# Kurzfassung

Es wird eine Web-Applikation erstellt werden, welche es ermöglicht vordefinierten (und selbst hinzugefügten Rezepten) in eine Einkaufsliste hinzuzufügen um sich so seine Einkaufsliste basierend auf Rezepten aufbauen und verwenden zu können.

# Requirements

Es soll eine Applikation erstellt werden, welche die Möglichkeit bietet, eine Einkaufsliste zu erzeugen, welche durch die Auswahl verschiedener vordefinierter Rezepte angereichert werden kann. Das heißt, wenn der Nutzer ein Rezept auswählt, werden die Zutaten des Rezepts der Einkaufsliste hinzugefügt. Die Rezepte bestehen dabei aus verschiedenen Zutaten, welche der Nutzer ebenfalls zu einer Zutatenliste hinzufügen kann. Dabei soll die Applikation in mehrere Webpages aufgeteilt werden. Innerhalb jeder Website soll die Möglichkeit bestehen auf die „Homepage“ / „Zutaten“ und zur „Einkaufsliste“ zur gelangen.

Die Startpage (oder auch die Homepage) ist die Webseite, die geöffnet wird, wenn die Applikation startet. Die Startseite bietet dabei eine Ansicht der verschiedenen vordefinierten Rezepte an mit der zusätzlichen Funktion, neue Rezepte hinzuzufügen. Des Weiteren soll der Nutzer die Möglichkeit besitzen, sich ein einzelnes Rezept per Detailansicht anzeigen zu lassen.

Wenn der Nutzer ein Rezept hinzufügen möchte, soll er zu einer eigenen Webpage gelangen, auf welcher er ein neues Rezept hinzufügen kann. Dabei soll ein Rezept ausfolgenden Komponenten bestehen:

* Namen
* Eine oder mehreren Zutaten
* Ein Bild (wird per Bild URL hinterlegt)
* Zubereitungsschritte des Rezepts in Textform.

Der Nutzer kann dieses Rezept dann per Knopf hinzufügen oder das Hinzufügen abbrechen, welchen ihn zurück zu der Homepage bringen würde.

Die Website „Zutaten“ bietet dem Nutzer die Funktion eine neue Zutat dem System hinzuzufügen und eine Auflistung der Zutaten mit aufgeteilt in ihre verschiedenen Kategorien. Diese Zutat kann dann einem Rezept, auf der vorher genannten Webpage für ein Rezept ausgewählt werden. Eine Zutat hat dabei folgende Attribute, die vom Nutzer gewählt werden müssen:

* Eine Zutaten Kategorie (vordefiniert, Bsp. Molke Produkt oder Süßwaren)
* Namen
* Einheit (vordefiniert, Bsp. Gramm, KG, Stück)

Der Nutzer kann die Zutat dann per Knopf hinzufügen oder das Hinzufügen abbrechen, welchen ihn zurück zu der Homepage bringen würde.

Die „Einkaufsliste“ Webpage soll eine Auflistung der ausgewählten Zutaten mithilfe der Auswahl der Rezepte darstellen. Zudem soll es eine Möglichkeit geben, zurück auf die Homepage zu kommen, per eigenem Button.

Eine Datenbankanbindung soll nicht existieren, jegliche Änderungen wären also nur bis zu einem neu laden der Webpage vorhanden.

## Anwendungsbeispiele

### Beispiel 1: „Der normale Einkauf“:

Der Nutzer wählt fügt auf der Homepage zwei Rezept direkt der Einkaufsliste hinzu. Zusätzlich klickt er auf ein Rezept, um auf die Detailseite dieses Rezept zu gelangen und fügt dieses auch hinzu. Der Nutzer betrachtet nun die vollständige Einkaufsliste, um seinen Einkauf im Supermarkt durchzuführen.

### Beispiel 2: „Rezept hinzufügen“:

Der Nutzer hat ein neues Rezept gefunden und möchte dies nun der Applikation hinzufügen. Dabei nutzt er die Zutaten Webpage, um die neuen, noch nicht vorhandenen Zutaten dem System hinzuzufügen. Anschließend füllt er dann das Formular für das Hinzufügen eines Rezeptes aus und nutzt hierbei die neu eingefügten Zutaten in Kombination mit bereits vorhandenen.

# Konzept/Aufbau

Die Website wird in verschiedene Komponenten aufgeteilt, um so einzelne Funktionen zu modularisieren. Eine grafische Veranschaulichung der einzelnen Komponenten ist beiliegenden PowerPoint Präsentation zu finden. Als Unterstützung für die Darstellung werden hierbei einzelne Komponenten von [https://material.angular.io](https://material.angular.io/) genutzt.

## Komponenten

### AppComponent

Die AppComponent wird hierbei auf jeder Webpage dargestellt und enthält die Navbar mit den jeweiligen Verlinkungen zu den folgenden Webpages per Button:

* Homepage, als Button (RecipeList Komponente)
* Zutaten, als Button (CreateIngridient Komponente)
* Einkaufsliste, als Button (ShoppingCart Komponente)

Zudem stellt diese Komponente das Routing für den Zugriff auf die anderen Komponenten.

### RecipeList

Die RecipeList Komponenten stellt die verschiedenen Rezepte mithilfe der <mat-card> dar. Innerhalb der dieser Card, befindet sich:

* Names des Rezepts
* Nicht anklickbares Bild des Rezepts
* Button, welche alle Zutaten des Rezepts der Einkaufsliste hinzufügt
* Button,, eine Verlinkung zu der RecipeDeteail Webpage des Rezepts

Zusätzlich bietet die Webpage einen Button (Bild: „+“ Zeichen“) welcher eine Verlinkung zur CreateRecipe Webpage hat.

### RecipeDetail

Bei der RecipeDetail Webpage wird eine <mat-card> dargestellt, welche den Inhalt eines einzelnen Rezeptes auflistet. Die einzelnen Bestandteile der Card werden hierbei mit folgenden Daten ausgestattet:

### Feld welcher einen Textwert für den Namen des ausgewählten Rezepts darstellt.

### Nicht anklickbares Bild des Rezepts

* Auflistung der einzelnen Zutaten
* Button, welcher die Zutaten zu der Einkaufsliste hinzufügt
* Button, eine Verlinkung zur Homepage

### CreateRecipe

Diese Webpage hat einen Ähnlichen Aufbau wie die RecipeDetail Webpage, jedoch sind hierbei die einzelnen Bestandteile Inputfelder. Diese Eingabewerte werden nicht auf Semantische Korrekheit geprüft und es können sogar Duplikate hinzugefügt werden.

### Aufbau der Card:

### Feld welches einen Textwert für den Namen des neuen Rezepts weitergibt.

### Textfeld für die URL des neuen Bildes

* Dropdown Feld, zur Auswahl einer vordefinierten Zutatenkategorie
* Dropdown Feld, zur Auswahl einer vordefinierten Einheit der Zutat
* Liste, welche eine Auflistung der hinzugefügten Zutaten bietet
* Add Button, welcher das neue Rezept an den Datenservice weitergibt, welche die Rezeptliste aktualisiert.
* Cancel Button, eine Verlinkung zur Homepage. Bei Betätigung wird der Vorgang abgebrochen und alle Daten eingegebenen Werte gehen verloren.

### CreateIngridient

Die Webpage ist mit 2 <mat-card> aufgebaut, eine auf der linken Hälfte und eine auf der rechten Hälfte.

Linke Card:

* Enthält eine Auflistung aller vorhandenen Ingredients, sortiert nach ihrer Kategorie.

Rechte Card:

* Eingabe Feld für den Namen der neuen Zutat
* Dropdown Feld, zur Auswahl einer vordefinierten Zutatenkategorie
* Dropdown Feld, zur Auswahl einer vordefinierten Einheit der Zutat
* Add Button, welcher die neue Zutat an den Datenservice weitergibt, welche die Zutatenliste aktualisiert.

### ShoppingCart

Bietet eine Darstellung der gesamten hinzugefügten Zutaten über die Rezepte an. Hierbei gibt es wieder eine große <mat-card>, welche die Zutaten auflistet. Die Auflistung erfolgt über die <mat-list> innerhalb der Card. Die Zutatenliste wird über den Datenservice abgefragt. Bsp.:

Die mat-list enhält dabei den Namen der Zutat, die Einheit und die momentane Menge für die Einheit. Bsp.: Eier, pcs, 2x.

Wenn der Nutzer eine Zutat über dem Hinzufügen eines Rezeptes eine bereits vorhandene Zutat hinzufügt, dann wird nur die momentane Menge erhöht und kein Duplikat erzeugt.

## PageNotFound

Stellt eine Fehlermeldung bei betreten der Komponente dar.

## DataService

Der DataService stellt die folgenden Daten allen Komponenten der Webpage bereit.

* recipeList: Enthält alle momentanen Rezepte mit allen möglichen Kategorien für Zutaten und die benötigten Zutaten.
* categoryList: Enthält alle möglichen Zutatenkategorien.
* ingridientList: Enhtält alle Zutaten.
* recipeItem: enthält das momentan betrachtete Rezept, zur weiteren Nutzungsmöglichkeit für andere Komponenten.
* Vordefinierten Testdaten die für Beispielrezepte benötigt werden.

## Routing

Die Routing Komponenten übernimmt die Verlinkung der einzelnen Komponenten. Dabei wird wie vorher beschrieben der Header der app.component immer auf jeder Webpage angezeigt. Somit tauscht der Router nur den zentralen Teil der Website bei Nutzung einer Verlinkung aus.

Dabei gibt es folgende Routen:

* Pfad: ““ (Homepage)'', Komponente: RecipeList}
* Pfad: 'RecipeItem', Komponente: RecipeItem
* Pfad: 'AddRecipe', Komponente: RecipeCreate
* Pfad: 'ShoppingCart', Komponente: ShoppingCart
* Pfad: Ingredients', Komponente: IngredientList

Zusätzich gibt es noch eine Wildcard Komponente, welche bei jeder inkorrekten Eingabe aufgerufen wird:

* Pfad: \*\*', Komponente: PageNotFound

# Zusätzliche Datenmodelle:

Um die Datenstruktur der Objekte (Rezepte, Kategorien und Zutaten) einheitlich zu gestalten werden diese eigenständig definiert. Hier eine kurze Auflistung der Objekte und deren Inhalte:

* RecipeList: Enthält alle Rezepte.
* Recipe:
  + Name
  + Url des genutzten Bildes
  + Beschreibung
  + Objekt: CategoryList
* CategoryList:
  + Objekt mit allen Kategorien
* Category:
  + Name
  + Objekt mit allen Zutaten
* IngredientList:
  + Objekt mit allen Zutaten
* Ingredient:
  + Name
  + Kategorie
  + Einheit
  + Anzahl
  + Anzahl im ShoppingCart

## KIassendiagramm:

