CPSC 476 - Java Enterprise Application Development

Programming Assignment 4 - Spring 2016

due May 10

In this last assignment, you will complete the transition of your event calendar application from servlets and JDBC to Spring MVC and JPA.

If you did not complete Programming Assignment 3, you may instead modify your application from a previous Programming Assignment to use Spring JPA Repositories.

In either case, the rest of the front-end code from the previous assignment should remain unchanged.

Entities

Create a Persistence Unit and define Entities and relationships for at least the following items:

- Users
- Events
- Likes

If your previous programming assignments stored additional values, you will need to create Entities for those items as well.

DAOs and Repositories

Your DAO interfaces from Programming Assignment 2 should remain unchanged.

Note: You will likely be annotating existing POJO classes from Programming Assignment 2, rather than defining new Entities from scratch. If switching to JPA requires your existing DAO interfaces to change, document the changes, explain why they are required, and discuss how the interfaces might have been designed differently in order to avoid the change.

If you did not complete Programming Assignment 2, define `@Repository` interfaces as in Chapter 21 instead.

Databases and JPA Providers

You may use either of the following databases:

- HSQLDB (as in Programming Assignment 2)
- ObjectDB (which is easy to configure, does not require a separate server process, and has excellent documentation)

You may use any JPA implementation. You might, for example, use Hibernate as shown in the textbook, or try EclipseLink configured with IntelliJ IDEA, or switch to ObjectDB.

Mapping and Schema Generation

If you are using HSQLDB, you may either annotate your Entities to map to your existing tables from Programming Assignment 2 or turn on schema generation and allow your JPA provider to create new tables for your Entities.

Working with Other Students

You may complete this project on your own, but you are encouraged to work together with one or two other students.

If you choose to work with others:

- Submit only a single assignment.
- Include the names of all members of a group on the submission.
- Each student in a group will receive the same grade.
- You may discuss the assignment with other groups, but each group must submit its own work.
- You may choose to work with a different group on future assignments.

Submission

E-mail the following in a single `.ZIP` or `.tar.gz` file to csuf.kenytt.net@gmail.com by 11:59p on the date indicated:

1. Your Maven `pom.xml` or other build / project files.

2. The `.java` and `.JSP` source code for your application

3. Any additional resources (e.g., graphics, deployment descriptors, XML files, SQL DDL scripts) required by your application.

4. Documentation in PDF format, including

    a. A brief description of your project's architecture
    b. Build instructions, if any are required beyond a simple `mvn package` command.
    c. Screenshots demonstrating your application's functionality

Include your name and the other members of your team (if you have one) in your e-mail.

Do not submit compiled code or `.WAR` files. Your submission should be in a condition to be re-built from source.

Set the `Subject:` line of your e-mail to

```
[CPSC 476] Assignment 4
```

You may submit multiple times before the deadline; I will only grade the most recent submission before the deadline, unless your e-mail indicates that I should do something else. Late work will not be accepted after the deadline.

Grading

This programming assignment will be graded on the following 10-point scale, based on the instructor's judgement of the amount of time required to submit A-level work:

10	Application builds and runs correctly as described above.
9	Minor issues with submission or format. Work required to correct the issues is likely to be less than 20 minutes.
8	Minor issues with the application. Work required to correct the issues is likely to be less than two hours.
7	Major issues. Work required to correct the issues is likely to be more than two hours.
6	Application substantially fails to meet requirements, but shows a good faith effort to attempt the project.
5	Application substantially fails to meet requirements. Minimal progress made.
4	Code submitted without documentation, or code fails to compile.
0	No assignment submitted.