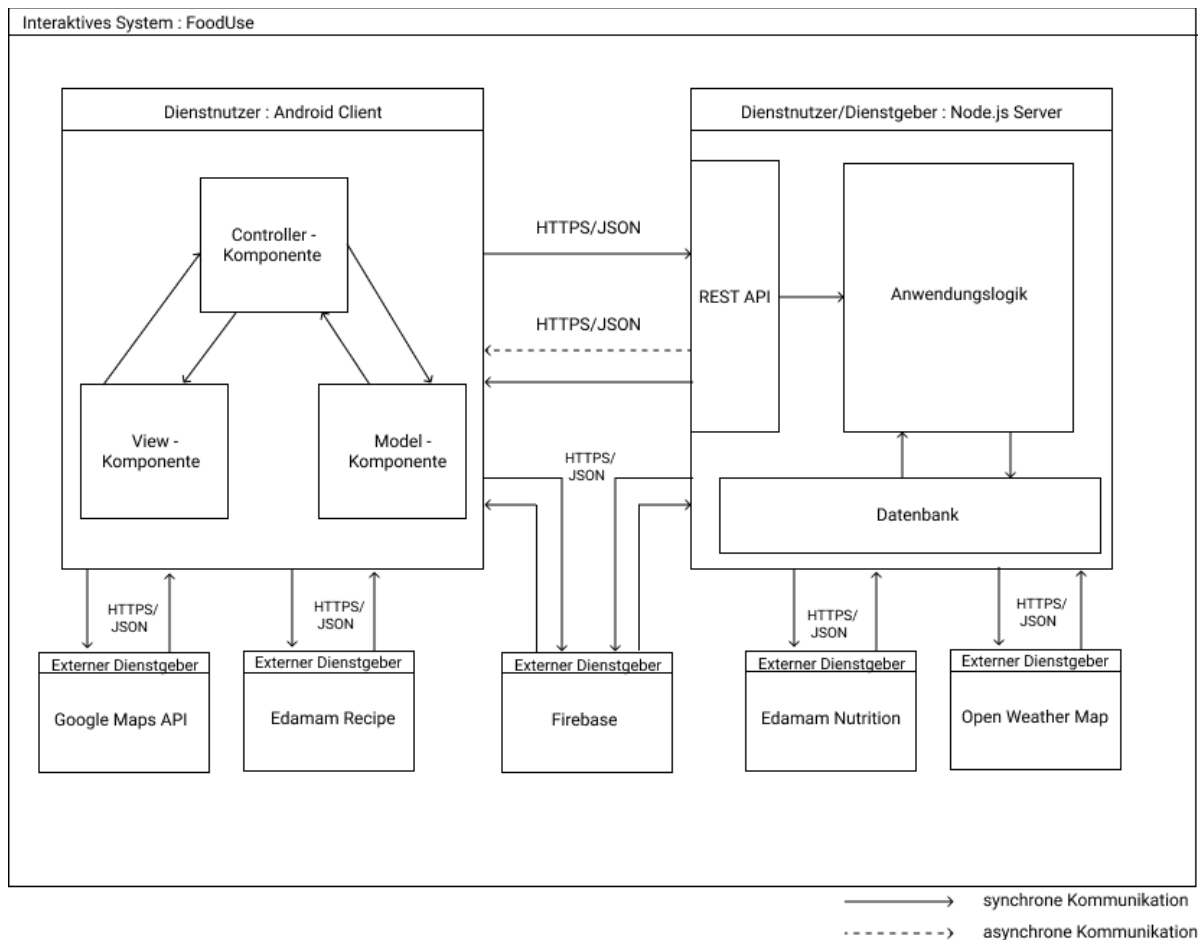


Abschnitt Webbasierte Anwendungen:

Iteration Architekturmodell:



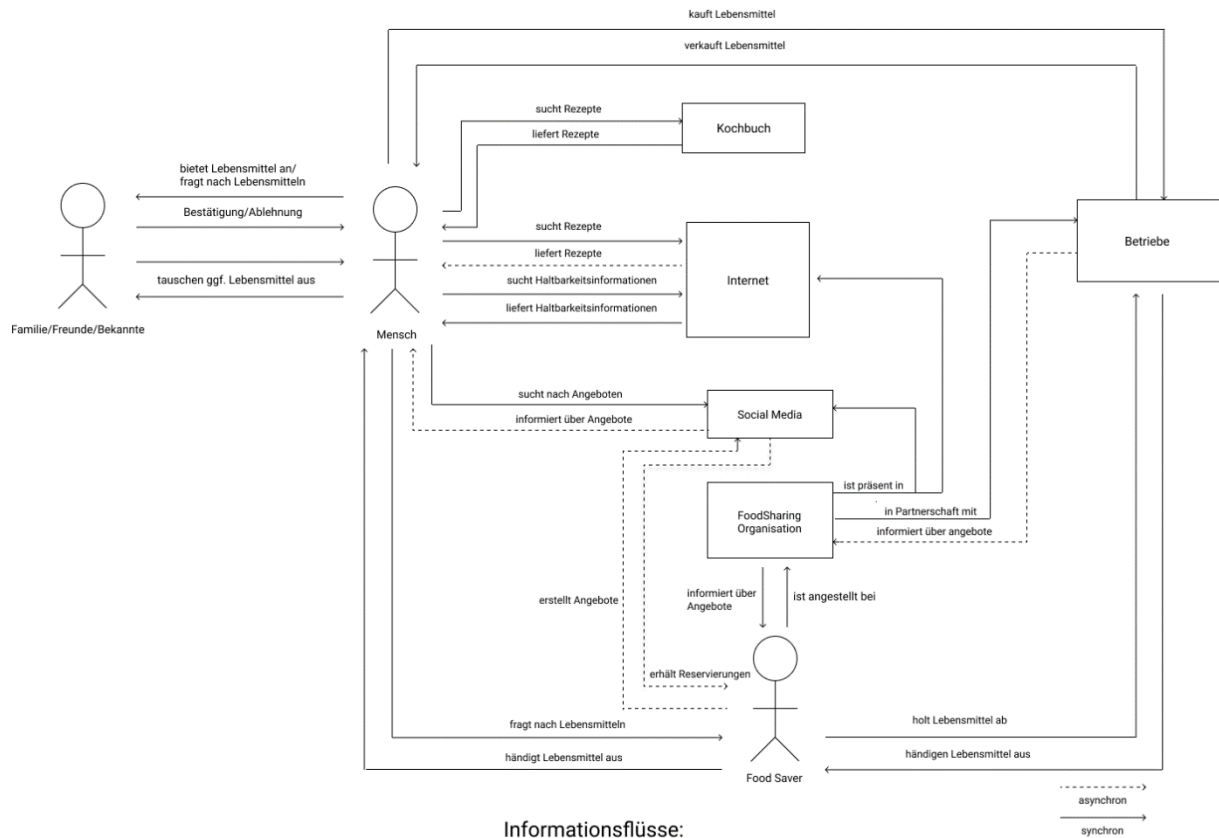
Das Grundlegende Architekturmodell hat sich nicht ausschlaggebend verändert. Die größte Veränderung ist auf der Seite des Clients zu sehen, welcher eine andere Architektur als zuvor besitzt. Es wurde sich für das Model-View-Controller Pattern entschieden, da dies im Vergleich zu der Schichten Architektur 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.

Ebenfalls wurde das Diagramm visuell verfeinert, da im Bezug auf die externen Unklarheiten bestanden. Der externe Dienstgeber Edamam, wurde in zwei separierte Dienste aufgeteilt, damit die notwendige Funktionalität besser zugeordnet werden kann. Der Dienst für die Bereitstellung von Rezepten ist nun auf der Seite des Clients, da an dieser Stelle die Lebensmittel des Benutzers gespeichert sind. Somit müssen Daten nicht unnötig zwischen Komponenten ausgetauscht werden. Die Analyse der Lebensmittel im Bezug auf gesundheitliche Einschränkungen ist auf der Server Seite zu finden, da dies für die Berechnung des Abholwerts relevant ist. Dies hätte theoretisch auch bereits auf dem Server geschehen können, jedoch ist so diese Funktionalität in sich abgeschlossen auf dem Server vorhanden. Würde dies auf der Clientseite liegen, so wäre Sie alleine nicht von besonderem nutzen.

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.



Informationsflüsse:

Rezept (Suche):
-vorhanden Lebensmittel

Haltbarkeitsinformations (Suche):
-Name

Lebensmittel verkauf der Betriebe(Beleg):
- Lebensmittel Namen
- Kaufdatum

Rezeptinformationen(Antwort):
-Titel
-Zutaten
-Beschreibung
-Bild

Haltbarkeitsinformation (Antwort):
- Datum

Informationen Lebensmitteltausch:
- Name
- Anzahl
- Zustand
- Uhrzeiten
- Treffpunkt

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 einige wichtige 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 angehören, 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.

Kommunikationsgang eines FoodSavers:

Um FoodSaver zu werden muss er sich zunächst bei einer FoodSharing Organisation melden und dort ein Zertifikat ablegen, um Lebensmittel an andere Menschen verteilen zu können. FoodSharing Organisationen sind grundsätzlich in Social Media Plattformen vertreten. Diese Organisationen stehen in Partnerschaften mit Supermärkten oder anderen Betrieben, und erhalten Nachrichten, falls es Lebensmittel zum abholen gibt. Falls ja erhält der FoodSaver in der Nähe des Supermarktes eine Nachricht und holt im Anschluss die Lebensmittel ab. Nachdem er Sie abholt stellt sich ihm genau die gleiche Frage wie der Privatperson. Der FoodSaver hat im Gegensatz zu der Privatperson die Möglichkeit die Lebensmittel in Social Media Plattformen anzubieten, da es sich in der Regel um größere Mengen handelt. Für eine Privatperson wurde diese Kommunikation nicht aufgeführt, da es so gut wie nicht vorkommt, dass eine Person einzelne Lebensmittel über Social Media Plattformen vermittelt.

Iteration Präskriptives Kommunikationsmodell:

An dieser Stelle wurden ebenfalls Antworten ergänzt, welche vernachlässigt bzw. vergessen wurden. Ebenfalls wurden die Informationen, welche zwischen den einzelnen Kommunikationspartner ausgetauscht werden ergänzt. Es ist zu erkennen das hier durch das System mehrere Informationen ausgetauscht werden, und diese Teilweise automatisiert durch das System ermittelt werden bzw. verarbeitet werden.

REST-Tabelle:

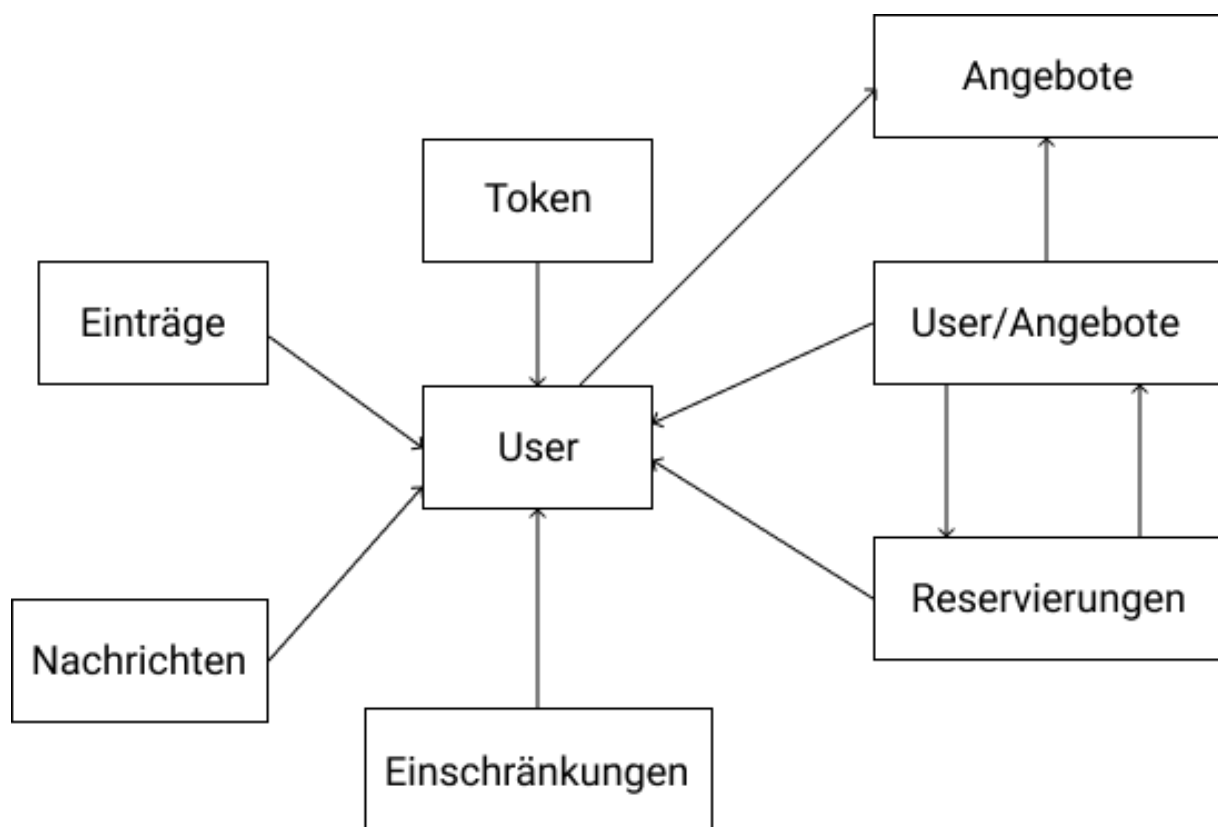
Im Folgenden wurde eine Tabelle mit allen Ressourcen der REST-Schnittstelle erstellt. Es werden verschiedene Ressourcen und Verben verwendet, um ein möglichst hohes Level des Richardson Maturity Models zu erreichen. Weiterhin wurden die Semantik, Typen von Request und Responses und die verwendeten Statuscodes.

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

/User/:id/Nachrichten	DELETE	Löscht eine Nachricht eines Benutzers	Application /JSON	Application /JSON	
/User/:id/Token	GET	Gibt den Token eines Benutzers zurück.	Application /JSON	Application /JSON	
/User/:id/Token	PUT	-	-	-	
/User/:id/Token	POST	Erstellt einen Token für einen Benutzer.	Application /JSON	Application /JSON	
/User/:id/Token	DELETE	Löscht den Token eines Benutzers.	Application /JSON	Application /JSON	
/User/:id/Angebote	GET	Gibt alle erstellten Angebote eines Benutzers zurück.	Application /JSON	Application /JSON	
/User/:id/Angebote/:id	GET	Gibt ein Angebot eines Benutzers zurück.	Application /JSON	Application /JSON	
/User/:id/ Angebote/:id	PUT	Verändert ein erstelltes Angebot eines Benutzers.	Application /JSON	Application /JSON	
/User/:id/ Angebote	POST	Erstellt ein neues Angebot eines Benutzers.	Application /JSON	Application /JSON	
/User/:id/ Angebote/:id	DELETE	Löscht ein Angebot eines Benutzers	Application /JSON	Application /JSON	
/User/:id/Reservierungen	GET	Gibt alle Reservierungen eines Benutzers zurück.	Application /JSON	Application /JSON	
/User/:id/Reservierungen/:id	GET	Gibt eine Reservierung eines Benutzers zurück.	Application /JSON	Application /JSON	
/User/:id/Reservierungen/:id	PUT	Verändert eine Reservierung eines Benutzers.	Application /JSON	Application /JSON	
/User/:id/Reservierungen	POST	Erstellt eine neue Reservierung für einen Benutzer.	Application /JSON	Application /JSON	
/User/:id/Reservierungen/:id	DELETE	Löscht eine Reservierung eines Benutzers.	Application /JSON	Application /JSON	
/Angebote	GET	Gibt eine Liste aller Angebote zurück.	Application /JSON	Application /JSON	
/Angebote/:id	GET	Gibt die optimalen Angebote für einen Benutzer zurück.	Application /JSON	Application /JSON	
/Angebote/:id?radius=X	GET	Gibt Angebote in einem	Application /JSON	Application /JSON	

		vorgegebenen Radius zurück.			
/Angebote/:id?AbholwertMax=X&AbholwertMin=Y	GET	Gibt Angebote in einem vorgegebenen Abholwertbereich zurück.	Application /JSON	Application /JSON	

Um auch das Hypermedia-Konzept des Modells nicht zu vernachlässigen wurde in folgender Grafik die geplante Verlinkung der Ressourcen untereinander dargestellt. An dieser Stelle haben wir Ressourcen untereinander verlinkt, welche bei weiteren Anfragen interessant bzw. nützlich sein könnten.



Iteration Domänenrecherche:

Welche Möglichkeiten hat der Benutzer Lebensmittel weiterzugeben? Grundsätzlich benutzt der die Möglichkeit Lebensmittel an die Familie weiterzugeben. Dies ist wahrscheinlich die an der häufigsten verwendeten Möglichkeit, da die meisten Menschen ihrer Familie helfen bzw. etwas Gutes tun wollen. Ebenfalls könnten Freunde oder Bekannte ebenfalls eine Möglichkeit sein Lebensmittel abzugeben. Doch welche Möglichkeiten bestehen außerhalb des Bekanntenkreises? An dieser Stelle gibt es lediglich die Möglichkeit Lebensmittel an Organisationen wie die Tafel z.B. abzugeben.

Die Tafel:

Aktuell gibt es 940 gemeinnützige Tafeln in Deutschland welche Menschen die Möglichkeit geben kostenlos Lebensmittel zu erhalten. Dies geht jedoch über Lebensmittel hinaus, auch Möbel, Kleidung oder andere Leistungen bietet die Tafel an. Die Tafel hat als Ziel von Armut betroffenen Haushalten zu helfen und ihre Situation zu verbessern. Grundsätzlich kann man als Privatperson, übrig gebliebene Lebensmittel an die Tafel spenden, welche an andere weitergegeben werden.

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 Domänenreche 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 Interview 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 x: Contextual Observation/Interview 1. Normaler Benutzer

Actor	Trigger	Use Case	Task Scenario Sequence	Problems
Normal Benutzer	Bemerkt, dass Lebensmittel übriggeblieben sind.	Nach geeigneter Verwendungsmöglichkeit für die Lebensmittel suchen.	<ol style="list-style-type: none"> 1. Benutzer realisiert, dass Lebensmittel übrig sind. 2. Benutzer merkt sich die Lebensmittel die er Verbrauchen möchte. 3. Benutzer wählt eine Recherchequelle (Internet, Kochbuch). Benutzer wählt Verwendungsmöglichkeit aus. 4. Benutzer vergleicht gefundene Verwendungsmöglichkeiten nach Aufwand, Einschränkungen, Vorlieben, Zubereitungszeiten und vorhandenen Zutaten. 	<ol style="list-style-type: none"> 1. Es ist ein großer Aufwand eine geeignete Verwendungsmöglichkeit zu finden. 2. Gefundene Rezepte können aus verschiedenen Gründen nicht geeignet sein. 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.	<ol style="list-style-type: none"> 1. Benutzer entdeckt ein Lebensmittel in einem unklarem Haltbarkeitszustand. 2. Benutzer merkt sich das Lebensmittel. 3. Benutzer sucht sich ein geeignetes Recherchemedium aus. (Internet) 4. Benutzer sucht nach Erfahrungswerten oder ähnlichem. 5. Benutzer entscheidet ob Lebensmittel noch verzehrbar ist oder entsorgt werden muss. 	<ol style="list-style-type: none"> 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.

Tabelle x: Contextual Observation/Interview 2. Anbieter

Actor	Trigger	Use Case	Task Scenarios Sequence	Problems
Anbieter	Bemerkt, dass Lebensmittel übriggeblieben sind, welche er nichtmehr verwenden möchte.	Nach einer Möglichkeit suchen die Lebensmittel anderen Menschen anzubieten.	<ol style="list-style-type: none"> 1. Anbieter bemerkt, dass Lebensmittel in seinen Haushalt sind, welche er nicht mehr verwenden will bzw. kann. 2. Sucht Lebensmittel heraus und legt sie ggf. zusammen, um einen besseren Überblick zu behalten. 3. Anbieter sucht nach Möglichkeiten seine Lebensmittel abzugeben. (Internet/Social Media) 	<ol style="list-style-type: none"> 1. Anbieter machen sich i.d.R nicht die Mühe (besonders bei einzelнем Lebensmittel) nach Möglichkeiten zur Abgabe oder Ähnlichem zu suchen. 2. Lebensmittel kann eventuell bereits abgelaufen sein und sollte nichtmehr angeboten werden.
	Bemerkt, dass Lebensmittel übriggeblieben sind, welche er nichtmehr verwenden möchte.	Lebensmittel im Bekanntenkreis anbieten.	<ol style="list-style-type: none"> 1. Anbieter bemerkt, dass Lebensmittel in seinen Haushalt sind, welche er nicht mehr verwenden will bzw. kann. 2. Sucht Lebensmittel heraus und legt sie ggf. zusammen, um einen besseren Überblick zu behalten. 3. Anbieter Überlegt sich, wer aus seinem Bekannten Kreis das Lebensmittel gebrauchen könnte. 4. Anbieter Kontaktiert diese Person (Internet/Telefon/Direkter Kontakt). 5. Anbieter übergibt die Lebensmittel. 	<ol style="list-style-type: none"> 1. Anbieter machen sich i.d.R nicht die Mühe (besonders bei einzelнем Lebensmittel) nach Möglichkeiten zur Abgabe oder Ähnlichem zu suchen. 2. Lebensmittel kann eventuell bereits abgelaufen sein und sollte nichtmehr angeboten werden. 3. Es gibt niemanden im Bekanntenkreis, den das Lebensmittel interessieren könnte. 4. Alle möglichen Interessenten wohnen zu weit weg.
	Bemerkt, dass Lebensmittel übriggeblieben sind, welche er nichtmehr verwenden möchte.	Lebensmittel an Organisationen spenden.	<ol style="list-style-type: none"> 1. Anbieter bemerkt, dass Lebensmittel in seinen Haushalt sind, welche er nicht mehr verwenden will bzw. kann. 2. Sucht Lebensmittel heraus und legt sie ggf. zusammen, um einen besseren Überblick zu behalten. 3. Anbieter sucht nach möglichen Organisationen die Lebensmittel zu spenden. 4. Anbieter setzt sich mit der Organisation in Kontakt und 	<ol style="list-style-type: none"> 1. Anbieter machen sich i.d.R nicht die Mühe (besonders bei einzelнем Lebensmittel) nach Möglichkeiten zur Abgabe oder Ähnlichem zu suchen. 2. Lebensmittel kann eventuell bereits abgelaufen sein und sollte nichtmehr angeboten werden. 3. Es gibt keine Organisation in der Nähe,

			erkundigt sich über den Abgabeprozess.	bei der gespendet werden kann.
--	--	--	--	--------------------------------

Tabelle x: Contextual Observation/Interview 3. Abholer

Actor	Trigger	Use Case	Task Scenarios Sequence	Problems
Abholer	Abholer, besitzt wenig Geld und möchte deshalb nach kostenlosen Angeboten in seiner Umgebung suchen.	Nach einer Möglichkeit suchen Lebensmittel kostengünstig abzuholen	<ol style="list-style-type: none"> 1. Benutzer stellt fest, dass er Lebensmittel benötigt 2. Benutzer sucht nach Quellen zur Recherche von Möglichen Angeboten. 3. Benutzer vergleicht verschiedene Angebote nach seinen Bedürfnissen. 4. Benutzer wählt ein geeignetes Angebot aus. 	<ol style="list-style-type: none"> 1. Kann sehr zeitaufwändig sein ein geeignetes Angebot zu finden. 2. Weiß ggf. nicht über den Zustand des Angebots Bescheid. 3. Lebensmittel kann nicht dem erwarteten entsprechen.
	Abholer hat Hunger und möchte Angebote aus der Umgebung wahrnehmen.	Nach einer Möglichkeit suchen Lebensmittel bei Familie/Freunde bzw. Bekannten abholen.	<ol style="list-style-type: none"> 1. Benutzer überlegt sich wen er kontaktieren kann, um Lebensmittel zu erhalten. 2. Benutzer wählt eine Person aus. 3. Abholer kontaktiert diese Person und erkundigt sich nach möglichen Angeboten. 4. Abholer nimmt das Angebot an oder sucht andere Möglichkeiten. 5. Bei gefundenem Angebot, welche für ihm annehmbar ist macht der Abholer mit dem Anbieter einen Treffpunkt und Uhrzeit aus. 6. Benutzer entscheidet sich, wie er zum Zielort gelangen soll. (Auto, zu Fuß, öffentliche Verkehrsmittel) 7. Abholer holt Lebensmittel ab. 	<ol style="list-style-type: none"> 1. Kann sehr zeitaufwändig sein ein geeignetes Angebot zu finden. 2. Weiß ggf. nicht über den Zustand des Angebots Bescheid. 3. Lebensmittel kann nicht dem erwarteten entsprechen. 4. Ggf. kann es für den Abholer unangenehm sein bei seiner Familie oder Freunden Lebensmittel zu holen. 5. Es kann sein keine Angebote zu finden, da der Freundeskreis oder die Familie sehr begrenzt ist.
	Abholer hat Hunger und möchte Angebote aus der Umgebung wahrnehmen.	Nach einer Möglichkeit suchen Lebensmittel bei Organisationen abzuholen bzw. kostengünstig zu erwerben.	<ol style="list-style-type: none"> 1. Benutzer erkundigt sich über Abholmöglichkeiten in seiner Nähe (im Internet, durch Bekannte) 2. Benutzer vergleicht verschiedene Abholmöglichkeiten miteinander 	<ol style="list-style-type: none"> 1. Kann sehr zeitaufwändig sein ein geeignetes Angebot zu finden. 2. Weiß ggf. nicht über den Zustand des Angebots Bescheid.

			<p>3. Benutzer wählt die Möglichkeit aus, die seinen Bedürfnissen entspricht</p> <p>4. Benutzer erkundigt sich über Abholort und Zeitpunkt</p> <p>5. Benutzer geht zur entsprechenden Zeit zu dem angegebenen Abholort</p> <p>6. Benutzer nimmt die Lebensmittel entgegen</p>	<p>3. Lebensmittel kann nicht dem erwarteten entsprechen.</p> <p>4. Abholer besitzt keinen Ausweis der ihn dazu berechtigt die Lebensmittel zu erhalten bzw. zu erwerben.</p>
--	--	--	---	---