HTL - Perg
Höhere Abteilung für Informatik

Diplomarbeit

Smarte Lebensmittelverwaltungsapp – Smood

Projektteam: Julia Bodingbauer
Jana Dannhofer
Moritz Lechthaler
Fabian Kopf
Projektbetreuer: Prof. Ing. D. Raffetseder

In Zusammenarbeit mit den Absolventenverband der HTL-Perg
Betreuer: Lukas Huber

Bearbeitungszeitraum: 01.05.2022 – 05.04.2023

HTL Perg | Machlandstr. 48 | A-4320 Perg | Tel.: +43 (0) 7262 53926

1 Eidesstattliche Erklärung

Die unterfertigten Kandidaten / Kandidatinnen haben gemäß § 34 (3) SchUG in Verbindung mit § 22 (1) Zi. 3 lit. b der Verordnung über die abschließenden Prüfungen in den berufsbildenden mittleren und höheren Schulen, BGBl. II Nr. 70 vom 24.02.2000 (Prüfungsordnung BMHS), die Ausarbeitung einer Diplomarbeit mit der umseitig angeführten Aufgabenstellung gewählt.

Die Kandidaten / Kandidatinnen nehmen zur Kenntnis, dass die Diplomarbeit in eigenständiger Weise und außerhalb des Unterrichtes zu bearbeiten und anzufertigen ist, wobei Ergebnisse des Unterrichtes mit einbezogen werden können.

Die Abgabe der Diplomarbeit hat bis spätestens 05.04.2023 beim zuständigen Betreuer / der zuständigen Betreuerin zu erfolgen.

Die Kandidaten / Kandidatinnen nehmen weiters zur Kenntnis, dass gemäß § 9 (6) der Prüfungsordnung BMHS nur der Schulleiter bis spätestens Ende des vorletzten Semesters den Abbruch einer Diplomarbeit anordnen kann, wenn diese aus nicht beim Prüfungskandidaten (bei den Prüfungskandidaten) gelegenen Gründen nicht fertiggestellt werden kann.

Kandidaten /Kandidatinnen inkl. Unterschrift:

Perg, _____ Unterschrift _____

(Julia Bodingbauer)

Perg, _____ Unterschrift _____

(Jana Dannhofer)

Perg, _____ Unterschrift _____

(Moritz Lechthaler)

Perg, _____ Unterschrift _____

(Fabian Kopf)

2 Gender-Erklärung

Für die bessere Lesbarkeit werden in dieser Diplomarbeit personenbezogene Bezeichnungen, die sich zugleich auf Frauen und Männer beziehen, generell nur in der Deutschen üblichen maskulinen Form angeführt.

Dies soll jedoch keinesfalls eine Geschlechterdiskriminierung oder eine Verletzung des Gleichheitsgrundsatzes zum Ausdruck bringen.

Perg, _____ Unterschrift _____

(Julia Bodingbauer)

Perg, _____ Unterschrift _____

(Jana Dannhofer)

Perg, _____ Unterschrift _____

(Moritz Lechthaler)

Perg, _____ Unterschrift _____

(Fabian Kopf)

3 Danksagung

Wir möchten uns bei sämtlichen Personen und Organisationen bedanken, die diese Diplomarbeit möglich gemacht haben.

Besonders erwähnen möchten wir unseren Betreuungslehrer Prof. Ing. Dominik Raffetseder, MSc, der uns stets für Fragen zur Seite stand und uns unterstützte, sowie den Obmann des Absolventenverbandes der HTL Perg, Herrn Lukas Huber, der als Auftraggeber und Ansprechperson das Projekt ermöglicht hat.

Weiters bedanken wir uns bei Herrn Prof. Ing. Werner Schöller sowie Herrn Clemens Denzinger, die uns als IT-Administratoren der HTL Perg die technische Basis durch die schuleigene Serverumgebung zur Verfügung gestellt haben.

4 Impressum

Schule

HTBLA Perg für Informatik

Machlandstraße 48

4320 Perg

Schuljahr

2022/23

Klasse

5BHIF

Projektname

Smood – Smarte Lebensmittelverwaltungsapp

Projektteam

Julia Bodingbauer

Jana Dannhofer

Moritz Lechthaler

Fabian Kopf

Betreuungslehrer

Prof. Ing. Dominik Raffetseder, MSc

Auftraggeber

P@BS – EDV-Absolventen und Förderer der HTL–Perg

Machlandstraße 48

4320 Perg

Obmann und Ansprechperson: Lukas Huber

5 Inhaltsverzeichnis

6 Kurzbeschreibung

Die Diplomarbeit Smarte Lebensmittelverwaltungsapp – Smood wurde im Rahmen der standardisierten Reife- und Diplomprüfung 2023 an der HTL Perg für Informatik erstellt.

Problemstellung

Den wenigsten Leuten ist klar, welche Lebensmittel zu Hause gelagert sind. Problematisch ist dies bezogen auf das Haltbarkeitsdatum der Lebensmittel. Viele Lebensmittel werden weggeschmissen, weil diese abgelaufen sind.
Das Ziel unserer Diplomarbeit ist es, eine Übersicht über die Lebensmittel im Haushalt zu bekommen. Die App soll den Benutzer darüber informieren, welche Produkte in naher Zukunft ablaufen bzw. welche Produkte zu Hause sind, damit der Benutzer die Chance hat, die Produkte zu konsumieren bevor diese schlecht werden. Darüber hinaus möchten wir, mithilfe unserer App, auf dieses Problem aufmerksam machen, indem wir den Österreichern ein Bewusstsein für die Lebensmittelverschwendung schaffen. Außerdem soll eine Einkaufsliste dabei helfen, nur die benötigen Produkte einzukaufen.

Realisierung

Die Basis für die Implementierung von Smood ist ein Linux-Server, welcher sich im Netzwerk der HTL Perg befindet. Dieser Server hostet eine MySQL-Datenbank und eine REST-API, welche in Java implementiert wurde. Mithilfe dieser Schnittstelle können Daten für die App bereitgestellt werden. Es ist jedoch zu beachten, dass diese App ausschließlich auf Android-Handys genutzt werden kann.
Weiters wurde ein Algorithmus entwickelt, welcher auf den individuellen Benutzerdaten basiert. Auf den Informationen aufbauend wird eine Einkaufsliste automatisch generiert und dem Nutzer zur Verfügung gestellt. Der Algorithmus ist in der Lage, sich kontinuierlich an die Einkaufsgewohnheiten der Nutzer anzupassen und diese zu optimieren.
So können Nutzer die Einkäufe gezielt planen, und nur das kaufen, was sie wirklich benötigen.

Ergebnis

Das Ergebnis des Projekts ist eine benutzerfreundliche App, die den Nutzern eine umfassende Übersicht über ihre Vorräte zu Hause gibt. Die App ist einfach zu bedienen und intuitiv gestaltet, so dass auch unerfahrene Nutzer schnell mit ihr zurechtkommen. Mit Hilfe der App können die Nutzer genau sehen, welche Produkte sich noch im Kühlschrank befinden. Auf einer anderen Seite der App werden alle abgelaufenen Produkte übersichtlich aufgelistet. Dadurch können die Nutzer schnell und einfach feststellen, welche Lebensmittel abgelaufen sind.

Ein weiterer wichtiger Bestandteil der App ist der Algorithmus, der eine personalisierte Einkaufsliste mit der Menge an benötigten Produkten generiert. Der Algorithmus berücksichtigt dabei die Daten jedes Benutzers und stellt eine Liste mit den notwendigen Einkäufen bereit.

In der App können Benutzer ihre Lebensmittelbestände einfach verwalten. Mit nur wenigen Klicks können neue Produkte hinzugefügt oder verbrauchte Produkte aus der Liste entfernt werden.

Eine weitere wichtige Funktion der App ist die Benachrichtigungsfunktion. Diese erinnert die Benutzer daran, wenn Lebensmittel bald ablaufen werden. Innerhalb von drei Tagen vor dem Ablaufdatum erhält der Benutzer eine Benachrichtigung, um auf ein ablaufendes Lebensmittel aufmerksam zu machen.

7 Abstract

The diploma thesis Smarte Lebensmittelverwaltungsapp – Smood was created within the scope of the A-levels 2023 at the College of Engineering for informatics in Perg.

Problem

Very few people are aware of what food is stored in their homes. This is problematic in relation to the expiration date of the food. A lot of food is thrown away simply because it has expired. A lot of resources are wasted due to food waste.
The aim of the diploma thesis is to provide an overview into the food household. The app should inform the user which products will expire in the near future or which products are already at home, so that the user has the chance to consume the products before they go bad. Furthermore, with the help of our app, we want to draw attention to this problem by creating awareness among Austrians about food waste. In addition, a shopping list will help users to buy only the necessary products they need.

Implementation

The basis for the implementation of Smood is a Linux server, which is located in the network of the HTL Perg. This server hosts a MySQL database and a REST-API, which is implemented in Java. With the help of this interface, data can be provided for the app. However, it should be noted that this app can only be used on Android mobile phones.
Furthermore, an algorithm has been developed based on individual user data. Based on the information, a shopping list is automatically generated and made available to the user. The algorithm is capable of continuously adapting to users' shopping habits and optimizing them. In this way, users can plan purchases in a targeted manner and only buy what they really need.

Result

The outcome of the project is a user-friendly app that provides users with a comprehensive overview of their supplies at home. The app is easy to use and designed intuitively, making it easy for even inexperienced users to quickly become familiar with it. With the help of the app, users can easily see which products are still in the fridge.

On another page of the app, all expired products are clearly listed. This allows users to quickly and easily identify which groceries have expired. Another important component of the app is the algorithm that generates a personalized shopping list with the number of products needed. The algorithm takes into account the data of each user and provides a list with the necessary purchases.

In the app, users can easily manage their food stocks. With just a few clicks, new products can be added or consumed products can be removed from the list. Another important feature of the app is the notification function. Within three days before the expiration date, users receive a notification to ensure that they consume the product in question in time.

8 Stakeholder

8.1 Betreuung

8.1.1 Prof. Ing. Dominik Raffetseder, MSc

Kontakt

d.raffetseder@htl-perg.ac.at

Abbildung 1: Prof. Ing. Dominik Raffetseder, MSc (HTL-Perg Lehrerteam, 2023)

Unsere Diplomarbeit wird von Herrn Professor Raffetseder betreut. Seit 2019 unterrichtet Professor Dominik Raffetseder an der HTL in Perg. Im heurigen Schuljahr unterrichtet er die Fächer Datenbanken und Informationssysteme, Netzwerktechnik und verteilte Systeme (Mobile Computing), Programmieren und Software-Engineering (Java) und Systemplanung und Projektentwicklung.

Professor Raffetseder begleitete uns durch die dritte Klasse im Gegenstand Programmieren und Software Engineering Java. Durch seine guten Kenntnisse in Datenbanken, Mobile Computing und Projektentwicklung ist er uns immer einse große Hilfe bei der Planung und Realisierung unserer Diplomarbeit gewesen. Durch seine jahrelange Berufserfahrung stand er uns stets mit Rat und Tat zur Seite.

8.2 Lehranstalt

HTL Perg, Höhere Abteilung für Informatik

Kontakt

HTBLA Perg

Machlandstraße 48

4320 Perg

Tel. 0 72 62 / 539 26

office@htl-perg.ac.at

www.htl-perg.ac.at

Abbildung 2: HTL-Perg (htl-perg.ac.at, 2023)

Die HTL Perg ist eine Bildungseinrichtung, die ihren Schülern umfassende Kenntnisse in verschiedenen Technologiebereichen vermittelt, darunter Softwareentwicklung, Hardware, Datenbanken, Netzwerke und Projektmanagement. Durch den praxisorientierten Ansatz der HTL Perg werden die Absolventinnen und Absolventen auf Führungsrollen in unterschiedlichen Unternehmen vorbereitet. Sie sind in der Lage, Analyse-, Administration- und Softwareentwicklungsaufgaben zu übernehmen, einschließlich der Erstellung, Testung und Wartung von Software. Die Absolventen der HTL Perg sind zudem in der Lage, verschiedene Rollen in Unternehmen von der Sachbearbeitung bis hin zu Führungspositionen zu übernehmen. (vgl. htl-perg.ac.at, 2023)

Eine Allgemeine Beschreibung der HTL Perg, Höhere Abteilung für Informatik:

> Der Schwerpunkt der Ausbildung liegt im Bereich der Softwareentwicklung, wobei betriebswirtschaftliche Aspekte besondere Berücksichtigung finden.
> Gleichzeitig wird auch eine fundierte Allgemeinbildung geboten. Diese Kombination bedeutet eine umfangreiche, praxisnahe Ausbildung, die später im Beruf direkt umgesetzt werden kann, aber auch die Möglichkeit zu Universitäts- bzw. Fachhochschulstudium bietet. Das rege Interesse der Wirtschaft an dieser Fachrichtung wird durch ständige Kooperationen mit Unternehmen in Form von Projekten unterstrichen, die von Schülern im Rahmen des Gegenstandes Projektentwicklung oder als Diplomarbeit realisiert werden. (htl-perg.ac.at, 2023)

8.3 Auftraggeber

P@bs – EDV-Absolventen und Förderer der HTL – Perg

Der Verein P@BS – EDV-Absolventen und Förderer der HTL – Perg hat den Zweck der Bildung und Aufrechterhaltung einer Informationsplattform zwischen Schule, Firmen, Absolventen und sonstigen Interessenten. Der Praxisbezug der Ausbildung soll so auf dem neuesten Stand erhalten bzw. erhöht werden.

Zudem ist die Förderung der Ausbildung an der HTL Perg durch Geld- und Sachwerte an die Schule bzw. die Schüler Inhalt der Tätigkeit. Die Mittel ergeben sich auf der einen Seite durch Mitgliedsbeiträge, Erträge aus Veranstaltungen und Spenden und auf der anderen Seite durch Vorträge, Diskussionsveranstaltungen, Firmenpräsentationen, Präsentationen von Schulprojekten, Firmenbesuche und Informationsveranstaltungen zur Präsentation des Vereins.

Mit der Mitgliedschaft sollte Unternehmen die Möglichkeit gegeben werden, Projekte mit der HTL Perg in Form von Jahres- oder Maturathemen zu realisieren. Weiters steht der Verein unterstützend bei der Vermittlung von Absolventen bzw. Adressen von Absolventen zur Seite.

In der Folge sollten sich gemeinsame Präsentationen von abgeschlossenen Themen, die in Kooperation umgesetzt wurden, ergeben. (Pabs.at, 2023)

Kontakt

Machlandstraße 48

4320 Perg

P@bs@htl-perg.ac.at

Obmann Lukas Huber

Unser Ansprechpartner vom Absolventenverband ist dessen Obmann, Lukas Huber. Wir haben mit ihm den Umfang der Diplomarbeit sowie Funktionalitäten definiert und er stand uns stehts als Ansprechperson für Fragen bei Seite.

Abbildung 3: Lukas Huber (vgl. Pabs.at, 2023)

Kontakt

lukashuber@kpmg.at

9 Einleitung

9.1 Ist-Zustand/ Hintergrund

Jedes Jahr wird allein in Österreich über eine Million Tonnen an Lebensmitteln in den Müll geworfen. Dabei handelt es sich circa um ein Drittel aller in Österreich im Umlauf befindlichen Produkte. Die größten Verschwender sind hierbei die privaten Haushalte. Pro Haushalt werden jedes Jahr zwischen 250 und 800 Euro in den Müll geworfen. Das bringt viele soziale und auch ökonomische Auswirkungen mit sich. Die Million Lebensmittel, welche jedes Jahr verschwendet werden, sind um eine Million zu viel und somit eine riesige Belastung für Natur und Klima. Diese Lebensmittelverschwendung muss gestoppt werden! (vgl. WWF, 2023)

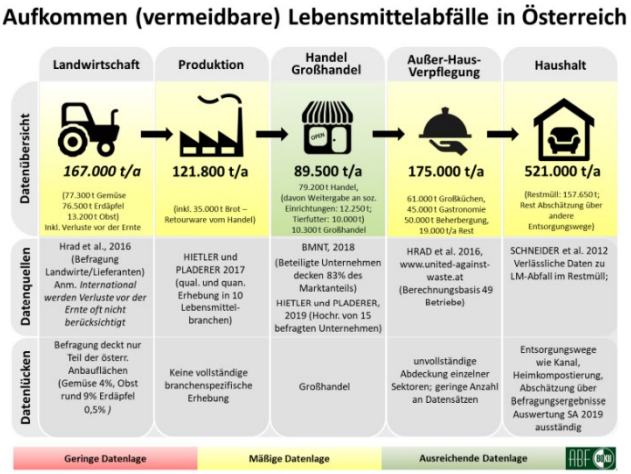


Abbildung 4: Lebensmittelabfälle in Österreich (vgl. WWF Lebensmittelabfälle, 2023)

Wie in der oben befindlichen Grafik unschwer zu erkennen ist, wird der größte Teil der Lebensmittel von privaten Haushalten weggeworfen. Mit unserer App kann in diesem Bereich die Verschwendung von Lebensmittel erheblich minimiert werden. Die Verwaltung des eigenen Kühlschrankes zu Hause, ist momentan sehr unübersichtlich. Selbst weiß man gar nicht, welche Produkte sich aktuell in seinem Kühlschrank befinden oder kurz vor dem Ablaufen stehen. Wenn der Kühlschrank geöffnet wird, bekommt man manche Lebensmittel gar nicht zu Gesicht, da diese hinter anderen liegen. Gleichermaßen kommt es vor, dass Lebensmittel eingekauft werden, welche sich bereits zu Hause im Kühlschrank befinden. Diese verweilen dann doppelt im Haushalt, somit wird es noch mühsamer diese Lebensmittel zu verbrauchen. Außerdem kann es vorkommen, dass beim Einkaufen viel zu viele Lebensmittel eingekauft werden, welche nicht auf den Verbrauch des Käufers abgestimmt sind. Dadurch wird es wiederum schwer, die Produkte aufzubrauchen.
Ebenfalls passiert es oft, dass Produkte gefunden werden, welche schon seit mehreren Tagen oder Wochen abgelaufen sind. Leider kommt es vor, dass diese abgelaufenen Produkte

dann entsorgt werden müssen, da diese auf Grund des fortgeschrittenen Ablaufdatums nicht mehr genießbar sind.
Durch unsere Android-App können all diese Dinge nicht mehr passieren.
Der Benutzer weiß zu jedem Zeitpunkt, welche Produkte sich aktuell in seinem Kühlschrank befinden. Auch die in den nächsten Tagen ablaufenden Produkte sind für den Benutzer von größter Bedeutung. All das realisiert Smood.

9.1.1 Relevanz hinsichtlich ökologischer, sozialer und ökonomischer Nachhaltigkeit

Ökologische Nachhaltigkeit

Eines der größten Probleme unserer Zeit ist der menschengemachte Klimawandel. Dieser Klimawandel wird durch zahlreiche Aspekte wie das Ausstoßen von CO_2 immer weiter verstärkt. Darüber hinaus behandeln wir unsere Erde nicht gut. Durch unsere marktwirtschaftlichen Systeme schaffen wir es, dass die Ressourcen auf unseren Planeten immer weniger werden. Die Überproduktion, die vom Menschen an den Tag gelegt wird, ist Grund dafür, dass alle natürlichen Ressourcen, die in einem Jahr nachwachsen, am 28. Juli eines Jahres bereits aufgebraucht sind. Über kurz oder lang werden wir so keine Ressourcen mehr auf der Erde vorfinden. Wir müssen lernen zu verstehen, dass der finanzielle Gewinn nicht unendlich sein kann und irgendwann einfach ein Limit erreicht ist. Die menschliche Gier nach Geld treibt uns so weit, dass wir unseren Lebensraum vernichten, nur um am Ende des Jahres den Aktionären ein Plus präsentieren zu können. Wir verstehen nicht, dass der größte Gewinn, den wir machen können, die Natur in ihrer Schönheit zu schützen ist. Es ist an der Zeit mit den Ressourcen, die wir haben, vernünftig umzugehen.
Unser Projekt zielt auf dieses Thema besonders ab, da wir mit unserem Vorhaben Ressourcen sparen wollen. 1,3 Mrd. Tonnen an Lebensmittel werden pro Jahr weltweit weggeschmissen. (vgl. global2000.at, 2023). Das sind 180kg pro Kopf. Für diese 1,3 Mrd. Tonnen wurde in der Produktion CO_2 ausgestoßen, umweltschädliche Dünger verwendet, Arbeitskraft aufgewandt etc. nur damit dies 1,3 Mrd Tonnen an Lebensmittel schlussendlich im Müll landen. Wenn wir schon 1kg pro Kopf an Lebensmittel vor dem Müll retten können, würden wir unsere Diplomarbeit als erfolgreich einstufen können.

Ökonomische Nachhaltigkeit

Etwas, was bereits bei der ökologischen Nachhaltigkeit angesprochen wurde, ist die Gier nach Geld. Unternehmen sind immer nach einer Gewinnmaximierung aus, koste es was es wolle. Für diese Optimierung des Gewinnes wird vor nichts zurückgeschreckt, nicht einmal vor der Umwelt. Die Unternehmen müssen lernen, wie sie ihre finanziellen Interessen, ohne die Umwelt zu schädigen, umsetzten können. Hierbei wird gefunden, dass es sehr wichtig ist unnötige Verpackungen (Plastikmüll) zu vermeiden. Oder generell verschiedene Verpackungen oder Behälter durch Mehrwegsysteme zu ersetzen. Darüber hinaus sollten sich Unternehmen Gedanken machen, wie sie die Langlebigkeit ihrer Produkte verbessern können, um Rohstoff zu sparen. Hinzugefügt wird noch, dass die Senkung des Wasserverbrauchs und Energien während der Produktion essenziell für eine nachhaltige Produktion ist. (vgl. net4energy.com, 2023)

Dieser Nachhaltigkeitsaspekt wird mit dieser Diplomarbeit ebenfalls verfolgt. Indem der Konsum von Lebensmittel effizienter gestaltet wird, müssen weniger Lebensmittel eingekauft werden. Daraus resultierend müssen private Haushalte weniger Geld ausgeben und verbrauchen weniger Ressourcen. Das Ziel dieser Nachhaltigkeitsform ist somit erfüllt. Das Budget der Privathaushalte wird geschont und damit einhergehend wird die Umwelt nicht so stark belastet, da weniger Ressourcen verschwendet / aufgewendet werden.

Diese Form der Nachhaltigkeit zu verstehen und richtig anzuwenden, ist die einzige Chance, dass der Kapitalismus und der Umweltschutz nebeneinander existieren können.

Soziale Nachhaltigkeit

Bei den ganzen Umweltdebatten darf man natürlich nicht auf die Menschen, die unter schlechten Arbeitsbedingungen, Ausbeutungen, Armut und mehr leiden, vergessen. Eine soziale Nachhaltigkeit hinsichtlich gleicher Rollenverteilung, Bekämpfung der Armut, Verringerung der Arbeitslosigkeit, gerechte Entlohnungen würden die Gesellschaft, in der wir leben, massiv stärken und voranbringen. Hierbei stehen vor allem Unternehmen in der Verantwortung, diese soziale Nachhaltigkeit zu fördern, da diese von ihnen derzeit nicht berücksichtigt wird. Mit einer stabileren und finanziell stärkeren Gesellschaft würden uns Themen wie der Umweltschutz um einiges leichter fallen, da wir mehr Budget hätten, um auf umweltschädigende Produkte verzichten zu können.

Bei genauerer Betrachtung wird dieser Nachhaltigkeitsgedanke ebenfalls von dieser Applikation abgedeckt. Familien, die wenig Geld für Lebensmittel zur Verfügung haben, können durch unsere App Geld sparen, indem sie weniger wegwerfen müssen bzw. durch die Empfehlungen der App effizienter einkaufen können. Darüber hinaus ist die Situation bei Familien mit wenig Budget besonders brisant. Da diese Familien weniger Geld für Lebensmittel zur Verfügung haben, müssen sie zwangsweise billigere Produkte kaufen. Diese Produkte haben oftmals ein niedrigeres Haltbarkeitsdatum. Darum ist es dort umso entscheidender den Überblick über die Lebensmittel zu haben.

Nur weil wir in diesem Beitrag die verschiedenen Formen der Nachhaltigkeit einzeln beleuchtet haben, heißt es nicht, dass ein Aspekt wichtiger ist als die anderen. Jede Form der Nachhaltigkeit, egal ob ökologisch, sozial oder ökonomisch muss zwangsweise angewendet werden, um den Klimawandel stoppen zu können.

9.2 Motivation

Grund für die Idee ist die Unübersichtlichkeit eines Kühlschrankes bzw. einer Speisekammer. Es ist oft den wenigsten klar, welche und wie viele Lebensmittel sie gerade auf Lager haben. Besonders problematisch ist dies bezogen auf die Haltbarkeit der Lebensmittel. Durch die hohen Lebensmittelverschwendungen, die jedes Jahr in vielen Haushalten anfallen, werden wichtige Ressourcen verschwendet und die Umwelt belastet. Die Idee hinter Smood ist es, eine Lösung zu schaffen, die es dem Endbenutzer ermöglicht, jederzeit den Überblick über seine Lebensmittelvorräte und deren Haltbarkeit zu behalten, um Lebensmittelverschwendung zu vermeiden.

Smood besteht aus einer App, die auf ein Android Smartphone heruntergeladen werden kann. In dieser App gibt es einen Barcode-Scanner. Jedes Mal, wenn ein Lebensmittel in den Kühlschrank gelegt wird, muss der Benutzer dieses Produkt erfassen. Die App zeigt dann an, welche Lebensmittel im Kühlschrank vorhanden sind und wie lange sie noch haltbar sind. Auf diese Weise kann der Nutzer leicht überblicken, was er zu Hause hat und was bald aufgebraucht werden muss, um die Lebensmittelverschwendung zu vermeiden.

Smood bietet auch einen Algorithmus, der Einkaufslisten erstellt. Auf diese Weise können Nutzer ihre Einkäufe gezielt planen und nur das kaufen, was sie wirklich brauchen.
Smood ist also eine innovative Lösung, die dazu beitragen kann, das Problem der Lebensmittelverschwendung in vielen Haushalten zu bekämpfen. Es hilft dem Nutzer, seine Lebensmittelvorräte zu organisieren und gezielt zu planen. Smood kann nicht nur zu Einsparungen beim Einkauf beitragen, sondern auch die Belastung der Umwelt durch Lebensmittelentsorgung verringern.

9.3 Ziele der Diplomarbeit

Das Ziel der Diplomarbeit ist die Entwicklung einer Android-App, mit der es möglich ist, Lebensmittel im Haushalt verwalten zu können. Diese App soll durch den P@bs vermarktet werden und dadurch soll der Bekanntheitsgrad sowie die innovative Präsenz des Absolventenverband gesteigert werden. Mittels der App soll der Einkauf eines Konsumenten so optimal gestaltet werden wie möglich. Der User soll zu jedem Zeitpunkt wissen, welche Produkte er bestmöglich kaufen soll, um den eigenen Konsum uneingeschränkt beibehalten zu können. Zudem soll ein „Überkonsum“ (mehr Einkaufen als man selbst verbrauchen kann) vermieden werden. Der User soll zu jedem Zeitpunkt wissen, welche Lebensmittel im Haushalt vorliegen und wann diese ablaufen.

Geschäftsziele

- Effektiveres Einkaufen
- Reduzierung von Lebensmittelverschwendung
- Breiteres Aufstellen des P@bs

Produktziele

- Der User kann seine Produkte händisch oder mittels Barcodes hinzufügen, verwalten und löschen, wenn diese verbraucht, wurden
- Der Algorithmus analysiert die Ablaufdaten und den Verbrauch und erstellt auf Basis dessen Einkaufsvorschläge mit einer Mengenangabe für den Benutzer
- Wenn Lebensmittel kurz vor dem Verfall stehen, soll der Benutzer eine Benachrichtigung auf sein Smartphone bekommen
- Das User Interface soll auf Android verfügbar sein

Prozessziele

- Verlässliche und pünktliche Einhaltung vereinbarter Termine.
- Qualitätssicherung aller Dokumente durch das Projektteam
- Projektstrukturplan wird laufend vom Projektteam angepasst

Erfolgskriterien

- Zeitgerechte Fertigstellung bis 04/2023
- Endprodukt soll keine Bugs enthalten
- Einkaufsvorschläge werden für jeden User gezielt getroffen und nicht willkürlich
- Aus 10 Fällen sollen 7 Vorschläge sinnvoll sein
- Produkt soll die oben genannten Produktziele erfüllen

9.4 Leistungsumfang

Leistungsumfang hinsichtlich der Organisation

In Scope	Out of Scope
Einteilung der Projektmitglieder in verschiedene Aufgabenbereiche	Abstimmung innerhalb des P@BS
Abstimmung mit dem Auftraggeber	
Organisatorische Tools bestimmen	

Tabelle 1: Leistungsumfang hinsichtlich der Organisation

Leistungsumfang hinsichtlich der Lokationen

In Scope	Out of Scope
Entwicklung im Rahmen der Diplomarbeit	andere Standorte
Abstimmung P@bs (Lukas Huber) und der HTL-Perg	

Tabelle 2: Leistungsumfang hinsichtlich der Lokationen

Leistungsumfang hinsichtlich der Applikation

In Scope	Out of Scope
Android-App	Installer
Datenbankschema und Datenbank	Online-Shop für Lebensmittel
Algorithmus	Rezeptvorschläge
REST-API	

Tabelle 3: Leistungsumfang hinsichtlich der Applikation

Leistungsumfang hinsichtlich der Technologie

In Scope	Out of Scope
Server	Andere Programmiersprachen
Datenbank	
Java, Android Studio	

Tabelle 4: Leistungsumfang hinsichtlich der Technologie

Leistungsumfang hinsichtlich der Daten

In Scope	Out of Scope
Erweiterbarkeit soll gewährleistet sein	
DSGVO einhalten	
Lebensmittelstammdatenbank einbinden	

Tabelle 5: Leistungsumfang hinsichtlich der Daten

9.5 Vorgehensweise

9.5.1 Vorgehensmodell

Als Vorgehensmodell wird bei dieser Diplomarbeit Scrum verwendet.

> Scrum ist ein Rahmenwerk für die Zusammenarbeit von Teams basierend auf einer Definition von Rollen, Meetings und Werkzeugen, die einem Team Struktur und einen klar definierten Arbeitsprozess basierend auf agilen Prinzipien geben. (digitaleneuordnung.de, 2022)

Scrum ist ein agiles Vorgehensmodell im Projektmanagement.
Das Hauptaugenmerk von Scrum liegt auf dem Product Backlog, wo die Anforderungen gesammelt werden. Außerdem wird die iterative Projektentwicklung in Zyklen zeitlich und klar definiert. Weiters gibt es drei zentrale Rollen. Der Product Owner vertritt den Auftraggeber. Die Aufgaben des Product Owners sind die Beschreibung und das Gewichten der Anforderungen. Der Scrum Master leitet den Entwicklungsprozess und entlastet den Product Owner. Der Scrum Master ist kein Teammitglied, da er das Projekt leitet. Schlussendlich gibt es auch noch das Team. Dieses ist eine Gruppe von Entwicklern, die das Projekt umsetzten.
Der Ablauf erfolgt wie nachfolgend beschrieben. Im Sprint Planning Meeting wird das Ziel der einzelnen Sprints festgelegt. Ein Scrum Sprint ist ein zeitlich begrenzter Zeitraum, in dem ein Team bestimmte Ziele erreichen möchte. Es werden Sprints verwendet, um kleine Teile eines Projekts zu erledigen und Fortschritte sichtbar zu machen. Sprints haben in der Regel eine festgelegte Dauer, die typischerweise zwischen einer und vier Wochen liegt, und sie sind von anderen Sprints abgegrenzt. Während eines Sprints plant das Scrum-Team, welche Aufgaben es erledigen möchte, und arbeitet dann an diesen Aufgaben, um sie am Ende des Sprints abzuschließen. Am Ende eines jeden Sprints wird eine Überprüfung (Review) durchgeführt, um die Fortschritte des Teams zu bewerten und Verbesserungsmöglichkeiten zu identifizieren. Sprints helfen dabei, Projekte aufzuteilen und Fortschritte sichtbar zu machen. (vgl. Systemplanung und Projektentwicklung, 2013, S. 209-211)

9.5.2 Rollen und Ablauf

Product Owner

Bei dieser Diplomarbeit wird der Product Owner durch Fabian Kopf repräsentiert.

Scrum Master

In unserem Team gibt es keinen Scrum Master, da wir eine zu kleine Diplomarbeitsgruppe sind. Die Tätigkeiten vom Scrum Master werden durch die Projektgruppe ersetzt. Das bedeutet, wenn Probleme auftreten, sei es technisch oder sozial, setzt sich die Gruppe zusammen und versucht diese zu lösen.

Team

Das Diplomarbeitsteam setzt sich aus vier Personen zusammen: Bodingbauer Julia, Dannhofer Jana, Lechthaler Moritz und Kopf Fabian.

Sprint

Ein Sprint hat eine Länge von drei Wochen. Hierbei hat sich angeboten, den Auftraggeber alle 3 Wochen einen Statusbericht zu präsentieren.

Sprint Planning Meeting

Im Sprint Planning Meeting werden die Ziele für den Sprint fest. Die Tasks werden nach ihrer Priorität ausgewählt und müssen vom Umfang in einen Sprint passen. Zudem werden die User Stories definiert.

Weekly Scrum

Bei dieser Diplomarbeit ist ein Weekly Scrum von Vorteil, da der Daily Scrum weniger Sinn ergeben würde, weil das Team nicht gewährleisten kann, dass jeder jeden Tag an der Diplomarbeit arbeitet. Darüber hinaus wäre es zu viel Organisatorischer Aufwand. Daher wurde das Scrum Meeting auf wöchentliche Abstände festgelegt.

Sprint Review

Das Ergebnis des Sprints wird alle drei Wochen besprochen und auftretende Fragen werden mit dem Auftraggeber geklärt.

Sprint Retrospektive

Nach jedem Sprint wird nach dem Sprint Review ein internes Meeting abgehalten, indem das ganze Team eine Sprint Retrospektive durchführt. Jeder legt Probleme, negative und positive Aspekte des vergangenen Sprints dar, damit Anpassungen für den nächsten Sprint gemacht werden können. Der Product Owner moderiert das Meeting.

Fortschrittskontrolle

User Stories

Die User Stories werden im Sprint Planning Meeting vom Product Owner mit Unterstützung der Teammitglieder definiert. Die jeweiligen Teammitglieder halten sich an diese User Stories und deren Aufgaben.

Sprint Backlog

Hierbei wird ein Board verwenden, damit offene, erledigte Aufgaben und User Stories, welche noch in Arbeit sind, grafisch darstellen.

Projektstrukturplan

Um zu veranschaulichen, in welche Teile das Projekt zu gliedern ist und wann welcher Teil erledigt sein sollte, damit der Abgabetermin gehalten werden kann, verwendet diese Diplomarbeit und sein Team einen Projektstrukturplan (PSP). An diesem Plan kann man sich im Laufe der Diplomarbeit orientieren.

Zeiterfassung

Als Zeiterfassungssoftware wird in dieser Diplomarbeit „Clockify“ verwendet.

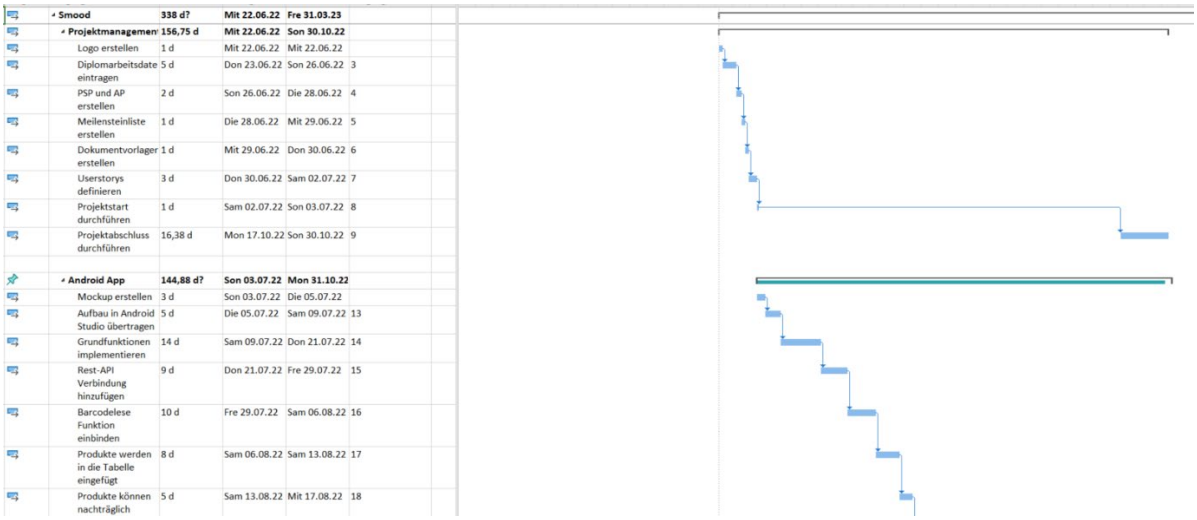
9.5.3 Projektmanagement

Die Projekt-Baseline ist eine akkordierte und freigegebene Projektplanung, die aus dem Projektstrukturplan und aus einer Meilensteinliste besteht.
Die Meilensteine werden im Kapitel Planung und Realisierung beschrieben.

Projektstrukturplan

Was ist ein Projektstrukturplan und warum sollte man ihn verwenden?

Ein Projektstrukturplan ist ein Dokument, das die Aufgaben, die für das erfolgreiche Abschließen eines Projekts erforderlich sind, in logische Einheiten unterteilt. Der Projektstrukturplan hilft, das Projekt in überschaubare Teile zu unterteilen, indem er die Aufgaben und Leistungen des Projekts in einer hierarchischen Struktur organisiert.
Ein Projektstrukturplan sollte verwendet werden, um die Ziele und die Ergebnisse des Projekts zu definieren und um sicherzustellen, dass alle erforderlichen Aufgaben abgeschlossen werden, um das Projekt erfolgreich abzuschließen. Er hilft auch dabei, das Projekt zu planen, indem er die Aufgaben und Leistungen in logische Einheiten unterteilt und den Fortschritt des Projekts verfolgt. (vgl. Systemplanung und Projektentwicklung, 2013, S. 116-118)
Im Darunterliegenden Bild befindet sich ein auszug aus dem Projektstrukturplan von der Diplomarbeit Smood.



	Smood	338 d?	Mit 22.06.22	Fre 31.03.23	
	Projektmanagemen	156,75 d	Mit 22.06.22	Son 30.10.22	
	Logo erstellen	1 d	Mit 22.06.22	Mit 22.06.22	
	Diplomarbeitsdate eintragen	5 d	Don 23.06.22	Son 26.06.22	3
	PSP und AP erstellen	2 d	Son 26.06.22	Die 28.06.22	4
	Meilensteinliste erstellen	1 d	Die 28.06.22	Mit 29.06.22	5
	Dokumentvorlager erstellen	1 d	Mit 29.06.22	Don 30.06.22	6
	Userstorys definieren	3 d	Don 30.06.22	Sam 02.07.22	7
	Projektstart durchführen	1 d	Sam 02.07.22	Son 03.07.22	8
	Projektabschluss durchführen	16,38 d	Mon 17.10.22	Son 30.10.22	9
	Android App	144,88 d?	Son 03.07.22	Mon 31.10.22	
	Mockup erstellen	3 d	Son 03.07.22	Die 05.07.22	
	Aufbau in Android Studio übertragen	5 d	Die 05.07.22	Sam 09.07.22	13
	Grundfunktionen implementieren	14 d	Sam 09.07.22	Don 21.07.22	14
	Rest-API Verbindung hinzufügen	9 d	Don 21.07.22	Fre 29.07.22	15
	Barcodelese Funktion einbinden	10 d	Fre 29.07.22	Sam 06.08.22	16
	Produkte werden in die Tabelle eingefügt	8 d	Sam 06.08.22	Sam 13.08.22	17
	Produkte können nachträglich	5 d	Sam 13.08.22	Mit 17.08.22	18

Abbildung 5:Auszug aus dem Projektstrukturplan

9.5.4 Projektorganisation

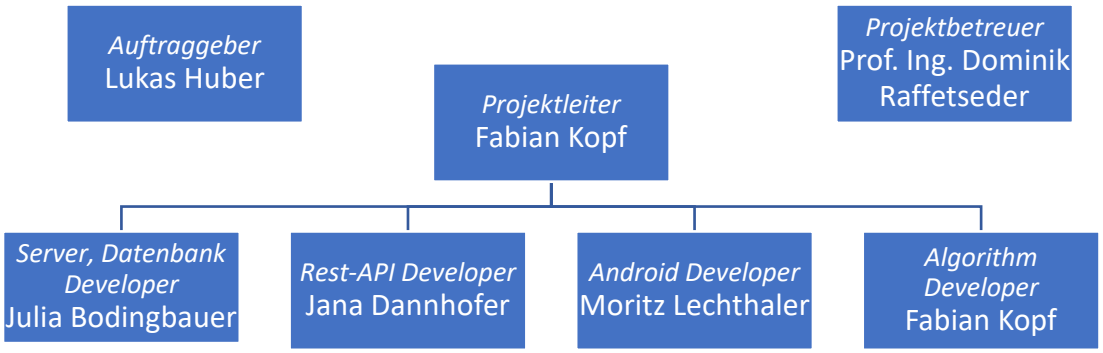


Abbildung 6: Projektorganisation

Projektleiter

Der Projektleiter spielt eine zentrale Rolle in der Diplomarbeit und hat eine Vielzahl von Verantwortlichkeiten. Eine seiner Hauptaufgaben besteht darin, die Kommunikation mit dem Auftraggeber sicherzustellen. Außerdem muss er sicherstellen, dass die Anforderungen und Erwartungen des Auftraggebers erfüllt werden. Ein weiterer wichtiger Aspekt der Rolle des Projektleiters besteht in der Koordination des Projektteams. Der Projektleiter ist dafür verantwortlich, dass das Team effektiv zusammenarbeitet und dass jeder Mitarbeiter seine Aufgaben erfüllt. Er muss sicherstellen, dass alle Arbeitsabläufe im Projekt reibungslos verlaufen und dass das Team effektiv kommuniziert. Diese Rolle hat Fabian Kopf übernommen.

Server, Datenbank Developer

Dieser Teilbereich umfasst die Zuständigkeit für den Server, die Datenbank und die Push-Notifications. Diesen Teilbereich bearbeitete Julia Bodingbauer.

REST-API Developer

Dieser Teilbereich umfasst die Zuständigkeit für die REST-API und wurde von Jana Dannhofer erledigt.

Android Developer

Dieser Teilbereich umfasst die Zuständigkeit für die Android-Applikation und wurde von Moritz Lechthaler bearbeitet.

Algorithm Developer

Dieser Teilbereich umfasst die Zuständigkeit für den Algorithmus und wurde von Fabian Kopf bearbeitet.

9.6 Funktionsbereiche der App

9.6.1 Benutzerverwaltung

In der Diplomarbeit kann sich jeder User einen Benutzer erstellen. Mit diesem Benutzer kann dieser sich in der App anmelden und die anfallenden Daten können jeden User zugeordnet werden. Um sich einen Benutzer zu erstellen, benötigt man ein Passwort und einen Usernamen.

9.6.2 Lebensmittelverwaltung

Jedes Lebensmittel wird in der Datenbank mit seinem Ablaufdatum abgespeichert. Im Kapitel Home auf der Seite 51 muss nur das Produkt gefunden oder abgescannt werden. Somit kann das Produkt aus der Datenbank geladen werden und das Ablaufdatum festgelegt werden. Danach findet man das Produkt unter seiner Kategorie, in der Liste auf der Lebensmittelansicht. Wenn das Produkt kurz vor dem Ablaufen ist, wird es dem Benutzer rechtzeigt auf der Abgelaufene – Empfohlene Lebensmittel Seite in der Liste der Abgelaufenen-Produkte, welche auf der Seite 50 erklärt werden, angezeigt. Sobald man das Produkt verbraucht hat, egal ob es schon abgelaufen ist oder nicht, muss man das Produkt auf Aufgebraucht setzten.

9.6.3 Analyse des Benutzerverhalten

Bei der Benutzeranalyse analysiert der Algorithmus alle Produkte des Benutzers, welche er jemals ins System eingetragen hatte. So kann der Algorithmus das Verhalten des Nutzers feststellen. Die genauere Erklärung findet man auf der **Fehler! Textmarke nicht definiert.**. Nach der Analyse sind Empfohlenen Lebensmittel in der App ersichtlich.

9.6.4 Empfehlung für Benutzer

Die Produkte, die zu dem angemeldeten Benutzer zugeordnet sind, werden aus der Datenbank geladen. Diese Datenbankeinträge wurden im vorherigen Schritt, durch den Algorithmus, eingefügt. Die genauen Aufrufe, welche die App tätigen muss, dass diese die Daten bekommt, werden im Kapitel Empfohlene-Produkte genauer beschrieben.

9.6.5 Push Benachrichtigungen

Was sind Push-Benachrichtigungen und wie funktionieren sie im Allgemeinen?

Push-Benachrichtigungen sind Nachrichten, die von einer App oder einem Dienst an ein mobiles Gerät gesendet werden, um den Benutzer über bestimmte Ereignisse oder Informationen zu informieren. Push-Benachrichtigungen werden normalerweise auf dem Sperrbildschirm oder in der Benachrichtigungsleiste des Geräts angezeigt, sodass der Benutzer diese leicht sehen kann. Push-Benachrichtigungen können verschiedene Arten von Informationen enthalten, wie z.B. Neuigkeiten, Wettervorhersagen, Echtzeit-Updates und Erinnerungen. Sie werden häufig von Apps verwendet, um die Benutzer über neue Inhalte oder Funktionen zu informieren oder um sie an wichtige Ereignisse oder Aufgaben zu erinnern. Push-Benachrichtigungen können auf den meisten modernen mobilen Geräten empfangen werden, die über eine Internetverbindung verfügen.
Push-Benachrichtigungen werden über ein Netzwerkprotokoll namens "Push Notification Service" (PNS) übertragen, das von den Betriebssystemen der mobilen Geräte unterstützt wird. Wenn eine App oder ein Dienst eine Push-Benachrichtigung senden möchte, sendet sie diese an den PNS, der sie dann an das mobile Gerät des Benutzers weiterleitet.

Um Push-Benachrichtigungen zu empfangen, muss das mobile Gerät über eine Internetverbindung verfügen. Der Benutzer muss außerdem die Erlaubnis erteilt haben, dass die App oder der Dienst Push-Benachrichtigungen senden darf. Sobald die Push-Benachrichtigung vom PNS empfangen wurde, wird sie auf dem Sperrbildschirm oder in der Benachrichtigungsleiste des Geräts angezeigt. Der Benutzer kann dann auf die Benachrichtigung tippen, um sie zu öffnen und weitere Informationen anzuzeigen. (vgl. bluesource.at, 2023), (vgl. twilio.com, 2023).

Push Benachrichtigungen – Smood

Push-Benachrichtigungen können sehr nützlich sein, um Produkte vor dem Ablaufen zu schützen, da sie dazu beitragen, dass sich die Menschen an das Verfallsdatum von Produkten erinnern. Oft vergessen sie, wann sie bestimmte Lebensmittel oder andere Produkte im Kühlschrank gekauft haben, und es ist leicht, dass diese Lebensmittel verderben, ohne dass die Leute es bemerken. Mit Push-Benachrichtigungen erhalten sie jedoch eine Erinnerung, die sie daran erinnert, dass ein bestimmtes Produkt bald ablaufen wird, und können es rechtzeitig verwenden, bevor es schlecht wird.
Wenn die Benutzer der App Smood jedoch regelmäßig Erinnerungen erhalten, werden sie sich bewusster, welche Produkte sie haben und wann sie verwendet werden müssen. Auf diese Weise können die Benutzer besser planen, was sie kaufen und essen möchten, und vermeiden, dass Lebensmittel gekauft werden, die bereits zu Hause im Kühlschrank sind oder die bald ablaufen.
Push-Benachrichtigungen können auch dazu beitragen, Lebensmittelverschwendung zu reduzieren. Durch das Erinnern an das Verfallsdatum von Produkten können die Benutzer der App sicherstellen, dass sie die Lebensmittel rechtzeitig verwenden und nicht in den Müll werfen müssen. Dies ist sowohl für den Geldbeutel als auch für die Umwelt von Vorteil.

10 Grundlagen

10.1 Verwendete Technologien

10.1.1 Ubuntu

Ubuntu ist eine auf Debian basierende Linux-Distribution. Dieses Betriebssystem wird von Canonical finanziert und von der Community weiterentwickelt. Ubuntu wird häufig als Alternative zu anderen Betriebssystemen wie Microsoft Windows oder macOS verwendet. Ubuntu ist leicht zu installieren und zu verwenden und bietet eine Vielzahl von Anwendungen und Werkzeugen, die für den täglichen Gebrauch und die Entwicklung von Software benötigt werden. Es wird auch regelmäßig aktualisiert und bietet zahlreiche Sicherheitsfunktionen, um den Schutz der Benutzerdaten zu gewährleisten. Ubuntu ist eine beliebte Wahl für Menschen, die ein stabiles und leistungsstarkes Betriebssystem suchen, das auf Open-Source-Technologien basiert. (vgl. heise.de, 2022)

Abbildung 7: Ubuntu Logo (design.ubuntu.com, 2022)

In dieser Diplomarbeit wird die Ubuntu Server-Version 22.04 LTS verwendet, auf der die MySQL-Datenbank und die REST-API laufen.

10.1.2 MySQL

MySQL ist ein weitverbreitetes Open-Source Datenbankmanagementsystem, das relationale Tabellen abbildet. MySQL arbeite als Client-Server-Modell. Es wird häufig in Web-Anwendungen eingesetzt, da es leicht zu verwenden und leistungsstark ist. MySQL verwendet eine relationale Datenbankstruktur, in der Daten in Tabellen gespeichert werden, die wiederum durch Beziehungen verbunden sind. Mit MySQL können Benutzer Daten abfragen, einfügen, aktualisieren und löschen, indem sie SQL-Anweisungen verwenden. MySQL bietet auch viele fortgeschrittene Funktionen wie Indexierung, Sicherheitsfunktionen und Replikation, die es zu einem leistungsstarken Werkzeug für die Datenverwaltung machen. Der MySQL-Server verarbeitet die Datenbankanweisungen

Abbildung 8: MySQL Logo (dev.mysql.com, 2022)

und Befehle. Der Server ist als Bibliothek erhältlich, der in verschiedene Applikationen eingebunden werden kann. (vgl. ComputerWeekly.de, 2022)
Bei dieser Diplomarbeit wird die kostenlose Version von MySQL verwendet.

10.1.3 SQL

SQL (Structured Query Language) ist eine standardisierte Datenbanksprache, die verwendet wird, um Daten in relationalen Datenbanken zu verarbeiten und zu verwalten. Mit SQL können Benutzer Daten abfragen, einfügen, aktualisieren und löschen, indem sie spezielle Anweisungen in Form von Abfragen verwenden. SQL ist eine wichtige Sprache in der Datenbankverarbeitung und wird häufig in Unternehmen und anderen Organisationen verwendet, um große Mengen an Daten zu verwalten und zu analysieren. Es gibt verschiedene Dialekte von SQL, die von verschiedenen Datenbankherstellern verwendet werden, aber es gibt auch eine Standardisierung der Sprache, die von der ISO (International Organization for Standardization) und der ANSI (American National Standards Institute) festgelegt wurde. Der Vorteil von SQL ist, dass diese Abfragesprache an die englische Sprache angelehnt und somit einfach zu verstehen ist. (vgl. YouTube: SQL Einführung 1, 2022)
Mithilfe von SQL wird in dieser Diplomarbeit mit der MySQL-Datenbank kommuniziert.

10.1.4 Retrofit

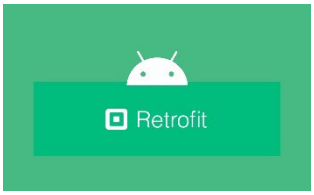


Abbildung 9: Retrofit Logo (engineering.creativesociety, 2022)

Retrofit ist eine typsicherer HTTP – Client für Android, Java und Kotlin der von Square entwickelt wurde. Mit Retrofit können Entwickler leicht Anfragen an eine API senden und die Antworten in POJOs (Plain Old Java Objects) umwandeln, die in der Anwendung verwendet werden können. Retrofit verwendet Annotations, um die API-Endpunkte und die Art der Anfragen (z.B. GET, POST, PUT, DELETE) zu definieren, und ermöglicht es den Entwicklern, die Anfragen und Antworten synchron oder asynchron zu verarbeiten. Retrofit bietet auch Unterstützung für die Integration von Convertern, die es ermöglichen, andere Datenformate als JSON zu verwenden, und für Authentifizierungsmechanismen wie OAuth. (vgl. topcoder.com, 2022; code.tutsplus.com, 2023)
Mithilfe von Retrofit und dem von uns erstellten Interface kann man in unserer Diplomarbeit auf die REST-API zugreifen.

10.1.5 Apache Tomcat

Abbildung 10: Apache Tomcat Logo (appuntisoftware.it, 2022)

Apache Tomcat ist eine Open-Source-Software für die Jakarte EE-Plattform. Diese stellt eine HTTP-Webserverumgebung zur Verfügung. Circa 40 % aller Webseiten wird von Apache Tomcat unterstützt.

Apache Tomcat bietet auch Unterstützung für andere Web-Technologien wie HTTP, SSL und TLS und kann als Reverse-Proxy-Server verwendet werden, um Anfragen von Clients an andere Server weiterzuleiten. Apache Tomcat ist ein leistungsstarker und zuverlässiger Web-Server, der häufig in Unternehmensumgebungen und in der Entwicklung von Web-Anwendungen eingesetzt wird. (vgl. biteno.com, 2022)

In dieser Diplomarbeit wird Apache Tomcat am Server verwendet, um dort die REST-API zu hosten.

10.1.6 Java-Technologie

Java ist eine kostenlose Technologie, welche im Jahr 1995 von Sun Microsystems entwickelt wurde. Java ist einerseits eine Computing – Plattform für Java-Anwendungen und andererseits eine weitverbreitete objektorientierte Programmiersprache[1]. Mittlerweile ist Java eine der bedeutendsten und zuverlässigsten Plattformen für viele Services und Applikationen. Auch in der heutigen Zeit basieren nach wie vor noch viele Anwendungen auf Java. Inzwischen wurde Sun Microsystems von Oracle aufgekauft.

Die Technologie Java beinhaltet das Java Runtime Environment oder kurz JRE. Dabei handelt es sich um die Laufzeitumgebung. Diese wird benötigt, um Java Programme ausführen zu können. Die Laufzeitumgebung beinhaltet die Java Virtual Machine (JVM) zur Interpretation des Bytecodes. Weiters sind die Plattform-Library und die Core-Klassen in der Technologie enthalten. Zur Java Entwicklung wird allerdings zusätzlich das Java Development Kit (JDK), zur Übersetzung des Sourcecodes in Bytecode und zusätzlich ein Editor, für den Sourcecode, benötigt.(vgl. Java, 2022)

[1] Softwarearchitektur welche der Wirklichkeit sehr nahe kommt

10.1.7 Java-Programmiersprache

Java ist eine der bekanntesten und meisten verwendeten Programmiersprachen. Eine Eigenschaft und gleichzeitig ein Vorteil der Programmiersprache ist die Plattformunabhängigkeit. Das bedeutet, dass eine in Java erstellte Anwendung auf jeder Plattform und jedem Betriebssystem gleich läuft.

Abbildung 11: Java Logo (Developer.com, 2022)

Java ist eine objektorientierte Programmiersprache, dadurch steigt auch die Wiederverwendbarkeit des Codes. Es werden sämtliche Techniken der Objektorientierung, wie Vererbungen und Überladungen unterstützt. Auch die Sicherheit spielt bei Java eine große Rolle. Unbeabsichtigte Zugriffe auf die Speicherbereiche werden durch die Interpreter[2] blockiert.
(vgl. Java-Tutorial, 2022)

Die REST-API und ebenfalls der Algorithmus dieser Diplomarbeit wurden in der Programmiersprache Java geschrieben.

10.1.8 Java Spring Boot

Seit 1995 ist Java eine der populärsten Programmiersprachen weltweit. Um das Arbeiten mit Java Programmen etwas zu vereinfachen, können Entwickler auf unzählige Frameworks zurückgreifen. Diese Frameworks stellen dem Entwickler ein fertiges und einsatzbereites Basisgerüst zur Verfügung. Eines der populärsten Frameworks ist Spring. Spring-Boot ist eine Erweiterung zu dem Spring Framework, welches die Konfiguration neuer Projekte vereinfacht. (vgl. IONOS, 2022)

[2] Zuständig für die Ausführung und Übersetzung eines Java Programmes

Was ist Spring-Boot?

Spring Boot ist ein 2012 erschienenes Framework, um das Programmieren von eigenständigen Anwendungen zu erleichtern. Dabei wird ein Basisgerüst vorgegeben, welches von den Entwicklern weiter aufgebaut wird. Einer der Vorteile von Spring-Boot ist der reduzierte Entwicklungsaufwand, besonders zu Beginn, der durch das vorgegebene Grundgerüst gegeben ist.

Abbildung 12: Spring Boot Logo (Java Guides, 2022)

Funktionen:

- Einfaches bauen von Stand-Alone Anwendungen
- Verbindung mit Applikationsservern wie Apache Tomcat oder Jetty[3]
- Einfache Build Konfigurationen
- Selbstständige Konfiguration von Spring Bibliotheken

(vgl. Spring Boot, 2022)

In dieser Diplomarbeit wird das Spring Boot Framework mit der Version 2.7.2 verwendet.

10.1.9 JPA

JPA steht für Java Persistence API, und ist eine Spezifikation, welche das Mapping auf relationale Datenbanken vereinfacht. Die Daten werden dadurch zwischen den Java Objekten und der Datenbank persistiert. Das geschieht durch den Einsatz von ORM[4] Tools, wie Hibernate[5] oder TopLink. Das hat den Vorteil, dass keine SQL-Abfragen verwendet werden müssen, sondern direkt mit Objekten gearbeitet werden kann.

(vgl. JavaPoint, 2022)

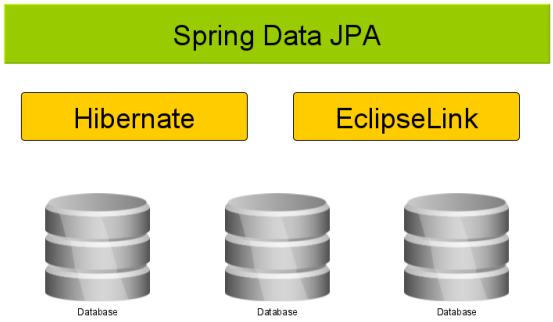


Abbildung 13: Spring Data JPA (Medium, 2022)

[3] Dienste, um mehrere Anwendungen in einem Prozess zu verwalten
[4] ORM = Object Relational Mapper, mappen Java- Objekten auf Tabellen einer Datenbank
[5] ORM-Tool für Java

Der größte Vorteil von Spring Data JPA ist die Fähigkeit, automatisch zur Laufzeit, eine Repository Implementierung aus einem Interface zu entwerfen.
(vgl. Spring, 2022)

In dieser Diplomarbeit wurde Spring Data JPA mit Hibernate verwendet. Als ORM-Tool ist auf Hibernate zurückgegriffen worden. Spring Data ist eine Erweiterung des Spring Frameworks.

10.2 Verwendete Entwicklungssysteme

10.2.1 JetBrains IntelliJ

IntelliJ IDEA ist eine spezielle Entwicklungsumgebung (IDE – Integrated Development Environment) für diverse Programmiersprachen. Hauptsächlich wird sie für Java und Python verwendet. IntelliJ bietet zahlreiche Assistenzfunktionen, die es für den jeweiliger Programmierer einfacher und übersichtlicher gestalten. Darüber hinaus verfügt IntelliJ IDEA über eine fortschrittliche Fehlerüberprüfung, die eine schnellere und einfachere Fehlerprüfung ermöglichen. (vgl. techopedia, 2022)

In der Diplomarbeit wird IntelliJ IDEA Ultimate in der Version 2022.1 und 2022.3 zur Entwicklung des Algorithmus, für das Analysieren des Benutzerverhalten, und zur Entwicklung der REST-API verwendet. Beide Bereiche werden mit Java programmiert.

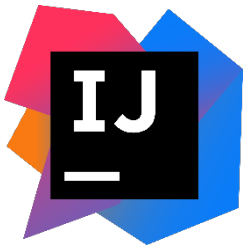


Abbildung 14: Intellij IDEA (https://www.jetbrains.com/de-de/idea/)

10.2.2 JetBrains Android Studio

Abbildung 15: Android Studio (https://www.jetbrains.com/)

Android Studio ist die offizielle IDE (Integrated development environment) für das Entwickeln von Android-Applikationen von Google. Das ganze Programm basiert auf IntelliJ IDEA und beinhaltet somit die gleichen Codebearbeitungen und Entwicklungswerkzeuge. Um das Entwickeln innerhalb des Android Betriebssystem zu unterstützen, verfügt die IDE einen Emulator, diverse Codevorlagen und ein Gradle basiertes Build-System. (vgl. techtarget, 2022)

In der Diplomarbeit wird Android Studio in der Version Dolphin 2021.3.1 Patch 1 zur Entwicklung der gesamten Android-App verwendet.

10.2.3MySQL Workbench

MySQL Workbench ist ein Werkzeug, welches ein User Interface für eine MySQL-Datenbank bereitstellt. Mit der MySQL Workbench können Datenmodellierungen und SQL-Entwicklungen durgeführt werden. Darüber hinaus umfasst diese Software verschieden Tools zur Serverkonfiguration, Benutzerverwaltung und für Backups. Über eine Datenbankverbindung können die bestehenden Daten in einer Datenbank angesehen, manipuliert und gelöscht werden. Des Weiteren ist es möglich, die Performance der Datenbank zu überwachen und zu analysieren. (vgl. mysql.com, 2022)

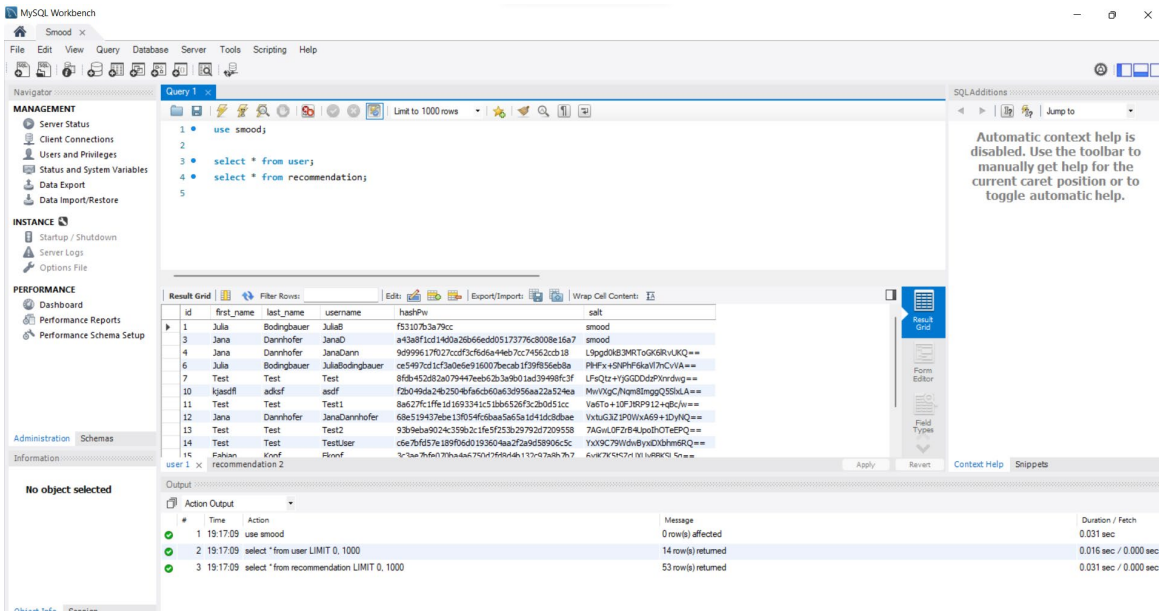


Abbildung 16: MySQL Workbench Abbildung

In der Diplomarbeit wird MySQL Workbench in der Version 8.0.29 zu jeglicher Interaktion mit der MySQL-Datenbank verwendet. Das bedeutet, zur Erstellung des Datenbankmodells und der Generierung der Datenbank mittels SQL-Scripts, wird MySQL Workbench genutzt. Darüber hinaus werden mit dieser Software-Daten selektiert, um diese auf ihre Richtigkeit und Vollständigkeit im Testbetrieb zu prüfen.

10.2.4 Sonstige verwendete Software

Postman

Abbildung 17: Postman-Logo (Postman, 2023)

Postman ist eine grafische IDEA zum Testen von API's. Mittels Postman ist es möglich API-Requests zu testen, bevor diese in der Diplomarbeit verwende werden. Des Weiteren lassen sich alle Request in einem Ordner sammeln damit die Übersicht über die bereits existierenden Requests stehts gewährleistet ist. (vgl. postman.com, 2023)

In dieser Diplomarbeit wird Postman zum Testen und zum Sammeln der API-Request verwendet.

Photoshop

Abbildung 18: Photohsop Logo (de.wikipedia.org, 2023)

Photoshop ist ein Bildbearbeitungsprogramm. Mit diesem Programm lassen sich Bilder bearbeiten, zusammenfügen oder neu erstellen. Die Software ist im Bereich der Bildbearbeitung Weltmarktführer. (vgl. adobe.com, 2023)

In der Diplomarbeit wird diese Software zur Gestaltung des Logos sowie diversen Plakaten verwendet.

Git

Abbildung 19: Github-Logo (upload.wikimedia.org, 2023)

Git ist ein Open Source Versionskontrollsystem von Dateien. Mittels git lassen sich Änderungen in Dateien erkennen und diese für jedes Teammitglied zugänglich machen. (vgl. git-scm.com, 2023)

Microsoft Office 365

Microsoft Office 365 ist eine Ansammlung von verschiedenen Applikationen. Von diesem Paket wurde folgende Applikationen verwendet:

- Word — Erstellen von Textdateien sowie verschriftlichen der Diplomarbeit
- Excel — Daten, Information übersichtlich in Tabellen darstellen
- PowerPoint — Erstellen von Präsentation über die Diplomarbeit
- To Do — Notizzettel für verschiedene Aufgaben
- Teams — Videokonferenzen mit dem Auftraggeber

(vgl. Microsoft.com, 2023)

Discord

Abbildung 20: Discord Logo (is3-ssl.mzstatic.com, 2023)

Discord ist ein Onlinedienst um per Textnachrichten, Voice Chats und Videobesprechung zu kommunizieren. (vgl. discord.com, 2023)
Diese Software wurde dazu genutzt, um Information auszutauschen, sich per Sprachchat verständigen und Termine im Überblick zu behalten.

Trello

Abbildung 21: Trello Logo (Logosmarken, 2023)

Trello ist ein auf Kanban basierender Aufgaben-Verwaltungs-Onlinedienst des Unternehmens Atlassian. Trello ist auch dafür geeignet, um Aufgaben zu visualisieren und zu veranschaulichen in welcher Phase sich eine Aufgabe befindet (ToDo, In Arbeit, Testen, Fertig). (vgl. stackfield.com, 2023)
In der Diplomarbeit wird Trello verwendet, um Aufgaben mittel des Kanban-Boards in Phasen einzuteilen und somit den Überblick über zu erledigenden Aufgaben und bereits Abgeschlossene zu haben.

11 Planung und Realisierung

In diesem Kapitel wird die Planung und Realisierung der Diplomarbeit genauer beleuchtet. Die Unterlagen der nachkommenden Unterkapitel wurden vor dem Start der praktischen Arbeit entworfen, um den fachlichen aber auch den zeitlichen Ablauf zu planen.

11.1 Meilensteine

Meilenstein Bezeichnung	SOLL-Termin	IST-Termin
Projektserver beantragt	20.05.2022	13.05.2022
Projektstrukturplan erstellt	19.06.2022	16.06.2022
Eintragen des Themas in Diplomarbeitsdatenbank	01.07.2022	26.06.2022
Projektstart erfolgt	08.07.2022	08.07.2022
Technologieentscheidung durchgeführt	08.07.2022	08.07.2022
Projektdefinition	08.07.2022	08.07.2022
Mockups für die App sind entworfen	10.07.2022	09.07.2022
Datenbankschema entwerfen	17.07.2022	15.07.2022
Server in Betrieb genommen	24.07.2022	23.07.2022
Datenbank aufgesetzt	27.07.2022	25.07.2022
Verbindung zwischen Algorithmus und Datenbank	07.08.2022	02.08.2022
Verbindung zwischen REST-API und Datenbank	07.08.2022	06.08.2022
Verbindung zwischen App und REST-API hergestellt	21.08.2022	25.08.2022
REST-API ist implementiert	03.09.2022	01.09.2022
Algorithmus gibt Einkaufsvorschläge	07.09.2022	03.09.2022

Algorithmus gibt Mengenvorschläge	07.09.2022	03.09.2022
Login in der App funktioniert	11.09.2022	12.09.2022
Registrierung in der App funktioniert	11.09.2022	15.09.2022
Produkte können in der App gespeichert werden	17.10.2022	16.10.2022
App kann Barcodes auslesen	23.10.2022	23.10.2022
App wird nach Vorlage der Mockups implementiert	23.10.2022	30.11.2022
Händisches einfügen der Lebensmittel von der App (wenn nicht in der Datenbank vorhanden)	23.10.2022	14.11.2022
Push-Notifications können gesendet werden	23.10.2022	01.11.2022
Projektende des praktischen Teils durchführen	31.01.2023	31.01.2023
Abstract in Englisch verfasst	28.02.2023	28.02.22
Diplomarbeit ist verschriftlicht	10.04.2023	5.4.2023

Tabelle 7: Meilensteine

In der oberen Grafik sind die einzelnen Meilensteine nochmal grafisch aufbereitet.

11.2 Verantwortlichkeiten

Die Verantwortlichkeiten hinsichtlich der technischen Ausarbeitung der Arbeit sind folgenden Maßen geregelt. Die vier wesentlichsten Bestandteile der Diplomarbeit sind REST-API, Server und die Push Benachrichtigungen, Algorithmus zum Analysieren des Benutzerverhalten und die Android-App.

- Julia Bodingbauer: Server + Push Benachrichtigen + Datenbank
- Jana Dannhofer: REST-API
- Moritz Lechthaler: Android-App
- Fabian Kopf: Algorithmus

Um die Verantwortlichkeiten im Allgemeinen besser darzustellen, wird mithilfe einer IVM-Matrix dargestellt, wie die einzelnen Teammitglieder in den einzelnen Bereichen partizipieren.

	Bodingbauer	Dannhofer	Lechthaler	Kopf
Projektmanagement	M	M	M	V
Server	V	I	I	I
Push Benachrichtigen	V	M	I	I
Rest-Schnittstelle	I	V	I	I
Android-App	M	M	V	M
Algorithmus	I	I	I	V
Datenbank	V	I	I	I

Tabelle 8: IVM-Matrix

V Verantwortlich

M Mitarbeiter

I Information

11.3 Use Case Diagramm

Gesamtes Use-Case-Diagramm

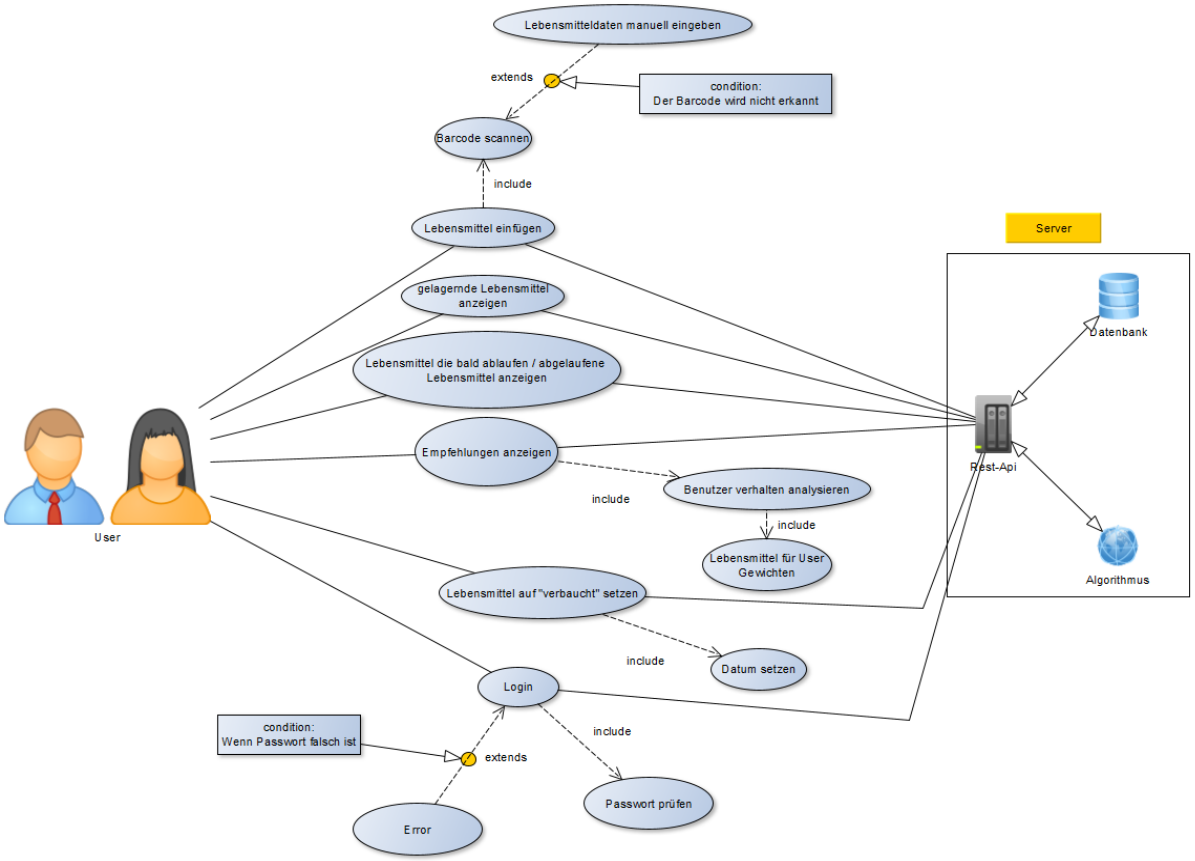


Abbildung 22: Gesamtes Use-Case-Diagramm

In der obigen Grafik sieht man alle User-Cases, die der Benutzer zur Verfügung hat. Die untenstehenden Grafiken zeigen die Use-Cases genauer.

Anmelden

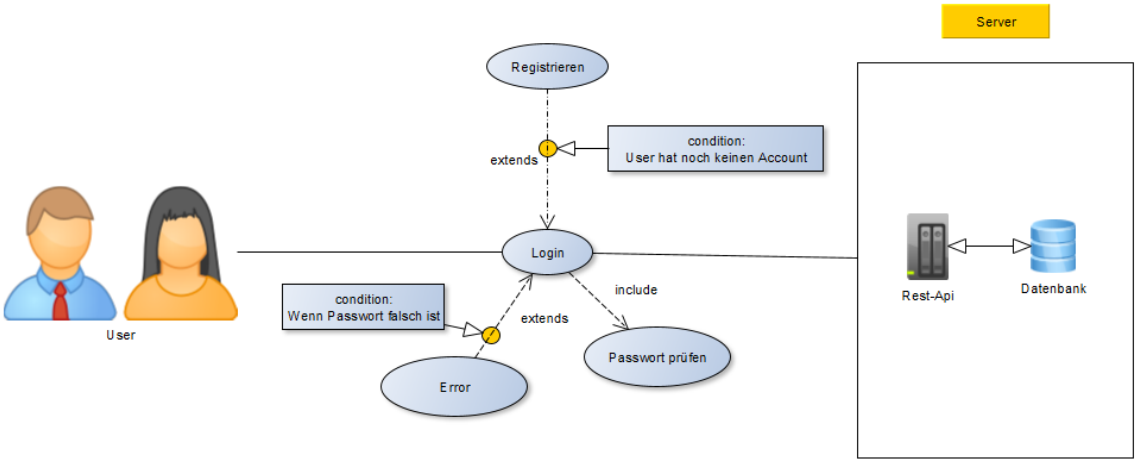


Abbildung 23: Use-Case Diagramm Anmeldung

In der oberen Grafik wird der Use-Case „Anmelden“ abgebildet. Damit der User die App benutzen kann, benötigt dieser einen Account. Falls der User noch keinen Account hat, muss er sich registrieren. Hat er bereits einen Account, muss er sich mit seinen Credentials anmelden. Sind diese falsch, wird dem User eine Fehlermeldung präsentiert. Die Kommunikation erfolgt hierbei von der App, die vom User benutzt wird, zur Rest- API. Diese Rest-Schnittstelle kommuniziert wiederum mit der Datenbank, die die Information für jeden Benutzer speichert. Die REST-API und Datenbank werden vom Server zur Verfügung gestellt.

Lebensmittel in den Haushalt hinzufügen

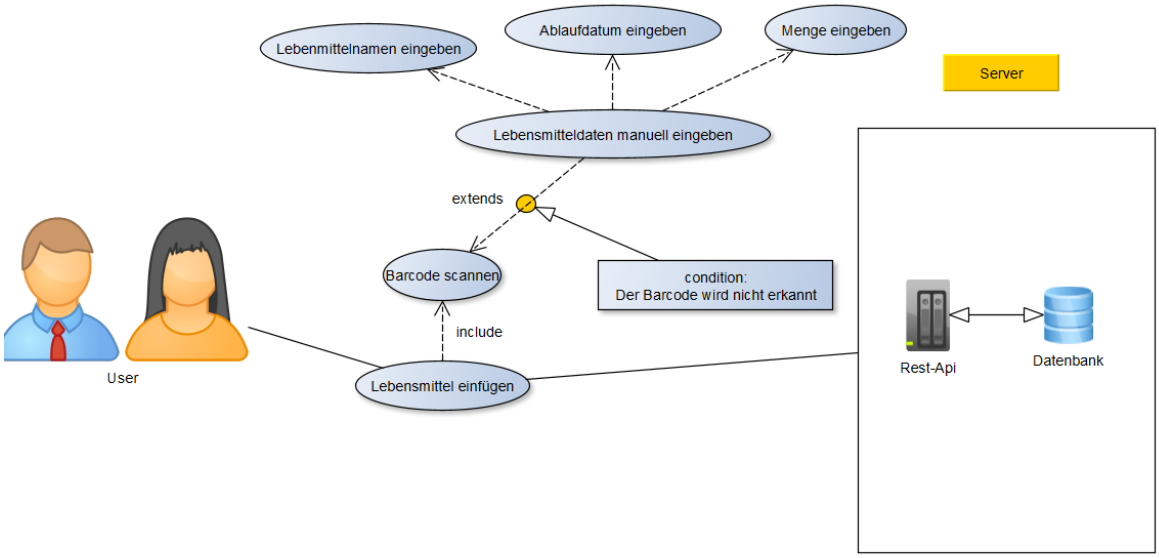


Abbildung 24: Use-Case Lebensmittel einfügen

Dieser User-Case veranschaulicht, wie das Einfügen von neuen Lebensmitteln funktioniert. Das Einfügen beinhaltet einen Barcode-Scanner, der aktiviert wird, wenn der User diese Aktion startet. Wird der Barcode nicht vom System erkannt, hat der User die Möglichkeit, das Produkt händisch einzutragen. Dazu muss dieser den Namen des Produktes eingeben. Danach wird das Ablaufdatum des spezifischen Produktes verlangt und schlussendlich noch die vorliegende Menge. Der User kommuniziert während dieses Prozesses mittels Android-App über die Rest-Schnittstelle mit der Datenbank, in der die Daten zu dem Lebensmittel gespeichert sind.

Empfehlungen für Produkte

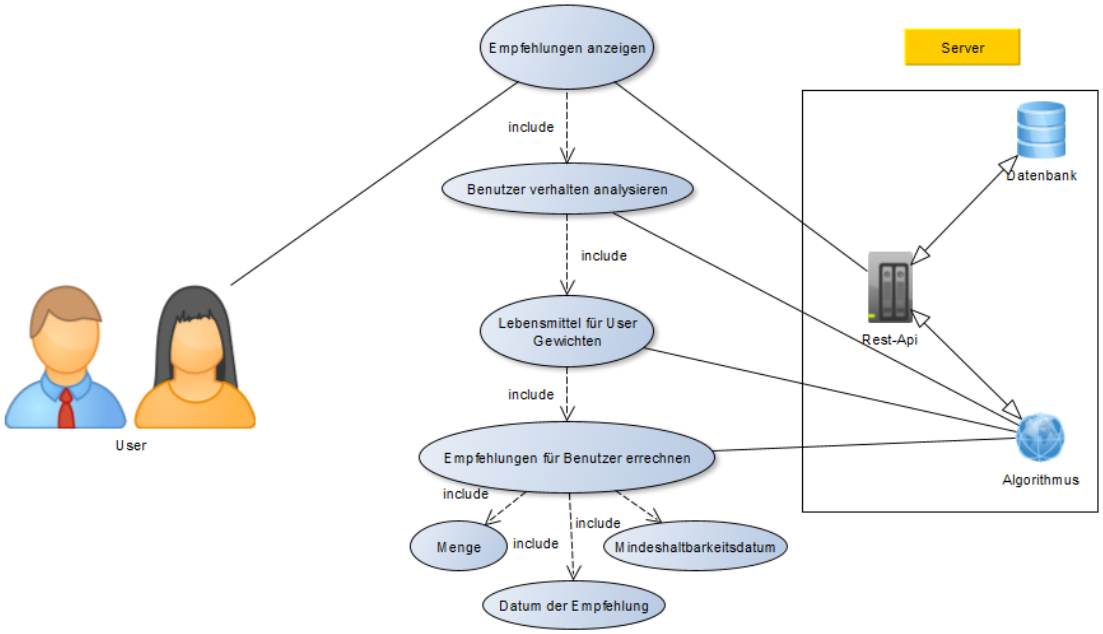


Abbildung 25: Use-Case Empfehlungen anzeigen

Dieser Use-Case bildet den einzelnen Schritt ab, die nötig sind damit der User sich seine individuellen Empfehlungen für verschiedene Lebensmittel anzeigen lassen kann. Um Empfehlungen anzeigen zu können, muss das Benutzerverhalten zuallererst analysiert werden. Dies wird vom Algorithmus übernommen, der mittels REST-API Zugriff auf die Benutzerdaten aus der Datenbank hat. Hierbei werden Kennzahlen errechnet, die im Kapitel 12.2.2 genauer erklärt werden. Als Nächstes wird für jedes Lebensmittel, das der Benutzer je besessen hat oder besitzt, Gewichtungen errechnet. Die Gewichtungen werden im Kapitel 12.2.4 genauer erläutert. Aufgrund der Gewichtung lässt sich nun eine Empfehlung treffen. Die Empfehlung besteht aus dem Namen des Produktes, Menge und Mindesthaltbarkeitsdatum. Sind diese durchgeführt worden, kann der Benutzer sich seine Empfehlungen für den heutigen Tag über die Android-App anzeigen lassen. Diese ruft wieder die Daten über die REST-API von der Datenbank ab. Die Empfehlungen werden im Kapitel 12.2.6 ausführlich erklärt.

Lebensmittel verbrauchen

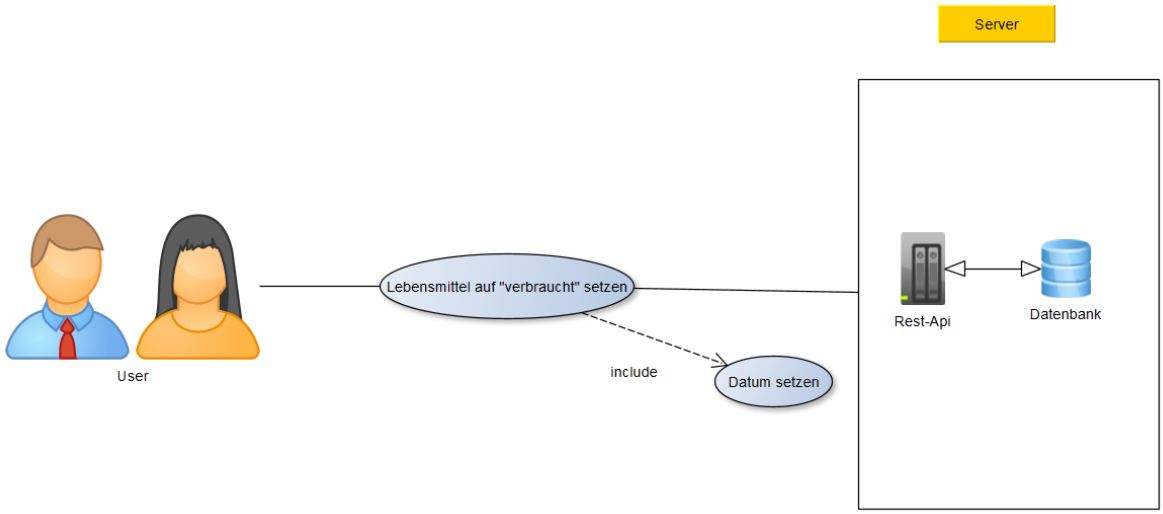


Abbildung 26: Use-Case Lebensmittel verbrauchen

Hat der User ein Lebensmittel verbraucht, gibt er dem System mittels eines Buttons bescheid. Im Hintergrund wird das Datum abgespeichert, an dessen das Produkt verbraucht wurde. Dieser Mechanismus ist für den Algorithmus geplant worden, um aus den Verbrauchsdatum Informationen ziehen zu können

Restliche Use-Cases

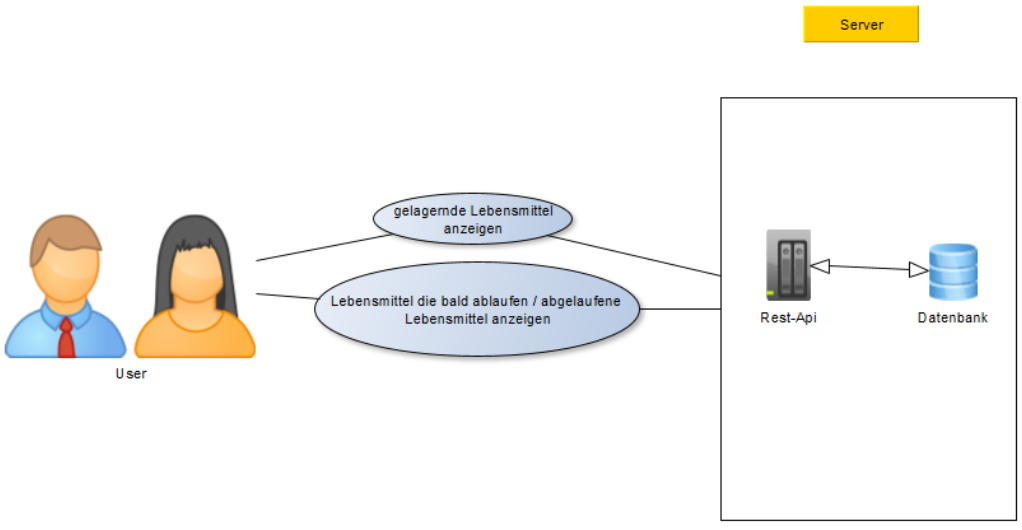


Abbildung 27: Restliche Use-Cases

In der obigen Grafik werden die zwei letzten Use-Cases veranschaulicht, die noch nicht näher beleuchtet wurden.

11.4 Design der Android-App

Abbildung 28: App-Icon

App – Icon

Für unsere App brauchten wir ein App – Icon, dieses mussten wir selbst erstellen. Hierfür haben wir unserer Logo verwendet und haben es, wie auf Abbildung 28: App-Icon, auf den Buchstaben „S“ reduziert.

Asset Studio

Mit diesem integrierten Tool in Android Studio ist es möglich, mit eigenen Bildern App-Symbole zu erstellen. Das Tool geniert einen Satz von Symbolen, welche alle die benötigten Auflösungen haben, um ein Icon hinzuzufügen zu können. (vgl. Google Developers, 2023)

11.4.1 Login-Seite

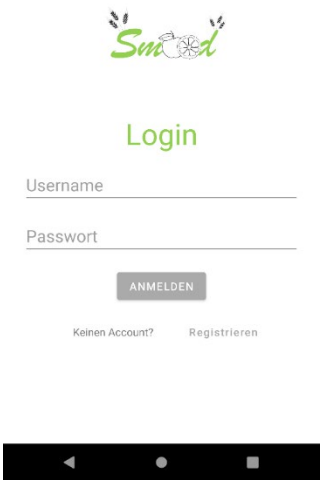


Abbildung 29: Login-Seite

Wenn man die App zum ersten Mal startet, landet man auf der „Login-Seite“, siehe Abbildung 29: Login-Seite. Auf dieser Seite wird man aufgefordert sich mit seinem „Username“ und „Passwort“ anzumelden, oder sich als neuer Nutzer zu registrieren. Nach dem Anmelden/Registrieren kommt man auf die „Home-Seite“, welchen mit den Daten des angemeldeten Nutzers befüllt sind.

Login

Wenn man bereits einen „Usernamen“ und ein „Passwort“ für diese App besitzt, kann man diese in die angegeben Felder schreiben. Im oberen Eingabefeld gehört der „Username“ und im unteren das „Passwort“. Nach dem Befüllen der Felder kann man auf den Button mit der Beschriftung „Anmelden“ drücken und kommt dann, wie oben beschrieben, auf die „Home-Seite“. Sollte ein Tippfehler passieren oder die Nutzerdaten wurden noch nicht erstellt, erscheint ein Feld mit der Inschrift „Falsche Anmeldedaten“.

Registrieren

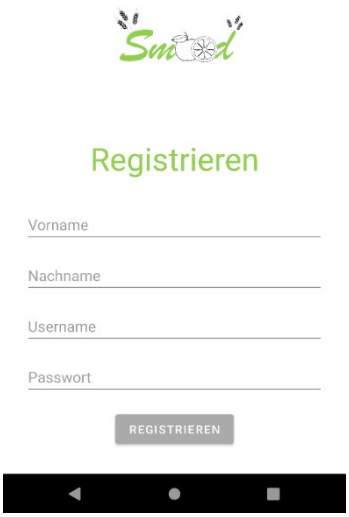


Abbildung 30: Registrierungs-Seite

Wenn noch kein Benutzer erstellt wurde, kann ein Neuer erstellt werden. Man klickt auf den Schriftzug „Registrieren“ rechts neben „Keinen Account?“. Durch diesen Klick wird man auf die „Registrierungs-Seite“ weitergeleitet, siehe Abbildung 30: Registrierungs-Seite. Auf dieser Seite muss man seine Daten für den neuen Benutzer in die vorgegebenen Feldern eintragen. In der ersten Texteingabe trägt meinen seinen „Vornamen“ und im zweiten seinen „Nachnamen“ ein. Im nächsten Feld trägt man den „Username“ und im letzten Feld das gewünschte „Passwort“, für den Account, ein. Wenn alle Felder korrekt ausgefüllt sind, muss man auf den unteren Button mit der Beschriftung „Registrieren“ klicken. So hat man einen Account erstellt, wird gleich angemeldet und wie oben beschrieben, auf die „Home-Seite“ weitergeleitet.

11.4.2 Abgelaufene – Empfohlene Lebensmittel Seite

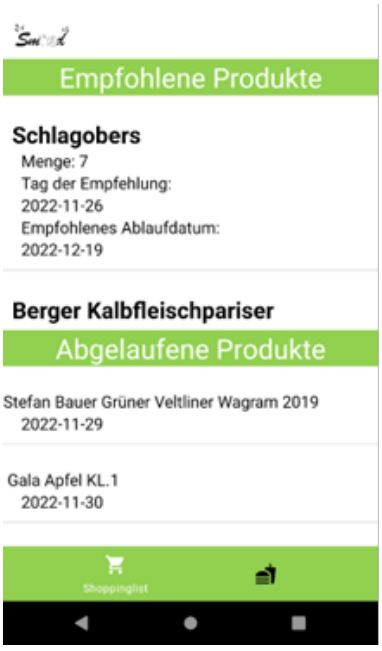


Abbildung 31: Abgelaufene - Empfohlene Lebensmittel Seite

Die zweite Seite, die man mittels der Navigationsleiste erreichen kann, ist die Shoppinglist-Seite. Auf dieser werden dem Nutzer Produkte angezeigt, die vom Algorithmus für die nächsten Einkäufe empfohlen werden. Weiters werden dem Nutzer auch Produkte angezeigt, die in den nächsten Tagen ablaufen werden oder schon abgelaufen sind.

Abgelaufene-Produkte

Die Daten der abgelaufenen Produkte werden in der unteren Liste mit der Überschrift „Abgelaufene Produkte“ angezeigt.

Alle abgelaufen Produkte können ebenfalls in einer detaillierteren Ansicht angezeigt werden. Das Design ist ähnlich der Lebensmittel-Details-Seite

11.4.3 Home

Die Home-Ansicht ist die Hauptzentrale der App. Von dieser Seite aus können die Lebensmittel verwaltet werden. Hier ist es möglich, Produkte mittels Barcodes oder durch einfache Suche hinzuzufügen. Weiters werden hier alle hinzugefügten Produkte anzeigt und auch in Kategorien unterteilt, so dass diese geordneter sind. Schlussendlich kann sich der Nutzer auch Abmelden, dies kann man, wenn man auf den „Abmelden“ Button rechts oben drückt, siehe Abbildung 32: Home-Seite (Produktkategorien).

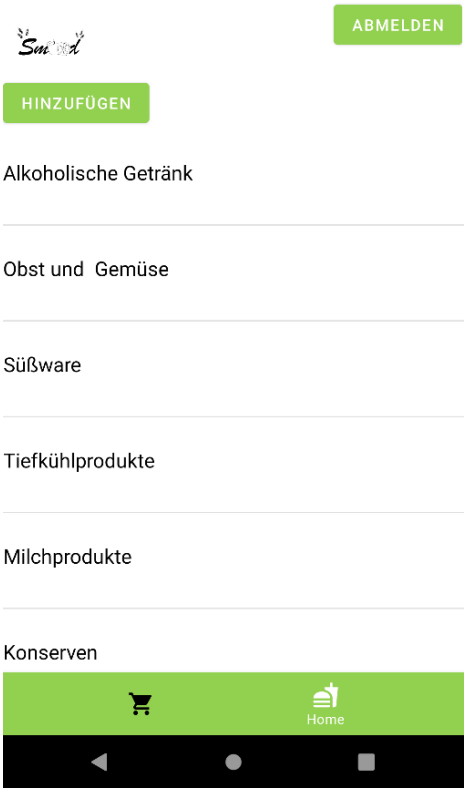


Abbildung 32: Home-Seite (Produktkategorien)

Lebensmittelkategorien

Auf dieser Seite werden alle möglichen Kategorien angeführt, bei denen der Benutzer Produkte hinzugefügt hat.

Wie oben schon erwähnt, zeigt die Seite alle Kategorien an, die zu den hinzugefügten Produkten gehören. Wenn ein Produkt entfernt wird und kein weiteres Produkt in der Kategorie enthalten ist, wird diese auch nicht mehr angezeigt. Kategorien werden auch nicht mehrmals bei mehreren Produkten der gleichen Kategorie angeführt. Alle Produkte, die zu dieser Kategorie gehören, werden durch Klicken auf das Kategorie-Feld, auf der „Lebensmittelansicht“ dargestellt.

Links oben befindet sich der „Hinzufügen“ Button, mit dem man auf die Ansicht navigieren kann, auf der ein Produkt gesucht werden kann.

Abbildung 33: Home-Seite (Produkte)

Lebensmittelansicht

Auf der „Lebensmittelansicht“ angekommen, werden dem Benutzer alle Produkte angezeigt, die man hinzugefügt hat und die zu einer spezifischen Kategorie gehören, siehe Beispiel Abbildung: Home-Seite (Produkte).

In der linken oberen Ecke der Abbildung: Home-Seite (Produkte) gibt es wie bei den Lebensmittelkategorien einen Button zum Hinzufügen von Produkten. Diese hat auch die gleichen Funktionen.

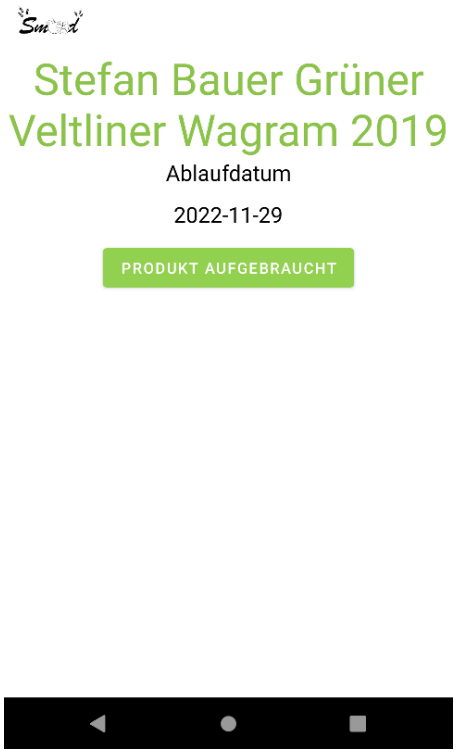


Abbildung 34: Detailansicht Produkte

Lebensmittel-Details

Diese Seite wird auch bei den abgelaufenen Produkten verwendet, um Details von diesen anzuzeigen (Ablaufdatum, Lagerort, Namen, …)

Oben auf der Seite steht der Name des Produktes. Darunter befindet sich das Ablaufdatum des jeweiligen Produktes. Weiter unten befindet sich ein Button, durch diesen wird das Produkt auf „aufgebraucht“ gesetzt.

Das Design der Detailansicht wurde so gewählt, dass alle wichtigen Daten, wie Name und Ablaufdatum zentral und schnell auffindbar angezeigt werden. Der Button, um ein Produkt auf „aufgebraucht“ zu setzten, ist darunter platziert.

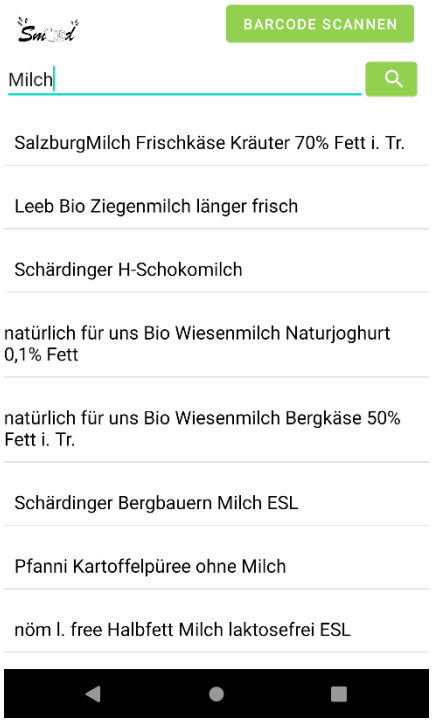


Abbildung 35: Produktsuche

Produktsuche

Wie auf der Abbildung 35: Produktsuche gut zu erkennen ist, geht es um die Produktauswahl mittels Textsuche. Rechts oben ist ein Button, der es dem Benutzer ermöglicht, seine gekauften Produkte mittels Barcodes einzuscannen, um das Produkt nicht manuell heraussuchen zu müssen. Die Funktionsweise des Barcodes wird im Kapitel Barcode weiter ausgeführt. Es können nur Produkte eingescannt werden, von welchen der Barcode in der Datenbank hinterlegt ist. Wenn das nicht der Fall ist, muss das Produkt mittels Texteingabe gesucht werden.

Damit eine Suche vorgenommen werden kann, muss ein Text in das Eingabefeld geschrieben werden, wie im Beispiel, in der Abbildung 35: Produktsuche, ersichtlich ist. Nach der Eingabe wird keine automatische Suche vorgenommen, da die Suchbegriffe an die REST-API geschickt werden, sonst würde bei einer Echtzeitsuche die REST-API mit Anfragen überfüllt werden.

Ablaufdatum definieren

Auf dieser Seite kann man für ein neues Produkt sein Ablaufdatum definieren.

In der Abbildung 36: Ablaufdatum definieren, kann man den Namen des ausgewählten Produktes sehen, für das ein Ablaufdatum festgelegt werden soll.

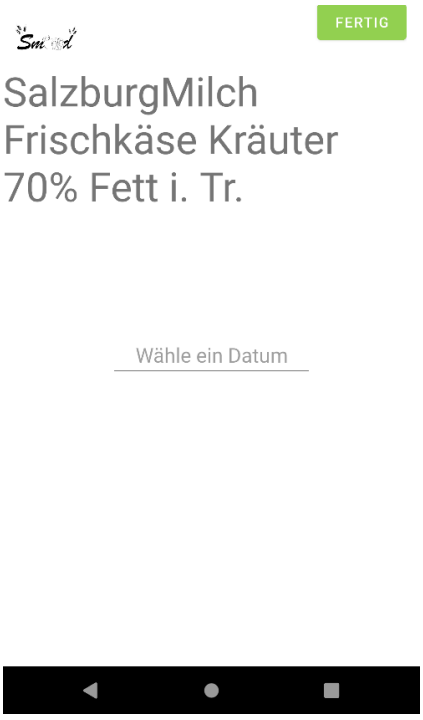


Abbildung 36: Ablaufdatum definieren

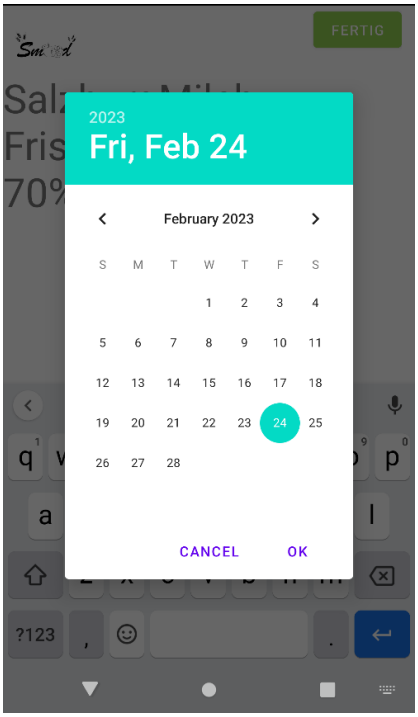


Abbildung 37: Ablaufdatum definieren (Date Picker offen)

In der Mitte des Bildschirmes befindet sich das Feld, indem das Ablaufdatum einzugeben ist. Sobald man dann auf den Button oben rechts klickt, werden die Daten an die REST-API geschickt.

12 Implementierung

12.1 Backend

Das Backend unseres Systems basiert auf der robusten und zuverlässigen Ubuntu Server-Plattform und läuft in einer virtuellen Maschine im Netzwerk der HTL Perg, um sicherzustellen, dass es leistungsfähig und stabil bleibt. Auf dem Ubuntu Server sind MySQL und Apache Tomcat installiert, um die Datenhaltung und den Webserver bereitzustellen. Die API, die von unserem System genutzt wird, wird als REST-Service über die Webserverumgebung von Apache Tomcat bereitgestellt und bezieht die notwendigen Daten aus der MySQL-Datenbank. Das Datenmodell und die API werden im Folgenden genauer beschrieben, um ein besseres Verständnis für die Funktionsweise unseres Systems zu ermöglichen.

Das Datenmodell, die API und der Algorithmus werden in diesem Kapitel beschrieben.

12.1.1 Server

12.1.1.1 Allgemeine Beschreibung

In dieser Diplomarbeit wird der Ubuntu Server-Version 22.04 LTS verwendet, da Ubuntu als Serverumgebung sehr viele Vorteile hat. Einige Vorteile sind, dass Ubuntu kostenlos und Open Source ist. Weiters ist Ubuntu sehr stabil und zuverlässig, was bedeutet, dass regelmäßige Sicherheitsupdates gemacht werden. Außerdem bietet Ubuntu viele Tools und Anwendungen, die für Server-Umgebungen entwickelt wurden. Ubuntu ist auch sehr leicht zu installieren und zu verwenden.
Was sehr essenziell für die Auswahl des Server-Betriebssystems ist, war die Anwendung von der Datenbank mit MySQL. Denn das Installieren und Einrichten dieser Datenbank geht mit einem Linux System sehr schnell und meist ohne Probleme. (vgl. www.vautron.de, 2023)

Um auf den Server zugreifen zu können, wurde eine ein Programm, Bitvise SSH, verwendet.

Webserver

Was ist ein Webserver und warum wird in dieser Diplomarbeit ein Webserver verwendet?

Ein Webserver ist ein Computer, der dazu konfiguriert ist, Webseiten und andere Web-Ressourcen über das Internet bereitzustellen. Wenn ein Benutzer eine Webseite über einen Browser aufruft, sendet der Browser eine Anfrage an den Webserver, der die gewünschte Webseite liefert, indem er sie an den Browser zurücksendet. Der Webserver kann auch andere Funktionen ausführen, wie das Verarbeiten von Formularen, das Ausführen von Skripten und die Authentifizierung von Benutzern. Es gibt verschiedene Arten von Webservern, wie z.B. Apache, IIS und Nginx, die alle unterschiedlichen Funktionen und Eigenschaften haben. Webserver sind ein wichtiger Bestandteil des Internets und ermöglichen es den Benutzern, auf Webseiten und andere Ressourcen zuzugreifen. (vgl. ionos.at, 2023)

In dieser Diplomarbeit wird ein Webserver verwendet, weil dieser eine stabile Plattform bietet, um Java Rest-Anwendungen auszuführen und bereitzustellen. Weiters kann der Webserver die Skalierbarkeit und die Leistung verbessern, indem er mehrere Anfragen gleichzeitig bearbeiten kann. Außerdem bietet er Sicherheitsfunktionen wie SSL/TLS-Verschlüsselung, um die Integrität und den Schutz von Daten sicherzustellen.

12.1.2 Datenbank

12.1.2.1 Allgemeine Beschreibung

In dieser Diplomarbeit wird die Datenbank MySQL verwendet, da diese Datenbank einige Vorteile hat. Mit dieser Datenbank kann man einfach interagieren. Weiters wurde mit dieser Datenbank schon vorher gearbeitet. Außerdem ist es sehr leicht, die Daten, die unser Team bekommen hat, zu importieren, da das Programm MySQLWorkbench verwendet wird.

Warum benötigt diese Diplomarbeit eine Datenbank?

Es gibt verschiedene Gründe, warum diese Diplomarbeit eine Datenbank benötigt. Ein Grund dafür ist, dass die Diplomarbeit eine große Menge an Daten verarbeiten muss, die organisiert und verwaltet werden. Diese Datenbank wird auch verwendet, um die Daten zu speichern und zu verwalten, um sicherzustellen, dass sie schnell abgerufen, aktualisiert und manipuliert werden können. Sie wird auch verwendet, um die Integrität der Daten zu gewährleisten, indem sie sicherstellt, dass die Daten korrekt und vollständig sind.

Woher kommen die (Lebensmittel-)Daten?

Das Diplomarbeitsteam hat sich dafür entschieden, eine Firma zu kontaktieren, um die benötigten Lebensmitteldaten zu erhalten. Vor dem Sammeln der Daten war es wichtig, sicherzustellen, dass das Team die Erlaubnis hatte, die Daten zu verwenden. Dies bedeutete, dass wir eine Einverständniserklärung von der Firma einholen mussten. Zusätzlich musste sichergestellt werden, dass alle personenbezogenen Daten anonymisiert werden. Insgesamt war es wichtig, ethisch und rechtlich korrekt vorzugehen, während die Lebensmitteldaten für die Diplomarbeit verwendet werden.

Da die Daten, die für diese Diplomarbeit benötigt wurden, in einem riesigen Excel-File geliefert wurden, das unformatiert und unsortiert war, musste viel Zeit investiert werden, um die Daten auszusortieren und manuell zu formatieren, bevor sie verwendet werden konnten. Dies war ein mühsamer Prozess, der viel Zeit in Anspruch nahm, aber letztendlich war es notwendig, um sicherzustellen, dass die Daten korrekt in die Datenbank eingespeichert werden.

12.1.2.2 Datenmodell

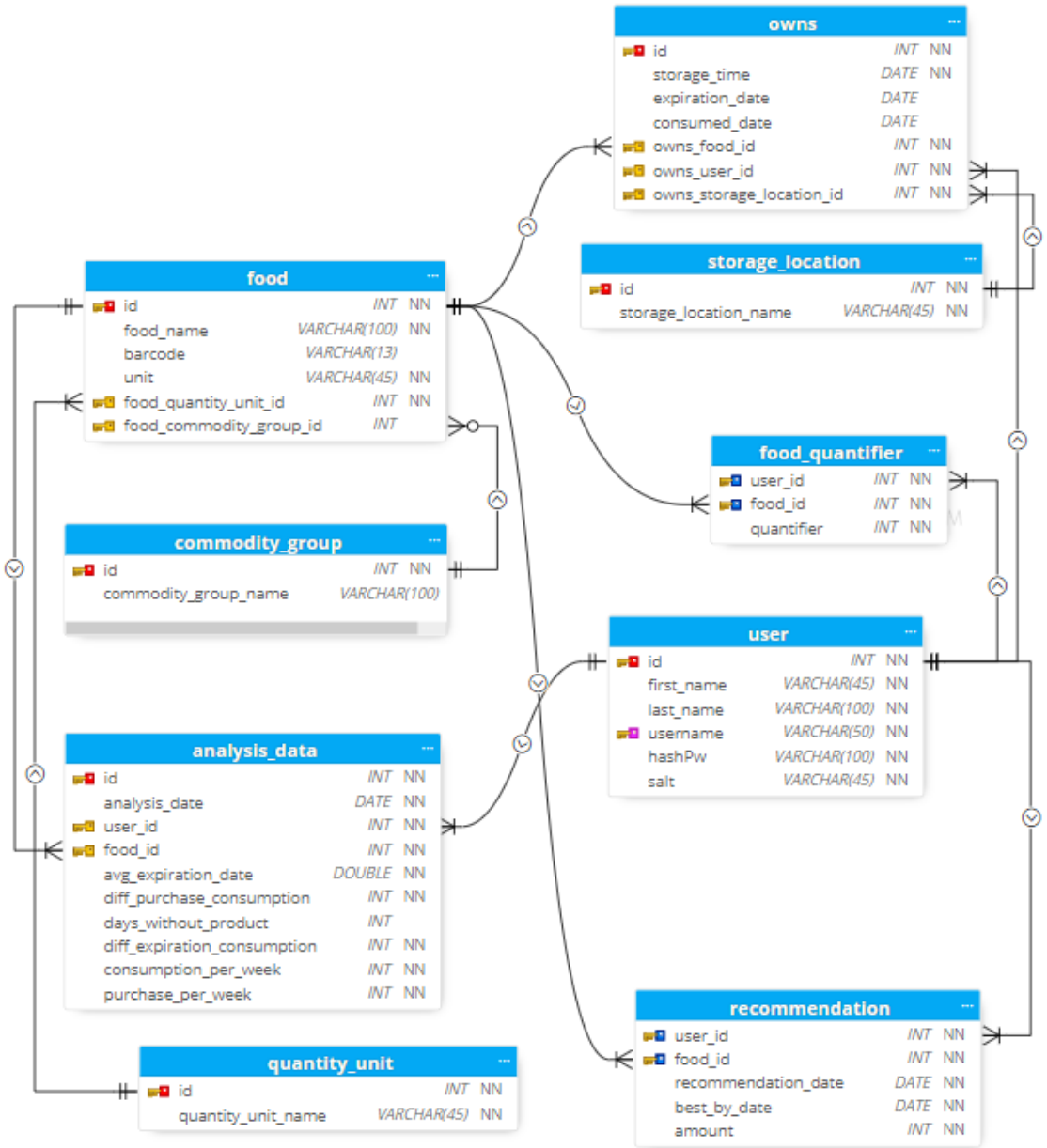


Abbildung 38: Datenmodell von Smood

12.1.2.3 Datenkatalog

Im Folgenden werden die einzelnen Tabellen mit ihrer Funktion und den zugehörigen Attributen kurz erläutert.

user

In der Tabelle user werden alle Benutzer von Smood abgespeichert. Die Tabelle kann als zentral angesehen werden und verfügt über Beziehungen zu sämtlichen anderen Tabellen.

Name	Typ	Beschreibung
id	INT	Primärschlüssel der Tabelle user. Dieser Primärschlüssel wird beim Einfügen eines neuen Datensatzes automatisch inkrementiert.
first_name	VARCHAR (45)	Vorname des Benutzers.
last_name	VARCHAR (100)	Nachname des Benutzers.
username	VARCHAR (50)	Username des Benutzers.
hash_pw	VARCHAR (100)	Der „verhashte“ Wert von Passwort und Salt des Benutzers.
salt	VARCHAR (45)	Zufällig generierter Salt zum Hashen des Passworts.

Tabelle 9: Attribute user

commodity_group

In der Tabelle commodity_group werden die Warengruppen der Lebensmittel gespeichert.

Name	Typ	Beschreibung
id	INT	Primärschlüssel der Tabelle commodity_group. Dieser Primärschlüssel wird beim Einfügen eines neuen Datensatzes automatisch inkrementiert.
commodity_group_name	VARCHAR (100)	Name der Warengruppe.

Tabelle 10: Attribute commodity_group

storage_location

Die Tabelle storage_location speichert den Lagerort eines Benutzers.

Name	Typ	Beschreibung
id	INT	Primärschlüssel der Tabelle storage_location. Dieser Primärschlüssel wird beim Einfügen eines neuen Datensatzes automatisch inkrementiert.
storage_location_name	VARCHAR (45)	Name des Lagerortes.

Tabelle 11: Attribute storage_location

food

Die Tabelle food speichert die Lebensmittel mit Barcode, die in der App verwendet werden können. Die Tabelle kann als zentral angesehen werden und verfügt über Beziehungen zu sämtlichen anderen Tabellen.

Name	Typ	Beschreibung
id	INT	Primärschlüssel der Tabelle food. Dieser Primärschlüssel wird beim Einfügen eines neuen Datensatzes automatisch inkrementiert.
food_name	VARCHAR (100)	Name des Lebensmittels.
barcode	VARCHAR (13)	Barcode des Lebensmittels.
unit	VARCHAR (45)	Mengenangabe des Lebensmittels (zum Beispiel 45).
food_quantity_unit_id	INT	Fremdschlüssel auf die Tabelle quantity_unit, die die Mengeneinheiten enthält.
food_commodity_group_id	INT	Fremdschlüssel auf die Tabelle commodity_group, die die Warengruppen enthält.

Tabelle 12: Attribute food

quantity_unit

Die Tabelle quantity_unit speichert die Mengeneinheiten der Lebensmittel.

Name	Typ	Beschreibung
id	INT	Primärschlüssel der Tabelle quantity_unit. Dieser Primärschlüssel wird beim Einfügen eines neuen Datensatzes automatisch inkrementiert.
quantity_unit_name	VARCHAR (45)	Name der Mengeneinheit.

Tabelle 13: Attribute quantity_unit

owns

Die Tabelle owns speichert die Lebensmittel, welche jeder User besitzt.

Name	Typ	Beschreibung
id	INT	Primärschlüssel der Tabelle owns. Dieser Primärschlüssel wird beim Einfügen eines neuen Datensatzes automatisch inkrementiert.
storage_time	DATE	Datum, an dem das Produkt von einem User hinzugefügt wird.
expiration_date	DATE	Ablaufdatum eines Produkts.
consumed_date	DATE	Datum, an dem das Produkt vom User konsumiert worden ist.
owns_food_id	INT	Fremdschlüssel auf die Tabelle food, die die Lebensmittel speichert.
owns_user_id	INT	Fremdschlüssel auf die Tabelle user, die die Benutzerdaten speichert.
owns_storage_location_id	INT	Fremdschlüssel auf die Tabelle stroage_location, die den Lagerort speichert.

Tabelle 14: Attribute owns

recommendation

Die Tabelle ist essenziell für den Algorithmus. Denn diese speichert die Kaufempfehlung, die berechnet werden.

Name	Typ	Beschreibung
user_id	INT	Zusammengesetzter Primärschlüssel. Dieser Schlüssel referenziert auf die Tabelle user, die die Benutzer speichert.
food_id	INT	Zusammengesetzter Primärschlüssel. Dieser Schlüssel referenziert auf die Tabelle food, die die Lebensmittel speichert.
recommendation_date	DATE	Datum, an dem die Empfehlung gegeben wird.
best_by_date	DATE	Empfohlenes Ablaufdatum eines Lebensmittels.
amount	INT	Empfohlene Menge die der Benutzer eines Lebensmittels kaufen sollte.

Tabelle 15: Attribute recommendation

food_quantifier

Die Tabelle food_quantifier speichert die Gewichtung der Lebensmittel für den einzelnen Benutzer. Die Gewichtungen benötigt der Algorithmus, um Kaufempfehlungen zu berechnen.

Name	Typ	Beschreibung
user_id	INT	Zusammengesetzter Primärschlüssel. Dieser Schlüssel referenziert auf die Tabelle user, die die Benutzer speichert.
food_id	INT	Zusammengesetzter Primärschlüssel. Dieser Schlüssel referenziert auf die Tabelle food, die die Lebensmittel speichert.
quantifier	INT	Gewichtung der Lebensmittel.

Tabelle 16: Attribute food_quantifier

analysis_data

Die Tabelle analysis_data speichert die Kennzahlen der Lebensmittel. Der Algorithmus berechnet Kennzahlen für Lebensmittel.

Name	Typ	Beschreibung
id	INT	Primärschlüssel der Tabelle analysis_data. Dieser Primärschlüssel wird beim Einfügen eines neuen Datensatzes automatisch inkrementiert.
analysis_date	DATE	Datum, an dem die Analyse durchgeführt worden ist.
user_id	INT	Fremdschlüssel auf die Tabelle user, die die Benutzerdaten speichert.
food_id	INT	Fremdschlüssel auf die Tabelle food, die die Lebensmittel speichert.
avg_expiration_date	DOUBLE	Durchschnittliches Haltbarkeitsdatum eines Lebensmittels.
diff_purchase_consumption	INT	Durchschnittliche Differenz des Einkaufsdatums und des Konsumdatums in Tagen.
days_without_product	INT	Anzahl der Tage, indem das Produkt nicht im Besitz des Benutzers war.
diff_expiration_consumption	INT	Durchschnittliche Differenz zwischen dem Ablauf und Konsumdatum in Tagen.
consumption_per_week	INT	Verbrauch eines Lebensmittels pro Woche.
purchase_per_week	INT	Einkauf eines Lebensmittels pro Woche.

Tabelle 17: Attribute analysis_data

12.1.3 REST-API

12.1.3.1 Allgemeines

Was ist eine REST-API?

REST bedeutet Representational State Transfer und API bedeutet Application Programming Interface. Mithilfe der REST-API können Server und Client über das World Wide Web miteinander kommunizieren. Eine Kommunikation zwischen Client und Server kann auch über andere Schnittstellen stattfinden, wie beispielsweise SOAP (Simple Access Object) oder GraphQL. Der Hauptgrund dafür war, dass während der Ausbildung an der HTL-Perg die REST-API bereits kenngelernt wurde. Weitere Gründe für die Entscheidung zur REST-API ist die Systemunabhängigkeit und der Fakt, dass der Code schnell und leicht in andere Programmiersprachen umgewandelt werden kann, falls dies notwendig ist.
Grundsätzlich hat die REST-API die Aufgabe, HTTP-Requests vom Client anzunehmen und weiterzuverarbeiten. Das schließt auch Datenbankoperationen, in diesem Fall durch MySQL, mit ein. Auch Daten für den Client aufzubereiten, gehört zu den Aufgaben einer REST-API. Durch die API bekommt das Frontend somit die Daten, die es benötigt.
(vgl. REST API Tutorial, 2023)

Dabei gibt es Standardoperationen, welche oft eingesetzt werden.

GET ALL

Durch den Aufruf von GET ALL können alle Einträge zu einer Entität abgefragt werden.

GET BY ID

Durch den Aufruf von GET BY ID Request können die Einträge zu einer Entität, über die ID, abgefragt werden.

POST

Mittels dem POST-Request können Daten an den Server gesendet werden. Dabei können Entitäten oder andere Daten mitgegeben werden, welche der die REST-API anschließend behandeln kann.

PUT

Durch den PUT-Request können Update Operationen ausgeführt werden, um Entitäten zu aktualisieren.

DELETE

Mittels des DELETE-Requests kann die Löschung einer Entität angefordert werden.
(vgl. Restfulapi - Http Methods, 2023)

Vorteile von REST

Einer der größten Vorteile von REST, ist die einfache und standardisierte Implementierung. Zusätzlich werden bei der Verwendung von REST-APIs, der Server und der Client voneinander getrennt. Das macht das Gesamtsystem viel übersichtlicher und einfacher skalierbar. Ein weiterer Vorteil ist die einfache Implementierung einer REST-API in eine bestehende Webtechnologie, wie Angular oder React. Zu guter Letzt, werden die Ressourcen effizienter verwendet und Anfragen schneller bearbeitet, durch die Zustandslosigkeit der REST-API. Das bedeutet, dass der Zustand zwischen Client und Server nicht gespeichert werden muss. Alle Daten, welche notwendig sind, um die Anfrage zu begreifen und auszuführen, werden in der Antwort des Servers eingebettet.
(vgl. REST API Tutorial, 2023)

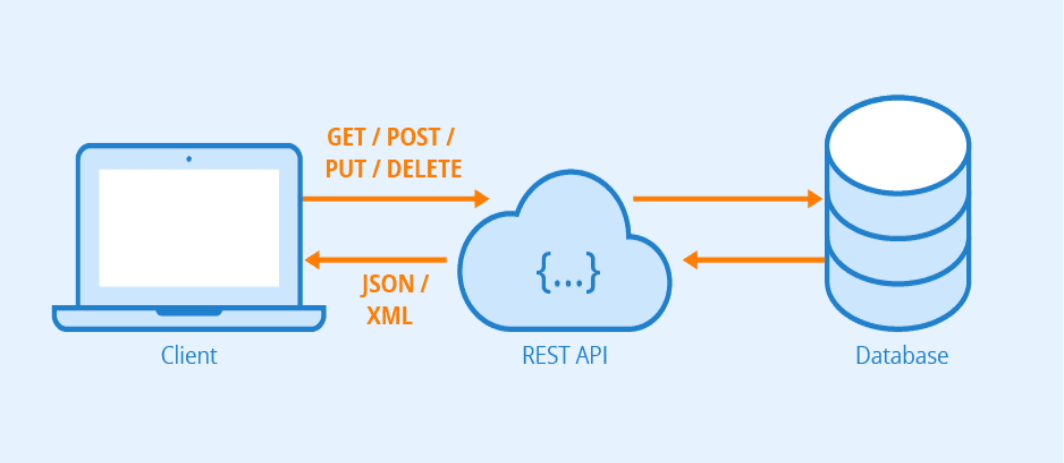


Abbildung 39: REST-API (Astera, 2023)

Wie man anhand der Grafik erkennen kann, wird zu Beginn eine Anfrage (Request) vom Client an die REST-API gesendet. Diese Anfrage enthält eine der vier Standardoperationen (GET, POST, PUT, DELETE). Dadurch weiß die REST-API, was zu tun ist. Die REST-API kommuniziert daraufhin mit der Datenbank, um die angeforderten Daten auszulesen. Nun sendet die API eine Antwort (Response), mit den gewünschten Daten, an den Client zurück. Wie man anhand der Grafik erkennen kann, wird die Antwort in dem Format JSON oder XML gesendet.

12.1.3.2 Aufbau der REST-API

Im Projekt wurden verschiedene Rest Endpoints für die unterschiedlichen Anwendungsbereiche erstellt, um die Kommunikation zwischen der App, der Datenbank und dem Algorithmus zu gewährleisten. Endpoints sind Orte, von denen aus auf die Ressourcen der REST-API zugegriffen werden kann, um Funktionen ausführen zu können. Unterteilt wurden die Methoden in Algorithmus, Food und Authentication. Die Basis-URL zum Aufruf der REST-API lautet: http://smood.htl-perg.ac.at/smood/.

Unter den Methoden des Algorithmus finden sich alle Datenbank Operationen und Kennzahlen, welche der Algorithmus zur Vorhersage benötigt.

Unter den Food Methoden finden sich alle Endpoints, welche die App zum Anzeigen, Hinzufügen und Aktualisieren der Daten benötigt.

Um die Anfragen ausführen zu können wird ein Zugriffstoken[6] benötigt. Mehr dazu kann im Kapitel Authentifizierung (Seite 82) gefunden werden.
Mittels der Authentifizierungsmethoden können die Registrierungs-, Anmelde- und Abmeldeoperationen verwirklicht werden.
Im folgenden Abschnitt können die Aufrufrouten, der einzelnen Methoden, genauer betrachtet werden.

Methoden des Algorithmus

Methode	Pfad	Übergabe	Rückgabe	Erklärung
GET	*Algorithmus/getFoodFromUser/<Username>*	Pfad: Username	Gibt eine Liste von Lebensmitteln zurück	Liefert alle aktuell im Lager befindlichen Lebensmittel zurück.
GET	*Algorithmus/getStockFromFood/<Username>/<Foodname>*	Pfad: Username, Foodname	Integer	Liefert den Lagerbestand eines Produktes, eines Benutzers.
GET	*Algorithmus/getAverageStorageAndConsumed/<Username>/<Foodname>*	Pfad: Username, Foodname	Double	Liefert zurück, wie lange ein Produkt durchschnittlich auf Lager ist.
GET	*Algorithmus/getAverageConsumptionPerWeek/<Username>/<Foodname>*	Pfad: Username, Foodname	Double	Gibt die durchschnittliche Dauer zwischen Kauf und Verbrauch eines Produkts zurück.
GET	*Algorithmus/getAveragePurchasePerWeek/<Username>/<Foodname>*	Pfad: Username, Foodname	Double	Liefert wie oft ein Produkt in einer Woche durchschnittlich eingefügt wurde.
GET	*Algorithmus/getUnconsumedProducts/<Username>/<Foodname>*	Pfad: Username, Foodname	Gibt eine Liste von noch nicht konsumierten Lebensmitteln zurück	Liefert eine Liste an noch nicht Verbrauchten Produkten zurück.
GET	*Algorithmus/getAverageConsumedExpiration/<Username>/<Foodname>*	Pfad: Username, Foodname	Double	Liefert die durchschnittliche Dauer wie lange ein Produkt vor dem Ablaufdatum verbraucht wurde.

[6] Zufällig generierte eindeutige Zeichenfolge zur Authentifizierung eines Benutzers

GET	*Algorithmus/avg_durability_length/ <Username>/<Foodname>*	Pfad: Username, Foodname	Double	Liefert die durchschnittliche Haltbarkeitslänge eines Lebensmittels des Benutzers.
POST	*Algorithmus/insertAnalysisData*	Analyse Daten im Body	Boolean	Fügt die Analyse Daten in die Datenbank ein.
POST	*Algorithmus/insertFoodQualifier*	Food Qualifier im Body Form: User ID, Food ID und den Qualifier Wert	Boolean	Fügt den Food Qualifier und die Datenbank ein.
POST	*Algorithmus/insertRecommendation*	Recommendation im Body Form: User ID, Food ID, Datum der Empfehlung, Mindesthaltbarkeitsdatum, und die Menge	Boolean	Fügt Empfehlungen in die Datenbank ein.
PUT	*Algorithmus/updateFoodQuantifier*	Food Quantifier im Body Form: User ID, Food ID und den Quantifier Wert	Boolean	Fügt den Food Quantifier in die Datenbank ein.
PUT	*Algorithmus/deleteAllRecommendation/ <username>*	Pfad: Username	Boolean	Methode zum Löschen aller bereits eingefügten Empfehlungen für einen Benutzer.

Tabelle 18: Algorithmus Methoden

Methoden Food

Methode	Pfad	Übergabe	Rückgabe	Erklärung
GET	*Food/getFood/<Username>*	Pfad: Username	Liste von Lebensmitteln	Liefert alle im Lager befindlichen Lebensmitteln des Benutzers.
GET	*Food/getSoonToExpireProducts/<Username>*	Pfad: Username	Liste von Lebensmitteln	Gibt alle Lebensmittel zurück, welche in den nächsten zwei Tagen auflaufen.
PUT	*Food/setOwnConsumed/<Username>/<Foodname>*	Pfad: Username, Foodname		Setzt das Lebensmittel des Benutzers auf verbraucht.
POST	*Food/addProduct/Name*	Body: Username, Ablaufdatum, Lebensmittelname, Lagerort		Fügt Lebensmittel des Benutzers über den Namen in die Datenbank ein.
POST	*Food/addProduct/Barcode*	Body: Username, Ablaufdatum, Barcode, Lagerort		Fügt Lebensmittel des Benutzers über den Barcode in die Datenbank ein.
GET	*Food/getCommodityGroupsFormUser/<Username>*	Pfad: Username	Liste von Warengruppen	Gibt alle Warengruppen des Benutzers zurück.
GET	*Food/searchFoodFromUser/<Username>/<Foodname>*	Pfad: Username, Lebensmittelname	Liste von Lebensmitteln	Ermöglicht die Suche nach einem Lebensmittel im Lager des Benutzers über den Namen.
GET	*Food/searchFood/Milch/<Foodname>*	Pfad: Lebensmittelname	Liste von Lebensmitteln	Ermöglicht die Suche nach einem Lebensmittel.
GET	*Food/getStorageLocationFromProduct/<Username>/<Foodname>*	Pfad: Username, Lebensmittelname	Liste an Lagerorten	Gibt die Lagerorte eines Produktes zurück.
GET	*Food/getAllLocationFromUser/<Username>*	Pfad: Username	Liste an Lagerorten	Gibt alle Lagerorte eines Benutzers zurück.
GET	*Food/getUserData/<Username>*	Pfad: Username	Daten des Users	Gibt die Daten des Benutzers zurück.
GET	*Algorithmus/getRecommendation/<Username>*	Pfad: Username	Listen von Empfehlungen	Gibt eine Liste an Empfehlungen zurück.

PUT	*Algorithmus/deleteRecommendation/ <Username> /<Foodname>*	Pfad: Username, Foodname	Boolean	Ermöglicht das Löschen von einzelnen Empfehlungen.
POST	*Food/addNewProduct*	Body: Foodname, Barcode und Warengruppe	Boolean	Ermöglicht das Hinzufügen neuer Produkte.
GET	*Algorithmus/startAlgorithmus/ <Username>*	Pfad: Username	Boolean	Zum Starten des Algorithmus für eine Benutzer.

Tabelle 19: Food Methoden

Methoden Authentifizierung

Methode	Pfad	Übergabe	Rückgabe	Erklärung
POST	*Authentication/logIn*	Body: Username, Passwort	Daten des Benutzers Form: Username, Vorname, Nachname, Token	Methode zum Einloggen in die App. Dabei wird ein Token generiert und zurückgegeben.
POST	*Authentication/registration*	Body: Vorname, Nachname, Username, Passwort	Daten des Benutzers Form: Username, Vorname, Nachname, Token	Methode zur Registrierung. Nach der Registrierung ist der Benutzer in der App angemeldet.
POST	*Authentication/logOut/<Username>*	Pfad: Username		Methode zum Ausloggen. Dabei wird der Token verworfen.

Tabelle 20: Authentifizierung Methoden

12.1.3.3 Statuscodes

Bei jedem Aufruf einer REST-API wird die Statuszeile der HTTP-Antwort genutzt, um dem Client das Ergebnis der Anfrage mitzuteilen. Standardmäßig definiert HTTP eine Menge an Statuscodes, welche den Benutzer über den Status einer Anfrage (Request) informieren. Allgemein können die Status in fünf Kategorien eingeteilt werden.

Code	Name	Bedeutung
1xx	Informaitonal	Die Anfragebearbeitung dauert noch an.
2xx	Success	Der Request wurde erfolgreich akzeptiert.
3xx	Redirection	Der Client muss zusätzliche Aktionen ausführen.
4xx	Client Error	Der Request wurde nicht durchgeführt, der Fehler liegt beim Client.
5xx	Server Error	Der Request wurde nicht durchgeführt, der Fehler liegt beim Server.

Tabelle 21: Kategorien Statuscodes

Oftmals verwendete Statuscodes dieser Diplomarbeit

Statuscode	Bedeutung
200	**OK** Der Request wurde ohne Fehler an den Server gesendet und ausgeführt.
204	**No Content** Der Request wurde erfolgreich gesendet und ausgeführt, allerdings ist keine Antwort des Servers zu erwarten.
400	**Bad Request** Der Request ist fehlerhaft.
401	**Unauthorized** Der Client sendet einen Request ohne ausreichend autorisiert zu sein. Eine Anmeldung ist notwendig.

404	**Not Found** Die API kann die Ressource zur angegebenen URL nicht finden.
405	**Method Not Allowed** Die falsche HTTP-Methode wird für den Zugriff auf die Ressource verwendet.
500	**Internal Server Error** Der Server kann die Anfrage nicht bearbeiten. Der Fehler tritt beim Server auf.

Tabelle 22: Oftmals verwendete Statuscodes

(vgl. Restful Api Statuscodes , 2023)

12.1.3.4 Kommunikation per REST-API

Damit die REST-API Anfragen entgegennehmen kann, wird ein Rest-Controller benötigt. Dieser empfängt die Anfragen, bereitet die Daten auf und übermittelt diese an den Anfragenden. Um die Anwendung überhaupt ausführen zu können wird ein öffentlicher Einstiegspunkt benötigt welcher die @SpringBootApplication-Annotation zugewiesen hat. Diese Annotation beinhaltet drei Funktionen, welche das Arbeiten mit der REST-API vereinfacht. Es umfasst einen Mechanismus, der zur automatischen Konfiguration verwendet wird. Dieser wird im Normalfall durch @EnableAutoConfiguration aktiviert. Weiters wird der Komponentenscan für Pakete der Anwendung aktiviert. Eine alternative Aktivierung besteht durch die @ComponentScan Annotation. Und zu guter Letzt wird angezeigt, dass die Klasse als Quelle für Beans[7] verwendet werden kann. Das könnte alternativ durch @Configuration ersetzt werden.
(vgl. docs.spring.io, 2023)

[7] Kapseln viele Objekte in ein einziges Objekt

```java
@SpringBootApplication
public class RestApiApplication extends SpringBootServletInitializer {

    Jana Dannhofer
    public static void main(String[] args) {
        SpringApplication.run(RestApiApplication.class, args);
    }

}
```

Abbildung 40: REST-API Application

Im folgenden Code-Ausschnitt werden beispielhaft einige Rest-Controller Methoden der Food Operationen dargestellt, welche als Rest-Endpunkte aufgerufen werden können. Dabei wird ein Rest-Controller erstellt. Durch diesen wird es ermöglicht Restful Webservices zu erzeugen und mit allen Standardoperationen (GET, POST, PUT, DELETE) umzugehen. Das erlaubt dem Benutzer, Anfragen an die API zu senden. Durch das erste, in der Graphik, ersichtliche Request Mapping können alle Methoden dieses Controllers unter dem Namen „Food“ in der URL gefunden werden. Durch das Einbinden der Komponente „FoodController“, können alle in diesem Controller befindlichen Methoden aufgerufen werden. Beispielsweise kann die Methode, um die Lebensmittel des Benutzers zu erhalten, aufgerufen werden. Im Request Mapping, beispielsweise der Methoden, wird festgelegt, durch welche der Standardoperationen (GET, POST, PUT, DELETE) und weiters, durch welche URL die Methode aufgerufen werden kann. Dabei können auch Variablen im Pfad mitgegeben werden. Wie weiter unten ersichtlich, können Variablen in der URL, mit geschwungenen Klammern und dem Namen der Variable, angehängt werden. Weiters muss die Methode diese Variable durch @Pathparam übergeben werden. Das Übergeben von Variablen im Header ist durch @RequestHeader möglich. Ganze Objekte werden im Body der Anfrage angegeben und in der Methode durch @RequestBody übergeben.

```java
@RestController
@RequestMapping("Food")
public class FoodRestController {

    13 usages
    @Autowired
    private FoodController foodController;

    Jana Dannhofer
    @RequestMapping(method = RequestMethod.GET, value = "getFood/{username}")
    public Object getFood(@PathVariable String username, @RequestHeader String token){
        return foodController.getFood(username, token);
    }

    Jana Dannhofer *
    @RequestMapping(method = RequestMethod.GET, value = "getSoonToExpireProducts/{username}")
    public List<ExpirationDateDTO> getSoonToExpireProducts(@PathVariable String username,
                                                           @RequestHeader String token){
        return foodController.getSoonToExpireProducts(username, token);
    }

    Jana Dannhofer *
    @RequestMapping(method = RequestMethod.PUT, value = "setOwnConsumed/{username}/{foodname}/{expirationDate}")
    public void setConsumed(@PathVariable String username, @PathVariable String foodname,
                            @PathVariable Date expirationDate, @RequestHeader String token){
        foodController.setConsumed(username, foodname, expirationDate, token);
    }


    Jana Dannhofer *
    @RequestMapping(method = RequestMethod.POST, value = "addProduct/Name")
    public void addProductByName(@RequestBody AddFoodNameDTO newProduct,
                                 @RequestHeader String token){
        foodController.addProductByName(newProduct, token);
    }

    Jana Dannhofer *
    @RequestMapping(method = RequestMethod.POST, value = "addProduct/Barcode")
    public void addProductByBarcode(@RequestBody AddFoodBarcodeDTO newProduct,
                                    @RequestHeader String token){
        foodController.addProductByBarcode(newProduct, token);
    }
```

Abbildung 41: Rest Controller Food

12.1.3.5 Zugriff auf die Datenbank

Die REST-API hat Zugriff auf die Datenbank und die Tabelle. Die Kommunikation geschieht über JPA, mittels des Database First Ansatzes. Hierbei existiert die Datenbank bereits und die Entitäten der Datenbank müssen nur noch im Projekt generiert werden. In Intellij IDEA ist dies durch das Einbinden der Datenbank im Fenster Datenbank möglich, wie in der Abbildung ersichtlich. Wenn die Datenbank hinzugefügt wurde, können die Tabellen, ganz einfach, durch einen Rechtsklick auf die Datenbank, mit „Generate Persistence Mapping“, erzeugt werden.

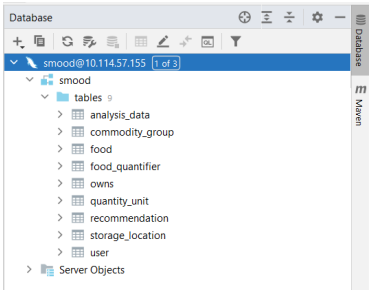


Abbildung 42: Datenbank Fenster Intellij IDEA

Im weiter unten befindlichen Code-Abschnitt ist die generierte Entität „Food“ zu betrachten.

Durch die @Entity- Annotation wird definiert, dass diese Klasse eine Tabelle einer Datenbank repräsentiert. Diese Entitäten sind POJOs[8] welche die Daten, die in einer Datenbank persistiert werden, darstellen.

Eine Klasse, welche eine Entität repräsentiert, braucht einen Primärschlüssel, dieser wird mit der Annotation @Id gekennzeichnet. Alle anderen Attribute der Entität werden mit @Column gekennzeichnet.

[8] Plain Old Java Objekts = Datenhaltungsklassen

```java
@Entity
public class Food {
    8 usages
    @Id
    @Column(name = "id", insertable = false, updatable = false)
    private Integer id;

    8 usages
    @Basic
    @Column(name = "food_name")
    private String foodName;

    8 usages
    @Basic
    @Column(name = "barcode")
    private String barcode;

    8 usages
    @Basic
    @Column(name = "unit")
    private String unit;

    8 usages
    @Basic
    @Column(name = "food_quantity_unit_id")
    private Integer foodQuantityUnitId;

    8 usages
    @Basic
    @Column(name = "food_commodity_group_id")
    private Integer foodCommodityGroupId;
```

Abbildung 43: Entity Food

Zugegriffen wird auf die Datenbank über Spring Data JPA. Durch die @Repository Annotation werden JPA-basierte Repositories eingebunden, über welche mit der Datenbank kommunizieren werden kann. Diese Repositories sind Interfaces, welche von JPARepository erben. Das Repository nimmt die Entität und ein Typargument entgegen.
In dem Repository Interfaces können für komplexere Abfragen SQL-Querys formuliert werden.

```
@Repository
public interface IFood extends JpaRepository<Food, Integer> {

    2 usages  Jana Dannhofer
    @Query(value = "select * from food where food_name = ?1", nativeQuery = true)
    List<Food> getFoodIdByName (String foodname);

    1 usage  Jana Dannhofer
    @Query(value = "select * from food where barcode = ?1", nativeQuery = true)
    Food getFoodIdByBarcode (String barcode);
```

Abbildung 44: Repository IFood

Verwendet wird das Repository in einer Spring Bean Klasse, welche mit der @Component oder @Service Annotation erzeugt wird. Die Interfaces werden durch die @Autowired Annotation eigebunden. Durch die eingebundenen Interfaces können, wie unten ersichtlich, Daten von der Datenbank abgefragt werden.
Im nachstehenden Code-Ausschnitt wird beispielhaft eine Komponenten Klasse dargestellt, welche Daten aus dem Repository abfragt.

```java
@Component
public class FoodController {
    11 usages
    @Autowired
    private IFood iFood;

    3 usages
    @Autowired
    private IUser iUser;

    1 usage   Jana Dannhofer *
    public Object getFood(String username, String token) {
        if (getUserId(username) != null && checkUserToken(username, token)) {
            return setFoodDTO(iFood.getFood(getUserId(username)));
        }
        throw new ResponseStatusException(HttpStatus.UNAUTHORIZED);
    }

    1 usage   Jana Dannhofer *
    public List<ExpirationDateDTO> getSoonToExpireProducts(String username, String token) {
        if (getUserId(username) != null && checkUserToken(username, token)) {
            return setExpirationDateDTO(username);
        }
        throw new ResponseStatusException(HttpStatus.UNAUTHORIZED);
    }
```

Abbildung 45: Component FoodController

12.1.3.6 Passwort Hashing

Um sich in die App von SMOOD einzuloggen, muss der Username mit dazugehörigem Passwort gespeichert werden. Da es sehr unsicher wäre, das Passwort unverschlüsselt in die Datenbank zu speichern, wird ein Hash mit einem dazugehörigem Salt gespeichert.

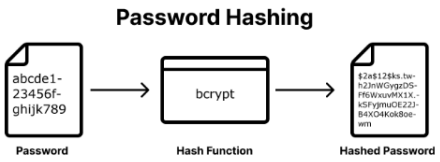


Abbildung 46: Passwort Hashing (Authgrear, 2023)

Hash

Beim Registrieren wird vom Benutzer ein Passwort eingegeben. Auf dieses eingegebene Passwort wird eine kryptographische Hashfunktion angewendet. Dabei entsteht eine neue Zeichenkette welche, an der Stelle des Klartext Passwortes, in der Datenbank gespeichert wird. Das hat den Vorteil, dass wenn die Datenbank gehackt werden sollte, das Passwort nicht einfach ausgelesen werden kann. Beim Hashing handelt es sich um eine Einwegfunktion. Die Hash Zeichenfolge kann nur mehr schwer in den Klartext umgewandelt werden. Beim Login übergibt der Benutzer wiederum sein Passwort. Mit der gleichen Funktion wird jetzt erneut ein Hashwert berechnet und mit dem Hashwert, welcher in der Datenbank gespeichert wird, verglichen. Stimmen diese beiden Zeichenfolgen überein, so gilt der Benutzer als eingeloggt und kann Abfragen tätigen.

Salt

Das Salt ist eine zusätzliche, zufällig generierte Zeichenfolge, welche an das übergebene Passwort des Benutzers angehängt wird. Dadurch wird das Passwort einzigartiger und sicherer. An dieser neuen Zeichenfolge wird dann eine kryptographische Funktion angewendet und ein Hashwert berechnet. Hashfunktionen ohne Salt erzeugen aus gleichen Passwörtern gleiche Hashes. Wenn ein Angreifer auf die Hashes Zugriff bekommt und Hashes mehrmals vorkommen, kann er darauf schließen, dass diese Benutzer das gleiche Passwort verwenden. Der Hacker kann dann eine Reihe von häufig genutzten Passwörtern ausprobieren, wie beispielsweise „1234“ oder „password“, bis dieser das Richtige gefunden hat. Das Verfahren mit Salt hat den Vorteil, dass es einen böswilligen Angriff erschwert. Dabei kann der Angreifer keine Rainbow-Tabelle[9] verwenden, um vom Hash auf ein Klartextpasswort zu schließen, da die Hash-Werte auch bei gleichen Passwörtern unterschiedlich sind.
(vgl. Supertokens, 2023)

[9] Passwort Hacking Tool, Tabelle mit häufig genutzten Passwörtern

Implementierung

Im nachstehenden Code-Ausschnitt wird eine Funktion zum Hashing von Passwörtern in der REST-API dargestellt. Mithilfe der Message-Digest-Klasse und eines SHA-256 Algorithmus wird das Passwort des Users mit dem Salt in einen Hashwert umgewandelt.
Sobald sich ein Benutzer in der App registriert, generiert das Projekt ein Salt für den Benutzer. Im nächsten Schritt wird die kryptographische Hashfunktion angewendet und das übergebene Passwort wird mit dem generierten Salt gehasht. Anschließend werden alle Daten des Benutzers, einschließlich Hash und Salt, in der Datenbank gespeichert.

```
public boolean registration (RegisterDTO register){
        String salt = generateSalt();
        String hashPw = hashPw(register.getPassword(), salt);
        if (checkUsername(register.getUsername())){
            iUser.addUser(register.getFirstname(), register.getLastname(), register.getUsername(), hashPw, salt);
            return true;
        }
    throw new ResponseStatusException(HttpStatus.UNAUTHORIZED);
}
```

Abbildung 47: Code zur Registrierung

Zum Einloggen in die App wird die „checkPassword()“ Methode verwendet. In dieser Methode wird das Salt des Benutzers aus der Datenbank ausgelesen und die Kryptographische Hashfunktion wird am übergebenen Passwort und dem Salt angewendet. Im letzten Schritt wird das eben generierte Hashpasswort, mit dem in der Datenbank gespeicherten verglichen. So wird festgestellt, ob das Passwort korrekt ist.

```
public boolean checkPassword(String username, String password) {
    int userId = getUserId(username);
    String salt = getSalt(userId);
    String hashpw = hashPw(password, salt);

    return iUser.checkUser(userId, hashpw) != null;
}
```

Abbildung 48: Prüfen des Passwortes

Das Salt wird durch die SecureRandom Klasse erzeugt. Dabei handelt es sich um eine Klasse, welche starke Zufallszahlen für die Verschlüsselung bereitstellt. Das Salt ist eine 16 Byte lange Zufallszeichenkette.

```
1 usage   Jana Dannhofer
public String generateSalt() {
    SecureRandom secureRandom = new SecureRandom();
    byte[] salt = new byte[16];
    secureRandom.nextBytes(salt);
    return String.valueOf(Base64.getEncoder().encodeToString(salt));
}

2 usages   new *
public String hashPw(String password, String salt) {
    try {
        MessageDigest md = MessageDigest.getInstance( algorithm: "SHA-256");
        md.update((password + salt).getBytes(StandardCharsets.UTF_8));
        return String.format("%040x", new BigInteger( signum: 1, md.digest()));
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    return null;
}
```

Abbildung 49: Password Hashing

12.1.3.7 Authentifizierung

Damit sich Benutzer in der App anmelden können, ist eine serverseitige Authentifizierung notwendig.

Um das zu realisieren, wird zu jedem Benutzer ein Token gespeichert. Dabei handelt es sich um eine zufällig generierte UUID. Beim Einloggen in die App, wird für den Benutzer dieser Token generiert und an diesen übergeben. Um sich Authentifizieren zu können muss der Benutzer bei jedem Request, den dieser an die REST-API sendet, den Token, den er bei der Anmeldung erhaltenen hat, im Header mit dem Key „token“ übergeben. Bei jeder Anfrage wird schließlich überprüft, ob der übergebene Token mit dem tatsächlichen Token des Benutzers übereinstimmt.

```
public Object getFood(String username, String token) {
    if (checkUserToken(username, token)) {
        return setFoodDTO(iFood.getFood(getUserId(username)));
    }
    throw new ResponseStatusException(HttpStatus.UNAUTHORIZED);
}

   public boolean checkUserToken(String username, String token) {
           try {
               String userToken = AuthenticationController.userToken.get(getUserId(username));
               return userToken.equals(token);
           } catch (Exception ex) {
               throw new ResponseStatusException(HttpStatus.UNAUTHORIZED);
           }
   }
```

Abbildung 50: Check the Token

Bei der Anmeldung wird nun ein neuer Token generiert und in eine HashMap zusammen mit der Benutzernummer gespeichert. Anschließend wird der Token mit den restlichen Benutzerinformationen, wie unten ersichtlich, zurück an den Benutzer gesendet.


```
{
    "id": 3,
    "username": "JanaD",
    "firstname": "Jana",
    "lastname": "Dannhofer",
    "token": "fbdd5e82-4dc0-40ac-8957-256551717f54"
}
```


Abbildung 51: Response bei Login

In nachstehenden Code-Ausschnitt wird beispielhaft eine Anmeldung am Server dargestellt. Dabei wird zuallererst geprüft, ob der Benutzer überhaupt schon ein Benutzerkonto erstellt hat. Wenn Ja, wird ein neuer Eintrag in die Hashmap gespeichert, um neu registrierte Benutzer in die Map hinzuzufügen. Im nächsten Schritt wird überprüft ob in der Hashmap ein Eintrag zu der Benutzernummer existiert und ob der Eintrag zur Benutzernummer leer ist, also der Benutzer nicht angemeldet ist. Sollte das zutreffen wird ein neuer Token generiert und das Passwort des Benutzers wird mit dem Salt einer Kryptographischen Hashfunktion unterzogen. Durch die Funktion „checkUser“ wird das Passwort geprüft. Im Letzten Schritt wird der Token in die Map gespeichert, um spätere Anfragen tätigen zu können.

```
1 usage   Jana Dannhofer *
public UserDataDTO logInUser(LoginDTO logInData) {
    int userId = getUserId(logInData.getUsername());

    if (checkUserExists(logInData.getUsername())) {
        userToken.put(userId, null);
    }

    if (userToken.containsKey(userId) && userToken.get(userId) == null) {
            String token = UUID.randomUUID().toString();
            String hash = hashPw(logInData.getPassword(), getSalt(userId));
            setUserDataDTO(logInData.getUsername(), token, iUser.checkUser(userId, hash));
            userToken.replace(userId, token);
    }
    throw new ResponseStatusException(HttpStatus.UNAUTHORIZED);
}
```

Abbildung 52: Login REST-API

Im unten befindlichen Bildausschnitt kann ein Beispiel für eine Übergabe des Tokens in Postman betrachtet werden. Dabei wird unter „Authorization“ der Typ API Key ausgewählt und mit dem Key „token“ wird der, beim Anmelden erhaltene Token, übergeben.

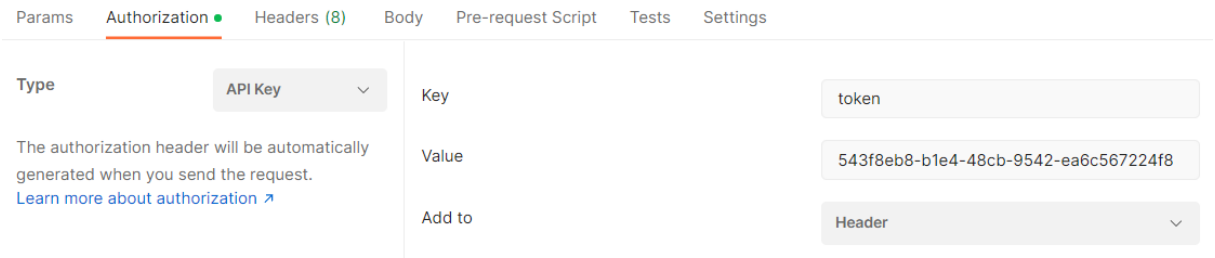


Abbildung 53: Übergabe des Tokens im Postman

12.1.3.8 Verbindung mit dem Algorithmus

Über die REST-API kann unser Algorithmus gestartet werden, um die Auswertungen der Lebensmittel und den Verbrauch dieser zu ermitteln. Um den Algorithmus starten zu können, wird ein neues Objekt der Algorithmus Klasse angelegt. Im Konstruktor werden die Benutzerdaten, wie Benutzername und Benutzernummer, übergeben. Nach dem Aufrufen, der im Codeausschnitt unten ersichtlichen Methoden im Try und Catch- Block, ist der Algorithmus gestartet. Falls ein Fehler beim Aufruf der Methoden des Algorithmus aufkommt, wird ein Fehler geworfen und dem Client wird ein Internal Server Error angezeigt.

```
public boolean startAlgorithm(String username, String token){
    Smood_Algorithm algorithm = new Smood_Algorithm( controller: this);
    algorithm._Smood_Algorithm(username, getUserId(username));
    if (checkUserToken(username, token)) {
        try {
            algorithm.analyseData();
            algorithm.insertFood_Weighting();
            algorithm.evaluate_recommendation();
            algorithm.insertRecommendations();
            algorithm.endLogFile();
        } catch (IOException e) {
            e.printStackTrace();
            throw new ResponseStatusException(HttpStatus.INTERNAL_SERVER_ERROR);
        }
        return true;
    }
    throw new ResponseStatusException(HttpStatus.UNAUTHORIZED);
}
```

Abbildung 54: Starten des Algorithmus

12.2 Algorithmus

Begriffserklärung

Der Begriff Algorithmus bedeutet, eine Vorgehensweise finden oder anwenden, um eine Problemstellung zu einem gewissen Thema zu lösen. Ein Algorithmus besteht aus einzelnen Schritten, die schlussendlich einen Output liefern. Eigenschaften eines Algorithmus sind die Terminisierung, Determiniertheit und Determinismus. Terminierung bedeutet, dass der Algorithmus nach endlichen vielen Operationen endet und ein Ergebnis zurückliefert. Die Eigenschaft Determiniertheit ist sehr essenziell bei Algorithmen. Sie besagt, dass der Algorithmus mit gleichen Daten / Eingabeparametern immer das gleiche Ergebnis errechnen kann. Somit sind die Ergebnisse miteinander vergleichbar und man kann auf diese aufbauen. Unter Determinismus ist zu verstehen, dass zu jedem Zeitpunkt der Operation, in der sich der Algorithmus zurzeit befindet, nur ein möglicher nächster Schritt besteht. Der Ablauf der einzelne Schritt ist eindeutig bestimmt. (vgl. dr-datenschutz.de, 2023)

Grundidee

Mit dieser Diplomarbeit soll Ordnung und Übersicht in den Lebensmittelhaushalt der User gebracht werden. Das Team war uns im Klaren, dass aus solcher Handy App noch mehr hinsichtlich der Nachhaltigkeit herauszuholen ist. Mithilfe einer gewissen Übersicht über die Lebensmittel und dessen Haltbarkeitsdaten kann man schon verhindern, dass weniger Lebensmittel weggeschmissen werden. Doch dem Team war bewusst, dass man bei dem Thema „Lebensmittelverschwendung“ schon um einiges früher ansetzen muss. Nämlich bereits beim Einkauf bzw. bei der Planung des nächsten Einkaufes.
Die Grundidee des Algorithmus ist daher mithilfe der Daten, die anfallen, wenn der Benutzer seine Produkte hinzufügt und wieder verbraucht, wichtige Erkenntnisse zu gewinnen und diese zu verwenden. Das Ziel ist es, von jedem Benutzer das Benutzerverhalten zu analysieren und auf Basis dessen Empfehlungen zu treffen, welche Lebensmittel er kaufen soll und in welcher Menge. Die Empfehlungen sollen so gestaltet werden, dass der Bestand der Lebensmittel, die eigentlich jederzeit zuhause sind (Mehl, Eier, Öl, …), nie auf 0 geht. Darüber hinaus ist es wichtig auf die Menge der Empfehlungen zu achten. Die Menge wird immer so gewählt, dass der User mit seinem Verbrauch die Lebensmittel bis zum Haltbarkeitsdatum konsumieren kann. Somit wird beim Algorithmus die optimale Menge evaluiert. Darüber

hinaus soll evaluiert werden, welche Lebensmittel regelmäßig konsumiert werden und nur für diese Einkaufvorschläge tätigen.
Mit diesem Algorithmus möchte man jedem Benutzer vor dem Einkauf helfen, die richtigen Lebensmittel in der optimalen Menge zu finden. Das Ziel ist es, dass weniger Lebensmittel aufgrund fehlerhafter Einschätzungen in den Müll wandern müssen.

Art des Algorithmus

Der Algorithmus fällt unter künstlicher Intelligenz. Der Begriff künstliche Intelligenz ist ein weit umgreifender Begriff, der Artificial Intelligence, Maschine Learning und Deep Learning beinhaltet. Der Algorithmus, der für diese Anwendung entwickelt wurde, fällt unter die Kategorie Artificial Intelligence.

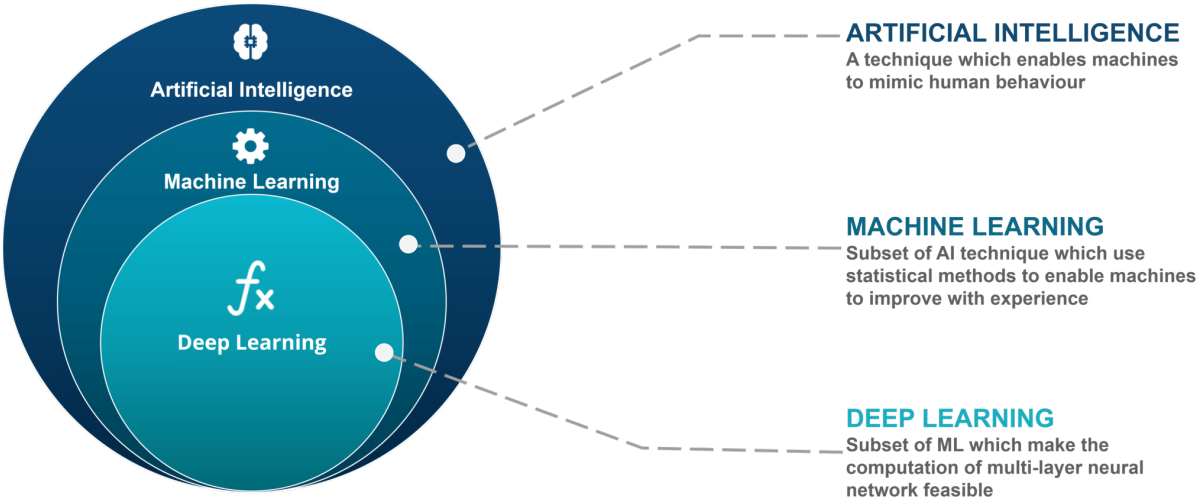


Abbildung 55: AI vs ML vs DL (vgl. www.edureka.co, 2023)

Eine künstliche Intelligenz ist dann vorhanden, wenn: „das Programm die Fähigkeit besitzt, zu lernen und zu agieren wie eine menschliche Person“ (Prof. Ing. Patrick Praher, 2023). Das bedeutet, der Algorithmus ist fähig von Daten zu lernen und mit dem Wissen, die Aktionen eines Menschen nachzubilden. Der Algorithmus ist jedoch nicht in der Lage Eigenschaften zu lernen die im Vorhinein nicht explizit programmiert worden sind. Wäre dies der Fall würde der Algorithmus in die nächste Kategorie, nämlich dem Maschine Learning, fallen.
In dieser Arbeit liegt eine AI vor. Aufgrund der Eigenschaften einer AI, wird der Algorithmus in dieser Diplomarbeit immer besser funktionieren, je mehr Daten er zu Verfügung hat.

Entwicklung der AI

Aufgrund der oben genannten Eigenschaften einer AI, wird mit folgender Entwicklung der Ergebnisse von dem Algorithmus gerechnet. Die Ergebnisse sind in hierbei die Empfehlungen.

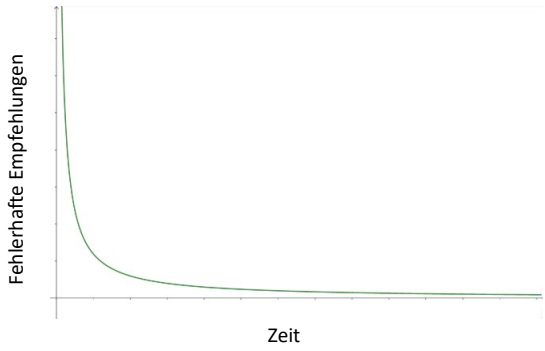


Abbildung 56: Entwicklung der AI

Es wird damit gerechnet, dass je mehr Daten der User produziert, der Algorithmus immer bessere Empfehlungen für den Benutzer errechnen kann. Am Anfang verfügt der Algorithmus noch nicht über das nötige Wissen über den User, um optimale Empfehlungen treffen zu können. Mit mehr Daten werden die Empfehlungen immer besser.

Die Entwicklung der Intelligenz lässt sich mit einer negativen Potenzfunktion bzw. einem beschränkten Wachstum beschreiben. Aus mehr Daten kann die KI immer mehr lernen und vor allem besseres Wissen generieren. In dieser Phase steigt die Intelligenz der KI exponentiell. Dieses exponentielle Wachstum wird nicht ewig anhalten. Nach einer gewissen Zeit wird die Lernkurve abflachen. Das Niveau der Empfehlung ist bereits hoch und die AI lernt nicht mehr so schnell wie am Anfang, da sie das Benutzerverhalten bereits kennt. Einzelne Veränderungen im Konsumverhalten werden aber dennoch erkannt und die Empfehlungen werden darauf angepasst. Somit wird mit einer leicht ansteigenden Lernkurve gerechnet, nach dem exponentiellen Anstieg.

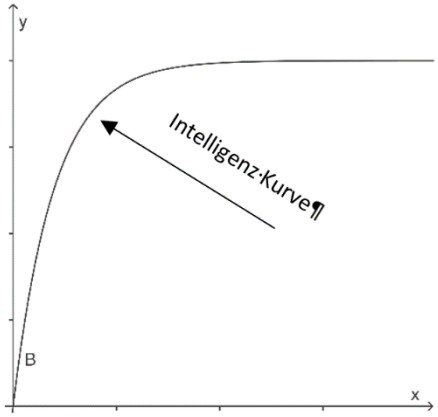


Abbildung 57: Intelligenz Kurve

Gefahren bei der Entwicklung

Wie bei jedem Artificial Intelligence Algorithmus ist die Gefahr des „Overfitting“ auch bei diesem Algorithmus gegeben. Overfitting bedeutet, den Algorithmus akkurat zu trainieren, dass aufgrund der Komplexität des Models keine vernünftigen Entscheidungen getroffen werden können. Genauer gesagt hat die AI zu viele Informationen und kann auf ähnliche Daten nicht mehr gleich reagieren. In unserem Fall könnten fehlerhafte Empfehlungen das Resultat von Overfitting sein. (vgl. Prof. Ing. Patrick Praher, 2023)

> Ein Modell ist überangepasst, wenn es so spezifisch für die ursprünglichen Daten ist, dass der Versuch, es auf in der Zukunft gesammelte Daten anzuwenden, zu problematischen oder fehlerhaften Ergebnissen und damit zu suboptimalen Entscheidungen führen würde.(datarobot.com, 2023)

Die Lernkurve einer solchen KI würde wie folgt aussehen.:

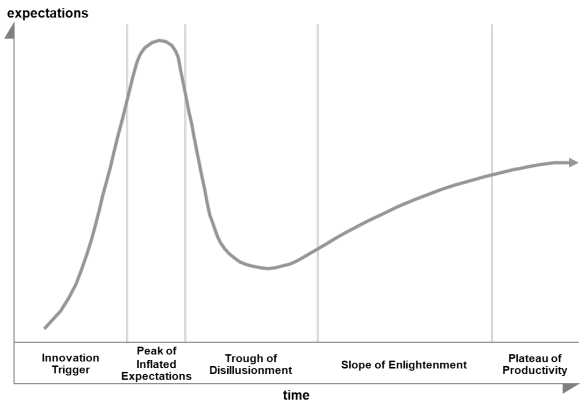


Abbildung 58: Intelligent einer Overfitting AI (datarobot.com, 2023)

Wie könnte ein Overfitting bei dieser Diplomarbeit stattfinden?

Die Gefahr bei Smood ist es, wenn der User die App über einen langen Zeitraum nutzt, fallen natürlich viele Daten zu seinen Konsumverhalten in diesem Zeitraum an. Betrachtet man bei der Entscheidung für neue Empfehlungen den gesamten Zeitraum, kann es sein, dass aufgrund der Vielzahl an Daten verzerrte Empfehlungen errechnet werden. Dies kann passieren, indem alte Daten mit Neuen vermischt werden. Die Empfehlungen würden nicht mehr das aktuelle Konsumverhalten repräsentieren, sondern das gesamte Verhalten über die Nutzung der App.

Um ein solches „Overfitting“ zu verhindern, wird eine essenzielle Maßnahme getroffen. Um das Benutzerverhalten zu analysieren, werden Kennzahlen errechnet (Kennzahlen werden im Kapitel 12.2.2 Kennzahlen behandelt). Beim Errechnen dieser Kennzahlen wird nicht die

Gesamtheit der Daten herangezogen, sondern nur die Daten die innerhalb der letzten 14 – 20 Tagen angefallen sind. Diese Daten spiegeln das tatsächlich aktuelle Konsumverhalten wider. Somit stellen wird das konstant hohe Intelligenzlevel des Algorithmus über eine lange Zeit sichergestellt.

12.2.1 Ablauf des Algorithmus

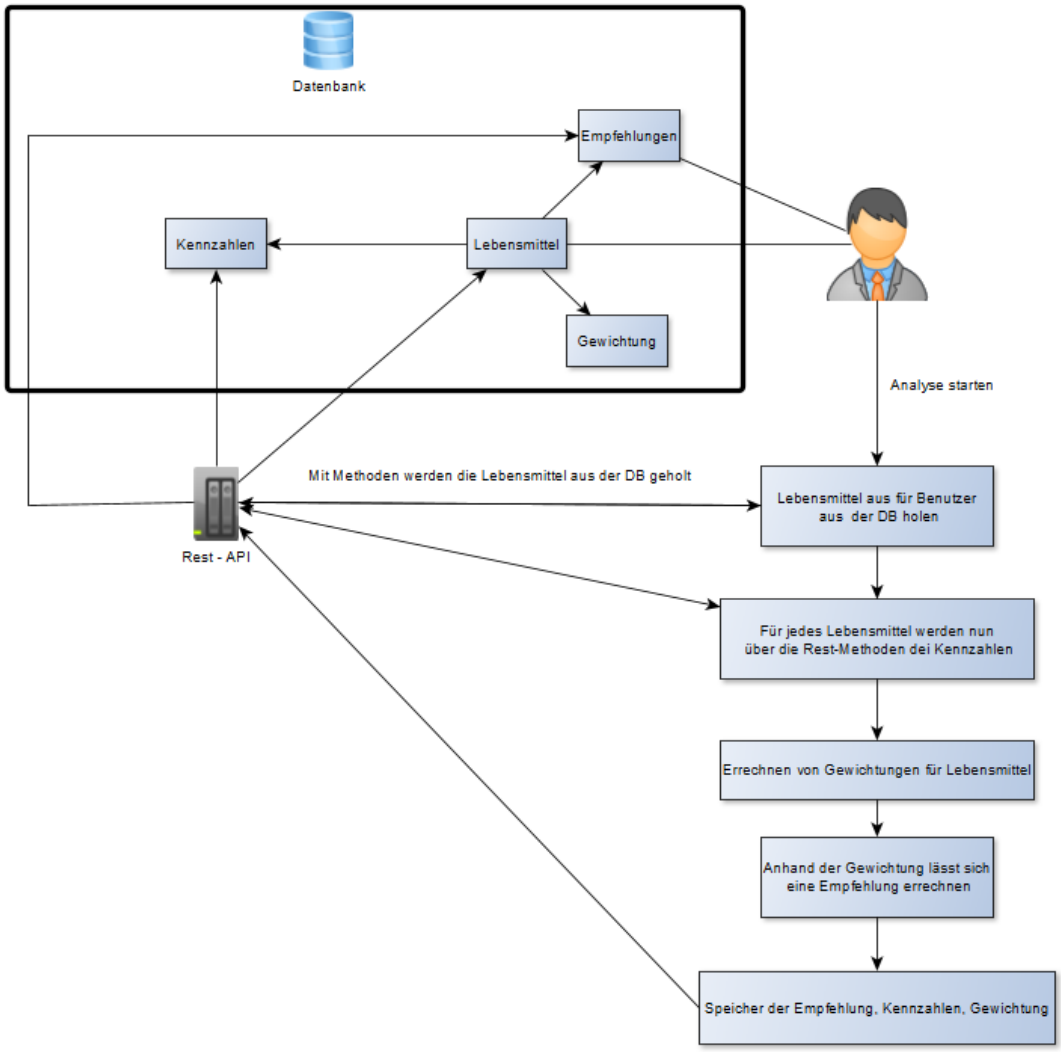


Abbildung 59: Ablauf Algorithmus

Gestartet wird der Algorithmus immer dann, wenn der Benutzer auf die Startseite der App navigiert. Hierbei muss der Benutzer keinen Button oder ähnliches betätigen. Das Aufrufen des Algorithmus wird von der App übernommen. Danach werden alle Lebensmittel, die der Benutzer bis jetzt im System eingefügt hat (auch jene die nicht mehr im Lager sind), mittels der REST-API in eine Variable geladen. Nun wird durch die Lebensmittel durchiteriert und für jedes Lebensmittel die Kennzahlen ermittelt. Das Errechnen der Kennzahlen wird von der

REST-API übernommen und diese Zahlen werden an den Algorithmus weitergegeben. Anhand dieser Kennzahlen werden vom Algorithmus für jedes Lebensmittel Gewichtungen errechnet. Die Gewichtungen und wie diese errechnet werden, findet man im Kapitel 12.2.4. Nachdem der Algorithmus für jedes Lebensmittel die Gewichtung errechnet hat, werden Empfehlungen evaluiert. Die Erklärung wie die Empfehlungen funktionieren und was diese beinhaltet wird im Kapitel 12.2.6 erklärt. Zum Schluss werden die Kennzahlen, sowie Gewichtungen und Empfehlungen an die REST-API übergeben und diese fügt die Daten in die Datenbank ein. Die Android-App kann nun die Empfehlungen des eingeloggten Users von der REST-API abfragen. In der oberen Abbildung ist der Ablauf des Algorithmus nochmals grafisch aufgearbeitet.

12.2.2 Kennzahlen

Das Benutzerverhalten wird mithilfe von zahlreichen Kennzahlen analysiert. Diese Kennzahlen errechnen sich aus den Daten, die während der Nutzung der App angefallen sind. Die Kennzahlen werden für jedes Lebensmittel, das der Benutzer jemals im Haushalt hatte, errechnet. Jedoch werden nur die Daten der letzten 14 bis 20 Tage dieser Lebensmittel herangezogen. Bedeutet, wurde ein Lebensmittel länger nicht konsumiert, haben die Kennzahlen dem Wert Null.
Das Berechnen der Kennzahlen übernimmt die REST-API, da diese einen direkten Zugriff auf die Daten hat. Diese stellt Methoden zur Verfügung, um die Kennzahlen abzufragen. Im Algorithmus werden alle Kennzahlen für ein Lebensmittel abgefragt und in einer List gespeichert. Die REST-API speichert die errechneten Kennzahlen nicht direkt in der REST-API. Die Kennzahlen werden zuerst vom Algorithmus gesammelt, danach verarbeitet und dann der REST-API zur Speicherung in der Datenbank übergeben. Jede dieser Kennzahlen bezieht sich auf ein Lebensmittel für genau einen User. Verschiedene Benutzer haben verschieden Lebensmittel und Kennzahlen für diese.

Kennzahlen:

Kennzahl	Funktionsweise
ϕ Haltbarkeitslänge	Mit dieser Kennzahl wird errechnet wie viele Tage durchschnittlich zwischen dem **Einlagerungsdatum** und dem Ende des **Haltbarkeitsdatum** liegen. Daraus kann folgendes erschlossen werden: • Wie lange ist das Produkt durchschnittlich haltbar • Welche Haltbarkeitslänge sollte eine Empfehlung für dieses Lebensmittel durchschnittlich aufweisen
ϕ Tage der Differenz des Einkaufsdatum und Verbrauchsdatum	Diese Kennzahl spiegelt die durchschnittliche Differenz zwischen Einkaufsdatum und Verbrauchsdatum wider. Vereinfacht gesagt, wie viele Tage zwischen den Einlagerungsdatum und Verbrauchsdatum durchschnittlich dazwischen sind. Hierzu werden die beide Datumswerte miteinander subtrahiert und mit den anderen Differenzen addiert. Um den

	durchschnitt zu berechnen wird diese Zahl durch die Anzahl der Menge von diesem Lebensmittel dividiert. BSP.: Zwei Jogurts Das eine Jogurt wurde am 17.12.2022 eingekauft und das Zweite am 20.12.2022. Verbraucht wurde das Erste am 19.12.2022 und das Zweite am 24.12.2022. Beim ersten Jogurt liegen drei Tage zwischen Einkaufs- und Verbrauchsdatum. Beim Zweiten liegen fünf Tage dazwischen. Diese Tage werden addiert (in unserem Fall haben wir nun 8 Tage) und durch zwei dividiert, da die Anzahl an Jogurt zwei ist. Somit bekommen wir einen durchschnittlichen Wert von 4 Tagen heraus. In einem Zeitraum von 14 bis 20 Tage könnte die Anzahl an Konsumierten Jogurt viel höher sein als in diesem Demonstrationsbeispiel. Aus diesem Durchschnittswert kann Folgendes geschlossen werden: • Konsumiert der User das Produkt eher kurz nachdem es eingekauft wurde, oder verweilt es länger im Kühlschrank • Wie lange kommt der User mit einer Einheit des Produktes aus
Tage ohne Produkt	Diese Kennzahl sagt aus, wie lange ein Lebensmittel nicht im Haushalt eingelagert war. Daraus kann geschlossen werden: • Ist dieses Produkt dauerhaft im Lager → wahrscheinlich wichtig für den Benutzer • Ist dieses Produkt über längere Zeit nicht eingelagert → der Benutzer kann auch ohne diesem Lebensmittel auskommen Diese Kennzahl macht einzeln nur beschränkt Sinn. Man kann nicht pauschal sagen, dass ein Produkt überaus wichtig

	für den User ist, nur weil das Lebensmittel dauerhaft zuhause hat. Vielleicht ist es ein Produkt, welches einfach nicht konsumiert wird oder welches eine so große Menge aufweist, dass es schwer ist das Produkt aufzubrauchen. Man muss diese Kennzahl mit anderen kombinieren, um sinnvolle Schlüsse ziehen zu können.
ϕ Tage der Differenz des Ablaufdatum und Verbrauchsdatum	Diese Kennzahl spiegelt die durchschnittliche Differenz zwischen Ablaufdatum und Verbrauchsdatum wider. Vereinfacht gesagt, wie viele Tage zwischen den Ablaufdatum und Verbrauchsdatum durchschnittlich dazwischen sind. Hierzu werden die beiden Datumswerte miteinander subtrahiert und mit den anderen Differenzen addiert. Um den Durchschnitt zu berechnen wird diese Zahl durch die Anzahl der Menge von diesem Lebensmittel dividiert. Diese Kennzahl funktioniert ähnlich wie die „ϕ Tage der Differenz des Einkaufsdatum und Verbrauchsdatum“ -Kennzahl nur mit anderen Datumswerten. Schlüsse die man aus dieser Kennzahl ziehen kann: • Wird das Produkt eher kurz vor dem Ablaufen konsumiert oder sehr viel früher. • Wird das Lebensmittel meistens nach den Ablaufdatum konsumiert.
Verbrauch pro Woche	Diese Kennzahl gibt den Verbrauch eines Lebensmittels pro Woche wieder. Hierbei wird evaluiert, wie oft das jeweilige Lebensmittel in einem Zeitraum von sieben Tagen verbraucht wurde. Diese Anzahl wird durchsieben dividiert (eine Woche = sieben Tage) und als Ergebnis bekommt man den Verbrauch pro Woche für das jeweilige Lebensmittel heraus.

	Schlüsse, die aus dieser Kennzahl gezogen werden können, sind: • Die Höhe des Verbrauches pro Woche Allein ist diese Kennzahl nur eine unbedeutende Zahl. Vergleicht man die Kennzahl mit anderen Verbräuchen pro Woche-Kennzahlen, kann gefiltert werden, ob diese Zahl ein hoher Verbrauch oder niedriger Verbrauch ist.
Einkauf pro Woche	Bei dieser Kennzahl handelt es sich um einen Indikator, der besagt, wie viel von diesem Lebensmittel in den letzten 7 Tagen eingekauft wurde. Es werden die Einkäufe addiert und durchsieben (1 Woche = sieben Tage) dividiert. Somit erhält man die Einkäufe pro Woche. Schlüsse, die aus dieser Kennzahl gezogen werden können, sind: • Wie viel wird pro Woche von diesem Lebensmittel eingekauft. Allein ist diese Kennzahl nur eine unbedeutende Zahl. Vergleicht man die Kennzahl mit anderen Einkäufen pro Woche-Kennzahlen, kann die Wertigkeit dieser Zahl herausgefiltert werden.

Tabelle 23: Kennzahlen

Wie bei manchen Kennzahlen bereits erwähnt, machen diese Zahlen allein keinen Sinn. Genauer gesagt sind alle Kennzahlen, wenn man sie einzelnen betrachtet, nur eine Zahl. Fügt man diese Kennzahlen mit anderen Kennzahlen von anderem Lebensmittel zusammen und vergleicht diese, erhält man ein klares Bild über das Benutzerverhalten des jeweiligen Benutzers. Nur wenn die Kennzahlen von einem Lebensmittel mit den anderen Kennzahlen von einem anderen Lebensmittel verglichen werden, stellt sich die Wertigkeit der Kennzahl heraus.

Genau das wird nun vom Algorithmus unternommen. Er sammelt die Kennzahlen von allen Lebensmitteln des Benutzers und fügt diese zu einem großen Bild zusammen.

12.2.3 Zugriff auf Daten

Der Algorithmus bekommt alle Daten von der REST-API. Hierbei werden vor allem die Informationen über den aktuell eingeloggten Benutzer und dessen Kennzahlen benötigt. Beim Start des Algorithmus werden all die Benutzerinformationen und dessen Kennzahlen beschafft. Somit sind die benötigten Daten nun im Cache des Algorithmus. Mit diesen Daten werden nun Operationen durchgeführt. Zum Schluss stehen Ergebnisse in Form von Empfehlungen fest. Für diese Ergebnisse stehen wiederum Rest-Methoden zur Verfügung. Diese Methoden fügen die Empfehlungen/ Endresultate in die Datenbank ein. Danach ist der Durchlauf des Algorithmus beendet.

12.2.4 Gewichtung der Lebensmittel

Die Lebensmittel des Konsumierenden werden in drei verschieden Kategorien eingeteilt. Mit diesen Kategorien kann die Beliebtheit bzw. die Wichtigkeit der Lebensmittel für den Benutzer dargestellt werden. Jedes Lebensmittel wird vom Algorithmus in eine der drei Gewichtungsstufen eingeordnet. Die drei Gewichtungskategorien sind.:

Gewichtung	Bedeutung
Gewichtung 1	Diese Gewichtung bedeutet, dass der Benutzer das jeweilige Lebensmittel nahezu immer zuhause hat. Er konsumiert dieses auch nahezu immer. Aus dieser Gewichtung kann geschlossen werden, dass der Konsument dieses Lebensmittel zwangsweise benötigt, um sich in seinem Konsum nicht einschränken zu müssen. Es kann davon ausgegangen werden, dass das Produkt jeden Tag konsumiert wird. Die Gewichtung 1 beinhalt die mit Abstand „wichtigsten“ Lebensmittel des Benutzers.
Gewichtung 3	Die Gewichtung 3 ist die Mittlere der 3 Gewichtsstufen. Fallen Lebensmittel unter die Gewichtung 3 bedeutet das, dass die Produkte regelmäßig vom Benutzer gekauft und konsumiert werden aber nicht zwangsweise im Haushalt vorhanden sein müssen.
Gewichtung 5	Lebensmittel, die mit der Gewichtung 5 gekennzeichnet sind, haben für das Konsumverhalten des Benutzers keinen Einfluss. Der Konsument verwendet dieses Produkt selten bis gar nicht.

Tabelle 24: Gewichtungen

Diese Gewichtungsstufen sind essenziell für die Empfehlungen. Auf Basies dieser drei Stufen werden Empfehlungen mit Mengenangaben errechnet. Die Ausführung der Empfehlungen befindet sich im Kapitel 12.2.6.

12.2.5 Analyse des Benutzerverhaltens

Die Analyse des Benutzerverhaltens ist ein Prozess, der aus mehreren Teilen besteht. Zuallererst beinhaltet dieser Prozess die Datenbeschaffung der Benutzerdaten und Kennzahlen. Zusätzlich wird für jede Kennzahl ein Durchschnittswert über alle Lebensmittel hinweg errechnet. Mit diesem Durchschnittswert pro Kennzahl können die einzelnen Kennzahlen der Lebensmittel verglichen werden. Hierbei lässt sich feststellen, ob eine Kennzahl über/unter/dem Durchschnittswert dieser Kennzahl liegt. Dieser Vorgang wird nun in einem Beispiel veranschaulicht.:

> Es soll das Benutzerverhalten für Milch evaluiert werden. Um diesen Vorgang zu veranschaulichen, wird sich nur auf eine Kennzahl fokussiert.
>
> Verbrauch pro Woche: 3,74
>
> Durchschnittlicher Verbrauch pro Woche: 1,52 →Dieser Wert ist der Durchschnittswert von der Kennzahl „Verbrauch pro Woche“ aller Lebensmittel, die der Algorithmus verarbeitet. (Durchschnittlich weisen alle Lebensmittel einen Verbrauch pro Woche von 1,52 auf)
>
> Diese beiden Zahlen werden nun verglichen und es kann gesagt werden, dass die Kennzahl klar über dem Durchschnitt liegt. Die Bedeutung ist klar, das Produkt ist eines der beliebteren Produkte, wenn man nur diese Kennzahl betrachtet.

Nachdem alle Daten, die benötigt werden, vorhanden sind, werden die Lebensmittel zu den Gewichtungsklassen zugeteilt. Die Kennzahlen, die für diesen Vorgang verwendet werden, sind:

- ϕ Tage der Differenz des Einkaufsdatum und Verbrauchsdatum
- ϕ Tage der Differenz des Ablaufdatum und Verbrauchsdatum
- Verbrauch pro Woche
- Einkauf pro Woche

Liegen alle Kennzahlen für ein Lebensmittel 20% unter den Durchschnittswerten, erhaltet das jeweilige Lebensmittel die Gewichtung 1. Wenn jedoch alle Kennzahlen für ein Lebensmittel 20% über die Durchschnittswerte liegen, erhaltet das jeweilige Lebensmittel die Gewichtung 5. Fällt der Vergleich inmitten dieser Spanne, wird das Lebensmittel der Gewichtung 3 zugeordnet.
Zusätzlich ist noch eine Regel implementiert. Wenn der Einkauf pro Woche oder Verbrauch pro Woche 0 ist, fällt das Produkt automatisch in die Gewichtungsstufe 5.
Die 20%ige Abweichung wurde bei einer Recherche festgestellt. Dabei stellte sich die Frage, ab wann ein Konsum von einem Produkt unterdurchschnittliche bzw. überdurchschnittlich ist. Mit den bereits gesammelten Daten wurden Berechnungen und Analysen angestellt. Pro Person wurden drei Lebensmittel herangezogen. Jedes Teammitglied hatte jeweils ein Lebensmittel, bei dem er glaubt, es falle unter die Gewichtung 1, 3 und 5. Als die Auswahl der Teammitglieder miteinander verglichen wurde, ist das Team auf die 20% gestoßen. Dieser Wert ist der „Sweet-Spot“ für eine Unterscheidung in die drei Gewichtungsstufen.

Nun sind alle Lebensmittel in eine Gewichtungsklasse zugeordnet worden.

12.2.6 Empfehlung für den Benutzer

Im letzten Akt des Algorithmus geht es darum, Empfehlungen für den Benutzer zu errechnen. Die Empfehlungen sollen dem User dabei helfen, immer die Lebensmittel zu Hause zu haben, die er aufgrund seines Benutzerverhalten benötigt. Des Weiteren sind die Empfehlungen mit der Mengenangabe so angepasst, damit der User nicht zu wenig aber auch nicht zu viel von einem Produkt einkauft. Hierbei ist der Punkt der Lebensmittelverschwendung besonders im Fokus. Wichtig für die Empfehlungen sind folgenden Daten/Kennzahlen:

- Gewichtung des Lebensmittels
- Wie viel ist von dem Lebensmittel noch zu Hause auf Lager
- Wie viele Tage ist ein neu gekauftes Produkt haltbar
- Verbrauch pro Woche
- Haltbarkeitsdaten

Empfehlung Gewichtung 1

Wenn das Lebensmittel in die Gewichtungsstufe 1 eingestuft worden ist, müssen zwei Sachen beachtet werden. Ob der Benutzer noch genug von diesem Lebensmittel auf Lager hat, oder nicht. Wenn der User noch genug Einheiten besitzt, soll eine Empfehlung mit der Menge 1 gegeben werden. Bedeutet, obwohl der Konsument noch genug von einem Lebensmittel mit der Gewichtung 1 hat, wird eine Empfehlung mit Menge eins gegeben. Der Grund dafür ist, da das Produkt in seinem Konsum überaus wichtig/beliebt ist, soll dieses Produkt dem Benutzer vor Augen geführt werden.
Wenn die Menge des Lebensmittels nicht mehr ausreicht, muss eine Menge berechnet werden, die auf das Konsumverhalten angepasst ist.
Der Ablauf einer Empfehlung mit Gewichtung 1 ist Programmiertechnisch folgendermaßen gelöst worden. Am Anfang wird eine empfohlene Menge errechnet. Dabei wird die durchschnittliche Haltbarkeitslänge mit dem Verbrauch pro Woche multipliziert.

```
int quantity = (int) (Math.ceil(key_figures.getAvg_durability_length() * key_figures.getConsumption_per_week()));
```

Abbildung 60: Einkaufmenge berechnen

Als nächstes wird ein Wert errechnet, welcher die Tage repräsentiert die zwischen dem heutigen und dem höchsten Haltbarkeitsdatum des Produktes liegen, plus die durchschnittliche Haltbarkeitslänge. Dieser Wert sagt aus, wie viele Tage höchstens mit einem Neukauf des Lebensmittels abgedeckt sind. Diese Tage beziehen sich auf die Haltbarkeit des Lebensmittels und nicht auf die Menge. Bedeutet, wie viele Tage wären abgedeckt, ohne die Halbbarkeit zu überschreiten.

```
int daysToExpire = (int) ChronoUnit.DAYS.between(LocalDate.now(), lastExpirationDate) + (int) Math.round(key_figures.getAvg_durability_length());
```

Abbildung 61: maximal mögliche Haltbarkeitstage

Danach wird in einem If-Else-Statement überprüft, ob der aktuelle Lagerbestand plus der errechneten Einkaufsmenge größer dem Ergebnis der Multiplikation von dem Verbrauch pro Woche und die maximal möglichen Tage, bezogen auf die Haltbarkeit, ist. Wenn der Lagerbestand und die Empfehlungsmenge zusammen größer wären, würde das bedeuteten, dass der User den zukünftigen Lagerbestand nicht innerhalb der Haltbarkeitsdauer konsumieren könnte. In diesem Fall wird eine Empfehlung mit der Menge 1 gegeben. Andernfalls beinhaltet die Empfehlung die errechnete Menge.

```
if (stock + quantity > key_figures.getConsumption_per_week() * daysToExpire) {
    recommendation = new Recommendation(getLockedInUserID(), f.getId(), LocalDate.now(), LocalDate.now().plusDays(Math.round(key_figures.getAvg_durability_length())),  amount: 1);
} else {
    if (quantity != 0) {
        recommendation = new Recommendation(getLockedInUserID(), f.getId(), LocalDate.now(), LocalDate.now().plusDays(Math.round(key_figures.getAvg_durability_length())), quantity);
    }
}
```

Abbildung 62: Entscheidung

Empfehlung Gewichtung 3

Diese Empfehlung ist mit Abstand die Operationsintensivste. Um eine Empfehlung zu treffen, werden alle Umstände (Lagerbestand, Haltbarkeitslängen, Verbrauch, …) betrachtet. Es wird nur eine Empfehlung getätigt, wenn diese Umstände es zulassen.

Zuallererst erst wird untersucht, ob das jeweilige Produkt noch einen Lagerbestand aufweist. Wenn nicht und der Verbrauch pro Woche über 0 ist, dann wird eine Empfehlung mit der zuvor errechneten Empfehlungsmenge angegeben.

```
if (stock == 0 && key_figures.getConsumption_per_week() > 0) {
    int quantity = (int) (Math.ceil(key_figures.getAvg_durability_length() * key_figures.getConsumption_per_week()));
    if (quantity != 0) {
        recommendation = new Recommendation(getLockedInUserID(), f.getId(), LocalDate.now(), LocalDate.now().plusDays(Math.round(key_figures.getAvg_durability_length())), quantity);
    }
```

Abbildung 63: Empfehlung 3, wenn Lagerbestand 0

Weist das Lebensmittel noch einen Lagerbestand auf, wird untersucht, ob die eingelagerten Einheiten ihr Haltbarkeitsdatum bereist überschritten haben. Ist dies nicht der Fall, wird der zukünftige Konsum des Produktes geschätzt bzw. errechnet. Ist dieser geschätzte Konsum höher als der Lagerbestand wird eine Empfehlung gegeben.

```
} else if (stock > 0) {
    if (lastExpirationDate != null) {
        if (lastExpirationDate.isAfter(LocalDate.now())) {
            int daysToLastExpireDate = (int) ChronoUnit.DAYS.between(LocalDate.now(), lastExpirationDate);
            double avg_predicted_consume = key_figures.getConsumption_per_week() * daysToLastExpireDate;
            if (avg_predicted_consume > stock) {
                int quantity = (int) (Math.ceil(key_figures.getAvg_durability_length() * key_figures.getConsumption_per_week()));
                System.out.println("Quantity: (Rec3) " + quantity);
                logger.info( msg: "Quantity: (Rec3) " + quantity);
                if (quantity != 0) {
                    recommendation = new Recommendation(getLockedInUserID(), f.getId(), LocalDate.now(), LocalDate.now().plusDays(Math.round(key_figures.getAvg_durability_length())), quantity);
                }
            }
```

Abbildung 64: Lagerbestand > 0

Ist jedoch das Haltbarkeitsdatum, des eingelagerten Produktes überschritten worden, muss evaluiert werden, ob das Konsumverhalten eine Empfehlung zulässt. Als Kriterien werden die Kennzahlen Verbrauch pro Woche und Einkauf pro Woche herangezogen. Das Lebensmittel muss in den letzten 7 Tagen mindestens 0,9-mal eingekauft und verbraucht worden sein. Dieser Wert wurde ebenfalls bei einer Analyse von bestehenden Daten evaluiert. Ist dies der Fall muss der Verbrauch pro Woche größer als der Einkauf pro Woche sein, um eine Empfehlung geben zu können. Diese Maßnahme ist notwendig, da sichergestellt wird, dass

nicht mehr eingekauft wird als verbraucht wird. Nun kann eine Empfehlung mit der zuvor errechneten Menge getroffen werden.

```
} else {

    if (key_figures.getConsumption_per_week() * 7 >= 0.9 && key_figures.getPurchase_per_Week() * 7 >= 0.9) {
        if (key_figures.getPurchase_per_Week() <= key_figures.getConsumption_per_week()) {
            int quantity = (int) (Math.ceil(key_figures.getAvg_durability_length() * key_figures.getConsumption_per_week()));
            if (quantity != 0) {
                recommendation = new Recommendation(getLockedInUserID(), f.getId(), LocalDate.now(), LocalDate.now().plusDays(Math.round(key_figures.getAvg_durability_length())), quantity);
            }
        }
```

Abbildung 65: Verbrauch > Einkauf

Nun wurden alle möglichen Fälle für eine Empfehlung aufgezeigt. Bestehen die anderen Kennzahlen ebenfalls die Überprüfungen nicht, wird keine Empfehlung für die Gewichtung 3 getroffen.

Empfehlung Gewichtung 5

Hat ein Lebensmittel die Gewichtung 5 zugeordnet bekommen, wird unter keinen Umständen eine Empfehlung getroffen.

12.2.7 Fehlerbehandlung

Wie bei jeder Software gibt es Probleme bzw. muss sie auf Probleme getestet werden. Um sicherstellen zu können, dass der Algorithmus nach den definierten Vorstellungen operiert, wurden ausgelesene Datensätze ins System eingefügt. Bei diesen Daten wussten wir genau, welche Ergebnisse resultieren sollen. Mit diesem Vorgang konnten die Operationen im Algorithmus so lange verfeinern werden, bis die erwarteten Ergebnissen auftraten. Danach haben wurden mehr Daten in das System gegeben. Auf Anhieb konnten die Resultate nicht überzeugen. Nachdem am Algorithmus Hand angelegt wurde und die Ergebnisse mit den zusätzlichen Daten stimmten, wurde ein Betabetrieb gestartet. Es wurden Daten über die App ins System gegeben. In dieser Phase des Projektes funktionierte der Algorithmus wie gewollt.

Um während der Entwicklung die Schritte des Algorithmus nachvollziehen zu können wurde auf Konsolenausgaben gesetzt. Mit dieser Weise konnte verstanden werden, was im Algorithmus gerade vor sich ging und wie dieser zu Empfehlungen gekommen ist. Um diese Nachvollziehbarkeit auch im regulären Betrieb aufweisen zu können, schreibt der Algorith-

mus seine Tätigkeiten in ein Logfile, welches am Server gespeichert ist. Tritt also ein Problem auf, könne zu jeder Zeit in diesem Logfile nachgesehen werden, um zu verstehen, was der Algorithmus gemacht hat.

```
Okt. 01, 2022 6:20:53 PM at.ac.htlperg.Smood_Algorithm analyseData
INFORMATION: User: JuliaBodingbauer
Okt. 01, 2022 6:20:54 PM at.ac.htlperg.Smood_Algorithm fillKeyFigures
INFORMATION: Name: Schlagobers
Okt. 01, 2022 6:20:54 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
INFORMATION: AVG Durability: NaN
Okt. 01, 2022 6:20:54 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
INFORMATION: AVG Storage Consumed: NaN
Okt. 01, 2022 6:20:54 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
INFORMATION: Days Product Not in Stock: 65
Okt. 01, 2022 6:20:54 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
INFORMATION: AVG Consumed Durability: NaN
Okt. 01, 2022 6:20:54 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
INFORMATION: Consumption per week: 0.0
Okt. 01, 2022 6:20:54 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
INFORMATION: Purchase per Week: 0.0
Okt. 01, 2022 6:20:55 PM at.ac.htlperg.Smood_Algorithm fillKeyFigures
INFORMATION: Name: Milch
Okt. 01, 2022 6:20:55 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
INFORMATION: AVG Durability: NaN
Okt. 01, 2022 6:20:55 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
INFORMATION: AVG Storage Consumed: NaN
Okt. 01, 2022 6:20:55 PM at.ac.htlperg.Smood_Algorithm logKey_Figures
```

Abbildung 66: Auszug aus dem Log-File

12.3 Frontend

12.3.1 Barcode

Unsere App sollte die Möglichkeit haben Produkte auch mittels Barcodescannung hinzuzufügen zu können. Dafür wurde eine Library benötigt mit der die Funktion zu der App hinzugefügt werden konnten. Dafür wurden die Library von (journeyapps, 2022) verwendet.

Die Library muss zu den Dependencies hinzugefügt werden.

```
//Barcode
implementation 'com.journeyapps:zxing-android-embedded:4.3.0'
```

Abbildung 67: Implementierung Barcode Library

Der Barcodescanner kann dann im Dokument hinzugefügt werden. Dafür wird als erstes eine Methode scanCode() in der "ProductSearchActivity.class“ erstellt, in der alle Grundkonfigurationen festgelegt werden. Von dieser Methode aus startet man auch den Scanner. In der Methode scanCode() wird als erstes ein Objekt ScanOption erstellt in den dann alle Grundfunktionen definiert werden. Mittels der Methode setPrompt() kann ein Text im Scanfenster angezeigt werden. Die zweite Methode setBeepEnabled() ist auf „true“ gesetzt, somit macht das Telefon einen Beep-Sound, wenn der Barcode erfolgreich erkannt wurde. Die dritte Methode setOrientationLocked() gibt der App die Möglichkeit zu sagen, ob sich der Bildschirm drehen lässt oder nicht. Mittels der vierten Methode setCaptureActivity(CaptureAct.class) wird der Scanner gesetzt. Die benötigte Klasse CaptureAct.class, wird unterhalb noch näher erklärt. In der Zeile 83 wird die Methode gestartet, in der der Barcode selbst ausgewertet wird.

```
77      private void scanCode() {
78          ScanOptions options = new ScanOptions();
79          options.setPrompt("Lautstärke nach oben, um die Taschenlampe einzuschalten");
80          options.setBeepEnabled(true);
81          options.setOrientationLocked(true);
82          options.setCaptureActivity(CaptureAct.class);
83          barLauncher.launch(options);
84      }
```

Abbildung 68: Methode scanCode() in der „ProductSearchActivity.class“

Die Klasse barLaunch wurde an die Bedürfnisse der App angepasst. Die App soll nicht den Barcode anzeigen, sondern mit ihm arbeiten. In der Zeile 81 wird der Barcode selbst an eine weitere Methode changeActivityBarcode() übergeben, die diesen wieder weiterschickt. Ein Beispielbarcode eines „Philadelphia Streichkäses“ sieht folgendermaßen aus: „7622300340285“. Dieser muss am Ende von der Datenbank verarbeitet werden. Die Definition eines Barcodes wir auf der Seite 104 unter der Überschrift „Was ist ein Barcode?“ erklärt. (vgl. Tutorial, 2022)

```
77    ActivityResultLauncher<ScanOptions> barLauncher = registerForActivityResult(new ScanContract(), result -> {
78
79        if (result.getContents() != null) {
80            Log.e("SCANN", "Barcode scanned");
81            changeActivityBarcode(result.getContents());
82        }
83    });
84
```

Abbildung 69: Klasse barlaunch() in der ProductSearchActivity.class

CaptureAct

Diese Klasse in Abbildung 70: "CaptureAct.class“ beinhaltet wenig Code, aber ist für das Scannen des Barcodes erforderlich. Die Klasse benötigt keinen weiteren Code, da der Import alle wichtigen Funktionen, wie die permission Anfordern für die Nutzung der Kamera und das Design.

```
3     import com.journeyapps.barcodescanner.CaptureActivity;
4
5     public class CaptureAct extends CaptureActivity {
6
7     }
8
```

Abbildung 70: „CaptureAct.class“

Jede Seite (Activity) der App muss im „AndroidManifest.xml“ hinterlegt sein, auch der Barcode Scanner, siehe Abbildung 72: AndroidManifest.xml Activity ".ProductSearch.CaptureAct". Hier wird, wie bei der gesamten App die „screenOrientation“ auf „portrait“ gesetzt. Das bedeutet, dass sich die App nicht drehen lässt und nur im Hochformat angezeigt wird.

Die Funktion, die in Zeile 47 auf „true“ gesetzt wurde, ist dazu da, dass die Seite keinen Zustand speichern muss, egal ob sie geschlossen wurde und neugestartet wurde (vgl. Google Developers, 2023).

In Zeile 48 muss ein eigener Style für die Seite von der Dependency eingefügt werden. (vgl. Tutorial, 2022)

```
44      <activity
45          android:name=".ProductSearch.CaptureAct"
46          android:screenOrientation="portrait"
47          android:stateNotNeeded="true"
48          android:theme="@style/zxing_CaptureTheme" />
```

Abbildung 71: AndroidManifest.xml Activity ".ProductSearch.CaptureAct"

Was ist ein Barcode?

Ein Barcode besteht aus verschieden dicken Balken in verschiedenen Abständen. Weiters befinden sich unter den Balken Buchstaben, Ziffern und Sonderzeichen, dies hängt von der Barcode-Art ab. Für den Nutzen der App sind die Barcodes mit der Bezeichnung „Codabar“ relevant, die auch die EN (Europäische Norm) 798 haben.

(vgl. Özdil, 2023)

12.3.2 Login / Benutzerverwaltung

Wie schon auf der Login-Seite (Seite 4950) ersichtlich, werden für die erfolgreiche Anmeldung der „Username und das „Passwort“ benötigt. Wenn die Anmeldung nicht erfolgreich war, wird das dem Benutzer ebenfalls mitgeteilt. Bei einer erfolgreichen Anmeldung kommt man wie gewohnt auf die nächste Seite. Aber was passiert überhaupt im Hintergrund? Bei einer Anmeldung bekommt der Benutzer einen Token, welcher bei jeder Kommunikation mit der Rest-API mitgeschickt wird. Am Ende ist ersichtlich, wer angemeldet ist und Abfragen tätigen möchte.

12.3.3 Registrierung

Bei der App kann man sich einen neuen Account erstellen. Dies geht wie bei der Überschrift Login-Seite (Seite 4950) beschrieben. Der „Vorname“ und „Nachname“ sind nur für Formalitätszwecke erforderlich, sind bei der Appnutzung nicht notwendig. Der „Username“ und das „Passwort“ sind im Gegenteil sehr relevant für die Nutzung der App. Diese Daten sind dazu da, dass sich der Benutzer anmelden kann.

12.3.4 Navigationsleiste

Das Design der Navigationsleiste muss selbst erstellt werden, welches im Unterordner „menu“ gespeichert wird, siehe Abbildung 85: Speicherort Design Navigationsleiste.

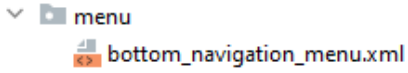


Abbildung 72: Speicherort Design Navigationsleiste

Am Ende sieht die Leiste wie auf Abbildung 73: Navigationsleiste Darstellung aus.

Abbildung 73: Navigationsleiste Darstellung

Die Bilder für die Navigationsleiste wurden aus dem „Configure Vector Asset“ geholt. Hier werden alle Standard Android Bilder zu Verfügung gestellt, siehe Abbildung 88: Vector Asset Konfigurator: Navigationsleiste Bilder.

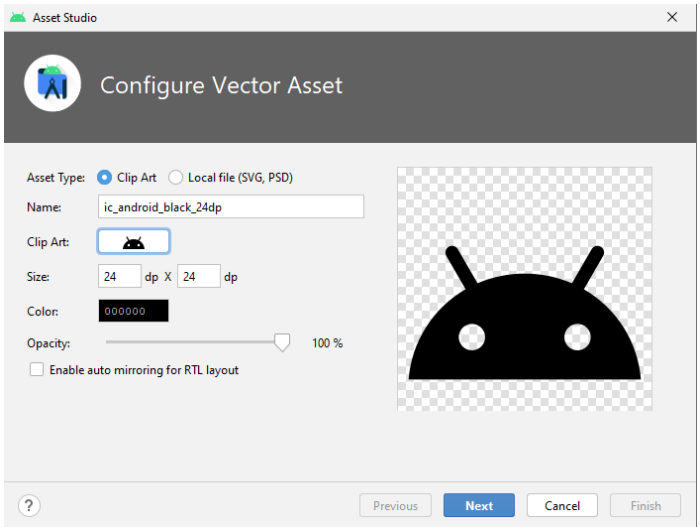


Abbildung 74: Vector Asset Konfigurator: Navigationsleiste Bilder

Die Navigationsleiste muss außerdem in die richtigen Activites eingebunden werden. Unter dem Punkt „app:itemBackground“ kann man die Hintergrundfarbe der Navigationsleiste festlegen. Unter den beiden Punkten „app:itemIconTint“ und „app:itemTextColor“ muss der „selector.xml“ hinterlegt werden.

```
52      <com.google.android.material.bottomnavigation.BottomNavigationView
53          android:id="@+id/bottom_navigation"
54          android:layout_width="wrap_content"
55          android:layout_height="56dp"
56          android:layout_alignParentBottom="true"
57          app:itemBackground="@color/smood"
58          app:itemIconTint="@drawable/selector"
59          app:itemTextColor="@drawable/selector"
60          app:menu="@menu/bottom_navigation_menu" />
```

Abbildung 75: Navigationsleiste in Activities einbinden

Die App muss ebenfalls wissen, welche Seite zu welcher „ID“ gehört und wann eine angezeigt werden muss. Dazu ist das Codesegment da, siehe Abbildung 76: Navigationsleiste managen. In einem „switch“ werden die beiden Seiten aufgelistet und wenn auf eine der Beiden geklickt wird, dann wird die geforderte Seite mittels eines “startActivity(new Intent(getApplicationContext(), *jeweilige Seite.class*));“ Aufrufes angezeigt. Wenn man auf den Button der aktuellen Seite klickt, passiert nichts, denn man befindet sich bereits auf der ausgewählten Seite. (vgl. jangirkaran17, 2022)

```
120         // Perform item selected listener
121         bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
122             @Override
123             public boolean onNavigationItemSelected(@NonNull MenuItem item) {
124
125                 switch (item.getItemId()) {
126                     case R.id.shoppinglist:
127                         startActivity(new Intent(getApplicationContext(), ShoppingListActivity.class));
128                         overridePendingTransition( enterAnim: 0,  exitAnim: 0);
129                         return true;
130                     case R.id.home:
131                         return true;
132                 }
133                 return false;
134             }
135         });
136     }
```

Abbildung 76: Navigationsleiste managen

12.3.5 Push-Benachrichtigungen

Push Notification in Android

NotificationCompat.Builder

Zu Beginn wird der Inhalt und der Kanal der Benachrichtigung mittels eines NotificationCompat.Builder-Objekts festgelegt. (vgl. developer.android.com, 2023)

Der folgende Codeausschnitt zeigt, wie die Benachrichtigung mit folgendem Inhalt erstellt wurde:

- Ein kleines Icon, wird mit setSmallIcon() gesetzt. Dieses ist für die Benutzer sichtbar.
- Ein Titel wird mittels setContentTitle() gesetzt.
- Die Nachricht wird duch setContentText() gesetzt.
- Mit setPriority() wird die Priorität der Nachricht gesetzt. Diese beschreibt, wie dringlich die Benachrichtigung sein soll.
- Mittels setCategory() setzt man die Kategorie der Benachrichtigung. Andere Auswahlmöglichkeiten abgesehen von Benachrichtigungen sind Alarm, E-Mail, Call und Error.
- Mit setVisibility() wird die Sichtbarkeit der Nachricht festgelegt. Weitere Möglichkeiten neben Public sind Private und Secret.
- Der Sound der Nachricht wird mit setSound() gesetzt.

```
NotificationCompat.Builder notification = new NotificationCompat.Builder( context: this, App.NOTIFICATION_NORMAL)
        .setSmallIcon(R.mipmap.ic_launcher_foreground)
        .setContentTitle(title)
        .setContentText(msg)
        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setCategory(NotificationCompat.CATEGORY_MESSAGE)
        .setVisibility(NotificationCompat.VISIBILITY_PUBLIC)
        .setSound(Settings.System.DEFAULT_NOTIFICATION_URI);
```

Abbildung 77: Codebeispiel NotificationCompat.Builder -Objekt

Notification Channel

Weiters braucht man auch einen Notification Channel.

Seit Android 8.0 müssen Benachrichtigungen einem Kanal zugeordnet sein. Diese Kanäle ermöglichen den Benutzern, die Benachrichtigungen der App zu steuern.

Mit einem Notification Channel können verschiedene Kategorien von Benachrichtigungen definiert, werden. Diese Kanäle können unterschiedliche Einstellungen für Lautstärke, Vibrationsmuster, Lichter und Priorität haben. Sie können dann vom Benutzer in den Einstellungen der App verwaltet werden.

Wenn eine App eine Benachrichtigung ausgibt, wird sie einem bestimmten Kanal zugeordnet, der von der App definiert wurde. Wenn der Benutzer die App öffnet, kann er die Benachrichtigungseinstellungen für jeden Kanal einstellen, um zu bestimmen, wie die Benachrichtigungen angezeigt werden sollen.

Um einen Notification Channel zu erstellen, muss ein NotificationChannel-Objekt erstellt werden. Diesem wird eine eindeutige Channel-ID, einem für den Benutzer ersichtlichen Namen und eine Benachrichtigungskategorie zuweisen.

Das Objekt NotificationManager ist dafür da, den Benutzer bei Ereignissen zu benachrichtigen. Mittels der Methode createNotificationChannel() wird der Benachrichtigungskanal registriert.

```
public void createNotificationChannels() {
    NotificationChannel notificationChannel = new NotificationChannel(
            App.NOTIFICATION_NORMAL,
            "Abgelaufenes Produkt",
            NotificationManager.IMPORTANCE_HIGH
    );
    notificationChannel.setLockscreenVisibility(Notification.VISIBILITY_PUBLIC);
    notificationChannel.enableVibration(true);
    notificationChannel.enableLights(true);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.Q) {
        notificationChannel.setAllowBubbles(true);
        notificationChannel.setLockscreenVisibility(Notification.VISIBILITY_PUBLIC);
    }
    notificationChannel.setDescription("Alle abgelaufenen Produkte");

    NotificationManager manager = context.getSystemService(NotificationManager.class);
    manager.createNotificationChannel(notificationChannel);
}
```

Abbildung 78: Codebeispiel Notification Channel

Im darauffolgenden Bild sieht man den Benachrichtigungskanal von Smood.

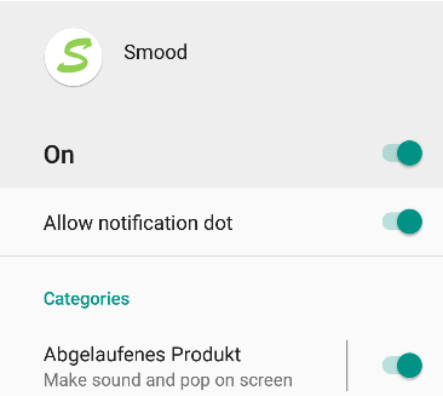


Abbildung 79: Notification-Channel von Smood

Wie werden die Push Notifications gestartet?

In dieser Diplomarbeit wird ein PeriodicWorkRequest verwendet, was für wiederkehrende Arbeiten genutzt wird.

Was ist ein PeriodicWorkRequest?

Ein PeriodicWorkRequest ist ein Konstrukt in der Android-Job-Scheduler-Bibliothek, mit dem periodische Aufgaben, in diesem konkreten Fall ein Hintergrundprozess, in einer Android-App geplant werden können.

Mit einem PeriodicWorkRequest kann der Entwickler eine Aufgabe definieren, die regelmäßig in bestimmten Zeitintervallen ausgeführt wird. Bei Smood ist das einmal am Tag. Diese Aufgabe wird dann von der Job-Scheduler-Bibliothek im Hintergrund ausgeführt, wenn bestimmte Kriterien erfüllt sind, wie zum Beispiel eine Netzwerkverbindung oder eine bestimmte Batteriestufe.

Die Arbeit, die im PeriodicWorkRequest definiert wird, kann eine einfache Methode sein, die eine bestimmte Aktion ausführt, oder eine komplexe Aufgabe, die im Hintergrund ausgeführt wird.

Der PeriodicWorkRequest kann auch konfiguriert werden, um das Verhalten der Aufgabe anzupassen, oder um das Verhalten der Aufgabe bei der Ausführung im Hintergrund zu kontrollieren. (vgl. developer.android.com, 2023; developer.android.com, 2023)

In der App Smood wird ein PeriodicWorkRequest Objekt erzeugt, dem man die Worker Klasse mit der doWork() Methode und die Zeit, in der es wiederholt wird, mitgibt.

```
PeriodicWorkRequest saveRequest =
        new PeriodicWorkRequest.Builder(NotificationService.class,
                repeatInterval: 24, TimeUnit.HOURS)
                .build();
```

Abbildung 80: PeriodicWorkRequest von Smood

Ein PeriodicWorkRequest wird normalerweise in Verbindung mit dem WorkManager verwendet, einem Systemdienst in Android, der die Ausführung von Aufgaben in der App plant und verwaltet.

Um nun den PeriodicWorkRequest in die Warteschlange zu stellen, wird ein WorkManger Objekt erstellt, und die Methode enqueueUniquePeriodicWork aufgerufen. Der PeriodicWorkRequest bekommt so einen eigenen Namen, mit dem er eindeutig identifiziert werden kann.

```
WorkManager.getInstance(this).enqueueUniquePeriodicWork( uniqueWorkName: "Work",
        ExistingPeriodicWorkPolicy.KEEP, saveRequest);
```

Abbildung 81: WorkManager von Smood

Wann muss die Nachricht gesendet werden?

Um zu überprüfen, wann ein Produkt drei Tage vor dem Ablaufen ist, wird der Rest Endpoint getSoonToExpireProducts aufgerufen. Dieser Endpunkt gibt eine Liste von Produkten zurück, deren Ablaufdatum innerhalb der nächsten drei Tage liegen.

Um sicherzustellen, dass der Endpunkt regelmäßig überprüft wird, wird das Datum des letzten Aufrufs des Endpunkts gespeichert. Wenn das Datum um einen Tag überschritten wurde, wird der Endpunkt erneut aufgerufen.

Pending Intent

Ein Pending Intent ist ein Objekt in der Android-Entwicklungsumgebung, das verwendet wird, um einen Intent zu beschreiben, der zu einem späteren Zeitpunkt ausgeführt werden soll. Es wird häufig in Verbindung mit Benachrichtigungen, Alarmen oder anderen Ereignissen verwendet, bei denen eine Aktion ausgelöst werden soll, auch wenn die App nicht aktiv im Vordergrund läuft.

Ein Pending Intent enthält Informationen über die Absicht, die ausgeführt werden soll, einschließlich der Aktion. Weiters enthält er die Komponente (z.B. Aktivität oder Service), die gestartet werden soll, und eventuell die Daten, die an die Komponente übergeben werden sollen. Es kann auch spezifische Optionen enthalten, wie zum Beispiel Flags, die bestimmen, wie die Absicht ausgeführt werden soll.

(vgl. DigitalOcean, 2023)

```
PendingIntent resultPendingIntent =
        stackBuilder.getPendingIntent( requestCode: 0,
                flags: PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);
```

Abbildung 82: Pending Intent

12.3.6 Sonstige Implementierung

Internet-Permission

„Internet-Permissions“ gehören zu dem Bereich „uses-permission“. Diese „uses-permission“ können für zwei Wege genutzt werden, der Erste ist für diese Diplomarbeit nicht relevant, dass Google Play die App richtig filtern und anzeigen kann und welche Freigaben die App vom Nutzer haben möchte. Der Vorteil warum wir solch eine „permission“, siehe Abbildung 83: Internet-Permission AndroidManifest.xml, in unserer App eingebunden haben ist, dass wir so Hardwarefunktionen, wie den Internetzugriff freischalten können. (vgl. Google Developers, 2023)

```
4       package="at.ac.htlperg.smood">
5
6       <uses-permission android:name="android.permission.INTERNET" />
7
8       <application
```

Abbildung 83: Internet-Permission AndroidManifest.xml

Retrofit2

Mittels Requests werden Daten der REST-API an die App oder von der App an die REST-API geschickt. Damit die App die Daten von der REST-API einlesen kann, um diese in späterer Folge zu nutzen, muss für jede Struktur der Daten ein Model erstellt werden. Zum Senden von Daten an die REST-API muss diese ebenfalls gemacht werden.

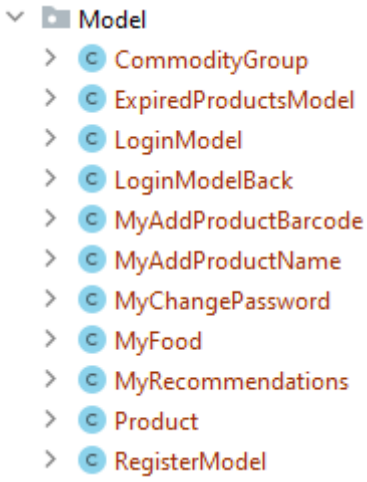


Abbildung 84: Liste der Models

Models

Die Models, welche für das Senden und Bekommen der Daten benötigt werden, werden in einem Unterordner (Subpackage) gespeichert, um so einen Überblick über alle Dateien zu haben. Für alle Seiten und Base Adaper gibt es auch einzelne Unterordner (Subpackage).

In dieser App haben werden 11 Models benutzt. Der Name sagt meist schon aus, wofür diese verwendet werden, siehe Abbildung 84: Liste der Models. Ein Model besteht aus den Attributen, die benötigt werden. Ebenfalls beinhaltet ein Model einen Konstruktor. Abschließend hat jedes Model zu jedem Attribut eine GET -und SET- Methode. Diese Methoden sind dafür da, dass der Inhalt der Attribute ausgegeben oder gesetzt werden kann.

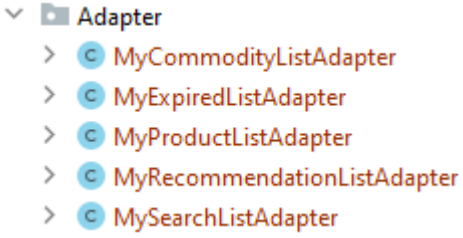


Abbildung 85: Liste der Adapter

Base Adapter

Für das Anzeigen der Daten in den Listen, welche von der REST-API kommen, benötigt man einen Base Adapter. Davon sind 5 Stück in der App vorhanden (siehe Abbildung 85: Liste der Adapter). Ein Base Adapter ist dafür da, um die Daten in die Zellen richtig einzufügen.

Auf der Abbildung 86: Base Adapter (MyRecommendationListAdapter) ist ein Beispiel Base Adapter abgebildet. In Zeile 17 wird ein Objekt mit dem Namen „LayoutInflator“ angelegt, in dem in der Zeile 44 das Layout (Aufbau) der Zelle, in der die Daten angezeigt werden, gespeichert wird, hier das Layout der Zelle für die Liste Abgelaufene – Empfohlene Lebensmittel Seite (Seite 50). Unter dem „LayoutInflator“ in Zeile 17, wird in Zeile 18 eine Liste aller Objekte gespeichert, diese wird benötigt, um dann die Zellen mit Daten zu befüllen. In Zeile 20 gibt es wieder einen Konstruktor, in dem die Liste der Daten und der Kontext der App übergeben wird.

```java
public class MyRecommendationListAdapter extends BaseAdapter {
    private Context context;
    private final LayoutInflater inflater;
    private List<MyRecommendations> myRecommendations;

    public MyRecommendationListAdapter(Context context, List<MyRecommendations> myRecommendations) {...}

    @Override
    public int getCount() { return myRecommendations.size(); }

    @Override
    public Object getItem(int i) { return myRecommendations.get(i).getFoodname(); }

    @Override
    public long getItemId(int i) { return i; }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {

        view = inflater.inflate(R.layout.recommendation_cell,  root: null);

        TextView nameProduct = view.findViewById(R.id.nameProduct);
        TextView amount = view.findViewById(R.id.amount);
        TextView recommendationDateProduct = view.findViewById(R.id.recommendationDateProduct);
        TextView bestByDateProduct = view.findViewById(R.id.bestByDateProduct);

        nameProduct.setText(myRecommendations.get(i).getFoodname());
        amount.setText(myRecommendations.get(i).getAmount() + "");
        recommendationDateProduct.setText(myRecommendations.get(i).getRecommendationDate());
        bestByDateProduct.setText(myRecommendations.get(i).getBestByDate());
        return view;
    }

    public List<MyRecommendations> getMyRecommendations() { return myRecommendations; }
}
```

Abbildung 86: Base Adapter (MyRecommendationListAdapter)

GET – Request

```
138    private void loadList() {
139        myCommodityListAdapter = new MyCommodityListAdapter(getApplicationContext(), commodityGroups);
140        listView.setAdapter(myCommodityListAdapter);
141
142        Retrofit retrofit = App.retrofitAPI;
143        SmoodAPI apiInterface = retrofit.create(SmoodAPI.class);
144        Call<List<CommodityGroup>> call = apiInterface.getCommodities(App.getUsername( context: MainActivity.this),
145                App.getToken( context: MainActivity.this));
146        call.enqueue(new Callback<List<CommodityGroup>>() {
147            @Override
148            public void onResponse(Call<List<CommodityGroup>> call, Response<List<CommodityGroup>> response) {
149                if (response.code() != 200) {
150                    Log.e( tag: "Error",  msg: "Loading Error!" + response.code());
151                    return;
152                }
153                List<CommodityGroup> loadCommodityGroup = response.body();
154                commodityGroups.addAll(loadCommodityGroup);
155                myCommodityListAdapter.notifyDataSetChanged();
156            }
157
158            @Override
159            public void onFailure(Call<List<CommodityGroup>> call, Throwable t) {
160                Log.i( tag: "",  msg: "Programm Failed Here");
161            }
162        });
163    }
```

Abbildung 87: GET-Request Retrofit2 (Home-Seite: Liste Kategorien)

In dieser Methode von Abbildung 87: GET-Request Retrofit2 (Home-Seite: Liste Kategorien) wird, wie die Abbildungsbeschreibung bereits erklärt, in einem GET-Request die Liste der Kategorien des angemeldeten Benutzers von der REST-API geholt.

In der Zeile 142 wird die Grundstruktur des Retrofit Objektes mit der Grund-URL (http://smood.htl-perg.ac.at) geladen. In der Zeile darunter wird der zweite Teil der URL zur Grund-URL hinzugefügt. Die Interfaceklasse „SmoodAPI.java“ beinhaltet alle URLs mit deren Request-Typen und deren Rückgabewerte und Übergabewerte, um alles übersichtlich zu halten und um Fehler auszuschließen, siehe . Die Zeile 144 erstellt ein Objekt, welches in der Zeile 146 den GET-Request-Aufruf durchführt. Im Objekt „call“, wird der Rückgabewert mit „<List<CommodityGroup>>“ (die Liste der Kategorien) festgelegt, sonst werden hier nur die Werte übergeben, die im Interface definiert wurden. Beispiel auf der Abbildung 88: GET-Request-URL-Teil vom Interface (SmoodAPI.java).

Im Request-Aufruf gibt es zwei Submethoden, eine davon wird aufgerufen, wenn ein Fehler beim Aufrufen der URL passiert. Die Methode ist in der Abbildung 87: GET-Request Retrofit2 (Home-Seite: Liste Kategorien) in Zeile 159 und heißt onFailure(). In der Methode wird nur

eine Fehlermeldung zurückgegeben. Wenn der Aufruf erfolgreich ist, wird die Methode in Zeile 148 mit dem Namen onResponse() aufgerufen. In der Methode ist anfangs in Zeile 150 noch eine Sicherheitsabfrage, welche prüft, ob der Statuscode der Rückgabe „200“ ist, ansonsten wird eine andere Fehlermeldung zurückgeben. Wenn bist jetzt alles erfolgreich ist, befindet man sich in Zeile 153 und holt sich den Rückgabewert mit response.body(). Dies ist dafür da, dass alles asynchron abläuft, um das Appverhalten nicht zu beeinträchtigen. Abschließend wird noch der Base Adapter benachrichtigt, dass neue Daten verfügbar sind.

```
25      @GET("/smood/Food/getCommodityGroupsFormUser/{username}")
26      Call<List<CommodityGroup>> getCommodities(@Path("username") String username, @Header("token") String token);
```

Abbildung 88: GET-Request

In der Abbildung 88: GET-Request ist in der Zeile 25 ganz am Anfang festgelegt, dass es ein „@GET“ Request ist. Daneben steht in der Klammer der URL-Teil, der für diesen Aufruf hinten hinzugefügt wird. Das „@Header“ ist immer der Token des angemeldeten Benutzers der als String bei fast jedem Aufruf mit übergeben werden muss. Man kann ebenfalls noch sehen, dass als Rückgabewert eine „<List<CommodityGroup>>“ kommt.

GetApplicationContext() gibt den Kontext des einzelnen, globalen Anwendungsobjekts des aktuellen Prozesses zurück. (vgl. Goolge Developers, 2023)

POST – Request

```
116    private void apiCallAddWithName() {
117        myAddProductName.setExpirationDate(expirationDate);
118        SmoodAPI smoodAPI = App.retrofitAPI.create(SmoodAPI.class);
119        Call<Void> call = smoodAPI.addProductName(myAddProductName, App.getToken( context: ProductActivity.this));
120        call.enqueue(new Callback<>() {
121            @Override
122            public void onResponse(Call<Void> call, Response<Void> response) {
123                if (response.code() != 200) {
124                    Log.e( tag: "ProductActivity", response.toString());
125                    return;
126                }
127                Log.e( tag: "",  msg: "Loading good!");
128            }
129
130            @Override
131            public void onFailure(Call<Void> call, Throwable t) {
132                t.printStackTrace();
133                Log.i( tag: "",  msg: "No Adding");
134            }
135        });
136    }
```

Abbildung 89: POST-Request Retrofit2

Der POST-Request ist ähnlich aufgebaut wie der GET – Request. In diesem Beispiel gibt es keinen Rückgabewert, aber das muss nicht zwangsweise sein. Der große Unterschied ist, dass in Zeile 119 ein Modelobjekt im Aufruf übergeben wird. In der kann man auch sehen, dass ein „@Body“ im Call-Aufruf ist und im nicht. Da dieser Aufruf keinen Rückgabewert hat, wird nur in Zeile 127 eine Meldung ausgegeben, dass der Aufruf erfolgreich war. Das hat keinen Einfluss auf die Nutzung und der Benutzer sieht diese auch nicht, dieser Rückgabewert ist nur für die Entwickler der App notwendig.

```
43    @POST("/smood/Food/addProduct/Name")
44    Call<Void> addProductName(@Body MyAddProductName myAddProductName, @Header("token") String token);
```

Abbildung 90: POST-Request

Der Aufbau des Interfaceeintrags Abbildung 90: POST-Request ist ähnlich, wie der der Abbildung 88: GET-Request. Es unterscheidet sich in der Zeile 43. In der Zeile 44 steht als Rückgabewert „Void“, das heißt es gibt keinen Rückgabewert, denn das kann bei GET-Request nicht auftreten. Wie oberhalb beschrieben, kann bei einem POST-Request auch ein Rückgabewert sein. Im Body können auch größere Mengen an Daten übergeben werden. Bei diesem Aufruf wird der Token mittels „@Header“ übergeben. Ein POST-Request kann ebenfalls Werte in der URL übergeben, wie bei Abbildung 88: GET-Request, dies wird jedoch bei diesem Beispiel nicht benötigt.

PUT – Request

```
38    private void putProductConsumed() {
39        Retrofit retrofit = App.retrofitAPI;
40        SmoodAPI apiInterface = retrofit.create(SmoodAPI.class);
41        Call<Void> call = apiInterface.setOnConsumed(App.getUsername( context: this), product.getExpirationDateAsSQLDate(),
42                makeToWebadressString(product.getFoodName()), App.getToken( context: ProductViewActivity.this));
43        call.enqueue(new Callback<>() {
44            @Override
45            public void onResponse(Call<Void> call, Response<Void> response) {
46                if (response.code() != 200) {
47                    Log.e( tag: "Consumed", response.toString());
48                }else{
49                    Log.e( tag: "Consumed",  msg: "Product " + product.getFoodName() + " set on Consumed");
50                }
51                goBackToHome();
52            }
53
54            @Override
55            public void onFailure(Call<Void> call, Throwable t) {
56                Log.i( tag: "",  msg: "Programm Failed Here");
57            }
58        });
59    }
```

Abbildung 91: PUT-Request Retrofit2 (Detailansicht Produkt: Produkt auf Abgelaufen setzten)

Die App ruft einen PUT-Request auf, um die Lebensmittel-Details zu bekommen. In dem Aufruf ist eine Methode ausgelagert, siehe Abbildung 92: Detailansicht Produkt: zu Home zurückgehen, so kann der Aufruf in Zukunft wieder verwenden werden. In der Zeile 42 wird für das Übergeben des Produktnamens eine Methode mit dem Namen makeToWebadress-String() verwendet.

In der Methode auf der Abbildung 93: Detailansicht Produkt: Produktnamen umschreiben für das Senden per URL, wird der übergebene String in ein für die URL sendebereiten String umgewandelt. (vgl. Rieke, 2022).

```
71    private void goBackToHome() {
72        Intent productInfo = new Intent( packageContext: ProductViewActivity.this, MainActivity.class)
73        startActivity(productInfo);
74    }
```

Abbildung 92: Detailansicht Produkt: zu Home zurückgehen

```
76 @      private String makeToWebadressString(String s) {
77            s = s.replace( target: "Ü",  replacement: "%C3%BC");
78            s = s.replace( target: "ß",  replacement: "%C3%9F");
79            s = s.replace( target: "ä",  replacement: "%C3%A4");
80            s = s.replace( target: "ö",  replacement: "%C3%B6");
81            s = s.replace( target: "Ä",  replacement: "%C3%84");
82            s = s.replace( target: "Ö",  replacement: "%C3%96");
83            s = s.replace( target: "Ü",  replacement: "%C3%9C");
84            s = s.replace( target: ".",  replacement: "%2E");
85            //s = s.replace("%", "%25");
86            s = s.replace( target: "'",  replacement: "%27");
87            s = s.replace( target: " ",  replacement: "%20");
88            return s;
89        }
```

Abbildung 93: Detailansicht Produkt: Produktnamen umschreiben für das Senden per URL

```
46      @PUT("/smood/Food/setOwnConsumed/{username}/{productName}/{dateToChange}")
47      Call<Void> setOnConsumed(@Path("username") String username, @Path("dateToChange") Date dateToChange,
48                               @Path(value = "productName") String productName, @Header("token") String token);
```

Abbildung 94: PUT-Request-URL-Teil vom Interface (SmoodAPI.java)

Das Interface ist auch wie bei Abbildung 91: PUT-Request Retrofit2 (Detailansicht Produkt: Produkt auf Abgelaufen setzten) ähnlich, wie bei der auf Abbildung 94: PUT-Request-URL-Teil vom Interface (SmoodAPI.java). Wie bei den anderen Request-Methoden (Post und Get) steht eine andere Standardoperation vor der URL. Dieser Aufruf hat keine Rückgabewert. Hierbei werden drei Parameter übergeben. Als erstes wird der Benutzername (username, dann der Produktname (productName) und abschließend das Ablaufdatum des Produktes (dateToChange) übergeben.

Date Picker

Die Lebensmittel-Details-Seite, soll eine schöne und leicht zu bedienende Ablaufdatumsfestlegung zeigen. (vgl. Tutorial, 2022)

```
<EditText
    android:id="@+id/et_date"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:hint="Wähle ein Datum"
    android:textAlignment="center"
    android:textSize="20sp" />
```

Abbildung 95: Date Picker: Layouteditor

```
81          etDate.setOnClickListener(view -> {
82              DatePickerDialog datePickerDialog = new DatePickerDialog(
83                      context: ProductActivity.this, (datePicker, year1, month1, day1) -> {
84                  month1 = month1 + 1;
85                  String date = year1 + "-" + month1 + "-" + day1;
86                  expirationDate = Date.valueOf(date);
87                  etDate.setText(date);
88              }, year, month, day);
89              datePickerDialog.show();
90          });
91
92      }
```

Abbildung 96: Detailansicht Produkt: Date-Picker-Methode

In der Java-Klasse wird aus dem Eingabefeld ein Button erstellt, der einen Kalender anzeigt. Das passiert mit dem Methodenaufruf „.setOnClickListener()“. Der Kalender erscheint nur, weil ein „DatePickeDialog“-Objekt erstellt wird. In den Folgezeilen wird festgelegt, welche Variablen wo befüllt werden. In der Zeile 85 wird das Ergebnis zu einem „String“, dieser wird in der nächsten Zeile im Eingabefeld angezeigt.

13 Ergebnis – Funktionen aus Benutzersicht

13.1 Hinzufügen der Lebensmittel ins Lager

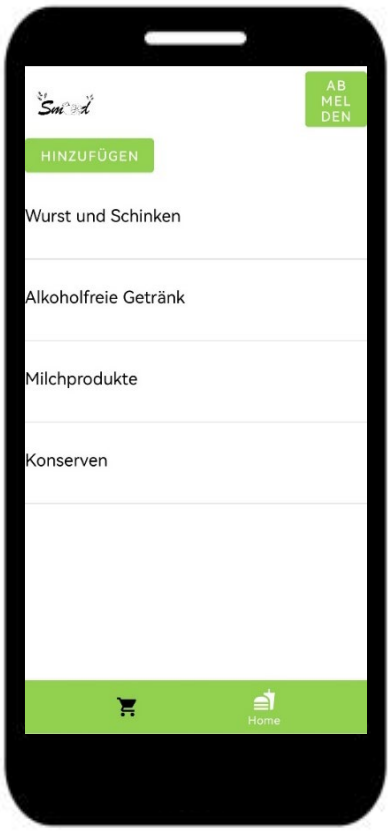


Abbildung 97: Hinzufügen der Lebensmittel

Das Hinzufügen von Lebensmitteln zur App ist in dieser Diplomarbeit ein wichtiger Bestandteil. Diese Funktion wird benötigt, um die gekauften Produkte oder die bereits im Kühlschrank befindlichen, in die App zu bekommen und diese dort anzuzeigen.

Um Lebensmittel in die App hinzufügen zu können muss man sich auf der Startseite befinden.
Hier kann der Button „Hinzufügen“ geklickt werden.

Danach wird man auf eine Ansicht zum Hinzufügen eines neuen Produktes weitergeleitet werden. Jetzt kann sich der Benutzer entscheiden auf welche Weise das Produkt erfasst werden soll. Einerseits kann der Barcodescanner verwendet werden, um den Barcode des Produktes zu scannen. Andererseits ist es ebenfalls möglich den Namen des Produktes in die Suchleiste einzugeben und so nach dem Produkt zu suchen.

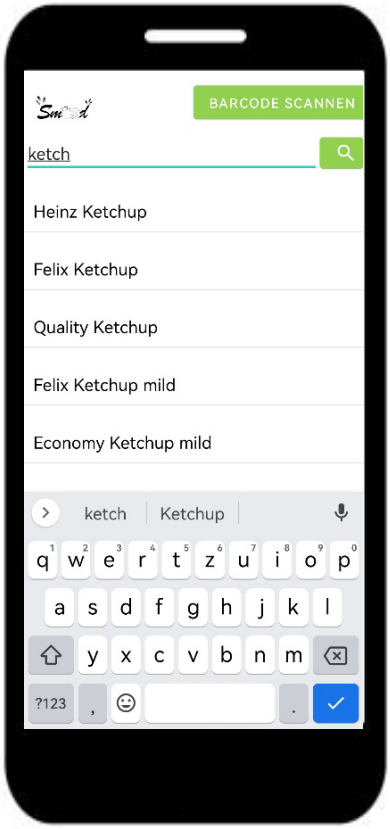


Abbildung 98: Suche von Produkten durch den Namen

Zuerst wird die Suche nach einem Lebensmittel mittels der Suchleiste beschrieben. Dabei muss lediglich der Name des Produktes eingegeben werden und auf den Suchbutton, rechts neben der Suchleiste, geklickt werden. Jetzt sind alle Suchvorschläge, die zur Eingabe des Benutzers passen in einer Liste ersichtlich. Als nächstes muss eines der angezeigten Produkte angeklickt werden, um auf die nächste Seite weitergeleitet zu werden.

Die zweite Möglichkeit zur Erfassung von Lebensmittel, ist der Barcodescanner. Dabei muss nur auf den Button „Barcode Scannen“ über den Suchbutton gedrückt werden. Jetzt öffnet sich der Barcodescanner am Handy und der Barcode des Produktes kann ganz einfach eingescannt werden.

Nach der Auswahl des Produktes oder dem Einscannen wird man auf die letzte Seite weitergeleitet werden. Hier muss das Ablaufdatum des Nahrungsmittels eingegeben werden. Dabei bestehen die Möglichkeiten das Datum manuell einzugeben oder durch das vorhandenen Datumsauswahlfenster ein Datum auszuwählen.

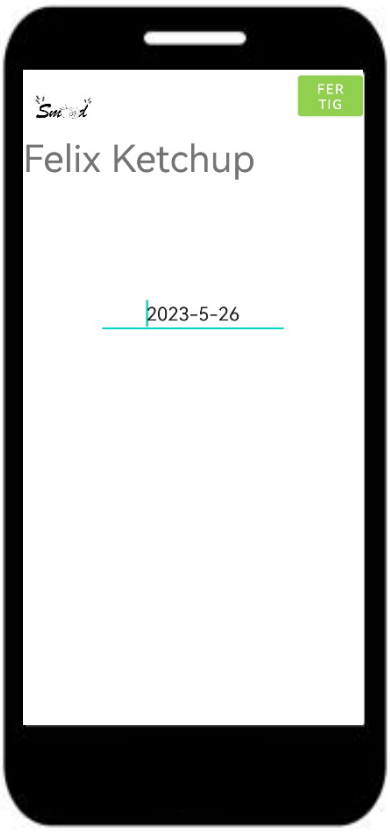


Abbildung 99: Datumsvergabe beim Hinzufügen der Lebensmittel

Um das Produkt endgültig hinzuzufügen, muss im Fenster zur Vergabe des Datums auf „Fertig“ geklickt werden. Jetzt ist das Produkt in der App vorhanden, wie in der Grafik ersichtlich. Nun ist das Lebensmittel bereit zum Verbrauch. Weitere Produkte können ganz einfach zur App hinzugefügt werden.

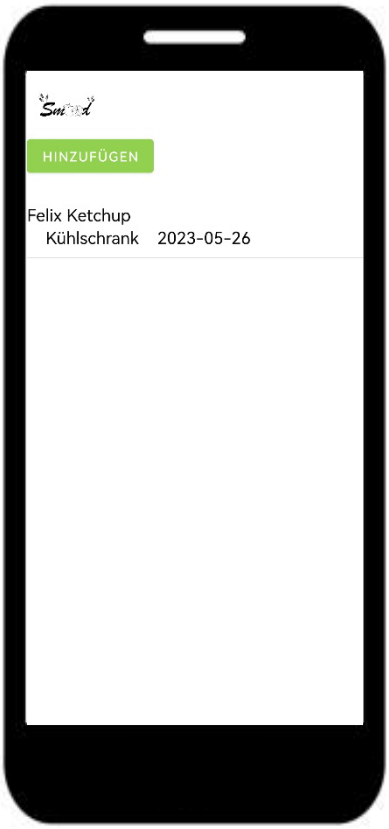


Abbildung 100: Lebensmittel hinzugefügt

13.2 Detailansicht der Lebensmittel

Sobald man auf die gewünschte Lebensmittelkategorie auf der Home Ansicht klickt, werden alle hinzugefügten Produkte in der darunterliegenden Lebensmittelansicht-Seite angezeigt. Die Detailansicht kann angezeigt werden, indem man auf das gewünschte Lebensmittel klickt.

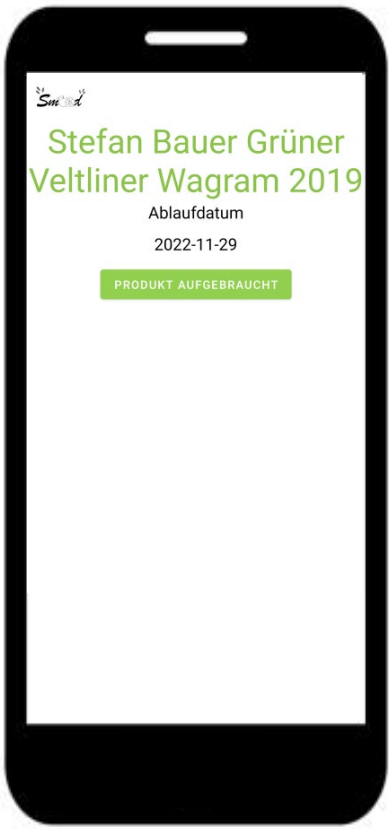


Abbildung 101: Lebensmittel Detailansicht

13.3 Empfehlungen für kommende Einkäufe

Abbildung 102: Empfohlene Lebensmittel

Eine weitere wichtige Funktion der App ist, die Ansicht für die Lebensmittel, welche aufgrund des Kaufverhaltens des Benutzers, empfohlen werden. Dabei handelt es sich um eine Einkaufsliste, welche vom Algorithmus zusammengestellt wird und genau auf den Kunden zugeschnitten ist. Sobald Produkte öfter hinzugefügt und wieder aufgebraucht werden, werden diese Produkte empfohlen und in der Ansicht der Empfohlenen Produkte angezeigt. Diese Ansicht ist auf der zweiten Seite, der Shoppinglist Seite, zu betrachten. In der nebenstehenden Grafik kann diese Ansicht betrachtet werden.

Empfohlene Produkte werden immer mit einer Menge angegeben. Die Menge gibt eine Aussage darüber wie viel Stück von dem Produkt gekauft werden soll, um den Bedarf des Benutzers zu decken. Darüber hinaus wird auch ein Mindesthaltbarkeitsdatum empfohlen, welches perfekt zum Verbrauch des Benutzers passt. Zu guter Letzt kann der Tag der Empfehlung betrachtet werden, um sicher zu stellen, dass es sich um eine aktuelle Empfehlung handelt.

13.4 Übersicht über bald ablaufende / abgelaufene Lebensmittel

Diese Seite wird in zwei Bereiche unterteil, in die Empfohlene-Produkte-Seite und in die Abgelaufenen-Produkte-Seite. Im unteren Teil der Abgelaufenen-Produkte-Seite, werden dem Benutzer alle Produkte, die er hinzugefügt hat oder welche in den nächsten Tagen ablaufen, angezeigt. Die Möglichkeit besteht, eine detaillierte Ansicht anzuzeigen. So ist es auch bei diesen Seiten möglich, dass die Produkte auf „Aufgebraucht“ gesetzt werden können.

13.5 Einloggen oder neuen Benutzer erstellen

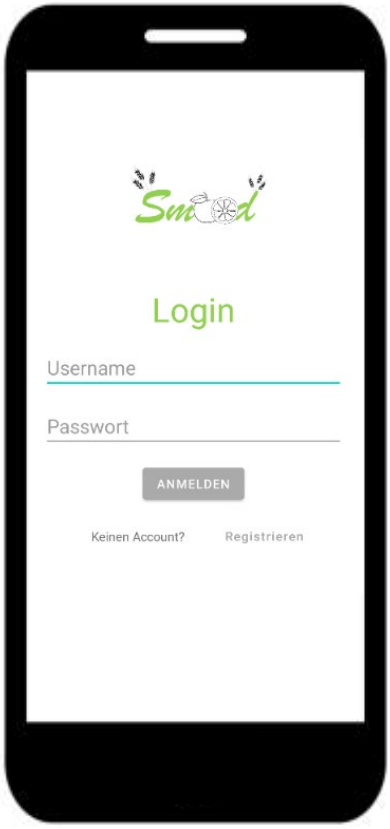


Abbildung 103: App Login Smood

Wenn ein Benutzer die App zum ersten Mal aufmacht, sieht er diese Startseite vor sich.

Wenn ein Benutzer bereits einen Account in der App hat, kann er sich jederzeit einfach mit seinem Usernamen und Passwort anmelden.
Wenn ein Benutzer noch keinen Account hat, ist es notwendig, sich neu zu registrieren.
Dies wird gemacht, indem man im Login Fenster auf Registrieren klickt. Danach wird man auf eine neue Seite, nämlich auf die Registrierungsseite weitergeleitet.

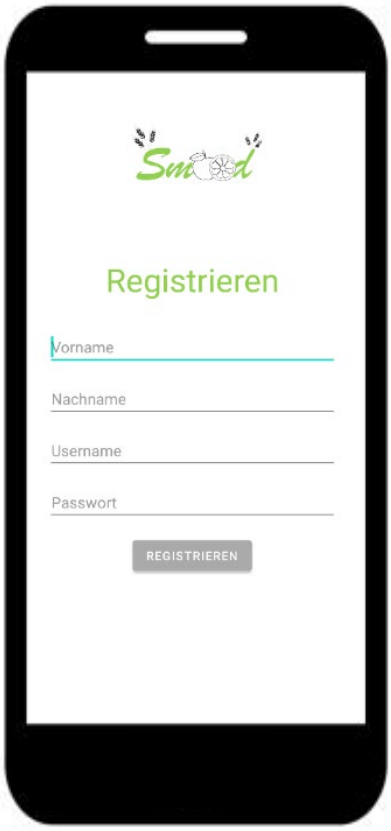


Abbildung 104: Registrierung Smood

Es ist wichtig, dass man bei der Eingabe des Vor- und Nachnamens sowie des Usernamens und des Passworts sehr sorgfältig vorgeht. Denn sobald diese Daten eingegeben und abgesendet wurden, kann man sie nicht mehr ändern. Deshalb sollte man sich Zeit nehmen, um sicherzustellen, dass alle Angaben korrekt sind und keine Tippfehler enthalten sind.
Besonders wichtig ist, sich den Usernamen und das Passwort gut zu merken.
Sobald alle erforderlichen Informationen eingegeben wurden, muss man nur noch auf den Button "registrieren" klicken, um den Registrierungsprozess abzuschließen. Der Benutzer wird danach sicher in der App angemeldet.

13.6 Lebensmittel aus Bestand löschen

Die aufgebrauchten Produkte müssen durch den Benutzer entfernt werden. Daher ist es auf der Lebensmittel-Details-Seite möglich, Produkte auf „Aufgebraucht“ zu setzen. Diese Funktion kann nicht nur von den Produkten auf der Lebensmittelansicht-Seite durchgeführt werden, sondern auch auf den der Detailansicht-Seite. Wenn man diese Funktion mittels eines Klicks auf den dafür zugewiesen Button ausführt, wird das Produkt entfernt und somit auch nicht mehr angezeigt.

14 Resümee

Durch das Verfassen unserer Diplomarbeit haben wir erstmals einen Einblick bekommen, wie wissenschaftlich gearbeitet wird, was uns im späteren Studium oder Berufsleben sicher helfen wird.

Da wir mit der Planung, Implementierung und Programmierung der Arbeit schon in den Sommerferien begonnen haben, konnten wir die App nach den Herbstferien fertigstellen. Deshalb ist uns noch genügen Zeit für das Schreiben geblieben. Was uns außerdem bei der Umsetzung unserer Arbeit sehr geholfen hat, waren der ständige Kontakt, sowie die Rückmeldungen von unserem Betreuer und unserem Auftraggeber. Auch die Kommunikation hat sehr gut funktioniert.

15 Aufgabenverteilung

15.1 Julia Bodingbauer

- Kurzbeschreibung
- Abtract
- Stackholder
    - Lehranstalt
- Einleitung
    - Motivation
    - Vorgehensweise
        - Rollen und Ablauf
        - Projektmanagement
        - Projektorganisation
        - Funktionsbereiche der App
            - Push-Benachrichtigungen
- Grundlagen
    - Verwendete Technologien
        - Ubuntu
        - MySQL
        - SQL (Stuctured Query Language)
        - Retrofit
        - Apache Tomcat
- Implementierung
    - Backend
        - Server
            - Allgemeine Beschreibung
        - Datenbank
            - Allgemeine Beschreibung
            - Datenmodell
            - Datenkatalog
- Ergebnis
    - Funktionen aus Benutzersicht
        - Einloggen oder neuen Benutzer erstellen

15.2 Jana Dannhofer

- Stackholder
    - Betreuung
- Einleitung
    - Ist-Zustand / Hintergrund
- Grundlagen
    - Verwendete Technologien
        - Java-Technologie
        - Java (Programmiersprache)
        - Java Spring Boot
        - JPA
- Implementierung
    - Backend
        - REST-API
            - Allgemeines
            - Aufbau der REST-API
            - Statuscodes
            - Kommunikation per REST-API
            - Zugriff auf die Datenbank
            - Passwort Hashing
            - Authentifizierung
            - Verbindung mit dem Algorithmus
- Ergebnis
    - Funktionen auf Benutzersicht
        - Hinzufügen von Lebensmitteln ins Lager
        - Empfehlungen für kommende Einkäufe

15.3 Moritz Lechthaler

- Funktionsbereiche der App
    - Benutzerverwaltung
    - Lebensmittelverwaltung
    - Analyse des Benutzerverhalten
    - Empfehlung für Benutzer
- Planung und Realisierung
    - Design der Android-App
        - Login
        - Abgelaufen – Empfohlene Lebensmittel Seite
        - Home
- Implementierung
    - Frontend
        - Barcode
        - Login / Benutzerverwaltung
        - Registrierung
        - Navigationsleiste
        - Sonstige Implementierung
- Ergebnis
    - Funktionen aus Benutzersicht
        - Detailansicht der Lebensmittel
        - Übersicht über bald ablaufende / abgelaufene Lebensmittel
        - Lebensmittel aus Bestand löschen

15.4 Fabian Kopf

- Stackholder
    - Auftraggeber
- Einleitung
    - Relevanz hinsichtlich ökologischer, sozialer und ökonomischer Nachhaltigkeit
    - Ziele der Diplomarbeit
    - Leistungsumfang
- Grundlagen
    - Verwendete Entwicklungssysteme
        - Jetbrains Intellij IDEA
        - Jetbrains Andriod Studio
        - MySQL Workbench
    - Sonstige verwendete Software
- Planung und Realisierung
    - Meilensteine
    - Verantwortlichkeiten
    - Use Case Diagramm
- Implementierung
    - Algorithmus
        - Ablauf des Algorithmus
        - Kennzahlen
        - Zugriff auf Daten
        - Gewichtung der Lebensmittel
        - Analyse des Benutzerverhaltens
        - Empfehlung für Benutzer
        - Fehlerbehandlung

16 Anhang

16.1 Abnahmeformular

HTBLA Perg	**HTBLA Perg** Höhere Lehranstalt für Informatik	**Reife- und Diplomprüfung**

Abnahmeprotokoll	**HTBLA Perg**
Projektname : ***Smood – Smarte Lebensmittelverwaltungsapp***	**Projektnummer :**
Auftraggeber : *P@BS – EDV Absolventen und Förderer der HTL – Perg*	**Empfänger :** *Obmann Lukas Huber*

Das Werk „Smood" Version 1.0

basierend auf dem Vertrag vom: 13.4.2022

wurden am 7.2.2023 zur Abnahme angemeldet.

Zu dem Werk gehören folgende **Komponenten**:

- ☑ Dieses Protokoll
- ☑ Datenbank SQL-Files
- ☑ War-File von der Rest-API und dem Algorithmus
- ☑ App als .zip Datei
- ☑ .zip Datei vom Algorithmus und Rest-API

Das Werk „Smood" Version 1.0

basierend auf dem Vertrag vom: 13.4.2022

wurden am 7.2.2023 abgenommen und zur Umsetzung freigegeben.

Mit seiner Unterschrift bestätigt der Auftraggeber die Abnahme des Teilwerks.

Datum:	7.2.2023		
	Unterschrift AG		Unterschrift AN

16.2 Abbildungsverzeichnis

16.3 Literaturverzeichnis

adobe.com. (1. 1 2023). *adobe.com*. Von adobe.com: https://www.adobe.com/at/products/photoshop/landpb.html?gclid=Cj0KCQiAnsqdBhCGARIsAAyjYjQaYR2DizHnC73ATT2db6asXwaKYUOR2d-XDXYYuQXi1imdsaIlvaYaAlQkEALw_wcB&mv=search&sdid=LZ32SYVR&ef_id=Cj0KCQiAnsqdBhCGARIsAAyjYjQaYR2DizHnC73ATT2db6asXwaKYUOR2d-XDXYYuQXi1i abgerufen

appuntisoftware.it. (21. 12 2022). Von appuntisoftware.it: https://www.appuntisoftware.it/installiamo-e-configuriamo-apache-tomcat-7/ abgerufen

Astera. (16. 1 2023). Von https://www.astera.com/de/ abgerufen

Authgrear. (21. 01 2023). Von https://www.authgear.com/post/password-hashing-salting abgerufen

biteno.com. (21. 12 2022). Von biteno.com: https://www.biteno.com/was-ist-tomcat/ abgerufen

bluesource.at. (04. 01 2023). Von bluesource.at: https://www.bluesource.at/blog/detail/wissenswertes-ueber-push-notifications abgerufen

code.tutsplus.com. (04. 01 2023). Von code.tutsplus.com: https://code.tutsplus.com/de/tutorials/getting-started-with-retrofit-2--cms-27792 abgerufen

Coding, A. (27. Juli 2022). *How to Create DatePickerDialog in Android Studio | DatePickerDialog | Android Coding*. Von YouTube: https://www.youtube.com/watch?v=E1LSY3g-CtY&list=TLPQMjIwNzIwMjLZJXGZCwk6Yw&index=3 abgerufen

ComputerWeekly.de. (19. 12 2022). Von ComputerWeekly.de: https://www.computerweekly.com/de/definition/MySQL abgerufen

datarobot.com. (4. 1 2023). *datarobot.com*. Von datarobot.com: https://www.datarobot.com/wiki/overfitting/ abgerufen

de.wikipedia.org. (8. 3 2023). *de.wikipedia.org*. Von de.wikipedia.org: https://de.wikipedia.org/wiki/Adobe_Photoshop#/media/Datei:Adobe_Photoshop_CC_icon.svg abgerufen

design.ubuntu.com. (06. 12 2022). Von design.ubuntu.com: https://design.ubuntu.com/brand/ubuntu-logo/ abgerufen

dev.mysql.com. (19. 12 2022). Von dev.mysql.com: https://dev.mysql.com/ abgerufen

developer.android.com. (04. 01 2023). Von developer.android.com: https://developer.android.com/develop/ui/views/notifications/build-notification abgerufen

developer.android.com. (26. 02 2023). Von developer.android.com: https://developer.android.com/reference/androidx/work/PeriodicWorkRequest abgerufen

developer.android.com. (26. 02 2023). Von developer.android.com: https://developer.android.com/guide/background/persistent/getting-started/define-work abgerufen

Developer.com. (20. 12 2022). Von https://www.developer.com/wp-content/uploads/2021/09/Java-tutorials-300x200.jpg abgerufen

digitaleneuordnung.de. (29. 12 2022). Von digitaleneuordnung.de: https://digitaleneuordnung.de/blog/scrum-methode/ abgerufen

DigitalOcean. (27. 02 2023). *www.digitalocean.com*. Von www.digitalocean.com: https://www.digitalocean.com/community/tutorials/android-notification-pendingintent abgerufen

discord.com. (2. 1 2023). *discord.com*. Von discord: https://discord.com/ abgerufen

docs.spring.io. (15. 03 2023). Von docs.spring.io: https://docs.spring.io/spring-boot/docs/2.0.x/reference/html/using-boot-using-springbootapplication-annotation.html abgerufen

dr-datenschutz.de. (3. 1 2023). *dr-datenschutz.de*. Von dr-datenschutz.de: https://www.dr-datenschutz.de/was-ist-ein-algorithmus-definition-und-beispiele/ abgerufen

engineering.creativesociety. (21. 12 2022). Von engineering.creativesociety: https://engineering.creativesociety.mx/retrofit-cliente-http-para-android/ abgerufen

Flow, C. i. (31. August 2022). *How to Change the App Icon in Android Studio (With Adaptive Icons)*. Von YouTube: https://www.youtube.com/watch?v=ts98gL1JCQU abgerufen

Flow, C. i. (28. Juli 2022). *Retrofit Tutorial Part 1 - SIMPLE GET REQUEST - Android Studio Tutorial*. Von YouTube: https://youtu.be/4JGvDUlfk7Y abgerufen

Flow, C. i. (12. August 2022). *Retrofit Tutorial Part 3 - POST REQUEST & FORM URLENCODED - Android Studio Tutorial*. Von YouTube: https://www.youtube.com/watch?v=GP5OyYDu_mU&list=PLrnPJCHvNZuCbuD3xpfKzQWOj3AXybSaM&index=4 abgerufen

Flow, C. i. (20. September 2022). *Retrofit Tutorial Part 4 - PUT, PATCH & DELETE REQUEST - Android Studio Tutorial*. Von YouTube: https://youtu.be/xKEFGsMUG8s abgerufen

Foxandroid. (18. Juli 2022). *Bottom Navigation Bar - Android Studio | Fragments | Java | 2022*. Von https://www.youtube.com/: https://www.youtube.com/watch?v=Bb8SgfI4Cm4 abgerufen

git-scm.com. (2. 1 2023). *git-scm.com*. Von git-scm.com: https://git-scm.com/ abgerufen

global2000.at. (4. 1 2023). *global2000.at*. Von global2000.at: https://www.global2000.at/lebensmittelverschwendung#:~:text=Rund%20ein%20Drittel%20der%20Lebensmittel,die%20jeden%20Tag%20verloren%20gehen. abgerufen

Google Developers. (7. Januar 2023). *<activity>*. Von https://developer.android.com/docs: https://developer.android.com/guide/topics/manifest/activity-element abgerufen

Google Developers. (16. Januar 2023). *<uses-permission>*. Von https://developer.android.com/: https://developer.android.com/guide/topics/manifest/uses-permission-element abgerufen

Google Developers. (17. Januar 2023). *Create app icons with Image Asset Studio*. Von https://developer.android.com/: https://developer.android.com/studio/write/image-asset-studio abgerufen

Goolge Developers. (25. Februar 2023). *Context*. Von https://developer.android.com/: https://developer.android.com/reference/android/content/Context abgerufen

heise.de. (06. 12 2022). Von heise.de: https://www.heise.de/download/product/ubuntu-22040 abgerufen

Heute.at. (08. 03 2023). Von Heute.at: https://www.heute.at/s/so-vermeiden-sie-lebensmittelverschwendung-100180485 abgerufen

HTL-Perg Lehrerteam. (08. 03 2023). Von HTL-Perg Lehrerteam: https://www.htl-perg.ac.at/organisation/lehrerteam abgerufen

htl-perg.ac.at. (03. 01 2023). Von htl-perg.ac.at: https://www.htl-perg.ac.at/ abgerufen

IONOS. (20. 12 2022). Von https://www.ionos.at/digitalguide/websites/web-entwicklung/spring-boot-tutorial/ abgerufen

ionos.at. (04. 01 2023). Von ionos.at: https://www.ionos.at/digitalguide/server/knowhow/webserver-definition-hintergruende-software-tipps/ abgerufen

is3-ssl.mzstatic.com. (03. 08 2023). Von is3-ssl.mzstatic.com: https://is3-ssl.mzstatic.com/image/thumb/Purple112/v4/68/c1/2b/68c12bed-4269-faac-2776-104f3c20c650/AppIcon-0-0-1x_U007emarketing-0-0-0-7-0-0-sRGB-0-0-0-GLES2_U002c0-512MB-85-220-0-0.png/1200x630wa.png abgerufen

jangirkaran17. (23. Juli 2022). *how-to-implement-bottom-navigation-with-activities-in-an-droid*. Von https://www.geeksforgeeks.org: https://www.geeksforgeeks.org/how-to-implement-bottom-navigation-with-activities-in-android/ abgerufen

Java. (20. 12 2022). Von https://www.java.com/de/download/help/whatis_java.html abgerufen

Java Guides. (20. 12 2022). Von https://4.bp.blogspot.com/-ou-a_Aa1t7A/W6IhNc3Q0gI/AAAAAAAAD6Y/pwh44arKiuM_NBqB1H7Pz4-7QhUxAgZkACLcBGAs/s1600/spring-boot-logo.png abgerufen

JavaPoint. (19. 12 2022). Von https://www.javatpoint.com/jpa-tutorial abgerufen

Java-Tutorial. (20. 12 2022). Von https://www.java-tutorial.org/java-eigenschaften.html abgerufen

journeyapps. (23. Juli 2022). *zxing-android-embedded*. Von https://github.com/: https://github.com/journeyapps/zxing-android-embedded abgerufen

Logosmarken. (08. 03 2023). Von logosmarken: https://logosmarken.com/wp-content/uploads/2021/03/Trello-Logo.png abgerufen

Medium. (20. 12 2022). Von https://miro.medium.com/max/828/1*jyZu40J1N3MPJwFF66MVSg.webp abgerufen

Microsoft.com. (2. 1 2023). *Microsoft.com*. Von Microsoft.com: https://www.microsoft.com/de-at/microsoft-365/business/compare-all-microsoft-365-business-products-b?&ef_id=Cj0KCQiAnsqdBhCGARIsAAyjYjSw0HX9Yp0BLtmruNVMGVz3w_FiZAw3PAYAfSKt7RvdbpDz-

86vClIaAvSxEALw_wcB:G:s&OCID=AIDcmm409lj8ne_SEM_Cj0KCQiAnsqdBhCGARIsAAyjYj abgerufen

mysql.com. (20. 12 2022). *mysql.com*. Von mysql.com: https://www.mysql.com/products/workbench/ abgerufen

net4energy.com. (3. 1 2023). *net4energy.com*. Von net4energy.com: https://www.net4energy.com/de-de/smart-living/oekonomische-nachhaltigkeit abgerufen

Özdil, E. (7. Januar 2023). *Barcode*. Von https://www.weclapp.com/de/lexikon/: https://www.weclapp.com/de/lexikon/barcode/ abgerufen

Pabs.at. (2. 1 2023). *pabs*. Von pabs: https://pabs.at/aboutus.html abgerufen

Postman. (8. 3 2023). *www.testautomatisierung.org*. Von www.testautomatisierung.org: (https://www.testautomatisierung.org/wp-content/uploads/postman-300x183.jpg) abgerufen

postman.com. (2. 1 2023). *postman.com*. Von postman.com: https://www.postman.com/ abgerufen

Prof. Dipl.-Ing. Felix Schwab, W. P.-M. (2013). *Systemplanung und Projektentwicklung.* Wien: Hölzel Verlag G.m.b.H.

Prof. Ing. Patrick Praher, M. B. (3. 1 2023). *moodle.htl-perg.ac.at*. Von moodle.htl-perg.ac.at: https://moodle.htl-perg.ac.at/moodle/pluginfile.php/50709/mod_resource/content/0/DSAI_IV_ML_KNN.pdf abgerufen

REST API Tutorial. (30. 01 2023). Von https://restfulapi.net/ abgerufen

Restful Api Statuscodes . (16. 1 2023). Von https://restfulapi.net/http-status-codes/ abgerufen

Restfulapi - Http Methods. (01. 03 2023). Von Restfulapi - Http Methods: https://restfulapi.net/http-methods/ abgerufen

Rieke, C. (2. 11 2022). *URL Kodierung von Sonderzeichen*. Von https://www.key-shortcut.com/en/: https://www.key-shortcut.com/zeichentabellen/ascii-url-kodierung abgerufen

Spring. (19. 12 2022). Von https://spring.io/guides/gs/accessing-data-jpa/ abgerufen

Spring Boot. (20. 12 2022). Von https://spring.io/projects/spring-boot abgerufen

Stack Exchange Inc. (30. Juli 2022). *Stackoverflow*. Von https://stackoverflow.com/ abgerufen

stackfield.com. (2. 1 2023). *stackfield.com*. Von stackfield.com: https://www.stackfield.com/de/nr1-trello-alternative?adv=20181113&gclid=Cj0KCQiAnsqdBhCGARIsAAyjYjQWztKhIiQLhwBGTNGewZxoPpijlhXjTsn1gRKvi7DiBlUYUeyiVZUaArzMEALw_wcB abgerufen

Supertokens. (13. 2 2023). Von Supertokens: https://supertokens.com/blog/password-hashing-salting abgerufen

techopedia. (20. 12 2022). *techopedia*. Von techopedia: https://www.techopedia.com/definition/7755/intellij-idea abgerufen

techtarget. (20. 12 2022). *techtarget*. Von techtarget: https://www.techtarget.com/searchmobilecomputing/definition/Android-Studio abgerufen

topcoder.com. (19. 12 2022). Von topcoder.com: https://www.topcoder.com/thrive/articles/retrofit-library-in-android?utm_source=thrive&utm_campaign=thrive-feed&utm_medium=rss-feed abgerufen

Tuto, E. (20. Juli 2022). *BottomNavigationView with Fragments | Android Studio Tutorial | 2022*. Von https://www.youtube.com/: https://www.youtube.com/watch?v=OV25x3a55pk abgerufen

Tutorial, C. (23. Juli 2022). *Implement Barcode QR Scanner in Android Studio Barcode Reader | Cambo Tutorial*. Von https://www.youtube.com/: https://www.youtube.com/watch?v=jtT60yFPeII abgerufen

twilio.com. (04. 01 2023). Von twilio.com: https://www.twilio.com/docs/glossary/what-is-push-notification abgerufen

upload.wikimedia.org. (8. 2 2023). *upload.wikimedia.org*. Von upload.wikimedia.org: https://upload.wikimedia.org/wikipedia/commons/thumb/e/ abgerufen

WWF. (20. 1 2023). Von https://www.wwf.at/artikel/lebensmittelverschwendung-im-haushalt/ abgerufen

WWF Lebensmittelabfälle. (22. 03 2023). Von WWF Lebensmittelabfälle: https://www.wwf.at/wp-content/cms_documents/studie_lebensmittelabfaelle-in-oesterreichischen-haushalten---status-quo.pdf abgerufen

www.edureka.co. (1. 3 2023). *www.edureka.co*. Von www.edureka.co: https://www.edureka.co/blog/wp-content/uploads/2018/03/AI-vs-ML-vs-Deep-Learning.png abgerufen

www.vautron.de. (02. 01 2023). Von www.vautron.de: https://www.vautron.de/blog/welche-vorteile-bietet-ein-linux-server abgerufen

YouTube: SQL Einführung 1. (19. 12 2022). Von YouTube: SQL Einführung 1: https://www.youtube.com/watch?v=2goPVJOvJVY&list=PLgZuSc7xewdc3oOyvYkjExbhoO8jUxvV3&index=1 abgerufen

16.4 Tabellenverzeichnis