

Fachkonzept Projekt LFH

FFHS 2013

Version: 0.5

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

Studiengang: Bsc Inf 2011

Ort: Zürich

Datum: 18.11.2013

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Wer | Bemerkungen | Datum |
| 0.1 | BOE | Entwurf / initiale Version | 14.11.2013 |
| 0.2 | VIL | Technologieentscheid ausgearbeitet | 14.11.2013 |
| 0.3 | DAL | Use Cases hinzugefügt | 16.11.2013 |
| 0.4 | ALD | GUI Entwürfe hinzugefügt | 17.11.2013 |
| 0.5 | VIL | Diagramme aktualisiert, Klassenbeschreibung hinzugefügt | 18.11.2013 |

Inhalt

2 Technologieentscheid 4

2.1 Einleitung 4

2.2 Frontend 4

2.2.1 HTML und CSS 4

2.2.2 JavaScript 4

2.3 Backend 5

2.3.1 Business-Logik 5

2.3.2 Web-Framework 5

2.3.3 Datenbank 5

3 Use Cases 6

3.1 Registrierung 6

3.2 Login 6

3.3 Userdaten editieren 7

3.4 Kunden editieren / sperren / löschen 7

3.4.1 Kunde editieren 7

3.4.2 Kunde sperren 8

3.4.3 Kunde löschen 8

3.5 Produkt erfassen/editieren 9

3.6 Produkt bewerten 9

3.7 Produkt suchen 10

3.8 Produkt in den Warenkorb 10

3.9 Warenkorb editieren 10

3.10 Bestellen 11

3.11 Transaktionsübersicht 11

3.12 Produkte importieren 12

3.13 Produkte exportieren 12

3.14 Kunden importieren 13

3.15 Kunden exportieren 13

3.16 Bestellungen importieren 14

3.17 Bestellungen exportieren 14

4 GUI Prototyp 16

4.1 Frontend 16

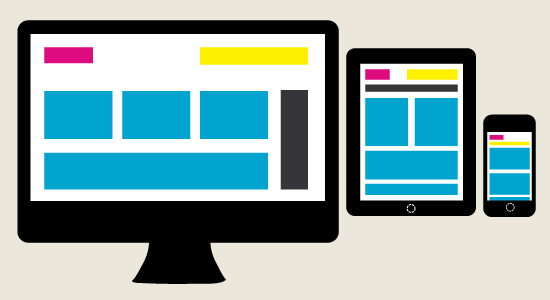
4.2 Backend 20

5 Klassendiagramm 24

6 ERM 26

# 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

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Benutzer registrieren | | |
| Kurzbeschreibung | Der Benutzer registriert sich am System und erhält Logindaten. | | |
| Akteur | ESHOMO-Besucher | | |
| Auslöser | Der Akteur beginnt mit der Registrierung | | |
| Vorbedingungen | * Verbindung zum System möglich | | |
| Eingehende Informationen | E-Mailadresse, Name, Vorname, Strasse, Strasse-Nr., PLZ, Ort, Telefon | | |
| Ergebnis | Registrierungsbestätigung mit Login Daten | | |
| Nachbedingungen | * Benutzerkonto wurde erstellt * E-Mail mit Login-Daten wurde verschickt. | | |
| Ablauf | 1. Akteur startet Registrierungsprozess 2. Akteur gibt E-Mailadresse als Username ein 3. Akteur gibt Adresse ein 4. Akteur bestätigt die Registrierung.   Akteur erhält vom System eine E-Mail mit dem Initialpasswort, welches er beim ersten Login ändern muss. | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Unverzichtbar, Hohe Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Login

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Benutzer Login | | |
| Kurzbeschreibung | Der Benutzer meldet sich am System an. | | |
| Akteur | Benutzer | | |
| Auslöser | Der Benutzer beginnt mit dem Login Prozess | | |
| Vorbedingungen | * Verbindung zum System möglich * Benutzer hat ein Benutzer Konto | | |
| Eingehende Informationen | E-Mailadresse, Passwort | | |
| Ergebnis | Benutzer ist am System angemeldet | | |
| Nachbedingung | * Der Benutzer ist angemeldet * Die Session wurde erstellt. | | |
| Ablauf | 1. Akteur gibt E-Mailadresse als Username ein 2. Akteur gibt sein Passwort ein 3. Akteur klickt auf Login | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Unverzichtbar, Hohe Priorität | | |
| Aufwand | 8 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Userdaten editieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Benutzerdaten editieren | | |
| Kurzbeschreibung | Der Benutzer ändert seine Daten | | |
| Akteur | Benutzer | | |
| Auslöser | Der Benutzer beginnt mit dem Login Prozess | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | E-Mailadresse, Name, Vorname, Strasse, Strasse-Nr., PLZ, Ort, Telefon | | |
| Ergebnis | Daten des Benutzers sind geändert. | | |
| Nachbedingung | * Die Daten des Benutzers sind auf der Datenbank geändert. | | |
| Ablauf | 1. Akteur öffnet die Seite „Persönliche Daten bearbeiten“ 2. Akteur gibt die neuen Daten ein 3. Akteur bestätigt die Änderungen | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Unverzichtbar, Hohe Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Kunden editieren / sperren / löschen

### Kunde editieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Benutzer editieren | | |
| Kurzbeschreibung | Die Daten des Benutzers werden vom Administrator editiert | | |
| Akteur | Administrator | | |
| Auslöser | Der Akteur will Benutzerdaten aktualisieren | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | E-Mailadresse, Name, Vorname, Strasse, Strasse-Nr., PLZ, Ort, Telefon | | |
| Ergebnis | Daten des Benutzers sind geändert. | | |
| Nachbedingung | * Die Daten des Benutzers sind auf der Datenbank geändert. | | |
| 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 | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Unverzichtbar, Hohe Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

### Kunde sperren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Benutzer sperren | | |
| Kurzbeschreibung | Das Konto des Benutzers wird vom Administrator gesperrt | | |
| Akteur | Administrator | | |
| Auslöser | Der Akteur will Benutzerdaten sperren | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | E-Mailadresse, Name, Vorname, Strasse, Strasse-Nr., PLZ, Ort, Telefon | | |
| Ergebnis | Benutzer ist gesperrt. | | |
| Nachbedingung | * Der Benutzer kann sich nicht mehr am System anmelden. | | |
| Ablauf | 1. Akteur öffnet die Seite „Benutzerdaten bearbeiten“ 2. Akteur sucht den Benutzer 3. Akteur sperrt den Benutzer 4. Akteur bestätigt die Änderungen | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

### Kunde löschen

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Benutzer löschen | | |
| Kurzbeschreibung | Das Konto des Benutzers wird vom Administrator gelöscht | | |
| Akteur | Administrator | | |
| Auslöser | Der Akteur will Benutzerdaten löschen | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | E-Mailadresse, Name, Vorname, Strasse, Strasse-Nr., PLZ, Ort, Telefon | | |
| Ergebnis | Benutzer ist gelöscht. | | |
| Nachbedingung | * Die Daten des Benutzers sind auf der Datenbank gelöscht. | | |
| Ablauf | 1. Akteur öffnet die Seite „Benutzerdaten bearbeiten“ 2. Akteur sucht den Benutzer 3. Akteur löscht den Benutzer 4. Akteur bestätigt die Änderungen | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 8 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Produkt erfassen/editieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Produkt erstellt bzw. editiert | | |
| Kurzbeschreibung | Das Produkt wird vom Akteur erstellt bzw. editiert | | |
| Akteur | Administrator | | |
| Auslöser | Der Akteur will Produkt erstellen bzw. editieren | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | Titel, Preis, Beschreibung, Artikel-Nr. | | |
| Ergebnis | Produkt ist erstellt bzw. geändert. | | |
| Nachbedingung | * Die Daten des Produkts sind auf der Datenbank erstellt bzw. geändert. | | |
| Ablauf | 1. Akteur sucht das Produkt 2. Akteur gibt die neuen Produktdaten ein bzw. löscht das Produkt 3. Akteur bestätigt die Änderungen | | |
| Alternativer Ablauf | 1. Akteur erstellt ein neues Produkt 2. Akteur speichert Änderungen | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Unverzichtbar, Hohe Priorität | | |
| Aufwand | 40 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Produkt bewerten

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Produkt bewerten | | |
| Kurzbeschreibung | Der Akteur bewertet ein Produkt | | |
| Akteur | Benutzer, Administrator | | |
| Auslöser | Der Akteur will ein Produkt bewerten | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | Bewertung, Beschreibung | | |
| Ergebnis | Die Bewertung ist hinterlegt. | | |
| Nachbedingung | * Die Bewertung ist in der Datenbank gespeichert. | | |
| Ablauf | 1. Akteur sucht das Produkt im Shop 2. Akteur bewertet das Produkt 3. Akteur bestätigt die Änderungen | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 24 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Produkt suchen

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Produkt suchen | | |
| Kurzbeschreibung | Der Akteur sucht ein Produkt | | |
| Akteur | Benutzer, Administrator | | |
| Auslöser | Der Akteur will ein Produkt bewerten | | |
| Vorbedingungen | * Verbindung zum System möglich | | |
| Eingehende Informationen | Suchbegriff | | |
| Ergebnis | Suchergebnisse werden angezeigt. | | |
| Nachbedingung | * Der Akteur sieht eine Liste der gefundenen Produkte. | | |
| Ablauf | 1. Akteur sucht das Produkt im Shop | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 8 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Produkt in den Warenkorb

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Produkt in den Warenkorb | | |
| Kurzbeschreibung | Der Akteur fügt ein Produkt zum Warenkorb hinzu. | | |
| Akteur | Benutzer, Administrator | | |
| Auslöser | Der Akteur will ein Produkt dem Warenkorb hinzufügen. | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | Suchbegriff, Produkt-ID | | |
| Ergebnis | Das Produkt ist im Warenkorb. | | |
| Nachbedingung | * Das Produkt wurde dem Warenkorb hinzugefügt. | | |
| Ablauf | 1. Akteur sucht das Produkt im Shop 2. Akteur klickt auf die Schaltfläche „In den Warenkorb“ | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Unverzichtbar, Hohe Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Warenkorb editieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Produkt in den Warenkorb editieren | | |
| Kurzbeschreibung | Der Akteur editiert ein Produkt im Warenkorb. | | |
| Akteur | Benutzer, Administrator | | |
| Auslöser | Der Akteur will ein Produkt im Warenkorb editieren. | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | Anzahl | | |
| Ergebnis | Die geänderten Warenkorb Daten sind gespeichert. | | |
| Nachbedingung | * Die Daten im Warenkorb wurden geändert. | | |
| Ablauf | 1. Akteur ruft den Warenkorb auf 2. Akteur editiert den Warenkorb 3. Akteur speichert die Änderungen | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Unverzichtbar, Hohe Priorität | | |
| Aufwand | 8 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Bestellen

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Bestellen | | |
| Kurzbeschreibung | Der Akteur bestellt einen Warenkorb. | | |
| Akteur | Benutzer, Administrator | | |
| Auslöser | Der Akteur will einen Warenkorb bestellen. | | |
| Vorbedingungen | * Akteur ist am System angemeldet * Produkte sind im Warenkorb | | |
| Eingehende Informationen | Warenkorb ID | | |
| Ergebnis | Die Bestellung ist erfolgreich ausgeführt. | | |
| Nachbedingung | * Der Warenkorb wurde im System zur Bestellung. | | |
| Ablauf | 1. Akteur ruft den Warenkorb auf 2. Akteur wählt Zahlungsart. 3. Akteur klickt auf „Kostenpflichtig bestellen“ | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Unverzichtbar, Hohe Priorität | | |
| Aufwand | 32 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Transaktionsübersicht

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Transaktionsübersicht | | |
| Kurzbeschreibung | Der Akteur will seine bisherigen Bestellungen sehen. | | |
| Akteur | Benutzer, Administrator | | |
| Auslöser | Der Akteur will seine bisherigen Bestellungen sehen. | | |
| Vorbedingungen | * Akteur ist am System angemeldet | | |
| Eingehende Informationen | Benutzer ID | | |
| Ergebnis | Die Bestellungen werden aufgelistet. | | |
| Nachbedingung | * Der Benutzer erhält eine Liste seiner bisherigen Bestellungen. | | |
| Ablauf | 1. Akteur ruft die Seite „Transaktionsübersicht“ auf | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 8 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Produkte importieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Produkte importieren | | |
| Kurzbeschreibung | Der Akteur importiert Produkte über eine Schnittstelle. | | |
| Akteur | MAVE System | | |
| Auslöser | Der Akteur will Produkte über eine Schnittstelle importieren. | | |
| Vorbedingungen | * Das System ist erreichbar | | |
| Eingehende Informationen | Produkt-Daten, Request-Type, Login-Information | | |
| Ergebnis | Die Produkte wurden dem System hinzugefügt. | | |
| Nachbedingung | * Die Produkte wurden in der Datenbank persistiert. | | |
| Ablauf | 1. Akteur ruft die Import URL auf 2. Login wird geprüft 3. Dateiformat wird geprüft. 4. Daten werden importiertf | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Produkte exportieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Produkte exportieren | | |
| Kurzbeschreibung | Der Akteur exportieren Produkte über eine Schnittstelle. | | |
| Akteur | MAVE System | | |
| Auslöser | Der Akteur will Produkte über eine Schnittstelle exportieren. | | |
| Vorbedingungen | * Das System ist erreichbar | | |
| Eingehende Informationen | Produkt-Daten, Request-Type, Login-Information | | |
| Ergebnis | Die Produkte wurden an MAVE übergeben. | | |
| Nachbedingung | * MAVE hat die Produkte erhalten. | | |
| Ablauf | 1. Akteur ruft die Export URL auf 2. Login wird geprüft 3. Daten werden gesucht. 4. Daten werden im gewünschten Format aufbereitet | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Kunden importieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Kunden importieren | | |
| Kurzbeschreibung | Der Akteur importiert Kunden über eine Schnittstelle. | | |
| Akteur | MAVE System | | |
| Auslöser | Der Akteur will Kunden über eine Schnittstelle importieren. | | |
| Vorbedingungen | * Das System ist erreichbar | | |
| Eingehende Informationen | Kunden-Daten, Request-Type, Login-Information | | |
| Ergebnis | Die Kunden wurden dem System hinzugefügt. | | |
| Nachbedingung | * Die Kunden wurden in der Datenbank persistiert.. | | |
| Ablauf | 1. Akteur ruft die Import URL auf 2. Login wird geprüft 3. Dateiformat wird geprüft. 4. Daten werden importiertf | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Kunden exportieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Kunden exportieren | | |
| Kurzbeschreibung | Der Akteur exportieren Kunden über eine Schnittstelle. | | |
| Akteur | MAVE System | | |
| Auslöser | Der Akteur will Kunden über eine Schnittstelle exportieren. | | |
| Vorbedingungen | * Das System ist erreichbar | | |
| Eingehende Informationen | Kunden -Daten, Request-Type, Login-Information | | |
| Ergebnis | Die Kunden wurden an MAVE übergeben. | | |
| Nachbedingung | * MAVE hat die Kunden erhalten. | | |
| Ablauf | 1. Akteur ruft die Export URL auf 2. Login wird geprüft 3. Daten werden gesucht. 4. Daten werden im gewünschten Format aufbereitet | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Bestellungen importieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Bestellungen importieren | | |
| Kurzbeschreibung | Der Akteur importiert Bestellungen über eine Schnittstelle. | | |
| Akteur | MAVE System | | |
| Auslöser | Der Akteur will Bestellungen über eine Schnittstelle importieren. | | |
| Vorbedingungen | * Das System ist erreichbar | | |
| Eingehende Informationen | Bestellungen -Daten, Request-Type, Login-Information | | |
| Ergebnis | Die Bestellungen wurden dem System hinzugefügt. | | |
| Nachbedingung | * Die Bestellungen wurden in der Datenbank persistiert.. | | |
| Ablauf | 1. Akteur ruft die Import URL auf 2. Login wird geprüft 3. Dateiformat wird geprüft. 4. Daten werden importiertf | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

## Bestellungen exportieren

|  |  |  |  |
| --- | --- | --- | --- |
| Beschreibung Systemanwendungsfall | | | |
| Name | Bestellungen exportieren | | |
| Kurzbeschreibung | Der Akteur exportieren Bestellungen über eine Schnittstelle. | | |
| Akteur | MAVE System | | |
| Auslöser | Der Akteur will Bestellungen über eine Schnittstelle exportieren. | | |
| Vorbedingungen | * Das System ist erreichbar | | |
| Eingehende Informationen | Bestellungen -Daten, Request-Type, Login-Information | | |
| Ergebnis | Die Bestellungen wurden an MAVE übergeben. | | |
| Nachbedingung | * MAVE hat die Bestellungen erhalten. | | |
| Ablauf | 1. Akteur ruft die Export URL auf 2. Login wird geprüft 3. Daten werden gesucht. 4. Daten werden im gewünschten Format aufbereitet | | |
| Ansprechpartner | LFH | | |
| Risiko |  | | |
| Verbindlichkeit, Priorität | Verzichtbar, Mittlere Priorität | | |
| Aufwand | 16 Std. | | |
| Stabilität | Stabil | | |
| Zeitpunkt, Dringlichkeit | Release 1.0 | | |
| Änderungen | Mitarb. | Status | Kommentar |
| 14.11.2013 | BOE | Entwurf | Erster Entwurf |

# GUI Prototyp

Die folgenden GUI Prototypen sollen das Layout und die Abläufe sowie Dialoge des Webshops visualisieren. Diese sind bewusst schlicht und in einem Comic ähnlichem Stil gehalten und sollen keine direkte Vorgabe für das Design sein.

## Frontend



Abbildung 2 Produkte Übersicht



Abbildung 3 Produkt Details



Abbildung 4 Registrierung



Abbildung 5 Warenkorb Übersicht



Abbildung 6 Bestellung



Abbildung 7 Persönliche Daten bearbeiten



Abbildung 8 Übersicht Bestellungen

## Backend



Abbildung 9 Produkte Übersicht im Backend



Abbildung 10 Produkt hinzufügen



Abbildung 11 User Übersicht



Abbildung 12 User hinzufügen

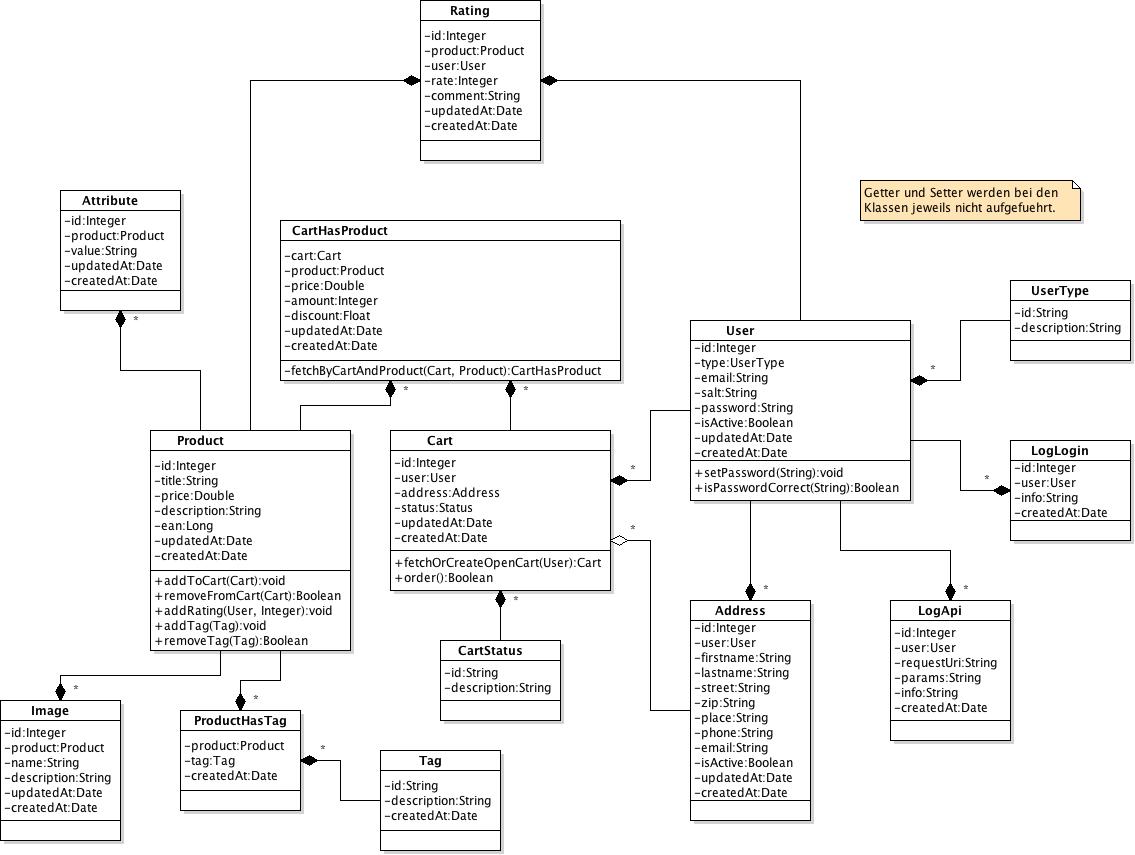


Abbildung 13 Übersicht Bestellung im Backend



Abbildung 14 Logs

# Klassendiagramm



# Beschreibung der Klassen

Im Folgenden werden die wichtigsten Klassen genauer beschrieben. Daraus erhält man einen guten Einblick in die Funktionalität der Business Logik.

## User

Die User-Klasse stellt einen zentralen Teil der Applikation dar. Die User-Klasse wird einerseits dafür verwendet, um in der Applikation zwischen einem Kunden und einem Admin zu unterscheiden. Nur der Admin hat jeweils Zugriff zum Administrationsbereich. Ausserdem wird die User-Klasse dafür verwendet, um einen User zu authentifizieren. Der User wird in der Session gespeichert, wenn sich ein Kunde oder Admin einloggt. Anhand des eingeloggten Users können wir der gerade eingeloggten Person den Warenkorb zuweisen, in dem Produkte gespeichert werden.

## Product

Jedes Produkt im System wird durch eine Instanz der Product-Klasse repräsentiert. Darauf können die Bilder des Produkts, die Beschreibung, der Titel, dessen Attribute und die Bewertungen abgefragt werden. Diese Klasse erlaubt uns also einerseits die Produkte im Shop anzuzeigen. Andererseits können damit die Produkte auch im Warenkorb abgelegt und bestellt werden.

## Cart

Die Klasse Cart repräsentiert den Warenkorb eines Users. Darin können mit der Zuordnungsklasse CartHasProduct die Produkte abgelegt werden. Bestellte Warenkörbe weisen den Status „ordered“ aus, der nicht bestellte Warenkorb ist mit dem Status „open“ versehen.

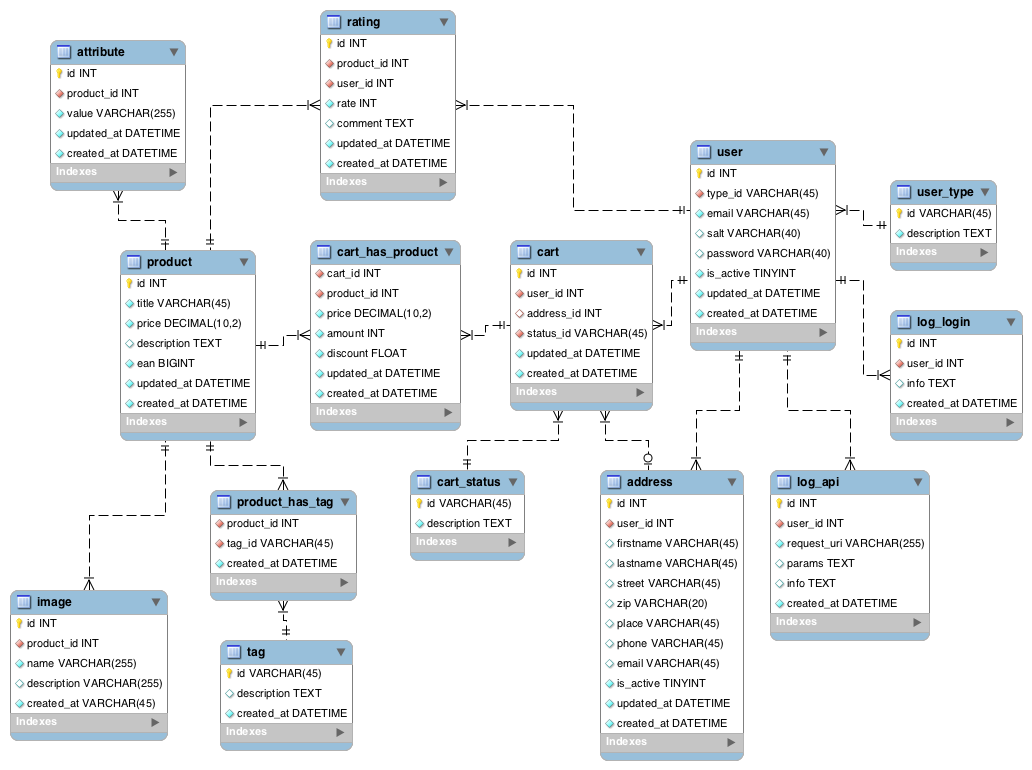
## Address

Die Klasse Address wird dafür Benötigt jeder Bestellung eine Adresse beizulegen. So kann der User mit der Zeit seine Adresse anpassen, doch jede bereits getätigte Bestellung ist mit der zur Bestellzeit aktuellen Adresse versehen.

## Rating

Produkte können von den Usern bewertet werden. Diese Bewertungen werden mithilfe der Klasse Rating dargestellt und im System abgelegt.

# 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)