



HÖHERE TECHNISCHE BUNDESLEHRANSTALT Wien 3, Rennweg
IT & Mechatronik

HTL Rennweg :: Rennweg 89b
A-1030 Wien :: Tel +43 1 24215-10 :: Fax DW 18

Diplomarbeit

365 365 Speiseplan

ausgeführt an der
Höheren Abteilung für Informationstechnologie/Medientechnik
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2018/2019

durch

**Pascal Skupa
Marina Yazykova
Christian Perl
Marwan Abdalla**

unter der Anleitung von

Mag. Andreas Fink
Dipl.-Ing Gabriela Herrele

Wien, 6. März 2019

Kurzfassung

Mit Hilfe der Webapplikation "365" lässt sich ein Speiseplan für ein ganzes Jahr generieren. Der Speiseplan ist individuell anpassbar und auf Useranforderungen abgestimmt. Mit der Einkaufslistenfunktion kann man alle Lebensmittel, die in der kommenden Woche gebraucht werden, offline abrufen und beim Einkaufen kontrollieren. Die Rezepte werden nach bestimmten Tags, wie vegetarisch, high-protein oder laktosefrei, sortiert. Sollte man gegen ein bestimmtes Lebensmittel allergisch sein, so kann man das angeben und alle Rezepte in denen dieses enthalten ist, werden ausgetauscht. Eine zusätzliche Funktion hilft einem die Woche genauer zu planen, so lässt sich zum Beispiel einstellen, dass zweimal in der Woche Fleisch und einmal in der Woche Fisch vorkommen sollen. Persönliche Präferenzen können in einem privaten Account festgelegt und gespeichert werden.

Abstract

The 365 application generates a customizable meal plan for the whole year. With the shopping list function all the ingredients that will be needed in the coming week are directly inserted into the shopping list. The recipes are sorted by specific tags, such as vegetarian, high-protein or lactose-free. Allergens to a particular food can be specified and all recipes that contain it will be replaced. An extra function helps to plan the week more precisely. For example, it is possible to set up meat twice a week and fish once a week. All details and personal preferences can be set and stored in a private account.

Ehrenwörtliche Erklärung

Ich erkläre an Eides statt, dass ich die individuelle Themenstellung selbstständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Wien, am 6. März 2019

Pascal Skupa

Marina Yazykova

Christian Perl

Marwan Abdalla

Inhaltsverzeichnis

Abbildungsverzeichnis	xi
1 Die Idee	1
2 Projektziele	3
2.1 Hauptziele	3
2.2 Optionale Ziele	3
2.3 Nicht-Ziele	4
3 Das Team	5
4 Datenquelle	7
4.1 Woher die Rezepte?	7
4.2 Rezeptdatenbanken mit APIs im Vergleich	7
4.2.1 Entscheidung	9
4.3 FatSecret Schnittstellen	9
4.3.1 JavaScript	10
4.3.2 Rest API	10
5 Algorithmus	13
6 Backend	15
6.1 Datenbank	15
6.1.1 Anforderungen an die interne Datenbank	15
6.1.2 Relationale und nicht-relationale Datenbanken	16
6.1.3 Open Source Datenbanken	18
6.1.4 Erstellen und Verwalten der Datenbankstruktur	20
7 Frontendfunktionalitäten	23
8 Design	25
8.1 Designidee[9]	25
8.2 Farben[8]	25
8.3 Logo[6]	26
8.3.1 Die Erstellung mittels Adobe Photoshop und Adobe Illustrator	26
8.4 Angular Frontend[?]	27
Literaturverzeichnis	29

Abbildungsverzeichnis

6.1 ER-Modell	20
-------------------------	----

1 Die Idee

2 Projektziele

2.1 Hauptziele

Ziel-H 1 Wochenkochplan

Ein Wochenkochplan ist für die laufende Woche ausgewertet und übersichtlich dargestellt. Für jeden Tag der Woche sind min. 0 und max. 4 Mahlzeiten vorhanden. Das kann in den Präferenzen eingestellt werden. Ebenso kann angegeben werden welche Mahlzeiten man an welchem Tag haben möchte. Für jede Mahlzeit ist eine Zutatenliste und ein Rezept vorhanden.

Ziel-H 2 Tags

Die Tags der Zutaten umfassen Mikronährstoffe, Allergene und Besonderheiten des Produkts. Die Tags der Speisen umfassen die einzelnen Zutaten. Mit allen Tags der Zutaten kann man auswerten welche Eigenschaften eine Mahlzeit hat (z.B. Unverträglichkeiten) um somit den Plan nach den Userpräferenzen einstellen zu können.

Ziel-H 3 Einkaufsliste

Anhand des Speiseplans für die ganze Woche kann eine Einkaufsliste mit allen Zutaten erstellt werden. Zutaten sind in der lokalen Datenbank gespeichert und können editiert, hinzugefügt und gelöscht werden.

Ziel-H 4 Userpräferenzen

Im Usermenü kann mit Hilfe eines Formulars eingestellt werden wie viele und welche Art von Mahlzeiten (Frühstück, Mittag, Abendessen, Snack) man an einem bestimmten Wochentag haben möchte. Ebenso können individuelle Essensgewohnheiten (z.B. dreimal die Woche Fleisch, jeden Tag eine Suppe), Allergien und ungewollte Speisen eingetragen werden. All das wird mit Hilfe von Tags, die die einzelnen Speisen und Zutaten zugeordnet haben, ausgewertet.

2.2 Optionale Ziele

Ziel-O 1 Effizienz

Um die Download- und Auswertungszeiten zu verkürzen, werden Daten für bis zu

5 Wochen aus der API zwischengespeichert und in der lokalen Datenbank abgelegt. Die neu geladenen Rezepte oder Zutaten werden mit den schon vorhandenen verglichen, dies spart Zeit.

Ziel-O 2 Gänge Menü

Man kann sich für gewünschte Tage (z.B. Feiertage) ein bis zu 5 Gänge Menü generieren lassen.

Ziel-O 3 Erweiterte Userpräferenzen

User können anhand einer Checkliste in dem Formular mit Hilfe von Speisen- und Zutatentags den Speiseplan nach bestimmten Ernährungskonzepten einstellen. Zum Beispiel: „high-protein“, „laktosefrei“, „vegetarisch“ usw.

Ziel-O 4 Offline

Der Wochenplan und die Einkaufsliste lassen sich exportieren und sind offline abrufbar. So kann man zum Beispiel im Geschäft, sollte man kein Internetzugang haben, offline die Produktliste und den Plan ansehen.

2.3 Nicht-Ziele

Ziel-N 1 Android-/IOS-App

Die Applikation ist als native Android- oder IOS-App verfügbar.

Ziel-N 2 Rezeptdatenbank

Das Projektteam erstellt eine eigene Rezeptdatenbank.

3 Das Team

4 Datenquelle

4.1 Woher die Rezepte?

Es gibt zwei Möglichkeiten die notwendigen Daten für die Rezepte zu sammeln und diese abzuspeichern. Das selbstständige Schreiben der Rezepte in einer Projekteigenen Datenbank oder das Einbinden einer Rezeptdatenbank aus dem Internet. Die Erste fiel weg, da der Aufwand alle Rezepte abzutippen und richtig zu formatieren zu groß wäre. Zusätzlich würde man die Nutzungs-, Verbreitungs-, und Veröffentlichungsrechte für die Rezepte brauchen. Sollten diese aus einem Kochbuch kommen, wäre dies kompliziert und sehr aufwändig geworden. Somit fiel die Wahl auf eine Rezeptdatenbank, die Rezepte sind direkt verfügbar, die Verbindung ist leicht herzustellen und die Benutzerrechte sind klar geregelt.

Der Zugriff soll mit Hilfe einer API[1] erfolgen.

Eine API, Application Programming Interface, ist eine Schnittstelle, die einen für alle Komponenten lesbaren Datenaustausch ermöglicht. Das macht es für komplizierte Anwendungen leichter zwischen den einzelnen Bestandteilen zu kommunizieren und den Programmieren diese zu warten. Dabei werden Module separiert und interagieren miteinander über eine API um somit die vielfältige Anwendbarkeit, leichtere Fehlerkorrektur und problemlose Erweiterungen zu ermöglichen.

Im Fall der 365 Speiseplan Anwendung sind diese Module der Algorithmus, die interne Datenbank mit der Userverwaltung und der Zwischenspeicherung sowie die Rezeptdatenbank. Die Rezepte werden automatisch durch schon von der Webseite vorgegebene und in den API Request (Zugriff) eingebunden Funktionen aus der Datenbank ausgelesen und sortiert. Die Daten werden von einem Anbieter eventuell kostenpflichtig zu Verfügung gestellt und können, wenn man eine Lizenz besitzt, in den Kochplan eingebaut werden.

4.2 Rezeptdatenbanken mit APIs im Vergleich

Die wichtigsten Entscheidungskriterien sind der Umfang der Rezepte, die Anzahl der erlaubten Zugriffe und der Preis für Schulgruppen und Non-Profit-Organisationen.

Für das Projekt relevante Datenbanken und kostengünstige Pakete		
Food2fork [5]		
Kriterien	Paket Nr.1	Paket Nr.2
Beschreibung	für Entwickler- und Lernzwecke	für Beginner und Studenten
Rezeptanzahl	keine Angabe	keine Angabe
Preis	gratis	7\$/Monat
API Calls	50/Tag	250/Tag
Notizen	Pro Anfrage liefert die API nur 30 Datensätze zurück. Für den Algorithmus, der viele Hunderte Datensätze mit einander vergleicht, sind 30 auf ein Mal zu wenig und es müssten viel mehr Zugriffe für eine Ausgabe gemacht werden.	
Spoonacular [16]		
Kriterien	Paket Nr.1	Paket Nr.2
Beschreibung	Basic-Paket	Akademisches Paket
Rezeptanzahl	360.000	360.000
Preis	Gratis bis 500Calls/Tag, danach 0.003\$ bis 0.007\$/Call	10\$/Monat
API Calls	500/Tag	5000/Tag
Notizen	Für den Algorithmus könnten mehr als 500 Calls benötigt werden. Die weiter steigenden Kosten sind schwer einschätzbar und könnten vor Allem während der Testphase hoch steigen.	
Edamam [2]		
Kriterien	Paket Nr.1	Paket Nr.2
Beschreibung	für Entwickler	für Startups
Rezeptanzahl	200	10.000
Preis	gratis	gratis für Startups
API Calls	keine Angabe	keine Angabe
Notizen	Caching ist erlaubt.	
FatSecret [3]		
Kriterien	Paket Nr.1	Paket Nr.2
Beschreibung	Basic	PREMIER Free
Rezeptanzahl	keine Angabe	keine Angabe
Preis	gratis	gratis für Studenten, Startups und Non-Profit Organisationen
API Calls	5000/Tag	unlimitiert
Notizen	Die Plattform wirbt damit, dass sie die meist verwendete Nährstoffdatenbank der Welt ist und dass mehr als 10.000 Entwickler in mehr als 50 Ländern diese verwenden.	

Zusätzlich wurde die Billa AG bezüglich ihrer Rezeptdatenbank kontaktiert. Sie bieten

keine Datenbank für öffentliche Verwendung an.

4.2.1 Entscheidung

Nach einer Bewertung und genauer Evaluierung aller Angaben ist die Entscheidung auf die "FatSecret API" gefallen. Die Basic Version ermöglicht den vollen Zugriff auf alle, in ihrer Datenbank vorhandenen, Rezepte und bietet die meisten gratis Zugriffe an.

PREMIER Free Zusätzlich wurde der Support von FatSecret kontaktiert und bezüglich einer "PREMIER Free" Version nachgefragt. Diese Version bietet alle Premium-Funktionen der US-Datenbank an. Mit einem Formular, dass der Schuldirektor, Dipl.-Ing Gerhard Jüngling, ausgefüllt und unterzeichnet hat, konnte die PREMIER Free Version beantragt werden. In wenigen Tagen wurde der FatSecret Account von 365 von Basic auf PREMIER Free hochgestuft.

4.2.1.1 Rechtliche Bedingungen

FatSecret bietet ein "PREMIER Free" Paket für Startups, Non-Profits und Studenten an. Um den Zugriff zu bekommen muss folgendes bestätigt, in unserem Fall vom Schuldirektor, und eingehalten werden:

- Das Team besteht nicht länger als 2 Jahre.
- Der Gewinn, der vom Team durch dieses Projekt in der Zukunft gemacht werden kann, darf die Höhe von 1 Million USD nicht übersteigen.

4.3 FatSecret Schnittstellen

Mit einem Zugangsschlüssel [4], den man nach der Anmeldung auf der FatSecret API Webseite erhält, kann man mit Hilfe der zu Verfügung gestellten API auf die Datensätze zugreifen. Für den Zugriff gibt es zwei Optionen, JavaScript API und REST API.

4.3.1 JavaScript

Der Zugriff über die JavaScript API beinhaltet viele Funktionen für Nährstoffe, Bewegungstracking und Gewichtskontrolle. Dabei kann man das grafische Interface der FatSecret Anwendung direkt in die Webseite einbinden. Zusätzlich werden Funktionen mitgeliefert, mit denen man das Layout anpassen kann. Um die Applikation einzubinden, muss man den JavaScript-Link in ein Script-Tag schreiben `<script src="http://platform.fatsecret.com/js?key=XXX&auto_load=true"></script>`. Durch das Hinzufügen des Abfrageparameters `"auto_load = true"` wird die Komponente automatisch geladen und den Benutzern in der Position angezeigt, in der dieses Skripttag auf der Webseite platziert ist. Zu beachten ist, dass jeder Verweis auf `"http://platform.fatsecret.com/js"` einen gültigen Zugriffsschlüssel enthalten muss, dieser muss an die URL angehängt werden. Den Schlüssel bekommt man nach der Anmeldung auf der FatSecret-API-Plattform.

Das Problem darin ist, dass die Anwendung, mit einer zum großen Teil vorgegebenen Formatierung, direkt in die Webseite eingebunden wird. Das bedeutet, dass man nur sehr schwer Rezepte und vor allem die Kalenderansicht in dem gewünschten Format darstellen kann. Eine Lösung dafür ist der direkte Zugriff auf die Daten und das Holen von reinen JSON Files, die leicht übernommen, umformatiert und in jeder gewollten Form dargestellt werden können. Dies funktioniert mit Hilfe des REST Zugriffs.

4.3.2 Rest API

Bei REST oder RESTful API (Representational State Transfer) [15] handelt es sich um einen Softwarearchitekturansatz zum Erstellen skalierbarer Webdienste. Dieser Ansatz legt die Prinzipien für die Organisation der Interaktionen einer Anwendung mit einem Server über das HTTP-Protokoll fest. REST APIs sind stateless. Das heißt, dass Aufrufe unabhängig voneinander ausgeführt werden können. Jeder Aufruf enthält alle Daten, die zum erfolgreichen Abschluss erforderlich sind, unter anderem die Benutzerinformationen zur Authentifizierung. Die gesamten Operationen sind auf 4 reduziert:

Mit einer GET-Anfrage kann eine Ressource, meist im JSON- oder XML-Format, abgerufen werden. Mit PUT können Datensätze geändert werden, mit POST können diese auch neu erstellt werden und mit DELETE kann man sie entfernen.

Ein Vorteil von REST APIs ist, dass sie sehr flexibel sind. Da Daten nicht an Ressourcen oder Methoden gebunden sind, kann REST mehrere Arten von Aufrufen verarbeiten und unterschiedliche Datenformate zurückgeben. FatSecret bietet eine REST API an, mit die Entwickler FatSecret Plattform-Funktionen in ihre Anwendungen integrieren können. Die REST API kann zum Erstellen von Ernährungs-, Diät- und Gewichtsmanagement-Lösungen auf jeder Plattform verwendet werden, einschließlich

Desktop-Client-Anwendungen und Lösungen für mobile Geräte. Zusätzlich sind viele Zugriffsfunktionen, die automatisch Daten sortieren und eingrenzen, auf dem Git Repository verfügbar.

Der Zugriff erfolgt mit einem Zugriffsschlüssel (Consumer API Key) und dem dazugehörigen Geheimschlüssel (Consumer Secret), den man ebenso nach der Anmeldung zugeschickt bekommt. Zu beachten ist, dass jeder Aufruf der REST API korrekt signiert sein muss. Weitere Informationen zum Ausgabeformat, Signieren der Abfragen und Methoden, die die API mit den Zugriffen liefert, sind im Kapitel „FatSecret API“ zu finden.

5 Algorithmus

6 Backend

6.1 Datenbank

6.1.1 Anforderungen an die interne Datenbank

In der internen Datenbank sollen die notwendigen Daten zum benutzerdefinierten Verarbeiten der Rezepte aus der FatSecret Datenbank abgespeichert werden. Neben den Userdaten und den Userpräferenzen werden Rezept-IDs, Allergene, Unverträglichkeiten und die Produkte für die Einkaufsliste gespeichert. Die FatSecret Datenbank stellt Rezepte mit Kochanleitungen und Nährstoffangaben zu Verfügung und ordnet diese abhängig von ihren Eigenschaften in unterschiedliche Kategorien ein. So befinden sich zum Beispiel alle Rezepte mit einem hohen Proteinanteil unter der Kategorie „high-protein“ und können somit schnell herausgefiltert werden. Es gibt dennoch notwendige Informationen, die FatSecret nicht beinhaltet. Die Allergene. Dafür sollen alle Allergene recherchiert werden und in die interne Datenbank geladen werden. Später können sie verwendet werden um mit dem Algorithmus die Rezepte durchzusuchen und die Speisen, auf die der User allergisch ist, weg zu geben.

Für die interne Datenbank sind folgende Anforderungen gesetzt: Die logische Struktur der FatSecret Datenbank soll rekonstruiert und in die interne Datenbank mit übernommen werden. Abgespeichert werden die IDs der Rezepte und die Userdaten mit ihren Präferenzen. Diese sollen über einen Plan verbunden werden, der je nach Userangaben wochentagabhängig anpassbar ist. Die unterschiedlichen Allergene sollen in einer eigenen Tabelle stehen und den Usern zuordenbar sein. Zusätzlich sollen die Produkte aus der Einkaufsliste in der Datenbank vorkommen. Da nicht alle Daten aus einer anderen Datenbank kommen und schnell weiter verarbeitet werden, muss Wert darauf gelegt werden, dass die Verbindung verzugsfrei funktioniert. Mehrere User sollen gleichzeitig den Plan generieren lassen können. Dabei werden sehr viele Zugriffe sowohl auf die FatSecret API als auch auf die interne Datenbank gemacht.

6.1.2 Relationale und nicht-relationale Datenbanken

6.1.2.1 Relationale Datenbanken

In einer relationalen Datenbank werden Daten in miteinander über ein Fremdschlüssel verbundenen Tabellen gespeichert. Jede Tabelle besteht aus Spalten (genannt Attribute) und Zeilen (Datensätze). Jede Spalte hat einen eigenen Informationstyp. Einige Attribute sind Informationen zu der Tabelle. Andere enthalten Verweise, Fremdschlüssel, auf Primärschlüssel, einem logischen oder künstlich festgelegten Indikator, anderer Tabellen. Beziehungen sind das Hauptmerkmal dieses Modells.

Tabellen in relationalen Datenbanken müssen einigen Kriterien entsprechen, sie müssen „normalisiert“ werden. Mit der Normalisierung können Anomalien und Redundanzen vermieden und Konsistenz sichergestellt werden.

1NF

Die erste Normalform besteht, wenn alle Attribute atomar sind. Das bedeutet nicht mehr weiter trennbar. Zusätzlich dürfen Datensätze nicht mehrmals vorkommen. Dafür werden zusammenhängende Daten wie zum Beispiel Adresse in Ort, Stadt und PLZ getrennt. Dieser Schritt macht es möglich die einzelnen Adressteile anzusprechen.

2NF

Eine Tabelle entspricht dann der zweiten Normalform, wenn sie der 1NF entspricht und jedes Nichtschlüsselattribut von dem gesamten Schlüsselattribut abhängt. Das stellt sicher, dass alle Informationen einer Tabelle logisch zusammenhängen. Besitzt eine Tabelle in der ersten Normalform nur ein Primärschlüssel, so entspricht die Tabelle auch gleichzeitig der zweiten Normalform. Als Beispiel kann man sich eine Restaurantrechnung vorstellen. Diese hat als Primärschlüssel die Rechnungsnummer und die Kundennummer. Zusätzlich stehen aber in der Tabelle der Preis und die Personendaten des Kunden. Diese hängen nicht direkt mit beiden Primärschlüsseln zusammen und können somit in externe Tabellen ausgelagert werden.

3NF

Bei der dritten Normalform muss die Tabelle der 2NF entsprechen und kein Nichtschlüsselattribut darf von einem anderen Nichtschlüsselattribut transitiv abhängen. Wenn also das Attribut B von A abhängt und das Attribut C von B, muss C in eine externe Tabelle ausgelagert werden, da C nur indirekt von A abhängt. Kurzgefasst bedeutet es, dass die Nichtschlüsselattribute nur direkt von einem Schlüsselattributen abhängen dürfen. Hat eine Tabelle mit Kundendaten auch die Postleitzahl drinnen, so hängt die Postleitzahl von der Adresse, aber nicht direkt von dem Kunden ab. So kann man die Postleitzahl auslagern und über eine Zwischentabelle mit dem Kunden verbinden.

Relationale Datenbanken werden mit der strukturierten Abfragesprache SQL „Structured Query Language“ erstellt, befüllt, gelöscht und bearbeitet. In den 70er Jahren unter der Bezeichnung SEQUEL „Structured English Query Language“ veröffentlicht ist SQL mittlerweile der Standard für die Datenbankinteroperabilität, also der Besonderheit Systeme einwandfrei miteinander zusammenzuführen. SQL ist die Grundlage für die aktuellen Datenbank Anwendungen, unter anderem Oracle.

Datenbankentwicklung mit MySQL[10] Die Beliebtheit von MySQL stieg schon in den 90er Jahren wegen des freien Sourcecodes. MySQL wird im Allgemeinen mit der Programmiersprache PHP und dem Apache Web Server in Linux-Distributionen verwendet, was zur Abkürzung LAMP (Linux, Apache, MySQL, PHP) für das Open Source-Programmpaket führte. Bei Windows kennt man es unter XAMPP (cross-platform, Apache, MySQL, PHP, Perl). In der Entwicklung von MySQL ermöglicht unter anderem MySQLi die Verbindung zwischen PHP und der MySQL-Datenbank. MySQLi (improved) ist eine aktualisierte Version des PHP-MySQL-Treibers. Es unterstützt objektorientierte und prozedurale Programmierung. Es beinhaltet Prepared Statements, die Angriffe durch SQL-Injections verhindern. Mit den SQL-Injections können von Hackern Befehle eingeschleust werden, die etwas an der Datenbank verändern können. Sowie MySQLi, stellt PDO „PHP Data Objects“ auch eine Schnittstelle zwischen PHP und Datenbanken zu Verfügung. Es ist mit 11 anderen Datenbanksystemen zusätzlich zu MySQL kompatibel und unterstützt ebenso wie MySQLi die Prepared Statements. 2010 wurde MySQL von Oracle übernommen und erlangt seit dem immer mehr Kritik. [19] Der Grund dafür sind die eingeschränkten Möglichkeiten der freien Version im Gegensatz zu denen der kommerziellen Version. Viele neue Funktionen werden nur mehr von der kostenpflichtigen Version unterstützt oder erst viel später in die neuen Updates eingebunden. Daraus folgt die Entscheidung auf das zwar erlernte, aber im späteren Verlauf kostenpflichtige MySQL zu verzichten.

Datenbankentwicklung mit Oracle Die Oracle-Datenbank ist ein relationales Datenbankverwaltungssystem von der Oracle Corporation. Der Zugriff erfolgt über SQL. Die Erweiterung von SQL für die Arbeit mit Oracle heißt PL/SQL „Procedural Language“, sie verknüpft die prozedurale Programmiersprache mit dem gewohnten SQL. Es wurde von der Oracle Corporation entwickelt, um die SQL-Funktionen zu erweitern. Auf Grund der kostenpflichtigen Lizenzen ist Oracle für die Anwendung nicht geeignet.

6.1.2.2 Nicht-relationale Datenbanken

Nicht-relationale Datenbanken mit NoSQL NoSQL [11] „Not only SQL“ Datenbanken unterscheiden sich stark von dem relationalen Modell. SQL folgt einem strikten Schema und hat eine genau definierte Struktur. Die Datentypen und Integritätsbedingungen sind genau vordefiniert und es gibt wenig Spielraum. Bei NoSQL beinhalten die Datensätze gleich ihre Beschreibung und es ist kein einheitliches Abfrageformat

definiert. Es ist vergleichbar mit den JSON und XML Formaten, bei denen die einzelnen Felder einen Namen haben, der sie definiert. Nicht-relationale Datenbanksysteme erstellen eine Struktur im laufenden Betrieb, erlauben jeden Datentyp und speichern Objekte, Listen, Wertepaare und Dokumente. NoSQL erleichtert die Speicherung und den Zugriff auf Daten. Das am weitesten verbreitete NoSQL Datenbanksystem ist MongoDB.

6.1.2.3 Entscheidung

Wenn es um Datenintegrität geht, nehmen SQL-Datenbank-Transaktionen immer noch eine führende Position ein. Die Art und Weise wie eine NoSQL Datenbank programmiert wird muss von Anfang an erlernt und angeeignet werden. Auf Grund der im oberen Absatz genannten Argumente und dem Zeitvorteil, der durch das schon Erlernte während dem Unterricht entsteht, fällt die Entscheidung auf eine relationale Datenbank. Zusätzlich erfordert die interne Datenbank eine festgelegte Struktur und definierte Datentypen damit die Verbindungen zwischen dem Plan, den Allergenen und den Userpräferenzen genau in den Algorithmus eingebaut werden können.

6.1.3 Open Source Datenbanken

Von den relationalen Open Source Systemen stehen folgende zur Verfügung:

6.1.3.1 MariaDB [7]

MariaDB wurde vom selben Entwickler geschrieben wie MySQL. Es ist eine Abspaltung von MySQL und ihr Hauptziel besteht darin, weiterhin ein Open Source Datenverwaltungsprogramm anzubieten. Gleichzeitig soll es erhebliche Verbesserungen des Codes mit sich bringen. Wegen der Ähnlichkeit kann Zeit für das Erlernen des Codes gespart werden. MariaDB wird öfters auf Grund der häufigen Sicherheitupdates MySQL vorgezogen. MariaDB wird völlig offen entwickelt, alle Lösungen und neuen Ideen zur Entwicklung können im E-Mail-Newsletter sowie im Fehlermeldungs-system frei diskutiert werden.

6.1.3.2 PostgreSQL [13]

PostgreSQL ist ein leistungsfähiges Open Source System. Ebenso wie andere relationale Datenbanksysteme unterstützt es das Transaktionskonzept von SQL, das ACID-Konzept (Atomicity, Consistency, Isolation, Durability). Das beschreibt Kriterien, die von

SQL-Datenbanken verwendet werden, um sicherzustellen, dass Datenbankänderungen konsistent, sicher und dauerhaft gespeichert werden.

Basierend auf der leistungsstarken Technologie bewältigt PostgreSQL die gleichzeitige Bearbeitung mehrerer Aufgaben. Die Datenbank ist weit skalierbar, flexibel und kann in sehr vielen Programmiersprachen verwendet werden. Nach einer richtigen Einrichtung der Programmierungsumgebung verläuft die Verwaltung der Datenbanken sehr schnell. Ein weiteres Feature ist die Unterstützung von JSON Format, was eine schemalose Speicherung ermöglicht. Dies kann nützlich sein, wenn die Datenstruktur ein gewisses Maß an Flexibilität erfordert: Wenn sich zum Beispiel die Struktur während des Entwicklungsprozesses noch ändert und unbekannt ist, welche Felder das Datenobjekt enthalten wird.

Zusätzlich ist der Entwicklersupport auf hohem Level und die Community groß. Bei PostgreSQL wird viel Wert auf die Sicherheit und Stabilität der Daten gelegt, ebenso wie auf Integrationsmöglichkeiten und das Verwalten und Erweitern komplexer Prozesse.

6.1.3.3 SQLite [17]

SQLite ist in Form einer Bibliothek in die Anwendung eingebettet und funktioniert ganz ohne einen eigenen Server. Die meisten SQL-Datenbankmodule werden als separater Serverprozess erstellt. Programme, die auf die Datenbank zugreifen möchten, kommunizieren mit dem Server über Prozessinteraktionen (normalerweise TCP/IP), um Anforderungen an den Server zu senden und die Ergebnisse zurückzuholen. SQLite arbeitet anders. In SQLite liest und schreibt ein Prozess, der auf eine Datenbank zugreifen möchte, direkt aus den Datenbankdateien auf der Festplatte. Dadurch werden die Vermittlungsprozesse des Servers nicht benötigt. Die ganze Verwaltung funktioniert mit PHP. All das verbessert die Geschwindigkeit und Leistung der Vorgänge. Auf Grund des geringen Speicherbedarfs wird SQLite oft bei embedded Systemen angewendet, zum Beispiel bei Handy Applikationen.

Die Implementierung verläuft sehr schnell, die Datenbank ist auf Grund ihrer Bau- und -Funktionsart für das schnelle Anlegen kleiner Projekte, auch Prototypen ausgelegt. Durch das kleine Format sind die Fähigkeiten von SQLite beschränkter als die von MySQL[18]. Befinden sich die Daten auf einer anderen Datenbank, auf die über das Netz zugegriffen werden muss, so ist SQLite nicht die optimalste Lösung. Die Engine-zur-Disk Verbindung müsste mit einer höheren Bandbreite über das Netzwerk laufen. Ebenso wird es nicht empfohlen, wenn mehrere User gleichzeitig an der Datenbank Änderungen vornehmen. SQLite unterstützt nicht mehr als eine Userinteraktion mit einer Tabelle zum selben Zeitpunkt. Eine Datenbank-Engine, die einen Server verwendet, kann zusätzlich einen besseren Schutz vor Fehlern in der Clientanwendung bieten als SQLite: Ein parasitärer Zeiger im Client kann nämlich den Speicher auf dem Server nicht zerstören.

6.1.3.4 Auswahl der passenden Datenbank

Laut dem Ranking von DB-Engines [12], von Februar 2019, steht PostgreSQL auf dem 4. Platz, SQLite auf 10 und MariaDB auf Platz 12. Die Plätze darüber besetzen kostenpflichtige Datenbankentwicklungstools. Die Entwicklung von PostgreSQL begann in den 1980ern und gewinnt seit dem immer mehr Unterstützung von der Entwicklercommunity.

Eines der Ziele der Diplomarbeit ist es, neue Techniken und Programme innerhalb der Entwicklungszeit auszuprobieren und sich diese anzueignen. Aus diesem Grund fällt MariaDB weg, da im Unterricht in MySQL entwickelt wurde und diese Umgebungen sehr ähnlich sind. SQLite ist zwar schnell und leicht zu implementieren, die Größenlimitierungen reichen für das Projekt aus und dennoch kommen die Daten aus einer externen Datenbank, die über das Netzwerk zu erreichen ist, dadurch könnten hohe Wartezeiten entstehen. Daraus folgt die Entscheidung PostgreSQL als Verwaltungsprogramm für die projektinterne Datenbank zu wählen. Es läuft auf einem Server, unterstützt viele Userabfragen gleichzeitig, ist flexibel und auf alle Projektstrukturen anpassbar.

6.1.4 Erstellen und Verwalten der Datenbankstruktur

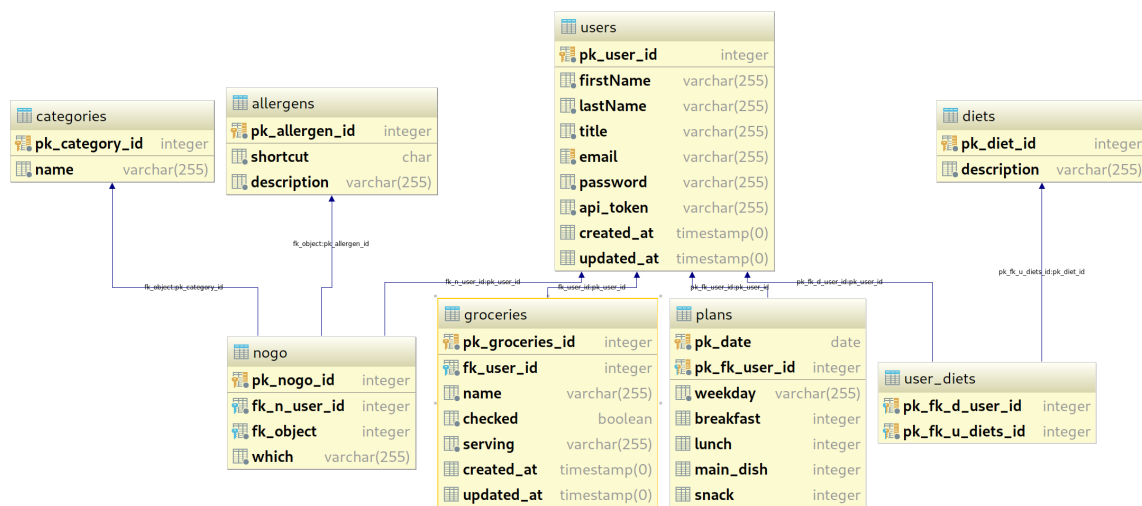


Abbildung 6.1: ER-Modell

Der Plan Der Plan wird nach Tagen mit Daten aufgeteilt und einer Person zugeteilt. Pro Tag wird ein Plan erstellt, der pro Mahlzeit (Frühstück, Mittagessen, Hauptspeise und Snack) eine Rezept-IDs aus der FatSecret Datenbank enthält. Mit Hauptspeise ist das Abendessen gemeint, die Definition kommt aus der FatSecret Datenbank. Sollte in dem Anmeldeformular festgelegt werden, dass Montags keine Frühstücke gibt, so

werden alle Frühstückseinträge mit dem Backendalgorithmus auf NULL gesetzt. Das weekday-Feld erleichtert die Arbeit für das Algorithmus, so muss nicht zusätzlich bei jeder Abfrage aus dem Datum der Wochentag ermittelt werden.

No-Gos Wenn ein User eine Allergie (allergens) oder ein Produkt aus der Liste der typischen Unverträglichkeiten (categories, eine Definition aus der FatSecret Datenbank) angibt, so wird es in der „nogo“-Tabelle eingetragen. Das Feld "which" legt fest ob es eine Allergie oder Unverträglichkeit ist, so kann man die zwei Tabellen leichter in dem Algorithmus von einander unterscheiden. Ist zum Beispiel ein User auf Fisch allergisch, so füllt er das in dem Registrierungsformular aus und es wird ein Eintrag in die „nogo“-Tabelle gemacht. In das Feld „fk_object“ wird die Allergen-ID von Fisch geschrieben, das Feld "which" mit einem "a" befüllt und beim Algorithmus darauf geachtet, dass nur Rezepte vorgeschlagen werden, die kein Fisch enthalten. Die Intoleranzen sind eine festgelegte Liste an Produkten, die oft nicht gerne gegessen werden. Darunter fallen Meeresfrüchte, Soja, Nüsse, Schweinefleisch, Rindfleisch und vieles andere. Gibt der User dies in dem Formular an, so gibt es ebenso einen Eintrag in der „nogo“-Tabelle. Somit hat ein User mehrere „nogo“-Einträge, die bei dem Algorithmus direkt berücksichtigt werden.

Groceries Die Einkaufsliste speichert alle notwendigen Produkte für die kommende Woche. Das „checked“-Statusfeld wird dann auf true gesetzt, wenn das Produkt in der Liste abgehakt, also gekauft, wurde. Die Felder "created_at" und "updated_at" sind notwendig um festzustellen seit wann das Produkt in der Liste ist und ob es seit einer Woche abgehakt ist, dann kann das Produkt aus der Liste entfernt werden.

Diets Beim Generieren des Plans werden Diäten, falls eine Diät ausgewählt ist, die in der „diet“-Tabelle stehen, miteinberechnet. Verfügbare Diäten sind „Milchfrei“, „Glutenfrei“, „High Protein“, „Kalorienarm“ und „Low Carb“. Je nach der ausgewählten Diät werden bestimmte, in der FatSecret API implementierte, Tags zur Abfrage hinzugefügt, um somit die Rezepte zu filtern. Zum Beispiel werden bei „High“-Protein automatisch Rezepte genommen, die den „high-protein“-Tag haben und somit einen hohen Anteil an Protein besitzen.

Users Die User-Tabelle speichert die typischen Userdaten, wie Name, Nachname, Titel sowie zur Anmeldung notwendige E-Mail-Adresse und das verschlüsselte Passwort. Die Felder "created_at" und "updated_at" werden für Operationen im Algorithmus gebraucht.

6.1.4.1 Datenbankverwaltungstools

Der erste Schritt von der Einrichtung einer PostgreSQL Entwicklungsumgebung ist den Installer herunterzuladen. Der Installer beinhaltet die PostgreSQL Engine, das Paketverwaltungsprogramm StackBuilder und ein grafisches User Interface für die Datenbankverwaltung pgAdmin.

Die Datenbank soll während ihrer Entwicklung nur lokal laufen. Aus diesem Grund gibt man als Speicherort den C:/xampp Ordner an. XAMPP ist ein von Apache entwickeltes Open Source-Paket für plattformübergreifende Webserverlösungen, das hauptsächlich aus dem Apache HTTP Server, der MariaDB-Datenbank und Interpreter für die Programmiersprachen PHP und Perl besteht. Es ist erweiterbar und PostgreSQL lässt sich leicht einbinden, in dem man die heruntergeladenen Ordner unter C:/xampp/pgsql hinterlegt. Zum Starten und Stoppen des PostgreSQL Servers muss man unter Windows 10 die services.msc (Windows Dienste) öffnen und mit einem Rechtsklick auf PostgreSQL 11.xx (Versionsnummer) starten, beenden oder anhalten wählen.

Mit PgAdmin lässt sich die Datenbank ganz leicht generieren und die Daten verwalten. Dennoch kann man zusätzlich phpPgAdmin installieren um die Administration der Datenbank zu erleichtern.[14] Dazu muss man die Installationsdateien in einem neuen Ordner C:/XAMPP/phpPgAdmin hinterlegen und die Inhalte der Dateien C:/XAMPP/phpPgAdmin/conf/config.inc.php, C:/XAMPP/php/php.ini und C:/XAMPP/apache/conf/extra/http-xampp.conf ändern. In der ersten Datei muss `\$conf[,extra_login_security']` auf „false“ gesetzt werden und in der zweiten muss die Zeile `;extension=php_pgsql.dll` aktiviert werden in dem man den Kommentar „;“ davor entfernt. Die http-xampp.conf Datei muss um folgende Zeilen erweitert werden:

```
Alias phppgadmin „C:/XAMPP/phppgadmin/“
<Directory „C:/XAMPP/phppgadmin“>
Options Indexes FollowSymLinks MultiViews
AllowOverride all
Require local
Order Deny,Allow
Allow from all
</Directory>
```

Jetzt kann man nach dem Starten des Apache Servers über den Browser unter localhost/phppgadmin auf PhpPgAdmin zugreifen.

7 Frontendfunktionalitäten

8 Design

8.1 Designidee[9]

Die Grundidee hinter dem Design entspringt aus dem Wunsch eine moderne und intuitiv bedienbare Benutzeroberfläche zu bieten. Diese hat sowohl simpel als auch ansprechend und kreativ auszusehen.

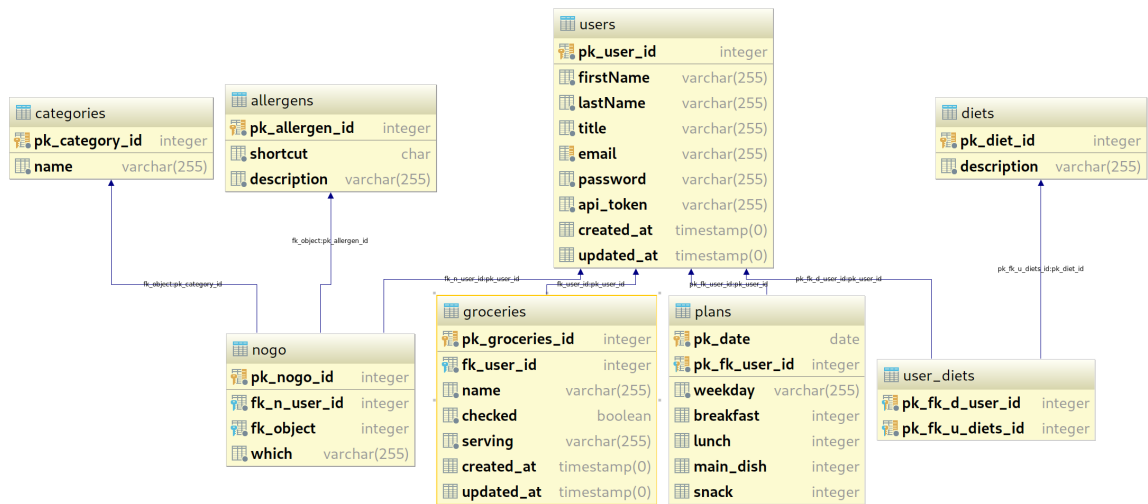
Die größten Faktoren für Zufriedenheit der Nutzer ist die Gestaltung, Aufteilung und Präsentation einer Applikation. Deshalb wird sich an die Material Design Richtlinien gehalten. Laut Material Design ist die Designsprache eine Vereinigung aus klassischen Designprinzipien und den Möglichkeiten, welche durch Technische Innovationen geschaffen werden. Die Einheitlichkeit des Designs ist ein ebenso wichtiger Aspekt. Das Nutzererlebnis soll Plattform, Gerät und Eingabemethode unabhängig gleich sein. Unter allen diesen Vorgaben und Richtlinien ein Unikates und Ansprechendes Design zu entwerfen ist die Aufgabe, welche es umzusetzen gilt.

8.2 Farben[8]

Gerade bei Webseiten, welche im Zusammenhang mit Essen stehen sind Farben wichtig. Die Farben bestimmen die Stimmung des Nutzers und bieten einen größeren Anreiz auf der Seite zu bleiben.

Die Farben sollen einen gewissen genießbaren Eindruck machen, deshalb wurden ausschließlich Farben verwendet welche in natürlichen Lebensmitteln Vorkommen.

Der Prozess lief folgendermaßen ab: Es wurden sich Farben gefunden, welche laut Material Design zu einem Harmonischem Ganzen beitragen und genügend Kontrast bieten. Dann wurden diese durch Reverse-Lookup über Google.com in Bildern welche Tags wie „Essen“, „Tasty“ oder „Delicious“ hatten wiedergefunden. Die Bilder zeigen nun Lebensmittel welche ähnlichen Farben wie die Wunschfarben aufweisen. Nun wurden die Wunschfarben nur noch an eben genau diese Natürlichen Farben angepasst.



BILD

8.3 Logo[6]

Das Logo ist der einfachste und direkteste Weg sich als Marke, Produkt oder Applikation Visuell auszudrücken. Es übermittelt die Grundidee der Applikation auf eine simple und dennoch aussagekräftige Art und Weise. Die für die Applikation konzipierte Designsprache ist in dem Logo erkennbar und reflektiert so die Identität der Applikation.

8.3.1 Die Erstellung mittels Adobe Photoshop und Adobe Illustrator

Das Logo sollte die Applikation repräsentieren und leicht mit Kochen und Rezepten in Verbindung gebracht werden können.

Mit der Umsetzung wurde begonnen indem auf Papier grobe Gedanken in Form von Skizzen aufgezeichnet wurden und so wurde mit diversen Ideen herumprobiert, um eine genauere Vorstellung zu bekommen. In diesem Prozess entstand die Idee einen Pfannenwender und Kochlöffel als Typische Symbole des Kochens zu verwenden.

Nun wurden der Pfannenwender und Kochlöffel in Photoshop mithilfe von groben Formen erstellt. Die beiden Symbole wurden überkreuzt um ein „Wappen“-ähnliches Aussehen zu erreichen. Der Entwurf wurde weiter verfeinert und angepasst. In diesem Prozess wurde das gesamte Team involviert sodass jedes Mitglied seine Gedanken preisgeben konnte. Nachdem es zu der Zufriedenheit aller verbessert wurde hat man den Entwurf zu Adobe Illustrator exportiert.

Name	Dateiendung	Verwendung
Adobe Illustrator	.ai	Das Original File für spätere Editierungen.
Portable Document Format	.pdf	Zur Verbreitung / Teilung des Logos. Universell verwendbar auf nahezu allen Geräten. Hier wird eine Vektordatei abgespeichert.
Portable Network Graphic	.png	Pixelgrafikformat welches Transparenz erlaubt. Zur Verwendung an Desktop Computern und bei der Digitalen Bildbearbeitung.
Scalable Vector Graphic	.svg	Vektordatei zur Verwendung auf Webseiten. Verlustfrei skalierbar.

Adobe Illustrator bietet hervorragende Tools zur Erstellung von Vektorgrafiken. Letzte kleine Verbesserungen wie die exakte Positionierung wurde vorgenommen. Der Entwurf wurde in Illustrator nun vektorisiert indem er mit Pfaden nachgezogen wurde. Nach letzter Überprüfung wurde das fertige Logo exportiert. Hierzu wurden diverse Formate abgespeichert.

8.4 Angular Frontend[?]

Angular ist ein Framework, um interaktive Komponenten für eine Webseite zu erstellen. Es wurde als umfangreiches JavaScript Framework von der Google LLC entwickelt. Angular ist Open-Source-Software und ist eines der größten Front-End-Webapplikationsframeworks. Es ist in TypeScript geschrieben und aufgrund der eindeutigen Unterteilung der einzelnen Komponenten eignet es sich perfekt für ThreeSix-Five.

Angular basiert auf der Model-View-Controller Art Code zu schreiben. Entwickler haben dieses Modell seit langem verwendet jedoch ist Angular das erste JavaScript Framework, welches darauf aufbaut.

Dieses Prinzip teilt die Entwicklung auf drei Ebenen auf.

„View/Template“ als das sichtbare welches die Userexperience ausmacht und mit HTML und CSS/SCSS geschrieben wird.

„ViewModel/Component“ jede Komponente definiert eine Klasse, welche die

Applikations-Daten und -Logik für ein Template enthalten.

„Service/Injector“ für Daten und Logik, welche keinem speziellen Template zugeordnet sind und welche man Komponenten-übergreifend verwenden möchte. Diese müssen mittels Injector den Komponenten als Dependency übergeben werden.

Das Model-View-Controller Modell erlaubt die Wiederverwendung von Templates und Komponenten und ist somit ideal für die Applikation ThreeSixFive. Außerdem erlaubt das two-way data binding zwischen Template und Component ständige Updates zwischen beiden Ebenen. Ohne Angular müsste der Entwickler selbst sich um den ständigen Austausch zwischen Nutzer Eingaben und Aktion/Werte-Veränderungen in der Logik kümmern. Solch eine push und pull Logik selbst zu schreiben ist umständlich und fehleranfällig.

Im Folgenden Diagram wird aufgezeigt wie Angular mit dem DOM kommuniziert. Es werden vier verschiedene Formen des data-bindings gezeigt. Jede Form hat eine Richtung:

- Zu dem DOM
- Vom DOM
- Beidseitig sowohl vom DOM als auf zum DOM

Angular modifiziert direkt die DOM-Struktur einer Seite und fügt sich nicht in das HTML ein. Dies führt zu einer besseren Performance der Applikation.

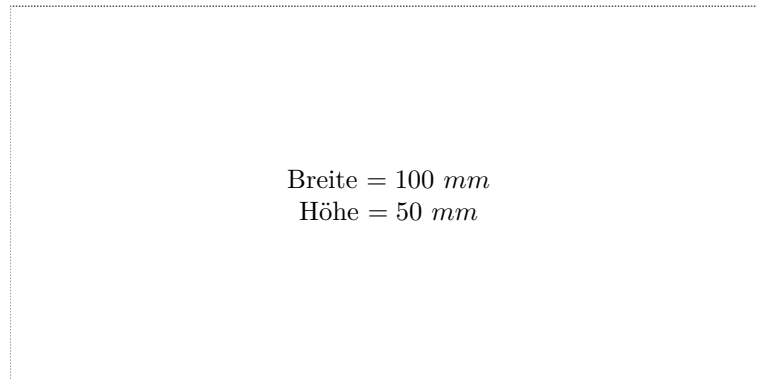
In dieser Umgebung, welche von Angular geboten wird, gilt es nun unsere Applikation umzusetzen.

Literaturverzeichnis

- [1] *API Definition.* https://praxistipps.chip.de/was-ist-api-einfach-erklaert_41370, Abruf: 2019-01-25
- [2] *Edamam API.* <https://developer.edamam.com/edamam-nutrition-api>, Abruf: 2019-01-10
- [3] *FatSecret API.* <https://platform.fatsecret.com/api/>, Abruf: 2019-01-10
- [4] *FatSecret API JavaScript.* <https://platform.fatsecret.com/api/Default.aspx?screen=jsapiqs>, Abruf: 2019-01-25
- [5] *Food2Fork API.* <https://www.food2fork.com/about/api>, Abruf: 2019-01-10
- [6] *Logo.* material.io/design/iconography/product-icons.html#design-principles, Abruf: 2019-01-09
- [7] *MariaDB Definition.* <https://www.informatik-aktuell.de/betrieb/datenbanken/mariadb-und-mysql-vergleich-der-features.html>, Abruf: 2019-02-11
- [8] *Material Design Colors.* material.io/design/color/the-color-system.html#color-usage-palettes, Abruf: 2019-01-09
- [9] *Material Design Definition.* material.io/design/introduction/#principles, Abruf: 2019-01-08
- [10] *MySQLi und PDO im Vergleich.* <https://websitebeaver.com/php-pdo-vs-mysql>, Abruf: 2019-02-11
- [11] *NoSQL Definition.* <https://www.bigdata-insider.de/was-ist-nosql-a-615718/>, Abruf: 2019-02-11
- [12] *Offizielles Ranking aller Datenbanken.* <https://db-engines.com/de/ranking>, Abruf: 2019-02-13

- [13] *PostgreSQL Anwendungsfälle*. <https://www.postgresql.org/about/>, Abruf: 2019-02-10
- [14] *PostgreSQL Verwaltungstools*.
<http://www.fly2mars-media.de/seoblog/server/anleitung-postgresql-servers-unter-xampp-installieren-102570/>, Abruf: 2019-02-20
- [15] *REST API Definition*. <https://www.mulesoft.com/resources/api/restful-api>,
Abruf: 2019-01-26
- [16] *Spoonacular API*. <https://spoonacular.com/food-api>, Abruf: 2019-01-10
- [17] *SQLite Definition*. <https://blog.capterra.com/free-database-software/>, Abruf: 2019-02-11
- [18] *SQLite Fähigkeiten und Stärken*. <http://www.dbtalks.com/tutorials/learn-sqlite/what-are-the-limitations-of-sqlite>, Abruf: 2019-02-17
- [19] *Vergleich zwischen kostenlosen Datenbanken*.
<https://www.capterra.com.de/blog/14/die-top-7-kostenlosen-und-open-source-datenbank-softwarelosungen>, Abruf: 2019-02-10

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —

Diese
Seite
nach dem
Druck
entfer-
nen!