Comsats University, Lahore Campus

Data Structures and Algorithms

Submitted to: Mam Humera Faisal

Semester Project

Assignment no 3

Group Members:

Abeera Amir …… FA23-BCS-020-C

Muhammad Taha…… FA23-BCS-132-C

Saira Saeed…….. FA23-BCS-153-C

Name of Project:

Inventory Management System

Introduction:

The **Inventory Management System** is a project designed to help manage and organize inventory efficiently. It allows users to perform key operations such as adding new items, updating existing records, deleting obsolete items, and searching for specific inventory. The project applies data structures like linked lists, hash tables, and trees to provide efficient data handling and retrieval.

Requirements:

1. **Programming Language:** C++ .
2. **Development Environment:** Standard IDE or compiler (Code::Blocks).
3. **Core Features:**
    - Add items to inventory.
    - Delete items from inventory.
    - Search for items using various criteria.
    - Update item details like quantity or price.
    - Display the inventory list.
    - Alert for low-stock items.

Data Structure Choice:

1. **Linked List:**
    - Suitable for managing dynamic collections of inventory items.
    - Allows efficient addition and deletion of nodes.

2. **Hash Table:**
    - Provides quick lookup for items using unique identifiers like IDs.
3. **Binary Search Tree (BST):**
    - Helps maintain items in a sorted order for faster search and retrieval.
4. **Queue (Optional):**
    - Can be used to manage order processing in a First-In-First-Out (FIFO) manner.

Functionality:

1. Menu System:

The system offers a menu-based interface with the following options:

- Add Item
- Delete Item
- Update Item
- Search Item
- Display Inventory
- Exit Program

2. Add New Item:

- The user provides details such as:
    - Item ID (unique)
    - Item Name
    - Quantity
    - Price
- The system stores this information in the chosen data structure.

3. Delete Item:

- Items are removed based on their ID.
- The data structure is updated accordingly.

4. Update Stock Levels:

- Modify specific item details such as quantity or price.

5. Search Item:

- Users can search for items:
    - By ID (using hash table or BST).
    - By name (using linear traversal or BST).

6. Display Inventory:

- Displays all inventory items with details:
    - Item ID
    - Name
    - Quantity
    - Price
- If implemented with a BST, the items can be displayed in a sorted order.

7. Low Stock Alert:

- The system flags items with quantities below a certain threshold for restocking.

System Flow:

1. **User Input:**
    - The system takes input through a menu-driven interface.
2. **Operation Execution:**
    - Based on the user's choice, appropriate operations are performed on the data structure.
3. **Output:**
    - Results of operations (e.g., successful addition, item details, or error messages) are displayed to the user.

Potential Challenges:

- **Handling Duplicate IDs:** Ensure unique IDs for all items.
- **Efficient Searching:** Use a combination of data structures for optimal search performance.

- **Data Consistency:** Maintain consistent and accurate updates across all data structures.

Conclusion:

The **Inventory Management System** integrates theoretical knowledge of data structures into a practical application. It simplifies inventory operations while emphasizing efficient data handling techniques. This project provides a comprehensive learning experience for implementing linked lists, hash tables, and trees in real-world scenarios.