EIS

Entwicklung interaktiver Systeme

Sommersemester 2018

Konzept MS1

Team:

Jan Omar Mehr

Armin Weinrebe

Mentor:

Robert Gabriel

Dozenten:

Prof. Dr. Gerhard Hartmann

Prof. Dr. Kristian Fischer

Inhaltsverzeichnis

1 Einleitung

Nur zu oft kommt es vor, dass in einer Gruppe aus Freunden, Familie oder sogar fremden Leuten gemeinsam eingekauft wird und später abgerechnet werden soll. Allerdings ist dies oft sehr umständlich: Kassenzettel müssen durchsucht und nachgerechnet werden, und es wird oft diskutiert, wer welchen Artikel verwendet hat, da es nicht mehr frisch in Erinnerung ist. Zudem gehen Kassenzettel verloren oder nutzen sich ab und sind nicht mehr identifizierbar. Außerdem verliert man einen Überblick über die Gesamtausgaben.

Was wäre wenn eine Anwendung die analogen Kassenzettel digitalisieren kann und daraus Daten wie Supermarkt, alle einzelnen Artikel mit Preis, Gesamtpreis, Datum, etc. gewinnt? Diese Daten würden so weit wie möglich kategorisiert und statistisch aufbereitet werden können. Zudem könnte die Anwendung ermöglichen kollaborativ mit mehreren Nutzern zusammen zu agieren, sodass ein Überblick über Ausgaben geschaffen werden kann und Abrechnungen unterstützt werden und somit leichter fallen. Für so eine Zielsetzung würde sich ein verteiltes System in Form einer Server-Client-Architektur anbieten. Die auf dem Client gewonnen Daten durch den Scan des Kassenzettels können so auf einen Server geschickt werden, der diese qualitativ weiterverarbeitet und speichert. In der Gesellschaft spielt der korrekte Umgang mit Geld überall auf der Welt und in jeder Altersgruppe eine Rolle. Zum einen würde eine Entlastung stattfinden, indem finanzielle Fragen unter den Gruppenmitgliedern auf bequeme und schnelle Weise geklärt werden. Zum anderen wird die Planung unterstützt, sodass Investitionen in Bezug auf damit verbundenen Kosten angezeigt werden.

Wie realistisch und wertvoll wäre die Entwicklung eines solchen Systems? Eine Marktanalyse soll prüfen, welche Lösungen bereits existieren. Eine Domänenrecherche soll ermitteln wo und inwiefern ein Kassenzettel unter seinem Potential verwendet wird und wie die Voraussetzungen für Entwicklungen in dem Bereich sind.

2 Domänenrecherche

In der Domänenrecherche wird mit der “Digitalisierung und Abrechnung von Kassenzetteln” begonnen, bei der ein Nutzungsproblem gesehen wird: Dem ineffizienten Umgang mit herkömmlichen Kassenzetteln. Danach wird die Recherche um die Domäne Backpacker erweitert und untersucht, wie real und relevant das Nutzungsproblem in dieser spezifischen Benutzergruppe ist. Auch ob komplexere Probleme aus dem sozialen Kontext der Backpacker dort zum Teil mit gelöst werden könnten. Zumindest dort, wo es um den gemeinsamen Umgang mit Geld unter fremden Menschen geht.

Zunächst wird der Kassenzettel betrachtet auf der Grundlage persönlicher Erfahrungen und Beobachtungen der Entwickler, sowie Artikeln die diese Eindrücke und Feststellungen bestätigen[1, 2, 3]. Es stellt sich die Frage, wann und warum er aus der Sicht eines Kunden beachtet wird.

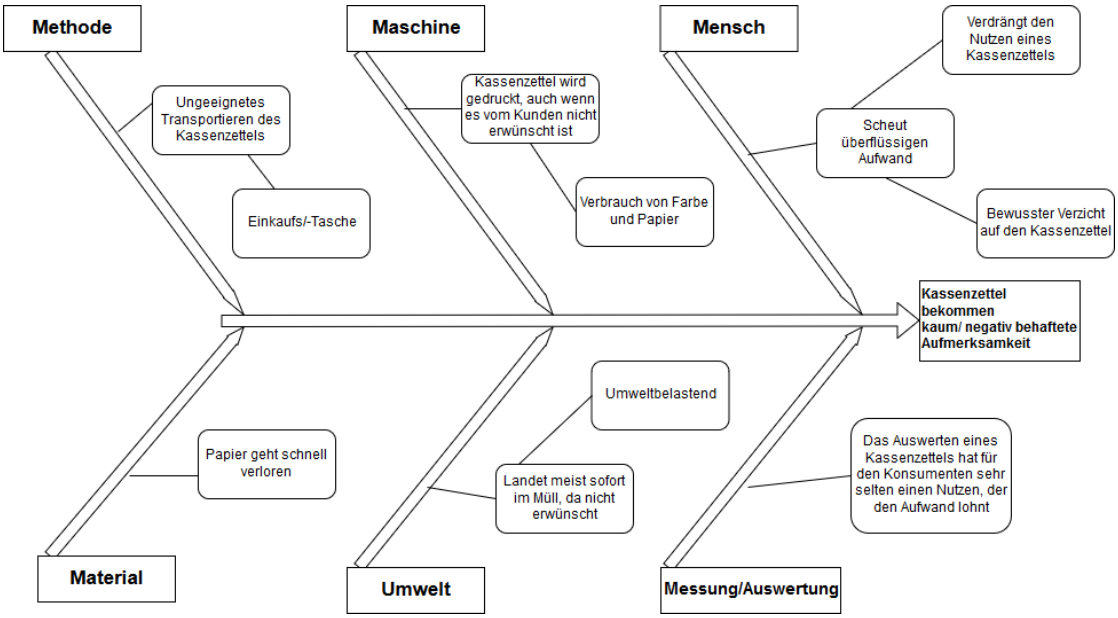

Abbildung 1: Ishikawa-Diagramm

In Abb. 1 ist zu erkennen, dass Kassenzettel einige negative Eigenschaften besitzen und es lässt sich im Alltag beobachten, dass der Umgang mit dem Papier oft dem mit Bonbonpapier gleicht. Offensichtlich besteht aus der Sicht eines Kunden ein Missverhältnis, in welchem das Produzieren von Kassenzetteln in keiner Relation zum tatsächlichen Nutzen steht.

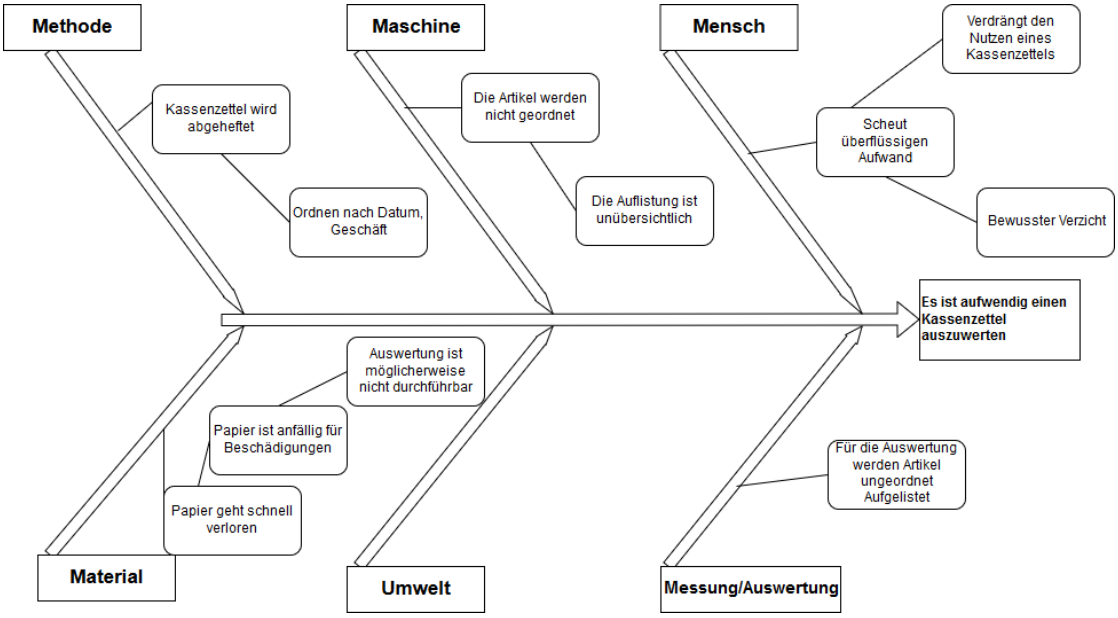

Abbildung 2: Ishikawa-Diagramm

In siehe Abb. 2 werden die Ursachen betrachtet, warum der Nutzen eines Kassenzettels nur selten zur Geltung kommt. Die Untersuchung ergibt, dass Kassenzettel oft in Ausnahmefällen eingesetzt werden, wenn es praktisch ist sie zu haben. Eine nachhaltige Auswertung von Kassenzetteln ist aufwendig und hat für die meisten Privatpersonen keine Relevanz.

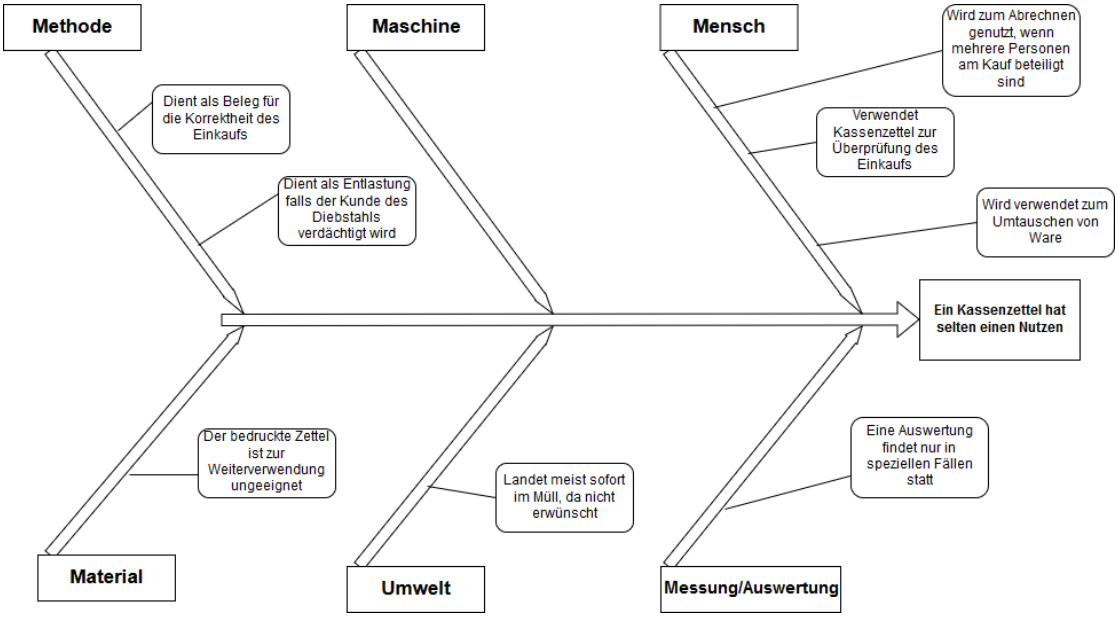

Abbildung 3: Ishikawa-Diagramm

Das Ishikawa-Diagramm in Abb. 3 zeigt, dass viele kleine Zettel mit ungeordnet aufgelisteten Artikeln aufwendige Methoden benötigen, um sie auszuwerten. Warum Kassenzettel dennoch ohne Protest gegen die Gesetzgebung in Massen gedruckt werden, liegt nicht nur an dem oben beschriebenen Nutzen. Sondern mit Sicherheit auch am positiven Symbolcharakter, den Kassenzettel besitzen. Ein Kassenzettel ist eine faire Geste von Geschäften gegenüber dem Kunden. Der Kunde ist in der Lage die Korrektheit des Kaufs zu überprüfen, auch wenn er es in vielen Fällen gar nicht will. Das fördert das Vertrauensverhältnis und somit auch ein positives Kauferlebnis.

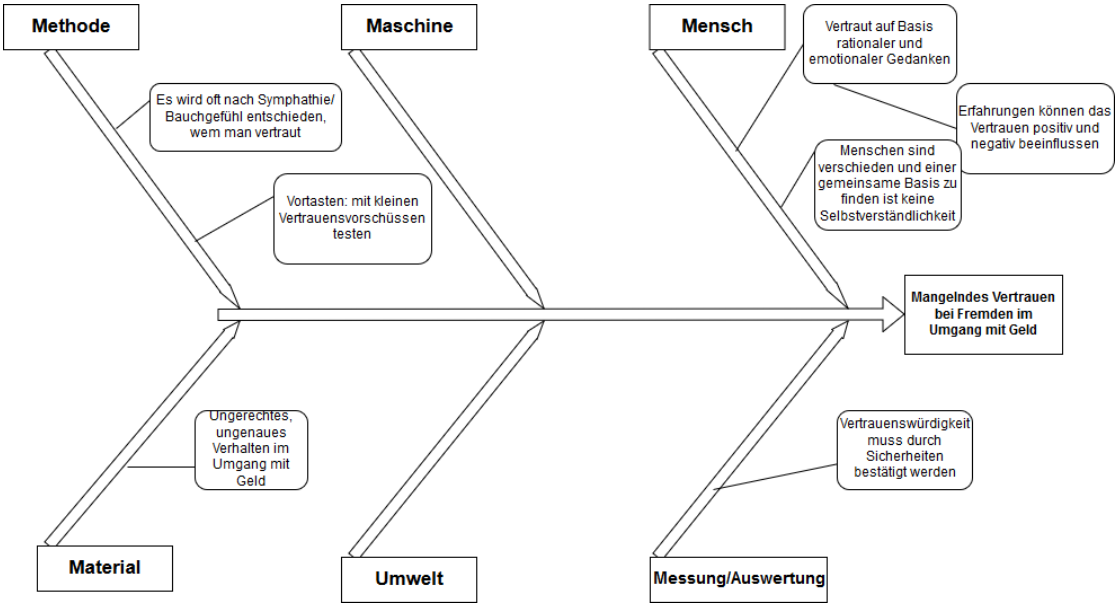

Abbildung 4: Ishikawa-Diagramm

Dieses Ishikawa-Diagramm (Abb. 4) ist als eine vorsichtige Annäherung an das komplizierte Thema, wie wichtig Vertrauen im gemeinsamen Umgang mit Geld ist, zu verstehen. Die wichtigste Erkenntnis für die Entwickler ist die, dass Vertrauen auch über emotionale Komponenten aufgebaut wird und rational nachvollziehbare Sicherheiten die Vertrauensbasis stabilisieren. Das Konzept des Kassenzettels wird bereits global eingesetzt und lässt sich nach Ansicht der Entwickler auf fremde Menschen, die untereinander Einkäufe abrechnen wollen, anwenden.

Backpacker sind sehr häufig alleine oder zu zweit unterwegs, um flexibel zu sein. Jedoch schließen sich Backpacker häufig spontan in Gruppen zusammen, wenn sie sich vor Ort treffen und Erlebnisse teilen wollen, wie sich aus Dokumentationen und Artikeln entnehmen lässt [4, 5, 6, 7].Ein festes Budget erfordert eine hohe Kompetenz im korrekten Umgang mit Geld. Die oft interkulturellen und sozialen Unterschiede und die Tatsache, dass man sich nicht kennt, sind für den gemeinsamen Umgang mit Geld eine spezielle Herausforderung. Trotz des Konfliktpotentials ist der gemeinsame Umgang mit Geld in vielen Situationen sinnvoll. Denn so kann besser organisiert werden und sich in Problemfällen, wenn einem beim Kauf unerwünschte Währungen oder Bezahlformen zur Verfügung stehen, untereinander geholfen werden. Das Konzept des Kassenzettels kann auf diese Zielgruppe angewendet und durch den Vorgang der Digitalisierung das Auswerten, das Abrechnen und die Verteilung von Informationen verbessert werden. Neben einem technisch innovativen Effekt sind Designvorlagen für benutzerspezifische Assoziationen in beiden Domänen zu finden und kombinierbar. So könnten Zettel in einem virtuellen Backpacker-Rucksack gespeichert werden. Alte Zettel könnten virtuell zerrissen oder zerknüllt werden. Markierungen in einem virtuellen Kassenzettel können aussehen wie mit

einem Marker gemalt. Der Kassenzettel könnte auch einem realen Erlebnis zugeordnet werden. Am Ende einer Abrechnung würde ein Schlussstrich gezogen.

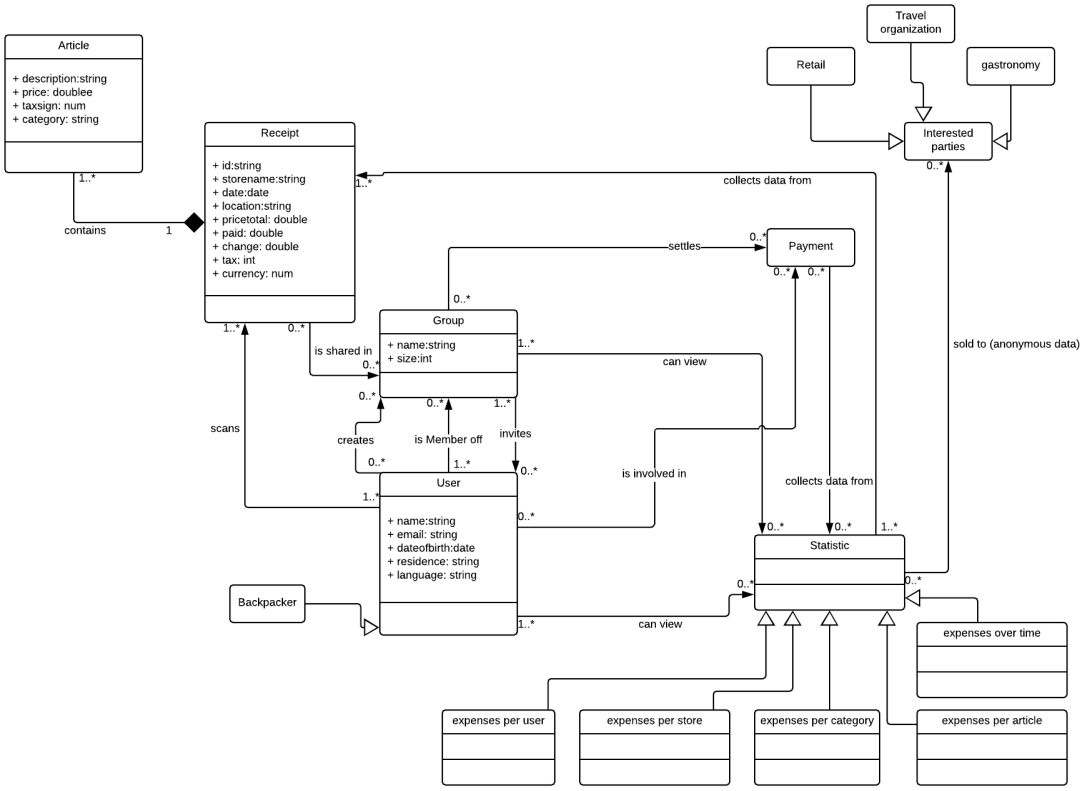

Abbildung 5: Domänenmodell

Das Domänenmodell in Abb. 5 zeigt wie auf technischer Ebene Entitäten, die aus den Domänen entstehen, in Beziehung gebracht werden können. Backpacker können eine Gruppe bilden und Kassenzettel, sowie einzelne Artikel gemeinsam bearbeiten. Dabei können vielen Nutzern vertraute Bezahldienste mit eingebunden werden. Statistiken über Käufe oder anteilige Käufe helfen bei der Übersicht und Gesamtauswertung. Dabei könnten graphische Aufbereitungen bei späteren Designentscheidungen mit Assoziationen zum Backpacken verbunden werden. Die Auswahl einer Kategorie von Ausgaben könnte mit typischen Rucksack-Utensilien dargestellt werden. Eine steigende Bilanz, wie das steigen auf einen Berg. Hohe Ausgaben wie ein zu schwerer Rucksack.Außerdem ließen die Statistiken sich auf ein Geschäftsmodell übertragen. Wenn sie mit Geschäftspartnern unter Beachtung von Bedingungen, geschäftlicher und moralischer Natur, geteilt werden, kann ein kostenloser Service entstehen.

3 Marktanalyse

Die Marktanalyse wurde zeitgleich mit der Domänenrecherche erarbeitet und hat sie dementsprechend beeinflusst. Denn eine Spezialisierung auf eine spezifische Zielgruppe ist auch eine sinnvolle Strategie, um direkte Konkurrenz zu meiden. In diesem Abschnitt werden Systeme vorgestellt, die sich am Markt etabliert haben und verdeutlichen wie stark die Konkurrenz in der Domäne "Digitalisierung und Abrechnung von Kassenzetteln" ist. Mobile Anwendungen, Desktop-Anwendungen und Systeme mit Hardware gehen auf unterschiedliche Weise an das Problem der ineffizienten Nutzung von Kassenzetteln heran. Es wird besonders untersucht, wo die Stärken und Schwächen der Systeme liegen.

3.1 Mobile Anwendungen

3.1.1 Friendcash, Tricount, Trip Splitter

Friendcash ist ein Lösungsansatz dem Problemraum der Zettelwirtschaft auf Reisen entgegen zu treten. Die App ist eventbasiert ausgelegt, sodass einer Reise Mitglieder zugeordnet werden, die mit Hilfe eines Finanz-Rechners Abrechnungen in verschiedenen Währungen vornehmen können. Dabei können Mitglieder in die Abrechnung mit einbezogen oder ausgeschlossen werden, in Bezug auf eine Kaufaktion. Die Daten werden jeweils manuell eingetragen, was je nach größe der Reise und den damit verbundenen Kosten sehr aufwendig sein kann. Die Funktionen sind größtenteils unabhängig vom Internet. Allerdings bedeutet das auch, dass mehrere Clients der einzelnen Mitglieder die einem Event zugeordnet werden, sich nicht automatisch synchronisieren. Die Anwendung bietet hierfür die als Lösung manuell Daten zu integrieren oder das Teilen der Abrechnung als E-Mail-Verteilung zu realisieren. Tricount und Trip Splitter überschneiden sich mit Friendcash im Aufbau und in ihren Funktionen. Sie bieten diese nahezu identisch an, haben jedoch in der Präsentationslogik und dem Design leichte Unterschiede. So bietet Tricount schlichtere Darstellungen an, die für eine bessere Übersicht sorgen. Trip Splitter hat fertige Icons integriert, die bildhaft darstellen für was die Ausgaben getätigt wurden. Dies beschleunigt den Abrechnungsvorgang, da weniger Text manuell eingetippt werden muss. [8, 9, 10] Vorteile:

- In der Komplexität gering gehalten und dadurch sofort benutzbar.
- In den zentralen Funktionen unabhängig vom Internet.
- Benötigt wenig Ressourcen.
- Der eventbasierte Aufbau ist zielführend und ermöglicht eine Problemlösung in wenigen Interaktionsschritten zu realisieren.
- Die Anwendungen sind kostenlos.

Nachteile:

- Wenig Automatisierung und dadurch eine aufwendige Erstellung eines Events.
- Kein verteiltes System, daraus folgt:
  - Keine Verknüpfung bzw. automatische Synchronisation von Clients.
  - Keine Unterstützung des Systems durch einen Server.
- Kein Bezug zu realen Dokumenten.

3.1.2 Klippa

Diese App ist ein starker Konkurrent. Sie umfasst kostenlose Features wie das fotografieren von Kassenzetteln, dem Speichern von Daten wie Einkaufsladen, Gesamtpreis, etc. und Kategorien zu dem jeweiligen Zettel. Eine Suche durch alle gespeicherten Zettel anhand von Datum, Laden oder Kategorie ist ebenfalls implementiert. Außerdem können einfache Statistiken anhand der Daten erstellt werden. Die Kassenzettel mit ihren Daten können in verschiedenen Formaten (pdf, jpeg) exportiert und geteilt werden. Für einen In-App-Kauf kann man das OCR-(Optical Character Recognition) Feature freischalten, mit dem man laut Hersteller einzelne Daten auf dem Kassenzettel automatisch erkennen lassen kann. Leider konnten wir dies aufgrund der Paywall nicht testen. Im Hinblick auf das Alleinstellungsmerkmal konnten wir feststellen, dass die Statistiken nur sehr rudimentär sind und keinen detaillierten Einblick in das Kaufverhalten der Nutzer bieten. Außerdem ist es nicht möglich sich innerhalb der App in Gruppen zusammenzuschließen, sonder nur mit einzelnen Nutzern. [11]
Vorteile:

- Die Anwendung ermöglicht die Integrierung von Dokumenten in das System.
- Die Anwendung enthält praktische OCR-Funktionalität und Suchfunktion.
- Die Anwendung ist plattformübergreifend (Android, iOS und Browser)
- Der Ermöglicht das Exportieren von Daten in verschiedene Formate.
- In der Komplexität gering gehalten und dadurch sofort benutzbar.
- Übersicht durch Kategorisierung.
- Gesamtübersicht auch durch Graphische Darstellungen.
- Die Anwendung präsentiert sich schlicht und übersichtlich.

Nachteile:

- Die praktische OCR-Funktionalität verbirgt sich hinter einer Paywall.
- Es wird keine kollaborative Zusammenarbeit ermöglicht.

3.2 Desktopanwendungen

3.2.1 Shoeboxed

Diese Buchhaltungssoftware wurde für kleinere Unternehmen designed. Sie beinhaltet ebenfalls Features wie OCR, automatische Suchfunktionen, Kategorisierung und Export. Zudem können Konten verknüpft werden und es ist ein direkter Export zu Finanzberatern/-organisationen möglich. Allerdings ist die Software ziemlich teuer (skalierbarer Beitrag zwischen 15 und 69 USD/Monat) und es ist nicht möglich einzelne Artikel auf dem Beleg automatisch zu erkennen. Der mobile Client erhielt sehr schlechte Bewertungen aufgrund von Bugs, schlechter Gebrauchstauglichkeit, etc. [12]
Vorteile:

- Die Anwendung enthält praktische OCR-Funktionalität und Suchfunktion.
- Übersicht durch Kategorisierung.
- Ermöglicht das Exportieren von Daten in verschiedene Formate.
- Verknüpfung mit betroffenen Institutionen.
- Die Anwendung ist plattformübergreifend.

Nachteile:

- Die Software ist kostenintensiv.
- Der Client erfüllt nicht die Anforderungen, die aus Sicht der Benutzer erforderlich sind.

3.2.2 Growth from Knowledge (GfK)

Bei Gfk handelt es sich um ein Unternehmen, welches mit technologischen und wissenschaftlichen Verfahren unternehmensspezifische Fragen zu Verbrauchern, Märkten, Marken und Medien behandelt. Den Kunden verspricht es mit Hilfe von Forschung und Analytik Wirtschaftswachstum zu erlangen. Die unternehmenseigene Software wird nicht gehandelt, jedoch bietet das Unternehmen einen Client für Kunden an, zur besseren Zusammenarbeit. In Bezug auf das Projekt ist die Messung zu Verbraucherverhalten besonders interessant. Nach Angaben der GfK wurde beispielsweise eine umfassende Untersuchung zum Kaufverhalten von Urlaubern bei der Planung der nächsten Reise durchgeführt. Dabei wurden in 15.000 Haushalten über ein Browser-Plug-In Verhaltensweisen mitgeschnitten während ein Media Efficiency Panel Demografiedaten, Absichten und Kaufaktionen zusammenträgt. [13]
Vorteile:

- Fundierte Verfahren zur Ermittlung von Kaufverhalten.
- Repräsentative Ergebnisse durch enorme Quantität der Daten.

Nachteile:

- Sehr kostenintensiv.
- Sehr undurchsichtig.

3.3 Hardware mit zugehöriger Desktopanwendung

3.3.1 The Neat Company

Diese Firma verkauft Scanner mit zugehöriger Software. Der Nutzer kann so Dokumente mit OCR scannen, Kategorisieren, Durchsuchen, Importieren und Exportieren. Zudem ist es möglich mit anderen Nutzern zu kollaborieren - Allerdings anscheinend nur in Form von Kommentaren und Chats. Es werden auch viele Schnittstellen, wie z.B LinkdIn, OnlineBanking, etc. geboten. Dieses Angebot ist jedoch auch sehr teuer (Scanner anschaffung zwischen 250 und 400 Euro + monatlicher Beitrag für Software zwischen 6 und 21 USD/Monat). Zudem ist die Anwendung langsam, erkennt keine einzelnen Artikel und hat schlechte Bewertungen[14].[15]
Vorteile:

- Die Anwendung ermöglicht die Integrierung von Dokumenten in das System.
- Die Anwendung enthält praktische OCR-Funktionalität und Suchfunktion.
- Ermöglicht das Exportieren von Daten in verschiedene Formate.

Nachteile:

- Sehr kostenintensiv.
- Die Software ist sehr eingeschränkte in ihrer Funktionalität

4 Alleinstellungsmerkmal

Anhand der Marktanalyse wurde festgestellt, dass es für einige Teilaspekte der Nutzungsprobleme bereits Lösungsansätze in verschiedenen Formen gibt. So können Kassenzettel bereits in Ihrer Gesamtheit (also dem Gesamtpreis) gescannt und ausgewertet werden (Klippa, Shoeboxed), aber eine automatische Auswertung anhand einzelner Artikel ist noch nicht möglich. Einige bieten daraufhin auch eine firmeninterne Abrechnung an. Andere Anbieter ermöglichen eine Abrechnung für Freunde oder Reisegruppen anhand von manuell eingetragenen Daten (Friendcash, Tricount, Trip Splitter). Allerdings ist uns dabei aufgefallen, dass es noch keine Kombination aus der automatischen Erfassung der Daten zu einer weitestgehend automatischen Abrechnung innerhalb einer Gruppe gibt. Außerdem ist es in keiner Anwendung möglich die Abrechnung mit allen Teilnehmern in Echtzeit zu synchronisieren. Nur der Export oder das Teilen mit einzelnen Nutzern wird unterstützt. Zudem stellen die Entwickler fest, dass einige Nutzergruppen eventuell gar kein Interesse an unserer Lösung haben könnten. Familien oder enge Freunden nehmen die Abrechnung möglicherweise nicht besonders ernst oder können sich zumindest ohne die Anwendung so gut verständigen und vertrauen, sodass viele unserer Lösungsansätze für diese Nutzer obsolet wären. Daraufhin wurde eine weitere Nutzergruppe untersucht, bei der Geld bzw. eine genaue Abrechnung eine größere Rolle spielt, Kommunikationsschwierigkeiten herrschen oder noch kein Vertrauen aufgebaut wurde. In Folge dessen kristallisierte sich in die Nutzergruppe “Backpacker” oder “junge Erwachsene auf Reisen” heraus, welche in fremden Ländern auf fremde Menschen treffen und sich dort zu neuen Reisegruppen zusammenfinden. Die besten Beispiele hierfür wären Australien, Amerika, Asien oder Kanada, welche als die beliebtesten Ziele für Backpacker gelten (vgl. Domänenrecherche). Im Bezug auf die Recherche und die herausgestellte Nutzergruppe können folgende Alleinstellungsmerkmale konkretisiert werden.

- Automatische Abrechnung anhand von Scan einzelner Artikel eines Kassenzettels.
- Echtzeit-Synchronisation und Einbindung aller Teilnehmer in die Abrechnung.
- Spezielles Design hinsichtlich der Nutzergruppe “Backpacker”.

5 Zielhierarchie

Die Zielhierarchie wird zur Definition von lang- bis kurzfristigen Zielen des Projektes verwendet. Daraus soll ersichtlich werden, welche Vorgänge Priorität haben, wo eventuelle Probleme auftauchen könnten und es soll eine der Grundlagen für die spätere Evaluation des Projektes darstellen.

5.1 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.

5.2 Taktische Ziele

- Benutzer müssen permanent auf technische Unterstützung beim Abrechnen zugreifen können.
- Kollaborative Anpassung und Überarbeitung der Abrechnung soll möglich sein.
- Benutzer erhalten einen transparenten Überblick über gemeinsam erzeugte Kosten.
- Eine Abrechnung soll unmissverständlich auf gegenseitigem Einverständnis beruhen.
- Die Vertrauenswürdigkeit der Benutzer soll messbar gemacht werden.
- Finanzielle Konflikte sollen für betroffene Benutzer transparent und frühzeitig erkennbar sein.
- Unterschiedliche Sprachen und Währungen sollen beim Kommunikationsfluss und Abrechnungsvorgang berücksichtigt und behandelt werden.

5.3 Operative Ziele

- Vorgefertigte Dialoge in Worten, Sätzen und bildlichen Darstellungen sollen für eine einfache eindeutige Kommunikation sorgen.
- Persönliche Daten sollen für eine Vertrauensbildung verschlüsselt werden.
- Kosten sollen abschnittsweise abgerechnet werden können.
- Eine Gruppe aus Benutzern soll sich im System verknüpfen können.
- Kassenzettel sollen digitalisiert und die Informationen weiterverarbeitet werden können.
- Enzelne Artikel sollen in einer Abrechnung für eine transparente und eindeutige Kommunikation gekennzeichnet werden können.
- Das System soll eine angemessene Verwaltung der Bearbeitungsrechte der Daten in Bezug auf das Schuldverhältnis bieten.
- Das System soll durch mobile Endgeräten erreicht werden können.
- Finanzielle und kommunikative Konflikte sollen im System deutlich erkennbar sein.

5.4 Minimalziele

- Die Digitalisierung von Kassenzetteln.
- Die virtuelle Erstellung von Gruppen und deren kollaborative Interaktion und Kommunikation im System.
- Darstellung und Lösung von Konflikten im System.

6 Methodischer Rahmen

Auf Grundlage der Zielsetzung und in Anbetracht der vorliegenden Ressourcen wird abgewägt, welches Vorgehensmodell für die Zielerreichung innerhalb des laufenden Projektes passt.

6.1 User-Centered Design

Im User-Centered Design stehen Benutzer bzw. Benutzergruppen im Fokus. In Bezug auf die kommende Phase des Projekts, stellt sich für die Entwickler die Frage, in welchem Ausmaß auf die individuellen Bedürfnisse der Nutzer eingegangen werden sollte.

6.1.1 ISO-Norm 9241- 210

In der ISO-Norm 9241 Teil 210 wird das Vorgehen auf einer strategischen Ebene in wechselseitig abhängigen menschzentrierten Gestaltungsaktivitäten dargestellt. Die Nutzungsanforderungen, die z.T. noch im Konzept aus den ermittelten Erfordernissen und Erwartungen resultieren werden (vgl. Kap. 7), nehmen hier eine zentrale Rolle ein. Denn gemessen an der Erfüllung dieser Anforderung werden mindestens die Erarbeitung von Gestaltungslösungen und deren Evaluation iteriert. Für die Entwicklung des Systems hieße dies vor allem eine intensivere Auseinandersetzung mit Backpackern vorzunehmen, um den Nutzungskontext präzise eingrenzen und Nutzungsanforderungen ableiten und gewichten zu können.

6.1.2 Usability Engineering Lifecycle

Der "Usability Engineering Lifecycle" nach Deborah Mayhew besteht aus drei Phasen deren Aktivitäten die Anforderungsanalyse, das Design, Testen, die Entwicklung und zuletzt die Installation sind. Das Vorgehensmodell entspricht im Allgemeinen den Grundsätzen der o.g. ISO-Norm, da die erste Phase Anforderungen fokussiert, deren Erfüllung ebenfalls Maßstab iterativ evaluierter Ergebnisse der dritten Phase ist. Es sei besonders zu betrachten, dass aus der Anforderungsanalyse sogenannte "Styleguides" entstehen. Diese geben präskriptiv vor, wie das Design in der zweiten Phase konkret gestaltet wird, damit die zuvor definierten Ziele ("Usability Goals") erreicht werden. Zum einen verspricht der "Usability Engineering Lifecycle", dass sich das System am Ende des Projekts gegenüber dem Benutzer mindestens erwartungskonform präsentiert. Zum anderen ist es aufgrund der ressourcenintensiven Methoden schwer auf das Projekt zu skalieren.

6.1.3 Scenario-Based Usability Engineering

Für das Scenario-based Usability Engineering nach Rosson und Carrol kann vorweggenommen werden, dass es einen mindestens ebenso hohen Arbeitsaufwand für die Entwicklung des Systems bedeutet und dementsprechend auch schwer skalierbar ist. Die Hauptaktivitäten des Modells sind in die Analyse, das Design und das Prototyping, Evaluieren gegliedert. Anders als beim "Usability Engineering Lifecycle" steht bei der Analyse der Problemraum im Mittelpunkt der Betrachtung und die erstellten Artefakte in der Designphase haben ausschließlich deskriptiven Charakter. Der Vorteil für das Projekt, lange in einer konzeptionell abstrakten Eben zu entwickeln, bestünde darin, besonders innovative kreative Designideen in Bezug auf die Domänen (vgl. Kapitel 3) zu finden. Beispielsweise bei Gestaltlösungen für graphische Darstellungen und Verwendung von Metaphern. Jedoch weisen Szenarien ein hohes Defizit an Agilität und Modalität auf. Auch das Fehlen konkreter UI's sind für das Entwickeln der Funktionen im intendierten Systems unvorteilhaft.

6.1.4 Discount Usability Engineering

Das "Discount Usability Engineering" nach Jakob Nielsen ist im Vergleich zu den o.g. Vorgehensmodellen bei Weitem nicht so komplex. Es erfordert von den Entwicklern weniger Aufwand und Ressourcen, setzt allerdings ein hohes Maß an Gestaltungsfähigkeiten voraus. Angefangen mit der Erstellung von Szenarien werden auf Basis dieser "UI-Mockups" gefertigt. Anhand derer werden mittels der Evaluationstechniken "Simplified Thinkaloud" und "Heuristic Evaluation" der Grad der Gebrauchstauglichkeit gemessen. Diese drei Aktivitäten werden chronologisch wiederholt, ohne dass ein Abbruchkriterium erkennbar ist. Die Entwickler haben eine enorme Verantwortung, da sie selber unvoreingenommen einschätzen können müssen, inwieweit das Vorgehensmodell überhaupt auf das Projekt skalierbar ist und vor allem wie die Qualität der Ergebnisse aus der Evaluation zu bewerten sind. Die mangelnde Erfahrung der Entwickler im Bereich des angewandten Usability Engineering und die Nähe zum Projekt stellen für die Wahl dieses Vorgehensmodells ein zu hohes Risiko dar. Trotzdem könnte es für die Evaluation des Projekts sinnvoll sein, das heuristische Verfahren einzusetzen, um Nielsens empirischen Erfahrungenen zu nutzen.

6.2 Usage-Centered Design

Das Usage-Centered Design verfolgt eine etwas andere Vorgehensweise. Es umfasst von vorneherein eine größere Benutzergruppe und nimmt damit keine Rücksicht auf spezielle Kenntnisse und Bedürfnisse eines spezifischen Benutzertypen. Der Werkzeugcharakter und die Funktionalität der Elemente einer Anwendung stehen im Mittelpunkt der Entwicklung.

6.2.1 Essential Models

Larry Constantine und Lucy Lockwood haben im wesentlichen drei essentielle Modelle in Beziehung gesetzt. Das Role Model, das Task Model und das Content Model. Diese werden durch das Operational Modell und das Implementation Model erweitert. Der Benutzer wird als Individuum betrachtet und nimmt gegenüber dem System eine definierte Rolle ein, sodass Rollen miteinander verglichen und in Beziehung gesetzt werden können. Für die descriptive Planung der Benutzung des Systems werden auf Szenarien verzichtet und stattdessen Use Cases modelliert, die im Task Model untersucht werden: Inwiefern kann das System Aufgaben übernehmen und wie stehen die Aufgaben zueinander. Im Content Modell wird die Modellierung konkret gemacht. Das heißt Tools und Objects werden aus den Aufgaben abgeleitet und festgelegt bzw. mit Attributen und Funktionalitäten definiert. Das und Einflüsse aus den Modellen zum Nutzungskontext (Operational Model) bilden die Basis für ein Implementation Model. In Bezug auf das Projekt bieten die Essential Models gute Voraussetzungen für solide Ergebnisse. Das strukturierte Anpassen des Systems an den Soll-Zustand, verläuft eng an der Schnittstelle zum Software Engineering. Die erwarteten Ergebnisse der Implementation sind dadurch bereits vorher gut abschätzbar. Das Risiko für das Projekt, dass die Anwendung letztendlich die Zielgruppe der Backpacker nicht erreicht und sich am Markt gegenüber einer größeren Nutzergruppe, aufgrund der hohen Konkurrenz nicht durchsetzen kann, ist allerdings ebenfalls gegeben.

6.3 Fazit

Das Projekt nähert sich einem sensiblen Thema an, nämlich dem Umgang mit Geld. Nutzer erwarten von so einem System ein Format, indem Funktionen eindeutig, verlässlich und absolut fehlerfrei ablaufen. Das Usage-Centered Design ist für Systeme dieser Art i.d.R eine gute Wahl, weil im Prinzip jedes Mitglied der modernen Gesellschaft bereits Kompetenzen im Umgang mit Geld besitzt. Dadurch herrschen gute Bedingungen, dass eine breite Nutzergruppe sich mit den Funktionen schnell vertraut machen und hohe Erwartungen an die Funktionalität erfüllt werden können. Jedoch zeigt die Marktanalyse, dass viele Systemen existieren, die sowohl in der Breite, wie auch in der Tiefe allgemeine qualitativ hochwertige Lösungen bereitstellen. Deswegen wurde die Domänenrecherche erweitert und sich auf die Zielgruppe der Backpacker konzentriert. Mit einem Vorgehensmodell des User-Centered Design kann die Benutzergruppe genauer untersucht werden, sodass sie gemeinsamen Umgang mit Geld im besonderen Nutzungskontext nicht scheut. Die ISO-Norm 9241 Teil 210 ist auf das Projekt sehr gut skalierbar, da sie Freiraum für die Wahl angemessener Methoden lässt, solange diese vorgegebene Kriterien erfüllen und vordefinierten Richtlinien entsprechen. Um den Nutzungskontext festzulegen werden im Projekt Benutzer- und Aufgaben-Modellierungen vorgenommen, sodass bspw. resultierende Personae in Use-Cases oder Szenarien integriert werden. Die daraus extrahierten Erkenntnisse sind für die Festlegung der Nutzungsanforderungen fundiert, welche wie

oben beschrieben (vgl. ISO-Norm 9241- 210) eine zentrale Rolle einnehmen. Bei der Erarbeitung von Gestaltlösungen, könnten bspw. UI-Mockups erstellt werden, die die im Projekt erarbeiteten Nutzungsanforderungen erfüllen und sich gleichzeitig an Richtlinien wie der ISO 9241 Teil 10 (Grundsätze der Dialoggestaltung) und 11 (Anforderungen an die Gebrauchstauglichkeit) orientieren. “Anticipated Use”, effektive und effiziente Aufgabenerledigung und die emotionale Bindung der Nutzer an das Produkt werden für das intendierte System angestrebt. Es sollte auf dieser Grundlage eine positive User Experience in der Evaluation gemessen werden, um mit dem intendierten System am Markt konkurrenzfähig zu sein. Die ISO-Norm 9241 bietet dafür adäquate Orientierungspunkte und ist letztendlich die Wahl der Entwickler.

7 Stakeholderanalyse

Zu untersuchende Stakeholder sind primär die direkte Nutzer (Priorität 1), sekundär Geschäftspartner aus dem Einzelhandel (Priorität 2) und tertiär externe Systeme, zu denen Schnittstellen gebildet werden sollen (Priorität 3). Es wird dabei berücksichtigt, dass Nutzer die Rolle des Gläubigers und/oder des Schuldners annehmen können, wenn sie mit dem intendierten System interagieren.

Bezeichnung	**Beziehung**	**Objektbereich**	**Erfordernisse/ Erwartungen**	**Priorität**
Alle Nutzer	Anrecht	persönliche Informationen	vertraulicher Umgang mit Informationen	1
	Anspruch	Abrechnung	korrekte automatische Berechnung der Abrechnung	1
		Kollaboration	alle involvierten Nutzer einladen und Daten teilen	1
		Kommunikation	Eindeutige Kommunikation mit Nutzern anderer Sprachen	1
		Währungstausch	Reibungslose Vermittlung	1
		Bezahldienst	Reibungslose Vermittlung	1
		Datenspeicherung	Persistenz und Konsistenz	1
	Interesse	Kassenzettel-Scan	automatische Datenerhebung	1
		Kommunikation	Behandlung von Konflikten	1
		Währungstausch	Anpassungsmöglichkeit an eine Währung	1
		Bezahldienst	Überweisungsmöglichkeit	1
Nutzer: Gläubiger	Anspruch	Datenspeicherung	Zugriff auf eigene digitalisierte Kassenzettel	1
		Kommunikation	Referenzierung und Beteiligung der/des Schuldner/s	1
	Interesse	Kollaboration	Abrechnung mit Schuldnern teilen und dokumentieren	1
		Kommunikation	Skepsis des Schuldners überwinden und Vertrauen stärken	1
		Kommunikation	Details des Abrechnungs-Verlaufs mitteilen	1
Nutzer: Schuldner	Anspruch	Kollaboration	Die Forderung nachvollziehen und überprüfen	1
		Kommunikation	Korrektur der Abrechnung	1
	Interesse	Kommunikation	Aufschiebung der Abrechnung erbitten	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		Einbettung/ Weiterleitung	Kundengewinnung	3
Bezahldienst		Einbettung/ Weiterleitung	Kundengewinnung	3

Tabelle 1: Stakeholderanalyse

7.1 Erfordernisse und Erwartungen

Durch die Stakeholderanalyse können erste Erfordernisse und Erwartungen abgeleitet werden. Diese werden im Rahmen der Methoden zur Festlegung des Nutzungskontextes nach der ISO-Norm 9241 Teil 210 später aufgegriffen, überprüft und erweitert werden können.
Jeder Benutzer muss ...

- wissen, dass mit seinen persönlichen Daten vertraulich umgegangen wird, um sich für die Benutzung der Anwendung zu entscheiden.
- auf den korrekten Algorithmus für die Abrechnung zugreifen können, sodass innerhalb der Reisegruppe korrekt und sicher Finanzen ausgeglichen werden können.
- Mitglieder im System finden können, um sie in eine Gruppe einladen und Daten mit ihnen teilen zu können.
- Methoden zur Verfügung haben, um eine eindeutige Kommunikation mit anderssprachigen Benutzern zu ermöglichen.
- Zugriff auf einen Währungsrechner haben, um den aktuellen Wechselkurs verschiedener Währungen abrufen zu können.
- auf einen Online-Bezahldienst weitergeleitet werden, um offene Abrechnung mit anderen Benutzern sicher begleichen zu können.
- davon ausgehen können, dass alle Daten persistent und konsistent gespeichert werden, dass diese zuverlässig abrufbar sind.

- die Möglichkeit haben Kassenzettel zu scannen, um die Daten daraus automatisch erheben zu können.
- die Möglichkeit für eine eindeutige Kommunikation mit anderen Benutzern zur Verfügung haben, um potentielle Konflikte vorbeugen oder behandeln zu können.
- die Möglichkeit haben im Algorithmus unterschiedliche Währungen anzugeben, um die Korrektheit der Abrechnung zu garantieren.
- die Möglichkeit haben die Überweisung auf einem externen Bezahldienst umzuleiten (z.B. PayPal), um die Abrechnung zu vollziehen.

Ein Benutzer in der Rolle eines Gläubigers muss ...

- den von ihm digitalisierten Kassenzettel zur Verfügung haben, um andere Gruppenmitglieder für eine Abrechnung referenzieren/beteiligen zu können.
- auf alle in der Abrechnung involvierten Artikel der Kassenzettel zur Verfügung haben, um diese referenzieren und mit den Schuldnern teilen zu können, sodass die Abrechnung transparent und korrekt wird.
- In der Abrechnung die präferierte Methode der Begleichung und bevorzugte Währungen angeben können, um einen eindeutigen Verlauf der Abrechnung zu kommunizieren.

Ein Benutzer in der Rolle eines Schuldners muss ...

- die an ihn gestellten Forderungen einer Abrechnung einsehen können, um nachvollziehen und überprüfen zu können, ob diese korrekt sind.
- de Möglichkeit haben fehlerhafte Einträge in der Abrechnung markieren zu können und dem Gläubiger mitzuteilen, sodass dieser die Fehler überprüfen und ggf. korrigieren kann.
- Die Möglichkeit haben die Gläubiger um eine zeitliche Verschiebung der Schuldenbegleichung zu bitten, um einen Konflikt präventiv zu vermeiden.

7.2 Anforderungen

7.2.1 Funktionale Anforderungen

Aus den ersten Erfordernissen und Erwartungen resultierende Anforderungen, sind ein erstes Ergebnis, dass helfen kann den Soll-Zustand des Systems konkreter planen zu können. Der Kommunikationsverlauf und die Details der Systemarchitektur können fundiert überlegt und angepasst werden.
Das System muss...

- den Algorithmus für die Abrechnung gebrauchstauglich anbieten.

- den Algorithmus an verschiedene Währungen anpassen können.
- Nutzern ermöglichen andere Nutzer im System zu finden.
- Methoden bereitstellen die eine eindeutige Kommunikation mit anderssprachigen Benutzern ermöglicht.
- Externe Services einbinden bzw. eine korrekte Weiterleitung ermöglichen.
- eine Scan-Funktion und automatische Datenverarbeitung anbieten.
- eine Scan-Funktion und automatische Datenverarbeitung anbieten
- Nutzern ermöglichen andere Nutzer auf eine Abrechnung zu referenzieren.

7.2.2 Qualitative Anforderungen

- persönliche Daten vertraulich behandeln.
- Daten persistent und konsistent speichern.

7.2.3 Organisationale Anforderungen

- Nutzer authentifizierbar und verifizierbar ins System integrieren.
- Schnittstellen zum soziotechnischen System ermöglichen.

8 Kommunikationsverlauf

Anhand der Kommunikationsmodelle soll der Nachrichtenaustausch zwischen Nutzern und Instanzen des Systems verdeutlicht werden. Das deskriptive Modell zeigt den Ist-Zustand, also wie die Kommunikation ohne unseren Lösungsansatz aussieht. Das präskriptive Modell zeigt dagegen den Soll-Zustand, also wie wir den Vorgang mit der Einbindung unseres Systems verbessern wollen.

8.1 Deskriptives Kommunikationsmodell

Im deskriptiven Modell (Abb. 6) ist ein Großteil der Kommunikation synchron. Da das Vertrauen bei den Beteiligten eventuell noch nicht besonders gut ausgeprägt ist, warten fast alle Beteiligten auf die direkte Antwort ihres Kommunikationspartners. Aufgrund von möglichen Sprachbarrieren, müsste ein Dolmetscher (entweder aus der Reisegruppe oder sogar extern) einbezogen werden. Dies macht die Kommunikation noch komplizierter und anfällig für falsche Interpretationen oder Mißverständnisse. Externe Services wie Währungsrechner bzw. -tauscher oder (Online-) Banking, in Form einer Person oder einer Anwendung, müssen separat kontaktiert bzw. benutzt werden.

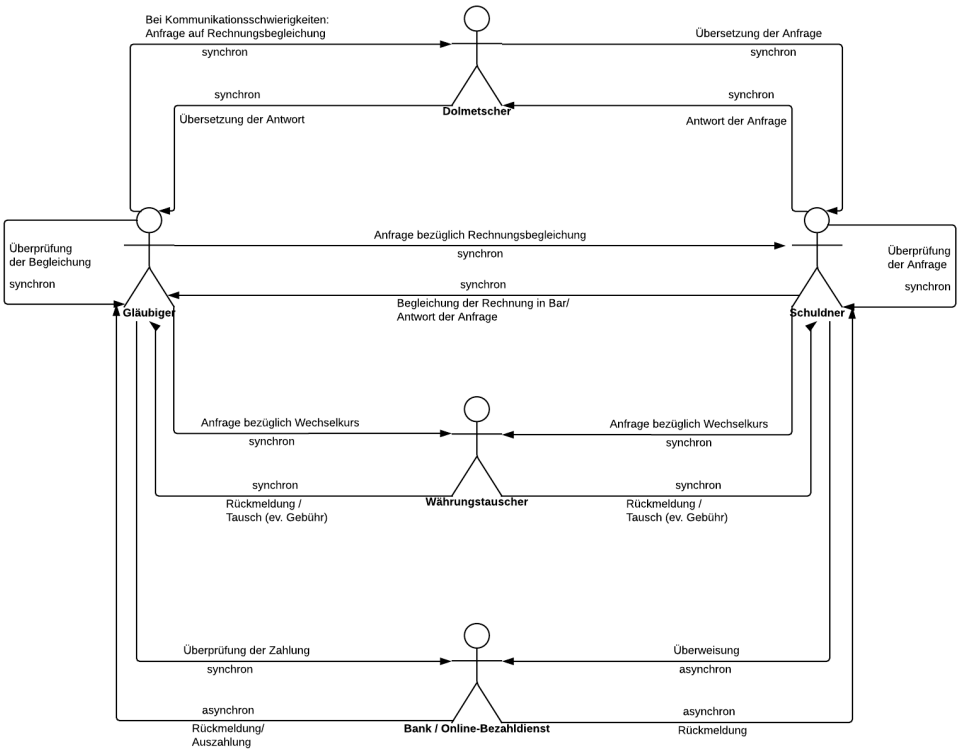

Abbildung 6: Deskriptives Kommunikationsmodell

8.2 Präskriptives Kommunikationsmodell

Man kann erkennen das im präskriptiven Modell (Abb. 7) die Instanz des Dolmetschers komplett wegfällt und von unserem System übernommen werden würde. So soll das Risiko der Fehlinterpretation bzw. der falschen Übersetzung durch vorgefertigte, klar verständliche Dialoge, welche automatisch in die vom Nutzer präferierte Sprache übersetzt werden, minimiert oder gänzlich eliminiert werden. Des Weiteren soll die Kommunikation über unser System so weit wie möglich zeitlich entkoppelt, also asynchron, sein. So sollen die Nutzer sich mehr auf die positiven Aspekte der Reise konzentrieren können. Dies ist jedoch nur möglich, wenn das System die Bildung des Vertrauens der Nutzer unterstützt. Zudem übernimmt das System die Weiterleitung und direkte Kommunikation zu den externen Diensten Währungsrechner und Online-Bezahldienst, was dem Nutzer einige Interaktionsschritte ersparen würde.

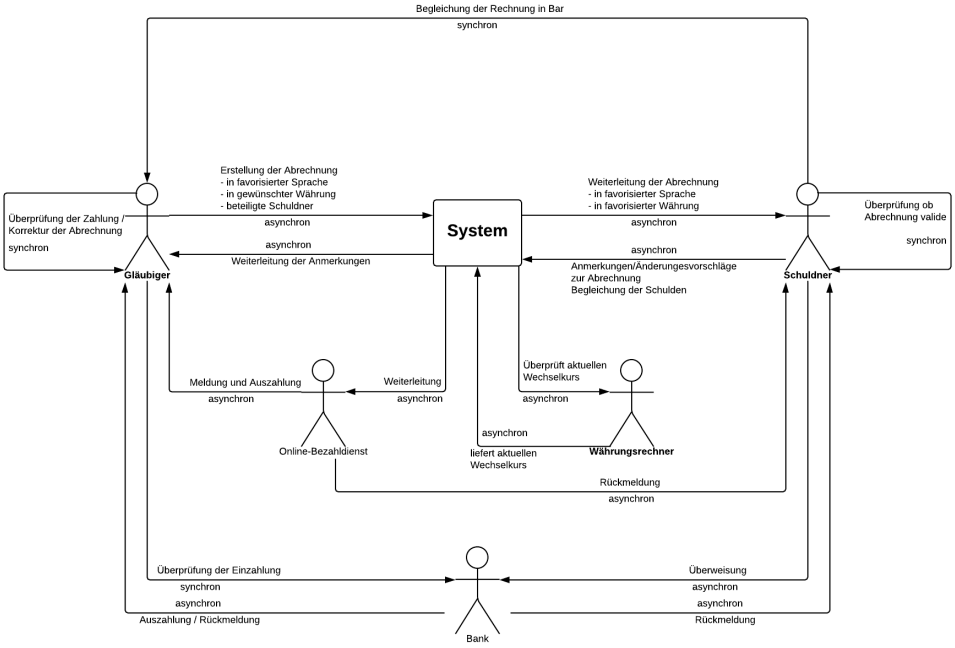

Abbildung 7: Präskriptives Kommunikationsmodell

9 Systemarchitektur

Die Systemarchitektur soll abwägen, welche Softwarekomponenten es geben wird, wie diese miteinander kommunizieren und welche Daten bzw. Informationen untereinander ausgetauscht werden. In Abb. 8 wird verdeutlicht, welche Systemkomponenten letztendlich implementiert werden sollen.

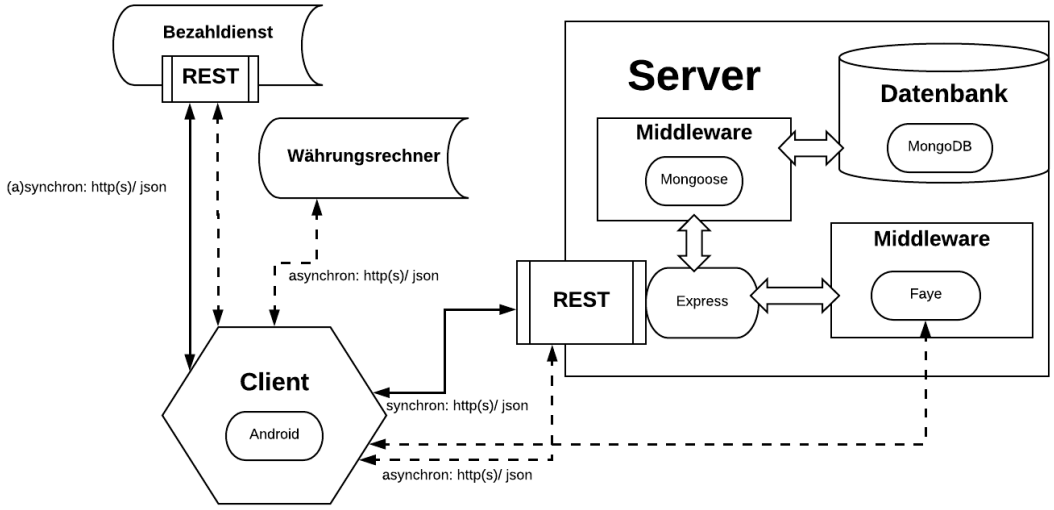

Abbildung 8: Architektur-Modell

9.1 Server

Um den Datenaustausch zwischen Clients, Datenbank und Webservice zu garantieren und gleichzeitig diese Daten qualitativ anreichern zu können, ist ein zentraler Server nötig. Für die Implementierung wird die Plattform Node.js verwendet. Zum Einen eignet sie sich für das Projekt aufgrund ihrer Performanz und Skalierbarkeit, als auch Dank der guten Dokumentation und dem großen Umfang an integrierbaren Modulen, welche durch npm zur Verfügung stehen. Außerdem bietet sie asynchrone Funktionen an, welche für uns eine große Rolle spielen. Als Programmiersprache wird, aufgrund der verwendeten Plattform und bereits vorhandener Erfahrungen, Javascript verwendet.

9.2 Client

Die Clients haben die Anforderung mobil zu sein, um vor allem auf Reisen genutzt werden zu können. Da heutzutage so gut wie niemand mehr ohne Smartphone das Haus verlässt, soll er in Form einer Smartphone-Applikation implementiert werden. Zudem

sollte der Client nativ sein und kein Web-Hybrid, da Offline-Funktionen vorhanden sein sollen und native Anwendungen generell performanter sind und bessere Dokumentationen bieten. Dafür stehen bestimmte Betriebssysteme zur Verfügung: Android, iOS und Windows. Unser Team besitzt bereits Vorkenntnisse im Bereich Android-Development und Android ist derzeit das Betriebssystem, das in den meisten verkauften mobilen Endgeräten vorinstalliert (87,7 Prozent Marktanteil [16]) ist. Daher fällt die Wahl im Rahmen dieser Veranstaltung auf Android, wobei eine zukünftige Implementierung auf allen Betriebssystemen wünschenswert wäre, um möglichst viele Nutzer anzusprechen. Da die Android-Dokumentation für die Sprache Java verfasst wurde und das Team darin die größte Erfahrung hat, verwenden wir sie für die Implementation des Clients.

9.3 Datenbank

Für die persistente Speicherung der Daten wird MongoDB, eine dokumentenbasierte NoSQL-Datenbank, verwendet. Sie bietet sich unter anderem aufgrund ihrer einfachen Einbindung in Node.js, als auch durch ihre hohe Performanz und effektivem Querrying an. Zum anderen lässt sich die Form unserer Daten (Kassenzettel, Nutzerdaten, etc) sehr gut in einem JSON-Format speichern, welches von MongoDB, in Form von BSON, unterstützt wird. Da MongoDB allerdings keine festen Schematas verwendet, kann jeder Eintrag einen anderen Datentyp besitzen. Um die Form der Daten trotzdem besser validieren und kontrollieren zu können nutzen wir die Middleware-Erweiterung Mongoose.

9.4 Dateisystem

Im intendierten System soll nicht nur mit JSON-Daten umgegangen werden, sondern auch mit Bildern. Deren Speicherung in JSON bzw BSON-Format ist nicht optimal, da das Bild in einen Blob von binären Daten dekodiert und bei jedem Zugriff wieder zusammengesetzt werden muss. Die effektivere Lösung ist ein Dateisystem neben der verwendeten Datenbank zu erstellen. Die Verknüpfung von Dateien und ihren Metadaten erfolgt über eine Referenz in JSON. Die Aufspaltung der Speichersysteme erlaubt bessere Skalierung und individuelle Regeln für Caching.

9.5 Webservices

Es ist geplant, dass im System mind. zwei externe Webservices anhand ihrer API im System eingebunden werden. Zum Einen soll ein Bezahldienst für den Nutzer verfügbar sein um Abrechnungen durchführen zu können. Hier preferieren die Entwickler Paypal, aufgrund der weiten Verbreitung des Dienstes, der guten Dokumentation und der, mit unseren System übererinstimmenden, Schnittstellen und Programmiersprachen (REST, Nodejs, Java) [17]. Zum Anderen soll ein Währungsrechner genutzt werden, mit dem die Nutzer bei Bedarf jede Währung, zum aktuellen Tauschkurs, umrechnen können.

Dafür soll die API currencylayer verwendet werden. Auch hier ist eine ausführliche Dokumentation vorhanden und es werden die von uns eingesetzten Programmiersprachen unterstützt [18]. Außerdem sind beide Services kostenlos (currencylayer bis zu 1000 Zugriffe im Monat).

9.6 Protokolle

Für die Kommunikation zwischen allen Systemkomponenten und externen Webservices soll das Protokoll HTTP / HTTPS verwendet werden. Es ist ein sehr verbreiteter und gut dokumentierter Standard im Internet, eignet sich für eine REST-Architektur und ermöglicht die Übertragung der von uns genutzten Dateiformate JSON und Bilder (jpeg, png, etc), sowie Authentifizierung der Nutzer.

9.7 Kommunikation

Die Kommunikation zwischen Client und Server bzw den eingebunden Webservices kann entweder synchron oder asynchron verlaufen. Dabei blockiert synchrone Kommunikation den Client solange bis der Server seine Antwort sendet. Während die asynchrone zeitlich entkoppelt ist und dem Nutzer währenddessen Freiraum für andere Aktivitäten bietet.
synchron:
Für diese Art der Kommunikations eignen sich Vorgänge, bei denen Sicherheit eine Rolle spielt. Bei der Authentifizierung muss der Nutzer erst die Bestätigung vom Server abwarten, bevor der Zugang zu sensiblen Daten und Nutzungsrechten gewährt werden kann. Auch beim Bezahlvorgang der Abrechnungen muss Synchronität bewahrt werden um mögliche invalide Zahlungen zu vermeiden.
asynchron:
Hier bieten sich alle restlichen (nicht sicherheitsrelevanten) Vorgänge an. Der Nutzer sollte nicht abwarten müssen, bis sein Upload des Kassenzettels beendet ist um andere Aktionen ausführen zu dürfen. Solange der Server den Vorgang registriert (HTTP-Statuscode 202 Accepted) kann die Rückmeldung für erfolgreiche oder fehlerhafte Speicherung der Daten auch später erfolgen. Dies funktioniert auch im Zusammenhang der Offline-Features (automatischer Upload sobald Verbindung besteht). Auch die Abfrage der Daten sollte asynchron verlaufen. So kann die UI automatisch aktualisiert werden, sobald sich die Daten auf dem Server ändern. Die Kommunikation und Interaktion der Nutzer im Bereich der Abrechnungen sollte mit einem PubSub-Service verbunden werden, sodass keine sofortige Rückmeldung aller involvierten Nutzer erforderlich ist. Für die Implementierung der asynchronen Kommunikation zwischen einem Android-Client und dem Nodejs Server können Callbacks und Long-Polling verwendet werden.

10 Risiken

10.1 Konnektivität

Auf mobilen Clients und besonders auf Reisen kann eine stabile Internetverbindung nicht garantiert werden. Es kann vorkommen, dass genau zum Zeitpunkt eines Downloads von bzw. Uploads zu dem Server die Verbindung abbricht und die Daten verloren gehen und/oder Race-Conditions auftreten. So können essentielle Funktionen des Systems nicht verwendet werden und die Synchronisation mit anderen Benutzern ist nicht möglich.

10.2 Speicherplatz

Trotz der heutzutage geringen Kosten von Speicherplatz ist er nicht unendlich groß oder verfügbar. Man muss sich bei der Konzeptionierung Gedanken machen, wie man diesen verwaltet und ob/wann man bestimmte Daten wieder löschen kann. Vor allem mobile Clients haben noch sehr begrenzte Kapazitäten und können dadurch, aufgrund von einer suboptimalen Nutzung der Ressourcen, an Performance-Schwächen leiden.

10.3 Webservices

Trotz der Möglichkeit die Webservices kostenfrei in das System einzubinden, können dabei einige Risiken auftreten. So ist zum einen der Datenschutz und die Verfügbarkeit zu berücksichtigen. Wie geht man damit um, wenn der Dienst ausfällt oder die Nutzungskapazität überschritten wird? Zudem kann es im Entwicklerteam Probleme mit der Einarbeitung in die externe API geben oder Inkompatibilität mit dem restlichen System verursachen. Daher sollte man die wichtigsten Features selbst implementieren und nur auf Webservices zurückgreifen, auf die man Notfalls auch verzichten kann und das Alleinstellungsmerkmal somit nicht gefährden.

10.4 OCR

Die Technik OCR (optical character recognition) kann für die automatische Erkennung und Verarbeitung von Schrift verwendet werden. Auf dem Client sollen so Kassenzettel gescanned werden. Obwohl an dieser Technik bereits seit einigen Jahren geforscht wird, gibt es keine Garantie, dass alle Daten korrekt erkannt werden, da viele externe Faktoren (Licht, Oberfläche, Abnutzung, etc) eine Rolle spielen und das Ergebnis verfälschen können. Zudem ist die Technologie ziemlich komplex und es gibt im Team keine direkten Vorkenntnisse, was die Implementation erschweren könnte.

10.5 Datenbanken/Querying

In der geplanten Anwendung sollen viele Daten wie Bilder, Kassenzettel, Benutzerdaten, etc bearbeitet werden können und abrufbar sein. Je größer der Datensatz wird bzw. je komplexer die Daten werden, desto höher ist die Wahrscheinlichkeit für Komplikationen (lange Ladezeiten etc). Daher ist eine Datenbank nötig, die von der Größe skalierbar ist und das Querying ermöglicht. Jedoch ist das Querying bei komplexen und verschachtelten Daten oft sehr schwer und unübersichtlich. Zudem wurden im Team bisher noch keine umfangreiche Datenbank implementiert, so das dafür zum Teil Vorwissen fehlen könnte.

10.6 Abrechnung

In der Anwendung soll der Prozess der Abrechnung gemeinsam gekaufter Artikel vereinfacht werde, aber es besteht immer noch die Gefahr, dass Unstimmigkeiten über die Korrektheit der Abrechnung zwischen den involvierten Nutzern aufkommen. Der Algorithmus könnte die Abrechnung falsch kalkulieren, nicht die korrekten Daten verwenden oder die falschen Nutzer involvieren. Zudem könnte bei einer unvorteilhaften Präsentation des Ergebnisses Unstimmigkeiten und Missverständnis zwischen den Nutzern aufkommen.

10.7 Arbeitsaufwand

Aufgrund der hohen Anzahl an verschiedenen Features und Anforderungen an das System könnte der Arbeitsaufwand unterschätzt werden und im Rahmen dieser Veranstaltung nicht realisierbar sein. Daher sollten bestimmte Features der Anwendung anhand von der Ausprägung des Alleinstellungsmerkmals und den Voraussetzungen der Veranstaltung priorisiert werden.

10.8 Konkurrenzfähigkeit

Im Vergleich zu den etablierten Systemen, die in der Marktrecherche gefunden werden konnten, sind nach Einschätzung der Entwickler deutliche Verbesserungen und eine Anpassung an eine spezifische Nutzergruppe ein starkes Argument für ein neues System. Jedoch besteht das Risiko bei der Entwicklung des Systems, sich gegenüber der Konkurrenz nicht deutlich genug abzuheben, sodass die Verbesserungen und Anpassungen von den Nutzern als Überflüssig erachtet werden könnten.

11 Proof of Concept

11.1 Abrechnung

Damit ein Benutzer an einer Abrechnung mit einbezogen werden kann, muss dieser entweder in den betreffenden Artikeln eines Kassenzettels oder allgemein in der Abrechnung referenziert werden. Nachdem alle benötigten Teilnehmer in der Abrechnung enthalten sind, soll ein Algorithmus die entsprechende Kostenverteilung pro Benutzer anhand der Artikelpreise ausrechnen und die entstehende individuelle Abrechnung an die jeweilige Nutzer senden.

Exit-Kriterium 1:
Der Algorithmus hat die Kostenverteilung anhand der referenzierten Kassenzetteln und Nutzern sowie Parametern (z.B prozentualer Verbrauch) korrekt ausgerechnet. Dem Nutzer ist es möglich das Ergebnis bei Bedarf zu überprüfen.
Exit-Kriterium 2:
Das individuelle Ergebnis des Algorithmus wird an den Beteiligten gesendet.
Exit-Kriterium 3:
Die Ergebnisse werden von allen Beteiligten anerkannt.

Fail-Kriterium 1a:
Der Algorithmus hat die Kostenverteilung anhand von Parametern falsch ausgerechnet.
Fallback 1a:
Das Ergebnis des Algorithmus, mit allen berücksichtigten Parametern, sollte für alle Beteiligten sichtbar sein. Wenn eine falsche Rechnung anhand inkorrekter Parameter erkannt wird, sollte der Algorithmus mit korrigierten Parametern erneut ausgeführt werden können.

Fail-Kriterium 1b:
Der Beteiligte kann sich im individuellen Ergebnis der Abrechnung nicht selbst identifizieren, bekommt keine Einsicht in die referenzierten Artikel oder erkennt falsch referenzierte Artikel.
Fallback 1b:
Der Beteiligte sollte den Ersteller/Moderator der Abrechnung benachrichtigen und anhand der markierten Fehler eine Korrektur verlangen können.

Fail-Kriterium 2a:
Wenn sich der Beteiligte im individuellen Ergebnis der Abrechnung nicht selbst identifizieren kann, ist anzunehmen, dass diese an den falschen Beteiligten gesendet wurde.
Fallback 2a:
Der Beteiligte sollte den Ersteller/Moderator der Abrechnung benachrichtigen und auf den Fehler hinweisen können, sodass dieser den korrekten Nutzer referenzieren und beteiligen kann. Ansonsten kann der Beteiligte die Rechnung direkt an den korrekten Nutzer weiterleiten, falls dieser bekannt ist und der Moderator dies akzeptiert.

Fail-Kriterium 2b:
Die Ergebnisse wurden aufgrund von fehlender Internetverbindung nicht an alle Beteiligten gesendet werden.
Fallback 2b:
Wenn das System erkennt, dass die Ergebnisse der Abrechnung nicht an gewisse oder alle Beteiligten gesendet werden konnte, sollte das System es nach einer gewissen Zeit erneut probieren. Wenn mehrere Versuche fehlschlagen muss der Ersteller/Moderator vom System davon in Kenntnis gesetzt werden.

Fail-Kriterium 3a:
Das Ergebnis der Abrechnung wurde von einem, mehreren oder allen Beteiligten nicht anerkannt aufgrund eines Fehlers oder Betrugsversuchs.
Fallback 3a:
Der Beteiligte muss einen Fehler markieren und dem Ersteller um Korrektur bitten bzw. selbst eine Korrektur anbieten. Wenn dies vom Ersteller nicht angenommen wird, kann ein Voting-System eingeführt werden, dass alle Beteiligten betrifft.

11.2 Konnektivität

Um Daten vom Server beziehen oder speichern bzw. um sich mit anderen Benutzern aus der selben Reisegruppe synchronisieren zu können, muss eine stabile Internetverbindung oder eine Verbindung mit dem Server vorhanden sein. Die Kommunikation darf nicht während der Übertragung der Daten unterbrochen werden. Falls dies doch geschieht, müssen die entstehenden Komplikationen behandelt werden.

Exit-Kriterium 1: Der Client konnte vom Server erfolgreich Daten beziehen und diese dem Benutzer lokal präsentieren oder speichern.
Exit-Kriterium 2:
Der Client konnte Daten auf dem Server erfolgreich speichern und erhielt eine Bestätigung dafür.

Fail-Kriterium 1a:
Der Client konnte aufgrund fehlender Internetverbindung oder eines Absturz des Servers keine Verbindung aufbauen und hat keinen Zugriff auf dessen Daten oder Funktionen.
Fallback 1a:
Der Client sollte für einen bestimmten Zeitraum wiederholt versuchen Verbindung mit dem Server oder dem Internet aufzubauen. Wenn dies in dem Zeitraum nicht erfolgreich war, muss der Benutzer dies mitgeteilt werden, sodass er mit dem Fehler selbst umgehen kann.
Fail-Kriterium 2a:
Der Client konnte aufgrund fehlender Internetverbindung oder eines Absturz des Servers keine Verbindung aufbauen oder verlor diese während des Uploads von Daten.
Fallback 2a:

Der Client sollte für einen bestimmten Zeitraum wiederholt versuchen Verbindung mit dem Server oder dem Internet aufzubauen. Währenddessen müssen die Daten lokal zwischengespeichert werden, sodass diese nicht verloren gehen. Wenn dies in dem Zeitraum nicht erfolgreich war, muss dem Benutzer dies mitgeteilt werden.

11.3 Speicherplatz

Jedesmal wenn auf dem Server oder Client Daten gespeichert werden müssen, ist es notwendig zu überprüfen, ob überhaupt genug Speicherplatz dafür verfügbar ist. Wichtige bzw aktuelle Daten dürfen nicht verloren gehen, wenn der Speicher voll ist.

Exit-Kriterium 1:
Die Speicherplatz auf dem Server ist skalierbar (zusätzlicher Server/Festplatten) und/oder noch nicht voll. Die Daten wurden erfolgreich gespeichert.
Exit-Kriterium 2:
Der Speicher auf dem Client ist skalierbar (größere SD-Karte) und/oder noch nicht voll. Die Daten wurden erfolgreich gespeichert.

Fail-Kriterium 1a:
Auf dem Server ist der Speicherplatz nicht mehr skalierbar und voll. Neue Daten können nicht mehr gespeichert werden.
Fallback 1a:
Der Administrator der Datenbank sollte rechtzeitig eine Meldung vom Server bekommen, um mehr Speicherplatz zur Verfügung zu stellen oder alte, nicht mehr benötigte, Daten zu komprimieren oder zu löschen. Das System könnte auch automatisch ermitteln und festlegen, wann Daten nicht mehr benötigt werden und diese daraufhin aus der Datenbank löschen.

Fail-Kriterium 2a:
Auf dem Client ist der Speicherplatz nicht mehr skalierbar und voll. Neue Daten können nicht mehr gespeichert werden.
Fallback 2a:
Der Benutzer des Clients sollte rechtzeitig eine Meldung vom Client bekommen, um mehr Speicherplatz zur Verfügung zu stellen oder nicht mehr benötigte Daten zu löschen. Der Client sollte auch nur die aktuellsten, für den Nutzer relevanten, Daten lokal speichern und diese automatisch löschen, sobald dies nicht mehr der Fall ist.

11.4 Webservices

Das System soll mit mehreren Webservices in Verbindung stehen, um auf Funktionen zugreifen zu können, deren eigene Implementation sich nicht lohnen würde. So soll die Begleichung der Abrechnungen über einen externen Online-Bezahldienst vollführt werden

können. Außerdem sollen Nutzer den aktuellsten Umtauschkurs zwischen den angegebenen Währungen über eine API-Schnittstelle abrufen können.

Exit-Kriterium 1:
Der Gläubiger einer Abrechnung kann an alle involvierten Schuldner eine Referenz an einen Online-Bezahldienst schicken, sodass diese die Rechnung direkt darüber begleichen können.
Exit-Kriterium 2:
Nutzer können bei Bedarf und vorhandener Internetverbindung für eine angegebene Währung und einen bestimmten Wert den aktuellen Umtauschkurs in die Währungen ihrer Wahl abrufen. Das Abrufen der Daten erfolgt in weniger als 1 Sekunde.

Fail-Kriterium 1a:
Die Referenz an den verwendeten Bezahldienst des Gläubigers ist nicht korrekt oder konnte nicht an die Schuldner versendet, bzw. von diesen verwendet werden (z.B wird der Bezahldienst von ihnen nicht genutzt).
Fallback 1a:
Der Gläubiger bekommt eine Nachricht, wenn die Referenz an den Online-Bezahldienst nicht wahrgenommen wurde. Dieser kann nun eine Begleichung der Schuld in Bargeld verlangen und dies den Schuldnern über das System mitteilen.

Fail-Kriterium 2a:
Die Währungsrechner-API ist nicht ansprechbar und/oder liefert nicht die gewünschten Daten.
Fallback 2a:
Der Nutzer erhält eine Nachricht, dass der Dienst nicht zur Verfügung steht oder dass zu den gewünschten Kriterien keine Daten gefunden wurden. Das System bietet dem Nutzer die Möglichkeit den Dienst erneut zu kontaktieren oder zeigt alte, noch im System gespeicherte Daten zu den Kriterien an, mit einer Warnung, dass diese nicht mehr valide sein könnten.

11.5 OCR

Um die Daten der Kassenzettel automatisch erfassen zu können, müssen diese fehlerfrei, anhand von Optical Character Recognition Algorithmen erkannt werden. Auch die korrekte Abrechnung der Kosten basiert auf dieser Funktion, da die genauen Preise einzelner Artikel erkannt werden müssen.

Exit-Kriterium 1:
Alle relevanten Daten eines Kassenzettels werden anhand eines Kamerabildes des Clients erkannt. **Exit-Kriterium 2:**
Der Nutzer hat die Möglichkeit die Daten zu überprüfen bevor diese gespeichert und weiterverarbeitet werden.

Fail-Kriterium 1a:
Es wurden nicht alle relevanten Daten korrekt erkannt.
Fallback 1a:
Der Nutzer sollte die Möglichkeit haben den OCR-Algorithmus erneut anzuwenden, ein besseres Bild erstellen oder Daten per manueller Eingabe ändern zu können.

Fail-Kriterium 2a:
Dem Nutzer wurde keine Möglichkeit die Daten zu überprüfen.
Fallback 2a:
Das System muss dem Nutzer die Möglichkeit bieten, falsche, auf dem Server gespeicherten, Daten nachträglich zu ändern oder zu löschen.

11.6 Datenbanken/Querying

Für die Wahl der Datenbank muss deren Leistung für bestimmte Anwendungsfälle, die für die Entwickler und Nutzer von Bedeutung sind, berücksichtigt werden. So muss die Datenbank z.B schnelle Antworten liefern und komplexe Anfragen unterstützen können.

Exit-Kriterium 1:
Jede Anfrage an die Datenbank soll nach höchstens zwei Sekunden beantwortet werden.
Exit-Kriterium 2:
Komplexe Queries, wie z.B die Schuldner-Id einer Abrechnung von bestimmten Artikeln eines bestimmten Kassenzettels kann den Namen des Schuldners referenzieren.

Fail-Kriterium 1a:
Die Beantwortung der Anfrage an die Datenbank benötigt länger als zwei Sekunden.
Fallback 1a:
Die Anwendungslogik der Datenbank muss optimiert werden oder eine eine leistungsfähigere Art der Datenverwaltung implementiert werden.

Fail-Kriterium 2a:
Komplexe Queries sind nicht möglich oder liefern falsche Ergebnisse.
Fallback 2a:
Die Anwendungslogik der Queries muss optimiert werden und/oder das Datenschema muss umstrukturiert werden, um einfachere Queries zu ermöglichen.

12 Umsetzung des vorgezogenen PoC

Der Proof of Concept 'Abrechnung' behandelt das wichtigste Risiko, da das Alleinstellungsmerkmal davon abhängt. Daher wurde dieser bei der Implementierung des Rapid Prototypen priorisiert. Für den Test des Exit-Kriteriums 1 wurden zu einer Abrechnung zwei verschiedene Kassenzettel mit je zwei verschiedenen Artikeln hinzugefügt. An deren Kosten waren wiederum zwei Nutzer mit unterschiedlichen Prozentzahlen beteiligt (siehe Abb. 9).


```
_id: ObjectId("5af2f5d2bea66a11dcd65157")
date: 2018-05-09 15:19:32.571
owner: ObjectId("5aee1cdd2a47720b64b13c31")
store: "REWE"
imagePath: "nicht implementiert"
article: Array
  0: Object
    category: Array
      _id: ObjectId("5af2f5d2bea66a11dcd6515b")
      name: "Milch"
      price: 0.78
      amount: 12
      priceAll: 4.36
    participation: Array
      0: Object
          _id: ObjectId("5af2f5d2bea66a11dcd6515d")
          participant: ObjectId("5aee3db751bfa727f804afca")
          percentage: 0.5
      1: Object
          _id: ObjectId("5af2f5d2bea66a11dcd6515c")
          participant: ObjectId("5aee3f1708bfb6240cdde9ee")
          percentage: 0.5
  1: Object
    category: Array
      _id: ObjectId("5af2f5d2bea66a11dcd65158")
      name: "Cola"
      price: 0.78
      amount: 12
      priceAll: 9.36
    participation: Array
      0: Object
          _id: ObjectId("5af2f5d2bea66a11dcd6515a")
          participant: ObjectId("5aee3db751bfa727f804afca")
          percentage: 0.5
      1: Object
          _id: ObjectId("5af2f5d2bea66a11dcd65159")
          participant: ObjectId("5aee3f1708bfb6240cdde9ee")
          percentage: 0.5
total: 18.72
payment: 20
change: 1.28
__v: 0
```


Abbildung 9: Datensatz eines Kassenzettels

Jedes mal wenn der Algorithmus der Abrechnung mit einem PATCH auf http://localhost:8081/settlement/5af2f75e3a230a04d0edb648/calc ausgeführt wurde, erhielten alle Clients, welche diese Abrechnung subscribed hatten, folgende Nachricht:

(siehe Abb. 10)

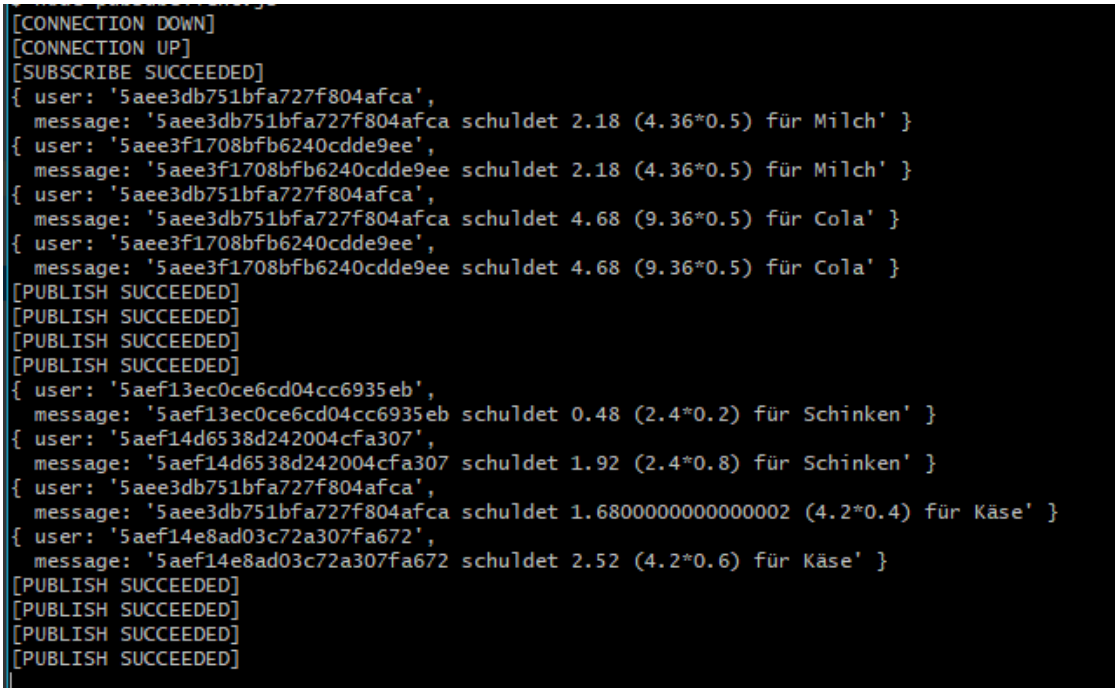

Abbildung 10: Output eines Clients

Der Algorithmus hatte den zu zahlenden Betrag jedesmal korrekt, anhand des Gesamtpreises und dem prozentualen Anteil des Nutzers, ausgerechnet. Durch die präsentierten Informationen des Outputs kann der Nutzer dies auch nachrechnen. Ein genauerer bzw. geordneterer Output konnte aus Zeitmangel leider nicht umgesetzt werden. Falls inkorrekte Parameter anhand des Outputs erkannt werden, ist es möglich diese mit einem PATCH-Requests zu ändern, falsche Abrechnungsdaten mit einem DELETE-Request zu löschen und den Algorithmus erneut auszuführen. Exit-Kriterium 1 und Fallback 1a gelten somit als implementiert und funktionstüchtig. Exit-Kriterium 2 konnte aus Zeitmangel ebenfalls nicht umgesetzt werden. Bisher war es den Entwicklern nur möglich jeden abgerechneten Artikel an jeden beteiligten Nutzer zu schicken, wie es im Fallback-Kriterium 1a beschrieben ist. Dies dient zwar der Transparenz des Systems, was auch eine Erwartung der Zielgruppe ist, aber in Zukunft sollte der Nutzer die Möglichkeit haben, den Output selbst zu bestimmen. Exit-Kriterium 3 konnte bisher nur statisch umgesetzt werden. So ist die Bestätigung des Nutzers noch fest im System implementiert. Dies zeigt zwar, dass die Kommunikation bereits funktioniert, aber es ist nicht das gewünschte Endprodukt des Systems.Anhand den Erkenntnissen dieses Proof of Concepts sind sich die Entwickler jedoch sicher, dass alles im Rahmen des Projekts erreichbar ist.

13 Tabellen- und Abblidungsverzeichnis

Tabelle

Abbildungen

Literatur

[1] "Braucht man den Kassenbon?" https://www.n-tv.de/ratgeber/Braucht-man-den-Kassenbon-article19937290.html, nTV; Aufgerufen am: 2018-04-21.

[2] "Bons gehoeren nicht ins Altpapier," https://www.umweltbundesamt.de/themen/bons-gehoeren-nicht-ins-altpapier, bundesamt, Umwelt; Aufgerufen am: 2018-04-21.

[3] "Warum Sie Ihren Kassenbon nicht im Geldbeutel aufbewahren sollten," https://www.focus.de/finanzen/videos/verbraucherschuetzer-warnen-warum-sie-ihren-kassenbon-nicht-im-geldbeutel-aufbewahren-sollten_id_6387651.html, online, Focus; Aufgerufen am: 2018-04-21.

[4] "Alleine reisen und doch nie einsam," https://www.planetbackpack.de/alleine-reisen-und-doch-nie-einsam-5-tipps-fur-solo-backpacker/, planet Backpack; Aufgerufen am: 2018-04-24.

[5] "Backpacker Australien," http://www.in-australien.com/backpacker_10218, in Australien; Aufgerufen am: 2018-04-24.

[6] "Alleine reisen ? So findest du Anschluss zu anderen Reisenden," https://www.travel-forever.de/alleine-reisen-so-findest-du-anschluss-zu-reisenden/, travel Forever; Aufgerufen am: 2018-04-24.

[7] "Ohne Mama nach Australien," https://www.youtube.com/watch?v=IEoS2VF9_rQ, youtube (ref. NDR); Aufgerufen am: 2018-04-24.

[8] "Friendcash," https://www.iphone-ticker.de/iphone-app-geld-ausgaben-verteilen-37159/, iphone-Ticker; Aufgerufen am: 2018-04-27.

[9] "Tricount," https://play.google.com/store/apps/details?id=com.tribab.tricount.android&utm_source=website&utm_medium=home&utm_content=lang%3Ade%2Cbtn%3Abottom-btn&%24fallback_url=https%3A%2F%2Fplay.google.com%2Fstore%2Fapps%2Fdetails%3Fid%3Dcom.tribab.tricount.android&_branch_match_id=517666614845263475, googlePlay; Aufgerufen am: 2018-04-27.

[10] "Trip Splitter," http://appcrawlr.com/ios/trip-splitter#authors-description, appCrawlr; Aufgerufen am: 2018-04-27.

[11] "Klipper," https://www.klippa.com/en/homepage/, klipper; Aufgerufen am: 2018-04-18.

[12] "Shoeboxed," https://www.shoeboxed.com/features/, shoeboxed; Aufgerufen am: 2018-04-18.

[13] “Growth from Knowledge,” http://www.gfk.com/de/success-stories/success-story/kaufverhalten-von-urlaubern-messen-bei-der-planung-ihrer-naechsten-reise/, gfK; Aufgerufen am: 2018-04-19.

[14] “Amazon Reviews,” https://www.amazon.com/Neat-Company-NeatDesk-Desktop-2005410/product-reviews/B01A0FQ8Q4, amazon; Aufgerufen am: 2018-04-18.

[15] “The Neat Company,” https://www.neat.com/how-neat-works/, neat; Aufgerufen am: 2018-04-18.

[16] “Gartner: Android steigert weltweiten Marktanteil auf 87,7 Prozent,” https://www.zdnet.de/88308701/gartner-android-steigert-weltweiten-marktanteil-auf-877-prozent/, zDNet; Aufgerufen am: 2018-04-19.

[17] “Dokumentation der Paypal Developers API,” https://developer.paypal.com/reference/, paypal; Aufgerufen am: 2018-05-13.

[18] “Dokumentation der currencylayer API von apilayer,” https://currencylayer.com/documentation, currencylayer, Aufgerufen am; 2018-05-13.