



---

# **DIPLOMARBEIT**

---

Herr  
**Stefan Raschbacher**

**Datenbankentwicklung und  
Implementierung der  
Messgeräte-Datenimport-  
Schnittstelle**

2016



Fakultät Angewandte Computer- und  
Biowissenschaften

---

# **DIPLOMARBEIT**

---

## **Datenbankentwicklung und Implementierung der Messgeräte- Datenimport-Schnittstelle**

Autor:  
**Herr Ing.**

**Stefan Raschbacher**

Studiengang:  
**Technische Informatik**

Seminargruppe:  
**KT12wWA-F**

Erstprüfer:  
**Prof. Dr.-Ing. Rolf Hiersemann**

Zweitprüfer:  
**Dipl. Ing. Patrick Öhlinger**

---

Faculty of Applied Computer Sciences and  
Biosciences

---

## **DIPLOMA THESIS**

---

# **Database creation and deployment of the measuring import interface**

author:  
**Mr. Ing.**

**Stefan Raschbacher**

course of studies:  
**Technische Informatik**

seminar group:  
**KT12wWA-F**

first examiner:  
**Prof. Dr.-Ing. Rolf Hiersemann**

second examiner:  
**Dipl. Ing. Patrick Öhlinger**

## **Bibliografische Beschreibung:**

Raschbacher, Stefan:

Datenbankentwicklung und Implementierung der Messgeräte-Datenimport-Schnittstelle

V, 45, 3 S.

Mittweida, Hochschule Mittweida, Fakultät Angewandte Computer- und Biowissenschaften, Diplomarbeit, 2016

## **Referat:**

Das Ziel der Diplomarbeit ist die Datenbankentwicklung und Implementierung der Messgeräte-Datenimport Schnittstelle.

Der Benutzer soll für simple Messgerätedateien selbst eine Importschnittstelle erstellen können.

Neben der Implementierung der Schnittstelle für den Import von Messgerätedateien in die Labordatenbank soll auch eine graphisch aufbereitete Historie aller Importe sowie eine Verwaltung der bereits erstellten Schnittstellen umgesetzt werden.



# I. Inhaltsverzeichnis

I. Inhaltsverzeichnis.....	V
II. Abbildungsverzeichnis.....	VII
III. Tabellenverzeichnis.....	IX
IV. Abkürzungsverzeichnis.....	XI
1. Einleitung.....	1
1.1. Motivation.....	1
1.2. Zielsetzung.....	3
2. Grundlagen und Stand der Technik.....	5
2.1. LIMS Begriffsdefinition.....	5
2.2. SaaS Begriffsdefinition.....	6
2.3. CSV Begriffsdefinition.....	6
2.4. Datenbank Begriffsdefinition.....	8
2.5. Aufbau der Labordatenbank und verwendete Software.....	13
2.6. Recherche zum Stand der Technik.....	20
3. Präzisierung der Aufgabenstellung.....	22
3.1. Anforderungen an die Schnittstelle.....	22
3.2. Importschnittstellen Funktionen.....	24
3.2.1.Übersichts-Interface aller Schnittstellen.....	25
3.2.2.Konfiguration und Bearbeitung der Schnittstellen.....	25
3.2.3.Konfiguration und Bearbeitung der Parameterzuordnung.....	26
3.2.4.Importmaske mit Übersicht aller bisherigen Importe.....	27
3.2.5.Ansicht einzelner Importe einer Schnittstelle.....	27
3.3. Bisherige Schnittstellen einbinden.....	28
4. Systemkonzept.....	29
5. Softwareentwicklung und Implementierung.....	31
5.1. Installation der Entwicklungsumgebung.....	31
5.1.1.Auswahl des Servers.....	31
5.1.2.Installation Ubuntu Linux 14.04 LTS.....	31
5.1.3.Installation LAMP.....	32
5.1.4.Installation CakePHP.....	32
5.1.5.Installation Git.....	32
5.1.6.Installation Bootstrap.....	33
5.2. Einrichten der CakePHP Ordnerstruktur.....	33
5.3. Erstellen der Datenbanktabellen.....	35
6. Ergebnisse und Ausblick.....	39
6.1. Ergebnisse.....	39
6.2. Ausblick.....	42
Literaturverzeichnis.....	45

Anhang.....	47
Anhang A CakePHP Ordnerstruktur.....	47
Anhang B Crontab und Bash-Skript.....	48
Anhang C Erstellung der Datenbanktabellen in SQL.....	49
Selbstständigkeitserklärung.....	51



---

## II. Abbildungsverzeichnis

Abbildung 1: Grafik LIMS.....	5
Abbildung 2: Aufbau einer Tabelle (Steiner 2014).....	11
Abbildung 3: MVC Konzeptdarstellung (eigene Darstellung nach Kargl (2009))..	15
Abbildung 4: CakePHP MVC Struktur (Golding 2008).....	17
Abbildung 5: vereinfachte Datenstruktur der Labordatenbank (Öhlinger 2014)...	17
Abbildung 6: Schematische Darstellung der Messgeräteanbindung (eigene Darstellung).....	24
Abbildung 7: Use Case Diagramm Schnittstelle (eigene Darstellung).....	24
Abbildung 8: Datenbankstruktur der drei neuen Tabellen (eigene Darstellung)...	35
Abbildung 9: Schnittstellenübersicht (eigene Darstellung).....	39
Abbildung 10: Übersicht importierter Messdaten (eigene Darstellung).....	40
Abbildung 11: Zusammenfassung eines einzelnen Imports (eigene Darstellung)	40



### III. Tabellenverzeichnis

Tabelle 1 Beispiel CSV Datei (eigene Darstellung).....	7
Tabelle 2: importinterfaces Datenbanktabelle (eigene Darstellung).....	36
Tabelle 3: importmappings Datenbanktabelle (eigene Darstellung).....	37
Tabelle 4: Imports Datenbanktabelle (eigene Darstellung).....	38



---

## IV. Abkürzungsverzeichnis

<b>AWS</b>	Amazon Web Services
<b>Bash</b>	Bourne-Again Shell Benutzerschnittstelle unter Linux
<b>Cron</b>	zeitbasierte Ausführung von Prozessen/Programmen in Linux
<b>CSS</b>	Cascading Style Sheets. Beschreibt die Darstellung von HTML Elementen auf dem Bildschirm
<b>CSV</b>	Comma Separated Value (Komma getrennte Werte)
<b>GIT</b>	Git ist eine Software zur verteilten Versionsverwaltung von Dateien
<b>GUI</b>	Graphical User Interface (Grafische Benutzeroberfläche)
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol (Hypertext-Übertragungsprotokoll)
<b>jpg</b>	JPEG Joint Photographie Experts Group Bildkomprimierungsnorm nach ISO/IEC 10918
<b>JS</b>	JavaScript Skriptsprache als Erweiterung für HTML und CSS
<b>LAMP</b>	Linux, Apache, MySQL, PHP
<b>LDB</b>	Labordatenbank
<b>LIMS</b>	Labor-Informations- und Management-System
<b>LTS</b>	Long-Term Support
<b>MVC</b>	Model View Controller (Modell-Präsentation-Steuerung)
<b>pdf</b>	Portable Document Format
<b>PHP</b>	PHP: Hypertext Preprocessor
<b>QS/QK</b>	Qualitätssicherung und Qualitätskontrolle
<b>SaaS</b>	Software as a Service (Software als Dienstleistung)
<b>SQL</b>	Structured Query Language, eine Datenbanksprache
<b>URL</b>	Uniform Resource Locator
<b>VM</b>	Virtuelle Maschine
<b>WWW</b>	World Wide Web



# 1. Einleitung

Im folgenden Kapitel werden die Motivation und Aufgabenstellung der Diplomarbeit besprochen.

Hintergrund der Firma LDB Labordatenbank GmbH (LDB). Die 2006 gegründete LDB Labordatenbank GmbH hat sich auf die Entwicklung, Bereitstellung und Betreuung von Labor-Informationssystemen spezialisiert. Seit 2007 ist die Labordatenbank in akkreditierten Laboren in den Bereichen QS/QK, Lebensmittel, Trinkwasseranalytik, Mikrobiologie und Veterinärmedizin erfolgreich im Einsatz.

Sie gilt als schnell wachsendes Unternehmen das inzwischen 50 Laborbetriebe und Firmen in Deutschland, Österreich, Schweiz, Japan und USA zu seinen Kunden zählt.

## 1.1. Motivation

Seit einigen Jahren ist in den Medien immer öfter vom Thema „Big Data“ zu lesen. Durch die zunehmende Digitalisierung der Geräte werden vermehrt Daten produziert die dann auch verarbeitet und gespeichert werden müssen. Durch die im Laufe der Zeit gesammelten Daten und Informationen können auch neue Korrelationen hergestellt werden die zuvor einfach nicht möglich waren. Als Beispiel ist hier eine Materialproduktion, in dem der Bereich Qualitätssicherung und Qualitätskontrolle (QS/QK) eine sehr große Rolle spielt, zu nennen. Hier können die genaue Menge und Beschaffenheit der Inhaltsstoffe

mit den angesammelten Maschinendaten während der Verarbeitung in Zusammenhang gebracht werden.

Die Auswertungen und Schlüsse, die aus den Daten gezogen werden können, sind umso aussagekräftiger je mehr Daten über einen längeren Zeitraum gesammelt werden.

Wenn diese Daten schon vorhanden sind, sollten sie auch möglichst optimal genutzt werden.

Dies erfordert jedoch sehr viel Zeit, beziehungsweise ist es teilweise einfach unmöglich, diese Daten alle manuell in die Labordatenbank einzugeben.

Glücklicherweise liefern schon sehr viele Geräte und Maschinen diese Daten in digitaler Form.

Es ist ein enormer Zeitaufwand bei jedem neuen Kunden die Eigenheiten und Einstellungen der jeweiligen Messgeräte zu evaluieren und die Importfunktion der Labordatenbank dementsprechend im Hintergrund anzupassen.

Viele Kunden möchten sich mit dem Thema Importschnittstelle beschäftigen und sind durchaus auch bereit sich darum zu kümmern. Dies ist aber derzeit leider nicht möglich, da jede Schnittstelle für sich individuell angepasst und in der Labordatenbank programmiert werden muss.

Dieser Anpassungsprozess bei der Projekteinführung der Labordatenbank dauert oft mehrere Monate und erst dann kann der Messdaten-Import im Produktionsbetrieb verwendet werden.

Der Autor dieser Arbeit ist bei dem Unternehmen LDB Labordatenbank GmbH beschäftigt und wurde mit der Umsetzung und Implementierung der



Importschnittstellen, Erstellung von individualisierten Berichten und Befunden sowie der Betreuung der Serverlandschaft beauftragt. Als betrieblicher Betreuer stellte sich der Unternehmensgründer Dipl. Ing. Patrick Öhlinger zur Verfügung.

## 1.2. Zielsetzung

Mit dieser Diplomarbeit soll eine neues Interface und die dahinterliegende Datenstruktur in der Datenbank für die Importschnittstelle geschaffen werden. Eine Recherche welche Importschnittstellen sich für die Umsetzung eignen und ein erster grafischer Prototyp für die Importschnittstelle wurde bereits im Praxisprojekt entwickelt.

Die neue Datenbankstruktur soll ein leicht erweiterbares Fundament für die Weiterentwicklung der Importschnittstelle sein, welches Platz für Erweiterungen bietet.

Weiters sollen mindestens zehn der bisher eigens entwickelten Importschnittstellen in das neue System migriert werden. Dadurch profitieren dann auch die bestehenden Kunden von den Vorteilen des neuen Systems.

Die erwähnten Schnittstellen betreffen (auszugsweise) Geräte wie MALDI Biotyper System - MALDI-TOF Massenspektrometer, photoLab® 7600 UV-VIS - Spectrophotometer<sup>1</sup>, Multi 9430 - Digitales Messgerät für IDS Sensoren<sup>2</sup>, BioChek ELISA Assay Robot - veterinärmedizinische Untersuchungen, ICP-OES Spectrometer und verschiedenste Typen der Firma Zwick für

---

<sup>1</sup> <http://www.wtw.com/de/produkte/produktklassen/labormessgeraete/filter-und-spektralphotometer/photolabr-7600-uv-vis.html?mobile=0>

<sup>2</sup> <http://www.wtw.com/de/produkte/produktklassen/labormessgeraete/inolabr-labormessgeraete/inolabr-multi-9430-ids.html?mobile=0>

Materialstoffprüfung, die über eine USB-B Datenschnittstelle beziehungsweise Ethernet angeschlossen werden.

Die verwendeten Skript- beziehungsweise Auszeichnungssprachen werden PHP und HTML sein. Diese sind weitgehend vom CakePHP Framework vorgegeben. Als Datenbank im Backend wird das Datenbankverwaltungssystem MySQL verwendet. Die Labordatenbank ist an dieses System mittels CakePHP angebunden.

Herangehensweise an die Umsetzung:

1. Recherche und Analyse der bisherig verwendeten Datenbank
2. Recherche nach bereits vorhandenen und alternativen Lösungen
3. Evaluierung möglicher und für den Kunden relevanter Lösungsansätze
4. Vorlage der Lösungsansätze an die Geschäftsführung
5. Recherche zum Stand der Technik und dessen Bewertung
6. Präzisierung der Aufgabenstellung
7. Systemkonzipierung
8. Erstentwurf
9. Umsetzung
10. Tests
11. Abnahme durch den Kunden

## 2. Grundlagen und Stand der Technik

### 2.1. LIMS Begriffsdefinition

Ein Labor-Informations- und Management-System (LIMS) ist ein softwarebasiertes System mit einer Datenbank im Hintergrund zur Verwaltung von allen für das Labor relevanten Daten.

Dazu gehören unter anderem:

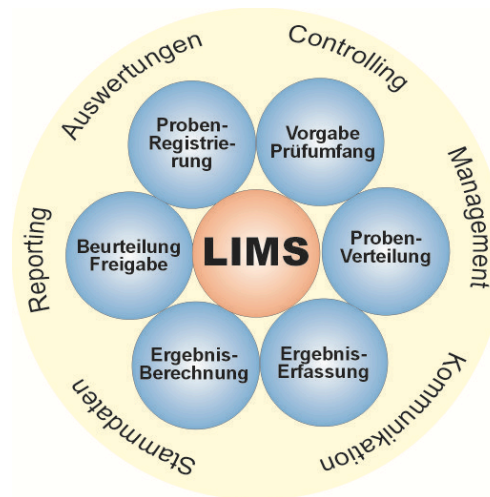


Abbildung 1: Grafik LIMS

*LIMS Applikationen*<sup>3</sup>:

- *Probeneingang und Probenregistrierung*
- *Festlegung des Untersuchungsumfangs*
- *Unterstützung bei Probenverteilung und Bearbeitung*
- *Ergebnisdatenerfassung*
- *Berechnung von Ergebnissen bzw. Grenzwertprüfungen*
- *Freigabe der Prüfergebnisse*
- *Berichtswesen und Auswertungen*

<sup>3</sup> <http://www.imcor.de/de/labor-it/labor-it-lösungen/lims-intro> (WWW Dokument, WWW Abbildung, abgerufen am 29. April 2016)

## 2.2. SaaS Begriffsdefinition

Software as a Service (SaaS) ist einer der vielen Diensttypen beim Cloud Computing<sup>4</sup>.

*Bei SaaS greifen Nutzer zumeist mit einem Web Browser direkt auf eine Anwendung zu.*

(Appelrath, 2014)

Es muss also nicht auf jedem Arbeitsplatz eine eigene Software installiert werden, sondern es genügt ein moderner Web Browser mit dem auf das LIMS System zugegriffen wird.

Die Labordatenbank bietet hier zwei Betriebsmodelle an um die Software zu benutzen.

Bei Variante eins läuft die Labordatenbank als Cloud Service im Amazon Web Services (AWS) Rechenzentrum in Frankfurt und die Nutzer greifen mittels einer SSL verschlüsselten Internetverbindung auf die Labordatenbank zu.

Bei Variante zwei ist eine Virtuelle Maschine (VM) im Netzwerk des Kunden.

## 2.3. CSV Begriffsdefinition

Comma Separated Value (CSV)-Dateien sind bei Messgeräten verschiedenster Hersteller ein de facto Standard zum Austausch tabellenartig dargestellter Informationen (Messdaten).

Leider gibt es keinen allgemein gültigen Standard welcher das Dateiformat CSV beschreibt.

---

<sup>4</sup> Cloud Computing: Die Ausführung von Programmen die nicht lokal am Rechner installiert sind aus zum Beispiel dem Internet.

Microsoft Excel zum Beispiel speichert CSV Dateien mit einem Semikolon als Trennzeichen ab.

Der Dateiaufbau einer CSV Datei sieht dabei z.B. wie folgt aus:

```
Probe,Prüfer,Datum,Gewicht
A16768,Max Mustermann,11.04.2016,"13,739"
A16769,Hermine Mustermann,12.04.2016,"15,335"
A16770,Klaus Müller,13.04.2016,"14,102"
A16771,Hermine Mustermann,14.04.2016,"13,001"
A16772,Max Mustermann,15.04.2016,"14,523"
```

Diese Datei sieht in tabellarischer Darstellung dann wie folgt aus:

Probe	Prüfer	Datum	Gewicht
A16768	Max Mustermann	11.04.2016	13,739
A16769	Hermine Mustermann	12.04.2016	15,335
A16770	Klaus Müller	13.04.2016	14,102
A16771	Hermine Mustermann	14.04.2016	13,001
A16772	Max Mustermann	15.04.2016	14,523

Tabelle 1 Beispiel CSV Datei (eigene Darstellung)

Innerhalb dieser Datei haben manche Zeichen einen Sonderstatus.

Bei diesem Beispiel hat das Semikolon die Funktion die einzelnen Spalten voneinander zu trennen.

Die Anführungszeichen zum Beispiel kennzeichnen jeweils den Anfang und das Ende der einzelnen Spalten. Es können somit auch die definierten Trennzeichen innerhalb der jeweiligen Spalte verwendet werden. Diese Anführungszeichen werden selten von Messgeräten verwendet, dadurch gestaltet sich das Erkennen von Dezimalzahlen mit verschiedenen Ländereinstellungen leider sehr schwierig oder ist teilweise gar nicht möglich.

Bei unterschiedlichen Ländern gibt es bei Datums-, Zeit- und Dezimalwerten meist unterschiedliche Trennzeichen.

Beispiel 26.04.2014 in Deutschland entspricht 04/26/2014 in der USA.

Dies erschwert die automatische Erkennung von Zeit und Datumswerten für den Import in die Datenbank und erfordert einige Konvertierungen vor der Speicherung.

Auch gibt es bei den verschiedenen Messgeräten unterschiedliche Formen der Zahlendarstellung und Schreibweise.

Ein Beispiel hierfür wäre 48.8E+3 und 24,3.

In Deutschland 48800 und 24,3

In der Schweiz 48'800 und 24.3

Weiters sind es je nach Region 1000er Trennzeichen und Punkt oder Komma als Dezimaltrennzeichen üblich.

## 2.4. Datenbank Begriffsdefinition

*Eine Datenbank ist eine selbständige und auf Dauer ausgelegte*

*Datenorganisation, welche einen Datenbestand sicher und flexibel verwalten kann (Steiner 2014).*

*Eine Datenbank hat folgende Aufgaben nach Steiner (2014):*

- *Sie soll dem Benutzer den Zugriff auf die gespeicherten Daten ermöglichen, ohne dass dieser wissen muss, wie die Daten im System organisiert sind.*

- *Sie muss verhindern, dass ein Benutzer Daten sichten oder manipulieren kann, für die er keine Zugriffsberechtigung hat. Außerdem darf es nicht passieren, dass wegen Fehlmanipulationen des Benutzers Daten zerstört werden können oder gar der ganze Datenbestand unbrauchbar wird.*
- *Es muss möglich sein, die interne Datenorganisation ändern zu können, ohne dass der Benutzer seine Anwenderprogramme (Applikationen) anpassen muss. Im Idealfall merkt der Benutzer von der Strukturänderung nichts.*

Erklärung der wichtigsten Datenbankbegriffe nach Steiner (2014):

*Die Datenbankbegriffe lassen sich aufteilen in Begriffe der Datenbanktheorie und der Informatik. Bei den folgenden Begriffserklärungen werden zuerst die Bezeichnungen im konzeptionellen Datenmodell und in Klammern die Bezeichnungen im physischen Datenmodell aufgeführt. Die Definitionen dieser beiden Begriffe folgen*

*Entität (Tabellenname): Eine Entität stellt einen Themenkreis dar, welcher Elemente mit gleichen Merkmalen umfasst. Beispiel: Personen, Kurse, Ersatzteile etc.*

*Entitätsmenge (Datensätze): Die Entitätsmenge beinhaltet alle zu den Merkmalen einer Entität gehörenden Werte. Sie entspricht allen gespeicherten Datensätzen einer Tabelle.*

*Relation (Tabelle): Eine Relation umfasst eine Entität mit der dazugehörigen Entitätsmenge. Man versteht darunter eine komplette Tabelle mit Entitätsbezeichnung, Attributen und Tupel.*

*Tupel (Datensatz): Ein Tupel umfasst alle Merkmalswerte eines Elementes als Bestandteil einer Entitätsmenge. Ein Tupel entspricht einem vollständigen Datensatz. Beispiel: Müller, Hugo, Planetenweg 7, 1234 Neustadt. Alle Tupel einer Entität zusammen bilden die Entitätsmenge.*

*Attribut (Spaltenname): Das Attribut entspricht einem Merkmal eines Tupels und beschreibt somit eine spezifische Eigenschaft einer Entitätsmenge. Beispiel Name, Adresse, Alter etc.*

*Attributwert (Wert, Datum): Dies ist ein Datenwert, welcher das zugehörige Attribut eines Tupels beschreibt. Beispiel: Attribut = Name; Attributwert = Müller. Vielfach wird anstatt des Wortes „Wert“ der Begriff „Datum“ verwendet. Datum und Wert sind Synonyme, haben aber mit dem Kalenderdatum nichts zu tun.*

*Domäne (Wertebereich): Gewisse Attribute schränken die zugehörigen, möglichen Attributwerte ein. Das Attribut „Wochentag“ lässt beispielsweise nur*



die Attributwerte *Mo, Di, Mi, Do, Fr, Sa* und *So* zu. Diese Wertebeschränkung nennt man *Domäne* oder *Wertebereich*.

*Nullwerte*: Wenn ein Attribut eines Tupels einen Nullwert enthält, so bedeutet dies, dass dieses Attribut keinen Attributwert besitzt und somit keine Information beinhaltet. Der Nullwert darf nicht mit der Zahl Null verwechselt werden. Die Zahl Null stellt eine Information dar, der Nullwert jedoch nicht.

Die bisher beschriebenen Begriffe lassen sich am einfachsten mit einer Tabelle darstellen, wie dies Abbildung 2 zeigt.

### Tabelle (Relation) Personen:

Attribute →

Tupel (Datensatz) →

Personen ← Entität (Tabellenname)

PNr.	Name	Vorname	Größe	Geschlecht
1234	Müller	Hans	182	m
5634	Suter	Ernst		m
2456	Tarelli	Claudia	170	w
1123	Brunner	Diana	172	w

↑  
Attributwerte

Abbildung 2: Aufbau einer Tabelle (Steiner 2014)

Definition der verschiedenen Datenbankmanipulationsarten nach Steiner

(2014):

*Beim späteren Datenbankbetrieb muss der Datenbestand manipuliert werden. Es ist deshalb nötig, alle zulässigen Manipulationsarten (Transaktionen) und deren Ablauf klar zu definieren, wobei die Datenkonsistenz erhalten bleiben muss.*

*Bei einer Transaktion handelt es sich um eine Operation auf den Datenbestand, wobei es folgende Möglichkeiten gibt:*

- *Daten abfragen (Query)*
- *Daten einfügen (Insert)*
- *Daten nachführen (Update)*
- *Daten löschen (Delete)*

*Das Einfügen und Nachführen von Daten kann unter dem Begriff Datenmutation zusammengefasst werden. Unter den Begriff Datenmanipulation fallen alle Operationen auf den Datenbestand, welche aber im Gegensatz zu den Transaktionen nicht notwendigerweise konsistenzhaltend sein müssen. Eine Transaktion kann auch aus mehreren Operationsschritten bestehen.*

## 2.5. Aufbau der Labordatenbank und verwendete Software

Die Labordatenbank benötigt als Grundsystem eine typische Linux Apache<sup>5</sup> MySQL<sup>6</sup> PHP<sup>7</sup> (kurz LAMP) Installation. Die aktuelle im Einsatz befindliche Linux Version ist Ubuntu Server 14.04 LTS<sup>8</sup>. LTS steht dabei für Long Term Support. Diese Version wird bis April 2019 mit Sicherheitsupdates unterstützt. Auf dem bereits installierten Ubuntu Server wird ein Apache Webserver installiert.

Für die Datenbank wird anschließend noch eine MySQL Datenbank hinzugefügt.

Als Web-Skriptsprache für die dynamische Erstellung der Seiten in Labordatenbank wird PHP verwendet.

Als Software Versionierungs- und Verwaltungssystem ist das auf Git<sup>9</sup> aufbauende System GitHub<sup>10</sup> im Einsatz. Dieser Online-Service speichert den Quellcode der Labordatenbank und bietet einen gesicherten Zugang über die Cloud. So können alle Mitarbeiter, egal wo auf der Welt sie gerade für das Unternehmen tätig sind, an der Entwicklung der Labordatenbank mitarbeiten.

Um eine strukturierte Softwareentwicklung zu haben, baut die Labordatenbank auf das Model View Controller System (MVC) auf.

---

5 Webserver <http://httpd.apache.org>

6 Datenbanksoftware <http://www.mysql.com>

7 PHP: Hypertext Preprocessor <http://php.net>

8 Ubuntu Server 14.04 LTS <http://releases.ubuntu.com/14.04/>

9 Versionsverwaltungssoftware <https://git-scm.com>

10 Versionsverwaltungssoftware <https://github.com>

Dabei wird das Architekturmuster in drei Komponenten aufgeteilt, die je nach Einsatz unterschiedlich stark voneinander abhängig sind.

Aufbau des MVC Systems nach Karl (2009):

*Die Modell-Rolle: Das Modell ist ein nicht-visuelles Objekt, das Informationen der Domäne darstellt. Alle Methoden, die nicht für die Benutzerschnittstelle verwendet werden, sind enthalten.*

*Die View-Rolle: Die View stellt die visuelle Anzeige des Modells in der Benutzerschnittstelle dar. Diese ist ausschließlich mit der Anzeige von Informationen befasst. Alle Änderungen der Informationen werden von der dritten Rolle verarbeitet: dem Controller.*

*Die Controller-Rolle: Der Controller nimmt die Eingaben des Benutzers entgegen (daher als Input-Controller bezeichnet), bearbeitet das Modell und sorgt für Aktualisierung der Anzeige. Die Benutzerschnittstelle ist folglich eine Kombination aus View- und Controller-Rolle und dient zur Interaktion mit den Benutzern.*

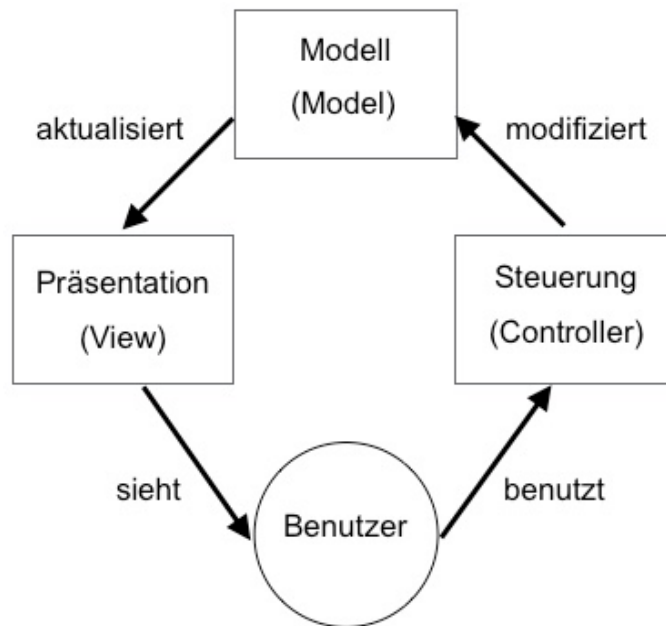


Abbildung 3: MVC Konzeptdarstellung (eigene Darstellung nach Kargl (2009))

Die Labordatenbank verwendet das CakePHP<sup>11</sup> Framework, welches einige der MVC Funktionalitäten abbildet.

Wie das CakePHP Framework die MVC Struktur benutzt soll im folgenden erläutert werden (Golding 2008).

1. Der Client sendet eine Seitenaufrufanforderung an die Anwendung, entweder indem Sie eine URL eingeben oder indem Sie einen Link klicken. Nach der CakePHP Konvention wird eine typische URL wie folgt strukturiert:

`http://{Domain}.com/{Application}/{Controller}/{Action}/{Parameter , etc.}`

<sup>11</sup> <http://cakephp.org>

2. Das dispatcher Skript parst die URL-Struktur und bestimmt, welche Controller auszuführen ist. Er liefert dem jeweiligen Controller auch alle notwendigen Parameter.
3. Die Funktion im Controller benötigt eventuell mehr Daten als die Parameter, die von den dispatcher Script weitergeleitet wurden. Es werden Datenbankankorderungen an das Modell Skript gesendet.
4. Das Modell Skript legt durch die vom Controller übermittelten Anfragen fest, wie mit der Datenbank kommuniziert wird. Es kann einfache Abfragen jedoch aber auch viele komplexere Datenmanipulationsanweisungen ausführen.
5. Sobald das Modell alle Daten an die Datenbank gesendet beziehungsweise empfangen hat, wird das Ergebnis umgehend an den Controller übermittelt.
6. Der Controller verarbeitet die Daten und stellt diese zur Präsentation bereit.
7. Die Präsentation fügt etwaige Designs hinzu oder übermittelt Daten an den Controller und liefert das Ergebnis an den Web-Browser des Benutzers.

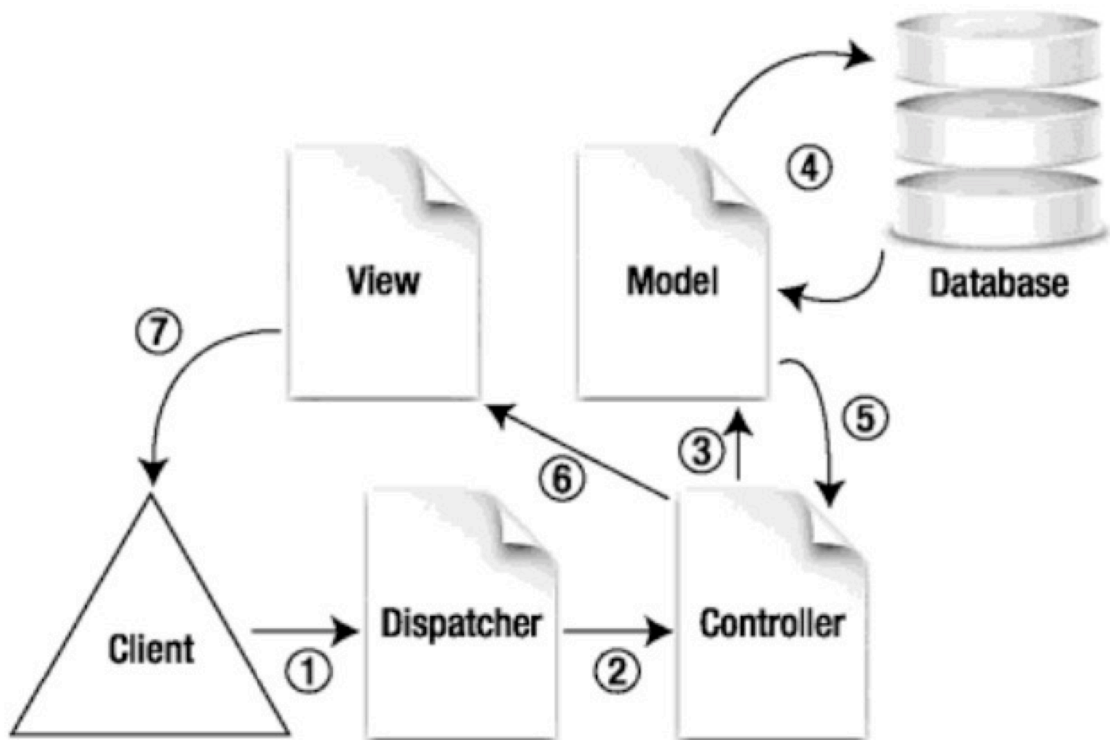


Abbildung 4: CakePHP MVC Struktur (Golding 2008)

Die Labordatenbank basiert auf folgenden Datenstruktur, die auch in der MySQL-Datenbank abgebildet ist. nach Öhlinger (2014)

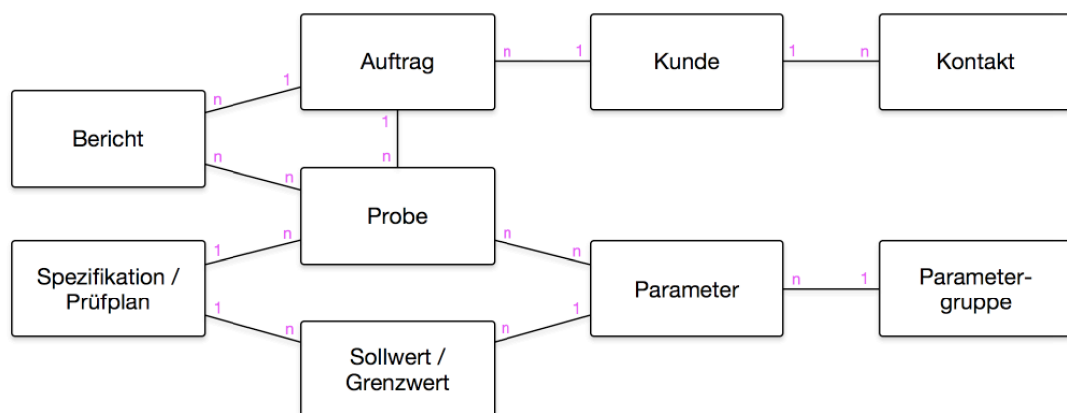


Abbildung 5: vereinfachte Datenstruktur der Labordatenbank (Öhlinger 2014)

Das obige Diagramm zeigt eine vereinfachte Darstellung der Datenstruktur der Labordatenbank. (Für diese Darstellung nicht benötigte Tabellen wie Mitarbeiter, Zugriffsrechte, Angebote, Rechnungen, Rezepturen, Notizen, E-Mails, und viele andere wurden zwecks übersichtlicherer Darstellung ausgeblendet.)

Erklärung der Datenbeziehungen:

- Ein Kunde hat  $n$  Kontakte und  $n$  Aufträge ( $n = 0, 1$  oder mehrere).
- Ein Auftrag gehört zu einem Kunden, hat  $n$  Proben und  $n$  Berichte.
- Eine Probe gehört zu einem Auftrag, hat eine Spezifikation / Prüfplan und  $n$  Parameter.
- Ein Bericht gehört zu einem Auftrag und hat  $n$  Proben.
- Eine Spezifikation / Prüfplan hat  $n$  Sollwerte die jeweils für einen Parameter gelten.
- Ein Parameter kommt in  $n$  Proben vor, hat  $n$  Sollwerte und gehört zu einer Parametergruppe.
- Eine Parametergruppe hat  $n$  Parameter.

Bisherige Umsetzung:

Derzeit erfolgt der Import von CSV Dateien über eine für jeden Kunden eigens angepasste Importschnittstelle. Das Unternehmen benötigt für jede Schnittstelle vorab einige Testdateien um die Art der von dem jeweiligen Gerät verwendeten Dateistruktur feststellen zu können.



Für den Import wird die Datei soweit mittels PHP Programmierung aufbereitet, dass die für den Import notwendigen Informationen in einem Array gesammelt werden. Dieser Array wird danach verwendet um die Daten in der Datenbank abzuspeichern.

Diese Informationen umfassen Datensätze mit zum Beispiel Probennummer, Messparameter, Messwerte, Messdatum und Mitarbeitername. Das Format der von den Messgeräten gelieferten Datensätze ändert sich in der Regel nicht laufend.

Beispiel:

- Welche Probe (Proben-ID der Datenbanktabelle samples)
- Welcher Parameter (Parameter-ID der Datenbanktabelle parametertypes)
- Welches Ergebnis (Ergebniswert in welcher Spalte der Datenbanktabelle parameters)
- Welcher Mitarbeiter (Mitarbeiter-ID in der Datenbanktabelle employees)
- Welcher Zeitpunkt (UNIX-Zeitstempel in Datenbanktabelle parameters)

## 2.6. Recherche zum Stand der Technik

Die derzeit am Markt befindliche Programme für Datenbankmanagement gibt als online sowie auch als offline Versionen. (online Beispiele: phpMyAdmin<sup>12</sup>, offline Beispiele: SequelPro<sup>13</sup>, MySQL Workbench<sup>14</sup>).

Diese Programme sind aber mehr zur Administration und Verwaltung von ganzen Datenbanksystemen gedacht und weniger um CSV Dateien in eine Datenbank einzuspielen. Es gibt hier zwar vereinzelt die Möglichkeit CSV Dateien zu Importieren, dabei ist die Bedienung aber leider sehr umständlich. Außerdem wird hier vorausgesetzt, dass der Benutzer mit der Datenbankstruktur vertraut ist und auch der Administrator der jeweiligen Datenbank ist. Oft fehlen auch für den Benutzer notwendige grafische Oberflächen, damit sich dieser besser zurecht findet. Auch muss meist mit einer für den Kunden ungewohnten Kommandozeilenumgebung gearbeitet werden. Ein zusätzlicher Nachteil ist, dass diese Softwareprodukte keine bis wenig Konfigurationsmöglichkeiten bieten. Die einzelnen Spalten der CSV Dateien müssen den jeweiligen Attributen der Tabelle zugeordnet werden. Leider gibt es hier keinerlei Möglichkeit die Zuordnung für den nächsten Import zu speichern. So muss beim nächsten Import komplett von vorne begonnen werden. Ein weiterer großer Nachteil ist, dass ein direkter Datenbankzugang mit Insert- und Update-Rechten notwendig ist. Dadurch gibt es natürlich entsprechende Sicherheitsbedenken, denn durch den Import könnten unabsichtlich auch

---

12 <http://www.phpmyadmin.net>

13 <http://sequelpro.com>

14 <http://mysqlworkbench.org>

andere interne Labordatenbank Daten überschrieben werden, was die einwandfreie Funktion der Labordatenbank selbst gefährdet.

Diese Importe und Änderungen werden außerdem auch nicht im eingebautem Änderungsreport (Audit-trail) der Labordatenbank mitprotokolliert. Diese Reports sind für die Labore aber notwendig, da bei diversen Audits nachgewiesen werden muss, wer, wann, welche Werte in die Labordatenbank eingetragen, beziehungsweise importiert hat.

## 3. Präzisierung der Aufgabenstellung

### 3.1. Anforderungen an die Schnittstelle

Für die Importschnittstelle ergeben sich folgende Anforderungen:

Auswahl der Werkzeuge für die Umsetzung.

Ein Übersichts-Interface für die Auswahl aller verfügbaren Schnittstelle.

Ein Interface zur Übersicht für die jeweils ausgewählte Schnittstelle.

Eine Bearbeitungsmaske um die Eigenschaften der verschiedenen Schnittstellen zu bearbeiten.

Eine Bearbeitungsmaske um die Parameterzuordnung zu ändern oder zu erweitern.

Nach erfolgreichem Import soll eine Zusammenfassung des Imports mit den gespeicherten Werten auch im Nachhinein noch abrufbar sein.

Einlesen von Schnittstellen CSV-Testdateien.

Speichern der eingelesenen Daten in der Datenbank.

Es wird außerdem laufend mitprotokolliert wer, wann, welche Werte und welche Proben importiert.

Funktionale Anforderungen:

- Der Benutzer muss selbstständig neue Importschnittstellen anlegen können.
- Beim Speichern der Schnittstelleneinstellungen sollen die ausgewählten Informationen in der Datenbank gespeichert werden.

- Benutzer die keinen Zugriff auf eine Schnittstelle haben, sollen diese auch nicht im Schnittstellenmenü angezeigt bekommen.
- Benutzer welche einen Zugriff auf die Schnittstelle haben, sollen diese im Schnittstellenmenü jedoch angezeigt bekommen.
- Benutzer müssen die Möglichkeit haben Schnittstellen mit einem Knopf in Bleistiftform zu bearbeiten.
- Ein Import muss eindeutig mit einer fortlaufenden Nummer gekennzeichnet sein.
- Ein Import muss einen Link enthalten der auf die jeweiligen Importinformationen verweist.
- Die Importinformationen müssen die Rohdaten der hochgeladenen CSV Datei enthalten.
- Die importierten Werte eines Imports müssen in den Importinformationen als Tabelle angezeigt werden.
- Der angemeldete Benutzer, der den Import durchführt, muss als Benutzer in der Importinformation angezeigt werden.

Nicht funktionale Anforderungen:

- Die Anwendung muss mit PHP, HTML und dem CakePHP Framework entwickelt werden.
- Die Benutzeroberfläche muss bedienungsfreundlich und intuitiv sein.
- Der Import einer Schnittstellendatei soll weniger als eine Minute dauern.
- Es sollen 5 Benutzer gleichzeitig Schnittstellendaten importieren können.

- Das Produkt muss auf den letzten beiden aktuellen Versionen von Firefox, Chrome, Safari und Internet Explorer lauffähig sein.

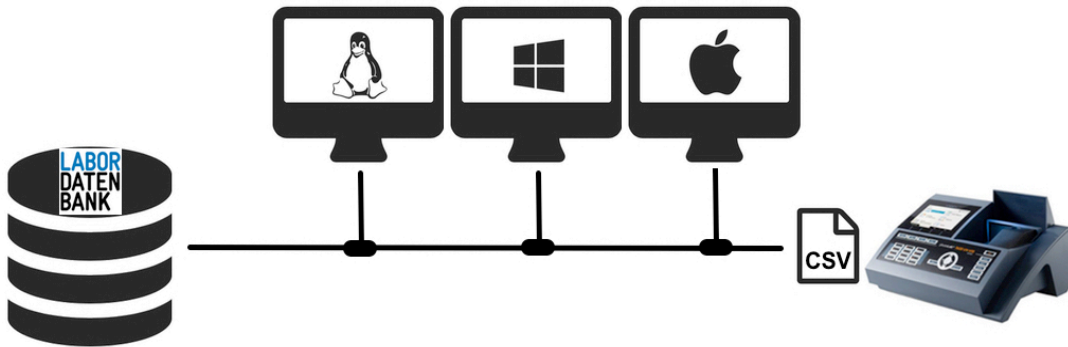


Abbildung 6: Schematische Darstellung der Messgeräteanbindung (eigene Darstellung)

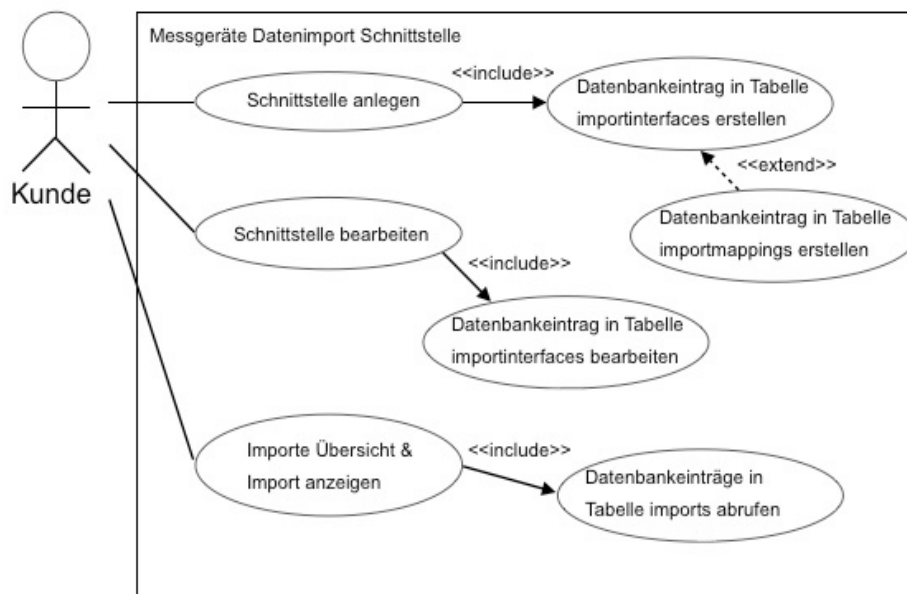


Abbildung 7: Use Case Diagramm Schnittstelle (eigene Darstellung)

## **3.2. Importschnittstellen Funktionen**

Allgemeiner Überblick über die Menüebenen.

### **3.2.1. Übersichts-Interface aller Schnittstellen**

In dieser Übersicht muss ein Knopf zum Anlegen neuer Schnittstellen vorhanden sein.

Diese Ansicht muss einen Überblick aller bereits konfigurierten Schnittstellen geben.

Das Interface muss bei jeder Schnittstelle einen Link zum Bearbeiten der vorhandenen Schnittstellen bereitstellen.

Beim Auswählen der Schnittstelle durch Klicken auf den Namen der Schnittstelle, gelangt man in die Import-Maske.

### **3.2.2. Konfiguration und Bearbeitung der Schnittstellen**

Für die Konfigurationsmaske der Schnittstelle müssen folgende Punkte erfüllt werden:

Dem Benutzer muss mittels eines Eingabefeldes die Vergabe einer individuellen Bezeichnung für die Schnittstelle ermöglicht werden.

Der Benutzer muss mittels eines Auswahlfeldes einstellen können, welche Trennzeichen seine CSV Datei enthält.

Auch muss der Benutzer anhand eines Auswahlfeldes festlegen können, in welcher Zeile sich die Attributszeile der CSV Datei befindet.

Der Benutzer muss mittels Auswahlfeld auswählen können, wie mit den importierten Daten umgegangen werden soll, wenn eine komplette Probe oder einzelne Parameter einer Probe bereits Werte enthält.

Folgende Importmodi sind derzeit in Verwendung.

Diese Beschreibung kann im Weiteren auch gleich Anleitung in der Eingabemaske angezeigt werden:

- **insert:** Wenn der Parameter bereits existiert und leer ist, werden die neuen Werte eingefügt, sonst wird der Parameter angelegt und die Werte gespeichert.
- **ignore:** Wenn der Parameter bereits existiert und leer ist, werden die neuen Werte eingefügt. Wenn der Parameter bereits existiert und bereits Werte hinterlegt sind, werden die neuen Werte ignoriert. Wenn der Parameter noch nicht existiert, wird der Parameter angelegt und die neuen Werte gespeichert.
- **ignore\_without\_insert:** Wenn der Parameter bereits existiert und leer ist, wird der neue Wert eingefügt. Wenn der Parameter bereits existiert und etwas eingetragen ist, wird der neue Wert ignoriert. Wenn der Parameter noch nicht existiert, wird der Parameter nicht angelegt.
- **update:** Schaut ob es den Parameter bereits gibt, wenn ja wird er überschrieben. Sonst wird der Parameter angelegt.
- **update\_without\_insert:** Der Wert wird nur importiert wenn der Parameter bei der Probe ausgewählt ist. Ist bereits etwas eingetragen, wird der eingetragene Wert überschrieben.

Dem Benutzer soll auch die Möglichkeit geboten werden per Textfeld eine kurze Beschreibung der neu angelegten Schnittstelle zu verfassen.

Diese Anleitung soll dann später in der Import-Maske angezeigt werden.

Dieses Interface muss einen Upload-Button für eine Beispielsdatei bereitstellen.

Anhand dieser gelangt man zu der Maske für die Parameterzuordnung.



### **3.2.3. Konfiguration und Bearbeitung der Parameterzuordnung**

In dieser Ansicht soll eine Tabelle anhand der hochgeladenen Datei angezeigt werden.

Die Attributszeile der CSV Datei soll in der ersten Spalte der Tabelle sichtbar sein.

In der zweiten Spalte muss ein Auswahlfeld für die Auswahl der Parameter erstellt werden, um die Zuordnung zu den einzelnen Spalten der CSV Datei zu ermöglichen.

Um dem Benutzer die Zuordnung zu erleichtern sollen die ersten beiden Zeilen der CSV Datei auch dargestellt werden.

### **3.2.4. Importmaske mit Übersicht aller bisherigen Importe**

Diese Übersicht muss auf einem Blick zeigen, welcher Benutzer zu welchem Zeitpunkt einen Import durchgeführt hat.

Jeder Import muss mit einer fortlaufenden Nummer gekennzeichnet sein um die Fehlersuche zu erleichtern.

Der Benutzer muss auf einem Blick mittels Beschreibung und Farbkennzeichnung sehen können, ob der Import erfolgreich durchgeführt worden ist.

Es muss von dieser Ansicht einen Link geben, der zu der Detailansicht der einzelnen Importe führt.

In der Beschreibung soll auch der Dateiname der importierten Datei stehen.

### **3.2.5. Ansicht einzelner Importe einer Schnittstelle**

Diese Ansicht muss eine Tabelle aller in diesem Import importierter Proben auflisten.

Diese Tabelle muss die Probennummer(n) und die Parameter mit den dazugehörigen Werte der importierten Probe beinhalten.

Ganz unten müssen der Dateiname mit den Rohdaten der hochgeladenen Datei sichtbar sein.

### **3.3. Bisherige Schnittstellen einbinden**

Es sollen alle bisherigen Schnittstellen auf das neue Schnittstellensystem umgestellt werden.

Für jede Schnittstelle soll im Modell eine Funktion angelegt werden.

Die durch die Konfigurationsmaske festgelegten Eigenschaften der neuen Schnittstelle wie Trennzeichen, Attributszeile, Import-Mode und Anleitung sollen an die jeweilige Funktion übergeben werden.

## 4. Systemkonzept

Für die Umsetzung des für den Benutzer sichtbaren Interfaces wurde der Fokus auf ein intuitiv bedienbares GUI (Grafische Benutzeroberfläche) gelegt. Um hier ein einheitliches Design wie in der restlichen Labordatenbank zu haben, wurde das weit verbreitete Bootstrap Framework verwendet.

Die verwendeten Skript- beziehungsweise Auszeichnungssprachen sollen PHP und HTML sein. Diese Programmiersprachen sind durch das CakePHP Framework vorgegeben.

Das CakePHP Framework bot zum Zeitpunkt der Firmengründung der LDB Labordatenbank GmbH im Jahr 2006 die optimale Funktionalität und war damals eines der am weit verbreitetsten PHP Frameworks.

Heute, knapp zehn Jahre später, wurden die verfügbaren PHP Frameworks natürlich laufend weiterentwickelt und verbessert, doch der Aufwand auf ein anderes Framework zu übersiedeln ist mit der Komplexität und der Anzahl der Codezeilen jedoch leider nicht mehr tragbar.

Im Laufe der Zeit sind zahlreiche neue Funktionen in der Labordatenbank hinzugekommen.

Beispiele für derartige Funktionen wären, Formelsystem, Qualitätskennzahlen 6-Sigma-Bereich Cp, CpK, Varianz, Korrelationsdiagramme, Histogramme, 8D-Report um nur einige zu nennen.

Als Datenbank im Backend wird eine MySQL Datenbank verwendet.

Für die in Kapitel 3 beschriebenen Interfaces sollen analog dazu Tabellen in der Datenbank angelegt werden.

Auf die spezielle Namensgebung der Tabellen aufgrund von CakePHP wird im nächsten Kapitel genauer eingegangen.

Die erste Tabelle „importinterfaces“ soll zur Speicherung der Informationen zu den verschiedenen Schnittstellen erstellt werden.

Die darin gespeicherten Informationen können auch für die Schnittstellenübersicht verwendet werden.

Die zweite Tabelle „importmappings“ soll sämtliche Informationen für die Parameterzuordnung beinhalten. Hier soll es Beziehungen zu der Tabelle „importinterfaces“ geben. Außerdem soll so eine Beziehung zur Tabelle „parametertypes“, welche alle Informationen zu den in der Labordatenbank angelegten Parametern enthält, hergestellt werden.

Die Tabelle „imports“ soll alle Informationen zu den bisher importierten Messdaten speichern. Hier gibt es eine Beziehung zur Tabelle „importinterfaces“. Die Informationen dieser Tabelle sollen für die Übersicht aller Imports einer Schnittstelle verwendet werden.

Mit diesen drei Tabellen sind alle Anforderungen erfüllt. Diese Tabellen können später auch erweitert werden, falls weitere Funktionen benötigt werden.

## 5. Softwareentwicklung und Implementierung

### 5.1. Installation der Entwicklungsumgebung

In den nachfolgenden Kapiteln werden kurz alle Module für die Installation der Labordatenbank erläutert.

#### 5.1.1. Auswahl des Servers

Als Server kann entweder ein physischer Server, ein virtualisierter Server oder ein Server von einem Hostingprovider in einem Rechenzentrum verwendet werden. Die Labordatenbank verwendet die Webservices und Serverlandschaft von Amazon in Frankfurt.<sup>15</sup> Ein anderer Anbieter der weltweit virtuelle Server in Rechenzentren zur Verfügung stellt ist zum Beispiel DigitalOcean.<sup>16</sup> Als Mindestanforderungen soll ein x86-64-bit kompatibles System mit mindestens 2 CPU Cores und mindestens 4GB RAM zur Verfügung stehen.

#### 5.1.2. Installation Ubuntu Linux 14.04 LTS

Als Betriebssystem wird eine auf Debian basierende Linux-Distribution verwendet.

Da ein langer Supportzyklus für Sicherheitsupdates sehr wichtig ist, wird hier Ubuntu Linux 14.04 LTS verwendet. Bei dieser Distribution ist der Support für Sicherheitsupdates bis April 2019 gewährleistet.

---

<sup>15</sup> <https://aws.amazon.com/de/>

<sup>16</sup> <https://www.digitalocean.com>

Das Servierbetriebssystem kann unter folgendem Link heruntergeladen werden.<sup>17</sup>

### 5.1.3. Installation LAMP

Für den Betrieb des Webservers, der Skriptsprache und der Datenbank werden Softwarepakete. Apache, MySQL und PHP benötigt. Diese können mittels der verlinkten Installationsanleitung installiert werden<sup>18</sup>. (Eine solche Installation reicht für die Entwicklungsumgebung. Für einen Produktivbetrieb müssen jedoch noch zusätzliche Sicherheitsanforderungen erfüllt werden auf die hier nicht genauer eingegangen werden kann.)

### 5.1.4. Installation CakePHP

Das für den Betrieb der Labordatenbank notwendige quelloffene in PHP geschriebene Web-Framework kann unter folgender URL heruntergeladen werden.<sup>19</sup> Die Installation erfolgt nach Anleitung des Herstellers und wird unter folgendem Link genau beschrieben.<sup>20</sup>

### 5.1.5. Installation Git

Das auf Git basierende Onlineversionsverwaltungssystem Github kann direkt online verwaltet werden.<sup>21</sup> Mit dieser verlinkten Installationsanleitung kann Git

---

17 <http://www.ubuntu.com/download/server>

18 <https://wiki.ubuntuusers.de/LAMP/>

19 <http://cakephp.org>

20 <http://book.cakephp.org/1.3/en/The-Manual/Developing-with-CakePHP/Installation.html>

21 <https://github.com>

auf dem Ubuntu Linuxsystem installiert werden.<sup>22</sup> Zur Identifikation des Servers gegenüber des Github Servers muss ein SSH key auf dem Ubuntu Server generiert und bei GitHub hinterlegt werden.<sup>23</sup> Jetzt kann der aktuelle Softwarestand der Labordatenbank bei Github heruntergeladen werden.<sup>24</sup>

### 5.1.6. Installation Bootstrap

Bootstrap ist ein in CSS geschriebenes Framework für HTML und JS, welches zur Frontend Entwicklung der Labordatenbank verwendet wird. Es enthält Vorlagen für Typografie, Formulare, Buttons, Links, Tabellen und ein Rastersystem für die Gestaltung der Webseiten. Damit sich der Benutzer in den verschiedenen Menüs der Labordatenbank besser zurechtfindet wird hier auf ein einheitliches GUI gesetzt. Diese dazu notwendige Erweiterung, um Bootstrap auf CakePHP lauffähig zu machen, kann unter folgender Adresse heruntergeladen werden<sup>25</sup>.

### 5.2. Einrichten der CakePHP Ordnerstruktur

Für das Framework müssen einige Dateien erstellt werden. Auf dieser Website wird sehr anschaulich anhand der Programmierung eines Internet Blogs erklärt, wie das Ordnersystem von CakePHP und dessen Komponenten zusammenspielen<sup>26</sup>.

---

22 <https://wiki.ubuntuusers.de/Git/>

23 <https://help.github.com/articles/generating-an-ssh-key/>

24 <https://help.github.com/articles/fork-a-repo/#platform-linux>

25 <https://github.com/slywalker/TwitterBootstrap>

26 <http://book.cakephp.org/1.2/en/The-Manual/Tutorials-Examples.html>

Zusätzlich müssen für das Erstellen einer Webseite, den Zugriff auf die Datenbank und die Funktionalität der Messgeräte-Datenimport-Schnittstelle einige Dateien angelegt werden.

Durch die in Kapitel 4 ausgearbeiteten Datenstruktur ergeben sich folgende Dateien die in Anhang A dargestellt werden.



### 5.3. Erstellen der Datenbanktabellen

Die Zuordnung von Datenbanktabellen zu Models erfolgt in CakePHP über die Namensgebung im Singular und Plural.

Model (Singular): „importinterface“

Controller (Plural plus controller): „importinterfaces\_controller“

Datenbanktabelle (Plural): „importinterfaces“

Die Fremdschlüssel in den Datenbanktabellen werden automatisch erkannt.

Datenbankattribut (Singular plus \_id): „importinterface\_id“

Die Datenbankstruktur für die drei neuen Tabellen sieht dabei wie folgt aus:

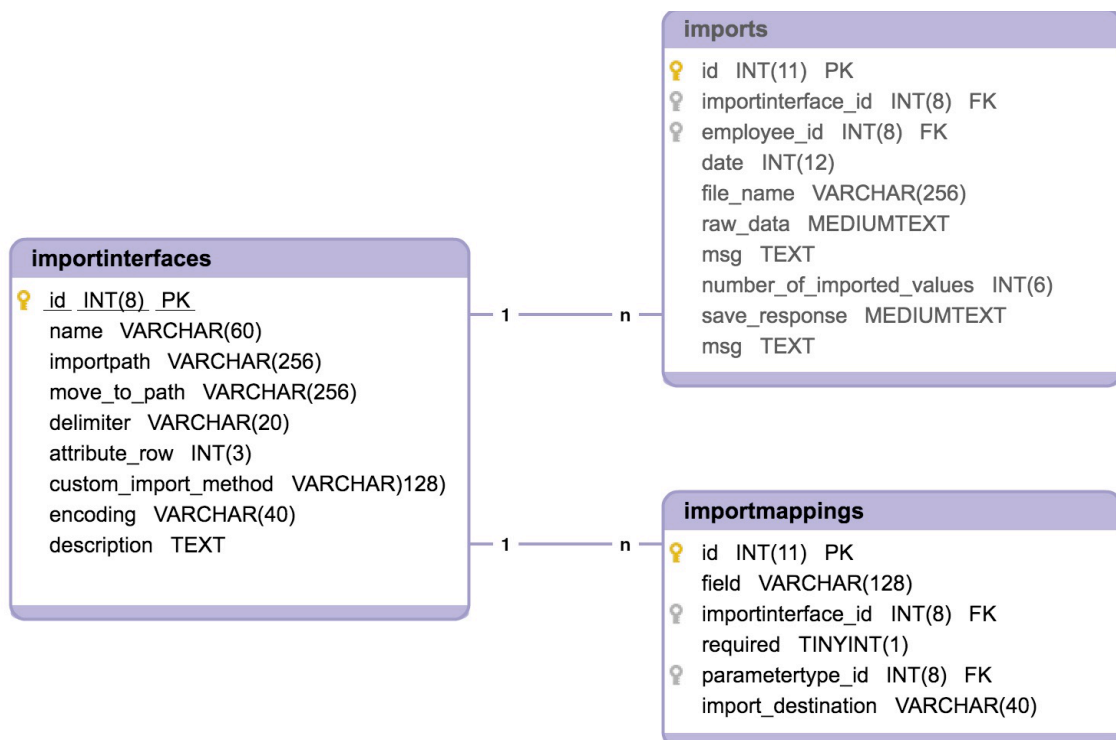


Abbildung 8: Datenbankstruktur der drei neuen Tabellen (eigene Darstellung)

Während der Testphase gab es ein Problem mit Sonderzeichen und auch einige Kundenwünsche, wodurch die Tabellen um einige Funktionalitäten erweitert wurden.

Beim Import gab es Darstellungsprobleme mit Sonderzeichen, da die CSV Dateien mit verschiedenen Zeichensätzen hochgeladen wurden. Es wurde die Schnittstelle um alle von Kunden verwendeten Zeichensätzen erweitert.

#### importinterfaces:

Es wurde beim Erstellen der Schnittstelle ein neues Auswahlfeld mit den gängigsten Zeichensätzen hinzugefügt. Diese sind UTF8, UTF16, UTF16LE. Als Defaultwert wird Latin-1 verwendet welches dem ISO/IEC 8859-1 entspricht und auch der Standardzeichensatz der Labordatenbank ist.

Drei Kunden wollten die Möglichkeit haben, Messdaten automatisch von einem Netzlaufwerk einzulesen, zu importieren und die CSV Datei dann auf ein anderes Laufwerk zu verschieben.

Um diese Funktionalität für den Kunden transparenter umzusetzen wurden Quell- und Ziellaufwerkspfade als Textfeldeingaben in die „importinterfaces“ Tabelle aufgenommen.

Derzeit ist noch ein Bash-Skript, welches zu fix eingestellten Abständen mittels eines Cron-Eintrags aufgerufen wird, notwendig. Nähere Informationen zu Crontab und zum Bash-Skript sind im Anhang B zu finden.

Attributname	Beschreibung
id	Primärschlüssel (zählt automatisch hoch)
name	Bezeichnung der Schnittstelle
importpath	Absoluter Pfad des Netzlaufwerks mit den CSV Messdaten
move_to_path	Absoluter Pfad des Netzlaufwerks wohin die CSV Messdaten verschoben werden sollen
delimiter	Trennzeichen der CSV Datei
attribute_row	Position der Attributzeile in der CSV Datei
custom_import_method	Funktionsname der alten Schnittstellen
encoding	Auswahl des Zeichensatzes

Attributname	Beschreibung
mode	Importmodus für CSV Datei
description	Beschreibung der Schnittstelle

Tabelle 2: importinterfaces Datenbanktabelle (eigene Darstellung)

importmappings:

Da bei den Messungen nicht immer alle beim Erstellen der Schnittstelle definierten Parameter gemessen werden, ergab sich die Notwendigkeit gewisse Parameter als Pflichtfelder und die restlichen als optional zu definieren.

Es wurde daher bei der Tabelle „importmappings“ noch ein weiteres Kontrollkästchen hinzugefügt um diese Eigenschaft festlegen zu können..

Attributname	Beschreibung
id	Primärschlüssel (zählt automatisch hoch)
field	Spaltenname der CSV Datei
importinterface_id	id der Tabelle importinterfaces
required	Info ob dieser Parameter notwendig und vorhanden sein muss
parametertype_id	parametertype_id wohin die Probe gespeichert werden soll
import_destination	Info wohin der CSV Wert importiert werden soll

Tabelle 3: importmappings Datenbanktabelle (eigene Darstellung)

imports:

Attributname	Beschreibung
id	Primärschlüssel (zählt automatisch hoch)
importinterface_id	id der Tabelle importinterfaces
employee_id	id des Mitarbeiters der die Mitarbeiter der die CSV Datei importiert
date	Zeitpunkt des Imports im Unixtime Format
file_name	Dateiname der CSV Datei
raw_data	Rohdaten der CSV Datei
msg	Importmeldung nach dem Import
number_of_imported_values	Anzahl der Importierten Werte

Attributname	Beschreibung
save_response	Tabelle der Importierten Werte

Tabelle 4: Imports Datenbanktabelle (eigene Darstellung)

Der SQL Quellcode für die Erstellung der Tabellen ist in Anhang C zu finden.

## 6. Ergebnisse und Ausblick

### 6.1. Ergebnisse

Das Ziel der Diplomarbeit war die Datenbankentwicklung und Implementierung der Messgeräte-Datenimport Schnittstelle.



Importschnittstellen		
Zwickmaschine Gewebe	144	 bearbeiten
Zwickmaschine Bändchen	575	 bearbeiten
MC-Protokoll	56	 bearbeiten

+ Importschnittstellen anlegen

Abbildung 9: Schnittstellenübersicht (eigene Darstellung)

Als Ergebnis liegt eine fertig komplett in das CakePHP Framework implementierte Messgeräte-Datenimport Schnittstelle vor.

Neben der eigentlichen Schnittstelle für den Import von Messgerätedateien können verschiedene Schnittstellen verwaltet werden. Des Weiteren können Parameterzuordnungen im Nachhinein angepasst werden.

Es wurde die Zielsetzung von zehn Schnittstellen, die in die neue Messgeräte-Datenimport Schnittstelle übertragen werden sollen, übertroffen und sogar insgesamt 16 bestehende Schnittstellen migriert.

Diese Schnittstellen sind bei den Kunden aktiv im Einsatz und werden täglich verwendet.

Sollte bei einer Schnittstelle ein Fehler, zum Beispiel durch eine Änderung der zu Importierenden CSV Datei auftreten, gelingt die Fehlersuche um einiges Schneller.

## Zwickmaschine Gewebe Importschnittstelle

CSV-Datei auswählen

no file selected

oder [abbrechen](#)

### Die letzten Imports

Nr	Datum		Beschreibung	
779	01.06.2016 13:15	FUS	A003_coated_CST_bh2.ERG 6 Werte bei 1 Proben importiert	<a href="#">Import anzeigen</a>
778	01.06.2016 13:15	FUS	A003_coated_CST_bh1.ERG 6 Werte bei 1 Proben importiert	<a href="#">Import anzeigen</a>
777	01.06.2016 10:13	FUS	A003_coated_CST_bh2.ERG 0 Werte bei 0 Proben importiert Importschlüssel <a href="#">bh2Clock time</a> nicht gefunden!	<a href="#">Import anzeigen</a>

Abbildung 10: Übersicht importierter Messdaten (eigene Darstellung)

## Zwickmaschine Gewebe-Import Nr. 779 01.06.2016 13:15 von Venuse Fuchs

Importschnittstelle: Zwickmaschine Gewebe

6 Werte bei 1 Proben importiert

Probe	Colour	Load Cell	Testing speed [mm/min]	Clamping length [mm]
<a href="#">C20160601-472</a>	white	500	100	200

Abbildung 11: Zusammenfassung eines einzelnen Imports (eigene Darstellung)

Durch die jetzige Historie aller bisher Importierten Daten lässt sich der Zeitpunkt der veränderten CSV Datei sehr gut rekonstruieren.

Das Problem mit den nicht darstellbaren Sonderzeichen, verursacht durch einen anderen Zeichensatz, konnte sehr schnell durch das Auswahlfeld „Zeichensatz“ gelöst werden. Ebenso wurde der Kundenwunsch mit dem automatischen Import von einem Netzlaufwerk umgesetzt.

## 6.2. Ausblick

Um die mit dieser Schnittstelle geschaffenen Funktionen für andere Anwendungsfälle noch weiter auszubauen, müssen noch einige Erweiterungen hinzukommen.

Eine Funktion die von einigen Kunden benötigt wird ist, nicht nur wie in dieser Diplomarbeit beschrieben, das Importieren von Testdaten in bestehende Proben, sondern auch das Anlegen der Proben selbst.

Eine weitere Funktion die mit einer weiteren, neuen Implementierung einhergeht, ist die Möglichkeit Dateien zu Proben hinzuzufügen. Einige Messgeräte erzeugen nicht nur die Messwerte sondern auch Grafiken zu den einzelnen Messergebnissen. Diese werden meist zusätzlich zu den Messdaten als PDF oder JPG-Datei zu den Messdaten mit dazu abgespeichert.

Bisher müssen diese mit einem Upload-Knopf in jeder Probe einzeln hochgeladen werden.

Im Ausleselaufwerk befinden sich aktuell verschiedene Dateien.

In Zukunft soll der Benutzer die Möglichkeit haben die Dateiendungen anzugeben, die im Ausleselaufwerk mit überwacht werden sollen. Es werden dann automatisch die Dateien mit der richtigen Dateiendung entsprechend zu den einzelnen Proben hinzugefügt und anschließend in das angegebene Verschiebelaufwerk verschoben.



Weiters soll noch auswählbar sein, wie oft das Ausleselaufwerk nach neuen Dateien durchsucht werden soll.

Im Hintergrund soll das Laufwerk zum Beispiel alle 15 Minuten durchsucht werden. Sobald neue Dateien in den Ordner kopiert werden, sollen diese mittels der jeweils ausgewählten Schnittstelle importiert oder hochgeladen werden.

Dies könnte durch neue Datenbankattribute und eine automatisierte Erstellung des Bash-Scripts und des Cron-Eintrags realisiert werden.



## Literaturverzeichnis

Appelrath H.-J., H. Kagermann, und H. Krcmar, 2014, Future Business Clouds, Utz Verlag GmbH 2.2.1: 38.

Golding David, 2008, Beginning CakePHP: From Novice to Professional, Apress, 1, 4-5

Kargl Jürgen, 2009, PHP-MVC-Frameworks: Evaluierung von Web Application Frameworks für die serverseitige Schriftsprache PHP, GRINN Verlag 3.2.1: 14(1)

Öhlinger Patrick, 2014, Labordatenbank Anleitung, WWW Dokument, URL <http://labordatenbank.at/manualpages/view/44>, abgerufen am 22.Mai 2016)

Steiner René, 2014, Grundkurs Relationale Datenbanken, Springer Vieweg 2.1: 5

Steiner René, 2014, Grundkurs Relationale Datenbanken, Springer Vieweg 3.1.1: 16

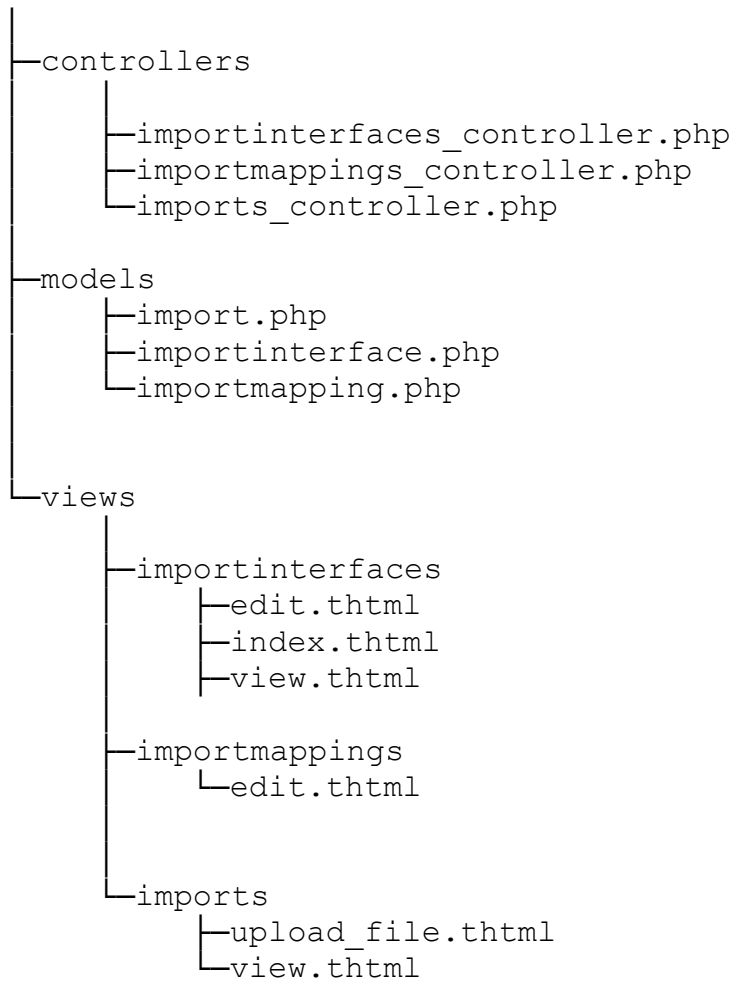
Steiner René, 2014, Grundkurs Relationale Datenbanken, Springer Vieweg 3.4.8: 96



# Anhang

## Anhang A CakePHP Ordnerstruktur

Kundenordner



## Anhang B Crontab und Bash-Skript

Aufbau eines Crontab Eintrages:

```

|-----| Minuten (0 - 59)
|-----| Stunden (0 - 23)
|-----| Tag des Monats (1 - 31)
|-----| Monat (1 - 12)
|-----| Tag der Woche (0 - 6) 0=So, 1=Mo, usw.
* * * * * Befehl oder Skript welches aufgerufen werden
soll

```

Bei diesem Eintrag wird das `watch_import_folder.sh` Skript alle 5 Minuten aufgerufen:

```
* /5 * * * * /home/lims/watch_import_folder.sh
```

Beispiel Bash-Skript zur Überwachung des Laufwerkfolders:

Sobald sich eine oder mehr `.csv` Dateien auf dem `Import_Laufwerk` befinden wird die Funktion `csv_import` der Labordatenbank gestartet.

```
#!/bin/sh
cd /home/lims/mnt/Import_Laufwerk

CSV_PDF_COUNT=$(ls -l | grep -E ".csv" | wc -l)

if [ $#CSV_PDF_COUNT != 0 ]
then curl
„http://localhost/cake/kunde/orders/csv_import“
fi

```

## Anhang C Erstellung der Datenbanktabellen in SQL

```
CREATE TABLE `importinterfaces` (  
    `id` int(8) NOT NULL AUTO_INCREMENT,  
    `name` varchar(60) DEFAULT NULL,  
    `importpath` varchar(256) DEFAULT NULL,  
    `move_to_path` varchar(256) DEFAULT NULL,  
    `delimiter` varchar(20) DEFAULT NULL,  
    `attribute_row` int(3) DEFAULT '0',  
    `custom_import_method` varchar(128) DEFAULT NULL,  
    `encoding` varchar(40) DEFAULT NULL,  
    `mode` varchar(40) DEFAULT NULL,  
    `description` text,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

```
CREATE TABLE `importmappings` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `field` varchar(128) DEFAULT NULL,  
    `importinterface_id` int(8) DEFAULT NULL,  
    `required` tinyint(1) DEFAULT '0',  
    `parametertype_id` int(8) DEFAULT NULL,  
    `import_destination` varchar(40) DEFAULT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

```
CREATE TABLE `imports` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `importinterface_id` int(8) DEFAULT NULL,  
    `employee_id` int(8) DEFAULT NULL,  
    `date` int(12) DEFAULT NULL,  
    `file_name` varchar(256) DEFAULT NULL,  
    `raw_data` mediumtext,  
    `msg` text,  
    `number_of_imported_values` int(6) DEFAULT '0',  
    `save_response` mediumtext,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```





## **Selbstständigkeitserklärung**

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Wien, den 6. Juni 2016

Stefan Raschbacher