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

ЗВІТ

ДО ЛАБОРАТОРНОЇ РОБОТИ №3

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 №3

Проектування REST веб-API

Варіант №1

Завдання:

Теоретична частина:

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.

Практична частина:

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

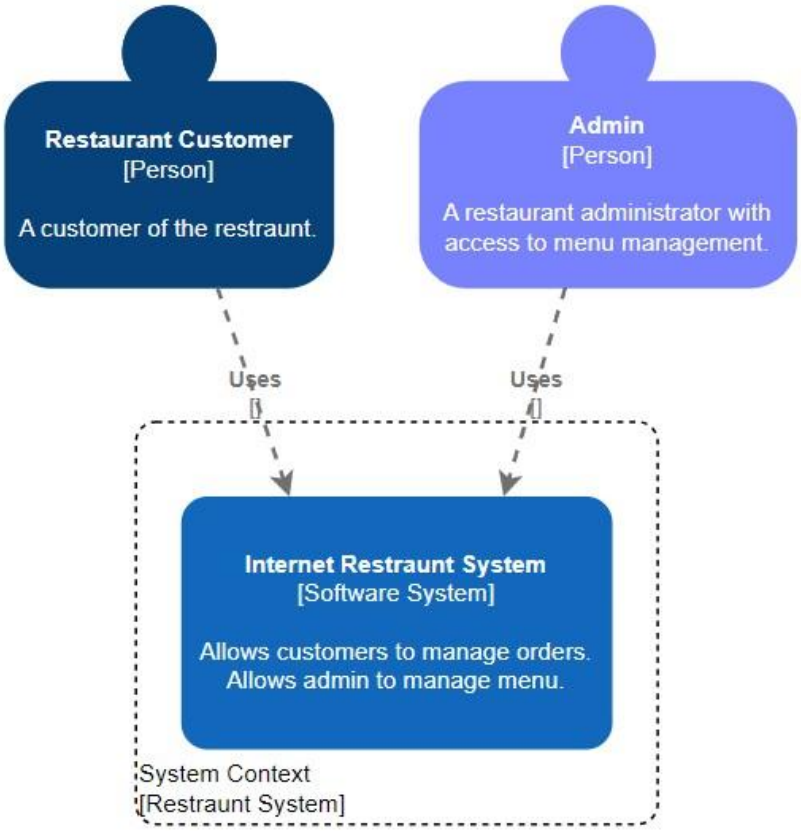
Хід роботи

C4 model (Context, Containers, Components, Code) - це методологія для моделювання архітектури системи.

Дана методологія включає 4 основні рівні архітектури:

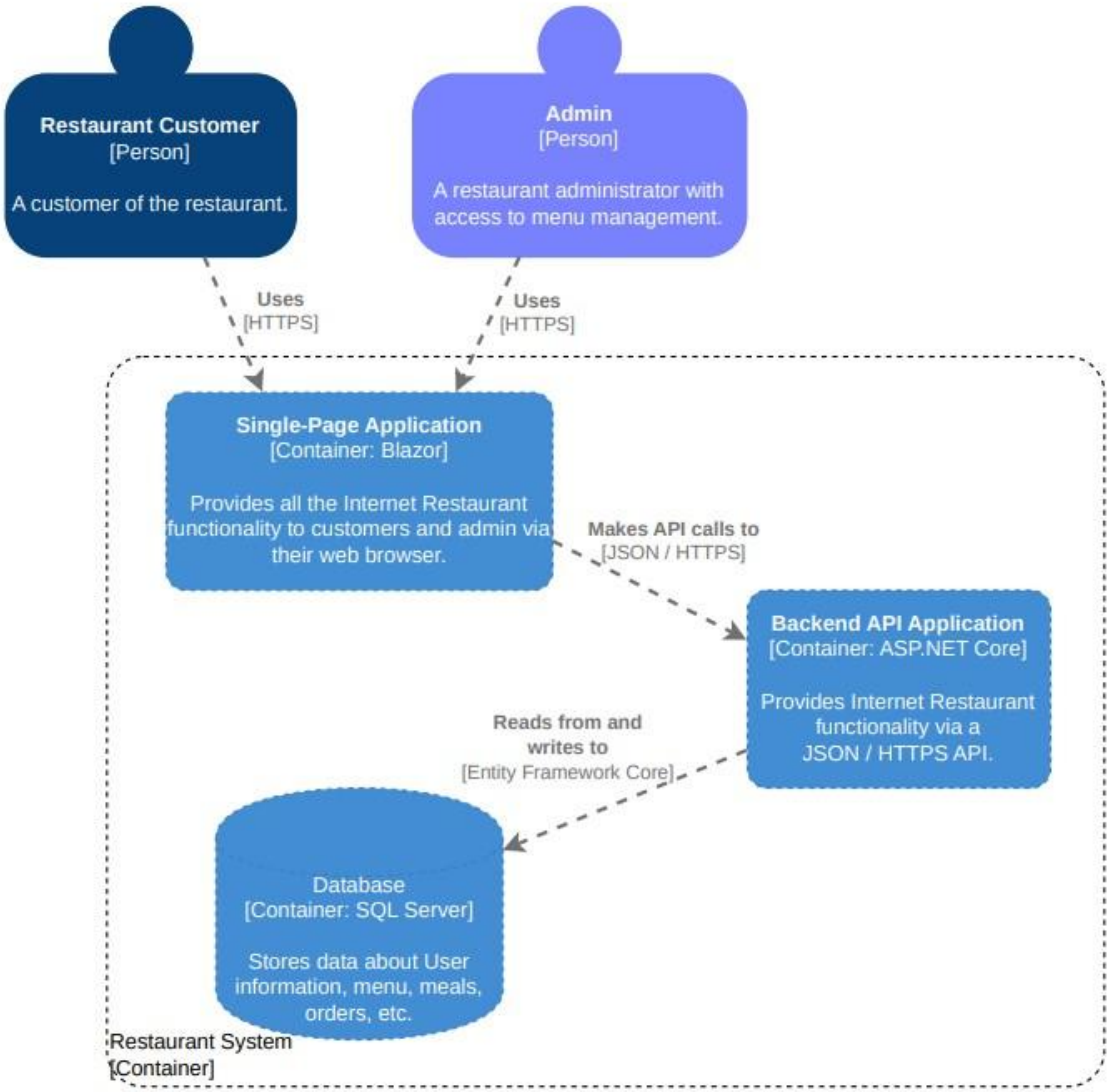
1) Контекстна діаграма (Context): Зображує систему як чорний ящик та її взаємодію з іншими системами.
2) Діаграма контейнерів (Containers): Розкриває внутрішню структуру системи та взаємодію контейнерів.
3) Діаграма компонентів (Components): Подробиці про компоненти в межах контейнерів та їх взаємодію.
4) Діаграма коду (Code): Деталізує компоненти та їх взаємодію на рівні коду.

Level 1 (Context)



Для взаємодії із системою застосовуються дві ролі – клієнт системи та адміністратор ресторану. Клієнт може взаємодіяти із веб-застосунком таким чином: переглядати меню, формувати замовлення. Адміністратору, у свою чергу, доступний той же контент, що і для клієнта, але окрім того в нього є доступ до редагування меню та страв.

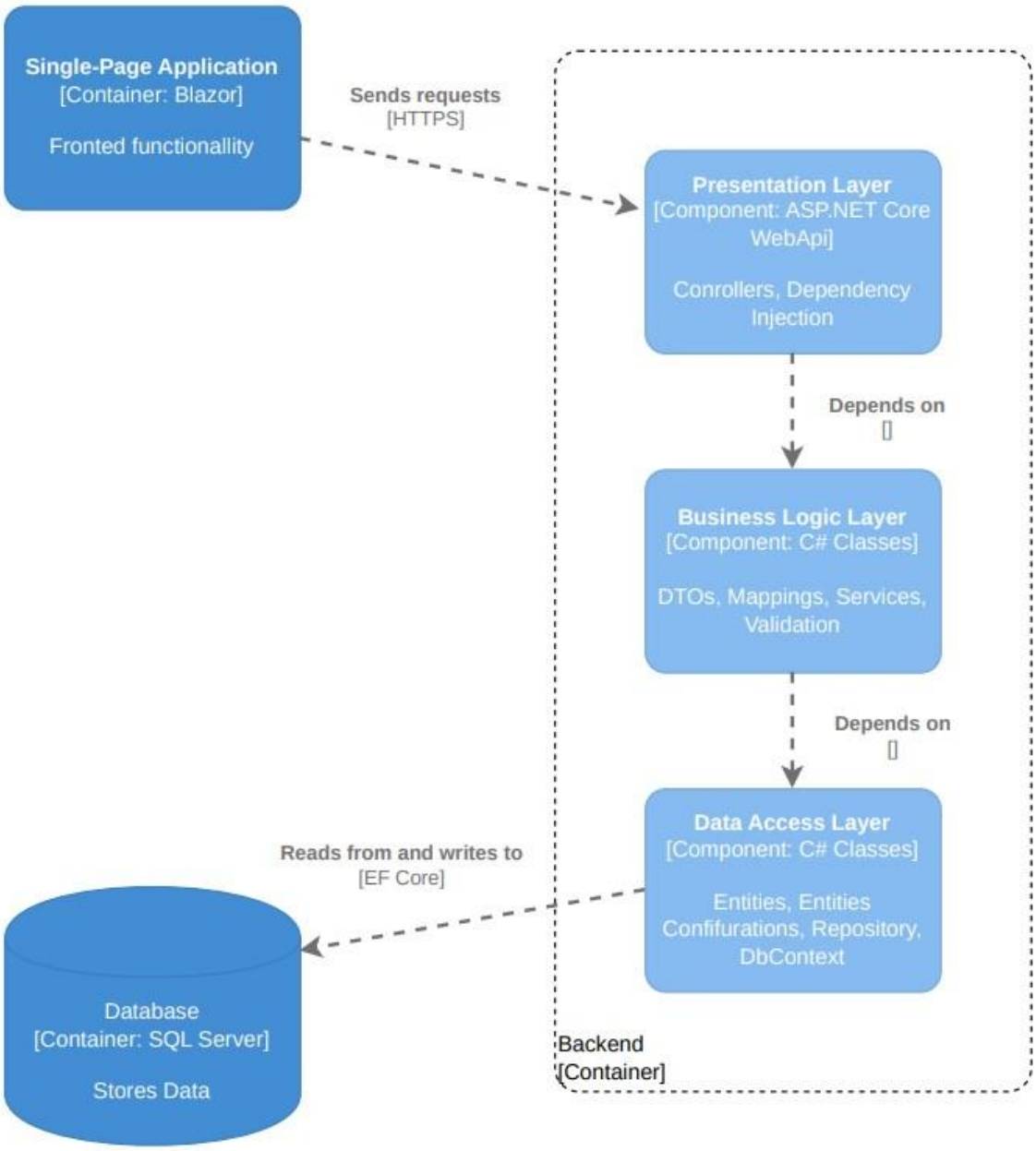
Level 2 (Container)



На рівні контейнерів можна виділити три основні складові системи:

- Односторінковий веб-застосунок, з яким відбувається взаємодія на стороні браузера.
- Backend реалізація у вигляді серверного застосунку REST API.
- База даних, що містить у собі дані системи.

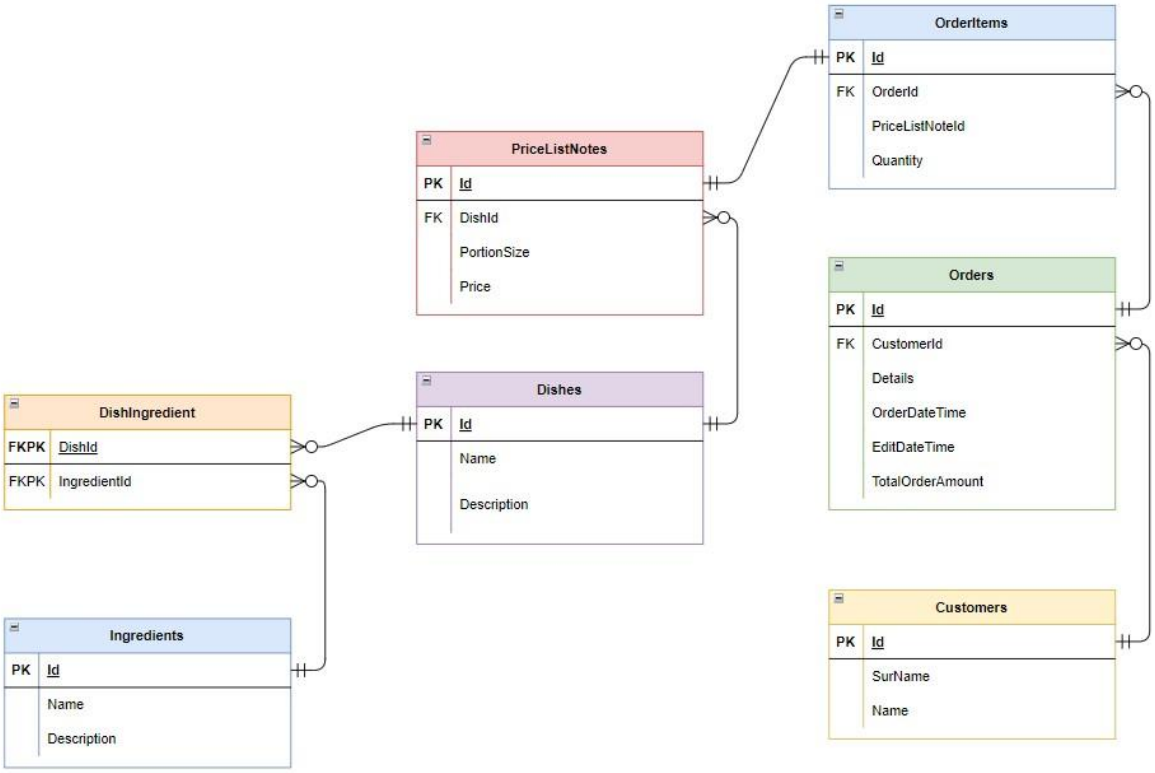
Level 3 (Components)



В свою чергу серверний Backend за стосунок можна розділити на три складові, що реалізують три рівні N-Layer архітектури:

- Presentation Layer – цей шар реалізує взаємодію із HTTP запитами;
- Business Logic Layer – шар, у якому розташована бізнес-логіка, в свою чергу, тут відбувається обробка даних та взаємодія з DAL.
- Data Access Layer – містить у собі контекст бази даних, репозиторії, опис сутностей, за допомогою EF Core реалізує взаємодію з БД.

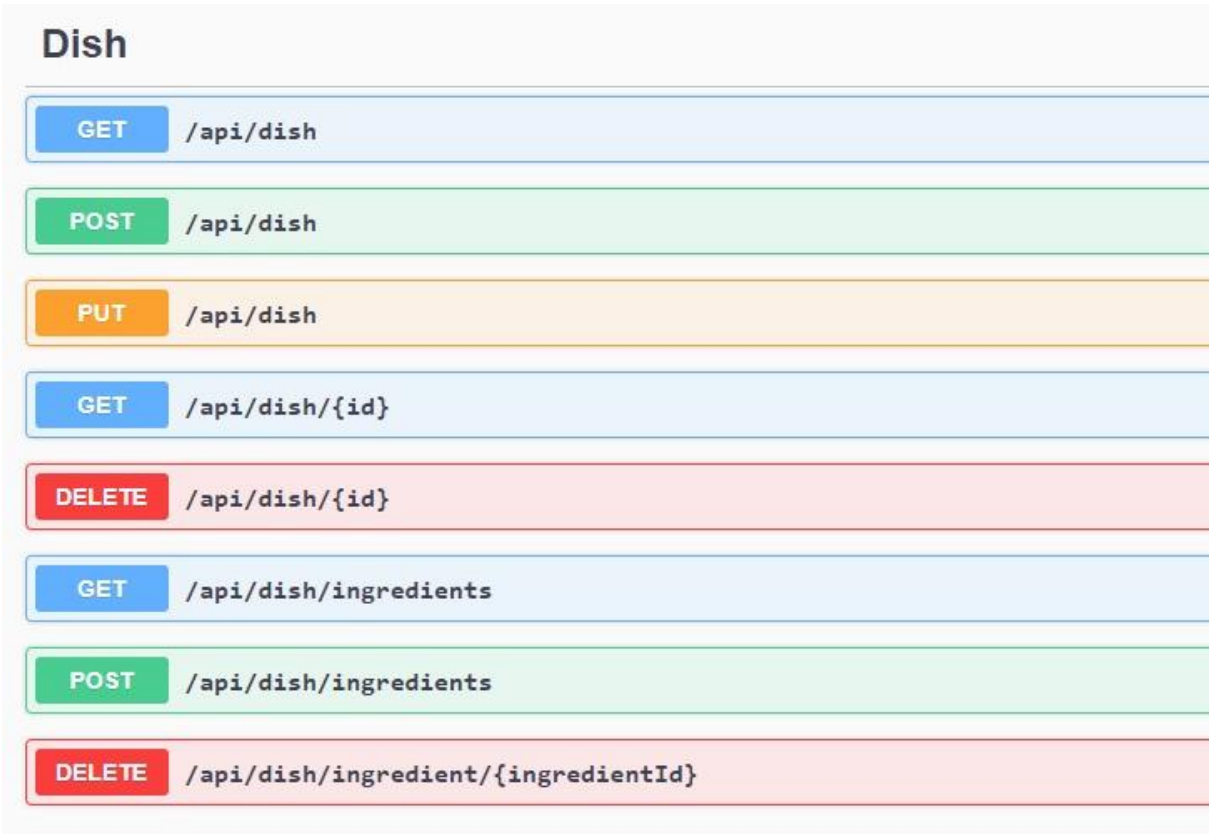
ER-діаграма



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;

Кінцеві точки (Endpoints)



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/{orderItemId}