

# Bachelorarbeit

zur Erlangung des Abschlusses als  
Bachelor of Science

an der

Berufsakademie Sachsen  
Staatliche Studienakademie Riesa  
Studiengang Labor- und Verfahrenstechnik

Kurs: 6BT14-1

Studienrichtung: Biotechnologie

## **Thema:**

Entwicklung eines webbasierten Chemikalienmanagementsystems zur  
Dokumentation der Bestandsmengen mittels Echtzeiterfassung der Auswaagen

## **Eingereicht von:**

Martin Schneider

## **Firma:**

QuoData GmbH  
Prellerstraße 14  
01187 Dresden

## **Prüfer und Gutachter:**

Herr Dipl.-Phys. Christian Bläul (QuoData GmbH)

Herr Prof. Dr.-Ing. Andreas Zehnsdorf (Helmholtz-Zentrum für Umweltforschung)

## Themenstellung zur Bachelorarbeit

von: Herrn Martin Schneider

Firma: QuoData GmbH Quality & Statistics

Thema: **„Entwicklung eines webbasierten Chemikalienmanagement-systems zur Dokumentation der Bestandsmengen mittels Echtzeiterfassung der Auswaagen“**

1. Gutachter: Dipl.-Phys. Christian Bläul  
QuoData GmbH Quality & Statistics

2. Gutachter: Prof. Dr.-Ing. Andreas Zehnsdorf  
Helmholtz-Zentrum für Umweltforschung

Das Thema wurde vom Leiter des Studienganges Labor- und Verfahrenstechnik der Staatlichen Studienakademie Riesa am 21.04.2017 bestätigt.

Die Bearbeitungszeit der Bachelorarbeit beginnt am 24.04.2017. Abgabetermin ist der 24.07.2017 bis 16:00 Uhr bzw. Posteingang.

Riesa, 16.05.2017



Prof. Dr. Gläser  
Leiter des Studienganges

## **Abstrakt**

Ziel dieser Arbeit war die Entwicklung eines verteilten Chemikalienmanagementsystems für die Dokumentation der firmeninternen Chemikalienbestände. Dabei sollte durch eine Verbindung mit einer Laborwaage eine Echtzeiterfassung der Auswaagen realisiert werden. Es wurde eine MySQL-Datenbank auf einem internen Server angelegt und ein Webinterface für den Zugriff bereitgestellt.

Über das Webinterface können Nutzer grundlegende Arbeitsschritte für die Chemikalienverwaltung durchführen, Administratoren können auch direkt die Datenbank verändern. Für die Verbindung zu den Waagen wurde eine lokale Anwendung geschrieben, die über eine serielle Verbindung mit einer Waage kommunizieren kann. Über ein Drittanbieter-Tool kann diese Anwendung mit der Datenbank kommunizieren und die eingewogenen Chemikalien in Echtzeit eintragen.

Schneider, Martin: Entwicklung eines webbasierten Chemikalienmanagementsystems zur Dokumentation der Bestandsmengen mittels Echtzeiterfassung der Auswaagen

Seitenzahl: 47

Anzahl der Abbildungen: 20

# Thesen

- Zur Erfassung der Chemikaliendaten eignet sich ein relationales Datenmodell.
- Die grundlegenden Arbeitsschritte der Chemikalienverwaltung können in einem Webinterface benutzerfreundlich bereitgestellt werden.
- Ein datenbankbasiertes Chemikalienmanagementsystem kann als Grundlage für Inventuren genutzt werden.
- Der Umstieg von einem Excel-Verzeichnis zu einer webbasierten Datenbank kann durch geeignete Import-Funktionen erheblich erleichtert werden.
- Durch die Echtzeiterfassung von Einwaagen können Restbestände von Chemikalien ermittelt und überwacht werden.
- Die Echtzeiterfassung von Einwaagen kann durch eine dedizierte Anwendung auf einem Laborrechner halbautomatisch durchgeführt werden.
- Die Rechteverwaltung des Chemikalienmanagementsystems kann in die firmeninterne Benutzerverwaltung integriert werden.
- Regelmäßige Sicherheitskopien der Datenbank gewährleisten eine angemessene Datensicherheit.

# Inhaltsverzeichnis

|   |     |
|---|-----|
| Inhaltsverzeichnis .....                              | I   |
| Abbildungsverzeichnis .....                           | III |
| Tabellenverzeichnis .....                             | IV  |
| Abkürzungsverzeichnis .....                           | V   |
| 1 Einleitung .....                                    | 1   |
| 2. Theoretische Grundlagen .....                      | 3   |
| 2.1 Serielle Kommunikation .....                      | 3   |
| 2.2 RS-232 .....                                      | 7   |
| 2.3 Datenbanken .....                                 | 9   |
| 2.4 Softwarestacks .....                              | 14  |
| 2.4.1 Webservices .....                               | 14  |
| 2.4.2 C# .....  | 16  |
| 3 Konzeption .....                                    | 18  |
| 3.1 Serverseitig .....                                | 18  |
| 3.1.1 Datenbank- und Speicherstruktur .....           | 18  |
| 3.1.2 Webinterface .....                              | 21  |
| 3.1.3 Zugangsbeschränkung und Rechteverwaltung .....  | 22  |
| 3.2 Clientseitig .....                                | 23  |
| 3.2.1 Benutzeroberfläche .....                        | 23  |
| 3.2.2 Serielle Schnittstelle zur Waage .....          | 24  |
| 3.2.3 Datenbankschnittstelle .....                    | 24  |
| 4 Geräte und Software .....                           | 26  |
| 5 Implementation .....                                | 27  |
| 5.1 Serverseitig .....                                | 27  |
| 5.1.1 Datenbank- und Speicherstruktur .....           | 27  |
| 5.1.2 Webinterface .....                              | 29  |
| 5.1.3 Rezepte definieren und verwalten .....          | 34  |
| 5.1.4 Administrativer Bereich und Erweiterungen ..... | 35  |
| 5.2 Clientseitig .....                                | 37  |

---

|                                    |    |
|------------------------------------|----|
| 5.2.1 Benutzeroberfläche .....     | 37 |
| 5.2.2 Serielle Schnittstelle ..... | 40 |
| 5.2.3 Datenbankverbindung .....    | 41 |
| 6 Produktionsumgebung.....         | 43 |
| 6.1 Serverseitig .....             | 43 |
| 6.1.1 Einsatz.....                 | 43 |
| 6.1.2 Ausblick .....               | 44 |
| 6.2 Clientseitig .....             | 44 |
| 6.2.1 Einsatz.....                 | 44 |
| 6.2.2 Ausblick .....               | 45 |
| Zusammenfassung .....              | 47 |
| Literaturverzeichnis .....         | 49 |

## Abbildungsverzeichnis

|  |    |
|--|----|
| Abbildung 1: Spannungsverhalten von bipolar NRZ-codierten binären Werten.....  | 4  |
| Abbildung 2: Prinzip der asymmetrischen (a) und symmetrischen (b) Verbindung als Anwendung in der Tontechnik [ <a href="http://hbernstaedt.de/Test/Asm_Asm.jpg">http://hbernstaedt.de/Test/Asm_Asm.jpg</a> ] .....   | 6  |
| Abbildung 3: Pinbelegung des 9-poligen D-Sub-Steckers [ <a href="https://www.db9-pinout.com/db9-pinout/db9-pinout.gif">https://www.db9-pinout.com/db9-pinout/db9-pinout.gif</a> ] .....  | 8  |
| Abbildung 4: Rahmenbildung nach RS-232 mit 8 Datenbits und einem Stoppbit [Frenzel Handbook] .....   | 9  |
| Abbildung 5: Beispielhafter Aufbau einer Tabelle als Teil einer relationalen Datenbank .....   | 10 |
| Abbildung 6: Verknüpfung von zwei Tabellen über einen foreign key .....  | 11 |
| Abbildung 7: Aufbau des LAMP-Pakets ( <a href="https://i2.wp.com/blog.novatrend.ch/wp-content/uploads/2014/11/LAMP.png">https://i2.wp.com/blog.novatrend.ch/wp-content/uploads/2014/11/LAMP.png</a> ) .....  | 15 |
| Abbildung 8: Dynamische Erstellung von Inhalten mit PHP ( <a href="https://de.wikipedia.org/wiki/LAMP_(Softwarepaket)#/media/File:PHP_funktionsweise.svg">https://de.wikipedia.org/wiki/LAMP_(Softwarepaket)#/media/File:PHP_funktionsweise.svg</a> )..... | 16 |
| Abbildung 9: Konzeptionelle Datenbankstruktur mit Attributen und zugehörigen Datentypen .....  | 19 |
| Abbildung 10: Implementierte Datenbankstruktur des Chemikalienverzeichnisses mit Attributen und zugehörigen Datentypen .....   | 27 |
| Abbildung 11: Ausschnitt der Übersichtsseite für die gefilterte Anzeige von Chemikalienlieferungen .....   | 29 |
| Abbildung 12: Detailansicht einer beispielhaften Lieferung .....   | 31 |
| Abbildung 13: Struktur der Oberfläche zum Einfügen und Bearbeiten von Lieferungen .....  | 31 |
| Abbildung 14: Struktur der Oberfläche zum Eintragen neuer Chemikalien.....   | 33 |
| Abbildung 15: Nutzeroberfläche für die Verwaltung von Rezepten im Webinterface   | 34 |
| Abbildung 16: Formular für das Eintragen eines Rezeptes mit 5 Komponenten, teilweise gefüllt mit Beispielwerten .....  | 35 |
| Abbildung 17: Initiale Oberfläche der clientseitigen Software .....  | 38 |
| Abbildung 18: Ausschnitt der graphischen Nutzeroberfläche nach Auswahl eines Rezeptes .....  | 39 |

|  |    |
|--|----|
| Abbildung 19: Oberfläche zur Konfiguration der seriellen Schnittstelle in der clientseitigen Software..... | 40 |
| Abbildung 20: Oberfläche zur Konfiguration der Datenbankverbindung in der clientseitigen Software.....     | 41 |

## **Tabellenverzeichnis**

|  |    |
|--|----|
| Tabelle 1: Liste der verwendeten Entwicklungsumgebungen .....                  | 26 |
| Tabelle 2: Liste der verwendeten Softwarepakete und Programmierwerkzeuge ..... | 26 |



## Abkürzungsverzeichnis

|      |   |
|------|---|
| CAS  | Chemical Abstract Service                                   |
| CIL  | Common Intermediate Language                                |
| COM  | Communication Equipment oder Component Object Model         |
| CTS  | Clear To Send   |
| DB   | Datenbank   |
| DBMS | Datenbank-Managementsystem                                  |
| DCE  | Data Circuit-Terminating Equipment                          |
| DIN  | Deutsche Industrienorm                                      |
| DSR  | Dataset Ready   |
| DTE  | Data Terminal Equipment                                     |
| DTR  | Data Terminal Ready   |
| EIA  | Electronic Industries Alliance                              |
| GHS  | Global harmonisiertes System zur Einstufung von Chemikalien |
| HTML | Hypertext Markup Language                                   |
| LAMP | Akronym: Linux, Apache, MySQL, PHP                          |
| LDAP | Lightweight Directory Access Protocol                       |
| NF   | Normalform  |
| NRZ  | No-Return-To-Zero   |
| PC   | Personal Computer   |
| PDF  | Portable Document Format                                    |
| PHP  | Hypertext Preprocessor                                      |
| RFID | Radio-Frequency Identification                              |
| RI   | Ring Indicator  |
| RS   | Recommended Standard  |
| RTS  | Request To Send   |
| RZ   | Return-To-Zero  |
| SDB  | Sicherheitsdatenblatt                                       |
| SQL  | Structured Query Language                                   |
| URL  | Uniform Resource Locator                                    |

# 1 Einleitung

In Laboren, wie in Betrieben, hängt die Fähigkeit, Leistung zu erbringen, in erster Linie von dem Vorhandensein der nötigen Materialien ab. Immer wieder müssen Versuche verschoben oder unterbrochen werden, weil Chemikalien nicht in ausreichender Menge oder überhaupt nicht vorliegen. Dennoch wird der Bestand der Chemikalien in vielen Labors noch immer mit unübersichtlichen Excel-Tabellen oder analog mit Bestandslisten erfasst. Eine Erfassung des Chemikalienverbrauchs und des derzeitigen Bestandes einer bestimmten Chemikalie beschränkt sich dabei meist darauf, entleerte Behälter aus der Liste zu entfernen. Dementsprechend ist es nicht möglich, den exakten Bestand einer Chemikalie mit diesen Methoden zu erfassen oder Einträge nach mehreren Kriterien filtern zu lassen. [WILD, 2002]

Als Alternative zu den herkömmlichen Excel-Tabellen und Chemikalienlisten wurden deshalb datenbankbasierte Managementsysteme entwickelt, in denen alle Chemikalien erfasst, bearbeitet und verwaltet werden können. Die Systeme unterscheiden sich dabei stark in ihrem Funktionsumfang. Ein einfaches, kostenfreies System ist beispielsweise das webbasierte Quartzzy, das eine einfache Verwaltung von Chemikalien und eine zentrale Verwaltung von Bestellanträgen bietet [Quartzzy, 2015]. Andere Systeme, wie das Grainger Keepstock Inventory Management, bieten direkte Anbindungen zu Anbietern von Chemikalien, um Chemikalien automatisch nachzubestellen. Software für große Labore bieten teilweise Möglichkeiten, den Lagerort der Chemikalien live zu verfolgen, genannt Live-Tracking. Ein Beispiel dafür ist der Syngo Lab Inventory Manager von Siemens, bei dem alle Chemikalien mit einem RFID-Chip versehen werden und somit der Lagerungsort permanent erfasst werden kann. [SIEMENS, 2017]

Mithilfe dieser kommerziellen Systeme ist es möglich, den Chemikalienbestand zu erfassen und manuell den Verbrauch zu vermerken, um einen Überblick über die verbleibenden Stoffmengen zu erhalten. Allerdings erfordert dies die manuelle Übertragung von Messwerten aus Waagen in die Software, wodurch diese Systeme fehleranfällig und unhandlich werden. Dieser Arbeitsschritt könnte durch die Verwendung spezialisierter Software, die auf den Datenausgang der Waagen zugreifen kann, umgangen werden. Nahezu alle aktuellen Laborwaagen, beispielsweise der Marke Kern, sind mit Anschlüssen für die serielle

Datenübertragung ausgestattet, über die die Waage gesteuert und Messwerte ausgegeben werden können [KERN, 2016]. Allerdings unterscheiden sich die Modelle in den akzeptierten Befehlen und dem Format der Datenausgabe, weshalb eine allgemeine Software für alle gängigen Modelle schwer zu realisieren ist [KERN, 2017; KERN, 2010].

Deshalb sollte im Zuge dieser Arbeit eine Software geschrieben werden, die das gängige System der datenbankbasierten Chemikalienverwaltung über ein Kopplungsmodul mit den vorhandenen Laborwaagen verbinden und damit eine Dokumentation aller eingewogenen Chemikalien ermöglichen kann. Die Datenbank sollte dabei auf einem firmeninternen Server abgelegt werden und über ein Webinterface erreichbar gemacht werden. Die Anforderungen an das Interface waren:

- Intuitive, benutzerfreundliche Benutzeroberfläche
- Passwortschutz
- Bereitstellung von grundlegenden Verwaltungsfunktionen:
  - Chemikalienlieferungen eintragen
  - Lieferungen bearbeiten
  - Lieferungen filtern und anzeigen

Das Modul für die Kommunikation mit den Waagen sollte direkt auf die Datenbank zugreifen können und volle Rechte für die Datenmanipulation besitzen. Die Benutzeroberfläche sollte einfach gestaltet werden, um eine schnelle Bedienung zu ermöglichen.

Als Datenbank sollte eine mindestens in der ersten Normalform vorliegende relationale Datenbank mit dem Datenbankmanagementsystem MySQL verwendet werden. Die Datenbank sollte auf Grundlage eines vorliegenden Chemikalienverzeichnisses, einer Excel-Datei, entwickelt und strukturiert werden.

## 2. Theoretische Grundlagen

### 2.1 Serielle Kommunikation

Für den modernen Betrieb von industriellen Geräten ist die Kommunikation zwischen Gerät und einer Steuereinheit, meist ein Rechner, oft unumgänglich. Es werden dabei Messwerte oder Statusinformationen des Gerätes an den Rechner und Befehle von dem Rechner auf das Gerät übertragen. Diese Daten bestehen aus einem Binärcode, also der Verkettung von binären Signalen mit definierter Struktur. Der Binärcode repräsentiert dabei häufig Zeichen oder ganze Zeichenketten, die übertragen werden sollen. [HULZEBOSCH, 2008]

Die einzelnen binären Signale werden in der Informatik als Bits bezeichnet. Diese Bits können die logischen Zustände 0 oder 1, bzw. *high* oder *low*, annehmen. Jeweils 8 Bits werden zu einer geschlossenen Informationseinheit, dem Byte, zusammengefasst. Dadurch ergeben sich für jedes Byte 256 mögliche Zustände zwischen 0 und 255. Nach dem 8-Bit-Code der DIN 66303 kann jedes Byte ein Schriftzeichen, beispielsweise Buchstaben oder Zahlen, aber auch Sonderzeichen, repräsentieren. [DIN66303:2000-06; HULZEBOSCH, 2008]

Die Bytes können entweder seriell oder parallel übertragen werden. Bei der parallelen Übertragung werden alle Bits eines Bytes gleichzeitig übertragen. Voraussetzung für die parallele Übertragung ist eine Verbindung mit mindestens einem Datenleiter pro übertragenem Bit und weiteren Kabeladern für Steuersignale. Weiterhin müssen die Kabel mit zunehmender Übertragungsgeschwindigkeit immer höheren Qualitätsanforderungen entsprechen, um die Datenintegrität sicherzustellen. So müssen beispielsweise die Impedanzen aller Kabel übereinstimmen, um Störungen zu vermeiden, und Längenunterschiede in den Verbindungen können die zeitliche Abstimmung der Signale stören. Weiterhin kommt es durch die langen, parallel liegenden Leitungen und schnell wechselnden Spannungen zu Induktionserscheinungen, die als „*crosstalk*“ bezeichnet werden. Dieser Effekt senkt durch wechselhafte Störungen die Übertragungsqualität. Gerade bei hohen Übertragungsraten, im Bereich von Gb/s, ist die maximale Übertragungsweite deshalb auf wenige Zentimeter begrenzt. Durch die begrenzte, beim Entwurf der Verbindung festgelegte, Anzahl von Kabeladern ist auch die Übertragung größerer Datenpakete, zum Beispiel durch zusätzliche Paritätsbits oder Datenpaketen mit

mehr als 8 Bits, nachträglich nicht mehr möglich. Die parallele Kommunikation wird deshalb in modernen Systemen vorrangig für die Übertragung von Daten innerhalb eines Rechners oder innerhalb einer Leiterplatte verwendet. [FRENZEL, 2016]

Die zweite Möglichkeit zur Datenübertragung ist die serielle Kommunikation. Dabei werden die Bits eines Bytes sequentiell über ein einzelnes Kabel versendet. Für die einfachste serielle Kommunikation ist dementsprechend nur ein einzelnes Datenkabel und ein Kabel für die gemeinsame Masse, als Referenz, nötig. Der technische Aufwand für die Hardware ist also deutlich geringer als für die parallele Übertragung. Andererseits ist die Übertragungsgeschwindigkeit geringer, da die Bits nacheinander gesendet werden. Für normale Anwendungen ist dieser Verlust an Übertragungsgeschwindigkeit kaum relevant, da trotzdem Übertragungsraten von mehreren Gb/s erreicht werden können. Es können auch hybride Methoden verwendet werden, bei denen mehrere serielle Verbindungen parallel Teile der Daten übertragen. [FRENZEL, 2016]

Bei der Übertragung der Bits müssen die logischen Werte, 1 oder 0, durch Spannungslevel dargestellt werden. Die verwendeten Verfahren werden Leitungscodes genannt. Ein einfaches Verfahren ist das bipolare Non-Return-to-Zero (NRZ), bei dem eine 1 durch einen positiven Pegel und eine 0 durch einen negativen Pegel ausgedrückt wird. Der Zustand wird für die gesamte Dauer des Bits gehalten und kehrt nicht auf 0 zurück. Das Spannungsverhalten, abhängig von den zu übertragenden logischen Werten, ist in Abbildung 1 dargestellt. Die verwendeten Spannungslevels sind in den jeweiligen Standards der seriellen Schnittstellen festgelegt. [FRENZEL, 2016]

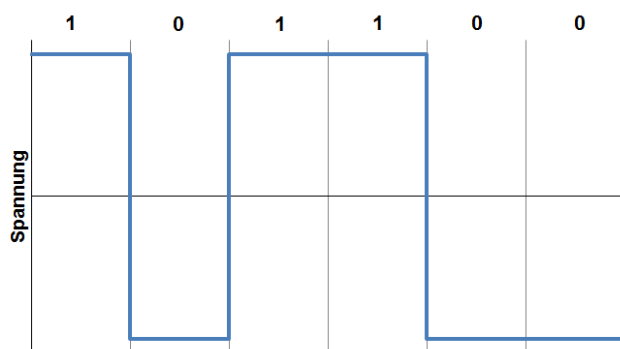


Abbildung 1: Spannungsverhalten von bipolar NRZ-codierten binären Werten

Ein allgemeiner Vorteil der bipolaren Codierungen ist, dass sich eine geringere mittlere Gleichspannung in der Leitung aufbaut als bei unipolaren Codierungen. Diese Störspannung entsteht durch den schnellen Wechsel mehrerer Signallevel der gleichen Polarität und würde die Übertragungsqualität senken und damit die Übertragungsrate limitieren. Ein Nachteil der NRZ-Methode ist, dass bei der Übertragung langer Folgen des gleichen Signals die einzelnen Signale nicht klar voneinander getrennt werden können und somit Unsicherheiten über die Anzahl der gesendeten Signale entstehen. Ein bekannter Standard auf Basis dieses Verfahrens ist der EIA-232, besser bekannt als RS-232. [DALLAS, 1998]

Es existieren verschiedene andere Verfahren zur Umsetzung von logischen Werten in Spannungslevel, wie das Manchestercoding oder die Return-to-Zero-Methoden (RZ). Bei dem Manchestercoding werden beide logischen Zustände durch ein positives und ein negatives Signal von jeweils halber Bitdauer repräsentiert. Eine 1 wird als Abfolge von Positiv-negativ und eine 0 als negativ-positiv dargestellt. Dadurch kann die Bildung eines Gleichstrompotentials in den Leitungen verhindert werden und Bits können klar voneinander getrennt werden. Nachteil dieser Methode ist, dass aufgrund der Verwendung von 2 Pegeln pro Bit die Frequenz der Pegelwechsel etwa doppelt so hoch ist wie bei Methoden mit konstanten Pegeln und dementsprechend die Taktraten für die Datenübertragung begrenzt sind. Die Return-to-Zero-Methoden verwenden ebenfalls bis zu zwei Spannungslevels pro Bit, dabei ist der erste Pegel der für den logischen Zustand charakteristische Pegel und der zweite Pegel ist 0. Die Spannungslevels werden jeweils für die Hälfte der Übertragungsdauer jedes Bits gehalten. Die Manchester- und RZ-Verfahren werden vorrangig für die möglichst fehlerfreie und kontinuierliche Übertragung von Daten mit geringer Geschwindigkeit verwendet. [FRENZEL, 2016]

Bei der Übertragung von seriellen Daten ist weiterhin die Art der Verbindung und in diesem Zusammenhang die Wahl eines passenden Übertragungsmedium zu beachten. Es kann in symmetrische und asymmetrische Verbindungen unterschieden werden. Die Prinzipien beider Verbindungsarten sind in Abbildung 2 dargestellt. [FRENZEL, 2016]

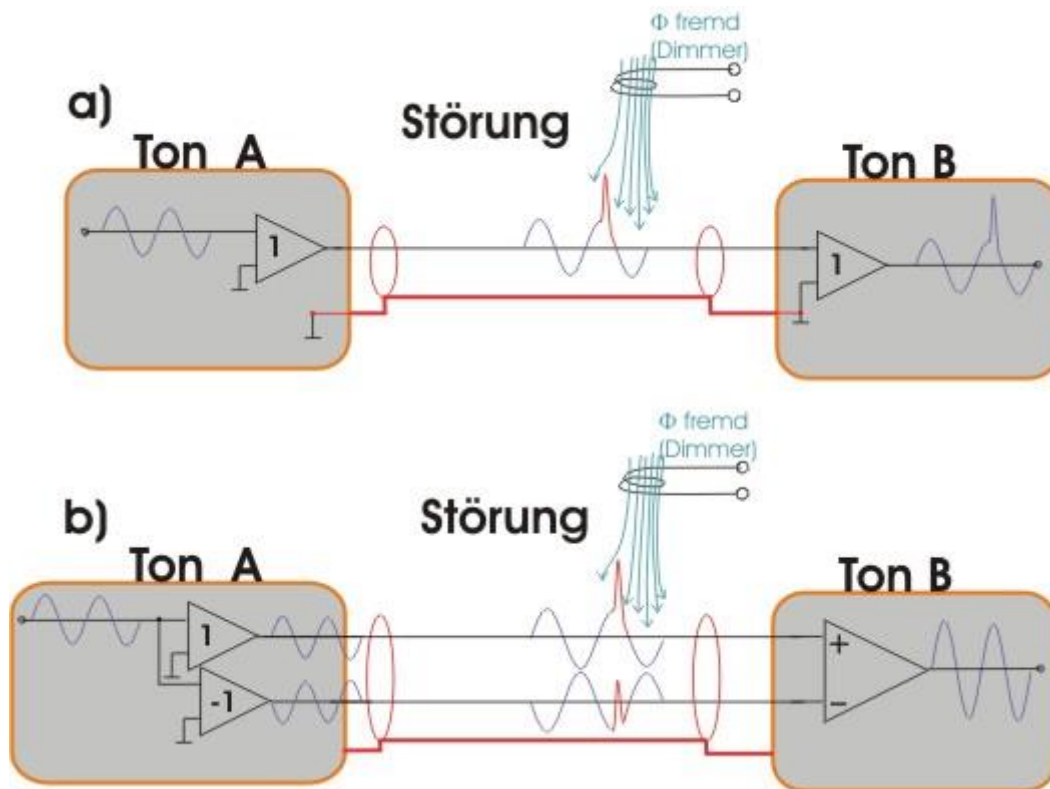


Abbildung 2: Prinzip der asymmetrischen (a) und symmetrischen (b) Verbindung als Anwendung in der Tontechnik [[http://hbernstaedt.de/Test/Asm\\_Asm.jpg](http://hbernstaedt.de/Test/Asm_Asm.jpg)]

Die asymmetrischen Verbindungen führen neben der Datenleitung eine weitere Leitung für die gemeinsame Masse mit. Diese Masseleitung wird als Referenz für das Spannungslevel im unbeschalteten Zustand, also als 0 V, verwendet. Die symmetrische Verbindung verwendet ein weiteres Datenkabel mit invertierter Polarität. Als Signal wird bei der symmetrischen Verbindung die Differenz der, auf den beiden Datenkabeln übermittelten, Spannungslevel erfasst. Der in der Abbildung gezeigte Störfall, beispielsweise durch elektromagnetische Felder in der Umgebung, wirkt sich in der asymmetrischen Verbindung direkt auf das empfangene Signal aus. In der symmetrischen Verbindung tritt die Störung in beiden Datenleitungen auf und kann von dem Empfänger durch die Ermittlung der Signaldifferenz korrigiert werden. Typisch sind das Koaxialkabel als Medium der asymmetrischen Verbindung und der Cannon-Stecker als Träger von symmetrischen Signalen. [FRENZEL, 2016]

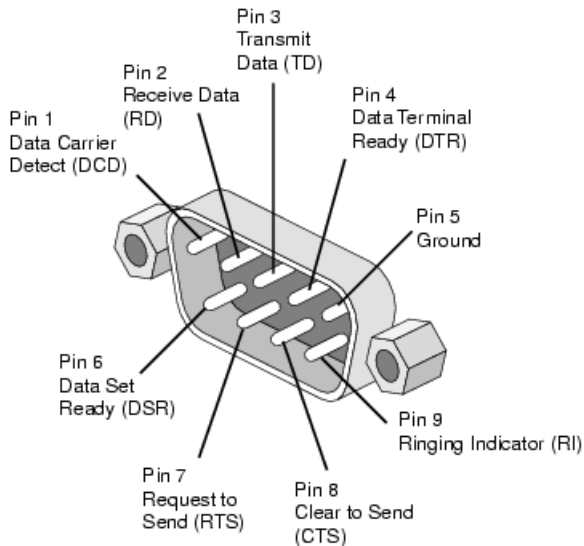
## 2.2 RS-232

RS-232 wurde als Standard für die serielle Direktverbindung zwischen einem Computer oder Terminal (Data Terminal Equipment, DTE) und einem Peripheriegerät (Data Circuit-Terminating Equipment, DCE) entwickelt. Ausgehend von dem Entwickler, Electronic Industry Association, lautet die ursprüngliche Bezeichnung EIA-232. Aufgrund der großen Popularität dieses Standards ist die verbreitetere Bezeichnung RS-232 für „Recommended Standard“. Der offizielle Name für den Standard lautet „Interface between Data Terminal Equipment and Data Circuit-Termination Equipment Employing Serial Binary Data Interchange“ [DALLAS, 1998].

Der Standard wurde erstmalig 1962 vorgestellt und war für die Kommunikation von Rechnern mit Modems ausgelegt. Aufgrund der Einfachheit und Robustheit des Standards und der wachsenden Anzahl von PCs mit entsprechender Schnittstelle entwickelte sich dieser Standard schnell zu dem verbreitetsten Standard für serielle Kommunikation und fand unter anderem in einigen der ersten Mäuse, Plotter, Scanner und den meisten industriellen Geräten Anwendung. Der Standard wurde mehrfach überarbeitet, die aktuellste Revision ist der EIA-232-F von 1997. Nach der Einführung des USB-Standards wurde der RS-232 weitgehend aus privaten Anwendungen verdrängt, stellt aber für die Steuerung der meisten industriellen Geräte weiterhin das verbreitetste Protokoll dar. [TEXAS INSTRUMENTS, 2002]

In dem Standard wurde als mechanisches Interface der 25-polige D-Sub-Stecker mit definierten Maßen und Pinbelegungen festgelegt. Mit dieser Anzahl Pins können alle in dem Standard festgehaltenen Signale und Funktionen bedient werden. Da für die meisten Anwendungen nicht die volle Ausnutzung aller möglichen Signale nötig ist, wird aus Platzgründen häufig der 9-polige D-Sub-Stecker DB9 verwendet. Die Belegung des DB9-Steckers ist in Abbildung 3 gezeigt. [TEXAS INSTRUMENTS, 2002]



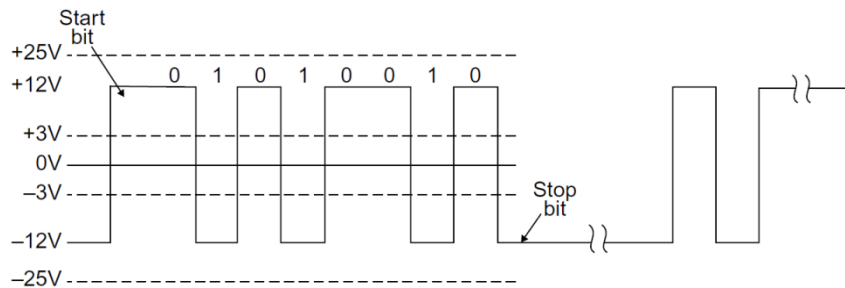


**Abbildung 3: Pinbelegung des 9-poligen D-Sub-Steckers** [<https://www.db9-pinout.com/db9-pinout/db9-pinout.gif>]

Pin 2 und Pin 3 sind die Leitungen zum Senden und Empfangen von Daten, Pin 5 bietet die Möglichkeit, eine gemeinsame Masseleitung aufzubauen. DTR und DSR auf Pin 4 und Pin 6 sind Signale zur Bereitschaft des Computers und des angeschlossenen Gerätes zur Datenübertragung. RTS und CTS auf Pin 7 und Pin 8 sind Signale für die Koordination der Datenübertragung, um Datenverlust durch blockierte Leitungen zu vermeiden. Die Koordination mit den genannten Datenleitungen erfolgt nach einem Protokoll, das als Handshake bezeichnet wird. Das RI-Signal über Pin 9 war ursprünglich für die Übertragung des Klingel-Signals zwischen Modem und Telefon vorgesehen und ist für moderne Anwendungen kaum noch relevant. [TEXAS INSTRUMENTS, 2002]

Beide Datenleitungen stellen eine asymmetrische Verbindung zwischen Sender und Empfänger dar, auf der Daten mit bipolarer NRZ-Codierung übertragen werden. Eine logische 0 wird von dem Sender durch eine Spannung zwischen 5 V und 25 V repräsentiert, eine logische 1 durch Spannungen zwischen -5 V und -25 V. Der Empfänger erkennt Signale zwischen 3 V und 25 V bzw. -3 V und -25 V, Spannungen zwischen -3 V und 3 V sind undefiniert. Dementsprechend beträgt der minimale Puffer für Störungen zwischen dem Sender und Empfänger 2 V. Anhand dieses Puffers wurde eine maximale Kapazität des Signalleiters mit 2500 pF festgelegt, was die maximale Leiterlänge auf etwa 15-20 m begrenzt. Als maximale Übertragungsrate wurden in dem Standard 20 kbit/s definiert, allerdings können in der Anwendung Datenraten bis etwa 120 kbit/s erreicht werden. Diese Datenraten

können allerdings nur mit deutlich kürzeren Datenkabeln und Treibern, die die benötigten Stromstärken für schnellere Spannungswechsel unterstützen, erreicht werden. [TEXAS INSTRUMENTS, 2002]



**Abbildung 4: Rahmenbildung nach RS-232 mit 8 Datenbits und einem Stoppsbit [Frenzel Handbook]**

Um den, bei der NRZ-Codierung auftretenden, Verlust des Datentaktes durch lange Folgen gleicher Signale zu verhindern, werden die Bits bei diesem Standard in definierte Rahmen verteilt. Jeder Rahmen beginnt dabei, wie in Abbildung 4 gezeigt, mit einem Start-Bit, gefolgt von 5 bis 8 Datenbits und maximal 2 Stoppbits. Optional kann ein Paritätsbit mitgeführt werden, um die Integrität der übertragenen Datenbits zu überprüfen. Jedes Signal, als Repräsentant eines Bits, wird dabei für eine festgelegte Dauer gesendet. Die Dauer des Signals hängt von der Baudrate der Verbindung ab. Dieser Wert gibt die maximale Anzahl von Signaländerungen pro Sekunde an und ist damit direkt proportional zur Übertragungsrate. Je höher die Baudrate, desto kürzer ist die Dauer jedes Bits. Die Baudrate, die Anzahl der Daten- und Stoppbits, das verwendete Handshakeprotokoll sowie die Art des optionalen Paritätsbits sind Parameter, die bei Computer und Peripheriegerät übereinstimmen müssen, um eine Kommunikation zu ermöglichen, und beeinflussen die Übertragungsgeschwindigkeit der Verbindung. [FRENZEL, 2016, TEXAS INSTRUMENTS, 2002]

## 2.3 Datenbanken

Datenbanken sind eine Form der organisierten elektronischen Datenspeicherung mit dem Ziel, die Speicherung, Formatierung, Verwaltung und Manipulation der Daten konsistent, platzsparend und performant zu realisieren. Die Daten können abhängig von der Form der Datenbank Texte, Zahlen, logische Werte oder Dateien sein. Innerhalb von Datenbanken können Transaktionen zur Manipulation der Daten durchgeführt werden. Die Transaktionen sind Abfolgen der vier grundlegenden



festgelegt ist. Das Tabellenschema gibt an, wie viele Attribute eine Tabelle hat, sowie deren Namen und Vorgaben für die Werte, wie vorgeschriebene Datentypen, Einmaligkeit der Werte oder die Möglichkeit, leere Werte einzufügen. [LAKE, 2014]

Jeder Datensatz muss über einen oder mehrere Schlüssel („keys“) eindeutig identifizierbar sein. In den meisten Fällen wird aus Gründen der Einfachheit und Skalierbarkeit ein fortlaufender Integer-Wert als ID genutzt, der dem Datensatz bei der Speicherung zugewiesen wird. Der Schlüssel bezieht sich nur auf den Datensatz, nicht auf die Position des Datensatzes in der Tabelle. [LAKE, 2014]

Sollen in einer relationalen Datenbank komplexere Daten verwaltet werden, werden Tabellen untereinander verknüpft. Die Verknüpfung der Tabelle „Autos“ mit der Tabelle „Einwohner“ ist in Abbildung 2 dargestellt.

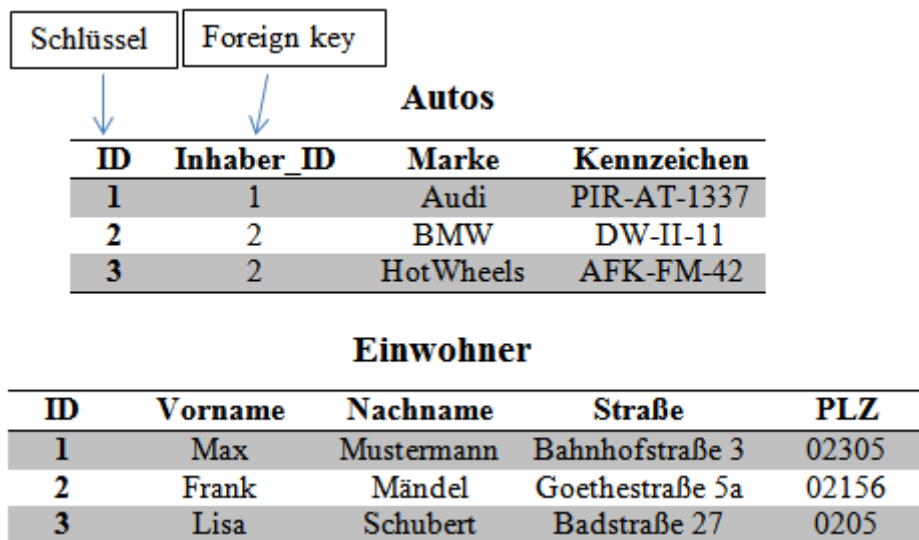


Abbildung 6: Verknüpfung von zwei Tabellen über einen foreign key

Um den Inhaber eines Autos in der Tabelle zu identifizieren, ist der Schlüssel des Datensatzes aus der Tabelle „Einwohner“ angegeben. Sollte ein Einwohner mehrere Autos besitzen, kann mehrfach auf den Eintrag verwiesen werden. Dadurch müssen die Daten des Einwohners nur ein einziges Mal angegeben werden und das Risiko für Inkonsistenzen bei Mehrfacheingaben, zum Beispiel durch Schreibfehler, wird vermieden. Ebenfalls vereinfacht wird die Manipulation der Daten. Ändern sich die Daten eines Einwohners, muss nur ein einziger Datensatz aktualisiert werden. Die Aufspaltung von Tabellen zur Vermeidung von Redundanzen zugunsten von strukturierten Verweisen wird Normalisierung genannt. Es wurden von E. Codd drei Normalformen (NF) definiert, die sequentiell die Struktur der Datenbank verbessern

sollen. Nachträglich wurden zwei weitere Formen angefügt, die allerdings in der Praxis wenig Anwendung finden. Für die Umsetzung einer Normalform ist es nötig, alle vorhergehenden Stufen der Normalisierung bereits umgesetzt zu haben. [HOPPE-KIAUK, 2004]

Die erste Normalform ist erreicht, wenn jedes Attribut nur atomare Werte enthält. Eine Verletzung der ersten Normalform wäre in dem Beispiel in Abbildung 6 das Attribut „Straße“ in der Tabelle „Einwohner“, das die atomaren Werte „Straßenname“ und „Hausnummer“ enthält. [HOPPE-KIAUK, 2004]

Die zweite Normalform ist erreicht, wenn jedes Attribut, das kein Schlüssel ist, von allen Schlüsselattributen abhängig ist. Tabellen in der ersten Normalform mit nur einem Schlüsselattribut befinden sich dementsprechend automatisch in der zweiten Normalform. In der dritten Normalform müssen alle Attribute ausschließlich von den Schlüsselattributen abhängig sein. Funktionale Abhängigkeiten zwischen Feldern, die keine Schlüssel sind, sind nicht erlaubt. [HOPPE-KIAUK, 2004]

Die meisten relationalen Datenbanksysteme unterstützen als Datenbanksprache SQL (Structured Query Language), eine in den 70er Jahren entwickelte und international standardisierte Sprache zur Definition der Datenstruktur und Manipulation der Datenbestände relationaler Datenbanken. Vorteile von SQL sind die einfache Syntax und die an die englische Sprache angelehnte Semantik. Die Implementierung der Sprache ist Teil des DBMS, Modifizierungen zur Anpassung des Funktionsumfanges und der Performance sind gängige Praxis. Die systemspezifischen Implementierungen von SQL werden als Dialekte bezeichnet. [HOPPE-KIAUK, 2004]

Ein verbreitetes DBMS ist MySQL von Oracle. MySQL ist sowohl als Open-Source-Software als auch als kommerzielle Enterpriseversion erhältlich und ist auf allen verbreiteten Betriebssystemen lauffähig. MySQL erfüllt die folgenden Kriterien [LAKE, 2013]:

- Atomarität: Eine Sequenz von Datenbankoperationen wird entweder vollständig oder gar nicht ausgeführt. Kann eine Operation der Sequenz nicht ausgeführt werden, werden alle vorherigen Operationen der Sequenz rückgängig gemacht und die Ausführung beendet.

- **Konsistenzerhaltung:** Jede durchgeführte Transaktion in der Datenbank hinterlässt einen konsistenten Datenzustand. Der konsistente Datenzustand bezieht sich dabei inhaltlich auf die Einhaltung von Datentypen oder Speichergrößen und strukturell auf die Existenz von referenzierten Datensätzen und die Einmaligkeit von Schlüsselattributen.
- **Isolation:** Garantie, dass bei der parallelen Durchführung mehrerer Transaktionen der finale Zustand dem Zustand nach der sequentiellen Durchführung der Transaktionen entspricht. Dieses Kriterium ist besonders wichtig für Online-Datenbanken mit mehreren Clients.
- **Dauerhaftigkeit:** Gespeicherte Daten müssen dauerhaft in der Datenbank hinterlegt sein und auch im Falle eines Systemfehlers nicht verloren gehen. Alternativ kann ein Transaktions-Log mitgeführt werden, um alle getätigten Transaktionen bei Datenverlust nachvollziehen und nachstellen zu können.

Als Werkzeug zur Administration von MySQL-Datenbanken wird in den meisten Fällen die Webanwendung phpMyAdmin verwendet, die neben einem SQL-Terminal auch eine graphische Oberfläche zur Anzeige und Manipulation der Daten bietet.

Als aktuelle Alternative zu relationalen Datenbanksystemen sind die NoSQL-Ansätze zu nennen. NoSQL steht dabei für „Not Only SQL“ und ist ein Sammelbegriff für Datenbankmanagementsysteme, die auf strukturierte Datenbanken und die entsprechenden Datenbanksprachen verzichten und flexiblere Strukturen zur Verfügung stellen. Diese Ansätze sind für die flexible und dynamische Erweiterung von Datenbanken gedacht, besonders für Daten, die mit relationalen Modellen schwer beschrieben werden können. Weiterhin sind relationale Datenbanken schlecht skalierbar, die Geschwindigkeit der Datenbankoperationen nimmt für große Datenmengen exponentiell ab. Dies ist teilweise auf die Einhaltung der ACID-Kriterien zurückzuführen und den damit verbundenen zusätzlichen technischen Aufwand. Die von Google Inc. entwickelte NoSQL-Datenbank BigTable hingegen ist für die Speicherung und Manipulation von Daten im Bereich von Petabytes ( $10^{15}$  Bytes) ausgelegt und bildet die Grundlage für verschiedene Services wie Google Maps oder Google Earth [CHANG, 2006]. Öffentlich verfügbare Systeme sind beispielsweise MongoDB oder Apache Cassandra und werden von bekannten Plattformen wie Twitter, eBay oder GitHub verwendet. [LAKE, 2013]

## 2.4 Softwarestacks

Als Softwarestacks, oder Softwarestapel, wird ein Satz aufeinander aufbauender Softwaresysteme bezeichnet, die zusammen eine Plattform ergeben. Für diese Stacks entwickelte Anwendungen funktionieren mit diesen Plattformen ohne zusätzliche Software. Man spricht davon, dass die Anwendungen „auf der Plattform laufen“. Softwarestacks werden eingesetzt, um einheitliche Umgebungen für Anwendungen zu erhalten und damit die Ausführbarkeit von Anwendungen auf verschiedenen Geräten zu garantieren. [SEIDLER, 2006]

### 2.4.1 Webservices

Software Services sind essentielle Bestandteile für die Entwicklung von verteilten Anwendungen. Es sind Module mit beschränktem Funktionsumfang, die über ein Interface in Softwarepakete eingebunden werden können. Werden Services über das Internet zur Verfügung gestellt, spricht man von Webservices. Diese werden durch eine festgelegte URL identifiziert und können über standardisierte Internetprotokolle angesprochen werden. Diese Anwendungen benötigen einen Softwarestack aus Betriebssystem, Webserver, Programmiersprache, Programmierwerkzeugen, Software für die Datenpersistenz und Frameworks. Bekannte Stacks sind beispielsweise das LAMP- und das XAMPP-Paket. LAMP besteht dabei aus den vier Hauptkomponenten Linux als Betriebssystem, Apache als Webserver, MySQL als Datenbanksystem und PHP als Programmiersprache. Zusätzlich können weitere Komponenten wie das Bootstrap-Framework installiert werden. Der Aufbau des LAMP-Paketes ist in Abbildung 7 gezeigt. [BOUGUETTAYA, 2014]

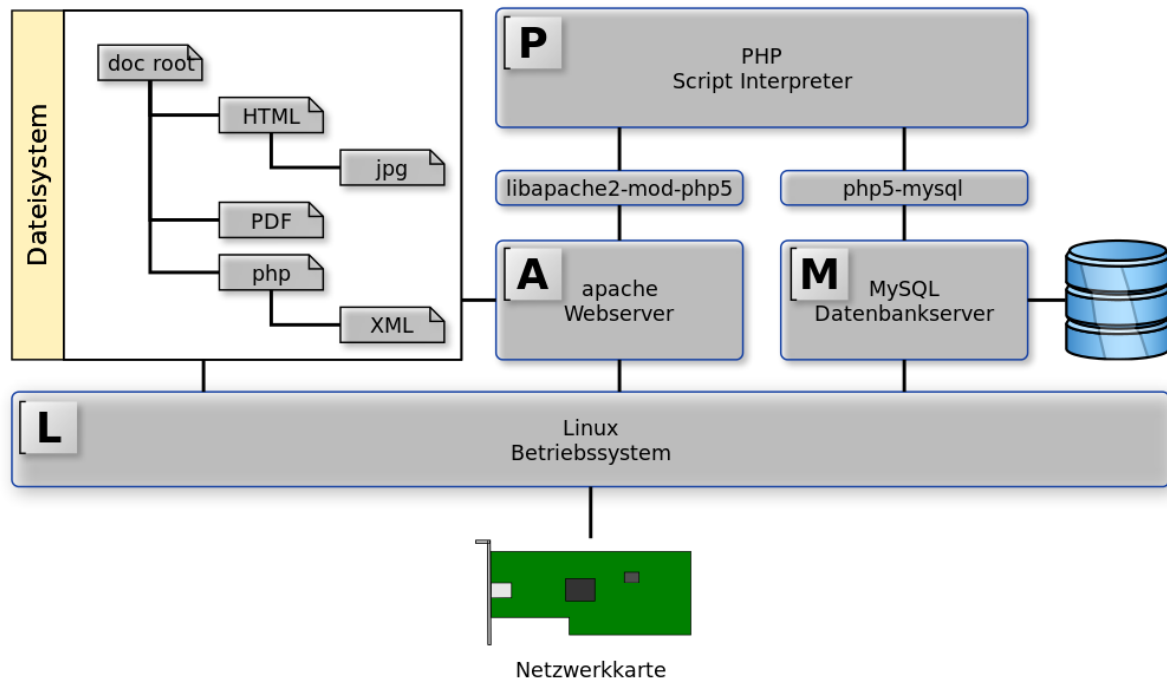


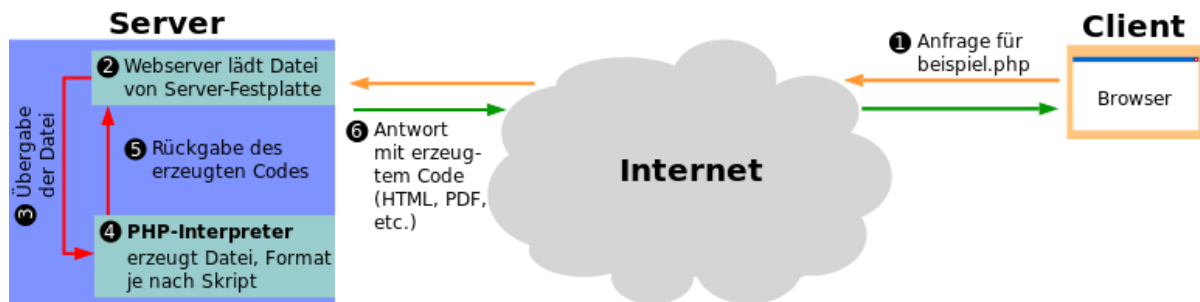
Abbildung 7: Aufbau des LAMP-Pakets (<https://i2.wp.com/blog.novatrend.ch/wp-content/uploads/2014/11/LAMP.png>)

Das Betriebssystem stellt die Grundlage für die Installation des Webserver und des Datenbanksystems dar. Dabei kann der Webserver auf das Dateisystem des Betriebssystems zugreifen und Dateien und Funktionen über das Internet zugänglich machen. Der Zugriff auf die Datenbank erfolgt über die Programmiersprache PHP mit den dargestellten Modulen für Apache und MySQL. Die Komponenten des Pakets können auch in anderer Zusammensetzung verwendet werden, beispielsweise auf dem Betriebssystem Windows aufbauend. Ein für viele Betriebssysteme einsetzbarer Stack ist das XAMPP-Paket, das die gleichen Komponenten verwendet wie das LAMP-Paket, allerdings die Kompatibilität mit beispielsweise Windows gewährleistet. [GILMORE, 2002]

Anfragen aus dem Netzwerk erreichen über die Netzwerkkarte das Betriebssystem. Anhand der explizit oder implizit in der Anfrage enthaltenen Portnummer wird die Anfrage an das entsprechende Programm weitergeleitet. Erreicht eine Anfrage den Webserver, holt der Webserver anhand der enthaltenen Adresse eine Datei aus dem Dateisystem. Handelt es sich dabei um HTML-Dokumente mit Skripten, wird der Script-Interpreter aufgerufen. In den meisten Fällen handelt es sich dabei um PHP oder .NET-Technologien [W3TECHS, 2017]. Werden in den Skripten Informationen aus der Datenbank benötigt, ruft der Interpreter das Datenbanksystem auf. Das bearbeitete HTML-Dokument wird an das Betriebssystem übergeben und über die



Netzwerkkarte versandt. Der schematische Ablauf einer Abfrage mit PHP ist zusammenfassend in Abbildung 8 gezeigt.



**Abbildung 8: Dynamische Erstellung von Inhalten mit PHP**  
[https://de.wikipedia.org/wiki/LAMP\\_\(Softwarepaket\)#/media/File:PHP\\_funktionsweise.svg](https://de.wikipedia.org/wiki/LAMP_(Softwarepaket)#/media/File:PHP_funktionsweise.svg)

Der Client sendet, beispielsweise mit einem Browser, eine Anfrage mit einer Adresse an den Server. Der Webserver lädt die von der Adresse identifizierte Datei aus dem Dateisystem und übergibt sie an den PHP-Interpreter. Der Interpreter führt den, in der Datei enthaltenen, Code aus und erzeugt eine dynamische Antwort, meist in Form von HTML-Code. Die erzeugte Datei wird an den Webserver zurückgegeben und über das Internet an den Client versendet. [GILMORE, 2002]

### 2.4.2 C#

C# ist eine objektorientierte Programmiersprache, die wie Java und C++ zur Familie der C-Sprachen zählt. Die Sprache ist für den Einsatz mit dem .NET-Framework optimiert und ist kompatibel mit .NET-Komponenten und COM-Komponenten. Weiterhin unterstützt es innerhalb der .NET-Sprachen sprachübergreifende Vererbung, Debugging und Fehlerbehandlung. Aufgrund der Abhängigkeit von der .NET-Plattform war C# anfangs auf die Anwendung auf Windowsplattformen beschränkt. Als Resultat der zahlreichen Portierungen, beispielsweise das Mono .NET-Framework für Linux oder MacOS, kann C# inzwischen auch plattformübergreifend verwendet werden. [TROELSEN, 2015]

Um Software auf einem Rechner auszuführen, muss zuerst der programmierte Code in maschinenlesbare Sprache umgeschrieben werden. Dieser Schritt wird Kompilieren genannt und setzt einen Compiler voraus, der mit der verwendeten Programmiersprache kompatibel ist. Der Compiler liest den Code ein, analysiert auf Fehler und speichert die Software als Assembler-Code, den das vorliegende Betriebssystem verarbeiten kann. Die erstellte Datei ist dementsprechend ausschließlich für das vorliegende Betriebssystem geeignet. Um die Kompatibilität

über mehrere Systeme hinweg zu ermöglichen, wurde mit der .NET-Plattform deshalb die Common Intermediate Language (CIL) eingeführt. Alle .NET-Sprachen bieten einen Compiler, der den Programmcode anstelle des Assembler-Codes in CIL-Code umwandeln kann. Der CIL-Code kann zur Laufzeit von einem, in der .NET-Plattform enthaltenen, Compiler in Assembler-Code umgewandelt. Der CIL-Code ist dementsprechend universell auf jedem System ausführbar, das eine Form der .NET-Plattform installiert hat. [RAHMAN, 2014]

## 3 Konzeption

### 3.1 Serverseitig

#### 3.1.1 Datenbank- und Speicherstruktur

Als Grundlage für die Datenbank wurde das bestehende Chemikalienverzeichnis, eine Excel-Tabelle mit allen gelieferten Chemikalien, verwendet. In dem Verzeichnis wurden die folgenden Eigenschaften jeder Lieferung gespeichert:

- Name der Chemikalie
- CAS-Nummer
- Kategorie
- Unterkategorie
- Lagerungsvorschrift
- Hersteller
- Menge
- Qualität/Spezifikationen
- Batch-/Chargennummer
- Summenformel
- Molmasse
- Gefahrstoff
- Zugehöriges Sicherheitsdatenblatt
- R-/S-Sätze
- H-/P-Sätze

Um den Verbrauch von Chemikalien zu erfassen, wurde die Liste um die folgenden Attribute erweitert:

- Lieferdatum
- Lieferung geöffnet
- Öffnungsdatum
- Restmenge in g

Anhand der Attribute wurde eine Datenbankstruktur entwickelt, die die Eigenschaften in mehreren Tabellen zusammenfasst. Die Struktur ist in Abbildung 9 gezeigt.

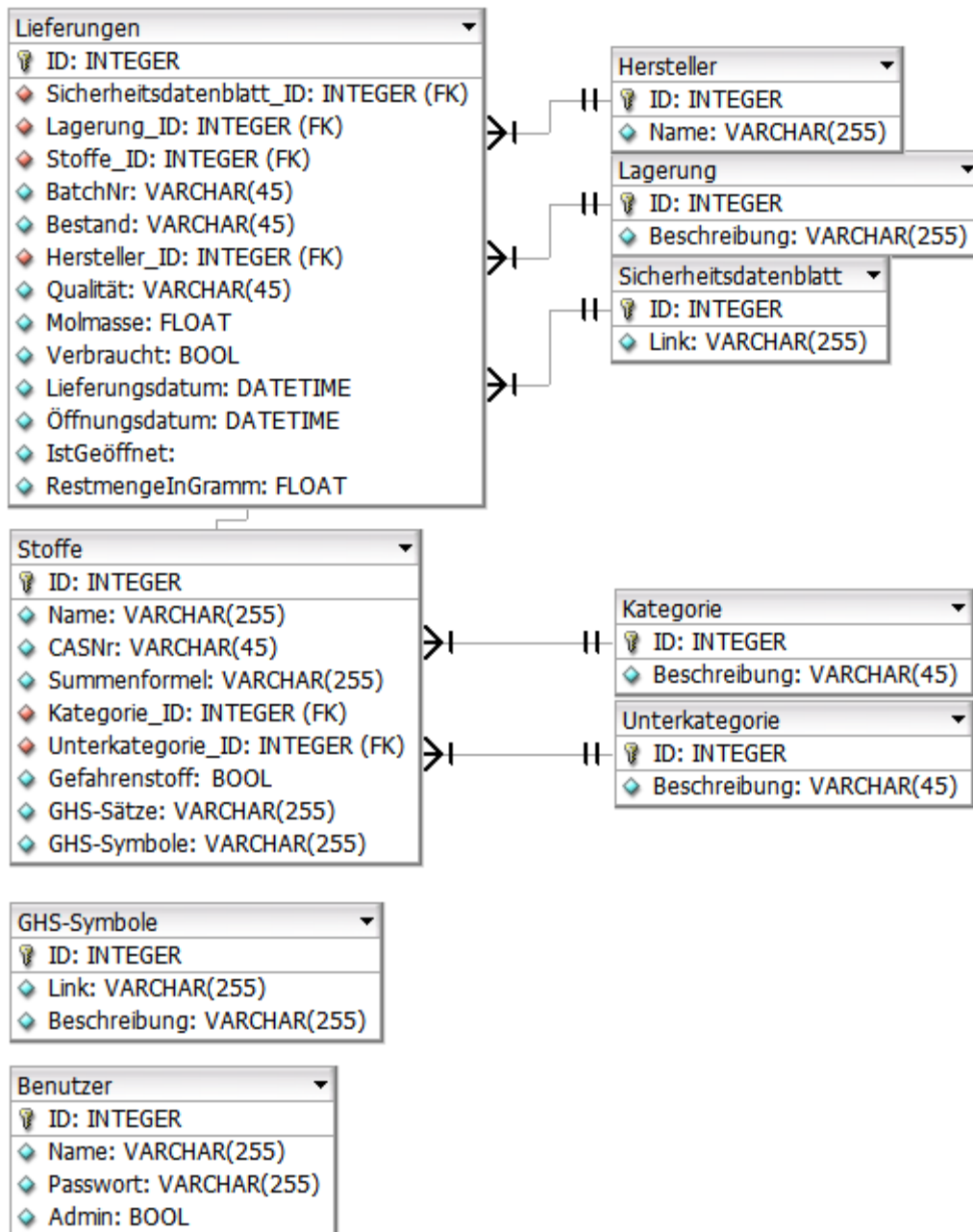


Abbildung 9: Konzeptionelle Datenbankstruktur mit Attributen und zugehörigen Datentypen

Die Attribute wurden anhand ihrer Abhängigkeiten in Tabellen eingeteilt. Jeder Block repräsentiert eine Tabelle, die Zeilen in dem Block die zugehörigen Attribute. Nach den Namen der Attribute sind die jeweiligen geforderten Datentypen vermerkt. *Varchar* erfordert eine beliebige Zeichenkette, *int* eine Ganzzahl, *float* eine Gleitkommazahl und *bool* einen Wahrheitswert. Die in Klammern stehenden Zahlen geben die maximale Länge der zu speichernden Zeichenketten an.

Als zentrale Tabelle wurden die Lieferungen gewählt. Jede gelieferte Chemikalie sollte in diese Tabelle eingetragen werden, unter Angabe von lieferungsspezifischen

Parametern wie der Batchnummer, der Liefermenge und dem Lieferdatum. Um Redundanzen zu vermeiden, wurden eigene Tabellen für die Hersteller, Lagerungsvorschriften, Sicherheitsdatenblätter und Stoffe angelegt, auf deren Einträge in der Lieferungstabelle verwiesen wird. Die Tabelle der Stoffe enthält alle, für die Chemikalien spezifischen, Parameter. Dazu gehören der Name und die CAS-Nummer sowie die Einteilung in Kategorien und Unterkategorien. Die Gefahrstoffsätze wurden als eine vollständige Zeichenkette zusammengefasst. Die Gefahrstoffsymbole nach dem Global harmonisierten System zur Einstufung und Kennzeichnung von Chemikalien (GHS) wurden ebenfalls in der Tabelle „Stoffe“ als Zeichenkette abgelegt. Diese Zeichenkette sollte als kompakter Schlüssel für die zugehörigen Symbole dienen. Die Symbole wurden als Einträge in die Tabelle „GHS-Symbole“ abgelegt, mit einem Link zu der zugehörigen Abbildung. Weiterhin wurde eine eigenständige Tabelle „Benutzer“ angelegt, in der die Benutzernamen und Passwörter von Nutzern des Datenbanksystems gespeichert werden können. Der Wahrheitswert „Admin“ in der Tabelle bezieht sich auf die Gewährung von Administratorrechten.

Die Gefahrstoffsymbole wurden als Bilddateien in der Ordnerstruktur des Webserver abgelegt. Die Sicherheitsdatenblätter wurden als PDF-Dateien ebenfalls in der Ordnerstruktur hinterlegt.

### 3.1.2 Webinterface

Für den Zugriff auf die Chemikaliendatenbank sollte eine browserbasierte Benutzeroberfläche erstellt werden, die folgende Funktionalitäten bieten sollte:

- Anzeige aller Lieferungen
  - Filter für die angezeigten Lieferungen
  - Ein- und Ausblenden von verbrauchten Chemikalien
  - Anzeige von Restmengen
- Bearbeiten von Einträgen
- Löschen von Einträgen
- Einfügen
  - Neuer Chemikalien
  - Neuer Lieferungen
  - Sonstiger Datensätze (Hersteller, Kategorien etc.)

Der Zugang zu der Nutzeroberfläche sollte passwortgeschützt sein. Bei Aufruf der Startseite des Webinterfaces sollte der Benutzer also erst auf ein Anmeldeformular weitergeleitet werden. Die Eingaben sollten über ein PHP-Skript mit den Einträgen in der „Benutzer“-Tabelle der Datenbank verglichen werden. Bei Verifizierung der Eingaben sollte der Nutzer auf die Startseite der Benutzeroberfläche gelangen.

Auf der Startseite sollte eine Auflistung aller Lieferungen angezeigt werden. Ebenfalls sollten einige Filter zur Verfügung stehen, um gezielt nach Datensätzen zu suchen. Um die Übersichtlichkeit zu erhöhen, sollten verbrauchte Lieferungen standardmäßig ausgeblendet sein und, nach explizitem Einblenden, durch eine Formatierung der Schriftart erkennbar sein. Die Gefahrstoffe sollten farblich gekennzeichnet sein. Auf der Übersichtsseite sollten wesentliche Informationen zu den Lieferungen sichtbar sein, für eine detaillierte Ansicht aller gespeicherten Attribute einer spezifischen Lieferung sollte eine separate Seite bereitgestellt werden. Weiterhin sollte eine, für alle Seiten standardisierte, Navigationsleiste eingefügt werden, über die alle Funktionen erreichbar sein sollten.

Über die Navigationsleiste sollten die Seiten für das Eintragen von neuen Chemikalien, Lieferungen und sonstigen Datensätzen wie Hersteller oder Kategorien erreichbar sein. Jede dieser Seiten sollte ein Formular enthalten, in dem die Eingabe der benötigten Daten erfolgt. Für Attribute, die einen Verweis auf Datensätze anderer

Tabellen darstellen, sollten Drop-Down-Listen für die Auswahl der hinterlegten Datensätze bereitgestellt werden. Nach Bestätigung der Eingaben sollten alle Werte auf Konformität mit den, in der Datenbank vorgegebenen, Datentypen der entsprechenden Attribute und auf grobe Fehler geprüft werden. Wurden keine Konflikte entdeckt, sollten die Einträge in der Datenbank gespeichert und der Nutzer auf die Startseite weitergeleitet werden.

Es sollte ein separater Bereich für die direkte Verwaltung der Datenbank angelegt werden, der nicht über Schaltflächen erreichbar ist. Der Zugang sollte auf Benutzer beschränkt werden, die Administrator-Rechte besitzen. Benutzer ohne diese Rechte sollten aus dem administrativen Bereich der Oberfläche automatisch auf die Startseite umgeleitet werden. In diesem Bereich sollten die folgenden Funktionalitäten bereitgestellt werden:

- Verwaltung der zugelassenen Nutzer
- Export und Download der Datenbank
- Entfernen aller Datensätze
- Direkter Zugriff auf alle Datensätze über das Tool phpMyAdmin

### **3.1.3 Zugangsbeschränkung und Rechteverwaltung**

Die Zugangsbeschränkung und Rechteverwaltung in der Datenbank sollte über die Tabelle „Benutzer“ in der Datenbank erfolgen. In der Tabelle sollten alle Nutzer mit Name und Passwort eingespeichert werden. Weiterhin sollte die Verwaltung der Administrator-Rechte über einen, in dem Datensatz enthaltenen, Wahrheitswert erfolgen. Bei Anmeldung des Nutzers sollten der Name und gegebenenfalls der Administrator-Status für die Dauer der Sitzung gespeichert werden.

## 3.2 Clientseitig

In dieser Arbeit wird das Programm für die Anbindung der Waagen sowie alle weiteren, auf den Client-Rechnern gespeicherten, Daten von Relevanz für die Funktionsfähigkeit des Programms als „clientseitige Software“ bezeichnet. Ein Client-Rechner ist dabei jeder vom Server verschiedene Rechner, der mit dieser Software ausgestattet und an die Datenbank angeschlossen ist.

### 3.2.1 Benutzeroberfläche

Bei der Benutzeroberfläche der Kommunikationssoftware für die Waagen sollte darauf geachtet werden, die Bedienung auf Einfachheit und Schnelligkeit zu optimieren. Die Oberfläche sollte folgende Funktionen bieten:

- Zugangsbeschränkung mit Passwortschutz
- Durchsuchen der Datenbank nach Chemikalienlieferungen
- Automatisches Erfassen und Anzeigen von Wägewerten
- Manuelle Korrektur der Messwerte
- Speichern der Einwaagen in der Datenbank
- Konfiguration der Datenbankverbindung
- Konfiguration der seriellen Schnittstelle
- Überwachung der seriellen Kommunikation
- Serielles Terminal für die direkte Übertragung von Befehlen an die Waage

Die Funktionen sollten in separate Bereiche gegliedert werden, um die Oberfläche übersichtlich zu gestalten. Nach dem erfolgreichen Anmelden sollte eine Oberfläche mit den wichtigsten, für die vorgesehenen Arbeitsschritte relevanten, Funktionen erscheinen. In diesem Bereich sollte die Auswahl einer Chemikalienlieferung aus der Datenbank sowie die Erfassung und manuelle Korrektur von aktuellen Wägewerten möglich sein. Die Speicherung der Einwaagen sollte, ebenfalls in diesem Bereich, entweder manuell über eine Bestätigung oder automatisch nach dem Erfassen der eingewogenen Gewichte erfolgen.

In einem weiteren Bereich sollte die Konfiguration der seriellen Schnittstelle möglich sein. Dabei sollte die Auswahl der verwendeten Baudraten, Stopp- und Paritätsbits sowie des Handshakes und der Nummer des verwendeten seriellen Ports einzustellen sein. Weiterhin sollte in diesem Bereich ein einfaches serielles Terminal



zu Verfügung gestellt werden, um die serielle Kommunikation mit der Waage zu überwachen und direkt Befehle an die Waage übertragen zu können.

Zudem sollte ein Bereich für die Konfiguration der Datenbankverbindung bereitgestellt werden, in dem die Adresse, der verwendete Port sowie die Zugangsdaten für den zu verwendenden Datenbankbenutzer festzulegen sind.

Alle getroffenen Einstellungen, für die serielle Schnittstelle wie für die Datenbankverbindung, sollten gespeichert und bei einem Neustart des Programmes wieder geladen werden. Das Passwort für den Datenbankzugriff sollte dabei verschlüsselt gespeichert werden.

### **3.2.2 Serielle Schnittstelle zur Waage**

Als Anschluss am Rechner sollte ein COM-Port, eine RS-232-Schnittstelle, verwendet werden. Dabei handelt es sich um einen 9-poligen seriellen Anschluss, der besonders bei Modems, aber früher auch bei Rechnern häufig integriert wurde. Es sollte zur Entwicklung der Software vorerst nur eine bestimmte Waage angeschlossen werden und eventuell später die Unterstützung verschiedener Waagen integriert werden. Für diesen Zweck sollte die Präzisionswaage KERN ABJ120-4M mit einem Computer verbunden und für die Einbindung in die Software konfiguriert werden.

Es sollte anschließend ein Modul programmiert werden, über das die Kommunikation mit der Waage realisiert werden sollte. Dieses Modul sollte einerseits in der Lage sein, beliebige Befehle an die Waage zu senden und andererseits die empfangenen Informationen verarbeiten und an die grafische Bedienfläche der Software übergeben zu können. Dieses Modul sollte auf den nativen Komponenten des .NET-Frameworks aufgebaut werden, um Kompatibilität mit jedem .NET-fähigen Betriebssystem zu gewährleisten.

### **3.2.3 Datenbankschnittstelle**

In dem .NET-Framework Version 4.6 sind keine Treiber für die Verbindung mit MySQL-Datenbanken vorhanden, deshalb sollte für dieses Modul auf frei erhältliche Software aus Open-Source-Projekten zurückgegriffen werden. Mithilfe dieser Software sollte ein Modul entwickelt werden, dass auf lokale und externe MySQL-Datenbanken zugreifen kann. Das Modul sollte die wichtigsten Befehle, wie das gefilterte Abrufen von Chemikalienlieferungen und das Eintragen des Verbrauchs,

bereitstellen. Das Modul sollte dabei eine Abstraktionsschicht zwischen der Benutzeroberfläche und der Datenbank darstellen, um die Ausführung von ungewollten und potentiell schädlichen SQL-Befehlen durch die Nutzer zu verhindern.

## 4 Geräte und Software

Für diese Arbeit wurde eine Präzisionswaage des Modells ABJ120-4M von KERN verwendet. Für die Programmierung wurden die folgenden Entwicklungsumgebungen verwendet:

Tabelle 1: Liste der verwendeten Entwicklungsumgebungen

| Software                      | Entwickler            |
|-------------------------------|-----------------------|
| Visual Studio C# 2010 Express | Microsoft Corporation |
| Visual Studio Community 2017  | Microsoft Corporation |
| NetBeans IDE                  | Oracle Corporation    |
| PhpStorm                      | JetBrains             |
|                               |                       |

Weiterhin wurden im Zuge der Softwareentwicklung die folgenden Werkzeuge und Softwarepakete genutzt:

Tabelle 2: Liste der verwendeten Softwarepakete und Programmierwerkzeuge

| Software                 | Entwickler          |
|--------------------------|---------------------|
| PuTTY                    | Simon Tatham        |
| Connector/Net            | Oracle Corp.        |
| Blowfish Block Cipher C# | Defuse.ca           |
| XDebug                   | Derick Rethans      |
| DB Designer Fork         | Fabian Becker       |
| jQuery                   | jQuery Team         |
| MySQL Dumper             | Daniel Schlichtholz |
| XAMPP                    | Apache Friends      |

## 5 Implementation

### 5.1 Serverseitig

#### 5.1.1 Datenbank- und Speicherstruktur

Die unter 3.1.1 beschriebene Datenbankstruktur wurde während des Programmierprozesses mehrfach verändert, um weitere Funktionen implementieren zu können. Die finale Struktur der Datenbank ist in Abbildung 10 gezeigt:

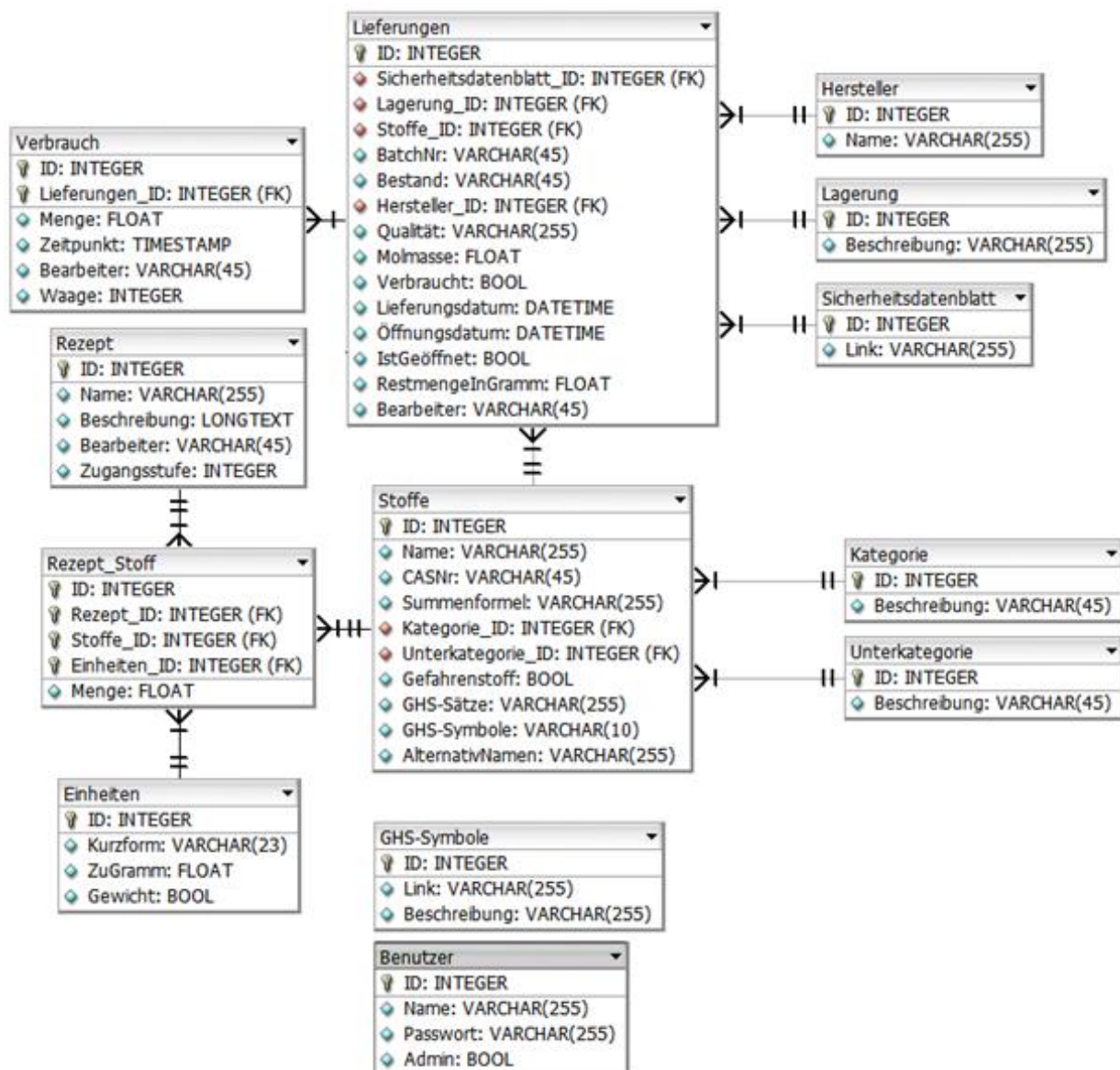


Abbildung 10: Implementierte Datenbankstruktur des Chemikalienverzeichnisses mit Attributen und zugehörigen Datentypen

Verglichen mit der Konzeption, in Abbildung 9 gezeigt, wurden in der finalen Struktur 4 weitere Tabellen integriert sowie jeweils ein zusätzliches Attribut in die Tabellen

„Lieferung“ und „Stoffe“ eingetragen. Es wurde eine Tabelle für Rezepte angelegt, in denen häufig genutzte Stoffzusammensetzungen erfasst und mit einer Beschreibung gespeichert werden können. Um eine variable Anzahl von Stoffen einem Rezept zuzuordnen, wurde eine Zwischentabelle namens „Rezept\_Stoff“ implementiert, in der eine bestimmte Stoff-ID mit einer Rezept-ID verknüpft werden kann. Weiterhin wurde in dieser Tabelle die benötigte Menge gespeichert. Die Masse wurde aus einer Gleitkommazahl und einer Einheit gebildet, wobei die Einheit in einer weiteren Tabelle gespeichert wurde. Zusätzlich zu der Einheit wurde eine Gleitkommazahl, „zuGramm“, gespeichert, die das Verhältnis der Einheit zu Gramm angibt, beispielsweise 0,001 für mg oder 0,035274 für Unzen. Einheiten, die nicht in Gramm umgerechnet werden können, wurden mit einem Faktor von 0 gespeichert. Das Attribut „Gewicht“ war ein Wahrheitswert, der angibt, ob die Einheit ein Gewicht darstellt.

In der Tabelle „Verbrauch“ wurde jede registrierte Nutzung von Chemikalien aufgezeichnet, einschließlich der Menge und dem Zeitpunkt der Einwaage. Anhand der Einträge in die Verbrauchstabelle könnten später Statistiken zur Chemikaliennutzung erstellt oder die genauen Einwaagen eines hergestellten Reagenz nachvollzogen werden. Weiterhin könnte anhand dieser Tabelle der, in den Lieferungen gespeicherte, Restbestand der Chemikalienlieferungen korrigiert oder neu berechnet werden, wenn ein Fehler in der Verbrauchshistorie festgestellt werden sollte.

Zu jedem Rezept sowie zu jeder Lieferung und jedem eingetragenen Verbrauch wurde außerdem gespeichert, welcher Bearbeiter für den Eintrag verantwortlich war. Zu den gespeicherten Stoffen konnten außerdem alternative Namen eingetragen werden, um die Suche nach bestimmten Chemikalien zu erleichtern. Beispielsweise konnten in diesem Attribut Abkürzungen und umgangssprachliche Bezeichnungen vermerkt werden oder die Namen in verschiedenen Übersetzungen eingespeichert werden.

### 5.1.2 Webinterface

Das Webinterface wurde, wie unter 3.1.2 beschrieben, implementiert. Die Anmeldung erfolgte über einen firmeninternen Authentifikationsserver, der die Benutzer- und Rechteverwaltung der Anwender übernimmt. Für den Zugang zu dem Webinterface war ein Account in dem firmeninternen Netzwerk nötig, der in einer definierten Nutzergruppe des Laborpersonals eingetragen sein musste. War der Account Teil der Administratorengruppe für die Chemikaliendatenbank, erhielt er bei der Anmeldung zusätzlich Zugang zu dem administrativen Bereich der Nutzeroberfläche. Die Zugangsrechte und der Name des Benutzers wurden daraufhin in der Sitzung des Browsers gespeichert und waren gültig, bis der Browser geschlossen wurde oder die Session-Variable, beispielsweise durch eine Abmeldung von der Oberfläche, zurückgesetzt wurde.

Die Verwendung der Tabelle „Nutzer“ für die Nutzerverwaltung, wie in der Konzeption vorgesehen, wurde ebenfalls implementiert, allerdings zugunsten der Integration in die firmeninterne Nutzerverwaltung deaktiviert.

Nach der Anmeldung wurde der Nutzer auf eine Übersichtsseite weitergeleitet, die alle aktuell eingetragenen Lieferungen anzeigt. Ein Ausschnitt der Übersichtsseite ist in Abbildung 11 gezeigt.

The screenshot shows the 'Chemikalienverzeichnis' (Chemical Inventory) web interface. At the top, there is a navigation bar with links: 'Lieferung eintragen', 'Neue Chemikalie definieren', 'SDB hochladen', 'Ressource eintragen', 'Inhalt Chemikalienschrank', 'Rezepte', and 'Logout'. Below this is a header with the text 'We Let Your Data Speak' and the 'quo data' logo. The main title is 'Chemikalienverzeichnis'. A 'Filter' section on the left allows searching by Name, CAS-Nr., Summenformel, Kategorie, Unterkategorie, Hersteller, and Lagerung, with checkboxes for 'Inklusive verbrauchter Chemikalien' and 'Nur Gefahrstoffe anzeigen'. A 'Suchen' button is below the filters. The main content is a table of chemical deliveries.

| Name  | CAS-Nr.  | BatchNr    | Bestand | Restmenge | Lieferdatum | Hersteller    | Lagerung                       | Aktionen |
|---|----------|------------|---------|-----------|-------------|---------------|--------------------------------|----------|
| <a href="#">11-Mercaptoundecanoic acid</a>          | 71310... | BGBC0041V  | 5 g     | 0.000000  | 1901-01-01  | Sigma-Aldrich | RT, Chemikalienschrank         |          |
| <a href="#">2-Nitrophenyl-β-D-galactopyranoside</a> | 369-0... | 041167070  | 5g      | 0.000000  | 1901-01-01  | Roth          | Kühlschrank Labor 1a           |          |
| <a href="#">5α-Androstan-17β-ol-3-on (DHT)</a>      | 521-1... | FE06021403 | 1 ml    | 0.000000  | 1901-01-01  | Sigma-Aldrich | Gefrierfach Kühlschrank Lab... |          |
| <a href="#">5α-Androstan-17β-ol-3-on (DHT)</a>      | 521-1... | BCBF1698V  | 5 g     | 0.000000  | 1901-01-01  | Fluka         | Kühlschrank Labor 1a           |          |
| <a href="#">Agarose Broad Range</a>                 | k.A.     | 066240487  | 10 g    | 0.000000  | 2016-11-03  | Roth          | RT, Chemikalienschrank         |          |
| <a href="#">Ammonium metavanadate</a>               | 7803-... | MKBT2265V  | 50 g    | 0.000000  | 1901-01-01  | Sigma-Aldrich | RT, Chemikalienschrank         |          |

Abbildung 11: Ausschnitt der Übersichtsseite für die gefilterte Anzeige von Chemikalienlieferungen

Unter der Kopfzeile und der Navigationsleiste wurde das Formular für die Filterfunktion positioniert. Als Filter wurden die Attribute (Chemikalien-)„Name“, „Summenformel“, „CAS-Nummer“, „Kategorie“, „Unterkategorie“, „Hersteller“ und „Lagerung“ bereitgestellt. Weiterhin bestand die Möglichkeit, verbrauchte Lieferungen in die Suche einzubeziehen oder ausschließlich Gefahrstoffe anzuzeigen.

In der Übersicht wurden ausgewählte Attribute der Lieferungen gezeigt. Die Attribute „Name“, „CAS-Nummer“, „Batchnummer“, und „Lieferdatum“ wurden ausgewählt, um Lieferungen auf einen Blick identifizieren zu können. Die Attribute „Hersteller“, „Bestand“ und „Lagerung“ wurden gewählt, um das Auffinden der Lieferung in den Laboren zu erleichtern. Weiterhin wurde das Attribut „Restmenge“ angezeigt, um Lieferungen mit niedrigem Restbestand schnell erkennbar zu machen.

Der Name der gelieferten Chemikalien wurde mit einem Hyperlink belegt, der auf eine Detailansicht der jeweiligen Lieferung verweist. Auf dieser Seite, gezeigt in Abbildung 12, wurden alle gespeicherten Attribute der Lieferung dargestellt, sowie ein Textfeld für die Ergänzung von weiteren Bezeichnungen der jeweiligen Chemikalien.

## Details zu Imidazol

**ID:** 421  
**Bearbeiter:** Nagel  
**Stoffname:** Imidazol  
**Andere Namen:**  
**CAS-Nummer:** 288-32-4  
**Summenformel:** C<sub>3</sub>H<sub>4</sub>N<sub>2</sub>  
**Batch-Nummer:** 296244375  
**Verpackungseinheit:** 50 g  
**Hersteller:** Roth  
**Qualität:** >=99 %  
**Molmasse:** 68,08  
**Verbraucht:** Nein  
**Lieferungsdatum:** 2017-02-06  
**Öffnungsdatum:**  
**Kategorie:** Chemikalie  
**Unterkategorie:** Proteinbestimmung  
**Lagerungshinweis:** RT, Chemikalienschrank  
**Gefahrstoff:** Ja  
**R:**  
**S:**  
**P:** 260-280-301+330+331-303+361+353-305+351+338-308+313  
**H:** 302-314-360D

GHS-Symbole:



SDB-Link: [Imidazol.pdf](#)

Synonyme eintragen

Synonyme speichern

Abbildung 12: Detailansicht einer beispielhaften Lieferung

Am rechten Rand der Tabelle in Abbildung 11 wurden weiterhin zwei Schaltflächen implementiert, die das Löschen oder Bearbeiten der Lieferungen erlauben. Die Seite für das Bearbeiten der Lieferungen entspricht strukturell der Oberfläche zum Eintragen neuer Lieferungen, wie gezeigt in Abbildung 13.

## Lieferung bearbeiten

### Navigation

[Übersicht](#)  
[Lieferung eintragen](#)  
[Neue Chemikalie definieren](#)  
[SDB hochladen](#)  
[Ressource eintragen](#)  
[Inhalt Chemikalienschrank](#)  
[Rezepte](#)  
[Logout](#)

|  |   |               |                               |
|--|---|---------------|-------------------------------|
| 11- Mercaptoundecanoic acid  | Wählen sie den Stoff aus  | Sigma-Aldrich | Wählen sie den Hersteller aus |
| 218,36   | Molmasse  | BGBC0041V     | Batch-/Chargen-Nr.            |
| 5 g  | Verpackungseinheit  | 0,95          | Qualität/Spezifikationen      |
| 1901-01-01   | Lieferdatum   | 1901-01-01    | Öffnungsdatum                 |
| RT, Chemikalienschrank   | Wählen sie die Lagerungsvorschrift aus  |               |                               |
| SDB/Gefahrenstoffe/11-Mercap   | Wählen sie das Sicherheitsdatenblatt aus oder <a href="#">laden sie es hoch</a> |               |                               |
| Verbraucht: <input checked="" type="radio"/> Nein <input type="radio"/> Ja |   |               |                               |
| <input type="button" value="Speichern"/>                                   |   |               |                               |

Abbildung 13: Struktur der Oberfläche zum Einfügen und Bearbeiten von Lieferungen



Auf dieser Seite konnten alle lieferungsspezifischen Werte eingegeben werden. Für Attribute mit Freitext, beispielsweise „Qualität/Spezifikationen“, wurden einfache Textfelder implementiert. Für Attribute, die einen *foreign key* zu anderen Tabellen darstellen, wurden Auswahllisten implementiert, in denen der jeweilige Eintrag ausgewählt werden konnte, auf den verwiesen wurde. Für Datumsangaben wurden JavaScript-basierte Auswahlfelder gewählt, in denen die Datumsauswahl mittels eines einfachen Kalenders möglich ist. Nach dem Absenden des Formulars erfolgte eine Überprüfung der Eingaben auf beispielsweise leere Werte, falsche Datumsformate oder nicht erkennbare Formate der angegebenen Liefermenge. Wurde ein Fehler erkannt, erhält der Nutzer eine Fehlermeldung und konnte die Eingaben korrigieren. Wurde die angegebene Liefermenge nicht erkannt, beispielsweise aufgrund einer unbekannten Einheit, wurde ein neuer Bereich gezeigt, in dem der Nutzer eine, in der Datenbank gespeicherte, Einheit auswählen und eine in diese Einheit umgerechnete Menge angeben musste. Waren alle Prüfungen erfolgt, wurde der Eintrag in der Datenbank aktualisiert. Die Liefermenge wurde dabei in Gramm umgerechnet und als Restmenge eingetragen. War kein Gewicht, sondern beispielsweise ein Volumen gegeben, wurde als Restmenge automatisch 0 g festgelegt.

Auf der linken Seite wurde eine vertikale Navigationsleiste implementiert, die für alle Seiten der Nutzeroberfläche einheitlich gestaltet wurde. Ausgenommen waren die Oberfläche für die Anmeldung und die Übersichtsseite, die mit keiner bzw. horizontaler Navigationsleiste umgesetzt wurden.

Um Lieferungen einzutragen, mussten die zu referenzierenden Einträge für die Chemikalie, die Lagerungsvorschrift, den Hersteller und das Sicherheitsdatenblatt bereits in der Datenbank hinterlegt sein. Kategorien, Unterkategorien, Lagerungsvorschriften und Hersteller konnten unter „Ressourcen eintragen“ in der Datenbank hinterlegt werden. Für Sicherheitsdatenblätter wurde eine separate Seite eingefügt, die über den Menü-Eintrag „SDB hochladen“ erreichbar ist.

Chemikalien konnten unter dem Menü-Punkt „Neue Chemikalie definieren“ eingefügt werden. Die Oberfläche ist in Abbildung 14 gezeigt.

## Stoff eintragen

|   |              |  |
|---|--------------|--|
| <input type="text" value="Name"/>         | Name         | <input type="text" value="Ähnliche Stoffe"/>                             |
| <input type="text" value="CASNr"/>        | CASNr        | <input type="text" value="Bakterium"/> Wählen sie die Kategorie aus      |
| <input type="text" value="Summenformel"/> | Summenformel | <input type="text" value="Adsorbenz"/> Wählen sie die Unterkategorie aus |

*Angaben zur Gefahrstoffverordnung:*

|                      |         |                      |         |
|----------------------|---------|----------------------|---------|
| <input type="text"/> | R-Sätze | <input type="text"/> | S-Sätze |
| <input type="text"/> | H-Sätze | <input type="text"/> | P-Sätze |

☐ Gefahrstoff










|  |  |  |
|--|--|--|
|   |   |   |
| GHS01  | GHS02  | GHS03  |
|   |   |   |
| GHS04  | GHS05  | GHS06  |
|  |  |  |
| GHS07  | GHS08  | GHS09  |

Abbildung 14: Struktur der Oberfläche zum Eintragen neuer Chemikalien

Im oberen Teil des Formulars wurden Eingabefelder für die stoffspezifischen Attribute gegeben. Im unteren Teil konnten Angaben zur Gefahrstoffeinordnung getroffen werden. Dabei bestand die Möglichkeit, verschiedene Gefahrstoffsätze einzutragen und die zugehörigen Gefahrstoffsymbole auszuwählen. Ausgewählte Symbole wurden dabei mit dem in der Abbildung um GHS01 gezeigten blauen Rahmen markiert. Der Haken in der Checkbox „Gefahrstoff“ wurde automatisch ergänzt, sobald ein Gefahrstoffsatz eingetragen wurde, konnte aber manuell entfernt werden.

Während der Eingabe des Chemikaliennamens wurde automatisch in der Datenbank nach Chemikalien mit ähnlichen Namen gesucht, welche in dem Textfeld auf der rechten Seite angezeigt wurden. Der Name und die CAS-Nummer wurden nach dem Absenden des Formulars mit den Einträgen in der Datenbank verglichen, um doppelte Einträge zu verhindern. Nach der Prüfung wurde die Chemikalie in der Datenbank gespeichert. Die Gefahrenstoffsätze wurden dabei mit Trennzeichen ergänzt und, zu einer Zeichenkette zusammengesetzt, als „GHS-Sätze“ gespeichert. Die ausgewählten Gefahrenstoffsymbole wurden durch eine 1 repräsentiert, nicht

ausgewählte Symbole als 0. Diese Werte wurden, ebenfalls zusammengesetzt, als „GHS-Symbole“ gespeichert.

### 5.1.3 Rezepte definieren und verwalten

Es wurde in dem Webinterface eine Möglichkeit implementiert, häufig verwendete Stoffzusammensetzungen als „Rezept“ zu definieren. Diese Funktionalität sollte die Bedienung der Clientsoftware für die Waagen erleichtern, indem Rezepte anstelle der einzelnen Komponenten ausgewählt werden können und somit der Aufwand des Suchens verringert wurde. Weiterhin sollte der Nutzer durch die Anzeige der benötigten Menge weiter unterstützt werden. Die Rezepte beziehen sich daher ausschließlich auf die Komponenten, die eingewogen werden können. Benötigte Flüssigkeiten und Lösungen sollten in einem Kommentarbereich angegeben, allerdings nicht in die Liste der Komponenten eingetragen werden.

Das Eintragen der Rezepte sollte von beliebigen Arbeitsplätzen aus möglich sein und wurde daher in das Webinterface integriert. Weiterhin sollte über die Oberfläche das Entfernen von fehlerhaften oder nicht mehr benötigten Rezepten möglich sein. Ein Ausschnitt der Nutzeroberfläche ist in Abbildung 15 gezeigt. Die Oberfläche ist über den Menüpunkt „Rezepte“ erreichbar.

## Rezepte

### Neues Rezept definieren:

Neues Rezept mit  Einwaagen

### Auswahl filtern:

|                                       |                 |
|---------------------------------------|-----------------|
| Rezeptname                            | Rezeptname      |
| Chemikalienname                       | Chemikalienname |
| <input type="button" value="Suchen"/> |                 |



| Rezept         | Beschreibung                      | Zusammensetzung  |   |
|----------------|-----------------------------------|--|---|
| 4-Salze-Lösung | Lösung aus 4 Salzen in Wass...    | 19.557 g 4-Aminophenylphosphat<br>28.85 g Adenin Hemisulfat Salz<br>15.151 g Natriumphosphat Dihydrat<br>50 mg Iodonitrotetrazoliumchlorid |  |
| Generic Medium | Beispielhaftes Rezept für ein ... | 50 g D-Glucose<br>50 g Agarose Broad Range   |  |

Abbildung 15: Nutzeroberfläche für die Verwaltung von Rezepten im Webinterface

In dem unteren Teil der Seite wurde eine Tabelle mit allen eingetragenen Rezepten implementiert, die mithilfe der Filter darüber nach Rezeptnamen und Komponenten gefiltert werden können. Auf der rechten Seite der Tabelle wurden Schaltflächen angebracht, um die Rezepte, nach einer weiteren Bestätigung, zu löschen.

Mithilfe des Formulars im oberen Teil der Seite konnten weitere Rezepte eingetragen werden. Nach Eingabe der Anzahl der einzuwiegenden Komponenten wurde der Nutzer auf eine separate Seite weitergeleitet, auf der ein weiteres Formular für die Erstellung des Rezeptes bereitgestellt wurde. Das Formular für ein Rezept mit 5 Komponenten ist in Abbildung 16 gezeigt.

### Rezept Hinzufügen

|         |              |                |
|---------|--------------|----------------|
| Generic | Rezeptname   | Generic Medium |
|         | Beschreibung |                |

#### Auswahl der Einwaagen

|                        |   |        |    |   |
|------------------------|---|--------|----|---|
| 2, 3, 7, 8 TCDD        | ▼ | 88,78  | µg | ▼ |
| Ammoniumsulfat         | ▼ | 7,8    | g  | ▼ |
| Huminsäure Natriumsalz | ▼ | 123,44 | mg | ▼ |
| Canamycinsulfat        | ▼ | 25     | µg | ▼ |
| D-Sorbitol             | ▼ | 0      | g  | ▼ |
| Bestätigen             |   |        |    |   |

**Abbildung 16:** Formular für das Eintragen eines Rezeptes mit 5 Komponenten, teilweise gefüllt mit Beispielwerten

Im oberen Teil des Formulars wurden zwei Textfelder für den Namen und die Beschreibung des Rezeptes bereitgestellt. In einem weiteren Textfeld wurden während der Eingabe des Rezeptnamens Rezepte mit einem ähnlichen Namen angezeigt.

Im unteren Teil des Formulars wurden Kombinationen aus zwei Drop-Down-Listen und einem Textfeld bereitgestellt, mit der die Komponenten des Rezeptes festgelegt werden konnten. Über die Schaltfläche darunter konnten die Eingaben abgesendet werden. In diesem Formular wurde darauf geprüft, dass keine Komponenten doppelt verwendet wurden. Weiterhin wurde auf Komponenten ohne Mengenangabe geprüft. Diese leeren Komponenten wurden bei der Speicherung ignoriert.

#### 5.1.4 Administrativer Bereich und Erweiterungen

Der administrative Bereich des Webinterfaces wurde nicht im Nutzerbereich verlinkt. Der Bereich war durch die manuelle Eingabe der Adresse „Hostname/ChemieDB/Admin.php“ erreichbar. Besaß der angemeldete Nutzer keine Administrator-Rechte, wurde von dieser Seite automatisch auf die Übersichtsseite der Nutzeroberfläche weitergeleitet und eine Fehlermeldung ausgegeben.

In dem administrativen Bereich wurden die, unter 3.1.2 beschriebenen, Funktionalitäten gewährt:

- Export und Download der Datenbank
- Entfernen aller Datensätze
- Direkter Zugriff auf alle Datensätze über das Tool phpMyAdmin

Der Export der Datenbank erfolgte durch das OpenSource-Tool MySQL Dumper, das in die Oberfläche eingebunden wurde. Mit diesem Tool bestand die Möglichkeit, Sicherungskopien der Datenbank anzulegen, auf dem Server zu verwalten und lokal zu speichern. Mithilfe der freien Software phpMyAdmin konnte direkt auf die Datenbank zugegriffen werden. Dabei wurde dem Nutzer sowohl eine graphische Oberfläche für einfache Abläufe als auch eine SQL-Konsole für benutzerdefinierte Arbeitsschritte zur Verfügung gestellt.

Es wurden außerdem mehrere Werkzeuge programmiert, die den Übergang von einem vorliegenden Excel-Verzeichnis zur Datenbank erleichtern sollten. Ein Tool wurde für den erleichterten Import der Sicherheitsdatenblätter entwickelt, um schnell große Mengen Sicherheitsdatenblätter in der Datenbank zu speichern. Zuerst mussten die Datenblätter in einen festgelegten Ordner in dem Webserver kopiert werden, dann konnte über das Interface der Import gestartet werden. Über eine mit C# geschriebene Anwendung wurden alle Dateinamen auf unzulässige Zeichen wie Umlaute oder 'ß' überprüft und gegebenenfalls angepasst. Anschließend wurde mit PHP für jede Datei ein Link erstellt und in der Datenbank gespeichert.

Ein weiteres Tool erlaubt den Import der Datensätze aus dem Excel-Verzeichnis. Die mit C# geschriebene Anwendung griff über die von Microsoft zur Verfügung gestellte Interop-Schnittstelle auf das Verzeichnis zu und importierte alle Daten in einen Zwischenspeicher. Anschließend wurden zuerst die Kategorien, Unterkategorien, Lagerungsvorschriften und Hersteller in der Datenbank eingetragen, dann die Stoffe und abschließend die Lieferungen. Dadurch wurde sichergestellt, dass alle Datensätze, auf die verwiesen werden sollte, in der Datenbank vorhanden waren. Um das Tool nutzen zu können, musste das Chemikalienverzeichnis vollständig sein, es mussten alle obligatorischen Angaben eingetragen sein. Außerdem sollten Abweichungen wie „GmbH“ und „gmbh“ vermieden werden. Das Tool war über die Navigationsleiste im Administrator-Bereich erreichbar und startete automatisch den

Import der Sicherheitsdatenblätter. Einige Attribute wie „Restmenge“ oder „Lieferdatum“ wurden in dem Excel-Verzeichnis nicht erfasst. Während des Importes wurden dafür Standardwerte gesetzt, beispielsweise 0 g als Restmenge oder 01.01.1901 als Lieferdatum.

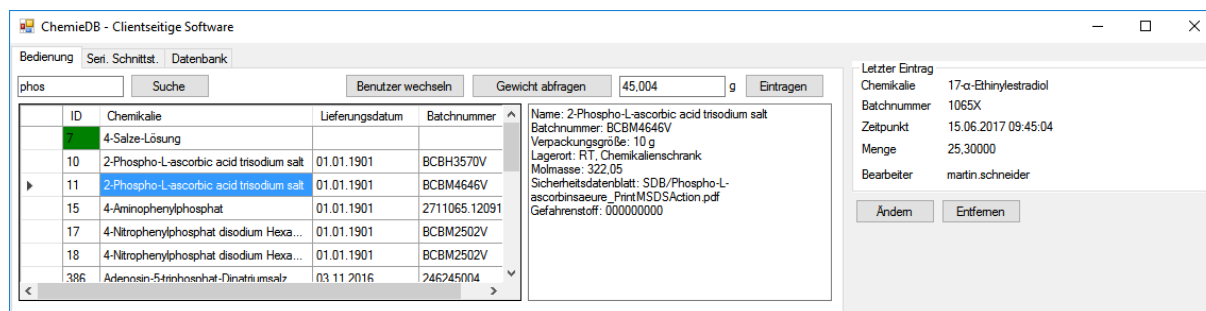
Über die normale Navigationsleiste im Nutzer-Bereich des Webinterfaces wurde außerdem eine Export-Funktion für den Inhalt des Chemikalienschrankes geboten. Dabei wurden alle Lieferungen mit einer bestimmten Lagerungsvorschrift in einer druckfertigen Excel-Datei gesammelt und zum Download angeboten. Diese Funktion sollte die Lagerhaltung direkt im Labor vereinfachen oder kann als Grundlage für eine Inventur der Chemikalienbestände genutzt werden.

## **5.2 Clientseitig**

### **5.2.1 Benutzeroberfläche**

Die clientseitige Software sollte eine Schnittstelle zwischen den Laborwaagen und der Datenbank darstellen. Die Bedienung sollte durch die Mitarbeiter des Labors parallel zum Vorgang des Einwiegens stattfinden und dementsprechend so einfach wie möglich gestaltet werden, um den Bearbeiter so wenig wie möglich zu belasten.

Bei Start des Programms wurde ein Anmeldefenster geöffnet, in dem der Bearbeiter die Zugangsdaten eingeben musste. Analog zu der Anmeldung im Webinterface wurde dabei eine Anfrage an einen Authentifikationsserver gesendet, um die eingegebenen Daten zu validieren. War die Anmeldung erfolgreich, wurde der Nutzer auf die Oberfläche der Software weitergeleitet. Wie unter 3.2.1 beschrieben, wurde die Oberfläche in 3 Bereiche gegliedert, die verschiedene Funktionalitäten bereitstellen. Dabei sollte der erste Bereich, der initial angezeigt wird, alle für Standardabläufe wichtigen Funktionen bieten, während alle Einstellungen in anderen Bereichen getroffen werden konnten. Die initiale Oberfläche ist in Abbildung 17 gezeigt.



**Abbildung 17: Initiale Oberfläche der clientseitigen Software**

Die Oberfläche war geteilt zwischen der Eingabemaske auf der linken Seite und der Anzeige des letzten Eintrages auf der rechten Seite. In der Eingabemaske wurde ein Tabellenbereich gegeben, in dem Lieferungen aus der Datenbank mit Batchnummer und Lieferdatum aufgelistet wurden. Mithilfe des Textfeldes darüber konnten die Einträge nach dem Chemikaliennamen gefiltert werden. Dabei wurden auch die eingetragenen Synonyme aus der Datenbank beachtet, um die Suche zu erleichtern. Wurde eine Lieferung aus der Tabelle ausgewählt, wurden die Details der Lieferungen aus der Datenbank abgerufen und in einem Textbereich rechts der Tabelle angezeigt. Die angezeigten Attribute waren: Name der Chemikalien, Batchnummer der Lieferung, Verpackungsgröße, Lagerort, Molmasse, Sicherheitsdatenblatt und Zahlencode der GHS-Symbole.

In der Tabelle standen ebenfalls Rezepte zur Auswahl, die entweder im Namen mit der Filter-Eingabe übereinstimmten oder Stoffe beinhalteten, die von dem Filter eingeschlossen wurden. Die Rezepte wurden mit einer grün hinterlegten ID gekennzeichnet und am Anfang der Tabelle gesammelt. Außerdem wurden für die Rezepte kein Lieferdatum und keine Batchnummer angezeigt. Wurde ein Rezept ausgewählt, wurden in dem rechten Textfeld der Name, die Beschreibung und die Liste der Komponenten des Rezeptes angezeigt. Um ein Rezept durchzuführen, musste doppelt auf die Zeile geklickt werden. Daraufhin wurden alle Lieferungen der Stoffe, die für das Rezept benötigt wurden, in der Tabelle angezeigt. In der ersten Zeile wurde weiterhin das Rezept angezeigt, um dem Bearbeiter einen schnellen Zugriff auf die benötigten Mengen zu bieten. Ein Ausschnitt der Tabelle und des Textfeldes für die Anzeige der Details eines beispielhaften Rezeptes ist in Abbildung 18 gezeigt.

|   | ID  | Chemikalie                  | Lieferungsdatum | Batchnummer     |  |
|---|-----|-----------------------------|-----------------|-----------------|--|
| ▶ | 7   | 4-Salze-Lösung              |                 |                 | 4-Salze-Lösung<br>Lösung aus 4 Salzen in Wasser. Für 400 ml Lösung |
|   | 15  | 4-Aminophenylphosphat       | 01.01.1901      | 2711065.120910A | 4-Aminophenylphosphat: 19,557 g                                    |
|   | 26  | Adenin Hemisulfat Salz      | 01.01.1901      | 071M1375V       | Adenin Hemisulfat Salz: 28,85 g                                    |
|   | 141 | Iodonitrotetrazoliumchlorid | 01.01.1901      | SLBK2252V       | Iodonitrotetrazoliumchlorid: 50 mg                                 |
|   | 207 | Natriumphosphat Dihydrat    | 01.01.1901      | 1199062         | Natriumphosphat Dihydrat: 15,151 g                                 |
| * |     |                             |                 |                 |  |
| < |     |                             |                 |                 | >  |

**Abbildung 18: Ausschnitt der graphischen Benutzeroberfläche nach Auswahl eines Rezeptes**

Über der Tabelle befand sich die Schaltfläche „Gewicht abfragen“, die das aktuelle Gewicht auf der angeschlossenen Waage abfragte und in dem Textfeld rechts davon anzeigte (vgl. Abbildung 17). Alternativ konnte das aktuelle Gewicht erfasst werden, indem auf der Waage die „Print“-Taste gedrückt wurde. Dieser Vorgang startete automatisch die Speicherung des aktuellen Gewichtes als Verbrauch in der Datenbank ein. Die Schaltfläche „Eintragen“ wurde für die Speicherung von manuell eingetragenen oder korrigierten Gewichten konzipiert und startet ebenfalls die Speicherung des Verbrauchs. In beiden Speichervorgängen wurde zuerst auf ein gültiges Zahlenformat des Gewichtes und auf die Auswahl einer gültigen Chemikalienlieferung geprüft. Anschließend wurden die verbrauchte Menge, die ID der Lieferung und der angemeldete Bearbeiter in der Datenbank in der Tabelle „Verbrauch“ gespeichert und die Restmenge der Lieferung um die eingetragene Menge vermindert.

Über eine weitere Schaltfläche mit dem Titel „Benutzer wechseln“ konnte der Bearbeiter gewechselt werden. Dabei wurde erneut das Anmeldefenster gezeigt und nach Verifizierung der Anmeldedaten geschlossen. Wurde das Authentifizierungsfenster während des Vorgangs manuell geschlossen, blieb der aktuelle Bearbeiter angemeldet. Diese Funktionalität sollte den schnellen Wechsel zwischen den Labormitarbeitern ermöglichen, indem die Software für die Ummeldung nicht geschlossen werden musste.

Auf der rechten Seite der Benutzeroberfläche wurde ein separater Bereich für die Anzeige des letzten Verbrauches integriert (vgl. Abbildung 17). In diesem Feld wurden die Werte des letzten eingetragenen Verbrauchs angezeigt. Über die Schaltfläche „Entfernen“ konnte der Eintrag aus der Datenbank gelöscht werden. Dabei wurde die Restmenge der eingetragenen Lieferung um die gespeicherte Verbrauchsmenge erhöht. Über die Schaltfläche „Ändern“ konnte der gezeigte



Verbrauch geändert werden. Nach dem Klick auf die Schaltfläche wurde die gespeicherte Lieferung in der Tabelle ausgewählt und die verbrauchte Menge in dem Textfeld für die Gewichtsangabe eingetragen. Der Nutzer hatte nun die Möglichkeit, Änderungen an der Verbrauchsmenge vorzunehmen oder eine andere Lieferung auszuwählen, bevor die Änderungen mit einem zweiten Klick auf die Schaltfläche gespeichert wurden. Für die Speicherung wurde zuerst der fehlerhafte Eintrag in der Verbrauchstabelle gelöscht und die Restmenge der gespeicherten Lieferung um die Verbrauchsmenge erhöht. Anschließend wurde ein neuer Eintrag in die Tabelle „Verbrauch“ mit den angepassten Werten gespeichert und die Restmenge der ausgewählten Lieferung um die entsprechende Menge vermindert.

### 5.2.2 Serielle Schnittstelle

In dem Reiter „Serielle Schnittstelle“ bestand die Möglichkeit, die Parameter der seriellen Übertragung zu ändern oder zwischen verschiedenen seriellen Anschlüssen zu wechseln. Ein Überblick der bereitgestellten Nutzeroberfläche ist in Abbildung 19 gegeben.

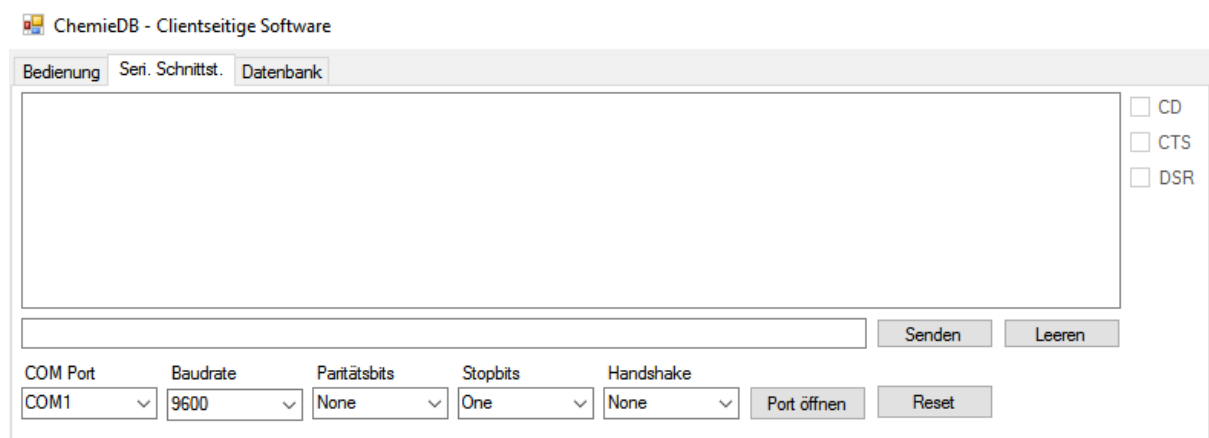


Abbildung 19: Oberfläche zur Konfiguration der seriellen Schnittstelle in der clientseitigen Software

Die Konfiguration erfolgte benutzerfreundlich über die Auswahl von Optionen aus Drop-Down-Listen. Es konnten die genutzte Schnittstelle am Rechner, die verwendete Baudrate sowie die Parameter Paritätsbits, Stoppbits und Handshakeprotokoll verändert werden. Über die Schaltfläche „Reset“ konnten Änderungen der Einstellungen rückgängig gemacht werden. Mithilfe der Schaltfläche „Port öffnen“ konnte die Verbindung zu der Waage hergestellt oder geschlossen werden. Beim Herstellen der Verbindung wurden weiterhin alle Einstellungen der

seriellen Schnittstelle in einer Datei gespeichert und bei Neustart der Software aus der Datei geladen.

In dem oberen Teil des Reiters wurde ein serielles Terminal zur Verfügung gestellt, mit dem einfache Befehle an die Waage übertragen und die Antwort dargestellt werden konnte. So konnten beispielsweise aktuelle Wägewerte abgerufen oder Tara-Funktionen aufgerufen werden. Weiterhin konnte der Auto-Print-Modus aktiviert werden, bei dem die Waage nach Stabilisierung des Gewichtes automatisch die aktuellen Werte an die Software sendete.

### 5.2.3 Datenbankverbindung

Für die Verbindung zu dem MySQL-Server wurde die OpenSource-Software „Connector/Net“ von Oracle Corp. verwendet. In dem Reiter „Datenbank“ wurde die Möglichkeit zur Konfiguration der Verbindung mit der Chemikaliendatenbank geboten. Parameter der Verbindung waren der Hostname, der genutzte Port der MySQL-Datenbank, der in der Datenbank registrierte Nutzernamen und das entsprechende Passwort. Die Oberfläche in dem Reiter ist in Abbildung 20 gezeigt

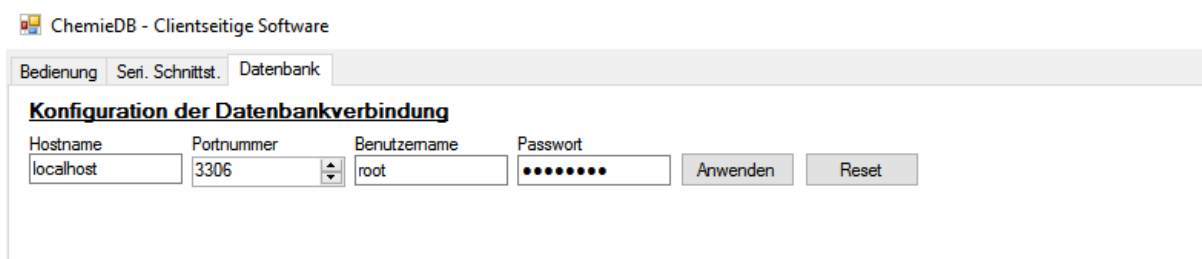


Abbildung 20: Oberfläche zur Konfiguration der Datenbankverbindung in der clientseitigen Software

Der Inhalt des Passwort-Feldes wurde automatisch mit Punkten ersetzt, um die Sicherheit der Eingaben zu gewährleisten. Nach der Bestätigung mit der „Anwenden“-Schaltfläche wurde mit den aktuellen Einstellungen versucht, eine Verbindung zu einer Datenbank aufzubauen. Wurde die Verbindung bestätigt, wurden alle Einstellungen in einer Datei gespeichert und bei jedem weiteren Systemstart automatisch wieder verwendet. Das eingegebene Passwort wurde dabei verschlüsselt gespeichert, um das Auslesen aus der Datei zu verhindern. Für die Verschlüsselung wurde der symmetrische Blockverschlüsselungsalgorithmus *Blowfish* verwendet. Die verwendete C#-Implementation stammte aus dem OpenSource-Projekt „Blowfish Block Cipher C#“ von Defuse.ca. Konnte die

Verbindung nicht hergestellt werden, konnte mit der „Reset“-Schaltfläche die letzte funktionierende Konfiguration wiederhergestellt werden.

Die in der Oberfläche erforderlichen Anmeldedaten entsprechen dabei nicht den Anmeldedaten für die Software oder das Webinterface, sondern sind von dem MySQL-Server verwaltete Nutzer und für die Rechteverwaltung bei Zugriffen auf die Datenbanken relevant. So können in dem MySQL-Server Nutzer festgelegt werden, die ausschließlich Daten lesen können, ohne Änderungen durchführen zu können. In diesem Fall konnte die Client-Software die Verbindung herstellen, konnte allerdings keinen Verbrauch in der Datenbank speichern, löschen oder ändern. Es sollte für die Clientsoftware stets ein Nutzer angelegt werden mit Rechten zur Datenänderung, aber ohne Rechte zur Änderung der Datenbankstruktur.

## 6 Produktionsumgebung

### 6.1 Serverseitig

#### 6.1.1 Einsatz

Die Software wurde zum Zeitpunkt dieser Arbeit in dem beschriebenen Funktionsumfang in der QuoData GmbH unter realen Bedingungen verwendet. Die Chemikaliendatenbank lief dabei auf der virtuellen Maschine *intern01* mit einem Softwarestack aus Windows 7 Professional und dem XAMPP-Paket mit einem Apache 2.4 Webserver und einer MySQL-Datenbank. Die virtuelle Maschine lief auf einem HyperV-Host mit Windows Server 2012 Standard, der mit dem lokalen Netzwerk der QuoData GmbH verbunden und nicht über das Internet erreichbar war. Für den Zugriff auf die Datenbank durch die clientseitige Software der Chemikaliendatenbank wurde ein neuer Benutzer angelegt, der ausschließlich auf die benötigte Datenbank zugreifen konnte und keine Rechte für die Veränderung ihrer Struktur besaß. Der Zugang zu dem Webinterface und der Client-Software wurde über den firmeninternen LDAP-Server realisiert. Jeder Mitarbeiter der „Labor“-Gruppe wurde für den Zugang autorisiert, Administratoren wurden außerdem der „ChemieDB-Admin“-Gruppe zugewiesen. Der direkte Zugriff auf die virtuelle Maschine war den Systemadministratoren der QuoData GmbH vorbehalten.

Um die Sicherheit der gespeicherten Daten zu gewährleisten, lief auf dem Host der virtuellen Maschine ein PowerShell-Script, das mithilfe des *Windows Volume Shadow Copy Service* kontinuierliche Backups der gesamten virtuellen Maschine auf einer externen Festplatte sicherte.

Die Chemikaliendatenbank wurde von den Mitarbeitern des Labors der QuoData zum Zeitpunkt dieser Arbeit bereits in den normalen Arbeitsfluss aufgenommen. Das bestehende Excel-Chemikalienverzeichnis wurde in die Datenbank importiert. Die Chemikalienverwaltung mit dem Webinterface wurde bereits ausführlich getestet und aufgetretene Fehler schnellstmöglich behoben. Für die Nutzung des Webinterfaces wurden die beiden Browser Google Chrome und Mozilla Firefox verwendet. Alle vorgestellten Abläufe, Anzeige- und Auswahlmodi wurden mit den Labormitarbeitern besprochen und an deren Vorschläge angepasst, um die Bedienung so intuitiv wie möglich zu gestalten. Die Anzeige von Restmengen, Lieferdaten und

Gefahrstoffsymbolen erfolgte nicht für alle Einträge, da für die importierten Einträge aus dem Excel-Verzeichnis keine Werte vorhanden waren. Die für den Import und Export von Datensätzen entwickelten Erweiterungen wurden ebenfalls getestet und funktionierten fehlerfrei.

### **6.1.2 Ausblick**

Um das Potential der gespeicherten Daten weiter zu nutzen, wurde die Einrichtung eines Systems geplant, dass den Lieferungen Warnwerte zuweist, bei deren Unterschreitung der Nutzer eine Benachrichtigung erhält. Auf diese Weise könnten Verzögerungen durch Lieferzeiten effektiv verringert werden. Weiterhin sollte bei der Anzeige von Rezepten gewarnt werden, falls die Komponenten nicht mehr in der benötigten Menge auf Lager sind. Optional sollte diese Prüfung auch bei der Anmeldung eines Nutzers für alle eingetragenen Rezepte durchgeführt werden. Außerdem ist es möglich, anhand der Verbrauchswerte Statistiken zu dem monatlichen Chemikalienbedarf zu erstellen und für eine effizientere Lagerhaltung zu verwenden. Diese Features konnten aus Zeitgründen nicht umgesetzt werden und sollen im Anschluss an diese Arbeit implementiert werden.

## **6.2 Clientseitig**

### **6.2.1 Einsatz**

Die Software für die Anbindung der Waagen lief auf einem Rechner mit Windows 10 und dem .NET-Framework 4.6. Die Grundfunktionen der Software, beispielsweise das Eintragen oder Ändern von Verbrauchseinträgen oder die Suche nach bestimmten Chemikalienlieferungen, konnten bereits unter realitätsnahen Bedingungen getestet werden. Der Aufbau der seriellen Verbindung mit einem Kabel führte allerdings zu einer Störung der Arbeitsplätze um die Waage herum und konnte nicht langfristig verwendet werden.

### 6.2.2 Ausblick

Im Anschluss an diese Arbeit sollten die folgenden Features in die clientseitige Software implementiert werden:

- Unterstützung weiterer Waagenmodelle, beispielsweise KERN-EG2200
- Einfache Funktion zur Einrichtung anderer Waagen
- Drahtlose Kommunikation mit den Waagen
- Änderung beliebiger Verbrauchseinträge
- Vereinfachte Auswahl der eingewogenen Chemikalienlieferung

Für die Unterstützung weiterer Waagenmodelle soll weiterhin eine zusätzliche Tabelle in der Datenbank angelegt werden, in der die Parameter für die serielle Verbindung sowie wichtige Befehle und Datenformate abgelegt werden sollen. Damit könnten die Konfigurationen stets synchronisiert werden und eine erneute Einrichtung an anderen Arbeitsplätzen entfielen.

Die drahtlose Kommunikation mit den Waagen könnte mithilfe eines RS-232-Bluetooth-Adapters umgesetzt werden. Diese Adapter, beispielsweise der Bluetooth\*<sup>1</sup>-Adapter von LinTech GmbH, werden an die COM-Ports der Waage und des Rechners angeschlossen und übertragen die jeweiligen Signale untereinander über Bluetooth [LINTECH, 2008]. Die Software sollte weiterhin ausgebaut werden, um die gleichzeitige Verbindung zu mehreren Waagen zu unterstützen.

Die manuelle Auswahl der Chemikalienlieferung vor dem Einwiegen wurde als der zeitaufwändigste Arbeitsschritt bei der Bedienung der clientseitigen Software erkannt. Als alternativer Ansatz zur manuellen Eingabe war eine Erkennung über eine angeschlossene Kamera und ein Barcode-Scan oder die Erkennung eines QR-Codes auf der Verpackung der Chemikalie. Dafür müsste die ID jeder Lieferung aus der Datenbank entsprechend codiert auf der Verpackung angebracht werden. Über die Kamera könnte ein Bild aufgenommen und mithilfe der .NET Barcode Reader Library beziehungsweise OpenSource-Projekten wie dem „QR-Coder“ von R. Herrman ausgewertet werden [CODEBUDE, 2017]. Nach Erkennung der ID könnte die Einwaage sofort erfolgen, ohne manuell auf die Software am Rechner zuzugreifen. Eine weitere Möglichkeit wäre die Steuerung der Software über eine Spracherkennung, beispielsweise mit der Microsoft Speech-Plattform [MICROSOFT, 2017]. Dabei ist allerdings zu beachten, dass viele Systeme zur

Spracherkennung Probleme bei einem Wechsel der Sprache haben, wie sie bei den wechselnden deutschen und englischen Bezeichnungen unvermeidbar sind. Mit der Umsetzung einer der beiden Ansätze für die Chemikalienerkennung könnte perspektivisch ein Betrieb ohne manuelle Bedienung der Software realisiert werden, wodurch die Rechner nicht mehr mit Eingabegeräten oder Monitoren ausgestattet werden müssten.

## Zusammenfassung

Die Lagerhaltung von Chemikalien ist ein wichtiger Faktor für die Leistungsfähigkeit von Laboren. Wartezeiten durch fehlende Chemikalien verlangsamen Forschungsprozesse, während durch fehlende Übersicht andere Chemikalien verfallen. Aus diesem Grund wurde von mehreren Anbietern spezialisierte Software entwickelt, die die Verwaltung von Chemikalien innerhalb einer Firma erleichtern sollten. Das Angebot reicht von Freeware-Webanwendungen wie Quartzzy bis hin zu hoch vernetzten Anwendungen mit spezieller Hardware wie der Syngo Lab Inventory Manager von Siemens.

Mithilfe dieser kommerziellen Software können gelieferte Chemikalien in einer Datenbank gespeichert, ein Lagerort und eine Liefermenge angegeben und verbrauchte Batches wieder entfernt werden. Allerdings konnte zum Zeitpunkt dieser Arbeit keine Software gefunden werden, die den aktuellen Restbestand von teilweise verbrauchten Chemikalienlieferungen erfasste. Mittels dieser Information könnte der Nachkauf von Chemikalien optimiert und Verzögerungen durch unzureichende Restbestände minimiert werden. Weiterhin könnte die Datenbank als Grundlage für eine Gefährdungsbeurteilung nach §6 Abs. 12 GefStoffV dienen oder für automatische Chemikalieninventuren genutzt werden. Deshalb sollte im Rahmen dieser Arbeit eine Datenbank für die Verwaltung von Chemikalien und ein Modul für die Echtzeiterfassung von Auswaagen entwickelt und für die firmeninterne Nutzung angepasst werden. Die Grundlage für die Datenbank sollte das bestehende Chemikalienverzeichnis darstellen, ein Tabellenblatt mit den bisher relevantesten Attributen aller Chemikalienlieferungen. Es wurde eine MySQL-Datenbank realisiert und eine einfache webbasierte Benutzeroberfläche für die Verwaltung der Datenbank geschrieben. Die Datenbank wurde auf einem lokalen Webserver abgelegt und für alle Rechner im Intranet erreichbar gemacht. Weiterhin wurde eine lokale Anwendung in C# geschrieben, die über eine serielle Verbindung mit einer Laborwaage die aktuellen Einwaagen abrufen und in der Datenbank speichern konnte. Ziel war es, ein verteiltes System aus einer zentralen Datenbank mit Webinterface und mehreren Client-Rechnern mit Anbindung an Laborwaagen zu ermöglichen.



In dieser Arbeit konnte ein *Proof of Concept* für das Projekt erbracht werden. Es wurde gezeigt, dass die Echtzeiterfassung und Speicherung von Einwaagen der Laborwaage KERN ABJ120-4M möglich ist und daraus Restmengen des Chemikalienbestandes ermittelt werden können. Die Verwaltung der Chemikalien mithilfe der Datenbank wurde unter realitätsnahen Bedingungen getestet und bereits in den Arbeitsfluss der Laboranten der QuoData GmbH integriert. Das bestehende Tabellenverzeichnis der Chemikalienlieferungen konnte mithilfe eines Import-Moduls in die Datenbank übernommen werden. Es wurden weiterhin Funktionen für den Export der Datenbank und regelmäßige Sicherungskopien des Webservers implementiert, um die Sicherheit der Daten zu gewährleisten.

Bei der Entwicklung der lokalen Anwendung für die Verbindung zu den Waagen konnte aufgrund der unterschiedlichen Kommunikationsprotokolle verschiedener Waagenmodelle keine allgemeingültige Schnittstellenkonfiguration festgelegt werden. Dementsprechend war die entwickelte Software ausschließlich mit der Laborwaage KERN ABJ120-4M kompatibel. Mit dieser Waage konnten das Auslesen der Einwaagen und das Eintragen in die Datenbank erfolgreich getestet werden. Perspektivisch sollten die Kommunikationsparameter verschiedener Waagenmodelle ebenfalls in der Datenbank gespeichert werden, um einen schnellen Wechsel der Modelle zu ermöglichen.

## Literaturverzeichnis

- 1 QUARTZY (2015): QUARTZY Teaching Presentation, 2015
- 2 LAKE, L. et al. (2013): Concise Guide to Databases, Springer Verlag, London, 2013
- 3 HOPPE-KIAUK, F. (2004): Entwicklung von Datenbank Anwendungen, 2004
- 4 BOUGUETTAYA, A. et al. (2014): Advanced Web Services, Springer Verlag, New York, 2014
- 5 GILMORE, W. (2002): A Programmers Introduction to PHP 4.0, Apress Verlag, 2002
- 6 TROELSEN, A. et al. (2015): C# 6.0 and the .NET 4.6 Framework, 7. Auflage, Apress Verlag, 2015
- 7 Rahman, M (2014): C# Deconstructed, Apress Verlag, 2014
- 8 PEACOCK, C. (1998): Interfacing the Serial/RS-232 Port, Craig Peacock, 1998
- 9 WILD, T. (2002): Best Practice in Inventory Management, 2. Auflage, Elsevier Science Ltd., 2002
- 10 PASZKO, C. et al. (2002): Laboratory Information Management Systems, 2. Auflage, Marcel Dekker Inc., 2002
- 11 CHANG, F. et al. (2006): Bigtable: A Distributed Storage System for Structured Data, Google Inc., 2006
- 12 KERN (2017): Bedienungsanleitung Kern & Sohn Analysenwaage ABJ 120-4M, Kern & Sohn, 2017
- 13 SIEMENS (2017): The Benefits of Automating Laboratory Inventory Management, Siemens Healthcare, 2017
- 14 SEIDLER, K. et al. (2006): Das XAMPP-Handbuch, Addison-Wesley-Verlag, München, 2006
- 15 W3TECHS (2017): Usage of server-side programming languages for websites.  
[https://w3techs.com/technologies/overview/programming\\_language/all](https://w3techs.com/technologies/overview/programming_language/all)  
Stand 15.06.2017
- 16 LINTECH (2008): Bluetooth\*RS232-Adapter, LinTech GmbH, 2008.  
[http://www.lintech.de/download/downloads/bluetooth-rs232-datenadapter-class\\_1/Handbuch-RS232-1407-dt.pdf](http://www.lintech.de/download/downloads/bluetooth-rs232-datenadapter-class_1/Handbuch-RS232-1407-dt.pdf)

- 17 CODEBUDE (2017): QRCode – eine OpenSource QR-Code Implementierung in C#, CodeBude.net, 2017  
<https://code-bude.net/2013/10/17/qrcoder-eine-open-source-qr-code-implementierung-in-csharp/>  
Stand 01.07.2017
- 18 MICROSOFT (2017): Microsoft Speech Plattform, Microsoft Developer Network, 2017  
[https://msdn.microsoft.com/en-us/library/office/hh361572\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/office/hh361572(v=office.14).aspx)  
Stand 05.07.2017
- 19 KERN (2010): Bedienungsanleitung Kern & Sohn Präzisionswaage EG 2200-2NM, Kern & Sohn, 2010
- 20 KERN (2016): Waagen & Prüfservice, Kern & Sohn, 2016
- 21 HULZEBOSCH, J. (2008): USB in der Elektronik, Franzis Verlag, Poing, 2008
- 22 DIN 66303:2000-06, Informationstechnik: 8-Bit-Code, 2000
- 23 FRENZEL, L. (2016): Handbook of Serial Communications Interfaces, Elsevier Inc., Oxford, 2016
- 24 DALLAS (1998): Application Note 83, Fundamentals of RS-232 Serial Communications, Dallas Semiconductors, 1998
- 25 TEXAS INSTRUMENTS (2002): Interface Circuits for TIA/EIA-232-F (Rev. A), Texas Instruments Inc., 2002

### E r k l ä r u n g

gemäß § 19 (Abs. 1) der Prüfungsordnung für den  
Studienganges Labor- und Verfahrenstechnik vom 01. Oktober 2008.

Ich habe die vorliegende Bachelor selbst-  
ständig verfasst und keine anderen als die an-  
gegebenen Quellen und Hilfsmittel benutzt.

.....

(Ort)

.....

(Datum)

(Unterschrift)