

# Projektbeschreibung „Gebäude-Konstruktion“

## Projektbeschreibung

Das Projekt wird eine Applikation herstellen, wo man via Backend ein Gebäude konstruieren kann und diese in der eigenen internen Datenbank gespeichert wird.

Mit einem Gebäude kann man die einzelnen Etagen, Räume und die Möbel innen verwalten.

Zu dem kann man einen Besitzer einem Gebäude zuweisen.

Zu allen benötigten Daten in der Datenbank soll man die Möglichkeit haben via REST API diese zu verwalten (Hinzufügen/Aktualisieren/Entfernen) und anzuschauen.

## Personengruppen

Für diese Applikation haben wir folgende verschiedene Benutzer

Benutzerrolle:	Beschreibung:
Admin (Administrator)	Der Admin hat die Gesamtkontrolle über die Applikation und kann dementsprechend alle Funktionen anwenden.
Kunde	Kunde haben das Recht alle gespeicherten Daten ansehen zu dürfen, Ausnahmen sind Personen-Bedingte Daten (wie vom Besitzer) und sie dürfen keine Änderungen bei den Daten durchführen.
Architekt	Architekten besitzen die Rechte Daten rund um das Gebäude und die Platzierungen der Möbel verwalten zu dürfen. Aber Ihnen wird verweigert Daten zu Möbel ändern zu können. Sie haben dennoch Zugriff zum Anschauen aller Daten.
Möbelverkäufer	Möbelverkäufer besitzen einzig nur das Recht Daten rund um „Möbel“ verwalten zu dürfen und anzuschauen. Zu restlichen Daten wird diese blockiert.

## User-Stories

Für diese Applikation sollen folgende Funktionen zur Verfügung sein (diese via REST API anzusteuern sind):

Login:

- **Als** Admin/Kunde/Architekt/Möbelverkäufer **möchte ich** einloggen **damit ich** bestimmte Rechte für Funktionen zu erhalten.
- **Als** Admin/Kunde/Architekt/Möbelverkäufer **möchte ich** ausloggen **damit ich** als anderen Benutzer danach einloggen kann.

Besitzer:

- **Als** Admin **möchte ich** einen neuen Besitzer hinzufügen **damit ich** diese später einem Gebäude zuweisen kann.
- **Als** Admin **möchte ich** einen Besitzer verwalten **damit ich** diesen aktualisieren oder/und entfernen kann.

- **Als** Admin/Architekt **möchte ich** die Daten eines Besitzers aufrufen können **damit ich** diese mit der Zuweisung verwenden kann.

Gebäude:

- **Als** Admin/Architekt **möchte ich** ein neues Gebäude erstellen **damit** daraus ein Gebäude-Konstruktion erstellt werden kann.
- **Als** Admin/Architekt **möchte ich** eine neue Etage und Zimmer dem Gebäude zuweisen **damit ich** die Konstruktion verfeinern kann.
- **Als** Admin/Architekt **möchte ich** das Gebäude verwalten **damit ich** dieses Aktualisieren und/oder verwalten kann (inklusive der Etagen und Zimmer)
- **Als** Admin/Kunde/Architekt **möchte ich** alle Daten zum Gebäude mit den Etagen und Zimmer aufrufen **damit ich** die Konstruktion vom Gebäude vor mir sehe.

Möbel:

- **Als** Admin/Möbelverkäufer **möchte ich** neue Möbelstücke hinzufügen **damit ich** diese für die Gebäuden zur Verfügung stelle.
- **Als** Admin/Architekt **möchte ich** die Möbelstücke in einem bestimmten Raum zufügen **damit** es in der Konstruktion registriert wird.
- **Als** Admin/Kunde/Architekt **möchte ich** alle Möbelstücke in einem Raum sehen können **damit ich** weiss, welche Möbelstücke im Raum ist.

## Technologien

Für die Applikation werden folgende Technologien verwendet:

Name	Beschreibung
Docker	Eine Engine, das einzelne Containers mit Applikationen und Konfigurationen verwaltet und hosted.
Docker-Compose	Mit einer „.yaml“-Datei lässt sich durch die Applikation ein neues „Image“ erstellen, woraus man dann ein Container laufen lassen kann.
Spring	Ein Framework für Java-Plattformen. (Speziell geeignet für Web-Anwendungen)
Java	Die Programmiersprache, welches für das Backend geschrieben wird.
PostreSql	Die Datenbank für die Applikation. Alle Anfragen werden als Transaktionen angenommen.
Postman	Eine Applikation, die API-Abfragen verschicken kann und zum Testen der API vom Backend dient.
Github	Webseite, wo der ganze Code/Projekt gelagert wird und daraus heruntergeladen werden.