

---

EIS

# Entwicklung interaktiver Systeme

Sommersemester 2018

---

## Modellierungsgrundlage MS2

**Team:**

Jan Omar Mehr

Armin Weinrebe

**Mentor:**

Robert Gabriel

**Dozenten:**

Prof. Dr. Gerhard Hartmann

Prof. Dr. Kristian Fischer

# Inhaltsverzeichnis

<b>1 Iteration: Zielhierarchie</b>	<b>1</b>
1.1 Strategische Ziele . . . . .	1
1.2 Taktische Ziele . . . . .	1
1.3 Operative Ziele . . . . .	2
1.4 Minimalziele . . . . .	3
<b>2 Verstehen und Festlegen des Nutzungskontexts</b>	<b>4</b>
2.1 Recherche . . . . .	4
2.1.1 Reisetypen . . . . .	4
2.1.2 Interview . . . . .	4
2.1.3 Spezifizierte Benutzergruppe . . . . .	5
2.1.4 Kulturelle Unterschiede . . . . .	5
2.2 Benutzermodellierung . . . . .	6
2.2.1 Benutzer Profile . . . . .	6
2.2.2 Personae . . . . .	8
2.2.3 Stakeholderanalyse . . . . .	9
2.3 Aufgabenmodellierung . . . . .	15
2.3.1 Szenarien . . . . .	15
2.3.2 Erkenntnisse zum Nutzungskontext . . . . .	16
2.3.3 Hierarchical Task Analysis (HTA) . . . . .	19
<b>3 Festlegen der Nutzungsanforderungen</b>	<b>21</b>
3.1 Funktionale Anforderungen . . . . .	21
3.2 Qualitative Anforderungen . . . . .	22
3.3 Organisatorische Anforderungen . . . . .	22
<b>4 Erarbeiten von Gestaltungslösungen</b>	<b>24</b>
4.1 Struktur . . . . .	24
4.1.1 Dialoggestaltung . . . . .	24
4.1.2 Navigation . . . . .	25
4.2 Seitenlayout . . . . .	27
4.2.1 Navigation . . . . .	27
4.2.2 Abrechnungübersicht . . . . .	30
4.2.3 Abrechnungdetails . . . . .	31
4.2.4 Gruppen-Kassenzettel (Liste) . . . . .	33
4.2.5 Kassenzettel-Details . . . . .	34
4.2.6 Einkaufsliste . . . . .	37
4.3 Visuelle Gestaltung . . . . .	39
4.3.1 Farbe . . . . .	39
4.3.2 Schriften . . . . .	40
4.3.3 Piktogramme/ Symbole . . . . .	40

<b>5 Evaluation</b>	<b>42</b>
5.1 Durchführung der Heuristischen Evaluation . . . . .	42
<b>6 Systemarchitektur</b>	<b>45</b>
6.1 Server, Kommunikation, Protokoll . . . . .	45
6.2 Datenmodellierung . . . . .	48
6.3 REST URI Modellierung: . . . . .	50
6.4 Statuscodes . . . . .	53
6.5 PubSub-Topics . . . . .	54
<b>7 Anwendungslogik</b>	<b>56</b>
<b>8 Proof Of Concept</b>	<b>57</b>
8.1 Optical Character Recognition (Scan) . . . . .	57
8.2 Abrechnungs-Algorithmus . . . . .	58
8.3 Client-Server Kommunikation . . . . .	59
8.4 Firebase Cloud Messaging (FCM) . . . . .	60
8.5 Webservice: Online-Bezahldienst “Paypal” . . . . .	61
8.6 Webservice: Währungsrechner “currencyconverterapi” . . . . .	63
<b>9 Ausführung der PoC</b>	<b>64</b>
9.1 OCR . . . . .	64
<b>10 Anhang</b>	<b>66</b>
10.1 zusätzliche Wireframes . . . . .	66
10.1.1 Login . . . . .	66
10.2 Use Cases . . . . .	67
10.3 Durchführung der Interviews . . . . .	69
10.4 User Profiles . . . . .	72
10.5 Personae . . . . .	76
10.6 Szenarien . . . . .	81
10.7 HTA's . . . . .	84

# **1 Iteration: Zielhierarchie**

Um das Projekt später hinsichtlich des Erfogsgrades richtig evaluieren zu können, musste die Zielhierarchie verbessert und iteriert werden.

## **1.1 Strategische Ziele**

- Es soll ein System entstehen, welches unbekannten, anderssprachigen, kulturell unterschiedlich geprägten Benutzern ermöglicht, Kassenzettel gemeinsam konfliktfrei, verständlich und gebrauchstauglich abrechnen zu können.
- Es soll erreicht werden, dass bei diesen Benutzern, innerhalb einer Gruppe, das Vertrauensverhältnis gefördert und soziale Spannungen vermieden werden, durch
  - eine eindeutig kommunizierte Kostenverteilung.
  - eine positive User Experience.
- Benutzer sollen sich durch das System permanent unterstützt fühlen, unabhängig von Ort und Zeit.
- Es können Erkenntnisse zum Kaufverhalten von Benutzern gewonnen werden.

## **1.2 Taktische Ziele**

- Es müssen Funktionen zum kollaborativen Abrechnen erarbeitet werden auf der Grundlage von
  - Modellen aus der Softwaretechnik
  - standardisierten Methoden der Erstellung Webbasiert Anwendungen.
- Es muss eine Systemarchitektur erarbeitet werden, die Systemkomponenten verteilt und eine Erweiterung auf eine unbegrenzte Anzahl mobiler Endgeräte ermöglicht.
- Es muss eine Struktur für einen umfangreichen Austausch an Informationen erarbeitet werden, auf der Grundlage von Kommunikationsparadigmen.
- Es soll nach einem Vorgehensmodell gearbeitet werden, das das Design für die spezifische Zielgruppe anpasst und sich gleichzeitig auf Modelle der Softwaretechnik skalieren lässt.
- Es müssen Methoden aus dem Usability Engineering verwendet werden, um ein hohes Ausmaß an Gebrauchstauglichkeit zu erzielen.
- Es können Probanden aus der spezifischen Benutzergruppe bei der Entwicklung von Modellen mit einbezogen werden.

- Es müssen Risiken für das Projekt definiert werden und auf deren Grundlage PoC's erstellt werden.
- Es müssen Evaluationsmethoden angewendet werden, sodass Modelle geprüft und ggf. neu erarbeitet oder überarbeitet werden.
- Es muss eine Implementation des intendierten Systems auf Grundlage der evaluierten Modelle stattfinden.
- Das Kaufverhalten von Benutzern kann analysiert und
  - für sie aufbereitet und sichtbar gemacht werden.
  - für Geschäftspartner auf kommerziell verwertbare Daten reduziert werden.

### **1.3 Operative Ziele**

- Es sollen weitere Erkenntnisse durch eine intensivere Recherche der spezifizierten Benutzergruppe gewonnen werden.
- Es müssen Ressourcen der benötigten Daten spezifiziert werden.
- Es soll ein Topic-Modell für asynchrone Kommunikation erarbeitet werden.
- Es sollen Modelle zur geplanten Datenstruktur erarbeitet werden.
- Es muss eine Stakeholderanalyse durchgeführt werden.
- Es müssen Benutzer- und Nutzungsmodelle erarbeitet werden.
- Es müssen Anforderungen aus den Erwartungen der Benutzer abgeleitet werden.
- Es müssen Gestaltungslösungen erarbeitet und evaluiert werden.
- Es müssen PoC's formuliert und durchgeführt werden.
- Es muss ein Server mit Datenbankanbindung implementiert werden.
- Es muss ein Client implementiert werden.
- Es soll OCR-Technologie eingebunden werden.
- Es können Schnittstellen zu externen Services eingebunden werden.
- Das Projekt kann in der Planung angepasst und Aktivitäten ggf. iteriert werden.

## **1.4 Minimalziele**

- Die virtuelle Erstellung von Gruppen
- Die Erstellung von Abrechnungen anhand von Kassenzetteldaten
- Die kollaborative Interaktion mit den Abrechnungen und Kommunikation der Gruppe im System.
- Darstellung und Lösung von Konflikten in einer Abrechnung.

## **2 Verstehen und Festlegen des Nutzungskontexts**

### **2.1 Recherche**

Um den Nutzungskontext und die Zielgruppe präziser eingrenzen zu können wird die Recherche im Bereich Backpacker erweitert.

#### **2.1.1 Reisetypen**

Grundsätzlich lassen sich laut TravelWorks Backpacker in verschiedene Reisetypen klassifizieren, die je nach Ausprägung der Kriterien Abenteuerlust, Reiseerfahrung, Aufgeschlossenheit, Entdeckerdrang, Planungsbedürfnis und Budget zugeordnet werden.[1] Auf dessen Grundlage wird im Projekt zwischen erfahrenen Abenteurern, Entdeckern, Weltenbummlern und Nomaden oder Reise-Anfängern, die ihre Ausprägungen erst noch genauer kennenlernen müssen, unterschieden. Die Altersgruppe von Backpackern ist im weltweiten Vergleich sehr schwer zu bestimmen, da es auf der einen Seite Familien gibt, die Kinder mitnehmen, sodass Backpacking als Teil eines Kurzurlaubes zuzuordnen ist. Auf der anderen Seite gibt es hochmotivierte Einzelgänger (oder Kleingruppen) zwischen 18 und 80 Jahren, deren teilweise extreme Backpacker-Routen Weltreise-Charakter annehmen oder in einer anderen Form sehr speziell sind.[2]

#### **2.1.2 Interview**

Um den Nutzungskontext und die Benutzer besser zu verstehen werden zwei Interviews durchgeführt, aus denen Erkenntnisse für die Stakeholderanalyse gewonnen werden konnten. Zum Nachteil des Projektes fließen die Informationen nicht in die Benutzer- und Aufgabenmodelle mit ein, da sich erst spät Personen mit Backpacker-Erfahrung dazu bereit erklärt haben. Jedoch überschneiden sie sich teilweise mit Recherchen aus Sekundärquellen.

- Es werden verschiedene Apps genutzt, auch um Kontakte zu finden
- Es werden vor Ort fremde Personen kennen gelernt, woraus Freundschaften entstanden
- Die Reisen wurden alleine und zu zweit unternommen
- Finanzielle Aushilfen basieren auf einer Vertrauensbasis
- Finanzielle Engpässe kommen in beiden Schilderungen vor, aus unterschiedlichen Gründen
- Konflikte basieren i.d.R. auf kommunikativen Missverständnissen

### **2.1.3 Spezifizierte Benutzergruppe**

Die überwiegende Anzahl an Backpackern lässt sich jedoch mit der Reisekategorie Work and Travel zusammenführen. Diese Gruppe ist zwischen 18 und 30 Jahre alt. In den Jahren von 2000 bis 2015 ließ sich ein weltweiter Anstieg beantragter Working-Holiday Visa von 79.000 auf 226.812 messen. Dies zeigt, wie stark sich die Globalisierung auf den Bereich Backpacking auswirkt. Mit ihr hat sich gleichzeitig die Digitalisierung entwickelt, sodass viele alte Mittel zur Kommunikation, Dokumentation, Information und Transaktion durch Anwendungen in Smartphones ersetzt werden. [3]

Am Beispiel Australiens, einem der beliebtesten Work and Travel Ziele, kann die Benutzergruppe der 18 bis 30 Jährigen als dominanter Anteil von 90 Prozent aller Backpacker genauer betrachtet werden. 60 Prozent der Backpacker kommen aus Europa und sie "haben ein ausgeprägt kurzfristiges Buchungsverhalten". [4] Die Reiseplanung befragter deutscher Backpacker findet entweder grob im vorhinein oder spontan statt. Zwei Drittel der Befragten reist entweder allein oder zu zweit für im Schnitt acht bis neun Monate. Geschlafen wird in Hostels oder im Van. Die häufigsten Erwartungen der Befragten sind [3]:

- Entwicklung der Persönlichkeit
- Abenteuer erleben
- Leute kennenlernen

Die häufigsten Ausgaben sind:

- Verpflegung, Unterkunft
- Flüge,
- Touren, Sightseeing.

### **2.1.4 Kulturelle Unterschiede**

Auch hier wird das Beispiel Australien angeführt, da die Regierung Australiens die Zahlen beantragter WHV (Working Holiday Visa) der verschiedenen Nationalitäten für das Jahr 2014/2015 online veröffentlicht hat und als einzige beziehbare Quelle zur Verfügung steht. [5] Eine Rangliste der zehn Nationen mit den meisten beantragten WHV für Australien:

- United Kingdom
- Taiwan
- South Korea
- Fed. Republic of Germany

- France
- Japan
- HKSAR of the PRC (China)
- Republic of Ireland
- United States of America
- Canada

Es wird davon ausgegangen, dass in diesen Nationen auch weltweit mit die meisten WHV beantragt werden, unabhängig von der exakten Rangfolge. Die Grundlage für die Ermittlung kultureller Unterschiede soll sich primär auf westlich geprägte Länder beziehen und sekundär auf fern östliche Bereiche. Der Vergleich findet auf Basis wissenschaftlich erforschter Untersuchungen zu interkulturellen Unterschieden statt, die für Geschäftsbeziehungen von der IHK Mittlerer Niederrhein neben allgemeiner interkultureller Kompetenzen in einer Sammlung an Informationen zusammengefasst werden[6]. Für das Projekt sind soziale Aspekte, wie dem typischem Kommunikationsstil und Tendenzen der allgemeinen Orientierungspunkte in den Kulturen interessant (siehe Anhang). Das Hauptaugenmerk liegt jedoch vor allem auf der Sprachbarriere, die im Nutzungskontext zu Missverständnissen und Folgeproblemen führen kann.

## **2.2 Benutzermodellierung**

### **2.2.1 Benutzer Profile**

Die Recherchen zu den verschiedenen Reisetypen hat ergeben, dass es einige erfahrene Reisende gibt, die über wertvolles Wissen verfügen könnten. Jedoch resultiert aus weiteren Recherchen, dass der junge unerfahrene Reisetyp des “Reise-Anfängers” am häufigsten vertreten ist unter den Backpackern. Das Interesse an dem intendierten System könnte durchaus stark sein, da der Gebrauch moderner Technologien genutzt wird (vgl. Domänenrecherche) und ein soziales und organisatorisches Bedürfnis nach transparenter Kostenverteilung in Form des kollektiven Tragens der Verpflegungsausgaben mit fremden Personen besteht. Als folgendes Beispiel wird der Benutzer-Typ des “Entdeckers” dargestellt, der im Gegensatz zum Durchschnitt-Benutzer ein wenig Erfahrung besitzt.

<b>Merkmal</b>	<b>Ausprägung</b>
Alter	Meist zwischen 22 und 30 Jahren
Geschlecht	m/w
Herkunft	Überwiegend Europa, Asien, Nord Amerika
Bildung	mittel bis sehr gut
Sprachen	Erstsprache, Englisch und mehr
Computerkenntnisse	mittel bis viel
Körperl. Verfassung	gesund
Abenteuerlust	wenig bis mittel
Reiseerfahrung	mittel bis viel
Aufgeschlossenheit	sehr viel
Entdeckerdrang	viel bis sehr viel
Planungsbedürfnis	mittel bis viel
Budget	mittel bis viel
Spezielle Fähigkeiten	Kann schnell Kontakt herstellen
Nutzungsmotivation	mittel bis hoch
Nutzungseinstellung	interessiert
Verantwortungsbewusstsein	Eher hoch
Rolle bei gemeinsamen Kosten in einer Gruppe	Gläubiger

Tabelle 1: Benutzerprofil eines Entdeckers

## 2.2.2 Personae

Die Personae orientieren sich an Ergebnisse der Domänenrecherche und den Benutzerprofilen und beschreiben deskriptiv, wie sich die definierten Merkmale in fiktive Personenbeschreibungen integrieren lassen. Als Beispiel wird der Reisetyp eines Entdeckers vorgestellt.



Abbildung 1: William aus Kanada (Quelle: FreeImages.com/Angela Granger)

William ist 29 Jahre alt und als Forscher in Vancouver an der University of British Columbia in der Abteilung Sozial- und Persönlichkeitspsychologie tätig. Er besitzt einen ausgeprägten Entdeckerdrang und lebt diesen sowohl in seinem Arbeitsumfeld wie auch in seiner Freizeit aus. Während seines Grundstudiums hat er zwei Auslandssemester in England verbracht. Dort zog es ihn in den Semesterferien in die raue Natur im Norden Großbritanniens. Seitdem reist er mindestens einmal im Jahr an Orte die für Backpacker geeignet sind. Er weiß, dass er die Kosten durch Selbstversorgung minimieren kann. Außerdem und schließt sich deswegen gerne mit Gleichgesinnten zusammen, die er entweder in Backpacker-Foren oder vor Ort kennenlernt. Bei der Planung und Beschaffung gemeinsamer Unterkünfte und Versorgungen übernimmt er sehr gerne die Verantwortung. Sein

alltäglicher Umgang mit seinen Mitarbeitern und ständig wechselnden Probanden machen ihn neben seinem Fachwissen zum professionellen Menschenkenner. Er scheut es nicht fremde Menschen anzusprechen, kennen zu lernen und auf sie einzugehen. Auch wenn er ein gewisses Plan-Bedürfnis verspürt, ist er offen für spontane Planänderungen, falls seine Mitreisenden ihn davon überzeugen können.

### 2.2.3 Stakeholderanalyse

Bezeichnung	Beziehung	Objektbereich	Erfordernisse/ Erwartungen	Priorität
Alle Nutzer	Anrecht	Daten	müssen wissen, dass mit ihren persönlichen Daten vertraulich umgegangen wird, um in Kontakt mit potentiell fremden Menschen zu treten	1
		Daten	müssen wissen, dass ihre Daten nicht verloren gehen oder missbraucht werden, um diese einem externen System oder Gruppenmitglied anzuvertrauen	1
	Anspruch	Daten	müssen wissen, dass sie zu jedem Zeitpunkt auf ihre Daten und Dokumente zugreifen können, um mit diesen arbeiten zu können	1
	Interesse	Statistik	müssen zu jedem Zeitpunkt auf ihre Kassenzettel zugreifen können, um ihr Kaufverhalten analysieren zu können	1
Nutzer: Gruppenmitglieder	Anrecht	Daten	müssen zu jedem Zeitpunkt wissen oder überprüfen können, wer in ihrer Gruppe ist, um sicher zu sein, dass ihre Daten nicht von Fremden/Unbefugten eingesehen werden können	1

			müssen Abrechnungen, in denen sie einbezogen wurden, einsehen und nachprüfen können, um die Korrektheit bestätigen zu können	1
		Geld/ Kommunikation	müssen wissen, dass sie überschüssiges Geld rückerstattet bekommen können, falls dieses Geld bereits vor dem Einkauf zusammengetragen wurde	1
	Anspruch	Kommunikation	müssen zu jedem Zeitpunkt wissen oder überprüfen können, wer in ihrer Gruppe ist, um mit den Mitgliedern über gruppeninterne Themen kommunizieren zu können	1
			müssen zu jedem Zeitpunkt wissen oder überprüfen können, welche Sprachen die Mitglieder sprechen, um ihre Kommunikation mit diesen anpassen zu können	1
			müssen der Gruppe austreten können, wenn keine Rechnungen offen sind	1
			müssen wissen, wer der Gruppe ausgetreten ist, um entscheiden zu können, wer nicht mehr in gruppeninterne Themen einbezogen wird	1

			müssen ehemalige Gruppenmitglieder nach Austritt aus der Gruppe erreichen können, um offene Rechnungen zu kommunizieren	1
	Interesse	Kollaboration/ Kommunikation	müssen mit den restlichen Mitgliedern Wünsche für einen Einkauf vorbringen und diese besprechen können, um diesen planen zu können	1
		Daten	müssen Zugriff auf aktuelle Daten des Währungsumtausch-Kurses haben, um Abrechnungen in verschiedenen Währungen begleichen zu können	1
Nutzer: Gruppen- ersteller	Anspruch	Kommunikation	muss andere Personen in die Gruppe einladen können, um mit diesen über gruppeninterne Themen kommunizieren und kollaborieren zu können	1
Nutzer: Gläubiger	Anrecht	Geld	muss einen Kassenzettel zur Verfügung haben, um eine Abrechnung beweisen und Geld vom Schuldner einfordern zu können	1

	Anspruch	Daten/ Kommunikation	müssen zu jedem Zeitpunkt auf seine Kassenzettel zugreifen können, um Abrechnungen mit anderen Gruppenmitgliedern erstellen zu können	1
		Daten/ Kollaboration	muss die Einkaufsliste mit dem Kassenzettel vergleichen können, um aus den Überschreidungen/Differenzen eine Abrechnung erstellen zu können	1
		Kommunikation, Kollaboration	muss andere Gläubiger zu der Abrechnung hinzufügen , falls diese Geld beigeleistet haben, um die Korrektheit der Abrechnung zu gewährleisten	1
			muss Schuldner zu der Abrechnung hinzufügen können, um Geld zurückgestattet zu bekommen und die Korrektheit der Abrechnung zu verifizieren	1
	Interesse	Kommunikation	muss den Artikeln Schuldner zuordnen können, um mit diesen hinsichtlich der Abrechnung darüber kommunizieren zu können	1

			muss die gewünschte Art der Währung und Transaktionsweise den Schuldner mitteilen, um die Abrechnung durchführen zu können	1
		Bezahldienste	muss Zahlungen über einen Online-Bezahldienst annehmen können, um Abrechnungen von einem Schuldner zu erhalten, der nicht am selben Ort ist oder diese Art der Zahlung präferiert	1
Nutzer: Schuldne	Anrecht	Daten/ Kommunikation	muss den Original-Kassenzettel des Gläubigers einsehen können, um die Korrektheit der Abrechnung zu verifizieren	1
Nutzer: Gruppen-ersteller	Anspruch	Kommunikation	muss andere Personen in die Gruppe einladen können, um mit diesen über gruppeninterne The men kommunizieren und kollaborieren zu können	1
Nutzer: Gläubiger	Anrecht	Geld	muss einen Kassenzettel zur Verfügung haben, um anhand diesen eine Abrechnung beweisen zu und Geld vom Schuldner einfordern zu können	1
	Anspruch	Daten/ Kommunikation	müssen zu jedem Zeitpunkt auf seine Kassenzettel zugreifen können, um Abrechnungen mit anderen Gruppenmitgliedern erstellen zu können	1

		Daten/ Kollaboration	muss die Einkaufsliste mit dem Kassenzettel vergleichen können, um aus den Überschreidungen/Differenz eine Abrechnung erstellen zu können	1
	Interesse	Kommunikation	muss dem Gläubiger Zahlungen über einen Online-Bezahldienst zukommen lassen können, um Abrechnungen abschließen zu können, falls dieser nicht mehr am selben Ort ist oder diese Zahlungsmethode präferiert wird	1
Einzelhandel	Anrecht	persönliche Informationen	vertraulicher Umgang mit Informationen	2
		Vertrag	Einhaltung der Vertragsbedingungen	2
	Anspruch	Kaufverhalten	anonymisierte Daten des Kaufverhaltens der Touristen	2
		Werbung	Schalten von Werbung anhand von Kaufdaten der Nutzer	2
	Interesse	Werbung	Anpassung der Werbung an Nutze	2
Währungstausch (currencylayer)		Einbettung/ Weiterleitung	Kundengewinnung	3
Bezahldienst (Paypal)		Einbettung/ Weiterleitung	Kundengewinnung	3

Tabelle 2: Stakeholderanalyse

Es sei zu erwähnen, dass die Stakeholderanalyse, in der Planungsphase bereits aufgestellt wurde und durch stetige Iterationen, während der Erfassung des Nutzungskontext vertieft wurde.

## **2.3 Aufgabenmodellierung**

### **2.3.1 Szenarien**

“Für eine ergonomische Diaologgestaltung müssen [...] aus den Gestaltungsgrundsätzen konkrete Kriterien unter der Berücksichtigung der Anwendung und der Zielgruppe abgeleitet werden.” [7] Diese Kriterien werden anhand von Szenarien entwickelt, die nach Heinecke Angaben zu

- Tätigkeit und Arbeitsaufgabe,
- Beabsichtigtes Arbeitsergebnis,
- Angestrebter Arbeitsaufwand am Bildschirmarbeitsplatz,
- Kontext der Nutzung,
- Eigenschaften der Benutzer,
- Eigenschaften des verwendeten Systems

enthalten. Um die Nähe zu der Zielgruppe und dem Nutzungskontext zu gewährleisten, werden hier die zuvor erstellten Personae verwendet. Die Eigenschaften der Personae und dem intendierten System werden dabei als bekannt vorausgesetzt.

Persona: William Taylor

Rolle im System: Gläubiger

Arbeitsaufgabe: Erstellen einer Gruppe und eines Gruppen-Kassenzettels

Es ist Sommer und William befindet sich in Österreich. Er hat sich mit vier weiteren Backpackern zu einer Wandertour in Kärnten verabredet. Die erste Route ist für fünf Tage angelegt und startet in Villach und verläuft dann über Klagenfurt am Wörthersee südlich entlang der Drau durch Slowenien und wieder hoch ins österreichische Graz. Die Strecke ist teilweise sehr bergisch und mit Wald und Gewässern versehen. Es werden aber auch mehrere Städte und Siedlungen durchquert und in Österreich ist fast durchgängig eine mobile Internetleitung (LTE) verfügbar. Jedes Mitglied installiert die Anwendung und registriert sich. William legt einen ersten Termin für eine Abrechnung fest, die in Graz nach fünf Tagen stattfinden soll. Danach lädt er alle Mitglieder in die Gruppe ein, die mit den festgelegten Bedingungen konfrontiert werden und mit der Annahme der Einladung den Bedingungen zustimmen. William bezahlt den ersten Einkauf und erhält einen Kassenzettel, in dem Proviant für zwei Tage enthalten ist. Er scannt den Einkaufszettel und erhält zum einen ein Bild von dem Kassenzettel und zum anderen einen digitalen Kassenzettel, in dem jeder Artikel mit Preis angegeben wird. Er überprüft den digitalen Kassenzettel kurz auf Korrektheit und Vollständigkeit und erstellt daraufhin einen Gruppen-Kassenzettel. In den Gruppen-Kassenzettel kommen alle Artikel, die von Schuldern beansprucht wurden. Das System verteilt die Kosten der im Gruppen-Kassenzettel befindlichen Artikel automatisch gleichermaßen auf alle Mitglieder auf. William kann nun in wenigen Minuten einzelne Artikel auswählen, die

für bestimmte Personen eingeplant sind und diese entsprechend zuordnen bzw. die Kostenverteilung adäquat anpassen. Er speichert den modifizierten Gruppen-Kassenzettel mit der Kostenverteilung ab und sendet ihn als Vorschlag an die Gruppenmitglieder. Er hat sich kaum anstrengen müssen und hat ein gutes Gefühl, dass eine gerechte Kostenverteilung vollzogen wird. Er widmet sich nun der Natur Österreichs, denn das System übernimmt das rechnen und die klare Vermittlung von Forderungen.

### 2.3.2 Erkenntnisse zum Nutzungskontext

Anhand der Recherchen und der deskriptiven Benutzungsmodelle können Probleme identifiziert werden, die in den Überlegungen während der Konzeptionierung kaum bis gar nicht berücksichtigt wurden. Im Folgenden stellen sich die Entwickler folgende Fragen:

- Internet: Wie geht das System mit schlechten Internet-Verhältnissen um?
- Einkauf: Es sind kaum Stationen für den Einkauf auf der Route, sodass ein großer Einkauf benötigt wird, der das Budget einer Einzelperson (Benutzer in der Rolle eines Gläubigers) übersteigt.
  - Kann das System den Benutzern das zumuten?
  - Inwieweit kann das System an dieser Stelle den Benutzer unterstützen?
- Abrechnung: Wie kann das System einem Benutzer garantieren, dass am Ende des Abrechnungszeitraums wirklich gezahlt wird?
- Einkaufszettel: Sollte ein Einkaufszettel digital entstehen und ins System integriert werden?
- Sprachbarriere im Einkauf: Wie geht das System mit (Fremd-) Produkten um, die einheimisch sind und für den Benutzer einen Fremdsprachigen Namen tragen?
  - Sollte das System darauf reagieren?
- Unfair behandelter Außenseiter: Sollte das System darauf reagieren, wenn sich eine Person ausgeschlossen und unfair behandelt fühlt von den übrigen Mitgliedern?
- Betrug: Ein Benutzer in der Rolle eines Gläubigers wird von Händlern offensichtlich betrogen. Sollen die Gruppenmitglieder das reklamieren können und ihren Anteil damit drücken?
- Fluktuation: Wie geht das System damit um, wenn eine hohe Dynamik des Ein- und Austretens in Gruppen stattfindet?
- Kommunikation: Was wird im technischen System kommuniziert und was in der realen Welt?
  - Wie geht das System mit Benutzern um, die im System integriert sind, aber jegliche Kommunikation dazu verweigern?

\* Welche Gründe gibt es dafür?

Anhand dieser Fragen wird überlegt, ob sich Paradigmen aus dem Nutzungskontext als Lösung anbieten bzw. sich auf das System übertragen ließen und ob bereits etablierte Lösungen in anderen Systemen existieren. Wenn keine Lösungen gefunden werden, fließen diese Probleme in die Stakeholderanalyse mit ein. Es ist möglich, dass eigene Schemata entwickelt werden müssen. Für die Entwickler scheinen Lösungsansätze für diese Probleme wichtig, um valide präskriptive Aufgabenmodelle erarbeiten zu können.

- Internet: Eine schlechte Internetverbindung würde sich vor allem negativ auf den Kommunikationsvorgang auswirken, der für eine kollaborative Erstellung eines für die Abrechnung tauglichen Gruppen-Kassenzettels nötig ist. Die Kommunikation muss dann dementsprechend technologiefrei zwischen den Beteiligten vor Ort stattfinden. Für diesen Fall benötigt das System einen Offline-Modus und eine Person, die stellvertretend für die Gruppe zeitabhängige Aufgaben erledigt. Für diesen Fall könnte sich ein Admin, der die benötigten Rechte erhält, eignen.
- Einkauf: Für den Fall, dass ein besonders großer Einkauf getätigt wird, liegt es nahe, dass die Personen vor Ort unmittelbar Geld zusammenlegen. Das System sollte für diese Situationen eine Gemeinschaftskasse zur Verfügung stellen, sodass der Wunsch danach berücksichtigt werden kann.
- Abrechnung: Mit dem Modell einer Gemeinschaftskasse könnte das System auch hier punkten. Eine Vorauszahlung aller Gruppenmitglieder an die Entwickler könnte einen vollständig korrekten Abrechnungsvorgang in jedem Fall garantieren. Die Entwickler könnten hier als starke Instanz fungieren, die Sicherheit und Zuverlässigkeit anbieten.
- Sprachbarriere im Einkauf: Um die Benutzer dabei zu unterstützen Nähe zu Fremdprodukten aufzubauen, sollten die gekauften Produkte bei der Erstellung des Gruppen-Kassenzettels Kategorien untergeordnet werden. So können die Benutzer die den Einkauf vollziehen und das Produkt genau betrachten können, den anderen Benutzern aus der Gruppe eine ungefähre Vorstellung von dem Produkt vermitteln.
- Ein unfair behandeltes Gruppenmitglied: Damit das System langfristig Benutzer erkennen kann, die fremde Gruppenmitglieder ausbeuten, könnte ein gut überlegtes Rating System entworfen werden, wie es aus etablierten Systemen wie bspw. den Apps für Mitfahrgemeinschaften bekannt ist. Als direkte Lösung kann erneut die Methode der Bildung einer Gruppenkasse genannt werden, die in diesem Beispiel eine Schadensminimierung erzielen könnte, falls das Budget einzelner Benutzer in der Abrechnung nicht überschritten werden kann.
- Betrug: Da das System auf der Grundlage von Kassenzetteln entwickelt wird, müssen sich alle Mitglieder an den dokumentierten Daten orientieren. Das drücken von Anteilen aufgrund von Betrugsvorwürfen zu Händlern ist nicht gültig und wird hier als unfaires Behandeln eines Gruppenmitglieds gesehen.

- Fluktuation: Das Beitreten von Benutzern soll grundsätzlich immer möglich sein. Das Austreten aus der virtuellen Gruppe, sollte erst nach dem Abrechnungszeitraum möglich sein. Für Ausnahmefälle könnte ein Admin die Berechtigung haben einem Mitglied den vorzeitigen Austritt zu ermöglichen.
- Kommunikation: Die Kommunikation findet überwiegend vor Ort statt. Das System hat dabei den Anspruch die reale Kommunikation durch Eindeutigkeit und Transparenz zu unterstützen. Benutzer, die sich an der Kommunikation im System nicht beteiligen, könnten nach dem Ermessen der Gruppenmitglieder durch ein Rating sanktioniert werden.

### 2.3.3 Hierarchical Task Analysis (HTA)

Um eine präzise Vorstellung von möglichen Handlungssequenzen und benötigten Interaktionsschritten zu bekommen, werden für die wichtigsten Aufgaben, bzw. den Aufgaben die der Grundidee und den Zielen des Projektes entsprechen, präskriptive Aufgaben Modelle erstellt. Diese werden als Hierarchical Task Analysis entworfen nach dem Muster von A. Dix, J. Finlay, G.D. Abowd und R. Beale aus dem Buch Human-Interaction Design.[8]

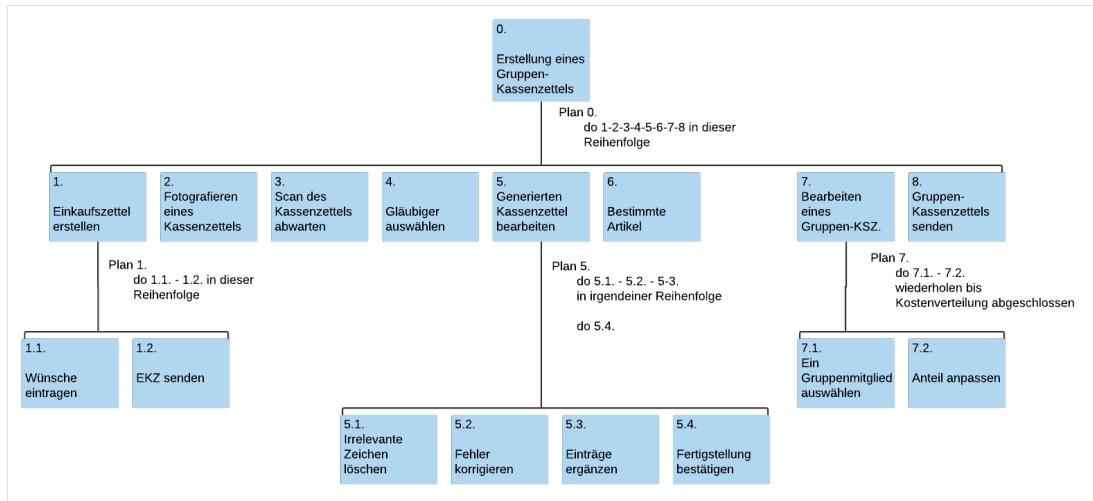


Abbildung 2: HTA, Erstellung eines Gruppen-Kassenzettel

In Abbildung 1. ist die Erstellung eines Gruppen-Kassenzettels zu erkennen. Nach den aktuellen Darstellungen des Systems sind bereits hier bestimmte Reihenfolgen von Interaktionsschritten nötig und andere möglich. Für das Erstellen von Gestaltlösungen muss abgewägt werden, welche Aufgaben nötig sind, um den Ziele und Anforderungen gerecht zu werden. Nimmt man die anderen priorisierten Aufgaben hinzu, das Erstellen einer Gruppe, Reaktion auf einen Kassenzettel, Vorbereitung einer Abrechnung, Abrechnen, durchführen eines Votings und Ratings, dann wird deutlich wie umfangreich und komplex das System wird. Die Sequenzen an Aufgaben, die für einen Abrechnungszyklus angedacht werden, könnten beispielsweise so dargestellt werden:

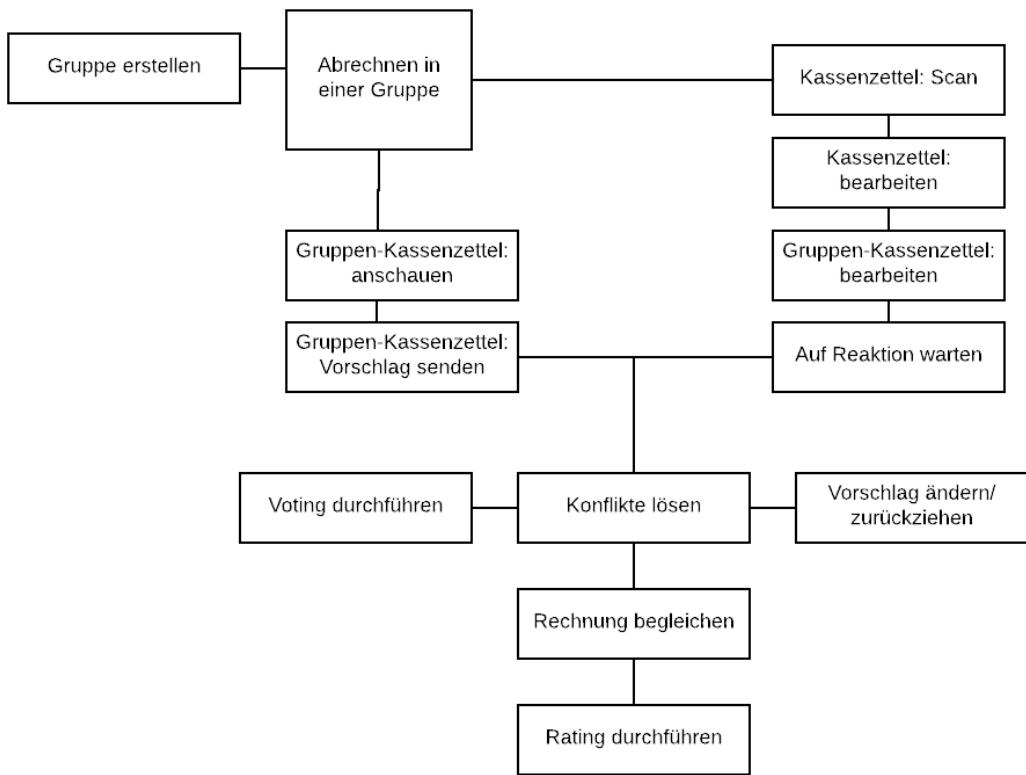


Abbildung 3: Aufgaben in einem Abrechnungszeitraum

### **3 Festlegen der Nutzungsanforderungen**

#### **3.1 Funktionale Anforderungen**

- Kollaborative Anpassung und Überarbeitung der Abrechnung soll möglich sein.
- Eine Abrechnung soll unmissverständlich auf gegenseitigem Einverständnis beruhen.
- Die Vertrauenswürdigkeit der Benutzer soll messbar gemacht werden
- Eine Gruppe aus Benutzern soll sich im System verknüpfen können.
- Kassenzettel sollen digitalisiert und die Informationen weiterverarbeitet werden können.
- Finanzielle Konflikte sollen im System deutlich erkennbar sein.

Der Benutzer...

- muss permanent auf technische Unterstützung beim Abrechnen zugreifen können.
- muss einen transparenten Überblick über gemeinsam erzeugte Kosten erhalten.
- muss permanenten Zugriff auf die von ihm erstellten Kassenzettel haben.
- muss Informationen erhalten oder beschaffen können, wer Zugriff oder Einsicht in die von ihm erstellten Kassenzettel hat.
- muss Informationen erhalten oder beschaffen können, auf welche Quelle sich die von anderen erstellten Kassenzettel beziehen und diese überprüfen können.
- muss Informationen erhalten oder beschaffen können, wer mit ihm kommuniziert oder interagiert.
- soll Informationen erhalten oder beschaffen können, wie sie ehemalige Gruppenmitglieder kontaktieren können.
- soll kollaborativ geplante Einkäufe notieren und speichern können.
- soll Informationen beschaffen können, zu Wechselkursen von Währungen haben.
- muss eine Gruppe erstellen können, in der er unabhängig vom Zeitpunkt und der Anzahl an Mitgliedern neue Personen einladen kann.
- soll finanzielle Konflikte transparent und frühzeitig erkennen können.

Der Benutzer in der Rolle eines Gläubigers...

- muss mitteilen oder darstellen können, welche Gläubiger und Schuldner einer Abrechnung zugeordnet sind.
- muss mitteilen oder darstellen können, zu welchen Artikeln/Produkten welche Schuldner gehören.
- soll mitteilen oder darstellen können, in welcher Form er bezahlt werden möchte.

Der Benutzer in der Rolle eines Schuldners...

- muss ein Beleg sehen dürfen, wenn von ihm eine Zahlung gefordert wird.
- muss mitteilen oder darstellen können, welche Änderungen er an einer Abrechnung vornehmen will.
- soll mitteilen oder darstellen können, dass er zu einem bestimmten Zeitpunkt nicht bezahlen kann.

### **3.2 Qualitative Anforderungen**

- Das Vertrauensverhältnis zwischen Nutzern und dem System soll sich stetig verbessern.
- Vorgefertigte Dialoge in Worten, Sätzen oder bildlichen Darstellungen sollen eine einfache eindeutige Kommunikation unterstützen.
- Die Integrität sensibler Daten muss gewährleistet sein.

### **3.3 Organisatorische Anforderungen**

- Unterschiedliche Sprachen und Währungen sollen beim Kommunikationsfluss und Abrechnungsvorgang behandelt werden.
- Kosten sollen abschnittsweise, nach individuellen Vorstellungen, abgerechnet werden können.
- Das System soll durch mobile Endgeräten erreicht werden können.

Der Benutzer...

- soll in seiner Erstsprache Dialoge führen können.
- muss ein Selbstbestimmungsrecht haben, aus einer Gruppe austreten zu dürfen, wenn keine Rechnungen offen sind.

Der Benutzer in der Rolle eines Gläubigers...

- muss einen gültigen Beleg zur Verfügung haben, wenn er von Gruppenmitgliedern Geld fordert.

Der Benutzer in der Rolle eines Schuldners...

- muss ein Beleg sehen dürfen, wenn von ihm eine Zahlung gefordert wird.

## 4 Erarbeiten von Gestaltungslösungen

Für die Erarbeitung von Gestaltungslösungen, wurde vom geplanten Projektplan abgewichen. Aus der Aufgabenmodellierung und den Anforderungen wird deutlich, dass die Entwicklung des Systems sowohl in die Breite, als auch in die Tiefe erweitert werden kann. Um den Erfordernissen der Benutzer gerecht werden zu können, muss eine Struktur entworfen werden, in der das Layout und die Navigation konzeptioniert werden.

In der Diskussion über Gestaltungslösungen wird von den Entwicklern zunächst überlegt, wie das Interface designt werden soll. Da die Handlungssequenzen auf dem Scan eines Kassenzettels basieren, der sichtbare Information enthält, liegt es für die Entwickler nahe eine rein visuelle Benutzerschnittstelle zu gestalten. Für ein Interface, dass auditives Design mit einbezieht ist der Nutzungskontext ungeeignet. Benutzer sind in vielen Fällen von Umgebungsgeräuschen beeinflusst. Da die Anwendung in Android entwickelt wird, richten sich die Entwickler beim Design neben den vorgestellten ISO Normen auch nach den Prinzipien der Developer Web Page von Android [9] und den global etablierten Guidelines des Material Design von Google [10]. Im Folgenden wird auf die neu geplante Struktur und schließlich auf die detaillierte visuelle Gestaltung eingegangen.

### 4.1 Struktur

Bevor die Strukturen der Anwendung festgelegt werden, wird noch einmal bedacht, dass bei den Untersuchungen der Benutzern kulturelle Unterschiede identifiziert wurden. Daher werden vor allem international und global bekannte Design-Muster bevorzugt, sofern sie mit den Anforderungen vereinbar sind und die Kreativität der Entwickler nicht einschränken.

#### 4.1.1 Dialoggestaltung

In der ISO 9241 Teil 110 sind sieben Grundsätze der Dialoggestaltung definiert. Jede dieser Prinzipien wird von Empfehlungen und Beispielen begleitet. Aus Zeitgründen war es jedoch bisher nicht möglich auf feingranularer Ebene die Empfehlungen im Einzelnen in das Design mit einzubeziehen. Daher muss an dieser Stelle auf Iterationen verwiesen werden, in denen man hier skaliert werden könnte.

- Aufgabenangemessen
  - Die Form von Zahlen und Ziffern, wie Geldbeträge, richten sich nach der Schreibweise der ausgewählten Sprache.
  - Der Benutzer wird informiert, wenn er den Scan des fotografierten Kassenzettels abwarten muss.
  - Der Benutzer wird gewarnt, wenn ein finanzieller Konflikt vorhanden ist.

- Selbstbeschreibungsfähigkeit
  - Für die Buttons werden Symbole ausgesucht, die global und international verständlich sind und wenn möglich, aus anderen Anwendungen bereits bekannt sind.
  - Für die Eingabe von z.B. dem Datum und Geldbeträgen gibt die Anwendungen Vorgaben zum Format der ausgewählten Sprache an.
  - Wenn ein Benutzer auf ein Ereignis reagieren kann, wird er deutlich zur Interaktion aufgefordert.
- Erwartungskonformität
  - Die Gesamtstruktur und Navigation, sowie die Symbole sollen den von anderen Apps ähneln, damit die Benutzer einen Wiedererkennungswert haben und die Qualität der Benutzung schnell verbessern können.
- Lernförderlichkeit
  - Artikel können in Kategorien untergeordnet werden, sodass Benutzer unbekannte Fremdprodukte zuordnen können.
- Steuerbarkeit
  - Der Benutzer soll die Möglichkeit haben zu jedem Punkt navigieren zu können, unabhängig wo er sich befindet.
- Fehlertoleranz
  - Der Benutzer soll die Möglichkeit haben einen Scan manuell zu überarbeiten.
  - Wenn Pflichtfelder nicht ausgefüllt werden, wird der Benutzer dazu aufgefordert diese auszufüllen.
- Individualisierbarkeit
  - Es soll in die Bilder der Kassenzettel gezoomt werden können.

#### **4.1.2 Navigation**

Die Navigation soll dem Benutzer ermöglichen, schnellstmöglich jede Aktivität ausführen zu können, unabhängig davon, wo er sich in der Anwendung zu diesem Zeitpunkt befindet. Um logische Handlungssequenzen zu ermöglichen, müssen verschiedene Komponenten zur Navigation entwickelt werden. Um so mehr Richtungen in einer Komponente angesteuert werden können, desto besser wirkt sich das auf die Steuerbarkeit aus. Im Umkehrschluss darf es sich allerding nicht negativ auf die Übersichtlichkeit auswirken. Aus diesen Gründen und in Anbetracht des erweiterbaren Umfanges von möglichen Handlungssequenzen, wird mit einem Menüsystem geplant, welches aus einem "Navigation Drawer"[11] und einer "Bottom Navigation Bar"[12] besteht, um hier eine Balance

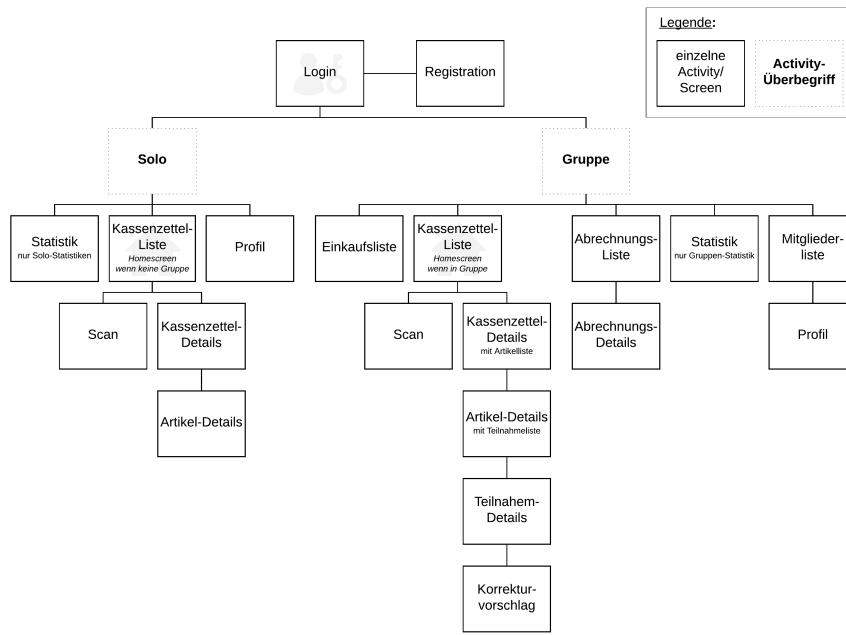


Abbildung 4: Flowchart der Hauptaktivitäten

zu finden. Ausgangspunkt der Navigation soll für einen im System registrierten und angemeldeten Benutzer die Übersicht zu den generierten Kassenzetteln sein, wobei hier differenziert werden muss, ob er sich zur Zeit in einer Gruppe befindet oder nicht. An dieser Stelle besteht Diskussionsbedarf, ob der Benutzer in die Übersicht der Gruppen-Kassenzettel startet, wenn er sich in einer Gruppe befindet. Oder erst auswählen kann, ob er in der persönlichen Umgebung oder der Gruppen-Umgebung starten möchte. Oder ob er erst aus der persönlichen Umgebung in die Gruppen-Umgebung navigieren muss. Da der Benutzer sich, zu einem Zeitpunkt, nur in einer Gruppe befinden kann und die Prioritäten des Systems auf eine Interaktion zwischen Gruppen-Mitgliedern basiert, wird entschieden, dass der Benutzer automatisch in der Übersicht der Gruppen-Kassenzettel startet, falls er sich in einer Gruppe befindet. Befindet er sich in keiner Gruppe, startet er dementsprechend bei den persönlichen Kassenzetteln. Bevor im Layout genauer auf die Navigations-Komponenten eingegangen wird, muss berücksichtigt werden, dass der Benutzer sich von einem Ausgangspunkt zwischen Start und Ziel, in einer Handlungssequenz, vor und zurück bewegen können muss. In Abbildung 3 wurde ein Flowchart für alle Hauptaktivitäten der Anwendung dargestellt. Der Benutzer startet beim Login bzw. bei der Registration, wenn er die Anwendung zum ersten mal verwendet. Nach einer erfolgreichen Anmeldung kann startet er, wie bereits beschrieben, in einem der beiden Homescreens, jenachdem ob er in einer Gruppe ist oder nicht. Von hier aus werden die jeweiligen Handlungssequenzen versucht darzustellen. Am Beispiel der Kassenzettelliste bietet sich die Möglichkeit, entweder einen neuen Zettel zu scannen und der Liste hinzuzufügen, oder weiter in die Kassenzetteldetailhierarchie einzutauchen. Dabei soll

eine Flache Aktivitäten-Hierarchie beibehalten werden, sodass der Benutzer mit wenigen Interaktionsschritten, zwischen allen Hauptaktivitäten wechseln kann. Jedoch ist den Entwicklern dabei aufgefallen, dass die Hierarchie der Gruppen-Kassenzetteldetails sehr verschachtelt ist. Im Wireframe "Kassenzettel-Details" wurde versucht dies zu verbessern und die Hierarchie flacher zu gestalten.

## 4.2 Seitenlayout

### 4.2.1 Navigation

**Layout:** Das persistente Element der Navigation, welches in jeder Aktivität des Clients eingesetzt wird, ist die Toolbar. Das rechte Icon soll den Benutzer auf eingehende Push-Nachrichten aufmerksam machen. Ein Klick darauf zeigt die Nachrichten an und leitet, anhand deren Inhalts, in die jeweilige Aktivität weiter. Der Titel beschreibt immer den Namen der, sich im Vordergrund befindenden, Aktivität. In den Hauptaktivitäten bein-



Abbildung 5: Toolbar

haltet das linke Symbol ein Hamburger-Menü, welches den "Navigation Drawer" öffnet. Von dort können alle Hauptaktivitäten, die Einstellungen und der Logout mit einem Klick angesteuert werden. Falls sich der Nutzer in keiner Gruppe befindet, werden ihm die Gruppen-Aktivitäten nicht angezeigt und er kann nur eine neue Gruppe erstellen. Das runde Interaktionselement stellt das Profilbild des Nutzers dar, welches auf sein persönliches Profil weiterleitet. Wenn sich der Nutzer in einer Detail-Aktivität (z.B Kassenzetteldetails, Abrechnungsdetails, etc) befindet, dann wird das Hamburger-Menü mit einem Zurück-Pfeil ersetzt. Mit diesem Element kann der Nutzer zurück in die jeweilige Hauptaktivität navigieren.

Falls sich der Nutzer in den Gruppen-Aktivitäten befindet, dann wird zusätzlich noch eine "Bottom Navigation Bar" angezeigt, die persistent bei den drei wichtigsten Gruppen-Aktivitäten (Einkaufsliste, Kassenzettelliste, Abrechnungsliste) verfügbar ist. So kann der Benutzer mit einem Klick zwischen den wichtigsten Gruppen-Aktivitäten wechseln

#### Prinzip/Begründung:

Der Nutzer sollte zu jedem Zeitpunkt mit wenigen Interaktionsschritten (höchstens drei bis vier) zu jeder Aktivität seiner Wahl navigieren können. Daher wurde der "Navigation Drawer" gewählt. So ist es dem Benutzer möglich, aufgrund der Ähnlichkeit zu



Abbildung 6: NavigationDrawer mit Gruppe

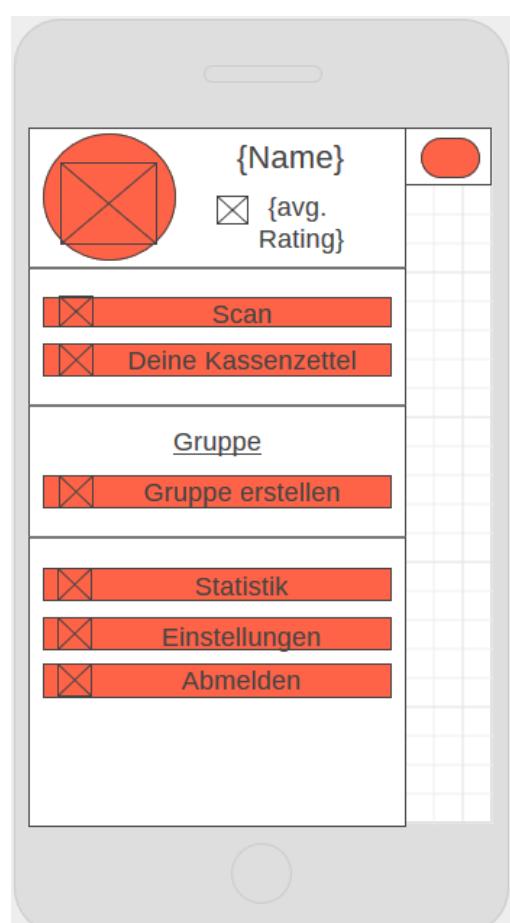


Abbildung 7: NavigationDrawer ohne Gruppe

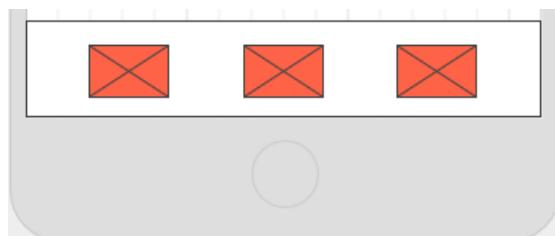


Abbildung 8: Bottom Navigation Bar

einem Inhaltsverzeichnis, diesen intuitiv und effizient zu nutzen. In seiner zusammengeklappten Form nimmt der Navigation Drawer zusätzlich keinen Platz ein und das Design bleibt so übersichtlich. Der Nutzer muss außerdem jederzeit auf eingehende Push-Benachrichtigungen reagieren können. Daher wurde der fixe Platz in der Toolbar gewählt.

So ist er immer für den Nutzer einseh- und interagierbar. In den Gruppen-Aktivitäten wurde die “Bottom Navigation Bar” gewählt, da die Entwickler davon ausgehen, dass die drei Interaktionselemente die meistgenutzten Aktivitäten für den tagtäglichen Gebrauch darstellen. Die Nutzer können so mit dem Daumen sehr schnell zwischen den Gruppen-Aktivitäten wechseln, ohne den “Navigation-Drawer” überhaupt verwenden zu müssen

**Szenario:**

1. Der Nutzer erhält eine Push-Benachrichtigung über eine aktualisierte Abrechnung. Ein Klick auf das Icon zeigt ein Dropdown-Menü der letzten Nachrichten, welche absteigend nach dem Zeitpunkt ihrer Ankunft geordnet sind. Ein Klick auf die Nachricht leitet den Nutzer zu der Aktivität weiter.
2. Der Nutzer möchte von der Liste der Gruppen-Kassenzettel in die Liste seiner persönlichen Kassenzettel wechseln. Ein Klick auf das Hamburger-Menü öffnet den “Navigation Drawer” und ein weiterer Klick das Element “Deine Kassenzettel” leitet den Benutzer zu dem Ziel weiter.
3. Der Nutzer will einen Einkauf planen und von der Liste der Gruppen-Kassenzettel zur Einkaufsliste wechseln. Ein einfacher Klick auf das Kassenliste-Symbol in der “Bottom Navigation Bar” leitet sofort weiter.

**Prüfkriterien:**

Werden die richtigen Elemente angezeigt, wenn der Nutzer in einer Gruppe ist bzw. nicht ist? Wird der richtige Name und Rating im “Navigation Drawer” angezeigt? Leiten die Interaktionselemente in die jeweiligen Aktivitäten? Wird das Symbol der Push-Nachrichten bei eingehenden Nachrichten aktualisiert? Zeigt es die Nachrichten an?

#### 4.2.2 Abrechnungsübersicht

##### Layout:

Dies ist der Startbildschirm der Gruppen-Aktivitäten. Fokus liegt auf der aktuellen Abrechnung der Gruppe, welche eine Übersicht der wichtigsten Daten, wie Abrechnungszeitraum, Anzahl der involvierten Mitglieder und Anzahl der involvierten Kassenzettel anzeigt. Außerdem zeigt es das aktuelle Schuldverhältnis des angemeldeten Benutzers als positiver Betrag, wenn er Gläubiger ist bzw. als negativer Betrag, wenn er Schuldner ist. Diese Angaben sind für eine schnelle Identifikation des Abrechnungsstandes nötig. Darunter ist eine Liste alter Abrechnungen mit Markierung, ob bereits Geld erhalten/bezahlt wurde, vorhanden. Ein Klick auf eine Abrechnung leitet zu der dazugehörigen Detailübersicht weiter. Der Footer leitet auf zusätzliche Aktivitäten der Gruppe (Kassenzettel-Liste, Einkaufsliste) weiter.

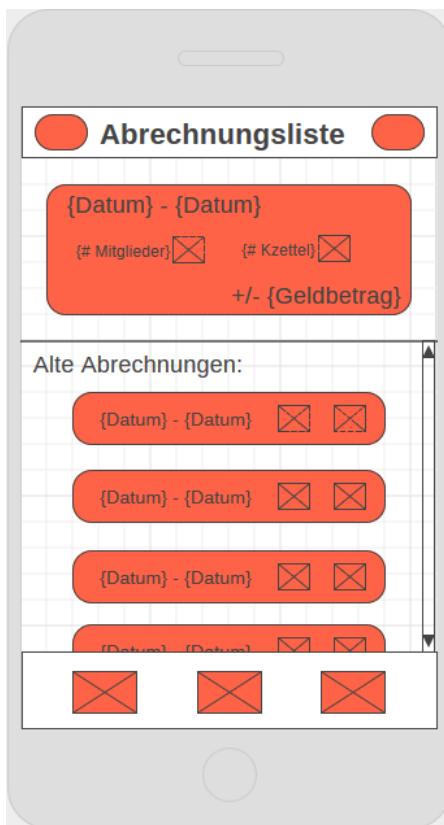


Abbildung 9: Abrechnungsliste

##### Prinzip/Begründung:

Nutzer wollen höchstwahrscheinlich mit der aktuellsten Abrechnung interagieren oder den allgemeinen Status einsehen. Der erste Blick fällt auf den oberen Bereich des Lay-

outs. Alte Abrechnungen mit geringerer Priorität, sind dennoch schnell und mit wenig Interaktion im unteren Bereich des Layouts einsehbar. Anhand des Footers ist eine schnelle Navigation innerhalb der Gruppenfeatures möglich.

**Szenario:**

1. Der Nutzer will überprüfen, wie tief er bei der aktuellen Abrechnung in der Schuld steht. Der persönliche, System dynamisch berechnete, Geldbetrag kommt dem Nutzer nicht richtig vor. Falls er die Detailübersicht, über verrechnete Kassenzettel etc., einsehen oder in der Abrechnung eine Änderung vorschlagen will, leitet ein Klick darauf den Nutzer in die Detailübersicht weiter.
2. Der Nutzer will überprüfen, in welchen alten, beendeten Abrechnungen er noch Geld bezahlen bzw. erhalten muss. Ein Blick auf die Liste "Alte Abrechnung" zeigt ihm, in welchen Abrechnungen die Bezahlung noch nicht abgeschlossen wurde. Ein Klick auf die Abrechnung leitet in die Detailübersicht weiter.

**Prüfkriterien:**

Leiten die Buttons in die korrekten Aktivitäten weiter? Werden die korrekten Daten in den Interaktionselementen der Abrechnungen angezeigt?

#### 4.2.3 Abrechnungdetails

**Layout:**

Dies ist die Detailübersicht einer Abrechnung. Jedes Gruppenmitglied, dass in der Abrechnung partizipiert, wird hier, mit allen Daten, die zur Verifikation der Abrechnung nötig sind, aufgelistet. Fokus liegt auf den individuellen Abrechnungsdaten des angemeldeten Nutzers, fixiert im oberen Bereich des Layouts. Weiter unten ist eine Liste der restlichen Mitglieder mit ihren Daten. Da Mitglieder jedoch nur zu eigenen Daten Änderungsvorschläge/Korrekturen einreichen können, sollte diese immer sofort einsehbar erreichbar sein. Ein Klick auf das Feld des jeweiligen Mitglieds klappt eine Liste, aller an ihn referenzierten Kassenzettel, auf. Ein Klick auf den Kassenzettel leitet den Nutzer in die Detailansicht des jeweiligen Zettels weiter.

**Prinzip/Begründung:**

Nutzer wollen höchstwahrscheinlich Details ihrer eigenen Abrechnungsdaten einsehen oder ändern. Jedes Mitglied sollte jedoch aufgrund der gewünschten Transparenz alle Abrechnungsdaten der anderen Mitglieder einsehen können und sich so eine Meinung über die Korrektheit der Abrechnung bilden können (hinsichtlich Voting). Es wurde eine Dropdown-Liste für die Übersicht der Kassenzettel gewählt, da es bereits eine eigene Aktivität für eine Auflistung aller Kassenzettel einer Gruppe gibt und so einer möglichen Verwirrung zwischen diesen Beiden vorgebeugt wird.

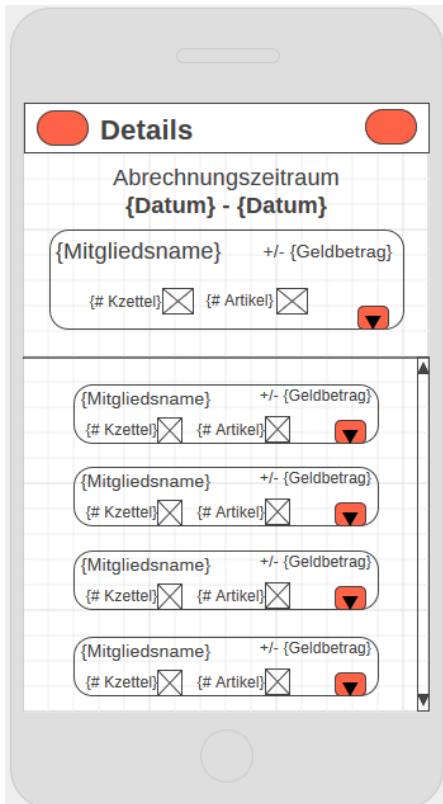


Abbildung 10: Abrechnungsdetails zusammengeklappt

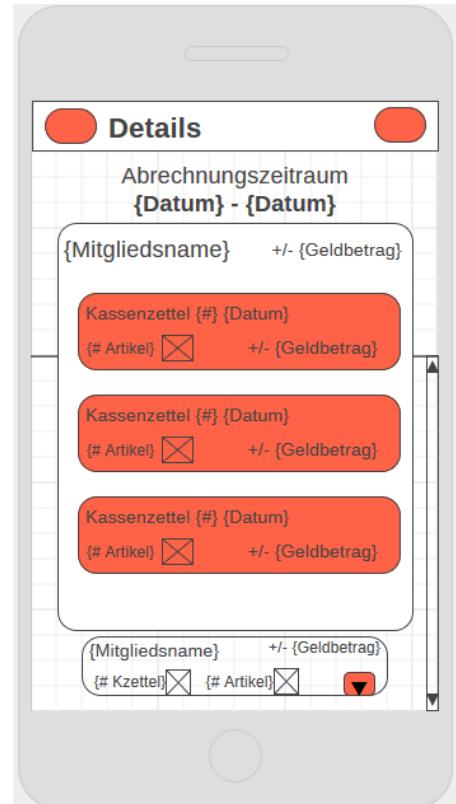


Abbildung 11: Abrechnungsdetails aufgeklappt

### Szenario:

1. Der Nutzer ist Schuldner in einer Abrechnung. Er will überprüfen für welche Einkäufe er Geld bezahlen muss. Ein Klick auf das Interaktionselement mit dem Namen des Nutzers klappt die Liste mit der an ihn referenzierten Kassenzettel mit den zusammengefassten Daten. Die Übersicht beinhaltet für den Benutzer noch nicht genügend Daten. Ein Klick auf einen Kassenzettel leitet den Nutzer zu der Detailübersicht des jeweiligen Kassenzettels weiter.
2. Ein Benutzer will sehen, wie sich seine Gruppenmitglieder an gemeinschaftlichen Einkäufen beteiligt haben. In der Listen sieht er die zusammengefassten Beteiligungs-Daten der restlichen Nutzer. Ein Wert scheint nicht korrekt zu sein, daher klappt der Benutzer die Liste auf und überprüft die verlinkten Kassenzettel

### Prüfkriterien:

Stimmt der Abrechnungszeitraum? Sind die zusammengefassten Daten der einzelnen Mitglieder korrekt? Ist der angemeldete Benutzer ganz oben angeheftet? Kann man Liste der referenzierten Kassenzettel aufklappen? Leiten die Kassenzettel-Elemente zu den Detail-Aktivitäten des jeweiligen Kassenzettels weiter?

#### 4.2.4 Gruppen-Kassenzettel (Liste)

**Layout:** In dieser Aktivität werden alle Kassenzettel aufgelistet, welche von Mitgliedern einer Gruppe in dieser verlinkt werden. Die Kassenzettel, welche vom angemeldeten Benutzer in der Gruppe referenziert wurde, sind ganz oben angepinnt. Die Kassenzettel der restlichen Mitglieder werden darunter aufgelistet. Die einzelnen Elemente enthalten eine Auswahl an Daten aus dem jeweiligen Kassenzettel. Ein Klick auf den diese leitet in dessen Detail-Aktivität weiter.

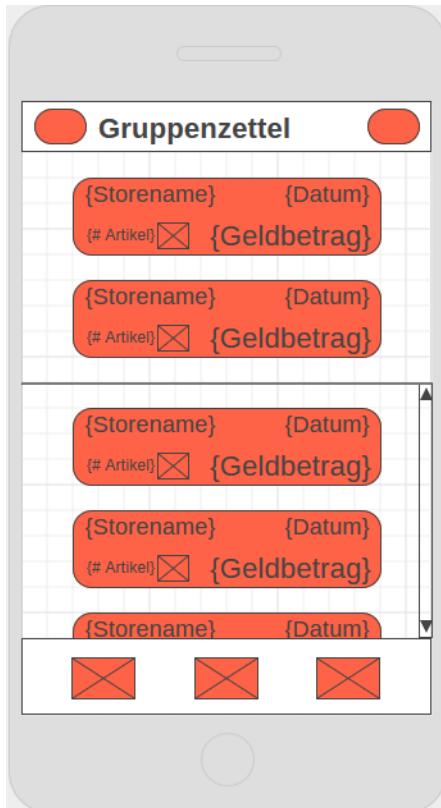


Abbildung 12: Gruppenkassenzettel-Liste

#### Prinzip/Begründung:

Wie bereits in den Abrechnungs-Aktivitäten, werden auch hier die, zum angemeldeten Benutzer gehörenden, Elemente priorisiert und oben angeheftet. Dies dient der besseren Übersichtlichkeit und Effizienz. Zudem ist die Wahrscheinlichkeit groß, dass der Nutzer seine eigenen Kassenzettel einsehen will, weil nur dieser die Daten ändern darf. Die Zettel der restlichen Mitglieder sind dennoch schnell und einfach erreichbar. Die Daten, der einzelnen Kassenzettel, wurden gewählt, weil sie die schnelle, eindeutige Identifikation ermöglichen.

**Szenario:**

1. Der Nutzer will Daten seiner Kassenzettel ändern, weil er gemerkt hat, dass er ursprünglich falsche Werte bei den Teilnahmen eingetragen hat. Ein Klick auf den Kassenzettel leitet in die Detailansicht des Kassenzettels weiter, wo die Daten manuell editierbar sind.
2. Ein Nutzer, der noch keinen Kassenzettel hochgeladen hat, will den zuletzt hinzugefügten Kassenzettel anschauen. Es sind also keine Zettel im oberen Bereich angeheftet. Per default werden die Kassenzettel nach Datum (neu-alt) sortiert. Er klickt also auf den ersten Eintrag der Liste und wird somit auf den aktuellsten Kassenzettel weitergeleitet.

**Prüfkriterien:**

Werden die Kassenzettel des angemeldeten Benutzers oben angeheftet? Werden alle Kassenzettel der Gruppe angezeigt, insofern diese nicht gelöscht wurden? Werden in den einzelnen Kassenzettel-Elementen die richtigen Daten angezeigt? Leitet das Kassenzettel-Element zu der korrekten Detailansicht weiter?

#### 4.2.5 Kassenzettel-Details

**Layout:** In der Detailansicht eines Kassenzettels werden im oberen Bereich des Layouts Daten angepinnt, die für eine eindeutige Identifikation nötig sind. Ein Klick auf das Element, rechts neben Stadt, könnte auf eine externe Karten-Anwendung verlinken, welche dann die Adresse auf einer Karte anzeigt. Sobald man durch die Liste der Artikel scrollt, könnte das Element aus dem Bild verschwinden und wieder erscheinen falls der Benutzer wieder nach oben scrollt. Im unteren Bereich des Layouts sind Daten, wie Gesamtbetrag, bezahlter Betrag und Wechselgeld, angegeben. Dieses Element bleibt immer sichtbar. In der Liste der Artikel werden alle Artikel des Kassenzettels angezeigt. Jedes Artikel-Element beinhaltet Daten wie Name/Bezeichnung, Gesamtpreis, Menge und Einzelpreis. Zusätzlich eine Zahl, welche die, an diesem Artikel, beteiligten Mitglieder symbolisiert. Falls der Benutzer auf einen Artikel klickt, wird das Element erweitert und die einzelnen Teilnehmer werden angezeigt. Es wird der Name und die prozentuale Beteiligung an dem Produkt sichtbar gemacht. Der Gläubiger bzw kann mit einem Klick auf Beteiligung diesen Wert manuell ändern. Ein Klick auf das Interaktionselement rechts neben der Beteiligung eines Teilnehmers kann der ganze Eintrag gelöscht werden. Mit einem Klick auf das Interaktionselement unter den Teilnahmen kann eine neue Teilnahme hinzugefügt werden. Schuldner können zu Artikeln, an denen sie teilgenommen haben Änderungen vorschlagen. Wie dies aussehen könnte zeigt folgendes Wireframe. Hier hat der Teilnehmer Rando R. vorgeschlagen seine Beteiligung zu löschen. Daraufhin hat das System automatisch die Beteiligung der anderen Teilnehmer korrigiert um bei 100 Prozent zu bleiben. Der Vorschlag bleibt für alle Mitglieder einsehbar bis der Gläubiger ihn akzeptiert oder ablehnt.

**Prinzip/Begründung:**

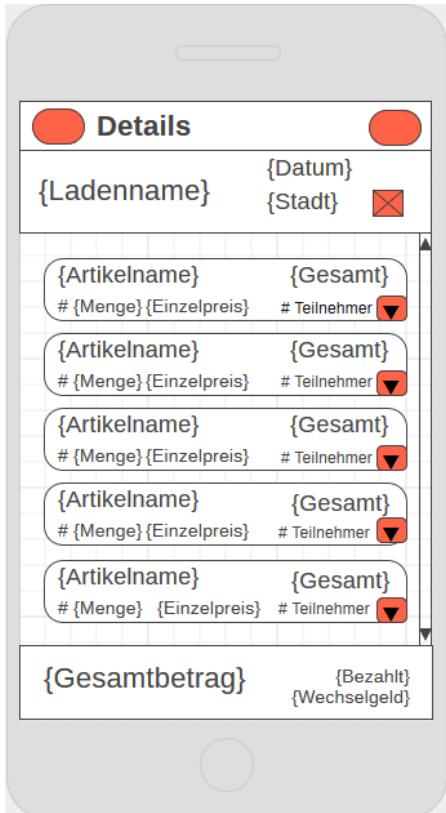


Abbildung 13: Kassenzetteldetails zusammengeklappt

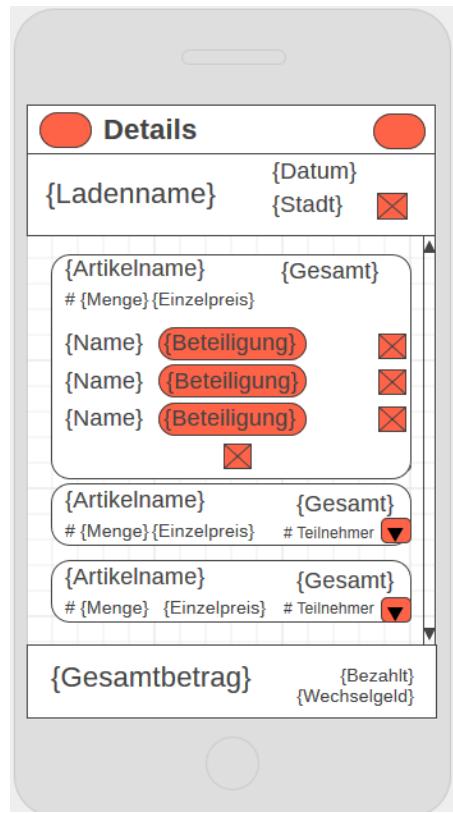


Abbildung 14: Kassenzetteldetails aufgeklappt

{Artikelname}	{Gesamt}
# {Menge} {Einzelpreis}	
Jan M.	50%
Armin W.	25% 50%
Rando R.	25% 0%

Abbildung 15: Vorschlag einer Änderung

Die Identifikationsdaten im oberen Bereich des Zettels wurden dort positioniert um den Nutzer, mit dem ersten Blick, die Prüfung, ob er die korrekte Detailansicht vor sich hat, zu ermöglichen. Wenn er mit der Interaktion des Kassenzettels fortführt, kann davon ausgegangen werden, dass die Identifikation erfolgreich war und die Daten können ausgeblendet werden. Die Daten des Gesamtbezahls im unteren Bereich bleiben jedoch persistent dort fixiert, da es dem Nutzer ermöglicht werden soll zu jeder Zeit einzelne Artikel, oder die Artikel in ihrer Gesamtheit, mit dem Gesamtbezahlgut zu vergleichen. Es

wurde abgewogen, ob für die Darstellung der Teilnahme an Artikel, eine eigene Aktivität verwendet werden sollte. Der Vorteil wäre eine klare Abgrenzung und einfachere Übersicht der Daten. Der Nachteil wäre jedoch eine komplexere Navigation und der Nutzer hätte nicht mehr alle Informationen in einer Aktivität, so dass dieser gezwungen wäre mehrmals zwischen Aktivitäten zu springen. Daher wurde entschieden, dass Teilnahmedaten zusammen mit Artikeldaten abgebildet werden sollen. Um dennoch einen Kompromiss für eine klare Abgrenzung und Übersichtlichkeit der Daten zu finden, wurden für die Artikel erweiterbare Elemente verwendet. So wird der Benutzer auf den ersten Blick nicht mit Informationen überflutet, kann aber sehr schnell gewünschte Teilnahmedaten abrufen und bearbeiten.

**Szenario:**

1. Der Nutzer will sich eine Übersicht verschaffen, welche Artikel beim letzten gemeinschaftlichen Einkauf gekauft wurden und wer an diesen teilgenommen hat. Dafür scrollt er durch die Liste der Artikel und klickt auf diejenigen, bei denen er Details über Teilnehmer und Kostenaufteilung einsehen will.
2. Ein Gläubiger hat gemerkt, dass ein Teilnehmer gar nicht an einem Artikel teilgenommen hat, wie vorher angenommen. Er sucht den betroffenen Artikel in der Liste, öffnet mit einem Klick darauf die Teilnehmerübersicht und löscht mit einem weiteren Klick auf das Interaktionselement, rechts neben dem zu löschen Benutzer, diesen aus dem Artikel. Das System rechnet die aktuelle Kostenverteilung neu aus. Der Gläubiger überprüft mit einem Blick auf die Beteiligung der Mitglieder, ob das neue Teilnahme-Verhältnis stimmt. Falls dies nicht der Fall ist, editiert er die Werte der verbleibenden Teilnehmer manuell.
3. Ein Schuldner ist mit der Kostenverteilung der aktuellen Abrechnung nicht zufrieden. Er wurde aus der Detailansicht der aktuellen Abrechnung zu der Detailansicht des neuesten Kassenzettels weitergeleitet und will dort eine Änderung vorschlagen. Er klickt auf einen Artikel, an dem er sich beteiligt hat, klickt auf den Eintrag mit seinem Namen und editiert den Wert. Der neue Wert steht nun rechts neben dem alten. Der Schuldner wartet nun auf die Reaktion des Gläubigers.

**Prüfkriterien:**

Sind die Daten zur Identifikation des Kassenzettels im oberen Bereich des Layouts angepinnt? Sind die Daten des Gesamtbetrags im unteren Bereich des Layouts angepinnt? Sind alle Artikel des Kassenzettels in der Liste vorhanden? Sind die einzelnen Artikel aufklappbar und zusammenklappbar? Sind Teilnahmewerte editierbar? Kann man einzelne Teilnahmen löschen und neue erstellen? Werden überall die korrekten Daten angezeigt?

#### 4.2.6 Einkaufsliste

**Layout:** In der Einkaufsliste sollen alle Mitglieder Artikelwünsche für den nächsten Einkauf eintragen können. An erster Stelle des Layouts ist der Gesamtbetrag des Budgets, der aus dem Budget aller eingetragenen Mitglieder für den Einkauf berechnet wird, angepinnt. Darunter werden die Wünsche des aktuell angemeldeten Benutzers und sein Budget angezeigt. Der Benutzer kann den Artikelwunsch und die benötigte Menge manuell in das editierbare Feld eintragen und es mit dem Interaktionselement daneben speichern. Gespeicherte Wünsche können mit dem Interaktionselement neben dem jeweiligen Eintrag wieder gelöscht werden. Die Wünsche der restlichen Mitglieder sind nach dem selben Layout in einer Liste im unteren Bereich angesiedelt, jedoch ohne Interaktionselemente.

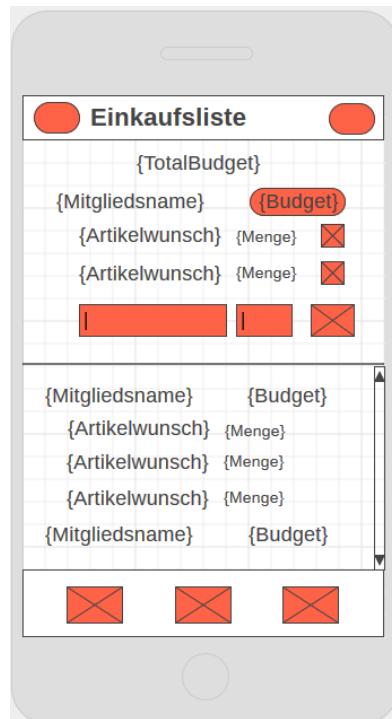


Abbildung 16: Einkaufsliste

#### Prinzip/Begründung:

Der Betrag für das gesamte Budgets des nächsten Einkauf wurde ganz oben angesiedelt, so dass die Mitglieder diesen auf den ersten Blick wahrnehmen und sich eine Vorstellung machen können, welchen Umfang der Einkauf haben könnte. Zudem kann man so schon im voraus planen ob der Einkauf eventuell zu teuer für eine einzige Person wird und der zukünftige Gläubiger finanzielle Hilfe braucht. Die Artikelwünsche des angemeldeten Benutzers sind auch oben angepinnt, da er so seine Wünsche direkt editieren und erweitern

kann. Durch die Interaktionselemente ist dies sehr schnell und effizient möglich. Anhand der Angabe des Budgets, zusammen mit den Artikelwünschen und Mengen, können sich die Benutzer zudem sehr schnell eine Meinung Bilden, ob das gesamte Budget für den Einkauf überhaupt ausreicht und dies kommunizieren.

**Szenario:**

1. Ein Nutzer hat sich entschieden den nächsten Einkauf für die Gruppe zu bezahlen und Gläubiger zu werden. Bevor er die Verantwortung jedoch auf sich nimmt, will er überprüfen, ob das angegebene Budget der Gruppenmitglieder überhaupt im Rahmen ihrer angegebenen Wünsche ist. Bei einem Mitglied scheint das Budget unverhältnismäßig zu der Menge an Wünschen zu sein. Der Nutzer ist nun in der Lage das Problem mit dem Mitglied zu kommunizieren und so einen möglichen Konflikt in der kommenden Abrechnung zu vermeiden.
2. Der Nutzer will mit seiner Gruppe einen Einkauf planen und dafür seine Wünsche in die Einkaufsliste eintragen. Zuerst gibt er, nach einem Klick auf das Budget-Element, an wie viel er für den Einkauf ausgeben will. Danach füllt er die editierbaren Felder für Artikelname und Menge aus, speichert die Eingabe ab und wiederholt diesen Vorgang solange bis alle Wünsche gespeichert sind.
3. Der Nutzer will kurz vor dem Einkauf doch noch einen Wunsch löschen, weil er es sich kurzfristig anders überlegt hat. Ein Klick auf das Interaktionselement neben dem Wunschartikel, löscht den Eintrag.

**Prüfkriterien:**

Wird das Gesamtbudget aus dem Budget aller Teilnehmer in der Einkaufsliste korrekt berechnet? Ist das Gesamtbudget und die Artikelwünsche des angemeldeten Nutzers oben angepinnt. Ist das Budget und die Artikelwünsche editierbar? Kann der Nutzer Artikelwünsche erstellen und löschen? Werden die korrekten Daten, wie Mitgliedsname etc., angezeigt?

## 4.3 Visuelle Gestaltung

### 4.3.1 Farbe

Bei der Auswahl der Farben wird zum einen die Harmonie der Kombination und zum anderen die Wirkung auf die Benutzergruppe abgewägt. Um ein Design besonders barrierefrei zu gestalten, kann in vielen Apps beobachtet werden, dass hohe Kontraste erzielt werden. Um diesen Anspruch zu erfüllen werden komplementäre Farben verwendet, die mit steigenden Sättigungsgrad den Kontrast maximieren. Jedoch handelt es sich bei der spezifischen Benutzergruppe vor allem um junge Personen, deren Merkmale mit Gesundheit und Vitalität verbunden werden kann. Deswegen bevorzugen die Entwickler eine ansprechende, attraktive Kombination, die einen starken Kontrast jedoch nicht unbedingt berücksichtigt. Ein deutlicher Kontrast zur Schrift muss allerdings immer gegeben sein.



Abbildung 17: Sekundärfarben

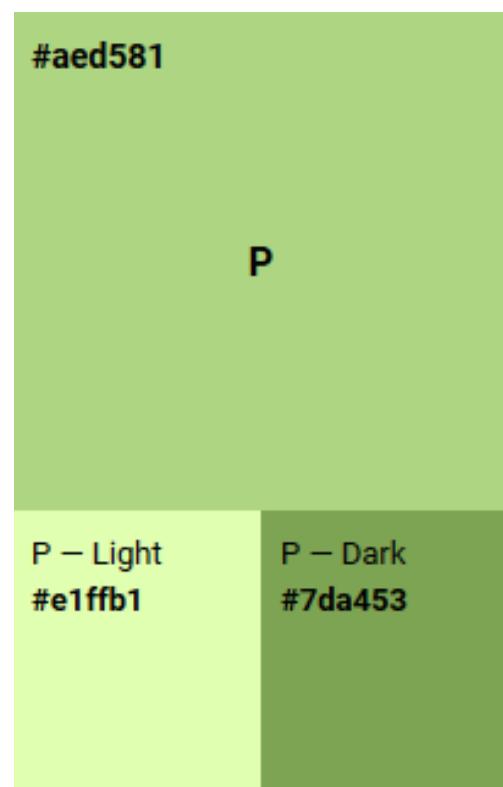


Abbildung 18: Primärfarben

Als Primärfarbe wird eine helles Grün ausgesucht, das durch leichte Kontraste mit einem helleren und einem dunkleren Grün mit dem selben Sättigungsgrad kombiniert wird. Die Sättigung wird dabei gering gehalten. Die Idee dahinter ist, dass Benutzer die Oberflächen mit etwas positivem verbinden. Backpacker kommen häufig mit der Natur in Berührung und sind meist durch Merkmale wie Abenteuerlust und Entdeckerdrang positiv gegenüber der Natur eingestellt. Mit der Auswahl der Farben soll die Natur assoziiert werden. Eine künstlich starke Sättigung könnte gegenteiliges bewirken. Bei der Sekundärfarbe handelt es sich um einen cyan-blauen Farnton der zusammen mit der Primärfarbe einen analogen Farbklang bilden soll. Er ist in der Helligkeit und Sättigung leicht versetzt, um die Primärfarbe etwas herauszustellen. Die Idee, dass die Assoziation zur Natur entstehen soll, ist auch hier gegeben. Während das Grün für statische Bereiche, wie dem Hintergrund und der Toolbar eingesetzt wird, können mit dem Cyan dynamische Elemente, wie dem Menü, dargestellt werden. Der Interaktionsfluss wird mit dem natürlichen Gewässern verbunden werden.

#### 4.3.2 Schriften

Bei der Wahl der Schriftart, wird zunächst überlegt, welche Schriftart in welchem Kontext passend eingesetzt werden kann. Die Überlegung ist, z.B. die Font Courier New oder Times New Roman für die Darstellung von Kassenzettel zu benutzen, da es den Schriftarten vieler realer Kassenzettel entspricht. Jedoch sollte das Font global eingesetzt werden können, also auch für asiatische Schriftsysteme. Hier fehlt für viele Schriftarten jedoch die Unterstützung. Die Google Font Familie bietet mit Roboto und Noto die beiden global eingesetzten Schriftarten. Im westlichen Raum ist Roboto zum Standard für Apps geworden [13]. Sie unterstützt eine Vielzahl an Schriftzeichen und ist für mobile Endgeräte optimiert. Im Rahmen des Projektes fällt die Entscheidung für die gesamt Gestaltung auf Roboto und speziell für Kassenzettel auf Times New Roman.

Auf eine detaillierte typographische Ausrichtung wird im Rahmen des Projektes nicht weiter eingegangen. Jedoch kann beim Design darauf geachtet werden, dass Texte in Schriftgrößen und Schnitt Konsistenz aufweisen.

#### 4.3.3 Piktogramme/ Symbole

Die Auswahl der Piktogramme werden aus dem Open Source Bereich für Android-Entwicklung von Google bezogen [14]. Die Symbole sind für das Gestalten von mobilen Endgeräten optimiert und haben einen hohen Wiedererkennungswert. Bei der detaillierten Auswahl ist für die Entwickler entscheidend, dass die Symbole möglichst selbsterklärend sind und ausfüllend gestaltet wurden. Sie sollen sich von den hellen Farben des Designs deutlich hervorheben. Symbole, die sich für Elemente der Steuerung global etabliert haben werden dementsprechend hier ebenfalls eingesetzt. Symbole, die eine systemspezifische Bedeutung haben, werden an den Nutzungskontext angelehnt. Aus den Wireframes lassen sich folgende Symbole ableiten:



Für Erweiterungen, die im Rahmen des Projekts nicht weiter verfolgt wird, werden diese Symbole als sinnvolle Ergänzungen hinzugefügt.



## 5 Evaluation

Aufgrund der Abweichungen des Projektplans durch Unterschätzungen beim Entwickeln von Gestaltlösungen, konnte kein regulärer Prototyp in fertigem Design erstellt werden. Folglich beschränkt sich die Evaluation auf die bis dahin fertig gestellten Wireframes. Die erste Evaluationsmethode, Mängel anhand der Ten Usability Heuristics nach Jacob Nielsen zu entdecken, findet ohne das Miteinbeziehen von Probanden statt und ist daher, als Ansatz für kleinere Iterationen, lediglich ein Einstieg in den Optimierungs-Mechanismus.

### 5.1 Durchführung der Heuristischen Evaluation

#### Visibility of system status

- Der Client kann vom Server Push-Benachrichtigungen, hinsichtlich der PubSub-Topics, bekommen, welche in einem dafür vorgesehenen Interaktionselement angezeigt werden
- Das System der Push-Nachrichten ist beliebig weit erweiterbar. Z.B neue Topics, neue Funktionen, Updates etc.
- Der Status eines Abrechnung-Fortschritts wird in der Abrechnungsliste oder der Detailansicht nicht dargestellt.
- Bei der Abrechnungsliste ist es unklar, welche Symbolik eingesetzt wird, um den Bezahlstatus alter Abrechnungen anzuzeigen.
- Bei den Details zum Kassenzettel wird nicht ersichtlich, ob eine Beteiligung noch geändert werden kann oder bereits endgültig ist.
- Beim löschen von Einträgen aus der Einkaufsliste sollte der Benutzer bspw. via Toast über den Erfolg des Löschvorgangs informiert werden.
- Die Informationen, die über das Meldesignal Aufschluss über neue Ereignisse geben sollen sind völlig ungeklärt.
- In der Kassenzettel-Detailsicht ist der Abrechnungszeitraum des Kassenzettels und der Gläubiger nicht direkt einsehbar.
- Das Layout zum annehmen/ablehnen und absenden eines Vorschlags existiert nicht

#### Match between system and the real world

- Das Interface ist hauptsächlich aus vorgefertigten Dialogen und Optionen aufgebaut, diese können auf dem Client in jede Sprache übersetzt werden, welche die Entwickler implementieren. Es ist also sehr skalierbar. Da die Nutzer die Anwendung in ihren nativen Sprachen verwenden können herrscht weniger Verwirrung und es gibt weniger Kommunikationsprobleme.

- Die Symbole sind sehr universell und eindeutig für erfahrenen Benutzer
- Kassenzettel sind in der Times New Roman Schrift abgebildet, was sehr an einen gedruckten Kassenzettel erinnert
- Die verwendeten Icons und Symbole sind ein Android Standard und werden auf der ganzen Welt verwendet. Die Nutzer sollten diese Icons sehr leicht verstehen können.
- Die Objekte (in Listen) sind nicht nach natürlicher bzw. logischer Reihenfolge (alphabetisch) geordnet, sondern nach der Priorität des Nutzers.

### **User control and freedom**

- Alle Interaktionselemente mit denen der Nutzer Werte erstellt, ändert und löscht bieten die Möglichkeit Abbruch- und Wiederholungs-Funktionen zum implementieren, die meisten wurden jedoch noch nicht designt.
- Der Benutzer kann sich an jedem Punkt vor oder zurück bewegen.
- Die Menüführung wurde so gestaltet, dass der Nutzer zu jedem Zeitpunkt mit 1-3 Interaktionen die aktuelle Aktivität verlassen kann und die Aktivität seiner Wahl aufrufen kann.
- Der Benutzer kann in den meisten Interfaces via Bottom Navigation Bar zu den wichtigsten Punkten springen.
- Der Benutzer kann zu jedem Zeitpunkt via Bottom Navigation Bar zu den wichtigsten Punkten springen.
- kritische Vorgänge wie das Löschen von Daten haben noch keine explizite Bestäigungsfunktion.
- Bei einem Drop-Down ist nicht erkennlich, wie der Drop-Down wieder geschlossen wird.

### **Consistency and standards**

- Die Navigation und Symbole entsprechen den Android Standards.
- Die Titelwahl der Listen- und Detailaktivitäten kann zu Verwirrung führen.
- Die Abgrenzung zwischen Solo- und Gruppen-Kassenzetteln kann zu Verwirrung führen.

### **Error prevention**

- Nutzer müssen gescannte Kassenzettel, welche eine hohe Fehlerwahrscheinlichkeit haben, immer erst vor dem Speichern bestätigen.
- Ein fehlerhafter Kassenzettel kann bisher nicht im Design gelöscht werden.

### **Recognition rather than recall**

- Die Objekte sind in der Darstellung bereits deutlich voneinander abgegrenzt und werden durch Icons unterstützt.
- Daten kommen in den Aktivitäten wiederholt vor, so dass der Nutzer sich keine Daten über Aktivitäten hinweg merken muss.
- Die einfachen Icons, Symbole und die Anordnung des Layouts sollten die Aufgabeninstruktionen selbsterklärend machen.

### **Flexibility and efficiency of use**

- Die Handlungen können anhand der Navigation beschleunigt werden, in dem Hauptpunkte, von denen Handlungen ausgehen, immer verfügbar sind.
- Die Menüführung wurde so gestaltet, dass der Nutzer zu jedem Zeitpunkt mit 1-3 Interaktionen die aktuelle Aktivität verlässt und die Aktivität seiner Wahl aufrufen kann.
- Jeder Arbeitsschritt ist in wenigen Klicks zu erreichen.
- Ein Autocomplete für das Eintragen von Wünschen, könnte diesen Vorgang beschleunigen.

### **Aesthetic and minimalist design**

- Die Informationen sind weitestgehend auf das nötigste reduziert.
- Es werden die wichtigsten Informationen zur Durchführung der wichtigsten Arbeitsschritte angezeigt. Zudem werden diese in verschiedenen Aktivitäten angezeigt, sodass der Nutzer sich keine Daten/Werte merken muss.
- In der Abrechnungsliste könnte die Information "Alter Rechnungen:" eventuell entfernt werden, da dies bereits durch Datum und Trennstrich ersichtlich ist.

### **Help users recognize, diagnose, and recover from errors**

- Es gibt keine Fehlermeldungen
- Es wurden jedoch HTTP Fehlercodes formuliert, deren Syntax den Nutzern schon viel über das Problem sagen sollten und Lösungen Vorschlagen

### **Help and documentation**

- Hilfefunktionen oder Tutorials wurden bisher nicht designed und formuliert

## 6 Systemarchitektur

Aufgrund von neuen Erkenntnissen und Änderungen, in sowohl Anwendungslogik als auch benötigten Systemkomponenten, muss die Systemarchitektur aus MS1 teilweise angepasst und neu formuliert werden.

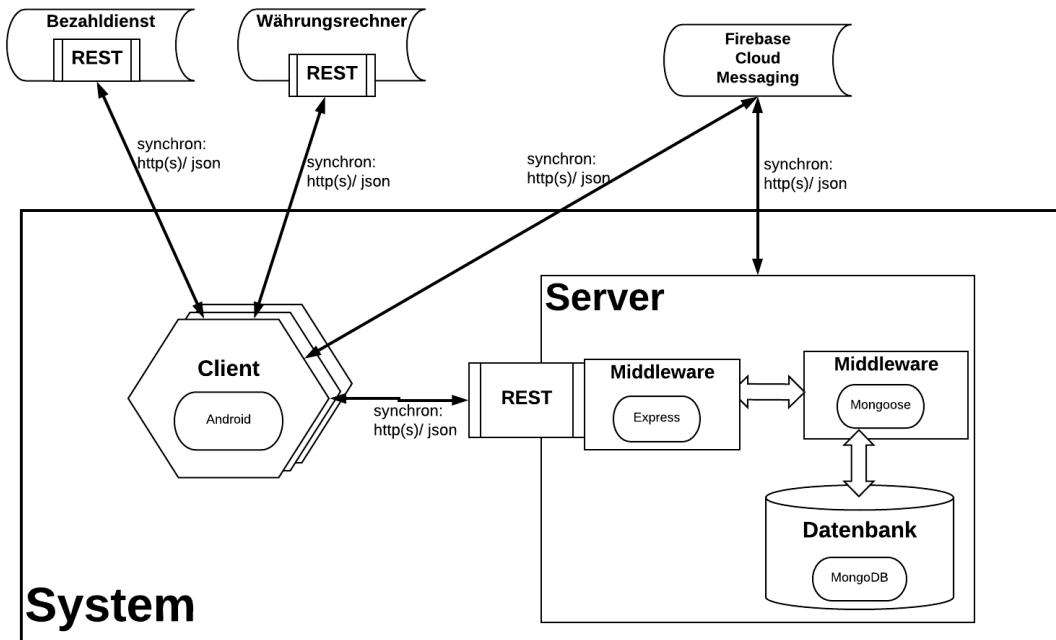


Abbildung 21: Abbildung der Systemarchitektur

### 6.1 Server, Kommunikation, Protokoll

Das System benötigt eine Möglichkeit Benutzer auf abgeschlossene Algorithmen (Abrechnung) oder bestimmten, sie betreffenden, Änderungen in der Datenbank aufmerksam zu machen. Allerdings lässt sich dies nicht mit einer REST-Architektur über HTTP/1.1 so einfach implementieren, wie bisher angenommen (siehe Anhang: Konzept). HTTP/1.1 ist halb-duplex, d.h. der Server antwortet nur auf Anfragen des Clients und kann nicht eigenständig Nachrichten senden. Um Clients kontaktieren zu können müssen deren Daten (Geräte-Id, Token, etc) in Sessions verwaltet werden. Dies verstößt gegen das Prinzip der Statuslosigkeit. Man müsste sich also Gedanken machen, ob man sich von der REST-Architektur abwendet und eine Websocket-Architektur (vollduplex) verwendet. Oder sogar eine Mischung aus beidem (siehe Zukunftsaussicht). Dafür muss man die Vor- und Nachteile der Architekturen abwägen [15].

REST:

pro

- Standards für sichere und idempotente Methoden
- HTTP Fehlercodes
- synchronisation (Sicherheit, dass Nachricht angenommen wird oder nicht)
- kein großer Aufwand pro Verbindung Client-Server
- performanter für wenige, große Daten in großem Zeitraum
- performanter für Daten die sich nicht mehr oder relativ selten ändern

contra

- großer Header, lohnt sich nicht für sehr kleine Daten
- keine vollduplexe, bidirektionale Kommunikation
  - keine Server-Client Kommunikation

Websockets:

pro

- performanter für viele, kleine Daten in
- kurzem Zeitraum
- performanter für sich ständig ändernde Daten
- native Push-Features, vollduplexe und bidirektionale Kommunikation
- kleine Header

contra

- größerer Aufwand pro Verbindung Client-Server
- keine Standards für sichere und idempotente Methoden
- keine nativen Fehlercodes

Um sich nun für die passende Architektur entscheiden zu können, muss man definieren wie oft und in welcher Form Daten versendet werden. In dieser Anwendung sind sowohl Daten vorhanden, die sich regelmäßig ändern (Kassenzettel, Abrechnung, Statistik), als auch Daten die sich gar nicht bzw selten ändern (Benutzer, Gruppe, Kategorien, einzelne Ratings). Jedoch hat die Analyse der Benutzergruppe festgestellt, dass sich die häufiger aktualisierenden Daten nur voraussichtlich mehrmals am Tag verändern. Am Beispiel des Kassenzettels wäre dies der initiale Upload und daraufhin einige Veränderungsvorschläge der Gruppenmitglieder. Bzw. bei der Statistik immer nur dann, wenn ein neuer Kassenzettel hochgeladen/geändert wurde (höchstens 1-2x am Tag pro Benutzer). Die

Daten, welche zum Server gesendet oder von diesem empfangen werden, sind nicht besonders groß (vsl. mehrere Kilobyte für die größten Datensätze). Eine Ausnahme wäre das Kassenzettelbild, aber dieses wird nur einmal erstellt und in komprimierter Form hochgeladen, so dass die Speicherkapazität nicht zu sehr belastet wird. Zudem kann man davon ausgehen, dass die Gruppenmitglieder sich auf ihre Reise konzentrieren wollen und persönlich miteinander kommunizieren. Da keine Chat-Funktion implementiert wird, muss kein Fokus auf Echtzeitkommunikation gelegt werden. Wenn man nun daraufhin REST und Websockets miteinander abwägt, ist REST, für das System, die optimalere Architektur. Um den Benutzer dennoch, in bestimmten Bereichen der Anwendung, über den Server kontaktieren zu können, kann der Push-Service FCM (Firebase Cloud Messaging) von Google einbezogen werden. Dieser bietet Schnittstellen für die von uns verwendeten Plattformen Node.js und Android. Er ist einfach zu implementieren, effizient und kostenlos. Um diesen Dienst jedoch verwenden zu können, muss der Datenschutz berücksichtigt werden. In den Push-Nachrichten, die von unserem Server über FCM an die betreffenden Clients gesendet werden, sind nur pseudonymisierte Daten enthalten. So stehen nur die hexadezimalen Ids der referenzierten Objekte in der Nachricht. Man benötigt direkten Zugang zu der Datenbank um diese zu entschlüsseln. So können die Daten von Google und von Sniffing geschützt werden. Notfalls kann die Nachricht zusätzlich noch mit Kryptographie-Algorithmen verschlüsselt werden. Der legitime Benutzer kann beim Empfang der Push-Nachricht die relevanten Daten, anhand der Datenbank-Id, mit einer HTTP GET Methode querien und einsehen. Obwohl die direkten Daten nicht von Google einsehbar sind, so haben sie dennoch Zugriff auf die Geräte-Id des Benutzers, welche sie eindeutig zuordnen können, falls dieser bereits andere Google-Dienste verwendet. Daraus bilden wir die Möglichkeit den externen Push-Service abzuschalten. So müssen die Benutzer jedoch regelmäßig den Server selbst kontaktieren, um Änderungen an Daten einsehen zu können.

Abschließend kann man also sagen, dass der Großteil der Kommunikation zwischen Client und Server, synchron ablaufen wird. Begründet wird dies durch die relativ seltene Verbindung und die geringe Größe der versendeten Daten. Die benötigte Zeit und die Ressourcen für diese Transaktion ist so gering, dass sich vollduplexe und bidirektionale Kommunikation nicht lohnt. Die Kommunikation über FCM ist in sich selbst zwar auch synchron (HTTP). Im Blick auf das ganze System ist es jedoch zeitlich entkoppelt, also asynchron, da der Benutzer bei seiner Ursprünglichen Nachricht an den Server nicht warten muss, sondern eine Benachrichtigung bekommt.

## **6.2 Datenmodellierung**

Für die Datenmodellierung wurde das Domänenmodell aus M1 erweitert und aktualisiert. Es kristallisieren sich fünf Hauptklassen heraus: Nutzer, Gruppen, Abrechnungen, Kassenzettel und Statistik. Diese haben wieder untergeordnete Klassen, mit denen sie in Beziehung stehen (z.B Kassenzettel -> Artikel -> Teilnahmen -> Vorschlag). Da MongoDB mit der Erweiterung Mongoose in der Architektur verwendet wird, handelt es sich um eine dokumentenbasierte Datenbank. Zudem sind die Daten im JSON-Format in der Datenbank vorhanden. Anhand von ihren IDs können die Klassen angesprochen und in anderen Klassen referenziert werden. So können komplexe Strukturen verwendet werden, welche dennoch leicht zu querien sind, dank der populate- und aggregate-Funktionen von Mongoose. Die Daten sind über die REST-Architektur anhand ihrer Ressourcen (siehe REST-Modellierung) ansprechbar. Die REST-Methoden sind in dem Diagramm nicht berücksichtigt worden. Die angegebenen Funktionen stellen hingegen die restliche Anwendungslogik des Servers dar, mit der z.T Daten automatisch und qualitativ angereichert werden.

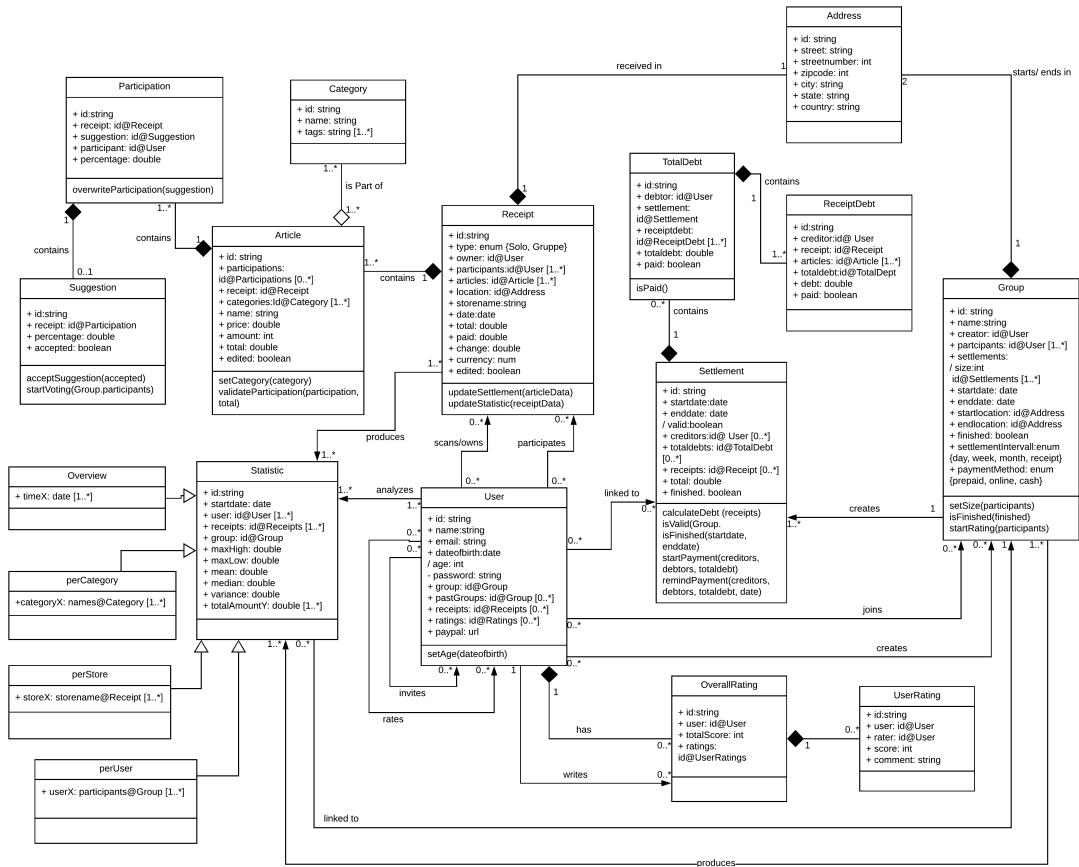


Abbildung 22: Klassendiagramm der Datenstruktur

### 6.3 REST URI Modellierung:

Für die Umsetzung der REST-Schnittstelle, welche in der Systemarchitektur aufgegriffen und erörtert wurde, wird ein Katalog an Ressourcen benötigt auf denen sich atomare CRUD-Methoden abbilden lassen. In der folgenden Tabelle wurden diese genau detailliert. Anhand dessen kann eine API entwickelt werden, welche die Anforderungen an das System befriedigen kann.

Ressource	Methode	Semantik	req	res
/users	GET	Gibt Liste mit allen Usern aus	appl./json	appl./json
	POST	Erzeugt neuen User	appl./json	appl./json
/users/:user-id	GET	Gibt Daten eines bestimmten User aus	appl./json	appl./json
	PATCH	Aktualisiert bestehenden User	appl./json	appl./json
	DELETE	Löscht bestehenden User	appl./json	appl./json
/users/:user-id/ratings	GET	Gibt Liste aller Bewertungen eines Nutzers aus	appl./json	appl./json
	POST	Erzeugt Bewertung für einen Nutzer	appl./json	appl./json
/users/:user-id/ratings/:rating-id	GET	Gibt bestimmte Bewertung eines bestimmten Nutzers aus	appl./json	appl./json
	PATCH	Ändert bestimmte Bewertung eines bestimmten Nutzers	appl./json	appl./json
	DELETE	Löscht bestimmte Bewertung eines bestimmten Nutzers	appl./json	appl./json
/groups	GET	Gibt Liste mit allen Gruppen aus	appl./json	appl./json
	POST	Erzeugt neue Gruppe	appl./json	appl./json
	GET	Gibt Daten einer bestimmte Gruppe aus	appl./json	appl./json
	PATCH	Aktualisiert bestehende Gruppe	appl./json	appl./json
	DELETE	Löscht bestehende Gruppe	appl./json	appl./json

/groups/:group-id	GET	Gibt Liste mit allen Teilnehmern der Gruppe aus	appl./json	appl./json
	POST	Fügt Teilnehmer in Gruppe hinzu	appl./json	appl./json
	DELETE	Löscht Teilnehmer aus Gruppe	appl./json	appl./json1
/groups/:group-id/participants	GET	Gibt bestimmten Kassenzetteln aus	appl./json	appl./json
	POST	Fügt Teilnehmer in Gruppe hinzu	appl./json	appl./json
	DELETE	Löscht Teilnehmer aus Gruppe	appl./json	appl./json
/groups/:group-id/settlements	GET	Gibt Liste aller Abrechnungen in der Gruppe aus	appl./json	appl./json
/receipts	GET	Gibt Liste mit allen Kassenzetteln des angemeldeten Nutzers aus	appl./json	appl./json
	POST	Erzeugt neuen Kassenzettel	appl./json	appl./json
/receipts/:receipt-id	GET	Gibt bestimmten Kassenzetteln aus	appl./json	appl./json
	PATCH	Aktualisiert bestehende Kassenzettel	appl./json	appl./json
	DELETE	Löscht bestehende Kassenzettel	appl./json	appl./json
/receipt/:receipt-id/image	POST	Lädt ein Bild des Kassenzettels hoch	appl./json	appl./json
/receipts/:receipt-id/articles	GET	Gibt Liste mit allen Artikeln eines Kassenzettels aus	appl./json	appl./json
	POST	Fügt neuen Artikel zu Kassenzettel hinzu	appl./json	appl./json
/receipts/:receipt-id/articles/:article-id	GET	Gibt einzelnen Artikel aus	appl./json	appl./json
	PATCH	Aktualisiert bestehenden Artikel	appl./json	appl./json
	DELETE	Löscht bestehenden Artikel	appl./json	appl./json

/receipts/:receipt-id/articles/:article-id/participations	GET	Gibt Liste mit allen Teilnahmen an einem Artikel aus	appl./json	appl./json
	POST	Fügt neue Teilnahme eines Artikels hinzu	appl./json	appl./json
/receipts/:receipt-id/articles/:article-id/participations/:participations-id	GET	Gibt Daten einer einzelnen Teilnahme an einem Artikel aus	appl./json	appl./json
	PATCH	Aktualisiert bestehende Daten zu einer Teilnahme	appl./json	appl./json
	DELETE	Löscht bestehende Daten zu einer Teilnahme	appl./json	appl./json
/receipts/:receipt-id/articles/:article-id/participations/:participation-id/suggestion	GET	Gibt Änderungs- bzw Korrekturvorschlag einer Beteiligung aus	appl./json	appl./json
	POST	Fügt Änderungs- bzw Korrekturvorschlag einer Beteiligung hinzu (Schuldner kann Vorschläge hinzufügen)	appl./json	appl./json
	PATCH	Aktualisiert Änderungs- bzw Korrekturvorschlag einer Beteiligung	appl./json	appl./json
	DELETE	Löscht Änderungs- bzw Korrekturvorschlag einer Beteiligung	appl./json	appl./json
/settlements	GET	Gibt Liste mit allen Abrechnungen aus	appl./json	appl./json
/settlements/:settlement-id	GET	Gibt bestimmte Abrechnung aus	appl./json	appl./json
	PATCH	Aktualisiert bestehende Abrechnungen	appl./json	appl./json
/settlements/:settlement-id/debts	GET	Gibt Liste mit Daten von allen berechneten Schulden aus	appl./json	appl./json

	GET	Gibt Daten einer bestimmten Schuld aus	appl./json	appl./json
/settlements/:settlement-id/debts/:debt-id	GET	Gibt Daten einer bestimmten Schuld aus	appl./json	appl./json
	PATCH	Aktualisiert Daten einer bestimmten Schuld	appl./json	appl./json
	DELETE	Löscht eine bestimmte Schuld	appl./json	appl./json
/settlements/:settlement-id/receipts	GET	Gibt List mit Referenzen der abzurechnenden Kassenzettel aus	appl./json	appl./json
	POST	Fügt Referenz eines Kassenzettels hinzu	appl./json	appl./json
	DELETE	Löscht Referenz eines Kassenzettels	appl./json	appl./json

Tabelle 3: REST-Modellierung

## 6.4 Statuscodes

Diese Liste beinhaltet die wichtigsten HTTP Statuscodes, welche in der Implementierung der REST-Schnittstelle des Servers eine Rolle spielen könnten. Für eine gute Usability könnten die Codes das Errorhandling übernehmen und benutzerfreundliche Rückmeldung des Systems repräsentieren.

Nachricht	GET	POST	PATCH	DELETE
Success	200 - Ok	200 - Ok	200 - Ok	200 - Ok
		201 - Created		
	202 - Accepted	202 - Accepted	202 - Accepted	202 - Accepted
Client-Error	400 - Bad Request			
	401 - Unauthorized	401 - Unauthorized	401 - Unauthorized	401 - Unauthorized
	404 - Not Found	404 - Not Found aus	404 - Not Found	404 - Not Found
			409 - Conflict	
	413 - Too Large	413 - Too Large	413 - Too Large	
Server-Error	500 - Server Error			
	503 - Not available			

Tabelle 4: HTTP-Errorcodes

## 6.5 PubSub-Topics

Dies ist die Liste möglicher Topics für den Publish-Subscribe-Dienst. Es ist vorgesehen, dass die Clients mit bestimmten Aktionen und Einstellungen bestimmte Topics subscribe können (z.B ein Beitritt in eine Gruppe subscribes automatisch einige Topics der Gruppe). Der Server verwaltet die Topics und sendet eine Nachricht, mit den Topics und einer Nachricht, über das jeweilige Topic an den externen Dienst Firebase Cloud Messaging. Dieser leitet die Nachricht an alle Subscriber weiter. Clients sollen nicht selbst publishen können. Für die direkte, synchrone Kommunikation vom Client zum Server ist die REST-Schnittstelle vorgesehen.

Überbegriff	Topic	Trigger	Semantik
Gruppe	:groupId-participants	POST group/:group-id/participants DELETE group/:group-id/participants	Mitglied ist der Gruppe beigetreten und ausgetreten
	:groupId-settlements	POST group/:group-id DELETE group/:group-id	Offene Abrechnung in Gruppe verlinkt, abgeschlossene Abrechnung entfernt
	:groupId-info	PATCH group/:group-id	Informationen der Gruppe geändert (Datum etc)
	:groupId-vote	PATCH /receipts/:receipt-id/articles/:article-id/participations/:participation-id	Aufruf zur Abstimmung über einen Konflikt in der Gruppe
	:groupId	DELETE group/:group-id	Gruppe gelöscht
User	:userId	POST group/:group-id/participants DELETE group/:group-id/participants aus	Du wurdest Gruppe hinzugefügt/ entfernt
		POST DELETE/receipts/:receipt-id/articles/:article-id/participations	Du hast an einem Artikel teilgenommen, Teilnahme an einem Artikel wurde entfernt
		PATCH /receipts/:receipt-id/articles/:article-id/participations	Deine Teilnahme an einem Artikel wurde geändert ODER Dein Vorschlag wurde Abgelehnt

		POST /receipts/:receipt-id/articles/:article-id/participations/:participation-id/suggestion	(Gläubiger wird benachrichtigt) Es wurde ein Änderungs- bzw Korrekturvorschlag hinzugefügt
		DELETE/settlements/:settlement-id/debts/:debt-id	Du wurdest aus einer Abrechnung entfernt
		POST/user/:user-id/rating	Du hast eine Bewertung bekommen
Abrechnung	:settlementId-debts	POST DELETE PATCH /settlements/:settlement-id/receipts (automatischer Trigger wenn neuer Kassenzettel verlinkt wird etc)	aus Abrechnung gelöscht, aktualisiert
	:settlementId-receipts	POST DELETE /settlement/:settlement-id/receipts	Ein Kassenzettel wurde der Abrechnung hinzugefügt, aus Abrechnung entfernt
	:settlementId-info	PATCH /settlement/:settlement-id	Informationen der Abrechnung wurden verändert

Tabelle 5: PublishSubscribe-Topics

## 7 Anwendungslogik

In folgendem Pseudocode soll verdeutlicht werden, wie die aktuelle Abrechnung mit jedem Kassenzettel, der im Zeitraum erstellt, geändert oder gelöscht wird, automatisch aktualisiert wird. Wie im Klassendiagramm gezeigt wird, besteht eine Abrechnung aus den Gesamt-Schulden der einzelnen Gruppenmitglieder, diese bestehen wiederum aus Kassenzettel-Schulden, in denen das Schuldverhältnis der einzelnen Kassenzettel für das jeweilige Mitglied zusammengerechnet wird. Anhand dieser Daten können die Aktivitäten (siehe Wireframes) mit Inhalt gefüllt werden. Die Anwendungslogik der Abrechnung wurde seit dem Proof of Concept in MS1 weitestgehend umgeändert und aktualisiert. Die Entwickler konnten, aus zeitlichen Gründen, den aktualisierten PoC nicht umsetzen, daher wurde dieser Pseudocode formuliert, um die Durchführbarkeit dennoch ansatzweise evaluieren zu können.

```
updateAbrechnung (Methode, Abrechnung, Kassenzettel)
    if (Methode == POST)
        FOREACH Teilnehmer IN Kassenzettel.Teilnehmer
            if (Abrechnung.TotaleSchuld != Null)
                FOREACH TotaleSchuld IN Abrechnung
                    if (TotaleSchuld.Schuldner == Teilnehmer)
                        erstelleKassenzettelSchuld (TotaleSchuld, Teilnehmer, Kassenzettel)
                    else
                        erstelleTotalSchuld (Abrechnung, Teilnehmer, Kassenzettel)
            else
                erstelleTotalSchuld (Abrechnung, Teilnehmer, Kassenzettel)

        if (Methode == PATCH)
            FOREACH TotaleSchuld IN Abrechnung
                if (TotaleSchuld.Schuldner == Kassenzettel.Teilnehmer)
                    FOREACH KassenzettelSchuld IN TotaleSchuld
                        if (KassenzettelSchuld.Kassenzettel == Kassenzettel)
                            berechneSchuld (KassenzettelSchuld, Kassenzettel)
                        else
                            erstelleKassenzettelSchuld (TotaleSchuld, Kassenzettel.Teilnehmer, Kassenzettel)
                else
                    erstelleTotalSchuld (Abrechnung, Kassenzettel.Teilnehmer, Kassenzettel)

        if (Methode == DELETE)
            FOREACH TotaleSchuld IN Abrechnung
                FOREACH KassenzettelSchuld IN TotaleSchuld
                    if (KassenzettelSchuld.Kassenzettel == Kassenzettel)
                        TotaleSchuld.Totalschuld -= KassenzettelSchuld.Schuld
                        DELETE KassenzettelSchuld

berechneSchuld (KassenzettelSchuld, Kassenzettel)
    FOREACH Artikel IN Kassenzettel
        FOREACH Teilnehmer IN Artikel
            IF (Teilnehmer == KassenzettelSchuld.Schuldner)
                KassenzettelSchuld.Schuld += Artikel.Preis * Artikel.Teilnahme.Teilnahmewert

erstelleTotalSchuld (Abrechnung, Teilnehmer, Kassenzettel)
    new TotaleSchuld (Abrechnung.ID, Teilnehmer)
    erstelleKassenzettelSchuld (TotaleSchuld, Teilnehmer, Kassenzettel)

erstelleKassenzettelSchuld (TotaleSchuld, Teilnehmer, Kassenzettel)
    new KassenzettelSchuld (TotaleSchuld.ID, Kassenzettel.Owner.ID, Kassenzettel.ID)
    berechneSchuld (KassenzettelSchuld, Kassenzettel)
    TotaleSchuld.add(KassenzettelSchuld)
    TotaleSchuld.TotaleSchuld += Kassenzettelschuld.Schuld
```

Abbildung 23: Pseudocode für den Abrechnungsalgorithmus

## 8 Proof Of Concept

### 8.1 Optical Character Recognition (Scan)

Mit der Kamera des Android-Clients soll der Nutzer Kassenzettel fotografieren und das entstandene Bild mit Hilfe von OCR-Algorithmen scannen können, so dass der Inhalt des Kassenzettels automatisch erkannt wird. Es ist möglich OCR sowohl auf dem Client (in Java) auszuführen, als auch auf dem Server (Javascript). Da man die OCR-Funktion aber auch im Offline-Modus verwenden können sollte, wird der Algorithmus vorerst auf dem Client implementiert. Die bekanntesten Bibliotheken sind Tesseract [16] und Google Vision [17]. Aufgrund der Open Source Lizenzierung und der ausführlichen Dokumentation wird dies vorerst mit Tesseract implementiert. Für die Implementation auf Android gibt es den Java-Wrapper "tess-two", einen Fork von Tesseract [18]. OCR ist jedoch ziemlich fehleranfällig und komplex. Es benötigt qualitativ hochwertige Bilder, was aufgrund der unterschiedlichen Kameraqualität der Nutzer und deren Methoden Bilder zu erstellen, nicht voraussetzbar ist. Zudem liefert es die besten Ergebnisse nur anhand von maschinellem Lernen. Dies ist jedoch im Rahmen des Projektes aus zeitlichen Gründen nicht umsetzbar. Daher wird das erfolgreiche Exit-Kriterium vorerst tief angesetzt.

#### **Exit-Kriterium:**

Der Client konnte ein Bild des Kassenzettels mit der Kamera erstellen und diesen mit Hilfe von OCR scannen. Mind. 50 Prozent des Textes auf dem Kassenzettel wird korrekt erkannt. Der Algorithmus benötigt höchstens 10 Sekunden. Der Nutzer muss immer die Möglichkeit haben den falsch erkannten Text manuell zu ändern.

#### **Fail-Kriterium 1a:**

Es wurde weniger als 50 Prozent des Textes korrekt erkannt, aufgrund von schlechter Bildqualität (durch Licht, Kamerawinkel, Auflösung, Kontrast, etc.). Der Nutzer merkt dies bei einem direkten Vergleich des Output-Textes mit dem Kassenzettel.

#### **Fallback 1a:**

Das System die Möglichkeit bieten ein neues Foto zu erstellen. Es wäre auch möglich das bestehende Foto weiter zu bearbeiten (Helligkeit, Kontrast, etc).

#### **Fail-Kriterium 1b:**

Es wurde weniger als 50 Prozent des Textes korrekt erkannt, weil der OCR-Algorithmus fehlerhaft oder ineffizient implementiert wurde.

#### **Fallback 1b:**

Die Entwickler müssen mehr Zeit in den Algorithmus investieren und/oder neue Bibliotheken testen. Falls dies zu keiner Verbesserung führt gilt der PoC als gescheitert.

#### **Fail-Kriterium 2:**

Der Algorithmus benötigt länger als 10 Sekunden.

#### **Fallback 2:**

Dies könnte an der geringen CPU-Leistung des Clients liegen. Die Entwickler müssten

eine OCR-Implementation auf dem Server testen. Andernfalls müsste der OCR-Vorgang auf dem Client asynchron sein und den Nutzer nicht von anderen Interaktionen mit dem System blockieren. So würde eine längere Wartezeit dem Nutzer nicht so negativ auffallen.

**Fail-Kriterium 3:**

Es ist dem Nutzer nicht möglich falsch erkannte Daten nach dem Scan manuell zu ändern.

**Fallback 3:**

Diese Funktion ist für den Erfolg des PoC kritisch, da die Wahrscheinlichkeit für falsch erkannte daten sehr hoch ist. Falls der Nutzer auch bei wiederholtem Scan nicht die Möglichkeit hat Daten nachträglich zu ändern gilt der PoC als gescheitert.

## 8.2 Abrechnungs-Algorithmus

Damit ein Benutzer in einer Abrechnung mit einbezogen werden kann, muss dieser in den betreffenden Artikeln eines Gruppen-Kassenzettels referenziert werden. Der Abrechnungs-Algorithmus wird jedesmal automatisch vom Server gestartet, wenn in der Abrechnung ein neuer Kassenzettel referenziert wird oder Daten in einem referenzierten Kassenzettel geändert werden. Der Algorithmus für die Berechnung der Kostenverteilung folgt dem Muster: Artikel(gesamt)preis \* prozentualer Verbrauch des jeweiligen Nutzers. Dadurch kann nachgerechnet werden, ob der Algorithmus die korrekte Kostenverteilung bestimmt hat. Wenn in einem Artikel keine individueller prozentualer Verbrauch angegeben ist, verteilt der Server den Verbrauch automatisch gleichmäßig an alle Beteiligten des Gruppen-Kassenzettels.

**Exit-Kriterium:** Der Algorithmus hat die Kostenverteilung anhand der referenzierten Kassenzettel und Nutzer, sowie Parametern (z.B prozentualer Verbrauch) in mind. 9 von 10 Fällen korrekt ausgerechnet. Die Durchführung des Algorithmus dauert nicht länger als 3 Sekunden.

**Fail-Kriterium 1a:**

Der Algorithmus hat die Kostenverteilung in 2 oder mehr Fällen falsch ausgerechnet, weil der Algorithmus falsch oder ineffizient ist.

**Fallback 1a:**

Der Algorithmus muss von den Entwicklern korrigiert und verbessert werden, indem überprüft wird, ob falsche Variablen im Code verwendet werden oder ob im Algorithmus ein mathematischer Fehler vorhanden ist. Falls dies nicht möglich ist müssen die Nutzer die Abrechnung selbst ausrechnen und der PoC gilt als gescheitert.

**Fail-Kriterium 1b:**

Der Algorithmus hat die Kostenverteilung in 2 oder mehr Fällen falsch ausgerechnet, weil

die falschen Kassenzettel referenziert wurden oder falsche Parameter angegeben wurden.

**Fallback 1b:**

Der Nutzer muss die korrekten Kassenzettel erneut referenzieren oder Parameter ändern können.

**Fail-Kriterium 2:**

Die Durchführung dauert länger als 3 Sekunden.

**Fallback 2:**

Der Algorithmus muss von den Entwicklern optimiert werden, indem überprüft wird, ob die schwache Leistung an dem Querying der Datenbank, dem Datenformat selbst oder dem Aufbau des Algorithmuses liegt. Falls dies keine Verbesserung mit sich bringt sollte der Nutzer währenddessen andere Aktionen ausführen können und vom Client nicht bis zur Antwort des Servers blockiert werden.

## 8.3 Client-Server Kommunikation

Der Großteil der Anwendungs- und Präsentationslogik basiert auf der synchronen Client-Server Kommunikation über HTTP und einer REST-Schnittstelle. So können die verschiedenen Ressourcen aus der Datenbank vom Client abgefragt oder dort erstellt, geändert und gelöscht werden.

**Exit-Kriterium:**

Mehr als 90 Prozent der, in der REST-Modellierung beschriebenen, Ressourcen müssen in mind 9 von 10 Fällen, anhand ihren Methoden und Semantik vom Client aufgerufen und bearbeitet werden können. Die Kommunikation von Anfrage des Clients bis zur Antwort des Servers dauert nicht länger als 2 Sekunden.

**Fail-Kriterium 1:**

In 2 oder mehr Fällen führen die CRUD-Methoden auf die Ressourcen nicht zu den, in der Semantik beschriebenen, Ergebnissen. Bzw. mehr als 10 Prozent der definierten Ressourcen können nicht angesprochen werden.

**Fallback 1:**

Die Entwickler müssen mehr Zeit in die Programmierung der REST-Architektur und Programmierung stecken und mögliche Fehler beheben. Falls dies nicht möglich ist gilt der PoC als gescheitert.

**Fail-Kriterium 2:** Die Kommunikation zwischen Client und Server dauert länger als 2 Sekunden.

**Fallback 2:**

Die Entwickler müssen die REST-Architektur optimieren oder testen ob eine TCP Socket-Kommunikation bessere Ergebnisse bringt.

## **8.4 Firebase Cloud Messaging (FCM)**

In einer REST-Architektur kann der Server normalerweise den Client nicht kontaktieren, ohne vorher eine direkte Anfrage erhalten zu haben. Es wird jedoch eine zeitlich entkoppelte, asynchrone Benachrichtigung des Clients zu bestimmten Aktivitäten des Servers oder anderer Nutzer benötigt. FCM bietet dies, anhand eines Push-Dienstes [19]. Der Server sendet eine HTTP Nachricht mit den Datenbank-IDs der benötigten Daten an den FCM-Server, welcher diese an alle Clients weiter leitet, die eine bestimmte Topic abonniert haben. Direkte Nachrichten an einzelne Clients sind auch möglich.

### **Exit-Kriterium:**

In mind. 9 von 10 Fällen kann der Server eine Nachricht mit den benötigten Daten-IDs an den FCM-Server senden und dieser leitet die Nachricht an alle empfangsbereiten Clients weiter. Mind. 75 Prozent der nicht-empfangsbereiten Clients (kein Internet, kein Strom) sollen die Nachricht automatisch erhalten, sobald sie wieder online sind. Die Nachricht ist kleiner als 4KB, da FCM keine größeren Payloads akzeptiert. Die Nachricht benötigt nicht länger als 5 Sekunden um vom Server über FCM zum Client zu gelangen.

### **Fail-Kriterium 1:**

In 2 oder mehr Fällen kann der Server die Nachricht nicht an den FCM-Server senden oder dieser leitet diese nicht an alle empfangsbereiten Clients weiter.

#### **Fallback 1:**

Die Entwickler müssen die Einbindung des FCM-Dienstes optimieren und eventuelle Fehler beheben. Falls dies nicht möglich ist muss ein TCP Socket Server für diese Art der asynchronen Kommunikation implementiert werden. Falls der Nutzer FCM bewusst nicht nutzen will, muss er die geforderten Daten manuell aktualisieren.

### **Fail-Kriterium 2:**

Mehr als 25 Prozent der nicht-empfangsbereiten Clients erhalten die Nachricht nicht automatisch wenn sie wieder online sind.

#### **Fallback 2:**

Die Entwickler müssen die Einbindung des FCM-Dienstes optimieren und eventuelle Fehler beheben. Der Server kann die Nachricht, falls diese sehr wichtig ist, erneut über den FCM an die Clients senden oder die Clients rufen die Daten eigenständig per HTTP GET vom Server auf.

### **Fail-Kriterium 3:**

Der Payload der Nachricht ist größer als 4KB und FCM versendet die Nachricht nicht.

#### **Fallback 3:**

Die Entwickler müssen den Payload der Nachrichten kürzen oder in mehrere Nachrichten

aufteilen, so dass der Payload einzelner Nachrichten nicht 4KB übersteigt

**Fail-Kriterium 4:**

Die Nachricht benötigt länger als 5 Sekunden vom Server über den FCM-Server zu den Clients

**Fallback 4:**

Die Entwickler müssen die Einbindung des FCM-Dienstes optimieren und eventuelle Fehler beheben. Die Clients rufen die Daten eigenständig per HTTP GET vom Server auf. Webservice: Online-Bezahldienst “Paypal” Für die Begleichung der Abrechnungen sollen die Gruppenmitglieder einen Online-Bezahldienst verwenden können. Im finalen Produkt kann dies durch eine Vielzahl von Anbietern geschehen, aber im Rahmen dieses Projekts wird die Umsetzung des Dienstes “Paypal” priorisiert. Für eine garantierte, sichere Überweisung der Abrechnung muss im Vorfeld Geld an die Entwickler überwiesen werden, welche nach der Abrechnung das Geld wieder, anhand der errechneten Kostenverteilung an die Mitglieder auszahlen. Hier fallen Gebühren an. Alternativ kann die unsichere Zahlungsmethode über paypal.me verwendet werden. Hier gibt der Gläubiger einen Link zu seiner Paypal-Seite mit dem gewünschten Zahlungsbetrag. Die Durchführung der Zahlung kann von den Entwicklern allerdings nicht garantiert werden.

## 8.5 Webservice: Online-Bezahldienst “Paypal”

Für die Begleichung der Abrechnungen sollen die Gruppenmitglieder einen Online-Bezahldienst verwenden können. Im finalen Produkt kann dies durch eine Vielzahl von Anbietern geschehen, aber im Rahmen dieses Projekts wird die Umsetzung des Dienstes “Paypal” priorisiert. Die Paypal-API wird anhand einer REST-Schnittstelle mittels HTTP den Entwicklern zur Verfügung gestellt [20]. Für eine garantierte, sichere Überweisung der Abrechnung muss im Vorfeld Geld an die Entwickler überwiesen werden, welche nach der Abrechnung das Geld wieder, anhand der errechneten Kostenverteilung an die Mitglieder auszahlen. Hier fallen Gebühren an [21]. Alternativ kann die unsichere Zahlungsmethode über paypal.me verwendet werden [22]. Hier gibt der Gläubiger einen Link zu seiner Paypal-Seite mit dem gewünschten Zahlungsbetrag. Die Durchführung der Zahlung kann von den Entwicklern allerdings nicht garantiert werden.

**Exit-Kriterium:**

In mind. 9 von 10 Fällen müssen Gruppenmitglieder einen Betrag an die Entwickler senden können und der Betrag muss automatisch vom Server registriert werden. In mind 9 von 10 Fällen muss der Server den gesamten Betrag, anhand einer Kostenverteilung, die durch den Abrechnungs-Algorithmus erhoben wurde, an die Mitglieder, abzüglich der Transaktions- und Servicegebühren, wieder auszahlen. In mind. 9 von 10 Fällen müssen Gläubiger bei der Abrechnung einen paypal.me-Link angeben können, mit dem Schuldner die Abrechnung begleichen können und der Client muss den Link an den Browser

des Clients weiterleiten und dort öffnen können.

**Fail-Kriterium 1a:**

In 2 oder mehr Fällen können die Gruppenmitglieder keinen Betrag an die Entwickler senden.

**Fallback 1a:**

Die Entwickler müssen die Einbindung von Paypal ins System verbessern und die Ursachen des Fehlers finden und beheben. Falls dies nicht möglich ist, muss über die Einbindung eines anderen Online-Bezahldienstes nachgedacht werden. Alternativ kann den Mitgliedern die Bezahlung in Bar oder dem Paypal.me-Link vorgeschlagen werden.

**Fail-Kriterium 1b:**

In 2 oder mehr Fällen konnte der eingegangene Betrag vom Server nicht automatisch registriert werden.

**Fallback 1b:**

Die Entwickler müssen die Einbindung von Paypal ins System verbessern und die Ursachen des Fehlers finden und beheben. Ist dies nicht möglich ist, muss über die Einbindung eines anderen Online-Bezahldienstes nachgedacht werden. Falls ein Betrag eingegangen ist, aber vom System nicht automatisch bearbeitet wurde, müssen die Entwickler diesen manuell bearbeiten.

**Failkriterium 2:**

In 2 oder mehr Fällen zahlt der Server nicht den gesamten Betrag, anhand einer Kostenverteilung, abzüglich der Gebühren, an die Mitglieder aus, sondern zu wenig oder an die falschen Mitglieder.

**Fallback 2:**

Die Entwickler müssen den restlichen Betrag manuell nachträglich überweisen. Falls das Geld an die falschen Mitglieder gesendet wurde, müssen die Entwickler das Geld wieder einfordern und manuell an die richtigen Mitglieder überweisen. Außerdem müssen die Entwickler die Ursachen des Fehlers finden und beheben.

**Fail-Kriterium 3:**

In 2 oder mehr Fällen ist der Paypal.me-Link des Gläubigers nicht in der Abrechnung angegeben worden, weil das System die Daten verloren oder nicht an den Schuldner gesendet hat.

**Fallback 3:**

Der Schuldner kann den Paypal.me-Link vom Gläubiger erneut anfordern oder bar bezahlen, falls der Gläubiger dies annimmt.

## **8.6 Webservice: Währungsrechner “currencyconverterapi”**

Um den Nutzern eine schnelle Währungsumrechnung für eine Abrechnung in Bar zu ermöglichen, soll eine externe API eines Währungsrechners mit dem System verknüpft werden. Für die Planung und diesen PoC verwenden wir die kostenlose API currencyconverterapi. Diese lässt jedoch nur insgesamt 100 Anfragen pro Stunde zu und die Umrechnung von 2 Währungen gleichzeitig pro Anfrage [23]. Für die Veröffentlichung müsste ein kostenpflichtiger Dienst berücksichtigt werden. Daher wäre dieses Feature wohl nur in einer Premium-Version des Produktes vorhanden.

### **Exit-Kriterium:**

In mind 9 von 10 Fällen können Nutzer einmal pro Stunde bis zu zwei Währungen ihrer Wahl umrechnen.

### **Fail-Kriterium 1:**

In 2 oder mehr Fällen können Nutzer nicht mind. einmal pro Stunde bis zu zwei Währungen ihrer Wahl umrechnen.

### **Fallback 1:**

Die stündliche Nutzungsrate der API wurde wohl überschritten, die Anwendung teilt dies dem Nutzer mit. Andernfalls müsste eine kostenpflichtige API eingebunden werden, welche eine höhere Nutzungsrate und eine Währungsumrechnungs-Funktion für alle Währungen anbietet.

## 9 Ausführung der PoC

### 9.1 OCR

Für die Durchführung des Proof of Concepts der OCR-Technologie wurde mit der tess-two Bücherei ein kleine Anwendung programmiert, welche in der Lage ist, mit der Kamera des Clients ein Foto zu machen. Dieses wird anschließen automatisch gescannt. Es wurden testweise drei Fotos von zwei sehr unterschiedlichen Kassenzetteln gemacht. Lichtverhältnisse und Kassenzettelqualität sind nicht perfekt. Die Fotos haben Schatten, nicht viel Kontrast und die Kassenzettel haben zum Teil Gebrauchsspuren (Knicke, Falten, etc). Man kann davon ausgehen, dass die durchschnittlichen Nutzerfotos eine



Abbildung 24: Kassenzettel 1

Abbildung 25: Kassenzettel 2

Abbildung 26: Kassenzettel 3

ähnliche oder bessere Qualität haben werden, da diese Fotos mit einem mittelständigen mobilem Endgerät erzeugt wurden (Huawei Honor X5). Dennoch wurden erstaunlich gute Ergebnisse, für diese rudimentäre Umsetzung der OCR-Technologie, erzielt. Sonder- und Formatierungszeichen, sowie Bilder und Strichcodes, werden zwar kaum bis gar nicht erkannt, aber die Genauigkeit bei Schrift und Zahlen ist relativ hoch. Nach einem manuellen und visuellen Vergleich der Kassenzettelfotos und der Resultat des OCR-Algorithmus kann man sagen, dass das Exit-Kriterium von 50 Prozent Genauigkeit erreicht wurde. Auch die Ausführungsduer lag bei jedem Durchgang bei etwa drei bis vier Sekunden. Somit wurde auch dieses Exit-Kriterium bestanden. Es ist jedoch noch nicht möglich den Text manuell zu ändern. Dies sollte jedoch kein großes Problem in der Umsetzung bieten. Zu diesem Zeitpunkt konnte dies aber leider aus Zeitmangel nicht implementiert werden.

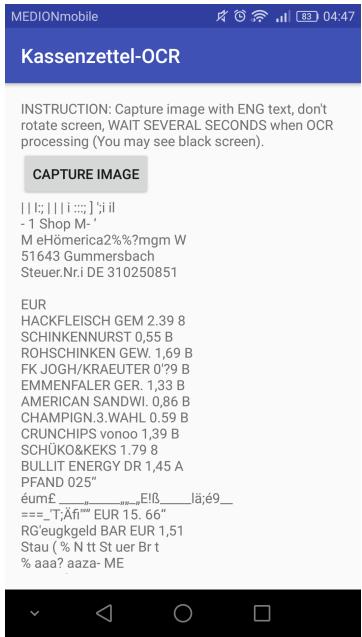


Abbildung 27: Resultat 1



Abbildung 28: Resultat 2



Abbildung 29: Resultat3

Für ein echtes Produkt müsste die Schrifterkennungs-Genauigkeit jedoch noch optimiert werden. In diesem Fall müsste das Exit-Kriterium bei etwa 80-90 Prozent liegen, da es den Benutzer frustrieren würde, mehr als 20 Prozent eines Kassenzettel manuell editieren zu müssen.

## 10 Anhang

### 10.1 zusätzliche Wireframes

#### 10.1.1 Login

##### Layout:

Für den Login werden einzigartige Emailadresse und ein Passwort benötigt. Es sollen möglichst wenige interaktive Elemente (Inputs und Buttons) für die erfolgreiche Nutzung der Aktivität bzw zur Weiterleitung in die nächste betätigt werden müssen

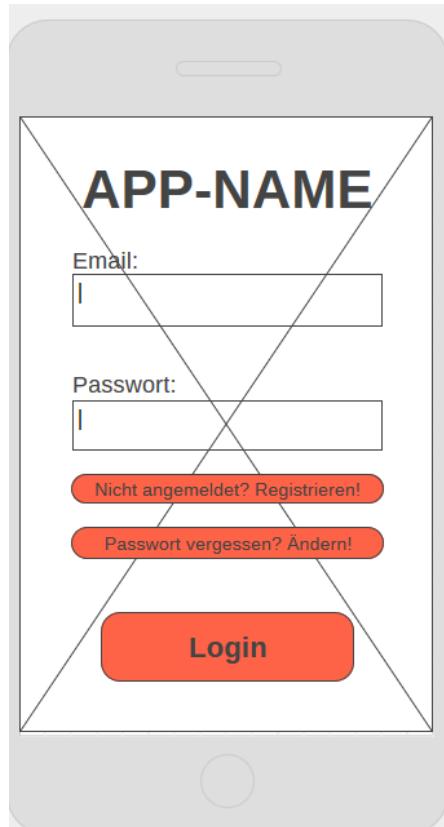


Abbildung 30: Wireframe für Login

##### Prinzip/Begründung:

Je weniger Interaktionen ein Benutzer ausführen muss, um sein Ziel zu erreichen, desto übersichtlicher und verständlicher ist das Design und umso effizienter ist die Nutzung.

##### Szenario

- 1) Bei erstmaliger Nutzung besitzt der Nutzer noch keinen Account, ein Klick auf den

- Registrierungsbutton leitet ihn zu der Aktivität weiter
- 2) Nutzer hat sich bereits registriert. Nach Eingabe der Email und des Passworts in die Eingabefelder leitet ein Klick auf Login zum Homescreen weiter
  - 3) Klick auf "Login" leitet nicht weiter, da Passwort falsch ist. Ein Klick auf "Passwort vergessen" sendet eine Nachricht an die Email des Nutzers, welche die Änderung des Passworts ermöglicht

### Prüfkriterien

Leiten die Buttons in die korrekten Aktivitäten weiter? Erkennen die Eingabefelder falsche Eingaben?

## 10.2 Use Cases

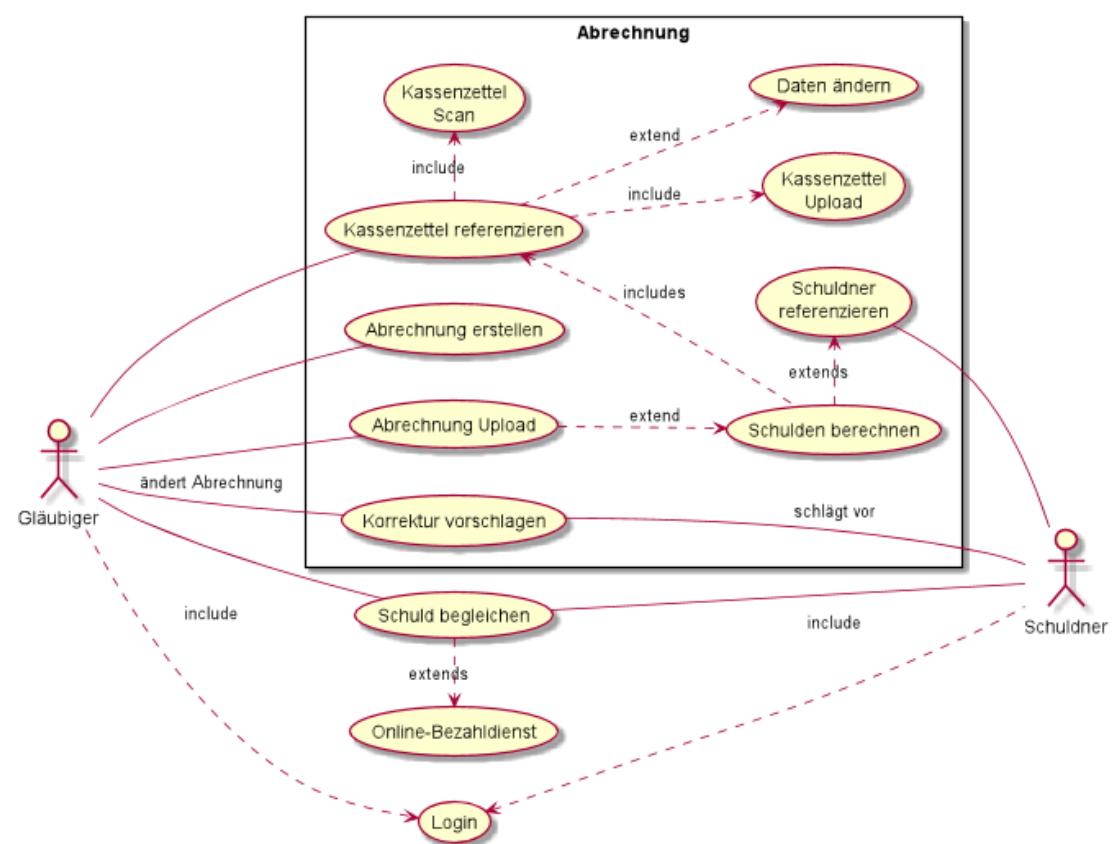


Abbildung 31: Use Case einer Abrechnung

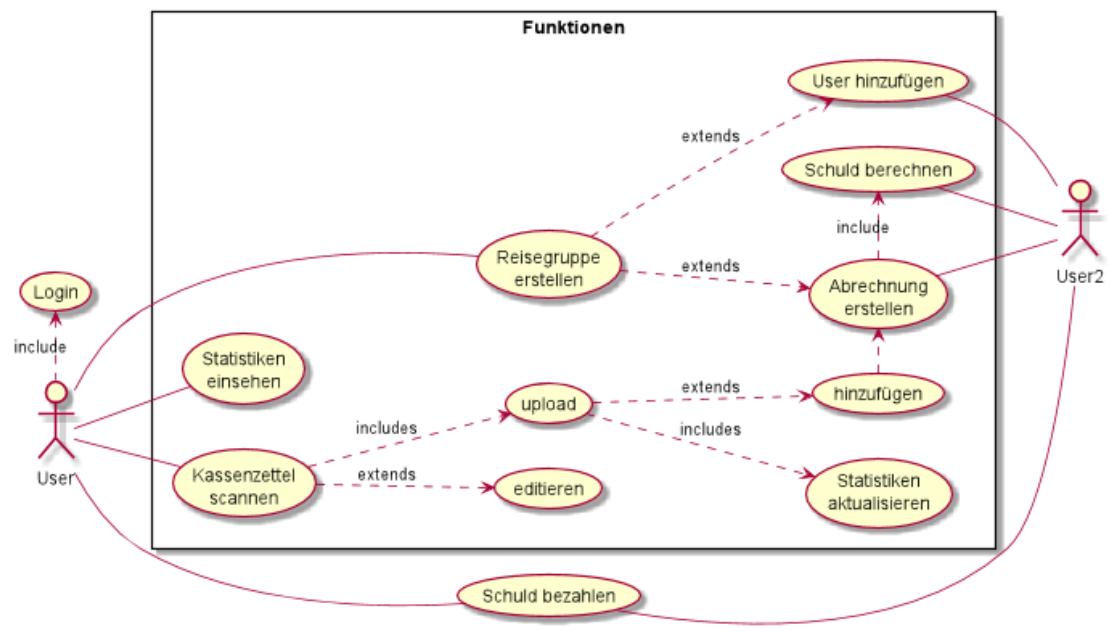


Abbildung 32: Use Case der generellen Funktionen

### **10.3 Durchführung der Interviews**

1. Alex, Student, Backpacking in Asien
2. Jan, Student, Work Travel in Australien
  - Wie alt sind Sie?
    1. 24 (22 zum Zeitpunkt der Reise)
    2. 25 (19 zum Zeitpunkt der Reise)
  - In welchem Land leben Sie?
    1. Deutschland (München)
    2. Deutschland (Köln)
  - Wo hat (haben) die Reise(n) stattgefunden?
    1. Asien (zwei Monate in Indonesien, ein Monat Malaysia, ein Monat Thailand, ein Monat Laos und Cambodia, und einen Monat in Indien.)
    2. Australien (2 Monate in Sydney, 1 Woche Backpacking in Blue Mountains, 3 Wochen Roadtrip nach Melbourne, ca. 2 Monate Roadtrip nach Brisbane, ca 1 Monat Roadtrip nach Cairns)
  - Welche besondere Erwartungen haben sie an eine Reise?
    1. Lernen alleine zurechtzukommen/leben, neue Leute kennenzulernen, viel erleben und neues entdecken
    2. Neue Leute kennen lernen, zu lernen alleine zu leben, neue Kultur kennen lernen, Fremdsprache verbessern, generell viel neues erleben.
  - Haben Sie Apps während der Reise benutzt? Wenn ja, um welche Aufgaben zu erledigen?
    1. Ja. GoogleMaps für Navigation, Konten- und Kreditkarten-Apps für Geld-Management, Whatsapp, Facebook, Skype für Kommunikation.
    2. Ja. Vor allem GoogleMaps für Navigation, Whatsapp für Kommunikation und Gumtree um neue Leute für Roadtrips kennenzulernen und nach Jobs zu suchen.
  - Welche Reiseerfahrungen in Bezug auf Backpacking haben Sie gesammelt?
    - Wie lange ist so eine Backpacker Tour?
      1. 6 Monate
      2. 6 Monate
    - Haben Sie Backpacking mit Fremden zusammen unternommen?

1. Nicht direkt. Hauptsächlich mit der Freundin gereist. Gelegentlich Fremde kennengelernt und abschnittsweise mit ihnen gereist
  2. Ja. Ich bin alleine nach Australien gegangen. Alle Roadtrips waren mit vorerst Fremden, die ich vor Ort kennen gelernt habe. Man hat sich aber sehr schnell kennen gelernt und Freundschaften geschlossen.
- Haben Sie oder jemand anderes dabei in irgendeiner Weise Verantwortung für andere Übernommen?
1. Gelegentlich wurde Geld ausgelegt, wenn jemand keinen Zugriff zum ATM hatte oder die Kreditkarte nicht funktioniert hat.
  2. Ja man ist mit anderen viel Auto gefahren, hat gelegentlich Geld ausgeliehen oder hat viel auf das Wort der andere vertraut
- Gabe es Probleme bzw. ein Hauptproblem?
1. Meine App zur Verifizierung vom Online-Banking hat gesponnen oder wurde deinstalliert, sodass ich kein Geld mehr auf meine Kreditkarte überweisen konnte. Sonst viele Kommunikationsprobleme mit Einheimischen, man wurde ständig über den Tisch gezogen.
  2. Es gab oft Probleme mit den alten, billigen Autos. Sonst gab es in einer Reisegruppe (2 Monate nach Brisbane) nach einiger Zeit Probleme mit den Gemeinschaftsausgaben, da niemand richtig Buch geführt hat und man nach einiger Zeit nicht mehr durchgeblickt hat. Daher haben sich einige benachteiligt gefühlt.
- Haben Sie jemandem Geld geschuldet oder sich umgekehrt geliehen?
1. Meine App zur Verifizierung vom Online-Banking hat gesponnen oder wurde deinstalliert, sodass ich kein Geld mehr auf meine Kreditkarte überweisen konnte. Sonst viele Kommunikationsprobleme mit Einheimischen, man wurde ständig über den Tisch gezogen.
  2. Es gab oft Probleme mit den alten, billigen Autos. Sonst gab es in einer Reisegruppe (2 Monate nach Brisbane) nach einiger Zeit Probleme mit den Gemeinschaftsausgaben, da niemand richtig Buch geführt hat und man nach einiger Zeit nicht mehr durchgeblickt hat. Daher haben sich einige benachteiligt gefühlt.
- Haben Sie jemandem Geld geschuldet oder sich umgekehrt geliehen?
1. Ja gegen Ende musste ich mir wegen den Problemen mit meiner Banking-App öfter Geld leihen. Ansonsten wurde oft Geld bei kleinen Beträgen, wie z.B Essen, geliehen.
  2. Ja. Vor allem in den Reisegruppen wurde oft gemeinschaftlich eingekauft und später abgerechnet.

- Haben Sie sich an gemeinschaftlichen Kosten beteiligt?
  - 1. Ja
  - 2. Ja, sehr oft.
- Haben Sie sich einen Überblick über die eigenen Kosten verschaffen können? Wenn ja, unter welcher Kategorie fielen die meisten Kosten an? Wenn nein, warum nicht?
  - 1. Die Gesamtkosten der Reise hatte ich nur grob über mein Konto im Blick. Ich habe mir nicht jeden Betrag genau aufgeschrieben. Ein Großteil fiel jedoch für Transport ( z.B Flüge, Bus ,Zug, Tuctuc oder Taxis) an.
  - 2. Wie bereits gesagt wurde kaum Buch geführt. Der grobe Blick aufs Konto hat meist ausgereicht. Aber ich gehe davon aus, dass das meiste Geld für Benzin , Verpflegung und ab und zu für Unterkunft ausgegeben wurde.
- Gab es soziale Konflikte?
  - Gab es soziale Konflikte?
  - Wenn ja, wie?
  - Wenn nein, haben Sie eine Idee, wie man diese lösen könnte?
    - 1. Nein, eigentlich nicht wirklich.
    - 2. Ja es gab zeitweise Probleme mit gemeinschaftlichen Einkäufen. Diese wurden zwar durch langes ausdiskutieren und nachrechnen gelöst, aber es hat die Stimmung gegen Ende des Roadtrips etwas getrübt.
- Gab es interkulturelle Missverständnisse?
  - Falls ja, wurden diese gelöst?
  - Wenn ja, wie?
  - Wenn nein, haben Sie eine Idee, wie man diese lösen könnte?
    - 1. Ja das kam schon vermehrt vor. Meistens wurde dies durch Kommunikation über jemanden, der beide Sprachen sprechen konnte, gelöst. Oft auch über Google Translator oder kleine Wörterbücher oder Wörter, die wir während der Reise aufgeschrieben haben.
    - 2. Eher nicht. Ich war aber auch viel mit andere Europäern unterwegs und auch die Australier sind sehr offen und die englische Sprache macht Kommunikation relativ einfach.
- Welche Reisemotive treffen auf Sie am meisten zu: Abenteuerlust, Reiseerfahrung, Aufgeschlossenheit, Entdeckerdrang, Planungsbedürfnis, Budget ?
  - 1. Ich würde sagen: Abenteuerlust, Budget und Aufgeschlossenheit.
  - 2. Abenteuerlust, Entdeckerdrang, Budget.

## 10.4 User Profiles

Merkmal	Ausprägung
Alter	Meist zwischen 22 und 30 Jahren
Geschlecht	m/w
Herkunft	Überwiegend Europa, Asien, Nord Amerika
Bildung	Schlecht bis sehr gut
Sprachen	Erstsprache, Englisch und mehr
Computerkenntnisse	mittel bis viel
Körperl. Verfassung	Gesund, sportlich bis sehr sportlich
Abenteuerlust	Sehr viel
Reiseerfahrung	Mittel - sehr viel
Aufgeschlossenheit	mittel - viel
Entdeckerdrang	Mittel - sehr viel
Planungsbedürfnis	Wenig - mittel
Budget	Wenig - mittel
Spezielle Fähigkeiten	Ungewöhnliche Situationen meistern
Nutzungsmotivation	Wenig bis mittel
Nutzungseinstellung	Nicht eindeutig zugeneigt
Verantwortungsbewusstsein	Eher wenig
Rolle bei gemeinsamen Kosten in einer Gruppe	Eher Schuldner

Tabelle 6: Benutzerprofil eines Abenteurers

<b>Merkmal</b>	<b>Ausprägung</b>
Alter	Meist zwischen 22 und 30 Jahren
Geschlecht	m/w
Herkunft	Überwiegend Europa, Asien, Nord Amerika
Bildung	Schlecht bis sehr gut
Sprachen	Erstsprache, Englisch und mehr
Computerkenntnisse	mittel bis viel
Körperl. Verfassung	Gesund, sportlich bis sehr sportlich
Abenteuerlust	mittel
Reiseerfahrung	Viel bis sehr viel
Aufgeschlossenheit	Mittel - viel
Entdeckerdrang	Mittel - viel
Planungsbedürfnis	mittel
Budget	Wenig - mittel
Spezielle Fähigkeiten	Sehr anpassungsfähig
Nutzungsmotivation	Mittel bis hoch
Nutzungseinstellung	Eher zugeneigt
Verantwortungsbewusstsein	mittel
Rolle bei gemeinsamen Kosten in einer Gruppe	Eher Gläubiger

Tabelle 7: Benutzerprofil eines Weltenbummlers

<b>Merkmal</b>	<b>Ausprägung</b>
Alter	Meist zwischen 25 und 30 Jahren
Geschlecht	m/w
Herkunft	Überwiegend Europa, Asien, Nord Amerika
Bildung	Schlecht bis sehr gut
Sprachen	Erstsprache, Englisch und mehr
Computerkenntnisse	Wenig bis mittel
Körperl. Verfassung	Gesund, sportlich bis sehr sportlich
Abenteuerlust	Sehr viel
Reiseerfahrung	sehr viel
Aufgeschlossenheit	mittel - viel
Entdeckerdrang	Sehr viel
Planungsbedürfnis	Meist Sehr wenig
Budget	Wenig - mittel
Spezielle Fähigkeiten	Genügsam, Überlebenstechniken
Nutzungsmotivation	Gering bis mittel
Nutzungseinstellung	Nicht eindeutig zugeneigt
Verantwortungsbewusstsein	Wenig bis mittel
Rolle bei gemeinsamen Kosten in einer Gruppe	Eher Schuldner

Tabelle 8: Benutzerprofil eines Nomaden

<b>Merkmal</b>	<b>Ausprägung</b>
Alter	Meist 18 - 25
Geschlecht	m/w
Herkunft	Überwiegend Europa, Asien, Nord Amerika
Bildung	Wenig - sehr viel
Sprachen	Erstsprache, Englisch
Computerkenntnisse	Mittel bis viel
Körperl. Verfassung	gesund
Abenteuerlust	Wenig - mittel
Reiseerfahrung	Mittel - sehr viel
Aufgeschlossenheit	mittel - viel
Entdeckerdrang	Mittel - sehr viel
Planungsbedürfnis	Viel bis sehr viel
Budget	Wenig - mittel
Spezielle Fähigkeiten	unbestimmt
Nutzungsmotivation	Wenig bis mittel
Nutzungseinstellung	eindeutig zugeneigt
Verantwortungsbewusstsein	Eher wenig
Rolle bei gemeinsamen Kosten in einer Gruppe	Eher Schuldner

Tabelle 9: Benutzerprofil eines Reise-Anfängers

## 10.5 Personae



Abbildung 33: Lydia Adams aus England (Quelle: FreeImages.com/Angela Granger)

Lydia ist 19 Jahre alt und hat gerade erst das College in Liverpool abgeschlossen. Da sie sich bisher für kein zukünftige Berufslaufbahn entscheiden konnte hat sie relativ spontan beschlossen nach Schweden zu reisen um Distanz zu ihrem Zuhause zu gewinnen. Sie ist gänzlich unerfahren im Bereich Backpacking und schließt sich gerne einer Gruppe skandinavischer Backpacker an, um zu lernen, wie man sich in den weiten Wäldern Schwedens zurechtfindet und um ländliche Orte Schwedens zu erkunden. Eigentlich kam sie nach Stockholm, um etwas Geld in einem Eiscafé als Servicekraft zu verdienen, als sie von einer netten jungen Dänin angesprochen wird und überglocklich den ersten ausländischen Kontakt in ihrem Smartphone verbuchen kann. Sie hat in ihrem sozialen Umfeld schlechte Erfahrungen erlebt, wenn sie Geld verliehen hat. Deswegen hat sie große Probleme anderen Geld anzuvertrauen, wenn es um gemeinsame Planung und Beschaffung von Versorgungen geht. Andererseits ist sie froh, wenn andere Verantwortung dafür übernehmen.



Abbildung 34: John Teller aus den USA (Quelle: FreeImages.com/Gerrit Prenger)

John ist 26 Jahre alt und ein leidenschaftlicher Reisender. Er ist schon als Kind mit seiner seinen Eltern und seiner Schwester des öfteren in verschiedene Staaten Amerikas umgezogen. So konnte er bereits früh lernen sich an neue Mentalitäten und Umgebungen anzupassen. Vor allem über sein Hobby Basketball, konnte er sich immer schnell integrieren. Nach der Highschool hat er Work and Travel in Australien unternommen. Dabei hat er den Ehrgeiz entwickelt, so viele Orte wie möglich zu erkunden und ist vorzugsweise zu Fuß. Größere Distanzen legt er via spontaner Mitfahrtgelegenheiten zurück. Harte Arbeit und bescheidene Unterkünfte machen ihm nichts aus. Er schläft im Sommer sogar schonmal gerne unter freiem Himmel. Er liebt es Sand in den Schuhen und frische Meeresbrisen zu spüren. Als das Visum droht auszulaufen beantragt er ein neues. Diesmal in Europa. Er will im Süden Portugals am Mittelmeer entlang starten und dann weiter bis nach Paris trampen. Wenn er Menschen trifft verbringt er gerne Zeit mit ihnen und lässt sich auf spontane Aktivitäten mit ihnen ein.

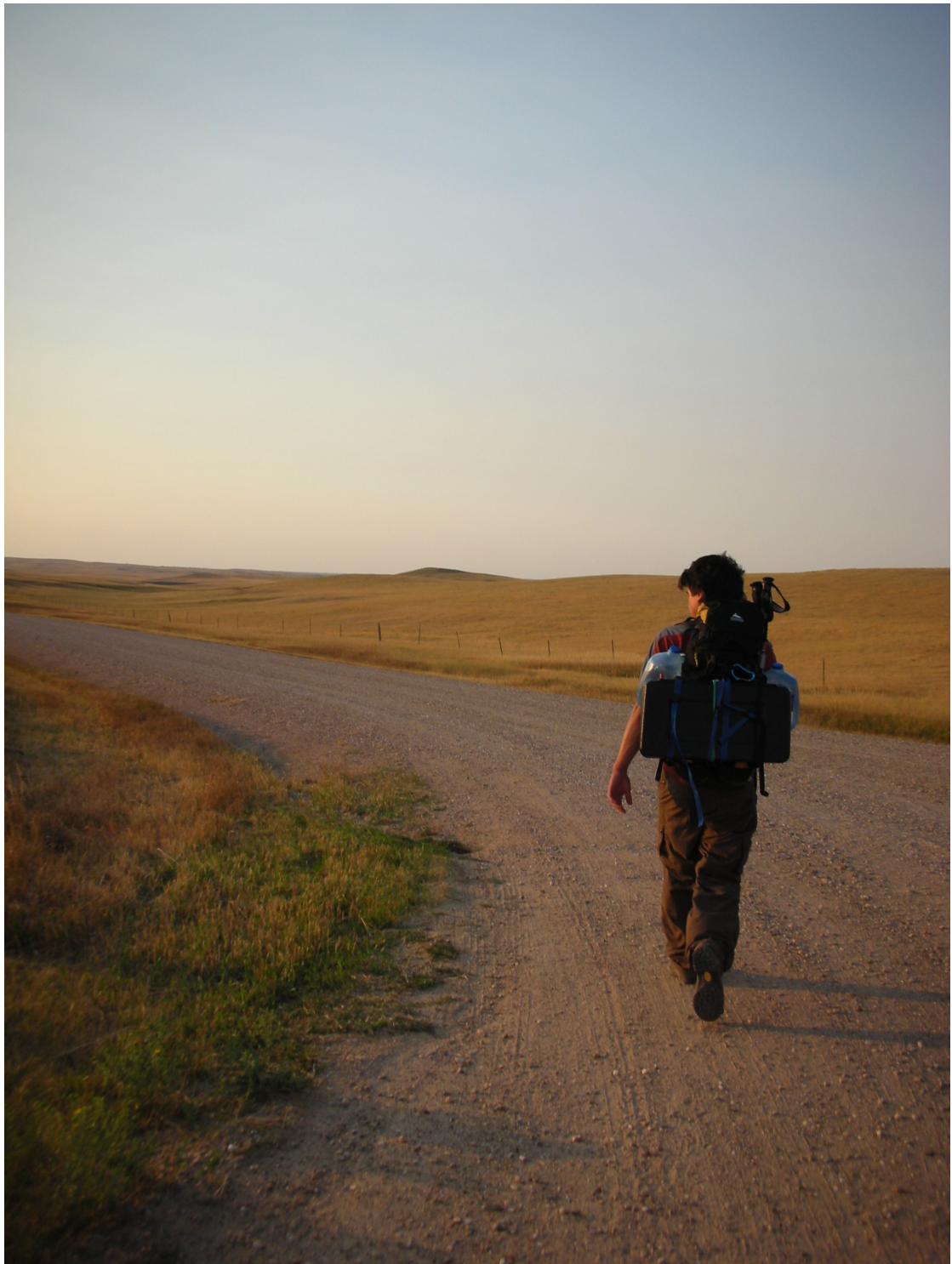


Abbildung 35: Seong-Kyu Von aus Süd-Korea (Quelle: FreeImages.com/Kevin Rohr)

Seong-Kyu 23 Jahre alt und kommt aus Gwangju. Er ist Buddhist und war häufig in den nahegelegenen Tempelstätten in seiner Umgebung um zu meditieren. Das einfache Leben der Mönche hat ihn inspiriert. Durch sein jahrelanges Taekwondo-Training hat er viel Kondition und kann weite Strecken zu Fuß zurücklegen. Aufgrund seines Glaubens hat er bereits mehrere Pilgerreisen unternommen zu Heiligen stätten in Nord-Indien und Neapel. Auch wenn seine erste Motivation zu Reisen aus spirituellen begründet waren, hat sich eine Sehnsucht nach Abenteuern entwickelt. Er beantragt WHV in verschiedenen Ländern und will sich dann ganz spontan entscheiden wo er hinfliegen will. Über sein Smartphone sucht er nach Apps die ihn bei seinem Vorhaben unterstützen könnten. Auch sein Rucksack soll gut gepackt sein. Den Rest will er auf sich zukommen lassen. Er freut sich besonders Menschen aus fernen Ländern kennen zu lernen.



Abbildung 36: Bernard Rigot aus Frankreich (Quelle: FreeImages.com/Benjamin Earwicker)

Bernard ist 25 Jahre alt und ein Genießer. Er ist bei Bordeaux aufgewachsen und liebt es ausgedehnte Abende in Gesellschaft zu verbringen, die mit Wein und Speisen verbunden werden. Im Sommer fährt er gerne an die nahe gelegene Atlantik-Küste, um zu surfen. Oder er fährt mit seinem besten Freund eine weitere Strecke in die Schweiz, um in den Alpen zu wandern, in einer Berghütte zu verweilen und den Ausblick zu genießen. Solange er noch sein Studium auf Lehramt in den Fächern Sport und Spanisch nicht beendet

hat, kann er sich keine größeren Reisen leisten. Doch will er seine Reiseziele ausdehnen, wenn er es sich einmal leisten kann. Er ist Fremden gegenüber sehr aufgeschlossen und ein geselliger, humorvoller Gesprächspartner. Er ist für Gruppenwanderungen die von einem Tal in ein anderes führen, immer zu haben. Solange er am Ende mit dem Bus zurückfahren kann.

## 10.6 Szenarien

Persona: Lydia Adams

Rolle im System: Schuldner

Arbeitsaufgabe: Reaktion auf einen für die Abrechnung modifizierten Gruppen-Kassenzettel

Lydia befindet sich in Gullunge in Schweden. Sie ist mit drei weiteren Backpackern von Stockholm gestartet und möchte mit ihnen Gräddö an der Küste erreichen und anschließend wieder zurück. Während sie und ihre Begleitung in eine zuvor online gemietete Unterkunft in Gullunge einchecken, gehen die anderen beiden Backpacker einkaufen. Alle sind bereits im System integriert und als Gruppe verbunden. Nachdem sich Lydia in ihrem Zimmer eingerichtet und sich mit dem WLAN der Unterkunft verbunden hat, entdeckt sie eine Nachricht. Sie weiß nun, dass der Einkauf durchgeführt wurde und sieht den Gruppen-Kassenzettel, in dem bereits ihre prozentualen Anteile angegeben sind. Sie kennt die schwedischen Produkte und auch die Namen der Produkte nur teilweise und ist sich insgesamt unsicher inwieweit sie sich beteiligen will. Für die Produkte, die sie kennt und für sich in Auftrag gegeben hat, kann sie jedoch schon einmal eine Bestätigung vormerken. Die anderen Beteiligungen will sie festlegen, wenn sie mehr Klarheit hat, um was es sich handelt und inwieweit sie am Verbrauch teilnimmt.

Persona: John Teller

Rolle im System: Schuldner

Arbeitsaufgabe: Vorbereiten einer Abrechnung

John Teller ist in Portugal längst angekommen und befindet sich mittlerweile auf halber Strecke zwischen Lissabon und Porto. Am Strand von Figueira da Foz lernt er drei Portugiesen aus Vagos kennen die ebenfalls zu Fuß in Richtung Norden unterwegs sind. Sie berichten ihm von den Strandbars die auf dem Weg zu finden seien. Also beschließt John für ein paar Tage mit seinen neuen Weggefährten zusammen zu wandern und in den Bars zu feiern und gemeinsam an der Küste zu Grillen. Für die Kostenverteilung wird die Anwendung genutzt und John ist ein Schuldner. Als die Gruppe sich Vagos nährt steht die Abrechnung an und John hat noch nicht einen Euro beigesteuert. Ein Geldpool wurde nicht gebildet, sodass John nun auf die Gruppen-Kassenzettel reagieren muss mit dem Gedanken alle auf einmal zu bezahlen. Insgesamt sind fünf Gruppen-Kassenzettel entstanden, die aus zwei Einkäufen, einem Restaurantbesuch und zwei Barbesuchen bestehen. Luís, einer der portugiesischen Begleiter hat sich als Gläubiger im System eingetragen und für jeden Einkaufszettel vorgeschlagen die Kosten gleich zu verteilen, so wie es im System als Standard voreingestellt ist. John ist damit nicht ganz einverstanden. Er findet zwar, dass die Lebensmittel in etwa gleich aufgeteilt wurden,

aber er hat das Geld für den Restaurantbesuch Luís bereits gegeben und in den Bars hat er viel langsamer getrunken als seine trinkfesten Begleiter. Er setzt seinen für den Restaurantbesuch auf “bereits bezahlt” und streicht seine Beteiligung am Barbesuchen um fünf von 42 Getränken. Nun wartet er, ob dabei ein Konflikt entsteht oder Luís seine Darstellung zu seinem Verbrauch so akzeptiert. Er befürchtet, dass er in einem Voting schlechte Chancen hätte sein Recht durchzusetzen, falls die anderen Gruppenmitglieder sich gegen ihn verbinden und er als Außenseiter unfair behandelt wird.

Persona: Seong-Kyu Von  
Rolle im System: Gläubiger  
Arbeitsaufgabe: Vorbereitung einer Abrechnung

Seong-Kyu wurde ein WHV (Working Holiday Visum) in Australien genehmigt und er ist in Sydney angekommen. Er hatte über die App Hostelbookers bereits eine Unterkunft gebucht und hat dort andere Menschen kennen gelernt. Zusammen mit einem Deutschen und zwei Engländerinnen hat Seong-Kuy eine einwöchige Backpacker-Tour unternommen. Für die Verpflegung in dieser Zeit werden die Einkäufe mittels der Sharing-App geteilt, sodass alle Gruppenmitglieder bereits im System integriert sind. Zwei Einkäufe hat Seong-Kuy bezahlt und hat für beide einen Gruppen-Kassenzettel für eine Kostenverteilung versendet. Er hat für den ersten Gruppen-Kassenzettel bereits Reaktionen erhalten, sodass er ihn vollständig bearbeiten kann. Jedoch wurde der prozentuale Anteil an einigen Artikeln so verändert, dass eine vollständige Bezahlung nicht erreicht werden kann. Seong-Kuy kann diese Reaktion so nicht bestätigen und fordert den Schuldner auf seine Reaktion noch einmal zu überdenken, bevor es zu einem Rating kommt. Für den zweiten Kassenzettel fehlen ihm zwei Reaktionen. Da der Abrechnungszeitraum bald zuende ist, bittet er die beiden Schuldner höflich ihre Reaktion schnell abzusenden, da die Abrechnung sonst droht ohne ihre Zustimmung abgerechnet zu werden.

Persona: Bernard Rigot  
Rolle: Gläubiger  
Arbeitsaufgabe: Durchführung eines Votings

Bernard Rigot befindet sich in der Schweiz mit seinem besten Freund Claude. Zusammen haben sie sich mit einer Engländerin, einem Deutschen und einem Iren zum Wandern verabredet. Für den letzten Tag wollen Bernard und Claude eine alternative Wanderroute zu zweit nehmen, die jedoch am selben Zielort endet wie die der drei Begleiter. Dadurch ist die Gruppe am Ende des Abrechnungszeitraums getrennt. Alle sind bereits im System integriert. Bernard hat für den ersten Abrechnungszeitraum zwei Ein-

käufe getätigt, für die jeweils eine Einkaufsliste erstellt wurde. Bei einem Kassenzettel stimmt der prozentuelle Anteil durch die Reaktionen nicht überarbeiten. Bernard vergleicht die Anteile mit der Einkaufsliste und erkennt, dass bei einer Person Abweichungen der Angaben mit der Einkaufsliste vorhanden sind. Die betroffene Person überarbeitet die Reaktion jedoch nicht und zieht die Reaktion auch nicht zurück. Der Konflikt für die Abrechnung muss jedoch gelöst werden, denn der Abrechnungszeitraum endet an diesem Tag. Bernard möchte Klarheit haben über die Situation und sendet ein Voting raus, damit jeder in der Gruppe über die Konfliktsituation informiert ist und eine gemeinsame Gruppenentscheidung stattfinden kann. Das Voting bestätigt die Forderung von Bernard, sodass er die Abrechnung nach seinen Angaben abschließen kann.

Persona: William Taylor

Rolle: Gläubiger

Arbeitsaufgabe: Durchführen eines Ratings

William ist mit seiner Backpacker-Gruppe in Graz angekommen (Siehe Szenario 1). Drei der fünf Gruppenmitglieder verlassen die Gruppe jetzt. Somit löst William die Gruppe auf. Alle Abrechnungen sind abgeschlossen. Im Abrechnungszeitraum sind zwei Konflikte entstanden. Der erste Konflikt ist entstanden, weil eine Reaktion auf einen Abrechnungsvorschlag durch den Gruppen-Kassenzettel zu starke Abweichungen hatte. Der Konflikt konnte gelöst werden, weil die Person ihm gesagt hat, dass sie sich schlicht geirrt hat. Im Rating kann William nun die Kriterien Glaubwürdigkeit, Konfliktlösung und Kommunikation bewerten und zwischen sehr schlecht und sehr gut gewichten. Trotz des Konfliktes gibt William dieser Person in allen drei Kriterien ein sehr gut und schreibt in den Kommentar, dass der Konflikt durch einen Irrtum entstanden ist. Der zweite Konflikt ist entstanden, weil eine weitere Person auf einen Abrechnungsvorschlag trotz Aufforderungen bis zum Ende des Abrechnungszeitraums nicht reagiert hat. In der Bewertung dieser Person bewertet er die Kriterien Kommunikation und Konfliktlösung mit sehr schlecht und zweifelt auch etwas an der Glaubwürdigkeit dieser Person, sodass er diese mit drei, also mittel bewertet. Im Kommentar weist er auf das nicht reagieren dieser Person hin.

## 10.7 HTA's

### 1. Benutzergruppe

0. Erstellung einer Gruppe mit registrierten Benutzer
  1. Gruppennamen schreiben
  2. Datum: Beginn der Reise auswählen (default: aktuelles Datum)
  3. Datum: Ende der Reise auswählen (keine Pflichtangabe)
  4. Intervall aussuchen das den Abrechnungszeitraum bestimmt
  5. Methode zur Rechnungsbegleichung aussuchen
    - 5.1. M1 auswählen: Geld im Voraus an die Entwickler zahlen
      - 5.1.1. Akzeptierte Währung festlegen
    - 5.2. M2 auswählen: Geld im Nachhinein selbstständig an Gläubiger überweisen (nicht sicher)
      - 5.2.1. Zeitraum für die Bezahlung festlegen
      - 5.2.2. Akzeptierte Währungen festlegen (ev. fallen Umtauschgebühren an)
    - 5.3. M3 auswählen: Geld im Nachhinein selbstständig in bar übergeben (nicht sicher)
      - 5.3.1. Zeitraum für die Bezahlung festlegen
      - 5.3.2. Akzeptierte Währungen festlegen
  6. Erstellte Daten absenden
  7. Benutzer hinzufügen
    - 7.1. Benutzernamen/ E-mail eintragen
    - 7.2. Benutzer einladen absenden

Abbildung 37: HTA: Benutzergruppe

### Plan 0.

**Do** 1, 2, 3, 4, 5 in irgendeiner Reihenfolge  
**Then do** 6 - 7 in dieser Reihenfolge

### Plan 5.

**Do** 5.1  
**Or do** 5.2  
**Or do** 5.3

#### Plan 5.1

**Do** 5.1.1

#### Plan 5.2

**Do** 5.2.1, 5.2.2 in irgendeiner Reihenfolge

#### Plan 5.3

**Do** 5.3.1, 5.3.2 in irgendeiner Reihenfolge

### Plan 7.

**Do** 7.1 - 7.2 in dieser Reihenfolge

Abbildung 38: HTA: Benutzergruppe Plan

## 2. Gruppen-Kassenzettel

0. Erstellung eines Gruppen-Kassenzettels
  1. Einkaufszettel erstellen
    - 1.1. Wünsche in den Einkaufszettel schreiben
    - 1.2. Einkaufszettel senden
  2. Fotografieren eines Einkaufszettels
  3. Scan des Einkaufszettels abwarten
  4. Gläubiger auswählen|
  5. Generierten Kassenzettel bearbeiten
    - 5.1. Irrelevante Zeichen löschen
    - 5.2. Fehler korrigieren
    - 5.3. Fehlende Einträge ergänzen
    - 5.4. Abschließen der Bearbeitung bestätigen
  6. Bestimmte Artikel verschieben
  7. Bearbeiten des Gruppen-Kassenzettels
    - 7.1. Einen Gruppenmitglied an einem Artikel auswählen
    - 7.2. Anteile der Kostenbeteiligung anpassen
  8. Gruppen-Kassenzettel senden/teilen

Plan 0.

**Do** 1. - 2. - 3. - 4. - 5. - 6. - 7. - 8. In dieser Reihenfolge

Plan 1.

**Do** 1.1 - 1.2. in dieser Reihenfolge

Plan 5.

**Do** 5.1. - 5.2. - 5.3. In irgendeiner Reihenfolge

**Do** 5.4.

Plan 7.

**Do** 7.1 - 7.2 In dieser Reihenfolge

**If** Kostenverteilung nicht vollständig **do again**

Abbildung 39: HTA: Gruppen-Kassenzettel

### 3. Reaktion auf einen Gruppen-Kassenzettel

0. Bearbeiten eines empfangenen Gruppen-Kassenzettels
  1. Den Gruppen-Kassenzettel öffnen
  2. Den Gruppen-Kassenzettel bearbeiten
    - 2.1. Artikel markieren



- 2.2. Anteil verstellen
- 2.3. Änderung bestätigen
3. Den Gruppen-Kassenzettel absenden

Plan 0.

**Do** 1.

**If** Angaben sind nicht korrekt, **do** 2

**Then do** 3

Plan 2.

**Do** 2.1 - 2.2 in dieser Reihenfolge

**If** Angaben sind nicht korrekt, **do** 2.1 - 2.2 in dieser Reihenfolge

**Then do** 2.3

Abbildung 40: HTA: Reaktion

## 4. Vorbereitung einer Abrechnung

0. Gruppen-Kassenzettel fertigstellen
  1. Gruppen-Kassenzettel öffnen
  2. Konflikte lösen
    - 2.1. Einzelne Änderungen akzeptieren
    - 2.2. Einzelne Änderungen ablehnen
      - 2.2.1. Vergleich mit dem Einkaufszettel angeben
    - 2.3. Warten auf Reaktion
    - 2.4. Voting durchführen
  3. Kostenverteilung festlegen

Plan 0.

**Do** - 1.

**If** - Konflikt erkennbar, **then do** 2

**Then do** - 3.

Plan 2.

**If** Änderung korrekt, **do** 2.1

**If** Änderung ist nicht korrekt, **do** 2.2

**If** Einkaufszettel vorhanden, **then do** 2.2.1

**then do** 2.3

**If** Konflikt ist nicht gelöst, **then do** 2.4

Abbildung 41: HTA: Vorbereitung einer Abrechnung

## 5. Abrechnung

0. Begleichen einer Rechnung nach der Methode M1
  1. Empfangene Nachricht zur Zahlungsaufforderung ins Gruppen-Konto auswählen
  2. Den Link zum Online-Bezahldienst auswählen
  3. Den Instruktionen des Online-Bezahldienstes befolgen
  4. Die Bestätigung der Zahlung abwarten
    - 4.1. Überprüfen des eigenen Kontos
    - 4.2. Überprüfung der Daten
    - 4.3. Bestätigung fordern
  5. Die automatische Kostenverteilung abwarten
  6. Option zu beanspruchten Überschuss im Gruppen-Konto auswählen
    - 6.1. Geld zurückfordern auswählen
    - 6.2. Geld im Gruppen-Konto behalten auswählen
    - 6.3. Vorgang bestätigen

Plan 0.

**Do** 1. - 2. - 3. - 4. - 5. - 6. in dieser Reihenfolge

Plan 4.

**Do** 4.1. - 4.2. In irgendeiner Reihenfolge

**Then do** 4.3.

Plan 6.

**Do** 6.1.

**Or do** 6.2.

**Then do** 6.3.

Abbildung 42: HTA: Abrechnung M1

0. Begleichung einer Rechnung nach der Methode M2
  1. Empfangene Nachricht zur Zahlungsaufforderung auswählen
  2. Den Instruktionen des Online-Bezahldienstes oder ähnlichem folgen
  3. Die Zahlungsbestätigung abwarten

Plan 0

**Do** 1. - 2. - 3. In dieser Reihenfolge

0. Begleichung einer Rechnung nach der Methode M3
  1. Empfangene Nachricht zur Zahlungsaufforderung auswählen
  2. Die Zahlungsbestätigung abwarten

Plan 0

**Do** 1. - 2. In dieser Reihenfolge

Abbildung 43: HTA: Abrechnung M2 M3

## 6. Votings

0. Durchführung eines Votings
  1. Empfangenes Voting öffnen
  2. Voten
    - 2.1. Eine der beiden Option auswählen und markieren
    - 2.2. Sich enthalten markieren
  3. Das Voting senden

Plan 0

**Do** 1. - 2. - 3. in dieser Reihenfolge

Plan 2

**Do** 2.1.

**Or do** 2.2.

Abbildung 44: Voting

## 7. Ratings

0. Durchführen eines Ratings
  1. Empfangenes Rating öffnen
  2. Person auswählen
    - 2.1. Öffentliche Bewertung durchführen
      - 2.1.1. Eine Zahl von 1 bis 5 auswählen
      - 2.1.2. Einen Kommentar schreiben
    - 2.2. Anonyme Bewertung ausführen
      - 2.2.1. Glaubwürdigkeit: Eine Zahl von 1 bis 5 auswählen
      - 2.2.2. Konfliktlösung: Eine Zahl von 1 bis 5 auswählen
      - 2.2.3. Kommunikation: Eine Zahl von 1 bis 5 auswählen
  3. Rating senden

Plan 0.

**Do** 1. - 2. - 3. In dieser Reihenfolge

Plan 2.

**Do** 2.1. - 2.2. In einer beliebigen Reihenfolge

Plan 2.1.

**Do** 2.1.1. - 2.1.2. In einer beliebigen Reihenfolge

Plan 2.2.

**Do** 2.2.1. - 2.2.2. - 2.2.3. In einer beliebigen Reihenfolge

Abbildung 45: Raiting

## Abbildungsverzeichnis

1	Persona: Entdecker . . . . .	8
2	Darstellung einer HTA . . . . .	19
3	Aufgaben in einem Abrechnungszeitraum . . . . .	20
4	Flowchart der Hauptaktivitäten . . . . .	26
5	Toolbar . . . . .	27
6	NavigationDrawer mit Gruppe . . . . .	28
7	NavigationDrawer ohne Gruppe . . . . .	28
8	Bottom Navigation Bar . . . . .	28
9	Abrechnungsliste . . . . .	30
10	Abrechnungsdetails zusammengeklappt . . . . .	32
11	Abrechnungsdetails aufgeklappt . . . . .	32
12	Gruppenkassenzettel-Liste . . . . .	33
13	Kassenzetteldetails zusammengeklappt . . . . .	35
14	Kassenzetteldetails aufgeklappt . . . . .	35
15	Vorschlag einer Änderung . . . . .	35
16	Einkaufsliste . . . . .	37
17	Sekundärfarben . . . . .	39
18	Primärfarben . . . . .	39
19	Icons . . . . .	41
20	Erweiterung von Icons . . . . .	41
21	Abbildung der Systemarchitektur . . . . .	45
22	Klassendiagramm der Datenstruktur . . . . .	49
23	Pseudocode für den Abrechnungsalgorithmus . . . . .	56
24	Kassenzettel 1 . . . . .	64
25	Kassenzettel 2 . . . . .	64
26	Kassenzettel 3 . . . . .	64
27	Resultat 1 . . . . .	65
28	Resultat 2 . . . . .	65
29	Resultat 3 . . . . .	65
30	Wireframe für Login . . . . .	66
31	Use Case einer Abrechnung . . . . .	67
32	Use Case der generellen Funktionen . . . . .	68
33	Persona: Reise-Anfängerin . . . . .	76
34	Persona: Reise-Nomade . . . . .	77
35	Persona: Reise-Abenteurer . . . . .	78
36	Persona: Weltenbummler . . . . .	79
37	HTA: Benutzergruppe . . . . .	84
38	HTA: Benutzergruppe Plan . . . . .	84
39	HTA: Gruppen-Kassenzettel . . . . .	85
40	HTA: Reaktion . . . . .	86
41	HTA: Vorbereitung einer Abrechnung . . . . .	87

42	HTA: Abrechnung M1 . . . . .	87
43	HTA: Abrechnung M2 M3 . . . . .	88
44	HTA: Voting . . . . .	88
45	HTA: Rating . . . . .	89

## **Tabellenverzeichnis**

1	Benutzer Profil Typ 1 . . . . .	7
2	Stakeholderanalyse . . . . .	14
3	REST-Modellierung . . . . .	53
4	HTTP-Errorcodes . . . . .	53
5	PublishSubscribe-Topics . . . . .	55
6	Benutzer Profil Typ 2 . . . . .	72
7	Benutzer Profil Typ 3 . . . . .	73
8	Benutzer Profil Typ 4 . . . . .	74
9	Benutzer Profil Typ 5 . . . . .	75

## Literatur

- [1] “Entdecker-Katalog 2018 ,” <https://www.travelworks.de/fileadmin/media/pdf/TravelWorks-Entdecker-Katalog-2018.pdf>, work and travel S.14 - 17; Aufgerufen am: 2018-05-16.
- [2] “Backpacker Profiles,” <https://www.backpacker.com/stories/people/profiles#>, backpacker; Aufgerufen am: 2018-05-18.
- [3] “Work and Travel und Backpacking,” <https://blog.workntravel.info/work-and-travel-australien-fakten-und-kurioses-infografik/>, work n travel; Aufgerufen am: 2018-05-18.
- [4] “Besucherzahlen,” <http://www.australien-info.de/daten-besucher.html>, australien info; Aufgerufen am: 2018-05-16.
- [5] “Working Holiday Maker visa programme report ,” <https://www.homeaffairs.gov.au/ReportsandPublications/Documents/statistics/working-holiday-report-june15.pdf>, homeaffairs aus. gov.; Aufgerufen am: 2018-05-19.
- [6] “Interkulturelle Kompetenzen ,” <https://www.ihk-krefeld.de/de/international/aussenwirtschaftspraxis2/leitfaeden-links-und-literatur-zur-interkulturellen-kompetenz.html>, ihk mittlerer niederrhein; Aufgerufen am: 2018-05-19.
- [7] “Andreas M. Heidecke: MCI, Basiswissen für Entwickler und Gestalter,” Springer-Verlag Berlin Heidelberg, 2012, isbn 978-3-642-13507-1, S.225.
- [8] “Alan Dix, Janet Finlay, Gregory D. Abowd, Russel Beale,” Pearson Education Limited, 2004, isbn 978-0-13-046109-4 , S.510ff.
- [9] “Android Developer Leitfäden und Referenzen,” <https://developer.android.com/guide/topics/ui/look-and-feel/>, android developer; Aufgerufen am: 2018-06-17.
- [10] “Google Material Design Leitfäden,” <https://material.io/design/>, material design; Aufgerufen am: 2018-06-17.
- [11] “Drawer Layout aus Andorid Dokumentation,” <https://developer.android.com/training/implementing-navigation/nav-drawer>, drawer layout; Aufgerufen am: 2018-06-17.
- [12] “Bottom Navigation aus Material Design,” <https://material.io/design/components/bottom-navigation.html>, bottom navigation; Aufgerufen am: 2018-06-17.
- [13] “Eine Schrift für die ganze Welt von Antje Dohmann,” <https://page-online.de/typografie/eine-schrift-fuer-die-ganze-welt>, sockets rest; Aufgerufen am: 2018-06-17.
- [14] “Android Material Design Icons,” [https://material.io/tools/icons/?search=shop&icon=shopping\\_basket&style=baseline](https://material.io/tools/icons/?search=shop&icon=shopping_basket&style=baseline), sockets rest; Aufgerufen am: 2018-06-17.

- [15] “Abwägung zwischen Websockets und REST von Jeff Carnahan,” <https://blogs.windows.com/buildingapps/2016/03/14/when-to-use-a-http-call-instead-of-a-websocket-or-http-2-0/>, sockets rest; Aufgerufen am: 2018-06-17.
- [16] “Github-Repository von Tesseract,” <https://github.com/tesseract-ocr/tesseract>, github; Aufgerufen am: 2018-06-17.
- [17] “Google Vision API,” <https://cloud.google.com/vision/>, vision; Aufgerufen am: 2018-06-17.
- [18] “tess-two, ein Fork von Tesseract,” <https://github.com/rmtheis/tess-two>, tess-two; Aufgerufen am: 2018-06-17.
- [19] “Firebase Cloud Messaging,” <https://firebase.google.com/docs/cloud-messaging/>, fcm; Aufgerufen am: 2018-06-17.
- [20] “Paypal Developers Guide für REST-API,” <https://developer.paypal.com/docs/api/overview/>, paypal developer; Aufgerufen am: 2018-06-17.
- [21] “Paypal Transaktionsgebühren,” <https://www.paypal.com/us/webapps/mpp/merchant-fees>, paypal fees; Aufgerufen am: 2018-06-17.
- [22] “Direkte Transaktion mit Paypal.me,” <https://www.paypal.me/>, paypal.me; Aufgerufen am: 2018-06-17.
- [23] “kostenlose Wahrungsrechner API von Manny Vergel,” <https://free.currencyconverterapi.com/>, currencyconverter; Aufgerufen am: 2018-06-17.