

uek225doku-Gruppe5

Thema: üK 223, Group 6

Dokumentinformationen

Dateiname: uek225doku-Gruppe5.docx
Speicherdatum: 19.09.2025

Autoreninformationen

Autors: Diego G. De Corso
Inaam Ahmed

Inhaltsverzeichnis

Abbildungsverzeichnis	Fehler! Textmarke nicht definiert.
Tabellenverzeichnis	Fehler! Textmarke nicht definiert.
1 Einleitung	3
1.1 Sinn und Zweck	3
1.2 Referenzdokumente.....	Fehler! Textmarke nicht definiert.
2 Diagramme	4
2.1 Domain Diagram	4
2.2 ERD	4
2.3 Sequence Diagramm	5
3 Use Cases	6
3.1 Definition Use-Cases	6
4 Testing	9
4.1 Vorgehensweise	9
4.2 Getestete Endpoints	10

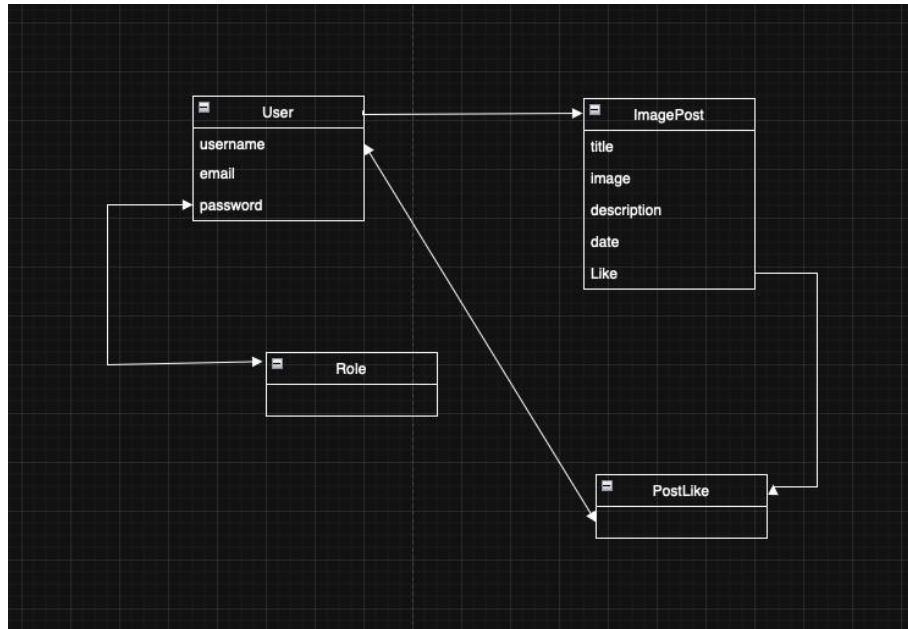
1 Einleitung

1.1 Sinn und Zweck

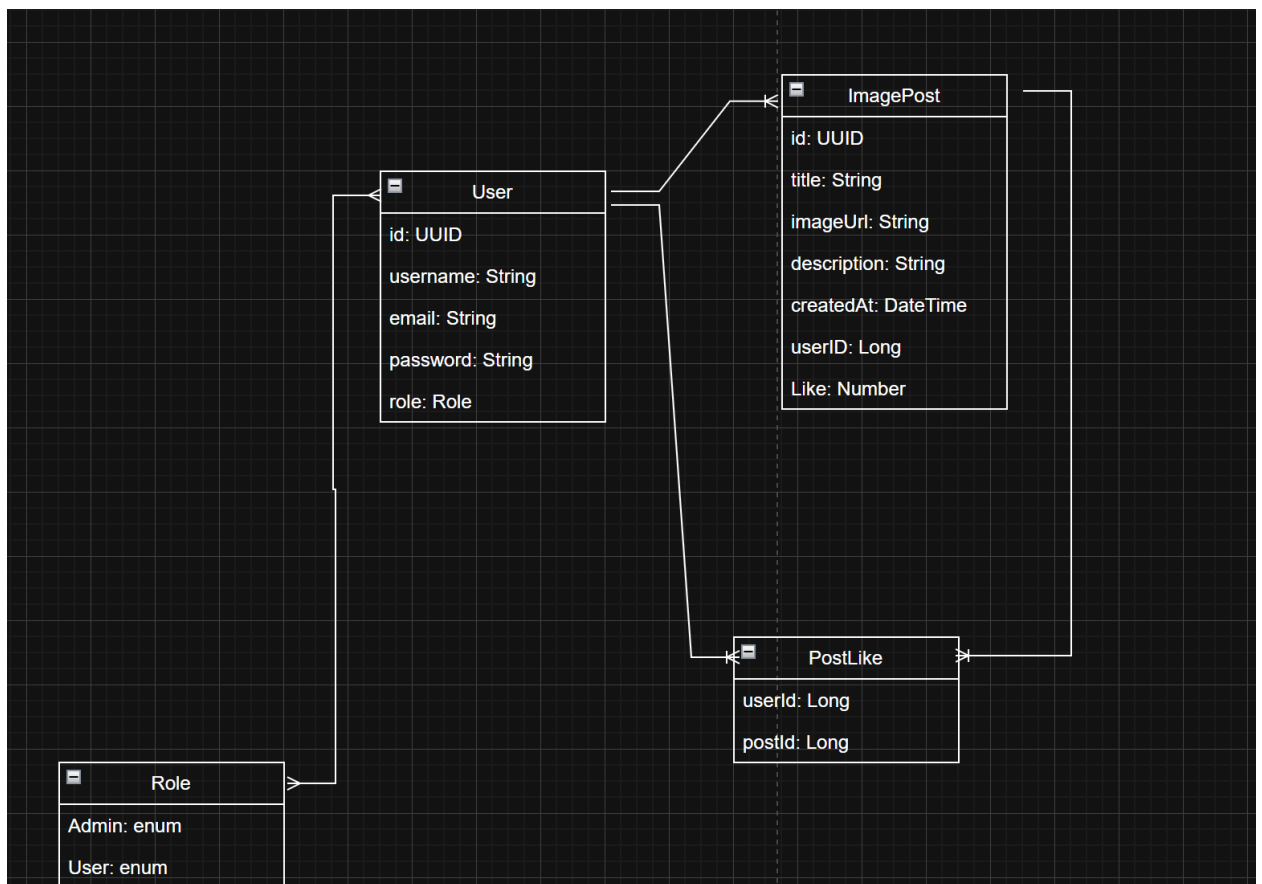
Das vorliegende Dokument beschreibt die im Rahmen des üK 223 entwickelte Full-Stack-Komponente *Image Gallery*. Es soll einen Überblick über Zielsetzung, Anforderungen, Diagramme, Use-Cases sowie die Teststrategie geben. Damit dient es als Nachweis für die korrekte Umsetzung der Projektaufgaben und ermöglicht Kursleitenden sowie Projektmitgliedern die Nachvollziehbarkeit von Mehrbenutzerfähigkeit, Sicherheit und Rechteverwaltung.

2 Diagramme

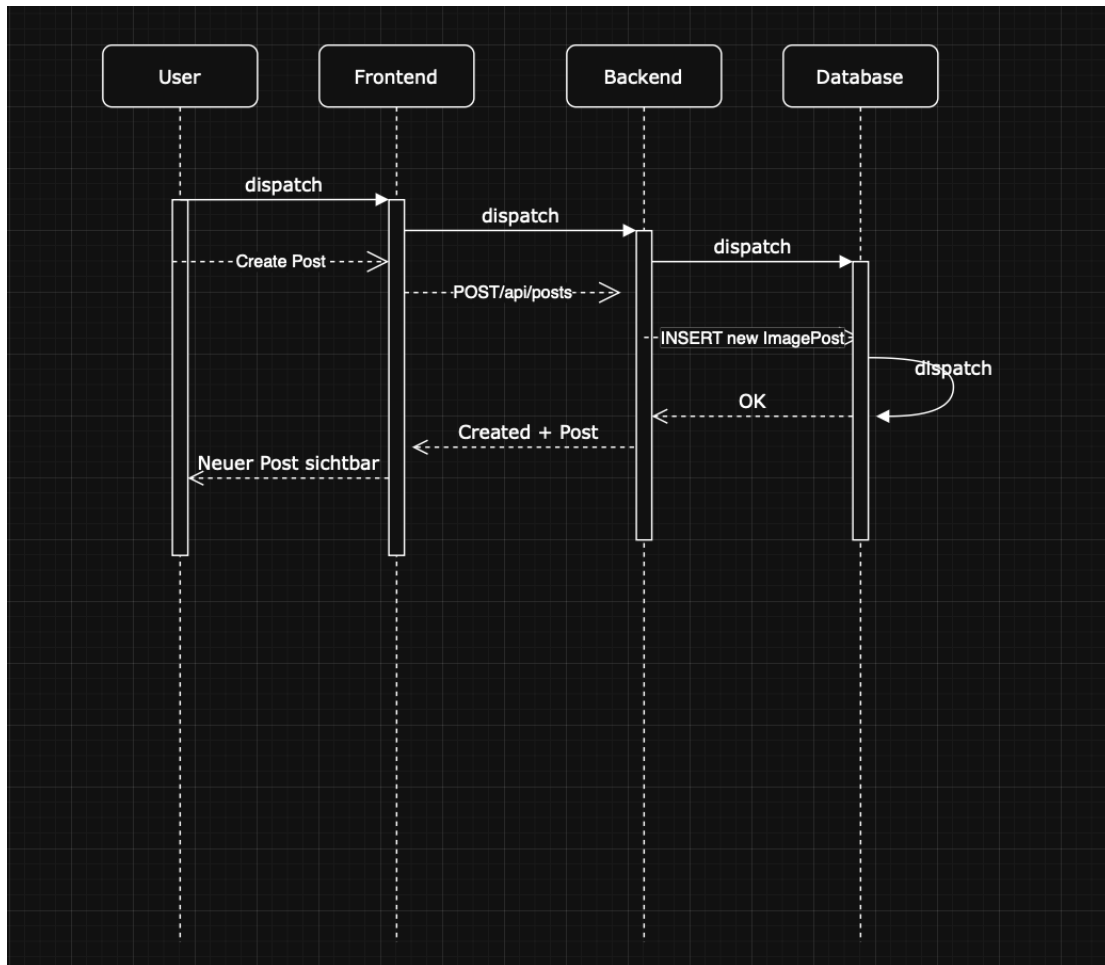
2.1 Domain Diagram



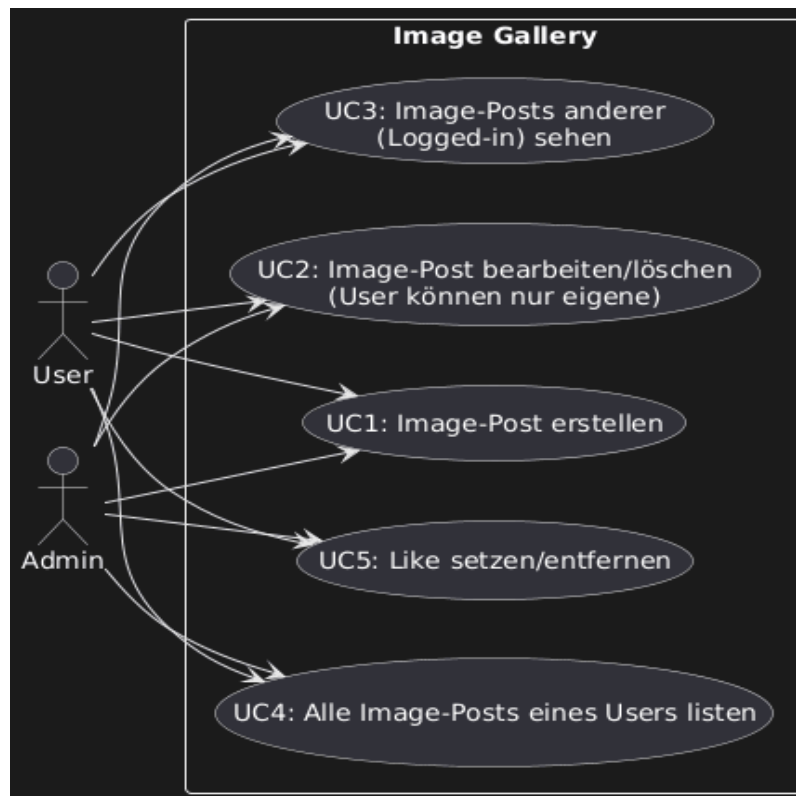
2.2 ERD



2.3 Sequence Diagramm



3 Use Cases



3.1 Definition Use-Cases

UC1

Actor(s):	User
Description:	User erstellt einen Image-Post.
Preconditions:	<ul style="list-style-type: none"> Der User hat sich eingeloggt
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 Plus-Button User wählt Foto und Beschreibung Daten werden validiert User klickt auf Submit-Button
Alternative Courses:	4f. Falls Validierung fehlschlägt, zeig Error Message. Zwingt dich Foto einzufügen
Exceptions:	/

UC2

Actor(s):	User
Description:	User bearbeitet oder löscht seine eigenen Image-Post.
Preconditions:	<ul style="list-style-type: none"> • Der User hat sich eingeloggt • Rechte wurden überprüft • User hat Image-Posts
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 Image-Post • User bearbeitet Image-Post. • 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:	/
Exceptions:	/

Actor(s):	Admin
Description:	Administrator bearbeitet oder löscht einen Image-Post.
Preconditions:	<ul style="list-style-type: none"> • Der Admin hat sich eingeloggt • Rechte wurden überprüft • Users haben Image-Post
Postconditions:	<ul style="list-style-type: none"> • Daten werden umgeschrieben/gelöscht/geändert • Admin navigiert weg oder bleibt
Normal Course:	<ul style="list-style-type: none"> • Admin Navigiert zu einer User Image-Post-Liste • Admin klickt auf einen Image-Post • Admin bearbeitet Image-Post • Admin kann die Bearbeitung speichern, indem er auf den «speichern» Buttons drückt oder löschen, indem er auf den «löschen» 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:	/

UC3

Actor(s):	User
Description:	Eingeloggte User sehen Image-Post 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 weg navigieren oder bleibt
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 geht wieder raus
Alternative Courses:	/
Exceptions:	/

UC4

Actor(s):	User
Description:	Eingeloggte User sehen Image-Post 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 weg navigieren oder bleibt
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 geht wieder raus
Alternative Courses:	/
Exceptions:	/

UC5

Actor(s):	User
Description:	User können die Post von einem spezifischen User sehen
Preconditions:	<ul style="list-style-type: none"> User wählt User
Postconditions:	<ul style="list-style-type: none"> User kann weg navigieren oder bleibt
Normal Course:	<ul style="list-style-type: none"> User wählt anderen user User sieht seine Posts
Alternative Courses:	/
Exceptions:	/

4 Testing

- [1] Test ob Bilder gespeichert werden
- [2] Sieht ein User die Posts der anderen?
- [3] User Genehmigung
- [4] User Authentifizierung
- [5] Ich kann Admin Post/Infos verändern bzw. löschen
- [6] Ich kann Admin User verändern bzw. löschen
- [7] Wenn Bild fehlt, muss Seite meckern
- [8] Bei ungültiger URL meckern
- [9] Nur wenn ein User angemeldet ist kann er posten
- [10] Kann ein User seinen eigenen Post bearbeiten/löschen
- [11] Bei nichtexistierendem Endpoint 404

4.1 Vorgehensweise

Tools & Umgebung:

- Backend: Spring Boot + JUnit/MockMvc
- Frontend: React + Cypress
- Manuelle Checks: Postman
- Auth: JWT (User, Admin, Gast)

Testdaten:

- User A (ROLE_USER) → Besitzer
- User B (ROLE_USER) → Fremder
- Admin (ROLE_ADMIN)
- Mind. 2 Posts von A

4.2 Getestete Endpoints

Registrieren: <http://localhost:8080/user/register>

Login: <http://localhost:8080/user/login>

User Update: <http://localhost:8080/user/{userId}>

Image-Post Update: <http://localhost:8080/posts/{postId}>

Delete User: <http://localhost:8080/user/{userId}>

Delete Image-Post: <http://localhost:8080/posts/{postId}>

Alle User: <http://localhost:8080/user>

Alle Image-Posts: <http://localhost:8080/posts>

User Image-Posts: <http://localhost:8080/posts/user/{userId}>

Like Image-Post: <http://localhost:8080/posts/{postId}/like>

4.3 Erwartetes Verhalten

Testschritte:

1. UC1 – Erstellen

- Happy Path: Post wird gespeichert ([1])
- Fehler: fehlendes Bild ([7]), ungültige URL ([8])
- Nur eingeloggte User dürfen posten ([4],[9])

2. UC2 – Bearbeiten/Löschen

- Owner/ darf eigenen Post ändern/löschen ([10])/Admin darf andere Post ändern/löschen ([5])
- Fremder blockiert (403) ([3])
- Admin darf alle ändern/löschen ([5])
- Unbekannte ID → 404 ([11])

3. UC3 – Anzeigen

- Eingeloggter User sieht Posts anderer ([2])
- Gast blockiert (falls gefordert)

4. UC4 – Image-Posts bestimmten Users

- User klickt auf bestimmten User
- Alle Post von dem Spezifischen User werden angezeigt

5. UC5 – Liken und Like entfernen

- User sieht einen Post
- User drückt auf Like
- User kann like wieder entfernen(falls er will)

Abnahme:

- Richtige HTTP-Codes (200/201/204/400/401/403/404)
- Daten korrekt gespeichert/gelöscht
- UI zeigt Fehlermeldungen bei Validierungsfehlern