

Erarbeitungs und Reflexionsphase

1- Technologieübersicht

Für die Entwicklung der Webanwendung KochFreunde wurden folgende Technologien und Werkzeuge gewählt:

- Python 3.x: Eine weit verbreitete, gut lesbare Sprache mit einem umfangreichen Ökosystem für die Webentwicklung und Datenverarbeitung.
- Flask: Ein leichtgewichtiges Web-Framework, das sich für prototypische bis mittlere Webanwendungen eignet; es hat eine einfache Lernkurve und ist gut erweiterbar.
- Flask-SQLAlchemy: Ein ORM zur komfortablen Modellierung der Datenbank, der Abfragen und Relationen vereinfacht.
- SQLite: Eine leichtgewichtige relationale Datenbank, die keine Serverkonfiguration erfordert; für die Produktion könnte beispielsweise PostgreSQL gewählt werden.
- Flask-WTF / WTForms: Zur Formularvalidierung und zum CSRF-Schutz.
- Flask-Login: Für die sitzungsbasierte Authentifizierung und Benutzerverwaltung.
- Pillow (PIL): Zur Bildverarbeitung (Vorschaubilder, Größenänderung) von Profil und Rezeptbildern.
- Bcrypt: Eine Bibliothek für sicheres Passwort Hashing.
- HTML / Jinja2: Für Frontend Templates.

Begründung: Diese Kombination ermöglicht ein schnelles Prototyping (Flask, SQLite) bei gleichzeitig guter Sicherheit (bcrypt, CSRF) und einer gut wartbaren Codebasis (ORM).

2- Architekturübersicht

Die Anwendung "KochFreunde" basiert auf einer dreischichtigen Architektur. Dieses Modell trennt die Anwendung in verschiedene logische Schichten, von denen jede eine spezifische Aufgabe hat. Dieser Ansatz fördert die Skalierbarkeit, Sicherheit und Wartbarkeit der Anwendung durch eine konsequente Trennung der Belange.

Beschreibung der Schichten

➤ Präsentationsschicht (Presentation Layer):

- Verantwortlichkeit: Diese Schicht ist für die Benutzeroberfläche und die Interaktion mit dem Benutzer zuständig. Sie ist der Teil, den der Benutzer sieht und mit dem er interagiert.

- Komponenten: Dazu gehören alle HTML-Templates (z. B. home.html, dashboard.html, recipe_detail.html) sowie die zugehörigen CSS und JavaScript Dateien. Sie verwendet die Jinja2-Template-Engine, um Daten, die von der Geschäftslogikschicht empfangen werden, dynamisch zu rendern.
 - Interaktion: Sie sendet Benutzeranfragen (wie Formulareingaben oder Klicks) an die Geschäftslogikschicht und zeigt die von diesen zurückerhaltenen Informationen an.
- Geschäftslogikschicht (Business Logic Layer):
- Verantwortlichkeit: Dies ist das Herzstück der Anwendung. Sie verarbeitet Benutzeranfragen, wendet die Geschäftsregeln der Anwendung an und fungiert als Vermittler zwischen der Präsentations- und der Datenschicht.
 - Komponenten: Diese Schicht wird hauptsächlich durch Ihre Datei app.py und die Datei forms.py repräsentiert. Sie enthält die Flask-Routen (@app.route), die Validierungslogik und alle Kernfunktionen, die z. B. das Hinzufügen von Rezepten, die Verwaltung von Benutzersitzungen oder das Hinzufügen/Entfernen von Favoriten steuern.
 - Interaktion: Sie empfängt Anfragen von der Präsentationsschicht, nutzt die Datenschicht zum Abrufen oder Speichern von Daten und sendet dann die verarbeiteten Informationen zur Anzeige an die Präsentationsschicht zurück.
- Datenzugriffsschicht (Data Access Layer):
- Verantwortlichkeit: Diese Schicht ist für die gesamte Datenspeicherung und abfrage verantwortlich. Sie stellt eine saubere Schnittstelle für die Geschäftslogikschicht bereit, um mit der Datenbank zu interagieren, ohne die Details des Datenbanksystems auf niedriger Ebene kennen zu müssen.
 - Komponenten: Sie wird durch Ihre Datei models.py und die SQLAlchemy-Bibliothek definiert. Sie enthält die Datenmodelle (User, Recipe, Comment, Rating) und die Logik für die Datenbankverbindung.
 - Interaktion: Sie empfängt Befehle von der Geschäftslogikschicht (z. B. "alle Rezepte abrufen", "einen neuen Benutzer speichern") und führt diese auf der SQLite Datenbank aus.

Diese Struktur ermöglicht eine saubere Trennung des Codes. Wenn Sie in Zukunft beispielsweise von SQLite zu PostgreSQL wechseln möchten, müssten Sie nur die Datenschicht ändern, ohne die Präsentation oder Geschäftslogikschicht zu beeinflussen.

3- Structure

Class Diagram

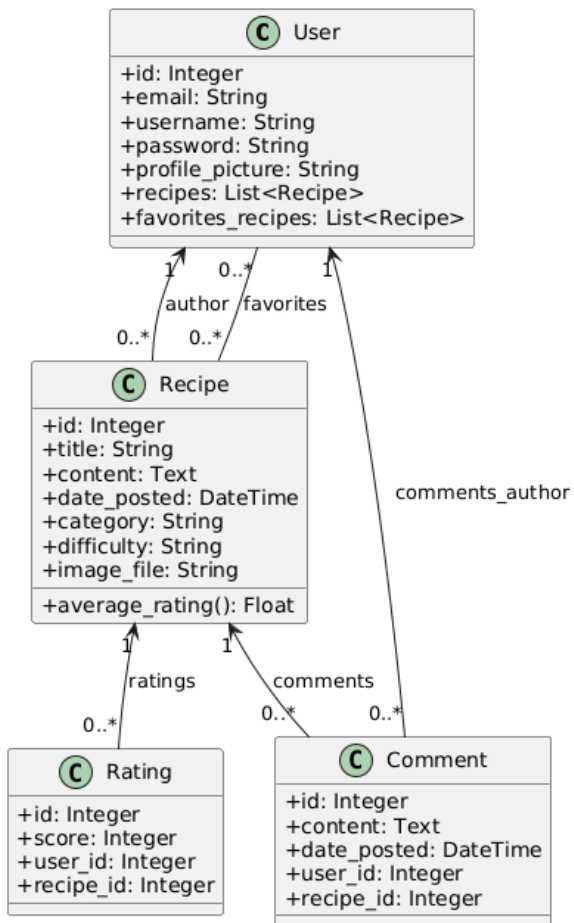


Abbildung 1:UML Class Diagram

Sequence Diagram

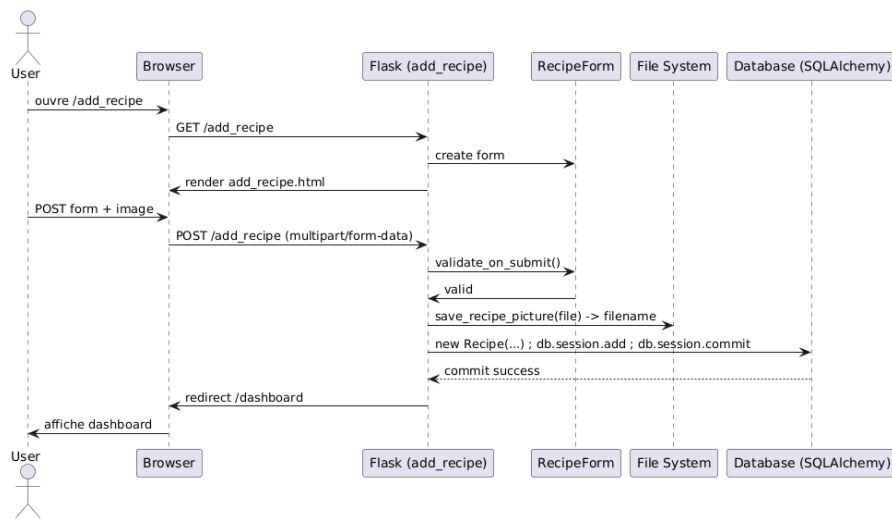


Abbildung 2: UML Sequence Diagram

4-Git

Der Quellcode für dieses Projekt befindet sich im folgenden GitHub-Repository.

<https://github.com/TaoubaMahmoudi/KochFreunde-Project>