Technisch Ontwerp

iConcepts

“EenmaalAndermaal”

|  |  |
| --- | --- |
| *Auteurs* | Groep 14  Mehmet Batal: ID 600720,  Danny Hageman 608227,  Hendrik Otten 619129,  Bram Soutendam 599164,  Elviana Cornelissen 598830 |
| *Cursus* | *I-Project* |
| *Opdrachtgever* | *iConcepts* |
| *Begeleiders* | *Rein Harlé – Product Owner*  *Ben van Hoof – Scrum coach*  *Rody Aldenhoven – Expert Professional Skills* |
| *Onderwijsinstelling* | *Hogeschool van Arnhem en Nijmegen* |
| *Plaats & datum* | *Arnhem, 11 juni 2019* |
| *Versie nummer*  *Status* | *1.0*  *In Bewerking* |



Inhoudsopgave

[1. Inleiding 3](#_Toc9584632)

[2. Front-end-framework 4](#_Toc9584633)

[3. Responsiveness 4](#_Toc9584634)

[4. Netwerkarchitectuur 5](#_Toc9584635)

[5. Database 6](#_Toc9584636)

[5.1 Datatypen & defaults 6](#_Toc9584637)

[5.2 Check-Contraints 6](#_Toc9584638)

[5.3 Conversie 6](#_Toc9584639)

# 1. Inleiding

In dit document beschrijft het team de technische specificaties van de website ‘’EenmaalAndermaal” voor iConcepts. Er wordt beschreven welke functies en variabelen er zijn gebruikt voor het realiseren van de website. Ook worden de gebruikte servers getoond en hoe ze in verbinding staan met de website. In dit document wordt dus vooral verteld ‘hoe’ verschillende componenten werken op/met de site.

# 2. Front-end-framework

Wij hebben gekozen voor ‘Foundation’ als front-end-framework. Hoe zijn wij tot deze keuze gekomen? Er is een onderzoeksrapport gemaakt over verschillende front-end-frameworks. Eerst is er een longlist gemaakt van vijf verschillende front-end-frameworks. Op die 5 frameworks hebben wij criteria toegepast, waardoor er drie overbleven. Deze drie zijn allemaal getest, en de beste hebben wij gekozen. Foundation werkte voor ons het makkelijkst en het best, ook waren er veel voorbeelden te vinden op hun website. Doordat er zo veel voorbeelden te vinden waren is onze keuze uiteindelijk gevallen voor Foundation.



Figuur 1: Foundation

# 3. Responsiveness

Hoe is onze site responsive gemaakt? Wij hebben gebruik gemaakt van het framework ‘Foundation’. Foundation stelde een grid beschikbaar. Deze grid is opgedeeld in 4 vakken: een header, aside, main en een footer. Alles wat wij gemaakt hebben zit verwerkt in de grid. Hierdoor is alles netjes responsive. Op de mobiel hebben wij zelf een aantal aanpassing gemaakt, zoals het verplaatsen van de zoekbalk en een kleiner logo gebruiken zodat alles netjes blijft.

# 4. Netwerkarchitectuur

# 5. Database

## 5.1 Datatypen & defaults

## 5.2 Check-Contraints

In onze database zitten een aantal check-constraints. Deze check constraints zitten in onze database om vuile data te omzeilen, zodat de database niet vol raakt met onzinnige data. Hieronder staan al onze check-constraints per tabel beschreven.

**Tabel ‘gebruiker’**

* Op de kolom ‘geboortedatum’ zit een check constraint die een datum dan later dan vandaag tegenhoud.
* Op de kolom ‘wachtwoord’ zitten twee check constraints, de eerste is dat een wachtwoord minimaal 7 karakters moet bevatten. De tweede is dat een wachtwoord minimaal één letter en één cijfer moet bevatten.

**Tabel ‘voorwerp’**

* Op de kolom ‘betalingswijze’ zit een check constraint die kijkt of de betalingswijze in een van de vier opgegeven opties zit.
* Op de kolom ‘startprijs’ zit een check constraint die kijkt of de opgegeven startprijs groter is dan nul.
* Op de kolom ‘veilinggesloten’ zit een check constraint die kijkt of de data ‘wel’ of ‘niet’ is. Hiermee wordt gekeken of de veiling is gesloten.
* Met de check-constraint ‘ck\_voorwerp\_verkoopprijs\_negatief’ wordt gekeken of de verkooprijs gelijk of groter is dan de startprijs.
* Met de check-constraint ‘ck\_voorwerp\_verkoper\_geen\_koper’ wordt gekeken of de koper van een product niet ook de verkoper was van dat product.
* Met de check-constraint ‘ck\_looptijd’ wordt gekeken of de data ‘1-3-5-7-10’ gebruikt wordt, dit is de looptijd in dagen. Er kan dus bijvoorbeeld geen 12 dagen gekozen worden want dit wordt tegen gehouden door de check-constraints.

**Tabel ‘bod’**

* Met de check constraint ‘ck\_NietBiedenEigenProduct’ wordt gekeken of de koper niet op zijn eigen product biedt.
* Met de check constraint ‘CHK\_VeilingOpen’ wordt gekeken of een veiling nog open staat of al gesloten is.

## 5.3 Conversie

# 6 Javascript, AJAX en jQuery

## 6.1 Waarom javascript?

In het product is javascript gebruikt. Voor de timer, limiet karakters, navigatiebar, laagste categorieën kiezen en het toevoegen van meerdere afbeeldingen.

Waarom is er gebruik gemaakt van javascript? De grootste reden is dat het mogelijk is om met javascript een pagina te updaten zonder dat de gebruiker de pagina zelf herlaadt. Met PHP is dat wat lastiger bijvoorbeeld om een timer te gebruiken die afloopt zonder dat de gebruiker zelf de pagina herlaadt. Omdat je niet live ziet wanneer het precies afloopt op seconden. Kan het zijn dat de gebruiker te laat is om te bieden zonder dat hij het zelf merkt. Dus hierdoor voorkom je dat hij zelf de pagina moet updaten.

Er zijn verschillende manieren gebruikt in javascript om een deel werkend te krijgen. Hieronder zal er uitgelegd worden welke delen in het product javascript gebruikt heeft. Ook wordt er vermeldt hoe het gedaan is.

## 6.2 Javascript

Gewoon standaard javascript zijn de volgende delen gemaakt.

Hoe werkt de timer?   
De timer voert een functie uit waar het begint van welke datum je af wilt tellen tot de einddatum. De functie berekend elke seconde, minuten, uren en dagen totdat de veiling in dit geval eindigt. Daarna zal er weergegeven worden dat de veiling is afgelopen.

Functie max aantal karakters in een <textarea>, in deze functie geef je het limiet aan. Het zal de karakters tellen en als deze het max aantal karakters gehaald heeft kan er niet meer verder getypt worden.

## 6.3 Ajax

Bij het navigatie menu en het kiezen van de laagste rubriek is er beide gebruik gemaakt van AJAX. AJAX is een manier om informatie op te vragen via de server en ook informatie te verkrijgen van de server nadat de pagina geladen is. De informatie wordt automatisch geüpdatet zodat de relevante informatie wordt laten zien.

Hoe werkt het navigatie menu?

Het navigatie menu werkt als volgt, er wordt op een rubriek geklikt en deze klapt een sub-rubriek uit. En zo verder totdat het laagste niveau is bereikt. De reden dat hiervoor javascript gebruikt is komt doordat er performance issues ontstond. De pagina laadt te lang doordat het door een heel script moet waar het controleert of elk rubriek en sub rubriek heeft en zo verder. Met 3000+ rubrieken is dat wel te begrijpen waarom het zo lang duurt. Daardoor is er gebruik gemaakt van jQuery en Ajax. Met elke klik zal er een verzoek verstuurd worden die controleert of elk rubriek een sub-rubriek heeft. En zal een HTML-code aanmaken die de sub-rubriek laat zien. Als er nog een keer op gedrukt op dezelfde rubriek zal het inklappen en de HTML-code verwijderen zodat het voorkomt dat de pagina eigenlijk te veel html heeft en te lang moet laden.

Hoe werkt het zoeken van rubriek op laagste niveau?

Als er een optie gekozen is zal er automatisch zoeken naar sub rubriek en die weergeven in de volgende select box. Totdat het op het laagste niveau is. Dit wordt verstuurd met het form naar de database. Dit werkt hetzelfde met het navigatie menu als er een rubriek gekozen is zal hij vragen naar de sub rubrieken.   
  
  
6.4 jQuery

jQuery is ook gebruikt in het maken van het product. jQuery is een library die je kan gebruiken wanneer je javascript gebruikt. Het is makkelijk om te gebruiken om functies uit te laten voeren, na het klikken op een link bijvoorbeeld.

Met de ingebouwde functies on() en append() kon er veel gedaan worden. Door bijvoorbeeld het toevoegen van meerdere bestanden. Als er op de knop geklikt wordt dan zal er een functie worden uitgevoerd die er voor zorgt dat er een nieuwe rij wordt toegevoegd. Met de append() functie kan er gemakkelijk een HTML-code toegevoegd worden.

De knop verdwijnt nadat er vier rijen beschikbaar zijn. Dit komt door de css() functie waar de inhoud van de css veranderd wordt. Dus in dit geval is dat *visibility : visible* naar *visibility : hidden* .

Als je binnen een <div> oftewel selector bent en je wilt op <a> klikken dan kan je on() functie gebruiken. Dit zorgt ervoor dat je binnen in de container een andere selector de child-selector kan oproepen en dat die de functie doorloopt. Je geeft specifieker aan welke selectors de functie uit moeten voeren.

Als een voorbeeld kan ik de navigatie menu gebruiken (zie deel van de code hieronder), hierbij als er op een sub-rubriek geklikt wordt, controleert de functie of het binnen de id van *menuRubriek* zit. Daarna zal er gecheckt worden of de child-selector een class heeft die de naam *addressClick4* heeft. Als dat allemaal voldoet aan de gevraagde eisen zal deze de functie uit voeren.

$("#menuRubriek").on("click", ".addressClick4", function() {

//wat de functie moet uitvoeren

});

*Figuur @@ : Voorbeeldcode on() functie*

//Wordt nog aangepast

//nog niet klaar