POINT OF SALE SYSTEM

COURSE PROJECT REPORT

OF

CST 201 - DATA STRUCTURES

Submitted by

NANDANA RAMACHANDRAN - ASI23CA046
NAVNEETH SANTHOSH - ASI23CA047
NIMA FATHIMA - ASI23CA048
PAVITHRA DEEPU E - ASI23CA049
PRITHVI P - ASI23CA050

Under the guidance of

Ms. MEENATCHI K V

To
APJ Abdul Kalam Technological University
In partial fulfillment of the requirements for the award of the
Degree of
Bachelor of Technology in Computer Science and Engineering
(Artificial Intelligence)

Department of Computer and Engineering Science
(Artificial Intelligence)

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY
KALADY

NOVEMBER 2024

POINT OF SALE SYSTEM

COURSE PROJECT REPORT
OF
CST 201 - DATA STRUCTURES
Submitted by

NANDANA RAMACHANDRAN - ASI23CA046
NAVNEETH SANTHOSH - ASI23CA047
NIMA FATHIMA - ASI23CA048
PAVITHRA DEEPU E - ASI23CA049
PRITHVI P - ASI23CA050

Under the guidance of

Ms. MEENATCHI K V

To

APJ Abdul Kalam Technological University
In partial fulfillment of the requirements for the award of the Degree of
Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence)

Department of Computer Science and Engineering (Artificial Intelligence)

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY, KALADY

NOVEMBER 2024

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY, KALADY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE)

CERTIFICATE

Certified that this is a bonafide record of the course project entitled

POINT OF SALE SYSTEM

Submitted by

NANDANA RAMACHANDRAN - ASI23CA046
NAVNEETH SANTHOSH - ASI23CA047
NIMA FATHIMA - ASI23CA048
PAVITHRA DEEPU E - ASI23CA049
PRITHVI P - ASI23CA050

during the year 2024-25 in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology in Computer Science (Artificial Intelligence) and Engineering

Faculty in Charge

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to all those who contributed to the successful completion of this course project. This endeavor was a collaborative effort, and we are sincerely thankful for the support, guidance, and assistance provided by various individuals and resources.

First and foremost, we extend our appreciation to **Prof. Meenatchi K V**, whose expertise and unwavering support played a pivotal role in shaping our project. Their valuable insights and constructive feedback were instrumental in refining our ideas and enhancing the overall quality of our work.

We would also like to thank our fellow classmates and team members for their dedication and hard work throughout the project. Each member brought a unique set of skills and perspectives, contributing to the synergy that defined our group.

Furthermore, we are grateful to **Adi Shankara Institute of Engineering And Technology, Kalady Department Of Computer Science And Engineering (Artificial Intelligence)** for providing us with the necessary resources and a conducive learning environment. The facilities and access to research materials greatly facilitated our project development. In conclusion, this course project would not have been possible without the collective effort and support of everyone involved.

VISION AND MISSION OF THE DEPARTMENT

VISION

To be in the frontier of AI technology through quality of education, collaborative research and produce globally competitive, industry ready engineers with social commitment.

MISSION

M1: Achieve excellence in the educational experience, fostering collaborative research through state-of-the-art infrastructure and innovative elements.

M2: Establish industry collaboration to address interdisciplinary challenges across diverse applications.

M3: Inspire students to develop into ethical, Innovative and entrepreneurial leaders through a socially-centered program

CONTENTS

CHAPTER 1
INTRODUCTION

The Point of Sale (POS) system is an essential tool for managing sales transactions in supermarkets, retail stores, and various customer-facing businesses. This project presents a simple POS system tailored to simulate a supermarket checkout process. It is designed to assist cashiers in adding items to a cart, generating bills, managing stock, and updating inventory efficiently.

This POS system uses fundamental data structures like arrays and implements binary search for optimized item lookup. It allows the cashier to:

1. Add items to a customer's cart.
2. Complete the checkout process by calculating the total bill.
3. View the current stock inventory.
4. Add new items to the stock or update quantities of existing items.

The POS system is developed in **C programming language** using arrays for stock and cart management, and **binary search** for efficient item lookup. This project demonstrates how data structures like arrays, combined with efficient search algorithms, can be applied to solve practical problems in retail management.

CHAPTER 2
OBJECTIVES

The primary objectives of this project are:

1. **Efficient Item Management**: Enable easy addition of items to the cart and calculation of bills while reducing stock quantities accordingly.
2. **Inventory Management**: Allow the cashier to view and update inventory in real time, ensuring accurate stock levels.
3. **Binary Search Optimization**: Use binary search to quickly locate items within stock, improving the system's response time and scalability.

CHAPTER 3
PROGRAM DESIGN

The POS (Point of Sale) System program is intended to simulate a checkout process for a supermarket, allowing cashiers to add items to a cart, view stock, process checkout, and manage inventory. It is implemented in the C programming language, using modular design principles to enhance code readability and maintainability.

1.Data Structure

The program uses a struct named `Item` to store details of individual stock items, including the item name, price, and quantity available. An array of Item structures represents the store's inventory, while an array of CartItem structures is used to store items added to a customer's cart during the checkout process.

Key structures:

- **Item**: Stores item name, price, and quantity in stock.
- **CartItem**: Stores item name and quantity added to the cart.

2. Functionality

The program is composed of several primary functions, each responsible for a specific task:

1. **addStock**: This function allows the cashier to add new items to the inventory or update quantities of existing items. If an item already exists, it updates its quantity; otherwise, it adds a new item to the stock.
2. **displayStock**: This function displays all items currently in stock, along with their prices and available quantities. It informs the user if the inventory is empty.
3. **addToCart**: This function allows the cashier to add items to the customer's cart. It verifies if the item is available in stock and whether the requested quantity is sufficient before adding it to the cart.
4. **checkout**: This function calculates the total price of items in the cart, displays a bill, and reduces the stock quantities for the purchased items. It also clears the cart after processing the checkout.
5. **searchStock**: This function allows the cashier to search for a specific item in the stock by name. It displays the item details if found.

3. User Interface

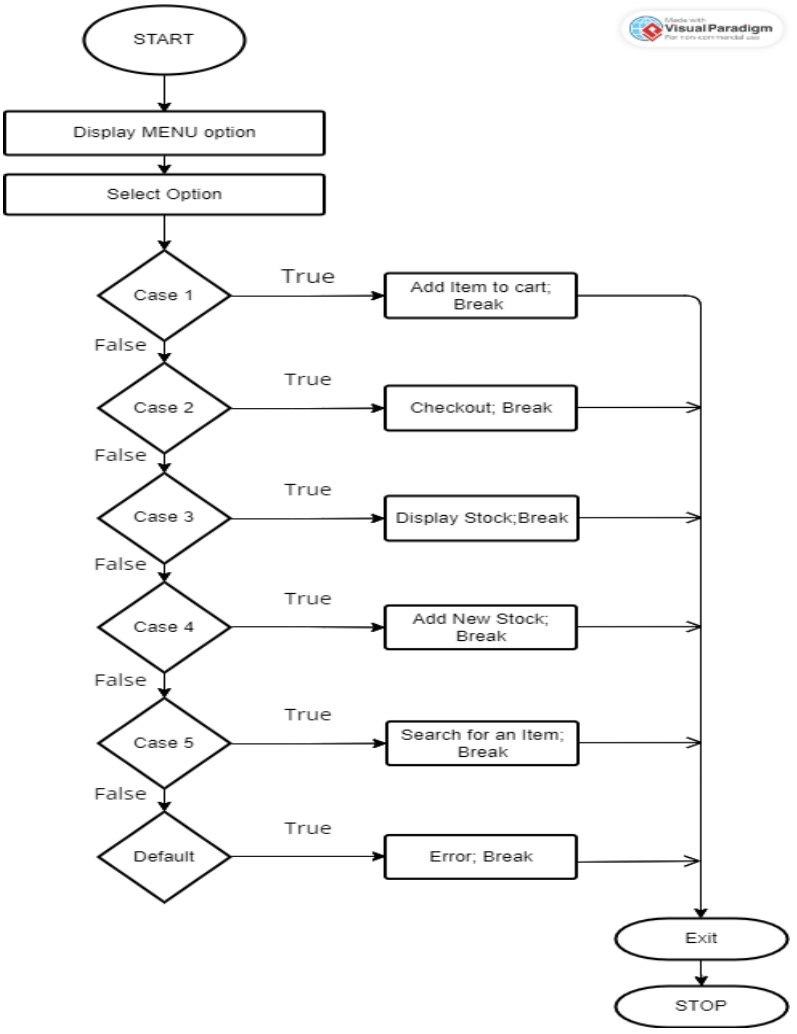
The program interacts with the user through a command-line interface with a menu-driven format. Each feature of the POS system is mapped to a unique menu option. The cashier can select options by entering the corresponding number, making the interface intuitive and easy to navigate.

4. Flow of Execution

Upon running the program, the user is prompted to enter the initial inventory size. The main menu is then displayed, providing options to manage stock, add items to the cart, view inventory, and perform checkout operations. The program continues to execute in a loop until the cashier chooses the exit option.

CHAPTER 4
FLOWCHART AND PROGRAM LOGIC

Flowchart



1. **Initialize** stock items with names, prices, and quantities.
2. Start an infinite loop with the menu displayed:
   - Display menu options:
     - 1: Add item to cart
     - 2: Checkout
     - 3: Display stock
     - 4: Add new stock
     - 5: Search for an item
     - 6: Exit
3. Based on the selected option:

- **Option 1 (Add to Cart)**:

    - Prompt for item name and quantity.
    - Search for the item in stock.
    - If found and quantity is available:
        - Add item to the cart.
        - Display success message.
    - If not found or insufficient stock:
        - Display error message.
- **Option 2 (Checkout)**:
    - Calculate the total bill of items in the cart.
    - Display each item, quantity, price, and total cost.
    - Deduct quantities from stock.
    - Clear the cart after checkout.
- **Option 3 (Display Stock)**:
    - Print all stock items with name, price, and quantity.
- **Option 4 (Add New Stock)**:
    - Prompt for item name, price, and quantity.
    - Search for the item in stock.
    - If found:
        - Update quantity.
    - If not found:
        - Add a new item to the stock list.
- **Option 5 (Search Item)**:
    - Prompt for item name.
    - Search and display details if found; show error if not found.
- **Option 6 (Exit)**:
    - Terminate the program.

PROGRAM LOGIC

This POS system allows the cashier to:

1. Add items to the cart: The cashier can enter an item name and quantity, which are then added to the cart if available in stock.
2. Checkout: Calculate the total bill of items in the cart and update stock quantities.
3. Display stock: Shows the current stock levels for all items.
4. Add new stock: Add a new item to stock or update the quantity of an existing item.
5. Search for an item: Search for an item by name in stock.
6. Exit: Close the program.

CHAPTER 5
PROGRAM CODE

Code:

```
#include <stdio.h>
#include <string.h>
#define MAX_ITEMS 100
#define MAX_CART 50

// Item structure
typedef struct {
    char name[20];
    float price;
    int quantity;
} Item;

Item stock[MAX_ITEMS] = {
    {"apple", 0.5, 100},
    {"banana", 0.2, 150},
    {"grapes", 1.5, 200}
};
int stock_size = 3;

// Cart structure
typedef struct {
    char name[20];
    int quantity;
} CartItem;

CartItem cart[MAX_CART];
int cart_size = 0;

// Function to display stock items
void display_stock() {
    printf("\nStock:\n");
    for (int i = 0; i < stock_size; i++) {
        printf("%s - $%.2f - %d in stock\n", stock[i].name, stock[i].price, stock[i].quantity);
    }
}

// Binary search function to find an item in stock
int find_item(char *name) {
    int left = 0, right = stock_size - 1;
    while (left <= right) {
```

```
        int mid = left + (right - left) / 2;
        int cmp = strcmp(stock[mid].name, name);
        if (cmp == 0) return mid;
        if (cmp < 0) left = mid + 1;
        else right = mid - 1;}
    return -1;}

// Function to add an item to the cart
void add_to_cart() {
    char item_name[20];
    int quantity;

    printf("Enter item name: ");
    scanf("%s", item_name);
    printf("Enter quantity: ");
    scanf("%d", &quantity);

    int index = find_item(item_name);
    if (index != -1 && stock[index].quantity >= quantity) {
        // Add to cart
        strcpy(cart[cart_size].name, item_name);
        cart[cart_size].quantity = quantity;
        cart_size++;
        printf("Added %d of %s to the cart.\n", quantity, item_name);
    } else {
        printf("Item is out of stock or insufficient quantity.\n");
    }
}

// Function to checkout and generate bill
void checkout() {
    float total = 0.0;

    printf("\n--- Bill ---\n");
    for (int i = 0; i < cart_size; i++) {
        int index = find_item(cart[i].name);
        if (index != -1) {
            float cost = cart[i].quantity * stock[index].price;
            total += cost;
            stock[index].quantity -= cart[i].quantity;
            printf("%s x %d - $%.2f\n", cart[i].name, cart[i].quantity, cost);
        }
    }
    printf("Total bill: $%.2f\n", total);
```

```
    cart_size = 0; // Clear the cart after checkout
}

// Function to add new stock
void add_new_stock() {
    char item_name[20];
    float price;
    int quantity;

    printf("Enter new item name: ");
    scanf("%s", item_name);
    printf("Enter price: ");
    scanf("%f", &price);
    printf("Enter quantity: ");
    scanf("%d", &quantity);

    int index = find_item(item_name);
    if (index != -1) {
        // Item already exists, update quantity
        stock[index].quantity += quantity;
        printf("Updated quantity of %s to %d.\n", item_name, stock[index].quantity);
    } else if (stock_size < MAX_ITEMS) {
        // Add new item
        strcpy(stock[stock_size].name, item_name);
        stock[stock_size].price = price;
        stock[stock_size].quantity = quantity;
        stock_size++;
        printf("Added new item %s with quantity %d.\n", item_name, quantity);
    } else {
        printf("Stock list is full, cannot add more items.\n");
    }
}

// Function to search for an item in stock and display details
void search_item() {
    char item_name[20];
    printf("Enter item name to search: ");
    scanf("%s", item_name);

    int index = find_item(item_name);
    if (index != -1) {
        printf("Item found: %s - $%.2f - %d in stock\n", stock[index].name, stock[index].price,
stock[index].quantity);
    } else {
```

```
        printf("Item not found in stock.\n");
    }
}

// Main function
int main() {
    int choice;
    while (1) {
        printf("\nPOS System Menu:\n");
        printf("1. Add item to cart\n");
        printf("2. Checkout\n");
        printf("3. Display stock\n");
        printf("4. Add new stock\n");
        printf("5. Search for an item\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                add_to_cart();
                break;
            case 2:
                checkout();
                break;
            case 3:
                display_stock();
                break;
            case 4:
                add_new_stock();
                break;
            case 5:
                search_item();
                break;
            case 6:
                printf("Exiting program.\n");
                return 0;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
    return 0;
}
```

CHAPTER 6
TESTING EXECUTION

The **testing and validation phase** is essential in ensuring that every function in the POS (Point of Sale) system operates as expected, providing a reliable and efficient experience for cashiers managing retail transactions. This phase involves extensive testing of all core functions, including **add to cart**, **checkout**, **display stock**, **add new stock**, and **search for items**. Each function is evaluated under various conditions to confirm the program's functionality, reliability, and stability.

OUTPUT

```
C:\Pavithra\DS\CourseProject>gcc POSSystem.c -o POS.exe

C:\Pavithra\DS\CourseProject>POS.exe

POS System Menu:
1. Add item to cart
2. Checkout
3. Display stock
4. Add new stock
5. Search for an item
6. Exit
Enter your choice: 3

Stock:
apple - $0.50 - 100 in stock
banana - $0.20 - 150 in stock
grapes - $1.50 - 200 in stock

POS System Menu:
1. Add item to cart
2. Checkout
3. Display stock
4. Add new stock
5. Search for an item
6. Exit
Enter your choice: 1
Enter item name: apple
Enter quantity: 5
Added 5 of apple to the cart.
```

```
POS System Menu:
1. Add item to cart
2. Checkout
3. Display stock
4. Add new stock
5. Search for an item
6. Exit
Enter your choice: 1
Enter item name: grapes
Enter quantity: 10
Added 10 of grapes to the cart.

POS System Menu:
1. Add item to cart
2. Checkout
3. Display stock
4. Add new stock
5. Search for an item
6. Exit
Enter your choice: 2

--- Bill ---
apple x 5 - $2.50
grapes x 10 - $15.00
Total bill: $17.50

POS System Menu:
1. Add item to cart
2. Checkout
3. Display stock
4. Add new stock
5. Search for an item
6. Exit
Enter your choice: 3
Stock:
apple - $0.50 - 95 in stock
banana - $0.20 - 150 in stock
grapes - $1.50 - 190 in stock

POS System Menu:
1. Add item to cart
2. Checkout
3. Display stock
4. Add new stock
5. Search for an item
6. Exit
Enter your choice: 4
Enter new item name: watermelon
Enter price: 6
Enter quantity: 100
Added new item watermelon with quantity 100.

POS System Menu:
1. Add item to cart
2. Checkout
3. Display stock
4. Add new stock
5. Search for an item
6. Exit
Enter your choice: 5
Enter item name to search: watermelon
Item found: watermelon - $6.00 - 100 in stock
```

```
POS System Menu:
1. Add item to cart
2. Checkout
3. Display stock
4. Add new stock
5. Search for an item
6. Exit
Enter your choice: 6
Exiting program.
```

CHAPTER 7
CONCLUSION

In conclusion, the development of this Point of Sale (POS) system exemplifies the application of fundamental programming principles and data structures, resulting in a practical tool for retail transactions. The system allows cashiers to efficiently add items to a cart, calculate total costs, and update stock levels through a user-friendly menu-driven interface.

This system enhances the efficiency of the checkout process, allowing cashiers to quickly add products to a cart and generate bills for customers. The incorporation of stock management features ensures that inventory levels are accurately maintained, providing real-time updates that prevent overselling of items. Additionally, the ability to search for items and add new stock contributes to the overall effectiveness of the POS system, making it adaptable to changing inventory needs.

Overall, this POS system not only reinforces theoretical knowledge in data structures and algorithms but also provides practical insights into software development for retail applications. It highlights the importance of efficient inventory management and customer service in a supermarket context. By successfully completing this project, we have laid the groundwork for further exploration and innovation in point of sale solutions, ultimately contributing to improved operational efficiency in retail environments.

CHAPTER 8
REFERENCES

- P.J. Plauger - *The Standard C Library*
- Al Kelley and Ira Pohl - *A Book on C*
- **Data Structures and Algorithms in C** - Goodrich, Tamassia, Mount: Covers arrays, lists, and algorithms.
- **The C Programming Language** - Kernighan and Ritchie: Essential C programming basics.
- **GeeksforGeeks** and **TutorialsPoint**: Tutorials on C programming and data structures.
- **Stack Overflow**: Helpful for specific POS system examples and logic.
- **Algorithms in a Nutshell** - Heineman, Pollice, Selkow: Guide for algorithm design, pseudocode, and flowcharts

THANK YOU