



# Inventory Management System

FINAL PROJECT

# CONTENT

- 01** PROBLEM
- 02** SOLUTION
- 03** GOALS
- 04** EXAMPLES

- 05** SCALABILITY
- 06** CONCLUSIONS

# PROBLEM

Normally companies need to classify and organize all their merchandise and any products they need to have within reach, but sometimes it is very difficult to have a constant count and tracking of large quantities of products.

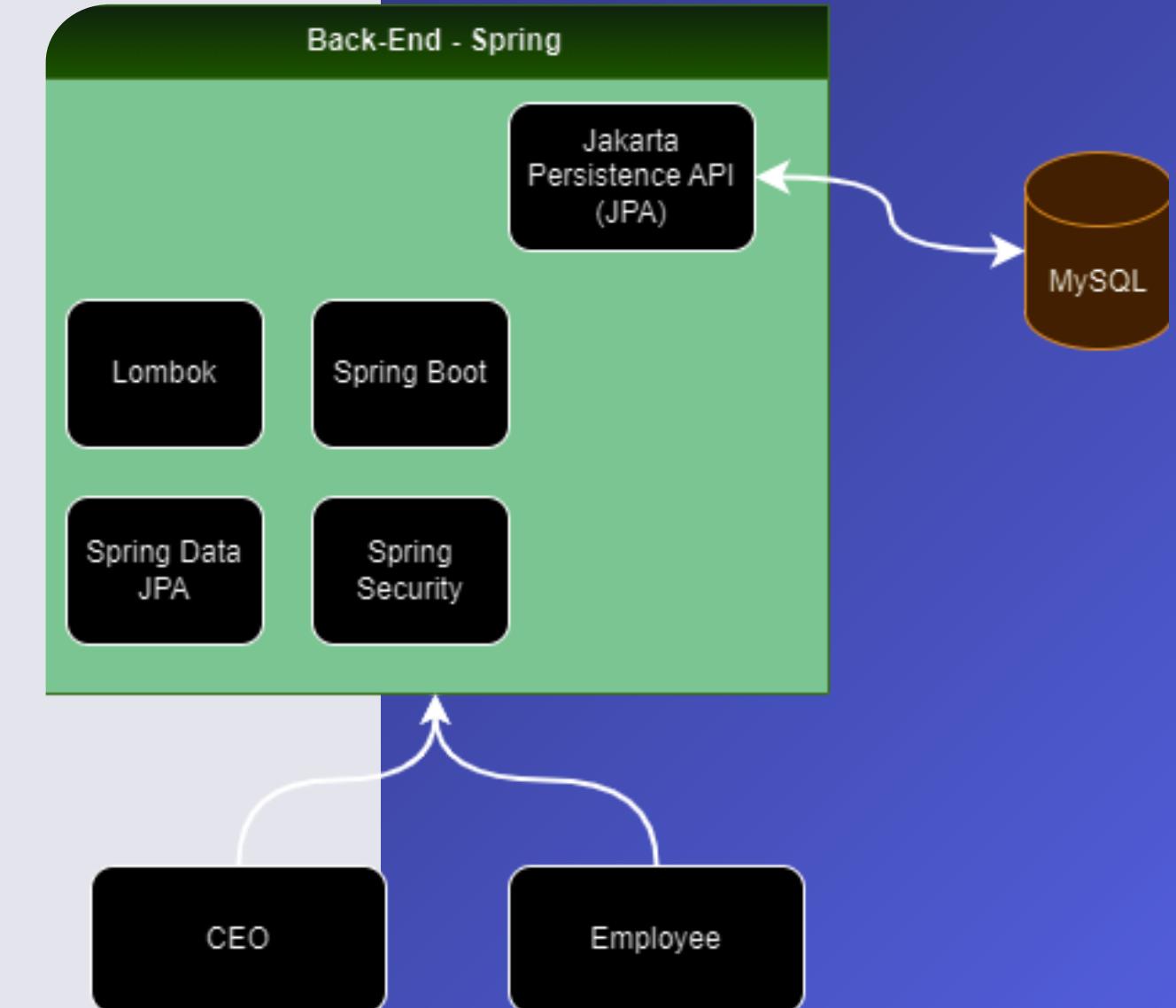
The classic solution is to make manual notes of all the products and the quantity of each one, either in Excel or with pencil and paper, but it is an outdated and not very scalable solution for companies that need to store large quantities of products and classify them.



# SOLUTION

Using current technologies, we decided to create a digital solution to the problem of inventory management with software focused on the administration, editing and modification of inventory product records. A web application that the employees of a company and its CEO or leader can access to be able to count their supply, perform statistics and facilitate the task of managing their company's supply in a comfortable and fast way.

It would be a cloud application connected to a database also in the cloud, which will ensure that only members of the company can have access to the system.



# GOALS

In this way we managed to solve several problems with old inventories.

- 01 Increase productivity.
- 02 Improve the asset management process.
- 03 Automate asset management.
- 04 Facilitate data analysis for business decisions.
- 05 Be a scalable project for future updates and improvements.

# EXAMPLE #1

Obtain product groupings by stock.

The screenshot displays two applications side-by-side: Postman on the left and MySQL Workbench on the right.

**Postman:** Shows a GET request to `(URL)stocks/grouped`. The request parameters are set to `Key: id, Value: 1`. The response area contains a placeholder message: `Click Send to get a response.`

**MySQL Workbench:** Shows the results of a SQL query: `SELECT * FROM `product``. The results are displayed in a table with columns: `id`, `stock_id`, and `start_at`. The data shows multiple entries for each stock ID, indicating grouping. The table has 16 rows.

	<code>id</code>	<code>stock_id</code>	<code>start_at</code>
<input type="checkbox"/>	1	1	2024-09-01 08:30:00
<input type="checkbox"/>	2	1	2024-09-02 09:15:00
<input type="checkbox"/>	3	1	2024-09-03 10:45:00
<input type="checkbox"/>	4	2	2024-09-04 11:00:00
<input type="checkbox"/>	5	2	2024-09-05 14:20:00
<input type="checkbox"/>	7	3	2024-09-07 15:30:00
<input type="checkbox"/>	8	3	2024-09-08 16:10:00
<input type="checkbox"/>	9	4	2024-09-09 17:45:00
<input type="checkbox"/>	10	4	2024-09-10 18:30:00
<input type="checkbox"/>	11	5	2021-03-03 12:00:00
<input type="checkbox"/>	12	5	2024-09-11 13:09:02
<input type="checkbox"/>	13	5	2024-09-11 13:09:03
<input type="checkbox"/>	14	5	2024-09-11 13:09:04
<input type="checkbox"/>	15	5	2024-09-11 13:09:05
<input type="checkbox"/>	16	5	2024-09-11 13:09:06

**MySQL Workbench Console:** Displays the executed SQL queries:

```
> SELECT * FROM `product`
> SELECT * FROM `stock`
> SELECT * FROM `product`
```

# EXAMPLE #2

Obtain the list of products ordered from those with the least to the most stock supply.

The screenshot illustrates a dual-monitor setup for API development and database management. On the left monitor, the Postman application is open, displaying a project named "Java Xideral Academy Final Project". A GET request is being prepared to the URL `(URL)/products/sortedByStock/ord`. The "Params" tab shows a single parameter `ord` set to `Ascendant`. The "Response" tab contains placeholder text: "Click Send to get a response." On the right monitor, the MySQL Workbench application is running a query: `SELECT * FROM `product``. The results show 16 rows of product data, each with columns `id`, `stock_id`, and `start_at`. The data is sorted by `start_at` in ascending order. The MySQL Workbench interface also includes a "Console" tab at the bottom where the same query was run.

id	stock_id	start_at
1	1	2024-09-01 08:30
2	1	2024-09-02 09:15
3	1	2024-09-03 10:45
4	2	2024-09-04 11:00
5	2	2024-09-05 14:20
7	3	2024-09-07 15:30
8	3	2024-09-08 16:10
9	4	2024-09-09 17:45
10	4	2024-09-10 18:30
11	5	2021-03-03 12:00
12	5	2024-09-11 13:05
13	5	2024-09-11 13:05
14	5	2024-09-11 13:05
15	5	2024-09-11 13:05
16	5	2024-09-11 13:05
27	6	2024-09-12 17:30

# EXAMPLE #3

Delete a stock from inventory, along with all its products, in this case reservation #2

The screenshot shows two applications side-by-side: Postman on the left and MySQL Workbench on the right.

**Postman:** A screenshot of the Postman interface. It shows a project named "Java Xideral Academy Final Project". Inside the project, there is a collection named "Stocks" which contains a request named "Delete stocks". The "DELETE" method is selected, and the URL is set to "[URL]/stocks/delete/:id". In the "Params" tab, there is a "Path Variables" section with a key "id" and a value "8". Below the request, there is a "Response" section with a placeholder message "Click Send to get a response".

**MySQL Workbench:** A screenshot of the MySQL Workbench interface. It shows a query results table titled "product". The table has columns: id, stock\_id, start\_at. The data is as follows:

	id	stock_id	start_at
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1	1	2024-09-01 08:30
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	2	1	2024-09-02 09:15
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	3	1	2024-09-03 10:45
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	4	2	2024-09-04 11:00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	5	2	2024-09-05 14:20
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	7	3	2024-09-07 15:30
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	8	3	2024-09-08 16:10
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	9	4	2024-09-09 17:45
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	10	4	2024-09-10 18:30
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	11	5	2021-03-03 12:00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	12	5	2024-09-11 13:05
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	13	5	2024-09-11 13:05
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	14	5	2024-09-11 13:05
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	15	5	2024-09-11 13:05
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	16	5	2024-09-11 13:05
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	27	6	2024-09-12 17:30

**MySQL Workbench Console:** The console shows the following SQL queries:

```
> SELECT * FROM `product`
> SELECT * FROM `stock`
> SELECT * FROM `product`
```

# SCALABILITY

Due to the technology used, in addition to the design of the service and its architecture, the result is a project that is not only profitable, but also scalable, with the possibility of expanding to new horizons and frontiers.

The basic idea of a flexible and digital inventory is a nest of new ideas that could improve the current service, and with that idea manage to invent a more complete system with great potential in the industry.



# CONCLUSION

The idea of a digital inventory favors supply companies or even small companies that have complications with the management and administration of their goods.

A bright future awaits and there are still opportunities to improve.

Original repository

