The Idea

- This is a backend project, written in Java, made possible by the Spring Framework. This is a backend written for a Project Management application to serve the purpose and solve the problem of project and task management issues. I plan to upscale this project in the future, as it will be used and presented to the NHS.

Below are the various outlined tools and functionalities that make this backend work.

NB: Some of these features will be implemented in time while the frontend is implemented and in the later future when the project will be upscaled.

Core Functionality

- **Project Management:**
  - Define a clear data model (database entities and relationships) for projects, tasks, subtasks, milestones, dependencies, and deadlines.
  - Implement RESTful APIs for creating, reading, updating, and delet-ing (CRUD) projects and associated tasks.
  - Use of the JPA (Java Persistence API) library and Hibernate to sim-plify database interactions.
- **User Management:**
  - Implement robust authentication and authorization (use Spring Se-curity).
  - Develop simple user profiles, role-based permissions (project man-agers, team members, etc.).
  - Create APIs for user registration, login, and profile management.
- **Task Management:**
  - Assign tasks to project members.
  - Track task statuses (Not Started, In Progress, Completed, etc.).
  - Set priorities for tasks.
  - Allow for task comments and history logs.
- **Collaboration:**
  - Real-time updates using technologies like WebSockets for seamless collaboration.
  - A commenting system for tasks and project-level discussions.
  - File sharing and attachments linked to tasks or projects.
- **Reporting & Analytics:**
  - Generate project progress reports.
  - Time tracking and resource allocation reporting.
  - Visualizations like Gantt charts or Kanban-style boards to enhance project overviews.

Entity Relationships Considered

Backend Architecture

- **REST API Design:** Adopt best practices for RESTful API development, ensuring consistency and ease of integration for the Next.js frontend.

- **Data Validation:** Enforce strong input validation on the backend to prevent invalid data from corrupting the project management system.

- **Error Handling:** Implement graceful error handling, providing meaningful feedback to the frontend to enhance the user experience.

The features below will be implemented in the later future

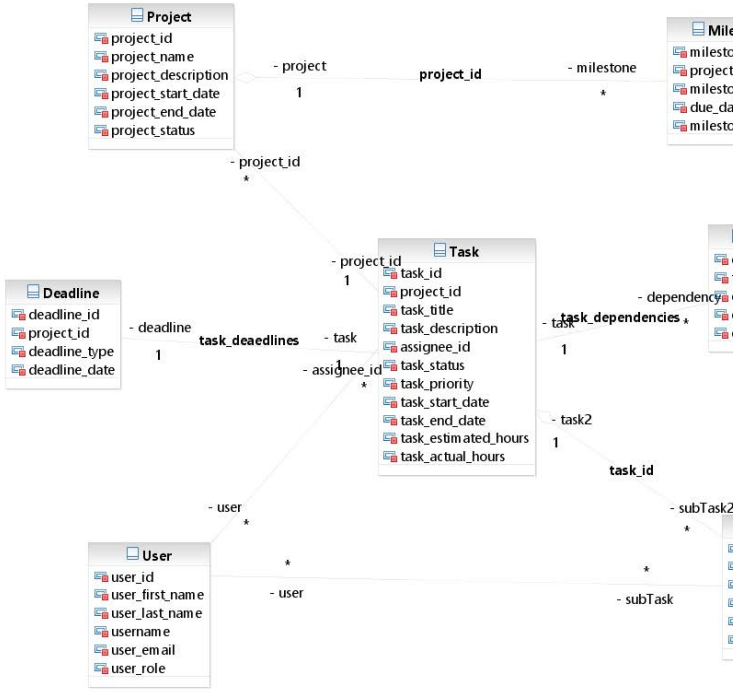
Additional Features

- **Notifications:** Implement email or in-app notifica-tions for task assignments, deadlines, project updates.

- **Search & Filtering:** Provide robust search anc and users.

- **Calendar Integration:** Consider integrating v Calendar, Outlook) to streamline project timelines

- **Version Con-trol:** Integrate with a version c documents or code.



Scalability & Performance

- **Database Optimization:** Design e icient databa caching strategies as your usage grows.

- **Asynchronous Operations:** Use asynchronou improve backend responsiveness.

- **Load Balancing:** Imple-ment load balancing t base or resource-intensive features.

Security

- **Authentication & Authorization:** Implement strong security measures with Spring Security for authentication and role-based authorization.

- **Input Sanitization:** Protect against XSS (Cross-Site Scripting) and SQL injection attacks.

- **Regular Updates:** Keep your Spring Boot dependencies up-to-date to address potential vulnerabilities.