

Abschlussprüfung Summer 2022

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

**Automatisierte Anwendung**

**zum einseitigen Synchronisieren von internen Produkten in Akeneo über eine neu erstellte Middleware**

Abgabe Termin: TT:MM: YYYY

Prüfungsbewerber:

Maher Kassem Naser

Bayerisch Haibach 26

94032 Passau

****

Praktikumsbetrieb:

Auto-Leebmann GmbH

Terminer Str. 1

94036 Passau

Inhaltsverzeichnis

# **Einleitung**

Die folgende Projektdokumentation schildert den Ablauf des IHK-Abschlussprojektes, welches im Rahmen der Ausbildung zur Fachinformatiker Fachrichtung Anwendungsentwicklung durchgeführt wurde. Ausbildungsbetrieb ist die Auto-Leebmann GmbH, BMW-Vertragspartner und Betreiber eines führenden Online-Shops in Deutschland.

## **Projektbeschreibung**

Die Auto Leebmann GmbH, BMW-Vertragspartner und Betreiber eines führenden Online-Shops in Deutschland, möchte ihr Online-Shop auszubauen. Dazu wird ein Produkt-Informations-Management-System (PIM) eingeführt. Die Befüllung sowie die ständige einseitige Synchronisation sollen automatisiert werden. Es wird eine Headless-Middleware auf Basis Typescript erstellt. Diese fügt sich durch das Google-Coud-Function-Framwork optimal in die bestehende Struktur ein. Die Stammdaten liegen in einer internen MYSQL-Datenbank. Um die Daten in das bestehende PUB-SUB-System bei GCP (Google Cloud Platform) zu exportieren, wird ein internes Skript in PHP entwickelt. Auf dieses Event greift die oben genannte Middleware zu und verarbeitet den Payload. Dieser wiederum beinhaltet drei Ausprägungen (Create, Update, Delete).

Create: Artikel wird aus den Stammdaten des ERP erstellt. Hier muss der Primärschlüssel innerhalb des Akeneo erstellt werden. Dieser ist nicht Auto-Increment gekennzeichnet und ähnelt in der Darstellung einer UID.

Update: Schnell veränderte Daten (Preis, Bestand, …) werden anhand des Primärschlüssels aktualisiert

Delete: Artikel wird in Akeneo deaktiviert. Es wird nicht gelöscht da dies ein Löschen innerhalb des Shops als Folge ergeben würde. Dies ist im aktuellen Schritt nicht gewünscht.

## **Projektziel**

Das Ziel des Projektes war Entwicklung einer Middleware als Kommunikationsschnittselle zwischen dem ERP-System und Akeneo. Das Projekt das Importieren, Aktualisieren und Deaktivieren der internen MySQL-Daten von Leebmann in Akeneo automatisieren. Wenn Artikel in Datenbank nicht zur Verfügung stehen, soll es für Benutzer nicht sichtbar sein, aber in Akeneo deaktiviert sein. Außerdem soll das Projekt manuelle Arbeit ablösen.

## **Projektumfeld**

Praktikumsbetrieb ist der Auto-Leebmann GmbH, ein familiengeführtes Unternehmen mit etwa 220 Mitarbeitern und Mitarbeiterinnen und steht für zwei Automobilmarken (BMW und MINI) und vier Motorradmarken (BMW Motorrad, Honda, KTM, Yamaha). Der Firmensitz aller Unternehmen der Leebmann Gruppe ist in Passau. Neben neuen und gebrauchten Fahrzeugen wird an vier Standorten in Passau, Pocking und Hauzenberg eine breite Servicepalette rund um das Thema Mobilität angeboten. Seit 2010 werden Ersatzteile, Zubehör und Accessoires der Marken auch online vertrieben. Das Projekt wird innerhalb der IT-Abteilung von Auto-Leebmann in Zusammenarbeit mit Herrn Trefz durchgeführt.

## **Projektbegründung**

Die Artikelstammdaten können aktuell nicht aus dem ERP automatisch importiert, aktualisiert und deaktiviert werden. Auch Zeitlicher Aufwand ist groß, um die Daten manuell von Auto Leebmann in Akeneo zu importieren oder aktualisieren. Außerdem wenn Produkte nicht zur Verfügung stehen, bleiben sie trotzdem für Benutzer sichtbar und werden in Akeneo nicht deaktiviert. Aus genannten Gründen ist eine Automatisierung für die Synchronisation von internen Produkten in Akeneo erforderlich.

## **Projektschnittstellen**

## **Projektabgrenzung**

# **Projektplanung**

## **Projektphasen**

Für die Umsetzung des Projektes standen insgesamt 70 Stunden zur Verfügung. Die wurden vor Projektbeginn auf verschiedene verteilt, die während der Softwareentwicklung durchlaufen werden. Eine grobe Zeitplanung sowie die Hauptphasen lassen sich der Tabelle 1: Grobe Zeitplanungen entnehmen. Außerdem können die einzelnen Hauptphasen noch in kleinere Unterpunkte zerlegt werden. Eine detaillierte Übersicht dieser Phasen befindet sich im Anhang A.1: Detaillierte Zeitplanung auf S. .

|  |  |
| --- | --- |
| Projektphase | Geplante Zeit |
| Analysephase | 8 h |
| Entwurfsphase | 12 h |
| Implementierungsphase | 38 h |
| Abnahme und Einführung | 2 h |
| Erstellen der Dokumentation | 10 h |
| Gesamt | **70 h** |

Tabelle 1: Grobe Zeitplanung

## **Ressourcenplanung**

In der Übersicht, welche sich im Anhang A.2: Verwendete Ressourcen auf S. befindet, sind alle Ressourcen aufgelistet, die für das Projekt eingesetzt wurden. Damit sind sowohl Hard- und Softwareressourcen als auch das Personal gemeint. Bei der Auswahl der verwendeten Software wurde darauf geachtet, dass diese kostenfrei (z.B. als Open Source) zur Verfügung steht. Dadurch sollen anfallende Projektkosten möglichst geringgehalten werden.

## **Entwicklungsprozess**

Vor der Implementierung des Projektes musste ein geeigneter Entwicklungsprozess ausgewählt werden, um eine definierte Vorgehensweise bei der Entwicklung festzulegen. Dabei wurde sich in Absprache mit der Projektbetreuung für einen Spiralmodell entschieden, das dazu diente sicherzustellen, dass alle Anforderungen erfüllt werden. Das verkürzen der Zyklen hatte auch den Vorteil das auf Änderungswünsche schnell reagiert werden konnte.

# **Analysephase**

Um zu ermitteln, was sich die IT-Abteilung wünscht und wie der derzeitige Stand ist, wurde eine Analyse durchgeführt.

## **Ist-Analyse**

Erweiternd zu 1.3 (Projektbeschreibung) wird im Folgenden die Aktuelle Situation analysiert. Aktuell können die Artikelstammdaten aus dem ERP nur manuell importiert, aktualisiert und deaktiviert werden, dadurch ist die Zeitlicher Aufwand groß. Produkte bleiben für Benutzer sichtbar, auch wenn sie nicht zur Verfügung stehen und werden in Akeneo deaktiviert.

## **Wirtschaftlichkeitsanalyse**

Aufgrund der Probleme, die in den Abschnitten 1.4 (Projektbegründung) und 3.1 (Ist-Analyse) geschildert und erläutert wurden, ist die Umsetzung des Projektes erforderlich.

### **Make or Buy-Entscheidung**

Da es sich bei dem Projekt um Online-Shops der Auto-Leebmann handelt, lässt sich auf dem Markt keine Lösung finden, die diese Wünsche erfüllen könnte. Daher soll das Projekt in Eigenentwicklung durchgeführt werden.

### **Projektkosten**

Nachfolgend befinden sich die kalkulierten Kosten, die während der Entwicklung des Projektes anfielen. Dabei werden sowohl die Personal-, als auch die Hard- und Softwarekosten berücksichtigt.

## **Anwendungsfälle**

Als nächster Schritt wurden die notwendigen Anwendungsfälle analysiert. Dafür wurde als grobe Übersicht ein Use-Case-Diagramm erstellt, die alle anwendungsfälle enthält, die vom Anwender benötigt werden.

## **Lastenheft**

Am Ende der Analysephase wurde zusammen mit Projektleiter ein Lastenheft erstellt. dieses umfasst alle Anforderungen des Projektleiters an die erstellende Anwendung. Ein Auszug aus dem Lastenheft befindet sich im Anhang A. 4: Lastenheft (Auszug) auf S. iv.

# **Entwurfsanalyse**

Nach der Analyse und der Entscheidung des Projektes umzusetzen, wurde ein konkreter, technischer Entwurf der Anwendung erstellt. Außerdem wurde entschieden, welche Technologien sich am besten für die Umsetzung eignen. Wie dabei vorgegangen wurde, wird in diesem Kapitel beschrieben.

## **Zielplattform**

Die Entwicklung des gesamten Projekts wird vom Arbeitsplatz PC durchgeführt. Das Projekt wurde unter Verwendung des Google Functions-Frameworks für Node.js erstellt und als Entwicklungsumgebung wurde WebStorm verwendet. Zusätzlich findet für den firmeninternen Teil PHP als Programmiersprache mit der Entwicklungsumgebung PhpStorm seinen Platz. Das Projekt wird in einer lokalen, virtuellen Umgebung getestet

## **Architekturdesign**

Zur Umsetzung des Projektes wurde das node.js Framework Express genutzt. Dies bietet eine leichtgewichtige Grundlage für Webanwendungen und APIs, also genau richtig für die zu erstellende Programm. Express nutzt ein simples Routing Konzept, aufgesetzt auf einen Http-Websever. Es ermöglicht die Implementierung von Middleware, die auf die http Anfragen reagiert.

## **Schnittstellen**

## **Pflichtenheft**

Am Ende der Entwurfsphase wurde vom Entwickler ein Pflichtenheft erstellt. Dieses beschreibt wie die Anforderungen des Lastenhefts umgesetzt werden sollen. Er Entwickler verpflichtet sich alle im Pflichtenheft genannten Punkte zu realisieren, sodass es als roter Faden für die Implementierungsphase dienen kann. ein Auszug findet sich im Anhang A.4: Pflichtenheft (Auszug) auf Seite iii

# **Implementierungsphase**

## **Aufsetzen und Einrichten der benötigten Software**

## **Implementierung der Verbindung zur MYSQL-Datenbank**

## **Implementierung der Verbindung zur Akeneo PIM**

## **Implementierung der Middleware**

## **Rücksprache mit Projektleiter**

1. **Abnahme-** **und** **Einführungsphase**

# **Fazit**

## **Soll-/Ist-Vergleich**

Gestartet wurde das Projekt mit einer Idee. Gewissheit, dass Idee auch funktioniert, brachte erst der erste Test. Am Ende steht der Erfolg, dass Projektziel erreicht werden konnte.

In der Tabelle Soll/ Ist-Vergleich ist der benötigte Zeitaufwand gegenübergestellt. Es ist zu erkennen, dass es eine gewisse Zeitdifferenz bei der Entwicklung der Analysephase, Entwurfsphase sowie Implementierungsphase gegeben hat. Am Ende wurde das Projekt in den festgelegten 70 Stunden umgesetzt. Diese Differenzen sollten für zukünftige Projekte mit bedacht werden.

|  |  |  |  |
| --- | --- | --- | --- |
| Projektphase | Geplant | Tatsächlich | Differenz |
| Analysephase | 6 h | 8 h | +1 h |
| Entwurfsphase | 12 h | 11 h | -1 h |
| Implementierungsphase | 38 h | 38 h | 0 h |
| Abnahme | 1 h | 1 h | 0 h |
| Einführung | 1 h | 1 h | 0 h |
| Dokumentation | 10 h | 10 h | 0 h |
| Gesamt | **70** | **70** | **0 h** |

Tabelle 4: Soll-/Ist-Vergleich

## **Gewonnene Erkenntnisse**

Im Laufe des Projektes konnte ich viele Dinge im Bereich Projektmanagement, Language Understanding und Zyklus Softwareentwicklung lernen, aber auch neue Erfahrungen im Bereich Typescript, Node.js und PHP sammeln. Besonders die hinzugewonnenen Kenntnisse im Functions-Framework werden auch weiterhin sehr nützlich sein. Inhaltlich war es das erste Projekt in node.js.

Somit war das Projekt nicht nur für die IT-Abteilung der eine Entlastung, sondern auch für mich eine Bereicherung meiner Fähigkeiten und Kenntnisse. Für die Zukunft könnte ein größerer Fokus auf Entwurfsphase und vor allem auch das Festhalten in schriftlicher oder graphischer Form den späteren Entwicklungsprozess weiter vereinfachen.

## **Ausblick**

# **Anhang**

## **Detaillierte Zeitplanung**

|  |  |
| --- | --- |
| Analysephase | 8 h |
| 1. Ist-Analyse durchführen (Erstellen eines Use-Case-Diagramm) | 3 h |
| 1. Wirtschaftlichkeitsanalyse | 2 h |
| 1. Erstellen des Lastenheftes | 3 h |
| Entwurfsphase | **11 h** |
| 1. Zielplattform festlegen | 1 h |
| 1. Erstellen Programmablaufplan | 3 h |
| 1. Erstellen UML-Klassendiagramm | 3 h |
| 1. Erstellen des Pflichtenhefts | 4 h |
| Implementierungsphase | **38 h** |
| 1. Anlegen der node.js Projektstruktur | 2 h |
| 1. Implementierung der Verbindung zur internen MySQL-Datenbank | 1 h |
| 1. Implementierung von Router, Middleware | 22 h |
| 1. Implementierung von PUB/SUB-System | 7 h |
| 1. Testen von API | 3 h |
| 1. Rücksprache mit Projektleiter | 3 h |
| Abnahme und Einführung | **2 h** |
| 1. Abnahme | 1 h |
| 1. Einführung | 1 h |
| Erstellen der Dokumentation | **10 h** |
| 1. Erstellen der API Dokumentation | 2 h |
| 1. Erstellen der Projektdokumentation | 8 h |
|  |  |
| Gesamt | **70 h** |

## **Verwendete Ressourcen**

**Hardware**

* Büroarbeitsplatz mit Arbeitsplatzrechner
  + Tastatur
  + Maus
  + Zweiter Monitor

**Software**

* Jetbrains WebStorm 2021 – Entwicklungsumgebung
* Jetbrains PHPStorm 2021 – Entwicklungsumgebung
* Git und Github – Versionsverwaltung
* Linux Ubuntu 21.04
* Drawio
* Libereoffice

**Personal**

* Projektleiter (Festlegung der Anforderungen, Abnahme Projekt)

Alle oben aufgelisteten Ressourcen sind entweder bereits vorhanden oder unter einer freien Lizenz verfügbar, so dass nichts extra angeschafft werden muss.

## **Lastenheft (Auszug)**

Die Anwendung muss Folgende Anforderungen erfüllen:

* Die Anwendung muss Auto-Leebmann-Artikeln aus der Datenbank einlesen und PubSub veröffentlichen.
* Die Anwendung muss die veröffentliche Nachrichten empfangen und in Akeneo importieren.

## **Use-Case-Diagramm**

## **Programmablaufplan**

## **UML-Klassendiagramm**

## **Iterationsplan**

## **Listing der Klasse Datenbank-Verbindung**

## **Listing der Klasse Akeneo**