**Points possible:** 70

|  |  |  |
| --- | --- | --- |
| Category | Criteria | % of Grade |
| Functionality | Does the code work? | 25 |
| Organization | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| Creativity | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| Completeness | All requirements of the assignment are complete. | 25 |

**Coding Steps:**

**🗸**

1. In the application you've been building add a DAO layer:

**🗸**

* 1. Add the package, com.promineotech.jeep.dao.

**🗸**

* 1. In the new package, create an interface named JeepSalesDao.

**🗸**

* 1. In the same package, create a class named DefaultJeepSalesDao that implements JeepSalesDao.

**🗸**

* 1. Add a method in the DAO interface and implementation that returns a list of Jeep models (class Jeep) and takes the model and trim parameters.

**🗸**

1. In the Jeep sales service implementation class, inject the DAO interface as an instance variable. The instance variable should be private and should be named jeepSalesDao. Call the DAO method from the service method and store the returned value in a local variable named jeeps. Return the value in the jeeps variable (we will add to this later).

**🗸**

1. In the DAO implementation class (DefaultJeepSalesDao):

**🗸**

* 1. Add the class-level annotation: @Service.
  2. Add a log statement in DefaultJeepSalesDao.fetchJeeps() that logs the model and trim level. Run the integration test. Produce a screenshot showing the DAO implementation class and the log line in the IDE's console. 

I finished the coding for all of the videos, I am too far past this point for a screen shot

**🗸**

* 1. In DefaultJeepSalesDao, inject an instance variable of type NamedParameterJdbcTemplate.

**🗸**

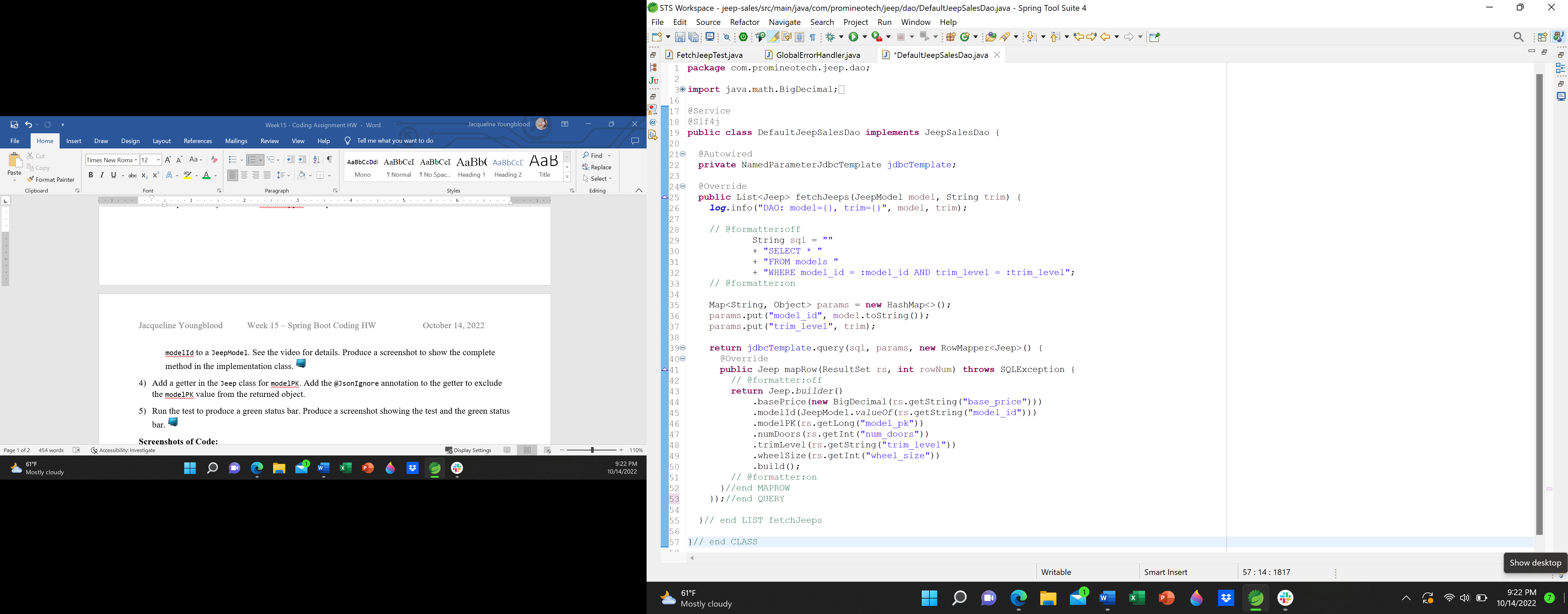
* 1. Write SQL to return a list of Jeep models based on the parameters: model and trim. Be sure to utilize the SQL Injection prevention mechanism of the NamedParameterJdbcTemplate using :model\_id and :trim\_level in the query.

**🗸**

* 1. Add the parameters to a parameter map as shown in the video. Don't forget to convert the JeepModel enum value to a String (i.e., params.put("model\_id", model.toString());)

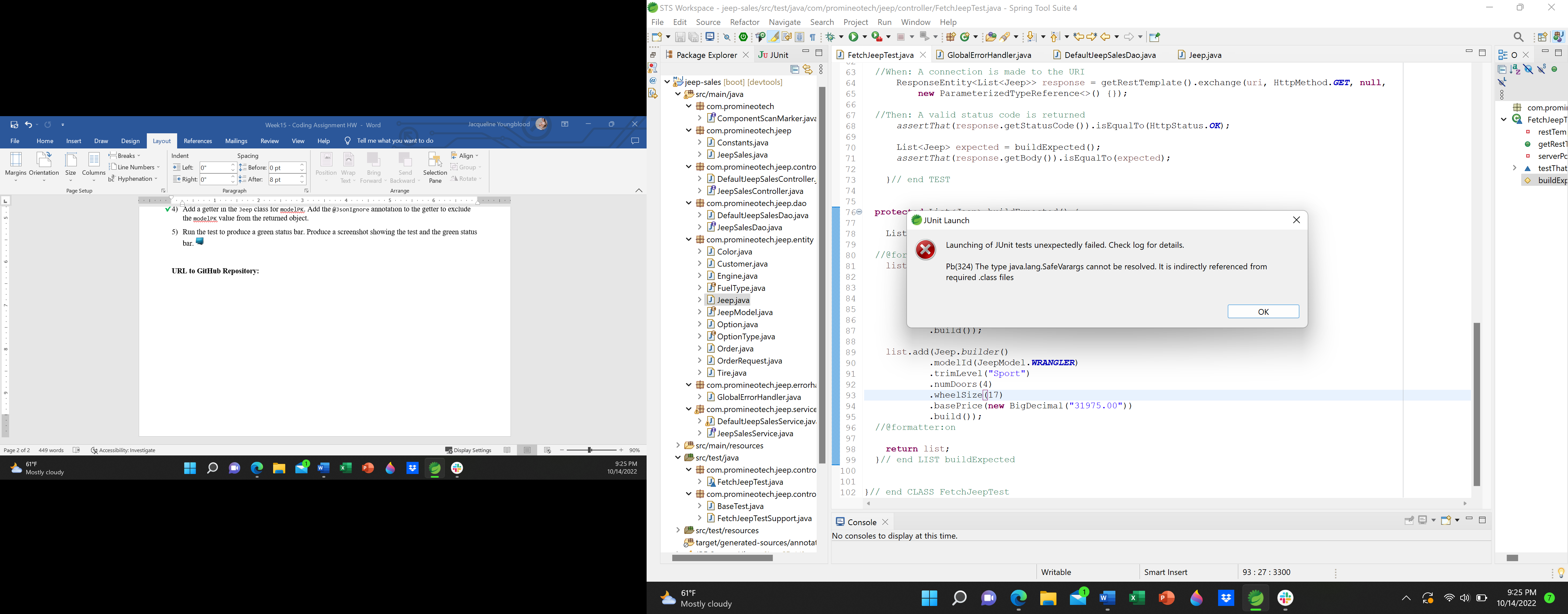
**🗸**

* 1. Call the query method on the NamedParameterJdbcTemplate instance variable to return a list of Jeep model objects. Use a RowMapper to map each row of the result set. Remember to convert modelId to a JeepModel. See the video for details. Produce a screenshot to show the complete method in the implementation class. 



**🗸**

1. Add a getter in the Jeep class for modelPK. Add the @JsonIgnore annotation to the getter to exclude the modelPK value from the returned object.
2. Run the test to produce a green status bar. Produce a screenshot showing the test and the green status bar.  At this point my Junit was unable to run. I have contacted a mentor and awaiting help



**URL to GitHub Repository:**

**https://github.com/JaxYoungblood/Week15-SpringBootCodingAssignment.git**