Trần Lê Hiền Đức

2131200129

Device Management System - Project Report

The Device Management System is a web application that is developed on ASP.NET Core frameworks, which helps manage devices, categories, and users in organizations. It provides an extremely user-friendly way to track assignment of devices, categorize assets, and manage users' activities.

1. Summary of Project

This system reduces the bustles of organizations by providing a single platform for the management of devices and users. It has role-based access control, Search and Filter feature, and Real-Time Statistics to serve a variety of modern needs.

2. Project Structure

This project implements the Model View Controller architectural pattern so that systems can be scaled and maintained well.

The Models folder contains the major classes:

- `Device.cs` refers to devices with attributes of name and code and status.
- `DeviceCategory.cs` maintains the categorization.
- `User.cs` provides user details and assignments.
- `Role.cs` assigns roles and permissions.

Controllers implement business logic:

- `DevicesController.cs` handles definition of operations regarding devices while
- `DeviceCategoryController.cs` governs the functionalities of category.
- `UsersController.cs` performs activities concerning users.
- `HomeController.cs` implements the dashboard navigation and statistics.

Views establish the user interface involving the CRUD operation by each controller. Besides that, the shared layout uses reusable partial views to keep the consistency.

3. Execution Details

User Management

User management, in terms of CRUD operations, role-based access control, and user-device assignments, is to be implemented. Administrators would, therefore, manage users through forms that have been thoroughly validated and allow for searching by attributes such as name, email, or phone.

Controllers use services to ensure data integrity and drive business rules while views render interactive interfaces that feature validation and error feedback.

Device Management

This is a full tracking system for the lifecycle of devices. Controllers provide actions such as add, edit, delete devices, and log status and ownership changes. The system has implemented advanced filtering in the service layer, utilizing LINQ to provide efficient querying. Devices are made through many-to-one relationships and vis-a-vis on the frontend with card-based UI elements.

Category Management

Categories are logically designed to group the devices. The system is relationally mapped, meaning each device has one-to-one relation to each category. Its implementation includes dynamic dropdowns in forms, which would ensure that assignment to any specific category can be done easily during device creation or editing. CRUD operations for categories are managed just like that of devices but with services handling validation and interaction with a database.

Dashboard Implementation

It has a dashboard, which summarizes and presents system metrics in real-time. Controllers query the database for user, device, and category counts and format the data into dynamically appealing summaries using card components. Quick access links to frequented features were made available for smoother navigation.

4. Database Design

The system uses Entity Framework Core to manage data with key relationships:

- User-Device (One-to-Many): Tracks which devices are assigned to each user.
- DeviceCategory-Device (One-to-Many): Groups devices under specific categories.
- User-Role (Many-to-One): Assigns permissions through roles.

Migrations are used for database updates, and seed data initializes roles and admin users.

5. Search and Filter Implementation

The search and filter functionality works not statically but dynamically within the service layer. It delegates the processing of queries via controllers to services, where the last constructs efficient and well-performing LINQ expressions for fetching results. Then, using open-ended applications, the frontend leads itself to real-time view changes and instant search feedback. Filtering herein works with dropdowns as well as checkboxes-for such attributes like status, category, and assignment.

6. Validation Implementation

Validation is an essential facility in the operation of the system by which integrity is maintained in the data created and edited for users, devices, and categories. Users: Validation exists on the fields that are usually found in user creation and editing forms. This includes name, email, and phone fields. In the server, checks are to ensure that the combination of email addresses is unique and checks role

assignments in proper scenarios, while client-side validations provide instant feedback where required fields were left blank. Devices: In the creation/editing of forms, the essential fields are validated. For example: device name, code, status, and their descriptions. Unique device codes from the server side prevent duplication, and categories are also enforced to have all-paged categories. Categories: This validation guarantees the names of categories to be unique and compulsory; Server checks do not allow the creation of duplicate entries while Client Validation indicates missing or incorrect input in real-time. All these validations use data annotations in model classes and some custom validation logic in services. View errors will actually represent these messages to guide the user in solving the issues.

7. Features of UI/UX

In the end, it is the responsive UI because it would suffice for display on every type of device. Tabular data could also be grid layouts; otherwise, it can be card-based for some easily readable summary metrics. Validation messages and buttons add interactivity to the user experience, while the same aesthetics feature throughout the application improves usability.

Conclusion

The Device Management System is described as a scalable, secure, and efficient system now capable of managing all of the organizational resources. With the use of ASP.NET Core MVC, the system offers flexible features for device management-exclusively included in the categories and users-to make it an enterprise resource to use for optimizing their workflows.