Versie 1.0

Ontwerpdocument

Rowan Dings

# Documentgeschiedenis

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Versie** | **Belangrijkste wijzigingen** | **Auteur** |
| 1-12-2015 | 0.1 | Opstellen template | Rowan |
| 2-12-2015 | 1.0 | Invullen data | Rowan |
|  |  |  |  |
|  |  |  |  |

# Referenties

|  |  |  |
| --- | --- | --- |
| **Naam** | **Datum** | **Locatie** |
| Acceptatie Testplan | 2 – 12 - 2015 | Bijlage |
| Analysedocument | 2 – 12 – 2015 | Bijlage |

Inhoudsopgave

[Documentgeschiedenis 2](#_Toc436833892)

[Referenties 2](#_Toc436833893)

[Inleiding 4](#_Toc436833894)

[Productaanleiding 4](#_Toc436833895)

[Architectuur 5](#_Toc436833896)

[ERD en Databaseontwerp 7](#_Toc436833897)

[Praktisch onderdeel 7](#_Toc436833898)

[Casebeschrijving 7](#_Toc436833899)

[ERD 7](#_Toc436833900)

[Theoretisch onderdeel 8](#_Toc436833901)

[Databaseontwerp 8](#_Toc436833902)

[Klassendiagrammen 9](#_Toc436833903)

[Eerste design 9](#_Toc436833904)

[Tweede design 9](#_Toc436833905)

[User interface 10](#_Toc436833906)

Inleiding

# Productaanleiding

De opdrachtomschrijving beschrijft het volgende:

“*Ieder jaar komt bij het bezorgen van de cadeaus door Sinterklaas toch wel weer goed, maar voor zijn  
gevoel moet het toch mogelijk zijn om "het heerlijk avondje" wat vlotter te laten verlopen. Hij zou daarom  
graag een PietenPlanner willen hebben.  
Met deze planner heeft de Sint voor ogen dat hij, gegeven een gemeente in Nederland, op eenvoudige  
wijze de benodigde Pieten kan bepalen. Hij heeft natuurlijk de beschikking over de gegevens van alle  
kinderen en gemeenten in Nederland. Wat hij hieraan wil koppelen is een categorie van cadeaus welke  
van toepassing is op een gemeente. Op basis van deze gegevens moet het mogelijk zijn om een  
zogenoemde "Pieten Posse" samen te stellen. Deze gaan dan, al dan niet vergezeld door de WegwijsPiet, op stap*.”

Dit houdt dus in dat voor het makkelijker laten verlopen van de sinterklaasavond, er een planner moet worden gemaakt, die per gemeente gemakkelijk kan aangeven welke pieten er nodig zijn om aan alle wensen te voldoen.

In het ontwerpdocument wordt toegelicht hoe de applicatie intern eruit zal zien, zowel als alle verbindingen en hoe functionaliteit wordt toegepast.

Architectuur

De architectuur van de applicatie ziet er als volgt uit:



Dit met oog op het vervangen van het GUI. Voor nu zal er een prototype worden gemaakt die basaal alles kan, terwijl bij het eindresultaat er mogelijk een compleet andere GUI van toepassing zal moeten zijn. Een klant bepaald immers wat hij of zij wilt zien in een applicatie.

Ook zullen de basale klassen in de Class Library statische methodes hebben om hunzelf in te vullen en te retourneren.

Het komt daarbij dus op neer dat de administratieklasse alleen voor het volgende wordt gebruikt:

* Veiligheidschecks
* Berekenen van data
* Doorsluisen van data naar de GUI

De GUI alleen voor

* Het weergeven van informatie
* Het teruggeven van meldingen
* Het doorgeven van informatie naar de administratieklasse

De onderliggende klassen voor

* Het opvragen van informatie van de database (.Get())
* Het aanpassen van informatie in de database (.Update(obj o))
* Het aanmaken van nieuwe entiteiten van eigen klassen (daarbij ook meteen in de database)
* Het valideren van bestaande objecten in de database (.Validate(obj o))

En de database als laatste voor

* Het beveiligen van queries
* Specifieke queries
* Basale queries zoals update, insert, delete en select

Of in tabel-vorm:

|  |  |  |
| --- | --- | --- |
| Klasse | Beschrijving functie | Functie |
| GUI | Doorsluisluik voor informatie, input | Luik |
| Administrator | Doorsluisluik voor informatie, zorgt voor verificatie | Security |
| Basaal | Vragen informatie op en passen informatie aan | Informatie |
| Database | Haalt informatie op en zorgt voor verificatie | Security & Informatie |

ERD en Databaseontwerp

# Praktisch onderdeel

## Casebeschrijving

Voor het gemak is het scenario uit het analysedocument genomen. Hieruit zijn de entiteiten aangegeven die nodig zijn voor het opzetten van een ERD en databasemodel.

Legenda:

**Tabel**

Kolom

*“Sinterklaas wilt graag weten welke* ***pieten*** *hij dit jaar moet versturen naar de* ***gemeente*** *Eindhoven. Hij weet voor de gemeente dat deze ongeveer 30.000 kinderen bevat en heeft voor Eindhoven en zowel andere gemeentes de benodigde informatie* (lees: cadeautypes, x, y, provincie) *al in een bestand gezet. Hij opent deze in de applicatie, die alles inlaadt, selecteert Eindhoven en bekijkt nogmaals naar de informatie of hij nog iets wilt toevoegen, aanpassen en of alles in orde is. Hij stelt aan zichzelf dat dit het geval is, en maakt een* ***PietenPosse*** *aan. Deze geeft automatisch aan welke pieten er nodig zijn en dat het aantal verdubbelt is. Er zijn immers meer dan 10.000 kinderen in deze gemeente.*

*Sinterklaas slaat dit op en sluit daarna tevreden de applicatie af, om de pieten de informeren over de posse die nodig is.”*

# ERD

ERD als opgevat in de wereld van de klant:



# Theoretisch onderdeel

Vanuit het ERD is op te stellen dat er de volgende tabellen nodig zijn:

**LEGENDA**: Primary key, *<Foreign Key>*, Vereist, Optioneel.

Gemeente (ID, x, y, provincie, kinderen)

Cadeautype (Naam, <*Gemeente\_ID*>)

Piet (ID, naam)

Vaardigheid (Naam, *<Piet\_ID>, <Gemeente\_ID>*)

# Databaseontwerp



Klassendiagrammen

## Eerste design



Indien geen constructor is opgegeven, dan wordt er gebruik gemaakt van de volledige constructor. De database heeft enkel alleen statische methodes.

De Piet-klasse heeft een lijst van alle Vaardigheden die bij deze piet horen, en de Gemeente-klasse heeft een lijst van alle CadeauTypes die bij deze gemeente horen.

Belangrijk is dat in dit klassendiagram nog geen rekening is gehouden met statische methodes of beveiliging aan de database kant. Voor het eerste prototype wordt dit dan ook alleen als grond bijgehouden; er kunnen veel veranderingen nog aan optreden, maar in het grootste gedeelte blijft het ongeveer zoals dit.

## Tweede design

User interface

Hier komen alle GUI’s te staan met uitleg over de user interface;

* Hoe tot dit scherm gekomen wordt
* Wat er op het scherm kan worden gedaan
* Mogelijke gemaakte overwegingen