Awin Kroesen

11-12-2015

Gegevensverzameling

Project Omega

Inhoud

[Inleiding 2](#_Toc443669996)

[Data dictionary 3](#_Toc443669997)

[Relaties 5](#_Toc443669998)

[Gegevens 8](#_Toc443669999)

[Algemeen 8](#_Toc443670000)

[vormgeving 8](#_Toc443670001)

[Pagina’s 8](#_Toc443670002)

[Interacties 8](#_Toc443670003)

[Software gegevensverzameling 9](#_Toc443670004)

[WOrd 9](#_Toc443670005)

[Xampp 9](#_Toc443670006)

[Phpstorm 9](#_Toc443670007)

[Github 9](#_Toc443670008)

# Inleiding

De opdracht is om voor Omega-opleidingen een website te maken. Nu is mij gevraagd of ik dit wil maken. Ik heb een aantal gegevens gekregen zoals het plan van aanpak, functioneel en technisch ontwerp. Aan de hand van die verslagen heb ik een aantal gegevens verzameld. Dit is te vinden in dit verslag.

Ik heb een data dictionary gemaakt om de database tabellen in kaart te krijgen. Ook heb ik een SQL-script toegevoegd van de tabellen. Verder heb ik nog gegevens verzameld over de inhoud voor de pagina’s, de vormgeving en interacties.

Als slot vertel ik nog over de software die ik gebruik en waarom ik dit gebruik. Ik gebruik Word, Xampp, Phpstorm en Github.

# Data dictionary

Een aantal gegevens zijn anders omdat ik Symfony 2 gebruik. Tussentabellen worden automatisch gegenereerd door Symfony/Doctrine, daarom heb ik die hierin niet in opgenomen.

|  |  |  |
| --- | --- | --- |
|  | Cursus | |
| Definitie | Gegevens van een cursus | |
| Attribuutnaam | Datatype | Toelichting |
| Id | Int() | Unique identiteit (Primary key) 1 |
| Begindatum | Date | De datum waarop de cursus begint. |
| Einddatum | Date | De datum waarop de cursus eindigt. |
| Image | Varchar(255) | Afbeelding 2 |
| Soort cursus | Varchar(3) | De soort cursus |

1. Dit wordt automatisch door Symfony gedaan.
2. Omdat de website een professionele uitstraling moet hebben en het gaat om nieuwe cursisten erbij te krijgen is het handig dat zij dan ook een beeld hebben.

|  |  |  |
| --- | --- | --- |
|  | SoortCursus | |
| Definitie | De soort van de cursus | |
| Attribuutnaam | Datatype | Toelichting |
| Id | Int() | Unique identiteit (primary key)1 |
| Naam | Varchar(100) | De naam van de soort cursus2 |
| Prijs | Int(10) | De prijs in centen |

1. Dit wordt automatisch door Symfony gedaan.
2. Ik heb attribuutnaam van soortCursus veranderd in naam. Dit is een logischer benaming.

|  |  |  |
| --- | --- | --- |
|  | Gebruiker1 | |
| Definitie | Een cursist of beheerder | |
| Attribuutnaam | Datatype | Toelichting |
| id | Int(3) | Unique identiteit (primary key)2 |
| E-mailadres | Varchar(100) | Cursist zijn e-mailadres |
| Roepnaam | Varchar(20) | Cursist zijn roepnaam |
| Tussenvoegsels | Varchar(8) | Tussen voegsels van de cursist |
| Achternaam | Varchar(100) | Achternaam van de cursist |
| Adres | Varchar(100) | Adres van de cursist |
| Woonplaats | Varchar(100) | Woonplaats van cursist |
| Telefoon | Varchar(12) | Telefoonnummer(rekening gehouden met +31) van cursist |

1. Dit is een cursist of een gebruiker. Dit wordt bepaald door de tabel hieronder.
2. Dit wordt automatisch door Symfony gedaan.

|  |  |  |
| --- | --- | --- |
|  | Role | |
| Definitie | Gebruiker roles | |
| Attribuutnaam | Datatype | Toelichting |
| id | Int() | Unique identifier (primary key) |
| name | Varchar(30) | Leesbare naam van de role |
| role | Varchar(20) | De role. |

Deze tabel maakt het makkelijker om meerdere gebruiker ‘levels’ aan te maken. Momenteel zijn er 2, je hebt namelijk cursisten en administrators.

## Relaties

|  |  |  |
| --- | --- | --- |
| **Van** | **Naar** | **Type** |
| **SoortCursus** | **Cursus** |  |
| soortCursus | id | One to many |
| **Cursist** | **cursus** |  |
| id | id | Many to many |

SQL-script

In Symfony werk je niet echt met de database. Je maakt een bestand met daarin de kolom namen en met Orm annotaties zet je wat voor type en lengte.

|  |
| --- |
| -- -----------------------------------------------------  -- Table `Soortcursus`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `Soortcursus` (  `soortcode` INT(3) NOT NULL COMMENT '',  `cursussoort` VARCHAR(100) NULL COMMENT '',  `prijs` INT(10) NULL COMMENT '',  PRIMARY KEY (`soortcode`) COMMENT '',  UNIQUE INDEX `cursussoort\_UNIQUE` (`cursussoort` ASC) COMMENT '')  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `Cursus`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `Cursus` (  `cursuscode` INT(3) NOT NULL COMMENT '',  `begindatum` DATE NULL COMMENT '',  `einddatum` DATE NULL COMMENT '',  `soort\_cursus` INT(3) NULL COMMENT '',  PRIMARY KEY (`cursuscode`) COMMENT '',  UNIQUE INDEX `soort\_cursus\_UNIQUE` (`soort\_cursus` ASC) COMMENT '',  CONSTRAINT `SoortCursus`  FOREIGN KEY (`soort\_cursus`)  REFERENCES `Soortcursus` (`soortcode`)  ON DELETE NO ACTION  ON UPDATE NO ACTION)  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `Cursist`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `Cursist` (  `id` INT(3) NOT NULL COMMENT '',  `email\_adres` VARCHAR(100) NOT NULL COMMENT '',  `roepnaam` VARCHAR(20) NULL COMMENT '',  `tussenvoegsels` VARCHAR(8) NULL COMMENT '',  `achternaam` VARCHAR(100) NULL COMMENT '',  `adres` VARCHAR(100) NULL COMMENT '',  `woonplaats` VARCHAR(100) NULL COMMENT '',  `telefoon` VARCHAR(12) NULL COMMENT '',  PRIMARY KEY (`id`) COMMENT '',  UNIQUE INDEX `email\_adres\_UNIQUE` (`email\_adres` ASC) COMMENT '')  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `CursistCursus`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `CursistCursus` (  `cursuscode` INT(3) NOT NULL COMMENT '',  `cursist` INT(3) NULL COMMENT '',  PRIMARY KEY (`cursuscode`) COMMENT '',  UNIQUE INDEX `cursist\_UNIQUE` (`cursist` ASC) COMMENT '',  CONSTRAINT `cursistCursusCursist`  FOREIGN KEY (`cursist`)  REFERENCES `Cursist` (`id`)  ON DELETE NO ACTION  ON UPDATE NO ACTION,  CONSTRAINT `cursistCursusCursus`  FOREIGN KEY (`cursuscode`)  REFERENCES `Cursus` (`cursuscode`)  ON DELETE NO ACTION  ON UPDATE NO ACTION)  ENGINE = InnoDB; |

# Gegevens

## Algemeen

|  |  |
| --- | --- |
| Doel: | Belangstelling opwekken voor het opleidingscentrum en alle cursussen |
| Doelgroep: | Leergierige consument die een cursus wil volgen.  Medewerkers die verschillende overzichten kunnen raadplegen |

## vormgeving

|  |  |
| --- | --- |
| Hoofdkleuren: | Rood en zwart |
| Titel en logo: | Midden bovenin |
| Account acties: | Links bovenin, inloggen, na inloggen: uitloggen en naam |

## Pagina’s

|  |  |  |
| --- | --- | --- |
| **Pagina** | **Inhoud** | **Voor wie?** |
| Homepagina | Algemene informatie met een foto van het gebouw en een klaslokaal | Iedereen |
| Contactpagina | Adressen en telefoonnummers van het opleidingscentrum “Omega”, algemeen e-mailadres | Iedereen |
| Registratie | Registratieformulier | Iedereen1 |
| Cursussen | Pagina met alle cursussen, waar cursisten zich voor 1 of meer kunnen aanmelden | cursisten |
| Wijzig account | Pagina waar een gebruiker zijn gegevens kan aanpassen | Ingelogden |
| Ingeschrevenen | Welke cursist er is ingeschreven voor bepaalde cursussen | medewerkers |
| Cursussen | Pagina met alle cursussen, waar cursisten voor ingeschreven zijn en of ze vol zijn of niet. | medewerkers |

1. Iedereen die niet is ingelogd.

## Interacties

* Cursisten moeten zich kunnen registreren.
* Cursisten moeten na registratie kunnen inloggen om zich in te schrijven voor een of meer cursussen.
* Cursisten moeten kunnen uitloggen.
* Cursisten moeten hun gegevens kunnen wijzigen.
* Medewerkers moeten kunnen in- en uitloggen.
* Medewerkers moeten een lijst kunnen uitdraaien waarop de inschrijvingen van een bepaalde cursus staan.
* Medewerkers moeten een lijst kunnen uitdraaien van alle cursussen in een bepaalde periode.
* Medewerkers moeten cursussen kunnen toevoegen, gegevens van cursussen kunnen wijzigen en kunnen verwijderen.

# Software gegevensverzameling

De benodigde software dat ik ga gebruiken zijn Word, Apache server (Xampp), MySQL server (Xampp), PhpMyAdmin (Xampp) en tekstverwerker (Phpstorm). Hiernaast gebruik ik ook nog Github.

### WOrd

Voor de documentatie gebruik ik Word.

### Xampp

Om een website te laten werken is er een server nodig. Om lokaal op je eigen laptop of computer een server te krijgen is er Xampp. Hiermee kunt u met een paar klikken een eigen server met database laten draaien. Om vervolgens connectie te maken met de database zijn er meerdere mogelijkheden. Één daarvan is phpmyadmin die ik gebruik.

### Phpstorm

Ik heb een tekstverwerker nodig voor de code. Een hele goede is Phpstorm. Ook zijn er wel andere maar ik heb zelf veel hiermee gewerkt.

### Github

Github is een online versiebeheer dat gebruik maakt van Git. Doormiddel van Git kun je makkelijker met elkaar aan een project samen werken en kun je versies terug zetten. Erg handig wanneer er iets in de code niet helemaal juist is en de applicatie niet meer werkt. De aanpassingen kunnen eenvoudig terug bekeken worden en eventueel dingen teruggezet worden.