

Technische verslag

Het CMS voor iedereen

Dante Klijn



Samenvatting

Hier komt de nieuwe samenvatting

Contactgegevens

Student

Naam	Dante Klijn
Studentnummer	4565908
Academisch jaar	2023/2024
E-mail	dante.klijn@student.nhlstenden.com
Telefoonnummer	+31 (0)6 24 76 59 74

Onderwijsinstelling

Naam	NHL Stenden University of Applied Sciences
Course	HBO-ICT
Locatie	Rengerslaan 8-10, 8917 DD, Leeuwarden
Telefoonnummer	+31 (0)88 991 7000

Docentbegeleider

Naam	Stefan Rolink
Email	stefan.rolink@nhlstenden.com
Telefoonnummer	+31 (0)6 42 28 30 77

Afstudeercommissie

Email	afstuderenschoolofict@nhlstenden.com
-------	--------------------------------------

Examencommissie

Email	examencommissiehboict@nhlstenden.com
-------	--------------------------------------

Organisatie

Naam	Snakeware New Media B.V.
Locatie	Veemarktplein 1, 8601 DA, Sneek
Telefoonnummer	+31 (0)515 431 895

Bedrijfsbegeleider

Naam	Thom Koenders
Email	thom@snakeware.com
Telefoonnummer	+31 (0)6 13 09 18 51
Rol	Senior software developer

Versiebeheer

Versie	Datum	Veranderingen
0.1	TBD	Eerste hoofdstukken

Woordenlijst

Contentmanagementsysteem Een contentmanagementsysteem is een softwaretoepassing, meestal een webapplicatie, die het mogelijk maakt dat mensen eenvoudig, zonder veel technische kennis, documenten en gegevens op internet kunnen publiceren (contentmanagement). Als afkorting wordt ook wel CMS gebruikt.

Graphical user interface Een graphical user interface (GUI), is een manier van interacteren met een computer waarbij grafische beelden, widgets en tekst gebruikt worden.

Search engine optimization Search Engine Optimisation (SEO), zijn alle processen en verbeteringen die als doel hebben een website hoger in Google te laten verschijnen.

Inhoudsopgave

Samenvatting	iii
Woordenlijst	v
1 Inleiding	2
1.1 Organisatieomschrijving	2
1.2 Context	2
1.3 Aanleiding	3
1.4 Opdrachtomschrijving	4
1.5 Leeswijzer	4
2 Ontwerp	5
2.1 Scenario's	5
2.2 Logical View	5
2.3 Development view	5
2.4 Process View	5
2.5 Physical view	5
3 Realisatie	6
3.1 Tools	6
4 Beheer Validatie en Verificatie	7
4.1 Codestandaarden	7
4.2 Codereviews	7
4.3 Versiebeheer	7
5 Conclusie	8
6 Reflectie	9

Hoofdstuk 1

Inleiding

Dit is het technische verslag voor het “Het CMS voor iedereen” project. Het technische verslag is een onderdeel van de afstudeerperiode binnen NHL Stenden Hogeschool. Het technische verslag dient als documentatie en verantwoording voor het product dat gemaakt is tijdens de afstudeerperiode. In de volgende sectie wordt de organisatie beschreven verder wordt de aanleiding en de context van de afstudeeropdracht beschreven.

1.1 Organisatieomschrijving

Snakeware New Media B.V. (Snakeware) is een E-business bureau gevestigd in Nederland. Haar aangeboden diensten omvatten het adviseren, bouwen en onderhouden van digitale producties, met een focus op websites, webshops en mobiele apps (Snakeware, 2022b). Op het moment van schrijven telt Snakeware meer dan 60 werknemers, elk met verschillende specialiteiten. Ze leveren services aan welbekende organisaties zoals DPG Media, DekaMarkt en Poiesz supermarkten (Snakeware, 2022a).

1.2 Context

Snakeware heeft een platform genaamd “Snakeware Cloud” dit platform is een contentmanagementsysteem (CMS) waarmee ze digitale content kunnen leveren voor haar (grotere) klanten. Snakeware Cloud is een applicatie waarmee Snakeware en haar klanten webapplicaties kan inrichten en voorzien van content.

De klant van Snakeware kan zijn of haar website zelf inrichten door middel van het specificeren van de content op de verschillende pagina's. Dit wordt gedaan door middel van artikelen die door het CMS gebruikt kunnen worden. De content van het artikel kan verschillen tussen simpele tekst, vragenlijst, webshop producten, etc. Hiernaast zijn er ook search engine optimization (SEO) opties binnen Snakeware Cloud om de site goed te kunnen vinden op het internet. Hierbij zijn er opties zoals de mogelijkheid om de title tags en zoekwoorden toe te kunnen voegen in de head (Mozilla, 2023c)

Hierom heeft Snakeware Cloud veel features en configuratie stappen wat het complex en duur maakt om een relatief kleine webapplicatie te maken voor kleinere klanten. Dit zorgt ervoor dat Snakeware zich niet kan vestigen in een markt met veel kleinere klanten, en hierdoor omzet misloopt.

1.3 Aanleiding

Het huidige platform is 21 jaar oud en er is veel functionaliteit in de loop der jaren aan toegevoegd. Omdat Snakeware Cloud een oud platform is zijn er veel technieken en best practices gebruikt die nu niet meer als optimaal worden beschouwd. Deze technieken waren erg geïntegreerd in Snakeware Cloud en er is het verleden gekozen om niet de code te herschrijven om het aan de huidige standaarden te voldoen van andere projecten. Een voorbeeld hiervan is tabel naam prefix afkortingen bij elke kolom zetten, of gigantische C# (Microsoft, 2022) files van 10 000 regels met verschillende functies. Deze functies houden zich niet aan de *Single Responsibility Principle* van de SOLID ontwerpmethode (Watts, 2020) wat het moeilijk maakt om het huidige CMS te onderhouden.

Ook zijn er technieken toegepast die nu niet meer relevant zijn. Een voorbeeld hiervan is dat het CMS gebruikmaakt van JavaScript (Mozilla, 2023b) en toen ze er mee begonnen bestonden JavaScript classes (Mozilla, 2023a) nog niet, dus hebben ze die zelf geïmplementeerd. Deze oudere technieken en standaarden zorgen ervoor dat het meer tijd kost om het CMS te onderhouden vanwege de extra code. Dit zorgt ervoor dat het meer tijd en geld kost om het Snakeware Cloud uit te breiden.

Een van de voornaamste uitdaging met Snakeware Cloud betreft de verouderde datastructuur van de applicatie. Deze veroudering is het gevolg van een initiële ontwikkeling waarbij onvoldoende rekening werd gehouden met toekomstige functionaliteitsuitbreidingen in het systeem. Als gevolg daarvan is de onderliggende datastructuur niet aangepast, maar zijn er elementen aan toegevoegd. Dit heeft geresulteerd in database query's van duizenden regels en complexe relaties tussen tabellen in de database. Dit huidige scenario bemoeilijkt aanzienlijk het toevoegen van nieuwe functionaliteiten, wat resulteert in aanzienlijke tijd en kosten investeringen.

Hierom wil Snakeware een nieuw systeem met een nieuwe datastructuur. Door het gebruiken van een nieuwe softwarearchitectuur zouden er velen problemen opgelost kunnen worden die nu voor komen. Omdat er een nieuwe datastructuur moet komen en de logica van het oude systeem nauw verbonden is met de datastructuur is het niet mogelijk om de oude code opnieuw te gebruiken.

1.4 Opdrachtomschrijving

De opdracht is om een proof of concept CMS-API te ontwikkelen die gebruikt maakt van een datamodel en systeemarchitectuur dat flexibeler, onderhoudbaarder is en gebruik maakt van moderne best practices. Tijdens de afstudeeropdracht wordt er primair op het datamodel en de systeemarchitectuur gefocust. Omdat er nog geen concreet datamodel en systeemarchitectuur is zal dit onderzocht en ontworpen moeten worden.

De opdracht omvat het achterhalen van de requirements, ontwerpen en ontwikkelen van het proof of concept met als focus een nieuw datamodel, met de essentiële functionaliteiten.

Het huidige Snakeware Cloud platform bestaat uit 2 verschillende graphical user interfaces (GUI):

- Snakeware Cloud GUI
- Klant webapplicatie

Met de Snakeware Cloud GUI kan de klant de content van de website aanpassen. Door middel van de webapplicatie kan de eindgebruiker de content bekijken en er mee interacteren. Er is voor gekozen om niet de Snakeware Cloud GUI te realiseren om de afstudeeropdracht in scope te houden. Er is wel voor gekozen om de klant webapplicatie in zijn minimale vorm uit te werken. Om de userflows van de applicatie toch te kunnen testen wordt er gebruik gemaakt van postman workflows (Postman.com, 2023)

Het doel van het proof of concept is dat er aangetoond kan worden dat door het gebruiken van een nieuw datamodel en systeemarchitectuur ook services verleend kunnen worden aan kleinere klanten. Dit zou eventueel ook een startpunt zijn om op verder te bouwen.

1.5 Leeswijzer

als laatste

Hoofdstuk 2

Ontwerp

dit is het ontwerp hoofdstuk

2.1 Scenario's

Dit worden de Senario's

2.2 Logical View

dit wordt de logical view

2.3 Development view

Dit wordt de development view

2.4 Process View

dit wordt de process view

2.5 Physical view

dit wordt de Physical view

Hoofdstuk 3

Realisatie

dit is het realisatie hoofdstuk

3.1 Tools

Er zijn 2 verschillende applicaties ontwikkeld dit zijn de frontend en de backend applicatie.

Frontend

Voor de frontend is er gekozen om te werken met Vue en Nuxt 3. Vue is een Javascript / Typescript framework dat het makkelijker maakt om grotere frontenden te maken voor web applicaties. Nuxt 3 is een Vue framework wat het makkelijker maakt om vue applicaties te maken en meerdere rendering opties aan bied. Er is voor Vue en Nuxt 3 gekozen om dat de gebruikte standaard is binnen Snakeware. Door dit te doen kunnen de frontend developers snel het proof of concept op pakken. Voor het schrijven van de frontend code is gebruik gemaakt van **Neo Vim** / **webstorm**.

Backend

Voor de Backend is gebruik gemaakt van C# en .Net 8 een C# framework. Dit is gedaan omdat C# de taal is waar alle andere services en developers gebruik van maken. Er is voor .Net 8 gekozen omdat de laatste *LTS* versie is van .Net. De code voor de backend wordt geschreven in Visual Studio 2022.

Database Er is voor een NoSQL database gekozen (kijk naar ontwerp). Voor de NoSQL database is er gekozen voor MongoDB. MongoDB is de grootste NoSQL database provider en heeft goede drivers voor C#.

Hoofdstuk 4

Beheer Validatie en Verificatie

Dit is het hoofdstuk over het beheer validatie en verificatie

4.1 Codestandaarden

4.2 Codereviews

Tijdens het development proces is er gebruik gemaakt van codereviews voor zowel de frontend en backend code. Voor de backend code is Kevin (Backend software engineer) benaderd om de code te bekijken en begeven van feedback. Daarnaast is er samen met Thom (Senior software engineer) een keer per week een progressie update gegeven van de development en is daar ook veel kennis gedeeld.

voor frontend code zal ik nog vragen voor advies maar dat is nog niet voor gekomen.

4.3 Versiebeheer

Voor versiebeheer is er gebruik gemaakt van git en als repository platform Bitbucket.

Hoofdstuk 5

Conclusie

dit is de conclusie

Hoofdstuk 6

Reflectie

dit is de reflectie

Bibliografie

- Microsoft. (2022). *Een rondleiding door de C#-taal*. Verkregen 10 oktober 2023, van <https://learn.microsoft.com/nl-nl/dotnet/csharp/tour-of-csharp/>
- Mozilla. (2023a). *Classes*. Verkregen 11 oktober 2023, van <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes>
- Mozilla. (2023b). *JavaScript*. Verkregen 10 oktober 2023, van <https://www.javascript.com/>
- Mozilla. (2023c). *What's in the head? Metadata in HTML*. Verkregen 11 oktober 2023, van https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML
- Postman.com. (2023). *Customize request order in a collection run*. Verkregen 6 oktober 2023, van <https://learning.postman.com/docs/collections/running-collections/building-workflows/>
- Snakeware. (2022a). *Cases*. Verkregen 10 oktober 2023, van <https://www.snakeware.nl/cases>
- Snakeware. (2022b). *What we do*. Verkregen 10 oktober 2023, van <https://www.snakeware.com/what-we-do>
- Watts, S. (2020). *The importance of SOLID Design Principles*. Verkregen 11 oktober 2023, van <https://www.bmc.com/blogs/solid-design-principles/>