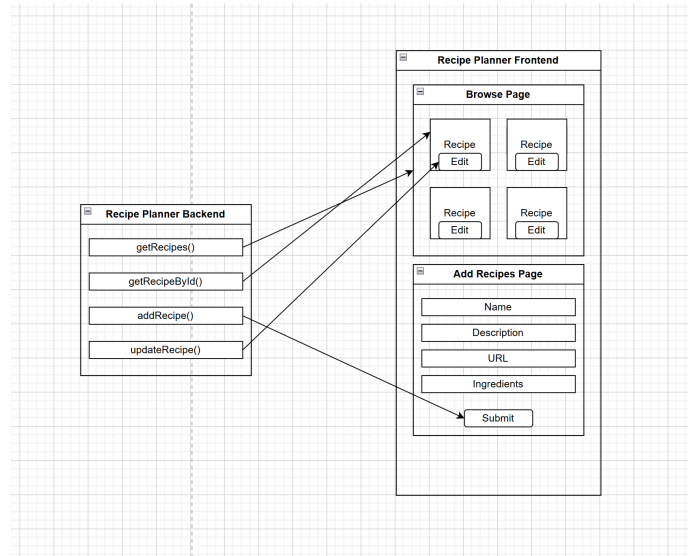


# Recipe Planner Testkonzept

## Big Picture:



Wir programmieren eine Webseite, auf der man verschiedene Rezepte anschauen und auch eigene hinzufügen kann. Wir haben im Frontend zwei Seiten. Eine Seite, auf der man alle Rezepte anschauen kann und, wenn man es will, diese ändern kann. Auf der anderen Seite kann man seine eigenen Rezepte hinzufügen. Beide Seiten benutzen verschiedene CRUD Operationen, die sie vom Backend bekommen.

## Test Features:

### Backend:

- Unit Tests werden für jede Service-Funktion geschrieben. Diese sollten positive und negative Testfälle abdecken.
- Für alle Endpoints werden automatisierte Postman-Tests geschrieben.
- Ein Load Test wird auf dem Backend ausgeführt.

### Frontend:

- Das ganze Frontend wird mit automatisierten Cypress Tests getestet. Zu diesen Tests gehören automatisierte E2E Tests wie z.B. ein Rezept editieren.
- Während der Entwicklung werden auch manuelle Tests durchgeführt.

## Diverses:

- Im Github unseres Projektes werden zusätzlich CI/CD Tests hinzugefügt, um neu hinzugefügter Code abzutesten.

## Features not to be tested:

- Sicherheitslücken im Frontend und im Backend werden nicht abgetestet.
- Ausser Load Tests im Backend werden keine Performance Aspekte des Programms abgetestet.

## Testvorgehen:

Wir werden mit Hilfe von TDD (Test Driven Development) dieses Projekt durchführen. Für bereits geschriebene Teile des Codes werden Tests nachträglich hinzugefügt. Die Addition von Edit-Page wird von Anfang an mit TDD durchgeführt.

Im GitHub werden alle Feature-Changes in einem separaten individuellen Branch gepusht und dann nach einem Code Review in den dev Branch gemerged. Nach jedem Arbeitstag werden diese changes im dev Branch in den main Branch gepusht.

## Testumgebung:

### Backend:

- Postman
- JUnit
- IntelliJ
- Mockito

### Frontend:

- Cypress
- VSCode
- Node.js
- React

### Sonstiges:

- GitHub
- Draw.io
- Google Docs

## Kurze Planung:

Wir werden uns zuerst mit dem Backend beschäftigen. Jemand wird die bereits geschriebenen Codezeilen mit Tests abdecken und der andere wird mit dem neuen Endpoint beginnen. Wenn

jemand schon früher fertig ist, wird diese Person schon mit den CI/CD Tests anfangen. Sobald der letzte mit den Backend Tests fertig ist, wird diese Person die Frontend Tests schreiben. Da wir nur 2 Personen im Team sind, wird jeder den Code des anderen abchecken und Feedback dazu geben.

## **Reflexion:**

Das Projekt „Recipe Planner“ war unser erstes Projekt, bei dem wir sowohl Test-Driven Development (TDD) als auch Pull Requests auf GitHub verwendet haben– und es war definitiv eine lehrreiche Erfahrung.

Einer der grössten Stolpersteine war, uns an die Arbeitsweise von TDD zu gewöhnen. Normalerweise beginnen wir damit, den Code direkt zu schreiben, aber diesmal mussten wir zuerst die Tests erstellen. Am Anfang gab es ein paar Fragezeichen, da wir nicht wussten, wo wir beginnen sollten. Aber mit der Zeit haben wir es verstanden. Die Tests gaben uns eine klare Richtung und stellten sicher, dass jede Codeänderung tatsächlich funktionierte. Das Gefühl, einen Test „grün“ zu sehen, war jedes Mal eine kleine Belohnung.

Auch die Arbeit mit Pull Requests war neu für uns. Anfangs wussten wir nicht, wie das geht, wie man Kommentare schreiben kann, oder die Änderungen des Teamkollegen zu verstehen. Es hat etwas gedauert, bis wir einen Workflow etabliert hatten, der für uns beide funktionierte. Wir haben gelernt, wie wichtig klare Kommunikation bei Code-Reviews ist.

Ausserdem haben wir gelernt, Geduld zu haben mit dem ganzen Prozess, da es nicht von Anfang an funktionieren wird.

In der Zukunft sind viele Vorteile darin zu sehen, was wir gelernt haben. TDD gibt einem ein sicheres Gefühl, dass der Code funktioniert und robust ist. Pull Requests sorgen dafür, dass man nicht alleine arbeitet, sondern unsere Ideen von anderen geprüft werden– was die Codequalität enorm verbessert.

Zusammenfassend war das Projekt herausfordernd, aber wir beide fühlen uns jetzt viel sicherer im Umgang mit modernen Entwicklungspraktiken. Es hat uns gezeigt, wie wichtig es ist, strukturiert zu arbeiten, sauber zu kommunizieren und nicht nur auf das Ziel, sondern auch auf den Weg dorthin zu achten.

.