



Abschlussprüfung Sommer 2016

Fachinformatiker für Anwendungsentwicklung  
Dokumentation zur betrieblichen Projektarbeit

# Entwicklung eines Kalkulationskatalogs

**Webanwendung zur Verwaltung von  
Kalkulationspositionen**

Abgabetermin: Chemnitz, den 30.05.2016

**Prüfungsbewerber:**

Norman Paschke



**Ausbildungsbetrieb:**

DMK E-BUSINESS GmbH

Dresdner Straße 40

09130 Chemnitz

**Projektbetreuer:**

Tom Starke, B. Sc.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>III</b>
<b>Tabellenverzeichnis</b>	<b>IV</b>
<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Projektbeschreibung . . . . .	1
1.2 Projektziel . . . . .	1
1.3 Projektbegründung . . . . .	2
1.4 Projektschnittstellen . . . . .	2
1.5 Projektabgrenzung . . . . .	2
<b>2 Projektplanung</b>	<b>3</b>
2.1 Projektphasen . . . . .	3
2.2 Ressourcenplanung . . . . .	3
2.3 Entwicklungsprozess . . . . .	4
<b>3 Analysephase</b>	<b>4</b>
3.1 Ist-Analyse . . . . .	4
3.2 Wirtschaftlichkeitsanalyse . . . . .	5
3.2.1 „Make or Buy“-Entscheidung . . . . .	5
3.2.2 Projektkosten . . . . .	5
3.2.3 Amortisationsdauer . . . . .	6
3.3 Anwendungsfälle . . . . .	6
3.4 Qualitätsanforderungen . . . . .	7
<b>4 Entwurfsphase</b>	<b>7</b>
4.1 Zielplattform . . . . .	7
4.2 Architekturdesign . . . . .	7
4.3 Entwurf der Benutzeroberfläche . . . . .	8
4.4 Datenmodell . . . . .	8
4.5 Geschäftslogik . . . . .	9
4.6 Maßnahmen zur Qualitätssicherung . . . . .	9
<b>5 Implementierungsphase</b>	<b>9</b>
5.1 Implementierung der Datenstrukturen . . . . .	9
5.2 Implementierung der Benutzeroberfläche . . . . .	9
5.3 Implementierung der Geschäftslogik . . . . .	10

<b>6</b>	<b>Testphase</b>	<b>12</b>
<b>7</b>	<b>Einführungsphase</b>	<b>12</b>
<b>8</b>	<b>Dokumentation</b>	<b>12</b>
<b>9</b>	<b>Fazit</b>	<b>13</b>
9.1	Soll-/Ist-Vergleich . . . . .	13
9.2	Lessons Learned . . . . .	13
9.3	Ausblick . . . . .	13
	<b>Literaturverzeichnis</b>	<b>15</b>
	<b>Eidesstattliche Erklärung</b>	<b>16</b>
<b>A</b>	<b>Anhang</b>	<b>i</b>
A.1	Detaillierte Zeitplanung . . . . .	i
A.2	Verwendete Ressourcen . . . . .	ii
A.3	Ausschnitt einer Excel-Tabelle . . . . .	iii
A.4	Datenbankmodell . . . . .	iv
A.5	SQL - Auszug . . . . .	v
A.6	Entwürfe User Interface . . . . .	vi
A.7	Screenshots der Anwendung . . . . .	viii

## Abbildungsverzeichnis

1	getCompletePriceMin()-Methode . . . . .	11
2	Ausschnitt einer Excel-Tabelle einer Kalkulation . . . . .	iii
3	Datenbankmodell . . . . .	iv
4	Auszug aus der Datenbank in SQL-Format . . . . .	v
5	Liste der Kalkulationen mit Möglichkeit zum Filtern und Hinzufügen	vi
6	Bearbeiten bzw. Hinzufügen einer neuen Kalkulation mit Positionen	vii
7	Liste der Kalkulationen . . . . .	viii
8	Bearbeitung einer Kalkulation . . . . .	ix

## Tabellenverzeichnis

1	Zeitplanung . . . . .	3
2	Kostenaufstellung . . . . .	6

## Abkürzungsverzeichnis

<b>BE</b>	Back-End
<b>CMS</b>	Content Management System
<b>CSS</b>	Cascading Style Sheets
<b>DMK</b>	DMK E-BUSINESS GmbH
<b>FE</b>	Front-End
<b>HTML</b>	Hypertext Markup Language
<b>MVC</b>	Model View Controller
<b>PDF</b>	Portable Document Format
<b>PHP</b>	Hypertext Preprocessor
<b>SQL</b>	Structured Query Language
<b>URL</b>	Uniform Resource Locator
<b>XAMPP</b>	Cross-Platform, Apache, MariaDB, PHP und Perl

# 1 Einleitung

Die vorliegende Projektdokumentation beschreibt ein IHK-Abschlussprojekt, welches der Autor im Rahmen der Ausbildung zum Fachinformatiker mit der Fachrichtung Anwendungsentwicklung erstellt hat. Der Ausbildungsbetrieb, die Firma DMK E-BUSINESS GmbH (DMK), ist eine in Chemnitz ansässige Online-Agentur, deren Schwerpunkt auf der Umsetzung von Webauftritten für mittelständige Unternehmen im privaten und öffentlichen Bereich liegt. Gegenwärtig beschäftigt DMK 38 Mitarbeiter. Der Autor war bereits vor seiner Ausbildungszeit mehrjähriger Angestellter im Unternehmen und ist hauptsächlich mit Arbeiten an Kundenprojekten im Bereich Front-End (FE) betreut. Als zertifizierter TYPO3-Integrator<sup>1</sup> ist er an der Einrichtung und Betreuung von Umgebungen des CMS ebenso beteiligt wie an der Dynamisierung von HTML/CSS-Prototypen.

## 1.1 Projektbeschreibung

Um an öffentlichen Ausschreibungen teilnehmen zu können, muss DMK in Vorleistung gehen, die Aufwände für jede Position des Lastenheftes schätzen und in ein Angebot überführen. Oftmals werden wiederkehrende Positionen für öffentliche Ausschreibungen immer und immer wieder neu geschätzt. Um diesem Vorgehen Einhalt zu gebieten, soll mit der Entwicklung eines Kalkulationskatalogs der Prozess der Schätzung verkürzt werden.

## 1.2 Projektziel

Die fertige Anwendung soll in erster Linie als Nachschlagewerk für bereits kalkulierte Positionen dienen, um bei neuen Ausschreibungen schneller eine sichere Aussage über den Aufwand der Aufgaben zu geben. Darüber hinaus soll sie die Möglichkeit bieten, ganze Kalkulationen möglichst bequem zusammenzustellen. Nützlich wäre es zudem, wenn die Aufwände zusammengerechnet würden, um eine erste grobe Einschätzung des Projekts zu bekommen und dessen Umsetzbarkeit zu bewerten oder gegebenenfalls nicht an der Ausschreibung teilzunehmen. Es ist wichtig, dass dabei ein leichter Einstieg und die einfache Bedienung der Anwendung gewährleistet wird. Wenn die Hürde zu groß wäre, würde keine Nutzung stattfinden und das Projekt wäre gescheitert. Eine enge Zusammenarbeit und regelmäßige Absprachen

---

<sup>1</sup>TYPO3 ist ein Open-Source Content Management System, das besonders im deutschsprachigen Raum Einsatz findet

mit dem Auftraggeber und den späteren Anwendern ist daher unerlässlich und von Anfang an ein wichtiges Projektziel.

### 1.3 Projektbegründung

Die Schätzung von Ausschreibungen ist ein zeitaufwändiger Prozess. Zeit ist in einer Online-Agentur, wie in jedem anderen Unternehmen auch, eine extrem kostbare Ressource. Neben der Bearbeitung von Kundenprojekten und organisatorischen Dingen im Unternehmen, müssen einzelne Entwickler zusätzlich Zeit für Kalkulationen bekommen, in der keine Entwicklung an Projekten stattfinden kann. Es geht also wertvolle Entwicklerzeit verloren, weil Kalkulationen für eine Angebotserstellung eine Vorleistung für die Agentur darstellen. Vergütungen für die Teilnahme an einer Ausschreibung oder für die Abgabe eines Angebots finden nur in den seltensten Fällen statt.

Die Einführung eines Katalogs, der wiederkehrende Anforderungen beinhaltet, kann Ressourcen im Unternehmen freisetzen, wenn dadurch weniger Mitarbeiter über die gesamte Dauer einer Kalkulation im Prozess involviert sind und die Dauer des Schätzens verringert wird. Anstatt jeden Punkt für jede Ausschreibung neu zu schätzen, kann im System nach der Anforderung gesucht werden. Selbst Projektmanager, die nicht das technische Wissen besitzen, das für eine Schätzung der meisten Anforderungen nötig ist, können Kalkulationen damit selbstständig erstellen.

### 1.4 Projektschnittstellen

Die vorliegende Software, die im Laufe dieser Projektarbeit erstellt wurde, ist unabhängig von Schnittstellen zu anderen Anwendungen.

Als Nutzergruppe der neuen Anwendung werden alle Mitarbeiter des Unternehmens gesehen. Dies beinhaltet Projektmanager mit Fokus auf betriebswirtschaftlichen Aspekten sowie Entwickler im FE- und BE-Bereich, die während der Umsetzung beratend zur Seite standen.

### 1.5 Projektabgrenzung

Die Webanwendung zur Verwaltung von Kalkulationspositionen soll explizit keine Angebote automatisch erstellen (z. B. als PDF-Export). Weiterhin war es keine



Anforderung, aktuelle Schätzungen in das System zu importieren oder anderweitig zu verarbeiten. Es ist ebenfalls kein Werkzeug, um eine Verwaltung von Kunden abzubilden. Das Hauptaugenmerk gilt der Schätzung und Listung von Kalkulationspositionen.

## 2 Projektplanung

### 2.1 Projektphasen

Eine grobe Zeitplanung wurde im Vorfeld erstellt, dabei war sich an der Vorgabe von 70 Stunden zu halten. Tabelle 1: Zeitplanung stellt den groben zeitlichen Ablauf des Projektes dar. Die Entwicklung des Katalogs begann Mitte April 2016 und zog sich über sieben Wochen hin, in denen regelmäßig ca. 12 Stunden pro Woche am Projekt gearbeitet wurde.

Projektphase	Geplante Zeit
Analysephase	2 h
Entwurfsphase	8 h
Umsetzung der Programmlogik in PHP	48 h
Testphase	2 h
Dokumentation	10 h
<b>Gesamt</b>	<b>70 h</b>

Tabelle 1: Zeitplanung

Die detaillierte Zeitplanung findet sich im Anhang A.1: Detaillierte Zeitplanung auf Seite i.

### 2.2 Ressourcenplanung

Alle im Projekt verwendeten Ressourcen finden sich in einer Übersicht im Anhang A.2: Verwendete Ressourcen auf Seite ii. Benötigte Hardware wurde von DMK zur Verfügung gestellt. Bei der Auswahl der Software wurde darauf geachtet, dass die Produkte kostenfrei (z. B. Open Source) erhältlich und nutzbar sind oder – wie im Fall von Windows – das Unternehmen bereits Lizenzen dafür besitzt.

## 2.3 Entwicklungsprozess

Bevor es in die Umsetzungsphase gehen konnte, war es nötig sich für einen passenden Entwicklungsprozess zu entscheiden. Dabei galt es zwischen einem Wasserfallmodell oder einem agilen Prozess abzuwägen. Um eine stetige Rückmeldung von den Testern zu erhalten, fiel die Wahl schnell auf ein agiles Vorgehen. Dabei wurde aber nicht explizit ein Vorgehen nach Scrum<sup>2</sup> favorisiert, sondern eine flexiblere Variante, die regelmäßige Treffen in Form von Test-Sitzungen vorsah. Klassische Daily Stand-Ups oder Sprintplanungen fanden daher nicht statt.

Die strikte Einteilung in Phasen, wie sie in 2.1: Projektphasen vorgenommen wurde, musste entsprechend aufgeweicht werden, da mehrere Phasen zyklisch auftraten und sich mit geringerem Aufwand wiederholten.

Agiles Vorgehen hat den Vorteil, dass sehr schnell ein arbeitsfähiger Stand präsentiert werden kann. Am Ende jeder Woche wurde eine kurze Vorstellung des Projektes durchgeführt. Anmerkungen und Fehler wurden notiert und im nächsten Zyklus bearbeitet, sodass eine kontinuierliche Verbesserung des Programms stattgefunden hat.

## 3 Analysephase

### 3.1 Ist-Analyse

Um Kalkulationen für öffentliche Ausschreibungen durchzuführen, wird momentan jeweils eine große Excel-Tabelle angelegt, auf die alle an der Kalkulation beteiligten Mitarbeiter zugreifen können. In die Tabelle werden untereinander händisch alle Anforderungen, die es zu schätzen gilt, aus dem Lastenheft übertragen. Daneben werden die Aufwände eingegeben und am Ende rechnet die Tabellenkalkulation eine Summe aller Aufwände multipliziert mit dem Tagessatz des Unternehmens aus. Eine solche Tabelle befindet sich im Anhang A.3: Ausschnitt einer Excel-Tabelle auf Seite iii.

Durch eine mündliche Befragung der Mitarbeiter wurde schnell klar, dass keiner mit der bisherigen Lösung zu 100 % zufrieden ist. Die Tabelle wird schnell unübersichtlich, das Eintragen der Positionen ist zeitaufwendig. Die meiste Zeit geht aber beim Schätzen selbst verloren: Positionen, die regelmäßig in Ausschreibungen auftreten, müssen jedesmal neu geschätzt werden. Das führt nicht selten zu

---

<sup>2</sup><https://de.wikipedia.org/wiki/Scrum>

Stresssituationen unter den Beteiligten, die mit anhaltender Dauer der Schätzung zusehends angespannter werden. Diesen Zustand gilt es um jeden Preis zu vermeiden. Eine Sammlung bereits geschätzter Positionen würde die Arbeit wesentlich beschleunigen und zur psychischen Entlastung beitragen.

## 3.2 Wirtschaftlichkeitsanalyse

### 3.2.1 „Make or Buy“-Entscheidung

Die Mitarbeiter behelfen sich mit der in 3.1: Ist-Analyse genannten Excel-Tabelle, so gut sie können. Leider findet sich auf dem Markt für eine derartig spezielle Form der Schätzung keine passende Software, die allen Vorgaben gerecht wird. Selbst wenn es Softwarelösungen gäbe, wären anfallende Lizenzkosten ein großes Hindernis. Die Alternative, eine eigene Software zu programmieren, stand schon zur Debatte, wurde aber aus Zeitgründen nie verwirklicht. Deshalb hat es sich angeboten diese Aufgabe in Form einer betrieblichen Projektarbeit umzusetzen.

### 3.2.2 Projektkosten

Die Kosten für die Durchführung des Projekts setzen sich sowohl aus Personal-, als auch aus Ressourcenkosten zusammen. Laut Tarifvertrag verdient ein Auszubildender im dritten Lehrjahr pro Monat 1000 € Brutto und arbeitet pro Monat etwa 20 Tage, 8 Stunden täglich.

$$\begin{aligned} 8 \text{ h/Tag} \cdot 20 \text{ Tage/Monat} &= 160 \text{ h/Monat} \\ \frac{1000 \text{ €/Monat}}{160 \text{ h/Monat}} &= 6,25 \text{ €/h} \end{aligned}$$

Es ergibt sich also ein Stundenlohn von 6,25 €. Die Durchführungszeit des Projektes beträgt 70 Stunden. Für die Nutzung von Ressourcen<sup>3</sup> beträgt der Stundensatz pauschal 15 €. Für andere Mitarbeiter wird firmenintern ein pauschaler Stundenlohn von 25 € angenommen. Daraus ergibt sich eine Kostenaufstellung, die sich in Tabelle 2: Kostenaufstellung befindet. Die Gesamtkosten belaufen sich damit auf 1687,50 €.

---

<sup>3</sup>Räumlichkeiten, Arbeitsplatzrechner etc.

Vorgang	Zeit	Kosten/h	Kosten
Entwicklungskosten	70 h	6,25 € + 15 € = 21,25 €	1487,50 €
Befragung	3 h	25 € + 15 € = 40 €	120 €
Tests und Review	2 h	25 € + 15 € = 40 €	80 €
			<b>1687,50 €</b>

Tabelle 2: Kostenaufstellung

#### 3.2.3 Amortisationsdauer

Durch die Einführung eines Katalogs entsteht Sicherheit bei der Schätzung und der Gesamte Vorgang wird beschleunigt oder Entfällt teilweise ganz.

Im Nachfolgenden soll der Zeitpunkt ermittelt werden, bei dem sich die Entwicklung der Anwendung wirtschaftlich rechnet. Anhand des Wertes kann beurteilt werden, ob das Projektes aus wirtschaftlicher Sicht sinnvoll ist und sich zukünftig Kostenvorteile ergeben. Die Amortisationsdauer berechnet sich aus den Anschaffungskosten geteilt durch Kostenersparnis, die durch die Anwendung entsteht.

Bei einer Zeiteinsparung von einer Stunde pro Schätzung und Woche für jeden der drei Mitarbeiter<sup>4</sup> und 52 Wochen im Jahr, ergibt sich eine Zeitersparung von:

$$3 \cdot 1 \text{ h/Woche} \cdot 52 \text{ Wochen/Jahr} = 156 \text{ h/Jahr}$$

Dadurch ergibt sich eine jährliche Einsparung von:

$$156 \text{ h} \cdot (25 + 15) \text{ €/h} = 6240 \text{ €}$$

Die Amortisationszeit beträgt demnach:

$$\frac{1687,50 \text{ €}}{6240 \text{ €/Jahr}} \approx 0,27 \text{ Jahre} \approx 14 \text{ Wochen}$$

### 3.3 Anwendungsfälle

Im Wesentlichen muss der Kalukulationskatalog zwei Anwendungsfälle abdecken, die im Folgenden festgelegt sind.

---

<sup>4</sup>an Schätzungen für öffentliche Ausschreibungen nehmen in der Regel drei Mitarbeiter teil

Zum einen soll die Anwendung als Nachschlagewerk während einer Kalkulation dienen. Dabei muss die Liste der Positionen gefiltert werden, um eine effektive Suche nach dem gewünschten Element zu sichern.

Zum anderen muss es möglich sein, neue Kalkulationen mit den entsprechend nötigen Feldern und Positionen anzulegen. Hierbei ist ebenfalls darauf zu achten, dass der Anwendungsfall mit geringem zeitlichen Aufwand vollzogen werden kann. Eine Berechnung des Gesamtpreises über alle Aufwände hinweg, wird als selbstverständlich angesehen.

### 3.4 Qualitätsanforderungen

Es wurden keine spezifischen Qualitätsanforderungen an die Anwendung gestellt. Daher gelten die allgemeinen Anforderungen für Software wie sie in ISO/IEC 9126-1 [2001] formuliert sind.

## 4 Entwurfsphase

### 4.1 Zielplattform

Das Programm sollte als eigenständige Anwendung zum Einsatz kommen. Da bisher keine Sammlung von Kalkulationspositionen vorhanden ist, muss eine Datenbank eingerichtet werden. Der Einsatz des MySQL-Fork MariaDB bewährt sich täglich im Unternehmen. PHP dagegen ist eine Skriptsprache, die besonders geeignet ist, um eine dynamische Webanwendung auf Basis freier Software umzusetzen. Sie ist die am häufigsten verwendete Programmiersprache zum Erstellen von Websites. WIKIPEDIA [2016b] Die Mitarbeiter sind mit den Technologien vertraut und können gegebenenfalls das System warten und weiterentwickeln.

### 4.2 Architekturdesign

Der Kalkulationskatalog soll auf Basis einer MVC-Architektur umgesetzt werden. Diese teilt jede Softwarekomponente in einen der drei Bestandteile – Datenmodell (engl. model), Präsentation (engl. view) oder Programmsteuerung (engl. controller) WIKIPEDIA [2016a] – zu. Jeder Teil hat einen speziellen Aufgabenbereich, der von den anderen weitestgehend unabhängig ist.

Das Model repräsentiert die Daten, der View stellt die Daten dar und der Controller steuert den Ablauf der Anwendung – das Bindeglied zwischen Model und View. Durch die lose Verbindung der Komponenten, erhöht sich die Wiederverwendbarkeit und Austauschbarkeit. So lässt sich beispielsweise die Oberfläche komplett verändern, ohne dabei das Model anpassen zu müssen. Die Trennung bevorteilt außerdem das Testen und Warten der Komponenten.

Es wurde sich bewusst für die Erstellung eines eigenen kleinen Frameworks entschieden. Zum einen, weil der Lerneffekt dadurch am größten ist, zum anderen, weil jedes vorhandene Framework eine Unmenge an nicht genutzten Code mitgebracht hätte. Durch ein eigens entwickeltes Framework bleibt der Code schlank und übersichtlich.

### 4.3 Entwurf der Benutzeroberfläche

Eine strukturierte Oberfläche ist für eine einfache und zielgerichtete Bedienung von großer Bedeutung. Erste Wireframes wurden mit dem Tool Balsamiq BALSAMIQ [2016] erstellt und geben die Richtung für den Aufbau der Oberfläche vor. Die Entwürfe befinden sich im Anhang A.6: Entwürfe User Interface auf Seite vi.

Um schnell eine saubere, responsive<sup>5</sup> Weboberfläche zu generieren, wurde auf das Framework Bootstrap BOOTSTRAP TEAM [2016] (ehemals *Twitter Bootstrap*) gesetzt. Als CSS-Framework bietet es Vorlagen für alle erdenklichen Module: Textstyling, Formulare, Listen und Tabellen sind einige wenige Vorlagen, die in diesem Projekt Verwendung finden.

### 4.4 Datenmodell

Das Datenmodell mit eingezeichneten Kardinalitäten ist im Anhang A.4: Datenbankmodell auf Seite iv einzusehen. Im Zentrum stehen die Kalkulationen, die jeweils einen Kunden und einen Status besitzen. Kalkulationen bestehen außerdem aus einer Vielzahl von Positionen. Diese wiederum besitzen eine Kategorie. Die n:m-Beziehung zwischen Kalkulationen und Positionen wird durch eine separate Tabelle abgebildet. Globale Einstellungen, wie Firmenname und Mehrwertsteuersatz, werden in einer gesonderten Tabelle (Company) abgelegt.

---

<sup>5</sup>[https://de.wikipedia.org/wiki/Responsive\\_Design](https://de.wikipedia.org/wiki/Responsive_Design)

### 4.5 Geschäftslogik

Jedes Objekt innerhalb der Anwendung ist eine Instanz einer Model-Klasse. Daraus ergeben sich folgende Klassen: **Calculation**, **Category**, **Company**, **Customer**, **Item** und **Status**. Mit der Ausnahme von **Company** bekommt jede der Klassen eine Methode **show()** (Anzeigen) und **create()** (Anlegen), sowie **delete()** (Löschen), **edit()** (Bearbeiten) und ggf. **all()** (Listenausgabe). Weiterhin gibt es Methoden, um an Informationen anderer Tabellen zu gelangen, **Calculation::getCustomer()** z. B. liefert das Objekt des Kunden, der der aktuellen Kalkulation zugeordnet ist. Analog dazu sind alle anderen Methoden aufgebaut.

### 4.6 Maßnahmen zur Qualitätssicherung

Um die Funktionalität der Anwendung zu gewährleisten, wurden Anwendertests geschaffen. Diese sehen die systematischen Tests aller Funktionen vor, indem alle Aktionen der Software, mit unterschiedlichen Eingaben ausgeführt werden.

Wird ein Test nicht bestanden, kann in git ein Fehlerbericht angelegt werden.<sup>6</sup> Git wird außerdem benutzt, um jederzeit einen stabilen Entwicklungsstrang der Anwendung zu haben, jeder kann sich an der Weiterentwicklung beteiligen und Vorschläge zur Verbesserung einbringen.

## 5 Implementierungsphase

### 5.1 Implementierung der Datenstrukturen

Die Datenbank wurde von Hand in Form einer SQL-Datei angelegt. Sie wird per phpMyAdmin in das Datenbanksystem importiert und enthält für jeden Bereich eine kleine Anzahl an Beispieldaten. Ein Auszug der Datei liegt im Anhang A.5: SQL - Auszug auf Seite v.

### 5.2 Implementierung der Benutzeroberfläche

Bei der Implementierung der Benutzeroberfläche wurde darauf geachtet, dass eine Übersichtliche Struktur bewahrt bleibt. Farben sollten sehr sparsam und nur an Stellen eingesetzt werden, wo sie für den Nutzer eine sinnvolle Hilfestellung darstellen.

---

<sup>6</sup>[https://github.com/darthnorman/calc\\_catalog/issues](https://github.com/darthnorman/calc_catalog/issues)

Dies macht sich vor allem bei der Darstellung von Buttons bemerkbar. Sie stellen die primären Interaktionspunkte für Anwender dar.

Die Navigation, die es ermöglicht zwischen den Listenansichten der Komponenten zu wechseln, ist ständig prominent am oberen Bildschirmrand platziert. Von dort aus kann jederzeit auf eine andere Komponente der Anwendung zugegriffen werden. Die Einstellungen (Firmenname und Mehrwertsteuersatz) werden nur extrem selten gebraucht und wurden deshalb an den rechten Rand der Navigation gesetzt.

Ein großes Suchfeld über jeder Listenansicht bekommt bei einem Neuladen der Seite immer den Fokus, sodass direkt mit der Eingabe und damit einer Filterung begonnen werden kann, ohne dabei mit der Maus erst ins das Feld navigieren zu müssen. Das Leeren des Feldes funktioniert über das Drücken der `ESC`-Taste. Damit ist die Filterfunktion äußerst schnell rein über die Tastatur bedienbar. Durch zusätzliche Platzhalter in den Eingabefeldern, wird der Nutzer auf das Format seiner Eingaben hingewiesen.

Screenshots der Anwendung in der Entwicklungsphase mit Dummy-Daten befinden sich im Anhang A.7: Screenshots der Anwendung auf Seite viii.

### 5.3 Implementierung der Geschäftslogik

Um die Logik den Anforderungen entsprechend abzubilden, wurde zunächst eine Konfigurationsdatei angelegt, die die Zugangsdaten für die Datenbank enthält. Die *PHP Data Objects*-Erweiterung stellt die zentrale Schnittstelle zur Datenbank bereit und wird für alle Interaktionen mit ihr genutzt.

Für die Ausgabe wurde im nächsten Schritt die Funktion `render()` geschrieben, die anhand der aufgerufenen URL den Controller und den View bestimmt. Ein Controller-Array beinhaltet alle erlaubten Kombinationen und vergleicht diese mit der übergebenen URL, dargestellt in 5.3: `getCompletePriceMin()`-Methode.

Die einzelnen Controller bestimmen mittels des gewählten Views, welche Daten geholt und durch welches Template diese ausgegeben werden. Das Auslesen der Daten wird durch die Model-Klassen realisiert, indem Datenbankabfragen getätigt werden und die Objekte an das Template übergeben werden.

Nach und nach wurden die Methoden zur Bearbeitung, Speicherung und Erstellung neuer Datensätze geschaffen. Helferfunktionen, die kleine Aufgaben bei der Darstellung von Werten im Template oder innerhalb von Methoden übernehmen, wurden gesondert in der Datei `functions.php` gelistet und global eingebunden.



Am Beispiel der Klasse Calculation soll die Methode `getCompletePriceMin()` vorgestellt werden. Die Methode gibt den minimalen Gesamtpreis (ohne Steuern) der übergebenen Kalkulation zurück.

### `getCompletePriceMin()`-Methode

```
1 /**
2  * Berechnet den minimalen Gesamtpreis einer Kalkulation
3  * @param $id ID der aktuellen Kalkulation
4  * @return $pt minimaler Gesamtpreis in Euro
5  */
6 public static function getCompletePriceMin($id) {
7     // hole alle Positionen dieser Kalkulation
8     $items = self::getItems($id);
9     // hole aktuelle Kalkulation
10    $currentCalculation = self::show($id);
11
12    // Iteriere ueber alle Positionen und berechne
13    // den minimalen Gesamtaufwand
14    $pt = 0;
15    for ($i = 0; $i < count($items); $i++) {
16        $task = $items[$i]—>tmin;
17        $pt = $pt + $task;
18    };
19
20    // Tagessatz des Teams
21    $team = floatval( $currentCalculation —>price_team);
22
23    // Multipliziere 10% des Gesamtaufwands
24    // mit dem Tagessatz der PMs
25    $pm = floatval( $currentCalculation —>price_pm);
26    $pm = $pt * 0.1 * $pm;
27
28    // Multipliziere die Gesamtzahl der Personentage
29    // mit dem Tagessatz des Teams
30    $pt = $pt * $team;
31
32    // Addiere PM–Aufwand auf Gesamtaufwand
33    $pt = $pt + $pm;
34
35    // Gib minimalen Gesamtpreis der Kalkulation zurueck
36    return $pt;
37 }
```

Abbildung 1: `getCompletePriceMin()`-Methode

Im Template `showCalculation.php` wird diese Methode wie folgt aufgerufen.

```
1 <?php echo getBrutto($calculation—>getCompletePriceMin($calculation—>id)) ?>
```

Die Funktion `getBrutto()` ist eine Hilfsfunktion und daher nicht Teil der Klasse, sie holt sich den Mehrwertsteuersatz aus den Einstellungen und berechnet dazu einen Wert inklusive der Steuer.

## 6 Testphase

Aus Zeitgründen wurden keinerlei automatische Tests implementiert. Das Testen der Anwendung wird manuell vorgenommen, wie unter 4.6: Maßnahmen zur Qualitätssicherung beschrieben.

## 7 Einführungsphase

Um die Software zu deployen, muss lediglich ein Webserver mit einer Datenbank bereitgestellt werden. Die Quelldateien können entweder von GitHub heruntergeladen oder geklont werden. Damit eine Verbindung zur Datenbank aufgebaut werden kann, muss ein Duplikat der Datei `/Classes/config_sample.php` in `config.php` umbenannt werden und die Verbindungsdaten (Datenbankhost, -Name, -Nutzer und Passwort) müssen eingetragen sein. Ein Import der beiliegenden Datenbank ist nicht zwingend erforderlich, die Datensätze können auch aus der Anwendung heraus angelegt werden.

Eine Migration alter Kalkulationen hat nicht stattgefunden, da es nicht Bestandteil des Projektes war. Dennoch sollte der Import einiger Bestandsdaten in naher Zukunft durchgeführt werden, es würde den Einstieg und die initiale Arbeit mit dem Katalog erheblich erleichtern.

## 8 Dokumentation

Die Dokumentation der Anwendung besteht aus drei Teilen. Die vorliegende Projektdokumentation beschreibt die Phasen der Umsetzung. Eine Administratorenanleitung befindet sich in Form einer README auf GitHub.<sup>7</sup> Die Entwicklerdokumentation beinhaltet die Auflistung der Geschäftslogik und deren Bedeutung. Sie ist ebenfalls im genannten Git-Repository unter `/Documentation/Classdoc` zu finden. Erstellt wurde diese mit phpDocumentor PHPDOC.ORG [2016], einer Software zur automatischen Generierung von kommentierten PHP-Code.

---

<sup>7</sup>[https://github.com/darthnorman/calc\\_catalog](https://github.com/darthnorman/calc_catalog)

Eine Anwenderdokumentation musste nicht erstellt werden, alle Benutzer durchlaufen eine kurze mündlichen Unterweisung und sind danach selbstständig in der Lage, den Kalkulationskatalog zu benutzen.

## 9 Fazit

### 9.1 Soll-/Ist-Vergleich

Das Projektziel, einen Katalog mit bereits geschätzten Positionen zu schaffen, war erfolgreich. Auf lange Sicht muss sich die Anwendung aber erst noch beweisen. Die Mitarbeiter sind bislang sehr zufrieden und wollen sie auch weiterhin für ihre Kalkulationen verwenden.

Die Projektplanung wurde im Großen und Ganzen eingehalten. Abweichungen gab es im Bereich der Umsetzung der Geschäftslogik. Besonders die Methoden zum Bearbeiten und Hinzufügen neuer Objekte erwiesen sich als schwieriger als zunächst angenommen. Demzufolge musste an dieser Stelle mehr Aufwand investiert werden. Im Gegensatz dazu konnte bei der Umsetzung der Oberfläche Zeit gespart werden, das lag unter anderem an den bereits vorhandenen Erfahrungen mit dem Framework Bootstrap.

### 9.2 Lessons Learned

Während der Bearbeitung könnte der Autor wertvolle Erfahrungen in verschiedenen Bereichen der Softwareentwicklung machen. Zwar war mit dem Schreiben eines eigenen MVC-Frameworks ein nicht unerheblicher Zeitaufwand verbunden, jedoch war der Lerneffekt dabei umso größer. Das Wissen bzgl. des Umgangs mit  $\text{\LaTeX}$ , und phpDocumentor konnte ebenfalls ausgebaut und gefestigt werden.

### 9.3 Ausblick

Der Katalog läuft stabil und kann im Projektalltag eingesetzt werden. Erweiterungen sind natürlich trotzdem wünschenswert. Ein wichtiger Punkt für die Zukunft wäre die in Kapitel 7: Einführungsphase bereits erwähnte Migration bestehender Daten, um den Einstieg zu erleichtern.

Darüber hinaus gibt es derzeit keine Möglichkeit, Positionen für eine bestimmte Kalkulation anzupassen. Weitere Felder, wie bspw. Multiplikatoren für Positionen, könnten nützliche Ergänzungen darstellen. Auch eine Einstellmöglichkeit für den Prozentsatz von Projektmanagerleistungen, die auf eine Kalkulation aufgeschlagen werden, fehlt bislang. Gleiches gilt für eine Rabattfunktion. Die modulare Struktur des Programms steht diesen Neuerungen nicht im Weg.

Wenn möglich, sollten nach Abschluss eines Projektes die kalkulierten Aufgaben mit den tatsächlich benötigten Zeiten verglichen werden. Bei Abweichungen können die geschätzten Zeiten im Programm angepasst werden, damit zukünftige Schätzungen noch genauer und zuverlässiger werden.

## Literaturverzeichnis

### **Balsamiq 2016**

BALSAMIQ: *Balsamiq Website*. Version: 2016. <https://balsamiq.com/>,  
Abruf: 15.05.2016

### **Bootstrap Team 2016**

BOOTSTRAP TEAM: *Bootstrap Website*. Version: 2016. <http://getbootstrap.com/>, Abruf: 29.05.2016

### **ISO/IEC 9126-1 2001**

ISO/IEC 9126-1: *Software-Engineering – Qualität von Software-Produkten  
– Teil 1: Qualitätsmodell*. Juni 2001

### **phpdoc.org 2016**

PHPDOC.ORG: *phpDocumentor-Website*. Version: 2016. <http://www.listjs.com/>, Abruf: 28.05.2016

### **Wikipedia 2016a**

WIKIPEDIA: *Model View Controller - Wikipedia*. Version: 2016. [https://de.wikipedia.org/wiki/Model\\_View\\_Controller](https://de.wikipedia.org/wiki/Model_View_Controller), Abruf: 28.05.2016

### **Wikipedia 2016b**

WIKIPEDIA: *PHP - Wikipedia*. Version: 2016. <https://de.wikipedia.org/wiki/PHP#Verbreitung>, Abruf: 30.05.2016

## Eidesstattliche Erklärung

Ich, Norman Paschke, versichere hiermit, dass ich meine **Dokumentation zur betrieblichen Projektarbeit** mit dem Thema

*Entwicklung eines Kalkulationskatalogs – Webanwendung zur Verwaltung von Kalkulationspositionen*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Chemnitz, den 30.05.2016

---

NORMAN PASCHKE

## A Anhang

### A.1 Detaillierte Zeitplanung

<b>Analysephase</b>	<b>1 h</b>
1. Analyse des Ist-Zustands	1 h
1.1. Fachgespräch mit den Mitarbeitern	1 h
<b>Entwurfsphase</b>	<b>5 h</b>
1. Prozessentwurf	2 h
2. ER-Modell der Datenbank erstellen	2 h
3. Benutzeroberflächen entwerfen	1 h
<b>Implementierungsphase</b>	<b>41,75 h</b>
1. Entwicklungs-Umgebung einrichten	4,25 h
1.1 Git-Repository anlegen	0,25 h
1.2 Datenbanktabellen anlegen	4 h
2. Erstellen einer Listenansicht jeder Komponente	9 h
2.1 Ausgabe aller relevanten Felder	7 h
2.2 Filter- und Sortiermöglichkeit für Listenansicht	2 h
3. Erstellen einer Einzelansicht jeder Klasse	20,5 h
3.1 Ausgabe aller relevanten Felder	8 h
3.2 Möglichkeit zum Bearbeiten jeder Klasse	5 h
3.3 Möglichkeit zum Löschen jeder Klasse	3,5 h
3.4 Möglichkeit zum Erstellen einer neuen Klasse	4 h
4. Berechnung des Gesamtaufwands einer Kalkulation	2 h
5. Umsetzung der HTML-Oberflächen und Stylesheets	6 h
<b>Testphase</b>	<b>7 h</b>
1. Testprotokoll erstellen	1 h
2. Durchführung der Tests durch Mitarbeiter	2 h
3. Reaktion auf Testergebnisse	4 h
<b>Dokumentation</b>	<b>15 h</b>
1. Erstellen der Projektdokumentation	10 h
1.1 $\LaTeX$ -Plugin für Eclipse installieren	0,5 h
1.2 $\LaTeX$ -Vorlage erstellen	0,5 h
1.3 Projektdokumentation verfassen, Korrektur	9 h
2. Programmdokumentation	1 h
3. Präsentation	4 h
<b>Pufferzeit</b>	<b>0,25 h</b>
1. Puffer	0,25 h
<b>Gesamt</b>	<b>70 h</b>

## A.2 Verwendete Ressourcen

### Hardware

- Büroarbeitsplatz mit Desktop-PC

### Software

- Windows 10 Pro 64 bit mit Service Pack 1 – Betriebssystem
- Eclipse Mars mit TeXlipse – Entwicklungsumgebung mit Erweiterung für L<sup>A</sup>T<sub>E</sub>X
- XAMPP – Apache-Distribution, die MariaDB, PHP und Perl enthält
- MariaDB – Datenbanksystem
- phpMyAdmin – Datenbankmanagementsystem
- phpDocumentor – Software zur Erstellung von Entwicklerdokumentationen
- git – dezentrale Versionsverwaltung
- MiKTeX – Distribution des Textsatzsystems T<sub>E</sub>X
- Gimp – Bildbearbeitungsprogramm
- Bootstrap – Framework für Oberflächen
- jQuery – JavaScript-Bibliothek
- List.js – Bibliothek zum Sortieren und Filtern von Listen im FE
- Balsamiq – Programm zur Erstellung von Wireframes
- Mozilla Firefox – Internetbrowser zum Betrachten von Webanwendungen

### Personal

- Projektmanagerin – Festlegung der Anforderungen und Abnahme
- Entwickler – Umsetzung des Projektes
- Anwendungsentwickler – Beratung



## A.3 Ausschnitt einer Excel-Tabelle

	A	B	C	D	E
1	Punkt	ToDo	Beschreibung	aufw	aufw
2		<b>Projektmanagement:</b> Projektmanagement zur konzeptionellen und technischen Projektumsetzung inkl. Zurverfügungstellung eines persönlichen Projektmanagers und eines Projektmanagementsystems.  Durchführung laufender telefonischer Abstimmungsgespräche und von 4 Vor-Ort-Meetings (Kick-Off-Workshop, lfd. Projektbesprechung) zwischen Auftragnehmer und Auftraggeber.	10% des Gesamtaufwands + 4 Vor-Ort-Meetings		
3					
4		<b>Projektkonzeption:</b> Erstellung von projektspezifischen Konzepten in enger Abstimmung mit dem Auftraggeber wie folgt:			
5		Erstellung eines Gestaltungskonzeptes	0,25-0,50 PT je Seitentyp	3	6
6		Erstellung eines technischen Konzeptes		2	2
7					
8		<b>Schulung</b> Durchführung einer Anwenderschulung für Redakteure zur Einweisung in die Bedienung von TYPO3 inkl. Redakteurleitfaden und Spesen.		2	2
9					
10		<b>Projektdokumentation:</b> Entwicklungsbegleitende Erstellung einer System- und Anwenderdokumentation zur Datenpflege und Nutzung des TYPO3-Systems.	5% des Entwicklungsaufwands		
11					
12		<b>User Interface Design:</b> Umsetzung des erarbeiteten Gestaltungskonzeptes auf Basis der Corporate Design-Vorgaben des Auftraggebers für insgesamt 11 definierte Seitentypen als JPG- und .PSD-Dateien unter Berücksichtigung der Prinzipien eines Responsive Webdesigns und unter Beachtung der Barrierefreiheit.	Startseite, Content-Liste, Content-Detail, Meldungen-Liste, Meldungen-Detail, Veranstaltungen-Liste, Formulare (Newsletteranmeldung, Kontaktformular), Bildergalerie, Website-Suche & Gemeindesuche, Visitenkarten  Startseite: 3 PT für 2 Entwurfsvarianten inkl. Korrekturschleife Unterseiten: 0,75 PT inkl. Korrekturschleife = 6,75 Puffer: 0,25 PT	10	12
13					
		HTML- und CSS-Programmierung			

Abbildung 2: Ausschnitt einer Excel-Tabelle einer Kalkulation

## A.4 Datenbankmodell

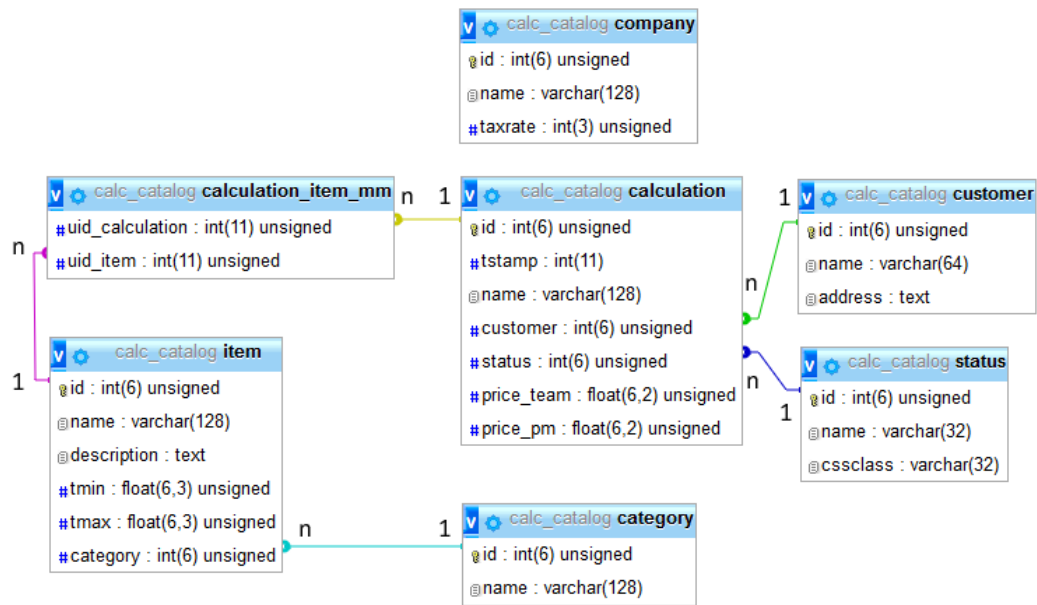


Abbildung 3: Datenbankmodell

## A.5 SQL - Auszug

```
1 CREATE TABLE 'item' (  
2   'id' int(6) unsigned NOT NULL auto_increment,  
3   'name' varchar(128) collate utf8_unicode_ci NOT NULL,  
4   'description' text(1000),  
5   'tmin' float(6,3) unsigned NOT NULL DEFAULT '0',  
6   'tmax' float(6,3) unsigned NOT NULL DEFAULT '0',  
7   'category' int(6) unsigned NOT NULL,  
8   PRIMARY KEY ('id'),  
9   CONSTRAINT 'fk_category' FOREIGN KEY ('category') REFERENCES category('id')  
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
    AUTO_INCREMENT=8;  
11  
12 INSERT INTO 'item' VALUES(1, 'Berechtigungen vergeben', 'Den Nutzern werden  
13 Rechte vergeben, damit sie ihre Aufgaben erledigen koennen.', 1.25, 2, 1);
```

Abbildung 4: Auszug aus der Datenbank in SQL-Format

## A.6 Entwürfe User Interface

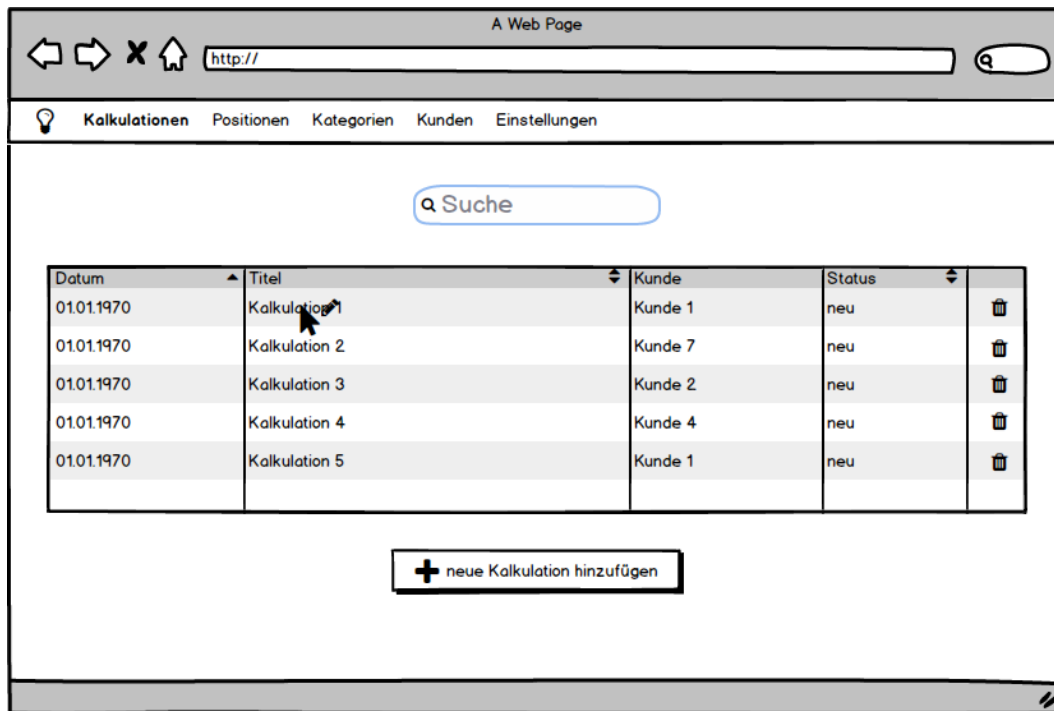


Abbildung 5: Liste der Kalkulationen mit Möglichkeit zum Filtern und Hinzufügen

A Web Page

http://

Kalkulationen Positionen Kategorien Kunden Einstellungen

## Kalkulation bearbeiten

Titel

Kunde

Status

Tagessatz Team  €

Tagessatz PM  €

Gesamtpreis  €

Positionen

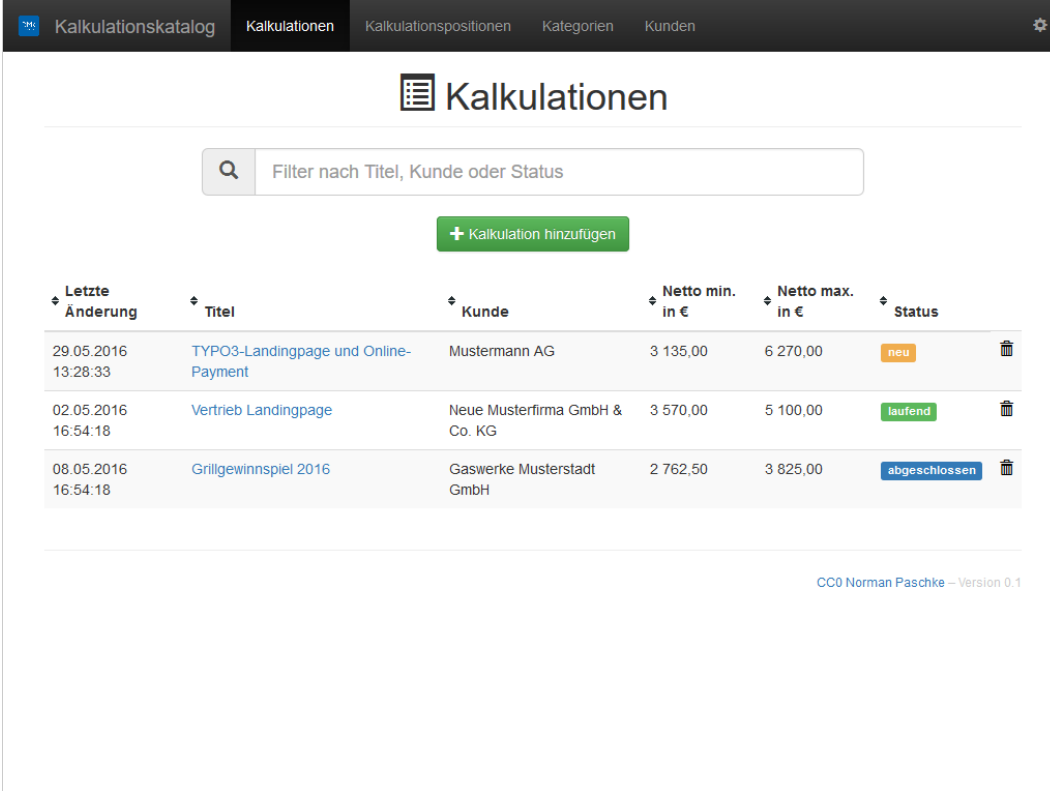
<b>Position 1</b> Beschreibung: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et 2 PT	2,5 PT
<b>Position 2</b> Beschreibung: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et 5 PT	7 PT

+ neue Kalkulation hinzufügen

✓ Speichern ✗ Abbrechen

Abbildung 6: Bearbeiten bzw. Hinzufügen einer neuen Kalkulation mit Positionen

## A.7 Screenshots der Anwendung



The screenshot shows the 'Kalkulationen' (Calculations) section of a software application. The interface includes a top navigation bar with tabs for 'Kalkulationskatalog', 'Kalkulationen' (selected), 'Kalkulationspositionen', 'Kategorien', and 'Kunden'. Below the navigation bar is a search bar with the placeholder text 'Filter nach Titel, Kunde oder Status'. A green button labeled '+ Kalkulation hinzufügen' is positioned above the table. The table itself has six main columns: 'Letzte Änderung', 'Titel', 'Kunde', 'Netto min. in €', 'Netto max. in €', and 'Status'. Each row represents a calculation entry, with the status column containing colored buttons: 'neu' (orange), 'laufend' (green), and 'abgeschlossen' (blue). Each row also has a trash icon for deletion.

Letzte Änderung	Titel	Kunde	Netto min. in €	Netto max. in €	Status
29.05.2016 13:28:33	<a href="#">TYPO3-Landingpage und Online-Payment</a>	Mustermann AG	3 135,00	6 270,00	neu
02.05.2016 16:54:18	<a href="#">Vertrieb Landingpage</a>	Neue Musterfirma GmbH & Co. KG	3 570,00	5 100,00	laufend
08.05.2016 16:54:18	<a href="#">Grillgewinnspiel 2016</a>	Gaswerke Musterstadt GmbH	2 762,50	3 825,00	abgeschlossen

CC0 Norman Paschke – Version 0.1

Abbildung 7: Liste der Kalkulationen

Kalkulationskatalog

Kalkulationen

Kalkulationspositionen

Kategorien

Kunden

## TYPO3-Landingpage und Online-Payment

**Titel**

**Kunde**

**Status**

**Tagessatz Entwicklung**
 €

**Tagessatz PM**
 €

**Brutto min.**
 €  
inkl. 10 % PM und 19 % Mehrwertsteuer

**Brutto max.**
 €  
inkl. 10 % PM und 19 % Mehrwertsteuer

**Kalkulationspositionen**

### TYPO3-Einrichtung

Berechtigungen vergeben

Den Nutzern werden Rechte vergeben, damit sie ihre Aufgaben erledigen können.

1,25 PT

2 PT

Seitenbaum anlegen

0,25 PT

1 PT

### Prototyping

Responsive Design

Prototypen müssen geräteunabhängig darstellbar sein

2 PT

4 PT

Barrierefreiheit

Prototypen sollen sich an BITV-Vorgaben halten und möglichst barrierearm sein.

2 PT

4 PT

+ Position hinzufügen

✓ Speichern

✗ Abbrechen

CC0 Norman Paschke – Version 0.1

Abbildung 8: Bearbeitung einer Kalkulation