FUNCTIONEEL ONTWERP

Hotel california

# Inhoudsopgave

[1 Samenvatting 2](#_Toc401260718)

[1.1 Samenvatting voor de klant 2](#_Toc401260719)

[1.2 Versiebeheer 2](#_Toc401260720)

[1.3 Verzendlijst 2](#_Toc401260721)

[2 Plan van aanpak 3](#_Toc401260722)

[2.1 Op te leveren producten 3](#_Toc401260723)

[2.2 Planning 3](#_Toc401260724)

[3 Programma van eisen 4](#_Toc401260725)

[3.1 Doelstelling 4](#_Toc401260726)

[3.2 Huidige situatie 4](#_Toc401260727)

[3.3 Reden van verandering/aanpassing 4](#_Toc401260728)

[3.4 Nieuwe situatie 4](#_Toc401260729)

[3.5 Functionele eisen gesteld aan de nieuwe situatie 4](#_Toc401260730)

[3.5 Systeemeisen gesteld aan de nieuwe situatie 6](#_Toc401260731)

[4 Interface 7](#_Toc401260732)

[4.1 Interface 7](#_Toc401260733)

[4.2 Schema’s 7](#_Toc401260734)

[5 Use Case en Datamodel 8](#_Toc401260735)

[5.1.1 De Use Case 8](#_Toc401260736)

[5.1.2 Verklaring van de Use Case 8](#_Toc401260737)

[5.2.1 De database 8](#_Toc401260738)

[5.2.2 Het ER Model 8](#_Toc401260739)

[5.2.3 Inhoud van de database 8](#_Toc401260740)

[5.2.4 Beheer en beveiliging van de database 8](#_Toc401260741)

[6 Slotconclusie 9](#_Toc401260742)

[6.1 Verantwoording 9](#_Toc401260743)

[6.2 Bronvermelding 9](#_Toc401260744)

# 1 Samenvatting

## 1.1 Samenvatting voor de klant

Dit document omvat de rapportage over de fase functioneel ontwerp van <project>. Alle hoofdonderdelen evenals conclusies en aanbevelingen zijn verwerkt in dit totaaloverzicht,

met eventuele verwijzingen naar separate uitgebreide documenten, die gedurende deze fase

zijn opgesteld.

## 1.2 Versiebeheer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versie** | **Status** | **Datum** | **Auteur** | **Aanpassingen** |
| **X.X** |  |  |  |  |
| **X.X** |  |  |  |  |
| **X.X** |  |  |  |  |
| **X.X** |  |  |  |  |

## 1.3 Verzendlijst

Dit document wordt ter beschikking gesteld aan:

[naam en functie]

# 2. Plan van aanpak

## 2.1 Op te leveren producten

Hierin neem je op welke producten je gaat opleveren. Denk hierbij:

functionele eisen

systeemeisen

datamodel

schema’s

schermen

documenten

Dit zijn de zogenaamde mijlpaalproducten. Deze worden gebruikt om te kijken of je nog op schema ligt

## 2.2 Planning

De mijlpaalproducten in combinatie met de data wanneer ze opgeleverd worden vormen je planning. Uiteindelijk moet je hierin kunnen lezen wanneer je je Functioneel Ontwerp oplevert en hoe je dit doet. Via een presentatie of via een rapport of anders.

# 3 Programma van eisen

## 3.1 Doelstelling

De doelstellingen van het functioneel ontwerp wordt hier omschreven. Deze worden opgesteld in samenspraak met de opdrachtgever Samengevat geef je hier aan wat je gaat ontwikkelen.

## 3.2 Huidige situatie

Geef daartoe eerst de huidige situatie weer en wat daar goed en slecht aan is. Geef ook aan welke mensen en middelen hierbij nu worden ingezet en hoe. Geef ook aan welke gegevensbronnen nu worden gebruikt

## 3.3 Reden van verandering/aanpassing

Het ontwikkelen van een nieuwe applicatie komt nooit uit de lucht vallen. Er is altijd een reden. Hierbij geef je aan waarom er reden is voor verandering. Wat is er goed of fout aan het huidige systeem.

## 3.4 Nieuwe situatie

Hier geef je aan hoe het nieuwe systeem eruit ziet. Welke mensen en middelen gaan worden ingezet en hoe. Waar gaan de gegevens vandaan komen in de nieuwe situatie en voor wie zijn ze bedoeld.

## 3.5 Functionele eisen gesteld aan de nieuwe situatie

Puntsgewijs gedetailleerd opschrijven wat het nieuwe systeem moet kunnen:

Bv. een bibliotheek

* artikelen uitlenen met de volgende voorwaarden:
  + niet meer dan 3 artikelen tegelijkertijd
  + als er een boete openstaat niet
  + als het artikel gereserveerd is door een ander niet
* artikelen terugnemen met de volgende voorwaarde:
  + boete uitrekenen bij te laat terug
* artikelen reserveren
* artikelen toevoegen
* artikelen verwijderen
* artikelen aanpassen
* Lenergegevens toevoegen
* lenergegeven verwijderen
* lenergegevens aanpassen
* overzicht maken van uitgeleende artikelen op in te voeren periode
* overzicht maken van uitgeleende artikelen op huidige datum
* overzicht maken van gereserveerde artikelen op in te voeren periode
* overzicht maken van boeken die te laat zijn
* Briefje maken per lener wat hij/zij moet betalen voor te laat ingeleverde boeken
* Briefje maken voor lener dat gereserveerde boek binnen is.
* Elke week een back-up maken van de database
* Mogelijkheid om gegevens ouder dan een bepaalde datum te kunnen wissen
* etc.

Deze lijst wordt gebruikt door de klant om de applicatie goed te keuren.

Vergeet niet dat beheer als de applicatie draait ook een onderdeel van de functionele eisen is.

Hieronder zie je hoe zo’n acceptatietest formulier er dan uit kan zien (komt na het technisch ontwerp)

|  |  |  |
| --- | --- | --- |
| Eisen | werkt (ja/nee) | verdere acties |
| artikelen uitlenen met de volgende voorwaarden: |  |  |
| * + niet meer dan 3 artikelen tegelijkertijd |  |  |
| * + als er een boete openstaat niet |  |  |
| * + als het artikel gereserveerd is door een ander niet |  |  |
| artikelen terugnemen met de volgende voorwaarde: |  |  |
| * + boete uitrekenen bij te laat terug |  |  |
| artikelen reserveren |  |  |
| artikelen toevoegen |  |  |
| artikelen verwijderen |  |  |
| artikelen aanpassen |  |  |
| Lenergegevens toevoegen |  |  |
| lenergegeven verwijderen |  |  |
| lenergegevens aanpassen |  |  |
| overzicht maken van uitgeleende artikelen op in te voeren periode |  |  |
| overzicht maken van uitgeleende artikelen op huidige datum |  |  |
| overzicht maken van gereserveerde artikelen op in te voeren periode |  |  |
| overzicht maken van boeken die te laat zijn |  |  |
| Briefje maken per lener wat hij/zij moet betalen voor te laat ingeleverde boeken |  |  |
| Briefje maken voor lener dat gereserveerde boek binnen is. |  |  |
| * Elke week een back-up maken van de database |  |  |
| * Mogelijkheid om gegevens ouder dan een bepaalde datum te kunnen wissen |  |  |
| etc. |  |  |

## 3.5 Systeemeisen gesteld aan de nieuwe situatie

Hier geef je aan welke eisen worden gesteld aan de hard- en software van de klant. Het gaat hierbij om een globale indruk. Als het echt technisch wordt dan hoort het thuis in het technisch ontwerp. Let op de volgende zaken:

* Standalone of netwerk en de daarbij behorende eisen
* Internet of lokaal en de daarbij behorende eisen
* Eisen aan de hardware (processor, geheugen, WIFI, netwerk, harddisk, grafisch)
* Eisen aan de software (besturingssysteem, office, )
* lever je een run-time versie

# 4 Interface

Hierin beschrijf je de interface tussen programma en gebruiker.

## 4.1 Interface

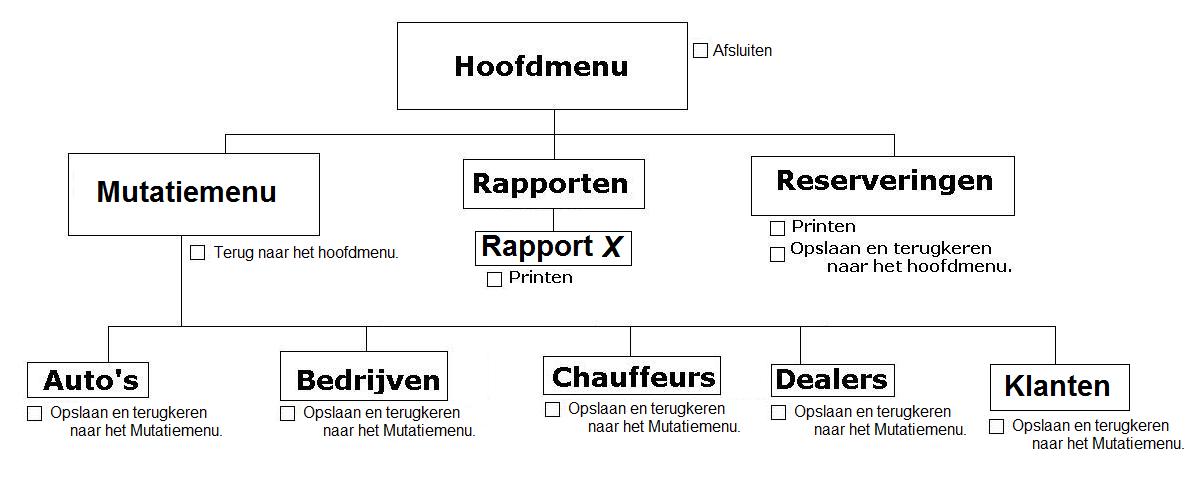
Alle schermen, inclusief vorm en lay-out van rapporten, formulieren, pop-up windows, foutboodschappen, meldingen, knoppen, menu’s, submenu’s, etc.

Je geeft aan hoe het eruit ziet en wat het doet (vanuit het standpunt van de gebruiker)

## 4.2 Schema’s

Hierin beschrijf je in welke volgorde en wanneer de verschillende schermen worden aangeroepen. Van belang is dat de klant ziet wanneer welk scherm komt en wat de mogelijkheden in zo’n scherm zijn.

Hieronder zie je een voorbeeld:



Als je via het OO model gaat werken komen hier de diagrammen die bij het OO model gebruikt worden, zoals use-case diagram en klassendiagram.

**Ook andere schema’s en diagrammen die inzicht geven in de werking van de applicatie kun je hier opnemen of in hoofdstuk 5 of in het technisch ontwerp.**

# 5. De use-case en datamodellen

## 5.1.1 De USE Case

De use case beschrijft "wie" met het betreffende systeem "wat" kan doen

Een use case beschrijft een systeem vanuit het gebruikersperspectief. Het beschrijft de actor, de initiator van de interactie, en het systeem zelf als een opeenvolging van eenvoudige stappen. Actoren kunnen iets of iemand zijn, die bestaat buiten het te bestuderen systeem, en die deelneemt in de opeenvolgende activiteiten in een dialoog met het systeem om een bepaald doel te bereiken. Actoren kunnen eindgebruikers, andere systemen of hardware (apparatuur) zijn. Elke use case is een complete serie van zogenaamde "events", beschreven vanuit het standpunt van de actor.

### 5.1.2 Verklaring van de Use Case

USE Cases zijn eenvoudigweg een beschrijving van een reeks gebeurtenissen, die met elkaar het systeem in staat stellen iets bruikbaars te doen.[2] Elke use case beschrijft, hoe de actor zich zal gedragen ten opzichte van het systeem om een doel te bereiken.

# 5.2.1 De database

In de meeste gevallen wordt een database gebruikt om gegevens op te halen of op te slaan.

Het is van belang hoe deze database eruit ziet.

Daarom moet je vanuit de functionele kant alles beschrijven.

* Hoe zien de tabellen eruit
* Welke velden bevat een tabel en waarom
* welke typering heeft een veld en waarom
* Zijn er sleutels gedefinieerd (m.a.w. is een veld uniek)
* Is er een format op het veld gelegd, bv. een postcode

<<code en specifieke formats horen bij het technisch ontwerp, tenzij de klant dat moet weten>>

### 5.2.2 Het ER Model

Welke relatie is er tussen de tabellen. Geef dit weer in een schema. Wat betekent deze relatie voor het muteren van gegevens. Waarom maak je deze keuzes.

Zorg ervoor dat de database genormaliseerd is.

### 5.2.3 Inhoud van de database

Ook de grootte is van belang voor de applicatie. Hoe groter de database, hoe sneller het systeem moet zijn i.v.m. performance. Ook de hoeveelheid mutaties die verwacht gaan worden zijn van belang evenals de mogelijkheid tot het maken van back-up.

### 5.2.4 Beheer en beveiliging van de database

Beschrijf hier op welke manier hoe de klant het beheer wenst en hoe de beveiliging er uit moet zien. Wie mag wat?

# 6 Slotconclusie

## 6.1 Verantwoording

* Hier beschrijf je tegen welke problemen je bent opgelopen.
* Welke problemen heb je wel aangepakt, welke niet en waarom
* Welke consequenties gaat de nieuwe situatie veroorzaken en hoe moet daar mee worden omgegaan.

## 6.2 Bronvermelding

Documentatie: welke bronnen heb je gebruikt voor je functioneel ontwerp