

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-застосунків на платформі Microsoft.NET»

“Проектування REST веб-API”

Варіант 9

Виконав студент ІП-12 Єльчанінов Артем Юрійович

Перевірів(ла) Бардін Владислав

Київ 2023

**Мета лабораторної роботи** – ознайомитися з основами створення REST веб-API та методологією С4 для відображення архітектури системи. Ознайомитися з основами створення ER-діаграм для представлення структури бази даних.

## Завдання

### 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.

|   |                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9 | Реєстратура лікарні. Запис до лікарів на прийом | <p>1. Реєстратура надає дані стосовно наявності лікарів та розкладу прийому хворих.</p> <p>2. Хворим можливо записатись на прийом до лікаря, якщо є вільний час у розкладі лікаря.</p> <p>3. В реєстратурі ведуться картки відвідування хворими лікарні, в які записується час відвідання лікаря, діагноз та лікар, що його поставив.</p> <p><b>Функціональні вимоги:</b></p> <p>1. Ведення картотеки лікарні;</p> <p>2. Керування прийомами хворих у лікарів.</p> |
|---|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

# Хід роботи

## Діаграма C4

### Рівень 1:

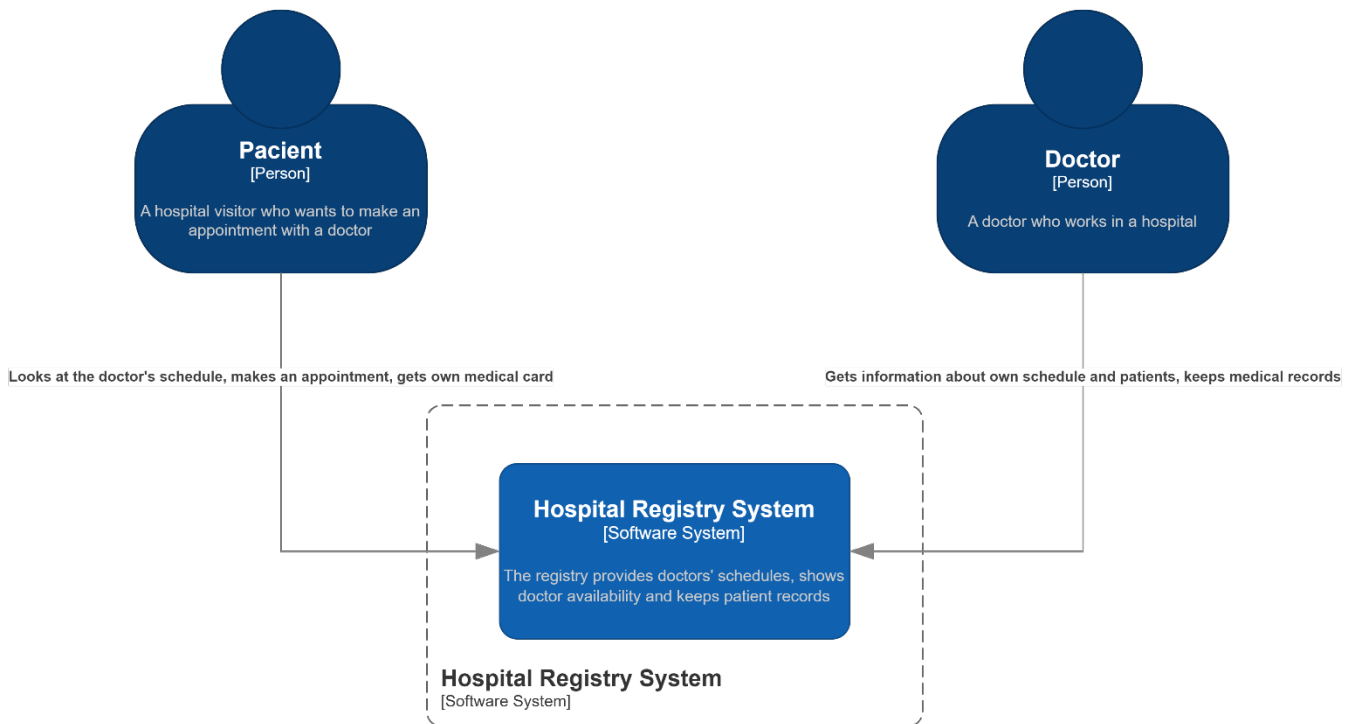


Рисунок 1 – System Context Diagram

У системі реєстратури лікарні є два користувача: пацієнт та лікар. Пацієнт може переглядати розклад лікарів на заданий час, яких би він хотів відвідати та має можливість запису до лікаря на прийом, можливість отримати власну медичну картку якщо така є. Лікар може отримувати інформацію з реєстратури про свій розклад на заданий час та отримувати інформацію про пацієнта, який до нього записаний, вести лікарські висновки. Сама система зберігає ці всі дані, та веде картки пацієнтів.

## Рівень 2:

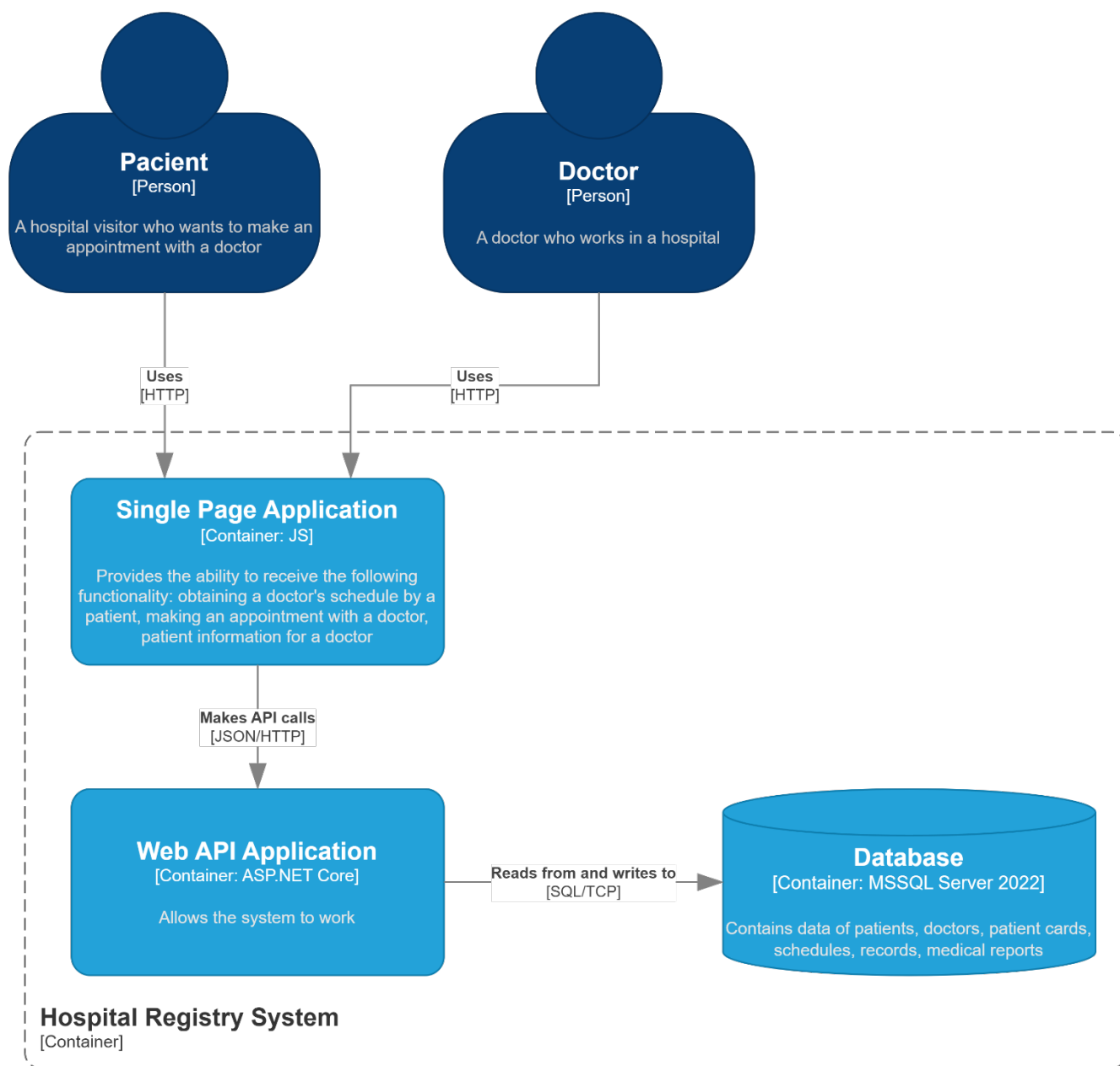


Рисунок 2 – Container Diagram

Система поділяється на такі компоненти:

1. Веб-сторінка, що надає можливість користувачу взаємодіяти з системою.
2. Серверний застосунок, що виконує запити користувачів.
3. База даних, яка зберігає всю потрібну інформацію.

### Рівень 3:

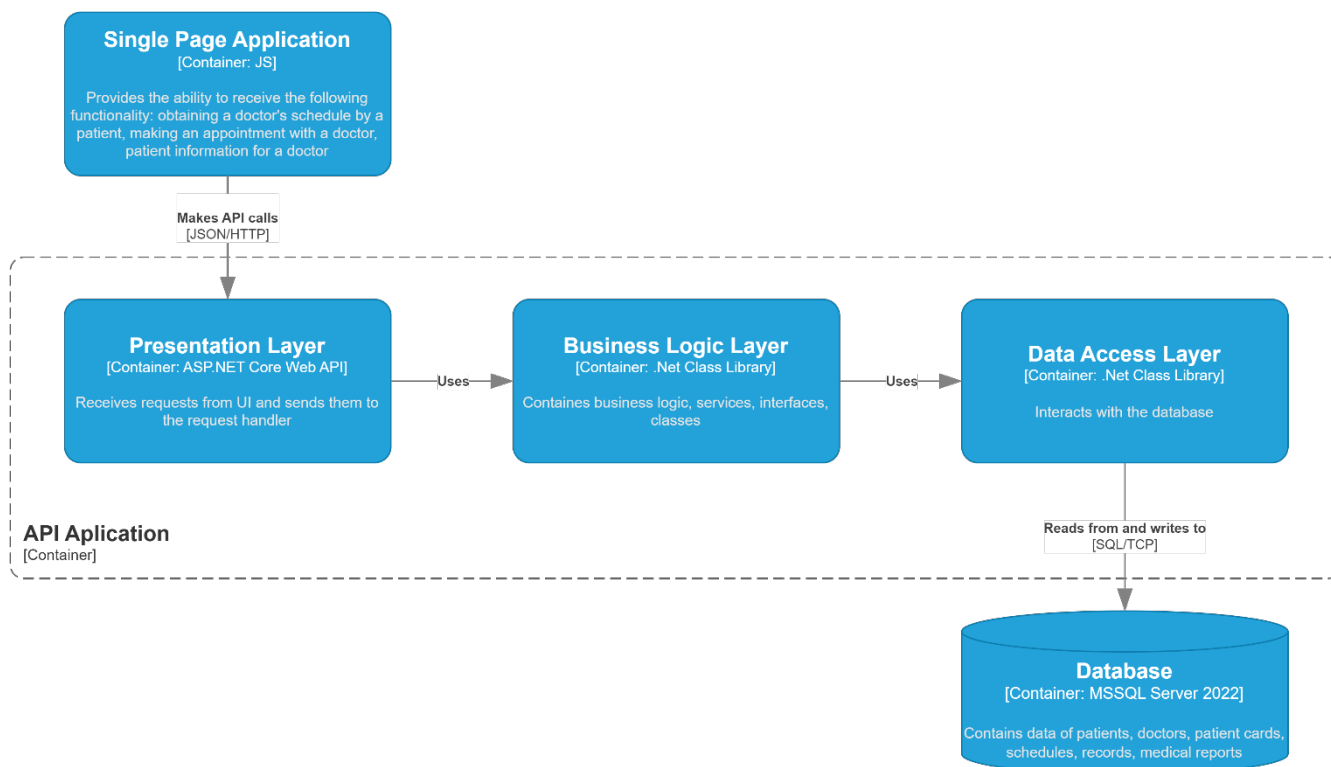
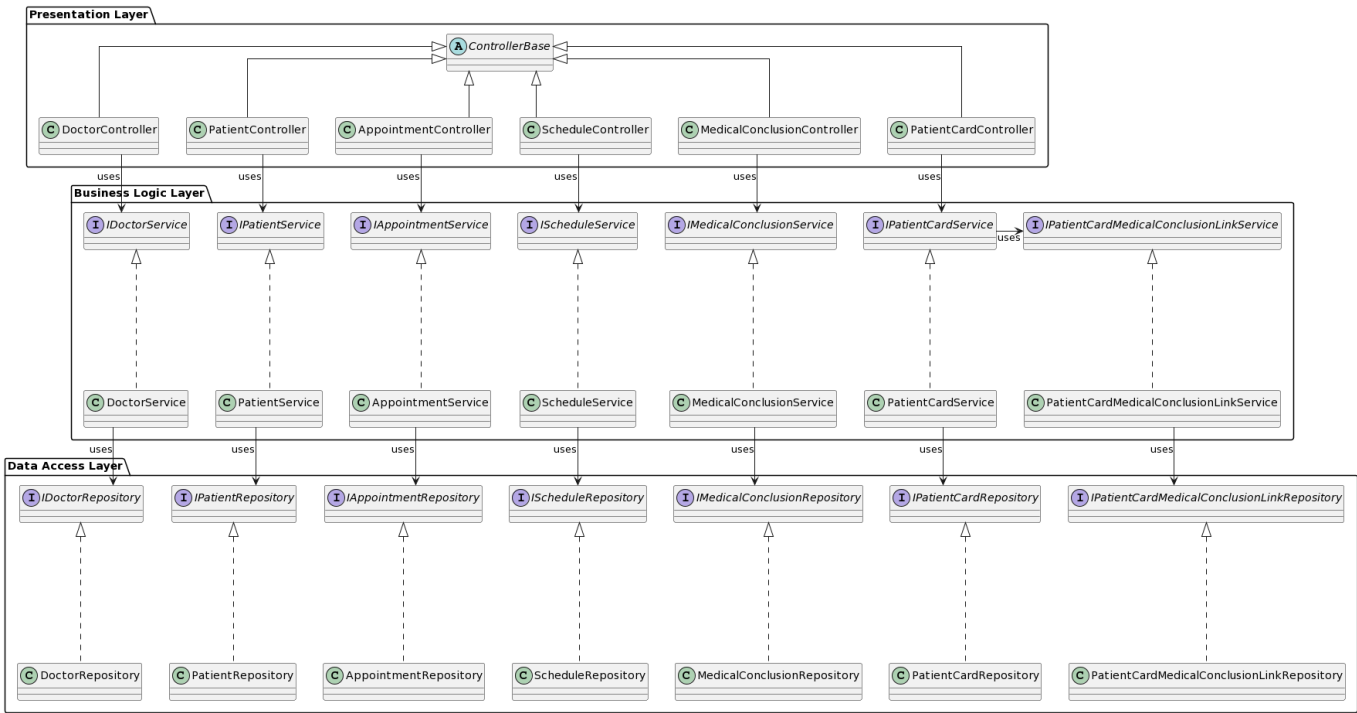


Рисунок 3 – Components Diagram

При розробці архітектури застосунку була застосована багатошарова архітектура, яка містить наступні шари:

1. Presentation Layer – відповідає за взаємодію з користувачем та представлення інформації, містить контролери.
2. Business Logic Layer – визначає логіку операцій та обробку даних, містить інтерфейси сервісів та їх імплементацію.
3. Data Access Layer – відповідає за взаємодію з базою даних та містить контекст бази даних.

Рівень 4:



## ER-діаграма

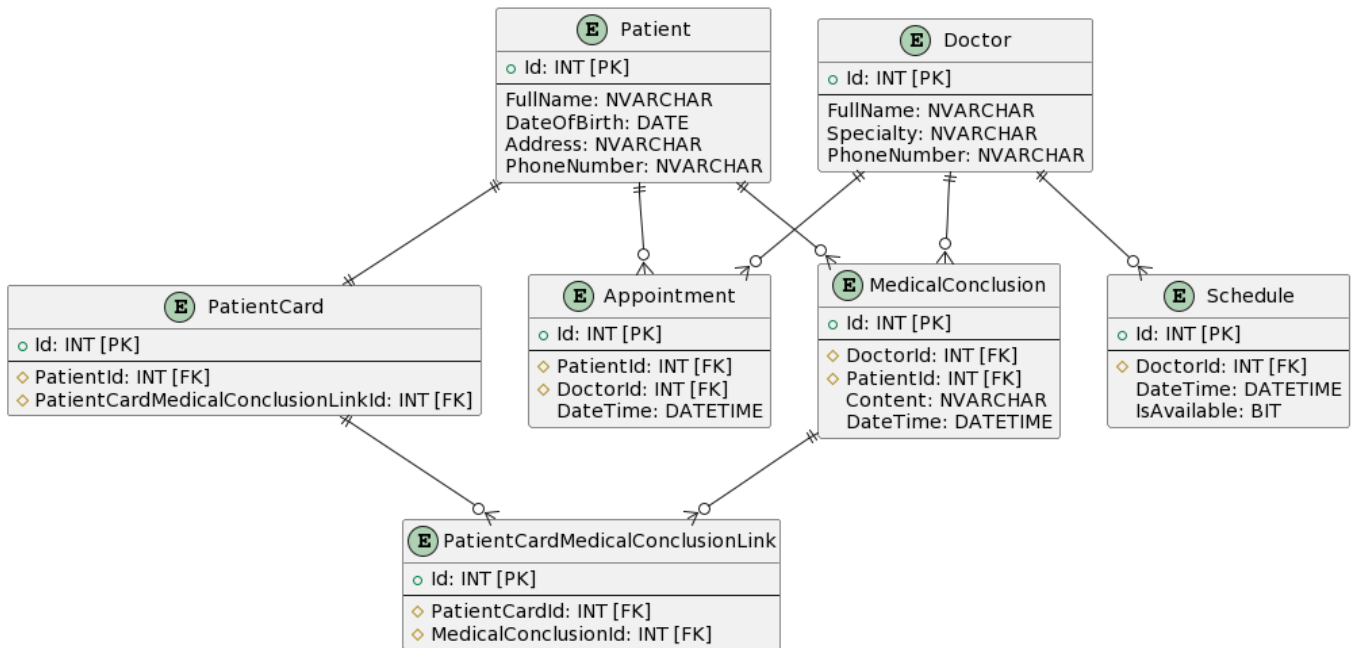


Рисунок 5 – ER-діаграма бази даних

### Ендпоінти:

#### 1. DoctorController:

- POST /doctors - Створення нового лікаря.
- GET /doctors - Отримання списку всіх лікарів.
- GET /doctors/{id} - Отримання детальної інформації про лікаря за ID.
- PUT /doctors/{id} - Оновлення інформації про лікаря за ID.
- DELETE /doctors/{id} - Видалення лікаря за ID.

#### 2. PatientController:

- POST /patients - Створення нового пацієнта.
- GET /patients - Отримання списку всіх пацієнтів.
- GET /patients/{id} - Отримання детальної інформації про пацієнта за ID.
- PUT /patients/{id} - Оновлення інформації про пацієнта за ID.
- DELETE /patients/{id} - Видалення пацієнта за ID.

#### 3. AppointmentController:

- POST /appointments - Створення нового запису на прийом.

- GET /appointments - Отримання списку всіх записів.
- GET /appointments/{id} - Отримання детальної інформації про запис за ID.
- PUT /appointments/{id} - Оновлення запису на прийом за ID.
- DELETE /appointments/{id} - Видалення запису за ID.

#### 4. ScheduleController:

- POST /schedules - Створення нового розкладу.
- GET /schedules - Отримання списку всіх розкладів.
- GET /schedules/{id} - Отримання детальної інформації про розклад за ID.
- PUT /schedules/{id} - Оновлення розкладу за ID.
- DELETE /schedules/{id} - Видалення розкладу за ID.

#### 5. MedicalConclusionController:

- POST /medicalconclusions - Створення нового лікарського висновку.
- GET /medicalconclusions - Отримання списку всіх висновків.
- GET /medicalconclusions/{id} - Отримання детальної інформації про висновок за ID.
- PUT /medicalconclusions/{id} - Оновлення висновку за ID.
- DELETE /medicalconclusions/{id} - Видалення висновку за ID.

#### 6. PatientCardController:

- POST /patientcards - Створення нової медичної картки.
- GET /patientcards - Отримання списку всіх медичних карток.
- GET /patientcards/{id} - Отримання детальної інформації про медичну картку за ID.
- PUT /patientcards/{id} - Оновлення медичної картки за ID.
- DELETE /patientcards/{id} - Видалення медичної картки за ID.