

Technology Arts Sciences TH Köln

Entwicklungsprojekt interaktiver Systeme

WS 2018/19

Dozenten

Prof. Dr. Gerhard Hartmann
Prof. Dr. Kristian Fischer

Mentoren

Ngoc-Anh Gabriel
Lena Wirtz

Teilnehmer

Tristan Heuser
Tobias Rutsch

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Methodischer Rahmen	4
The Usability Engineering Lifecycle	4
User Profiles	5
Contextual Task Analysis	6
Work Environment Analysis	8
Task Scenarios	9
Task: Fahrt suchen	9
Task: Hilfe suchen	10
List of User Tasks	11
User Task Organization Model	11
Platform Capabilities and Constraints	12
General Design Principles	13
Usability Goals	14
Qualitative Goals	14
Quantitative Goals	16
Work Re-Engineering	17
User Task Organization Model Re-Engineering	18
Task Scenarios Re-Engineering	19
Use-Cases	20
Conceptual Model Design	20
Produkt- oder Prozessorientiert	20
Dateneingabe	21
Ergebnisanzeige	21
Detailansicht Helfer	23
Navigation	23
Screen Design Standards	24
Detailed UI Interface	25
Navigation	26
Dateneingabe	26
Ergebnisanzeige	27
Detailansicht Helfer	28
Profil	29
Dashboard	29

Evaluation	30
Fazit	32
WBA Abschnitt	32
Architekturmodell	32
Externe Web-Services	33
Proof of Concept	34
Datenstrukturen	35
Eintrag	35
User	38
Messages	39
Anwendungslogik	39
Packages/Libraries	41
REST Modellierung	41
Topic Modellierung	44
Quellenverzeichnis:	45
Anhang:	45
User Profiles	45
Use-Cases	50
UI-Prototyp	54
Evaluation Domänenrecherche	54

Methodischer Rahmen

Das System soll nach dem Vorgehensmodell "The Usability Engineering Lifecycle" von Deborah J. Mayhew entwickelt werden. Nun wird das Vorgehensmodell genauer erläutert und eventuelle Anpassungen vorgenommen.

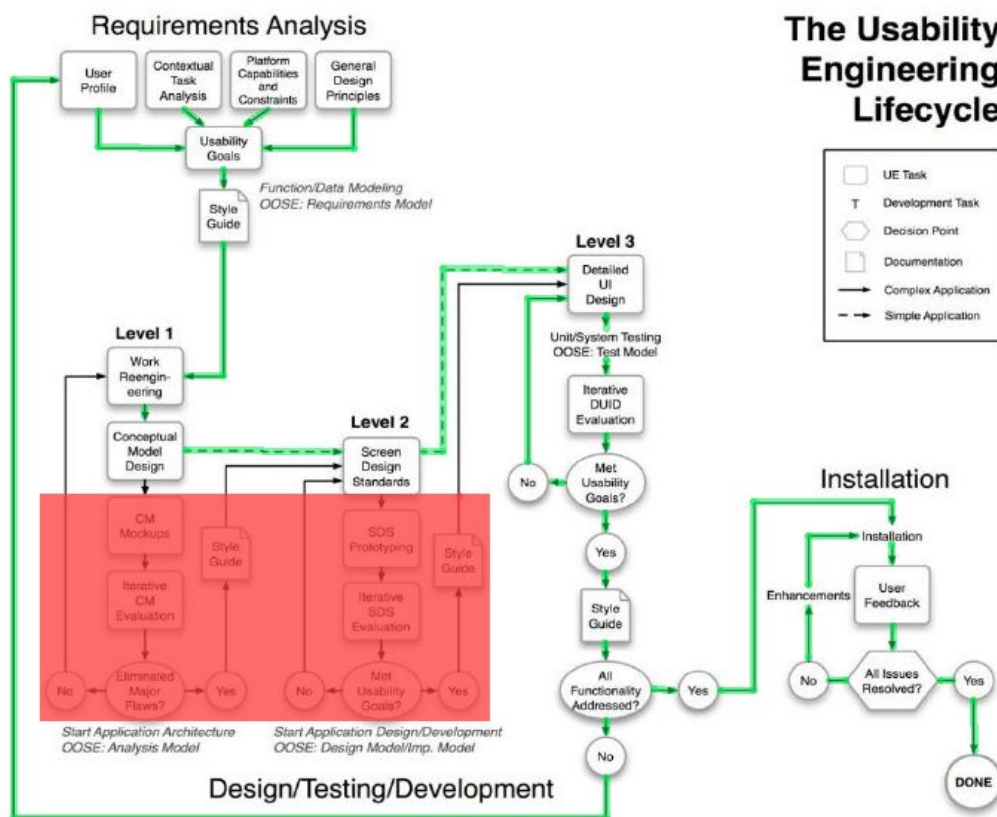
The Usability Engineering Lifecycle

Das Vorgehensmodell "The Usability Engineering Lifecycle" ist eine sehr gute Grundlage für die Entwicklung eines Systems. Es bietet viele Informationen zu den einzelnen Arbeitsschritten, die mittels Beispielen und Schablonen für die verschiedenen Artefakte eine sehr gute Stütze darstellen. Es bietet eine gute Übersicht, welche Artefakte erforderlich sind und wie diese aufeinander aufbauen. Es gliedert die Entwicklung in einzelne Phasen, welche ein strukturiertes und zielgerichtetes Arbeiten ermöglichen. Durch Angaben zu Aufwandsschätzungen und möglichen Abkürzungen ist das Projekt gut skalierbar und lässt sich an die Gegebenheiten wie Projektzeit und Größe des Teams skalieren.

Da das vollständige Erarbeiten aller Schritte der verschiedenen Phasen zu viel Zeit in Anspruch nehmen würde und dies vom Team in der kurzen Zeit nur sehr unwahrscheinlich bewerkstelligt werden kann, wird von den Reduzierungen, die im Usability Engineering Lifecycle angeboten werden, Gebrauch gemacht.

So wird z.B. auf die Evaluation und dem daraus resultierende Styleguide der einzelnen Phasen verzichtet, da diese einen zu hohen Arbeitsaufwand verursachen würden.

Stattdessen soll abschließend eine gemeinsame Evaluation für alle drei Phasen durchgeführt werden. Insgesamt soll das Projekt nach dem Ablauf des Usability Engineering Lifecycle für kleine Projekte durchgeführt werden. Die Grafik zeigt das Vorgehensmodell für Projekte mit geringem Ausmaß. Die Arbeitsschritte, die rot markiert wurden, werden nicht benötigt. Die grün markierten Pfeile zeigen die zu bearbeitenden Artefakte.



Grafik #01: The Usability Engineering Lifecycle nach Deborah Mayhew für Projekte von geringem Ausmaß

User Profiles

Um das System genau den Bedürfnissen der Benutzer anpassen zu können, werden diese genauer betrachtet. Dafür werden verschiedene User Profiles aufgestellt, die einige relevante Eigenschaften, Kenntnisse und Fähigkeiten der Nutzer definieren, anhand der sich Anforderung an das System und besonders des User-Interface-Designs ableiten lassen. Folgende Benutzermerkmale sind relevant für den weiteren Entwicklungsprozess:

- **Demografie:** Alter und Geschlecht der Nutzer haben Auswirkungen auf Designentscheidungen im User Interface. Personen unterschiedlicher Altersgruppen weisen verschiedene Präferenzen im UI-Design auf. So werden für Kinder z.B. häufig bunte Farben und für Senioren größere Schriftgrößen verwendet. Um eine gute Gebrauchstauglichkeit erreichen zu können, werden diese Faktoren berücksichtigt.
- **Sozioökonomischer Status:** Bildungsgrad, Beruf und finanzielle Möglichkeiten der Nutzer sind häufig Ursache einiger Transportschwierigkeiten, die es zu lösen gilt, weshalb auch darauf ein besonderes Augenmerk gelegt werden sollte.
- **Motivation:** Der Grund warum die Nutzer das System nutzen wird berücksichtigt, da die Benutzer ein bestimmtes Ziel erreichen wollen und dieses soll möglichst schnell und effektiv erreicht werden. So sollen mögliche Funktionen, die nicht zur Zielerreichung beitragen vermieden oder in den Hintergrund gerückt werden.

- **Technologie:** Die Technologie, mit der sich die Nutzer auskennen und welche häufig von ihnen genutzt wird sollte beachtet werden. So können bekannte Abläufe und Strukturen aufgegriffen werden, um vor allem die erste Nutzung zu vereinfachen und für den Nutzer angenehm zu gestalten, da er nicht mit zu vielen neuen Abläufen konfrontiert wird.
- **Technische Erfahrung:** Die Erfahrung im Umgang mit technischen Systemen der Benutzer spielt eine Rolle im Entwicklungsprozess. Aufgrund der Erfahrungswerte muss ein Design entworfen werden, dass den Nutzer weder über- noch unterfordert.
- **Fachkenntnis:** Kenntnisse in den Bereichen Fahrzeug führen, Transporte, Ladung, Montage und Installation sind ausschlaggebend für die Transportschwierig- und Transportleichtigkeiten der Nutzer.
- **Aufgabe:** Die Aufgabe/n die der Benutzer im System erfüllen soll. Darauf soll im Entwicklungsprozess besonders geachtet werden, um unnötige Arbeitsschritte zu vermeiden und dem Nutzer nur mit Aufgaben zu konfrontieren, die für ihn von Bedeutung sind oder seinen Input benötigen.

Eine vollständige Liste der User-Profiles befindet sich im **Anhang**.

Es lassen sich noch weitere User-Profiles finden, die relevante Nutzer für das System repräsentieren. Allerdings stellen die definierten User-Profiles den Kern der Zielgruppe, weshalb diese im weiteren Projektverlauf im Fokus stehen sollen.

Contextual Task Analysis

Im der folgenden Contextual Task Analyse wird der deskriptive Zustand der ausgeführten Aufgaben in einem Modell nach Deborah Mayhews "The Usability Engineering Lifecycle" dargestellt. Dieser Schritt wird ohne die Einbindung von Benutzern durchgeführt, da dies durch Zeitmangel nicht realisiert werden konnte. Stattdessen ist viel aus eigener Erfahrung und Vorwissen aus der Domänenrecherche eingeflossen.

Actor	Trigger	Use Case (Task)	Task Scenario Sequence	Errors, Problems, Comments
Helfer	Muss Sperrgut transportieren	sucht nach Optionen	1: Sucht im Internet nach Liefermöglichkeiten 2. Vergleicht Lieferoptionen	Es braucht viel Zeit einen Anbieter zu finden, wenn er nicht von einem Online-Shop angeboten wird.
	Entscheidung Lieferung oder Abholung	findet Optionen	3: Vergleicht Lieferungsoption mit Abholungsoption 4. Trifft eine	

			Entscheidung	
	VARIANTE 1: Lieferung	wählt Lieferung	5. wählt Lieferunternehmen aus (optional) 6. wählt Lieferungszeitpunkt aus 7. Lieferunternehmen liefert Sperrgut aus 8. Sperrgut wird angenommen	Lieferungen mit Sperrgut von Speditionsunterneh- men können sehr teuer sein. Es kann zu erheblichen Verzögerung kommen, wenn die Spediteure ausgelastet sind.
	VARIANTE 3: Abholung mit gemieteten Transporter		5. fährt/geht zu Transporterverleih 6. leiht sich Transporter von Verleihfirma oder Privatperson	Ohne Planung kann man sich nicht darauf verlassen, dass der Verleih einen passenden Transporter zu Verfügung stellen kann.
	Sperrgut ist in der Wohnung		x. Sperrgut wird aufgebaut x. elektronische Geräte werden installiert.	Die Installation von elektronischen Geräten kann teuer werden.
Fahrer	Muss Sperrgut transportieren.	Abholung mit Transport- er	1. fährt mit Transporter zum Abholungsort 2. (kauft und) lädt das Sperrgut in den Transporter 3. fährt Sperrgut zum gewünschten Ort 4. Sperrgut wird entladen 5. Sperrgut wird in die Wohnung transportiert	
	Muss Sperrgut aufbauen	sucht nach Optionen	6. sucht im Bekanntenkreis jemanden mit handwerklichen Fähigkeiten (optional) 7. sucht im Internet oder im Telefonbuch nach einem	

			Handwerker/Monteur/Installateur (optional) 8. vergleicht Optionen	
		Aufbau/Installation von einem Handwerker	9. beauftragt einen Handwerker/Monteur/Installateur 10. Handwerker kommt zu der abgemachten Zeit nach Hause 11. Handwerker hilft beim Aufbauen des Sperrguts 12. Handwerker hilft beim installieren des Sperrguts (optional) 13. Handwerker stellt seine Rechnung. 14. Handwerker wird bezahlt.	Handwerker/Techniker/Monteur können oft ausgebucht sein und es könnte verzögern bis man einen qualifizierten Handwerker/Techniker/Monteur zur Verfügung stehen hat. Handwerker wollten bezahlt werden und es entstehen Kosten.
		Hilfe bei dem Aufbau/Installation des Sperrguts von einem Bekannten	15. kontaktiert den Bekannten 16. Bekannter kommt zur Wohnung des Fahrers 17. Bekannter hilft beim Aufbau und der Installation	

Tabelle #01: Contextual Task Analysis

Work Environment Analysis

Das Arbeitsumfeld des Nutzers ist meist bei sich zu Hause, da meist dort die Abholung eines Sperrguts organisiert wird. Ein weiteres Umfeld könnte auch das Büro darstellen oder auch vielleicht die öffentlichen Verkehrsmittel. Zu Hause und im Büro steht den Benutzern meist Computer und Smartphone zu Verfügung, welche dort genutzt werden können. Im Büro und in Bus und Bahn ist man möglicherweise Lautstärke ausgesetzt und zu Hause kann man durch Mitbewohner oder den Partner kurz gestört werden. In Bus und Bahn könnte es sein, dass die Internetverbindung des Smartphones abbrechen könnte und dadurch könnten Prozesse, welche eine Internetverbindung benötigen, gestört werden.

Task Scenarios

Task: Fahrt suchen

User: Benutzer

Beschreibung:

1. Der Benutzer bestellt im Internet ein sperriges Produkt oder muss Sperrgut von einem Platz zu einem anderen Platz transportieren
2. Der Benutzer recherchiert im Internet nach Möglichkeiten zum Transport für Sperrgut oder zur Lieferung (optional)
3. In seinem Bekanntenkreis fragt der Nachbar nach Möglichkeiten zum Transport für Sperrgut (optional)
4. Die Optionen werden miteinander verglichen und eine Entscheidung wird getroffen.

Variante Lieferung

5. Der Benutzer wählt die Lieferung aus dem Onlineshop aus oder vereinbart eine Lieferung mit einem Speditionsunternehmen
6. Es wird einige Zeit gewartet.
7. Das Sperrgut wird von einem Speditionsunternehmen ausgeliefert und vom Benutzer ausgeladen und entgegengenommen
8. Das Sperrgut wird vom Speditionsunternehmen zur Etage des Benutzers hochgetragen. (optional)
9. Das Sperrgut wird, wenn es sich um ein Elektrogerät handelt von einem Techniker installiert und wenn es sich um ein Möbelstück handelt von einem Monteur aufgebaut. (optional)
10. Der Benutzer trägt das Sperrgut in seine Wohnung und installiert und montiert es, falls notwendig. (optional)

Variante Abholung mit gemieteten Transporter

5. Der Benutzer rechnet aus, wieviel Ladefläche er für das Sperrgut benötigt.
6. Der Benutzer recherchiert wieviel Ladefläche bestimmte Transporter haben.
7. Der Benutzer fährt oder geht zu einem Autoverleih bzw. Transporterverleih-Unternehmen und leiht sich dort einen Transporter.
8. Der Benutzer fährt bestenfalls an seinem Wunschtage zu seinem Abholort.
9. Der Benutzer packt das Sperrgut am Abholort in den Transporter
10. Der Benutzer fährt vom Abholort zu seinem Wunschziel/nach Hause
11. Dort trägt er (und/oder ein Bekannter) das Sperrgut in die Wohnung des Benutzers.
12. Benutzer fährt den Transporter wieder zurück zum Transporterverleih, um hohe Kosten zu vermeiden.
13. In der Wohnung wird das Sperrgut aufgebaut und montiert und bei Bedarf von einem Techniker oder dem Benutzer selbst installiert. (optional)

Task Closure: Das Szenario nimmt mindestens ein paar Stunden in Anspruch und mit optionalen Schritten könnte es mehrere Tage oder Wochen in Anspruch nehmen.

Um diese Aufgabe zu unterstützen, sollte die Benutzeroberfläche:

- Das finden von Menschen mit Transportern vereinfachen
- Die Abholung eines Transporters vermeiden
- Unnötige Kosten vermeiden

Task: Hilfe suchen

User: Fahrer

Beschreibung:

1. Der Benutzer bestellt im Internet ein sperriges Produkt oder muss Sperrgut von einem Platz zu einem anderen Platz transportieren
2. Der Benutzer fragt im Bekanntenkreis nach Leuten, die handwerklich begabt sind und dem Benutzer helfen, das Sperrgut ein- und auszuladen, zu montieren und optional zu installieren.
3. Der Benutzer recherchiert im Internet oder in den Gelben Seiten nach Handwerkern, Monteuren oder Technikern. (optional)
4. Der Benutzer macht einen Termin mit dem Handwerker/Monteur/Techniker aus (optional)
5. Der Benutzer organisiert die Fahrt mit den Helfern (optional)
6. Der Benutzer geht zu seinem Transporter und fährt an seinem Wunschtage zu seinem Abholort.
7. Der Benutzer (und/oder ein Helfer) packt das Sperrgut am Abholort in den Transporter
8. Der Benutzer fährt vom Abholort zu seinem Wunschziel/nach Hause
9. Dort trägt er (und/oder ein Helfer) das Sperrgut in die Wohnung des Benutzers.
10. In der Wohnung wird das Sperrgut von einem Monteur oder Handwerker aufgebaut
11. Bei Bedarf wird das Gut noch von einem Helfer oder einem Techniker/Handwerker installiert und betriebsfertig gemacht. (optional)

Task Closure: Das Szenario nimmt mindestens einen Tag, aber wahrscheinlich mehrere Tage/Wochen in Anspruch, da ein Handwerker engagiert werden könnte und aufgrund seiner Auslastung kann es zu Verzögerungen kommen.

Um diese Aufgabe zu unterstützen, sollte die Benutzeroberfläche:

- Das finden von Helfern vereinfachen
-

List of User Tasks

List of User Tasks - Helfer

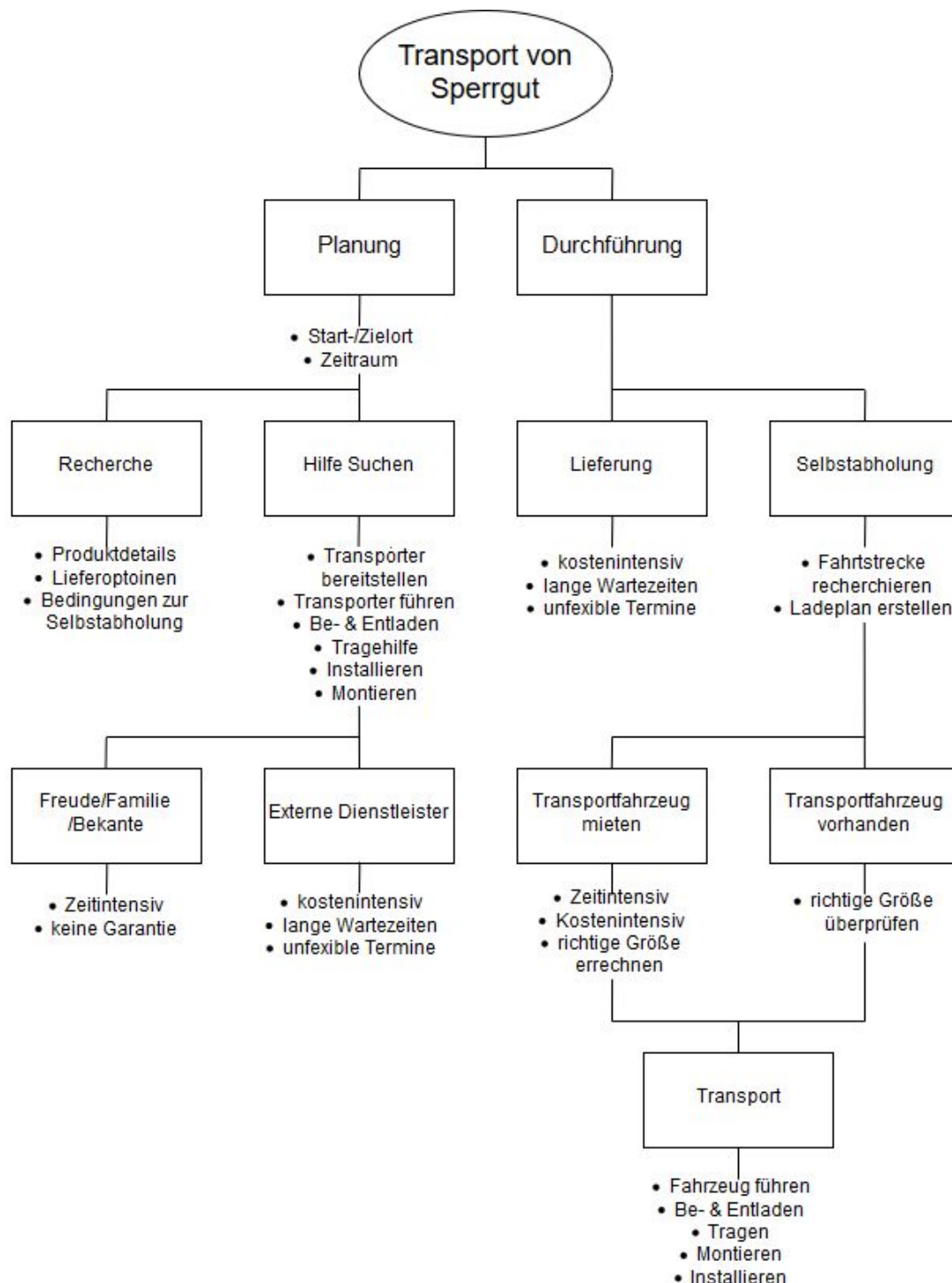
- Recherche nach Transportmöglichkeit
- Transporter organisieren
- Ladeplan erstellen
- Termin mit Transporterinhhaber/-unternehmen absprechen
- Transporter leihen
- Transporter be- und entladen
- Sperrgut tragen
- Montieren
- Installieren

List of User Tasks - Fahrer

- Transporter fahren
- Recherche nach Hilfe
- Hilfe bzw. Helfer organisieren
- Termin mit Helfer absprechen
- Ladeplan erstellen

User Task Organization Model

Anhand der Recherche lässt sich folgendes User Task Organization Model aufstellen. Dafür wurden besonders die Ergebnisse aus der User Task Analyse, der Domänerecherche und den User Profiles verwendet. So lässt sich auch optisch gut erkennen welche Aufgaben zum Prozess der Planung und Durchführung beim Transport von Sperrgut gehören. So lassen sich Schwachstellen und Verbesserungsmöglichkeiten identifizieren. Außerdem lässt es sich mit einem Aufgabenmodell mit Berücksichtigung des zu erstellenden Systems vergleichen, um zu erkennen welche Tasks vereinfacht oder vollständig eingespart werden können.



Grafik #02: User Task Organisation Model

Platform Capabilities and Constraints

Die Applikation wird auf der Android Plattform entwickelt, welche vor allem sich auf Smartphones spezialisiert. Deswegen ergeben sich ein paar Bedingungen. Es muss auf folgendes geachtet werden:

Eigenschaften	Möglich	Nicht möglich
Display Größe	4 Zoll +	Unter 4 Zoll
Bit-Mapped-Display	X	
Eingabegeräte	Virtuelle Tastatur, Touchscreen	
Multitasking	X	
Spezial Effekte	Animationen, 3D-Effekte	
Kommunikation/Sensoren	WLAN, Mobiles Internet, GPS	
OS	Android 6.0 (API 26) oder höher	Android Version unter 6.0
Energieversorgung	Akku, Netzbetrieb	

Tabelle #02: *Platform Capabilities and Constraints*

General Design Principles

Im folgendem Schritt werden nun Designprinzipien festgelegt, die während des Designprozesses Einfluss auf die Gestaltung nehmen. Vor Allem werden Prinzipien gewählt und dargestellt, welche für Android bzw. mobile Plattformen sinnvoll sind.

Material Design

Das Material Design der Plattform Google soll als Designansatz verwendet werden, da die Benutzer Vorerfahrungen im Design haben und sich nicht an neue Ansätze gewöhnen müssen. Der Benutzer kann dadurch das System möglichst erfolgreich und effizient nutzen. Das Material Design ähnelt zusätzlich dem Android Design und den Android bzw. Google eigenen Launchern. Es ist zudem in Android Studio integriert, was es einfacher macht umzusetzen. Seit einem Update kann es noch flexibler gestaltet nach den Bedürfnissen der Benutzer gestaltet werden.

Keep it brief

Dieses Prinzip besagt, dass Texte und Wörter möglichst kurz gefasst werden und das Interface sich dadurch nicht überlädt und diese gelesen.

Decide for me but let me have the final say

Da ein Packplan in der Anwendung erstellt wird und dieser auch grafisch dargestellt werden soll, soll der Benutzer die finale Entscheidung treffen, ob das Sperrgut nach seinem Wunsch in den Transporter passt und die Fahrt realisiert werden kann.

Only show what i need when i need it

Um den Benutzer nicht zu überlasten sollen Informationen und Optionen nicht in dem Moment zu sehen sein, wenn sie nicht benötigt werden.

Sprinkle encouragement

Zu jeder Aktion, die der Benutzer auf dem Smartphone ausgeführt hat, sollte es Feedback geben.

Make important things fast

Wichtige Aktionen, wie speichern eines Eintrags oder ein Chat sollte möglichst schnell und ohne große Ladezeit funktionieren

Never lose my stuff

Da im System Instate erstellt werden, welches Zeit in Anspruch nimmt, sollten diese nicht verloren gehen oder dem Benutzer nicht zu Verfügung stehen, da dies den Benutzer frustrieren könnte.

Usability Goals

Anhand der User Profiles, Contextual Task Analysis, Platform Capabilities & Constraints und General Design Principles lassen sich Usability Goals formulieren, die es zu erfüllen gilt, um eine gute Gebrauchstauglichkeit zu erreichen. Sie gelten als Richtlinien und Richtwerte, die es beim Entwickeln des User-Interface zu berücksichtigen gilt und helfen dabei, sich auf die wichtigsten Aspekte zu fokussieren.

Hauptsächlich werden projektspezifische Usability Goals festgelegt, die aus User Profiles, Task Analysis und Domänenrecherche resultieren. Da die Altersgruppe sehr weit gefächert ist, soll das Design schlicht gehalten werden. Außerdem soll berücksichtigt werden, dass ältere Nutzer eventuell besondere Bedürfnisse an das User-Interface aufweisen. Besonders soll darauf geachtet werden, dass trotz der großen Menge an Informationen alle relevanten Daten Übersichtlich und ohne großen Aufwand angezeigt werden.

Um die Usability Goals ihrem Gewicht nach angemessen behandeln zu können, werden diese priorisiert. Dafür werden drei Prioritätsstufen verwendet.

- **1** - Zwingend erforderlich vor der Veröffentlichung
- **2** - Sollte nach Möglichkeit erfüllt sein, wenn Zeit- oder Arbeitsaufwand nicht zu hoch ist
- **3** - Wünschenswert aber nicht erforderlich

Qualitative Goals

#01: Für die Vertrauenswürdigkeit relevante Daten sollen einstellbar und für andere Nutzer einsehbar sein.

Priorität: 1

Bei gemeinsamen Aktivitäten mit Fremden aus dem Internet, legen die meisten Nutzer einen hohen Wert auf die Vertrauenswürdigkeit der anderen Nutzer. Daher soll das System den Nutzern einige Möglichkeiten bereitstellen, um die Vertrauenswürdigkeit ihres Profils zu gewährleisten. So sollen neben den Bewertungen anderer Nutzer auch einige Profildaten wie z.B. Profilbild eingetragen werden können, um positiv und seriös auf andere Nutzer wirken zu können.

#02: Die Eingabe der Daten zum gewünschten Transport soll auch bei Unterbrechungen beständig sein.

Priorität: 1

Bei der Eingabe der Daten kann es schnell passieren, dass der Nutzer die Eingabe unterbricht, um z.B. die Maße der zu transportierenden Ware zu recherchieren oder um in den Terminkalender zu schauen, um den idealen Zeitraum angeben zu können. Daher soll auch bei Eingabeunterbrechungen dafür gesorgt werden, dass bereits eingegebene Daten nicht verloren gehen und der Nutzer schnell sieht wo er aufgehört hat.

#03: Es sollen nur relevante Transporte zum teilnehmen angezeigt werden.

Priorität: 1

Um den Planungsprozess so effizient wie möglich zu gestalten sollten Transporte die auf Grund von zu großen Entfernungen oder unpassenden Zeitraum irrelevant sind vom System selbständig aussortiert werden, damit der Nutzer nicht manuell nach diesen Kriterien aussortieren muss.

#04: Die Übersicht der Transporte soll gut strukturiert sein, damit sich der Nutzer gut orientieren kann.

Priorität: 1

Um den Planungsprozess möglichst effizient zu gestalten soll der Nutzer auf dem ersten Blick gleich mehrere mögliche Transporte samt wichtigsten Daten sehen können, damit er gleich nach persönlichen Kriterien entscheiden kann welche potentiellen Helfer für ihn am meisten in Frage kommen, ohne viel Zeit in die Entscheidung investieren zu müssen.

#05: Die relevanten Eigenschaften wie Start-/Zielort, Zeitraum sowie Transportleichtigkeiten und Transportschwierigkeiten sollen in der Übersicht sichtbar sein.

Priorität: 2

Ist ein geeigneter Helfer gefunden soll dieser ohne viele Zwischenschritte kontaktiert werden können, um diese direkt zu kontaktieren und den Fokus nicht zu verlieren, damit die Planung des Transports frühzeitig gestartet werden kann und alle beteiligten die Chance haben sich nach anderen Helfern umzusehen, falls der Transport doch nicht zustande kommt.

#06: Persistente Daten sollen bei erneuten Einträgen gespeichert werden, sodass diese nur bestätigt werden müssen.

Priorität: 2

Da sich einige Daten, wie z.B. Transportschwierigkeiten und Transportleichtigkeiten eher selten ändern sollen diese für eventuelle weitere Inserate gespeichert werden, sodass diese nicht erneut eingegeben werden müssen, sondern lediglich Kleinigkeiten bei Bedarf geändert werden können.

#07: Auch ältere Menschen mit Schwierigkeiten kleine Texte zu lesen sollen das System nutzen können

Priorität: 2

Da ältere Personen häufig Schwierigkeiten haben zu kleine Symbole oder Schriftgrößen richtig zu erkennen, sollen diese ausreichend groß sein. Gegebenenfalls kann dies auch mittels einer Funktion zum Vergrößern der besagten Komponenten gelöst werden.

#08: Die vorgeschlagene Fahrtstrecke soll für den Nutzern leicht verständlich übermittelt werden.

Priorität: 3

Um den Planungsprozess der Fahrtstrecke zu vereinfachen soll das System eine geeignete Strecke vorschlagen. Diese soll den Nutzern mit Hilfe einer Karte oder Ähnlichem schnell und einfach zugänglich gemacht werden.

#09: Der vorgeschlagene Ladeplan soll für den Nutzern leicht verständlich übermittelt werden.

Priorität: 3

Um den Nutzer beim Beladen des Fahrzeugs zu unterstützen soll bei Bedarf ein Ladeplan vorgeschlagen werden. Dieser soll den Nutzern mit Hilfe einer Grafik schnell und einfach zugänglich gemacht werden.

Quantitative Goals

#01: Die Daten zum Transport sollen innerhalb von 5 Minuten eingetragen werden können.

Priorität: 1

Damit der Nutzer bei der Planung des Transports Zeit einsparen kann und damit er die Motivation und den Fokus nicht verliert soll der Vorgang der Datenerfassung möglichst schnell ablaufen. Dafür wurde ein Zeitlimit von 5 Minuten festgelegt.

#02: Ein eingestelltes Inserat soll sofort aufgenommen werden und für andere Nutzer sichtbar sein.

Priorität: 1

Um keine Zeit zu verschwenden soll ein eingestelltes Inserat sofort aufgenommen und für andere Nutzer sichtbar gemacht werden. Dafür soll schon beim Erstellen auf Vollständigkeit und Gültigkeit der Daten geachtet werden, sodass eine anschließende Überprüfung nicht benötigt wird und das Inserat direkt zugänglich gemacht werden kann.

#03: Für ein Inserat soll innerhalb von drei Tagen, spätestens aber innerhalb von einer Woche ein passender Helfer gefunden werden.

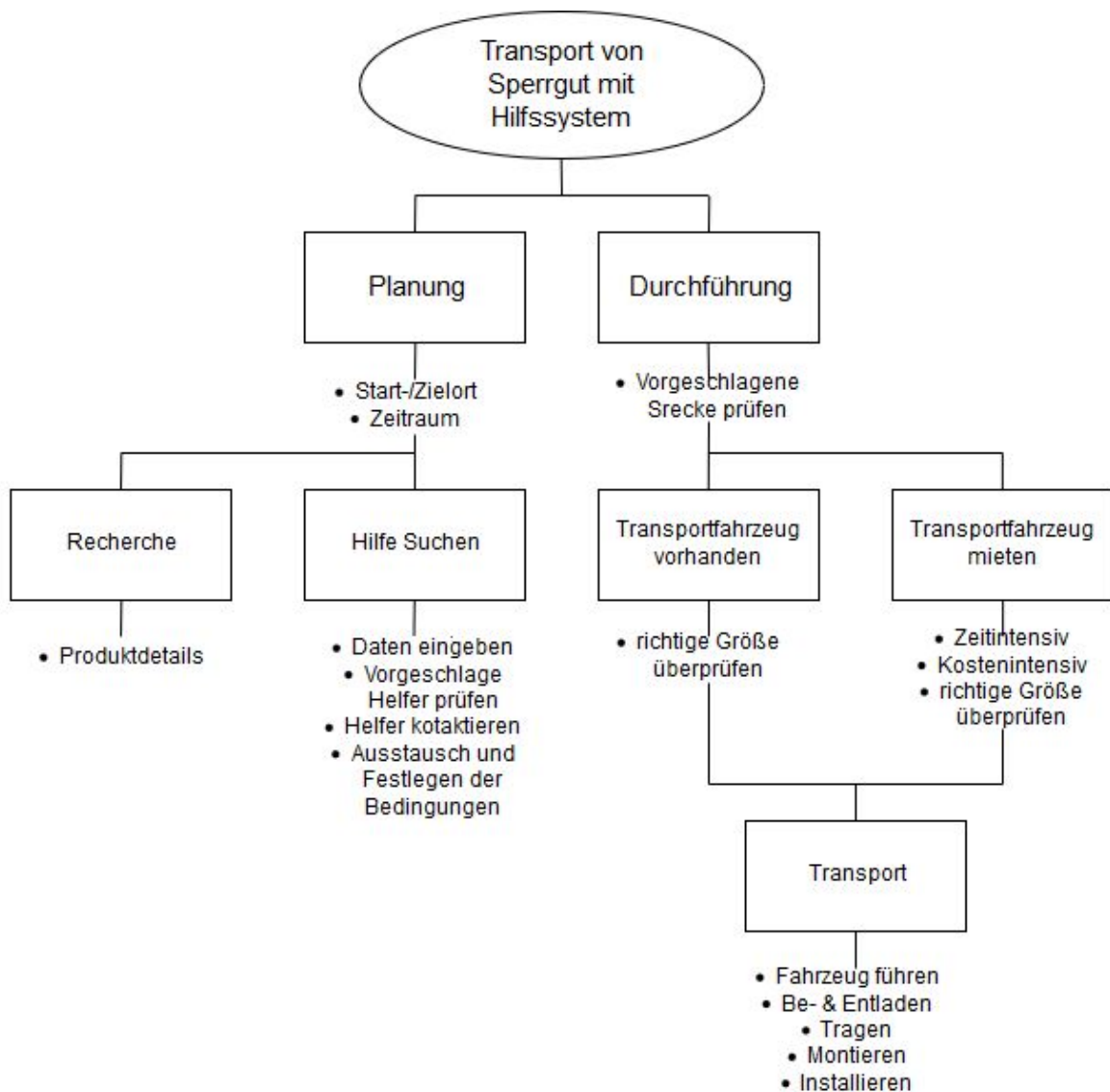
Priorität: 2

Damit der Nutzer das System für nützlich und vertrauenswürdig erachtet, soll für ein eingestelltes Inserat möglichst schnell ein passender Helfer gefunden werden. Eine schnelle positive Reaktion sorgt dafür, dass der Nutzer die Motivation nicht verliert und auch rechtzeitig vom System profitieren kann. Allerdings kann man auf diesen Punkt nur wenig Einfluss nehmen. Dennoch soll in der weiteren Entwicklung darauf geachtet werden, dass dieser Prozess nicht negativ beeinflusst wird.

Work Re-Engineering

Im nächsten Schritt geht es darum, an welchen Stellen der Planungsprozess für den Transport von Sperrgut mit Hilfe eines technischen Systems vereinfacht werden kann. Dabei soll der Nutzer vom System profitieren indem einzelne Aufgaben vom System übernommen oder vereinfacht werden. An manchen Stellen können Aufgaben auch so gelöst werden, dass weitere Schritte überflüssig werden.

User Task Organization Model Re-Engineering



Grafik #03: User Task Organization Model Re-Engineering

Nach dem Re-Engineering des User Task Organization Model fällt auf dem ersten Blick auf, dass die Menge der Aufgaben reduziert wurde. Das liegt daran, dass der besonders zeitaufwendige Prozess "Hilfe suchen" am meisten vom System unterstützt wird. So ist es nicht mehr nötig bei Familie, Freunden und Bekannten nach Hilfe zu suchen, was bisher der aufwendigste Arbeitsschritt war. So müssen die Transportbedingungen lediglich ein Mal ins System eingetragen werden und nicht mehr an viele Personen separat übermittelt werden. Das System unterstützt den Nutzer dabei anhand seiner Daten einen geeigneten Helfer zu finden. Dadurch muss eine viel geringe Anzahl potentielle Helfer (im optimalfall nur einer) kontaktiert werden. Außerdem bietet das System zusätzlich die Möglichkeit selbst von anderen Hilfesuchenden kontaktiert zu werden. Da eine viel größere Anzahl an potentiellen

Helfern überprüft und angefragt werden kann, ist die Wahrscheinlichkeit auf einen externen Dienstleister zurückgreifen zu müssen, deutlich geringer. Daher werden Aufgaben wie Recherchen und Kontaktaufnahme zu diesem Dienstleister deutlich seltener und in den meisten Fällen gar nicht gebraucht. Das gleiche gilt für die Aufgabe "Transportfahrzeug mieten". Aufgrund der großen Menge an potentiellen Helfern, wird diese Option auch deutlich seltener benötigt. Da, jedoch nicht immer garantiert werden kann, dass ein Helfer mit Transportfahrzeug gefunden wird, kann diese Aufgabe nicht vollständig gelöst werden. Allerdings muss bei einem gemeinsamer Transport nur ein Teilnehmer einen Transporter mieten und ist mit den damit verbunden Aufgaben konfrontiert. Zusätzlich können weitere Aufgaben erleichtert werden, indem das System eine Fahrtstrecke und einen Ladeplan vorschlägt, die nicht mehr vom Nutzer selbst erstellt werden müssen.

Task Scenarios Re-Engineering

Task: Fahrt suchen

User: Benutzer

Beschreibung:

1. Der Benutzer bestellt im Internet ein sperriges Produkt oder muss Sperrgut von einem Platz zu einem anderen Platz transportieren.
2. Die Produktdetails zu Verpackung, Volumen und Paketmaße werden recherchiert. (optional)
3. Das System wird gestartet und der Benutzer loggt sich ein. (und meldet sich ggf. vorher an)
4. Der Benutzer gibt seine Start- und Zielort, gewünschter Zeitraum, Transportschwierig- und Leichtigkeiten und Daten zum Sperrgut ein.
5. Das System vergleicht die Daten mit denen anderer Nutzer und gibt die Daten der Helfer aus, die sich am meisten eignen.
6. Die Optionen werden miteinander verglichen und eine Entscheidung wird getroffen.
 - a. Wurde ein geeigneter Helfer gefunden geht es weiter mit Schritt 7.
 - b. Wurde kein geeigneter Helfer gefunden werden die Daten den Benutzers gespeichert und bei der Suche anderer Nutzer berücksichtigt.
 - c. Wird der Benutzer von einem anderen Nutzer kontaktiert geht es weiter mit Schritt 7.
 - d. Der Benutzer prüft zu einem späteren Zeitpunkt, ob nun ein geeigneter Helfer gefunden werden kann.
 - e. Schritt 6. wird wiederholt.
7. Der Helfer wird kontaktiert und eventuelle Unklarheiten werden besprochen
8. Ein Termin wird festgelegt

9. Am vereinbarten Termin fahren Benutzer und Helfer die vom System vorgeschlagene Strecke zu allen Abholorten und laden das Sperrgut nach vom System vorgeschlagenen Ladeplan in das Transportfahrzeug.
10. Benutzer und Helfer fahren das Sperrgut nach vom System vorgeschlagener Strecke zu den Zielorten
 - a. Bei Bedarf wird das Sperrgut gemeinsam Ausgeladen.
 - b. Bei Bedarf wird das Sperrgut gemeinsam an den Zielort getragen.
 - c. Bei Bedarf werden Möbel gemeinsam montiert.
 - d. Bei Bedarf werden Geräte gemeinsam installiert.

Task Closure: Das Szenario nimmt mindestens ein paar Stunden in Anspruch und mit optionalen Schritten könnte es mehrere Tage oder Wochen in Anspruch nehmen. Allerdings fällt der Planungsprozess viel kürzer aus, da Helfer deutlich schneller gefunden werden können. Zusätzlich besteht die Möglichkeit Helfer zu finden ohne aktiv zu Suchen, indem der Helfer den Kontakt aufnimmt. Auch am Tag des Transports kann Zeit eingespart werden, da das System die Benutzer beim Beladen des Fahrzeugs und Recherchieren der zu fahrenden Strecke unterstützt.

Use-Cases

Es wurden einige Use-Cases formuliert, welche den Ablauf einiger Usertasks und die dazugehörigen Aufgaben des System festhalten. Um Zeit zu sparen wurden "Essential Use-Cases" und "Concrete Use-Cases" angefertigt und auf ausführlichere Darstellungsweisen wie Use-Cases nach Cockburn verzichtet.

Die erstellten Use-Cases sind im Anhang zu finden.

Conceptual Model Design

Nun soll anhand der bisherigen Ergebnisse festgelegt werden, welche Screens benötigt werden und welche Komponenten diese beinhalten sollen. Dabei werden erste Wireframes erstellt werden, welche zeigen wie ein simples System mit den notwendigen Komponenten aussehen könnte.

Produkt- oder Prozessorientiert

Da der Planungsprozess des Sperrguttransports unterstützt werden soll und keine eindeutigen Produkte behandelt werden, gilt das System als prozessorientiert. Die benötigten Prozesse wurden bereits während dem Work Re-Engineering identifiziert. Am relevantesten sind dabei das Eintragen der Daten, der Ergebnis-Bildschirm der Helfersuche und die Detailansicht der Helfer. Nun wird festgelegt wie diese Prozesse dargestellt werden sollen.

-schwierigkeiten sollen mittels

auswählen, ob für die identifizierten Leichtig- und Schwierigkeiten Hilfe benötigt wird oder angeboten werden kann. Auch die Maße des Laderaums und der einzelnen zu transportierenden Produkte sollen auf valide Maßeinheiten überprüft werden. Weitere Produkte können über einen Button hinzugefügt werden. Schließlich wird der Eintrag über einen weiteren Button angelegt und der Prozess abgeschlossen.

Ergebnisanzeige

Usability Goals: Qualitative: #01, #03, #04, #05, #07

In der Ergebnisanzeige ist es wichtig, trotz der großen Datenmenge einen guten Überblick über die verschiedenen potenziellen Helfer zu vermitteln. Hier soll der Benutzer einen ersten Überblick über die verschiedenen Helfer erhalten und diese vergleichen. Da neben den ausschlaggebenden Kriterien wie Zeit, Ort und Hilfe auch Angaben zur Person des Helfers als wichtig für die den Entscheidungsprozess identifiziert wurden, sollen auch diese

Dateneingabe

Usability Goals:

Qualitative: #02, #06, #07

Quantitative: #01, #02

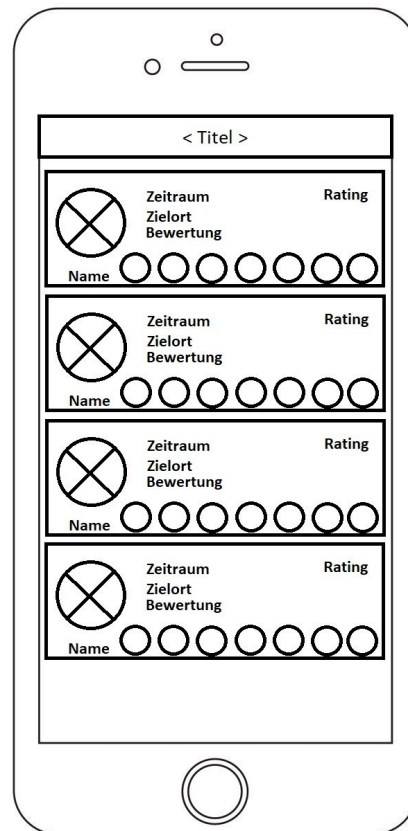
Die Dateneingabe soll gut verständlich und übersichtlich gestaltet werden. Um dies zu ermöglichen, sollen bekannte Strukturen verwendet werden. Zusätzlich sollen Auswahlfelder, die nur bestimmte Optionen zulassen, fehlerhafte Eingaben weitestgehend verhindern. So soll auch bei den Eingabefeldern für den Zeitraum lediglich valide Zeitangaben ausgewählt werden können. Beim übermitteln des Start- und Zielorts sollen die Benutzer die Möglichkeit haben die Adresse einzutragen, oder den gewünschten Ort mittels einer Karte auszuwählen. Die Transportleichtigkeiten und

Grafik #04: Wireframe Dateneingabe

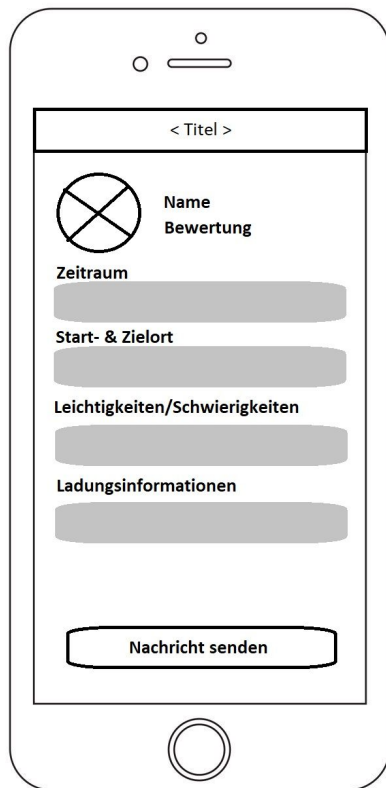
Auswahlflächen übermittelt werden. So kann der Nutzer

Informationen bereits in die Ergebnisanzeige einfließen. Dazu bieten sich Profilbild, Name und die durchschnittliche Bewertung des Helfers an. Das Profilbild ist im Verhältnis zum verbrauchten Platz die ausdrucksstärkste Information. Ein Gesicht zu einer Person zu kennen, ist ausschlaggebend für die Vertrauenswürdigkeit des Helfers und unterstützt den Nutzer dabei sich an diesen Helfer besser zu erinnern und ihn von anderen zu unterscheiden. Auch der Name spielt dafür eine Rolle. Die Bewertungen anderer Nutzer ist ebenfalls ein wichtiges Kriterium, um die Vertrauenswürdigkeit des Helfers zu steigern.

Um Platz zu sparen sollen die Leichtig- und Schwierigkeiten mittels kleiner Symbole angezeigt werden, welche je nach Bedarf und Angebot in unterschiedlichen Farben oder mit kleinen Änderungen angezeigt werden.



Grafik #05: Wireframe Ergebnisanzeige



Detailansicht Helfer

Usability Goals:

Qualitative: #01, #07, #08, #09

Hat der Benutzer einen Helfer ausgewählt, gelangt er zu einer Detailansicht für den Helfer. Diese Anzeige enthält die vollständigen Informationen, die der Benutzer benötigt. Neben den Daten, die auch in der Übersicht schon angezeigt wurden, bekommt der Benutzer hier noch weitere Informationen zu dem zu transportierenden Sperrgut des Helfers. Von hier aus soll der Benutzer ebenfalls Zugriff auf den Ladeplan und die Karte mit der vorgeschlagenen Route bekommen. Zusätzlich befindet sich auf diesem Screen ein Button, um den angezeigten Helfer zu kontaktieren.

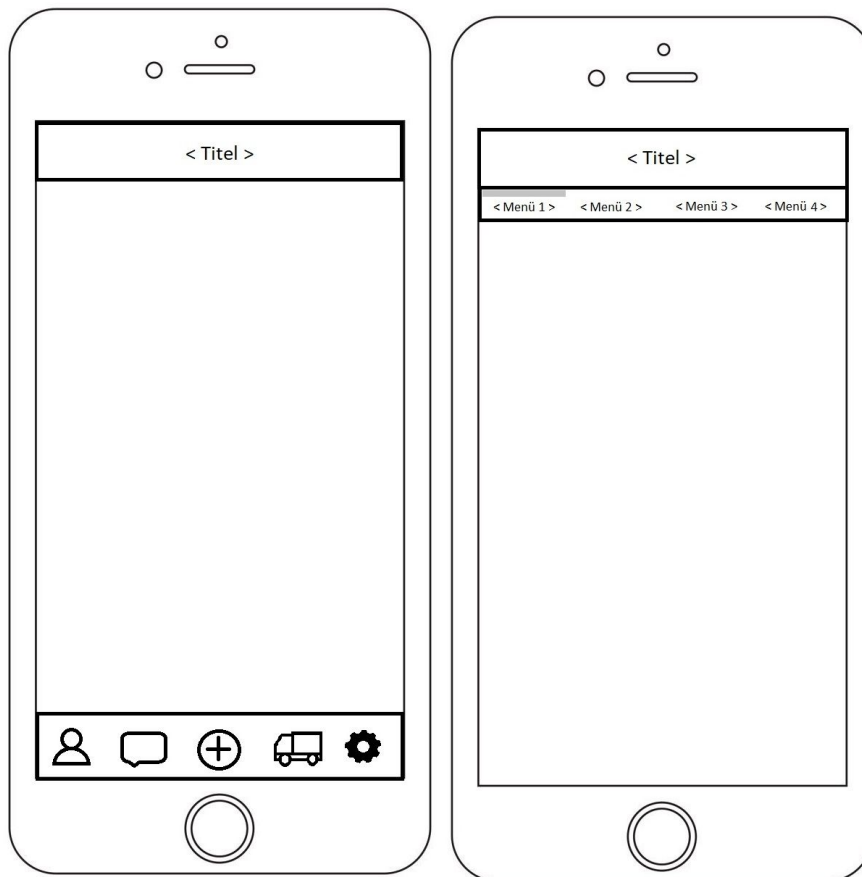
Grafik #06: Wireframe Detailansicht Helfer

Navigation

Für die Navigation, kommen verschiedene Methoden in Frage. Die gängigste Art der Navigation ist das "Hamburger Menü" dieses hat den besonderen Vorteil, dass es sehr platzsparend ist, da nur ein kleines Icon auf dem Bildschirm abgebildet wird, welches bei Interaktion das Navigationsmenü einblendet. Diese Methode bietet sich besonders bei Systemen mit vielen verschiedenen Menüpunkten an, da so ermöglicht wird diese platzsparend unterzubringen. Allerdings kann dies für Anfänger unübersichtlich wirken oder nur sehr schwer als Menü identifiziert werden, falls der Nutzer mit dem üblichen Icon nicht vertraut ist. Da im System nur wenige Menüpunkte benötigt werden, wird das "Hamburger Menü" nicht benötigt. Stattdessen kann man die Vorteile anderer Navigationsmethoden nutzen.

Für Projekte mit besonders wenig Menüpunkten eignet sich besonders die "Bottom Navigation". Diese ist auf allen Screens durchgehend eingeblendet und bietet den großen Vorteil, dass der Nutzer immer genau sehen kann in welchem Menüpunkt er sich gerade befindet und diesen mit nur einer Interaktion schnell wechseln kann. Allerdings ist diese Methode weniger platzsparend und kann zu Verwirrung führen, wenn die einzelnen Abschnitte nicht klar definiert sind und nicht eindeutig ist, welche Funktionen unter welchen Menüpunkten zu finden sind. Da das System über nicht allzu viele Menüpunkte verfügt und sich diese relativ gut gliedern lassen, soll diese Navigationsmethode verwendet werden.

Es gibt verschiedene Darstellungsweisen der “Bottom Navigation”. Neben der klassischen Methode mit einem Menübalken am unteren Bildschirmrand, gibt es die Darstellungsweise mittels Tabs. Dabei lassen sich die verschiedenen Menüpunkte durch swipen nach links und rechts durchwählen. Allerdings bietet die klassische Methode weitere Vorteile die für das System von Relevanz sind. Diese ist besonders für Anfänger leichter zu bedienen, da ein einfacher Klick auf das Symbol zum entsprechenden Menüpunkt führt. Die Navigation ist immer am unteren Bildschirmrand zu sehen und hebt sich klar vom Inhalt der Seite ab.



Grafik #07: Wireframes Bottom Navigation klassisch (links) und mit Tabs (rechts)

Screen Design Standards

Zuvor in den “General Design Principles” wurde sich auf das Material Design von Google festgelegt. Dieses bietet einige schon gegebene Standards, die hier nun näher festgelegt werden. Es wurde sich für folgende Farben und Schriften entschieden.

Schriften	Schriftart: Roboto Schriftfarbe: #FFFFFF
Farben	Primär: #4b58d8

	Abwandlungen: #002fa6 (dunklere Schattierung) #8485ff (hellere Schattierung), Sekundär: #ff5249
Hintergrund	Primär: #FFFFFF
Buttons	Die Buttons werden nach dem Material Design von Google designt und nutzen die Farben, die zuvor festgelegt wurden.
Control Standards	Hier wird ebenso das Material Design als Vorbild benutzt und es wird auf deren Standards gesetzt. Diese wären "Familiar", "Scannable" und "Effizient". Dies bedeutet, dass die Nutzer dank den Erfahrungen, den die Benutzer wahrscheinlich mit den Apps von Google und anderen, die App nach den Erwartungen benutzen können. Außerdem sollen ausgewählte Items auffälliger designt sein, als die nicht ausgewählten.
Dialog Box Standards	Auch hier werden die Standards von Googles Material Design verwendet. Hier werden beide Formen der Screen Dialog Designs, also im Vollbildschirm und die Dialogbox über dem Screen, verwendet.
Message Box Standards	Für die Message Boxes werden die in Android integrierten "Push-Notifications" verwendet.
Feedback Standards	Wie auch in den vorherigen Festlegung von Standard wird auch hier auf die Standarts von dem Material Design gesetzt.

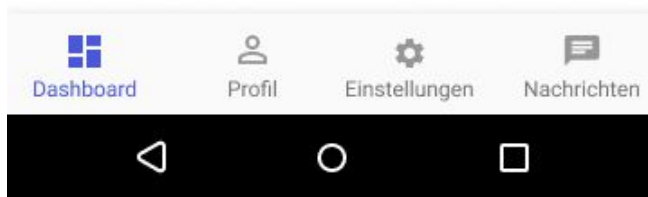
Tabelle #03: Screen Design Standards

Detailed UI Interface

Im nächsten Schritt werden einige Komponenten des Systems unter Berücksichtigung der festgelegten Design Prinzipien und Standards designt. Diese sollen das später implementierte Interface repräsentieren und als Vorlage für die Implementation dienen. Aus Zeitmangel konnten nicht alle Screens detailliert modelliert werden, weshalb hier nur die wichtigsten Komponenten aufgeführt werden können.

Navigation

Die "Bottom Navigation" beinhaltet vier Menüpunkte, welche jeweils mit einem kleinen Symbol angezeigt werden. Darunter ist der Titel des Menüpunkt ebenfalls in Textform angegeben, um Nutzern, denen die Symbole nicht bekannt sind bei der Identifizierung der Menüpunkte zu unterstützen. Zusätzlich wird der der aktuelle Menüpunkt farblich markiert, um bei der Orientierung im System zu helfen. In diesem Fall befindet man sich im Dashboard.



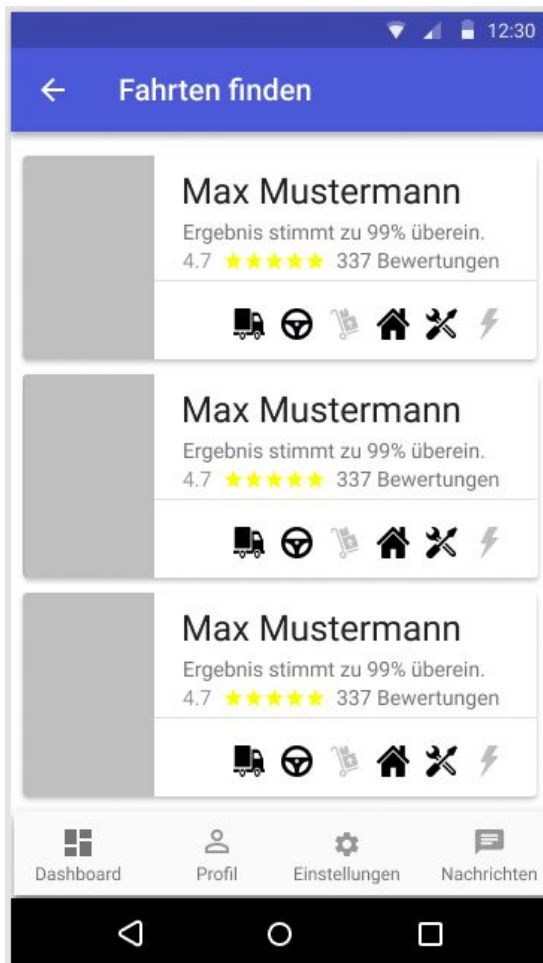
Grafik #08: Detailed UI Interface Navigation

Dateneingabe

Die Dateneingabe wurde nach den festgelegten Prinzipien visualisiert. So lassen sich die Zeitangaben aus bestimmten Werten auswählen und nicht frei eintragen. Auch die Transportleichtigkeiten und -schwierigkeiten werden mittels Auswahlfeld markieren. Um die Daten besser vergleichen zu können werden hier nur die in der Domänenrecherche herausgestellten Faktoren zum Auswählen visualisiert. Zu jedem Punkt gibt es eine kleine Erklärung, um unterschiedliche Definitionen zu vermeiden. Dabei werden Hilfe suchen und Hilfe anbieten nacheinander abgefragt, da so verhindert werden kann, dass der Nutzer durch den Klick auf ein falsches Feld unabsichtlich eine nicht zutreffende Information angibt. Die Abstände zwischen den Informationen sollen nicht zu klein geraten, damit der Nutzer seine Daten auch gewissenhaft und gründlich einträgt, da unkorrekte Informationen in den weiteren Schritten viele Probleme bereiten können.

Grafik #09: Detailed UI Interface Dateneingabe

Ergebnisanzeige





Die Ergebnisanzeige zeigt eine Liste mit potentiellen Helfern und den dazugehörigen Daten. Dafür wird jeder Helfer mit einer Card dargestellt. Diese beinhaltet neben Profilbild und Namen des Helfers das Ergebnis des Matchings und die durchschnittliche Bewertung. Diese Informationen dienen zur Unterstützung der Vertrauenswürdigkeit. Entgegengesetzt zur Planung im Conceptual Model Design werden Zeitraum und Strecke nun nicht in diesem Screen angezeigt, da entschieden wurde Helfer, die aus diesen Gründen als ungeeignet gelten schon im Matchingvorgang auszusortieren.

Die Transportleichtigkeiten des Helfers werden mittels kleinen Symbolen dargestellt. Die Transportschwierigkeiten des Helfers, sind für den Nutzer erstmal weniger relevant und werden in der Detailansicht sichtbar.

Grafik #10: Detailed UI Interface Ergebnisanzeige

Symbole Schwierig- und Leichtigkeiten

Um möglichst viel Platz zu sparen werden diese Faktoren mittels Symbolen dargestellt. Die Symbole haben je nach Zustand eine unterschiedliche Farbausprägung. Aktive Symbole werden kräftig in schwarz und inaktive Symbole in einem blassen grau dargestellt. Bietet der Helfer die Hilfe, die das Symbol abbildet, an wird dieses aktiv, andernfalls passiv angezeigt. Folgende Symbole wurden verwendet:

Symbol	Bezeichnung	Erklärung
	Transportfahrzeug bereitstellen	Hat Zugriff auf ein Transportfahrzeug
	Fahrzeug führen	Kann/darf ein Transportfahrzeug im Straßenverkehr führen





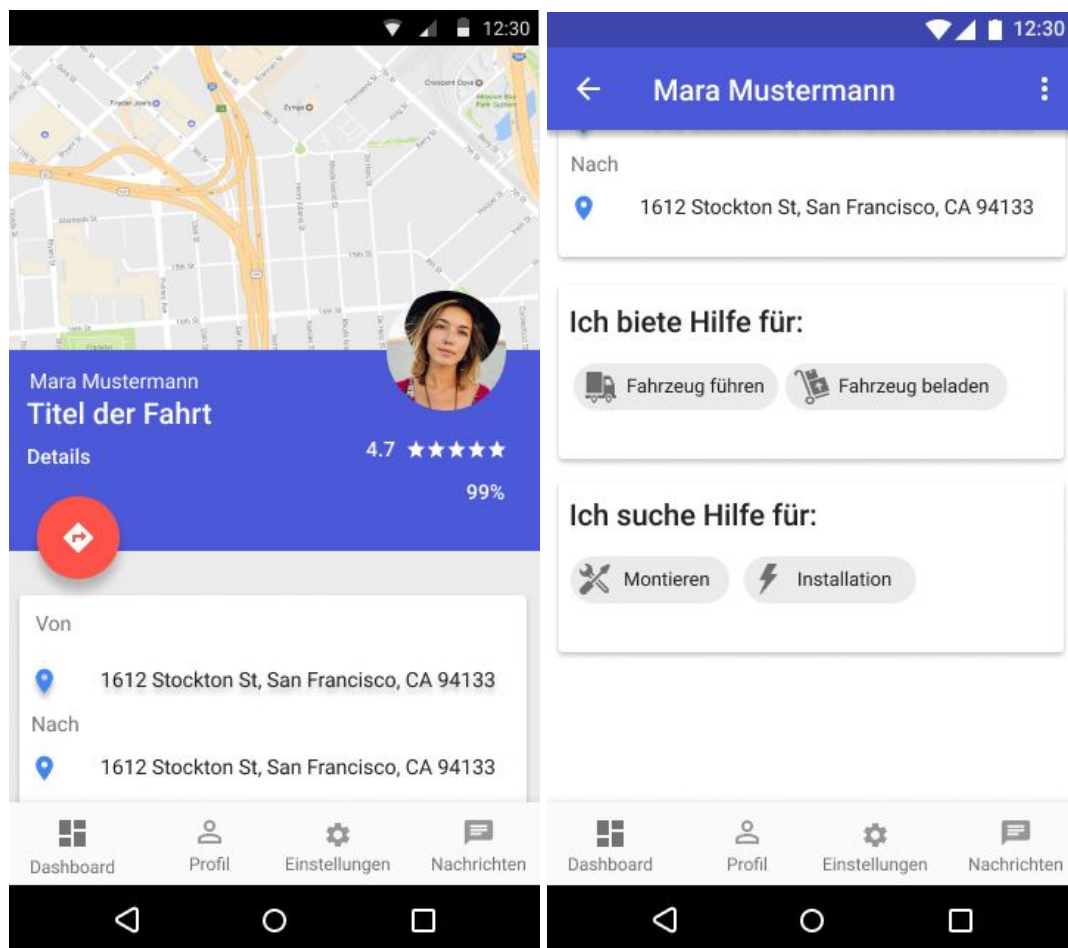
	Be- und Entladen des Transportfahrzeug	Ist in der Lage auch schwere Ladung in und aus dem Transportfahrzeug zu laden
	Tragehilfe bis an den gewünschten Platz	Ist in der Lage auch schwere Ladung (evtl. über Treppen) an den gewünschten Platz zu tragen
	Montage	Kann Möbel usw. montieren
	Installation von Elektrogeräten	Ist fähig und befugt Elektrogeräte zu installieren oder anzuschließen

Tabelle #04: Symbole Schwierig- und Leichtigkeiten

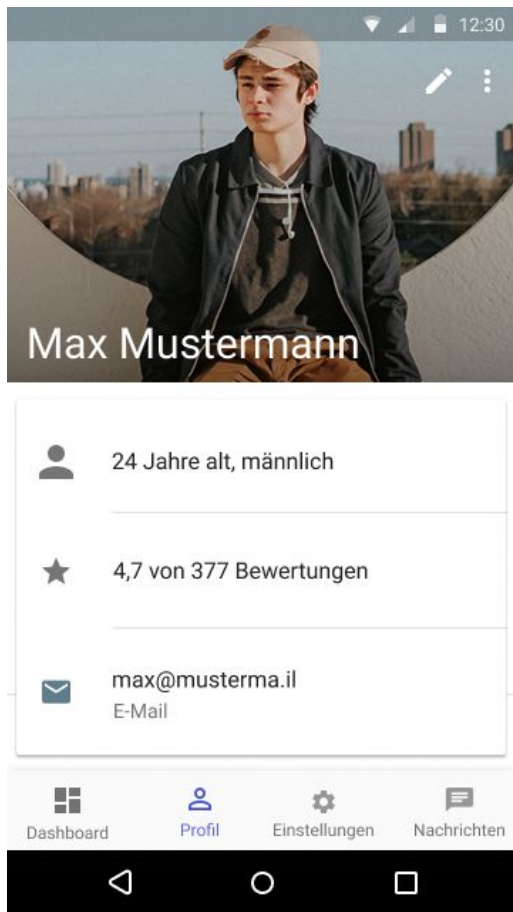
Detailansicht Helfer



Grafik #11: Detailed UI Interface Detailansicht Helfer

Die Detailansicht für die einzelnen Helfer wurde mittels Cards gegliedert. Für jedes wichtige Informationsfeld wurde eine Card angelegt. So sind die Informationen geordnet und übersichtlich dargestellt, sodass der Nutzer sich gut zurechtfinden kann. Aus Zeitmangel konnten nicht alle Informationen visualisiert werden, diese sollen jedoch nach demselben Schema dargestellt werden.

Profil

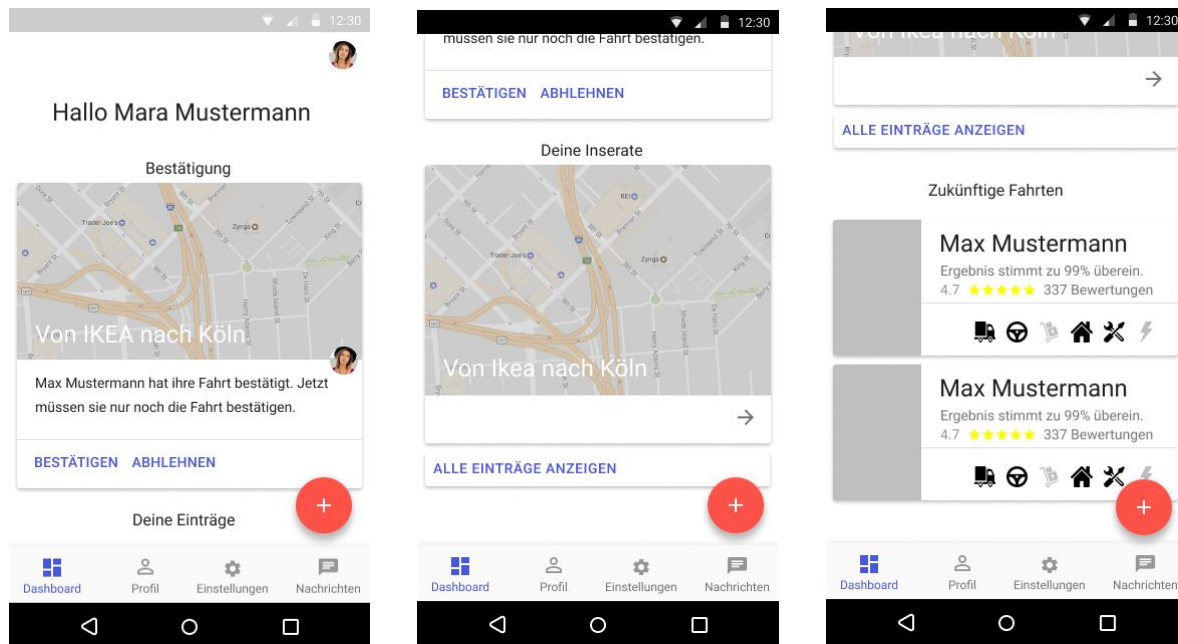


Das Profil soll einen persönlichen Einblick auf einen Benutzer verschaffen. Es wird ein großes Profilbild angezeigt, sowie eine „Card“ mit Informationen des Benutzers. Hier werden Daten wie das Alter des Benutzers, das Geschlecht oder auch der Wohnort. Auch bietet der Screen die Möglichkeit den Benutzer des Profils zu kontaktieren, da Kontaktangaben angezeigt werden. Außerdem befindet sich in einem Dropdown Menü die Möglichkeit den Benutzer eine Nachricht über die App zu schreiben. Die Bewertung des Benutzers steht in einer Zeile nochmal extra zu den allgemeinen Informationen. Dies soll Vertrauen schaffen und die Benutzer motivieren gute Bewertungen zu erhalten.

Grafik #12: Detailed UI Interface Detailansicht Profil

Dashboard

Das Dashboard soll dem Benutzer einen persönlichen Überblick bieten. Der Benutzer kann durch das Dashboard durchscrollen und findet dort einige personalisierte Informationen. Dort sollen Anfragen zu gemeinsamen Fahrten angezeigt werden, sowie die bereits bestätigten Fahrten und die eigen eingestellten Inserate. Hier soll der Benutzer einen schnellen Einblick erhalten können und immer auf den aktuellen Stand gebracht werden. Wichtig ist hier, dass man aus dem Screen die Fahrten bestätigen kann. So wird dies vor allem vorrangig angezeigt, so dass der Benutzer dies beim Start der mobilen Anwendung sieht und schnellstmöglich reagieren kann, um für einen reibungslosen Ablauf zu sorgen. Außerdem hat der Benutzer hier die Möglichkeit einen neuen Eintrag zu erstellen, wenn er den rot dargestellten „Add“-Button betätigt. Dies soll für einen schnellen „Flow“ sorgen.



Grafik #13: Detailed UI Interface Detailansicht Dashboard

Evaluation

Abschließend soll geprüft werden, ob die aufgestellten Usability Goals erfüllt wurden. Dieser Schritt ist besonders wichtig, da die Evaluation bisher eher vernachlässigt und nur selten dokumentiert wurde. Allerdings sind die Bedingungen für die Evaluation nicht optimal, da nicht alle Prototypen designt werden konnten. Das führt dazu, dass einige Komponenten nicht vollständig geprüft werden können. Dazu kommt, dass die Entwickler zu nah am Projekt sind, was häufig dafür sorgt, dass einige Fehler nicht auffallen, weil man sie auf dauer einfach übersieht. Außerdem ist es schwierig in so einer kurzen Zeit alle Seiten der Domäne vollständig zu verstehen und zu berücksichtigen. Für eine lückenlose Evaluation wäre ein vollständiger Prototyp der gesamten Anwendung nötig. Dieser muss von einer breit gefächerten Gruppe der möglichen Nutzer bewertet werden. Unter den aktuellen Umständen ist es nicht möglich alle Fehler oder eventuellen Probleme zu identifizieren.

Trotzdem soll eine erste Evaluation der Usability Goals nach der "Cognitive Walkthrough"-Technik von beiden Entwicklern durchgeführt werden, um zumindest einen Teil der Probleme zu identifizieren und passende Lösungsansätze vorzubereiten.

Qualitative Goals

#01: Für die Vertrauenswürdigkeit relevante Daten sollen einstellbar und für andere Nutzer einsehbar sein.

- Erforderliche Daten sind vorhanden und einsehbar
- Weitere persönliche Daten könnten eingefügt werden

#02: Die Eingabe der Daten zum gewünschten Transport soll auch bei Unterbrechungen beständig sein.

- Erst nach der Implementation überprüfbar

#03: Es sollen nur relevante Transporte zum teilnehmen angezeigt werden.

- Erst nach der Implementation überprüfbar
- Helfer deren Zeitraum nicht passt oder deren Entfernung zu Groß sind werden gefiltert

#04: Die Übersicht der Transporte soll gut strukturiert sein, damit sich der Nutzer gut orientieren kann.

- Usability Goal unpräzise formuliert
- Mehrere Helfer werden mit wichtigsten Attributen angezeigt

#05: Die relevanten Eigenschaften wie Start-/Zielort, Zeitraum sowie Transportleichtigkeiten und Transportschwierigkeiten sollen in der Übersicht sichtbar sein.

- Es sind nicht alle geforderten Daten zu sehen, Start-/Zielort und Zeitraum nicht mehr erforderlich (siehe Qualitative Usability Goal #03)
- Usability Goal muss nach neuen Erkenntnissen angepasst werden

#06: Persistente Daten sollen bei erneuten Einträgen gespeichert werden, sodass diese nur bestätigt werden müssen.

- Erst nach der Implementation überprüfbar

#07: Auch ältere Menschen mit Schwierigkeiten kleine Texte zu lesen sollen das System nutzen können

- Wurde berücksichtigt
- Muss durch Zielgruppe überprüft werden

#08: Die vorgeschlagene Fahrtstrecke soll für den Nutzern leicht verständlich übermittelt werden.

- Mit aktuellem Stand nicht überprüfbar
- Eine Karte in der Detailansicht unterstützt den Prozess

#09: Der vorgeschlagene Ladeplan soll für den Nutzern leicht verständlich übermittelt werden.

- Mit aktuellem Stand nicht überprüfbar

Quantitative Goals

#01: Die Daten zum Transport sollen innerhalb von 5 Minuten eingetragen werden können.

- Muss durch Zielgruppe überprüft werden
- Kann erst nach der Implementierung vollständig geprüft werden

#02: Ein eingestelltes Inserat soll sofort aufgenommen werden und für andere Nutzer sichtbar sein.

- Erst nach Implementation überprüfbar

#03: Für ein Inserat soll innerhalb von drei Tagen, spätestens aber innerhalb von einer Woche ein passender Helfer gefunden werden.

- Nicht überprüfbar
- Sehr schwer beeinflussbar (ist dieses Usability Goal sinnvoll?)

Fazit

Insgesamt fällt das Ergebnis der Evaluation nur sehr minimal aus, da viele Punkte mit dem aktuellen Stand nur schwierig oder gar nicht überprüfbar waren. Es ist schon sehr sinnvoll zwischen den einzelnen Schritten immer wieder eine Evaluation der Teilabschnitte durchzuführen, da manche Probleme so erst sehr spät auffallen und eine späte Korrektur viele Änderungen nach sich zieht, sodass die Zeit, die man durch das Weglassen der Teilevaluation einspart nur selten die Zeit deckt, die Fehler zu beheben.

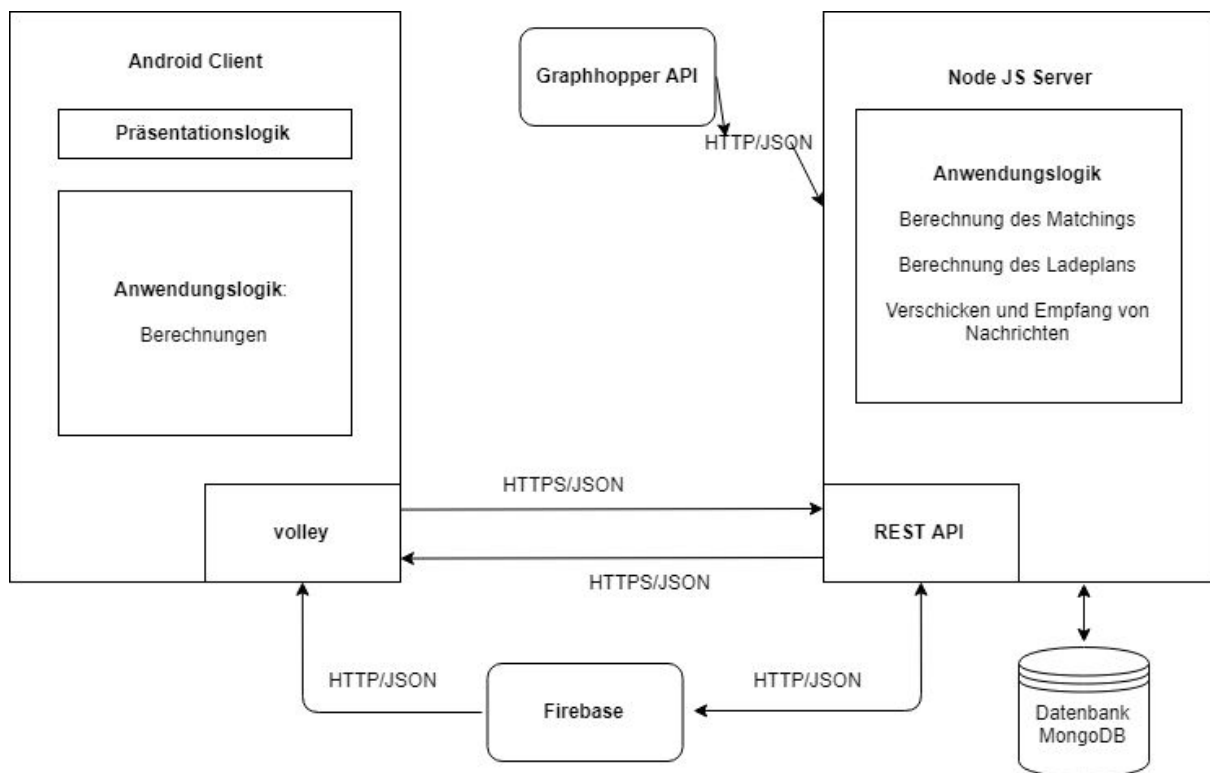
Aufgrund der Fehleinschätzung des Arbeitsumfang (siehe Projektplan) und den damit zusammenhängenden unvollständigen UI-Prototypen war die Evaluation nur sehr eingeschränkt möglich und sollte zu einem fortgeschritteneren Zeitpunkt wiederholt werden.

WBA Abschnitt

Architekturmodell

Das Architekturmodell wurde nochmals bearbeitet und verfügt nun über Informationen über die Verteilung der Anwendungslogik, sowie die Details über die Datenübertragung.

Außerdem wird die Ladeplan API nicht mehr berücksichtigt, da diese selbst umgesetzt wird und nun Teil vom NodeJS Server ist. Die Struktur wird allerdings beibehalten und wurde nicht erneuert. Die Google Maps API wurde zudem mit der Graphhopper API ausgetauscht, was im Unterkapitel "Externe Web Services" thematisiert wird.



Grafik #14: Architekturmodell

Externe Web-Services

Graphhopper

Für die Messung der Entfernung wurde die Graphhopper Directions API gewählt. Es gibt wenige Konkurrenten, die ihren Dienst kostenlos und ohne Eingabe von Zahlungsdaten anbieten. Der größte Konkurrent und Marktführer ist wohl die Google Maps API, die allerdings seit April 2018 eingegebene Zahlungsdaten voraussetzen und wenn die API häufig genutzt wird, automatisch Zahlungen abbuchen. Da eine komplett kostenlose Möglichkeit gesucht wird, scheidet die Maps API von Google aus. Eine weitere Alternative wäre OpenStreetMap, die allerdings nicht selbst eine Schnittstelle zur Routenberechnung anbieten. Es gibt allerdings einige Anbieter, die auf OpenStreetMap aufbauen und mit deren Kartendaten eine Schnittstelle zur Routenberechnung erstellt haben. Da aber auch hier die meisten das eingeben von Zahlungsinformationen voraussetzen, fällt diese Alternative raus. Die Graphhopper API hat sich im Rapid Prototyping bewährt und bietet auch an, dass man kostenlos eine Route mit mehreren Punkten berechnen kann. Die Abfragen werden über eine HTTPS Verbindung übermittelt, welches es relativ sicher macht, die Schnittstelle zu nutzen. Außerdem ist der Source Code öffentlich zugänglich, was es zu einem Open Source Projekt macht. Dies bietet vor allem auch gegenüber die Maps API von Google einen guten Datenschutz, da der Nutzer sehen kann, was genau mit seinen Daten passiert. Außerdem werden hier und wie in keiner anderen Routenplanung-Schnittstelle keine personenbezogenen Daten verarbeitet, was es aus sicherheits- und datenschutzbedenken sicher zu nutzen ist.

Proof of Concept

Die Proof of Concepts haben sich stark verändert und es wurden neue entworfen und formuliert. Der programmierte Prototyp wird beibehalten, aber der Proof of Concept an den Prototyp angepasst.

PoC #1 - Das Finden von potentiellen Helfern

Beschreibung	Es werden Helfer gesucht, die sich mit den eigenen Hilfsfähigkeiten möglichst gut ergänzen, sich örtlich in der Nähe befinden und die Fahrtroute möglichst identisch ist.
Exit	Anhand der Schwierigkeiten des Benutzers und der Fahrtroute wird ein weiterer Benutzer gefunden, der sich möglichst mit den Hilfsangeboten des Benutzers ergänzt und wird vom Benutzer akzeptiert.
Fail	Der Algorithmus schlägt fehl oder die Community ist zu klein.
Fallback	Der Benutzer muss selbst die Inserate durchgehen um zu sehen, welches am besten auf ihn zutrifft.

Tabelle #05: PoC #1 - Das Finden von potentiellen Helfern

PoC #2 - Das Erstellen eines Ladeplans

Beschreibung	Dem Benutzer wird ein Ladeplan erstellt, wo er gucken kann ob das Sperrgut in den gewünschten Transporter passt. So wird sichergestellt, dass die Fahrt zu Stande kommen kann und der Transporter nicht überfüllt ist. Berücksichtigt wird die Ladefläche des Transporters und die Größe des Sperrguts.
Exit	Der Ladeplan wird erstellt und es wird berechnet ob das Sperrgut in den gewünschten Transporter passt.
Fail	Es wird kein Projektplan erstellt.
Fallback	Der Benutzer muss selber kontrollieren und absprechen ob das Sperrgut in den Transporter passt.

Tabelle #06: PoC #2 - Das Erstellen eines Ladeplans

Datenstrukturen

Im folgenden Abschnitt wird auf die Datenstruktur eingegangen. Als Datenformat wurde sich für die Schemasprache JSON entschieden. Da sich entschieden wurde den Server in JavaScript umzusetzen bietet sich vor allem JSON an, da es eine hohe Kompatibilität zu JS bietet. Die Alternative wäre XML, die weniger kompatibel zu JavaScript bietet. XML hat allerdings einen größeren Overhead als JSON, der in diesem Projekt allerdings nicht benötigt wird und nur unnötig Ressourcen in Anspruch nehmen würde. Außerdem wird eine Mobile Anwendung entwickelt, wo auch mobile Daten möglichst wenig beansprucht werden sollten. Hierfür ist JSON idealer, da JSON weniger Daten benötigt, da der Overhead geringer ist. So können die Daten schneller und effizienter versendet werden. Außerdem bietet NodeJS eine natürliche Unterstützung zu JSON. YAML wäre eine weitere Alternative zu JSON. Allerdings können Daten im JSON Format schneller bearbeitet werden als im YAML Format, da es zwar weniger Funktionen beinhaltet, die wiederum aber nicht benötigt werden. Es sind folgende Daten-Schemata entstanden, welche nun beschrieben werden:

Eintrag:

```
{
  "id": Number,
  "userID": String,
  "date": String,
  "title": String,
  "route": {
    "start": {
      "lat": String,
      "lng": String
    },
    "destination": {
      "lat": String,
      "lng": String
    }
  },
  "period": {
    "start": String,
    "end": String
  },
  "needObstacles": {
    "haveTransporter": Boolean,
    "driveTransporter": Boolean,
    "canMontate": Boolean,
```

```

    "canInstall": Boolean,
    "canDischarge": Boolean,
    "canTransport": Boolean
  },
  "haveObstacles": {
    "haveTransporter": Boolean,
    "driveTransporter": Boolean,
    "canMontate": Boolean,
    "canInstall": Boolean,
    "canDischarge": Boolean,
    "canTransport": Boolean
  },
  "charge": {
    "package": Array,
    "weight": Array
    "size": {
      "height": Array,
      "length": Array,
      "width": Array
    },
  },
},
  "matchedPartner": String,
  "active": boolean,
  "succeed": boolean,
  "transporter": String
}

```

UserID

Die User ID speichert die ID des Benutzers, der den Eintrag angelegt hat und wird gespeichert, da man den Eintrag einem Benutzer zuweisen kann. In der User Tabelle werden weitere Informationen über den Nutzer gespeichert.

Date

Das Datum an dem, der Eintrag erstellt wird, wird gespeichert, um Einträge zu sortieren und um deren Aktualität zu prüfen.

Title

Die Benutzer können Titel erstellen um ihren Eintrag zu personalisieren oder detaillierter zu beschreiben.

Route

“Route” enthält die Route, die vom Benutzer gefahren wird um das Sperrgut abzuholen und zum Ziel zu bringen. In Form von dem Breitengrad (lat) und dem Längengrad (lng) wird ein bestimmter Punkt auf der Erde beschrieben. Der Startpunkt (start) gibt an wo der Benutzer das Sperrgut abholen möchte und der Zielpunkt (destination) wohin das Sperrgut transportiert werden soll.

Period

“Period” enthält die Zeit, in der es möglich ist die Fahrt durchzuführen. Hier gibt es den Startzeitpunkt (start) und den Endzeitpunkt (end), welcher in JavaScript als Time dargestellt und interpretiert wird. Dieses Attribut soll die Absprache zwischen den Benutzern verringern und eine Möglichkeiten bieten die Einträge filtern zu können um ein möglichst gutes Ergebnis zu bekommen.

Obstacles

Die Hilfestellungen werden in dem Attribut “Obstacles” gespeichert. Hier gibt es folgende Hilfestellungen, die angeboten werden können oder gesucht werden:

- “haveTransporter”: In diesem Attribut wird gespeichert, ob der Benutzer einen Transporter anbieten kann oder ob er einen sucht.
- “driveTransporter”: Hier wird gespeichert, ob der Benutzer einen Transporter fahren kann oder ob er einen Benutzer sucht, der einen Transporter fahren kann und darf.
- “canMontate”: In diesem Attribut wird gespeichert, ob der Benutzer sich handwerklich in der Lage führt Sachen zu montieren oder Möbelaufzubauen oder ob er jemanden dafür sucht, weil er sich nicht in der Lage fühlt
- “canInstall”: Hier wird gespeichert ob der Benutzer qualifiziert ist, bei der Installtion von elektronischen Geräten (wie z.B. Lampe oder Herd) zu helfen oder ob er einen Benutzer braucht, der eine solche Tätigkeit ausführen kann und darf.
- “canDischarge”: Hier wird angegeben ob der Benutzer beim Aus- und Entladen des Transporters Hilfe geben kann.
- “canTransport”: Hier wird gespeichert, ob der Benutzer das Sperrgut in die Wohnung eines anderen Benutzers tragen kann oder ob auch da Hilfe benötigt wird.

Der Benutzer kann im “needObstacles” Bereich die Hilfestellungen angeben, die er braucht und im Bereich “haveObstacles”, die Hilfestellungen bei denen der Benutzer anderen Benutzern geben kann.

Charge

Hier wird das Sperrgut aufgelistet, welches transportiert werden soll. Es wird mit einem Namen beschrieben, sowie Größe (size) und Gewicht (weight). Die Größe wird wiederum mit Höhe, Breite und Tiefe beschrieben.

MatchedPartner

Hier wird die id des Nutzers gespeichert, der dem Benutzer, der den Eintrag erstellt hat, eingewilligt hat zu helfen. Wenn hier ein Datensatz vorliegt, sollte der Eintrag inaktiv sein.

Active

Im Attribut "Active" wird gespeichert, ob der Eintrag noch aktiv oder inaktiv ist. Wenn dieser Eintrag inaktiv ist, sollte es nicht mehr möglich sein diesen Eintrag in der Applikation zu sehen. Außerdem sollte der Eintrag nicht mehr die Möglichkeit haben mit anderen Einträgen zu matchen.

Transporter

Hier wird das genaue Modell des Transporters angegeben, um die Ladefläche zu ermitteln, falls der Autor des Eintrags einen Transporter zur Verfügung stehen hat.

User:

```
{
  "userID": String,
  "username": String,
  "name": String,
  "surname": String,
  "residence": String,
  "email": String,
  "gender": String,
  "picture": String,
  "rating": {
    "userID": Array,
    "points": Array,
    "date": Array,
    "comment": Array
  }
}
```

UserID

Das Attribut "UserID" beinhaltet die eindeutige Identifikationsnummer eines Users. Diese wird benötigt, damit man den User eindeutig identifizieren kann.

Picture

Bei diesem Attribut wird die URL und der Name des Profilbildes angegeben. So ist jedem User ein Profilbild zugewiesen. Der Benutzer soll die Möglichkeit haben sein Profil persönlicher zu gestalten indem er ein Profilbild setzen.

Rating

In diesem Attribut wird die Bewertung des Nutzers gespeichert. Im Attribut "points" wird die einzelne Bewertung gespeichert, welche ein Nutzer (rating.userID) gegeben hat. Zusätzlich kann dieser einen Kommentar (rating.comment) abgeben.

Messages

```
{
  "messageID": String,
  "time": String,
  "fromUserID": String,
  "toUserID": String,
  "text": String,
  "attachment": {
    "type": String,
    "body": String
  }
}
```

Dieses Schema dient dazu, dass Nutzer Nachrichten austauschen können. Dadurch fällt es ihnen leichter Informationen auszutauschen. Jede Nachricht hat seine ID (messageID) und den Zeitpunkt, wann die Nachricht erstellt wurde. Der Nutzer, der die Nachricht geschrieben hat wird gespeichert (fromUserID) und auch der Nutzer, der die Nachricht empfangen soll. Der Benutzer hat die Möglichkeit einen Text zu schreiben (text) und einen Anhang hinzuzufügen (attachment) wie z.B. ein Bild oder einen Standort.

Anwendungslogik

Finden von potentiellen Helfern

Bei der Berechnung des Finden von potentiellen Helfern sind ein paar Kriterien zu berücksichtigen:

- Die Helfer sollten keinen langen Extraweg bei der Fahrt haben
- Bei Abholung eines Sperrguts bei längeren Strecken, sollten ebenfalls Helfer angezeigt werden, die ebenfalls längere Strecken zurücklegen müssen.
- Die Schwierig- und Leichtigkeiten der einzelnen Benutzer sollten sich möglichst gut ergänzen.
- Der Benutzer sollte den für sich am besten Helfer angezeigt bekommen, der sich möglichst gut mit den eigenen Eigenschaften ergänzt

Um dies zu bewerkstelligen und alle Kriterien zu erfüllen wurde analysiert, welche Daten gebraucht werden. Diese sind im Punkt "Datenstruktur" in dem Schema "Entry" aufgeführt. Im Rapid Prototyping wurde der Algorithmus schon bereits umgesetzt. Hier hat sich folgender Code ergeben, der hier nun erklärt und teilweise dargestellt wird:

Die Einträge werden nach der Reihe immer miteinander verglichen. Zuerst wird verglichen ob es Überschneidungen bei den Zeitbereichen gibt. Wenn es keinen gibt, wird direkt der nächste Eintrag verglichen, da es wahrscheinlich ist, dass die Benutzer wahrscheinlich keinen Zeitpunkt finden werden die Fahrt zu realisieren. Im nächsten Schritt wird ein Score berechnet, der das Verhältnis der Fahrdistanz des Eintrags jeweils mit der Gesamtroute des anderen Eintrags berechnen soll. Die Distanz bekommt man mit einer Abfrage auf die API des externen Web-Services Graphhopper. Daraus hat sich folgende Gleichung ergeben:

$$\frac{\text{Distanz des Vergleichseintrags}}{\text{Distanz der gesamten Route}} * (\text{Distanz der gesamten Route} - \text{Distanz des Vergleichseintrags})^2$$
$$\text{Distanz des Vergleichseintrags}$$

Die Distanz wird in der Gleichung jeweils in Kilometer angegeben. Der Score soll möglichst niedrig ausfallen, was heißt, dass ein Eintrag mit einem niedrigen Score-Wert ein besseres Ergebnis bietet als ein Eintrag mit einem hohen Score-Wert. Es wird ebenfalls berücksichtigt, dass Einträge mit einer geringen Gesamtstrecke einen niedrigeren Score-Wert haben als Einträge mit einer hohen Gesamtstrecke. Ein Score, der höher als 20 geht, wird nicht berücksichtigt, da der Weg, welcher ein Benutzer alleine zurücklegen müsste nicht verhältnismäßig zu dem Gesamtweg wäre und ein großer Umweg entstehen würde.

Danach werden in einfachen if-Abfragen mit verschiedenen Bedingungen die Schwierig- und Leichtigkeiten miteinander verglichen. Wenn sie sich ergänzen, wird dies auf einen zusätzlichen Score draufgerechnet. Dieser Score kann einen Wert bis maximal 100 erreichen.

Danach wird ein allgemeiner Score erstellt, der beide vorher genannten Scores zusammenrechnet. Nach dem ausgerechneten Score werden die Einträge sortiert und danach ausgegeben.

Packages/Libraries

mongoose (NodeJS)

Für die Verbindung von dem NodeJS und der externen MongoDB wird ein Plugin für die Datenabfrage benötigt. Hier wurde sich für das Modul mongoose entschieden, da es sehr einfach zu bedienen ist, wenn man wenig Extrafunktionen benötigt.

moment (NodeJS)

Um die Zeitabschnitte einfach zu berechnen, wird das Modul moment verwendet. Hier gibt es eine Funktion, die ausgibt, ob sich zwei Zeitintervalle überschneiden.

volley (Android)

Um simple HTTP Request durchzuführen mit den bekannten REST Methoden, wird in Android das Plugin volley verwendet. Dieses Plugin erlaubt es auf die Schnittstelle bzw. den NodeJS zuzugreifen. Dieses Plugin beinhaltet HTTPURLConnection und sendet Requests an eine bestimmte URL.

REST Modellierung

Ressource	Methode	Semantik	content-type(req)	content-type(res)
/entry	GET	Listet alle Einträge	application/json	application/json
	POST	Erstellt neuen Eintrag	application/json	application/json
	PUT	501	application/json	application/json
	DELETE	501	application/json	application/json
/entry/:entryID	GET	Listet alle Informationen über einen Eintrag	application/json	application/json
	POST	501	application/json	application/json
	PUT	Ändert einen Eintrag	application/json	application/json
	DELETE	Löscht einen Eintrag	application/json	application/json
/user	GET	Listet alle User	application/json	application/json
	POST	Erstellt einen neuen User	application/json	application/json

	PUT	501	application/json	application/json
	DELETE	501	application/json	application/json
/user/:userID	GET	Listet alle Informationen über einen User	application/json	application/json
	POST	501	application/json	application/json
	PUT	Ändert eine Userprofil	application/json	application/json
	DELETE	Löscht einen User	application/json	application/json
/user/:userID/entry	GET	Listet alle Einträge und deren Informationen von einem User	application/json	application/json
	POST	Erstellt einen neuen Eintrag	application/json	application/json
	PUT	501	application/json	application/json
	DELETE	Löscht alle Einträge von einem Nutzer	application/json	application/json
/user/:userID/rating	GET	Gibt die Bewertung eines Nutzers aus	application/json	application/json
	POST	Fügt eine Bewertung eines Nutzers hinzu	application/json	application/json
	PUT	Ändert die Bewertung eines Nutzers	application/json	application/json
	DELETE	501	application/json	application/json
/entry/match/:entryID	GET	Listet optimale Einträge, die zu dem Eintrag passen.	application/json	application/json
	POST	501	application/json	application/json
	PUT	501	application/json	application/json
	DELETE	501	application/json	application/json

/message/:messageID	GET	Listet alle Informationen über eine Nachricht	application/json	application/json
	POST	501	application/json	application/json
	PUT	501	application/json	application/json
	DELETE	Löscht eine Nachricht	application/json	application/json
/message/user/:userID	GET	Listet alle Nachrichten von einem User	application/json	application/json
	POST	Schickt eine Nachricht an einen User	application/json	application/json
	PUT	501	application/json	application/json
	DELETE	501	application/json	application/json
/message/user/:userID/user/:userID	GET	Listet die Konversation zwischen zwei Usern	application/json	application/json
	POST	501	application/json	application/json
	PUT	501	application/json	application/json
	DELETE	501	application/json	application/json
/entry/:entryID/loadplan/entryID	GET	Gibt den erstellten Ladeplan zweier Einträge aus	application/json	application/json
	POST	501	application/json	application/json
	PUT	501	application/json	application/json
	DELETE	501	application/json	application/json

Tabelle #07: Rest Modellierung

HTTP Status Codes:

HTTP Status Codes	Reason
-------------------	--------

200	Ok
201	Erfolgreich erstellt
204	PUT / DELETE nicht durchführbar, NO Content
400	Die Anfrage-Nachricht war fehlerhaft aufgebaut.
403	Problem bei der Abfrage vom externen Webservice
404	Konnte nicht gefunden werden.
406	The requested representation format is not available.
500	Prozesserror
501	Nicht implementiert

Tabelle #08: HTTP Status Codes

Topic Modellierung

Um den Benutzern "Push-Notifications" bzw. Benachrichtigungen zu übermitteln muss eine Verbindung zu Googles Firebase hergestellt werden. Die "Push-Notifications" sind in Firebase implementiert und so ist es notwendig die Firebase API zu nutzen.

Benachrichtigungen werden an den Benutzer geschickt, wenn folgendes passiert:

1. Wenn dem Benutzer eine neue Nachricht von einem anderen Benutzer erreicht. So kann der Benutzer schneller auf die Nachricht reagieren und muss nicht jedes mal sein Smartphone herausnehmen um seine Nachrichten zu checken. So kann eine Fahrt schneller und mit weniger Problemen gestartet und durchgeführt werden.
2. Wenn der Benutzer bewertet wird. So kann der Benutzer direkt feststellen, wie er von einem anderem Benutzer nach einer Fahrt bewertet wurde.
3. Wenn ein Inserat von beiden Seiten bestätigt wird, wird ebenfalls eine Benachrichtigung verschickt, da der Benutzer dann zeitnah von der Bestätigung erfährt ohne immer auf dem Smartphone zu gucken.

Daraus entsteht folgende Topic-Modellierung mit folgenden Publish und Subscribe Methoden:

Subscriptions:

- **Eingehende Nachrichten:** /messages/user/userID
- **Eingehende Bewertungen:** /user/:userID/rating
- **Bestätigung der Fahrt:** /entry/succeed

Publishing:

- **Neue Nachricht:** /messages/user/userID
- **Neue Bewertungen:** /user/:userID/rating
- **Bestätigung der Fahrt:** /entry/succeed

Quellenverzeichnis:

Deborah J. Mayhew. The Usability Engineering Lifecycle. 1999 by Academic Press. ISBN 1-55860-561-4.

Material Design by Google, <https://material.io/> (Stand: 16.12.2018)

Google Maps API by Google <https://cloud.google.com/maps-platform> (Stand 16.12.2018)

OpenStreetMap <https://www.openstreetmap.de/> (Stand 16.12.2018)

<https://developer.android.com/design/> (Stand 16.12.2018)

Was schafft Vertrauen in der Onlinewelt? von Chronos & BlaBlaCar

https://d2hgsaoq1cfj21.cloudfront.net/blogstatics/images/de/trustman/trustman_german.pdf
(Stand 16.12.2018)

Anhang:

User Profiles

Demografie	18-25 Jahre Männlich oder Weiblich Guter körperlicher Zustand
Sozioökonomischer Status	Student oder in Ausbildung geringes Einkommen
Motivation	Möchte den Transport seines Sperrguts kostengünstig und unter

	fairen Bedingungen ermöglichen. Sucht jemanden, der Zugang zu einem Transportfahrzeug hat und dieses führen kann und darf. Möchte hohe Lieferkosten vermeiden. Ist neu in der Stadt und würde gern Kontakte knüpfen.
Technologie	Smartphone & Computer
Technische Erfahrung	Viel Erfahrung im Umgang mit technischen Systemen
Fachkenntnis	Kann/darf kein Fahrzeug führen Durchschnittliche Erfahrung in der Montage von Möbeln. Keine Erfahrung in der Installation von Elektrogeräten
Aufgabe	Daten eintragen, vorgeschlagene Helfer überprüfen, Helfer kontaktieren

Tabelle #09: User Profile #1

Demografie	35 Jahre und älter Männlich oder Weiblich Guter körperlicher Zustand
Sozioökonomischer Status	Beruf mit handwerklichem Schwerpunkt Durchschnittliches bis gutes Einkommen
Motivation	Möchte den Transport seines Sperrguts kostengünstig und unter fairen Bedingungen ermöglichen. Sucht jemanden, der dabei hilft das Sperrgut in seine Wohnung (3. Etage) zu tragen, da einige Teile alleine nicht getragen werden können. Möchte hohe Lieferkosten vermeiden.
Technologie	Smartphone & Computer
Technische Erfahrung	Durchschnittliche Erfahrung im Umgang mit technischen Systemen
Fachkenntnis	Kann/darf ein Transportfahrzeug führen Besitzt berufsbedingt ein Transportfahrzeug Viel Erfahrung im Beladen von Fahrzeugen Viel Erfahrung in der Montage von Möbeln Viel Erfahrung in der Installation von Elektrogeräten
Aufgabe	Daten eintragen, vorgeschlagene Helfer überprüfen, Helfer

	kontaktieren
--	--------------

Tabelle #10: User Profile #2

Demografie	55 Jahre und älter Männlich oder Weiblich Schlechter körperlicher Zustand
Sozioökonomischer Status	Rentner/Frührentner Durchschnittliches bis geringes Einkommen
Motivation	Möchte den Transport seines Sperrguts kostengünstig und unter fairen Bedingungen ermöglichen. Sucht jemanden, der dabei hilft das Sperrgut in das Transportfahrzeug zu laden, in seine Wohnung zu tragen und zu montieren. Möchte hohe Lieferkosten und lange Lieferzeiten vermeiden.
Technologie	Computer & evtl. Smartphone
Technische Erfahrung	Durchschnittliche bis geringe Erfahrung im Umgang mit technischen Systemen
Fachkenntnis	Kann/darf ein Fahrzeug führen
Aufgabe	Daten eintragen, vorgeschlagene Helfer überprüfen, Helfer kontaktieren

Tabelle #11: User Profile #3

Demografie	18 Jahre und älter Männlich oder Weiblich Körperlich eingeschränkt
Sozioökonomischer Status	Gemischt
Motivation	Möchte den Transport seines Sperrguts kostengünstig und unter fairen Bedingungen ermöglichen. Sucht jemanden, der dabei hilft das Sperrgut in das Transportfahrzeug zu laden und in seine Wohnung zu tragen Sucht jemand, der ein Transportfahrzeug fahren kann. Möchte hohe Lieferkosten und lange Lieferzeiten vermeiden
Technologie	Computer & Smartphone

Technische Erfahrung	Durchschnittliche bis gute Erfahrung im Umgang mit technischen Systemen
Fachkenntnis	Wenig Erfahrung in allen Bereichen des Transports von Sperrgut Kann/darf kein Transportfahrzeug führen
Aufgabe	Daten eintragen, vorgeschlagene Helfer überprüfen, Helfer kontaktieren

Tabelle #12: User Profile #4

Demografie	18 Jahre und älter Männlich oder Weiblich Durchschnittlicher körperlicher Zustand
Sozioökonomischer Status	Durchschnittliches bis gutes Einkommen Engagiert im Umweltschutz
Motivation	Möchte hohe Lieferkosten und lange Lieferzeiten vermeiden Möchte die Umwelt schonen und daher Fahrten mit unvollständig gefüllten Transportfahrzeugen vermeiden Möchte den Transport seines Sperrguts kostengünstig und unter fairen Bedingungen ermöglichen Würde gerne neue Kontakte knüpfen
Technologie	Computer & Smartphone
Technische Erfahrung	Durchschnittliche Erfahrung im Umgang mit technischen Systemen
Fachkenntnis	Kann/darf ein Transportfahrzeug führen Durchschnittliche Erfahrung in Transport und Montage
Aufgabe	Daten eintragen, vorgeschlagene Helfer überprüfen, Helfer kontaktieren

Tabelle #13: User Profile #5

Demografie	18 Jahre und älter Männlich oder Weiblich Durchschnittlicher körperlicher Zustand
Sozioökonomischer Status	Geringes Einkommen
Motivation	Möchte hohe Lieferkosten und lange Lieferzeiten vermeiden

	Möchte den Transport seines Sperrguts kostengünstig und unter fairen Bedingungen ermöglichen
Technologie	Computer & Smartphone
Technische Erfahrung	Durchschnittliche Erfahrung im Umgang mit technischen Systemen
Fachkenntnis	Kann/darf ein Transportfahrzeug führen Durchschnittliche Erfahrung in Transport und Montage
Aufgabe	Daten eintragen, vorgeschlagene Helfer überprüfen, Helfer kontaktieren

Tabelle #14: User Profile #6

Demografie	18 Jahre und älter Männlich oder Weiblich Guter körperlicher Zustand
Sozioökonomischer Status	Hobby-Handwerker Durchschnittliches Einkommen
Motivation	Möchte Kontakte knüpfen Würde gerne mit Anderen an einem gemeinsamen Projekt arbeiten
Technologie	Computer & Smartphone
Technische Erfahrung	Durchschnittliche Erfahrung im Umgang mit technischen Systemen
Fachkenntnis	Kann/darf ein Transportfahrzeug führen Besitzt ein Transportfahrzeug Viel Erfahrung im Beladen von Fahrzeugen Viel Erfahrung in der Montage von Möbeln Viel Erfahrung in der Installation von Elektrogeräten
Aufgabe	Daten eintragen, vorgeschlagene Helfer überprüfen, Helfer kontaktieren

Tabelle #15: User Profile #7

Demografie	18 Jahre und älter Männlich oder Weiblich Durchschnittlicher körperlicher Zustand
-------------------	---

Sozioökonomischer Status	Sozial engagiert Durchschnittliches Einkommen
Motivation	Möchte Kontakte knüpfen Möchte anderen Helfen ihre Transportprobleme zu lösen
Technologie	Computer & Smartphone
Technische Erfahrung	Durchschnittliche Erfahrung im Umgang mit technischen Systemen
Fachkenntnis	Kann/darf ein Transportfahrzeug führen Durchschnittliche Erfahrung in Transport und Montage
Aufgabe	Daten eintragen, vorgeschlagene Helfer überprüfen, Helfer kontaktieren

Tabelle #16: User Profile #8

Use-Cases

User Action	System Response
Möchte Informationen eintragen Trägt die Daten ein	Stellt Eingabefenster zum eintragen der Daten dar. Gibt Rückmeldung zur Gültigkeit der Daten und speichert diese wenn sie gültig sind.

Tabelle #17: Essential Use-Case #1: Informationen eintragen

User Intention	System Response
Der Benutzer möchte Sperrgut transportieren, benötigt jedoch an einer oder mehreren Stelle Hilfe.	Das System visualisiert eine Option zum Anlegen eines Eintrags.
Er startet die Interaktion mit dem System und hält alle relevanten Daten bereit.	Das System liefert die Eingabefelder für die Nutzereingaben und liefert bei Bedarf weitere Informationen zu den benötigten Daten.
Der Benutzer trägt seine Informationen in die vorgesehenen Felder ein und bestätigt die Eingabe.	Das System prüft die eingetragenen Daten auf Vollständigkeit und Validität. Sind die Daten nicht korrekt wird der Nutzer darauf

	hingewiesen und gebeten diese zu korrigieren. Sind alle Eingaben gültig erstellt das System anhand dieser Daten einen Eintrag.
--	--

Tabelle #18: Concrete Use-Case #1: Informationen eintragen

User Action	System Response
Möchte Helfer finden	Gibt eine Übersicht der geeignetsten Helfer aus
Prüft die möglichen Helfer und wählt einen aus.	Präsentiert weitere Informationen zum ausgewählten Helfer und gibt Möglichkeiten diesen zu kontaktieren
Kontaktiert den Helfer oder kehrt zurück zur Übersicht um weitere Helfer zu prüfen.	

Tabelle #19: Essential Use-Case #2: Helfer finden

User Intention	System Response
Der Benutzer möchte einen geeigneten Helfer finden.	Das System matcht den Eintrag des Nutzers mit den Einträgen anderer User. Die Einträge mit den höchsten Übereinstimmungen werden sortiert angezeigt.
Der Benutzer prüft die Informationen der potentiellen Helfer und vergleicht diese miteinander. Er möchte weitere Informationen zu einem Helfer und wählt die Anzeige dieses Helfers an.	Das System zeigt eine Detailseite, die alle relevanten Informationen zum Helfer und seinem Transport beinhaltet.
Nun hat der Benutzer die Möglichkeit diesen Helfer zu kontaktieren. Hat sich der Benutzer für diesen Helfer entschieden nimmt er den Kontakt auf.	Das System visualisiert einen Chat, durch den die Kommunikation zu einem Helfer aufgenommen werden kann.
Andernfalls kehrt er zurück zur Übersicht und vergleicht weitere potentielle Helfer.	Das System zeigt dem Nutzer die Übersicht der potenziellen Helfer an.
Findet er jedoch keinen geeigneten Helfer, hat er weiterhin die Möglichkeit von einem anderen Nutzer selbst als Helfer gefunden und kontaktiert zu werden.	Das System nutzt den Eintrag des Benutzers beim Matching anderer User und informiert diesen, falls ein anderer Nutzer Kontakt aufnehmen möchte.
Außerdem kann er seine Suche erneut	Das System matcht den Eintrag des

durchlaufen, um zu überprüfen ob neue Einträge eingegangen sind, die seine Ansprüche erfüllen.	Benutzers erneut und zeigt die aktuelle Liste potentieller Helfer an.
--	---

Tabelle #20: Concrete Use-Case #2: Helfer finden

User Action	System Response
Möchte Informationen zu einem Eintrag ändern Wählt den entsprechenden Eintrag aus Ändert die gewünschten Informationen	Gibt Liste seiner Einträge aus Gibt detaillierte Ansicht aus, in der die Informationen geändert werden können. Gibt Rückmeldung zur Gültigkeit der Daten und speichert diese wenn sie gültig sind.

Tabelle #21: Essential Use-Case #3: Eintrag bearbeiten

User Intention	System Response
Der Benutzer möchte einen seiner Einträge ändern Er startet das System.	Das System zeigt dem Benutzer eine Übersicht seiner Einträge
Der Benutzer wählt den Eintrag aus, den er ändern möchte.	Das System liefert dem Benutzer eine Detailansicht zu dem ausgewählten Eintrag
Der Benutzer wählt die Informationen aus, die er bearbeiten möchte und trägt die neuen Informationen ein.	Das System prüft die eingetragenen Daten auf Vollständigkeit und Validität. Sind die Daten nicht korrekt wird der Nutzer darauf hingewiesen und gebeten diese zu korrigieren. Sind alle Eingaben gültig aktualisiert das System den Eintrag.

Tabelle #22: Concrete Use-Case #3: Eintrag bearbeiten

User Action	System Response
Möchte die vorgeschlagene Strecke einsehen Wählt den entsprechenden Eintrag aus Wählt den Streckenvorschlag aus	Gibt Liste seiner geplanten Fahrten aus Gibt detaillierte Ansicht aus. Liefert die vorgeschlagene Strecke

Tabelle #23: Essential Use-Case #4: Strecke einsehen

User Intention	System Response
----------------	-----------------

Der Benutzer möchte einen gemeinsamen Transport durchführen und dafür die vom System vorgeschlagene Strecke nutzen. Er startet das System.	Das System visualisiert eine Option, um die geplanten Transporte anzuzeigen.
Der Benutzer wählt den gesuchten Transport aus.	Das System zeigt dem Benutzer eine Repräsentation der gemeinsamen Fahrt mit allen relevanten Informationen an.
Der Nutzer wählt unter den Daten weitere Informationen zur Strecke aus.	Das System zeigt die vorgeschlagene Strecke mit allen Stopps mittels Karte an.

Tabelle #24: Concrete Use-Case #5: Strecke einsehen

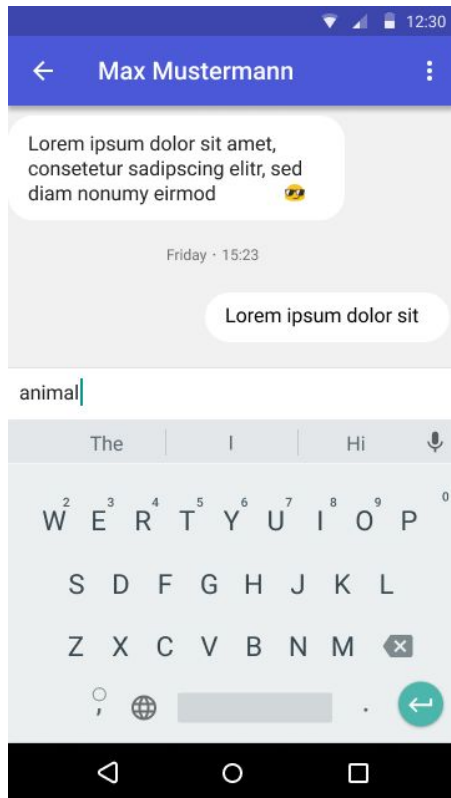
User Action	System Response
Möchte den vorgeschlagenen Ladeplan einsehen	Gibt Liste seiner geplanten Fahrten aus Gibt detaillierte Ansicht aus. Liefert den vorgeschlagenen Ladeplan
Wählt den entsprechenden Eintrag aus	
Wählt den Ladeplan aus	

Tabelle #25: Essential Use-Case #5: Ladeplan einsehen

User Intention	System Response
Der Benutzer möchte einen gemeinsamen Transport durchführen und dafür den vom System vorgeschlagenen Ladeplan nutzen. Er startet das System.	Das System visualisiert eine Option, um die geplanten Transporte anzuzeigen.
Der Benutzer wählt den gesuchten Transport aus.	Das System zeigt dem Benutzer eine Repräsentation der gemeinsamen Fahrt mit allen relevanten Informationen an.
Der Nutzer wählt unter den Daten weitere Informationen zum Ladeplan aus.	Das System zeigt den Ladeplan des Sperrguts mittels Grafik an.

Tabelle #26: Concrete Use-Case #5: Ladeplan einsehen

UI-Prototyp



Evaluation Domänenrecherche

Vertrauen in der Onlinewelt

Da man im Internet oftmals mit Betrügern, Abzockmaschinen und Fake Angeboten konfrontiert wird, ist es wichtig Vertrauen zum System und besonders auch zwischen den Nutzern zu schaffen. Besonders das Vertrauen zwischen den Nutzern ist dafür notwendig, da die Nutzer aufgerufen werden sich gegenseitig zu helfen.

Um einen Überblick darüber zu bekommen, welche Möglichkeiten bestehen, um dies zu realisieren wird das Konzept von "BlaBlaCar" unter die Lupe genommen. BlaBlaCar vermittelt schon seit Jahren erfolgreich Mitfahrgelegenheiten zwischen ihren Usern. Hier spielt das Vertrauen eine ähnlich starke Rolle, da die Nutzer auch hier aufgefordert werden sich gegenseitig zu helfen.

Um zu Verstehen welche Maßnahmen BlaBlaCar ergreift, um Vertrauen in Netz zu schaffen wurde 2012 eine Umfrage zum Thema "Was schafft Vertrauen in der Onlinewelt?" durchgeführt, welche anschließend von dem Soziologen Bruno Marzloff im Auftrag von BlaBlaCar analysiert wurde.

Wie zu erwarten ist das Vertrauen in einen fremden im Internet grundsätzlich erstmal sehr gering. Laut Umfrage kann dieser Wert jedoch durch weitere Profilinformationen leicht gesteigert werden. So haben ein Profilbild, eine überprüfte Nummer, positive Bewertungen einen positiven Einfluss auf den Grad des Vertrauens. Dabei lässt sich die Vertrauenswürdigkeit mit einem vollständigem Profil sogar mehr als verdoppeln. (1.92/5 Ohne Info zu 4.23/5 mit

vollständigem Profil) Im Vergleich kommt ein Nutzer mit vollständigem Profil sogar nah an Freunde und Familie heran, welche bei einer Vergleichsumfrage mit etwas über 4.5/5 abgeschlossen haben. Des Weiteren schätzten etwa 75% der Befragten die Rolle von BlaBlaCar als Vertrauenswürdiger Dritter ein, der auf Seriosität der Nutzer großen Wert legt.

Fazit

Abschließend lässt sich festhalten, dass die Vertrauenswürdigkeit in auf Teilen basierten Onlinegemeinschaften sehr wichtig ist und dass diese mit einigen Schritten stark gesteigert werden kann. Diese Erkenntnisse sollen im weiteren Projektverlauf beachtet werden, damit ein hoher Vertrauensgrad in der Onlinegemeinschaft erreicht werden kann.