



Portfolio

Projekt: Software Engineering

Weiterbildung: Software engineer- Python und
Künstlicher Intelligenz

August - Oktober 2025

Taouba Mahmoudi

Matrikelnummer: UP10744750

IU AKADEMIE



Inhalt

1. Konzeptionsphase	3
1.1 Projektdokumentation	3
1.1.1 Projektübersicht	3
1.1.2 Risikomanagement	4
1.1.3 Zeitplanung	5
1.2 Anforderungsdokument	5
1.2.1 Funktionale Anforderungen (F)	6
1.2.2 Nichtfunktionale Anforderungen (NF)	7
1.2.3 Glossar	8
1.3 Spezifikationsdokument	9
1.3.1 Datenmodell	9
1.3.2 Geschäftsprozesse	10
1.3.3 Geschäftsregeln	11
1.3.4 Schnittstellen	12
1.3.5 GUI-Beschreibung	12
2. Erarbeitungs und Reflexionsphase	13
2.1 Technologieübersicht	13
2.2 Architekturübersicht	14
2.3 Structure	16
2.4 Git	17
3. Finalisierungsphase	17
3.1 Test und Qualitätssicherungsphase	17
3.1.1 Teststrategie	17
3.1.2 Testfälle	18
3.1.3 Fazit der Qualitätssicherung	20
3.2 Reflexionsphase	20
3.2.1 Vergleich: Konzeption vs. Implementierung (Vergleich: Phase 1 vs. Phase 2)	20
3.2.2 Zeitmanagement und Einhaltung der Meilensteine	21
3.2.3 Herausforderungen und Lösungen	22
3.2.4 Kritische Würdigung	23
3.2.5 Wichtige Lektionen	23
3.2.6 Zukünftige Erweiterungen	24
3.3 Benutzerhandbuch	25
Fazit:	27

1.Konzeptionsphase

1.1 Projektdokumentation

1.1.1 Projektübersicht

Projekttitel: KochFreunde

Projektidee:

Ziel des Projekts ist die Entwicklung einer voll funktionsfähigen Webanwendung, in der Nutzer:innen eigene Rezepte mit Bildern, Zutaten und Zubereitungsschritten veröffentlichen können. Andere registrierte Nutzer:innen können diese Rezepte ansehen, kommentieren, bewerten und als Favoriten speichern.

Die Anwendung bietet folgende Hauptfunktionen:

- Registrierung und Login von Nutzer:innen
- Rezeptverwaltung
- Kommentarsystem
- Bewertungssystem (1–5 Sterne)
- Favoriten Funktion
- Filterbare Rezeptsuche (z. B. nach vegetarisch, glutenfrei, Schwierigkeit)

Zielgruppe sind Hobbyköch:innen, Familien, Studierende und alle Kochbegeisterten, die ihre Lieblingsrezepte teilen oder neue entdecken möchten.

Projektlaufzeit:

12.08.2025 – 31.10.2025

Meilensteine:

- M1 (03.09.2025): Konzept freigegeben
- M2 (20.10.2025): Implementierung abgeschlossen
- M3 (31.10.2025): Release 1.0 veröffentlicht

Projektstruktur (PSP-Codes):

- Phase 1 Konzeption: Projektauftrag, Anforderungen, UML-Entwürfe, Risikomanagement, Projektplan
- Phase 2 Implementierung: Backend (Flask, SQLite), Frontend (HTML/CSS/Bootstrap), Integration (Controller, Interfaces), Persistenzschicht
- Phase 3 Tests & Abschluss: Unit- und Integrationstests (pytest), Qualitätssicherung, Release-Paket

Werkzeuge und Technologien:

- Backend: Python (Flask), SQLite (Persistenz)
- Frontend: HTML, CSS, Bootstrap
- Diagramme & Planung: draw.io / Lucidchart (UML, Systemkontext), matplotlib (GANTT)
- Tests: pytest (automatisierte Testfälle)
- Versionskontrolle: GitHub

1.1.2 Risikomanagement

Risiko	Eintrittswahrsch	Auswirkung	Gegenmassnahmen	Frühwarnindikatoren
Authentifizierung fehlerhaft implementiert	Mittel	Hoch	Login System mit bekannten Frameworks bauen	Häufige Login-Fehlversuche bei Tests, fehlende Unit-Tests für Authentifizierung
Komplexität zu groß Features unvollständig	Hoch	Mittel	MVP-Ansatz: zuerst Basisfunktionen liefern	Verzögerungen bei Zwischen-Deliverables, viele offene Anforderungen kurz vor Abgabe
Zeitüberschreitung durch Nebentätigkeiten	Mittel	Mittel	Zeitpuffer einplanen, klare Deadlines setzen	Nichterreichen von Meilensteinen, steigende Überstunden
Datenverlust oder SQL Fehler	Niedrig	Hoch	Datenbank-Tests und Validierungen durchführen	Vermehrte DB-Fehlermeldungen in Logs, lange Antwortzeiten bei Abfragen

1.1.3 Zeitplanung

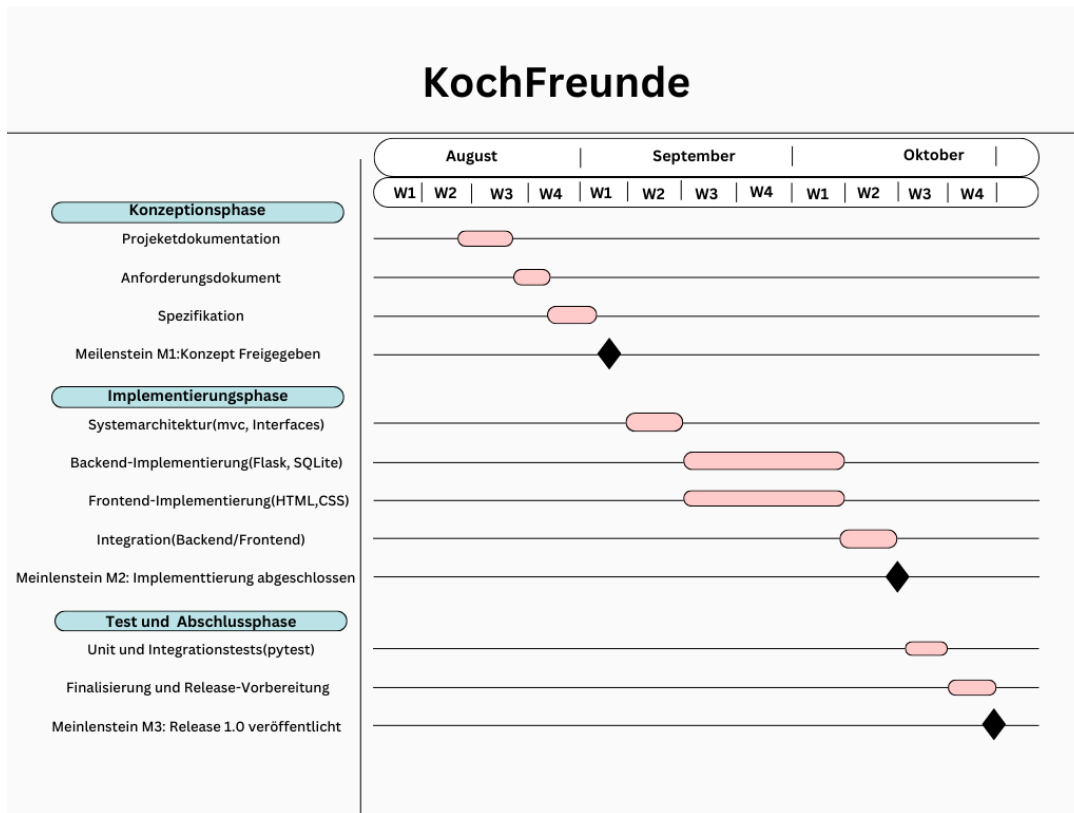


Abbildung 1: Gantt Diagram

1.2 Anforderungsdokument

➤ Stakeholder

✓ Registrierte Nutzer:innen

- Beschreibung: Die Hauptzielgruppe des Projekts. Dazu gehören Hobbyköche, Familien und Studierende, die die Anwendung aktiv nutzen werden, um Rezepte zu finden, zu teilen und zu bewerten.
- Interessen: Eine intuitive und benutzerfreundliche Anwendung, die schnell und zuverlässig ist. Sie legen Wert auf eine einfache Navigation, klare Funktionen und die Möglichkeit, ihre Lieblingsrezepte zu speichern.

✓ Besucher (Nicht-registrierte Nutzer:innen)

- Beschreibung: Personen, die die Website besuchen, aber noch kein Konto erstellt haben. Sie sind potenziell zukünftige Nutzer.
- Interessen: Sie möchten die Anwendung erkunden und Inhalte wie Rezepte, Bilder, Kommentare und Bewertungen ansehen. Eine

schnelle, ansprechende und einfach zu durchsuchende Benutzeroberfläche ist für sie entscheidend, um einen positiven ersten Eindruck zu gewinnen und sich möglicherweise zu registrieren.

✓ Projektteam

- Beschreibung: Das Team, das für die Entwicklung und Gestaltung der Webanwendung verantwortlich ist.
- Interessen: Sie streben klare und präzise Anforderungen an, um eine reibungslose Implementierung zu gewährleisten. Sie sind daran interessiert, eine robuste und wartbare Architektur zu schaffen.

✓ Systemadministrator

- Beschreibung: Die Person, die für die technische Infrastruktur, das Hosting und die Wartung der Anwendung auf dem Server verantwortlich ist.
- Interessen: Er benötigt eine stabile und sichere Anwendung, die einfach zu installieren und zu überwachen ist. Er legt Wert auf eine gute Performance und minimale Betriebsstörungen.

1.2.1 Funktionale Anforderungen (F)

Nr.	Anforderung	Kriterium zur Erfüllung
F1	Besucher:innen können sich registrieren.	Registrierung mit E-Mail & Passwort möglich; Konto wird in DB angelegt.
F2	Nutzer:innen können sich einloggen.	Nach korrektem Login Zugriff auf Profil; falsche Daten → Fehlermeldung.
F3	Nutzer:innen können eigene Rezepte anlegen, bearbeiten und löschen (CRUD).	Pflichtfelder (Titel, Zutaten, Schritte) werden geprüft; Daten korrekt in DB gespeichert.
F4	Nutzer:innen können Rezepte bewerten (1–5 Sterne).	Bewertung wird gespeichert; Durchschnittswert automatisch aktualisiert.
F5	Nutzer:innen können Rezepte kommentieren.	Kommentar erscheint sofort unter dem Rezept; max. Länge 300 Zeichen.
F6	Nutzer:innen können Rezepte als Favoriten speichern und abrufen.	Favorit wird gespeichert und erscheint auf Profilseite.
F7	Nutzer:innen können Rezepte nach Kategorien/Ernährung filtern.	Filter liefert nur passende Ergebnisse (z. B. vegetarisch).

Use Case Diagram :

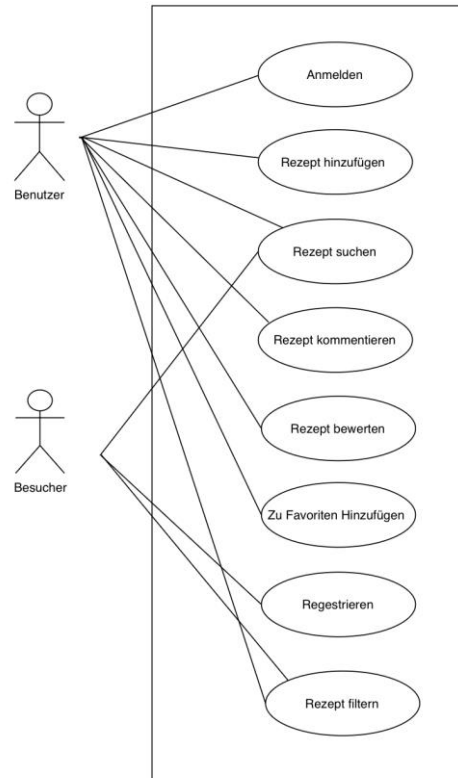


Abbildung 2:Case Diagram

1.2.2 Nichtfunktionale Anforderungen (NF)

Nr.	Anforderung	Kriterium zur Erfüllung
NF1 – Sicherheit	Passwörter müssen sicher gespeichert werden.	Speicherung ausschließlich gehasht (bcrypt, ≥ 12 Runden).
NF2 – Performance	Rezepte-Suche muss schnell reagieren.	Antwortzeit ≤ 2 Sekunden bei 1000 Datensätzen.
NF3 – Benutzerfreundlichkeit	Navigation soll einfach sein.	Rezept ist mit max. 3 Klicks auffindbar.

NF4 – GUI-Klarheit	Oberfläche muss übersichtlich sein.	Hauptfunktionen (Login, Suche, Rezept hinzufügen) direkt von Startseite erreichbar.
NF5 – Responsives Design	Anwendung muss mobil nutzbar sein.	Darstellung korrekt bei Bildschirmbreite $\geq 320\text{px}$.
NF6 – Erweiterbarkeit	System soll leicht erweitert werden können.	Neue Module (z. B. Video-Upload) integrierbar ohne Änderung am Kernsystem.

1.2.3 Glossar

- Agile: Eine Methode zur Projektverwaltung, die sich auf die iterative Entwicklung.
- Anforderung: Eine Beschreibung eines Merkmals, das ein System haben muss, um einen Zweck oder ein Ziel zu erfüllen.
- API (Application Programming Interface): Eine Schnittstelle, die es verschiedenen Softwarekomponenten ermöglicht, miteinander zu kommunizieren.
- Architektur: Die grundlegende Struktur und Organisation eines Softwaresystems.
- Backend: Die serverseitige Logik der Anwendung, die für die Verarbeitung der Daten, die Datenbankverwaltung und die Authentifizierung zuständig ist.
- Bewertung: Ein Punktesystem (1-5 Sterne) zur Beurteilung eines Rezepts.
- bcrypt: Ein sicherer Hashing-Algorithmus, der speziell zum Schutz von Passwörtern entwickelt wurde.
- Client-Server-Modell: Ein Netzwerkarchitekturmodell, bei dem eine Anwendung in zwei Teile aufgeteilt ist: den Client (Frontend) und den Server (Backend).
- CSS (Cascading Style Sheets): Eine Stylesheet-Sprache, die das Aussehen und die Formatierung von Webseiten beschreibt.
- Favorit: Ein vom Nutzer gespeichertes Rezept zur späteren Nutzung.
- Filter: Ein Kriterium, um eine Suche nach Rezepten einzuschränken (z. B. "vegetarisch", "glutenfrei").
- Flask: Ein leichtgewichtiges Web-Framework für Python, das für die Entwicklung des Backends verwendet wird.
- Frontend: Der Teil der Anwendung, mit dem Nutzer:innen direkt interagieren, einschließlich der Benutzeroberfläche.
- Gantt-Diagramm: Ein Balkendiagramm, das die zeitliche Planung eines Projekts visualisiert.

- GUI (Graphical User Interface): Die grafische Benutzeroberfläche einer Anwendung, bestehend aus visuellen Elementen.
- HTML (HyperText Markup Language): Die Auszeichnungssprache, die zur Strukturierung von Webinhalten verwendet wird.
- Meilenstein: Ein wichtiger Kontrollpunkt in der Projektzeitlinie, der das Ende einer Phase oder das Erreichen eines wichtigen Ziels markiert.
- MVC (Model-View-Controller): Ein Architekturmuster, das eine Anwendung in die drei Bereiche Modell (Daten), Ansicht (Benutzeroberfläche) und Controller (Logik) trennt.
- Nutzerkonto: Ein persönliches Konto, das durch eine E-Mail-Adresse und ein Passwort geschützt ist.
- PSP-Code (Projektspezifische-Pakete): Ein Kodierungssystem, das zur Strukturierung der Arbeitspakete in einem Projekt dient.
- pytest: Ein Python-Framework zur Erstellung und Ausführung von automatisierten Tests.
- Rezept: Eine Sammlung von Zutaten, Zubereitungsschritten und einem oder mehreren Bildern.
- Responsives Design: Ein Ansatz im Webdesign, der sicherstellt, dass sich die Benutzeroberfläche an verschiedene Bildschirmgrößen anpasst.
- SQLite: Ein leichtes, dateibasiertes Datenbank-Managementsystem, das in die Anwendung integriert wird.
- Stakeholder: Eine Person oder Gruppe, die ein Interesse an oder Einfluss auf das Projekt hat.
- UML (Unified Modeling Language): Eine standardisierte Sprache zur Visualisierung, Spezifikation, Konstruktion und Dokumentation von Softwaresystemen.

1.3 Spezifikationsdokument

1.3.1 Datenmodell

Class Diagram :

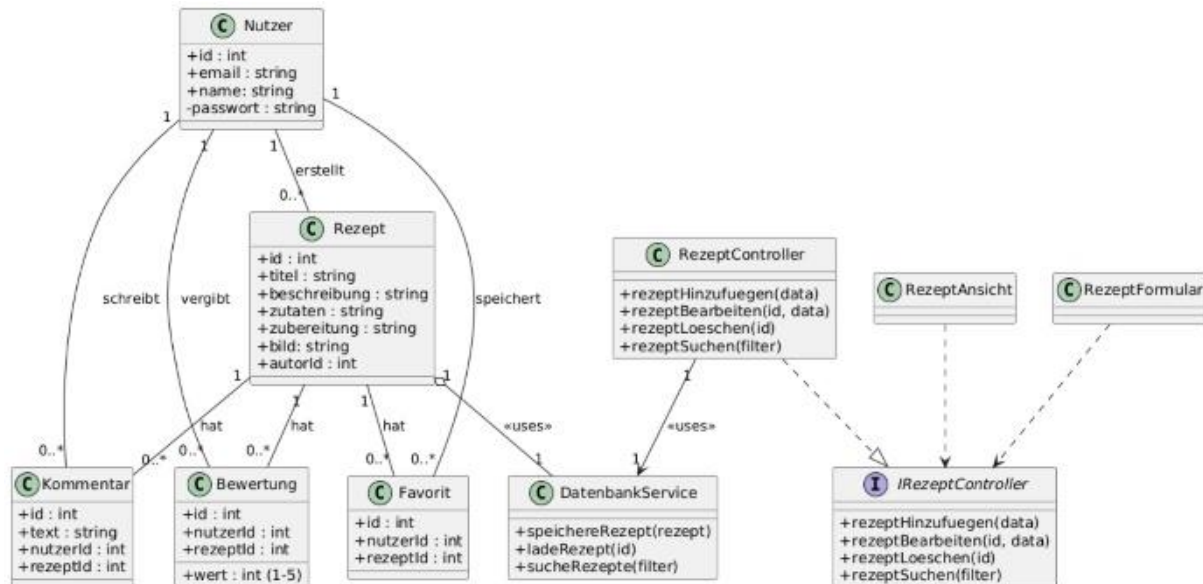


Abbildung 3: Class Diagram

1.3.2 Geschäftsprozesse

Rezept erstellen

1. Der/die Nutzer:in klickt auf "Rezept hinzufügen" im Hauptmenü.
2. Ein Eingabeformular wird angezeigt mit Feldern für Titel, Beschreibung, Zutatenliste, Zubereitungsschritte, Kategorie (z. B. vegetarisch) und Bild.
3. Der/die Nutzer:in füllt die Felder aus und lädt optional ein Bild hoch.
4. Beim Absenden wird das Formular validiert.
5. Das Rezept wird in der Datenbank gespeichert und dem Profil des Nutzers zugeordnet.
6. Eine Bestätigungsmeldung wird angezeigt und das Rezept erscheint auf der Startseite.

Nutzer registrieren

1. Der/die Besucher:in klickt auf "Registrieren".
2. Es erscheint ein Formular zur Eingabe von Namen, E-Mail-Adresse und Passwort.
3. Die Eingaben werden geprüft (z. B. Mindestlänge, E-Mail Format).
4. Wenn alles gültig ist, wird das Passwort gehasht und der Nutzeraccount in der Datenbank gespeichert.

5. Der/die Nutzer:in wird automatisch eingeloggt und zur Startseite weitergeleitet.

Rezept bewerten

1. Der/die eingeloggten Nutzer: in öffnet die Detailseite eines Rezepts.
2. Er/sie wählt eine Bewertung von 1 bis 5 Sternen aus.
3. Die Auswahl wird übermittelt und in der Datenbank gespeichert, zusammen mit der Nutzer-ID.
4. Die durchschnittliche Bewertung des Rezepts wird aktualisiert und angezeigt.

Kommentar schreiben

1. Der/die eingeloggten Nutzer: in sieht unter einem Rezept das Kommentarfeld.
2. Er/sie schreibt einen Text und klickt auf "Kommentar posten".
3. Der Text wird geprüft (z. B. max. Länge).
4. Der Kommentar wird mit Nutzer-id gespeichert.
5. Er erscheint sofort unter dem Rezept mit dem Namen des Nutzers.

Activity diagram :

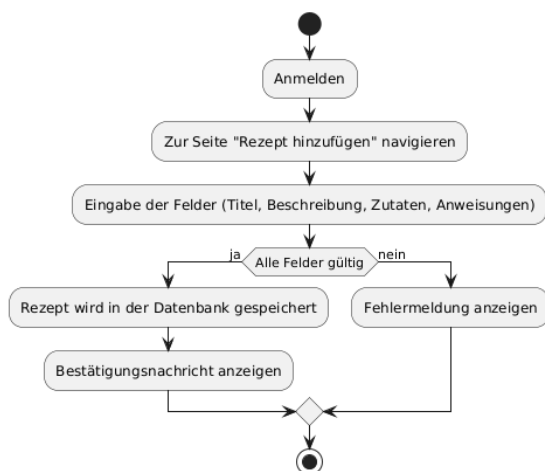


Abbildung 4: Activity Diagram

1.3.3 Geschäftsregeln

- Passwortlänge: Ein Passwort muss mindestens 8 Zeichen lang sein.
- Rezeptbewertung: Jeder angemeldete Nutzer darf jedes Rezept nur einmal bewerten.

- Rezeptinhalt: Ein Rezept muss mindestens eine Zutat und einen Zubereitungsschritt enthalten.
- Kommentarlänge: Kommentare dürfen maximal 300 Zeichen umfassen.

1.3.4 Schnittstellen

- Frontend zu Backend: Diese Schnittstelle ist entscheidend für die Interaktion mit den Nutzer:innen.
 - Technologie: Sie verwenden REST (Representational State Transfer).
 - Kommunikation: Daten werden im JSON-Format zwischen dem Frontend und dem Backend ausgetauscht.
 - Anwendung: Diese Schnittstelle ermöglicht es der grafischen Benutzeroberfläche (GUI) im Browser, Daten wie Rezepte und Kommentare vom Server abzurufen und zu speichern.
- Backend zur Datenbank: Dies ist die Schnittstelle für die Persistenz (Speicherung) der Daten.
 - Technologie: Sie verwenden die Datenbank SQLite.
 - Kommunikation: Das Backend kommuniziert mit der Datenbank, um Rezepte, Nutzerprofile, Kommentare und Bewertungen zu speichern und abzurufen.
- Optionale externe API-Schnittstelle: Dies ist eine zukünftige Schnittstelle, die das System erweiterbar macht.
 - Zweck: Sie könnte in Zukunft verwendet werden, um Daten von externen Diensten zu integrieren.

1.3.5 GUI-Beschreibung

- Startseite: Top-Rezepte, Suche, Filter, Login/Registrieren
- Login/Register: Eingabefelder + Validierung
- Profilseite: Favoriten, eigene Rezepte
- Rezeptansicht: Bild, Bewertung, Kommentare
- Rezeptformular: Eingabe + Upload



Abbildung 5: Startseite

2. Erarbeitungs und Reflexionsphase

2.1 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).

2.2 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.

2.3 Structure

Class Diagram

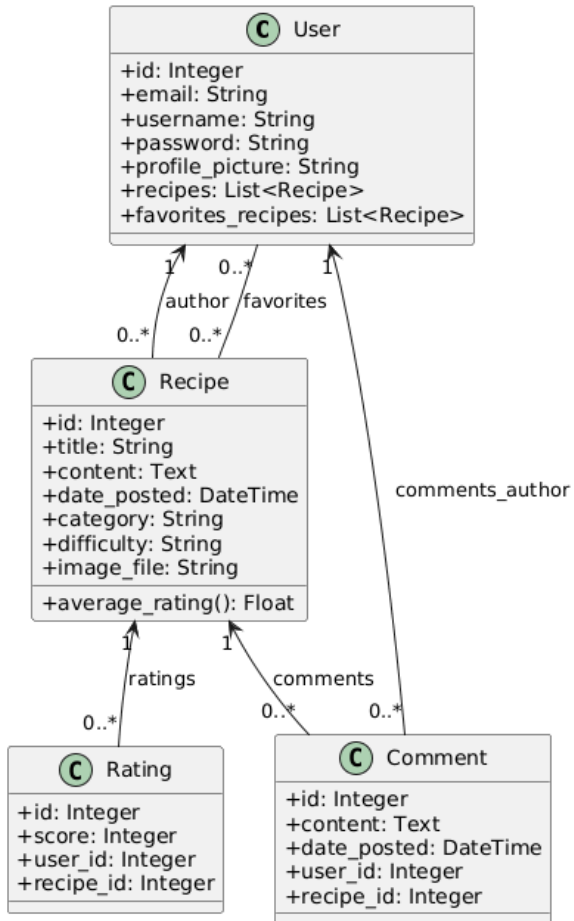


Abbildung 6:UML Class Diagram

Sequence Diagram

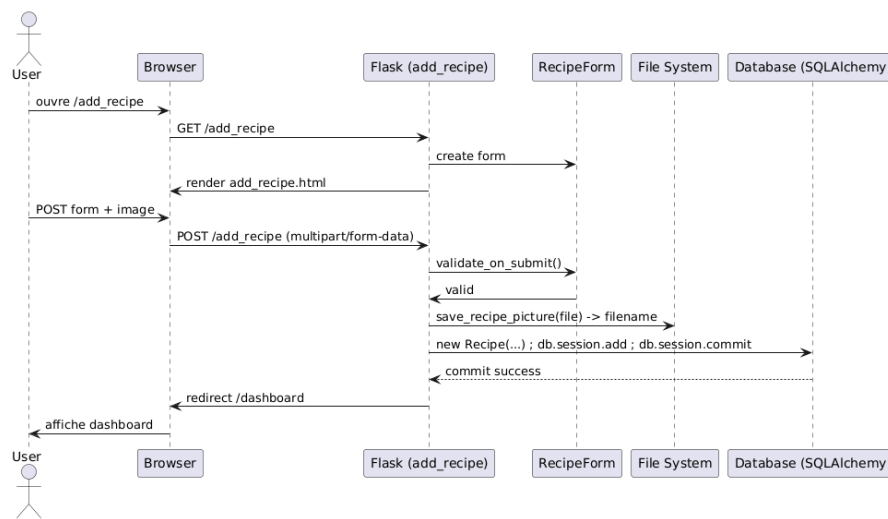


Abbildung 2: UML Sequence Diagram

2.4 Git

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

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

3.Finalisierungsphase

3.1 Test und Qualitätssicherungsphase

Die Testphase dient dazu, die Funktionsfähigkeit, Zuverlässigkeit und Benutzerfreundlichkeit der Anwendung "KochFreunde" gemäß den definierten Anforderungen zu überprüfen.

3.1.1 Teststrategie

Als Teststrategie wird eine Kombination aus Funktionstests (Black-Box) und Benutzbarkeits-Tests (Usability-Tests) gewählt.

- Funktionstests (Black-Box-Tests): Überprüfung, ob jede einzelne Funktion (z. B. Registrierung, Rezept hinzufügen, KI-Suche) wie erwartet funktioniert und die richtigen Daten verarbeitet.

- Benutzbarkeits-Tests: Überprüfung, ob die Anwendung intuitiv ist und die Benutzerziele effizient erreicht werden können (z. B. Navigation, Verständlichkeit der Formulare).

3.1.2 Testfälle

Die folgenden Testfälle decken die Kernfunktionen der Webanwendung ab.

Test-id	Modul	Testbeschreibung	Erwartetes Ergebnis	Status (ok / fehler)
A1	Authentifizierung	Registrierung eines neuen Benutzers mit gültigen Daten.	Der Benutzer wird registriert und zum Dashboard weitergeleitet.	ok
A2	Authentifizierung	Versuch der Anmeldung mit ungültigem Passwort.	Fehlermeldung wird angezeigt; Anmeldung wird verweigert.	ok
F1	Formularvalidierung (Flask-WTF)	Überprüfung der Validierung des Registrierungsformulars. Absenden des Formulars mit einer fehlerhaft formatierten E-Mail ("test@invalide").	Die Validierung muss False zurückgeben und die entsprechende Fehlermeldung generieren.	ok
F2	Formularvalidierung	Überprüfung der Pflichtfelder im Rezeptformular. Absenden des Formulars, wobei das Feld "Zubereitung" (Anweisungen) leer bleibt.	Die Validierung muss False zurückgeben. Der Benutzer bleibt auf der Formularseite, und die Fehlermeldung "Dieses Feld ist erforderlich." wird neben dem leeren Feld angezeigt.	ok
D1	Dashboard/Profil	Zugriff auf das Dashboard als	Dashboard wird mit den richtigen Links	ok

		angemeldeter Benutzer.	(Profil, Rezept hinzufügen) angezeigt.	
D2	Dashboard/Profil	Anzeige des Profils	Veröffentlichte Rezepte und Favoriten werden korrekt aufgelistet.	ok
D3	Datenmodell (Rating/Recipe)	Berechnung der durchschnittlichen Bewertung einer Rezeptinstanz mit 3 Bewertungen (5, 3, 4).	Die Methode <code>average_rating()</code> muss 4,0 zurückgeben.	ok
R1	Rezept-CRUD	Bearbeiten eines eigenen Rezepts	Die Änderungen werden gespeichert und im Detail-Template angezeigt.	ok
R2	Rezept-CRUD	Hinzufügen eines neuen Rezepts (Formular) mit Bild-Upload.	Rezept wird in der Datenbank gespeichert und ist auf der Hauptseite/Rezeptliste sichtbar.	ok
S1	Suche	Globale Suche nach einem bestimmten Stichwort (z. B. "Pasta").	Es werden alle Rezepte angezeigt, die das Stichwort im Titel enthalten.	ok
KI1	KI-Suche (Externe API)	Suche in der KI-Rezeptsuche mit mehreren Zutaten	Es werden passende Rezepte von der Spoonacular API abgerufen. (<i>Test de récupération</i>)	ok
KI2	KI-Suche (Detail)	Aufruf der Detailseite eines KI-Rezepts	Titel, Bild, Zutaten und Zubereitung werden korrekt dargestellt.	ok

KI3	Funktionstest KI-Detailansicht (Übersetzung)	Aufruft der Detailseite eines KI-Rezepts auf Deutsch	Die Rezepthinweise werden in der Originalsprache (Englisch) angezeigt.	Fehler
I1	Interaktion	Hinzufügen eines Rezepts zu den Favoriten	Das Rezept erscheint in der Liste der Favoriten auf der Profilseite.	ok
I2	Interaktion	Posten eines neuen Kommentars unter einem Rezept.	Der Kommentar wird sofort unter dem Rezept angezeigt.	ok
I3	Interaktion	Abgabe einer 5-Sterne-Bewertung	Die Bewertung wird gespeichert, und die Durchschnittsbewertung wird auf der Detailseite aktualisiert.	ok

3.1.3 Fazit der Qualitätssicherung

Die durchgeführten Tests bestätigen, dass die Kernfunktionalitäten der Webanwendung "KochFreunde" den anfänglich definierten Anforderungen entsprechen. Das System ist stabil, die Authentifizierungs und Datenverwaltungsmechanismen (CRUD) sind funktionsfähig und die Integration der externen KI-Rezeptsuche (Spoonacular-API) ist erfolgreich. Die Benutzeroberfläche ist konsistent und intuitiv bedienbar.

3.2 Reflexionsphase

3.2.1 Vergleich: Konzeption vs. Implementierung (Vergleich: Phase 1 vs. Phase 2)

Dieser Abschnitt analysiert die Übereinstimmung zwischen dem anfänglichen Konzept (Phase 1) und dem tatsächlich implementierten Produkt (Phase 2) unter Berücksichtigung strategischer Anpassungen während der Entwicklung.

Merkmal	Konzeption (Phase 1)	Implementierung (Phase 2)	Abweichung / Begründung
Datenbank	SQLite	SQLite	Beibehalten für Prototyping und schnelle Entwicklung. PostgreSQL bleibt die Option für die Produktivsetzung/Skalierbarkeit.
Rezeptsuche	Interne Suche (Titel, Inhalt)	Interne Suche + KI-Suche (Spoonacular)	Erweiterung. Die KI-Suche wurde in Phase 2 hinzugefügt, um den Mehrwert der Anwendung über die ursprünglichen Anforderungen hinaus zu steigern.
Übersetzung KI-Rezepte	Nicht erwähnt	Geplant: Automatische Übersetzung (Englisch → Deutsch)	Funktionsfehler. Die Implementierung scheiterte aufgrund der Authentifizierungs-/Abrechnungsanforderungen der Google Cloud Translation API.
CRUD & Authentifizierung	Alle Funktionen geplant	Alle Funktionen implementierten	Vollständige und erfolgreiche Implementierung aller Authentifizierungs, Favoriten, Bewertungs und Kommentarfunktionen.

3.2.2 Zeitmanagement und Einhaltung der Meilensteine

Die Einhaltung des Zeitplans war ein zentrales Ziel des Projekts und wurde erfolgreich umgesetzt. Das gesamte Projekt wurde im geplanten Zeitraum vom 12.08.2025 bis 31.10.2025 durchgeführt.

Alle definierten Meilensteine wurden fristgerecht erreicht:

- Meilenstein M1 (03.09.2025) (Konzept freigegeben): Die Dokumentation der Konzeptionsphase wurde pünktlich fertiggestellt und genehmigt.

- Meilenstein M2 (20.10.2025) (Implementierung abgeschlossen): Die gesamte Backend- und Frontend-Entwicklung sowie die Integration aller Kernfunktionen wurden planmäßig beendet.
- Meilenstein M3 (31.10.2025) (Release 1.0 veröffentlicht): Die abschließenden Tests, die Qualitätssicherung und die Erstellung der finalen Portfoliodokumentation wurden erfolgreich bis zum Enddatum abgeschlossen.

Trotz technischer Schwierigkeiten konnte der Zeitplan dank einer effizienten Priorisierung der Aufgaben eingehalten werden.

3.2.3 Herausforderungen und Lösungen

Während der Entwicklung traten kritische technische Herausforderungen auf.

Herausforderung 1: Unüberwindbare Blockade der zuverlässigen automatischen Übersetzung

Dieser technische Engpass erforderte eine strategische Neuausrichtung, um die KI-Suchfunktion zu retten.

- Ausgangsproblem: Das Ziel war die zuverlässige automatische Übersetzung externer Rezepte (Titel, Anweisungen) von der Spoonacular API von Englisch nach Deutsch mithilfe der Google Cloud Translation API.
- Blockadeursache (Authentifizierung/Abrechnung): Die Integration scheiterte, da Google Cloud für die Aktivierung der Translation API zwingend die Hinterlegung einer Zahlungsmethode (Kreditkarte) verlangt, obwohl die Nutzung unterhalb der Freigrenze lag. Diese Anforderung war für ein Hochschulprojekt inakzeptabel.
- Gescheiterte Alternativen: Versuche mit kostenlosen Web-Scraping-Bibliotheken (googletrans, translate mit mymemory) scheiterten sofort aufgrund von Instabilität, Inkompatibilitätsfehlern (ModuleNotFoundError: No module named 'cgi') und den inhärenten Ratenbegrenzungen dieser inoffiziellen Dienste.
- Angenommene Lösung (Strategischer Pivot): Um das Projekt innerhalb der Frist (Meilenstein M2) freizugeben, wurde die Strategie angepasst. "KochFreunde" zeigt die von der KI gefundenen Rezepte in der Originalsprache (Englisch) an. Das Frontend bleibt deutschsprachig, aber der API-importierte Inhalt wird unverändert präsentiert, wodurch der Zugang zur KI-Suchfunktionalität trotz des externen Hindernisses gesichert wurde.

Herausforderung 2: Verwaltung komplexer Datenbankbeziehungen (Many-to-Many)

- Problem: Die Funktionen Favoriten und Bewertungen erforderten komplexe Many-to-Many-Beziehungen mit Zusatzdaten (z. B. der score der Bewertung).
- Lösung: Dies wurde durch die Erstellung dedizierter Assoziationstabellen (z. B. Rating und Favorite über die secondary-Relation von Flask-SQLAlchemy) gelöst, was eine saubere, leistungsfähige und wartbare Implementierung der Datenbankabfragen ermöglichte.

Diese Erfahrungen stärkten meine Fähigkeit, flexibel auf technische Herausforderungen zu reagieren und pragmatische Lösungen zu finden.

3.2.4 Kritische Würdigung

Die kritische Würdigung umfasst sowohl die Bewertung der gewählten Technologien als auch des Projektmanagements, um die Gesamtqualität und Effizienz des Entwicklungsprozesses zu reflektieren.

Bewertung der Technologie-Wahl: Der gewählte Technologie-Stack (Python/Flask, Jinja2, SQLite) erwies sich als äußerst geeignet für dieses Portfolio-Projekt. Die Nutzung von Flask ermöglichte ein schnelles Prototyping und eine klare Strukturierung der Logik, während Flask-SQLAlchemy die Datenbankverwaltung erheblich vereinfachte. Für eine Anwendung mit hohem Traffic und umfangreichen Daten wäre jedoch der Wechsel der Datenbank von SQLite zu PostgreSQL oder MariaDB zu empfehlen, um die Skalierbarkeit zu gewährleisten.

Bewertung des Projektmanagements: Das Projektmanagement war durch die Einhaltung des Zeitplans und die erfolgreiche Bewältigung kritischer technischer Herausforderungen gekennzeichnet. Die Fähigkeit, auf den Blockadefehler der Übersetzungs-API mit einem strategischen Pivot (Anzeige der Originalsprache) zu reagieren, ohne den Endtermin zu verpassen, ist ein Beweis für die erfolgreiche Risiko und Problembewältigung.

3.2.5 Wichtige Lektionen

Die Durchführung des Projekts KochFreunde lieferte wertvolle Erkenntnisse für zukünftige Entwicklungsarbeiten:

- Risikobewertung externer Abhängigkeiten: Die Erfahrung mit der Google Cloud Translation API zeigte, dass selbst scheinbar stabile, kostenfreie Dienste unvorhergesehene finanzielle oder Authentifizierungs-Barrieren aufweisen können. Lektion: Die Machbarkeit und die genauen Anforderungen externer APIs müssen

frühzeitig durch eine vollständige Authentifizierung und eine gründliche Prüfung der Nutzungsbedingungen validiert werden.

- Wert eines strategischen Pivots: Der erzwungene Verzicht auf die automatische Übersetzung demonstrierte die Bedeutung der Fähigkeit, einen strategischen Pivot durchzuführen. Lektion: Bei kritischen Fehlern in einer Sekundärfunktion ist es besser, die Primärfunktion (die KI-Suche) mit einer funktionalen Einschränkung (keine Lokalisierung) zu liefern, als das gesamte Projekt zu verzögern oder die gesamte Funktionalität aufzugeben.
- Vorteil der modularen Architektur: Die Dreischicht-Architektur erwies sich als entscheidend für die Stabilität des Projekts. Lektion: Die klare Trennung von Präsentations, Geschäftslogik- und Datenschicht ermöglichte es, das Problem zu isolieren, ohne die gesamte Benutzeroberfläche oder die Datenbankstruktur zu destabilisieren.

3.2.6 Zukünftige Erweiterungen

Um die Benutzererfahrung zu vertiefen und die Reichweite der Anwendung zu erhöhen, wurden folgende Erweiterungen für zukünftige Versionen identifiziert:

- **Erweiterte Rezeptinformationen:**

Hinzufügen eines Feldes für die Portionsgröße (z. B. "4 Personen") und die Gesamtkochzeit (Vorbereitungs- + Kochzeit) zum Rezeptformular, um die Benutzerfreundlichkeit zu erhöhen.

- **Soziale Medien-Integration (API-Erweiterung):**

Integration von APIs wie Facebook oder Instagram, um Nutzer:innen das direkte Teilen von Rezepten in ihren sozialen Netzwerken zu ermöglichen. Dies würde die Sichtbarkeit von KochFreunde deutlich erhöhen.

- **Verbesserte Such und Filterfunktionen:**

Hinzufügen von Filteroptionen auf der Hauptseite, basierend auf den neuen Attributen Portionsgröße und Kochzeit.

- **Erweiterte Kommentarverwaltung:**

Implementierung von Funktionen zum Bearbeiten oder Löschen der eigenen Kommentare durch den Benutzer.

3.3 Benutzerhandbuch

Quick Start Guide (Applikation starten)

1. Ordner vorbereiten: Öffnen Sie das Terminal im Stammverzeichnis des Projekts (dem Ordner, der die requirements.txt-Datei enthält).
2. Virtuelle Umgebung erstellen und aktivieren: `python -m venv venv /.`
`\venv\Scripts\activate`
3. Abhängigkeiten installieren: `pip install -r requirements.txt`
4. Anwendung starten: `flask run`
5. Im Browser öffnen: Navigieren Sie zu <http://127.0.0.1:5000/>

Die folgenden Abbildungen zeigen die wichtigsten Ansichten der Anwendung KochFreunde aus Benutzersicht.

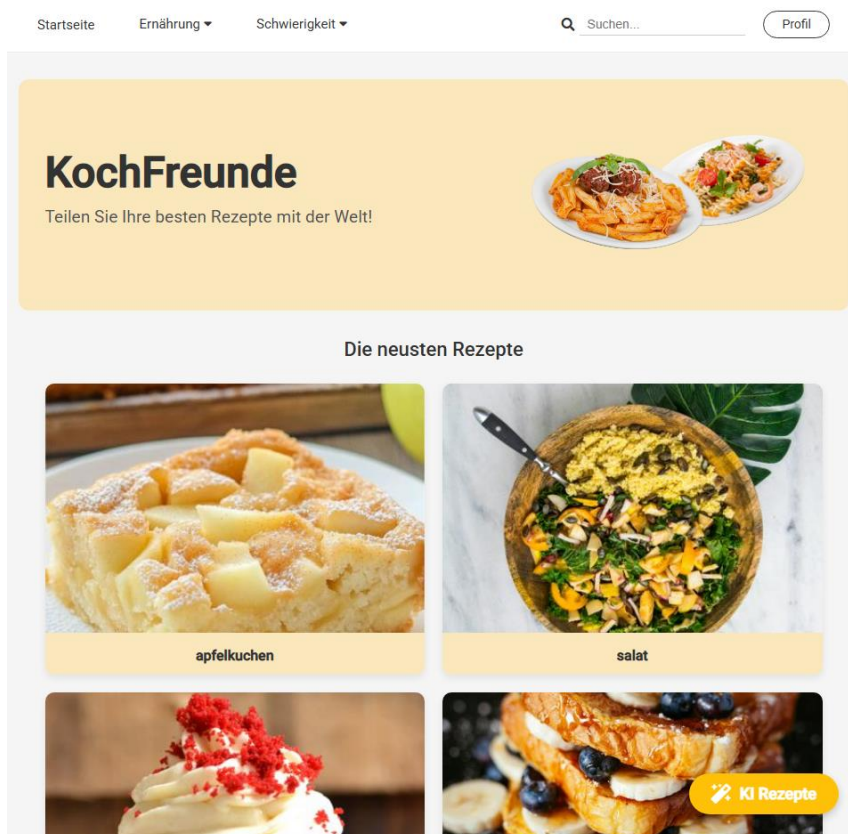


Abbildung 7: Startseite

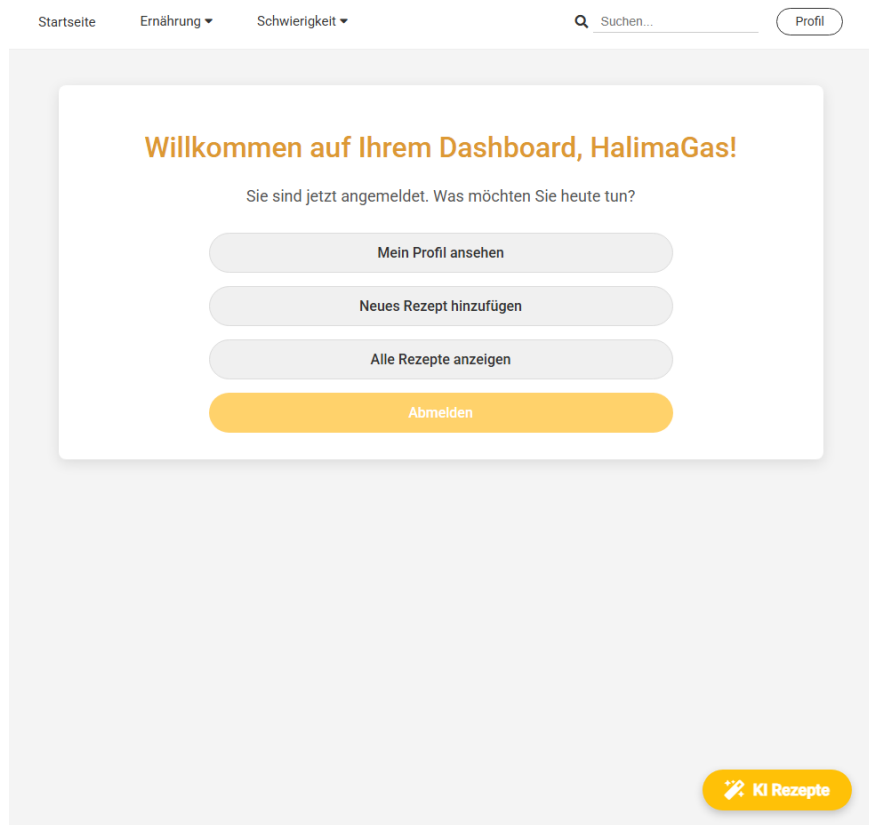


Abbildung 8:Dashboard

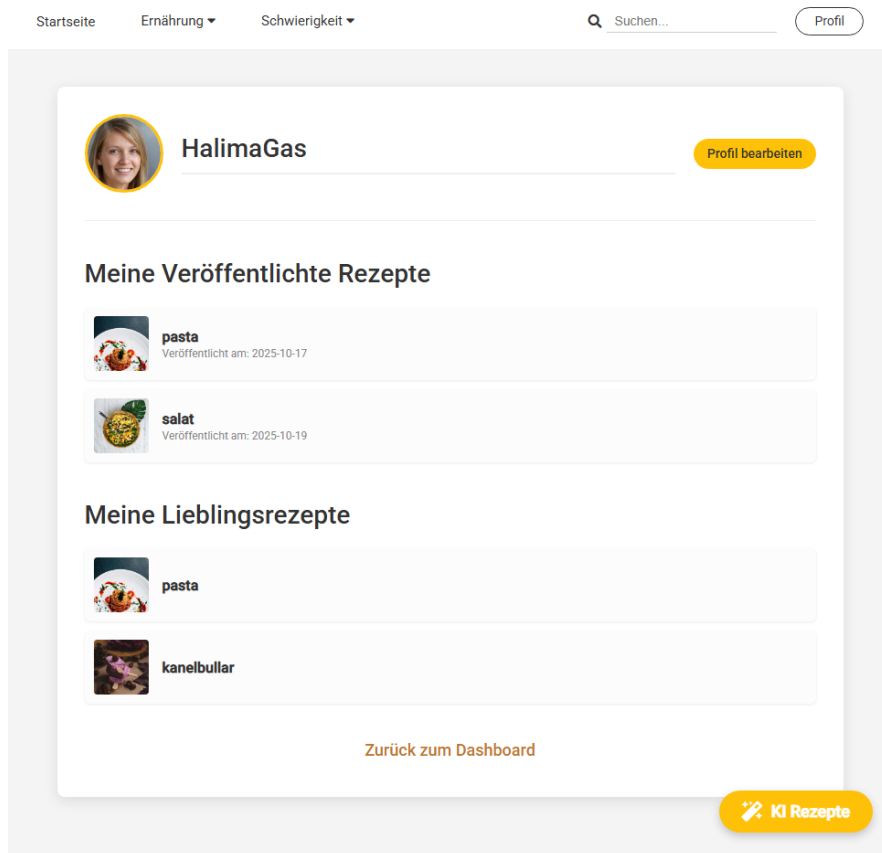


Abbildung 9:Profil

[Startseite](#) [Ernährung ▼](#) [Schwierigkeit ▼](#) [Profil](#)

Was koche ich heute?

Geben Sie die Zutaten ein, die Sie zu Hause haben (getrennt durch ein Komma). Die KI schlägt passende Rezepte vor!

Meine Zutaten hinzufügen

Oliveöl, Aubergine, Knoblauch

Rezepte finden


 KI Rezepte

Abbildung 10: KI Rezept

Fazit:

Das Projekt KochFreunde ist eine stabile und funktionsreiche Webanwendung. Es demonstriert erfolgreich die Beherrschung des gesamten Software-Engineering-Prozesses, von der Konzeption über die Implementierung bis zur Qualitätssicherung und kritischen Reflexion. Trotz des unvorhergesehenen externen Hindernisses bei der API-Integration wurden alle Kernfunktionen erfüllt und das Projekt wurde termingerecht abgeschlossen. Darüber hinaus hat die Entwicklung von KochFreunde meine Fähigkeiten in der Planung und Umsetzung komplexer Webanwendungen gestärkt. Besonders die Arbeit mit APIs und Datenbankbeziehungen hat mir gezeigt, wie wichtig saubere Architektur und Flexibilität im Entwicklungsprozess sind. Zukünftig möchte ich die Anwendung weiter optimieren und um KI-basierte Empfehlungen erweitern.