**PE\_SBA301\_SUM25\_TrialExam**

SUMMER 2025  
Subject: SBA301  
Duration: 85 minutes

**INSTRUCTIONS**

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

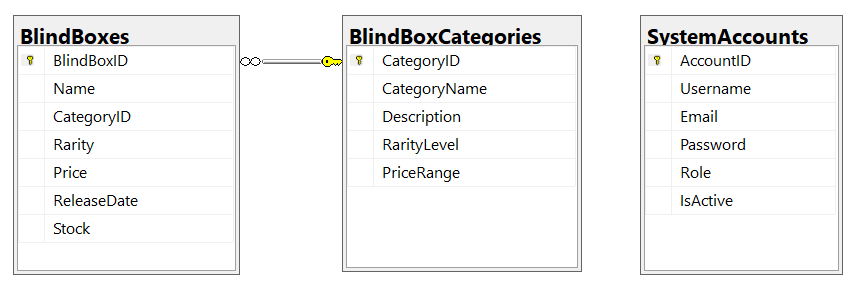
* You are **NOT allowed** to use any other materials. You are **NOT allowed** to use any device to share data with others.
* You must use IDE as **IntelliJ IDEA 2023 or later, MSSQL Server 2019 or** **later** for your development tools.

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

1. To do your program, you must use **Summer 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 Service and Repository. 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 point.***
2. ***If there are syntax errors or compilation errors in your PE program, you will not pass the PE requirements, the point will be 0.***
3. Create a new ReactJS project in Visual Studio Code named **pe\_sba301\_sum25\_trial\_fe\_studentcode**
4. Create a new Summer REST project in IntelliJ IDEA named **pe\_sba301\_sp25\_trial\_be\_studentcode**
5. Create your MS SQL database named **Sum2025BlindBoxesDB** using a code-first approach with Summer Data JPA and ensure that it includes the data provided in the **PE\_SBA301\_SUM25\_TrialExam\_Note** file
6. ***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 **Sum2025BlindBoxesDB** database.



Note that: *Role: Administrator = 1; Moderator = 2; Developer = 3; Member=4*

***Task 1. Setup Environment***

1. *(2.0 point) Database migration* using JPA/Hibernate. Create related models then migrate to database named **Sum2025BlindBoxesDB** (MS SQL Server).
2. *(1.0 point)* Check authentication/authorization using JSON Web Token (JWT) for Summer REST (*using Email, IsActive and Password*).

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

1. *(1.0 point)* List all items in *BlindBoxes* table (the information includes all information in this table and CategoryName from *BlindCategories* table). **Note that, No permissions for this function.**
2. (0.5 point) Check if login successfully with *Administrator* role, delete the selected item with the confirmation then update the list of items.
3. (1.5 points) Check if login successfully with *Administrator* 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 Stock <= 100 and >= 1.
* Value for Name is greater than 10 characters.
* Value for ReleaseDate = Current date

***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 Name SBA301 - PE Summer 25”*

*Navbar.Link = Home*

*Nav.Dropdown = Blind Boxes Management (List all items ,Create a new blind box)*

*Login Dialog with Close Button*

*Modal.Title = “Login to Blind Boxes Management System”*

1. (2.5 point) Design Blind Boxes Management page, this form includes UI controls for CRUD actions with *blind box* information.

Note:

* The *CategoryName* will come from the ***BlindBoxCategories*** 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 category names