

Datenbank-Programmierung

Prof. Dr. Jörg Hettel
Hochschule Kaiserslautern
Studiengang IT-Analyst
– Version: 21. Mai 2020–

SPEZIFIKATION ZUR PROJEKTARBEIT

Inhaltsverzeichnis

1	Vorbemerkung	1
1.1	Technologien	1
2	Die Web-Anwendung	2
3	Projektaufgabe	3
3.1	Entwurf eines geeigneten Datenbank-Schemas	3
3.2	Populations-Tool	3
3.3	Datenbank-Anbindung der Web-Anwendung	3
4	Abgabe	4

1 Vorbemerkung

Ziel der Projektarbeit ist die Anwendung der in dem Modul besprochenen Persistenz-Technologien. Im Rahmen der Projektarbeit soll ausgehend von einer vorgegebenen Web-Anwendung der Zugriff auf eine Datenbank implementiert werden. Im Vordergrund steht die Persistenz-Funktionalität und nicht das Design der Oberfläche.

1.1 Technologien

Folgende Technologien sollen eingesetzt werden:

- **JDBC:** Ein einfaches Java-Programm, das für den Datenbankzugriff JDBC benutzt
- **JPA:** Die Webanwendung benutzt für den Datenbankzugriff JPA unter Verwendung einer DataSource.
- **JPA-EclipseLink:** Als JPA-Implementierung soll EclipseLink verwendet werden.
- **mySQL:** Als Datenbank soll eine mySQL verwendet werden.

2 Die Web-Anwendung

Auf der OLAT-Web-Seite finden Sie eine kleine Web-Anwendung, die einen kleinen Book-Shop realisiert. Sie kennen die Anwendung vom Modul Web-Programmierung. Die Anwendung funktioniert ohne Datenbank. Die Daten zu den Büchern werden beim Start der Anwendung dynamisch erzeugt und die Daten der Benutzer nur transient verwaltet.

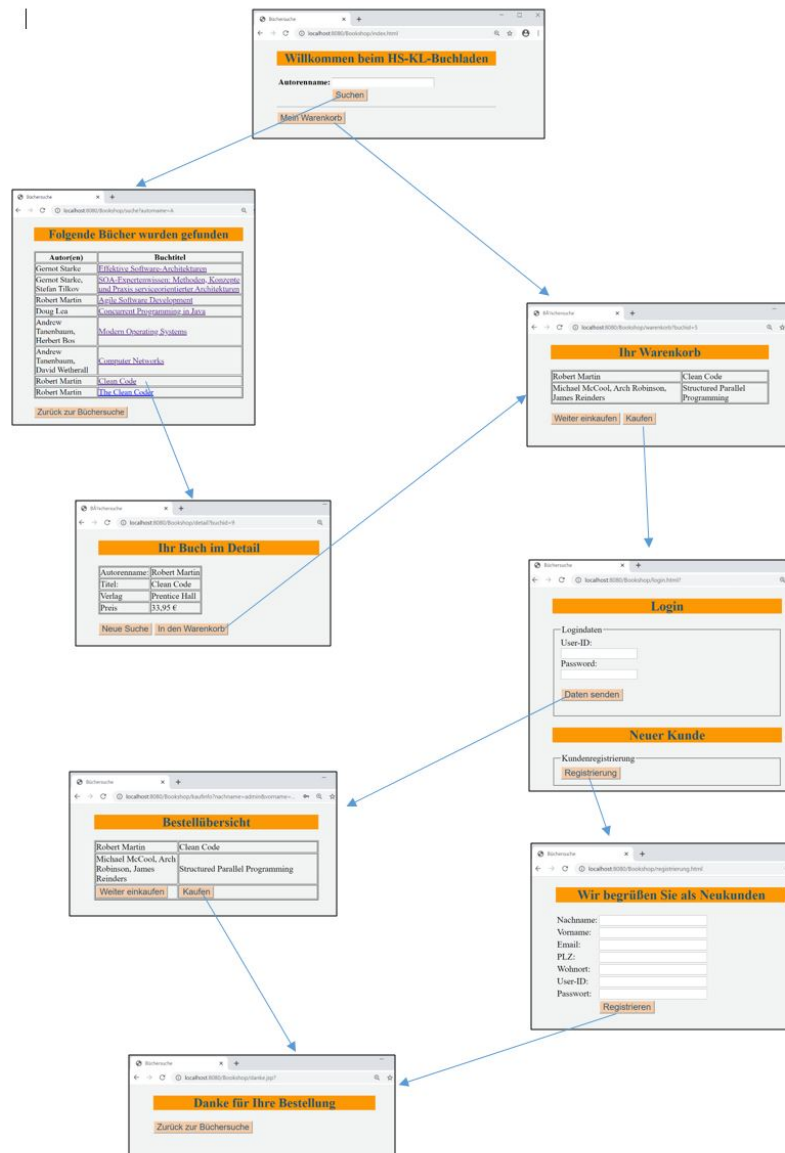


Abbildung 1: Ablauf der Web-Anwendung.

3 Projektaufgabe

Die Projektaufgabe besteht darin, die Web-Anwendung um eine „Persistenz“ zu erweitern. In einer Datenbank sollen dabei folgende Entitäten abgelegt bzw. verwaltet werden:

- Bücher und Autoren
- Verlage (neu)
- Benutzer (neu)
- Bestellungen (neu)

Folgende Teilaufgaben sind dabei zu bearbeiten:

3.1 Entwurf eines geeigneten Datenbank-Schemas

Entwerfen Sie ein für die Anwendung geeignetes Datenbank-Schema. Überlegen Sie sich hierbei, wie Sie insbesondere mit den Bilddateien umgehen. Hier gibt es verschiedene Möglichkeiten, die alle ihre Vor- und Nachteile besitzen. Beachten Sie bitte auch, dass Passwörter nicht im Klartext in der Datenbank stehen sollen. Weiter sollten Passwörter möglichst nicht im Klartext abgelegt werden. Das Schema sollte die oben aufgeführten Entitäten widerspiegeln.

3.2 Populations-Tool

In der Vorlage werden bei der Initialisierung der `ModelFacade` die Buch- und Autoren-Objekte erzeugt. Implementieren Sie ein einfaches Import-Tool (Java-Konsolen-Programm), das die Datenbank initial mit den vorgegeben Buch- und Autor-Daten befüllt. Das Import-Tool kann hierbei auch das Schema anlegen. Die Implementierung des Import-Tool soll die gewöhnliche JDBC-Schnittstelle benutzen.

Das Tool wird nun einmalig zum Datenimport benutzt. Die Daten können dabei „hart codiert“ im Java-Code stehen (analog zur `ModelFacade`).

3.3 Datenbank-Anbindung der Web-Anwendung

Die Web-Anwendung soll nun so geändert werden, dass alle Daten (Buch, Autor, Verlag, Benutzer, Bestellung) bei Suchanfragen aus der Datenbank gelesen werden. Für den Datenbankzugriff soll innerhalb der Web-Anwendung das JPA-Framework benutzt werden.

Beachten Sie, dass auch registrierte Benutzer und deren Bestellungen in der Datenbank persistiert werden. Erweitern Sie die Anwendung nun auch so, dass auf der „Danke“-Seite, die nach einem Kauf gezeigt wird, alle bisherigen Bestellungen des Benutzers angezeigt werden (vgl. Abb. 2). Es genügt, das Bestelldatum und die bestellten Buchtitel anzuzeigen.



Abbildung 2: Alle Bestellungen eines benutzers.

4 Abgabe

Abzugeben ist ein sinnvoll dokumentierter Anwendungscode. Eine explizite Projektdokumentation ist nicht erforderlich, kann aber auch mitgeliefert werden. Falls Sie auch einen Standard-User zu Beginn erzeugen, können Sie den Benutzernamen und das Passwort in einer Readme-Datei hinterlegen, die Sie ins WebContent-Verzeichnis legen.

Im Einzelnen sind abzugeben

1. Das SQL-Skript zur Erzeugung des Datenbankschemas (mySQL-kompatibel). Das Skript können Sie z.B. auch im Projektverzeichnis mit dem Namen dbScripts ablegen. Vermerken Sie das dann bitte in der Readme-Datei.
2. Das Java-Programm zum initialen Befüllen der Datenbank (Sourcecode). Die können, das Java-Programm auch direkt zur Web-Anwendung mit dazu packen. Vermerken Sie das dann bitte auch in der Readme-Datei.
3. Die eigentliche Web-Anwendung in Form einer war-Datei oder als Maven-Projekt (mit enthaltenem Sourcecode und eventuell den anderen Artefakten)
4. Die Konfigurationsdateien (server.xml und context.xml) des Tomcat-Servers

Packen Sie bitte alle Artefakte in eine Zip-Datei. Benutzen Sie hierfür bitte folgende Namenskonvention:

DBProg_<<Nachname>>_<<Vorname>>.zip

Wichtig: Achten Sie darauf, dass jeweils auch die Java-Sourcen enthalten sind. Die war-Datei muss lauffähig sein, d.h. sie darf keine Abhängigkeiten zu externen Bibliotheken oder ähnlichem besitzen.

Die Abgabe erfolgt über OLAT. Sie finden einen entsprechenden Abgabeordner auf der OLAT-Seite zum Modul.