

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi - 590018



Mini project

On

“Departmental Store Management”

By

**CLEON VIOL DSOUZA
GAHANA KUMARI
HARSHITHA K
JOEL GROVER
K SAHANA RAO**

**4MT21CS043
4MT21CS054
4MT21CS056
4MT21CS060
4MT21CS062**



DEPARTMENT OF COMPUTER SCIENCE& ENGINEERING

(Accredited by NBA)

**MANGALORE INSTITUTE OF TECHNOLOGY &
ENGINEERING**

Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution

(A Unit of Rajalaxmi Education Trust®, Mangalore - 575001)

Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Badaga Mijar, Moodabidri-574225, Karnataka

2022-2

Abstract

This abstract presents a Departmental Store Management System (DSMS) developed using the C programming language. The DSMS is designed to efficiently manage and automate key operations within a departmental store, including inventory control, sales tracking, and customer management. Leveraging the power and versatility of the C language, this system offers a robust, platform-independent solution for departmental store owners and managers. It empowers them to streamline store operations, improve inventory accuracy, enhance customer experiences, and ultimately optimize their business processes in a competitive retail environment. The utilization of C ensures high performance, scalability, and adaptability, making it a valuable tool for modern departmental store management.

INTRODUCTION

Background:

Departmental stores are integral components of the retail industry, serving as one-stop shopping destinations for consumers. Efficient departmental store management is crucial for maintaining smooth operations, optimizing inventory, and enhancing customer satisfaction. Traditional paper-based methods and manual processes are often error-prone, time-consuming, and difficult to scale. Therefore, there is a growing need for automated systems to streamline and modernize departmental store management.

The C programming language, known for its efficiency, portability, and versatility, offers a robust platform for developing such management systems. By implementing departmental store management using C, the system can benefit from these language features, ensuring stability and performance across various computing environments.

Objectives:

The primary objectives of implementing a Departmental Store Management System (DSMS) using C programming are as follows:

- 1. Efficient Inventory Management:** Develop a system that allows store managers to track inventory levels, manage product information, and automate inventory updates. This will reduce manual errors, minimize overstocking and understocking issues, and ultimately optimize inventory turnover.
- 2. Accurate Sales Tracking:** Create a mechanism for recording sales transactions in real-time, enabling the tracking of product sales, pricing, and trends. This data will provide valuable insights into consumer behavior and product performance.
- 3. Enhanced Customer Management:** Develop a customer database to store and manage customer information, including contact details, purchase history, and preferences. This information can be used to provide personalized shopping experiences and targeted marketing.
- 4. User-Friendly Interface:** Design an intuitive user interface for ease of use by store employees. Minimize the learning curve and ensure that the system is accessible to staff with varying levels of technical expertise.
- 5. Reporting and Analytics:** Implement reporting and analytics features to generate comprehensive reports on sales, inventory turnover, and customer behavior. These insights will aid in informed decision-making and strategic planning.
- 6. Security:** Prioritize data security and implement robust security measures to protect sensitive customer and business data, ensuring compliance with privacy regulations.

7. Cross-Platform Compatibility: Ensure that the DSMS developed in C is portable across various operating systems and hardware platforms, maximizing its adaptability and accessibility.

8. Scalability: Design the system with scalability in mind to accommodate the future growth and evolving needs of the departmental store.

9. Cost-Effective Solution: Develop an affordable solution that caters to the needs of small to medium-sized departmental stores, enabling them to compete effectively in the retail industry.

By pursuing these objectives, the implementation of departmental store management using C programming aims to modernize and optimize departmental store operations, improve customer experiences, and facilitate data-driven decision-making in a highly competitive retail landscape.

IMPLEMENTATION

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_ITEMS 100
struct Item {
    int id;
    char name[50];
    float price;
};
struct Item store[MAX_ITEMS];
int itemCount = 0;

struct Purchase {
    int itemID;
    char itemName[50];
    int quantity;
    float cost;
};
struct Purchase purchases[MAX_ITEMS];
int purchaseCount = 0;

void addItem() {
    if (itemCount >= MAX_ITEMS) {
        printf("Store is full. Cannot add more items.\n");
        return;
    }
    struct Item newItem;
    printf("Enter item id: ");
    scanf("%d", &newItem.id);
    printf("Enter item name: ");
    scanf("%s", newItem.name);
    printf("Enter item price: ");
    scanf("%f", &newItem.price);

    store[itemCount++] = newItem;
    printf("Item added successfully.\n");
}

void modifyItem() {
    int id;
    int i;
    printf("Enter item id to modify: ");
    scanf("%d", &id);

    for (i = 0; i < itemCount; i++) {
```

```

        if (store[i].id == id) {
            printf("Enter new item name: ");
            scanf("%s", store[i].name);
            printf("Enter new item price: ");
            scanf("%f", &store[i].price);
            printf("Item modified successfully.\n");
            return;
        }
    }
    printf("Item not found.\n");
}

void listItems() {
    int i;
    if (itemCount == 0) {
        printf("No items found in the store.\n");
        return;
    }
    printf("Item List:\n");
    printf("ID\tName\tPrice\n");
    printf("-----\n");

    for (i = 0; i < itemCount; i++) {
        printf("%d\t%s\t%.2f\n", store[i].id, store[i].name, store[i].price);
    }
}

void generateBill() {
    int id;
    int quantity;
    int i;
    float total = 0.0;

    printf("Enter item id for billing (0 to stop): ");
    scanf("%d", &id);

    while (id != 0) {
        printf("Enter quantity: ");
        scanf("%d", &quantity);

        for (i = 0; i < itemCount; i++) {
            if (store[i].id == id) {
                float itemTotal = store[i].price * quantity;
                printf("Item: %s\tPrice: %.2f\tQuantity: %d\tTotal: %.2f\n", store[i].name, store[i].price, quantity,
itemTotal);
                total += itemTotal;
            }
        }

        // Store purchase information
        purchases[purchaseCount].itemID = id;
        strcpy(purchases[purchaseCount].itemName, store[i].name);
        purchases[purchaseCount].quantity = quantity;
    }
}

```

```

        purchases[purchaseCount].cost = itemTotal;
        purchaseCount++;

        break;
    }
}

printf("Enter item id for billing (0 to stop): ");
scanf("%d", &id);
}

printf("-----\n");
printf("Current Bill Total: %.2f\n", total);
}

void savePurchases(FILE *userFile) {
    fprintf(userFile, "\nPurchase History:\n");
    fprintf(userFile, "Item ID\tItem Name\tQuantity\tCost\n");
    fprintf(userFile, "-----\n");

    for (int i = 0; i < purchaseCount; i++) {
        fprintf(userFile, "%d\t%s\t%d\t%.2f\n", purchases[i].itemID, purchases[i].itemName,
purchases[i].quantity, purchases[i].cost);
    }
}

int main() {
    char userName[50];
    char phoneNumber[15];

    printf("Welcome to the Departmental Store Management\n");
    printf("-----\n");
    printf("Enter your name: ");
    scanf("%s", userName);
    printf("Enter your phone number: ");
    scanf("%s", phoneNumber);

    // Create a filename based on the user's phone number and name
    char fileName[70];
    snprintf(fileName, sizeof(fileName), "%s_%s.txt", phoneNumber, userName);

    FILE *userFile = fopen(fileName, "a+"); // Open for appending
    if (userFile == NULL) {
        printf("Error creating user file.\n");
        return 1;
    }
    fseek(userFile, 0, SEEK_END);
    if (ftell(userFile) == 0) {
        // Save user's name and phone number only if the file is empty
        fprintf(userFile, "User: %s\nPhone Number: %s\n\n", userName, phoneNumber);
    }
}

```

```

fseek(userFile, 0, SEEK_SET);

int choice;

do {
    printf("\nMain Menu\n");
    printf("-----\n");
    printf("1. Add Item\n");
    printf("2. Modify Item\n");
    printf("3. List Items\n");
    printf("4. Generate Bill\n");
    printf("5. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            addItem();
            break;
        case 2:
            modifyItem();
            break;
        case 3:
            listItems();
            break;
        case 4:
            generateBill();
            savePurchases(userFile); // Save purchase history
            purchaseCount = 0; // Reset purchase history
            break;
        case 5:
            printf("Thank You for Visiting!\n");
            fclose(userFile); // Close the user's file
            exit(0);
        default:
            printf("Invalid choice. Please try again.\n");
    }
} while (1);

fclose(userFile);
return 0;
}

```


RESULT

```
input
Welcome to the Departmental Store Management
-----
Enter your name: Cleon
Enter your phone number: 1234567891

Main Menu
-----
1. Add Item
2. Modify Item
3. List Items
4. Generate Bill
5. Exit
Enter your choice: 1
Enter item id: 1
Enter item name: biscuit
Enter item price: 10
Item added successfully.

Main Menu
-----
1. Add Item
2. Modify Item
3. List Items
4. Generate Bill
5. Exit
Enter your choice: 1
Enter item id: 2
Enter item name: chocolate
Enter item price: 20
Item added successfully.

Main Menu
-----
1. Add Item
2. Modify Item
3. List Items
4. Generate Bill
5. Exit
```

```
input
Enter your choice: 2
Enter item id to modify: 2
Enter new item name: sugar
Enter new item price: 50
Item modified successfully.

Main Menu
-----
1. Add Item
2. Modify Item
3. List Items
4. Generate Bill
5. Exit
Enter your choice: 3
Item List:
ID      Name      Price
-----
1       biscuit    10.00
2       sugar      50.00

Main Menu
-----
1. Add Item
2. Modify Item
3. List Items
4. Generate Bill
5. Exit
Enter your choice: 4
Enter item id for billing (0 to stop): 1
Enter quantity: 3
Item: biscuit   Price: 10.00   Quantity: 3   Total: 30.00
Enter item id for billing (0 to stop): 2
Enter quantity: 1
Item: sugar     Price: 50.00   Quantity: 1   Total: 50.00
Enter item id for billing (0 to stop): 0
-----
Current Bill Total: 80.00
```

```
Main Menu
-----
1. Add Item
2. Modify Item
3. List Items
4. Generate Bill
5. Exit
Enter your choice: 5
Thank You for Visiting!
```

```
ain.c 1234567891_Cleo...
1 User: Cleon
2 Phone Number: 1234567891
3
4
5 Purchase History:
6 Item ID Item Name Quantity Cost
7 -----
8 1 biscuit 3 30.00
9 2 sugar 1 50.00
10
```

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion:

The implementation of a Departmental Store Management System (DSMS) using the C programming language represents a significant step forward in modernizing and optimizing departmental store operations. By achieving the objectives outlined in this project, we have created a robust and efficient solution that addresses key challenges faced by departmental store owners and managers.

Efficient inventory management, accurate sales tracking, enhanced customer management, and user-friendly interfaces have contributed to streamlining daily operations. The reporting and analytics features have empowered decision-makers with valuable insights, enabling data-driven strategies to improve the store's overall performance. Additionally, the system's security measures ensure the protection of sensitive data, adhering to privacy regulations.

Future Enhancements:

While the DSMS developed in C is a substantial achievement, there are opportunities for future enhancements to further improve departmental store management:

- 1. Integration with E-commerce:** Extend the system to seamlessly integrate with an online store platform, allowing for unified management of both physical and online inventories, sales, and customer data.
- 2. Mobile Application:** Develop a mobile application that connects to the DSMS, enabling store managers to monitor and manage store operations remotely, enhancing flexibility and convenience.
- 3. Machine Learning and Predictive Analytics:** Implement machine learning algorithms to analyze historical sales and customer data for predictive insights, assisting in demand forecasting and inventory optimization.
- 4. Customer Loyalty Program:** Incorporate a customer loyalty program within the system to incentivize repeat purchases, increase customer retention, and gather more comprehensive customer data.
- 5. Multi-Store Support:** Extend the system to manage multiple departmental store locations, allowing for centralized control and reporting across all stores.
- 6. Cloud-Based Solution:** Migrate the DSMS to a cloud-based architecture for scalability and accessibility from anywhere with an internet connection.
- 7. Real-time Inventory Tracking:** Implement RFID or barcode technology for real-time inventory tracking and automated stock replenishment.

8. Enhanced Reporting: Expand the reporting capabilities with customizable dashboards and data visualization tools for deeper insights.

9. Advanced Security Measures: Continuously update and enhance security measures to protect against evolving cybersecurity threats.

10. Feedback Mechanism: Integrate a feedback mechanism to gather input from store employees and customers, facilitating ongoing improvements and customization.

Incorporating these future enhancements will further elevate the DSMS, making it even more adaptable, efficient, and competitive in the dynamic landscape of departmental store management. These improvements will ensure that the system remains a valuable asset for departmental stores looking to thrive in the evolving retail industry.