

Hausübung 03 5XHIF 26.02.2015

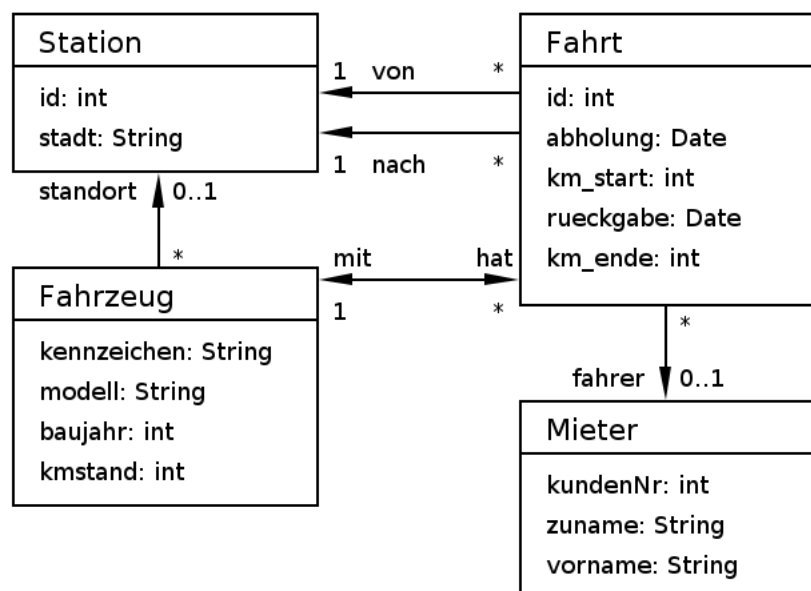
Problemstellung

Die Deluxe-Car4Rent-GesmbH möchte eine Webapplikation entwickeln, in der die Daten zu den einzelnen Stationen, den Mietern, den Fahrzeugen und deren Fahrten erfasst und dargestellt werden können.

Ihre Aufgabe ist es, unter Berücksichtigung des gegebenen UML-Diagramms und der definierten Geschäftsregeln einen Prototypen nach folgenden Vorgaben zu entwickeln:

1. Die Webapplikation ist mit Hilfe von Java EE unter Verwendung von JSF 2.0 zu entwickeln. Dabei ist die Facelet-Technologie zu verwenden, als externe Komponentenbibliothek werden Primefaces empfohlen.
2. Datenbankzugriffe sind mit Hibernate zu realisieren, als DBMS ist PostgreSQL zu verwenden.

UML-Diagramm



Geschäftsregeln

- Ein Mieter kann ein Fahrzeug an einer Vermietstation abholen und an einer anderen wieder abgeben (repräsentiert durch die Assoziationen von und nach).
- Einer noch nicht abgeschlossenen Fahrt ist noch keine Station über die nach-Beziehung zugeordnet.
- Ein Fahrzeug, das gerade unterwegs ist (also einer noch nicht abgeschlossenen Fahrt zugeordnet ist), hat keinen Standort an einer der Vermietstationen.
- Jeder Fahrt muss ein Mieter zugeordnet sein.
- Fahrten stellen eine Historisierung von Vorgängen dar, daher dürfen die initial getroffenen Zuordnungen zu Abfahrtsort, Mieter und Fahrzeug nicht nachträglich geändert werden.

Aufgaben

Aufgabe 1 - Geschäftsmodell und Testdaten

Erstellen Sie ein geeignetes Entitymodell und entwickeln Sie zugehörige Geschäftsklassen mit Hibernate - Annotations. In der Entity Fahrzeug ist das Kennzeichen als ID zu verwenden, Mieter erhalten die in der gegebenen XML-Datei definierten ID's. Alle anderen Primarykeys sind automatisch zu erzeugen.

Zusätzlich zu den durch das gegebene UML-Diagramm beschriebenen Klassen sind in der Datenbank Logindaten (Username und Passwort) zu wenigstens 3 Mitarbeitern zu erfassen. Für diesen Prototyp darf das Passwort im Klartext in der Datenbank gespeichert sein.

Schreiben Sie eine Konsole Applikation, welche aus dem erstellten Entitymodell mit Hilfe eines Schemaexportes eine Datenbank erzeugt und die jede Tabelle mit wenigstens 3 Datensätzen füllt. Bei den Fahrten ist auch eine nicht abgeschlossene Fahrt in den Testdaten vorzusehen.

Aufgabe 2 - Webinterface

Entwickeln Sie mit Hilfe von JavaServerFaces und Primefaces eine auf Facelets basierende Webapplikation mit folgender Funktionalität:

- **Loginseite**
Mit Hilfe dieses Formulars kann sich ein Mitarbeiter anmelden. Stellen Sie über ein Cookie sicher, dass der Loginname vorausgefüllt ist, wenn ein User innerhalb von 30

Tagen vom gleichen Client aus das Formular aufruft. Nach erfolgreichem Login ist das Timeout der HttpSession auf 5 Minuten zu setzen. Nach Verlust der Session soll wieder auf die Loginseite weitergeleitet werden.

- **Neue Fahrt erfassen**

Nach Auswahl einer Station werden alle an dieser Station zur Verfügung stehenden Fahrzeuge in einer Dropdownbox angezeigt. Das Abholdatum wird mit dem aktuellen Datum vorbelegt, kann aber auch verändert werden. Mit Hilfe eines geeigneten Steuerelements kann ein in der Datenbank gespeicherter Mieter ausgewählt werden. Der aktuelle km-Stand des Fahrzeuges wird übernommen.

Bevor der neue Datensatz in der Datenbank gespeichert wird, werden alle eingegebenen Daten noch einmal angezeigt und der User kann den Vorgang abbrechen.

- **Fahrt abschließen**

Alle nicht abgeschlossenen Fahrten werden in einem geeigneten Control angezeigt und der User kann eine Fahrt selektieren. Nach Eingabe des Rückgabedatums und des aktuellen Kilometerstandes kann die Fahrt abgeschlossen werden. Der aktuelle Kilometerstand ist in das entsprechende Fahrzeug zu übernehmen.

- **Alle Fahrten zu einem Fahrzeug anzeigen**

Alle Fahrzeuge werden in einem geeigneten Steuerelement dargestellt. Nach Auswahl eines Fahrzeuges werden alle Fahrten zu diesem Fahrzeug in Tabellenform dargestellt. Außerdem ist die durchschnittliche Kilometerleistung pro Fahrt zu berechnen und anzuzeigen.

Auf dieser Seite ist auch ein Download aller zum aktuell ausgewählten Fahrzeug gespeicherten Fahrten im XML-Format (vgl. Aufgabe 3) anzubieten. Darüber hinaus sollen diese Informationen auch als PDF-Download angeboten werden.

Aufgabe 3 - XML-Export

Implementieren Sie für den in Aufgabe 2 beschriebenen XML-Download eine Methode, welche die Daten aller zu einem bestimmten Fahrzeug gespeicherten und abgeschlossenen Fahrten in eine XML-Datei mit folgendem Aufbau exportiert:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fahrten SYSTEM "fahrten.dtd">
<fahrten kennzeichen="P-349XY">
  <fahrt>
    <mieter id="123">
      <name>Huber Franz</name>
    </mieter>
    <start datum="03.03.2012" km="3212" />
    <ende datum="05.03.2012" km="3518" />
  </fahrt>
</fahrten>
```

```
</fahrt>  
<!-- weitere Elemente fahrt -->  
</fahrten>
```

Erstellen Sie auch eine zugehörige Document Type Definition, die das oben angegebene XML-Dokument möglichst exakt beschreibt.

Allgemeine Hinweise

Die Lösungen der Aufgaben sind wie folgt auszuführen:

- Erstellen Sie ein handschriftliches Konzept, aus dem eine strukturierte Planung der Lösung ersichtlich ist. Im Speziellen sind das Datenmodell, UML-Diagramme zu den Controllern und eine grafische Übersicht der zur Lösung der Aufgabe 2 geplanten Webkomponenten inklusive Navigationsregeln anzugeben.
- Beschreiben Sie die Funktionalität Ihrer Lösungen. Bei fehlerfrei arbeitenden Lösungen sind Screenshots mit einer kurzen Funktionsbeschreibung zu erstellen, im Fehlerfall sind Beschreibungen der Fehler und der daraus resultierenden Funktionalitätsdefizite anzugeben.