**.NET Project Conventions (3-Layer Architecture with EF Core and SQL Server)**

**1. Project Structure**

/GOCAP

|-- /GOCAP.Api

|-- /GOCAP.Api.Model

|-- /GOCAP.Api.Validation

|-- /GOCAP.Service

|-- /GOCAP.Domain

|-- /GOCAP.Repository

|-- /GOCAP.Database

|-- /GOCAP.Common

|-- /GOCAP.Test

|-- /GOCAP.ExternalService

|-- /GOCAP.Migrations

**2. Naming Conventions**

**2.1. General Naming**

* Use **PascalCase** for class names, method names, and properties.
* Use **camelCase** for local variables and method parameters.

**2.2. Entity Naming**

* Entity class names should be singular (e.g., UserEntity, StoryEntity).
* Use Entity suffix (e.g., UserEntity).

**2.3. Interface Naming**

* Prefix interfaces with I (e.g., IUserRepository).
* Repository interfaces should follow IEntityRepository<T> convention.

**2.4. Controller Naming**

* Use plural form (e.g., UsersController, StoriesController).
* Use Controller suffix (e.g., UsersController).

**2.5. Service Naming**

* Use Service suffix (e.g., UserService).
* Use Manager if the service performs complex logic (e.g., StoryManager).

**2.6. Model Naming**

* Use Model suffix (e.g., UserModel, StoryCreationModel).

**2.7. Repository Naming**

* Repository implementations should have Repository suffix (e.g., UserRepository).

**3. Database Conventions**

**3.1. Table Naming**

* Use **singular** form (e.g., Users, Stories).

**3.2. Column Naming**

* Primary keys should be Id
* Foreign keys should follow {entity}Id (e.g., Userid).

**3.3. Index Naming**

* Follow {table}\_idx\_{column} format (e.g., user\_idx\_email).

**4. Coding Conventions**

**4.1. Entity Framework Core**

* Use Fluent API for configurations instead of attributes.
* Use DbSet<T> for each entity in DbContext.
* Use AsNoTracking() for read-only queries.
* Use ValueObjects for reusable domain concepts.

**4.2. API and Routing**

* Use RESTful principles.
* Use [controller]s ex (“users”) for base routing.
* Use GET, POST, PATCH, PUT, DELETE methods properly.
* Use HttpStatusCode properly (200 OK, 201 Created, 400 Bad Request).

**4.3. Exception Handling**

* Use a global exception handler middleware.
* Do not expose stack traces in production.
* Use custom exception classes for domain-specific errors.

**4.4. Logging and Monitoring**

* Use ILogger<T> for structured logging.
* Store logs in structured format (JSON if needed).
* Use Application Insights or Serilog for logging.

**4.5. Dependency Injection**

* Use constructor injection for all dependencies.
* User [RegisterService] attribute to register one service or repository.

**4.6. Security**

* Use JWT for authentication.
* Use role-based or policy-based authorization.
* Store secrets in environment variables or Azure Key Vault.

**4.7. Testing**

Unit tests should follow {ClassName}Tests.cs naming.

* Use Moq for dependency mocking.
* Use xUnit or NUnit for testing.
* Cover **services** and **repositories** with unit tests.
* Cover **API endpoints** with integration tests.

**5. Git and CI/CD Conventions**

**5.1. Branch Naming**

* main: Production-ready.
* develop: Latest development.
* feature/{feature-name}: New features.
* bugfix/{issue-number}: Bug fixes.

**5.2. Commit Message Format**

[Type] Short description

Details about the change.

* feat: New feature.
* fix: Bug fix.
* refactor: Code refactoring.
* test: Adding or modifying tests.
* docs: Documentation update.

**5.3. Pull Request Guidelines**

* Use clear and descriptive titles.
* Link to issues (if applicable).
* Request at least one code review before merging.

**5.4. CI/CD Pipeline**

* Use GitHub Actions/Azure DevOps for CI/CD.
* Automate build, test, and deploy steps.
* Use environment-specific configurations.

**6. Performance and Optimization**

* Use caching for frequently accessed data (e.g., Redis).
* Use database indexes on frequently queried columns.
* Use asynchronous programming (async/await) where applicable.
* Optimize queries to reduce N+1 problems.

This convention guide ensures consistency and maintainability for .NET projects using a 3-layer architecture with EF Core and SQL Server.