JEE SEMESTERARBEIT 2017

Onlineshop mit JSF, Spring Security und AJAX

Abstract

Das vorliegende Dokument beschreibt die Semesterarbeit für das Modul   
Java Enterprise Edition (JEE).  
Die Schwerpunkte liegen dabei in der Implementation in JSF mit Spring Security und AJAX.

birgit.sturzenegger@gmx.ch

Inhaltsverzeichnis

[1 Ziel 6](#_Toc500890681)

[1.1 Fokus 6](#_Toc500890682)

[1.2 Schwerpunkte 6](#_Toc500890683)

[2 Analyse 7](#_Toc500890684)

[2.1 Anwendungsfälle 8](#_Toc500890685)

[2.2 Fachklassenmodell 17](#_Toc500890686)

[3 Softwareentwurf 18](#_Toc500890687)

[3.1 Softwarearchitektur 18](#_Toc500890688)

[3.2 Klassendiagramme 20](#_Toc500890689)

[3.3 Datenmodell 23](#_Toc500890690)

[3.4 Sequenzdiagramm 24](#_Toc500890691)

[3.5 Speziﬁkation der Bedienoberﬂächen (GUI-Skizzen) 25](#_Toc500890692)

[3.6 Testkonzept 40](#_Toc500890693)

[4 Implementation 49](#_Toc500890694)

[4.1 Technische Infrastruktur 49](#_Toc500890695)

[4.2 Schwerpunkt JSF 50](#_Toc500890696)

[4.3 Schwerpunkt Spring Security 50](#_Toc500890697)

[4.4 Schwerpunkt AJAX 60](#_Toc500890698)

[5 Test 63](#_Toc500890699)

[5.1 Testprotokolle 63](#_Toc500890700)

[6 Evaluierungsbericht 68](#_Toc500890701)

[7 Literaturverzeichnis 69](#_Toc500890702)

Abbildungsverzeichnis

[Abbildung 1 UseCase-Diagramm 7](#_Toc500890703)

[Abbildung 2 Fachklassenmodell 17](#_Toc500890704)

[Abbildung 3 MVC-Muster JSF 18](file:///I:\JEE_Semesterarbeit.docx#_Toc500890705)

[Abbildung 4 Verteilungsdiagramm 19](#_Toc500890706)

[Abbildung 5 Software Architektur - logische Layer 19](#_Toc500890707)

[Abbildung 6 Package-Struktur 20](#_Toc500890708)

[Abbildung 7 Klassendiagramm package ffhs.onlineshop (Controllers) 21](#_Toc500890709)

[Abbildung 8 Klassendiagramm package ffhs.onlineshop.model (Models) 22](#_Toc500890710)

[Abbildung 9 Klassendiagramm ffhs.onlineshop.repository (Repositories) 23](#_Toc500890711)

[Abbildung 10 Diagramm Datenbank-Schema 23](#_Toc500890712)

[Abbildung 11 Sequenzdiagramm - Einkauf bewerten 24](#_Toc500890713)

[Abbildung 12 UI Produktseite 26](#_Toc500890714)

[Abbildung 13 UI Produktdetail 27](#_Toc500890715)

[Abbildung 14 UI Registrieren 28](#_Toc500890716)

[Abbildung 15 UI Einloggen 29](#_Toc500890717)

[Abbildung 16 UI Meine Adresse 30](#_Toc500890718)

[Abbildung 17 UI Meine Adresse bearbeiten 31](#_Toc500890719)

[Abbildung 18 UI Meine Einkäufe 32](#_Toc500890720)

[Abbildung 19 UI Bewertung abgeben 33](#_Toc500890721)

[Abbildung 20 UI Meine Angebote 34](#_Toc500890722)

[Abbildung 21 UI Verkaufen 35](#_Toc500890723)

[Abbildung 22 UI Kein Zugriff 36](#_Toc500890724)

[Abbildung 23 UI Verwaltung Kategorien 37](#_Toc500890725)

[Abbildung 24 UI Verwaltung Kategorien - Validerungs-Beispiel 38](#_Toc500890726)

[Abbildung 25 UI Verwaltung Zustand 39](#_Toc500890727)

[Abbildung 26 UI Abgemeldet 40](#_Toc500890728)

[Abbildung 27 Spring Security - Abhängigkeiten in pom.xml 51](#_Toc500890729)

[Abbildung 28 Spring Security - Filter Konfiguration 52](#_Toc500890730)

[Abbildung 29 Spring Security - ContextLoaderListener 52](#_Toc500890731)

[Abbildung 30 Spring Security - Verweis auf die Konfigurations-Datei 52](#_Toc500890732)

[Abbildung 31 Spring Security - Login/Logout-Konfiguration (spring-security.xml) 53](#_Toc500890733)

[Abbildung 32 Spring Security – Webseite login.xhtml 54](#_Toc500890734)

[Abbildung 33 Spring Security - JDBC (spring-security.xml) 56](#_Toc500890735)

[Abbildung 34 Spring Security - Autorisierung-Konfiguration (spring-security.xml) 56](#_Toc500890736)

[Abbildung 35 Spring Security - Gesamte Konfiguration (spring-security.xml) 57](#_Toc500890737)

[Abbildung 36 Spring Security - Verweis auf die Security-Taglib-Datei 58](#_Toc500890738)

[Abbildung 37 Spring Security – Security Taglib (springsecurity.taglib.xml) 58](#_Toc500890739)

[Abbildung 38 Spring Security - Verwendung der Security Taglib (header.xhtml) 59](#_Toc500890740)

[Abbildung 39 Klassische Webanwendung ohne AJAX 60](file:///I:\JEE_Semesterarbeit.docx#_Toc500890741)

[Abbildung 40 Webanwendung mit AJAX 61](file:///I:\JEE_Semesterarbeit.docx#_Toc500890742)

Tabellenverzeichnis

[Tabelle 1 UseCase 1 - Registrieren 8](#_Toc500890743)

[Tabelle 2 UseCase 2 - Anmelden 8](#_Toc500890744)

[Tabelle 3 UseCase 3 - Abmelden 9](#_Toc500890745)

[Tabelle 4 UseCase 4 - Produktübersicht anschauen 9](#_Toc500890746)

[Tabelle 5 UseCase 5 - Produkt-Detail anschauen 9](#_Toc500890747)

[Tabelle 6 UseCase 6 Verkäufer-Infos und Bewertungen anschauen 10](#_Toc500890748)

[Tabelle 7 UseCase 7 - Produkt suchen 10](#_Toc500890749)

[Tabelle 8 UseCase 8 - Produkt kaufen 10](#_Toc500890750)

[Tabelle 9 UseCase 9 - Registrierungsdaten ändern 11](#_Toc500890751)

[Tabelle 10 UseCase 10 - Einkaufsübersicht anschauen 11](#_Toc500890752)

[Tabelle 11 UseCase 11 - Einkauf bewerten 12](#_Toc500890753)

[Tabelle 12 UseCase 12 - Angebotsübersicht anschauen 12](#_Toc500890754)

[Tabelle 13 UseCase 13 - Verkauf bewerten 12](#_Toc500890755)

[Tabelle 14 UseCase 14 - Angebot löschen 13](#_Toc500890756)

[Tabelle 15 UseCase 15 - Angebot ändern 13](#_Toc500890757)

[Tabelle 16 UseCase 16 - Angebot erstellen 13](#_Toc500890758)

[Tabelle 17 UseCase 17 - Kategorie Übersicht anschauen 14](#_Toc500890759)

[Tabelle 18 UseCase 18 - Kategorie löschen 14](#_Toc500890760)

[Tabelle 19 UseCase 19 - Kategorie hinzufügen 15](#_Toc500890761)

[Tabelle 20 UseCase 20 - Kategorie ändern 15](#_Toc500890762)

[Tabelle 21 UseCase 21 - Zustandsübersicht anschauen 15](#_Toc500890763)

[Tabelle 22 UseCase 22 - Zustand löschen 16](#_Toc500890764)

[Tabelle 23 UseCase 23 - Zustand hinzufügen 16](#_Toc500890765)

[Tabelle 24 UseCase 24 - Zustand ändern 16](#_Toc500890766)

[Tabelle 25 Testkonzept - Teststufen 40](#_Toc500890767)

[Tabelle 26 Testkonzept - Test-Arten 41](#_Toc500890768)

[Tabelle 27 Testkonzept - Manuelle Tests 42](#_Toc500890769)

[Tabelle 28 Testfall 001 - Produkt suchen (anonymer Benutzer) 42](#_Toc500890770)

[Tabelle 29 Testfall 002 - Produkt nach Kategorie filtern (anonymer Benutzer) 42](#_Toc500890771)

[Tabelle 30 Testfall 003 - Produktdetail anschauen (anonymer Benutzer) 43](#_Toc500890772)

[Tabelle 31 Testfall 004 - Verkäufer-Infos und Bewertung anschauen(anonymer Benutzer) 43](#_Toc500890773)

[Tabelle 32 Testfall 005 - Eine geschützte Seite wählen (anonymer Benutzer) 43](#_Toc500890774)

[Tabelle 33 Testfall 006 - Registrieren 43](#_Toc500890775)

[Tabelle 34 Testfall 007 - Anmelden 44](#_Toc500890776)

[Tabelle 35 Testfall 008 - Produkt suchen (angemeldeter Benutzer) 44](#_Toc500890777)

[Tabelle 36 Testfall 009 - Produkt nach Kategorie filtern (angemeldeter Benutzer) 44](#_Toc500890778)

[Tabelle 37 Testfall 010 - Produktdetail anschauen (angemeldeter Benutzer) 45](#_Toc500890779)

[Tabelle 38 Testfall 011 - Registrierungsdaten ändern 45](#_Toc500890780)

[Tabelle 39 Testfall 012 - Angebot erstellen 45](#_Toc500890781)

[Tabelle 40 Testfall 013 - Produkt kaufen 45](#_Toc500890782)

[Tabelle 41 Testfall 014 - Einkaufsübersicht anschauen 46](#_Toc500890783)

[Tabelle 42 Testfall 015 - Einkauf bewerten 46](#_Toc500890784)

[Tabelle 43 Testfall 016 - Angebotsübersicht anschauen 46](#_Toc500890785)

[Tabelle 44 Testfall 017 - Angebot ändern 47](#_Toc500890786)

[Tabelle 45 Testfall 018 - Verkauf bewerten 47](#_Toc500890787)

[Tabelle 46 Testfall 019 - Angebot löschen 47](#_Toc500890788)

[Tabelle 47 Testfall 020 - Kategorie hinzufügen 47](#_Toc500890789)

[Tabelle 48 Testfall 021 - Kategorie ändern 48](#_Toc500890790)

[Tabelle 49 Testfall 022 - Kategorie löschen 48](#_Toc500890791)

[Tabelle 50 Testfall 023 - Zustand hinzufügen 48](#_Toc500890792)

[Tabelle 51 Testfall 024 - Zustand ändern 49](#_Toc500890793)

[Tabelle 52 Testfall 025 - Zustand löschen 49](#_Toc500890794)

[Tabelle 53 Testfall 026 - Abmelden 49](#_Toc500890795)

[Tabelle 54 Testergebnis T-001 63](#_Toc500890796)

[Tabelle 55 Testergebnis T-002 63](#_Toc500890797)

[Tabelle 56 Testergebnis T-003 63](#_Toc500890798)

[Tabelle 57 Testergebnis T-004 63](#_Toc500890799)

[Tabelle 58 Testergebnis T-005 63](#_Toc500890800)

[Tabelle 59 Testergebnis T-006 64](#_Toc500890801)

[Tabelle 60 Testergebnis T-007 64](#_Toc500890802)

[Tabelle 61 Testergebnis T-008 64](#_Toc500890803)

[Tabelle 62 Testergebnis T-009 64](#_Toc500890804)

[Tabelle 63 Testergebnis T-010 64](#_Toc500890805)

[Tabelle 64 Testergebnis T-011 64](#_Toc500890806)

[Tabelle 65 Testergebnis T-012 65](#_Toc500890807)

[Tabelle 66 Testergebnis T-013 65](#_Toc500890808)

[Tabelle 67 Testergebnis T-014 65](#_Toc500890809)

[Tabelle 68 Testergebnis T-015 65](#_Toc500890810)

[Tabelle 69 Testergebnis T-016 65](#_Toc500890811)

[Tabelle 70 Testergebnis T-017 65](#_Toc500890812)

[Tabelle 71 Testergebnis T-018 65](#_Toc500890813)

[Tabelle 72 Testergebnis T-019 66](#_Toc500890814)

[Tabelle 73 Testergebnis T-020 66](#_Toc500890815)

[Tabelle 74 Testergebnis T-021 66](#_Toc500890816)

[Tabelle 75 Testergebnis T-022 66](#_Toc500890817)

[Tabelle 76 Testergebnis T-023 66](#_Toc500890818)

[Tabelle 77 Testergebnis T-024 66](#_Toc500890819)

[Tabelle 78 Testergebnis T-025 67](#_Toc500890820)

[Tabelle 79 Testergebnis T-026 67](#_Toc500890821)

Abkürzungsverzeichnis

AJAX Asynchronous JavaScript And XML

API Application Programming Interface

CSS Cascading Style Sheet

CDI Contexts and Dependency Injection

D2D Direct-To-DOM

DOM Document Object Model

EJB Enterprise Java Beans

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

JEE Java Enterprise Edition

JSF JavaServer Faces

JSP JavaServer Pages

MVC Model-View-Controller

UI User Interface

Unified-EL Unified Expression Language

XHTML Extensible Hypertext Markup Language

XML Extensibel Markup Language

# Ziel

## Fokus

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

Aufbauend auf dem Onlineshop von A. Salvanos [Sal] habe ich die Anwendungsfälle weiterentwickelt und 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.

Der Aufbau der Dokumentation für die Semesterarbeit orientiert sich am Buch *Analyse und Design mit der UML 2.5* von *Bernd Oestereich*.

## Schwerpunkte

Für die JEE-Applikation habe ich folgende Schwerpunkte gesetzt:

* Umsetzung mit JSF (JavaServerFaces)
* Login/Logout sowie Zugangsberechtigung mit Spring Security
* AJAX-Technologie

In Kapitel Implementierung werde ich Näher auf diese Themen eingehen.

# Analyse

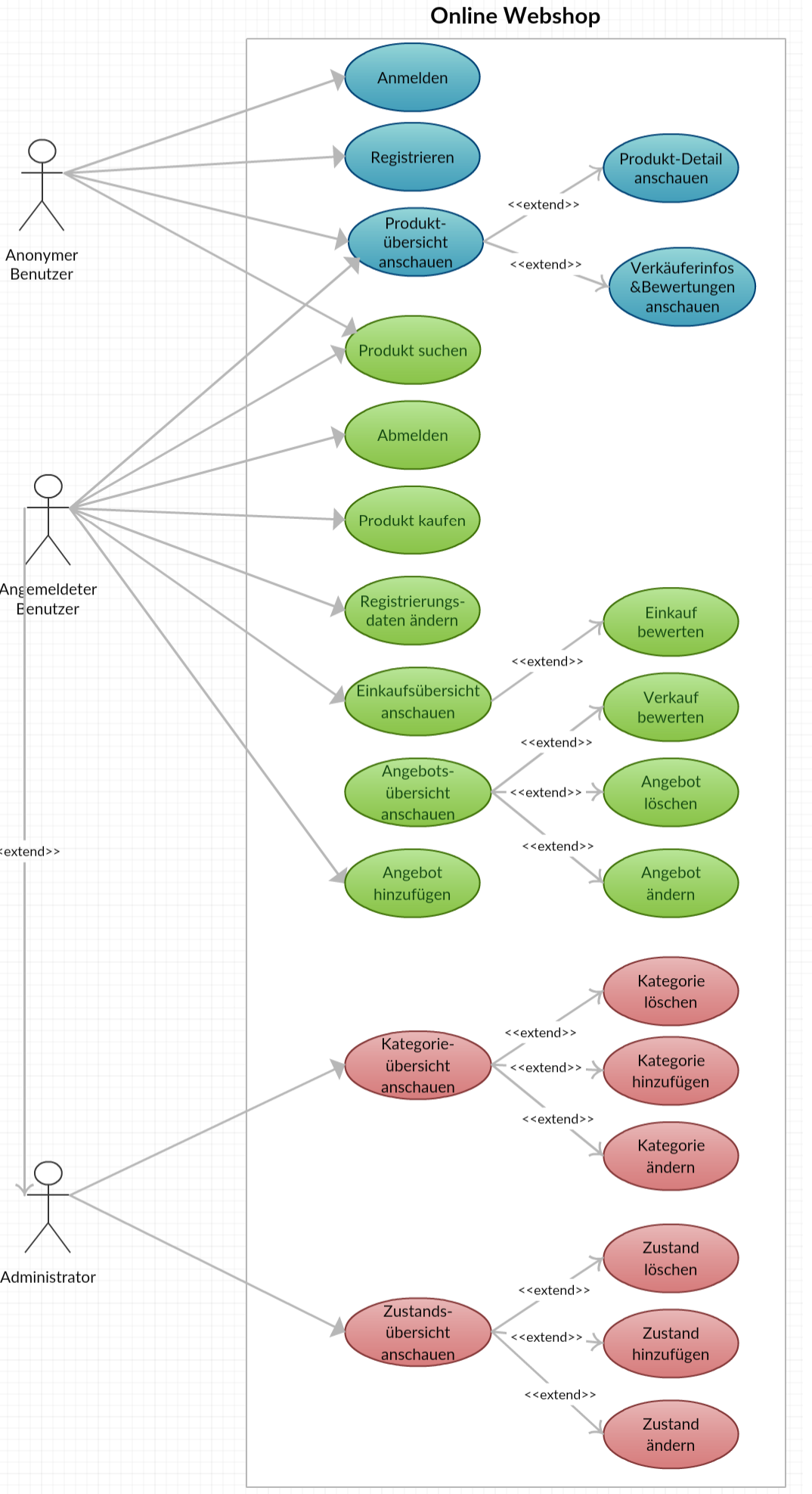


Abbildung 1 UseCase-Diagramm

## 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, E-Mail-Adresse, Passwort |
| Ablauf | 1. Ein Benutzer füllt das Registrierungsformular aus 2. Der Benutzer drückt Registrieren |
| Nachbedingungen | Die Daten des Benutzers sind gespeichert |
| Kommentar | Die E-Mail-Adresse und das Passwort wird für das Login benutzt |

Tabelle 1 UseCase 1 - Registrieren

|  |  |
| --- | --- |
| **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 Login Formulars sind obligatorisch |
| Eingehende Informationen | E-Mail-Adresse, Passwort |
| Ablauf | 1. Ein Benutzer füllt das Login Formular 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 Benutzer angezeigt |
| Nachbedingungen | Bei einer gültigen Anmeldung, ist die Login-Funktion nicht mehr ersichtlich, dafür steht jetzt die Logout-Funktion zur Verfügung. |
| Kommentar |  |

Tabelle 2 UseCase 2 - Anmelden

|  |  |
| --- | --- |
| **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 Abmelden 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 UseCase 3 - Abmelden

|  |  |
| --- | --- |
| **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 UseCase 4 - Produktübersicht anschauen

|  |  |
| --- | --- |
| **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 UseCase 5 - Produkt-Detail anschauen

|  |  |
| --- | --- |
| **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 UseCase 6 Verkäufer-Infos und Bewertungen anschauen

|  |  |
| --- | --- |
| **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 UseCase 7 - Produkt suchen

|  |  |
| --- | --- |
| **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 UseCase 8 - Produkt kaufen

|  |  |
| --- | --- |
| **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, E-Mail-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 UseCase 9 - Registrierungsdaten ändern

|  |  |
| --- | --- |
| **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 UseCase 10 - Einkaufsübersicht anschauen

|  |  |
| --- | --- |
| **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 getätigt 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 UseCase 11 - Einkauf bewerten

|  |  |
| --- | --- |
| **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 UseCase 12 - Angebotsübersicht anschauen

|  |  |
| --- | --- |
| **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 Produkt 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 UseCase 13 - Verkauf bewerten

|  |  |
| --- | --- |
| **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 UseCase 14 - Angebot löschen

|  |  |
| --- | --- |
| **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 | Entsprechende 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 UseCase 15 - Angebot ändern

|  |  |
| --- | --- |
| **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 UseCase 16 - Angebot erstellen

|  |  |
| --- | --- |
| **UseCase 17** | **Kategorie Übersicht anschauen** |
| Kurzbeschreibung | Ein angemeldeter Benutzer schaut die Kategorie Übersicht an |
| Akteure | Angemeldeter Benutzer (Administrator) |
| Auslösendes Ereignis | Ein Benutzer möchte die Kategorie Übersicht 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 2. Ein Benutzer sortiert die Tabelle ab- oder aufsteigend nach der Bezeichnung |
| Nachbedingungen | Eine Tabelle mit allen Kategorien wird angezeigt. Pro Kategorie gibt es eine Bearbeitung- sowie eine Löschen-Funktion.  Zusätzlich ist die Möglichkeit vorhanden, eine neue Kategorie hinzuzufügen. |
| Kommentar | Die Tabelle in der Kategorie Übersicht ist sortierbar |

Tabelle 17 UseCase 17 - Kategorie Übersicht anschauen

|  |  |
| --- | --- |
| **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 Kategorie Übersicht 2. Der Benutzer wählt eine Kategorie aus 3. Der Benutzer löscht die Kategorie |
| Nachbedingungen | Die Kategorie ist gelöscht |
| Kommentar |  |

Tabelle 18 UseCase 18 - Kategorie löschen

|  |  |
| --- | --- |
| **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 | Bezeichnung der Kategorie |
| Ablauf | 1. Ein Benutzer geht auf die Kategorie Übersicht 2. Der Benutzer gibt eine Bezeichnung ein 3. Der Benutzer speichert die neue Kategorie |
| Nachbedingungen | Die neue Kategorie ist gespeichert |
| Kommentar |  |

Tabelle 19 UseCase 19 - Kategorie hinzufügen

|  |  |
| --- | --- |
| **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 Kategorie Übersicht 2. Der Benutzer wählt eine Kategorie aus 3. Der Benutzer ändert die Bezeichnung der Kategorie |
| Nachbedingungen | Das Kategorie ist geändert |
| Kommentar |  |

Tabelle 20 UseCase 20 - Kategorie ändern

|  |  |
| --- | --- |
| **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 2. Ein Benutzer sortiert die Tabelle ab- oder aufsteigend nach der Bezeichnung |
| Nachbedingungen | Eine Tabelle mit allen Zuständen wird angezeigt. Pro Zustand gibt es eine Bearbeitung- sowie eine Löschen-Funktion.  Zusätzlich ist die Möglichkeit vorhanden, einen neuen Zustand hinzuzufügen. |
| Kommentar | Die Tabelle in der Zustandsübersicht ist sortierbar |

Tabelle 21 UseCase 21 - Zustandsübersicht anschauen

|  |  |
| --- | --- |
| **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 UseCase 22 - Zustand löschen

|  |  |
| --- | --- |
| **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 | Bezeichnung des Zustands |
| Ablauf | 1. Ein Benutzer geht auf die Zustandsübersicht 2. Der Benutzer gibt eine Bezeichnung ein 3. Der Benutzer speichert den neuen Zustand |
| Nachbedingungen | Der neue Zustand ist gespeichert |
| Kommentar |  |

Tabelle 23 UseCase 23 - Zustand hinzufügen

|  |  |
| --- | --- |
| **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 UseCase 24 - Zustand ändern

## 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.

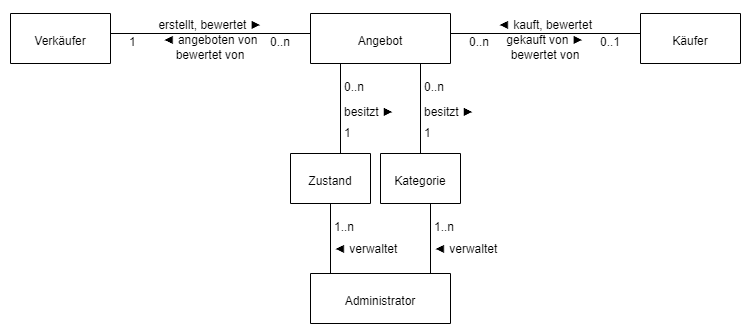


Abbildung 2 Fachklassenmodell

# Softwareentwurf

## Softwarearchitektur

Um die verschiedenen Aspekte der Anwendung zu trennen, wurde das MVC-Pattern von JSF eingesetzt. Dieses Pattern eignet sich gut, um eine flexible und lose gekoppelte Web-Anwendung zu entwickeln, da mit MVC eine strikte Trennung zwischen Anwendungslogik und Benutzerschnittstelle erreicht werden kann. Es wird also das Prinzip Separation of Concerns umgesetzt. Der Vorteil des Architekturmusters besteht darin, dass zum Beispiel einzelne Komponenten ersetzt werden können, ohne andere Komponenten auch ersetzen zu müssen. Des Weiteren wird eine Vermischung von Codeteilen der Benutzeroberfläche und der Anwendungslogik vermieden.

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 JEE (zu dem JSF gehört) werden diese Artefakte normalerweise von EJB bzw. JPA implementiert.
* **View**: Die grafische Repräsentation der darzustellenden Komponenten. Diese Komponenten werden durch JSF-Komponenten, welche Teil der Java Enterprise Edition sind, beschrieben. Die View informiert den Controller durch Ausführung von Aktionen und das Senden von Daten.
* **Controller**: Komponente für die Programmsteuerung. Er hat die Aufgabe, das Model zu aktualisieren, Business-Logik auszuführen sowie die View zu aktualisieren.

Der Controller ist im Framework eingebettet. Fälschlicherweise habe ich zu Beginn meiner JSF-Programmierung die Backing-Beans als Steuerelement bezeichnet. Das lag vermutlich daren, dass ich die Beans als serverseiter Einstiegspunkt gesehen haben. Im Laufe der Semesterarbeit habe ich festgestellt, da der eigentliche Controller das *FacesServlet* bildet. Das FacesServlet nimmt Anfragen entgegen und leitet den JSF-Lebenszyklus für die einzelnen Anfragen.

Abbildung 3 MVC-Muster JSF



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

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).

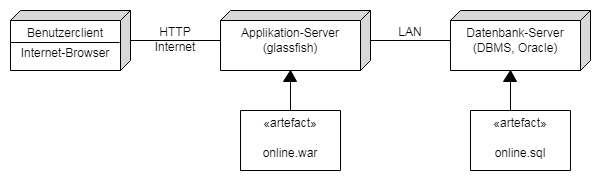


Abbildung 4 Verteilungsdiagramm

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

* Presentation layer
* Data access layer

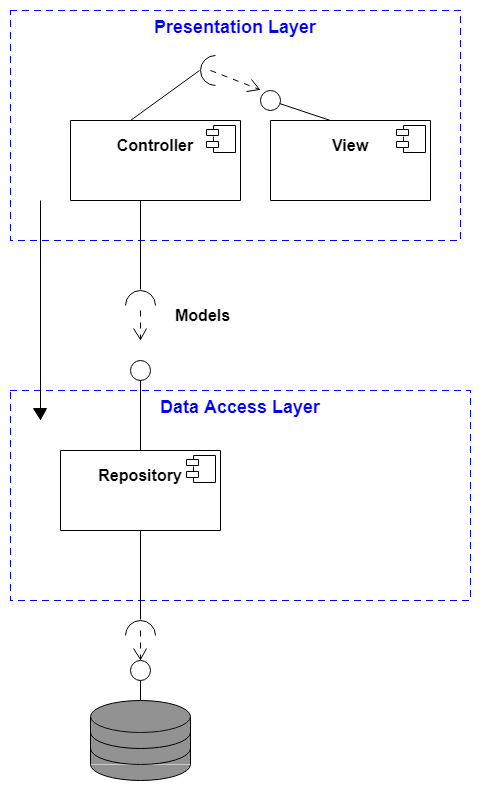


Abbildung 5 Software Architektur - logische Layer

## 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.

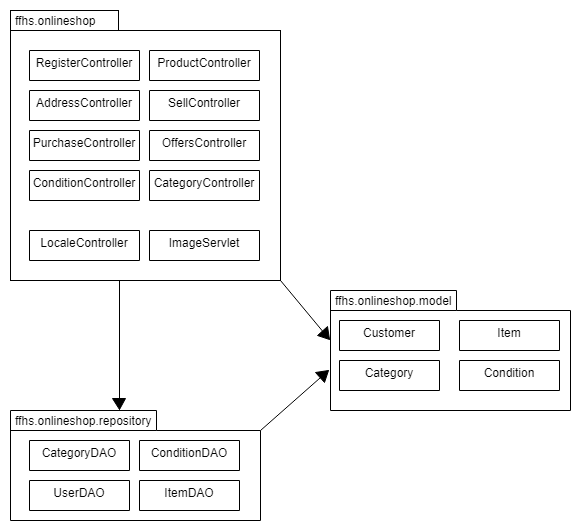


Abbildung 6 Package-Struktur

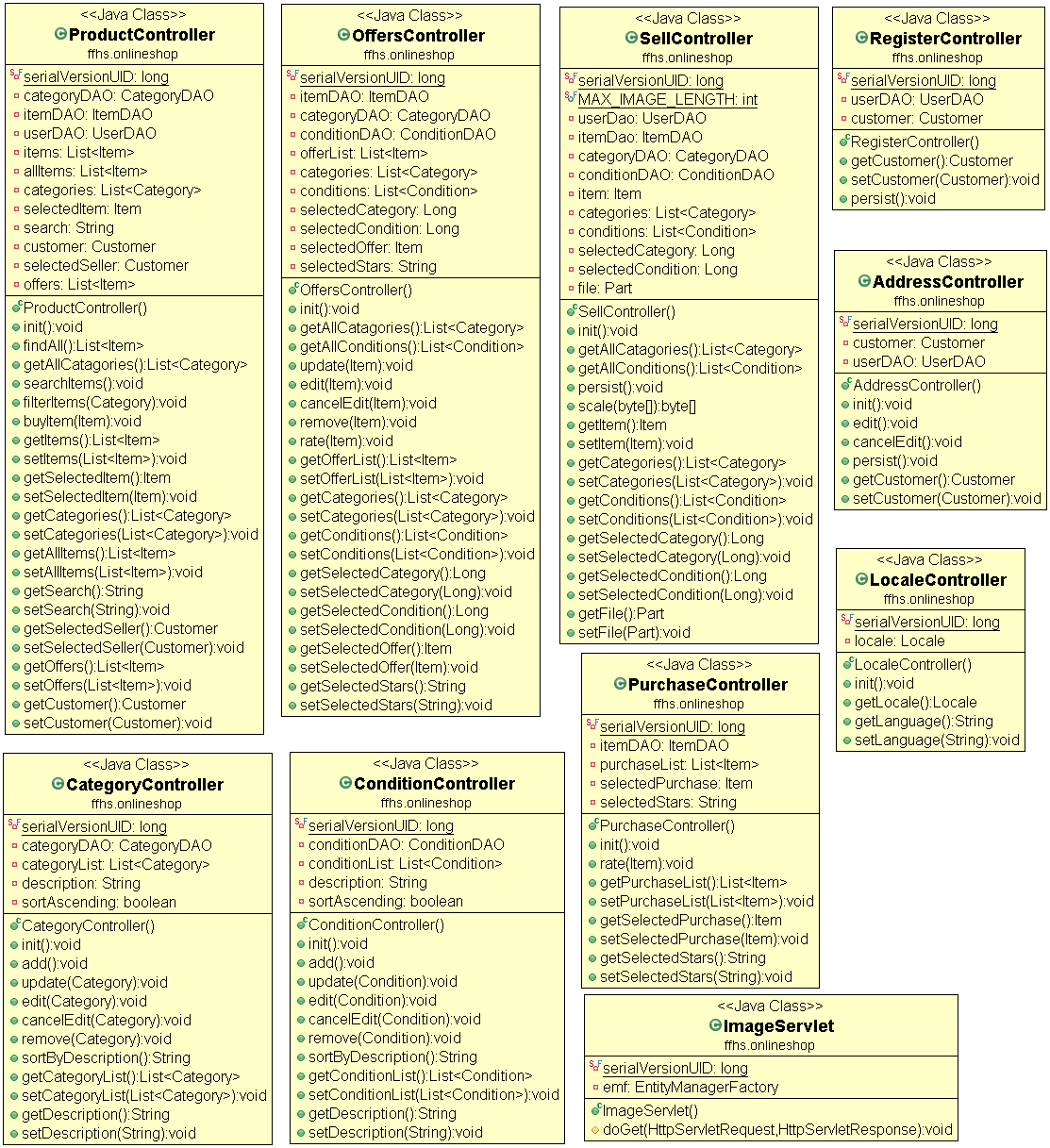


Abbildung 7 Klassendiagramm package ffhs.onlineshop (Controllers)

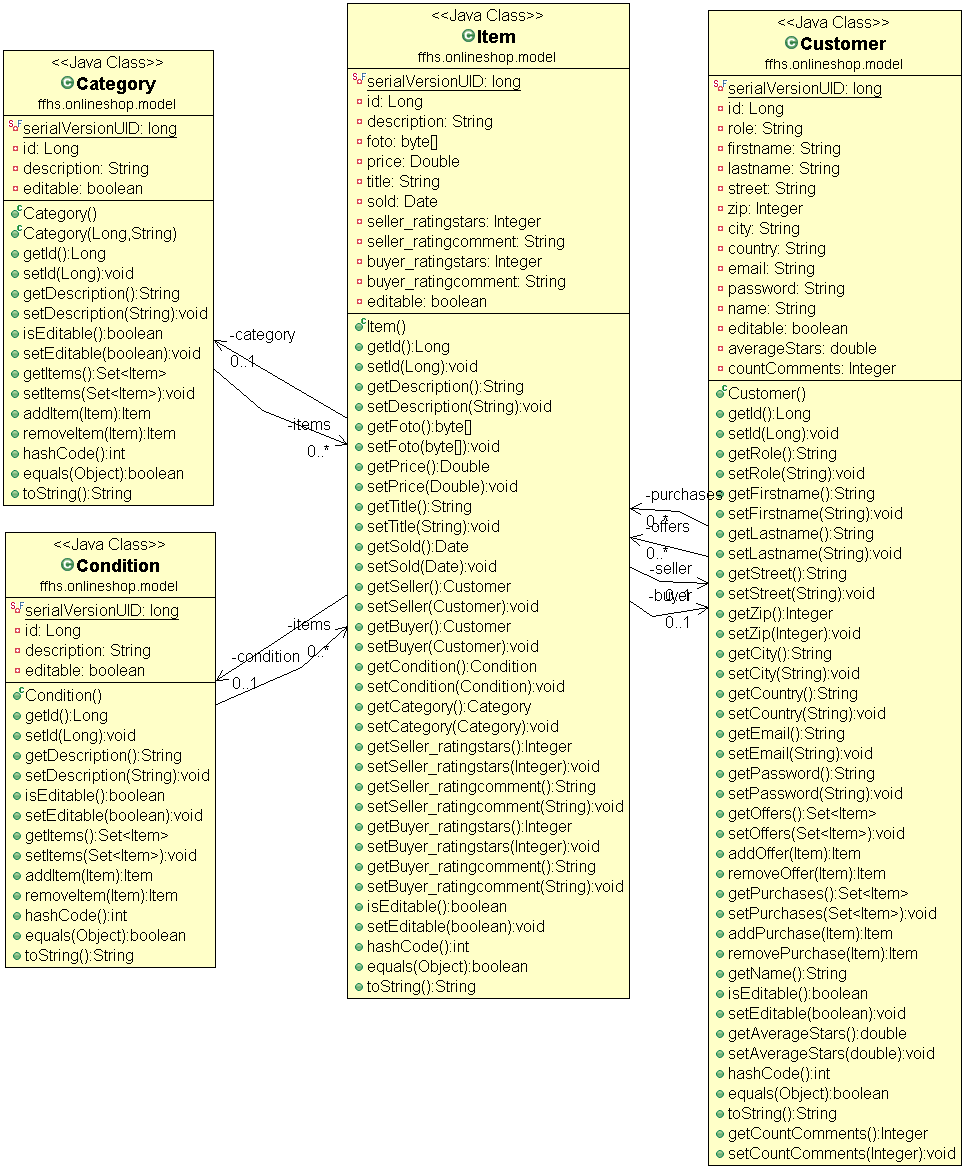


Abbildung 8 Klassendiagramm package ffhs.onlineshop.model (Models)

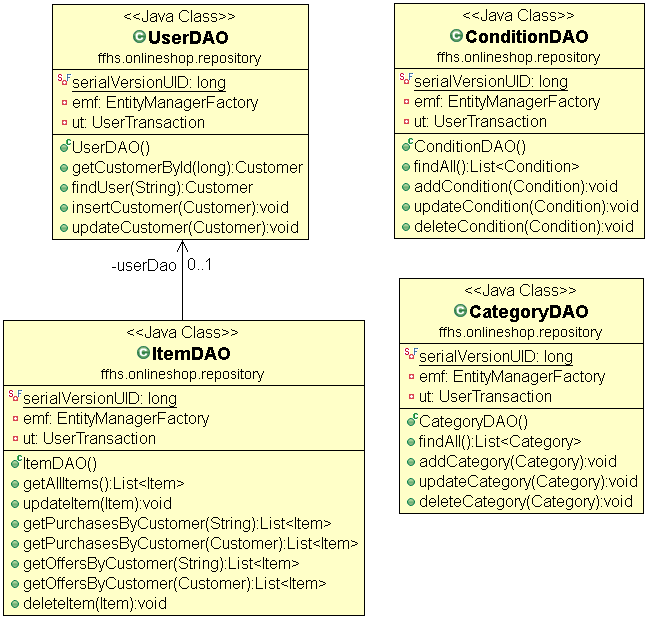


Abbildung 9 Klassendiagramm ffhs.onlineshop.repository (Repositories)

## Datenmodell

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

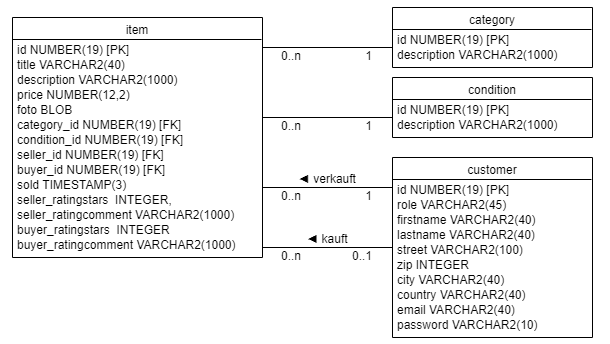


Abbildung 10 Diagramm Datenbank-Schema

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.

## Sequenzdiagramm

Hier wird der Ablauf einer Einkaufs-Bewertung aufgezeit. Voraussetzung ist, dass der Benutzer mindestens ein Produk gekauft hat. Diesen Einkauf kann er dann bewerten.

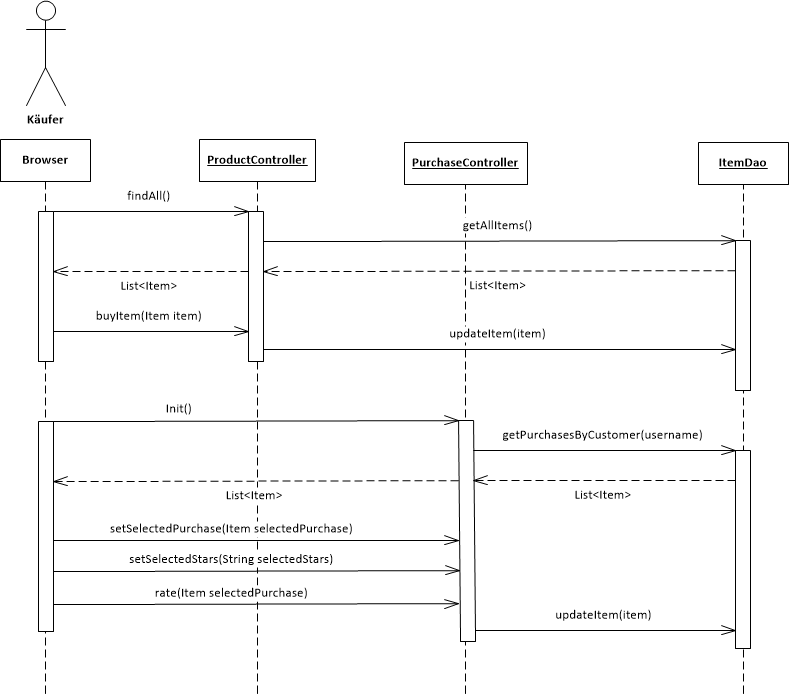


Abbildung 11 Sequenzdiagramm - Einkauf bewerten

## 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.

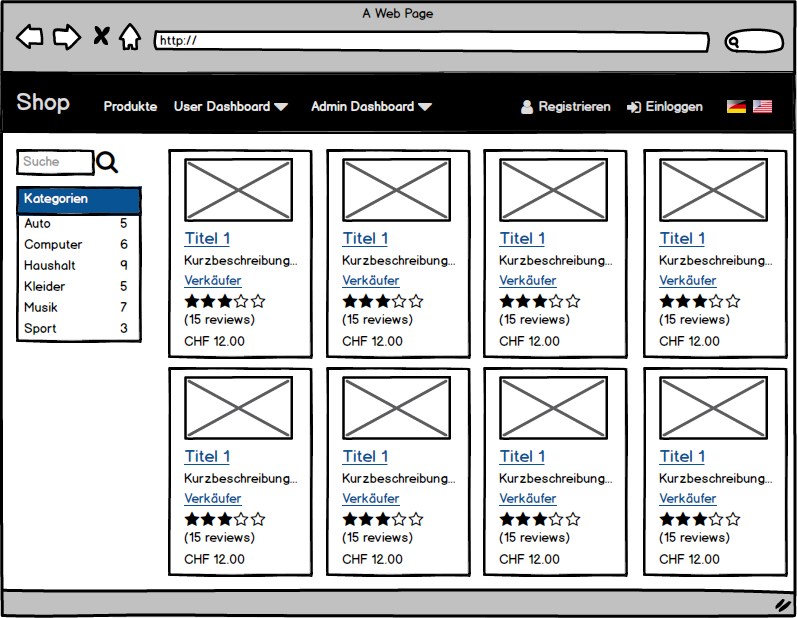


Abbildung 12 UI Produktseite

Produkt-Detail

Wenn der Benutzer auf den Titel eines Produktes klickt, erscheint eine detaillierte 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).

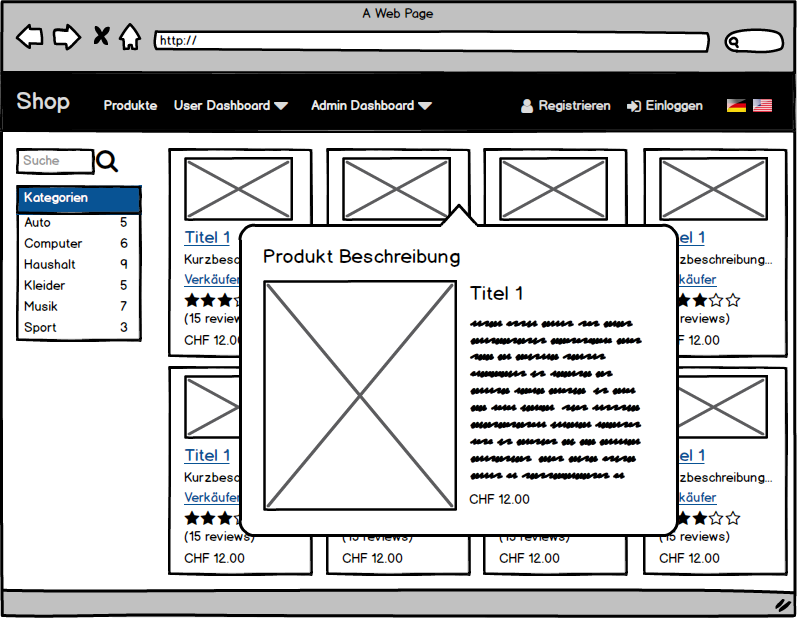


Abbildung 13 UI Produktdetail

Registrieren

Ein Benutzer kann sich jederzeit registrieren, soweit er nicht angemeldet ist. Sobald ein Benutzer sich angemeldet hat (Einloggen) ist der Menüpunkt Registrieren 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.

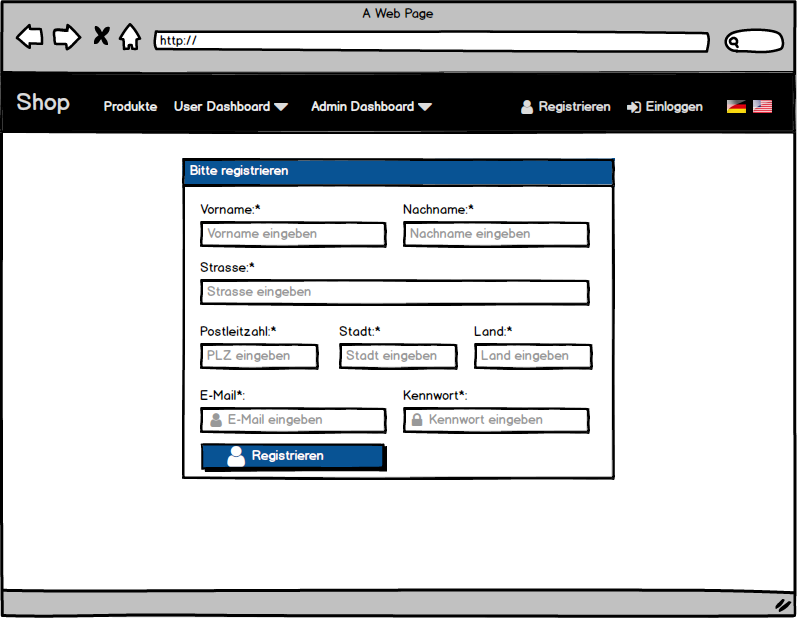


Abbildung 14 UI Registrieren

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.

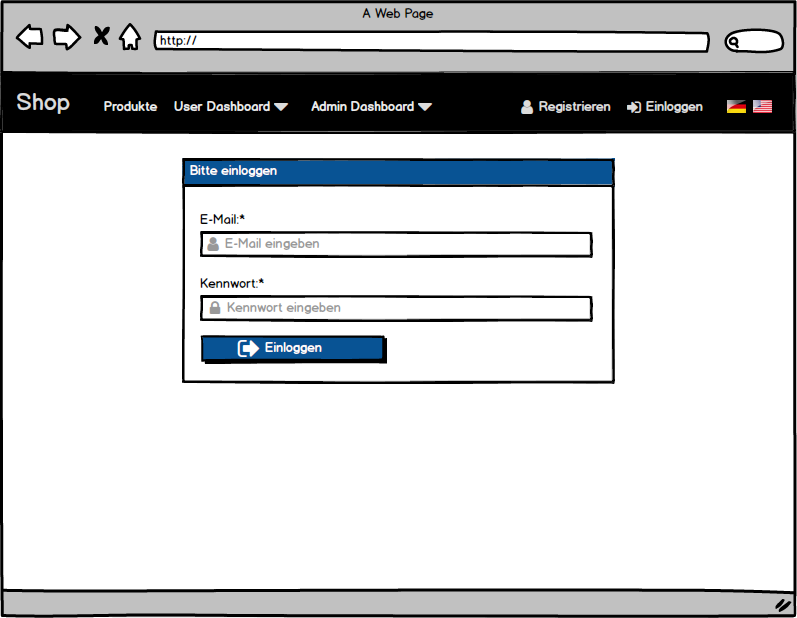


Abbildung 15 UI Einloggen

User Dashboard – Meine Adresse

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

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

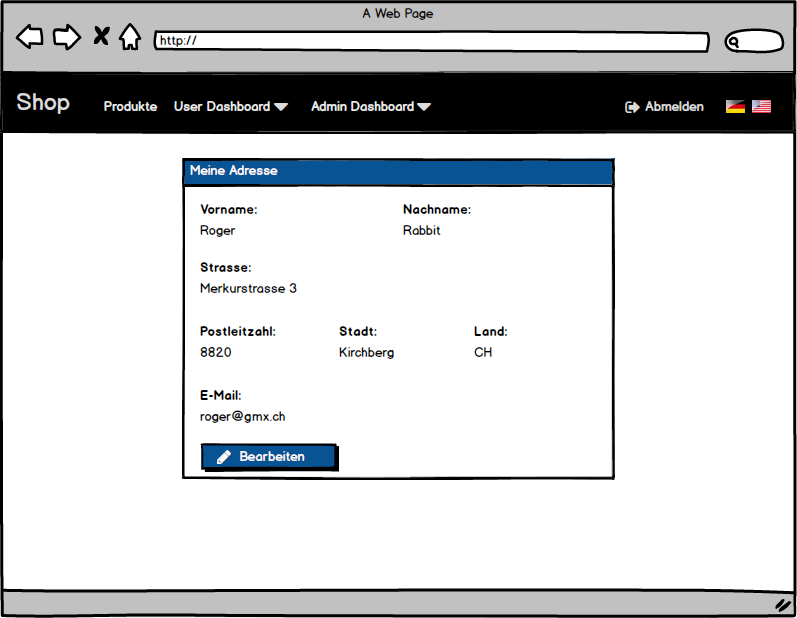


Abbildung 16 UI Meine Adresse

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.

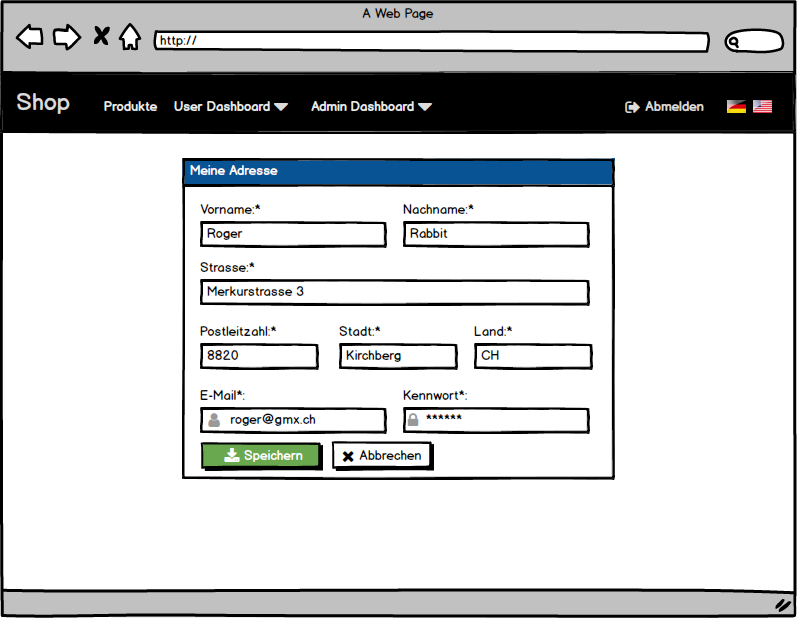


Abbildung 17 UI Meine Adresse bearbeiten

User Dashboard – Meine Einkäufe

Diese Funktion benötigt User-Benutzerrechte. Wenn *Meine Einkäufe* via Menü ausgewählt wird und noch keine Berechtigungsüberprüfung 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 durchführen. Die Bewertung kann pro Einkauf nur einmal abgegeben werden. Eine einmal abgegebene Bewertung kann nicht mehr geändert werden.

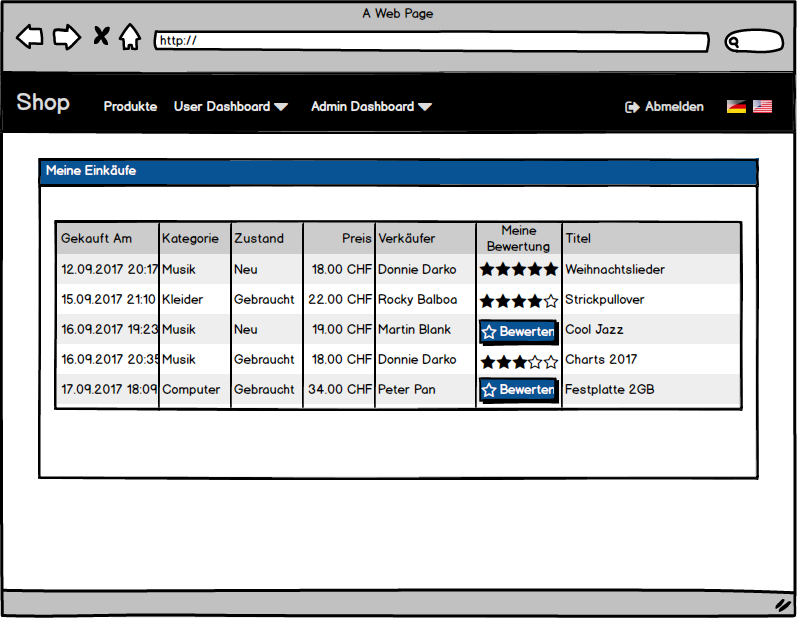


Abbildung 18 UI Meine Einkäufe

Bewertung abgeben:

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

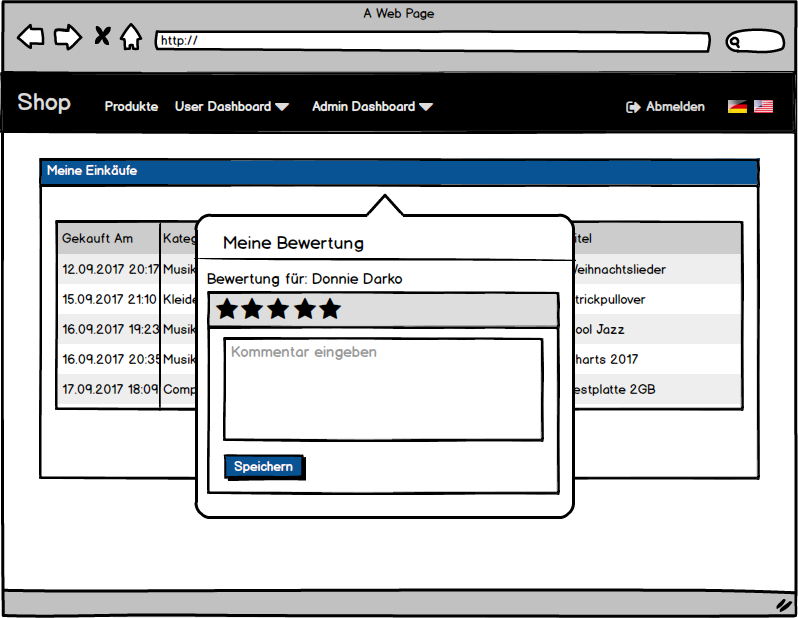


Abbildung 19 UI Bewertung abgeben

User Dashboard – Meine Angebote

Diese Funktion benötigt User-Benutzerrechte. Wenn *Meine Angebote* via Menü ausgewählt wird und noch keine Berechtigungsüberprüfung 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 durchführen. 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 validiert.

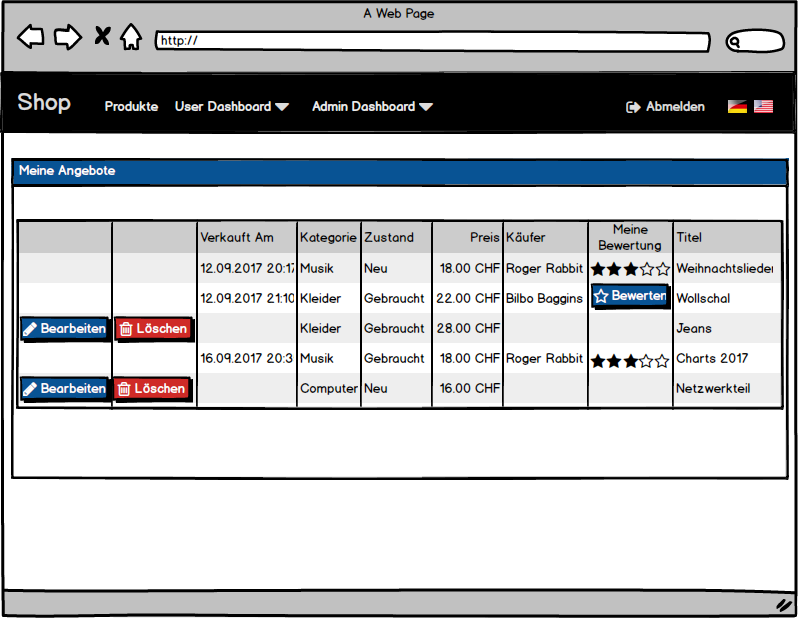


Abbildung 20 UI Meine Angebote

User Dashboard – 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.

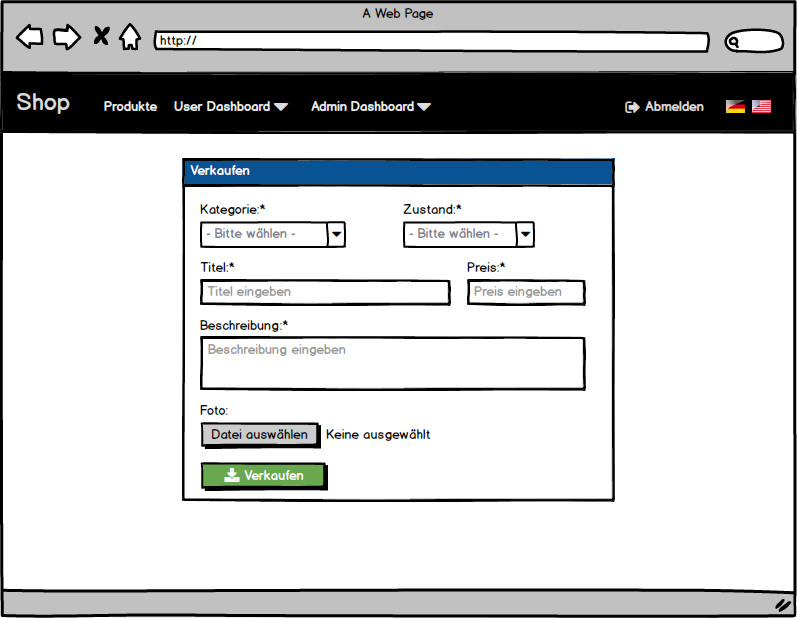


Abbildung 21 UI Verkaufen

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 zu wenig Berechtigung besitzt, wird angezeigt, dass der Benutzer kein Zugriff hat.

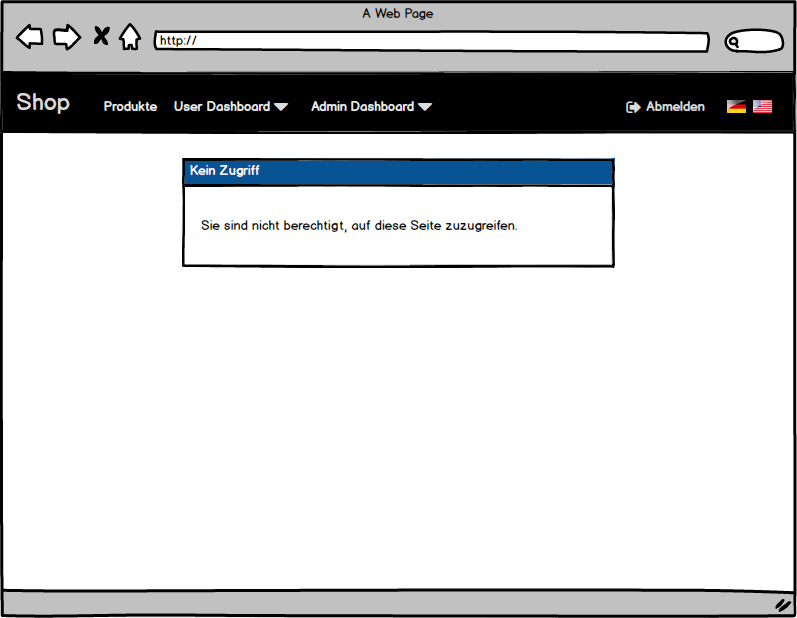


Abbildung 22 UI Kein Zugriff

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 obligatorisches Feld ist die Beschreibung. Die Beschreibung wird validiert.

Beispiel:

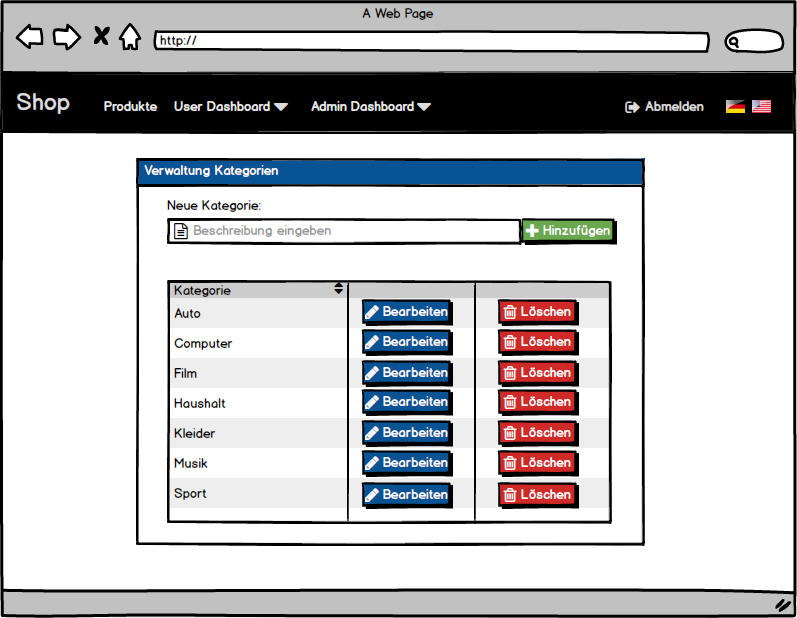


Abbildung 23 UI Verwaltung Kategorien

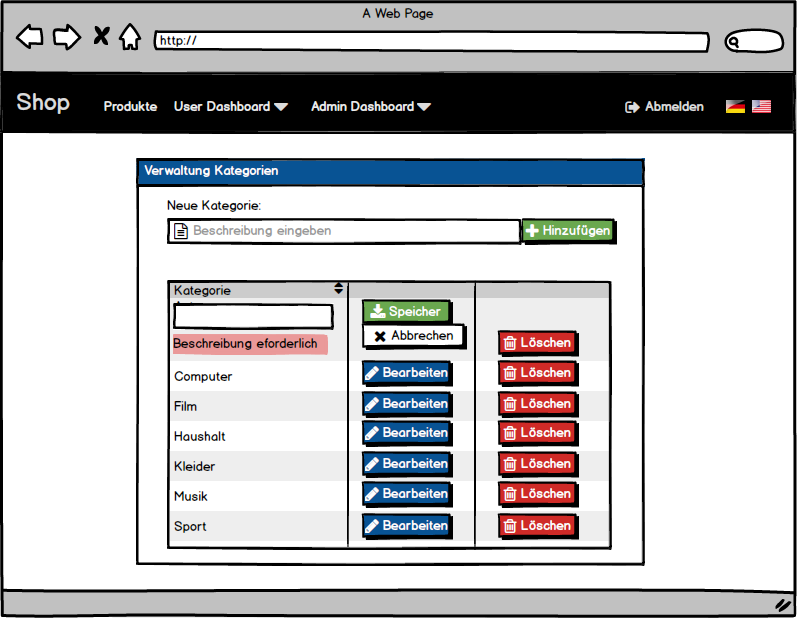


Abbildung 24 UI Verwaltung Kategorien - Validerungs-Beispiel

Admin Dashboard – Verwaltung Zustand

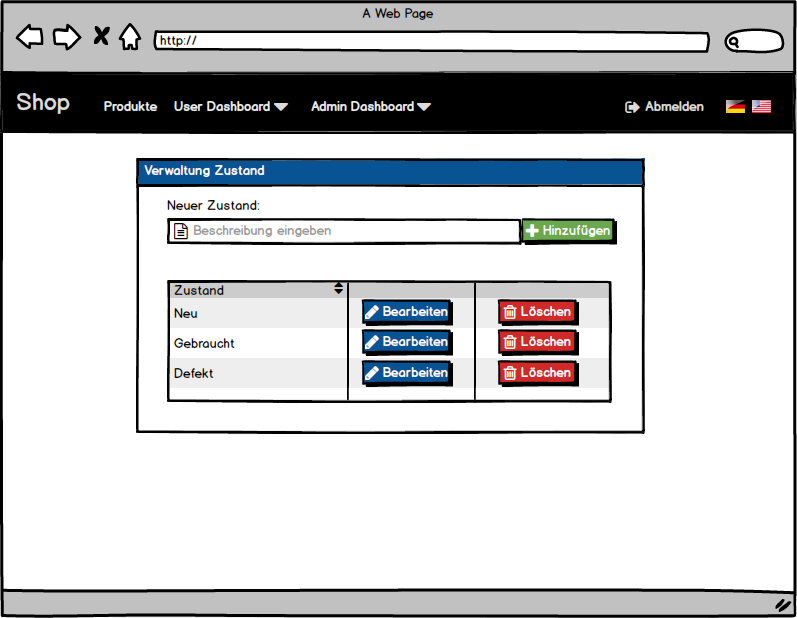


Abbildung 25 UI Verwaltung Zustand

Abmelden

Sobald sich 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.

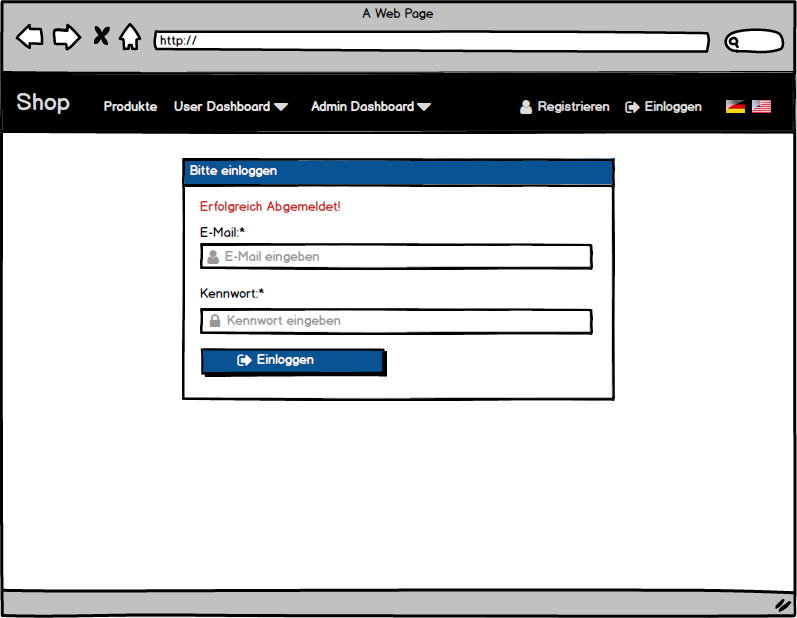


Abbildung 26 UI Abgemeldet

## 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, das 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.  Verkaufte Produkte sind oben rechts rot markiert und haben dort den Text Sold. Nicht verkaufte Produkte sind oben rechts grün markiert. | |

Tabelle 28 Testfall 001 - Produkt suchen (anonymer Benutzer)

| ID / Bezeichnung | **T-002** | 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. Verkaufte Produkte sind oben rechts rot markiert und haben dort den Text Sold. Nicht verkaufte Produkte sind oben rechts grün markiert. | |

Tabelle 29 Testfall 002 - Produkt nach Kategorie filtern (anonymer Benutzer)

| ID / Bezeichnung | **T-003** | 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 003 - Produktdetail anschauen (anonymer Benutzer)

| ID / Bezeichnung | **T-004** | 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 004 - Verkäufer-Infos und Bewertung anschauen(anonymer Benutzer)

| ID / Bezeichnung | **T-005** | 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 005 - Eine geschützte Seite wählen (anonymer Benutzer)

| ID / Bezeichnung | **T-006** | Registrieren |
| --- | --- | --- |
| 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 006 - Registrieren

| ID / Bezeichnung | **T-007** | 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 007 - Anmelden

| ID / Bezeichnung | **T-008** | 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.  Verkaufte Produkte sind oben rechts rot markiert und haben dort den Text Sold. Nicht verkaufte Produkte sind oben rechts grün markiert. | |

Tabelle 35 Testfall 008 - Produkt suchen (angemeldeter Benutzer)

| ID / Bezeichnung | **T-009** | 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.  Verkaufte Produkte sind oben rechts rot markiert und haben dort den Text Sold. Nicht verkaufte Produkte sind oben rechts grün markiert. | |

Tabelle 36 Testfall 009 - Produkt nach Kategorie filtern (angemeldeter Benutzer)

| ID / Bezeichnung | **T-010** | 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 010 - Produktdetail anschauen (angemeldeter Benutzer)

| ID / Bezeichnung | **T-011** | 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 011 - Registrierungsdaten ändern

| ID / Bezeichnung | **T-012** | 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 012 - Angebot erstellen

| ID / Bezeichnung | **T-013** | 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 013 - Produkt kaufen

| ID / Bezeichnung | **T-014** | 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 014 - Einkaufsübersicht anschauen

| ID / Bezeichnung | **T-015** | Einkauf bewerten |
| --- | --- | --- |
| Beschreibung | Der Tester bewertet einen seiner Einkäufe | |
| Testvoraussetzung | Es ist mindestens ein Produkt vom Tester gekauft worden.  Der Benutzer hat den entsprechenden 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 015 - Einkauf bewerten

| ID / Bezeichnung | **T-016** | 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 016 - Angebotsübersicht anschauen

| ID / Bezeichnung | **T-017** | 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 017 - Angebot ändern

| ID / Bezeichnung | **T-018** | Verkauf bewerten |
| --- | --- | --- |
| Beschreibung | Der Tester bewertet einer seiner Verkäufe | |
| Testvoraussetzung | Es ist mindestens ein Angebot vom Tester verkauft worden.  Der Benutzer hat den entsprechenden 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 018 - Verkauf bewerten

| ID / Bezeichnung | **T-019** | 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 019 - Angebot löschen

| ID / Bezeichnung | **T-020** | 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 020 - Kategorie hinzufügen

| ID / Bezeichnung | **T-021** | 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 021 - Kategorie ändern

| ID / Bezeichnung | **T-022** | 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 022 - Kategorie löschen

| ID / Bezeichnung | **T-023** | 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 023 - Zustand hinzufügen

| ID / Bezeichnung | **T-024** | 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 024 - Zustand ändern

| ID / Bezeichnung | **T-025** | 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 025 - Zustand löschen

| ID / Bezeichnung | **T-026** | 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 026 - Abmelden

# 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 von [SAL] beschrieben
* Oracle wie von [SAL] beschrieben
* Maven Projekt (POM)

Zusätzliche eingesetzte Frameworks

* Spring Security
* Bootstrap

## Schwerpunkt 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 JavaServerFaces Version 2.2 verwendet.

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

## Schwerpunkt Spring Security

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 Framework 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.

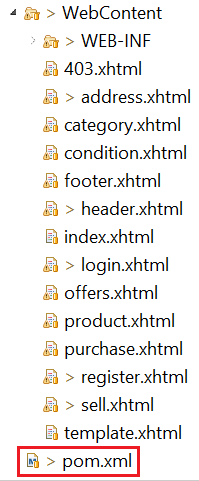




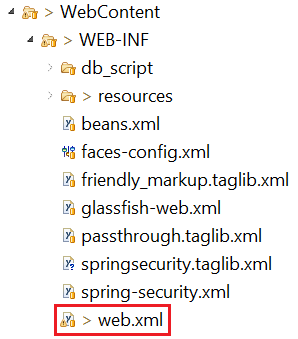
Abbildung 27 Spring Security - Abhängigkeiten in pom.xml

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:



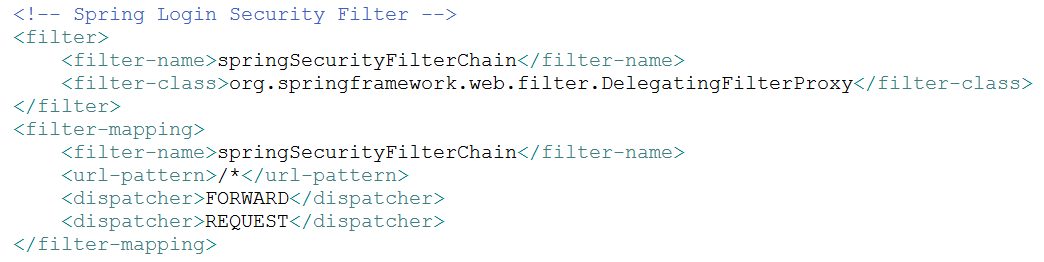


Abbildung 28 Spring Security - Filter Konfiguration

*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:

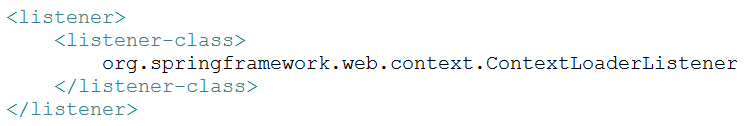


Abbildung 29 Spring Security - ContextLoaderListener

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

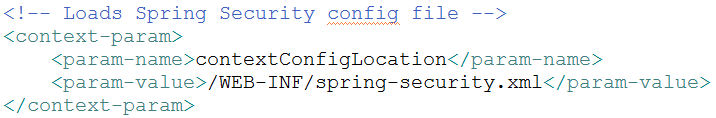


Abbildung 30 Spring Security - Verweis auf die Konfigurations-Datei

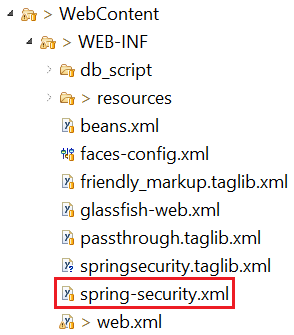




Abbildung 31 Spring Security - Login/Logout-Konfiguration (spring-security.xml)

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**

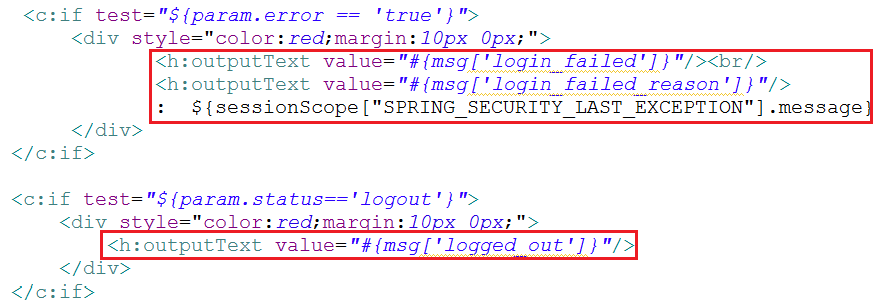
**Wenn keine eigene Login-Seite definiert wurde, wird von Spring Security eine Default-Login-Seite angezeigt. In meinem Projekt** habe ich jedoch eine eigene Login-Seite erstellt. Damit die Login-Seite wie gewünscht funktioniert, ist einiges zu beachten. Daher gehe ich hier etwas genauer auf die verschiedenen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



Abbildung 32 Spring Security – Webseite 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}

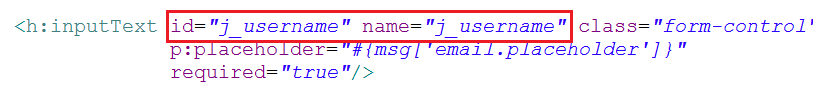
In der Konfiguration habe ich definiert, dass der Benutzer nach einer erfolgreichen Abmeldung direkt auf die Login-Seite umgeleitet wird. Der Wert logout des Parameters param.status aus der Konfiguration wird verwendet, um dem Benutzer bei einer erfolgreichen Abmeldung die Bestätigung mitzuteilen.

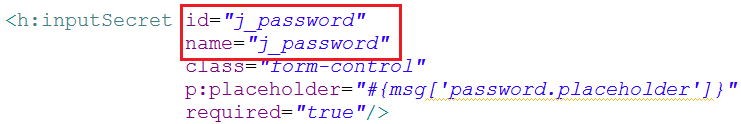


Es ist zudem sehr wichtig, dass für die Form der Standard HTML-Tag verwendet wird (also nicht den Form-Tag von JSF). Die Action muss den Wert erhalten, wie er in der Konfiguration *spring-security.xml* definiert wurde. In diesem Projekt habe ich für die login-processing-url den Standard-Wert j\_spring\_security\_check 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 vorhandene Feld email verwendet. Zusätzlich habe ich noch das Feld role eingeführt. Hier wird die Rolle des Benutzers gespeichert. Zurzeit 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. Mit authentication-manager wird definiert, wie der Benutzer authentifiziert werden soll. Um die Passwortsicherheit zu erhöhen, könnte man hier z.B. noch eine md5-Passwort-Kodierung wählen (password-encoder hash=“md5“), ich habe jedoch keine Passwort-Kodierung verwendet.

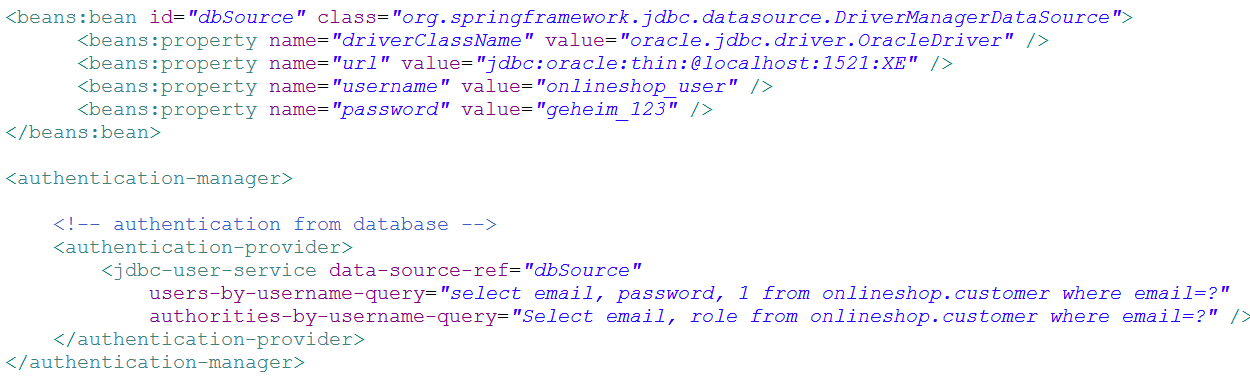


Abbildung 33 Spring Security - JDBC (spring-security.xml)

**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 Ressourcen müssen wiederum Anpassungen im File *spring-security.xml* vorgenommen werden.

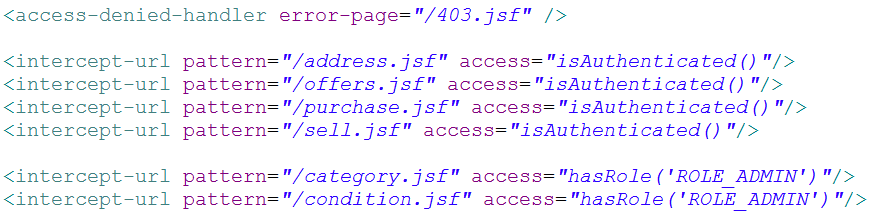


Abbildung 34 Spring Security - Autorisierung-Konfiguration (spring-security.xml)

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 Ressourcen 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 35 Spring Security - Gesamte 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 Taglib von Spring Security in einer JSP-Seite 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.

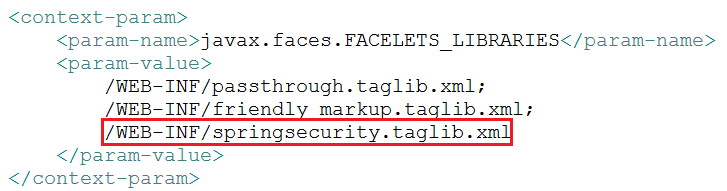


Abbildung 36 Spring Security - Verweis auf die Security-Taglib-Datei

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



Abbildung 37 Spring Security – 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 (*header.xhtml*). Um dieses Verhalten zu implementieren, ist der entsprechende Quellcode in das Spring-Security-Authorize-Tag geschachtelt.

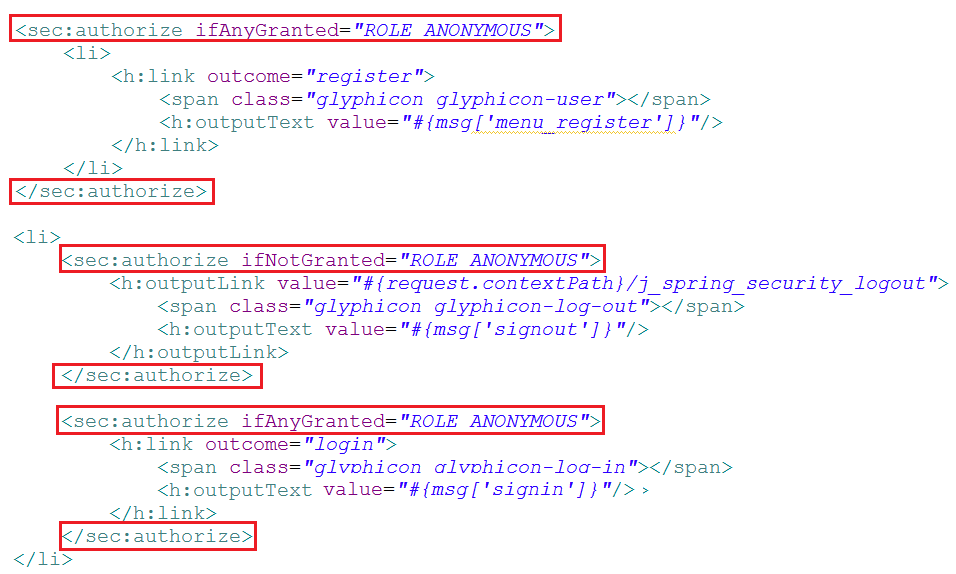


Abbildung 38 Spring Security - Verwendung der Security Taglib (header.xhtml)

Genauso 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.

## Schwerpunkt AJAX

Eine Webanwendung ohne Ajax-Funktionalität ist heute kaum noch vorstellbar. Mit JSF 2.0 wurde eine Schnittstelle für AJAX erarbeitet.

In diesem Abschnitt werde ich kurz beschreiben, wie ich einige JSF-Seiten in meinem Projekt um Ajax (Asynchronous JavaScript and XML) erweitert habe, so dass bei Änderungen nicht die komplette Seite neu geladen werden muss, sondern nur der Teil, der sich auch wirklich geändert hat. Dieser Abschnitt ist keine Einführung in die JSF-Programmierung oder Ajax. Ich möchte vielmehr zeigen, wie die grundlegende Funktionsweise von JSF-Seiten mit Ajax ist. Nach einem kurzen Vergleich von klassischen Webanwendungen ohne AJAX und Webanwendungen mit AJAX, werde ich die Anwendung von AJAX in JSF anhand des UseCase *Kategorie hinzufügen* aufzeigen.

Webanwendung ohne AJAX

Bei einer klassischen Webanwendung ohne AJAX wird bei jeder Änderung oder Aktion des Benutzers ein komplettes Neuladen der Webseite benötigt. Es muss eine Http-Verbindung zum Server aufgebaut werden, welcher dann die gesamten, aktualisierten Daten zurück an den Browser schickt.



Abbildung 39 Klassische Webanwendung ohne AJAX

Dieses Verhalten hat zur Folge, dass während der Server arbeitet, der Benutzer auf die Antwort warten muss. Erst nachdem der Server die Antwort geschickt hat, kann der Benutzer weiter-arbeiten. Des Weiteren muss immer die ganze Seite neu geladen werden, was natürlich zu Performance-Einbussen führt.

Webanwendung mit AJAX

Mit AJAX ist es möglich nur kleine Teile einer Webanwendung – genauer Daten und

Seiteninhalte der Webanwendung – an den Browser zu schicken.

Der grosse Vorteil ist, dass wesentlich weniger Daten übertragen werden müssen, da nur die Änderungen übertragen werden. Die Antwort des Servers kann dann mit JavaScript ausgewertet werden. Dies hat zur Folge, dass nur ein kleiner Teil der Webseite aktualisiert wird. Der Benutzer wird dabei nicht in seiner Arbeit auf der Webseite unterbrochen.



Abbildung 40 Webanwendung mit AJAX

JSF bietet seit der Version 2.0 zwei verschiedene Ansätze, um Ansichten mit AJAX-Funktionalität auszustatten. Zum einen können Entwickler zum Absetzen einer AJAX-Anfrage direkt die Funktion *jsf.ajax.request ()* der JavaScript-API aufrufen. Zum anderen gibt es mit dem Tag

*<f: ajax>* eine deklarative Variante, um eine Komponente oder sogar einen ganzen Bereich einer Seitendeklaration mit AJAX-Funktionalität auszustatten. In meinem Projekt habe ich ausschliesslich dieses Tag verwendet, denn damit konnte ich meiner Anwendung deklarativ das Verhalten von Ajax hinzufügen. Ein weiterer Vorteil ist, dass ich mir überhaupt keine Gedanken mehr um JavaScript machen musste.

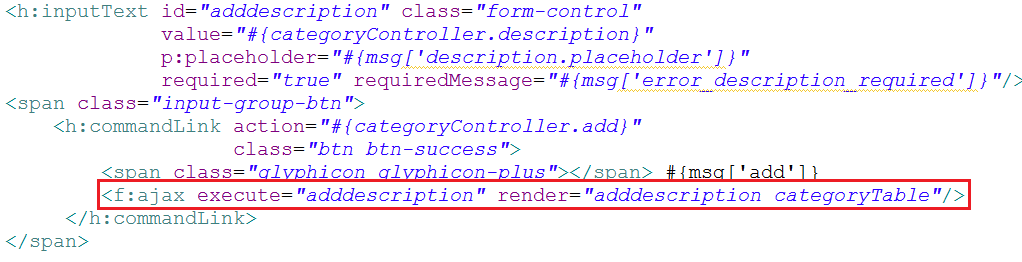
AJAX mit JSF für den UseCase *Kategorie hinzufügen*

Im UseCase Kategorie hinzufügen kann der Benutzer eine neue Kategorie erstellen, indem er die Bezeichnung der Kategorie in das entsprechende Feld schreibt und auf Hinzufügen drückt. Die neue Kategorie ist nach dem Hinzufügen sofort in der Tabelle mit allen Kategorien ersichtlich. Dabei wird nicht die gesamte Form zum Server geschickt, sondern nur der Inhalt des Feldes für die Bezeichnung. Gerendert wird ebenfalls nicht die gesamte View, sondern nur die Tabelle selbst.

Das Verwenden von AJAX in JSF ist im Prinzip sehr einfach: *<f:ajax>* muss in einem Formular stehen und schickt die Daten aller enthaltenen Felder, oder der Felder, deren IDs im Attribut *execute* in einer leerzeichengetrennten Liste aufgezählt sind, zum Server. Im Attribut *render* werden auf gleiche Weise die IDs der zu ändernden Komponenten eingetragen.

Der Server wertet die Seite neu aus und schickt für das Ziel bzw. die Ziele aus *render* das geänderte HTML in einem XML-Dokument an den Browser, der per JavaScript die Änderungen im DOM vornimmt.

Mit dem Tag <f:ajax> innerhalb des <h:commandLink> habe ich das Ajax-Verhalten hinzugefügt. Beim Drücken des CommandLinks erfolgt nun kein Full Page Reload mehr, sondern es wird ein Ajax-Request ausgelöst. Da ich die Attribute execute="adddescription" und render="adddescription categoryTable" definiert habe, wird der Lifecycle hierbei für die Komponente mit der id adddescription durchlaufen und danach die Tabelle mit der id categoryTable*,* also ein Teil der Gesamtseite, neu gerendert. Im render-Parameter habe ich zwei Elemente angegeben, da ich zusätzlich noch das Feld für die Bezeichnung neu gezeichnet haben möchte (ansonsten steht dort noch der Wert, den der Benutzer eingeben hat):



In diesem Projekt habe ich AJAX an mehreren Orten verwendet:

* Kategorien hinzufügen
* Kategorien ändern oder löschen
* Kategorie-Tabelle sortieren
* Zustand hinzufügen
* Zustand ändern oder löschen
* Zustands-Tabelle sortieren
* Anzeige der Produkte-Details
* Kaufen eines Produkts
* Produkte nach Kategorie filtern
* Bewerten von Einkauf oder Verkauf
* Adresse ändern

# Test

## Testprotokolle

| ID / Bezeichnung | **T-001** | Produkt suchen (anonymer Benutzer) |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 12.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 54 Testergebnis T-001

| ID / Bezeichnung | **T-002** | Produkt nach Kategorie filtern (anonymer Benutzer) |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 12.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 55 Testergebnis T-002

| ID / Bezeichnung | **T-003** | Produktdetail anschauen (anonymer Benutzer) |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 12.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 56 Testergebnis T-003

| ID / Bezeichnung | **T-004** | Verkäufer-Infos und Bewertung anschauen (anonymer Benutzer) |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 10.11.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 57 Testergebnis T-004

| ID / Bezeichnung | **T-005** | Eine geschützte Seite wählen (anonymer Benutzer) |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 18.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 58 Testergebnis T-005

| ID / Bezeichnung | **T-006** | Registrieren |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 15.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 59 Testergebnis T-006

| ID / Bezeichnung | **T-007** | Anmelden |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 15.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 60 Testergebnis T-007

| ID / Bezeichnung | **T-008** | Produkt suchen (angemeldeter Benutzer) |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 15.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 61 Testergebnis T-008

| ID / Bezeichnung | **T-009** | Produkt nach Kategorie filtern (angemeldeter Benutzer) |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 15.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 62 Testergebnis T-009

| ID / Bezeichnung | **T-010** | Produktdetail anschauen (angemeldeter Benutzer) |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 15.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 63 Testergebnis T-010

| ID / Bezeichnung | **T-011** | Registrierungsdaten ändern |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 15.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 64 Testergebnis T-011

| ID / Bezeichnung | **T-012** | Angebot erstellen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 65 Testergebnis T-012

| ID / Bezeichnung | **T-013** | Produkt kaufen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 66 Testergebnis T-013

| ID / Bezeichnung | **T-014** | Einkaufsübersicht anschauen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 67 Testergebnis T-014

| ID / Bezeichnung | **T-015** | Einkauf bewerten |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 12.11.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 68 Testergebnis T-015

| ID / Bezeichnung | **T-016** | Angebotsübersicht anschauen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 69 Testergebnis T-016

| ID / Bezeichnung | **T-017** | Angebot ändern |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 70 Testergebnis T-017

| ID / Bezeichnung | **T-018** | Verkauf bewerten |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 12.11.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 71 Testergebnis T-018

| ID / Bezeichnung | **T-019** | Angebot löschen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 72 Testergebnis T-019

| ID / Bezeichnung | **T-020** | Kategorie hinzufügen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 73 Testergebnis T-020

| ID / Bezeichnung | **T-021** | Kategorie ändern |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 74 Testergebnis T-021

| ID / Bezeichnung | **T-022** | Kategorie löschen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 75 Testergebnis T-022

| ID / Bezeichnung | **T-023** | Zustand hinzufügen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 76 Testergebnis T-023

| ID / Bezeichnung | **T-024** | Zustand ändern |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 77 Testergebnis T-024

| ID / Bezeichnung | **T-025** | Zustand löschen |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 17.10.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 78 Testergebnis T-025

| ID / Bezeichnung | **T-026** | Abmelden |
| --- | --- | --- |
| Tester | Birgit Sturzenegger | |
| Datum Testdurchführung | 05.11.17 | |
| Testergebnis | Keine Fehler | |
| Fehlerbeschreibung |  | |

Tabelle 79 Testergebnis T-026

# Evaluierungsbericht

Anfangs war ich gar nicht begeistert von der JSF-Technologie. Am liebsten arbeite ich mit purem HTML und guten JavaScript-Libraries wie jQuery, Boostrap und Angular.js. Bei vorgefertigten Wrappern für UI-Controls bin ich eher etwas vorsichtig, da ich nicht weiss, wie gut diese funktionieren und ich nicht gleich erkenne, wie die einzelnen Controls in HTML-Tags aufgelöst werden. Ich habe mich aber trotz anfänglichen Vorbehalte entschieden, die Webapplikation komplett mit JSF umzusetzen. Im Laufe der Semesterarbeit gefiel mir das Framework jedoch immer besser und am Ende der Semesterarbeit fällt mein Resümee sehr positiv aus. JSF ist ein gut gelungenes und ausgereiftes UI-Framework für Webanwendungen im Java-Umfeld.

JSF erleichtert die Entwicklung von Web-Anwendungen durch eine API und eine Tag-Library. Die Tag-Library bietet eine grosse Anzahl an wiederverwendbaren Standard-UI-Komponenten an. Die Umwandlung der JSF-Tags in normale HTML Tags funktionierte gut. Einzig vom RadioButton war ich enttäuscht - dieser wird als HTML-Table aufgelöst, was mir gar nicht zugesagt hat. Sehr gut gefallen hat mir dafür, dass JSF den Datentransfer zwischen UI-Komponenten sehr einfach macht. Zusätzlich fand ich es äusserst hilfreich, dass JSF auf dem MVC-Pattern basiert – so habe ich im Code eine saubere Trennung zwischen Zustand, Verhalten und Darstellung.

Die AJAX Umsetzung von JSF finde ich sehr gelungen. Es ist wirklich sehr einfach AJAX in JSF zu verwenden und man benötigt dazu keine Zeile JavaScript Code.

Die Beans im Projekt wurden mit CDI annotiert, welches ein JEE Standard ist (z.B. *@Named, @SessionScoped, @Inject*). Dadurch kann man verschiedene Arten von Java-Komponenten mittels loser Koppelung miteinander verbinden, dies erfolgt sogar typensicher. Ich finde, das CDI einige sehr hilfreiche Ergänzungen zu JSF bietet und vermute, dass CDI noch weit mehr kann, als wie ich es verwendet habe.

Das Framework JSF kann durch die Verwendung anderer Frameworks leicht ergänzt werden, wie ich in meinem Schwerpunkthema Spring Security aufgezeigt habe.

Bei JSF ist sicherlich dann ein Mehrwert gegenüber anderen Frameworks gegeben, wenn ein eher Java-lastiges Team am Werk ist und es sich um eine hoch integrierte Business-Anwendung handelt oder die Anwendung stark formularbasiert ist. Der JSF Lifecycle ist für die Verarbeitung von Nutzereingaben optimiert und so ist die Konvertierung von Eingaben, deren Validierung und das Binding an Domänenobjekte bereits fest in den Lifecycle integriert.

Für diejenigen, die eine Webanwendung lieber mit reinen Webtechnologien schreiben, ist wahrscheinlich AngularJS oder evtl. Spring MVC die bessere Wahl. Mir ist es nach wie vor lieber, wenn ich grössere Kontrolle darüber habe, was der Browser des Anwenders am Ende wirklich zeigt und ausführen soll.

# 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.