NOSERYOUNG

# Dokumentation UserProfile

AMANDA WALSER

CEDRIC ZOLLINGER

DANNY NGUYEN

# Einleitung und Projektanforderungen

## Auftrag

Der User soll ein UserProfile für die Social Media Webseite “OurSpace” erstellen können. Dieses Profil soll an den User geknüpft sein, das bedeutet, man muss bereits ein User sein, um ein Profil erstellen zu können.

## Ausgangslage

Als Ausgangslage haben wir bereits ein TypeScript React Projekt und ein Spring Boot Projekt bekommen. Darin gab es bereits einen User, eine Homepage und ein Log-in. Darauf haben wir unser Projekt aufgebaut und wir haben die bereits vorhandene Security angepasst.

## Projektanforderungen

Dies sind die Projektanforderungen, die wir in unserem Projekt bearbeitet haben.

### Frontend

- Homepages für eingeloggte User:
    - Eine für normale User, wo sie ihr eigenes Profil anschauen, bearbeiten und löschen können
    - Eine für Admin User, sie können alle Profile anschauen, bearbeiten oder löschen
- Eine UserProfile Seite wo man ein Profil bearbeiten und anschauen kann

### Backend

- Die zugehörigen Funktionen zu den oben genannten Seiten vom Frontend
- Security
    - Die Endpoints dürfen nur von den Usern mit den korrekten Berechtigungen benutzt werden
    - Authentifizierung mithilfe eines JWT
    - Nicht authentifizierte Benutzer können sich nur registrieren oder anmelden
    - Admins dürfen Profile für sich selber managen und die Profile anderer Benutzer anschauen, bearbeiten oder löschen, jedoch nicht erstellen

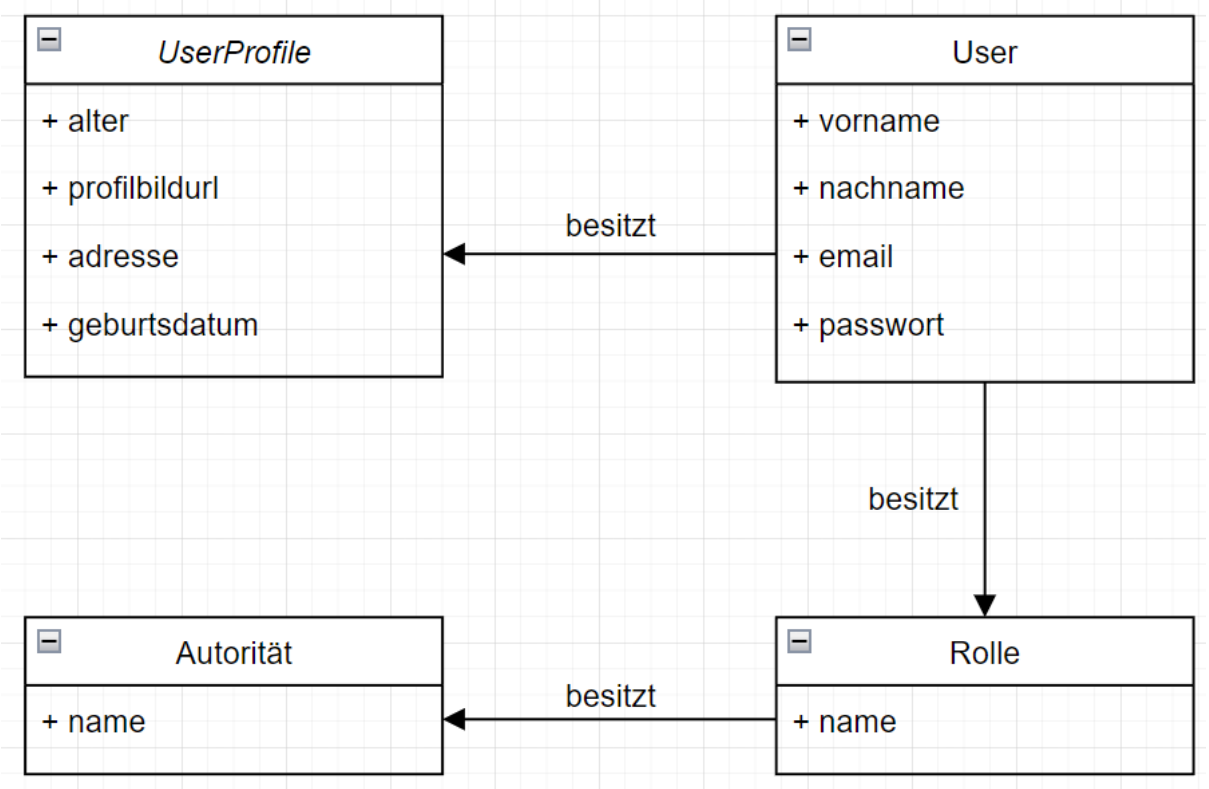
# Domain Model

Im UserProfile, welches wir implementieren, sollen folgende Werte sein:

- Alter
- Adresse
- Geburtsdatum
- Profilbild URL

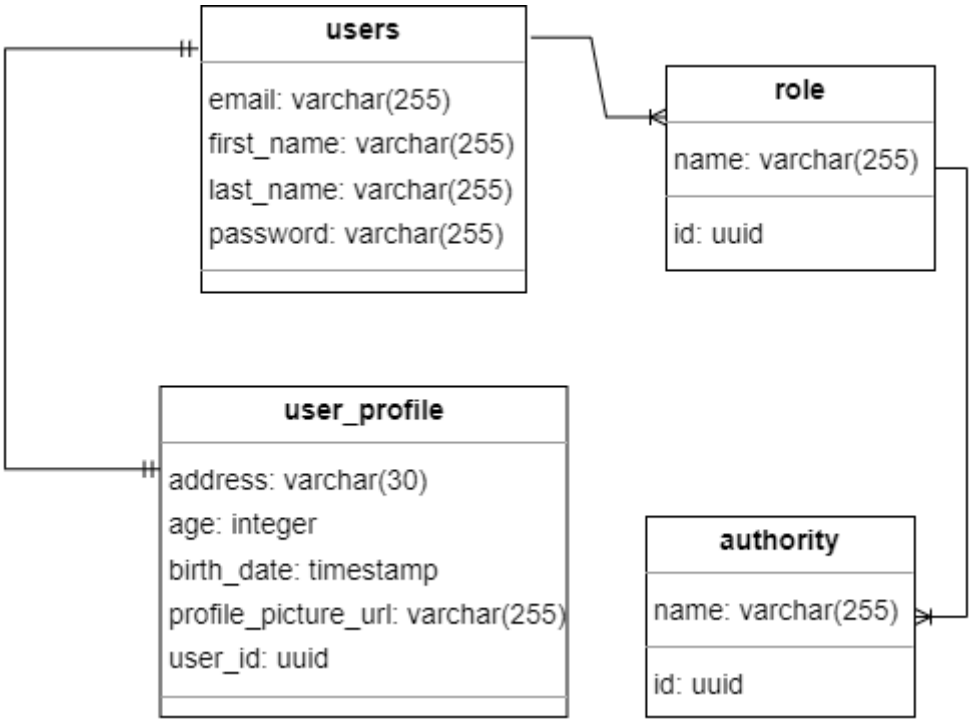
Dieses Profil ist optional für den Benutzer. Der Benutzer hat zudem Rollen, welche Autoritäten besitzen. Denn ein Benutzer kann ein einfacher Benutzer sein oder ein Admin. Der Admin hat mehr Bearbeitungsrechte der Profile als die Benutzer. Das bedeutet, ein Admin kann auch die Profile anderer Nutzer anschauen, bearbeiten und löschen. Wobei ein Benutzer nur sein eigenes Profil erstellen, bearbeiten, anschauen und löschen kann.

Da ein UserProfile einem Benutzer angehört, kann es nur von einem eingeloggten User, der bereits die Werte Vorname, Nachname, E-Mail und Passwort hat, erstellt werden.



# Entity Relationship Diagram

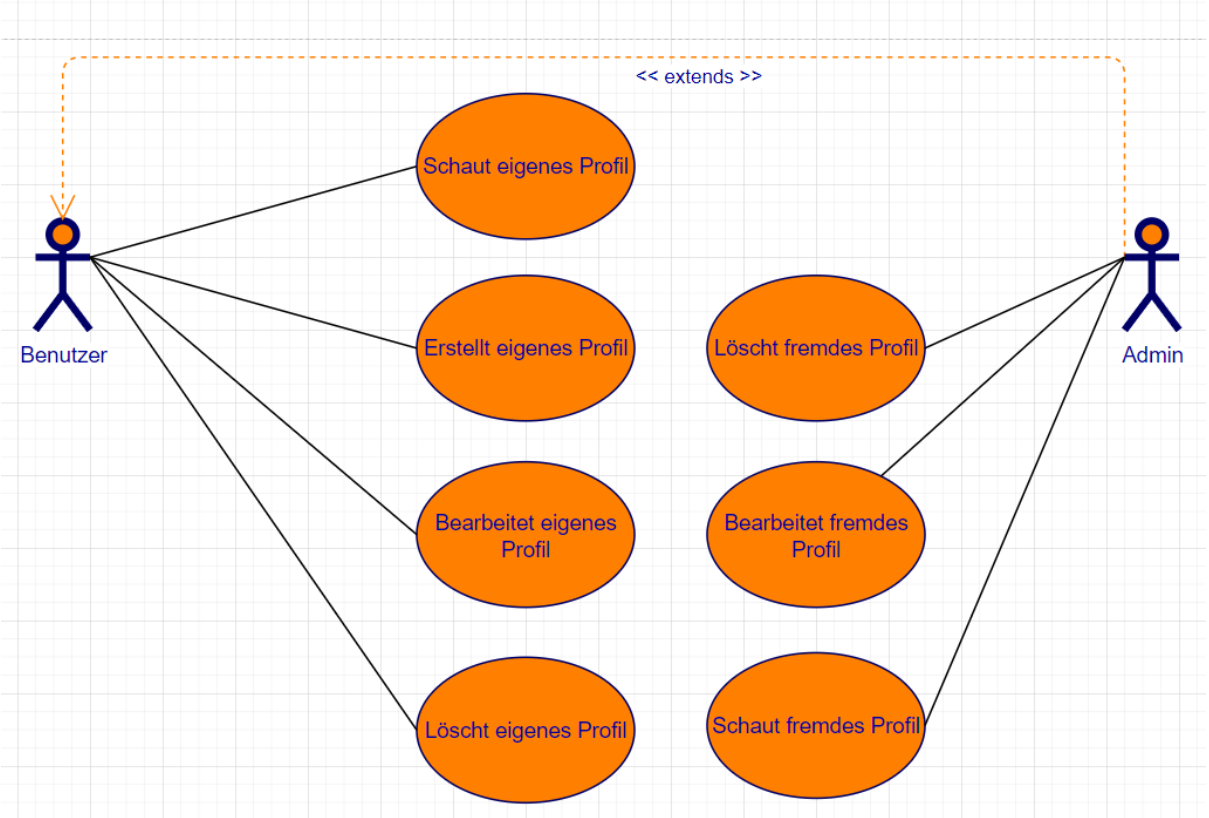
In unserem ERD kann man sehen, dass das UserProfile - welches wir erstellen - an einen User gekoppelt ist. Zudem sieht man, dass wir User haben, welche dann Rollen zugewiesen bekommen, in unserem Falle sind das Admin und User. Der Admin hat mehr Berechtigungen im Bezug auf das Bearbeiten der Profile anderer User.



# Use-Case-Diagram

Im Projekt gibt es zwei Rollen, die ein Benutzer kann. Zum einen den Benutzer - dies ist die Default Rolle. Ein Benutzer darf ein eigenes Profil erstellen, dieses bearbeiten, löschen und natürlich auch anschauen.

Die zweite Rolle ist die Rolle des Admins. Dieser erweitert den Benutzer, wodurch er all seine Funktionen für sich selbst auch anwenden kann. Ausserdem hat er das Recht, die Profile anderer Benutzer anzuschauen, zu bearbeiten und zu löschen.



# Testing Strategie

### Frontend

In unserem Projekt gibt es End-to-End Testing mit Cypress. Wir testen also das Verhalten unseres Programms bei der Durchführung eines Prozesses durch einen simulierten Benutzer. Welche Prozesse genau geprüft werden, ist unten in der Use-Case Beschreibung ersichtlich.

### Backend

Das Backend in unserem Projekt wird durch Unit- und Component Tests getestet. Dabei haben wir Unit Tests für den Grossteil der Funktionen. Das Backend wird auch im Zusammenhang mit den End-to-End Tests des Frontends getestet. Da die Lokalisierung von Problemen im Backend mit Cypress eher schwieriger ist, verlassen wir uns beim Backend mehr auf die Unit- und Component Tests.

# Use-Case Beschreibung

In der Use-Case Beschreibung sind die Funktionen von unserem Teil der Applikation beschrieben, also alles was man mit dem UserProfile machen kann. Diese werden von uns auch getestet.

## Benutzer

### Benutzer besucht eigenes Profil

| | |
|---|---|
| Aktor(en): | Benutzer |
| Beschreibung: | Benutzer besucht eigenes Profil |
| Voraussetzungen: | Benutzer:<br>● hat ein Profil<br>● ist authentifiziert |
| Nachbedingungen: | Benutzer:<br>● sah sein Profil<br>● navigiert weg |
| Normaler Ablauf: | Benutzer:<br>● navigiert zu seinem Profil<br>● sieht dort die Daten zu: Alter, Adresse, Profilbild-URL und Geburtsdatum |
| Alternativer Ablauf: | 1. Der User hat noch kein Profil und kann deswegen auch keines anschauen |
| Ausnahmen: | keine |

### Benutzer erstellt eigenes Profil

| | |
|---|---|
| Aktor(en): | Benutzer |
| Beschreibung: | Benutzer erstellt eigenes Profil |
| Voraussetzungen: | Benutzer:<br>● hat kein Profil<br>● ist authentifiziert |
| Nachbedingungen: | Benutzer:<br>● konnte sein Profil erstellen<br>● hat Daten korrekt eingegeben |
| Normaler Ablauf: | Benutzer:<br>● navigiert zum Erstellformular für ein UserProfile<br>● gibt korrekte Daten für Alter, Adresse, Profilbild-URL und Geburtsdatum ein<br>● speichert Daten |
| Alternativer Ablauf: | 1. Wenn falsche oder unvollständige Daten eingegeben werden, wird kein Profil erstellt<br>2. Wenn das Erstellen des Profils abgebrochen wird, wird kein Profil erstellt<br>3. Der User hat bereits ein Profil und es kann keines mehr erstellt werden |
| Ausnahmen: | keine |

### Benutzer bearbeitet eigenes Profil

| | |
|---|---|
| Aktor(en): | Benutzer |
| Beschreibung: | Benutzer bearbeitet Profil |
| Voraussetzungen: | Benutzer:<br>● hat ein Profil<br>● ist authentifiziert |
| Nachbedingungen: | Benutzer:<br>● konnte sein Profil bearbeiten<br>● hat Daten korrekt eingegeben |
| Normaler Ablauf: | Benutzer:<br>● navigiert zu seinem Profil<br>● bearbeitet eine oder mehrere der folgenden Daten: Alter, Adresse, Profilbild-URL, Geburtsdatum mit korrekten Werten<br>● speichert Daten |
| Alternativer Ablauf: | 1. Wenn falsche oder keine Daten |

| | |
|---|---|
| | eingegeben werden, den Initialzustand des Profils wiederherstellen<br>2. Wenn das Bearbeiten abgebrochen wird, Initialzustand wiederherstellen |
| Ausnahmen: | keine |

### Benutzer löscht eigenes Profil

| | |
|---|---|
| Aktor(en): | Benutzer |
| Beschreibung: | Benutzer löscht eigenes Profil |
| Voraussetzungen: | Benutzer:<br>● hat ein Profil<br>● ist authentifiziert |
| Nachbedingungen: | Benutzer:<br>● konnte sein Profil löschen<br>● Profil existiert nicht mehr<br>● landet auf Registrierung/Log-in |
| Normaler Ablauf: | Benutzer:<br>● navigiert zur Homepage für eingeloggte User<br>● löscht Profil (und die Daten: Alter, Adresse, Profilbild-URL und Geburtsdatum werden mitgelöscht) |
| Alternativer Ablauf: | 1. Wenn das Löschen abgebrochen wird, den Initialzustand des Profils wiederherstellen |
| Ausnahmen: | keine |

## Admin

### Admin besucht Profil(e)

| | |
|---|---|
| Aktor(en): | Admin |
| Beschreibung: | Admin besucht Profil(e) |
| Voraussetzungen: | ● ist authentifiziert und autorisiert |
| Nachbedingungen: | ● sah Profil(e)<br>● navigiert weg |
| Normaler Ablauf: | Admin:<br>● navigiert zu den Profilen<br>● navigiert zu einem einzelnen Profil<br>● sieht Alter, Adresse, Profilbild-URL und Geburtsdatum des Users oder der User, wenn er mehrere Profile |

| | |
|---|---|
| | ansieht |
| Alternativer Ablauf: | 1. Wenn keine Profile vorhanden sind, wird eine Benachrichtigung darüber angezeigt |
| Ausnahmen: | ● Wenn nicht autorisiert, wird zum Login weitergeleitet |

### Admin bearbeitet Profil(e)

| | |
|---|---|
| Aktor(en): | Admin |
| Beschreibung: | Admin bearbeitet Profil(e) |
| Voraussetzungen: | ● ist authentifiziert und autorisiert<br>● es existiert/existieren Profil(e) |
| Nachbedingungen: | ● bearbeitete Profil(e)<br>● navigiert weg |
| Normaler Ablauf: | Admin:<br>● navigiert zu den Profilen<br>● navigiert zu einem einzelnen Profil<br>● bearbeitet eine oder mehrere der folgenden Daten: Alter, Adresse, Profilbild-URL, Geburtsdatum mit korrekten Werten und dies bei einem oder mehreren Konten<br>● speichert Daten |
| Alternativer Ablauf: | 1. Wenn falsche oder keine Daten eingegeben werden, den Initialzustand des Profils wiederherstellen<br>2. Wenn Ablauf abgebrochen wird, Initialzustand wiederherstellen |
| Ausnahmen: | keine |

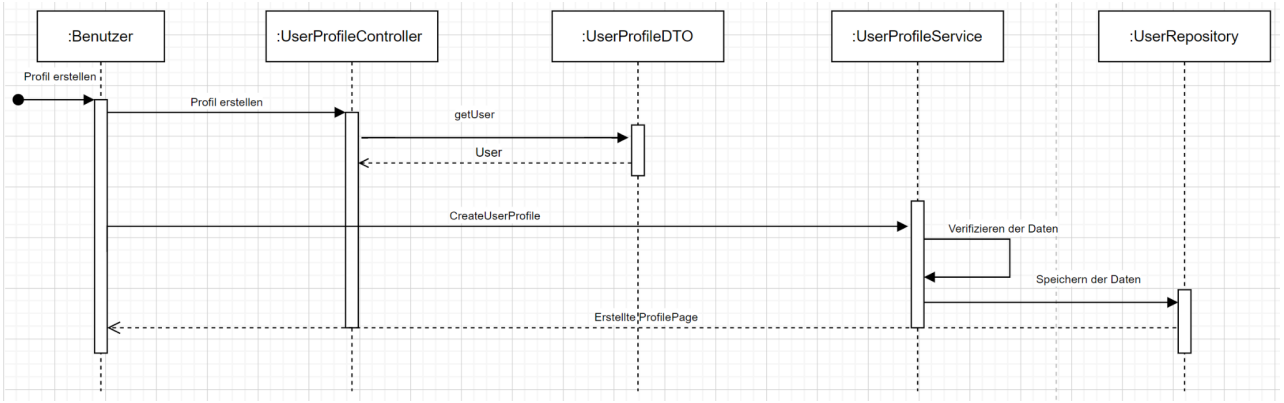
### Admin löschte Profil(e)

| | |
|---|---|
| Aktor(en): | Admin |
| Beschreibung: | Admin löscht Profil(e) |
| Voraussetzungen: | ● ist authentifiziert und autorisiert<br>● es existiert/existieren Profil(e) |
| Nachbedingungen: | ● Benutzer existiert noch<br>● Profil(e) existiert/existieren nicht mehr |
| Normaler Ablauf: | Admin:<br>● navigiert zu den Profilen |

| | |
|---|---|
| | • navigiert zu einem einzelnen Profil<br>• löscht Profil(e) und die Daten: Alter, Adresse, Profilbild-URL und Geburtsdatum werden mitgelöscht<br>• wird auf Übersicht weitergeleitet |
| Alternativer Ablauf: | 1. Wenn Ablauf abgebrochen wird, Initialzustand wiederherstellen |
| Ausnahmen: | keine |

## Sequenz Diagramm

Dieses Sequenz Diagramm stellt den Create-Prozess eines UserProfiles aus Sicht eines Users dar. Dabei gehen wir davon aus, dass der User bereits eingeloggt und somit authentifiziert ist.




## Swagger

Unsere Endpoints haben wir mithilfe von Swagger dokumentiert:
http://localhost:8080/myapi/swagger-ui.html