**Analyse Casus Zeeslag**

Project: Project zeeslag

Project team: Zeeslag Team

Teamleden: Martin Frenken (studentnummer)

Pepijn Pinckers (studentnummer)

Opdrachtgever: Fontys

Versie: 0.3

Versiedatum: 6 Februari 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 |
| 0.3 |  |  |  |
|  |  |  |  |
|  |  |  |  |

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.*

Fontys heeft ons team de opdracht gegeven om het beroemde spel Zeeslag digitaal in vorm van een gedistribueerde app na te maken. In dit document volgt er een analyse over deze opdracht, zodat wij als team een beter idee krijgen wat wij moeten gaan maken.

# 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 | Schip plaatsen | Een speler kan een schip plaatsen. | Must | Hoog | UC1 |
| F2 | Schot afvuren | Een speler kan een schot afvuren. | Must | Hoog | UC2 |
| F3 | Schip verplaatsen | Een speler kan een schip verplaatsen. | Must | Hoog | UC3 |
| F4 | Schip verwijderen | Een speler kan een schip vewijderen | Should | Midden | UC4 |
| F5 | Aangeven klaar met plaatsen | Een speler kan aangeven dat hij klaar is met het plaatsen van zijn schepen | Should | Midden | UC5 |
| F6 | Naam invoeren | Een speler kan zijn naam invoeren | Should | Laag | UC6 |
|  |  |  |  |  |  |

# 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 speler heeft een unieke naam | Should | Laag |
| R2 | Er zijn twee spelers | Must | Midden |
| R3 | Er zijn twee speelvelden | Must | Midden |
| R4 | Een speelveld bestaat uit 10 bij 10 vakjes | Must | Midden |
| R5 | Het speelveld met de eigen schepen wordt het ‘oceaanraster’ genoemd. | Should | Laag |
| R6 | Het speelveld met de schepen van de tegenstander wordt het ‘doelraster’ genoemd. | Shoud | Laag |
| R7 | De eigen schepen zijn zichtbaar in het ‘oceaanraster’ | Should | Hoog |
| R8 | De schepen van de tegenstander zijn niet zichtbaar | Should | Hoog |
| R9 | Een vloot bestaat uit 5 schepen: 1 vliegdekschip, 1 slagschip, 1 kruiser, 1 onderzeeër en 1 mijnenveger | Should | Midden |
| R10 | Een vliegdekschip is 5 vakjes lang en 1 vakje breed | Should | Midden |
| R11 | Een slagschip is 4 vakjes lang en 1 vakje breed. | Should | Midden |
| R12 | Een kruiser is 3 vakjes lang en 1 vakje breed. | Should | Midden |
| R13 | Een onderzeeër is 3 vakjes lang en 1 vakje breed. | Should | Midden |
| R14 | Een mijnenveger is 2 vakjes lang en 1 vakje breed. | Should | Midden |
| R15 | Als een schip geplaatst of verplaatst wordt mag er geen overlap zijn met een ander schip dat al geplaatst is. | Should | Hoog |
| R16 | Het spel begint nadat beide spelers hebben aangeven dat alle schepen geplaatst zijn. | Should | Hoog |
| R17 | Een speler mag zijn schepen niet verplaatsen tijdens het spel. | Must | Hoog |
| R18 | Indien een schip van de tegenstander wordt geraakt, wordt het betreffende vakje met rood gemarkeerd in het ‘doelraster’. | Must | Hoog |
| R19 | Indien een eigen schip wordt geraakt wordt het betreffende vakje met rood gemarkeerd in het ‘oceaanraster’. | Must | Hoog |
| R20 | Indien een schot geen schip van de tegenstander raakt wordt het betreffende vakje met blauw gemarkeerd in het ‘doelraster’. | Must | Hoog |
| R21 | Een schip is tot zinken gebracht indien alle vakjes van het betreffende schip zijn geraakt. | Must | Hoog |
| R22 | Een speler heeft gewonnen indien alle schepen van de tegenstander tot zinken zijn gebracht. | Must | Hoog |
| R23 | Indien een schip tot zinken wordt gebracht, verschijnt bij beide spelers een mededeling dat een schip van de betreffende speler tot zinken is gebracht. | Should | Hoog |
| R24 | Indien een speler heeft gewonnen verschijnt bij beide spelers een mededeling welke speler heeft gewonnen. | Should | Midden |

# 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*

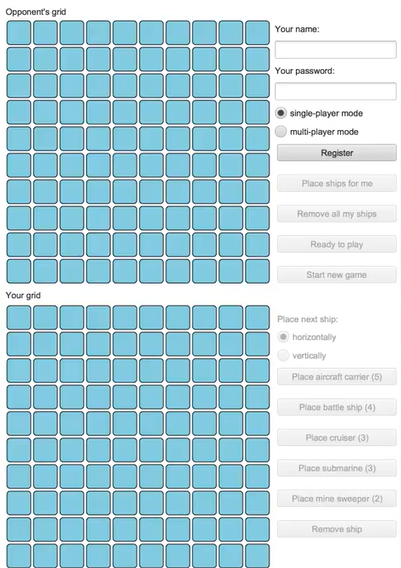
*http://iso25000.com/index.php/en/iso-25000-standards/iso-25010*

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

|  |  |  |
| --- | --- | --- |
| **ID** | **Categorie ISO 25010** | **Omschrijving** |
| Q1 | Performance | De responsetijd bedraagt maximaal 500 ms |
| Q2 | Performance | Het duurt niet langer dan 10 seconden om de applicatie op te starten |
| Q3 | Performance | Het duurt niet langer dan 10 seconden om een spel/lobby te starten |
| Q4 | Compatibiliteit | Documentatie van interfaces is beschikbaar drie dagen voor de inleverdatum |
| Q5 | Bruikbaarheid | Het systeem kan door 95% van de doelgroep binnen 5 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 java coderingsstandaard. |
| Q12 | Onderhoudbaarheid | Source code is gedocumenteerd. |
| Q13 | Onderhoudbaarheid | Tenminste 20 % 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**: Schip plaatsen |
| **Samenvatting** | De actor kan een schip plaatsen |
| **Actoren** | Speler |
| **Aannamen** | 1. De actor heeft een spel begonnen  2. Er kunnen nog geen schepen geraakt worden |
| **Beschrijving** | 1. De actor kiest een schip  2. De actor selecteerd een positie  3. Het systeem plaatst het schip op de aangegeven positie. [1]  4. Het systeem slaat de positie van het schip op. |
| **Uitzondering** | [1] De aangegeven positie zit niet op het oceaanraster. |
| **Resultaat** | Er is een schip geplaatst op het oceaanraster. |

|  |  |
| --- | --- |
| **Naam** | **UC2**: Schot afvuren |
| **Samenvatting** | De actor kan een schot afvuren |
| **Actoren** | Speler |
| **Aannamen** | 1. De actor is aan de beurt |
| **Beschrijving** | 1. De actor kiest een positie  2. De actor vuurt een schot af  3. Het systeem detecteerd of de actor een schip geraakt heeft[1]  4. Het systeem ge. |
| **Uitzondering** | [1] De aangegeven positie zit niet op het oceaanraster. |
| **Resultaat** | Er is een schip geplaatst op het oceaanraster. |

# 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.*