



Eco Kitchen

«Web-Applikation zur Prävention von Foodwaste»

Gukhwa Jeon

TEKO Dipl. Technikerin HF Informatik
Applikationsentwicklung

Z-TIN-16-T-a
2019

Management Summary

Heutzutage gibt es viel Foodwaste. Alleine in der Schweiz werden pro Jahr total etwa 2.5 Millionen Tonnen Lebensmittel weggeworfen. Davon stammen etwa 40% aus Privathaushalten. Foodwaste zu minimieren ist für eine individuelle Person ein einfacher Weg, um die Umwelt zu schützen. Die Idee von der Web-Applikation ist, die Lebensmittelverschwendung im privaten Bereich ohne grossen Aufwand zu reduzieren.

Das Prinzip ist simpel: Wir sollen nur so viele Nahrungsmittel kaufen, wie wir essen können. Wir sollen essen, was wir schon gekauft und gekocht haben. Um das zu realisieren, sollten wir vorher schon wissen, was wir kochen und welche und wie viele Nahrungsmittel gekauft werden sollten. Jedoch ist dies in einer modernen Welt nicht so leicht, da man nach einem stressigen Arbeitstag nicht noch lange über das Kochen und Einkaufen nachdenken möchte. Diese Web-Applikation vereinfacht den ganzen Prozess der Rezeptverwaltung und des Einkaufens. Der Benutzer schreibt und speichert Rezepte mit allen Zutaten und den Mengenangaben, die er oft kocht. Der Zutatenrechner erstellt eine Einkaufsliste mit den berechneten Zutaten und Mengen von all den Rezepten, die er ausgewählt hat.

Vorteile der Applikation:

- Wir können durch das Erstellen der Einkaufslisten die Umwelt auf eine einfachere Art besser schützen.
- Wir müssen nicht alle Gerichte und Zutaten mühsam in Kochbüchern und Notizheften zusammensuchen. Alles ist an einem Ort gespeichert.
- Zutaten können ganz einfach in der benötigten Menge der Einkaufsliste hinzugefügt werden.
- Die Anzahl der Einkäufe kann durch diese Planung reduziert werden.
- Die Rezepte können mit anderen Benutzern geteilt werden.

Lebenslauf

BERUFSERFAHRUNG

03.2018 - 08.2018 **Swiss International Air Lines, Kloten, Schweiz**

Praktikum Application Management, Technical Development

Aufgaben:

- Design und Testen von elektronischen Arbeitsabläufen in Microsoft SharePoint
- Aktualisieren von Dokumentationen und geschäftlichen Arbeitsabläufen für das Flugzeugwartungssystem AMOS
- Vorbereiten von AMOS Test vor der Veröffentlichung

03.2010 - 02.2013 **Kyung Hee University, Suwon, Südkorea**

Forschungsassistentin in dem Integrated Circuits Interface Lab

Forschung: Mixed Mode Reconfigurable Interface

Aufgaben:

- Integrierter Schaltkreis Design, Simulation, Layout und Durchführung von Tests

03.2010 - 02.2011 **Kyung Hee University, Suwon, Südkorea**

Lehrassistentin für Schaltungstechnik Experimente und Electronische Schaltungen

Aufgaben:

- Durchführung von Schaltungstechnik Experimenten für Studenten
- Begleiten von Seminar für Hardware-Design-Software-Tools
- Kontrollieren und Benoten von Arbeiten der Studenten

STUDIUM

10.2016 - bis heute **TEKO Zürich, Schweiz**

Dipl. Technikerin HF Informatik, Applikation

03.2010 - 02.2013 **Kyung Hee University, Suwon, Südkorea**

Master of Engineering, Graduate School of Electronics and Radio Engineering

Masterarbeit: *'Reconfigurable CMOS analogue to pulse converter for analysis of EEG signals'*

03.2004 - 02.2010 **Kyung Hee University, Suwon, Südkorea**

Bachelor of Electronic Engineering, School of Electronics and Information

SPRACHEN

Englisch	fließend (Cambridge First Certificate of English: B2)
Deutsch	Gute Kenntnisse (telc Zertifikat Deutsch: B2)
Koreanisch	Muttersprache

INFORMATIK

Programmiersprachen:	Java, JavaScript
Sonstige Sprachen:	SQL, UML, HTML
Web-Technologien:	Node.js, React, Redux
Datenbanksysteme:	MS SQL Server, MongoDB, SQLite
Software IDE:	Eclipse, Android Studio
Administration:	MS SharePoint, Nintex Workflow
Anwendersoftware:	MS Office, Apple iWork
Betriebssysteme:	Windows, Linux, OSX
Hardware-Design-Sprachen:	Verilog, VHDL
Hardware-Design-Software:	Cadence Virtuoso von Mentor Graphics Calibre, Altera Quartus, Modelsim, Activ-HDL

Inhaltsverzeichnis

1.	Einleitung	5
2.	Aufgabenstellung	6
3.	Terminplan	7
4.	Analyse	8
4.1.	Systemkontext	8
4.2.	Systemanwendungsfälle	9
4.3.	Materialsammlung	14
4.4.	Anforderungen	14
5.	Datenbankdesign	17
6.	Implementation	18
6.1.	Technologien für das Projekt	18
6.1.1.	Back-End	18
6.1.2.	Front-End	20
6.2.	Architektur	23
6.2.1.	Back-End	23
6.2.2.	Front-End	29
7.	Reflexion	34
8.	Material	35
9.	Literaturverzeichnis	43
10.	Eigenständigkeitserklärung	44

1. Einleitung

In TEKO lernte ich die Programmiersprache Java und habe Erfahrung gesammelt, Desktop- und Android-Applikationen zu erstellen. Für dieses Diplomprojekt wollte ich eine praktische Applikation mit Technologien entwickeln, die von der Industrie am häufigsten verwendet werden.

Die Idee von 'Eco Kitchen' kam aus meiner persönlichen Erfahrung. Als ich in meinem Heimatland, Südkorea war, musste ich überhaupt nicht kochen, weil es immer viele Restaurants gab, wo man zu erschwinglichen Preisen gut essen konnte. Nachdem ich in die Schweiz gekommen war, musste sich dieser Lebensstil ändern und ich fing an zu kochen. Ich fand das Kochen und dessen Vorbereiten sehr mühsam. Oft habe ich Nahrungsmittel weggeworfen, weil ich diese gekauft habe, jedoch nicht zum Kochen benutzte. Deshalb brauche ich einen Service wie Eco Kitchen.

Der Kunde und der Entwickler dieser Applikation bin im Moment nur ich. Für die Nutzung wollte ich eine Applikation, die auf der iOS sowohl auch auf dem Desktop betrieben werden kann. Daher war die Entwicklung einer Web-Applikationen sehr attraktiv, da sie vom Desktop, Smartphone oder Tablet verwendet werden kann. Zusätzlich kann die Web-Applikation auch als Offline-Version verwendet werden.

In Kapitel 2, Aufgabenstellung, wird das Ziel des Projektes beschrieben und wie die Applikation funktionieren soll.

In Kapitel 3, Terminplan, stehen der ursprüngliche Terminplan und der tatsächliche Ablauf von dem Projekt.

In Kapitel 4, Analyse, werden Diagramme und Tabellen der Analysephase gezeigt, die mittels der Methode aus dem Software-Engineering-Kurs, den ich in TEKO besuchte, erstellt wurden.

In Kapitel 5, Datenbankdesign, sind Datenbankmodelle und die Beziehung beschrieben.

In Kapitel 6, Implementation, sind die Technologien meiner Wahl und der Grund beschrieben, weshalb ich sie ausgewählt habe. Des Weiteren werden die Paketdiagramme der Applikation aufgezeigt und erklärt.

In Kapitel 7, Reflexion, erwähnte ich, welche Schwierigkeiten ich während des Projekts hatte und was ich gelernt habe.

In Kapitel 8 habe ich die Screenshots von der Web-Applikation hinzugefügt, die in Kapitel 4.3, Materialsammlung, beschrieben sind.

2. Aufgabenstellung

Die Applikation wird für das Web geschrieben und hat den Zweck, dass Benutzer Rezepte auf dem Desktop schreiben und verwalten können.

Die Hauptfunktionen der Applikation beinhalten:

- Rezeptbuch: Der Benutzer schreibt Rezepte mit allen Zutaten und Mengen auf, die er oft kocht.
- Essplan: Der Benutzer wählt Rezepte aus dem Rezeptbuch und fügt sie der Essplan-Tabelle mit der gewünschten Anzahl Portionen hinzu.
- Zutatenrechner: Der Benutzer wählt Rezepte aus der Essplan-Tabelle aus und fügt die entsprechenden Zutaten dem Einkaufswagen hinzu. Der Zutatenrechner generiert einen Einkaufszettel mit allen Zutaten und Mengen nach Kategorien.
- Benutzerkonto: Der Benutzer speichert seine Rezepte sowie Esspläne und teilt sie mit den anderen Benutzern.

Zieldefinition: Eine Web-Applikation für den Haushalt zu programmieren.

<p>Die oben erwähnten vier Hauptfunktionen sind erfolgreich implementiert.</p> <p style="text-align: right;">Endergebnis</p>	<p>Menschen, die zu Hause kochen die einen PC und eine Internet Verbindung besitzen die am Umweltschutz interessiert sind die nicht jeden Tag überlegen möchten, was sie kochen und einkaufen sollen.</p> <p>Kunde</p>
<p style="text-align: right;">Sinn und Zweck</p> <p>Foodwaste zu minimieren Zeit zu sparen von wiederholender Hausarbeit</p>	<p>Erfolgskriterien</p> <p>Die oben erwähnten vier Hauptfunktionen sind erfolgreich implementiert.</p>

3. Terminplan

Der Ablauf des Projektes verlief anfänglich wie geplant. Jedoch war für die Implementation der Zeitaufwand sehr hoch, da es mein erster Versuch war, eine Web-Applikation zu programmieren. Es war aufwendig, passende Technologien zu finden. Ich wechselte mehrere Male die Programmiersprache, Plattform, Framework, Datenbanksystem und Programmbibliotheken. Deswegen arbeitete ich gleichzeitig an der Analyse und dem Design, wenn es Änderungen in der Funktion und dem Design gab.

Eine der Änderungen war der Austausch von der Essplan-Funktion zu der Lesezeichen-Funktion, da so der Zeitaufwand geringer ausfiel. Die beiden Funktionen sind ähnlich. Jedoch war die Lesezeichen-Funktion simpler, da die Benutzerschnittstelle einfacher zu implementieren ist und keine grossen Änderungen von den Datenbankmodellen und API-Diensten von dem Back-End-Server nötig waren.

SOLL

Tätigkeit	24.6	8.7	22.7	5.8	19.8	2.9	16.9	30.9	14.10
1. Aufgabenstellung									
2. Analyse									
3. Design									
4. Implementation									
5. Dokumentation									

IST

Tätigkeit	24.6	8.7	22.7	5.8	19.8	2.9	16.9	30.9	14.10
1. Aufgabenstellung									
2. Analyse									
3. Design									
4. Implementation									
5. Dokumentation									

4. Analyse

4.1. Systemkontext

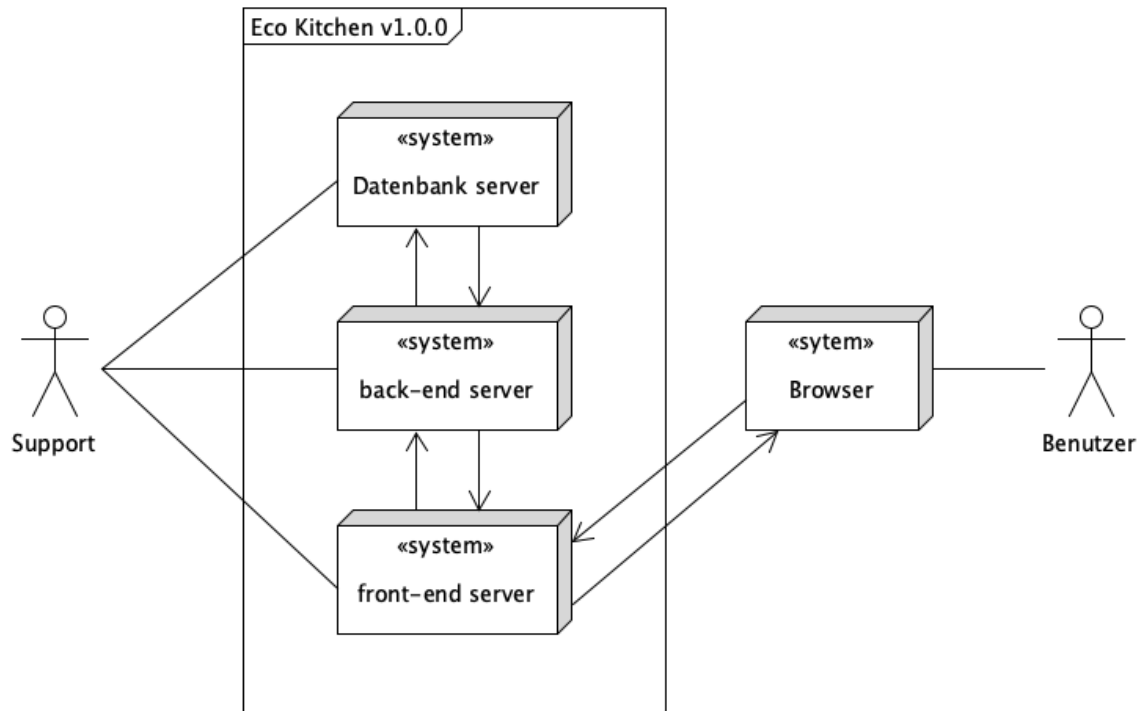


Abbildung.1 Systemkontext

Diese Applikation besteht aus diesen drei folgenden Systemen:

1. Der Front-End-Server rendert die Benutzeroberfläche und kommuniziert mit dem Browser des Clients und dem Back-End-Server.
2. Der Back-End-Server stellt HTTP-APIs bereit und kommuniziert mit dem Front-End-Server und dem Datenbanksystem.
3. Das Datenbanksystem verwaltet die persistenten Daten für diese Web-Applikation und kommuniziert mit dem Back-End-Server.

4.2. Systemanwendungsfälle

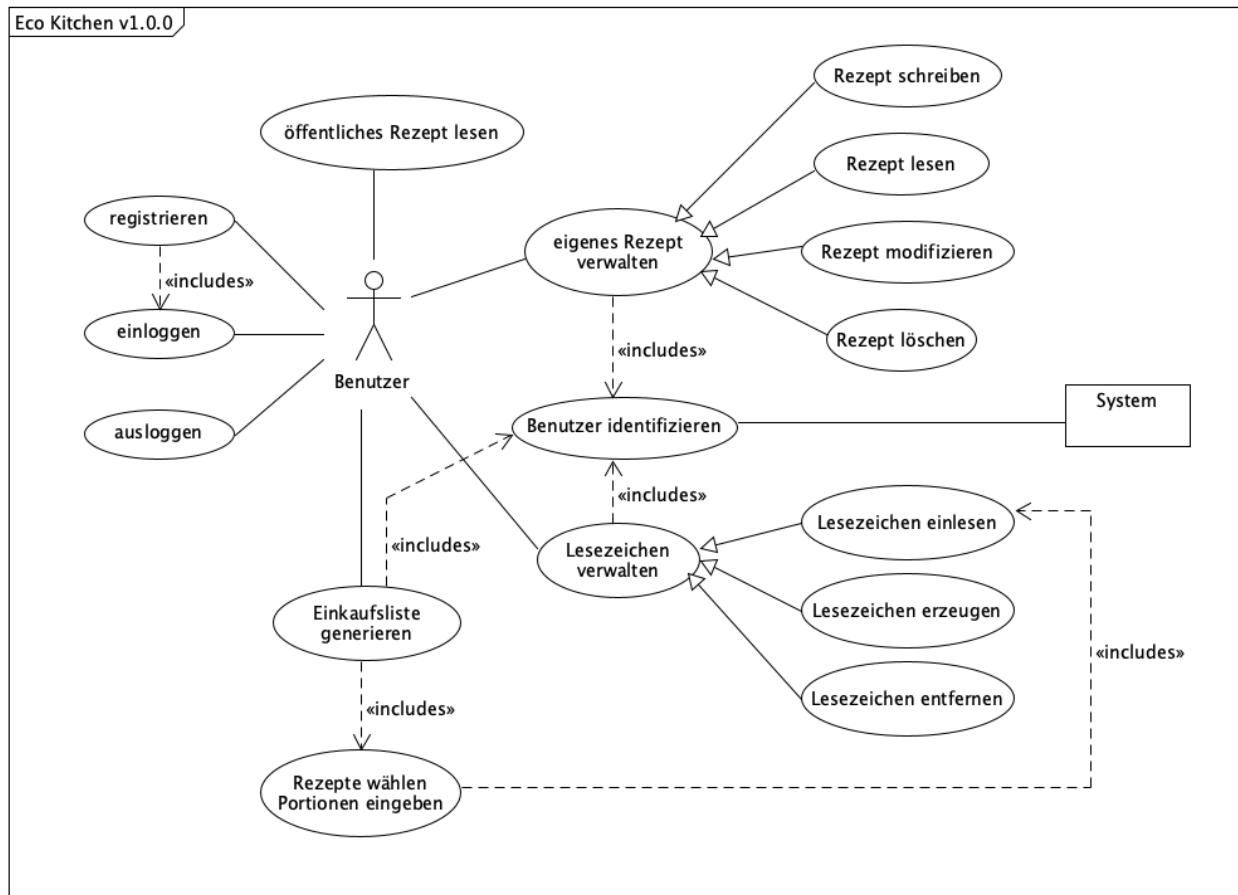


Abbildung.2 Systemanwendungsfälle

Diese sechzehn Systemanwendungsfälle können in fünf Gruppen unterteilt werden.

1. Nicht angemeldete Benutzer können nur öffentliche Rezepte lesen.
2. Ein Benutzer kann ein Konto besitzen.
3. Ein angemeldeter Benutzer speichert und verwaltet seine eigenen Rezepte.
4. Ein angemeldeter Benutzer speichert und verwaltet seine Lesezeichenliste.
5. Ein angemeldeter Benutzer erstellt eine Einkaufsliste, basierend auf seiner Lesezeichenliste.

Die nachfolgenden Tabellen beschreiben alle Systemanwendungsfälle.

Tabelle.1-1 Beschreibung des Systemanwendungsfalls - registrieren

Name	Registrieren
Kurzbeschreibung	Der Benutzer erstellt ein Konto
Akteure	Benutzer
Auslöser	Der Benutzer möchte ein Konto erstellen
Eingehende Daten	ID, Passwort
Vorbedingungen	Der Benutzer hat den Knopf 'registrieren' gedrückt
Ergebnis	Das System erstellt ein neues Benutzerkonto
Essenzieller Ablauf	1. Der Benutzer gibt eine ID ein 2. Der Benutzer gibt ein Passwort ein 3. Der Benutzer gibt das Passwort nochmals ein, um dieses zu bestätigen 4. Das System bestätigt die ID Validierung und erstellt ein Benutzerkonto 5. Das System loggt den Benutzer automatisch ein

Tabelle.1-2 Beschreibung des Systemanwendungsfalls - einloggen

Name	Einloggen
Kurzbeschreibung	Der Benutzer loggt sich ein
Akteure	Benutzer
Auslöser	Der Benutzer möchte sich einloggen
Eingehende Daten	ID, Passwort
Vorbedingungen	Der Benutzer ist registriert Der Benutzer hat den Knopf 'einloggen' gedrückt
Ergebnis	Der Benutzer ist eingeloggt
Essenzieller Ablauf	1. Der Benutzer gibt eine ID ein 2. Der Benutzer gibt ein Passwort ein 3. Das System identifiziert den Benutzer 4. Das System sendet die Authentifizierungsinformationen des Benutzers an den Browser

Tabelle.1-3 Beschreibung des Systemanwendungsfalls - ausloggen

Name	Ausloggen
Kurzbeschreibung	Der Benutzer loggt sich aus
Akteure	Benutzer
Auslöser	Der Benutzer möchte sich ausloggen
Eingehende Daten	-
Vorbedingungen	Der Benutzer ist eingeloggt Der Benutzer hat den Knopf 'ausloggen' gedrückt
Ergebnis	Der Benutzer ist ausgeloggt
Essenzieller Ablauf	Das System löscht die Authentifizierungsinformationen des Benutzers im Browser

Tabelle.1-4 Beschreibung des Systemanwendungsfalls - Benutzer identifizieren

Name	Benutzer identifizieren
Kurzbeschreibung	Das System identifiziert den Benutzer
Akteure	System
Auslöser	Der Benutzer möchte seine eigenen Rezepte verwalten Der Benutzer möchte seine Lesezeichen verwalten Der Benutzer möchte eine Einkaufsliste generieren
Eingehende Daten	-
Vorbedingungen	Der Benutzer ist eingeloggt
Ergebnis	Der Benutzer ist identifiziert
Essenzieller Ablauf	Das System prüft die Authentifizierungsinformationen von dem Browser

Tabelle.1-5 Beschreibung des Systemanwendungsfalls - öffentliches Rezept lesen

Name	Öffentliches Rezept lesen
Kurzbeschreibung	Der Benutzer liest ein öffentliches Rezept
Akteure	Benutzer
Auslöser	Der Benutzer möchte ein öffentliches Rezept lesen
Eingehende Daten	-
Vorbedingungen	A. Der Benutzer hat den Knopf 'Home' oder 'öffentliches Rezept' gedrückt B. Der Benutzer hat auf den Namen des Autors von einem Rezept geklickt C. Der Benutzer hat auf ein Schlagwort von einem Rezept geklickt
Ergebnis	Die öffentlichen Rezepte sind dargestellt
Essenzieller Ablauf	1-A. Der Browser stellt eine Liste aller Rezepte von allen Benutzern dar 1-B. Der Browser stellt eine Liste aller Rezepte von dem ausgewählten Benutzer dar 1-C. Der Browser stellt eine Liste mit allen Rezepten zusammen, die das ausgewählte Schlagwort enthalten 2. Der Benutzer klickt auf ein Rezept aus der Liste 3. Der Browser stellt das Rezept dar

Tabelle.1-6 Beschreibung des Systemanwendungsfalls - eigenes Rezept verwalten

Name	Eigenes Rezept verwalten
Kurzbeschreibung	Der Benutzer schreibt, liest, modifiziert und löscht sein eigenes Rezept (CRUD)
Akteure	Benutzer
Auslöser	Der Benutzer möchte sein Rezept verwalten
Eingehende Daten	A. ein neues Rezept schreiben Titel Standardportion Zutaten mit Kategorien und Mengen Sonstige Notizen, wie der Ablauf des Kochens Schlagwörter B. lesen -

	C. ein bestehendes Rezept modifizieren Titel Standardportion Zutaten mit Kategorien und Mengen Sonstige Notizen, wie der Ablauf des Kochens Schlagwörter D. löschen -
Vorbedingungen	Der Benutzer ist identifiziert
Ergebnis	A. schreiben Ein neues Rezept wurde gespeichert und dargestellt B. lesen Ein bestehendes Rezept wurde abgerufen und dargestellt C. modifizieren Ein bestehendes Rezept wurde modifiziert und dargestellt D. löschen Ein bestehendes Rezept wurde gelöscht
Essenzieller Ablauf	A. schreiben 1. Der Benutzer drückt den Knopf 'schreiben' 2. Der Benutzer gibt den erforderlichen Datensatz ein 3. Der Benutzer drückt den Knopf 'bestätigen' 4. Das System speichert den neuen Datensatz B: lesen 1. Der Benutzer klickt auf 'Meine Rezepte' in der Navigationsleiste 2. Der Browser stellt eine Liste aller Rezepte von dem Benutzer dar 3. Der Benutzer klickt auf ein Rezept aus der Liste 4. Der Browser stellt das Rezept dar C: modifizieren 1. Der Benutzer liest sein Rezept und drückt den Knopf 'modifizieren' 2. Der Benutzer gibt den Datensatz ein, der modifiziert werden soll 3. Der Benutzer drückt den Knopf 'bestätigen' 4. Das System erneuert den Datensatz D: löschen 1. Der Benutzer liest sein Rezept und drückt den Knopf 'löschen' 2. Der Benutzer drückt den Knopf 'bestätigen' 3. Das System löscht den Datensatz

Tabelle.1-7 Beschreibung des Systemanwendungsfalls - Lesezeichen verwalten

Name	Lesezeichen verwalten
Kurzbeschreibung	Der Benutzer verwaltet seine Lesezeichen
Akteure	Benutzer
Auslöser	Der Benutzer möchte seine Lesezeichen verwalten
Eingehende Daten	A. ein Rezept der Lesezeichenliste hinzufügen Rezept ID

	B. Lesezeichenliste lesen - C. ein Rezept aus der Lesezeichenliste entfernen Rezept ID
Vorbedingungen	Der Benutzer ist identifiziert
Ergebnis	A. ein Rezept der Lesezeichenliste hinzufügen Ein Rezept wurde auf die Lesezeichenliste gespeichert. B. Lesezeichenliste lesen Eine bestehende Lesezeichenliste wurde abgerufen. C. ein Rezept aus der Lesezeichenliste entfernen Ein Rezept wurde aus der Lesezeichenliste entfernt.
Essenzieller Ablauf	A. ein Rezept der Lesezeichenliste hinzufügen Der Benutzer drückt den Knopf 'Lesezeichen hinzufügen' auf der Rezept-Seite Das System fügt das Rezept zu der Lesezeichenliste hinzu B. Lesezeichenliste lesen Der Benutzer klickt auf 'Lesezeichen' in der Navigationsleiste Der Browser stellt eine Rezeptliste aus der Lesezeichenliste dar C: ein Rezept aus der Lesezeichenliste entfernen Der Benutzer drückt den Knopf 'Lesezeichen entfernen' auf der Rezept-Seite Das System entfernt das Rezept aus der Lesezeichenliste

Tabelle.1-8 Beschreibung des Systemanwendungsfalls - Einkaufsliste generieren

Name	Einkaufsliste generieren
Kurzbeschreibung	Der Benutzer generiert eine Einkaufsliste aus Rezepten der Lesezeichenliste
Akteure	Benutzer
Auslöser	Der Benutzer möchte eine Einkaufsliste generieren
Eingehende Daten	Rezepte IDs, Kochportionen
Vorbedingungen	Die Rezepte sind schon auf der Lesezeichenliste Der Benutzer klickt auf 'Einkaufen' in der Navigationsleiste
Ergebnis	Eine Einkaufsliste wurde im Browser dargestellt
Essenzieller Ablauf	1. Der Browser stellt eine Liste aller Rezepte aus der Lesezeichenliste dar 2. Der Benutzer fügt ein Rezept dem Einkaufswagen hinzu 3. Der Browser öffnet ein Pop-up, das die Anzahl der Kochportionen abfragt 4. Der Benutzer gibt die Anzahl der Kochportionen für das Rezept ein 5. Repetition von Punkt 1-4. So viele Rezepte dem Einkaufswagen hinzufügen, wie der Benutzer möchte 6. Der Benutzer drückt den Knopf 'Rechnen' 7. Das System addiert die Menge aller gleichen Zutaten der Rezepte in dem Einkaufswagen 8. Der Browser stellt eine gerechnete Zutatenliste nach Kategorien dar

4.3. Materialsammlung

Die Screenshots der Applikationsseiten können in Kapitel 8 gefunden werden.

Tabelle.2 Materialien

ID	Beschreibung
M1	Home-Page / Public-Recipes-Page (ausgeloggt)
M2	Login-Page
M3	Register-Page
M4	Home-Page / Public-Recipes-Page (eingeloggt)
M5A	Read-Recipe-Page (für das Rezept, das nicht im Lesezeichen gespeichert ist)
M5B	Read-Recipe-Page (für das Rezept, das schon im Lesezeichen gespeichert ist)
M6	Write-Recipe-Page
M7A	Shopping-Page (vor dem Generieren der Einkaufsliste)
M7B	Shopping-Page (nach dem Generieren der Einkaufsliste)

4.4. Anforderungen

Die nachfolgenden Tabellen beschreiben die Anforderungen.

Tabelle.3-1 Beschreibung der Anforderung – Datenmanipulations- und Dienstbeschränkung

Anforderung #1	Datenmanipulations- und Dienstbeschränkung
Art	Funktionale Anforderung
Beschreibung	Der Benutzer muss sich einloggen, um seine eigenen Rezepte zu bearbeiten, um Lesezeichen zu verwalten und um eine Einkaufsliste zu generieren.
Stabilität	Absolut stabil
Verbindlichkeit	Pflicht
Priorität	Hoch

Tabelle.3-2 Beschreibung der Anforderung - Benutzerfreundliches UI, um Einkaufsliste zu generieren

Anforderung #2	Benutzerfreundliches UI, um Einkaufsliste zu generieren
Art	Funktionale Anforderung
Beschreibung	Wenn der Benutzer Rezepte dem Einkaufswagen hinzufügt, kann er das durch die Drag-And-Drop Funktion ausführen.
Stabilität	Stabil
Verbindlichkeit	Pflicht
Priorität	Hoch

Tabelle.3-3 Beschreibung der Anforderung - Kategorien und Masseinheiten der Zutaten

Anforderung #3	Kategorien und Masseinheiten für Zutaten
Art	Funktionale Anforderung
Beschreibung	Der Benutzer kann selbst entscheiden, welche Kategorien und Masseinheiten für Zutaten angewandt werden
Stabilität	Stabil
Verbindlichkeit	Pflicht
Priorität	Hoch

Tabelle.3-4 Beschreibung der Anforderung - Private Rezepte

Anforderung #4	Private Rezepte
Art	Funktionale Anforderung
Beschreibung	Der Benutzer kann seine Rezepte auch privat speichern
Stabilität	Stabil
Verbindlichkeit	Wunsch
Priorität	Mittel

Tabelle.3-5 Beschreibung der Anforderung - Rezepte exportieren

Anforderung #5	Rezepte exportieren
Art	Funktionale Anforderung
Beschreibung	Der Benutzer kann seine eigenen Rezepte als PDF-Format exportieren
Stabilität	Stabil
Verbindlichkeit	Wunsch
Priorität	Mittel

Tabelle.3-6 Beschreibung der Anforderung - Rezepte von anderen Benutzern modifizieren

Anforderung #6	Rezepte von anderen Benutzern modifizieren
Art	Funktionale Anforderung
Beschreibung	Der Benutzer kann Rezepte von anderen Benutzern kopieren und modifizieren. Der Autor vom modifizierten Rezept ist dann der Benutzer selber und das modifizierte Rezept wird automatisch privat
Stabilität	Instabil
Verbindlichkeit	Wunsch
Priorität	Mittel

Tabelle.3-7 Beschreibung der Anforderung - Zutaten verwalten

Anforderung #7	Zutaten verwalten
Art	Funktionale Anforderung
Beschreibung	Der Benutzer kann seine Zutatenliste mit Kategorien und Masseinheiten bestimmen, um ein Rezept das nächste Mal einfacher zu schreiben Diese Funktion verhindert eine falsche Eingabe in der Zutatenliste
Stabilität	Flüchtig
Verbindlichkeit	Wunsch
Priorität	Mittel

Tabelle.3-8 Beschreibung der Anforderung - Essplan erstellen

Anforderung #8	Essplan erstellen
Art	Funktionale Anforderung
Beschreibung	Der Benutzer kann wöchentliche oder monatliche Esspläne erstellen und speichern
Stabilität	Flüchtig
Verbindlichkeit	Wunsch
Priorität	Mittel

5. Datenbankdesign

Die Namen der Variablen, der Funktionen, und der Verzeichnisse für die Applikation sind alle auf Englisch geschrieben. Daher sind alle Diagramme ab diesem Kapitel auf Englisch verfasst, da sie die Diagramme Namen der Variablen beinhalten.

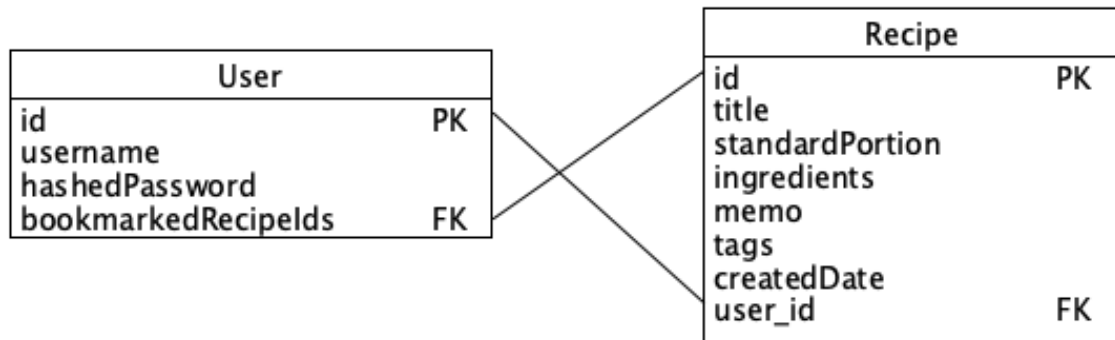


Abbildung.3 Datenbankmodell für Eco Kitchen Version 1

Es gibt zwei Datenbankschemen. 'User' ist für Benutzer, und 'Recipe' ist für Rezepte. Das Attribut 'bookmarkedRecipeIds' in 'User' ist ein Array des Attributes 'id' in 'Recipe'.

Die Beziehung zwischen den zwei Schemen ist:

- Ein Benutzer hat null bis mehrere Rezepte in der Lesezeichenliste.
- Ein Rezept hat nur einen Autor.

6. Implementation

6.1. Technologien für das Projekt

Meine Standards zur Entscheidung der Technologien für dieses Projekt:

- Sie müssen kostenlos sein.
- Sie sollten von der Industrie am häufigsten verwendet werden.
- Sie sollten eine grosse Community haben.
- Sie sollten keine grosse Lernkurve haben.

Die Haupttechnologien sind hier beschrieben. Andere wichtige Middleware und Tools wurden in Kapitel 6.2, Architektur, erwähnt.

Programmiersprache: JavaScript

Zu Beginn des Projektes verfügte ich über keine Programmiersprachekenntnisse, ausser Java. Das heisst, ich konnte die Sprache und das passende Framework wählen. Ich fand das Framework Django mit der Programmiersprache Python passend, da dort die Community sehr gross ist und zahlreiche Web-Applikationen mit Django erstellt wurden. Dazu war die Lernkurve von Python nicht so gross und mit Django konnte ich sowohl Front-End als auch Back-End programmieren.

Letztendlich entschied ich mich aus zwei Gründen für JavaScript:

1. Im Webengineering Kurs an der Schule begannen wir, eine Web-Applikation mit TypeScript, das auf JavaScript basiert, zu programmieren.
2. JavaScript ist, zusammen mit HTML und CSS, der Web-Standard und die am meisten verwendete Programmiersprache.

6.1.1. Back-End

Plattform: Node.js

Node.js ist eine serverseitige Plattform in der Softwareentwicklung zum Betrieb von Netzwerk-anwendungen. Insbesondere lassen sich damit Webserver realisieren. ^[1] Das ist die meist genutzte Plattform für Back-End-Server, die auf JavaScript basieren.

Framework: Koa ^[2]

Das Entwicklungsteam von Express, das beliebteste Webframework für Node.js, hat ein neues Webframework Namens Koa erstellt. Koa ist viel schlanker als Express, da man einen Server nur mit den Funktionen erstellen kann, die er benötigt. Dazu unterstützt Koa die 'async-await' Syntax von Node.js Version 7, und das erleichtert asynchrone Operationen.

Datenbank: MongoDB

MongoDB ist eine dokumentenorientierte NoSQL-Datenbank. Da die Datenbank dokumentenorientiert ist, kann sie Sammlungen von JSON-ähnlichen Dokumenten verwalten. So können viele Anwendungen Daten auf natürlichere Weise modellieren, da die Daten zwar in komplexen Hierarchien verschachtelt werden können, dabei aber immer abfragbar und indizierbar bleiben.^[3]

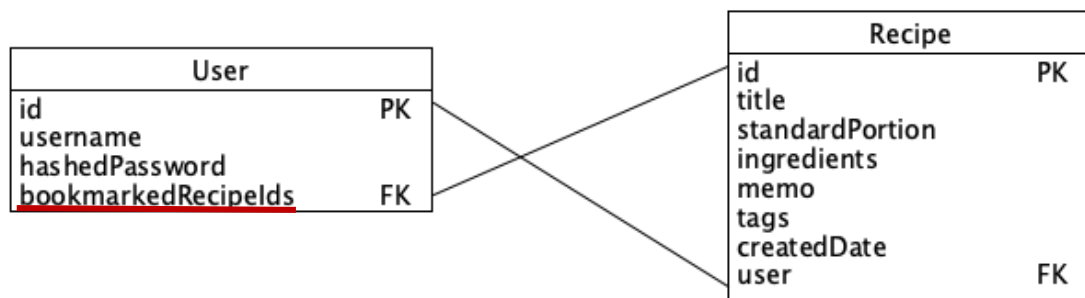


Abbildung.4 Datenbankmodell für Eco Kitchen Version 1

Diese Abbildung ist das Datenbankmodell von Kapitel 5, Datenbankdesign. Das Attribut 'bookmarkedRecipeIds' ist ein Array des Attributes 'id' in Recipe. Für Eco Kitchen Version 1 ist der Type des Attributes 'ingredients' String, aber für Eco Kitchen Version 2 soll es noch ein neues Schema für die Zutaten geben. Mit MongoDB wird die Datenverwaltung, wie im folgenden Beispiel aufgezeigt, erleichtert.

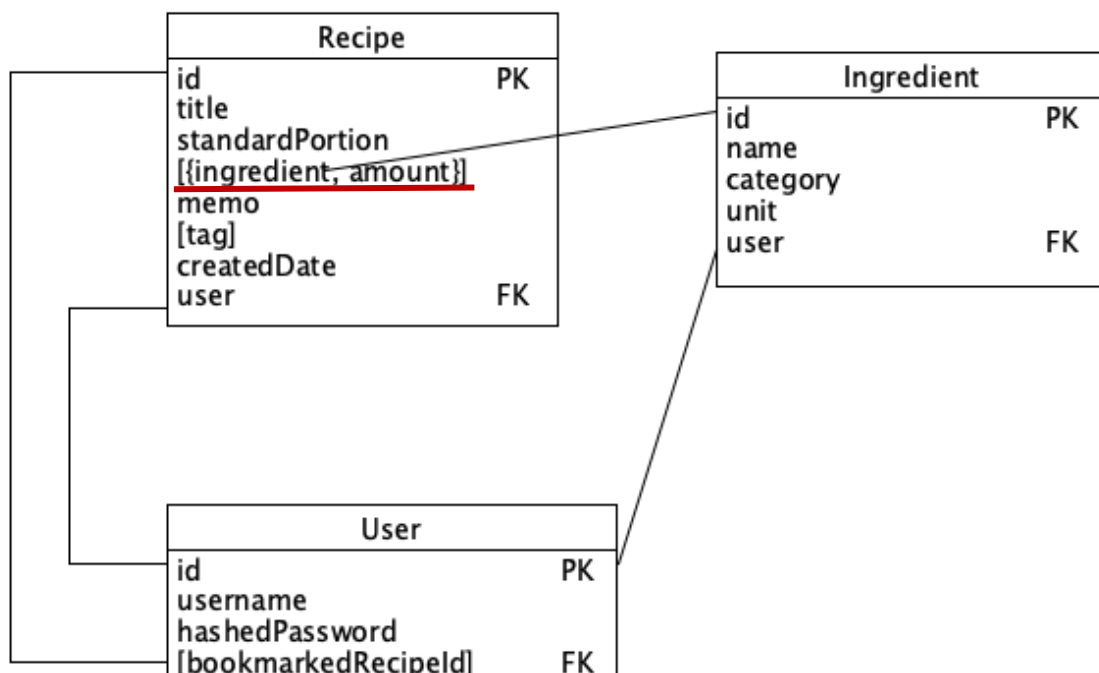


Abbildung.5 Datenbankmodell für Eco Kitchen Version 2

Ein Objekt-Array für die Zutatenliste ist im Schema 'Reciepe'. Und das Objekt im Array beinhaltet ein Attribut 'Ingredient', welches ein Objekt 'Ingredient' indiziert sowie ein anderes Attribut 'amount'. Diese Funktion ist auch praktisch für die Essplan-Funktion, da Rezeptobjekt und zusätzliche Attribute zusammen in einem Objekt des Arrays gespeichert werden müssen.

Ein weiterer Grund, weshalb ich MongoDB verwendete ist, dass es keinen Einfluss auf die alten Datensätze hat, obwohl das Schema geändert wurde. Während des Projektes wurde das Modell häufig geändert, mit MongoDB musste ich nicht immer alle alten Daten löschen oder ändern. Am Anfang der Implementation benutzte ich MySQL und SQLite, diese waren jedoch sehr mühsam für mein Datenbankmodell anzuwenden.

355 systems in ranking, October 2019

Rank			DBMS	Database Model	Score		
Oct 2019	Sep 2019	Oct 2018			Oct 2019	Sep 2019	Oct 2018
1.	1.	1.	Oracle +	Relational, Multi-model	1355.88	+9.22	+36.61
2.	2.	2.	MySQL +	Relational, Multi-model	1283.06	+3.99	+104.94
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model	1094.72	+9.66	+36.39
4.	4.	4.	PostgreSQL +	Relational, Multi-model	483.91	+1.66	+64.52
5.	5.	5.	MongoDB +	Document, Multi-model	412.09	+2.03	+48.90
6.	6.	6.	IBM Db2 +	Relational, Multi-model	170.77	-0.79	-8.91
7.	7.	↑ 8.	Elasticsearch +	Search engine, Multi-model	150.17	+0.90	+7.85
8.	8.	↓ 7.	Redis +	Key-value, Multi-model	142.91	+1.01	-2.38
9.	9.	9.	Microsoft Access	Relational	131.18	-1.53	-5.62
10.	10.	10.	Cassandra +	Wide column	123.22	-0.18	-0.17

Abbildung. 6 DB Ranking ^[4]

MongoDB ist das beliebteste, nicht relationale Datenbanksystem. Im September 2019 wurde es auf eine Nicht-Open-Source-Lizenz migriert, aber die Community Version steht kostenlos zur Verfügung.

6.1.2. Front-End

Programmbibliothek: React

React ist eine JavaScript-Softwarebibliothek, die ein Grundgerüst für die Ausgabe von User-Interface-Komponenten von Webseiten zur Verfügung stellt. Komponenten werden in React hierarchisch aufgebaut und können in dessen Syntax als selbst definierte HTML-Tags repräsentiert werden. React bildet typischerweise die Basis für Single-Page-Applikation. ^[5]

Einer der Gründe, warum ich mich für diese Bibliothek entschieden habe, war nicht die Leistung, sondern um meinen Code zu organisieren. Je mehr Web-Pages zum Anzeigen und je mehr Benutzerinteraktionen es gibt, desto schwieriger ist es, den Code zu verwalten. Dazu ist React eine der beliebtesten Technologien, Web-UI zu erstellen. Auch für spätere Versionen dieser Applikation gibt es die Möglichkeit, eine iOS und eine Android Applikation mit 'React Native' zu entwickeln.

Programmbibliothek: Redux

Das Mühsame bei der Verwendung von React war, den globalen 'State' und die 'Actions' zu verwalten. Es müssen häufig 'props' zwischen den übergeordneten und untergeordneten Komponenten ausgetauscht werden. Je tiefer die Komponententiefe ist, desto mehr überflüssige 'props' werden übermittelt. Die Zustandsverwaltungsbibliothek wurde benötigt, um dies zu vermeiden. Anstatt React Context-API zu benutzen, verwendete ich die Redux-Bibliothek, da sich die Entwicklungstools für das Debuggen als sehr hilfreich erwiesen.

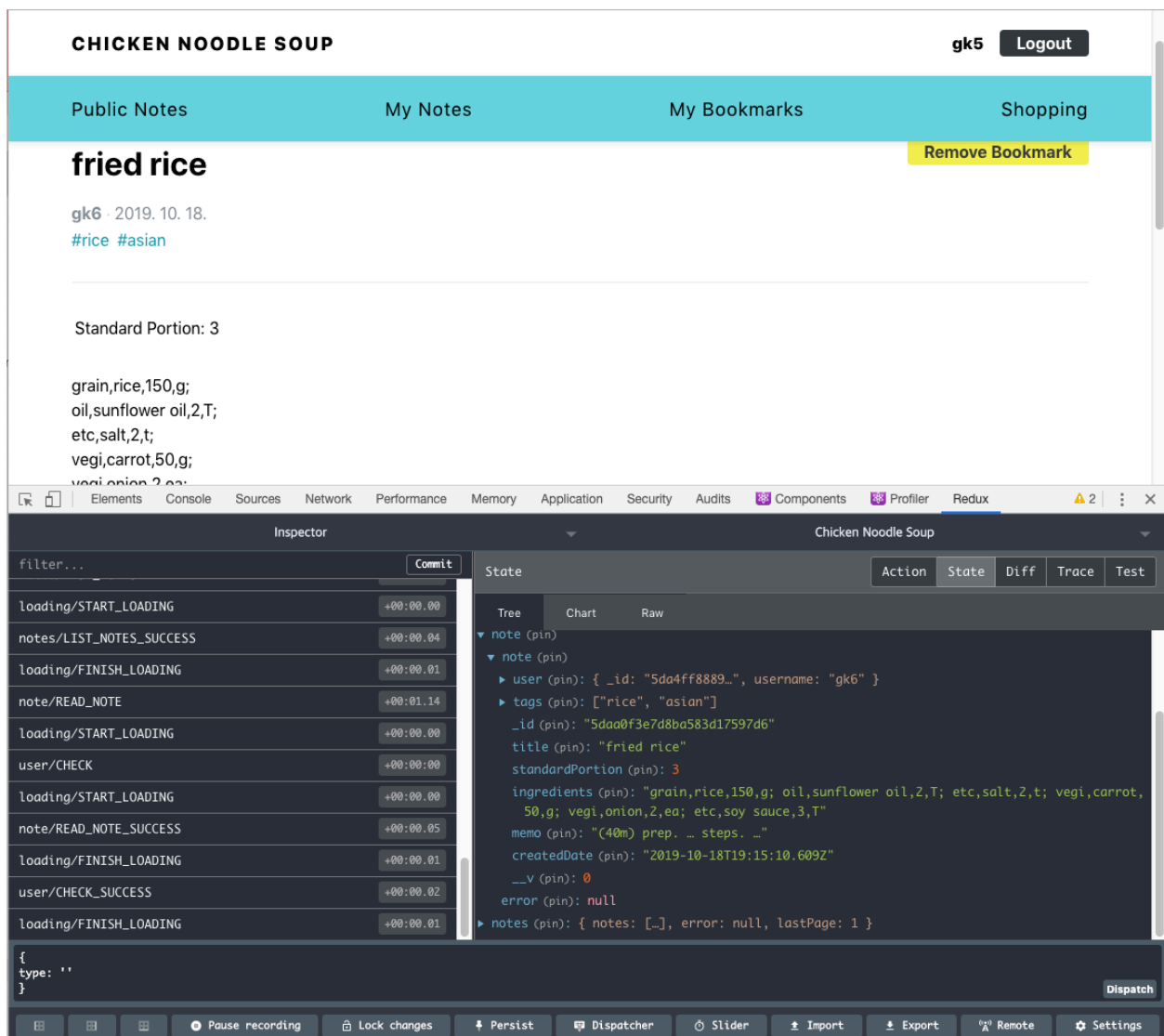


Abbildung. 7 Redux DevTools

Das linke Fenster zeigt die Reihenfolge der 'Actions', die durch Benutzerinteraktionen verursacht werden. Das rechte Fenster zeigt die 'State'. Beim Klick auf eine einzelne 'Action' im linken Fenster, erscheint gleichzeitig die 'State' im rechten Fenster vom Zeitpunkt als die 'Actions' ausgeführt wurden.

Programmbibliothek: styled-components ^[6]

‘styled-components’ ist die beste der vorhandenen React CSS-in-JS-Bibliotheken. Anstatt CSS-Dateien zu erstellen, kann man CSS in eine JavaScript-Datei schreiben. Die Programmbibliothek ‘styled-components’ wird auch in grossen ausländischen Unternehmen wie Atlassian, Reddit, Coinbase usw. verwendet.

Man kann den übergemittelten ‘props’ für die Komponenten, die mittels des ‘styled-components’ erstellt werden, verwenden. Diese Beispiele zeigen, wie ‘styled-components’ genutzt werden können.

```
const Box = styled.div`
  /* props können direkt gesetzt werden */
  background: ${props => props.color || 'blue'};
  padding: 1rem;
  display: flex;
`;
```

Code. styled-components: eine direkte Benutzung von ‘props’

```
import styled, { css } from 'styled-components';

const Button = styled.button`
  background: white;
  color: black;
  border-radius: 4px;
  padding: 0.5rem;
  display: flex;
  align-items: center;
  justify-content: center;
  box-sizing: border-box;
  font-size: 1rem;
  font-weight: 600;

  /* wenn ‘props.inverted === true’, diese CSS ist angewandt werden */
  ${props =>
    props.inverted &&
    css`
      background: none;
      border: 2px solid white;
      color: white;
    `
  };
  & + button {
    margin-left: 1rem;
  }
`;
```

Code. styled-components: eine Benutzung der bedingten Anweisung

Da der ‘CSS-Style’ sehr oft durch Benutzerinteraktionen geändert wird, verwendete ich anstelle von separaten CSS-Dateien, CSS-in-JS-Bibliotheken.

6.2. Architektur

Vor der Beschreibung der Implementationsprozesse möchte ich darauf hinweisen, dass der Programmcode in diesem Projekt auf dem Code von Blog-Tutorial im Buch ‘Art of React’ ^[2] basiert. Ich hatte grosse Bedenken hinsichtlich der Datenbankauswahl und der Benutzerauthentifizierungsmethoden. Das Buch war in dieser Hinsicht bei deren Auswahl sehr hilfreich. Die Methode der Benutzerauthentifizierung wird im folgenden Kapitel 6.2.1 Back-End beschrieben.

Zusätzlich möchte ich erwähnen, dass das Projekt nicht für die Verteilung vorgesehen ist, daher werden nur lokale Server verwendet.

6.2.1. Back-End

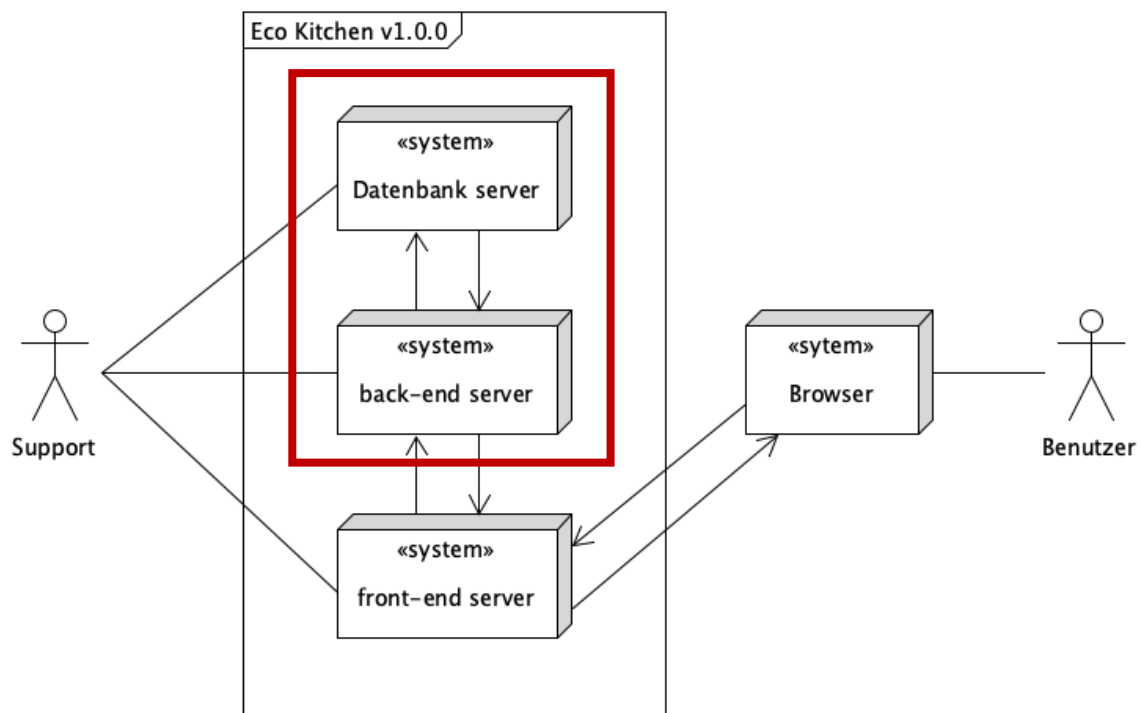


Abbildung.8 Systemkontext von der Analysephase

Die rote Box zeigt das Back-End-System. MongoDB verwaltet Benutzer- und Rezeptdateien. Der Koa-Back-End-Server liest oder manipuliert Daten von der MongoDB und antwortet dem React-Front-End-Server, wenn der Koa-Server die HTTP-Anfrage von dem React-Server bekommt.

Paketdiagramm

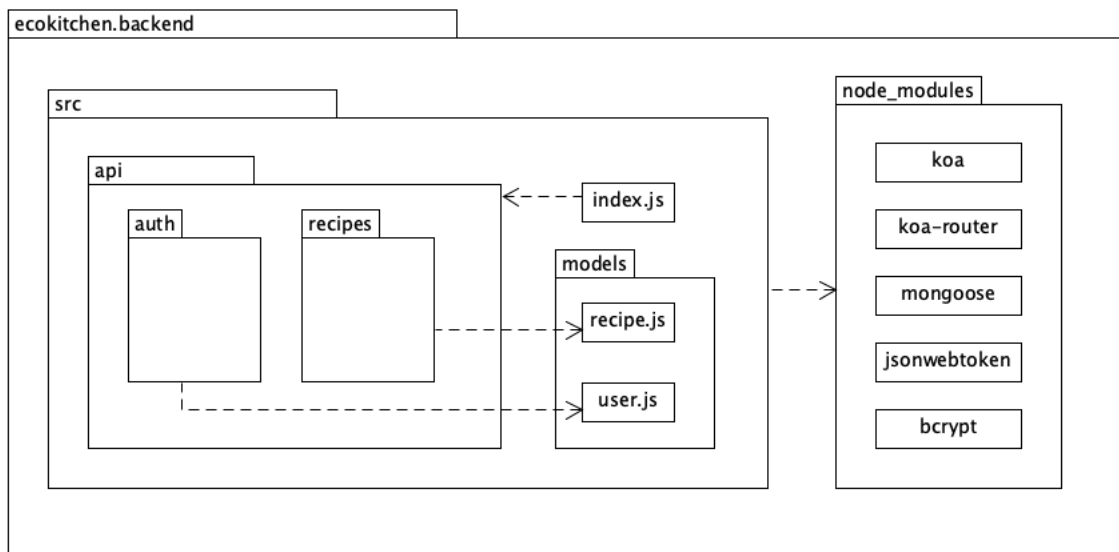


Abbildung.9 Paketdiagramm des Koa-Back-End-Server

Dieses Diagramm ist im Vergleich zu der tatsächlichen Struktur vereinfacht. Das Paket 'api' beinhaltet APIs für den Front-End-Server und benutzt das Paket 'models'. Im Paket 'models' sind die Datenbankschemen des Benutzers und des Rezepts mittels Programmbibliothek 'mongoose' definiert.

Programmbibliothek: mongoose

'mongoose' ist eine auf MongoDB basierende Objektdatenmodellierungsbibliothek, die in der Umgebung von node.js verwendet wird. Mit dieser Bibliothek kann das Datenbankdokument als JavaScript-Objekt verwenden werden. Unten ist ein Auszug von meinem Code abgebildet.

```
import mongoose, { Schema } from 'mongoose';

const UserSchema = new Schema({
  username: String,
  hashedPassword: String,
  bookmarkedNoteIds: [
    {
      type: mongoose.Types.ObjectId,
      ref: 'Recipe'
    }
  ]
});
```

Code. 'UserSchema' in user.js

Abbildung 10 zeigt dasselbe Schema als Diagramm. Das Attribut '_id' wird automatisch pro Dokument generiert.

User	
_id	PK
username	
hashedPassword	
bookmarkedRecipeIds: [Recipe_id]	FK

Abbildung.10 'UserSchema' Diagramm

Mit der 'populate' Methode kann leicht ein andres Dokument gelesen werden.

```
const result = User.findOne({ _id: user._id }).populate('bookmarkedNoteIds');
```

Code. Benutzung der 'UserSchema' und 'mongoose' im 'api' Packet

```
{
  "bookmarkedNoteIds": [
    "5da8cb02606ceb756a2e8237",
    "5da8c9ce606ceb756a2e8235",
  ],
  "_id": "5da381ae31472830ce1de568",
  "username": "gk5",
  "__v": 68
}
```

Beispiel API Antwort. Ohne 'populate' Methode

```
{
  "bookmarkedNoteIds": [
    {
      "user": {
        "_id": "5da4ff888928a348a712dd3f",
        "username": "gk6"
      },
      "tags": [
        "snack",
        "sandwich"
      ],
      "_id": "5da8cb02606ceb756a2e8237",
      "title": "banana sandwich",
      "standardPortion": 2,
      "ingredients": "fruit,banana,100,g;\nbread,toasbread,4,ea; ",
      "memo": "0.5 hr",
      "createdDate": "2019-10-17T20:11:46.639Z",
      "__v": 0
    },
    {
      "user": {
```

```
        "_id": "5da4ff888928a348a712dd3f",
        "username": "gk6"
    },
    "tags": [
        "chicken",
        "noodle",
        "soup"
    ],
    "_id": "5da8c9ce606ceb756a2e8235",
    "title": "chicken noodle soup",
    "standardPortion": 4,
    "ingredients": "meat,chicken,600,g; \nvegi,carrot,200,g; ",
    "memo": "1.5 hr",
    "createdDate": "2019-10-17T20:06:38.407Z",
    "__v": 0
  }],
  "_id": "5da381ae31472830ce1de568",
  "username": "gk5",
  "__v": 68
}
```

Beispiel API Antwort. Mit 'populate' Methode

Programmbibliothek: jsonwebtoken ^[7]

Es gibt zwei repräsentative Authentifizierungsmethoden, mit denen der Anmeldestatus des Benutzers auf dem Server verarbeitet werden kann. Eine ist die Authentifizierung, basierend auf der Session. Die andere Möglichkeit ist die Authentifizierung anhand von Token. Das Projekt verwendet ein Token-basiertes Authentifizierungssystem, da es einfacher ist, dieses zu implementieren.

Die Programmbibliothek 'jsonwebtoken' wird benutzt, um JSON-Web-Token zur Benutzerauthentifizierung zu generieren. JSON-Web-Token sind eine offene, branchenübliche RFC 7519-Methode zur sicheren Darstellung von Claims zwischen zwei Parteien. JSON-Web-Token wird von den meisten gängigen Programmiersprachen unterstützt.

Programm für die Entwicklung: Postman

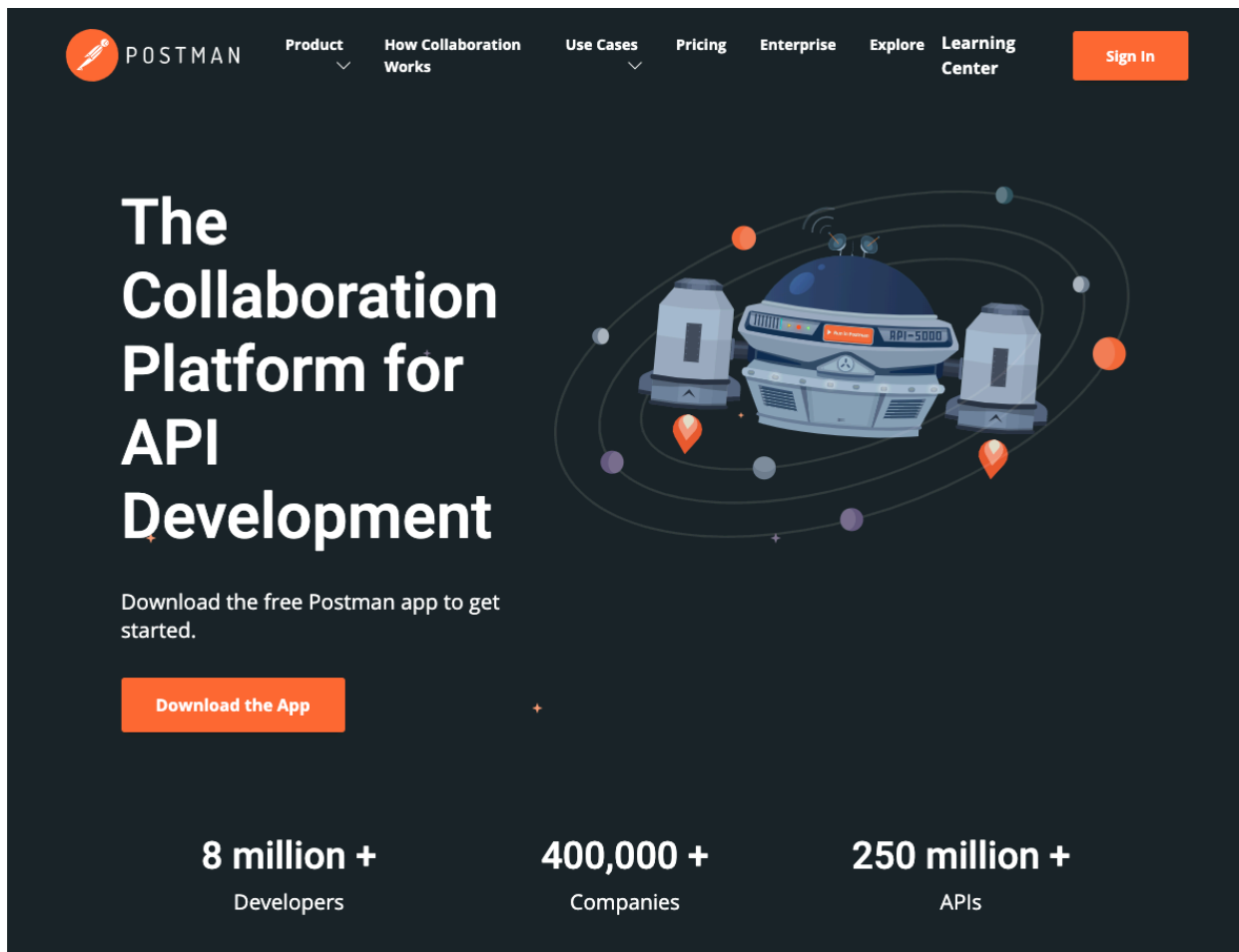


Abbildung.11 Postman Homepage ^[8]

‘Postman’ ist eine Kollaborationsplattform für die API-Entwicklung. Die Funktionen von Postman vereinfachen jeden Schritt beim Erstellen einer API und optimieren die Zusammenarbeit. So können APIs schneller erstellt werden. ’ ^[8]

Das Programm steht für das Mac OS, Windows, Linux und den Chrom Browser zur Verfügung. Für die Applikation wurden die HTTP-Anfragemethoden GET, POST, PATCH und DELETE benutzt. Mit dem Programm ‘Postman’ können auch solche Methoden getestet werden.

Die Abbildung 12 zeigt den HTTP-POST-Anfrage Test. Im oberen Fenster wird die Anfrage-Methode ausgewählt und die URL in das Eingabefeld geschrieben. Bei Bedarf kann im mittleren Bereich die Anfrage eingegeben werden. Beim Anklicken von ‘Send’, erscheint unten die API-Antwort.

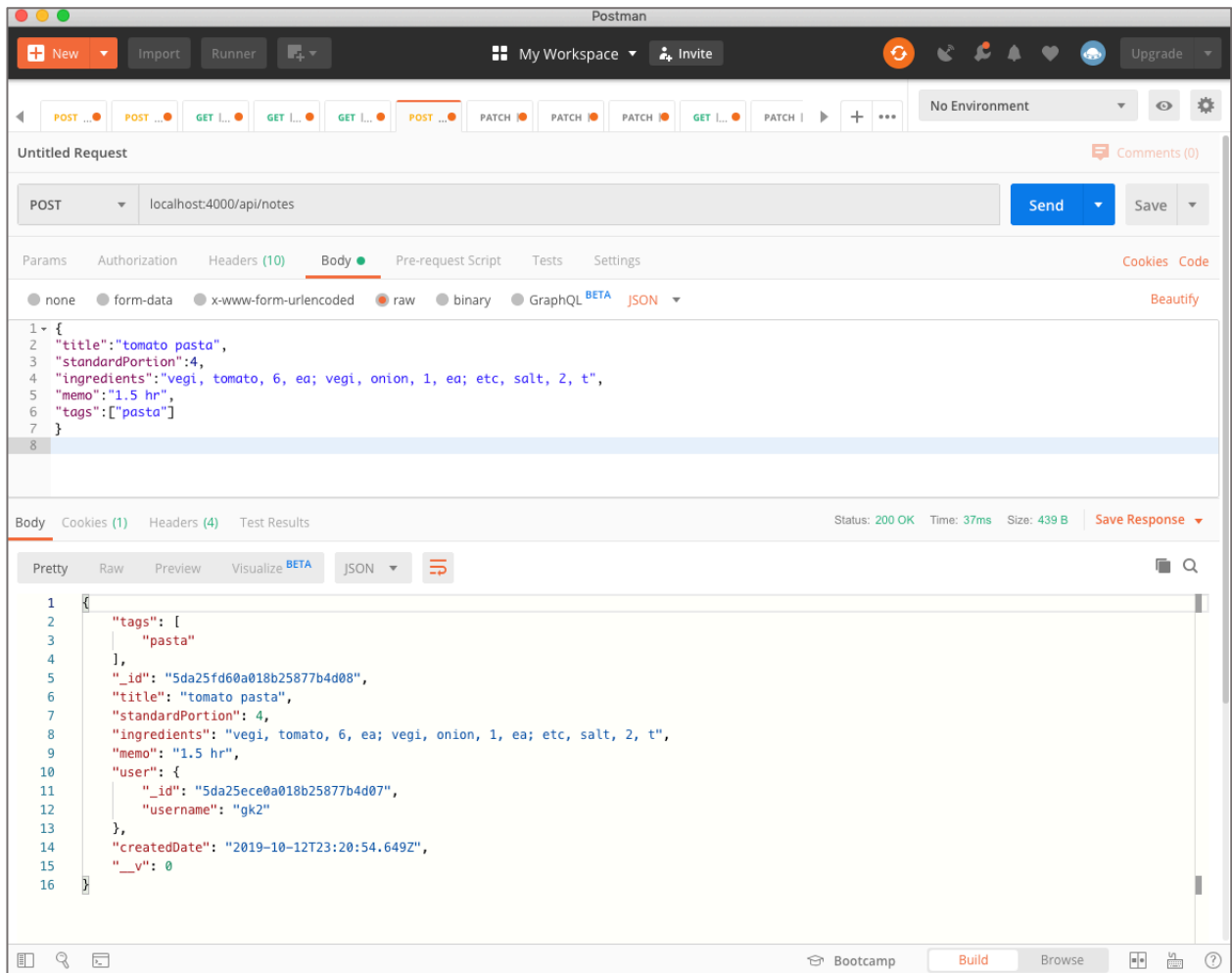


Abbildung.12 Postman Programm – POST-Anfrage

6.2.2. Front-End

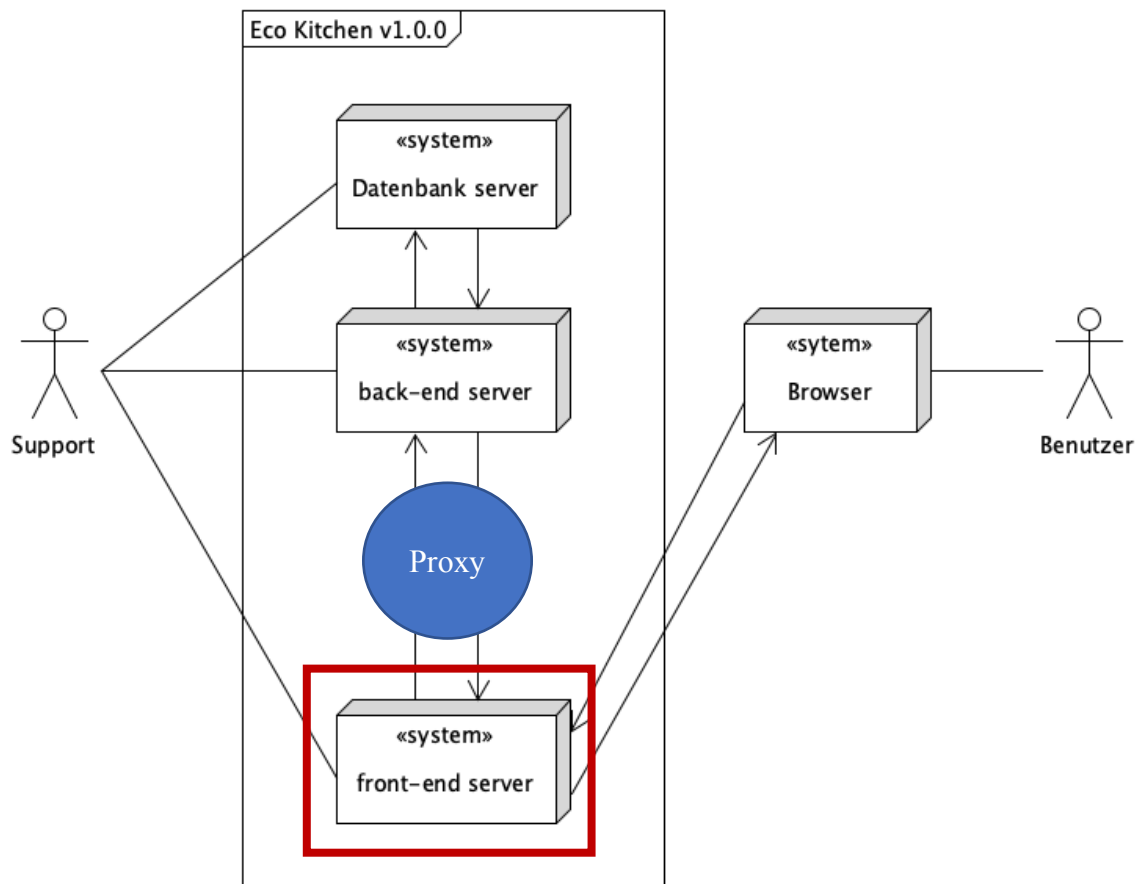


Abbildung.13 Systemkontext von der Analysephase

Die rote Box zeigt das Front-End-System. Zu Entwicklungszwecken sind der Koa-Back-End-Server mit 'localhost:4000' und der React-Front-End-Server mit 'localhost:3000' geöffnet. Die vom 'webpack-dev-server' unterstützte Proxy-Funktion verbindet die beiden Server.

Paketdiagramm

Die Abbildung 14 zeigt das Paketdiagramm des React-Front-End-Servers. Dieses Diagramm ist im Vergleich zu der tatsächlichen Struktur vereinfacht.

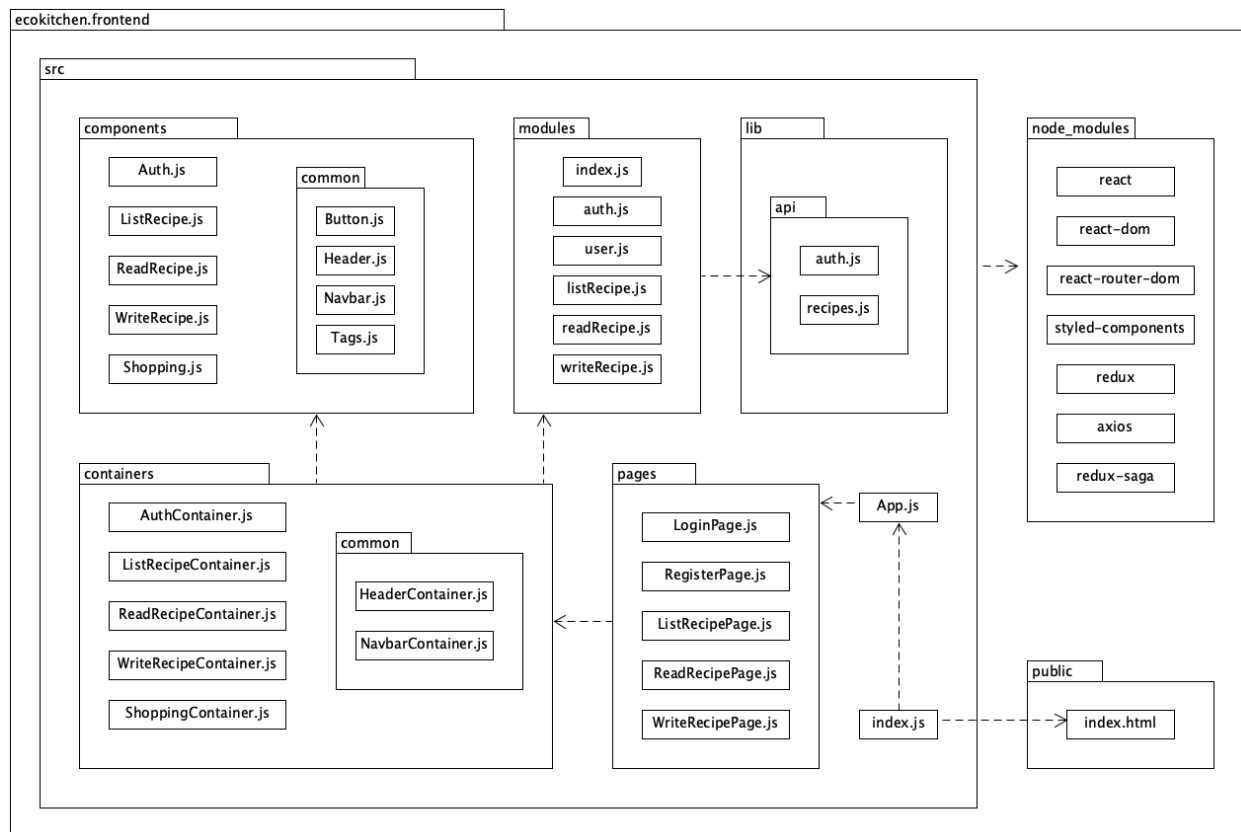


Abbildung.14 Paketdiagramm des React-Front-End-Servers

Alle JavaScript-Dateien, die mit einem Grossbuchstaben beginnen, sind Komponenten für die Benutzeroberfläche. Die Erklärung von diesem Paketdiagramm beginnt von der obersten Ebene der Applikation.

index.js und App.js

Das Paket 'react-dom' in 'node-modules' enthält DOM-spezifische Methoden, die auf der obersten Ebene der Applikation als Escape-Luke zum Verlassen des React-Modells verwendet werden. ^[9] 'index.js' im Paket 'src', welche die oberste Ebene der React Applikation ist, greift auf das 'root' Element in 'index.html' im Paket 'public' zu und rendert 'App.js'.

'App.js' verwendet die Router-Funktion für die fünf Applikationsseiten im Paket 'pages'.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import { BrowserRouter } from 'react-router-dom';
```

```
ReactDOM.render(  
  <BrowserRouter>  
    <App />  
  </BrowserRouter>,  
  document.getElementById('root'),  
);
```

Code. index.js im Paket 'src'

```
...  
  
const App = () => {  
  return (  
    <>  
      <Route component={ListRecipePage} path={['/@:username', '/']} exact />  
      <Route component={RegisterPage} path="/register" />  
      <Route component={LoginPage} path="/login" />  
      <Route component={WriteRecipePage} path="/write" />  
      <Route component={ReadRecipePage} path="/@:username/:RecipeId" />  
      <Route component={ShoppingPage} path="/shopping" />  
    </>  
  );  
};  
  
export default App;
```

Code. App.js im Paket 'src'

Paket 'pages'

JavaScript Dateien im Paket 'pages' rendern die entsprechenden Container im Paket 'containers'.

```
...  
  
const ShoppingPage = () => {  
  return (  
    <>  
      <HeaderContainer />  
      <NavbarContainer />  
      <ShoppingContainer />  
    </>  
  );  
};  
  
export default ShoppingPage;
```

Code. ShoppingPage.js im Paket 'pages'

Paket 'containers'

Die eigentliche Logik ist in den Container geschrieben. Sie benutzen das entsprechende Redux-Modul im Paket 'modules' und rendern entsprechende Komponenten im Paket 'components'.

Paket 'modules'

Das Redux-Modul im Paket 'modules' verwaltet 'Actions' und 'State' der Applikation. Das Modul benutzt die entsprechende Datei im Paket 'api'. 'redux-saga' und wird für die Verwaltung der asynchronen Operationen benutzt.

Programmbibliothek: **redux-saga** ^[10]

'redux-saga' zielt darauf ab, Applikationsnebeneffekte einfacher zu verwalten, effizienter auszuführen, einfacher zu testen und besser mit Fehlern umzugehen. Eine Saga ist wie ein separater Thread in der Applikation, der ausschließlich für Nebenwirkungen verantwortlich ist. 'redux-saga' ist eine Redux-Middleware, das heisst, dieser Thread kann mit normalen Redux-Actions von der Hauptapplikation aus gestartet, angehalten und abgebrochen werden. Er erlaubt Zugriff auf den vollständigen Redux-Applikation-State und kann auch Redux-Actions auslösen.

Paket 'api'

Alle HTTP-Anfragemethoden sind geschrieben, um mit dem Back-End-Server zu kommunizieren. 'axios' wird für HTTP-Anfragen benutzt. Die Programmbibliothek 'axios' ist der beliebteste JavaScript-HTTP-Client.

```
import axios from 'axios';

const client = axios.create();

// login
export const login = ({ username, password }) =>
  client.post('/api/auth/login', { username, password });

// register
export const register = ({ username, password }) =>
  client.post('/api/auth/register', { username, password });

// check
export const check = () => client.get('/api/auth/check');

// logout
export const logout = () => client.post('/api/auth/logout');
```

Code. auth.js im Paket 'api'

Paket ‘components’

Dieses Paket enthält alle Präsentationskomponenten, die mit der Programmbibliothek ‘styled-components’ gezeichnet wurden.

7. Reflexion

Während des Projekts lernte ich viel über die Projektarbeit und die Webentwicklung.

Der erste Punkt, aus dem ich gelernt habe ist, nicht etwas zu planen und umzusetzen, was ich vorher in dieser Art noch nie gemacht habe. Da dieses Projekt schlussendlich viel mehr Zeit in Anspruch genommen hat, als der eigentlich geplante Zeitaufwand für die Diplomarbeit der Techniker HF.

Es war schwierig mich zu entscheiden, welche Technologien ich für dieses Projekt verwenden soll. Auch nachdem ich mich entschieden habe, gab es nicht genügend Zeit, um mich an die neue Programmiersprache, an die neuen Tools und Programmbibliotheken zu gewöhnen. Ich musste so schnell wie möglich vorgehen, um die Applikation rechtzeitig fertig zu stellen. Einige Anforderungen in Kapitel 4.4 konnte ich schlussendlich aufgrund des überbordenden Zeitaufwandes nicht mehr umgesetzt, was ich sehr bedaure.

Der zweite Punkt, den ich erwähnen möchte, ist die Dokumentation. Technische Begriffe vom Englischen ins Deutsche zu übersetzen war oft verwirrend. Es war schwierig herauszufinden, ob das übersetzte deutsche Wort tatsächlich verwendet wird, da fast alle offiziellen Dokumente der von mir verwendeten Technologien nur auf Englisch geschrieben waren. Ausserdem benutzte ich englische Wörter für den Code, sowie für die Diagramme und die Applikation. Es wäre besser, solche Dokumente auf Englisch zu schreiben.

Der dritte Punkt ist, dass man sich bezüglich der Web-Sicherheit um viele Dinge kümmern muss. Um das Injizieren von Malware zu verhindern, wurde die Programmbibliothek 'sanitize-html' verwendet. Um Cookies, die JWT enthalten, vor Malware zu schützen, musste die Option für Cookie 'httpOnly' eingestellt werden. Würde das Benutzerpasswort als einfacher Text gespeichert, würde das sehr grosse Problem nach sich ziehen. Um das Passwort zu schützen, wird die Programmbibliothek 'bcrypt' verwendet. Ich vermute, es gibt viel mehr Dinge zu bedenken und zu lernen bezüglich der Web-Sicherheit.

Das Endergebnis des Projektes war in Ordnung, aber nicht ganz zufriedenstellend. Trotzdem bin ich froh, dass ich endlich eine Web-Applikation mit modernen Technologien erstellt habe. Das war mein Hauptziel für dieses Projekt.

8. Material

M1: Home-Page / Public-Recipes-Page (ausgeloggt)

CHICKEN NOODLE SOUP

Login

Public NotesMy NotesMy BookmarksShopping

fried rice

gk6 · 2019. 10. 18.
[#rice](#) [#asian](#)

(40m) prep. ... steps. ...

banana sandwich

gk6 · 2019. 10. 17.
[#snack](#) [#sandwich](#)

0.5 hr

banana shake

gk6 · 2019. 10. 17.
[#drinks](#)

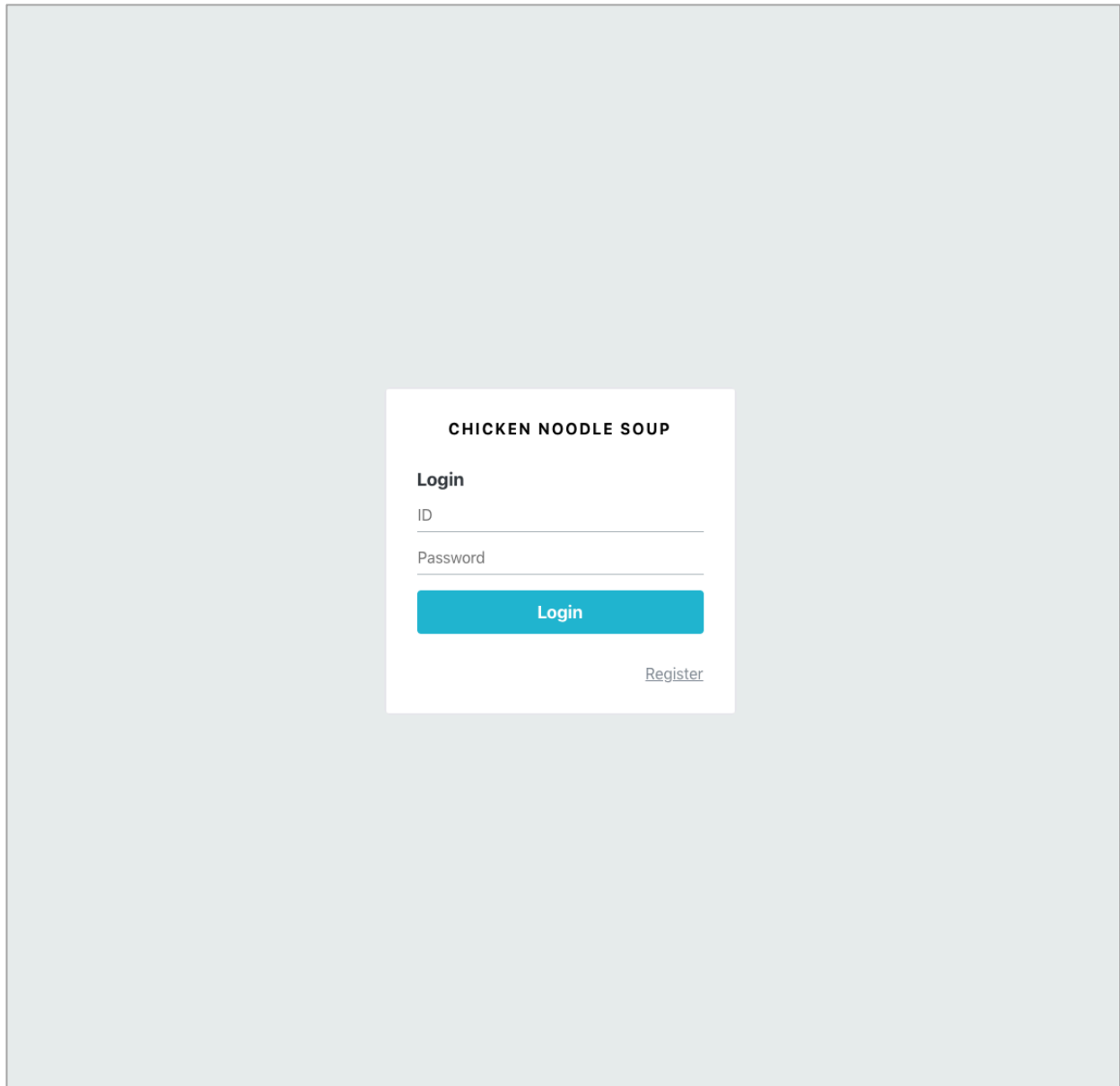
prep.6hr 1. prepare ice cube steps 1.pill the banana 2.put every ingredients in the mixer 3.enjoy

chicken noodle soup

gk6 · 2019. 10. 17.
[#chicken](#) [#noodle](#) [#soup](#)

1.5 hr

M2: Login-Page



The image shows a login page for a system named "CHICKEN NOODLE SOUP". The page has a light gray background. In the center, there is a white rectangular box containing the login form. The form has a title "CHICKEN NOODLE SOUP" in bold. Below the title, the word "Login" is displayed. There are two input fields: one for "ID" and one for "Password". Below these fields is a blue button labeled "Login". To the right of the button, there is a link labeled "Register".

CHICKEN NOODLE SOUP

Login

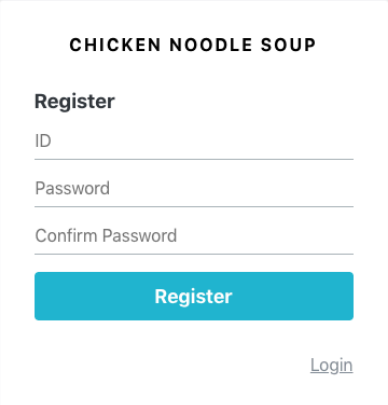
ID

Password

Login

[Register](#)

M3: Register-Page



The image shows a registration form titled "CHICKEN NOODLE SOUP" centered on a light gray background. The form is a white rectangle containing the following elements:

- CHICKEN NOODLE SOUP**: The title of the application, centered at the top of the form.
- Register**: A section header for the registration process.
- ID**: A text input field for the user's ID.
- Password**: A text input field for the user's password.
- Confirm Password**: A text input field for confirming the password.
- Register**: A blue button with white text for submitting the registration.
- Login**: A text link for existing users to log in.

M4: Home-Page / Public-Recipes-Page (eingeloggt)

CHICKEN NOODLE SOUP

gk5 Logout

Public NotesMy NotesMy BookmarksShopping

New Note

fried rice
gk6 · 2019. 10. 18.
[#rice](#) [#asian](#)

(40m) prep. ... steps. ...

banana sandwich
gk6 · 2019. 10. 17.
[#snack](#) [#sandwitch](#)

0.5 hr

banana shake
gk6 · 2019. 10. 17.
[#drinks](#)

prep.6hr 1. prepare ice cube steps 1.pill the banana 2.put every ingredients in the mixer 3.enjoy

chicken noodle soup
gk6 · 2019. 10. 17.
[#chicken](#) [#noodle](#) [#soup](#)

1.5 hr

M5A: Read-Recipe-Page (für das Rezept, das nicht im Lesezeichen gespeichert ist)

CHICKEN NOODLE SOUPgk5 Logout

Public NotesMy NotesMy BookmarksShopping

tomato pasta

gk6 · 2019. 10. 17.
[#pasta](#)

Standard Portion: 2

vegi,tomato,6,ea;
vegi,onion,1,ea;
vegi,carrot,100,g;
etc,salt,2,t

1.5 hr

Add Bookmark

M5B: Read-Recipe-Page (für das Rezept, das schon im Lesezeichen gespeichert ist)

CHICKEN NOODLE SOUPgk5 Logout

Public NotesMy NotesMy BookmarksShopping

banana sandwich

Remove Bookmark

gk6 · 2019. 10. 17.
#snack #sandwich

Standard Portion: 2

fruit,banana,100,g;
bread,toast bread,4,ea;
sauce,nutella,4,T

0.5 hr

M6: Write-Recipe-Page

CHICKEN NOODLE SOUPgk5 Logout

title e.g) Banana Shake

Standard Portion in number?

ingredients

semicolon separate for each ingredients
comma separate for category, ingredient name, amount, unit

e.g)
dairy,milk,1000,ml;
fruit,banana,100,g;
etc,ice cube,100,g;
etc,sugar,50,g;

memo

e.g)
prep. 6hr
1. prepare ice cube

steps
1. pill the banana
2. put every ingredients in the mixer
3. enjoy

Tag

Enter tagsAdd

PublishCancel

M7A: Shopping-Page (vor dem Generieren der Einkaufsliste)

CHICKEN NOODLE SOUPgk5 Logout

Public Notes	My Notes	My Bookmarks	Shopping
bookmarked notes: drag to shopping cart		shopping cart: drop here	
<div>banana sandwich</div> <div>banana shake</div> <div>chicken noodle soup</div>			
<div>hi hi</div>		<div>Calculate</div>	

M7B: Shopping-Page (nach dem Generieren der Einkaufsliste)

CHICKEN NOODLE SOUPgk5 Logout

Public Notes	My Notes	My Bookmarks	Shopping
bookmarked notes: drag to shopping cart		shopping cart: drop here	
		<div>banana sandwich</div> <div>banana shake</div> <div>chicken noodle soup</div>	
<div>bread toast bread 8.00 ea dairy milk 1000.00 ml etc ice cube 100.00 g etc salt 1.00 t etc sugar 50.00 g fruit banana 300.00 g meat chicken 600.00 g sauce nutella 8.00 T vegi carrot 200.00 g</div>		<div>Calculate</div>	

9. Literaturverzeichnis

- [1] de.wikipedia.org/wiki/Node.js
- [2] Kim, Minjun (2019): The Art of React, 2. Auflage, Seoul: GILBUT, Kapitel 21
- [3] de.wikipedia.org/wiki/MongoDB
- [4] db-engines.com/en/ranking
- [5] de.wikipedia.org/wiki/React
- [6] velog.io/@velopert/react-component-styling
- [7] Kim, Minjun (2019): The Art of React, 2. Auflage, Seoul: GILBUT, Kapitel 23
- [8] getpostman.com
- [9] reactjs.org/docs/react-dom.html
- [10] github.com/redux-saga/redux-saga

10. Eigenständigkeitserklärung

Ich bestätige hiermit, dass ich die vorstehende Projektarbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und sowohl wörtliche, als auch sinngemäss verwendete Textteile, Grafiken oder Bilder kenntlich gemacht habe. Diese Arbeit ist in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt worden.

Bülach, 23. Oktober

Gukhwa Jeon

