# Task 3

Summary:

In this assignment, you will develop a Chocolate Manufacturing Factory API, where users can interact with the application by creating and managing chocolate products. The application will be implemented as a Spring Boot project, utilizing RESTful APIs and JPA for data persistence. The assignment should be completed individually within 2 working days.

Functional Requirements:

Product Creation: Implement an API endpoint for creating new chocolate products, where authenticated users can specify the details of the product, including the name, ingredients, price, and quantity available.

Product Management: Implement API endpoints to enable product creators to edit or delete their own products.

Order Management: Implement API endpoints to allow users to place orders for chocolate products, including the ability to specify the quantity and shipping details.

Inventory Management: Implement API endpoints to enable factory managers to track inventory levels of chocolate products, including the ability to update the quantity available and view current stock levels.

Non-Functional Requirements:

Security: Implement authentication and authorization mechanisms to ensure that only authorized users can create products, manage inventory, and place orders.

Performance: Optimize the API implementation to handle multiple concurrent requests efficiently, ensuring minimal response times.

User-Friendly Interface: Design the API endpoints and response structures to be intuitive and easily understandable by client applications.

Data Integrity: Use JPA for data persistence, ensuring data consistency and reliability when creating, updating, and deleting products and orders.

Unit Testing: Use Junit to write effective unit tests.

CRON Job: Implement a CRON job to run at midnight every day to update the inventory levels of all chocolate products, based on the orders that have been placed and fulfilled that day. The job should subtract the quantity of each product that was ordered from the available quantity in the inventory.

Jasper Reports: Implement Jasper reports to allow factory managers to generate inventory reports for the chocolate products. The reports should show the current inventory levels, as well as any products that are low on stock and need to be reordered.

Examples of API Usage:

Product Creation:

Endpoint: POST /api/products

Request Payload: { "name": "Dark Chocolate Bar", "ingredients": ["cocoa", "sugar", "butter"], "price": 2.99, "quantity": 100 }

Response: 201 Created

Product Management:

Endpoint: PUT /api/products/{productId}

Request Payload: { "name": "Milk Chocolate Bar", "ingredients": ["cocoa", "sugar", "milk", "butter"], "price": 1.99, "quantity": 150 }

Response: 200 OK

Order Management:

Endpoint: POST /api/orders

Request Payload: { "product\_id": 1, "quantity": 5, "shipping\_address": "123 Main St, Anytown, USA" }

Response: 200 OK

Inventory Management:

Endpoint: GET /api/inventory

Response: 200 OK

Response Payload: { "products": [{ "id": 1, "name": "Dark Chocolate Bar", "quantity": 100 }, { "id": 2, "name": "Milk Chocolate Bar", "quantity": 150 }] }

CRON Job:

Endpoint: POST /api/cron/updateInventory

Response: 200 OK

Jasper Reports:

Endpoint: GET /api/reports/inventory

Response: 200 OK

Response Payload: { "products": [{ "id": 1, "name": "Dark Chocolate Bar", "quantity": 100, "reorder\_threshold": 20 }, { "id": 2, "name": "Milk Chocolate Bar", "quantity": 150, "reorder\_threshold": 30 }] }

Note: The above examples demonstrate the usage of the Chocolate Manufacturing Factory API endpoints. You should implement these API endpoints as part of the assignment, along with any additional endpoints required to fulfill the functional requirements.