen zijn erg belangrijk, de opdrachtgever moet deze goedkeuren zodat zowel de opdrachtgever als ik weten wat ons te wachten staat en wat het resultaat van het project zal zijn. Daarnaast speelt ook de doelgroepomschrijving een belangrijke rol. Dit zijn immers de eindgebruikers van het product. De volgende elementen komen in het Voorbereidend ontwerprapport.

- Userneeds
- Siteobjectives
- Systeemeisen
- Doelgroepomschrijving
- Persona's
- Moodboard

### *3.4.3 Technisch ontwerprapport*

In dit document worden alle gemodelleerde aspecten van het systeem geplaatst. De relatie tussen de verschillende diagrammen en schema's wordt duidelijk en het geheel geeft een helder overzicht van wat er ontworpen moet worden en hoe er ontworpen moet worden. Het document bevat de volgende onderdelen.

- Klassen
- Klassediagram
- Usecases
- Usecasediagram
- Sequentiediagrammen
- Database ontwerp
- Navigatiemodel of architecture diagram
- Bestandsstructuur

#### *3.4.4 Visueel ontwerprapport van iconen en schermen*

In dit document worden de verschillende onderwerpen van het ontwerpproces opgenomen. Tevens wordt de feedback van de opdrachtgever vermeld. Het visueel ontwerp bevat de volgende elementen.

- Wireframes
- Schetsen
- Iconen en schermen
- Gegeven feedback op de drie stappen

#### *3.4.5 Testplan voor fase 'Usability testen van iconen en schermen'*

Dit plan bevat alle informatie om het usability testen in goede banen te leiden. Er worden testtaken beschreven en vragenlijsten opgesteld. Bij elke testtaak wordt beschreven welke variabele wordt geoperationaliseerd en op welke wijze dat dient te gebeuren. De onderdelen van het testplan zijn als volgt.

- Doel
- Beschrijving van testtaken
- Vragenlijsten

#### *3.4.6 Testresultaten en verwerking van fase 'Usability testen van iconen en schermen'*

Dit document geeft alle resultaten weer, vervolgens worden deze resultaten verwerkt en worden er conclusies aan verbonden. Deze lijst met aanbevelingen reflecteert de zaken die aan het ontwerp van de applicatie moeten worden gewijzigd. De volgende onderdelen komen aan bod.

- Ruwe testresultaten
- Verwerkte testresultaten
- Conclusie en lijst met aanbevelingen

### *3.4.7 Visueel ontwerprapport van alle schermen van de applicatie*

In dit document worden de verschillende onderwerpen van het ontwerpproces opgenomen. Tevens wordt de feedback van de opdrachtgever vermeld en welke wijzigingen als gevolg van die feedback zijn doorgevoerd. Het visueel ontwerp bevat de volgende elementen.

- Wireframes
- Schetsen
- Uiteindelijke vormgeving
- Gegeven feedback op de drie stappen

### *3.4.8 Testplan voor fase 'Usability testen van de gehele applicatie'*

Dit plan bevat alle informatie om het usability testen in goede banen te leiden. Er worden testtaken beschreven, vragenlijsten opgesteld en registratieformulieren gegeven. Bij elke testtaak wordt beschreven welke variabele wordt geoperationaliseerd en op welke wijze dat dient te gebeuren. De onderdelen van het testplan zijn als volgt.

- Doel
- Technieken en hulpmiddelen
- Beschrijving van testtaken
- Vragenlijsten
- Registratieformulieren
- Planning van het testen

### *3.4.9 Testresultaten en verwerking van fase 'Usability testen van de gehele applicatie'*

Dit document geeft alle resultaten weer, vervolgens worden deze resultaten verwerkt en worden er conclusies aan verbonden. Deze lijst met conclusies reflecteert de zaken die aan het ontwerp van de applicatie moeten worden gewijzigd. De volgende onderdelen komen aan bod.

- Ruwe testresultaten
- Verwerkte testresultaten
- Conclusie en lijst met aanbevelingen

#### *3.4.10 Applicatie*

Aan het einde van het project zal ik de applicatie opleveren. Deze zal voldoen aan de systeemeisen die zijn gesteld in de voorbereidende ontwerpfase.

#### *3.4.11 Overdrachtsverslag*

Omdat naar verwachting het technisch ontwerp niet 100% aansluit bij de uiteindelijke applicatie zal dit verslag de applicatie nogmaals in beeld brengen. Hierbij worden dezelfde technieken gebruikt als tijdens het technisch ontwerp, het verslag kent dus ook dezelfde onderdelen.

## 4 PROJECTORIËNTATIE EN PROJECTORGANISATIE OPSTELLEN

Dit hoofdstuk bespreekt de wijze waarop de eerste fase 'Projectoriëntatie en projectorganisatie' is verlopen.

### 4.1 Probleem- en doelstelling opstellen

Bij het opstellen van de probleem- en doelstelling heb ik informatie gehaald uit de gesprekken met de opdrachtgever. Aanvankelijk was het lastig om iets concreets te formuleren omdat de opdrachtgever zich erg richtte op de applicatie en minder op de achterliggende gedachte. Na enkele gesprekken heb ik toch genoeg informatie weten te verzamelen om een degelijke probleemstelling te formuleren. De doelstelling slaat direct terug op de probleemstelling.

**“De doelstelling van het project is het mogelijk maken om een bredere doelgroep aan te spreken, een doelgroep die videospelers in de stijl van de rest van het product wenst.”**

*fig. 4.1 Doelstelling (gedeeltelijk)*

### 4.2 Urenbegroting opstellen

Om een degelijke inschatting te kunnen maken van de tijd die ik aan de verschillende activiteiten kwijt heb ik het project in fases verdeeld. Vervolgens heb ik per fase de werkzaamheden vastgesteld en zo een Work Breakdown Structure (WBS) opgesteld. Het opstellen van deze WBS gaf mij een goed inzicht in de hoeveelheid werk die voor me lag.

Na het opstellen van de WBS heb ik het aantal uren per activiteit ingeschat. Om tot een solide inschatting te komen heb ik gebruik gemaakt van de Standaard Componenten techniek van Putnam zoals beschreven in hoofdstuk 3 'Aanpak'. De Standaard Componenten techniek van Putnam hanteert de volgende formule.

$$\text{productgrootte} = (\text{ondergrens} + (4 \times \text{geschatte grootte}) + \text{bovengrens}) / 6$$

Om de productgrootte te kunnen berekenen heb ik naast het schatten van de productgrootte per activiteit de ondergrens en bovengrens bepaald. Het bepalen hiervan heb ik gebaseerd op

ervaring uit eerdere projecten. Dit leidde tot een compleet schema van alle werkzaamheden waaruit het project bestaat. Een gedeelte van dit schema wordt hieronder weergegeven.

activiteit	geschat	ondergrens	bovengrens	schatting
Definieren van klassen	2,0	1,0	3,0	2,0
Klassediagram opstellen	6,0	4,0	10,0	6,3
Usecases opstellen	8,0	6,0	14,0	8,7
Usecasediagram opstellen	2,0	1,0	2,0	1,8
Sequentiediagrammen opstellen	14,0	10,0	20,0	14,3
Database ontwerpen	2,0	1,0	3,0	2,0
Bestandstructuur opstellen	5,0	4,0	7,0	5,2
Navigatiemodel ontwerpen	1,5	1,5	4,0	1,9

*fig. 4.2 Ureninschatting (gedeeltelijk)*

### 4.3 Planning opstellen

De opgestelde urenbegroting heb ik gebruikt om de planning op te stellen. Eerst heb ik per fase de doorlooptijd bepaald zodat ik een globaal idee van de verhouding van de fases kreeg. Daarna heb ik per fase de activiteiten op doorlooptijd ingeschat. Al deze gegevens heb ik vervolgens in Gantt-charts geplaatst. Om het overzicht te bewaren heb ik per fase een Gantt-chart gemaakt zodat ik direct kan zien waar ik in het project sta.

Hoewel het opsplitsen van de activiteiten in verschillende Gantt-charts het overzicht ten goede komt kleeft er ook een nadeel aan omdat de fases elkaar op sommige punten overlappen. Het is dus belangrijk dat ik me niet beperk tot de Gantt chart van de fase waarop ik op dat moment aan werk, maar ook de andere Gantt-charts in de gaten houd.

Ondanks dit nadeel heb ik besloten verschillende Gantt-charts op te stellen. Het voordeel van overzicht weegt op tegen het nadeel van fase overlap.

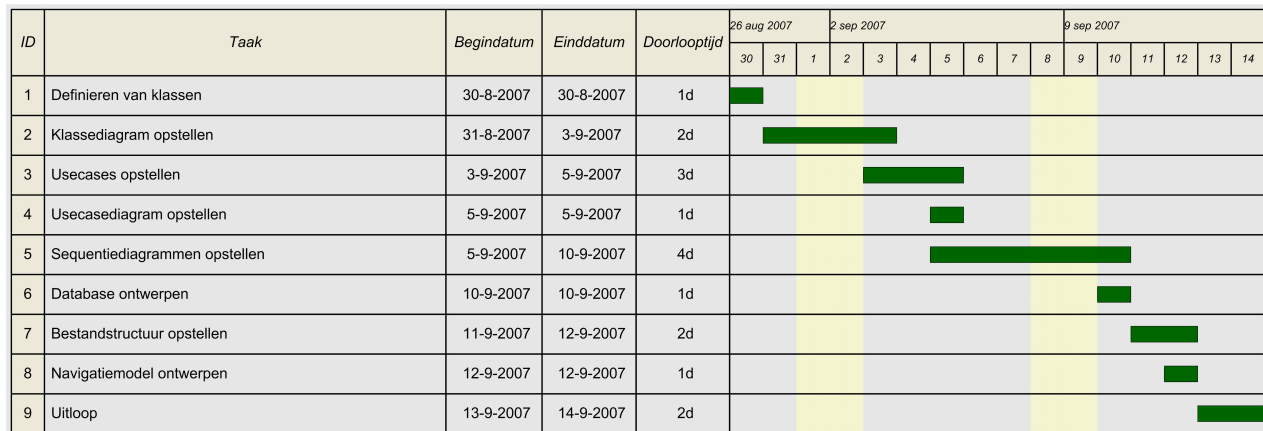


fig. 4.3 Gantt-chart van de fase 'Technisch ontwerp'

Bij het opstellen van de Gantt-charts heb ik bij elke fase een aantal dagen uitloop ingecalculleerd. Uit ervaring weet ik dat er altijd een aantal zaken anders lopen dan gedacht waardoor werkzaamheden meer tijd kunnen innemen dan ingepland. Door een buffer in te bouwen kan de uitloop enigszins worden opgevangen.

#### 4.4 Projectmanagement methode vastleggen

Om het inzicht in de status van het project te bewaren zal ik gedurende het hele project een schema bijhouden waarin wordt weergegeven hoeveel tijd ik besteed heb aan de verschillende werkzaamheden. Deze methode sluit goed aan op de eerder opgestelde WBS en Gantt-charts; het schema bestaat uit dezelfde lijst met werkzaamheden die is gebruikt bij de WBS en de Gantt-charts. Zoals uitgelegd in hoofdstuk 3 'Aanpak' kan op basis van dit schema een grafiek worden gemaakt die het verloop van het project weergeeft. Met deze grafiek weet ik op elk moment wat de status van het project is. Zo kan ik eventuele wijzigingen in de planning in een vroeg stadium opmerken en naar de opdrachtgever communiceren zodat hier rekening mee gehouden kan worden. Daarnaast is het belangrijk dat ik aan de opdrachtgever kan

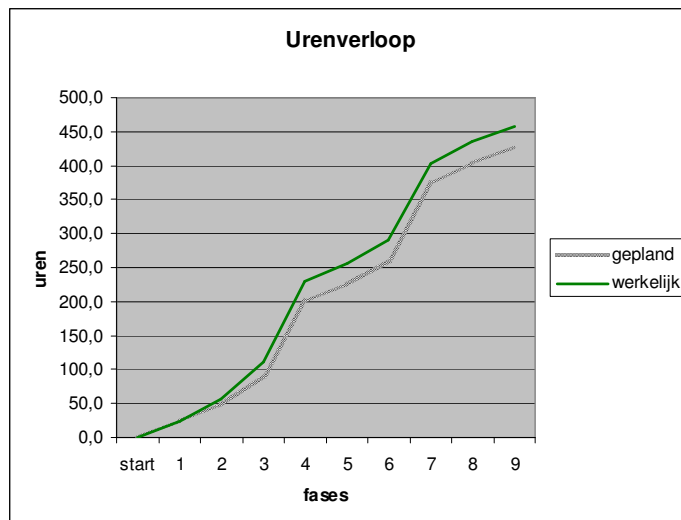


fig. 4.4 Urenverloop grafiek

aantonen dat als ik een bepaalde wijziging moet doorvoeren de deadline misschien niet meer gehaald kan worden. Met de grafiek kan ik dan duidelijk maken hoe het verloop van het project zal zijn.



## 5 VOORBEREIDEN OP ONTWERP

Dit hoofdstuk bespreekt de wijze waarop de fase ‘Voorbereiding op ontwerp’ is verlopen.

### 5.1 Userneeds en siteobjectives opstellen

Voor het opstellen van de userneeds en siteobjectives (JJG, strategy plane) heb ik een aantal gesprekken met de opdrachtgever gevoerd. Op basis van de informatie die uit deze gesprekken is voortgevloeid en de reeds opgestelde probleem- en doelstelling heb ik de userneeds en siteobjectives opgesteld.

**“De gebruikers van de applicatie willen video’s die op de website worden getoond zelf op eenvoudige wijze kunnen beheren.”**

*fig. 5.1 Userneeds (gedeeltelijk)*

### 5.2 Doelgroepomschrijving opstellen

Ook om de doelgroep te omschrijven heb ik informatie gehaald uit de gesprekken met de opdrachtgever. Daarnaast heb ik een lijst met vragen opgesteld en die aan de klanten van Wend voorgelegd. Deze vragenlijst heb ik ingedeeld in de categorieën ‘Woonomgeving’, ‘Werkomgeving’, ‘Vrije tijd’ en ‘Computer- en internetgebruik’. De antwoorden op de vragen uit deze categorieën geven mij een goed beeld van hoe de doelgroep eruit ziet. De vragen zijn zo opgesteld dat de antwoorden niet alleen informatie met een directe relatie tot het gebruik van de applicatie geven maar ook informatie over de persoonlijke leefomgeving en situatie van de doelgroep aan het licht komen. Antwoorden op vragen als ‘Wat doet u graag in uw vrije tijd?’ of

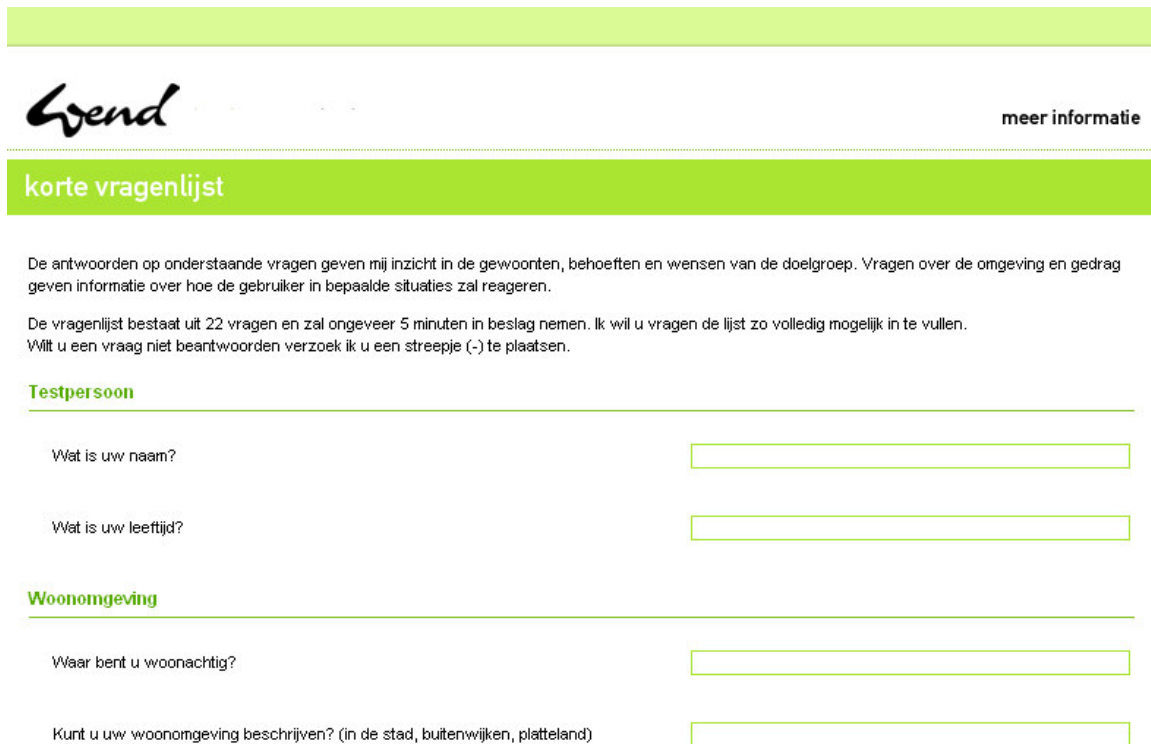
**“Het zijn middelgrote tot grote bedrijven die vaak beter opgeleide mensen tussen de 20 en 50 jaar in dienst hebben. Het budget is meer dan gemiddeld.”**

*fig. 5.2 Doelgroepomschrijving (gedeeltelijk)*

‘Doet u op regelmatige basis aan sport?’ kunnen me een beter inzicht verschaffen in hoe de doelgroep naar de applicatie zal kijken en hoe ze de applicatie zullen gebruiken. Hierbij kun je bijvoorbeeld denken aan eventuele ervaringen met reserveringssystemen die op tennisbanen en in fitnesscentra worden gebruikt.

Verschillende mensen hebben deze vragen beantwoord. Op basis van deze antwoorden heb ik een algemeen verhaal opgesteld die de doelgroep beschrijft. Daarna heb ik de traditionele geografische, demografische en gedragsmatige variabelen gebruikt om de doelgroep specifieker weer te geven. Omdat de doelgroep bestaat uit andere bedrijven heb ik de psychografische variabele achterwege gelaten; het gaat immers om een business-to-business situatie.

De klanten van Wend en andere testpersonen zijn via email benaderd. Hierbij heb ik overwogen om de vragenlijsten als bijlage bij de email te zetten, de vragen direct in de mail te stellen of de vragen online te zetten. Uiteindelijk heb ik gekozen voor het online plaatsen van alle onderzoeken en vragenlijsten. Als de vragen als bijlage zouden worden bijgesloten en mensen deze zouden beantwoorden en opslaan wordt het bestand standaard in een tijdelijke map geplaatst die vrij lastig is terug te vinden wat tot gevolg heeft dat mensen lastig de beantwoorde vragen terug kunnen mailen. Daarnaast denk ik dat het terugsturen van een mail een hogere drempel is dan het invullen van een vragenlijst op internet. De vragen direct in de mail stellen leek me ongeleefd, mensen worden dan direct overvallen met veel informatie en allerlei vragen. Daardoor bleef er één optie over: vragen en onderzoeken online plaatsen. Daarnaast kan ik online gemakkelijker veel informatie kwijt zonder dat er direct veel tekst getoond wordt.



**Wend** [meer informatie](#)

---

### korte vragenlijst

De antwoorden op onderstaande vragen geven mij inzicht in de gewoonten, behoeften en wensen van de doelgroep. Vragen over de omgeving en gedrag geven informatie over hoe de gebruiker in bepaalde situaties zal reageren.

De vragenlijst bestaat uit 22 vragen en zal ongeveer 5 minuten in beslag nemen. Ik wil u vragen de lijst zo volledig mogelijk in te vullen. Wilt u een vraag niet beantwoorden verzoek ik u een streepje (-) te plaatsen.

**Testpersoon**

Wat is uw naam?

Wat is uw leeftijd?

**Woonomgeving**

Waar bent u woonachtig?

Kunt u uw woonomgeving beschrijven? (in de stad, buitenwijken, platteland)

fig. 5.3 Vragenlijst die o.a. antwoord moet geven op woon- en werksituatie van de doelgroep (gedeeltelijk)

Voor alle vragenlijsten en onderzoeken naar de klanten van Wend en testpersonen geldt dat deze zijn geplaatst in een omgeving waarbij de stijl en visuele elementen van Wend's website ([www.wend.nl](http://www.wend.nl)) zijn aangehouden. Dit zorgt ervoor dat de mensen een vertrouwde omgeving zien en daardoor eerder de vragenlijsten zullen invullen. Voor de onderzoeken van dit project is een speciaal subdomein aangemaakt: <http://onderzoek.wend.nl>.

De resultaten van deze korte vragenlijst weerleggen de vermoedens omtrent het grote verschil in internetervaring en kennis. Het aantal uren dat de doelgroep per week gebruik maakt van de computer varieert van 20 tot 55 uur. Voor internetgebruik liggen de waarden tussen de 8 en 30 uur per week. Het gemiddelde computergebruik is 36,67 uur per week. Het gemiddelde internetgebruik is 20,5 uur. Er is wel verschil, maar niet zo groot als in eerste instantie uit de gesprekken met de opdrachtgever bleek.

Daarnaast is er een verschil in beroepuitoefening. Onder de doelgroep worden bijvoorbeeld zowel onderzoekers als senior systeemontwikkelaars gerekend. Ik heb daarom besloten om deze twee verschillen als uitgangspunten voor de persona's te nemen. Hier zal ik in paragraaf 5.4 'Persona's ontwikkelen' op terugkomen.

### 5.3 Moodboard ontwikkelen

De omschreven doelgroep heb ik als uitgangspunt gebruikt om een moodboard te ontwikkelen. Dit moodboard geeft de sfeer en stijl die de applicatie moet uitstralen weer. Omdat de doelgroep een gemiddeld kennisniveau heeft heb ik gekozen voor kenmerken als 'simpel', 'strak' en 'overzichtelijk' zonder veel poespas zodat direct duidelijk zal zijn welke mogelijkheden de gebruiker heeft. Ook de huisstijl van Wend is met deze kenmerken te omschrijven. Daarnaast heb ik de huisstijl kleuren (groen, wit, zwart en verschillende grijs tinten) van Wend erin verwerkt. In het moodboard heb ik een aantal pictogrammen gezet om aan te geven dat de doelgroep visueel is ingesteld.



fig. 5.4 Moodboard (gedeeltelijk)

## 5.4 Persona's ontwikkelen

Persona's zorgen ervoor dat de doelgroep altijd een 'top-of-mind' positie heeft bij het ontwerpen of ontwikkelen van een product. De informatie die nodig is voor het opstellen van de persona's heb ik geput uit de doelgroepomschrijving, antwoorden van de ondervraagde personen en gesprekken met de opdrachtgever.

Uit de doelgroepomschrijving bleek dat er onderscheid gemaakt kon worden in internetervaring. Het verschil in internetervaring was niet zo groot als aanvankelijk gedacht en besproken in de initiële gesprekken met de opdrachtgever zoals reeds vermeld in paragraaf 5.2

'Doelgroepomschrijving opstellen'. Dit heeft enkele wijzigingen in de geplande werkzaamheden teweeggebracht. In hoofdstuk 7 'Visueel ontwerp van iconen en schermen opstellen' kom ik daar op terug.

Aanvankelijk was het idee om het onderscheid tussen de primaire en secundaire persona te baseren op dit verschil in internetervaring. Nu dit verschil niet bijzonder groot blijkt te zijn heb ik het onderscheid in persona's mede gebaseerd op het vermoeden dat de primaire persona zich meer op de vormgeving zou richten dan de secundaire persona.

Dit vermoeden werd bevestigd nadat ik de ontwikkelde persona's opnieuw aan de doelgroep voorlegde. De meeste ondervraagden konden duidelijk een persona aanwijzen die goed aansloot op hun eigen persoon, waarbij ze zowel voor Miriam, de primaire persona als voor Wilfried, de secundaire persona kozen.

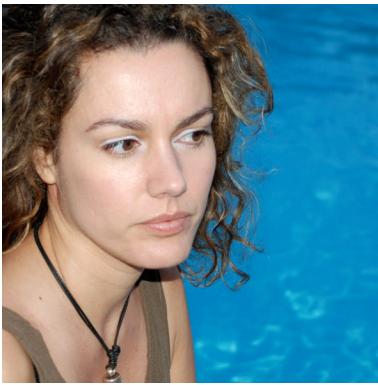
Bij deze tweede vragenlijst heb ik ook gevraagd welke elementen van de persona's ze in zichzelf herkenden. Daaruit bleek dat ik Wilfried te technisch had geschetst en het contrast tussen vormgeving en functionele werking te groot was. Daarop heb ik zijn quote enigszins genuanceerd. Wilfried's quote "Ik wil snel resultaat zien. Daarbij is de vormgeving voor mij minder belangrijk, zolang het werkt zoals het zou moeten werken" werd "Ik wil snel resultaat zien. Het systeem moet werken. Daarbij is vormgeving een pré, maar geen must."

**"Ik wil snel resultaat zien. Het systeem moet werken. Daarbij is vormgeving een pré, maar geen must."**

*fig. 5.5 Persona quote van Wilfried*

Miriam heeft gemiddelde internetervaring; Wilfried is zeer internet ervaren. De vormgeving en werking van de applicatie zal ik voornamelijk op Miriam richten, die minder internetervaren is dan Wilfried. Als de applicatie voor Miriam duidelijk is de kans groot dat ook Wilfried de weg kan vinden terwijl dat andersom geenszins het geval hoeft te zijn.

De persona's geven me inzicht in de leefsituatie van de doelgroep, werkomgeving en loopbaan worden omschreven, internet- en computerervaring en veelbezochte sites worden weergegeven maar ook de gezinssituatie en hobby's worden vermeld. Miriam en Wilfried geven dus functionele informatie die direct kan worden doorvertaald naar eisen voor de applicatie, maar ook niet-functionele informatie die niets met de applicatie te maken hoeft te hebben maar de persona's menselijker en 'echter' maakt waardoor de doelgroep tot leven wordt gewekt. Tijdens het ontwerpen en ontwikkelen houdt ik dan rekening met mensen en niet met een droog stuk tekst dat enkel functionele aspecten belicht.



Dit is Miriam Verrevelde. Miriam is 31 en woont in Gouda. Ze is werkzaam bij DigEdu, een middelgroot bureau dat digitale concepten bedenkt en uitwerkt voor de onderwijssector. Miriam doet onderzoek en adviseert de conceptontwikkelaars bij het bedenken van concept. Binnen het bedrijf heerst een informele sfeer. Haar afdeling luncht altijd gezamenlijk waar uitgebreid de tijd voor genomen wordt. Ze houdt van haar werk en maakt soms lange dagen.

Miriam beschouwt zichzelf als een gemiddeld ervaren internetgebruiker. Ze is visueel ingesteld en vindt het belangrijk dat websites duidelijk en overzichtelijk zijn, waarbij de vormgeving een belangrijke rol speelt. Hoewel ze af en toe wat videofragmenten op internet bekijkt, gebruikt ze internet

vooral om dingen op te zoeken, waar ze met name Google voor gebruikt. Ook houdt ze graag het nieuws in de gaten; de site nu.nl is dan ook een veelbezochte website.

**“Ik hou ervan als ik in één oogopslag kan zien wat mijn opties zijn. Hoe iets eruit ziet speelt daarbij een belangrijke rol.”**

*fig. 5.6 Persona Miriam (gedeeltelijk)*

Gedurende de ontwikkeling van de applicatie zal ik de persona's als mentaal model bij elke stap betrekken en me afvragen hoe Miriam en Wilfried op bepaalde keuzes of oplossingen zullen reageren.

## 5.5 Systeemeisen opstellen

Naast gesprekken over de doelgroep heb ik met de opdrachtgever ook gepraat over de applicatie. Bij dit project zijn vanuit Wend twee begeleiders betrokken. Om tot een degelijke lijst van systeemeisen te komen heb ik beide begeleiders geïnterviewd en deze sessies als brainstorm behandeld waarbij ik alle ideeën en mogelijkheden heb opgeschreven. Bij het interviewen wilde ik de begeleiders zowel een eigen ongeremde visie op de applicatie laten geven als ze in te laten spelen op elkaars ideeën. Daarom heb ik voor de ongeremde visie ze allebei apart geïnterviewd. Na deze sessies heb ik de eisen ingedeeld in 'Modules', 'Technische systeemeisen' en 'Visuele systeemeisen'. Ik heb ze geprioriteerd met behulp van de MoSCoW techniek. Daarna was mijn bedoeling dat ik met beide begeleiders nogmaals om de tafel zou gaan zitten om de systeemeisen door te spreken en vast te leggen. Door deze sessie wel met beide begeleiders te houden zouden ze op elkaars ideeën kunnen inspelen. Helaas had een van de begeleiders het op het moment dat deze sessie gepland stond te druk en was ook de komende tijd geen optie. Omdat ik niet veel kon doen zonder dat de systeemeisen vaststonden heb ik besloten de systeemeisen met de andere begeleider door te lopen en vast te leggen. Zoals altijd wilde de opdrachtgever meer dan ik had voorgesteld. Voor mij was een belangrijk argument dat in tegenstelling tot andere interne projecten er een periode is vastgesteld waarin de opdracht moet worden voltooid. De systeemeisen kunnen daarom niet veel worden uitgebreid omdat ik dan in tijdnood kom. Na enig onderhandelen hebben we de systeemeisen zo kunnen vastleggen dat we er allebei tevreden over waren. Deze systeemeisen zijn vervolgens ook aan de andere begeleider voorgelegd.

- Videospeler vormgeven (**M**)
  - Stijl kiezen (**M**)
  - Formaat instellen (**M**)
  - Kleuren kiezen (**M**)
  - Bedieningsknoppen selecteren (**M**)
  - Positie bedieningsknoppen instellen (**W**)
  - Afbeeldingen selecteren voor bedieningsknoppen (**W**)
  - SWF bestand of afbeelding selecteren die vóór het videofragment getoond wordt (**S**)
  - Afbeelding selecteren die tijdens het videofragment in een hoek van de videospeler getoond wordt (**S**)

*fig. 5.7 Systeemeisen (gedeeltelijk) met behulp van de MoSCoW techniek*

## 6 TECHNISCH ONTWERP OPSTELLEN

Dit hoofdstuk beschrijft de gang van zaken gedurende het technisch ontwerp.

### 6.1 (Kandidaat)klassen opstellen

Voor het technisch ontwerp ben ik begonnen met het klassediagram. Om tot een degelijk diagram te komen moet er bepaald worden wat de klassen van het systeem zijn. Het boek 'Praktisch UML' van Jos Warmer en Anneke Kleppe beschrijft twee technieken om tot geschikte klassen te komen. De eerste manier is het onderstrepen van alle zelfstandige naamwoorden in de tekst van de probleembeschrijving en die als kandidaatklassen te beschouwen. De tweede manier is brainstormen. Nadat ik beide technieken had uitgevoerd was ik er niet van overtuigd dat de lijst met kandidaatklassen compleet was. Daarom heb ik de trend van onderstrepen doorgezet in de doelstelling en de reeds opgestelde en goedgekeurde systeemeisen. Dit leverde een grote hoeveelheid kandidaatklassen op waarna ik naar mijn idee wél een complete lijst van alle kandidaatklassen had.

Daarna restte mij de taak te bepalen welke kandidaatklassen daadwerkelijk klassen zouden worden. Door de kandidaatklassen te bestempelen als 'te vaag', 'attribuut' (van klasse), 'te specifiek', 'subklasse' of 'klasse' kreeg ik een overzicht van alle klassen die in het klassediagram komen. Hierbij wordt per kandidaatklasse een aantal vragen beantwoordt. Als de volgende vragen allemaal met 'ja' kunnen worden beantwoordt kan de kandidaatklasse als 'klasse' worden bestempeld.

- Speelt de kandidaatklasse een zelfstandige rol in het probleemdomen?
- Willen we iets van de kandidaatklasse weten of willen we dat hij iets voor mij doet?
- Heeft de kandidaatklasse een duidelijke verantwoordelijkheid in het systeem?

Een gedeelte van de lijst met kandidaatklassen en de beoordeling wordt hieronder weergegeven.

Kandidaatklasse	Beoordeling
Video	klasse
Afspeellijst	klasse
Gebruiker	klasse
Beheerder	klasse
statistieken	te vaag
videobestand	attribuut
Stijl	klasse
code	te vaag



schijfruimte	te specifiek
stijleigenschappen	te vaag
<b>Kleurenpalet</b>	klasse
<i>Introbeeld</i>	subklasse van VideospelerBeeld
<i>Hoekafbeelding</i>	subklasse van VideospelerBeeld
<b>VideospelerBeeld</b>	klasse

fig. 6.1 Kandidaatklassen (gedeeltelijk)

## 6.2 Klassediagram opstellen

Nadat de klassen bekend waren moesten deze in een diagram worden geplaatst waarbij de relaties duidelijk worden weergegeven. Een groot gedeelte van de applicatie draait om het vormgeven van de videospeler. Daarnaast zijn veel handelingen direct te koppelen aan de gebruiker. Zowel videospeler als gebruiker zijn als klassen gedefinieerd. Deze klassen nemen dan ook een centrale rol in het diagram aan. Uitgaande van de klasse Videospeler heeft de gebruiker volgens de systeemeisen vijf mogelijkheden: 'kleuren kiezen', 'formaat instellen', 'knoppen selecteren', 'intro afbeelding kiezen' en 'videospeler afbeelding kiezen'. De wijze waarop deze mogelijkheden worden vertegenwoordigd is weergegeven in onderstaand schema.

<b>mogelijkheid</b>	<b>vertegenwoordigd door</b>
kleuren kiezen	klasse Kleurenpalet
formaat instellen	attributen van klasse Videospeler
knoppen selecteren	attributen van klasse Videospeler
intro afbeelding kiezen	klasse Introbeeld
videospeler afbeelding kiezen	klasse Hoekafbeelding

fig. 6.2 Videospelerpagina's vertegenwoordiging in klassediagram

Na generalisatie van Introbeeld en Hoekafbeelding ontstaat de klasse Videospelerbeeld zoals ook aangegeven in de lijst met kandidaatklassen.

Bij het bepalen van hoe deze mogelijkheden in het klassediagram worden verwerkt heb ik veel-op-veel relaties en één-op-veel relaties als klasse beschouwd. Eén-op-één relaties kunnen direct in de klasse Videospeler worden ondergebracht. Voorbeelden van één-op-één relaties zijn de mogelijkheden om het formaat in te stellen en de bedieningsknoppen te selecteren. Een uitzondering hierop is de klasse Kleurenpalet. Deze klasse vormt een één-op-één relatie met de klasse Videospeler. Aangezien de gebruiker de mogelijkheid heeft om acht verschillende kleuren te definiëren heb ik om het overzicht van het diagram te handhaven ervoor gekozen om deze acht kleuren in een aparte klasse te zetten.

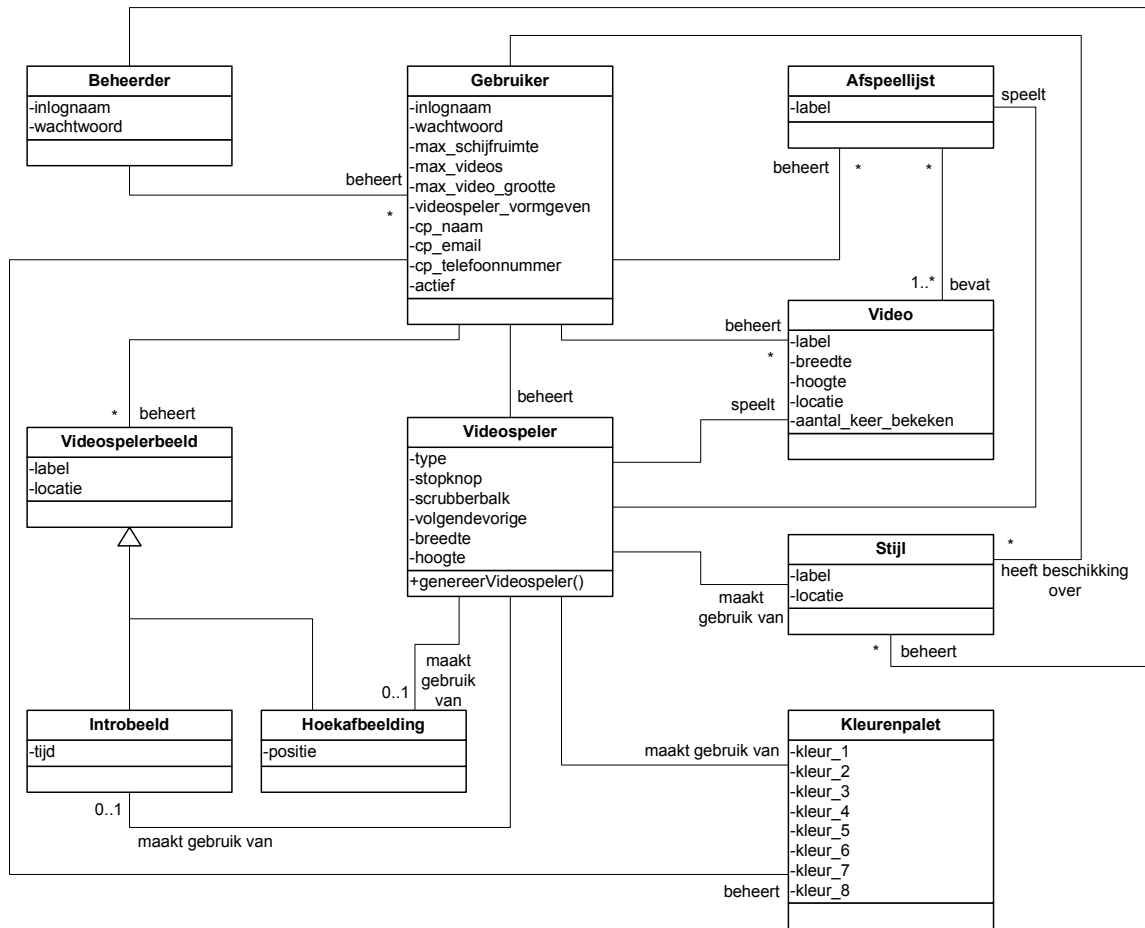


fig. 6.3 Klassediagram

### 6.3 Usecases opstellen

Om de verschillende mogelijkheden die de gebruiker heeft duidelijk in beeld te brengen heb ik usecases opgesteld. Deze usecases kunnen vrijwel één-op-één worden afgeleid uit de systeemeisen. Sommige groepen systeemeisen kunnen met een enkele usecase worden beschreven. Zo worden bijvoorbeeld de systeemeisen 'Aantal keer bekeken per video', 'Totaal aantal video's online' en 'Totaal hoeveelheid schijfruimte aan video's, SWF bestanden en afbeeldingen' vertaald naar de usecase 'Statistieken inzien'.

Op basis van de usecases worden de sequentiediagrammen opgesteld. Deze sequentiediagrammen staan aan de basis van het programmeerwerk. Daarom is het van belang de handelingen en communicatie tussen gebruiker en systeem en de communicatie tussen verschillende onderdelen van het systeem binnen de usecases zo uitgebreid mogelijk te

beschrijven. Als de usecases uitgebreid worden opgesteld en compleet zijn zal tijdens de programmeerfase niets over het hoofd worden gezien.

Een voorbeeld van een usecase is 'Afspeellijst wijzigen'.

Naam	Wijzigen van een afspeellijst.
Samenvatting	Gebruiker wijzigt een afspeellijst.
Actoren	Gebruiker.
Aannamen	Usecase 'Inloggen' is uitgevoerd.
Beschrijving	(1) Gebruiker navigeert naar de pagina 'afspeellijsten beheren'. (2) Het systeem toont een scherm met een lijst van alle beschikbare afspeellijsten met de opties 'wijzigen', 'verwijderen' en 'code genereren' en een knop voor het toevoegen van een afspeellijst. (3) Gebruiker klikt op de wijzigknop naast de afspeellijst die hij wil wijzigen. (4) Het systeem toont het scherm waarmee de afspeellijst kan worden gewijzigd. (5) Gebruiker geeft de wijzigingen in. (6) Het systeem controleert of het label veld is ingevuld en of er meerdere video's zijn geselecteerd. Als er geen label is ingevuld treedt uitzondering [Geen label ingevuld] op. Als er één of geen video's zijn geselecteerd treedt uitzondering [Te weinig video's geselecteerd] op. (7) Het systeem slaat de combinatie van video's op als een afspeellijst.
Uitzonderingen	[Geen label ingevuld] Het systeem geeft de melding dat er geen label is ingevuld. Gebruiker heeft opnieuw de mogelijkheid een label in te vullen. [Te weinig video's geselecteerd] Het systeem geeft de melding dat er te weinig video's zijn geselecteerd. Gebruiker heeft opnieuw de mogelijkheid video's te selecteren.
Resultaat	Gebruiker heeft een afspeellijst gewijzigd.
Sequentie diagrammen	(2) Afspeellijsten ophalen (5-7) Afspeellijst toevoegen

fig. 6.4 Usecase 'Wijzigen van een afspeellijst'

Naast de kopjes van de standaard usecase template 'Naam', 'Samenvatting', 'Actoren', 'Aannamen', 'Beschrijving', 'Uitzonderingen' en 'Resultaat' heb ik ook het kopje 'Sequentiediagrammen' toegevoegd. Verschillende handelingen binnen de usecase zijn vertaald naar verschillende sequentiediagrammen. De nummers van de handelingen onder het kopje 'Beschrijving' corresponderen met de nummers onder het kopje 'Sequentiediagrammen'. In dit geval wordt handeling (2) dus weergegeven in het sequentiediagram 'Afspeellijsten ophalen' en handelingen (5-7) in het sequentiediagram 'Afspeellijst toevoegen'. Op deze manier kan ik binnen het Technisch ontwerprapport gerelateerde onderdelen gemakkelijk terugvinden.

Nadat ik de usecases aan de opdrachtgever had voorgelegd vond de opdrachtgever dat er te weinig statistieken werden bijgehouden en het gebruik van de videospeler te weinig werd geregistreerd. De systeemeisen zijn echter in de vorige fase al vastgelegd. Ik heb aangegeven dat als er systeemeisen moeten worden toegevoegd, enkele systeemeisen ook zouden moeten

afvallen aangezien de planning weinig uitloop toelaat. Daarnaast is er werk voor niets geweest omdat de usecases op de systeemeisen zijn gebaseerd. Bij wijzigingen is er dus sowieso tijdverlies. We hebben daarom afgesproken dat de extra functionaliteit die de opdrachtgever voor ogen heeft in de systeemeisen als 'Won't have' wordt opgenomen en deze functionaliteit als eerste wordt gebouwd als er na de ontwikkeling van de 'Must have', 'Should have' en 'Could have' eisen tijd over is.

Gedurende deze fase zijn ook enkele wijzigingen op het oorspronkelijke plan van navigeren doorgevoerd. Zo had ik aanvankelijk het idee opgevat om bijvoorbeeld na een klik op het kopje 'video's beheren' drie links te tonen naar 'video toevoegen', 'video wijzigen' en 'video verwijderen'. Als de gebruiker dan op 'video wijzigen' klikt wordt een lijst met alle reeds geplaatste video's getoond waarna de gebruiker de video kiest die hij wil wijzigen. Daarna wordt een scherm getoond waarmee de gegevens van die video gewijzigd kunnen worden. Na even met de opdrachtgever hierover van gedachten te hebben gewisseld hebben we besloten dat de gebruiker meer gebaat zou zijn bij het direct tonen van de lijst met reeds geplaatste afbeeldingen na het klikken op het kopje 'afbeeldingen beheren'. Elke afbeelding in deze lijst krijgt verschillende knoppen waarmee handelingen als 'wijzigen' en 'verwijderen' kunnen worden uitgevoerd. Op deze manier is de gebruiker eerder op de pagina waar wijzigingen kunnen worden aangebracht.

Deze manier van werken is iets gecompliceerder dan aanvankelijk de bedoeling was maar omdat Miriam (de primaire persona) een gemiddeld ervaren internetgebruiker is zal dit niet voor problemen zorgen. Daarnaast heeft Wilfried (de secundaire persona) veel baat bij deze aanpassing omdat hij snel resultaat wil zien zonder allerlei acties te moeten uitvoeren. De nieuwe wijze van tonen van video's werkt dus in zijn voordeel.

Vergelijkbare pagina's als 'afbeeldingen beheren' en 'afspeellijsten beheren' zullen op dezelfde manier worden ontworpen. Zo blijft het systeem consequent en kan Wilfried zonder frustraties met het systeem overweg zonder dat Miriam daar moeilijkheden bij ervaart.

## 6.4 Usecasediagram opstellen

De usecasediagrammen geven aan welke actoren over welke usecases beschikken. Deze applicatie kent twee actoren, 'Gebruiker' en 'Beheerder'. Er is zowel in de systeemeisen als in de usecases zelf al onderscheid gemaakt tussen deze twee groepen. Het opstellen van het usecasediagram is dus slechts een kwestie van toepassen.

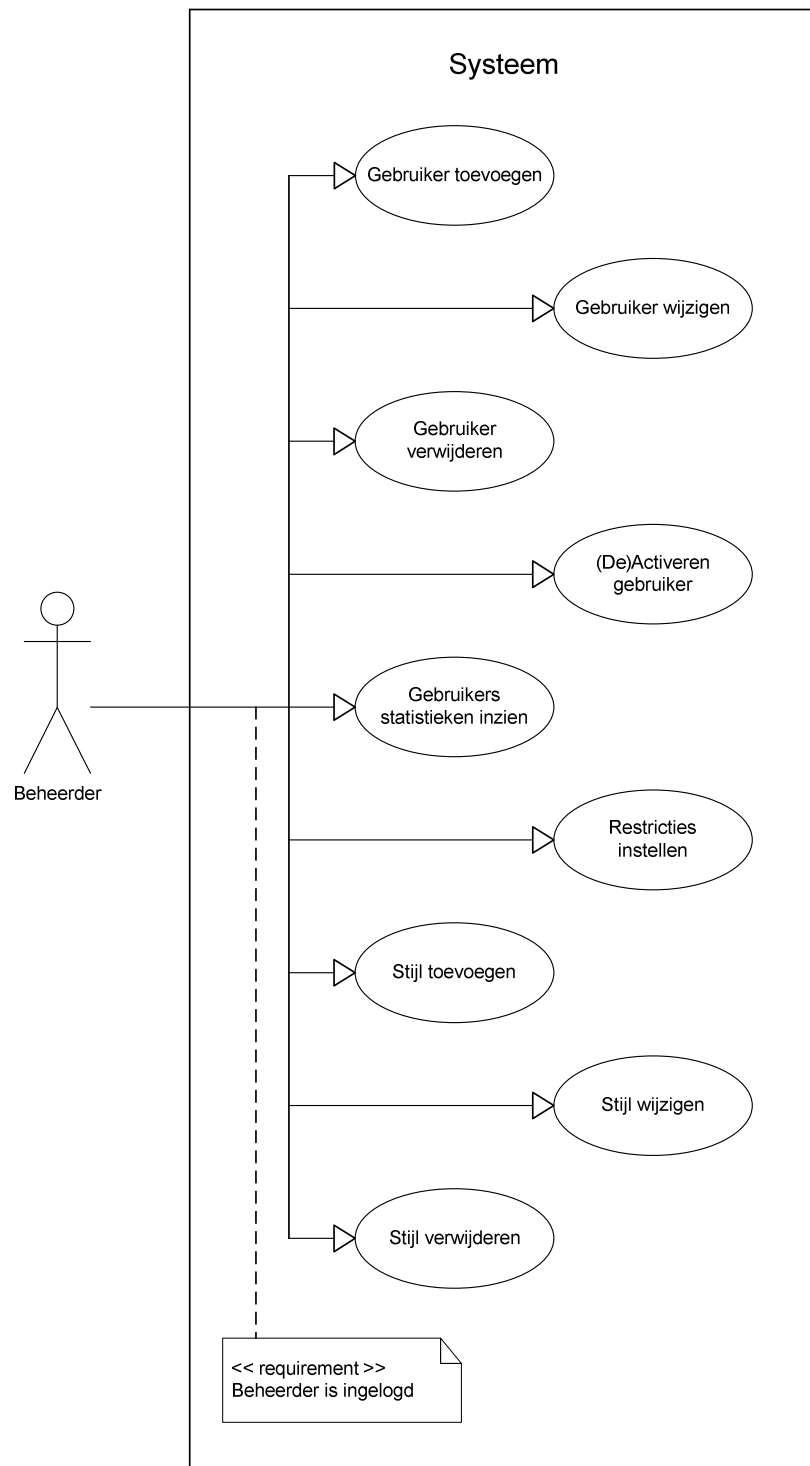


fig. 6.5 Usecasediagram van Beheerder

## 6.5 Sequentiediagrammen opstellen

Zoals eerder al aangegeven laten de usecases zich vertalen naar sequentiediagrammen. De sequentiediagrammen verbinden de handelingen van de usecases met de onderlinge communicatie van de klassen uit het klassediagram zodat deze drie technieken een consistent geheel vormen. Het is niet zo dat de usecases één-op-één de sequentiediagrammen vormen. Verschillende delen uit de usecases worden bij verschillende handelingen gebruikt. Om een afspeellijst te wijzigen moeten eerst alle afspeellijsten worden opgehaald waarna de gebruiker de te wijzigen afspeellijst kiest. Bij het verwijderen van een afspeellijst moeten ook eerst alle afspeellijsten worden opgehaald. Daarna kiest de gebruiker de afspeellijst die hij wil verwijderen. Het ophalen van afspeellijsten komt dus binnen meerdere handelingen voor. Daarom is hiervoor een apart sequentiediagram opgesteld waarnaar verschillende usecases verwijzen.

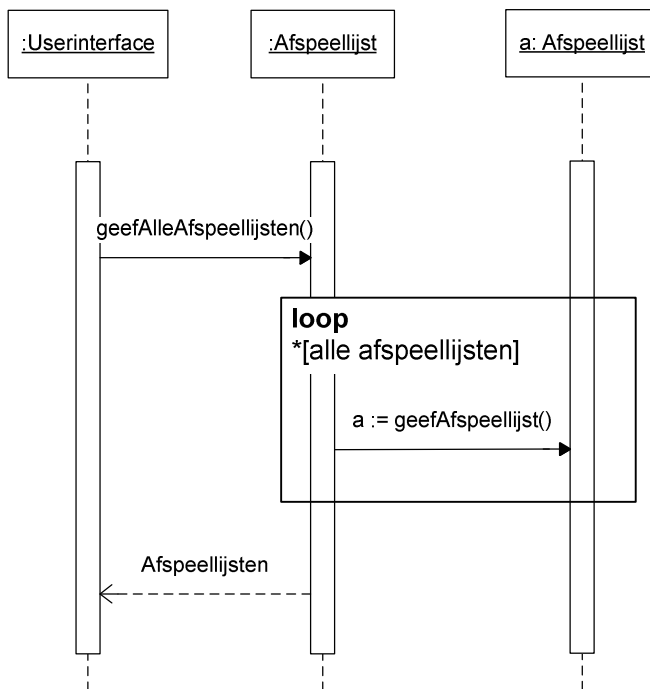


fig. 6.6 Sequentiediagram 'Afspeellijsten ophalen'

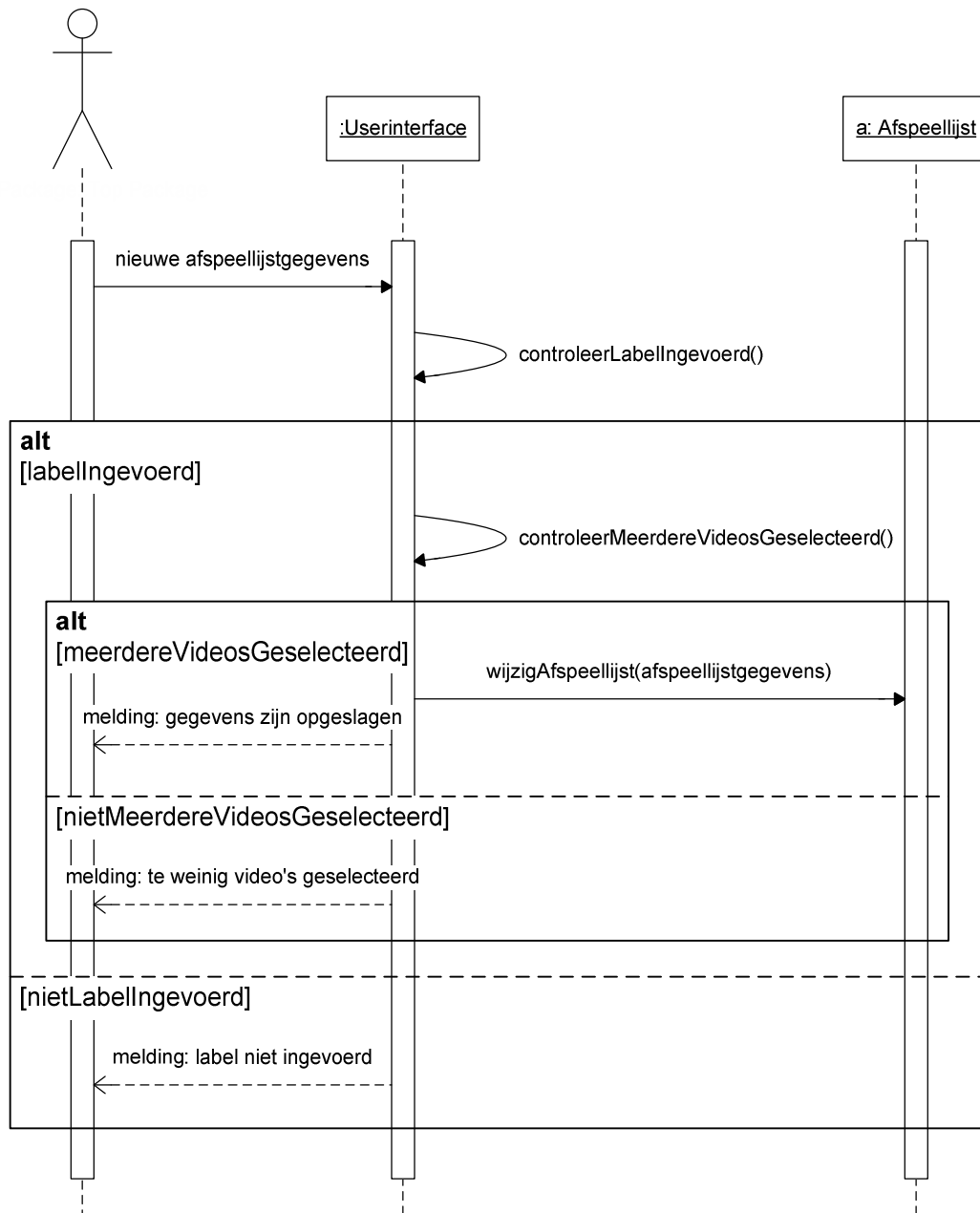


fig. 6.7 Sequentiediagram 'Afspeellijst wijzigen'

## 6.6 Database ontwerpen

Zoals in hoofdstuk 3 'Aanpak' al is vermeld wordt het database ontwerp gebaseerd op het klassediagram. Naast de relaties tussen verschillende tabellen met lijnen aan te geven wordt bij het database ontwerp ook gewerkt met unieke numerieke waarden (id's). Daarnaast is ook de

generalisatie van 'introbeeld' en 'hoekafbeelding' naar 'videospelerbeeld' vertaald naar een enkele tabel 'videospelerbeeld'. De subklassen 'introbeeld' en 'hoekafbeelding' hebben weliswaar andere eigenschappen ('tijd' en 'positie') maar deze worden omwille het overzicht van de database ondergebracht in de tabel 'videospeler' (resp. 'introtijd' en 'hoekafb\_pos'). De subklassen zijn daarom technisch gezien identiek en kunnen dus in dezelfde tabel 'videospelerbeeld' worden geplaatst. In de tabel 'videospeler' wordt naar de tabel 'videospelerbeeld' verwezen met 'introbeeld\_id' en 'hoekafbeelding\_id'. Beide velden slaan op het veld 'id' in de tabel 'videospelerbeeld'. Correspondierend met de keuzes die bij het opstellen van het klassediagram zijn gemaakt worden de acht kleuren ook in het database ontwerp in een aparte tabel gezet.



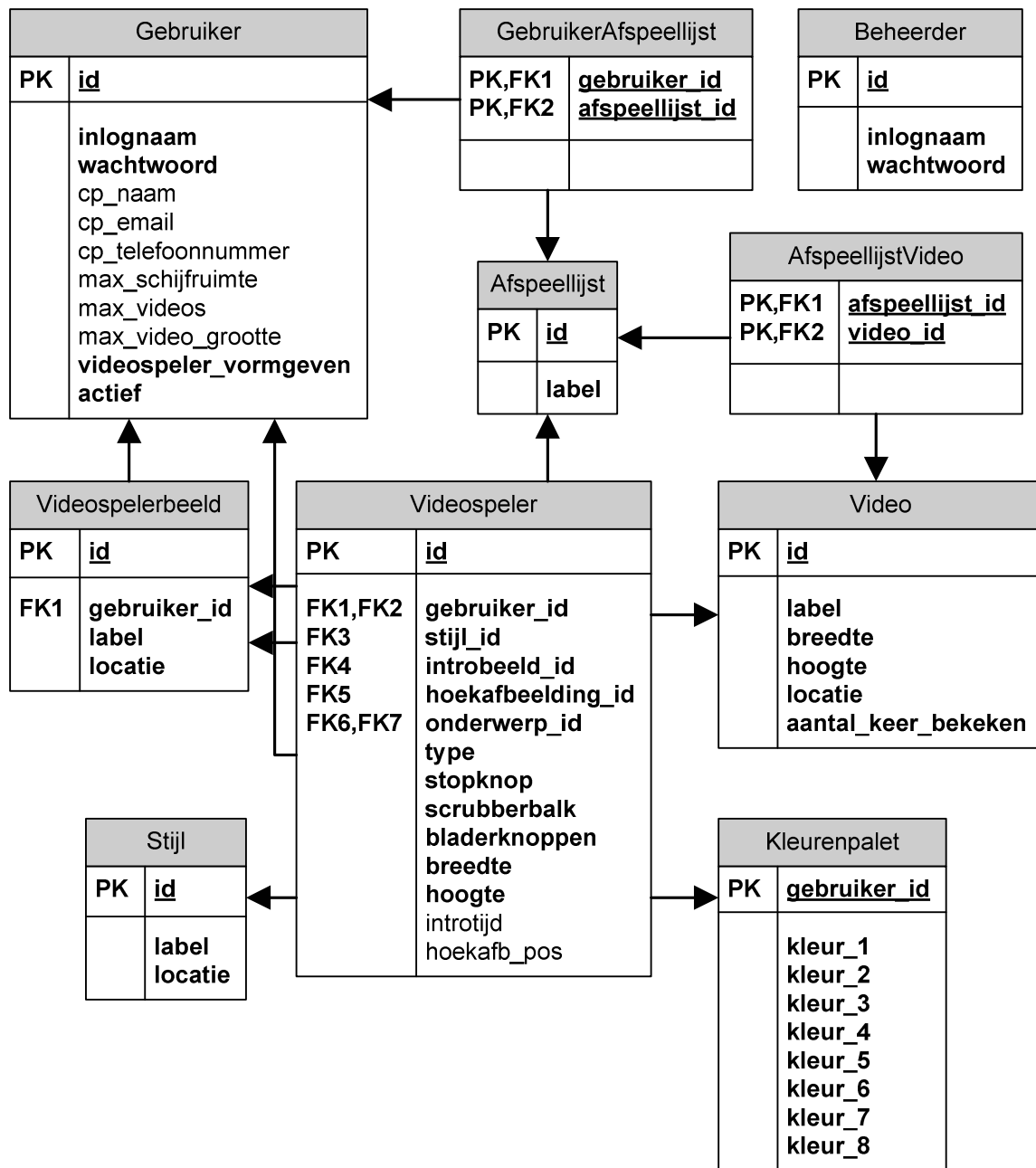


fig. 6.8 Database ontwerp

## 6.7 Navigatiemodel

Voor het navigatiemodel zijn de systeemeisen en usecases vertaald naar pagina's waarop de handelingen uit te voeren zijn. Dit leverde buiten de inlogpagina een zevental hoofdpagina's op.



fig. 6.9 Navigatiemodel voor gebruiker

## 6.8 Bestandstructuur opstellen

De bestandstructuur heb ik gebaseerd op mijn eigen ervaring met de ontwikkeling van content management-achtige systemen. Daarbij heb ik natuurlijk veelvuldig van het technisch ontwerp

gebruik gemaakt. Zo zijn de klassen uit het klassediagram ook gebruikt als klassen in PHP. De pagina's uit het navigatiemodel zijn ook als zodanig in de bestandstructuur terug te vinden.

Daarnaast heb ik rekening gehouden met de dynamische opzet van het systeem. Bepaalde gedeelten van de pagina's onder 'videospeler beheren' zullen op elke subpagina hetzelfde zijn. In plaats van de code op elke pagina te plaatsen heb ik een pagina 'videospelerbeheren\_footer.php' gedefinieerd. Op elke pagina wordt deze PHP pagina aangeroepen. Als er tijdens het programmeren een wijziging in dit gedeelte moet worden doorgevoerd hoef ik dat slechts eenmaal op deze pagina te doen. De vijf pagina's roepen alle vijf hetzelfde php script aan en zullen dus dezelfde wijzigingen tonen. Zou ik het PHP script rechtstreeks op alle vijf de pagina's hebben geplaatst dan zou ik wijzigingen ook telkens op alle vijf de pagina's moeten doorvoeren.

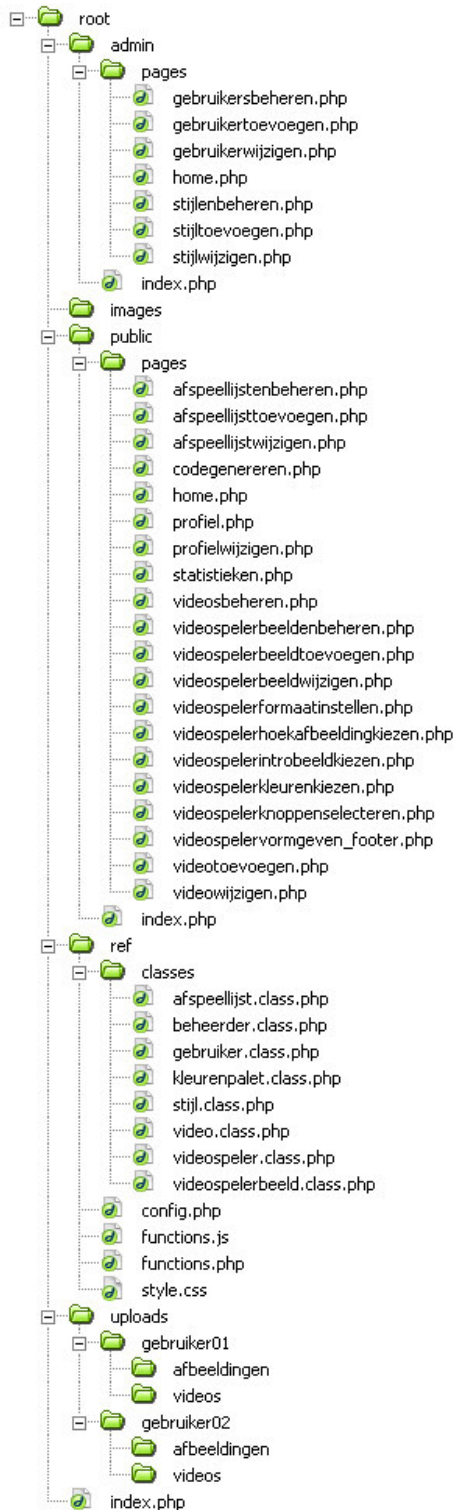


fig. 6.10 Bestandstructuur

## 7 VISUEEL ONTWERP VAN ICONEN EN SCHERMEN OPSTELLEN

Dit hoofdstuk beschrijft hoe de fase 'Visueel ontwerp van iconen en schermen' in zijn werk is gegaan. Aanvankelijk was het idee om - corresponderend met *twee* groepen gebruikers met verschillend kennisniveau en internetervaring - *twee* verschillende stijlen te ontwerpen en deze aan de doelgroep voor te leggen. De stijl of combinatie van stijlen die het beste zou werken zou vervolgens worden uitgewerkt. Zoals reeds aangegeven (zie paragraaf 5.2

'Doelgroepomschrijving opstellen') is het verschil in kennis en ervaring niet zo groot als aanvankelijk gedacht. Dit heeft mij doen besluiten één stijl te ontwikkelen. Het verschil was niet groot genoeg om twee verschillende stijlen te ontwikkelen.

### 7.1 Wireframes opstellen

Nadat alle gegevens over de doelgroep waren vastgelegd was het tijd om de informatie toe te passen. Om een indruk te krijgen hoe verschillende onderdelen en elementen zich met elkaar verhouden heb ik wireframes opgesteld. De wireframes beschrijven de globale opbouw van verschillende pagina's. Miriam wil graag duidelijk voor ogen hebben welke opties ze heeft. Daarom heb ik gekozen voor een klassieke aanpak, gebruikmakende van een verticale navigatie aan de linkerkant. Voor de subnavigatie heb ik gekozen voor een horizontale verdeling. De manier van navigeren zal een persoon met gemiddelde internetervaring niet in de problemen brengen en geeft een overzichtelijk beeld van de mogelijkheden die de gebruiker heeft. Daarnaast heb ik van de verschillende elementen een overzichtelijk geheel gemaakt, met veel ruimte tussen de onderdelen en grote knoppen.

APPLICATIE TITEL		LOGO	
		BREADCRUMBS	
		TITEL	
MENU	<div> <div> <input type="checkbox"/> <input type="text"/> </div> <div> <input type="checkbox"/> <input type="text"/> </div> <div> <input checked="" type="checkbox"/> <input type="text"/> </div> <div> <input type="checkbox"/> <input type="text"/> </div> <div> <input checked="" type="checkbox"/> <input type="text"/> </div> <div> <input type="checkbox"/> <input type="text"/> </div> </div> <div> <div>&gt;</div> <div>&gt;&gt;</div> <div>&lt;</div> <div>&lt;&lt;</div> </div> <div> <input type="checkbox"/> <input type="text"/> </div> <div> <input type="checkbox"/> <input type="text"/> </div>		

fig. 7.1 Wireframe van de pagina 'Afspeellijst toevoegen'

APPLICATIE TITEL		LOGO	
		BREADCRUMBS	
		TITEL	
MENU			SUBMENU
	VIDEOSPELER PREVIEW		SUBTITEL
			<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>
HELP KNOP		BESCHIKBARE STYLEN	

fig. 7.2 Wireframe van een subpagina van de pagina 'Videospeler beheren'

Op de pagina 'Video's beheren' worden alle video's in een lijst getoond. Achter elke video staan de knoppen 'wijzigen', 'verwijderen' en 'plaatsen'. De knop 'plaatsen' is bedoeld voor Wilfried en verwijst naar de pagina 'Videospeler plaatsen' bij wijze van 'shortcut'. Deze pagina genereert de te plaatsen code na het selecteren van een video of afspeellijst. Nadat op de knop 'plaatsen' is geklikt wordt de betreffende video op de pagina 'Videospeler plaatsen' alvast ingevuld zodat de code direct beschikbaar is. Dit geldt ook voor de pagina 'Afspeellijsten beheren'.

Bij het maken van de wireframes ben ik uitgegaan van een verhouding van 1003x599. Dit is de beschikbare grootte in de browser in pixels van een scherm in een 'standaard' opstelling. De 'standaard' opstelling is een scherm ingesteld op een resolutie van 1024x768 pixels en websites getoond worden in Internet Explorer waarbij de titelbalk, bestandbalk, standaardknoppen, adresbalk, statusbalk en scrollbar zichtbaar zijn.

## 7.2 Iconen schetsen

Miriam is zeer visueel ingesteld. Daarom heb ik ervoor gekozen om de verschillende opties die de gebruiker heeft en de verschillende handelingen die de gebruiker kan uitvoeren uit te beelden met behulp van iconen. Veel handelingen binnen het systeem zijn applicatie specifiek en komen dus niet voor in andere systemen. Een voorbeeld hiervan is 'videospeler plaatsen'. Voor deze handeling zal een geheel nieuw icoon ontworpen moeten worden. Omdat deze opdracht aan een deadline is verbonden en ik niet maanden de tijd heb om de icoon te ontwikkelen, te testen en te herontwikkelen zal ik de handelingen en opties niet met alleen een icoon tonen. Het zullen knoppen worden waarop de handeling tekstueel wordt vermeld en ondersteund wordt met behulp van een icoon. Ik zou ook kunnen volstaan met alleen een tekstuele knop, maar ik wilde de relatie tussen verschillende elementen van de pagina benadrukken. Dit zal ik in paragraaf 7.5 'Iconen uitwerken' nader toelichten. Daarnaast zijn visuele weergaven van handelingen soms sneller te begrijpen dan tekstuele verklaringen.

Ik heb ervoor gekozen de iconen eerst te schetsen en ze daarna digitaal uit te werken. Het schetsen stelt me in staat om veel verschillende iconen te maken zonder te veel aandacht te hoeven besteden aan precies werk en digitale beperkingen. Daarnaast ben ik met schetsen niet gebonden aan een werkplek, maar kan ik in een inspiratie vollere omgeving werken.



fig. 7.3 Schetsen van iconen en knoppen

### 7.3 Schermonderdelen schetsen

Zowel voor mezelf als voor de opdrachtgever heb ik verschillende schermonderdelen geschetst die de sfeer en het gevoel van het systeem moeten weergeven. Ik heb ervoor gekozen om ook de schermonderdelen eerst te schetsen en daarna digitaal uit te werken. Dezelfde redenen als bij het schetsen van de iconen zijn hier van toepassing.



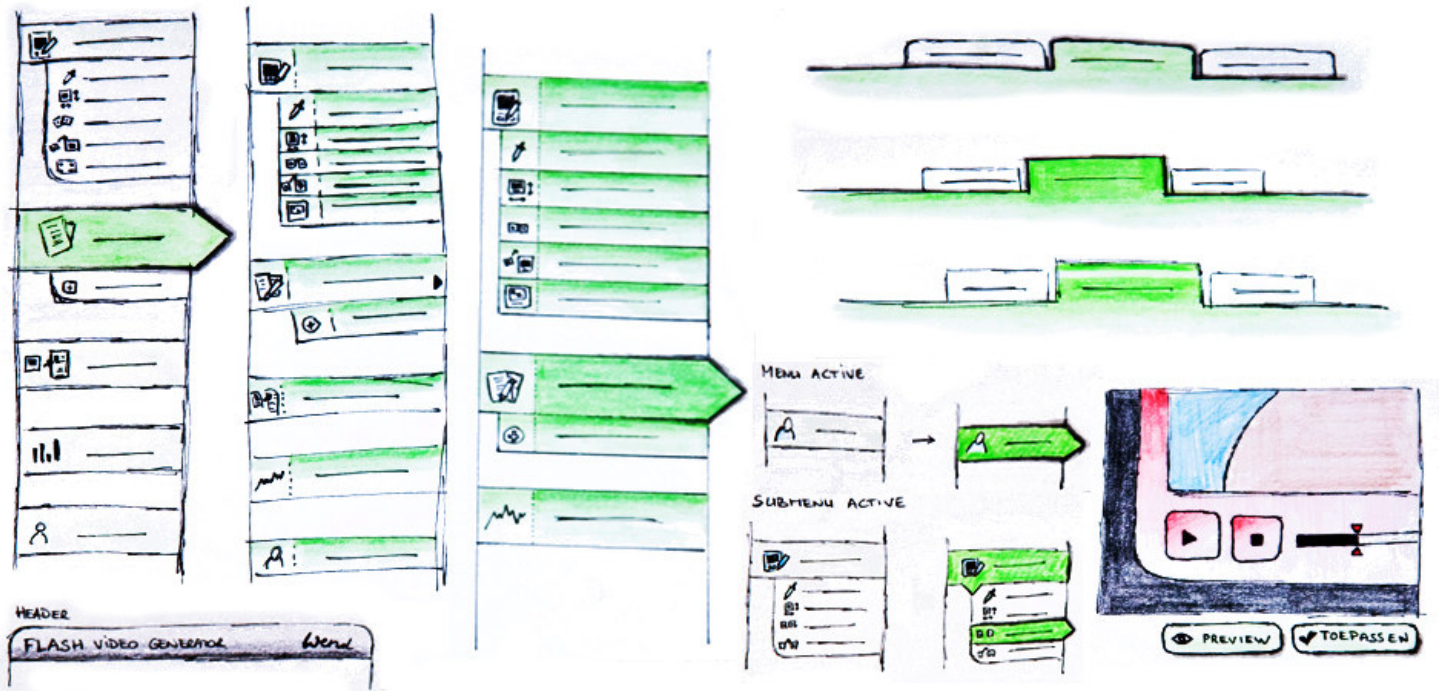


fig. 7.4 Schetsen van schermelementen

#### 7.4 Schermonderdelen uitwerken in schermen

Het digitale ontwerp is gebaseerd op de wireframes en de schetsen. Er was dus al veel materiaal van het uiterlijk van de applicatie. Veel elementen heb ik vrijwel één-op-één van de schets kunnen overnemen in de digitale versie van het ontwerp. De positie heb ik overgenomen van de reeds ontwikkelde wireframes. Tijdens het ontwerpen heb ik veel contact gehouden met de opdrachtgever waarbij ik mijn vorderingen telkens heb voorgelegd. Tijdens verschillende contactmomenten heb ik feedback van de opdrachtgever gekregen die hebben geleid tot wijzigingen in het ontwerp. Hieronder is bijvoorbeeld te zien hoe de knoppen zich ontwikkeld hebben.



fig. 7.5 Knoppen ontwikkeling

De opdrachtgever heeft aangegeven dat de eerste versie van de knoppen iets meer 'web 2.0' moest zijn. Daarmee werd het contrast met het hoofdmenu echter te groot (versie 2), de knop is daarom iets verkleind en is het witte verloop iets subtieler weergegeven (versie 3). Ook is het hoofdmenu meer afgestemd op de 'web 2.0 look'.





fig. 7.6 Menu aanpassing in stijl van nieuwe knoppen

De icoon was echter te weinig zakelijk, te speels en kinderlijk. Uiteindelijk is de icoon van de eerste versie gebruikt om de website een 'zakelijke web 2.0 look' te geven. De laatste versie van de knoppen werd door de opdrachtgever goedgekeurd.

Daarnaast heeft de opdrachtgever aangegeven dat de ruimte tussen de menu items wellicht weggelaten voor meer rust en overzicht. Deze keuze wordt tijdens de eerste usability test aan de testpersonen voorgelegd.



fig. 7.7 Menu met ruimte

of

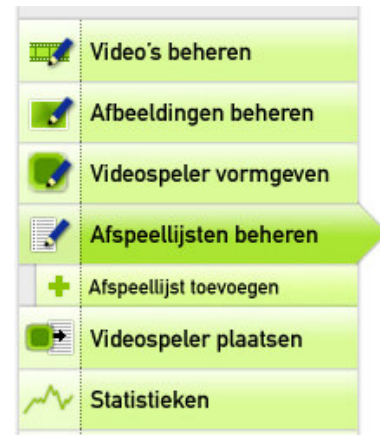


fig. 7.8 Menu zonder ruimte

Verschillende elementen (bijvoorbeeld het gebruik van tabs) en de gebruikte kleurtinten zijn gebaseerd op het CMS dat Wend gebruikt en het moodboard. Hieronder wordt een gedeelte van het CMS getoond.



fig. 7.9 CMS dat Wend hanteert (gedeeltelijk)

De pagina 'Afspeellijst wijzigen' is er als volgt uit komen te zien.

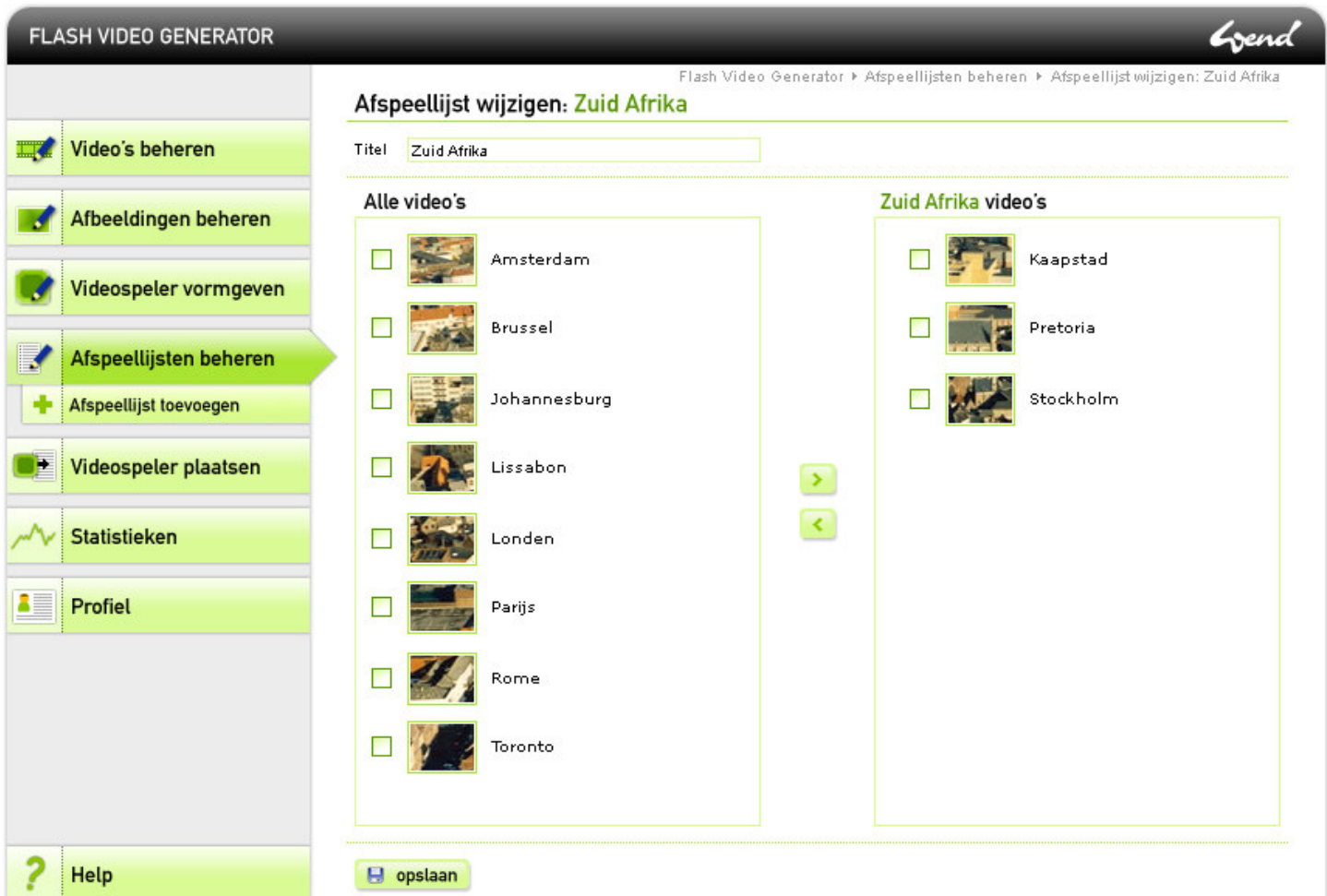


fig. 7.10 Ontwerp van de pagina 'Afspeellijst wijzigen'

De pagina 'Videospeler vormgeven' bestaat uit vijf verschillende subpagina's. Hiervoor heb ik gekozen voor tabs waarvan ook in het CMS dat Wend gebruikt veelvuldig gebruik wordt gemaakt.

Aanvankelijk zag deze pagina er als volgt uit.



fig. 7.11 Tabs ontwerp op de subpagina's van de pagina 'Videospeler beheren'

De groene kleur van de tabs en het groene verloop in de achtergrond zou echter kunnen gaan vloeien met de kleuren die de gebruiker aan zijn videospeler geeft. Daarnaast leidt het af van waar het op de pagina's werkelijk om gaat. Daarom is er uiteindelijk gekozen voor witte en grijze tabs.

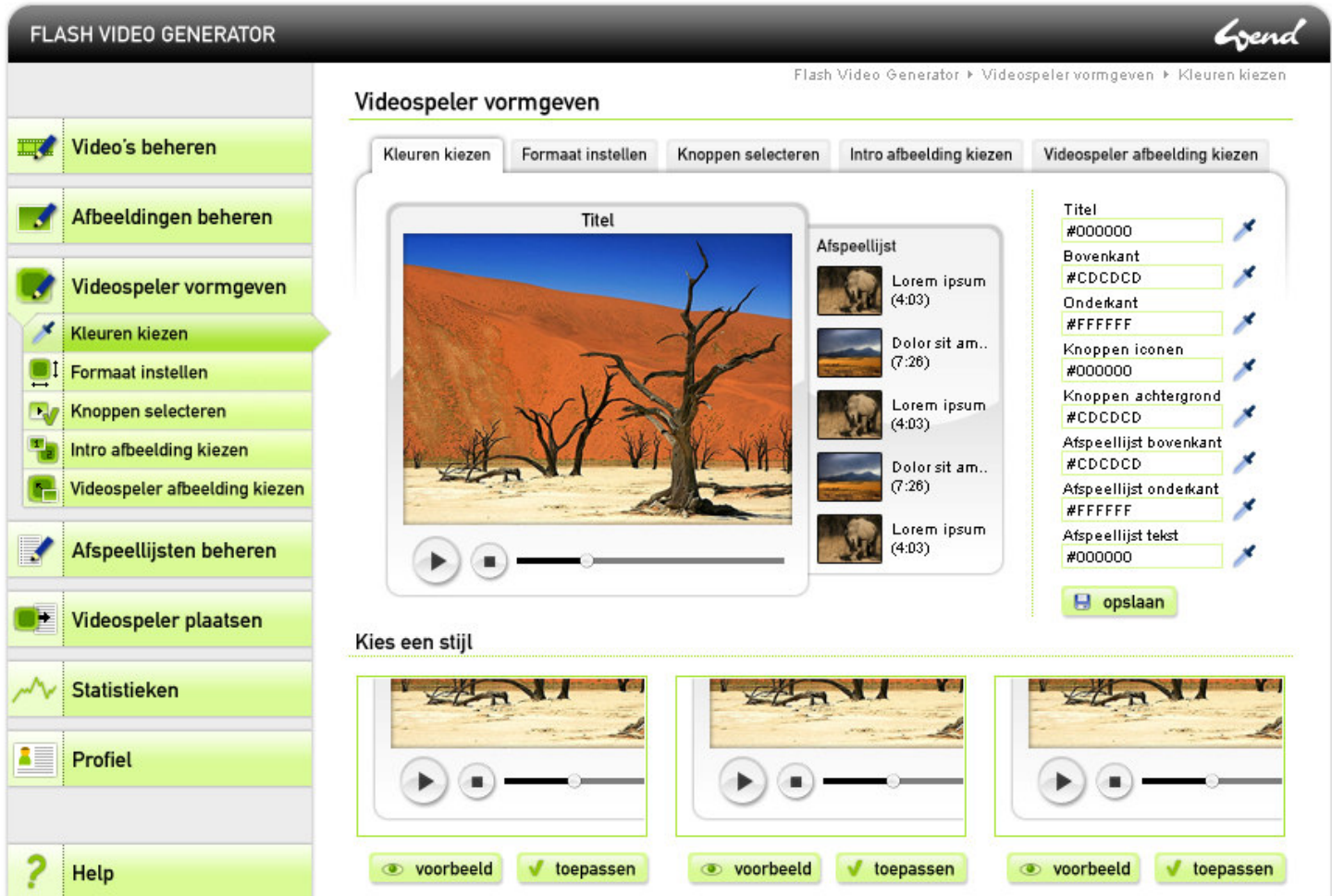


fig. 7.12 Ontwerp van de pagina 'Kleuren kiezen'

In dit scherm valt op dat de subnavigatie zowel met tabs als in het menu als subitems wordt getoond. Dit zou zowel een positieve als negatieve uitwerking kunnen hebben. De gebruiker zou sneller zijn weg kunnen vinden doordat de navigatie op meerdere plekken staat, maar dat zou ook averechts en verwarrend kunnen werken. Er is besloten om de gebruiker tijdens de usability test drie keuzes voor te leggen:

- navigatie middels een submenu én tabs;
- navigatie middels alleen een submenu;
- navigatie middels alleen tabs.

Door dit soort keuzes aan de gebruiker voor te leggen, wordt de doelgroep actief bij het ontwerpproces betrokken.

## 7.5 Iconen uitwerken

De meeste digitale versies van iconen zijn direct overgenomen van de schetsen. Voor een aantal iconen had ik verschillende ideeën. Voor deze iconen heb ik ook meerdere digitale uitwerkingen gemaakt. Bij de eerste test worden de verschillende versies voorgelegd. De icoon die het beste wordt ontvangen zal ik uiteindelijk in de applicatie gaan gebruiken.



*fig. 7.13 Iconen voor de handeling 'verwijderen'*

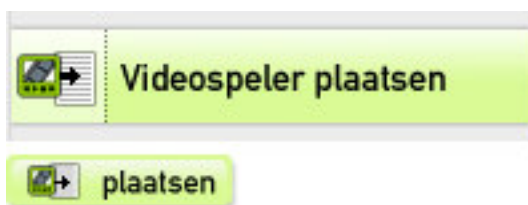
Ik heb de iconen zo ontworpen dat het niet op zichzelf staande afbeeldingen zijn maar dat het een serie is waarvan de verschillende iconen de dezelfde stijl hebben. Een aantal elementen zal in verschillende iconen voorkomen.



*fig. 7.14 Iconen voor de handelingen 'formaat instellen', 'videospeler beheren' en 'videospeler plaatsen'*

In bovenstaande iconen stelt het donkergroene vlak met de lichtgroene rand een versimpelde versie van de videospeler voor. Dit komt in verschillende iconen terug.

Zoals eerder vermeld in paragraaf 7.2 'Iconen schetsen' heb ik ervoor gekozen om iconen te gebruiken om de relatie tussen verschillende delen van het systeem te benadrukken. Een goed voorbeeld hiervan is de handeling 'videospeler plaatsen'. Dit is een menu item, maar is bij wijze van 'shortcut' ook op de pagina's 'Video's beheren' en 'Afspeellijsten beheren' geplaatst (zie paragraaf 7.1 'Wireframes opstellen'). Om deze pagina's niet te laten domineren door de veelheid en grootte van knoppen heb ik het label van de knop 'videospeler plaatsen' op deze pagina's tot 'plaatsen' beperkt. Door van hetzelfde icoon gebruik te maken als het menu item waarvan het label wel 'Videospeler plaatsen' is, zullen gebruikers gemakkelijker begrijpen dat deze onderdelen verband houden en dezelfde functie hebben.



*Fig. 7.15 Verband tussen schermelementen met behulp van iconen; hier is de laatste versie iconen gebruikt (zie hoofdstuk 9 'Visueel ontwerp uitwerken')*

## 7.6 Videospeler stijlen ontwerpen

Gedurende deze fase heb ik ook gewerkt aan het ontwerpen van videospeler stijlen. Deze ontwerpen dienen slechts als voorbeeld voor wat er mogelijk is. Wend kan later zelf stijlen ontwerpen en toevoegen via het beheersysteem.

In overleg met de opdrachtgever heb ik drie stijlen ontworpen, geschikt voor drie verschillende doelgroepen. Er is gekozen voor een stijl met een mac-look voor een brede doelgroep, een stijl met een web 2.0 look en een 'platte' sobere stijl waar weinig diepte in zit zodat deze opgaat in de achtergrond.

## 8 USABILITY TESTEN VAN ICONEN EN SCHERMEN

Dit hoofdstuk beschrijft de manier waarop het usability testen van iconen en schermen in zijn werk is gegaan. Tevens komt de verwerking van de testresultaten aan bod.

### 8.1 Doel bepalen

Een test of onderzoek begint met het formuleren van een doel. Het doel van dit eerste onderzoek is bepalen of de ontwikkelde iconen en schermen voldoende overeenkomen met de verwachtingen van de gebruiker. Ik heb dit doorvertaald naar een hoofdvraag (rechts) die vervolgens uiteen viel in een drietal deelvragen.

**“Komen de iconen en schermen voldoende overeen met de verwachtingen van de gebruiker?”**

*fig. 8.1 Hoofdvraag van het onderzoek naar iconen en schermen*

- *In hoeverre herkennen gebruikers de ontwikkelde iconen?*
- *In hoeverre herkennen gebruikers de iconen als de betekenissen bekend zijn?*
- *In hoeverre geven de schermen duidelijkheid over de wijze waarop taken uitgevoerd moeten worden?*

Om de antwoorden op bovenstaande vragen te achterhalen heb ik het onderzoek in drie delen opgesplitst. Elke deel vertegenwoordigt een deelvraag. Deze drie onderdelen (testtaken) worden in de volgende paragraaf behandeld.

Bij de iconen is het volgende onderscheid te maken.

- Algemene iconen
- Applicatie specifieke iconen

Algemene iconen zijn iconen die ook bij applicaties als Word, Excel en andere Content Management Systemen worden gebruikt.

De applicatie specifieke iconen zijn nieuwe iconen die speciaal zijn ontworpen voor de te ontwikkelen applicatie en hebben betrekking op handelingen die alleen binnen de applicatie worden uitgevoerd. De kans dat testpersonen de iconen herkennen is vrij laag omdat het geheel nieuw ontworpen iconen zijn.

Naar verwachting zullen de algemene iconen eerder worden herkend dan de applicatie specifieke iconen.


## **8.2 Testtaken opstellen**

Zoals in de vorige paragraaf vermeld wordt het onderzoek in drie delen opgesplitst. Deze paragraaf behandelt deze drie delen.

### *8.2.1 Iconen herkennen*

Het eerste deel van het onderzoek moet antwoord geven op de deelvraag 'In hoeverre herkennen gebruikers de ontwikkelde iconen?'. Het antwoord van die vraag heb ik geprobeerd te achterhalen met wat in The Icon Book (door William Horton) wordt beschreven als 'Open-ended meaning-for-image tests'. Hierbij krijgt de testpersoon een icoon voorgelegd en wordt gevraagd welke betekenis de icoon heeft of welke associatie de testpersoon legt. Hiermee wordt getest welke associaties de gebruiker voor een bepaald icoon al heeft. Als blijkt dat veel testpersonen al een sterke associatie met een bepaald icoon hebben en deze associatie heeft niets met de bedoelde betekenis te maken dan moet de icoon worden aangepast of herontworpen.






meer informatie


iconen en schermen - stap 1 van 3


Met het systeem waarin onderstaande iconen zullen worden gebruikt kunt u een videospeler van een bepaalde stijl voorzien en vervolgens op een website plaatsen. De iconen staan voor handelingen of navigatie onderdelen. Verschillende iconen kunnen dezelfde betekenis hebben.


Kunt u bij de volgende iconen aangeven wat het icoon volgens u betekent of welke associatie of gedachte u bij het icoon heeft?

Iconen










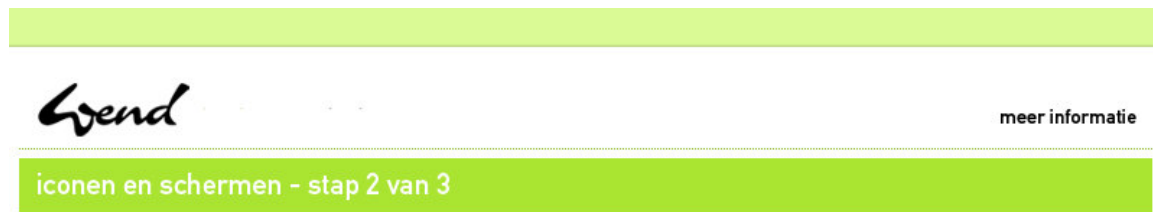


fig. 8.2 Iconen herkennen, stap 1 van het onderzoek naar iconen en schermen

### 8.2.2 Iconen en betekenissen koppelen

Het tweede deel van het onderzoek geeft antwoord op de deelvraag ‘In hoeverre herkennen gebruikers de iconen als de betekenissen bekend zijn?’. Een techniek die antwoord kan geven op deze vraag is ‘Image-meaning matching’ (The Icon Book). Hierbij worden de verschillende iconen weer gegeven, maar nu wordt er ook een lijst met betekenissen gegeven. Het is de bedoeling dat de testpersonen de betekenissen bij de iconen zoeken. Het is raadzaam om het aantal iconen en betekenissen niet gelijk te maken zodat de betekenissen niet door middel van eliminatie kunnen worden toegekend. Dit is in mijn onderzoek al opgelost omdat ik verschillende iconen voor dezelfde handeling heb gemaakt (zie figuur 7.12).



Met het systeem waarin onderstaande iconen zullen worden gebruikt kunt u een videospeler van een bepaalde stijl voorzien en vervolgens op een website plaatsen. De iconen staan voor handelingen of navigatie onderdelen. Verschillende iconen kunnen dezelfde betekenis hebben.

Bij de volgende iconen zijn mogelijke betekenissen gegeven. Kunt u aangeven welke betekenis volgens u de juiste is?

**Iconen**

	<input type="text" value="maak een keuze"/>	<b>Mogelijke betekenissen</b> 1 - afbeeldingen beheren 2 - afspeellijsten beheren 3 - formaat instellen 4 - help 5 - intro afbeelding kiezen 6 - kleuren kiezen 7 - knoppen selecteren 8 - opslaan 9 - profiel 10 - statistieken 11 - toepassen 12 - toevoegen 13 - toon voorbeeld 14 - verwijderen 15 - videos beheren 16 - videospeler afbeelding kiezen 17 - videospeler beheren 18 - videospeler plaatsen 19 - volgende 20 - vorige
	<input type="text" value="maak een keuze"/>	
	<input type="text" value="maak een keuze"/>	
	<input type="text" value="maak een keuze"/>	
	<input type="text" value="maak een keuze"/>	
	<input type="text" value="maak een keuze"/>	
	<input type="text" value="maak een keuze"/>	
	<input type="text" value="maak een keuze"/>	
	<input type="text" value="maak een keuze"/>	
	<input type="text" value="maak een keuze"/>	

fig. 8.3 Iconen en betekenissen koppelen, stap 2 van het onderzoek naar iconen en schermen

### 8.2.3 Vragen over schermen

De laatste stap betreft vragen over de werking van verschillende schermonderdelen. Dit onderdeel geeft antwoord op de deelvraag 'In hoeverre geven de schermen duidelijkheid over de wijze waarop taken uitgevoerd moeten worden?'. Daartoe heb ik een aantal taken opgesteld en de testpersonen gevraagd te vertellen welke handelingen ze zouden uitvoeren om de taak te volbrengen. Ik heb hiervoor taken opgesteld die wat complexer zijn dan een 'gemiddelde handeling' zoals het toevoegen van een afbeelding. Zo weet ik of ik op de goede weg ben wat betreft ontwerp en interactie. Als de testpersonen complexe handelingen weten uit te voeren is de kans groot dat minder gecompliceerde handelingen ook zullen lukken. Bovendien wordt dit nogmaals getest als de applicatie klaar is.

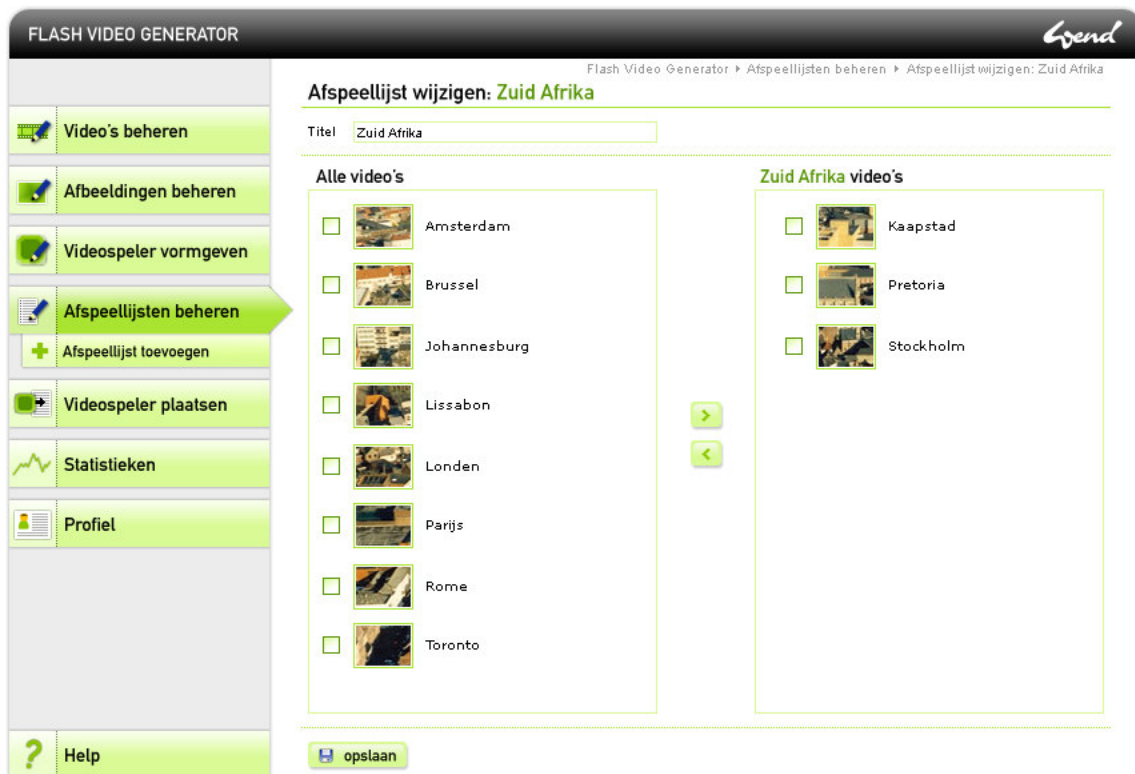


fig. 8.4 Testtaak 'Afspeellijst wijzigen', stap 3 van het onderzoek naar iconen en schermen



Bij bovenstaande afbeelding werd gevraagd om de video 'Johannesburg' aan de afspeellijst toe te voegen en de video 'Stockholm' uit de afspeellijst te halen.

### 8.3 Testresultaten verwerken

Deze paragraaf beschrijft de wijze waarop ik de testresultaten heb verwerkt.

#### 8.3.1 Iconen herkennen


Bij het verwerken van de testresultaten heb ik de nadruk gelegd op kwalitatieve verwerking. Dat wil zeggen dat ik vooral ga kijken naar de betekenissen die de testpersonen een icoon hebben toegekend. Mochten de testpersonen een icoon anders hebben geïnterpreteerd dan zal ik een manier moeten bedenken om dit misverstand te verhelpen. Daarnaast heb ik het percentage testpersonen berekend die de icoon de bedoelde betekenis toekende. Deze resultaten heb ik vervolgens in een overzicht geplaatst.

icoon	bedoelde betekenis	herkenningspercentage	geïnterpreteerde betekenis
	toevoegen	81,3 %	toevoegen; meer; combineren; drogist;
	opslaan	100,0 %	opslaan;
	wijzigen	62,5 %	wijzigen; tekenen / schrijven; aantekeningen;
	verwijderen	50,0 %	verwijderen; sluiten; stoppen / annuleren; vermenigvuldigen; wissen;
	verwijderen	75,0 %	verwijderen; vergrendelen;

*fig. 8.5 Testresultaten van de eerste stap 'Iconen herkennen', algemene iconen (gedeeltelijk)*

Hier valt op dat de icoon voor verwijderen weergegeven met een prullenbak beter scoort dan de icoon voor verwijderen weergegeven met een rood kruis. Als dit bij 'Iconen en betekenissen koppelen' ook het geval is zal de voorkeur uitgaan naar de icoon voor verwijderen weergegeven met een prullenbak.

Naar verwachting was het verschil tussen de algemene iconen en de applicatie specifieke iconen vrij groot.

icoon	bedoelde betekenis	herkenningspercentage	geïnterpreteerde betekenis
	video's beheren	56,3 %	video's bewerken / wijzigen; afbeelding toevoegen; ondertiteling schrijven; metadata plaatsen; film inkleuren; speellijst wijzigen; tijdlijn aanpassen; ?;
	afbeeldingen beheren	18,8 %	afbeeldingen bewerken / wijzigen; achtergrond bewerken; videospeler wijzigen; tekenen; titel geven; commentaar geven;
	afbeeldingen beheren	30,8 %	afbeeldingen bewerken / wijzigen; tekenen; beeld omkaderen;
	videospeler beheren	18,8 %	videospeler wijzigen / scherm aanpassen; afbeeldingen wijzigen; stemmen; titel geven; commentaar geven; ?;

*fig. 8.6 Testresultaten van de eerste stap 'Iconen herkennen', applicatie specifieke iconen (gedeeltelijk)*

Hierbij geldt weer dat de icoon voor afbeeldingen beheren weergegeven met een afbeelding van een bloem beter scoort en dus de voorkeur geniet.

Op basis van deze test en 'Iconen en betekenissen koppelen' zal ik per icoon een aantal aanbevelingen doen die de weergave verbeteren en dichterbij de verwachtingen van de gebruiker liggen. Hier kom ik op terug in paragraaf 'Iconen en betekenissen koppelen'.

### 8.3.2 Iconen en betekenissen koppelen

De verwerking van de testresultaten bij dit onderdeel verschilt niet van de verwerking van 'Iconen herkennen'.






icoon	bedoelde betekenis	herkenningspercentage	geïnterpreteerde betekenis
	toevoegen	92,3 %	toevoegen; toepassen; opslaan;
	opslaan	100,0 %	opslaan;
	wijzigen	69,2 %	wijzigen; afbeeldingen beheren; toepassen; knoppen selecteren; kleuren kiezen;
	verwijderen	84,6 %	verwijderen; volgende; video's beheren;
	verwijderen	100,0 %	verwijderen; zoeken;

fig. 8.7 Testresultaten van de tweede stap 'Iconen en betekenissen koppelen', algemene iconen (gedeeltelijk)

Wederom scoort de icoon voor verwijderen weergegeven met een prullenbak beter. In de applicatie zal ik dus deze icoon gebruiken, eventueel aangepast aan de hand van de aanbevelingen die ik ten tijde van deze verwerking nog moet doen.





icoon	bedoelde betekenis	herkenningspercentage	geïnterpreteerde betekenis
	video's beheren	38,5 %	video's beheren; kleuren kiezen; afbeeldingen beheren; videospeler beheren; afspeellijsten beheren;
	afbeeldingen beheren	46,2 %	afbeeldingen beheren; kleuren kiezen; intro afbeelding kiezen; videospeler beheren; videospeler afbeelding kiezen;
	afbeeldingen beheren	40,0 %	afbeeldingen beheren; kleuren kiezen; intro afbeelding kiezen; videospeler afbeelding kiezen;
	videospeler beheren	46,2 %	videospeler beheren; afbeeldingen beheren; knoppen selecteren; formaat instellen; kleuren kiezen; videospeler afbeelding kiezen;

fig. 8.8 Testresultaten van de tweede stap 'Iconen en betekenissen koppelen', applicatie specifieke iconen (gedeeltelijk)

Opvallend hier is dat de icoon voor afbeeldingen beheren weergegeven met een afbeelding van een groen vlak beter scoort dan de icoon weergegeven met een afbeelding van een bloem. Het verschil is hier echter significant kleiner waardoor ik toch kies voor de icoon voor afbeeldingen weergegeven met een afbeelding van een bloem.

Na voor beide stappen de overzichten te hebben gemaakt heb ik nagedacht over wijzigingen die de iconen beter op de doelgroep afgestemd kunnen maken. Dit resulteerde in onderstaand overzicht van aanbevelingen.

icoon	bedoelde betekenis	geïnterpreteerde betekenis	aanbevelingen
	wijzigen	wijzigen; s1: tekenen / schrijven; s1: aantekeningen; s1: afbeeldingen beheren; s2: toepassen; s2: knoppen selecteren; s2: kleuren kiezen;	potlood vervangen door steeksleutel;
	zoeken	zoeken; s1: vergroten; s2: toon voorbeeld;	-
	kleur kiezen	kleur kiezen; s1: snijden; s1: verwijderen; s2: formaat instellen; s2: verwijderen;	minder scherp; kleurvakjes; pipet vullen met kleur;
	help	help;	-
	videospeler beheren	videospeler beheren; s1: stemmen; s1: titel geven; s1: commentaar geven; s1 / s2: afbeeldingen beheren; s2: knoppen selecteren; s2: formaat instellen; s2: kleuren kiezen; s2: videospeler afbeelding kiezen;	videospeler voorzien van knoppen; content plaatsen; potlood vervangen door steeksleutel;

fig. 8.9 Aanbevelingen overzicht (gedeeltelijk)

Hierbij staat 's1' voor stap 1, 'Iconen herkennen' en 's2' voor stap 2, 'Iconen en betekenissen koppelen'.

De grootste wijziging is het vervangen van het potlood door een steeksleutel. Uit de tests is gebleken dat testpersonen bij gebruik van een potlood te snel de associatie met schrijven of tekenen leggen. De vervanging door een steeksleutel zou dit moeten verhelpen. Deze wijziging wordt doorgevoerd in alle iconen die een potlood bevatten.

Ook zijn er iconen die geen verdere uitwerking behoeven, bijvoorbeeld de icoon voor help. Voor anderen (bijvoorbeeld zoeken) heb ik geen aanbevelingen gegeven omdat ik denk dat een verdere of een meer gedetailleerde uitwerking van de icoon een averechts effect zal hebben.

### 8.3.3 Vragen over schermen

De testtaken werden boven verwachting goed gemaakt. Slechts enkele testpersonen hadden met de volgende twee vragen moeite.

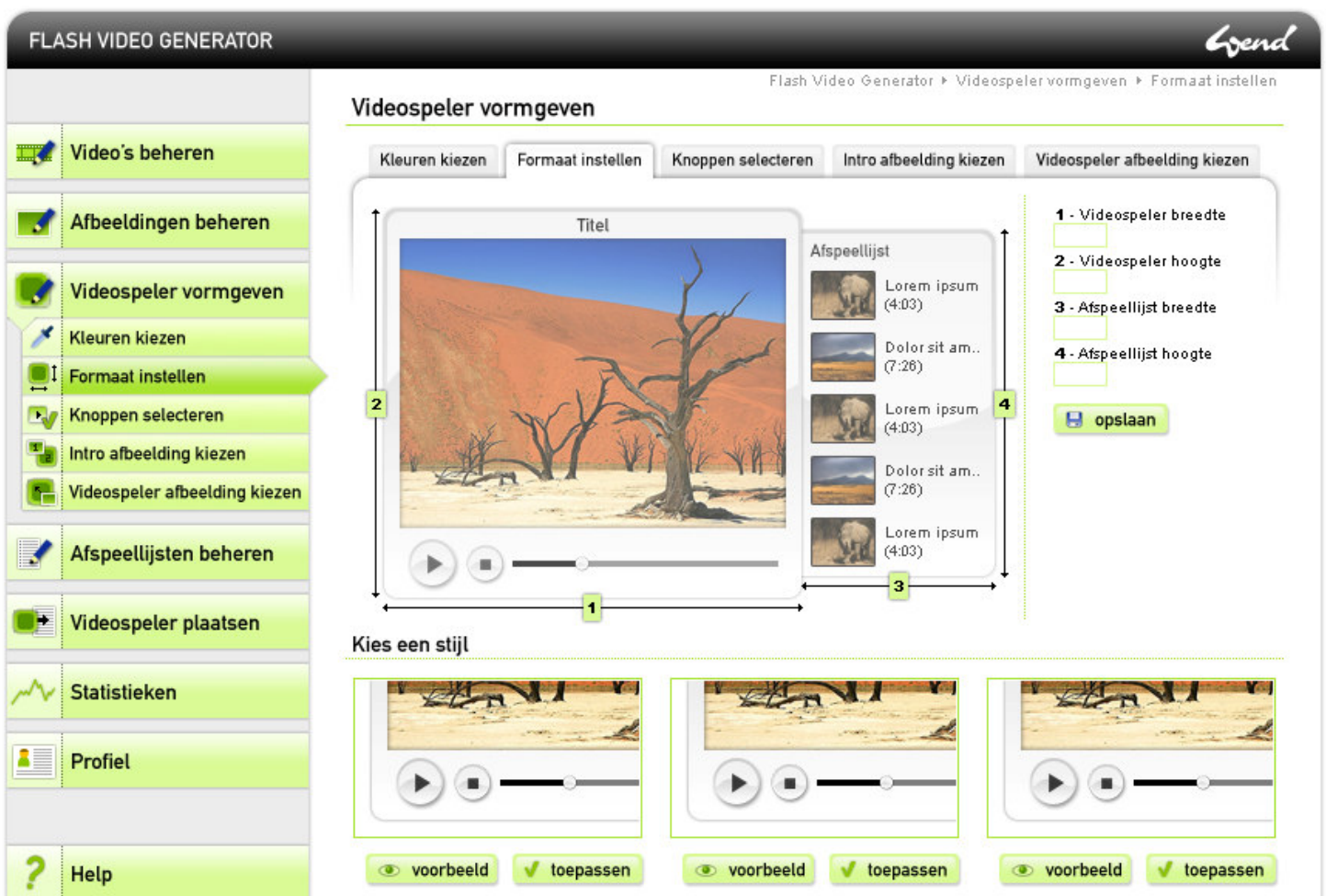


fig. 8.10 Vragen over schermen, 'Formaat instellen'

De testpersoon wordt gevraagd om de videospeler van een andere stijl te voorzien.



De testpersonen antwoordden het volgende.

actie	percentage
Klik op 'toepassen' bij een van de stijlen onderin	91,7 %
Klik op 'Videospeler vormgeven'	8,3 %

Een grote meerderheid weet dus direct hoe een nieuwe stijl kan worden gekozen.

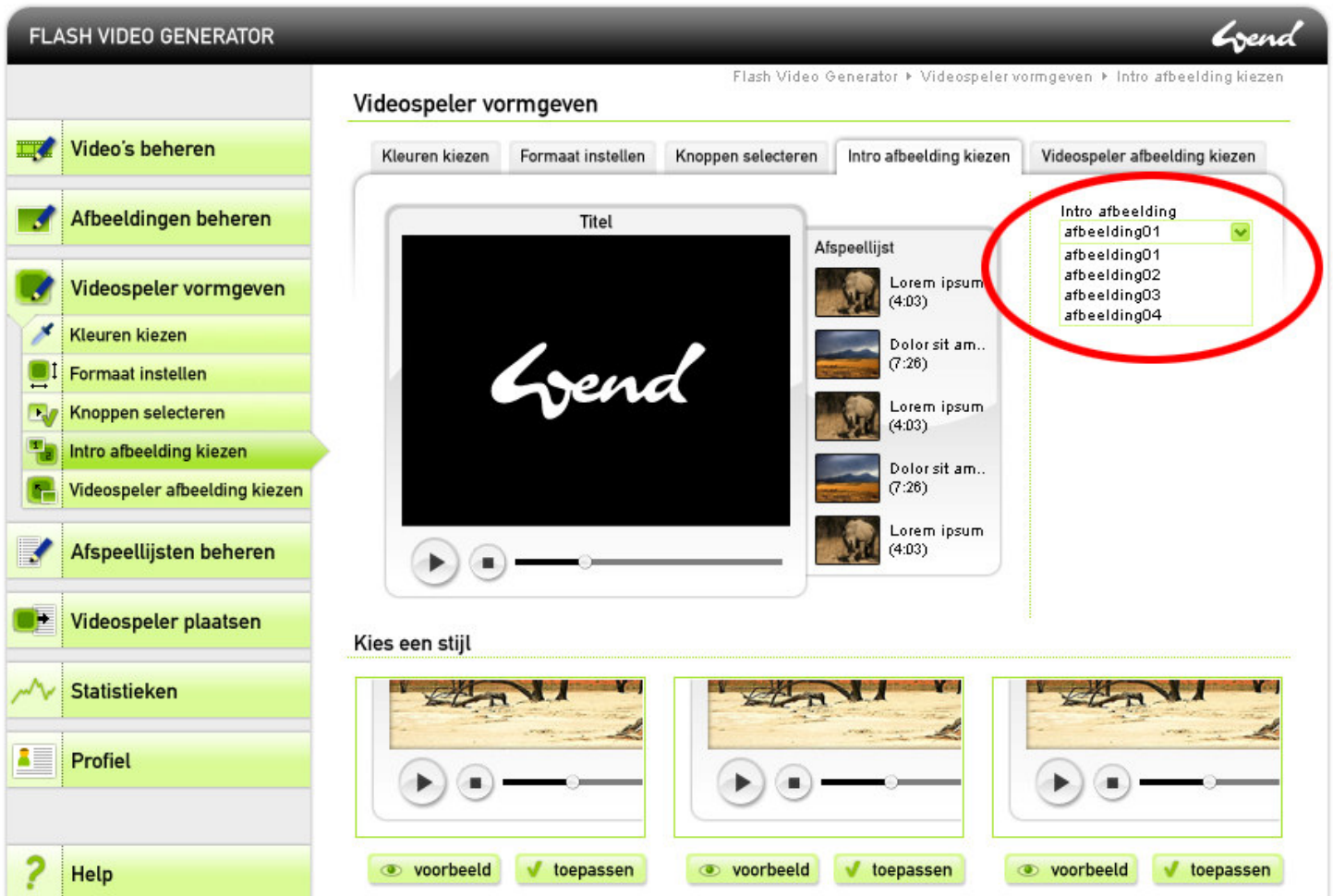


fig. 8.11 Vragen over schermen, 'Intro afbeelding kiezen'

De testpersoon wordt gevraagd waar de afbeeldingen in het aangegeven vlak beheerd kunnen worden.

De testpersonen antwoordden het volgende.

actie	percentage
'Afbeeldingen beheren' in menu	83,3 %
'Videospeler afbeelding kiezen'	8,3 %
geen idee	8,3 %

Een grote meerderheid begrijpt dus de relatie tussen deze afbeeldingen en 'Afbeeldingen beheren'.

Omdat bij beide vragen een grote meerderheid de vraag wel juist beantwoord heb ik besloten geen aanbeveling tot verbetering te doen.

## 9 VISUEEL ONTWERP UITWERKEN

Dit hoofdstuk beschrijft hoe ik met de resultaten van de usability test om ben gegaan.

### 9.1 Schetsen bijstellen

Nadat bleek dat een aantal iconen verbetering behoeften heb ik de reeds gemaakte schetsen bijgesteld. Enkele iconen zijn volledig opnieuw ontwikkeld. Een voorbeeld hiervan is de icoon voor 'wijzigen'. Dit was een weergave van een potlood maar is nu vervangen door een weergave van een steeksleutel. De wijzigingen in de schetsen zijn gebaseerd op de aanbevelingen die na het usability testen zijn geformuleerd.

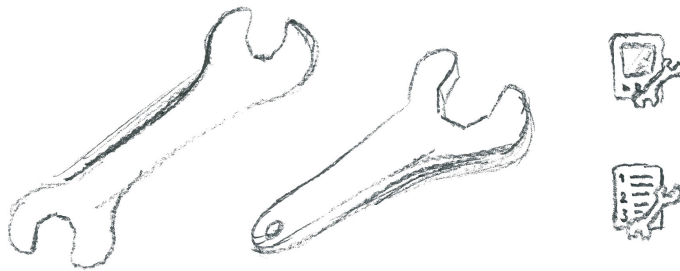


fig. 9.1 Steeksleutel, herziene schets van de handeling 'wijzigen'; rechts de iconen voor 'Videospeler beheren' en 'Afspeellijsten beheren'

Omdat de testpersonen de testtaken vrijwel allemaal gemakkelijk konden uitvoeren heb ik besloten de schermen niet opnieuw te schetsen, maar deze zo te houden.

### 9.2 Iconen herontwerpen

De opnieuw geschetste iconen heb ik vervolgens weer gedigitaliseerd. Voor een aantal iconen heb ik verschillende digitale uitwerkingen gemaakt. Een voorbeeld is de icoon voor 'verwijderen'. Uit de usability test bleek dat de handeling 'verwijderen' beter werd weergegeven met een prullenbak dan met een kruis. Daarom heb ik een aantal verschillende prullenbakken ontworpen die ik wederom aan de doelgroep zal voorleggen. Op basis van deze resultaten zal ik dan bepalen welk icoon ik zal gebruiken.



fig. 9.2 'Verwijderen', 4 versies

### 9.3 Iconen testen

Nadat ik de gewijzigde iconen had herontworpen heb ik ze per betekenis op een rijtje gezet en aan de doelgroep voorgelegd met de vraag welk iconen de handelingen het best omschreven. The Icon Book noemt dit 'comparative testing'. De twee grootste veranderingen, het vervangen van het potlood door de steeksleutel bij 'wijzigen' en beheer pagina's en het detailleren van de videospeler werden goed ontvangen.



*fig. 9.3 Wijzigen icoon aanpassing*



*fig. 9.4 Videospeler icoon aanpassing*

Maar liefst 76,5 % van de testpersonen koos de steeksleutel boven het potlood en 94,1 % gaf de voorkeur aan de meer gedetailleerde videospeler.

Met deze laatste stap is het visueel ontwerp afgerond. De schermen en iconen zijn aan de doelgroep voorgelegd en getest. Dit heeft gezorgd voor een ontwerp dat aansluit bij de verwachtingen van de doelgroep.

## 10 ONTWIKKELEN VAN DE APPLICATIE

Dit hoofdstuk beschrijft de gang zaken tijdens het ontwikkelen van de applicatie. Hierbij werd uiteraard veelvuldig gebruik gemaakt van zowel het visueel als het technisch ontwerp, maar heb ik ook de systeemeisen in de gaten gehouden zodat de uiteindelijke applicatie de afgesproken functionaliteiten bevat.

### 10.1 Technieken kiezen

Na het ontwerpproces moet het systeem worden ontwikkeld. Daarvoor moeten technieken worden gekozen waarmee het systeem gemaakt wordt. De techniek voor de videospeler zelf stond al voor het begin van het project vast; de videospelers zullen worden ontwikkeld in Flash, voor een Operating System en browser onafhankelijke weergave en hoge flexibiliteit.

Voor het systeem waarmee de videospelers kunnen worden beheerd is gekozen voor PHP in combinatie met een Apache server. PHP is een programmeertaal waarmee Wend veelvuldig werkt; ook het CMS waarmee Wend werkt is op PHP gebaseerd. Deze techniek is daarom een logische keuze omdat de medewerkers van Wend met deze taal overweg kunnen en het systeem dus eventueel kunnen aanpassen of doorontwikkelen.

Tot slot heb ik ervoor gekozen om het systeem op een Windows machine te plaatsen. Dit was noodzakelijk omdat het programma waarmee video formaten naar het juiste formaat (FLV) worden geconverteerd op Windows basis werkt.

Samengevat maak ik dus gebruik van de volgende technieken.

- Flash (videospelers)
- PHP en Apache (beheersysteem)
- Windows (server)

### 10.2 Basis van de applicatie leggen

Na de keuze van de technieken ben ik begonnen met het leggen van de basis van de applicatie. Ik heb de database aangemaakt volgens het technisch ontwerp. Vervolgens heb ik de bestandstructuur nagebouwd door volgens het technisch ontwerp lege bestanden te plaatsen. Daarna heb ik het ontwerp naar HTML vertaald. Deze HTML zal als template worden gebruikt waarin de verschillende pagina's worden getoond. Het systeem wordt zo gebouwd dat elke

pagina verwijst naar 'index.php'. Op deze PHP-pagina zal dan worden gekeken welke content getoond moet worden. Zo kan ik wijzigingen van gemeenschappelijke delen van de pagina's gemakkelijk doorvoeren; ik hoef deze wijzigingen slechts in 'index.php' door te voeren.

Nadat ik de HTML had ontwikkeld heb ik deze gekoppeld aan de database door in PHP een functie voor de database koppeling te schrijven. Daarnaast heb ik een aantal algemene functies geschreven voor uiteenlopende zaken als coderen en genereren van een wachtwoord en het uploaden van een bestand.

### 10.3 Klassen schrijven

Om het technisch ontwerp en de uiteindelijk applicatie overzichtelijk te houden heb ik gebruik gemaakt van klassen. Ik heb de klassen uit het klassediagram direct vertaald naar PHP klassen om het geheel consistent te houden.

Daarnaast heb ik gebruik gemaakt van de factory techniek. Deze techniek is gebaseerd op het 'echte leven' en wordt door veel programmeurs gebruikt. In het echte leven wordt een product gemaakt in een fabriek. De fabriek stelt de waarden van het product in en zorgt ervoor dat het product er goed uit komt te zien. Zo is het ook met de factory techniek. Er wordt een klasse (product) gedefinieerd, bijvoorbeeld Video. Deze klasse bevat alle eigenschappen van een Video object. Video heeft als eigenschappen onder andere 'label' en 'beschrijving'. Naast deze klasse wordt nog een klasse gemaakt die betrekking heeft op Video, de factory klasse. Deze factory kan video's maken, wijzigen en verwijderen, net als een fabriek in het echte leven. Op deze wijze worden de eigenschappen van een object (label, bestandstype) gescheiden van de functies die op een hoger niveau liggen en over het gehele object gaan (aanmaken, wijzigen). Dit zorgt voor overzicht en structuur.

```
class Video {
    var $id;                // unieke numerieke waarde waarmee video in het systeem wordt geïdentificeerd
    var $gebruiker_id;      // unieke numerieke waarde waarmee gebruiker in het systeem wordt geïdentificeerd
    var $titel;             // titel van de video
    var $label;            // label van de video
    var $breedte;          // breedte van de video
    var $hoogte;           // hoogte van de video
    var $aantal_keer_bekeken; // aantal keer video bekeken
    var $bestand;          // bestand bij uploaden

    //constructor
    function Video($id, $gebruiker_id, $titel, $label, $beschrijving, $breedte, $hoogte, $aantal_keer_bekeken, $bestand='') {
        $this->id = $id;
        $this->gebruiker_id = $gebruiker_id;
        $this->titel = $titel;
        $this->label = $label;
        $this->beschrijving = $beschrijving;
        $this->breedte = $breedte;
        $this->hoogte = $hoogte;
        $this->aantal_keer_bekeken = $aantal_keer_bekeken;
        $this->bestand = $bestand;
    }
}
```

fig. 10.1 Klasse 'Video' (gedeeltelijk), bevat de eigenschappen van een Video object

```
class VideoFactory {

    var $db;                // instantie van de database klasse

    //constructor
    function VideoFactory() {
        $this->db = new Database();
    }

    function geefVideoViaId($id) {
        $this->db->sqlQuery("SELECT * FROM ".DB_VIDEO." WHERE id = '$id'");
        $db_video = $this->db->sqlFetchArray();
        $video = new Video($db_video['id'], $db_video['gebruiker_id'], $db_video['titel'], $db_video['label'], $db_video['beschrijving'], $db_video['breedte'], $db_video['hoogte'], $db_video['aantal_keer_bekeken'], $db_video['bestand']);
        return $video;
    }

    function geefVideosVanAfspeellijst($afspeellijst_id) {
        $video_ids = '';
        $this->db->sqlQuery("SELECT * FROM ".DB_AFSPEELLIJSTVIDEO." WHERE afspeellijst_id = '$afspeellijst_id'");
        while ($db_afspeellijstvideo = $this->db->sqlFetchArray()) {
            $video_ids .= $db_afspeellijstvideo['video_id'].",";
        }
        $video_ids = substr($video_ids,0,-1);
        $video_ids = ($video_ids == '') ? "" : $video_ids;
        $videos = array();
        $resultaten = $this->db->sqlQuery("SELECT * FROM ".DB_VIDEO." WHERE id IN (".$video_ids.") ORDER BY label");
        while ($db_video = $this->db->sqlFetchArray($resultaten)) {
            $video = $this->geefVideoViaId($db_video['id']);
            array_push($videos,$video);
        }
        return $videos;
    }
}
```

fig. 10.2 Klasse 'VideoFactory' (gedeeltelijk), voert bewerkingen van een Video object op een hoger niveau uit

Om de wijze van programmeren zo consistent mogelijk te houden heb ik bepaalde programmeer conventies aangehouden. Zo is er bijvoorbeeld de variabele 'videospeler\_beheren', deze variabele geeft aan of een gebruiker toegang heeft tot de pagina's onder 'Videospeler beheren'. Om de waarde van deze variabele op te vragen moet er een functie worden gedefinieerd. Voor de naam van deze functie heb ik gekozen voor 'getVideospelerBeheren'. De variabele wordt dus

als het ware vertaald naar de bijbehorende functie waarbij in de variabele naam spaties worden vervangen door underscores (\_) en in de functie spaties worden vervangen door het gebruik van een hoofdletter bij het volgende woord. Deze conventie wordt bij soortgelijke gevallen in het hele systeem gebruikt zodat er eigenlijk niet hoeft te worden opgezocht of nagedacht bij het opvragen van de waarde van een variabele.

Omdat de gehele applicatie object georiënteerd geprogrammeerd is heb ik besloten van de database koppeling ook een object te maken. Daarnaast zijn er nog twee klassen die er toe doen, de Functions klasse en de Validator klasse. In de Functions klassen worden alle algemene functies gedefinieerd, zoals het uploaden van een bestand en het genereren van een wachtwoord. In de Validator klasse worden alle controle functies gedefinieerd; deze functies geven aan of een gegeven waarde voldoet aan de eisen die voor dat type variabele gelden. Zo is er bijvoorbeeld een functie die controleert of een gegeven emailadres daadwerkelijk volgens het stramen van een emailadres is opgebouwd.

#### **10.4 Pagina's ontwikkelen**

Nadat alle klassen waren geschreven is het eigenlijk een kwestie van alle verschillende pagina's opmaken en met behulp van de klassen de communicatie regelen. Bij dit onderdeel wordt er weer veel gebruik gemaakt van HTML. Hierbij heb ik rekening gehouden met de twee browsers die door gebruikers het meest worden gebruikt, Internet Explorer en Mozilla Firefox. Door het systeem voor beide browsers geschikt te maken zullen minder gebruikers problemen met het systeem ervaren.

Zoals ik had verwacht bleek bij het implementeren van de klassen dat een en ander beter op een andere manier ontwikkeld kan worden. Er zijn enkele wijzigingen in de klassen aangebracht om een logischer geheel te vormen. Zo is bijvoorbeeld het bijhouden van de locatie van een afbeelding komen te vervallen. Een object heeft immers altijd een uniek nummer waarmee het object wordt geïdentificeerd, het 'id'. Ik heb het systeem zo gemaakt dat de naam van een afbeelding die wordt geupload wordt gelijk is aan het id. De afbeelding 'logo.jpg' wordt bijvoorbeeld '37.jpg', vernoemd naar een object met het id 37. Aangezien alle geplaatste bestanden in voorgedefinieerde mappen komen te staan kan ik dus simpelweg het id van het object gebruiken om de locatie van een afbeelding te achterhalen. Dit soort wijzigingen vallen binnen de planning; er is van te voren rekening mee gehouden.



Dit onderdeel is naast het ontwikkelen van de pagina's in HTML dus ook heen en weer ontwikkelen tussen de pagina's en de klassen, net zolang tot het geheel consistent en logisch in elkaar steekt.

```
$filesize = $gebruikerFactory->geefGebruikteSchijfruimteVanGebruiker($gebruiker->getId(), false);
$videos = $gebruiker->video_factory->geefVideosVanGebruiker($gebruiker->getId());
echo Functions::introMsg($page);
?>
<table width="597" border="0" cellspacing="0" cellpadding="0" class="normal" style="line-height:20px;">
<tr>
<td width="250">&nbsp;</td>
<td width="115" style="font-weight:bold; font-size:11px;">Gebruikt</td>
<td width="30">&nbsp;</td>
<td style="font-weight:bold; font-size:11px;">Restrictie</td>
</tr>
<tr>
<td>Aantal video's online</td>
<td><?php echo count($gebruiker->video_factory->geefVideosVanGebruiker($gebruiker->getId())); ?></td>
<td>&nbsp;</td>
<td><?php echo $gebruiker->getMaxVideos(); ?></td>
</tr>
<tr>
<td>Gebruikte hoeveelheid schijfruimte video's</td>
<td><?php echo $filesize[0]; ?> mb</td>
<td rowspan="2"></td>
<td rowspan="2"><?php echo $gebruiker->getMaxSchijfruimte(); ?> mb</td>
</tr>
<tr>
<td>Gebruikte hoeveelheid schijfruimte afbeeldingen</td>
<td><?php echo $filesize[1]; ?> mb</td>
</tr>
</table>
```

fig. 10.3 Klassegebruik in HTML pagina's, in dit geval de pagina 'Statistieken'; de gemarkeerde tekst geeft weer hoe de klassen worden gebruikt

## 10.5 Videospeler ontwikkelen

Gelijktijdig met het ontwikkelen van de pagina's heb ik de videospeler in Flash ontwikkeld. Ik heb hierbij gekozen voor een dynamische opzet; zoveel mogelijk variabelen worden van te voren gedefinieerd. Zo worden bijvoorbeeld de verschillende onderdelen van de videospeler per kleur (die de ontwikkelaar kan aangeven, zie figuur 10.4) in een rij gezet. Een script (figuur 10.5) zorgt er later in de tijdlijn voor dat die onderdelen ook daadwerkelijk de opgegeven kleur krijgen. Door deze dynamische opzet kan Wend de ontwikkelde videospeler als template gebruiken voor het ontwikkelen van verschillende stijlen; alleen de visuele elementen hoeven aangepast te worden. Wend kan dan zelf aangeven welke onderdelen onder welk kleurlabel vallen zonder de achterliggende code te hoeven begrijpen.

Dit geldt ook voor de onderdelen die moeten worden vergroot bij een opgegeven formaat dat groter is dan het formaat waarop de videospeler oorspronkelijk ontworpen is. Van te voren kan worden opgegeven welke onderdelen in welke richting moeten worden vergroot of verplaatst.

```
var colors:Array = Array();  
//instance names  
colors[1] = Array('title'); //title  
colors[2] = Array('bgtop'); //video window top  
colors[3] = Array('bgbottom','fullscreenControlBar'); //video window bottom
```

*fig. 10.4 Kleuren aan onderdelen koppelen; de ontwikkelaar kan hier de namen van onderdelen invullen en zo aan een kleur koppelen*

```
for (i = 1; i <= 8; i++) {  
    for (j = 0; j < colors[i].length; j++) {  
        toAlter = getMcByName(colors[i][j]);  
        color = new Color(toAlter);  
        color.setRGB("0x"+_xml.data["videoplayer"] ["colors"] ["color_"+i]);  
    }  
}
```

*fig. 10.5 Kleur van de aangegeven onderdelen worden later in de tijdlijn automatisch veranderd*

Zo hoeft Wend dus nauwelijks technische know-how van de videospeler te hebben; door het opgeven van enkele variabelen wordt de videospeler aangestuurd en kunnen verschillende stijlen gemakkelijk worden toegevoegd.

Bij het implementeren van de videospeler liep ik tegen een technisch probleem op. Het systeem is zo gebouwd dat de videospeler zowel transparantie in de achtergrond als een fullscreen functie ondersteunt. Flash heeft echter de beperking dat deze twee functionaliteiten niet gecombineerd kunnen worden. Normaal gesproken wordt een fullscreen Flash element opgevuld met de aangegeven achtergrond kleur. Als er wordt aangegeven dat de achtergrond kleur niet getoond moet worden weet Flash niet waarmee het scherm moet worden opgevuld. Dit had tot gevolg dat de applicatie aangepast moest worden, gebruikers moeten deze beperking immers begrijpen. Daarom heb ik in het beheersysteem een tekst bij het selecteren van de knop 'fullscreen' toegevoegd die de gebruiker attendeert op deze beperking. Daarnaast heb ik besloten om tijdens de usability test te kijken of gebruikers de beschreven beperking daadwerkelijk begrijpen.

## 11 USABILITY TESTEN VAN DE GEHELE APPLICATIE

Dit hoofdstuk beschrijft hoe ik na de ontwikkeling van de applicatie het systeem heb getest op usability.

### 11.1 Doel bepalen

Net als bij de vorige test (zie hoofdstuk 8 'Usability testen van iconen en schermen') ben ik begonnen met het formuleren van het doel van het onderzoek. Het doel van dit onderzoek is bepalen of de doelgroep met de uiteindelijke applicatie overweg kan en of het systeem voldoet aan de verwachtingen van de gebruikers. Wederom heb ik dit doel vertaald naar een hoofdvraag die vervolgens weer uiteen viel in een aantal deelvragen.

**“Komen de werking en het ontwerp van de applicatie voldoende overeen met de verwachtingen van de gebruiker?”**

*fig. 11.1 Hoofdvraag van het onderzoek naar de usability van de applicatie*

- *Hoe effectief is de applicatie?*
- *Hoe efficiënt is de applicatie?*
- *In hoeverre leert men van de applicatie?*
- *In hoeverre geeft het werken met de applicatie voldoening en tevredenheid?*

Corresponderend met bovenstaande deelvragen worden de volgende variabele met behulp van testtaken en vragenlijsten geoperationaliseerd.

- *Effectiviteit*
- *Efficiëntie*
- *Leerbaarheid*
- *Tevredenheid*

### 11.2 Testtaken opstellen

Om de vier variabelen te kunnen meten heb ik twaalf testtaken opgesteld. De taken gaan over verschillende handelingen die binnen het systeem worden uitgevoerd. De testtaken zijn zo bedacht dat de verschillende principes waarop het systeem berust erin voorkomen.

Bij elke taak wordt aangegeven of de testpersoon deze heeft volbracht. Dit geeft inzicht in de *effectiviteit* van de applicatie.

Om de *efficiëntie* te meten wordt per taak bijgehouden hoeveel tijd en muisklikken de gebruiker nodig heeft. Vervolgens wordt de tijd vergeleken met de vooraf gestelde streeftijd en de muisklikken met de minimale hoeveelheid klikken die nodig is om de taak te volbrengen. De streeftijd is gebaseerd op de tijd die ik als ontwikkelaar nodig had om de taak uit te voeren en ik acceptabel achtte om de handeling te verrichten.

Ook heb ik een aantal vergelijkbare testtaken opgesteld. Door de tijd die de gebruiker nodig heeft om de taken te volbrengen met elkaar te vergelijken kan worden vastgesteld of de gebruiker leert van de applicatie en dus hoe het systeem qua *leerbaarheid* scoort.

Bij verschillende taken zal ook gebruik worden gemaakt van de 'cognitive walkthrough'. De gebruiker wordt bij het uitvoeren van een taak gewezen op de verschillende handelingen die moeten worden gedaan om de taak te volbrengen. Bij de taken waar dit niet het geval is wordt alleen het te behalen resultaat van de taak vermeld.

cognitive walkthrough	geen cognitive walkthrough
Navigeer naar de pagina 'Videospeler beheren'. Navigeer naar de subpagina 'Intro afbeelding kiezen'. Selecteer de afbeelding 'intro afbeelding'. Zet het aantal seconden op 6. Klik op de knop 'opslaan'.	Kies een andere stijl voor de videospeler.

fig. 11.2 Testtaken; links in stappen, rechts alleen het gewenste eindresultaat

### 11.3 Vragenlijsten opstellen

Nadat de gebruiker de testtaken heeft doorlopen zal ik hem of haar een aantal vragen met betrekking tot de applicatie stellen. Hieruit moet blijken of een aantal zaken uit de applicatie duidelijk is geworden. Zo wordt er bijvoorbeeld gevraagd '*Wat doet de 'plaatsen' knop op de pagina 'Afspeellijsten beheren'?*'. Als de gebruikers de vraag juist beantwoorden dan weet ik dat dit onderdeel door de gebruikers wordt begrepen. Beantwoorden de gebruikers de vraag niet juist dan moet ik op zoek gaan naar een oplossing.

Na enkele inhoudelijke vragen zal de *tevredenheid* worden gemeten door de gebruikers een lijst met stellingen voor te leggen. De lijst bevat stellingen als 'Het gebruik van dit programma is frustrerend' en 'De wijze waarop de systeemberichten worden weergegeven is duidelijk en begrijpbaar'. Deze lijst met stellingen is gebaseerd op een combinatie van de standaard lijsten Software Usability Measurement Inventory (SUMI) en Website Analysis and MeasureMent Inventory (WAMMI). Deze lijsten zijn door een groot aantal ontwikkelaars al ontelbaar keren gebruikt en worden gezien als een beproefde techniek om de tevredenheid van een systeem te meten. Ik heb de lijsten niet klakkeloos aan de gebruikers voorgelegd maar eerst doorgelopen en alle relevante onderdelen geselecteerd. De combinatie van de twee lijsten en selectie van relevante onderdelen heeft geleid tot een lijst met stellingen die afgestemd is op de ontworpen applicatie.

*Ik kan de berichten van dit computerprogramma begrijpen en er iets mee doen.*

☐ eens ☐ geen mening ☐ oneens

*Er moet veel gelezen worden voordat ik het computerprogramma kan gebruiken.*

☐ eens ☐ geen mening ☐ oneens

*De taken kunnen zonder moeilijkheden worden uitgevoerd met dit computerprogramma.*

☐ eens ☐ geen mening ☐ oneens

*Het gebruik van dit computerprogramma is frustrerend.*

☐ eens ☐ geen mening ☐ oneens

*fig. 11.3 Vragenlijst (gedeeltelijk; SUMI / WAMMI) waarbij de variabele tevredenheid wordt geoperationaliseerd*

## 11.4 Testinhoud verzorgen

Bij het testen moeten verschillende content gerelateerde handelingen worden verricht. Er worden afbeeldingen en video's toegevoegd en gewijzigd, afspeellijsten aangemaakt en informatie ingevoerd. Het gebruik van de applicatie wordt het beste getest als de situatie gelijk is aan hoe de situatie 'in het echt' zou zijn, dus als mensen daadwerkelijk met het systeem werken. Later zullen gebruikers het systeem in ontspannen toestand gebruiken, zonder dat iemand allerlei aantekeningen maakt en de tijd bijhoudt. De test situatie moet daar zoveel mogelijk op lijken.

Ik heb daarom content gekozen die bij de meeste personen uit de doelgroep bekend is. Dit zorgt ervoor dat de applicatie bij gebruikers vertrouwen wekt; de herkenning zorgt ervoor dat de gebruikers iets minder bezig zijn met een testtaak zo goed mogelijk uitvoeren en zich meer ontspannen voelen. Zo heb ik als video's trailers van bekende animatiefilms als Shrek en Toy Story gekozen. Deze content is bij jong en oud bekend en wordt door veel mensen gewaardeerd.



fig. 11.4 Testinhoud; er is gekozen voor animatiefilms dat voor herkenning bij de testpersonen zorgt

### 11.5 Testresultaten verwerken

Omdat ik tijdens de test zaken die aandacht behoeften heb genoteerd op beoordelingsformulieren, kan ik deze gegevens verwerken en vertalen naar aanbevelingen die de applicatie ten goede komen.

Tevens heb ik de testpersonen ten behoeve van de verwerking van de testresultaten vooraf geïnstrueerd om hardop te denken, de think-aloud techniek. Zo heb ik vastgesteld waar de gebruikers moeite mee hebben en welke zaken juist gemakkelijk gaan. Bij het testen heb ik gebruik gemaakt van opname apparatuur wat me in staat stelt om de gedachten en handelingen op een later tijdstip - nu, tijdens het verwerken van de resultaten - terug te luisteren.

Bij het verwerken van de test heb ik eerst alle resultaten beschrijvend weergegeven. Vervolgens heb ik betekenis aan de resultaten gegeven door aanbevelingen voor problemen te formuleren.

Bij het uitvoeren van de testtaken bleek bijvoorbeeld dat de gebruikers vrij veel moeite hadden om een andere stijl voor de videospeler te selecteren. Na doorvragen bleek dat het onderscheid tussen de verschillende gepresenteerde stijlen onvoldoende was. Ook was de help informatie niet toereikend. Voor dit probleem heb ik de aanbeveling 'Toevoegen van een label aan een stijl. Kleuren van de originele SWF gebruiken.' geformuleerd. De kleuren en label zorgen ervoor dat de stijlen

meer op zich zelf staande elementen worden die sneller door gebruikers als verschillende stijlen worden herkend.

Slechts een enkeling wist het antwoord op de vraag '*Wat doet de 'plaatsen' knop op de pagina 'Afspeellijsten beheren'?*', de andere testpersonen wisten niet waar de knop voor diende en voor sommigen zorgde de knop zelfs voor verwarring. De relatie tussen de icoon op de knop en de icoon in het hoofdmenu (zie figuur 7.14) blijkt dus onvoldoende duidelijk te zijn. De knop diende als shortcut en is voor Wilfried ('Ik wil *snel resultaat* zien. Het systeem moet werken. Daarbij is vormgeving een pré, maar geen must.') in het leven geroepen. De knop bracht Miriam (en ook sommige testpersonen die ik als Wilfried beschouw) in verwarring. Aangezien Miriam de primaire persona is en de knop geen essentiële functionaliteit betreft heb ik aanbevolen de knop te verwijderen.

Uit de antwoorden op de inhoudelijke applicatie vragen bleek ook dat de relatie tussen de (transparante) achtergrond en de fullscreen functie (zie paragraaf 10.5 'Videospeler ontwikkelen') niet duidelijk was. De aanbeveling die ik daarom heb gedaan is dat de gebruiker naast aangeven of de fullscreen knop beschikbaar moet zijn ook moet kunnen aangeven of de achtergrond transparant moet zijn. Door deze twee aanvinkvakjes vervolgens te koppelen door de fullscreen knop uit te schakelen als de achtergrond transparantie aan staat en vice versa is het voor gebruikers beter te begrijpen dat beide functionaliteiten niet te combineren zijn.

Ook de lijst met stellingen heeft aanbevelingen teweeggebracht. Een aantal testpersonen die ik als Miriam beschouw hebben aangegeven te weinig uitleg in de vorm van introductie teksten te hebben aangetroffen. Ik heb daarom aanbevolen elke pagina te voorzien van introductie teksten.

Zo heeft elk onderdeel uit de test verschillende aanbevelingen teweeggebracht. In het volgende hoofdstuk wordt beschreven op welke wijze ik de aanbevelingen heb opgevolgd.

## 12 HERZIEN VAN DE APPLICATIE

In de vorige fase zijn enkele aanbevelingen geformuleerd. Dit hoofdstuk beschrijft hoe ik deze aanbevelingen heb opgevolgd.

### 12.1 Delen van de applicatie herontwikkelen

In deze laatste fase worden de aanbevelingen uit de vorige fase opgevolgd waarna het product wordt opgeleverd. Deze fase is eigenlijk een kwestie van toepassen, het denkwerk heeft in de vorige fase bij het formuleren van de aanbevelingen plaatsgevonden. Ik moet uiteraard wel nog steeds rekening houden met Miriam en Wilfried en is het van belang dat wijzigingen in de stijl van de applicatie worden doorgevoerd. Ook is het zaak dat het systeem consequent werkt en blijft werken. Hierna volgen twee voorbeelden van hoe ik de aanbevelingen heb toegepast.

Eén van de aanbevelingen was om introductie teksten toe te voegen. De pagina 'Video's beheren' zag er vóór het opvolgen van de aanbeveling als volgt uit.

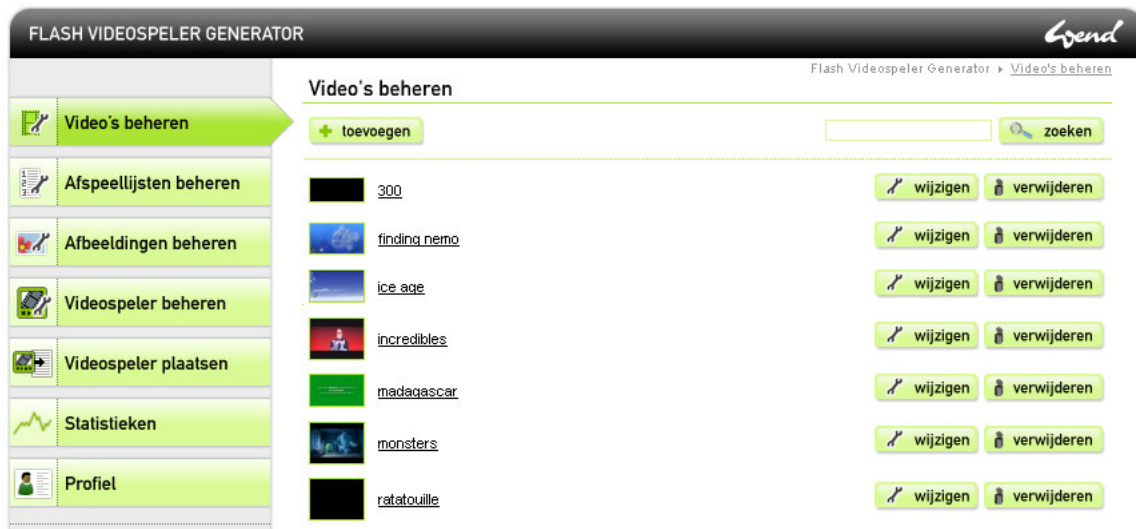


fig. 12.1 Introductie teksten nog toe te voegen op de pagina 'Video's beheren'

Uit het onderzoek bleek tevens dat vooral de testpersonen die ik als Miriam beschouw om introductie teksten vroegen. De testpersonen die ik als Wilfried beschouw gaven dit niet aan. Deze laatste groep zou de introductie teksten zelfs hinderlijk kunnen vinden. Daarom heb ik ervoor gekozen om de introductie teksten wel toe te voegen, maar bied ik gebruikers ook de mogelijkheid om de teksten ('tips') te minimaliseren. Deze keuze zal gedurende het gebruik van



de applicatie worden onthouden zodat de gebruiker niet op elke pagina de teksten weer hoeft te minimaliseren.



fig. 12.2 Introductie teksten ('tips') toegevoegd en gemaximaliseerd; rechtsboven heeft de gebruiker de mogelijkheid om de tips te minimaliseren

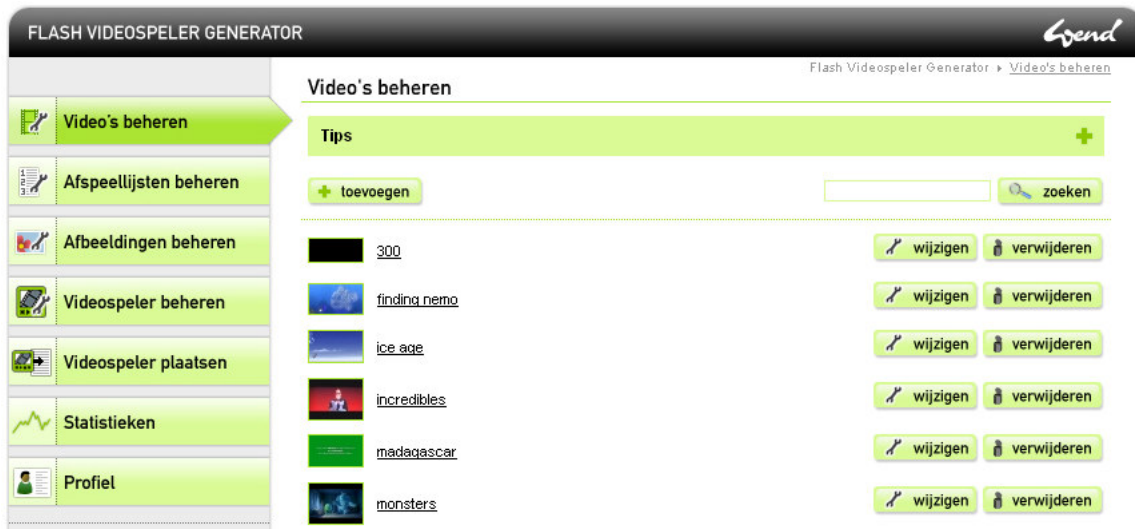
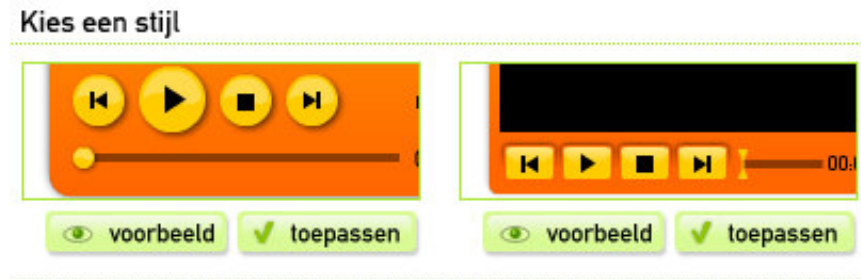


fig. 12.3 Introductieteksten ('tips') toegevoegd en geminimaliseerd; rechtsboven heeft de gebruiker de mogelijkheid om de tips te maximaliseren

Een andere aanbeveling betrof het onderscheid tussen de beschikbare stijlen. De aanbeveling was het toevoegen van labels en onderscheid in kleuren. Voor de kleuren moeten de originele kleuren van de SWF worden aangehouden. Dit is er als volgt uit komen te zien.



*fig. 12.4 Stijlkeuze vóór opvolging van de aanbeveling; geen labels en de gebruikte kleuren komen uit de videospelerconfiguratie*



*fig. 12.5 Stijlkeuze na het plaatsen van labels en het gebruiken van de originele kleuren*

Op deze manier is er onderscheid tussen de stijlen gemaakt. Wat betreft vormgeving vind ik de wijze van figuur 12.4 beter, maar omdat veel gebruikers moeite hadden met de stijlkeuze heb ik er toch voor gekozen om de aanbeveling op te volgen. Gebruiksvriendelijkheid heb ik in dit geval dus boven ontwerp geplaatst.

Met het doorvoeren van alle eerder geformuleerde aanbevelingen is de ontwikkeling van de applicatie voltooid.

## 12.2 Overdrachtsverslag schrijven

In hoofdzaak is het schrijven van het overdrachtsverslag voor eventuele vervolgon ontwikkelaars van de applicatie. Het systeem wordt in de opgeleverde vorm in kaart gebracht zodat duidelijk is welke elementen waar staan en hoe de communicatie tussen deze elementen in elkaar steekt.

Daarnaast is het voor mij interessant om te zien bij welke onderdelen ik van het initiële technische ontwerp ben afgeweken en waarom. Dit stelt me in staat om een volgende project beter te ontwerpen zodat er minder van het ontwerp hoeft te worden afgeweken.

Een voorbeeld van een wijziging is het weglaten van het veld 'locatie' in de video tabel. Zoals eerder beschreven (zie paragraaf 10.4 'Pagina's ontwikkelen') heb ik ervoor gekozen om de locatie van videobestanden - eerder opgeslagen in het veld 'locatie' - gelijk te stellen aan het id van de video (bijvoorbeeld 37.flv). Dit werkt efficiënter dan eerst het veld 'locatie' uitlezen bij het ophalen van een video waarbij het 'id' sowieso al bekend is. Bij een soortgelijk project zal ik dit dus ook op deze manier ontwerpen.

Om eventuele vervolgon ontwikkelaars tegemoet te komen heb ik in het overdrachtsverslag ook een lijst van mogelijke uitbreidingen gegeven; enkele volgen in onderstaande lijst.

- Video's bewerken.
- Positie bedieningsknoppen instellen.
- Afbeeldingen selecteren voor bedieningsknoppen.
- Meerdere videospeler configuraties samen kunnen stellen.
- Mogelijkheid om videospelers in Flash sites te implementeren.

## 13 EVALUATIE

Dit hoofdstuk beschouwt het opgeleverde product en blikt terug op het doorlopen proces. Tot slot volgt de conclusie.

### 13.1 Product

Ik denk dat het opgeleverde product geslaagd is. Het voldoet aan alle vooraf gestelde systeemeisen. Daarnaast is de doelgroep actief bij het project betrokken wat ervoor zorgt dat de applicatie voldoet aan de verwachtingen van de gebruiker.

Hoewel de opdrachtgever aanvankelijk meer functionaliteit en mogelijkheden wilde zien denk ik dat het opgeleverde product van een hogere kwaliteit is dan wanneer ik meer functionaliteit zou hebben ingebouwd. Het bouwen van deze extra functionaliteit (onder andere de 'Won't have' systeemeisen) zou veel tijd in beslag nemen en weinig ruimte laten voor het testen van de applicatie op usability. Ik heb de opdrachtgever ervan weten te overtuigen om een kwalitatief hoogwaardiger product op te leveren, in plaats van een product met meer mogelijkheden.

De opdrachtgever heeft aangegeven tevreden te zijn over het product, zowel qua ontwerp als functioneren. Het toevoegen van 'skins' is gemakkelijk uit te voeren en gebruikers zijn gemakkelijk te beheren. De opdrachtgever heeft wel aangegeven dat voor ingebruikname enkele minimale aanpassingen nog zouden moeten worden doorgevoerd. Dit gaat met name over het bijhouden van dataverkeer ('Won't have' systeemeis) voor een betere registratie van het gebruik van de applicatie. Op basis van het gebruik moet gebruikers immers gefactureerd worden.

Diepgang in het product zit vooral in de wijze van programmeren; het gebruik van klassen en de factory methode (zie paragraaf 10.3 'Klassen schrijven'). Deze manier van programmeren sluit aan bij de manier die Wend hanteert voor haar eigen klanten. Zo kunnen eventuele vervolgon ontwikkelaars van Wend ook met de structuur van de applicatie overweg.

Daarnaast zit er ook een stuk diepgang in het technisch ontwerp. Hierbij is gebruik gemaakt van een combinatie van klassen en klassediagram, usecases en usecasediagram en sequentiediagrammen. Deze manier van (technisch) ontwerpen is me erg goed bevallen en ik denk dat het de kwaliteit van de applicatie op een positieve manier sterk beïnvloedt heeft.

Samengevat, de geformuleerde doelstelling van het project is behaald:

- er kan een bredere doelgroep worden aangesproken;
- het systeem genereert videospelers naar wens van de gebruiker;
- Wend hoeft niet voor elke klant een afzonderlijke videospeler in de juiste stijl te ontwikkelen;
- het product biedt een hogere mate van vrijheid in vormgeving en uitstraling.

### 13.2 Proces

Ik denk dat het proces volgens het boekje is gegaan. Er is een duidelijke en complete fase doorloop geweest: onderzoek → ontwerp → usability test (en verwerking): ontwerp → ontwikkeling → usability test (en verwerking): applicatie → overdracht. Dit heeft ervoor gezorgd dat het gedurende het project voor mij, als ontwikkelaar, altijd duidelijk is geweest op welk punt ik me in planning en proces bevond. Daarnaast denk ik dat het ook voor de opdrachtgever helder is geweest met welke fases ik bezig ben geweest.

Diepgang in het proces zit vooral in de keuze voor het usability testen van zowel ontwerp als applicatie. Ik heb de doelgroep veel bij het project betrokken voor een optimale vormgeving en werking van het systeem. De testpersonen die ik heb benaderd vallen binnen de doelgroep en verschillende maken deel uit van de klantenkring van Wend. Ik ben tevreden over de hoeveelheid respondenten en de wijze waarop de tests zijn verlopen. Met name de laatste testfase waarbij ik bij de test aanwezig ben geweest heeft tot een aantal vernieuwde inzichten en nuttige tips geleid die de applicatie op verschillende punten hebben verbeterd.

Ook ben ik tevreden over de manier waarop het projectmanagement is gegaan. De methode die ik hiervoor heb gehanteerd heeft me geholpen het project in goede banen te leiden en heeft ervoor gezorgd dat ik niet alleen op elk moment wist waar ik in het project stond, maar ook dat ik dit op eenvoudige en visuele manier naar de opdrachtgever kon communiceren. Daarnaast kon ik met het overzicht van uren en de bijbehorende grafieken sterk beargumenteren waarom en in hoeverre wijzigingen planningsgerelateerde gevolgen hebben. Bovendien kunnen het overzicht van uren en de grafieken me bij een volgend project helpen om tot een betere ureninschatting te komen.

### 13.3 Conclusie

Ik denk dat ik kan terugkijken op een geslaagde afstudeerperiode. Ik ben tevreden over product en proces en heb het het afgelopen halfjaar naar mijn zin gehad. De doelstelling is behaald en de opdrachtgever is tevreden. Het opgeleverde product voldoet aan de systeemeisen en sluit aan bij de verwachtingen van de doelgroep. De applicatie heeft nog enkele verbeter- en uitbreidingspunten maar binnen de gestelde tijd kan ik naar mijn mening spreken van een geslaagd product. Daarnaast denk ik dat ik vervolgontwikkelaars met de wijze van programmeren en het overdrachtsverslag voldoende handvatten heb geboden om verbeteringen en uitbreidingen door te voeren.

## FIGUREN INDEX

### *Hoofdstuk 3: Aanpak*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Doelgroepvariabelen	fig. 3.6	14
Faseverhouding	fig. 3.1	5
Methoden, JJG, IAD en DSDM	fig. 3.2	8
Persona's	fig. 3.7	15
Planningstechnieken	fig. 3.5	13
Prioritering van systeemeisen	fig. 3.8	16
Productgrootte inschattingstechnieken	fig. 3.3	11
Projectmanagement techniek	fig. 3.4	13
UML en andere technieken voor het beschrijven van een technisch ontwerp	fig. 3.9	17
Usability meettechnieken	fig. 3.10	22

### *Hoofdstuk 4: Projectoriëntatie en projectorganisatie opstellen*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Doelstelling	fig. 4.1	28
Gantt-chart van de fase 'Technisch ontwerp'	fig. 4.3	30
Urenverloop grafiek	fig. 4.4	30
Ureninschatting (gedeeltelijk)	fig. 4.2	29

### *Hoofdstuk 5: Voorbereiden op ontwerp*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Doelgroepomschrijving (gedeeltelijk)	fig. 5.2	32
Moodboard (gedeeltelijk)	fig. 5.4	34
Persona Miriam (gedeeltelijk)	fig. 5.6	36
Persona quote van Wilfried	fig. 5.5	35
Systeemeisen (gedeeltelijk) met behulp van de MoSCoW techniek	fig. 5.7	38
Userneeds (gedeeltelijk)	fig. 5.1	32
Vragenlijst die o.a. antwoord moet geven op de woon- en werksituatie van de doelgroep (gedeeltelijk)	fig. 5.3	33

### *Hoofdstuk 6: Technisch ontwerp opstellen*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Bestandstructuur	fig. 6.10	50
Database ontwerp	fig. 6.8	48
Kandidaatklassen (gedeeltelijk)	fig. 6.1	39
Klassediagram	fig. 6.3	41
Navigatiemodel voor gebruiker	fig. 6.9	49
Sequentiediagram 'Afspeellijst wijzigen'	fig. 6.7	46
Sequentiediagram 'Afspeellijsten ophalen'	fig. 6.6	45
Usecase 'Afspeellijst wijzigen'	fig. 6.4	42
Usecasediagram van Beheerder	fig. 6.5	44
Videospelerpagina's vertegenwoordiging in klassediagram	fig. 6.2	40

### *Hoofdstuk 7: Visueel ontwerp van iconen en schermen opstellen*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
CMS dat Wend hanteert (gedeeltelijk)	fig. 7.9	56
Iconen voor de handeling 'verwijderen'	fig. 7.13	60
Iconen voor de handelingen 'formaat instellen', 'videospeler beheren' en 'videospeler plaatsen'	fig. 7.14	60
Knoppen ontwikkeling	fig. 7.5	55
Menu aanpassing in stijl van nieuwe knoppen	fig. 7.6	56
Menu met ruimte	fig. 7.7	56
Menu zonder ruimte	fig. 7.8	56
Ontwerp van de pagina 'Afspeellijst wijzigen'	fig. 7.10	57
Ontwerp van de pagina 'Kleuren kiezen'	fig. 7.12	59
Schetsen van iconen en knoppen	fig. 7.3	54
Schetsen van schermonderdelen	fig. 7.4	55
Tabs ontwerp op de subpagina's van de pagina 'Videospeler beheren'	fig. 7.11	58
Verband tussen schermelementen met behulp van iconen	fig. 7.15	60
Wireframe van de pagina 'Afspeellijst toevoegen'	fig. 7.1	52
Wireframe van een subpagina van de pagina 'Videospeler beheren'	fig. 7.2	52

### *Hoofdstuk 8: Usability testen van iconen en schermen*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Aanbevelingen overzicht (gedeeltelijk)	fig. 8.9	70
Hoofdvraag van het onderzoek naar iconen en schermen	fig. 8.1	62
Iconen en betekenissen koppelen, stap 2 van het onderzoek naar iconen en schermen	fig. 8.3	65
Iconen herkennen, stap 1 van het onderzoek naar iconen en schermen	fig. 8.2	64
Testresultaten van de eerste stap 'Iconen herkennen', algemene iconen (gedeeltelijk)	fig. 8.5	67
Testresultaten van de eerste stap 'Iconen herkennen', applicatie specifieke iconen (gedeeltelijk)	fig. 8.6	68
Testresultaten van de tweede stap 'Iconen en betekenissen koppelen', algemene iconen (gedeeltelijk)	fig. 8.7	69
Testresultaten van de tweede stap 'Iconen en betekenissen koppelen', applicatie specifieke iconen (gedeeltelijk)	fig. 8.8	69
Testtaak 'Afspeellijst wijzigen', stap 3 van het onderzoek naar iconen en schermen	fig. 8.4	66
Vragen over schermen, 'Formaat instellen'	fig. 8.10	71
Vragen over schermen, 'Intro afbeelding kiezen'	fig. 8.11	72

### *Hoofdstuk 9: Visueel ontwerp uitwerken*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Steeksleutel, herziene schets van de handeling 'wijzigen'	fig. 9.1	74
'Verwijderen', 4 versies	fig. 9.2	74
Videospeler icoon aanpassing	fig. 9.4	75
Wijzigen icoon aanpassing	fig. 9.3	75



### *Hoofdstuk 10: Ontwikkelen van de applicatie*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Klasse 'Video' (gedeeltelijk), bevat de eigenschappen van een Video object	fig. 10.1	78
Klasse 'VideoFactory' (gedeeltelijk), voert bewerkingen van een Video object op een hoger niveau uit	fig. 10.2	78
Klassengebruik in HTML pagina's, in dit geval de pagina 'Statistieken'; de gemarkeerde tekst geeft weer hoe de klassen worden gebruikt	fig. 10.3	80
Kleuren aan onderdelen koppelen; de ontwikkelaar kan hier de namen van onderdelen invullen en zo aan een kleur koppelen	fig. 10.4	81
Kleur van de aangegeven onderdelen worden later in de tijdlijn automatisch veranderd	fig. 10.5	81

### *Hoofdstuk 11: Usability testen van de gehele applicatie*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Hoofdvraag van het onderzoek naar de usability van de applicatie	fig. 11.1	82
Testinhoud; er is gekozen voor animatiefilms dat voor herkenning bij de testpersonen zorgt	fig. 11.4	85
Testtaken; links in stappen, rechts alleen het gewenste eindresultaat	fig. 11.2	83
Vragenlijst (gedeeltelijk; SUMI / WAMMI) waarbij de variabele tevredenheid wordt geoperationaliseerd	fig. 11.3	84

### *Hoofdstuk 12: Herzien van de applicatie*

<b>figuur</b>	<b>figuurnummer</b>	<b>pagina</b>
Introductie teksten ('tips') toegevoegd en gemaximaliseerd; rechtsboven heeft de gebruiker de mogelijkheid om de tips te minimaliseren	fig. 12.2	88
Introductie teksten ('tips') toegevoegd en geminimaliseerd; rechtsboven heeft de gebruiker de mogelijkheid om de tips te maximaliseren	fig. 12.3	88
Introductie teksten nog toe te voegen op de pagina 'Video's beheren'	fig. 12.1	87
Stijlkeuze na het plaatsen van labels en het gebruiken van de originele kleuren	fig. 12.5	89
Stijlkeuze vóór opvolging van de aanbeveling; geen labels en de gebruikte kleuren komen uit de videospelerconfiguratie	fig. 12.4	89

## LITERATUURLIJST

### *Boeken en readers*

Creating Quality Personas, Shannon Ford, 2005  
Een Goed Advies, Godelieve Kodde en Walleke ten Noever Bakker, eerste druk, 2006  
Het Marketingplan, Marian Burk Wood, 2003  
Methoden en Technieken, D.B. Baarda, M.P.M. de Goede, 1999  
Perfecting Your Personas, Kim Goodwin  
Persona Non Grata, Dan Saffer, Adaptive Path, 2005  
Planning en beheersing, Selma Boenders, februari 2006  
Praktisch UML, Jos Warmer en Anneke Kleppe, 3<sup>de</sup> editie, 2004  
The Elements of User Experience, Jesse James Garret, 2003  
The Icon Book, William Horton, 1994

### *Internet*

<http://hcvl.hci.iastate.edu:4489/articles/category/the-user-experience>  
<http://nl.wikipedia.org/wiki/DSDM>  
<http://nl.wikipedia.org/wiki/IAD>  
<http://www.actionscript.org>  
[http://www.itbroadcastanddigitalcinema.com/ffmpeg\\_howto.html](http://www.itbroadcastanddigitalcinema.com/ffmpeg_howto.html)  
<http://www.jjg.net/elements/resources/>  
<http://www.kirupa.com>  
<http://www.naarvoren.nl/artikel/personas/>  
<http://www.php.net>  
[http://www.steptwo.com.au/papers/kmc\\_personas](http://www.steptwo.com.au/papers/kmc_personas)  
<http://www.usabilityweb.nl>