

Assignment Case Study: Inventory Management System

Problem Statement: Inventory Management System

Objective: Develop a console-based Inventory Management System to help a small business track and manage its products and their quantities. The system should allow users to add new products, update product quantities, view the inventory in the order products were added, and ensure efficient lookups and updates for product quantities.

Requirements:

1. Product Addition:

- Users can add a new product to the inventory by providing a product name.
- Each product must have a unique name.
- Products should be stored in the order they are added for display purposes.
- Initially, the quantity of a new product is set to 0.

2. Quantity Update:

- Users can update the quantity of an existing product by specifying the product name and a quantity (positive to add, negative to reduce).
- Ensure quantities cannot become negative (set to 0 if a reduction would result in a negative value).
- Display appropriate error messages for invalid inputs (e.g., non-existent product, invalid quantity).

3. View Inventory:

- Display all products in the order they were added, along with their current quantities.
- If the inventory is empty, display a message indicating no products are available.

4. User Interface:

- Provide a simple menu-driven console interface with the following options:
 - Add Product
 - Update Quantity
 - View Inventory

- Exit
- Prompt users for inputs and validate them to ensure robust operation.

5. Data Structures:

- Use a `List` to maintain the order of products as they are added.
- Use a `Dictionary` to store product names as keys and their quantities as values for efficient lookups and updates.
- Ensure the two data structures remain synchronized (e.g., adding a product updates both the `List` and `Dictionary`).

6. Constraints:

- Product names are case-sensitive strings.
- Quantities are integers (positive or negative for updates, but final quantities must be non-negative).
- Handle invalid inputs gracefully (e.g., duplicate product names, non-existent products, or non-integer quantities).
- The system should be beginner to intermediate-friendly, using basic C# constructs and clear error messages.

Expected Output Example:

...

Inventory Manager

1. Add Product
2. Update Quantity
3. View Inventory
4. Exit

Choose an option: 1

Enter Product Name: Apples

Product added!

Inventory Manager

1. Add Product
2. Update Quantity

3. View Inventory

4. Exit

Choose an option: 1

Enter Product Name: Oranges

Product added!

Inventory Manager

1. Add Product

2. Update Quantity

3. View Inventory

4. Exit

Choose an option: 2

Enter Product Name: Apples

Enter Quantity to Add (can be negative): 10

Quantity updated!

Inventory Manager

1. Add Product

2. Update Quantity

3. View Inventory

4. Exit

Choose an option: 3

Inventory:

Apples: 10 units

Oranges: 0 units

Inventory Manager

1. Add Product

2. Update Quantity

3. View Inventory

4. Exit

Choose an option: 2

Enter Product Name: Bananas

Product not found.

Inventory Manager

1. Add Product

2. Update Quantity

3. View Inventory

4. Exit

Choose an option: 4

...

Technical Requirements:

- Use C# with .NET (any version compatible with `List` and `Dictionary`).
- Implement error handling for invalid inputs (e.g., duplicate products, invalid quantities).
- Use `List<string>` for maintaining product order and `Dictionary<string, int>` for mapping product names to quantities.
- Keep the code simple and well-commented for beginner to intermediate learners.
- Avoid advanced features like file I/O, databases, or multithreading to maintain focus on `List` and `Dictionary`.

Learning Outcomes:

- Understand how to use `List` for ordered collections and `Dictionary` for key-value mappings.
- Learn to combine `List` and `Dictionary` to leverage their strengths (order preservation and fast lookups).
- Practice input validation and error handling in a console application.
- Gain experience with basic C# programming constructs and data structure synchronization.