

Technology Arts Sciences TH Köln

Entwicklungsprojekt interaktive Systeme
Wintersemester 2018/19

Dozenten:

Prof. Dr. Gerhard Hartmann

Prof. Dr. Kristian Fischer

Mentoren:

Ngoc-Anh Gabriel

Lena Wirtz

von

Kristian Czepluch – (11112444)

Denise Kübler – (11119149)

Inhalt

Mensch-Computer-Interaktion Abschnitt:	5
User Profiles	5
Contextual Task Analysis	6
Work Environment Analysis:	7
Privatperson	7
Anbieter	8
Abholer.....	8
Task Scenarios	9
List of User Tasks	10
Tasks - Allgemeine Privatperson.....	10
Tasks - Anbietender.....	11
Tasks - Abholender	11
Task Organization Model	11
Plattform Capabilities and Constraints.....	12
General Design Principles:.....	12
Usability Goals.....	13
Qualitative Goals	13
Quantitative Goals.....	14
.....	17
.....	17
Work Re-engineering	17
Task Organization Model Re-engineering.....	18
Task Scenario.....	19
Use Cases	19
Conceptual Model Design	20
Orientierung des Systems:	20
Identifikation der Produkte:	21
Navigation	23
Screen Design Standard.....	25
Detailed User Interface Design.....	26
Evaluation	31
Benutzer	31
Aufgaben	31
Durchführung.....	31
Ergebnis der Evaluation.....	31

Abschnitt Webbasierte Anwendungen:.....	32
Begründung der gewählten Architektur	32
Entwicklung mit Client-Server Architektur:	32
Entwicklung mit Peer 2 Peer Architektur:	32
Begründung der gewählten Programmiersprachen	33
Clientseite	33
Serverseite.....	33
Entwicklung mit Java	33
Entwicklung mit Javascript	34
System Schnittstelle	34
Entwicklung mit REST	34
Entwicklung mit SOAP	35
Abwägung verschiedener Librarys.....	35
Iteration der Architektur	36
Datenschutz und Sicherheit der Dienste	36
Firebase Datenschutz und Sicherheit	36
Google Maps Datenschutz und Sicherheit.....	37
Edamam Datenschutz und Sicherheit.....	37
OpenWeatherMap Datenschutz und Sicherheit.....	37
Fazit.....	37
Domänenmodelle und Iterationen.....	37
Iteration Kommunikationsmodelle.....	39
Iteration Deskriptives Kommunikationsmodell	39
Iteration Präskriptives Kommunikationsmodell	41
Begründung der gewählten Verteilung der Anwendungslogik	42
Beschreibung der Datenstrukturen	43
Hypermedia-Konzept.....	46
Datenstruktur Client.....	47
Iteration Beschreibung der Anwendungslogik Client:	48
Iteration Beschreibung der Anwendungslogik des Servers:	51
Asynchrone Kommunikation:	52
Quellenverzeichnis:	54
Anhang	56
User Profiles	56
Contextual Task Analysis	66
Task Scenarios	68

Task Scenario Re-engineering	71
Qualitative Ziele	75
Concrete Use Cases	77
REST Tabelle	82
Iteration Stakeholderanalyse	88
Iteration Proof of Concept.....	91
Iteration Datenformate	94
Iteration Anforderungen:	95

Mensch-Computer-Interaktion Abschnitt

Wie bereits im Konzept beschrieben, wird die MCI Modellierung des Projektes anhand des Usability Engineering Lifecycle nach Deborah Mayhew durchgeführt.

User Profiles

Zu Beginn des Projektes ist es besonders wichtig, die späteren Benutzer des Systems genauer zu verstehen. Hierzu werden im *Usability Engineering Lifecycle* User Profiles angelegt. Diese tragen dazu bei, die bestimmten Merkmale sowie deren Ausprägungen der Benutzer zu identifizieren. Dies ist besonders wichtig um die Charakteristika sowie die allgemeinen Kenntnisse oder Fähigkeiten der Benutzer zu verstehen und in der Modellierung des Systems zu berücksichtigen.

Im Folgenden werden die für das Projekt relevanten Merkmale sowie deren mögliche Ausprägungen aufgelistet und erläutert.

- **Demografie:** Geschlecht, Alter, Wohnsitz und Wohnverhältnisse oder auch Sprachkenntnisse sind für die allgemeine Strukturierung und Planung des Systems von Bedeutung.
- **Sozial-ökonomischer Status:** Beruf, Einkommen, Qualifikationen und der allgemeine Bildungsstand sind wichtig, um die verschiedenen gesellschaftlichen Hintergründe der Benutzer berücksichtigen zu können. Innerhalb dieser Kategorie ist außerdem die Mobilität des Nutzers von Bedeutung, da diese die mögliche Distanz von Angeboten beeinflusst.
- **Fachkenntnisse:** Wieviel Wissen im Kontext der Domäne der Lebensmittelrettung oder Lebensmittelhaltbarkeit der Benutzer hat, kann für die Modellierung des Systems relevant sein, da in gewissen Fällen auf die verschiedenen Wissensstände eingegangen werden sollte.
- **Technische Erfahrungen und Kenntnisse:** Die Erfahrung der Benutzer mit technischen Systemen auf denen die Applikation zur Verfügung gestellt wird - in diesem Fall ein mobiles Endgerät mit Android Betriebssystem - ist relevant, da die Erfahrung mit den Systemen beispielsweise das Verstehen von gängigen Steuerungsmöglichkeiten mit sich bringt.
- **Verfügbare Technologien:** Die Hard- und Software die dem Benutzer zur Verfügung steht sollte identifiziert werden
- **Kultureller Hintergrund:** Der kulturelle Hintergrund kann von Bedeutung sein, da in verschiedenen Kulturen verschiedene Informationen oder Stimuli unterschiedlich aufgefasst werden können. Auf diese Auffassungen sollte dann bei dem Design geachtet werden.
- **Spezielle Produkterfahrung:** Unter Umständen kann es hilfreich sein die Erfahrung der Benutzer mit ähnlichen Produkten innerhalb der Domäne oder Produkten mit ähnlichen Funktionen zu identifizieren. Haben die Benutzer beispielsweise schon Erfahrungen mit Produkten gemacht, die eine Angebotserstellung anbieten, so kann es sein, dass die Angebotserstellung den Benutzern in diesem Produkt leichter fällt.

- **Motivation und Einstellung:** Hierbei ist es von Bedeutung zu identifizieren warum der Benutzer das System nutzen möchte und auch welche Einstellung er gegenüber der Domäne hat.
- **Aufgaben:** Hierzu gehören die Aufgaben die die Benutzer innerhalb des Nutzungskontextes im System erfüllen.
- **Lernfähigkeit:** Hierbei handelt es sich um die Fähigkeit des Benutzers die Benutzung des Systems zu erlernen sowie den Aufbau, die Navigation und die Funktionen innerhalb des Systems zu verstehen. Dies ist besonders relevant um die Komplexität des Systems planen zu können. Außerdem kann dies die späteren Änderungen innerhalb des Systems beeinflussen. Fällt es den Benutzern schwer Neues zu erlernen, so ist es nicht ratsam regelmäßig neue Funktionen im System zu ergänzen oder das Design zu verändern.

Die Folgenden User Profiles stellen nur eine Teilmenge der erwarteten tatsächlichen User dar. Zu den tatsächlichen Usern könnten unter anderem auch ältere Zielgruppen gehören. Diese haben in der Regel noch ganz andere Merkmale und Erfahrungsgrade, welche den Gesamtumfang des Projektes jedoch überschreiten würden. Die dargestellte Teilmenge beinhaltet die intendierte Zielgruppe des Systems, auf die sich im Verlauf des Projektes konzentriert wird und somit weiterhin verwendet wird. Die erstellten User Profiles sind im Anhang zu finden. Weiterhin wurden die Stakeholder überarbeitet und sind ebenfalls im Anhang zu finden.

Contextual Task Analysis

Bei der Task Analysis nach Mayhew geht es primär darum, den Benutzer und seine Aufgaben bzw. Arbeitsschritte zu identifizieren. Erst dann ist es möglich ein System zu schaffen, welches den Benutzer optimal bei der Verwendung einzelner Funktionen unterstützen kann. An dieser Stelle konzentrieren wir uns primär auf die Privatpersonen, da diese nach den Ergebnissen den Domänenrecherche im Zentrum stehen. Es werden in dieser Analyse die Benutzer verwendet, welche zuvor in den User Profiles ermittelt wurden. Es sollten nach Mayhew eine Vielzahl an Interviews geführt werden, jedoch kann dies auch durch einen Shortcut auf eine kleinere Zahl reduziert werden. Außerdem wurden eigene Erfahrungen mit eingebracht. Für die Beobachtung wurde die Vorlage: Contextual Interview Data-Collection Template verwendet.

Tabelle 10: Contextual Observation/Interview 1. Privatperson

Actor	Trigger	Use Case	Task Scenario Sequence	Problems
Privatperson	Bemerkt, dass Lebensmittel übriggeblieben sind.	Nach geeigneter Verwendungsmöglichkeit für die Lebensmittel suchen.	1. Privatperson realisiert, dass Lebensmittel übrig sind. 2. Privatperson merkt sich die Lebensmittel die er Verbrauchen möchte. 3. Privatperson wählt eine Recherchequelle (Internet, Kochbuch).	1. Es ist ein großer Aufwand eine geeignete Verwendungsmöglichkeit zu finden. 2. Gefundene Rezepte können aus verschiedenen Gründen nicht geeignet sein.

			Benutzer wählt Verwendungsmöglichkeit aus. 4. Privatperson vergleicht gefundene Verwendungsmöglichkeiten nach Aufwand, Einschränkungen, Vorlieben, Zubereitungszeiten und vorhandenen Zutaten.	3. Andere Zutaten, welche vorhanden sind werden nicht berücksichtigt. Somit wird neues gekauft wobei andere Zutaten liegen bleiben.
	Haltbarkeit eines Lebensmittels ist unklar.	Nach einem Haltbarkeitsdatum für das Lebensmittel suchen.	1. Privatperson entdeckt ein Lebensmittel in einem unklarem Haltbarkeitszustand. 2. Privatperson merkt sich das Lebensmittel. 3. Privatperson sucht sich ein geeignetes Recherchemedium aus. (Internet) 4. Privatperson sucht nach Erfahrungswerten oder ähnlichem. 5. Privatperson entscheidet ob Lebensmittel noch verzehrbar ist oder entsorgt werden muss.	1. Es ist viel Aufwand eine Recherche zu der Haltbarkeit zu betreiben. 2. Falls das Haltbarkeit unklar ist, machen sich viele nicht die Mühe nach Informationen zu suchen. 3. Umweltfaktoren werden bei der online Recherche nicht beachtet. (Lagerungsfaktoren) 4. Kaufdatum wurde sich nicht gemerkt.

An dieser Stelle wurde nur eine Contextual Task Analysis exemplarisch dargestellt, jedoch sind weitere im Anhang zu finden.

Work Environment Analysis

Im Folgenden wird das Arbeitsumfeld der Benutzer analysiert, unter Berücksichtigung der aus den Contextual Interviews gewonnenen Informationen. Es wurde bereits im Konzept der Nutzungskontext beschrieben, jedoch hat sich dieser an dem Präskriptiven Kommunikationsmodell gerichtet, statt den Ist-Zustand zu beschreiben. Statt einer Iteration des Nutzungskontextes wird die Work Environment Analysis durchgeführt, da diese die gleichen Ergebnisse, wie die Iteration des Nutzungskontextes, bringt.

Privatperson

Physical Work Environment

Das Arbeitsumfeld der Privatperson (als Normaler Benutzer) ist überwiegend Zuhause, wobei dieser bei der Erledigung der Aufgaben sich größtenteils in der Küche und im Wohnzimmer befindet. Dies liegt daran, dass die Lebensmittel sich i.d.R in der Küche, wobei diese entweder im Kühlschrank oder in oder auf Schränken gelagert werden. Der Punkt der Realisierung, dass Lebensmittel übriggeblieben sind, findet mit hoher Wahrscheinlichkeit in der Küche statt, da an dieser Stelle der Benutzer seine Lebensmittel vor Augen hat. Es ist jedoch nicht auszuschließen,

dass diese Realisierung auch im Wohnzimmer oder in anderen Räumen stattfinden kann. Haushalte sind in der Regel gut ausgeleuchtet und bieten eine ausreichend große Umgebung, in der sich der Benutzer frei bewegen kann. Die Umgebungsgeräusche sind gering einzustufen, können sich jedoch Situationsabhängig verändern.

Sociocultural Work Environment

Das Arbeitsumfeld besteht entweder aus Familienmitgliedern, Freunden, Bekannten oder Mitbewohnern. Diese können zu Unterbrechungen oder Ablenkungen der Erledigung der Aufgaben führen. Grundsätzlich befindet sich der Benutzer in einer ihm vertrauten Umgebung, welche in ihm Sicherheit und Vertrauen erweckt.

Anbieter

Physical Work Environment

Das Arbeitsumfeld der Privatperson (als Anbieter) liegt ebenfalls, wie beim Normalen Benutzer überwiegend zuhause, da eine Internetverbindung oder eine andere Kommunikationsmöglichkeit wie Telefon oder Ähnliches vorhanden sein muss. Ebenfalls liegt der konkrete Aufenthaltsort in der Küche, da sich dort die Lebensmittel befinden, die er anbieten möchte und im Wohnzimmer, da dort oftmals ein Computer mit Internetanschluss oder das Telefon vorhanden ist. Die Ausleuchtung, Lärmpegel und die Größe der Bewegungsumgebung ist an dieser Stelle analog zu der des normalen Benutzers. Ein weiteres Arbeitsumfeld kann eine Lokation sein, welche sich außerhalb der Wohnung befindet, an dem die Weitergabe der Lebensmittel stattfindet. An dieser Stelle ist mit einem erhöhten Geräuschpegel, von der Tageszeit und wetterabhängigen Lichtverhältnissen und generell unterschiedlichen Wetterfaktoren zu rechnen. Diese können den Benutzer bei der Erledigung der Aufgaben behindert bzw. beeinflussen und sollten nicht vernachlässigt werden.

Sociocultural Work Environment

Das Arbeitsumfeld ist im Umfeld des Haushaltes analog zu denen des Normalen Benutzers. Um eine Redundanz zu vermeiden wird lediglich auf dessen Soziokulturelle Umgebung in Bezug auf den Haushalt verwiesen. Ein Unterschied besteht jedoch bei der Umgebung außerhalb des Haushaltes. Der Benutzer nimmt Kontakt mit einer ihm Bekannten oder Fremden Person an. Der Anbieter befindet sich psychologisch gesehen in der dominanteren Position, da der Abholer mehr von diesem Abhängig ist als umgekehrt.

Abholer

Physical Work Environment

Das Arbeitsumfeld der Privatperson (als Abholer) unterscheidet sich zu den vorangegangenen Arbeitsumgebungen in der Gewichtung der einzelnen Arbeitsorte. Grundsätzlich ist auch hier wieder der eigene Haushalt der Hauptausgangspunkt, jedoch ist die Recherche eine Hauptaufgabe, welche mit hoher Wahrscheinlichkeit im Wohnzimmer oder ähnlichen Räumlichkeiten stattfindet, wo ein Kommunikationsgerät (Handy, Computer) zur Verfügung steht. Die Ausleuchtung, Lärmpegel und die Größe der Bewegungsumgebung ist an dieser Stelle analog zu der des normalen Benutzers oder des Anbieters. An dieser Stelle ist jedoch wichtig, dass ein Großteil des Arbeitsumfelds unterschiedliche Lokationen, welche außerhalb des Haushaltes liegen, da der Abholer i.d.R der ist, welcher die Lebensmittel auch abholen muss. Somit ist auch der Weg,

welcher zum verabredeten Abholort zurückgelegt werden muss und der Abholort Teil seines Arbeitsumfelds. Diese physikalischen Eigenschaften wurden bereits in der Analysis des Anbieters beschrieben und sind analog. An dieser Stelle können viele schwer kalkulierbare Ereignisse, wie z. B. Unterbrechungen von anderen Personen, Navigationsprobleme, externe Faktoren wie das Wetter den Abholer beeinflussen.

Sociocultural Work Environment

Das Arbeitsumfeld ist im Umfeld des Haushaltes analog zu denen des Normalen Benutzers. Um eine Redundanz zu vermeiden wird lediglich auf dessen Soziokulturelle Umgebung in Bezug auf den Haushalt verwiesen. Aus soziokultureller Sicht ist der Abholer im Abholprozess in einer niedrigeren Position in der Dominanzhierarchie, da er etwas von einer anderen Person anfordert. Dies kann zu leichter Anspannung während des Zusammentreffens führen, falls es sich um zuvor unbekannte Personen handelt.

Fazit

Das Arbeitsumfeld in Haushalten ist im Allgemeinen nicht all zu stressig und die Privatperson steht nicht Zwangsweise unter Druck. Der Haushalt vermittelt zudem ein Gefühl der Sicherheit und von Vertrauen. Aus diesem Grund können Designentscheidungen für das Interface an dieser Stelle umfangreicher und komplexer sein, sollten aber dennoch den Benutzer nicht überfordern. Das Gestaltung sollte Mögliche Unterbrechungen berücksichtigen. Im Gegensatz hierzu steht das Arbeitsumfeld, welches sich außerhalb des Haushaltes befindet. An dieser Stelle gibt es mehrere Ablenkungen, mehrere potentielle Stressquellen und ggf. zeitlicher Druck. Das Interface sollte an diesen Stellen für den Benutzer übersichtlich, leicht Erlernbar, leicht Verwendbar und simpel gestaltet.

Task Scenarios

Anhand der Task Scenarios wird noch einmal im Detail auf die Aufgaben der Benutzer eingegangen. Hierbei werden die Aufgaben, welche innerhalb der Contextual Observations/Interviews ermittelt wurden, noch einmal im Detail mit ihren jeweiligen Haupt- und Unteraufgaben beschrieben um den genauen Ablauf dieser zu veranschaulichen. Dies dient dazu mögliche Probleme zu identifizieren und anhand dessen Möglichkeiten zur Verbesserung herauszufiltern welche später von dem User Interface umgesetzt werden können.

Die in den Contextual Observations/Interviews identifizierten Use Cases „Lebensmittel an eine Organisation spenden“ und „Lebensmittel bei einer Organisation abholen“ werden vorübergehend nicht modelliert, da Organisationen im System vorerst nicht weiter berücksichtigt werden, und daher keine Anpassung an das Interface in Bezug auf das Finden von Organisationen vorgenommen werden muss.

Task: Über die Haltbarkeit eines Lebensmittels erkundigen

User: Privatperson (als Normal Benutzer)

Description: Jens würde zum Kochen gerne die letzte Zwiebel in seinem Kühlschrank verwenden. Es ist schon eine Weile her, dass er das Netz mit Zwiebeln gekauft hat und ist sich daher nicht ganz sicher ob die letzte Zwiebel noch haltbar ist. Dies würde er nun gerne herausfinden

Task Flow:

1. Jens holt die Zwiebel aus dem Kühlschrank und betrachtet sie genau um mögliche Spuren von Schimmel oder Fäule zu entdecken. Die Zwiebel sieht optisch jedoch noch gut aus.
2. Jens setzt sich an seinen Computer und gibt in einer Suchmaschine den Suchbegriff „Haltbarkeit einer Zwiebel“ ein
3. Jens findet auf einer Internetseite die Aussage, dass Zwiebeln circa 6 Monate haltbar sind, wenn sie trocken und dunkel gelagert sind.
4. Jens sucht zur Sicherheit nach einer weiteren Quelle.
5. Auf einer anderen Internetseite findet er die Aussage, dass Zwiebeln ein Jahr lang haltbar sind, wenn sie richtig gelagert werden.
6. Jens beschließt nun zu recherchieren, wie Zwiebeln richtig gelagert werden.
7. Er findet heraus wie Zwiebeln ideal gelagert werden und stellt fest, dass er seine Zwiebeln nicht ideal gelagert hat.
8. Jens sucht weiter nach Haltbarkeitsdaten für „nicht ideal gelagerte“ Zwiebeln aber findet keine entsprechenden Ergebnisse.
9. Jens ist sich nun immer noch nicht sicher ob die Zwiebel noch haltbar ist. Um sicher zu gehen entschließt er sich dazu die Zwiebel lieber nicht zu verwenden und entsorgt sie stattdessen

Task Closure: Dieses Szenario kann 5-30 Minuten dauern, je nachdem wie sorgfältig recherchiert wurde. Das Ergebnis dieser Recherche ist in vielen Fällen nicht zufrieden stellend, da viele verschiedene Faktoren Einfluss auf die Haltbarkeit der Lebensmittel haben. Diese Faktoren werden innerhalb der meisten Quellen jedoch nicht berücksichtigt.

Um diese Aufgabe zu unterstützen sollte die Benutzeroberfläche:

- Eine Möglichkeit bieten, das Kaufdatum des Lebensmittels zu spezifizieren, da dies leicht vergessen werden kann
- Den Benutzer bereits nach dem Kauf des Lebensmittels darüber informieren, wie ein Lebensmittel ideal gelagert werden sollte, um so die Haltbarkeit der Lebensmittel zu verlängern.

Die Rechercharbeit über Haltbarkeitsdaten zeitlich minimieren oder reduzieren, indem das System die Haltbarkeitsdaten der vorhandenen Lebensmittel anhand von Umweltfaktoren selbst ermittelt und dieses dann dem Benutzer anzeigt.

Weitere Task Scenarios wurden erstellt und sind im Anhang zu finden.

List of User Tasks

Im Folgenden werden jene Arbeitsaufgaben aufgelistet, welche die Privatperson im Moment tätigt. Aus diesen Aufgaben wird dann folgend das Task Organization Model aufgestellt.

Tasks - Allgemeine Privatperson

- Lebensmittel durchsuchen
- Lebensmittel herausuchen
- Lebensmittel aussortieren
- Lebensmittel begutachten
- Verwendungsmöglichkeiten suchen (z.B. Rezepte)
- Vergleich von Verwendungsmöglichkeiten (Zubereitungszeit, Zutaten, Vorlieben, ...)
- Mahlzeit zubereiten
- Haltbarkeit ermitteln
- Quellen vergleichen (Lagerung, angegebene Haltbarkeit,...)
- Lebensmittel verzehren

- Lebensmittel zubereiten
- Lebensmittel entsorgen

Tasks - Anbietender

- Lebensmittel verpacken (Tüte, Box, Dose,...)
- Interessenten finden
- Lebensmittel anbieten
- Lebensmittel übergeben
- Lebensmittel zum Interessenten transportieren (Öffentliche Verkehrsmittel, Auto, zu Fuß, Fahrrad,...)
- Abholort mitteilen

Tasks - Abholender

- Abholmöglichkeiten finden
- Benötigte Lebensmittel notieren
- Lebensmittel entgegennehmen
- Abholmöglichkeiten vergleichen (Angebot, Menge, Entfernung, Vorlieben, ...)
- Weg zum Angebot bestreiten (Öffentliche Verkehrsmittel, Auto, zu Fuß, Fahrrad,...)

Task Organization Model

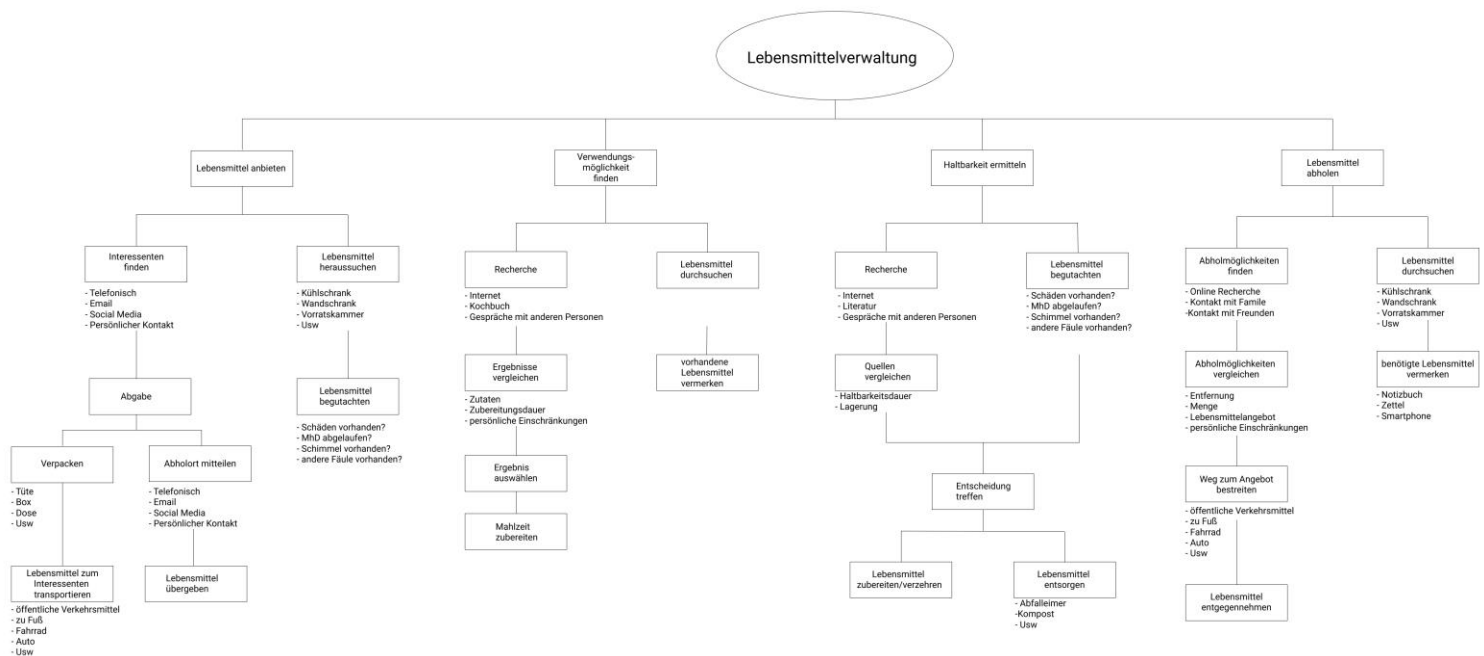


Abbildung 1. Task Organisation Model

Das hier dargestellte Task Organization Model stellt die Organisation und Abfolge der einzelnen Arbeitsaufgaben der Benutzer dar. Innerhalb des Modells wird nicht zwischen verschiedenen Rollen unterschieden, da jeder Benutzer theoretisch jede Arbeitsaufgabe erledigen könnte und es im Moment auch tut.

Der linke Zweig „Lebensmittel anbieten“ modelliert hierbei den Aufgabenablauf den ein Benutzer in der Rolle des Anbieters erledigen würde. Der rechte Zweig „Lebensmittel abholen“ modelliert den Aufgabenablauf den der Benutzer in der Rolle des Abholers erledigt. Die beiden Zweige

„Verwendungsmöglichkeit finden“ und „Haltbarkeit ermitteln“ können von jedem Benutzer unabhängig von der eingenommenen Rolle erledigt werden.

Plattform Capabilities and Constraints

In diesem Schritt soll die Zielplattform auf Beschränkungen untersucht, welche im Laufe des Designs beachtet werden müssen. Grundsätzlich kann dieser Schritt nach Mayhew, durch Verwendung eines Short Cuts übersprungen werden, jedoch wollen wir zumindest kurz auf diesen Punkt eingehen. Die Beschränkungen der Android Plattform werden im Folgenden Tabellarisch aufgeführt:

Tabelle 13: Plattform Capabilities and Constraints

Eigenschaft	Möglich	Möglich mit Aufwand	Nicht Möglich
Display Größe	2.5 -6 Zoll	-	Unter 2.5 Zoll
Bit-Mapped-Display	(.jpg, .png, .gif)	-	-
Farben	Alle Möglich	-	-
Input Devices	Touch, Digitale Tastatur, Sprachanweisung, Bewegung, Sensoren	-	-
Version	4.0 und höher	-	Kleiner als 4.0
Multitasking	x	-	-

General Design Principles

Im Folgenden werden Designprinzipien festgehalten, welche für das Projekt durchgängig befolgt werden sollen. Diese Prinzipien wurden nicht selbst neu Entwickelt, sondern entstanden einer Online Recherche und wurden Projektspezifisch ausgewählt. Mayhew empfiehlt bei diesem Schritt ebenfalls auf bekannte Prinzipien zurückzugreifen, wenn die Zeit, das Budget für eine umfangreiche Analyse oder einen externen Analysten fehlt.

Real Objects are more fun than button and menus

Dieses Prinzip besagt, dass es für den Benutzer angenehmer ist, mit Gegenständen der realen Welt zu interagieren, statt mit Buttons oder Menus. Dies ist jedoch nur an einigen Stellen Anwendbar, insbesondere wenn es um Lebensmittel, Rezepte oder Angebote geht.

Keep it Brief

Menschen tendieren dazu, lange Sätze nicht zu lesen bzw. diese nicht zu beachten. Aus diesem Grund ist es wichtig den Benutzer nicht mit Informationen zu überfluten und kurz und knapp auf den Punkt zu kommen. Dies wird, besonders in Stresssituationen relevant, da dort das Gehirn nur nach den wichtigsten Informationen sucht.

Decide for me but let me have the final say

Dies ist ein Prinzip, welches auf viele Bereiche von unserem Projekt anwendbar ist. Besonders bei der Auswahl der Lagerung, Verwendungs-Möglichkeiten oder Angeboten. Entscheidungen zu treffen bringt Aufwand und Last für den Benutzer mit sich. Diese soll dem Benutzer, wenn möglich abgenommen werden, jedoch wird ihm nicht die Möglichkeit verwehrt diese Entscheidung selbst treffen zu können.

Only show what I need when I need it

Es ist immer wichtig, aber besonders in Stresssituationen, dem Benutzer nur die für ihn relevanten Informationen zu zeigen. Dieses Designprinzip soll sich über alle Teile des User Interfaces erstrecken, um eine Wahrscheinlichkeit einer Überforderung durch zu viele Informationen zu minimieren.

(vgl. Becker, 2018)

Material Design

Das Material Design ist eine Guide Line, welche für die Entwicklung von Android Apps empfohlen wird, um ein einheitliches aussehen und eine gute Erlernbarkeit der App zu gewährleisten. Der Design Guide ist sehr umfangreich und leider nicht komplett umsetzbar, jedoch wird versucht sich an diesen so weit wie möglich zu richten. Android Benutzer sollen sich somit mit der Anwendung schneller Vertraut fühlen, da viele andere Anwendungen, wie das Betriebssystem selbst sich an diesen Designprinzipien richten. Einen eigenen kompletten Design Guide Line zu erstellen wäre in vorgegebener Zeit nicht umsetzbar. (vgl. Google Inc, 2018)

Usability Goals

Im Folgenden werden Ziele festgelegt, welche zu der Gebrauchstauglichkeit des späteren Systems beitragen sollen. Diese Ziele werden aus den zuvor erstellten User Profiles sowie der Contextual Task Analysis und allgemeinen Business Goals abgeleitet.

Die Usability Goals dienen in erster Linie den späteren Designentscheidungen sowie dem Testen dieser im späteren Verlauf des Projektes.

Die festgelegten Ziele werden im Folgenden beschrieben und priorisiert. Für die Priorisierung empfiehlt Mayhew drei Stufen zu verwenden. Bei diesen handelt es sich um:

1. Priorität -> Benötigt für die Veröffentlichung
2. Priorität -> Wichtig, aber nur falls das Erreichen nicht zu teuer oder zeitintensiv ist
3. Priorität -> Erstrebenswert aber nur bei geringen Kosten

Qualitative Goals

Die qualitativen Ziele sind jene, die den späteren Design Prozess „leiten“. Sie legen fest, was das System dem Benutzer bieten muss, um seine Arbeitsaufgaben zufriedenstellend und möglichst einfach zu erledigen.

Wie sich in den zuvor erstellten User Profiles erkennen lässt, muss bei den Zielen auf Benutzer mit verschiedenen Kenntnissen in den Bereichen Lebensmittel, Haltbarkeit, Erstellung von Angeboten sowie allgemeinen Kenntnissen im Umgang mit mobilen Endgeräten eingegangen werden. Dies führt dazu, dass das Design im Allgemeinen recht einfach und übersichtlich gestaltet werden sollte. Es sollten hierbei auf den einzelnen Screens auch nur die Funktionen im Mittelpunkt stehen, die für den

Benutzer zu diesem Zeitpunkt relevant sind. Alle weiteren angebotenen Funktionen könnten besonders bei unerfahrenen Nutzern zu Verwirrung führen, was möglichst vermieden werden sollte. Im Folgenden werden nun die konkreten qualitativen Ziele erläutert sowie deren Priorisierung begründet.

#01 Der Benutzer sollte seine gekauften Lebensmittel möglichst einfach und schnell zum Lebensmittelinventar hinzufügen können.

Priorität: 1

Die Lebensmittel, die sich im Besitz des Benutzers befinden, spielen bei der Verwendung des Systems eine tragende Rolle. Das Eintragen der Lebensmittel nachdem sie gekauft wurden sollte daher möglichst einfach und vor allem schnell gehen, da es sich in vielen Fällen um eine große Menge an Lebensmitteln handeln kann. Zusätzlich sollte das Eintragen der Lebensmittel möglichst wenig Aufwand in Anspruch nehmen da sonst die Motivation der Benutzer verloren gehen könnte.

#02 Das Lebensmittelinventar sollte dem Benutzer eine übersichtliche Darstellung der vorhandenen Lebensmittel bieten.

Priorität: 1

Je nach Benutzer kann es passieren, dass sich viele Lebensmittel gleichzeitig in dessen Inventar befinden. Möchte ein Nutzer nun die Haltbarkeit eines Lebensmittels abfragen, ein Lebensmittel anbieten oder sich eine Übersicht über die vorhandenen Lebensmittel schaffen wollen, damit keines aus Versehen vergessen wird, so ist es wichtig, dass das Lebensmittelinventar möglichst übersichtlich gestaltet ist. Zusätzlich sollten dem Benutzer Möglichkeiten geboten werden, diese nach verschiedenen Kriterien sortieren zu können oder nach konkreten Lebensmitteln im Inventar suchen zu können.

#03 Der Benutzer sollte für vorhandene Lebensmittel die Haltbarkeitsdaten im Überblick behalten können.

Priorität: 1

Für viele Menschen ist es besonders bei Obst und Gemüse oft der Fall, dass diese unerwartet ablaufen. Mithilfe des Lebensmittelinventars soll dem Benutzer zu jeder Zeit ersichtlich sein, wie lange das vorhandene Obst und Gemüse voraussichtlich noch haltbar ist, um das unerwartete Ablaufen einzuschränken. Das Haltbarkeitsdatum sollte hierzu zu jedem Lebensmittel möglichst offensichtlich angezeigt werden, sodass es nicht nötig ist, die Informationen über das Lebensmittel erst manuell abfragen zu müssen.

Viele weitere qualitative Ziele wurden zur Übersicht in den Anhang verlegt.

Quantitative Goals

Die Quantitativen Ziele sind jene objektiven Ziele, deren Erreichung später auch messbar sind. Daher spielen bei der Identifizierung dieser auch Faktoren wie „ease-of-learning“, „ease-of-use“, die generelle Performance des Benutzers und die beanspruchte Zeit eine Rolle.

Für die Definition der Ziele wurde die Vorlage nach Mayhew verwendet.

Der Faktor „ease-of-use“ bezieht sich hierbei auf die Verwendung des Systems von geübten Benutzern oder wie sie in der Vorlage auch bezeichnet werden - „Experts“. Sie benutzen das System regelmäßig und sind mit den Funktionsweisen des Systems vertraut.

Im Gegensatz dazu beschäftigt sich der Faktor „ease-of-learning“ mit jenen Benutzern (in der Vorlage „Novicen“ genannt), die neu oder sehr ungeübt in der Verwendung des Systems sind. Sie befinden sich im Lernprozess und sind mit Funktionen nur teilweise bis gar nicht vertraut.

Für die Ziele würde Aufgabenübergreifend festgelegt, dass alle Funktionen circa 1 bis 2 Durchführungen als Novice beanspruchen und ab der dritten Durchführung als Expert gelten. Natürlich handelt es sich hier nur um die Durchschnittliche Erwartung und Zielsetzung, welche in Einzelfällen auch abweichen kann.

Im Folgenden werden 6 grundlegende quantitative Ziele definiert. Dies sind nicht alle möglichen Ziele innerhalb des Projektes, jedoch würde sich auf diese Ziele beschränkt um den Aufwand etwas einzugrenzen.

Usability Goals

Goal #: 1

Task: Lebensmittel zum Inventar hinzufügen

Operational Definitions

Expert: Dritter Versuch

Novice: Erste zwei Versuche

Learn: Fehlerfreie Performance

Satisfaction: 1 = sehr unzufrieden, 4 = neutral, 7 = sehr zufrieden

Priority Definitions

1 = Benötigt für die Veröffentlichung

2 = Wichtig, falls nicht zu teuer / zeitaufwendig

3 = Erstrebenswert bei geringen Kosten

Ease-of-Learning Goals

Priority	Measure	Goal
1	Novice Time	< 2min
1	Novice Trials	3
1	Novice Errors	≤ 2

Ease-of-Use Goals

Priority	Measure	Goal
1	Expert Time	< 1min
1	Expert Errors	/

Satisfaction Goals

Priority	Measure	Goal
1	Expert	7
1	Novice	5

Abbildung 2. Quantitative Usability Goal #1

Usability Goals

Goal #: 2

Task: Haltbarkeit eines Lebensmittels anschauen

Operational Definitions

Expert: Dritter Versuch

Novice: Erste zwei Versuche

Learn: Fehlerfreie Performance

Satisfaction: 1 = sehr unzufrieden, 4 = neutral, 7 = sehr zufrieden

Priority Definitions

1 = Benötigt für die Veröffentlichung

2 = Wichtig, falls nicht zu teuer / zeitaufwendig

3 = Erstrebenswert bei geringen Kosten

Ease-of-Learning Goals

Priority	Measure	Goal
2	Novice Time	< 30 sek
1	Novice Trials	2
/	Novice Errors	/

Ease-of-Use Goals

Priority	Measure	Goal
1	Expert Time	< 10 sek
/	Expert Errors	/

Satisfaction Goals

Priority	Measure	Goal
2	Expert	7
2	Novice	5

Abbildung 3. Quantitative Usability Goal #2

Usability Goals

Goal #: 3

Task: Angebot erstellen

Operational Definitions

Expert: Dritter Versuch

Novice: Erste zwei Versuche

Learn: Fehlerfreie Performance

Satisfaction: 1= sehr unbefriedigend, 4= neutral, 7= sehr befriedigend

Priority Definitions

1 = Benötigt für die Veröffentlichung

2 = Wichtig, falls nicht zu teuer / zeitintensiv

3 = Erstrebenswert bei geringen Kosten

Ease-of-Learning Goals

Priority	Measure	Goal
1	Novice Time	<u>≤ 2 Min</u>
1	Novice Trials	<u>3</u>
1	Novice Errors	<u>2</u>

Ease-of-Use Goals

Priority	Measure	Goal
1	Expert Time	<u>< 1 Min</u>
1	Expert Errors	<u>/</u>

Satisfaction Goals

Priority	Measure	Goal
1	Expert	<u>7</u>
1	Novice	<u>4</u>

Abbildung 4. Quantitative Usability Goal #3

Usability Goals

Goal #: 4

Task: Angebot reservieren

Operational Definitions

Expert: Dritter Versuch

Novice: Erste zwei Versuche

Learn: Fehlerfreie Performance

Satisfaction: 1= sehr unbefriedigend, 4= neutral, 7= sehr befriedigend

Priority Definitions

1 = Benötigt für die Veröffentlichung

2 = Wichtig, falls nicht zu teuer / zeitintensiv

3 = Erstrebenswert bei geringen Kosten

Ease-of-Learning Goals

Priority	Measure	Goal
1	Novice Time	<u>< 30 sek</u>
1	Novice Trials	<u>2</u>
1	Novice Errors	<u>1</u>

Ease-of-Use Goals

Priority	Measure	Goal
1	Expert Time	<u>< 10 sek</u>
1	Expert Errors	<u>/</u>

Satisfaction Goals

Priority	Measure	Goal
2	Expert	<u>7</u>
2	Novice	<u>6</u>

Abbildung 5. Quantitative Usability Goal #4

Usability Goals

Goal #: 5

Task: Lebensmittelinventar aktualisieren

Operational Definitions

Expert: Dritter Versuch

Novice: Erste zwei Versuche

Learn: Fehlerfreie Performance

Satisfaction: 1= sehr unbefriedigend, 4= neutral, 7= sehr befriedigend

Priority Definitions

1 = Benötigt für die Veröffentlichung

2 = Wichtig, falls nicht zu teuer / zeitintensiv

3 = Erstrebenswert bei geringen Kosten

Ease-of-Learning Goals

Priority	Measure	Goal
1	Novice Time	<u>< 1 Min</u>
1	Novice Trials	<u>3</u>
1	Novice Errors	<u>2</u>

Ease-of-Use Goals

Priority	Measure	Goal
1	Expert Time	<u>< 20 sek</u>
1	Expert Errors	<u>/</u>

Satisfaction Goals

Priority	Measure	Goal
2	Expert	<u>7</u>
2	Novice	<u>5</u>

Abbildung 6. Quantitative Usability Goal #5

Usability Goals

Goal #: 6

Task: Informationen über Angebot einsehen

Operational Definitions

Expert: Dritter Versuch

Novice: Erste zwei Versuche

Learn: Fehlerfreie Performance

Satisfaction: 1= sehr unbefriedigend, 4= neutral, 7= sehr befriedigend

Priority Definitions

1 = Benötigt für die Veröffentlichung

2 = Wichtig, falls nicht zu teuer/zeitintensiv

3 = Erstrebenswert bei geringen Kosten

Ease-of-Learning Goals

Priority	Measure	Goal
2	Novice Time	< 30 sek
1	Novice Trials	2
/	Novice Errors	/

Ease-of-Use Goals

Priority	Measure	Goal
1	Expert Time	< 10 sek
/	Expert Errors	/

Satisfaction Goals

Priority	Measure	Goal
2	Expert	7
2	Novice	5

Abbildung 7. Quantitative Usability Goal #6

Work Re-engineering

Basierend auf den zuvor erstellten Artefakten wie dem Task Organization Model, den Task Scenarios und den User Profiles soll nun ein neues präskriptives Modell erstellt werden, welches die zukünftige Aufgabenbearbeitung darstellt. Hierbei spielt vor allem die spätere Automatisierung von vielen der Benutzeraufgaben eine große Rolle, da diese maßgeblich die Effizienz der Aufgabenbewältigung beeinflussen. Außerdem werden neben der Umstrukturierung der bestehenden Aufgaben auch neue Aufgaben in dem System ergänzt, die sich aus den Funktionen des späteren Systems ergeben. Daher handelt es sich nicht nur um ein Work Re-engineering sondern auch um ein Engineering neuer Aufgaben.

Im Folgenden werden das Task Organization Model sowie die Task Scenarios wie nach Mayhew empfohlen neu modelliert. Zusätzlich hierzu werden einige grundlegende Use Cases zu den zuvor ermittelten Anforderungen erstellt. Ebenfalls wurden die Anforderungen überarbeitet und sind im Anhang zu finden.

Task Organization Model Re-engineering

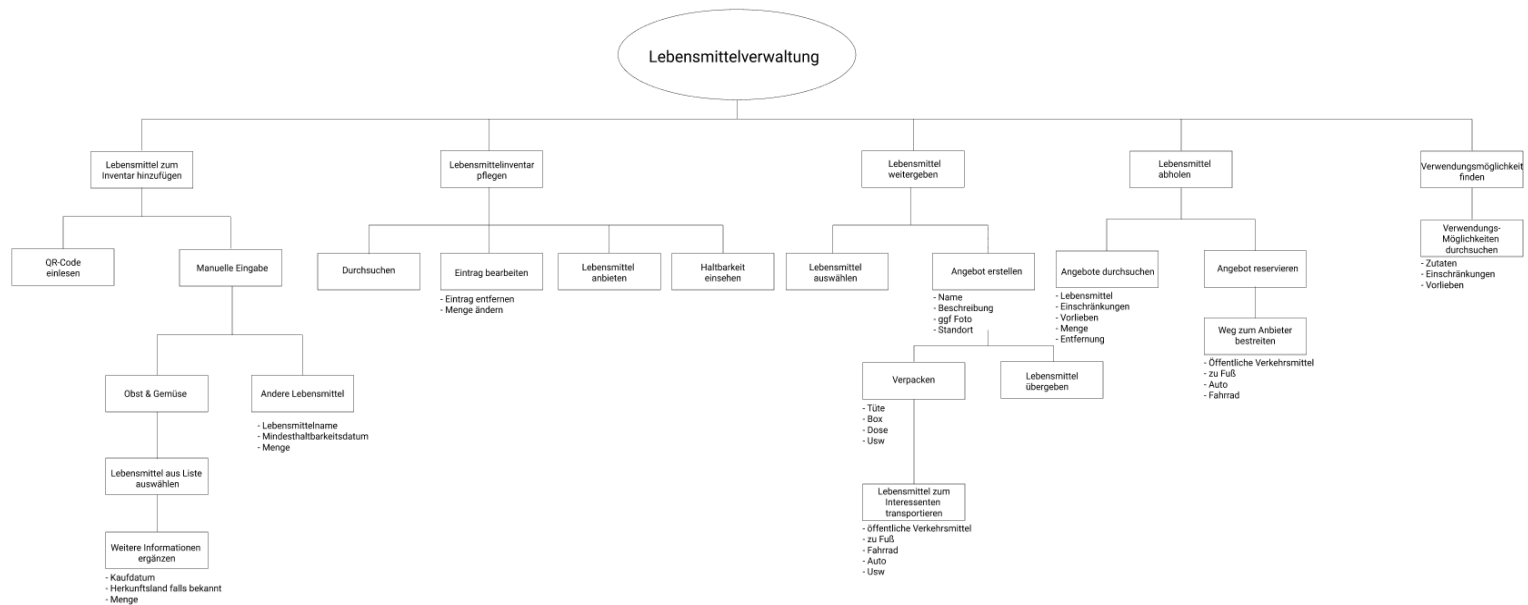


Abbildung 8. Task Organization Model Re-engineering

Im Modell des Re-engineering wird sichtbar, dass sich die Arbeitsaufgaben des Benutzers deutlich reduziert haben. Außerdem bestehen diese insgesamt aus weniger Schritten. Der Zweig der die Arbeitsaufgabe „Haltbarkeit ermitteln“ zuvor beinhaltete ist komplett entfallen. Der Grund hierfür ist, dass der Benutzer die Haltbarkeit der Lebensmittel nicht mehr manuell ermitteln muss, sondern das voraussichtliche Haltbarkeitsdatum wird direkt in der Übersicht aller Lebensmitteleinträge angezeigt. Auch die Arbeitsaufgabe „Lebensmittel anbieten“, welche im re-engineered Modell nun „Lebensmittel weitergeben“ heißt, hat an Aufwand abgenommen. Der Benutzer muss fortan nicht mehr die potentiellen Lebensmittel aus dem Kühlschrank herausuchen und diese auf Schäden untersuchen, umso eine Vorstellung über die Haltbarkeit des Lebensmittels zu erhalten. Er muss nur noch die Lebensmittel aus dem Lebensmittelinventar auswählen, sieht direkt, ob das Haltbarkeitsdatum abgelaufen ist, und kann für das Lebensmittel mit wenig Aufwand ein Angebot erstellen. Auch hier wird dem Benutzer der Aufwand des Suchens nach einem Interessenten abgenommen. Der Benutzer erstellt nur ein Angebot, welches dann von suchenden Benutzern gefunden werden kann.

Die Arbeitsaufgabe „Lebensmittel abholen“ befreit den Benutzer von der vorherigen Rechercharbeit. Der Benutzer kann die Liste aller Angebote durchsuchen und diese nach verschiedenen Kriterien filtern, um so passende Angebote möglichst schnell und einfach zu finden. Zusätzlich hat das System automatisch die Eigenschaft dem Benutzer primär Angebote anzuzeigen, die gut zu ihm passen - also seinen Einschränkungen und/oder Vorlieben entsprechen und sich möglichst in seiner unmittelbaren Nähe befinden. Zudem entfällt die Mühe, die benötigten Lebensmittel manuell zu vermerken, da der Benutzer das eigene Lebensmittelinventar direkt verfügbar hat und nach der Feststellung, dass ein Lebensmittel fehlt, dieses direkt in den Angeboten suchen kann.

Das Suchen nach Verwendungsmöglichkeiten wird dem Benutzer künftig ebenfalls erleichtert, da er nicht mehr mehrere Quellen sichten muss und Möglichkeiten mühsam vergleichen muss. Das System

ermöglicht ihm die Liste aller Verwendungsmöglichkeiten nach verschiedenen Kriterien und vorhandenen Lebensmitteln zu durchsuchen, um so direkt etwas Passendes zu finden.

Zusätzlich zu den re-engineered Aufgaben wurden auch neue Aufgaben des Benutzers ergänzt, welche mit der Verwendung des Systems anfallen. Bei diesen Arbeitsaufgaben handelt es sich um „Lebensmittel zum Inventar hinzufügen“ und „Lebensmittelinventar pflegen“.

Die Aufgabe „Lebensmittel zum Inventar hinzufügen“ ist sehr wichtig, da diese Aufgabe Voraussetzung für die sinnvolle Nutzung des Systems ist und nur so alle Funktionen richtig genutzt werden können. Das Hinzufügen kann entweder automatisch oder manuell erfolgen. Das automatische Hinzufügen erfolgt durch das Einscannen des QR-Codes auf dem Kassenbon des Benutzers, welcher er nach dem Einkaufen erhält. Diese Möglichkeit ist die einfachste und effektivste, da die QR-Codes mit denen das System geplant wird, alle Informationen enthalten, die für die Ermittlung des Haltbarkeitsdatums von Obst und Gemüse wichtig sind - nämlich das konkrete Lebensmittel, das Kaufdatum, der Herkunftsort sowie die anderen gekauften Lebensmittel, die die Haltbarkeit beeinflussen könnten.

Neben dem automatischen hinzufügen besteht außerdem die Möglichkeit ein Lebensmittel manuell zum Inventar hinzuzufügen. Dies ist besonders Hilfreich, wenn beispielsweise die Kamera des mobilen Endgerätes nicht funktioniert oder aber der Kassenbon nicht vorliegt. Wichtig für das manuelle Hinzufügen von Obst und Gemüse ist es im Idealfall von Nöten, dass der Benutzer das Herkunftsland des Lebensmittels kennt, um so die Haltbarkeitsermittlung zu unterstützen.

Die Arbeitsaufgabe „Lebensmittelinventar pflegen“ ist eine Art Überbegriff. Die Aufgabe enthält viele Unteraufgaben von denen einige ausgeführt werden können, es aber nicht müssen. Die wichtigste Aufgabe ist hierbei die Aufgabe „Eintrag bearbeiten“. Hierzu gehören Aktivitäten wie die Menge eines Lebensmittels anzupassen, wenn zum Beispiel einer von 6 eingetragenen Äpfeln verzehrt wurde. Außerdem ist es wichtig verbrauchte Lebensmittel aus dem System zu entfernen, da diese unter anderem die Haltbarkeitsermittlung anderer Lebensmittel so positiv beeinflussen können.

Task Scenario

Es wurde ein Task Scenario Re-engineering durchgeführt, welches im Anhang zu finden ist.

Use Cases

Bei den Use Cases wurden nicht zu jeder Anforderung ein Use Case erstellt, sondern es wurde versucht sich auf wenige, jedoch wichtige Use Cases zu konzentrieren.

Essential Use Cases

Tabelle 17: Essential Use Case #1 Lebensmittel verwalten

Users Intention	System Responsibility
Gekaufte Lebensmittel verwalten	Eine Möglichkeit zur Verwaltung präsentieren.
Möglichkeit zur Verwaltung auswählen	Präsentiert eine Möglichkeit die einzelnen Lebensmittel zu spezifizieren um Repräsentationen dieser zu erstellen.

Informationen zu gekauften Lebensmitteln spezifizieren	Repräsentation des Lebensmittels wird zu einer Verwaltungsmöglichkeit hinzugefügt.
--	--

Tabelle 18: Essential Use Case #2 Lebensmittel verwalten

Users Intention	System Responsibility
Lebensmittel von anderen Personen beziehen	Präsentiert Lebensmittel die von anderen Personen angeboten werden.
Informationen über ein angebotenes Lebensmittel einsehen.	Präsentiert die Informationen die über das Lebensmittel spezifiziert wurden.
Interesse daran das präsentierte Lebensmittel abzuholen äußern	Teilt anbietendem Benutzer das Interesse mit. Präsentiert dem Benutzer eine Repräsentation des Weges zu dem Angebot.
Lebensmittel nach dem Abholen in die eigene Lebensmittelverwaltung aufnehmen	Repräsentation des abgeholten Lebensmittels zu der Lebensmittelverwaltung des Benutzers hinzufügen.

Tabelle 19: Essential Use Case #2 Lebensmittel verwalten

Users Intention	System Responsibility
Nicht mehr benötigtes Lebensmittel anderen Personen weitergeben.	Präsentiert eine Möglichkeit zur Weitergabe des Lebensmittels
Informationen über das Lebensmittel spezifizieren	Präsentiert eine Möglichkeit zur Spezifikation von Informationen über das Lebensmittel.
Lebensmittelangebot anderen Menschen mitteilen.	Präsentiert die Repräsentation des Lebensmittelangebotes anderen Benutzern. Informiert den Benutzer über das Interesse anderer Benutzer.
Lebensmittel nachdem es abgeholt wurde aus der persönlichen Lebensmittelverwaltung entfernen.	Entfernt die Repräsentation des Lebensmittels aus der Lebensmittelverwaltung.

Ebenfalls wurden Concrete Use Cases erstellt. Diese wurden zur Übersicht in den [Anhang](#) verlagert.

Conceptual Model Design

Orientierung des Systems

Bei der Entscheidung zwischen Prozess- und Produktorientierung ist das System eindeutig der Produktorientierung zuzuordnen. Die Begründung dafür ist, dass innerhalb des Systems klar definierte Produkte in Form von Lebensmitteleinträgen und Angeboten von den Benutzern erstellt, bearbeitet und gespeichert werden.

Identifikation der Produkte

Lebensmitteleintrag im Inventar

Ein Lebensmitteleintrag muss mit verschiedenen Attributen versehen sein damit das System mit diesem arbeiten kann. Diese Attribute können entweder manuell von dem Benutzer spezifiziert werden oder automatisch von den QR-Codes auf den Kassensbons übernommen werden. Beide Arten werden im Folgenden als „Userinput“ bezeichnet.

Folgende Attribute sind bei der Erstellung eines Lebensmitteleintrages wichtig:

Userinput

- Name des Lebensmittels: Wie wird das Lebensmittel im Inventar bezeichnet. Um welches Lebensmittel handelt es sich?
- Art des Lebensmittels: Abgepacktes Lebensmittel mit vorgegebenem Mindesthaltbarkeitsdatum oder frische Lebensmittel in Form von Obst und Gemüse
- Haltbarkeitsdatum (falls das Lebensmittel abgepackt war): Wie lange ist das Lebensmittel voraussichtlich genießbar?
- Kaufdatum: Wann wurde das Lebensmittel gekauft?
- Herkunftsland: Woher kommt das Lebensmittel? Wie lange hat der Transport nach Deutschland voraussichtlich benötigt.
- Menge: Wieviel von dem Lebensmittel ist vorhanden? Wurde ein einzelner Apfel gekauft oder ein Netz mit 10 Äpfeln?

Weitere Attribute

Bei den weiteren Attributen handelt es sich um weitere Attribute die einen Lebensmitteleintrag identifizieren. Diese Attribute werden jedoch nicht von dem Benutzer spezifiziert, sondern von dem System ermittelt oder festgelegt.

- Haltbarkeitsdatum (von Obst und Gemüse): Wie lange ist das Lebensmittel voraussichtlich haltbar?
- Health Label: Welchen Einschränkungen oder Ernährungstypen entspricht das Lebensmittel? (Ist das Lebensmittel Vegetarisch oder Vegan? Ist es Laktosefrei?)

Lebensmittelangebot

Ein Lebensmittelangebot wird im Grundlegenden mit den gleichen Attributen wie ein Eintrag im Inventar identifiziert, da ein Angebot genau genommen nur ein angebotener Lebensmitteleintrag aus dem Inventar ist. Jedoch werden für ein Angebot noch einige weitere Attribute benötigt. Diese werden im Folgenden aufgeführt.

Userinput

Ein Angebotener Lebensmitteleintrag muss von dem Benutzer noch um einige Attribute ergänzt werden, damit das Angebot veröffentlicht werden kann. Diese Attribute sind:

- Foto: Ein Foto des angebotenen Lebensmittels ermöglicht es potentiellen Interessenten sich einen besseren Eindruck über das Angebot zu verschaffen, da eine visuelle Präsentation des angebotenen Lebensmittels vorliegt und Schäden so früh erkannt werden können.

- Beschreibung: Eine Beschreibung des Angebotes kann oft hilfreich sein um weitere als wichtig erachtete Informationen über das Angebot an potentielle Interessenten mitzuteilen.
- Abholzeit: Wann kann das Lebensmittel abgeholt werden?

Weitere Attribute

Folgende Attribute des Angebotes werden von dem System ermittelt:

- Abholwert (für jeden Interessenten individuell berechnet): Wie gut passt das Angebot zu dem Benutzer? Entspricht es den Vorlieben oder Einschränkungen? Wie weit ist es entfernt?
- Standort: Die Position des anbietenden Benutzers welche anhand von Längen- und Breitengrad ermittelt wird.

Design Regeln

Produktpräsentation

Im Folgenden wird festgelegt, wie die einzelnen Produkte innerhalb des Systems dargestellt werden. Hierbei wird einmal eine Listendarstellung für die Anzeige mehrerer Produkte und einmal die Detailansicht eines einzelnen modelliert.

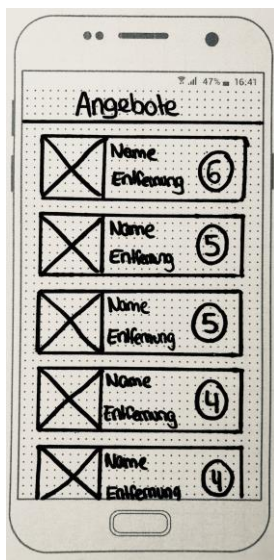


Abbildung 9.
Listendarstellung Angebote

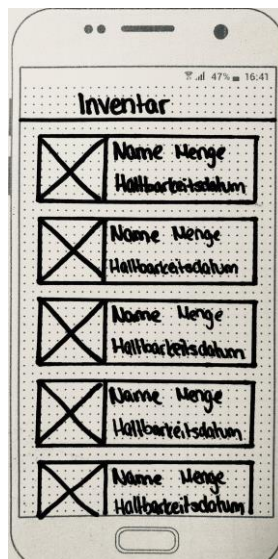


Abbildung 10.
Listendarstellung Inventar

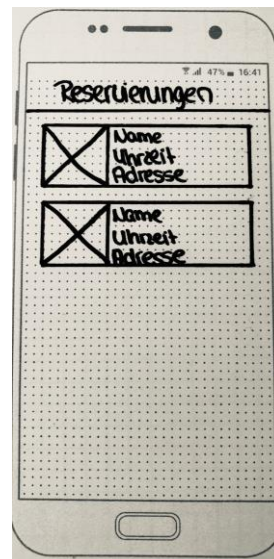


Abbildung 11. Listendarstellung
Reservierungen

Bei der Präsentation der Produkte Angebote (auch in Form von Reservierungen, da diese ein vermerktes Angebot sind) und Lebensmitteleinträgen innerhalb einer Liste wurde für alle Produkte die gleiche Darstellung gewählt. Hierbei werden sowohl ein kleines Bild als auch wichtige Textfelder zu dem jeweiligen Produkt angezeigt. Hierbei handelt es sich um jene Informationen die der Benutzer auf den ersten Blick sehen sollte und nicht darauf angewiesen ist die Detailansicht zu öffnen. Bei den Einträgen im Lebensmittelinventar ist hierbei wichtig, um welches Lebensmittel es sich handelt, wie oft es vorhanden ist und das vom System ermittelte Haltbarkeitsdatum.

Bei den Angeboten ist der Name, die Entfernung und der Abholwert wichtig. Bei den Reservierungen ist der Name, die vom Anbieter angegebene Abholzeit und die genaue Adresse von Relevanz.

Für die Lebensmitteleinträge ist keine Detailansicht notwendig, da alle relevanten Informationen zu den Einträgen bereits in der Übersicht angezeigt werden. Für die Angebote und Reservierungen sind folgende Präsentationen vorgesehen.



Abbildung 12.
Detailansicht Angebot



Abbildung 13. Detailansicht
Reservierung

In der Detailansicht sind die Informationen über die Angebote und Reservierungen nach ihrer Relevanz geordnet. Bei den Angeboten sind auf den ersten Blick ein Foto, der Name, eine kurze Beschreibung sowie der präferierte Abholzeitpunkt des Anbieters ersichtlich. Zudem sind noch weitere Details wie die angebotene Menge, das Haltbarkeitsdatum, das Health Label und die Entfernung ersichtlich. Außerdem wird dem Benutzer ermöglicht das Angebot zu reservieren oder aber den Anbieter per Nachricht bei Fragen zu kontaktieren.

Bei Reservierungen wird kein Foto mehr angezeigt, sondern anstatt dessen eine Karte auf der der Standort des Anbieters vermerkt ist. Außerdem werden die Adresse noch einmal textuell und der Abholzeitpunkt angezeigt. Von der Reservierung aus ist es dem Benutzer zudem möglich zum ursprünglichen Angebot zu gelangen um dort wieder alle anderen Details über das Angebot nachlesen zu können. Der Benutzer hat hier außerdem erneut die Möglichkeit den Anbieter zu kontaktieren oder aber die Möglichkeit die Reservierung des Angebotes zurück zu ziehen.

Navigation

Zusätzlich zu den Regeln zur Produktpräsentation werden auch Regeln benötigt, die die Navigation zwischen den verschiedenen Funktionen des Systems festlegen.



Abbildung 14. Bottom Navigation



Abbildung 15. Hamburger Menü

Für die Navigation würde eine Kombination aus Hamburger Menü und Bottom Navigation gewählt, da beide Navigationsmethoden aussagekräftige Vorteile mit sich bringen.

Beide der Navigationsmethoden werden von dem zuvor ausgewählten Material Design unterstützt was bereits einen großen Vorteil darstellt.

Der Hauptgrund für die Entscheidung zu einer Kombination beider Navigationsmethoden waren die verschiedenen Work Environments in denen Benutzer ihre Arbeitsaufgaben erledigen. Zum einen im eigenen Heim wo man in der Regel ungestört ist und sich vertraut und sicher fühlt, und zum anderen in der Öffentlichkeit wo man leicht Stress ausgesetzt werden kann, leicht abgelenkt werden kann und man oft nicht viel Zeit zur Verfügung hat.

Die Bottom Navigation erfordert eine Lernfähigkeit des Nutzers, da dieser während den ersten Benutzungen zuerst herausfinden muss, was die einzelnen Symbole bedeuten und wo sie ihn hinführen. Zwar stellt dies Anfangs eine kleine Hürde dar, da eventuell nicht jedem Benutzer die Navigation direkt klar wird. Sobald der Benutzer jedoch mit den Symbolen vertraut ist, ist die Bottom Navigation von großem Vorteil, da sie die Benutzer mit nur einem Klick zu der benötigten Funktion führt. Dies ist besonders in der Öffentlichkeit und in Stresssituationen von Vorteil, wenn ein Benutzer zum Beispiel im Supermarkt schnell im Inventar nachsehen möchte, ob er ein bestimmtes Lebensmittel nachkaufen muss. Der Benutzer muss nicht erst nach dem Inventar „suchen“ sondern hat dies direkt mit einem Klick verfügbar.

Ein grundlegender Nachteil der Gegen die Verwendung von der Bottom Navigation als Hauptnavigation spricht ist jedoch die Begrenzung der Funktionen. Die Bottom Navigation ist nur für 2 bis 5 Funktionen geeignet, alles andere muss auf eine andere Art und Weise erreicht werden. So ist die Navigation zwischen den Funktionen die in der Navigationsleiste angezeigt werden zwar einfach und schnell, aber alle anderen Funktionen sind viel unübersichtlicher und schwerer zu erreichen. Da die Bottom Navigation jedoch sehr viel Sinn für die Benutzung außerhalb des eigenen Heimes macht, wurden die wichtigsten Funktionen die primär in der Öffentlichkeit benötigt werden herausgefiltert und in Form einer Bottom Navigation dargestellt.

Das Hamburger Menü hat zwei grundlegende Vorteile. Zum einen ist es sehr platzsparend, was die Benutzung des Systems aufgrund von weniger Ablenkungen einfacher gestalten kann. Das Menü wird nur bei Bedarf geöffnet, zu jedem anderen Zeitpunkt ist es kompakt und unauffällig. Der zweite Vorteil ist, dass es mehr Funktionen und Menüpunkte beinhalten kann, was in den meisten Systemen benötigt wird. Dieser Vorteil bringt jedoch auch den Nachteil mit sich, dass das Menü durch zu viele Funktionen unübersichtlich erscheint oder bei einer Liste mit eng aneinander stehenden Funktionen in Stresssituationen leicht aus Versehen die falsche Funktion ausgewählt werden könnte. Außerdem

benötigt die Navigation zwischen zwei Funktionen mehr Klicks als bei der Bottom Navigation. Möchte ein Benutzer zum Beispiel von den Angeboten in sein Inventar wechseln, um zu sehen ob er das angebotene Lebensmittel selbst zuhause hat, und dann wieder zurück zu den Angeboten wechseln, so würde er bei dem Hamburger Menü 4 Klicks benötigen während es bei der Bottom Navigation nur 2 wären.

Trotzdem ist das Hamburger Menü eine sehr sinnvolle Navigationsmethode, welche im Falle des Systems gut für sekundäre Funktionen genutzt werden kann. Diese sekundären Funktionen werden in der Regel eher seltener genutzt und wenn sie genutzt werden, so ist es wahrscheinlicher, dass dies nicht in einer Stresssituation, sondern eher im vertrauten Eigenheim passiert.

Screen Design Standard

Im Folgenden werden genauere Richtlinien für die Gestaltung des Systems festgelegt. Da bereits in den General Design Principles das Material Design als Guideline festgelegt wurde, wird dieses auch hier als Basis für die Gestaltung der verschiedenen Elemente verwendet. Außerdem werden die Schriftart und die Schriftgröße sowie die Farbgebung des Systems bestimmt und festgehalten.

Tabelle 23: Screen Design Standard

Schriften	<p>Schriftart: Roboto</p> <ul style="list-style-type: none"> • Medium für die App Bar Titel & Buttons • Regular für den Text <p>Schriftgröße:</p> <ul style="list-style-type: none"> • 12 pt für kleinen Text • 14pt für normalen Text • 16pt für Titel • 18pt Schrift für Dialog Box Titel • 20pt für App Bar Titel <p>Schriftfarbe:</p> <ul style="list-style-type: none"> • #ffffff Weiß • #000000 Schwarz (100% Deckkraft & 53,9% Deckkraft)
Farben	<p>Primär</p> <p>#F79862 Muted Orange Regulär</p> <p>#f57000 Orange Dunkel</p> <p>#f7f3f0 Bottom Navigation</p> <p>Text auf Primär</p> <ul style="list-style-type: none"> • #ffffff Weiß • #000000 Schwarz <p>Diese Farben werden für die wichtigen Elemente wie zum Beispiel die Buttons des Material Designs genutzt.</p>
Hintergrund	<p>Primäre Hintergrundfarbe</p> <ul style="list-style-type: none"> • #ffffff Weiß <p>Sekundäre Hintergrundfarbe</p> <ul style="list-style-type: none"> • #E5E5E5 helles Grau

Control Standards	Als Buttons werden die im Material Design festgelegten „Contained Buttons“ verwendet, da diese eine gute Erkennbarkeit aufweisen. Bei anderen Control Standards wie Checkboxes, Switches und Ähnlichem wird sich ebenfalls nach dem Material Design gerichtet.
Dialog Box Standards	Bei den Dialog Boxen richten wir uns ebenfalls nach den vorgegebenen Standards, wobei an dieser Stelle die simple Dialogs und die Full-Screendialogs am wichtigsten sind.
Message Box Standards	Message Boxen sind in unserer Applikation als Notifications zu verstehen, welche sich ebenfalls an die Vorgaben des Material Designs richten. Diese sollten dem Benutzer bei einem gesperrten Screen angezeigt werden und bestehen aus einem Titel, einer unterliegenden Beschreibung und einem zusätzlichen large Icon, welches den Inhalt wiedergeben soll.
Feedback Standards	An dieser Stelle wird sich ebenfalls an dem Material Design gerichtet und Feedback in Form von Snackbars am unteren Bildbereich ausgegeben. Dieser soll sich über der Navigation befinden und diese nicht verändern.

Nach Mayhew sollten alle diese Angaben mit zusätzlichen Bildern versehen werden, da diese mehr sagen als Worte. An dieser Stelle verweisen wir an den gewählten Design Guide, da dort bereits alles dokumentiert und visualisiert worden ist.

Detailed User Interface Design

Für das Detailed User Interface Design wurden Prototypen der wichtigsten Screens erstellt. Hierbei lag der Fokus auf der Präsentation von Angeboten, Lebensmittelinventareinträgen und Reservierungen sowie den Dialog Boxen, der Navigation und dem allgemeinen Design des Systems.

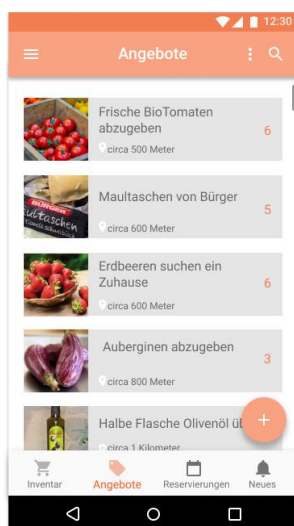


Abbildung 16. Detailed User Interface Angebote

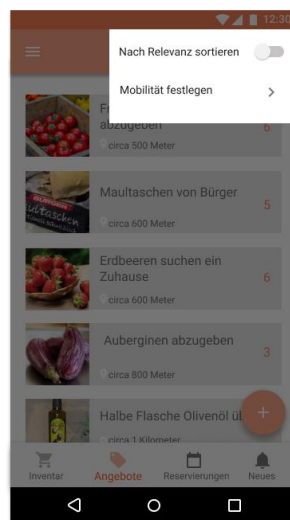


Abbildung 17. Detailed User Interface Angebote Mobilität

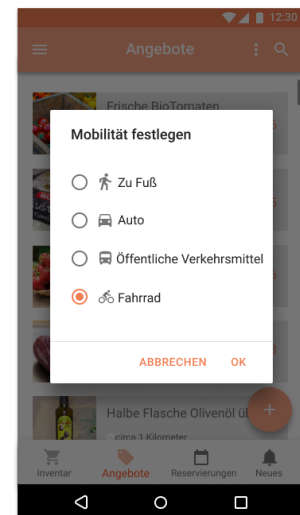


Abbildung 18. Detailed User Interface Dialog

Angebote

Die Auflistung der Angebote ist bereits innerhalb der Produktpräsentation festgelegt worden. Der Benutzer hat die Möglichkeit sich die Angebote nach deren Relevanz anzeigen zu lassen und so primär die Angebote aufgezeigt bekommt, die einen hohen Abholwert aufweisen. Außerdem kann der Benutzer für jede Suche manuell angeben, wie mobil er ist oder wie er vorhat das Angebot abzuholen. Anhand der Mobilität wird, wie im WBA2 Abschnitt beschrieben, der Abholwert für den Benutzer angepasst, sodass primär Rezept angezeigt werden, die beispielsweise gut zu Fuß zu erreichen sind.

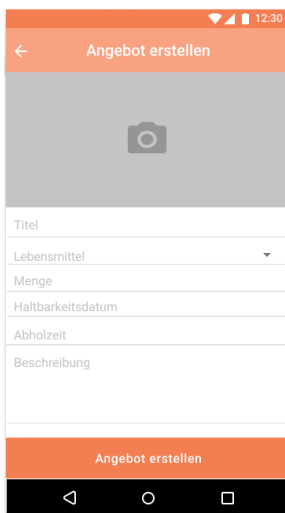


Abbildung 19. Detailed User Interface Angebot erstellen

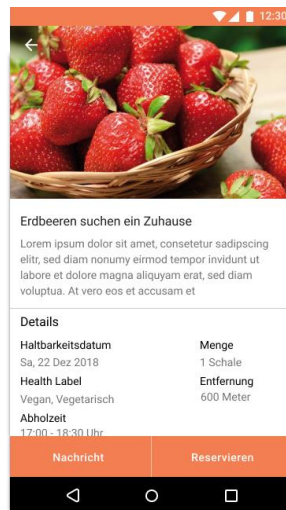


Abbildung 20. Detailed User Interface Angebot Detailansicht

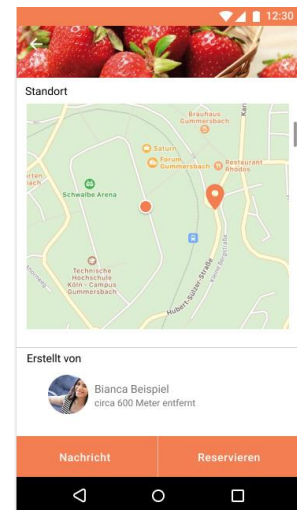


Abbildung 21. Detailed User Interface Angebot Detailansicht Fortsetzung

Bei der Detailansicht eines Angebotes wurde entschieden, dass besonders Bilder und Karten eine große Rolle spielen, da diese viel ansprechender sind als reiner Text. Zusätzlich wurde innerhalb des Angebots noch das Profil des Anbieters ergänzt. Bei dem Screen „Angeboterstellen“ wurden die auszufüllenden Felder mit Platzhaltern gefüllt, um den Benutzern so klar zu sagen, welche Angaben sie machen müssen.

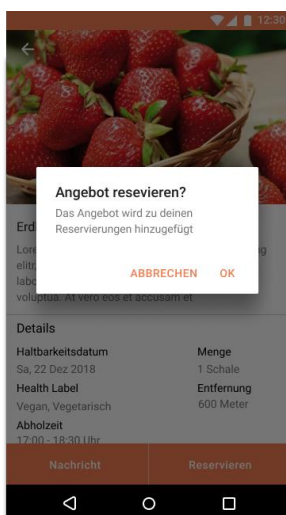


Abbildung 22. Angebot Reservieren Dialog

Will ein Benutzer ein Angebot reservieren, so öffnet sich eine Dialog Box, welche um eine erneute Bestätigung der Reservierung bittet. Dies soll dazu beitragen, dass verhindert wird, dass Benutzer ein Angebot aus Versehen reservieren ohne dies zu merken.

Lebensmittelinventar

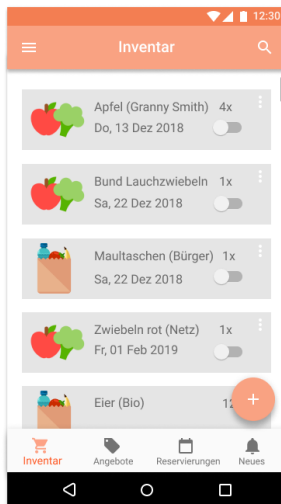


Abbildung 23. Detailed User Interface Lebensmittelinventar



Abbildung 24. Detailed User Interface Lebensmittelinventar Dialog1

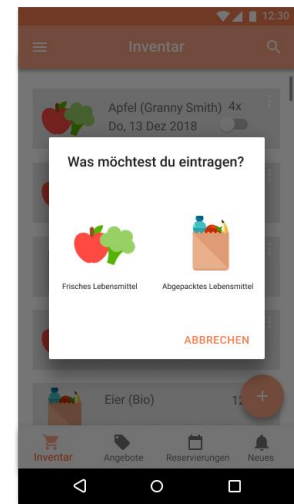


Abbildung 25. Detailed User Interface Lebensmittelinventar Dialog2

Auch die Auflistung der Inventar-Einträge wurde bereits in der Produktpräsentation festgelegt. Für die Unterscheidung zwischen Frischen und Abgepackten Lebensmitteln werden hier zwei verschiedene Icons angezeigt. Außerdem wurde ein Schalter ergänzt, mit dem der Benutzer spezifizieren kann, ob er das Lebensmittel im Kühlschrank lagert oder außerhalb des Kühlschranks. Außerdem wurde eine Schaltfläche für Optionen ergänzt, die Prototypen für das Öffnen der Optionen sind in Abbildung 27 zu sehen.

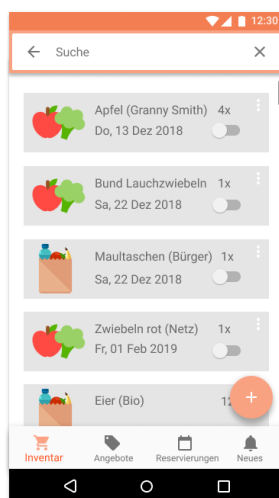


Abbildung 26. Detailed User Interface Lebensmittelinventar Suchleiste

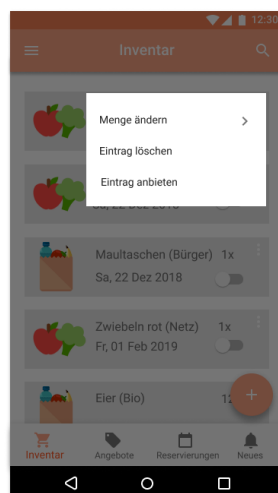


Abbildung 27. Detailed User Interface Lebensmittelinventar Optionen

Wird der Button zum Lebensmittel hinzufügen betätigt, so wird der Benutzer zuerst gefragt, ob er einen QR Einscannen möchte, oder das Lebensmittel manuell eintragen möchte. Hier wurde neben

Titeln auch mit Bildern gearbeitet, um so die Anschaulichkeit des Systems zu unterstützen. Wählt der Benutzer die Manuelle Eingabe, so muss er zwischen frischen oder abgepackten Lebensmitteln wählen.

Das Eingabe Fenster selbst wurde nicht noch einmal modelliert, da dieses grundlegend aussieht wie das Eingabe Fenster zum Erstellen eines Angebotes.

Der Benutzer hat außerdem die Möglichkeit die Menge eines Lebensmitteleintrages zu ändern, den Eintrag zu löschen, wenn das Lebensmittel verbraucht wurde und das Lebensmittel anzubieten. Der Punkt „Eintrag anbieten“ führt direkt zu dem bereits gezeigten Screen „Angebot erstellen“
Außerdem hat der Benutzer innerhalb des Inventars die Möglichkeit nach einem bestimmten Eintrag konkret zu suchen indem er auf die Lupe klickt.

Reservierungen

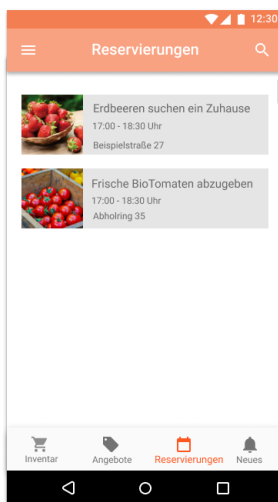


Abbildung 28. Detailed User Interface Reservierungen

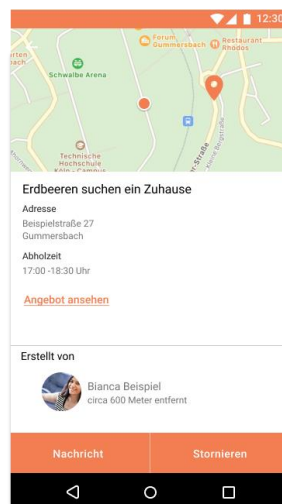


Abbildung 29. Detailed User Interface Reservierungen Details

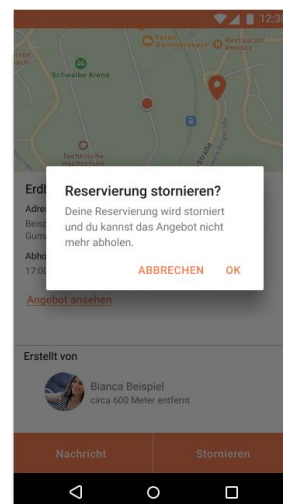


Abbildung 30. Detailed User Interface Reservierungen Stornieren

Auch die Präsentation der Reservierungen wurde bereits zuvor festgelegt. Bei der Detailansicht der Reservierungen wurde zusätzlich das Profil des Anbieters ergänzt, um so auf dieses zugreifen zu können. Der ursprünglich geplante Button für das Zurückkehren zum originalen Angebot wurde durch einen platzsparenden Link ersetzt. Außerdem kann der Benutzer die Reservierung stornieren. Bevor dies passiert wird er jedoch anhand von einer Dialogbox noch einmal zur erneuten Bestätigung der Stornierung aufgefordert, um so zu verhindern, dass ein Benutzer eine Reservierung aus Versehen storniert.

Die Screens für das Nachrichten schreiben an andere Benutzer wurde vorerst nicht modelliert, da dies eher eine sekundäre Funktion ist und nicht ausschlaggebend für die Veröffentlichung des Systems wäre, da ein Benutzer nur in Ausnahmefällen einen Anbieter kontaktieren müsste.

Hamburger-Menü und Ernährungsprofil

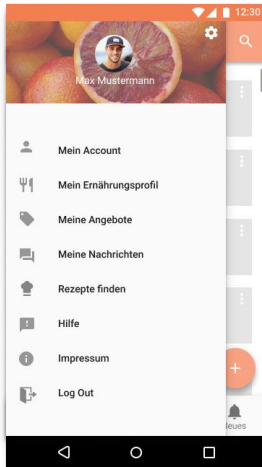


Abbildung 31. Detailed User Interface Hamburger Menü

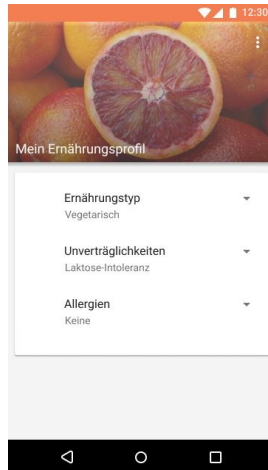


Abbildung 32. Detailed User Interface Einschränkungen

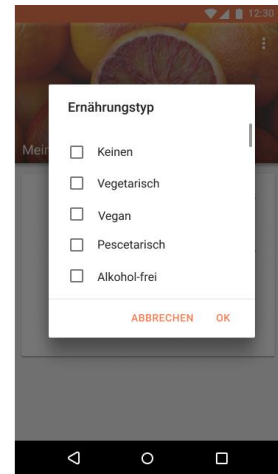


Abbildung 33. Detailed User Interface Einschränkungen Dialog

Neben der Bottom Navigation wurde zusätzlich noch das Hamburger Menü modelliert. Dieses beinhaltet die Funktionen die nicht so oft benötigt werden wie andere. Damit die Angebote passend auf den Benutzer zugeschnitten werden können, sollte dieser sein Ernährungsprofil ausfüllen. Innerhalb des Ernährungsprofils kann er seinen Ernährungstyp, Unverträglichkeiten und Allergien angeben. Bei der Auswahl einer der Optionen, wie hier zum Beispiel Ernährungstyp, öffnet sich eine Dialog Box. Innerhalb von dieser Dialog Box hat der Benutzer die Möglichkeit die Eigenschaften auszuwählen, die auf ihn zutreffen.

Evaluation

Für die Evaluation wurden 3 verschiedene Evaluations-Methoden ausgewählt.

Für das Testen der Gebrauchstauglichkeit werden hierbei die 10 Heuristiken nach Nielsen verwendet. Zusätzlich wird die Methode „Cognitive Walkthrough“ verwendet, da durch diese Methode sehr gut die Erfüllung der zuvor ermittelten Benutzer Anforderungen analysieren lässt, sowie mögliche Probleme in der Erfüllung der Arbeitsaufgaben aufzeigen.

Abschließend wurde noch die Methode „Think Aloud“ mit verschiedenen Test-Benutzern durchgeführt, um so weitere mögliche Usability Probleme herauszufiltern.

Benutzer

Als Benutzer werden die Personengruppen „Privatperson“, „Anbieter“ und „Abholer“ aus den User Profiles betrachtet. Hierbei wurden jeweils die Charakteristika des ersten User Profile der jeweiligen Gruppen verwendet.

Für die Think Aloud Methode wurden 2 Test-Benutzer befragt, die jeweils die Aufgaben aller 3 Benutzergruppen erledigen sollten, da auch im realen Gebrauch des Systems alle Benutzer alle Rollen einnehmen können und sollten.

Aufgaben

Als Aufgaben werden die in den Reengineered Task Scenarios und den präskriptiven Use Cases festgelegten Aufgaben „Lebensmittel zum Inventar hinzufügen“, „Lebensmittel anderen Personen anbieten“ und „Lebensmittel von anderen Personen abholen“ verwendet. Zu diesen Aufgaben wird das „Cognitive Walkthrough“ durchgeführt, wobei die folgenden vier vordefinierten Fragen gestellt und beantwortet werden.

- will the user try to achieve the right effect?
- will the user notice that the correct action is available?
- will the user associate the correct action with the effect that the user is trying to achieve?
- if the correct action is performed, will the user see the progress is being made towards the solution of the task?

Parallel dazu wird getestet, ob keine der 10 Heuristiken nach Nielsen verletzt werden.

Durchführung

Die Durchführung der Evaluation mit der zusätzlichen Think Aloud Methode ist im Anhang zu finden.

Ergebnis der Evaluation

Die Evaluation ergab, dass einige Funktionen eventuell klarer benannt oder gekennzeichnet werden sollten. Einige Bezeichnungen wie zum Beispiel bei dem Anbieten eines Lebensmittels aus dem Inventar sollten klarer als diese gekennzeichnet sein, da es sonst bei den Benutzern zu Verwirrung kommen kann. Auch sollten einzelne Informationen wie zum Beispiel der Abholwert klarer benannt werden, da vor allem dieser bei den Benutzern zu Missverständnissen führte und die Benutzer den Abholwert mit der Menge der angebotenen Lebensmittel verwechselten.

Da es sich bei den herausgestellten Problemen primär um Benennungsprobleme handelte, wurde hierfür keine zusätzliche Design Iteration vorgenommen. Trotzdem wurden die zu verbessernden Bezeichnungen vermerkt und in der Implementierung entsprechend angepasst. (vgl. Nielsen, 1995)

Abschnitt Webbasierte Anwendungen

Begründung der gewählten Architektur

Für das Projekt musste eine angemessene Architektur gewählt werden, wobei an dieser Stelle nach einem ersten Ansatz im Konzept zwischen zwei Alternativen nachträglich abgewägt wurde. Zum einen eine klassische Client-Server Architektur oder eine Peer2Peer Architektur. Im Folgenden wird eine kurze Pro- und Contra Liste der einzelnen Alternativen projektspezifisch durchgeführt.

Entwicklung mit Client-Server Architektur:

Vorteile

- Viele Clients können mit einer hohen Sicherheit bzw. geringen Ausfallsrate bedient werden. Dies ist in dem Bezug auf unsere Ziele der Abfall-Reduktion wichtig, da dies ansonsten den Benutzer stark demotivieren könnte.
- Zentralisierte Ressourcen sind für unser Projekt essentiell, da im übergeordneten Sinne ein Handelssystem erstellt werden soll.
- Grundsätzlich ist die Integration neuer Clients einfach, wodurch die Abfallreduktion ebenfalls gestärkt wird.
- Es liegt eine gute Grundlage für eine hohe Sicherheit vor, da es eine Schnittstelle zu dem System gibt, und somit grundsätzlich wenige Zugangspunkte bestehen, welche abgesichert werden müssen.
- Es liegt bereits Erfahrung durch vorangegangene Module (WBA2, Betriebssysteme) mit dieser Architekturform vor.
- Eine Anforderung an dieses Modul ist es ein verteiltes Modul mit verteilter Anwendungslogik zu schaffen. Dies ist durch die Client-Server Aufteilung der Komponenten gut umsetzbar.

Nachteile

- Die Entwicklung einer Client-Server Architektur ist aufwändig und erfordert eine umfangreiche Planung.
- In einem echten Projekt würden Kosten für den Server und für die Datenbank entstehen, welche an dieser Stelle zwar vernachlässigt werden, aber nicht unbeachtet bleiben sollen.

Entwicklung mit Peer 2 Peer Architektur

Vorteile

- Es wird keine zusätzliche Hardware benötigt und somit würden Kosten wegfallen. Dies könnte in einem realen Projekt, welches auf den Markt kommt ein Argument mit hohem Gewicht sein, da keine Einnahmen generiert werden.
- Es ist einfach spontane, sichere und stabile Verbindungen zwischen Clients aufzubauen. Dies stellt in unserem Kontext einen Vorteil dar, da Benutzer und deren Interaktion miteinander im Fokus stehen.

Nachteile

- Keine Möglichkeit einer zentralen Datensicherung und umfangreichen Zugriff (vieler Benutzer gleichzeitig) auf gemeinsame Ressourcen. Dies ist ein sehr entscheidender Nachteil, da wir grundsätzlich ein Handelssystem erstellen möchten.

- Der einzelne Peer ist im übertragenden Sinne sowohl Client als auch Server. Dies macht es schwer möglich eine verteilte Anwendungslogik zu implementieren.
- Im Vergleich zu Client Server Architekturen fehlt hier der Aspekt der Zuverlässigkeit und es ist keine Administration möglich.
- Es besteht keine Erfahrung mit der Entwicklung von Peer 2 Peer Architekturen.

Fazit: Nach der Gegenüberstellung beider Alternativen wurde sich auf eine Client–Server Architektur geeinigt. Diese stellt für unser Projekt die bessere Alternative dar, da Ressourcen zentral verwaltet werden können und eine hohe Anzahl an Anwendern somit bedient werden kann. Ebenfalls ist eine verteilte Anwendungslogik gut in diese Architektur implementierbar und die Nachteile sind teilweise in unserem Projektkontext vernachlässigbar. Ein Peer 2 Peer Netzwerk eignet sich gut, um Systeme mit File-Sharing Funktionen oder ähnlichem zu entwickeln, dies ist jedoch für unseren Projektkontext weniger hilfreich.

Begründung der gewählten Programmiersprachen

Grundsätzlich ist eine Vorgabe des Projektrahmens, dass das Projekt mit den Programmiersprachen Java und/oder JavaScript realisiert werden soll. An dieser Stelle wird ebenfalls nachträglich in kurzer Form zwischen diesen Alternativen auf Client und Server Seite abgewogen.

Clientseite

Grundsätzlich besteht die Möglichkeit den Client in Java oder JavaScript zu realisieren. Da der Nutzer in unserem Projekt mobil ist, wie in der Work Environment Analysis ergeben und somit das System auf einem mobilen Endgerät laufen soll, stehen hier die zwei Alternativen zur Verfügung. Zum einen eine reine Webapp, welche auf dem mobilen Endgerät aufgerufen wird und in JavaScript programmiert wird, oder eine native App in Java basierend auf dem Android Betriebssystem. Da wir jedoch Standortinformationen der Benutzer benötigen und mit Bildern arbeiten und diese lediglich durch Realisierung einer nativen App, durch den Zugriff auf Ressourcen des Betriebssystems möglich ist, ist die Wahl auf Java für den Client gefallen.

Serverseite

Hier besteht ebenfalls die Möglichkeit den Server in Java oder in Javascript zu implementieren. Da hier kein eindeutiges Kriterium vorliegt, wird zwischen beiden Alternativen abgewogen.

Entwicklung mit Java

Vorteile:

- Große Community
- Große Auswahl an Librarys
- Effizient im Hinblick auf CPU intensive Berechnungen
- Große Auswahl an IDEs

Nachteile:

- Von der Implementierung sehr zeitaufwendig
- Effizienz stark von „clean Code“ abhängig
- Keine Vorerfahrung mit der Erstellung von Servern mit Java

Entwicklung mit Javascript

Vorteile:

- Schneller Entwicklungsprozess
- Einfache Implementierung
- Große Auswahl an Modulen
- Große Community
- Bereits vorhanden Erfahrung
- Gut skalierbar auf eine hohe Benutzerzahl

Nachteile:

- Technologie im Gegensatz zu Java noch nicht so ausgereift
- Wenig gute Entwicklungsumgebungen

Fazit: Da unser Vorgehen aus einer iterativen Vorgehensweise besteht und wir einen schnellen Entwicklungsprozess benötigen, welcher auf eine hohe Benutzerzahl skalierbar sein muss haben wir uns für die Entwicklung mit JavaScript und Node.js entschieden. Die überwiegenden Vorteile von Javascript und Node.js und die erworbene Erfahrung bestärken ebenfalls diese Vorgehensweise.

System Schnittstelle

In diesem Abschnitt wird zwischen verschiedenen Alternativen für die Schnittstelle des System abgewägt. Grundsätzlich beziehen wir POX, URI-Tunneling und ähnliche REST-Vorgänger nicht in die Auswahl ein, da alleine die eindeutige Notwendigkeit von verschiedenen Ressourcen diese bereits ausschließt. Ebenfalls sind diese Vorgänger bereits obsolet und bringen eine Reihe von Sicherheitslücken mit sich.

Entwicklung mit REST

Vorteile:

- Es ist möglich mit verschiedenen Ressourcen zu arbeiten, welches für unser Projekt notwendig ist. Bereits in der Domänenrecherche wurden verschiedene reale Objekte identifiziert, welche mögliche Ressourcen darstellen könnten.
- Es besteht bereits Erfahrung mit der Entwicklung von REST-Schnittstellen. (Modul WBA2)
- Es ist möglich zwischen verschiedenen Repräsentationen von Daten für die Netzwerkdarstellung zu wählen. Grundsätzlich steht so eine größere Offenheit für die Wahl eines geeigneten Formats für die Netzwerkdarstellung.
- Repräsentationen von Ressourcen können auf weitere verweisen. Dies ist für unser Projekt interessant, da bereits verschiedene Objekte der realen Welt identifiziert wurden, welche semantisch zusammenhängen.
- Eine Statuslose Kommunikation ermöglicht eine lose Kopplung zwischen Client und Server. Somit ist es möglich eine große Anzahl an Clients zu bedienen, welches das Ziel der Abfallreduktion unterstützt.
- Anschließend an den letzten Punkt steht die gute Skalierbarkeit.

Nachteile:

- Der Entwickler muss sich ggf. selbständig, um das Marshalling und Unmarshalling kümmern, wodurch ein erhöhter Aufwand entsteht.
- Entwicklung benötigt in der Regel eine iterative Vorgehensweise, welche sich aufwendig gestalten kann.

Entwicklung mit SOAP

Vorteile:

- Eignet sich gut, um Transaktionen (besonders Zahlungsvorgänge, Telekommunikation oder Ähnliches) in einem System zu entwickeln.
- Es sind keine Marshalling bzw. Unmarshalling Implementierungen notwendig.
- Ermöglicht die Kommunikation von Objekten.

Nachteile:

- Für die Repräsentationen von Daten wird lediglich XML verwendet. XML ist eine solide und umfangreiche Metasprache, welche eine große Relevanz besitzt, jedoch kann es sein, dass andere Datenformate für die Repräsentationen sich ggf. besser eignen.
- SOAP wird i.d.R nicht für Webanwendungen und die Entwicklung mobiler Applikationen genutzt.
- Es wird eine große Bandbreite für die Kommunikation von Daten benötigt.
- Es besteht keine Vorerfahrung mit diesem Protokoll bzw. dieser Vorgehensweise für die Erstellung einer Schnittstelle zwischen Systemen.

Fazit: Es wurde sich nach Gegenüberstellung beider Alternativen für eine REST-Schnittstelle geeinigt, da diese für Webanwendungen ein gängiger Standard ist und bereits in der Domänenrecherche erkannt wurde, dass es nötig ist mit verschiedenen Ressourcen zu arbeiten. Ebenfalls ist die mangelnde Erfahrung, der Projektkontext und die Restriktion auf ein vorgegebenes Datenformat, welche eindeutig gegen die Verwendung von SOAP sprechen.

Abwägung verschiedener Librarys

An dieser Stelle wird über verschiedene Möglichkeiten für die Versendung der Daten von der Seite des Clients an den Server abgewogen. Grundsätzlich steht die Möglichkeit dies mit der in Java integrierten Methode der HTTP(S)URLConnection zu erledigen, oder eine Library wie: „Volley“ oder „Okhttp“. Im Folgenden werden die Alternativen näher betrachtet, um eine Entscheidung treffen zu können.

URLConnection ist eine Standardfunktion, welche von Java zum für HTTP/HTTPS anfragen verwendet werden kann. Der Vorteil der Nutzung dieser ist, dass nicht unnötig zusätzliche Abhängigkeiten zu Librarys geschaffen werden. Die Verwendung dieser Vorgehensweise würde jedoch umfangreicher von Implementierungsaufwand sein.

Okhttp und Volley sind Module bzw. Librarys, welche den Informationsaustausch effizient und einfach gestalten sollen. Der Vorteil ist an dieser Stelle, dass der Implementierungsumfang gesenkt werden würde, jedoch würden an dieser Stelle zusätzliche Abhängigkeiten entstehen.

Fazit: Da die URLConnection ein fester Teil von Java ist und somit nicht mit häufigen Änderungen zu rechnen ist, wird für das Projekt diese Vorgehensweise gewählt. Volley und Okhttp bieten zwar eine angenehmere Implementierung, jedoch würden zusätzliche Abhängigkeiten entstehen, welche vermieden werden können.

Die Abwägung zwischen den verschiedenen Datenformaten wurde ebenfalls verbessert und durch klare Vor- und Nachteile Tabellen ergänzt. Diese sind im Anhang zu finden. Im Anhang ist ebenfalls eine Iteration der PoCs zu finden, welche im Grunde gleichgeblieben sind, jedoch die Formulierungen in Hinsicht auf die Messbarkeit stattgefunden hat.

Iteration der Architektur

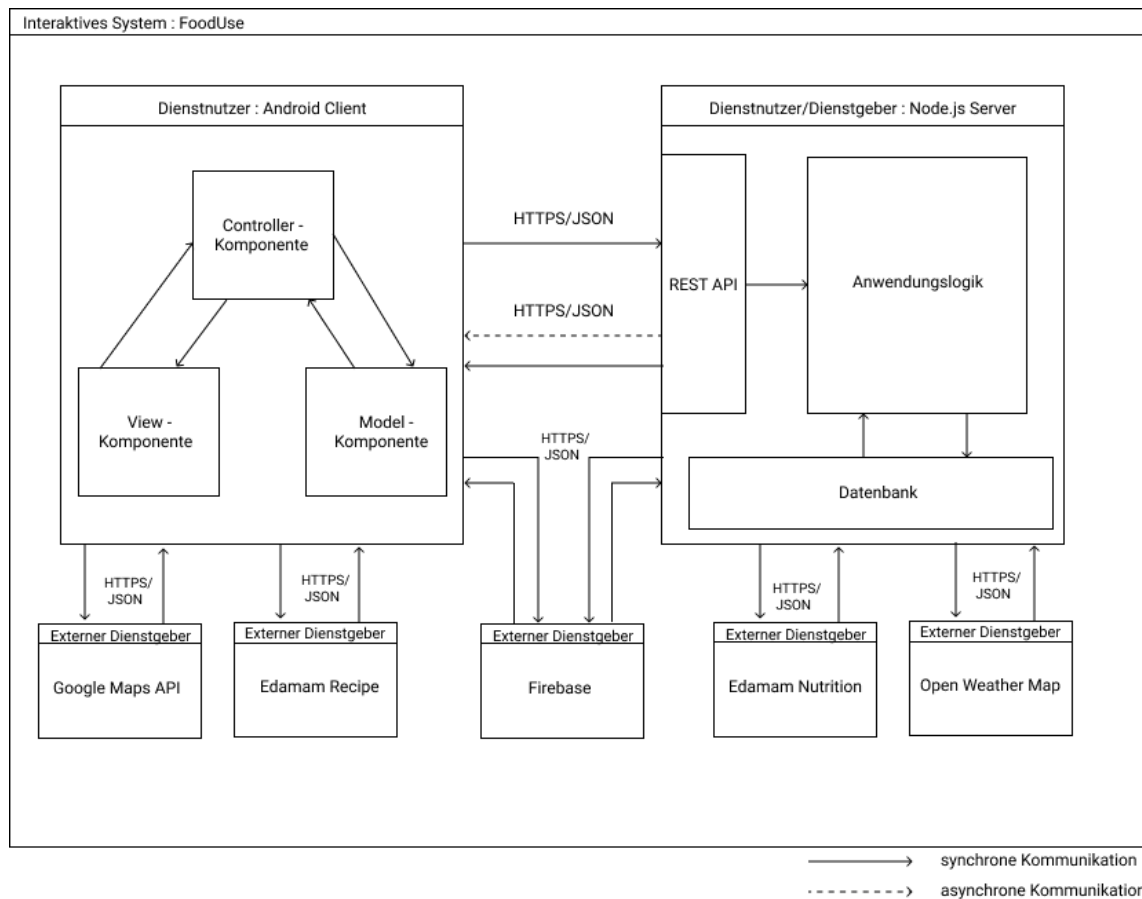


Abbildung 34. Iteration des Architekturmodells

Das Grundlegende Architekturmodell hat sich nicht ausschlaggebend verändert. Die größte Veränderung ist auf der Seite des Clients zu sehen, welcher eine andere Architektur als zuvor besitzt. Es wurde sich für das Model-View-Controller Pattern entschieden, da dies im Vergleich zu der Schichten Architektur eine einfachere Methode zur Synchronisierung zwischen den visuellen Komponenten und den Daten bietet. Zudem wird diese Architektur bei der Entwicklung mit Android Studio automatisiert vorgegeben, wodurch dies ebenfalls Entwicklungsaufwand einspart. Die im zuvor erstellen Modell angesiedelte Anwendungslogik ist nun Teil des Models in Form von Klassen. Die Datenbank, welche Daten für die Anwendungslogik bereitstellt ist ebenfalls Teil der Model Komponente. Für diese Zwecke bleibt weiterhin SQL-Lite bestehen.

Datenschutz und Sicherheit der Dienste

Im Folgenden werden die verschiedenen genutzten Dienste auf ihre Sicherheit und auf ihre Vertretbarkeit untersucht. Besonders ist hier der Dienst Firebase von relevant, da an dieser Stelle Personenbezogene Daten übermittelt werden.

Firebase Datenschutz und Sicherheit

Grundsätzlich haben Entwickler, welche Firebase nutzen und die Vorgaben der DSGVO umsetzen müssen, die Möglichkeit verschiedene Einstellungen zu den erhobenen Daten zu treffen. Es kann kontrolliert werden wie die Daten erhoben werden, die Aufbewahrungsdauer festgelegt werden oder erhobene Analytics Daten gelöscht werden. Firebase ist ein Dienst, welcher in Hinblick auf die

Sicherheit eine Reihe von Zertifizierungen und umgesetzten Standards aufweist. Zu den Standards gehören die ISO 27001, SOC1, SOC2, SOC3, ISO27001 und ISO 27018. Weiterhin werden die Daten mit Hilfe der HTTPS Protokolls übertragen. Ebenfalls unterliegt der Dienst dem EU-US Privacy Shield.

Google Maps Datenschutz und Sicherheit

Grundsätzlich ist Google Maps ebenfalls ein Dienst, welcher sensible Daten über Benutzer kommuniziert und auswertet. Diese Auswertung ist dahingehend vertretbar, da die erhobenen Daten ebenfalls kontrolliert werden können und lediglich zur Verbesserung des Dienstes beitragen. Daten werden an dieser Stelle ebenfalls verschlüsselt übertragen und der grundsätzlich ist der Zugriff auf diese Daten beschränkt auf Mitarbeiter und Ähnlichem. Kontakt mit diesen Daten unterliegt strengen Vertraulichkeitsverpflichtungen. Dieser Dienst unterliegt ebenfalls dem EU-US Privacy Shield.

Edamam Datenschutz und Sicherheit

Edamam selbst erhebt Personenbezogene oder Gerätbezogene Daten, wobei jedoch finanzielle Informationen davon ausgeschlossen sind. Die Daten, welche erhoben werden, werden verwendet, um die angeforderten Dienste bereit zu stellen. Der Dienst hat sich für das EU-US Privacy Shield zertifiziert und befolgt diese.

OpenWeatherMap Datenschutz und Sicherheit

OpenWeatherMap bezieht lediglich Daten, wenn ein Konto auf der Seite erstellt wird und diese besucht wird. Bei den Diensten stehen Wetterinformationen und Standorte im Vordergrund und keine Personenbezogene Daten, welche gefährdet werden können.

Fazit Grundsätzlich stellen die gewählten Dienste eine vertretbare Datenschutzerklärung und Sicherheit in Bezug auf das Projekt und die personenbezogenen Daten dar. Besonders bei der Übertragung durch Dienste von Google (Firebase, Google Maps) liegen hohe Sicherheitsstandards vor und eine ausführliche Datenschutzerklärung, welche die Kontrolle der erhobenen Daten ermöglicht. Ebenfalls werden die erhobenen Daten i. d. R. zur Bereitstellung und Optimierung des Angebotenen Dienstes verwendet.

Domänenmodelle und Iterationen

Im Verlauf des Projekts und besonders in der Domänenrecherche wurden viele Informationen gesammelt und in einem Domänenmodell gesammelt, um ein gemeinsames Lexikon bzw. Vokabular für die Zusammenarbeit zu erstellen. Insgesamt hat das Domänenmodell drei Iterationen durchlaufen.

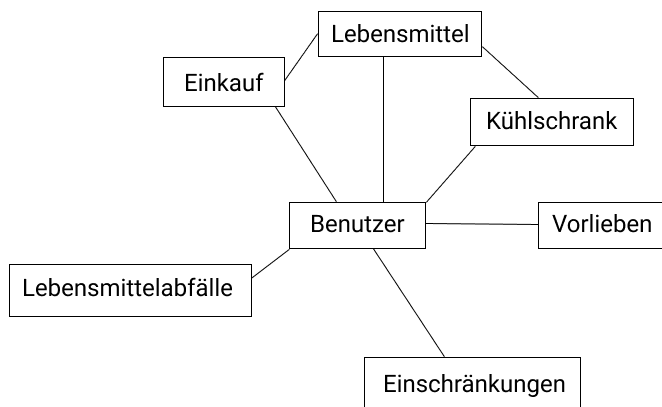


Abbildung 35 Domänenmodell erster Ansatz

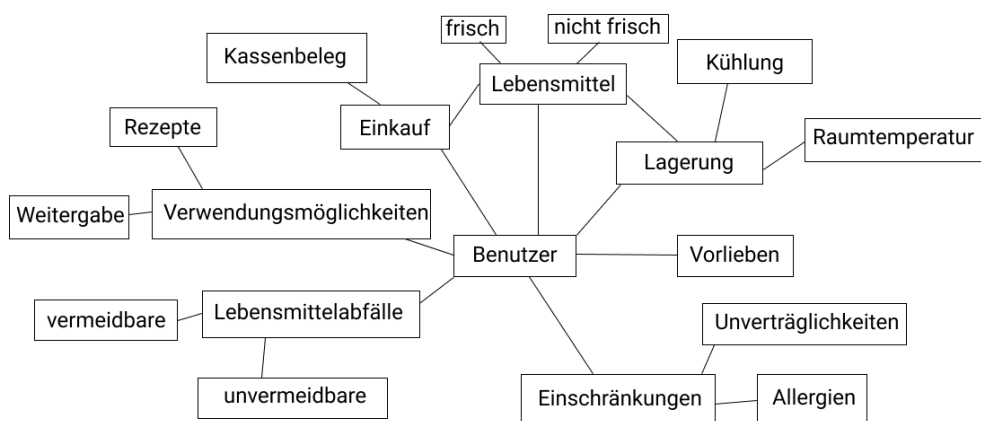


Abbildung 36 Erste Iteration Domänenmodell

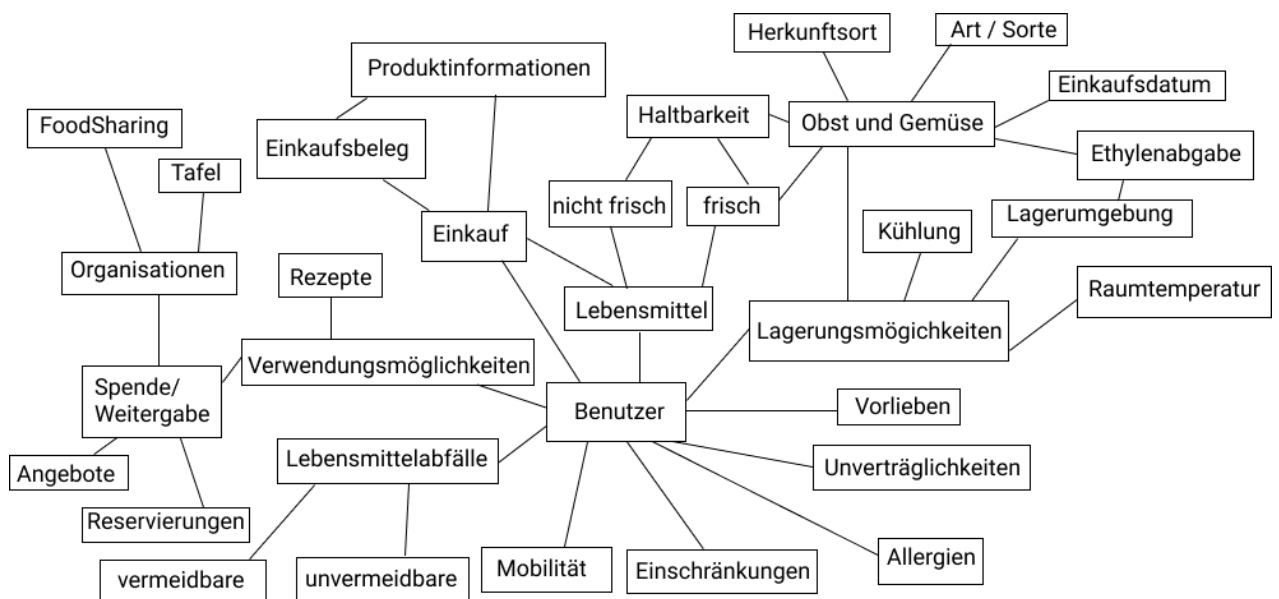


Abbildung 37 Letzte Iteration

Iteration Kommunikationsmodelle

Iteration Deskriptives Kommunikationsmodell

Bei dem deskriptiven Kommunikationsmodell wurden im ersten Ansatz teilweise Antworten auf Anfragen vernachlässigt beziehungsweise vergessen. Diese wurde ergänzt, um ein vollständiges Modell zu erhalten, welches den aktuellen Zustand bestmöglich beschreibt. Um eine möglichst gute Abbildung für die zukünftige Entwicklung abzuleiten wurden weiterhin die Informationen eingefügt, welche ausgetauscht werden. Somit ist ebenfalls eine präzisere Differenzierung zwischen dem Deskriptiven und Präskriptiven Modell möglich.

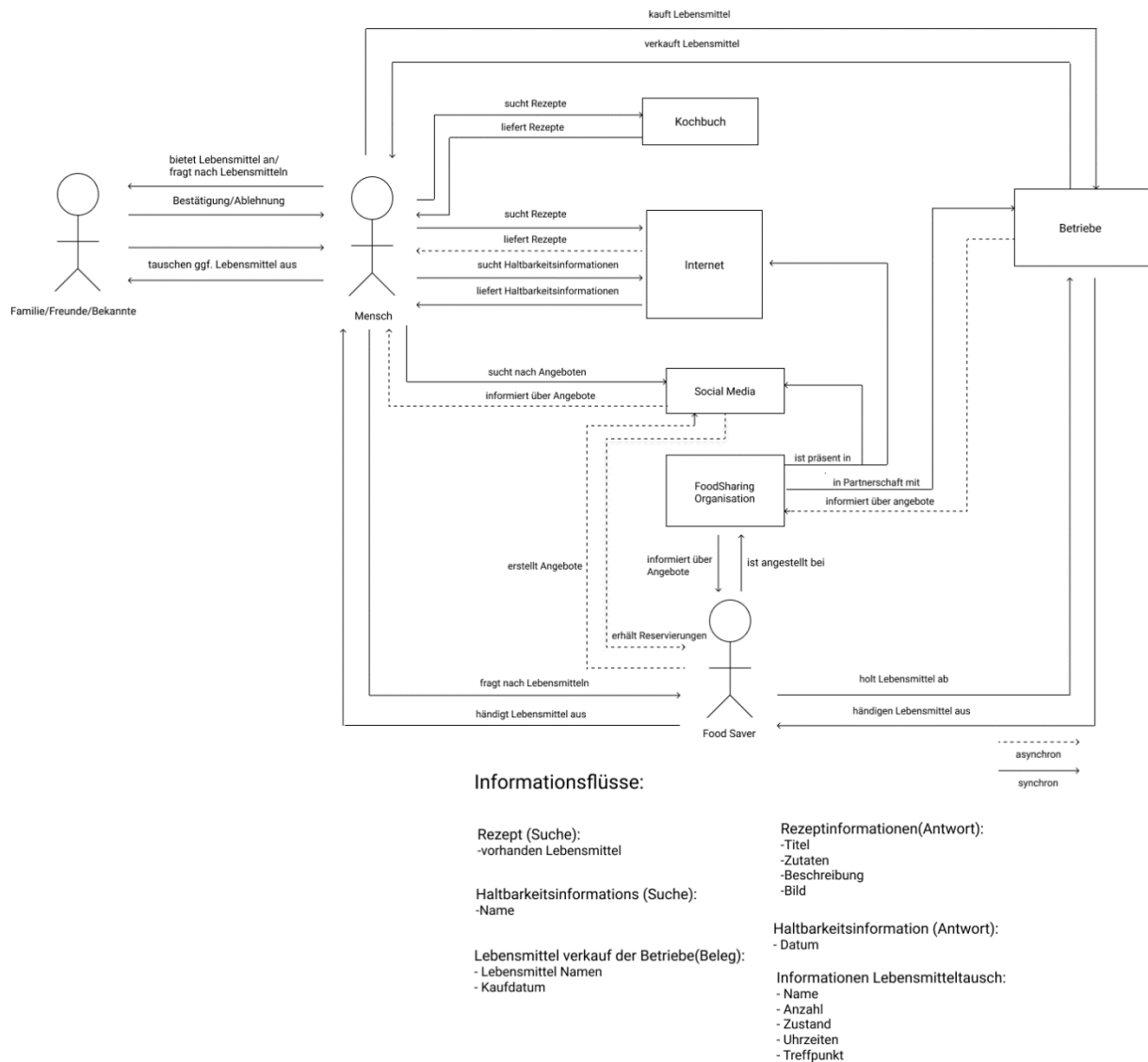


Abbildung 38 Iteration Deskriptives Kommunikationsmodell

Es ist nun zu erkennen, dass bereits eine Möglichkeit besteht Lebensmittel zwischen Privatpersonen auszutauschen. Dies beschränkt sich jedoch auf die Freunde, Familie und die Bekannten. Der Ablauf wurde etwas detaillierter in das Modell mit aufgenommen.

Eine Nummerierung des Kommunikationsablaufs ist schwer in das Modell miteinzubringen, da es grundsätzlich zu viele Möglichkeiten gibt. Im Folgenden werden deshalb die wichtigsten Kommunikationsabläufe textuell beschrieben:

Verwertung von Lebensmitteln

Grundsätzlich beginnt diese Kommunikation mit dem Erwerb der Lebensmittel, welche in der Zukunft angeboten werden sollen. In den meisten Fällen werden diese über Betriebe (Supermärkte) eingekauft. Die Informationen, welche an dieser Stelle ausgetauscht werden, wurden unter dem Punkt: „Lebensmittelverkauf der Betriebe (Beleg)“ genauer spezifiziert. Möchte der Käufer in diesem Fall die Lebensmittel verwerten, so verbraucht er Sie direkt, sucht nach Rezepten zur Verwertung (In Kochbüchern oder im Internet) oder kann Sie an die Familie/Freunde oder Bekannte nach kurzen Informationsaustausch (Beschrieben unter dem Punkt: „Informationen Lebensmittelaustausch“) abgeben. Dies ist momentan die einzig realistische Möglichkeit diese Lebensmittel zu verwerten. Diese Kommunikation findet i.d.R durch Telefon, Chat oder ein direktes Gespräch statt.

Suche von Lebensmitteln

Grundsätzlich beginnt diese Kommunikation damit, dass zunächst Freunde/Familie oder Bekannte gefragt werden. Dies muss jedoch nicht immer der Fall sein, da dies in einigen Fällen zu Schwierigkeiten (aus Schamgefühl oder ähnlichem) führen. Die nächste Möglichkeit besteht darin, nach Angeboten in sozialen Netzwerken zu suchen. Diese Angebote werden durch sogenannte FoodSaver, welche FoodSharing Organisationen angehörten, erstellt. So können die Suchenden nun mit dem FoodSavern Kontakt aufnehmen, Lebensmittel reservieren und diese im Anschluss abholen. Hier wird entweder direkt über den Chat der Social Media Plattform gearbeitet oder es werden Telefonnummern hinterlassen und Einzelheiten werden über ein kurzes Telefongespräch ausgetauscht. Die Informationen die hierbei ausgetauscht werden wurden unter dem Punkt: „Informationen Lebensmittelaustausch“ festgehalten.

Iteration Präskriptives Kommunikationsmodell

Es wurden an dieser Stelle ebenfalls vergessene Antworten ergänzt und neue Kommunikationswege eingezeichnet, welche sich im Rahmen des Projektverlaufs ergeben haben. Hierzu gehört der Nachrichtenaustausch zwischen Anbieter und Abholer. Die ausgetauschten Informationen werden im Bereich Datenstrukturen noch genauer erläutert.

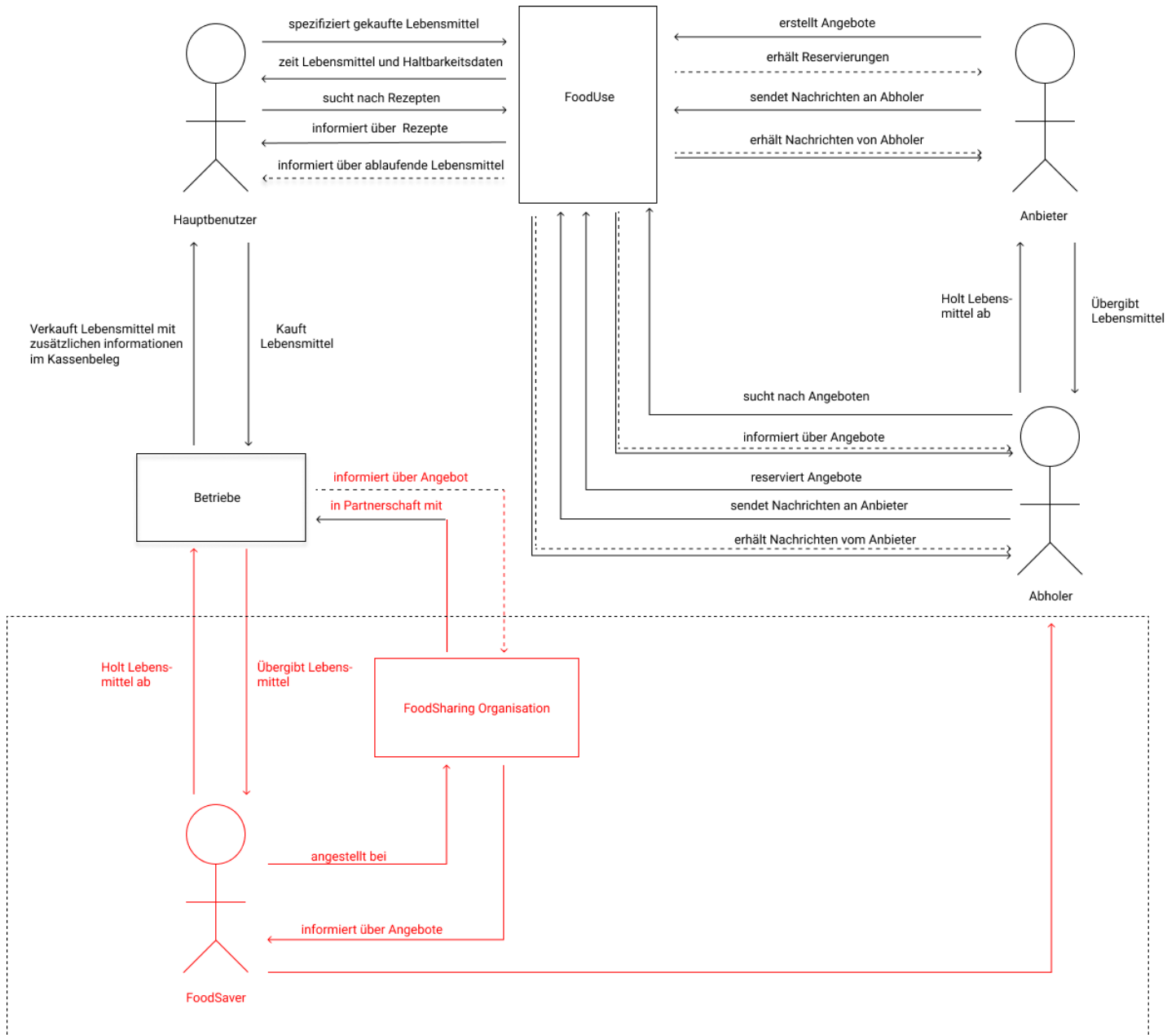


Abbildung 39 Iteration Präskriptives Kommunikationsmodell

Begründung der gewählten Verteilung der Anwendungslogik

Grundsätzlich lassen sich zwei größere Bereiche innerhalb unseres Domänenmodells identifizieren.

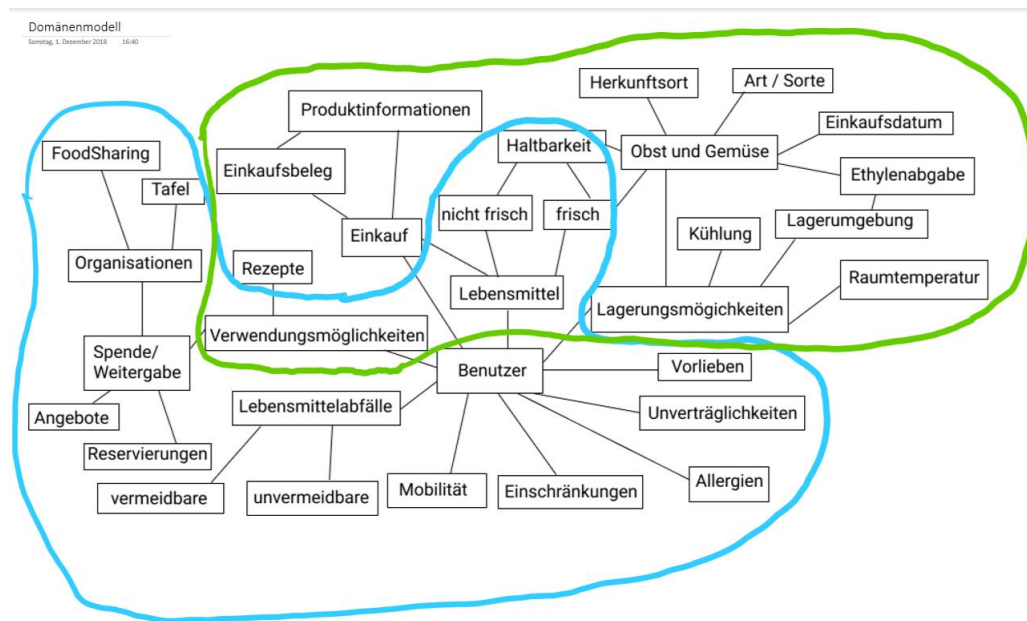


Abbildung 40. Domänenmodell Aufteilung

Der eine Teil (In Abb. 40 blau markiert) beschäftigt sich im Kern mit dem Austausch von Lebensmitteln, der Ermittlung geeigneter Angebote und der Kommunikation zwischen Anbieter und Abholer. Der andere Teil (In Abb. 40 grün markiert) beschäftigt sich größtenteils mit der Verwaltung der eigenen Lebensmittel, Berechnung von Haltbarkeitsdaten und finden von geeigneten Verwendungsmöglichkeiten, welche als Teilaufgaben des zuvor genannten Teils angesehen können. Die Einteilung ist nicht zu 100% akkurat, da es sich trotz verteilter Anwendungslogik dennoch um ein System handelt, und die Grenzen somit fließend sind.

Bei der Verteilung der Architektur wurde die Berechnung der geeigneten Lebensmittelangebote und die Verwaltung auf der Seite des Servers angeordnet, da so die Ressourcen zentralisiert vorliegen und die Benutzer so unabhängig voneinander darauf zugreifen können. Die eigenen Lebensmittel der Benutzer hätten an dieser Stelle ebenfalls auf die Seite des Servers angelegt werden können. Dies hätte den Vorteil, dass eigene Lebensmitteleinträge, von jedem Gerät aus verwaltet werden könnten und eine Ermittlung der Vorlieben anhand der eigenen Lebensmittel einfacher wäre. Dem steht jedoch der ausschlaggebende Nachteil gegenüber, dass die Lebensmittel und die Benachrichtigung bei ablaufenden Lebensmitteln von der Internetverbindung des Benutzers abhängen würden. Da Lebensmittel auch ablaufen, wenn kein Internet vorhanden ist, ist es sinnvoll diese clientseitig zu verwalten. Ebenfalls wird hierdurch der Server durch die stark verminderte Anzahl an Anfragen entlastet. Die Funktionalität der Rezeptsuche wurde ebenfalls auf die Clientseite gelegt, da hier die eigenen Lebensmittel der Benutzer liegen, welche verwendet werden sollen.

Die Berechnungen von Distanzen übernimmt sowohl der Client als auch der Server individuell. Somit wird die Auslastung des Servers ebenfalls reduziert. Ebenfalls sind diese Berechnungen auf Client und

Server unterschiedlich, da mit verschiedenen Formeln für die Berechnung gearbeitet wird, da die Größenordnung der Distanzen sich stark voneinander unterscheiden. Alternativ hätte der Server diese Berechnung ebenfalls übernehmen können, jedoch unterstützt bzw. beeinflusst diese Teilaufgabe nicht die Hauptaufgabe des Servers direkt.

Durch diese gewählte Verteilung liegt eine bestmögliche Verteilung der Rechenintensiven Aufgaben vor und die Komponenten sind sowohl unabhängig als auch in Zusammenarbeit miteinander sinnvoll nutzbar. Ein Ausfall des Servers (wenn auch unwahrscheinlich) würde nicht die Funktionalität der Lebensmittelverwaltung beeinflussen und einzelne Ausfälle der Clients würden die Verwaltung der Angebote und Reservierungen und die Berechnung der geeigneten Angebote nicht beeinflussen.

Beschreibung der Datenstrukturen

Im Folgenden werden die Datenstrukturen der Ressourcen, welche zwischen Client und Server ausgetauscht werden sollen, spezifiziert. Im Anschluss wird eine REST-Tabelle angelegt, welche diese Ressourcen mit ihren möglichen Verben, ihrer Bedeutung und Datentypen für die Anfragen bzw. Antworten und möglichen Statuscodes angegeben. Die Einzelnen Ressourcen und Attribute lassen sich in den bereits aufgeführten Modellen (Domänenmodelle, Kommunikationsmodelle) wiederfinden.

Primärressource: User

"Id" : "Einzigartige EintragsID"
"uid" : "Einzigartige Benutzerid"
"Token" : "Firebase Token eines Benutzers"
"Vorname" : "Vorname des Benutzers"
"Nachname" : "Nachname des Benutzers"
"Straße" : "Straße des Benutzers"
"Hausnummer" : "Hausnummer des Benutzers"
"Stadt" : "Stadt des Benutzers"
"Latitude" : "Breitengrad des Standorts"
"Longitude" : "Längengrad des Standorts"
"Bild" : "Bild des Benutzers"
"Email" : "E-Mail Adresse des Benutzers"
"Einschränkungen" : "Liste von Einschränkungen des Benutzers"
"Einträge" : "Liste von Lebensmitteleinträgen"

Abbildung 41. Primärressource User

Die User Ressource ist eine Primärressource, welche zunächst einmal alle Eigenschaften eines Benutzers enthält. Viele dieser Eigenschaften sind jedoch von der Datenstruktur her umfangreicher und es ist zu erwarten, dass oftmals Änderungen vorkommen. Aus diesem Grund ist es sinnvoll diese in Subressourcen zu modellieren. Hierdurch muss bei Anfragen nicht immer die gesamte Primärressource angegeben werden, sondern lediglich ein kleinerer Abschnitt. Zu diesen Ressourcen gehören: „Einträge“, „Einschränkungen“, „Nachrichten“ und „Token“. Auf diese Subressourcen wird im Folgenden weiter eingegangen. Die Id ist die einzigartige ID für die Datenbank, während die uid,

die ID ist, welche von Firebase vergeben wird, um den Benutzer, während der Authentifizierung eindeutig zu identifizieren. Die Eigenschaften Vorname, Nachname, Straße, Hausnummer, Stadt, Bild und Email sind selbsterklärend. Die Eigenschaften Latitude und Longitude bilden den Standort ab, an dem sich der Benutzer befindet.

Subresource: User/Einträge

```
"Id" : "Einzigartige EintragsID"  
  
"Name" : "Name des Lebensmittels"  
  
"Anzahl" : "Anzahl der Einkäufe"
```

Abbildung 42. Einträge

In dieser Subresource werden die Lebensmittel, welche der Benutzer Abholt oder selbst Kauft festgehalten. Diese Information soll im weiteren Verlauf dabei helfen, geeignete Angebote für den Benutzer zu finden. Diese Liste soll sich in regelmäßigen Abständen automatisch leeren, damit immer die aktuellsten Interessen des Benutzers berücksichtigt werden. Bei neu eingefügten Einträgen oder beim Löschen soll sich, falls bereits ein Eintrag vorhanden ist, die Anzahl erhöhen bzw. verkleinern.

Subresource: User/Einschränkungen

```
"id" : "Einzigartige EintragsID"  
  
"name" : "Name der Einschränkung"
```

Abbildung 43. Einschränkungen

In dieser Subresource werden die Esseneinschränkungen des Benutzers festgehalten, welche bei der Auswahl von geeigneten Angeboten relevant sind. Bei diesen Einschränkungen kann der Benutzer zwischen: Vegan, Vegetarisch, Laktoseintoleranz Molke-frei, Glutenunverträglichkeit, Ei-frei, Nussallergie, Schweinefleischfrei, Nierenfreundlich und Alkoholfrei und weiteren wählen. Die Eigenschaften der angebotenen Lebensmittel werden anhand des externen Dienstgebers Edamam Nutrition Analysis durchgeführt.

Subresource: User/Token

```
"Id" : "Einzigartige EintragsID"  
  
"Token" : "Firebase Token"  
  
"uid" : "Benutzerid für die Zuordnung"
```

Abbildung 44. Token

Mit Hilfe dieses Tokens wird die asynchrone Kommunikation mit Firebase realisiert. Dieser Token identifiziert das mobile Endgerät des Benutzers und wird sich ggf. falls verändern, falls der Benutzer sich in einem anderen Gerät einloggt. Bei jedem Login des Benutzers muss dieser auf dem Client abgefragt werden und auf dem Server hinterlegt sein.

Subressource: User/Nachrichten

"Id" : "Einzigartige EintragsID"
"Sender" : "Einzigartige BenutzerID des Senders"
"Empfänger" : "Einzigartige BenutzerID des Empfängers"
"Inhalt" : "Inhalt der Nachricht"
"Type" : "Typ der Nachricht"
"Zeit" : "Zeitpunkt an dem die Nachricht versendet wurde"

Abbildung 45. Nachrichten

Diese Subressource ist für den Nachrichtenaustausch zwischen Abholer und Anbieter verantwortlich. Die id ist für die Speicherung der Nachricht innerhalb der Datenbank notwendig. Die Attribute Sender und Empfänger sind selbstbeschreibend, an dieser Stelle ist die einzigartige BenutzerID nötig, um entsprechend Token für die Kommunikation zu ermitteln, welche zuvor beschrieben wurde.

Primärressource: Angebote

"Id" : "Einzigarte EintragsID"
"Uid" : "BenutzerID des Erstellers"
"Status" : "Status des Angebots"
"Zeit" : "Erstellungszeitpunkt des Angebots"
"Titel" : "Titel des Angebots"
"Abholzeitpunkt" : "Präferierter Abholzeitpunkt des Anbieters"
"Lebensmittel" : "Lebensmittel, welches Angeboten wird"
"Lebensmittelliste" : "Lebensmittelliste, welche das Angebot enthält"
"Ablaufdatum" : "Ablaufdatum des Lebensmittels"
"Beschreibung" : "Zusätzliche Beschreibung des Angebots"
"Bild" : "Bild des angebotenen Lebensmittels"
"Latitude" : "Breitengrad des erstellen Angebots"
"Longitude" : "Längengrad des erstellen Angebots"
"Einschränkungen" : "Ermittelten Einschränkungen für das Angebot"
"Entfernung" : "Ermittelte Entfernung des Angebots zum potentiellen Käufer"
"Abholwert" : "Abholwert, welcher Beschreibt, wie gut sich das Angebot für den Käufer eignet"

Abbildung 46. Angebot

Die Primärressource bildet die angebotenen Lebensmittel der Benutzer ab. Ein Angebot hat wie auch die anderen Ressourcen eine ID und selbstbeschreibende Attribute wie: „Titel“, „Lebensmittel“, „Zeit“ und „Beschreibung“. Jedes Angebot muss ebenfalls die BenutzerID des Erstellers enthalten. Die Attribute Latitude, Longitude werden automatisch von dem jeweiligen Benutzer übernommen und wie Einschränkungen werden wie zuvor beschrieben mit Hilfe eines externen Dienstes automatisch

übernommen. Die Entfernung und der Abholwert sind Attribute, welche erst dann vorliegen können, wenn ein Angebot von einem konkreten Benutzer (potentieller Abholer) angefragt wird. Diese werden benutzerbezogen serverseitig berechnet. Eine detaillierte Beschreibung der Ermittlung dieser Werte findet im Abschnitt: „Beschreibung der Anwendungslogik“ statt.

Primärressource: Reservierungen

```
"Id" : "Einzigartige EintragsID"  
"Aid" : "Einzigartige AngebotsID"  
"Uid" : "Einzigartige BenutzerID des Abholers"  
"Eid" : "Einzigartige BenutzerID des Anbieters"  
"Zeit" : "Erstellungszeitpunkt der Reservierung"  
"Straße" : "Straße des Benutzers"  
"Hausnummer" : "Hausnummer des Anbieters"  
"Stadt" : "Stadt des Anbieters"
```

Abbildung 47. Reservierungen

Die User Ressource ist eine Primärressource, welche Reservierungen eines Benutzers innerhalb des Systems abbildet. Hat ein Benutzer ein Angebot für interessant empfunden und möchte dieses entgegennehmen, so kann er eine Reservierung für dieses erstellen und sich mit dem Käufer in Kontakt setzen. Wenn ein Benutzer ein Angebot reserviert, so muss darauf ein Statusänderung des Angebots erfolgen, damit andere Benutzer dieses Angebot nicht ebenfalls reservieren können.

Die Ressourcen wurden innerhalb einer REST-Tabelle genauer spezifiziert und sind im [Anhang](#) zu finden. Im [Anhang](#) wurde ebenfalls eine Tabelle mit den möglichen Statuscodes und einer Beschreibung der Fehlerbehandlung beigelegt.

Hypermedia-Konzept

An dieser Stelle haben wir Ressourcen untereinander verlinkt, welche bei weiteren Anfragen interessant bzw. nützlich sein könnten. Nach der Anfrage der einzelnen Subressourcen des Benutzers kann es von Interesse sein eine Verlinkung zur Primärressource zu erhalten. Dies macht in die andere Richtung keinen Sinn, da dort bereits alle Informationen spezifiziert sind. Angebote sowie Reservierungen hängen im Kontext semantisch zusammen, daher macht es ebenfalls Sinn diese untereinander zu verlinken. Hat ein Benutzer seine Reservierungen abgefragt, so ist es wahrscheinlich, dass im weiteren Verlauf sich das dazugehörige Angebote ebenfalls angeschaut werden will. Es wurde abgewogen grundsätzlich ebenfalls die Angebote und Reservierungen als Subressource des Benutzers zu modellieren. Dies würde Sinn machen, da diese ebenfalls Teil des Benutzers sind, da Sie von ihm erstellt worden sind. Da jedoch auch andere Benutzer auf Angebote zugreifen und Reservierungen ebenfalls zu mehr als einer Person gehören, haben wir uns dazu entschieden diese als Primärressource zu modellieren. Der Zugriff auf die eigenen Angebote und Reservierungen erfolgt somit per Query Parameter, statt wie zuvor als Parameter. Dies macht ebenfalls mehr Sinn, da das suchen bzw. Filtern einer Teilmenge Aufgabe eines Querys ist. Da die Angebote immer von einem Benutzer erstellt werden, so ist es sinnvoll auch hier eine Verlinkung zu der Ressource des Benutzers mitzuliefern. Dies ist ebenfalls der Fall bei Reservierungen, an dieser Stelle werden Verlinkungen zu dem Anbieter bzw. Abholer mitgeliefert. Die Verlinkung der einzelnen Ressourcen wurde im Folgenden grob visualisiert.

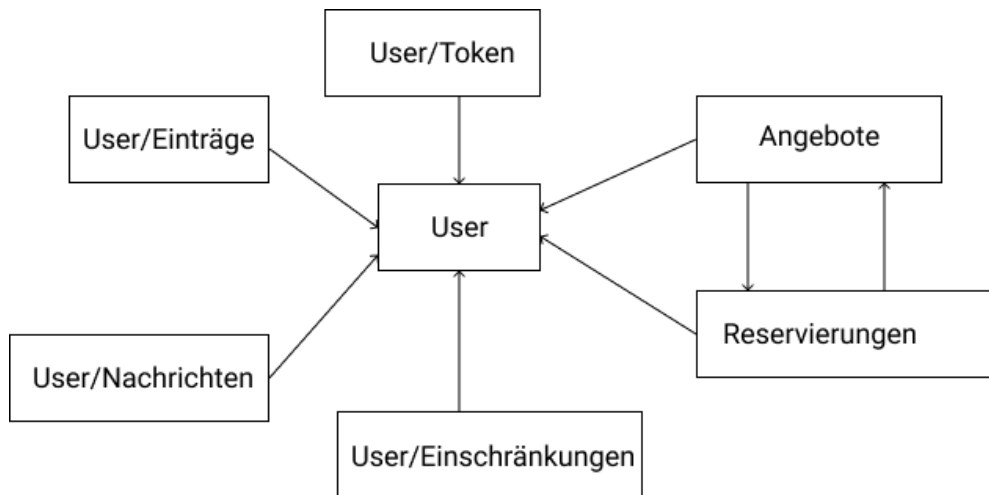


Abbildung 48. Hypermedia Verlinkung

Datenstruktur Client

Klasse FrischesLebensmittel

```

private String name;
private String herkunftsland;
private String kaufdatum;
private int anzahl;

```

Abbildung. 49. FrischesLebensmittel

Diese Klasse beschreibt ein frisches Lebensmittel, nachdem es über den QR-Code eingelesen wurde.

Klasse AbgepackterLebensmittelEintrag

```

private String name;
private String ablaufdatum;
private int anzahl;

```

Abbildung 50. AbgepackterLebensmittelEintrag

Diese Klasse beschreibt einen normalen Lebensmitteleintrag von abgepackten Lebensmitteln mit bekannten Haltbarkeitsdatum.

Klasse FrischerLebensmittelEintrag

```
private String name;  
private String herkunftsland;  
private String kaufdatum;  
private Date ablaufdatum;  
boolean kühlung;  
private int anzahl;
```

Abbildung. 51. Frischer Lebensmitteleintrag

Diese Klasse beschreibt einen Lebensmitteleintrag für ein frisches Lebensmittel. Es wurde um die Eigenschaften Kühlung und Ablaufdatum erweitert.

Iteration Beschreibung der Anwendungslogik Client

Die Anwendungslogik der Clients beschäftigt sich im Kern mit der Berechnung der Haltbarkeitsdaten für frische Lebensmittel. Da dies oftmals Recherche und somit Aufwand für den Benutzer bedeutet, wie in der Contextual Task Analysis ermittelt und viele Informationen an dieser Stelle verloren gehen, ist es sinnvoll den Benutzer bei diesem Schritt zu unterstützen bzw. ihn zu automatisieren. Für eine Berechnung der Haltbarkeitsdaten sind zunächst folgende Informationen notwendig: Kaufdatum, Sort/Art, Herkunftsland. Diese Informationen werden über die „Dummy Schnittstelle“ des Supermarktes mitgeliefert.

Diese werden in der Form:

"Kaufdatum/Sorte, Herkunftsland, Menge/Sorte, Herkunftsland, Menge/ ..."

innerhalb eines QR-Codes auf dem Kaufbeleg des Kunden mitgeliefert. Diese müssen bevor die Berechnung beginnen kann in einzelne Lebensmittelobjekte geparkt werden. Ein wichtiger Faktor ist ebenfalls, welche Lebensmittel in der Nähe gelagert werden, da diese Ethylen ausstoßen, wodurch ebenfalls die Haltbarkeit beeinflusst wird.

Anhand dieser Daten werden die Daten mit der optimalen Lagerung (Kühlung oder Raumtemperatur) berechnet und eingetragen. Somit ist die optimale Lagerung ebenfalls Teil der Anwendungslogik. An dieser Stelle gehen wir nach dem Designprinzip „Decide for me but let me have the final say“ vor und ordnen die Lebensmittel von der Lagerung her so an, dass die Lebensmittel so lange wie möglich haltbar sind. Hierzu müssen die Daten zunächst mit zusätzlichen Daten aus einer Datenbank angereichert werden. Dazu gehören die Grundhaltbarkeit, Ethylen Abgabe und Kühlungszuschlag.

Im Allgemeinen besteht die Formel aus:

*Haltbarkeitsdatum = Kaufdatum + (Grundhaltbarkeit + Kühlungszuschlag – Transportzeit) * Ethylen*

Jedoch kann es sein, dass einzelne Faktoren wie der Kühlungszuschlag oder der Ethylen Faktor entfallen, je nach Bestand des Inventars. Um eine optimale Lagerung zu erreichen wird folgendermaßen vorgegangen: Zunächst müssen alle Lebensmittel nach ihrer Ethylen Abgabe sortiert werden. Falls nur ein Lebensmittel vorliegt, so kann es automatisch in den Kühlschrank. Da dies so keine anderen beeinflusst. Die Formel lautet:

Haltbarkeitsdatum = Kaufdatum + (Grundhaltbarkeit + Kühlungszuschlag – Transportzeit)

Sollte die Zahl der Lebensmittel größer als 1 sein, so muss überprüft werden, ob die Lebensmittel sich auf die zwei verschiedenen Listen aufgeteilt haben oder nicht. Falls Nein, so können ebenfalls alle in den Kühlschrank. Für die Liste wäre die gleiche Formel wie zuvor genannt die Richtige. Handelt es sich um die Liste der Lebensmittel mit Ethylen, so muss

$$\text{Haltbarkeitsdatum} = \text{Kaufdatum} + (\text{Grundhaltbarkeit} + \text{Kühlungszuschlag} - \text{Transportzeit}) * \text{Ethylen}$$

Verwendet werden, da beide Lebensmittel sich gegenseitig beeinflussen.

Ist dies nicht der Fall und die Lebensmittel haben sich auf zwei verschiedene Listen aufgeteilt, so müssen die Haltbarkeitsdaten für die Listen separat berechnet werden und zum Schluss zusammengefügt werden. Für die Liste der Lebensmittel ohne Ethylen läuft die wie bereits beschrieben ab. Für die Ethylen haltige Liste muss zusätzlich überprüft werden, ob diese größer ist als 1. Falls ja, so beeinflussen sich die Lebensmittel wieder gegenseitig und ein Ethylen Abzug muss stattfinden. Falls nicht, so fällt dieser weg. Da diese Lebensmittel somit ebenfalls nicht innerhalb des Kühlschranks sind, so fällt auch der Kühlungszuschlag weg: $\text{Haltbarkeitsdatum} = \text{Kaufdatum} + (\text{Grundhaltbarkeit} - \text{Transportzeit})$

Auf Basis der Informationen aus der Domänenrecherche wird für Ethylen haltige Lebensmittel in der Umgebung die Haltbarkeit um 30% verringert. Die Berechnung für die Transportzeit findet auf der Basis von Längen und Breitenangaben statt und wird mit der Formel:

$$D = \sqrt{dx^2 + dy^2};$$

$$dx = 111.3 * \cosinus(lat) * (lon1 - lon2)$$

$$dy = 111.3 * (lat1 - lat2)$$

$$lat = (lat1 + lat2) / 2 * 0.01745$$

wobei gilt: lat1, lat2, lon1, lon2 entsprechen Breiten und Längenangaben in Grad.

(vgl. Kompf, 2018)

Durch die innerhalb der in der Domänenreche ermittelten Informationen wird an diese Stelle für Produkte aus demselben Land ein Tag Transportzeit abgezogen. Sollte es sich um ein importiertes Produkt handeln, so werden für eine Distanz innerhalb 1500km zwei Tage und alles außerhalb drei Tage subtrahiert. Die gesamte Berechnung wird immer durchgeführt, wenn neue Lebensmittel hinzukommen, oder entfernt werden.

Falls der Benutzer die optimalen Lagerungsentscheidungen verändert, so wird nach den Angaben des Benutzers eine Liste mit gekühlten und ungekühlten Lebensmitteln erstellt und die Berechnung strikt nach diesen Angaben und durch die bereits skizzierten Regeln durchgeführt. Der Ablauf für die Berechnungen der optimalen Lagerung wurden innerhalb eines Ablaufdiagramms skizziert:

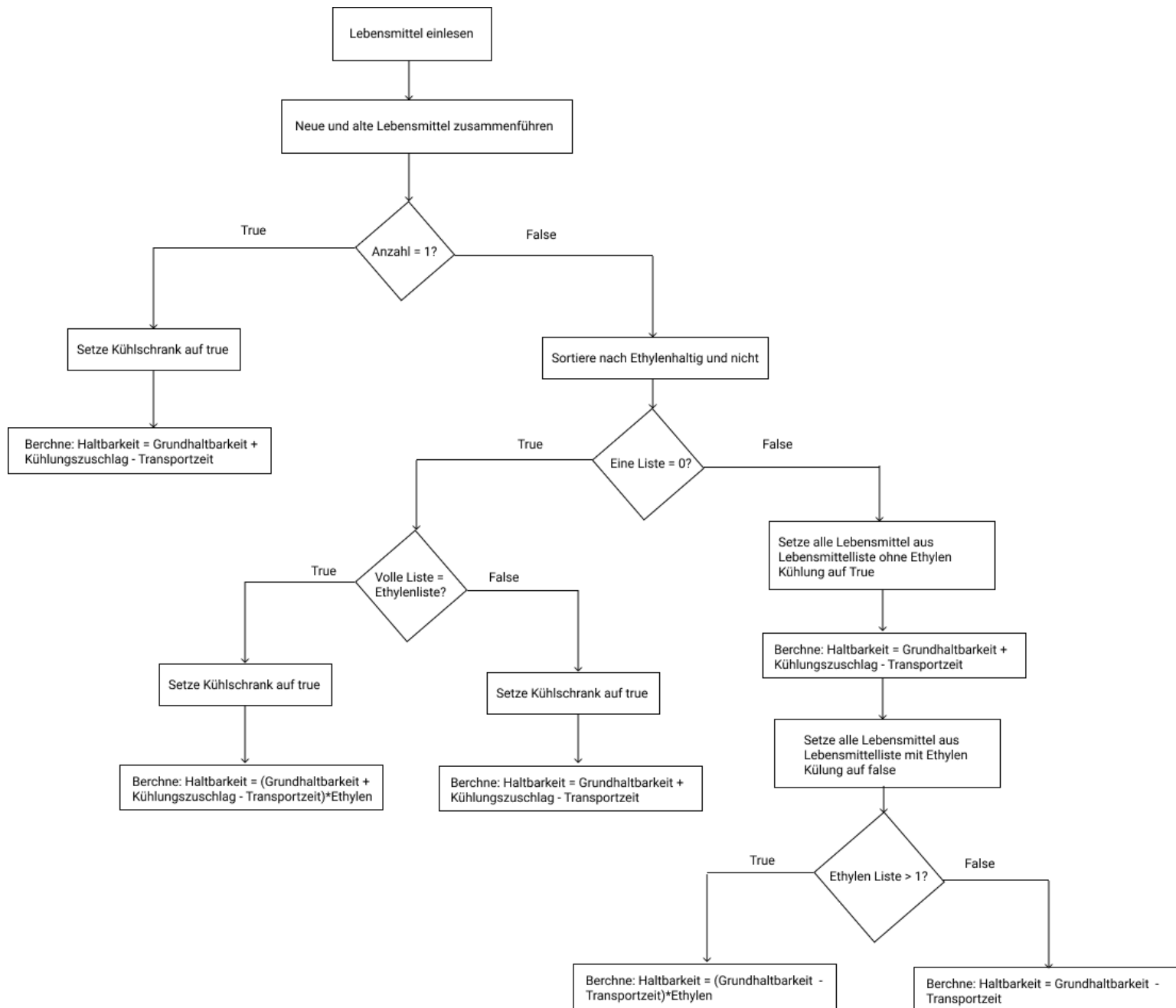


Abbildung 52. Client Anwendungslogik Ablaufdiagramm

Iteration Beschreibung der Anwendungslogik des Servers

Die Anwendungslogik des Servers hat sich durch die Erkenntnisse des MCI Abschnitts leicht verändert. Weitere Faktoren wie die Mobilität und eine andere Bewertung der Einschränkungen. Grundsätzlich ist jedoch die Idee der Berechnung eines Abholwerts die gleiche geblieben. Die folgende Tabelle stellt die verschiedenen Punkte für ein passendes Angebot dar.

Tabelle 42: Punkteverteilung Abholwert

Entfernung (falls zu Fuß)	Entfernung (falls zu Fuß)	Entfernung (falls kein Auto vorhanden)	Wetter	Vorlieben	Einschränkungen
Distanz innerhalb von 0,5 Km = 3 Punkte	Distanz innerhalb von 0,5-1 Km = 3 Punkte	Distanz innerhalb von 1,5 Km = 3 Punkte	Klarer Himmel = 1 Punkt	Mehr als 6 mal im gleichen Monat selbst erworben= 2 Punkte	Lebensmittel verstößt gegen keine Einschränkungen = 1
Distanz innerhalb von 0,5-1 Km = 2 Punkte	Distanz innerhalb von 1-1,5 Km = 2 Punkte	Distanz innerhalb von 1,5–2,5 Km = 2 Punkte	Bewölkt = 0 Punkte	1-6 mal im gleichen Monat selbst erworben = 1 Punkt	Lebensmittel verstößt gegen Einschränkungen = 0
Distanz innerhalb von 1 - 1,5 Km = 1 Punkt	Distanz innerhalb von 1,5-2 Km = 1 Punkte	Distanz innerhalb von 2,5 – 3,5 Km = 1 Punkt	Regen oder Sturm = - 1 Punkt	0 mal im gleichen Monat selbst erworben = 0 Punkte	-
Distanz größer als 1,5 Km = 0 Punkt	Distanz größer als 2 Km = 0 Punkte	Distanz größer als 3,5 Km = 0 Punkte			

Grundsätzlich setzt sich der Abholwert nach Folgender Berechnung zusammen:

$$\text{Abholwert} = (\text{Entfernungspunkte} + \text{Wetterpunkte} + \text{Vorliebspunkte}) * \text{Einschränkungen}$$

Bei der Berechnung ist das Ergebnis 0 sobald einer der Faktoren 0 ist. Aus diesem Grund macht es Sinn zunächst zu überprüfen, ob das Angebotene Lebensmittel den Einschränkungen entspricht. Hierzu wird verglichen ob die ermittelten Healthlabels, mit dem des Benutzers übereinstimmen. Sollte dies nicht der Fall sein, so ist der Abholwert 0.

Falls dies nicht der Fall sein sollte wird im Weiteren zunächst ermittelt, wie das Wetter ist, da dies eine Anfrage an einen externen Dienstgeber ist. Da dies eine kurze Zeit andauern kann wird dieser Schritt an erste Stelle gesetzt. Im Anschluss wird ermittelt, ob der Benutzer mobil (also ihm ein Auto, Fahrrad oder öffentliche Verkehrsmittel zur Verfügung steht) ist oder nicht. Anhand dessen wird entschieden, welche Punktetabelle für die Berechnung in Betracht gezogen werden soll. Die Angabe kann bei jeder Anfrage als Parameter spezifiziert werden.

Im letzten Schritt werden die Vorlieben ermittelt. Hierzu wird das Kaufverhalten des Benutzers analysiert. Jedes eingetragene Lebensmittel wird unter der Ressource Einträge monatlich festgehalten. Hierzu zählen ebenfalls erstellte Reservierungen. Diese Anzahl wird über die Tabelle anschließend ausgewertet. Grundsätzlich leer sich diese Liste für jeden Benutzer jeden Monat automatisch.

Für die Berechnung der Entfernung wurde zwischen zwei Alternativen abgewägt. Zum einem steht die eigene Berechnung mit Hilfe von Längen und Breitengraden zur Verfügung oder es wird ein externer Dienst für diese Berechnung verwendet. Die Einbindung eines Dienstes würde den Vorteil bringen, dass die Ergebnisse genauer wären und eine geringere CPU Auslastung für den Server. Ein gravierender Nachteil wäre, dass die Einbindung Kostenpflichtig wäre, und für ein reales Projekt nicht tragbar, da keine Einnahmen generiert werden. Der Vorteil einer eigenen Berechnung wäre, dass Sie kostenfrei wäre, und dennoch eine relative Abschätzung der Entfernung erlaubt. Der Nachteil wäre, dass diese auf sehr große Distanzen sehr unrealistisch wäre. Da es sich grundsätzlich, um kleinere Distanzen handelt und wir das Projekt möglichst realistisch halten wollen haben wir uns für eine eigene Berechnung der Distanz entschieden. Sollten Gelder für das Projekt zur Verfügung stehen, so könnte die andere Alternative gewählt werden.

Die Berechnung der Distanz findet statt durch:

$$D = \sqrt{dx^2 + dy^2}$$

$$dx = 71.5 * (lon1 - lon2)$$

$$dy = 111.3 * (lat1 - lat2)$$

wobei gilt: $lat1, lat2, lon1, lon2$ entsprechen Breiten und Längenangaben in Grad.

(vgl. Kompf, 2018)

Die Berechnung wurde gewählt, da Sie nicht sehr rechenintensiv ist, und auf kleinere Distanzen genaue Ergebnisse liefert.

Asynchrone Kommunikation

Der angebundene externe Dienstgeber ist unter anderem für die asynchrone Kommunikation mit den Clients verantwortlich. Die Benutzer können über verschiedene Ereignisse informiert werden, welche im Folgenden beschrieben werden.

Ereignis: Eine neue Nachricht liegt vor

Es ist für einen reibungslosen Ablauf der Transaktion ggf. nötig Nachrichten zwischen Anbieter und Abholer auszutauschen. Sollte eine neue Nachricht für einen Benutzer vorliegen, so wird über eine Push Notification darüber Benachrichtigt.

Ereignis: Erstelltes Angebot wurde reserviert

Hat ein Benutzer ein Angebot für andere Benutzer erstellt, so möchte er nicht in regelmäßigen zeitabständen kontrollieren, ob es von einem anderen Benutzer reserviert wurde oder nicht. Aus diesem Grund soll der Benutzer per Push Notification benachrichtigt werden, sobald eine Reservierung für sein Angebot getätigt wurde.

Ereignis: Ein eingetragenes Lebensmittel läuft ab

Hat ein Benutzer Lebensmittel in seinem Kühlschrank und es droht abzulaufen, so wird der Benutzer auch hier per Push Notification benachrichtigt. Diese Asynchrone Kommunikation findet nicht mit Hilfe von Firebase statt, sondern wird auf dem Endgerät des Clients realisiert.

Ereignis: Ein neues und besonders passendes Angebot wurde erstellt

Da es in der Anwendungslogik auf dem Server hauptsächlich, um das ermitteln geeigneter Angebote in der Umgebung geht, so kann der Benutzer sich benachrichtigen lassen, sobald ein optimales Angebot in seiner Umgebung erstellt wurde. Diese Kommunikation wird mit Hilfe von Push Notifications und Firebase realisiert.

Quellenverzeichnis

-Kompf(2018):

Entfernungsberechnung[online]<https://www.kompf.de/gps/distcalc.html>[10.11.201]

- Google Inc. (2018): Material Design, [online] <https://material.io/>

- Katrin Becker (2018): Design Principles | Android Developers [online]
<http://minkhollow.ca/beckerblog/2013/12/28/design-principles-android-developers/> [16.12.2018]

- Deborah Mayhew, The Usability Engineering Lifecycle, 1999 Oxford

- Jacob Nielsen (1995): 10 Usability Heuristics for User Interface [online]
<https://www.nngroup.com/articles/ten-usability-heuristics/> [16.12.2018]

Anhang

User Profiles

Tabelle 1: User Profile – Anbieter #1 (mit guten technischen Kenntnissen)

Demografie	18-50 Jahre alt Männlich oder Weiblich Unterschiedliche Wohnverhältnisse, führen einen Haushalt Gute bis mittelmäßige Sprachkenntnisse
Sozial-ökonomischer Status	Unterschiedliche Berufsfelder mit verschiedenen Einkommensklassen Durchschnittlicher bis guter Bildungsstand Mobil mit eigenem Fahrzeug, öffentlichen Verkehrsmitteln oder nicht mobil
Einschränkungen	Keine Einschränkungen oder verschiedene Einschränkungen wie zum Beispiel: Vegane Ernährungsweise, Vegetarische Ernährungsweise, Laktoseintoleranz, etc
Fachkenntnisse	Durchschnittliche bis Gute Kenntnisse im Bereich Lebensmittel Erkennen wenn ein Lebensmittel deutlich verdorben ist und nicht mehr angeboten werden sollte
Technische Erfahrungen und Kenntnisse	Durchschnittliche bis (sehr) Gute Kenntnisse im Umgang mit mobilen Endgeräten/Computern oder ähnlichen technischen Systemen
Verfügbare Technologien	Smartphone/Computer
Kultureller Hintergrund	Verschiedene Kulturen aber primär in der deutschen Kultur angesiedelt oder vertraut mit dieser
Spezielle Produkterfahrung	Ähnliche Anwendungen wurden bereits verwendet. Es besteht Erfahrung mit dem Erstellen von Angeboten innerhalb von anderen Systemen
Motivation und Einstellung	Möchte die persönlichen Lebensmittelabfälle reduzieren und möchte Lebensmittel die selbst nicht mehr verbraucht werden lieber an andere Menschen abgeben anstatt diese zu entsorgen. Möchte anderen Menschen die finanziell schwächer sind mit nicht mehr benötigten Lebensmitteln versorgen.

Aufgaben	Angebote für Lebensmittel erstellen, die er abgeben möchte. Wichtige Informationen, Beschreibungen und eventuell Bilder von dem Lebensmittel in dem Angebot spezifizieren. Die Standortermittlung im mobilen Endgerät aktivieren damit der Standort des Angebots bekannt ist.
Lernfähigkeit	Gute bis mittelmäßige Lernfähigkeit. Gutes Verständnis des Systems. Kann sich gut mit Änderungen abfinden und diese schnell verstehen.

Tabelle 2: User Profile - Anbieter #2 (mit durchschnittlichen bis schlechten technischen Kenntnissen)

Demografie	18-50 Jahre alt Männlich oder Weiblich Unterschiedliche Wohnverhältnisse, führen einen Haushalt Gute bis mittelmäßige Sprachkenntnisse
Sozial-ökonomischer Status	Unterschiedliche Berufsfelder mit verschiedenen Einkommensklassen Durchschnittlicher bis guter Bildungsstand Mobil mit eigenem Fahrzeug, öffentlichen Verkehrsmitteln oder nicht mobil
Einschränkungen	Keine Einschränkungen oder verschiedene Einschränkungen wie zum Beispiel: Vegane Ernährungsweise, Vegetarische Ernährungsweise, Laktoseintoleranz, etc
Fachkenntnisse	Durchschnittliche Kenntnisse im Bereich Lebensmittel. Erkennen wenn ein Lebensmittel deutlich verdorben ist und nicht mehr angeboten werden sollte
Technische Erfahrungen und Kenntnisse	Durchschnittliche bis Schlechte Kenntnisse im Umgang mit mobilen Endgeräten/Computern oder ähnlichen technischen Systemen
Verfügbare Technologien	Smartphone/Computer
Kultureller Hintergrund	Verschiedene Kulturen aber primär in der deutschen Kultur angesiedelt oder vertraut mit dieser

Spezielle Produkterfahrung	Hat ähnliche Produkte zuvor noch nicht oder nur begrenzt genutzt. Es besteht keine oder nur sehr geringe Erfahrung im Erstellen von Angeboten
Motivation und Einstellung	Möchte die persönlichen Lebensmittelabfälle reduzieren und möchte Lebensmittel die selbst nicht mehr verbraucht werden lieber an andere Menschen abgeben anstatt diese zu entsorgen. Möchte anderen Menschen die finanziell schwächer sind mit nicht mehr benötigten Lebensmitteln versorgen.
Aufgaben	Angebote für Lebensmittel erstellen, die er abgeben möchte. Wichtige Informationen, Beschreibungen und eventuell Bilder von dem Lebensmittel in dem Angebot spezifizieren. Die Standortermittlung im mobilen Endgerät aktivieren damit der Standort des Angebots bekannt ist.
Lernfähigkeit	Mittelmäßige bis schlechte Lernfähigkeit. Benötigt viel Zeit, um sich innerhalb eines neuen Designs zurechtzufinden oder neue Funktionen zu erlernen. Benötigt eine Art von Anleitung um Neues zu verstehen

Tabelle 3: User Profile - Abholer #1 (mit guten technischen Kenntnissen)

Demografie	18-50 Jahre alt Männlich oder Weiblich Unterschiedliche Wohnverhältnisse Gute bis mittelmäßige Sprachkenntnisse
Sozial-ökonomischer Status	Unterschiedliche Berufsfelder mit verschiedenen Einkommensklassen Durchschnittlicher bis guter Bildungsstand Mobil mit eigenem Fahrzeug, öffentlichen Verkehrsmitteln oder nicht mobil
Einschränkungen	Keine Einschränkungen oder verschiedene Einschränkungen wie zum Beispiel: Vegane Ernährungsweise, Vegetarische Ernährungsweise, Laktoseintoleranz, etc
Fachkenntnisse	Durchschnittliche Kenntnisse im Bereich Lebensmittel. Erkennt, wenn ein angebotenes Lebensmittel deutlich abgelaufen ist und nicht mehr abgeholt werden sollte

Technische Erfahrungen und Kenntnisse	Durchschnittliche bis gute Kenntnisse im Umgang mit mobilen Endgeräten/Computern oder ähnlichen technischen Systemen
Verfügbare Technologien	Smartphone/Computer
Kultureller Hintergrund	Verschiedene Kulturen aber primär in der deutschen Kultur angesiedelt oder vertraut mit dieser
Spezielle Produkterfahrung	Hat ähnliche Produkte zuvor bereits genutzt. Es besteht Erfahrung in der Suche nach Angeboten sowie dem Prozess des Abholens dieser.
Motivation und Einstellung	Möchte dazu beitragen, dass die Durchschnittlichen Lebensmittelabfälle in Deutschland reduziert werden, in dem er nicht mehr gewollte Lebensmittel abholt und verbraucht. Möchte Geld sparen. Sucht bestimmte Zutaten die beispielsweise zum Kochen fehlen und möchte nicht wegen einer Zutat einkaufen gehen.
Aufgaben	Ein Gesundheitsprofil erstellen um passende Angebote von dem System empfohlen zu bekommen. Die Standortermittlung im mobilen Endgerät aktivieren damit die Distanz zu den Angeboten ermittelt werden kann. Angebote für Lebensmittel suchen oder wahrnehmen. Angebote reservieren und diese bei den Anbietern abholen.
Lernfähigkeit	Gute bis mittelmäßige Lernfähigkeit. Gutes Verständnis des Systems. Kann sich gut mit Änderungen abfinden und diese schnell verstehen.

Tabelle 4: User Profile - Abholer #2 (mit guten technischen Kenntnissen & geringem Einkommen)

Demografie	18-50 Jahre alt Männlich oder Weiblich Unterschiedliche Wohnverhältnisse, eventuell vermehrt sozial schwächere Verhältnisse Gute bis mittelmäßige Sprachkenntnisse
Sozial-ökonomischer Status	Unterschiedliche Berufsfelder mit verschiedenen Einkommensklassen, primär geringes Einkommen Durchschnittlicher bis guter Bildungsstand Mobil mit eigenem Fahrzeug, öffentlichen Verkehrsmitteln oder nicht mobil

Einschränkungen	Keine Einschränkungen oder verschiedene Einschränkungen wie zum Beispiel: Vegane Ernährungsweise, Vegetarische Ernährungsweise, Laktoseintoleranz, etc
Fachkenntnisse	Durchschnittliche Kenntnisse im Bereich Lebensmittel. Erkennt, wenn ein angebotenes Lebensmittel deutlich abgelaufen ist und nicht mehr abgeholt werden sollte
Technische Erfahrungen und Kenntnisse	Durchschnittliche bis gute Kenntnisse im Umgang mit mobilen Endgeräten/Computern oder ähnlichen technischen Systemen
Verfügbare Technologien	Smartphone/Computer
Kultureller Hintergrund	Verschiedene Kulturen aber primär in der deutschen Kultur angesiedelt oder vertraut mit dieser
Spezielle Produkterfahrung	Hat ähnliche Produkte zuvor bereits genutzt. Es besteht Erfahrung in der Suche nach Angeboten sowie dem Prozess des Abholens dieser.
Motivation und Einstellung	Möchte dazu beitragen, dass die Durchschnittlichen Lebensmittelabfälle in Deutschland reduziert werden, in dem er nicht mehr gewollte Lebensmittel abholt und verbraucht. Hat wenig Geld und kann sich teure Lebensmittel aus dem Supermarkt nicht leisten und ist daher auf die Angebote angewiesen.
Aufgaben	Ein Gesundheitsprofil erstellen um passende Angebote von dem System empfohlen zu bekommen. Die Standortermittlung im mobilen Endgerät aktivieren damit die Distanz zu den Angeboten ermittelt werden kann. Angebote für Lebensmittel suchen oder wahrnehmen. Angebote reservieren und diese bei den Anbietern abholen.
Lernfähigkeit	Gute bis mittelmäßige Lernfähigkeit. Gutes Verständnis des Systems. Kann sich gut mit Änderungen abfinden und diese schnell verstehen.

Tabelle 5: User Profile - Abholer #3 (mit durchschnittlichen bis schlechten technischen Kenntnissen)

Demografie	18-50 Jahre alt Männlich oder Weiblich Unterschiedliche Wohnverhältnisse Gute bis mittelmäßige Sprachkenntnisse
Sozial-ökonomischer Status	Unterschiedliche Berufsfelder mit verschiedenen Einkommensklassen Durchschnittlicher bis guter Bildungsstand Mobil mit eigenem Fahrzeug, öffentlichen Verkehrsmitteln oder nicht mobil
Einschränkungen	Keine Einschränkungen oder verschiedene Einschränkungen wie zum Beispiel: Vegane Ernährungsweise, Vegetarische Ernährungsweise, Laktoseintoleranz, etc
Fachkenntnisse	Durchschnittliche Kenntnisse im Bereich Lebensmittel. Erkennt, wenn ein angebotenes Lebensmittel deutlich abgelaufen ist und nicht mehr abgeholt werden sollte
Technische Erfahrungen und Kenntnisse	Durchschnittliche bis Schlechte Kenntnisse im Umgang mit mobilen Endgeräten/Computern oder ähnlichen technischen Systemen
Verfügbare Technologien	Smartphone/Computer
Kultureller Hintergrund	Verschiedene Kulturen aber primär in der deutschen Kultur angesiedelt oder vertraut mit dieser
Spezielle Produkterfahrung	Hat ähnliche Produkte zuvor noch nicht oder nur begrenzt genutzt. Es besteht keine oder nur sehr geringe Erfahrung im Suchen und Abholen von Angeboten.
Motivation und Einstellung	Möchte dazu beitragen, dass die Durchschnittlichen Lebensmittelabfälle in Deutschland reduziert werden, in dem er nicht mehr gewollte Lebensmittel abholt und verbraucht. Möchte Geld sparen. Sucht bestimmte Zutaten die beispielsweise zum Kochen fehlen und möchte nicht wegen einer Zutat einkaufen gehen.

Aufgaben	Ein Gesundheitsprofil erstellen um passende Angebote von dem System empfohlen zu bekommen. Die Standortermittlung im mobilen Endgerät aktivieren damit die Distanz zu den Angeboten ermittelt werden kann. Angebote für Lebensmittel suchen oder wahrnehmen. Angebote reservieren und diese bei den Anbietern abholen.
Lernfähigkeit	Mittelmäßige bis schlechte Lernfähigkeit. Benötigt viel Zeit um sich innerhalb eines neues Designs zurechtzufinden oder neue Funktionen zu erlernen. Benötigt eine Art von Anleitung um Neues zu verstehen

Tabelle 6: User Profile - Abholer #4 (mit durchschnittlichen bis geringen technischen Kenntnissen & geringem Einkommen)

Demografie	18-50 Jahre alt Männlich oder Weiblich Unterschiedliche Wohnverhältnisse, eventuell vermehrt sozial schwächere Verhältnisse Gute bis mittelmäßige Sprachkenntnisse
Sozial-ökonomischer Status	Unterschiedliche Berufsfelder mit verschiedenen Einkommensklassen, primär geringes Einkommen Durchschnittlicher bis guter Bildungsstand Mobil mit eigenem Fahrzeug, öffentlichen Verkehrsmitteln oder nicht mobil
Einschränkungen	Keine Einschränkungen oder verschiedene Einschränkungen wie zum Beispiel: Vegane Ernährungsweise, Vegetarische Ernährungsweise, Laktoseintoleranz, etc
Fachkenntnisse	Durchschnittliche Kenntnisse im Bereich Lebensmittel. Erkennt, wenn ein angebotenes Lebensmittel deutlich abgelaufen ist und nicht mehr abgeholt werden sollte
Technische Erfahrungen und Kenntnisse	Durchschnittliche bis Schlechte Kenntnisse im Umgang mit mobilen Endgeräten/Computern oder ähnlichen technischen Systemen
Verfügbare Technologien	Smartphone/Computer
Kultureller Hintergrund	Verschiedene Kulturen aber primär in der deutschen Kultur angesiedelt oder vertraut mit dieser

Spezielle Produkterfahrung	Hat ähnliche Produkte zuvor noch nicht oder nur begrenzt genutzt. Es besteht keine oder nur sehr geringe Erfahrung im Suchen und Abholen von Angeboten.
Motivation und Einstellung	Möchte dazu beitragen, dass die Durchschnittlichen Lebensmittelabfälle in Deutschland reduziert werden, in dem er nicht mehr gewollte Lebensmittel abholt und verbraucht. Hat wenig Geld und kann sich teure Lebensmittel aus dem Supermarkt nicht leisten und ist daher auf die Angebote angewiesen.
Aufgaben	Ein Gesundheitsprofil erstellen um passende Angebote von dem System empfohlen zu bekommen. Die Standortermittlung im mobilen Endgerät aktivieren damit die Distanz zu den Angeboten ermittelt werden kann. Angebote für Lebensmittel suchen oder wahrnehmen. Angebote reservieren und diese bei den Anbietern abholen.
Lernfähigkeit	Mittelmäßige bis schlechte Lernfähigkeit. Benötigt viel Zeit um sich innerhalb eines neuen Designs zurechtzufinden oder neue Funktionen zu erlernen. Benötigt eine Art von Anleitung um Neues zu verstehen

Tabelle 7: User Profile - Normalbenutzer #1 (mit guten technischen Kenntnissen)

Demografie	18-50 Jahre alt Männlich oder Weiblich Unterschiedliche Wohnverhältnisse Gute bis mittelmäßige Sprachkenntnisse
Sozial-ökonomischer Status	Unterschiedliche Berufsfelder mit verschiedenen Einkommensklassen Durchschnittlicher bis guter Bildungsstand Mobil mit eigenem Fahrzeug, öffentlichen Verkehrsmitteln oder nicht mobil
Einschränkungen	Keine Einschränkungen oder verschiedene Einschränkungen wie zum Beispiel: Vegane Ernährungsweise, Vegetarische Ernährungsweise, Laktoseintoleranz, etc

Fachkenntnisse	Durchschnittliche Kenntnisse im Bereich Lebensmittel. Durchschnittliche bis Mangelnde Kenntnisse im Bereich Lebensmittelhaltbarkeit
Technische Erfahrungen und Kenntnisse	Durchschnittliche bis gute Kenntnisse im Umgang mit mobilen Endgeräten/Computern oder ähnlichen technischen Systemen
Verfügbare Technologien	Smartphone/Computer
Kultureller Hintergrund	Verschiedene Kulturen aber primär in der deutschen Kultur angesiedelt oder vertraut mit dieser
Spezielle Produkterfahrung	Hat ähnliche Produkte zuvor bereits genutzt. Hat Erfahrung in der Verwendung von QR Codes und der Spezifikation persönlicher Daten.
Motivation und Einstellung	Möchte dazu beitragen die eigenen Lebensmittelabfälle zu reduzieren indem er sich eine bessere Übersicht über die eigenen Lebensmittel verschafft. Würde gerne wissen, wie lange Obst und Gemüse haltbar sind und wie diese am besten gelagert werden sollten. Will aktiv Lebensmittel anbieten und abholen. Findet, dass genießbare Lebensmittel nicht entsorgt werden sollten.
Aufgaben	Die gekauften Lebensmittel im System eintragen. Ein Gesundheitsprofil mit persönlichen Einschränkungen und Präferenzen anlegen. Die Standortermittlung des mobilen Endgerätes aktivieren. Verbrauchte Lebensmittel aus dem System entfernen. Bei Bedarf Teilnahme an den Aufgaben der Abholer und Anbieter.
Lernfähigkeit	Gute bis mittelmäßige Lernfähigkeit. Gutes Verständnis des Systems. Kann sich gut mit Änderungen abfinden und diese schnell verstehen.

Tabelle 8: User Profile - Normalbenutzer #2 (mit durchschnittlichen bis geringen technischen Kenntnissen)

Demografie	18-50 Jahre alt Männlich oder Weiblich Unterschiedliche Wohnverhältnisse Gute bis mittelmäßige Sprachkenntnisse
------------	--

Sozial-ökonomischer Status	<p>Unterschiedliche Berufsfelder mit verschiedenen Einkommensklassen</p> <p>Durchschnittlicher bis guter Bildungsstand</p> <p>Mobil mit eigenem Fahrzeug, öffentlichen Verkehrsmitteln oder nicht mobil</p>
Einschränkungen	<p>Keine Einschränkungen oder verschiedene Einschränkungen wie zum Beispiel: Vegane Ernährungsweise, Vegetarische Ernährungsweise, Laktoseintoleranz, etc</p>
Fachkenntnisse	<p>Durchschnittliche Kenntnisse im Bereich Lebensmittel.</p> <p>Durchschnittliche bis Mangelnde Kenntnisse im Bereich Lebensmittelhaltbarkeit</p>
Technische Erfahrungen und Kenntnisse	<p>Durchschnittliche bis Geringe Kenntnisse im Umgang mit mobilen Endgeräten/Computern oder ähnlichen technischen Systemen</p>
Verfügbare Technologien	<p>Smartphone/Computer</p>
Kultureller Hintergrund	<p>Verschiedene Kulturen aber primär in der deutschen Kultur angesiedelt oder vertraut mit dieser</p>
Spezielle Produkterfahrung	<p>Hat ähnliche Produkte zuvor noch nicht genutzt.</p> <p>Hat keine bis geringe Erfahrung in der Verwendung von QR Codes und der Spezifikation persönlicher Daten.</p>
Motivation und Einstellung	<p>Möchte dazu beitragen die eigenen Lebensmittelabfälle zu reduzieren indem er sich eine bessere Übersicht über die eigenen Lebensmittel verschafft.</p> <p>Würde gerne wissen, wie lange Obst und Gemüse haltbar sind und wie diese am besten gelagert werden sollten.</p> <p>Will aktiv Lebensmittel anbieten und abholen. Findet, dass genießbare Lebensmittel nicht entsorgt werden sollten.</p>
Aufgaben	<p>Die gekauften Lebensmittel im System eintragen. Ein Gesundheitsprofil mit persönlichen Einschränkungen und Präferenzen anlegen. Die Standortermittlung des mobilen Endgerätes aktivieren. Verbrauchte Lebensmittel aus dem System entfernen. Bei Bedarf Teilnahme an den Aufgaben der Abholer und Anbieter.</p>
Lernfähigkeit	<p>Mittelmäßige bis schlechte Lernfähigkeit.</p> <p>Benötigt viel Zeit um sich innerhalb eines neuen Designs zurechtzufinden oder neue Funktionen zu erlernen.</p> <p>Benötigt eine Art von Anleitung um Neues zu verstehen</p>

Aufgaben	Die gekauften Lebensmittel im System eintragen. Ein Gesundheitsprofil mit persönlichen Einschränkungen und Präferenzen anlegen. Die Standortermittlung des mobilen Endgerätes aktivieren. Verbrauchte Lebensmittel aus dem System entfernen. Bei Bedarf Teilnahme an den Aufgaben der Abholer und Anbieter.
Lernfähigkeit	Mittelmäßige bis schlechte Lernfähigkeit. Benötigt viel Zeit um sich innerhalb eines neuen Designs zurechtzufinden oder neue Funktionen zu erlernen. Benötigt eine Art von Anleitung um Neues zu verstehen

Contextual Task Analysis

Tabelle 11: Contextual Observation/Interview 2. Privatperson (als Anbieter)

Actor	Trigger	Use Case	Task Scenarios Sequence	Problems
Privatperson	Bemerkt, dass Lebensmittel übriggeblieben sind, welche er nichtmehr verwenden möchte.	Nach einer Möglichkeit suchen die Lebensmittel anderen Menschen anzubieten.	<p>1. Privatperson bemerkt, dass Lebensmittel in seinen Haushalt sind, welche er nicht mehr verwenden will bzw. kann.</p> <p>2. Sucht Lebensmittel heraus und legt sie ggf. zusammen, um einen besseren Überblick zu behalten.</p> <p>3. Privatperson sucht nach Möglichkeiten seine Lebensmittel abzugeben. (Internet/Social Media)</p>	<p>1. Privatpersonen machen sich i.d.R nicht die Mühe (besonders bei einzelner Lebensmittel) nach Möglichkeiten zur Abgabe oder Ähnlichem zu suchen.</p> <p>2. Lebensmittel kann eventuell bereits abgelaufen sein und sollte nichtmehr angeboten werden.</p>
	Bemerkt, dass Lebensmittel übriggeblieben sind, welche er nichtmehr verwenden möchte.	Lebensmittel im Bekanntenkreis anbieten.	<p>1. Privatperson bemerkt, dass Lebensmittel in seinen Haushalt sind, welche er nicht mehr verwenden will bzw. kann.</p> <p>2. Sucht Lebensmittel heraus und legt sie ggf. zusammen, um einen besseren Überblick zu behalten.</p> <p>3. Privatperson überlegt sich, wer aus seinem Bekannten Kreis das Lebensmittel gebrauchen könnte.</p>	<p>1. Privatpersonen machen sich i.d.R nicht die Mühe (besonders bei einzelner Lebensmittel) nach Möglichkeiten zur Abgabe oder Ähnlichem zu suchen.</p> <p>2. Lebensmittel kann eventuell bereits abgelaufen sein und sollte nichtmehr angeboten werden.</p>

			<p>4. Privatperson Kontaktiert diese Person (Internet/Telefon/Direkter Kontakt).</p> <p>5. Privatperson übergibt die Lebensmittel.</p>	<p>3. Es gibt niemanden im Bekanntenkreis, den das Lebensmittel interessieren könnte.</p> <p>4. Alle möglichen Interessenten wohnen zu weit weg.</p>
	<p>Bemerkt, dass Lebensmittel übriggeblieben sind, welche er nichtmehr verwenden möchte.</p>	<p>Lebensmittel an Organisationen spenden.</p>	<p>1. Privatperson bemerkt, dass Lebensmittel in seinen Haushalt sind, welche er nicht mehr verwenden will bzw. kann.</p> <p>2. Sucht Lebensmittel heraus und legt sie ggf. zusammen, um einen besseren Überblick zu behalten.</p> <p>3. Privatperson sucht nach möglichen Organisationen die Lebensmittel zu spenden.</p> <p>4. Privatperson setzt sich mit der Organisation in Kontakt und erkundigt sich über den Abgabeprozess.</p>	<p>1. Privatpersonen machen sich i.d.R nicht die Mühe (besonders bei individuellem Lebensmittel) nach Möglichkeiten zur Abgabe oder Ähnlichem zu suchen.</p> <p>2. Lebensmittel kann eventuell bereits abgelaufen sein und sollte nichtmehr angeboten werden.</p> <p>3. Es gibt keine Organisation in der Nähe, bei der gespendet werden kann.</p>

Tabelle 12: Contextual Observation/Interview 3. Privatperson (als Abholer)

Actor	Trigger	Use Case	Task Scenarios Sequence	Problems
Privatperson	<p>Privatperson, besitzt wenig Geld und möchte deshalb nach kostenlosen Angeboten in seiner Umgebung suchen.</p>	<p>Nach einer Möglichkeit suchen Lebensmittel kostengünstig abzuholen</p>	<p>1. Privatperson stellt fest, dass er Lebensmittel benötigt</p> <p>2. Privatperson sucht nach Quellen zur Recherche von Möglichen Angeboten.</p> <p>3. Privatperson vergleicht verschiedene Angebote nach seinen Bedürfnissen.</p> <p>4. Privatperson wählt ein geeignetes Angebot aus.</p>	<p>1. Kann sehr zeitaufwändig sein ein geeignetes Angebot zu finden.</p> <p>2. Weiß ggf. nicht über den Zustand des Angebots Bescheid.</p> <p>3. Lebensmittel kann nicht dem erwarteten entsprechen.</p>
	<p>Privatperson hat Hunger und möchte Angebote aus der Umgebung wahrnehmen.</p>	<p>Nach einer Möglichkeit suchen Lebensmittel bei Familie/Freunde</p>	<p>1. Privatperson überlegt sich wen er kontaktieren kann, um Lebensmittel zu erhalten.</p> <p>2. Privatperson wählt eine Person aus.</p>	<p>1. Kann sehr zeitaufwändig sein ein geeignetes Angebot zu finden.</p>

		bzw. Bekannten abholen.	<p>3. Privatperson kontaktiert diese Person und erkundigt sich nach möglichen Angeboten.</p> <p>4. Privatperson nimmt das Angebot an oder sucht andere Möglichkeiten.</p> <p>5. Bei gefundenem Angebot, welche für ihm annehmbar ist macht die Privatperson mit dem Anbieter einen Treffpunkt und Uhrzeit aus.</p> <p>6. Privatperson entscheidet sich, wie er zum Zielort gelangen soll. (Auto, zu Fuß, öffentliche Verkehrsmittel)</p> <p>7. Privatperson holt Lebensmittel ab.</p>	<p>2. Weiß ggf. nicht über den Zustand des Angebots Bescheid.</p> <p>3. Lebensmittel kann nicht dem erwarteten entsprechen.</p> <p>4. Ggf. kann es für den Abholer unangenehm sein bei seiner Familie oder Freunden Lebensmittel zu holen.</p> <p>5. Es kann sein keine Angebote zu finden, da der Freundeskreis oder die Familie sehr begrenzt ist.</p>
	Privatperson hat Hunger und möchte Angebote aus der Umgebung wahrnehmen.	Nach einer Möglichkeit suchen Lebensmittel bei Organisationen abzuholen bzw. kostengünstig zu erwerben.	<p>1. Privatperson erkundigt sich über Abholmöglichkeiten in seiner Nähe (im Internet, durch Bekannte)</p> <p>2. Privatperson vergleicht verschiedene Abholmöglichkeiten miteinander</p> <p>3. Privatperson wählt die Möglichkeit aus, die seinen Bedürfnissen entspricht</p> <p>4. Privatperson erkundigt sich über Abholort und Zeitpunkt</p> <p>5. Privatperson geht zur entsprechenden Zeit zu dem angegebenen Abholort</p> <p>6. Privatperson nimmt die Lebensmittel entgegen</p>	<p>1. Kann sehr zeitaufwändig sein ein geeignetes Angebot zu finden.</p> <p>2. Weiß ggf. nicht über den Zustand des Angebots Bescheid.</p> <p>3. Lebensmittel kann nicht dem erwarteten entsprechen.</p> <p>4. Privatperson besitzt keinen Ausweis der ihn dazu berechtigt die Lebensmittel zu erhalten bzw. zu erwerben.</p>

Task Scenarios

Task: Nach einer geeigneten Verwendungsmöglichkeit für Lebensmittel suchen

User: Privatperson (Als Normaler Benutzer)

Description: Bei dem Durchsehen ihres Kühlschranks bemerkt Lisa, eine Vegetarierin, dass sie noch etwas Gemüse von dem gestrigen Kochen übrighat. Dieses würde sie gerne verbrauchen, aber weiß

nicht was genau sie daraus machen könnte. Sie möchte gerne nach einer passenden Möglichkeit suchen.

Task Flow:

4. Lisa sieht in ihrem Kühlschrank nach den einzelnen Lebensmitteln die sie verbrauchen könnte nach. Diese legt sie sich auf ihre Küchen-Arbeitsfläche.
5. Lisa hat alle Lebensmittel die sie gerne verbrauchen möchte herausgesucht.
6. Lisa stellt fest, dass sie in letzter Zeit oft das gleiche gekocht hat und daher gerne einmal ein neues Gericht kochen würde. Sie beschließt nach möglichen Rezepten zu suchen.

Variante 1

4. Lisa entschließt sich dazu, in einem ihrer Kochbücher nach einem passenden Rezept zu suchen.
5. Sie geht zu ihrem Bücherregal und holt sich ein passendes Kochbuch heraus.
6. Im Inhaltsverzeichnis des Buches sucht sie nach einem der Lebensmittel um ein passendes Rezept mit diesem Lebensmittel zu finden.
7. Lisa öffnet die Seite mit dem gefundenen Rezept und stellt fest, dass das gefundene Rezept Fleisch beinhaltet, welches sie als Vegetarierin nicht essen kann.
8. Lisa blättert zurück zu dem Inhaltsverzeichnis und sucht ein neues Rezept mit einem der vorhandenen Lebensmittel.
9. Die Suche nach passenden Rezepten wird so oft wiederholt, bis Lisa ein Rezept findet, das ihren Vorstellungen und Einschränkungen entspricht.
10. Lisa bereitet das Rezept zu.

Variante 2

4. Lisa beschließt im Internet nach einem passenden Rezept zu suchen.
5. Lisa setzt sich an ihren Computer und sucht in einer Suchmaschine nach einer Internetseite die Rezepte zur Verfügung stellt. Sie wählt eine der Seiten aus und besucht diese.
6. In der Suchzeile der Internetseite spezifiziert Lisa eines der Lebensmittel, das sie verbrauchen möchte.
7. (Optional) Lisa gibt innerhalb der Filterfunktion der Internetseite an, dass sie sich vegetarisch ernährt
8. In der angezeigten Liste aller Rezepte sucht Lisa das Rezept aus, das für sie am passendsten erscheint.
9. Lisa stellt fest, dass sie für das Rezept viele Lebensmittel benötigt, welche sie erst einkaufen müsste. Sie beschließt doch lieber ein anderes Rezept auszuwählen.
10. Lisa wiederholt die Suche so oft, bis sie ein passendes Rezept findet.

Task Closure: Dieses Szenario kann zwischen 10 -30 Minuten andauern. Hierbei kommt es zum einen darauf an, wie oft der Benutzer ein nicht passendes Rezept findet und den Suchvorgang wiederholen muss und zum anderen wie genau und ob er seine Ernährungseinschränkungen spezifizieren kann.

Um diese Aufgabe zu unterstützen sollte die Benutzeroberfläche:

- Eine Möglichkeit bieten, Einschränkungen oder Ernährungsstile zu spezifizieren, um so die Anzahl an „Fehlgriffen“ bei der Rezeptsuche zu reduzieren.
- Eine Möglichkeit bieten, nach Rezepten mit mehr als nur einem vorgegebenen Lebensmittel zu suchen.

- Die Recherchearbeit nach passenden Rezepten zeitlich verkürzen.
- Eine Übersicht über die vorhandenen Lebensmittel zur Verfügung stellen, um die Suche nach passenden Rezepten einfacher zu gestalten.
-

Task: Lebensmittel Familienmitgliedern oder Bekannten anbieten

User: Privatperson (als Anbieter)

Description: Hannah fliegt für 2 Wochen in den Urlaub. Sie stellt am Tag vor der Abreise fest, dass sie noch frisches Gemüse im Kühlschrank hat, welches zeitnah verbraucht werden sollte. Sie beschließt das Gemüse an jemanden abzugeben.

Task Flow:

1. Hannah durchsucht ihren Kühlschrank nach Lebensmitteln die sie gerne abgeben möchte. Diese Lebensmittel legt sie im Kühlschrank beiseite, um Überblick über diese zu behalten.
2. Hannah überlegt sich, wer diese Lebensmittel gebrauchen könnte. Sie entscheidet sich dazu, ihre beste Freundin Mia zu fragen.
3. Hannah kontaktiert Mia und bietet ihr die abzugebenden Lebensmittel an. Mia kann diese jedoch nicht gebrauchen, weil sie heute erst einkaufen war und ihr Kühlschrank voll ist.
4. Hannah überlegt sich eine weitere Person die die Lebensmittel gebrauchen könnte. Sie kontaktiert ihre Tante Greta.
5. Tante Greta kann einige der Lebensmittel gut gebrauchen und akzeptiert das Angebot.
6. Hannah packt die Lebensmittel die Tante Greta gebrauchen kann in eine Tasche.
7. Mit den Lebensmitteln macht sich Hannah auf den Weg zu Tante Greta und übergibt ihr die Lebensmittel. Dann begibt sich Hannah auf den Weg nach Hause.

Task Closure: Das Szenario kann 30 Minuten bis mehrere Stunden in Anspruch nehmen. Die Zeitspanne hängt vor allem von dem Kontaktmedium ab, welches verwendet wird. Auf einen Anruf erfolgt eine schnellere Antwort als auf eine Chat-Nachricht oder eine E-Mail. Zusätzlich kann es passieren, dass erst mehrere Personen kontaktiert werden müssen, bevor jemand gefunden wird, der das Lebensmittel benötigt.

Um diese Aufgabe zu unterstützen sollte die Benutzeroberfläche:

- Eine schnelle Möglichkeit bieten, jemandem Lebensmittel anzubieten und zu kontaktieren.
- Eine Übersicht über die vorhandenen Lebensmittel zur Verfügung stellen, damit man schnell sieht, was man Zuhause hat.
- Dem Benutzer die Verantwortung abnehmen, selbst einen passenden Empfänger der Lebensmittel zu suchen.
- Bei Bedarf eine Route zum Empfänger zur Verfügung stellen, falls der Weg dorthin unklar ist

Task: Lebensmittel bei Familienmitgliedern oder Bekannten abholen

User: Privatperson (als Abholer)

Description: Maria ist diesen Sonntag damit beschäftigt Weihnachts-Plätzchen zu backen. Während sie die Zutaten mischt fällt ihr auf, dass sie nur noch 3 Eier im Haus hat obwohl sie für den Teig 6 benötigt. Maria muss nun jemanden finden, der ihr aushelfen kann.

Task Flow:

1. Maria durchsucht ihre Küche um sicher zu gehen, dass sie wirklich keine Eier mehr im Haus hat. Nachdem sie keine gefunden hat notiert sie sich wie viele Eier sie benötigt.
2. Maria überlegt sich, wer in ihrer Nähe eventuell Eier hätte, die sie sich abholen könnte. Sie entschließt sich dazu ihre Mutter zu kontaktieren.
3. Maria kontaktiert ihre Mutter und fragt nach den Eiern. Ihre Mutter hat keine Eier zuhause da sie selbst heute Plätzchen gebacken hat.
4. Maria überlegt erneut, wen sie fragen könnte. Sie kontaktiert ihre Schwester Sylvia.
5. Sylvia ist nicht für Maria erreichbar und nach mehreren Kontaktversuchen gibt Maria auf.
6. Maria überlegt wen sie noch kontaktieren könnte. Sie ruft ihre Nachbarin Ingrid an.
7. Ingrid hat noch Eier zur Verfügung, welche Maria abholen kann.
8. Maria macht sich auf den Weg zu ihrer Nachbarin.
9. Ingrid übergibt Maria die Eier und Maria begibt sich wieder auf den Weg nach Hause.

Task Closure: Das Szenario kann 30 Minuten bis mehrere Stunden in Anspruch nehmen. Die Zeitspanne hängt vor allem von dem Kontaktmedium ab, welches verwendet wird. Auf einen Anruf erfolgt eine schnellere Antwort als auf eine Chat-Nachricht oder eine E-Mail. Zusätzlich kann es passieren, dass erst mehrere Personen kontaktiert werden müssen, bevor jemand gefunden wird, der das Lebensmittel zur Verfügung stellen kann. Zusätzlich kann es vorkommen, dass keine Familie oder Bekannte in der Nähe wohnen, und man so niemanden kontaktieren kann.

Um diese Aufgabe zu unterstützen sollte die Benutzeroberfläche:

- Eine schnelle Möglichkeit bieten, Menschen zu finden, bei denen man Lebensmittel abholen kann.
- Bei Bedarf eine Route zum Anbieter zur Verfügung stellen, falls der Weg dorthin unklar ist.
- Eine Möglichkeit bieten, nach konkreten Lebensmittel-Angeboten zu suchen.
- Die Entfernung zum Anbieter anzeigen.
- Eine Übersicht über die vorhandenen Lebensmittel zur Verfügung stellen, damit man schnell sieht, was man Zuhause hat und was gegebenenfalls fehlt.

Task Scenario Re-engineering

Im Folgenden werden die Task Scenarios, welche zuvor im Rahmen der Contextual Task Analysis erstellt wurden, überarbeitet. Hierbei werden die gleichen Scenarios wie zuvor verwendet und angepasst, um so die Veränderungen durch das System zu veranschaulichen. Zusätzlich wird ein Task Scenario für das Hinzufügen eines Lebensmittels zum Inventar erstellt.

User: Privatperson

Description: Bei dem Durchsehen ihres Kühlschranks bemerkt Lisa, eine Vegetarierin, dass sie noch etwas Gemüse von dem gestrigen Kochen übrighat. Dieses würde sie gerne verbrauchen, aber weiß

nicht was genau sie daraus machen könnte. Sie möchte gerne nach einer passenden Möglichkeit suchen.

Task Flow:

10. Lisa startet die Anwendung um eine passende Verwendungsmöglichkeit zu suchen.
11. Lisa loggt sich mit ihrem Usernamen und ihrem Passwort im System ein.
12. Lisa wählt die Option „Verwendungsmöglichkeit finden“ und gibt das zu verbrauchende Lebensmittel in der Suchzeile ein.
13. Das System zeigt Lisa eine Liste von Vorschlägen an, welche ihre im Gesundheitsprofil festgelegten Einschränkungen sowie die im Lebensmittelinventar eingetragenen Lebensmittel berücksichtigt.
14. Lisa wählt eine Möglichkeit aus die für sie Interessant klingt und lässt sich diese anzeigen.

Task Closure: Das Szenario nimmt mithilfe des Systems nur noch wenige Minuten in Anspruch, da das Finden von nicht passenden Rezepten anhand der Berücksichtigung von Einschränkungen und vorhandenen Lebensmitteln verhindert wird. Dem Benutzer werden nur noch Rezepte angezeigt, die für ihn relevant sein könnten, wodurch die Suche erleichtert wird. Außerdem müssen fortan nicht mehr mehrere Quellen besucht werden, sondern das System bündelt mehrere Quellen und reicht somit als „Recherche-Medium“ aus.

Task: Über die Haltbarkeit eines Lebensmittels erkundigen

User: Privatperson/Normaler Benutzer

Description: Jens würde zum Kochen gerne die letzte Zwiebel in seinem Kühlschrank verwenden. Es ist schon eine Weile her, dass er das Netz mit Zwiebeln gekauft hat und ist sich daher nicht ganz sicher ob die letzte Zwiebel noch haltbar ist. Dies würde er nun gerne herausfinden.

Task Flow:

15. Jens startet die Anwendung um das Haltbarkeitsdatum der Zwiebel zu ermitteln.
16. Jens loggt sich mit seinem Usernamen und seinem Passwort im System ein.
17. Jens besucht sein Lebensmittelinventar und durchsucht es um die Zwiebel zu finden.
18. Jens findet die Zwiebel und liest das von dem System ermittelte Haltbarkeitsdatum ab.

Task Closure: Das Szenario hat insgesamt höchstens eine Minute in Anspruch genommen, da das Haltbarkeitsdatum des jeweiligen Lebensmittels nur im Inventar abgelesen werden musste.

Task: Lebensmittel anderen Personen anbieten

User: Privatperson/Anbieter

Description: Hannah fliegt für 2 Wochen in den Urlaub. Sie stellt am Tag vor der Abreise fest, dass sie noch frisches Gemüse im Kühlschrank hat, welches zeitnah verbraucht werden sollte. Sie beschließt das Gemüse an jemanden abzugeben.

Task Flow:

19. Hannah startet die Anwendung um ein Angebot für das Gemüse zu erstellen.
20. Hannah loggt sich mit ihrem Usernamen und ihrem Passwort im System ein.

21. Hannah öffnet ihr Lebensmittelinventar und wählt die Lebensmittel aus die sie weitergeben möchte.
22. Hannah wählt die Optionen „Ausgewählte Lebensmittel anbieten“
23. Hannah erstellt eine Beschreibung für das Angebot.
24. Hannah macht ein Foto von den Lebensmitteln die sie anbieten möchte und fügt dieses dem Angebot hinzu.
25. Hannah gibt dem System die Erlaubnis ihren Standort dem Angebot zuzufügen.
26. Zusätzlich ergänzt Hannah zu welcher Uhrzeit das Angebot bei ihr abgeholt werden kann.
27. Das fertige Angebot veröffentlicht Hannah und wartet darauf, dass jemand es reserviert.
28. Einige Stunden später sieht Hannah, dass ein Benutzer das Angebot reserviert hat und es zur angegebenen Uhrzeit bei ihr abholen kommt.
29. Hannah holt das abzugebende Gemüse aus dem Kühlschrank und verpackt es in einer Papiertüte.
30. Als der Abholer bei ihr klingelt öffnet Hannah die Tür und übergibt ihm das Lebensmittel.

Task Closure: Szenario dauert zwischen 30 Minuten und mehreren Stunden. Das liegt daran, dass es unterschiedlich lang dauern kann, bis ein Benutzer das erstellte Angebot reserviert und abholt. Der tatsächliche Aufwand für die Erstellung des Angebots und die Übergabe der Lebensmittel nimmt jedoch nur 5-10 Minuten in Anspruch.

Task: Lebensmittel bei anderen Personen abholen

User: Privatperson/Abholer

Description: Maria ist diesen Sonntag damit beschäftigt Weihnachts-Plätzchen zu backen. Während sie die Zutaten mischt fällt ihr auf, dass sie nur noch 3 Eier im Haus hat obwohl sie für den Teig 6 benötigt. Maria muss nun jemanden finden, der ihr aushelfen kann.

Task Flow:

31. Maria startet die Anwendung um ein Angebot für Eier zu finden.
32. Maria loggt sich mit ihrem Usernamen und ihrem Passwort im System ein.
33. Maria wählt die Schaltfläche „Angebote“ und wird zu einer Liste aller Angebote weitergeleitet.
34. In der Suchzeile gibt Maria den Begriff „Eier“ ein um passende Angebote zu finden.
35. Die Liste zeigt nun alle Angebote in der Nähe an, die Eier beinhalten.
36. Maria überprüft welches der Angebote die benötigte Menge von Eiern anbietet und am nächsten bei ihr liegt.
37. Maria findet ein passendes Angebot und wählt die Option „Angebot reservieren“.
38. Auf einer Karte bekommt Maria nun den genauen Standort des Angebotes angezeigt.
39. Maria sieht, dass sie das Angebot direkt abholen kommen kann.
40. Maria macht sich auf den Weg zu dem Anbieter.
41. Beim Anbieter angekommen nimmt Maria die angebotenen Eier entgegen und macht sich auf den Heimweg.

Task Closure: Das Szenario hat ungefähr eine Stunde in Anspruch genommen, wobei die Suche nach einem passenden Angebot hier höchstens 10 Minuten dauerte. Die Dauer des Szenarios kann jedoch variieren, falls der Benutzer das Angebot erst zu einem späteren Zeitpunkt bei dem Anbieter abholen kann. Außerdem kann der Weg zu dem Anbieter unterschiedlich lang dauern, je nachdem welches Transportmittel man verwendet und wie weit der Anbieter entfernt ist.

Task: Lebensmittel zum Inventar hinzufügen

User: Privatperson/Normaler Benutzer

Description: Daniel kommt von seinem Wocheneinkauf nach Hause. Er hat viele neue Lebensmittel eingekauft, die er nun gerne in das Lebensmittelinventar hinzufügen um sie besser im Überblick zu behalten.

Task Flow:

42. Daniel startet die Anwendung um die neu eingekauften Lebensmittel zum Inventar hinzuzufügen.
43. Daniel loggt sich mit seinem Usernamen und seinem Passwort im System ein.
44. Daniel öffnet das Lebensmittelinventar und wählt die Option „Lebensmittel hinzufügen“

Variante 1 - Automatisches Hinzufügen:

4. Daniel wählt die Option „Automatisch hinzufügen“ und die Kamera seines mobilen Endgerätes öffnet sich und Daniel wird dazu aufgefordert den Kassenbon einzuscannen.
5. Daniel holt den Kassenbon und hält den darauf gedruckten QR-Code vor die Kamera.
6. Das System hat den Code eingelesen und zeigt Daniel eine Liste an, welche alle Lebensmittel die erkannt wurden beinhaltet.
7. Daniel wählt die Option „Hinzufügen“.
8. Das Lebensmittelinventar zeigt nun alle gekauften Lebensmittel mit den berechneten Haltbarkeitsdaten an.

Variante 2 - Manuelles Hinzufügen:

4. Daniel wählt die Option „Manuell hinzufügen“.
5. Nun wählt Daniel ob er ein abgepacktes Lebensmittel oder Obst und Gemüse hinzufügen möchte. Daniel wählt die Option „Obst und Gemüse“.
6. Aus der Liste mit vorgeschlagenem Obste und Gemüsesorten wählt Daniel das entsprechende Lebensmittel aus.
7. Daniel spezifiziert nun das Herkunftsland sowie das Kaufdatum und die Menge des Lebensmittels.
8. Daniel wählt die Option „Hinzufügen“.
9. Daniel wiederholt Schritt 4 bis 8 mit allen Lebensmitteln die er gekauft hat.
10. Das Lebensmittelinventar zeigt nun alle gekauften Lebensmittel mit den berechneten Haltbarkeitsdaten an.

Task Closure: Szenario nimmt im Falle des automatischen Hinzufügens nur eine Minute in Anspruch. Im Falle des manuellen Hinzufügens dauert es 2-10 Minuten, je nachdem wie viele Lebensmittel dem Inventar hinzugefügt werden.

Qualitative Ziele

#04 Der Benutzer kann ein Lebensmittel schnell und einfach anbieten ohne die zuvor spezifizierten Informationen über das Lebensmittel erneut angeben zu müssen.

Priorität: 1

Hat ein Benutzer sich bereits die Mühe gemacht, das Lebensmittel in sein Inventar hinzuzufügen, stehen diese Informationen bereits zur Verfügung. Müsste ein Benutzer für jedes Lebensmittel die Informationen erneut eingeben so kann die Motivation zum Anbieten der Lebensmittel verloren gehen. Zusätzlich gilt, dass das Anbieten eines Lebensmittels möglichst einfach und mit wenigen Schritten des Benutzers erfolgen damit der Vorgang für jeden Benutzer möglichst schnell verläuft und so nicht mehr Aufwand erfordert als das Lebensmittel zu entsorgen. Zudem sollte das System den Benutzer mit entsprechenden Überschriften und Informationen zu folgenden Arbeitsschritten während der Angebotserstellung unterstützen, sodass der Benutzer auch nach Unterbrechungen schnell zur Angebotserstellung zurückkehren kann und weiß was er als Nächstes tun muss.

#05 Der Benutzer sollte rechtzeitig über das zeitnahe Ablaufen eines Lebensmittels informiert werden.

Priorität: 1

Sollte der Benutzer nicht täglich in das Lebensmittelinventar schauen oder aus anderen Gründen das Ablaufdatum eines Lebensmittels nicht rechtzeitig selbst erkennen, so sollte er zeitnah daran erinnert werden, sodass vermieden wird, dass ein Lebensmittel im Inventar in Vergessenheit gerät. Hierbei sollte der Benutzer über das Ablaufdatum des Lebensmittels sowie gegebenenfalls über mögliche Verwendungsmöglichkeiten oder das Erstellen eines Angebotes für das Lebensmittel informiert werden.

#06 Bei der Lebensmittelsuche sollten dem Benutzer nur für ihn relevante Lebensmittelangebote angezeigt werden.

Priorität: 1

Ist ein Benutzer auf der Suche nach einem möglichen Angebot in seiner Nähe, so ist es von Vorteil den Benutzern nur die Lebensmittel anzuzeigen, die seinen Einschränkungen oder möglichen Vorlieben entsprechen. Werden dem Benutzer nur Angebote angezeigt die für ihn relevant sein könnten, so ist es dem Benutzer viel schneller möglich ein passendes Angebot zu finden. Zusätzlich ist hierbei wichtig, dem Benutzer nur Angebote anzuzeigen die auch in seiner Nähe sind. Wird einem Benutzer ein Angebot angezeigt welches viel zu weit weg ist, so kann er damit wenig anfangen.

#07 Bei der Lebensmittelsuche sollte der Benutzer die Möglichkeit dazu haben nach konkreten Lebensmitteln oder Eigenschaften suchen zu können.

Priorität: 2

Ist ein Benutzer auf der Suche nach einem bestimmten Lebensmittel, so wäre es demotivierend und würde viel Zeit beanspruchen, wenn der Benutzer alle vorhandenen Angebote durchsuchen müsste, bis er das gesuchte Lebensmittel findet. Dem Benutzer sollte daher eine passende Möglichkeit geboten werden, nach konkreten Lebensmitteln suchen zu können, oder die Angebote nach bestimmten Eigenschaften wie beispielsweise „Laktosefrei“ filtern zu können.

#08 Der Benutzer sollte ein Lebensmittelangebot schnell und einfach reservieren können.

Priorität: 1

Hat ein Benutzer Interesse an einem Angebot, so sollte es ihm möglich sein, dieses schnell zu reservieren. Dauert der Prozess der Reservierung zu lang, so besteht vor allem bei unerfahrenen Benutzern die Gefahr, dass ein anderer eventuell erfahrenerer Benutzer das Lebensmittel schneller reserviert und der Aufwand der Reservierung umsonst war. Zusätzlich hierzu, sollte eine Reservierung eines Lebensmittels möglich einfach sein um die Benutzer so nicht unnötig zu verwirren und mit der Aufgabe aufzuhalten.

#09 Die Informationen über ein Lebensmittelangebot sollten für den Benutzer leicht zu finden und übersichtlich dargestellt sein.

Priorität: 1

Um die Angebote der Lebensmittel möglichst übersichtlich zu gestalten ist es wichtig, nicht zu viele Informationen auf einmal anzuzeigen. Daher sollten Dinge wie beispielsweise die Beschreibung oder das Haltbarkeitsdatum nicht direkt in der Übersicht der Lebensmittel angezeigt werden, sondern müssen von dem Benutzer bei Bedarf manuell angezeigt lassen werden.

Dem Benutzer sollte hierbei möglichst klar mitgeteilt werden, wo er die weiteren Informationen vorfinden kann und was er dafür tun muss.

#10 Der Benutzer sollte die Möglichkeit haben, mit anderen Benutzern die ein Lebensmittelangebot reserviert haben, in Kontakt zu treten.

Priorität: 2

Bietet ein Benutzer ein Lebensmittel an, so kann es beispielsweise vorkommen, dass dieses nur zu gewissen Zeiten abgeholt werden kann. Daher ist es wichtig, dass Abholer und Anbieter untereinander kommunizieren können um gegebenenfalls wichtige Informationen auszutauschen.

#11 Der Benutzer sollte, wenn er ein Angebot abholen möchte, den Standort des Anbieters möglichst anschaulich auf einer Karte angezeigt bekommen.

Priorität: 2

Möchte ein Benutzer ein Angebot abholen, so ist es natürlich wichtig, diesem den jeweiligen Abholort konkret mitzuteilen. Dies sollte möglichst anhand von einer Karte geschehen, damit der Weg zum Standort des Anbieters möglichst anschaulich ist, und es auch Benutzern die sich in der Gegend nicht sehr gut auskennen trotzdem möglich ist, den Abholort zu finden.

#12 Der Benutzer sollte dazu in der Lage sein, sein Lebensmittelinventar ohne große Mühe aktualisieren zu können.

Priorität: 1

Hat ein Benutzer ein Lebensmittel (teilweise) verbraucht oder an einen anderen Benutzer abgegeben, so ist es von Nöten, das Lebensmittelinventar anzupassen. Dies sollte möglichst einfach erfolgen, da Benutzer sonst keine Motivation dazu haben, das Inventar langfristig weiterzuführen. Daher sollte das Anpassen der Einträge möglichst einfach und schnell erfolgen.

#13 Bei der Suche nach Verwendungsmöglichkeiten sollten dem Benutzer nur jene Möglichkeiten angezeigt werden, die seinen Vorlieben und Einschränkungen entsprechen.

Priorität: 1 Ist ein Benutzer auf der Suche nach Verwendungsmöglichkeiten für vorhandene Lebensmittel, so sollten ihm, wenn möglich nur jene Möglichkeiten angezeigt werden, welche für ihn relevant sind. Vor allem bei Verwendungsmöglichkeiten ist dies von Nöten, da der Benutzer sonst

eine große Menge Vorschlägen durchsuchen muss und es daher sehr viel Zeit in Anspruch nehmen kann etwas Passendes zu finden. Dies kann in Fällen dann demotivierend für den Benutzer sein, da er sich unnötig große Mühe machen muss.

#14 Bei vorgeschlagenen Verwendungsmöglichkeiten sollten möglichst viele der bereits vorhandenen Lebensmittel berücksichtigt werden.

Priorität: 2

Sucht ein Benutzer nach einer Verwendungsmöglichkeit, so wäre es von Vorteil, wenn möglichst viele der bereits vorhandenen Lebensmittel berücksichtigt werden. Ist dies nicht der Fall, so findet der Benutzer nur für ein vorhandenes Lebensmittel beispielsweise ein Rezept und muss die restlichen Zutaten für die Zubereitung zusätzlich einkaufen. So kann ein Kreislauf entstehen. Es wird nur ein Lebensmittel verbraucht aber 6 neue nachgekauft, wodurch gegebenenfalls andere Lebensmittel nicht verwendet werden können.

#15 Nach der Übergabe eines Lebensmittels sollte das Lebensmittel automatisch bei dem Anbieter aus dem Inventar entfernt sowie bei dem Abholer in das Inventar hinzugefügt werden.

Priorität: 2

Hat ein Benutzer ein Lebensmittel an jemand anderen abgegeben so kann es beispielsweise dazu kommen, dass er vergisst das Lebensmittel aus dem Lebensmittelinventar zu entfernen. Zusätzlich müsste sich ein Benutzer die Mühe machen ein Lebensmittel manuell in seinem System einzutragen. Daher wäre es von Vorteil, wenn die angebotenen Lebensmittel die auch abgeholt wurden direkt von dem Inventar des Anbieters in das des Abholers geschoben werden, um so beiden Benutzern die Mühe zu ersparen.

Concrete Use Cases

Tabelle 20: Concrete Use Case: Lebensmittel verwalten

User Intention	System Responsibility
Benutzer wählt die Funktionalität der Eintragung für Lebensmittel aus.	Das System visualisiert die Lebensmittel des Inventars und die Optionen weitere Lebensmittel manuell oder automatisch einzutragen.
Benutzer interagiert mit dem System und wählt die automatische Variante aus.	Das System bietet dem Benutzer eine automatische Möglichkeit seinen Einkaufszettel einlesen zu lassen.
Benutzer interagiert mit dem System und richtet dieses auf den Einkaufszettel.	Das System liest die Informationen aus dem Einkaufszettel aus und berechnet die Haltbarkeitsdaten und die optimale Lagerung für die frischen Lebensmittel und visualisiert die Ergebnisse dem Benutzer.

Tabelle 21: Concrete Use Case: Lebensmittel von anderen Personen beziehen

User Intention	System Responsibility
Benutzer wählt die Funktionalität der Angebots Suche aus.	Das System visualisiert die Angebote der anderen Benutzer und die Optionen, für die Reihenfolge und Auswahl der Anzeige.
Benutzer interagiert mit dem System und wählt die Fußgänger Option aus.	Das System registriert die Eingabe und passt die Auswahl der Angebote entsprechend an.

Benutzer interagiert mit dem System und wählt die Option aus die Angebote nach ihrer Relevanz zu sortieren.	Das System registriert die Eingabe und sortiert die Angebote nach den Bedürfnissen des Benutzers.
Benutzer interagiert mit dem System und wählt ein Angebot aus.	Das System registriert die Eingabe und visualisiert dem Benutzer weitere Informationen zu dem Angebot. Weiterhin wird die Option das Angebot wahrzunehmen visualisiert.
Benutzer interagiert mit dem System und wählt die Option aus, das Angebot zu reservieren.	Das System registriert die Eingabe und erstellt eine Reservierung für den Benutzer. Die erstellte Reservierung wird dem Benutzer angezeigt.

Tabelle 22: Concrete Use Case: Lebensmittel für andere Personen anbieten

User Intention	System Responsibility
Benutzer wählt die Funktionalität der Angebotserstellung aus.	Das System visualisiert die Eigenschaften mit den Eingabemöglichkeiten für das Angebot.
Benutzer interagiert mit dem System und schreibt Informationen in die visualisierten Felder.	Das System visualisiert die Eingabe für den Benutzer.
Benutzer wählt die Option ein Lebensmittel aus seinem Inventar auszuwählen.	Das System visualisiert die Lebensmittel, welche sich in dem Inventar von dem Benutzer befinden.
Benutzer wählt die ein Lebensmittel aus.	Das System fügt die Eigenschaften des Lebensmittels automatisch hinzu und visualisiert das Ergebnis dem Benutzer.
Benutzer wählt die Option Angebot erstellen aus.	Das System informiert den Benutzer über die erfolgreiche Erstellung des Angebots. Weiterhin wird das erstellte Angebot seiner Liste hinzugefügt und dem Benutzer visualisiert.

Evaluation

#1 Use Case - Lebensmittel zum Inventar hinzufügen

- 1. Action: Der Benutzer loggt sich im System ein und sucht in der Navigation nach dem Lebensmittelinventar.
 Response: Die Bottom Navigation beinhaltet einen Reiter mit der Beschriftung „Lebensmittelinventar“. Bei dem Klicken auf diesen wird man zu dem Lebensmittelinventar weitergeleitet.
 Heuristiken: Es wurden keine Heuristiken verletzt.
- 2. Action: Der Benutzer wählt den Button zum Hinzufügen von neuen Lebensmitteleinträgen aus.
 Response: Ein Dialogfenster öffnet sich. Dieses fordert den Benutzer dazu auf zwischen automatischem oder manuellem Hinzufügen von Einträgen zu wählen. Außerdem ist ein Button zum Abbrechen vorhanden.

Heuristiken: Es wurden keine Heuristiken verletzt.

- 3. Action: Der Benutzer wählt den Button zum automatischen Hinzufügen von frischen Lebensmitteln aus.
 Response: Das System öffnet die im System integrierte Kamera und eine Aufforderung dazu, den QR Code vorzuhalten. Außerdem ist ein Button zum Abbrechen vorhanden.
 Heuristiken: Es wurden keine Heuristiken verletzt.
- 4. Action: Der Benutzer hält Kamera auf den QR-Code, welcher sich auf dem Kassenbeleg befindet.
 Response: Das System bestätigt dem Benutzer anhand von einer Message Box, dass die Daten eingelesen wurden. Der erstellte Eintrag ist unter dem Reiter „Lebensmittelinventar“ für den Benutzer abrufbar.
 Heuristiken: Es wurden keine Heuristiken verletzt.

Tabelle 24: Cognitive Walkthrough Use Case #1

Action	Will users try to achieve the right effect?	Will users be able to notice that a correct action is available?	Will the user associate the correct action with the effect trying to be achieved?	If the correct action is performed, will the user see that progress is being made towards the solution of the task?
1	Yes	Yes	Yes	Yes
2	Yes	Yes	Yes	Yes
3	Yes	Yes	Yes	Yes
4	Yes	Yes	Yes	Yes

#2 Use Case - Lebensmittel anderen Personen anbieten

- 1. Action: Der Benutzer loggt sich im System ein und sucht in der Navigation nach den Angeboten.
 Response: Die Bottom Navigation beinhaltet einen Reiter mit der Beschriftung „Angebote“. Bei dem Klicken auf diesen wird man zu den Angeboten weitergeleitet.
 Heuristiken: Es wurden keine Heuristiken verletzt.
- 2. Action: Der Benutzer wählt den Button zur Angebotserstellung aus.
 Response: Es wird ein neuer Screen mit der Überschrift „Angebot erstellen“ geöffnet, welcher dem Benutzer alle zu spezifizierenden Eigenschaften anzeigt. Außerdem wird ein Button zum Abbrechen angezeigt.
 Heuristiken: Es werden keine Heuristiken verletzt.
- 3. Action: Der Benutzer wählt das abzugebende Lebensmittel aus seinem Inventar aus.
 Response: Das Lebensmittel sowie alle bereits bekannten Informationen werden als Eigenschaften in dem Erstellungs-Screen eingefügt.
 Heuristiken: Es wurden keine Heuristiken verletzt.
- 4. Action: Der Benutzer füllt die restlichen Felder für das Angebot aus
 Response: Das System präsentiert die eingegebenen Daten innerhalb der Eigenschafts-

Felder.

Heuristiken: Es werden keine Heuristiken verletzt.

- 5. Action: Der Benutzer wählt den Button zum Angebot erstellen aus.
Response: Das System bestätigt dem Benutzer anhand von einer Message Box, dass das Angebot erfolgreich erstellt wurde. Das erstellte Angebot ist unter dem Reiter „Angebote“ abrufbar.
Heuristiken: Es wurden keine Heuristiken verletzt.

Tabelle 25: Cognitive Walkthrough Use Case #2

Action	Will users try to achieve the right effect?	Will users be able to notice that a correct action is available?	Will the user associate the correct action with the effect trying to be achieved?	If the correct action is performed, will the user see that progress is being made towards the solution of the task?
1	Yes	Yes	Yes	Yes
2	Yes	Yes	Yes	Yes
3	Yes	Maybe (Yes)	Maybe (Yes)	Yes
4	Yes	Yes	Yes	Yes
5	Yes	Yes	Yes	Yes

#3 Use Case - Lebensmittel von anderen Personen beziehen

- 1. Action: Der Benutzer loggt sich im System ein und sucht in der Navigation nach den Angeboten.
Response: Die Bottom Navigation beinhaltet einen Reiter mit der Beschriftung „Angebote“. Bei dem Klicken auf diesen wird man zu den Angeboten weitergeleitet.
Heuristiken: Es wurden keine Heuristiken verletzt.
- 2. Action: Der Benutzer wählt die Einstellung für die Fußgänger aus.
Response: Das System bestätigt dem Benutzer die festgelegte Mobilität passt die Abholwerte der angezeigten Angebote an die festgelegte Mobilität an.
Heuristiken: Es wurden keine Heuristiken verletzt.
- 3. Action: Der Benutzer wählt die Option aus, die Angebote nach ihrer Relevanz zu sortieren.
Response: Das System bestätigt die Auswahl und verändert die Reihenfolge der angezeigten Angebote so, dass die angebotene mit den hohen Abholwerten zuerst angezeigt werden.
Heuristiken: Es wurden keine Heuristiken verletzt.
- 4. Action: Der Benutzer wählt ein konkretes Angebot aus.
Response: Es öffnet sich ein neuer Screen mit der detaillierten Präsentation des ausgewählten Angebotes. Außerdem wird ein Button zum zurück kehren zu den Angeboten angezeigt.
Heuristiken: Es wurden keine Heuristiken verletzt.
- 5. Action: Der Benutzer wählt den Button aus, um das Angebot zu reservieren.
Response: Das System bestätigt dem Benutzer anhand von einer Message Box, dass das

Angebot erfolgreich reserviert wurde. Das reservierte Angebot ist unter dem Reiter „Reservierungen“ abrufbar.

Heuristiken: Es wurden keine Heuristiken verletzt.

Tabelle 26: Cognitive Walkthrough Use Case #3

Action	Will users try to achieve the right effect?	Will users be able to notice that a correct action is available?	Will the user associate the correct action with the effect trying to be achieved?	If the correct action is performed, will the user see that progress is being made towards the solution of the task?
1	Yes	Yes	Yes	Yes
2	Yes	Maybe (Yes)	Yes	Yes
3	Yes	Maybe (Yes)	Yes	Yes
4	Yes	Yes	Yes	Yes
5	Yes	Yes	Yes	Yes

Think Aloud

Die Think Aloud Methode wurde mit drei verschiedenen Testpersonen durchgeführt. Sie bekamen die konkreten Aufgaben „Lebensmittel anbieten“ und „Lebensmittel finden/reservieren“. Außerdem wurden den Benutzern die Screens gezeigt und die Benutzer sollten die Screens so erklären wie sie sie verstehen und erklären, was eventuell unklar für die erscheint. Folgende Probleme am User Interface sind hierbei gefunden worden:

Tabelle 27: Think Aloud Ergebnisse

Nr	Kurzbeschreibung	Ausführliche Beschreibung	Priorität	Aufwand
1	Switch für die Lagerung wurde nicht erkannt.	Der Switch der die Lagerung innerhalb oder außerhalb des Kühlschranks festlegt wurde nicht erkannt. Den Benutzern fehlte ein visueller Hinweis darauf, was der Switch bedeutet, ohne diesen hatten die Benutzer keine Vorstellung darüber, was die Funktion des Switches ist.	1	3
2	Unklar wo „Mobilität festlegen“ zu finden ist.	Die Benutzer hatten vermutet, dass die Mobilität innerhalb der Account-Einstellungen festgelegt werden muss und nicht innerhalb der Angebotssuche manuell angepasst werden muss.	1	2
3	Unklar wie man in der Angebotserstellung ein Lebensmittel aus dem Inventar aussuchen kann.	Die Bezeichnung der Auswahlmöglichkeit für Lebensmittel aus dem Inventar war etwas unklar, da diese nur „Lebensmittel“ lautet. Besser hätten die Benutzer gefunden, wenn die Option klarer benannt gewesen wäre.	2	1

4	Der Abholwert der Angebote wurde missverstanden	Der Angegebene Abholwert wurde teilweise als Menge der Angebote dieses Typs oder Angebotene Menge interpretiert und erst nach weiterem Nachdenken als der Abholwert. Die Benutzer wünschten sich eventuell eine klarere Kennzeichnung des Abholwerts.	1	3
---	---	---	---	---

REST Tabelle

Tabelle 40: REST Modellierung

Ressourcen	Verb	Semantik	Type (req)	Type (res)	Status-codes
/User	GET	Gibt eine Liste aller Benutzer zurück.	-	Application/JSON	200
/User/:uid	GET	Gibt Information eines Benutzers zurück.	-	Application/JSON	200,404
/User/:uid	PUT	Verändert einen Benutzer.	Application/JSON	Application/JSON	200,404,400
/User	POST	Erstellt einen neuen Benutzer.	Application/JSON	Application/JSON	201,400
/User/:uid	DELETE	Löscht einen Benutzer.	-	Application/JSON	200,404
/User/:uid/Einschränkungen	GET	Gibt alle Einschränkungen eines Benutzers zurück.	-	Application/JSON	200,404
/User/:uid/Einschränkungen/:id	PUT	Verändert eine Einschränkung eines Benutzers.	Application/JSON	Application/JSON	200,404,400
/User/:uid/Einschränkungen	POST	Erstellt eine neue Einschränkung für einen Benutzer.	Application/JSON	Application/JSON	201,400,404
/User/:uid/Einschränkungen/:id	DELETE	Löscht eine Einschränkung eines Benutzers.	Application/JSON	Application/JSON	200,404
/User/:uid/Einträge	GET	Gibt alle Einschränkungen eines Benutzers zurück.	-	Application/JSON	200,404
/User/:uid/Einträge	POST	Erstellt einen neuen Eintrag für einen Benutzer.	Application/JSON	Application/JSON	201,400,404
/User/:uid/Nachrichten	GET	Gibt alle Nachrichten	-	Application/JSON	200,404

		eines Benutzers zurück.			
/User/:uid/Nachrichten?sender=value	GET	Gibt Nachrichtenverlauf mit einem konkreten Sender zurück.	-	Application /JSON	200,404
/User/:uid/Nachrichten	POST	Erstellt eine neue Nachricht für einen Benutzer.	Application /JSON	Application /JSON	201,400,404
/User/:uid/Nachrichten/:id	DELETE	Löscht eine Nachricht eines Benutzers	-	Application /JSON	200,404
/User/:uid/Token	GET	Gibt den Token eines Benutzers zurück.	-	Application /JSON	200,404
/User/:uid/Token	POST	Erstellt einen Token für einen Benutzer.	Application /JSON	Application /JSON	201,400,404
/User/:uid/Token	DELETE	Löscht den Token eines Benutzers.	-	Application /JSON	200,404
/Angebote	GET	Gibt alle erstellten Angebote zurück.	-	Application /JSON	200
/Angebote/:id	GET	Gibt ein konkretes Angebot zurück.	-	Application /JSON	200,404
/Angebote/:id	PUT	Verändert ein erstelltes Angebot.	Application /JSON	Application /JSON	200,400,404
/Angebote	POST	Erstellt ein neues Angebot.	Application /JSON	Application /JSON	201,400
/Angebote/:id	DELETE	Löscht ein Angebot eines Benutzers	-	Application /JSON	200,404
/Angebote?uid=value	GET	Filtert Angebote eines Benutzers heraus.	-	Application /JSON	200
/Angebote/User/:uid/:latitude/:longitude/:mobility	GET	Gibt Angebote für einen Benutzer zurück, mit errechneten Abholwert.	-	Application /JSON	200,400,404
/Angebote/User//:uid/:latitude/:longitude/:mobility?sortBy=Abholwert	GET	Gibt Angebote für einen Benutzer zurück mit errechneten Abholwert und sortiert nach diesem.	-	Application /JSON	200,400,404
/Reservierungen	GET	Gibt alle erstellten	-	Application	200

		Reservierungen zurück.		/JSON	
/Reservierungen/:id	GET	Gibt eine konkrete Reservierung zurück.	-	Applicati on /JSON	200,404
/Reservierungen?uid=value	GET	Gibt alle getätigten Reservierungen eines Benutzers zurück.	-	Applicati on /JSON	200
/Reservierungen?eid=value	GET	Gibt alle erhaltenen Reservierungen eines Benutzers zurück.	-	Applicati on /JSON	200
/Reservierungen	POST	Erstellt eine neue Reservierung.	Applica tion /JSON	Applicati on /JSON	201,400
/Reservierungen/:id	DELETE	Löscht eine konkrete Reservierung.	-	Applicati on /JSON	200,404

Statuscodes und Spezifikation der Fehlerbehandlung

Tabelle 41: Statuscodes und Fehlerbehandlung

Ressource	HTTP-Verb	Statuscodes	Fehlerbehandlung
/User	GET	200 OK	-
/User/:uid	GET	200 OK	-
/User/:uid	GET	404 Not Found	Antwort mit dem Hinweis, dass die Ressource nicht existiert wird versendet. Es wird empfohlen zunächst eine Ressource zu erstellen oder die uid zu überprüfen.
/User/:uid	PUT	200 OK	-
/User/:uid	PUT	404 Not Found	Antwort mit dem Hinweis, dass die Ressource nicht existiert wird versendet. Es wird empfohlen zunächst eine Ressource zu erstellen oder die uid zu überprüfen.
/User/:uid	PUT	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/User/:uid	DELETE	200 OK	-
/User/:uid	DELETE	404 Not Found	Antwort mit dem Hinweis das die Ressource nicht existiert wird versendet. Falls Ressource nicht abrufbar ist sind keine weiteren Schritte erforderlich.
/User	POST	201 Created	-
/User	POST	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/User/:uid/Einschränkungen	GET	200 OK	-

/User/:uid/Einschränkungen	GET	404 Not Found	Antwort mit dem Hinweis das der Benutzer nicht existiert. Es wird empfohlen die Uid zu überprüfen oder zunächst einen neuen User zu erstellen.
/User/:uid/Einschränkungen/:id	GET	200 OK	-
/User/:uid/Einschränkungen/:id	GET	404 Not Found	Antwort mit dem Hinweis, dass die User oder Einschränkung Ressource nicht existiert wird versendet. Es wird empfohlen zunächst eine entsprechende Ressource zu erstellen oder die Eingabe zu überprüfen.
/User/:uid/Einschränkungen/:id	PUT	200 OK	-
/User/:uid/Einschränkungen/:id	PUT	404 Not Found	Antwort mit dem Hinweis, dass die User oder Einschränkung Ressource nicht existiert wird versendet. Es wird empfohlen zunächst eine entsprechende Ressource zu erstellen oder die Eingabe zu überprüfen.
/User/:uid/Einschränkungen/:id	PUT	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/User/:uid/Einschränkungen	POST	201 Created	-
User/:uid/Einschränkungen	POST	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
User/:uid/Einschränkungen	POST	404 Not Found	Antwort mit dem Hinweis das der Benutzer nicht existiert. Es wird empfohlen die Uid zu überprüfen oder zunächst einen neuen User zu erstellen.
/User/:uid/Einschränkungen/:id	DELETE	200 OK	-
/User/:uid/Einschränkungen/:id	DELETE	404 Not Found	Antwort mit dem Hinweis das die Ressource nicht existiert wird versendet. Falls Ressource nicht abrufbar ist sind keine weiteren Schritte erforderlich.
/User/:uid/Einträge	GET	200 OK	-
/User/:uid/Einträge	GET	404 Not Found	Antwort mit dem Hinweis das der Benutzer nicht existiert. Es wird empfohlen die Uid zu überprüfen oder zunächst einen neuen User zu erstellen.
/User/:uid/Einträge	POST	201 Created	-
/User/:uid/Einträge	POST	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/User/:uid/Einträge	POST	404 Not Found	Antwort mit dem Hinweis das der Benutzer nicht existiert. Es wird empfohlen die Uid zu überprüfen oder zunächst einen neuen User zu erstellen.
/User/:uid/Einträge/:id	DELETE	405 Method Not Allowed	Antwort mit dem Hinweis, welche HTTP-Verben für diese Ressource erlaubt sind.

/User/:uid/Einträge/:id	PUT	405 Method Not Allowed	Antwort mit dem Hinweis, welche Verben für diese Ressource erlaubt sind.
/User/:uid/Nachrichten	GET	200 OK	-
/User/:uid/Nachrichten ?sender=value	GET	200 OK	-
/User/:uid/Nachrichten ?sender=value	GET	404 Not Found	Antwort mit dem Hinweis, dass die User Ressource nicht existiert wird versendet. Es wird empfohlen zunächst eine Ressource zu erstellen oder die uid zu überprüfen.
/User/:uid/Nachrichten	GET	404 Not Found	Antwort mit dem Hinweis, dass die User Ressource nicht existiert wird versendet. Es wird empfohlen zunächst eine Ressource zu erstellen oder die uid zu überprüfen.
/User/:uid/Nachrichten	POST	201 Created	-
/User/:uid/Nachrichten	POST	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/User/:uid/Nachrichten	POST	404 Not Found	Antwort mit dem Hinweis, dass die User Ressource nicht existiert wird versendet. Es wird empfohlen zunächst eine Ressource zu erstellen oder die uid zu überprüfen.
/User/:uid/Nachrichten/:id	DELETE	200 OK	-
/User/:uid/Nachrichten/:id	DELETE	404 Not Found	Antwort mit dem Hinweis, dass der Benutzer- oder die Nachrichten Ressource nicht existiert wird versendet. Es wird empfohlen die Uid oder id zu überprüfen. Falls die Nachrichten Ressource nicht abrufbar ist sind keine weiteren Schritte erforderlich. Weiterhin folgt ein Link zu den entsprechenden Ressourcen.
/User/:uid/Token	GET	200 OK	-
/User/:uid/Token	GET	404 Not Found	Antwort mit dem Hinweis das der Benutzer nicht existiert. Es wird empfohlen die Uid zu überprüfen oder einen neuen Benutzer zu erstellen.
/User/:uid/Token	POST	201 Created	-
/User/:uid/Token	POST	404 Not Found	Antwort mit dem Hinweis das der Benutzer nicht existiert. Es wird empfohlen die Uid zu überprüfen oder einen neuen Benutzer zu erstellen.
/User/:uid/Token	POST	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/User/:uid/Token	DELETE	200 OK	-
/User/:uid/Token	DELETE	404 Not Found	Antwort mit dem Hinweis das der Benutzer nicht existiert. Es wird empfohlen die Uid zu überprüfen oder einen neuen Benutzer zu erstellen.

/Angebote	GET	200 OK	-
/Angebote/:id	GET	200 OK	-
/Angebote/:id	GET	404 Not Found	Antwort mit dem Hinweis, dass das Angebot nicht existiert. Es wird empfohlen die id zu überprüfen oder ein neues Angebot zu erstellen.
/Angebote/:id	PUT	200 OK	-
/Angebote/:id	PUT	404 Not Found	Antwort mit dem Hinweis, dass das Angebot nicht existiert. Es wird empfohlen die id zu überprüfen oder ein neues Angebot zu erstellen.
/Angebote/:id	PUT	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/Angebote	POST	201 Created	-
/Angebote	POST	400 Bad Request	Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/Angebote/:id	DELETE	200 OK	-
/Angebote/:id	DELETE	404 Not Found	Antwort mit dem Hinweis, dass die Ressource nicht existiert wird versendet. Falls Ressource nicht abrufbar ist sind keine weiteren Schritte erforderlich.
/Angebote?uid=value	GET	200 OK	-
/Angebote/User/:uid/:latitude/:longitude/:mobility	GET	200 OK	-
/Angebote/User/:uid/:latitude/:longitude/:mobility	GET	400 Bad Request	Antwort mit dem Hinweis, welche Parameter im Body fehlerhaft übertragen wurden.
/Angebote/User/:uid/:latitude/:longitude/:mobility	GET	404 Ressource Not Found	Antwort mit dem Hinweis der Benutzer nicht gefunden werden kann. Es wird empfohlen die uid zu überprüfen oder einen neuen Benutzer zu erstellen.
/Angebote/User/:uid/:latitude/:longitude/:mobility?sortBy=Abholwert	GET	200 OK	-
/Angebote/User/:uid/:latitude/:longitude/:mobility?sortBy=Abholwert	GET	400 Bad Request	Antwort mit dem Hinweis, welche Parameter fehlerhaft übertragen wurden.
/Angebote/User/:uid/:latitude/:longitude/:mobility?sortBy=Abholwert	GET	404 Ressource Not Found	Antwort mit dem Hinweis der Benutzer nicht gefunden werden kann. Es wird empfohlen die uid zu überprüfen oder einen neuen Benutzer zu erstellen.
/Reservierungen	GET	200 OK	-
/Reservierungen/:id	GET	200 OK	-
/Reservierungen/:id	GET	404 Not Found	Antwort mit dem Hinweis, dass die Reservierung nicht gefunden werden kann. Es wird empfohlen die uid zu überprüfen oder eine neue Reservierung zu erstellen.
/Reservierungen?uid=value	GET	200 OK	-
/Reservierungen?eid=value	GET	200 OK	-
/Reservierungen	POST	201 Created	-

/Reservierungen	400 Bad Request		Antwort mit dem Hinweis, welche Elemente im Body fehlerhaft übertragen wurden.
/Reservierungen/:id	DELETE	200 OK	-
/Reservierungen/:id	DELETE	404 Not Found	Antwort mit dem Hinweis, dass die Ressource nicht existiert wird versendet. Falls Ressource nicht abrufbar ist sind keine weiteren Schritte erforderlich.

Zu jeder Fehlerausgabe wird ebenfalls ein Link zur Dokumentation mitgesendet und auf diese verwiesen.

Iteration Stakeholderanalyse

Im Folgenden wird eine Stakeholder Analyse durchgeführt, um zu ermitteln, welche Personen oder Organisationen ein Anrecht, Interesse, Anspruch oder Anrecht an dem zu entwickelndem System haben. Bei der Iteration wurde eine angemessenere Vorlage verwendet und Lösungsansätze möglichst vermieden.

Tabelle 9: Iteration Stakeholderanalyse

Bezeichnung	Bezug zum System	Objektbereich	Erfordernis/ Erwartung
Privatperson	Anspruch	Verwaltung der eingekauften Lebensmittel.	<ul style="list-style-type: none"> • Eine Privatperson muss Lebensmittel verfügbar haben, um die Verwaltung tun zu können. • Eine Privatperson muss Wissen Informationen über die Lebensmittel verfügbar haben, um die optimale Verwaltung tun zu können. • Eine Privatperson erwartet einen besseren Überblick über seine Lebensmittel.
Privatperson	Anspruch	Suche von geeigneten Rezepten für die Verwendung von Lebensmitteln.	<ul style="list-style-type: none"> • Eine Privatperson muss Lebensmittel verfügbar haben, um die Verwendung

			<p>entscheiden zu können.</p> <ul style="list-style-type: none"> • Eine Privatperson muss Wissen über den Zustand der Lebensmittel haben, um deren Verwendungstauglichkeit entscheiden zu können. • Eine Privatperson muss über ihre gesundheitlichen Einschränkungen bescheid wissen, um geeignete Rezepte auswählen zu können. • Eine Privatperson erwartet geeignete Rezepte zu finden.
Privatperson	Interesse	Ermittlung von Haltbarkeitsdaten von frischen Lebensmitteln.	<ul style="list-style-type: none"> • Eine Privatperson muss Lebensmittel verfügbar haben, um Ermittlungen über den Zustand tun zu können. • Eine Privatperson muss Informationen über Lebensmittel wissen, um die Haltbarkeit einschätzen zu können. • Eine Privatperson erwartet möglichst genaue Ergebnisse.
Privatperson	Anrecht	Sichere Verwaltung von persönlichen Daten.	<ul style="list-style-type: none"> • Eine Privatperson muss persönliche Daten verfügbar haben, um die Angabe tun zu können. • Eine Privatperson erwartet einen verantwortungsvol

			len und sicheren Umgang mit Daten.
Privatperson	Anspruch	Lebensmittel an andere Menschen weiterzugeben.	<ul style="list-style-type: none"> • Eine Privatperson muss Lebensmittel verfügbar haben, um diese weitergeben zu können. • Eine Privatperson muss Informationen über Lebensmittel, Abgabeort, Uhrzeit wissen, um Lebensmittel abgeben zu können. • Eine Privatperson muss ggf. Informationen über anderen Menschen verfügbar haben, um diesen bei der Weitergabe identifizieren zu können. • Eine Privatperson erwartet einen schnellen und sicheren Austausch.
Privatperson	Anspruch	Lebensmittelangebote in seiner Nähe in Anspruch nehmen zu können.	<ul style="list-style-type: none"> • Eine Privatperson muss Informationen über Angebote und Standort verfügbar haben, um eine geeignete Auswahl entscheiden zu können. • Eine Privatperson erwartet eine Vielzahl an vorhandenen Angeboten.

Müllabfuhr	Interesse	Lebensmittel an andere Menschen weiter zu geben.	<ul style="list-style-type: none"> Die Müllabfuhr erwartet eine Reduktion der Abfallmengen.
Supermarkt	Anteil	Ermittlung von Haltbarkeitsdaten von frischen Lebensmitteln.	<ul style="list-style-type: none"> Der Supermarkt muss Informationen über frische Lebensmittel wissen, um Haltbarkeitsinformationen weiter geben zu können.
BMEL (Bundesministerium für Ernährung und Landwirtschaft)	Interesse	Lebensmittel an andere Menschen weiterzugeben.	<ul style="list-style-type: none"> Das BMEL hat die erwartet eine Reduktion der aktuellen Lebensmittelabfälle.

Iteration Proof of Concept

Die Proof of Concepts haben eine Iteration durchlaufen, bei der die Formulierungen angepasst wurden, da sowohl Exit als auch Fail Bedingungen nicht messbar waren. Ebenfalls wurde eine Priorisierung durchgeführt, da einige PoCs konkreter auf projektspezifische Risiken eingehen, während die anderen Risiken durch mangelnde Erfahrung in der Implementierung entstehen. Die PoCs werden im Folgenden in absteigender Priorität aufgeführt. Besonders die ersten beiden PoCs werden als wichtig angesehen.

Erstellten Proofs of Concept

Tabelle 31: Begriffsglossar für die Proofs of Concept

Beschreibung	Beschreibung des zu prüfenden Konzepts.
Exit	Bedingung für das Erreichen des Erfolgsfalles.
Fail	Bedingung für das Scheitern.
Fallback	Alternative auf die zurückgegriffen werden kann.
PoC	Abkürzung für Proof of Concept.

Tabelle 32: PoC Nr. 1 – Berechnung des Haltbarkeitsdatums für frische Lebensmittel

Beschreibung	Für frische Lebensmittel, wobei vorrangig für Obst & Gemüse, soll ein Haltbarkeitsdatum mit Hilfe von unterschiedlichen Faktoren geschätzt werden.
Exit	Nach Maximal 25 Stunden Entwicklungszeit ist es Möglich das Haltbarkeitsdatum für Apfel, Banane und Blaubeeren berechnen (Schätzung) zu können.
Fail	Nach Maximal 25 Stunden Entwicklungszeit ist es nicht Möglich Haltbarkeitsdaten zu berechnen. Die

	Daten über oder unterschreiten die erwarteten Ergebnisse oder weichen anderweitig ab.
Fallback	Der Benutzer muss selbständig aus eigenem Erachten eine Abschätzung erstellen.
Ergebnis	Das Haltbarkeitsdatum konnte erfolgreich ermittelt werden.

Tabelle 33: PoC Nr. 2 – Berechnung des Abholwerts für angebotene Lebensmittel

Beschreibung	Für Lebensmittel soll ein Abholwert berechnet werden, welcher anhand von Personendaten und externen Faktoren ermittelt wird.
Exit	Nach 20 Stunden Entwicklung ist es möglich für eine Anzahl von 3 erstellten Lebensmittelangeboten, jeweils einen Abholwert zu berechnen, welcher den erwarteten Ergebnissen entspricht.
Fail	Nach 20 Stunden Entwicklung ist es nicht möglich für eine Anzahl von 3 erstellten Lebensmittelangeboten, jeweils einen Abholwert zu berechnen, welcher den erwarteten Ergebnissen entspricht. Die Werte weichen von dem Erwarteten Wert ab.
Fallback	Die Faktoren für die Berechnung des Abholfaktors müssen separat angezeigt werden.

Tabelle 34: PoC Nr. 3 – Verschlüsselte Datenübertragung

Beschreibung	Für die Datensicherheit der Benutzer sollen diese mit Hilfe des HTTPS Protokolls verschlüsselt übertragen werden.
Exit	Nach Maximal 3 Stunden Entwicklungszeit ist es möglich einen Datensatz erfolgreich mit Hilfe des HTTPS Protokolls zwischen Client und Server zu übertragen.
Fail	Nach Maximal 3 Stunden Entwicklungszeit ist es nicht möglich Daten mit Hilfe des HTTPS Protokolls zwischen Client und Server zu übertragen oder Daten können nicht erfolgreich entschlüsselt werden und sind somit unbrauchbar.
Fallback	Daten des Benutzers müssen unverschlüsselt versendet werden.

Tabelle 35: PoC Nr. 4 – Standort Ermittlung durch Sensorik

Beschreibung	Der Standort (Längen und Breitengrad) des Benutzers soll durch die eingebaute Sensorik im mobilen Endgerät ermittelt werden.
---------------------	--

Exit	Nach 1 Stunde Entwicklungszeit ist es möglich durch eingebaute Sensorik im mobilen Endgerät den aktuellen Standort (Längen und Breitengrade) des Benutzers korrekt zu ermitteln.
Fail	Nach 1 Stunde Entwicklungszeit ist es nicht möglich durch eingebaute Sensorik im mobilen Endgerät den aktuellen Standort (Längen und Breitengrade) des Benutzers korrekt zu ermitteln. Ist die Sensorik fehlerhaft oder defekt so ist dies ebenfalls ein Scheitern.
Fallback	Der Benutzer muss seinen Standort manuell eingeben.
Ergebnis	Es konnte erfolgreich der Standort ermittelt werden.

Tabelle 36: PoC Nr. 5 – Berechnung der Entfernung zwischen zwei Punkten

Beschreibung	Die Entfernung (Luftlinie) zwischen zwei verschiedenen Standorten, welche durch Längen und Breitengrad beschrieben sind, soll berechnet angenähert werden.
Exit	Nach 1 Stunde Entwicklungszeit ist es Möglich die Entfernung zwischen den Standorten Düsseldorf und Stuttgart wurde erfolgreich auf Basis der Längen und Breitengrade zu berechnen.
Fail	Nach 1 Stunde Entwicklungszeit ist es nicht möglich, das korrekte Ergebnis (322km) automatisch zu ermitteln.
Fallback	Die Entfernung muss mit Hilfe eines Dienstes ermittelt werden.
Ergebnis	Die Entfernung konnte erfolgreich ermittelt werden.

Tabelle 37: PoC Nr. 6 - QR – Code auslesen

Beschreibung	Ein QR – Code soll mit Hilfe der Sensorik des mobilen Endgeräts ausgelesen werden.
Exit	Nach 1 Stunde Entwicklungszeit ist es möglichen Inhalt eines QR – Codes („Testcode“) korrekt auszulesen.
Fail	Nach 1 Stunde Entwicklung ist es nicht möglich den QR – Code mit Hilfe der Kamera des mobilen Endgeräts auszulesen. Ist die Kamera fehlerhaft oder defekt so ist dies ebenfalls ein gescheiterter Proof of Concept.
Fallback	Der Benutzer muss Informationen manuell eingeben.
Ergebnis	Der QR-Code konnte erfolgreich ausgelesen werden.

Tabelle 38: PoC Nr. 7 – Benutzerkonto erstellen

Beschreibung	Der Benutzer soll ein Benutzerkonto erstellen können, um sich im System einloggen zu können.
Exit	Nach 2 Stunden Entwicklungszeit ist es Möglich sich ein Konto mit einem Benutzernamen, E-Mail-Adresse und ein Passwort erstellen zu können und sich mit diesen im System einzuloggen.
Fail	Nach maximal 2 Stunden Entwicklungszeit ist es nicht Möglich sich Konto erstellen oder sich nicht mit den erstellten Informationen im System einzuloggen.
Fallback	Es ist keine Alternative vorhanden, da Informationen immer Benutzerbezogen und geschützt gespeichert werden müssen.

Tabelle 39: PoC Nr. 8 – E-Mail Verifizierung

Beschreibung	Der Benutzer erhält nach der Registrierung einen Benutzerlink, um seine E-Mail zu Verifizierung und das System zu nutzen.
Exit	Nach maximal 2 Stunden Entwicklungszeit ist es Möglich sich nach der Registrierung einen Bestätigungslink per Email zusenden zu lassen und die Emailadresse zu verifizieren.
Fail	Nach maximal 2 Stunden Entwicklungszeit ist es nicht Möglich sich nach der Registrierung einen Bestätigungslink per Email zusenden oder diesen zu Verwenden, um sich im System zu verifizieren.
Fallback	Es muss auf eine E-Mail-Verifizierung verzichtet werden.

Iteration Datenformate

In der ersten Version der Abwägung zwischen verschiedenen Datenformaten wurde nicht eindeutig ersichtlich, warum JSON Darstellungsformat gewählt wurde. Aus diesem Grund wurde eine Tabellarische Gegenüberstellung von Vor und Nachteilen drei verschiedener Datenrepräsentationen erstellt und zwischen diesen abgewägt.

JSON

Tabelle 28: Vor- und Nachteile JSON

Vorteile	Nachteile
Eignet sich gut für die gewählte NoSQL Datenbank.	Keine Darstellung von Datentypen.
Ist ein Teil von Javascript und eignet sich somit gut für eine Entwicklung mit Node.js.	Keine Möglichkeit Metadaten zu formulieren.
Besitzt eine gute menschliche Lesbarkeit.	
Besitzt einen einfachen Aufbau.	

Lässt sich gegen Schemata prüfen.	
Es besteht praktische Vorerfahrung.	
Unkomplizierte Serverseitige Verarbeitung möglich.	

XML

Tabelle 29: Vor- und Nachteile XML

Vorteile	Nachteile
Ist weit verbreitet und somit für viele verständlich.	Es besteht nur theoretische und keine praktische Vorerfahrung.
Viele Quellen und Dokumentationen vorhanden.	Sprache ist im Vergleich zu JSON sehr komplex und umfangreich.
Besitzt eine gute menschliche Lesbarkeit.	Besitzt ein Hohes Datenvolumen.
Besitzt eine gute Erweiterbarkeit.	Umfangreicheres Parsen von Daten ist notwendig.
Lässt sich gegen Schemata prüfen.	

YAML

Tabelle 30: Vor- und Nachteile YAML

Vorteile	Nachteile
Es liegt eine leichte Lesbarkeit vor.	Es besteht keine Vorerfahrung mit diesem Datenformat.
Die Syntax ist im sehr simpel gehalten.	Wenig Quellen bzw. Dokumentierung.
Vermeidet Redundanzen durch Relationale Daten.	Wenig ausgereifte Technologien zum Parsen von Daten

Die Vorteile, welche eine Entwicklung mit JSON mit sich bringt überwiegen, im Vergleich zu XML und YAML. Besonders die hohe Kompatibilität mit der gewählten Datenbank, Datenbankart und der gewählten Programmiersprache sowohl für Server als auch für Client und die zusätzliche Vorerfahrung machen die Entscheidung eindeutig. Eine weitere Alternative könnte sein sowohl JSON als auch XML zu unterstützen, jedoch ist dies für unser Projekt zu umfangreich und zeitlich nicht möglich.

Iteration Anforderungen:

Nach dem erhaltenen Feedback für das Konzept müssen die Anforderungen für das Projekt angepasst werden. Hierbei wurde der nicht benötigte Anforderungsteil „Anforderungen an die Benutzungsschnittstelle“ entfernt, jedoch wurde ein Abschnitt zu qualitativen Anforderungen hinzugefügt. Die Anforderungen an die Organisation wurden zu organisatorischen Anforderungen unbenannt und die technologischen Anforderungen wurden mit den funktionalen zusammengeführt. Somit ist eine übersichtliche und sinnvolle Übersicht gewährleistet. Ebenfalls wurden neben „muss“ und „soll“ auch „kann“ Anforderungen hinzugefügt.

Tabelle Nr. 14: Funktionale Anforderungen

Anforderungs-ID	Beschreibung
[A-100]	Der Benutzer muss die Möglichkeit besitzen, sich im System zu registrieren und einzuloggen.
[A-101]	Der Benutzer muss die Möglichkeit besitzen, seine Daten zu bearbeiten.
[A-102]	Der Benutzer muss die Möglichkeit besitzen, Angebote für andere Benutzer zu erstellen und zu bearbeiten.
[A-103]	Der Benutzer muss die Möglichkeit besitzen nach Angeboten anderer Benutzer suchen zu können.
[A-104]	Der Benutzer muss die Möglichkeit besitzen, Reservierungen für angebotene Lebensmittel zu erstellen und zu bearbeiten.
[A-105]	Das System muss für Angebote einen benutzerspezifischen Abholwert berechnen können.
[A-106]	Der Benutzer muss die Möglichkeit besitzen, Lebensmitteleinträge in seinem „Virtuellen Kühlschrank“ zu erstellen und zu bearbeiten.
[A-107]	Der Benutzer soll die Möglichkeit haben, seine frischen Lebensmittel (Obst und Gemüse) in automatisierter Form eintragen zu können mit Abschätzung des Haltbarkeitsdatums.
[A-108]	Das System soll den Benutzer rechtzeitig (mindestens 1 Tag vorher) benachrichtigen, falls Lebensmittel drohen abzulaufen.
[A-109]	Das System soll den Benutzer über Angebote, welche sehr gut zu ihm passen (Abholwert > 6) benachrichtigen.
[A-110]	Das System kann dem Benutzer eine Möglichkeit geben eine Verwendungsmöglichkeit in Form eines Rezeptes auf der Grundlage seiner eignen Lebensmittel und in der Nähe angebotenen Lebensmittel suchen zu können.
[A-111]	Das System soll dem Benutzer eine Möglichkeit geben eine Verwendungsmöglichkeit in Form eines Rezeptes auf der Grundlage seiner eignen Lebensmittel und Einschränkungen anbieten zu können.
[A-112]	Das System soll Vorlieben des Benutzers, anhand seines Kaufverhaltens ermitteln können.
[A-113]	Das System kann dem Benutzer Karten anzeigen lassen.
[A-114]	Das System kann dem Benutzer eine Kommunikationsmöglichkeit mit anderen Benutzern anbieten.
[A-115]	Das System soll Benutzer über den Status eingestellter Angebote informieren können.
[A-116]	Der Benutzer soll sich in dem System authentifizieren können.
[A-117]	Der Benutzer soll die Möglichkeit besitzen, Angebote auf der Grundlage seines „virtuellen Kühlschranks“ bzw. seiner Lebensmitteleinträge für andere Benutzer zu erstellen ohne bereits erhobene Informationen erneut eingeben zu müssen.
[A-118]	Der Benutzer muss die Möglichkeit besitzen persönliche Angaben wie Gesundheitliche Einschränkungen, Standort, Mobilität und Vorlieben eintragen zu können.
[A-119]	Das System kann den Benutzer über erhaltene Nachrichten benachrichtigen können.

[A-120]	Das System muss Angebote und Reservierungen verwalten können, sodass kein Wettrennen zwischen Benutzern entsteht
----------------	--

Tabelle Nr. 15: Organisatorische Anforderungen

Anforderungs-ID	Beschreibung
[A-200]	Das System muss in den Programmiersprachen Java und/oder Javascript programmiert worden sein.
[A-201]	Für das Projekt muss ein Projektplan vorliegen.
[A-202]	Für das Projekt muss Github als Versions-Kontroll-System verwendet werden.
[A-203]	Das Projekt muss einen gesellschaftlichen und/oder wirtschaftlichen Nutzen verfolgen.
[A-204]	Das Projekt muss verteilte Anwendungslogik aufweisen.
[A-205]	Das Projekt soll anhand des Vorgehensmodells „Usability Engineering Lifecycle“ nach Deborah Mayhew durchgeführt werden.
[A-206]	Das Projekt kann externe Dienste bzw. Dienstgeber mit einbinden.

Tabelle Nr. 16: Qualitative Anforderungen

Anforderungs-ID	Beschreibung
[A-300]	Die Daten der Benutzer müssen voneinander gekapselt vorliegen und vor Fremdzugriffen geschützt sein.
[A-301]	Die Daten sollen verschlüsselt übertragen werden können.
[A-302]	Das System soll auf fehlerhafte Eingaben durch Warnungen und/oder Korrekturen reagieren können.
[A-303]	Das System soll Daten persistent speichern können
[A-304]	Das System soll den Benutzer nicht mit Informationen überfluten, welche für die Erledigung der Aufgabe irrelevant sind.
[A-305]	Das System kann für den Benutzer Entscheidungen treffen, jedoch ihm die endgültige Auswahlmöglichkeit überlassen.
[A-306]	Das System kann auf Benutzereingabe schnell (Antwortzeit kleiner als 1 Sekunde) reagieren können.