Inhaltsverzeichnis

[1.Einleitung 3](#__RefHeading___Toc1929_116211150)

[1.1. Projektumfeld 3](#__RefHeading__413_861071704)

[1.2. Projektziel 3](#__RefHeading___Toc1141_50268823)

[1.3. Projektbegründung 3](#__RefHeading___Toc1143_50268823)

[1.4. Projektschnittstellen 3](#__RefHeading___Toc1151_50268823)

[1.5. Projektabgrenzung 3](#__RefHeading___Toc1145_50268823)

[2.Projektplanung 4](#__RefHeading___Toc1931_116211150)

[2.1. Projektphasen 4](#__RefHeading___Toc1206_50268823)

[2.2. Abweichungen vom Projektantrag 4](#__RefHeading___Toc1208_50268823)

[2.3. Ressourcenplanung 4](#__RefHeading___Toc1210_50268823)

[2.4. Entwicklungsprozess 4](#__RefHeading___Toc1212_50268823)

[3.Analysephase 5](#__RefHeading___Toc1933_116211150)

[3.1. Ist-Analyse 5](#__RefHeading___Toc1216_50268823)

[3.2. Wirtschaftlichkeitsanalyse 5](#__RefHeading___Toc1218_50268823)

[3.2.1 „Make or Buy“-Entscheidung 5](#__RefHeading___Toc1220_50268823)

[3.2.2 Projektkosten 5](#__RefHeading___Toc1222_50268823)

[3.3.Anwendungsfälle 6](#__RefHeading___Toc1935_116211150)

[3.4. Qualitätsanforderungen 6](#__RefHeading___Toc1230_50268823)

[3.5. Lastenheft/Fachkonzept 7](#__RefHeading___Toc1232_50268823)

[4. Entwurfsphase 7](#__RefHeading___Toc2054_167878211)

[4.1. Zielplattform 7](#__RefHeading___Toc2056_167878211)

[4.2. Architekturdesign 7](#__RefHeading___Toc2058_167878211)

[4.3. Entwurf der Benutzeroberfläche 7](#__RefHeading___Toc2060_167878211)

[4.4. Maßnahmen zur Qualitätssicherung 8](#__RefHeading___Toc2062_167878211)

[4.5. Zwischenstand 8](#__RefHeading___Toc2064_167878211)

[5. Implementierungsphase 8](#__RefHeading___Toc2066_167878211)

[5.1. Implementierung der Datenstrukturen 8](#__RefHeading___Toc2068_167878211)

[5.2. Implementierung der Benutzeroberfläche 8](#__RefHeading___Toc2070_167878211)

[5.3. Implementierung der Geschäftslogik 9](#__RefHeading___Toc2072_167878211)

[5.4. Zwischenstand 9](#__RefHeading___Toc1238_50268823)

[6. Dokumentation 9](#__RefHeading___Toc1244_50268823)

[7. Fazit 10](#__RefHeading___Toc2074_167878211)

[7.1. Soll-/-Ist-Vergleich 10](#__RefHeading___Toc2076_167878211)

[7.1.1 Zielsystem 10](#__RefHeading___Toc2078_167878211)

[7.1.2 Frontend 10](#__RefHeading___Toc2080_167878211)

[7.1.3 Geschäftslogik 10](#__RefHeading___Toc2082_167878211)

[7.2. Ausblick 10](#__RefHeading___Toc2084_167878211)

[7.2.1 Maßnahmen zur Qualitätssicherung 10](#__RefHeading___Toc2086_167878211)

[7.2.2 Frontend 10](#__RefHeading___Toc2088_167878211)

[8.Anhänge 11](#__RefHeading___Toc1937_116211150)

[8.1. Anhangverzeichnis 11](#__RefHeading___Toc1939_116211150)

[8.2. Anhang 1 – Aktivitätenplan 11](#__RefHeading___Toc1941_116211150)

[8.3.Anhang 2 – Pflichtenheft 12](#__RefHeading___Toc1943_116211150)

[8.4. Zielbestimmung 14](#__RefHeading__413_8610717041)

[8.5. Musskriterien 14](#__RefHeading__415_861071704)

[8.6. Wunschkriterien 14](#__RefHeading__417_861071704)

[8.7. Abgrenzungskriterien 14](#__RefHeading__419_861071704)

[9. Produkteinsatz 15](#__RefHeading__421_861071704)

[9.1. Anwendungsbereiche 15](#__RefHeading__423_861071704)

[9.2. Zielgruppen 15](#__RefHeading__425_861071704)

[9.3. Betriebsbedingungen 15](#__RefHeading__427_861071704)

[10. Produktübersicht 15](#__RefHeading__429_861071704)

[11. Produktfunktionen 16](#__RefHeading__431_861071704)

[12. Produktdaten 16](#__RefHeading__433_861071704)

[13. Produktleistungen 16](#__RefHeading__435_861071704)

[14. Qualitätsanforderungen 16](#__RefHeading__437_861071704)

[15. Benutzungsoberfläche 16](#__RefHeading__439_861071704)

[16. Nicht-funktionale Anforderungen 16](#__RefHeading__441_861071704)

[17. Technische Produktumgebung 17](#__RefHeading__443_861071704)

[17.1. Software 17](#__RefHeading__445_861071704)

[17.2. Hardware 17](#__RefHeading__447_861071704)

[17.3. Orgware 17](#__RefHeading__449_861071704)

[17.4. Produkt-Schnittstellen 17](#__RefHeading__451_861071704)

[18. Spezielle Anforderungen an die Entwicklungs-Umgebung 17](#__RefHeading__453_861071704)

[19.Anhang 3 - DV-Konzept 18](#__RefHeading___Toc3308_116211150)

[Basis/Symfony 19](#__RefHeading__413_8610717042)

[Symfony 19](#__RefHeading___Toc941_1141482830)

[Basisinstallation 19](#__RefHeading___Toc943_1141482830)

[Routing 19](#__RefHeading___Toc945_1141482830)

[Frontend 20](#__RefHeading___Toc947_1141482830)

[19.1.Prozessbeschreibung 20](#__RefHeading___Toc3310_116211150)

[19.2.Mögliche Fehlerquellen 20](#__RefHeading___Toc3312_116211150)

[Backend 20](#__RefHeading___Toc949_1141482830)

[Prozessbeschreibung 20](#__RefHeading___Toc951_1141482830)

[Mögliche Fehlerquellen 20](#__RefHeading___Toc3314_116211150)

[Datenbank 21](#__RefHeading___Toc955_1141482830)

[Technische Basis 21](#__RefHeading___Toc957_1141482830)

[Datenbankmodell 21](#__RefHeading___Toc3316_116211150)

# 

# Einleitung

## Projektumfeld

Auftraggeber des Projektes ist der Einzelhandel „Tante Emma“. Ein Kiosk welcher einen Lieferservice anbietet. Dieser soll durch das Produkt verwaltet werden

## Projektziel

Projektziel ist die Entwicklung einer Anwendung zur Verwaltung der Geschäftsprozesse des Einzelhandels. Dazu gehören die Verwaltung und Erfassung von Kunden und Mitarbeitern, das Durchführen von Geschäften und Anlieferungen und die Verwaltung des Sortiments und der aktuellen Bestände.

## Projektbegründung

Durch das Projekt erhofft sich das Unternehmen die Ablösung der bisher papierbasierten Verwaltung. Daher können Verkäufe leichter im Überblick behalten und auch die Bestände einfacher verwaltet werden können. Dies sorgt für schnellere Abläufe und weniger Fehler im Betrieb

## Projektschnittstellen

Benutzer der Anwendung sind die Angestellten des Einzelhandeluntermehmens, sowie der Geschäftsführerin. Ihr wird auch das Projekt am Ende der Entwicklung vorgestellt.

## Projektabgrenzung

Das Projekt soll explizit nicht dazu dienen buchhalterische Aufgaben zu erfüllen. Auch steht die Kundenintegrierung in das System nicht zur Debatte. Es handelt sich um eine interne Anwendung welche nur von den Mitarbeitern des Unternehmens genutzt wird.

# Projektplanung

## Projektphasen

Das Projekt muss in 5 Terminen mit 4 Zeiteinheiten je 45 Minuten durchgeführt werden. Zusätzlich dazu kommen eventuelle außerordentliche Arbeiten welche die Mitarbeiter selbst wählen können.

Diese Zeiteinheiten wurden wie folgt eingeteilt:

|  |  |
| --- | --- |
| Phase | Zeiteinheiten (1ZE ≙ 45 min) |
| Analysephase | 2 |
| Konzeptphase | 20 |
| Implementierungsphase | 42 |
| Summe | 64 |

## Abweichungen vom Projektantrag

Es gab nur geringfügige Abweichungen vom Projektantrag. Diese waren der Usability und technischen Hintergründen geschuldet. So wurde zum Beispiel eine Möglichkeit gegeben Anlieferungen zu erfassen welche im Initialkonzept nicht erwähnt wurden

## Ressourcenplanung

Die benötigten Ressourcen sind ein Rechner pro Entwickler. Idealerweise mit dem Betriebssystem Ubuntu und entsprechenden Administrationsrechten zum installieren der Pakete.

Zudem ist eine Internetanbindung erforderlich um die Paketquellen zu installieren und um Zugriff auf das GitHub Repository zu haben.

Eine detaillierte Zeitplanung findet sich im Aktivitätenplan ([Anhang 1](#_toc538))

## Entwicklungsprozess

Das Projekt wurde nach einem einfachen Wasserfallmodell durchgeführt wobei beachtet werden muss, dass generell nach den Phasen vorgegangen wurde. Nach Abschluss der Konzeptionsphase und Enrichtung der Basis gab es kaum Abhängigkeiten zwischen den einzelnen Modulen wodurch eine stark verzahnte Entwicklung entstanden ist

# Analysephase

## Ist-Analyse

Der momentane Stand ist eine papierbasierte Verwaltung. Es kommt häufig zu Fehlern und eine Überwachung der Prozesse ist fast unmöglich. Ein modernes IT-System ist nicht vorhanden.

Daher ist die Anforderung das erstellen eines neuen Systems von Grund auf.

## Wirtschaftlichkeitsanalyse

### „Make or Buy“-Entscheidung

Da alle vorhandenen Lösungen zu teuer, zu umfangreich oder nicht genug abgestimmt sind wurde sich dazu entschieden eine angepasste lightweight Lösung zu implementieren. Marktlösungen beinhalten normalerweise große Mengen an nicht benötigten Funktionen und kosten Unsummen in der Anschaffung.

### Projektkosten

Arbeitsstunden: 64 ZE \* 45 min/ZE = 1,42 / 60 = 48 Stunden

Stunden pro Jahr = 8h/Tag \* 220 Tage/Jahr = 1760 h/Jahr

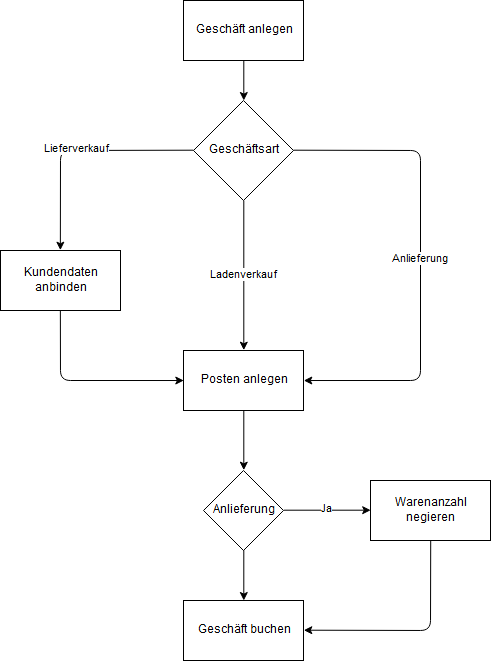
Kosten pro Jahr = 910€ \* 13,3 Monate/Jahr = 12103€

Kosten pro Stunde:

Pauschalkosten Ressourcen: 15€/h

Personalkosten:

## Anwendungsfälle



## Qualitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Sehr wichtig | wichtig | Weniger wichtig | Unwichtig |
| Robustheit |  |  | x |  |
| Zuverlässigkeit |  | x |  |  |
| Korrektheit | x |  |  |  |
| Benutzer-  freundlichkeit |  | x |  |  |
| Effizienz |  |  |  | x |
| Portierbarkeit |  |  | x |  |
| Kompatibilität |  |  | x |  |

## Lastenheft/Fachkonzept

Siehe [Anhang 2](#_toc543) - Pflichtenheft

# Entwurfsphase

## Zielplattform

Software soll auf Windows 7 Schulrechner (XAMP) laufen.

Beötigt wird php 5.4 oder neuer und eine MYSQL Datenbank.

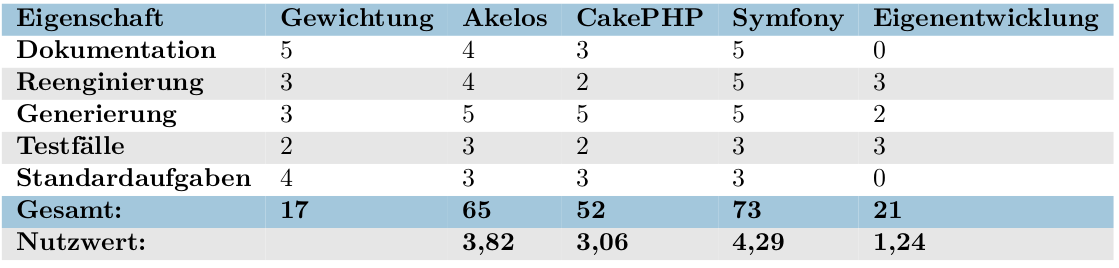
Dies ist eine vorgabe vom Auftraggeber.

## Architekturdesign

Die Webseite läuft nach dem MVC (Model-View-Controller) Prinzip.

Dies ist eine einfaches gut Strukturierbares Prinzip, wo Kompetenzen gut verteilt werden können.

Wir setzen Komplet auf das Symfony PHP Framework dies bietet Diverse Features und nimmt ein sehr vieles ab (Siehe DV-Konzept)

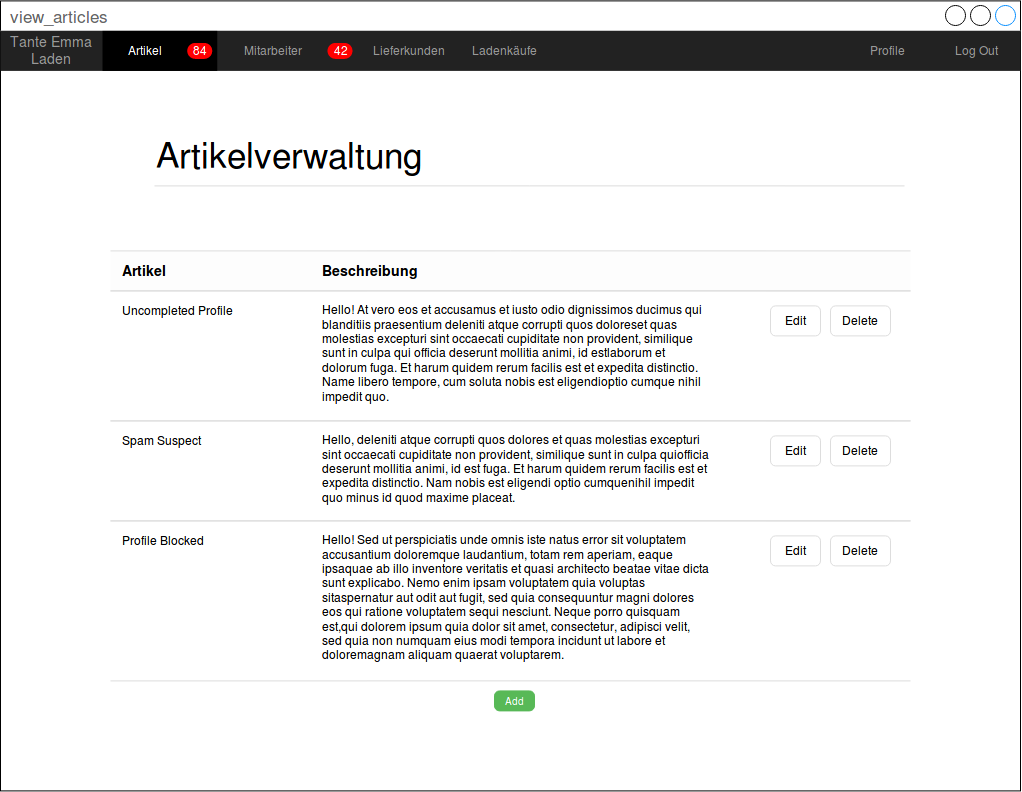
– Quelle: <https://fachinformatiker-anwendungsentwicklung.net/vorlage-fuer-die-projektdokumentation/>

## Entwurf der Benutzeroberfläche

Wir haben uns für ein Webfrontend entschieden einfach dieses auf jeden Device einfach aufzurufen ist und keine Installation nötig ist.

Wir wollen ein modernes einfaches Design das für jeden Bestanteil übersichtlich die crud‘s anbietet

Hier ein Beispiel:



## Maßnahmen zur Qualitätssicherung

Symfony bietet diverse Möglichkeiten um Backend und Frontend zu testen.

Auf grund der geringen Zeit und Komplexität wurden aber in der Entwurfsphase keine Maßnamen ergriffen.

Dies soll bei in der eigenverantwortlichen Abnahme nachgeholt werden.

## Zwischenstand

Konzeption fertig unstimigkeiten bei der Zielplattform aufgrund von kompatibilitätsproblemen

# Implementierungsphase

## Implementierung der Datenstrukturen

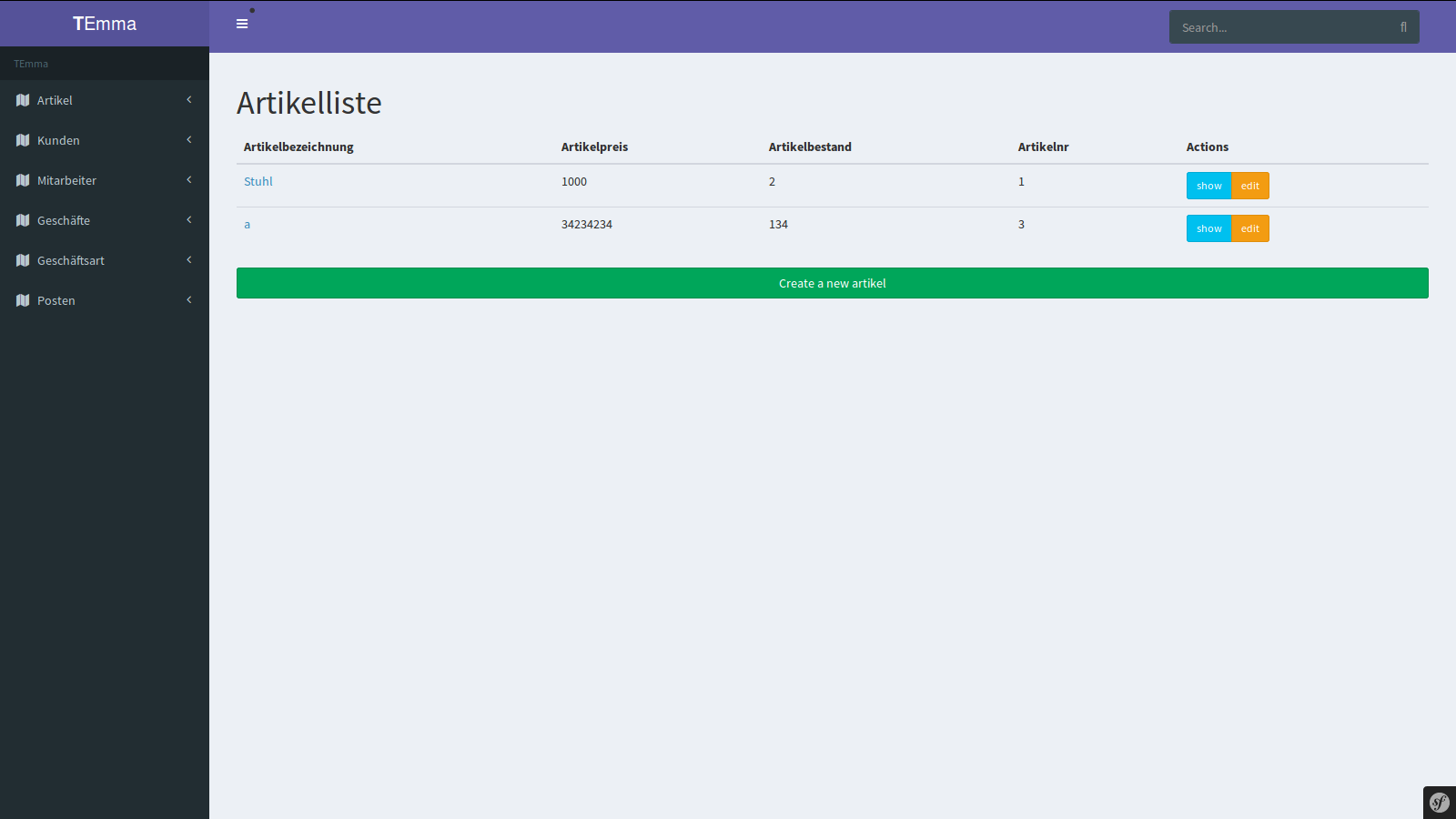
Datenbank wird aus SQL Schema händisch angelegt und dann über Doctrine in XML-Schmas übertragen.

Automatische Generierung von cruds (create/read/update/delete) (Siehe DV-Konzept).

## Implementierung der Benutzeroberfläche

Optmierung von Doctrine Generierten Twig Templates.

Implementierung von Style-Template Admin-LTE und CSS Framework Bootstrap.

– Aktueller Stand des Frontends

## Implementierung der Geschäftslogik

Anpassen von Generierten PHP Controllern und der crud Methoden.

Ermöglichen von Login.

## Zwischenstand

Generierte cruds machen einige Probleme und benötigte an meisten Arbeit.



# Dokumentation

Siehe Anhang

# Fazit

## Soll-/-Ist-Vergleich

### Zielsystem

Das Zielsystem (Windows 7 Schulrechner) hat sich als eine zu große Herrausfoderung heraus gestellt

Projekt läuft nun auf einem Laptop mit Ubuntu 16.04.

### Frontend

|  |  |
| --- | --- |
| Entwurf | Ergebnis |
|  |  |

### Geschäftslogik

Das Konzept wurde voll eingehalt wir haben das Symfony Framework mit dem vorgesehenden MVC Prinzip genutzt.

## Ausblick

### Maßnahmen zur Qualitätssicherung

Die Maßnahmen zur Qualitätssicherung müssen in der selbst verantwortlichen Abnahme erfolgen.

### Frontend

Einige Formulare müssen noch erweitert werden.

# Anhänge

## Anhangverzeichnis

* Anhang 1 – Aktivitätenplan
* Anhang 2 – Pflichtenheft
* Anhang 3 – DV-Konzept

## Anhang 1 – Aktivitätenplan



## Anhang 2 – Pflichtenheft

**Pflichtenheft:**

Gruppe:

Gruppenmitglieder: Christian Ziermann, Marvin Kröker, Dennis Wrobel, Lukas Prediger, Joshua Failing

**Auftrag:**

Inhaltsverzeichnis

[1 Zielbestimmung 3](#__RefHeading__413_861071704)

[1.1 Musskriterien 3](#__RefHeading__415_861071704)

[1.2 Wunschkriterien 3](#__RefHeading__417_861071704)

[1. 3 Abgrenzungskriterien 3](#__RefHeading__419_861071704)

[2 Produkteinsatz 4](#__RefHeading__421_861071704)

[2.1 Anwendungsbereiche 4](#__RefHeading__423_861071704)

[2.2 Zielgruppen 4](#__RefHeading__425_861071704)

[2.3 Betriebsbedingungen 4](#__RefHeading__427_861071704)

[3 Produktübersicht 4](#__RefHeading__429_861071704)

[4 Produktfunktionen 5](#__RefHeading__431_861071704)

[5 Produktdaten 5](#__RefHeading__433_861071704)

[6 Produktleistungen 5](#__RefHeading__435_861071704)

[7 Qualitätsanforderungen 5](#__RefHeading__437_861071704)

[8 Benutzungsoberfläche 5](#__RefHeading__439_861071704)

[9 Nicht-funktionale Anforderungen 5](#__RefHeading__441_861071704)

[10 Technische Produktumgebung 6](#__RefHeading__443_861071704)

[10.1 Software 6](#__RefHeading__445_861071704)

[10.2 Hardware 6](#__RefHeading__447_861071704)

[Ein funktionsfähiger Webserver mit Zugriff ins Netzwerk sowie Terminalrechner mit einem aktuellen Webbrowser zum Beispiel Firefox oder Google Chrome. 6](#__RefHeading___Toc1113_1770382683)

[10.3 Orgware 6](#__RefHeading__449_861071704)

[10.4 Produkt-Schnittstellen 6](#__RefHeading__451_861071704)

[11 Spezielle Anforderungen an die Entwicklungs-Umgebung 6](#__RefHeading__453_861071704)

## Zielbestimmung

Die Webanwendung TEmma soll die Lagerverwaltung, Verkaufs- und Lieferverwaltung eines Geschäfts zur Verfügung stellen. Im folgenden Bezeichnet Mitarbeiter einen männlichen oder weiblichen Angestellten des Geschäfts und Kunde einen männlichen oder weiblichen Kunden des Geschäfts. Aus Gründen besserer Lesbarkeit wurde auf die explizite Geschlechtsform verzichtet.

## Musskriterien

* Lagerverwaltung
  + Artikel mit Bezeichnung und Menge erfasst werden können
  + Artikel müssen bei direktem Verkauf im Lager aus dem System ausgebucht werden können
* Lieferverwaltung
  + Es müssen Bestellungen registrierter Kunden erfasst und ausgeführt werden können
* Kundenverwaltung
  + Kunden müssen im System erfasst werden können
    - Name
    - Telefonnummer
    - Lieferadresse
    - Lieferhinweis
* Mitarbeiterverwaltung
  + Ein Mitarbeiter muss sich im System hinterlegt werden können
    - Name
    - Telefonnummer
    - Adresse
  + Mitarbeiter müssen sich im System einloggen können (Siehe Berechtigungsverwaltung)
* Berechtigungsverwaltung
  + Jedem Mitarbeiter müssen unabhängige Berechtigungen zugeordnet werden können
  + Die Berechtigungen müssen bei versuchten Änderungen und Aufrufen bestimmter Daten überprüft werden

## Wunschkriterien

* Lokalisierung des Programms
  + Hauptsprache Deutsch
  + Geplant: Englisch
* Periodisches wiederholen von Lieferaufträgen
* Statusverwaltung für Lieferaufträge
* Dynamische Berechtigungsverwaltung

## Abgrenzungskriterien

* Keine Buchungssystem

# Produkteinsatz

## Anwendungsbereiche

Die Anwendungsbereiche sind Kleinhandelsbetriebe mit einer Option für einen Lieferbetrieb (nicht erforderlich). Die Anwendung richtet sich an Firmenkunden, bzw. private Unternehmer.

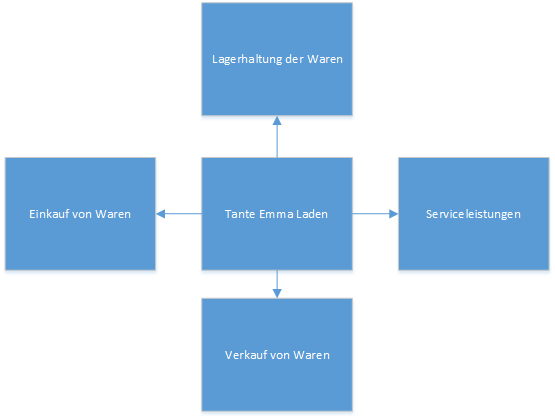
## Zielgruppen

Die Zielgruppe sind Mitarbeiter der oben genannten Unternehmen. Dabei sind Basiskenntnisse im Umgang mit Computern vonnöten.

## Betriebsbedingungen

Die physikalische Umgebung des Systems ist für das Warenlager, sowie den Verkaufsraum und für das Büro vorgesehen.Es sind überall die gleichen Basiskenntnisse empfehlenswert. Die Betriebszeit erstreckt sich über die Öffnungszeiten des Ladens die je nach Wochentag unterschiedlich sein können. Eine ständige Aufsicht des Systems ist nicht notwendig.

# Produktübersicht



# Produktfunktionen

Hauptfunktion des Produktes ist die Verwaltung von Geschäften. Ein Geschäft ist hierbei die Änderung des Bestands von einem oder mehrerer Artikel. Zusätzlich können Artikel und deren Bestände verwaltet sowie Mitarbeiter und Kundendaten anlegen und verwalten.

Ein Mitarbeiter autorisiert sich mit seinem Kürzel sowie dem hinterlegten Passwort.

# Produktdaten

Kundendaten bestehen aus Kundennummer,Name, Adresse, Kommunikationsdaten, Geburtsdatum .

Lagerdaten bestehen aus Warennummer, Bestand, Lagerort, Preis. Mitarbeiterdaten bestehen aus Name, Telefonnummer und Adresse und Berechtigungen.

# Produktleistungen

Einkäufe durch Ladenkunden oder Lieferkunden müssen zeitnah im System verzeichnet werden bzw. der Warenbestand aktualisiert werden.

# Qualitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Sehr wichtig | wichtig | Weniger wichtig | Unwichtig |
| Robustheit |  |  | x |  |
| Zuverlässigkeit |  | x |  |  |
| Korrektheit | x |  |  |  |
| Benutzer-  freundlichkeit |  | x |  |  |
| Effizienz |  |  |  | x |
| Portierbarkeit |  |  | x |  |
| Kompatibilität |  |  | x |  |

# Benutzungsoberfläche

Die Benutzeroberfläche ist Maus/Tastaturgesteuert und ist eine Fensterarchitektur im Browser.

# Nicht-funktionale Anforderungen

Die Ordnungsgemäße Buchung ist einzuhalten. Das Produkt sollte auf der Plattform Windows laufen.

Es werden alle Anforderungen aufgeführt, die sich nicht auf die Funktionalität, die Leistung und die Benutzungsoberfläche beziehen, z.B. einzuhaltende Gesetze einzuhaltende Normen Testat durch externe Prüfungsgesellschaft Revisionsfähigkeit

Ordnungsmäßigkeit der Buchführung

Sicherheitsanforderungen, z.B. Passwortschutz, Mitlaufen von Protokollen, sichere Übertragung

Plattformabhängigkeiten

# Technische Produktumgebung

## Software

Auf dem Server muss folgende Software installiert werden:

* MySQL-Server
* PHP Minimum Version: 5
* NPM
  + GULP + Erweiterung
  + JQuery
  + Webpack
* Composer
  + Symfony Full-Stack Framework
  + Doctrine
  + Twig

Auf dem Client muss ein aktueller Webbrowser installiert werden.

## Hardware

Ein funktionsfähiger Webserver mit Zugriff ins Netzwerk sowie Terminalrechner mit einem aktuellen Webbrowser zum Beispiel Firefox oder Google Chrome.

## Orgware

Zur Verwaltung von Dokumenten und Sourcecode wird die Software Git über die Plattform GitHub genutzt

## Produkt-Schnittstellen

Es sind keine Anwendungschnittstellen geplant

# Spezielle Anforderungen an die Entwicklungs-Umgebung

Es gibt keine speziellen Anforderungen für die Entwicklungsumgebung

# 

# Anhang 3 - DV-Konzept

**DV-Konzept:**

Gruppe:

Gruppenmitglieder: Christian Ziermann, Marvin Kröker, Dennis Wrobel, Lukas Prediger

**Auftrag:**

[1. 2](#_Toc484617268)

[2. Basis/Symfony 2](#_Toc484617269)

[2.1. Symfony 2](#_Toc484617270)

[2.2. Basisinstallation 2](#_Toc484617271)

[2.3. Routing 2](#_Toc484617272)

[3. Frontend 3](#_Toc484617273)

[4. Backend 4](#_Toc484617274)

[4.1. Prozessbeschreibung 4](#_Toc484617275)

[4.2. Mögliche Fehlerquellen 4](#_Toc484617276)

[5. 5](#_Toc484617277)

[6. Datenbank 5](#_Toc484617278)

[6.1. Technische Basis 5](#_Toc484617279)

[6.2. Datenbankmodell 6](#_Toc484617280)

Basis/Symfony

Symfony

Wir nutzen das full-stack PHP Framework Symfony.

« Symfony is a set of PHP Components, a Web Application framework, a Philosophy, and a Community — all working together in harmony. »

Basisinstallation

Installiert wird Symfony über den Packetmanager Composer.

Routing

Das Routing läuft nach dem MVC (Model View Controller) Prinzip.

Und wird in YML-Datein Konfiguriert.

Frontend

## Prozessbeschreibung

Technologien:

* Bootstrap(CSS Templates)
* AdminLTE(WebApp Template)

Das komplette Frontend, bis auf die Navbar, wird mit Bootstraps CSS-Klassen bestückt, um die Webseite sauber und responsive zu gestalten.

Die Navbar wird mit AdminLTE erstellt.

## Mögliche Fehlerquellen

Flüchtigkeitsfehler aus Zeitdruck.

Backend

Prozessbeschreibung

Technologien:

* Doctrine (Framework das eine Objektrelationalen Abbildung und eine Datenbankabstraktionsschicht bietet)
* Twig (PHP Template-engine

Die Datenbank wird über eine SQL-Datei Angelegt.

Doctrine erstellt ein XML-Schema aus der Datenbank.

Aus diesem Schema können CRUDs generiert werden, daraus wird folgendes erzeugt:

* PHP Controller mit CRUD-Methoden
* Twig Templates pro Entity pro CRUD
* PHP Repository
* PHP Entity

Mögliche Fehlerquellen

* Fehler durch fehlende Kenntnisse (insbesondere Symfony)

Datenbank

Technische Basis

Als DBMS Basis dient eine MySQL Datenbank.

Datenbankmodell

