

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»



Кафедра інформаційних систем та технологій

Лабораторна робота №3

з дисципліни «Розробка програмного забезпечення на платформі .Net»

на тему:

«Проектування REST веб-API»

Викладач:
Бардін В.

Виконав:
Студент групи ІС-12

Канупа Максим

Київ – 2023

Завдання:

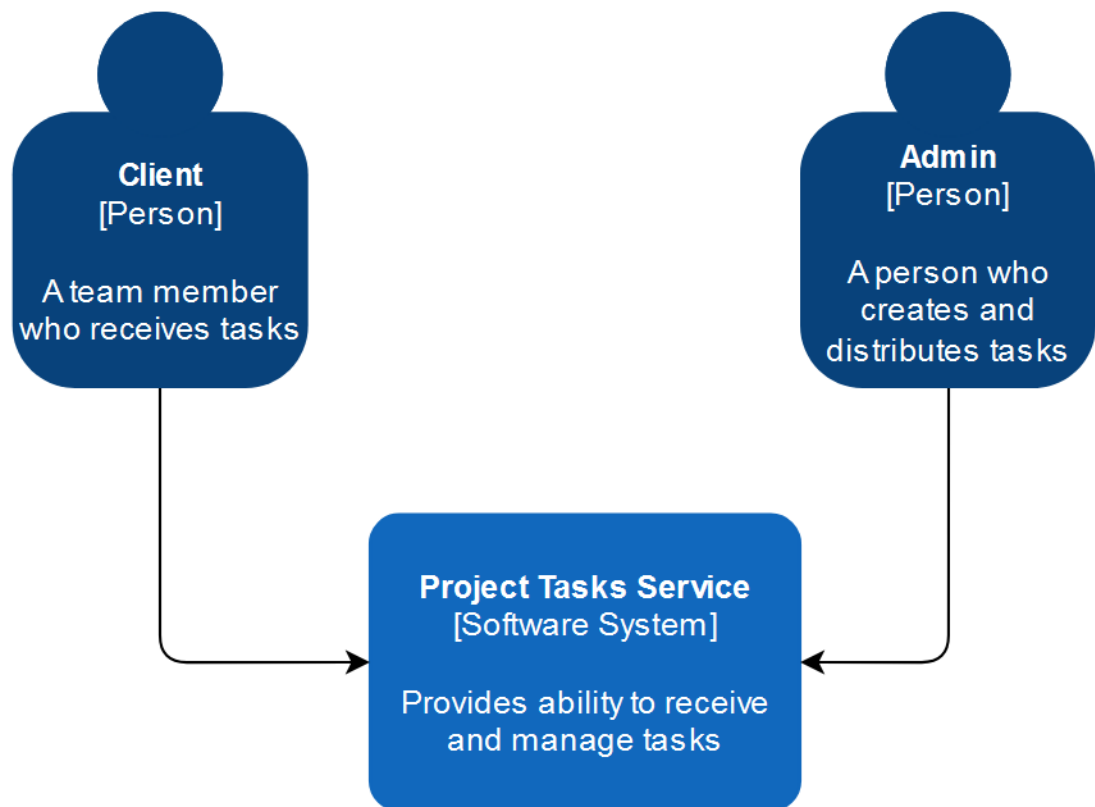
1. З дотриманням вимог REST-у спроектувати веб-API для обраної(згідно варіанту) доменної області, використовуючи методологію С4 для створення діаграми архітектури системи.
2. Створити ER-діаграму для DAL (Data Access Layer), яка відображатиме структуру бази даних веб-API.
3. Оформити спроектоване рішення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

Варіант:

7	Планувальник завдань. Розподіл та контроль за виконанням завдань членами команди проекту	<ol style="list-style-type: none">1. При виконанні проекту постає низка завдань, які потребують їх виконання.2. Завдання має опис його сутності, час, необхідний для його виконання, та пріоритет.3. Завдання перебуває в одному зі статусів: не розпочато, на виконанні, виконано.4. Команда проекту складається з робітників, яким розподіляються завдання на виконання відповідно до їхньої зайнятості. <p>Функціональні вимоги:</p> <ol style="list-style-type: none">1. Планування завдань проекту;2. Контроль за їх виконанням.
---	---	--

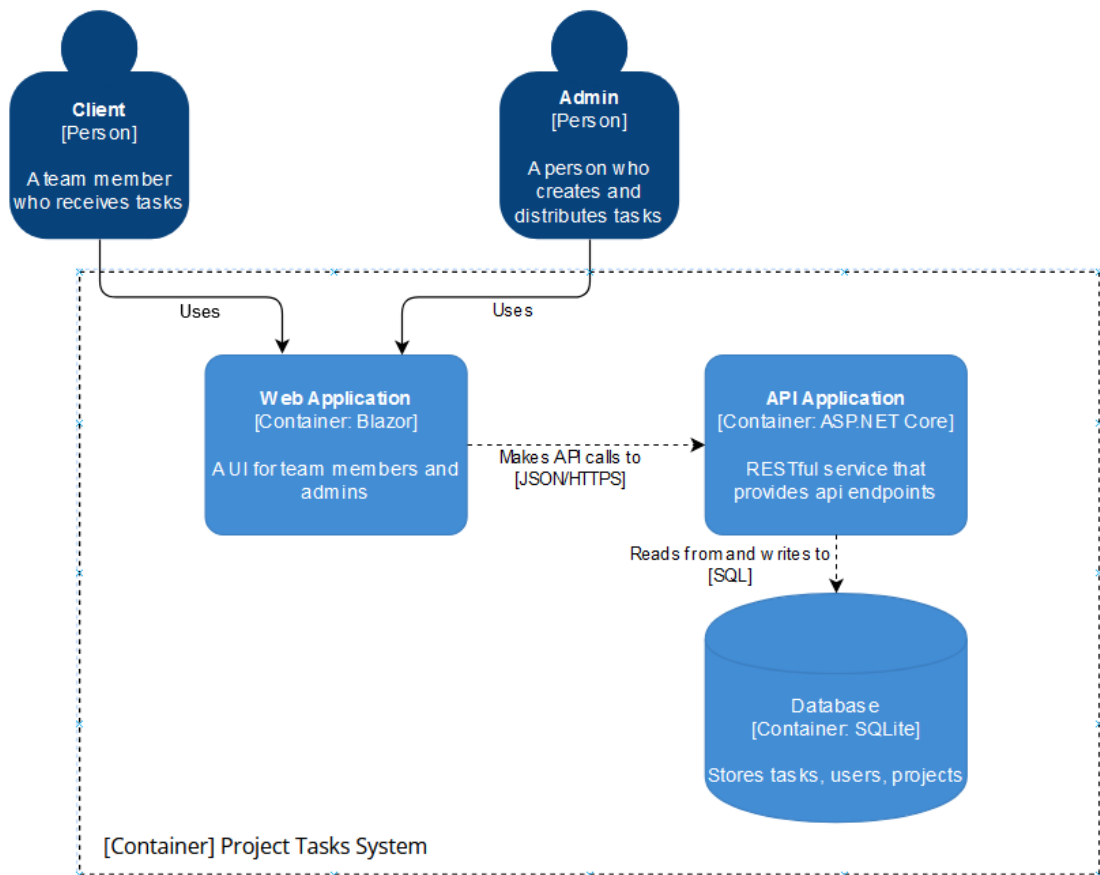
1. З дотриманням вимог REST-у спроектувати веб-АРІ для обраної(згідно варіанту) доменної області, використовуючи методологію С4 для створення діаграми архітектури системи.

Level 1. Context diagram



Існують два типи користувачів: клієнт (член команди проекту, що отримує завдання) і адмін (має можливість створювати задачі і розподіляти між проектами).

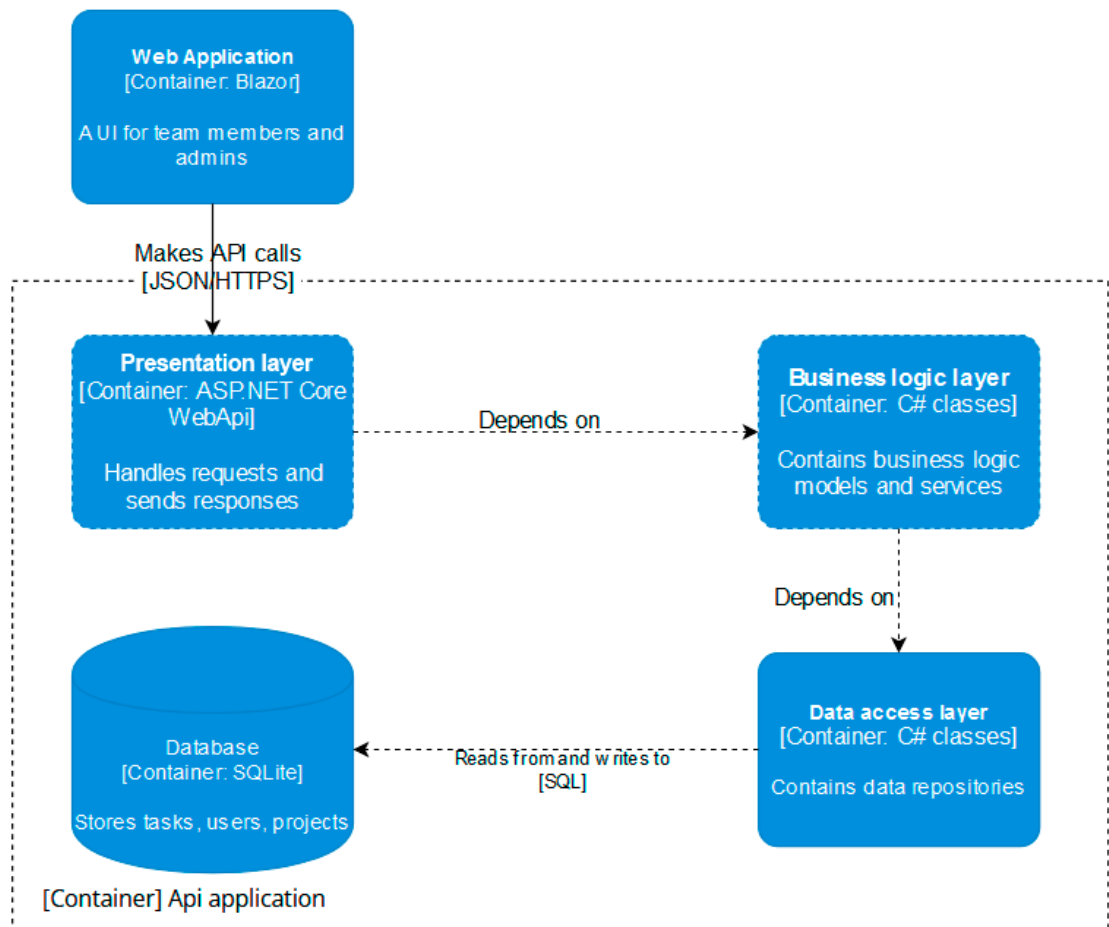
Level 2. Container diagram



Система планування задач складається з 3х контейнерів:

- Веб-застосунку на Blazor, що представляє веб-інтерфейс застосунку
- Серверна програма, яка реалізує функціонал REST api
- База даних

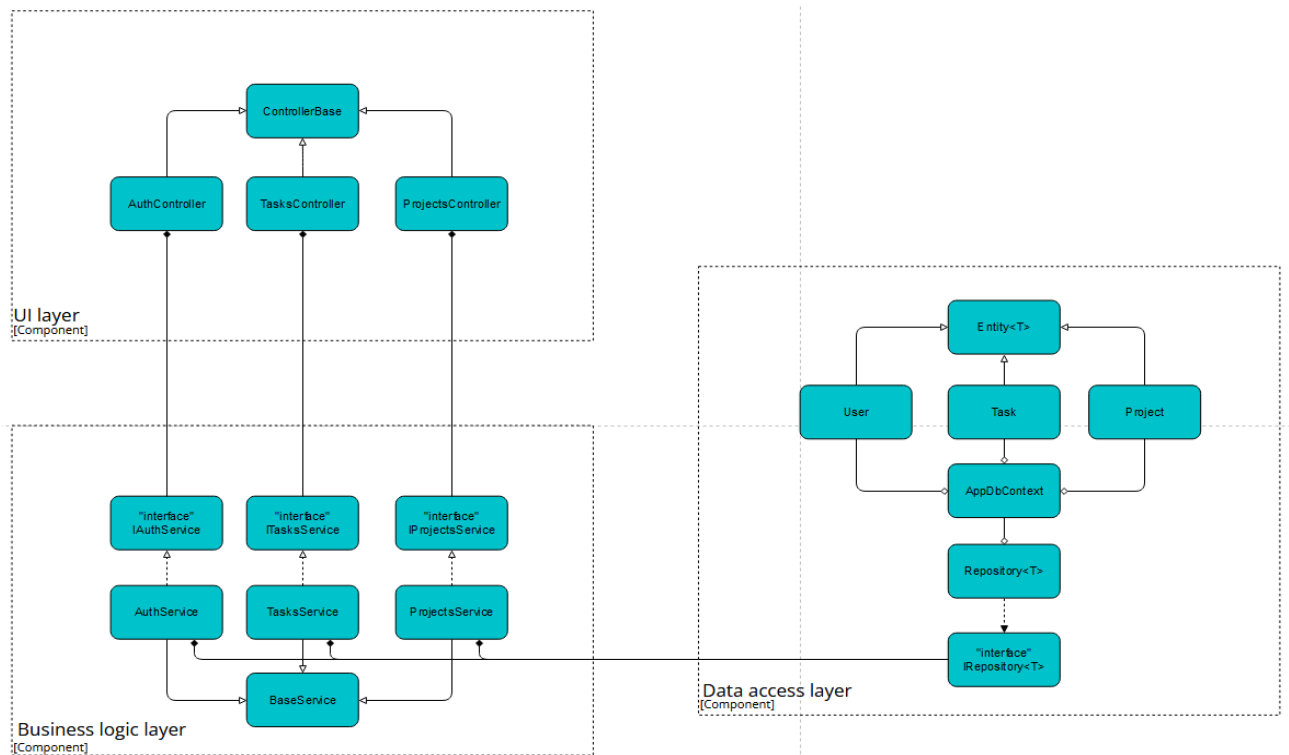
Level 3. Components diagram



Для розробки застосунку використовується N-layer архітектура, яка поділяє застосунок на 3 шари:

- **Presentation Layer** (Шар представлення): Цей рівень відповідає за взаємодію з користувачем та представлення інформації. Він містить обробку введення. Тут знаходяться контролери.
- **Business logic layer** (Шар бізнес-логіки): Цей рівень містить бізнес-логіку програми, яка визначає логіку операцій та обробку даних. Містить інтерфейси сервісів та їхню імплементацію.
- **DAL** (Шар доступу до даних): Цей рівень відповідає за доступ до даних та взаємодію з базою даних

Level 4. Code diagram



Auth Controller:

1. POST /api/auth/register
 - Опис: Реєстрація нового користувача.
2. POST /api/auth/login
 - Опис: Аутентифікація та генерація токена для користувача.

Tasks Controller:

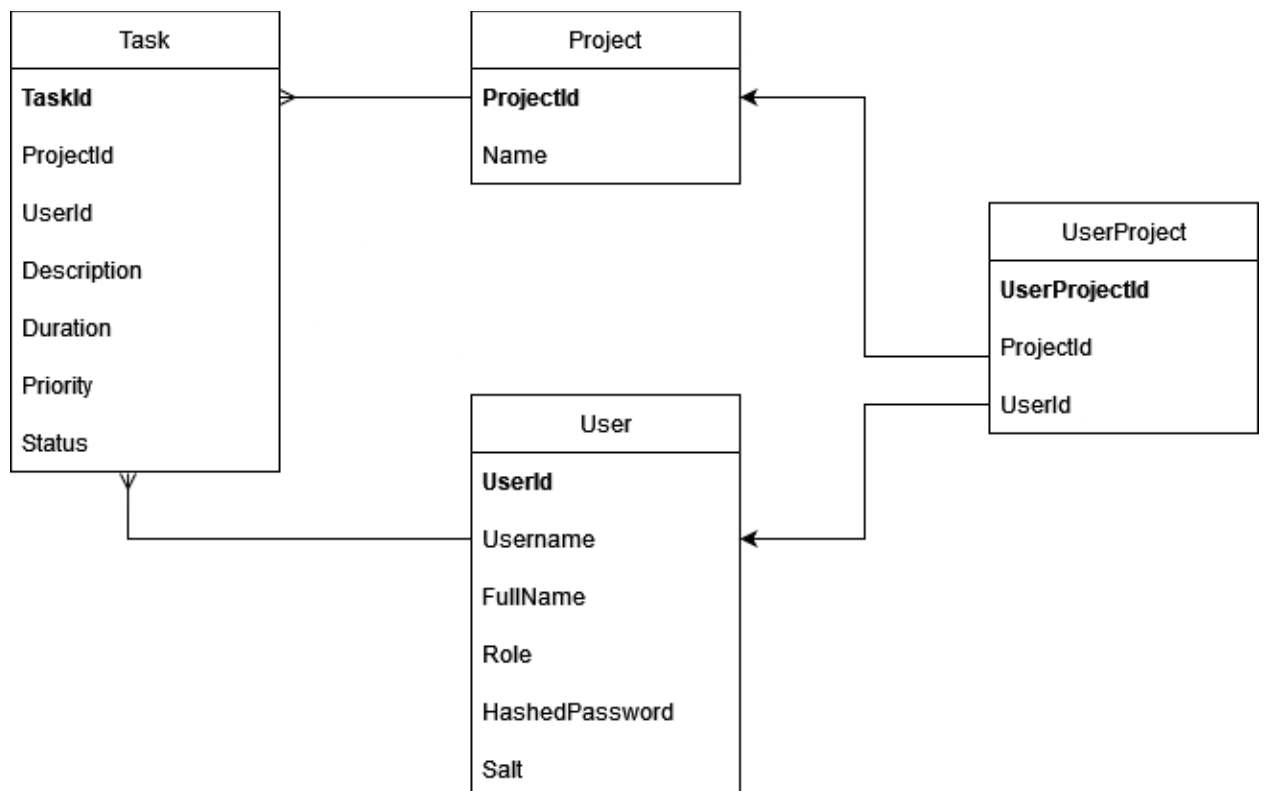
1. GET /api/tasks
 - Опис: Отримати список всіх завдань.
2. GET /api/tasks/{taskId}
 - Опис: Отримати деталі конкретного завдання за його ідентифікатором.
3. POST /api/tasks
 - Опис: Створити нове завдання для проекту.
4. PUT /api/tasks/{taskId}
 - Опис: Оновити деталі конкретного завдання за його ідентифікатором.
5. PUT /api/tasks/{taskId}/setUser
 - Опис: змінити виконавця завдання
6. DELETE /api/tasks/{taskId}

- Опис: Видалити завдання з проекту.

Projects Controller:

1. GET /api/projects
 - Опис: Отримати список проектів.
2. GET /api/projects/{projectId}
 - Опис: Отримати деталі конкретного проекту за його ідентифікатором.
3. POST /api/projects
 - Опис: Створити новий проект.
4. PUT /api/projects/{projectId}
 - Опис: Оновити деталі конкретного проекту за його ідентифікатором.
5. DELETE /api/projects/{projectId}
 - Опис: Видалити проект.
6. GET /api/projects/{projectId}/tasks
 - Опис: Отримати список завдань, пов'язаних із конкретним проектом.

ER-діаграма



Таблиця: Task

Призначення: Зберігає інформацію про завдання.

Поля:

- TaskId (Guid): унікальний ідентифікатор завдання.
- ProjectId (Guid): ідентифікатор проекту, до якого відноситься завдання.
- UserId (Guid): ідентифікатор користувача, який відповідає за завдання.
- Description (string): опис завдання.
- Duration (int): тривалість виконання завдання.
- Priority (int): пріоритет завдання.
- Status (string): поточний стан завдання.

Таблиця: Project

Призначення: Зберігає інформацію про проекти.

Поля:

- ProjectId (Guid): унікальний ідентифікатор проекту.
- Name (string): назва проекту.

Таблиця: User

Призначення: Зберігає інформацію про користувачів системи.

Поля:

- UserId (Guid): унікальний ідентифікатор користувача.
- Username (string): ім'я користувача.
- FullName (string): повне ім'я користувача.
- Role (int): роль користувача в системі.
- HashedPassword (string): хеш пароля користувача для забезпечення безпеки.
- Salt (string): "сіль" для ускладнення процесу хешування пароля.

Таблиця: UserProject

Призначення: Встановлює зв'язок багато до багатьох між користувачами і проектами.

Поля:

- UserProjectId (Guid): унікальний ідентифікатор зв'язку користувача і проекту.
- UserId (Guid): ідентифікатор користувача.
- ProjectId (Guid): ідентифікатор проекту.