

WG App

Matthias Ramsauer

wg-app@matthias-ramsauer.de

24. Oktober 2019

INHALTSVERZEICHNIS

1	Einführung	2
1.1	Beschreibung und Vision	2
1.2	Zielgruppe	2
1.3	Vermarktungsstrategie	2
2	Planung und Konzeption	3
2.1	Funktionale Anforderungen	3
2.1.1	Authentifizierung	4
2.1.2	WG Erstellen	4
2.1.3	WG Beitreten	4
2.1.4	Vorratsverwaltung	4
2.1.5	Kostenübersicht	5
2.1.6	Lesen von Barcodes	6
2.1.7	Event Logging	6
2.1.8	Ressourcenverwaltung	6
2.1.9	Lesen von NFC-Tags	7
2.1.10	Löschen der kompletten Nutzerdaten	7
2.1.11	Nutzerdaten als JSON	7
2.1.12	iOS Version	8
2.1.13	In-App-Payment	8
2.2	Mockup & Navigation	8

1 EINFÜHRUNG

1.1 BESCHREIBUNG UND VISION

In einer Wohngemeinschaft lebt man mit verschiedenen parallel laufenden Individuen zusammen. Diese haben ihren eigenen Prozessablauf und teilen sich unterschiedlichste Ressourcen. In Wohngemeinschaften, die keinem starren System untergeordnet sind, kann es ein organisatorischer Albtraum sein, diese Ressourcen zu verteilen und zu protokollieren. Besonders wenn sich das System aus verschiedensten Gründen skaliert.

Um im Zeitalter von Cloud mithalten zu können, bietet sich hierfür eine technische Lösung an, die sich möglichst einfach und ohne großen zusätzlichen Aufwand betreiben lassen soll.

Den Mitgliedern einer WG soll es somit möglich sein, exklusiven Zugriff auf einzelne Ressourcen schnell und einfach anmelden zu können. Des Weiteren sollen die Kosten für Güter fair verteilt werden. Da die Zeit ein teures Gut ist, soll dies möglichst schnell durchführbar sein. Dafür bieten sich NFC-Tags oder Barcodes an.

1.2 ZIELGRUPPE

Die Zielgruppe sind Wohngemeinschaften mit lockeren Gewohnheiten und dynamischen Tagesabläufen. Dies ist unter anderem bei Studenten WGs gegeben. Jedoch sind auch moderne Familien mit älteren Kindern möglich.

Beim derzeitigen Projektumfang ist eine nativ implementierte Android App angedacht. Bei der Android Version wird auf eine Abdeckung von minimal 80 % abgezielt.

1.3 VERMARKTUNGSSTRATEGIE

Um aus der App Einkommen zu generieren soll die App kostenpflichtig im PlayStore und zukünftig auch im AppStore zur Verfügung stehen. Zusätzlich soll man zusätzlichen Speicher als monatliches Abo hinzubuchen können, um größere WGs zu unterstützen.

Die App soll in den verschiedenen Verkaufsplätzen in die Kategorien „Produktiv“ und „Smart Home“ gehören. Als Alleinstellungsmerkmal soll die Nutzung von Barcodes und NFC-Tags gehören, um den Workflow zu beschleunigen.

Da es sich hier um eine Studienarbeit handelt, ist keine Aufwandsentschädigung angedacht, da es sich hier um ein rein theoretisches Konzept handelt.

2 PLANUNG UND KONZEPTION

2.1 FUNKTIONALE ANFORDERUNGEN

Die Anforderungen haben unterschiedliche Prioritäten, diese werden mit der MoSCoW-Priorisierung beschrieben. Nach einer kurzen Übersicht werden die einzelnen Features genauer erklärt.

Must have

- Authentifizierung
- WG Erstellen
- WG Beitreten
- Vorratsverwaltung
- Kostenübersicht
- Lesen von Barcodes

Should have

- Event Logging
- Ressourcenverwaltung
- Löschen der kompletten Nutzerdaten

Could have

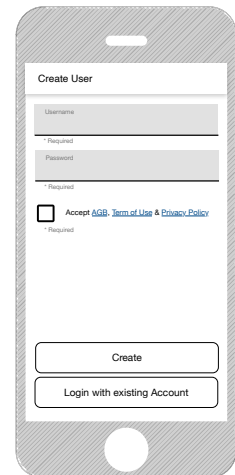
- Nutzerdaten als JSON
- Lesen von NFC-Tags
- HTTP Application-Schema

Won't have

- iOS Version
- In-App-Payment

2.1.1 AUTHENTIFIZIERUNG

Der Nutzer muss sich am Anfang einen Account erstellen. Der Login und die Registrierung sollen durch die von Firebase verfügbaren Funktionen abgedeckt werden. Es sollen Placeholder für spätere Links für AGB, Datenschutz- und Nutzungsbedingung vorhanden sein.



2.1.2 WG ERSTELLEN

Wenn der Nutzer eingeloggt ist, kann er entweder eine WG erstellen, oder einer beitreten. Bei der Erstellung kann ein optionaler Name angegeben werden und der Ersteller wird Teil der WG. Daraufhin wird der Nutzer auf den „Share“ Screen weitergeleitet. Dieser ist später auch in den Einstellungen verfügbar.

Der Link läuft nach einer konstanten Dauer ab und muss dann neu erstellt werden und gilt nur für diese WG. Ein möglicher Link wäre:

`https://<domain>/G4tAmDaM`

Oder ohne http, was jedoch verhindert, dass die Links zu einer entsprechenden Downloadseite weiterleiten können.

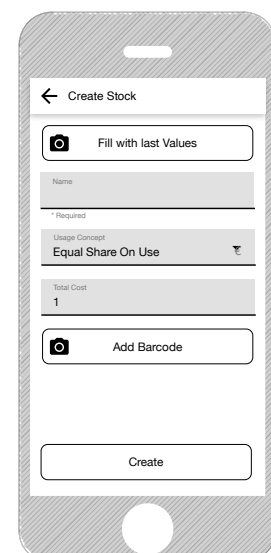
`wgapp://G4tAmDaM`

2.1.3 WG BEITRETEN

Der Nutzer öffnet den geteilten Link über die App. Daraufhin wird dem Nutzer der Name der WG, falls vorhanden, und die Mitglieder gezeigt und er kann die Einladung annehmen, oder ablehnen. Falls der Link nicht über die App geöffnet werden kann, soll es eine Möglichkeit geben den Link einzugeben.

2.1.4 VORRATSVERWALTUNG

Bei der Vorratsverwaltung geht es darum Einkäufe bzw. Kosten zu teilen. Nach einem Einkauf soll es dem Nutzer möglich sein einen Vorrat hinzuzufügen. Das geschieht entweder durch manuelles Ausfüllen des entsprechenden Formulars oder durch Einscannen des Barcodes das die Werte aus dem letzten Vorrat mit demselben Barcode setzt.



Die Formularpunkte bestehen aus:

Name Der Name des Elements um es leicht wieder zu finden

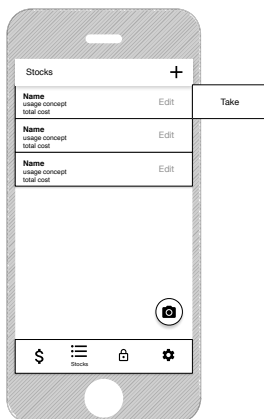
Teil-Konzept Ist der Wert, der angibt wie die Kosten unter den Benutzern geteilt werden. Mögliche Werte sind hier:

- **Gleichverteilung bei Benutzung**
Man bezahlt seinen Teil ab dem Zeitpunkt an dem man es benutzt. Bsp.: Ein Laib Brot
- **Gleichverteilung**
Man bezahlt seinen Teil ab dem Kauf.
Bsp.: Neuer Duschkopf
- **Anzahl**
Man bezahlt die Anzahl an Elementen die man benutzt.
Bsp.: Kaffepads

Wenn Anzahl gewählt wird, muss ein zusätzliches Feld „Anzahl“ freigeschaltet werden.

Gesamtkosten Die Gesamtkosten des Elements

Barcode Der Barcode wird durch scannen hinzugefügt



Zum Bearbeiten bzw. Löschen kann man bei einem Vorrat die „Edit“ Aktion durchführen.

Vorräte können dann einfach durch Scannen des Elements, oder durch suchen und durchführen der „Take“ Aktion entnommen werden.

Ist ein Element nicht mehr vorrätig wird es aus der Übersicht und aus den scannbaren Elementen entnommen. Dabei kann eine Mitteilung versendet werden.

2.1.5 KOSTENÜBERSICHT

Die Kostenübersicht zeigt eine Aggregation der „Take“ Events der verschiedenen Mitglieder in Bezug auf den derzeit eingeloggteten Nutzer. Das bedeutet, wenn ein Vorrat x entnommen wird und der von Mitbewohner A hinterlegt wurde, schuldet sich A den verwendeten Wert von Vorrat x. Diese Schulden sollen dann in einer Liste



aller Mitbewohner angezeigt werden. Wenn ein Mitbewohner Schulden beim Nutzer zurückzahlt, kann der Empfänger die Schulden entsprechend als getilgt hinterlegen.

2.1.6 LESEN VON BARCODES

Um den Workflow möglichst schnell zu halten, kann man so gut wie alles mit Barcodes steuern. Zum Lesen von Barcodes bietet Firebase eine entsprechende Bibliothek an, die verwendet werden soll.

2.1.7 EVENT LOGGING

Zum Speichern der Daten und um möglichst große Transparenz für die Nutzer zu schaffen, bietet sich an Event Logging bzw. Event Sourcing zu verwenden. Dabei werden die Events jeweils in die Datenbank hinterlegt, aber es gibt keine Datenstruktur die bereits aggregierte Werte enthält und die Daten werden „append-only“ hinzugefügt. Firebase bietet hierfür auch entsprechende Methoden, um auf Datenänderungen in der App zu horchen und somit die Aktualisierung der Daten auf einzelne Änderungen zu minimieren. Um die initiale Anzeige der Daten möglichst schnell zu gestalten bietet es sich an einen Snapshot mit aggregierten Daten auf dem Gerät selbst zu hinterlegen. Damit ist die App theoretisch auch zum Teil offline fähig.

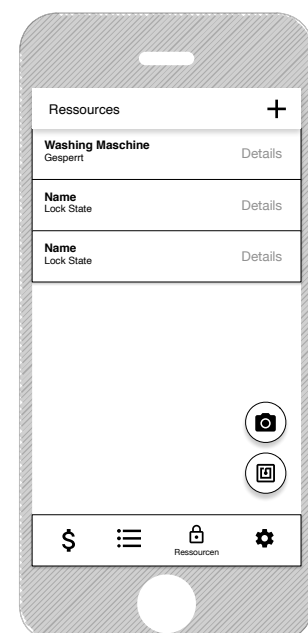
Der Event Log sollte in der App auch einsehbar sein, so können die einzelnen Mitglieder einer WG nachvollziehen wer welche Änderung an welchen Daten wann vorgenommen hat und in welcher Reihenfolge.

2.1.8 RESSOURCENVERWALTUNG

In einer WG gibt es bestimmte geteilte Ressourcen, auf die gleichzeitig immer nur eine Person Zugriff hat. Um seinen Tagesablauf optimal zu Planen, bietet es sich deswegen an, diese Ressourcenbelegungen zu hinterlegen. Zu diesen Ressourcen zählt unter anderem die Waschmaschine, oder auch der Gemeinschaftsraum für eine Party.

Daraus folgt die entsprechenden Anforderungen, Ressourcen statisch zu definieren und für jede Ressource entsprechende exklusive Zugriffe zu hinterlegen. Diese können unter anderem auch in der Zukunft stattfinden.

Und um den Workflow möglichst schnell zu gestalten, bietet sich hier auch wieder eine Barcode/NFC-Tag Lösung an.



Werte zum Anlegen einer Ressource:

Name Der Name der Ressource, um sie in der Ressourcenauswahl wiederzufinden

Lock-Type Dies ist die Art, wie die Ressource gesperrt bzw. entsperrt werden kann. Je nach Auswahl muss hier auch ein entsprechendes Scannen erfolgen. Mögliche Werte sind hier NFC-Tag, Barcode und Manuell.

Parallel Usages Von wie vielen Personen kann eine Ressource parallel belegt werden ohne einen exklusiven Zugriff zu benötigen. Bsp.: die Küche.

Man soll eine Ressource dann auch aus der Detailübersicht bearbeiten und im Bearbeitungsscreen löschen können.

In der Ressourcenübersicht soll man auch direkt einen Scanvorgang einleiten können, um eine Ressource entweder zu sperren, oder entsprechend zu entsperrern. Alternativ kann man auch auf die Detailübersicht der Ressource gehen und sie dort sperren oder entsperrern. In dieser Übersicht sieht man die letzten paar Sperren, alle zukünftigen Sperren und derzeit laufende Sperren. Man kann auch eine zukünftige Sperre anlegen. Diese besteht aus dem Zeitpunkt, der Dauer und einer kurzen Beschreibung des Anlasses der Sperre.

2.1.9 LESEN VON NFC-TAGS

Für die Sperrung der Ressourcen bieten sich auch NFC-Tags an. Diese sind etwas besser zu verarbeiten als Barcodes.

2.1.10 LÖSCHEN DER KOMPLETTEN NUTZERDATEN

Die Löschung der kompletten Nutzerdaten vom System ist eine datenschutzmäßige Vorschrift und sollte im aktuellen Projektzeitraum umgesetzt werden. Dabei werden aus den Events der WG die Referenzen zum entsprechenden Nutzer entfernt. Außerdem werden alle Daten aus dem Nutzer Objekt gelöscht und somit muss der Nutzer sich einen neuen Account erstellen.

Ein entsprechender Warnungs-Dialog sollte gezeigt werden.

2.1.11 NUTZERDATEN ALS JSON

Alle Informationen des Nutzers müssen in maschinenlesbarer Form dem Nutzer dargeboten werden können, das schreibt die DSGVO vor.

2.1.12 iOS VERSION

In späteren Verlauf der App sollte auch eine iOS Version veröffentlicht werden, um möglichst gute Abdeckung zu haben.

2.1.13 IN-APP-PAYMENT

Für das spätere Bezahlen von zusätzlichem Speicher wären In-App-Payments gedacht. Jedoch werden diese im aktuellen Projektumfang nicht verlangt.

2.2 MOCKUP & NAVIGATION

Die folgende Abbildung zeigt die Entwürfe der geplanten Oberflächen und deren Navigierbarkeit. Aus Gründen der Übersichtlichkeit wurde weder die Up-Navigation (über den Pfeil links in der ActionBar) noch die Navigation über die Zurück-Taste des Android-Gerätes explizit visualisiert. Validierungsdialoge und Bearbeitungsscreens werden nicht gezeigt. Bearbeitungsscreens sehen genauso aus wie Erstellungsdialoge nur mit angepasstem Button und einem zusätzlichen Löschesymbol in der ActionBar.

Die blau dargestellten Pfade stellen in der Abbildung die Vorwärtsnavigation dar. Die roten Pfade zeigen dagegen, welche Möglichkeiten (neben der Up-Navigation und der Zurück-Taste) existieren, um zur vorher angezeigten Ansicht zurückzukehren.

