



TERMINAL 8

ERWEITERUNG TYPO3 CMS WEBSITE MIT PRODUKTEDATENBANK

Terminal8 GmbH

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 :	0.4

Autor: Moritz Burn
IPA 2021, Kanton Bern

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
Ablauf Organisation und Umfeld	7
1. Aufgabenstellung.....	8
1.1 Titel der Arbeit	8
1.2 Thematik.....	8
1.3 Ausgangslage.....	8
1.4 Detaillierte Aufgabenstellung	9
1.5 Teil 1:	9
1.6 Teil 2:	10
1.7 Mittel und Methoden	11
1.8 Vorkenntnisse.....	11
1.9 Vorarbeiten.....	11
2. Organisation der IPA.....	12
2.1 Datensicherung der IPA.....	12
2.2 Versionierung	13
2.2.1 Dokumentation	13
2.2.2 Software	13
2.3 Hilfestellungen	13
3. Planung.....	14
3.1 Phasen	15
3.2 Meilensteine	15
3.3 Szenario	16
3.4 Module.....	16
3.5 Projekorganisation	18
3.5.1 Projektrollen	18
4. Zeitplan	19
5. Firmenstandards	20
6. Arbeitsjournal.....	21
6.1 Arbeitsjournal 07.04.2021	21
6.2 Arbeitsjournal 08.04.2021	22
6.3 Arbeitsjournal 09.04.2021	23
6.4 Arbeitsjournal 12.04.2021	24
7. Abschlussbericht	26
7.1 Vergleich Ist/Soll.....	26
7.2 Mittelbedarf.....	26
7.3 Realisierungsbericht	26
7.4 Testbericht.....	26
7.5 Fazit zum IPA (Projekt)	26
7.6 Persönliches Fazit	26
7.7 Schlussreflexion	26
8. Unterschriften und Abnahmeprotokoll.....	27

9.	Initialisierung.....	28
9.1	Ist-Zustand	28
9.2	Projektziele	28
9.3	Anforderungen.....	29
9.3.1	Funktionale Anforderungen.....	29
9.3.2	Nicht funktionale Anforderungen.....	29
9.4	Risikoanalyse	30
9.5	Risikograph	31
9.5.1	Kurze Stellungnahmen zu den Risiken	31
9.6	Schutzbedarfsanalyse	31
9.7	Varianten	32
9.7.1	Filterfunktion	32
9.7.2	Nutzwertanalyse	33
9.7.3	Entscheid.....	33
9.7.4	Zwischenspeichern der Merklste.....	34
9.7.5	Nutzwertanalyse	35
9.7.6	Entscheid.....	35
10.	Konzept.....	36
10.1	ISDS-Konzept.....	36
10.2	Systemumgebung, Anforderungen und Konfiguration.....	36
10.2.1	Vorarbeiten Webserver.....	36
10.2.2	Vorarbeiten TYPO3	37
10.3	Datenbankkonzept (Nochmals prüfen).....	39
10.4	Klassendiagramm (noch anpassen, oberstes Element falsch, Rest richtig=>Datentypen nochmals prüfen)	40
10.5	Use-Cases.....	41
10.5.1	Use-Case Beschreibungen	42
10.6	Activity-Diagramm	45
10.7	Architekturkonzept.....	45
10.7.1	Model – View – Controller.....	45
10.7.2	Listenansicht mit Filter	46
10.7.3	Detailansicht.....	46
10.7.4	Merklste	46
10.8	Testkonzept.....	46
11.	Realisierung	47
11.1	Domainmodel erstellen	47
11.2	Erweiterung installieren	47
11.4	Testprotokoll.....	48
11.4.1	Testfall.....	48
11.4.2	Testdurchführung und Testergebnis	48
11.4.3	Testwürdigung	49
11.4.4	Testfazit.....	49

11.4.5	Weiteres Vorgehen	49
11.5	Einführung vorbereiten	50
11.6	Schutzmassnahmen umsetzen	50
12.	Glossar	51
13.	Literatur- und Quellenverzeichnis.....	55
14.	Tabellen- und Abbildungsverzeichnis	56
	Abbildungsverzeichnis.....	56
	Tabellenverzeichnis.....	56
15.	Anhang	58

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

Verwendete Abkürzungen

Abkürzung	Bedeutung
IPA	Individuelle praktische Arbeit
OdA	Organisation der Arbeitswelt
QV	Qualifikationsverfahren
VP	Verantwortliche Fachkraft
VZ	Verzeichnis
CMS	Content Management System
URL	«Uniform Ressource Locator»
DB	Datenbank

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 TYPO3](#).

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](#)» erstellt, welche die in PHP nötigen [Getter und Setter](#) und die Datenbankankbindung kreiert. Für den Produkte-Filter wird ein Ajax-Call gemacht, der die nötigen Daten per [URL](#)-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](#) 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
Jahr der Durchführung
Projektvorgehensmodell

Moritz Burn
2021
HERMES 5.1

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



Abbildung 1, Arbeitsplatz



TERMINAL8

Abbildung 2, Logo T8

1. Aufgabenstellung

1.1 Titel der Arbeit

Erweiterung bestehende TYPO3 CMS Website mit Produktdatenbank.

1.2 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.

1.3 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.

1.4 Detaillierte Aufgabenstellung

Die IPA besteht aus zwei Teilen:

Teil 1 Backend und Teil 2 Frontend.

1.5 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:
 - o Titel
 - o Kurz-Beschreibung
 - o ext
 - o Vorschau Bild
 - o Antrieb (Ohne, Benzin, Elektro, Diesel)
 - o Leistung
 - o Preis
 - o Baujahr
 - o Status (Neu, Demo, Occasion)
 - o weitere Bilder für eine Galerie auf der Detailseite
 - o Angehängte Dateien (PDF & Word, zusätzliche Bilder)
- Kategorien:
 - o Hauptkategorie (z. B. Marke: Hitachi, Yanmar, CAT)
 - o Nebenkategorie (z. B. Art: Minibagger, Raupenbagger, Grossbagger, Mobilbagger, Radlader)
- auch Kategorien können durch die Redaktoren übersetzt werden

1.6 Teil 2:

- Listenansicht mit allen Produkten
- Listenansicht ist wie folgt Dynamisch konfigurierbar:
 - o Sortierung: Alphabetisch oder Reihenfolge des Backends übernehmen
 - o Anzeigen von Produkten der gewählten Kategorie(n)
 - o Anzeigen von Produkten aus gewählten Ordner(n)
 - o Anzeigen von einzeln ausgewählten Produkten
 - o Auswahl der Kategorie(n) für die Filterung
 - o Filter wird nur angezeigt, wenn Kategorien im Plugin gewählt wurden
- Merkliste
 - o Produkte können einer Merkliste hinzugefügt werden
 - o 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 Mobilien 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:
 - o eine/mehrere Marken wählen = zeigt alle Produkte, die Marke X ODER Marke Y enthalten
 - o 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:
 - o Erstellen Backend User mit Berechtigungen zur Erfassung von Produkten
 - o Benutzeranleitung zur Erfassung der Produkte im Backend
 - o Testing
 - o Erfassung von Produkten von 3. Person
 - o «Benutzen» des Frontends durch 3. Person
- Kein weiteres Testing vorgesehen
- Nicht Bestandteil dieser Arbeit sind:
 - o Installation & Grundkonfiguration TYPO3 - Bereitstellen des Servers
 - o Abfüllen der Produkte - Einführung der Erweiterung
 - o Allgemeines Websitelayout (Navigation, Footer)

1.7 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](#)

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>

1.8 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

1.9 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

2. Organisation der IPA

2.1 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.

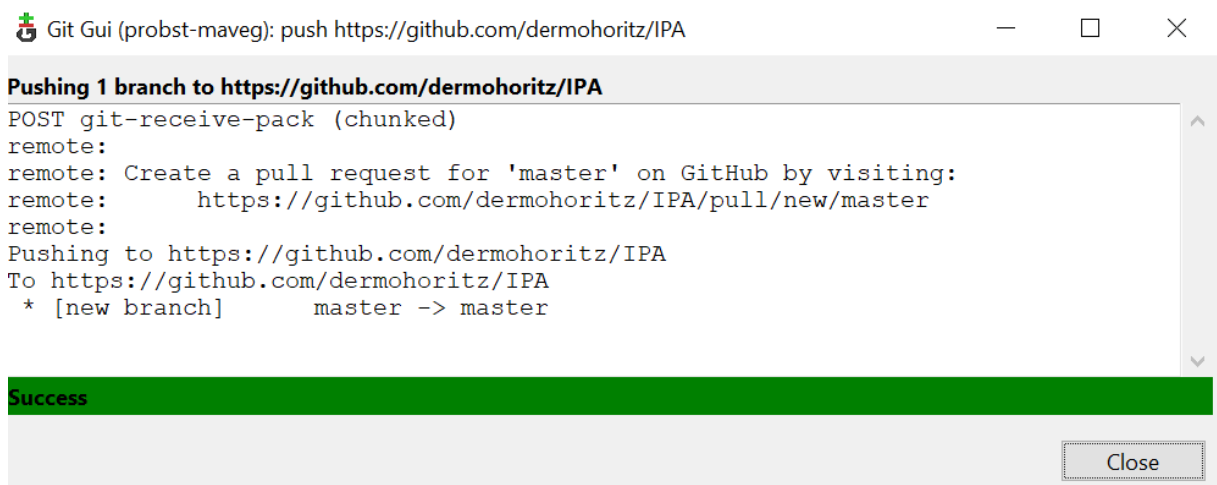


Abbildung 3, Git Push

```
Moritz@DESKTOP-2U9IC69 MINGW64 ~/Desktop/Projekte/probst-maveg (master)
$ git pull https://github.com/dermohoritz/IPA master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 653 bytes | 36.00 KiB/s, done.
From https://github.com/dermohoritz/IPA
* branch      master      -> FETCH_HEAD
Updating 1e705bf..a0a7a43
Fast-forward
 test_pull | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 test_pull

Moritz@DESKTOP-2U9IC69 MINGW64 ~/Desktop/Projekte/probst-maveg (master)
$ |
```

Abbildung 4, Git Pull

2.2 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 Anderen dem eigentlichen Softwareprodukt.

2.2.1 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.

2.2.2 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](#) dafür gemacht wurde. Welche Änderungen und Funktionen genau dazugekommen sind, kann man der GitHub-Dateiablage oder dem Arbeitsjournal entnehmen.

2.3 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.

3. 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
 - a. Die Initialisierung hat aufgrund der bereits im Detail geklärten Aufgabenstellung bereits begonnen und wird nun weitergeführt.
 - b. 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 einem späteren Zeitpunkt.
 - c. 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.

3.1 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.

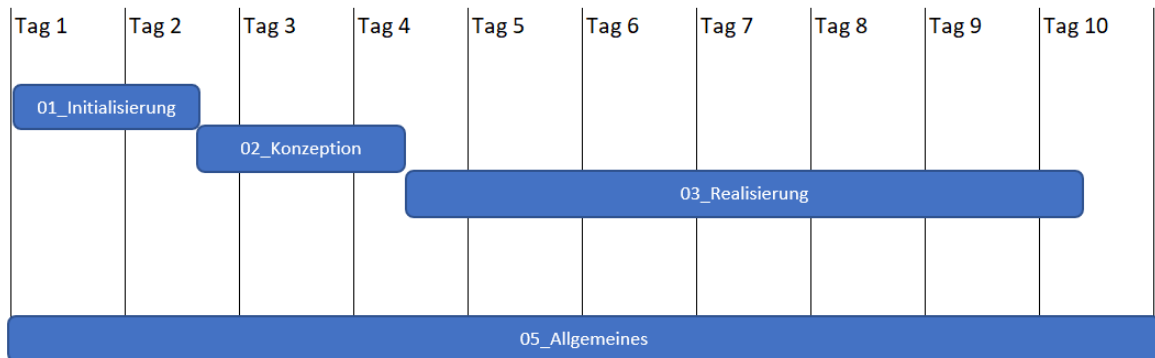


Abbildung 5, Phasenplanung

3.2 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:

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.

3.3 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

3.4 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
 - o Das Modul Geschäftsorganisation hat insofern nichts mit dem eigentlichen Projekt zu tun, da es sich um einen externen Auftrag handelt.
- Projektgrundlagen
 - o Da die Anforderungen an die Applikation bereits im Voraus definiert wurden, wird dieses Modul nicht berücksichtigt.
- Einführungsorganisation
 - o Da die Website erst zu einem späteren Zeitpunkt fertiggestellt wird, bedarf es keiner Einführungsorganisation.
- IT-Migration
 - o Es handelt sich bei diesem Projekt um ein komplett neues System, was eine Migration unmöglich macht. Deshalb wird dieses Modul nicht berücksichtigt.
- Beschaffung
 - o 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

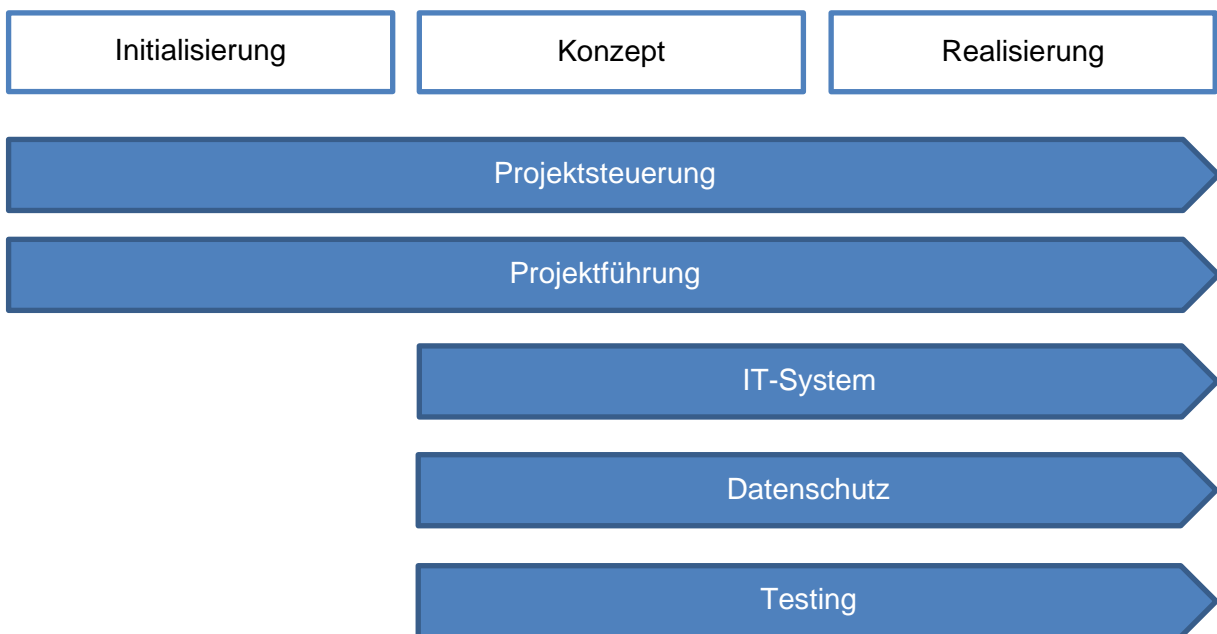


Abbildung 7, Modulplan nach Phasen

3.5 Projekorganisation

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

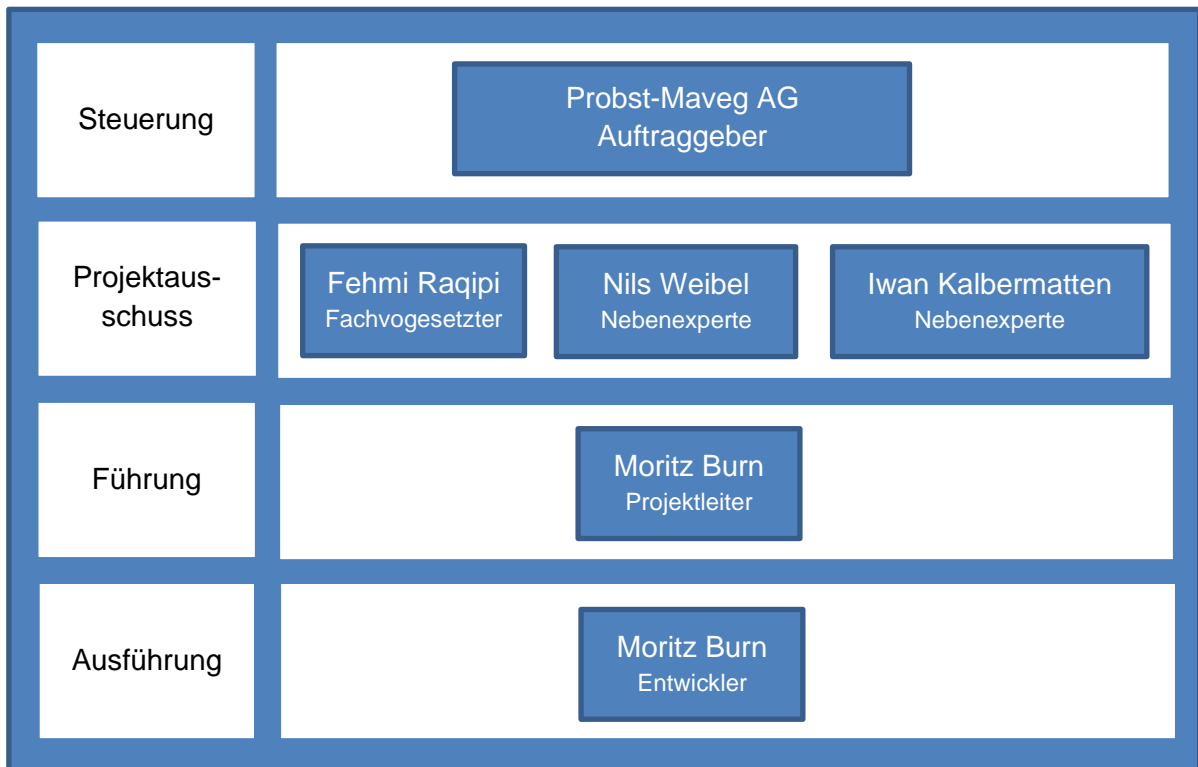


Abbildung 8, Projektrollen

3.5.1 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.

4. Zeitplan

5. 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 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](https://docs.typo3.org/m/typo3/book-extbasefluid/master/en-us/a-CodingGuidelines/Index.html). Hier wird beispielsweise die Ordnerstruktur und welche Arten von Dateien dort abgespeichert werden dürfen vorgegeben. Auch die Namenskonventionen von Datenbanktabellen, Klassen- und Variablenamen sind hier definiert. Genauer kann über diesen Link nachgeschlagen werden:
<https://docs.typo3.org/m/typo3/book-extbasefluid/master/en-us/a-CodingGuidelines/Index.html>

6. Arbeitsjournal

6.1 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.			
Hilfestellungen			
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			
<p>Was lief gut:</p> <p>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.</p> <p>Was lief weniger gut:</p> <p>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.</p> <p>Meine Erkenntnis von heute:</p> <p>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.</p>			
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.

6.2 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.			
Hilfestellungen			
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.

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.

6.3 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			
<p>Am Anfang des Tages habe ich nochmals die Variantenentscheide durchgelesen, wobei mir ein paar Dinge wie Rechtschreibfehler 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.</p>			

Hilfestellungen
AbstractWidgetViewhelper
Reflexion
<p>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.</p> <p>Was lief weniger gut: Schlechte Erfahrungen habe ich heute keine gemacht, da ich konzentriert arbeiten konnte.</p>
Nächste Schritte
Die nächsten Schritte sind einige Diagramme (Klassendiagramm, ERD, Activity-Diagramm) und die Use-Cases als auch das Testkonzept.

6.4 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			
<p>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 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.</p>			
Hilfestellungen			
<p>Tool für die Erstellung vom ERD : https://online.visual-paradigm.com</p> <p>Tool für die Erstellung des Klassendiagramms : https://lucid.app/</p>			

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 beschern könnte.

7. Abschlussbericht

7.1 Vergleich Ist/Soll

7.2 Mittelbedarf

7.3 Realisierungsbericht

7.4 Testbericht

7.5 Fazit zum IPA (Projekt)

7.6 Persönliches Fazit

7.7 Schlussreflexion

8. 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.

Tabelle 7, Unterschriften Phasenfreigabe

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

9. Initialisierung

9.1 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.

9.2 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.

9.3 Anforderungen

9.3.1 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 Produktnamen.
 - Produkte können in verschiedenen Sprachen angezeigt werden.
 - Auf Detailseiten erscheint eine Lightbox, wenn ein Bild angeklickt wird.

9.3.2 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.

9.4 Risikoanalyse

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 handelt, hat es keinen Einfluss. Bei einer neuen LTS-Version 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

9.5 Risikograph

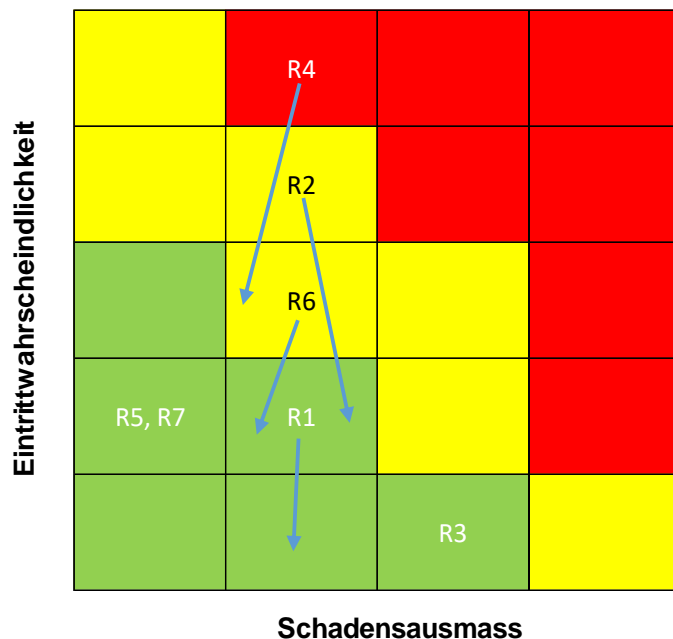


Abbildung 11, Risikoportfolio

9.5.1 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.

9.6 Schutzbedarfsanalyse

Eine Schutzbedarfsanalyse ist in diesem Fall nicht unbedingt notwendig, da die Produktdatenbank 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.

9.7 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.

9.7.1 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	Beschreibung	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
2	AJAX	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-Experience Zeitsparender Übergang kann mit SCSS animiert werden	Ungefähr 1-2 Stunden Aufwand
3	AbstractWidget-ViewHelper	Das ist die TYPO3-eigene Version die sog. ajaxWidgets 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.

9.7.2 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%		<u>1.95</u>		<u>2.70</u>		<u>2.35</u>

9.7.3 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 «einfiegen» 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](#) möglich ist. Der [AbstractWidgetViewHelper](#) 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](#)» keine Sicherheitsrisiken erwartet werden und es somit kein Grund zur Deaktivierung von JavaScript gibt.

9.7.4 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	Beschreibung	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	Ein normales Cookie ist im Grunde genommen eine Textdatei, die auf dem Gerät des Endnutzers gespeichert wird, um das «Surferlebnis» 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	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.

9.7.5 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%		<u>1.65</u>		<u>2.90</u>		<u>2.50</u>

Tabelle 6: NWA, Nutzwertanalyse Merklistenspeicherung

9.7.6 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.

10. 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.

10.1 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](#) 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.

10.2 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.

10.2.1 Vorarbeiten Webserver

Die Systemumgebung ist in diesem Fall ein Ubuntu-Server, auf dem die Websites gehostet werden. Dieser wird von [Plesk](#) 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 Webserver 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](#), einer Abspaltung von MySQL, wird ebenfalls auf dem Server erstellt.

10.2.2 Vorarbeiten TYPO3

Installation

Jetzt kann man über die Entwicklungsumgebung, in diesem Fall [VS-Code](#) und der Hilfe einer Erweiterung per FTP auf den Server zugreifen. Man startet 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 hier vorerst diese Version installiert, später wird dann ein entsprechendes Update vorgenommen.

Konfiguration

Nachdem der sog. TYPO3-Core heruntergeladen und entpackt wurde, kann man im Browser unter der Adresse <http://probst-maveq.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:

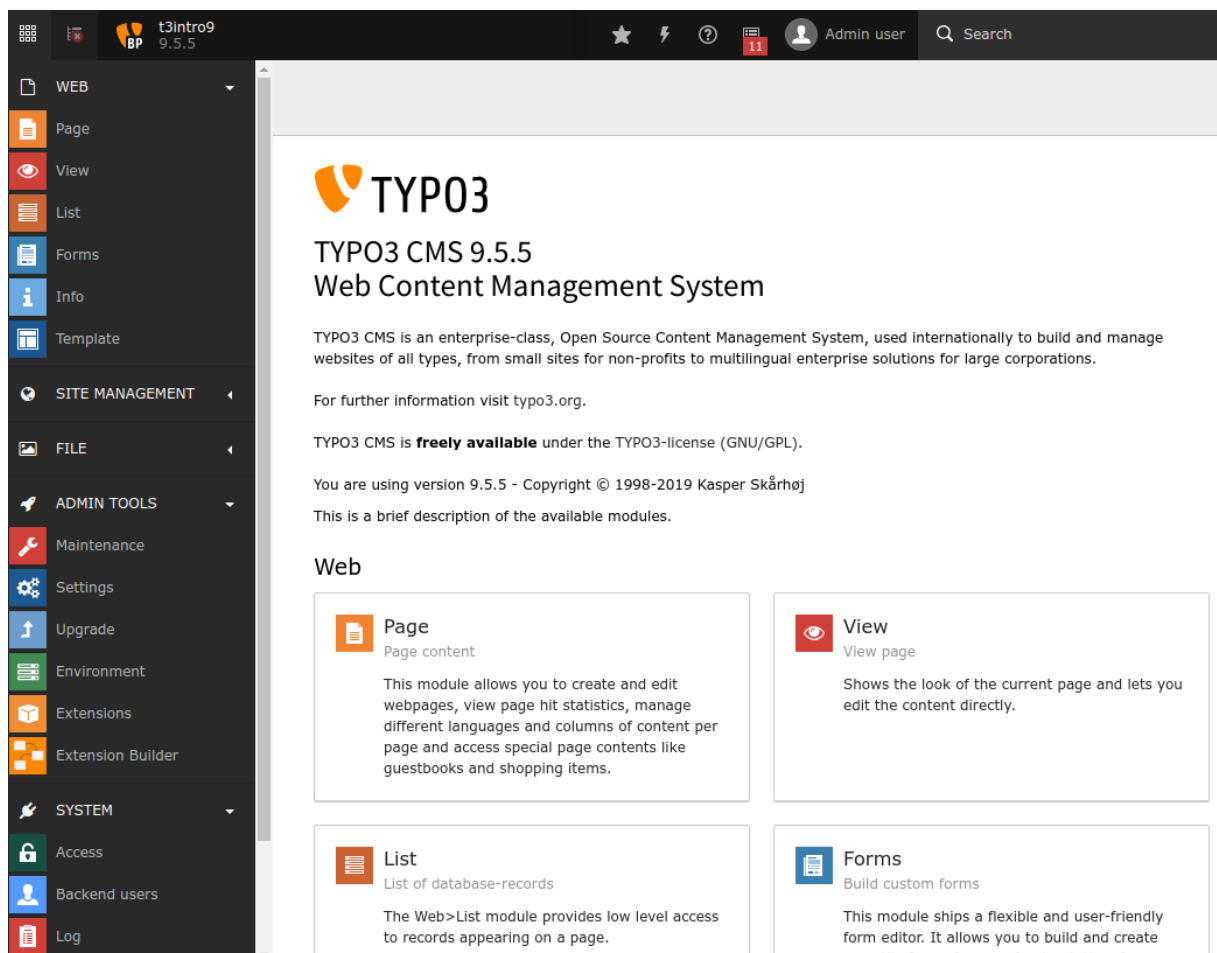


Abbildung 12, TYPO3-Backend

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 zur Ausgabe von Bildern zur Verfügung.
CKEditor Fontawesome	Bewirkt, dass im CKEditor ein Feld erscheint, mit dem Font-Awesome-Icons 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](#) für einen 404-Error werden hier definiert. Am wichtigsten ist aber die sog. [BaseURL](#), 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.

10.3 Datenbankkonzept (Nochmals prüfen)

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

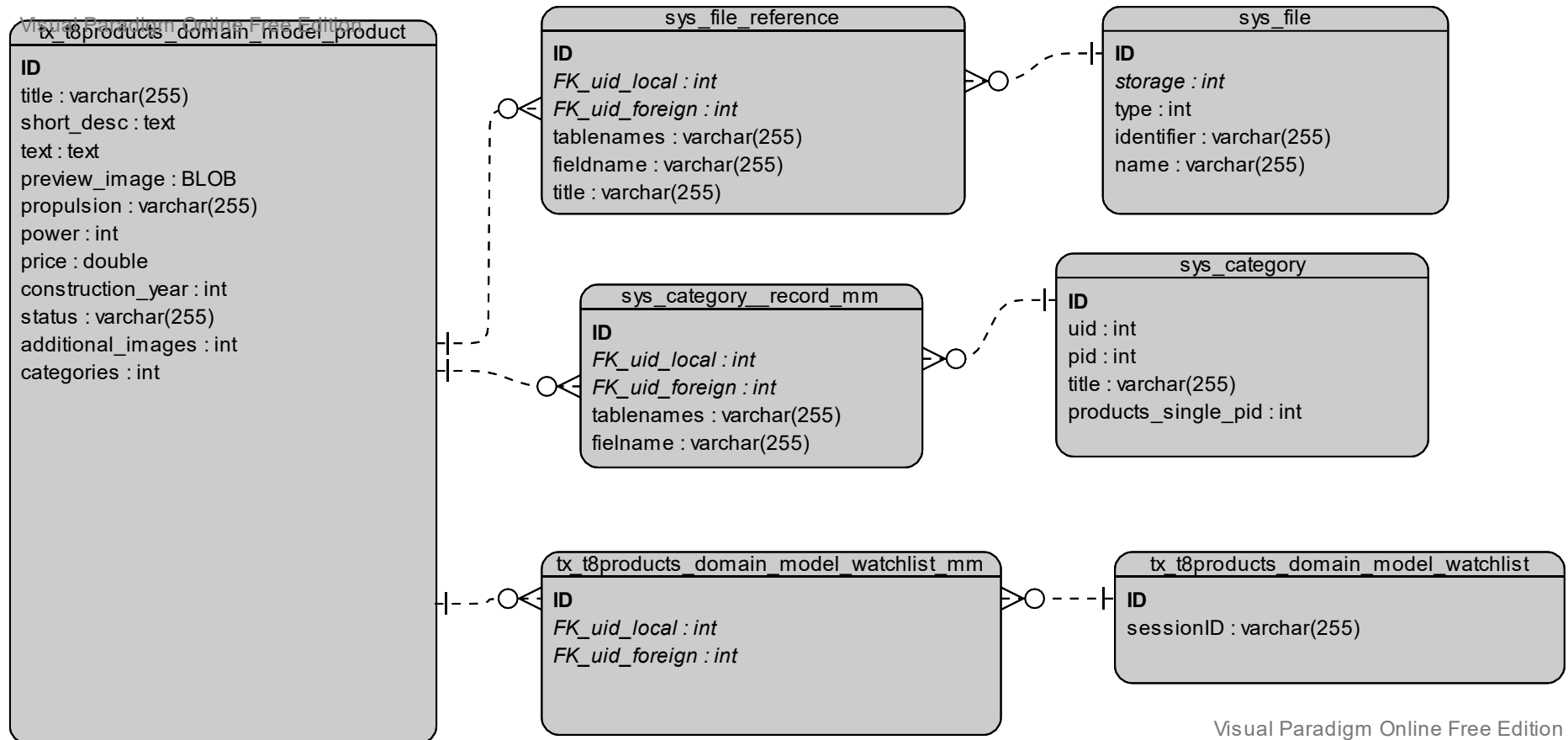


Abbildung 13, ERD

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

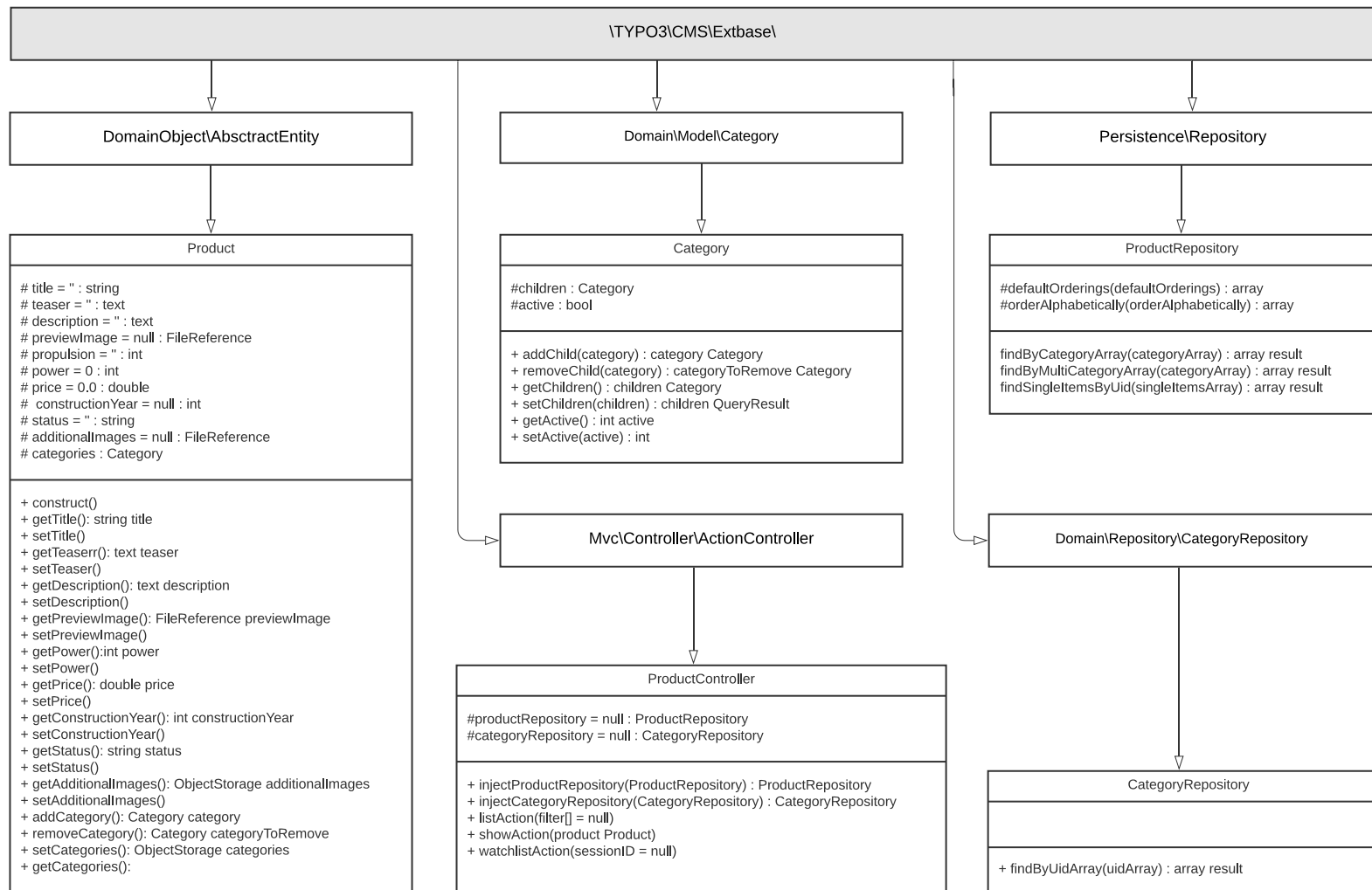


Abbildung 14, Klassendiagramm

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](#), 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.

10.5 Use-Cases

Um die umgesetzten Funktionalitäten testen zu können werden sog. Use-Cases erstellt. Untenstehend wurde ein Diagramm erstellt, das die verschiedenen Use-Cases 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 15, Use-Case-Diagramm

Tabelle 15, Use-Case 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.

10.5.1 Use-Case Beschreibungen

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

Tabelle 16, Use-Case 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, Use-Case 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	/

Tabelle 18, Use-Case 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 Unterseite 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, Use-Case 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, Use-Case 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-Menü 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, Use-Case 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, Use-Case 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, Use-Case 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, Use-Case 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	/

10.6 Activity-Diagramm

10.7 Architekturkonzept

Das Architekturkonzept beschreibt das Zusammenspiel zwischen den einzelnen Komponenten, in diesem Fall ist primär die Funktionsweise der MVC-Architektur gemeint, die von [ExtBase](#) 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.

10.7.1 Model – View – Controller

Die folgende Tabelle definiert einzelnen Komponenten anhand einer kurzen Erklärung:

Tabelle 25, MVC-Erläuterung

Name	Beschreibung und Funktionsweise
Model	Der Code für das Model befindet sich im Ordner «Classes/Model».
View	Fluid-Templates lassen sich im Ordner «Resources/Private» finden.
Controller	Der Controller von Extbase befindet sich im Ordner «Classes/Controller». ExtBase 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

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](#) wird an den Controller geschickt. Je nach Plugin wird eine andere Methode aufgerufen.
2. Der Controller ruft eine Methode des [Repository's](#) 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](#)».
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.
8. Zum Schluss wird das Fluid-Template «gerendert», soll heissen, die effektiven Daten werden in das HTML-Gerüst abgefüllt, anschliessend wird es dem Benutzer zugesandt.

Untenstehend wird der Ablauf eines solchen [Request's](#) zu Verständniszwecken nochmals grafisch abgebildet:

10.7.2 Listenansicht mit Filter

10.7.3 Detailansicht

10.7.4 Merkliste

10.8 Testkonzept

Das Testkonzept entspricht exakt den Use-Cases, die im Kapitel [10.5](#) definiert wurden. Aus Logiktechnischer Sicht werden alle Aspekte der Applikation abgedeckt, die unter Umständen in falschen Ergebnissen resultieren könnten. Ein Unit-Test wird ausgelassen, da sich die einzelnen Klassen nur schlecht isolieren lassen und die Funktionalitäten bereits bei der Entwicklung zu sehen sind. Allfällige Fehler im Programmcode fallen daher sofort auf und können direkt behoben werden.

Die Use-Case Tests werden zunächst von Drittpersonen anhand einer Anleitung 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](#) durch den Entwickler, um den Programmcode auf seine Richtigkeit zu überprüfen.

11. Realisierung

11.1 Domainmodel erstellen

11.2 Erweiterung installieren

11.4 Testprotokoll

Von Drittperson testen lassen

11.4.1 Testfall

Hinweis auf Testkonzept

ID / Bezeichnung	T-001	
Beschreibung		
Testvoraussetzung		
Testschritte		
Erwartetes Ergebnis		

11.4.2 Testdurchführung und Testergebnis

Tester	
Datum Testdurchführung	
Fehlerklasse (Testergebnis)	
Fehlerbeschreibung	

11.4.3 Testwürdigung

11.4.4 Testfazit

11.4.5 Weiteres Vorgehen

11.5 Einführung vorbereiten

11.6 Schutzmassnahmen umsetzen

12. Glossar

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

Tabelle 26, Glossar

Begriff	Bedeutung
AbstractWidgetViewHelper	TYPO3-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.
Request	Bei einem Request wird eine Aktion im Controller aufgerufen. Der Benutzer klickt also beispielsweise auf einen Filter, worauf die Controlleraktion «listAction()» aufgerufen wird.
ajaxWidgets	siehe «AbstractWidgetViewHelper»
BaseURL	Variable, die in TYPO3 dazu verwendet wird, das Frontend aufzurufen.
Visual Paradigm	Tool zum Erstellen von Datenbankmodellen
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 eines Websitebenutzers zu speichern.
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.
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-Queries)
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
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	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 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.
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 eines Websitebenutzers, wird aber beim Schliessen des Browsers gelöscht.
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.
Tajo	Interner Server von Terminal8, auf dem unsere Websites gehostet und verwaltet werden.
Testing	Testen von Software
Transition	Übergang, der mit CSS definiert werden kann. Beispielsweise bei anklicken eines Links, der dann die Farbe wechselt.
TYP03	Eine bestimmte Art eines CMS, vergleichbar mit Joomla oder WordPress, einfach komplexer.
TypoScript	TYP03-Eigene Konfigurationssprache.

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.
VS-Code	Tool von Microsoft zum Bearbeiten von Code, bietet viele Möglichkeiten zur Erweiterung.

13. 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

14. Tabellen- und Abbildungsverzeichnis

Abbildungsverzeichnis

Abbildung 1, Arbeitsplatz	7
Abbildung 2, Logo T8	7
Abbildung 3, Git Push	12
Abbildung 4, Git Pull	12
Abbildung 5, Phasenplanung	15
Abbildung 6, Meilensteinplanung	15
Abbildung 7, Modulplan nach Phasen	17
Abbildung 8, Projektrollen	18
Abbildung 9, Projektrollen	18
Abbildung 10, Projektrollen	18
Abbildung 11, Risikoportfolio	31
Abbildung 12, TYPO3-Backend	37
Abbildung 13, ERD	39
Abbildung 14, Klassendiagramm	40
Abbildung 15, Use-Case-Diagramm	41

Tabellenverzeichnis

Tabelle 1, Module	17
Tabelle 2, Projektrollen	18
Tabelle 3, Arbeitsjournal 07.04.2021	21
Tabelle 4, Arbeitsjournal 08.04.2021	22
Tabelle 5, Arbeitsjournal 09.04.2021	23
Tabelle 6, Arbeitsjournal 12.04.2021	24
Tabelle 7, Unterschriften Phasenfreigabe	27
Tabelle 8, Risikoanalyse	30
Tabelle 9: Risikoanalyse Tabelle	30
Tabelle 10, Varianten Filterfunktion	32
Tabelle 11, Nutzwertanalyse Filterfunktion	33
Tabelle 12, Varianten Merklste	34
Tabelle 13, Nutzwertanalyse Merklste	35
Tabelle 14, Benötigte Extensions	38
Tabelle 15, Use-Case Akteure	41
Tabelle 16, Use-Case 01	42
Tabelle 17, Use-Case 02	42
Tabelle 18, Use-Case 03	42
Tabelle 19, Use-Case 04	43
Tabelle 20, Use-Case 05	43
Tabelle 21, Use-Case 06	43
Tabelle 22, Use-Case 07	44

Tabelle 23, Use-Case 08.....	44
Tabelle 24, Use-Case 09.....	44
Tabelle 25, MVC-Erläuterung.....	45
Tabelle 26, Glossar	51
Tabelle 27, Literatur- und Quellenverzeichnis.....	55

15. Anhang

Code einfügen, Layout abklären