



# Projektdokumentation

**Projektseminar I4**

**„Angewandte Informationswissenschaft“**

Name: Dennis Gelfart

Matrikelnr.: 2181655

GitHub: degel100

## **Einleitung und Projektidee**

Mein Projekt, welches ich im Rahmen des Projektseminars „Angewandte Informationswissenschaft“ realisiert habe, basiert auf der Idee, eine Datenbank für die olympischen Sommerspiele in Rio de Janeiro 2016 zu gestalten und zu programmieren.

## **Definitionen, Hintergrundinformationen und Begründungen**

Die olympischen Sommerspiele in Rio sind vor kurzer Zeit zu Ende gegangen, was bedeutet, dass das Thema *Olympia* eine noch andauernde Aktualität darstellt.

Die Arbeit mit Datenbanken hat mir bisher in meinem Studium schon im Modul „Datenbanksysteme“ viel Spaß gemacht und darüber hinaus schon einige Erfahrungen in der Gestaltung von grafischen Benutzeroberflächen mit Java und JavaFX eingebracht.

Schlussendlich wurde im Laufe der Vorgespräche innerhalb der Seminarwoche das Ziel gesetzt, eine Datenbank zu den Olympischen Sommerspielen 2016 mithilfe Java und JavaFX zur grafischen Gestaltung zu programmieren.

Zum Projektbeginn habe ich das Ergebnis klar definiert: Die entstehende Datenbank soll dem Nutzer die Möglichkeit geben, Informationen zu den Sportlern der olympischen Sommerspiele 2016 einzuholen, sowie in Form eines Medaillenspiegels einen allgemeinen Überblick über die erfolgreichsten Sportler und Sportarten sowie Länder gewährleisten.

Des Weiteren sollen autorisierte Nutzer mittels Zugangsdaten die Möglichkeit bekommen, Spieler aus der Datenbank zu entfernen oder hinzuzufügen.

## **Projektaufbau**

### **Elemente und Funktionen**

Das fertige Programm besteht schlussendlich aus vier Elementen. Jedes Element stellt eine eigene grafische Benutzeroberfläche dar.

Das erste Element ist die Startseite des Programms zur Datenbankverwaltung. Sie besteht aus dem Logo der olympischen Sommerspiele 2016 sowie einem Hintergrundbild der olympischen Stadt Rio de Janeiro in Brasilien mit der monumentalen Christusstatue (Abbildung 1).

Von ihr aus gelangt man mithilfe des Buttons „Olympische Spieler“ zu Element zwei, dem Login-Fenster.



Abbildung 1. Element eins, die Startseite.

An dieser Stelle ist es möglich, sich entweder mittels Login-Daten, bestehend aus einem Benutzernamen und einem Passwort, Zugang zu den erweiterten Optionen der Datenbank zu verschaffen, oder sie als Gast zu betreten (Abbildung 2.1). Die Zugangsdaten für autorisierte Nutzer setzen sich aus dem Benutzernamen *Nutzer* und dem Passwort *Zugang* zusammen. Sind entweder Benutzername oder Passwort nicht korrekt, so erscheint eine Meldung die darauf hinweist (Abbildung 2.2).

Im Folgenden gehe ich zunächst auf den Fall eines erfolgreichen Logins ein, womit sich anschließend bei einem Klick auf dem Button „Login“ Element drei, die Spielerdatenbank mit den erweiterten Optionen öffnet.



Abbildung 2.1. Element zwei, das Login-Fenster.



Abbildung 2.1. Element zwei, das Login-Fenster nach nicht erfolgreicher Anmeldung.

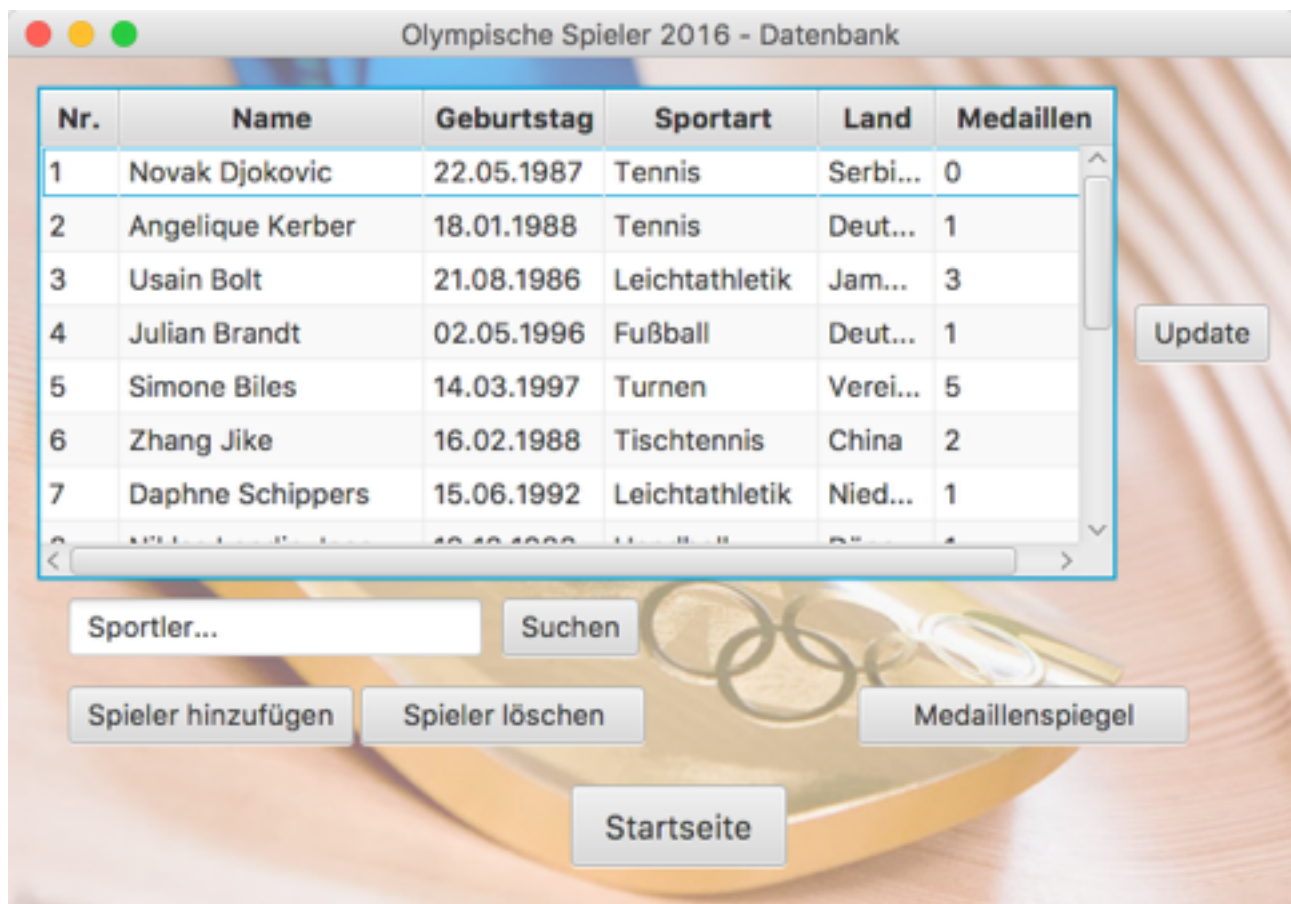


Abbildung 3.1. Element drei, die Spielerdatenbank für autorisierte Nutzer.

Die Spielerdatenbank für autorisierte Nutzer (Abbildung 3.1) beinhaltet folgende Komponenten:

- Spielertabelle mit den Spalten *Nr.*, *Name*, *Geburtsdag*, *Sportart*, *Land* und *Medaillen*
- Suchfunktion zu einzelnen Sportlern, bestehend aus Suchfeld und Suchen-Button
- Funktion, einen Spieler zur Datenbank hinzuzufügen („Spieler hinzufügen“)
- Funktion, einen Spieler aus der Datenbank zu löschen („Spieler löschen“)
- Funktion, sich ein Spielerprofil in einem Extrafenster anzeigen zu lassen („Profil anzeigen“); erst sichtbar nach erfolgreicher Suche
- Medaillenspiegel anzeigen („Medaillenspiegel“)
- Update der Tabelle nach Löschen eines Spielers aus der Datenbank („Update“)
- Rückkehr zur Startseite

Dem Nutzer ist es möglich, die Spielertabelle mittels Klick auf die einzelnen Spaltennamen auf die jeweilige Spalte zu sortieren sowie in ihr hin- und her zu scrollen.

Möchte man einen bestimmten Sportler suchen, so kann man dies mit der Suchfunktion tun. Dazu tippt man den Namen des Sportlers in das Suchfeld ein und klickt anschließend auf „Suchen“. Ist der Sportler in der Datenbank enthalten, so wird einem über ein Label angezeigt, an welcher Stelle in der Tabelle er gelistet ist und der „Profil anzeigen“-Button erscheint (Abbildung 3.2). Im Falle eines negativen Treffers wird dem Nutzer angezeigt, dass der Sportler nicht in der Datenbank enthalten ist und der Profil-Button verschwindet wieder (Abbildung 3.3).

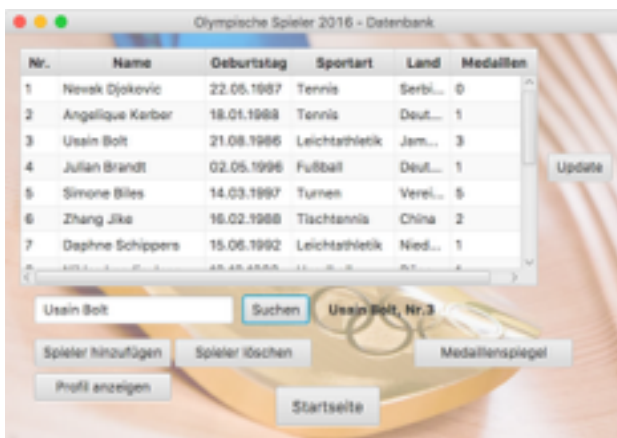


Abbildung 3.2.

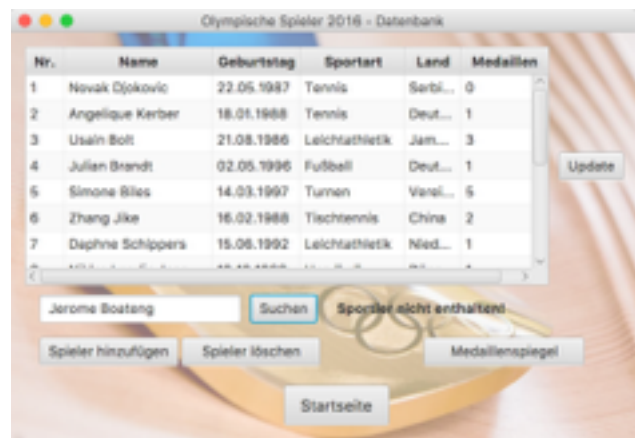


Abbildung 3.3.

Als autorisierter Nutzer hat man nun die Möglichkeit einen Spieler zu der Datenbank hinzuzufügen, oder einen Spieler zu entfernen. Möchte man einen Spieler hinzufügen, kann man dies über den Button „Spieler hinzufügen“ tun.

Anschließend öffnet sich das Fenster, in dem man das Profil des zu hinzufügenden Spielers ausfüllen kann (Abbildung 3.4)

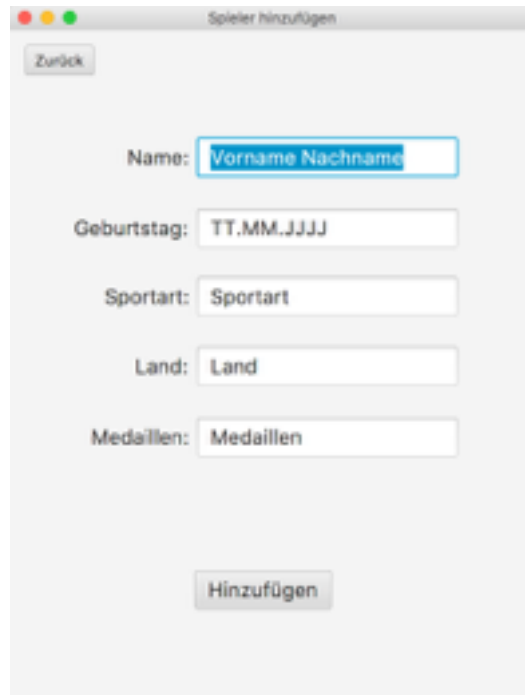


Abbildung 3.4.

Hat man alle Daten ausgefüllt, so wird der Spieler bei einem Klick auf „Hinzufügen“ der Datenbank hinzugefügt und man gelangt zurück zur Spielerdatenbank, in die nun der neue Spieler eingefügt wurde. Durch einen Klick auf „Zurück“ ist es optional auch möglich wieder zur Spielerdatenbank zurückzukehren, falls man sich doch dagegen entschieden hat, einen neuen Spieler hinzuzufügen.

Die Löschoption ist von der Funktionsweise ähnlich aufgebaut. Bei einem Klick auf den Button „Spieler löschen“ öffnet sich ein kleines Eingabefeld (Abbildung 3.5), in welches man den Namen des Spielers eintippen kann, dem man gerne aus der Datenbank entfernen möchte.

Mit dem „Löschen“-Button bestätigt man anschließend seine Eingabe und kehrt zur Spielerdatenbank zurück, welche man jedoch nun über den „Update“-Button aktualisieren muss (Abbildung 3.1).



Abbildung 3.5. Das Fenster zur Löschung eines Spielers.

Die Funktion sich ein Spielerprofil genauer anzuschauen, ist mit der Suchfunktion gekoppelt. Das bedeutet, dass ein positiver Treffer in der Suche den „Profil anzeigen“-Button sichtbar macht, womit man sich anschließend das Profil des Spielers in einem Extra-Fenster ansehen kann (Abbildung 3.6).



Abbildung 3.6. Die Profildfunktion.

Die nächste Funktion führt zum vierten und letzten Element des Programms, dem Medaillenspiegel, welcher in seiner Erscheinung simpler erscheint, als er programmiertechnisch aufgebaut werden musste. Mit dem „Medaillenspiegel“-Button wird dieser mit der standardmäßig voreingestellten Anzeige *Land* aufgerufen (Abbildung 4.1).



Nr.	Land	Medaillen
1	Vereinigte Staaten	121
2	Großbritannien	67
3	China	70
4	Russland	56
5	Deutschland	42
6	Japan	41
7	Frankreich	42
8	Südkorea	21
9	Italien	28

Anzeigen: ☒ Land ☐ Sportart ☐ Sportler

Land... Suchen

Zurück

Abbildung 4.1. Element vier, der Medaillenspiegel mit der Voreinstellung „Land“.

Angezeigt werden bei der Einstellung *Land* die Länder mit den meisten gewonnenen Medaillen, absteigend sortiert. Natürlich bietet sich dem Nutzer an dieser Stelle wieder die Option, die Tabelle nach den einzelnen Spalten zu sortieren.

Durch das Suchfeld hat man - wie auch schon zuvor bei der Spielerdatenbank - die Möglichkeit, je nach angezeigter Tabelle, nach einem Land, einer Sportart oder einem Sportler zu suchen. Die Tabellen lassen sich durch die drei RadioButton „Land“, „Sportart“ und „Sportler“ auswählen (Abbildungen 4.1, 4.2, 4.3).



Nr.	Sportart	Medallen
1	Basketball	6
2	Beachvolleyball	6
3	Fußball	6
4	Handball	6
5	Ring	47
6	Tennis	15
7	Leichtathletik	138
8	Turnen	42
9	Schwimmen	98

Anzeigen: ☐ Land ☒ Sportart ☐ Sportler

Zurück Suchen

Abbildung 4.2. Die Einstellung „Sportart“.

Nr.	Sportler	Medallen
1	Michael Phelps	6
2	Katie Ledecky	5
3	Simone Biles	5
4	Katinka Hosszu	4
5	Jason Kenny	3
6	Danuta Kosak	3
7	Ryan Murphy	3
8	Usain Bolt	3
9	Simone Manuel	4

Anzeigen: ☐ Land ☐ Sportart ☒ Sportler

Zurück Suchen

Abbildung 4.3. Die Einstellung „Sportler“.

Auch hier wird wieder über ein Label angezeigt, an welcher Stelle ein positiver Treffer gelistet ist und auf einen negativen Treffer aufmerksam gemacht (Abbildungen 4.4, 4.5, 4.6, 4.7, 4.8, 4.9).

Nr.	Land	Medallen
1	Vereinigte Staaten	121
2	Großbritannien	67
3	China	70
4	Russland	56
5	Deutschland	42
6	Japan	41
7	Frankreich	42
8	Südkorea	21
9	Italien	28

Anzeigen: ☒ Land ☐ Sportart ☐ Sportler

Japan, Nr. 6

Zurück Suchen

Abbildung 4.4. Einstellung „Land“.

Nr.	Land	Medallen
1	Vereinigte Staaten	121
2	Großbritannien	67
3	China	70
4	Russland	56
5	Deutschland	42
6	Japan	41
7	Frankreich	42
8	Südkorea	21
9	Italien	28

Anzeigen: ☒ Land ☐ Sportart ☐ Sportler

Land nicht enthalten!

Zurück Suchen

Abbildung 4.5. Einstellung „Land“.

Nr.	Sportart	Medallen
1	Basketball	6
2	Beachvolleyball	6
3	Fußball	6
4	Handball	6
5	Ring	47
6	Tennis	15
7	Leichtathletik	138
8	Turnen	42
9	Schwimmen	98

Anzeigen: ☐ Land ☒ Sportart ☐ Sportler

Turnen, Nr. 8

Zurück Suchen

Abbildung 4.6. Einstellung „Sportart“.

Nr.	Sportart	Medallen
1	Basketball	6
2	Beachvolleyball	6
3	Fußball	6
4	Handball	6
5	Ring	47
6	Tennis	15
7	Leichtathletik	138
8	Turnen	42
9	Schwimmen	98

Anzeigen: ☐ Land ☒ Sportart ☐ Sportler

Sportart nicht enthalten!

Zurück Suchen

Abbildung 4.7. Einstellung „Sportart“.



Abbildung 4.8. Einstellung „Sportler“.



Abbildung 4.9. Einstellung „Sportler“.

Mithilfe des „Zurück“-Buttons gelangt man wieder in die Spielerdatenbank.

Hat man seine Arbeit mit der Datenbank abgeschlossen, kann man schlussendlich mithilfe des „Startseite“-Buttons wieder zur Startseite zurückkehren.

## Die Spielerdatenbank als Gast

Betrifft man die Spielerdatenbank über das Login-Element als Gast, so fallen die Optionen „Spieler hinzufügen“ und „Spieler löschen“ sowie folglich auch die „Update“-Funktion weg. Alle anderen Möglichkeiten bieten sich dem Gastnutzer weiterhin (Abbildung 5).

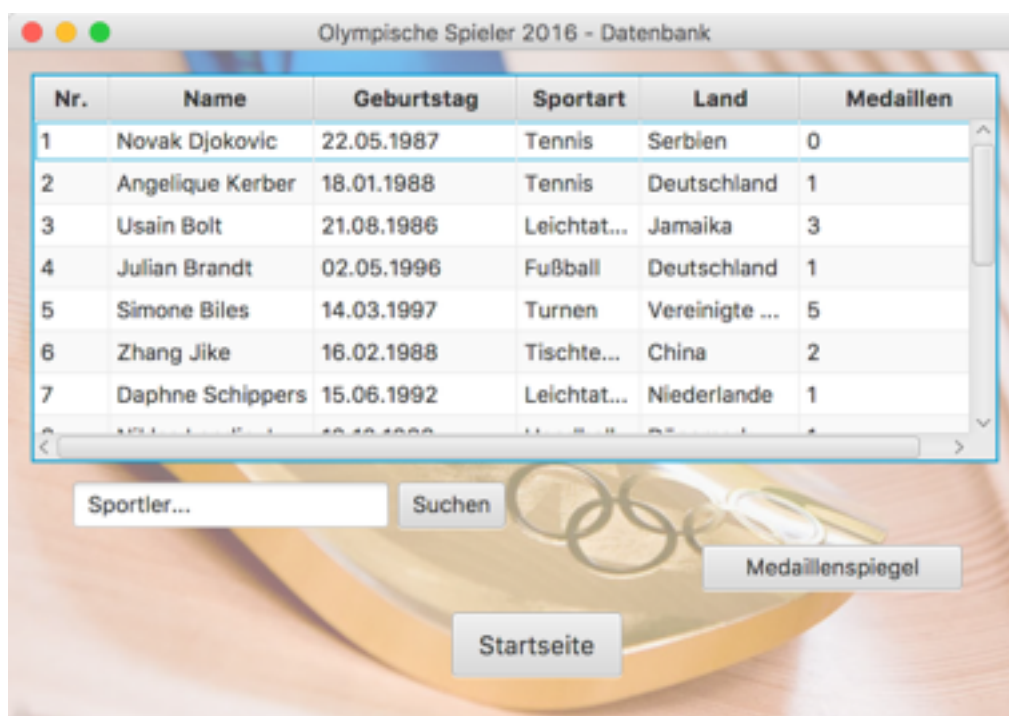
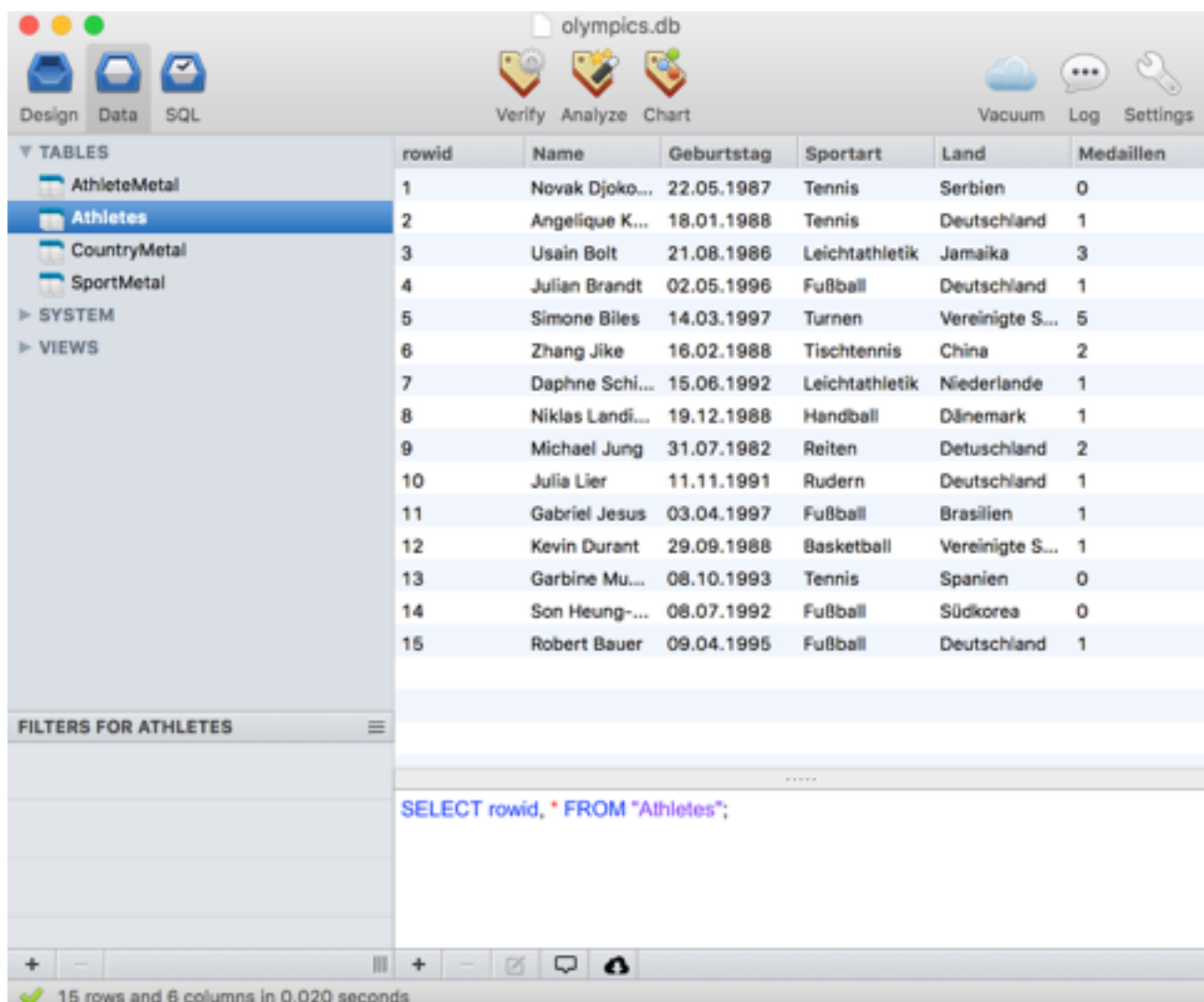


Abbildung 5. Die Spielerdatenbank als Gast.

## Das Programm: Eigenschaften, Werkzeuge und Hilfsmittel

Die grafische Oberfläche, genauer gesagt die Interaktion der Tabelle mit der Datenbank setzt eine gewisse Vorarbeit voraus. Zunächst ist es nötig, die Daten, welche in der Tabelle angezeigt werden, in einer extern angelegten Datei anzulegen und abzusichern.

Dies geschieht über den *SQLiteManager*. Der SQLiteManager bietet einem in der kostenfreien und nicht registrierten Version die Möglichkeit, eine Datenbank anzulegen und bis zu 20 Zeilen zu ihr hinzuzufügen. Abbildung 6 zeigt die Datenbank, welche in der Tabelle erscheint, sobald man sich entweder als autorisierter Nutzer eingeloggt hat, oder als Gast die olympische Spielerdatenbank betritt.



rowid	Name	Geburtstag	Sportart	Land	Medaillen
1	Novak Djoko...	22.05.1987	Tennis	Serbien	0
2	Angelique K...	18.01.1988	Tennis	Deutschland	1
3	Usain Bolt	21.08.1986	Leichtathletik	Jamaika	3
4	Julian Brandt	02.05.1996	Fußball	Deutschland	1
5	Simone Biles	14.03.1997	Turnen	Vereinigte S...	5
6	Zhang Jike	16.02.1988	Tischtennis	China	2
7	Daphne Schi...	15.06.1992	Leichtathletik	Niederlande	1
8	Niklas Landi...	19.12.1988	Handball	Dänemark	1
9	Michael Jung	31.07.1982	Reiten	Detuschland	2
10	Julia Lier	11.11.1991	Rudern	Deutschland	1
11	Gabriel Jesus	03.04.1997	Fußball	Brasilien	1
12	Kevin Durant	29.09.1988	Basketball	Vereinigte S...	1
13	Garbine Mu...	08.10.1993	Tennis	Spanien	0
14	Son Heung-...	08.07.1992	Fußball	Südkorea	0
15	Robert Bauer	09.04.1995	Fußball	Deutschland	1

```
SELECT rowid, * FROM "Athletes";
```

15 rows and 6 columns in 0,020 seconds

Abbildung 6. Der SQLiteManager, die Spielerdatenbank mit Daten.

Die angelegte Datenbank wird anschließend im Projektordner abgespeichert, in meinem Fall unter dem Namen *olympics.db* .

Ohne weiteres kann man innerhalb des Java-Programms jedoch nicht auf die Datenbank zugreifen. Abhilfe schafft an dieser Stelle die *jdbc.jar* - Datei, welche auf [bitbucket.org](https://bitbucket.org/xerial/sqlite-jdbc/downloads) kostenlos zum Download bereitsteht [<https://bitbucket.org/xerial/sqlite-jdbc/downloads>].

Diese ist ebenfalls in den Projektordner zu integrieren, um einen Lese- und Schreibzugriff auf die angelegte Datenbank zu gewährleisten.

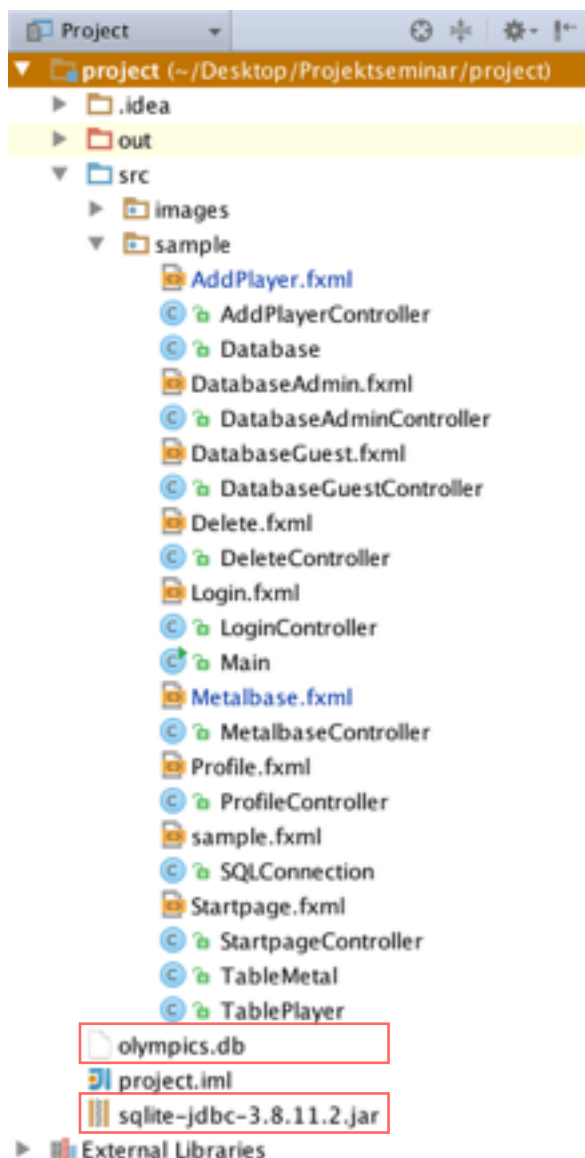


Abbildung 7. Ordnerstruktur - Datenbank und JDBC-Datei.

Innerhalb des Java-Programms war es nun nur noch notwendig, die besagte Verbindung zur Datenbank herzustellen, was über die Database.java Klasse geschieht (Abbildung 8).

```
package sample;

import java.sql.Connection;
import java.sql.DriverManager;

/**
 * Created by denniselfart on 16.08.16.
 */
public class Database {

    public static Connection Connector(){
        try {
            Class.forName("org.sqlite.JDBC");
            Connection conn = DriverManager.getConnection("jdbc:sqlite:olympics.db");
            return conn;
        } catch (Exception e) {
            return null;
        }
    }
}
```

Abbildung 8. Die Database.java Klasse.

Die grafische Oberfläche des Programms ist mit JavaFX programmiert. Eine große Hilfestellung bezüglich der Integration von Datenbanken in SQL und JavaFX haben mir die YouTube-Tutorials von *ProgrammingKnowledge* geleistet.. Durch sie habe ich erfahren, wie man eine Datenbank in Java integriert und welche einzelnen Schritte dafür nötig sind.

Einen großen Vorteil der Oberflächengestaltung mit JavaFX bringt der *javaFX Scene Builder 2.0* mit sich. Mit diesem Tool ist es zumindest design-technisch möglich, Grafiken per Drag and Drop zu gestalten (Abbildung 9). Das Tool schreibt dann die Einstellungen in eine .fxml-Datei.

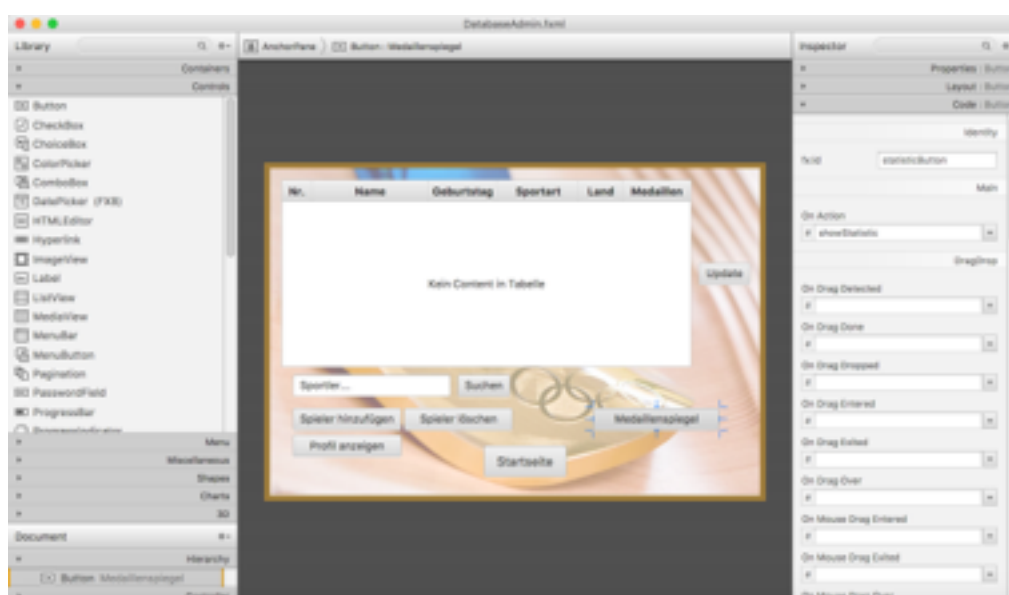


Abbildung 9. Der JavaSceneBuilder 2.0 mit dem Fenster der Spielerdatenbank.

Die rechte Spalte in Abbildung 9 zeigt die Kartei *Code*. Dort muss für jeden einzelnen Button festgelegt werden, welche Methode hinterher aufgerufen werden soll, sollte man auf den Button klicken. Jede erstellte .xml-Datei muss somit einer Controller-Klasse zugeordnet werden, in welcher diese einzelnen Methoden implementiert sind (s. Abbildung 7).

## **Anleitung zur Ausführung / Benutzung**

Das Programm lässt sich ganz einfach ausführen. Nach dem Herunterladen des Projektordners *project* findet man in diesem die ausführbare .jar-Datei Olympia2016.jar. Öffnet man diese, so befindet man sich direkt auf Element eins, der Startseite.

Klickt man nun auf den „Olympische Spieler“-Button, so gelangt man, wie schon im Abschnitt *Projektaufbau* beschrieben, zum Login-Fenster. Die Login-Daten, um die erweiterten Funktionen nutzen zu können sind *Nutzer* als Benutzername und *Zugang* als Passwort.

Das weitere Vorgehen ist selbsterklärend, wobei sich eventuelle Fragen aus dem Abschnitt *Projektaufbau* entnehmen lässt.

## **Reflexion**

Rückblickend betrachtet kann ich sagen, dass ich das mir gesteckte Ziel erreicht habe. Der Zeitplan, welchen ich mir gesteckt hatte, habe ich voll und ganz eingehalten.

Dennoch konnte ich einige Details an dem Projekt nicht so implementieren, wie ich sie mir während des Entwurfes des Projektplans gedacht hatte.

Eine aufwendige Sache, welche ich trotz sorgfältiger Recherche nicht gelöst bekommen habe, ist das Interagieren mit der Tabelle bei der Spielerdatenbank für autorisierte Nutzer.

Das fertige Programm zwingt den Nutzer dazu, durch ein extra erscheinendes Eingabefeld einen einzelnen Spieler zu löschen. Mir persönlich schwebte jedoch der Gedanke vor, den betreffenden Spieler in der Tabelle mit einem Klick zu markieren, was ja auch so prinzipiell möglich ist, und ihn anschließend dann mit dem „Spieler löschen“-Button aus der Datenbank zu entfernen.

Ähnlich verhält es sich mit der Suchfunktion, sowohl im Bereich der Spielerdatenbank als auch im Bereich des Medaillenspiegels. Ursprünglich sollten die Funktionen so implementiert werden, dass im Falle eines positiven Treffers das gefundene Ergebnis in der Tabelle markiert wird. Auch dort habe ich schlussendlich einen Kompromiss mithilfe

eines Labels gemacht, welches dann im Falle eines positiven Treffers Aufschluss über die Zeilennummer des gefundenen Sportlers, Landes oder der gefundenen Sportart gibt.

Eine zusätzliche Tatsache, welche meiner Meinung nach die Lebhaftigkeit des Programms wesentlich einschränkt, ist seitens des SQLiteManagers die Begrenzung der Datenbank auf 20 Zeilen. Somit habe ich nur 15 Spieler in die Spielerdatenbank eingepflegt, damit dem Nutzer noch Freiraum bleibt, zusätzliche Spieler hinzuzufügen. Setzt man eine Datenbank auf, für die ein gewisses Budget gegeben ist, so finde ich die Investition in die kostenpflichtige und registrierte Version des SQLiteManagers sehr sinnvoll.

## **Fazit**

Durch das Projektseminar I4 *Angewandte Informationswissenschaft* hatte ich die Möglichkeit, mir die Programmierung mit Java insbesondere JavaFX einen großen Schritt näher zu bringen. Große Vorteile hat der Zeitrahmen gebracht, in dem das Projekt abgeleistet werden musste. Mir ist aufgefallen, dass ich selbst sehr motiviert war, aufgrund der Tatsache, dass man sich das Projekt selbst ausdenken und planen durfte bzw. musste, was nicht zuletzt auch daran lag, dass mir die Arbeit mit JavaFX und Datenbanken sehr viel Spaß macht.

## **Literatur**

### **Bildquellen**

- Logo Olympia 2016. *Wikipedia*. Abgerufen von [https://upload.wikimedia.org/wikipedia/de/thumb/4/4f/Olympia\\_2016\\_-\\_Rio\\_\(2\).svg/2000px-Olympia\\_2016\\_-\\_Rio\\_\(2\).svg.png](https://upload.wikimedia.org/wikipedia/de/thumb/4/4f/Olympia_2016_-_Rio_(2).svg/2000px-Olympia_2016_-_Rio_(2).svg.png)
- Cristo Redentor. *Blogspot*. Abgerufen von <http://4.bp.blogspot.com/-w-CjboH-poM/VhAh7EIUEml/AAAAAABO5c/FtrKxdGZi8A/s1600/CUBILETE1.jpg>
- Olympische Medaille. *Pinterest*. Abgerufen von <https://s-media-cache-ak0.pinimg.com/736x/68/0a/9c/680a9c23beef6c73d2a229e36552b980.jpg>

- Olympische Ringe. *Google+*. Abgerufen von <https://lh5.googleusercontent.com/-pTENG7tPcVg/AAAAAAAAAAI/AAAAAAAAACQ/0buoj0O4u6E/photo.jpg>
- Olympisches Feuer. *Sportimeny*. Abgerufen von [http://www.sportimeny.com/gallery/event\\_2656main.jpg](http://www.sportimeny.com/gallery/event_2656main.jpg)
- Profilicon. *Art-cc*. Abgerufen von [http://art-cc.com/image/cache/default\\_avatar-400x400.jpg](http://art-cc.com/image/cache/default_avatar-400x400.jpg)

### **Youtube-Tutorials**

- ProgrammingKnowledge, (21. Juni 2016). JavaFx Sqlite Database Tutorial. *YouTube*. Abgerufen am 21. September 2016, von <https://www.youtube.com/playlist?list=PLS1QuIW01RibqOzdu1jShN0xah41S6ILP>