

## Topic: Supplier & Product Management System

---

### 1. GENERAL REQUIREMENTS

#### Technology Stack

- Spring Boot (MVC pattern, Thymeleaf View) + Spring Data JPA + MS SQL Server/MySQL

#### Architecture

- Apply N-Layer Architecture: Controller – Service – Repository – Entity
- Do **not** connect to the database directly from Controller or Service.
- All database operations must go through the Repository layer.

#### Project Structure

- Project name: HSF302\_GroupNumber
- Must include at least 4 packages: entity, repository, service, controller

#### Database

- Database name: HSF302

#### Authentication & Authorization

- Use **HttpSession** to store login information.
- Use **Spring Interceptor** to check all requests:
  - If the user has not logged in → automatically redirect to the login page.

---

### 2. DATABASE STRUCTURE

#### 2.1. user\_accounts

Column	Type	Note
--------	------	------

id	int	Primary key, auto-increment
----	-----	-----------------------------

username	varchar	Account username
----------	---------	------------------

password	varchar	Account password
----------	---------	------------------

role	int	1 = Manager, 2 = Staff, 3 = Guest
------	-----	-----------------------------------

---

#### 2.2. suppliers

Column	Type	Note
--------	------	------

id	int	Primary key, auto-increment
----	-----	-----------------------------

Column	Type	Note
suppliername	varchar	Supplier name

---

### 2.3. products

Column	Type	Note
id	int	Primary key, auto-increment
productid	varchar	Unique product code
name	varchar	5–50 characters
price	float	$\text{price} \geq 0$
supplier_id	int	Foreign key $\rightarrow$ suppliers.id
created_at	date	Creation date
updated_at	date	Last update date
created_by	varchar	Username of the creator

---

## 3. FUNCTIONAL REQUIREMENTS

### Task 1. User Interface

#### Login Page

- URL: `http://localhost:8080/`
  - Attributes:
    - Title = *Login Page*
    - Students can freely design the page style
  - Allows input of username and password for login
  - Use **HttpSession** to store login information
  - If the user's role is not a Manager, Staff should see an error message:
    - "You have no permission to access this function!"
  - If login credentials (Manager, Staff) are invalid, show:
    - "username or password is invalid!"
- 

#### Suppliers Page

- Attributes:
  - Title = *Suppliers Page*

- Students can design styles freely

Only user's role is manager can access this page (from Product Management Page) to do CRUD.

**Page layout:** design is up to the student.

---

### **Product Management Page**

- Only **Staff** can perform CRUD operations
- **Manager** can only view the product list (only the 5 products with the highest prices), go to the Suppliers page.

### **Page Attributes:**

- Title = *Product Management Page*
- Students can freely design the style

### **Page Layout:**

#### **Input Form (top section):**

- ProductID (TextField)
- Name (TextField)
- Price (TextField)
- Supplier (ComboBox – loaded from the suppliers table), Button **Suppliers**
- Buttons: **Add, Update**

#### **Product List (bottom section):**

- Columns: ID, ProductID, Name, Price, Supplier, CreatedBy, CreatedAt, UpdatedAt
- Each row includes **Edit** and **Delete** buttons

**Display the username** of the logged-in user at the top-right corner of the Product Management page.

**Logout** button located at the top-left corner.

### **Pagination Requirement**

- Product Management Page must support **pagination** when displaying the product list.
- Each page displays **5 products**.
- Use **Spring Data JPA Paging and Sorting** (Pageable, PageRequest).
- The UI must include navigation controls: **Previous / Next**
- **Manager view** (Top 5 by highest Price) is **not paginated**.
- **Staff view** (CRUD) uses pagination to manage all products.

### **Validation Requirement**

- Use **Spring Validation** with annotations such as:

- @NotBlank, @Size, @Min, @Max, @NotNull, @Valid
- Display clear error messages on the UI when input is invalid.
- Validation applies to both **Add** and **Update** forms.

---

## Task 2. CRUD + Authentication Features

### Entity & JPA Mapping

- Map products → suppliers using a **Many-to-One** relationship

---

### Login & Authentication

- Login using username and password (via HttpSession)
- Only accounts with roles **Manager** or **Staff** are allowed to access the system

---

### Authorization

Role	Permission
------	------------

Manager	View product list ( top 5 by highest Price), CRUD Suppliers
---------	---

Staff	CRUD products
-------	---------------

Not logged in Redirect to login page

---

### Add Product

#### Validation:

- name: 5–50 characters
- price  $\geq 0.0$

#### When adding:

created\_at = updated\_at = current\_date, created\_by = logged-in username

---

### Update Product

- Same validation rules as adding
  - Can **only update** products created by the current user
  - When updating, set updated\_at = current\_date
-

### Delete Product

- Can **only delete** products created by the current user

### Logout Function

- When clicked → remove session info and redirect to login page

### Add Suppliers

- **Validation:** suppliername: 5-50 characters
- 

## Task 3. Spring Data JPA Requirements

1. Entity Mapping & Validation
2. Repository Layer
3. Paging & Sorting
4. CRUD qua Repository

Ensure proper Entity mapping, build standard JPA repositories, support pagination, sorting and perform CRUD operations through Service → Repository.

---

## 4. NOTES

- The user\_accounts table must contain predefined data for login (at least **1 Managers, 2 Staff and 1 Guest**) to test role-based restrictions.
- Use **HttpSession** to manage login and authorization.
- Use an **Interceptor** to block all requests except /login and /.
- **UI:** Students can freely design a suitable interface. User's role is manager need to login first to create Suppliers