Name: Mariz Erian Ibrahim

API Assignment Documentation

Project Overview

This is a .NET Core Web API project for managing students and departments with authentication and authorization features. The application uses Entity Framework Core, ASP.NET Core Identity, and JWT-based authentication.

Key Components

1. Authentication and Authorization

- Uses ASP.NET Core Identity for user management
- Implements JWT (JSON Web Token) authentication
- Supports two roles: Admin and Student
- Registration automatically assigns roles based on username

Authentication Workflow

1. User registers with `AccountController.Register()`
    - Usernames containing "admin" get the Admin role
    - Other usernames get the Student role
2. User logs in with `AccountController.Login()`
    - Generates a JWT token with claims
    - Token expires after 1 day

2. Controllers

AccountController

- `/api/Account/Register`: User registration
- `/api/Account/Login`: User authentication
- `/api/Account/Role`: Role management

DepartmentController

- Supports CRUD operations for departments
- Requires authentication
- Admin can add, edit, and delete departments
- Students and Admins can view departments

StudentController

- Supports CRUD operations for students
- Implements role-based access control
- Students can only edit their own profiles
- Admins can edit and delete any student profile

3. Data Models

- `Student`: Represents student information
- `Department`: Represents department information
- Custom validation attributes:
  - `BdBefore2013`: Ensures birthdate is before or in 2013
  - `Unique`: Validates unique email addresses

4. Repository Pattern

- `IGenericRepoServices<T>`: Generic interface for repository operations
- `DepartmentRepoServices`: Department-specific repository
- `StudentRepoServices`: Student-specific repository

5. Data Transfer Objects (DTOs)

- `LoginDTO`: Login credentials
- `RegisterDTO`: Registration information
- `StudentWithDeptName`: Student details with department name
- `DeptWithStdNames`: Department details with student names

6. Middleware and Filters

- `ExceptionHandlingMiddleware`: Global exception handling
- `CustomExceptionFilter`: Custom exception filtering
- `OverLoadStudentsResultFilter`: Limits number of students per department

Configuration

- Uses SQL Server as the database
- Connection string in `appsettings.json`
- Use dependency injection

Development Environment

- Uses .NET Core
- CORS policy allows all origins, headers, and methods (development configuration)

API Endpoints

Authentication

- `POST /api/Account/Register`: Register new user
- `POST /api/Account/Login`: Authenticate user
- `POST /api/Account/Role`: Create new role

Department

- `GET /api/Department`: List all departments
- `GET /api/Department/{id}`: Get department by ID
- `GET /api/Department/{name}`: Get department by name
- `POST /api/Department`: Add new department (Admin only)
- `PUT /api/Department`: Edit department (Admin only)
- `DELETE /api/Department/{id}`: Delete department (Admin only)

Student

- `GET /api/Student`: List all students
- `GET /api/Student/{id}`: Get student by ID
- `GET /api/Student/{name}`: Get student by name
- `POST /api/Student`: Add new student (Admin only)
- `PUT /api/Student`: Edit student (Student can edit own profile, Admin can edit any)
- `DELETE /api/Student/{id}`: Delete student (Admin only)