

Fachkonzept Projekt LFH

FFHS 2013

Version: 0.1

Autor: Jonas Alder, Patrick Bösch, Sandro Dallo, Andy Villiger

Studiengang: Bsc Inf 2011

Ort: Zürich

Datum: 14.11.2013

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Wer | Bemerkungen | Datum |
| 0.1 | BOE | Entwurf | 14.11.2013 |

Inhalt

[2 Technologieentscheid 2](#_Toc372176735)

[2.1 Einleitung 2](#_Toc372176736)

[2.2 Frontend 2](#_Toc372176737)

[2.2.1 HTML und CSS 2](#_Toc372176738)

[2.2.2 JavaScript 2](#_Toc372176739)

[2.3 Backend 3](#_Toc372176740)

[2.3.1 Business-Logik 3](#_Toc372176741)

[2.3.2 Web-Framework 3](#_Toc372176742)

[2.3.3 Datenbank 3](#_Toc372176743)

[3 Use Cases 4](#_Toc372176744)

[3.1 Registrierung 4](#_Toc372176745)

[3.2 Login 4](#_Toc372176746)

[3.3 Userdaten editieren 4](#_Toc372176747)

[3.4 Kunden editieren / sperren / löschen 4](#_Toc372176748)

[3.4.1 Kunde editieren 4](#_Toc372176749)

[3.4.2 Kunde editieren / löschen 5](#_Toc372176750)

[3.5 Produkt erfassen 5](#_Toc372176751)

[3.6 Produkt editieren 5](#_Toc372176752)

[3.7 Produkt bewerten 5](#_Toc372176753)

[3.8 Produkt suchen 6](#_Toc372176754)

[3.9 Produkt in den Warenkorb 6](#_Toc372176755)

[3.10 Warenkorb editieren 6](#_Toc372176756)

[3.11 Bestellen 6](#_Toc372176757)

[3.12 Transaktionsübersicht 7](#_Toc372176758)

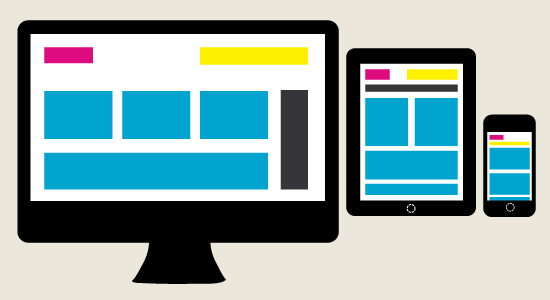
[4 GUI Prototyp 8](#_Toc372176759)

[5 UML 9](#_Toc372176760)

[6 ERM 10](#_Toc372176761)

# Technologieentscheid

## Einleitung

Da der Webshop auf verschiedensten Geräten und demzufolge Auflösungen funktionieren soll, haben wir uns dazu entschieden, die Applikation *responsive*[[1]](#footnote-1) zu machen. Das heisst wir brauchen im Frontend eine Technologie welche sich flexibel den Gegebenheiten jedes Geräts anpassen kann.

Im Backend haben wir verschiedene Technologien untersucht. Diese muss sich für die schnelle und praxistaugliche Entwicklung von Webapplikationen eignen. Eine weitere Vorgabe war, dass der Code im Backend zum grössten Teil in Java geschrieben wird. Dies schränkte uns in der Auswahl zwar ein, jedoch sind für Java sehr viele verschiedene Web-Frameworks verfügbar.

Abbildung 1: Responsive Design

## Frontend

### HTML und CSS

Die Applikation basiert im Frontend grundsätzlich auf HTML (in der Version 5) und CSS. CSS werden wir mithilfe von LESS[[2]](#footnote-2) schreiben. Das erleichtert uns verschachtelte CSS-Anweisungen zu schreiben. Zudem können beispielsweise für Grössenangaben und für Farben verschiedene Variablen definiert werden, die man dann an unterschiedlichen Stellen verwenden kann.

### JavaScript

Wir haben uns hier für Bootstrap[[3]](#footnote-3) entschieden, da einige Mitglieder des Projektteams bereits Erfahrung mit diesem JavaScript-Framework haben und sich dieses seit der neusten Version ausgezeichnet zur Entwicklung von *responsive* Applikationen eignet. Bootstrap, eine Technologie die ursprünglich im Hause Twitter entwickelt wurde, hat sich in der Zwischenzeit zu einem der beliebtesten Frameworks in diesem Bereich entwickelt. Es gibt dem Entwickler viele Werkzeuge in die Hand, z. B. bezüglich der Struktur der Applikation, Standardelemente wie Buttons und Formulare, dynamische Elemente wie Tooltips, Popups und Navigationen.

Zusätzlich werden wir im Bereich JavaScript auf jQuery[[4]](#footnote-4) setzen. Das ist der quasi Standard für jede Webapplikation heutzutage. Damit wird es uns vereinfacht Elemente im DOM zu selektieren und zu verändern, AJAX-Requests[[5]](#footnote-5) abzusenden und Animationen einzubauen.

## Backend

### Business-Logik

Die Business-Logik (das „Model“ im Sinne von MVC[[6]](#footnote-6) – Model View Controller) wird in Java geschrieben. Java ist eine Vorgabe für dieses Projekt von der FFHS.

### Web-Framework

Wir setzen auf das Play Framework[[7]](#footnote-7) in der Version 2. Play ist ein modernes Web-Framework für Java und Scala. Es ist nach dem Prinzip von MVC aufgebaut. Das Model und die Controller werden in Java geschrieben, die Views werden in einer speziellen Template-Sprache angefertigt (HTML und beschränkte Logik-Elemente im Template-Dialekt).

Das Play Framework ist in der Ausführungsgeschwindigkeit sehr schnell, da es auf dem Prinzip von Aktoren[[8]](#footnote-8) basiert. Jeder Request der vom Play Framework verarbeitet wird ist als eigener Aktor gekapselt. Aktoren werden parallel ausgeführt, über mehrere Prozessorkerne, Prozessoren oder sogar Rechner. Dies ermöglicht der Applikation eine nahezu lineare Skalierung.

Zudem ist das Play Framework in der Handhabung sehr einfach. Der Einstieg wird uns dadurch erleichtert, dass mit Generatoren die Applikation, neue Controller, Models oder Views erstellt werden können. Play hat zudem einen integrierten Web-Server, den wir während der Entwicklung auf unseren Rechnern ohne zusätzlichen Aufwand ausführen können.

### Datenbank

Als Datenbank setzen wir auf MySQL.[[9]](#footnote-9) Alle Projektmitglieder haben damit Erfahrung. Die Datenbank ist sehr verbreitet, Open-Source und steht deshalb frei zur Verfügung. Ausserdem sind für fast alle Programmiersprachen und Frameworks standardmässig Schnittstellen vorhanden, mit denen einfach auf MySQL zugegriffen werden kann.

Als Schnittstelle zwischen unserer Applikation und der Datenbank setzen wir Ebean[[10]](#footnote-10) ein, das Standard-ORM[[11]](#footnote-11) des Play Frameworks. Die Verbindung kommt dabei per JDBC zu stande, um die sich aber das Framework selber kümmert.

# Use Cases

## Registrierung

|  |  |
| --- | --- |
| Use Case ID | USC\_001 |
| Ziel | Benutzer ist am System registriert |
| Akteur | Benutzer |
| Vorbedingung | Verbindung zum System möglich |
| Nachbedingung | Akteur hat ein Benutzerkonto angelegt |
| Ablauf | 1. Akteur startet Registrierungsprozess 2. Akteur gibt E-Mailadresse als Username ein 3. Akteur gibt Adresse ein 4. Akteur bestätigt die Registrierung. 5. Akteur erhält vom System eine E-Mail mit dem Initialpasswort, welches er beim ersten Login ändern muss. |
| Varianten |  |

## Login

|  |  |
| --- | --- |
| Use Case ID | USC\_002 |
| Ziel | Benutzer loggt sich ins System ein |
| Akteur | Benutzer, Administrator |
| Vorbedingung | Verbindung zum System möglich |
| Nachbedingung | Akteur ist am System angemeldet |
| Ablauf | 1. Akteur gibt E-Mailadresse als Username ein 2. Akteur gibt sein Passwort ein 3. Akteur klickt auf Login |
| Varianten |  |

## Userdaten editieren

|  |  |
| --- | --- |
| Use Case ID | USC\_003 |
| Ziel | Benutzer ändert seine Daten |
| Akteur | Benutzer, Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Die neuen Daten sind im System gespeichert |
| Ablauf | 1. Akteur öffnet die Seite „Persönliche Daten bearbeiten“ 2. Akteur gibt die neuen Daten ein 3. Akteur bestätigt die Änderungen |
| Varianten |  |

## Kunden editieren / sperren / löschen

### Kunde editieren

|  |  |
| --- | --- |
| Use Case ID | USC\_004 |
| Ziel | Benutzer editieren |
| Akteur | Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Die Änderungen sind im System gespeichert. |
| Ablauf | 1. Akteur öffnet die Seite „Benutzerdaten bearbeiten“ 2. Akteur sucht den Benutzer 3. Akteur gibt die neuen Daten ein 4. Akteur bestätigt die Änderungen |
| Varianten |  |

### Kunde editieren / löschen

|  |  |
| --- | --- |
| Use Case ID | USC\_005 |
| Ziel | Benuter sperren oder löschen |
| Akteur | Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Die Änderungen sind im System gespeichert. |
| Ablauf | 1. Akteur öffnet die Seite „Benutzerdaten bearbeiten“ 2. Akteur sucht den Benutzer 3. Akteur löscht oder sperrt den Benutzer 4. Akteur bestätigt die Änderungen |
| Varianten |  |

## Produkt erfassen

|  |  |
| --- | --- |
| Use Case ID | USC\_006 |
| Ziel | Produkt erfassen |
| Akteur | Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Die Änderungen sind im System gespeichert. |
| Ablauf | 1. Akteur öffnet die Seite „Produkt erfassen“ 2. Akteur gibt Produktdaten ein 3. Akteur bestätigt die Änderungen |
| Varianten |  |

## Produkt editieren

|  |  |
| --- | --- |
| Use Case ID | USC\_007 |
| Ziel | Produkt editieren |
| Akteur | Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Die Änderungen sind im System gespeichert. |
| Ablauf | 1. Akteur öffnet die Seite „Produkt editieren“ 2. Akteur sucht das Produkt 3. Akteur gibt die neuen Produktdaten ein 4. Akteur bestätigt die Änderungen |
| Varianten |  |

## Produkt bewerten

|  |  |
| --- | --- |
| Use Case ID | USC\_008 |
| Ziel | Produkt bewerten |
| Akteur | Benutzer, Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Die Änderungen sind im System gespeichert. |
| Ablauf | 1. Akteur sucht das Produkt im Shop 2. Akteur bewertet das Produkt 3. Akteur bestätigt die Änderungen |
| Varianten |  |

## Produkt suchen

|  |  |
| --- | --- |
| Use Case ID | USC\_009 |
| Ziel | Produkt suchen |
| Akteur | Benutzer, Administrator |
| Vorbedingung | Verbindung zum System möglich |
| Nachbedingung | Das gewünschte Produkt wird gefunden. |
| Ablauf | 1. Akteur sucht das Produkt im Shop |
| Varianten |  |

## Produkt in den Warenkorb

|  |  |
| --- | --- |
| Use Case ID | USC\_010 |
| Ziel | Produkt in den Warenkorb |
| Akteur | Benutzer, Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Das gewünschte Produkt ist im Warenkorb. |
| Ablauf | 1. Akteur sucht das Produkt im Shop 2. Akteur klickt auf die Schaltfläche „In den Warenkorb“ |
| Varianten |  |

## Warenkorb editieren

|  |  |
| --- | --- |
| Use Case ID | USC\_011 |
| Ziel | Warenkorb editieren |
| Akteur | Benutzer, Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Die Änderungen am Warenkorb sind im System. |
| Ablauf | 1. Akteur ruft den Warenkorb auf 2. Akteur editiert den Warenkorb 3. Akteur speichert die Änderungen |
| Varianten |  |

## Bestellen

|  |  |
| --- | --- |
| Use Case ID | USC\_012 |
| Ziel | Bezahlen |
| Akteur | Benutzer, Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Die Bestellung ist verbindlich aufgeben. |
| Ablauf | 1. Akteur ruft den Warenkorb auf 2. Akteur wählt Zahlungsart. 3. Akteur klickt auf „Kostenpflichtig bestellen“ |
| Varianten |  |

## Transaktionsübersicht

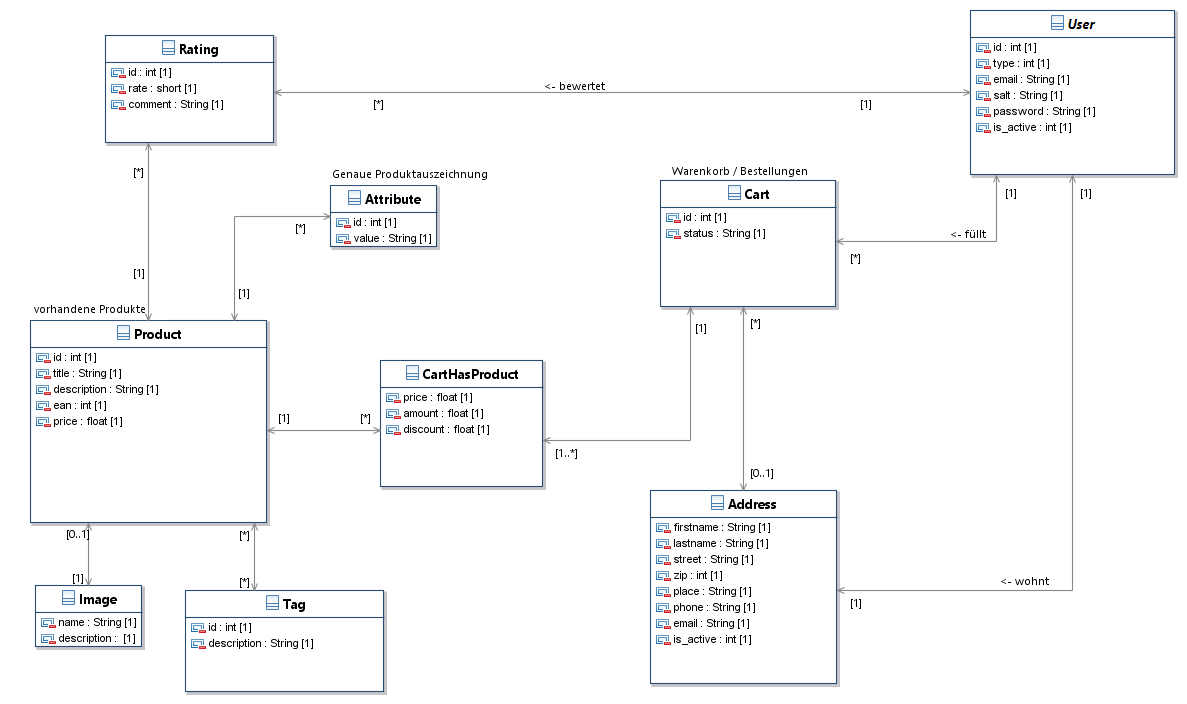
|  |  |
| --- | --- |
| Use Case ID | USC\_013 |
| Ziel | Transaktionsübersicht anzeigen |
| Akteur | Benutzer, Administrator |
| Vorbedingung | Akteur ist am System angemeldet |
| Nachbedingung | Der Akteur sieht die Übersicht seiner Transaktionen. |
| Ablauf | 1. Akteur ruft die Seite „Transaktionsübersicht“ auf |
| Varianten |  |

# GUI Prototyp



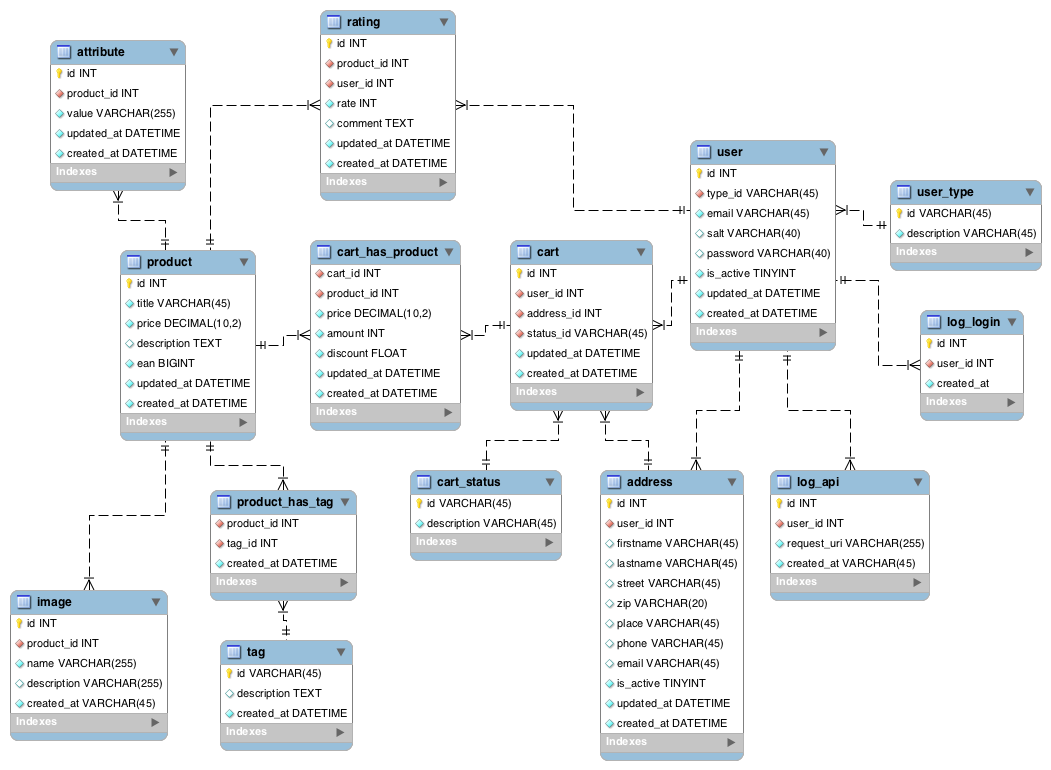
# 

# UML



# 

# ERM



1. http://de.wikipedia.org/wiki/Responsive\_Webdesign [↑](#footnote-ref-1)
2. http://lesscss.org/ [↑](#footnote-ref-2)
3. http://getbootstrap.com/ [↑](#footnote-ref-3)
4. http://jquery.com/ [↑](#footnote-ref-4)
5. http://de.wikipedia.org/wiki/Ajax\_(Programmierung) [↑](#footnote-ref-5)
6. http://de.wikipedia.org/wiki/Model\_View\_Controller [↑](#footnote-ref-6)
7. http://www.playframework.com/ [↑](#footnote-ref-7)
8. http://de.wikipedia.org/wiki/Actor\_Model [↑](#footnote-ref-8)
9. http://dev.mysql.com/ [↑](#footnote-ref-9)
10. http://www.avaje.org/ebean/introduction.html [↑](#footnote-ref-10)
11. http://de.wikipedia.org/wiki/Objektrelationale\_Abbildung [↑](#footnote-ref-11)