**Student Management Web API**

This document describes a web API for managing student and group data. The API is built with C#, .NET, EF6 and utilizes dependency injection principles. You can interact with the API and explore its functionality using Swagger UI.

Classes

StudentDbcontext: This class provides the foundation for interacting with the database tables for students and groups. It inherits from the DbContext class and likely has methods for data access and manipulation.

IStudentRepository, IGroupRepository, IAdminRepository, IAuthRepository, ITeacherRepository: These interfaces define the functionalities for working with student and group data. They act as abstractions, specifying the operations (like adding, retrieving, updating) without detailing the implementation.

StudentRepository, GroupRepository, AdminRepository, AuthRepository, TeacherRepository: These classes implement the respective interfaces, providing concrete logic for student and group data operations. They likely interact with StudentDbcontext to access the database.

Student, Group, Admin, Teacher, LoginUser, RefreshToken: These classes represent the data models for students and groups, with properties corresponding to the database table columns.

StudentInputDto, GroupInputDto, AdminInputDto, AuthResponseDto, LoginDto, TeacherInputDto: These classes likely represent data transfer objects (DTOs) used to receive student and group data from external sources like a user interface.

StudentOutputDto, GroupOutputDto, AdminOutputDto, TeacherOutputDto: These classes likely represent DTOs used to send student and group data back to external sources like the user interface.

StudentsController, GroupsController, AdminsController, AuthController, TeachersController: These classes handle user interactions related to students and groups. They likely depend on the repository interfaces to perform operations and inherit from a base controller class for common functionalities.

Relationships

StudentRepository, GroupRepository, AdminRepository, AuthRepository, TeacherRepository depend on StudentDbcontext for database access.

StudentsController, GroupsController, AdminsController, AuthController, TeachersController depend on the corresponding repository interfaces (IStudentRepository, IGroupRepository, IAdminRepository, IAuthRepository, ITeacherRepository) to interact with student and group data.

Dependency Injection

The use of interfaces suggests the application employs dependency injection. This means classes receive their dependencies (like repositories) through injection rather than creating them themselves. This promotes loose coupling and improves testability.

UML Diagram:  
A diagram of a computer program

Description automatically generated with medium confidence

UseCase Diagram:  
A diagram of a network

Description automatically generated

Swagger UI Integration

This web API offers Swagger UI integration, allowing you to explore the functionalities and try out API calls directly within your web browser. You can typically access Swagger UI by appending /swagger to the base URL of your web API.

Benefits of Swagger UI

API Documentation: Provides clear and interactive documentation of the API endpoints, including request parameters, response formats, and authentication details.

API Testing: Enables you to directly test API calls using various HTTP methods (GET, POST, PUT, DELETE) with different parameters and payloads. It also enables login and authorization vie JWT. This facilitates API development and troubleshooting.

Database Diagram:  
A screenshot of a computer

Description automatically generated

Conclusion

This web API offers a structured approach to managing student and group data with clear separation of concerns and potential for easy interaction through dependency injection and Swagger UI integration.