**PE\_SBA301\_SP25\_TrialTest**

SPRING 2025  
Subject: SBA301  
Duration: 85 minutes

**INSTRUCTIONS**

**Please read the instructions carefully before doing the questions.**

* You are **NOT allowed** to use any other materials. You are **NOT allowed** to use any device to share data with others.

* You must use IDE as **IntelliJ IDEA 2023 or later, MSSQL Server 2019 or** **later** for your development tools.

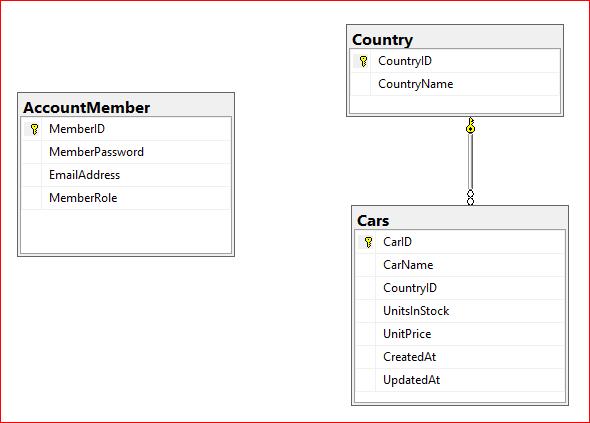
**IMPORTANT – Before you start doing your solution, MUST do the following steps:**

* To do your program, you must use **Spring REST, apply 3-Layer architecture. *Note that*** *you are not allowed to connect directly to the database from REST Controller, every database connection must be used with Repository and Data Access Objects. The database connection string must get from hibernate.cfg.xml file or application.properties file.* ***In the case your program connects directly to the database from a REST Controller or hardcode connection string, you will get 0 points.***

1. ***If there are syntax errors or compilation errors in your PE program, you will not pass the PE requirements, the point will be 0.***
2. Create a new ReactJS Project in Visual Studio Code named **PE\_SBA301\_SP25\_FE\_StudentName**.
3. Create a new Spring REST Project in IntelliJ IDEA named **PE\_SBA301\_SP25\_BE\_StudentCode**.
4. Create your MS SQL database named **CarStore2025DB** using a code-first approach with Hibernate, and ensure that it includes the data provided in the **PE\_SBA301\_SP25\_TrialTest\_Note** file
5. ***Your work will be considered invalid (0 point) if your code inserts stuff that is unrelated to the test.***

A MS SQL Server database will be created to persist the data and it will be used for reading and managing data.

The Figure below is a part of **CarStore2025DB** database.



Note that: Member*Role: Admin = 1; Staff = 2; Member=3*

| ***Role***  ***Function*** | ***Admin*** | ***Staff*** | ***Member*** |
| --- | --- | --- | --- |
| ***Create*** | *P* | *P* |  |
| ***Update*** | *P* | *P* |  |
| ***Delete*** | *P* |  |  |
| ***View list of items*** | *P* | *P* |  |
| ***View detail of an item*** | *P* | *P* |  |
| ***Search*** | *P* | *P* |  |
| ***Login*** | *P* | *P* |  |

*Table 01 : Permission Role*

***Task 1. Setup Environment***

1. *(2.0 point) Database migration* using JPA/Hibernate. Create related models then migrate to a database named **CarStore2025DB** (MS SQL Server).
2. *(1.0 point)* Check authentication/authorization using JSON Web Token (JWT) for Spring REST

***Task 2. Implement CRUD actions with Spring REST.***

***Please check the permissions listed in Table 01; In case you do the authentication/authorization wrong, you will not get any points for this section;***

| ***Function*** | ***API endpoint*** | ***Description*** | ***HTTP Status Code*** |
| --- | --- | --- | --- |
| *Lists* | *GET /api/infocars* | *Get all* | *200, 401, 403* |
| *Create* | *POST /api/infocars* | *Create a new item* | *201, 401, 403* |
| *Update* | *PUT /api/infocars/{id}* | *Update an exist item* | *200, 401, 403, 404* |
| *Delete* | *DELETE /api/infocars/{id}* | *Delete an existing item* | *200, 401, 403, 404* |

1. *(1.0 point)* List all cars in Cars table (the information includes CarID, CarName, UnitsInStock, UnitPrice, CountryName, CreatedAt, UpdatedAt) **(Note that, No permissions for this function).**
2. (0.5 point) Check if login successfully with *Admin* role, delete the selected item with the confirmation then update the list of cars list.
3. (1.5 points) Check if login successfully with *Admin* role, add new item with the r*equirements:*

* *The new item will display at the top of the list.*
* All fields are required.
* Value for UnitsInStock <= 20 and >= 5.
* Value for CarName is greater than 10 characters.
* Value for CreatedAt = CurrentDate and CreatedAt <=UpdatedAt

***Task 3. Design FE components/pages.***

1. (1.5 point) Using React-Bootstrap to design Navbars Screen includes UI controls for login function.

*Navigation Bar*

*Navbar.Brand = “Student Code - Student Name PE Spring 25”*

*Navbar.Link = Home*

*Nav.Dropdown = Car Management (List all cars ,Create a new car)*

*Login Dialog with Close Button*

*Modal.Title = “Login to Cars Management System”*

1. (2.5 point) Design Cars Management page, this form includes UI controls for CRUD actions with *car* information.

Note:

* The *CountryName* will come from the ***Country*** table. Design a form which allows you to view the list of records, create a new item, update the existing item, and delete a specific item.
* No permission for “List all items” function
* *Component: Form.Select=* List of country names