



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

**ЗВІТ**  
**ДО ЛАБОРАТОРНОЇ РОБОТИ №4**  
з дисципліни  
«Сучасні технології розробки WEB-застосувань на платформі Microsoft.NET»  
студента III курсу ФІОТ групи ІК-12  
Басюка Валентина

Перевірив:  
Бардін Владислав

Київ 2023

## ЛАБОРАТОРНА РОБОТА №4

### Імплементація REST API

#### Варіант №1

#### Завдання:

Практична частина:

1. Використовуючи архітектуру розроблену в попередній роботі Імплементувати REST веб-API, використавши N Layered архітектуру.
2. Покрити модульними та інтеграційними тестами основні компоненти рішення.
3. Експортувати розроблене API до Postman.

Варіант	Предметна галузь	Функціональні вимоги
1	Ресторан. Формування замовлень.	<p>1. Страви складаються із інгредієнтів. Інгредієнти можуть складати різні страви. Страви складають прайсліст, в якому вказано ціну для різних порцій страви.</p> <p>2. Замовлення може містити в собі набір декількох порцій різних страв.</p> <p><b>Функціональні вимоги:</b></p> <p>1. Складання страв та меню; 2. Формування замовлень.</p>

## Репозиторій з імплементацією REST API:

<https://github.com/Valyon450/OrderingRestaurant/tree/develop-Valyon/Restaurant.Backend>

### Хід роботи

Імplementована Backend частина для бізнес-задачі OrderingRestaurant у вигляді REST API базується на N-Layer архітектурі з трьома шарами: Data Access (доступ до даних), Business Logic (бізнес-логіка) та Presentation (презентаційний шар). Використовується ASP.NET, Entity Framework (EF) Core з підходом Code-First для роботи з базою даних.

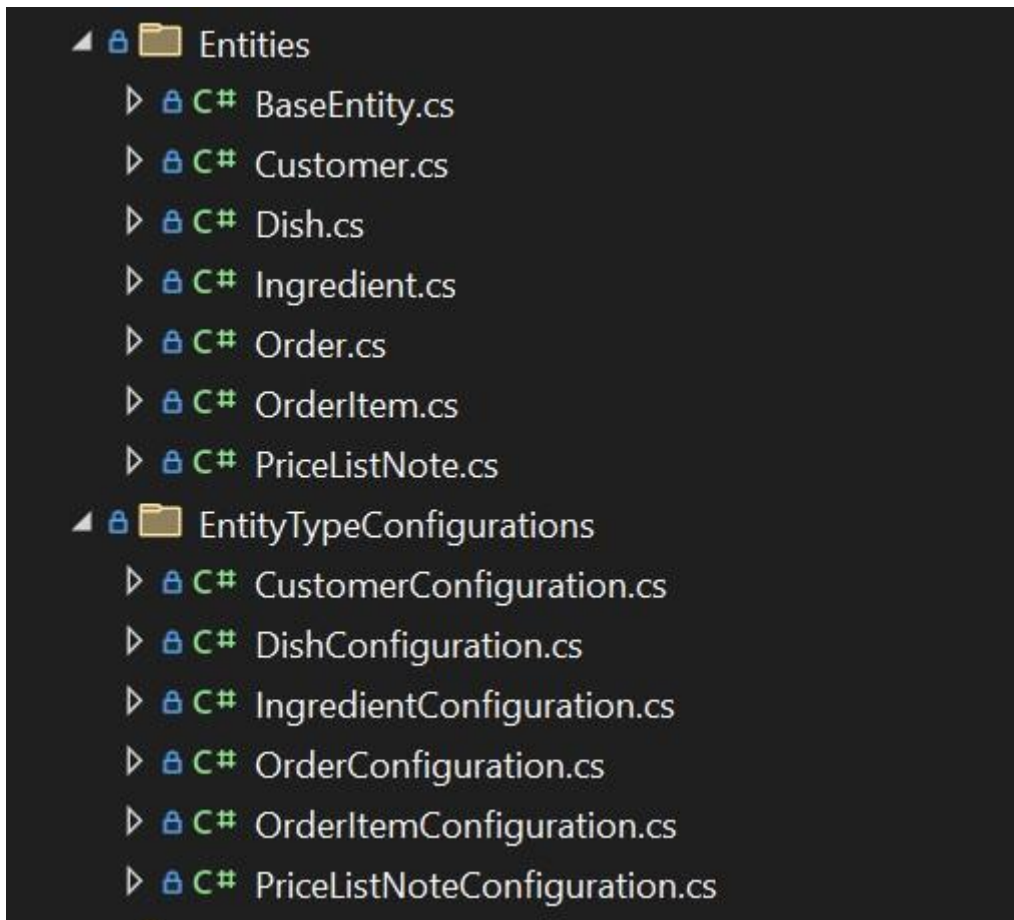
Data Access Layer (Шар доступу до даних):

DbContext:

DbContext визначає зв'язок із базою даних та містить набір DbSet для кожної сутності.

Entity Configuration:

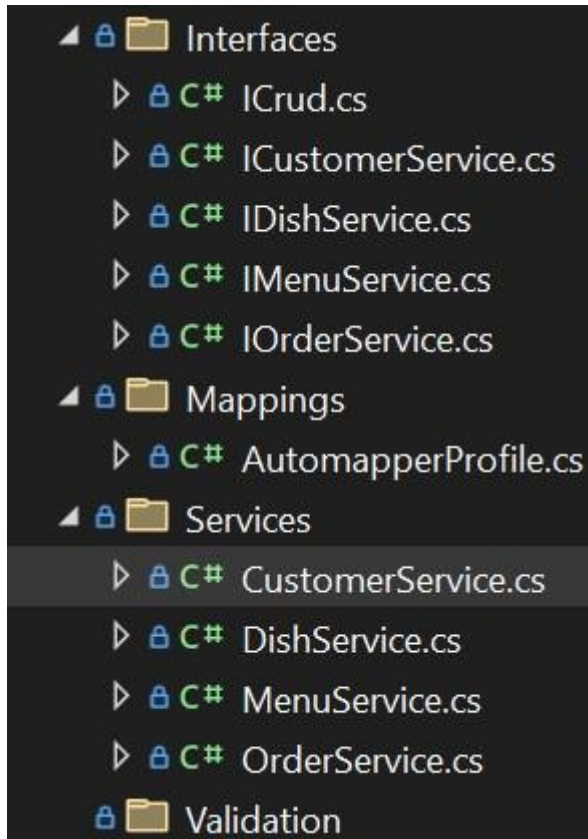
Для кожної сутності створено окремий клас конфігурації для визначення властивостей, відносин і інших налаштувань.



## Business Logic Layer (Бізнес-логіка):

### Сервіси:

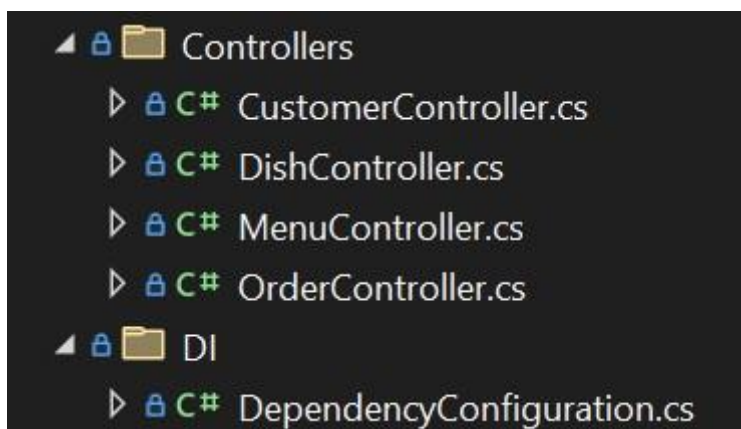
Створено сервіси для виконання бізнес-логіки, такі як обробка замовлень, робота зі стравами, меню тощо.



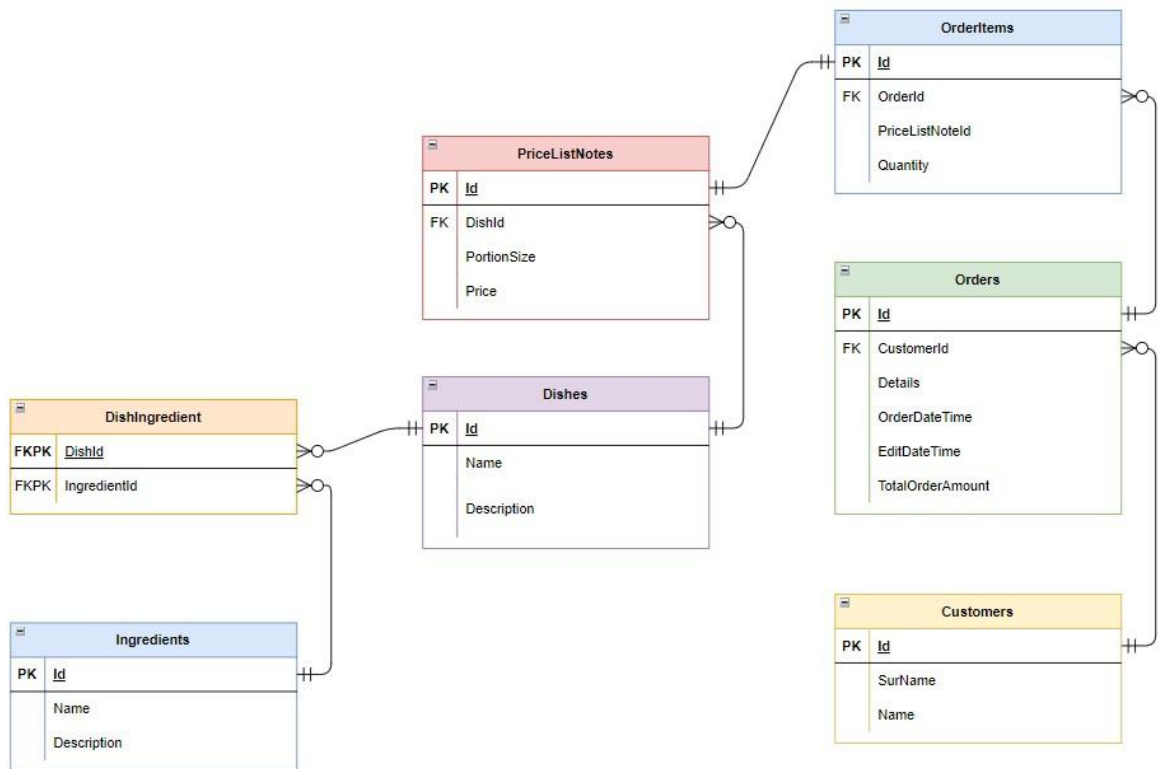
## Presentation Layer (Презентаційний шар):

### Контролери:

Для обробки запитів контролери створено контролери, які в свою чергу взаємодіють з бізнес-логікою.



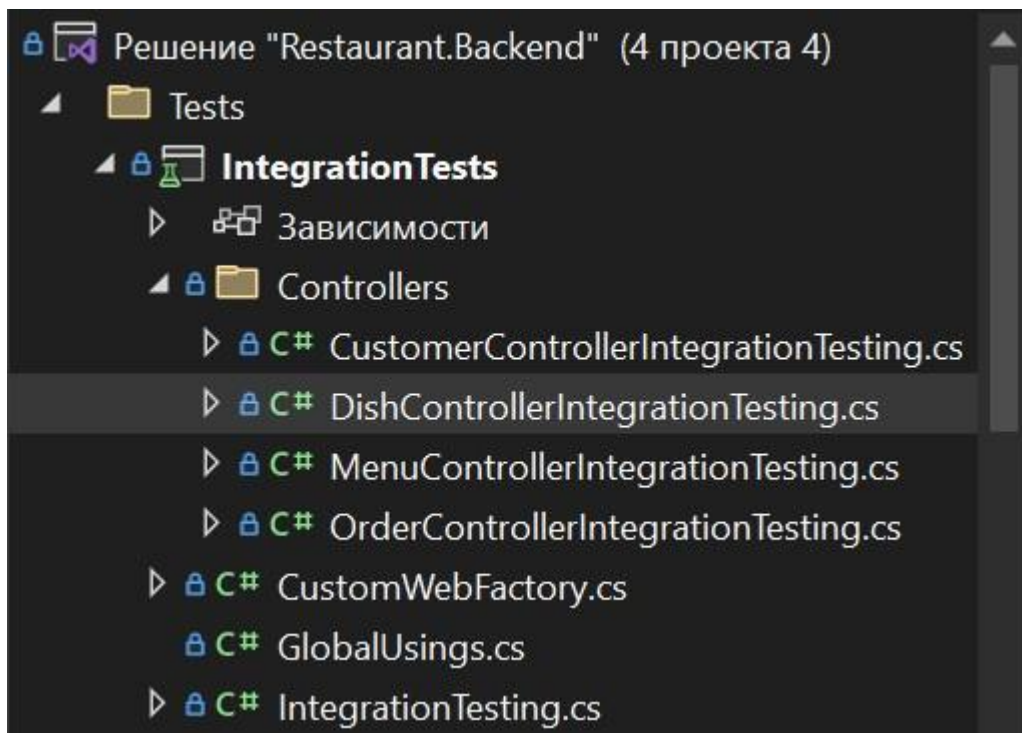
## ER-діаграма



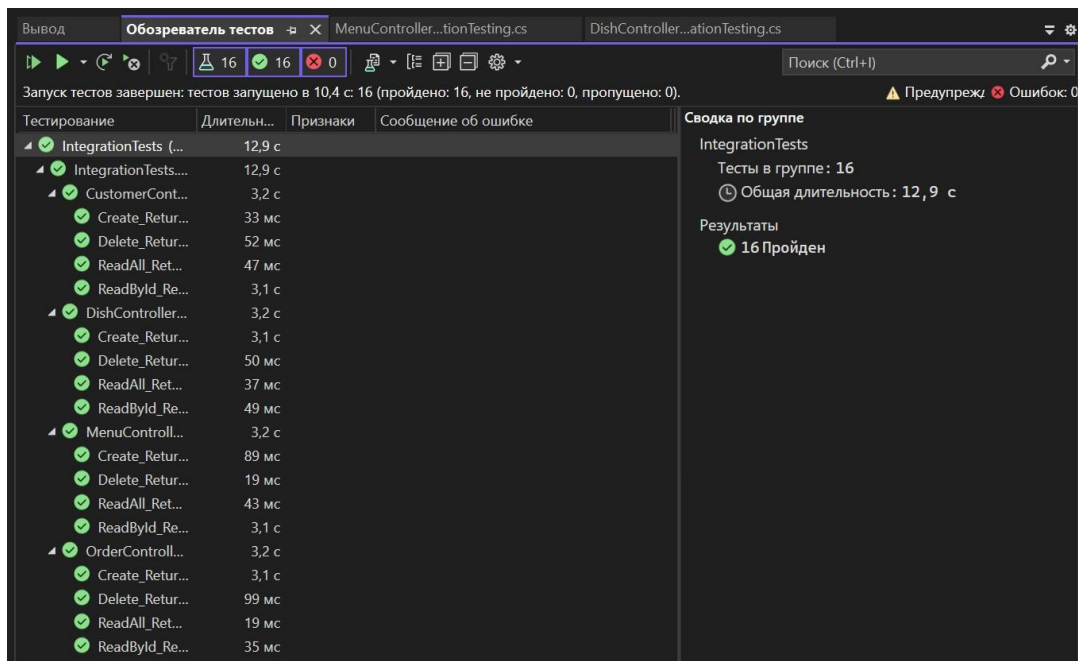
## Зв'язки між сутностями

- Customer-Order – one-to-many;
- Order-OrderItem – one-to-many;
- OrderItem-PriceListNote – one-to-one;
- PriceListNote-Dish – many-to-one;
- Dish-Ingredient – many-to-many;

Також імплементовано інтеграційне тестування:



Результати виконання тестів:



## Кінцеві точки (Endpoints)

### Customer

**GET** /api/customer

**POST** /api/customer

**PUT** /api/customer

**GET** /api/customer/{id}

**DELETE** /api/customer/{id}

### Dish

**GET** /api/dish

**POST** /api/dish

**PUT** /api/dish

**GET** /api/dish/{id}

**DELETE** /api/dish/{id}

**GET** /api/dish/ingredients

**POST** /api/dish/ingredients

**DELETE** /api/dish/ingredient/{ingredientId}

## Menu

**GET** /api/menu

**POST** /api/menu

**GET** /api/menu/{id}

**DELETE** /api/menu/{id}

## Order

**GET** /api/order

**POST** /api/order

**PUT** /api/order

**GET** /api/order/{id}

**DELETE** /api/order/{id}

**GET** /api/order/order-items

**POST** /api/order/order-items

**DELETE** /api/order/order-item/{orderId}