HOCHSCHULE ESSLINGEN

Software Architectures

Part 71: Project Jodel

Project Objectives

Features

- Develop a simple web based Jodel application
- Share thoughts, hints, and observations via posts
- Connect with people nearby
- Comment and vote on comments

Technologies

- Use PostgreSQL RDBMS
- Use Java 17 and Spring Boot
- Use ReactJS as client side technology
- Use Docker and Keycloak

Rationale behind using these technologies

1. PostgreSQL is one of the most widely used open source databases.
2. Java has the most complete, freely available library set of all computer languages. Java systems are state of the art as far as software architecture is concerned.
3. Spring Boot has collected the best libraries for each purpose and makes them available in a single framework. It supports compilation to native code for most efficient memory footprints and fast startup times as required by the upcoming microservices architecture. It's the hottest sh*t on the market.
4. ReactJS is a widely used and modern JavaScript framework by Facebook. An alternative would be Googles Angular framework with its traditional model-view-controller architecture, but we think ReactJS with its state-based architecture is cooler and allegedly faster.

Features

- Login against your Google or Facebook account, Keycloak, or a local user name and password
- Read all posts 10 km around your current location that have been posted or commented lately
- Newest posts or comments should show first
- Comment and vote on comments (one vote up or down per person per comment)

Deployability

1. You need to keep in mind that your application later on needs to run on some server, not just localhost:8080
2. Avoid using absolute URLs in your code like https://localhost:8080/xxx/yyy, use /xxx/yyy instead
3. Make sure that the base URL is configurable when you start the server, for example via command line parameters or environment variables
4. Make sure the database user and password are configurable when you start the server, for example via command line parameters or environment variables

Outline of the development process

1. Create an UML class diagram for the entities in the database
2. Write the enitiity classes (the model) in Java
3. Using Spring Boot create the database schema and filll it with some test data
4. Develop the ORM classes to access the database entities
5. Develop the ressource classes for the server side REST interface
6. Develop the client side ReactJS code for the login screen
7. Develop screen by screen the rest of the client side functionality

Requirements and Grading

- You must use one Gitlab account to manage your projects version control. Each team member must have a significant amount of commits. **Failure to show these commits means you fail your project, no kidding!**
- You must have Java unit tests for all non-trivial classes.
- You must be present on the dates announced for project reviews on Moodle.

Grading

- Database design (correctness, completeness): 10 points
- ORM and database related functionality (design, correctness, completeness, test coverage): 20 points
- REST API (functional completeness, correctness, test coverage): 20 points
- UI login (completeness, design, usability): 10 points
- UI posting (completeness, design, usability): 10 points (**SWB students only**)
- UI comments and voting (completeness, design, usability): 20 points (**SWB students only**)
- Build and deploy process (installation from scratch on new computer): 10 points