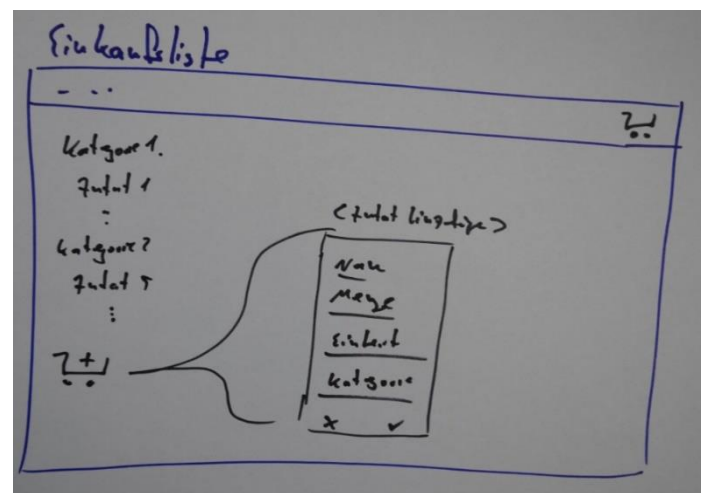
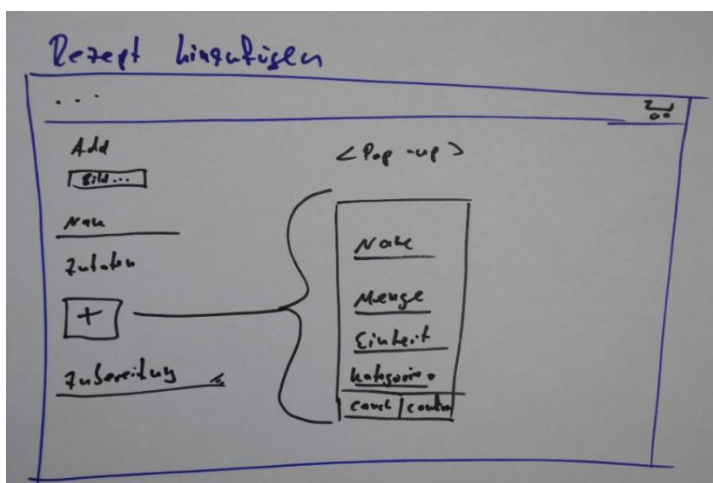
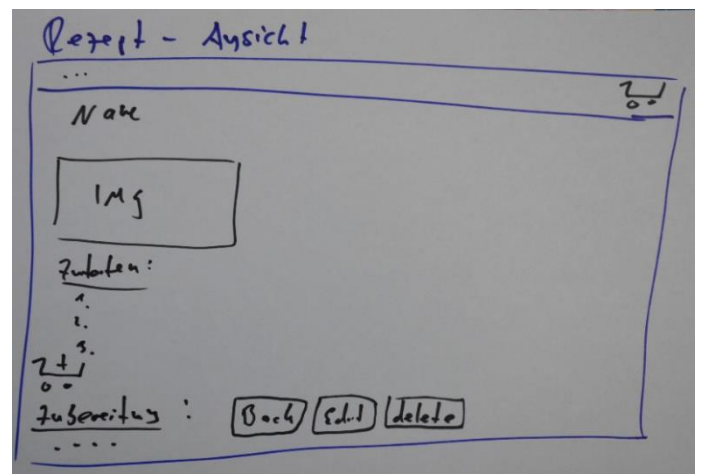
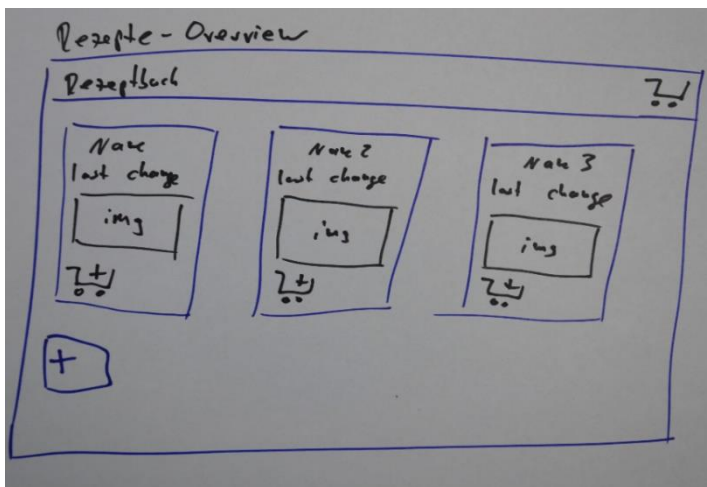


Leider war es mir nicht möglich die Dokumentation auf 2 Seiten herunterzubrechen. Ich habe versucht mich dennoch möglichst kurz zu fassen und hoffe der Umfang ist den Anforderungen entsprechend.

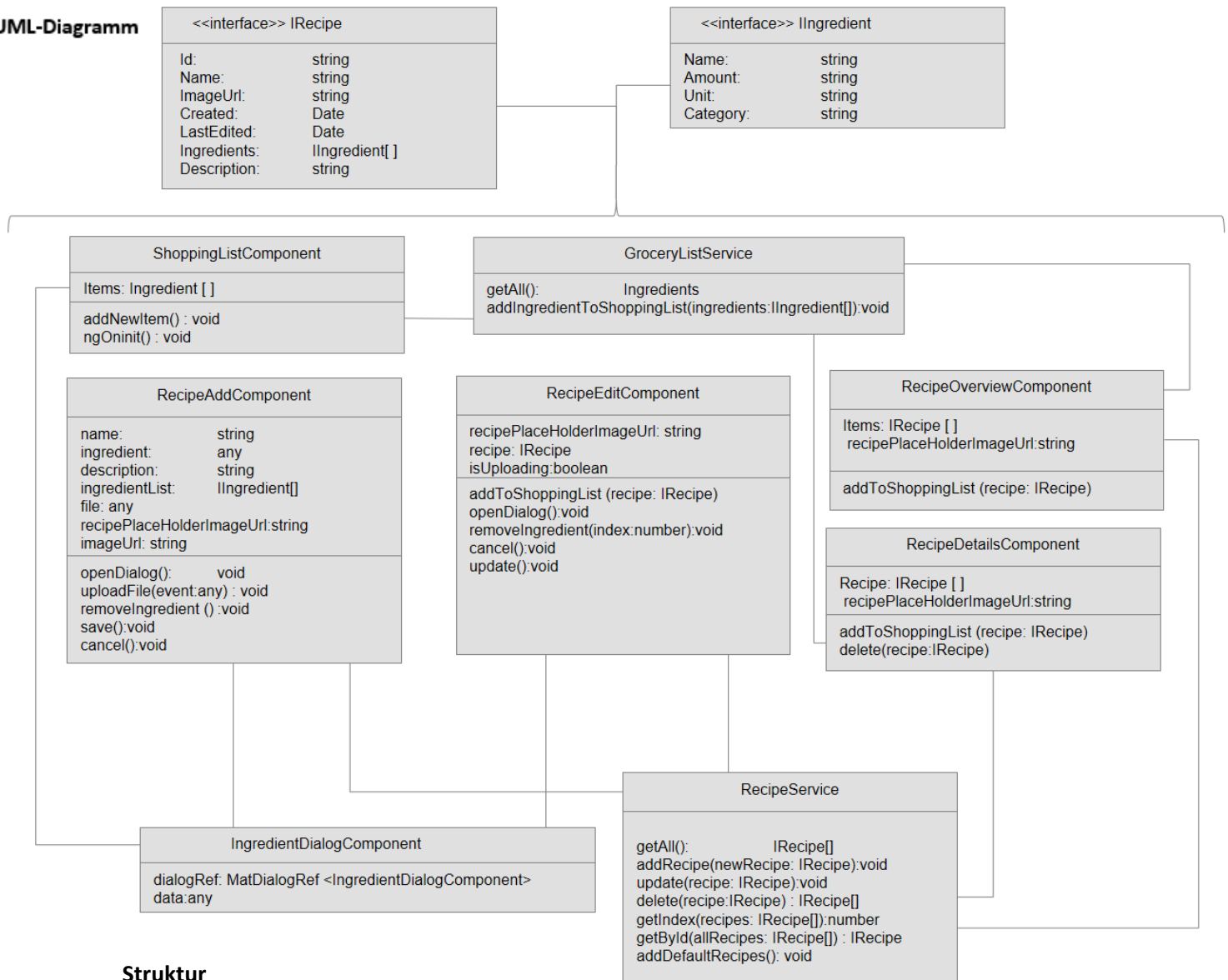
Anforderungen

Es ist nach einem Online Rezeptbuch gefragt, welches die Zubereitung diverser Rezepte beschreibt, die Zutaten anzeigt und die Möglichkeit bietet die entsprechenden Zutaten für die Rezepte in eine Einkaufsliste hinzuzufügen. Des Weiteren soll es möglich sein, eigene Rezepte mit entsprechendem Bild, Zutaten und einer Anleitung anzulegen. Es muss möglich sein, Zutaten für ein bestimmtes Rezept anzulegen, aber auch einzelne Zutaten zu der Einkaufsliste hinzuzufügen. In der Einkaufsliste werden die hinzugefügten Zutaten nach deren entsprechenden Kategorien sortiert und kumuliert angezeigt.

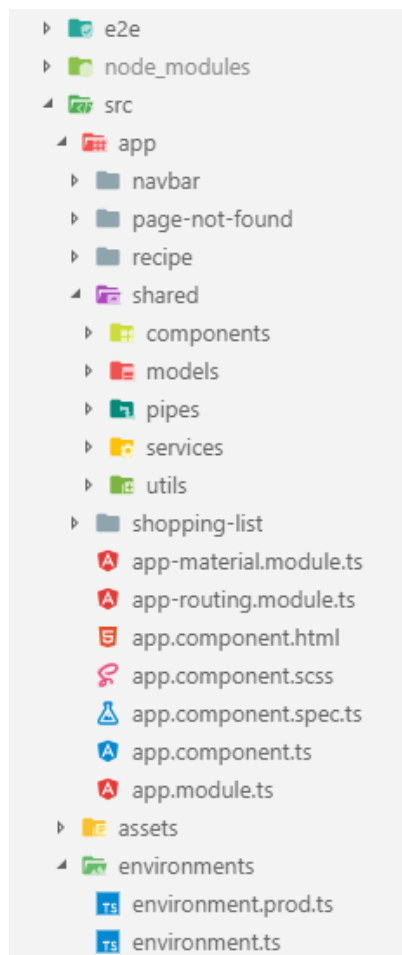
Papier-Prototyp



UML-Diagramm



Struktur



Grundlage für die Entwicklung sind die Angular Material Module.

Links ist die Ordnerstruktur zu erkennen, welche folgenden Aufbau aufzeigt:

- Recipe und shopping-list enthalten die Hauptfunktionen für die Funktionalität der Web Seite.
- Recipe enthält hierbei die Funktionalitäten: *add*, *details*, *edit* und *overview*
- Die Deklaration der Interfaces wurde unter *models* abgelegt.
- Der Ordner *Shared* enthält übergreifende Komponenten:
 - ➔ *Components*: Enthält das Pop-Up Fenster: Zutaten hinzufügen
 - ➔ *Models*: Enthält die Deklarationen der benutzen Interfaces
 - ➔ *Pipes*: Enthält eine GroupBy-Pipe, die das Sortieren nach Category möglich macht
 - ➔ *Services*: Beinhaltet das Ablegen, Ändern und Abfragen der Local Storage Daten für die Rezeptzutaten und die Einkaufsliste
 - ➔ *Utils*: Beinhaltet die Helper Methode, welche für das summierte Anzeigen der Zutaten im Einkaufswagen benötigt wird und zur Generierung einer GUID

Komponenten

Rezepte-Übersicht

Die Startseite der Webseite zeigt die Verfügbaren Rezepte an. Hierbei kann durch das Klicken auf das Rezept zu der jeweiligen Detailansicht gewechselt werden. Außerdem können durch einen Klick auf den „Zutaten Hinzufügen“ Button, die benötigten Zutaten für das Rezept direkt in die Einkaufsliste hinzugefügt werden.

Mat-card: Beinhaltet im Header den Titel und das Änderungsdatum, zudem das Bild des Rezeptes und den Button zum Hinzufügen der Zutaten.

→ Über das Klicken auf das Rezept wird zum entsprechenden Rezept gewechselt: „/recipe/view, id“

Mat-Card-Action <Button>: Über den Button können dir Zutaten aus dem Rezept direkt in die Liste hinzugefügt werden:

→ Über die Funktion `addToShoppingList(recipe)` werden die entsprechenden Zutaten zur Shopping List hinzugefügt.

New-recipe-Button: Über den Button unterhalb der Rezeptübersicht kann ein neues Rezept angelegt wird indem zu „/recipe/add“ gewechselt wird.

Rezept Details

In der Detailansicht werden die benötigten Zutaten und die Zubereitung angezeigt. Zudem kann man über den selben Button wie in der Übersicht die Zutaten in seine Einkaufsliste hinzufügen.

Mat-Card-Img: Zeigt das entsprechende Bild des Rezeptes an

Zutatenliste: Die Zutaten werden Simpel in einer Liste angezeigt mit der Reihenfolge: Name, Menge & Einheit.

Button: Über die vier vorhandenen werden folgende Funktionalitäten bereitgestellt:

1. Zutaten hinzufügen → führt „`addToShoppingList()`“ aus
2. Back → navigiert zurück zur Rezeptübersicht (/recipes)
3. Edit → wechselt in die Bearbeitungsansicht des Rezeptes (/recipe/edit) (siehe Rezept ändern)
4. Delete → löscht das Rezept

Rezept anlegen

Hier kann ein entsprechendes Bild hochgeladen, ein Name, Zutaten und eine Zubereitung hinzugefügt werden.

Mat-Form-Field: Dienen als Eingabefelder für den Namen und die Zubereitung des Rezeptes.

Button: Über die fünf verfügbaren Button werden folgende Funktionen realisiert

1. File Upload, des Bildes
2. Zutaten hinzufügen → Öffnet den entsprechenden Mat-Dialog zum Zutaten hinzufügen (siehe Zutaten hinzufügen)
3. Zutaten löschen → ermöglicht das Löschen entsprechender Zutaten
4. Cancel → bricht den Vorgang ab
5. Save → speichert das Rezept mit „save()“ und, falls der User ein Bild ausgewählt hat, so wird dieses in den Firebase Storage hochgeladen. Wenn kein Bild vorhanden ist, wird die imageUrl nicht gesetzt, sodass er bei den anderen Komponenten ein Platzhalter Bild anzeigt.

Rezept ändern/ löschen

Über die Rezept-Detailansicht ist es möglich das Rezept zu ändern oder zu löschen. Die Bearbeitungsseite ist hierbei gleich aufgebaut wie die Seite, über welche neue Rezepte hinzugefügt werden. Der einzige Unterschied besteht darin, dass der Save Button die Beschriftung „Update“ trägt

Zutaten hinzufügen

Das Zutaten hinzufügen Fenster ist ein Pop-Up, welches sich auf der jeweiligen Seite (Rezept hinzufügen oder in der Einkaufsliste) entsprechend öffnet. Dort können neue Zutaten mit den Eigenschaften: Name, Menge, die Einheit und die Kategorie hinzugefügt werden. Bei der Menge muss mindestens „1“ ausgewählt werden und die Kategorien sind vorab vorhanden und der >Benutzer muss zwischen: „Fleisch, Obst, Gemüse und Sonstiges“ auswählen.

Einkaufsliste

In der Einkaufsliste werden die hinzugefügten Zutaten aus dem jeweiligen Rezepten nach den entsprechenden Kategorien (Object.key) sortiert und den Mengen summiert aufgelistet, sodass ein Einkauf im Supermarkt durch diese Liste möglichst einfach ist.

Mat-Dialog: Ermöglicht die Anzeige der durch den GroceryListServices bereitgestellten Zutaten.

- GrocerylistService greift hierbei alle Zutaten, welche im Local Storage gespeichert sind ab.

Besonderheiten: Für die summierte Anzeige der Zutaten war eine *Helper*-Funktion nötig, welche die entsprechenden Zutaten aufaddiert, welche den selben Namen tragen.

Shared

Über die gesamte Webseite mehrmals verwendete Komponenten wurden alle gesammelt unter dem Shared Ordner abgelegt, welcher folgende Komponenten enthält:

- Das Pop-up Fenster für das Hinzufügen der Zutaten
- Deklaration der Interfaces
- Pipe-Deklaration zur Sortierung der Zutaten nach Category
- Services → Zutaten & Einkaufslisten-Operationen (CRUD)
- Helper-Class mit Helper-Methoden (GUID-Generierung und Gruppierung der Zutaten)

404-Seite

Für den Fall, dass ein User auf einen nicht hinterlegten Link zugreift erscheint die Error-Page mit entsprechender Navigation zurück zur Übersicht.

Snackbar

Die Snackbar informiert den User nach getätigten Aktion über dessen Abschluss. Beispielsweise über das erfolgreiche Hinzufügen der Zutaten in die Einkaufsliste oder das Hinzufügen eines neuen Rezeptes.

Herausforderungen

Die Herausforderungen bei diesem Projekt bestand darin, dass ich mich zunächst mit Firebase und Pipes auseinandersetzen musste. Dies nahm einige Zeit in Anspruch um die Grundlagen hierfür zu lernen und zu verstehen.

Reflektion

Bei abschließendem Begutachten des Projektes sind einige Verbesserungspotentiale aufgefallen. Welche im folgenden aufgelistet werden:

- Zutatenliste: Das Hinzufügen von einzelnen Zutaten zu Rezepten und der Einkaufsliste erfolgt stets über das Dialog Pop-Up. Somit werden Zutaten stets mit entsprechenden Eigenschaften neu angelegt und bereits hinzugefügt Zutaten können nicht einfach ausgewählt werden. Realisierbar wäre diese Optimierung, indem man auf die bereits gespeicherten Zutaten im Local Storage über ein Dialogfenster zugreifen könnte.
 - Local Storage: Da die Daten (bis auf das Bild) alle im Local Storage gespeichert werden wäre ein Button, welcher diesen säubert (`localStorage.clear()`; ausführt) unter Umständen ein „nice-to-have“ feature.
- Einkaufsliste: Wünschenswert wäre eventuell noch die Möglichkeit die entsprechend hinzugefügten Zutaten auch wieder aus der Einkaufsliste zu entfernen, bzw. die Einkaufsliste als ganzes zu löschen.
- Helper-Methode: Ermöglicht zwar das summierend der Zutaten, bietet jedoch noch Verbesserungspotential im Punkt Fehlerfreiheit. Im Allgemeinen kann gesagt werden, dass es hierfür eine elegantere Lösung gegeben hätte.

Alles in allem bietet die Website die nötigen Funktionalitäten und erfüllt die Anforderungen an ein Online-Rezeptbuch. Auch wenn es an einigen Stellen etwas an Sauberkeit und Genauigkeit mangelt, kann hier von einem erfolgreich abgeschlossenen Projekt gesprochen werden.