Національний технічний університет України «КПІ»

Факультет інформатики та обчислювальної техніки

Кафедра Інформаційних систем та технологій

Лабораторна робота №3

з дисципліни « Сучасні технології розробки WEB-застосувань на платформі Microsoft.NET»

на тему: « Проектування REST веб-API»

Виконав:

студент гр. ІС-11

Іваніцький М.

Викладач:

Бардін В.

2023 рік

**Мета**:

1. Ознайомитися з основами створення REST веб-API та методологією

C4 для відображення архітектури системи.

2. Ознайомитися з основами створення ER-діаграм для представлення

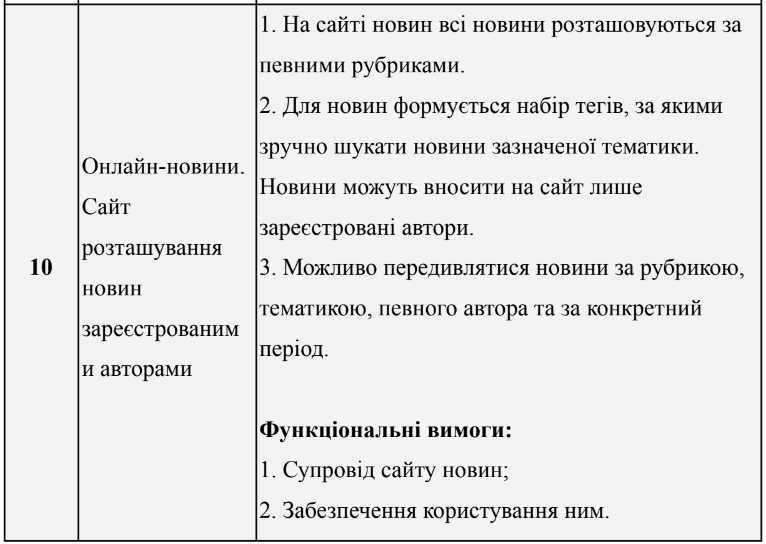
структури бази даних.

**Завдання:**

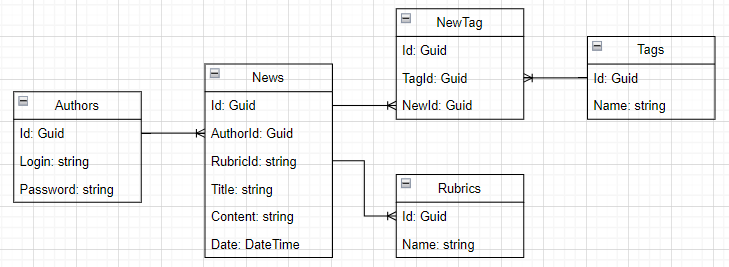
1. З дотриманням вимог REST-у спроектувати веб-API для обраної(згідно варіанту) доменної області, використовуючи методологію C4 для створення діаграми архітектури системи.

2. Створити ER-діаграму для DAL (Data Access Layer), яка відображатиме структуру бази даних веб-API.

3. Оформити спроєктоване рішення у вигляді звіту до лабораторної роботи.

**Варіант 10:**

**Хід роботи**

**ER-Діаграма:**

Таблиця: *Authors*

Призначення: містить інформацію про зареєстрованих авторів

Поля:

* Id: Guid – унікальний ідентифікатор
* Login: string – ім’я/логін
* Password: string – пароль

Таблиця: *News*

Призначення: містить інформацію про всі опубліковані новини

Поля:

* Id: Guid – унікальний ідентифікатор
* AuthorId: Guid – унікальний ідентифікатор автора
* RubricId: Guid – унікальний ідентифікатор рубрики
* Title: string – назва новини
* Content: string – власне новина
* Date: DateTime – дата створення новини

Таблиця: *Rubrics*

Призначення: містить інформацію про наявні новинні рубрики

Поля:

* Id: Guid – унікальний ідентифікатор
* Name: string – назва рубрики

Таблиця: *Tags*

Призначення: містить інформацію про наявні теги

Поля:

* Id: Guid – унікальний ідентифікатор
* Name: string – назва тегу

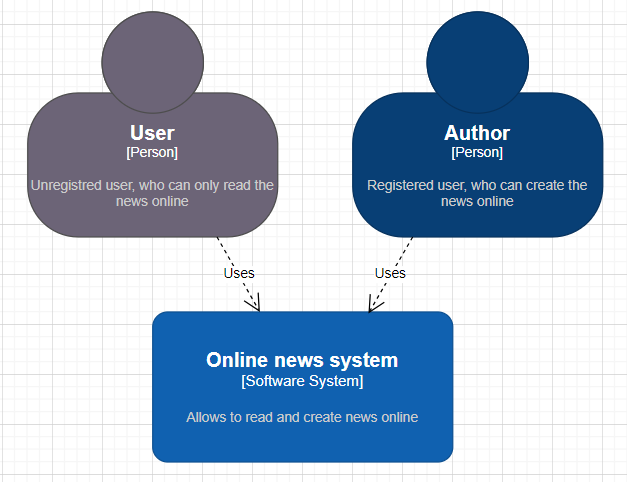
Таблиця: *NewTag*

Призначення: містить інформацію про наявні новинні рубрики

Поля:

* Id: Guid – унікальний ідентифікатор
* NewsId: Guid – унікальний ідентифікатор новини
* TagId: Guid – унікальний ідентифікатор тегу

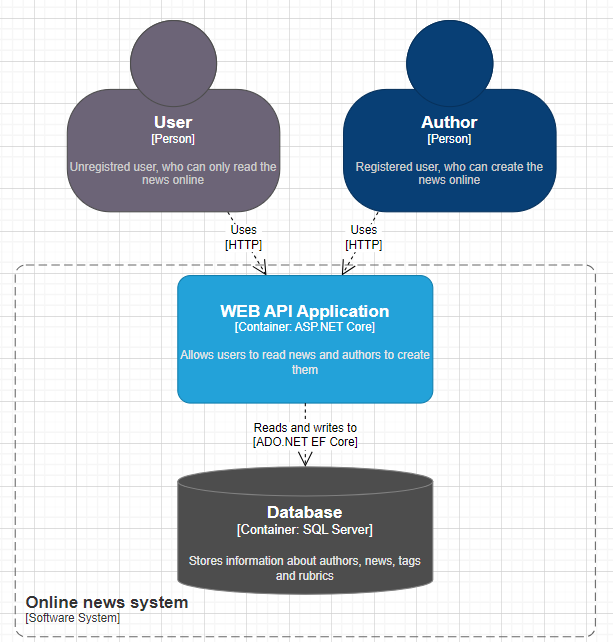
**Методологія С4:**

**Сontext:**

Система онлайн-новин має 2 типи користувачів:

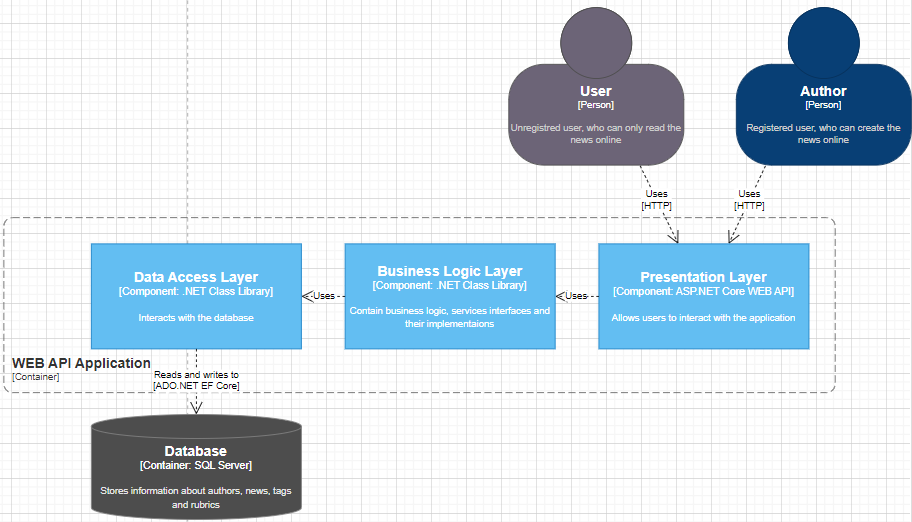
*User* – неавторизований, який може тільки читати новини, виділяючи за рубриками та тегами.

*Author* – авторизований, може також створювати власні новини.

**Container:**

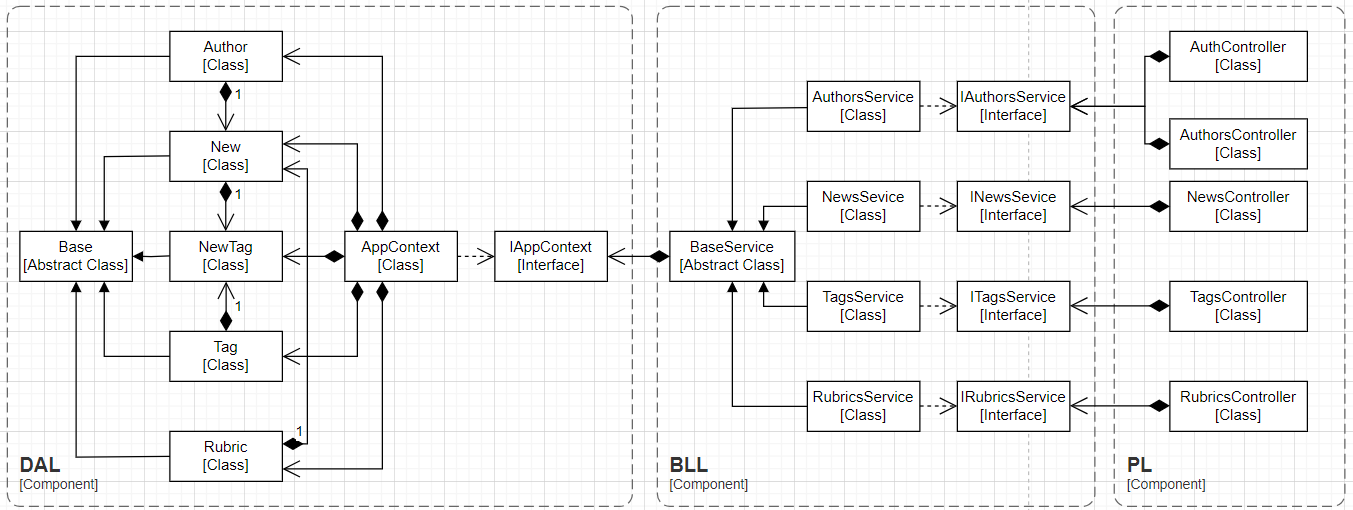
Система онлайн новин поділяється на дві частини:

1. *WEB API Застосунок,* який реалізує весь функціонал за допомогою ASP.NET
2. *Database,* яка зберігає всю потрібну інформацію.

**Component:**

При розробці архітектури застосунку була застосована багатошарова архітектура, яка містить наступні шари:

1. Presentation Layer – відповідає за взаємодію з користувачем та представлення інформації, містить контролери.
2. Business Logic Layer – визначає логіку операцій та обробку даних, містить інтерфейси сервісів та їх імплементацію.
3. Data Access Layer – відповідає за взаємодію з базою даних та місить контекст бази даних.

**Code:**

**Опис WEB-API:**

Web-API матмие башатошарову архітектуру, як було описано вище. DAL містить класи основних сутностей доменної області а також клас AppContext, який відповідатиме за зв’язок із базою даних, реалізованою у програмному середовищі Miscrosoft SQL Server.

BLL відповідає за виконання бізнес-логіки додатку та містить абстрактний клас BaseService, який буде батьківським для специфічних класів, в яких будуть зберігатись запити до відповідних сутностей БД: AuthorService, NewsService, TagsService, RubricService. Ці ж класи реалізовуватимуть специфічні інтерфейси для створення слабкої залежності між шаром представлення та шаром бізнес-логіки.

PL буде представлений у вигляді реалізації API. Цей шар міститиме класи AuthController (відповідає за автентифікацію автора або реєстрації нового автора, які будуть працювати за допомогою Identity) та контролери під кожну нашу сутність.

**Ендпоінти:**

*AuthController:*

1. POST /api/auth/register/{authorInfo} - Реєстрація нового користувача
2. POST /api/auth/login - Аутентифікація користувача

*AuthorController:*

1. GET /api/authors - Отримати список всіх зареєстрованих авторів
2. GET /api/authors/{authorId} - Отримати інформацію про конкретного автора

*NewsController:*

1. POST /api/news - Додати нову новину (Аутентифікація)
2. GET /api/news - Отримати список всіх новин
3. GET /api/news/{newsId} - Отримати інформацію про конкретну новину
4. GET /api/news/by-tag/{tagId} - Отримати список новин за певним тегом
5. GET /api/news/by- rubric/{rubricId} - Отримати список новин за певною рубрикою

*TagsController:*

1. GET /api/tags - Отримати список всіх тегів новин
2. POST /api/tags/{tagInfo} - Додати новий тег (Аутентифікаці)

*RubricsController:*

1. GET /api/rubrics - Отримати список всіх тегів новин
2. POST /api/rubrics/{rubricInfo} - Додати нову рубрику (Аутентифікаці)

**Посилання на діаграми:**

<https://drive.google.com/file/d/1s5ekfStsLzH2CQzpALb1_GRy42wT3HpN/view?usp=sharing>

**Посилання на репозиторій:**

<https://github.com/Maksvell/dotNET_WEB_lab3-4>