

# **Meilenstein 3**

**HCI 2021S: Interior Designer**

Asım OĞUZ, Dominik PEGLER, Sophia PUM

17. Mai 2021

**Inhaltsverzeichnis**

# 1 Konzept

*Autor: Dominik PEGLER*

## 1.1 Ziele

Es handelt sich um **keine Augmented-Reality-App**. Wir lesen die Realität ein und speichern sie für später ab. Anschließend bearbeiten wir diese Realität, bis wir zum gewünschten Ergebnis kommen. Weil wir nachhaltige Ergebnisse möchten, können diese nicht im selben Augenblick entstehen und erarbeitet werden, wenn die Kamera eingeschaltet ist, wie es bei AR der Fall ist. Durch die Trennung von Scan- und Entscheidungsprozess sollen aus der Hürde geschossene Entscheidungen vermieden und der Nutzer bei einer gut durchdachten Planung unterstützt werden. Netter Nebeneffekt: Die Kamera muss nur zum Scannen eingeschaltet werden, das spart Ressourcen.

Aufgrund der kurzen Entwicklungsdauer von wenigen Wochen und der Unerfahrenheit allerq Teammitglieder im Bereich der Frontend-Entwicklung kann der vorliegende Prototyp nur eine erste grobe Skizzierung unserer Vorstellungen von dieser Applikation sein. Wir sind jedoch zuversichtlich, dass er diese Vorstellungen bereits gut repräsentiert sowie für Außenstehende klar und verständlich kommuniziert.

Die für uns wesentlichen Eigenschaften, auf die wir uns in der Entwicklung konzentrieren wollten, waren das User-Interface und die Demonstration der beiden Basisfunktionen, nämlich der des Scannens von Räumen und des Einfügens von virtuellen Objekten in diese folglich ebenso virtuellen Räumen.

## 1.2 Screens

Dabei haben wir sechs für das Userinterface und die Basisfunktionen wichtige Bildschirme definiert, die wir in dieser Arbeit realisieren wollten, sowie einen zusätzlichen Bildschirm für die Applikationseinstellungen. Diese sieben „Screens“ sind:

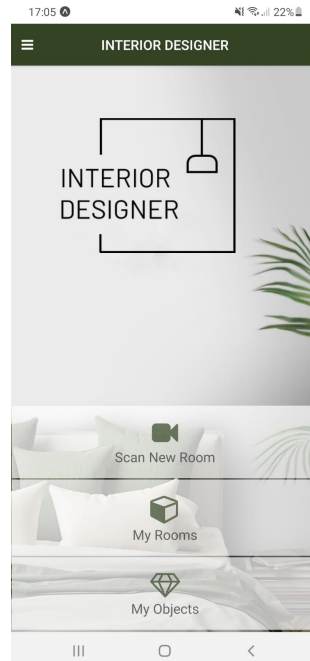
1. HomeScreen
2. RoomScanScreen
3. RoomScanConfirmScreen
4. RoomSelectScreen
5. RoomScreen
6. ObjectSelectScreen
7. SettingsScreen (**Abb. ??**)

## 1.3 Navigation

Ihre Funktionen lassen sich am besten durch typische Navigationsabfolgen beschreiben:

1. User startet die App, gelangt zum **HomeScreen** (**Abb. ??**) und findet dort 3 große Buttons:

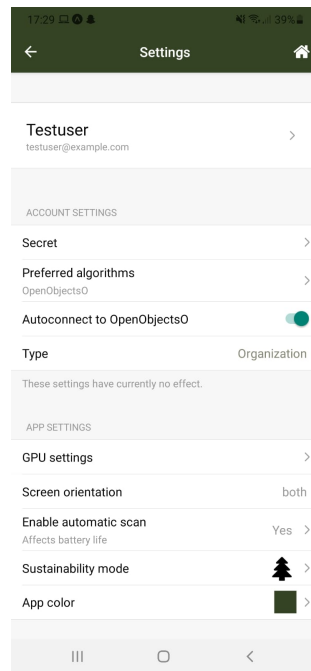
- a) Scan New Room
- b) My Rooms
- c) My Objects



**Abbildung 1**

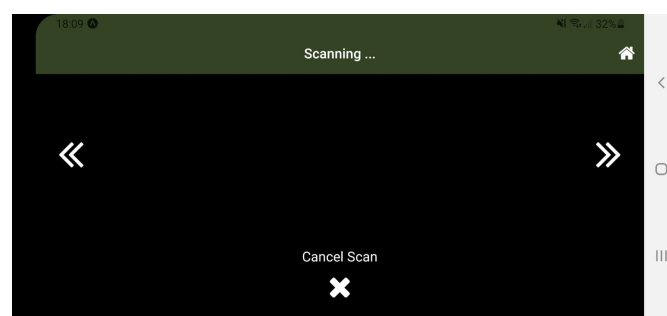
*M3 Prototype: HomeScreen*

1. User entschließt sich zum Scannen eines neuen Raumes und tippt „**Scan New Room**“ an.
1. User befindet sich jetzt im **RoomScanScreen** (Abb. ??), die Kamera des Geräts schaltet sich ein, es erscheinen visuelle Indikatoren, die den Scanprozess unterstützen sollen. Die Simulation des Scanprozesses dauert im Prototypen etwa 25 Sekunden. Anschließend wird der RoomScanScreen geschlossen.
1. User ist nun im **RoomScanConfirmScreen** (Abb. ??) angelangt, sieht eine Skizze eines Raumes mit Meter- und Quadratmeterangaben vor sich, darunter zwei Buttons, mit denen sich das Scanresultat bestätigen oder ablehnen lässt. In der fertigen Applikation führt das Bestätigen zu einem „Speichern“-Dialog, im Prototypen zurück zum HomeScreen. Das Ablehnen führt sowohl in der fertigen App wie auch im Prototypen zurück zum HomeScreen.



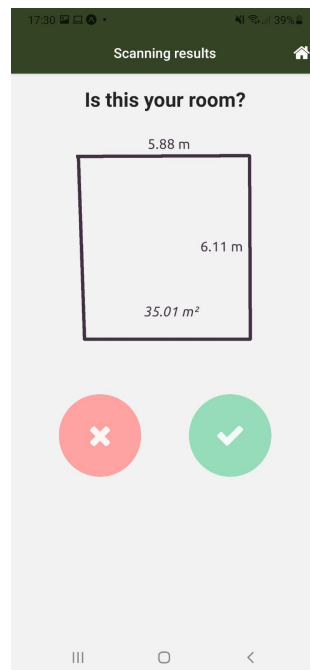
**Abbildung 2**

*M3 Prototype: SettingsScreen*



**Abbildung 3**

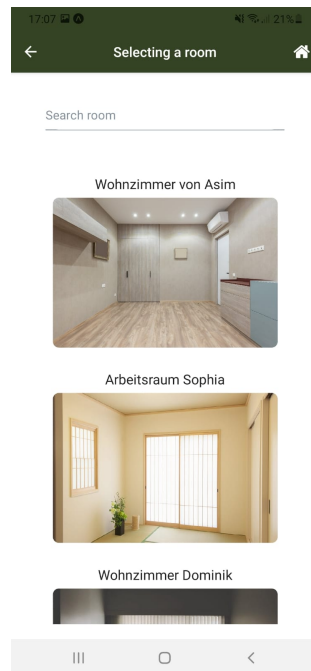
*M3 Prototype: RoomScanScreen (derzeit nur iPhone)*



**Abbildung 4**

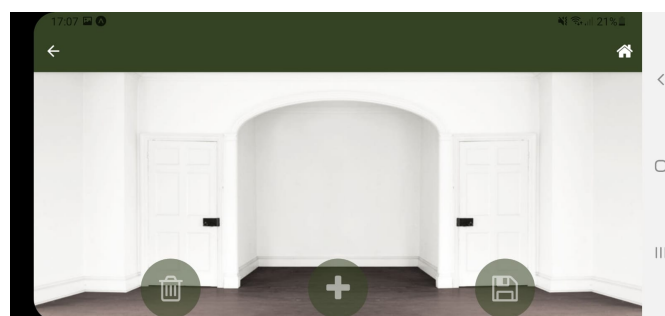
*M3 Prototype: RoomScanConfirm*

1. User befindet sich wieder im **HomeScreen** und wählt dort diesmal „**My Rooms**“
  2. Im **RoomSelectScreen** (Abb. ??) wird eine Liste an Räumen angezeigt. Diese Liste kann auch durchsucht werden. Als Hilfe werden user auch Preview-Bilder der gelisteten Räume angezeigt. Im Unterschied zum Prototypen sollen hier auch weitere Details wie Anlagedatum, letzte Bearbeitungsdatum angezeigt werden. Außerdem soll es auch möglich sein, Kategorien zu bilden und Räume zu löschen, umzubenennen oder zu duplizieren (Swipes nach links oder rechts). User kann im Prototypen einen Raum auswählen. Wird das gemacht, wechselt die App in den nächsten Bildschirm.
  3. User gelangt in den **RoomScreen** (Abb. ??) und findet sich damit im gerade ausgewählten Raum wieder. In der fertigen Applikation wird es möglich sein, sich im Raum herumzubewegen und verschiedene Perspektiven wie Grundrissperspektive einzunehmen und Objekte maßgenau manuell oder per Algorithmus anzuordnen. Im Prototypen wird das mit einem einfachen Bild des Raumes und drei runden Buttons simuliert. Ein Button dient dem Hinzufügen von Objekten, einer dem Entfernen aller Objekte und ein weiterer speichert das Projekt bzw. den Raum in dieser Anordnung ab. Im Prototypen wird das Speichern mittels eines Pop-up-Dialogs simuliert. Über den Button mit dem Plus-Symbol, der für das Hinzufügen neuer Objekte steht, gelangt man in den nächsten Screen.
- 
1. User ist im **ObjectSelectScreen** (Abb. ??). Dieser Screen ähnelt dem RoomSelectScreen. Es soll hier eine Art „Bibliothek der Objekten“ entstehen und Objekte sollen in der fertigen Applikation die zentrale Rolle einnehmen. Die Funktionalität des Prototypen beschränkt sich jedoch noch auf das Durchsuchen und Auswählen der Objekte. Hat



**Abbildung 5**

*M3 Prototype: RoomSelectScreen*



**Abbildung 6**

*M3 Prototype: RoomScreen*

user ein solches ausgewählt, führt die Navigation zurück in den **RoomScreen** (Abb. ??) und das Objekt wird mittig im Bild platziert.



**Abbildung 7**

*M3 Prototype: ObjectSelectScreen*

1. Zuletzt kann user den Raum speichern (Abb. ??) oder abrechnen. In beiden Fällen gelangt man wieder in den **HomeScreen**.

## 2 Designentscheidungen

*Autor: Dominik PEGLER*

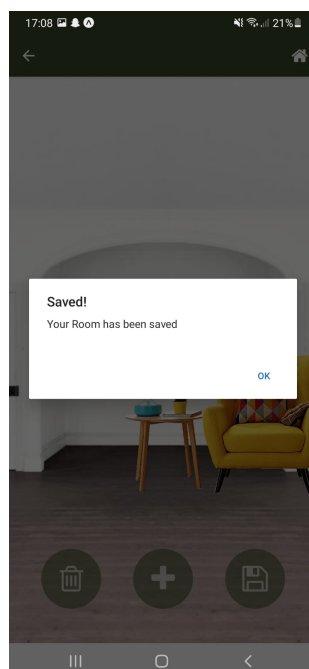
### 2.1 Personas

Wir haben darauf Wert gelegt, dass die Applikation minimal gehalten werden kann, wenig Text und mehr Icons zum Einsatz kommen können, da man von unseren primären Personas erwarten kann, dass sie ein hohes Verständnis für moderne Technologien mitbringen und ein Mehr an Hinweistexten und Navigationsassistenz für sie als störend und das UX beeinträchtigend empfunden würde. Da unsere Personas relativ dynamisch ihre Umwelt gestalten und Wohnortwechsel bei ihnen selbst sowie auch in ihrem Freundes- und Bekanntenkreis



**Abbildung 8**

*M3 Prototype: RoomScreen (Objekt hinzugefügt)*



**Abbildung 9**

*M3 Prototype: RoomScreen (Speicher)*



regelmäßig stattfinden können, gehen wir außerdem davon aus, dass unsere Applikation von ein und derselben Person häufiger benutzt wird. Vieluser würden sich durch Features, die den Einstieg in die App zwar erleichtern, aber später keinen Nutzen mehr bringen, wahrscheinlich in ihrer Zielerreichung, dem schnellen Zusammenstellen von Raumlösungen, gehindert sehen.

## 2.2 Farbwahl

Wir haben außerdem versucht, **Farben** so zu verwenden, dass sie durch Hervorhebung bestimmter Elemente Bedeutung für die Navigation durch die App haben. Daher wollten wir Farben überall dort vermeiden, wo sie keinen wirklichen Zusatznutzen bringt oder sogar der Usability im Weg steht.

Die App soll auch eine Akzentfarbe enthalten, die der App Identität verleiht. In Statusleisten, Logos und Menüs verwenden wir diese.

Probleme mit Rot-Grün-Kontrasten wollen wir vermeiden, indem wir diesen im Falle von Confirm-Denial-Situationen auch noch passende Icons zur Seite stellen. Die Farben sind hier angeführt:

- Appfarbe/Akzentfarbe: Farbe A
- Hintergrundfarbe: Weiß: #ffffff
- Hauptfarben: ca. 4 Farben
  - Farbe A: #303E14
  - Farbe B (rot / decline / cancel / danger): #ff9999
  - Farbe C (grün / accept / okay): #8cd9b3
  - Farbe D (gelb / extra features): #d9d98c

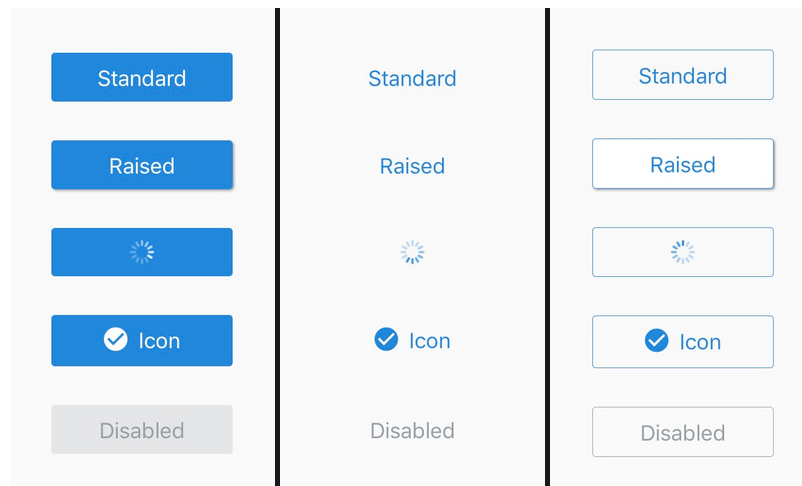
## 2.3 Konsistenz

**App-Komponenten:** konsistentes Design Components aus nur möglichst einer Library verwenden. Wo möglich haben wir Komponenten aus der Library „React Native Elements“ und Icons von „FontAwesome“ verwendet.

# 3 Projektmanagement

*Autor: Dominik PEGLER*

Es wurden auf github für das Projekt eine Organisation mit den drei Mitgliedern sowie ein Repository für die App und ein weiteres für die Projektdokumentation angelegt. Die Programmieraufgaben werden so verteilt, dass die Personen möglichst unabhängig voneinander arbeiten können. Den Zwischenstand des geschriebenen Codes (z.B. „HomeScreen.js“) laden Sophia und Asım zumindest einmal am Tag in den GoogleDrive-Folder des Projekts oder direkt in eigenen Branch auf Github. Von dort wird der Code wenn funktionstauglich von Dominik in Branch „main“ übernommen. Ziel dabei ist, dass Main-Branch immer eine funktionstaugliche App ohne Fehler oder Warnungen enthält.



**Abbildung 10**

*Beispiel-Components react-native-elements: Solid, Clear, Outline*

### 3.1 Aufgabenverteilung:

(siehe GoogleSheet)