# Co-Working Space Planung

Inhalt

[Co-Working Space Planung 1](#_Toc114662049)

[Anwendungsfälle 1](#_Toc114662050)

[Funktionale Anwendungsfälle 1](#_Toc114662051)

[Nicht-funktionale Anwendungsfälle 2](#_Toc114662052)

[Akteure 3](#_Toc114662053)

[Akteur Admin 3](#_Toc114662054)

[Akteur Besucher 4](#_Toc114662055)

[Akteur Mitglied 5](#_Toc114662056)

## Anwendungsfälle

### Funktionale Anwendungsfälle

› Als Besucher möchte ich mich mit meinem Vor- und Nachnamen, meiner E-Mail-Adresse und einem Passwort registrieren, damit ich die Rolle Mitglied bekommen kann.

› Als Besucher möchte ich mich mit meiner E-Mail-Adresse und meinem Passwort anmelden, damit ich mich als Mitglied oder Administrator authentifizieren kann.

› Als Mitglied möchte ich halbe und ganze Tage an bestimmten Daten im Coworking Space als Buchung anfragen, damit ich die Angebote des Coworking Space nutzen kann.

› Als Mitglied möchte ich den Status meiner Buchungen überprüfen, damit ich erfahre, ob meine Buchung bestätigt oder abgelehnt wurde.

› Als Mitglied kann ich meine zukünftigen Buchungen stornieren, damit ich auf Veränderungen in meiner Terminplanung reagieren kann.

› Als Administrator kann ich Mitglieder verwalten (erstellen, bearbeiten, löschen), damit ich die Mitglieder organisieren kann.

› Als Administrator kann ich Buchungsanfragen akzeptieren und ablehnen, damit die Mitglieder das Angebot des Coworking Space nutzen können.

› Als Administrator kann ich Buchungen verwalten (erstellen, bearbeiten, löschen), damit ich die Buchungen organisieren kann.

› Als Mitglied kann ich einen zweiten Monitor Buchen, um produktiver Arbeiten zu können

› Als Mitglied soll ich pro Tag nur 5 Cafés können Trinken, um exzessive Kosten und Verbrauch vorbeugen zu können

› Als Mitglied soll ich in der Lage sein mein Benutzerkonto zu löschen, damit ich die Möglichkeit habe alle meine Informationen zu löschen und mich abzumelden.

### Nicht-funktionale Anwendungsfälle

› Das Datenmodell erfüllt die erste, zweite und dritte Normalform nach der relationalen Entwurfstheorie.

› Der erste Besucher bekommt nach der Registrierung die Rolle Administrator anstatt Mitglied.

› Die Authentifizierung erfolgt mittels JSON Web Token (JWT nach RFC 7519) über den HTTP Header 'Authorization'.

› Das JWT läuft 24 Stunden nach der Ausstellung ab und verliert seine Gültigkeit.

› Das JWT wird clientseitig während dessen Lebensdauer persistent aufbewahrt.

› Das Passwort des Users wird verschlüsselt und nirgends in Klartext zu finden sein.

› Das Multi-User System funktioniert ohne Probleme Transaktionell, ohne gegen einer der ACID-Punkten zu verstossen

› Die Webseite soll nicht länger als 2000 ms Laden, weil längere Ladezeiten sorgen für unzufriedene User

# Akteure

## Akteur Admin



## Akteur Besucher

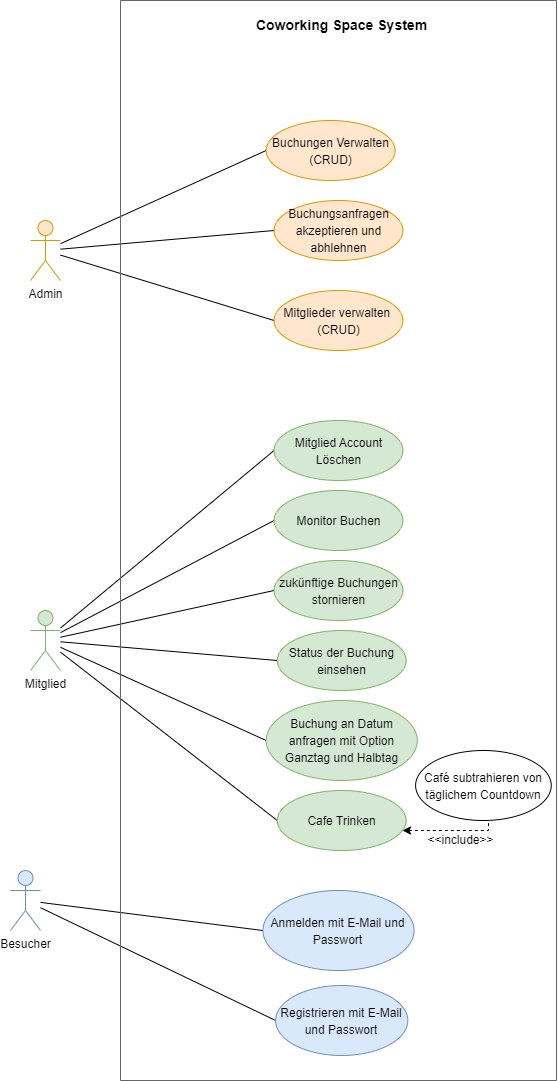


## Akteur Mitglied

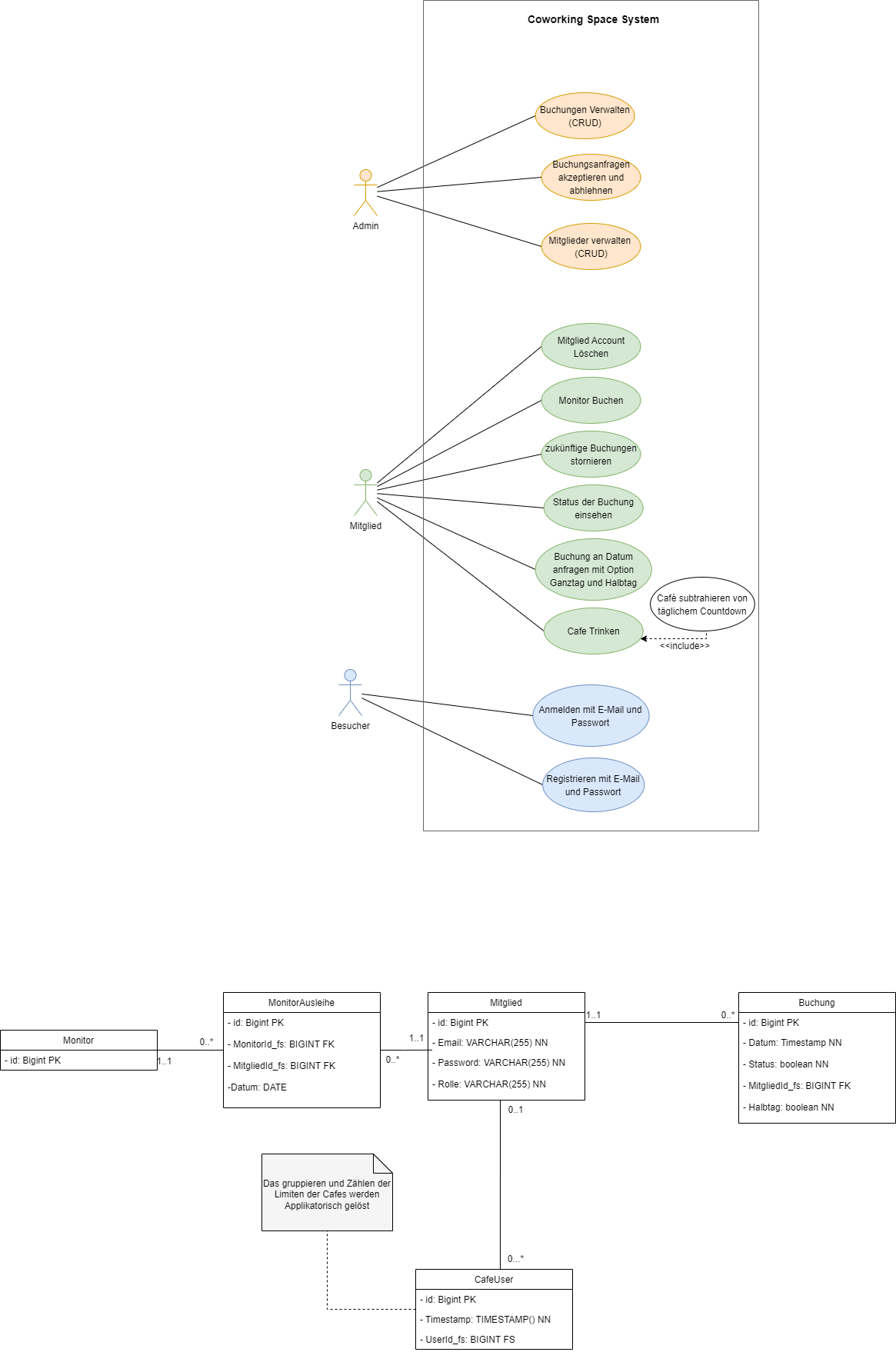
Ein Bild, das Text, Screenshot, Schild enthält.

Automatisch generierte Beschreibung

## Anwendungsfalldiagramm



## Fachklassendiagramm



## Schnittstellenplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HTTP-Methode** | **Pfad** | **Use Case** | **Parameter** | **Erfolgsfälle** | **Fehlerfälle** | **Allowed User** |
| POST | /buchung | Neue Buchung | RequestBody Buchung buchung | 201 Created: Wenn die Buchung erfolgreich erstellt wurde | 403 Forbidden  405 Method Not Allowed  408 Request Timeout  400 Bad Request | Admin  Mitglied |
| GET | /buchung | Alle Buchungen anzeigen | -- | 200 OK | 403 Forbidden  400 Bad Request | Admin |
| GET | /buchung/{id} | Buchung nach ID anzeigen | PathParameter ID id | 200 OK | 403 Forbidden  400 Bad Request | Admin  Mitglied aber nur seine eigene |
| PUT | /buchung/{id} | Buchung updaten /  Auch verwendet für Status update | RequestBody Buchung buchung, PathParameter Id id | 200 OK | 403 Forbidden  400 Bad Request | Admin |
| DELETE | /buchung/{id} | Buchung löschen | PathParameter Id id | 200 OK | 403 Forbidden  400 Bad Request | Admin  Mitglied aber nur seine eigene |
| POST | /mitglied | Neues Mitglied | RequestBody Mitglied mitglied | 201 Created: Wenn die Mitglied erfolgreich erstellt wurde | 403 Forbidden  405 Method Not Allowed  408 Request Timeout  400 Bad Request |  |
| GET | /mitglied | Alle Mitglieder anzeigen | -- | 200 OK | 403 Forbidden  400 Bad Request | Admin |
| GET | /mitglied /{id} | Mitglied nach ID anzeigen | PathParameter ID id | 200 OK | 403 Forbidden  400 Bad Request | Admin |
| PUT | /mitglied/{id} | Mitglied updaten | RequestBody Buchung buchung, PathParameter Id id | 200 OK | 403 Forbidden  400 Bad Request | Admin |
| DELETE | /mitglied/{id} | Mitglied löschen | PathParameter Id id | 200 OK | 403 Forbidden  400 Bad Request | Admin  Mitglied aber nur seine eigene |
| POST | /monitor | Neuen Monitor hinzufügen | RequestBody Monitor monitor | 201 Created | 403 Forbidden  400 Bad Request | Admin |
| POST | /monitorAusleihe | Monitor buchen | RequestBody MonitorAusleihe  monitorAusleihe | 201 Created | 403 Forbidden  400 Bad Request | Mitglied |
| POST | /CafeUser | Kafee Trinken | RequestBody CafeUser  cafeUser | 201 Created | 403 Forbidden  400 Bad Request | Mitglied |
| POST | /login | Login | RequestBody Mitglied  mitglied | 201 OK | 403 Forbidden  400 Bad Request |  |
|  |  |  |  |  |  |  |