

Національний технічний університет України  
«КПІ ім. Ігоря Сікорського»  
Факультет Інформатики та Обчислювальної Техніки  
Кафедра Інформаційних Систем та Технологій

## Комп'ютерний практикум №3

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

на тему

Проектування REST веб-API»

Виконав:  
студент гр. ІС-11  
Побережний Олександр

Київ 2023

## Завдання:

### Теоретична частина:

1. Ознайомитися з основами створення REST веб-API та методологією С4 для відображення архітектури системи.
2. Ознайомитися з основами створення ER-діаграм для представлення структури бази даних.

### Практична частина:

1. З дотриманням вимог REST-у спроектувати веб-API для обраної(згідно варіанту) доменної області, використовуючи методологію С4 для створення діаграми архітектури системи.
2. Створити ER-діаграму для DAL (Data Access Layer), яка відображатиме структуру бази даних веб-API.
3. Оформити спроектоване рішення у вигляді звіту до лабораторної роботи.

### Документація:

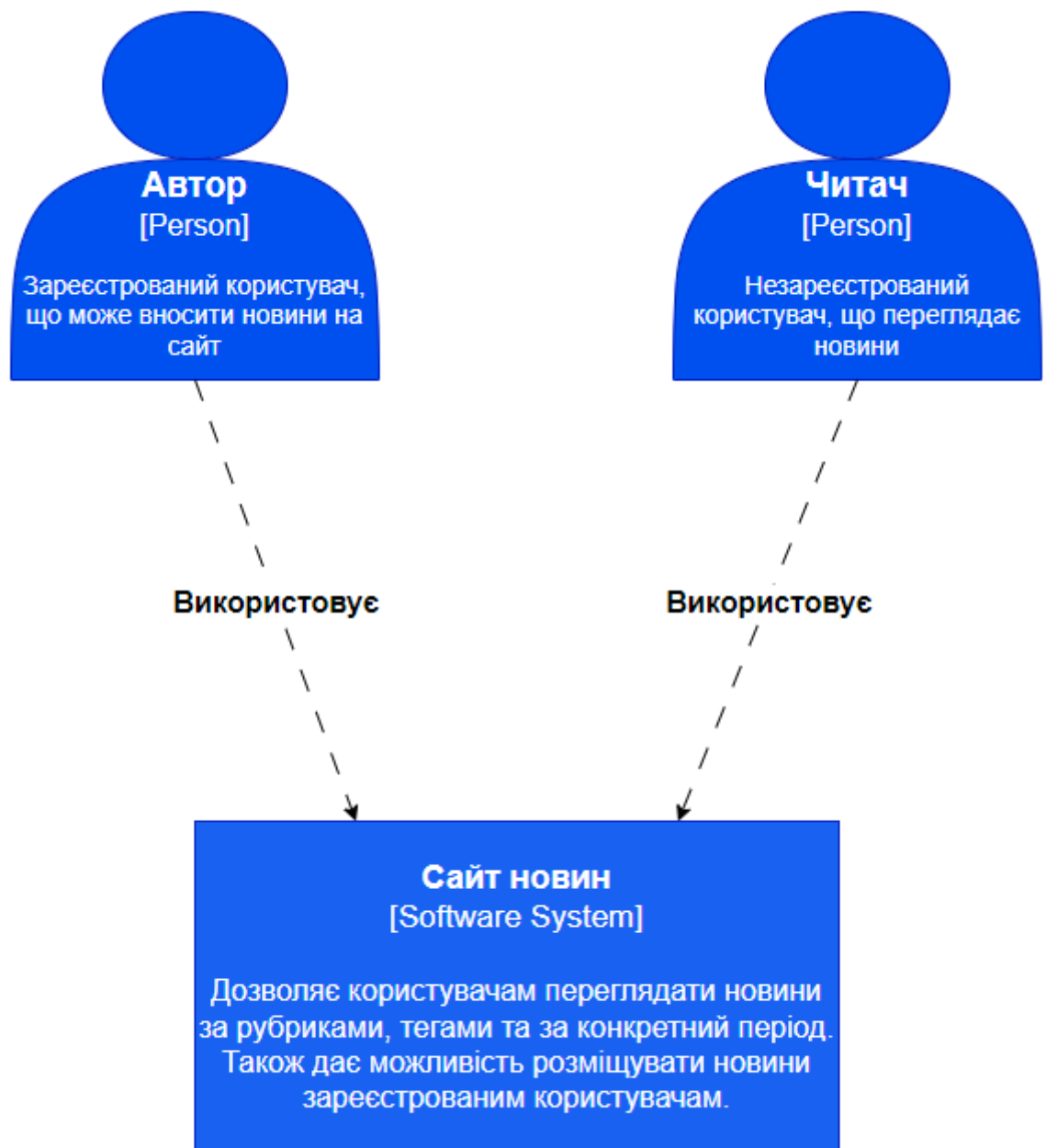
1. Підготувати документацію(звіт до ЛР), яка включатиме опис веб-API, а також структуру бази даних з урахуванням ER-діаграми.

### Варіант – 10:

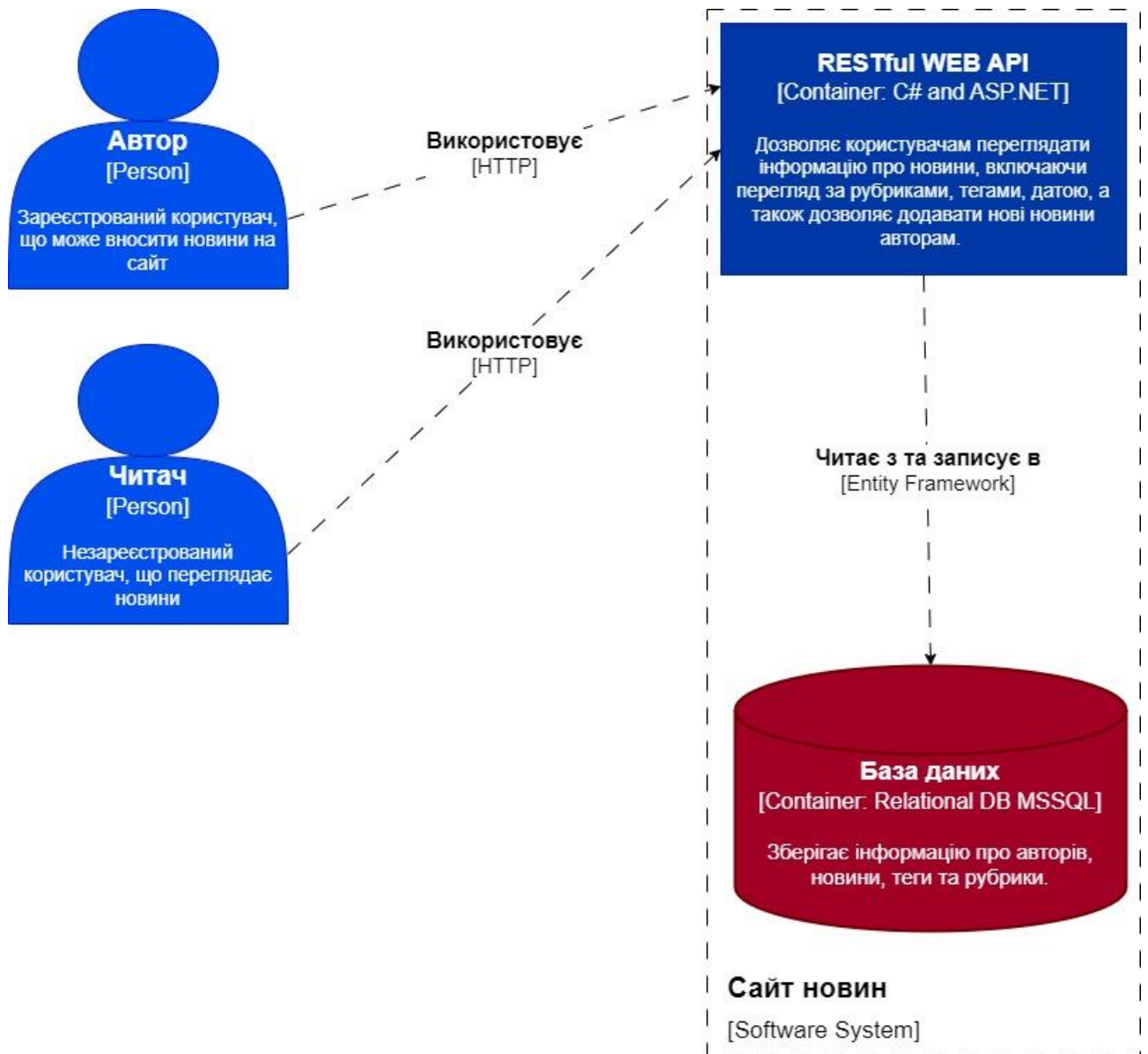
10	Онлайн-новини. Сайт розташування новин зареєстрованим и авторами	<p>1. На сайті новин всі новини розташовуються за певними рубриками.</p> <p>2. Для новин формується набір тегів, за якими зручно шукати новини зазначеної тематики. Новини можуть вносити на сайт лише зареєстровані автори.</p> <p>3. Можливо передивлятися новини за рубрикою, тематикою, певного автора та за конкретний період.</p> <p><b>Функціональні вимоги:</b></p> <p>1. Супровід сайту новин;</p> <p>2. Забезпечення користування ним.</p>
----	---	--

# Діаграма архітектури системи (за методологією С4)

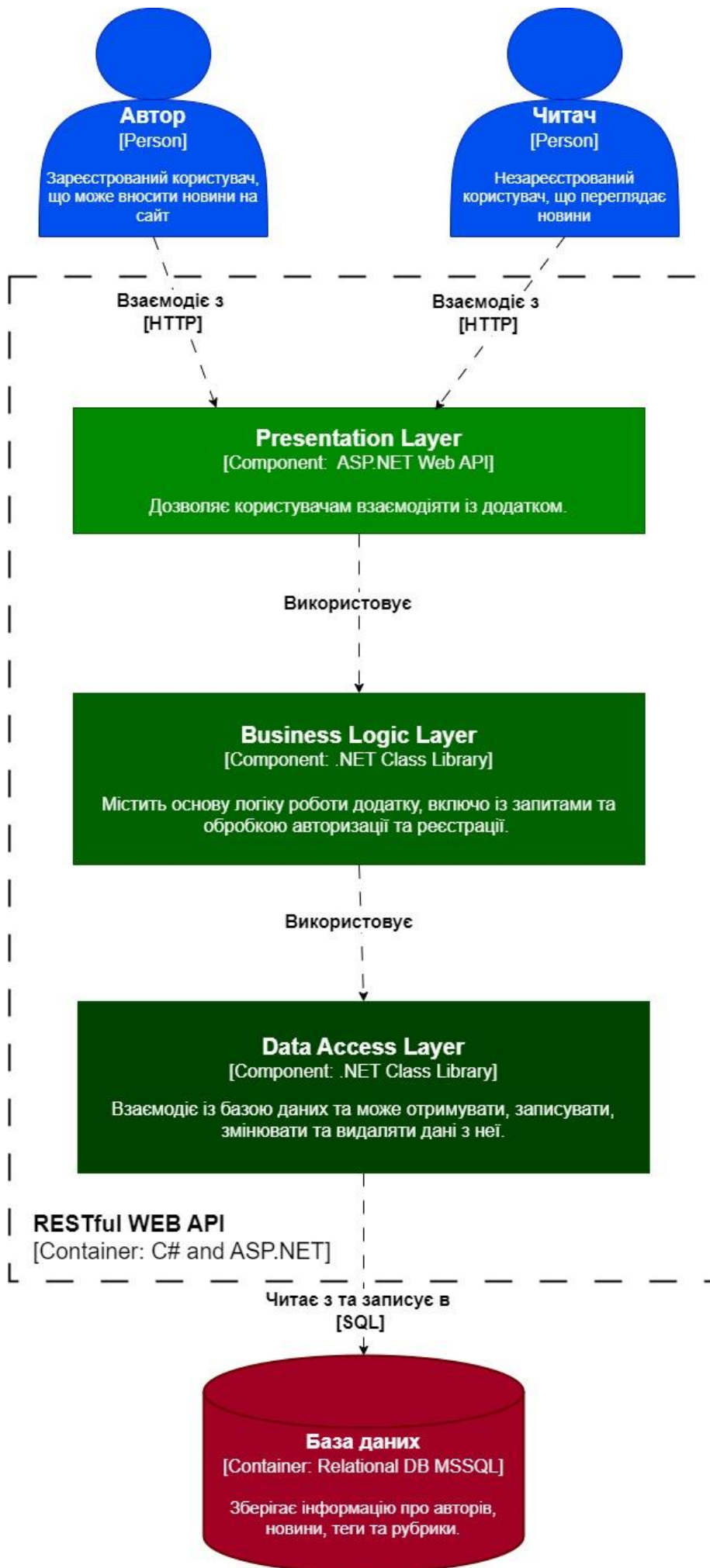
## 1. Діаграма контексту



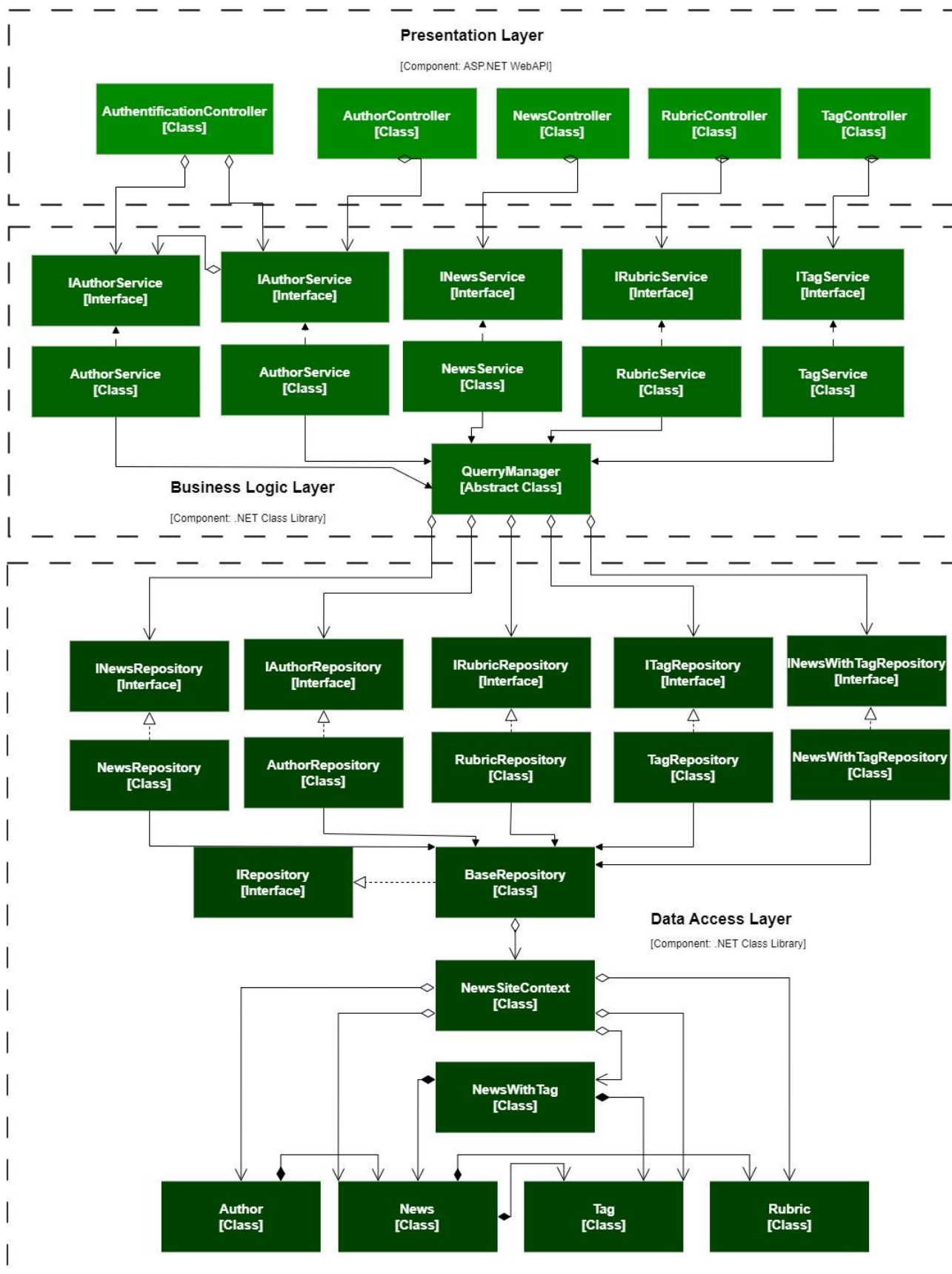
## 2. Діаграма контейнерів



### 3. Діаграма компонентів



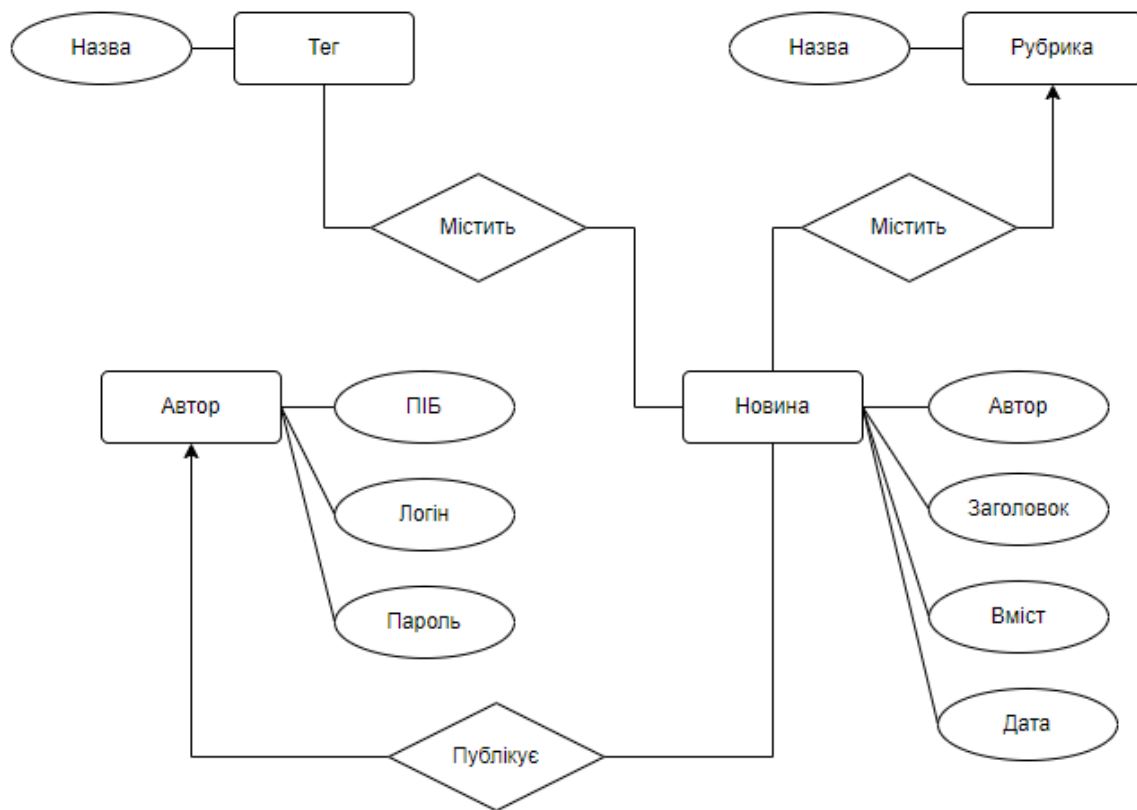
#### 4. Діаграма коду



Вихідні дані діаграм доступні за посиланням:

<https://drive.google.com/file/d/1eSLxJyd389Uj1szWPK4xLhbOVNODyyycv/view?usp=sharing>

## ER-діаграма для DAL



Якщо говорити про базу даних, то вона міститиме такі таблиці:

Author – таблиця для зберігання сутностей авторів. Містить поля:

- Id (guid)– унікальний ідентифікатор.
- Name (string) – ім'я автора.
- Login (string) – логін автора.
- Password (string) – захешований пароль.

News – сутність для зберігання інформації про новини. Містить поля:

- Id (guid) – ідентифікатор новини.

- Header (string) – заголовок статті.
- AuthorId (guid) – ідентифікатор автора.
- RubricId (guid) – ідентифікатор рубрики.
- Data (string) - текст новини.
- Date (DateTime) – дата створення новини.

Tag – сутність для зберігання інформації про теги. Містить поля:

- Id (guid) – ідентифікатор тегу.
- Name (string) – назва тегу.

NewsWithTag – сутність для реалізації зв'язку «багато-до-багатьох» між сутностями News та Tag. Містить поля:

- NewsId (guid) – ідентифікатор новини.
- TagId (guid) – ідентифікатор тегу.

Rubric – сутність для зберігання даних про рубрику. Має поля:

- Id (guid) – ідентифікатор рубрики.
- Name (string) – назва рубрики.

## **Опис web-API**

Web-API матиме N-layered архітектуру з такими шарами, як DAL, BLL та PL(API).

DAL містить класи основних сутностей доменної області а також клас NewsSiteContext, який відповідатиме за зв'язок із базою даних, реалізованою у програмному середовищі Microsoft SQL Server 2022. Даний клас реалізує паттерн Unit Of Work шляхом наслідування від базового класу DbContext, також в ньому буде використано паттерн Repository, у вигляді DbSet<T>.

BLL відповідає за виконання бізнес-логіки додатку та містить абстрактний клас QueryManager, який буде батьківським для специфічних класів, в яких будуть зберігатись запити до відповідних сутностей БД (AuthorQueryManager,



NewsQueryManager, TagQueryManager, RubricQueryManager). Ці ж класи реалізовуватимуть специфічні інтерфейси для створення слабкої залежності між шаром представлення та шаром бізнес-логіки (IAuthorQueryManager, INewsQueryManager, ITagQueryManager, IRubricQueryManager).

PL буде представлений у вигляді реалізації API. Цей шар міститиме класи AuthenticationController (відповідає за реалізацію автентифікації автора або реєстрації нового автора), AuthorController (відповідає за функції додавання новин, редагування новин та видалення новин для конкретного автора), NewsController (відповідає за виведення новин за рубриками та тегами).

## **Ендпоінти**

### **NewsController:**

- [GET] /news – показ всіх новин за рубриками.
- [POST] /news – публікація новини автором.
- [GET] /news/rubrics/{id} – показ новин за певною рубрикою.
- [GET] /news/tags/{id} – показ новин за певним тегом.
- [GET] /news/authors/{id} – перегляд новин конкретного автора.
- [GET] /news/dates/{datefrom}&{dateto} – перегляди новин за датою створення.
- [DELETE] /news/{id}/authors/{id} – видалення певної новини певного автора.
- [PUT] /news/{id}/authors/{id} – редагування певної новини певного автора.

### **AuthenticationController:**

- [POST] /register – реєстрація нового автора.
- [POST] /login – автентифікація автора.

### **RubricController:**

- [POST] /rubrics – додавання рубрики.

- [GET] /rubrics – перегляд всіх рубрик.
- [GET] /rubrics/{id} – перегляд певної рубрики.
- [PUT] /rubrics/{id} – оновлення назви рубрики.
- [DELETE] /rubrics/{id} – видалення рубрики.

### **TagController:**

- [POST] /tags – додавання тегу.
- [GET] /tags – перегляд всіх тегів.
- [GET] /tags/{id} – перегляд певного тегу.
- [PUT] /tags/{id} – оновлення назви тегу.
- [DELETE] /tags/{id} – видалення тегу.

### **AuthorController:**

- [GET] /authors – перегляд всіх авторів.
- [GET] /authors/{id} – перегляд певного автора.
- [PUT] /authors/{id} – оновлення даних певного автора.
- [DELETE] /authors/{id} – видалення певного автора.

**Висновок:** у ході виконання лабораторної роботи, було проведено проєктування RESTful веб-API для доменної області сайту новин, використовуючи методологію С4. Головною метою цього завдання було створення ефективної та розширюваної системи для управління новинами, яку можна легко інтегрувати та розширювати. Спочатку було проведено аналіз вимог та ідентифіковано ключові аспекти функціональності системи. На основі цього аналізу були визначені компоненти системи та їх взаємодії, які були відображені на діаграмах С4. Архітектурна діаграма контексту надає високорівневий огляд системи та її зовнішніх взаємодій. Діаграма контейнерів визначає окремі компоненти системи та їх взаємодії, концентруючись на взаємодії між веб-сервером, додатком та базою даних. Детальніше, діаграма компонентів надає уявлення про внутрішню структуру кожного контейнера, визначаючи конкретні компоненти та їх взаємодії.

У результаті проведених робіт стало очевидним, що застосування методології C4 дозволяє ефективно моделювати архітектуру системи, роблячи її зрозумілою та легко розширюваною. Розроблене RESTful веб-API має потенціал для успішної реалізації функціоналу сайту новин, забезпечуючи зручний інтерфейс для взаємодії з даними та їх управління.