

OurSpace Dokumentation

Thema: ÜK 223, Group 6

Dokumentinformationen

Dateiname: ourspace-documentation.docx
Speicherdatum: 19.09.2025

Autoreninformationen

Autors: Jevgenia Shylovich
Luka Grujic

Inhaltsverzeichnis

Abbildungsverzeichnis	2
Tabellenverzeichnis	2
1 Einleitung	3
1.1 Sinn und Zweck	3
2 Diagramme	4
2.1 Domain Diagram	4
2.2 ERD	4
2.3 Sequence Diagramm	5
3 Use Cases	5
3.1 Definition	5
3.2 Diagram	7
4 Testing	8
4.1 Getestete Endpoints	8

Abbildungsverzeichnis

Abb. 1 Domain Diagram	4
Abb. 2 ERD	4
Abb. 3 Sequence Diagramm	5
Abb. 4 Use Case Diagram	7

Tabellenverzeichnis

Tabelle 1 UC1	5
Tabelle 2 UC2 User	6
Tabelle 3 UC2 Admin	6
Tabelle 4 UC3	7
Tabelle 5 Endpoint Testing	9

1 Einleitung

1.1 Sinn und Zweck

Das vorliegende Dokument beschreibt die Planung unserer OurSpace-Applikation. Es enthält eine grosse Anzahl an UML-Diagrammen und Tabellen welche das Verhalten, die Struktur und die Abläufe der verschiedenen Komponenten beschreibt.

2 Diagramme

2.1 Domain Diagram

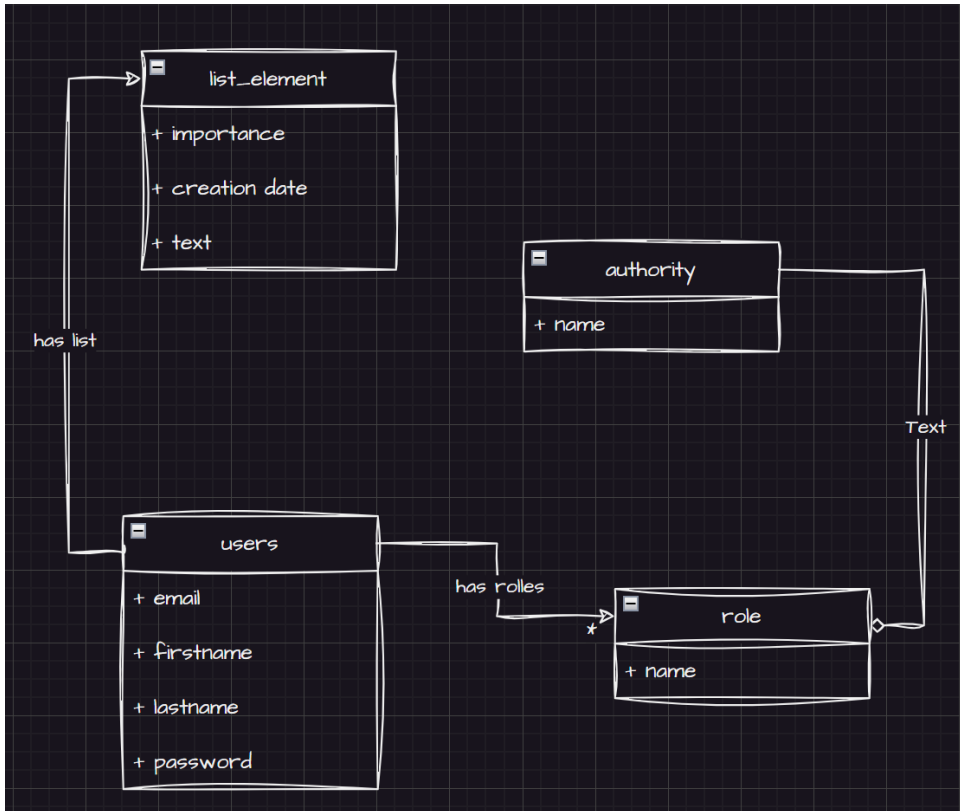


Abb. 1 Domain Diagram

2.2 ERD

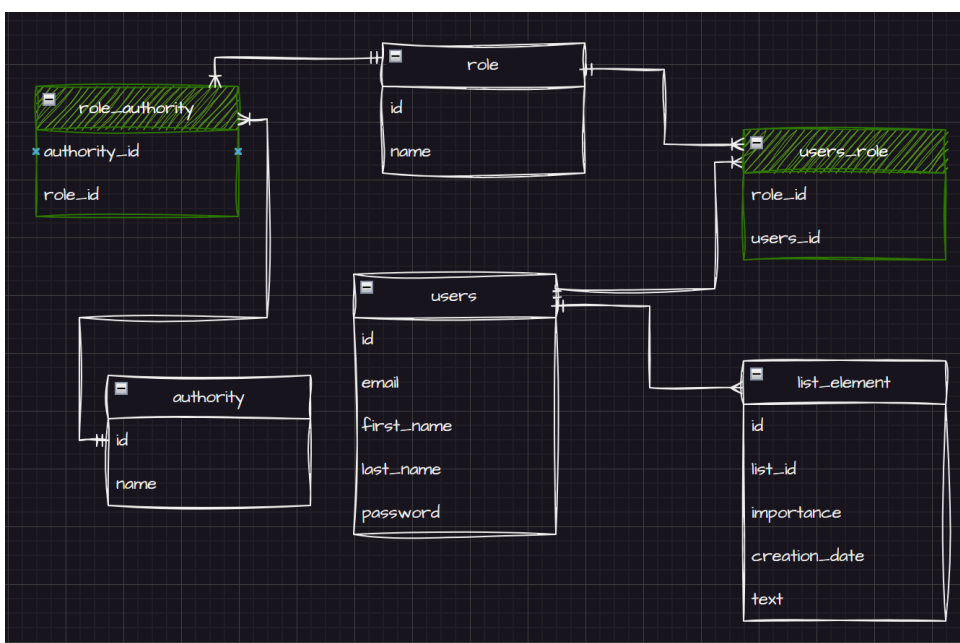


Abb. 2 ERD

2.3 Sequence Diagramm

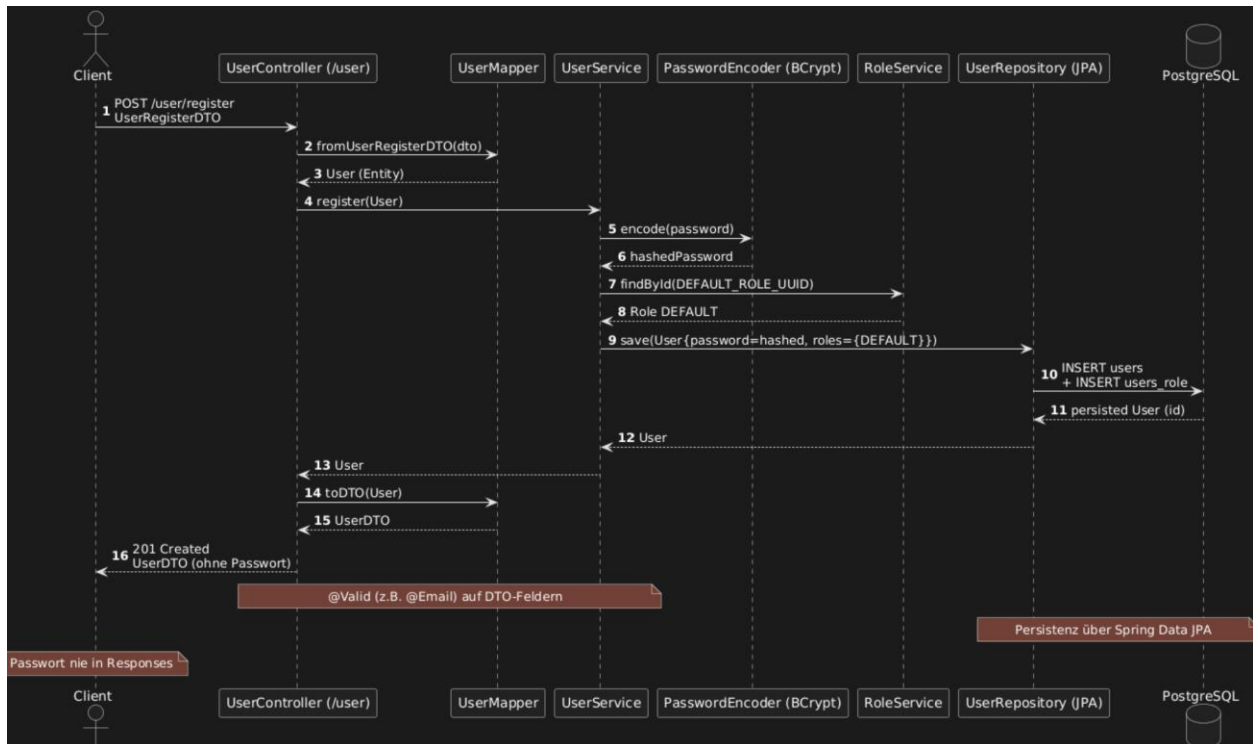


Abb. 3 Sequence Diagramm

3 Use Cases

3.1 Definition

3.1.1 UC1

Actor(s):	User
Description:	User erstellt einen Listeneintrag.
Preconditions:	<ul style="list-style-type: none"> Der User hat sich eingeloggt Rechte wurden überprüft
Postconditions:	<ul style="list-style-type: none"> Daten werden gespeichert User navigiert weg oder bleibt
Normal Course:	<ul style="list-style-type: none"> User Navigiert zu seiner Page User klickt auf Erstell-Button User gibt Daten ein und wählt Kategorie Daten werden validiert User klickt auf Submit-Button
Alternative Courses:	4f. Falls Validierung fehlschlägt, zeig Error Message. Zwing dich etwas reinzuschreiben
Exceptions:	Keine

Tabelle 1 UC1

3.1.2 UC2

Actor(s):	User
Description:	User bearbeitet oder löscht seine eigenen Listeneintrag.
Preconditions:	<ul style="list-style-type: none"> • Der User hat sich eingeloggt • Rechte wurden überprüft • User hat Listeneinträge
Postconditions:	<ul style="list-style-type: none"> • Daten werden umgeschrieben/gelöscht • User navigiert weg oder bleibt
Normal Course:	<ul style="list-style-type: none"> • User Navigiert zu seiner Page • User klickt auf einen Listeneintrag • User bearbeitet Listeneintrag • User kann die Bearbeitung speichern, indem er auf den submit button drückt oder löschen, indem er auf den lösche button drückt • Falls Daten bearbeitet wurden, werden sie validiert
Alternative Courses:	5f. Falls Validierung fehlschlägt, weil test leer gelassen wurde, zeig Error Message. Zwing dich etwas reinzuschreiben
Exceptions:	Keine

Tabelle 2 UC2 User

Actor(s):	Admin
Description:	Administrator bearbeitet oder löscht einen Listeneintrag.
Preconditions:	<ul style="list-style-type: none"> • Der Admin hat sich eingeloggt • Rechte wurden überprüft • Users haben Listeneinträge
Postconditions:	<ul style="list-style-type: none"> • Daten werden umgeschrieben/gelöscht • Admin navigiert weg oder bleibt
Normal Course:	<ul style="list-style-type: none"> • Admin Navigiert zu einer User Liste • Admin klickt auf einen Listeneintrag • Admin bearbeitet Listeneintrag • Admin kann die Bearbeitung speichern, indem er auf den submit button drückt oder löschen, indem er auf den lösche button drückt • Falls Daten bearbeitet wurden, werden sie validiert
Alternative Courses:	5f. Falls Validierung fehlschlägt, weil test leer gelassen wurde, zeig Error Message. Zwing dich etwas reinzuschreiben
Exceptions:	Keine

Tabelle 3 UC2 Admin

3.1.3 UC3

Actor(s):	User
Description:	Eingeloggte User sehen Listeneinträge eines anderen Users.
Preconditions:	<ul style="list-style-type: none"> • User loggt sich ein • User navigiert zur Listenübersicht
Postconditions:	<ul style="list-style-type: none"> • User kann raus oder weg navigieren
Normal Course:	<ul style="list-style-type: none"> • User werden alle listen angezeigt • User kann auf Listen draufklicken, um dann die listen Einträge zu sehen • User navigiert raus wieder
Alternative Courses:	Keine
Exceptions:	Keine

Tabelle 4 UC3

3.2 Diagram

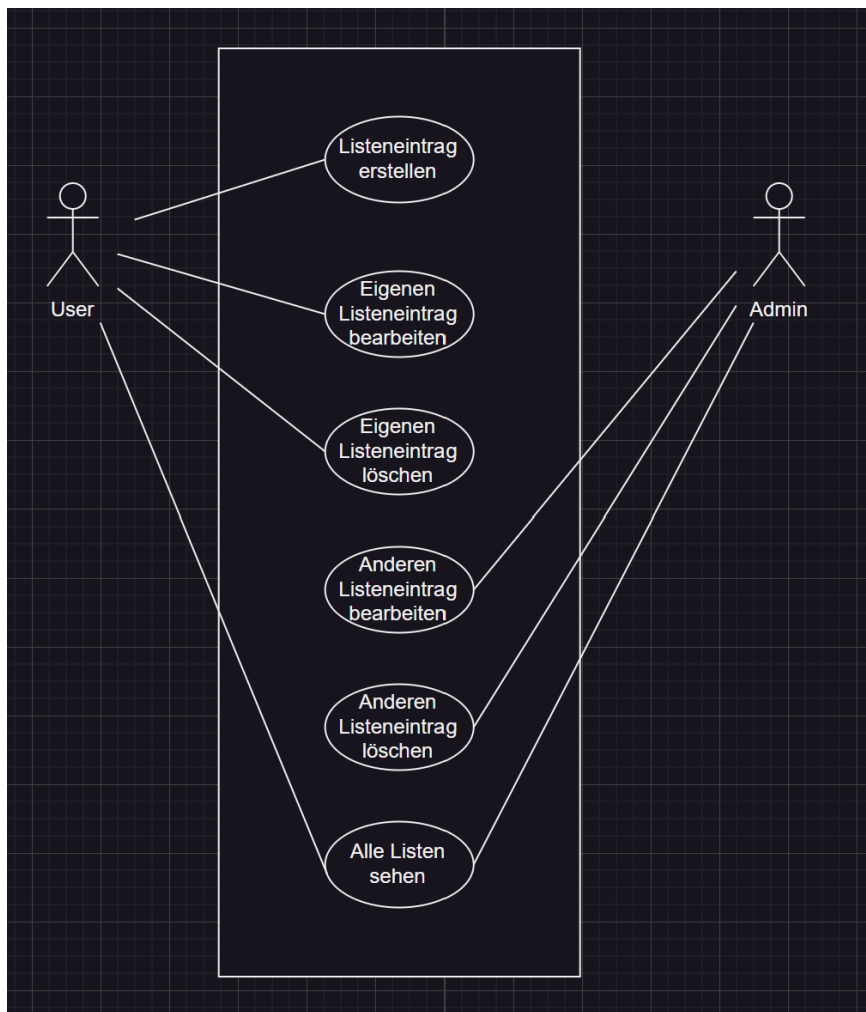


Abb. 4 Use Case Diagram

4 Testing

4.1 Getestete Endpoints

Endpoint	Was wird getestet	Erwartetes Ergebnis	Technologie
[Post] /list-element/	Wenn man als User eingeloggt ist, kann man über die eigene ProfileListPage einen neuen Listeneintrag erstellen. Dafür klickt man auf "Add List Element" Button, wobei ein Form aufgeht. Als Test gibt man beim Titel und Inhalt keine Daten ein, wobei der Submit-Button disabled ist, weil die Validierung fehlschlägt. Dazu bekommt man als User auch direkt eine Yup-Fehlermeldung. Dann gibt man alle korrekten Daten ein und erstellt den Listeneintrag über den "Submit"-Button welcher nun available ist, da die Validierung stimmt.	Wenn alle Daten korrekt sind, wird ein neuer Listeneintrag erstellt.	Cypress
[Post] /list-element/	Wenn man als Admin eingeloggt ist, kann man keine neuen Listeneinträge erstellen. Man versucht dabei auf die ProfileListPage eines Users zu gehen und dort den "Add"-Button zu betätigen. Dieser existiert für Admins nicht.	Man kann den Button nicht betätigen, da er hier nicht existiert.	Cypress
[Post] /list-element/	Wenn man nicht eingeloggt ist, kann man keine Listeneinträge erstellen. Dabei versucht man auf eine ProfileListPage zu gehen, wird aber direkt auf die Login Seite weitergeleitet.	Man wird direkt auf die Login-Page weitergeleitet.	Cypress
[Get] /list-element/{id}	Wenn man als User eingeloggt ist, kann man irgendein Listenelement anschauen.	Als Response Body bekommt man alle Listeneinträge des Users.	Postman
[Get] /list-element/{id}	Wenn man als Admin eingeloggt ist, kann man irgendein Listenelement anschauen.	Man bekommt einen Status Code 403 zurück.	Postman
[Get] /list-element/{id}	Wenn man nicht eingeloggt ist, kann man kein Listenelement anschauen.	Listenelement wird nicht geschickt. Der in der Respons Body zeigt die Error Message an. Man kriegt einen 200 Status Code zurück	Postman
[Delete] /list-element/{id}	Wenn man als Admin eingeloggt ist, kann man Listeneinträge von egal welchen Users löschen.	Listeneintrag eines beliebigen Users wird gelöscht. Man bekommt einen 200 Status Code	Postman

		zurück. Der Response Body ist leer	
[Delete] /list-element/{id}	Wenn man als User eingeloggt ist, kann man seine eigenen Listeneinträge löschen	Eigener Listeneintrag wird gelöscht. Man erhält einen 200 Status Code zurück. Der Response Body ist leer	Postman
[Delete] /list-element/{id}	Wenn man als User eingeloggt ist, kann man die Listeneinträge von anderen Usern nicht löschen.	Listeneintrag wird nicht gelöscht. Man erhält den Status Code 403, Forbidden.	Postman
[Delete] /list-element/{id}	Wenn man nicht eingeloggt ist, kann man keine Listeneinträge löschen.	Listeneintrag wird nicht gelöscht. Man bekommt den Status Code 403, Forbidden.	Postman
[Put] /list-element/{id}	Wenn man als User eingeloggt ist, kann man einen seiner eigenen Listenelemente bearbeiten.	Listeneintrag wird geupdated. Man bekommt den Status Code 200.	Postman
[Put] /list-element/{id}	Wenn man als User eingeloggt ist, kann man nicht anderen deren Listenelemente bearbeiten.	Listeneintrag wird nicht geupdated. Man bekommt den Status Code 403. In der Requestbody steht die Errormessage	Postman
[Put] /list-element/{id}	Wenn man als Admin eingeloggt ist, kann man egal alle existierende Listenelemente bearbeiten.	Listeneintrag wird geupdated. Man bekommt den Status Code 200.	Postman
[Put] /list-element/{id}	Wenn man nicht eingeloggt ist, kann man keine Listeneinträge bearbeiten.	Listeneintrag wird nicht geupdated. Man bekommt den Status Code 403. In der Requestbody steht die Errormessage	Postman

Tabelle 5 Endpoint Testing