JEE SEMESTERARBEIT

Onlineshop mit JSF, Spring Security und AJAX

birgit.sturzenegger@gmx.ch

Abstract

Das vorliegende Dokument beschreibt dieSemesterarbeit zum Fach JavaEnterpriseEdition(JEE)

Inhaltsverzeichnis

[1 Einführung 3](#_Toc500279996)

[2 Analyse 4](#_Toc500279997)

[3 Entwurf 14](#_Toc500279998)

[4 Implementation 21](#_Toc500279999)

[Technische Infrastruktur 21](#_Toc500280000)

[Konfiguration 21](#_Toc500280001)

[5 Test 22](#_Toc500280002)

[6 Schwerpunkt Themen 22](#_Toc500280003)

[7 AJAX Technologie 22](#_Toc500280004)

[6 Evaluierungsbericht 23](#_Toc500280005)

Abbildungsverzeichnis

[Abbildung 1 UseCase-Diagramm 4](#_Toc500280009)

[Abbildung 2 MVC-Muster JSF 15](file:///C:\%23FFHS\7.%20Semester\JEE_Semesterarbeit.docx#_Toc500280010)

[Abbildung 3 Package-Struktur 16](#_Toc500280011)

[Abbildung 4 Klassendiagramm package ffhs.onlineshop (Controllers) 17](#_Toc500280012)

[Abbildung 5 Klassendiagramm package ffhs.onlineshop.model (Models) 18](#_Toc500280013)

[Abbildung 6 Klassendiagramm ffhs.onlineshop.repository (Repositories) 19](#_Toc500280014)

[Abbildung 7 GUI Produktseite 20](#_Toc500280015)

[Abbildung 8 GUI Produktdetail 21](#_Toc500280016)

Tabellenverzeichnis

[Tabelle 1 Registrieren (UseCase 1) 5](#_Toc500274245)

[Tabelle 2 Anmelden (UseCase 2) 5](#_Toc500274246)

[Tabelle 3 Abmelden (UseCase 3) 6](#_Toc500274247)

[Tabelle 4 Produktübersicht anschauen (UseCase 4) 6](#_Toc500274248)

[Tabelle 5 Produkt-Detail anschauen (UseCase 5) 6](#_Toc500274249)

[Tabelle 6 Verkäufer-Infos und Bewertungen anschauen (UseCase 6) 7](#_Toc500274250)

[Tabelle 7 Produkt suchen (UseCase 7) 7](#_Toc500274251)

[Tabelle 8 Produkt kaufen (UseCase 8) 7](#_Toc500274252)

[Tabelle 9 Registrierungsdaten ändern (UseCase 9) 8](#_Toc500274253)

[Tabelle 10 Einkaufsübersicht anschauen (UseCase 10) 8](#_Toc500274254)

[Tabelle 11 Einkauf bewerten (UseCase 11) 8](#_Toc500274255)

[Tabelle 12 Angebotsübersicht anschauen (UseCase 12) 8](#_Toc500274256)

[Tabelle 13 Verkauf bewerten (UseCase 13) 9](#_Toc500274257)

[Tabelle 14 Angebot löschen (UseCase 14) 9](#_Toc500274258)

[Tabelle 15 Angebot löschen (UseCase 14) 10](#_Toc500274259)

[Tabelle 16 Angebot erstellen (UseCase 16) 10](#_Toc500274260)

[Tabelle 17 Kategorieübersicht anschauen (UseCase 17) 10](#_Toc500274261)

[Tabelle 18 Kategorie löschen (UseCase 18) 11](#_Toc500274262)

[Tabelle 19 Kategorie hinzufügen (Use Case 19) 11](#_Toc500274263)

[Tabelle 20 Kategorie ändern (UseCase 20) 11](#_Toc500274264)

[Tabelle 21 Zustandsübersicht anschauen (UseCase 21) 11](#_Toc500274265)

[Tabelle 22 Zustand löschen (UseCase 22) 12](#_Toc500274266)

[Tabelle 23 Zustand hinzufügen (UseCase 23) 12](#_Toc500274267)

[Tabelle 24 Zustand ändern (UseCase 24) 12](#_Toc500274268)

# 1 Einführung

* 1. Fokus

Die Semesterarbeit besteht darin, einen vordefinierten Webshop zu erweitern. Dabei werden Schwerpunkte gesetzt, implementiert und dokumentiert.

Ich habe den Webshop um folgende Funktionen erweitert:

* Implementation des Login/Logout mit Rollen (User, Administrator)
* Das Objekt Benutzer wurde um die Attribute Adresse erweitert (Registrierung)
* Der Benutzer kann seine Registrierungs-Daten ansehen und ändern
* Der Benutzer kann seine Angebote und seine Einkäufe ansehen
* Der Benutzer kann Änderungen an seinen Angeboten vornehmen, solange diese nicht gekauft sind
* Der Benutzer kann seine Einkäufe und Verkäufe bewerten
* Das Item-Objekt (Angebot) wurde durch die Attribute Kategorie und Zustand erweitert

Ein Administrator kann Kategorien und Zustands-Optionen verwalten.

* 1. Schwerpunkte

Der Schwerpunkt der JEE-Applikation liegt in der Umsetzung mit JSF (JavaServerFaces). Zusätzlich wurde das Login/Logout sowie die Zugangsberechtigung mit Spring-MVC-Security umgesetzt. Ein weiteres Thema, mit welchem ich mich befasst habe, ist die AJAX-Technologie.

# 2 Analyse

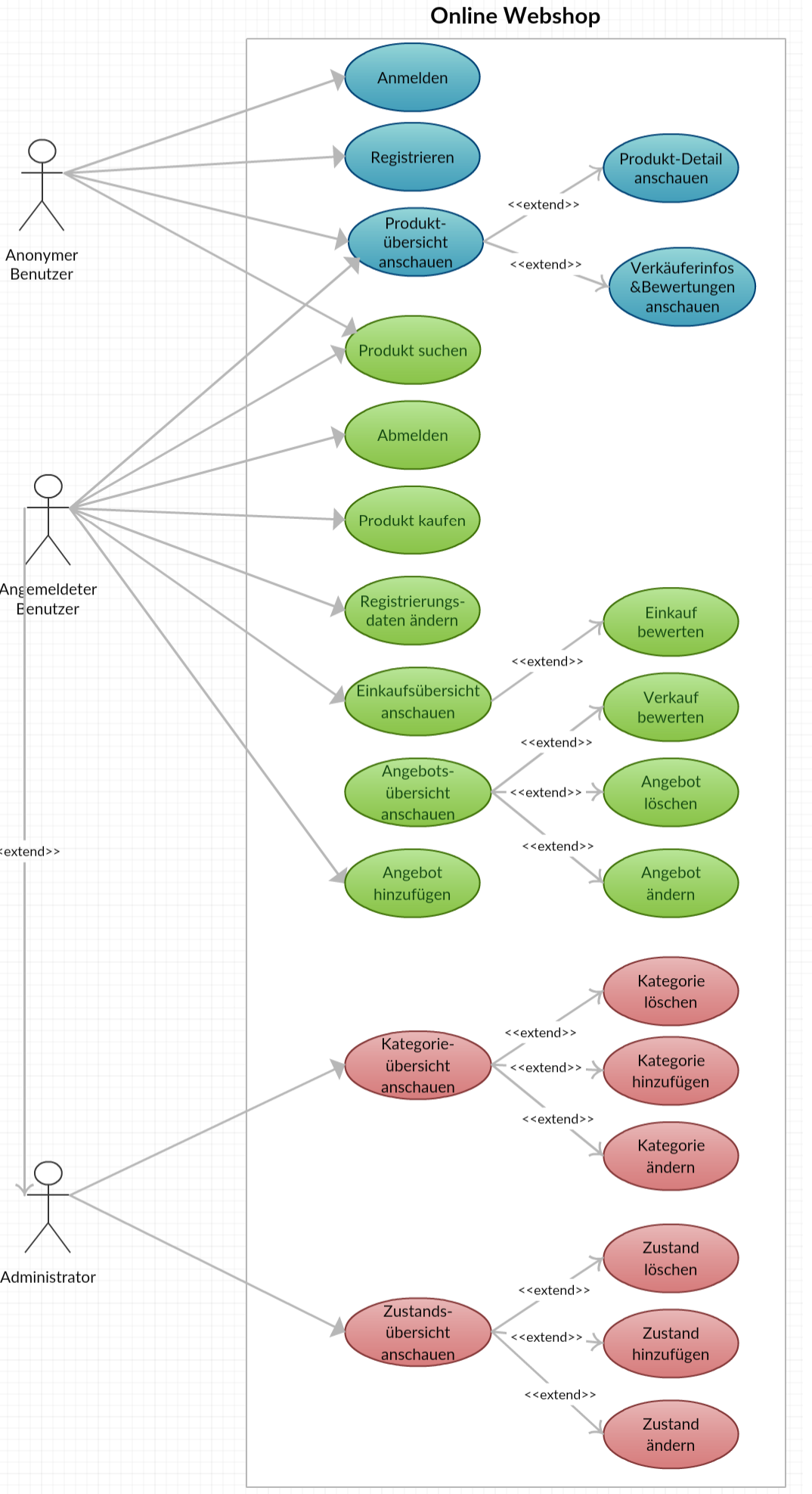


Abbildung 1 UseCase-Diagramm

4.1 Anwendungsfälle

In diesem Kapitel werden die relevanten Anwendungsfälle dokumentiert sowie deren Bedingungen für eine erfolgreiche Durchführung festgehalten.

|  |  |
| --- | --- |
| UseCase 1 | Registrieren |
| Kurzbeschreibung | Ein anonymer Benutzer registriert sich |
| Akteure | Anonymer Benutzer |
| Auslösendes Ereignis | Ein anonymer Benutzer möchte sich registrieren |
| Vorbedingung | Der Benutzer darf nicht angemeldet sein.  Alle Felder des Registrierungsformulars sind obligatorisch.  Das Passwort muss aus mindestens 6 Zeichen bestehen. |
| Eingehende Informationen | Vorname, Nachname, Strasse, PLZ, Ort, Land, eMail-Adresse, Passwort |
| Ablauf | 1. Ein Benutzer füllt das Registrierungsformular aus 2. Der Benutzer drückt Registrieren |
| Nachbedingungen | Die Daten des Benutzers sind gespeichert |
| Kommentar | Die eMail-Adresse und das Passwort wird für das Login benutzt |

Tabelle 1 Registrieren (UseCase 1)

|  |  |
| --- | --- |
| UseCase 2 | Anmelden |
| Kurzbeschreibung | Ein anonymer Benutzer meldet sich an |
| Akteure | Anonymer Benutzer |
| Auslösendes Ereignis | Ein anonymer Benutzer möchte sich anmelden |
| Vorbedingung | Der Benutzer darf nicht angemeldet sein.  Alle Felder des Loginformulars sind obligatorisch |
| Eingehende Informationen | eMail-Adresse, Passwort |
| Ablauf | 1. Ein Benutzer füllt das Loginformular aus 2. Der Benutzer drückt Login 3. Die eingehenden Daten werden überprüft 4. Wenn diese gültig sind, dann wird der Benutzer auf die 5. gewünschte Seite weitergeleitet 6. Wenn diese nicht gültig sind, wird diese dem Bentuzer angezeigt |
| Nachbedingungen | Bei einer gültigen Anmeldung, ist die Login-Funktion nicht mehr ersichtlich, dafür steht jetzt die Logout-Funktion zur Verfügung. |
| Kommentar |  |

Tabelle 2 Anmelden (UseCase 2)

|  |  |
| --- | --- |
| UseCase 3 | Abmelden |
| Kurzbeschreibung | Ein angemeldeter Benutzer meldet sich ab |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein angemeldeter Benutzer möchte sich abmelden |
| Vorbedingung | Der Benutzer muss angemeldet sein. |
| Eingehende Informationen |  |
| Ablauf | 1. Ein angemeldeter Benutzer drückt auf Logout 2. Der angemeldeter Benutzer wird abgemeldet |
| Nachbedingungen | Der Benutzer ist abgemeldet.  Die Logout-Funktion ist nicht mehr ersichtlich, dafür steht jetzt die Login- sowie Registrierungs-Funktion zur Verfügung |
| Kommentar |  |

Tabelle 3 Abmelden (UseCase 3)

|  |  |
| --- | --- |
| UseCase 4 | Produktübersicht anschauen |
| Kurzbeschreibung | Ein Benutzer schaut sich die Produkte an |
| Akteure | Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte sich die Produkte anschauen |
| Vorbedingung | Produkte müssen vorhanden sein |
| Eingehende Informationen | Keine |
| Ablauf | 1. Ein Benutzer schaut sich die Produkte an |
| Nachbedingungen | Alle Produkte werden angezeigt.  Wenn der Benutzer angemeldet ist, steht die Kaufen-Funktion zur Verfügung. |
| Kommentar |  |

Tabelle 4 Produktübersicht anschauen (UseCase 4)

|  |  |
| --- | --- |
| UseCase 5 | Produkt-Detail anschauen |
| Kurzbeschreibung | Ein Benutzer schaut sich die Details zu einem Produkt an |
| Akteure | Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte sich die Details eines Produktes anschauen |
| Vorbedingung | Der Benutzer muss ein Produkt auswählen |
| Eingehende Informationen | Ausgewähltes Produkt |
| Ablauf | 1. Ein Benutzer wählt ein Produkt aus 2. Der Benutzer drückt auf den Titel des Produktes 3. Die Details zum ausgewählten Produkt werden angezeigt |
| Nachbedingungen | Die Produkt-Details werden angezeigt.  Wenn der Benutzer angemeldet ist, steht die Kaufen-Funktion zur Verfügung. |
| Kommentar |  |

Tabelle 5 Produkt-Detail anschauen (UseCase 5)

|  |  |
| --- | --- |
| UseCase 6 | Verkäufer-Infos und Bewertungen anschauen |
| Kurzbeschreibung | Ein Benutzer schaut sich die Infos zu einem Verkäufer an |
| Akteure | Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte Informationen über den Verkäufer anschauen |
| Vorbedingung | Der Benutzer muss ein Produkt auswählen |
| Eingehende Informationen | Ausgewähltes Produkt |
| Ablauf | 1. Ein Benutzer wählt ein Produkt aus 2. Der Benutzer drückt auf den Verkäufers des Produktes 3. Die Informationen zum Verkäufer sowie dessen Bewertungen werden angezeigt |
| Nachbedingungen | Die Infos und Bewertungen des Verkäufers werden angezeigt |
| Kommentar |  |

Tabelle 6 Verkäufer-Infos und Bewertungen anschauen (UseCase 6)

|  |  |
| --- | --- |
| UseCase 7 | Produkt suchen |
| Kurzbeschreibung | Ein Benutzer sucht nach einem Produkt |
| Akteure | Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte nach einem Produkt suchen |
| Vorbedingung | Der Benutzer muss etwas ins Suchfeld eingeben |
| Eingehende Informationen | Suchbegriff |
| Ablauf | 1. Ein Benutzer gibt einen Suchbegriff an 2. Der Benutzer löst die Suche aus 3. Es werden alle Produkte mit dem Suchbegriff im Titel angezeigt |
| Nachbedingungen |  |
| Kommentar |  |

Tabelle 7 Produkt suchen (UseCase 7)

|  |  |
| --- | --- |
| UseCase 8 | Produkt kaufen |
| Kurzbeschreibung | Ein angemeldeter Benutzer kauft ein Produkt |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte ein Produkt kaufen |
| Vorbedingung | Der Benutzer muss angemeldet sein |
| Eingehende Informationen | Produkt ID |
| Ablauf | 1. Ein Benutzer wählt ein Produkt aus 2. Der Benutzer kauft das Produkt 3. Das Produkt wird als gekauft markiert |
| Nachbedingungen | Das Produkt ist als Verkauft markiert und kann nicht mehr gekauft werden. |
| Kommentar |  |

Tabelle 8 Produkt kaufen (UseCase 8)

|  |  |
| --- | --- |
| UseCase 9 | Registrierungsdaten ändern |
| Kurzbeschreibung | Ein angemeldeter Benutzer ändert seine Registrierungsdaten |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte seine Registrierungsdaten ändern |
| Vorbedingung | Der Benutzer muss angemeldet sein |
| Eingehende Informationen | Vorname, Nachname, Strasse, PLZ, Ort, Land, eMail-Adresse, Passwort |
| Ablauf | 1. Ein Benutzer ändert seine Registrierungsdaten 2. Der Benutzer speichert seine geänderten Daten |
| Nachbedingungen | Die Daten des Benutzers sind geändert |
| Kommentar |  |

Tabelle 9 Registrierungsdaten ändern (UseCase 9)

|  |  |
| --- | --- |
| UseCase 10 | Einkaufsübersicht anschauen |
| Kurzbeschreibung | Ein angemeldeter Benutzer schaut sich seine Einkäufe an |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte seine Einkäufe ansehen |
| Vorbedingung | Der Benutzer muss angemeldet sein |
| Eingehende Informationen |  |
| Ablauf | 1. Ein Benutzer schaut seine Einkäufe an |
| Nachbedingungen | Die Daten des Benutzers sind geändert |
| Kommentar |  |

Tabelle 10 Einkaufsübersicht anschauen (UseCase 10)

|  |  |
| --- | --- |
| UseCase 11 | Einkauf bewerten |
| Kurzbeschreibung | Ein angemeldeter Benutzer bewertet einen Einkauf |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte einen Einkäufe bewerten |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Kauf muss von diesem Benutzer getätig worden sein |
| Eingehende Informationen | ID des Kaufs (Item), Bewertung (ganze Zahl zwischen 1 und 5), Kommentar |
| Ablauf | 1. Ein Benutzer geht auf seine Einkaufsübersicht 2. Der Benutzer wählt einen Kauf aus 3. Der Benutzer bewertetet den Kauf mit einem Sterne-Rating (1 bis 5) und einem Kommentar |
| Nachbedingungen | Der Einkauf ist bewertet |
| Kommentar | Die Bewertung kann nach dem Speichern nicht mehr geändert werden. |

Tabelle 11 Einkauf bewerten (UseCase 11)

|  |  |
| --- | --- |
| UseCase 12 | Angebotsübersicht anschauen |
| Kurzbeschreibung | Ein angemeldeter Benutzer schaut sich seine Angebote an |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte seine Angebote ansehen |
| Vorbedingung | Der Benutzer muss angemeldet sein |
| Eingehende Informationen |  |
| Ablauf | 1. Ein Benutzer schaut seine Angebote an |
| Nachbedingungen |  |
| Kommentar |  |

Tabelle 12 Angebotsübersicht anschauen (UseCase 12)

|  |  |
| --- | --- |
| UseCase 13 | Verkauf bewerten |
| Kurzbeschreibung | Ein angemeldeter Benutzer bewertet einen Verkauf |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte einen Verkauf bewerten |
| Vorbedingung | Der Benutzer muss angemeldet sein  Das Angebot muss von diesem Benutzer sein  Das Produt muss gekauft worden sein |
| Eingehende Informationen | ID des Verkaufs (Item), Bewertung (ganze Zahl zwischen 1 und 5), Kommentar |
| Ablauf | 1. Ein Benutzer geht auf seine Angebotsübersicht 2. Der Benutzer wählt einen Verkauf aus 3. Der Benutzer bewertetet den Verkauf mit einem Sterne-Rating (1 bis 5) und einem Kommentar |
| Nachbedingungen | Der Kauf ist bewertet |
| Kommentar | Die Bewertung kann nach dem Speichern nicht mehr geändert werden. |

Tabelle 13 Verkauf bewerten (UseCase 13)

|  |  |
| --- | --- |
| UseCase 14 | Angebot löschen |
| Kurzbeschreibung | Ein angemeldeter Benutzer löscht ein Angebot |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte einen Verkauf bewerten |
| Vorbedingung | Der Benutzer muss angemeldet sein  Das Angebot muss von diesem Benutzer sein  Das Produkt darf noch nicht verkauft sein |
| Eingehende Informationen | ID des Angebots (Item) |
| Ablauf | 1. Ein Benutzer geht auf seine Angebotsübersicht 2. Der Benutzer wählt ein Angebot aus 3. Der Benutzer löscht das Angebot |
| Nachbedingungen | Das Angebot ist gelöscht |
| Kommentar |  |

Tabelle 14 Angebot löschen (UseCase 14)

|  |  |
| --- | --- |
| UseCase 15 | Angebot ändern |
| Kurzbeschreibung | Ein angemeldeter Benutzer löscht ein Angebot |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte einen Verkauf bewerten |
| Vorbedingung | Der Benutzer muss angemeldet sein  Das Angebot muss von diesem Benutzer sein  Das Produkt darf noch nicht verkauft sein |
| Eingehende Informationen |  |
| Ablauf | 1. Ein Benutzer geht auf seine Angebotsübersicht 2. Der Benutzer wählt ein Angebot aus 3. Der Benutzer löscht das Angebot |
| Nachbedingungen | Das Angebot ist gelöscht |
| Kommentar |  |

Tabelle 15 Angebot löschen (UseCase 14)

|  |  |
| --- | --- |
| UseCase 16 | Angebot erstellen |
| Kurzbeschreibung | Ein angemeldeter Benutzer erstellt ein Angebot |
| Akteure | Angemeldeter Benutzer |
| Auslösendes Ereignis | Ein Benutzer möchte einen neues Angebot erstellen |
| Vorbedingung | Der Benutzer muss angemeldet sein |
| Eingehende Informationen |  |
| Ablauf | 1. Ein Benutzer erstellt ein neues Angebot 2. Der Benutzer speichert das neue Angebot |
| Nachbedingungen | Das Angebot ist gespeichert und steht zum Kauf bereit |
| Kommentar |  |

Tabelle 16 Angebot erstellen (UseCase 16)

|  |  |
| --- | --- |
| UseCase 17 | Kategorieübersicht anschauen |
| Kurzbeschreibung | Ein angemeldeter Benutzer schaut die Kategorieübersicht an |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte die Kategorieübersicht anschauen |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Benutzer muss die Rolle Administrator haben |
| Eingehende Informationen |  |
| Ablauf | 1. Ein Benutzer schaut alle vorhanden Kategorien an |
| Nachbedingungen |  |
| Kommentar |  |

Tabelle 17 Kategorieübersicht anschauen (UseCase 17)

|  |  |
| --- | --- |
| UseCase 18 | Kategorie löschen |
| Kurzbeschreibung | Ein angemeldeter Benutzer löscht eine Kategorie |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte eine Kategorie löschen |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Benutzer muss die Rolle Administrator haben  Die Kategorie darf nicht bereits von einem Angebot verwendet worden sein |
| Eingehende Informationen | ID der Kategorie |
| Ablauf | 1. Ein Benutzer geht auf die Kategorieübersicht 2. Der Benutzer wählt eine Kategorie aus 3. Der Benutzer löscht die Kategorie |
| Nachbedingungen | Die Kategorie ist gelöscht |
| Kommentar |  |

Tabelle 18 Kategorie löschen (UseCase 18)

|  |  |
| --- | --- |
| UseCase 19 | Kategorie hinzufügen |
| Kurzbeschreibung | Ein angemeldeter Benutzer erstellt eine neue Kategorie |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte eine neue Kategorie erstellen |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Benutzer muss die Rolle Administrator haben |
| Eingehende Informationen | Bezeichung der Kategorie |
| Ablauf | 1. Ein Benutzer geht auf die Kategorieübersicht 2. Der Benutzer gibt eine Bezeichnung ein 3. Der Benutzer speichert die neue Kategorie |
| Nachbedingungen | Die neue Kategorie ist gespeichert |
| Kommentar |  |

Tabelle 19 Kategorie hinzufügen (Use Case 19)

|  |  |
| --- | --- |
| UseCase 20 | Kategorie ändern |
| Kurzbeschreibung | Ein angemeldeter Benutzer ändert eine bestehende Kategorie |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte eine bestehende Kategorie ändern |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Benutzer muss die Rolle Administrator haben |
| Eingehende Informationen | ID des Kategorie, Bezeichnung der Kategorie |
| Ablauf | 1. Ein Benutzer geht auf die Kategorieübersicht 2. Der Benutzer wählt eine Kategorie aus 3. Der Benutzer ändert die Bezeichnung der Kategorie |
| Nachbedingungen | Das Kategorie ist geändert |
| Kommentar |  |

Tabelle 20 Kategorie ändern (UseCase 20)

|  |  |
| --- | --- |
| UseCase 21 | Zustandsübersicht anschauen |
| Kurzbeschreibung | Ein angemeldeter Benutzer schaut die Zustandsübersicht an |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte die Zustandsübersicht anschauen |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Benutzer muss die Rolle Administrator haben |
| Eingehende Informationen |  |
| Ablauf | 1. Ein Benutzer schaut alle vorhanden Zustände an |
| Nachbedingungen |  |
| Kommentar |  |

Tabelle 21 Zustandsübersicht anschauen (UseCase 21)

|  |  |
| --- | --- |
| UseCase 22 | Zustand löschen |
| Kurzbeschreibung | Ein angemeldeter Benutzer löscht einen Zustand |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte einen Zustand löschen |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Benutzer muss die Rolle Administrator haben  Der Zustand darf nicht bereits von einem Angebot verwendet worden sein |
| Eingehende Informationen | ID des Zustands |
| Ablauf | 1. Ein Benutzer geht auf die Zustandsübersicht 2. Der Benutzer wählt einen Zustand aus 3. Der Benutzer löscht den Zustand |
| Nachbedingungen | Der Zustand ist gelöscht |
| Kommentar |  |

Tabelle 22 Zustand löschen (UseCase 22)

|  |  |
| --- | --- |
| UseCase 23 | Zustand hinzufügen |
| Kurzbeschreibung | Ein angemeldeter Benutzer erstellt einen neuen Zustand |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte einen neuen Zustand erstellen |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Benutzer muss die Rolle Administrator haben |
| Eingehende Informationen | Bezeichung des Zustands |
| Ablauf | 1. Ein Benutzer geht auf die Zustandsübersich 2. Der Benutzer gibt eine Bezeichnung ein 3. Der Benutzer speichert den neuen Zustand |
| Nachbedingungen | Der neue Zustand ist gespeichert |
| Kommentar |  |

Tabelle 23 Zustand hinzufügen (UseCase 23)

|  |  |
| --- | --- |
| UseCase 24 | Zustand ändern |
| Kurzbeschreibung | Ein angemeldeter Benutzer ändert einen bestehenden Zustand |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte einen bestehenden Zustand ändern |
| Vorbedingung | Der Benutzer muss angemeldet sein  Der Benutzer muss die Rolle Administrator haben |
| Eingehende Informationen | ID des Zustands, Bezeichnung des Zustands |
| Ablauf | 1. Ein Benutzer geht auf die Zustandsübersicht 2. Der Benutzer wählt einen Zustand aus 3. Der Benutzer ändert die Bezeichnung des Zustands |
| Nachbedingungen | Der Zustand ist geändert |
| Kommentar |  |

Tabelle 24 Zustand ändern (UseCase 24)

4.2 Fachklassenmodell

4.3 Anwendungsfallabläufe

# 3 Entwurf

3.1 Softwarearchitektur

Der Webshop ist eine Webapplikation und besteht aus folgenden zwei logischen Schichten:

* Presentation layer
* Data access layer

Um die verschiedenen Aspekte der Anwendung zu trennen wurde das MVC-Pattern eingesetzt. Dieses Pattern eignet sich gut, um eine flexible und lose gekoppelte Web-Anwendung zu entwickeln, da mit MVC eine strikte Trennung der Geschäftslogik von der Darstellung erreicht werden kann.

Des Weiteren wurde Dependency Injection (DI) eingesetzt, um eine enge Koppelung der Systemkomponenten zu verhindern. Mit Dependency Injection werden die Klassen entkoppelt und lassen sich leichter wiederverwenden und testen.

In meinem JSF-Projekt ist das MVC-Mapping wie folgt:

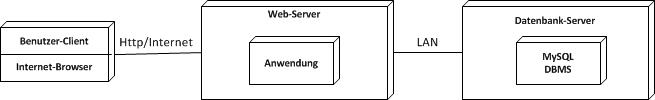
* **Modell**: Komponente zum Halten des Datenmodells. In diesem Projekt sind das die DAO-Klassen und die Entitäten (Models). Der Einstiegspunkt ist die verwaltete Bean, aber in Java EE (zu dem JSF gehört) werden diese Artefakte normalerweise von EJB bzw. JPA implementiert.
* **Ansicht**: Die UI-Komponenten und ihre Zusammensetzung zu einer ganzen Seite. Diese sind vollständig in der Domäne von JSF und wird von JSF UIComponents bzw. Facelets implementiert (Facelets-XHTML-Dokumente).
* **Controller**: Komponente für die Programmsteuerung, welche Befehle und eingehende Daten vom Benutzer handhabt, diese zu den richtigen Teilen leitet und eine Ansicht zur Anzeige auswählt. In JSF schreibt man diesen Controller nicht, denn als Controller fungiert das *FacesServlet* (wird bereits vom Framework bereitgestellt, Front-Controller-Pattern).

Abbildung 2 MVC-Muster JSF



Die Software läuft auf einem Webserver (Glassfish) und die Datenbank auf einem Datenbank-Server.

Für dieses Projekt wurde die Software mit der Datenbank zusammen nur lokal getestet (localhost).



5.1 Klassendiagramme

Das Klassenmodell beschreibt die statische Struktur der Klassen in einem System sowie die Beziehungen untereinander. Die Eigenschaften und das Verhalten der Klassen werden durch ihre Attribute und Methoden abgebildet.

Die Software besteht aus folgenden drei Paketen:

* ffhs.onlineshop
* ffhs.onlineshop.model
* ffhs.onlineshop.repository

Um einen guten Überblick zu erhalten, werden zuerst die Packages mit ihren wichtigsten Klassen ohne Attribute und Methoden abgebildet.

Das Klassendiagramm mit sämtlichen Attributen und Operationen wird nachfolgend abgebildet. Bei den Klassendiagrammen wird bei Klassen, welche ein Interface implementieren die Methoden nur im Interface abgebildet und nicht nochmals in der Implementation des Interfaces. Private Methoden werden nicht abgebildet. Private Attribute werden abgebildet, insofern sie eine Assoziation ausdrücken.

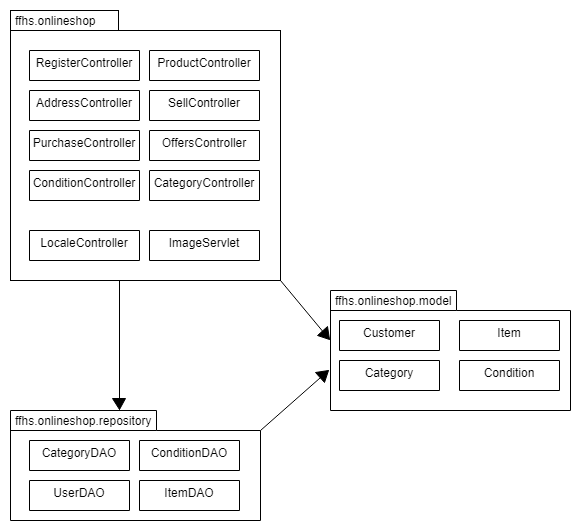


Abbildung 3 Package-Struktur

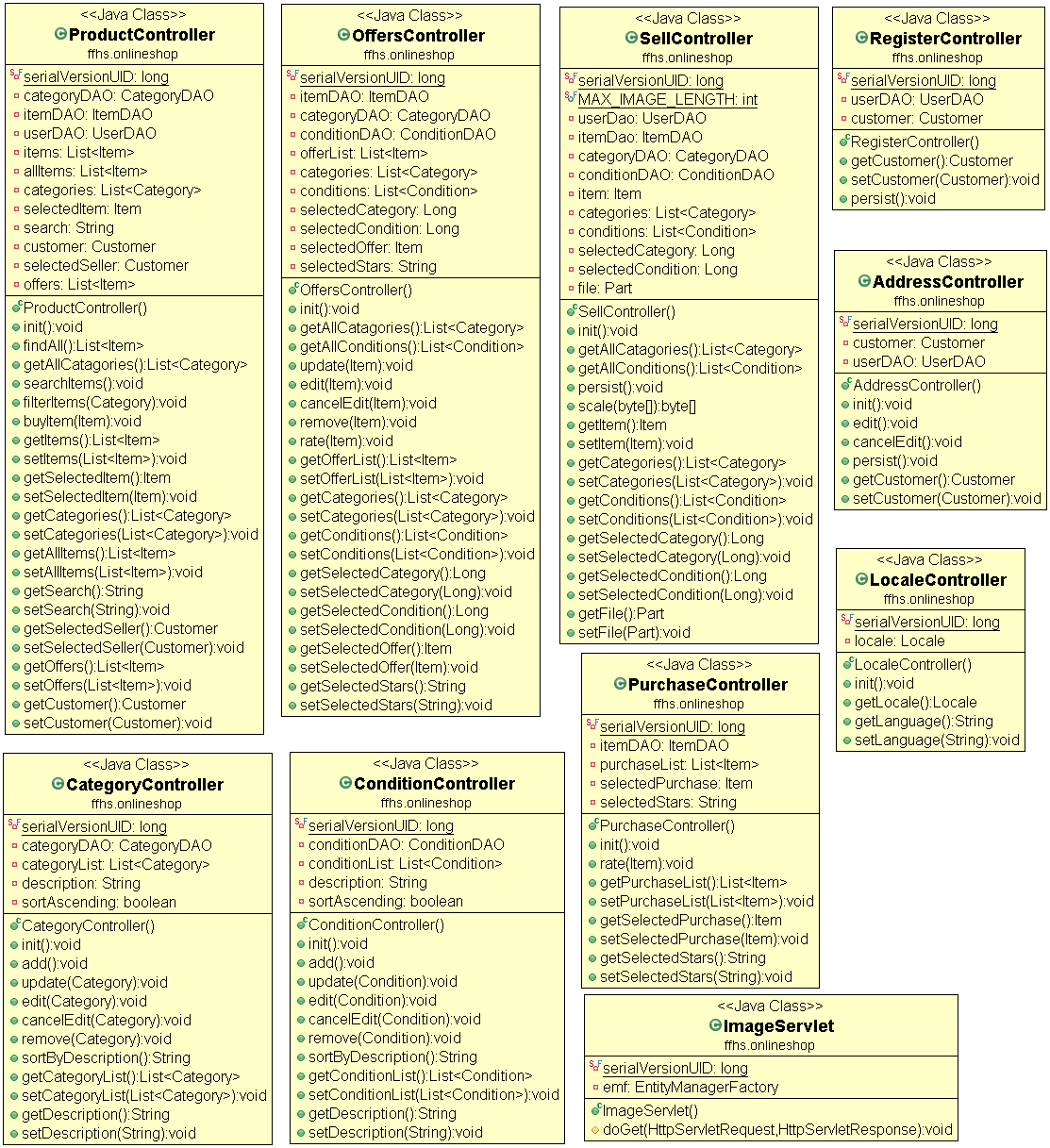


Abbildung 4 Klassendiagramm package ffhs.onlineshop (Controllers)

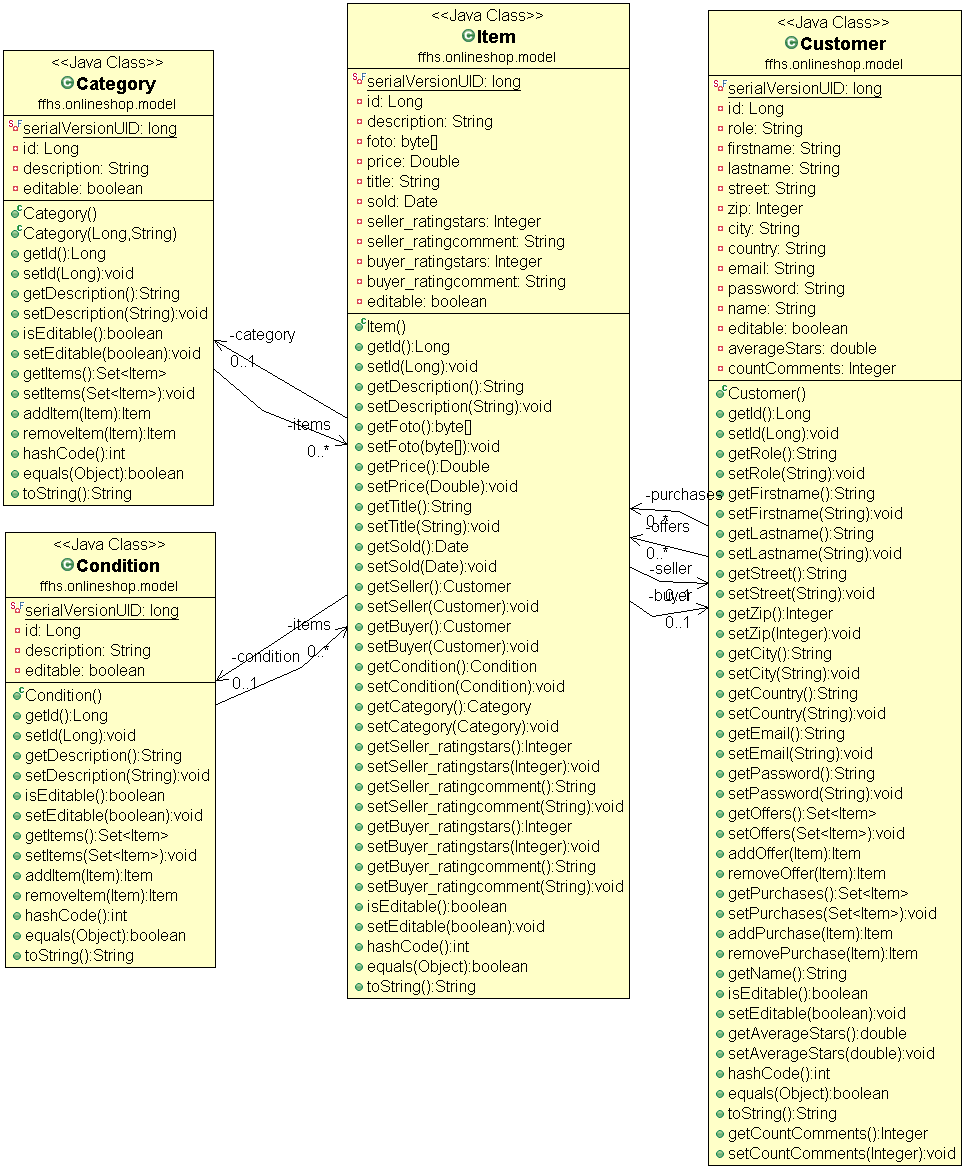


Abbildung 5 Klassendiagramm package ffhs.onlineshop.model (Models)

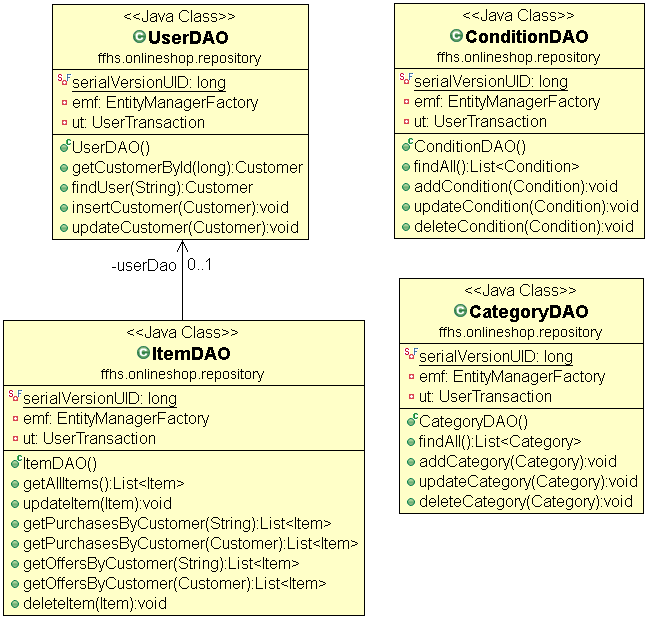


Abbildung 6 Klassendiagramm ffhs.onlineshop.repository (Repositories)

5.2 Datenmodell

Für das Entity-Relationship-Diagramm gibt es zwei Modelle. Das erste Modell zeigt die Entitäten und ihre Beziehungen, wie sie in der Software verwendet werden. Das zweite Model zeigt das Datenbankschema mit ihren Tabellen und Beziehungen.

5.3 Sequenzmodell

...

5.5 Speziﬁkation der Bedienoberﬂächen (GUI-Skizzen)

Für die Bedienoberflächen wurden Mockups mit dem Werkzeug Balsamiqu erstellt. Diese Mockups dienten als Vorlage für die Implementation.

Produktseite

Die Produktseite ist auch zugleich die Welcome-Seite (die Einstiegsseite).

Auf der Produktseite sind alle Produkte aufgelistet. Es kann nach Produkten gesucht werden, indem man einen Suchbegriff in das entsprechende Feld einträgt und den Suchen-Button betätigt. Es wird nur nach dem Titel des Produktes gesucht. Die Produkte können nach Kategorien gefiltert werden.

Pro Produkt sind ein Foto, der Titel, ein Teil der Beschreibung, der Verkäufer, die Sterne-Bewertung des Verkäufers (Kauf-Bewertung), die Anzahl an Bewertungen, der Preis sowie der Kaufen-Button ersichtlich.

Ein Benutzer muss nicht angemeldet sein, um die Produkte zu sehen. Ein anonymer Benutzer sieht jedoch den Kaufen-Button nicht (so wie hier abgebildet).

Wenn ein Produkt verkauft ist, wird die obere rechte Ecke rot markiert und mit einem Label Sold vermerkt, ansonsten ist die obere rechte Ecke vom Produkt grün.

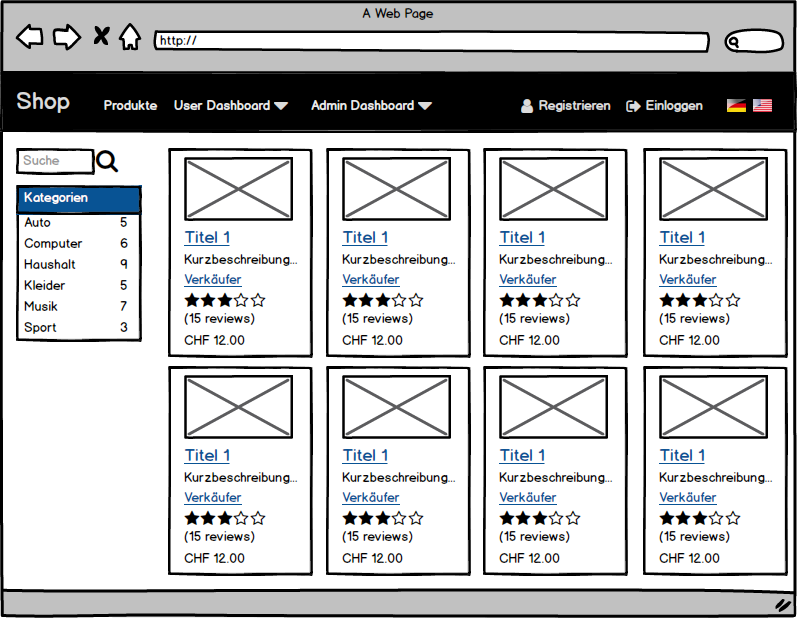


Abbildung 7 GUI Produktseite

Produkt-Detail

Wenn der Benutzer auf den Titel eines Produktes klickt, erscheint eine detallierte Beschreibung des Produkts als Popup-Fenster. Es ist das Foto (vergrössert), der Titel, die gesamte Beschreibung, der Peis sowie der Kaufen-Button ersichtlich.

Wenn der Benutzer nicht angemeldet ist, ist der Kaufen-Button nicht ersichtlich (wie hier abgebildet).

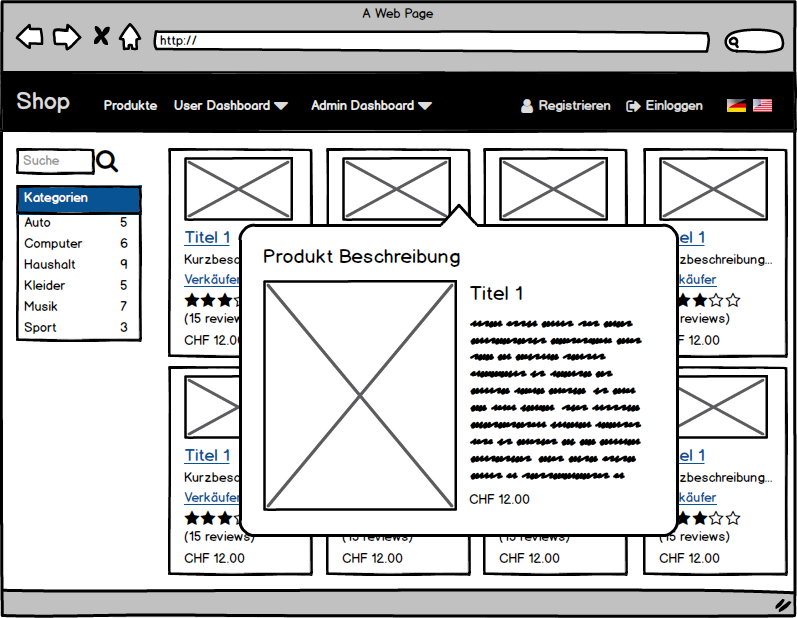


Abbildung 8 GUI Produktdetail

Registrieren

Ein Benutzer kann sich registerien, soweit er nicht angemeldet ist. Sobals ein Benutzer sich angemeldet hat (Einloggen) ist das Registrieungsmenü nicht mehr ersichtlich.

Login

User Dashbord – Meine Adresse

User Dashbord – Meine Einkäufe

User Dashbord – Meine Angebote

User Dashbord – Neues Angebot

Admin Dashboard – Verwaltung Kategorien

Admin Dashboard – Verwaltung Zustand

5.4 Testkonzept

Es wurde ein Testkonzept erarbeitet um die umgesetzten Anwendungsfälle auf der Gui zu testen.

# 4 Implementation

# Technische Infrastruktur

Für die Semesterarbeit wurden folgende Komponenten genutzt:

* Glassﬁsh Open Source Edition v.4.1.1 (Web App Container)
* Oracle Express Edition 11g (DBMS)
* Java Enterprise Edition 1.8 (JDK)

# Konfiguration

* Glassfish wie im Buch
* Oracle wie im Buch
* Maven Projekt (POM)

Zusätzliche Frameworks

* Spring Security
* Bootstrap
* ...

# 5 Test

5.1 Testprotokoll

# 6 Schwerpunkt Themen

6.1 JSF

JavaServer Faces (JSF) ist ein komponentenbasiertes Web Application Framework und stellt in Java EE den Standard dar, um grafische Oberflächen über einen Java-EE-Webserver anzubieten, die als interaktive HTML-Webseiten im Webbrowser angezeigt werden.

In meinem Projekt habe ich JavaServer Faces Version 2.2 verwendet.

6.2 Spring Security Framework

Mit dem Open-Source-basierten Security Framework existiert eine interessante Alternative zur Container-Managed-Security von Java EE.

Die Zugriffsberechtigung des Onlineshop wurde mittels Spring Security umgesetzt.

Libraries

Was macht diese Konfiguration?

Spring Security stellt Funktionen zur Autorisierung aller Systemschichten (Tiers) und zur Integration einer Authentifizierung mit dutzenden Anbietern bereit. Spring Security lässt sich gut in Kombination mit anderen Spring Technologien wie Spring MVC, Spring Integration und Spring Web Services nutzen. Anwendungsfälle wie ein Audit von internen Systemnachrichten, das Aussperren von unerwünschten Zugriffen auf Webanwendungen oder das Verhindern von unerlaubten Aufrufen von Service-Methode lassen sich mit Spring Security realisieren.

**Spring Security**

# 7 AJAX Technologie

# 6 Evaluierungsbericht

8.1 Zusammenfassung

8.2 Schwierigkeiten

8.3 Fazit