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 Ziel 5](#_Toc500629378)

[2 Analyse 6](#_Toc500629379)

[3 16](#_Toc500629380)

[Softwareentwurf 17](#_Toc500629381)

[1.1.1 Manuelle Tests 39](#_Toc500629382)

[4 Implementation 47](#_Toc500629383)

[Technische Infrastruktur 47](#_Toc500629384)

[Konfiguration 47](#_Toc500629385)

[6 Schwerpunkt Themen 48](#_Toc500629386)

[Test 59](#_Toc500629387)

[6 Evaluierungsbericht 59](#_Toc500629388)

[Literaturverzeichnis 60](#_Toc500629389)

Abbildungsverzeichnis

[Abbildung 1 UseCase-Diagramm 6](#_Toc500589088)

[Abbildung 2 Fachklassenmodell 16](#_Toc500589089)

[Abbildung 3 MVC-Muster JSF 17](file:///C:\%23FFHS\7.%20Semester\JEE_Semesterarbeit.docx#_Toc500589090)

[Abbildung 4 Package-Struktur 19](#_Toc500589091)

[Abbildung 5 Klassendiagramm package ffhs.onlineshop (Controllers) 20](#_Toc500589092)

[Abbildung 6 Klassendiagramm package ffhs.onlineshop.model (Models) 21](#_Toc500589093)

[Abbildung 7 Klassendiagramm ffhs.onlineshop.repository (Repositories) 22](#_Toc500589094)

[Abbildung 8 Diagramm Datenbank-Schema 22](#_Toc500589095)

[Abbildung 9 GUI Produktseite 24](#_Toc500589096)

[Abbildung 10 GUI Produktdetail 25](#_Toc500589097)

[Abbildung 11 GUI Registrieren 26](#_Toc500589098)

[Abbildung 12 GUI Einloggen 27](#_Toc500589099)

[Abbildung 13 GUI Meine Adresse 28](#_Toc500589100)

[Abbildung 14 GUI Meine Adresse bearbeiten 29](#_Toc500589101)

[Abbildung 15 GUI Meine Einkäufe 30](#_Toc500589102)

[Abbildung 16 GUI Bewertung abgeben 31](#_Toc500589103)

[Abbildung 17 GUI Meine Angebote 32](#_Toc500589104)

[Abbildung 18 GUI Verkaufen 33](#_Toc500589105)

[Abbildung 19 GUI Kein Zugriff 34](#_Toc500589106)

[Abbildung 20 GUI Verwaltung Kategorien 35](#_Toc500589107)

[Abbildung 21 GUI Verwaltung Kategorien - Validerungs-Besipiel 36](#_Toc500589108)

[Abbildung 22 GUI Verwaltung Zustand 37](#_Toc500589109)

[Abbildung 23 GUI Abgemeldet 38](#_Toc500589110)

[Abbildung 24 Spring Security Abhängigkeiten in pom.xml 49](#_Toc500589111)

Tabellenverzeichnis

[Tabelle 1 Registrieren (UseCase 1) 7](#_Toc500629390)

[Tabelle 2 Anmelden (UseCase 2) 7](#_Toc500629391)

[Tabelle 3 Abmelden (UseCase 3) 8](#_Toc500629392)

[Tabelle 4 Produktübersicht anschauen (UseCase 4) 8](#_Toc500629393)

[Tabelle 5 Produkt-Detail anschauen (UseCase 5) 8](#_Toc500629394)

[Tabelle 6 Verkäufer-Infos und Bewertungen anschauen (UseCase 6) 9](#_Toc500629395)

[Tabelle 7 Produkt suchen (UseCase 7) 9](#_Toc500629396)

[Tabelle 8 Produkt kaufen (UseCase 8) 10](#_Toc500629397)

[Tabelle 9 Registrierungsdaten ändern (UseCase 9) 10](#_Toc500629398)

[Tabelle 10 Einkaufsübersicht anschauen (UseCase 10) 10](#_Toc500629399)

[Tabelle 11 Einkauf bewerten (UseCase 11) 11](#_Toc500629400)

[Tabelle 12 Angebotsübersicht anschauen (UseCase 12) 11](#_Toc500629401)

[Tabelle 13 Verkauf bewerten (UseCase 13) 11](#_Toc500629402)

[Tabelle 14 Angebot löschen (UseCase 14) 12](#_Toc500629403)

[Tabelle 15 Angebot ändern (UseCase 15) 12](#_Toc500629404)

[Tabelle 16 Angebot erstellen (UseCase 16) 12](#_Toc500629405)

[Tabelle 17 Kategorieübersicht anschauen (UseCase 17) 13](#_Toc500629406)

[Tabelle 18 Kategorie löschen (UseCase 18) 13](#_Toc500629407)

[Tabelle 19 Kategorie hinzufügen (Use Case 19) 14](#_Toc500629408)

[Tabelle 20 Kategorie ändern (UseCase 20) 14](#_Toc500629409)

[Tabelle 21 Zustandsübersicht anschauen (UseCase 21) 14](#_Toc500629410)

[Tabelle 22 Zustand löschen (UseCase 22) 15](#_Toc500629411)

[Tabelle 23 Zustand hinzufügen (UseCase 23) 15](#_Toc500629412)

[Tabelle 24 Zustand ändern (UseCase 24) 15](#_Toc500629413)

[Tabelle 25 Testkonzept – Teststufen 38](#_Toc500629414)

[Tabelle 26 Testkonzept – Test-Arten 39](#_Toc500629415)

[Tabelle 27 Testkonzept – manuelle Tests 40](#_Toc500629416)

[Tabelle 28 Testfall - Produkt suchen (anonymer Benutzer) 40](#_Toc500629417)

[Tabelle 29 Testfall - Produkt nach Kategorie filtern (anonymer Benutzer) 40](#_Toc500629418)

[Tabelle 30 Testfall - Produktdetail anschauen (anonymer Benutzer) 41](#_Toc500629419)

[Tabelle 31 Testfall - Verkäufer-Infos und Bewertung anschauen(anonymer Benutzer) 41](#_Toc500629420)

[Tabelle 32 Testfall - Eine geschützte Seite wählen (anonymer Benutzer) 41](#_Toc500629421)

[Tabelle 33 Testfall - Registirieren 41](#_Toc500629422)

[Tabelle 34 Testfall - Anmelden 42](#_Toc500629423)

[Tabelle 35 Testfall - Produkt suchen (angemeldeter Benutzer) 42](#_Toc500629424)

[Tabelle 36 Testfall - Produkt nach Kategorie filtern (angemeldeter Benutzer) 42](#_Toc500629425)

[Tabelle 37 Testfall - Produktdetail anschauen (angemeldeter Benutzer) 43](#_Toc500629426)

[Tabelle 38 Testfall - Registrierungsdaten ändern 43](#_Toc500629427)

[Tabelle 39 Testfall - Angebot erstellen 43](#_Toc500629428)

[Tabelle 40 Testfall - Produkt kaufen 43](#_Toc500629429)

[Tabelle 41 Testfall - Einkaufsübersicht anschauen 44](#_Toc500629430)

[Tabelle 42 Testfall - Einkauf bewerten 44](#_Toc500629431)

[Tabelle 43 Testfall - Angebotsübersicht anschauen 44](#_Toc500629432)

[Tabelle 44 Testfall - Angebot ändern 45](#_Toc500629433)

[Tabelle 45 Testfall - Verkauf bewerten 45](#_Toc500629434)

[Tabelle 46 Testfall - Angebot löschen 45](#_Toc500629435)

[Tabelle 47 Testfall - Kategorie hinzufügen 45](#_Toc500629436)

[Tabelle 48 Testfall - Kategorie ändern 46](#_Toc500629437)

[Tabelle 49 Testfall - Kategorie löschen 46](#_Toc500629438)

[Tabelle 50 Testfall - Zustand hinzufügen 46](#_Toc500629439)

[Tabelle 51 Testfall - Zustand ändern 47](#_Toc500629440)

[Tabelle 52 Testfall - Zustand löschen 47](#_Toc500629441)

[Tabelle 53 Testfall - Abmelden 47](#_Toc500629442)

[Tabelle 54 Testergebnis T-001 59](#_Toc500629443)

# 1 Ziel

* 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 verkauft sind
* Der Benutzer kann seine Einkäufe und Verkäufe bewerten
* Das Item-Objekt (Angebot) wurde durch die Attribute Kategorie und Zustand erweitert
* Ein Benutzer mit der Rolle 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-Security umgesetzt. Ein weiteres Thema, mit welchem ich mich befasst habe, ist die AJAX-Technologie.

In Kapitel Implementierung werde ich Näher auf diese Themen eingehen.

# 2 Analyse

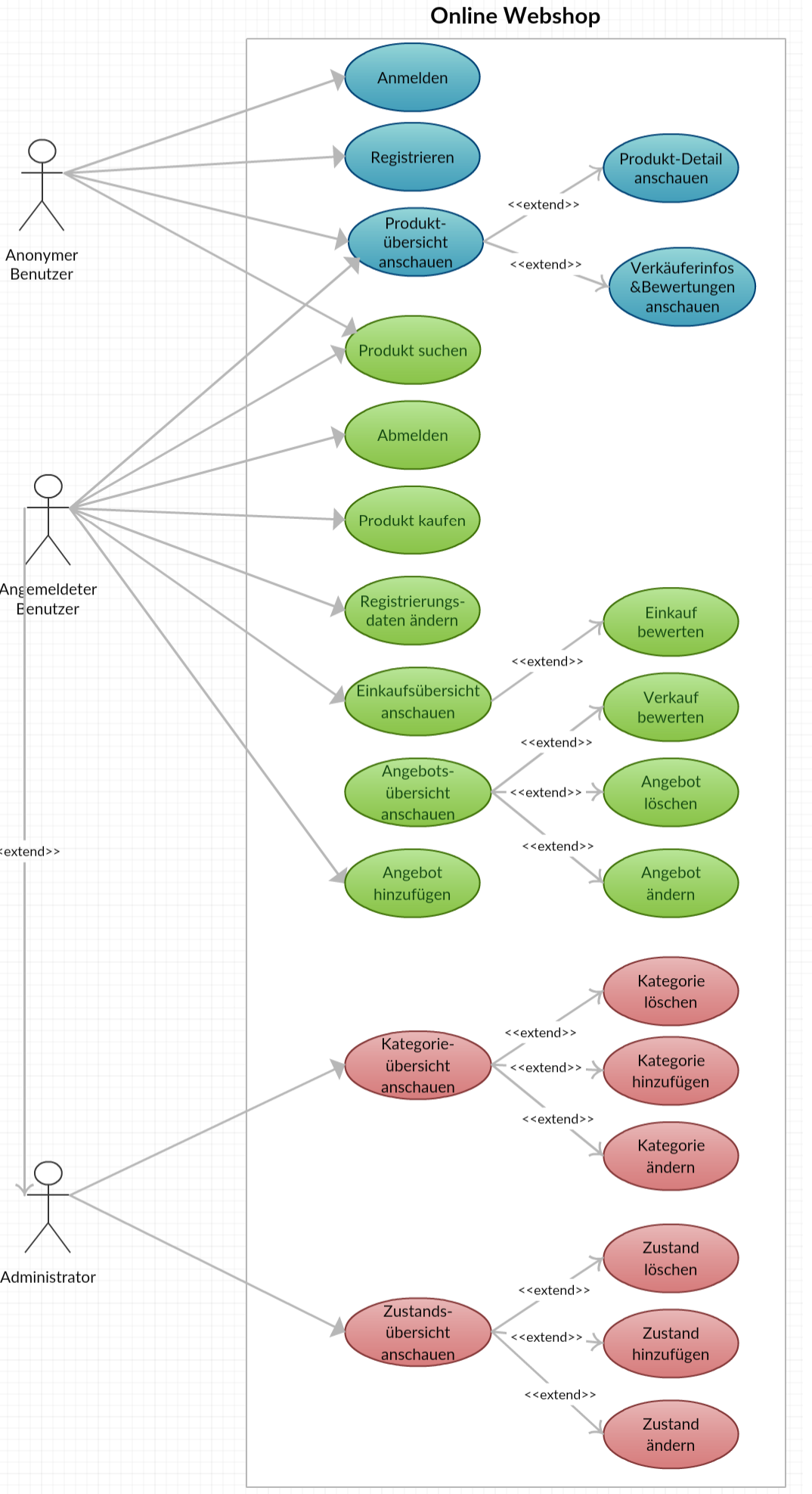


Abbildung 1 UseCase-Diagramm

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 ein Angebot löschen |
| 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 eines seiner Angebote ändern |
| Vorbedingung | Der Benutzer muss angemeldet sein  Das Angebot muss von diesem Benutzer sein  Das Produkt darf noch nicht verkauft sein |
| Eingehende Informationen | Entspechende Änderungen |
| 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 ändern (UseCase 15)

|  |  |
| --- | --- |
| 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

Das Fachklassenmodell dient in erster Linie für ein besseres Verständnis der zu entwickelnden Software für Fachabteilungen und Entscheidungsträgern. Es wird hier der grundsätzliche strukturelle Zusammenhang ohne grossen Detailierungsgrad beschrieben. Der Fokus liegt hier bei der Darstellung der Beziehungen zwischen den Objekten.

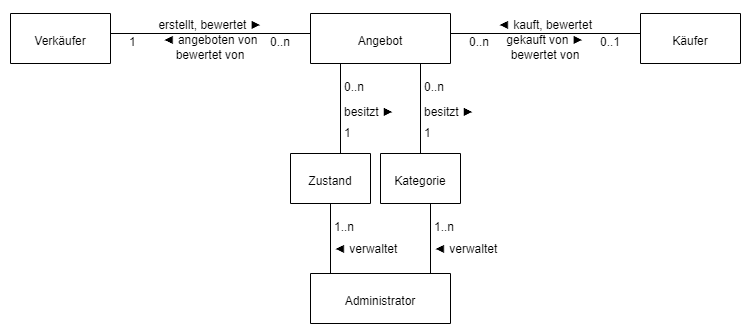


Abbildung 2 Fachklassenmodell

# 3 Softwareentwurf

* 1. Softwarearchitektur

Der Onlinehsop 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.

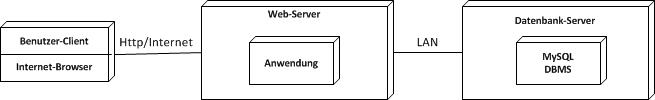
In meinem JSF-Projekt ist das MVC-Mapping wie folgt:

* **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 3 MVC-Muster JSF

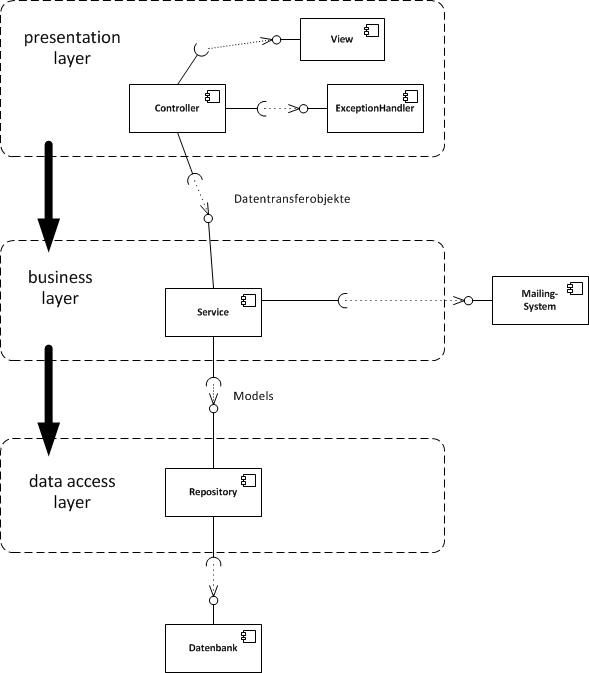


Die Software läuft auf einem Webserver (Glassfish) und die Datenbank auf einem Datenbank-Server. Für dieses Projekt wurde die Software inklusive der Datenbank jedoch nur lokal getestet (localhost).



TODO

Für die Repositories wird das JPA-Repository, welches von der Java-Community sehr empfohlen wird, eingesetzt.



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.

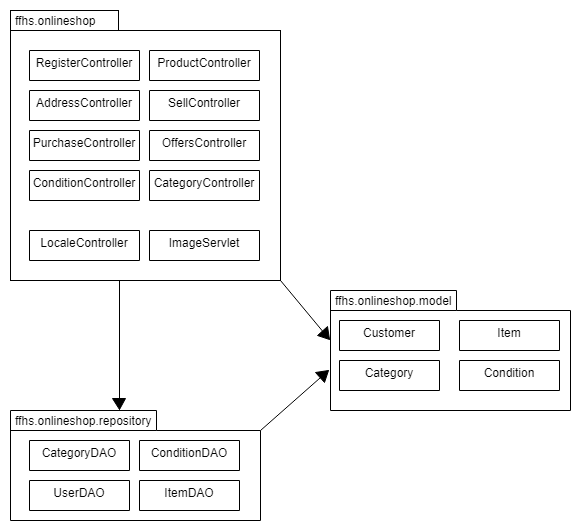


Abbildung 4 Package-Struktur

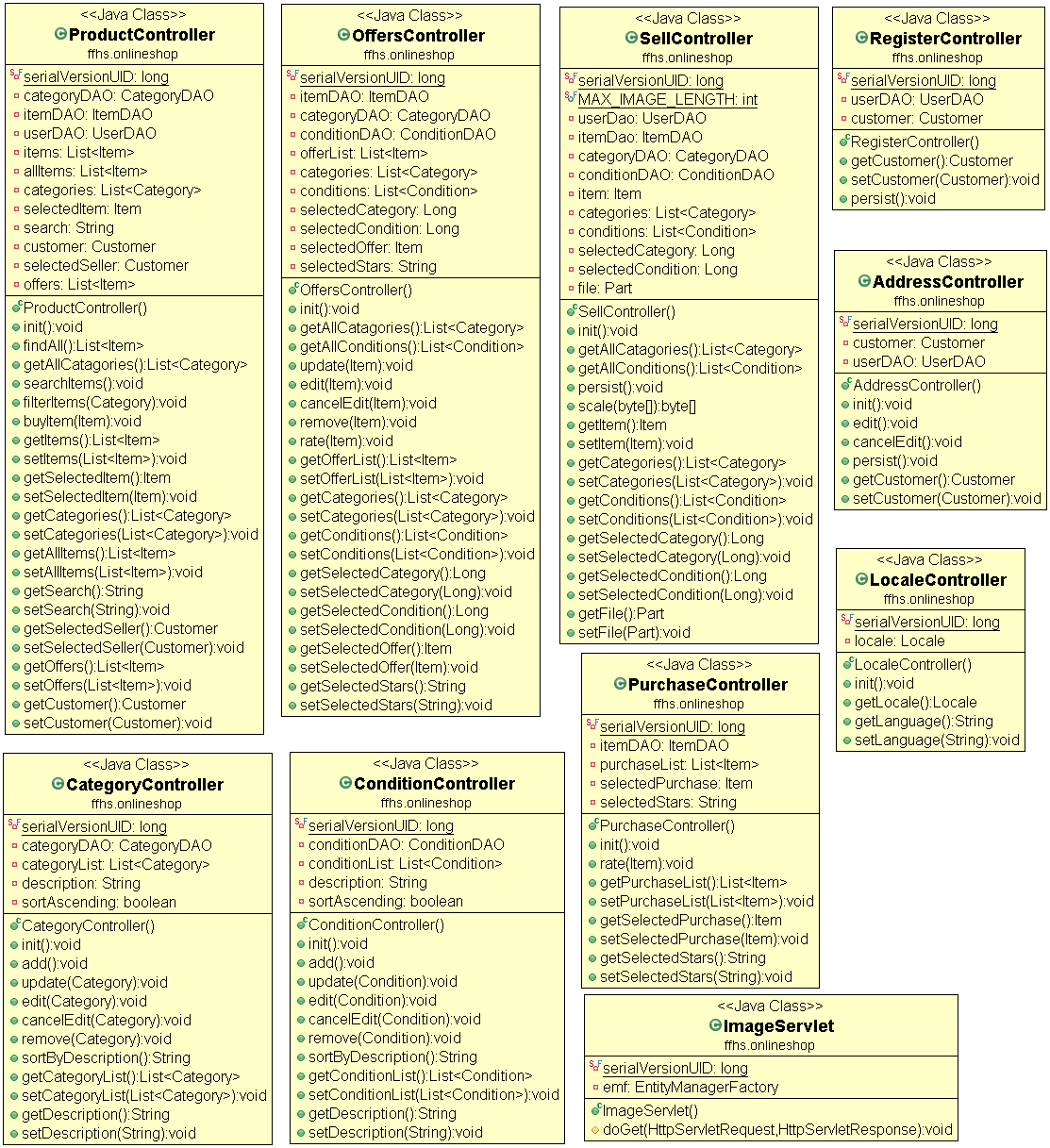


Abbildung 5 Klassendiagramm package ffhs.onlineshop (Controllers)

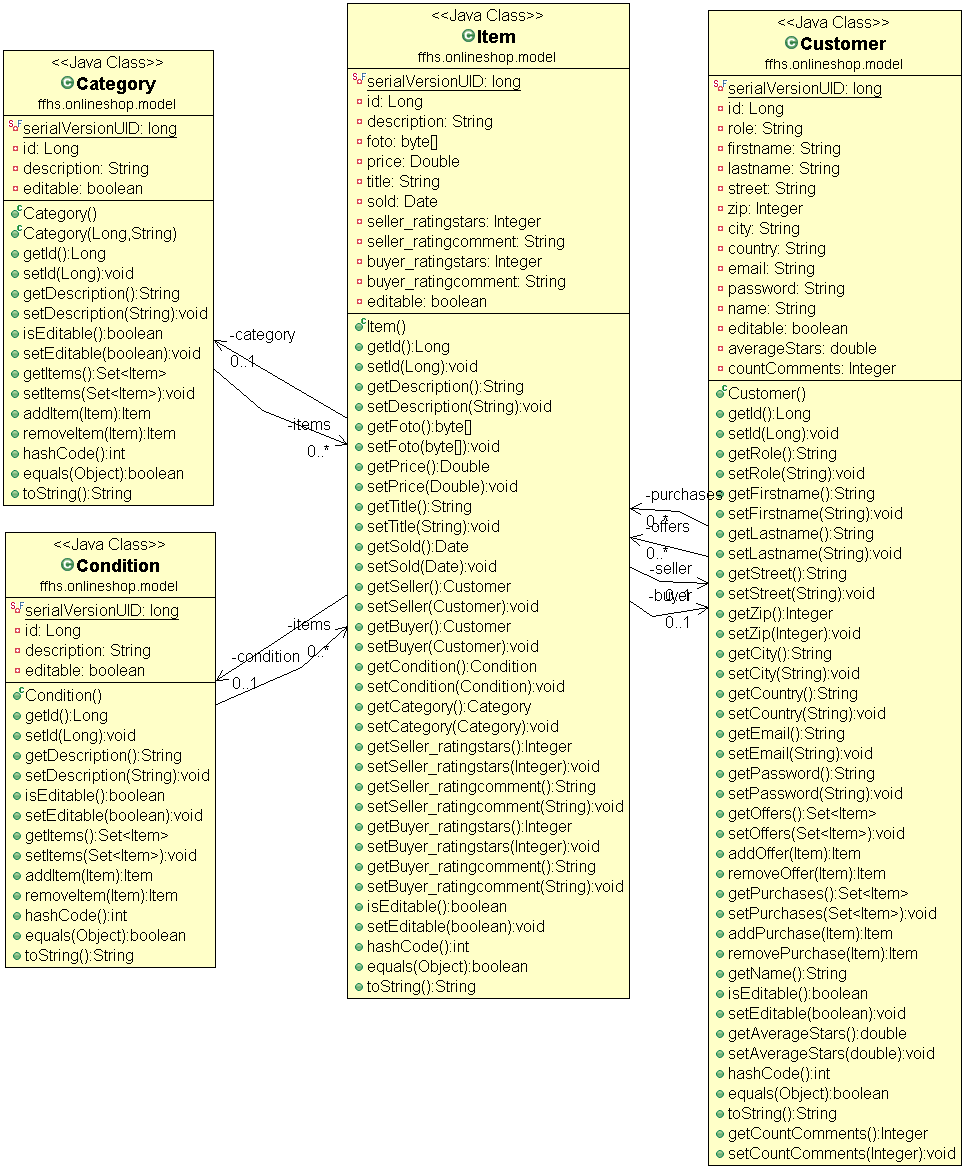


Abbildung 6 Klassendiagramm package ffhs.onlineshop.model (Models)

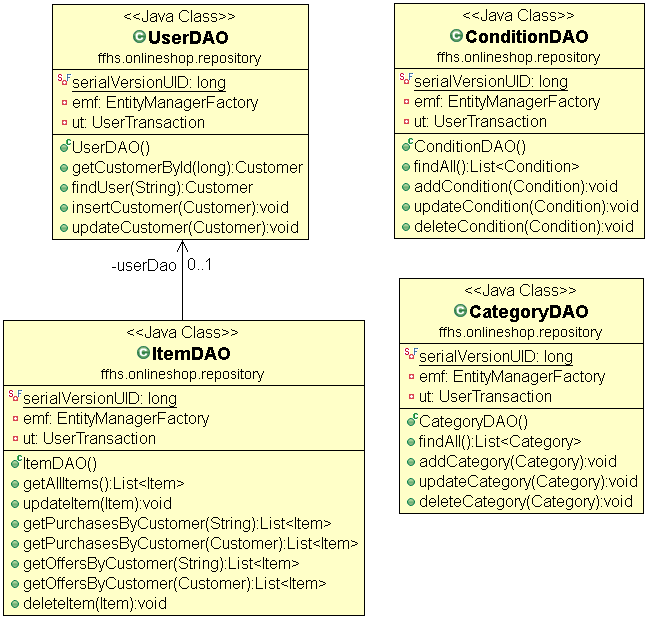


Abbildung 7 Klassendiagramm ffhs.onlineshop.repository (Repositories)

5.2 Datenmodell

Folgendes Datenmodell zeigt das Datenbankschema mit seinen Tabellen und Beziehungen, d.h. die physische Struktur auf der Datenbank.

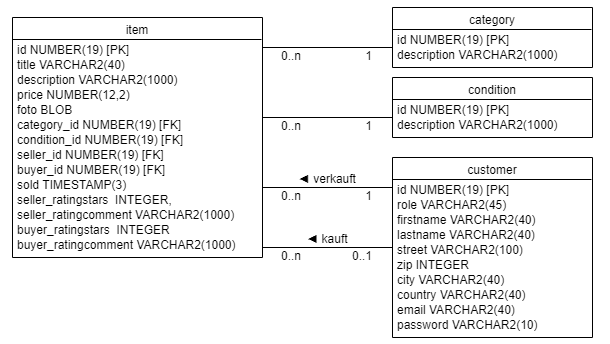


Abbildung 8 Diagramm Datenbank-Schema

Zwischen der Entität item und der Entität category ist eine many-to-one Beziehung vorhanden. Ein item hat genau eine category. Eine category kann keinem, einem oder mehreren items zugewiesen werden.

Zwischen der Entität item und der Entität condition gibt es ebenfalls eine many-to-one Beziehung. Ein item hat genau eine condition. Eine condition kann keinem, einem oder mehreren items zugewiesen werden.

Zwischen item und customer gibt es zwei many-to-one Beziehungen, je eine für die Rolle Verkäufer und Käufer.

5.3 Sequenzmodell

...

5.5 Speziﬁkation der Bedienoberﬂächen (GUI-Skizzen)

Für die Bedienoberflächen wurden Skizzen mit dem Werkzeug Balsamiq (<https://balsamiq.com/download/>) erstellt. Diese Skizzen 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 sowie der Preis ersichtlich. Wenn der Benutzer sich anmeldet ist auch 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. Zusätzlich ist bei einem verkauften Produkt der Kaufen-Button nicht mehr ersichtlich.

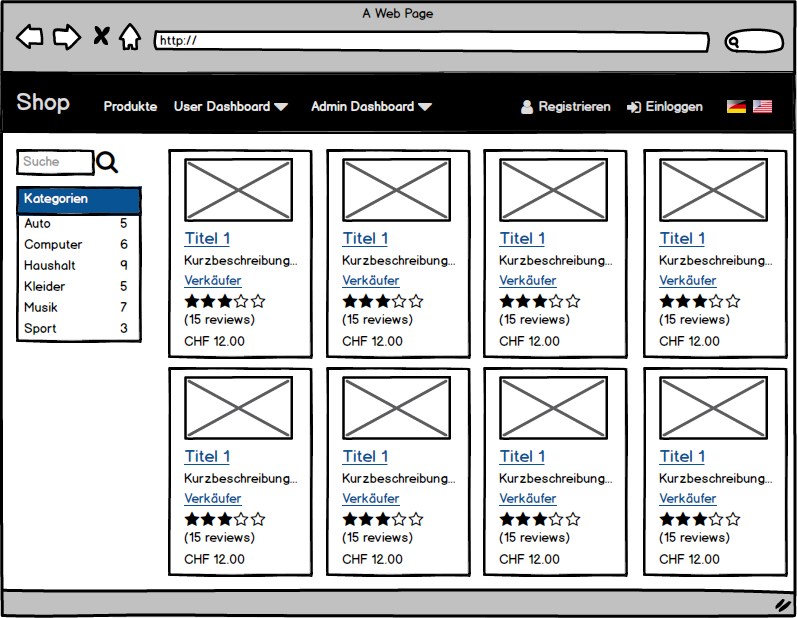


Abbildung 9 GUI Produktseite

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 sowie der Peis ersichtlich. Der Kaufen-Button ist nur ersichtlich, wenn der Benutzer angemeldet ist.

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

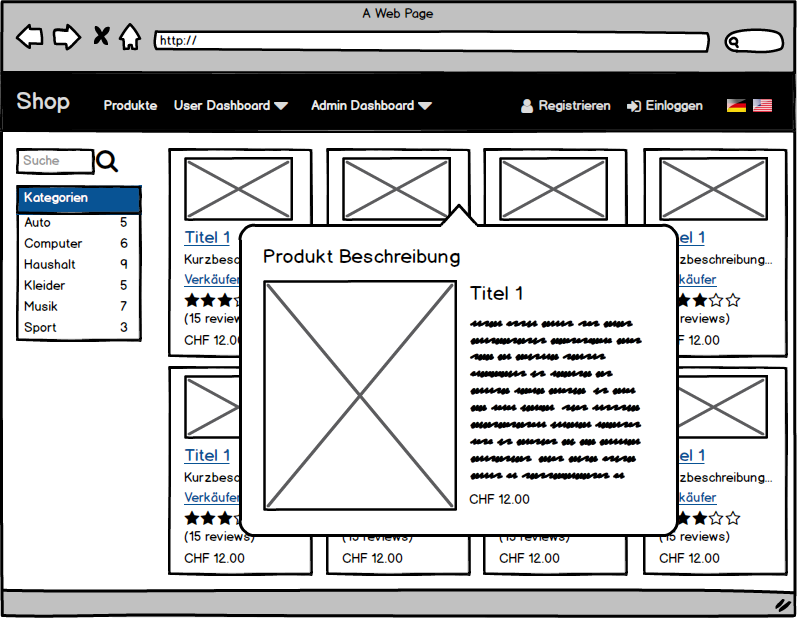


Abbildung 10 GUI Produktdetail

Registrieren

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

Folgende Daten sind einzugeben: Vorname, Nachname, Strasse, PLZ, Ort, Land, E-Mail, Kennwort. Alle Felder sind obligatorisch und werden validiert. Das Kennwort muss mindestens 6 Zeichen enthalten.

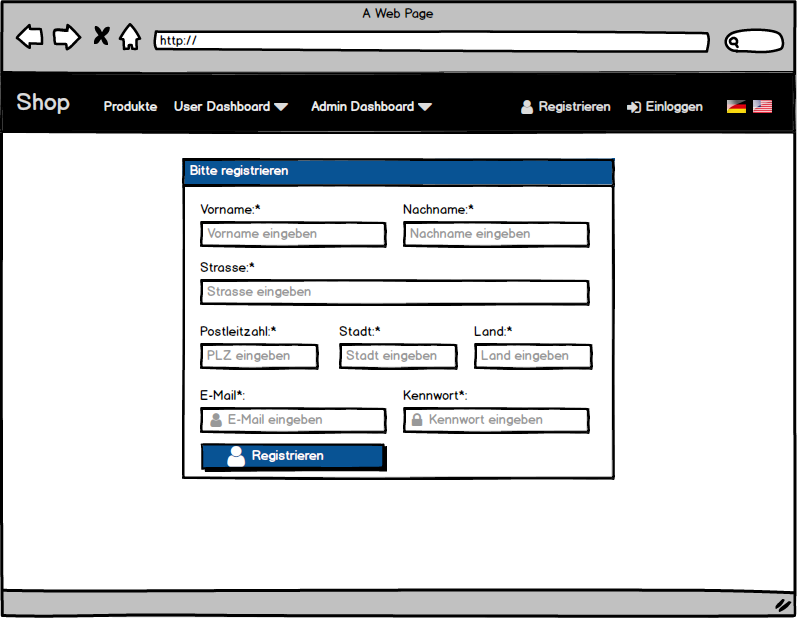


Abbildung 11 GUI Registrieren

Login

Es sind folgende Daten einzugeben: E-Mail und Kennwort.

Alle Felder sind obligatorisch und werden validiert. Sobald ein Benutzer sich erfolgreich angemeldet hat, sind die Menüpunkte Registrieren und Einloggen bis zur Abmeldung nicht mehr ersichtlich.

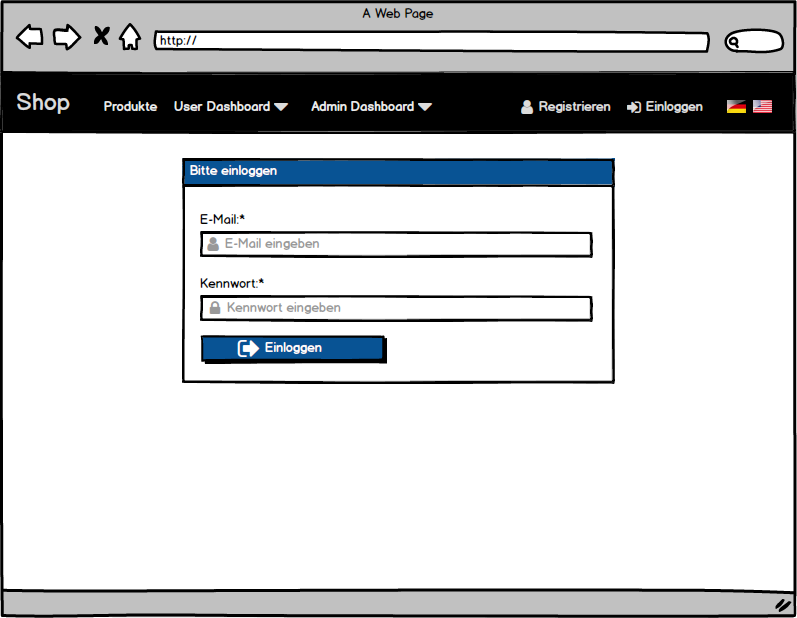


Abbildung 12 GUI Einloggen

User Dashbord – Meine Adresse

Diese Funktion benötigt User-Benutzerrechte. Wenn Meine Adresse via Menü ausgewählt wird und noch keine Berechtiungsüberprüfung stattfand, wird automatisch auf die Login-Seite weitergeleitet.

Die Adresse wird im Lese-Modus angezeigt. Mit Bearbeiten gelangt man in den Bearbeitungs-Modus.

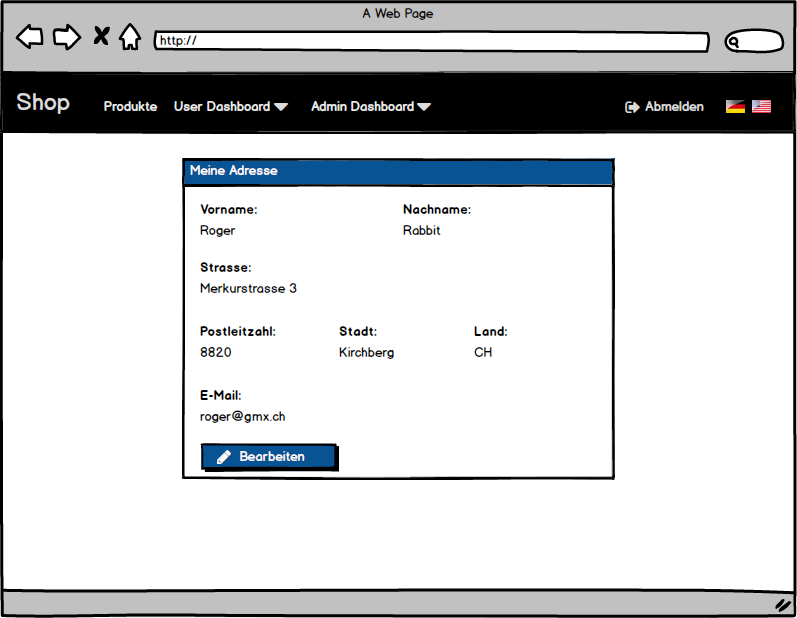


Abbildung 13 GUI Meine Adresse

Im Bearbeitungs-Modus sind folgende Daten einzugeben: Vorname, Nachname, Strasse, PLZ, Ort, Land, E-Mail, Kennwort. Alle Felder sind obligatorisch und werden validiert. Das Kennwort muss mindestens 6 Zeichen enthalten.

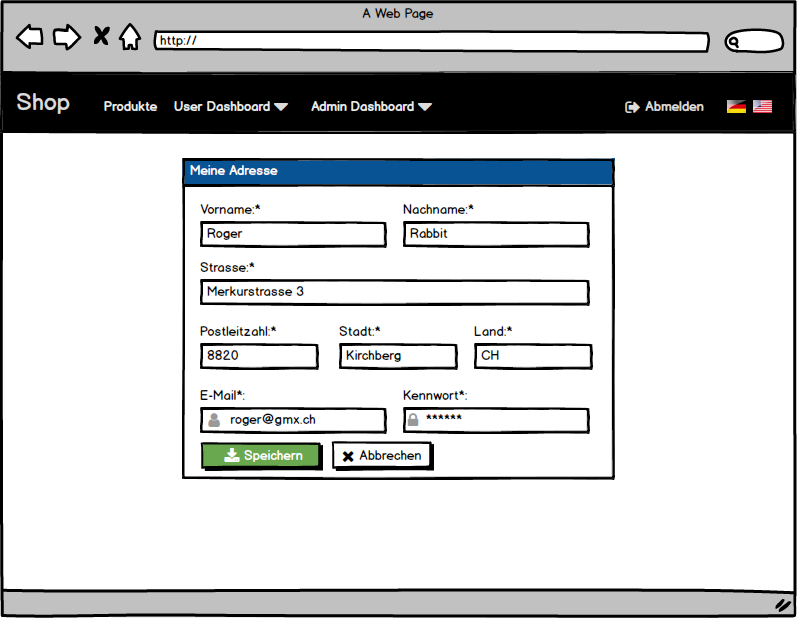


Abbildung 14 GUI Meine Adresse bearbeiten

User Dashbord – Meine Einkäufe

Diese Funktion benötigt User-Benutzerrechte. Wenn Meine Einkäufe via Menü ausgewählt wird und noch keine Berechtiungsüberprüfung stattfand, wird automatisch auf die Login-Seite weitergeleitet.

Es werden alle Einkäufe des angemeldeten Benutzers aufgelistet. Es sind folgende Felder zu sehen: Gekauft Am, Kategorie, Zustand, Preis, Verkäufer, Meine Bewertung, Titel.

Im Feld ‘Meine Bewertung‘ wird die Sterne-Bewertung angezeigt. Wenn der Benutzer noch keine Bewertung abgegeben hat, kann er dies mittels Bewertungs-Button jederzeit duchführen. Die Bewertung kann pro Einkauf nur einmal abgegeben werden. Eine einmal abgegebene Bewertung kann nicht mehr geändert werden.

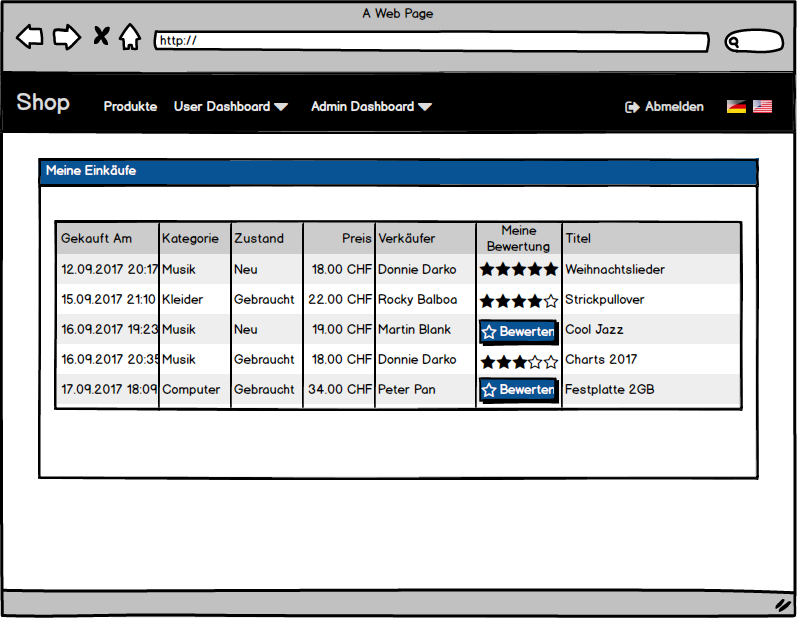


Abbildung 15 GUI Meine Einkäufe

Bewertung abgeben:

Um den Einkauf zu bewerten, kann der Benutzer auf den Button Bewerten klicken. Im aufgepopten Fenster kann er ein Sterne-Rating (ganzzahliger Wert 1 bis 5) sowie einen Kommentar abgeben.

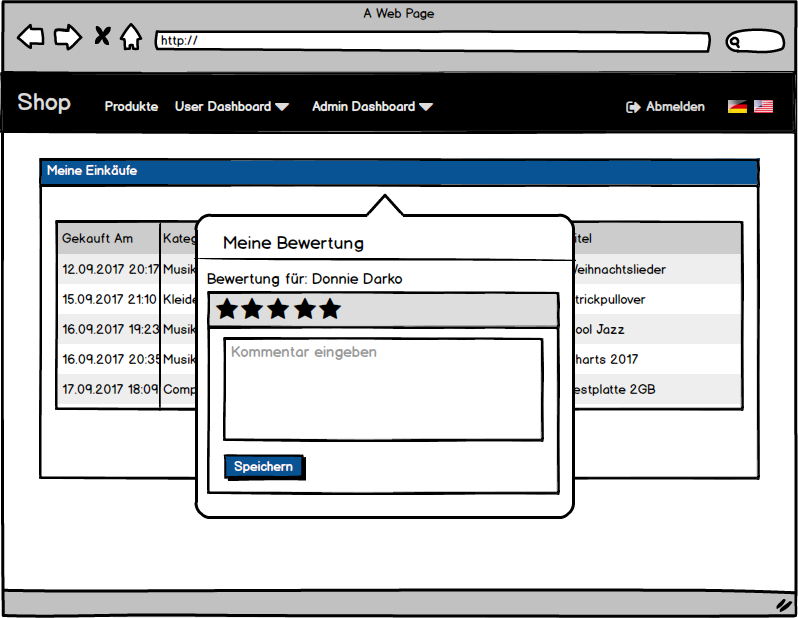


Abbildung 16 GUI Bewertung abgeben

User Dashbord – Meine Angebote

Diese Funktion benötigt User-Benutzerrechte. Wenn Meine Angebote via Menü ausgewählt wird und noch keine Berechtiungsüberprüfung stattfand, wird automatisch auf die Login-Seite weitergeleitet.

Es werden alle Angebote des angemeldeten Benutzers aufgelistet. Es sind folgende Felder zu sehen: Verkauft Am, Kategorie, Zustand, Preis, Käufer, Meine Bewertung, Titel.

Im Feld ‘Meine Bewertung‘ wird die Sterne-Bewertung angezeigt. Wenn der Benutzer noch keine Bewertung abgegeben hat, kann er dies mittels Bewertungs-Button jederzeit duchführen. Die Bewertung kann pro Verkauf nur einmal abgegeben werden. Eine einmal abgegebene Bewertung kann nicht mehr geändert werden. Die Bewertung kann nur für Angebote abgegeben werden, die gekauft wurden.

Zusätzlich stehen für noch nicht verkaufte Angebote zwei Funktionen zur Verfügung: Bearbeiten und Löschen. Bevor ein Angebot gelöscht wird, wird einmal nachgefragt, ob das Angebot wirklich gelöscht werden möchte.

Im Bearbeitungs-Modus können folgende Felder bearbeitet werden: Kategorie, Zustand, Preis, Titel. Es sind alle Felder obligatorisch und werden valdiert.

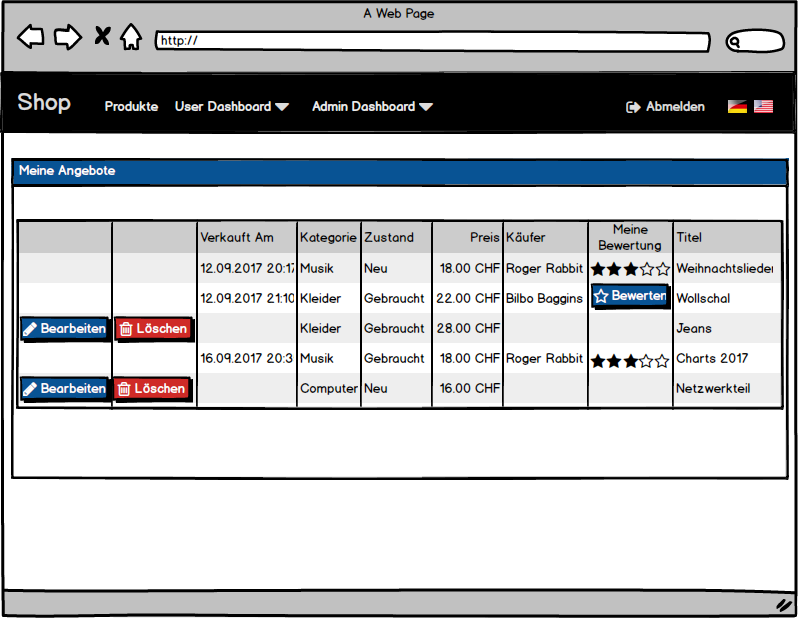


Abbildung 17 GUI Meine Angebote

User Dashbord – Neues Angebot

Ein angemeldeter Benutzer kann ein neues Angebot via Dashboard – Neues Angebot erstellen.

Im Bearbeitungs-Modus sind folgende Daten einzugeben: Kategorie, Zustand, Titel, Preis, Bezeichnung und Foto. Alle Felder sind obligatorisch und werden validiert. Wenn auf Verkaufen geklickt wird, ist das Angebot gespeichert und steht zum Verkauf bereit.

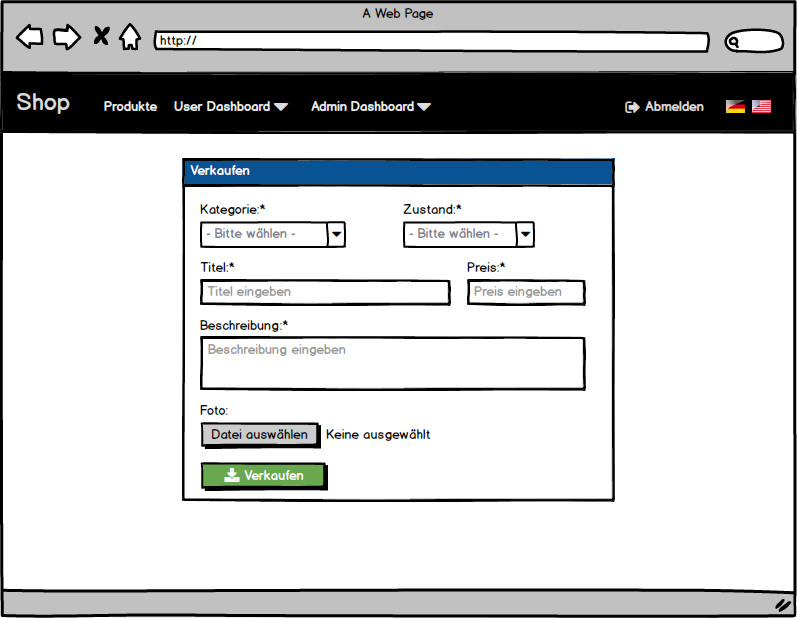


Abbildung 18 GUI Verkaufen

Admin Dashboard – Verwaltung Kategorien

Um Kategorien verwalten zu können, werden Admin-Berechtigung benötigt. Wenn der Benutzer noch nicht eingeloggt ist, wird automatisch auf die Login-Seite umgeleitet. Wenn der Benutzer angemeldet ist, aber zuwenig Berechtigung besitzt, wird angezeigt, dass der Benutzer kein Zugriff hat.

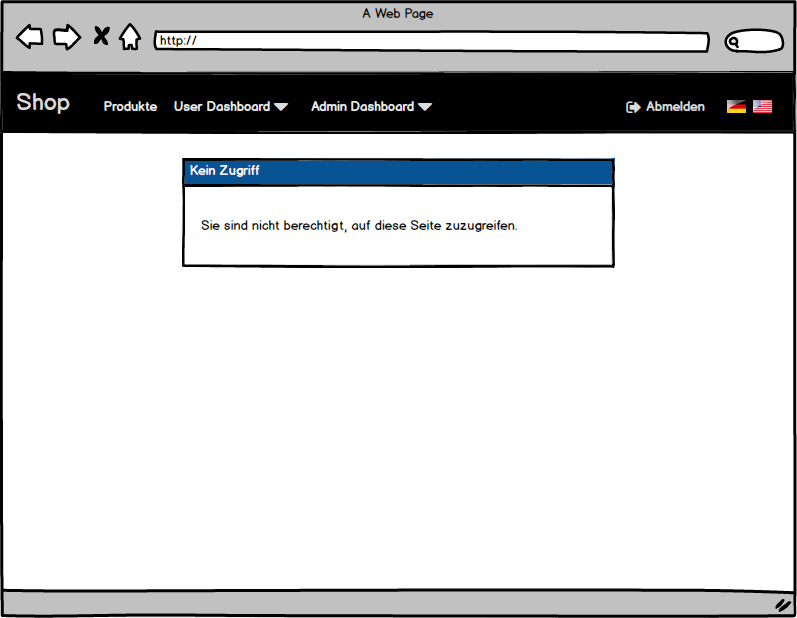


Abbildung 19 GUI Kein Zugriff

Wenn der angemeldete Benutzer Admin-Berechtigung besitzt, werden ihm alle Kategorien angezeigt. Die Liste kann nach Kategorie-Beschreibung alphabetisch aufsteigend sowie absteigend sortiert werden. Die Kategorie-Beschreibung kann direkt in der Liste geändert werden, indem der Benutzer auf den Button Bearbeiten klickt. Mit der Löschen-Funktion kann der Benutzer eine Kategorie löschen. Bevor eine Kategorie gelöscht wird, wird nachgefragt, ob die Kategorie wirklich gelöscht werden soll.

Zusätzlich besteht hier die Möglichkeit neue Kategorien hinzuzufügen, indem der Benutzer eine neue Beschreibung einträgt und auf Hinzufügen klickt.

Einziges obgligatorisches Feld ist die Beschreibung. Die Beschreibung wird validiert.

Beispiel:

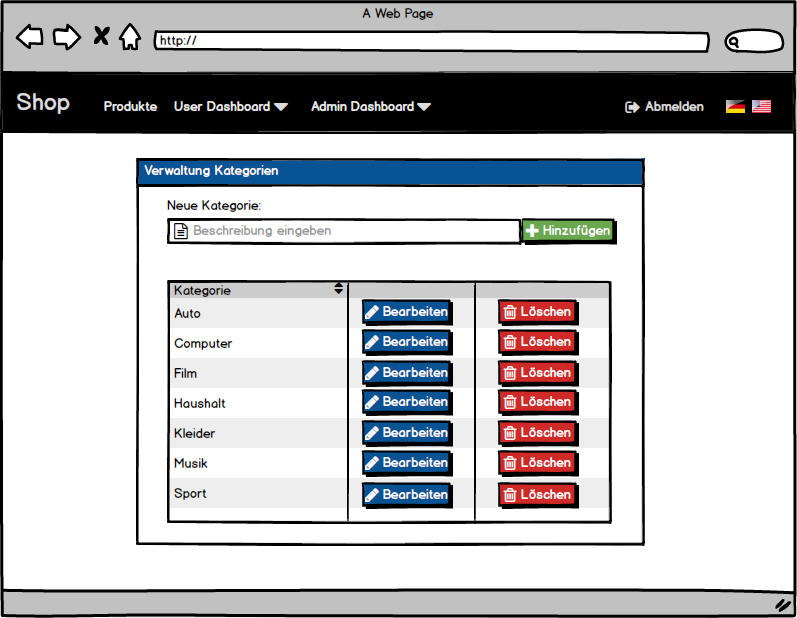


Abbildung 20 GUI Verwaltung Kategorien

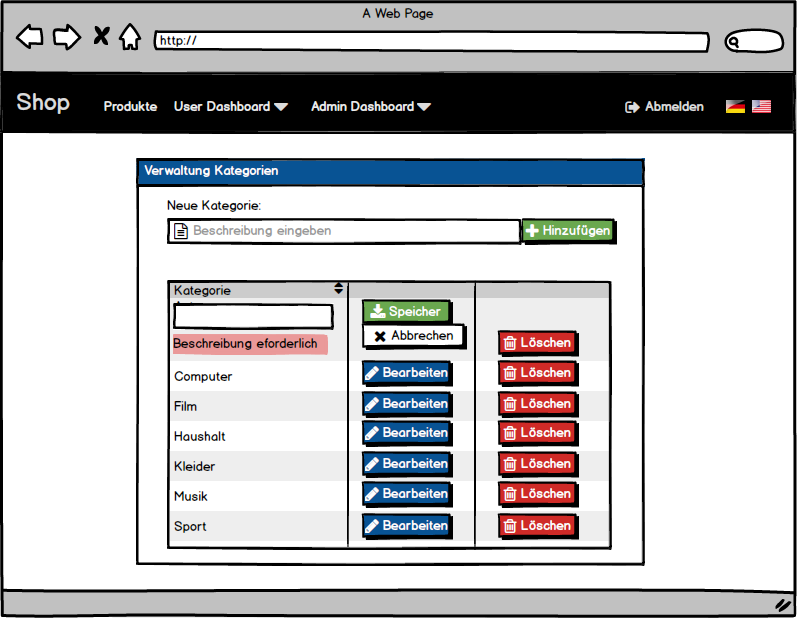


Abbildung 21 GUI Verwaltung Kategorien - Validerungs-Besipiel

Admin Dashboard – Verwaltung Zustand

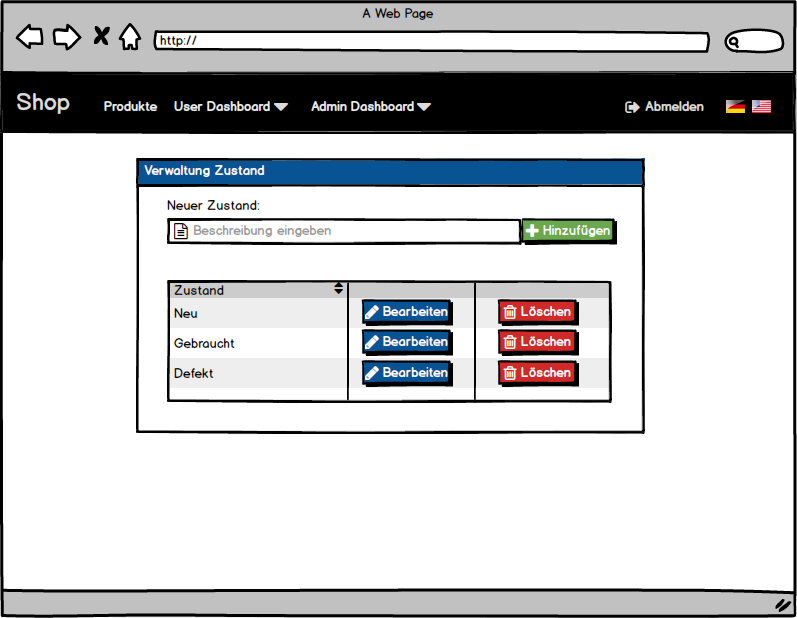


Abbildung 22 GUI Verwaltung Zustand

Logout

Sobals ich ein Benutzer angemeldet hat, erscheint die Abmelde-Funktion im Menü.

Wenn sich ein Benutzer abmeldet, erscheint das Login-Fenster mit dem Hinweis, dass der Benutzer erfolgreich abgemeldet wurde.

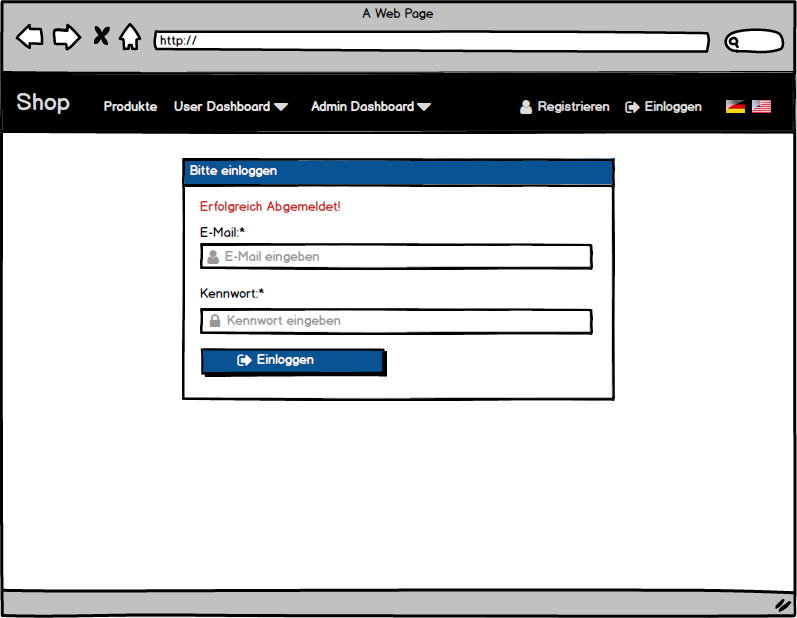


Abbildung 23 GUI Abgemeldet

5.4 Testkonzept

Um die Zuverlässigkeit und die Qualität der Software und des gesamten Systems zu gewährleisten, wurden Testfälle durchgeführt und protokolliert. Um sicherzustellen, dass alle Anwendungsfälle komplett vom GUI bis in die Datenbank einwandfrei funktionieren, wurden manuelle Tests durchgeführt (User-Tests).

In diesem Projekt wurden keine Komponententests durchgeführt. Für ein Projekt, dass später in der Produktion eingesetzt wird, empfehle ich für die kontinuierliche Qualitätssicherung unbedingt Komponententests zu erstellen (Unit-Tests).

|  |  |  |
| --- | --- | --- |
| Teststufe |  | Wer |
| Komponententests | White- Tests (Unit-Test) | Nicht durchgeführt |
| Manuelle Tests | Anwender-Tests | Birgit Sturzenegger |

Tabelle 25 Testkonzept – Teststufen

|  |  |  |
| --- | --- | --- |
| Art von Test | Wird durchgeführt | Wer |
| Integrationstests | Nein | - |
| Systemtests | Nein | - |
| Regressionstest | Nein | - |
| Abnahmetest | Nein | - |
| Komponententests | Nein | - |

Tabelle 26 Testkonzept – Test-Arten

Manuelle Tests

Mit den manuellen Tests wurde begonnen, sobald eine testbare Komponente fertiggestellt wurde. Anbei ist das Konzept für die manuellen Tests beschrieben.

|  |  |
| --- | --- |
| **Thema** | **Erläuterung** |
| Beschreibung des zu testenden Systems | Die Hauptfunktion der Webapplikation Webshop sind das Kaufen und Verkaufen von Produkten. Damit ein Kauf getätigt werden kann, müssen Angebote vorhanden sein. Da Käufe nur von angemeldeten Benutzern durchgeführt werden können, müssen Benutzerkonten vorhanden sein. Daher gehören zum System nebst der Verkauf- und Kauf-Funktionalitäten, eine Registrierungsmöglichkeit für neue Benutzer. |
| Definition der Testumgebung | Der Tester benötigt:   * Einen nach [Sal] konfigurierten Glassfish-Server Version 4 * eine nach [Sal] konfigurierte Oracle-Installation * die zu testende Applikation als WAR Datei * das SQL-Script für die Erstellung der Datenbank * Zugangsdaten des Benutzers mit Admin-Rolle |
| Vorbereitung der Tests | Einige Funktionen der Applikation benötigen einen Benutzer mit Admin-Rolle. Daher wird mittels beigefügtem Datenbank-Script bereits ein Benutzer mit dieser Rolle erstellt. |
| Verantwortlichkeiten, Aufgaben des Testers | Der Tester dokumentiert seine Ergebnisse in mit folgender Struktur:   * Wann wurde der Test durchgeführt * Welcher Anwendungsfall wurde getestet * Wie ist das Ergebnis des Tests |
| zu testende Eigenschaften, nicht zu testende Eigenschaften | Es sind alle Funktionalitäten zu testen.  Es soll auch auf allfällige Schreibfehler geachtet werden.  Es soll darauf geachtet werden, dass allenfalls auftretende Fehlermeldungen sinnvoll und nützlich für den Benutzer sind.  Bei den Test sollen bewusst Fehler provoziert werden (z.B. fehlende Angaben oder falsche Angaben), damit die Qualität der Software beurteilt werden kann. |
| Definition der Testobjekte | Es sind alle Anwendungsfälle gemäss UseCase-Diagramm zu testen.  Damit das System sinnvoll getestet werden kann, sind mehrere Benutzer und Angebote anzulegen. |
| Risikobetrachtung, Priorisierung | Einige Bereiche dienen als Basis für den Webshop. Diese sind besonders gut zu testen, da sie für die Gesamtfunktionalität ein grosses Risiko darstellen:   * Benutzer registrieren * Anmelden * Angebot erstellen * Angebot kaufen |
| Zeit-/Ressourcenplanung; Reihenfolge der Aktivitäten | Sobald eine Komponente fertig programmiert ist, kann sie getestet werden. |

Tabelle 27 Testkonzept – manuelle Tests

Testfälle

| ID / Bezeichnung | **T-001** | Produkt suchen (anonymer Benutzer) |
| --- | --- | --- |
| Beschreibung | Der Tester sucht als anonymer Benutzer ein bestimmtes Produkt | |
| Testvoraussetzung | Es sind bereits Angebote vorhanden | |
| Testschritte | 1. Suchbegriff eingeben 2. Auf Suchen klicken | |
| Erwartetes Ergebnis | Die Produkte mit dem Suchbegriff im Titel werden angezeigt.  Der Kaufen-Button ist bei allen Angeboten ausgeblendet.  Vekaufte Produkte sind oben rechts rot markiert und haben dort den Text Sold. Nicht verkaufte Produkte sind oben rechts grün markiert. | |

Tabelle 28 Testfall - Produkt suchen (anonymer Benutzer)

| ID / Bezeichnung | **T-001** | Produkt nach Kategorie filtern (anonymer Benutzer) |
| --- | --- | --- |
| Beschreibung | Der Tester filtert als anonymer Benutzer die Produkte nach einer Kategorie | |
| Testvoraussetzung | Es sind bereits Angebote vorhanden | |
| Testschritte | 1. Auf eine Kategorie klicken | |
| Erwartetes Ergebnis | Die Produkte mit der entsprechenden Kategorie werden angezeigt.  Der Kaufen-Button ist bei allen Angeboten ausgeblendet. Vekaufte Produkte sind oben rechts rot markiert und haben dort den Text Sold. Nicht verkaufte Produkte sind oben rechts grün markiert. | |

Tabelle 29 Testfall - Produkt nach Kategorie filtern (anonymer Benutzer)

| ID / Bezeichnung | **T-001** | Produktdetail anschauen (anonymer Benutzer) |
| --- | --- | --- |
| Beschreibung | Der Tester schaut sich als anonymer Benutzer die Details eines Produktes an | |
| Testvoraussetzung | Es sind bereits Angebote vorhanden | |
| Testschritte | 1. Produkt auswählen 2. Auf Produkt-Titel klicken | |
| Erwartetes Ergebnis | Ein Popup-Fenster mit Details zum Produkt erscheint.  Der Kaufen-Button ist ausgeblendet. | |

Tabelle 30 Testfall - Produktdetail anschauen (anonymer Benutzer)

| ID / Bezeichnung | **T-001** | Verkäufer-Infos und Bewertung anschauen (anonymer Benutzer) |
| --- | --- | --- |
| Beschreibung | Der Tester schaut sich als anonymer Benutzer die Verkäufer-Infos und Bewertungen an | |
| Testvoraussetzung | Es sind bereits Angebote vorhanden | |
| Testschritte | 1. Produkt auswählen 2. Auf Verkäufer-Name klicken | |
| Erwartetes Ergebnis | Ein Popup-Fenster mit dem Namen und Adresse des Verkäufers erscheint. | |

Tabelle 31 Testfall - Verkäufer-Infos und Bewertung anschauen(anonymer Benutzer)

| ID / Bezeichnung | **T-001** | Eine geschützte Seite wählen (anonymer Benutzer) |
| --- | --- | --- |
| Beschreibung | Der Tester wählt als anonymer Benutzer eine geschützte Seite an | |
| Testvoraussetzung |  | |
| Testschritte | 1. Diese Menüpunkte anwählen:  * Meine Adresse * Meine Einkäufe * Meine Angebote * Neues Angebot * Verwaltung Kategorien * Verwaltung Zustand | |
| Erwartetes Ergebnis | Das Login-Fenster wird angezeigt | |

Tabelle 32 Testfall - Eine geschützte Seite wählen (anonymer Benutzer)

| ID / Bezeichnung | **T-001** | Registirieren |
| --- | --- | --- |
| Beschreibung | Der Tester registriert sich als Benutzer | |
| Testvoraussetzung |  | |
| Testschritte | 1. Menüpunkt Registrieren wählen 2. Felder ausfüllen 3. Registrieren klicken | |
| Erwartetes Ergebnis | Bei unkorrekten Eingaben werden die Fehler angezeigt.  Bei korrekten Eingaben wird die Meldung “Erfolgreich registriert!“  angezeigt. | |

Tabelle 33 Testfall - Registirieren

| ID / Bezeichnung | **T-001** | Anmelden |
| --- | --- | --- |
| Beschreibung | Der Tester meldet sich als Benutzer an | |
| Testvoraussetzung |  | |
| Testschritte | 1. Menüpunkt Einloggen wählen 2. Felder ausfüllen 3. Einloggen klicken | |
| Erwartetes Ergebnis | Bei unkorrekten Eingaben wird die Meldung “Login fehlgeschlagen. Grund: Bad credentials“ angezeigt.  Bei korrekten Eingaben wird auf die Produktseite weitergeleitet. Wenn vorher eine andere Seite mit Zugriffsberechtigung gewählt wurde, wird nach einem erfolgreichen Login dorthin weitergeleitet.  Die Menüpunkte Registrieren und Einloggen sind nicht mehr ersichtlich, dafür erscheint der Menüpunkt Abmelden. | |

Tabelle 34 Testfall - Anmelden

| ID / Bezeichnung | **T-001** | Produkt suchen (angemeldeter Benutzer) |
| --- | --- | --- |
| Beschreibung | Der Tester sucht als angemeldeter Benutzer ein bestimmtes Produkt | |
| Testvoraussetzung | Es sind bereits Angebote vorhanden | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Suchbegriff eingeben 3. Auf Suchen klicken | |
| Erwartetes Ergebnis | Die Produkte mit dem Suchbegriff im Titel werden angezeigt  Der Kaufen-Button ist bei allen Angeboten, welche noch nicht verkauft wurden, ersichtlich. Bei bereits verkauften Produkten ist der Kaufen-Button nicht ersichtlich.  Vekaufte Produkte sind oben rechts rot markiert und haben dort den Text Sold. Nicht verkaufte Produkte sind oben rechts grün markiert. | |

Tabelle 35 Testfall - Produkt suchen (angemeldeter Benutzer)

| ID / Bezeichnung | **T-001** | Produkt nach Kategorie filtern (angemeldeter Benutzer) |
| --- | --- | --- |
| Beschreibung | Der Tester filtert als angemeldeter Benutzer die Produkte nach einer Kategorie | |
| Testvoraussetzung | Es sind bereits Angebote vorhanden | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Auf eine Kategorie klicken | |
| Erwartetes Ergebnis | Die Produkte mit der entsprechenden Kategorie werden angezeigt  Der Kaufen-Button ist bei allen Angeboten, welche noch nicht verkauft wurden, ersichtlich. Bei bereits verkauften Produkten ist der Kaufen-Button nicht ersichtlich.  Vekaufte Produkte sind oben rechts rot markiert und haben dort den Text Sold. Nicht verkaufte Produkte sind oben rechts grün markiert. | |

Tabelle 36 Testfall - Produkt nach Kategorie filtern (angemeldeter Benutzer)

| ID / Bezeichnung | **T-001** | Produktdetail anschauen (angemeldeter Benutzer) |
| --- | --- | --- |
| Beschreibung | Der Tester schaut sich als angemeldeter Benutzer die Details eines Produktes an | |
| Testvoraussetzung | Es sind bereits Angebote vorhanden | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Produkt auswählen 3. Auf Produkt-Titel klicken | |
| Erwartetes Ergebnis | Ein Popup-Fenster mit Details zum Produkt erscheint.  Der Kaufen-Button ist ersichtlich, falls das Produkt noch nicht verkauft ist, ansonsten ist dieser nicht ersichtlich. | |

Tabelle 37 Testfall - Produktdetail anschauen (angemeldeter Benutzer)

| ID / Bezeichnung | **T-001** | Registrierungsdaten ändern |
| --- | --- | --- |
| Beschreibung | Der Tester ändert seine Registrierungsdaten | |
| Testvoraussetzung |  | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Meine Adresse“ wählen 3. Auf Bearbeiten klicken 4. Felder ändern 5. Speichern drücken | |
| Erwartetes Ergebnis | Bei unkorrekten Eingaben werden die Fehler angezeigt.  Bei korrekten Eingaben wird die Meldung “Erfolgreich aktualisiert“ angezeigt. Die aktualisierten Daten sind im Lese-Modus ersichtlich. | |

Tabelle 38 Testfall - Registrierungsdaten ändern

| ID / Bezeichnung | **T-001** | Angebot erstellen |
| --- | --- | --- |
| Beschreibung | Der Tester erstellt ein Angebot | |
| Testvoraussetzung |  | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Neues Angebot“ wählen 3. Felder ausfüllen 4. Foto hochladen 5. Verkaufen drücken | |
| Erwartetes Ergebnis | Bei unkorrekten Eingaben werden die Fehler angezeigt.  Bei korrekten Eingaben wird die Meldung “Erfolgreich gespeichert!“ angezeigt. | |

Tabelle 39 Testfall - Angebot erstellen

| ID / Bezeichnung | **T-001** | Produkt kaufen |
| --- | --- | --- |
| Beschreibung | Der Tester kauft ein Produkt | |
| Testvoraussetzung | Es sind bereits Angebote vorhanden | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Produkte“ wählen 3. Bei einem Produkt auf Kaufen klicken | |
| Erwartetes Ergebnis | Die Markierung oben rechts wechselt beim gekauften Produkt von grün auf rot und es erscheint der Text “Sold“ in der roten Markierung. Der Kaufen-Button ist nicht mehr ersichtlich. | |

Tabelle 40 Testfall - Produkt kaufen

| ID / Bezeichnung | **T-001** | Einkaufsübersicht anschauen |
| --- | --- | --- |
| Beschreibung | Der Tester schaut sich seine Einkäufe an | |
| Testvoraussetzung |  | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Meine Einkäufe“ wählen | |
| Erwartetes Ergebnis | Eine Liste mit allen Einkäufen des angemeldeten Benutzers wird angezeigt. Bei allen Einkäufen, welche noch nicht bewertet wurden, ist der Bewerten-Button ersichtlich. | |

Tabelle 41 Testfall - Einkaufsübersicht anschauen

| ID / Bezeichnung | **T-001** | Einkauf bewerten |
| --- | --- | --- |
| Beschreibung | Der Tester bewertet einen seiner Einkäufe | |
| Testvoraussetzung | Es ist mindestens ein Produkt vom Tester gekauft worden.  Der Benutzer hat den entsprechende Einkauf noch nicht bewertet. | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Meine Einkäufe“ wählen 3. Bei einem Einkauf auf Bewerten drücken 4. Im Popup-Fenster den entsprechende Stern klicken (z.B für die Sterne-Bewertung 4: den 4.Stern drücken) 5. Kommentar eingeben 6. Speichern drücken | |
| Erwartetes Ergebnis | Das Popup-Fenster für die Abgabe der Bewertung öffnet sich. Nach dem Speichern, schliesst sich das Popup-Fenster. Der Button Bewerten ist beim entsprechenden Einkauf nicht mehr ersichtlich. Stattdessen wird dort jetzt die Sterne-Bewertung angezeigt. | |

Tabelle 42 Testfall - Einkauf bewerten

| ID / Bezeichnung | **T-001** | Angebotsübersicht anschauen |
| --- | --- | --- |
| Beschreibung | Der Tester schaut sich seine Angebote an | |
| Testvoraussetzung |  | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Meine Angebote“ wählen | |
| Erwartetes Ergebnis | Eine Liste mit allen Angeboten des angemeldeten Benutzers wird angezeigt. Bei allen Angeboten, welche verkauft und noch nicht bewertet wurden, ist der Bewerten-Button ersichtlich. Bei Angebote, welche noch nicht gekauft wurden, ist der Button Bearbeiten und Löschen ersichtlich. | |

Tabelle 43 Testfall - Angebotsübersicht anschauen

| ID / Bezeichnung | **T-001** | Angebot ändern |
| --- | --- | --- |
| Beschreibung | Der Tester ändert eines seiner Angebote | |
| Testvoraussetzung | Mindestens ein noch nicht verkauftes Angebot ist vorhanden | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Meine Angebote“ wählen 3. Bearbeiten-Button des entsprechenden Angebots drücken 4. Werte ändern 5. Speichern drücken | |
| Erwartetes Ergebnis | Die entsprechende Zeile geht zuerst in den Edit-Modus und nach dem Speichern wieder in den Lese-Modus. Nach dem Speichern sind die aktualisierten Daten ersichtlich. | |

Tabelle 44 Testfall - Angebot ändern

| ID / Bezeichnung | **T-001** | Verkauf bewerten |
| --- | --- | --- |
| Beschreibung | Der Tester bewertet einer seiner Verkäufe | |
| Testvoraussetzung | Es ist mindestens ein Angebot vom Tester verkauft worden.  Der Benutzer hat den entsprechende Verkauf noch nicht bewertet. | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Meine Angebote“ wählen 3. Bei einem Verkauf auf Bewerten drücken 4. Im Popup-Fenster den entsprechende Stern klicken (z.B für die Sterne-Bewertung 3: den 3.Stern drücken) 5. Kommentar eingeben 6. Speichern drücken | |
| Erwartetes Ergebnis | Das Popup-Fenster für die Abgabe der Bewertung öffnet sich. Nach dem Speichern, schliesst sich das Popup-Fenster. Der Button Bewerten ist beim entsprechenden Verkauf nicht mehr ersichtlich. Stattdessen wird dort jetzt die Sterne-Bewertung angezeigt. | |

Tabelle 45 Testfall - Verkauf bewerten

| ID / Bezeichnung | **T-001** | Angebot löschen |
| --- | --- | --- |
| Beschreibung | Der Tester löscht eines seiner Angebote | |
| Testvoraussetzung | Das Angebot darf noch nicht verkauft sein | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Meine Angebote“ wählen 3. Löschen-Button des entsprechenden Angebots drücken 4. Rückfrage zum Löschen bestätigen | |
| Erwartetes Ergebnis | Nach dem Bestätigen der Rückfrage ist das Angebot gelöscht und in der Liste nicht mehr ersichtlich. Beim Abbrechen der Rückfrage ist das Angebot nicht gelöscht und die Liste unverändert. | |

Tabelle 46 Testfall - Angebot löschen

| ID / Bezeichnung | **T-001** | Kategorie hinzufügen |
| --- | --- | --- |
| Beschreibung | Der Tester erstellt eine neue Kategorie | |
| Testvoraussetzung | Der Benutzer muss Admin-Berechtigung haben | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Verwaltung Kategorien“ wählen 3. Bezeichnung der neuen Kategorie eingeben 4. Hinzufügen drücken | |
| Erwartetes Ergebnis | Die neue Kategorie ist in der Liste ersichtlich. Das Feld für die Eingabe ist wieder leer. | |

Tabelle 47 Testfall - Kategorie hinzufügen

| ID / Bezeichnung | **T-001** | Kategorie ändern |
| --- | --- | --- |
| Beschreibung | Der Tester ändert eine bestehende Kategorie | |
| Testvoraussetzung | Der Benutzer muss Admin-Berechtigung haben | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Verwaltung Kategorien“ wählen 3. Bearbeiten drücken 4. Wert ändern 5. Speichern drücken | |
| Erwartetes Ergebnis | Die entsprechende Zeile geht zuerst in den Edit-Modus und nach dem Speichern wieder in den Lese-Modus. Nach dem Speichern sind die aktualisierten Daten ersichtlich. | |

Tabelle 48 Testfall - Kategorie ändern

| ID / Bezeichnung | **T-001** | Kategorie löschen |
| --- | --- | --- |
| Beschreibung | Der Tester löscht eine bestehende Kategorie | |
| Testvoraussetzung | Der Benutzer muss Admin-Berechtigung haben | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Verwaltung Kategorien“ wählen 3. Löschen-Button der entsprechenden Kategorie drücken 4. Rückfrage zum Löschen bestätigen | |
| Erwartetes Ergebnis | Nach dem Bestätigen der Rückfrage ist die Kategorie gelöscht und in der Liste nicht mehr ersichtlich. Beim Abbrechen der Rückfrage ist die Kategorie nicht gelöscht und die Liste unverändert. | |

Tabelle 49 Testfall - Kategorie löschen

| ID / Bezeichnung | **T-001** | Zustand hinzufügen |
| --- | --- | --- |
| Beschreibung | Der Tester erstellt einen neuen Zustand | |
| Testvoraussetzung | Der Benutzer muss Admin-Berechtigung haben | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Verwaltung Zustand“ wählen 3. Bezeichnung des neuen Zustands eingeben 4. Hinzufügen drücken | |
| Erwartetes Ergebnis | Der neue Zustand ist in der Liste ersichtlich. Das Feld für die Eingabe ist wieder leer. | |

Tabelle 50 Testfall - Zustand hinzufügen

| ID / Bezeichnung | **T-001** | Zustand ändern |
| --- | --- | --- |
| Beschreibung | Der Tester ändert einen bestehenden Zustand | |
| Testvoraussetzung | Der Benutzer muss Admin-Berechtigung haben | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Verwaltung Zustand“ wählen 3. Bearbeiten drücken 4. Wert ändern 5. Speichern drücken | |
| Erwartetes Ergebnis | Die entsprechende Zeile geht zuerst in den Edit-Modus und nach dem Speichern wieder in den Lese-Modus. Nach dem Speichern sind die aktualisierten Daten ersichtlich. | |

Tabelle 51 Testfall - Zustand ändern

| ID / Bezeichnung | **T-001** | Zustand löschen |
| --- | --- | --- |
| Beschreibung | Der Tester löscht einen bestehenden Zustand | |
| Testvoraussetzung | Der Benutzer muss Admin-Berechtigung haben | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Verwaltung Zustand“ wählen 3. Löschen-Button des entsprechenden Zustands drücken 4. Rückfrage zum Löschen bestätigen | |
| Erwartetes Ergebnis | Nach dem Bestätigen der Rückfrage ist der Zustand gelöscht und in der Liste nicht mehr ersichtlich. Beim Abbrechen der Rückfrage ist der Zustand nicht gelöscht und die Liste unverändert. | |

Tabelle 52 Testfall - Zustand löschen

| ID / Bezeichnung | **T-001** | Abmelden |
| --- | --- | --- |
| Beschreibung | Der Tester meldet sich ab | |
| Testvoraussetzung | Der Benutzer muss angemeldet sein | |
| Testschritte | 1. Anmelden (falls noch nicht erfolgt) 2. Menü “Abmelden“ wählen | |
| Erwartetes Ergebnis | Der Menüpunkt Abmelden ist nicht mehr ersichtlich, dafür erscheinen die Menüpunkte Registrieren und Einloggen. | |

Tabelle 53 Testfall - Abmelden

# 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

# 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.

Auf diese Schwerpunkt-Thema gehe ich nicht näher ein, da ich mich bei der Implementation bezüglich JSF an das Buch von Alexander Salvanos [SAL] gehalten habe.

6.2 Spring Security Framework

In diesem Projekt habe ich mir das Ziel gesetzt, meine Webanwendung mit Spring Security abzusichern. Da Security ein sehr grosses Thema ist, habe ich mich in diesem Projekt auf das Thema sicheres Login/Logout (Zugriffsberechtigung) und RedirectStrategy beschränkt.

Mit dem Open-Source-basierten Security Framework existiert eine interessante Alternative zur Container-Managed-Security von Java EE. Spring Security ist ein grossartiges Framwork für die Absicherung von Webanwendungen. Es ist einfach zu konfigurieren und hält die Klassen frei von Security-Codes (ausser man möchte ausdrücklich mit dem Sicherheitskontext interagieren).

Dieser Abschnitt beschreibt, wie ich meine JEE-Anwendung mit Spring Security abgesichert habe.

**Basiseinstellung**

Im ersten Schritt habe ich die Abhängigkeiten und grundlegenden Konfigurationen hinzugefügt, um die Spring Security in die Webanwendung zu integrieren. Da meine Applikation ein Maven-Projekt ist, habe ich die Spring Security Libraries in die *pom.xml* Datei eingetragen.

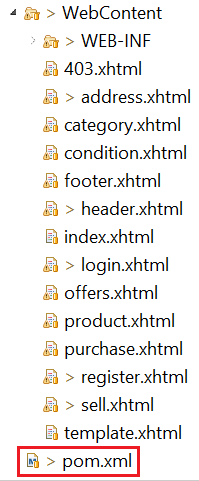




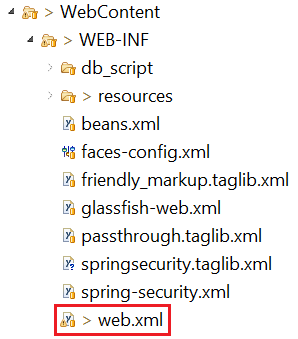
Abbildung 24 Spring Security Abhängigkeiten in pom.xml

Mein Projekt verwendet die Spring-Security-Version 4.2.3.RELEASE.

Die letzte Abhängigkeit spring-faces ist nicht zwingend notwendig. Wenn man die Security-Tags von Spring verwenden möchte, ist die spring-faces-Library aber unabdingbar. Ich gehe später in diesem Abschnitt noch näher auf die Security-Tags ein.

Um eine Web-Applikation mit Spring-Security abzusichern, werden diverse Servlet-Filter benötigt. Damit die Anzahl der Filter-Definitionen in der *web.xml* nicht überhandnimmt, habe ich lediglich ein DelegatingFilterProxy eingetragen, der wiederrum auf die anderen Filter delegiert. Dieser Servlet-Filter nimmt die Aufrufe vom Webcontainer entgegen und leitet sie Spring Security weiter. Wichtig ist dabei der Name springSecurityFilterChain des Servlet-Filters. Dieser sorgt dafür, dass durch den Filter tatsächlich die Spring Security Infrastruktur angesprochen wird.

Wie bei Servlet-Filtern üblich, muss ein URL-Pattern definiert werden, welches für die gesamte Web-Applikation gelten soll. Dazu habe ich die die *web.xml* Konfiguration folgendermassen angepasst:



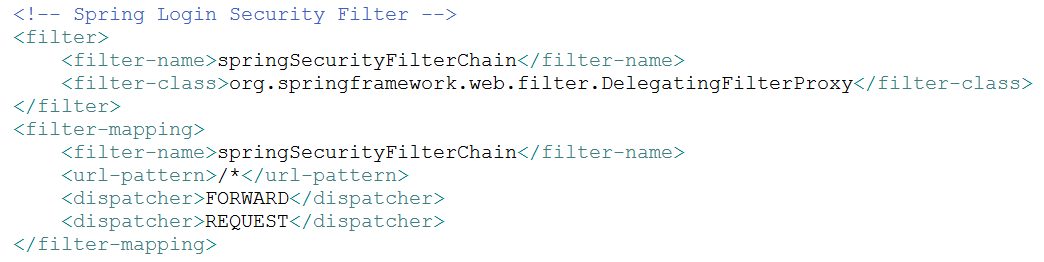


Abbildung 25 Spring Security Filter Konfiguration

*DelegatingFilterProxy* verwaltet standardmässig folgende vom Spring Security Framework zur Verfügung gestellten Servlet-Filter:

1. SecurityContextPersistenceFilter
2. LogoutFilter
3. UsernamePasswordAuthenticationFilter
4. RequestCacheAwareFilter
5. SecurityContextHolderAwareRequestFilter
6. AnonymousAuthenticationFilter
7. SessionManagementFilter
8. ExceptionTranslationFilter
9. FilterSecurityInterceptor

Bei einem Request werden die Filter der Reihe nach durchlaufen und nach dessen Vollendung in umgekehrter Reihenfolge erneut passiert.

Die Datei *web.xml* muss zusätzlich um den ContextLoaderListener erweitert werden, welcher für das Hoch- und Runterfahren von Springs-WebApplicationContext zuständig ist:

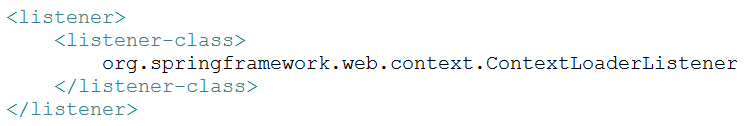


Abbildung 26 Spring Security - ContextLoaderListener

Die grundlegende Spring Security-Konfiguration habe ich in eine seperaten xml-Datei ausgelagert (*spring-security.xml*).

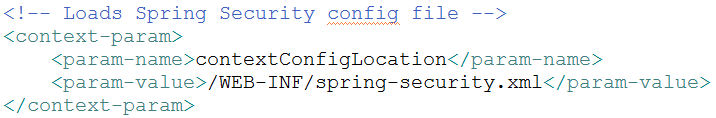


Abbildung 27 Spring Security - Verweis auf die Konfigurations-Datei

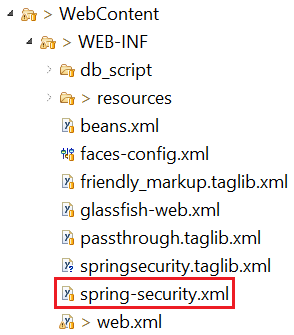




Abbildung 28 Spring Security Login/Logout-Konfiguration (spring-security.xml)

Der Tag form-login wird verwendet, wenn man wie in meinem Projekt eine eigene Login-Seite verwendet. Diese Parameter-Werte müssen dann im Login-Formular verwendet werden. Es wird hier auch eine Seite definiert, die im Fehlerfall aufgerufen wird. In meinem Fall wird wieder die Anmelde-Seite angezeigt. Zusätzlich wird durch den Parameter error=true eine Fehlermeldung zum vorliegenden Fehlerfall ausgegeben.

Mit dem Tag logout wird das Abmelden konfiguriert. Hier wird der Link für die Abmeldung definiert. Wenn dieser Link aufgerufen wird, wird der authentifizierte Benutzer vom System abgemeldet und auf die Anmeldeseite weitergeleitet.

**Sicherheitskonfiguration**

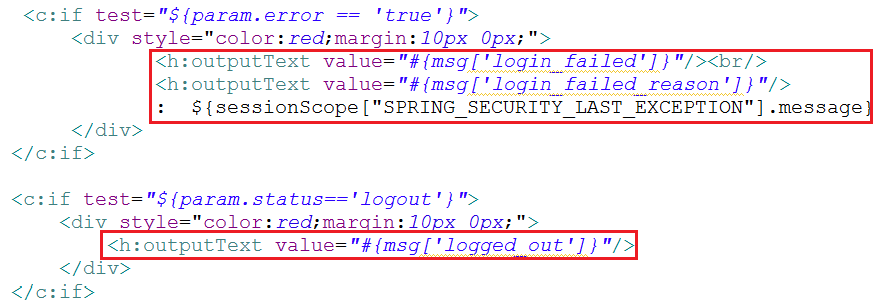
**Wenn keine eigene Login-Seite definiert wurde, wird von Spring Security eine Default-Login-Seite angezeigt. In meinem Projekt** habe ich jedoch eine eigene Login-Seite erstellt. Damit die Login-Seite wie gewünscht funktioniert, ist einiges zu beachten. Daher gehe ich hier etwas genauer auf die verschiednen Bestandteile der Login-Seite ein.



Abbildung 29 login.xhtml

Folgender Codeabschnitt dient dazu Meldungen anzuzeigen. Wenn die Anmeldung nicht erfolgreich durchgeführt werden konnte, wird die von Spring generierte Meldung angezeigt: ${sessionScope["SPRING\_SECURITY\_LAST\_EXCEPTION"].message}

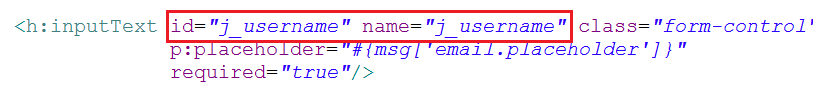
In der Konfiguration habe ich definiert, dass der Benutzer nach einer erfolgreichen Abmeldung direkt auf die Login-Seite umgeleitet wird. Der Wert logout des Parameters param.status aus der Konfiguration wird verwendet, um dem Benutzer bei einer erfolgreichen Abmeldung die Bestätigung mitzuteilen.

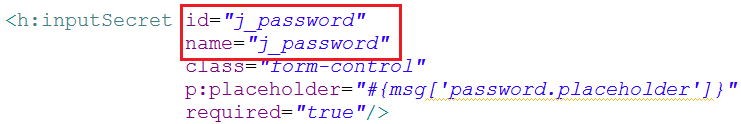


Es ist zudem sehr wichtig, dass für die Form der standard HTML-Tag verwendet wird (also nicht den Form-Tag von JSF). Die Action muss den Wert erhalten, wie er in der Konfiguration *spring-security.xml* definiert wurde. In diesem Projekt habe ich für die *login-processing-url* den Standad-Wert *j\_spring\_security\_check* verwendet:



Ebenfalls müssen die id- und name-Attribute der Input-Felder für Benutzername und Passwort den Werte aus der Konfiguration entsprechen.





Die Authentifizierung sorgt im Computer- und Internetumfeld dafür, dass die Identität eines Benutzers gegenüber einem System nachgewiesen und verifiziert werden kann.

Die Spring-Sicherheit unterstützt LDAP-, Datenbank-, XML- und Properties-Dateien, um Benutzer zu authentifizieren. In meinem Projekt verwende ich die Datenbankauthentifizierung. Um Benutzer und ihre Berechtigung zu persistieren, habe ich die vom Buch definierte Klasse Customer erweitert. Als Benutzername habe ich das bereits vorhande Feld email verwendet. Zusätzlich habe ich noch das Feld role eingeführt. Hier wird die Rolle das Benutzers gespeichert. Zur Zeit stehen die Rollen ‘ROLE\_USER‘ und ‘ROLE\_ADMIN‘ zur Verfügung. Man kann für die Rollen auch eine eigene Tabelle verwenden, das habe ich hier aber nicht als notwendig empfunden.

Um JDBC Spring-Security hinzuzufügen, müssen einige Änderungen in der *spring-security.xml* vorgenommen werden. Um die Passwortsicherheit zu erhöhen, habe ich die md5-Passwortkodierung gewählt. Mit authentication-manager wird definiert, wie der Benutzer authentifiziert werden soll.

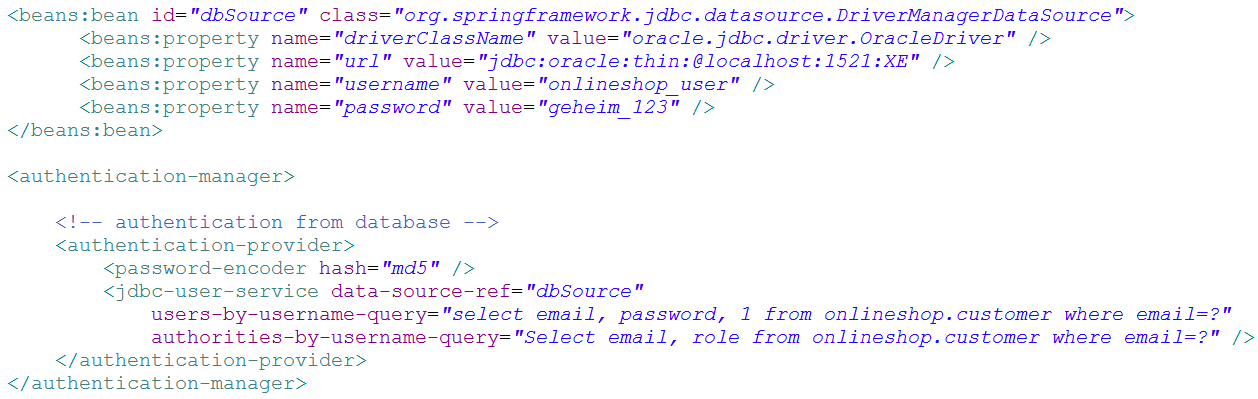
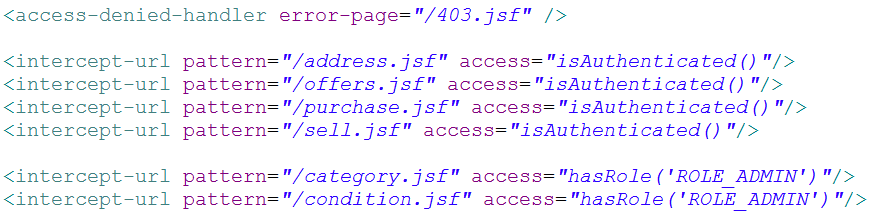


Abbildung 30 JDBC Spring-Security (spring-security.xml)

**Autorisierung**

Autorisierung dient der Festlegung von Zugriffsrechten auf Ressourcen, die im Zusammenhang mit Informationssicherheit und Computersicherheit im Allgemeinen und Zugriffskontrolle im Besonderen steht. Formal bedeutet "autorisieren", Zugriffsrichtlinien zu definieren.

Für die Absicherung von Resourcen müssen wiederum Anpassungen im File *spring-security.xml* vorgenommen werden.



*Abbildung 31 Spring Security* ***Autorisierung****-Konfiguration (spring-security.xml)*

Mit dem Tag access-denied-handler gibt man eine Seite an, auf welche bei einem unberechtigten Zugriff weitergeleitet wird. Dieser Tag wird nur benötigt, wenn man eine eigene Fehler-Seite anstelle der Standard-Page für den Error 403 angezeigt haben möchte.

Das Pattern im Tag intercept-url dient dazu, bestimmte Resourcen vor unberechtigtem Zugriff zu schützen. In meiner Applikation wird der Zugriff auf die Seiten address.jsf, offers.jsf, purchase.jsf und sell.jsf nur angemeldeten Benutzern gewährt. Dabei hat die Benutzer-Rolle keinen Einfluss auf den Zugriff. Für die Seiten category.jsf und condition.jsf reicht eine Anmeldung beim System jedoch nicht aus, der Benutzer braucht zusätzlich Administrations-Rechte.

Die gesamte Konfiguration sieht dann folgendermassen aus:



Abbildung 32 Gesamete Spring-Security-Konfiguration (spring-security.xml)

Spring Security verfügt über eine eigene Taglib, die grundlegende Unterstützung für den Zugriff auf Sicherheitsinformationen und die Anwendung von Sicherheitsbeschränkungen in JSPs bietet.

Um die Spring-Sicherheits-Taglib in JSP-Seiten zu verwenden, muss man diese Taglib-Deklaration zum Projekt hinzufügen:

<%@ taglib prefix="sec" uri="http://www.springframework.org/security/tags" %>

Die Verwendung der Spring Security Taglib in Facelets-Seiten ist ein bisschen komplizierter.

Zuerst muss man diese XML-Namespace-Deklaration für die Spring-Security-Taglib in die Facelet-Seite einfügen:

xmlns:sec=<http://www.springframework.org/security/tags>

Als nächstes habe ich eine xml-Datei mit dem Namen *springsecurity.taglib.xml* für die Taglibs im Ordner WEB-INF erstellt und eine Param-Deklaration in die *web.xml*, welche auf diese Datei verweist, hinzugefügt. Wie man sieht, verwende ich noch andere Taglibs, welche aber keinen Zusammenhang mit Spring Security haben.

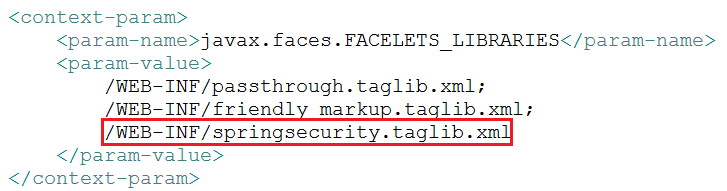


Abbildung 33 Verweis auf die Spring Security Taglib-Datei

Dies ist der Quellcode von *springsecurity.taglib.xml* wie ich sie in meinem Projekt verwende:



Abbildung 34 Spring Security Taglib springsecurity.taglib.xml

Die Taglib finde ich sehr praktisch, um z.B. Sichtbarkeiten in der JSF-Seite für Benutzer mit bestimmten Rollen einzuschränken. Im Hauptmenü verwende ich die Tags um die Menü Registrieren, Einloggen und Abmelden zu steuern. Um dieses Verhalten zu implementieren, ist der entsprechende Quellcode in das Spring-Security-Authorize-Tag geschachtelt.

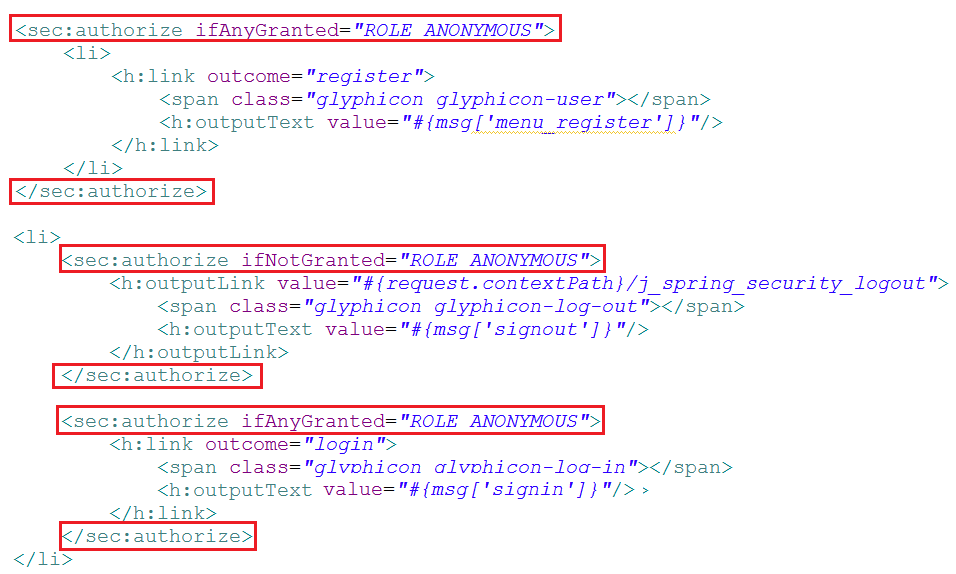


Abbildung 35 Verwendung der Spring Security Taglib in header.xhtml

Genau so einfach hätte ich die Sichtbarkeit des Menüs Admin Dashboard einschränken können, so dass nur für Benutzer mit Admin-Rolle das Menü sichtbar ist. Ich habe das aber bewusst nicht gemacht, da ich aufzeigen möchte, was passiert, wenn ein angemeldeter Benutzer ohne Admin-Rolle auf die Seite zugreift. In diesem Moment greift nämlich meine Einstellung für Access-Denied und der Benutzer wird auf die Seite 403.xhtml weitergeleitet.

Die Spring Security ermöglicht auch die Ausführung der Berechtigungsverwaltung für die Methodenausführung. Es stellt Annotationen bereit, die Ausdrucksattribute enthalten können, die vor und nach dem Methodenaufruf angewendet werden. Da ich diese nicht verwendet habe, gehe ich hier nicht weiter darauf ein.

6.3 AJAX Technologie

Xxxx

# Test

5.1 Testprotokoll

| ID / Bezeichnung | T-001 | Produktdetail anschauen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 12.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 54 Testergebnis T-001

# 6 Evaluierungsbericht

8.1 Zusammenfassung

8.2 Schwierigkeiten

8.3 Fazit

# Literaturverzeichnis

[OB] Bernd Oestereich und Alex Scheithauer. Analyse und Design mit der UML 2.5: Objektorientierte Softwareentwicklung. Oldenbourg Wissenschaftsverlag, 2013.

[Sal] Alexander Salvanos. Professionell entwickeln mit Java EE 7 - Das umfassende Handbuch. Rheinwerk Verlag GmbH, Bonn, 2016.