**Analyse Document**

Project: Mijn Project

Project team: Mijn Project Team

Teamleden: Student 1 (studentnummer)

Student 2 (studentnummer)

Student 3 (studentnummer)

Opdrachtgever: Mijn opdrachtgever

Versie: 0.2

Versiedatum: 17 juni 2018

Status: Concept

# Documenthistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Wijzigingen | Auteur | Datum |
| 0.1 | Opzet document | Nico Kuijpers | 11 dec 2017 |
| 0.2 | Beschrijving domeinmodel aangepast | Nico Kuijpers | 17 jun 2018 |

Inhoud

Documenthistorie 2

Termenlijst 4

H1 Inleiding 5

H2 Acties 6

H3 Regels 7

H4 Kwaliteitsattributen 8

H5 Globale schermopbouw 9

H6 Use cases 10

H7 Domeinmodel 11

# Termenlijst

*In dit template voor het analyse document worden aanwijzingen gegeven in italic (schuinschrift). Voor de uiteindelijke versie moeten de teksten in italic worden verwijderd.*

*In de termenlijst worden alleen termen opgenomen die betrekking hebben op het domein van de applicatie. Termen die gerelateerd zijn aan software engingeering in het algemeen (zoals use case diagram, use case, actor, kwaliteitsattribuut, unittest, etc.) worden NIET opgenomen in de termenlijst.*

|  |  |
| --- | --- |
| **Term** | **Definitie** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# H1 Inleiding

*Beschrijf in de inleiding:*

* *De context van het project: geef een beschrijving van het probleem, beschrijf wie de opdrachtgever is en beschrijf wie de stakeholders zijn.*
* *De (gedistribueerde) applicatie die je gaat maken.*
* *Het doel van dit document.*

# H2 Acties

*In dit hoofdstuk worden de acties beschreven. Acties vallen onder de functionele eisen. Beschrijf steeds 1 actie per requirement. Geef het belang aan de hand van MoSCoW (Must, Should, Could, Won’t have) en de urgentie met Hoog, Midden, Laag. Verwijs naar de betreffende use case door middel van een Use Case ID.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Naam** | **Omschrijving** | **Belang** | **Urgentie** | **Use Case ID** |
| F1 |  |  | Must | Hoog | UC1 |
| F2 |  |  | ….. | ….. | ….. |
| …. |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# H3 Regels

*In dit hoofdstuk worden de regels beschreven waar het systeem aan moet voldoen. Regels vallen onder de categorie functionele eisen. Regels kunnen bijvoorbeeld spelregels zijn, maar ook een minimum aantal karakters voor een wachtwoord. Een ander voorbeeld van een regel is dat een spel kan starten nadat twee spelers zich hebben aangemeld. Geef het belang aan de hand van MoSCoW (Must, Should, Could, Won’t have) en de urgentie met Hoog, Midden, Laag.*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Omschrijving** | **Belang** | **Urgentie** |
| R1 | Een wachtwoord bestaat uit minimaal …. karakters. | Must | Hoog |
| R2 | …. |  |  |
| R3 | …. |  |  |
| …. |  |  |  |

# H4 Kwaliteitsattributen

*In dit hoofdstuk worden de kwaliteitsattributen beschreven waar het systeem aan moet voldoen. Kwaliteitsattributen vallen onder de niet-functionele eisen. Enkele voorbeelden worden hieronder gegeven. Voor een opdeling in categorieën, zie*

*http://iso25000.com/index.php/en/iso-25000-standards/iso-25010*

*of https://nl.wikipedia.org/wiki/ISO\_25010*

|  |  |  |
| --- | --- | --- |
| **ID** | **Categorie ISO 25010** | **Omschrijving** |
| Q1 | Performance | De responsetijd bedraagt maximaal …… |
| Q2 | Performance | …. |
| Q3 | Performance | …. |
| Q4 | Compatibiliteit | Documentatie van interfaces is beschikbaar om …. |
| Q5 | Bruikbaarheid | Het systeem kan door 95% van de doelgroep binnen …. minuten worden gebruikt. |
| Q6 | Betrouwbaarheid | Het systeem is tenminste 99% van de tijd beschikbaar. |
| Q7 | Betrouwbaarheid | Het systeem is maximaal 1 uur per 24 uur niet beschikbaar. |
| Q8 | Beveiligbaarheid | Gegevens worden verstuurd over een beveiligde verbinding. |
| Q9 | Beveiligbaarheid | Wachtwoorden worden versleuteld opgeslagen. |
| Q10 | Onderhoudbaarheid | Het systeem is modulair opgebouwd. |
| Q11 | Onderhoudbaarheid | Source code is geschreven volgens de …. coderingsstandaard. |
| Q12 | Onderhoudbaarheid | Source code is gedocumenteerd. |
| Q13 | Onderhoudbaarheid | Tenminste …. % van de source code wordt getest met unittests. |
|  |  |  |

# H5 Globale schermopbouw

*In dit hoofdstuk wordt de globale schermopbouw getoond.*

# H6 Use cases

*In dit hoofdstuk wordt een use case diagram getoond en worden de use cases beschreven. Gebruik visual paradigm om een use case diagram te maken. Hieronder staat een voorbeeld van een use case.*

|  |  |
| --- | --- |
| **Naam** | **UC1**: Gegevens aanpassen |
| **Samenvatting** | Een klant kan zijn gegevens aanpassen. |
| **Actoren** | Klant |
| **Aannamen** | 1. De actor is ingelogd.  2. De actor is op zijn persoonlijke pagina. |
| **Beschrijving** | 1. De actor wijzigt zijn gegevens.  2. Het systeem controleert of gegevens uniek zijn. [1]  3. De actor klikt op gegevens wijzigen. [2]  4. Het systeem wijzigt de gegevens. |
| **Uitzondering** | [1] De gegevens zijn al bekend in het systeem: toon een melding aan de gebruiker.  [2] De actor breekt het wijzigen vroegtijdig af: ga terug naar de persoonlijke pagina. |
| **Resultaat** | De gegevens van de klant zijn gewijzigd. |

# H7 Domeinmodel

*In dit hoofdstuk wordt het klassendiagram van het domein getoond en een toelichting gegeven. Laat alleen klassen en relaties zien die een rol spelen in het domein. Geef iedere klasse een naam, attributen en methoden. Relaties tussen klassen worden aangegeven dmv associaties met mulitpliciteiten. Iedere associatie heeft een naam en/of rolnamen. Implementatiegerelateerde details zoals interfaces, overerving, façade-klassen, GUI-klassen, etc. worden achterwege gelaten.*