Case Study Documentation on

Inventory Management System

Problem Statement

A large-scale manufacturing company requires a Python-based automated inventory management system to streamline the tracking, procurement, and usage of raw materials and finished goods. The system must integrate with various production units, warehouse systems, and supplier networks to ensure optimal stock levels, minimize production delays, and avoid overstocking or stockouts.

The system should :

- Monitor the real-time availability of materials.
- Automate reordering when inventory falls below a predefined threshold.
- Manage the transfer of raw materials between warehouses.
- Generate reports on inventory turnover and forecast future demand based on historical data.
- Send alerts for discrepancies or delays in shipments.
- Support multiple concurrent processes to handle inventory updates from various production units simultaneously and ensure data security across all departments.

Description

This system is designed to provide seamless inventory management for a large manufacturing company. By utilizing object-oriented programming (OOP) principles in Python, the system manages the core operations of inventory tracking, reordering, material transfers, and forecasting. It ensures that the inventory system can be accessed and updated concurrently from various production units while safeguarding sensitive data via an authentication mechanism.

The system is divided into multiple modules:

- **Inventory Management**: Track the Material stock and Sales.
- **Reordering System**: Automatically triggers a reorder when stock levels fall below a predefined threshold and alert messages are sent to the user.
- **Transfer Database**: Store timely logs for all the material transactions in the database and can be viewed as a table for monitoring and evaluation.
- **Forecasting System**: Generates reports based on historical data to forecast future demand and calculate inventory turnover.

- **Alert System**: Automatically sends messages on shipments, reorders and alerts to the registered user for timely tracking.
- **Security**: Ensures authentication for secure access to inventory data and modifications.
- **Report Generation**: Generates reports on request by the user on the sales and turnover rate.

Operations

1. **Inventory Management**
    - **Add Material**: Allows the addition of new materials to the inventory.
    - **View Inventory**: Allows to view the inventory data in tabular format.
    - **Update Stock**: Updates the stock level of an existing material.
    - **Delete material:** Deletes the material on request by the user.
2. **Reorder System**
    - **Check for Reorder**: Checks stock levels and triggers an automatic reorder for materials with stock levels below a threshold.
    - **Reorder Alert**: If no materials need reordering, the system outputs a message indicating no reorder is required.
    - **Alert via email**: Email alerts are automatically triggered whenever reordering occurs.
    - **Reorder display**: Allows user to view the reorders in his main menu.
3. **Transfer System**
    - **Transfer Materials**: Manages the transfer of materials between warehouses, including time logs.
    - **Shipment Alert**: Sends email alerts when shipment is delivered .
    - **View shipments**: Allows user to view the shipments across the ware houses with their id , and their durations via transaction database .

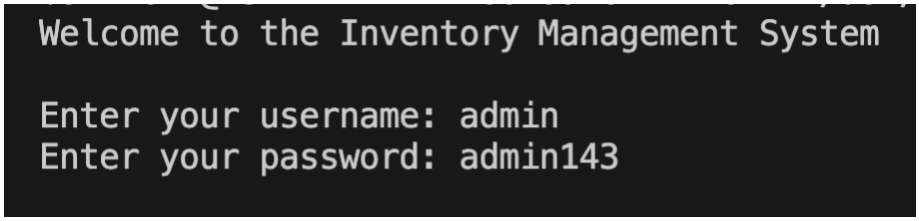
4. **Forecasting**
    - **Inventory Turnover Report**: Calculates the inventory turnover rate for each item.
    - **Sales Forecast**: Generates a forecast based on historical sales data with an assumed growth rate.
5. **Security**
    - **Authentication**: Ensures that only authorized users can access the system by verifying their credentials.

      NOTE: The credentials will be in authenticate file and can be edited as per user requirement. For the convenience of executing the program we have set both the username and password to **admin , admin143** respectively.

Sample Input and Output

1.Authentication

Input:



Output:

```
User authenticated successfully

Hello admin , Welcome to your Inventory Management

    ACTIONS

1. Add Material
2. View your inventory
3. Update your inventory
4. Generate Reports
5. Monitor your Shipments
6. View shipments
7. Transfer Materials
8. Delete material
9. View Reorders
10. Check if Reorder required
11. EXIT

Choose an action :
```

2. Adding a Material

Input:

```
Choose an action : 1
Enter Item ID: 01
Enter Material Name: Wheat Flour
Enter initial stock: 50
Enter reorder threshold: 10
Enter Price of material : 1275
Enter historical sales: 5200
```

Output:

```
Material Wheat Flour with ID 1 added successfully to the Inventory !
```

3. Displaying Stock Levels

Output:

```
 Choose an action : 2
+----+-------------+----------+-----------+-------------+------------+
| ID |   Material  | Qty in kg | Threshold | Price in Rs | Past_Sales |
+----+-------------+----------+-----------+-------------+------------+
| 1  | Wheat Flour |    50.0   |    10.0   |    1275.0   |   5200.0   |
| 2  | Corn Starch |    30.0   |    12.0   |    780.0    |   3400.0   |
| 3  |    Sugar    |   100.0   |    40.0   |    4000.0   |   9400.0   |
+----+-------------+----------+-----------+-------------+------------+
```

3. Update Materials

Input and output:

```
 Choose an action : 3
Enter Material ID to update : 2
Enter new quantity : 35
Enter the reorder threshold :11

 Material with id 2 has been updated successfully
------------------------------------------------------------
```

4. Inventory Turnover and forecast Report

Output:

```
 Choose an action : 4

 Material: Wheat Flour (ID: 1) - Turnover Rate: 4.08

 Material: Corn Starch (ID: 2) - Turnover Rate: 4.36

 Material: Sugar (ID: 3) - Turnover Rate: 2.35
Generating forecast report...

--- Forecast Report ---
   id         name  forecasted_sales
0   1  Wheat Flour            5720.0
1   2  Corn Starch            3740.0
2   3        Sugar           10340.0
```

5. Transfer Materials

Output:

```
 Choose an action : 5
Enter an id for shipment : 25
Enter source warehouse ID: w4
Enter destination warehouse ID: w5
Enter material ID to transfer: 1
Enter quantity to be transferred: 12
Transferring 12 units of material 1 from warehouse w4 to warehouse w5.

 Shipment alert has been sent successfully  !
```

6. **Monitor shipments**

```
 Choose an action : 6
Enter shipment id : 25
+----+--------+-----+--------+------+----------------------------+
| ID | Mat_ID | Qty | Source | Dest |            Time            |
+----+--------+-----+--------+------+----------------------------+
| 25 |   1    |  12 |   w4   |  w5  | 2024-09-22 13:06:29.905453 |
+----+--------+-----+--------+------+----------------------------+
```

7. **View Shipments**

```
 Choose an action : 7
+----+--------+-----+--------+------+----------------------------+
| ID | Mat_ID | Qty | Source | Dest |            Time            |
+----+--------+-----+--------+------+----------------------------+
| 1  |   1    |  10 |   w1   |  w2  | 2024-09-22 12:47:31.046017 |
| 12 |   3    |  5  |   w2   |  w3  | 2024-09-22 12:52:37.938593 |
| 25 |   1    |  12 |   w4   |  w5  | 2024-09-22 13:06:29.905453 |
+----+--------+-----+--------+------+----------------------------+
```

8. **Delete material**

```
 Choose an action : 8
Enter the Material_id to delete : 3
Material with id 3 has been deleted successfully
```

9. **Check if reorders required**

```
 Choose an action : 9
All are upto threshold stock
All are upto threshold stock
```

Concepts learned and applied

1. SMTP (Simple Mail Transfer Protocol)

SMTP is a protocol used to send emails from one server to another. In this project, I used SMTP for the **Alert System** to send automated email notifications for shipment delays. I learned how to configure an SMTP server, create an email message using MIME Text, and send it securely with TLS.

2. OOP (Object-Oriented Programming) Concepts

Throughout this project, I applied OOP principles such as:

- **Classes and Objects**: To model various system components (e.g., Inventory, Reorder, Transfer, Forecasting).
- **Encapsulation**: Each class encapsulates its functionality, making the code modular and maintainable.
- **Inheritance**: Although not used explicitly in this project, the concept of inheritance could be extended to enhance the code by creating a base class for shared functionality.

3. Pandas

I used the pandas library to manage and process the inventory data in a tabular format. Pandas is highly efficient in handling large datasets, and it allowed me to:

- **Generate Reports**: By transforming inventory data into DataFrame objects and calculating metrics like inventory turnover and sales forecasts.

4. Authentication

The **Security** class implemented a basic authentication system, where user credentials are stored in a dictionary and checked upon login. This ensures that only authorized personnel can perform sensitive operations like updating stock levels and managing materials.

5. Threading

Python's threading module allows multiple threads to run concurrently, which is useful in scenarios where multiple users need to interact with the system at the same time. Locks allow you to force multiple threads to access a resource one at a time.

6.Exception handling

For catching input errors from users and preventing the crash of the program or exiting.

Also for displaying suitable formats and giving warnings.

Required Modules

To run the project, the following Python modules need to be installed:

1. pandas - For data manipulation and analysis.
2. smtplib - For sending email notifications.
3. email.mime - For constructing email messages.
4. Prettytable- For displaying data in tabular format
5. DateTime – For logging shipments information and tracking.
6. Sqlite3 – For databases and querying.

Conclusion

This inventory management system project has been a great learning of Python programming concepts and various other libraries in python. It helped me strengthen my understanding of OOP, SMTP for sending email alerts, working with the pandas library for data analysis, exception handling.

Submitted by

Saketh Kukkadapu and Borusu Sri Venkata Maheswara Sai Krishna.