

Segelverein

- Dokumentation

Bernhard
Schmid

Index

Aufgabenstellung.....	3 - 5
Beschreibung	3
ER-Diagramm	3
Relationenmodell	4
Java und JDBC	4
Abgabe	5
 Designüberlegung	 6 - 9
Datenbank	6
Java	6 - 9
 Java Methodenüberlegung.....	 10 - 11
Boot & MannschaftsDaten.....	10
Control	10
DB_Connection	10
Main	10
UpdateBoot	10
View_Login	10
View	11
View_AddBoot	11
View_WMP	11
 Aufwandschätzung (nachträglich)	 12
 Quellen	 13

Aufgabenstellung

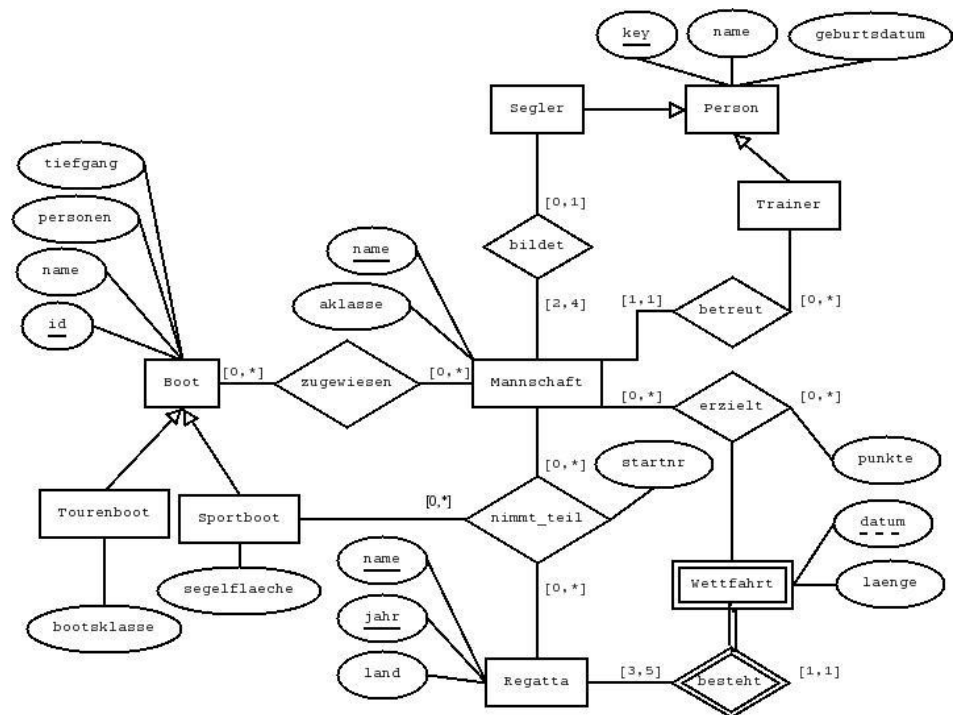
Beschreibung

Für Segler und Trainer sind Name (NAME) und Geburtsdatum (GEBURTSDATUM) bekannt. Sie werden beide identifiziert durch eine eindeutige Nummer (KEY). Mindestens zwei Segler, maximal jedoch vier Segler bilden eine Mannschaft. Für jede Mannschaft wird ein eindeutiger Name (NAME) und eine Altersklasse (AKLASSE) gespeichert. Jede Mannschaft wird genau von einem Trainer betreut. Ein Trainer kann jedoch mehrere Mannschaften betreuen.

Jeder Mannschaft sind Boote zugewiesen. Ein Boot kann mehreren Mannschaften zugewiesen sein. Ein Boot wird eindeutig durch eine Nummer (ID) identifiziert. Weiters sind zu jedem Boot ein Name (NAME), die Anzahl der Personen (PERSONEN) und der Tiefgang (TIEFGANG) bekannt. Es gibt Tourenboote und Sportboote. Tourenboote haben zusätzlich eine Bootsklasse (BOOTSKLASSE) und Sportboote haben zusätzlich eine Segelfläche (SEGELFLAECHE) gespeichert. Es ist außerdem bekannt welche Mannschaften mit welchen Sportbooten an welchen Regatten mit welcher Startnummer (STARTNR) teilgenommen haben.

Eine Regatta wird eindeutig identifiziert durch ihren Namen (NAME) und durch das Jahr (JAHR), in dem sie stattgefunden hat. Das Land (LAND) ist außerdem noch bekannt. Jede Regatta besteht aus mindestens drei jedoch maximal fünf Wettfahrten. Wettfahrten werden durch die zugehörige Regatta und das Datum (DATUM) identifiziert, außerdem wird die Länge (LAENGE) der Strecke gespeichert. Mannschaften können bei jeder Wettfahrt Punkte (PUNKTE) erzielen.

Er-Diagramm



Relationenmodell

Person(key, name, geburtsdatum)

Segler(key: Person.key)

Trainer(key: Person.key)

Boot(id, name, personen, tiefgang)

Tourenboot(id: Boot.id, bootsklasse)

Sportboot(id: Boot.id, segelflaeche)

Mannschaft(name, aklasse, key: Trainer.key)

Regatta(name, jahr, land)

Wettfahrt(name: Regatta.name, jahr: Regatta.jahr, datum, laenge)

bildet(key: Segler.key, name: Mannschaft.name)

zugewiesen(id: Boot.id, name: Mannschaft.name)

nimmt_teil(mname: Mannschaft.name, rname: Regatta.name, rjahr: Regatta.jahr, sportboot: Sportboot.id, startnr)

erzielt(mname: Mannschaft.name, wname: Wettfahrt.name, wjahr: Wettfahrt.jahr, wdatum: Wettfahrt.datum, punkte)

Java & JDBC

Schreiben Sie einen Java Client, der eine JDBC-Verbindung zur Datenbank herstellt und AUTOCOMMIT ausschaltet. Realisieren Sie eine GUI, die einfache CRUD-Befehle auf die Boote des Vereins implementiert (keine explizite SQL-Eingabe). Verwenden Sie dabei auf jeden Fall eine JTable, die auch eine grafische Veränderung der Datensätze erlauben soll.

Als Erweiterung (Bonuspunkte) soll bei der Anzeige der Boote die Möglichkeit der Sortierung und Filterung über ein neues SQL-Kommando bereitgestellt werden. Auch hier soll nicht der Benutzer die SQL-Befehle eingeben, sondern es muss die Funktionalität über entsprechende GUI-Elemente realisiert werden!

Ermöglichen Sie die gleichzeitige Verbindung von mehreren Clients auf die Datenbasis. Implementieren Sie dabei eine transaktionell, gesicherte Erstellung und Änderung von Wettfahrten. Beachten Sie dabei, dass der Punktestand der einzelnen Wettfahrten laufend und von mehreren Clients gleichzeitig aktualisiert werden könnte. Stellen Sie für die Eingabe der Wettfahrt und Mannschaft eine einfache grafische Möglichkeit zur Verfügung.

Abgabe

Die Abgabe ist am 03.05.2015 per E-Mail abzugeben. Es wird ein Protokoll (Metaregeln), die SQL-Files (drop.sql, create.sql, start.sql, insert.sql, insert-*.sql, queries.sql) in einem eigenen Verzeichnis. Dies alles wird in einem ausführbaren JAR-Archiv erwartet. Die Abgabe wird mit einem Prüfungsgespräch validiert, wobei auf eine eigenständige Lösung geachtet wird - kopierte Lösungen führen zu einer negativen Benotung! Quellen sollen somit auf den theoretischen Background und auf die Manuals beschränkt sein. Teile von bestehendem JDBC-Code aus dem Internet und vorgefertigte SQL-Abfragen dürfen somit nicht verwendet werden. Im Zweifelsfall ist es notwendig die Lehrkräfte um Freigabe von Quellen zu bitten. Das Beispiel soll für eine **Postgresql 9.4** Umgebung implementiert werden.

Die Inserts sollen mindestens 10.000 Einträge enthalten. Es bleibt Ihnen überlassen, ob Sie einen selbstgeschriebenen Generator oder ein externes Tool verwenden möchten. Auf jeden Fall muss Ihre Vorgehensweise gut dokumentiert und nachvollziehbar sein. Die Daten sollen so nahe wie möglich der Wirklichkeit entsprechen, um entsprechende Testfälle und Performancetests auf der Datenbank starten zu können.

Bei Problemen mit dem Create-Script und den Inserts kann ein Example-Set bei den Lehrenden angefordert werden. Dies muss aber für jeden Kandidaten einzeln geschehen! Anfragen bitte immer per eMail an BEIDE Lehrer.

Designüberlegung

Datenbank

Bei der Datenbank war bereits ein ERD sowie ein RM vorhanden. Daher waren zum Design der Datenbank keine Überlegungen mehr nötig.

Java

Nach genauem durchlesen der Aufgabenstellung wird schnell klar das der Benutzer über eine GUI verschiedene Arten von SQL – Befehle (CRUD) ausführen können soll, ohne diese selbst eingeben zu müssen. Wir sollten uns vorstellen ein einfacher Mitarbeiter in einem Büro der keine Ahnung von Postgresql oder Java hat, sollte das Programm bedienen können.

Es soll ebenfalls möglich sein das sich mehrere Benutzer „einloggen“ können, was nahe legt das beim Starten des Programmes ein Login Fenster erscheint. Wenn mehrere Benutzer gleichzeitig Daten in der Datenbank bearbeiten können, müssen bei Transaktionen die Daten vor Lost Updates geschützt werden (Es liegt nahe das Problem mit dem setzten einer Isolationsstuf (Serializable) zu lösen)

Nachdem sich der Benutzer (erfolgreich) eingeloggt hat soll er auf die Hauptseite des Programmes kommen welche alle Daten der Table Boote in einer JTable anzeigt. Sowie die Optionen zum adden, updaten, löschen & sortieren von Daten bereitstellen soll.

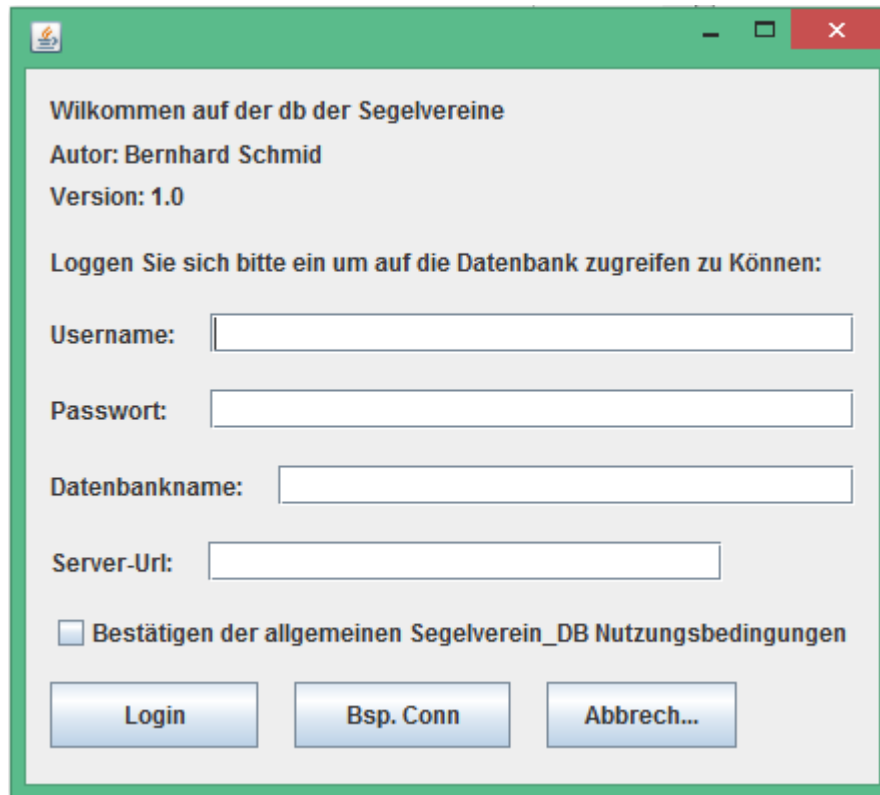
Es soll verhindert werden dass der Benutzer direkt etwas in die JTable hineinschreiben kann (also in die jeweiligen Cells). Wenn möglich soll er dies über eine zusätzliche GUI tun.

Die JTable muss auch eine Sortierfunktion besitzen (Absteigend & Aufsteigend sortieren).

Zusätzlich gibt es einen Hilfe Button dieser erklärt dem Benutzer die Funktionen des Programms und wie er es zu bedienen hat. (Bei Fehlerhaften Eingaben z.B. bei Update, Add oder Delete wird er sowieso auf seinen Fehler aufmerksam gemacht und es werden Lösungsvorschläge präsentiert.)

So sehen die einzelnen GUIs des Programmes aus:

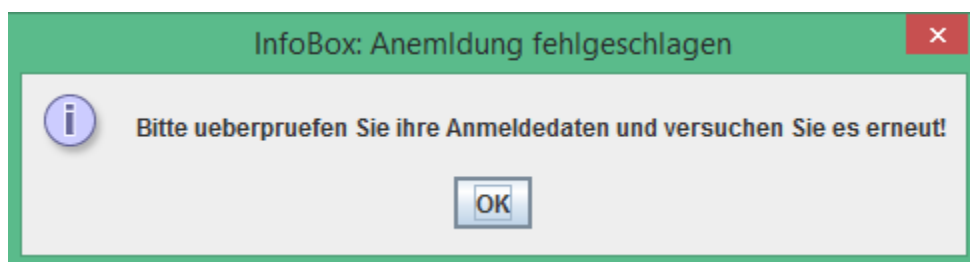
Login:



The login dialog box has a green title bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area is light gray and contains the following text and controls:

- Willkommen auf der db der Segelvereine**
- Autor: Bernhard Schmid**
- Version: 1.0**
- Loggen Sie sich bitte ein um auf die Datenbank zugreifen zu Können:**
- Username:** followed by a text input field.
- Passwort:** followed by a text input field.
- Datenbankname:** followed by a text input field.
- Server-Url:** followed by a text input field.
- A checkbox labeled **Bestätigen der allgemeinen Segelverein_DB Nutzungsbedingungen**.
- Three buttons at the bottom: **Login**, **Bsp. Conn**, and **Abbrech...**.

Fehlermeldung bei falschem Login:



The error message dialog box has a green title bar with the text **InfoBox: Anmeldung fehlgeschlagen** and a close button on the right. The main content area is light gray and contains the following text and controls:

- An information icon (a lowercase 'i' inside a circle) on the left.
- The text **Bitte ueberpruefen Sie ihre Anmeldedaten und versuchen Sie es erneut!**.
- An **OK** button at the bottom center.

Hauptseite des Programms

id	name	personen anz.	tiefgang
282436433	Titanic	21	0.31
287435398	Magdeburg	9	1.48
200298315	Cleopatra	23	0.82
245204542	Bellipotent	24	1.54
188044302	Viperous	3	0.11
232563445	Deutschland	7	1.94
329535268	Dresden	6	0.94
232421173	Venus	18	1.75
138759810	ChopChop	12	0.11
243160241	Bremse	1	0.46
310101160	Tegetthoff	14	0.38
306680324	Cleopatra	8	0.21
11470848	Sea Viper	2	1.19
170993388	Bellipotent	2	0.87
38149394	Indefatigable	10	0.91
113586236	Flower	23	0.03
77090844	Hero	15	1.06
84847685	King	21	1.68
78681953	Monarch	3	1.13

Add

Update

Delete

Zu den Wettfahrten

Hilfe?

Sortierung

ID:

 Name:

 Anz. Personen:

 Tiefgang:

Öffnet die GUI die es erlaubt den Punktestand der Mannschaften zu bearbeiten

Öffnet eine Reihe von Tipps und Lösungsvorschlägen

Die GUI der Update & der Add-Funktion sieht beinahe identisch aus

GUI zum Bearbeiten der Punktestände

Manschaft	Wettfahrt	Jahr	Datum	Punkte
3_Segelverein BSC - Gleisdorf_1978	Siren- Segelmeisterschaft_529694604	1927	1927-04-20	12
4_Segelverein BSC - Gaishorn am See_19...	RAW- Segelmeisterschaft_711844561	2003	2003-08-08	81
5_Segelverein KCH - Eisenstadt_1891	Stink & Sink- Segelmeisterschaft_5989403...	1959	1959-06-17	55
6_Segelverein SCB - Götzendorf an der Le...	Yacht- Segelmeisterschaft_294240715	1984	1984-11-11	83
7_Segelverein LSV - Bairisch Köldorf_1992	Schnell- Segelmeisterschaft_712006202	1983	1983-09-23	76
8_Segelverein BSC - Flachau_1924	Yombolatschoku- Segelmeisterschaft_132...	1955	1955-03-17	19
10_Segelverein LUV - Bad Gastein_1954	Senioren- Segelmeisterschaft_123418553	1953	1953-03-16	24
11_Segelverein SBV - Göllersdorf_1907	Yacht- Segelmeisterschaft_319510222	1967	1967-04-28	54
12_Segelverein BSC - Höchst_1986	Wagner- Segelmeisterschaft_45611273	1989	1989-07-02	96
13_Segelverein LSV - Johnsdorf-Brunn_19...	Wagner- Segelmeisterschaft_231008917	2004	2004-06-29	74
15_Segelverein SSV - St. Pölten_1959	Yombolatschoku- Segelmeisterschaft_390...	1922	1922-09-02	11
16_Segelverein WWV - Artstetten_1958	Fischer- Segelmeisterschaft_709449456	2009	2009-05-09	62
18_Segelverein OSV - Wien_1928	Fischer- Segelmeisterschaft_172707885	2013	2013-02-08	67
20_Segelverein SV - Gaishorn am See_1898	Spitzkopf- Segelmeisterschaft_381713344	2013	2013-05-27	22
21_Segelverein SV - Faistenau_1896	Spitzkopf- Segelmeisterschaft_481244217	2007	2007-07-23	58
22_Segelverein JKWB - Eisenstadt_1932	Lumpi- Segelmeisterschaft_611246243	1941	1941-06-08	49
23_Segelverein KS - Edlitz_1966	Schnellboot- Segelmeisterschaft_5665632...	1983	1983-04-27	66
24_Segelverein KS - Johnsdorf-Brunn_1964	Lumpi- Segelmeisterschaft_941408989	1927	1927-05-02	59
25_Segelverein JKWB - Dinersdorf_2008	Fischer- Segelmeisterschaft_866658062	1932	1932-05-30	70

Punktestand bearbeiten

Sortierung

Beenden

Eingabe

?

Bitte tragen Sie hier den neuen Punktestand ein

OK

Abbrechen

Suchfunktion

In der Suchfunktion ist es moeglich die Daten absteigend oder Aufsteigend (nach aktuellem Punktestand) zu sortieren__ASC = Aufsteigend__DESC = Absteigend

ASC

DESC

Abbrechen

Java Methodenüberlegung (genauere Beschreibung im Programm selbst)

Main

Die Main-Methode erzeugt einfach ein neues Objekt vom Controller um das Programm zu starten

Control

Die Klasse-Control erzeugt eine neue View_Login über die sich der Benutzer einloggen kann

DB_Connection

DB_Connection ist für die Verarbeitung der SQL-Queries zuständig (sowie das Erstellen der Datenbank Verbindung)

Boot & MannschaftsDaten

Sowohl Boot als auch MannschaftsDaten lesen die Daten aus der Datenbank aus, speichern diese in einen 2D-Array und übergeben Sie der jeweiligen View, je nachdem wer die Daten braucht. (Der 2D-Array wird benötigt um die Daten in die JTable einzulesen)

UpdateBoot

Die Klasse UpdateBoot ist für das aktualisieren der Einträge sowohl in der Datenbank als auch in der JTable verantwortlich. Es werden aus Textfeldern die Daten ausgelesen welche in der Datenbank gespeichert werden sollen. Entsprechen die eingegebenen Datentypen nicht den erwarteten Datentypen (NumberFormatException) bekommt der Benutzer eine Fehlermeldung und Verbesserungsvorschläge. Es ist auch nicht gestattet nichts einzugeben.

View_Login

Die Login View liest die eingegebenen Daten aus den Textfeldern aus und übergibt sie mittels Getter und Setter Methoden der DB_Connection. Bei inkorrekt Eingabe der Daten wird ebenfalls eine Fehlermeldung angezeigt.

View

Die View (Hauptview) kann man als Schaltzentrale des gesamten Programms betrachten. Sie verweist auf alle Methoden die zum Durchführen der einzelnen Operationen benötigt werden. (Z.B. Update -> UpdateBoot())

View_WMP

Die View_WMP ist eigentlich die Hauptview nur mit weniger Funktionen. Es ist nicht möglich Daten zu löschen oder hinzuzufügen sondern es ist nur gestattet bereits vorhandene Punktestände zu bearbeiten (zumindest hab es ich so verstanden)

View_AddBoot

Diese View arbeitet genauso wie UpdateBoot der einzige Unterschied liegt darin, dass AddBoot einen INSERT – Befehl ausführt und Update Boot einen UPDATE....

Aufwandsschätzung:

Aufgaben	Geschätzt	Real
CREATE.sql	~ 60 min	~ 25 min
DROP.sql	~ 10 min	~ 5 min
START.sql	~ 10 min	~ 5 min
INSERTS_*.sql	~ 120 min	~ 240 min
JDBC	~ 1800 min (30 st)	~ 3120 min (52 st)
Gesamt	~ 2000 min -> 34 st	~ 3395 min -> 57 st

Ich habe meine alte Aufwandschätzung doch noch auf meinem Rechner gefunden. Diese liegt aber im Vergleich zu der Zeit die ich wirklich zur Durchführung der Aufgabe gebraucht habe deutlich daneben. Ich habe fast das Doppelte der Zeit gebraucht als ich eigentlich geplant hatte.

Quellen:

<https://elearning.tgm.ac.at/mod/assign/view.php?id=32736>

<http://www.postgresql.org/files/documentation/pdf/9.4/postgresql-9.4-A4.pdf>

<http://docs.oracle.com/javase/7/docs/api/>

<http://docs.oracle.com/javase/7/docs/api/javax/swing/JTable.html>

<http://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>

<http://docs.oracle.com/javase/7/docs/api/javax/swing/table/AbstractTableModel.html>

<http://downloads.mysql.com/docs/refman-5.7-en.pdf>

Ich weiß das Professoren oder Mitschüler als Quellen anzugeben nicht ganz dem Sinn einer Quellangabe entsprechen dennoch:

Daniel Melichar

Adaresh Soni

Prof. Michael Borko

Prof. Erhard List