### **Project Documentation & UML Overview**

### **Project Title**

**Cumra Task - Role-Based Full Stack Application** 

#### **Tech Stack**

- Frontend: React.js (with Context API, Bootstrap)
- Backend: Spring Boot (Spring Security, JWT Auth)
- Database: MySQL
- Architecture: REST API, Three-Tier Architecture

### **Objective**

Build a secure, role-based application where users can submit data, and admins can manage users and submissions.

### **Project Documentation & UML Overview**

#### **Roles and Permissions**

#### Admin:

- Login
- View all users
- View, edit, delete all submissions
- Promote/Demote users
- Delete users

#### User:

- Sign up & login
- Submit, edit, delete own submissions
- View own dashboard and history

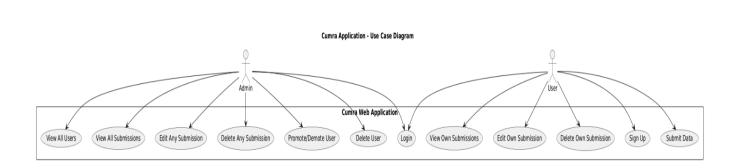
#### **Core Functionalities**

- JWT-based authentication
- Role-based route protection
- CRUD operations for submissions
- Admin dashboard with user controls

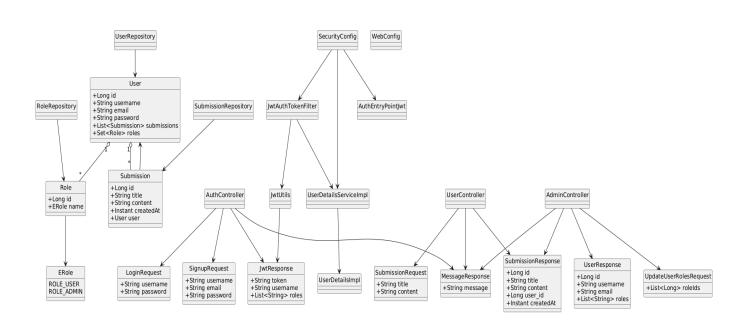
# **Project Documentation & UML Overview**

### **UML Diagrams**

#### 1. Use Case Diagram

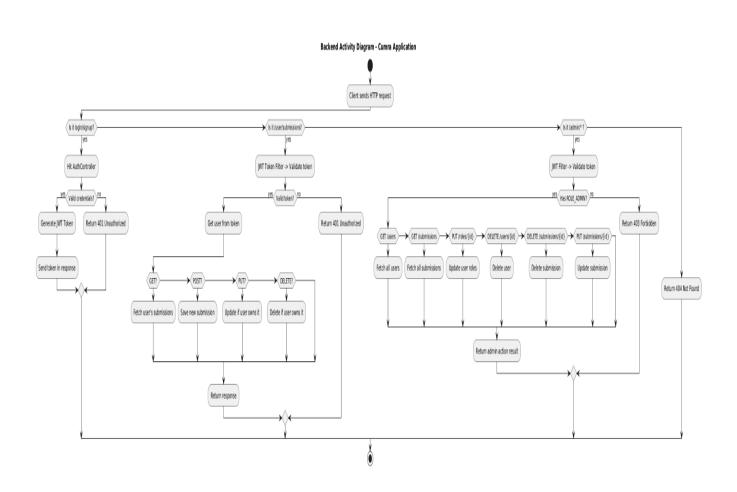


#### 2. Class Diagram



# **Project Documentation & UML Overview**

### 3. Activity Diagram (Backend Flow)



### **Project Documentation & UML Overview**

### 4. Sequence Diagram (Login Flow)

#### Cumra Application - User Login Sequence UserDetailsServiceImpl LoginPage (React) AuthController (Spring Boot) AuthenticationManager LocalStorage Enters username/password POST /api/auth/login authenticate(credentials) loadUserByUsername() Fetch User by username User Entity UserDetails Authentication Success generateToken(UserDetails) JWT token Response (token, roles, username) Store token, role Redirect to /dashboard LocalStorage LoginPage (React) UserDetailsServiceImpl AuthController (Spring Boot) AuthenticationManager JwtUtils

### **Project Documentation & UML Overview**

### 5. Component Diagram (Frontend)

