Nicht alle Elemente im Teil 2 müssen genutzt werden. Projektmethode, Auftrag etc.. entscheidet. **Das Designe des Titelblatt und der restlichen IPA ist frei wählbar, der Inhalt im Teil eins muss aber der der Vorlage übereinstimmen! (Überschriften etc. usw..** Dokumentinformationen. Gesamter „Oranger-Hinweistext“ kann unter Formatvorlage/ auf IPA-Hinweis rechtklick/Formatierungen von Instanzen, gelöscht werden. Vor dieser Seite darf ein Titelblatt gestaltet werden. **Kopfzeile:** „Vorlage IPA 2018“ kann unter Speichern Unter/Titel mit dem Titel der IPA abgeändert werden so wie Autor.

IPA 2021, Kanton Bern

Terminal8 GmbH

Erweiterung TYPO3 CMS Website mit Produktedatenbank

|  |  |
| --- | --- |
| Hauptexperte : | Iwan Kalbermatten |
| Nebenexperte : | Nils Weibel |
| Valid-Experte : | Michel Patrick |
| Verantwortliche Fachkraft : | Fehmi Raqipi |
| Berufsbildner : | Michael Peter |
| Auftraggeber : | Probst-Maveg AG |
| Berufsschule : | Gibb |
| Fachrichtung : | Applikationsentwicklung |
| Projektvorgehensmodell : | HERMES 5.1 |
| Version : | 0.5 |

Autor: Moritz Burn

Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc69246830)

[Ablauf Organisation und Umfeld 7](#_Toc69246831)

[1. Aufgabenstellung 8](#_Toc69246832)

[1.1 Titel der Arbeit 8](#_Toc69246833)

[1.2 Thematik 8](#_Toc69246834)

[1.3 Ausgangslage 8](#_Toc69246835)

[1.4 Detaillierte Aufgabenstellung 9](#_Toc69246836)

[1.5 Teil 1: 9](#_Toc69246837)

[1.6 Teil 2: 10](#_Toc69246838)

[1.7 Mittel und Methoden 11](#_Toc69246839)

[1.8 Vorkenntnisse 11](#_Toc69246840)

[1.9 Vorarbeiten 11](#_Toc69246841)

[2. Organisation der IPA 12](#_Toc69246842)

[2.1 Datensicherung der IPA 12](#_Toc69246843)

[2.2 Versionierung 13](#_Toc69246844)

[2.2.1 Dokumentation 13](#_Toc69246845)

[2.2.2 Software 13](#_Toc69246846)

[2.3 Hilfestellungen 13](#_Toc69246847)

[3. Planung 14](#_Toc69246848)

[3.1 Phasen 15](#_Toc69246849)

[3.2 Meilensteine 15](#_Toc69246850)

[3.3 Szenario 16](#_Toc69246851)

[3.4 Module 16](#_Toc69246852)

[4. Zeitplan 18](#_Toc69246853)

[5. Firmenstandards 19](#_Toc69246854)

[6. Arbeitsjournal 20](#_Toc69246855)

[6.1 Arbeitsjournal 07.04.2021 20](#_Toc69246856)

[6.2 Arbeitsjournal 08.04.2021 21](#_Toc69246857)

[6.3 Arbeitsjournal 09.04.2021 22](#_Toc69246858)

[6.4 Arbeitsjournal 12.04.2021 23](#_Toc69246859)

[6.5 Arbeitsjournal 13.04.2021 24](#_Toc69246860)

[7. Abschlussbericht 30](#_Toc69246861)

[7.1 Vergleich Ist/Soll 30](#_Toc69246862)

[7.2 Mittelbedarf 30](#_Toc69246863)

[7.3 Realisierungsbericht 30](#_Toc69246864)

[7.4 Testbericht 30](#_Toc69246865)

[7.5 Fazit zum IPA (Projekt) 30](#_Toc69246866)

[7.6 Persönliches Fazit 30](#_Toc69246867)

[7.7 Schlussreflexion 30](#_Toc69246868)

[8. Unterschriften und Abnahmeprotokoll 31](#_Toc69246869)

[9. Initialisierung 32](#_Toc69246870)

[9.1 Ist-Zustand 32](#_Toc69246871)

[9.2 Projektziele 32](#_Toc69246872)

[9.3 Anforderungen 33](#_Toc69246873)

[9.3.1 Funktionale Anforderungen 33](#_Toc69246874)

[9.3.2 Nicht funktionale Anforderungen 33](#_Toc69246875)

[9.4 Risikoanalyse 34](#_Toc69246876)

[9.5 Risikograph 35](#_Toc69246877)

[9.5.1 Kurze Stellungnahmen zu den Risiken 35](#_Toc69246878)

[9.6 Schutzbedarfsanalyse 35](#_Toc69246879)

[9.7 Varianten 36](#_Toc69246880)

[9.7.1 Filterfunktion 36](#_Toc69246881)

[9.7.2 Nutzwertanalyse 37](#_Toc69246882)

[9.7.3 Entscheid 37](#_Toc69246883)

[9.7.4 Zwischenspeichern der Merkliste 38](#_Toc69246884)

[9.7.5 Nutzwertanalyse 39](#_Toc69246885)

[9.7.6 Entscheid 39](#_Toc69246886)

[10. Konzept 40](#_Toc69246887)

[10.1 ISDS-Konzept 40](#_Toc69246888)

[10.2 Systemumgebung, Anforderungen und Konfiguration 40](#_Toc69246889)

[10.2.1 Vorarbeiten Webserver 40](#_Toc69246890)

[10.2.2 Vorarbeiten TYPO3 41](#_Toc69246891)

[10.3 Architekturkonzept 43](#_Toc69246892)

[10.3.1 Model – View – Controller 43](#_Toc69246893)

[10.3.2 Listenansicht mit Filter 45](#_Toc69246894)

[10.3.3 Detailansicht 45](#_Toc69246895)

[10.3.4 Merkliste 45](#_Toc69246896)

[10.4 Rollenkonzept 46](#_Toc69246897)

[10.4.1 Projektrollen 46](#_Toc69246898)

[10.5 Datenbankkonzept (Nochmals prüfen) 47](#_Toc69246899)

[10.6 Klassendiagramm (noch anpassen, oberstes Element falsch, Rest richtig=>Datentypen nochmals prüfen) 48](#_Toc69246900)

[10.7 Use-Cases 49](#_Toc69246901)

[10.7.1 Normale Use-Cases 50](#_Toc69246902)

[10.8 Activity-Diagramm Frontend 53](#_Toc69246903)

[10.9 Activity-Diagramm Backend 54](#_Toc69246904)

[10.10 Testkonzept 55](#_Toc69246905)

[11. Realisierung 56](#_Toc69246906)

[11.1 Domainmodel erstellen 56](#_Toc69246907)

[11.2 Erweiterung installieren 56](#_Toc69246908)

[11.4 Testprotokoll 57](#_Toc69246909)

[11.4.1 Testfall 57](#_Toc69246910)

[11.4.2 Testdurchführung und Testergebnis 57](#_Toc69246911)

[11.4.3 Testwürdigung 58](#_Toc69246912)

[11.4.4 Testfazit 58](#_Toc69246913)

[11.4.5 Weiteres Vorgehen 58](#_Toc69246914)

[11.5 Einführung vorbereiten 59](#_Toc69246915)

[11.6 Schutzmassnahmen umsetzen 59](#_Toc69246916)

[12. Glossar 60](#_Toc69246917)

[13. Literatur- und Quellenverzeichnis 64](#_Toc69246918)

[14. Tabellen- und Abbildungsverzeichnis 65](#_Toc69246919)

[Abbildungsverzeichnis 65](#_Toc69246920)

[Tabellenverzeichnis 65](#_Toc69246921)

[15. Anhang 67](#_Toc69246922)

Tipp: nicht mehr als maximal drei Stufen gliedern

Änderungskontrolle, Prüfung, Genehmigung

|  | | | |
| --- | --- | --- | --- |
| Version | Datum | Name | Beschreibung |
| 0.0 | 28.02.2021 | M. Burn | Vorlage erstellen |
| 0.1 | 07.04.2021 | M. Burn | Projektvorgehen |
| 0.2 | 08.04.2021 | M. Burn | Risikoanalyse, Initialisierung |
| 0.3 | 09.04.2021 | M. Burn | Rest Initialisierung, Konzept |
| 0.3 | 09.04.2021 | M. Burn, F. Raqipi | Phasenfreigabe Konzept, MS1 |
| 0.4 | 12.04.2021 | M. Burn |  |

Verwendete Abkürzungen

Nur Abkürzungen, keine Begriffserklärungen. Diese gehören in ein Glossar. Alphabetisch sortiert.

| Abkürzung | Bedeutung |
| --- | --- |
| IPA | Individuelle praktische Arbeit |
| OdA | Organisation der Arbeitswelt |
| QV  VP | Qualifikationsverfahren  Verantwortliche Fachkraft |
| VZ | Verzeichnis |
| CMS | Content Management System |
| URL | «Uniform Ressource Locator» |
| DB | Datenbank |
| MS | Meilenstein |
| PHP | Programmiersprache |
| ERD | Entity-Relationship-Diagram / Datenbankmodell |
| FTP | Steht für «File transfer protocol», siehe [Glossar](#_Glossar) |
| EXT | Synonym für Extension, siehe [Glossar](#_Glossar) |

**Kurzfassung des IPA-Berichtes**

Ausgangssituation

Die Design & Webagentur Terminal8 in Bern entwickelt kreative Webanwendungen und setzt diese mit aktuellen Technologien um. Der Fokus liegt im Design und der Entwicklung von responsive Websites und Applikationen mit dem [CMS](#_Glossar) [TYPO3](#_Glossar).

Für den Kunden, die PROBST MAVEG AG, einen der führenden Anbieter von Baumaschinen und Baugeräten in der Schweiz, soll die alte, in die Jahre gekommen Website abgelöst werden. Hierfür hat Terminal8 den Gesamtauftrag erhalten: Konzeption, Design & Entwicklung. Einen wesentlichen Bestandteil der neuen Website stellt die “Produktdatenbank” dar, in welcher die wichtigsten Produkte dargestellt werden. Die aktuelle Lösung ist nicht zufriedenstellend und ist mit sehr grossem administrativem Aufwand verbunden, wenn neue Produkte hinzugefügt werden.

Umsetzung

Die neue Website wird durch Terminal8 mit dem CMS TYPO3 entwickelt, welche mit einer selbstentwickelten Produktdatenbank, einer sog «Extension» ergänzt wird. Zusätzlich werden eigens von Terminal8 entwickelte Erweiterungen (t8\_Jetpack und t8\_Custom) eingesetzt, die die Entwicklung um ein Gewisses erleichtern.

Das Datenbankmodell wird mit einer Hilfs-extension namens «[Extensionbuilder](#_Glossar)» erstellt, welche die in PHP nötigen [Getter und Setter](#_Glossar) und die Datenbankanbindung kreiert.

Für den Produkte-Filter wird ein Ajax-Call gemacht, der die nötigen Daten per [URL](#_Glossar)-Parameter aus der Datenbank ausliest und an die View übergibt.

Es soll zudem möglich sein, Produkte in eine Merkliste hinzuzufügen, um z.B deren Preise oder andere Eigenschaften zu vergleichen.

Um das Ganze abzurunden werden die Eigenschaften der Produkte mit HTML strukturiert und mit [SCSS](#_Glossar) verschönert. Wo nötig, werden Elemente mit JavaScript und CSS-Klassen animiert.

Ergebnis

Der Kunde kann eigenständig Produkte erfassen, löschen und nach Belieben anpassen. Alles ist übersichtlich und sinnvoll platziert, die Bedienung ist einfach, im Backend wie im Frontend. Produkte können nach Kategorien gefiltert werden, ohne dass die Seite neu geladen werden muss. Die Produkte werden in einer Listenansicht dargestellt und man kann über einen Link auf die Detailseite des jeweiligen Produkts gelangen.

Pro Produkt gibt es einen Button, mit dem man das Produkt in die Merkliste aufnehmen kann. Die Merkliste kann über einen separaten Button eingesehen werden.

Das Ganze soll natürlich auch auf Smartphones und Tablets reibungslos funktionieren, Stichwort «Responsive».

Ablauf Organisation und Umfeld

|  |  |
| --- | --- |
| **Autor** | Moritz Burn |
| **Jahr der Durchführung** | 2021 |
| **Projektvorgehensmodell** | HERMES 5.1 |

Ev. Firmen-/ Projektlogo hinzufügen

Etwas spannend gestalten… das Auge liest mit ☺

Die IPA findet im Sitzungszimmer von Terminal8 statt. Untenstehend ein Bild von besagter Räumlichkeit und dessen Einrichtung.



Abbildung 1, Arbeitsplatz

Abbildung 2, Logo T8

# Aufgabenstellung

## Titel der Arbeit

Erweiterung bestehende TYPO3 CMS Website mit Produktdatenbank.

## Thematik

Das Projekt befasst sich mit TYPO3 CMS, welches einen unter Anderem ermöglicht, eigene Software-Erweiterungen zu entwickeln, zu installieren und zu verwenden. Das Herz der Applikation besteht aus PHP und einer Datenbank, in der die Produkte, respektive deren Eigenschaften gespeichert und ausgelesen werden können. Zusätzlich werden für eine Merkliste Cookies verwendet, um die jeweiligen Benutzer bei erneutem Besuch der Website wiederzuerkennen und anschliessend die richtigen Produkte in der Merkliste auszugeben.

## Ausgangslage

Probst-Maveg ist einer der führenden Anbieter von Baumaschinen und Baugeräten aller Art und benötigt für den Verkauf ihrer Produkte eine elegantere Lösung. Die aktuelle Lösung ist mit grossem administrativem Aufwand verbunden und ist darum nicht zufriedenstellend.

Die Webagentur Terminal8 in Bern hat von Probst-Maveg AG den Gesamtauftrag für die Neuentwicklung der alten Website erhalten. Die neue Website wird durch Terminal8 mit dem CMS TYPO3 entwickelt, welche mit einer selbstentwickelten Produktdatenbank ergänzt wird.

Das Basis Template der Website und Installation von TYPO3, in welchem die Produktdatenbank integriert wird, werden vor der IPA vorbereitet. Nach der IPA wird die Produktdatenbank vom Kunden getestet und eingeführt. Die restlichen Bestandteile der Website werden nach der IPA in Teamarbeit fertiggestellt.

## Detaillierte Aufgabenstellung

Die IPA besteht aus zwei Teilen:

Teil 1 Backend und Teil 2 Frontend.

## Teil 1:

Die bestehende TYPO3 Installation soll um eine Extension erweitert werden, welche es dem Redaktor mit einer kleinen Benutzeranleitung erlaubt, Produkte mehrsprachig zu erfassen. Diese können dann mithilfe von Plugins auf verschiedenen Seiten gefiltert und sortiert ausgegeben werden.

* Redaktoren können im CMS neue Produkte erstellen, sowie bestehende bearbeiten oder löschen
* Redaktoren können Produkte in eine zweite Sprache übersetzen
* Produkte enthalten folgende Eigenschaften:
  + Titel
  + Kurz-Beschreibung
  + ext
  + Vorschau Bild
  + Antrieb (Ohne, Benzin, Elektro, Diesel)
  + Leistung
  + Preis
  + Baujahr
  + Status (Neu, Demo, Occasion)
  + weitere Bilder für eine Galerie auf der Detailseite
  + Angehängte Dateien (PDF & Word, zusätzliche Bilder)
* Kategorien:
  + Hauptkategorie (z. B. Marke: Hitachi, Yanmar, CAT)
  + Nebenkategorie (z. B. Art: Minibagger, Raupenbagger, Grossbagger, Mobilbagger, Radlader)
* auch Kategorien können durch die Redaktoren übersetzt werden

## Teil 2:

* Listenansicht mit allen Produkten
* Listenansicht ist wie folgt Dynamisch konfigurierbar:
  + Sortierung: Alphabetisch oder Reihenfolge des Backends übernehmen
  + Anzeigen von Produkten der gewählten Kategorie(n)
  + Anzeigen von Produkten aus gewählten Ordner(n)
  + Anzeigen von einzeln ausgewählten Produkten
  + Auswahl der Kategorie(n) für die Filterung
  + Filter wird nur angezeigt, wenn Kategorien im Plugin gewählt wurden
* Merkliste
  + Produkte können einer Merkliste hinzugefügt werden
  + In der Merkliste sollen die Eigenschaften der Produkte als Vergleichstabelle angezeigt werden.
* Detailansicht eines Produkts bei Klick
* Die Ansicht im Frontend wird getreu einem Design Prototypen umgesetzt
* Das Frontend ist responsiv und auf Mobilen Geräten bedienbar.
* Produktdetailseiten haben eine «sprechende» URL mit dem Titel des Produkts
* Lightbox für die Bilder in der Detailansicht
* Die Listenansicht muss anhand vordefinierter Kategorien wie folgt filterbar sein:
  + eine/mehrere Marken wählen = zeigt alle Produkte, die Marke X ODER Marke Y enthalten
  + eine/mehrere Arten wählen = zeigt alle Produkte, die Art X ODER Art Y enthalten
* Ändern der Sprache im Frontend zeigt Produkte in der gewählten Sprache
* Weiteres:
  + Erstellen Backend User mit Berechtigungen zur Erfassung von Produkten
  + Benutzeranleitung zur Erfassung der Produkte im Backend
  + Testing
  + Erfassung von Produkten von 3. Person
  + «Benutzen» des Frontends durch 3. Person
* Kein weiteres Testing vorgesehen
* Nicht Bestandteil dieser Arbeit sind:
  + Installation & Grundkonfiguration TYPO3 - Bereitstellen des Servers
  + Abfüllen der Produkte - Einführung der Erweiterung
  + Allgemeines Websitelayout (Navigation, Footer)

## Mittel und Methoden

Hermes 5.1 als Projektmethode (vereinfacht) (Initialisierung, Konzept, Realisation)

Windows Laptop

VS Code als Editor

Stage-Umgebung auf Server mit SSH Zugang

TYPO3 CMS

[TypoScript](#_Glossar)

PHP

MySQL

HTML

JavaScript

(S)CSS

TYPO3 Extension Builder

Extbase Framework

TYPO3 Fluid

Terminal8 Frontend Framework

Coding Richtlinien: https://docs.typo3.org/m/typo3/reference-coreapi/master/en-us/CodingGuidelines/Index.html

## Vorkenntnisse

-TYPO3

-TypoScript

-PHP, SQL

-JavaScript, HTML, (S)CSS

-TYPO3 Fluid

-Erweitern und bearbeiten von bestehenden und neuen Websites

-Erweitern und bearbeiten von bestehen TYPO3 Extensions

## Vorarbeiten

Vorbereitung Dokumentation Installation TYPO3 Grundkonfiguration TYPO3, TypoScript (SCSS Compiling) Basislayout der Seite (Navigation, Footer)

Neue Lerninhalte

Erstellen TYPO3 Extension von Grund auf mit Extension Builder Filterung von Produkten Merkliste

Arbeiten in den letzten 6 Monaten

Erstellen von verschiedenen Kundenwebsites in verschiedenen Teams Probe-IPA Verwendung und Erweiterung bestehender TYPO3 Extensions (Templating, zusätzliche Felder/Einstellungsmöglichkeiten) Einsatz aller o.g. Tools

# Organisation der IPA

Wie wird die Dokumentablage gemacht? Ein Printscreen oder ein Schema.

Arbeitsplatz wo? wie eingerichtet? Fehlte etwas? (Ev. ein Bild vom Arbeitsplatz machen)

Strukturablage Ordner und Versionisierung wichtig! Wird ein Tool wie beispielsweise SVN oder GiT verwendet muss ein Printscreen gemäss Kriterium erstellt werden.

*Siehe im Kriterienkatalog Teil A ☺*

## Datensicherung der IPA

Die Daten, welche für die IPA von Nöten sind, werden einerseits auf einem Server von Terminal8 gespeichert, der via SFTP mit dem Entwicklungs-Endgerät verbunden ist. Andererseits werden alle Projektdateien auch auf eine GitHub-Dateiablage platziert. Zudem wird der gesamte Projektordner am Ende jedes Tages auf Google Drive abgelegt, als Alternative zum herkömmlichen USB-Stick.

Es wird alle Zwei Stunden der neueste Stand der Dokumentation und der Software selbst auf GitHub hochgeladen, um einen plötzlichen Verlust der Daten um jeden Preis zu vermeiden.

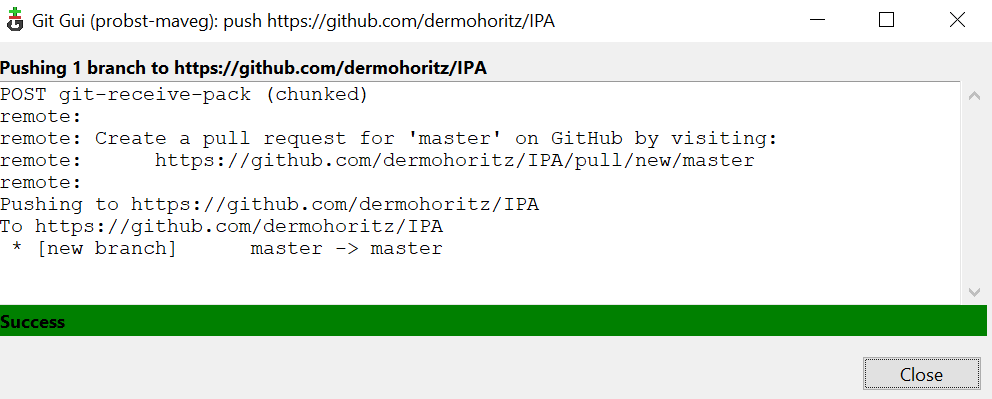


Abbildung 3, Git Push

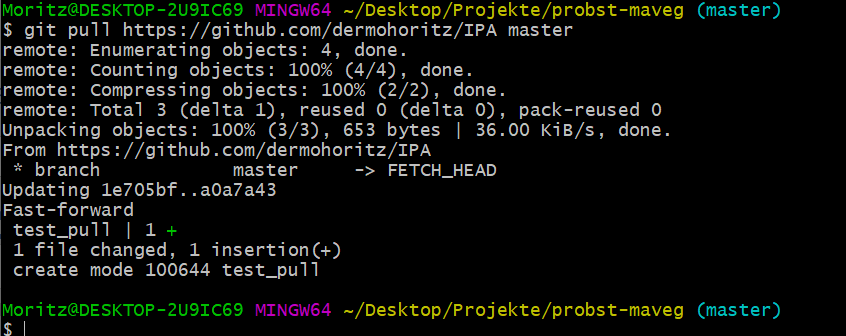


Abbildung 4, Git Pull

## Versionierung

Die erledigten Arbeiten werden versioniert und abgespeichert. Hier unterscheidet man zum Einen zwischen der Version der Dokumentation und damit in Verbindung stehenden Dateien wie z.B Zeitplan und Bilder und zum Andern dem eigentlichen Softwareprodukt.

### Dokumentation

Die Dokumentation wird folgendermassen versioniert:

Pro Arbeitstag wird der Ordner mit allen Dokumenten (Zeitplan, Dokumentation, Ordner mit Bildern) kopiert, mit der neuen Versionsnummer abgespeichert und anschliessend auf GitHub geladen. Der Einfachheit halber habe ich die Version der Dokumentation nach Anzahl verstrichener Arbeitstage definiert. So entspricht z.B die Version des 5. Arbeitstages der Version «0.5». Erst bei Beendigung der IPA wird die erste Ziffer geändert, die fertige Version wäre dann «1.0».

Eine vollumfängliche Korrektur der gesamten Dokumentation erfolgt bereits im Verlauf der IPA. Die Korrekturen werden durch den Projektleiter basierend auf Hinweisen der Testleser durchgeführt. Wann genau die Überarbeitungen stattfinden ist im Zeitplan unter der Phase «Allgemein» ersichtlich.

### Software

Die Version der Software wird durch die Anzahl der Funktionen, die im Verlauf der IPA umgesetzt werden bestimmt. Die zweite Ziffer ändert sich, wenn entweder die Listenansicht, die Detailansicht oder die Merkliste vollumfänglich implementiert wurde, wohingegen die dritte Ziffer sich schon bei kleinen Anpassungen ändern kann. Beispielsweise, wenn das Fluid-Template der Listenansicht umgesetzt ist und ein weiteres Mal, wenn das [Styling](#_Glossar) dafür gemacht wurde. Welche Änderungen und Funktionen genau dazugekommen sind, kann man der GitHub-Dateiablage oder dem Arbeitsjournal entnehmen.

## Hilfestellungen

Im Rahmen der IPA wurden einige Informationsquellen mehrfach eingesetzt:

* Korrekturleser: Die Korrekturleser für Grammatik und Rechtschreibung lesen die Dokumentation in regelmässigen Abständen und geben eine Rückmeldung. Basierend auf diesen Rückmeldungen werden Anpassungen vorgenommen.
* Internet: Bei Fragen zu TypoScript/Fluid/AJAX/PHP oder anderen Problemen wurde im Internet nach Lösungen gesucht. Folgende Seiten wurden öfters verwendet:
  + Intranet Terminal8: <https://intra.terminal8.ch/>
  + TYPO3 Reference: <https://docs.typo3.org>
  + ExtBase-Buch: <https://www.extbase-book.org/>

Falls zusätzliche Hilfestellungen verwendet werden mussten, werden diese im Arbeitsjournal festgehalten.

Wie/Wen/Wo werden die Daten gesichert?

Achtung DROPBOX und andere Cloudsysteme, welche nicht in der Schweiz beheimaten sind, sollten aus Informations- und Datenschutzgründen keine Daten gesichert werden!

ISDS-Konzept!!!

Beschreiben, wie der Kandidat die IPA und deren erarbeitete Dokumente sichert.

Auch USB-Stick ist eine Sicherung!

# Planung

Hermes 5.1 IPA: Hier handelt es sich um eine vereinfachte Projektmethode von Hermes 5, die optimal angepasst wurde für die Durchführung von IPAs. Die wichtigsten Grundsätze von HERMES 5.1 wurden beachtet:

* Projektorganisation mit Rollen
* Phasen: Initialisierung, Konzept, Realisierung, Allgemeines
  1. Die Initialisierung hat aufgrund der bereits im Detail geklärten Aufgabenstellung bereits begonnen und wird nun weitergeführt.
  2. Die Einführung ist nicht Teil dieser IPA, Grund dafür ist, dass die Website erst fertig entwickelt sein muss, bevor sie eingeführt werden kann. Die Website wird nicht innerhalb des Zeitraums der IPA fertigentwickelt, sondern erst zu einen späteren Zeitpunkt.
  3. Die Phase «Allgemeines» wurde zusätzlich eingeführt, da die Dokumentation ebenfalls Teil des Projekts ist und während der ganzen Zeit geführt wird. Im Zeitplan werden nur die Zeit für Korrekturen und Verbesserungen von bereits erledigtem berücksichtigt.
* Meilensteine: Nach jeder Phase ist ein Meilenstein abgeschlossen.
* Szenario
* Module, die dem gewählten Szenario entsprechend

Folgende Dokumente zur Planung wurden erstellt:

* **Phasenplan:** Zeigt die Länge der einzelnen Phasen auf.
* **Zeitplan:** Stundengenaue Darstellung der detaillierten Planung der einzelnen Projektschritte. Damit die Projektschritte auch in den anderen Plänen identifiziert werden können, sind sie nummeriert.
* **Meilensteinplan:** Darstellung und Auflistung der Meilensteine mit SOLL- und IST-Datum; Bemerkungen zur Phasenfreigabe werden dort vermerkt.
* **Arbeitsjournal:** Im Arbeitsjournal werden die erledigten Tätigkeiten beschrieben. Sie werden mit den geplanten Tätigkeiten verglichen. Das Arbeitsjournal dient als Mittel zur Überprüfung der Differenzen zwischen erledigten und geplanten Tätigkeiten und deren Aufwand.

## Phasen

Hier wird der Phasenplan für die IPA abgebildet. Die Phase «Allgemeines» zieht sich über die ganze Länge, da die Dokumentation während dieser Phase, sprich, durch den gesamten Verlauf des Projekts, geführt wird.

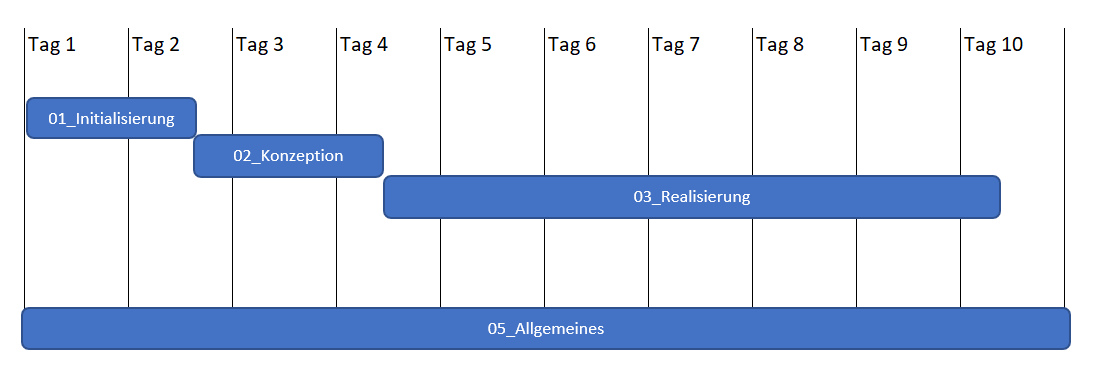


Abbildung 5, Phasenplanung

## Meilensteine

Nach jeder Phase befindet sich ein Meilenstein, eine sogenannte Phasenfreigabe. Bei der Realisierung gibt es zwei weitere Meilensteine, da diese Phase besonders lange dauert und weitere Meilensteine innerhalb dieser Phase Sinn ergeben.

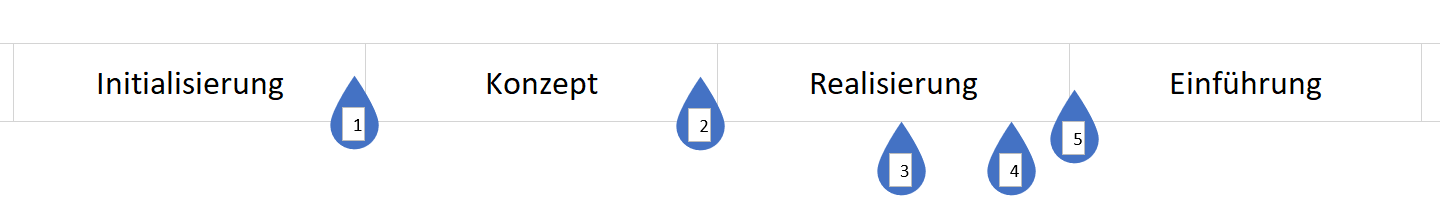


Abbildung 6, Meilensteinplanung

**MS1: Phasenfreigabe Konzept**

|  |  |
| --- | --- |
| Soll-Termin: | Donnerstag, 08.04.2021 um 18:00 Uhr |
| Ist-Termin: | Freitag, 09.04.2021 um 8:30 |
| Ergebnisse: | Der Auftraggeber gibt grünes Licht für die Weiterführung der Planung und ist mit dem bisherigen Ergebnis zufrieden. Die Initialisierung ist vollendet, alle organisatorischen Dinge sind geklärt und die eigentliche Aufgabe ist bis ins Detail definiert. Jetzt kann man mit der theoretischen Umsetzung beginnen. |
| Bemerkungen: |  |
|  |  |

**MS2: Phasenfreigabe Realisierung**

|  |  |
| --- | --- |
| Soll-Termin: | Dienstag, 13.04.2021 um 14:00 |
| Ist-Termin: | Dienstag, 13.04.2021 um 13:00 |
| Ergebnisse: | Alles, was man theoretisch definieren kann ist definiert, inklusive Diagrammen und jeglichen Testkonzepten. Der Auftraggeber ist mit dem Resultat zufrieden und gibt die nächste Phase frei. |
| Bemerkungen: |  |
|  |  |
|  |  |

**MS3: Alle Schlüsselfunktionalitäten implementiert**

|  |  |
| --- | --- |
| Soll-Termin: | Freitag, 16.04.2021 um 18:00 Uhr |
| Ist-Termin: |  |
| Ergebnisse: | Listview, Detailview + Merkliste sind implementiert und werden mit HTML/SCSS so dargestellt, dass es den Design-Vorgaben entspricht. |
| Bemerkungen: |  |
|  |  |

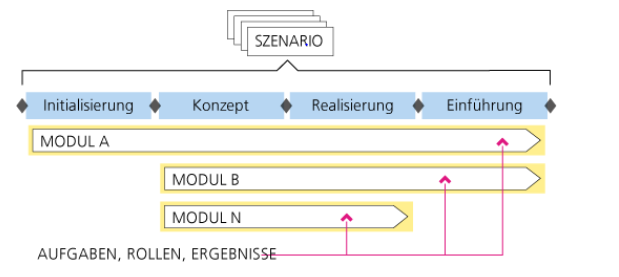
**MS4: Bugfixing und Testing beendet**

|  |  |
| --- | --- |
| Soll-Termin: | Dienstag, 20.04.2021 um 12:00 Uhr |
| Ist-Termin: |  |
| Ergebnisse: | Sämtliche Bugs sind nicht mehr vorhanden, die Applikation wurde entsprechend getestet und Verbesserungen wurden vorgenommen. |
| Bemerkungen: |  |
|  |  |

**MS5: Phasenfreigabe Einführung / Abschluss IPA**

|  |  |
| --- | --- |
| Soll-Termin: | Mittwoch, 21.04.2021 um 12:00 Uhr |
| Ist-Termin: |  |
| Ergebnisse: | Eine in ihrer Gesamtheit getestete Applikation ohne Bugs und eine fertig revidierte Dokumentation. |
| Bemerkungen: | Dieser Meilenstein stellt die definitive Beendigung der IPA dar. |
|  |  |

WICHTIG: Es muss die im „Detailbeschrieb“ beschriebene Projektmethode/vorgehen angewandt werden! Sie können frei wählen, welche Methode Sie benutzen, idealerweise ist die Methodik ihres Unternehmens zu wählen. Grundsätzlich muss die angewandte Projektvorgehensmethode den Richtlinien bzw. den Konventionen entsprechen (Projektmethode selber oder Firmenrichtlinien). Abweichungen müssen beschrieben bzw. Dokumentiert werden. Dies weil der Experte immer auf die Konventionen achtet.



Die Abbildung 2 ist nur aus Demo zwecken dargestellt. Diese muss mit einer angepassten Variante ersetzt werden

Siehe auch: http://www.hermes.admin.ch/ oder Hermes 5.1 IPA auf Pkorg.

**Szenario Wahl, Individuelles Szenario (z. B IT-Applikationserweiterung, JSF)**

In einer Organisation werden verschiedenartige Projekte durchgeführt. Die Projekte können sich bezüglich ihres Inhalts und der Komplexität stark unterscheiden. Um der Vielfalt der Projekte gerecht zu werden, bietet HERMES Szenarien an. Ein Szenario ist auf die Durchführung von Projekten mit einer spezifischen Charakteristik ausgerichtet. Das Szenario beinhaltet genau diejenigen Methodenelemente von HERMES, welche für das Projekt von Bedeutung sind. Dadurch ist HERMES rasch und einfach anwendbar.

## Szenario

Als Szenario wurde «IT-Individualanwendung» gewählt, da das Szenario «Dienstleistung/Produkt» beispielsweise kein «Testing»-Modul beinhaltet und somit für eine Informatik-IPA eher ungeeignet ist. Man könnte zwar eigene Module hinzufügen, jedoch erscheint dieses Szenario mit den vorgegebenen Modulen sinnvoller

## Module

Für das Szenario sind einige Module vorgesehen. Einige werden jedoch entfernt, da die entsprechenden Arbeiten schon abgeschlossen sind. Andere Gründe für einen Ausschluss werden in der untenstehenden Aufzählung genannt. Namentlich wurden folgende Module entfernt:

* Geschäftsorganisation
  + Das Modul Geschäftsorganisation hat insofern nichts mit dem eigentlichen Projekt zu tun, da es sich um einen externen Auftrag handelt.
* Projektgrundlagen
  + Da die Anforderungen an die Applikation bereits im Voraus definiert wurden, wird dieses Modul nicht berücksichtigt.
* Einführungsorganisation
  + Da die Website erst zu einem späteren Zeitpunkt fertiggestellt wird, bedarf es keiner Einführungsorganisation.
* IT-Migration
  + Es handelt sich bei diesem Projekt um ein komplett neues System, was eine Migration unmöglich macht. Deshalb wird dieses Modul nicht berücksichtigt.
* Beschaffung
  + Da für die Leistungen keine Ausschreibungen/Publikationen oder materielle Beschaffungen von Nöten sind, wird dieses Modul ebenfalls weggelassen.

Die verwendeten Module und deren Zweck werden untenstehend beschrieben.

Tabelle 1, Module

|  |  |  |  |
| --- | --- | --- | --- |
| Modul | Initialisierung | Konzept | Realisierung |
| Projektsteuerung | Initialisierung beauftragen und steuern | Projekt steuern | Projekt steuern |
| Projektführung | Initialisierung führen und kontrollieren | Projekt führen und kontrollieren | Projekt führen und kontrollieren |
| Testing |  | Testkonzept erarbeiten | Test durchführen |
| Datenschutz |  | ISDS-Konzept erarbeiten | ISDS-Konzept umsetzen |
| IT-System |  | Architekturkonzept erarbeiten | Architekturkonzept umsetzen |

Projektsteuerung

Projektführung

Initialisierung

Konzept

Realisierung

IT-System

Datenschutz

Testing

Abbildung 7, Modulplan nach Phasen

# Zeitplan

Planung und Umsetzung in Form von A3 Querseiten (entsprechend falten)

Individueller Zeitplan (nach Projektmethode). Gestaltung der Blöcke Gemäss Kriterium im Kriterienkatalog und Input Experte

# Firmenstandards

Es gelten keine besonderen Standards, da Terminal8 keine solchen definiert hat. Dokumentvorgaben gibt es ebenfalls keine.

Bezüglich der Struktur der Extension t8\_jetpack gibt es einige Richtlinien, diese betreffen aber dieses Projekt aufgrund der Eigenständigkeit nicht. t8\_jetpack stellt lediglich einige Funktionen zur Verfügung und übernimmt einen Teil der Grundkonfiguration für die Webseite, welche als Vorarbeit der IPA gemacht wurde. Die Extension t8\_Custom ist dafür gedacht, Mediendateien wie Bilder oder Videos zu speichern. Zusätzlich werden hier alle Dateien sonstiger Extensions ausgelagert, die auf dem TYPO3-System installiert sind. Das ist praktisch, wenn es ein [Major-Update](#_Glossar) geben sollte und die Extensions allesamt neu installiert werden müssen. Man muss dann nur noch darauf achten, dass die alte t8\_Custom-Version auf dem neuen System installiert ist. Die Änderungen an anderen Extensions werden so automatisch übernommen und es wird eine Menge Arbeit und Zeit gespart.

Zu beachten sind die Coding-Conventions von [ExtBase](#_Glossar). Hier wird beispielsweise die Ordnerstruktur, und welche Arten von Dateien dort abgespeichert werden dürfen, vorgegeben. Auch die Namenskonventionen von Datenbanktabellen, Klassen- und Variabelnamen sind hier definiert. Genaueres kann über diesen Link nachgeschlagen werden: <https://docs.typo3.org/m/typo3/book-extbasefluid/master/en-us/a-CodingGuidelines/Index.html>

Falls ein Dokument der Firmenstandards vorhanden ist, dieses unverändert in dem Anhang. des IPA Dokuments hinzufügen!

Diese ist vor allem für die Experten wichtig. Damit die IPA korrekt bewertet werden kann und es ggf. nicht zu Abzügen kommt.

# Arbeitsjournal

## Arbeitsjournal 07.04.2021

Tabelle 3, Arbeitsjournal 07.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| Bereitstellen der Dokumentvorlagen | Moritz Burn | 1 | 0.5 |
| Erstellen des Zeitplans | Moritz Burn | 2 | 1 |
| Projektvorgehen | Moritz Burn | 1 | 1.5 |
| Total: |  | 4 | 4 |
| Tages Ablauf |  |  |  |
| Für den heutigen Nachmittag habe ich mir vorgenommen, nochmals alle Dokumentvorlagen zu prüfen und gegebenenfalls anzupassen. Anschließend erstelle ich den Zeitplan und definiere einen Teil des Projektvorgehens und die allgemeinen Bestimmungen in der Dokumentation. (Kurzfassung, Kapitel 1, 2 + 3) Zudem habe ich ganz am Anfang GitHub installiert, ein Repository erstellt und mit der Lokalen Entwicklungsumgebung verknüpft. | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
| Hilfestellungen bekam ich heute von der offiziellen HERMES-Website. In Grunde ging es darum, welches Szenario und welche Module ich in diesem Projekt verwende, respektiv ausschließe. | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  Heute kam ich einigermassen gut voran mit der Dokumentation. Ich habe mein Ziel von heute mehrheitlich erreicht. Unter anderem habe ich ein paar neue Funktionen in Word gefunden, die mir das dokumentieren erleichtern. Unter anderem das Erstellen von Abbildungs- und Tabellenverzeichnissen.  **Was lief weniger gut:**  Anfangs hatte ich ein paar Probleme mit GitHub, da ich dort anscheinend einen Ordner abgelegt hatte, der Lokal nicht existierte. Da ich das Git-Repo aber nicht pullen konnte, habe ich einfachheitshalber ein neues Repository erstellt und die Sachen erneut verknüpft, was schliesslich funktionierte und mir die Fehlerbehebung ersparte.  **Meine Erkenntnis von heute:**  Meiner Meinung nach bin ich heute ein Wenig langsam vorangekommen, obwohl ich mein Ziel grundsätzlich erreicht habe. Ich liege im Zeitplan richtig und werde Morgen nochmals an der Dokumentation, respektive an der Initialisierung arbeiten. Womöglich kann ich auch schon mit der Konzeption beginnen. Was ich morgen besser machen werde ist die Zeiteinteilung, ich habe ein Wenig zu lange Mittag gemacht und musste darum entsprechend länger arbeiten am Abend. | | | |
| **Nächste Schritte** |  |  |  |
| Die nächsten Schritte sind, dass ich mit der Dokumentation weitermachen werde, genauer gesagt werde ich Morgen die Initialisierung abschliessen. Zusätzlich werde ich mir Gedanken zur Konzeption machen und voraussichtlich Übermorgen damit beginnen. Ich werde mich davor nochmals mit HERMES 5.1 befassen, um sicherzugehen, dass ich alles richtig verstanden habe. | | | |

## Arbeitsjournal 08.04.2021

Tabelle 4, Arbeitsjournal 08.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| Risikoanalyse erstellen | Moritz Burn | 2 | 2 |
| Projektvorgehen definieren | Moritz Burn | 1 | 2.5 |
| Zeitplan überarbeiten + Gesprächsprotokoll schreiben | Moritz Burn | 0 | 0.5 |
| Expertenbesuch | Iwan Kalbermatten  Nils Weibel  Fehmi Raqipi  Moritz Burn | 2 | 1 |
| Varianten erstellen | Moritz Burn | 2 | 2 |
| Variantenentscheid treffen | Moritz Burn | 1 | 1 |
| Total: |  | 8 | 9 |
| Tages Ablauf |  |  |  |
| Am Morgen habe ich den Rest des Projektvorgehens definiert, genauer gesagt die verschiedenen Module nochmals genauer unter die Lupe genommen und eine Phase namens „Allgemeines“ dem HERMES 5.1 hinzugefügt. Zudem habe ich die Risikoanalyse fertiggestellt. Nach dem Mittag kam dann das erste Expertengespräch, worauf ich einige Verbesserungen an der Zeitplanung und an der Dokumentation vorgenommen habe. Im Zeitplan waren die Meilensteine nicht benannt und die Phase „Allgemeines“ war noch nicht existent, weder im Zeitplan, noch in der Dokumentation. Im weiteren Verlauf des Nachmittags habe ich das Gesprächsprotokoll vom Expertengespräch verfasst und auf pkorg hochgeladen. Anschließend habe ich mir Gedanken zum ISDS-Konzept gemacht und im Internet nach einem Beispiel gesucht. | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
| Die heutigen Hilfestellungen war erneut die offizielle Website von HERMES 5.1, zusätzlich habe ich einen Kollegen nach der Bedeutung des Moduls „Beschaffung“ gefragt, da ich hier Verständnisprobleme hatte und mir nicht von Anfang an klar war, um was es sich hier eigentlich handelt. | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  Ich konnte viel vom Expertengespräch mitnehmen und entsprechend schnell die Verbesserungen bei Zeitplan und Dokumentation erledigen.  **Was lief weniger gut:**  Probleme gab es heute keine. Ich habe ständig das Gefühl, im Zeitplan hintendrein zu sein, was aber nicht stimmt. Beim Expertengespräch konnte ich die Fragen der Experten nur in bedingtem Umfang beantworten, obwohl ich eigentlich genau wusste, wovon ich rede. Zudem musste ich beim Zeitplan ein wenig aufholen und darum eine Überstunde machen.  **Meine Erkenntnis von heute:**  Beim nächsten Expertengespräch muss ich mich noch steigern und die Fragen ausführlicher beantworten. Wie gesagt verfüge ich über das nötige Wissen, jedoch war ich relativ nervös, was sich auf meine Performance ausgewirkt haben könnte. | | | |

## Arbeitsjournal 09.04.2021

Tabelle 5, Arbeitsjournal 09.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| Variantenentscheide nochmals durchgehen und Anpassungen vornehmen | Moritz Burn | 0 | 1 |
| Systemanforderungen definieren | Moritz Burn | 2 | 2.5 |
| Dokumentation überarbeiten | Moritz Burn | 2 | 2 |
| ISDS-Konzept erstellen | Moritz Burn | 1 | 0.5 |
| Architekturkonzept entwerfen | Moritz Burn | 2 | 2 |
| Rollenkonzept erstellen | Moritz Burn | 1 | 0 |
| Datenbankkonzept erstellen | Moritz Burn | 1 | 0 |
| Total: |  | 9 | 8 |
| Tages Ablauf |  |  |  |
| Am Anfang des Tages habe ich nochmals die Variantenentscheide durchgelesen, wobei mir ein paar Dinge wie Rechtschreibefehler und Inhaltliche Ungereimtheiten aufgefallen sind, die ich darauf korrigiert habe. Unter anderem habe ich mich nochmals mit dem AbstractWidgetViewhelper befasst und mich schlussendlich doch für den herkömmlichen Ajax-Refresh entschieden. Anlass dazu gab mir die Komplexität im Vergleich zum normalen AJAX. Da ich außerdem bereits mit normalen AJAX gearbeitet habe, erscheint mir diese Entscheidung sinnvoll. Im weiteren Verlauf des Tages fiel mir auf, dass ein ISDS-Konzept nicht unbedingt von Nöten ist, da TYPO3 selbst viele Sicherheitsrelevante Aspekte bereits abdeckt. Anschließend habe ich die Systemanforderungen (+ die nötigen Konfigurationen erläutert, die bereits bei der Vorarbeit gemacht wurden) und das Architekturkonzept definiert. Hierbei war mir vor Allem das MVC wichtig, da es viele Verständnisfragen klärt und eine Übersicht über das Gesamtsystem schafft. Bei den Systemanforderungen habe ich unter Anderem eine kleine Zusammenfassung bezüglich der TYPO3-Installation und der Webserverkonfiguration geschrieben. | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
| [AbstractWidgetViewhelper](https://api.typo3.org/9.5/class_t_y_p_o3_1_1_c_m_s_1_1_fluid_1_1_core_1_1_widget_1_1_abstract_widget_view_helper.html) | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  Ich bin heute gut voran gekommen und konnte vieles erledigen. Dazu muss ich aber sagen, dass es nicht sehr anspruchsvoll war und mir auch keine Mühe bereitet hat. Ich bin zufrieden.  **Was lief weniger gut:**  Schlechte Erfahrungen habe ich heute keine gemacht, da ich konzentriert arbeiten konnte. | | | |
| **Nächste Schritte** |  |  |  |
| Die nächsten Schritte sind einige Diagramme (Klassendiagramm, ERD, Activity-Diagramm) und die Use-Cases als auch das Testkonzept. | | | |

## Arbeitsjournal 12.04.2021

Tabelle 6, Arbeitsjournal 12.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| Klassendiagramm erstellen | Moritz Burn | 2 | 2 |
| Datenbankkonzept erstellen | Moritz Burn | 2 | 2 |
| Rollenkonzept erstellen | Moritz Burn | 0 | 1 |
| Dokumentation überarbeiten | Moritz Burn | 1 | 2 |
| Use-Cases erstellen | Moritz Burn | 3 | 2 |
| Total: |  | 9 | 9 |
| Tages Ablauf |  |  |  |
| Heute Morgen habe ich als erstes das Rollenkonzept erstellt, da ich dieses am vergangenen Freitag nicht mehr anfangen konnte. Anschließend konnte ich das ERD relativ schnell erstellen, da es eine nicht unbedingt komplizierte Datenstruktur braucht. Ich habe dazu [Visual-Paradigm](#_Glossar) verwendet, da ich mit diesem Tool bereits vertraut bin aus Zeiten der Gibb. Danach konnte ich meine Dokumentation nochmals auf Rechtschreibung überprüfen. Die Korrekturleserin war so freundlich, mir einige übers Wochenende Notizen zukommen zu lassen. Die Use-Cases waren Fleißarbeit aber keinesfalls schwierig zum Umsetzen, daher konnte heute mehr umsetzen, als ich eigentlich geplant hatte. Ich liege nun wieder gut im Zeitplan, was mich beruhigt und gleichermaßen für die kommenden Tage motiviert. | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
| Tool für die Erstellung vom ERD : <https://online.visual-paradigm.com>  Tool für die Erstellung des Klassendiagramms : <https://lucid.app/>  Hilfestellung für Klassendiagramm : <https://stackoverflow.com/questions/1874049/explanation-of-the-uml-arrows/23256583> | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  Die Aufgaben von heute waren, abgesehen vom Klassendiagramm, relativ einfach zu bewerkstelligen. Ich konnte die bei der Variantenwahl verlorene Zeit wieder gutmachen und liege nun wieder perfekt im Zeitplan.  **Was lief weniger gut:**  Beim Erstellen des Klassendiagramms war ich mir anfangs unsicher, was die Darstellung der Parent-Klassen angeht, die den eigentlichen Klassen übergestellt sind und die die Pfeile aussehen müssen(Stichwort «extends»). Mit einigen Internetsuchen konnte ich aber einige gute Beispiele in Form von Bildern und nebenstehenden Erklärungen finden. Diese sind in der Hilfestellung verlinkt.  **Meine Erkenntnis von heute**  Ich musste heute feststellen, dass ich bei der Erstellung von Klassendiagrammen noch nicht sattelfest bin, da ich es längere Zeit nicht mehr gemacht habe. Zukünftig sollte das aber wieder problemlos funktionieren, da ich ein gutes Mass an Repetition hinter mir habe. Glücklicherweise hat sich das nicht auf den Zeitplan ausgewirkt. | | | |
| **Nächste Schritte** |  |  |  |
| Ich muss morgen noch ein paar Ergänzungen bei den Use-Cases machen, das sollte aber nicht lange dauern. Im Anschluss erstelle ich noch ein Activity-Diagramm und das Testkonzept. Danach hoffe ich auf eine baldige Phasenfreigabe, die mir eventuell im Zeitplan noch ein Wenig Reserven bescheren könnte. | | | |

## Arbeitsjournal 13.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| Use-Cases erstellen | Moritz Burn | 1 | 1 |
| Activity-Diagramme erstellen | Moritz Burn | 2 | 2 |
| Testkonzept erstellen | Moritz Burn | 2 | 1 |
| Dokumentation überarbeiten | Moritz Burn | 2 | 3 |
| Projektsetup / Überprüfung | Moritz Burn | 1 | 1 |
| Total: |  | 8 | 8 |
| Tages Ablauf |  |  |  |
| Ich habe heute Morgen als erstes meine Use-Cases fertig gemacht, unter anderem hatte ich noch keine Grafik, also musste ich die noch erstellen. Ich habe dazu erneut das Tool [Visual Paradigm](#_Glossar) verwendet. Danach ging es direkt mit [Visual Paradigm](#_Glossar) weiter, nämlich habe ich die Activity-Diagramme erstellt. Da man hier Beispiele vom Tool selbst zur Verfügung gestellt bekommt, musste ich weder Googeln noch sonst jemanden fragen und konnte die Diagramme relativ rasch umsetzen. Ich war mir beim Frontend Diagramm teilweise unsicher, da man theoretisch unendlich viele Möglichkeiten hat, sich auf der Website zu bewegen. Darum habe ich ein paar Regeln dazu aufgestellt, welche in der Dokumentation oberhalb der Grafik erläutert werden. Anschließend wollte ich das Testkonzept entwerfen, mir fiel aber auf, dass in der detaillierten Aufgabenstellung „kein weiteres [Testing](#_Glossar) vorgesehen“ war. Es erschien mir deshalb als Sinnvoll, die sog. „Smoke“ Testmethode in Zusammenhang mit den Use-Cases zu verwenden, da die Funktionalitäten durch die Use-Cases auf Fehler geprüft werden können. Ziel eines „[Smoke-Test’s](#_Glossar)“ ist es, nur die Grundlegenden Funktionen zu testen, bevor man sich intensiver mit dem Testen einzelner Details auseinandersetzt. Da ich darum weniger Zeit für das Testkonzept brauchte, habe ich die Zeit genutzt, um meine Dokumentation Darstellungstechnisch zu verbessern. Am Schluss habe ich nochmals das ganze System auf dir korrekten Konfigurationen überprüft und kann nun mit der Umsetzung der eigentlichen Applikation beginnen. | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
| Infos gesammelt zu Smoke-Test : <https://www.guru99.com/smoke-testing.html>  Visual Paradigm : <https://online.visual-paradigm.com> | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  Ich konnte die verschiedenen Diagramme relativ rasch fertigstellen, was mir für den restlichen Tag ein Gefühl von Selbstsicherheit verlieh und mich motiviert hat. Dass ich aufgrund von Unaufmerksamkeit beim Durchlesen der detaillierten Aufgabenstellung zu viel Zeit im Zeitplan eingerechnet habe, werte ich als positiv, da ich die Zeit auch anders gut nutzen konnte.  **Was lief weniger gut:**  Schlechte Erfahrungen habe ich heute keine gemacht. Es verlief alles so, wie es sollte (abgesehen von der Fehleinschätzung im Zeitplan).  **Meine Erkenntnis von heute**  Dass ich in diesem Sinn kein richtiges «Testkonzept» entwerfen muss, hätte mir bereits am Anfang der IPA auffallen sollen. Ich konnte die Zeit aber für die Dokumentation nutzen, was sicher auch nicht geschadet hat. Die Anforderungen hätte ich aber trotzdem doppelt und dreifasch durchlesen müssen, damit solche Fehler nicht mehr passieren. | | | |
| **Nächste Schritte** |  |  |  |
| Als nächstes geht es an die Umsetzung der Extension. Ich bin motiviert und zufrieden mit den bisherigen Ergebnissen. Die Phasenfreigabe bekomme ich mit ziemlich grosser Wahrscheinlichkeit Morgen Vormittag. | | | |

## Arbeitsjournal 14.04.2021

Tabelle 7, Arbeitsjournal 14.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| Projektsetup | Moritz Burn | 1 | 0 |
| Datenbankmodell implementieren mit Extensionbuilder | Moritz Burn | 2 | 2 |
| Flexforms erstellen | Moritz Burn | 4 | 4 |
| Filterfunktion implementieren | Moritz Burn | 1 | 2 |
| Total: |  | 8 | 8 |
| Tages Ablauf |  |  |  |
| Da ich mit dem Projektsetup bereits gestern abschließen konnte, war es mir möglich, direkt mit der nächsten Aufgabe zu beginnen. Am Morgen habe ich damit begonnen, auf einer neuen Domain ein TYPO3 zu installieren, damit ich mit dem Extensionbuilder das Datenmodell erstellen kann. Grund für die Neuinstallation ist, dass der Extensionbuilder nur für die Version LTS 9.x.x verfügbar ist. Als ich damit fertig war, das ERD in die Realität umzusetzen, konnte ich die Extension herunterladen, die TYPO3-Version von 9.x.x auf 10.x.x ändern und auf dem „richtigen“ System installieren. Anschließend habe ich die Eingabemaske für das Listenplugin erstellt und geprüft, ob ich alle Daten, die dort eingegeben werden, an die View übergeben kann, was geklappt hat. Ich hatte ein paar Probleme bei den [Selectlists](#_Glossar), da ich die Werte (z.B „Diesel“) zuerst nicht mit Hilfe der Files Danach musste ich die Filterfunktion einbauen, wozu eine Klassenerweiterung der Klassen „Category“ nötig war, damit ich eigene Funktionen schreiben kann. Dabei habe ich die Klassen des „Products“ als Referenz genommen und entsprechend angepasst. Probleme gab es dabei keine, jedoch musste ich ein paar Dinge anders machen, da die Kategorien bereits im TYPO3-Core enthalten sind und die Tabelle beispielsweise nicht „tx\_..“ sondern „sys\_category“ heisst, es wird eine andere Klasse namens „Category“ erweitert und nicht „AbstractEntity“ etc. Ich habe eine guten Anleitung dazu gefunden, da ich die Klassennamen zum Erweitern nicht auswendig wusste und auch nicht lange suchen wollte. Die eigentliche Filterfunktion funktioniert momentan noch nicht, jedoch habe ich in der View alle Daten zur Verfügung, um Morgen die Logik für den SQL-Query, respektive das Repository einzubauen. | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
| Anleitung zur Erweiterung des CategoryRepository : <https://linawolf.de/typo3-blog/extensions/sys_category-system-kategorien-fuer-eigene-typo3-extension-nutzen-teil-1/> | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  Heute hatte ich keine Probleme und alles lief genau nach Plan. Ich war erleichtert, als ich es Zustande brachte, die ausgewählten Kategorien aus dem HTML-Form zu extrahieren und anschließend in der View auszugeben.  **Was lief weniger gut:**  Probleme hatte ich grundsätzlich keine, jedoch konnte ich heute nicht an der Dokumentation arbeiten, da ich mit der Realisierung beschäftigt war. Das muss ich später noch nachholen, was mich eventuell im Zeitplan zurückwerfen könnte.  **Meine Erkenntnis von heute**  Ich habe heute gelernt, dass ich in gewissem Sinne «vergessen» habe, die Dokumentation zu schreiben, da ich mich in das Programmieren vertieft habe. Zumindest habe ich morgen noch genügend Zeit, um das nachzuholen. | | | |
| **Nächste Schritte** |  |  |  |
| Logik für Kategorienfilterung, danach muss ich die Merkliste implementieren. Am wichtigsten ist morgen aber, dass ich die Dokumentation vom Vortag nachholen kann und im Zeitplan nicht zurückfalle. | | | |

## Arbeitsjournal 15.04.2021

Tabelle 8, Arbeitsjournal 15.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
|  | Moritz Burn |  |  |
|  |  |  |  |
|  |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| . | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
|  | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  **Was lief weniger gut:**  **Meine Erkenntnis von heute** | | | |
| **Nächste Schritte** |  |  |  |
|  | | | |

## Arbeitsjournal 16.04.2021

Tabelle 9, Arbeitsjournal 16.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
|  | Moritz Burn |  |  |
|  |  |  |  |
|  |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| . | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
|  | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  **Was lief weniger gut:**  **Meine Erkenntnis von heute** | | | |
| **Nächste Schritte** |  |  |  |
|  | | | |

## Arbeitsjournal 19.04.2021

Tabelle 10, Arbeitsjournal 19.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
|  | Moritz Burn |  |  |
|  |  |  |  |
|  |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| . | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
|  | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  **Was lief weniger gut:**  **Meine Erkenntnis von heute** | | | |
| **Nächste Schritte** |  |  |  |
|  | | | |

## Arbeitsjournal 20.04.2021

Tabelle 11, Arbeitsjournal 20.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
|  | Moritz Burn |  |  |
|  |  |  |  |
|  |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| . | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
|  | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  **Was lief weniger gut:**  **Meine Erkenntnis von heute** | | | |
| **Nächste Schritte** |  |  |  |
|  | | | |

## Arbeitsjournal 21.04.2021

Tabelle 12, Arbeitsjournal 21.04.2021

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
|  | Moritz Burn |  |  |
|  |  |  |  |
|  |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| . | | | |
| Beschreibung was gemacht wurde, Erfolgen und Misserfolgen (Problemen) was ist alles passiert, wie war das Vorankommen? | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht (auch Internet gehört dazu.  Z.B Msdn für Hilfestellung zu c#)? | | | |
|  | | | |
| Reflexion |  |  |  |
| **Was lief gut:**  **Was lief weniger gut:**  **Meine Erkenntnis von heute** | | | |
| **Nächste Schritte** |  |  |  |
|  | | | |

Das Arbeitsjournal ist den Experten beim Besuch vorzulegen. Es ist sinnvoll, korrekt (mit den echten Zeitangaben gemäss Zeitplan (Soll)) jeden Tag im Stundenraster zu führen.

Arbeitsprotokoll (Journal)

Nachvollziehbare tägliche Notizen (mit Datum) zu den ausgeführten Arbeiten

Erreichte Ziele / Anforderungen

Aufgetretene Probleme

Erfolgreiche oder erfolglose Tests

Beanspruchte Hilfestellung (wer, was)

Nacht- und Wochenendarbeit

Vergleich mit Zeitplan

Kritische Hinterfragung des Tages

Kritischer Rückblick (Reflexion) was war gut? was habe ich gelernt? was könnte ich besser machen? auf was bin ich stolz? Erster Tag (oder halber Tag): Wochentag, xx.yy.20xx

Diese Tabelle für jeden (halb-)Tag benutzen. (Tabelle Arbeitsjournal muss nicht in Tabellenverzeichnis, da dies bereits im IVZ gelistet ist)

Inhalt muss dem Tabelleninhalt entsprechen, designe darf frei gewählt werden.

Achtung auch Firmenstandart beachten!

Weitere Arbeitstage wie Wochenendarbeit hier auch vermerken.

Wie empfandest du die Arbeit, welche Erkenntnisse würdest du bei der nächsten Arbeit einfliessen lassen etc. (kritische Würdigung der Arbeit)

# Unterschriften und Abnahmeprotokoll

Die lernende Person bestätigt mit ihrer Unterschrift diese IPA aus Eigenleistung erbracht und nach den Vorgaben der Prüfungskommission Informatik Kanton Bern erstellt zu haben. Die Angaben im Arbeitsjournal entsprechen dem geleisteten Arbeitsaufwand.

Die Unterschriften für die Phasenfreigaben stammen von der verantwortlichen Fachkraft

Lernende und Fachvorgesetzte haben das Abnahmeprotokoll **vor der Abgabe zu unterzeichnen!!** und somit dessen Authentizität zu bezeugen. Diese Tabelle kann erweitert werden oder angepasst werden, es gilt nach jedem „Durchlauf/Phase“ muss eine Unterschrift gegeben werden. De**r Teil 1 ist zwingend und darf nicht gelöscht oder geändert werden!**

Tabelle 7, Unterschriften Phasenfreigabe

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Datum | Name / OE | Unterschrift |
| Teil1 | 08.04.2021 | Lernender |  |
| Teil 1 | 08.04.2021 | Verantwortliche Fachkraft |  |
| Initialisierung | 08.04.2021 | Lernender |  |
| Initialisierung | 09.04.2021 | Verantwortliche Fachkraft |  |
| Konzept |  | Lernender |  |
| Konzept |  | Verantwortliche Fachkraft |  |
| Realisation |  | Lernender |  |
| Realisation |  | Verantwortliche Fachkraft |  |
| Allgemeines |  | Lernender |  |
| Allgemeines |  | Verantwortliche Fachkraft |  |

# Initialisierung

Schwerpunkt:

Die Voranalyse ist ein Klärungsprozess, der mit vertretbarem Aufwand eine Entscheidung über die grundsätzliche Art der Systemrealisierung herbeiführt.

Erstellung und Beurteilung der Situationsanalyse sowie Überprüfung der Zielesetzungen, der Problemstellung und des Untersuchungsbereichs.

Erarbeitung von Lösungsvorschlägen (Varianten) und Abschätzung der Realisierbarkeit. Eine sinnvolle Risikoanalyse für das Projekt, welche Risiken eintreten könnten während -und nach dem Projekt

## Ist-Zustand

Die Probst-Maveg AG muss sich zur Zeit mit einer in die Jahre gekommenen Website herumschlagen, die für die Erfassung von Produkten einen hohen Zeitaufwand zur Folge hat. Im Moment werde die Produkte folgendermassen erfasst:

1. Ein neues Produkt soll in die Website aufgenommen werden, der Hersteller stellt die Daten zum jeweiligen Produkt zur Verfügung.
2. Nun muss der Mitarbeiter, der für die Informationen verantwortlich ist, das Wichtigste aus der Herstellerbeschreibung extrahieren und diese in einem separaten PDF-Dokument festhalten. Dieses Dokument wird anschliessend an die nächste Instanz, die IT-Spezialisten der jetzigen Website, weitergegeben.
3. Die IT-Spezialisten müssen nun die Daten aus dem PDF in die Website aufnehmen. Da es zur Zeit kein CMS gibt, müssen die Daten direkt ins HTML eingefügt werden.

Probst-Maveg AG möchte ihren Workflow verbessern, weshalb die Entwicklung einer neuen Website unumgänglich ist. Terminal8 hat hierfür den Gesamtauftrag, bestehend aus Design, Konzeption und Umsetzung erhalten.

Welches sind wichtige Bearbeitungsschritte

Persönliche Ziele sind gemeint, was für Ziele du dir stellst während der Arbeit

BSP: Ziel bis zum 31.5.2013 ist das Projekt abgeschlossen 🡪 Meilenstein Projektabschluss erreicht

Ziele definieren (SMART)

Achtung ZIELE müssen Lösungsneutral sein!

## Projektziele

Das Ziel ist es, die veraltete Erfassungsmethode den heutigen Bedürfnissen anzupassen, respektive diese neu zu gestalten. Das heisst: benutzerfreundliche und zielorientiertere Gestaltung der Oberfläche, Responsive, intuitive Nutzerführung, Übersichtlichkeit, Einheitlichkeit der Darstellung. Mit der Extension sollen Produkte vom Kunden selbst erfasst und verwaltet werden können. Es soll am Schluss ein Plugin geben, das man im TYPO3-Backend auf einer Seite einfügen kann. Man kann im Plugin verschiedene Einstellungen vornehmen. Zum einen soll es eine Einstellung geben, mit der bestimmt werden kann, ob das Plugin eine Listenansicht, eine Detailansicht oder

Diese Punkte werden, falls am Schluss des Projekts die Zeit dafür bleibt, anhand von Usability-Tests durch Dritt-Personen überprüft. Wichtig dabei ist vor allem, dass der Benutzer selbst nichts «kaputtmachen» kann mit seinen Eingaben, spricht, dass die Applikation solide funktioniert und nicht ungewollt abstürzen kann.

Projektziele und Anforderungen [(Punkt: 9.4)](#_Anforderungen) sollten bereits grob im Detailbeschrieb auf Pkorg (deine Aufgabenstellung) beschreiben sein. Diese sollte man überprüfen und nochmals genauer spezifizieren bzw. beschreiben, was damit erreicht werden soll.

Ziele definieren (SMART)

Achtung ZIELE müssen Lösungsneutral sein!

## Anforderungen

Siehe Projektziele.

Die Anforderungen können auch anders gelistet werden als funktional nicht funktional.

### Funktionale Anforderungen

* Backend
  + Produkte können erfasst / verwaltet werden.
  + Produkte können einer oder mehreren Kategorien zugewiesen werden
  + Produkte und Kategorien können in verschiedenen Sprachen erfasst werden.
  + Listenansicht ist konfigurierbar
    - Sortierung nach Alphabet oder Backend-Reihenfolge
    - Anzeigen von Produkten aus einem separatem Ordner
    - Anzeigen von einzelnen Produkten
    - Auswahl der Kategorien für die Filterung
* Frontend
  + Produkte können nach Kategorien gefiltert werden
  + Detailseiten haben eine «sprechende URL» mit dem Produktenamen.
  + Produkte können in verschiedenen Sprachen angezeigt werden.
  + Auf Detailseiten erscheint eine Lightbox, wenn ein Bild angeklickt wird.

Funktionale Anforderungen beschreiben gewünschte Funktionalitäten (was soll das System tun/können) eines Systems bzw. Produkts, dessen Daten oder Verhalten.

### Nicht funktionale Anforderungen

* Die Darstellung entspricht der Vorgabe
  + Listenansicht
  + Detailansicht
  + Merkliste
  + Schaltflächen (Filter, Buttons)
* Das Frontend ist responsiv und auf mobilen Geräten gut bedienbar.
  + Für die Ansicht auf mobilen Geräten bietet sich eine gewisse Freiheit, da hierfür keine Design-Vorgaben definiert wurden.

Nichtfunktionale Anforderungen sind Anforderungen, an die "Qualität" in welcher die geforderte Funktionalität zu erbringen ist.  
  
Qualität im vorgenannten Sinn meint beispielsweise

wie die Funktionalität ausgeführt werden soll (z.B. Reaktionszeit)

Bedingungen unter denen die Funktionalität ausgeführt wird (z.B. 7x24 Std.)

Oder einen schnell zu bedienen GUIs (z. B Software Ergonomie)

## Risikoanalyse

Welche Risiken ergeben sich, wenn das Projekt nicht realisiert wird? Was ist, wenn das Projekt scheitert? Welches sind die grössten Risiken bei diesem Projekt. (kann auch in den Anhang)

Tabelle 8, Risikoanalyse

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Risikobeschreibung** | **Auswirkung** | **Vor Massnahme** | | | | **Massnahmen/Erklärung** | **Nach Massnahme** | | | |
| **W** | **S** | **Risiko** | **Handlungsweise** | **W** | **S** | **Risiko** | **Handlungsweise** |
| **R1:** Ausfall Workstation | Arbeiten können für eine gewisse Zeit nicht weitergeführt werden. | 2 | 2 | Gering | Risikominderung | Anti-Virus Programm | 1 | 2 | Gering | Risikoakzeptanz |
| **R2:** Verspätete Abgabe der Dokumentation | Notenabzug von 0.5 | 4 | 2 | Mittel | Risikominderung | Zeitplanung im Auge behalten, genug Reserven einplanen | 2 | 2 | Gering | Risikoakzeptanz |
| **R3:** Ausfall von Server auf dem die Website gehostet wird | Keine weiteren Arbeiten möglich | 1 | 3 | Gering | Risikoakzeptanz | Keine, es existiert aber ein Backup des gesamten Servers auf einer externen Festplatte. Zudem gibt es ein Backup auf GitHub und Zwei andere Server, auf dem die Installation im Notfall laufen könnte. | 1 | 2 | Gering | Risikoakzeptanz |
| **R4:** Formale Fehler in der Dokumentation | Abzüge in der Bewertung von der Dokumentation. | 5 | 2 | Sehr wahrscheinlich | Risikominderung | Abgabe an Drittperson/en zum testlesen. | 3 | 2 | Vorstellbar | Risikoakzeptanz |
| **R5:** Krankheit | Ausfall von Projektleiter und Entwickler | 2 | 1 | Gering | Risikoakzeptanz | Viele Früchte essen und Sport treiben. | 2 | 1 | Gering | Risikoakzeptanz |
| **R6:** Zeitplan zu knapp | Überstunden müssen geleistet werden. | 3 | 2 | Eher Vorstellbar | Risikominderung | Genügend Reserven einplanen | 2 | 2 | Gering | Risikoakzeptanz |
| **R7:** Update von TYPO3-Core | Möglicherweise wird die Kompatibilität eingeschränkt. | 2 | 1 | Gering | Risikoakzeptanz | Wenn es sich um ein [Minor-Update](#_Glossar) handelt, hat es keinen Einfluss. Bei einer neuen [LTS-Version](#_Glossar) müssten gegebenenfalls Anpassungen gemacht werden. Da die nächste LTS-Version aber erst im Winter erscheint, gibt es keinen Grund zur Sorge. | 2 | 1 | Gering | Risikoakzeptanz |
| **Schadensausmaß:**  S1 = führt zu keiner Abwertung  S2 = geringe Abwertung bis 1.0 Notenpunkte  S3 = hohe Abwertung über 1,0 Notenpunkte  S4 = führt zu Nichtbestehen    **Eintrittswahrscheinlichkeit:**  W1 = unvorstellbar  W2 = unwahrscheinlich  W3 = eher vorstellbar  W4 = vorstellbar  W5= Eintreffen hoch | | | | | | | | | | |

Tabelle 9: Risikoanalyse Tabelle

Beispiel



## Risikograph

Abbildung 8, Risikoportfolio

### Kurze Stellungnahmen zu den Risiken

Die Risiken wurden, wo nötig und möglich, minimiert oder zumindest um ein Gewisses verkleinert. Die obenstehende Tabelle zeigt auf, um welchen Faktor sich die Risiken durch die getroffenen Massnahmen verkleinert haben. So ist beispielsweise das Risiko R4 um Zwei Positionen nach unten geschoben worden. Mit dem blauen Pfeilen wird die Verschiebung der Risiken erkenntlich gemacht.

## Schutzbedarfsanalyse

Eine Schutzbedarfsanalyse ist in diesem Fall nicht unbedingt notwendig, da die Produktedatenbank keine sensiblen Daten enthält. Ausserdem werden relevante Sicherheitsaspekte wie z.B das Verschlüsseln von Admin-Passworten bereits durch TYPO3 selbst abgedeckt und müssen nicht selbst implementiert werden.

Die Dokumentation ist geringfügig schutzbedürftig, sie wird darum nur Lokal und auf dem privaten GitHub-Repository abgespeichert. Andere Massnahmen wurden nicht getroffen.

Welche Gefährdung von Daten und Systemen sind gegebenenfalls vorhanden und wie können diese Gefährdungen bekämpft werden? Unterliegen die Daten dem Datenschutz?

(Dropbox, iCloud, MyCloud etc.). IPA Daten sollten nur „in house Server “ oder auf externe Festplatten gespeichert werden. (kann auch eine Anforderung in qualitativer und „Muss“ Anforderung sein).

## Varianten

Bei der Entwicklung der Produktdatenbank gibt es zwei Variantenentscheide, die im Folgenden Unterkapitel analysiert werden. Viele andere Aspekte werden gemäss TYPO3-Best-Practices abgedeckt und so gibt es dort gewisse vorgegebene Strukturen zu beachten, die weniger Spielraum von verschiedenen Lösungsmöglichkeiten zulassen (Stichwort „ExtBase“). Aus diesem Grund werden hier nur Teile der Applikation durch einen Variantenentscheid analysiert und ausgewertet.

### Filterfunktion

Der zentrale Faktor bei diesem Entscheid ist, dass die gesamte Seite beim Aktivieren eines Produkte-Filters jedes Mal neu geladen werden muss. Grund dafür ist, dass PHP zuerst im Hintergrund die Daten sammeln und der View übergeben muss.

Tabelle 10, Varianten Filterfunktion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Beschrebung | Vorteile | Nachteile |
| 1 | Herkömmlicher Page-Refresh: | Bei dieser Variante wird die gesamte Seite neu geladen, wenn ein Filter «aktiviert» wird. Je nach Internetverbindung kann das sehr mühsam sein. | Kein zusätzlicher Aufwand,  Weniger Fehlerquellen | Ganze Seite wird neu geladen,  Schlechte [User-Experience](#_Glossar) |
| 2 | [AJAX](#_Glossar) | Mit Ajax lässt sich der Inhalt, der von PHP übergeben wird, dynamisch in einen separaten HTML-Container abfüllen. Der Vorteil dieser Variante besteht darin, dass eben nur ein einzelner Container neu geladen werden muss. | Besseres [User-Exp](#_Glossar)erience  Zeitsparender  Übergang kann mit SCSS animiert werden | Ungefähr 1-2  Stunden Aufwand |
| 3 | [AbstractWidget-ViewHelper](#_Glossar) | Das ist die TYPO3-eigene Version die sog. [ajaxWidgets](#_Glossar) zulässt. Vom Prinzip her ist es das gleiche, wie ein AJAX-Refresh. Er bietet aber auch die Möglichkeit, diesen auszuschalten und auf herkömmliche Art zu laden. | Beide obenstehenden Varianten sind darin enthalten, es kann mit minimalem Aufwand «gewechselt» werden. | Grosser Zeitlicher Aufwand,  Kein signifikanter Mehrwert, da ein Page-Refresh nicht nötig ist, wenn man AJAX verwendet. |

### Nutzwertanalyse

Für den Variantenentscheid gibt es sowohl für die Filterfunktion als auch für die Merkliste eine entsprechende Nutzwertanalysen-Tabelle. Bezüglich der Bewertung, wenn sich etwas optimal eignet, gibt es ein Maximum von Vier Punkten. Wenn etwas völlig ungeeignet ist, gibt es Null Punkte. Die Gewichtungen wurden vom Auftraggeber definiert.

Je nach Nutzwertanalyse kann es Abweichungen bei den Kriterien geben. Grund dafür ist, dass nicht alle Kriterien zu allen Varianten passen, respektive dass diese dort Sinn ergeben.

Tabelle 11, Nutzwertanalyse Filterfunktion

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Kriterien | Gewicht | Page-Refresh | | AJAX | | AbstractWidget-ViewHelper | |
| Bewertung | Gesamt | Bewertung | Gesamt | Bewertung | Gesamt |
| Erfüllung der Anforderungen | 25% | 1 | 0.25 | 2 | 0.50 | 3 | 0.75 |
| Zeitlicher Aufwand | 30% | 3 | 0.90 | 2 | 0.60 | 1 | 0.30 |
| Produzierbarkeit | 20% | 4 | 0.80 | 3 | 0.60 | 2 | 0.40 |
| User-Exp. | 25% | 0 | 0.00 | 4 | 1.00 | 4 | 1.00 |
| Gesamt | 100% |  | **1.95** |  | **2.70** |  | **2.35** |

### Entscheid

Die Variante «AJAX» scheint für die Filterfunktion am besten geeignet, da es von der Implementation her relativ unkompliziert ist und das beste User-Experience bietet. Man könnte den Inhalt beispielsweise auch «einfliegen» lassen, da es sich um einen separaten HTML-Container handelt, der die Informationen hineingeladen bekommt. Dazu kommt die bereits gesammelte Erfahrung mit AJAX im Betrieb, sodass man nicht das Rad neu erfinden muss.

Der Page-Refresh kommt allein wegen dem User-Experience nicht in Frage, weil die Seite halt immer neu geladen werden muss und darum keine flüssige [Transition](#_Glossar) möglich ist. Der [AbstractWidgetViewHelper](#_Glossar) bietet insofern keinen Mehrwert, da der Aufwand grösser ist und ein herkömmliches AJAX die gleiche Funktion mit sich bringt. Der einzige Vorteil wäre ein Fallback, falls ein Benutzer JavaScript deaktiviert haben sollte. Davon wird aber nicht ausgegangen, da für Websites im «[Clear Web](#_Glossar)» keine Sicherheitsrisiken erwartet werden und es somit kein Grund zur Deaktivierung von JavaScript gibt.

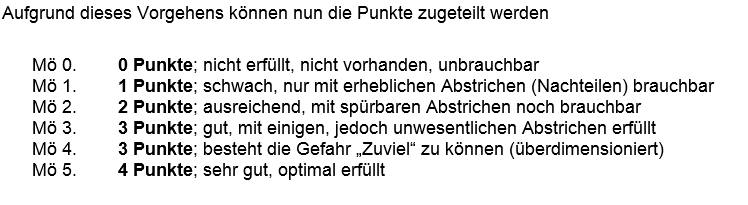
Beschreiben der Variante und ihre Vor-/ Nachteile aufzeigen.

Anforderungen gewichten (Präferenzmatrix) und in Kriterien wandeln.

Achtung: meistens sind qualitative Anforderungen Kriterien und K.O Kriterien sind meisten zwingende funktionale Anforderungen.

Idealerweise eine Bewertung von 0-3 vornehmen und diese kurz begründen.

Siehe BSP:



### Zwischenspeichern der Merkliste

Damit die Merkliste zwischengespeichert werden kann, müssen deren Produkte irgendwo abgespeichert werden. Hier stellt sich die Frage, ob diese nur temporär gespeichert werden oder ob man diese auch nach Beendigung der Browser-Session einsehen kann. Es bieten sich also folgende Möglichkeiten:

Tabelle 12, Varianten Merkliste

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Beschrebung | Vorteile | Nachteile |
| 1 | Login für Benutzer | Bei dieser Variante ist in jedem Fall ein Benutzer-Login erforderlich, die Daten werden in einer Datenbank gespeichert und werden einem Benutzer zugeordnet. | Produkte werden pro Benutzer in der DB gespeichert und gehen nicht verloren. | Viel zusätzlicher Aufwand, Benutzer müssen sich immer einloggen/registrieren, kann abschreckend wirken, Liste kann nicht direkt per Link geteilt werden. |
| 2 | Mit «normalem» [Cookie](#_Glossar) | Ein normales Cookie ist im Grunde genommen eine Textdatei, die auf dem Gerät des Endnutzers gespeichert wird, um das «Surf-erlebnis» zu verbessern. Bei sensiblen Daten besteht ein kleines Risiko, dass diese in falsche Hände geraten könnten, was in diesem Fall aber nicht relevant ist. Ein Cookie ist die standardmässige Variante zum Speichern von Einkaufswagen. | Keine Serverseitigen Daten, die gespeichert werden müssen, weniger Aufwand zum Entwickeln. Produkte sind auch nach dem Beenden einer Session noch gespeichert. | Sicherheitsrisiko, da die Daten eventuell über Jahre hinweg auf dem Gerät des Endnutzers lagern. In unserem Fall ist das aber nicht unbedingt relevant, da es sich nicht um Passwörter handelt. |
| 3 | Mit [Session-Cookie](#_Glossar) | Der Unterschied zum «normalen» Cookie besteht darin, dass Session-Cookies nach einer Session gelöscht werden. | Session-Cookies werden nach Beendigung der Session verworfen, die Wahrscheinlichkeit, dass Daten in die falschen Hände geraten ist wesentlich geringer. | Alles wird verworfen, wenn man den Browser schliesst, Liste kann nicht per Link geteilt werden. |

### Nutzwertanalyse

Tabelle 13, Nutzwertanalyse Merkliste

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Kriterien | Gewicht | Benutzer-Login | | Cookie | | Session-Cookie | |
| Bewertung | Gesamt | Bewertung | Gesamt | Bewertung | Gesamt |
| Erfüllung der Anforderungen | 25% | 3 | 0.75 | 3 | 0.75 | 3 | 0.75 |
| Termineinhaltung | 25% | 0 | 0.00 | 1 | 0.25 | 1 | 0.25 |
| Produzierbarkeit | 20% | 1 | 0.20 | 4 | 0.80 | 4 | 0.80 |
| User-Exp. | 25% | 2 | 0.50 | 4 | 1.00 | 2 | 0.50 |
| Security | 5% | 4 | 0.20 | 2 | 0.10 | 4 | 0.20 |
| Gesamt | 100% |  | **1.65** |  | **2.90** |  | **2.50** |

Tabelle 6: NWA, Nutzwertanalyse Merklistenspeicherung

Ggf. muss der Preis mit einberechnet werden!

Resultat der Gewichtung / durch Kosten + (ev. wiederkehrende Kosten z. B auf 5 Jahre gerechnet)

### Entscheid

Eine Entscheidung zu fällen war hier relativ einfach, da die meisten Websites Cookies für Ihre Warenkörbe verwenden. Das Benutzer-Login würde, zumindest für den vorgegebenen Zeitraum, den Rahmen sprengen. Ein Session-Cookie ist für diese Aufgabe eher ungeeignet, da man alle Daten verliert, sobald man den Browser schliesst. Also bleibt noch das «normale» Cookie, das grundsätzlich optimal zu dieser Aufgabe passt. Unter anderem auch darum, weil es in diesem Fall keine relevanten Sicherheitsrisiken gibt.

# Konzept

In der Konzeptphase werden die Anforderungen an die Applikation weiter konkretisiert und im Detail ausgearbeitet. Der Beginn dieser Phase wird mit Beendigung der Initialisierungsphase und des ersten Meilensteins markiert. Die Phasenfreigabe wurde erteilt.

## ISDS-Konzept

Das ISDS-Konzept ist für dieses Projekt überflüssig, da TYPO3, wie bereits in einem vorangehenden Kapitel erwähnt, die Sicherheit für Admin-Passwörter und eine Maßnahme gegen [SQL-Injections](#_Glossar) bereits mit sich bringt und diese nicht selber implementiert werden müssen. Zusätzlich handelt es sich hier um nicht sensible Daten, die von jedem beliebigen Website-Besucher eingesehen werden können, sprich öffentlich zugänglich sind.

## Systemumgebung, Anforderungen und Konfiguration

Einige Vorkehrungen wurden getroffen, die Vorarbeiten werden hier nochmals erläutert, um den Aufbau und die benötigten Komponenten/Einstellungen einer TYPO3-Installation und dessen Host klarzustellen.

### Vorarbeiten Webserver

Die Systemumgebung ist in diesem Fall ein Ubuntu-Server, auf dem die Websites gehostet werden. Dieser wird von [Plesk](#_Glossar) zur Verfügung gestellt. Es bedarf zuerst einer neuen Domain (<http://probst-maveg.ch.tajo.host.ch/>). Beim Erstellen der Domain muss ein FTP-Benutzer angelegt werden, damit man auf die Dateien des Webservers mit Anmeldedaten zugreifen kann. Dass TYPO3 funktioniert, muss zudem die PHP-Version stimmen, in diesem Fall ist es die Version 7.2, wobei auch andere Versionen kompatibel sind. Was ebenfalls benötigt wird ist eine Datenbank für das TYPO3-System, die anschließend beim effektiven Installieren als solche ausgewählt werden kann. Die Datenbank in Format [MariaDB](#_Glossar), einer Abspaltung von MySQL, wird ebenfalls auf dem Server erstellt.

### Vorarbeiten TYPO3

**Installation**

Jetzt kann man über die Entwicklungsumgebung, in diesem Fall [VS-Code](#_Glossar) und der Hilfe einer Erweiterung per FTP auf den Server zugreifen. Man starten ein FTP-Terminal und beginnt mit der Installation von TYPO3 mit der Version 9.5.26. Da es den ExtensionBuilder nur für die Version 9.x.x gibt, wurde diese Version auf einer separaten Domain installiert. Mehr dazu im Kapitel [Projektsetup](#_Projektsetup).

**Konfiguration**

Nachdem der sog. TYPO3-Core heruntergeladen und entpackt wurde, kann man im Browser unter der Adresse <http://probst-maveg.ch.tajo.host.ch/typo3> die Konfigurationen für die TYPO3-Installation vornehmen. Hier kann z.B die Datenbank ausgewählt werden, die vorhin erstellt wurde. Wenn man am Ende der Installation angelangt ist, sollte man folgendes auf dem Bildschirm sehen:

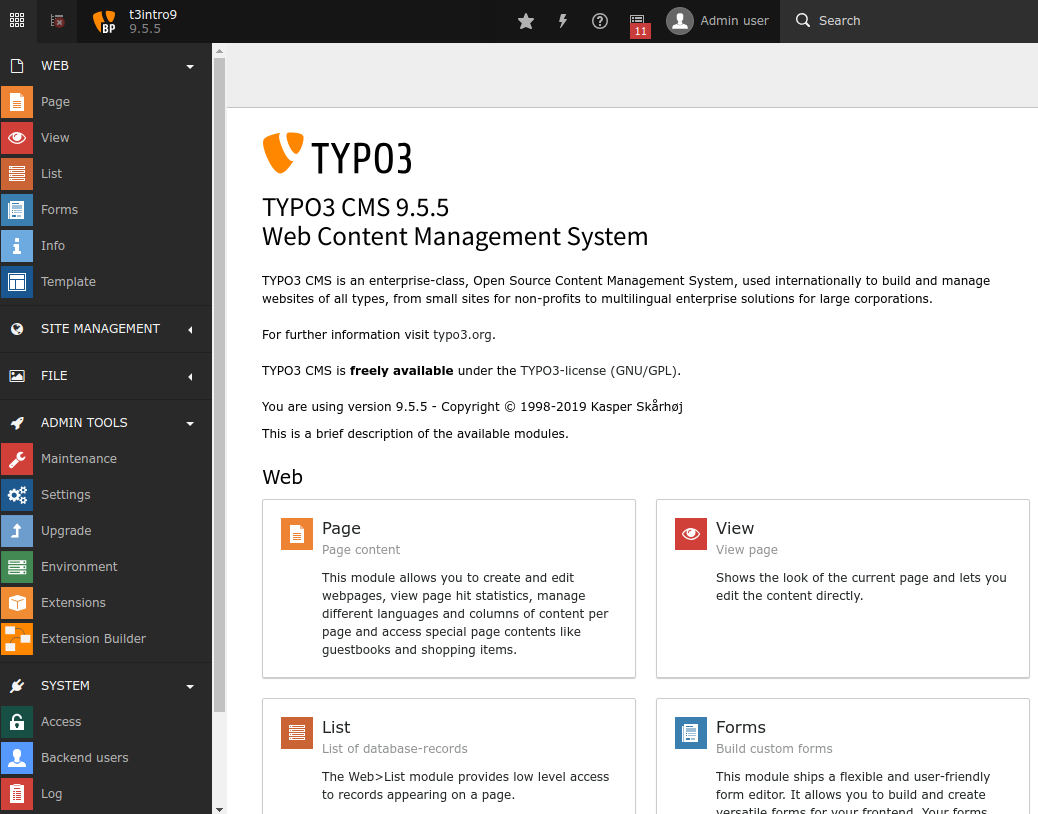


Abbildung 9, TYPO3-Backend

Die Betriebsinternen Erweiterungen haben eine gewisse Abhängigkeit zu anderen Erweiterungen, Um t8\_jetpack und t8\_custom installieren zu können, müssen zunächst folgende Extensions installiert werden:

Tabelle 14, Benötigte Extensions

|  |  |
| --- | --- |
| Name | Beschreibung |
| WS-SCSS | Ein SCSS-Compiler, der von t8\_custom benötigt wird. |
| Cookie Law Management | Wird benötigt, da t8\_jetpack Cookies verwendet. Die Benutzer müssen per Gesetz darüber informiert werden. |
| SMS-Responsive-Images | Stellt einen [Viewhelper](#_Glossar) zur Ausgabe von Bildern zur Verfügung. |
| CKEditor Fontawesome | Bewirkt, dass im [CKEditor](#_Glossar) ein Feld erscheint, mit dem [Font-Awesome-Icons](#_Glossar) in den Text eingefügt werden können. |

Jetzt können die beiden Extensions t8\_jetpack und t8\_custom installiert werden. T8\_jetpack erstellt eine simple Seitenstruktur und nimmt einige Konfigurationen vor.

Nun muss eine Seitenkonfiguration definiert werden, wo unter anderem die Sprachen für das Frontend ausgewählt werden können. Auch das [Errorhandling](#_Glossar) für einen 404-Error werden hier definiert. Am wichtigsten ist aber die sog. [BaseURL](#_Glossar), die bestimmt, unter welcher Adresse das Frontend aufgerufen werden kann. Diese muss einerseits in der Seitenkonfiguration und andererseits im t8\_custom selbst geändert werden. Die Installation und Konfiguration ist somit abgeschlossen.

## Architekturkonzept

Das Architekturkonzept beschreibt das Zusammenspiel zwischen den einzelnen Komponenten, in diesem Fall ist primär die Funktionsweise der MVC-Architektur gemeint, die von [ExtBase](#_Glossar) vorgeschrieben wird. Was ebenfalls von ExtBase vordefiniert wurde, ist die Ordnerstruktur und die Funktionen der darin enthaltenen Dateien, darum hat man hier nur sehr wenig Spielraum.

### Model – View – Controller

Die sogenannte MVC-Architektur ist weit verbreitet und bei vielen Systemen anzutreffen. Aufgrund seiner Einfachheit findet es auch bei komplexeren Softwarelösungen Verwendung. Grundsätzlich werden Systeme bei Anwendung des MVC’s in Drei Komponenten aufgeteilt, eine Datenbankkomponente, eine kontrollierende Komponente, die die logischen Entscheidungen basierend auf den Daten trifft, und schliesslich die Ausgabe dieser Daten.

Die folgende Tabelle definiert einzelnen Komponenten anhand einer kurzen Erklärung:

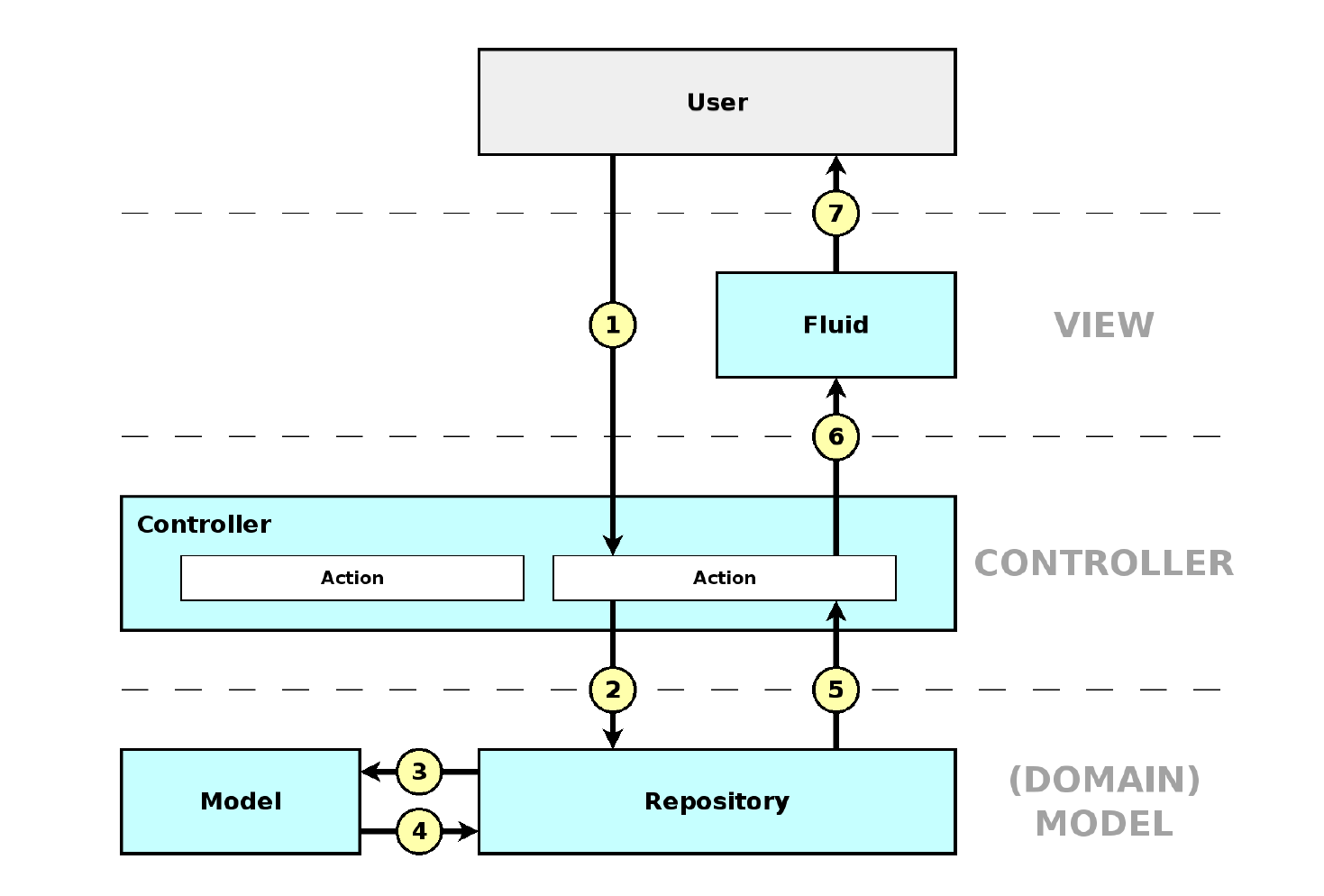
Tabelle 25, MVC-Erläuterung

|  |  |
| --- | --- |
| Name | Beschreibung und Funktionsweise |
| Model | Das Model besteht grundsätzlich nur aus einer Datenbank, die Daten speichert und diese für die Applikation zur Verfügung stellt. Im Fall von TYPO3 befindet sich der Code für das Model im Ordner «Classes/Model». |
| View | Die View ist für die Ausgabe zuständig, die sie vom Controller übergeben bekommt. Bei Web-Anwendungen ist das meistens ein HTML oder XML, in dem die Daten «abgelegt» werden. Bei TYPO3 gibt es sogenannte Fluid-Templates, die sich im Ordner «Resources/Private» finden lassen. Das spezielle an Fluid ist, dass die HTML-Templates gerendert werden müssen, da Fluid den Einsatz von Schleifen und Verzweigungen erlaubt. Die Ausgabe spezifischer Daten wie z.B einer bestimmten Anzahl Bilder wird also bei TYPO3 von der View selbst gesteuert. |
| Controller | Der Controller ist für das Beschaffen der Daten und deren Auswertung zuständig, bei TYPO3 sieht das aber ein Wenig spezieller aus: [ExtBase](#_Glossar) benutzt einen sogenannten «slim controller», was heisst, dass der Controller nur dafür da ist, die Daten aus dem Model zu holen und diese an die View weiterzugeben. Es wird keine «Business-Logic» ausgeführt. Der Controller von [Extbase](#_Glossar) befindet sich im Ordner «Classes/Controller». |

TYPO3-Extension bauen auf dem einfachen aber effektiven MVC auf. Der Ablauf, wenn ein Benutzer beispielsweise die Produkte filtern möchte, sieht folgendermassen aus:

1. Der Benutzer klickt auf einen Filter, ein sogenannter [Request](#_Glossar) wird an den Controller geschickt. Je nach Plugin wird eine andere Methode aufgerufen.
2. Der Controller ruft eine Methode des [Repository’s](#_Glossar) auf und gibt die Daten aus der View als Parameter weiter.
3. Das Repository formt anhand von Abfragen innerhalb dieser Methode einen SQL-Query, der im Anschluss an die Datenbank geschickt wird.
4. Die Datenbank findet die Daten, die dem SQL-Query entsprechen und übergibt diese als Array an das Repository. Ein sogenanntes «[queryResult](#_Glossar)».
5. Das Repository sendet die erhaltenen Daten weiter an den Controller.
6. Der Controller wertet die Daten aus und trifft die Entscheidung, welche Daten an die View übergeben werden sollen.
7. Die View, genauer gesagt das Fluid-Template trifft basierend auf diesen Daten die Entscheidung, was ausgegeben werden soll und was nicht. Zum Schluss wird das Fluid-Template «gerendert», soll heissen, die effektiven Daten werden in das HTML-Gerüst abgefüllt, anschliessend wird es dem Benutzer übergeben, der ursprünglich den Request ausgelöst hat.

Untenstehend wird der Ablauf eines solchen [Request’s](#_Glossar) zu Verständniszwecken nochmals grafisch abgebildet (Grafik von Extbase, verlinken)



### Listenansicht mit Filter

### Detailansicht

### Merkliste

## Rollenkonzept

Der Auftraggeber ist in diesem Fall ein Kunde, der seine Website erneuert haben möchte. Im Projektausschuss stehen sowohl die verantwortliche Fachkraft als auch die Haupt- und Nebenexperten.

Moritz Burn

Entwickler

Moritz Burn

Projektleiter

Iwan Kalbermatten

Nebenexperte

Fehmi Raqipi

Fachvogesetzter

Nils Weibel

Nebenexperte

Probst-Maveg AG

Auftraggeber

Steuerung

Projektaus-schuss

Führung

Ausführung

Abbildung 10, Projektrollen

### Projektrollen

Tabelle 2, Projektrollen

|  |  |
| --- | --- |
| Rollenbeschreibung der IPA |  |
| Auftraggeber: | PROBST MAVEG AG |
| Projektausschuss | Fehmi Raqipi,  Iwan Kalbermatten  Nils Weibel |
| Projektleiter: | Moritz Burn |
| Entwickler | Moritz Burn |

|  |  |
| --- | --- |
| **Auftraggeber:** | Ein Kunde, der den Auftrag für die Umsetzung der neuen Website gegeben hat. |
| **Projektausschuss:** | Personen, die die Projektdaten einsehen können und in Kontakt mit der Projektleitung stehen. |
| **Projektleiter:** | Führt das Projekt und überprüft, ob alle funktionalen Anforderungen erfüllt sind. Verantwortlich für den Reibungslosen Ablauf. |
| **Entwickler** | Setzt das geplante in die Realität um und entwickelt die Softwarelösung. |

## Datenbankkonzept (Nochmals prüfen)

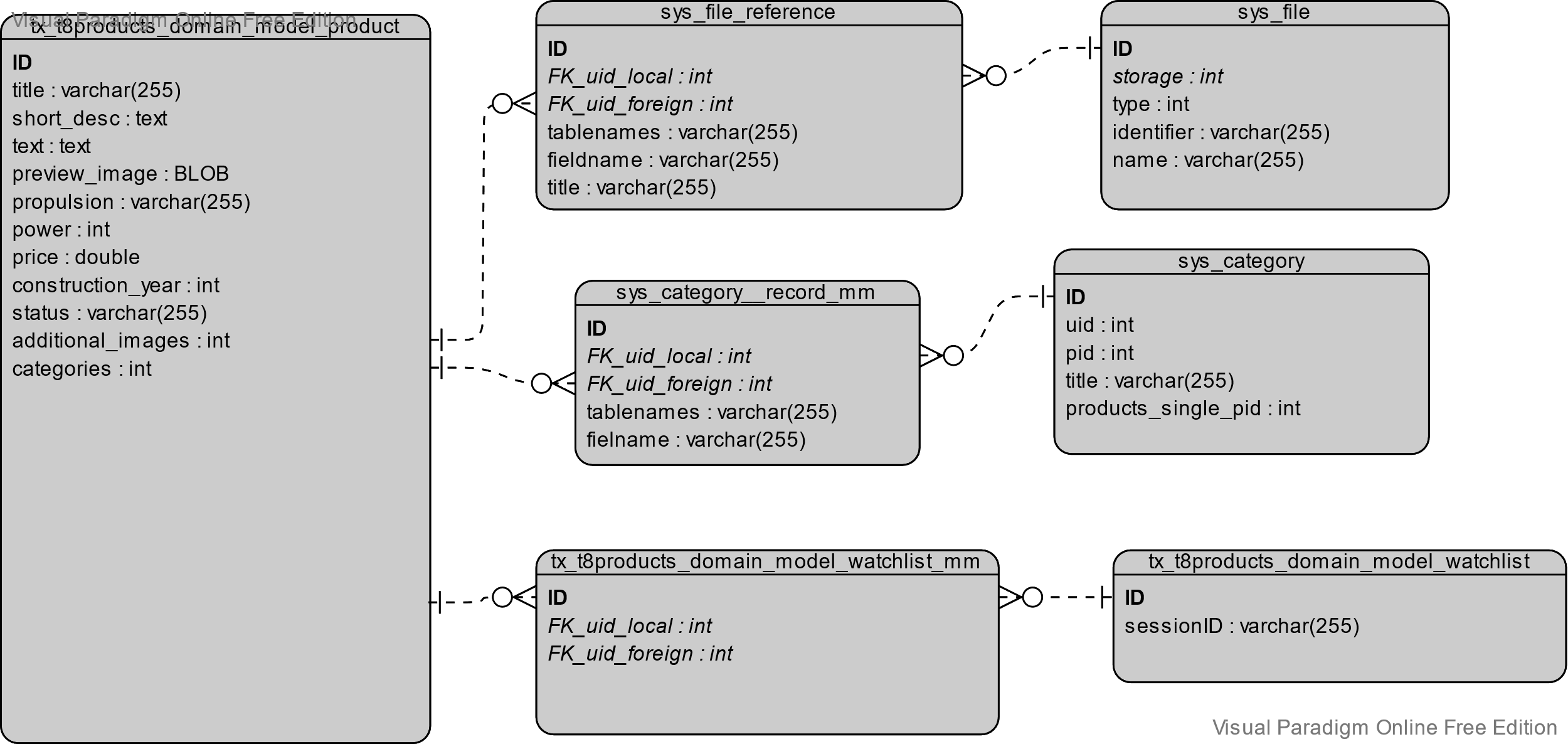
Untenstehend ist ein ERD-Diagramm zu sehen, das die Datenbankstruktur darstellen soll.

Abbildung 11, ERD

## Klassendiagramm (noch anpassen, oberstes Element falsch, Rest richtig=>Datentypen nochmals prüfen)

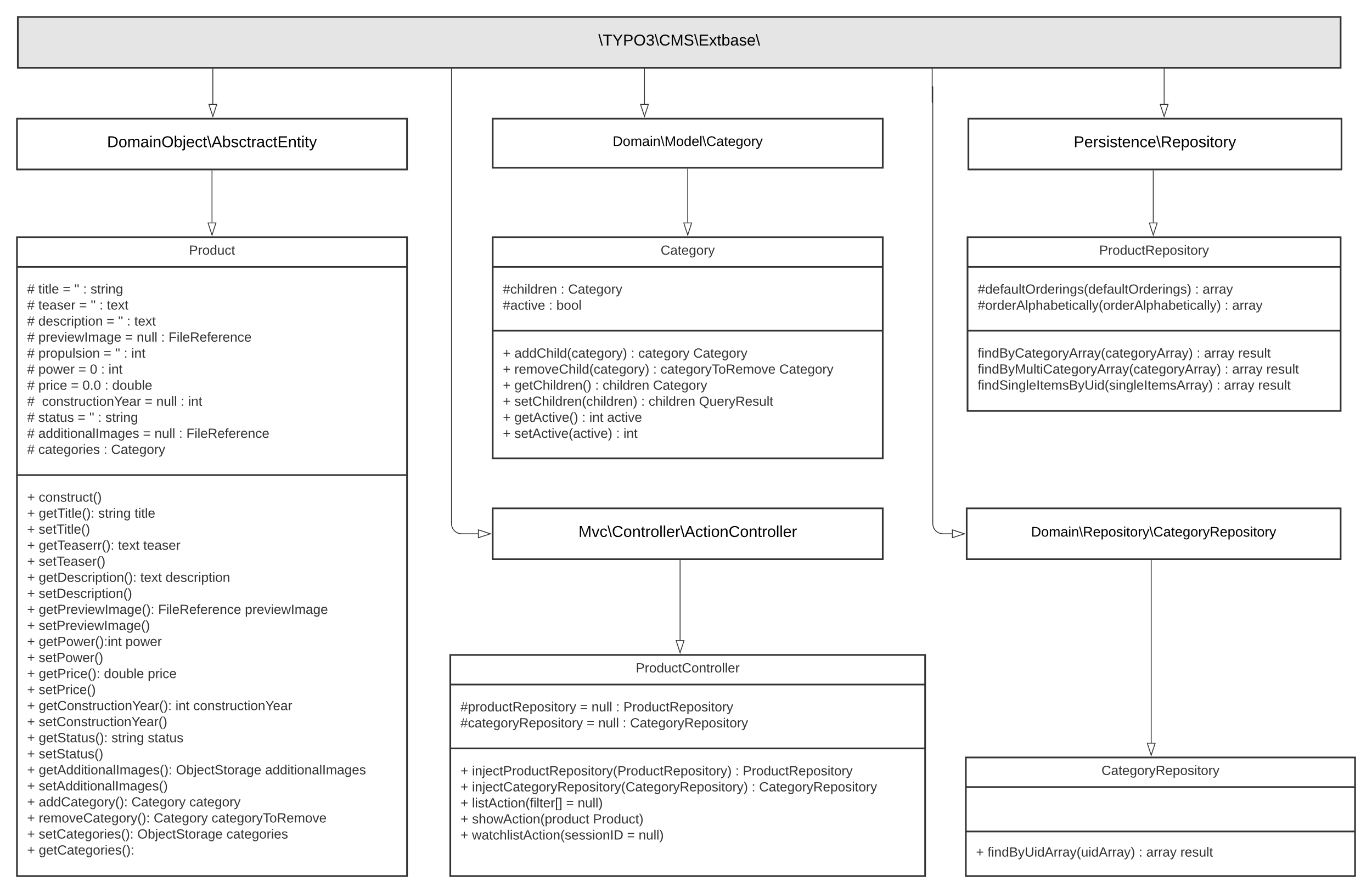


Abbildung 12, Klassendiagramm

Das Klassendiagramm stellt die Beziehungen zwischen den verschiedenen Komponenten der Applikation dar. Auf der linken Seite sieht man die Klasse für ein sog. [Domain-Object](#_Glossar), nämlich das Produkt mit seinen Eigenschaften. In dieser Klasse befinden sich die Getter und Setter. In der Mitte stehen die Klassen, die für die Controllerfunktionen zuständig sind, dazu gehören unter anderem die «listAction», die «showAction» und die «watchlistAction». Je nach Plugin wird eine dieser Drei Methoden aufgerufen und die Produkte werden aus der Datenbank, dem Repository geholt. Auf der rechten Seite sieht man schliesslich den Teil «Repository, es ist grob gesagt die Datenbankanbindung, mit dessen Hilfe die Produkte ausgelesen werden.

## User-Stories

Um die umgesetzten Funktionalitäten testen zu können werden sog. User-Stories erstellt. Untenstehend wurde ein Diagramm erstellt, das die verschiedenen User-Stories grafisch darstellt. Es wird zwischen Zwei Arten von Benutzern unterschieden, zum Einen dem Frontend-Benutzer, zum Anderen dem Redaktor/Admin-Benutzer, der die Produkte im Backend verwalten kann.

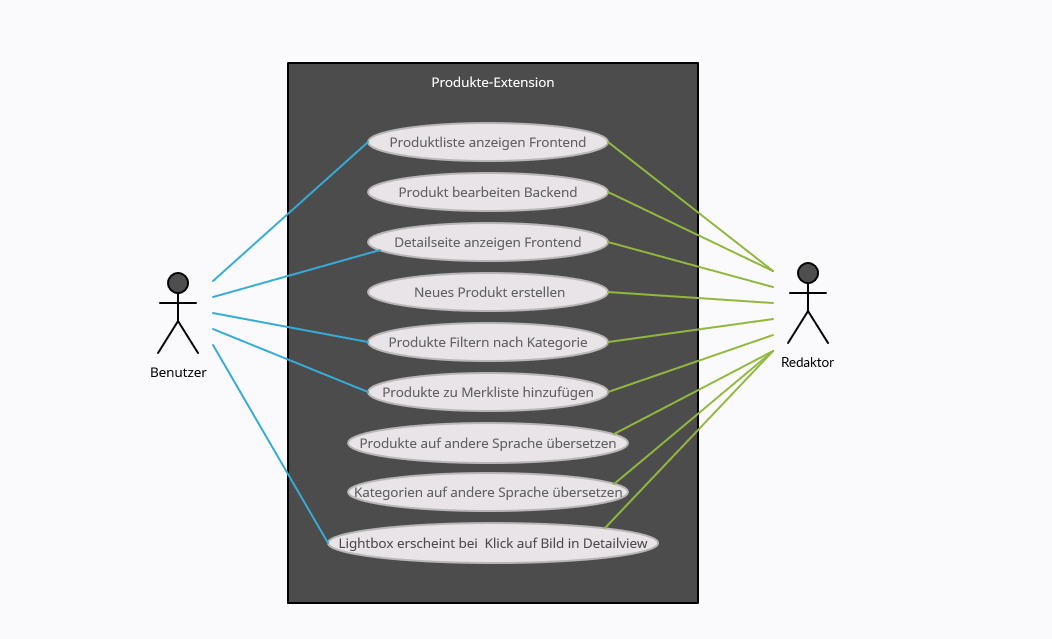


Abbildung 13, User-Story-Diagramm

Tabelle 15, User-Story Akteure

|  |  |
| --- | --- |
| Akteure | Beschreibung |
| Benutzer | Benutzt die Website als Besucher, er hat keinen Einfluss auf Daten, die angezeigt werden. |
| Redaktor | Kann im Backend Inhalte bearbeiten, hinzufügen und löschen. |

### Beschreibung User-Stories

Die User-Stories aus dem vorangehenden Kapitel werden hier genauer definiert und beschrieben. Änderungen oder zusätzliche User-Stories werden, falls vorhanden, mit einer anderen Farbe gekennzeichnet.

Tabelle 16, User-Story 01

|  |  |
| --- | --- |
| Use-Case: Produktliste anzeigen Frontend | Beschreibung |
| Use-Case Nummer | 01 |
| Kurze Beschreibung | Eine Liste mit allen Produkten wird ausgegeben. |
| Voraussetzungen | Plugin muss im Backend auf der Seite mit den entsprechenden Einstellungen vorhanden sein. |
| Akteure | Benutzer, Redaktor |
| Auslöser / Trigger | Ein Kunde möchte sich das Angebot anschauen oder der Redaktor überprüft seine Backend-Eingaben, mit denen er ein Produkt erfasst hat. |
| Standardablauf | Der Benutzer navigiert zur entsprechenden Seite und sieht eine Liste aller Produkte. |
| Erwartetes Ergebnis | Jedes einzelne Produkt soll angezeigt werden. |
| Bemerkungen | / |

Tabelle 17, User-Story 02

|  |  |
| --- | --- |
| Use-Case: Produkte bearbeiten Backend | Beschreibung |
| Use-Case Nummer | 02 |
| Kurze Beschreibung | Der Redaktor kann im Backend seine Produkte verwalten. |
| Voraussetzungen | Die Eingaben müssen in der Datenbank gespeichert und ausgelesen werden können. |
| Akteure | Redaktor |
| Auslöser / Trigger | Der Redaktor möchte ein bestehendes Produkt anpassen. |
| Standardablauf | Der Redaktor navigiert im Backend zur Listenansicht (Links) und wählt den entsprechenden «Produkte»-Ordner aus. Er wählt hier ein Produkt aus, welches er bearbeiten möchte. |
| Erwartetes Ergebnis | Die Eigenschaften des Produkts werden angepasst, wenn der Redaktor das Produkt speichert, den Cache löscht und das Frontend neu ladet. |
| Bemerkungen | / |

Testobjekt und Testfall

Tabelle 18, User-Story 03

|  |  |
| --- | --- |
| Use-Case: Detailseite anzeigen im Frontend | Beschreibung |
| Use-Case Nummer | 03 |
| Kurze Beschreibung | Der Benutzer kann im Frontend die Detailseite eines Produkts einsehen. |
| Voraussetzungen | Das Plugin muss auf einer entsprechenden **Unter**seite der Listenansicht eingefügt werden. Im Plugin für die Listenansicht muss die Unter-seite als Zielseite für die Detailview angegeben werden. |
| Akteure | Benutzer, Redaktor |
| Auslöser / Trigger | Ein Benutzer interessiert sich für ein Produkt und möchte mehr darüber erfahren. |
| Standardablauf | Der Benutzer wird mit einem Klick auf das entsprechende Produkt auf die Detailseite weitergeleitet. |
| Erwartetes Ergebnis | Alle Informationen zum Produkt werden auf der Detailseite korrekt angezeigt. |
| Bemerkungen | / |
|  |  |

Tabelle 19, User-Story 04

|  |  |
| --- | --- |
| Use-Case: Neues Produkt erstellen | Beschreibung |
| Use-Case Nummer | 04 |
| Kurze Beschreibung | Der Redaktor kann im Backend neue Produkte erfassen. |
| Voraussetzungen | Die Eingaben müssen in der Datenbank gespeichert werden können. |
| Akteure | Redaktor |
| Auslöser / Trigger | Der Redaktor möchte ein neues Produkt erfassen. |
| Standardablauf | Durch wechseln auf die Listenansicht im Backend (Links) und auswählen des «Produkte»-Ordners kann der Redaktor ein neues Produkt erstellen und speichern |
| Erwartetes Ergebnis | Das Produkt wird, nach speichern und Cache löschen, im Frontend angezeigt. |
| Bemerkungen | / |

Tabelle 20, User-Story 05

|  |  |
| --- | --- |
| Use-Case: Produkte filtern nach Kategorie | Beschreibung |
| Use-Case Nummer | 05 |
| Kurze Beschreibung | Im Frontend können die Produkte nach Kategorien gefiltert werden. |
| Voraussetzungen | Der AJAX-Filter muss korrekt implementiert sein und wird bei Erstellen eines Listen-Plugins angezeigt. |
| Akteure | Benutzer, Redaktor |
| Auslöser / Trigger | Der Benutzer sucht nach etwas spezifischem, beispielsweise einem bestimmten Fahrzeugtyp. |
| Standardablauf | Durch Klick auf das Dropdown-Menu erscheinen die verfügbaren Filter, diese können per Klick aktiviert/deaktiviert werden. Bei Klick auf «Suche» werden die Produkte gesucht. |
| Erwartetes Ergebnis | Die die dem Filter entsprechenden Produkte werden angezeigt. |
| Bemerkungen | / |

Tabelle 21, User-Story 06

|  |  |
| --- | --- |
| Use-Case: Produkte zu Merkliste hinzufügen | Beschreibung |
| Use-Case Nummer | 06 |
| Kurze Beschreibung | Jedes Produkt kann einzeln zu einer Merkliste hinzugefügt werden. |
| Voraussetzungen | Cookies müssen aktiviert und implementiert sein. |
| Akteure | Benutzer, Redaktor |
| Auslöser / Trigger | Ein Benutzer möchte, dass seine Produkte zwischen-gespeichert werden, damit er diese bei der nächsten Sitzung nicht nochmals suchen muss. |
| Standardablauf | In der Detailansicht in Frontend wird ein Produkt über den Button «Zur Merkliste hinzufügen» zur Merkliste hinzugefügt. |
| Erwartetes Ergebnis | Die Produkte, welche der Benutzer in die Merkliste einfügt, erscheinen, wenn man über einen Link die Merkliste einsehen möchte. |
| Bemerkungen | / |

Tabelle 22, User-Story 07

|  |  |
| --- | --- |
| Use-Case: Produkte auf eine andere Sprache übersetzen | Beschreibung |
| Use-Case Nummer | 07 |
| Kurze Beschreibung | Jedes Produkt kann einzeln in eine andere Sprache übersetzt werden. |
| Voraussetzungen | Die Sprache muss auf dem TYPO3-System installiert und aktiviert sein. |
| Akteure | Redaktor |
| Auslöser / Trigger | Eine neue Sprache wird auf dem System installiert, nun müssen auch die Produkte einzeln übersetzt werden. |
| Standardablauf | Man wechselt im Backend auf die Listenansicht (Links), wählt den «Produkte»-Ordner aus und sollte die Produkte mit Klick auf «Lokalisieren» übersetzen können. Das Produkt wird dupliziert und es öffnen sich die Eigenschaften des in der neuen Sprache erfassten Produkts. Diese können nun angepasst/übersetzt werden. |
| Erwartetes Ergebnis | Wenn man auf der Frontend-Seite die Sprache wechselt, sollten die entsprechenden Übersetzungen als Produkte-Eigenschaften angezeigt werden. |
| Bemerkungen | / |

Tabelle 23, User-Story 08

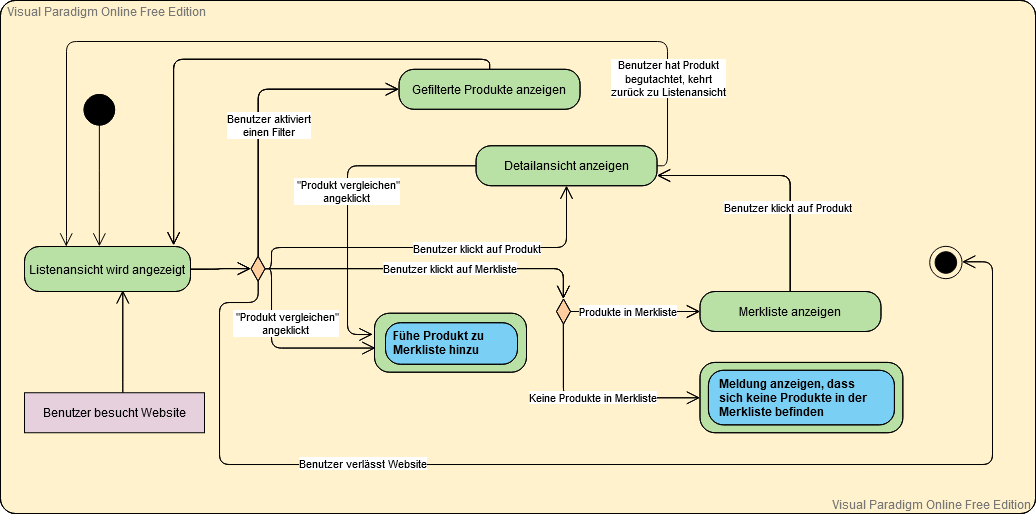
|  |  |
| --- | --- |
| Use-Case: Kategorien in eine andere Sprache übersetzen | Beschreibung |
| Use-Case Nummer | 08 |
| Kurze Beschreibung | Die Kategorien für den Filter sollen übersetzt werden. |
| Voraussetzungen | Die Sprache muss auf dem TYPO3-System installiert und aktiviert sein. |
| Akteure | Redaktor |
| Auslöser / Trigger | Eine neue Sprache wird auf dem System installiert, nun müssen auch die Kategorien einzeln übersetzt werden. |
| Standardablauf | Man wechselt im Backend auf die Listenansicht (Links), wählt den «Produkte»-Ordner aus und sollte die Kategorien mit Klick auf «Lokalisieren» übersetzen können. Die Kategorie wird dupliziert und kann übersetzt werden. |
| Erwartetes Ergebnis | Wenn man auf der Frontend-Seite die Sprache wechselt, sollten die Filteroptionen in der richtigen Sprache erscheinen. |
| Bemerkungen | / |

Tabelle 24, User-Story 09

|  |  |
| --- | --- |
| Use-Case: Lightbox bei Bilder in Detailview | Beschreibung |
| Use-Case Nummer | 09 |
| Kurze Beschreibung | Wenn man sich auf der Detailseite eines Produkts befindet und man auf ein Bild klickt, sollte sich eine Lightbox öffnen. |
| Voraussetzungen | Lightbox muss implementiert sein. |
| Akteure | Benutzer, Redaktor |
| Auslöser / Trigger | Ein Bild erweckt spezielles Interesse und man möchte es in vollem Detail bestaunen. |
| Standardablauf | Man befindet sich auf der Detailseite eines Produkts und klickt auf ein Produktbild. |
| Erwartetes Ergebnis | Die Lightbox öffnet sich. |
| Bemerkungen | / |
|  |  |

## Activity-Diagramm Frontend

Für die Darstellung des Ablaufs einerseits für einen Website-Benutzer im Frontend und andererseits für einen Redaktor im Backend wurden Zwei verschiedene Activity-Diagramme mit [Visual-Paradigm](#_Glossar) erstellt. Das erste stellt den Ablauf für die Benutzung der Website an sich dar. Die blauen Felder stellen einen Endpunkt dar, bei dem keine weiteren Aktionen ausgelöst werden können. Bemerkung: Bei jedem grünen Feld kann der Benutzer im Browser auf den «Zurück» klicken, er gelangt dann an den vorangehenden Punkt.



In der Konzepterarbeitung werden die Grundlagen für die Realisierung und Einführung eines Informatiksystems entwickelt.

Das Konzept wird schrittweise mit folgenden Schritten entwickelt.

Es ist wichtig, die Ergebnisse so weit zu deklarieren, dass damit die Systemarchitektur bestimmt werden kann. Abgestimmt mit der schrittweisen Entwicklung des Konzepts werden die Fertigprodukte? Sachmittel evaluiert.

Abbildung 14, Activity-Diagramm Frontend

## Activity-Diagramm Backend

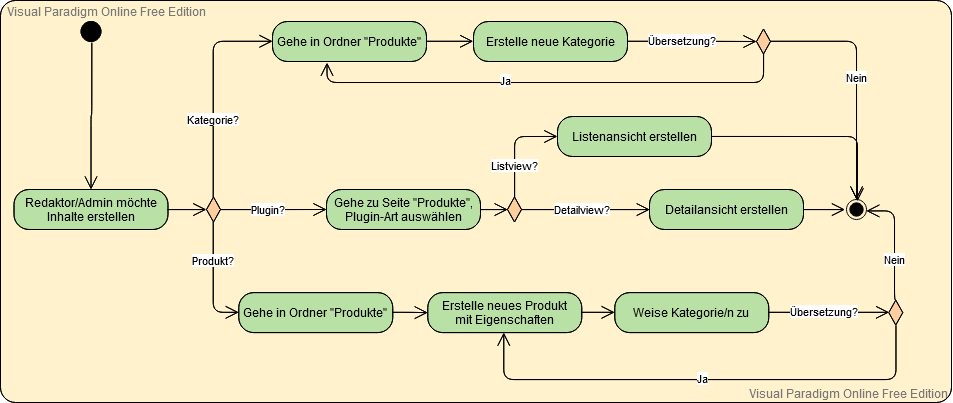
Dieses zweite Diagramm stellt den groben Ablauf dar, wie man als Redaktor oder Admin die benötigten Elemente und Plugins erstellen kann, damit das Potenzial der Extension ausgeschöpft wird. Die genaue Vorgehensweise der Konfiguration etc. wird später in der Benutzer-Dokumentation beschrieben.

Abbildung 15, Activity-Diagramm Backend

## Testkonzept

Das [Testing](#_Glossar) wurde bereits durch die detaillierte Aufgabenstellung eingeschränkt und wird daher nur in bedingtem Umfang stattfinden. Grob gesagt handelt es sich hier um einen sogenannten «[Smoke Test](#_Glossar)», der nur die wichtigsten Funktionen der Applikation auf ihre Richtigkeit überprüft. Der Sinn dahinter ist, dass nicht unnötige Ressource verschwendet werden bei Tests, die weiter ins Detail gehen. Wenn die Grundfunktionen fehlerhaft sind, ist detailliertes [Testing](#_Glossar) schlichtweg Zeitverschwendung und sollte vermieden werden.

Das Testkonzept entspricht exakt den User-Stories die im Kapitel [10.5](#_Use-Cases) definiert wurden. Aus Logiktechnischer Sicht werden alle Aspekte der Applikation abgedeckt, die unter Umständen in falschen Ergebnissen resultieren könnten. Ein [Unit-Test](#_Glossar) wird aus einerseits aus Zeitgründen und andererseits darum ausgelassen, weil die Funktionalitäten bereits bei der Entwicklung einsehbar sind. Allfällige Fehler im Programmcode fallen daher ziemlich rasch auf und können direkt behoben werden.

Die Tests werden zunächst von Drittpersonen anhand einer Benutzer-Dokumentation ähnlich der Use-Stories durchgeführt und die Ergebnisse schriftlich festgehalten. Wenn es zu unerwarteten Ergebnissen kommen sollte, wird sich der Entwickler der Aufgabe «Bugfixing» annehmen und die Fehler beheben. Darauf folgt erneutes [Testing](#_Glossar) durch den Entwickler, um den Programmcode auf seine Richtigkeit zu überprüfen.

# Realisierung

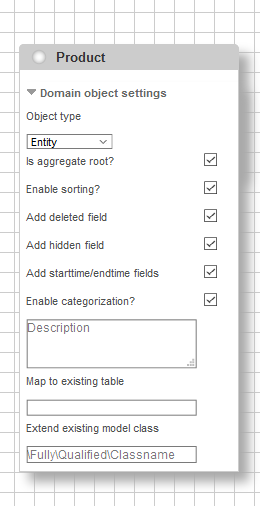
## Projektsetup

Wie bereits in einem vorangehenden Kapitel erwähnt, braucht es für den [Extensionbuilder](#_Glossar) eine TYPO3-Installation mit der LTS-Version 9.x.x. Aus diesem Grund wurde auf einer zusätzlichen Domain die entsprechende Version installiert. Hierzu muss man sich über [FTP](#_Glossar) mit dem Server verbinden und in der Kommandozeile einige Befehle eintippen. Unter anderem muss man die [tar.gz](#_Glossar)-Datei vom [TYPO3-Core](#_Glossar) herunterladen und diese entpacken. Danach müssen einige [Symlinks](#_Glossar) erstellt werden, damit die Index.php-Datei gefunden wird, wenn man die URL eingibt. Man klickt sich anschließend durch den Installationsprozess im TYPO3, danach kann man den [Extensionbuilder](#_Glossar) auf dem System installieren. Dazu geht man auf der linken Seite auf den Reiter „Extensions“ und sucht nach der entsprechenden Extension, die man installieren möchte. Nachdem das erledigt ist, kann die eigentliche Arbeit beginnen.

|  |  |
| --- | --- |
| Bild | Beschreibung des Arbeitsinhaltes/Vorgehen |
| … | … |
| … | ... |

## Domainmodel erstellen

Da das Datenmodell die Grundlage für die folgenden Arbeiten darstellt, wird es hier umfassend erklärt.

Mit dem [Extensionbuilder](#_Glossar) kann man nun das Datenmodell umsetzen, dass im Konzept als ERD dargestellt ist. Hierzu muss man einen neues «Model Object» erstellen, dem man den Namen «Product» gibt. Jetzt können verschiedenen Einstellungen vorgenommen werden, die im Anschluss den Programmcode entsprechend diesen Einstellungen generiert. Auf der Linken Seite ist zu sehen, welche Einstellungen genau vorgenommen werden können:

**«Is aggregate root?»** definiert, ob das Objekt eigenständig existieren kann, was bei einem Produkt der Fall ist.

**«Enable sorting»** erstellt im Repository eine Array -Variable namens $defaultOrderings, die für die Sortierung zuständig, im Verlauf der Realisierung werden hier noch zwei weitere Variablen hinzukommen.

**«Add ..»** erstellt entsprechende Datenbankfelder. **«Enable categorization»** bewirkt, dass ein Produkt im Backend einer Kategorie zugewiesen werden kann.

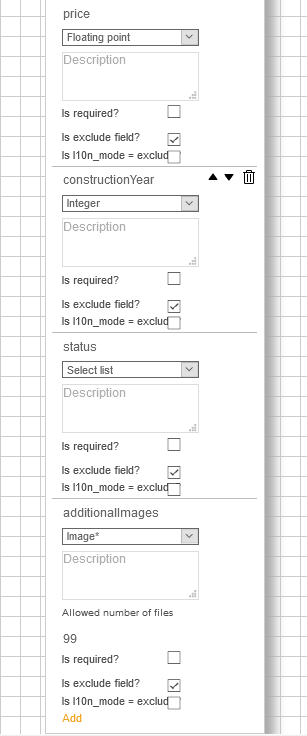
Da wir bei dem Filter eigene Funktionen im CategoryRepository schreiben müssen, könnte man ein weiteres Domain Model mit dem Namen «Category» erstellen und dieses bei «Map to existing table» der Tabelle sys\_category zuweisen. In diesem Fall wird das CategoryRepository aber manuell erstellt, bzw. erweitert, da dafür fast kein Aufwand betrieben werden muss.

Abbildung 16, Extensionbuilder Einstellungen

Weiter gibt es die Einstellungen «[Default actions](#_Glossar)», diese werden einerseits in der Datei ext\_localconf.php registriert, andererseits befindet sich die eigentliche [Action](#_Glossar) im Controller des jeweiligen Domain Objects. In unserem Fall werden die [Actions](#_Glossar) «list» für die Listenansicht und «show» für die Detailansicht benötigt. Weiter braucht es eine «Custom action» für die Merkliste. Diese wird einfachheitshalber «watchList» genannt.

Jetzt können die Eigenschaften, respektive die individuellen Datenbankfelder erstellt werden.

Hierbei ist es vor allem wichtig, die richtigen Datentypen zu wählen, da ansonsten ein falsches SQL für die Tabellenerstellung erzeugt wird. Auch bei den Eigenschaften gibt es einige Einstellungen, die man vornehmen sollte. Die Einstellungen sind wie folgt:

Das erste Feld bezeichnet den Namen des Datenbankfeldes, in diesem Fall sind das die Eigenschaften, die ein Produkt haben soll. Als nächstes wählt man den Datentypen für das jeweilige Feld, man hat hier relativ viele Möglichkeiten.

Beim Feld «additionalImages» ist das aber ein wenig speziell. Wenn man als Datentyp «image» auswählt, kann man definieren, wie viele Bilder maximal dem Produkt zugewiesen werden können. Wenn dieser Wert grösser als 1 ist, wird automatisch eine Zwischentabelle erstellt, was am Anfang ein Bisschen verwirrend sein kann.

Die Checkbox «Is required» fügt beim SQL-Query nach dem Datentypen zusätzlich die Worte «NOT NULL» hinzu, was eine Speicherung eines Datensatzes, bei dem dieses Feld leer ist, unmöglich macht. Ein Produkt benötigt in diesem Sinn nichts weiter, als einen Namen (Datenbankfeld «title»).

Wenn «is excluded field» ausgewählt ist, kann nur der Admin-Benutzer der Website dieses Feld sehen und bearbeiten, nicht aber die Redaktoren.

Um bei einzelnen Feldern die Übersetzung automatisch zu kreieren, kann man «is l10n\_mode =exclude» anwählen. Das ist beispielsweise bei Bildern ohne Text ein gute Idee, da es einen Arbeit erspart. Sicherheitshalber wird das hier aber bei keinem der Felder ausgewählt, da man eben nie wissen kann, wann und ob man vielleicht trotzdem eine Übersetzung für ein Feld braucht.

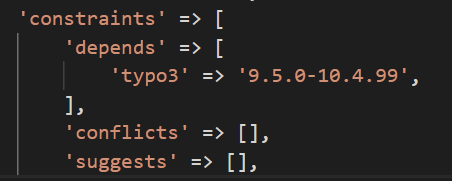
Da jetzt alles soweit definiert ist, kann man die Extension bereits herunterladen und auf dem neuen System installieren. Das das funktioniert, muss zuerst in der neu erstellten Extension die benötigte TYPO3-Version auf LTS 10.x.x erhöht werden. Diese Einstellung findet man in der Datei ext\_emconf.php (siehe Bild rechts).

Abbildung 17, TYPO3-Version für EXT anpassen

Nun erfolgt die Installation auf dem neuen System mit der Version 10.

## Flexform

Ein [Flexform](#_Glossar), aka [TCA](#_Glossar) beschreibt die Eingabemaske bei Inhaltselementen oder anderen Domain Objects wie beispielsweise Produkten des TYPO3 Backends.

## Filterfunktion

Merkliste

Detailseite

## Fluid-Templates und Styling

## Buxfixing

## Testing

Von Drittperson testen lassen

* Die Testprotokolle fassen die Ergebnisse zusammen:
* Testobjekt
* Angaben zur Durchführung (konnten alle Tests vom Konzept durchlaufen werden?
* Testresultate
* Was passiert mit einem „Failtesting“? „Re-Testing“ oder leichter Fehler?
* Testfazit und Empfehlung

### Testfall

Hinweis auf Testkonzept

| ID / Bezeichnung | *T-001* |  |
| --- | --- | --- |
| Beschreibung |  | |
| Testvoraussetzung |  | |
| Testschritte |  | |
| Erwartetes Ergebnis |  | |

### Testdurchführung und Testergebnis

| Tester |  |
| --- | --- |
| Datum Testdurchführung |  |
| Fehlerklasse (Testergebnis) |  |
| Fehlerbeschreibung |  |

### Testwürdigung

Feedback zu den Testergebnissen

### Testfazit

Fazit als Tester..

### Weiteres Vorgehen

Erfolgreich, Nachtests Wiederholung Testfall xy, Regression nötig

## Einführung vorbereiten

Die betrieblichen und organisatorischen Änderungen, welche bei der Einführung auftreten, müssen vorbereitet werden:

* Sicherstellen der Produktionsumgebung
* Datenmigration
* Ausbildung planen
* Ausbildungsmaterial und Handbücher erstellen
* Manuals

## Schutzmassnahmen umsetzen

# Abschlussbericht

Wurden die geforderten Ziele erreicht?

## Vergleich Ist/Soll

Ist die Umsetzung wie geplant oder gab es Differenzen?

## Mittelbedarf

Welche Mittel wurden gebraucht oder mussten noch beschafft werden?

## Realisierungsbericht

Gab es Probleme während der Realisation? Ungeplante Sachen zum Vorschein oder Ergänzungen?

## Testbericht

Wie verliefen die Tests (Erfolgreich / weniger Erfolgreich)? Gibt es Fehler, die schwerwiegend sind oder solche die später korrigiert werden können/müssen?

## Fazit zum IPA (Projekt)

Wie ist das Projekt verlaufen (Objektive) Meinung

Hatte das Projekt Stolpersteine, welche etc…

## Persönliches Fazit

Feedback und Reflexion des Lernenden rund um die IPA.

Was war gut, was weniger? Was habe ich gelernt und was würde ich ev. das nächste Mal anders machen? Ausführlich formulieren.

## Schlussreflexion

Die vorgesehenen Massnahmen zum Schutz des Systems werden umgesetzt, deren Umsetzung überprüft und falls deren Wirkung durch gezielte Verbesserungen erhöht, so dass die Anforderungen an die Sicherheit und den Datenschutz erfüllt sind.

ISDS-Konzept überprüfen und ergänzen

Schutzmassnahmen durchführen

Schutzmassnahmen verbessern

# Glossar

Hier werden jegliche relevanten Begrifflichkeiten aufgelistet, die unter Umständen einer Erläuterung bedürfen.

Tabelle 26, Glossar

|  |  |
| --- | --- |
| Begriff | Bedeutung |
| AbstractWidgetViewHelper | TYP03-eigenes AJAX zur dynamischen Ausgabe von Inhalten aus einer Datenbank. |
| AJAX | “Asynchronous JavaScript and XML”  Wird unter anderem dafür verwendet, um per JavaScript Datenbanken auszulesen. |
| AJAX-Call | Wird ausgelöst, wenn der Benutzer einen Filter aktiviert. Daten werden gesammelt und ausgegeben. |
| ajaxWidgets | siehe «AbstractWidgetViewHelper» |
| BaseURL | Variable, die in TYPO3 dazu verwendet wird, das Frontend aufzurufen. |
| **Bugfixes** | Je nachdem gibt es Situationen, in denen die TYPO3 Installation etwas tut (oder eben nicht tut), was so nicht vorgesehen ist. Das nennt man einen Bug. Ein Bugfix löst dieses Problem, damit die Installation einwandfrei läuft. |
| CKEditor | Eingabefeld im Backend, z.B für Inhaltselemente. Der Text kann z.B fett gemacht oder einer Überschriftengruppe zugewiesen werden. |
| **Clear Web** | Der Teil des Internets, der mit herkömmlichen Browsern erreicht werden kann. Das Gegenstück wäre das sog. «Deep Web», in dem es viele Sicherheitsrisiken gibt. |
| **CMS** | Content Management System, der Gedanke dahinter ist, dass Websites von den Besitzern selbst verwaltet werden können. |
| **Code** | Jegliche Art von Programmzeilen. |
| **Cookie** | Ermöglicht es, eine [Session](#_Glossar) eines Websitebenutzers zu speichern. |
| Default actions | Funktionen, die im File ext\_localconf.php registriert werden und sich im Controller befinden (listAction). |
| **Domain Object** | Eine Instanz von «Produkt» oder eines anderen abstrakten Objekts, deren Eigenschaften in der Datenbank gespeichert werden. |
| Errorhandling | Für eine nicht existierende Seite kann z.B eine 404-Seite ausgegeben werden, die auf den Fehler hinweist. Ziel ist es, dass die Applikation nicht abstürzt, sondern den Fehler erkennt und entsprechend handelt. |
| SelectList | Dropdown-Liste in einem TCA zum Auswählen von z.B Antriebsarten oder Status. |
| **ExtBase** | Vorgehensweise und Definition von z.B Namensgebungen zum Entwickeln von TYPO3-Extensions. Stichwort «Vereinheitlichung», jede Extension sollte gleich aufgebaut sein. |
| Extension | Ein Stück Software, das in TYPO3 installiert werden kann. Eine Extension bietet individuelle Funktionen, die in TYPO3 Verwendung finden können. Beispielsweise Plugins für einen Warenkorb. |
| Extensionbuilder | TYPO3 Erweiterung zum Erstellen von Datenmodellen (SQL-Querys) |
| Flexform / TCA | Eingabemaske für Elemente, die in TYPO3 erstellt werden können. Mit Flexform ist der XML-Code gemeint, das TCA ist das, was man effektiv im Backend sieht, also das Resultat aus dem XML- Flexform. |
| Fluid | HTML-Struktur die es einen ermöglicht, mit Abfragen, Schleifen etc. Inhalte auszugeben. |
| Font-Awesome | Stellt Piktogramme für den Gebrauch im Internet zur Verfügung. Siehe <https://fontawesome.com/icons> |
| **FTP** | Steht für “File transfer protocol». Es ermöglicht den Austausch von Dateien zwischen einem Webserver und einem Client. |
| Getter und Setter | "Getter" und "Setter" sind Objektmethoden, mit denen man den Zugriff auf eine bestimmte Klassenvariable / Eigenschaft steuern kann. Man kann die Werte von Objekteigenschaften auslesen oder diese definieren. |
| **Lightbox** | Eine JavaScript-Bibliothek mit der man Bilder \*in Szene» setzen kann.  <https://lokeshdhakar.com/projects/lightbox2/> |
| [**LTS-Version**](#_Glossar) | Eine LTS-Version ist eine neue Version von TYPO3. Anders als bei Minor-Updates kann es sein, dass die Systemarchitektur angepasst wird und es müssen, je nach Extension, viele Dinge angepasst werden. |
| **MariaDB** | Datenbanktyp, andere wären z.B MySQL oder SQLite. MariaDB ist [OpenSource](#_Glossar) und ein kleines Bisschen schneller als MySQL. |
| **Minor-Update** | TYPO3-Core Update, Bugfixes und evtl. Securityupdate |
| **OpenSource** | Eine Software wird als OpenSource bezeichnet, wenn der Code öffentlich zugänglich ist und nach belieben angepasst werden darf. WW |
| **Plesk** | International agierender Hosting-Anbieter |
| queryResult | Array bestehend aus Datenbankfeldern, das einem SQL-Query aus dem Repository entspricht. |
| Repository | Stellt in TYPO3 einen Teil des Models vom MVC dar, dessen Aufgabe ist es, SQL-Querys an die Datenbank zu schicken und diese an den Controller weiterzugeben. |
| Request | Bei einem Request wird eine Aktion im Controller aufgerufen. Der Benutzer klickt also beispielsweise auf einen Filter, worauf die Controlleraktion «listAction()» aufgerufen wird. |
| **SCSS** | Steht für «Syntactically Awesome Style Sheets». Eine bessere Alternative zu herkömmlichem CSS, da Verschachtelungen möglich sind. Benötigt aber deswegen einen Compiler. |
| **SCSS-Compiler** | Kompiliert die Dateien, die auf .scss enden und wandelt diesen in herkömmliches CSS um. |
| **Securityupdate** | Verbessert die Sicherheit der Installation, beispielsweise durch Passwortverschlüsselung. |
| **Session** | Identifiziert den Browser und den Benutzer, dieser erhält eine einzigartige ID zugewiesen, woran er wiedererkannt werden kann. |
| **Session-Cookie** | Speichert die [Session](#_Glossar) eines Websitebenutzers, wird aber beim Schliessen des Browsers gelöscht. |
| Smoke-Test | Testmethode deren Ziel es ist, die Grundfunktionen einer Applikation zu testen und die Ergebnisse festzuhalten. Detaillierte Tests werden vorerst ausser Acht gelassen, da diese reine Zeitverschwendung wären, wenn die Grundfunktionen fehlerhaft sind. |
| **SQL-Injection** | Versuch, durch Eingabe von SQL-Querys in z.B einem Suchfeld unautorisiert Daten auszulesen. Beispielsweise Benutzernamen und Passwörter. |
| SQL-Query | Eine Datenbankabfrage mit oder ohne Bedingungen. |
| **Styling** | Die grafische Darstellung einer Website, z.B Schriftarten, Farben etc. |
| Symlink | Erfüllt den gleichen Zweck wie eine Verknüpfung in Windows. |
| **Tajo** | Interner Server von Terminal8, auf dem unsere Websites gehostet und verwaltet werden. |
| Tar.gz | Im Grunde genommen eine Zip-Datei. Das Prinzip ist das gleiche, nur dass es auf Linux-Betriebssystemen verwendet wird. |
| **Testing** | Testen von Software |
| **Transition** | Übergang, der mit CSS definiert werden kann. Beispielsweise bei anklicken eines Links, der dann die Farbe wechselt. |
| **TYPO3** | Eine bestimmte Art eines CMS, vergleichbar mit Joomla oder WordPress, einfach komplexer. |
| TYPO3-Core | Synonym für TYPO3-Installation |
| **TypoScript** | TYPO3-Eigene Konfigurationssprache. |
| Unit-Test | Testmethode für ein Programm, in dem der Programmcode soweit möglich in Einzelteile zerlegt und getestet wird. |
| URL | Steht für «Einheitlicher Ressourcenzeiger», mit dessen Hilfe werden Websites identifiziert und lokalisiert. Im Grunde genommen ein Alias für eine IP-Adresse, kann aber auch dazu verwendet werden, Daten zu übergeben. |
| **User-Experience** | Synonym für Benutzerfreundlichkeit. |
| Viewhelper | Kann im Fluid-Template zur Ausgabe jeglicher Elemente verwendet werden. |
| Visual-Paradigm | Tool zum Erstellen von aller Art Diagrammen (Datenbankmodell, Klassendiagramm etc.) |
| **VS-Code** | Tool von Microsoft zum Bearbeiten von Code, bietet viele Möglichkeiten zur Erweiterung. |

Tipp: Word 2010/13 Layout/Sortieren nach… ;-)

# Literatur- und Quellenverzeichnis

Tabelle 27, Literatur- und Quellenverzeichnis

|  |  |
| --- | --- |
| Bezeichnung | Bemerkungen |
| Hermes 5.1 | <https://www.hermes.admin.ch/de/projektmanagement/verstehen/ubersicht-hermes/methodenubersicht.html> |
| Abbildung «TYPO3-Backend» | https://docs.typo3.org/m/typo3/tutorial-getting-started/master/en-us/\_images/BackendAboutModules.png |

# Tabellen- und Abbildungsverzeichnis

Abbildungsverzeichnis

[Abbildung 1, Arbeitsplatz 7](#_Toc69282654)

[Abbildung 2, Logo T8 7](file:///C:\Users\Moritz\Desktop\Projekte\probst-maveg\Dokumentation_0.6\05_Dokumentation.docx#_Toc69282655)

[Abbildung 3, Git Push 12](#_Toc69282656)

[Abbildung 4, Git Pull 12](#_Toc69282657)

[Abbildung 5, Phasenplanung 15](#_Toc69282658)

[Abbildung 6, Meilensteinplanung 15](#_Toc69282659)

[Abbildung 7, Modulplan nach Phasen 17](file:///C:\Users\Moritz\Desktop\Projekte\probst-maveg\Dokumentation_0.6\05_Dokumentation.docx#_Toc69282660)

[Abbildung 8, Risikoportfolio 35](#_Toc69282661)

[Abbildung 9, TYPO3-Backend 41](#_Toc69282662)

[Abbildung 10, Projektrollen 46](#_Toc69282663)

[Abbildung 11, ERD 47](#_Toc69282664)

[Abbildung 12, Klassendiagramm 48](file:///C:\Users\Moritz\Desktop\Projekte\probst-maveg\Dokumentation_0.6\05_Dokumentation.docx#_Toc69282665)

[Abbildung 13, Use-Case-Diagramm 49](#_Toc69282666)

[Abbildung 14, Activity-Diagramm Frontend 53](file:///C:\Users\Moritz\Desktop\Projekte\probst-maveg\Dokumentation_0.6\05_Dokumentation.docx#_Toc69282667)

[Abbildung 15, Activity-Diagramm Backend 54](file:///C:\Users\Moritz\Desktop\Projekte\probst-maveg\Dokumentation_0.6\05_Dokumentation.docx#_Toc69282668)

Tabellenverzeichnis

[Tabelle 1, Module 17](#_Toc69282621)

[Tabelle 3, Arbeitsjournal 07.04.2021 20](#_Toc69282622)

[Tabelle 4, Arbeitsjournal 08.04.2021 21](#_Toc69282623)

[Tabelle 5, Arbeitsjournal 09.04.2021 22](#_Toc69282624)

[Tabelle 6, Arbeitsjournal 12.04.2021 23](#_Toc69282625)

[Tabelle 7, Arbeitsjournal 14.04.2021 25](#_Toc69282626)

[Tabelle 8, Arbeitsjournal 15.04.2021 26](#_Toc69282627)

[Tabelle 9, Arbeitsjournal 16.04.2021 27](#_Toc69282628)

[Tabelle 10, Arbeitsjournal 19.04.2021 27](#_Toc69282629)

[Tabelle 11, Arbeitsjournal 20.04.2021 28](#_Toc69282630)

[Tabelle 12, Arbeitsjournal 21.04.2021 28](#_Toc69282631)

[Tabelle 7, Unterschriften Phasenfreigabe 31](#_Toc69282632)

[Tabelle 8, Risikoanalyse 34](#_Toc69282633)

[Tabelle 9: Risikoanalyse Tabelle 34](#_Toc69282634)

[Tabelle 10, Varianten Filterfunktion 36](#_Toc69282635)

[Tabelle 11, Nutzwertanalyse Filterfunktion 37](#_Toc69282636)

[Tabelle 12, Varianten Merkliste 38](#_Toc69282637)

[Tabelle 13, Nutzwertanalyse Merkliste 39](#_Toc69282638)

[Tabelle 14, Benötigte Extensions 42](#_Toc69282639)

[Tabelle 25, MVC-Erläuterung 43](#_Toc69282640)

[Tabelle 2, Projektrollen 46](#_Toc69282641)

[Tabelle 15, Use-Case Akteure 49](#_Toc69282642)

[Tabelle 16, Use-Case 01 50](#_Toc69282643)

[Tabelle 17, Use-Case 02 50](#_Toc69282644)

[Tabelle 18, Use-Case 03 50](#_Toc69282645)

[Tabelle 19, Use-Case 04 51](#_Toc69282646)

[Tabelle 20, Use-Case 05 51](#_Toc69282647)

[Tabelle 21, Use-Case 06 51](#_Toc69282648)

[Tabelle 22, Use-Case 07 52](#_Toc69282649)

[Tabelle 23, Use-Case 08 52](#_Toc69282650)

[Tabelle 24, Use-Case 09 52](#_Toc69282651)

[Tabelle 26, Glossar 60](#_Toc69282652)

[Tabelle 27, Literatur- und Quellenverzeichnis 64](#_Toc69282653)

# Anhang

Code einfügen, Layout abklären

Ausgedruckter Code, Sitzungsprotokolle, usw.: Vollständig aufführen und der Dokumentation beilegen.

Listings von Scripten und Programmen. Die Eigenleistung der Kandidatin oder des Kandidaten muss vollständig dokumentiert sein. Automatisch generierten Code weglassen, wenn für das Verständnis nicht zwingend nötig.

Falls Handbücher erstellt wurden, können diese hier als Anhang beigelegt werden.