M223 Projektdokumentation

# Anforderungen analysieren

## Erweiterte Anforderungen

|  |  |
| --- | --- |
| **Funktional** | **Nicht-Funktional** |
| Als Mitglied kann ich alle Buchungen anonymisiert ansehen. | Die Datenbank wird automatisch erstellt und bei jedem Start werden allfällige Änderungen ausgeführt. |
| Als Administrator kann ich alle Buchungen ansehen (nicht anonymisiert) | Es wird ein Refresh Token erstellt, der 2 Wochen gültig ist, mit dem man neue Tokens generieren kann. |
| Als Administrator kann ich alle Buchungsanfragen ausgeben lassen (nicht anonymisiert) | Das Passwort muss in der Datenbank gehasht gespeichert werden. |

## Personas

Ein Bild, das Person, draußen, darstellend, schließen enthält.

Automatisch generierte Beschreibung

**Name**: Siras Sangapilai (Mitglied)

**Alter**: 21

**Geschlecht**: weiblich

**Beruf**: Kauffrau

**Anwendung**: Siras arbeitet in einem Kleinunternehmen, welches kein eigenes Büro besitzt. Deshalb nutzt ihre Firma das Coworking Space Angebot für ihre Mitarbeitenden.

Ein Bild, das Person, draußen, lächelnd, Gras enthält.

Automatisch generierte Beschreibung

**Name**: Sarah Müller (Mitglied)

**Alter**: 24

**Geschlecht**: weiblich

**Beruf**: HR

**Anwendung**: Sarah arbeitet in einem grossen Unternehmen, welches ein eigenes Grossraumbüro besitzt. Dies ist ihr allerdings zu laut und ungeeignet für ihren Workflow, weshalb sie sich entschieden hat, ein Coworking Space Angebot zu nutzen.

Ein Bild, das Mann, Wand, Person, drinnen enthält.

Automatisch generierte Beschreibung **Name**: André Chabulani (Admin **Alter**: 43

**Geschlecht**: männlich

**Beruf**: IT-Administrator

**Anwendung**: André arbeitet bei der Firma, die Coworking Spaces anbietet. Damit alles korrekt verläuft und die Buchungen richtig verwaltet werden, schaut er, dass keine Konflikte entstehen und unterstützt die Kunden bei Problemen.

## Anwendungsfalldiagramm

In Projektordner **/docs/charts/usecase.plantuml**

# Persistenz Schicht planen

## Fachklassendiagramm

In Projektordner **/docs/charts/usecase.plantuml**

# Schnittstellenplanung

### Schnittstellen planen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Methode** | **Endpoint** | **Security** | **Beschreibung** | **Request Payload (JSON)** | **Responses** |
| POST | /api/members/register | OPEN | Ein Gast sendet eine E-Mail, Vor- und Nachname, sowie ein Passwort um sich als neuen User zu registrieren. | E-Mail, Vorname, Nachname und Passwort | **201 – CREATED**  User erstellt  **409 – CONFLICT**  User existiert bereits (E-Mail ist Unique) |
| POST | /api/members/login | OPEN | Ein Gast meldet sich mittels E-Mail / Passwort an und erhält einen JWT-Token, sowie einen Refresh Token | E-Mail & Passwort | **200 – OK**  Login erfolgreich  **401 – UNAUTHORIZIZED**  Ungültige Anmeldedaten |
| GET | /api/bookings/{onlymy} | MITGLIED / ADMIN | Alle Buchungen ansehen (Für MITGLIED anonymisiert) oder nur eigene Buchungen ansehen (onlymy Parameter) | - | **200 – OK**  Erfolgreich (Buchungsliste)  **401 – UNAUTHORIZIZED**  Ungültiger JWT-Token |
| DELETE | /api/bookings/{bookingid} | MITGLIED / ADMIN | Storniert / löscht eine bestimmte Buchung. Mitglieder können nur eigene Buchungen löschen, Admins alle. | - | **204 – NO CONTENT**  Erfolgreich  **401 – UNAUTHORIZIZED**  Ungültiger JWT-Token  **404 – NOT FOUND**  Buchung nicht gefunden |
| POST | /api/bookings | MITGLIED / ADMIN | Erstellt eine neue Buchungsanfrage oder für einen Admin direkt eine Buchung. | Booking Objekt | **200 – OK**  Erfolgreich (Booking)  **401 – UNAUTHORIZIZED**  Ungültiger JWT-Token |
| PUT | /api/bookings/{bookingid} | ADMIN | Verändert eine bestehende Buchung. | Booking Objekt | **200 – OK**  Erfolgreich (Booking)  **401 – UNAUTHORIZIZED**  Ungültiger JWT-Token oder keine Admin Rechte  **404 – NOT FOUND**  Buchung nicht gefunden |
| GET | /api/members | ADMIN | Gibt alle registrierten Benutzer zurück. | - | **200 – OK**  Erfolgreich (Userliste)  **401 –UNAUTHORIZIZED**  Ungültiger JWT-Token oder keine Admin Rechte |
| PUT | /api/members/{memberid} | ADMIN | Bearbeitet einen bereits existierenden Member. | Member Objekt | **200 – OK**  Erfolgreich (User)  **401 – UNAUTHORIZIZED**  Ungültiger JWT-Token oder keine Admin Rechte  **404 – NOT FOUND**  Member nicht gefunden |
| DELETE | /api/members/{memberid} | ADMIN | Löscht einen bereits existierenden Benutzer | - | **204 – NO CONTENT**  Erfolgreich  **401 – UNAUTHORIZIZED**  Ungültiger JWT-Token oder keine Admin Rechte  **404 – NOT FOUND**  Member nicht gefunden |

## Sequenzdiagramm

In Projektordner **/docs/charts/sequence\_cancelbook.plantuml**