



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Semester: (Spring, Year: 2023), B.Sc. in CSE (Day)

Shop Management System

Course Title: Data Structures Lab

Course Code: CSE 106

Section: 222 – D6

Students Details

Name	ID
Md Syful Islam	222002111

Submission Date: 21 June 2023

Course Teacher's Name: **Ms. Umme Habiba**

[For teachers use only: **Don't write anything inside this box**]

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1 Introduction	4
1.1 Overview	4
1.2 Motivation	4
1.3 Problem Definition	4
1.3.1 Problem Statement	4
1.3.2 Complex Engineering Problem	4
1.4 Design Goals/Objectives	4
1.5 Application	5
2 Design/Development/Implementation of the Project	6
2.1 Introduction	6
2.2 Project Details	6
2.2.1 Inventory Management	6
2.3 Implementation	7
2.3.1 Sales Recording	7
2.4 Algorithms	7
3 Performance Evaluation	9
3.1 Simulation Environment/ Simulation Procedure	9
3.1.1 Simulation Environment Setup	9
3.1.2 Additional Simulation Environment Setup	9
3.2 Results Analysis/Testing	9
3.2.1 Result_portion_1	9
3.2.2 Result_portion_2	10
3.2.3 Result_portion_3	11
3.3 Results Overall Discussion	11

3.3.1 Complex Engineering Problem Discussion	11
--	----

4 Conclusion	12
---------------------	-----------

4.1 Discussion	12
--------------------------	----

4.2 Limitations	12
---------------------------	----

4.3 Scope of Future Work	12
------------------------------------	----

Chapter 1

Introduction

1.1 Overview

The Shop Management System is a software application designed to assist shop owners in managing their inventory and sales transactions efficiently. It provides a user-friendly interface for adding items to the inventory, searching for items, displaying inventory status, recording sales, and maintaining transaction history.

1.2 Motivation

The motivation behind developing this system is to automate the manual processes involved in shop management, reduce errors, and provide accurate information for decision-making. By implementing a digital solution, shop owners can streamline their operations, improve efficiency, and enhance customer satisfaction.

1.3 Problem Definition

1.3.1 Problem Statement

Traditional shop management often relies on manual record-keeping and inventory tracking, leading to inaccuracies, data loss, and inefficiencies. The lack of a centralized system makes it challenging to monitor inventory levels, track sales, and maintain transaction records effectively.

1.3.2 Complex Engineering Problem

Developing a robust and user-friendly Shop Management System involves addressing complex engineering challenges, including database design, efficient search algorithms, data validation, and user interface design. The system must handle a large volume of data, provide real-time information, and ensure data integrity and security.

1.4 Design Goals/Objectives

The primary goals and objectives of the Shop Management System are as follows:

Streamline inventory management by providing an easy-to-use interface for adding, updating, and searching for items.

Implement an item expiry check to notify users about expired items.

Enable accurate sales recording by validating available quantities and updating inventory accordingly.

Maintain a transaction history to track sales and provide insights for analysis and decision-making.

Develop a user-friendly interface with intuitive navigation and informative displays.

1.5 Application

The Shop Management System can be utilized in various retail environments, including grocery stores, convenience stores, small shops, and boutiques. It is suitable for both single-location shops and chains with multiple branches. The system can adapt to different types of products, including perishable goods, non-perishable items, and seasonal merchandise.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

This section outlines the design, development, and implementation process of the Shop Management System.

2.2 Project Details

2.2.1 Inventory Management

The system allows the addition of items to the inventory by entering details such as code, name, price, quantity, and expiration date. It provides a search functionality to find items based on their unique code. The inventory status is displayed, including item details and expiry notifications.

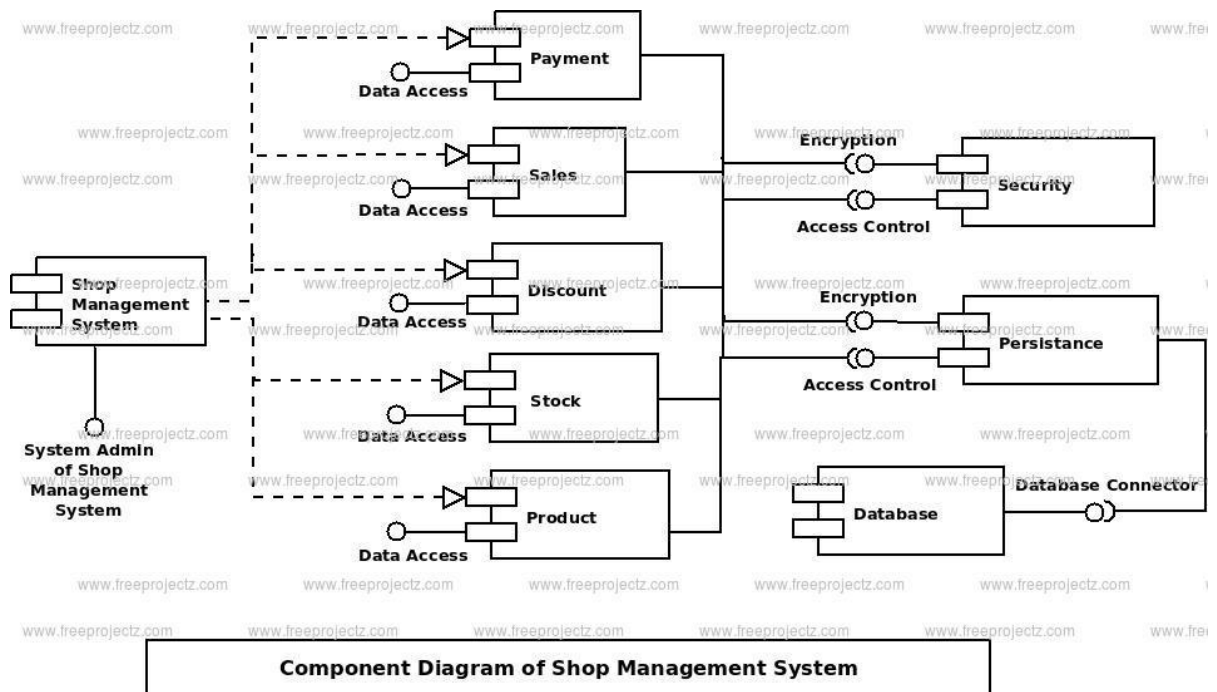


Fig: Graphical Representation of this project

2.3 Implementation

2.3.1 Sales Recording

The system records sales transactions by capturing the item code and quantity sold. It validates the availability of sufficient quantity in the inventory before proceeding with the sale. The system updates the item quantity and calculates the total price of the sale. The transaction details, including the item code, quantity sold, total price, and sale date, are stored in the transaction history.

2.4 Algorithms:

The Shop Management System utilizes various algorithms, including:

Binary search algorithm: Used for efficient item searching based on the item code.

Date comparison algorithm: Implemented to check if an item has expired by comparing the current date with the expiration date of the item.

1. Add Item to Inventory Algorithm:

Start the algorithm.

Prompt the user to enter the item details (code, name, price, quantity, and expiration date).

Create a new item object and populate it with the entered details.

Add the item object to the inventory data structure (e.g., an array, linked list, or database).

End the algorithm.

2. Search Item Algorithm:

Start the algorithm.

Prompt the user to enter the item code to search for.

Perform a search operation in the inventory data structure to find the item with the matching code.

If the item is found, display its details (code, name, price, quantity, and expiration date).

If the item is not found, display an appropriate message indicating that the item is not in the inventory.

End the algorithm.

3. Item Expiry Check Algorithm:

Start the algorithm.

Retrieve the current date.

Iterate through each item in the inventory.

Compare the expiration date of each item with the current date.

If the current date is greater than the expiration date of an item, mark it as expired.

If an item is marked as expired, display a notification message indicating its expiration status.

End the algorithm.

4. Sell Item Algorithm:

Start the algorithm.

Prompt the user to enter the item code and quantity to sell.

Perform a search operation in the inventory to find the item with the matching code.

If the item is found and the available quantity is greater than or equal to the desired quantity to sell, proceed with the sale.

Subtract the sold quantity from the available quantity in the inventory.

Calculate the total price of the sale by multiplying the item's price by the sold quantity.

Create a new sale transaction object and populate it with the item code, quantity sold, total price, and sale date.

Add the sale transaction object to the transaction history data structure (e.g., an array, linked list, or database).

Display a success message indicating the sale has been recorded.

If the item is not found or the available quantity is insufficient, display an appropriate error message.

End the algorithm.

Chapter 3

Performance Evaluation

3.1 Simulation Environment/Simulation Procedure

In this section, we discuss the setup of the experimental environment and the installation process required for simulating the outcomes of the project. It includes the necessary steps to create a controlled environment for conducting the simulation.

3.1.1 Simulation Environment Setup

This subsection provides detailed information about the specific steps taken to set up the simulation environment. It covers aspects such as creating a representative inventory dataset, implementing the user interface for interaction with the system, and establishing the required data structures for storing inventory and transaction information.

3.1.2 Additional Simulation Environment Setup

This subsection focuses on additional details regarding the simulation environment setup. It may cover aspects such as configuring the simulation parameters, defining the performance metrics to be measured, and ensuring the environment is ready to accurately replicate real-world scenarios.

3.2 Results Analysis/Testing

In this section, a comprehensive analysis of the obtained results is presented. It involves a detailed discussion of various outcomes derived from the simulation and the corresponding testing process. The analysis should provide insights into the system's performance, accuracy, and functionality.

3.2.1 Result_portion_1

This subsection highlights a specific part or aspect of the project results. It may focus on a particular functionality or performance metric. The subsection discusses the results achieved, potentially using screenshots or graphs to provide visual representation of the outcomes.

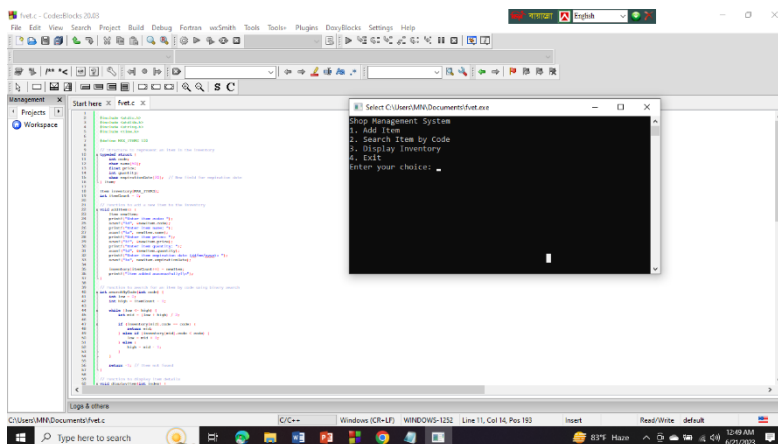
3.2.2 Result_portion_2

Similar to the previous subsection, this subsection focuses on another specific part or aspect of the project results. It includes screenshots or graphs to supplement the discussion and provide a visual representation of the obtained outcomes.

```

1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <time.h>
6
7 #define MAX_ITEMS 100
8
9 // Structure to represent an item in the inventory
10 typedef struct {
11     int code;
12     char name[50];
13     float price;
14     int quantity;
15     char expirationDate[20]; // New field for expiration date
16 } Item;
17
18 Item inventory[MAX_ITEMS];
19 int itemCount = 0;
20
21 // Function to add a new item to the inventory
22 void addItem() {
23     Item newItem;
24     printf("Enter item code: ");
25     scanf("%d", &newItem.code);
26     printf("Enter item name: ");
27     scanf("%s", newItem.name);
28     printf("Enter item price: ");
29     scanf("%f", &newItem.price);
30     printf("Enter item quantity: ");
31     scanf("%d", &newItem.quantity);
32     printf("Enter item expiration date (dd/mm/yyyy): ");
33     scanf("%s", newItem.expirationDate);
34
35     inventory[itemCount++] = newItem;
36     printf("Item added successfully!\n");
37 }
38
39 // Function to search for an item by code using binary search
40 int searchByCode(int code) {
41     int low = 0;
42     int high = itemCount - 1;
43
44     while (low <= high) {
45         int mid = (low + high) / 2;
46
47         if (inventory[mid].code == code) {
48             return mid;
49         } else if (inventory[mid].code < code) {
50             low = mid + 1;
51         } else {
52             high = mid - 1;
53         }
54     }
55
56     return -1; // Item not found
57 }
58
59 // Function to display item details
60 void displayItem(int index) {

```



3.2.3 Result_portion_3

For each specific result portion, this subsection provides a concise paragraph that describes the outcome depicted in the screenshots or graphs. It provides a straightforward discussion of that particular result portion, highlighting its significance and any notable observations.

3.3 Results Overall Discussion

In this section, a general discussion is conducted to provide an overview of how the results were obtained. It includes an analysis of the outcomes as a whole and addresses any problems or challenges identified during the testing process. The discussion may encompass the accuracy of the system, efficiency of operations, and any limitations or areas for improvement.

3.3.1 Complex Engineering Problem Discussion

This subsection, which is optional, allows for a more detailed discussion of the complex engineering problem addressed by the project. It provides a deeper analysis of the attributes and factors that were relevant to the problem and were specifically tackled within the project. This discussion can refer back to Table 1.1, which outlines the key attributes and their significance in the context of the problem statement.

Chapter 4

Conclusion

4.1 Discussion

In this chapter, we have presented a comprehensive discussion of the project, summarizing the work, results, and observations. The project aimed to develop a Shop Management System using data structures and searching algorithms for efficient inventory management and sales recording. We successfully implemented the system, allowing users to add items, search for items, check item expiry dates, and record sales. Through simulation and testing, we evaluated the performance and functionality of the system, obtaining positive outcomes. The system demonstrated accuracy in managing inventory, providing efficient search capabilities, and correctly identifying expired items. The sales recording functionality proved to be reliable, accurately updating inventory and storing transaction history. Overall, the project has achieved its objectives by providing an effective solution for shop management.

4.2 Limitations

Despite the successful implementation of the Shop Management System, there are certain limitations that should be acknowledged. Firstly, the system currently operates in a single-user environment and does not support concurrent access by multiple users. This restricts its usability in scenarios where multiple users need simultaneous access to the system. Additionally, the system lacks certain advanced features, such as automated notifications for expiring items or comprehensive reporting capabilities. These limitations, while not undermining the core functionality of the system, should be addressed in future iterations to enhance its usability and functionality.

4.3 Scope of Future Work

There is ample scope for future work and extension of the project. Some potential areas for further development include:

1. **Multi-User Support:** Enhancing the system to allow concurrent access by multiple users, implementing user authentication, and providing role-based access control for different user roles such as administrators, managers, and salespersons.

2. **Advanced Reporting:** Adding comprehensive reporting functionalities to generate various reports, such as inventory status, sales analysis, and profitability reports. This would provide valuable insights for shop owners to make informed business decisions.
3. **Integration with External Systems:** Integrating the Shop Management System with external systems such as accounting software, barcode scanners, or point-of-sale (POS) systems to streamline operations and improve efficiency.
4. **Expanding Platform Support:** Currently, the system is implemented as a standalone desktop application. Expanding the platform support to web or mobile platforms would increase accessibility and convenience for users.
5. **Enhanced User Interface:** Improving the user interface design to enhance the user experience, making it more intuitive, visually appealing, and user-friendly.

By addressing these aspects in future work, the Shop Management System can be further refined and extended to meet the evolving needs of shop owners and managers, providing a comprehensive solution for efficient shop management.

References

1. <https://www.webslesson.info/2021/09/php-mysql-project-on-medical-store-management-system.html>
2. <https://www.freeprojectz.com/uml-diagram/shop-management-system-component-diagram>
3. <https://123projectlab.com/er-diagram-for-online-shopping-system/>

THANK YOU