1. Einleitung

Motivation, Zielstellung, Vorteile

2 Theoretische Grundlagen

2.1 Serielle Kommunikation/COM-Schnittstelle

2.2 RS232

2.3 Datenbanken – SQL -> NoSQL als Alternative erwähnen

2.4 Datenstacks/Datenfluss -> C# vs. PHP

3 Konzeption

3.1 Serverseitig

3.1.1 Datenbankstruktur

3.1.2 Webinterface

3.1.3 Zugangsbeschränkung und Rechteverwaltung

3.2 Clientseitig

3.2.1 Benutzeroberfläche

3.2.2 Serielle Schnittstelle zur Waage

3.2.3 Datenbankschnittstelle

4 Implementierung

4.1 Serverseitig

4.1.1 Datenbankstruktur

4.1.2 Nutzeroberfläche

4.1.3 Zugangsbeschränkung und Rechteverwaltung

4.1.4 Erweiterungen

4.2 Clientseitig

4.2.1 Interface

4.2.2 Serielle Schnittstelle zur Waage

4.2.3 Datenbankschnittstelle

4.2.4 Erweiterungen

5 Produktion

5.1 Testprozeduren

5.2 Datensicherheit

6 Zusammenfassung

2. Zielstellung: Live-Tracking zur Erkennung von Engpässen, statistische Auswertung vom Verbrauch,

Allg. Hinweise:

Genauere Überschriften

Produktion: Verlauf der Entwicklung aus Entwicklungsumgebung -> Testumgebung -> Produktionsumgebung; Überwachung und Backups

# 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 immer noch 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 Wert der aktuellen Chemikalienbestände mit diesen Methoden zu erfassen oder Einträge nach mehreren Kriterien filtern zu lassen.

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 wäre beispielsweise das webbasierte Quartzy, das eine einfache Verwaltung von Chemikalien und eine zentrale Verwaltung von Bestellanträgen bietet [Quartzy, 2014]. 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, 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. [Hm…]

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 der Marke Kern sind Anschlüssen für die serielle Datenübertragung ausgestattet, über die die Waage gesteuert und Messwerte ausgegeben werden können [KERN Waagenübersicht]. 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 [ABJ und EG Datenblätter].

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 mitführen kann. Die Datenbank sollte dabei auf einem internen Server abgelegt werden und über ein Webinterface erreichbar gemacht werden. Die Anforderungen an das Interface waren:

* Intuitive, benutzerfreundliche Struktur
* Passwortschutz
* Bereitstellung von Grundfunktionen:
  + Chemikalien einfügen
  + Einträge bearbeiten
  + Einträge 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.

RFID

# 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, J.]

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; HULZEBOSCH, J.]

Die Bytes können entweder seriell oder parallel übertragen werden. Bei der parallelen Übertragung werden alle Bits eines Bytes gleichzeitig übertragen. Voraussetzung hierfür ist eine Verbindung mit mindestens einem Kabel pro übertragenem Bit und weiteren Kabeln für Steuersignale. Weiterhin müssen die Kabel mit zunehmender Übertragungsgeschwindigkeit immer höheren Qualitätsanforderungen entsprechen, um die Datenintegrität sicherzustellen. So müssen beispielsweise die Impendanzen 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 fest verbaute Anzahl von Kabeln ist auch die Übertragung größerer Datenpakete, zum Beispiel von zusätzlichen 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.

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 in 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.

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.



Abbildung 1: Spannungsverhalten von bipolar NRZ-codierten binären Werten

Die Verwendung von bipolaren Codierungen erbringt den Vorteil, dass sich keine mittlere Gleichspannung in der Leitung aufbaut, wie es bei der Verwendung von unipolaren Codierungen, durch den schnellen Wechsel von mehreren positiven Spannungen, der Fall ist. Diese Störspannung 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. [Appl. Note 83]

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 geringerer Geschwindigkeit verwendet. [Frenzel, Handbook]

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. Das Prinzip beider Verbindungsarten sind in Abbildung 2 dargestellt.



Abbildung 2: Prinzip der asymmetrischen (a) und symmetrischen (b) Verbindung als Anwendung in der Tontechnik [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 Ausgangssignal auf. 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, Handbook]

NRZ, DIN, RZ

## 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“ [Appl. Note 83].

Der Standard wurde 1962 erstmalig 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 vor allem 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 Instr. Appl. Note]

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 Instr. Appl. Note]



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 sind Signale für die Koordination der Datenübertragung, um Datenverlust durch blockierte Leitungen zu vermieden. Das Protokoll, nach dem diese Signale den Datenfluss steuern, wird als Handshake bezeichnet. 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 Instr. Appl. Note]

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 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 Instr. Appl. Note]



Abbildung 4: Rahmenbildung nach RS-232 mit 8 Datenbits und einem Stopbit [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 Stopbits. 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 Stopbits, 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 Handbook, Texas Instr. Appl. Note]

DCE, DTE, EIA, RS, PC, DTR, DSR, RTS, CTS, RI

## 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 Operationen Erstellen (Create), Lesen (Read), Ändern (Update) und Löschen (Delete). [Lake Guide, Datenbankgrundlagen]

Datenbanken bestehen aus der Menge zu verwaltender Daten, die die eigentliche Datenbank darstellt, und dem Datenbankmanagementsystem (DBMS), das die Strukturierung und Verarbeitung der Daten realisiert. Um Zugriffe auf die Datenbank zu ermöglichen, bietet das DBMS eine Datenbanksprache an. [Lake Guide]

Anhand der vom DBMS vorgegebenen Strukturierung der Daten wird in verschiedene Datenbankmodelle eingeteilt, die die Beziehungen der gespeicherten Objekte zueinander angibt. Die wichtigsten Modelle sind:

• Hierarchisch

• Netzwerkartig

• Dokumentorientiert

• Objektorientiert

• Relational

Relationale Datenbanken wurden erstmalig 1970 von E. Codd beschrieben und bestehen aus mehreren Tabellen, in denen die Daten zeilenweise gespeichert werden. Die Spalten verschiedener Tabellen können als Verknüpfung zwischen den Tabellen verwendet werden. Die Struktur und Verknüpfungen der Tabellen werden ebenfalls in separaten Tabellen gespeichert. Der Aufbau einer Tabelle ist in Abbildung 1 anhand einer fiktiven Tabelle dargestellt. [Lake Guide]



Abbildung 5: Beispielhafter Aufbau einer Tabelle als Teil einer relationalen Datenbank

Jede Tabelle besteht aus mehreren Attributen, die in Spalten dargestellt sind. Jede Zeile ist ein Datensatz oder „Tupel“, in dem ein Wert für jedes Attribut der Tabelle 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 Guide]

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, die 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 Guide]

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, 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. Um eine Normalform zu erreichen, müssen alle vorhergehenden Normalformen umgesetzt sein. [Datenbankgrundlagen]

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. [datenbankgrundlagen]

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. [Datenbankgrundlagen]

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. [8]

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 ACID-Kriterien:

* 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 skalieren relationale Managementsysteme aufgrund des technischen Aufwandes für die Einhaltung der ACID-Kriterien schlecht mit großen Datenmengen, wie sie bei großen Konzernen wie Facebook oder Banken anfallen. Die von Google Inc. entwickelte NoSQL-Datenbank BigTable hingegen ist für die Speicherung und Manipulation von Daten im Bereich von Petabytes (1015 Bytes) ausgelegt und bildet die Grundlage für verschiedene Services wie Google Maps oder Google Earth. Öffentlich verfügbare Systeme sind beispielsweise MongoDB oder Apache Cassandra und werden von bekannten Plattformen wie Twitter, Ebay oder GitHub verwendet.

DB, DBMS, SQL, ACID, NoSQL, NF

## 2.4 Softwarestacks

Als Softwarestacks, oder Softawarestapel, 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 zu garantieren.

### 2.4.1 Webservices

Software Services sind essentielle Bestandteile für die Entwicklung von verteilten Anwendungen. Es sind Module mit beschränktem Funktionsumfang, die mit einem selbsterklärenden Interface in Sofwarepakete 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. [Dolog advanced Web services]



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 Webservers 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 Zusammensatzung 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 programmers intro to PHP]

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 um eine HTML-Datei, werden alle weiteren nötigen Inhalte, wie Bilder oder Inhalte im PDF-Format, ebenfalls aus dem Dateisystem geladen. Sind in der Datei Skripte enthalten, wird der Script-Interpreter aufgerufen, um die Skripte auszuführen. In den meisten Fällen handelt es sich dabei um PHP oder Perl-Skripte. 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)

Der Client sendet, beispielsweise mit einem Browser, eine Anfrage mit einer Adresse an den Server. Der, auf dem Server installierte, 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 programmers intro to PHP]

### 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 Windows beschränkt. Als Resultat der zahlreichen Portierungen, beispielsweise das Mono .NET-Framework für Linux oder MacOS, kann C# inzwischen auch plattformübergreifend verwendet werden. [Throelsen C# 6.0]

Um Software auf einem Rechner auszuführen, muss zuerst der programmierte Code in maschinenlesbare Sprache umgeschrieben werden. Dieser Schritt wird Compilieren 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 die Common Intermediate Language (CIL) eingeführt. Alle .NET-Sprachen bieten einen Compiler, der den Programmcode 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 universell auf jedem System ausführbar, das eine Form der .NET-Plattform installiert hat. [Rahman C# Deconstructed]

# 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:

* Lieferung geöffnet
* Restmenge in g

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

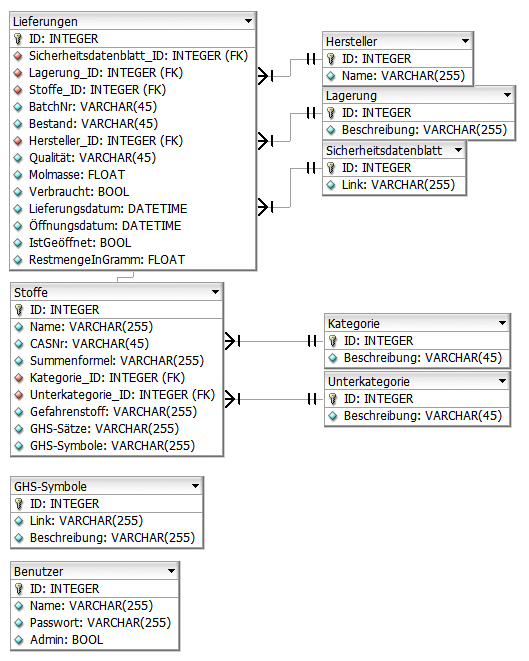


Abbildung 9: Konzeptionelle Datenbankstruktur mit Attributen und zugehörigen Datentypen

Die Attribute wurden anhand ihrer Abhängigkeiten nach in Tabellen eingeteilt. Jeder Block repräsentiert eine Tabelle, die Zeilen in dem Block die zugehörigen Attribute. Nach den Namen der Attribute sind die jeweiligen 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 Gefahrenstoffsätze wurden als eine vollständige Zeichenkette zusammengefasst. Die Gefahrenstoffsymbole 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 Gefahrenstoffsymbole nach dem Global harmonisierten System zur Einstufung und Kennzeichnung von Chemikalien (GHS) wurden als Bilddateien in der Ordnerstruktur des Webservers abgelegt. Die Sicherheitsdatenblätter wurden als PDF-Dateien ebenfalls in der Ordnerstruktur hinterlegt. Bei jeder Eintragung eines neuen Sicherheitsdatenblattes sollte das neue SDB ebenfalls in den Ordner gespeichert werden.

### 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.)
* Manuelles Eintragen

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 mit PHP 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. 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 Herstellern 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
* Ausführung beliebiger SQL-Befehle

### 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

### 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 Features bieten:

* Durchsuchen der Datenbank nach Chemikalienlieferungen
* Automatisches Erfassen und Anzeigen von Wägewerten
* Manuelle Korrektur der Messwerte
* Konfiguration der Datenbankverbindung
* Konfiguration der seriellen Schnittstelle
* Überwachung der seriellen Kommunikation
* Serielles Terminal für die direkte Übertragung von Befehlen an die Waage