

Architectuur van de Jeu de Boules club webapp

Huub van Thienen

9 januari 2014

1 Inleiding

Voor het vak WEB wordt er een praktijktoets gehouden over het bouwen van webapplicaties. Hierbij worden de vaardigheden van de studenten op het gebied van webapplicatie ontwikkeling getoetst, in het bijzonder op het gebied van JSPs, servlets en hibernate.

Om praktische redenen bestaan de opgaven voor de toets uit uitbreidingen en aanpassingen van een gegeven (kleine) webapplicatie. Tot nu toe werd daarvoor steeds een applicatie van het practicum gebruikt die de studenten al kenden. In het kader van de onderwijsvernieuwing is de laatste tijd door docenten van de HvA gewerkt aan een modernisering van het praktikum en op de praktijktoets van januari 2014 zal van één van de nieuw ontwikkelde praktikumapplicaties gebruik worden gemaakt.

Omdat de studenten van de huidige generatie die opgave nog niet kennen wordt die in de week voor de toets gepubliceerd, samen met deze beschrijving. Bij de voorbereiding van de toets kan hiervan gebruik worden gemaakt.

In dit document worden achtereenvolgens het domein, de technische achtergrond en de voorbereiding van de toets besproken.

2 Domein

De webapplicatie is gemaakt voor de administratie van een jeu de boules club. In deze paragraaf worden de relevante aspecten van de club besproken en beschreven hoe de applicatie de ledenadministratie ondersteunt.

Jeu de Boules is een vooral in Frankrijk populair spelletje. Ook veel Nederlanders spelen het graag, vooral als ze in Frankrijk op vakantie zijn. Het is dan ook geen verrassing dat er in Nederland clubs zijn waarin het jeu de boules-spel in wedstrijdverband gespeeld wordt. De applicatie die hier beschreven is gaat over de ledenadministratie van zo'n club.

De naam van de club is eenvoudigweg Jeu De Boules Club, afgekort tot JDBC. Dat heeft in dit geval helemaal niets te maken met het Java DataBase Connectivity framework (natuurlijk niet, we gebruiken Hibernate toch?)

De JDBC houdt een ledenlijst bij waarin voorlopig alleen de voor- en achternaam van de leden worden vastgelegd. Omdat er alleen in teamverband aan wedstrijden wordt meegedaan is er ook een administratie van de teams. De spelregels van jeu de boules schrijven voor dat een team uit twee of drie deelnemers kan bestaan. Ieder lid kan in maximaal één team deelnemen. Omdat er ook leden zijn die alleen voor de gezelligheid komen en niet aan wedstrijden willen deelnemen is deelname aan een team niet verplicht. Ieder team heeft een coach. Meestal is één van de deelnemers ook de coach maar dat hoeft niet, er kan een aparte coach zijn die zelf niet deelneemt in het team. In dat geval is het mogelijk dat de coach deelneemt in een ander team maar dat is niet gebruikelijk.

Omdat de teams klein zijn, zijn er ook heel veel. Het bestuur van de JDBC organiseert alleen de topteams waarvan je alleen op uitnodiging lid kan worden. Alle andere teams worden door de leden zelf georganiseerd. Iedereen kan een team starten. De initiator van een team is meteen de eerste deelnemer en de coach maar om aan een wedstrijd mee te kunnen doen moet er tenminste nog één andere deelnemer zijn.

Leden kunnen zoeken naar een team met minder dan drie deelnemers en zich daar voor aanmelden. Het is niet mogelijk je aan te melden voor een team met drie deelnemers. Alle aanmeldingen worden in eerste instantie geaccepteerd maar de coach van een team kan deelnemers verwijderen.

3 Functionaliteit

Omdat het beheer van de leden en de teams veel tijd kost wil het bestuur van de JDBC een webapplicatie waarmee de administratie kan worden ondersteund. Die applicatie moet uiteindelijk tenminste de volgende functionaliteit bieden:

Voor het bestuur:

1. Leden toevoegen.
2. Gegevens van leden wijzigen.
3. Leden verwijderen.
4. Teams toevoegen.
5. Teams wijzigen.
6. Teams verwijderen.
7. Deelnemers aan een team toevoegen.
8. Deelnemers van een team verwijderen.

Voor gewone leden:

1. Eigen gegevens wijzigen.
2. Nieuw team aanmaken.
3. Teams met minder dan drie deelnemers zoeken.
4. Zichzelf aan een team toevoegen.
5. Deelnemers (andere dan zichzelf) aan een team toevoegen (alleen voor coaches van een team).
6. Deelnemers uit een team verwijderen (alleen voor coaches van een team).

De applicatie is nog volop in ontwikkeling, op dit moment wordt alleen het wijzigen van ledengegevens door het bestuur ondersteund. Voor de praktijktoet maken we een kleine uitbreiding. Het is niet de bedoeling dat je de volledige functionaliteit oplevert!

4 Architectuur

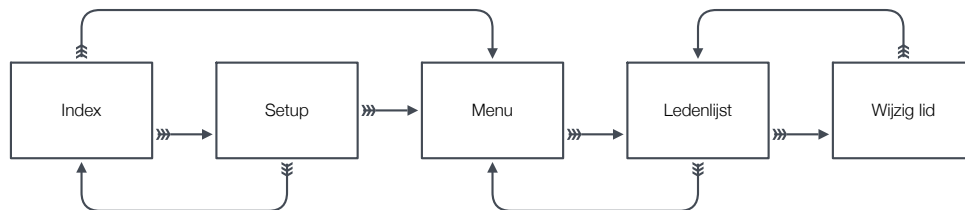
Bij het ontwerpen van de webapplicatie is vooral rekening gehouden met drie aspecten die niet toevallig ook veel aandacht hebben gekregen in het vak WEB:

1. De navigatie, d.w.z. hoe de flow van het ene scherm naar het andere wordt gerealiseerd.
2. De request logistiek, d.w.z. hoe een request en een response door het systeem worden gevoerd vanaf het moment dat de request binnenkomt totdat de definitieve response wordt verzonden.
3. De koppeling met de database en het gebruik van Hibernate daarvoor.

Het gevolg van deze keuze is dat een aantal andere aspecten onderbelicht zijn gebleven. In het bijzonder gaat het dan over de grafische vormgeving en het gebruik van moderne user interface technieken.

Navigatie De applicatie is vormgegeven als een interactie dialoog tussen systeem en gebruiker. Iedere stap in die dialoog wordt een *form* genoemd. Een form bestaat uit twee delen: de *form page* en de *form handler*. De form page is een JSP die HTML genereert met een web formulier. Dat formulier wordt aan gebruiker getoond. De gebruiker vult het formulier in en verstuurt het naar de server. De form handler is een servlet die de inhoud van dit formulier verwerkt (bijv. door de inhoud in en database te zetten of een

validatie uit te voeren). De laatste stap van de form handler is de bepaling van het volgende form. Met een `RequestDispatcher.forward(req, res)` wordt naar deze form gesprongen.



Figuur 1: Dialog flow van de applicatie

Naamgeving Iedere form in de dialoog heft een naam. In de applicatie zoals die er nu uit ziet zijn er vijf forms (zie ook fig. 1):

1. Index
2. Setup
3. Menu
4. Ledenlijst
5. Wijzig lid

De JSP en de servlet die een bepaalde form implementeren hebben namen die hiervan zijn afgeleid. De JSP van form `XYZ` heet `xyz.jsp`. Voor een servlet bestaat de naamgeving uit drie onderdelen: de naam van de servlet, de naam van de implementerende Java klasse en het url patroon waaronder de servlet van buiten af bereikbaar is.

De naam van de servlet van form `XYZ` is `xyzhandler`, de implementerende klasse heet `formhandler.Xyz` en het URL patroon is `/xyzhandler`.

Request logistiek Vanuit het oogpunt van de gebruiker ziet de applicatie eruit zoals hierboven beschreven als een opeenvolging van forms. Iedere form begint met een JSP en eindigt met een servlet. Maar vanuit de applicatie moet de situatie subtiel anders worden beschreven.

Stel dat een gebruiker het formulier van form `XYZ` invult. Een submit van dit formulier (door op een submit button te klikken) resulteert in een http post request. Deze request wordt verwerkt door de servlet `xyzhandler`. (De form action van het formulier is daarom ook `acion="xyzhandler"`). De servlet bepaalt uiteindelijk welke form als volgende wordt gekozen. De JSP van de vervolg form genereert dan de response.

Het gevolg van deze observaties is dat de “unit of work” voor de gebruiker een form is, maar voor de applicatie is dat de combinatie servlet van het ene form - JSP van het volgende form. In deze combinatie wordt de hele verwerking van het binnenkomende request tot en met het versturen van het bijbehorende uitgaande request afgehandeld. De hele combinatie valt binnen dezelfde request scope.

Het is daarom mogelijk dat een servlet informatie genereert die in de volgende JSP nodig is. Denk aan een lijst objecten die door de servlet uit de database wordt gehaald en die in de JSP nodig is om een tabel met gegevens te vullen. Omdat zowel de servlet als de JSP in dezelfde request scope vallen is het mogelijk deze informatie als attribuut aan de request te binden. Bij een `RequestDispatcher.forward(request, response)` wordt de request met alle attributen doorgegeven van de servlet naar de JSP.

Een ander gevolg is dat de JSP en de servlet van een form *niet* binnen dezelfde request scope vallen. Doorgeven van informatie van de JSP naar de servlet kan dus alleen via de HTTP session!

Domain model en database koppeling Het domeinmodel bestaat uit twee klassen: `model.Member` en `model.Team`. Deze klassen worden geper sisteerd in de database in de tabellen `ClubMember` en `Team`. De koppeling met de database is gebouwd met Hibernate. Voor iedere entiteit in het domeinmodel is er een dao (`dao.MemberDao` resp. `dao.TeamDao`) gemaakt die de communicatie met de database regelt.

5 Voorbereiding voor de praktijktoets

Bij de beoordeling van je werk wordt een groot gewicht toegekend aan de correcte werking van je applicatie. Daarom is het van belang dat de docent je opgeleverde werk snel en eenvoudig in zijn eigen omgeving kan installeren en testen. Om dat mogelijk te maken moet je werk aan bepaalde standaarden voldoen die hier nader worden toegelicht. Een deel van dit werk kan je thuis doen voordat de toets begint.

Installatie Je *moet* werken met de volgende tools en systemen:

1. Netbeans 7.4
2. Tomcat 7.0
3. MySQL 5

Het is handig om een database client te hebben. MySQL Workbench is een goede keuze maar je mag ook een andere gebruiken.

Je kan het project downloaden van de VLO (als je dit document leest heb je dat waarschijnlijk al gedaan). Na uitpakken in Netbeans zie je dat het project Naam-Herkansing-WEB-JEE heet. Vervang de string “Naam” hierin door je eigen naam. (Je moet dus alleen het project een nieuwe naam geven, niet de applicatie!).

Maak een lege database met de naam WEB-JEE-Club aan in MySQL en maak een connectie met die database vanuit Netbeans. Pas de file hibernate.cfg.xml aan door de gebruikersnaam en het password voor de database in te voeren.

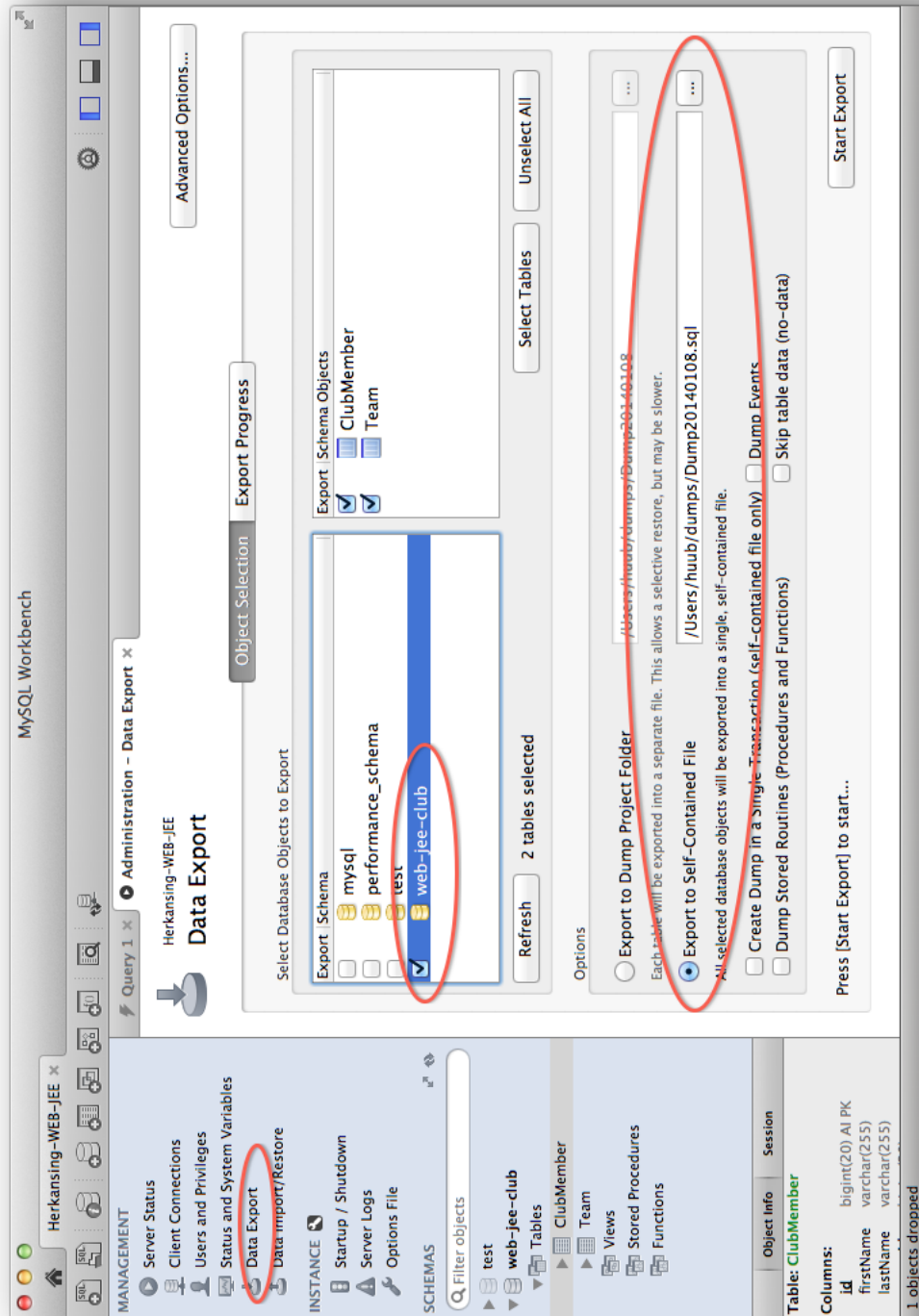
Start de applicatie vanuit Netbeans en klik op de link “Klik [hier](#) voor de database setup”. Je komt dan in een pagina met twee knoppen (en de tekst “Hier wordt de database opgezet”). Klik op de knop Setup database. De applicatie maakt dan de twee tabellen die je nodig hebt en vult die met voorbeeld data. Daarna kom je terecht in het hoofdmenu van de applicatie. Controleer met je MySQL client of de database correct is aangemaakt. Klik in het hoofdmenu van de applicatie op de knop Ledenlijst en controleer of je inderdaad de ledenlijst ziet.

Pas op! Als je de database terug wil zetten op de oorspronkelijke inhoud moet je eerst de twee tabellen weggooien (drop table, vanuit de MySQL Workbench) voordat je je opnieuw op deze link klikt.

Oplevering Als je klaar bent met de toets moet je je werk uploaden naar Moodle. Je moet zowel de applicatie als de database inleveren! Ga daarvoor als volgt te werk:

1. Exporteer de database m.b.v. je mySQL client als een self-contained file. (Zie de bijlage voor deze actie in de MySQL Workbench).
2. Kopieer de sql file uit stap 1 naar de root directory (default package) van de applicatie in Netbeans.
3. Exporteer je Netbeans project (inclusief de sql file uit de vorige stappen) als zip file. Pas op! Alleen zip files worden in behandeling genomen. Andere compressieformaten worden niet geaccepteerd.

6 Bijlage



Figuur 2: Export van de database als self-contained file