Sjabloon Functioneel Ontwerp

Voor de Student

**Naam: Sjabloon FO**

**Versie / Variant : 1**

**Opleiding: Media Developer**

**Crebo : 95313**

**Kerntaak : 1 + 2**

**Werkprocessen : 1.3**

# Versiebeheer:

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Beschrijving | Datum | Aanpassing |
| 1.0 | Initieel document | 03-10-2014 |  |
| 1.1 | Aanpassing | 10-6-2015 | Aanpassing scenario toegevoegd |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Naam: | Sjabloon Functioneel Ontwerp | |
| Kwalificatie en crebocode: | Media Developer | 95313 |
| Vaststellingsdatum: |  |  |

© Marcel Koningstein. Auteursrechten voorbehouden. Niets uit dit werk mag zonder voorafgaande schriftelijke toestemming worden openbaar gemaakt en/of verveelvoudigd door middel van druk, fotokopie/reprografie, scanning, computersoftware, microfilm, geluidskopie, film- of videokopie, of op welke wijze dan ook

# Functioneel Ontwerp

## Voorblad

Onderdelen die op het voorblad moeten staan zijn de volgende:

* De naam van het project
* De naam van het product: Functioneel Ontwerp
* De datum, waarop het product is ingeleverd.
* De naam of namen van degenen die aan het product hebben meegewerkt.
* Het versienummer
* Eventueel een logo

## Inhoudsopgave

In het functioneel ontwerp moet een automatische inhoudsopgave staan.  
Graag erop letten dat alle belangrijke dingen hier wel in komen te staan.

## Inleiding

Geef een korte omschrijving van de opdracht / het project. Waarom maak je het, voor wie, en hoe ga je er mee om.

## Kernbehoefte

De kernbehoefte beschrijft in één zin het doel van het systeem. De kernbehoefte bestaat altijd maar uit één eis. Zodra je geneigd bent om een samengestelde eis te formuleren, ben je nog niet tot de kern van de zaak doorgedrongen.

De kernbehoefte beschrijft op geen enkele manier een oplossing. Alle mogelijkheden worden open gelaten. Het is in deze fase te makkelijk om voorbarige conclusies te trekken en mogelijke oplossingen over het hoofd te zien.

## Gebruikers eisen & wensen

In dit hoofdstuk beschrijf je wat de gebruikers van het systeem verwachten. Dit doe je door de wensen en behoeften in kaart te brengen. Deze wensen en behoeften vormen het uitgangspunt bij het kiezen van een oplossing. Je mag bij het beschrijven van de uitgangssituatie dus niet uitgaan van een bepaalde oplossing; het is niet eens gezegd dat er een computer in het uiteindelijke systeem voorkomt.

De beschrijving van de wensen en behoeften komt tot stand na uitvoerig overleg met de toekomstige gebruikers. Zodra je de eerste versie van dit hoofdstuk geschreven hebt, laat je het aan de gebruikers lezen. Zij bepalen of de wensen en behoeften correct en volledig beschreven zijn. Zo niet, dan pas je de beschrijving aan. Dit doe je totdat de gebruikers tevreden zijn.

Geeft bij elk onderdeel aan wat de prioriteit is van het onderdeel. Hierbij zal je het principe van **M**o**SC**o**W** moeten toepassen:

* M - MUST: deze eis moet in het eindresultaat terugkomen, zonder deze eis is het product niet bruikbaar;
* S - SHOULD: deze eis is zeer gewenst, maar zonder is het product wel bruikbaar;
* C - COULD: deze eis mag alleen aan bod komen als er tijd genoeg is;
* W - WON'T: deze eis zal in dit project niet aan bod komen maar kan in de toekomst, bij een vervolgproject, interessant zijn.

## Scenario’s

Scenario's zijn heel korte beschrijvingen van potentiële praktijksituaties, vanuit een gebruiker. Essentieel is het doel wat de gebruiker heeft. Scenario's worden gemaakt om de functionaliteit van het programma op een informele wijze te inventariseren. Het gaat bij scenario's om de toekomstige, gewenste, situatie: Op deze wijze wordt de variëteit van het toekomstige programma in beeld gebracht. Uiteraard kan hierbij ook de huidige situatie een rol spelen:

Scenario's formuleer je op basis van de huidige situatie (bijvoorbeeld vastgelegd in taakanalyses), de gewenste situatie (omschreven in de opdracht) en gezond verstand. Als er voldoende scenario's zijn worden deze gegroepeerd in use-cases. Ook kunnen dan de globaal verwoorde 'wensen en eisen' in de opdracht worden omgezet in nauwkeurig geformuleerde programmaspecificaties.

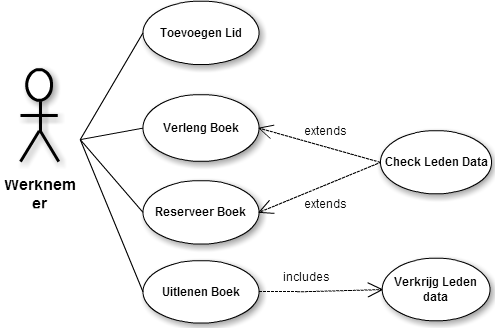
**Voorbeeld:**

* Kees plaatst zijn bankpas in de geldautomaat
* Kees krijgt op het scherm de optie om zijn pincode in te vullen
* Kees vult zijn pincode in
* Kees krijgt op het scherm een melding dat de pincode correct was.
* Kees krijgt op het scherm een keuze hoeveel geld hij wil opnemen. 10, 20, 40, 50, 70 en 100 euro.
* Kees kiest dat hij 50 euro wil hebben
* De geldautomaat geeft de bankpas terug
* De geldautomaat geeft kees 50 euro.
* De geldautomaat geeft kees op het scherm een optie om het bonnetje uit te printen (20 sec)

## Usecase

Use-cases zijn een middel om de functionele eisen die gesteld worden aan het softwaresysteem weer te geven. Use-cases beschrijven hoe de gebruiker met het systeem om wil gaan.

**Voorbeeld:**



## Usecase specificatie

Een use-case kan je geheel specificeren volgens het volgende schema:

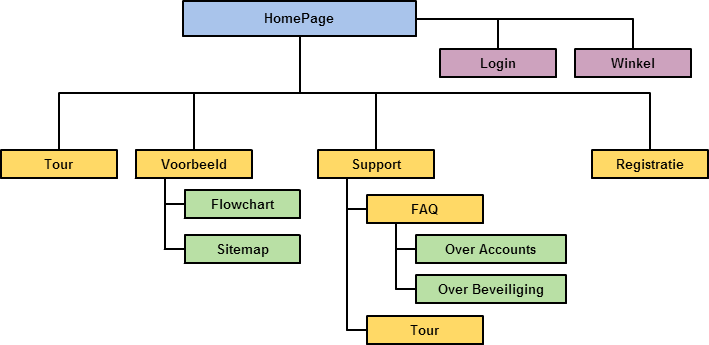
**Voorbeeld:**

|  |  |
| --- | --- |
| Naam | Inloggen |
| Samenvatting | De gebruiker dient zich in te loggen om toegang te krijgen tot het  systeem. |
| Actoren | Rijschoolhouder, Rijinstructeur, Leerling |
| Aannamen (pre-condition) | De gebruiker heeft een bepaalde rol, geldige gebruikersnaam en  wachtwoord |
| Resultaat (post-condition) | Gebruiker is ingelogd |
| Primair pad | 1.Gebruiker selecteer een rol  2.Gebruiker voert zijn/haar gebruikersnaam in  3.Gebruiker voert zijn/haar wachtwoord in  4. Gebruiker klikt op het loginknop  5a. Gebruiker is ingelogd in het systeem |
| Alternatieve paden | 5b.Gebruiker krijgt een foutmelding te zien  6.Gebruiker klikt op wachtwoord vergeten knop  7.Gebruiker voert zijn/haar e-mailadres in  8.Gebruiker klikt op verzendknop  9a.Gebruiker e-mailadres wordt verwerkt  9b. Gebruiker krijgt een foutmelding te zien |
| Relatie met andere use-cases | Uitloggen |

## Indeling

Met behulp van een Sitemap kan je uitleggen welke pagina’s op een website komen, en hoe deze onderling met elkaar verbonden zijn. Leg tevens uit waarom je voor deze indeling hebt gekozen.

**Voorbeeld:**



## Interface

Met behulp van een wireframe kan je het uiterlijk en de indeling van pagina’s vormgeven. Hierin is niet alleen de indeling belangrijk, maar ook hoe bijvoorbeeld tekst op een pagina staat, waar plaatjes, tabellen en formulieren komen. Tevens kan je hier ook al een keuze maken wat voor kleurgebruik op de pagina is, en tevens het lettertype. Zorg ervoor dat je de eigenschappen van kleuren en tekst nog apart noteert.

**Voorbeeld:**

