**Spring Boot Week 17 Final Project Requirements**

Over the next two weeks, you will be implementing or “building” the approved project that you proposed to your instructor.

**INSTRUCTIONS:**

**\*\* How to Get Started -- Guidelines (first steps to start your Final Project):**

1. Create your project in STS (or an IDE of your choice). Create the basic package structure for all of the layers (e.g. controller, service, DAO, entity, errorhandling, etc.).
2. Create the entry point for your project.
3. Create an Interface for each implemented class in the programming layers.
4. Modify the OpenAPIDocumentation in the Controller Interface to represent the correct responses for each of your operations. (e.g. @GetMapping expects an HTTPStatus.OK, but a @PostMapping expects an HTTPStatus.CREATED)
5. Implement your simplest entities first (no dependencies), and get them working in all layers.
6. Use jeep-sales as a guide, and build upon what you have already done!
7. UnitTesting is encouraged, but not required.

Use Swagger to test your API — including the OpenAPIDocumentation in the Controller Interface for your Web API, just as was done in jeep-sales! Another testing option is Postman, however, the Swagger interface is quite easy to use.

**\*\* How to Submit your Spring Boot Week 18 Final Project**

**\* Create a GitHub Repository for your Final Project**

1. Push all of your code to your Final Project Repository (Repo).

2. Push any relevant .SQL files to your Repo.

3. Push a PDF of your ERD to your Repo.

4. Include any additional documentation for your project.

5. Add link to Text Box Submission

**\* Create a Showcase Video (Public Link)**

Create a **video**, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed. This video should be done using screen share and voice over. This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend. You can create a new meeting, start screen sharing, and start recording. This will create a video recording on your computer. This should then be uploaded to a publicly accessible site, such as YouTube, Dropbox, or Google Drive. *MAKE SURE THE LINK YOU SHARE IS PUBLIC*. If it is not accessible by your grader, your project will be graded based on what they can access. The link should be pasted in the submission text box after the GitHub repo link.  ***REQUIRED****:  PUBLIC link to video, and GitHub repo link with everything listed above!* [Notes: dual monitors, swagger, eclipse, also dbeaver/MySQL Work bench Shows that STS starts, pull up swagger (refresh browser), showcase CRUD operations (http server response 200, etc.), optional: demo dbeaver/MySQL if time permits to successful creation of a record; voice over. Zoom meeting with private meeting with myself.

***Video Instructions for Group Projects:***

- In a **Group Project**, each member of the group is required to submit their own 5-minute max video of their part of the project. The video should include an overview of the project, and show the functionality that this person implemented.   Including the Swagger output after requests are run.

**-** In a **Group Project**, it is **required** to document **which team member is responsible for which entities**.  The easiest way to accomplish this is to comment the code that each team member writes.  Since each team member is required to code an entity from top to bottom, add an Author  comment at the top of each file.

**REQUIREMENTS**:  

**Project Requirements:**

**1-person Project:**

* + Database design which contains at least 3 entities and 3 tables
  + Contains all CRUD operations (Create, Read, Update & Delete)
  + **Each entity** should have**CRUD operations** with one entity having all 4 CRUD operations (Create, Read, Update & Delete).
  + Contains at least 1 one-to-many relationship
  + Contains at least 1 many-to-many relationship with one or more CRUD operations on this relationship.

**Group Project (max 3 people):**

* + Database should have at least two tables per group member + one.
  + Contains 2 entities per person, plus 1 entity done together at the beginning to solidify the structure expected for the project.
  + Each member of the group is required **to code their entities top to bottom**, including entity, controller, service, & DAO.
  + Each member of the group is required have CRUD operations on both of their entities, and implement all 4 CRUD operations (Create, Read, Update, & Delete) on one of their 2 entities
  + Contains at least 1 one-to-many relationship
  + Contains at least 1 many-to-many relationship with one or more CRUD operations on this relationship

Web API Documentation Resources:

https://blog.api.rakuten.net/best-practices-for-writing-api-documentation/

https://www.altexsoft.com/blog/api-documentation/

https://swagger.io/blog/api-documentation/best-practices-in-api-documentation/

**Nothing needs to be submitted to the Learning Management System this week. You will submit your final project in Week 18: Final Project under *Week 18 Final Project Submission***