

# Plan van Aanpak

CMS-lite

Arjan Flipphi  
Maart 2007

# Inhoudsopgave

---

<b>Hoofdstuk 1 De inleiding .....</b>	<b>3</b>
1.1 Opdracht.....	3
1.2 CMS .....	3
1.3 Over dit document .....	3
<b>Hoofdstuk 2 De opdracht.....</b>	<b>4</b>
2.1 Huidige situatie.....	4
2.2 Probleemstelling.....	4
2.3 Opdrachtoomschrijving .....	4
2.4 Doel .....	5
<b>Hoofdstuk 3 Het proces .....</b>	<b>6</b>
3.1 Plan van Aanpak .....	6
3.2 Functioneel Ontwerp .....	6
3.3 Technisch Ontwerp.....	6
3.4 CMS-lite .....	6
3.5 Maken Website Generator Module .....	7
3.6 Testen.....	7
3.7 Applicatie aanpassen .....	7
3.8 Gebruikershandleiding.....	7
<b>Hoofdstuk 4 De planning.....</b>	<b>8</b>

# Hoofdstuk 1 De inleiding

---

## **Inleiding**

In dit document zal beschreven worden hoe dit project, die loopt in de periode februari 2007 - juni 2007, aangepakt gaat worden.

Dit plan van aanpak dient als richtlijn en als planning voor dit project

In dit document zullen zowel de opdracht als het proces worden vastgelegd.

## **1.1 Opdracht**

Het project zal ontwikkeld worden met behulp van een Functioneel en Technisch Ontwerp (Zie Hoofdstuk 3 paragraaf 3.2 en 3.3 voor meer informatie).

Het project zal een iteratief karakter hebben.

Voor dit project zal dat inhouden dat de fases Functioneel en Technisch Ontwerp mogelijke meerdere malen herschreven zal worden .

Pas nadat deze fases (Zie Hoofdstuk 3 voor meer informatie over de fases) door de opdrachtgever goedgekeurd zijn, zal er met ontwikkeling van de applicatie worden begonnen.

Alle fases bij elkaar zullen uiteindelijk een werkende applicatie, die zal voldoen aan de wensen en eisen van de opdrachtgever, opleveren met daarbij de benodigde documentatie.

## **1.2 CMS**

Een CMS is een applicatie waarmee eigenaren van een de website de content van hun website kunnen beheren.

De term CMS staat voor Content Management Systeem.

Een CMS wordt vooral gebruikt bij websites waarvan de inhoud regelmatig aangepast wordt.

## **1.3 Over dit document**

Dit document is, naast deze inleiding, opgedeeld in nog drie hoofdstukken.

Als eerste hoofdstuk zal de opdracht worden besproken.

In dit hoofdstuk zal er worden gesproken over de huidige situatie, de probleemstelling, de opdracht die uit deze probleemstelling voort zal komen en welke doelen deze opdracht moet gaan verwezenlijken.

Daarna zal het proces voor het maken de opdracht worden besproken.

Hierin zal kort de fases worden besproken die tijdens dit project doorlopen zullen worden met daarbij aangegeven welke mijlpalen deze fases zullen voortbrengen.

Vervolgens zal er in hoofdstuk 4 de planning voor het gehele traject worden vastgelegd.

Hierin zal worden aangegeven welk fase wanneer van start gaat en wanneer deze dan ook weer afgerond moet zijn.

## Hoofdstuk 2 De opdracht

---

### Opdracht

In dit hoofdstuk zal worden beschreven wat de opdracht is tijdens dit project. Het doel van dit hoofdstuk is de lezer inzicht te geven in waarom er voor deze opdracht gekozen en welke doelen dit project moet gaan verwezenlijken.

### 2.1 Huidige situatie

Bij Hebbizz maken ze sinds enkele jaren gebruik van een eigen geschreven Content Management Systeem (CMS).

Dit systeem wordt, voor elke nieuwe klant die een website met CMS wil, aangepast naar zijn of haar wensen.

### 2.2 Probleemstelling

Doordat dit systeem al een aantal jaren gebruikt wordt en er elke keer aanpassingen op gedaan zonder dat deze aanpassingen goed gedocumenteerd zijn is dit systeem “vervuild” geraakt. Er is daarom ook eigenlijk geen sprake meer van één CMS-versie, maar van een hele serie verschillende versies. Onderhoud, inzetbaarheid en aanpasbaarheid nemen steeds hierdoor meer tijd in beslag.

Doordat dit proces een steeds langere tijd in beslag neemt zijn de kosten voor de klanten aan de hoge kant.

Daarnaast kost het ontwikkelen van een website veel tijd.

### 2.3 Opdrachtomschrijving

Het project zal dan ook zijn, het bouwen van een CMS dat voldoet aan de wensen en eisen van de opdrachtgever

De opdrachtgever heeft de volgende wensen en eisen voor dit project.

Deze wensen en eisen kunnen we onderverdelen in functionele eisen en technische eisen.

Functionele wensen en eisen:

- Gebruiksvriendelijk
- Overzichtelijk
- Beheersmogelijkheid ( voor het beheren van de klanten van het CMS-lite )
- Geen submenu items
- Geen volgorde bepaling van menu-items
- Verschillende mogelijkheden voor de lay-out van de website

Deze mogelijkheden zijn weer onder te verdelen in een aantal wensen en eisen namelijk: (deze zullen in de toekomst eventueel uitgebreid gaan worden)

- o Keuze uit 2 lay-out mogelijkheden
- o Eigen logo kunnen toevoegen
- o Lettertype kunnen kiezen
- o Lettergrootte kunnen kiezen
- o Letterkleur kunnen kiezen
- o De opmaak van een link op de website

Technische wensen en eisen:

- Maximaal 5 pagina's
- Snel inzetbaar
- 1 CMS voor alle klanten

Naast het bouwen van een CMS zal er ook een module "Website Generator" worden gemaakt.

Deze is bedoeld om de bouwen van een website sneller en eenvoudiger te maken.

## **2.4 Doel**

Uit de probleemstelling van paragraaf 2.1 kunnen we de volgende doelen voor dit project opstellen

- Het bouwen van een goed gedocumenteerde, tijdsbesparende en kostenbesparend CMS
- Het bouwen van een aparte module voor het eenvoudig kunnen maken van een website, welke de klant naar eigen wil kan aanpassen.

Uiteindelijk moet men in staat zijn dmv dit CMS-lite binnen korte tijd een (standaard) website op te zetten die makkelijk aan te passen is en te onderhouden is.

De voordelen voor de klant van dit product zijn kosten besparing, overzicht, snel een website kunnen maken.

De voordelen voor Hebbizz zijn uniformiteit, makkelijk te onderhouden omdat het product centraal staat, product dat te verkopen is etc.

## Hoofdstuk 3 Het proces

---

### Proces

Voor een goede structuur binnen dit project zal het project worden opgedeeld in een aantal fases.

Deze fases zullen in chronologische volgorde gevolgd gaan worden.

Naast de verschillende fases binnen dit project die hieronder kort beschreven worden zullen er ook een aantal mijlpalen moeten worden opgeleverd. Al deze mijlpalen zullen onderdeel worden van de opdracht.

In hoofdstuk 4 zal de planning van deze fases besproken worden.

### **3.1 Plan van Aanpak**

Het doel van deze fase is het schrijven van het document dat u nu leest: het Plan van Aanpak. Zie hoofdstuk 1 voor een beschrijving van dit document.

Mijlpaal: opleveren van het Plan van Aanpak

### **3.2 Functioneel Ontwerp**

In deze fase zal het functioneel ontwerp voor het CMS-lite worden geschreven

In dit document worden de functionele aspecten van het CMS-lite uitgelegd.

Hierbij kan men denken aan een navigatieschema, de verschillende schermen ed.

Met andere woorden, de visuele kant van de applicatie.

Mijlpaal: opleveren van het Functioneel ontwerp

### **3.3 Technisch Ontwerp**

In deze fase zal het technisch ontwerp voor het CMS-lite worden geschreven

In dit document worden de technische aspecten van het CMS-lite uitgelegd.

Hierbij kan men denken aan een databaseschema, de programmeerstructuur ed.

Met andere woorden, de technische kant van de applicatie.

Voor de tot stand koming van de technische aspecten binnen dit document zullen er een aantal technieken worden toegepast, bijvoorbeeld XUAN-model

Mijlpaal: opleveren van het Technisch ontwerp

### **3.4 CMS-lite**

In deze fase zal de module CMS-lite worden ontwikkeld.

De ontwikkeling van deze applicatie zal gebeuren aan de hand van de het functioneel en technisch ontwerp.

Zoals al eerder gezegd bevatten deze documenten alle aspecten om de applicatie te kunnen maken.

Mijlpaal: opleveren van de applicatie CMS-lite

### ***3.5 Maken Website Generator Module***

In deze fase zal de module Website Generator gemaakt gaan worden.  
Deze module moet er voor zorgen dat klanten op een snelle en eenvoudige manier hun eigen website kunnen indeling en beheren.

Mijlpaal: opleveren van Module Website Generator voor de testfase

### ***3.6 Testen***

In deze fase zullen de applicatie CMS-lite en de module Website Generator worden getest aan de hand van de in fase 3.2 Functioneel ontwerp en 3.3 Technisch ontwerp aangegeven testmethodes

De resultaten uit deze test zullen in de fase 3.7 Applicatie aanpassen worden aangepast.  
Mocht het nog zijn zal deze fase tezamen met de fase 3.7 Applicatie aanpassen net zo lang worden herhaald totdat alle wensen en eisen, die in fase 3.1 Plan van Aanpak, zijn vastgelegd behaald zijn.

Mijlpaal: opleveren van de Testresultaten

### ***3.7 Applicatie aanpassen***

In deze fase zullen alle aanpassingen, die uit de testfase naar voren zijn gekomen, worden uitgevoerd

Mijlpaal: opleveren van de aanpaste applicatie(s)

### ***3.8 Gebruikershandleiding***

In deze fase zal de gebruikershandleiding worden geschreven.  
In deze handleiding zullen o.a. de basisfunctie van de applicatie worden beschreven

Mijlpaal: opleveren van de Gebruikershandleiding

## Hoofdstuk 4 De planning

---

### Planning

In dit hoofdstuk zal er een globale planning geven worden aan de opdracht, gebaseerd op de opdeling in fases zoals die in hoofdstuk 3 is besproken.

Week 6 - Week 7: Opstellen van het Plan van Aanpak

Week 7 t/m Week 9: Opstellen Functioneel Ontwerp CMS-lite

Week 9 t/m Week 11: Opstellen Technisch Ontwerp CMS-lite

Week 11 t/m Week 19: CMS-lite applicatie maken

Week 17 t/m Week 20: Website generator module maken

Week 20 - Week 21: Testen van de gemaakte applicaties

Week 21 - Week 22: Aanpassen van de applicatie (indien nodig)

Week 22 - Week 23: Gebruikershandleiding schrijven

(Weken zijn kalenderweken)

## INHOUDSOPGAVE

---

1	INLEIDING .....	2
2	NAVIGATIESHEMA .....	3
3	BESTURINGSMODEL.....	5
	3.1 DIALOGSTRUCTUUR .....	5
	3.2 SCHERMLAY-OUTS .....	6
4	SYSTEEMTESTPLAN.....	13

## 1. Inleiding

---

Dit document is geschreven in het kader van het afstudeerproject “Ontwikkelen van een modulair CMS”.

Dit afstudeerproject is een onderdeel van de opleiding Vormgeving en ontwerp van InterActie (VIA) aan de sector Informatica & Informatiekunde aan de Haagse Hogeschool.

Dit document is een functioneel ontwerp.

In dit document zullen de volgende onderdelen per hoofdstuk beschreven worden.

Hoofdstuk 2 Navigatieschema

In dit hoofdstuk zal door middel van een navigatieschema beschreven worden welke mogelijkheden er zijn binnen het CMS.

Hoofdstuk 3 Besturingsmodel

In dit hoofdstuk zullen de eerste schermafbeeldingen van de CMS applicatie worden beschreven.

Hoofdstuk 4 Systeemtestplan

In dit hoofdstuk zal beschreven worden met welke teststrategie deze applicatie getest zal worden.

Arjan Flipphi  
Hebbizz B.V.  
Amsterdam, 26 Maart 2007

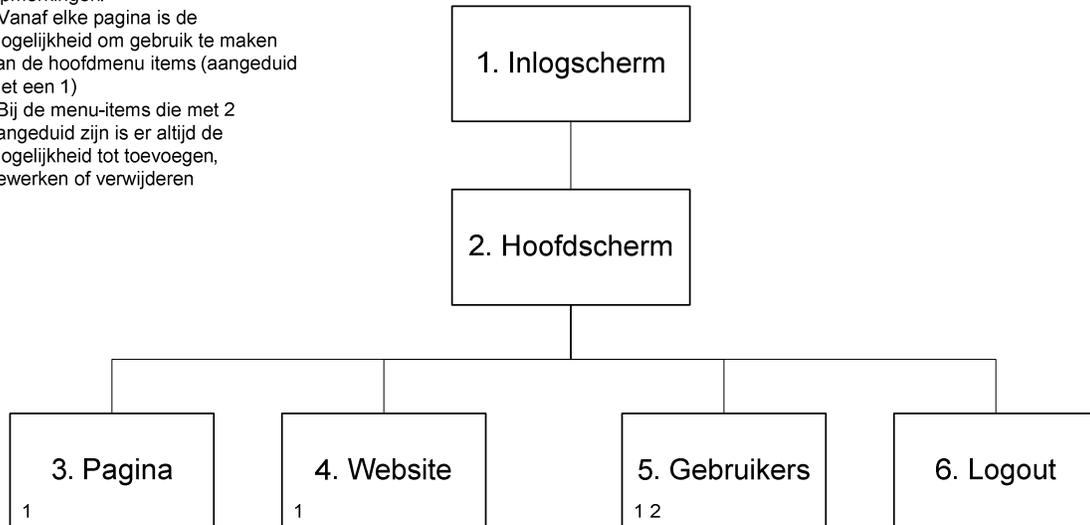
## 2. Navigatieschema

---

### Klant

Opmerkingen:

- Vanaf elke pagina is de mogelijkheid om gebruik te maken van de hoofdmenu items (aangeduid met een 1)
- Bij de menu-items die met 2 aangeduid zijn is er altijd de mogelijkheid tot toevoegen, bewerken of verwijderen

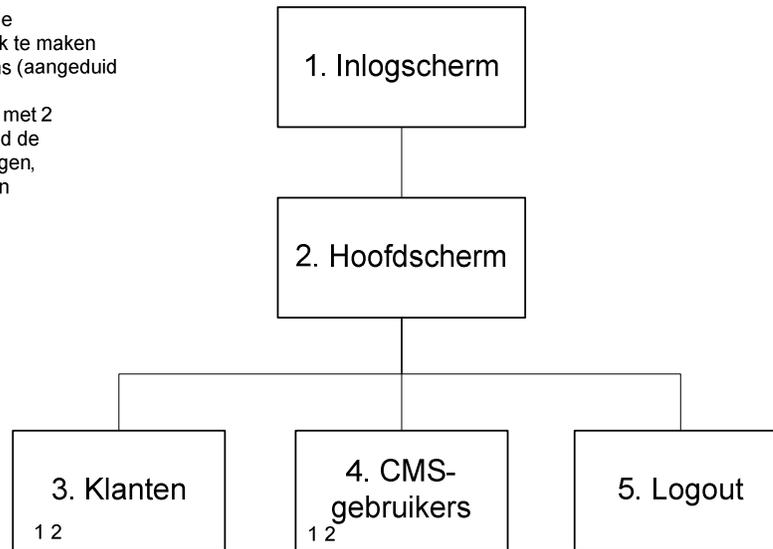


Figuur 1 Navigatieschema Klanten

## Hebbizz administrator

Opmerkingen:

- Vanaf elke pagina is de mogelijkheid om gebruik te maken van de hoofdmenu items (aangeduid met een 1)
- Bij de menu-items die met 2 aangeduid zijn is er altijd de mogelijkheid tot toevoegen, bewerken of verwijderen

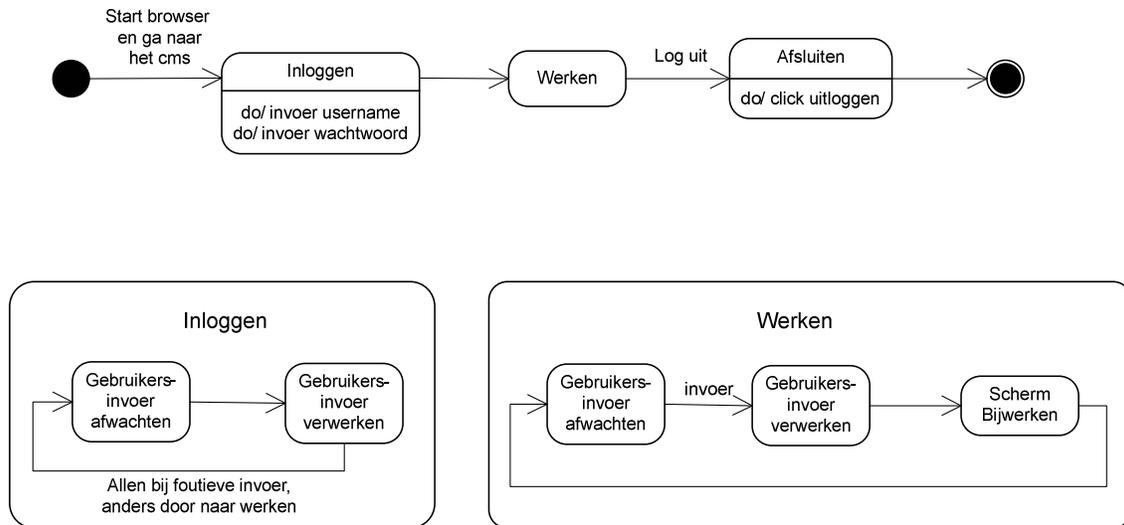


*Figuur 2 Navigatieschema Hebbizz administrator*

### 3. Besturingsmodel

In het besturingsmodel wordt de dialoogstructuur weergegeven met behulp van een toestandsdiagram. Tevens worden de schermlay-outs en de formulierlay-outs beschreven.

#### 3.1 Dialoogstructuur

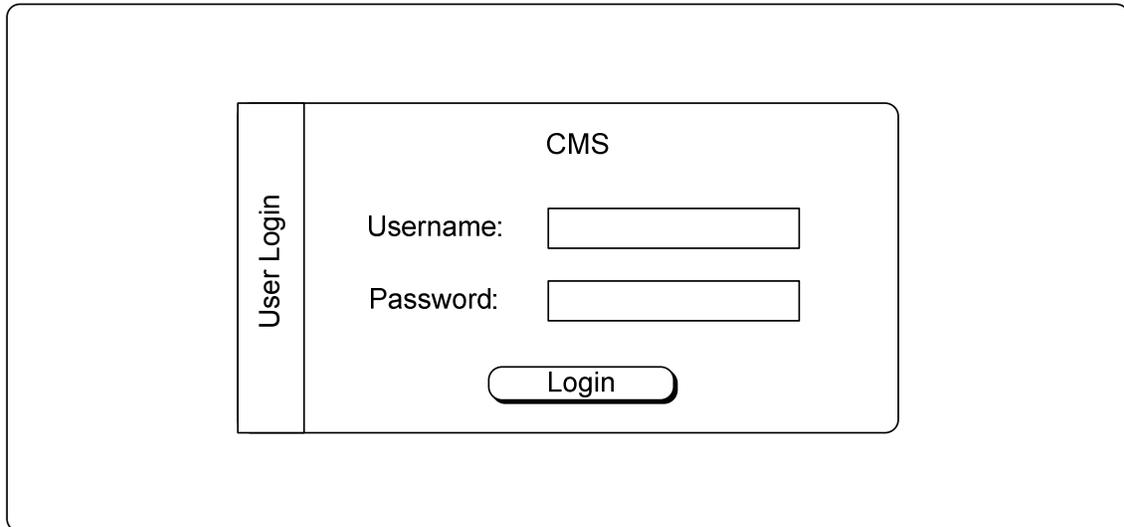


Figuur 3 Toestandsdiagram CMS Lite

### 3.2 Schermlay-outs

In deze paragraaf worden per functie de schermlay-outs beschreven.  
Per scherm worden zullen de volgende zaken worden vastgelegd:

Scherf "Inloggen"



The image shows a wireframe of a login form. It is contained within a large rounded rectangle. Inside this rectangle is a smaller rounded rectangle representing the form itself. On the left side of this inner rectangle is a vertical label "User Login". To the right of this label, the text "CMS" is centered at the top. Below "CMS" are two input fields: "Username:" followed by a rectangular box, and "Password:" followed by another rectangular box. Below these two fields is a rounded rectangular button labeled "Login".

*Figuur 4 Inlogscherf*

#### Beschrijving

Dit is het inlogscherf voor het CMS.

De gebruiker voert zijn of haar username en wachtwoord in.

Er zal na invoer een check worden uitgevoerd of deze user inderdaad bestaat en rechten heeft tot het CMS.

Als deze check eenmaal gedaan is, zal aan de hand van de invoerde gegevens in de database gekeken worden welke instelling bij de gebruiker horen.

## Scherma "Home"



*Figuur 5 Scherm "na inloggen"*

### Beschrijving

Dit scherm krijgt de gebruiker te zien zodra deze ingelogd.

In dit scherm krijgt de gebruiker een aantal vaste opties te weten:

- Pagina
- Website
- Gebruikers
- Uitloggen

## Scherf "Pagina"

Pagina Website Gebruikers Uitloggen

Ingelogd als .....

Nieuw Delete

Hier komen de aanwezige pagina's te staan

<input type="checkbox"/> Naam	Actief
<input type="checkbox"/> Home	 publish
<input type="checkbox"/> About Us	 unpublish
<input type="checkbox"/> Contact	 unpublish

Figuur 6 Scherm Pagina

### Beschrijving

In dit scherm worden alle aanwezige pagina getoond.

De tekst "nieuw" is een link naar "pagina aanpassen", alleen worden er dan geen gegevens ingevuld.

Het vakje voor de teksten die in het blauw gedrukt staan is een checkbox.

Door één van deze boxen aan te vinken en dan vervolgens op de tekst "delete" te drukken zal deze pagina verwijderd worden.

Klikt men op het vakje voor de tekst "Naam" dan zullen alle checkboxes voor de teksten in het blauw geselecteerd worden en kunnen ze verwijderd worden.

De teksten die in het blauw staan zijn links naar "pagina aanpassen".

Door op de tekst "publish" of "unpublish" te klikken zal de pagina al dan niet actief zijn op de website

## Scherma "Pagina aanpassen"

Pagina Website Gebruikers Uitloggen

Pagina -> Pagina wijzigen Ingelogd als .....

Menunaam:

Koptekst:

Tekst: 

Hier komt de tekst van de pagina

Actief:  Ja  Nee

Figuur 7 Scherm Pagina aanpassen

### Beschrijving

In dit scherm kan de pagina inclusief de naam van het menu-item worden aangepast. Bij het veld tekst zal een FCKEditor<sup>1</sup> worden getoond. In deze editor kan men de tekst die op website staat aanpassen.

<sup>1</sup> FCKEditor : Is een editor waarmee men opmaak mee kan geven aan de tekst

## Scherma "Website-setup"

Figuur 8 Schermafbeelding Website Setup

In dit scherm heeft de klant de mogelijkheid om zijn of haar website vorm te geven.

Bij "Kies lay-out" zullen een, nog nader te bepalen, aantal website lay-outs getoond worden d.m.v. een zogehete dropdown menu.

Bij "Kies uw logo" kan de klant zijn of haar logo uploaden naar de server en deze wordt dan vervolgens in de website verwerkt.

Bij "Kies uw lettertype" heeft de klant een keuze uit een aantal lettertype namelijk:

- Arial
- Times New Roman
- Georgia
- Verdana
- Courier
- Geneva

Dit zijn namelijk lettertype die je kan gebruiken op een website zonder dat dit problemen oplevert.

Bij "Kies Link" heeft de klant de mogelijkheid om de kleur te kiezen die alle website links hebben en de mogelijkheid om de kleur aan te geven die de link heeft zodra een gebruiker met zijn of haar muis over de link gaat.

Indien er al een website setup is gemaakt door de klant zal in dit scherm de gegevens worden getoond.

Anders zijn alle invoerschermen leeg.

## Scherf "Gebruikers"

Site   Website-setup   Gebruikers   Uitloggen

Gebruikers   Ingelogd als .....

Nieuw   Delete

Hier komen de aanwezige gebruikers te staan

<input type="checkbox"/>	Naam	Email	Actief
<input type="checkbox"/>	Jan	jan@cms.nl	✘
<input type="checkbox"/>	Piet	piet@cms.nl	✔
<input type="checkbox"/>	Klaas	klaas@cms.nl	✔

Figuur 9 Scherm Gebruiker

### Beschrijving

In dit scherm worden alle aanwezige gebruikers getoond.

De tekst "nieuw" is een link naar "gebruikers aanpassen", alleen worden er dan geen gegevens ingevuld.

Het vakje voor de teksten die in het blauw gedrukt staan is een checkbox.

Door één van deze boxen aan te vinken en dan vervolgens op de tekst "delete" te drukken zal deze pagina verwijderd worden.

Klikt men op het vakje voor de tekst "Naam" dan zullen alle checkboxen voor de teksten in het blauw geselecteerd worden en kunnen ze verwijderd worden.

Onder het kopje "Actief" wordt aangegeven of de gebruiker actief is en rechten heeft om het cms te gebruiken.

Een vinkje geeft aan dat de gebruiker actief is en dat deze gebruiker recht heeft om het cms te gebruiken.

Met een kruisje is de gebruiker niet actief en heeft geen toegang tot het cms.

De teksten die in het blauw staan zijn links naar "gebruikers aanpassen".

## Scherma "Gebruikers aanpassen"

The screenshot shows a web interface with a navigation bar at the top containing 'Site', 'Website-setup', 'Gebruikers', and 'Uitloggen'. Below the navigation bar is a breadcrumb trail: 'Gebruikers -> Gebruikers aanpassen'. The main content area is titled 'Nieuw' and contains a form with the following fields and options:

- Naam:
- Voornaam:
- Achternaam:
- Wachtwoord:
- Email:
- Actief:  Ja  Nee

At the bottom of the form is a button labeled 'Opslaan'.

Figuur 10 Scherm Gebruiker aanpassen

### Beschrijving

In dit scherm kunnen bestaande gebruikers worden aangepast of nieuwe gebruikers worden aangemaakt.

Dit is afhankelijk van welke keuze er bij het scherm paragraaf gemaakt is.

Keuze "Nieuw":

Alle velden in het scherm zullen blanco zijn.

Keuze "Huidige aanpassen"

Het veld "Naam" zal ingevuld zijn met de huidige naam.

Het veld "Voornaam" zal ingevuld zijn met de huidige voornaam van de gebruiker.

Het veld "Achternaam" zal ingevuld zijn met de huidige achternaam van de gebruiker.

Het veld "Wachtwoord" zal ingevuld zijn met het huidige wachtwoord van de gebruiker.

Het veld "Email" zal ingevuld zijn met het huidige e-mail adres van de gebruiker.

Het veld "Actief" zijn radiobuttons. Door hier een keuze te maken tussen "Ja" of "Nee"

Zal de gebruiker wel of geen toegang hebben tot het cms.

## 4. Systeemtestplan

---

Bij het testen van het hele systeem zal er gekeken worden naar een aantal zaken te weten:

- Functionaliteit (doet het systeem wat ik wil)
- Betrouwbaarheid (doet het systeem het wanneer ik het wil)
- Bruikbaarheid (kan ik goed met het systeem werken )

Het doel van het testen van dit systeem is niet alleen om te kijken of er nog fouten zitten in het systeem maar ook om te zien of het systeem aan de wensen en eisen voldoet.

Het CMS bestaat een aantal onderdelen, zoals te zien is in hoofdstuk 3.

Voor het testen van deze onderdelen zal er gebruik worden gemaakt van een testcase (zie figuur 11).

Deze testcase zal worden doorlopen door een x aantal personen.

Tijdens het doorlopen van de cases zal er door de observeerder gebruikt worden gemaakt de observatieformulieren (zie figuur 12 en 13).

De resultaten van de 2 formulieren zullen met elkaar worden vergeleken en de uitkomst hiervan zal worden verwerkt in het nieuwe CMS.

Dus mochten er plekken zijn waar het nieuwe CMS nog verder verbeterd zou kunnen worden dan zal dat uit deze test naar voren komen.

Scenario	Omschrijving	Evaluatie Hoe vond je het gaan?
0	Log in met de volgende gegevens: gebruikersnaam: klant wachtwoord: klant	
1	Maak een nieuwe pagina aan voor op de website met zelf in te vullen gegevens	
2	Wijzig van de zojuist aangemaakte pagina de naam van de knop in 'Testpagina'	
3	Maak een nieuwe gebruiker met de naam 'Jan' aan	

4	Wijzig het wachtwoord van gebruiker 'Jan' in 'jantje01' en controleer of het nieuwe wachtwoord werkt	
---	--	--

Figuur 11 Testcase voor huidige en nieuwe CMS

### Observatieformulier voor nieuwe CMS

	Scenario	Juiste windows acties	Aantal Kliks	Aantal fouten	Aantal stagnaties	Aantal afvragen
0	Log in met de volgende gegevens:  gebruikersnaam: klant  wachtwoord: klant	- Typ 'naam' in - Typ 'wachtwoord' in - druk op 'login'				
1	Maak een nieuwe pagina aan voor op de website met zelf in te vullen gegevens	- Klik op 'pagina' - Klik op 'nieuw' - Voer data in - klik op 'opslaan'				
2	Wijzig van de zojuist aangemaakte pagina de naam van de knop in 'Testpagina'	- klik op de zojuist aangemaakte pagina. - Verander de menu naam - Klik op opslaan				
3	Maak een nieuwe gebruiker met de naam 'Jan' aan	- Klik op 'Gebruikers' - Klik op 'nieuw' - Vul de gegevens in. - Klik op opslaan				
4	Wijzig het wachtwoord van gebruiker 'Jan' in 'jantje01' en controleer of het nieuwe wachtwoord werkt	- Klik op gebruiker 'Jan' - Vul het nieuwe wachtwoord in - Klik op opslaan - Klik op logout - Vul gebruikersnaam in - Vul nieuwe wachtwoord in - Klik op inloggen				

Figuur 82 Observatieformulier nieuwe cms

## Observatieformulier voor huidige CMS

	Scenario	Juiste windows acties	Aantal Kliks	Aantal fouten	Aantal stagnaties	Aantal afvragen
0	Log in met de volgende gegevens:  gebruikersnaam: klant  wachtwoord: klant	- Typ 'inlognaam' in - Typ 'wachtwoord' in - druk op 'login'				
1	Maak een nieuwe pagina aan voor op de website met zelf in te vullen gegevens	- Klik op 'site -> paragraaf' - Klik op 'nieuw' - Voer data in - klik op 'opslaan' - Klik op publish bij de nieuwe paragraaf  - Klik op 'site -> pagina' - Klik op 'nieuw' - Voer data in en koppel de nieuwe paragraaf aan de pagina - klik op 'opslaan'  - Klik op 'site -> menu-items' - Klik op 'nieuw' - Voer data in - klik op 'opslaan'				
2	Wijzig van de zojuist aangemaakte pagina de naam van de knop in 'Testpagina'	- klik op 'site -> menu-item.' - klik op de zojuist aangemaakte menu-item - Verander de menu naam - Klik op opslaan				
3	Maak een nieuwe gebruiker met de naam 'Jan' aan	- Klik op 'Admin -> CMS-gebruikers -> nieuwsgebruikers' - Vul de gegevens in. - Klik op opslaan				
4	Wijzig het wachtwoord van gebruiker 'Jan' in 'jantje01' en controleer of het nieuwe wachtwoord werkt	- Klik op Admin -> CMS-gebruikers -> aanpassen gebruikers' - Klik op 'wachtwoord' achter jan - Vul het nieuwe wachtwoord in - Klik op opslaan - Klik op logout - Vul gebruikersnaam in - Vul nieuwe wachtwoord in - Klik op inloggen				

Figuur 13 Observatieformulier nieuwe cms

Voor het testen van de technische aspecten (werking van de verschillende functies ed.) van het “nieuwe“ cms zal er gebruik gemaakt gaan worden van de onderstaande formulieren.

De taken waarvan de uitkomst “nee” is zullen na het testen worden aangepast

Zoals al eerder beschreven bestaat dit “nieuwe CMS” uit twee delen namelijk een klanten kant en een Hebbizz kant.

Deze twee kanten zullen door middel van verschillende formulieren worden getest.

Als eerste zullen de formulieren van de klantenkant worden weergegeven (figuur 14 t/m 17).

Daarna de testformulieren van de Hebbizz kant (figuur 18 en figuur 19)

“Inloggen”

Naam van de module	Uitkomst	Opmerkingen
Inloggen		
Check inloggen met onbekende gebruiker - wordt er een melding getoond dat de gebruiker onbekend is?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
Check inloggen met onjuist wachtwoord - wordt er een melding getoond dat het wachtwoord onjuist is?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
Check inloggen met juist wachtwoord - wordt de gebruiker doorgestuurd naar het hoofdscherm?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	

Figuur17 Testformulier inloggen

Module “Pagina”

Naam van de module	Uitkomst	Opmerkingen
Pagina toevoegen		
- wordt er een melding getoond dat de pagina is opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- zijn alle ingevulde gegevens goed opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt er een melding getoond als er meer dan 5 pagina's aanwezig zijn	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
Pagina wijzigen		
- worden de huidige gegevens in de juiste velden getoond?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt er een melding getoond dat de pagina is gewijzigd?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	

Pagina verwijderen		
- wordt er een waarschuwing getoond zodra er op verwijderen is geklikt?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt de juiste pagina verwijderd?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	

*Figuur 15 Testformulier pagina*

Module “Website-setup”

Naam van de module	Uitkomst	Opmerkingen
Website lay-out toevoegen		
- wordt er een melding getoond dat de lay-out is opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- zijn alle ingevulde gegevens juist opgeslagen? (Dit is te controleren op de website.)	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
Website lay-out wijzigen		
- wordt er een melding getoond dat de gewijzigde lay-out is opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- zijn alle ingevulde gegevens juist opgeslagen? (Dit is te controleren op de website.)	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	

*Figuur 16 Testformulier website setup*

Module “Gebruikers”

Naam van de module	Uitkomst	Opmerkingen
Gebruiker toevoegen		
- wordt er een melding getoond dat de gebruiker is opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- zijn alle ingevulde gegevens juist opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
Gebruiker wijzigen		
- worden de huidige gegevens in de juiste velden getoond?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt er een melding getoond dat de gegevens zijn gewijzigd?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
Gebruiker verwijderen		
- wordt er een waarschuwing getoond zodra er op verwijderen is geklikt?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt de juiste gebruiker verwijderd?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	

Figuur 17 Testformulier gebruikers

Voor de Hebbizz kant van het CMS zullen de volgende formulieren gebruikt worden.

Module “CMS-gebruikers”

Naam van de module	Uitkomst	Opmerkingen
CMS-gebruiker toevoegen		
- wordt er een melding getoond dat de gebruiker is opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- zijn alle ingevulde gegevens juist opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
CMS-gebruiker wijzigen		
- worden de huidige gegevens in de juiste velden getoond?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt er een melding getoond dat de gegevens zijn gewijzigd?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
CMS-gebruiker verwijderen		
- wordt er een waarschuwing getoond zodra er op verwijderen is geklikt?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt de juiste gebruiker verwijderd?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	

Figuur 18 Testformulier CMS-gebruikers

Module “Klant setup”

Naam van de module	Uitkomst	Opmerkingen
Klant toevoegen		
- wordt er een melding getoond dat de gebruiker is opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- zijn alle ingevulde gegevens juist opgeslagen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
Klant wijzigen		
- worden de huidige gegevens in de juiste velden getoond?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt er een melding getoond dat de gegevens zijn gewijzigd?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
Klant verwijderen		
- wordt er een waarschuwing getoond zodra er op verwijderen is geklikt?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	
- wordt de juiste klant verwijderd?	<input type="checkbox"/> Ja <input type="checkbox"/> Nee	

Figuur 19 Testformulier klant setup

## Inhoudsopgave

---

<b>1</b>	<b>Inleiding .....</b>	<b>2</b>
<b>2</b>	<b>De database .....</b>	<b>3</b>
	2.1 Gegevensmodellen .....	3
	2.2 Database schema .....	5
	2.3 Database beperkingregels .....	6
	2.4 XUAN-model .....	7
<b>3</b>	<b>Programmeertaal .....</b>	<b>10</b>
<b>4</b>	<b>Technische Infrastructuur .....</b>	<b>14</b>
	4.1 Benodigde Hard - en Software .....	14
	4.2 De omgeving.....	15

# 1. Inleiding

---

Dit document is geschreven in het kader van het afstudeerproject “Ontwikkelen van een modulair CMS”.

Dit afstudeerproject is een onderdeel van de opleiding Vormgeving en ontwerp van InterActie (VIA) aan de sector Informatica & Informatiekunde aan de Haagse Hoge School.

Dit document is het Technisch ontwerp.

In dit document zullen de volgende onderdelen besproken worden.

Hoofdstuk 2

In dit hoofdstuk zal de database opzet worden besproken worden van de applicatie.

Hierbij zal er onderscheidt worden gemaakt tussen de database van de klanten en die van Hebbizz.

Hoofdstuk 3.

In dit hoofdstuk de programmeertaal worden besproken die er tijdens dit project gebruikt zal worden

Hoofdstuk 4

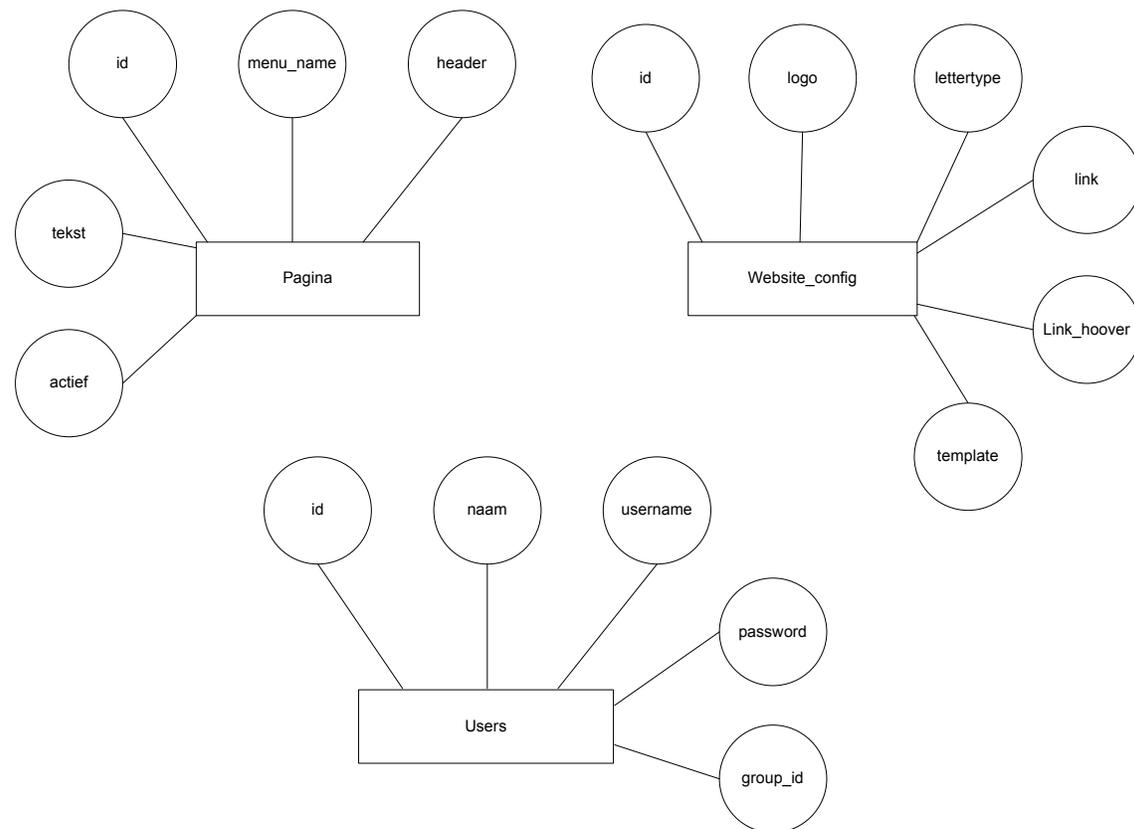
In dit hoofdstuk zal besproken worden waar de applicatie zal worden ontwikkeld en waar de applicatie gebruikt zal worden door de klant

## 2. De database

In dit hoofdstuk zal de tot stand koming van de database worden uitgelegd. Door in deze modellen aan te geven of er relaties zijn tussen de verschillende tabellen zal het later in dit hoofdstuk makkelijker zijn om de definitieve database schema te maken. Verder zal in dit hoofdstuk aangegeven worden welke beperkingen er op database zullen zijn.

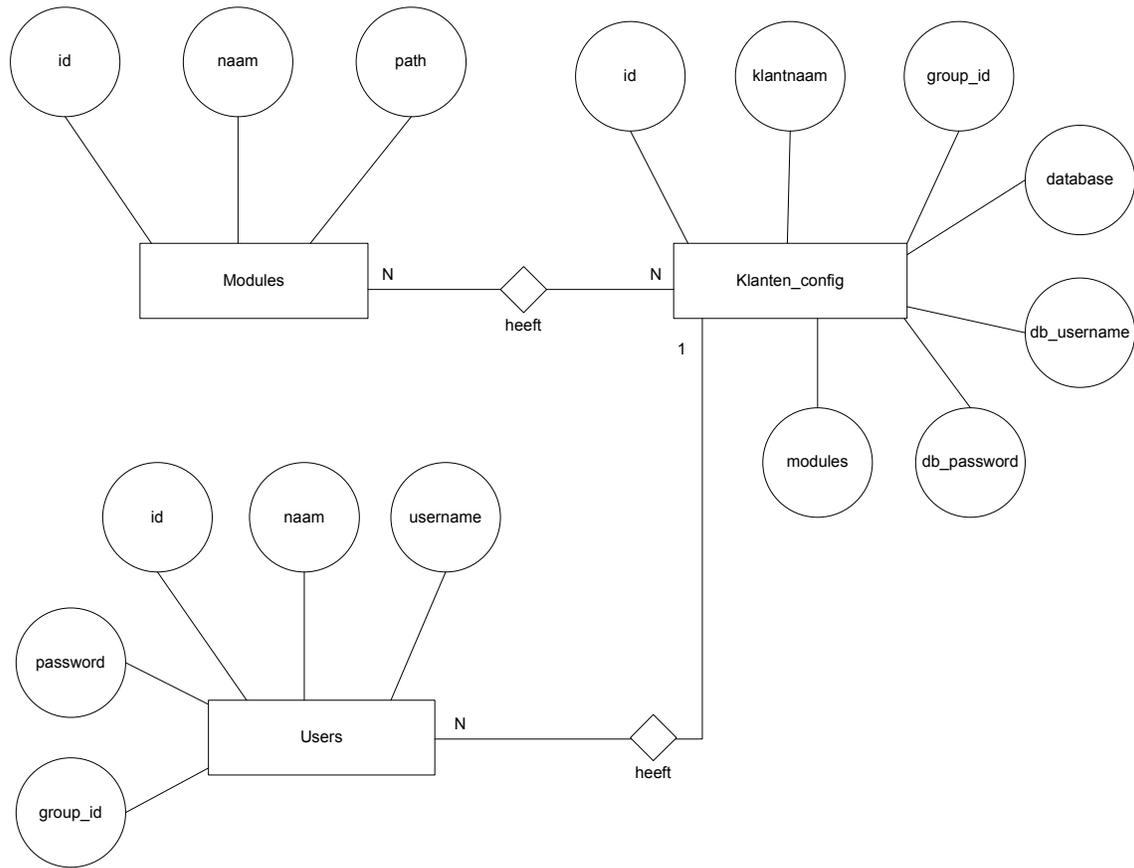
### 2.1 Gegevensmodellen

In het onderstaande schema is te zien hoe de opmaak zal zijn van de database van de klant.



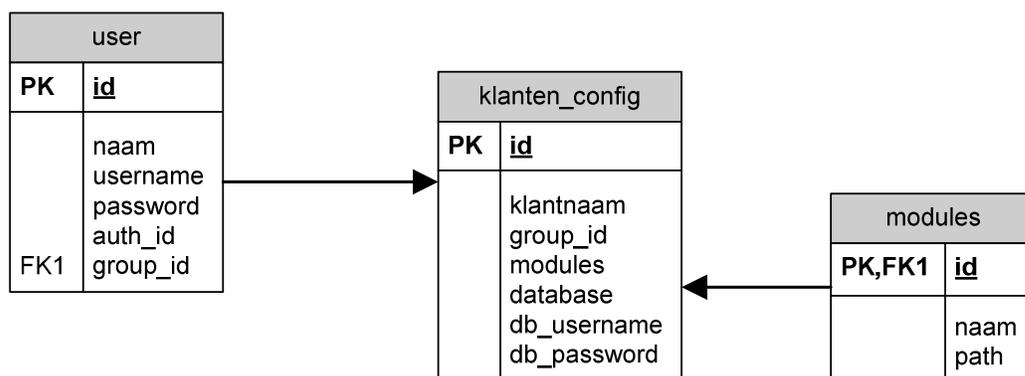
Figuur 1 Gegevensmodel klant

In het onderstaande schema is te zien hoe de opmaak zal zijn van de database voor Hebbizz.



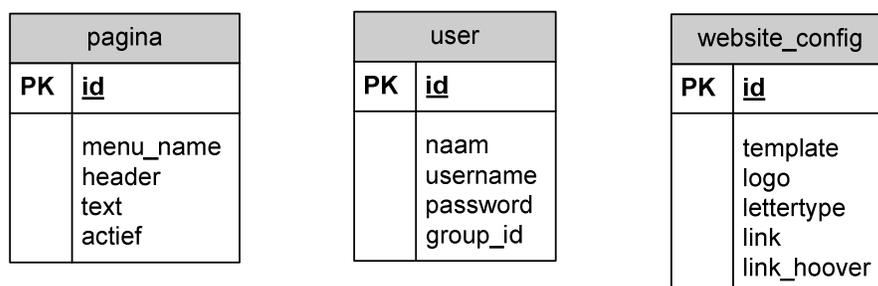
Figuur 2 Gegevensmodel Hebbizz

## 2.2 Database schema



Figuur 3 Database schema voor Hebbizz

In deze database worden alle klanten geregistreerd die gebruik mogen maken van het CMS. Op basis van de inlognaam wordt er een connectie gemaakt met dat database van de klant. (zie figuur 2).



Figuur 4 Database voor de klant

Zoals te zien is zijn alle tabellen op zichzelf staande tabellen.

De tabel "user" gebruikt worden door de klant om gebruiker toegang te geven tot "hun" CMS.

De gebruikers die door de klant worden aangemaakt zullen tevens worden weggeschreven in de database van Hebbizz.

Dit om te weten hoeveel gebruikers er per klant zijn die toegang hebben tot het CMS.

De tabel "website\_setup" zal gebruikt worden door de module die standaard in het CMS zal zitten.

Binnen deze module is het mogelijk voor de klant zelf om makkelijk een website op te zetten en eventuele wijziging aan te brengen aan kleuren ed.

### **2.3 Database beperkingregels**

Database van de klant.

Voor de klant zal het niet mogelijk zijn om meer dan 5 pagina's aan te maken.  
Het verwijderen van de website\_setup of het toevoegen van een website\_setup zal niet mogelijk zijn

Database van Hebbizz.

Voor de database van Hebbizz zullen geen beperkingsregels van kracht zijn.  
Wel moet er bij het bouwen van de applicatie gelet worden op dat als de cms-gebruiker een nieuwe gebruiker toevoegt dat deze ook in de database van Hebbizz komt te staan.

## 2.4 XUAN-model

Een XUAN-model is bedoeld om inzicht te geven in welke acties (taken) de gebruiker van het CMS kan uitvoeren.

Hierbij wordt aangegeven wat de gebruiker moet doen om een bepaalde actie uit te voeren.

Tevens wordt er weergegeven wat de database doet zodra er een taak uitgevoerd wordt en wat de output<sup>1</sup> is die het systeem teruggeeft aan de gebruiker.

(zie het onderstaande overzicht voor de verschillende taken)

### CMS-Klant

Taak	User action	Window action	System response	Database response
1.1 Overzicht tonen pagina's	Klik op "Pagina"	Geklikt op de pagina	Opent pagina met een overzicht huidige pagina's	Haalt alle pagina's op uit de database
1.2 Nieuw pagina maken	In getoonde scherm van taak 1.1 op "nieuw" klikken	Geklikt op nieuw	Opent pagina om nieuwe pagina te maken	Gegevens worden weggeschreven
1.3 Huidige pagina bewerken	In getoonde scherm van taak 1.1 op de naam van de betreffende pagina klikken	Geklikt op de naam van de pagina die gewijzigd moet worden	Opent scherm met de gegevens van de desbetreffende pagina	Haalt gegevens op uit de database en gegevens worden weggeschreven
1.4 Pagina verwijderen	In het getoonde scherm van taak 1.1 de checkbox voor het betreffende pagina aanvinken en op "verwijderen" klikken	Geklikt op verwijderen	Toont bevestiging voor verwijderen	Verwijdert de geselecteerde pagina(s) uit de database
2.1 Nieuw website lay-out maken	Klik op "Website"	Geklikt op de "Website"	Opent pagina om website lay-out te maken	Gegevens worden weggeschreven
2.2 Huidige website lay-out bewerken	Klik op "Website"	Geklikt op de "Website"	Opent scherm met de huidige lay-out van de website (mits deze bestaat)	Haalt gegevens op uit de database en Gegevens worden weggeschreven

<sup>1</sup> Output: dit zal bij het CMS een melding zijn wat er gebeurt is. Dus bijvoorbeeld bij het opslaan zal er een melding worden gegeven dat de gegevens zijn opgeslagen.

3.1 Overzicht tonen gebruikers	Klik op "Gebruikers"	Geklikt op de gebruikers	Opent pagina met overzicht van de huidige gebruikers	Haalt alle gebruikers op uit de database
3.2 Nieuw gebruiker maken	In getoonde scherm van taak 3.1 op "nieuw" klikken	Geklikt op nieuw	Opent pagina om nieuwe gebruiker aan te maken	Gegevens worden weggeschreven
3.3 Huidige Gebruiker bewerken	In getoonde scherm van taak 3.1 op de naam van het betreffende gebruiker klikken	Geklikt op de naam van de gebruiker die gewijzigd moet worden	Opent scherm met de gegevens van de desbetreffende gebruiker	Haalt gegevens op uit de database en gegevens worden weggeschreven
3.4 gebruiker verwijderen	In het getoonde scherm van taak 1.1 de checkbox voor het betreffende gebruiker aanvinken en op "verwijderen" klikken	Geklikt op verwijderen	Toont bevestiging voor verwijderen	Verwijderd de geselecteerde gebruiker(s) uit de database

CMS-Hebbizz administrator

Taak	User action	Window action	System response	Database response
1.1 Overzicht tonen cms-gebruikers	Klik op "CMS-gebruikers"	Geklikt op de cms-gebruikers	Opent pagina met overzicht van alle cms-gebruikers	Haalt alle gebruikers op uit de database
1.2 Nieuw cms-gebruiker aanmaken	In getoonde scherm van taak 1.1 op "nieuw" klikken	Geklikt op nieuw	Opent pagina om nieuwe gebruiker aan te maken	Gegevens worden weggeschreven
1.3 Huidige cms-gebruiker wijzigen	In getoonde scherm van taak 1.1 op de naam van de betreffende gebruiker klikken	Geklikt op de naam van de gebruiker die gewijzigd moet worden	Opent scherm met de gegevens van de desbetreffende gebruiker	Haalt gegevens op uit de database en gegevens worden weggeschreven
1.4 cms-gebruiker verwijderen	In het getoonde scherm van taak 1.1 de checkbox van de gebruiker aanvinken en op "verwijderen" klikken	Geklikt op verwijderen	Toont bevestiging voor verwijderen	Verwijdert de geselecteerde gebruiker(s) uit de database
2.1 Overzicht tonen klanten	Klik op "Klanten"	Geklikt op de "Klanten"	Opent pagina met overzicht huidige klanten	Haalt alle cms klanten op uit de database
2.2 Nieuw klant aanmaken	In getoonde scherm van taak 2.1 op "nieuw" klikken	Geklikt op "nieuw"	Opent pagina om nieuwe klant aan te maken)	Gegevens worden weggeschreven
2.3 Huidige klant gegevens wijzigen	In getoonde scherm van taak 2.1 op de naam van de betreffende klant klikken	Geklikt op de naam van de klant die gewijzigd moet worden	Opent scherm met de gegevens van de desbetreffende klant	Haalt gegevens op uit de database en gegevens worden weggeschreven
2.4 Klant verwijderen	In het getoonde scherm van taak 2.1 de checkbox voor het betreffende klant aanvinken en op "verwijderen" klikken	Geklikt op verwijderen	Toont bevestiging voor verwijderen	Verwijdert de geselecteerde klant(en) uit de database

### 3. Programmeertaal

---

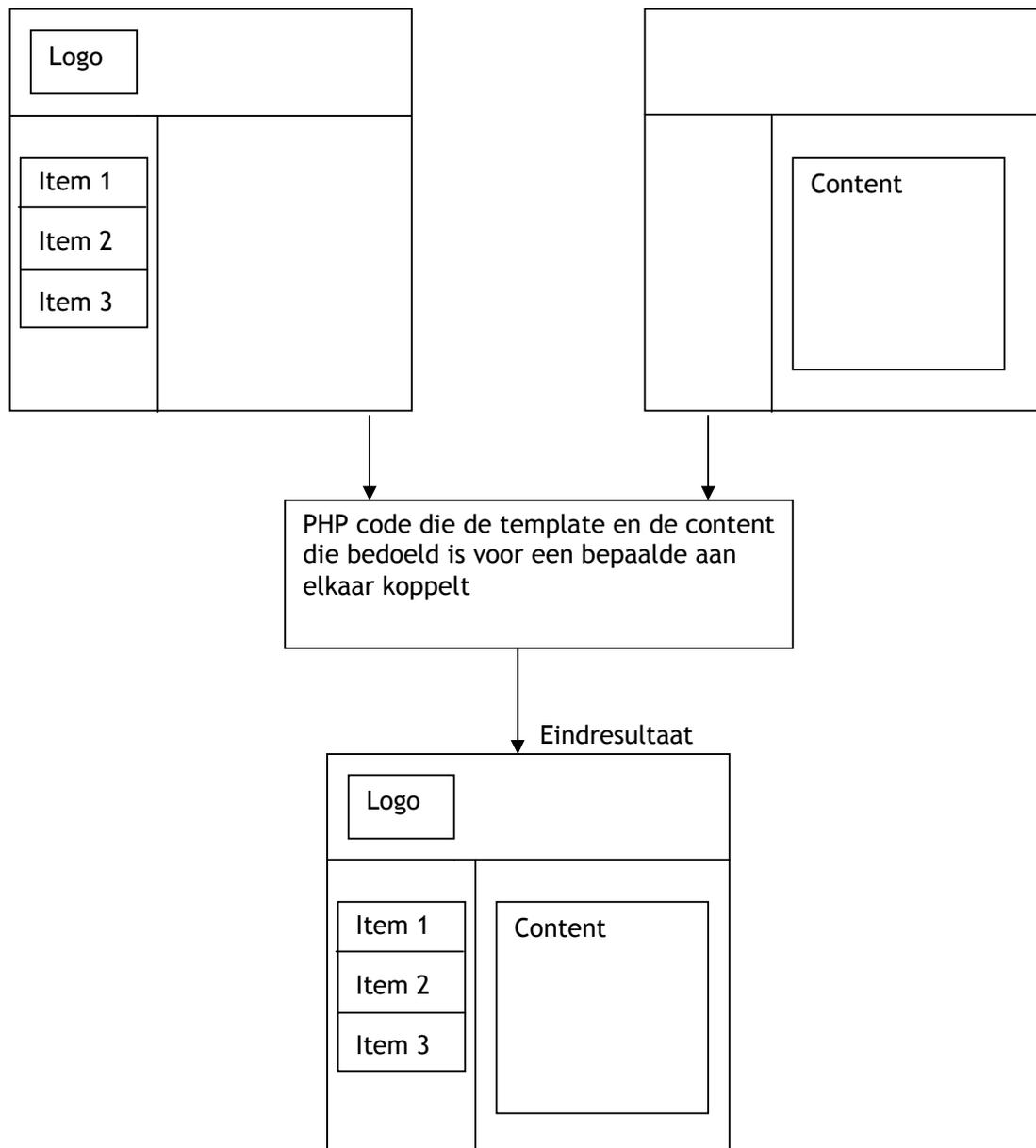
Tijdens dit project zal er gebruik worden gemaakt van de PHP als programmeertaal. Dit heeft mede te maken met het feit dat bij Hebbizz de website server waarop het CMS zal komen te staan Apache is.

Ook zullen de pagina die er gemaakt gaan worden gebruik maken een template. Hiervoor zal XTemplate worden gebruikt. XTemplate heeft als voordeel dat je de opmaak maar een keer hoeft te maken en deze opmaak kan je dan voor elke pagina hergebruiken. Tevens heeft XTemplate als voordeel dat de fysieke PHP code gescheiden blijft van de HTML code die gebruikt wordt voor de opmaak van de verschillende pagina.

Hieronder is een klein schema te zien van hoe XTemplate werkt en hoe deze in praktijk kan worden toegepast.

De template

HTML pagina met opmaak voor de content



Voorbeeld van hoe het bovenstaande eruit ziet in programmeertaal

De template:

```
<!-- BEGIN: template -->
<html>
<head>
<title>Voorbeeld</title>
</head>

<body>

<div id="header">
  
</div>

<div id="left">
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
  </ul>
</div>

<div id="content">
  <!-- BEGIN: content -->
    {FILE {FILE_CONTENT}}
  <!-- END: content -->
</div>

</body>
</html>
<!-- END: template - - >
```

Bij het in het rood aangegeven stukje zal door middel van XTemplate de content worden neergezet.

De HTML pagina met de content:

```
<p>
Hier komt de content die voor elke pagina kan verschillen.
</p>
```

Mocht er nou gebruik worden gemaakt van een database dan zal de html pagina er zo uitkomen te zien

```
<p>
{content}
</p>
```

Het stukje in het rood zal straks door de php pagina worden ingevuld.

De PHP pagina die de vorige twee pagina's samenvoegt.

```
<?
$GLOBALS["templateDSP"]="dsp/templateDSP.php"; = De Template

// template voor de content
$GLOBALS["contentDSP"]="dsp/contentDSP.php"; = De HTML pagina met de content

include('xtemplate.class.php');

    $xtpl = new XTemplate($GLOBALS["templateDSP"]);
    $xtpl->assign_file("FILE_CONTENT", $GLOBALS["contentDSP"]);

    if(file_exists($GLOBALS["templateDSP"]))
    {

        if(file_exists($GLOBALS["contentDSP"]))
        {

            $xtpl->assign("content", "Dit is de content voor de website die dus
bijvoorbeeld uit een database kan komen"); = Hier wordt de tekst in de HTML pagina gezet
            // parse de content
            $xtpl->parse("template.content"); = Hier wordt de content zichtbaar
gemaakt voor de website

            // -----
            $xtpl->parse('template'); = Hier wordt het geheel bij elkaar gevoegd.
            $xtpl->out('template'); = Hier wordt alles wat hiervoor geparsed
is naar de browser toegestuurd.
        } else {

            print "error";
        }
    } else {

        print "error";
    }
?>
```

Uiteindelijk zal dit dan de broncode van de pagina worden:

```
<html>
<head>
<title>Voorbeeld</title>
</head>

<body>

<div id="header">
  
</div>

<div id="left">
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
  </ul>
</div>

<div id="content">
  <p>
    Dit is de content voor de website die dus bijvoorbeeld uit een database kan komen
  </p>
</div>

</body>
</html>
```

## 4. Technische Infrastructuur

---

### 4.1 Benodigde Hard - en Software

In dit hoofdstuk zal beschreven worden aan welke minimale benodigdheden qua hardware en software de ontwikkelomgeving en productieomgeving moeten voldoen.

#### Ontwikkelomgeving

Hardware:

- Internetverbinding
- Monitor
- Desktopcomputer met muis en toetsenbord

Software:

- Windows XP professional
- Macromedia Dreamweaver 8
- PHP 5.0
- Apache 2
- Mysql 5
- Adobe Photoshop CS2
- Adobe Illustrator CS2
- FTP programma
- Firefox
- Internet Explorer 6 / 7

#### Productieomgeving

Hardware:

- Server
- Internetverbinding

Software:

- Linux
- FTP server
- Apache 2
- Mysql 5
- PHP 5.0

Daarnaast moet de productieomgeving toegankelijk zijn voor mensen van buiten af. Dit omdat de applicatie op 1 centrale plek zal komen te staan waar alle klanten voor het CMS inloggen.

## **4.2 De omgeving**

### **Ontwikkel - en Testomgeving**

Het ontwikkelen en testen van de verschillende onderdelen van het CMS zal gebeuren in de bij Hebbizz aanwezige acceptatie omgeving.

In deze omgeving worden alle websites voor klanten gemaakt.

Deze websites worden getest op functionaliteit en de wensen van de klant.

Als deze klant tevreden is dan zal de website over worden gezet naar de productieomgeving.

### **Productieomgeving**

De applicatie zal in “productie” worden genomen op een server van Hebbizz.

Deze server is de hoofdserver van Hebbizz wat betreft het hosten van websites.

Omdat het CMS op een centraal punt zal staan, zal deze door alle klanten worden gebruikt door het intypen van de een url. (bijvoorbeeld <http://cms.hebbizz.nl>, dit zal in de toekomst nog worden beslist).

In deze bijlage zijn een tweetal voorbeelden (opties) te zien van het gebruik van XTemplate.

**Optie 1** (zie Procesverslag figuur 16).

Bij deze optie wordt er gebruik gemaakt van 2 pagina's.  
In 1 van deze pagina's staat de gehele opmaak van de website geschreven.  
In de andere pagina wordt de php code geschreven.

Hieronder een voorbeeld van de verschillende pagina's

De HTML pagina:

```
<!-- BEGIN: template - - >
<html>
<head>
<title>XTemplate Test</title>
<style type="text/css">

body {
    margin:0;
    padding:0;
    width:100%;
    height:100%;
}

#header {
    margin:0;
    padding:0;
    width:100%;
    height:100px;
}

#spacer {
    clear:both;
    height:10px;
}

#leftmenu {
    float:left;
    width:300px;
    height:80%;
}

#content {
    float:left;
}

</head>

<body>
```

```
<div id="header">
  
</div>

<div id="spacer">
</div>

<div id="leftmenu">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Wie zijn we?</a></li>
    <li><a href="#">Wat doen we?</a></li>
  </ul>
</div>

<div id="content">
  {tekst}
</div>

</body>
</html>
<!-- END: template -->
```

De PHP pagina:

```
<?
include ('xtemplate.class.php');

$xtpl = new XTemplate('htmlpagina.html');

$xtpl->assign('logo', 'gfx/logo.gif');
$xtpl->assign('tekst', 'Dit is de content die op de plek komt te staan waar in HTML pagina
{tekst} staat').

$xtpl->parse('template');
$xtpl->out('template');

?>
```

**Optie 2** (zie Procesverslag figuur 16):

Bij deze optie wordt er gebruik gemaakt van 3 pagina's.

Bij het bouwen van het CMS is er gekozen voor deze optie.

Het voordeel van deze opzet is dat je de daadwerkelijke opmaak gescheiden houdt van de content.

Zo hoef je maar één keer de opmaak van de hele website of CMS te maken en deze file kan je dan meerdere keren gebruiken en elke aanpassing die je in deze file maakt geldt gelijk voor de pagina's waarin deze file gebruikt wordt.

**De HTML pagina** (de lay-out):

```
<!-- BEGIN: template -->
<html>
<head>
<title>XTemplate Test</title>
<style type="text/css">

body {
    margin:0;
    padding:0;
    width:100%;
    height:100%;
}

#header {
    margin:0;
    padding:0;
    width:100%;
    height:100px;
}

#spacer {
    clear:both;
    height:10px;
}
```

```

#leftmenu {
    float:left;
    width:300px;
    height:80%;
}

#content {
    float:left;
}

</head>

<body>

<div id="header">
    
</div>

<div id="spacer">
</div>

<div id="leftmenu">
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Wie zijn we?</a></li>
        <li><a href="#">Wat doen we?</a></li>
    </ul>
</div>

<div id="content">
    <!--BEGIN : content -->
        {FILE {FILE_CONTENT}}
    <!-- END: content -->
    {tekst}
</div>

</body>
</html>
<!-- END: template -->

```

## De HTML pagina (de content):

```
<table>
  <tr>
    <th>Naam:</th>
    <th>Username:</th>
    <th>Password:</th>
  </tr>
  <tr height="5px">&nbsp;</tr>
  <!-- BEGIN: overzicht -->
  <tr>
    <td>{data.naam}</td>
    <td>{data.username}</td>
    <td>{data.password}</td>
  </tr>
  <!-- END: overzicht -->
</table>
```

## De PHP code

```
<?
1. $template = "dsp/templateDSP.php";

// template voor de content
2. $content = "dsp/contentDSP.php";

3.    include('xtemplate.class.php');

4.    $xtpl = new XTemplate($template);
5.    $xtpl->assign_file("FILE_CONTENT", $content);

6.    if(file_exists($template))
    {

7.        if(file_exists($content))
        {

8.            $sql = "SELECT * FROM users";
9.            $RS = mysql_query($sql);
10.           while($data = mysql_fetch_array($RS))
            {
11.                $xtpl->assign("data", $data);
12.                $xtpl->parse("template.content.overzicht");
            }

13.           $xtpl->parse("template.content");

// -----
14.           $xtpl->parse('template');
15.           $xtpl->out('template');

        } else {
```

```

                print "error";
            }
        } else {
            print "error";
        }
    }
?>

```

Hier volgt de uitleg van bovenstaande code.

1. Hier wordt aangegeven welke bestand met de opmaak gebruikt gaat worden. Het kan dus voorkomen dat er voor een website meerdere templates gebruikt worden.
2. Hier wordt aangegeven welke bestand gebruikt wordt voor de content.
3. Hier wordt een bestand “geinclude”. In dit bestand staan alle functies die XTemplate te bieden heeft.
4. Hier wordt de lay-out pagina in het geheugen geladen.
5. Met `$xtpl->assign_file` wordt de inhoud van HTML pagina met de content op de plek gezet waar in de HTML pagina lay-out `{FILE {FILE_CONTENT}}` staat.
6. Dit is een check om te kijken of het bestand met de lay-out (oftewel de template) wel bestaat.
7. Dit is een check om te kijken of het bestand voor de content wel bestaat.
8. Dit is een simpele SQL-statement.
9. Hier wordt de SQL-statement uitgevoerd op de database.
10. Hier wordt er een loop uitgevoerd op alle regels die zijn teruggekomen van de Database.
11. Met `$xtpl->assign` kan je zowel alleen een stukje tekst aan een bepaalde waarde toekennen (zoals te lezen was bij optie 1), maar je kan ook een hele “array” aan een bepaalde waarde toekennen zoals in dit voorbeeld. Deze waardes uitbijvoorbeeld een array zullen bijvoorbeeld op deze manier in de pagina met de content worden aangeroepen `{data.username}`.
12. Met `$xtpl->parse` wordt de HTML code die bijvoorbeeld tussen de tekst `<! - - BEGIN: overzicht - - >` en `<! - - END: overzicht - - >` op het scherm getoond. Het belangrijk om met XTemplate altijd van binnen naar buiten te parsen. Dus zoals er in het voorbeeld in optie 2 bij de HTML pagina’s te zien is hebben we eerst een `<! - - BEGIN: overzicht - - >` gevolgd door `<! - - BEGIN: content - - >` en als laatste `<! - - BEGIN: template - - >`. En als we dan gaan kijken in de PHP code zal ook eerst “overzicht” worden geparsed (nr 12.), daarna “content” (nr 13.) en als laatste “template” (nr 14.).

Bij nummer 15 wordt hete geheel dat nu geparsed is naar de internet browser toegestuurd d.m.v de functie `$xtpl->out`.