



Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

My cookbook

Entwicklung einer funktionalen Kochrezepte Anwendung

Abgabedatum: 19.05.2025

Umschüler:

Fatme Salim

Untere Kanalstraße 11

90429 Nürnberg

Ausbildungsunternehmen:

Lutz & Grub Academy GmbH

Südliche Fürther Str. 18-20

90429 Nürnberg

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
1. Einleitung.....	3
1.1 Projektziel.....	3
1.2 Projektbegründung.....	3
2. Projektplanung.....	3
2.1 Projektphasen.....	3
2.2 Abweichungen vom Projektantrag.....	5
3. Entwurfphase.....	6
3.1 Zielplattform.....	6
3.2 Entwurf der Benutzeroberfläche.....	6
3.3 Datenmodell.....	7
4. Implementierungsphase.....	8
4.1 Implementierung.....	8
Literaturverzeichnis.....	10

1. Einleitung

1.1 Projektziel

- “My cookbook” ist eine Kochbuchanwendung. Sie dient zur Erstellung, Bearbeitung und Speicherung von Rezepten.
- Ziel ist es, Rezepte anlegen zu können und diese auch Speichern zu können, sodass man jederzeit Rezepte aus der Sammlung aufrufen und bearbeiten und löschen kann.

1.2 Projektbegründung

- Das Projekt erleichtert die Ansammlung von Rezepten, da so nicht mehrere Rezepte auf verschiedenen Seiten gesammelt werden müssen, sondern alle unter einer Datei abgelegt werden können.
- Die Motivation hinter diesem Projekt ist eindeutig: Ich koche gerne und habe zu viele verschiedene Apps für meine Rezepte. Mit “My cookbook” kann ich diese alle an einem Ort sammeln.

2. Projektplanung

2.1 Projektphasen

- Projektantrag

Projektantrag FI U 24-4 Fatme Salim

Projektaufgabe

Funktionale Softwareanwendung für die Verwaltung von Kochrezepten.

Projektbeschreibung

Eine Kochbuchanwendung für die Erstellung, Bearbeitung und Erweiterung von Rezepten.
Das Projekt wird als Konsolenanwendung in C# mit dem .NET Core Framework entwickelt.
Die Rezeptesammlung wird im JSON-Format gespeichert und aufgerufen.

Folgende Funktionen wird das Projekt beinhalten:

- Rezepte anlegen in der Rezeptesammlung mit Zutatenliste, Zubereitungsanweisungen und Kalorienanzahl
- Rezepte aufrufen über die Rezeptesammlung zur Ausgabe der Kochanleitung
- Rezepte aus der Rezeptesammlung bearbeiten
- Rezepte löschen aus der Rezeptesammlung nach erneuter Bestätigung
- OPTIONAL Einkaufsliste durch Zutatenliste erstellen (falls genug Zeit vorhanden ist)

Geplante Bearbeitungszeit

Beginn: 09.05.2025 8:30 Uhr
Ende: 19.05.2025 8:15 Uhr

- Grobe Zeitplanung

Projektphase	Geplante Zeit
Entwurf	10 h
Implementierung	30 h
Abnahme	2 h
Dokumentation	14 h
Gesamt	56 h

- Genaue Zeitplanung

09.05.2025

- Projektplanung
- Projektantragsstellung
- Erstellung eines Github repository
- initial commit

12.05.2025

- Grundgerüst via Excalidraw erstellt
- Flowchart erstellt
- Groben Zeitplan erstellt
- Grundgerüst Dokumentation erstellt (Überschriften + Teilüberschriften)
- Quellen ausgesucht
- Doku Textbausteine eingefügt
- Implementierung “Main Menu” in Visual Studio

13.05.2025

- Udemy Kurs zum auffrischen
- Aufbau “Recipe Model” in Visual Studio
- Implementierung der Serialisierung über JSON
- Doku weiter bearbeitet

14.05.2025

- Implementierung “Add Recipe” in Visual Studio
- Doku weiter bearbeitet

15.05.2025

- Implementierung des Nullable handlings in Visual Studio
- Implementierung “Show Recipe”in Visual Studio
- Doku weiter bearbeitet

16.05.2025

- Umstrukturierung meiner Projektordner für mehr Ordnung
- Implementierung "Update Recipe" in Visual Studio
- Implementierung "Delete Recipe" in Visual Studio
- Visualisierung der Ausgabe im Terminal überarbeitet
- Fertigstellung der Dokumentation

- Letztendliche Zeiteinteilung

Projektphase	Geplante Zeit
Entwurf	5 h
Implementierung	36 h
Abnahme	3 h
Dokumentation	12 h
Gesamt	56 h

2.2 Abweichungen vom Projektantrag

- Optionale Funktion "Einkaufsliste erstellen" wurde nicht implementiert, da der Zeitaufwand zu hoch wäre. Aufgrund dessen wurde die Option nicht im Projekt realisiert.
- Letztendliche Zeiteinteilung verlief etwas anders als in der groben Zeitplanung, da die Implementierung mehr Zeit in Anspruch nahm und die Entwurf Phase etwas zügiger erfolgte.

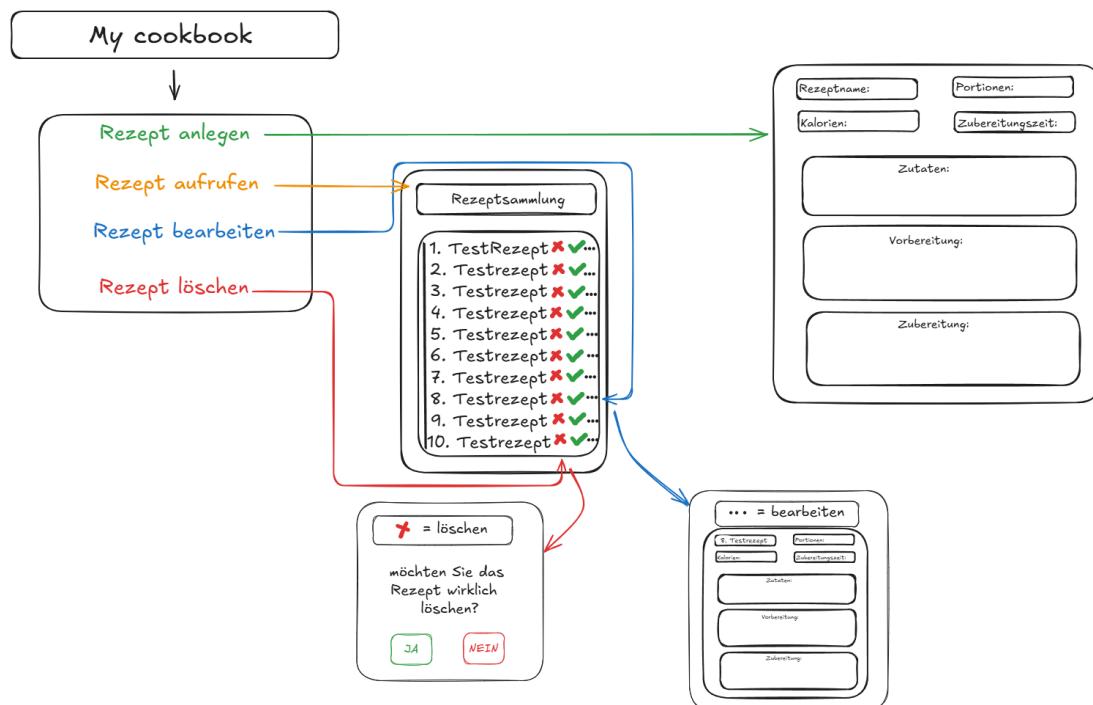
3. Entwurfphase

3.1 Zielplattform

- Das Projekt wird als Konsolenanwendung in C# mit dem .Net Core Framework entwickelt.
- Die Rezepte Sammlung wird im JSON-Format gespeichert und aufgerufen (Serialisierung)

3.2 Entwurf der Benutzeroberfläche

- Visuelle Veranschaulichung über die Idee der Benutzeroberfläche via Excalidraw

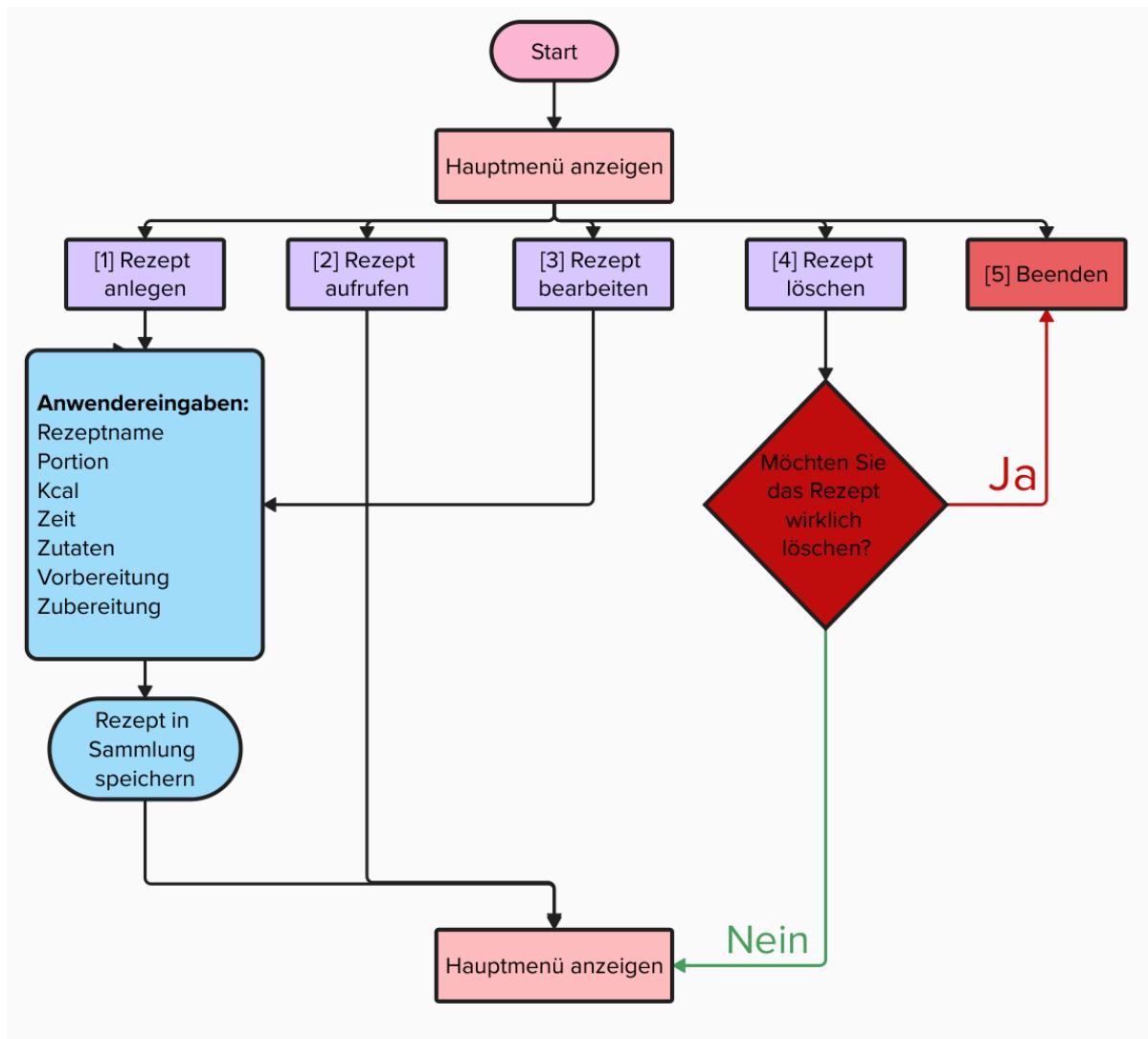


Das “Main Menu” verfügt über 4 Auswahlmöglichkeiten (+ Beenden des Programms). Über diese Auswahlmöglichkeiten kommt man zu den verschiedenen Funktionen der Anwendung.

Die Rezepte werden in einer Rezepte Sammlung gespeichert. Von dort aus kann man diese aufrufen, bearbeiten und löschen.

3.3 Datenmodell

- Visuelle Veranschaulichung des Programmablaufes via Flowchart



4. Implementierungsphase

Die Anwendung wurde objektorientiert entwickelt und folgt einem modularen Aufbau, der die Kernkomponenten des Projekts sauber voneinander trennt:

- **Models**
Enthält das Datenmodell “RecipeModel”, dass die Eigenschaften eines Rezepts beschreibt (Titel, Zutaten, Anleitungen etc.)
- **RecipeService**
Beinhaltet die Geschäftslogik in Form von Klassen, die das Interface “IRecipeService” implementieren. Dazu zählen:
 - RecipeAdd – neues Rezept hinzufügen
 - RecipeShow – Rezepte anzeigen
 - RecipeUpdate – bestehende Rezepte bearbeiten
 - RecipeDelete – Rezepte entfernen
- **Storage**
Kümmert sich um das Laden und Speichern der Rezepte in einer JSON-Datei mittels “StorageService”
- **Utilities**
Beinhaltet Hilfsklassen wie “Headline” (für Überschriften im UI) und “MainMenu” (zur Menüführung im Terminal)
- **Program.cs**
Einstiegspunkt der Anwendung- startet das Hauptmenü.

Besonderheiten:

-  **Datenkonsistenz**
Beim Speichern/Laden wurde sichergestellt, dass die JSON-Datei bei Schreib- und Lesevorgängen korrekt verarbeitet wird – insbesondere bei fehlerhaften oder leeren Dateien.
-  **Benutzerführung**
Das “MainMenu” bietet eine strukturierte Navigation durch die Funktionen. Überschriften und Pausen erhöhen die Lesbarkeit

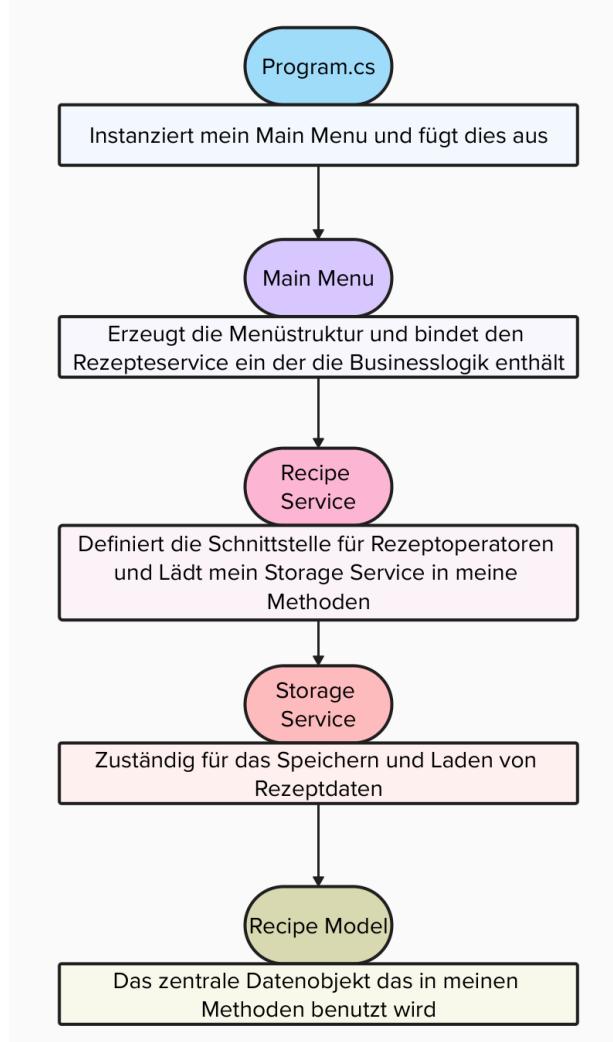
- Beispiel für eine Interaktion

```
[1] Add recipe
[2] Show recipe
[3] Edit recipe
[4] Delete recipe
[0] Quit

Your selection: |
```

Je nach Auswahl wird die entsprechende Serviceklasse aufgerufen und führt die gewünschte Aktion durch.

- Visualisierte Abhängigkeit der Komponenten



Literaturverzeichnis

Aufbau Doku:

<https://www.youtube.com/watch?v=ybcbS15xVlw>

https://dieperfekteprojektdokumentation.de/MicrosoftWord_Vorlage_Projektdokumentation_ITBerufe.pdf

Visualisierung:

Headlines:

<https://patorjk.com/software/taag/#p=testall&f=Babyface%20Lame&t=My%20cookbook>

Tipps und Hilfe:

Commit Aufbau:

<https://gist.github.com/qoomon/5dfcdf8eec66a051ecd85625518cf13>

Vertiefung C#:

<https://www.udemy.com/course/ultimate-csharp-masterclass/>

Versuch ChatGPT nach Hilfe zur Abhängigkeit zu fragen... nicht hilfreich, wurde dann selbst besser ausgearbeitet:

