

Analysedocument

Live performance

Michelle Broens | S21M | 13/06/2017

Inhoudsopgave

[Inleiding 2](#_Toc472263879)

[Begrippen 3](#_Toc472263880)

[Opdrachtgever 4](#_Toc472263881)

[Scenario 5](#_Toc472263882)

[Functionele eisen 6](#_Toc472263883)

[Niet-functionele eisen 7](#_Toc472263884)

[Use case 8](#_Toc472263885)

[Use case diagram 8](#_Toc472263886)

[Use case beschrijvingen 9](#_Toc472263887)

[User interface 10](#_Toc472263888)

[EER model 11](#_Toc472263889)

[EER bijlage 12](#_Toc472263890)

[Productdecompositie 13](#_Toc472263891)

[Planning 14](#_Toc472263892)

# Inleiding

In dit document komen er een aantal verschillende onderwerpen aanbod die te maken hebben met de analysefasen van het maken van een applicatie over … De onderwerpen die aan bod komen zijn:

* Een lijst met begrippen: in deze lijst staan woorden die de lezer misschien niet begrijpt of misschien niet helemaal begrijpt.
* Opdrachtgever: in dit stuk kom een beschrijving over wie de opdrachtgever is en wat de opdrachtgever algemeen verwacht.
* Het scenario: een korte beschrijving over een persoon die door de ontwikkelde applicatie loopt.
* De functionele eisen: de eisen die het programma moet kunnen, bijvoorbeeld een gebruiker kan inloggen.
* De niet-functionele eisen: de eisen die niet te maken hebben met het programmeren van code zoals de functionele eisen, bijvoorbeeld de code moet herbruikbaar zijn.
* De use cases: de functionele eisen worden in een diagram gezet zodat het duidelijk wordt door wie deze functie uitgevoerd wordt. Deze functies worden in de use case beschrijving van start tot eind doorgelopen.
* De userinterface: hier wordt een schets gemaakt van hoe de applicatie er ongeveer uit gaat zien.
* Het EER model: een schets van hoe de structuur van het programma er in de database uit gaat zien.
* De productdecompositie: een schets van de verschillende onderdelen die opgeleverd worden.
* Planning: de planning van hoe dit project uitgewerkt wordt. Per onderdeel staat een tijd van wanneer er een onderdeel begonnen wordt en wanneer deze af is.

# Begrippen

|  |  |
| --- | --- |
| **Begrip** | **Uitleg** |
| Use case | In software engineering is een use case een lijst van acties die een persoon (actor) en het systeem uitvoert om een doel te bereiken, bijvoorbeeld het inloggen. |
| Userinterface | Hier praten de gebruiker en het systeem met elkaar. |
| Database | Een georganiseerde collectie van data in tabellen. |
| AFK | Een afkorting voor het woord afkorting. |
| MoSCoW | Een methode om aan te geven hoe belangrijk een requirement is, met de M voor must(moet), S voor shoulds(zou moeten), C voor could(zou kunnen) en W voor will not(wordt niet geïmplmenteerd). |
|  |  |
|  |  |

# Opdrachtgever

De opdrachtgever van dit project is …

# Scenario

De gebruiker logt in …

# Functionele eisen

Dit hoofdstuk gaat over de functionele requirements die geïmplementeerd gaan worden in de applicatie. Deze worden beschreven met een korte beschrijving, aangeduid hoe belangrijk deze requirement is met de MoSCoW analyse en een eventuele opmerking.

|  |  |  |  |
| --- | --- | --- | --- |
| **AFK** | **Requirement** | **MoSCoW** | **Opmerking** |
| **BS1** | **Gegevens van de huurder die een boot wil huren zijn naam en e-mailadres. Deze dienen in een database te worden opgeslagen.** | **M** |  |
| **BS2** | **Elke boot heeft een (unieke) naam. Een motorboot heeft daarnaast een tankinhoud in liters brandstof. Van een motorboot kan de actieradius (in kilometer) worden opgevraagd. De actieradius van een motorboot is gelijk aan haar tankinhoud in liters vermenigvuldigd met 15.** | **M** |  |
| BS3 | Gegevens van een huurcontractzijn een boot, een lijst van bijkomende artikelen, een verhuurder en datum vanaf en datum tot. Alle gegevens van een huurcontract dienen in de database te worden opgeslagen. | M |  |
| BS4 | De gebruiker kan met het systeem nieuwe huurcontracten aanmaken en hij kan een lijst van alle huurcontracten op het scherm bekijken | M |  |
| BS5 | Men kan in de userinterface zijn budget (een bedrag in euro dat men aan de bootvakantie - huur - wil besteden) invullen en of de huurder op de Noordzee wil varen, en of de huurder op het IJsselmeer wil varen. Vervolgens bepaalt het systeem hoeveel Friese meren de gebruiker mag bevaren. Houd hiermee rekening met de huurprijs, te betalen sluisgeld, enz. Resumerend: input voor de berekening is het budget (een bedrag in euro) alsmede de gehuurde boten plus de eventueel gehuurde bijkomende artikelen en de output is het aantal Friese meren dat je kunt bevaren met dat budget. | M |  |
| **BS6** | **Elk huurcontract met alle gegevens dient te kunnen worden geëxporteerd naar een tekstbestand dat met een eenvoudige teksteditor zoals Windows Kladblok of OS X TextEdit te lezen is door een mens** | **M** |  |

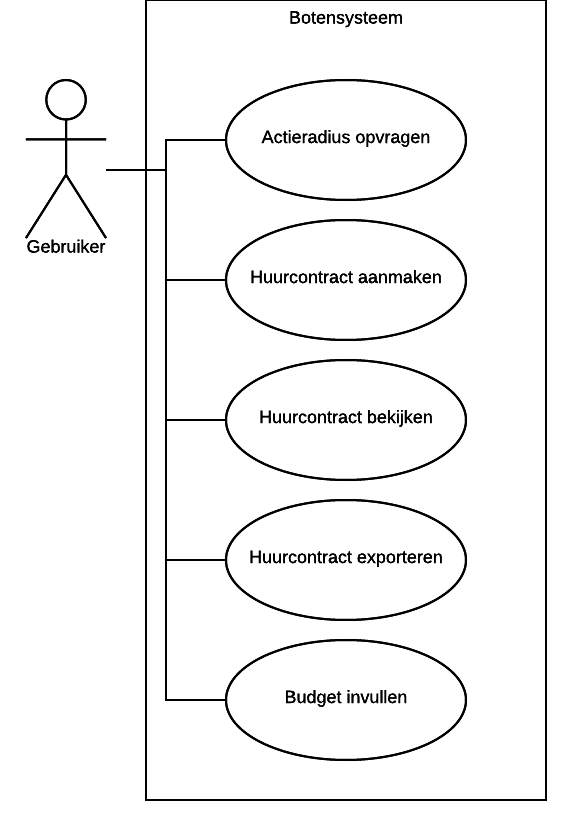
# Niet-functionele eisen

|  |  |  |  |
| --- | --- | --- | --- |
| **AFK** | **Requirement** | **MoSCoW** | **Opmerking** |
| NFR1 | Het ontwerp en de C#-klassen in de code moeten zijn voorbereid op de toekomst. Dit betekent dat er later door de programmeur op een handige manier meerdere soorten spierkrachtaangedreven boten (bijv. een surfplank) en motorbotentypen (bijv. een waterscooter) moeten kunnen worden toegevoegd. Maak daarom in je ontwerp al duidelijke Live performance vj2016 2 keuzen op het gebied van klassen, interfaces en OO-technieken als abstracte klassen, oververving, enz. | M |  |
| NFR2 | Ook wil men in de toekomst naast de Friese meren, IJsselmeer en Noordzee vermoedelijk ook de Waddenzee als vaargebied op een eenvoudige wijze kunnen toevoegen. | M |  |

# Use case

In dit hoofdstuk komt het use case diagram en de beschrijvingen die bij het diagram horen.

## Use case diagram



## Use case beschrijvingen

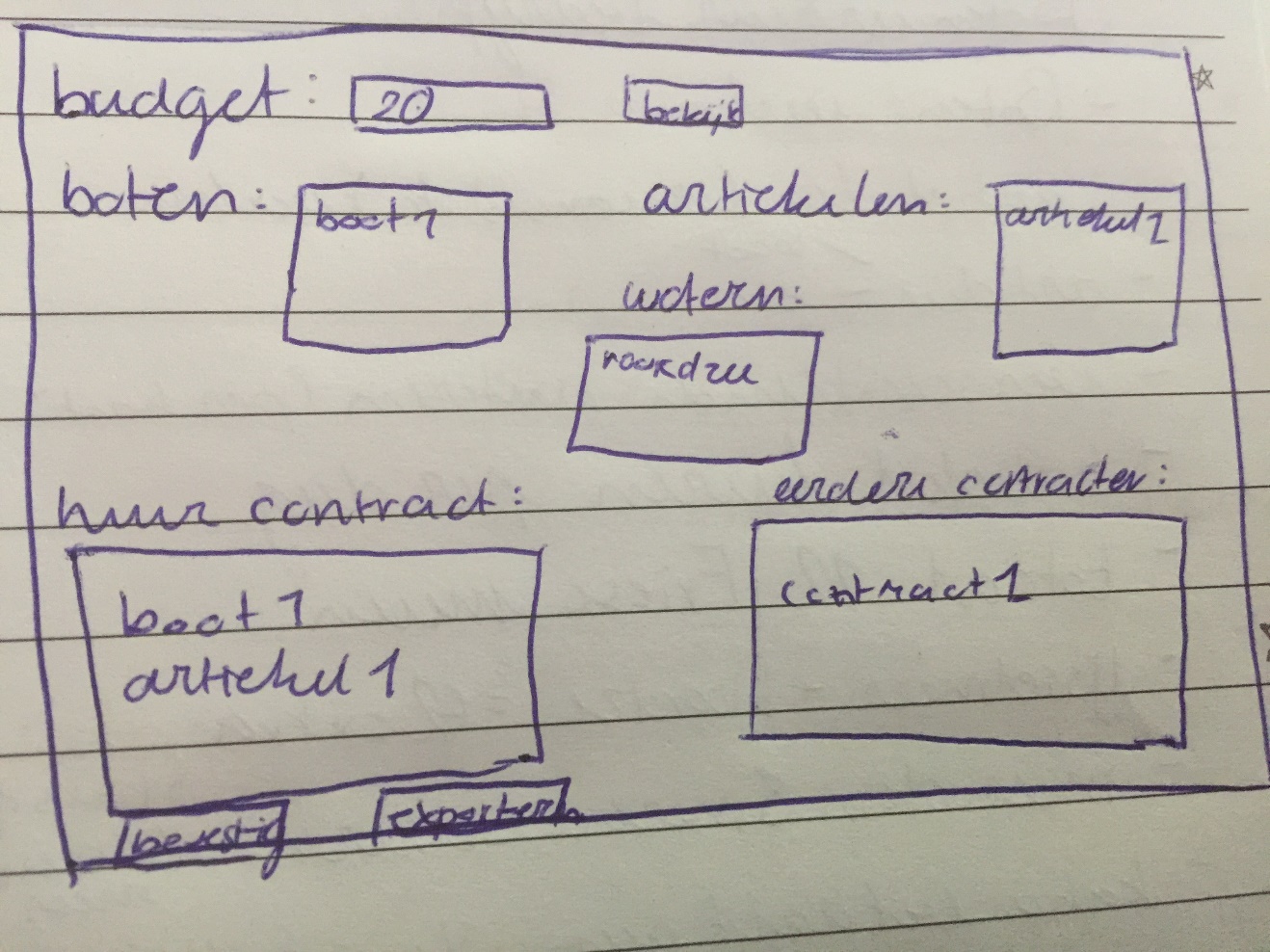
|  |  |
| --- | --- |
| **Naam** |  |
| **Samenvatting** |  |
| **Actor** |  |
| **Aanname** |  |
| **Beschrijvingen** |  |
| **Uitzondering** | [1] |
| **Resultaat** |  |

|  |  |
| --- | --- |
| **Naam** |  |
| **Samenvatting** |  |
| **Actor** |  |
| **Aanname** |  |
| **Beschrijvingen** |  |
| **Uitzondering** | [1] |
| **Resultaat** |  |

|  |  |
| --- | --- |
| **Naam** |  |
| **Samenvatting** |  |
| **Actor** |  |
| **Aanname** |  |
| **Beschrijvingen** |  |
| **Uitzondering** | [1] |
| **Resultaat** |  |

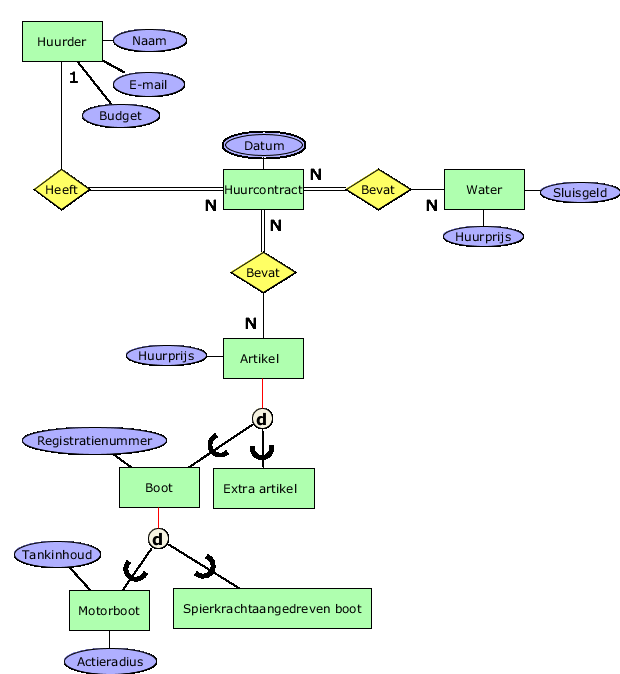
# User interface

In dit hoofdstuk komen een paar designs voor hoe de applicatie er van de buitenkant uit komt te zien.



# EER model

In dit hoofdstuk komt het EER model samen met een bijlage om het model verder uit te leggen.



## EER bijlage

Ik heb ervoor gekozen om een Registratienummer toe te voegen aan boot, deze is er om aan te geven dat een boot een registratienummer heeft en een artikel niet.

# Productdecompositie

Dit hoofdstuk geeft je een beeld in welke delen het project opgedeeld wordt, bijvoorbeeld wat er allemaal opgeleverd moet worden.