**Pflichtenheft – Rezeptverwaltung**

**1. Ziel des Projekts**

Die Anwendung dient der Verwaltung von Kochrezepten in einer Desktop-Umgebung. Sie ermöglicht es dem Benutzer, Rezepte zu erfassen, zu bearbeiten, zu speichern, zu löschen, anzuzeigen sowie nach ihnen zu suchen und sie zu filtern. Zudem können Bilder zu Rezepten hinzugefügt werden. Die Daten werden lokal mittels SQLite gespeichert.

**2. Funktionale Anforderungen**

**2.1 Rezeptverwaltung**

Ein Rezept besteht aus folgenden Attributen:

* Name (String)
* Zutaten (String)
* Anleitung (String)
* Zubereitungszeit (Integer, in Minuten)
* Schwierigkeitsgrad (Integer, 1–5)
* Kategorie (String, z. B. „Dessert“, „Hauptgericht“)
* Bildpfad (optional, String)

Der Benutzer kann:

* Neue Rezepte erstellen
* Bestehende Rezepte bearbeiten
* Rezepte löschen
* Alle Rezepte anzeigen
* Nach Rezeptnamen oder Kategorien suchen
* Rezepte filtern
* Ein Bild zum Rezept auswählen und anzeigen

**2.2 Datenbankintegration**

* Lokale SQLite-Datenbank rezept.db
* Tabelle recipes mit folgenden Spalten:
  + id (Primary Key)
  + name, ingredients, instructions, time, difficulty, category, imagePath

**Operationen:**

* INSERT: Neues Rezept speichern
* SELECT: Rezepte anzeigen
* UPDATE: Rezept bearbeiten
* DELETE: Rezept löschen

Die Tabelle wird beim Start automatisch erstellt, falls sie nicht existiert.

**2.3 Benutzeroberfläche (JavaFX)**

* Tab-basiertes Layout (Tab „Rezepte“)
* Eingabefelder für:
  + Name
  + Zutaten
  + Anleitung
  + Zubereitungszeit
  + Schwierigkeitsgrad
  + Kategorie (Dropdown)
  + Bildauswahl (FileChooser)
* Buttons:
  + **Speichern** – Rezept speichern
  + **Bearbeiten** – Ausgewähltes Rezept aktualisieren
  + **Löschen** – Ausgewähltes Rezept entfernen
  + **Bild auswählen** – Bild über Dateiauswahl hinzufügen
  + **Liste aktualisieren** – Tabelle aktualisieren
  + **Suche** – Textfeld zur Filterung nach Name oder Kategorie

**Tabelle zeigt:**

* Name
* Zutaten
* Zubereitungszeit
* Schwierigkeitsgrad
* Kategorie
* (Bild wird separat angezeigt)

**3. Nicht-funktionale Anforderungen**

* **Plattform:** Desktop-Anwendung
* **Technologie:** JavaFX GUI
* **Datenspeicherung:** Lokal, keine Internetverbindung notwendig
* **Benutzerfreundlichkeit:** Intuitive Oberfläche, einfache Bedienung
* **Fehlerbehandlung:**
  + Validierung der Eingaben (z. B. numerische Felder)
  + Rückmeldungen bei Fehlern über Alerts oder Konsole

**4. Technische Anforderungen**

* **Programmiersprache:** Java 17 oder höher
* **GUI-Framework:** JavaFX
* **Datenbank:** SQLite (über JDBC)
* **IDE:** IntelliJ IDEA
* **Build-Tool:** Maven (inkl. pom.xml)
* **Dateispeicher:** Bilder werden lokal referenziert über Dateipfad

**5. Fehlerbehandlung**

* Fehler beim Parsen (z. B. Zahl erwartet, aber Text eingegeben) werden abgefangen
* Fehler beim Datenbankzugriff (INSERT/SELECT/UPDATE/DELETE) werden in der Konsole ausgegeben
* Fehlende Pflichtfelder werden durch Warnungen abgefangen (z. B. AlertBox)
* Leere oder ungültige Felder verhindern die Verarbeitung