SmartWG

Pflichtenheft

Oskar Marciniak | Joshua Bräucker  
*Mobile Anwendungen*

Inhaltsverzeichnis

[Intro 3](#_Toc106014950)

[Funktionsübersicht - MoSCoW 3](#_Toc106014951)

[Activitys 4](#_Toc106014952)

[Übersicht 4](#_Toc106014953)

[Putzplan: Highscore Liste 4](#_Toc106014954)

[Putzplan: Putz Uhr 4](#_Toc106014955)

[Einkaufsliste 4](#_Toc106014956)

[Einstellungen 4](#_Toc106014957)

[Benutzersystem 5](#_Toc106014958)

[Anmeldescreen 5](#_Toc106014959)

[Einstellungen 5](#_Toc106014960)

[WC-Status 5](#_Toc106014961)

[Datenmodell – ERM 6](#_Toc106014962)

[Backend 6](#_Toc106014963)

[Rahmenbedingungen 7](#_Toc106014964)

[MockUps 7](#_Toc106014965)

# Einleitung

**Thema:** Haushaltsverwaltungsapplikation mit Fokus auf Wohngemeinschaften

**Pitch**

* Eine Mobile Anwendung, welche die Organisation einer Wohngemeinschaft vereinfacht. Die App soll primär dabei helfen die Lebensmittelbeschaffung und Ordnung zwischen den Mitbewohnern im Haushalt zu unterstützen. Bonusfeatures wären zum Beispiel die Integration einer Statusanzeige, ob Toilette oder Bad besetzt sind und Widgets für den Home Screen.

**Zielgruppen**: Wohngemeinschaften

# Funktionsübersicht - MoSCoW

**Anforderungen**

* MUST HAVE
  + Putzplan
    - High Score Liste + / Putzuhr
      * Bad / Toilette / Küche
  + Einkaufsliste , falls für alle aus der WG eingekauft wird soll die Rechnung hochgeladen werden. Mitbewohner können nachvollziehen für wie viel und was eingekauft wurde. (Rechnungshistorie) -
  + Die Features sollen erst für eine Instanz (WG) realisiert werden (erstmal private Anwendung) +
* SHOULD HAVE
  + Benutzersystem (zentrale Datenbank) +
    - Nutzer können sich vor der Nutzung der Anwendung registrieren und dannach anmelden.
  + Vorratsverwaltung/Einkaufsliste in CSV Format übertragbar -
* COULD HAVE
  + Features auf mehrere Instanzen erweitern +
  + WGGBS (AGBs des Hauses) +
  + Widgets
  + Toilette Besetzstatus +
    - Lichtschrankensensor / Bewegungssensor mit einem Raspberry Pi, Sensor schickt Status sobald Sensor ausgelöst wird.
* WON’T HAVE
  + Interner Kalender - Rad nicht neu erfinden, wenn nötig extern einen Kalender einbinden
  + Wohngemeinschaft Chat – Keine kommunikations App, dafür Signal/Whatsapp

# Stand der Technik

# Activitys - Anforderungsdokumentation

## Übersicht

Homscreen der Applikation, der den Zugriff auf die Navigierungselemente: Toilettenstatus, WGBs, WG Bild, Putzplan, Einkaufen & Einstellungen bietet.

## Putzplan: Highscore Liste

Die Highscore Box führt eine Strichliste (Zählliste) über die ausgeführten Putzdienste der Mitbewohner:innen.

Darunter werden die zuletzt ausgeführten Dienste in Chronologisch absteigender Reihenfolge angezeigt.

Zwei Filteroptionen:

1. Raum (Default: Alle)
2. Zeit (Default: Monat)

Gemachter Dienst wird über das + Symbol hinzugefügt. Hierbei wird angegeben welcher Raum geputzt wurde.

Alternative Anforderungen:

* Was gilt als geputz?
* Sauberkeitsstufen hinzufügen?
* Geputzte Elemente hinzufügen?
* Definition of Geputzt in den Einstellungen?
* Notizen?
* Blame?

## Putzplan: Putz Uhr

Eine „Uhr“ mit allen definierten Diensten wird angezeigt. Die WG-Mitglieder werden auf diese Dienste verteilt. Sind mehr WG-Mitglieder als Dienste vorhanden, so wird ein Dinestfeld mit einem „freien“ Feld gefüllt.  
Wollen WG-Mitbewohner:innen angeben, dass Sie ihren Dienst erledigt haben, so müssen Sie lediglich einen Knopf drücken. Die Uhr dreht zu einem von den Nutzer:innen definiertem Datum einen Dienst weiter. In den Einstellungen besteht auch die Möglichkeit, Dienste zu tauschen oder zu verschieben. Passiert dies, werden die anderen Nutzer Benachrichtigt.

Wurde ein Dienst nicht erledigt, so wird dies vermerkt. Z.B in einer Liste. (Missing Feature)

## Einkaufsliste

Es können mehrer Einkaufslisten erstellt werden. Diese werden in einer scrollbaren Ansicht angezeigt. Ein Einkaufslistenelement hat einen Titel, Author:in & einen „zuletzt Bearbeitet“ Tag.

Einer Einkaufsliste können Elemente hinzugefügt werden. Hierbei können Anzahl und Elementname spezifiziert werden. (Missing Feature)

## Einstellungen

Hier werden WG-Informationen und Nutzer-Informationen angezeigt.

## Benutzersystem

Nutzer können sich vor der Nutzung der Anwendung Registrieren und unmittelbar dannach anmelden.

## Anmeldescreen

Widerkehrende Nutzer können sich hier wieder anmelden. Die Option soll bestehen, dass Nutzer angemeldet bleiben, wenn sie die Anwendung erneut aufrufen.

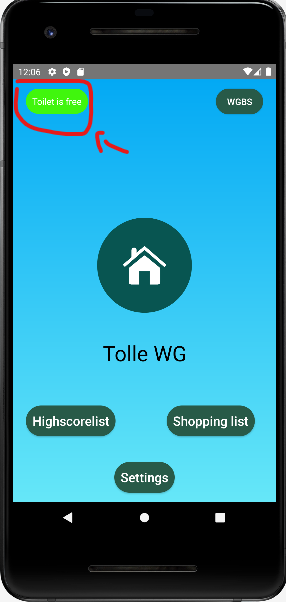
## Einstellungen

Anzeigen und Ändern der ...

1. WGBS
2. Kontoinformationen
3. SA-Code eines Nutzers

## WC-Status

Der WC Status ist das IOT-Feature des Projektes. Ein Raspbery Pi hat einen Sensor, wenn dieser eine Bewegung erkennt, wird ein ein Endpoint im Backend angesprochen. Dieser Endpoint toggelt für einen bestimmten Wert für eine Wohngemeinschaft. (Bool isToilet) Ist der Wert auf 1 so ist die Toilette besetzt, und der Nutzer kann das in der Applikation einsehen. Ist der Wert auf 0 so ist das Label in der App dementsprechend als frei gesetzt.



# Architekturbeschreibung

## Datenmodell – ERM

Im Folgenden wird das Datenmodell der Anwendung beschrieben.

Diagram

Description automatically generated

## Backend

Die Beschreibung und Funktionalitäten des Backends werden im Folgenden beschrieben.

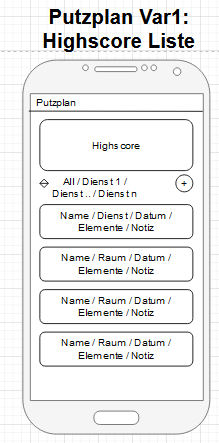
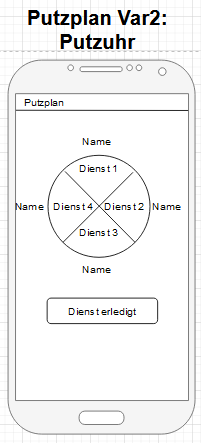
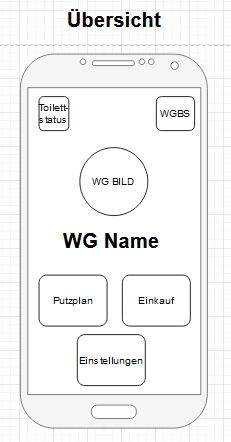
## RaspberryPi - Informationen

# Implementierung

# Test und Useability

# Zusammenfassung

# MockUps



Entwicklungsumgebung: Android Studio

Programmiersprache: Kotlin

Datenbank: MySQL

Laufzeitumgebung: Android API 28

# 