

**Bachelorarbeit (Informatik)**

Foodsharing App mit Nährwert­berechnung

|  |  |  |
| --- | --- | --- |
| **Autoren** |  | Betim Kabashi  Julien Wenger |
| **Hauptbetreuung** |  | Beat Seeliger |
| **Datum** |  | 10.06.2022 |

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Erklärung betreffend das selbstständige Verfassen einer   
Bachelorarbeit an der School of Engineering**

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

**Ort, Datum:** **Name Studierende:**

Zürcher Fachhochschule

**Zusammenfassung**

Hier kommt die Zusammenfassung.

**Abstract**

Hier kommt das Abstract.

**Vorwort**

Hier kommt das Vorwort.

**Inhaltsverzeichnis**

[1. Einleitung 8](#_Toc102556208)

[1.1. Ausgangslage 8](#_Toc102556209)

[1.2. Zielsetzung 8](#_Toc102556210)

[1.3. Aufgabenstellung 8](#_Toc102556211)

[1.3.1. App Funktionen 8](#_Toc102556212)

[1.3.2. Nährwertberechnung 8](#_Toc102556213)

[1.4. Anforderungen 8](#_Toc102556214)

[2. Theoretische Grundlagen 9](#_Toc102556215)

[2.1. Aspekte der Nährwerte 9](#_Toc102556216)

[2.1.1. Kalorien (Energie/Brennwert) 9](#_Toc102556217)

[2.1.2. Fette 10](#_Toc102556218)

[2.1.3. Kohlenhydrate 10](#_Toc102556219)

[2.1.4. Proteine (Eiweiss) 11](#_Toc102556220)

[2.2. Mathematische Grundlage für Kalorienberechnung 11](#_Toc102556221)

[2.2.1. Durchschnittsberechnung anhand von n-Rezepten 12](#_Toc102556222)

[3. Vorgehen / Methoden 13](#_Toc102556223)

[3.1. Datenanalyse und Mahlzeitberechnung 13](#_Toc102556224)

[3.1.1. Rahmenbedingungen 13](#_Toc102556225)

[3.1.2. Datenimport und manuelles Merging 13](#_Toc102556226)

[3.1.3. Berechnung der Kilokalorien (kcal) anhand des Basisdatensatzes 14](#_Toc102556227)

[3.1.4. Genauigkeit der Mahlzeitschätzung 14](#_Toc102556228)

[3.2. Anforderungsanalyse des Prototyps 14](#_Toc102556229)

[3.2.1. Rahmenbedingungen 15](#_Toc102556230)

[3.2.2. Funktionale Anforderungen 15](#_Toc102556231)

[3.2.3. Nicht-funktionale Anforderungen 15](#_Toc102556232)

[3.2.4. User Stories 15](#_Toc102556233)

[3.3. Technische Evaluation Backend 18](#_Toc102556234)

[3.3.1. Einschlusskriterien 18](#_Toc102556235)

[3.3.2. Ansatz 1: Backend-as-a-Service (BaaS) 18](#_Toc102556236)

[3.3.3. Ansatz 2: Platform-as-a-Service (PaaS) 18](#_Toc102556237)

[3.3.4. Entscheidungsmatrix 19](#_Toc102556238)

[3.3.5. Ergebnis Entscheidungsmatrix 20](#_Toc102556239)

[4. Resultate 21](#_Toc102556240)

[1. Diskussion und Ausblick 22](#_Toc102556241)

[2. Verzeichnisse 23](#_Toc102556242)

[2.1. Literaturverzeichnis 23](#_Toc102556243)

[2.2. Glossar 23](#_Toc102556244)

[2.3. Abbildungsverzeichnis 23](#_Toc102556245)

[2.4. Tabellenverzeichnis 23](#_Toc102556246)

[3. Anhang 24](#_Toc102556247)

[3.1. Projektmanagement 24](#_Toc102556248)

# Einleitung

## Ausgangslage

Ausgangslage

## Zielsetzung

Zielsetzung

## Aufgabenstellung

Der Prototyp einer App zum Foodsharing wird implementiert. Neben den zu erwartenden Funktionen wird ein spezieller Fokus auf die Berechnung der Nährwerte von Gerichten gelegt. Hierfür werden Daten gesammelt und mittels geeigneter Modelle ausgewertet. Die Berechnung geschieht nicht wie bei vergleichbaren Apps pro Zutat, sondern anhand der Gerichte.

### App Funktionen

* Rollen: Köche und Konsumenten
* Köche erfassen Gerichte und Zutaten
* Workflow zur Suche und Abholung
* Suche nach Gerichten
* Bewertungsmechanismus
* Evaluation einer geeigneten SW Architektur

### Nährwertberechnung

* Einarbeitung in die theoretischen Grundlagen der Nähwertberechnung
* Einbindung oder Erstellung einer Nährwertdatenbank
* Einbindung oder Erstellung einer Rezeptdatenbank
* Konzept zur Auswertung von Gerichten
* Evaluation einer geeigneten SW Architektur (spezielles Augenmerk auf Wiederverwendbarkeit auch ausserhalb der App)
* Implementierung Auswertungs-Algorithmus
* Geeigneter Mechanismus zur Validierung der Resultate

## Anforderungen

Anforderungen

# Theoretische Grundlagen

In diesem Kapitel werden die grundlegenden Informationen rund um das Thema Nährwertberechnung erläutert.

## Aspekte der Nährwerte

Um eine Nährwertmittelberechnung durchzuführen, muss zuerst eine grundlegende Basis dafür geschaffen und verstanden werden. Für in dieser Arbeit relevanten Informationen werden erst einmal die wichtigsten Nahrungsbestandteile erläutert. Es wird auf die sogenannten „Die Grossen 4“ eingegangen [1]. Unter den Grossen 4 zählen die Energie/ der Brennwert, Fett, Kohlenhydrate und Eiweiss. Dabei handelt es sich um sogenannte Makronährstoffe. Sie bilden den grössten Teil der Nahrungsaufnahme und bilden für die Nährmittelberechnung später die Basis.



Abbildung 1: Nährwertangaben [2]

Unter Abbildung 1 ist ein Beispiel ersichtlich, wie für ein Lebensmittel pro 100 g die Nährwerte abgebildet werden. Neben den gängigen Nährwerten wie Energie, Fett, Kohlenhydrate und Eiweiss werden weitere, wichtige Angaben aufgelistet, die seit der EU-Verordnung 1169/2011 [1] obligatorisch sind.

### Kalorien (Energie/Brennwert)

Kalorien werden als Maß für die Energiemenge verstanden. Die meisten Produkte werden jedoch nicht mehr mit der veralteten Kalorie beschrieben, sondern mit Joule, wobei 1 Kalorie 4,185 Joule entspricht. Verbreiteter ist dennoch die Angabe in Kilokalorien (kcal), was 1000 cal entspricht, bzw. Kilojoule. So kann in weiterer Folge auf den Energiegehalt der drei wichtigsten Hauptnährstoffkomponenten geschlossen werden, die der Körper in Energie umwandeln kann:

* Kohlenhydrate enthalten pro g 17,6 kJ bzw. **4,2 kcal**
* Proteine enthalten pro g 17,2 kJ bzw. **4,1 kcal**
* Fette enthalten pro g 38 kJ bzw. **9,3 kcal**

Der Energieverbrauch und die Energiezufuhr bestimmen unser Körpergewicht. Wird mehr Energie zugeführt als verbraucht, speichert der Körper diesen ‘Überfluss‘, das Körpergewicht steigt. Liegt dagegen die Energieaufnahme unter dem Energieverbrauch, sinkt das Körpergewicht, da der Organismus die fehlenden Kalorien aus seinen Energiespeichern (z.B. Fettgewebe) deckt.“

### Fette

Fette stellen die energiereichsten Nährstoffe und bedeutende Energiespeicher dar. Fette sind neben Glycerol aus Fettsäuren aufgebaut. Man unterteilt sie in gesättigte (vorwiegend tierische Fette) und ungesättigte FS (vorwiegend pflanzliche Fette). Des Weiteren ist der Nährstoff eine wichtige Komponente, um eine vollständige Resorption von fettlöslichen Vitaminen (A, D, E, K) zu garantieren.

Einfluss von Fetten auf den menschlichen Organismus: Die Aufnahme von gesättigten Fettsäuren ist für den menschlichen Organismus überflüssig und kann für hohes Cholesterin verantwortlich sein, was in weiterer Folge zu Problemen für Herz, Gefäße und Stoffwechsel führen kann. Ungesättigte Fettsäuren haben positive Auswirkungen und sind äußerst bedeutend für den Hormonhaushalt und die Vitaminaufnahme [3].

### Kohlenhydrate

Kohlenhydrate bezeichnet man auch als Saccharide, was aus dem Griechischen Zucker heisst. Kohlenhydrate setzt sich aus den chemischen Elementen Kohlenstoff, Wasserstoff und Sauerstoff zusammen und werden anhand der Anzahl der Baustoffe in verschiedene Gruppen eingeteilt. Diese Gruppen werden in Einfach-, Zweifach-, Mehrfach- und Vielfachzucker unterschieden. Von Mehrfachzucker spricht man bei einer Anzahl von drei bis neun Zucker-Baustoffe. Beim Vielfachzucker beträgt die Anzahl mindestens zehn Zucker-Baustoffen.

Mit Hilfe der Sonnenenergie wird der Zucker bei den Pflanzen durch Photosynthese gebildet. Die gebildeten Zuckermolekülen werden dann in Stärke umgewandelt. Da Menschen als auch Tiere ihre Energie vorwiegend in Form von Fett speichern, enthalten tierische Lebensmittel weniger Kohlenhydrate als pflanzliche Lebensmittel.

Kohlenhydrate gelten als zentrale Energielieferanten und werden nebst Fett auch als Energiespeicher genutzt. Wie bereits erwähnt, werden Kohlenhydrate in verschiedene Gruppen eingeteilt. Einfach- sowie auch Zweifachzucker begünstigen einen raschen Anstieg des Blutzuckerspiegels. Das bedeutet, dass dem Körper eine grosse Menge Energie für eine kurze Zeit zur Verfügung steht. Allerdings bewirken diese Zucker auch, dass die Energie schnell wieder verbraucht wird. Langkettige und komplexe Kohlenhydrate wie Mehrfach- und Vielfachzucker liefern hingegen kontinuierlich für einen längeren Zeitraum Energie. Das führt dazu, dass der Blutzuckerspiegel langsamer ansteigt und dem Körper längerfristig genügend Energie zur Verfügung steht [3].

### Proteine (Eiweiss)

Eiweisse versorgen den Körper mit wichtigen Aminosäuren und Stickstoff. Dabei wird zwischen 20 verschiedene Aminosäuren unterschieden, die für die körpereigene Proteinsynthese zuständig sind. Die Aminosäure ist der kleinste Baustoff, die für die Zusammensetzung der Proteine zuständig ist.

Bei diesen 20 Aminosäuren wird zwischen entbehrlichen und unentbehrlichen unterschieden. Bei den entbehrlichen Aminosäuren gibt es deren elf und bei den unentbehrlichen neun. Der menschliche Körper kann die unentbehrlichen Aminosäuren nicht selbständig produzieren. Aminosäuren können im menschlichen Körper nicht zwischen­gespeichert werden, um zu einem späteren Zeitpunkt Proteine aufzubauen. Daher ist eine tägliche Zufuhr dieser Aminosäuren nötig, um einem Eiweissmangel zu verhindern.

Die täglich empfohlene Eiweissaufnahme eines gesunden Erwachsenen beträgt in der Regel 0.8 bis 1.0 g/kg Körpergewicht.

Nebst der Qualität spielt auch die Quantität des Eiweisses eine wichtige Rolle. Dabei spricht man von der biologischen Wertigkeit. Die Proteinsynthese wird begrenzt, wenn eine unentbehrliche Aminosäure im Verhältnis zum benötigten Bedarf in einer zu geringen Menge aufgenommen wird. Die Aminosäurenzusammensetzung des Menschen ähnelt sich mehr dem tierischen als dem pflanzlichen Eiweiss. Daher weist das tierische Eiweiss in der Qualität eine höhere biologische Wertigkeit auf als das pflanzliche Eiweiss. Da aber das tierische Eiweiss meist in Verbindung mit anderen eher ungesunden Nährstoffe auftaucht, ist es ratsam auch pflanzliche Eiweisse aufzunehmen. Am besten ist eine Kombination aus beidem, um eine relativ hohe biologische Wertigkeit aufzuweisen, aber auch auf einem gesünderen Niveau zu sein [3].

## Mathematische Grundlage für Kalorienberechnung

Wie in Kapitel 2.1.1 angedeutet, kann anhand der Mengenangabe von Fett, Kohlenhydrate und Proteine die Gesamtkalorien einer Mahlzeit berechnet werden. Dabei multipliziert man die Menge in Gramm \* Kcal pro Gramm für jedes der drei Nährwerte und summiert sie auf. Beispiel:

Die Voraussetzung, um diese Berechnung genau durchzuführen, ist die Mengenangabe jeder einzelnen Zutat in einem Gericht zu kennen. Dies führt zum Problem, dass ein Koch während der Zubereitung jede Zutat abwiegen oder sich strikt an die Mengen im Rezept halten muss. Um diesem Problem entgegenzuwirken, wird im nächsten Kapitel ein Ansatz vorgestellt, bei denen die Berechnung der Kalorienanzahl auf Schätz- und Mittelwerte basieren, anstatt auf genaue Mengenangaben der Zutaten.

### Durchschnittsberechnung anhand von n-Rezepten

Man nehme n verschiedene Rezepte vom gleichen Gericht und macht sie vergleichbar, indem die Portionsgrössen für jedes Rezept auf eine gleiche Basis angepasst werden. Im zweiten Schritt schaut man auf die Menge der einzelnen Zutaten und berechnet den Durschnitt der Menge über alle n Rezepte. Somit wird ein weiteres Rezept erzeugt, welches Durchschnittsmengenangaben enthält, von dem dann die Kalorien berechnet werden. In Kapitel 3 wird die Berechnung, die Genauigkeit der Rechnung und die Realisierung dieser Idee im Rahmen vom Prototyp veranschaulicht.

# Vorgehen / Methoden

Das Vorgehen wird in zwei Teilbereiche aufgeteilt. Der eine Teilbereich beinhaltet das Vorgehen, die Konstruktion und Realisierung des Prototyps einer Smartphone-Applikation. Der andere Teilbereich befasst sich mit der Nährwertberechnung im Detail.

## Datenanalyse und Mahlzeitberechnung

Wie in den theoretischen Grundlagen angedeutet, wird in diesem Abschnitt die konkrete Realisierung der Mahlzeitberechnung beschrieben.

### Rahmenbedingungen

Da es keine vorgefertigten Datensätze gibt und ein ETL Prozess (Extraktion, Transformation, Laden) nicht Teil der Arbeit ist, wird der initiale Datenimport der Gerichte und das Zusammenführen der Rezepte manuell durchgeführt. Aus diesem Grund enthält der Prototyp initial nur 30 vordefinierte Rezepte.

### Datenimport und manuelles Merging

Der erste Schritt vor dem Datenimport ist die Suche nach Gerichten. Mit Hilfe des Webs werden für jedes Gericht drei Rezepte aus verschiedenen Quellen gesucht. Im zweiten Schritt werden die Portionsgrössen über die drei Rezepte vereinheitlicht und Grössen wie EL(Esslöffel), TL(Teelöffel) u.ä. in Gramm umgewandelt, sodass jedes Rezept die gleiche Portionsgrösse in Gramm hat. Dabei werden Gewürze oder Kräuter in der Berechnung ignoriert, da deren Kaloriengehalt nur wenig bis gar keine Auswirkung auf das Resultat haben. Zusätzlich werden Zutaten für ein Gericht als optional markiert, wenn diese nicht in der Mehrheit der Rezepte vorkommen. Optional bedeutet, dass diese Zutaten nicht in die Basisberechnung miteinfliessen, sie können aber im Prototyp angewählt und dann zu den Gesamtkalorien dazugezählt werden. Als dritten und letzten Schritt werden über alle Rezepte (im Fall des Prototyps sind es drei pro Gericht) der Mittelwert berechnet und so ein Basisdatensatz generiert, der dann verwendet werden kann, um die Kilokalorien für gekochte Gerichte zu berechnen. Im nächsten Kapitel wird auf die Genauigkeit der Schätzung gegenüber der effektiven Kalorienanzahl eingegangen.

In der folgenden Grafik ist der beschriebene Prozess an einem Beispiel für «Ghackets mit Hörnli» veranschaulicht.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 2: Beispiel Rezept Merging

### Berechnung der Kilokalorien (kcal) anhand des Basisdatensatzes

### Genauigkeit der Mahlzeitschätzung

## Anforderungsanalyse des Prototyps

Die Anforderungen des Prototyps wurden in funktionale und nicht-funktionale Anforderungen aufgeteilt. Funktionale Anforderungen legen fest, was das Programm am Ende tun soll, während nicht-funktionale Anforderungen beschreiben, wie gut der Prototyp performt.

### Rahmenbedingungen

|  |  |
| --- | --- |
| Entwickler | 2 |
| Zielgruppe | Privatpersonen, die Gerichte zur Verfügung stellen möchten oder aber auch nach verfügbaren Gerichten suchen. |
| Endgeräte | Alle gängigen Smartphones ab 2018 |
| Software | Für die Umsetzung des Frontends wird React Native eingesetzt. |

Tabelle 1: Rahmenbedingungen (Prototyp)

### Funktionale Anforderungen

|  |  |
| --- | --- |
| **ID** | **Anforderungen** |
| F-1 | Koch und Konsument können sich registrieren. |
| F-2 | Koch und Konsument haben ein eigenes Benutzerkonto. |
| F-3 | Koch kann Gerichte erfassen. |
| F-4 | Konsument kann nach Gerichten suchen. |
| F-5 | Konsument hat Einsicht in die veröffentlichten Gerichte und dessen Nährwerte. |
| F-6 | Konsument kann ein Gericht auswählen und die Menge des Gerichts angeben. |
| F-7 | Koch und Konsument können vereinbaren, wann sie die Übergabe des Gerichts durchführen wollen. |
| F-8 | Konsument kann den jeweiligen Koch bewerten. |

Tabelle 2: Funktionale Anforderungen

### Nicht-funktionale Anforderungen

|  |  |
| --- | --- |
| **ID** | **Anforderungen** |
| NF-1 | Der Prototyp soll zuverlässig und stabil auf allen Android-Geräten laufen mit mindestens der Android-Version 7.0. |
| NF-2 | Der Prototyp soll eine ausgeglichene Ressourcen­auslastung aufweisen. |
| NF-3 | Der Prototyp soll einfach und intuitiv auf dem Smartphone installiert werden können. |
| NF-4 | Der Prototyp soll vom Benutzer einfach und intuitiv bedienbar sein. |

Tabelle 3: Nicht-funktionale Anforderungen

### User Stories

|  |  |
| --- | --- |
| Titel | Benutzerkonto |
| User Story ID | US-1 |
| Funktionale Anforderung ID | F-1, F-2, F-3, F-4 |
| Beschreibung | Der Koch und der Konsument können sich registrieren und somit ein eigenes Benutzerkonto erstellen. |
| Akzeptanzkriterien | * Bei der Registrierung wird zwischen Koch und Konsument unterschieden. * Ein Koch kann Zutaten und Gerichte auf seinem Benutzerkonto veröffentlichen. * Ein Konsument kann nach Gerichten suchen und diese auswählen. |

Tabelle 4: User Story - Benutzerkonto

|  |  |
| --- | --- |
| Titel | Zutat und Gericht erfassen |
| User Story ID | US-2 |
| Funktionale Anforderungs-ID | F-3 |
| Beschreibung | Der Koch kann beliebig viele Zutaten und diverse Gerichte erfassen und veröffentlichen. |
| Akzeptanzkriterien | * Anhand einer Zutaten-Datenbank kann der Koch die jeweilige Zutat auswählen. * Anhand einer Gerichten-Datenbank kann der Koch das jeweilige Gericht auswählen. * Die Zutaten und Gerichte kann der Koch schliesslich veröffentlichen, sodass Konsumenten diese finden können. |

Tabelle 5: Zutat und Gericht erfassen

|  |  |
| --- | --- |
| Titel | Gericht suchen und auswählen |
| User Story ID | US-3 |
| Funktionale Anforderungs-ID | F-4, F-5, F-6 |
| Beschreibung | Der Konsument kann nach Gerichten suchen und erhält Einsicht in das veröffentlichte Gericht und dessen Nährwertangaben. Schliesslich kann ein Gericht ausgewählt werden. |
| Akzeptanzkriterien | * Der Konsument kann nach Gerichten suchen. * Der Konsument hat Einsicht in das jeweilig veröffentlichte Gericht und dessen Nährwert­angaben. * Schliesslich kann der Konsument ein Gericht auswählen und dessen Menge angeben. |

Tabelle 6: Gericht suchen und auswählen

|  |  |
| --- | --- |
| Titel | Vereinbarung Abholung |
| User Story ID | US-4 |
| Funktionale Anforderungs-ID | F-7 |
| Beschreibung | Der Konsument und der Koch können bei erfolgter Auswahl des Gerichtes vereinbaren, wann das Gericht abgeholt werden soll. |
| Akzeptanzkriterien | * Der Konsument und der Koch können sich vereinbaren, wann das Gericht abgeholt werden soll. |

Tabelle 7: Vereinbarung Abholung

|  |  |
| --- | --- |
| Titel | Koch bewerten |
| User Story ID | US-5 |
| Funktionale Anforderungs-ID | F-8 |
| Beschreibung | Der Konsument kann nach erfolgter Abholung den Koch optional bewerten. |
| Akzeptanzkriterien | * Die Bewertung kann nur nach erfolgter Abholung erfolgen. |

Tabelle 8: Koch bewerten

## Technische Evaluation Backend

In diesem Kapitel werden für die Realisierung des Backends zwei moderne Ansätze verglichen und evaluiert. Das Resultat der Evaluation sollte ein klares Bild verschaffen, welches Framework oder welche Technologie für das Projekt bzw. den Prototyp am geeignetsten ist. Aus diesem Grund werden für beide Ansätze zwei bis drei konkrete Technologien und/oder Frameworks in die Evaluation mit einbezogen. Das wichtigste ist hierbei, ob und wie gut die ausgewählten Lösungen unsere nicht-funktionalen Anforderungen erfüllen. Wir beschränken uns bei der folgenden Evaluation nur auf Cloud-basierte Lösungen, da es sich bei der App um einen Prototyp handelt und bei einer On-Premise Lösung der initiale Aufwand zu gross wäre.

### Einschlusskriterien

In der Evaluation werden nur Technologien evaluiert, die folgende Kriterien erfüllen:

* Backend-App auf NodeJS (Gilt nur für Ansatz 2)
* Volltextsuche muss möglich sein
* DBMS Modell entweder Document Store oder Relationale DB
* Einfache Verbindung von DBMS und NodeJS (Gilt nur für Ansatz 2)

### Ansatz 1: Backend-as-a-Service (BaaS)

Als Backend-as-a-Service werden Produkte bezeichnet, bei dem der Anbieter vollständige Backend-Lösungen (inklusive Datenbank) zur Verfügung stellt. Der Konsument kümmert sich im Besten Fall nur noch um die Datenmodelle oder die Anbindung ans UI. Aufgrund der Tatsache, dass sich die Anbieter, auf die nicht-funktionalen Anforderungen bezogen, kaum unterscheiden, wird nur der bekannteste Anbieter in die Evaluation miteinbezogen.

**Kandidaten**

**Google Firebase**

Firebase ist eine Entwicklungs- und Hostingplatform spezialisiert auf Mobile- und Webanwendungen. Es verfügt über eine Realtime Dokumentdatenbank und hat zusätzlich noch weitere nützliche Backendfeatures, wie zum Beispiel User Authentifizierung sowie Monitoring- und Analyse Tools. Ein Nachteil der Firebase Datenbank ist die fehlende Volltextsuchfunktionalität, diese muss mit externen Suchanbieter realisiert werden. Firebase gehört zu den bekanntesten Backend-as-a-Service Anbieter.

### Ansatz 2: Platform-as-a-Service (PaaS)

Als Product-as-a-Service werden Produkte bezeichnet, die es Konsumenten ermöglichen, mit wenig Aufwand, ihre Applikation (inklusive Datenbank) auf konfigurierbare und wartbare Cloud Umgebungen zu verwalten. Dies hat den Vorteil, dass sich der Konsument nur auf die Entwicklung der Applikation fokussieren und infrastrukturbedingte Fragestellungen dem Anbieter überlassen kann. In diesem Segment gibt es sehr viele Anbieter. Im Wesentlichen unterscheiden sie sich in der Kompatibilität von Frameworks/Technologien und in der Anzahl von zusätzlichen Middleware Tools für die Wartung oder Analyse der Applikation oder der Daten. Aus diesem Grund ist es in diesem Ansatz primär wichtiger, den Fokus auf die Auswahl der Frameworks bzw. Technologien zu legen anstatt der Auswahl des PaaS Anbieters. Hierbei kommen nur Technologien in Frage, die mit dem Betreuer besprochen wurden und die Einschlusskriterien erfüllen.

**Kandidaten**

**ExpressJS + PostgreSQL auf Heroku**

ExpressJS ist ein NodeJS Framework für die Entwicklung von Backend Schnittstellen. Heroku gehört zur PaaS Kategorie, welches unter Anderem NodeJS unterstützt und es dem Entwickler mit geringem Aufwand ermöglicht, seine Applikation auf der Cloud zu bauen, hosten und zu verwalten. PostgreSQL ist im Gegensatz zu Firebase-DB oder MongoDB eine relationale Datenbank, die aber Volltextsuche, durch SQL-Funktionen, anbietet. CloudSQL ist ein PaaS von Google, die relationale Datenbanksysteme, wie MySQL, PostgreSQL und SQL-Server, vollständig verwalten kann.

**Apollo Server + PostgreSQL Heroku**

Einzigartig an dieser Lösung sind die Backendschnittstellen. Anders als bei ExpressJS oder anderen üblichen Frameworks werden hier mittels Apollo Server Framework GraphQL Schnittstellen zur Verfügung gestellt. GraphQLs grosse Stärke gegenüber REST ist, dass Schnittstellen anpassbar sind. Das heisst, während REST für einen Endpunkt immer die gleiche Struktur an Daten zurückliefert, kann man mittels GraphQL während der Abfrage definieren, welche Felder man wirklich braucht. Somit kann eine GraphQL Schnittstelle für mehrere Fälle benutzt werden.

**NodeJS + Elasticsearch auf Heroku + MongoDBAtlas**

Dieses Setup hat, verglichen mit den anderen Kandidaten, mit Elasticsearch Cloud die leistungsstärkste Volltextsuche. Zusätzlich bietet es Analyse und Visualisierungstools an. Es ist bekannt, dass sich Elasticsearch nicht als primäre Datenbank für die Speicherung von User- oder sonstigen statischen Daten eignet, deshalb braucht dieser Ansatz ein zusätzliches Speichermedium, nämlich PostgreSQL.

### Entscheidungsmatrix

In der folgenden Tabelle ist die Auswertung der Kandidaten zu sehen. Die Kriterien und Gewichtung in der Entscheidungsmatrix haben sich aus der Anforderungsanalyse herauskristallisiert und beschreiben folgendes:

* Setup: Wie einfach und aufwendig ist das initiale Setup des Backends? Wie schnell kann man mit der eigentlichen Entwicklung starten?
* Out-of-the-Box Volltextsuche: Enthält die Lösung eine Volltextsuchfunktionalität oder müssen zusätzliche Erweiterungen aufgesetzt und konfiguriert werden?
* Verwendet bekannte Protokolle und Sprache: Ist die Technologie hinter der Lösung bekannt? Werden Sprachen oder Protokolle verwendet, die neu sind und noch gelernt werden müssen?
* Einfache DB-Anbindung ans NodeJS-Backend: Gibt es standardisierte Guides für die Anbindung? Unterstützt das Framework die gewählte DB?
* Tools für Datenanalyse: Welche Analytik-Tools bietet die Lösung an? Können diese ohne grossen Aufwand genutzt werden oder braucht es zusätzliche Konfigurationen/ Installationen
* Wartbarkeit: Angebot an: Deploymentmöglichkeiten, Automatisierungen, Aufsetzen von Test-/Produktionsumgebungen.
* Vorhandenes Know-How: Vorhandene Erfahrungen mit den Technologien.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 3: Entscheidungsmatrix Backend

### Ergebnis Entscheidungsmatrix

Wie in der Entscheidungsmatrix zu sehen, ist Firebase mit 29.9 Punkten die geeignetste Lösung für den Prototyp. Grundsätzlich kann man sagen, dass jede der Kandidaten genutzt werden kann, um den Prototyp zu realisieren, jedoch hat Firebase den grossen Vorteil, dass die Datenbank integriert ist und keine zusätzlichen Installationen bzw. Anbindungen notwendig sind. Die Evaluation hat aufgezeigt, dass es keine «perfekte» Backendlösung gibt. Vielmehr stellt sich die Frage, welche Strategie verfolgt werden soll. Bei Backend-as-a-Service Ansätzen kann die Entwicklung abhängig vom Lieferanten (in diesem Fall Google) sein. Dadurch können Engpässe entstehen, die die Realisation eines Produktes verzögern können. Auf der anderen Seite haben Eigenbaulösungen den grossen Nachteil, dass der initiale Aufwand grösser ist, jedoch ist die Abhängigkeit zu Lieferanten nicht gegeben und gewinnt so an Flexibilität.

## Realisierung Backend mit Firebase

In diesem Kapitel wird die Realisierung des Backends mittels Firebase beschrieben. Dazu zählen: Aufsetzen des Projekts, Erstellung der APIs und die Definition sowie Speicherung der Datenstrukturen.

### Cloud Firestore vs. Realtime Database

Google Firebase bietet zwei Arten von Datenbanken an, nämlich «Cloud Firestore» und «Realtime Database». «Cloud Firestore» ist die neuere Variante, die auf die "Realtime Database" aufbaut und zusätzliche Funktionen anbietet. Der essenzielle Unterschied ist die Speicherung. Während «Realtime Database» die Daten nur in reinem JSON Format abspeichert kann, bietet «Cloud Firestore» die Möglichkeit, die Daten hierarchisch zu strukturieren. Da man Gerichte und deren Zutaten sowie Nährwerte sinnvoll hierarchisch aufbauen kann, wird für dieses Projekt «Cloud Firestore» verwendet.

### Erstellung der Datenstrukturen mittels Collections und Sub-Collections

Der Aufbau und die Anordnung der einzelnen Dokumente bzw. der Daten wird hier als Datenstruktur bezeichnet. Dabei gilt die Regel, dass das Backend nur die Daten liefern soll, die für die Darstellung oder Funktion im GUI notwendig ist. Man spricht auch von einer schlanken API, bei der die Datenmenge, die über die REST API geschickt wird, so klein wie möglich gehalten werden soll. Um dieses Ziel zu erreichen, bietet Cloud Firestore Sub-Collections an. Hiermit können die Daten hierarchisch angeordnet werden, mit der Möglichkeit nur einzelne Elemente in der Hierarchie anzusprechen, ohne die darunterliegenden Elemente einzubeziehen. Am einfachsten lässt sich dies am folgenden Beispiel erklären:

Die Strukur eines Rezeptes bzw. Gericht kann folgendermassen aussehen:

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 4: Gliederung Rezept Daten

Anhand der Abbildung oben kann eine Hierarchie erkannt werden. Ein Rezept (recipe) hat mehrere Zutaten (ingredients). Eine Zutat hat die Nährwerte: Kohlenhydrate (carbs), Protein (protein) und Fett (fat). Bei einer üblichen Datenspeicherung ohne Sub-Collections gibt es keine Möglichkeit nur Teilobjekte über die API zu schicken, sondern es muss das ganze Objekt bzw. Dokument Rezept (recipe) zurückgeliefert werden. Wenn aber die Zutat (ingredients) als Sub-Collection definiert wird, ist es möglich nur die Zutaten (ingredients) oder nur das Rezept (recipe) ohne Zutaten (ingredients) über die API zu transferieren. Ein weiteres Beispiel wäre die Bewertungen eines Koches. Ein Koch ist in der Datenbank als User Dokument abgelegt. Ohne Sub-Collection wären die Bewertungen des Kochs Teil des User Dokumentes und werden somit bei jeder Anfrage mitgeliefert, auch wenn die Bewertungen im GUI für gewisse Use Cases gar nicht gebraucht werden. Mit Hilfe von Sub-Collections kann die mobile Datennutzung auf Endgeräten optimiert werden.

# Resultate

Resultate

# Diskussion und Ausblick

Diskussion und Ausblick

# Verzeichnisse

## Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | S. Täuber, «Produktanalysen und Nährwert,» [Online]. Available: https://www.ufag-laboratorien.ch/lebensmittel-analytik/produktanalysen-und-naehrwert/. [Zugriff am 2 April 2022]. |
| [2] | M. Blättler, «Nährwerttabelle: Darauf sollte man beim Kauf von Lebensmitteln achten,» [Online]. Available: https://gymperformance.ch/naehrwerttabelle-darauf-sollte-man-beim-kauf-von-lebensmitteln-achten/. [Zugriff am 2 April 2022]. |
| [3] | D. E.-. &. -informationsnetz, «Ernährungsinformationen - Nahrungsbestandteile,» [Online]. Available: https://www.ernaehrung.de/tipps/allgemeine\_infos/ernaehr11.php. [Zugriff am 3 April 2022]. |

## Glossar

Glossar

## Abbildungsverzeichnis

[Abbildung 1: Nährwertangaben [2] 8](#_Toc102311289)

[Abbildung 2: Beispiel Rezept Merging 13](#_Toc102311290)

[Abbildung 3: Entscheidungsmatrix Backend 19](#_Toc102311291)

## Tabellenverzeichnis

[Tabelle 1: Rahmenbedingungen (Prototyp) 14](#_Toc102311281)

[Tabelle 2: Funktionale Anforderungen 14](#_Toc102311282)

[Tabelle 3: Nicht-funktionale Anforderungen 14](#_Toc102311283)

[Tabelle 4: User Story - Benutzerkonto 15](#_Toc102311284)

[Tabelle 5: Zutat und Gericht erfassen 15](#_Toc102311285)

[Tabelle 6: Gericht suchen und auswählen 16](#_Toc102311286)

[Tabelle 7: Vereinbarung Abholung 16](#_Toc102311287)

[Tabelle 8: Koch bewerten 16](#_Toc102311288)

# Anhang

Anhang

## Projektmanagement

Projektmanagement