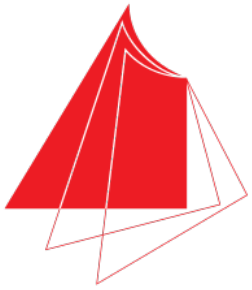

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Dokumentation

Verteilte Systeme Master Labor

Pol Zeimet (65834)
zepo1012@hs-karlsruhe.de

Yannick Stephan (65934)
stya1012@hs-karlsruhe.de

Aufgabe 1

29. Oktober 2019

1 Beschreibung

Das zu analysierende Projekt beschreibt einen über den Browser erreichbaren Web-Shop. Der Webshop stellt mehrere Funktionen zur Verfügung, welche in diesem Abschnitt beschrieben werden.

- Registrieren und Anmelden
  Beim ersten Besuch des Shops muss ein eigener Account mit Namen und Passwort erstellt werden. Bei weiteren Besuchen werden diese zum Login verwendet.
- Anlegen und Löschen Kategorien
  Im Webshop ist es dem Nutzer möglich, Produktkategorien anzulegen und zu löschen.
- Produkte erstellen und löschen
  Nutzer können Produkte erstellen und sie einer vorher definierten Kategorie hinzufügen. Erstellte Produkte können in der gleichen Übersicht auch gelöscht werden.
- Nach Produkten suchen
  Produkte können nach ihren angegebenen Attributen Preis, Name und Kategorie gefiltert und abgefragt werden.

2 Aufbau

2.1 MVC

Das Projekt ist nach dem MVC-Prinzip aufgebaut.

2.1.1 View

Die View besteht aus .jsp Dateien. Diese enthalten den Aufbau der Seite, sowie die entsprechenden auszuführenden Aktionen beim Bedienen der Schaltflächen. Über Struts sind die in der View beschriebenen Aktionen mit dem Controller verbunden. Bei erfolgreicher Aktion wir die in der struts.xml hinterlegte Darstellung angezeigt.

2.1.2 Controller

Der Controller besteht aus Aktionsklassen, die jeweils mit über Struts von der View aus aufgerufen werden. Die Aktionsklassen prüfen die übergebenen Parameter auf ihre Gültigkeit sowie die Berechtigungen den aktuellen Session-

Kontext und somit den Nutzer und rufen dann die entsprechende Businesslogik aus dem Model auf, die zum ausführen der Aufgabe benötigt ist. Bei erfolgreicher Ausführung wird eine Bestätigung zurückgegeben.

2.1.3 Model

Das Model, also die zugrundeliegende Businesslogik, ist in den Manager-Klassen implementiert. diese Klassen enthalten die Methoden, die zum ausführen der gewünschten Nutzeraktionen benötigt werden. Dies beinhaltet Datenbankzugriffe und die Verarbeitung von Suchbegriffen. Die Tatsächlichen Datenbankzugriffe werden dabei nicht durch den Manager selber, sondern durch ein Data Access Objekt, kurz DAO durchgeführt.

2.2 Objekte

Der allgemeine Aufbau des Quellcodes sowie die Abhängigkeiten der verschiedenen Objekte lassen sich gut anhand eines UML Diagrammes darstellen. Der Übersicht halber werden wir uns eine einzelne Aktion ansehen, da alle Aktionen die gleiche Struktur verfolgen.
So haben wir unsere Aktionsklasse, welche über eine execute() Funktion die Businesslogik des Models aufruft. Eine Aktionsklassen ist hierbei eine Erweiterte *Actionsupport* Klasse, Manager Klassen implementieren ein ihrer Rolle entsprechendes Manager Interface. Da es in diesem Fall um das hinzufügen einer neuen Kategorie geht, sind die Aktions- sowie die Manager-Klasse mit der Kategorie-Klasse verknüpft. Die Validierung der Eingabe übernimmt ebenfalls die Aktionsklasse. Das ablegen der Kategorie in, sowie das Laden der aktualisierten Kategorie-Liste aus der Datenbank erfolgen über ein Data Access Objekt. Dieses baut auf einem HybernateDAO auf. Jedes Objekt, das in die Datenbank abgespeichert werden soll, verfügt über ein eigenes DAO. Hybernate erlaubt es uns hierbei, sogenannte „POJOs“, „Plain old Java objects“, also normale Java Objekte über Annotationen als Tabellen anzulegen und mit den entsprechenden Objekt-Variablen zu füllen.

3 Ablauf

Der Ablauf der beschriebenen Funktionen aus Kapitel 1 kann in Aktivitäts– und Klassendiagrammen zusammengefasst werden:

3.1 Registrieren und Login

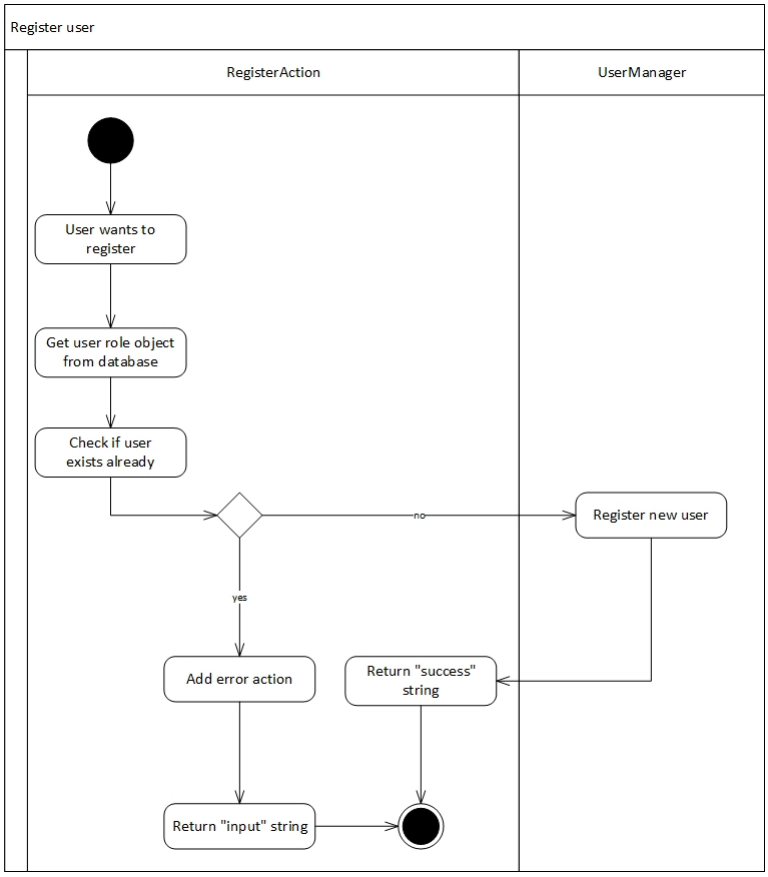

Abbildung 1: Aktivitätsdiagramm für das Registrieren eines neuen Benutzers

3.2 Anlegen und Löschen Kategorien

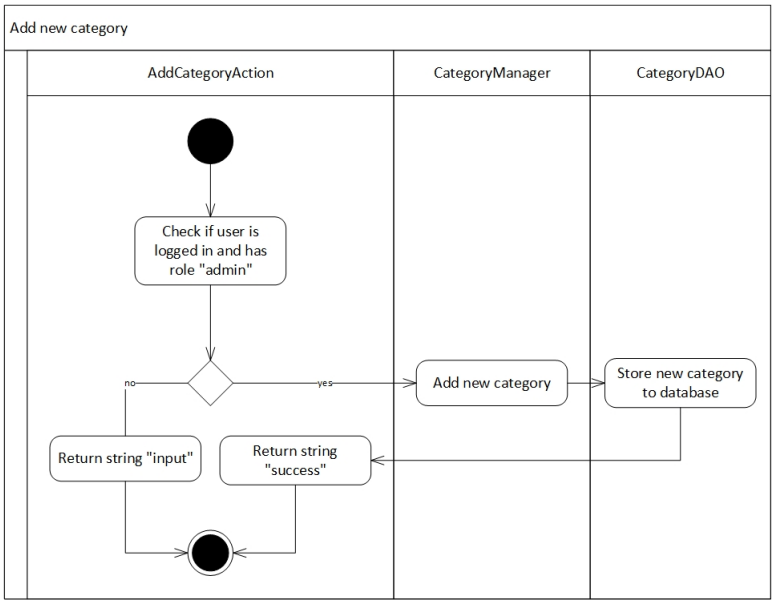

Abbildung 2: Aktivitätsdiagramm für das Hinzufügen neuer Kategorien

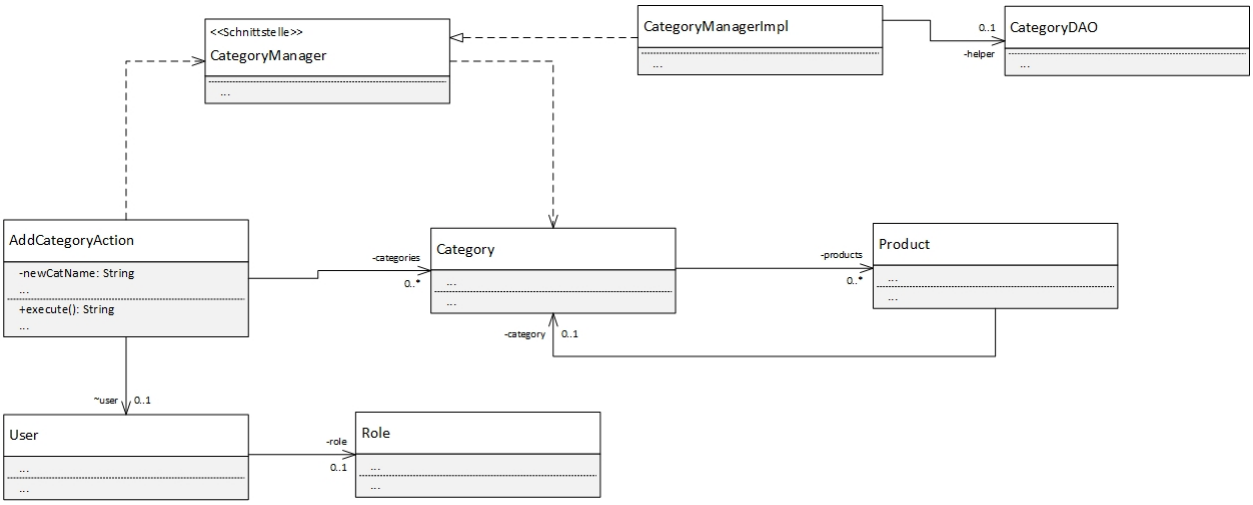

Abbildung 3: Klassendiagramm der *AddCategoryAction*

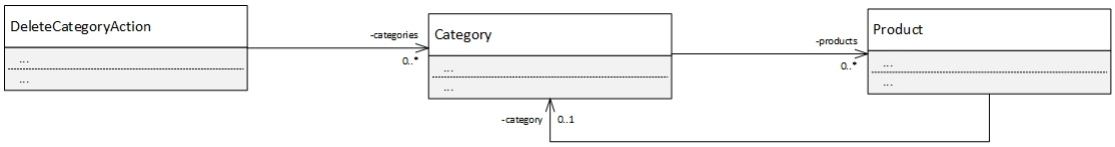

Abbildung 4: Klassendiagramm der *DeleteCategoryAction*

3.3 Produkt erstellen und löschen

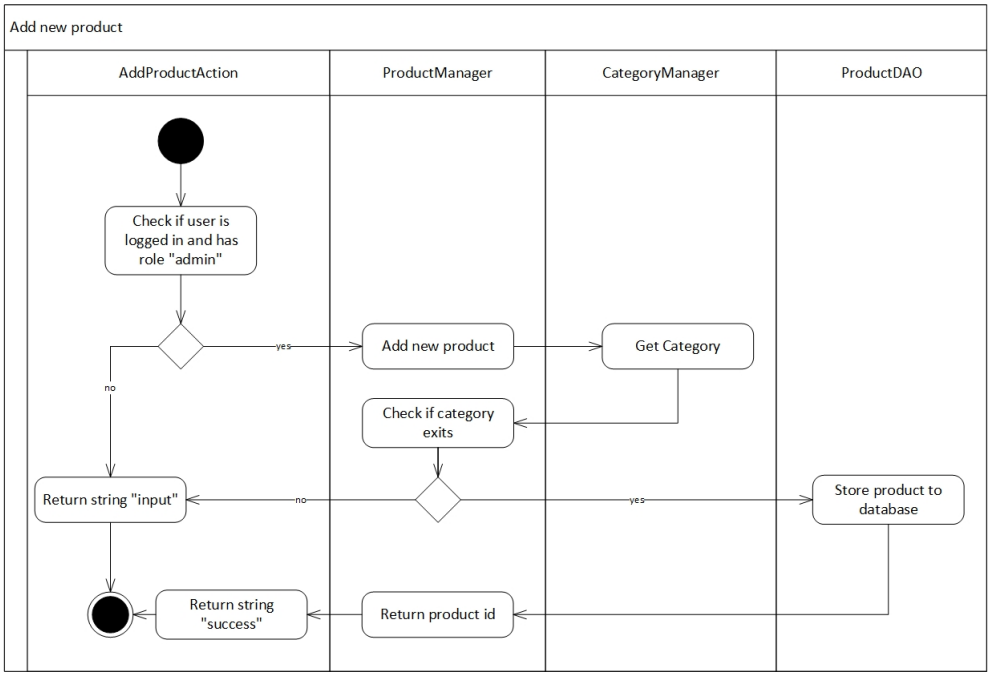

Abbildung 5: Aktivitätsdiagramm für das Hinzufügen neuer Produkte

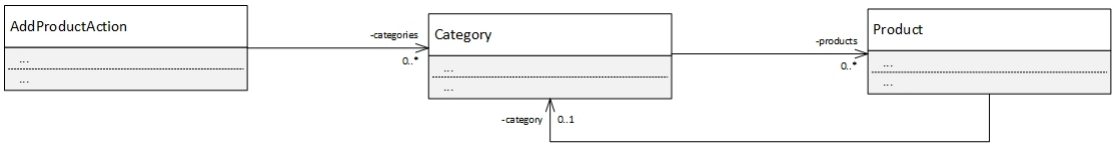

Abbildung 6: Klassendiagramm der *AddProductAction*

3.4 Nach Produkten suchen

TODO ...