

Praxisprojekt 4 (4. Semester)

an der

Berufssakademie Sachsen
Staatliche Studienakademie Riesa
Studiengang Labor- und Verfahrenstechnik

Kurs: RT14LV1

Studienrichtung: Biotechnologie

Thema: Konzeption und Entwicklung einer Browser-basierten Datenbank für das
firmeninterne Chemikalienmanagement

Eingereicht von:

Martin Schneider

Firma:

QuoData
Gesellschaft für
Qualitätsmanagement und Statistik mbH

Prellerstraße 14
01187 Dresden

Betrieblicher Betreuer: M. Sc. Martin Jähne

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Tabellenverzeichnis	I
Abbildungsverzeichnis	II
Abkürzungsverzeichnis	III
1 Einleitung	1
1.1 Webservices	1
1.2 Datenbanken	1
1.3 PHP/HTML/CSS	5
1.4 Ansätze zum Chemikalienmanagement	7
1.4.1 Webservices	7
1.2.2 Lokale Datenbanken	7
2 Zielstellung	9
3 Software	10
4 Konzeption	11
4.1 Datenbankstruktur	11
4.2 Web-Interface	13
5 Implementierung	15
5.1 Datenbankstruktur	15
5.2 Web-Interface	17
5.3 Nutzerverwaltung	19
5.4 Erweiterungen	21
5.5 Dokumentation	22
5.6 Validierung	23
6 Zusammenfassung	30
Literaturverzeichnis	32

Tabellenverzeichnis

Tabelle 1: Fiktiver Datensatz zur Prüfung der Import-Funktion.....	25
Tabelle 2: Erwartete Werte der Tabellen „Stoffe“ und „Lieferung“ nach dem Import	26
Tabelle 3: Werte des einzutragenden Stoffes "Catalase"	28
Tabelle 4: Werte der manuell einzutragenden Lieferung	29

Abbildungsverzeichnis

Abbildung 1: Beispielhafter Aufbau einer Tabelle als Teil einer relationalen Datenbank	2
Abbildung 2: Verknüpfung von zwei Tabellen über einen foreign key	3
Abbildung 3: Funktionsweise von PHP [12]	5
Abbildung 4: Datenbankschema abgeleitet aus dem bestehenden Excel-Verzeichnis, mit Tabellennamen, Attributen und zugehörigen Datentypen	12
Abbildung 5: Aktualisierte Datenbankstruktur	16
Abbildung 6: Ausschnitt der Startseite des Web-Interfaces, Darstellung mit Google Chrome	17
Abbildung 7: Exemplarischer Ausschnitt der Website	18
Abbildung 8: Oberfläche der Nutzerverwaltung	20
Abbildung 9: Teilschritte der Testprozedur mit Fehlerindikatoren	23

Abkürzungsverzeichnis

HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
CSS	Cascading Stylesheets
W3C	World Wide Web Consortium
SQL	Structured Query Language
DB	Datenbank
DBMS	Datenbankmanagementsystem
SDB	Sicherheitsdatenblatt
WSDL	Web Services Description Language
ID	Identifikator
PDF	Portable Document Format
GHS	Global harmonisiertes System zur Einstufung und Kennzeichnung von Chemikalien

1 Einleitung

1.1 Webservices

Laut W3C ist ein Webservice eine Softwareanwendung zur Maschine-zu-Maschine-Interaktion, die innerhalb eines Netzwerkes bereitgestellt wird. Die Beschreibung des Interfaces erfolgt in einem maschinenlesbaren Format (Standard: WSDL). Die Kommunikation erfolgt typischerweise über HTML oder ähnliche Web-Standards [1].

Ein Webservice ist typischerweise nach dem Server-Client-Prinzip aufgebaut, mit einem Server und mindestens einer Client-Anwendung. Typische Anwendungen für Webservices sind Synchronisationsdienste, Feedreader oder Datenbanksysteme [1].

Webanwendungen sind Webservices, die ein Interface für die Kommunikation mit Benutzern bereitstellen. Diese werden anders als normale Software nicht auf dem Rechner des Benutzers installiert, sondern werden typischerweise vom Benutzer mit einem Browser aufgerufen. Nutzereingaben werden auf dem Server verarbeitet und das Ergebnis an den Benutzer übergeben. Typische Anwendungen sind zum Beispiel Verbindungsdaten oder Shopping-Portale. Es ist möglich, Teile der Logik und Rechenlast der Webanwendungen auf den Client, also den Rechner des Benutzers, zu übertragen. Dieses Prinzip wird als „Fat Client“-Prinzip bezeichnet, im Gegensatz zum „Thin Client“-Prinzip, bei dem der Client nur die Darstellung der Ergebnisse übernimmt. Einige Webanwendungen verwenden bestimmte Laufzeitumgebungen wie JavaScript oder Flash, die auch auf dem Rechner des Clients installiert werden müssen [2,3].

1.2 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 [4].

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 jedes DBMS eine Datenbanksprache an [4,5].

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

Relationale Datenbanken bestehen aus mehreren Tabellen, in denen die Daten zeilenweise gespeichert werden. Die Datensätze verschiedener Tabellen können beliebig miteinander verknüpft werden. Diese Verknüpfungen werden ebenfalls in Tabellen gespeichert. Der Aufbau einer Tabelle ist in Abbildung 1 anhand eines Beispiels dargestellt [4,5,6].

Das Diagramm zeigt den strukturierten Aufbau einer Tabelle. Oben befinden sich zwei Kästen: 'Schlüssel' auf der linken Seite und 'Attribute' auf der rechten Seite. Ein Pfeil führt von 'Schlüssel' zu einer blau umrandeten Spalte 'ID' in der Tabelle. Ein weiterer Pfeil führt von 'Attribute' zu einer Reihe von Spaltennamen: 'Vorname', 'Nachname', 'Straße' und 'PLZ'. Ein Kasten 'Wert' unter der Tabelle weist auf einen spezifischen Wert in der Zelle 'Max' im Spaltenbereich 'Vorname' hin. Rechts neben der Tabelle befindet sich ein Kasten 'Datensätze', der durch einen geschweiften Klammerstrich mit dem gesamten Tabellendatenrahmen verbunden ist.

ID	Vorname	Nachname	Straße	PLZ
1	Max	Mustermann	Bahnhofstraße 3	02305
2	Frank	Mändel	Goethestraße 5a	02156
3	Lisa	Schubert	Badstraße 27	0205

Abbildung 1: 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 Tabellschema gibt an, wie viele Attribute eine Tabelle hat sowie deren Namen und Vorgaben für die Werte wie vorgeschriebene Datentypen, Einmaligkeit der Werte oder das Verbot von fehlenden Werten [4,5,6].

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 [5,6].

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

Autos

ID	Inhaber_ID	Marke	Kennzeichen
1	1	Audi	PIR-AT-1337
2	2	BMW	DW-II-11
3	2	HotWheels	AFK-FM-42

Einwohner

ID	Vorname	Nachname	Straße	PLZ
1	Max	Mustermann	Bahnhofstraße 3	02305
2	Frank	Mändel	Goethestraße 5a	02156
3	Lisa	Schubert	Badstraße 27	0205

Abbildung 2: 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 verringert. Ebenfalls vereinfacht wird die Manipulation der Daten. Wenn sich zum Beispiel die Adresse des Einwohners ändert, muss nur der entsprechende Datensatz aktualisiert werden. Diese Art der Verknüpfung wird als 1-zu-n-Verknüpfung bezeichnet, da ein Datensatz der Tabelle „Einwohner“ mit beliebig vielen Datensätzen der Tabelle „Autos“ verknüpft werden kann. Eine andere Form der Verknüpfung ist die 1-zu-1-Verknüpfung, bei dem genau einem Datensatz der Tabelle Einwohner genau ein Datensatz der Tabelle Autos zugewiesen würde, also jeder Einwohner genau ein Auto haben müsste. Die dritte Form ist die m-zu-n-Verknüpfung, wenn jeder Einwohner beliebig viele Autos besitzen könnte und jedes Auto beliebig vielen Einwohnern gehören könnte. Die Verknüpfung erfordert eine eigene Tabelle, in der die Beziehungen zwischen den Einwohnern und den Autos gespeichert werden.

Die Aufteilung von Attributen auf verschiedene Tabellen zur Vermeidung von Redundanzen wird Normalisierung genannt. Der Grad der Normalisierung einer Datenbank wird als Normalform bezeichnet. Eine gebräuchliche Einteilung geht von der ersten bis zur fünften Normalform. Eine vollständig normalisierte Datenbank enthält keine vermeidbaren Redundanzen mehr [4,7].

Die erste Normalform ist erreicht, wenn die gespeicherten Attribute nur noch atomare Werte enthalten. Die Normalform wäre in Abbildung 2 verletzt, wenn das Attribut „Vorname“ und „Nachname“ zu dem Attribut „Name“ zusammengefasst würden.

Die zweite Normalform bezieht sich auf die Abhängigkeit der Attribute von den Schlüsseln der Tabellen und ist bei Tabellen in der ersten Normalform mit nur einem Schlüssel automatisch erfüllt. Die dritte Normalform ist erreicht, wenn keine Nichtschlüsselattribute voneinander funktional abhängig sind. Angenommen, Personen mit einem bestimmten Nachnamen müssten zwingend einen bestimmten Vornamen tragen, wären Vor- und Nachname in Abbildung 2 funktional voneinander abhängig und müssten zur Erfüllung der dritten Normalform als eigene Tabelle gespeichert werden [7]. Die vierte und fünfte Normalform wurden in dieser Arbeit nicht verwendet.

Eine weit verbreitete Datenbanksprache für relationale Datenbanken ist 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. In dieser Arbeit wurde die Open-Source-Version verwendet. Als Werkzeug zur Administration der Datenbank wurde die Webanwendung phpMyAdmin verwendet, die neben einem SQL-Terminal auch eine graphische Oberfläche zur Anzeige und Manipulation der Daten bietet.

1.3 PHP/HTML/CSS

PHP, HTML und CSS sind die derzeit führenden Webstandards für Webdesign und Webanwendungen. HTML (Hypertext Markup Language) ist eine Auszeichnungssprache, die für die Strukturierung von digitalen Dokumenten und die Ausstattung mit Inhalten wie Hyperlinks oder Bildern verwendet wird. HTML-Dateien werden mit Webbrowsern dargestellt und bilden die Grundlage von Webseiten. Eine HTML-Datei enthält neben dem im Webbrowser angezeigten Text Angaben zur Formatierung und Metadaten wie die im Text verwendeten Sprachen oder den Namen des Autors. Für dynamische und interaktive Webseiten werden meist Programmiersprachen wie PHP oder Javascript verwendet [9,10,11].

PHP (Hypertext Preprocessor) ist eine Skriptsprache, die zur Gestaltung dynamischer Webseiten verwendet wird. Wird vom Client eine Anfrage an den Server gestellt, wird serverseitig aus dem PHP-Code ein HTML-Dokument generiert und an den Client gesendet. Die Funktionsweise eines PHP-Skriptes ist in Abbildung 3 dargestellt.

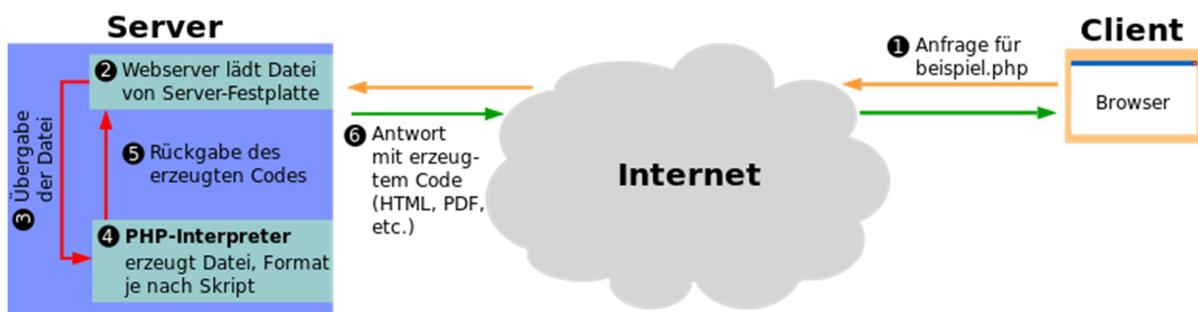


Abbildung 3: Funktionsweise von PHP [12]

Der Server reagiert automatisch auf alle in der Anfrage erhaltenen Eingaben und erzeugt eine entsprechende HTML-Datei (seltener auch PDF- oder Bilddateien). Nachteil der serverseitigen Verarbeitung der Eingaben ist die hohe Rechenlast und Rechendauer für den Server, da bei jeder neuen Anfrage der Code neu verarbeitet werden muss. Andererseits wird dadurch die zu übermittelnde Datenmenge reduziert. Weiterhin kann der Benutzer nicht auf den serverseitig gespeicherten Code zugreifen oder die PHP-Skripte manipulieren [12].

CSS (Cascading Style Sheet) ist eine Stylesheet-Sprache, die für die Formatierung der HTML-Dokumente verwendet wird. Die Sprache wurde vom World Wide Web

Konsortium entwickelt und wird beständig erweitert. Ziel der Sprache war eine Trennung von Inhalt und Design einer Website, um Designs seitenübergreifend und unabhängig vom Seiteninhalt wiederverwenden zu können. CSS kann für die Erstellung von externen Stylesheets (engl. für Formatvorlage) für die seitenübergreifende Formatierung oder von internen Styles zur Formatierung von Textabschnitten innerhalb einer Datei verwendet werden. In den Stylesheets werden die mit HTML strukturierten Segmente der Datei mit Formatierungs-Attributen versehen, die der Browser interpretiert und bei der Anzeige der Datei berücksichtigt. In den Stylesheets sind Angaben zur Positionierung von Seitenelementen, zur Textformatierung und zur Formatierung des Browserfensters gespeichert. Die Interpretation der Angaben ist teilweise vom Browser abhängig, wodurch Seiten in verschiedenen Browsern unterschiedliche Designs haben können [17].

1.4 Ansätze zum Chemikalienmanagement

1.4.1 Webservices

Eine Alternative zu der Webanwendung wäre ein Webservice ohne Benutzeroberfläche. Stattdessen würde eine Client-Software eingesetzt werden, um mit der Datenbank zu kommunizieren und eine graphische Oberfläche bereitzustellen. Diese Software könnte auf jeder Sprache mit Bibliotheken für Datenbankanbindung geschrieben werden. Ein Vorteil dieses Ansatzes wäre eine geringere Serverlast, da nur SQL-Anfragen bearbeitet und keine HTML-Dateien generiert werden müssten. Die Wartezeiten beim Laden einer neuen Seite würden entfallen. Benutzerzugriffe könnten besser kontrolliert und reguliert werden. Allerdings müsste die Kommunikation mit der Datenbank zusätzlich gesichert werden, um Zugriffe von Unbefugten zu verhindern. Um die Software unabhängig vom Betriebssystem der Benutzer einsetzen zu können, müssten plattformübergreifende Programmiersprachen wie Java oder Python genutzt werden. Die Software müsste auf jedem Rechner installiert werden, der Zugriff auf die Datenbank benötigt.

1.2.2 Lokale Datenbanken

Ein weiterer Ansatz wäre eine lokal gespeicherte SQLite-Datenbank mit einem Nutzer-Interface. SQLite-Datenbanken sind speziell für die Einbindung in Programme konfiguriert und nicht für den Zugriff über ein Netzwerk. Die direkte Einbindung der Datenbank bedingt, dass Daten normalerweise nicht auf andere Rechner synchronisiert werden. Es gäbe einen Rechner, auf dem das Programm installiert ist und auf dem die Datenbank gespeichert wird. Der Zugriff auf die Daten könnte nur über diesen Rechner erfolgen. Dieser Ansatz würde ein Maximum an Sicherheit garantieren, da eventuelle Angreifer Zugriff auf den Rechner mit der Anwendung benötigen würden. Allerdings wäre die Nutzung unkomfortabel und die Überprüfung von Beständen mit einem anderen Rechner nicht möglich. Die gespeicherte Datenbank müsste regelmäßig auf einem anderen Speichermedium gesichert werden, um bei einem Festplattenschaden Datenverlust zu vermeiden [14].

Es besteht die Möglichkeit, die lokal gespeicherte Datenbank nach Bearbeitung mit dem Interface zum Beispiel mit einem Cloud Service zu synchronisieren, um auch von anderen Rechnern darauf zugreifen zu können. Allerdings kann es bei diesem

Verfahren zu Versionskonflikten kommen, wenn die Datenbank zeitgleich von mehreren Benutzern verändert wird.

2 Zielstellung

Ziel der Praxisarbeit war die Entwicklung und Implementierung eines browserbasierten Datenbanksystems zur firmeninternen Chemikalienverwaltung. Die Datenbank sollte auf einem internen Server liegen und über ein Webinterface für alle Rechner im Firmennetz erreichbar sein. Die Anforderungen an das Interface waren:

- Intuitiver, benutzerfreundlicher Aufbau
- Passwortschutz
- Einfaches Datenmanagement ohne manuelle Eingabe von SQL-Befehlen
- Bereitstellung von Grundfunktionen:
 - Einträge einfügen
 - Einträge bearbeiten
 - Einträge anzeigen, suchen oder filtern

Als Datenbank sollte eine mindestens in der ersten Normalform vorliegende relationale Datenbank mit dem Datenbankmanagementsystem MySQL verwendet werden. Die Grundlage soll ein bestehendes System in Form einer Excel-Tabelle sein.

3 Software

Im Zuge dieser Arbeit wurden verschiedene Entwicklungsumgebungen für die Programmierung verwendet. Die verwendeten Entwicklungsumgebungen und Editoren waren:

- Netbeans IDE
- Microsoft Visual Studio C# 2010 Express
- Atom Editor
- Notepad++

Die Strukturierung und Verwaltung der Datenbank wurde mit folgender Software realisiert:

- phpMyAdmin
- DB Designer Fork
- MySQL
- MySQLDumper

Für die Einrichtung und den Betrieb des lokalen Webservers wurde das XAMPP-Paket genutzt.

4 Konzeption

4.1 Datenbankstruktur

Das bestehende Excel-basierte Verzeichnis enthielt alle gelieferten Chemikalien, einschließlich der bereits verbrauchten Lieferungen. Jeder Eintrag wurde als Zeile mit folgenden Attributen erfasst:

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

Anhand der Attribute wurde eine Datenbankstruktur entwickelt und in die 3. Normalform gebracht:

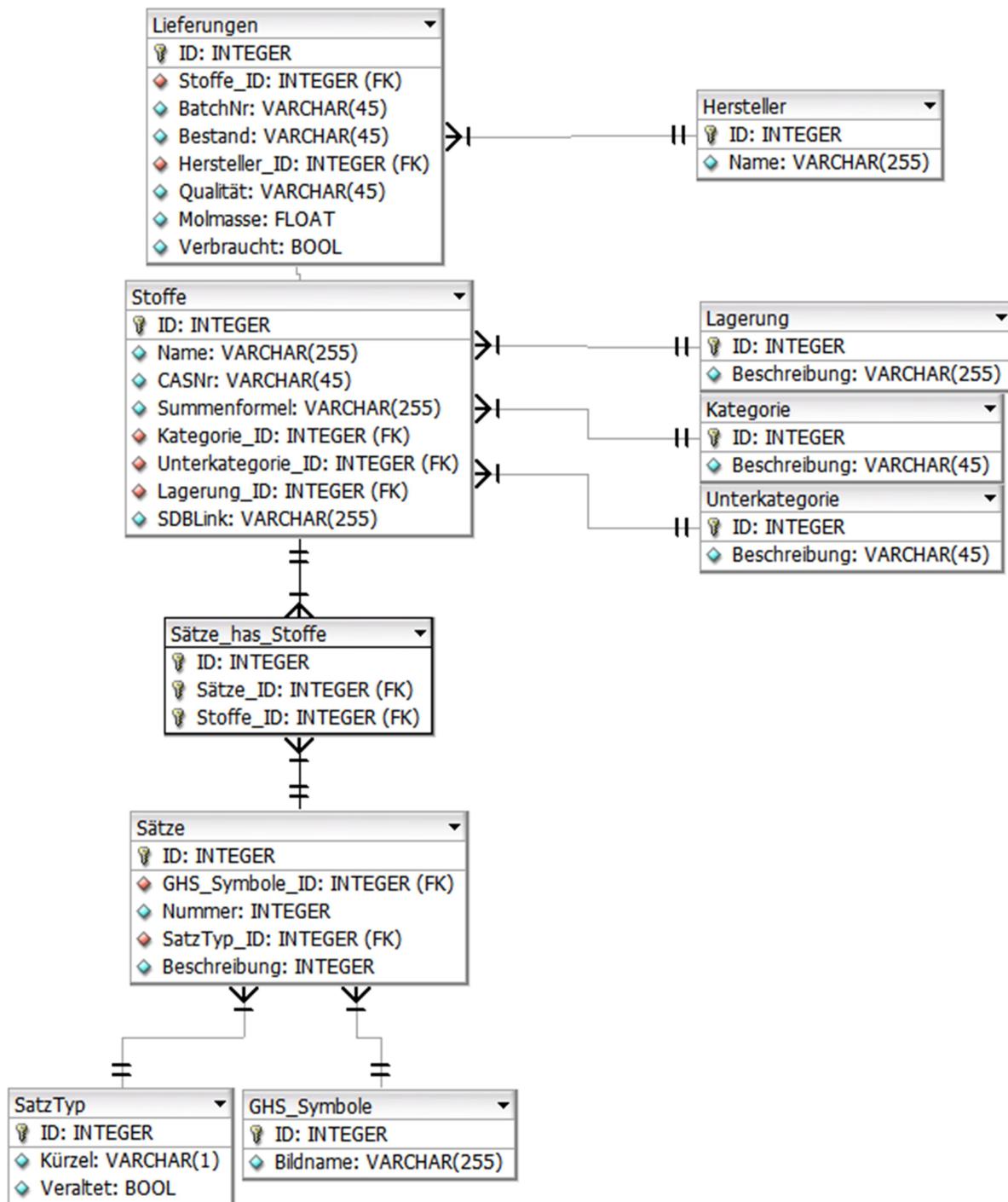


Abbildung 4: Datenbankschema abgeleitet aus dem bestehenden Excel-Verzeichnis, mit Tabellennamen, Attributen und zugehörigen Datentypen

Die vorgegebenen Attribute wurden ihrer Abhängigkeit nach in eigene Tabellen ausgelagert oder Tabellen zugewiesen. Hinter den Attributen ist der geforderte Dateityp des Attributes angegeben, zusammen mit der maximalen Länge der Werte in Klammern. Alle Stoffe wurden mit den stoffspezifischen Attributen in einer Tabelle

gespeichert, die in den Lieferungen lediglich referenziert wird. Dadurch werden Redundanzen vermieden und Speicherplatz gespart. Auch die Lagerungsvorschriften, Kategorien, Unterkategorien und Hersteller wurden als eigene Tabellen umgesetzt.

Die in der Excel-Datei als Nummern gespeicherten Sätze wurden erweitert zu eigenen Datensätzen mit der zugehörigen Nummer, der Art der Sicherheitsverordnung (hier „Satztyp“), der Beschreibung und gegebenenfalls einem GHS-Symbol. Die Verknüpfung von Stoffen und Gefahrstoffsätzen ist eine m-zu-n-Beziehung, die über eine Tabelle mit gepaarten ID's realisiert wurde.

4.2 Web-Interface

Das Interface sollte folgende Funktionalitäten bieten:

- Anzeige aller gelieferten Chemikalien
 - Filterfunktionen
 - Ein- und Ausblenden von verbrauchten Chemikalien
- Bearbeiten von Einträgen
- Löschen von Einträgen
- Einfügen
 - Neuer Chemikalien
 - Neuer Lieferungen
 - Sonstiger Datensätze

Auf der Startseite sollte eine Übersicht aller Lieferungen gezeigt werden, mit verschiedenen Filtern und Schaltflächen, um die Lieferungen einzeln zu bearbeiten oder zu löschen. Der Zugang zu dem Interface sollte über eine Anmeldungsseite geschützt werden. Ist der Nutzer nicht angemeldet, sollte er von dem Interface automatisch auf die Anmeldungsseite umgeleitet werden.

Weiterhin sollte eine für alle Seiten standardisierte Navigationsleiste eingefügt werden. Über die Leiste sollten die Seiten zum Einfügen von Chemikalien, Lieferungen und sonstigen Datensätzen erreichbar sein. Jede dieser Seiten sollte eine Eingabemaske für die einfache Eingabe der benötigten Werte bereitstellen. Wenn ein Verweis auf eine andere Tabelle nötig ist, sollten dafür die Einträge der referenzierten Tabelle in einer Drop-down-Liste gesammelt werden und zur

einfachen Auswahl zur Verfügung stehen. Nach Bestätigung der Eingaben sollten alle Werte auf Einhaltung der Datentypen und grobe Fehler überprüft werden, bevor die Eingaben in der Datenbank abgelegt werden. Bei Fehleingaben sollte eine Fehlermeldung ausgegeben werden.

Für die Verwaltung der Datenbank und des Interfaces sollte ein administrativer Bereich angelegt werden, der nicht über Schaltflächen erreichbar ist. In diesem Bereich sollten folgende Funktionalitäten realisiert werden:

- Nutzerverwaltung
- Datenbank sichern
- Datenbank leeren
- Zugang zu phpMyAdmin
- Ausführung beliebiger SQL-Befehle

5 Implementierung

5.1 Datenbankstruktur

Während der Umsetzung der geplanten Datenbankstruktur wurde das Verzeichnis neu strukturiert, um die Abhängigkeiten zwischen den Attributen besser abzubilden. Es wurde eine Tabelle für die Nutzerverwaltung eingefügt, die keine Relationen zu den Chemikalien besitzt. Es wurde zudem für Lieferungen das Datum der Lieferung und der Öffnung eingefügt und der Verweis auf die Gefahrstoffsätze für die Stoffe durch eine Zeichenkette ersetzt. Die implementierte Datenbankstruktur ist in Abbildung 5 gezeigt.

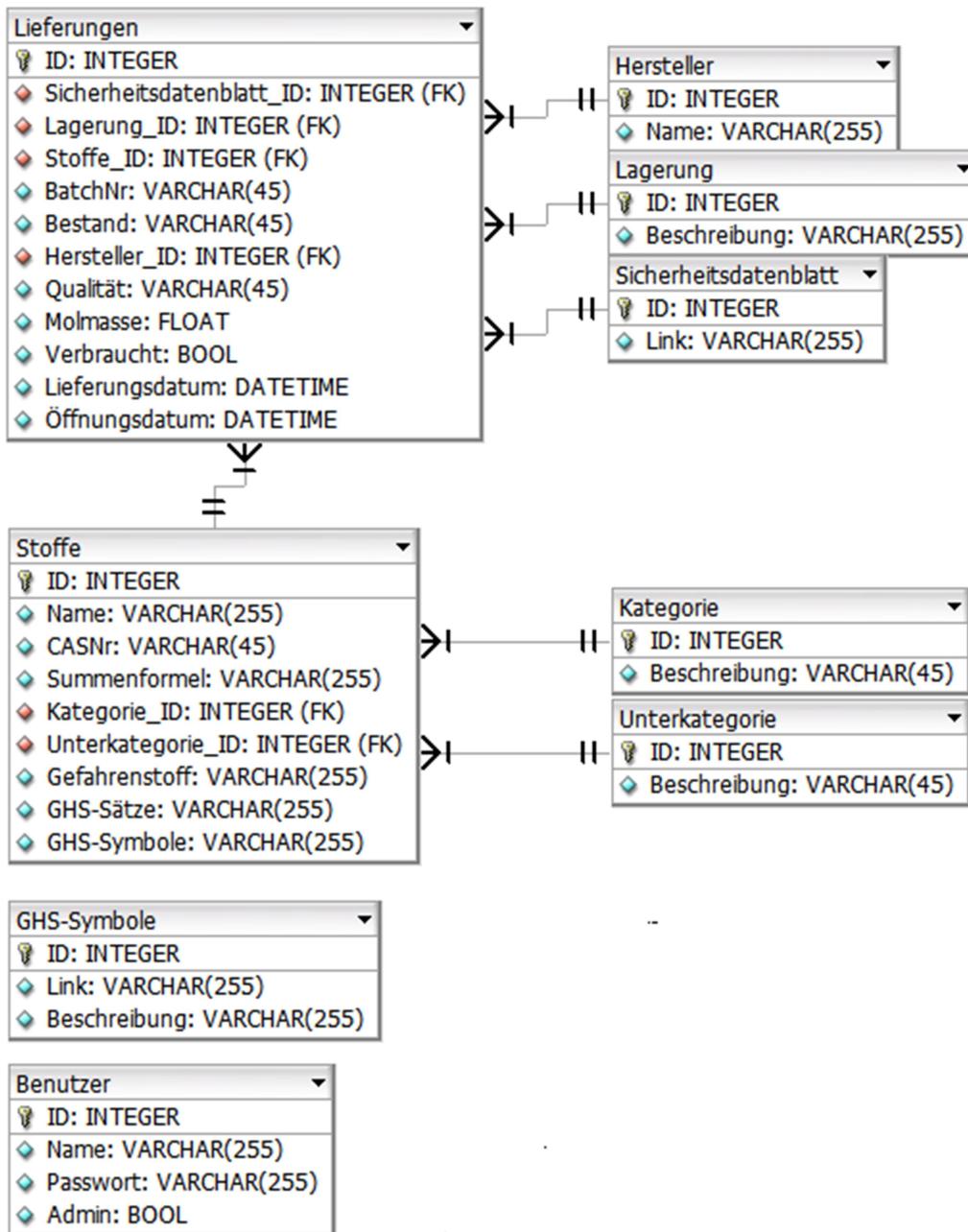


Abbildung 5: Aktualisierte Datenbankstruktur

Die Tabelle „GHS-Symbole“ hat in der Datenbankstruktur keine Relationen zu den Chemikalien. Die Verbindung wird erst über ein PHP-Skript hergestellt, das aus der Zeichenkette „GHS-Symbole“ in der Tabelle „Stoffe“ die zugehörigen Symbole ermittelt und darstellt.

Das Attribut „SDBLink“ der Stoffe, der Link zum Sicherheitsdatenblatt, wurde ersetzt durch einen Verweis auf die Tabelle „Sicherheitsdatenblatt“ in der Tabelle der Lieferungen. Der Verweis auf die Lagerungsvorschrift wurde aus der Stoff-Tabelle in

die Lieferungstabelle verschoben. Der Verweis auf die Gefahrenstoffsätze wurde durch eine Zeichenkette ersetzt, aus der mithilfe von PHP die Gefahrenstoffsätze ermittelt werden kann.

5.2 Web-Interface

Das Interface wurde wie in 4.2 beschrieben programmiert. Nach der Anmeldung wird der Nutzer auf die Startseite weitergeleitet, auf der das Verzeichnis aufgelistet ist. Ein Ausschnitt der Startseite ist in Abbildung 6 gezeigt.

The screenshot shows a web application for managing chemical inventories. At the top, there's a navigation bar with links: 'Lieferung eintragen', 'Neue Chemikalie definieren', 'SDB hochladen', 'Ressource eintragen', 'Inhalt Chemikalienschrank', and 'Logout'. Below the navigation is a blue header bar with the text 'We Let Your Data Speak' and the 'quo data' logo. The main content area has a title 'Übersicht der eingetragenen Chemikalien' (Overview of registered chemicals). On the left, there's a 'Filter' section with input fields for Name, Summenformel, Unterkategorie, Lagerung, CASNr, Kategorie, and Hersteller. There are also checkboxes for 'Verbrauchtes einschließen' and 'Nur Gefahrstoffe anzeigen', with a 'Anwenden' button. The main part of the page is a table listing chemicals:

Name	BatchNr	Bestand	Qualität	Hersteller	Lagerung	Aktionen
(L)-Dehydroascorbinsäure	BCBP3679V	250 mg		Aldrich	RT, Chemikalienschrank	
11-Mercaptoundecanoic acid	BGBC0041V	5 g	0,95	Sigma-Aldrich	RT, Chemikalienschrank	
17- <i>o</i> -Ethinylestradiol	SZBE070XV	250 mg	0,998	Sigma-Aldrich	RT, Chemikalienschrank	
17- <i>o</i> -Ethinylestradiol	119H0921	100 mg	K.A.	Sigma	Kühlschrank Labor 1a	
17- <i>o</i> -Ethinylestradiol	1065X	250 mg	K.A.	Riedel de Haen	Kühlschrank Labor 1a	
17- <i>o</i> -Ethinylestradiol-Lösung	FN01281501	1 mL	1 mg/ml in Methanol; Referenzstandard	Cerilliant	Gefrierschrank -24°C	
17- <i>β</i> -Estradiol	044K1027	250 mg	>98 %	Sigma	Kühlschrank Labor 1a	

Abbildung 6: Ausschnitt der Startseite des Web-Interfaces, Darstellung mit Google Chrome

Über die Filter im linken oberen Bereich können die in der Tabelle angezeigten Einträge eingegrenzt werden. Für jeden Eintrag gibt es am rechten Rand Schaltflächen zum Bearbeiten oder Löschen oder Inspizieren des Datensatzes. Auf der Startseite wurden aus Platzgründen nicht alle Attribute dargestellt. Der Name der Stoffe ist ein Link zu einer Detail-Ansicht der Lieferung, in der alle hinterlegten Werte angezeigt werden. Gefahrstoffe sind farblich hinterlegt, verbrauchte Lieferungen werden kursiv dargestellt. Ebenfalls aus Platzgründen wurde die Navigationsleiste als Balken an das obere Ende der Seite gelegt.

Das typische Layout der Seiten ist in Abbildung 7 gezeigt.

The screenshot shows a web application interface for substance entry. On the left, there's a navigation sidebar with links like 'Übersicht', 'Lieferung eintragen', 'Neue Chemikalie definieren', 'SDB hochladen', 'Ressource eintragen', 'Inhalt Chemikalienschrank', and 'Logout'. The main area is titled 'Stoff eintragen'. It contains input fields for 'Name', 'CASNr', and 'Summenformel', along with dropdowns for 'Ähnliche Stoffe' (similar substances) and category selection ('Bakterium' and 'Adsorbenz'). Below these are sections for 'Angaben zur Gefahrstoffverordnung:' (Hazardous Substances Regulation) with fields for 'R-Sätze', 'H-Sätze', 'S-Sätze', and 'P-Sätze'. A checkbox labeled 'Gefahrstoff' is checked. Below it, nine hazard pictograms are displayed in a 3x3 grid, each with a code: GHS01 (explosion), GHS02 (flammable), GHS03 (combustible), GHS04 (corrosive), GHS05 (irritant), GHS06 (toxic), GHS07 (warning), GHS08 (hazardous to the environment), and GHS09 (danger for the environment). At the bottom is a 'Bestätigen' (Confirm) button.

Abbildung 7: Exemplarischer Ausschnitt der Website

Die Navigationsleiste ist links positioniert, der Seiteninhalt nach rechts verschoben. Über die Navigationsleiste sind alle Seiten zum Einfügen von Datensätzen erreichbar, weiterhin besteht die Möglichkeit sich von der Sitzung abzumelden oder zur Startseite zurückzukehren.

Die Kommunikation mit der Datenbank erfolgt automatisch anhand der Nutzereingaben, wobei alle SQL-Befehle mit PHP erzeugt, überprüft und an die Datenbank gesendet werden. Die manuelle Ausführung von SQL-Befehlen ist im Administrator-Bereich möglich.

Der Administrator-Bereich wurde mit den folgenden Funktionalitäten umgesetzt:

- Nutzerverwaltung
- Datenbank sichern
- Zugang zu phpMyAdmin
- Ausführung beliebiger SQL-Befehle
- Chemikalienverzeichnis importieren

Für die Sicherung der Datenbank wurde das Tool „MySQLDumper“ [16] angepasst und in die Website eingebunden. Sicherungskopien der Datenbank mitsamt der

Struktur und allen Datensätzen können lokal als eine .sql-Datei gespeichert und bei Datenverlust wieder auf den Server importiert werden.

Es wurde weiterhin eine Oberfläche für das manuelle Eingeben und Ausführen von SQL-Befehlen erstellt. Für die direkte Verwaltung der Datenbank sollte allerdings das Tool phpMyAdmin genutzt werden, da es eine graphische Oberfläche und bessere Ergebnisausgabe besitzt.

Das Design und Layout der Seiten wurde über ein externes Stylesheet mit CSS definiert und ist einheitlich für alle Seiten. Unterschiede wie die breite Tabelle auf der Startseite wurden durch das Einbinden von CSS-Code in den Quelltext der Seiten realisiert. Das Design der Seiten wurde an die Homepage der QuoData GmbH angepasst.

5.3 Nutzerverwaltung

Die Nutzerverwaltung wurde zuerst in Form eines Textdokuments auf dem Server realisiert. Dieses Dokument enthielt alle eingetragenen Benutzer und Passwörter. Bei einer Anmeldung wurde mit PHP auf das Dokument zugegriffen um die Eingaben zu überprüfen. Zur Verwaltung der Nutzer konnte das Dokument mit einem einfachen Editor bearbeitet werden. Auf das Dokument konnte allerdings auch mit dem Browser zugegriffen werden, wenn die Verzeichnisstruktur bekannt war.

Um dieses Sicherheitsrisiko zu vermeiden, wurden die Nutzerdaten in die Datenbank integriert. Es wurde eine neue Tabelle „Benutzer“ angelegt, die keine Relationen mit den anderen Tabellen besaß. Vorteil der Speicherung in der Datenbank war die verbesserte Skalierbarkeit und erhöhte Sicherheit der Daten. Da die Verwaltung nicht mehr über einen Editor erfolgen konnte, wurde ein Tool dafür in den Administrator-Bereich des Interfaces integriert. Die Oberfläche des Tools mit einem Beispiel-Datensatz ist in Abbildung 8 gezeigt.

Nutzerverwaltung

Fehler beim Eintragen in die Datenbank: Duplicate entry '2' for key 'PRIMARY'

Eintrag bearbeiten

ID	Name	Passwort	Admin
1	Martin	PW123	<input type="checkbox"/>

Bestätigen

Übersicht der gespeicherten Nutzer

ID	Name	Passwort	Admin	Aktionen
1	Martin	PW123	Nein	
2	Admin	admin	Ja	
3	Labor	BioTechnik1	Nein	

User hinzufügen

Abbildung 8: Oberfläche der Nutzerverwaltung

Mit dem Tool können alle Benutzer angezeigt, bearbeitet oder gelöscht werden. Weiterhin können Administrator-Rechte vergeben oder entzogen werden und neue Benutzer eingefügt werden. Die Passwörter der Benutzer liegen sowohl in dem Tool als auch in der Datenbank als nicht codierter Klartext vor, um die Performance zu verbessern. Allerdings sind die Daten dadurch anfällig gegen Angriffe von außen, bei denen die Inhalte von Tabellen ausgelesen werden. Gelingt das Auslesen der Benutzertabelle, hätte der Angreifer alle Benutzernamen und Passwörter und somit vollen Zugriff auf das Interface.

Eine Alternative wäre das chiffrieren der Passwörter in der Datenbank. Nach einem Angriff mit SQL-Injections hätte der Angreifer nur die chiffrierten Passwörter. Allerdings müsste für die Anzeige in der Nutzerverwaltung ein Entschlüsselungsalgorithmus hinterlegt werden, der durch Probieren oder das Auslesen der PHP-Skripte eventuell von dem Angreifer umgangen werden kann. Die sicherste Methode wäre, die Passwörter mit einem nicht rückrechenbaren Verfahren wie einem Hash-Algorithmus zu verschlüsseln und nur das verschlüsselte Datenpaket zu speichern. Aus diesem Paket kann nur mit sehr hohem Rechenaufwand und unter Kenntnis der Funktionsparameter wie Passwortlänge, Zeichensatz und Hash-Algorithmus das Passwort extrahiert werden. Die Anzeige der Passwörter in der Nutzerverwaltung wäre mit diesem Verfahren nicht möglich [17].

Da das Interface und die Datenbank nur über ein lokales Netzwerk erreichbar sein sollen, ist die Gefahr eines gezielten Angriffes auf die Benutzer-Tabelle unwahrscheinlich, da ein potentieller Angreifer erst in das Netzwerk eindringen müsste. Aufgrund der besseren Performance wurde die Speicherung als Klartext umgesetzt.

5.4 Erweiterungen

Zu der Datenbank wurden weitere Tools geschrieben, die den Übergang von dem 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. Im Anschluss kann der Import über das Interface gestartet werden. Über eine mit C# geschriebene Anwendung werden alle Dateinamen auf unzulässige Zeichen wie Umlaute oder 'ß' überprüft und gegebenenfalls angepasst. Anschließend wird 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 Konsolenanwendung greift über die von Microsoft zur Verfügung gestellte Interop-Schnittstelle auf das Verzeichnis zu und importiert alle Daten als Array. Anschließend werden zuerst die Kategorien, Unterkategorien, Lagerungsvorschriften und Hersteller importiert, dann die Stoffe und abschließend die Lieferungen. Dadurch ist sichergestellt, dass alle Datensätze, auf die verwiesen werden soll, in der Datenbank vorhanden sind. Das Tool kann nur vollständige Verzeichnisse importieren. Fehlen obligatorische Angaben, bricht das Programm mit einer Fehlermeldung ab. Außerdem sollten Abweichungen wie „GmbH“ und „gmbh“ vermieden werden. Das Tool ist über die Navigationsleiste im Administrator-Bereich erreichbar und startet automatisch den Import der Sicherheitsdatenblätter. Die Angaben zum Lieferungs- und Öffnungsdatum sind in der Excel-Datei nicht enthalten, daher wurde für alle importierten Lieferungen das Datum „01.01.1901“ festgelegt.

Für die vereinfachte Interaktion von Client-Software wie dem Import-Tool mit der Datenbank wurde eine einfache Schnittstelle mit PHP programmiert, das beliebige SQL-Befehle entgegennehmen und die Antwort der Datenbank ausgeben kann. Die

Ausgabe erfolgt als zusammengesetzte Zeichenkette mit festgelegten Trennzeichen, die alle ausgegebenen Informationen enthält. Die Anwendungen können daraus die Antwort der Datenbank auslesen und die Daten verarbeiten. Es wurde keine Passwortsicherung für diese Art des Datenbankzugriffes festgelegt.

Weiterhin wurde ein Export-Tool für die Generierung eines Reports geschrieben, der alle in einem bestimmten Schrank gelagerten Chemikalien in eine druckfertige Excel-Ansicht exportiert und zum Download bereitstellt.

5.5 Dokumentation

Für die Benutzung des Webinterfaces wurde ein kurzes Handbuch geschrieben, das die grundlegenden Schritte zur Verwaltung der Chemikalien stichpunktartig zusammenfasst. Die Benutzung des Administrator-Bereiches ist ebenfalls beschrieben, sollte aber nur von Personal mit Erfahrung im Umgang mit Datenbanken durchgeführt werden. Weiterhin wurde für alle Mitglieder, die die Software nutzen sollen, eine Einführung gehalten und ein Ansprechpartner bei Fragen und Problemen festgelegt.

Der Quellcode des Webinterfaces wurde auf einem internen GitLab-Server hinterlegt und ist für alle Mitarbeiter des Projektes einsehbar. Der Quellcode der Erweiterungstools ist auf einem für Administratoren erreichbaren Netzlaufwerk gespeichert. Der Code wurde kommentiert, falls das Projekt von einem neuen Mitarbeiter bearbeitet werden muss. Eine chronologische Liste der Änderungen im Projekt wurde automatisch mit GitLab erstellt und ist für alle Mitarbeiter des Projektes einsehbar.

5.6 Validierung

Zur Validierung der Software wurde eine Testprozedur entwickelt, die alle essentiellen Funktionen des Interfaces prüft, ohne dabei die gespeicherten Daten zu gefährden. Die Teilschritte der Prozedur und die Fehlerindikatoren für jeden Schritt sind in Abbildung 9 gezeigt.

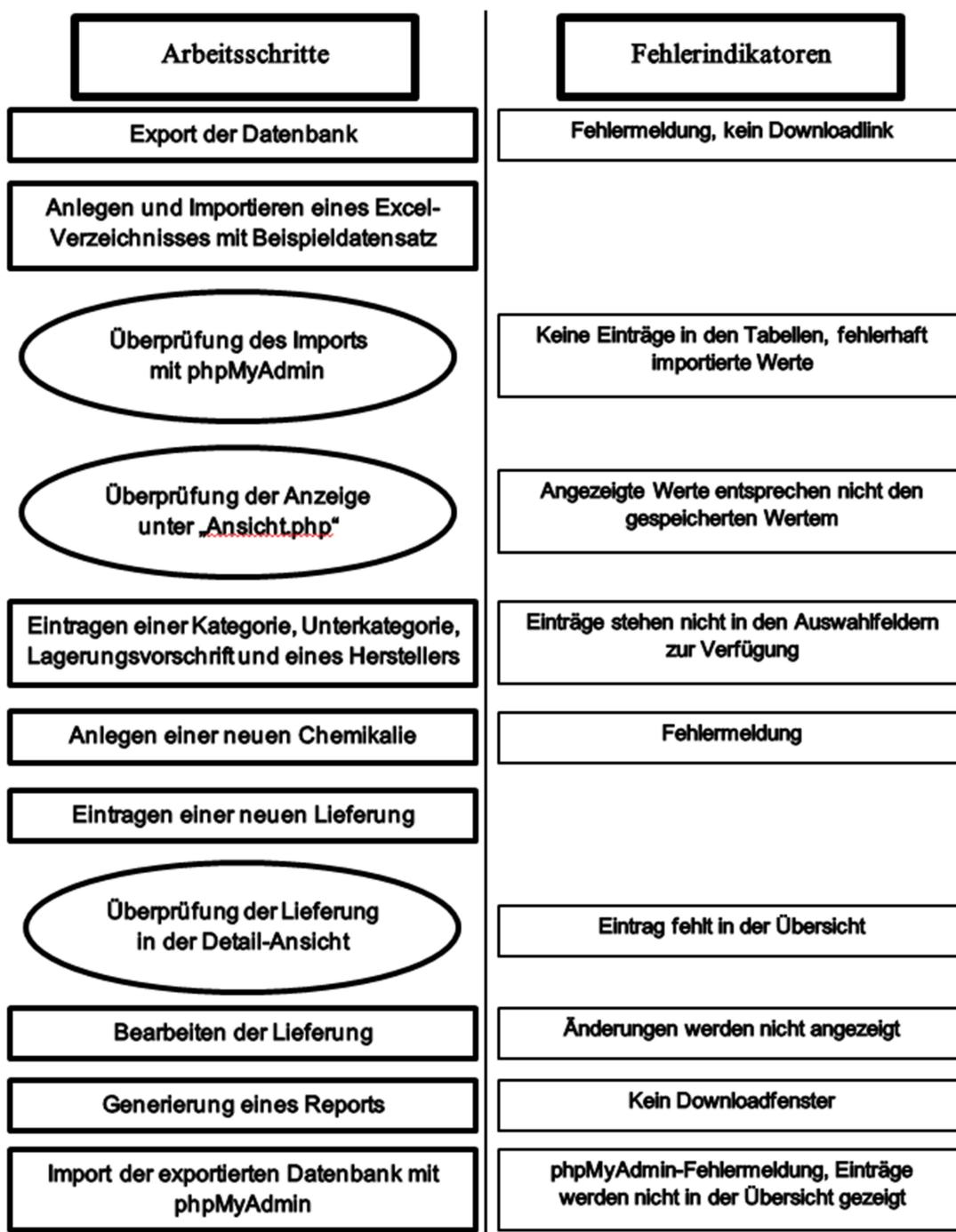


Abbildung 9: Teilschritte der Testprozedur mit Fehlerindikatoren

Die Prozedur ist unterteilt in rechteckig dargestellte Arbeitsschritte und oval dargestellte Überprüfungsschritte. Arbeitsschritte beinhalten eine Manipulation der gespeicherten Daten, Überprüfungsschritte dienen dem Abgleich von ein- und ausgegebenen Werten. Die Prozedur wurde mit fiktiven Werten durchgeführt. Tritt während der Prozedur ein Fehler auf, sollte ein Entwickler benachrichtigt werden.

Zuerst wurde die Anmeldung als Benutzer mit Administrator-Berechtigungen durchgeführt. Anschließend wurde die Seite „Admin.php“ aufgerufen, um den administrativen Bereich zu erreichen. Über die Schaltfläche „Datenbank exportieren“ wurde die Sicherung der Datenbank eingeleitet. Nach erfolgreichem Export der Datenbank sollte ein Link erscheinen, der zum Download der Sicherheitskopie führt. Die Sicherheitskopie wurde lokal gespeichert, um im Falle eines Fehlers in der Software die gespeicherten Daten sicher aufzubewahren. Erscheint kein Downloadlink oder gibt es eine Fehlermeldung bei der Ausführung des Exportes, sollte die Testprozedur nicht weiter ausgeführt werden, da sonst die gespeicherten Daten verloren gehen.

Nach dem Export wurde erneut der administrative Bereich aufgerufen und über die Schaltfläche „Chemikalienverzeichnis importieren“ eine Kopie des Excel-Chemikalienverzeichnisses importiert, die nur einen Datensatz enthielt. Der Datensatz enthielt die folgenden Werte:

Tabelle 1: Fiktiver Datensatz zur Prüfung der Import-Funktion

Attribut	Werte
Chemikalie	Ammoniummolybdat Tetrahydrat
CAS-Nummer	12054-85-2
Kategorie	Chemikalie
Unterkategorie	
Lagerungsvorschrift	RT, Chemikalienschrank
Hersteller	Sigma-Aldrich
Menge	25 g
Qualität	Analytical grade
Batch-Nummer	BCBN3102V
Summenformel	$H_{24}Mo_7N_6O_{24} * 4H_2O$
Molmasse	1235,86
Gefahrstoff	Nein

Anschließend wurde über die Schaltfläche „phpMyAdmin“ die Oberfläche des Verwaltungstools aufgerufen. War der Import erfolgreich, muss in den Tabellen „Kategorie“, „Lagerung“ und „Hersteller“ ein Eintrag mit den angegebenen Werten vorhanden sein. In der Tabelle „Unterkategorie“ darf kein Eintrag erscheinen. Die Tabellen „Stoffe“ und „Lieferung“ müssen jeweils genau einen Eintrag zeigen, der die folgenden Werte hat:

Tabelle 2: Erwartete Werte der Tabellen „Stoffe“ und „Lieferung“ nach dem Import

Stoff	Werte
ID	1
Name	Ammoniummolybdat Tetrahydrat
CASNr	12054-85-2
Summenformel	$H_{24}Mo_7N_6O_{24} * 4H_2O$
Kategorie_ID	1
Unterkategorie_ID	0
Gefahrstoff	0
Sätze	
GHS-Symbole	000000000
Lieferung	Werte
ID	1
Stoff_ID	1
Batch-Nummer	BCBN3102V
Menge	25 g
Hersteller_ID	1
Qualität	Analytical Grade
Molmasse	1235,86
Lagerung_ID	1
Verbraucht	0
Lieferungsdatum	01.01.1901
Öffnungsdatum	01.01.1901

SDB_ID	1
---------------	---

Hierbei ist zu prüfen, ob alle Eingaben mit den angezeigten Werten übereinstimmen oder ob Daten bei dem Import verloren gingen oder verfälscht wurden. Nach der Prüfung mit phpMyAdmin wurde die eingetragene Lieferung auf der Seite „Ansicht.php“ weiter inspiziert. Die in der Tabelle angezeigten Werte müssen exakt mit den in der Excel-Tabelle getätigten Eingaben übereinstimmen. Ist dies nicht der Fall, handelt es sich um einen Fehler beim Abrufen der Daten aus der Datenbank oder beim Anzeigen der Werte.

Anschließend wurde über die Schaltfläche „Ressource eintragen“ die Seite zum Eintragen von Kategorien, Unterkategorien, Herstellern und Lagerungsvorschriften aufgerufen. Es wurde die Kategorie „Protein“, die Unterkategorie „Enzym“, der Hersteller „Fischer Scientific“ und die Lagerungsvorschrift „-20 °C“ eingetragen. Über die Schaltfläche „neue Chemikalie definieren“ wurde ein neuer Stoff mit den folgenden Werten eingetragen:

Tabelle 3: Werte des einzutragenden Stoffes "Catalase"

Attribut	Werte
Name	Catalase
CASNr	9001-05-2
Summenformel	
Kategorie	Protein
Unterkategorie	Enzym
R-Sätze	1,2
S-Sätze	27-38
P-Sätze	
H-Sätze	5
Gefahrstoff	Ja

Weiterhin wurde für den Stoff das Gefahrstoffsymbol „Totenkopf“ und „Gasflasche“ ausgewählt. Über die Schaltfläche „SDB hochladen“ wurde die PDF-Datei „Sicherheitsdatenblatt_Catalase.pdf“ hochgeladen und in die Datenbank eingetragen.

Anschließend wurde über die Schaltfläche „Lieferung eintragen“ eine Lieferung mit den folgenden Werten eingetragen:

Tabelle 4: Werte der manuell einzutragenden Lieferung

Lieferung	Werte
Stoff	Catalase
Hersteller	Fischer Scientific
Molmasse	
Batch-Nummer	k.A.
Menge	10 mg
Qualität	
Lieferdatum	21.05.2016
Öffnungsdatum	22.05.2016
Lagerungsvorschrift	-20 °C
Sicherheitsdatenblatt	Sicherheitsdatenblatt_Catalase.pdf
Verbraucht	Ja

In der Übersicht darf diese Lieferung nur zu sehen sein, wenn der Filter „Verbrauchtes einschließen“ ausgewählt wurde. Über die Schaltfläche mit der grünen Lupe neben dem Eintrag wurden die gespeicherten Werte auf Abweichungen von den Eingaben geprüft. Dann wurde die manuell eingetragene Lieferung bearbeitet, um die gleichen Werte wie die importierte Lieferung zu haben. Über die Schaltfläche „Inhalt Chemikalienschrank“ wurde ein Report generiert, der alle Chemikalien mit der Lieferungsvorschrift „RT, Chemikalienschrank“ einschließt.

Nach der Prüfung des Reports auf Abweichungen von den eingegebenen Werten wurde im Administrator-Bereich über das Tool „phpMyAdmin“ die gespeicherte Datei mit den ursprünglichen Daten importiert, um die Datenbank wieder mit den gespeicherten Werten zu füllen. Die Software wurde bereits von Benutzern getestet. Alle auftretenden Fehler wurden dokumentiert und umgehend behoben.

6 Zusammenfassung

Das Chemikalienmanagement ist ein essentieller Bestandteil der Laborverwaltung und der Qualitätssicherung. Dazu gehört, die Chemikalien zentral zu erfassen, die Integrität und Konsistenz der Daten zu sichern und die Zugriffe auf die Daten zu regulieren. Es existieren Software-Lösungen für das Chemikalienmanagement, wie das kostenpflichtige Programm „Chemikalienverzeichnis“ von Network Inventory® oder die kostenfreie Webanwendung „Quartz.com“. Diese Anwendungen bieten vorgefertigte Eingabemasken und eingebundene Datenbanken zur Speicherung der Daten. Allerdings kann nicht direkt auf die Datenbanken zugegriffen werden. Es werden außerdem keine Informationen zu Sicherheitsmaßnahmen bereitgestellt. Weiterhin ist der Quellcode nicht einsehbar, dementsprechend können die Anwendungen nicht ohne Absprache mit den Entwicklern mit Erweiterungen versehen werden. Eine mögliche Erweiterung ist die Anbindung von anderen Inventarisierungsmethoden, zum Beispiel für Geräte, an die Datenbank oder eine Funktion, um überlagerte Chemikalien herauszufiltern und zur Entsorgung zu markieren. Ebenfalls ein wichtiger Punkt war die Möglichkeit zur lokalen Datensicherung und volle Kontrolle über die Datensicherheit. Dementsprechend wurde eine eigene, angepasste und erweiterbare Software benötigt, die sowohl eine intuitive Benutzeroberfläche als auch einen direkten Zugriff auf die Datenbank ermöglicht.

Um diesen Anforderungen gerecht zu werden, wurde im Rahmen dieser Arbeit eine MySQL-Datenbank mit einem Webinterface programmiert und auf einem firmeninternen Server bereitgestellt. Der Zugang zu der Benutzeroberfläche ist passwortgeschützt. Über das Interface sind grundlegende Funktionalitäten wie das Anzeigen, Bearbeiten, Einfügen und Löschen von gelieferten Chemikalien möglich sowie die Generierung eines Reports zu den Beständen in bestimmten Lagerorten. Über einen gesonderten Administrator-Bereich ist die Verwaltung der zugelassenen Benutzer, der Import und Export der Datenbank sowie die manuelle Ausführung von Datenbankbefehlen möglich. Das Projekt ist nicht für die Nutzung im Internet geeignet, es fehlen essentielle Sicherheitsmaßnahmen wie die Verschlüsselung der gespeicherten Passwörter und Maßnahmen gegen das Einbringen von Code über die Eingabemasken. Auf diese Maßnahmen wurde

aufgrund der Beschränkung auf das firmeninterne Netzwerk zugunsten der Performance verzichtet. Der Quellcode des Projektes wurde für alle Mitarbeiter des Projektes zugänglich auf einem internen Server oder Netzlaufwerk gespeichert.

Die Software wurde von Benutzern auf Fehler und Probleme getestet. Weiterhin wurde eine Test-Prozedur zum systematischen Testen der Funktionalitäten entwickelt und durchgeführt. Perspektivisch könnte diese Test-Prozedur noch weiter ausgebaut und standardisiert werden, um einen größeren Teil des Funktionsumfangs abzudecken.

Ebenfalls geplant ist ein verbessertes SQL-Terminal, das in den Administrator-Bereich integriert werden soll. Unter anderem soll es verbesserte Fehlerbehandlung und eventuell ein Syntax-Highlighting besitzen. Weiterhin wäre eine Seite zur Bearbeitung der gespeicherten Kategorien, Unterkategorien, Hersteller und Lagerungshinweise denkbar, da bisher nur Administratoren diese Einträge bearbeiten können.

Literaturverzeichnis

- [1] <https://www.w3.org/TR/ws-arch/#whatis>, Stand: 14.07.2016
- [2] <http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>, Stand: 16.07.2016
- [3] Monitoring in Webanwendungen, Kim R. Mertens, 2013
- [4] <http://www.itwissen.info/definition/lexikon/Datenbank-DB-database.html>, Stand: 15.07.2016
- [5] Entwicklung von Datenbankanwendungen, Aus- und Fortbildungszentrum Bremen, Mai 2004. Abgerufen: 03.08.2016
- [6] <https://www.hdm-stuttgart.de/~riekert/lehre/db-kelz/chap6.htm>, Stand: 13.07.2016
- [7] <https://www.hdm-stuttgart.de/~riekert/lehre/db-kelz/chap4.htm>, Stand: 13.07.2016
- [8] Datenbanken: Grundlagen und Design, F. Geisler, 2014
- [9] <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>, Stand: 14.07.2016
- [10] https://w3techs.com/technologies/overview/programming_language/all, Stand: 14.07.2016
- [11]
https://w3techs.com/technologies/history_overview/programming_language/ms/y, Stand: 14.07.2016
- [12] <https://www.neuhold.pro/php/kapitel0>, Stand: 13.07.2016
- [13] <https://www.w3.org/Style/CSS/>, Stand: 13.07.2016
- [14] <https://www.sqlite.org/>, Stand: 14.07.2016
- [15] <http://www.softguide.de/programm/chemikalienverzeichnis>, Stand: 14.07.2016
- [16] <http://chemikalienverzeichnis.de/>, Stand: 14.07.2016
- [17] <http://www.mysqldumper.de/>, Stand: 10.07.2016

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 Praxisarbeit selbst-
ständig verfasst und keine anderen als die an-
gegebenen Quellen und Hilfsmittel benutzt.

.....

.....

.....

(Ort) (Datum) (Unterschrift)