



## Inhaltsverzeichnis

1. Einleitung.....	3
1.1. Projektumfeld.....	3
1.2. Projektziel.....	3
1.3. Projektbegründung.....	3
1.4. Projektschnittstellen.....	3
1.5. Projektabgrenzung.....	3
2. Projektplanung.....	4
2.1. Projektphasen.....	4
2.2. Abweichungen vom Projektantrag.....	4
2.3. Ressourcenplanung.....	4
2.4. Entwicklungsprozess.....	4
3. Analysephase.....	5
3.1. Ist-Analyse.....	5
3.2. Wirtschaftlichkeitsanalyse.....	5
3.2.1 „Make or Buy“-Entscheidung.....	5
3.2.2 Projektkosten.....	5
3.3. Anwendungsfälle.....	6
3.4. Qualitätsanforderungen.....	6
3.5. Lastenheft/Fachkonzept.....	7
4. Entwurfsphase.....	7
4.1. Zielplattform.....	7
4.2. Architekturdesign.....	7
4.3. Entwurf der Benutzeroberfläche.....	7
4.4. Maßnahmen zur Qualitätssicherung.....	8
4.5. Zwischenstand.....	8
5. Implementierungsphase.....	8
5.1. Implementierung der Datenstrukturen.....	8
5.2. Implementierung der Benutzeroberfläche.....	8
5.3. Implementierung der Geschäftslogik.....	9
5.4. Zwischenstand.....	9
6. Dokumentation.....	9
7. Fazit.....	10
7.1. Soll-/Ist-Vergleich.....	10
7.1.1 Zielsystem.....	10
7.1.2 Frontend.....	10
7.1.3 Geschäftslogik.....	10
7.2. Ausblick.....	10
7.2.1 Maßnahmen zur Qualitätssicherung.....	10
7.2.2 Frontend.....	10
8. Anhänge.....	11
8.1. Anhangverzeichnis.....	11
8.2. Anhang 1 – Aktivitätenplan.....	11
8.3. Anhang 2 – Pflichtenheft.....	12
8.4. Zielbestimmung.....	14
8.5. Musskriterien.....	14
8.6. Wunschkriterien.....	14
8.7. Abgrenzungskriterien.....	14
9. Produkteinsatz.....	15
9.1. Anwendungsbereiche.....	15
9.2. Zielgruppen.....	15
9.3. Betriebsbedingungen.....	15
10. Produktübersicht.....	15
11. Produktfunktionen.....	16

	Fach: <b>ANWENDUNGSENTWICKLUNG</b>	Mittelstufe	<b>EVA</b>
	<b>Prozessorientierter Bericht</b>		Seite <b>2/21</b>

12. Produktdaten.....	16
13. Produktleistungen.....	16
14. Qualitätsanforderungen.....	16
15. Benutzungsoberfläche.....	16
16. Nicht-funktionale Anforderungen.....	16
17. Technische Produktumgebung.....	17
17.1. Software.....	17
17.2. Hardware.....	17
17.3. Orgware.....	17
17.4. Produkt-Schnittstellen.....	17
18. Spezielle Anforderungen an die Entwicklungs-Umgebung.....	17
19. Anhang 3 - DV-Konzept.....	18
Basis/Symfony.....	19
Symfony.....	19
Basisinstallation.....	19
Routing.....	19
Frontend.....	20
19.1. Prozessbeschreibung.....	20
19.2. Mögliche Fehlerquellen.....	20
Backend.....	20
Prozessbeschreibung.....	20
Mögliche Fehlerquellen.....	20
Datenbank.....	21
Technische Basis.....	21
Datenbankmodell.....	21

	Fach: <b>ANWENDUNGSENTWICKLUNG</b>	Mittelstufe	<b>EVA</b>
	<b>Prozessorientierter Bericht</b>		Seite <b>3/21</b>

# 1. Einleitung

## 1.1. Projektumfeld

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

## 1.2. 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.

## 1.3. 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

## 1.4. Projektschnittstellen

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

## 1.5. Projektabgrenzung

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

## 2. Projektplanung

### 2.1. 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 $\triangleq$ 45 min)
Analysephase	2
Konzeptphase	20
Implementierungsphase	42
Summe	64

### 2.2. 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

### 2.3. 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](#))

### 2.4. 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 Einrichtung der Basis gab es kaum Abhängigkeiten zwischen den einzelnen Modulen wodurch eine stark verzahnte Entwicklung entstanden ist

### 3. Analysephase

#### 3.1. 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.

#### 3.2. Wirtschaftlichkeitsanalyse

##### 3.2.1 „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.

##### 3.2.2 Projektkosten

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

Stunden pro Jahr =  $8 \text{ h/Tag} * 220 \text{ Tage/Jahr} = 1760 \text{ h/Jahr}$

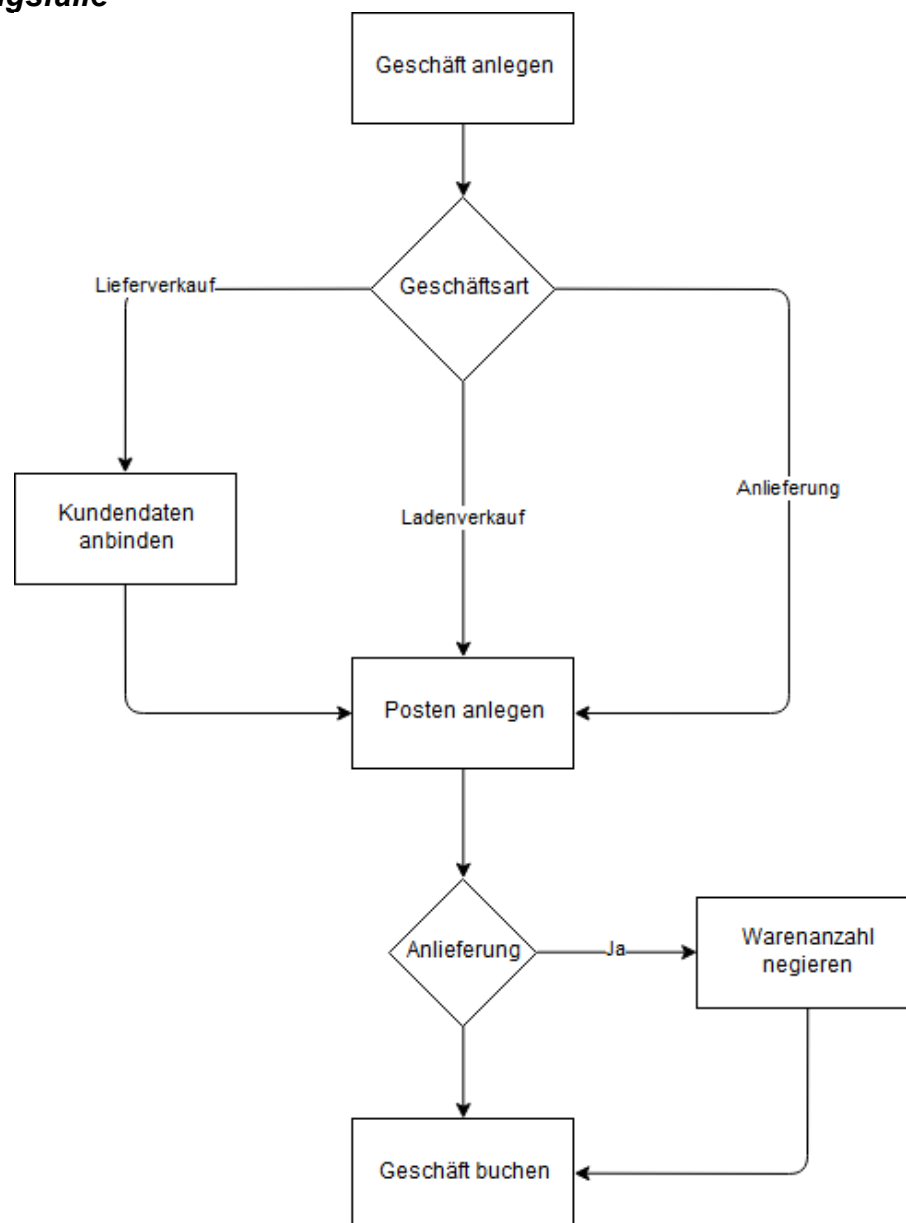
Kosten pro Jahr =  $910\text{€} * 13,3 \text{ Monate/Jahr} = 12103\text{€}$

Kosten pro Stunde:  $\frac{12103 \text{ € / Jahr}}{1760 \text{ h / Jahr}} = 6,88 \text{ € / h}$

Pauschalkosten Ressourcen:  $15\text{€ / h}$

Personalkosten:  $(48 \text{ h} * (6,88 \text{ € / h} + 15 \text{ € / h})) * 4 \text{ Entwickler} = 4200,96 \text{ €}$

### 3.3. Anwendungsfälle



### 3.4. Qualitätsanforderungen

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

### 3.5. Lastenheft/Fachkonzept

Siehe [Anhang 2](#) - Pflichtenheft

## 4. Entwurfsphase

### 4.1. 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.

### 4.2. Architekturdesign

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

Dies ist ein einfaches gut strukturierbares Prinzip, wo Kompetenzen gut verteilt werden können.

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

Eigenschaft	Gewichtung	Akelos	CakePHP	Symfony	Eigenentwicklung
Dokumentation	5	4	3	5	0
Reengineering	3	4	2	5	3
Generierung	3	5	5	5	2
Testfälle	2	3	2	3	3
Standardaufgaben	4	3	3	3	0
<b>Gesamt:</b>	<b>17</b>	<b>65</b>	<b>52</b>	<b>73</b>	<b>21</b>
<b>Nutzwert:</b>		<b>3,82</b>	<b>3,06</b>	<b>4,29</b>	<b>1,24</b>

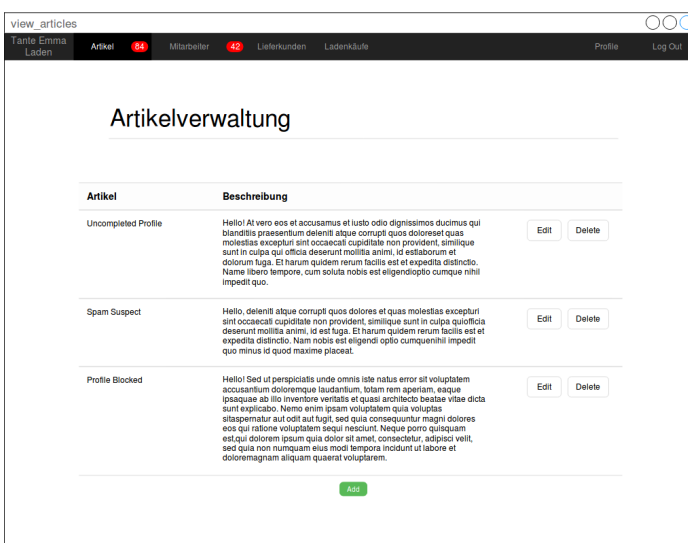
– Quelle: <https://fachinformatiker-anwendungsentwicklung.net/vorlage-fuer-die-projektdokumentation/>

### 4.3. Entwurf der Benutzeroberfläche

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

Wir wollen ein modernes einfaches Design, das für jeden Bestandteil übersichtlich die CRUDs anbietet.

Hier ein Beispiel:



#### 4.4. 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 Maßnahmen ergriffen.

Dies soll bei in der eigenverantwortlichen Abnahme nachgeholt werden.

#### 4.5. Zwischenstand

Konzeption fertig unstimigkeiten bei der Zielplattform aufgrund von kompatibilitätsproblemen

#	Tätigkeitsbezeichnung	HVP	Ergebnis	Von	Bis	ZE	%-Fertig
1	Datenbankkonzept	Lukas	DV-Konzept/ERM	03.05.17	10.05.17	2	100%
2	Backendkonzept	Dennis/Christian	DV-Konzept	03.05.17	10.05.17	4	100%
3	Frontendkonzept	Marvin	DV-Konzept	03.05.17	10.05.17	4	100%
4	Pflichtenheft	Lukas	Pflichtenheft	03.05.17	10.05.17	3	60%
5	POB	Dennis	POB	07.06.17	07.06.17	6	10%

### 5. Implementierungsphase

#### 5.1. Implementierung der Datenstrukturen

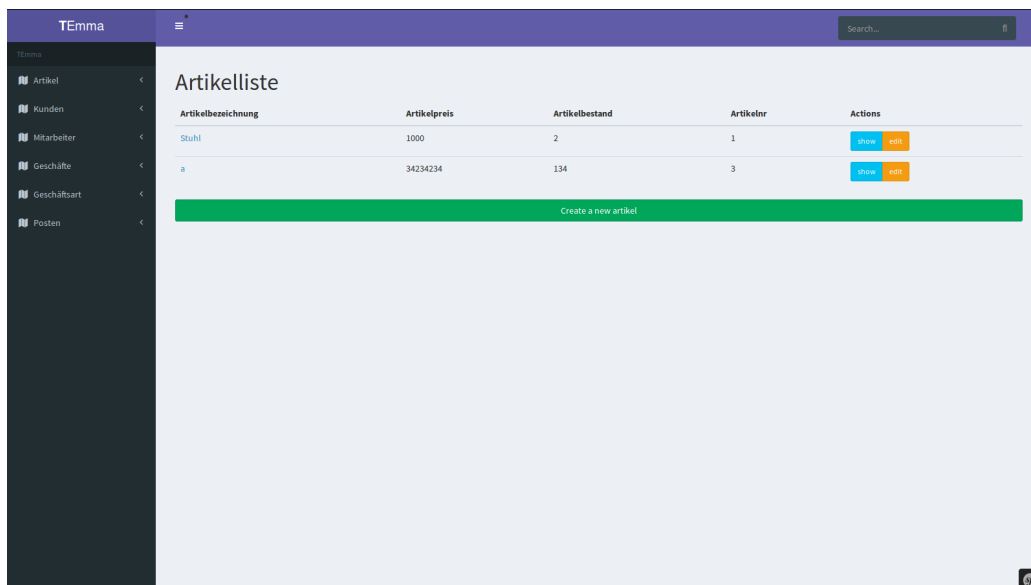
Datenbank wird aus SQL Schema händisch angelegt und dann über Doctrine in XML-Schmas übertragen.

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

#### 5.2. Implementierung der Benutzeroberfläche

Optmierung von Doctrine Generierten Twig Templates.

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



– Aktueller Stand des Frontends



### 5.3. Implementierung der Geschäftslogik

Anpassen von Generierten PHP Controllern und der crud Methoden.

Ermöglichen von Login.

### 5.4. Zwischenstand

Generierte cruds machen einige Probleme und benötigte an meisten Arbeit.

Testkonzept	Christian	Testkonzept	03.05.17	10.05.17	4	100%
Realisierung Datenbank	Lukas	Datenbankskript	10.05.17	17.05.17	4	100%
Realisierung Frontend	Marvin	CRUD-Dateien	17.05.17	31.05.17	8	95%
Lagerverwaltung	Marvin	CRUD-Dateien	17.05.17	31.05.17	2	100%
Lieferverwaltung	Marvin	CRUD-Dateien	17.05.17	31.05.17	2	80%
Kundenverwaltung	Marvin	CRUD-Dateien	17.05.17	31.05.17	2	100%
Mitarbeiterverwaltung	Marvin	CRUD-Dateien	17.05.17	31.05.17	2	100%
Realisierung Backend	Dennis/Christian	CRUD-Controller	24.05.17	07.06.17	16	90%
Lagerverwaltung	Dennis/Christian	CRUD-Controller	24.05.17	07.06.17	4	100%
Lieferverwaltung	Dennis/Christian	CRUD-Controller	24.05.17	07.06.17	4	60%
Kundenverwaltung	Dennis/Christian	CRUD-Controller	24.05.17	07.06.17	4	100%
Mitarbeiterverwaltung	Dennis/Christian	CRUD-Controller	24.05.17	07.06.17	4	100%
Login/Sicherheit	Lukas	Konfigurationsdatei (security.yaml)	31.05.17	31.05.17	1	100%
Testdurchführung		Testergebnis	07.06.17	07.06.17	2	0%
Einrichtung Symfony	Christian	Symfony Server	03.05.17	10.05.17	6	100%
Infrastruktur Symfony für Windows	Christian	Lauffähige Windows Version	10.05.17	17.05.17	4	80%

## 6. Dokumentation

Siehe Anhang

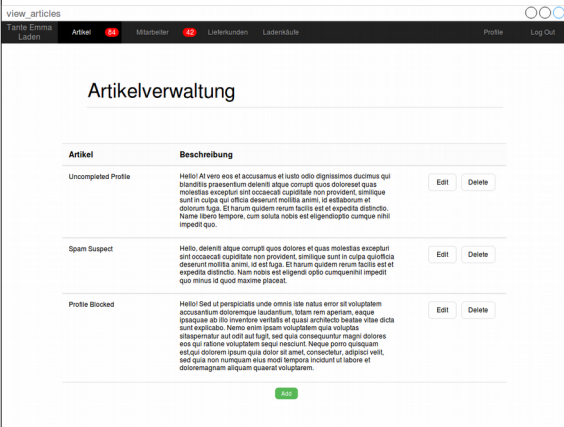
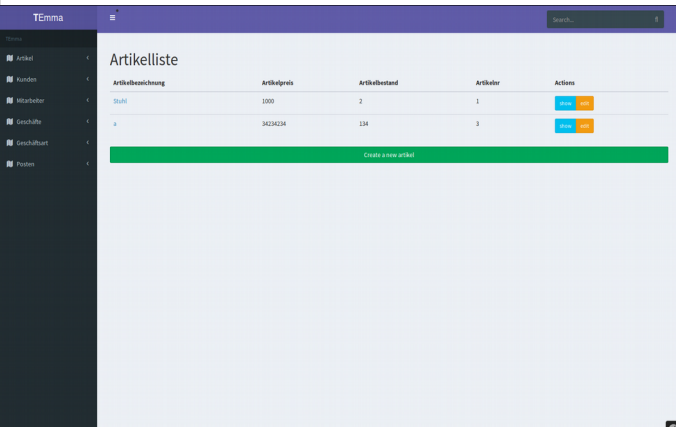
## 7. Fazit

### 7.1. Soll-/Ist-Vergleich

#### 7.1.1 Zielsystem

Das Zielsystem (Windows 7 Schulrechner) hat sich als eine zu große Herausforderung heraus gestellt  
Projekt läuft nun auf einem Laptop mit Ubuntu 16.04.

#### 7.1.2 Frontend

Entwurf	Ergebnis
	

#### 7.1.3 Geschäftslogik

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

### 7.2. Ausblick

#### 7.2.1 Maßnahmen zur Qualitätssicherung

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

#### 7.2.2 Frontend

Einige Formulare müssen noch erweitert werden.


## 8. Anhänge

### 8.1. Anhangverzeichnis

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

### 8.2. Anhang 1 – Aktivitätenplan

Aktivitätenplan			1 ZE = 45 Minuten	4 ZE/Tag	Start: 10.05.2017	Ende: 31.05.2017	
#	Tätigkeitsbezeichnung	HVP	Ergebnis	Von	Bis	ZE	%-Fertig
1	Datenbankkonzept	Lukas	DV-Konzept/ERM	05/03/17	05/10/17	2	100%
2	Backendkonzept	Dennis/Christian	DV-Konzept	05/03/17	05/10/17	4	100%
3	Frontendkonzept	Marvin	DV-Konzept	05/03/17	05/10/17	4	100%
4	Pflichtenheft	Lukas	Pflichtenheft	05/03/17	05/10/17	3	60%
5	POB	Dennis	POB	06/07/17	06/07/17	6	10%
6	Testkonzept	Christian	Testkonzept	05/03/17	05/10/17	4	100%
7	Realisierung Datenbank	Lukas	Datenbankskript	05/10/17	05/17/17	4	100%
8	Realisierung Frontend	Marvin	CRUD-Dateien	05/17/17	05/31/17	8	95%
9	Lagerverwaltung	Marvin	CRUD-Dateien	05/17/17	05/31/17	2	100%
10	Lieferverwaltung	Marvin	CRUD-Dateien	05/17/17	05/31/17	2	80%
11	Kundenverwaltung	Marvin	CRUD-Dateien	05/17/17	05/31/17	2	100%
12	Mitarbeiterverwaltung	Marvin	CRUD-Dateien	05/17/17	05/31/17	2	100%
14	Realisierung Backend	Dennis/Christian	CRUD-Controller	05/24/17	06/07/17	16	90%
15	Lagerverwaltung	Dennis/Christian	CRUD-Controller	05/24/17	06/07/17	4	100%
16	Lieferverwaltung	Dennis/Christian	CRUD-Controller	05/24/17	06/07/17	4	60%
17	Kundenverwaltung	Dennis/Christian	CRUD-Controller	05/24/17	06/07/17	4	100%
18	Mitarbeiterverwaltung	Dennis/Christian	CRUD-Controller	05/24/17	06/07/17	4	100%
20	Login/Sicherheit	Lukas	Konfigurationsdatei (security.yaml)	05/31/17	05/31/17	1	100%
21	Testdurchführung		Testergebnis	06/07/17	06/07/17	2	0%
22	Einrichtung Symfony	Christian	Symfony Server	05/03/17	05/10/17	6	100%
23	Infrastruktur Symfony für Windows	Christian	Lauffähige Windows Version	05/10/17	05/17/17	4	80%

	Fach: <b>ANWENDUNGSENTWICKLUNG</b>	Mittelstufe	<b>EVA</b>
	<b>Prozessorientierter Bericht</b>		Seite <b>12/21</b>

### **8.3. Anhang 2 – Pflichtenheft**

#### **Pflichtenheft:**

Gruppe:

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

Auftrag:

---

## Inhaltsverzeichnis

1 Zielbestimmung.....	3
1.1 Musskriterien.....	3
1.2 Wunschkriterien.....	3
1.3 Abgrenzungskriterien.....	3
2 Produkteinsatz.....	4
2.1 Anwendungsbereiche.....	4
2.2 Zielgruppen.....	4
2.3 Betriebsbedingungen.....	4
3 Produktübersicht.....	4
4 Produktfunktionen.....	5
5 Produktdaten.....	5
6 Produktleistungen.....	5
7 Qualitätsanforderungen.....	5
8 Benutzungsoberfläche.....	5
9 Nicht-funktionale Anforderungen.....	5
10 Technische Produktumgebung.....	6
10.1 Software.....	6
10.2 Hardware.....	6
Ein funktionsfähiger Webserver mit Zugriff ins Netzwerk sowie Terminalrechner mit einem aktuellen Webbrowser zum Beispiel Firefox oder Google Chrome.....	6
10.3 Orgware.....	6
10.4 Produkt-Schnittstellen.....	6
11 Spezielle Anforderungen an die Entwicklungs-Umgebung.....	6

#### **8.4. 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.

#### **8.5. 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

#### **8.6. Wunschkriterien**

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

#### **8.7. Abgrenzungskriterien**

- Keine Buchungssystem

## 9. Produkteinsatz

### 9.1. Anwendungsbereiche

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

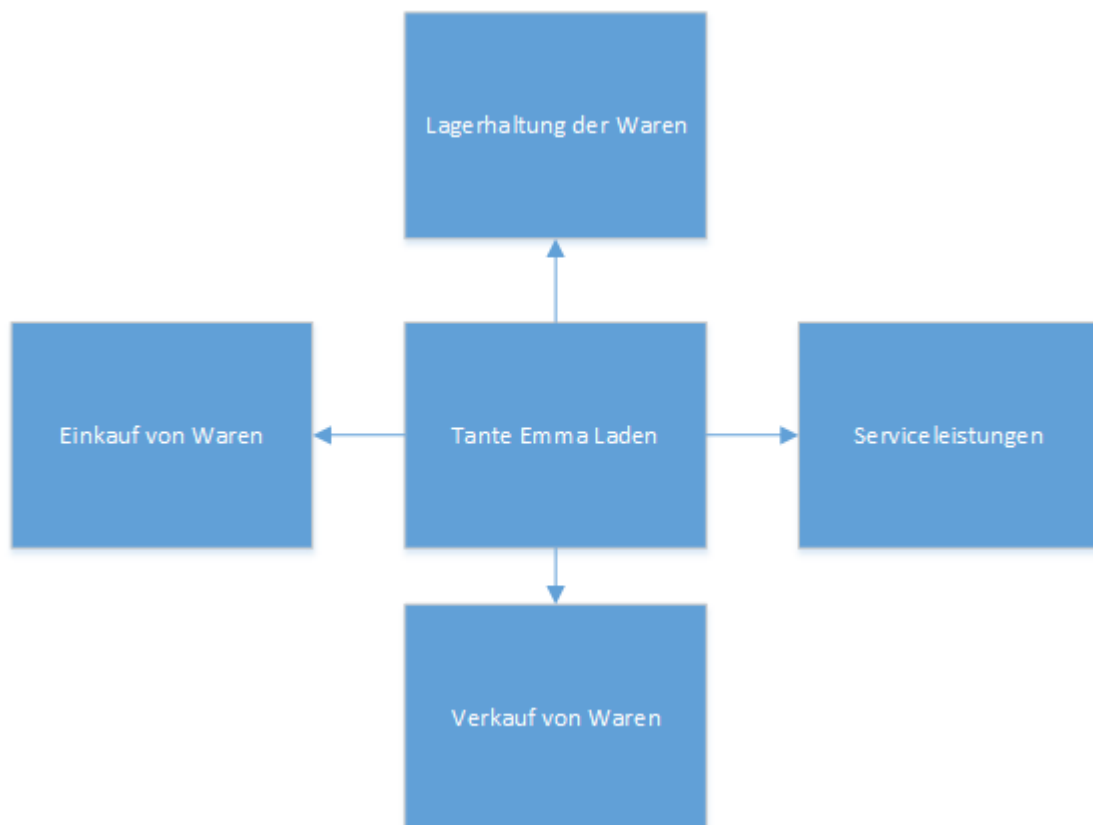
### 9.2. Zielgruppen

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

### 9.3. 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.

## 10. Produktübersicht



## 11. Produktfunktionen

Hauptfunktion des Produktes ist die Verwaltung von Geschäften. Ein Geschäft ist hierbei die Änderung des Bestands von einem oder mehreren 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.

## 12. 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.

## 13. Produktleistungen

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

## 14. Qualitätsanforderungen

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

## 15. Benutzungsoberfläche

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

## 16. 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



## 17. Technische Produktumgebung

### 17.1. 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.

### 17.2. Hardware

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

### 17.3. Orgware


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

### 17.4. Produkt-Schnittstellen

Es sind keine Anwendungsschnittstellen geplant

## 18. Spezielle Anforderungen an die Entwicklungs-Umgebung

Es gibt keine speziellen Anforderungen für die Entwicklungsumgebung

	Fach: <b>ANWENDUNGSENTWICKLUNG</b>	Mittelstufe	<b>EVA</b>
	<b>Prozessorientierter Bericht</b>		Seite <b>18/21</b>

## 19.Anhang 3 - DV-Konzept

### DV-Konzept:

Gruppe:

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

Auftrag:

1.....	2
2. Basis/Symfony.....	2
2.1. Symfony.....	2
2.2. Basisinstallation.....	2
2.3. Routing.....	2
3. Frontend.....	3
4. Backend.....	4
4.1. Prozessbeschreibung.....	4
4.2. Mögliche Fehlerquellen.....	4
5.....	5
6. Datenbank.....	5
6.1. Technische Basis.....	5
6.2. Datenbankmodell.....	6

## 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 YAML-Dateien Konfiguriert.

## Frontend

### 19.1. 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.

### 19.2. 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

