Ontwerpdocument

Live performance

Roy v.Oldenbeek

2016

Inhoud

[Inleiding 2](#_Toc454370072)

[Architectuur 3](#_Toc454370073)

[Database model 4](#_Toc454370074)

[Klassendiagram 5](#_Toc454370075)

# Inleiding

In dit ontwerpdocument voor de bootverhuur applicatie van ’t Slopke. Met deze applicatie moet het eenvoudiger worden om boten en het bijbehorende materiaal te verhuren voor ’t Slopke. In dit document wordt het databasemodel en het klassendiagram weergegeven. Deze twee diagrammen zullen helpen bij het realiseren van de applicatie. Ook wordt in dit document vastgelegd met welke programmeer architectuur er gebruikt gaat worden.

# Architectuur

De applicatie zal gebruik maken van een 3-lagenstructuur. In deze structuur is een duidelijke afscheiding tussen de User Interface, de achterliggende logica en de communicatie met een externe database. De User Interface mag alleen maar interactie hebben met de business laag (de laag met achterliggende logica) en vanuit de business laag moet met de database laag interactie komen. Er zal een globale Database klasse komen vanwaar de subclassen geinherit worden voor de verschillende onderdelen van de applicatie; elke business laag die iets met de database doet zal zijn eigen bijbehorende database klasse krijgen.

# Database model

Dit is het datamodel, hierin staat per tabel aangegeven welke data er in op wordt geslagen, samen met welke foreign en primary keys er zijn en alle relaties tussen de tabellen.

Zoals te zien is zitten er geen dubbele relaties meer in dit model. Dit is allemaal opgelost met koppeltabellen. De discriminator Type is opgelost door een type tabel te genereren en de foreign key Type in de Boot tabel. Vaarwater zit nu via Type gekoppeld aan een boot, dit omdat er veel minder data in de database komt te staan hierdoor aangezien er veel minder types zijn als boten. In de koppel tabellen zijn de foreign keys samen de primary key; zodat er geen extra ID moet worden aangemaakt. Een N betekend dat het veld niet null mag zijn.



# Klassendiagram

Het klassendiagram geeft de verschillende te implementeren klassen weer voor de Applicatie. Met deze klassen en de functies die zij bevatten moet de applicatie goed functioneren.



Er is voor gekozen om de database en boot klasse abstract te maken; deze abstracte klassen zorgen er voor dat er bij de database eenvoudig een andere connectie kan worden toegepast en dit gelijk voor alle klassen geld, wat de onderhoudbaarheid en draagbaarheid ten goede komt. Voor de boot klassen betekend dit dat er eenvoudig niet alleen meerdere boten van een type toegevoegd kunnen worden maar ook nog eens een extra type boot mocht dit nodig zijn. Boot en Materiaal hebben beide de interface iProduct, dit omdat beide klassen een naam en een prijs hebben. Verder heeft elke klasse zijn eigen Database klasse, die afgeleid wordt van de abstracte database klasse.