



Rezepti

Christoph Sherr , Tobias Lotz, Dominik Hummel

June 2023

0.1 Kurzbeschreibung:

Eine Seite Für Kochrezepte von verschiedenen Ländern all around the globe.

0.2 Zielgruppe Nutzer und Beeinflusser:

Unsere Zielgruppe sind alle Personen die Kochen wollen. Nutzer könnten alle Personen sein die ein bestimmten Rezept suchen oder einfach Rezepte Entdecken wollen. Beeinflusser können Firmen sein die unseren konntet für ihre Zwecke nutzen wollen zb. Influencer die Rezepte probieren oder Firmen die Produkte durch Rezepte vermarkten wollen.

0.3 Referenzdokument

(mit Zielen, Funktionsrahmen bzw. allg. Funktionsbeschreibung, Projektrahmen, Umsetzungsstrategie, Aufgaben und Zuständigkeiten)

Ziel war es ein Plattform zu schaffen auf der man Rezepte mit anderen Menschen auf der ganzen Welt teilen kann oder von anderen Menschen auf der ganzen Welt Rezepte entdecken kann.

Zu Funktionen soll gehören das man sich einen account anlegen kann und von diBreach4-Stunner-Refuse-Doseesem aus rezepte erstellen liken und speichern kann. Es soll möglich sein rezepte ohne anmedlung einzusehen. Es soll die möglichlichkeit bestehen rezepte zu suchen oder nach besrimmten zutaten ländern oder kategorien des rezeptes zu suchen. Der admin soll die funktion haben angelegte user zu löschen.

Der Projektrahmen sind die gegebenen requirements die erfüllt wurden.

Die Umsetzungsstrategie war sehr dynamisch jeder war in einem bereich einigermaßen gut deshalb ging es einfach los und gelegentlich vergleichen bis wir an einem punkt waren das es darum ging die requirements zu erfüllen ab dann wurde mit planung schritt für schritt die requirements übernommen.

Zuständigkeit Domme Frontend, Tobias Backend und Christoph vermittler zwischen Front und Backen und ebenfalls Entwicklung in beiden Bereichen.

0.4 Komplettes Product-Backlog als Tabelle

- detail view
- Rezept suche
- Startseite
- Rezepte aus DB laden
- Rating System
- account seite
- bilder hochladen
- rezepte erstellen
- admin section
- session management
- "similar" recipies section - injection resistance
- user profile page
- Filter funktion
- Zutaten kaufen

0.5 Technologien/Requirements

- Clienttechnologie:
 - CSS
 - JS
 - jQuery
 - jQueryUI
 - Bootstrap o.ä als Addon-Frameworks
 - HTML5
- Servertechnologie:
 - PHP (8.0)
 - MariaDB (MySQL) kein PHP-Framework wie Symfony o.ä
 - nginx
- Funktionelle Vorgaben:
 - Sessionverwaltung
 - * Wir in der common.php datei (l.7) gestartet
 - * Session expire in (l. 9 - 17)
 - * Session abfrage z.B: in admin.php (l.14-19)
 - Login-Logout; automatisches Logout nach bestimmter Zeit
 - * Session expire in (l. 9 - 17)
 - Admin-Bereich mit Useraccount-Verwaltung
 - * In admin.php
 - * Nutzerverwaltung in den JavaScript Funktionen
 - Passwort-Verwaltung
 - * In admin.php (l.105-129)
 - Rechte-Verwaltung (Adminrechte-Userrechte)
 - * Die Rechteverwaltung wird hier mit in Sessionvariablen gespeicherten IDs geregelt. z.B: in admin.php (l.14-19)

- Dateneingaben client- und serverseitig prüfen (RegExp)
 - * Implementiert in common.php test_for_bad_chars, test_for_bad_chars_array.
Die Funktionen selbst sind in der datei common.php (l.341-345) und (l.347-353) deklariert.
- Datensatzmanipulation in SQL-Server (speichern; auslesen/ausgeben; bearbeiten; löschen)
 - * Implementiert an vielen stellen im Projekt. Die Rezepte, welche auf der Startseite ausgegeben werden, werden aus der Datenbank geladen
z.B: featured_recipies (l.5-8 ...), gespeichert in z.: admin.php (l.137-149), bearbeitet in z.B: admin.php (l.105-132), gelöscht in z.B: admin.php (l.72-99).
- Konfigurationsdaten via Konfigurationsdatei einlesen
 - * Eine Konfigurationsdatei wird mit (PHP INI) realisiert. Diese wird in common.php (l.20-55) geladen und deren Daten verwendet.
- Einbindung von jQuery und jQuery UI
 - * JQery und JQuery UI werden in der admin.php Datei verwendet, um z.B: einen Confirmation-Dialog zu realisieren, der den Nutzer fragt, ob eine aktion ausgefhrt werden soll, sowie auch um AJAX einzubinden.
- Dynamische laden/nachladen mit Hilfe von AJAX
 - * Realisiert in admin.php (siehe oben)
- Meldungsfenster und Userbestätigungen mit jQuery und jQuery UI
 - * Realisiert in admin.php (siehe oben)
- Datenexport via JSON
 - * Realisiert in get_user_data.php
- Dynamische Menüstruktur mit responsive Webdesign
 - * Auf der gesamten Seite vorhanden

0.6 Verzeichnisstruktur

Das Projekt hat mehrere Verzeichnisse, die nur für das Setup der Dockercontainer relevant sind. Diese sind die nginx, php, site.conf und dev Verzeichnisse. Der tatsächliche code der Webapp befindet sich im Verzeichnis webdir.

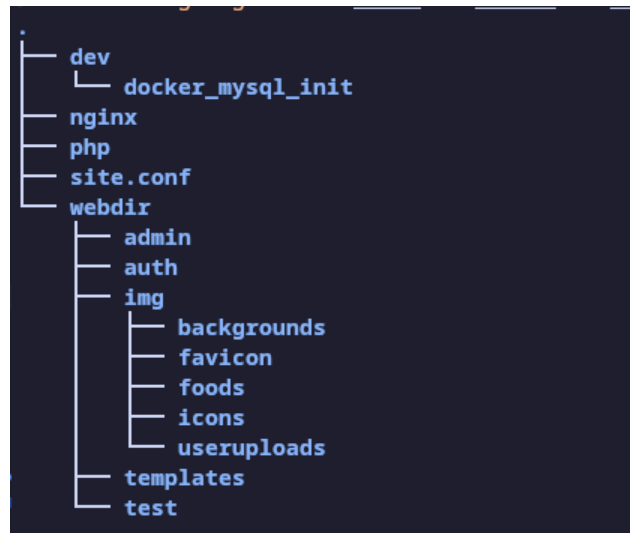


Figure 1: Directories

0.7 Beschreibung der Datenstrukturen und Tabellen

Für die Datenbank wurde nach Vorgabe MariDB verwendet. Diese Datenbank wurde in einem Dockercontainer gehostet und dann zur Bereitstellung der Daten angebunden.

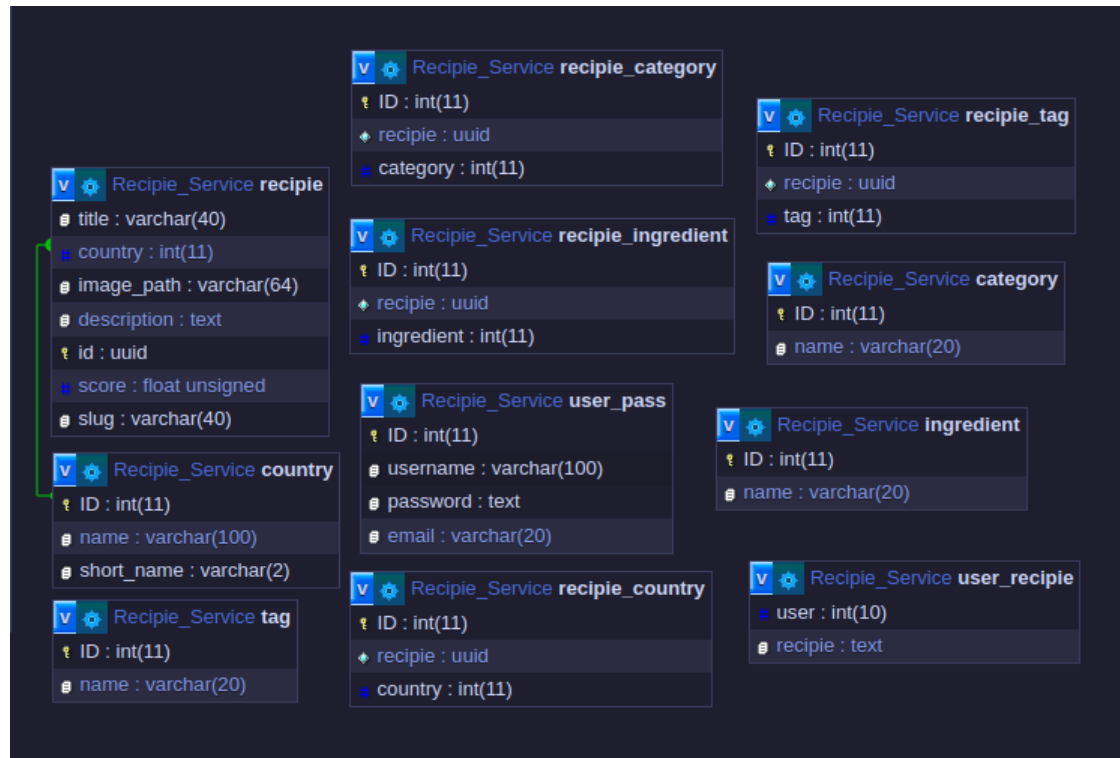


Figure 2: Datenbank Tabellen