Practice Assignment 2

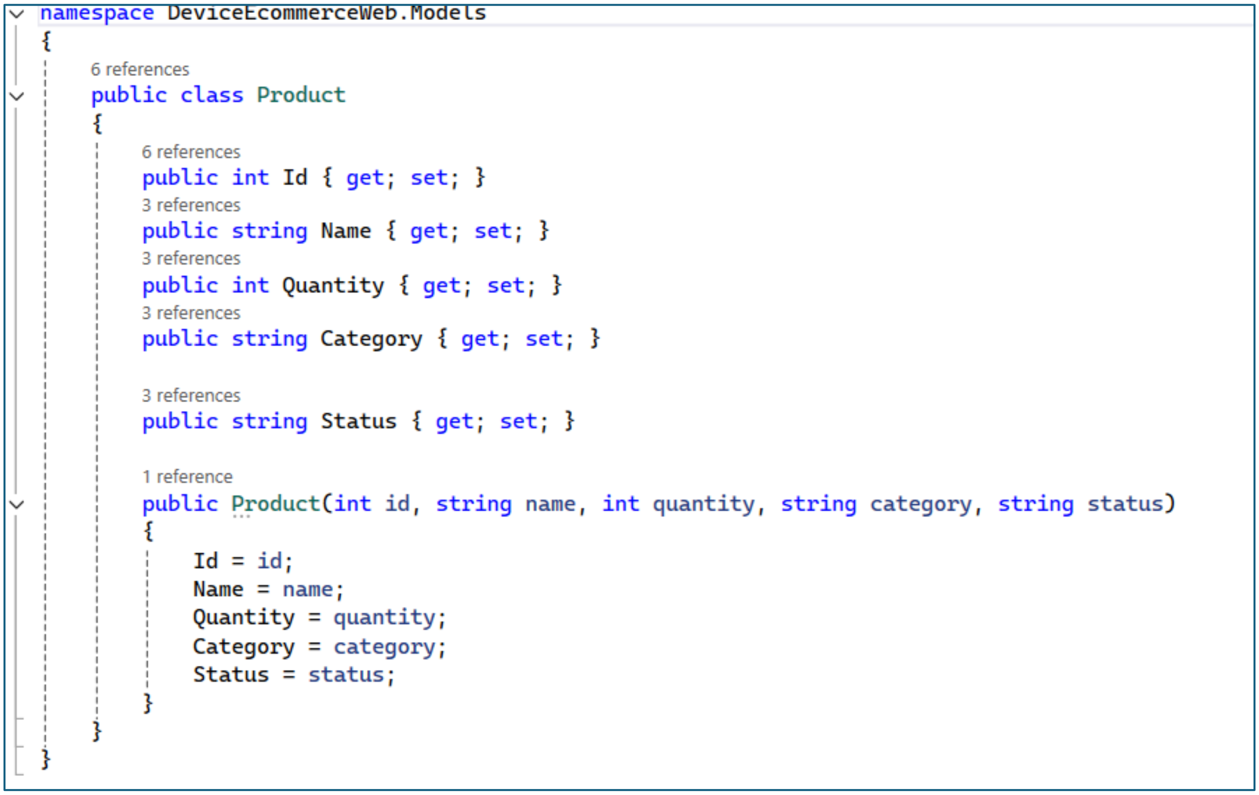
1. Introduction

The Device Management Software is a .NET Core MVC-based system designed to manage device categories, devices, and users efficiently. It offers functionalities to add, edit, and delete device categories, devices, and users while ensuring data integrity through validation. The system also includes search and filter capabilities for better accessibility.

2. Project Structure

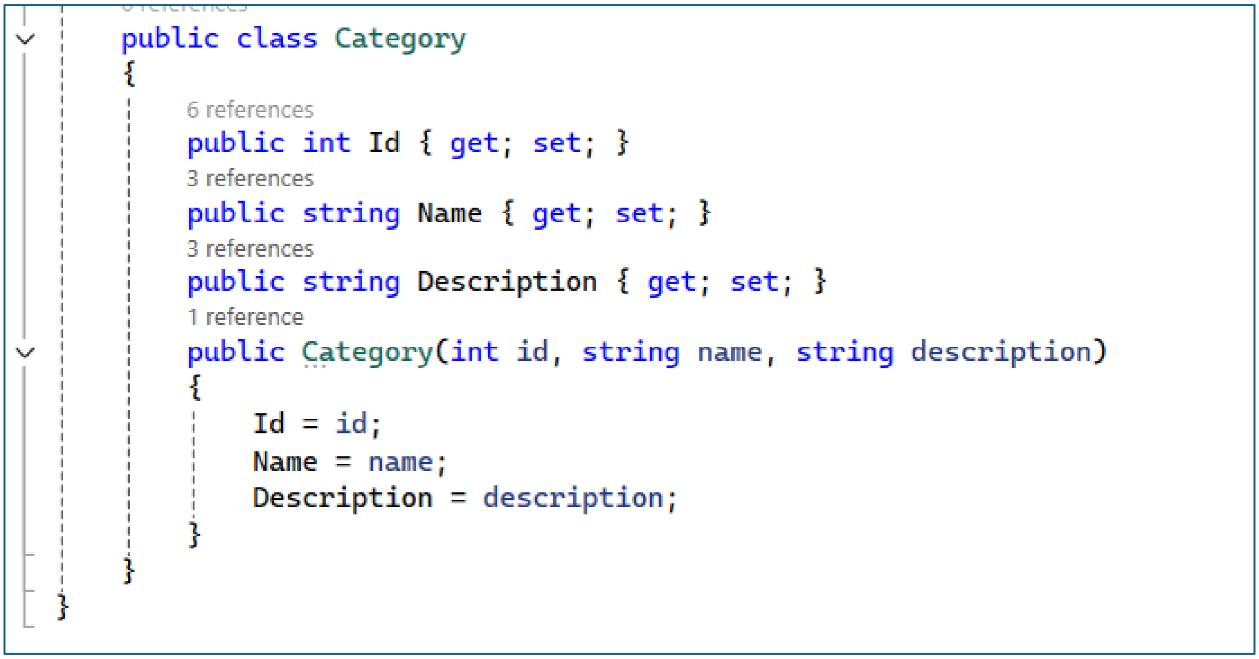
3. Created Classes and Their Purpose

3.1 Product.cs



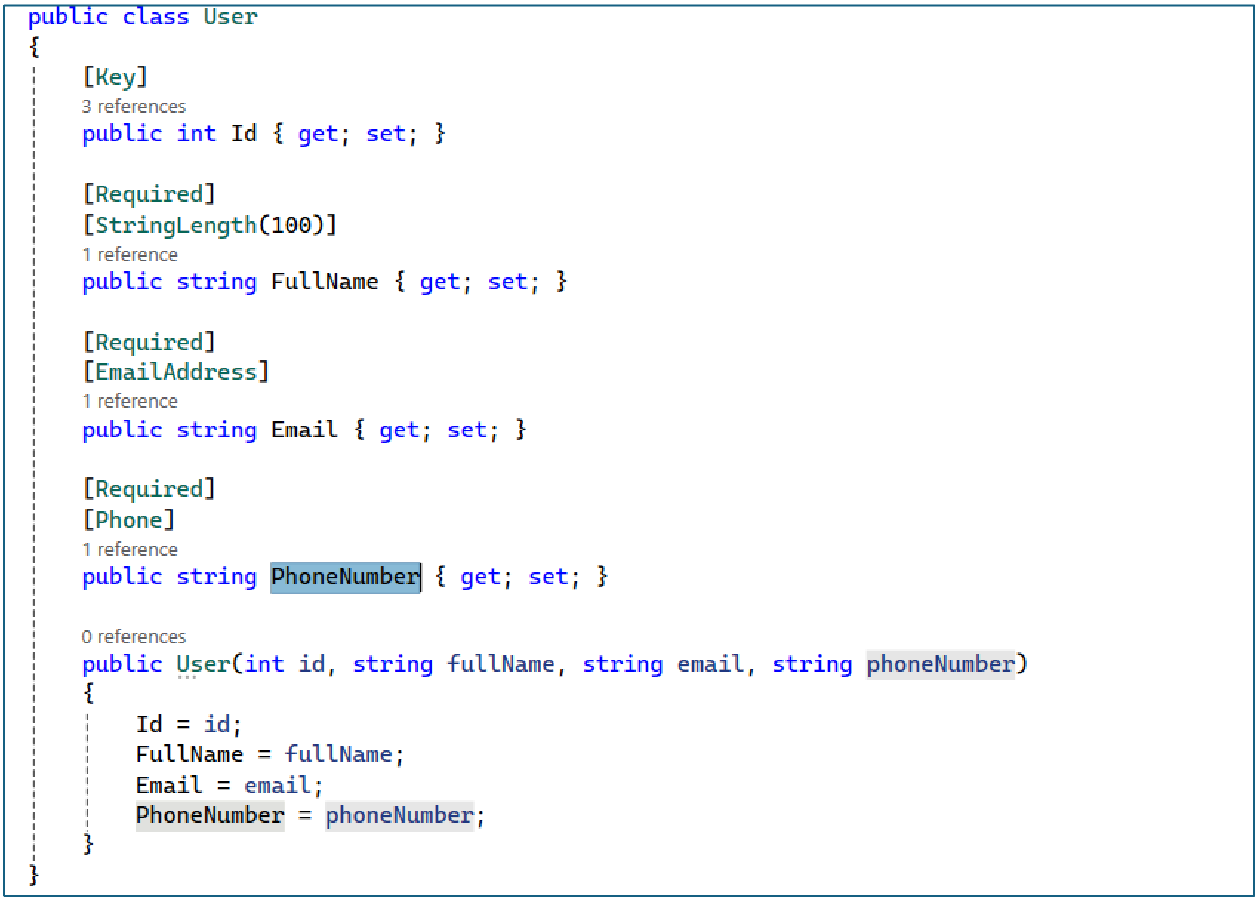
```
namespace DeviceEcommerceWeb.Models
{
    6 references
    public class Product
    {
        6 references
        public int Id { get; set; }
        3 references
        public string Name { get; set; }
        3 references
        public int Quantity { get; set; }
        3 references
        public string Category { get; set; }

        3 references
        public string Status { get; set; }

        1 reference
        public Product(int id, string name, int quantity, string category, string status)
        {
            Id = id;
            Name = name;
            Quantity = quantity;
            Category = category;
            Status = status;
        }
    }
}
```

3.2 Category.cs



```
    0 references
    public class Category
    {
        6 references
        public int Id { get; set; }
        3 references
        public string Name { get; set; }
        3 references
        public string Description { get; set; }
        1 reference
        public Category(int id, string name, string description)
        {
            Id = id;
            Name = name;
            Description = description;
        }
    }
}
```

3.3 User.cs



```
public class User
{
    [Key]
    3 references
    public int Id { get; set; }

    [Required]
    [StringLength(100)]
    1 reference
    public string FullName { get; set; }

    [Required]
    [EmailAddress]
    1 reference
    public string Email { get; set; }

    [Required]
    [Phone]
    1 reference
    public string PhoneNumber { get; set; }

    0 references
    public User(int id, string fullName, string email, string phoneNumber)
    {
        Id = id;
        FullName = fullName;
        Email = email;
        PhoneNumber = phoneNumber;
    }
}
```

4. Implementation of Features

4.1 Category Management

- Implemented CRUD operations using CategoryController.
- Used Entity Framework Core for database interactions.
- Created Razor Views for user-friendly UI.

4.2 Product Management

- Implemented CRUD operations in ProductController.
- Provided a dropdown list for selecting device categories.
- Ensured validation of required fields before submission.
- Displayed a categorized list of devices in the UI.

4.3 User Management

- Implemented CRUD operations in UserController.
- Ensured validation for email and phone number formats.
- Provided a UI for easy user management.

4.4 Search and Filter Data

- Implemented search by device name and code.
- Enabled filtering by device status and category.

5. UI Design

- Implemented a simple and responsive design using Bootstrap.
- Created tables for listing categories, devices, and users.
- Used form validation to enhance user experience.

6. Validation

Ensured error messages appear in the UI when inputs are invalid.

7. Coding Standards & Clean Code Principles

- Followed proper naming conventions and code formatting.
- Used Dependency Injection for better maintainability.

8. Conclusion

The Device Management Software successfully meets the requirements of managing device categories, devices, and users. With a structured MVC approach, efficient database handling, and intuitive UI, the system ensures a seamless user experience while maintaining clean and maintainable code.