

Dokumentation Projekt Rezeptbuch

Anforderung:

Es soll eine Webanwendung mit dem Framework Angular und den zur Verfügung stehenden UI-Elementen aus Angular Material entwickelt werden.

Diese Webanwendung stellt ein Rezeptbuch dar, welches Folgende Funktionen beinhaltet:

- Eine Übersichtsseite der Rezepte, in der alle Verfügbaren Rezepte aufgelistet werden.
 - o In dieser Ansicht ist es möglich ein neues Rezept erstellen zu können oder die Details eines vorhandenen Rezeptes zu sehen
- Eine Detailansicht eines Rezeptes die, die Bestandteile eines Rezeptes auflistet werden
 - o Bestandteile eines Rezeptes sind:
 - Bild, Name, Zutaten, Zubereitung
- Ein Rezept neu erstellt werden, indem die Bestandteile eines Rezeptes neu eingegeben werden und darauffolgend das Rezept gespeichert wird
- Eine Einkaufsliste, in der Zutaten angezeigt werden
 - o Diese angezeigten Zutaten stammen entweder aus einem Rezept, welches die Funktion bietet alle Zutaten dieses Rezeptes zur Einkaufsliste hinzuzufügen

oder

 - o Durch eine Möglichkeit in der Einkaufsliste eine Zutat separat hinzuzufügen
 - o Die angezeigten Zutaten sollen nach ihrer Kategorie Gruppirt sein
- Eine neue Zutat erstellen, indem die Bestandteile einer Zutat neu hinzugefügt werden
 - o Bestandteile einer Zutat sind Name, Kategorie, Einheit

Konzeption:

Recipe – Komponente:

Die Aufgabe dieser Komponente ist das Anzeigen aller Rezepte. Die Darstellung erfolgt über Cards.

RecipesComponent
+recipes: Recipe
+constructor (recipeService:RecipeService
+getRecipes()

Recipe – Service:

Die Aufgabe dieses Service ist die Bereitstellung der zur Verfügung stehenden Rezepte. Rezepte werden als Observables behandelt, sodass alle Änderungen vorgenommen werden können.

RecipeService
+recipes: Recipe[]
+constructor ()
+getRecipes ()
+getRecipe (id)
+saveRecipe (recipe:Recipe)

Recipe – Edit – Komponente:

In dieser Komponente werden alle Bestandteile eines Rezeptes angezeigt. Hierbei werden Elemente aus dem Angular Material wie z.B. Input-Felder benutzt. Zudem ist es möglich ein Rezept zu bearbeiten und alle Zutaten eines Rezeptes über einen Button in die Einkaufsliste transferiert werden kann.

RecipeEditComponent
+panelOpenState: boolean
+isDisabled: boolean
+recipe: Recipe
+ingredients: Object[]
+isExisting: boolean
+compareFn: Object
+constructor (route:ActivatedRoute, shoppingListService:ShoppingListService, ingredientService:IngredientService, recipeService:RecipeService)
+compareByValue (f1, f2)
+editRecipe ()
+getRecipe ()
+getIngredients ()
+addToShoppingList ()
+save ()

Shopping-List-Komponente:

Aufgabe dieser Komponente ist das anzeigen aller Zutaten gruppiert nach ihrer Kategorie, die der Einkaufsliste hinzugefügt wurde. Zudem ist es möglich eine Zutat mit Mengenangabe über einen Dialog der Einkaufsliste hinzuzufügen.

ShoppingListComponent	IngredientDialog
+shoppingList: Object[]	
+ingredients: Object[]	
+constructor (dialog:MatDialog, ingredientService:IngredientService, shoppingListService:ShoppingListService)	+constructor (dialogRef:MatDialogRef<IngredientDialog>, data:DialogData, shoppingListService:ShoppingListService)
+openDialog ()	+onNoClick()
+deleteShoppingList ()	
+getShoppingList ()	
+getIngredients ()	

Shopping-List-Service:

Dieser Service stellt die Zutaten, die der Einkaufsliste hinzugefügt wurden zur Verfügung. Zudem nimmt er Zutaten entgegen die der Einkaufsliste hinzugefügt werden sollen. Diese werden in diesem Service nach ihrer Existenz sortiert sodass keine Dopplungen auftreten können.

ShoppingListService
+shoppingList: any[]
+constructor ()
+getShoppingList ()
+deleteShoppingList ()
+add (ingredients:Object[])

Ingredient-Create-Komponente:

Die Aufgabe dieser Komponente ist das ermöglichen eine Zutat neu anlegen zu können. Dies wird mit Input-Feldern Visuell dargestellt. Das Hinzufügen einer Zutat geschieht mit der Kommunikation des Ingredient-Services.

IngredientCreateComponent
+ingredient: Object
+constructor (router:Router, ingredientService:IngredientService)
+save ()

Ingredient-Service:

Dieser Service stellt alle vorhandenen Zutaten zur Verfügung.

IngredientService
+ingredients: Ingredient[]
+constructor ()
+getIngredients ()
+getIngredient (id)
+save (ingredient:Ingredient)

Page-Not-Found-Komponente:

Diese Komponente wird aufgerufen falls ein falscher Pfad angefragt wird.

Navbar-Komponente:

Diese Komponente stellt alle verfügbaren Pfade zur Navigation durch die Applikation Visuell zur Verfügung.

Header-Komponente:

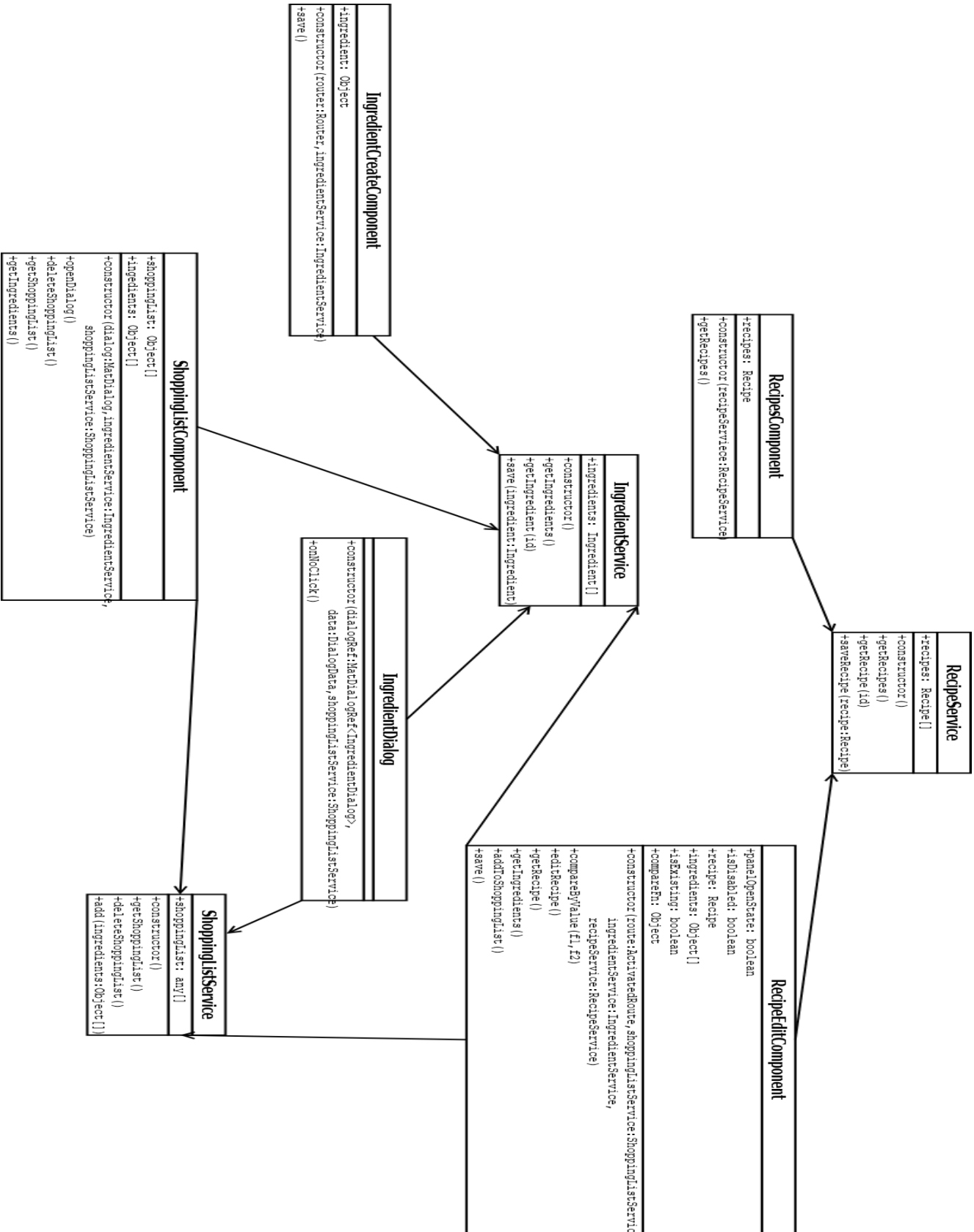
Diese Komponente ist für die jeweiligen Aktions-Buttons der anderen Komponenten verantwortlich.

Fazit:

Die Konzeption konnte vollständig implementiert werden. Alle Anforderungen an die Webanwendung sind damit erfüllt.

Schwierigkeiten gab es jedoch bei der Installation von Angular bzw. Angular Material, da Versionen da bestimmte Versionen der beiden Frameworks sowohl untereinander als auch mit gewissen Versionen von Node.js und WebPack nicht kompatibel sind. Dies hatte einen erhöhten Zeitaufwand zur Folge, da passende Versionen für alle nötigen Komponenten gefunden werden musste.

UML-Diagramm:



Paper – Mock-Up:

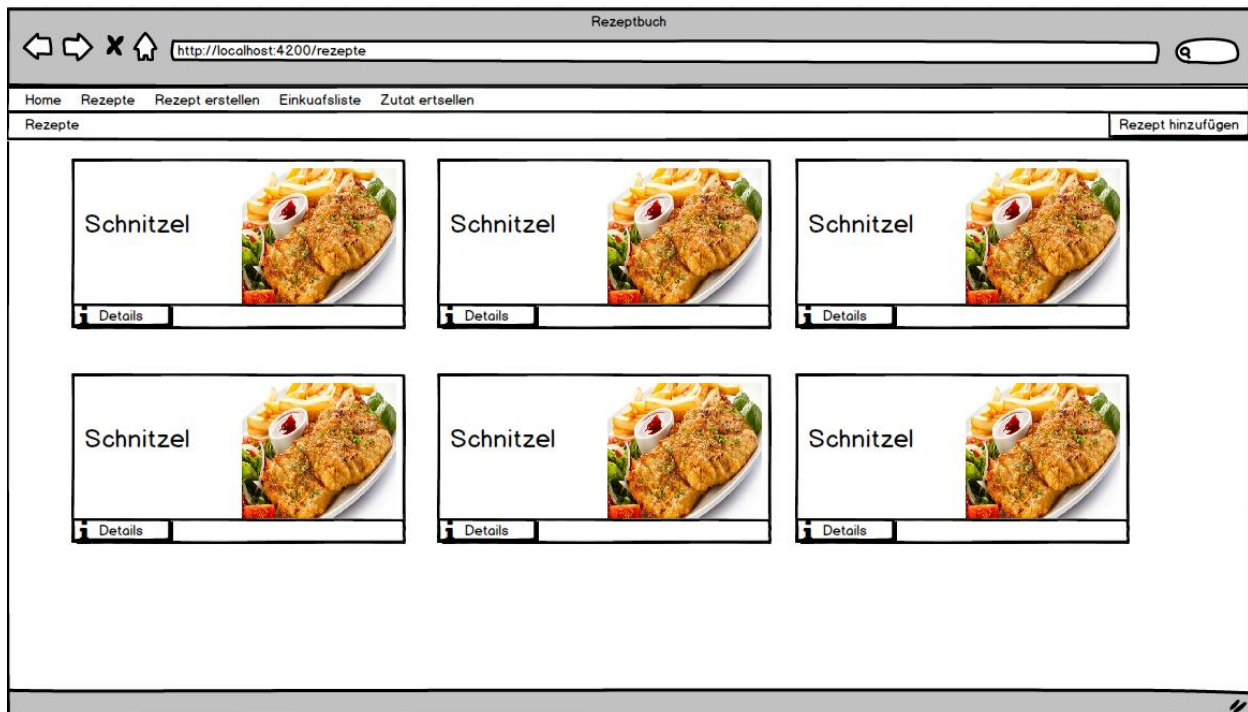


Abbildung 1: Übersicht der Rezepte

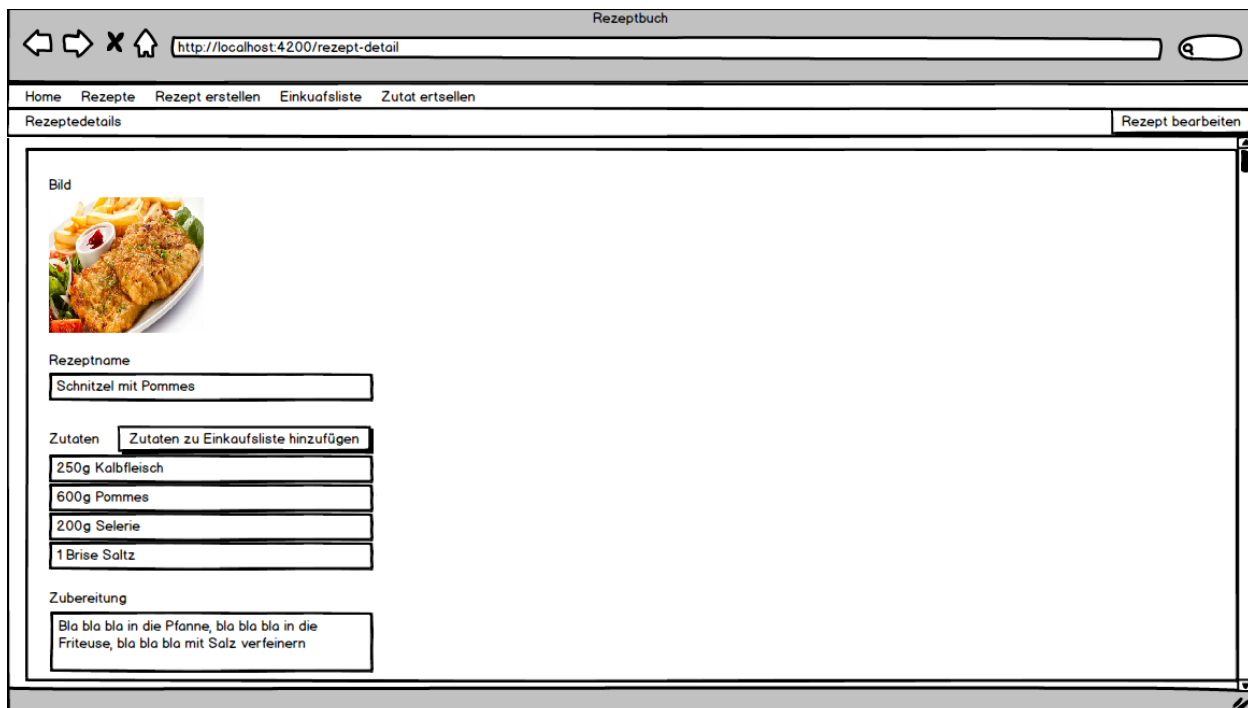


Abbildung 2: Ansicht/Bearbeitung/Anlegen eines Rezeptes

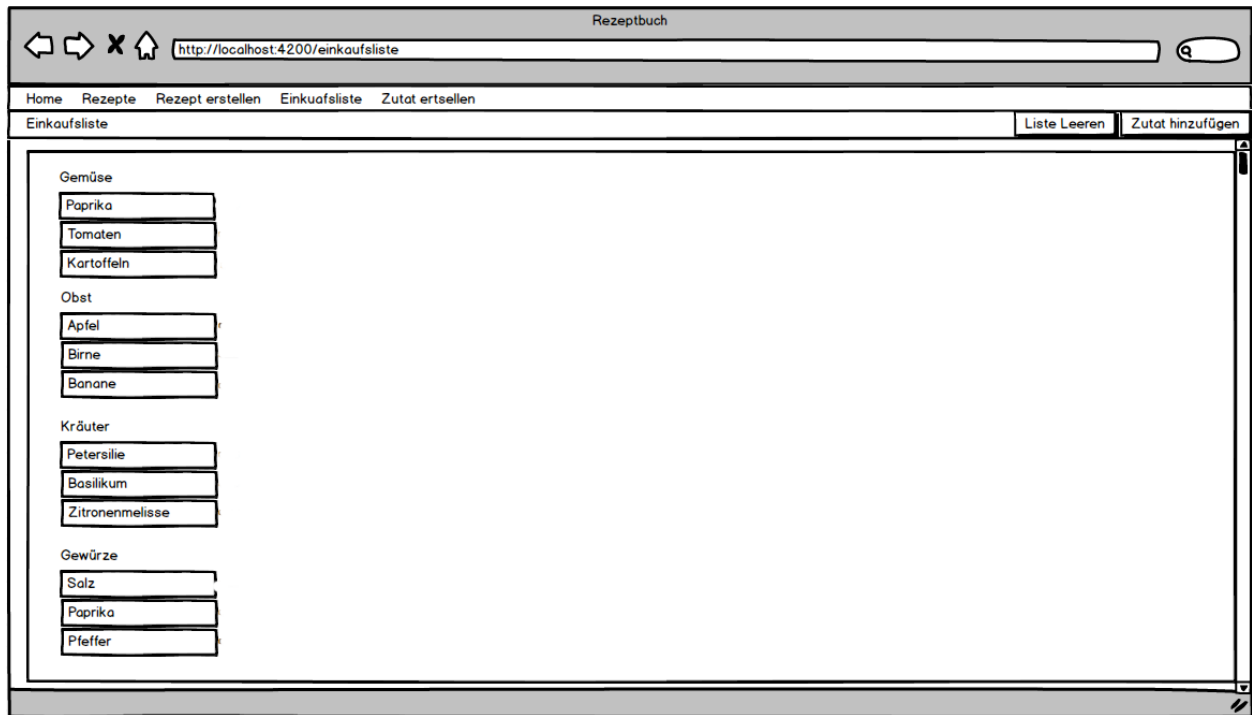


Abbildung 3: Einkaufsliste

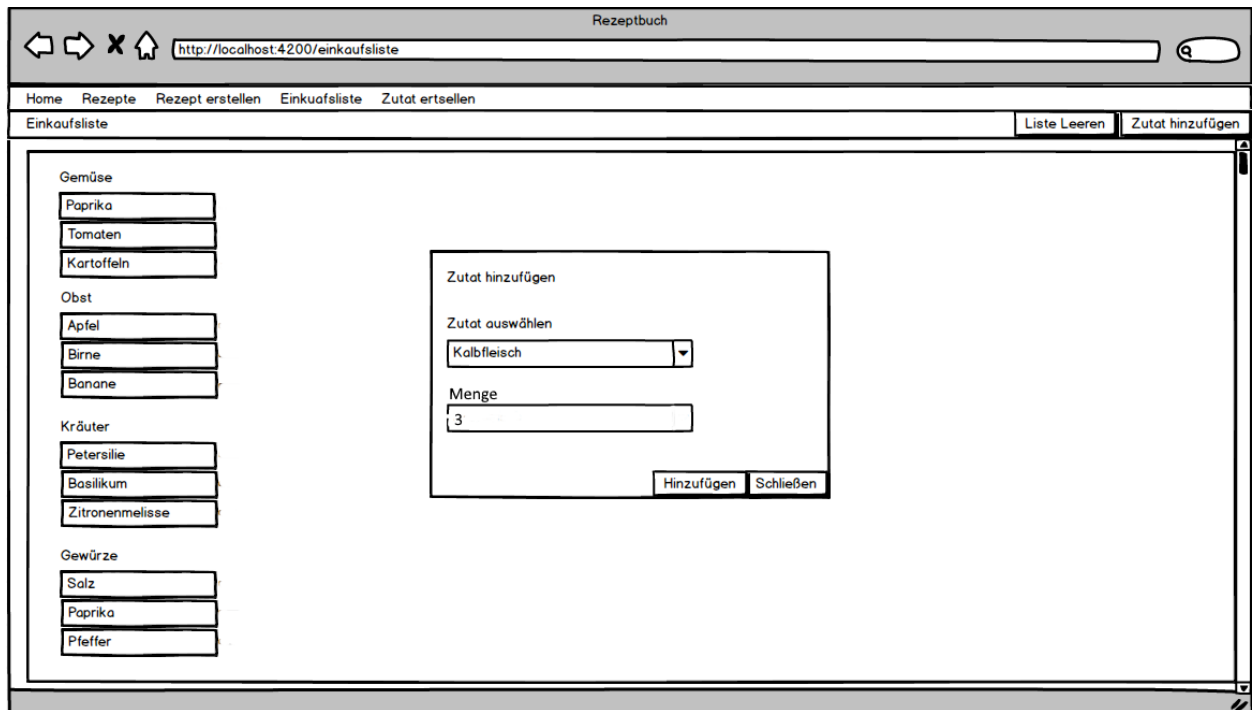


Abbildung 4: Zutat in der Einkaufsliste hinzufügen via eines Dialoges

Rezeptbuch

http://localhost:4200/zutat-erstellen

Home

Rezepte

Rezept erstellen

Einkaufsliste

Zutat ertstellen

Zutat erstellen

Name

Banane

Einheit

Stk

Kategorie

Obst

Abbildung 5: Erstellen einer neuen Zutat