**TAGOLOAN COMMUNITY COLLEGE**

Baluarte, Tagoloan, Misamis Oriental

*Member*: Association of Local Colleges and Universities (ALCU)

*Member: (ALCU - Commission of Accreditation)*

**COLLEGE OF INFORMATION TECHNOLOGY**

*Member: Philippines Society of Information Technology Educators (PSITE)*

**(INVENTORY SYSTEM)**

***Submitted by:***

*Sevilla, Rebecca Mae B.*

*Piloton, Elaiza Mae R.*

*Cabulong, Lyn Nicole S.*

*Sosas, Jhon Vhel H.*

***In partial fulfillment of the course***

***IT 102 Computer Programming 01***

***2023***



**Introduction**

Nowadays, advancement in technology has made the business to grow up more rapidly. Technology can primarily speed up processes, allows flexible new ways of working, transforms how your business functions and can often save you time and money. As life without any contemporary technology is not complete similarly, no business is complete without these technologies.

Items they have available and the location in which they reside. Inventory systems provide detailed records of new and returned products as they're entering or leaving the warehouse to help companies organize and account for their stock. These systems can also track data such as the number of units, cost per unit, serial number, lot numbers, purchase dates and production dates.

The proposed Inventory System aims to lessen the problems encountered in the current system. It will simplify the processing of records, reduce transaction errors, provide safe storage, produce reliable information, search product’s and customer’s records easily, monitor stock movement to avoid miscount of products, and generate accurate sales report. It will also facilitate the adding, updating, and searching of company’s customers regarding their account balances and past transactions. The system also has a web page which can contribute in the promotion of their products and services. Consequently, this system helps to assist in building a lasting customer relationship which in return will result to customer’s satisfaction and customer’s loyalty.

**OBJECTIVES**:

**Optimizing Stock Levels**: Ensuring that there’s enough inventory to meet customer demand without excess, minimizing carrying costs.

**Minimizing Costs**: Order processing costs, and stockouts to enhance overall cost efficiency.

**Improving Customer Service**: Ensuring products are available when customers need them, enhancing customer satisfaction and loyalty.

**Reducing Lead Time**: Streamlining the supply chain to minimize the time it takes for products to move from supplier to customer.

**Enhancing Accuracy**: Implementing systems and processes to reduce errors in inventory tracking, order fulfillment, and record-keeping.

**Managing Obsolescence**: Minimizing the risk of holding obsolete or expired stock through effective product life cycle management.

**Compliance**: Ensuring compliance with regulatory requirements related to inventory, especially in industries with specific storage and handling guidelines.

**Optimizing Space**: Efficiently utilizing storage space to maximize the capacity and minimize storage costs.

**Supply Chain Resilience**: Building a resilient supply chain that can adapt to disruptions, ensuring continuity in product availability.

**Scope And Limitation:**

**Scope:**

**1. Functionality:**

* The system will handle the tracking and management of inventory items.
* It should provide features for adding orders, updating inventory stock, and deleting items from the inventory.
* Generate reports on current inventory status, such as stock levels and product details.

**2. Security:**

* Implement user authentication to ensure that only authorized users can access the system.

**3. Scalability:**

* Design the system to handle a growing number of inventory items and adapt to changing business needs.

**Limitation:**

**1. Hardware Limitations:**

* The system’s performance may be constrained by the hardware on which it is deployed.

It's essential to consider these aspects while developing and implementing an inventory system in Java to ensure it meets the intended requirements while acknowledging its limitations.

Chooses 1 or 8

Display Menu

FLOWCHART

END

Payment

Update Inventory

Place Order

Print Receipt

LOG IN

START

**package InventorySystem7;**

**import java.util.Scanner;**

**public class InventorySystem7 {**

**private static final int INITIAL\_STOCK = 100;**

**public static void main(String[] args) {**

**Scanner input = new Scanner(System.in);**

**String username;**

**String password;**

**int ID;**

**System.out.println("Login ---->");**

**System.out.print("Enter username: ");**

**username = input.nextLine();**

**System.out.print("Enter password: ");**

**password = input.nextLine();**

**System.out.print("Enter ID number: ");**

**ID = input.nextInt();**

**System.out.println("\nWelcome!");**

**System.out.print("\nHello " + username + ".");**

**double total = 0.0;**

**int[] stocks = new int[]{INITIAL\_STOCK, INITIAL\_STOCK, INITIAL\_STOCK, INITIAL\_STOCK, INITIAL\_STOCK, INITIAL\_STOCK, INITIAL\_STOCK, INITIAL\_STOCK};**

**while (true) {**

**System.out.println("\n\nMAIN MENU");**

**System.out.println("[1] Burger");**

**System.out.println("[2] Burger Steak");**

**System.out.println("[3] Chicken");**

**System.out.println("[4] Spaghetti");**

**System.out.println("[5] Chicken Sandwich");**

**System.out.println("[6] Hotdog");**

**System.out.println("[7] Fries");**

**System.out.println("[8] Beverages");**

**System.out.println("\n\n[9] Delete Item");**

**System.out.print("\nEnter your choice: ");**

**int choice = input.nextInt();**

**if (choice == 9) {**

**deleteItem(input, stocks);**

**continue;**

**}**

**if (choice < 1 || choice > 8) {**

**System.out.println("\nInvalid choice. Please try again.");**

**continue;**

**}**

**double itemTotal = processCategory(choice, input, stocks);**

**total += itemTotal;**

**System.out.println("Subtotal: " + itemTotal);**

**System.out.print("\nWould you like to add more, \ndelete an item, or finish ordering? \n(A(ADD)/D(DELETE)/F(FINISH)): ");**

**String continueOrdering = input.next();**

**if (continueOrdering.equalsIgnoreCase("F")) {**

**break;**

**} else if (continueOrdering.equalsIgnoreCase("D")) {**

**deleteItem(input, stocks);**

**}**

**}**

**System.out.println("\n[Payment]");**

**System.out.println("Total: " + total);**

**double payment = processPayment(total, input);**

**updateInventoryStock(stocks);**

**printReceipt(username, ID, total, payment);**

**}**

**public static double processCategory(int category, Scanner scanner, int[] stocks) {**

**double[] prices = {};**

**String[] itemNames = {};**

**switch (category) {**

**case 1:**

**prices = new double[]{32.00, 35.00, 30.00, 38.00, 40.00, 25.00, 25.00, 20.00};**

**itemNames = new String[]{"Burgers", "Burger Steak", "Chicken", "Spaghetti", "Chicken Sandwich", "Hotdog", "Fries", "Beverages"};**

**break;**

**case 2:**

**prices = new double[]{32.00, 35.00, 30.00, 38.00, 40.00, 25.00, 25.00, 20.00};**

**itemNames = new String[]{"Burgers", "Burger Steak", "Chicken", "Spaghetti", "Chicken Sandwich", "Hotdog", "Fries", "Beverages"};**

**break;**

**case 3:**

**prices = new double[]{32.00, 35.00, 30.00, 38.00, 40.00, 25.00, 25.00, 20.00};**

**itemNames = new String[]{"Burgers", "Burger Steak", "Chicken", "Spaghetti", "Chicken Sandwich", "Hotdog", "Fries", "Beverages"};**

**break;**

**case 4:**

**prices = new double[]{32.00, 35.00, 30.00, 38.00, 40.00, 25.00, 25.00, 20.00};**

**itemNames = new String[]{"Burgers", "Burger Steak", "Chicken", "Spaghetti", "Chicken Sandwich", "Hotdog", "Fries", "Beverages"};**

**break;**

**case 5:**

**prices = new double[]{32.00, 35.00, 30.00, 38.00, 40.00, 25.00, 25.00, 20.00};**

**itemNames = new String[]{"Burgers", "Burger Steak", "Chicken", "Spaghetti", "Chicken Sandwich", "Hotdog", "Fries", "Beverages"};**

**break;**

**case 6:**

**prices = new double[]{32.00, 35.00, 30.00, 38.00, 40.00, 25.00, 25.00, 20.00};**

**itemNames = new String[]{"Burgers", "Burger Steak", "Chicken", "Spaghetti", "Chicken Sandwich", "Hotdog", "Fries", "Beverages"};**

**break;**

**case 7:**

**prices = new double[]{32.00, 35.00, 30.00, 38.00, 40.00, 25.00, 25.00, 20.00};**

**itemNames = new String[]{"Burgers", "Burger Steak", "Chicken", "Spaghetti", "Chicken Sandwich", "Hotdog", "Fries", "Beverages"};**

**break;**

**case 8:**

**prices = new double[]{32.00, 35.00, 30.00, 38.00, 40.00, 25.00, 25.00, 20.00};**

**itemNames = new String[]{"Burgers", "Burger Steak", "Chicken", "Spaghetti", "Chicken Sandwich", "Hotdog", "Fries", "Beverages"};**

**break;**

**}**

**System.out.println(itemNames[0] + " Sub-Menu:");**

**for (int i = 0; i < prices.length; i++) {**

**System.out.println("[" + (i + 1) + "] " + itemNames[i] + " : " + prices[i] + " - Stock: " + stocks[i]);**

**}**

**System.out.print("\nEnter your choice again: ");**

**int choice = scanner.nextInt();**

**if (choice < 1 || choice > prices.length) {**

**System.out.println("\nInvalid choice. Please try again.");**

**return 0.0;**

**}**

**System.out.print(itemNames[choice - 1] + " - Qty: ");**

**int qty = scanner.nextInt();**

**if (qty > stocks[choice - 1]) {**

**System.out.println("\nInsufficient stock. Please try again.");**

**return 0.0;**

**}**

**stocks[choice - 1] -= qty;**

**double itemTotal = prices[choice - 1] \* qty;**

**return itemTotal;**

**}**

**public static double processPayment(double total, Scanner scanner) {**

**double payment;**

**while (true) {**

**System.out.print("\nEnter Payment (Cash): ");**

**payment = scanner.nextDouble();**

**if (payment >= total) {**

**double change = payment - total;**

**System.out.println("Change: " + change);**

**System.out.println("\nThank you for Ordering. Enjoy!");**

**return payment;**

**} else {**

**System.out.println("\nInsufficient Cash!!!");**

**System.out.println("Please Try Again!");**

**}**

**}**

**}**

**public static void updateInventoryStock(int[] stocks) {**

**System.out.println("\n===== UPDATED INVENTORY STOCK =====");**

**for (int i = 0; i < stocks.length; i++) {**

**System.out.println("[" + (i + 1) + "] Stock for Item: " + stocks[i]);**

**}**

**System.out.println("===============================");**

**}**

**public static void printReceipt(String username, int ID, double total, double payment) {**

**double change = payment - total;**

**System.out.println("\n===== RECEIPT =====");**

**System.out.println("Username: " + username);**

**System.out.println("ID Number: " + ID);**

**System.out.println("Total: " + total);**

**System.out.println("Payment: " + payment);**

**System.out.println("Change: " + change);**

**System.out.println("Thank you for Ordering with us. Enjoy!");**

**System.out.println("============================");**

**}**

**public static void deleteItem(Scanner scanner, int[] stocks) {**

**System.out.println("\nDELETE ITEM MENU");**

**System.out.println("\nSelect item to delete:");**

**// Display items**

**for (int i = 0; i < stocks.length; i++) {**

**System.out.println("[" + (i + 1) + "] Stock for Item: " + stocks[i]);**

**}**

**System.out.print("\nEnter item number to delete: ");**

**int itemToDelete = scanner.nextInt();**

**if (itemToDelete < 1 || itemToDelete > stocks.length) {**

**System.out.println("\nInvalid item number. Please try again.");**

**return;**

**}**

**// Confirm deletion**

**System.out.print("\nAre you sure you want to delete item " + itemToDelete + "? (Y/N): ");**

**String confirmDelete = scanner.next();**

**if (confirmDelete.equalsIgnoreCase("Y")) {**

**stocks[itemToDelete - 1] = INITIAL\_STOCK;**

**System.out.println("\nItem " + itemToDelete + " deleted successfully. Stock reset to initial value.");**

**} else {**

**System.out.println("\nDeletion canceled.");**

**}**

**}**

**}**