Electronics - E-Commerce Management System

1. Overview

Electronics is an e-commerce management system developed using ASP.NET MVC. It enables efficient management of products, orders, customers, and admin controls. The system follows the Model-View-Controller (MVC) architecture, ensuring scalability and maintainability.

Core Modules:

1. **Product Management –** Handles product catalog, inventory, and descriptions.

2. **Cart Management –** Allows users to add, remove, and update cart items.

3. **Order Management –** Manages order placements, processing, and tracking.

4. **Admin Control –** Handles authentication, product management, and user roles.

Assumptions

1. The application will be deployed locally during development using a relational database (e.g., MySQL or SQL Server).

2. Role-based authentication will secure sensitive information.

3. ORM frameworks (Hibernate for Java or Entity Framework for .NET) will handle database interactions.

4. No containerization will be used for local deployment.

2. Use Cases

2.1 Product Management Module

Purpose: Manages the product catalog including adding, updating, and displaying products.

Controller:

ProductsController

- AddProduct(productData) – Adds a new product.
- UpdateProduct(productId, productData) – Updates product details.

- GetProductDetails(productId) – Retrieves product details.

Service:

ProductService - Handles product-related operations and validations.

Model:

Entity: Product

Attributes:

- productId (PK)
- name (VARCHAR)
- description (TEXT)
- price (DECIMAL)
- stock (INTEGER)

2.2 Cart Management Module

Purpose: Allows users to add, remove, and manage items in their shopping cart.

Controller:

CartController

- AddToCart(cartItemData) – Adds an item to the cart.
- RemoveFromCart(cartItemId) – Removes an item from the cart.
- GetCartItems(userId) – Retrieves items in the user's cart.

Service:

CartService -

Manages cart operations and pricing calculations.

Model:

Entity: CartItem

Attributes:

- cartItemId (PK)
- userId (FK)
- productId (FK)

- quantity (INTEGER)

2.3 Order Management Module

Purpose: Manages customer orders, payments, and tracking.

Controller:

OrdersController

- PlaceOrder(orderData) – Places a new order.
- UpdateOrderStatus(orderId, status) – Updates the order status.
- GetOrderDetails(orderId) – Retrieves order details.

Service:

OrderService -

Handles order processing and tracking.

Model:

Entity: Order

Attributes:

- orderId (PK)
- userId (FK)
- totalAmount (DECIMAL)
- orderDate (DATE)
- status (VARCHAR)

2.4 Admin Control Module

Purpose: Provides administrative functionalities such as product management and user roles.

Features:

- Role-based authentication for secure access.
- Admin dashboard for product and order management.
- Secure login and user role differentiation.

4. Database Schema

4.1 Table Definitions

1.PRODUCTS TABLE:

```
CREATE TABLE Products (
                    ProductID INT PRIMARY KEY,
                    Name VARCHAR(255),
                    Description TEXT,
                    Price DECIMAL(10, 2),
                    CategoryID INT,
                    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
                );
```

2. USER TABLE:

```
CREATE TABLE Users (
                    UserID INT PRIMARY KEY,
                    Username VARCHAR(50),
                    Password VARCHAR(255),
                    Email VARCHAR(255),
                    Role VARCHAR(20)
                );
```

3. ORDERS TABLE

```
        CREATE TABLE Orders (
          OrderID INT PRIMARY KEY,
          UserID INT,
          OrderDate DATETIME,
          TotalAmount DECIMAL(10, 2),
          Status VARCHAR(20),
          FOREIGN KEY (UserID) REFERENCES Users(UserID)
        );
```

4.IMAGES TABLE

```
CREATE TABLE Images (
   ImageID INT PRIMARY KEY,
 ProductID INT,
   URL VARCHAR(255),
   FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

Conclusion:

This document outlines the low-level design for the E-commerce website, ensuring modularity, scalability, and compatibility with Spring MVC and ASP.NET Core MVC framework