



# SNACK TRACK

## Audit 3

Sebastian Fuhrmann & Angelina Maurer

22.01.2024

1



# Überarbeitung Artefakte

## Audit 1

22.01.2024

2

# #Projektplan

2023			2024		
Oktöber	November	Dezember	Januar	Februar	März
1 So 1 Mi	1 Fr 2 Sa	1 Mo 3 Mi	1 Do 4 Fr	1 Do 5 Fr	1 Fr
2 Mo 4 Do	Artefakte bearbeiten	2 So 3 Di	2 Di 3 Mi	2 Fr 3 Sa	funktionaler Prototyp
3 Di Tag der offenen		3 So Advent	3 Mi Modellierung der Anwendungslogik	3 So 4 So	Poster erstellen
4 Mi 5 Do	4 Sa 5 Di	4 Mo Open Space	4 Do 5 Fr	4 So 5 Mo	
5 Do 6 Fr	6 Mi Feedback einarbeiten	5 Di 6 Mi	5 Fr 6 Sa	5 Mo Open Space	6 Di
6 Fr 7 Sa	Audit vorbereiten	6 Mi Audit +	6 Sa neige zur reise	6 Di Poster bearbeiten	6 Mi
7 Sa 8 So	7 Di 8 Mi	7 Do 9 So	7 So 8 Fr	7 Mi 9 Sa	
8 So 9 Mo	Ideenfindung	9 Do 10 Fr	9 Di 10 Mi	9 Sa 10 So	Karneval
10 Di 11 Mi		11 Sa 11 Mo 2. Audit	11 Do Durchgeführte PoCs	11 So 12 Mo	11 Mo
11 Mi 12 Do		12 Di 13 Fr	12 Fr 13 Mi	12 Mo 13 Di	
12 Do 13 Fr	13 Mo 4. Audit	13 Mi Erster vertikaler Rapid	13 Sa 14 So	13 Mi 14 Mi	
13 Fr 14 Sa	Projektkrisiken	14 Do Prototype	14 Do 15 Fr	14 Do 15 Do	Fest und kreativ
14 Sa 15 So	15 Mi (Architekturmodell, Kommunikation)	15 Fr PoC APIs & DB	15 Fr 16 Mi	15 Fr 16 Fr	reflektiertes Prozessseminar
15 So 16 Mo	16 Do Exposé	16 Do PoC APIs & DB	16 Mi 17 So	16 Sa 17 So	
16 Mo 17 Di	17 Fr 18 Sa	17 Fr 18 Mo Open Space	17 Mi Feedback einarbeiten	17 So 18 So	
17 Di 18 Mi	18 Do Artefakte festlegen	18 Mo Open Space	18 Do Audit vorbereiten	18 Mo 19 So	18 Mo
18 Mi 19 Do	19 So 20 Fr	19 So PoC (API + DB)	19 Do 20 Mi	19 Di 20 Di	
19 Do 20 Fr	20 Mo Open Space	20 Mi Iterierte Modellierungen und -Begründungen	19 Fr 20 Sa	19 Fr 20 Fr	Feedback
20 Fr 21 Sa	21 Di 22 So	21 Do 22 Fr	20 Sa 21 So	20 Mi 21 Mi	20 Mi
21 Sa 22 So	22 Mi 23 Do	21 Do 22 Fr	21 So 22 Do	21 Do 22 Do	
22 So 23 Mo	23 Do PoC (API + DB)	22 Fr 23 So	22 Do 23 Fr	22 Do 23 Fr	Audit vorbereiten
23 Mo 24 Di	24 Fr 25 Mi	23 So 24 So	23 Mi 24 Fr	23 Fr 24 Fr	Poster
24 Di 25 Mi	Exposé +	24 So neigereise	24 Mi funktionaler Prototyp	24 So 25 So	
25 Mi 26 Do	26 So Artefakte bearbeiten	25 Mo 1. Audit	25 Do 26 Fr	25 So 26 So	
26 Do 27 Fr	27 Mo 28 Sa	26 Mi 27 Mi	25 Do 26 Fr	25 Mo 26 Mi	
27 Fr 28 Sa	Projektkrisiken	27 Mi Iterierte Modellierungen und -Begründungen	27 Fr 28 So	27 Mi 28 Mi	
28 Sa 29 So	28 Di Design Prototype	28 Do 29 Fr	28 So 29 Do	28 Do 29 Fr	
29 So Ende der Kompat	29 Mi 30 Do	29 Fr 30 Sa	29 Do 30 So	29 Do 30 So	
30 Mo 31 Di	30 Do Retur- brachte	30 Sa 31 So	30 So 31 Mi	30 So 31 So	
Blockveranstaltungen / Projektwochen			Feiertage		
Projektwoche HIP 27.11.2023 - 01.12.2023			Weihnachten		
Blockveranstaltungen 02.01.2023 - 05.01.2023			Karneval		
Blockveranstaltungen 29.01.2023 - 02.02.2023			Audits		
			1. Audit 13.11.2023		
			2. Audit 11.12.2023		
			3. Audit 22.01.2024		
			4. Audit 26.02.2024		

22.01.2024



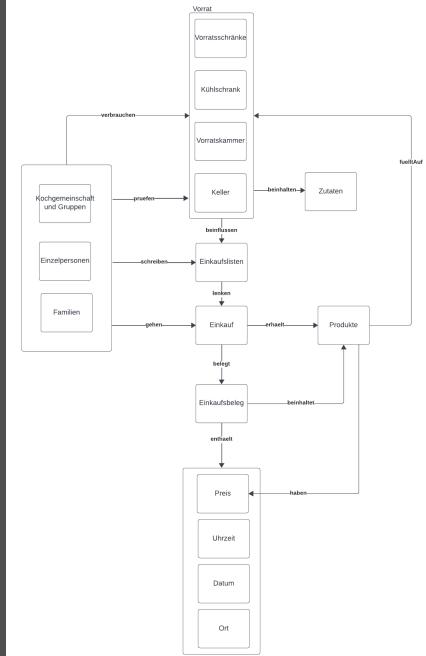
## Anpassung des Projektplans aufgrund der Verschiebung des Posterslams (Email

- Ermöglicht mehr Zeit für die durchgeführten POCs und der Modellierung der Anwendungslogik
- Die Erstellung des Posters wurde nun zwischen dem 3 und 4 Audit eingeplant, wodurch jedoch die Zeit für die Erstellung des funktionalen Prototyps ein wenig eingeschränkt werden musste

# #Deskriptives Domänenmodell

22.01.2024

4

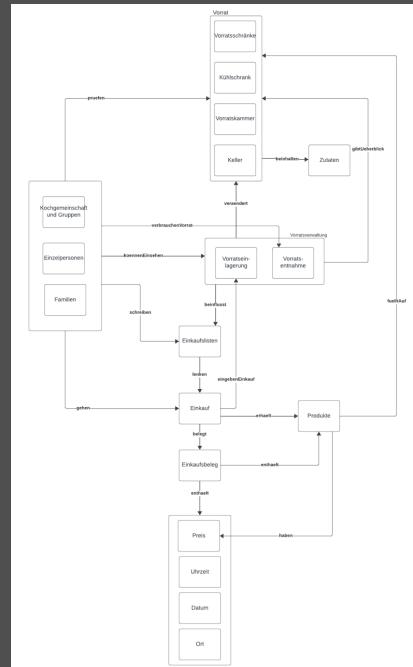


Erweiterung des deskriptiven Domänenmodells um folgende Bestandteile:

- Gruppierung **Vorrat**: um den gesamten Lebensmittelvorrat zu verwalten (nicht nur Kühlzettel)
- **Einkaufsbeleg**: Um Preis, Uhrzeit, Datum und Ort des Einkaufs festzuhalten
- **Produkte**: die in Verbindung mit **Einkaufsbeleg** und **Einkauf** steht. Diese bezeichnen die erworbenen Produkte, welche anschließend den Vorrat auffüllen

# #Präskriptives Domänenmodell

22.01.2024



5

Erweiterung des deskriptiven Domänenmodells um folgende Bestandteile:

- Gruppierung **Vorrat**: um den gesamten Lebensmittelvorrat zu verwalten (nicht nur Kühlschrank)
- **Einkaufsbeleg**: Um Preis, Uhrzeit, Datum und Ort des Einkaufs festzuhalten
- Gruppierung **Vorratsverwaltung**: Vorratsverwaltung aufgeteilt in **Vorratseingabe** und **Vorratsentnahme**
- **Produkte**: die in Verbindung mit **Einkaufsbeleg** und **Einkauf** steht. Diese bezeichnen die erworbenen Produkte, welche anschließend den Vorrat auffüllen
- **Verbindungen Vorratsverwaltung**: Verändern den Vorrat, beeinflussen die Einkaufsliste und gibt einen Überblick über die vorhandenen Vorräte

# #Stakeholder

Bezeichnung	Bezug zum System	Objektbereich	Erfordernis/ Erwartung
Einzelpersonen	Anspruch	System	Lebensmittel effizient verwalten
Familien	Anspruch	System	Zeitgenaue Lebensmittelverwaltung
Bewohner einer Wohngemeinschaft	Anspruch	System	Getrenntes Verwalten von Lebensmitteln
Koch-Communitys	Anspruch	System	Gemeinsames Verwalten der Lebensmittel
Lebensmittel- und Kochblogs	Anspruch	System	Leser dazu einzuladen, ihre Lebensmittel effektiv zu verwalten
Lebensmittelhersteller und -händler	Interesse	System	Benutzern Produktinformationen und Sonderangebote bereitzustellen. Dies könnte dazu beitragen, Verkäufe und Marketingmöglichkeiten zu steigern
Gesundheits- und Ernährungsexperten	Anspruch	System	Gesunde Rezepte und Ernährungsempfehlungen bereitzustellen
Technologieanbieter	Interesse	System	Unternehmen, die Technologien für Lebensmittelerkennung und -verfolgung entwickeln, könnten von der Integration ihrer Lösungen in die Anwendung profitieren
Einzelhandelsgeschäfte	Anspruch	System	Benutzern ermöglichen, Einkaufslisten direkt in ihren Geschäften zu erstellen, basierend auf den in der Anwendung verfügbaren Rezepten und Zutaten

22.01.2024

6

Für die Aufgabenstellung wurde mit Rücksicht auf die vorher gegangene Recherche und Domänenanalyse eine Stakeholder Tabelle mit Bezeichnung der Stakeholder, deren Bezug zum System, dem Objektbereich und Erfordernissen beziehungsweise Erwartungen angefertigt (siehe Tabelle). Die daraus resultierenden Ergebnisse wurden für den weiteren Verlauf des Projektes, insbesondere in der Anforderungsermittlung, weiterverwendet.

Nachdem Zusammentragen der Stakeholder in einer Tabelle, wurden sie in drei verschiedene Kategorien unterteilt. Die erste Kategorie bilden die primären Stakeholder, welche das System tatsächlich benutzen und somit die Endbenutzer darstellen. Die zweite Kategorie beinhaltet sekundäre Stakeholder, die das System nicht direkt benutzen, jedoch einen Input für das System bereitstellen oder einen Output davon erhalten. Als letztes wurden die Personen, die keine der beiden Kategorien erfüllen, aber direkt vom Erfolg oder Misserfolg des Systems betroffen sind in die Kategorie der tertiären Stakeholder zusammengefasst.

Die Stakeholder Tabelle wurde in Primäre, sekundäre und tertiäre

Stakeholder gruppiert.

Hinweis: Bewohner einer WG als Teil der Benutzer Stakeholder Gruppe separat aufgeführt, aufgrund von unterschiedlichen bzw. erweiterten Erfordernissen.

# #Persona



 <p><b>Peter Müller</b></p>	<table border="1"> <tbody> <tr> <td>Alter</td><td>55</td></tr> <tr> <td>Ort</td><td>Berlin</td></tr> <tr> <td>Technologien</td><td>Grundkenntnisse Computer und Mobile Anwendungen</td></tr> <tr> <td>Einschränkungen</td><td>Keine</td></tr> <tr> <td>Hobbies</td><td>Schwimmen, Reisen</td></tr> <tr> <td>Ziele</td><td>Bessere Planung der Lebensmitteleinkäufe</td></tr> </tbody> </table> <p>Peter Müller möchte zusammen mit seiner Ehefrau die Einkäufe für die Familie besser planen. Da beide Parteien individuell einkaufen gehen, werden oftmals Lebensmittel doppelt gekauft. Dies ist zum einen eine finanzielle Belastung, zum anderen ärgern sich die beiden viele Lebensmittel wegwirken zu müssen.</p>	Alter	55	Ort	Berlin	Technologien	Grundkenntnisse Computer und Mobile Anwendungen	Einschränkungen	Keine	Hobbies	Schwimmen, Reisen	Ziele	Bessere Planung der Lebensmitteleinkäufe						
Alter	55																		
Ort	Berlin																		
Technologien	Grundkenntnisse Computer und Mobile Anwendungen																		
Einschränkungen	Keine																		
Hobbies	Schwimmen, Reisen																		
Ziele	Bessere Planung der Lebensmitteleinkäufe																		
 <p><b>Luis Baumgartner</b></p>	<table border="1"> <tbody> <tr> <td>Alter</td> <td>20</td> </tr> <tr> <td>Jobtitel</td> <td>Ausbildender</td> </tr> <tr> <td>Bildungsgrad</td> <td>Mittlere Reife</td> </tr> <tr> <td>Ort</td> <td>Lindau</td> </tr> <tr> <td>Technologien</td> <td>Sehr gute Kenntnisse im Bereich Computer und Mobile Anwendungen</td> </tr> <tr> <td>Einschränkungen</td> <td>keine</td> </tr> <tr> <td>Familienstand</td> <td>ledig</td> </tr> <tr> <td>Hobbies</td> <td>Musik hören, Freunde besuchen</td> </tr> <tr> <td>Ziele</td> <td>Sicherer Job, Lebensmittelverschwendungen reduzieren</td> </tr> </tbody> </table> <p>Luis hat seine Schule erfolgreich abgeschlossen und hat eine Ausbildung zum Industrie Mechaniker begonnen. Es ist ihm wichtig, einen guten Job in einem sicheren Umfeld zu haben. Luis möchte seine Lebensmittelverschwendungen einschränken. Dies fällt ihm jedoch schwer, da er direkt nach dem Arbeiten einkaufen geht und daher vorher nicht in den Kühlschrank schauen kann, um zu prüfen was er bereits zu Hause hat.</p>	Alter	20	Jobtitel	Ausbildender	Bildungsgrad	Mittlere Reife	Ort	Lindau	Technologien	Sehr gute Kenntnisse im Bereich Computer und Mobile Anwendungen	Einschränkungen	keine	Familienstand	ledig	Hobbies	Musik hören, Freunde besuchen	Ziele	Sicherer Job, Lebensmittelverschwendungen reduzieren
Alter	20																		
Jobtitel	Ausbildender																		
Bildungsgrad	Mittlere Reife																		
Ort	Lindau																		
Technologien	Sehr gute Kenntnisse im Bereich Computer und Mobile Anwendungen																		
Einschränkungen	keine																		
Familienstand	ledig																		
Hobbies	Musik hören, Freunde besuchen																		
Ziele	Sicherer Job, Lebensmittelverschwendungen reduzieren																		

22.01.2024

7

Um zielgerichtet zu kommunizieren, wurden verschiedene Personas erstellt, um die Überlegungen welche Eigenschaften der mögliche Nutzer haben könnte festzuhalten. Dafür wurde ein fiktiver Personen-Steckbrief erstellt, der bestimmte Merkmale aufweist, die für die Interaktion mit den publizierten Inhalten von Bedeutung sind. Für dieses Projekt wurden insgesamt drei verschiedene Personas erstellt.

## Peter Müller:

- Familievater der u.a. Einkaufen geht
- Will/ Muss aus grünen und finanziellen Gründen auf das Geld achten
- Kommunikation bezüglich Einkaufen funktioniert nicht immer
- Ärgert sich wenn Lebensmittel unnötig gekauft werden und dann im Müllbeimer landen

## Anna Sommer:

- Studentin in einer WG
- Ärgert sich wenn WG Mitbewohner ihre Lebensmittel verbrauchen
- Haben oft Streit aus diesem Grund in der WG
- Wünscht sich weniger Streit

- Versuchen die Lebensmittel so gut wie möglich zu trennen, funktioniert jedoch nicht

Luis Baumgartner:

- Azubi in erster eigener Wohnung
- Geht nach dem Arbeiten einkaufen und kauft daher oft Lebensmittel die er eigentlich noch zu Hause hat
- Vergisst was er bereits zu Hause an Lebensmitteln besitzt
- Möchte aus umwelttechnischen Gründen seine Lebensmittelverschwendungen reduzieren



## #Task Szenario

Maria plant einen Lebensmitteleinkauf für ein neues Rezept, welches sie ausprobieren möchte. Sie ist sich jedoch unsicher über den aktuellen Inhalt ihres Kühlschranks. Ohne genaue Informationen darüber, welche Lebensmittel bereits vorhanden sind, besteht die Gefahr, dass sie unnötigerweise Produkte doppelt kauft. Nach dem Einkauf stellt sie fest, dass sie einige Artikel bereits zu Hause hat, andere wichtige Produkte für das Rezept jedoch vergessen hat. Sie muss daher Zeit und Ressourcen verschwenden, um aufgrund der fehlenden Produkte erneut einkaufen zu gehen und um doppelt gekaufte Produkte zu verbrauchen bevor diese verderben.



## #Use Szenario

Maria plant einen Lebensmitteleinkauf für ein neues Rezept, welches sie ausprobieren möchte. Maria möchte ihren Lebensmitteleinkauf optimieren, indem sie ihr Online-Inventar nutzt. Vor und während dem Einkaufen schaut sie in ihre digitale Liste, die alle vorhandenen Lebensmittel im Kühlschrank, im Gefrierschrank und in den Schränken auflistet. Sie kann schnell sehen, welche Artikel noch vorrätig sind. Aufgrund dieser Informationen kann Maria gezielt nur die Produkte kaufen, die sie tatsächlich benötigt. Nach dem Einkauf räumt sie die neuen Lebensmittel effizient in die Schränke ein und hält in ihrem online Inventar fest welche Produkte sie erworben hat und welche sie für das Rezept verbraucht hat, ohne sich Gedanken über bereits vorhandene Produkte machen zu müssen. Durch die Nutzung ihres Online-Inventars spart Maria Zeit, Geld und vermeidet unnötige Verschwendungen.

22.01.2024

9

## Claim Analysis

### Task Scenario:

Im Task-Szenario plant Maria einen Lebensmitteleinkauf für ein neues Rezept, ohne genaue Kenntnisse über den aktuellen Inhalt ihres Kühlschranks. Dies führt dazu, dass sie einige Artikel doppelt kauft und andere, für das Rezept wichtige Produkte vergisst. Die Konsequenz ist ein Verlust von Zeit und Ressourcen, da Maria erneut einkaufen gehen muss und doppelt gekaufte

Produkte vor dem Verderben verwenden muss.

### Use Scenario:

Im Use-Szenario plant Maria ebenfalls einen Lebensmitteleinkauf für das gleiche Rezept, nutzt jedoch ihr Online-Inventar, um den Prozess zu optimieren. Durch die digitale Liste behält sie den Überblick über vorhandene Lebensmittel und kann gezielt nur die benötigten Produkte kaufen. Nach dem Einkauf dokumentiert sie die erworbenen und verbrauchten Produkte in ihrem Online-Inventar, was Zeit, Geld und Verschwendungen reduziert.

### **Claim Analysis:**

Der Vergleich der beiden Szenarien zeigt, dass die Nutzung eines Online-Inventars im Use-Szenario entscheidende Vorteile bietet. Maria kann ihre Einkaufsplanung verbessern, da sie jederzeit Zugriff auf eine aktuelle Übersicht über ihre Vorräte hat. Durch die Dokumentation im Online-Inventar vermeidet sie doppelte Einkäufe und kann effizienter mit den erworbenen Lebensmitteln umgehen. Der Claim lautet daher: Die Integration eines Online-Inventars in den

Lebensmitteleinkaufsprozess ermöglicht eine optimierte Planung, spart Zeit und Geld, minimiert Verschwendungen und verbessert das Gesamterlebnis bei der Zubereitung neuer Rezepte.

# #Erfordernisse



1. Der Benutzer muss die Möglichkeit haben, Lebensmittel nach dem Einkauf ohne Mehraufwand zu erfassen.
2. Der Benutzer möchte eine benutzerfreundliche Schnittstelle, um neue Lebensmittel hinzuzufügen.
3. Der Benutzer muss die Möglichkeit haben, Lebensmittel nach dem Verbrauch zu erfassen.
4. Der Benutzer muss eine Funktion haben, um den aktuellen Lagerbestand zu aktualisieren.
5. Ein Bewohner einer Wohngemeinschaft möchte die Möglichkeit haben, Lebensmittel individuell zuzuordnen, um eine klare Trennung der Bestände zu gewährleisten.
6. Jeder Benutzer möchte eine klare Anzeige haben, welche Lebensmittel ihm gehören.
7. Der Benutzer muss eine Übersicht über seinen Vorrat haben.
8. Der Benutzer muss seine Lebensmittel ohne Mehraufwand finden und verwalten können.
9. Die Benutzer möchten die Verwaltung der Lebensmittel auf verschiedene Plattformen durchführen können.

# #Funktionale Anforderungen



1. Um die Verwaltung der Lebensmittel durch die Anwendung ohne Mehraufwand zu ermöglichen, muss der Benutzer eine einfache und intuitive Benutzeroberfläche zur Verfügung gestellt bekommen.
2. Um neue Lebensmittel hinzuzufügen, muss der Benutzer eine benutzerfreundliche Schnittstelle dazu bereitgestellt bekommen.
3. Um Lebensmittel schnell und effizient zu erfassen, sollte die Anwendung eine dazu passende Funktion bereitstellen.
4. Um Lebensmittel nach dem Verbrauch zu erfassen, muss die Anwendung eine dazu passende Funktion bereitstellen.
5. Um den aktuellen Lagerbestand zu erfahren, muss die Anwendung die Vorräte anzeigen können.
6. damit der Benutzer Lebensmittel unabhängig vom verwendeten Gerät oder Betriebssystem erfassen kann, muss die Anwendung plattformunabhängig sein.

# #Qualitative Anforderungen



1. Um eine zeitlose Funktionalität zu bieten, muss die Software fähig sein, unabhängig von technologischen Entwicklungen zu funktionieren.
2. Um die Lebensmittel zu erfassen, muss die Software fähig sein, auch in der Zukunft effektiv und zuverlässig funktionieren.
3. Um das Erfassen der Lebensmittel zu optimieren, muss die Software fähig sein, dies ohne zeitlichen Mehraufwand zu vollziehen.
4. Um dem Benutzer die Anwendung zu erleichtern, wird die Software dem Benutzer eine selbsterklärende Benutzeroberfläche bieten.
5. Um dem Benutzer die Anwendung zu erleichtern, wird die Software dem Benutzer die Möglichkeit bieten, eine Beschreibung/ Anleitung anzusehen.
6. Um eine effektive Lebensmittelverwaltung zu gewähren, muss die Software dem Benutzer die Möglichkeit bieten, alle Arten von Lebensmitteln zu erfassen, unabhängig von Verpackungstypen, Kategorien oder sonstigen Einschränkungen.
7. Um eine breite Benutzerreichweite zu generieren, muss die Software fähig sein, Barrierefreiheit zu gewähren.

## #Rechtlich-vertragliche Anforderungen



1. Die Software muss während der gesamten Nutzungsdauer den Datenschutzgesetzen der Bundesrepublik Deutschland entsprechen. Dies umfasst die sichere Speicherung und Verarbeitung personenbezogener Daten sowie die Einhaltung aller relevanten Datenschutzbestimmungen.
2. Die Anwendung sollte kontinuierlich aktualisiert werden, solange der Benutzer das System verwendet. Dies beinhaltet Sicherheitsupdates, Fehlerkorrekturen und mögliche Erweiterungen der Funktionalität.
3. Während der Nutzungsdauer muss die Software ausreichende Sicherheitsmechanismen implementieren, um vor Datenmissbrauch zu schützen. Dies umfasst sichere Authentifizierung, Verschlüsselungstechnologien und Maßnahmen zur Verhinderung unbefugten Zugriffs.
4. Die Anwendung muss sicherstellen, dass die bereitgestellten Informationen zu Lebensmitteln wissenschaftlich korrekt sind und allen relevanten rechtlichen Bestimmungen entsprechen. Dies betrifft insbesondere Angaben zu Nährwerten, Verfallsdaten und anderen relevanten Aspekten.
5. Die Kommunikation innerhalb der Anwendung, einschließlich Benutzerhinweise und Anleitungen, muss fachlich korrekt und verständlich sein. Insbesondere sollte darauf geachtet werden, dass rechtliche Begriffe und Anweisungen in einer klaren und präzisen Sprache formuliert sind.



# Überarbeitung Artefakte

## Audit 2

22.01.2024

14

# Style Guide



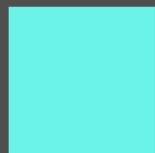
Sebastian Fuhrmann  
Angelina Maurer

22.01.2024

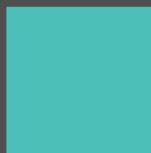
15

Einheitliches CD, welches bereits als CSS Files festgehalten wurden um ein einheitliches Erscheinungsbild zu erhalten

# #Colors



**accent**  
#69F3E9  
R 105  
G 243  
B 233



**dark accent**  
#4CC0B8  
R 76  
G 192  
B 184



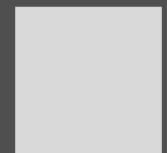
**darkest**  
#1D4745  
R 29  
G 71  
B 69



**dark**  
#2D706F  
R 45  
G 112  
B 111



**lightdark**  
#3D9993  
R 61  
G 153  
B 147



**light**  
#D9D9D9  
R 217  
G 217  
B 217

22.01.2024

16

Color Auswahl:

Farbharmonie: Schattierungen im RGB Farbmodus

# #Fonts



## Aa

Font Family Helvetica Neue LT Std

Headline 1 45 Light 48px

I'm a Headline

Paragraph 1 45 Light 36px

Corem ipsum dolor sit amet,  
consectetur adipiscing elit. Nunc  
vulputate libero et velit interdum.

Paragraph 2 45 Light 24px

Jorem ipsum dolor sit amet, consectetur adipiscing elit.  
Nunc vulputate libero et velit interdum, ac aliquet odio  
mattis.

22.01.2024

17

**1.Lesbarkeit:** Helvetica Neue LT Pro 45 Light ist eine leichte Variante der Helvetica-Schriftfamilie.

**2.Stil und Design:** klaren und modernen Stil. Passt zur Markenidentität und dem Design der Webanwendung. Es ist wichtig, dass die Schriftart das gewünschte visuelle Erscheinungsbild der Anwendung unterstützt.

## #Logo



SNACK TRACK



SNACK TRACK

22.01.2024

18

Logo in zwei verschiedenen Varianten (insbesondere für Anwendungen im Light- und Darkmode wichtig)

Logo kann mit und ohne Schriftzug stehen

Abstraktere Visualisierung von Obst und Gemüse in einem Korb

# #DesignPrototype



The wireframe displays the 'Inventar' (Inventory) screen of the Snack Track application. The desktop version shows a table with five items: Milch (2 l), Butter (500 gr), Zucker (2 kg), Mehl (5 kg), and Kekse (4 Pck). Each item has a 'Löschen' (Delete) button. The mobile version shows a simplified view with the same five items, each with a delete icon.

22.01.2024

19

## Wireframes:

- die grundlegende Struktur und Konzeption der Webseite oder Anwendung zu planen.
- Konzentration auf grundlegende Elemente und Funktionen
- Kommunikationsmittel zwischen den beiden Teammitgliedern → Gemeinsame Zielorientierung
- UX testen / Feedback sammeln

## Design Prototyp Mobil:

<https://www.figma.com/proto/MUbH2l6k6Payi9Em3XGpp0/snackTrack?page-id=88%3A48&type=design&node-id=88-53&viewport=-10%2C-45%2C1.49&t=ypoM6CpmxFAoCmdx-1&scaling=scale-down&starting-point-node-id=88%3A49&mode=design>

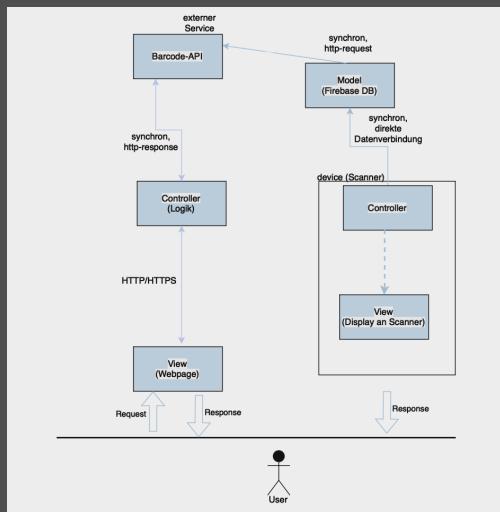
## Design Prototyp Desktop:

<https://www.figma.com/proto/MUbH2l6k6Payi9Em3XGpp0/snackTrack?page-id=88%3A48&type=design&node-id=88-53&viewport=-10%2C-45%2C1.49&t=ypoM6CpmxFAoCmdx-1&scaling=scale-down&starting-point-node-id=88%3A49&mode=design>

[id=0%3A1&type=design&node-id=3-538&viewport=-616%2C-115%2C0.51&t=qWLuklvVaLPSPSLs-1&scaling=scale-down&starting-point-node-id=3%3A538&mode=design](#)



## #Architekturdiagramm



22.01.2024

20

Der User bekommt am Scanner eine Antwort ob das Produkt, welches er eingescannt hat erfolgreich in die Datenbank aufgenommen wurde.

Gleichzeitig kann der User die Anfragen an die Webpage (neu laden/öffnen der Webpage) schicken und bekommt daraufhin die Seite mit den aktuellen Informationen der Datenbank gefüllt wiedergegeben.

In der Datenbank werden ausschließlich Barcodes und ihre Anzahl gespeichert (mehrmaliges Einstellen des gleichen Barcodes führt nicht dazu, dass mehrmals der gleiche barcode untereinander aufgeführt wird, sondern dass sich die Anzahl des einen Barcodes um 1 erhöht)

In der externen API werden die Barcodes um den Namen erweitert, sodass auf der Webpage die Namen und die Anzahl welche sich in der Datenbank befindet ausgegeben werden.

## #Datenbank



Leichte Integration



Realtime Database



Kostenloses Nutzungskontingent

22.01.2024

21

Firebase bietet eine Vielzahl von Funktionen, die es zu einem attraktiven Backend-as-a-Service (BaaS) und Mobile-Backend-as-a-Service (MBaaS) machen. Im folgenden wird die Entscheidung begründet:

### **Firebase**

Firebase bietet eine Vielzahl von Funktionen, die es zu einem attraktiven Backend-as-a-Service (BaaS) und Mobile-Backend-as-a-Service (MBaaS) machen. Hier sind einige Vorteile von Firebase im Vergleich zu anderen Anbietern:

**1. Einfache Integration:** Firebase bietet eine nahtlose Integration in verschiedene Plattformen, einschließlich iOS, Android, Web und Unity. Die SDKs sind gut dokumentiert und erleichtern die Implementierung von Funktionen in verschiedenen Anwendungen.

**2. Echtzeitdatenbank (Realtime Database):** Firebase bietet eine Echtzeitdatenbank, die in Echtzeit synchronisiert wird. Dadurch können Benutzer sofortige Updates erhalten, ohne die Seite aktualisieren zu müssen. Dies ist besonders nützlich für Anwendungen, die Echtzeitkollaboration oder Messaging benötigen.

**3. Authentifizierung:** Firebase bietet robuste Authentifizierungsdienste, die

verschiedene Authentifizierungsmethoden unterstützen, einschließlich E-Mail/Passwort, soziale Anmeldungen (Google, Facebook, Twitter), Telefonnummer und mehr. Dies macht die Benutzerverwaltung einfach und sicher.

**4. Cloud Firestore:** Firebase bietet neben der Echtzeitdatenbank auch Cloud Firestore, eine NoSQL-Datenbank, die skalierbar und leistungsstark ist. Firestore ermöglicht komplexe Abfragen und unterstützt Echtzeit-Updates.

**5. Cloud Functions:** Firebase ermöglicht das Schreiben von serverseitigem Code mithilfe von Cloud Functions. Dadurch können Sie benutzerdefinierte Logik implementieren, die in Reaktion auf bestimmte Ereignisse, wie das Hinzufügen eines neuen Datensatzes oder das Authentifizieren eines Benutzers, ausgelöst wird.

**6. Hosting und Cloud Storage:** Firebase bietet Hosting-Services für Webanwendungen und Cloud Storage für die Speicherung von Dateien. Die Integration ist einfach, und Sie können problemlos statische und dynamische Inhalte hosten.

**7. Analytics und Crash Reporting:** Firebase bietet integrierte Analysetools, mit denen Sie das Nutzerverhalten verstehen und die Leistung Ihrer Anwendung überwachen können. Darüber hinaus gibt es Funktionen für das Crash-Reporting, um Fehler schnell zu identifizieren und zu beheben.

**8. Push-Benachrichtigungen:** Firebase unterstützt das Senden von Push-Benachrichtigungen an iOS- und Android-Geräte, was für Benutzerbindung und Benachrichtigungen in Echtzeit wichtig ist.

**9. Automatisches Skalieren:** Firebase skaliert automatisch, um den Anforderungen Ihrer Anwendung gerecht zu werden. Dies ermöglicht eine einfache Skalierbarkeit, ohne dass Sie sich um die Infrastruktur kümmern müssen.

**10. Kostenloses Nutzungskontingent:** Firebase bietet ein großzügiges kostenloses Nutzungskontingent, was besonders für kleinere Projekte oder Entwickler mit begrenztem Budget attraktiv ist.

## **MongoDB**

**1. Dokumentenorientiertes Modell:** MongoDB verwendet ein dokumentenorientiertes Datenmodell, bei dem Daten in BSON (Binary JSON)-Dokumenten gespeichert werden. Dies erlaubt eine flexible und hierarchische Struktur, was besonders vorteilhaft ist, wenn sich die Anforderungen an das Datenmodell im Laufe der Zeit ändern.

**2. Skalierbarkeit:** MongoDB ermöglicht horizontale Skalierung, indem es Daten über mehrere Server verteilt. Dies verbessert die Leistung und ermöglicht eine einfache Anpassung an wachsende Datenmengen und Lasten.

**3. Leistungsstarke Abfragemöglichkeiten:** MongoDB unterstützt leistungsstarke Abfragen, Indexierung und Aggregation, was die Datenbank für komplexe Abfragen und Analysen geeignet macht. Es bietet auch Textsuche und geografische Indexierung für spezifische Anwendungsfälle.

**4. Eingebaute Replikation und Hochverfügbarkeit:** MongoDB bietet eingebaute

Replikation mit automatischem Failover. Durch die Konfiguration von Replikationssätzen kann MongoDB sicherstellen, dass Daten redundant gespeichert werden und der Dienst weiterhin verfügbar ist, auch wenn ein Server ausfällt.

**5. Flexible Schema-Designs:** Anders als bei relationalen Datenbanken erfordert MongoDB kein vordefiniertes Schema. Dies ermöglicht eine flexible Gestaltung der Datenmodelle, wodurch Anwendungen einfacher angepasst und erweitert werden können.

**6. Geo-Verteilung und Sharding:** MongoDB bietet Sharding-Unterstützung, wodurch Daten über mehrere Server oder Cluster verteilt werden können. Dies ermöglicht eine effiziente Verwaltung großer Datenmengen und eine verbesserte Skalierbarkeit.

**7. Community-Support und Dokumentation:** MongoDB verfügt über eine aktive Entwickler-Community und umfangreiche Dokumentation. Dies erleichtert die Integration, Fehlerbehebung und den Austausch von Wissen.

**8. Unterstützung für mehrere Plattformen:** MongoDB ist auf verschiedenen Plattformen verfügbar, einschließlich Windows, Linux und macOS. Es gibt auch Treiber und Bibliotheken für verschiedene Programmiersprachen, was die Integration in verschiedene Anwendungen erleichtert.

**9. Agile Entwicklung:** MongoDB passt gut zu agilen Entwicklungspraktiken, da es einfacher ist, Änderungen am Datenmodell vorzunehmen, ohne bestehende Datenstrukturen zu beeinträchtigen.

**10. Integrationsmöglichkeiten:** MongoDB lässt sich gut in moderne Anwendungsstapel und Frameworks integrieren. Es bietet offizielle Treiber für viele Programmiersprachen und Frameworks.

## **Vergleich der Datenbanken**

**1. Komplexität:** MongoDB ist eine eigenständige Datenbank, während Firebase eine vollständige Plattform für die Entwicklung von Web- und Mobilanwendungen bietet. Firebase umfasst nicht nur die Echtzeit-Datenbank, sondern auch Funktionen wie Authentifizierung, Cloud Functions, Hosting und mehr. Wenn Sie nur eine einfache Datenbank benötigen, kann Firebase einfacher zu verwenden sein, während MongoDB mehr Konfiguration und Wartung erfordern könnte.

**2. Skalierbarkeit:** Firebase bietet automatische Skalierung und Handhabung des Datenbankverkehrs auf der Grundlage der Anforderungen Ihrer Anwendung. MongoDB erfordert möglicherweise mehr manuelle Anpassungen und Skalierungsmanagement, insbesondere wenn Ihre Anwendung stark wächst.

**3. Flexibilität des Datenmodells:** MongoDB ist eine dokumentenorientierte Datenbank, was bedeutet, dass Sie flexiblere Datenmodelle haben können. Firebase Realtime Database ist jedoch eine NoSQL-Datenbank mit einem JSON-Datenmodell, das möglicherweise nicht so flexibel ist wie das BSON-Format von MongoDB.

**4. Abfragemöglichkeiten:** MongoDB bietet leistungsstarke Abfragefunktionen, die es Ihnen ermöglichen, komplexe Abfragen zu erstellen. Firebase Realtime Database hat

weniger leistungsstarke Abfragemöglichkeiten im Vergleich zu MongoDB. Firestore, das als Alternative zu Realtime Database angeboten wird, bietet jedoch verbesserte Abfragemöglichkeiten.

**5. Transaktionen:** Firebase Realtime Database hat in der Vergangenheit Einschränkungen bei Transaktionen gehabt. MongoDB hat zwar auch einige Einschränkungen bei komplexen Transaktionen, aber es bietet mehr Möglichkeiten für Transaktionen im Vergleich zu Firebase Realtime Database. Firestore, als Teil von Firebase, bietet verbesserte Unterstützung für Transaktionen im Vergleich zu Realtime Database.

**6. Kosten:** Die Kostenstruktur zwischen MongoDB und Firebase kann je nach Nutzungsszenario variieren. Firebase bietet einen nutzungsabhängigen Preismodell, während MongoDB typischerweise nach Kapazität (z. B. Speicher und CPU) abgerechnet wird. Je nach den Anforderungen Ihres Projekts kann die Kostenstruktur unterschiedlich ausfallen.

## **Entscheidung**

Die Wahl zwischen MongoDB und Firebase hängt stark von den spezifischen Anforderungen und Zielen eines Projekts ab. Wenn Einfachheit und eine integrierte Plattform für die Entwicklung von Web- und Mobilanwendungen im Vordergrund stehen, könnte Firebase die bevorzugte Wahl sein. Firebase bietet nicht nur eine Echtzeit-Datenbank, sondern auch eine Vielzahl von Funktionen wie Authentifizierung, Cloud Functions und Hosting, was die Entwicklung und Skalierung von Anwendungen erleichtert.

Die automatische Skalierung und das Management des Datenbankverkehrs machen Firebase besonders attraktiv für Anwendungen mit variabler Last. Die einfache Handhabung von Skalierungsvorgängen kann die Entwicklungszeit verkürzen und die Ressourcen effizienter nutzen.

In Bezug auf die Datenmodellflexibilität ist MongoDB mit seinem dokumentenorientierten Ansatz besser geeignet, wenn sich die Anforderungen an das Datenmodell im Laufe der Zeit ändern können. MongoDB erfordert jedoch möglicherweise mehr Konfiguration und Wartung.

Die leistungsstarken Abfragemöglichkeiten von MongoDB machen sie ideal für Anwendungen, die komplexe Abfragen und Analysen erfordern. Firebase Realtime Database mag in dieser Hinsicht etwas beschränkter sein, aber Firestore bietet verbesserte Abfragemöglichkeiten.

Transaktionen könnten ein entscheidender Faktor sein, wenn komplexe Transaktionen erforderlich sind. MongoDB bietet hier mehr Möglichkeiten im Vergleich zur Vergangenheit, während Firestore in Firebase verbesserte Unterstützung für Transaktionen bietet.

Die Kostenstruktur spielt ebenfalls eine Rolle. Firebase basiert auf einem nutzungsabhängigen Preismodell, während MongoDB nach Kapazität abgerechnet

wird. Die genaue Kostenstruktur hängt von den spezifischen Anforderungen des Projekts ab.

Letztendlich wurde sich für Firebase entschieden, was darauf hindeutet, dass die integrierte Plattform, die automatische Skalierung und andere Funktionen von Firebase besser zu den spezifischen Anforderungen des Projekts passen.

# #API



Umfassende Dokumentation



Keine Begrenzung



Kostenloses Nutzungskontingent

22.01.2024

22

Um eine geeignete API zu finden, wurden mehrere APIs begutachtet und bewertet. Die folgenden Pro- und Contra-Aspekte stellen dabei die jeweils wichtigsten Aspekte in Bezug auf das Projekt dar:

## 1. Spoonacular:

- a) + Große Auswahl an Parameter Übergabe
- b) + Umfassende Dokumentation
- c) + Student Access für 10\$/month
- d) + Umfassende Auswahl an Rezepten (günstig, vegan, etc.)
- e) - Nur 150 points/ day
- f) - Keine Barcode DB
- g) - Backlink required

## 2. Edamam:

1. + Kostenlos
2. - Kurze Dokumentation
3. - Keine Barcode DB
4. - /+ 10.000 Request pro Monat

## 3. Fatsecret:

1. + 5000 API calls/ day

2. + Premier Free für non-Profit und Startups
  3. + Food Tracking
  4. - Unausgearbeitete/ Unübersichtliche Dokumentation
  5. - Sprache nur Englisch
  6. - UPC Barcode nur mit uneingeschränktem Premier Zugang
1. OpenFoodFacts:
    1. + Keine Call Einschränkung
    2. + 100% kostenlos
    3. + Datenbank wird stetig erweitert
    4. + Barcodes zu jedem eingespeicherten Produkt vorhanden
    5. + Umfassende Dokumentation

Die Open Food Facts API wurde für das Projekt ausgewählt, da diese alle Ansprüche an das Projekt erfüllen. Zusätzlich ist das kostenlose Nutzerkontingent nicht an eine Einschränkung von Daten oder Aufrufen gebunden.

Die Open Food Facts ist eine kostenlose Online-Datenbank mit Lebensmitteln aus der ganzen Welt, die unter der Open Database License (ODBL) lizenziert ist, während die von den Mitwirkenden hochgeladenen Grafiken unter einer Creative Commons Attribution-Share Alike-Lizenz veröffentlicht werden.



# Proof of Concept

Überarbeitung aus Audit 2

22.01.2024

23

	<b>Korrekte Platzierung von Scanner und Barcode</b> Der Barcode wird <b>in einem 90°-Winkel (gerade)</b> vor den Scanner gehalten		<b>Aktivierung eines Scanners</b> Der ESP32 wird an eine Stromquelle angeschlossen
Beschreibung	Der Produktcode wird korrekt erfasst	Exit Kriterien	Der Scanner zeigt dem Benutzer durch Aufleuchten des weißen Lichtes, dass er startbereit ist
Fail Kriterien	Der Produktcode kann nicht erfasst werden und der Barcode muss neu vor den Scanner ausgerichtet werden	Fail Kriterien	Der Scanner leuchtet nicht auf und der Benutzer bekommt kein Feedback
Fallbacks	Andere Platzierung des Scanners oder Barcode, sodass letzterer sich im Scan Feld (weißes Licht) befindet.	Fallbacks	Die Verbindung von ESP und Scanner muss überprüft und gegebenenfalls korrigiert werden. Stromversorgung überprüfen.
Durchführung	<p>1) Der Barcode wurde in 90°-Winkel vor den Scanner gehalten. Danach wurde der Barcode in mehreren Abständen von je 1cm vor den Scanner wegbewegt (Beginnend bei 0cm). Der Scanner war fähig den Barcode in einem Abstandsbereich von 5-20cm zu lesen. Der Produktcode wurde erfasst und der Nutzer erhält eine Output Indication zur Rückmeldung.</p> <p>2) Der Barcode wurde in verschiedenen Winkeln (&lt;90°, &gt;90°) vor den Scanner gehalten. Die Barcodes konnten nicht eingelesen werden.</p> <p>3) Der Barcode wurde in einem 90°-Winkel, mit einem Abstand von 10cm vor den Scanner gehalten. Zwischen Scanner und Barcode befand sich ein eingefärbtes Glas. Der Barcode konnte nicht gelesen werden.</p> <p>4) Der Barcode wurde in einem Abstand von 0cm vor den Scanner gehalten. Der Barcode konnte nicht gelesen werden. Das gleiche Ergebnis wurde in den Versuchen 1-4cm und &gt;20cm erzielt.</p>	Durchführung	<p>1) Der ESP32 wurde an eine passende Stromquelle angeschlossen. Der Scanner leuchtet rot und weiß. Der Scanner war einsatzbereit.</p> <p>2) Der ESP32 wurde an eine passende Stromquelle angeschlossen. Es erleuchtete kein Licht. Anschließend wurden die Stromversorgung und die Verbindung zwischen ESP und Scanner überprüft. Das „GROUND“-Kabel zwischen ESP und Scanner hatte sich gelöst und dies wurde korrigiert. Nach eines neuen Versuchs wurde das Problem gelöst und der Scanner leuchtete rot und weiß auf.</p>
	<b>Aufleuchten einer LED</b> Bei <b>erfolgreichem</b> Abscannen des Barcodes erhält der Benutzer <b>immer</b> eine visuelle Rückmeldung in Form einer aufleuchtenden farbigen LED ( <b>grün oder rot</b> )		<b>Rückmeldung des Scanners an den Benutzer</b> Der bereits im Scanner verbaute Buzzer gibt dem Benutzer bei <b>erfolgreichem Scan</b> eine Rückmeldung
Beschreibung	Der Barcode wird erfasst und eine LED beginnt zu leuchten	Exit Kriterien	Der Barcode wird erfasst und der Buzzer gibt eine Rückmeldung
Fail Kriterien	Der Barcode wird erfasst und keine LED beginnt zu leuchten	Fail Kriterien	Der Barcode wird erfasst, der Benutzer bekommt jedoch keine Rückmeldung des Buzzers
Fallbacks	Überprüfung der Komponenten LEDs, wenn nötig sind diese auszutauschen	Fallbacks	Scannen des Barcodes für OUPUT INDICATION (Buzzer), um die gewünschte Rückmeldung zu erhalten
Durchführung	<p>1) Der Barcode wurde (siehe Korrekte Platzierung von Scanner und Barcode) korrekt vor den Scanner gehalten. Eine der LEDs leuchtet auf.</p> <p>2) Der Barcode wurde erfolgreich abgescannt. Es leuchtet keine LED auf. Nach Überprüfung der Verbindung und erneutem Anschluss der LEDs zum ESP32 wurde der Vorgang wiederholt. Der Barcode wurde erfolgreich abgescannt. Eine der LEDs leuchtet auf.</p>	Durchführung	<p>1) Der Barcode wurde in korrektem Abstand und Winkel vor dem Scanner platziert. Der Scanner gab ein haptisches Feedback (Buzzer) zurück.</p> <p>2) Der Barcode wurde in abweichendem Abstand und Winkel vor dem Scanner platziert. Der Scanner gab kein haptisches Feedback (Buzzer) zurück</p>

22.01.2024

24

(POCs siehe Folie)

	<b>Benutzer verändert die Anzahl eines Barcodes</b>		<b>Benutzer speichert Daten in Realtime Database (RD)</b>
Beschreibung	Der Switch befindet sich auf Position „0“. Der Benutzer kann durch Einscannen eines Barcodes die Anzahl eines bereits in der Datenbank vorhandenen Barcodes um 1 dekrementieren. Die geringste Anzahl eines Barcodes in der Datenbank beträgt 0		Der Switch befindet sich auf Position „1“. Der Benutzer kann verschiedene Barcodes speichern. Die grüne LED leuchtet auf, wenn der Barcode erfolgreich in der RD gespeichert wurde. Es wird ein neuer Barcode mit der Anzahl 1 gespeichert. Falls dieser Barcode bereits in der Database vorhanden ist, wird die Anzahl um 1 inkrementiert.
Exit Kriterien	Die Anzahl eines Barcodes wird um 1 dekrementiert		Der eingelesene Barcode wird in der Datenbank gespeichert
Fail Kriterien	Die RD kann nicht erreicht werden, die rote LED leuchtet auf und die Anzahl bleibt unverändert		Die RD kann nicht erreicht werden, die rote LED leuchtet auf
Fallbacks	Überprüfung der Internetverbindungen des ESPs. Überprüfen der Inhalte der Firebase		Überprüfung der Internetverbindung des ESPs, ggf. muss der Standort des ESP32 so verändert werden, dass eine stabile Internetverbindung gewährleistet ist.
Durchführung	<ul style="list-style-type: none"> <li>1) Ein Barcode wurde eingescannt und die grüne LED leuchtete auf. Nach Überprüfen der Firebase Konsole wurde bestätigt, dass die Anzahl des Barcode erfolgreich um 1 dekrementiert wurde.</li> <li>2) Ein Barcode wurde ohne aktive Datenverbindung abgescannt. Die rote LED leuchtet auf. Nach Überprüfung der Firebase Realtime Database wurde festgestellt, dass die Anzahl des Barcodes nicht dekrementiert wurde.</li> </ul>		<ul style="list-style-type: none"> <li>1) Ein Barcode wurde eingescannt und die grüne LED leuchtete auf. Nach Überprüfen der Firebase Konsole wurde bestätigt, dass der Barcode erfolgreich abgespeichert wurde.</li> <li>2) Ein Barcode wurde ohne aktive Datenverbindung abgescannt. Die rote LED leuchtet auf. Nach Überprüfung der Firebase Realtime Database wurde festgestellt, dass der Barcode nicht abgespeichert wurde</li> </ul>
	<b>Einbindung der API</b>		<b>Darstellung der Datenbank auf einer Webseite</b>
Beschreibung	Ein request wird an die OpenFoodFacts-API geschickt		Die von der API umgewandelten Daten werden dem Benutzer auf einer eingebundenen Webseite zur Verfügung gestellt
Exit Kriterien	Die Datenbank gibt den Statuscode 200 zurück		Die korrekt ausgelesenen API-Datensätze zu den passenden Barcodes werden auf der Webseite angezeigt. (Statuscode 200)
Fail Kriterien	Der API-Anbindung ist nicht korrekt, es gibt einen Statuscode 4XX zurück		Die Webseite gibt einen Fehler beim Lesen der HTML-File zurück (interner Serverfehler, Statuscode 500)
Fallbacks	Die URL der OpenFoodFacts muss überprüft und ggf. korrigiert werden		Korrigieren des HTML-File Pfads
Durchführung	<ul style="list-style-type: none"> <li>1) Der request war erfolgreich und erhält eine response (statuscode 200)</li> <li>2) Der request war nicht erfolgreich (statuscode 4xx)</li> </ul>		<ul style="list-style-type: none"> <li>1) Die Website wird mit korrektem Inhalt angezeigt</li> <li>2) Die Website wird nicht mit korrektem Inhalt angezeigt, die Konsole erzeugt einen Error (Fehler beim Lesen der HTML-File, die Webseite zeigt den Schriftzug „interner Serverfehler“)</li> </ul>

22.01.2024

25

(POCs siehe Folie)

# #Projektrisiken



22.01.2024

26

## 1. Technologische Risiken:

1. **Hardwarefehler:** Probleme mit der Scanner-Hardware könnten auftreten, was zu Ausfällen oder ungenauen Scans führt.
2. **Kompatibilitätsprobleme:** Schwierigkeiten bei der Integration des Scanners mit anderen Systemen oder Geräten in der Umgebung.

## 2. Datenschutz und Sicherheit:

1. **Datenlecks:** Die gesammelten Informationen könnten durch unsachgemäße Sicherheitsmaßnahmen gefährdet werden.
2. **Unbefugter Zugriff:** Risiken in Bezug auf den Zugriff auf die Datenbank und die Sicherheit der gespeicherten Informationen.

## 3. Regulatorische Risiken:

1. **Compliance-Anforderungen:** Das Projekt muss möglicherweise verschiedenen gesetzlichen Vorschriften und branchenspezifischen Standards entsprechen, insbesondere im Bereich Lebensmittelsicherheit und Datenschutz.

## 4. Benutzerakzeptanz:

1. **Schulungsbedarf:** Benutzer müssen möglicherweise geschult werden, um den Scanner effektiv zu nutzen.
2. **Benutzeroberfläche:** Wenn die Benutzeroberfläche nicht intuitiv ist,

könnte dies zu Akzeptanzproblemen führen.

#### **5. Projektmanagement:**

1. **Verzögerungen:** Unvorhergesehene Probleme könnten zu Verzögerungen bei der Projektumsetzung führen.
2. **Budgetüberschreitungen:** Die Kosten könnten höher sein als ursprünglich geplant.

#### **6. Lieferanten- und Beschaffungsrisiken:**

1. **Lieferantenprobleme:** Verzögerungen oder Qualitätsprobleme bei den Hardwarekomponenten könnten auftreten.
2. **Preisschwankungen:** Kosten für Hardwarekomponenten könnten steigen, was das Budget beeinträchtigen könnte.

#### **7. Änderungsmanagement:**

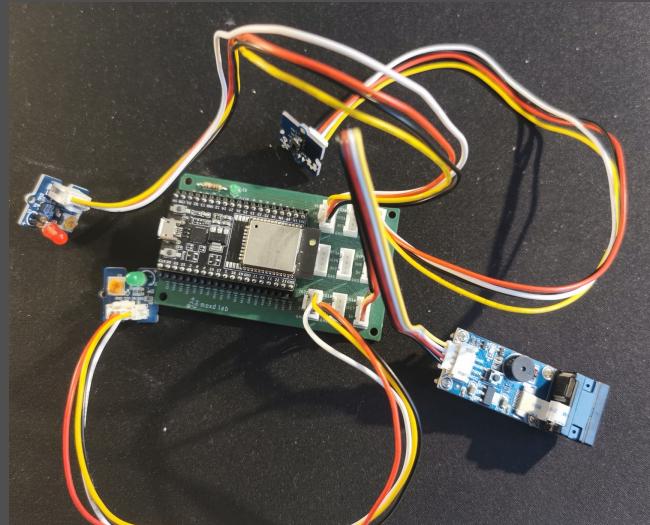
1. **Anforderungsänderungen:** Unvorhergesehene Änderungen in den Anforderungen könnten den Entwicklungsprozess beeinflussen.
2. **Ungenauie Spezifikationen:** Wenn die Anforderungen nicht klar definiert sind, könnte dies zu Missverständnissen und Fehlern führen.

#### **8. Umgebungsbedingte Risiken:**

1. **Umwelteinflüsse:** Der Scanner könnte durch Umweltbedingungen wie Temperaturschwankungen oder Feuchtigkeit beeinträchtigt werden



## #Rapid Prototype



22.01.2024

27

Für den Rapid Prototype wurden die folgenden Bauteile verwendet:

- ESP Espressif 32
- Rote LED
- Grüne LED
- Barcode Scanner Module, 1D/2D Codes Reader
- Switch (P)

Diese Bauteile wurden aus dem MoxdLab der Technischen Hochschule Köln ausgeliehen.

# #Projektaussicht

- Erstellen des funktionalen Prototyps
- Poster erstellen
- Kritisches Reflektieren des Prozessassessments

22.01.2024

2024		
Januar	Februar	März
1 Mo Neujahr	1 Do funktionaler Prototype	1 Fr
2 Di	2 Fr +	2 Sa
3 Mi Modellierung der Anwendungslogik	3 Sa Poster erstellen	3 So
4 Do	4 So	4 Mo 10
5 Fr	5 Mo Open Space	5 Di
6 Sa Heilige Drei Könige	6 Di Poster bearbeiten	6 Mi
7 So	7 Mi	7 Do
8 Mo Open Space	8 Do	8 Fr
9 Di	9 Fr	9 Sa
10 Mi Durchgeführte PoCs	10 Sa	10 So
11 Do	11 So	11 Mo 11
12 Fr	12 Mo	12 Di
13 Sa	13 Di	13 Mi
14 So	14 Mi Fazit und kritisch reflektiertes Prozessassessmentsmen	14 Do
15 Mo Open Space	15 Do	15 Fr
16 Di	16 Fr	16 Sa
17 Mi Feedback einarbeiten	17 Sa	17 So
18 Do Audit vorbereiten	18 So	18 Mo 12
19 Fr	19 Mo Open Space	19 Di
20 Sa	20 Di Feedback einarbeiten	20 Mi
21 So	21 Mi	21 Do
22 Mo 3. Audit	22 Do Audit vorbereiten	22 Fr
23 Di	23 Fr + Poster	23 Sa
24 Mi funktionaler Prototype	24 Sa	24 So
25 Do	25 So	25 Mo 13
26 Fr	26 Mo 4. Audit + Postersammlung	26 Di
27 Sa	27 Mi	27 Do
28 So	28 Mi	28 Do
29 Mo Open Space	29 Do	29 Fr Karneval
30 Di funktionaler Prototype		30 Sa
31 Mi		31 So Beginn der Sommerferien



28



## #Abbildungsverzeichnis

- <https://firebase.google.com/?authuser=0>
- [https://www.waveshare.com/wiki/Barcode\\_Scanner\\_Module](https://www.waveshare.com/wiki/Barcode_Scanner_Module)
- <https://world.openfoodfacts.org/>