EasyInventory
(Inventory Management Application)

A Project Report
By

Md. Sakibul Islam

Department of Computer Science & Engineering
Pundra University of Science & Technology
Rangpur Road, Gokul, Bogura, Bangladesh.

December - 2024

Approval Page

EasyInventory
(Inventory Management Application)

A Project Report
By

Md. Sakibul Islam
ID:032222010510103

Batch: 21st
Session: Summer-22

Approved and supervised by

Mrittika Mahbub

Lecturer
Department of Computer Science and Engineering
Pundra University of Science & Technology

Abstraction

The project is a simple inventory management system implemented in C that allows users to perform various operations on a product inventory, such as adding, updating, deleting, viewing, searching, and selling products. The system handles product information, including product ID, name, brand, category, price, and quantity in stock, and stores this data in a text file for persistence. The program provides an interactive menu-driven interface for users to insert new products, update existing product details, and delete products from the inventory. It also includes a search feature that allows users to find products based on category or price range. For sales management, users can sell products by updating their stock quantities, ensuring sufficient stock is available before completing the sale. The project incorporates input validation to ensure correct data entry for integers and floating-point values, improving system reliability. By utilizing file handling, basic data structures, and standard input/output operations, this project serves as a practical demonstration of an inventory management application in C.

Table of Contents

Chapter Description	Page No.
1.Introduction	1
1.1 Background and Context	1
1.2 Purpose of GadgetGear	1
1.3 Target Audience	1
2. Related Work	2
3. Methodology	3-4
3.1 Requirement Analysis	3
3.2 System Design	3
3.3 Implementation	3
3.4 Testing	4
4.Conclusion and Future Implementation	5
4.1 Conclusion	5
4.2 Future Implementation	5

Introduction

Background and Context

In today's fast-paced retail and e-commerce industries, efficient inventory management plays a pivotal role in ensuring that businesses can meet customer demand while minimizing operational costs. With the advancement of technology, inventory management systems have become more complex, requiring businesses to adopt automated solutions to track and manage their products effectively. However, small businesses or those in the early stages of digital transformation often need a simple yet functional system to manage their inventory.

The **Easy**Inventory management system was developed with this need in mind. Designed as a straightforward, text-based application in C, GadgetGear provides an easy-to-use solution for managing a product inventory. It is suitable for small businesses or educational purposes, allowing users to track product details, update stock levels, and manage sales.

Purpose of **Easy**Inventory

The primary purpose of **Easy**Inventory is to provide a basic, efficient inventory management system that simplifies the process of handling products in a business setting. It enables users to insert new products into the inventory, update product details, delete outdated products, view current stock, search products by category or price range, and track sales by updating the quantity of products in stock. By using a text file for storage, GadgetGear ensures that the inventory data persists between program executions, making it suitable for continuous use in a business or educational environment.

Target Audience

The target audience for **Easy**Inventory includes small business owners, entrepreneurs, and students learning about inventory management and basic programming concepts. Small businesses with limited resources can benefit from a simple, low-cost solution to manage their product inventory. Wish to learn how to design and implement inventory management software using basic programming principles. Additionally, GadgetGear serves as a practical learning tool for those looking to understand file handling, data validation, and the implementation of basic business logic in C.

Related Works

Commercial Inventory Management Systems:

- **QuickBooks**, **Zoho Inventory**, **Fishbowl**, and **TradeGecko** are popular commercial software solutions offering advanced inventory management features like:
    - Real-time tracking of products
    - Barcode scanning
    - Multi-location management
    - Sales order and restocking automation
- These systems are tailored for medium to large businesses and typically come with:
    - High costs
    - Complexity in setup and maintenance
    - Integration with accounting and other business functions
- While feature-rich, they may be unsuitable for small businesses or startups due to the significant investment required.

Educational Inventory Management Systems:

- Basic inventory systems developed in programming languages like Python, C, and Java are commonly used in educational settings to teach fundamental concepts of:
    - File handling
    - User input validation
    - Database integration
- These systems typically focus on:
    - Simplicity
    - Limited features
    - Use in academic environments where students can learn how such systems function at a technical level
- They serve as a foundation for understanding inventory management without the complexity of commercial solutions.

Methodology

The development of the **Easy**Inventory Inventory Management System follows a systematic approach involving the following key stages:

1. **Requirement Analysis**:
    - Initially, the core functionality needed for the inventory system was identified. This included basic inventory operations like inserting, viewing, updating, deleting, and selling products. Additionally, searching products by category or price was prioritized to provide efficient access to product details.
2. **System Design**:
    - **Data Structure**: The system is built around a simple data structure, Product, which contains fields like id, product_name, brand, category, price, and quantity_in_stock. These attributes are sufficient for managing product information in the context of small inventory operations.
    - **File-Based Storage**: The inventory data is stored in a plain text file (products.txt). This approach was chosen to minimize complexity and allow easy persistence of product data without requiring advanced database systems.
    - **Menu-Driven Interface**: The user interacts with the system through a text-based menu that allows them to perform various operations. This interface was designed to be intuitive, enabling users to easily navigate between actions like inserting, viewing, updating, and deleting products.
3. **Implementation**:
    - **Input Validation**: The system ensures data integrity through input validation. Validations were implemented for integer inputs (e.g., product ID and quantity) and floating-point inputs (e.g., product price). This is done using custom functions to ensure correct input types and prompt the user for re-entry in case of invalid data.
    - **Inventory Operations**:
        - **Insert Product**: New products are added to the inventory with an automatically generated ID. Product details like name, brand, category, price, and quantity are entered by the user.
        - **View Products**: The user can view the list of all products in the inventory, displaying key details.
        - **Update Product**: Users can update product details based on the product ID, allowing for modifications such as changing the price or quantity.

- **Delete Product**: Products can be removed from the inventory by providing the product ID, with all data subsequently shifted to close the gap.
- **Search Products**: This operation allows users to search products by category and/or price range, enabling more targeted inventory management.

```
=>GadgetGear - Inventory==========
1. Insert Product
2. View Products
3. Update Product
4. Delete Product
5. Search Products
6. Sell Product
7. Exit
Enter your choice:
```

```
All Products:
ID: 1, Name: Iphone 15, Brand: Apple, Category: Phone, Price: $1000.00, Quantity: 10
ID: 2, Name: Mackbook Air, Brand: Apple, Category: Laptop, Price: $1500.00, Quantity: 5

=>GadgetGear - Inventory==========
1. Insert Product
2. View Products
3. Update Product
4. Delete Product
5. Search Products
6. Sell Product
7. Exit
```

4. **Testing**:
    - The system was tested using sample product data to ensure all functions (insert, view, update, delete, sell, search) work as expected.
    - Various edge cases were considered, such as attempting to update a non-existing product, selling more stock than available, and providing invalid input types.

Conclusion

The **EasyInventory Inventory Management System** successfully addresses the need for a simple yet effective tool to manage product inventories for small businesses and educational purposes. By focusing on core functionalities such as adding, viewing, updating, deleting, and selling products, the system provides a streamlined approach to inventory management. With a text-based interface, file-based storage, and easy navigation, GadgetGear ensures that even users with minimal technical expertise can efficiently manage their products.

The project also emphasizes input validation, ensuring data integrity throughout various operations, and testing was conducted to ensure the system's stability and reliability. While the current version meets its primary objectives, there is potential for expansion to include more advanced features for broader applications.

Future Work

Despite its success, **Easy**Inventory can benefit from several enhancements and future improvements:

1. **Database Integration**: Transitioning from file-based storage to a database (e.g., MySQL, SQLite) would improve scalability and performance, especially when dealing with larger product inventories.
2. **User Authentication and Access Control**: Adding user authentication and role-based access would make the system more secure and allow multiple users to interact with the inventory, each with different permissions (e.g., admin, staff).
3. **Graphical User Interface (GUI)**: Moving from a text-based interface to a graphical user interface would enhance user experience, making the system more intuitive and accessible to non-technical users.
4. **Barcode Scanning**: Integrating barcode scanning functionality would streamline product insertion, selling, and stock tracking, offering a more efficient solution for real-world businesses.
5. **Reporting and Analytics**: Adding reporting features such as sales reports, stock-level alerts, and performance analytics would provide users with actionable insights into their inventory operations.