## 

|  |
| --- |
| Fontys |
| Ontwerpdocument |
| Ontwerpdocument |

|  |
| --- |
| Sander Koch S21M  20-1-2016 |

## 

Inhoud

[Inleiding 3](#_Toc441066121)

[ERD 4](#_Toc441066122)

[Relationeel database ontwerp 5](#_Toc441066123)

[Datamodel 5](#_Toc441066124)

[Klassendiagram 7](#_Toc441066125)

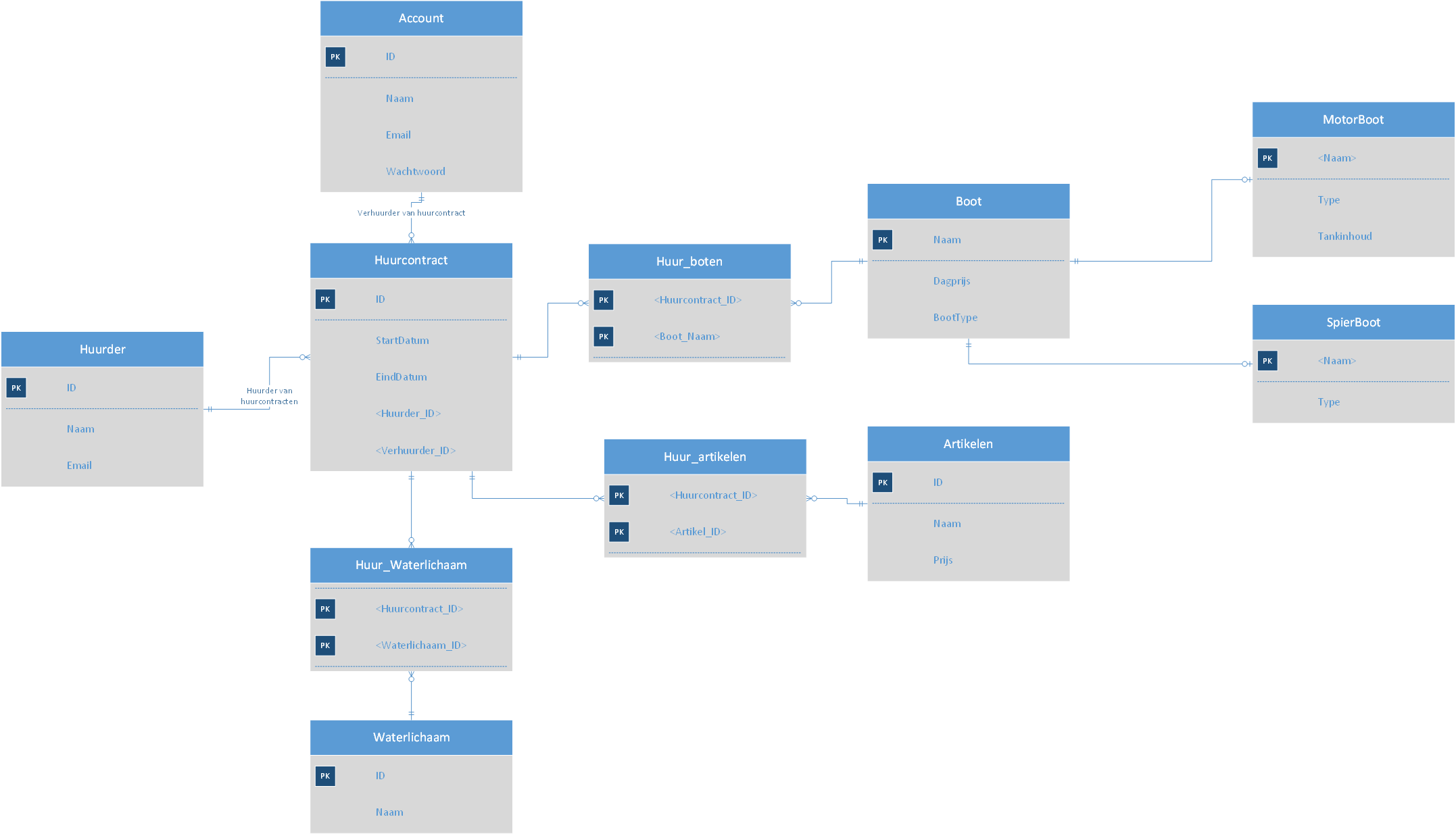
[Domeinmodel 7](#_Toc441066126)

[Implementatie model 8](#_Toc441066127)

## Inleiding

In dit document kunt u de ontwerpfase van ‘t Sloepske applicatie vinden. U kunt informatie vinden over de database ontwerp Vervolgens kunt u ook informatie vinden over de klassenstructuur, door middel van de Klassendiagram en toelichting daarvan.

## Database ontwerp



[Link naar grotere versie.](https://www.dropbox.com/s/nwjhv5wuz6fbosb/DBO.png?dl=0)

In tegenstelling tot de ERD, heeft de database ontwerp wel alle attributen/entiteiten uitwerkt. Hierbij kunt u al zien hoe de uit eindelijke database er uit gaat zien. Met de foreign key relaties aangeduid met een < >.

In plaats van de datamodel in de strokendiagram te zetten, heb ik er voor gekozen om dit in aparte tabellen te doen. Deze vind u hier onder, en op de volgende pagina. Hierbij zijn alle datatypes van elke attribuut weergegeven, en of ze verplicht ingevuld moeten zijn.

### Datamodel

|  |  |  |  |
| --- | --- | --- | --- |
| **Account** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| ID | NUMBER | Ja | PK |
| Naam | VARCHAR2 | Ja | Naam van gebruiker. |
| Email | VARCHAR2 | Ja | Unieke email |
| Wachtwoord | VARCHAR2 | Ja | Wachtwoord |

|  |  |  |  |
| --- | --- | --- | --- |
| **Huurder** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| ID | NUMBER | Ja | PK |
| Naam | VARCHAR2 | Ja | Naam van huurder. |
| Email | VARCHAR2 | Ja | Email van huurder |

|  |  |  |  |
| --- | --- | --- | --- |
| **HuurContract** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| ID | NUMBER | Ja | PK |
| BeginTijd | DATE | Ja | Begin tijd van verhuring |
| EindTijd | DATE | Ja | Eind tijd van verhuring. |
| <Huurder\_ID> | NUMBER | Ja | FK van huurder. |
| <Verhuurder\_ID> | NUMBER | Ja | FK van account. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Huur\_Waterlichaam** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| <HuurContract\_ID> | NUMBER | Ja | PK, FK van huurcontract. |
| <Waterlichaam\_ID> | NUMBER | Ja | PK, FK van UniekMeer. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Waterlichaam** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| ID | NUMBER | Ja | PK |
| Naam | VARCHAR2 | Ja | Naam van meer |

|  |  |  |  |
| --- | --- | --- | --- |
| **Huur\_Artikelen** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| <HuurContract\_ID> | NUMBER | Ja | PK, FK van huurcontract. |
| <Artikel\_ID> | NUMBER | Ja | PK, FK van Artikel. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Artikel** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| ID | NUMBER | Ja | PK |
| Naam | VARCHAR2 | Ja | Naam van artikel. |
| Prijs | NUMBER | Ja | Prijs van artikel. |

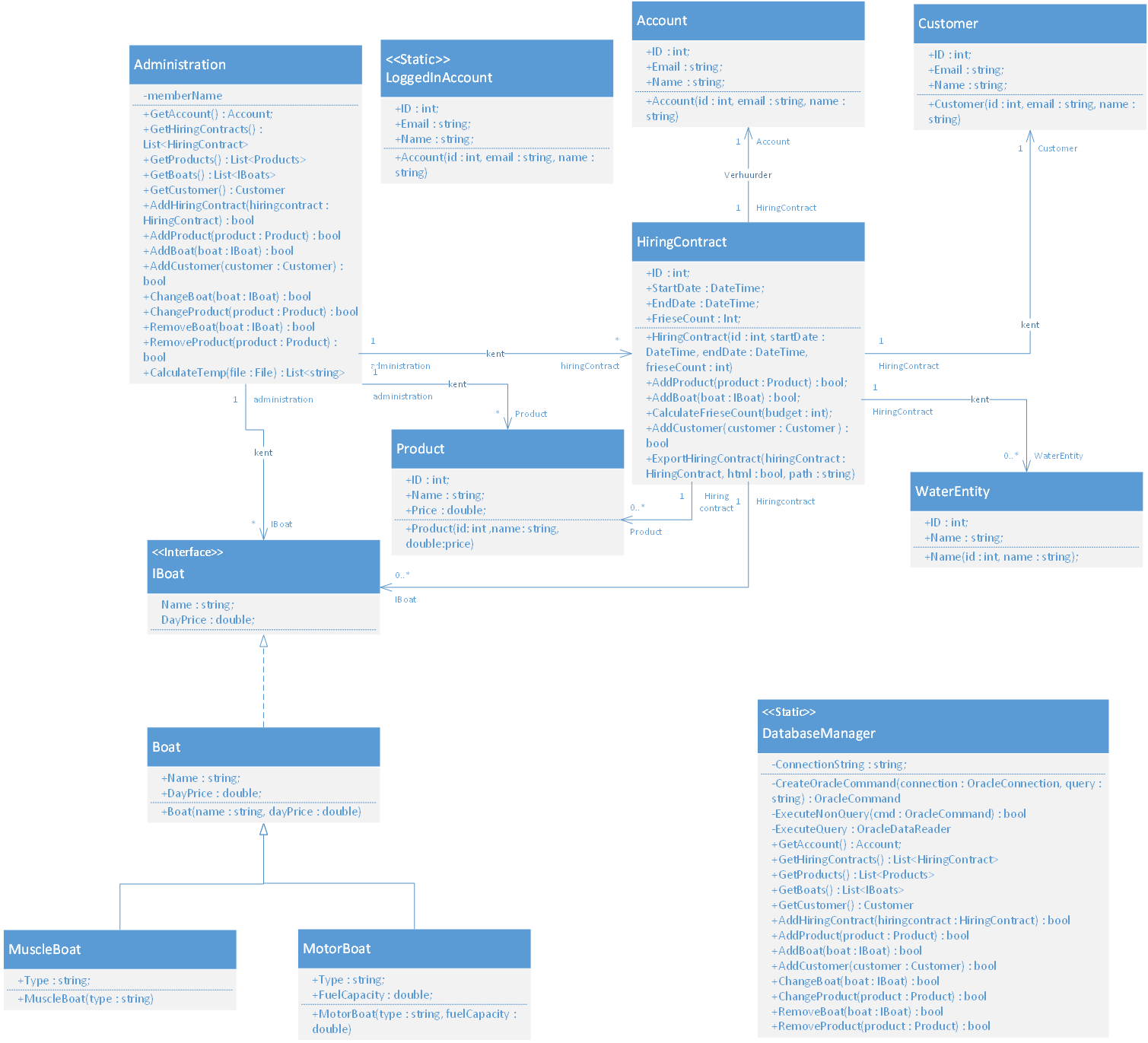
|  |  |  |  |
| --- | --- | --- | --- |
| **Huur\_boten** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| <HuurContract\_ID> | NUMBER | Ja | PK, FK van huurcontract. |
| <Boot\_Naam> | VARCHAR2 | Ja | PK, FK van Boot. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Boot** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| Naam | VARCHAR2 | Ja | PK, Naam van boot. |
| Dagprijs | NUMBER | Ja | Dag prijs van boot. |
| BootType | VARCHAR2 | Ja | Welke type boot het is, motorboot of spierboot. |

|  |  |  |  |
| --- | --- | --- | --- |
| **MotorBoot** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| <Naam> | VARCHAR2 | Ja | PK, FK van Boot. |
| Type | VARCHAR2 | Ja | Wat voor model boot het is. |
| Tankinhoud | NUMBER | Ja | Hoeveel benzine/diesel er in de brandstoftank past. |

|  |  |  |  |
| --- | --- | --- | --- |
| **SpierBoot** | | | |
| **Kolom** | **Datatype** | **Verplicht?** | **Omschrijving** |
| <Naam> | VARCHAR2 | Ja | PK, FK van Boot. |
| Type | VARCHAR2 | Ja | Wat voor model boot het is. |

## Klassendiagram



[Link naar grotere versie.](https://www.dropbox.com/s/iyitmzo2m81t4mz/KD.png?dl=0)

Ik heb gekozen voor één overkoepelende administratie klasse om de onderliggende objecten te benaderen via de GUI laag. Als eerste wat opvalt zijn de static klassen IngelogdAccount en de DatabaseManager. Door ze static te maken zijn ze altijd bereikbaar voor alle klassen zonder er nieuwe instanties voor hoeven te maken. Met betrekking tot de database is dit vooral handig. Dit zorgt er voor dat er niet telkens nieuwe instanties en dus nieuwe connecties met de database worden aangemaakt. De reden voor de statische klassen ingelogdAccount is om een account object bereikbaar te houden voor alle klassen indien er met dit ingelogd account handelingen worden verricht waarbij dit object moet worden meegestuurd.

Voor boten was er wel een interface nodig om makkelijk de child klassen van Boat, terug te kunnen casten van een IBoat naar een MuscleBoat of MotorBoat.

Daarnaast kent de administratie klasse een lijst van alle producten, boten, en verhuurcontracten. Dit was nodig om straks de administratieve taken voor administrator te realiseren, en om gebruikers een overzicht van verhuurcontracten te geven voor het handmatig exporteren van deze contracten.

De klasse WaterEntity is er voor de uitbreidbare wensen van de client, door alle unieke water entiteiten(meren of zeeën) daar in op te slaan. Zoals de Noordzee, IJsselmeer en in de toekomst de Waddenzee.

In de hierin bovenstaande implementatie model is de uiteindelijke versie van de software klassen te zien. Alle lijsten zijn weergegeven als lijnen(relaties), ook ziet u bij sommige constructors +1 staan. Dit komt vanwege de XML exporteer methode, deze had lege constructors nodig om de objecten om te zetten naar een XML formaat. Aan de hand van deze implementatie model hoop ik u een beeld te geven van hoe de applicatie er uit zou zien.

Er is gekozen voor een huidigbezoek en huidigproject, voor handigheid tijdens coderen. Dit zorgde ervoor dat ik niet telkens via de project op zoek moest gaan naar de bezoek waarmee ik aan het werk was.