CS489 – Applied Software Development

Final Exam: Structure and Review Points

The Final Exam will consist of questions given in 2 parts, as follows:

Part 1: Theory – This part will be composed of Knowledge-based questions, including Short answer questions, True/False questions (some + rationale), Multiple-Choice questions and some short/small coding questions.

Part 2: Software Design and Development skills – Analysis, Design, Coding / Problem solving skills

This part will be composed of questions requiring Object-oriented Software requirements analysis, domain modeling, software design and implementation tasks, including diagraming.

Part 1: Not more than 40%

Part 2: At least 60%

The Exam will be all computer-based. i.e. You will use computers for both parts 1 and 2. However, for diagrams, you are free to draw using Pencil and Paper.

The exam duration will be 2 hours (timed from 10 am to 12 noon).

The following are the relevant lessons/topics and sample questions:

Lesson 6: Data Persistence

Topics / Sample Questions :

1. What is Data Persistence
2. Database
3. JDBC
4. Java Persistence API (JPA)
    - 4.1.JPA Annotations
        - 4.1.1. @Entity
        - 4.1.2. @Id
        - 4.1.3. @GeneratedValue – strategy: GenerationType.AUTO, GenerationType.IDENTITY, GenerationType.SEQUENCE etc.
        - 4.1.4. Data Validation
            - 4.1.4.1. Database Column Constraints –
                - 4.1.4.1.1. @Column(nullable=false)

4.1.4.2. Bean validation annotations:
4.1.4.2.1. @NotNull – null value is invalid, empty string “” is valid
4.1.4.2.2. @NotEmpty – null value and empty string “” is invalid
4.1.4.2.3. @NotBlank - null, empty string “” and whitespace(s) is invalid
4.1.5. @OneToOne
4.1.6. @OneToMany and @ManyToOne
4.1.7. @ManyToMany (results in creating a Join Table)
4.1.8. mappedBy
4.1.9. @Transient
4.1.10. Cascade behaviors
4.1.10.1. CascaseType.PERSIST
4.1.10.2. CascaseType.MERGE
4.1.10.3. …
4.1.11. Fetch strategies
4.1.11.1. FetchType.EAGER
4.1.11.2. FetchType.LAZY

4.2.…
4.3.…

5. Hibernate – ORM framework that implements the JPA specification
6. Spring Data JPA – the dependency in Spring platform for Data persistence using JPA.
   6.1. Repository pattern and @Repository – Repository interface (Is just a marker interface), CrudRepository, ListCrudRepository, PagingAndSortingRepository, ListPagingAndSortingRepository, JpaRepository
   6.2. Commonly used abstract methods in the repository interface(s): findAll, findById – Optional<T>, save(T t) etc.
7. …
8. …

Lesson 7: RESTful Web API

1. What is REST?
2. What is a RESTful Web API?
3. Spring Web MVC
   3.1. @RestController
   3.2. @RequestMapping
   3.3. @GetMapping, @PostMapping, @PutMapping, @DeleteMapping
   3.4. @CrossOrigin – CORS, SOP etc.
4. Using DTO
5. …

Lesson 8: Full-stack Application Devt

Lesson 9: Security

1. What is Software Security
2. What is Authentication a.k.a authcn
3. What is Authorization a.k.a authzn
4. Session-based (using cookies) versus Token-based auth
5. JWT
6. Spring Security
7. …

Lesson10: Container

1. What is containerization?
2. What is a container?
3. What is Virtualization? How is it different from containers
4. How to containerize an application using Docker
5. …

Lesson 11: Software Testing

1. Why automated testing?

2. Types of tests
    a. Unit test
    b. Integration test
    c. What is mocking?
    d. JUnit framework – v4 and v5
    e. Mockito library for mocking
    f. Hamcrest library for assertions
    g. …
3. …

Lesson 12: Software Deployment / Azure Cloud

1. How to package and delivery software product for end user consumption
2. Microsoft Azure cloud services – PaaS
    a. Azure App Service
    b. Azure Database for MySQL or PostgreSQL
    c. Azure SQL database

Please note that you are expected to have and apply knowledge of topics in the following prior lessons where needed, though this Final Exam will not be targeted at testing that knowledge, directly.

Lesson 1a: Software Development tools

Lesson 1b: Software Build Automation

Lesson2: Software Requirements, Analysis and Design / Domain modeling

Lesson3: Software Development Platforms

Lesson4: Architecture

Lesson5a: Databases I (Relational Database design and development)

Lesson5b: Databases II (Non-relational Database a.k.a NoSQL database – MongoDB)

Note: *For some of the questions/tasks in the part 2 of the Exam, you will be expected to take screenshot(s) of your work/result(s), and save them to an image file (.png or .jpg or .jpeg only) and include these in the zip file, which you will submit.*

Also, when you take the screenshots, it should be of the entire computer screen (NOT a snippet or a window

Sample/Practice Enterprise Application Software Development (Part2) Question:

Problem Statement:

Assume that a company has hired you to design and develop an Enterprise Web Application for their Supplier Relationship Management (SRM) system, which they will be using to manage data about their Suppliers and the Products they supply. Specifically, the system will be used for viewing, updating and maintaining the Suppliers and Products details. They want you to design and implement the backend RESTful Web API with a Database, for this purpose.

An important need for the companys' managers, is to be able to view the list of all Star Suppliers**. A Star Supplier is a supplier who has supplied at least 2 Products, and the total dollar value of the products supplied is more than $100,000.00**.

For this question, it is given that a Supplier can supply zero or more Products. And each Product **must** have a Supplier.

IMPORTANT: Your solution model should consist of only the two entity classes, named:

Supplier
Product

Which will contain the Supplier's basic data and the Product(s) data, respectively.

Here are the attributes for the **Supplier** entity, including some useful descriptions and/or sample data values:

Supplier:

supplierId: long, PK

name, (e.g. Google, Inc., Apple Inc., Samsung etc.) - Required

contactPhone, (e.g. (123) 456-7890, etc.) - Optional

Here are the attributes for the **Product** entity, including some useful descriptions and/or sample data values:

Product:

productNo, (e.g. 3128874119, 2927458265, 9189927460 etc.)
Note: These productNos are numbers and are required data and should be unique

name, (e.g. Pixel 8 Pro, Apple iPad, Samsung Galaxy 20 etc.) - Required

dateSupplied, (e.g. 2023-01-24, 2022-12-09, 2023-03-31 etc.) - Optional

quantityInStock, (e.g. 124, 18, 89 etc.) - Optional

unitPrice, This is money (in dollars and cents) (e.g. $1599.55, $1700.09, $1225.99 etc.) - Required

Data:

Here is the company's existing data, which you are expected to create/load in the application, using a database:

Suppliers-and-Products data:

<table>
<tr><th>Supplier Id</th><th>Supplier Name</th><th>Contact Phone</th><th>Products</th></tr>
<tr><td>1</td><td>Google, Inc.</td><td>null</td><td>
<table>
<tr><th>ProductNo</th><th>Name</th><th>Date Supplied</th><th>Quantity</th><th>Unit Price</th></tr>
<tr><td>3128874119</td><td>Pixel 8 Pro</td><td>2023-01-24</td><td>124</td><td>$1599.55</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Apple, Inc.</td><td>(641) 123-0987</td><td>
<table>
<tr><th>ProductNo</th><th>Name</th><th>Date Supplied</th><th>Quantity</th><th>Unit Price</th></tr>
<tr><td>2927458265</td><td>iPad 14 Pro</td><td>2022-12-09</td><td>18</td><td>$1700.09</td></tr>
</table>
</td></tr>
</table>

			2927458266	iPhone 15	2023-10-11	95	$1650.85
3	Samsung Corporation	null					
4	Huawei	null					

For this question, you are required to do the following:

TASK A: Using a UML drawing tool (or you may draw by hand on paper and pencil/pen), draw the Domain model class diagram, showing the two entities and their relationship, including the multiplicity etc.

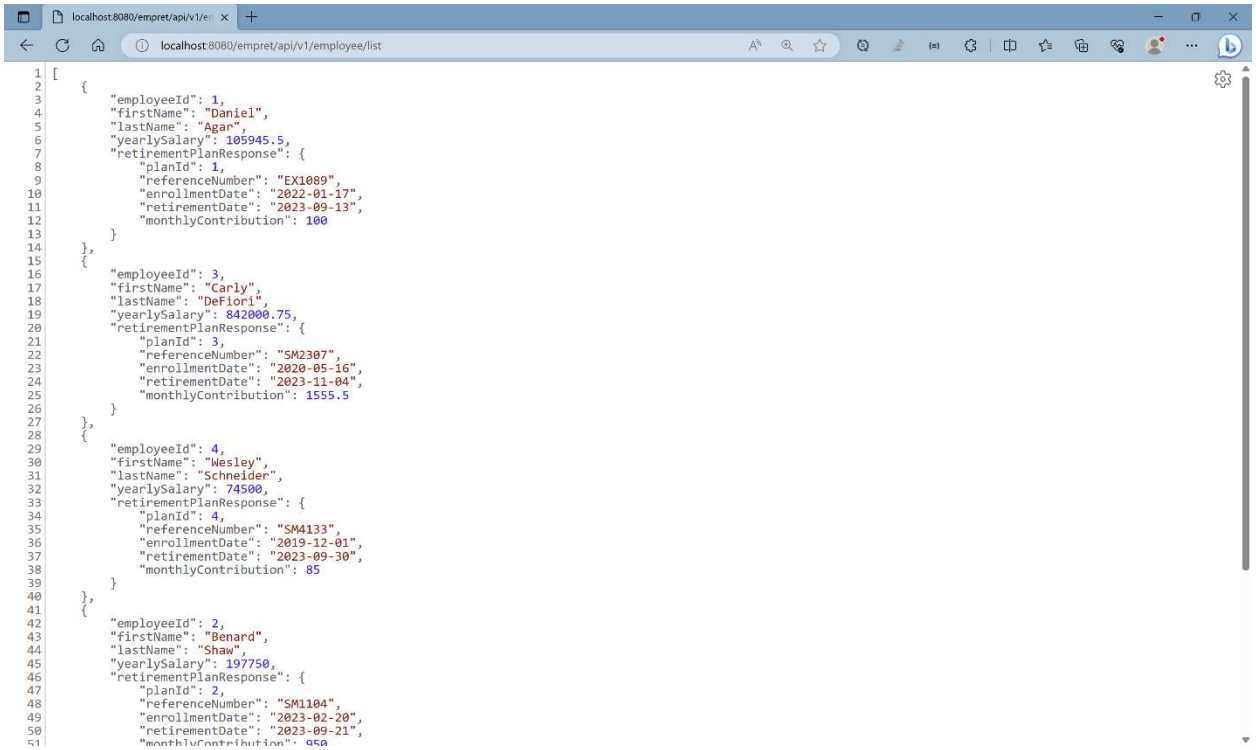
TASK B: Using some Enterprise Application development platform, framework, tools (such as Spring platform and Spring Boot or some other which you prefer), implement a RESTful Web API Project for the company's need.

Here are the features/functionalities that you are required to implement in code:

1. Present List of all Suppliers, including the Products sort by supplier name
2. Implement a feature which presents the data of the **Star Suppliers**, in JSON format. **Note:** This data should contain only the list of Suppliers who qualify as star suppliers (see above) and it should include/show the Products for each star Supplier. The company requires this list to be displayed sorted in ascending order of the Supplier Name.
3. Implement RESTful Web API endpoint url to update the data for an existing Product, given its productNumber
4. Present list of Products and include the Supplier data for each Product

Take screenshot(s) of your output result for the above requirement(s), and include in your zip file submission.

Example JSON-formatted list of all Suppliers:
(Note: Sorted in ascending order of the Supplier Names)




```
[
    {
        "employeeId": 1,
        "firstName": "Daniel",
        "lastName": "Agar",
        "yearlySalary": 105945.5,
        "retirementPlanResponse": {
            "planId": 1,
            "referenceNumber": "EX1089",
            "enrollmentDate": "2022-01-17",
            "retirementDate": "2023-09-13",
            "monthlyContribution": 100
        }
    },
    {
        "employeeId": 3,
        "firstName": "Carly",
        "lastName": "DeFiori",
        "yearlySalary": 842000.75,
        "retirementPlanResponse": {
            "planId": 3,
            "referenceNumber": "SM2307",
            "enrollmentDate": "2020-05-16",
            "retirementDate": "2023-11-04",
            "monthlyContribution": 1555.5
        }
    },
    {
        "employeeId": 4,
        "firstName": "Wesley",
        "lastName": "Schneider",
        "yearlySalary": 74500,
        "retirementPlanResponse": {
            "planId": 4,
            "referenceNumber": "SM4133",
            "enrollmentDate": "2019-12-01",
            "retirementDate": "2023-09-30",
            "monthlyContribution": 85
        }
    },
    {
        "employeeId": 2,
        "firstName": "Benard",
        "lastName": "Shaw",
        "yearlySalary": 197750,
        "retirementPlanResponse": {
            "planId": 2,
            "referenceNumber": "SM1104",
            "enrollmentDate": "2023-02-20",
            "retirementDate": "2023-09-21",
            "monthlyContribution": 950
```


//-- The End --//