Inventory System Management

Development:

Marcelo Eduardo Guillen Castillo

Java Monterrey Academy

September 17, 2024

Version history

Date	Reason	Version
09/09/2024	Creation of the file, requirements and user stories	1.0
10/09/2024	Adding the unit test documentation	1.1

Introduction

Business case

The company is requiring a new software system for managing a digital inventory system for its products, it is needed to create, read, update and delete registers of products and stocks, also the registration of the company on the platform. The problem is that it is very difficult to manage a lot of information manually at Excel and require another method to manage the company products in an efficient and productive way.

Objectives

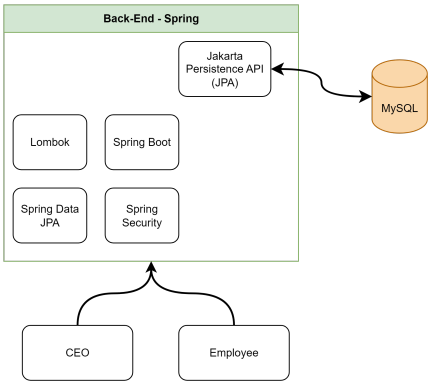
Create a robust application that can function in a long period of time, scalable and productive for the developers.

Scope

This will only scope MySQL databases, and is limited to a company’s stock, just a simple connection to a Java Back-End.

Summary

Architecture

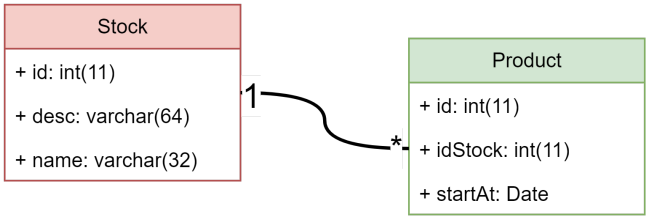


The Back-End will work with Spring framework, with dependencies like:

- Spring Data JPA
- Spring Security
- Lombok
- Spring Boot
- JPA

Where the connection will be with a MySQL server, managed by phpmyAdmin, and with unit tests made on postman.

Database tables



On the system will be products, where each one will have an initial date where it was acquired or made, an identification and the identification of the stock which it belongs to. A stock is a representation of the classification of a product, where it has an identification, a name and a description.

User roles

- Employee: In charge of obtaining data of the products and uploading new information about products and stocks.

- CEO: also known as the owner of the company, who can realize everything of the employee privileges, plus deleting existing data like products.

Limitations

- There will not be any Front-End application on any framework, just a Back-End app.
- The application will run just on a local server and will not be uploaded to any service.

Requirements

Functional requirements

1. Be able to view all the company's products
2. Create a stock for my products
3. View only products from a stock
4. Introduce a new product to my company's stock inventory
5. Being able to group my products
6. Modify any data about my products
7. Being able to access only one chosen product from a stock
8. Remove a product from my stock inventory
9. Sort the products in my inventory by date
10. Sort the products in my inventory by stock
11. Remove a stock from the inventory

Non-functional requirements

Exceptions of the system

- InvalidProductException: occurs at the moment when the user sends a product with invalid information for example that the user doesn’t input a correct date or an inexistent stock.

- NotIDProductException: occurs when the user sends a product to update doesn't have an ID to find the product or it has one but the ID number doesn't exist on the database.

Software requirements

- Windows 10
- Windows 11
- JDK 22

System attributes

- JDK 22
- Maven 3.7.9
- Postman 11.11.1
- MySQL 8.0
- Spring Boot 3.3.2
- JDBC 8.3.0
- Lombok 1.18.34
- Spring Data JPA 3.3.2
- Spring Security 3.3.2

User stories

US1. View all my company products	**Priority:** High	**Estimated time: 30 minutes**
Who: Employee and CEO **What:** Get all the products from all the stocks from my company **Why:** I could access to any register or stock for a product		
The employee and CEO can get all the products from all the stocks from a company to access to any register or stock for a product		
Acceptance criteria: ● When I consult for all the products, I will see all the products from all the stocks		

US2. Create a stock	**Priority:** High	**Estimated time: 30**

		minutes
Who: Employee and CEO **What:** Create a new stock for new products **Why:** Create a new classification of products		
Employees and CEO can create a new stock for new products to create a new classification of products		
Acceptance criteria: ● When I introduce correctly a name and a description for my stock, I will receive a message saying that the creation was successfully done ● When I introduce an empty name for the new stock, I will receive an error ● When I introduce a valid name, but with no description, the description default will say “No description”.		

US3. View all products from one stock	**Priority:** High	**Estimated time: 20 minutes**
Who: Employee and CEO **What:** View all the products from a stock **Why:** Get access to all the products from one classification		
An employee and the CEO can view all the products from a stock to get access to all the products from one classification		
Acceptance criteria: ● When I introduce the number ID of the selected stock, I will receive a list of all the products of that stock ● When I receive all the products from one stock, I expect to not receive the number of the stock which the product belongs ● When I introduce a not existent number ID of an stock, I will receive a message error indicating the inexistence of that stock		

US4. Introduce a new product into a stock	**Priority:** High	**Estimated time: 30 minutes**
Who: Employee and CEO **What:** Create a new product and put it on a new stock **Why:** Register it into the company inventory		
An employee and the CEO can create a new product and put it on a new stock to register it into the company inventory		
Acceptance criteria: ● When I introduce the data of the product correctly (without id), I will receive a message indicating that the creation was successful ● When I introduce the data of the product incorrectly, I will receive an error		

indicating incorrect information

US5. Being able to cluster products by stocks	**Priority:** Medium	**Estimated time: 2 hours**
Who: Employee and CEO **What:** Create clusters of products by stocks **Why:** Create groups of data and make data analysis		
An employee and CEO can create clusters of products by stocks to create groups of data and make data analysis.		
Acceptance criteria: ● When I tell it to group by a valid column from the product table, I will receive the list of JSON data, the stock data and then its products.		

US6. Modify any data about products	**Priority:** Medium	**Estimated time: 20 minutes**
Who: Employee and CEO **What:** Update the information about a product **Why:** Maintain updated the information on the database		
An employee and the CEO can update the information about a product to maintain updated information on the database.		
Acceptance criteria: ● When I send all the correct information about an already existing product with a different information (even ID), I will receive a notification saying that the information was successfully updated. ● When I send all the correct information without saying which ID of the product, I will receive a notification saying that the product doesn't exist. ● When I send incorrect information about a product, I will receive an error.		

US7. Access to the information about a single product	**Priority:** High	**Estimated time: 20 minutes**
Who: Employee and CEO **What:** Get information about a single product **Why:** Have a complete view about that product		
An employee and the CEO can get information about a single product to have a complete view about that product.		
Acceptance criteria:		

● When I introduce the correct ID for my product, I will receive all the information from that product. ● When I introduce an inexistent ID for my product, I will receive an error.

US8. Remove a product from inventory	**Priority:** High	**Estimated time: 20 minutes**
Who: CEO **What:** Delete a selected product **Why:** Get rid of an inexistent product		
The CEO can delete a selected product to get rid of an inexistent product.		
Acceptance criteria: ● When I introduce a valid product ID, I can see the message that the product was eliminated. ● When I introduce an invalid product ID, I would receive an error.		

US9. Sort products by date	**Priority:** Medium	**Estimated time: 1 hour**
Who: Employee and CEO **What:** Sort the products by date in ascendant or descendent ways **Why:** Make the task of searching more easier		
An employee and the CEO can sort the products by date in ascendant or descendent ways to make the task of searching easier.		
Acceptance criteria: ● When I send a 0, it indicates that I want the dates in ascending order. ● When I send a 1, it indicates that I want the dates in descending order. ● When I send an invalid data, I will receive an error.		

US10. Sort products by stock	**Priority:** Low	**Estimated time: 2 hours**
Who: Employee and CEO **What:** Sort the products by stock on the inventory, in ascendant or descendent ways **Why:** Have an estimation of the products with more or less supply		
An employee and the CEO can sort the products by stock on the inventory, in ascendant or descendent ways to have an estimation of the products with more or less supply.		
Acceptance criteria: ● When I send a 0, it indicates that I want the products stock in ascending		

order. ● When I send a 1, it indicates that I want the products stock in descending order. ● When I send an incorrect value, I will receive an error message.

US11. Remove a stock	**Priority:** Low	**Estimated time: 1 hour**
Who: CEO **What:** Eliminate a stock from the database **Why:** Get rid of an useless stock		
The CEO can eliminate a stock from the database to get rid of an useless stock.		
Acceptance criteria: ● When I send the correct stock ID, I will receive the message that the stock with its products were eliminated. ● When I send an incorrect stock ID, I will receive an error.		

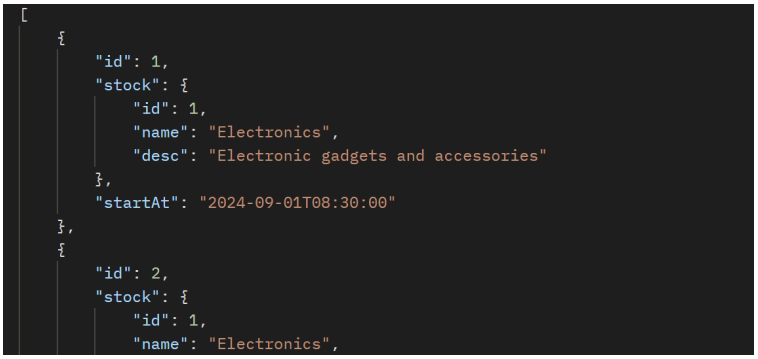
Unit tests

For the unit test cases, I am using Postman to create the requests and manually test the functionalities of the web service.

Here are some applications about all the tests made for the correct functioning of the project.

Test strategies

Test strategy
TS-1: US: 1:Be able to view all the company's products
Purpose
Verify that the system will bring all the products information correctly and complete.
Scenarios
● When I consult for all the products, I will see all the products from all the stocks, represented by JSONs with each product and it's stock which belongs.



```
[
    {
        "id": 1,
        "stock": {
            "id": 1,
            "name": "Electronics",
            "desc": "Electronic gadgets and accessories"
        },
        "startAt": "2024-09-01T08:30:00"
    },
    {
        "id": 2,
        "stock": {
            "id": 1,
            "name": "Electronics",
```

Test strategy
TS-2: US: 2:Create a stock for my products
Purpose
Verify that the creation of a new stock works correctly.
Scenarios
• When I correctly introduce a name and a description for my stock, I will receive a message saying “The stock was included correctly”. • When I introduce an empty name for the new stock, I will receive an error code status 404. • When I introduce a valid name, but with no description, the description default will say “No description”, and I will receive the message “The stock was included correctly”.

Test strategy
TS-3: US: 3:View all products from only one stock
Purpose
Verify that the system will show all the products from the correctly selected stock.
Scenarios
• When I introduce the number ID of the selected stock, I will receive a list of all the products of that stock.   ○ When I receive all the products from one stock, I expect to not receive the number of the stock which the product belongs to. • When I introduce a non-existent number ID of a stock, I will receive a

message “That stock doesn’t exist”.

Test strategy
TS-4: US: 4:Introduce a new product into the inventory
Purpose
Verify that the system will create a new product and put it into the inventory with its existing stock.
Scenarios
• When I introduce the data of the product correctly (without id), I will receive a message “The creation was successful!”. • When I introduce the data of the product incorrectly, I will receive an error indicating “Stock not found!”.

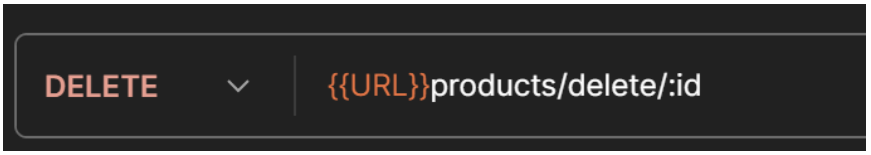
Test strategy
TS-5: US: 5:Being able to group by data my products by stocks
Purpose
Verify that the system will group by stocking my products.
Scenarios
• When I tell it to group by a valid column from the product table, I will receive the list of JSON data, the stock data and then its products.

Test strategy
TS-6: US: 6:Modify any data about my products
Purpose
Verify that the system will correctly modify an existing product from the database.
Scenarios
• When I send all the correct information about an already existing product with a different information (even ID), I will receive a notification saying “The

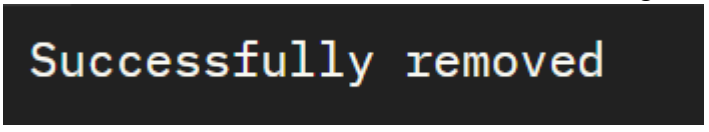
product was updated successfully”.
- When I send all the correct information without saying which ID of the product, I will receive a notification saying ”The product doesn’t exist”.
- When I send incorrect information about a product, I will receive an error saying “The information was not correct”.

Test strategy
TS-7: US: 7:Being able to see only one product
Purpose
Verify that the user will only seek a single product.
Scenarios
● When I introduce the correct ID for my product, I will receive all the information from that product. ● When I introduce an inexistent ID for my product, I will receive an error.

Test strategy
TS-8: US: 8:Remove a product from the inventory
Purpose
Verify that the system will correctly remove an existing product from the inventory.
Scenarios



- When I introduce a valid product ID, I can see the message that the product was eliminated and receive the message “Successfully removed”.



- When I introduce an invalid product ID, I would receive an error saying “The product doesn’t exist”.

```
"status": 404,
"error": "Not Found",
```

Test strategy
TS-9: US: 9:Sort the order of the products by date
Purpose
Verify that the system will return in a correct order the list of products by date, ascendant or descendant depending on the user's input.
Scenarios
• When I send a 0 or false, it indicates that I want the dates in ascending order. • When I send a 1 or true, it indicates that I want the dates in descending order. • When I send an invalid data, I will receive an error.

Test strategy
TS-10: US: 10:Sort the products in order by stock
Purpose
Verify that the products will return on ascendant or descendant depending on the stock’s disponibility.
Scenarios
• When I send a 0 or false, it indicates that I want the products stock in ascending order. • When I send a 1 or true, it indicates that I want the products stock in descending order. • When I send an incorrect value, I will receive an error message.

Test strategy

TS-11: US: 11:Remove a stock from the inventory
Purpose
Verify that a stock is completely removed, including its products from it.
Scenarios
• When I send the correct stock ID, I will receive the message that the stock with its products were eliminated. • When I send an incorrect stock ID, I will receive an error.