Ein Bild, das Text, Schrift, Logo, Grafiken enthält.

Automatisch generierte Beschreibung

Projektdokumentation

Internationale Hochschule Duales Studium

Studiengang: Informatik

**Fallstudie Software-Engineering**

Finnley Hintze (10234085)

Leon Kinski (10236063)

Oliver Mayer (10238283)

Maximilian Römhild (10234473)

Betreuende Person: Prof. Dr. Ulrich John

Abgabedatum: 27.05.2025

Inhaltsverzeichnis

[**1 Projektidee 3**](#_Toc198897493)

[**2 Projektorganisation 4**](#_Toc198897494)

[**3 Lessons learned 4**](#_Toc198897495)

[**4 Design 5**](#_Toc198897496)

[**5 Code (dokumentiert) 5**](#_Toc198897497)

# Projektidee

Bei der Konzeption unserer Kochbuch-Webanwendung „Kochbuch-IU“ haben wir uns bewusst mit etablierten Plattformen wie Lecker.de und Chefkoch.de auseinandergesetzt, um deren Funktionsumfang sowie Aufbau zu analysieren und daraus eigene Schwerpunkte abzuleiten.

Beide Systeme bieten ein breites Repertoire an Rezepten sowie zahlreiche Zusatzinhalte. Lecker.de legt dabei einen starken Fokus auf redaktionelle Inhalte, Rezeptvorschläge nach Saison, Werbung für Produkte und Newsletter. Die Plattform ist optisch ansprechend, jedoch inhaltlich schnell überladen – durch die Vielzahl an Filtern, Kategorien und zusätzlichen Rubriken wie „Küchenhelfer“ oder „Magazin“.

Chefkoch.de geht noch weiter und verbindet Rezepte mit einem Community-Ansatz. User können Rezepte bewerten, kommentieren, sich in Gruppen organisieren oder eigene Blogs führen. Außerdem gibt es Features wie ein „Rezept-Roulette“, Kochvideos, Foren sowie einen kostenpflichtigen Wochenplaner mit Einkaufslistenfunktion. Das System ist mächtig, aber gerade für neue Nutzer\*innen auch sehr komplex und mit Werbung durchzogen.

Unser Ansatz war bewusst ein anderer. Wir wollten kein Portal mit redaktionellen Inhalten, Community-Funktionen oder Produktwerbung bauen, sondern ein reduziertes, funktionales digitales Kochbuch. Der Fokus liegt auf den Kernfunktionen:

* einfache Verwaltung (Anlegen, Bearbeiten, Löschen) von Rezepten,
* gezielte Filterung nach Zutaten, Zeitaufwand, Schwierigkeitsgrad und Bewertung,
* eine minimalistische Benutzeroberfläche, die intuitiv bedienbar ist.

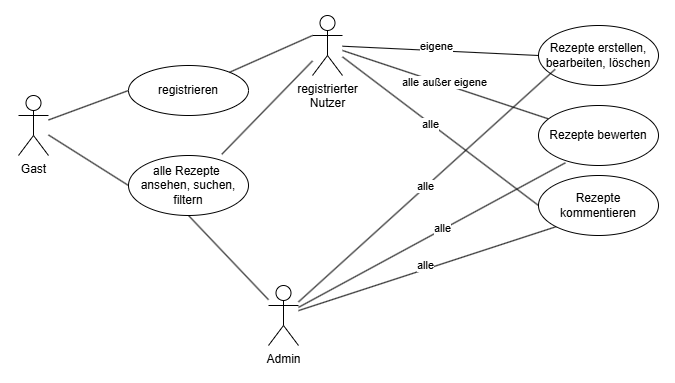
„Kochbuch-IU“ richtet sich an Nutzer\*innen, die ein schnelles, klares Werkzeug zur Rezeptverwaltung suchen – ohne Ablenkung durch Zusatzinhalte. Im Vergleich zu den großen Plattformen ist unsere Lösung daher übersichtlich, fokussiert und bewusst schlank gehalten – ideal für den Einsatz als persönliche Rezeptdatenbank oder als Lernprojekt für moderne Webentwicklung.

(Technikkram und DB Erklärung) Bspw: Technisch haben wir dafür… aufgrund [Eigenschaft und Einfachheit] benutzt um [Ziel XY] zu erreichen

Das Use-Case-Diagramm veranschaulicht die verschiedenen Nutzerrollen sowie deren jeweilige Interaktionen mit der Kochbuch-Webanwendung.

Es gibt drei Hauptakteure:

* Gast: Nicht angemeldete Benutzer können Rezepte ansehen, durchsuchen und filtern. Zudem haben sie die Möglichkeit, sich zu registrieren.
* Registrierter Nutzer: Nach der Registrierung können Nutzer eigene Rezepte erstellen, bearbeiten und löschen. Zusätzlich können sie alle anderen (nicht eigenen) Rezepte bewerten sowie Kommentare zu allen Rezepten verfassen.
* Admin: Der Administrator besitzt dieselben Zugriffsrechte wie registrierte Nutzer, kann jedoch zusätzlich auf alle Funktionen für alle Rezepte zugreifen – unabhängig vom Ersteller.

Diese Rollen und Rechte sorgen für ein klares und benutzerfreundliches System mit abgestuften Interaktionsmöglichkeiten, wobei der Fokus auf einfacher Rezeptverwaltung und Nutzerbeteiligung liegt.

# Projektorganisation

Die initiale Organisation erfolgte über Microsoft Teams, welches auch für Besprechungen genutzt wurde. Auf GitHub wurde die technische Umsetzung des Projekts aufgesetzt und die Versionen verwaltet. Zu Beginn der Projektarbeit wurden grobe Aufgabenbereiche und Zuständigkeiten aufgeteilt und im weiteren Verlauf weiter spezifiziert. In wöchentlichen Meetings wurden der aktuelle Stand sowie die nächsten Schritte und Aufgaben besprochen und verteilt.

# Lessons learned

Für Teile der Gruppe war das Arbeiten mit der Versionenverwaltung über GitHub neu, wodurch in dem Bereich neue Erfahrungen und Erkenntnisse gewonnen werden konnten.   
Bei der Ausformulierung der Spezifikation ist spürbar geworden, wie detailreich diese sein muss und welche Aspekte dabei zu berücksichtigen sind. Das klassische Problem im Projektmanagement der Baumschaukel-Analogie wurde auch für uns spürbar. Jeder hatte eine andere Vorstellung bspw. des Designs und Ausbaus im Kopf und diese mussten kommuniziert und zusammengebracht werden.

Auch die Programmierung bat Diskussionsmaterial, angefangen bei der Wahl der Programmiersprache. Wir haben und für Java entschieden und festgestellt, dass die Sprache zwar sehr mächtig ist, jedoch für ein eher kleines Projekt wie unseres schon bei kleinen Änderungen, durch Mangel an Erfahrung, viel Arbeit erfordert. Das Einarbeiten war ein langer Prozess, den man mit breiterem Wissen, was die verschiedenen Programmiersprachen angeht, sicher hätte verkürzen können.

Ein weiterer wichtiger Aspekt, den wir im Projektverlauf gelernt haben, betrifft die Organisation innerhalb des Teams. Da alle Gruppenmitglieder unterschiedliche zeitliche Verfügbarkeiten hatten – bedingt durch Arbeit, Studium und private Verpflichtungen – war es oft herausfordernd, gemeinsame Zeitfenster für Absprachen oder gemeinsames Arbeiten zu finden. Besonders in intensiveren Projektphasen wurde deutlich, wie wichtig es ist, sich bewusst Zeit freizuhalten, Prioritäten zu setzen und den Fokus auf die wichtigsten Aufgaben zu legen. Diese Erfahrung hat uns gezeigt, dass klare Absprachen, gute Selbstorganisation und realistische Zeitplanung zentrale Erfolgsfaktoren für eine effektive Zusammenarbeit sind.

Während der Zusammenarbeit im Team traten große Probleme auf mit der Verbindung zu einer lokal gehosteten Datenbank. Wir haben gelernt, dass lokale Datenbanken durch Firewalls und fehlende Portfreigaben nicht von außen erreichbar sind. Zur Vermeidung dieses Problems haben wir uns für eine MySQL-Datenbank von freesqldatabase.com entschieden. Die Einrichtung in Java erfolgte durch die Konfiguration des application.properties-Files, in dem wir die URL, den Benutzer und das Passwort der entfernten Datenbank hinterlegt haben. Durch die teamübergreifende Anbindung wurde die Entwicklung deutlich erleichtert und das Projekt startbereit.

# Design

Im Laufe unseres Projekts haben wir viele wertvolle Erkenntnisse im Bereich Design gesammelt. Besonders die gezielte Auswahl und der gezielte Einsatz von Farben haben eine zentrale Rolle gespielt.

Als Primärfarbe haben wir #003959 verwendet, ein tiefes Blau, das Seriosität, Stabilität und technische Präzision ausstrahlt – passend für den Charakter unserer Anwendung. Für Hervorhebungen und interaktive Elemente wie Hover-Effekte auf Links oder kleinere Beschreibungstexte kam die deutlich hellere, kontraststarke Farbe #55AFC2 zum Einsatz. Sie sorgt für visuelle Rückmeldung und unterstützt die Benutzerfreundlichkeit, ohne den Gesamteindruck zu stören.

Für Aktions-Icons wie Löschen oder Bearbeiten wurde bewusst die Signalfarbe #810101 gewählt. Sie erzeugt sofort Aufmerksamkeit und vermittelt klare Bedeutungen – ein roter Farbton für kritische Aktionen (z. B. „Löschen“) schafft dabei intuitive Verständlichkeit.

Ein zentrales gestalterisches Element ist der Hintergrund-Gradient, der aus den Farben Rot, Orange und Gelb besteht. Durch diesen Farbverlauf konnten wir dem Design mehr Dynamik und Wärme verleihen, ohne die Hauptfarben zu verdrängen. Der Gradient bringt visuelles Leben in die Seite und erzeugt eine angenehme Grundstimmung – vor allem in Kombination mit den zurückhaltenderen Hauptfarben.

Insgesamt haben wir gelernt, wie wichtig ein konsistentes Farbkonzept ist – sowohl funktional als auch ästhetisch. Die gezielte Farbwahl unterstützte nicht nur die Benutzerführung, sondern auch die Wiedererkennbarkeit und emotionale Wirkung unserer Anwendung.

# Code (dokumentiert)

Klassen:

Ein Bild, das Text, Screenshot, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Controller:

Ein Bild, das Text, Screenshot, Software, Display enthält.

KI-generierte Inhalte können fehlerhaft sein.