|  |  |
| --- | --- |
|  |  |

**Dokumentation zur schulischen Projektarbeit  
Thema:**

Florian Diehle, Jonas Wolf, Cornelius Müller, Niklas Soika

FIA22B

Lernfeld/Fach

Andreas-Gordon-Schule Erfurt

Erfurt, 20.02.2025

**Inhaltsverzeichnis**

[1 Einleitung 3](#_Toc191378733)

[1.1 Projektbeschreibung 3](#_Toc191378734)

[1.2 Projektziel 3](#_Toc191378735)

[1.3 Projektumfeld 3](#_Toc191378736)

[1.4 Projektbegründung 4](#_Toc191378737)

[1.5 Projektschnittstellen 4](#_Toc191378738)

[1.6 Projektabgrenzung 4](#_Toc191378739)

[2 Projektplanung 4](#_Toc191378740)

[2.1 Projektphasen 4](#_Toc191378741)

[2.2 Abweichungen vom Antrag 5](#_Toc191378742)

[2.3 Ressourcenplanung 5](#_Toc191378743)

[3 Analysephase 5](#_Toc191378744)

[3.1 Ist-Analyse 5](#_Toc191378745)

[3.2 Wirtschaftlichkeitsanalyse 6](#_Toc191378746)

[3.2.1 „Make or Buy“ -Entscheidung 6](#_Toc191378747)

[3.2.2 Projektkosten 6](#_Toc191378748)

[3.2.3 Amortisationsdauer 6](#_Toc191378749)

[3.2.4 Nutzwertanalyse 6](#_Toc191378750)

[4 Entwurfsphase 7](#_Toc191378751)

[4.1 Zielplattform 7](#_Toc191378752)

[4.2 Entwurf der Benutzeroberfläche 7](#_Toc191378753)

[5 Projektdurchführung 9](#_Toc191378754)

[5.1 Erstellen der Mockups 9](#_Toc191378755)

[5.2 Erstellen der Datenbank 9](#_Toc191378756)

[5.3 Erstellen der API 10](#_Toc191378757)

[5.4 Testen der API 11](#_Toc191378758)

[5.5 Fehlschläge 11](#_Toc191378759)

[5.6 Frontend Programmieren 11](#_Toc191378760)

[6 Testen der App 11](#_Toc191378761)

[6.1 Test der Anmeldefunktion 11](#_Toc191378762)

[6.2 Test der Registrierungsfunktion 12](#_Toc191378763)

[7 Projektreflexion 12](#_Toc191378764)

# 1 Einleitung

In der vorliegenden Projektdokumentation wird der Ablauf des Gruppenprojekts beschrieben, das im Rahmen des Berufsschulunterrichts an der Andreas-Gordon-Schule durchgeführt wurde. Das Projekt dient als Vorbereitung auf das Abschlussprojekt in der Ausbildung zum Fachinformatiker mit der Fachrichtung Anwendungsentwicklung. Ziel dieses Gruppenprojekts war es, erworbene Kenntnisse umzusetzen und zu vertiefen und gegebenenfalls neue zu erwerben.

### 1.1 Projektbeschreibung

Das Projekt besteht aus einer Android- Koch- App mit verschiedenen zusätzlichen Funktionen und einer ausgeklügelten Gamification, um das Kocherlebnis für weniger motivierte zu erleichtern oder sogar Spaß zu bringen. Diese App setzt genau da an, indem klassische Rezeptfunktionen mit spielerischen Elementen wie dem Sammeln von Erfolgen oder dem Leveln von Zutaten kombiniert werden. Für Personen die zum Beispiel gar nicht wissen was gekocht wird können Rezeptvorschläge nach einem neuen Level genau das richtige sein.

### 1.2 Projektziel

Das Hauptziel der App ist es, den Kochprozess attraktiver zu gestalten, unteranderem durch die Gamification mit Erfolgen und dem Leveln von Zutaten. Um diese freizuschalten und zu leveln werden die Nutzer ermutigt neue Rezepte auszuprobieren und regelmäßig zu kochen. Außerdem wird dazu ermutigt eigene Rezepte hochzuladen, sodass andere diese ebenfalls nachkochen können. Die freigeschalteten Erfolge können die Nutzer auf ihrem eigenen Profil anschauen und analysieren welche ihnen noch fehlen. Vorbereiten für den Einkauf können sie sich ebenfalls mithilfe der eingebauten Einkaufsliste und dem Barcode-Scanner.

### 1.3 Projektumfeld

Das Projekt wird im Rahmen eines schulischen Teams mit vier Mitgliedern umgesetzt und ist betriebsunabhängig. Es gibt keinen externen Auftraggeber oder Kunden, sondern das Projekt entsteht aus Eigeninitiative. Die Zielgruppe umfasst Hobbyköche, Kochanfänger und alle, die sich für spielerisches Kochen begeistern können. Da es sich um ein Schulprojekt handelt, werden alle erforderlichen Ressourcen eigenständig bereitgestellt. Die Entwicklung erfolgt auf privaten Endgeräten mit frei zugänglichen Software-Tools und Entwicklungsumgebungen. Die Ergebnisse des Projekts werden innerhalb der Schule präsentiert.

### 1.4 Projektbegründung

Viele Menschen haben wenig Motivation oder Zeit, regelmäßig zu kochen, obwohl sie gerne neue Rezepte ausprobieren würden. Eine spielerische Herangehensweise kann helfen, den Kochprozess interessanter zu gestalten und Nutzer dazu zu ermutigen, öfter selbst zu kochen. Ebenfalls kann die App dabei helfen gesünder zu kochen mithilfe der eingebauten Filter-Funktion. Das Leveln der Zutaten und die Achievements fürs Kochen können dabei helfen die Benutzer zu ermutigen mehr und vielfältiger zu kochen.

### 1.5 Projektschnittstellen

Da das Projekt eigenständig entwickelt wird, gibt es keine externen technischen Schnittstellen oder Anbindungen an bestehende Systeme. Die Umsetzung erfolgt in Eigenverantwortung der vier Teammitglieder ohne externe Finanzierung. Die Zielgruppe sind Hobbyköche und Kochinteressierte. Die fertige App wird innerhalb der Schule vor der Klasse vorgestellt.

### 1.6 Projektabgrenzung

Das Projekt wird nicht der Öffentlichkeit zur Verfügung gestellt, sondern wird zuerst nur für den schulischen Zweck genutzt.

# 2 Projektplanung

### 2.1 Projektphasen

Für das Projekt standen insgesamt 40h zur Verfügung. Diese wurden zum Anfang/vor dem Projekt auf alle 4 Schüler gerecht aufgeteilt. Hierfür wurde das Projekt in einzelne Projektphasen aufgeteilt.

|  |  |
| --- | --- |
| Projektphase | Geplante Zeit |
| Analysephase | 4h |
| Entwurfsphase | 10h |
| Implementierungsphase | 21h |
| Erstellung der Dokumentation | 5h |

### 2.2 Abweichungen vom Antrag

Bei dem Endprodukt gibt es Abweichungen zum gewünschten Endprodukt. Aufgrund von dem gleichzeitigen Ausfall der Gruppe in 50% der Zeit fehlen schlussendlich die Funktionen

### 2.3 Ressourcenplanung

|  |  |
| --- | --- |
| **Verwendete Ressourcen** |  |
| **Hardware** | |
| **Laptop** | 4x |
| **Android Tablet** | 1x |
| **Software** | |
| **Visual Studio 2022** | 4x |
| **MariaDB Datenbank** | 1x |
| **API-Eigenentwicklung** | 1x |
| **GitHub zur Sicherung der Daten** | 1x |
| **Postman** | 1x |
| **Personal** | |
| **Azubi** | 4x |

# 3 Analysephase

### 3.1 Ist-Analyse

Die Projektgruppe plant die Erstellung einer „Kochapp“ für Androidgeräte. Im aktuellen Markt existieren viele bereits etablierte andere Kochapps mit verschiedensten Funktionen und Features. Jedoch gibt es keinerlei Apps die die Funktionen eines Barcodescanners, eines Einkaufswagens, eines Rezeptbuchs und gleichzeitig die einer spielerischen Möglichkeit gesunde Gerichte zu kochen vereinen, was die Idee der Gruppe beschreibt. All diese zuvor aufgelisteten Features existieren einzeln in eigenen Programmen, aber nicht in einer App die alle Aspekte vereint.

### 3.2 Wirtschaftlichkeitsanalyse

Dadurch das keine App auf dem Markt existiert die die Funktionen des Projektes erfüllen, ergibt sich eine wirtschaftliche Möglichkeit, die ausgenutzt werden kann. Jedoch muss gesagt werden, dass es im Bereich der Kochbuchprogramme viele, vor allem langjährig etablierte, Apps mit ähnlichen Funktionen gibt. Dies könnte bei einer Markteinführung, die nicht passieren kann, da es ein Schulprojekt ist, zu erheblichen Marketingkosten beitragen oder sogar zum Aus des Projekts führen. Ob es sinnvoll ist diese App selbst zu entwickeln oder die Idee zu verkaufen soll in den folgenden Abschnitten geklärt werden.

### 3.2.1 „Make or Buy“ -Entscheidung

Aufgrund dessen, dass keine Projektlösung mit den Anforderungen an Funktion, Design oder Innovation existiert und das Projektteam über keinerlei reelles Budget verfügt, kann kaum von einer „Buy-Entscheidung“ gesprochen werden. Deshalb wurde entschieden das Programm in Eigenentwicklung zu erstellen ohne andere Dienstleister mit einbeziehen zu müssen. Auch alle Softwarelösungen zur Erstellung des Projekts sind kostenlos und andere Hilfstechniken wie zum Beispiel die OpenFoodFacts API sind frei zugänglich und ohne Kosten.

### 3.2.2 Projektkosten

Im Laufe des Projektes sind keine Kosten angefallen da Software sowie Hardware bereits vorhanden waren.

### 3.2.3 Amortisationsdauer

Die Amortisationsdauer dieses Projektes kann mit 0 angegeben werden da keine Kosten angefallen sind.

### 3.2.4 Nutzwertanalyse

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Gewich-tung** | **React Native** |  | **Flutter** |  | **.NET MAUI Blazor** |  |
| **Erlernbarkeit** | 6 | 0,5 | **3** | 0,5 | **3** | 2 | **12** |
| **Performance** | 8 | 4 | **32** | 5 | **40** | 4 | **32** |
| **Community** | 3 | 4 | **12** | 4 | **12** | 4 | **12** |
| **Wartbarkeit** | 5 | 3 | **15** | 4 | **20** | 4 | **20** |
| **Entwicklungsge-schwindigkeit** | 8 | 4 | **32** | 3 | **24** | 4 | **32** |
|  |  |  | **94** |  | **99** |  | **108** |

Für die Entwicklung unserer App haben wir uns, auf Grundlage dieser Analyse, für .NET MAUI Blazor entschieden.

# 4 Entwurfsphase

### 4.1 Zielplattform

Das Abschlussprojekt soll als mobile Anwendung mit einer zentralen und einer lokalen Datenbank realisiert werden. Für die plattformübergreifende Entwicklung wurde .NET MAUI Blazor als Framework gewählt, da es eine moderne und performante Benutzeroberfläche bietet. Dieses Framework ermöglicht eine einheitliche Entwicklung für verschiedene Plattformen, wobei sich das Projekt zunächst auf Android-Geräte konzentriert. Als zentrale Datenbank wird MySQL eingesetzt, da sie sich durch hohe Stabilität und Skalierbarkeit auszeichnet. Zusätzlich wird eine lokale Datenbank implementiert, um Offline-Zugriffe und eine nahtlose Nutzererfahrung zu gewährleisten. Die Kommunikation zwischen der App und der zentralen Datenbank erfolgt über eine eigens entwickelte API, die eine effiziente und sichere Datenübertragung ermöglicht. Die Wahl der Technologien erfolgte auf Basis der Anforderungen an Skalierbarkeit, Performance und einfache Wartbarkeit. .NET MAUI wurde aufgrund seiner plattformübergreifenden Entwicklungsmöglichkeiten gewählt, während MySQL als bewährtes relationales Datenbanksystem dient. Die Kombination einer zentralen und einer lokalen Datenbank gewährleistet eine flexible Nutzung, auch ohne durchgehende Internetverbindung. Zukünftig kann die App auf weitere Plattformen wie iOS ausgeweitet werden.

### 4.2 Entwurf der Benutzeroberfläche

Die Benutzeroberfläche unserer Koch-App wurde mit dem Ziel entwickelt, eine intuitive und ansprechende Nutzererfahrung zu bieten. Dabei wurde auf eine klare Strukturierung der Inhalte und eine einfache Bedienbarkeit geachtet. Um die Navigation für die Nutzer möglichst komfortabel zu gestalten, wurde eine mobile Benutzeroberfläche mit einer modernen grafischen Gestaltung (GUI) gewählt. Für die visuelle Gestaltung wurden mithilfe von Draw.io Mockups erstellt, die das Layout der Anwendung veranschaulichen. Die Startseite enthält einen Anmelde- und Registrierungsbereich, über den sich Nutzer in die App einloggen können. Nach der Anmeldung gelangen sie auf die Hauptseite, die die wichtigsten Funktionen der App bereitstellt. Die Navigation erfolgt über eine seitliche Menüführung, die Zugriff auf zentrale Bereiche wie das Kochbuch, die Einkaufsliste, Erfolge und Einstellungen ermöglicht. Die Hauptansicht bietet eine Rezeptübersicht mit Suchfunktion und personalisierten Empfehlungen. Im Benutzerprofil können Nutzer ihren Fortschritt verfolgen, gesammelte Erfolge einsehen und ihre Zutaten verwalten. Das Kochbuch dient zur Speicherung von Rezepten und ermöglicht das Hinzufügen neuer Gerichte. In der Einkaufsliste lassen sich benötigte Zutaten verwalten und nach Kategorien sortieren. In den Einstellungen können Nutzer das Design der App anpassen und die Sprache ändern. Das Farbschema setzt sich aus einem dunklen Hintergrund mit grünen Akzenten zusammen, um eine moderne und ansprechende Optik zu gewährleisten. Die Benutzeroberfläche wurde so gestaltet, dass sie sowohl für Kochanfänger als auch für erfahrene Hobbyköche leicht verständlich ist. Klare visuelle Hierarchien und eine einheitliche Menüstruktur sorgen für eine intuitive Nutzung. Große Buttons und Icons erleichtern die Bedienung auf Touchscreens, während eine kontrastreiche Farbgestaltung die Lesbarkeit verbessert.

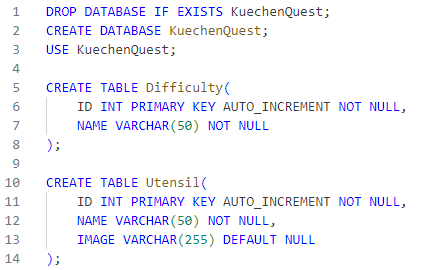
# 5 Projektdurchführung

### 5.1 Erstellen der Mockups

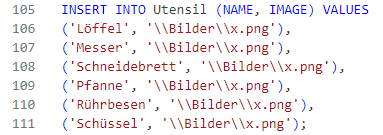
Die Mockups wurden gemeinsam in der Gruppe erstellt (siehe Anhang…………).

### 5.2 Erstellen der Datenbank

Mit dem Wissen aus vorherigen Projekten wurde ein Installations-Skript für die Datenbank erstellt. Dieses Skript beinhaltet die SQL-Anweisungen zur Erstellung der Datenbank sowie aller Tabellen, außerdem werden bereits ein paar Testdatensätze eingefügt um ein Arbeiten mit der Datenbank direkt zu ermöglichen.



In der Abbildung sieht man den Start des Skriptes. Falls eine Datenbank mit dem Namen “KuechenQuest” vorhanden sein sollte wird diese gelöscht deswegen sollte man dieses Skript nur ausführen wenn man Grundlegende Änderungen an den Tabellen vornehmen will. Nach der Erstellung der Datenbank wird in den neu erstellten Kontext gewechselt wo anschließend alle Tabellen angelegt werden. Testdaten werden am Ende des Skriptes eingefügt wie man an diesem Beispiel sehen kann.



### 5.3 Erstellen der API

Zur Programmierung der API haben wir uns für die Programmiersprache C# entschieden, da der Großteil der Gruppe bereits Erfahrung damit hat.

Die API besteht aus 4 Controllern die Anfragen bearbeiten:

* Benutzer-Controller (UserController)
* Rezept-Controller (RecipeController)
* Hilfsmittel-Controller (UtensilController)
* Zutaten-Controller (IngredientController)

Die Controller haben jeweils Endpoints wo die CRUD (Create-Read-Update-Delete) Anfragen bearbeitet werden. Der Datenbankzugriff der Controller erfolgt über eine Selbstentwickelte Database Klasse, in dieser Klasse sind alle SQL´s gespeichert und dort erfolgt auch die gesamte Fehlerbehandlung.

In der obigen Abbildung kann man exemplarisch den Login-Endoint des Benutzer-Controllers sehen. Die Klasse *MyLoginRequest* ist eine Kapselklasse, die den Benutzernamen und das Passwort enthält. Zunächst wird überprüft, ob der Request nicht leer ist. Anschließend wird die Login-Funktion der Datenbank aufgerufen. Bei erfolgreicher Anmeldung wird der Benutzer zurückgegeben, andernfalls wird *null* zurückgegeben. Auf Basis des Rückgabewerts wird dann ein entsprechender HTTP-Response-Code generiert, der an den Client gesendet wird.



### 5.4 Testen der API

Die API wurde bereits während der Entwicklungsphase mithilfe der Software Postman umfassend getestet, um ihre Funktionalität und Korrektheit sicherzustellen.

### 5.5 Fehlschläge

Die Kommunikation mit der OpenFoodFacts API konnte aufgrund von Krankheit leider nicht mehr implementiert werden.

### 5.6 Frontend Programmieren

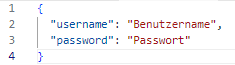
In der Planungsphase des Projektes wurde diskutiert welche Technologie für die App verwendet werden sollte, dazu haben wir auch eine Nutzwertanalyse erstellt (vgl. 3.1.4 Nutzwertanalyse). Schlussendlich haben wir uns dazu entschieden .NET MAUI Blazor zu verwenden.

Die Frontend Programmierung begann mit dem erstellen der Login Seite, da diese als Standardseite fungiert, wenn es zu Fehlern in der App kommt. Weiterführenden Seiten wie z.B. die Startseite oder die Profilseite wurden danach von einzelnen Teammitgliedern parallel erstellt/entwickelt.

# 6 Testen der App

### 6.1 Test der Anmeldefunktion

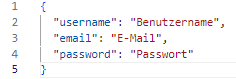
Die Login-Funktion wurde während der Entwicklung des Back- und Frontends zweimal unabhängig voneinander getestet. Im Vorfeld der Entwicklung wurde ein Kommunikationsformat vereinbart (siehe nachfolgende Abbildung).



In Zukunft ist die Übermittlung des Passwortes in gehashter Form geplant um die Sicherheit der Userdaten weiter zu erhöhen.

### 6.2 Test der Registrierungsfunktion

Die Registrierungsfunktion ähnelt der Login-Funktion in ihrer Funktionsweise. Beim Registrieren eines Benutzers muss hier zusätzlich zu Benutzername und Passwort eine einzigartige Email vergeben werden.



# 7 Projektreflexion

Die Planung des Projektes innerhalb des Projektteams verlief gut allerdings war der selbstgesteckte Zeitplan etwas zu gering bemessen. Die Zeitplanung wurde durch Abstimmung innerhalb des Teams festgelegt anschließend haben wir einen Projektstrukturplan und einen Netzplan erstellt (siehe Anhang…………). Der erste Schritt der Projektdurchführung war das Erstellen der Mockups hier haben wir festgestellt das ein Teammitglied sich bereist mit der Erstellung des Backends befassen konnte, ohne den Erstellungsprozess zu verlangsamen. Für die Durchführungsphase hat sich das Projektteam in zwei kleinere Teams unterteilt (Frontend, Backend). Die Aufteilung in zwei Teams hat sehr gut funktioniert da das Frontend Team bereits viele Funktionen des Backends benutzen konnte, obwohl die Entwicklung noch nicht abgeschlossen war. Aufgrund eines Komplettausfalls des gesamten Teams in der zweiten Hälfte des Projektes konnten wir nicht alle geplanten Funktionen im Projektzeitraum entwickeln. Durch eine außerplanmäßige Verlängerung des Projektzeitraumes konnte dennoch ein vorzeigbares Produkt entwickelt werden. Abschließend lässt sich sagen das die geplanten Funktionen unsere individuellen Fähigkeiten entsprechen und die Implementierung aller Funktionen bei ausreichender Zeit möglich gewesen wären.