**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний**

**інститут імені Ігоря Сікорського" Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

# Звіт

з лабораторної роботи № 3 з дисципліни

«Сучасні технології розробки WEB-застосувань на платформі Microsoft.NET» “**Проектування REST веб-API**”

**Виконав(ла)**

*ІC-11 Поліщук В.В.*

(шифр, прізвище, ім'я, по батькові)

**Перевірив**

*Бардін В.*

(прізвище, ім'я, по батькові)

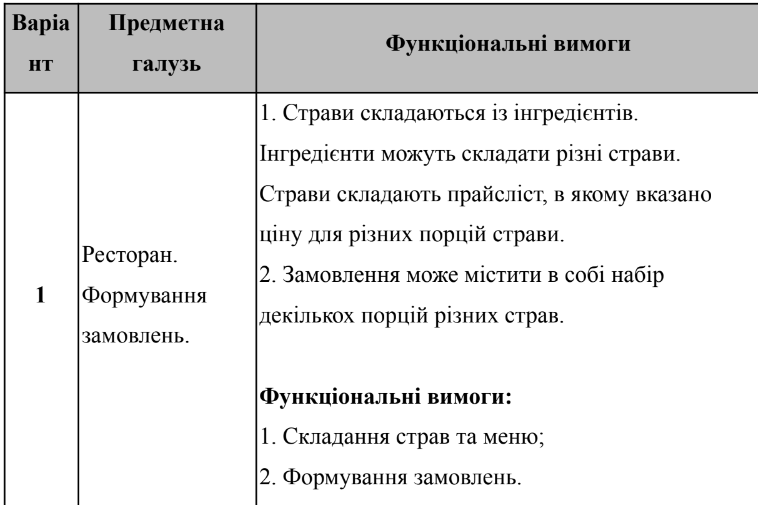
Київ 2023

# Лабораторна робота 3

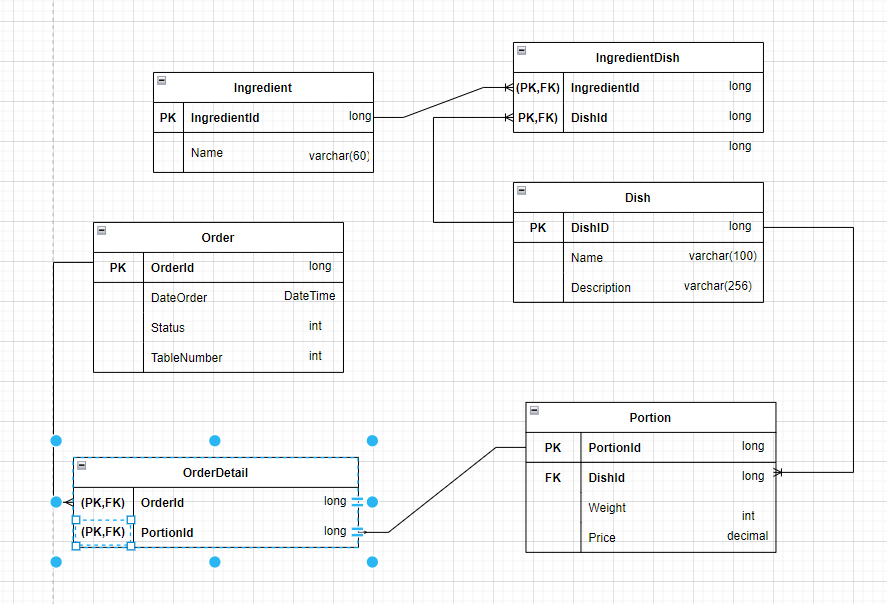
Проектування REST веб-API Завдання

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.

Доменна область:



ER-діаграма



Таблиця **Ingredient**:

Призначення – зберігає інфорамацію про інгредієнти

Поля:

* IngredientId : long – унікальний ідентифікатор інгредієнту
* Name : varchar(60) – назва інгредієнту

Таблиця **Dish**:

Призначення – зберігає інформацію про страву

Поля:

* DishId – long – унікальний ідентифікатор страви
* Name – varchar(100) – назва страви
* Description – varchar(256) – опис страви

Допоміжна таблиця **IngredientDish**

Призначення: для зв’язку між таблицями Ingredient та Dish

Поля :

* IngredientId : long – унікальний ідентифікатор інгредієнту
* DishId – long – унікальний ідентифікатор страви

Таблиця **Portion**

Призначення – зберігає інформацію про порції та їх ціни

Поля:

* PortionId : long - унікальний ідентифікатор порції
* DishId : long – унікальний ідентифікатор страви
* Weight : int – вага порції
* Price : decimal – ціна порції

Таблиця **Order**

Призначення – зберігає інформацію про замовлення

Поля:

* OrderId : long - унікальний ідентифікатор замовлення
* DateOrder : DateTime – час створення замовлення
* Status : int – статус замовлення
* TableNumber : int – номер столика , яке замовляє

Таблиця **OrderDetail**

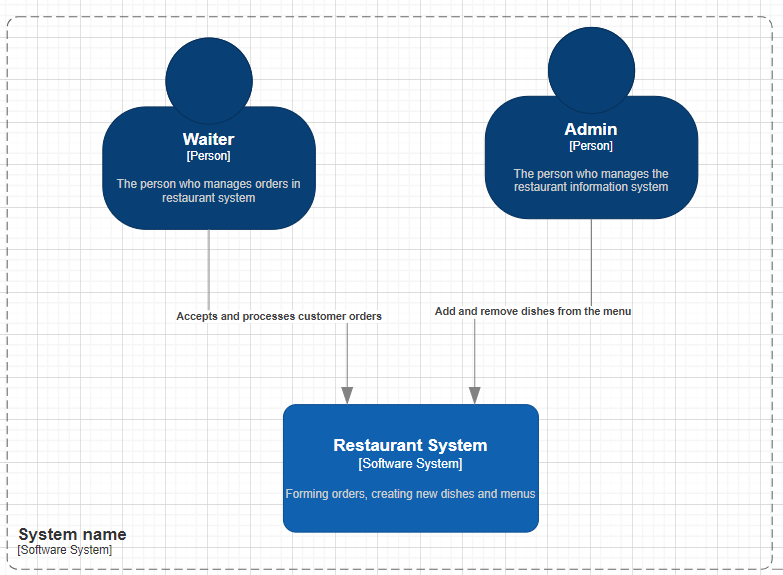
Призначення – допоміжна таблиця між Order та Portion

Поля :

* OrderId : long - унікальний ідентифікатор замовлення
* PortionId : long – унікальний ідентифікатор порції

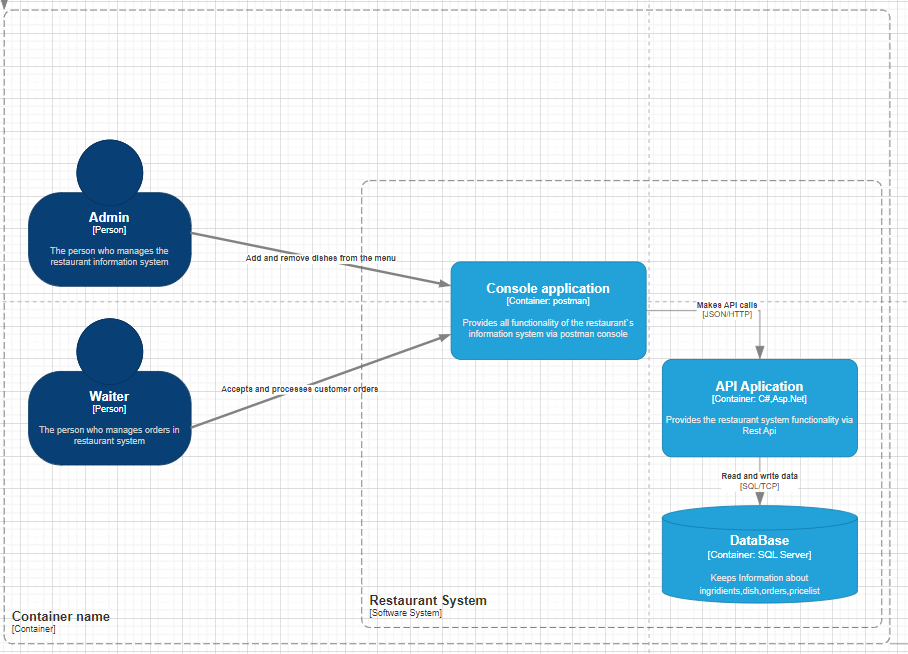
C4-діаграма

System Context Diagram



У інформаційної системи ресторану будуть 2 типи користувачів: Адмін – це людина, яка керує інформаційною системою ресторану, а саме складає страви , меню . Офіціант – це людина, яка приймає та обробляє замовлення.

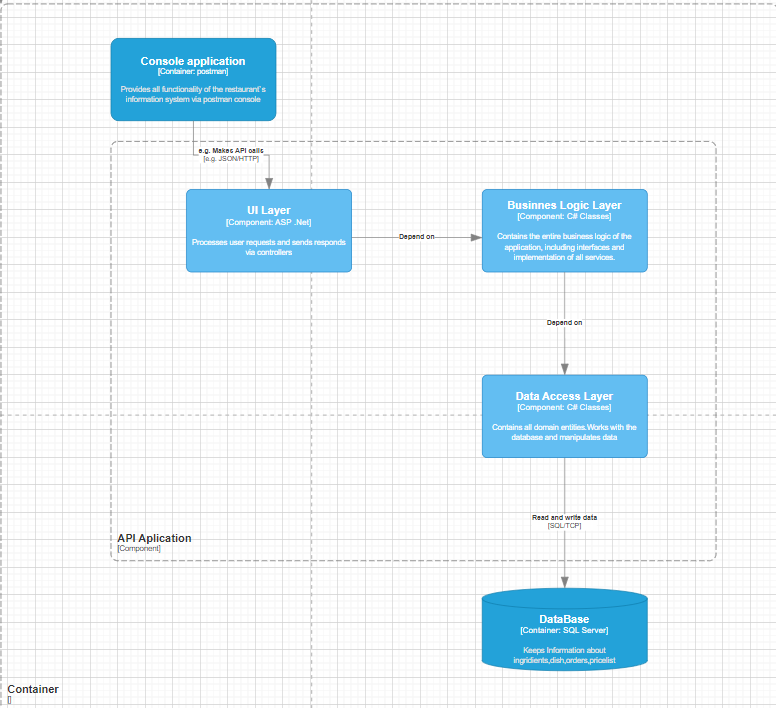
Container Diagram



Інформаційна система ресторану розділяється на три компоненти.

* Консольний застосунок, в нашому випадку Postman , для взаємодії Web-API
* Серверний застосунок , який реалізує функціонал через REST API
* База даних, яка зберігає всю інформацію про замовлення користувача , страви та їх інгредієнти , ціни на різні порції страв

Component Diagram



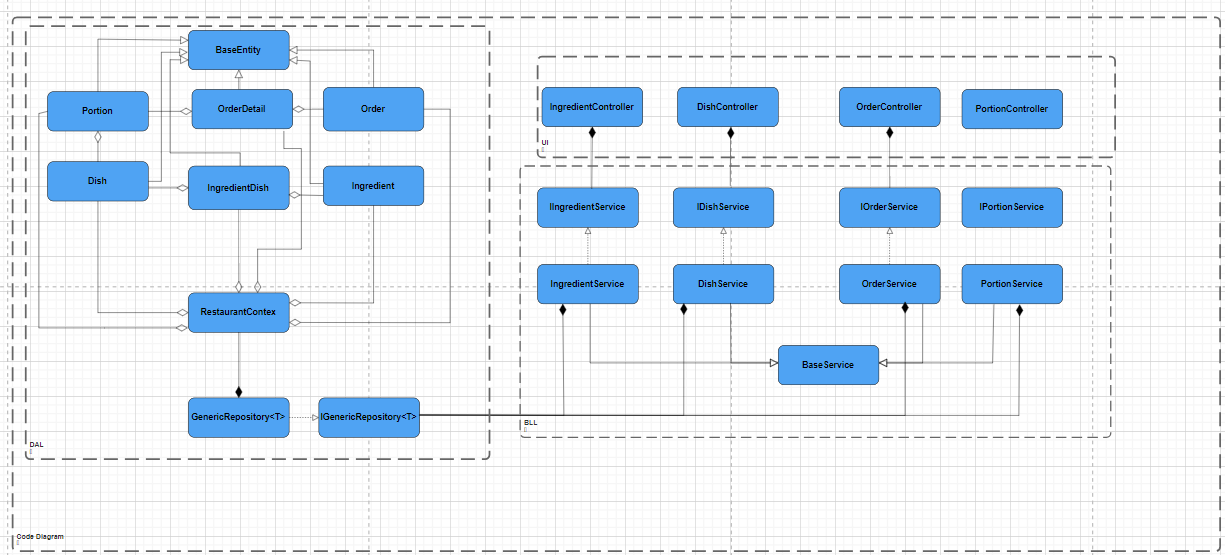
Для розробки програми .Net буде використовуватися типова трирівнева архітектура, яка розділить програму на такі рівні:

1. Рівень доступу до даних (DAL) - забезпечує простий доступ до даних, що зберігаються в SQL Server. Містить усі сутності домену та контекст бази даних.

2. Рівень бізнес-логіки (BLL) – містить всю бізнес-логіку програми. Залежить від DAL.

3. Рівень інтерфейсу користувача (UI Layer) – рівень, відповідальний за обробку запитів користувачів. Містить усі контролери. Залежить від BLL.

Code Diagram



1)**Ingredient**

-GET : api/ingredient – перегляд доступних інгрідієнтів

-POST : api/ingridient – створення інгрідієнту

-DELETE: api/ingridient/{id} – видалення інгрідієнтує

2)**Dish**

-GET- api/dish – перегляд доступних страв

-GET- api/dish/{id} – детальний перегляд обраної страви

-POST – api/dish – створення нової страви

-DELETE- api/dish/{id}- видалити страву

3)**Order**

- GET – api/order - перегляд замовлень

- GET – api/order/{Id}- перегляд конкретного замовлення

- POST – api/order- створення нового замовлення

- PATCH- api/order/{Id} – оновити статус замовлення

-DELETE- api/order/{id}- видалити замовлення

4)**Portion**

-PATCH-api/portion/{id}-зміна ціни за порцію