**Proof of Concept (PoC) Document for Ecommerce (Admin Page)**

**Document Version Control:**

| **Version** | **Date** | **Author** | **Description** |
| --- | --- | --- | --- |
| 1.0 | 2024-Oct-4 | Abhijeet. Hishobkar | Initial Draft |

## ****1. Introduction****

### ****1.1 Purpose****

This Proof of Concept (PoC) document demonstrates the functionality of a small administrative application developed using .NET WinForms. The app enables an administrator to manage a list of products by adding, editing, and deleting product details. The goal of the PoC is to validate the technical feasibility of the application and its core features in a real-world scenario.

### ****1.2 Scope****

The PoC focuses on providing a basic yet functional interface for managing product information. The key scope elements include:

* Listing products in a ListView control.
* Adding new products with a name and description.
* Editing existing products.
* Deleting products from the list.
* Basic validation to ensure data integrity.

This PoC is not intended to cover advanced features such as database integration, authentication, or role-based access control (RBAC), but such features can be incorporated in future development phases.

## ****2. System Overview****

### ****2.1 Technology Stack****

The application is built using the following technologies:

| **Component** | **Technology/Framework** |
| --- | --- |
| Front-End UI | .NET WinForms |
| Programming Language | C# |
| Platform | Windows |

### ****2.2 Architecture Overview****

The application follows a traditional desktop architecture, where all operations (add, edit, delete) are performed locally on the user’s machine. The interface uses Windows Forms and is driven by event-based programming in C#.

## ****3. Features & Functionalities****

### ****3.1 Product ListView****

The application’s primary interface is a ListView control that displays a list of products. Each product includes:

* **Product Name**: The name of the product (string).
* **Description**: A brief description of the product (string).

### ****3.2 Add Product****

An administrator can add new products to the list using an “Add Product” button. When clicked, a form is displayed where the user can input:

* **Product Name** (required).
* **Description** (optional).

Upon successful submission, the new product is added to the ListView.

### ****3.3 Edit Product****

Users can select an existing product from the ListView and edit its details using the "Edit" button. This opens a form pre-populated with the selected product’s details. The administrator can modify the fields and save changes.

* **Validations**: The product name must not be empty.

### ****3.4 Delete Product****

The "Delete" button allows administrators to remove a selected product from the list. Confirmation dialogs are used to prevent accidental deletions.

### ****3.5 Data Validation****

* Product name is required during both the addition and editing of items.
* The ListView enforces unique product names to avoid duplicates.

## ****4. User Interface****

### ****4.1 Main Interface****

The main interface consists of:

* **ListView Control**: Displays the list of products with columns for "Product Name" and "Description."
* **Add, Edit, Delete Buttons**: These buttons are placed below or beside the ListView control for easy access.
* **Search Functionality** (Optional for future extension): Allows searching through the list of products by name.

### ****4.2 Add/Edit Form****

The Add/Edit form is simple and contains:

* A textbox for entering the **Product Name**.
* A multiline textbox for entering the **Description**.

## ****5. Technical Details****

### ****5.1 Key Classes and Components****

| **Component** | **Description** |
| --- | --- |
| **MainForm** | The main form that holds the ListView and buttons. |
| **Product** | A model class representing a product with Name and Description properties. |
| **AddEditForm** | A dialog form that allows adding or editing a product. |
| **ProductManager** | Manages the in-memory list of products and handles add, edit, and delete operations. |

### ****5.2 ListView Management****

The ListView is populated using the ProductManager class. The following methods are key to managing the ListView:

* **AddProduct()**: Adds a new product to the internal list and refreshes the ListView.
* **EditProduct()**: Edits the selected product and updates the ListView with the new details.
* **DeleteProduct()**: Removes the selected product from the list and refreshes the view.

### ****5.3 Event Handling****

* **Button Click Events**: Handlers are attached to the Add, Edit, and Delete buttons to open forms or perform list actions.
* **Form Validation**: The Add/Edit form performs input validation to ensure that no product is added or updated with an empty name.

### ****5.4 Data Persistence**** (For Future Enhancement)

Currently, the product list is stored in memory. In future iterations, the app can be extended to use a database (SQL Server, SQLite) or file-based storage (e.g., JSON, XML) to persist the product list between sessions.

## ****6. Assumptions & Constraints****

### ****6.1 Assumptions****

* The administrator will only manage a moderate number of products (e.g., up to 30 products).
* The application will run on Windows platforms.
* No backend or database integration is required for this PoC version.

### ****6.2 Constraints****

* Data is not persisted between sessions.
* User roles and authentication mechanisms are out of scope.
* The ListView does not support advanced features like pagination or sorting, although these can be added later.

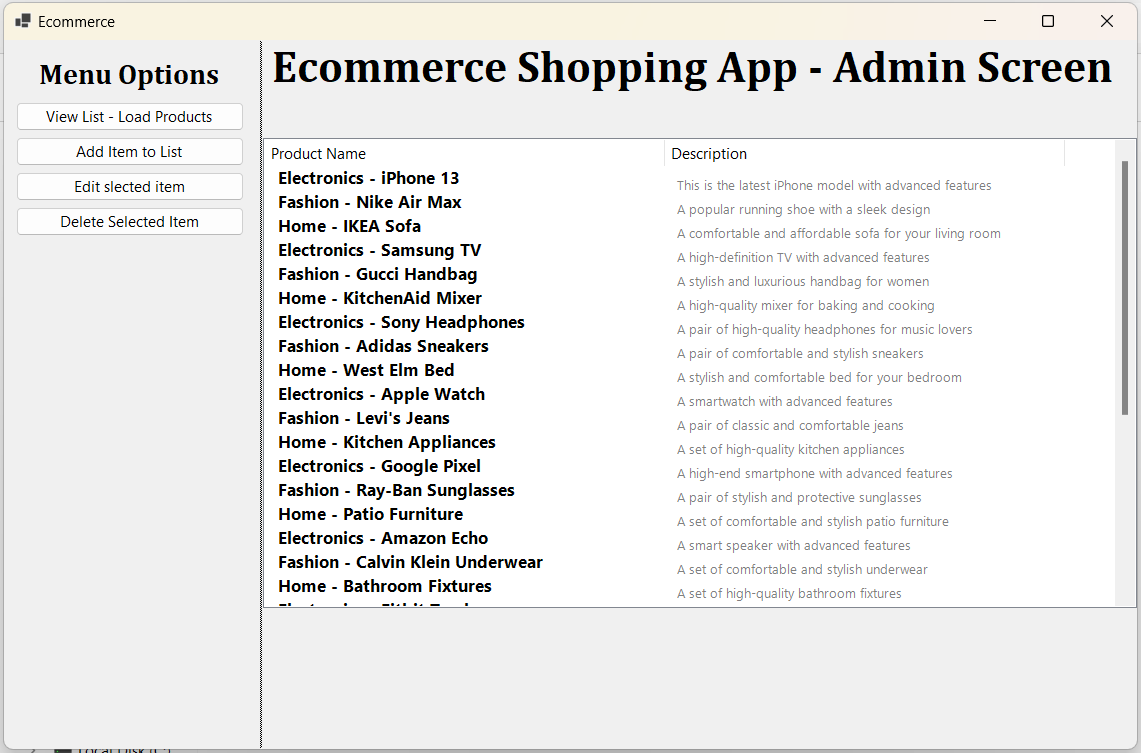
## ****7. Conclusion****

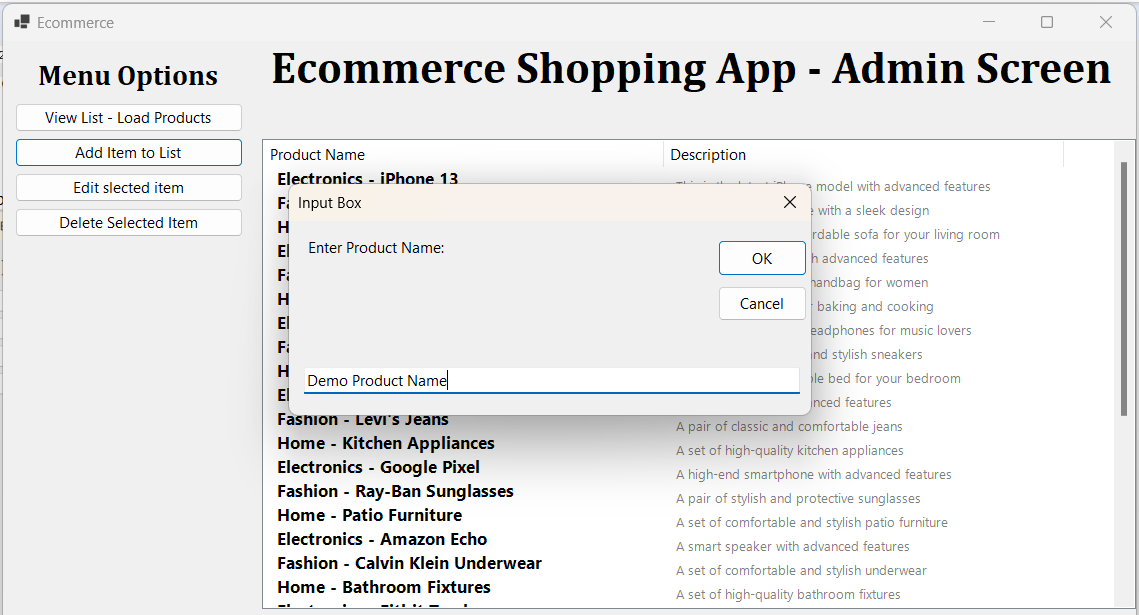
This PoC successfully demonstrates the core functionality of a small admin panel that allows administrators to manage a list of products. The current implementation is efficient and user-friendly, ensuring that basic CRUD (Create, Read, Update, Delete) operations can be performed with minimal effort.

The next steps may include extending the functionality with features like persistent storage, enhanced validation, search functionality, and user authentication if needed.

**Appendices :-**

**Main Screen:**



**Add Product Name Screen:** 

**Edit Selected Item Popup Screen:** 