Men's Wear Store Inventory Management System

1. Aim of Project:

The objective of this project is to develop a comprehensive inventory management system for a men's wear store. The system will facilitate efficient management of inventory, orders, and financial tracking for the store operations.

Key Features:

1. **Admin and User Authentication:**
   - Differentiate between administrative and user roles with secure login functionality.
2. **Admin Functions:**
   - **Display Menu:** View the current inventory with detailed item information.
   - **Add Item:** Incorporate new items into the inventory with specific details (item number, name, quantity, prices).
   - **Remove Item:** Remove items from the inventory based on item number.
   - **Total Stock Available:** Calculate and display the total quantity of all items currently in stock.
   - **Income and Profit:** Compute and present the overall income and profit based on sales data.
3. **User Functions:**
   - **Display Menu:** Show available items for purchase with their details (item number, name, available quantity, price).
   - **Place Order:** Allow users to place orders for desired items, updating the inventory accordingly.
   - **Cancel Order:** Provide functionality for users to cancel placed orders, adjusting inventory as necessary.
4. **Data Management:**
   - Utilize data structures such as lists of dictionaries to store and manage inventory information dynamically.
   - Implement data validation to ensure accurate and reliable input from users and administrators.
5. **User Interface:**
   - Develop a user-friendly interface with clear prompts and menus for both administrators and users.
   - Ensure robust error handling and informative messages to guide users through the system.
6. **Security and Scalability:**
   - Incorporate secure login mechanisms to protect sensitive data and restrict unauthorized access.
   - Design the system to be scalable, accommodating potential future expansions or modifications.

2. Business Problem Statement:

Managing inventory efficiently is crucial for ensuring smooth operations and maximizing profitability in a men's wear store. The current manual system lacks automation and real-time insights, leading to inefficiencies in inventory tracking, order management, and financial oversight.

Challenges:

Manual Inventory Management:

The store currently relies on manual methods for tracking inventory, which are time-consuming and prone to errors.

Inaccurate Stock Levels:

Difficulty in accurately assessing stock levels results in instances of stockouts or overstocking, impacting customer satisfaction and profitability.

Limited Financial Visibility:

Lack of automated financial reporting and analysis makes it challenging to monitor income, profit margins, and overall financial health accurately.

Customer Experience:

Inconsistent availability of products due to poor inventory management affects customer experience and loyalty.

By addressing these challenges and objectives, the proposed inventory management system aims to transform operations at the men's wear store, paving the way for sustainable growth, enhanced customer satisfaction, and improved profitability.

This problem statement encapsulates the core issues faced by the men's wear store and outlines the desired outcomes and objectives of your project. It provides a clear focus on how developing an effective inventory management system can address these challenges and bring tangible benefits to the business.

3. Project Description:

The "Smart Inventory Management System for Men's Wear Store" project aims to revolutionize the way inventory is managed and monitored in a retail environment specializing in men's apparel. The current manual methods for inventory tracking and order management are inefficient and prone to errors, leading to challenges such as stockouts, overstocking, and inaccurate financial reporting.Here are the functionalities:

1. **Inventory Tracking.**
2. **Order Management System.**
3. **Financial Reporting and Analysis.**
4. **User-Friendly Interface.**
5. **Security and Data Integrity.**

Inventory Tracking:

- Implement automated systems to track real-time inventory levels of various men's wear items including shirts, pants, and t-shirts.
- Utilize data structures to dynamically manage inventory data, ensuring accurate updates and minimizing discrepancies.

Order Management System:

- Develop a streamlined process for customers to place orders, check product availability, and receive order confirmations promptly.

- Include functionalities for order cancellations and adjustments to maintain customer satisfaction and operational efficiency.

Financial Reporting and Analysis:

- Integrate financial reporting capabilities to analyze sales revenue, profit margins, and inventory turnover.
- Provide insights into financial performance to support strategic decision-making and business growth.

User-Friendly Interface:

- Design an intuitive user interface tailored for both store administrators and customers.
- Ensure ease of navigation, clear presentation of product information, and seamless interaction to enhance user experience.

Security and Data Integrity:

- Implement robust security measures to protect sensitive data, ensuring confidentiality and integrity throughout the system.
- Employ encryption techniques and access controls to safeguard customer information and transaction records.

The "Smart Inventory Management System for Men's Wear Store" project aims to transform the operational efficiency and customer service standards of the men's wear retail sector. By leveraging automation, real-time data tracking, and advanced reporting capabilities, the system will empower the store to meet customer demands effectively while optimizing inventory control and financial management. Ultimately, the project seeks to establish a foundation for sustainable growth, profitability, and enhanced customer satisfaction in the competitive retail landscape.

4. Functionalities:

Inventory Management:

- **Real-time Tracking:** Monitor inventory levels of shirts, pants, t-shirts, and other items in real-time to prevent stockouts and overstocking.
- **Automated Updates:** Automatically update inventory counts with each transaction (sales, returns, additions) to maintain accuracy.
- **Inventory Alerts:** Receive alerts for low stock levels to prompt timely reordering and prevent disruptions in supply.

Order Management:

- **Order Placement:** Allow customers to browse available products, select items, and place orders efficiently through a user-friendly interface.
- **Order Confirmation:** Provide immediate order confirmations with details on expected delivery dates and payment information.
- **Order Modifications:** Enable customers to modify or cancel orders within a specified timeframe to accommodate changing needs.

Financial Tracking and Reporting:

- **Sales Analysis:** Generate comprehensive reports on sales performance, including revenue generated by product category, sales trends over time, and comparison with targets.
- **Profit Margin Calculation:** Calculate and analyse profit margins per item and overall, factoring in costs such as wholesale prices and operating expenses.
- **Expense Management:** Track operational expenses related to inventory management, logistics, and store operations to maintain profitability.

User Management:

- **Admin Dashboard:** Provide administrators with access to a centralized dashboard to oversee inventory levels, track orders, and manage customer interactions.
- **Customer Profiles:** Maintain customer profiles with purchase history, preferences, and contact information to personalize service and marketing efforts.
- **Security Measures:** Implement authentication and authorization protocols to protect sensitive data and ensure compliance with data privacy regulations.

Reporting and Analytics:

- **Customizable Reports:** Create customizable reports for administrators to gain insights into inventory turnover, product popularity, and customer purchasing patterns.
- **Data Visualization:** Present data through charts, graphs, and visual dashboards to facilitate quick decision-making and strategic planning.
- **Forecasting:** Use historical data and predictive analytics to forecast demand, optimize inventory levels, and plan promotions effectively.

5. Input Versatility with error handling and exception handling:

By incorporating input versatility, robust error handling, and exception handling into the Smart Inventory Management System, you ensure a more reliable and user-friendly experience for administrators and customers alike. These practices not only improve usability but also contribute to the overall resilience and stability of the system, reducing downtime and enhancing productivity in managing inventory, orders, and financial data in a men's wear retail environment.

6.Code Implementation:

```
mens_wear_stock = [{"Item_no":8000,"Item":"Shirts_H/S","Qty_Available":100,"Max_Retail_Price":495,"Wholesale_Price":300},
                   {"Item_no":8001,"Item":"Shirts_F/S","Qty_Available":115,"Max_Retail_Price":695,"Wholesale_Price":400},

{"Item_no":8002,"Item":"Cottan_Pants","Qty_Available":100,"Max_Retail_Price":995,"Wholesale_Price":600},

{"Item_no":8003,"Item":"Jean_Pants","Qty_Available":90,"Max_Retail_Price":895,"Wholesale_Price":500},
                   {"Item_no":8004,"Item":"T-Shirts","Qty_Available":80,"Max_Retail_Price":345,"Wholesale_Price":250}]
stock_items = mens_wear_stock
update_list = []
order = ""

def Login_Admin():
    print("**********************")
    print("1.Display_Menu")
    print("2.Add_Item")
    print("3.Remove_Item")
    print("4.Total_Stock_Available")
```

```
    print("5.Income_and_Profit")
    print("6.Logout")
    print("**********************")

def Display_Menu_Admin():
    print("Item_no\t\tItem\t\tQty_Available\tMax_Retail_Price\tWholesale_Price")
    print("*"*88)
    for d in mens_wear_stock:
            print(f"{d["Item_no"]}\t\t{d["Item"]}\t{d["Qty_Available"]}\t\t{d["Max_Retail_Price"]}
\t\t\t{d["Wholesale_Price"]}")

def Item_Include():
    n = int(input("Enter the no.of.items need to be added : "))
    for i in range(n):
        new_Item_no = int(input("Enter_Item_No : "))
        new_Item = input("Enter_Item : ")
        new_Qty_Available = int(input("Enter_Qty_Available : "))
        new_Max_Retail_Price = int(input("Enter_Max_Retail_Price : "))
        new_Wholesale_Price = int(input("Enter_Wholesale_Price : "))
                                                                                    d =
[{"Item_no":new_Item_no,"Item":new_Item,"Qty_Available":new_Qty_Available,"Max_Retail_Pri
ce":new_Max_Retail_Price,"Wholesale_Price":new_Wholesale_Price}]
        mens_wear_stock.extend(d)

    Display_Menu_Admin()

def Item_Exclude():
    dress_item_no = int(input("Enter_the_Item_no_need_to_be_deleted : "))
    found = False
    for d in stock_items:
        found = d["Item_no"] == dress_item_no
        if found != True:    #
            update_list.append(d)
            continue
        if found == True:
            d["Qty_Available"] -= dress_item_no
    print("Deleting_Item....")

    if len(update_list) == d: #
        print(f"{dress_item_no} Not_Found")
    else:
        print(f"{dress_item_no} is_available_and_removed_from_the_list")

    Display_Menu_Admin()

def Stock_Update():
```

```
    total_1 = 0
    print("\n")
    for d in mens_wear_stock:
        print(f"{d["Item"]} = {d["Qty_Available"]}")
        total_1 += (d["Qty_Available"])
    print(f"\nTotal_Available_StocK_ is : {total_1}")
def Income_Profit():
    total_2 = 0
    for d in mens_wear_stock:
                        total_2 += ((d["Qty_Available"]*d["Max_Retail_Price"])-
(d["Qty_Available"]*d["Wholesale_Price"]))
    print(f"\nTotal_Income_/_Profit_is : {total_2}")

def Logout():
    Login()


def Admin_Management():
    input_1 = int(input("Please_Enter_Admin_Input : "))
    if input_1 == 1:
        Display_Menu_Admin()
        print("*"*88)
        Login_Admin()
        print("\n*********************************************************\n")
        Admin_Management()
    elif input_1 == 2:
        Display_Menu_Admin()
        print("\n*********************************************************\n")
        Item_Include()
        print("\n*********************************************************\n")
        Login_Admin()
        print("\n*********************************************************\n")
        Admin_Management()
    elif input_1 == 3:
        Display_Menu_Admin()
        print("\n*********************************************************\n")
        Item_Exclude()
        print("\n*********************************************************\n")
        Login_Admin()
        print("\n*********************************************************\n")
        Admin_Management()
    elif input_1 == 4:
        Stock_Update()
        print("\n*********************************************************\n")
        Login_Admin()
```

```
        print("\n**************************************************\n")
        Admin_Management()
    elif input_1 == 5:
        Income_Profit()
        print("\n**************************************************\n")
        Login_Admin()
        print("\n**************************************************\n")
        Admin_Management()
    elif input_1 == 6:
        Logout()
    else:
        print("\nInvalid Choice. Please enter valid choice")
        print("\n**************************************************\n")
        Login_Admin()
        print("\n**************************************************\n")
        Admin_Management()

def Login_User():
    print("**********\n")
    print("1.Display_Menu")
    print("2.Place_Order")
    print("3.Cancel_Order")
    print("4.Logout")
    print("\n**********")

def Display_Menu_User():
            print("Item_no\tItem\tQty_Available\tMax_Retail_Price")
            print("**************************************************")
            for d in mens_wear_stock:
                                    print(f"{d["Item_no"]}\t{d["Item"]}\t{d["Qty_Available"]}
\t{d["Max_Retail_Price"]}")
def User_ID():
    Display_Menu_User()
    p_id = int(input("\nEnter_the_ID : "))

def Place_Order():
    order_number = int(input("Enter_order_qty :"))
    Display_Menu_User()
    p_id = int(input("\nEnter_the_Item_no : "))

    for d in mens_wear_stock:
        if d["Item_no"] == p_id:
            print("\nItem_no\tItem\tQty_Available\tMax_Retail_Price")
            print("**************************************************")
            print(f"{d["Item_no"]}\t{d["Item"]}\t{d["Qty_Available"]}\t\t{d["Max_Retail_Price"]}")
```

```
            order = '{d["Item_no"]}\t{d["Item"]}\t{d["Qty_Available"]}\t\t{d["Max_Retail_Price"]}'
            order_conformation = input("\n Please_confirm_your_order : Yes/No ")

            if order_conformation == 'Yes' or order_conformation == 'yes':
                print(f"\nOrder placed successfully on the product {d["Item_no"]} : {d["Item"]}")
                order_number += 1
                print(f"Your_order_number_is : {order_number}")
                d["Qty_Available"] -= order_number
                break
            elif order_conformation == 'No' or order_conformation == 'no':
                print("Order_not_placed, Thank you")
                break
            else:
                print("\nInvalid_Input,Try_again.\n")
                order_conformation = input("\nPlease_confirm_your_order : Yes/No ")
                break

    if d["Item_no"]!= p_id:
        print("\nYou have entered invalid id. Please enter valid id\n")
        User_ID()
    print("\nAvailable products : \n")
    Display_Menu_User()

def Order_Cancellation():
    found = False
    update_list = []
    order_item_no = input("Enter_the_order_item_no : ")
    for d in mens_wear_stock:
        found = d["Item_no"] == order_item_no
        if found != True:
            update_list.append(d)
    if len(update_list) == d:
        print(f"{order_item_no} is not found")
    else:
        print(f"{order_item_no} is removed from the placed order")

def User_Management():
    input_2 = int(input("Please_enter_user_Input : "))
    if input_2 == 1:
        Display_Menu_User()
        print("\n**************************************************\n")
        Login_User()
        print("\n**************************************************\n")
        User_Management()
    elif input_2 == 2:
```

```
        Place_Order()
        print("\n**************************************************\n")
        Login_User()
        print("\n**************************************************\n")
        User_Management()
    elif input_2 == 3:
        Order_Cancellation()
        print("\n**************************************************\n")
        Login_User()
        print("\n**************************************************\n")
        User_Management()
    elif input_2 == 4:
        Logout()
    else:
        print("Invalid Choice. Please enter valid choice")


def Login():
    login_input = input("Admin/User : ")
    if login_input == 'Admin' or login_input == 'admin' :
        password = input("Enter the password : ")
        if password == "Admin@789":
            Login_Admin()
            Admin_Management()
        else:
            print("Invalid password. Please enter valid password")

    elif login_input == 'User' or login_input == 'user':
        password = input("Enter the password : ")
        if password == "User@456":
            Login_User()
            User_Management()
        else:
            print("Invalid password. Please enter valid password")
    else:
        print("Invalid user type. Enter valid user type")
Login()
```

7. Result and Outcomes:

The implementation of the Smart Inventory Management System for a men's wear store is expected to transform business operations by enhancing efficiency, optimizing inventory management, and improving customer satisfaction. By leveraging automation, data analytics, and user-centric design, the system enables the store to meet customer demands effectively, make informed decisions, and achieve sustainable growth in a competitive retail environment. The anticipated outcomes include improved operational processes, increased profitability, and a stronger customer base, positioning the store for long-term success and profitability.

8. Conclusion:

The Smart Inventory Management System not only addresses current operational challenges but also positions the men's wear store for sustainable growth and competitiveness in the dynamic retail landscape. By leveraging technology to streamline processes, enhance customer interactions, and optimize financial outcomes, the system serves as a strategic asset that drives efficiency, profitability, and customer satisfaction.

In conclusion, the Smart Inventory Management System represents a pivotal investment in the store's future, aligning operational excellence with customer-centric strategies to achieve long-term success and leadership in the men's wear retail sector.