INTRODUCTION

The POS (Point of Sale) system is a software application designed for restaurants to facilitate the management of orders, inventory, and billing. This system allows cashiers to efficiently take customer orders, calculate totals, apply discounts for senior citizens and PWDs, and print receipts. Additionally, it provides restricted access to inventory management for authorized employees. The program is implemented in C++ and provides a simple console-based interface for ease of use.

STATEMENT OF THE PROBLEM

Restaurants often face challenges in managing orders, inventory, and billing efficiently. Manual systems can result in:

1. Miscalculations in the total bill.
2. Difficulty tracking inventory and stock levels.
3. Inaccurate application of discounts.
4. Time-consuming receipt generation.

This system addresses these problems by automating the order-taking, stock management, discount application, and receipt printing processes.

OBJECTIVES

General Objective:

- To develop a console-based POS system for restaurants that improves efficiency in order processing, inventory management, and billing.

Specific Objectives:

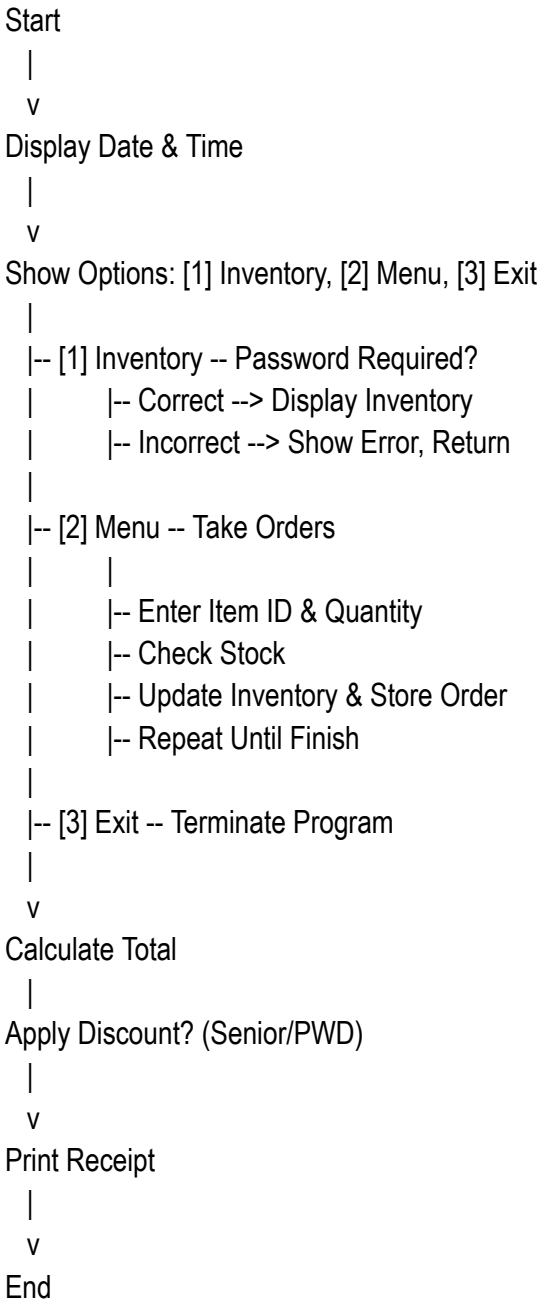
1. Allow cashiers to view the menu and process customer orders.
2. Enable the system to calculate totals and apply discounts automatically.

3. Print receipts that include order details, totals, discounts, and timestamps.

4. Restrict inventory access to authorized employees through a password system.

5. Maintain stock levels and alert when insufficient inventory is available.

FLOWCHART OF THE SYSTEM



PSEUDOCODE

```
PROGRAM RestaurantPOS
DECLARE Menu[], Inventory[], Orders[]
DISPLAY current Date & Time

WHILE true
   DISPLAY options: Inventory, Menu, Exit
   INPUT option

   IF option == 1 THEN
      CALL AccountLogin()
      IF login successful THEN
         DISPLAY Inventory
      ELSE
         DISPLAY "Incorrect Password"
      ENDIF
   ELSE IF option == 2 THEN
      DISPLAY Menu
      WHILE true
         INPUT ItemID
         IF ItemID == 5 THEN
            BREAK
         ENDIF
         INPUT Quantity
         IF Stock sufficient THEN
            ADD order
            UPDATE Inventory
         ELSE
            DISPLAY "Insufficient stock"
         ENDIF
      ENDWHILE
      CALCULATE total
      INPUT if customer is Senior/PWD
      APPLY discount if applicable
      PRINT receipt
   ELSE IF option == 3 THEN
      BREAK
   ELSE
      DISPLAY "Invalid Option"
   ENDIF
```

```
ENDWHILE

DISPLAY "System Closed"
END PROGRAM
```

DATA DICTIONARY

Variable	Type	Description
MenuItem	struct	Stores menu item details (ID, name, price, stock)
MenuItemInv	struct	Stores inventory item details
Order	struct	Stores order details (Item ID, Quantity)
menu	MenuItem[]	Array of available menu items
inv	MenuItemInv[]	Array of inventory items
orders	Order[]	Array of current customer orders
orderCount	int	Number of items ordered
cashier	string	Name of cashier
total	double	Total amount before discount
discount	double	Discount amount for Senior/PWD

Code:

```
#include <iostream>
#include <iomanip>
#include <string>
#include <ctime>
using namespace std;

struct MenuItem {
    int id;
    string name;
    double price;
    int stock;

};
struct MenuItemInv {
    int id;
    string name;
    double price;
    int stock;
};

struct Order {
    int itemID;
    int quantity;
};

void displayDateTime() {
    time_t now = time(0);
    char* dt = ctime(&now);
    cout << "Date & Time: " << dt;
}

void displayMenu(MenuItem menu[], int menuSize) {
    cout << "\n========== MENU ================\n";
    cout << left << setw(5) << "ID" << setw(20) << "Item" << setw(10) << "Price" << endl;
    cout << "--------------------------------\n";
    for (int i = 0; i < menuSize; i++) {
        cout << left << setw(5) << menu[i].id << setw(20) << menu[i].name
             << "₱" << fixed << setprecision(3) << setprecision(3)<< menu[i].price << "\t"<< endl;
    }
    cout << "==================================\n";
}
```

```
void displayMenuInv(MenuItemInv Inv[], int menuSizeInv) {
    cout << "\n========== Inventory ====================\n";
    cout << left << setw(5) << "ID" << setw(20) << "Item" << setw(10) << "Price" << setw(10) <<
"Stocks" << endl;
    cout << "-----------------------------------------\n";
    for (int i = 0; i < menuSizeInv; i++) {
        cout << left << setw(5) << Inv[i].id << setw(20) << Inv[i].name
             << "₱" << fixed << setprecision(3) << Inv[i].price << setprecision(3) << "\t" <<Inv[i].stock
<< endl;
    }
    cout << "=========================================\n";
}

double calculateTotal(Order orders[], int orderCount, MenuItem menu[], int menuSize) {
    double total = 0;
    for (int i = 0; i < orderCount; i++) {
        int id = orders[i].itemID;
        int qty = orders[i].quantity;
        for (int j = 0; j < menuSize; j++) {
            if (menu[j].id == id) {
                total += menu[j].price * qty;
                break;
            }
        }
    }
    return total;
}

double applyDiscount(double total, bool isSenior, bool isPWD) {
    double discount = 0.0;
    if (isSenior || isPWD) discount = total * 0.20;
    return discount;
}

void printReceipt(string cashier, Order orders[], int orderCount, MenuItem menu[], int menuSize,
double total, double discount) {
    cout << "\n\n======= RESTAURANT NAME =======\n";
    displayDateTime();
    cout << "Cashier: " << cashier << endl;
    cout << "---------------------------------\n";
    cout << left << setw(20) << "Item" << setw(10) << "Qty" << setw(10) << "Price" << endl;
    cout << "---------------------------------\n";

    for (int i = 0; i < orderCount; i++) {
```

```
        int id = orders[i].itemID;
        int qty = orders[i].quantity;
        for (int j = 0; j < menuSize; j++) {
            if (menu[j].id == id) {
                cout << left << setw(20) << menu[j].name
                    << setw(10) << qty
                    << "₱" << fixed << setprecision(2) << menu[j].price * qty << endl;
                break;
            }
        }
    }

    cout << "---------------------------------\n";
    cout << "Subtotal: ₱" << fixed << setprecision(2) << total << endl;
    cout << "Discount: ₱" << discount << endl;
    cout << "TOTAL:    ₱" << total - discount << endl;
    cout << "=================================\n";
    cout << "Thank you for dining with us!\n";
}
bool account(){
        int pass;
        cout << "Please enter your account password: ";
        cin >> pass;
        if (pass == 1196){
            cout << "Welcome Admin!" << endl;
            return true;
        }
         else {
            (pass != 1196);
            cout << "Incorrect password!" << endl;
            return false;
        }
    }
int main() {
    int menuSize = 5;
    MenuItemInv inv[5] = {
        {0, "Burger", 120.00, 100},
        {1, "Fries", 60.000, 100},
        {2, "Soda", 40.000, 100},
        {3, "Chicken Meal", 180.00, 100},
        {4, "Pasta", 150.00, 100}
    };

     MenuItem menu[5] = {
```

```
        {0, "Burger", 120.00, 100},
        {1, "Fries", 60.000, 100},
        {2, "Soda", 40.000, 100},
        {3, "Chicken Meal", 180.00, 100},
        {4, "Pasta", 150.00, 100}
    };
    int opt;
    string cashier;
    Order orders[50];
    int orderCount = 0;
    int choice, qty;
    char seniorChoice, pwdChoice;
    bool isSenior = false, isPWD = false;

    cout << "======= RESTAURANT NAME POS SYSTEM =======\n";
    displayDateTime();

    while (true){
         cout << "Select an option: \n1. Inventory(employee only) \n2. Menu \n3. Exit \nEnter a
number: ";
        cin >> opt;
        if (opt == 1){
            if(account()){
            displayMenuInv(inv, menuSize);
            }
        }

        else if(opt == 2){
        displayMenu(menu, menuSize);
    while (true){
        cout << "\nEnter Item ID (5 to finish): ";
        cin >> choice;

        if (choice == 5) break;

        bool found = false;
        for (int i = 0; i < menuSize; i++) {
            if (inv[i].id == choice) {
                cout << "Enter Quantity: ";
                cin >> qty;

                orders[orderCount].itemID = choice;
                orders[orderCount].quantity = qty;
                 if (inv[choice].stock < qty){
```

```
                cout << "Sorry, we do not have enough stock for your order." << endl;
                orderCount--;
                inv[i].stock += qty;
            }
            inv[i].stock -= qty;
            orderCount++;
            found = true;
            break;

        }

    }

    if (!found) cout << "Invalid Item ID! Try again.\n";
  }

  if (orderCount == 0) {
    cout << "\nNo items ordered.\n";

  }else {
    double total = calculateTotal(orders, orderCount, menu, menuSize);
    cout << "\nIs the customer a Senior Citizen? (Y/N): ";
    cin >> seniorChoice;
    cout << "Is the customer a PWD? (Y/N): ";
    cin >> pwdChoice;

    if (toupper(seniorChoice) == 'Y') isSenior = true;
    if (toupper(pwdChoice) == 'Y') isPWD = true;

    double discount = applyDiscount(total, isSenior, isPWD);
    printReceipt(cashier, orders, orderCount, menu, menuSize, total, discount);
          }
       }else if(opt == 3){
        break;
    }else{
        cout << "Invalid Number!" << endl;
    }
}
  cout << "\nSystem Closed. Goodbye!\n";
  return 0;

}
```