# ZLI Projekt Analysieren

## User stories

### Funktional

Als Administrator kann ich Mitglieder zum Admin machen, damit ich erfahrene Mitglieder befördern kann.

Oder

Als Administrator kann ich neue Plätze verwalten, damit sie buchbar sind.

Als Besucher wird mir die Optionen der Buchung angezeigt, damit ich mir vorstellen kann wo es ist und wieviel Plätze es hat

Als Mitglied wird mir angezeigt an welchen Tag welche Plätze frei sind, damit ich nicht besetzte buche.

### Nicht funktional

Im JWT wird in Payload angegeben, was für einen User die Person in der Datenbank repräsentiert.

Die REST API ist in Springboot geschrieben.

Die Springboot Applikation wurden mit Integration Tests getestet via J-Unit.

## Persona

### Persona 1:

Gianluca Ferrara: Besucher

Geschlecht: M

Alter: 23

Beruf: Software Engineur in einem Startup

Gianluca möchte das Coworking Space auschecken und über das Angebot erfahren.

Wenn es ihn gefällt wird er sich registrieren wollen.



### Persona 2:

Yves Huber: Mitglied

Geschlecht: M

Alter: 19

Beruf: Software Engineur Cognizant

Yves hat das Coworking Space schon mehrmals benutzt und ist sehr zufrieden damit, er will sich nicht von seinem Mitarbeiter stören lassen und daher kommt er gerne hier her.

### Persona 3:

Alex Smolders: Admin

Geschlecht: M

Alter: 69

Beruf: Business Analysist

Alex ist ein langjähriges Mitglied von dem Coworking Space und auch schon seit 10 Jahren Admin.   
Er war immer sehr hilfsbereit und daher, wurde er zu einem Admin.  
Er war meistens alleine in Office bei seiner Firma, da alle Homeoffice gemacht haben und er wollte nicht alleine arbeiten, daher suchte er damals ein Coworking Space.

## Use-case

A picture containing diagram

Description automatically generated

## Schnittstellenplanung:

|  |  |  |  |
| --- | --- | --- | --- |
| Endpoints | Parameter | Möglicher Fehler | Beschreibung Endpoint |
| /places/{id} | /{id} | 404 HTTP Error  403 HTTP Error  401 HTTP Error | Den Platz mit der gesuchten ID finden  Der Endpoint hat alle vier CRUD Methoden |
| /places | keine | Den Endpoint “/places” gibt es nicht  404 HTTP Error  403 HTTP Error  401 HTTP Error | Liste aller Places  Der Endpoint hat alle vier CRUD Methoden |
| /users/{id} | /{id} | 404 HTTP Error  403 HTTP Error  401 HTTP Error | User mit der ID anzeigen  Der Endpoint hat alle vier CRUD Methoden |
| /users | keine | Den Endpoint “/users” gibt es nicht  404 HTTP Error  403 HTTP Error | Liste alles Users anzeigen  Der Endpoint hat alle vier CRUD Methoden |
| /bookings/{id} | /{id} | 404 HTTP Error  403 HTTP Error  401 HTTP Error | Die Buchung mit der gesuchten ID anzeigen  Der Endpoint hat alle vier CRUD Methoden |
| /bookings | keine | 404 HTTP Error  403 HTTP Error  401 HTTP Error | Liste aller Buchungen anzeigen  Der Endpoint hat alle vier CRUD Methoden |
| /login | keine | 404 HTTP Error  401 HTTP Error | Returnt ein JWT |
| /logout | keine | 404 HTTP Error | Löscht das JWT |
| /info | Keine | 500 HTTP Error | Zeigt an infos über den cowrking space |

## Sequence diagram: A picture containing text Description automatically generated

## Fachklassendiagramm:

Diagram

Description automatically generated

ERM:

Graphical user interface

Description automatically generated with medium confidence