|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
|  | |  |  |
|  | |  |  |
| **Naam van opdracht:** | W.A.F. | |
| **Documentnaam:** | Onderzoeksrapport | |
| **Documentversie:** | 2.0 | |
| **Datum:** | 14-04-2014 | |
|  |  | | |
| **WpT(s):** | 1.1 t/m 1.9  2.1 t/m 2.5  3.1 t/m 3.3 | | |
|  |  | | |
| **Gemaakt door:** | Michael Oosten | | |
|  | Jan Hendrik Haanstra | | |
|  | Frans Tuinstra | | |
| **Klas:** | DEV42AB | | |
|  |  | | |

|  |  |
| --- | --- |
| Gegevens Docent |  |
| **Opdrachtgever:** | Jan Zuur |
| **Bedrijfsnaam:** | Friesland College |
| **Locatie:** | Heerenveen |
| **Adres:** | Abe Lenstraboulevard 29 |
| **Telefoon:** | 06 12 34 56 78 |
| **E-mail:** | j.zuur@fcroc.nl |
| **Functie:** | Docent |
|  |  |
| **Opdrachtgever:** | Rones Pander |
| **Bedrijfsnaam:** | Friesland College |
| **Locatie:** | Heerenveen |
| **Adres:** | Abe Lenstraboulevard 29 |
| **Telefoon:** | 06 12 34 56 78 |
| **E-mail:** | r.pander@fcroc.nl |
| **Functie:** | Docent |
|  |  |
| **Opdrachtgever:** | Akif Gümüssu |
| **Bedrijfsnaam:** | Friesland College |
| **Locatie:** | Heerenveen |
| **Adres:** | Abe Lenstraboulevard 29 |
| **Telefoon:** | ?? |
| **E-mail:** | ?? |
| **Functie:** | Docent |
| Gegevens Student |  |
| **Naam:** | Michael Oosten |
| **Locatie:** | Wolvega |
| **Adres:** | Govert Flickstraat 1 |
| **Telefoon:** | 0657335300 |
| **E-mail:** | michael.oosten001@fclive.nl |
| **Functie:** | Cursist |
|  |  |
| **Naam:** | Jan Hendrik Haanstra |
| **Locatie:** | Oudehorne |
| **Adres:** | Schoterlandseweg 99 |
| **Telefoon:** | 0615902349 |
| **E-mail:** | [janhendrik.haanstra001@fclive.nl](mailto:janhendrik.haanstra001@fclive.nl) |
| **Functie:** | Cursist |
|  |  |
| **Naam:** | Frans Tuinstra |
| **Locatie:** | Heerenveen |
| **Adres:** | Kwartelstraat 28 |
| **Telefoon:** | 0624381253 |
| **E-mail:** | [frans.tuinstra001@fclive.nl](mailto:frans.tuinstra001@fclive.nl) |
| **Functie:** | Cursist |
|  |  |

# Versiebeheer

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Datum** |
| 1.0 | Eerste opstelling van alle gegevens | 18-04-2014 |
| 2.0 | Feedback ontvangen en verwerkt | 22-04-2014 |

# Inhoudsopgave

[Inleiding 1](#_Toc387740217)

[Opdrachtomschrijving 1](#_Toc387740218)

[Doel van het onderzoeksrapport 1](#_Toc387740219)

[Onderzoek 2](#_Toc387740220)

[Soorten architectuur van software framework: 2](#_Toc387740221)

[Model-view-controller (MVC) 2](#_Toc387740222)

[Presentation–abstraction–control (PAC) 2](#_Toc387740223)

[Three-tier organization 3](#_Toc387740224)

[Framework toepassingen 3](#_Toc387740225)

[Wat gaan wij maken 3](#_Toc387740226)

[De eisen van het framework: 3](#_Toc387740227)

[Bronvermelding 4](#_Toc387740228)

[Overeenkomst 4](#_Toc387740229)

# Inleiding

Wij hebben de opdracht gekregen om een web application framework (WAF) te ontwikkelen die ons bij staat met het ontwikkelen van dynamische websites. In dit document is weergegeven wat wij hebben onderzocht en stellen wij de eisen en wensen op voor ons framework.

# Opdrachtomschrijving

De eerste weken zullen we bezig gaan met het onderzoeken en analyseren van het product d.m.v. een onderzoeksrapport(analyse), functioneel ontwerp en een technisch ontwerp. Vervolgens gaan we verder met het realiseren van het product als het onderzoeksrapport naar wens is.

# Doel van het onderzoeksrapport

Doel van het onderzoeksrapport is om te onderzoeken wat een framework is, en aan de hand van ons onderzoek gaan wij beslissen waar wij ons framework voor gaan maken. Dit zal ook worden opgeschreven in dit document.

# Onderzoek

## Soorten architectuur van software framework

De meeste web applicatie frameworks zijn gebaseerd op het MVC pattern.

Verschillende soorten architectuur van frameworks zijn:

* [Hierarchical model–view–controller](http://en.wikipedia.org/wiki/Hierarchical_model%E2%80%93view%E2%80%93controller)
* [Presentation–abstraction–control](http://en.wikipedia.org/wiki/Presentation%E2%80%93abstraction%E2%80%93control)
* [Three-tier architecture](http://en.wikipedia.org/wiki/Three-tier_architecture)

### Model-view-controller (MVC)

De applicatie kan verdeeld worden in drie delen: model, view en controller.

* Het Model is verantwoordelijk voor toegankelijkheid van de database(s) en het ophalen van gestructureerde data.
* De Controller heeft als taak om de business-logica van de applicatie te behandelen. Het moet de Model gebruiken om toegang te krijgen tot de database samen met de user-input om de informatie te bouwen of de actie van de user uit te voeren.
* De View is het deel dat voor de gebruiker zichtbaar is. Het bestaat uit gegevens die gegenereerd zijn door het Controller deel die worden weergegeven via het View gedeelte, en input die de gebruiker invoert komt in de View binnen en wordt doorgegeven aan de Controller.

#### Push-based vs. Pull based

Push-based en pull-based zijn 2 manieren waarop de MVC wordt gehandled. Zodra er contact komt tussen de view en de controller zijn er 2 opties:

* De view haalt (pull) gegevens uit de database. Hierbij verwacht de view wel data te ontvangen.
* De controller stuurt (push) de data naar de view toe die het dan laat zien aan de gebruiker.

### Presentation–abstraction–control (PAC)

Volgens onze docent werken wij volgens dit principe. Het lijkt wel op de MVC methode, maar is wel anders. Hier beneden is wat we hebben onderzocht.

In tegenstelling tot MVC worden de onderdelen van PAC als een hierarchische structuur behandeld. Deze bestaat uit presentation, abstraction en control. Deze middelen communiceren alleen met elkaar door het control gedeelte van elk drietal. Nog een verschil van MVC en PAC is dat presentation (view in MVC) en de abstraction (model in MVC) volledig geisoleerd is van elkaar. Dit biedt de mogelijkheid om de presentation en abstraction afzonderlijk multithreading kunt gebruiken. Hierdoor zullen de start tijden versneld worden, omdat de UI (presentation) eerder kan worden weergegeven dan de abstraction gedeelte begint.

### Three-tier organization

De three-tier organization wordt meestal gebruikt bij e-commerce/bedrijven. Deze is voor ons nog niet van toepassing omdat het nog te moeilijk voor ons te begrijpen is, maar wij hebben wel hierover nagedacht en opgeschreven wat wij volgens het onderzoek denken te begrijpen.

#### Presentation tier

de presentation tier is de front-end van het framework. Het laat de gebruiker alles zien wat hij doet en opvraagt uit de database.

#### Application tier

Zodra de gebruiker iets op de site doet moet de application tier dit allemaal verwerken. Het voert functies uit, vervoert data, etc. De application tier zorgt voor contact tussen de data en de presentation tier.

#### Data tier

De data tier heeft alle informatie van de database en de bestanden opgeslagen. De data tier verwerkt de acties van de application tier en stuurt de resultaten weer terug. Deze resultaten worden weer verwerkt in de application tier en worden dan weer teruggestuurd naar de presentation tier.

## Framework toepassingen

Een framework heeft verschillende toepassingen.

* Algemene websites
* Forum, wiki en weblogs
* Social media
* Content management systems (CMS)

# Wat gaan wij maken

Een web applicatie framework die ons zal ondersteunen met het realiseren van een portfolio website (blog). De framework zal niet veel grafische elementen hebben, en zal een simpel navigatie systeem hebben. Wij hebben gekozen voor het MVC (Model, View, Controller) model te gebruiken voor ons framework.

## De eisen van het framework

* Menu mogelijkheden
* Inpage mogelijkheden
* Verschillende styles per pagina
* Verschillende templates
* Mogelijkheid gebruiken van modules

# Bronvermelding

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

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

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

<http://www.guyrutenberg.com/2008/04/26/pull-vs-push-mvc-architecture/>

[Hierarchical model–view–controller](http://en.wikipedia.org/wiki/Hierarchical_model%E2%80%93view%E2%80%93controller)

[Model–view–adapter](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93adapter)

[Model–view–presenter](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93presenter)

[Model View ViewModel](http://en.wikipedia.org/wiki/Model_View_ViewModel)

[Observer pattern](http://en.wikipedia.org/wiki/Observer_pattern)

[Presentation–abstraction–control](http://en.wikipedia.org/wiki/Presentation%E2%80%93abstraction%E2%80%93control)

[Three-tier architecture](http://en.wikipedia.org/wiki/Three-tier_architecture)

# Overeenkomst

Bij akkoord graag ondertekenen

Opdrachtgever Opdrachtnemer

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_