Projektauftrag: Zusammenfassung

üK223: Multi-User-Applikation objektorientiert realisieren

# AUsgangslage

* Social-Media-Webseite
* Rollen:
  + User
    - Können Blog-Posts erstellen
    - Können den Blog-Posts Kategorien zuteilen
    - Können Teil einer Gruppe werden
  + Admin
    - Können Kategorien für Blog-Posts erstellen/verwalten

# Auftrag

* Umsetzung von Bedingungen mit Beachtung von Multiuser-Fähigkeit und Sicherheit
* Vorlage: Skelett-Projekt
  + Funktionalitäten:
    - Login bestehender User
    - Erstellen von User, Rollen, Autoritäten
    - Frontend: Login und Routing

## Pflicht-Anforderungen

* User Rollen & Privilegien
  + Mehrere Rollen und Autoritäten definieren (mindestens User, Admin)
    - Müssen für Aufträge sinnvoll sein
  + Passwortverschlüsselung implementieren
  + Persönliche Informationen
    - nur für Admins und zugehörigen User sichtbar
  + Admin Privilegien
    - Andere User bearbeiten, erstellen, löschen
* Frontend
  + Pages (minimum)
    - Login-Page (öffentlich)
    - Homepage (eingeloggte User)
    - Admin-Page (Admins)
  + Korrektes Rechte-Management und Error-Handling
* Security
  + Geeignete Autoritäten für jeden REST-Endpoint
    - Mit automatischen Tests geprüft
  + Exklusive Bereiche für eingeloggte User
  + Exklusive Bereiche für Admins
  + JWT-Authentifizierung implementieren

### Gruppenspezifisch

* Image Gallery
  + Image-Post Model
    - Enthält Informationen über geteiltes Bild
      * URL
      * Author
      * Beschreib
      * Likes (Liste von User)
  + User kann mehrere Image-Post erstellen
    - Jeder Image-Post gehört genau einem User
  + CRUD-Endpoints für Image-Post-Einträge
  + Image-Post nur vom Admin oder zugehörigem User bearbeitbar/löschbar

### Nicht-Funktionale Anforderungen

* Implmentation
  + DB: PostGreSQL, OR-Mapping mit JPA
  + Frontend: TypeScript React
  + Backend: Springboot Java
  + Git: tägliche Commits in einem Repo mit Front- und Backend
* Testing
  + Generell für Endpoints
  + Fokus auf Zugriffsberechtigung
  + Mindst. Ein Use-Case ausführlicher getestet
    - Endpoint mit mehreren Users und Rollen getestet
    - Mindst. Erfolgsfall und Error Fall getestet
    - Use-Cases nach UML-Standard
* Multiuserfähigkeit
  + Aspekte (bsp. ACID-Prinzipien)
* Dokumentation
  + Readme (im Code-Repo eingecheckt)
    - Wichtigste Informationen
    - Setup-Anleitung für Ausführung des Codes und Tests
  + Swagger
  + Domänenmodell
  + Testing-Strategie inkl. Use-Case Beschreibung
  + Sequenzdiagramm