

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)

ІП-15 Костін В.А.

(шифр, прізвище, ім'я, по батькові)

Перевірів

Бардін В.

(прізвище, ім'я, по батькові)

Київ 2023

Лабораторна робота 3

Проектування REST веб-API

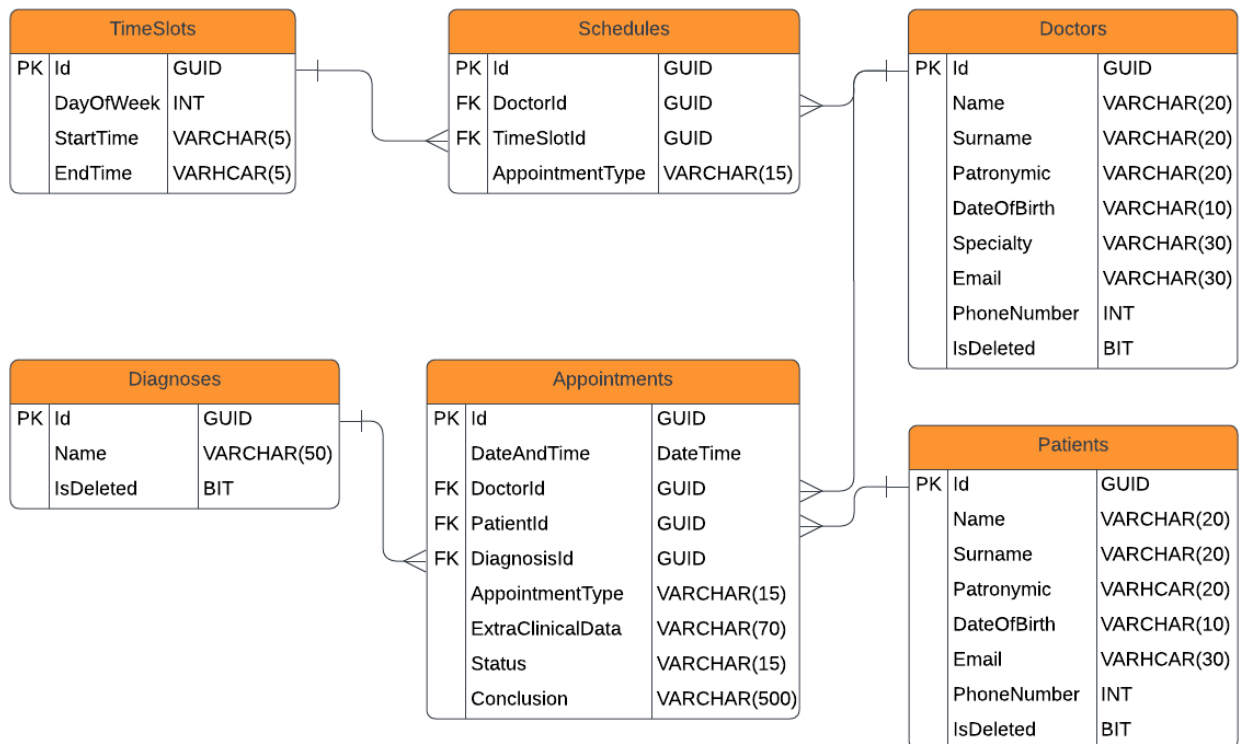
Завдання

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 спроектоване рішення у вигляді звіту до лабораторної роботи.

Доменна область:

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>Функціональні вимоги:</p> <p>1. Ведення картотеки лікарні;</p> <p>2. Керування прийомами хворих у лікарів.</p>
----------	---	--

ER-діаграма



Посилання:

https://lucid.app/lucidchart/9ed9785f-d2af-42ed-ba34-6cc8900082e7/edit?viewport_loc=-29%2C25%2C2219%2C1057%2C0_0&invitationId=inv_ef3455d0-bdeb-4247-8dd7-c8e9277388b5

Таблиця: TimeSlots

Призначення: Проміжок часу для складання робочого графіку доктора.

Поля:

- Id: GUID – унікальний ідентифікатор проміжку часу.
- DayOfWeek:INT – день тижня.
- StartTime:VARHCHAR(5) – початок проміжку часу.

- EndTime:VARCHAR(5) – кінець проміжку часу.

Таблиця: Schedules

Призначення: Часова одиниця робочого графіку доктора, опираючись на яку формуються записи.

Поля:

- Id: GUID – унікальний ідентифікатор часового графіку.
- DoctorId:GUID – унікальний ідентифікатор лікаря.
- TimeSlotId:GUID – унікальний ідентифікатор проміжку часу.
- AppointmentType:VARCHAR(15) – тип запису, який припадає на лікаря та встановленні години(також може бути перервою).

Таблиця: Doctors

Призначення: Облікова таблиця докторів лікарні.

Поля:

- Id: GUID – унікальний ідентифікатор доктору.
- Name:VARCHAR(20) – ім'я доктору.
- Surname:VARCHAR(20) – прізвище доктору.
- Patronymic:VARHCAR(20) – по-батькові.
- Specialty:VARCHAR(30) – спеціальність доктора(хірург, травматолог, тощо).
- Email:VARCHAR(30) – поштова адреса доктора.
- PhoneNumber:INT – номер телефону доктора.
- IsDeleted:BIT – прапорець програмного видалення.

Таблиця: Patients

Призначення: Облікова таблиця пацієнтів лікарні.

Поля:

- Id: GUID – унікальний ідентифікатор пацієнта.
- Name: VARCHAR(20) – ім'я пацієнта.
- Surname: VARCHAR(20) – прізвище пацієнта.
- Patronymic: VARCHAR(20) – по-батькові.
- Email: VARCHAR(30) – поштова адреса пацієнта.
- PhoneNumber: INT – номер телефону пацієнта.
- IsDeleted: BIT – прапорець програмного видалення.

Таблиця: Diagnoses

Призначення: Діагнози, через які пацієнти записуються до лікаря.

Поля:

- Id: GUID – унікальний ідентифікатор діагноза.
- Name: VARCHAR(50) – назва діагноза.
- IsDeleted: BIT – прапорець програмного видалення.

Таблиця: Appointments

Призначення: Прийом пацієнта до лікаря з усією клінічною інформацією.

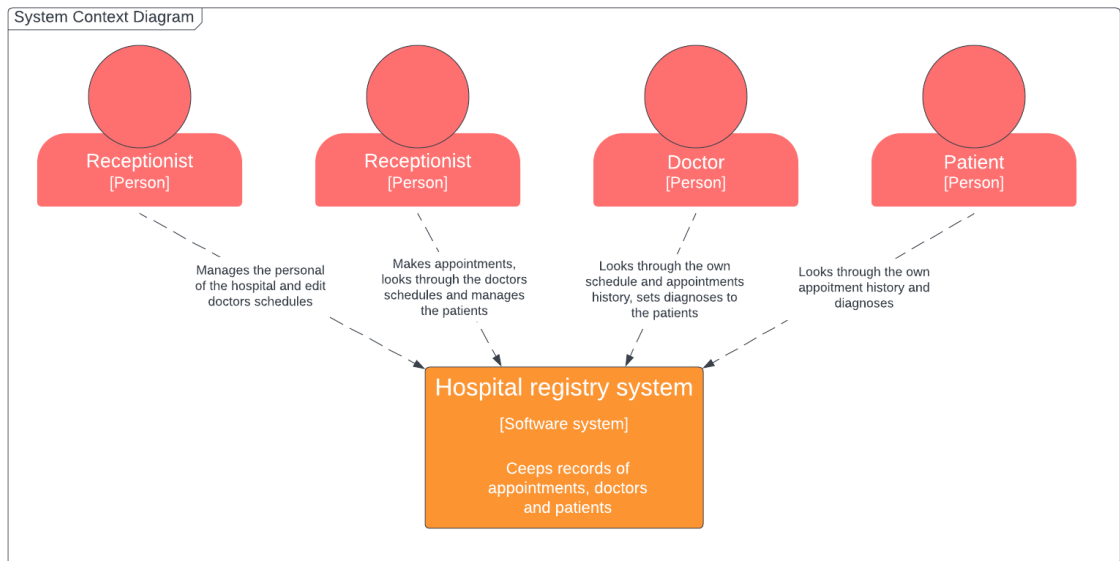
Поля:

- Id: GUID – унікальний ідентифікатор запису.
- DateAndTime: DateTime – дата та час запису.

- DoctorId:GUID - унікальний ідентифікатор доктора.
- PatientId:GUID - унікальний ідентифікатор пацієнта.
- DiagnosisId:GUID - унікальний ідентифікатор діагнозу.
- AppointmentType:VARCHAR(15) – тип запису, який припадає на лікаря та встановленні години(також може бути перервою).
- ExtraClinicalData:VARCHAR(70) – додаткові клінічні данні пацієнта перед прийомом.
- Status:VARCHAR(15) – статус прийому(“Scheduled”, “Active”, “Completed”, “Canceled”).
- Conclusion:VARCHAR(150) – висновок лікаря стосовно прийому.

C4-діаграма

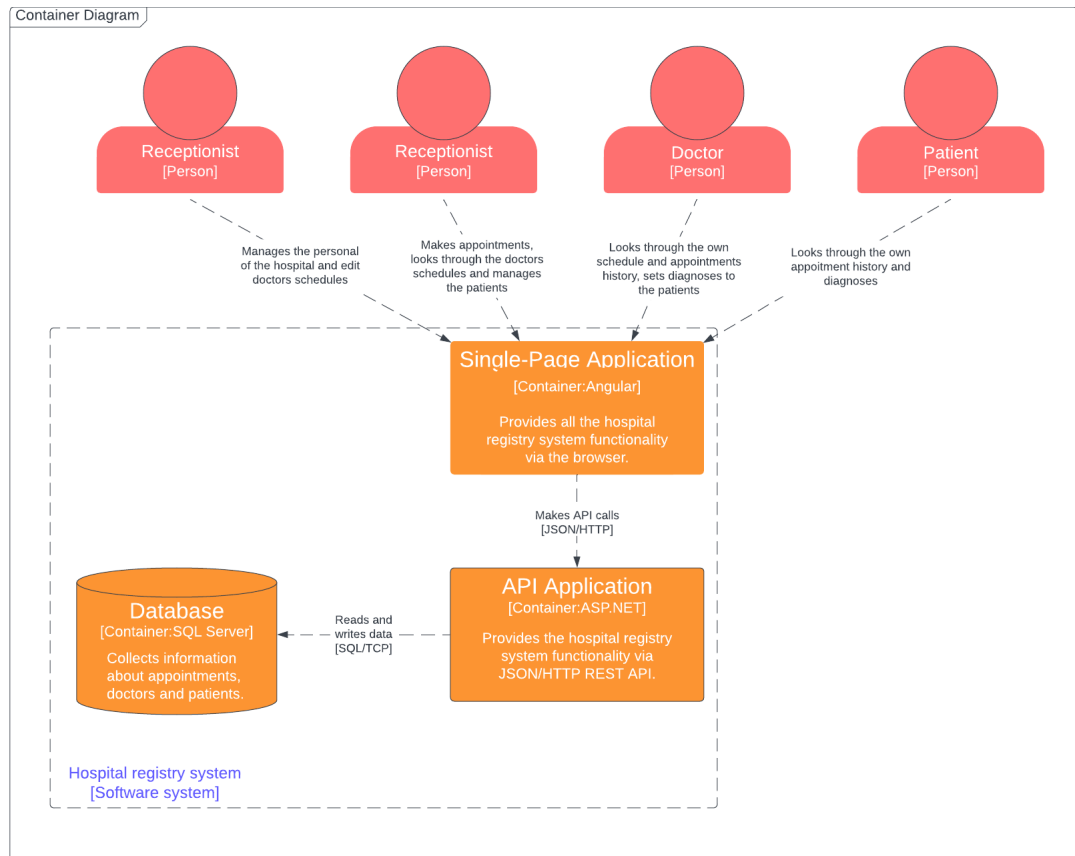
System Context Diagram



У системи “Реєстратура лікарні” будуть чотири типи користувачів:

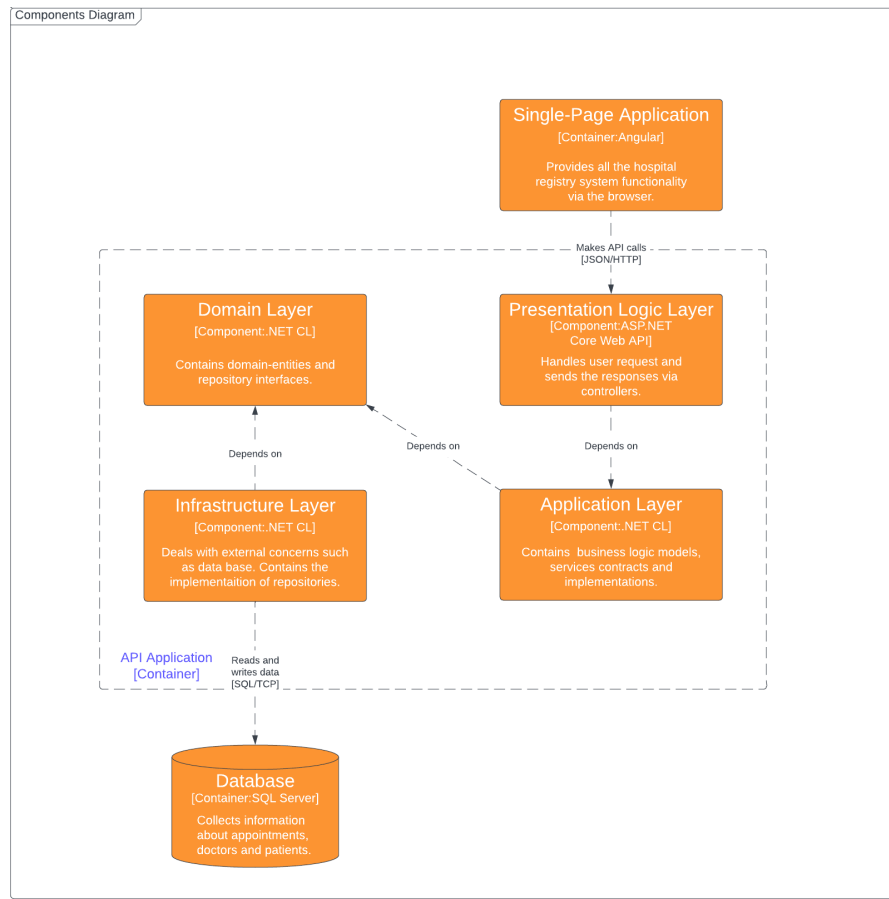
- Адмін – керує даними про персонал лікарні(докторів), також налаштовує додаткові данні, наприклад, діагнози;
- Реєстратор – записує пацієнтів на прийом до лікаря, може продивлюватись графік роботи лікаря, веде картотеку пацієнтів; Доктор – продивляється свій робочий графік, історію прийомів, ставити діагноз пацієнту та висновок прийому;
- Пацієнт – продивляється історію прийомів в лікарні.

Container Diagram



Система “Реєстратура лікарні” розділяється на три основні компоненти: веб-застосунок, який надає функціонал реєстратури через веб-браузер, серверний застосунок, який реалізує функціонал через API, та база даних – яка зберігає усю інформацію про лікарів, пацієнтів та прийоми.

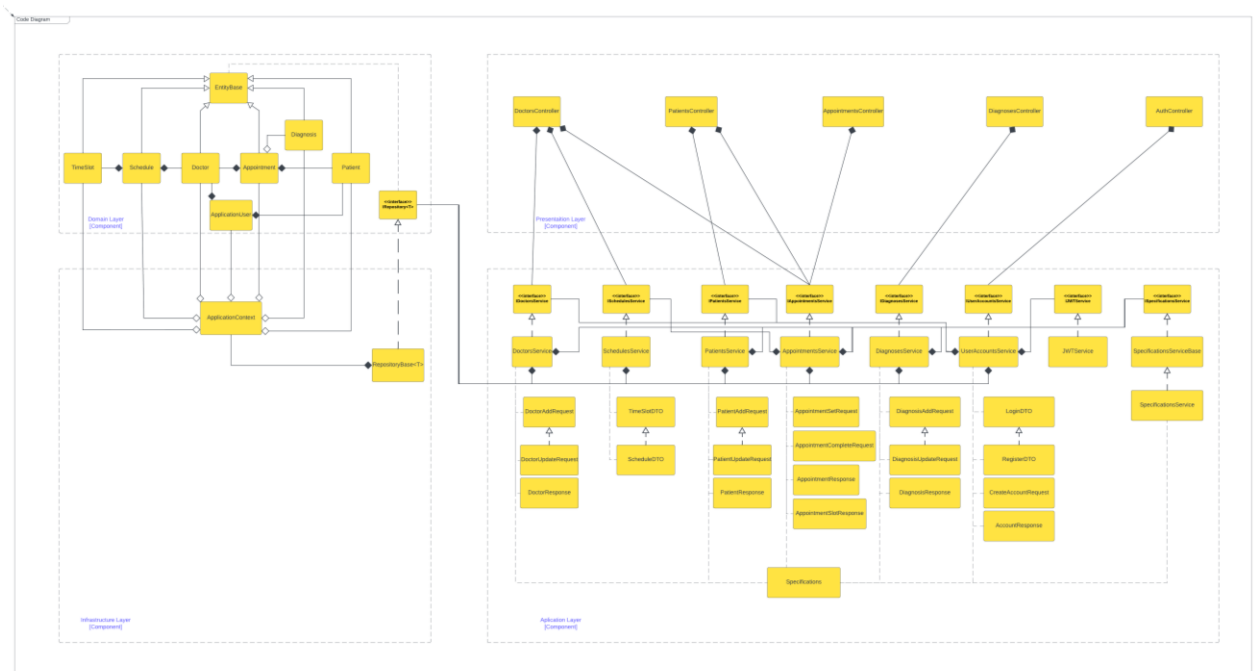
Component Diagram



Для розробки бекенд-застосунку буде застосовано архітектурний паттерн “Clean Architecture”, який розділяє застосунку на наступні шари:

- **Domain** – ядро застосунку, містить сутності доменної області та контракти репозиторіїв;
- **Infrastructure** – компонент, який комунікує зі сторонніми сервісами або джерелами даних(таких як база даних), містить контекст бази даних та реалізацію репозиторіїв;
- **Application** – шар бізнес логіки, містить контракти та реалізації сервісів, DTO класи та інші компоненти бізнес логіки;
- **Presentation** – шар, який відповідає за комунікацію з користувачами та обробку запитів, містить контролери.

Code Diagram



Всі шари застосунку будуть реалізовані в якості бібліотеки класів, окрім Presentation(це проект ASP.NET Core Web API).

Доменний шар буде реалізовано з сутностями бази даних, які успадковуються від абстрактного класу `EntityBase`. Це допоможе в дженеріковому інтерфейсі репозиторія прописати обмеження типу аргумента. Також шар містить клас `ApplicationUser`, який буде використовуватись для Identity.

Шар інфраструктури містить реалізацію дженерікового репозиторію `RepositoryBase` та контекст бази даних `ApplicationContext`, який в свою чергу містить датасети всіх сутностей з домену. Для комунікації з базою даних використовується `EntityFramework`.

Шар бізнес-логіки містить інтерфейси та реалізацію сервісів. Також в цьому шарі присутні DTO об'єкти.

Посилання:

https://lucid.app/lucidchart/fb9efa17-618d-41bf-a919-257b6424584c/edit?viewport_loc=-9191%2C-4159%2C32079%2C15288%2C0_0&invitationId=inv_6675198b-ced4-4c9b-b2ad-a63e059a9bd2

Нижче надано опис кінцевих точок API з URL, параметрами, методом та призначенням:

Accounts:

URL: /api/v1/accounts

Метод: GET

Параметри: специфікації фільтрування, сортування, пошуку та пагінації (query string)

Призначення: список активних користувачів системи

URL: /api/v1/accounts

Метод: POST

Параметри: об'єкт з інформацією про нового користувача(ім'я, роль, email, пароль) (request body)

Призначення: створення нового користувача тільки для ролі "Admin" або "Receptionist"

URL: /api/v1/accounts/login

Метод: POST

Параметри: об'єкт з інформацією про користувача для входу в систему(email, пароль) (request body)

Призначення: аутентифікація для існуючого користувача, повертає токен авторизації

URL: /api/v1/accounts/register

Метод: POST

Параметри: об'єкт з інформацією про користувача для реєстрації(ім'я, роль, email, пароль, ключ для реєстрації)
(request body)

Призначення: реєстрація нового користувача тільки для ролі
“Doctor” або “Patient”, повертає токен авторизації

URL: /api/v1/accounts/logout

Метод: GET

Призначення: вихід користувача із системи

Appointments

URL: /api/v1/appointments

Метод: GET

Параметри: специфікації фільтрування, сортування, пошуку та
пагінації (query string)

Призначення: список даних про записи пацієнтів до лікаря

URL: /api/v1/appointments/{id}

Метод: GET

Параметри: ідентифікатор запису (path)

Призначення: інформація про запис пацієнта до лікаря за
вказаним ідентифікатором

URL: /api/v1/appointments

Метод: POST

Параметри: об'єкт з інформацією про день та час запису, лікаря, пацієнта та додаткові клінічні данні (request body)

Призначення: створення нового запису пацієнта до лікаря.

URL: /api/v1/appointments

Метод: PUT

Параметри: об'єкт з ідентифікатором модифікованого запису, інформацією про діагноз та висновок лікаря (request body)

Призначення: завершення існуючого запису

URL: /api/v1/appointments/free-slots

Метод: GET

Параметри: специфікації фільтрування та пошуку вільних слотів для запису (query string)

Призначення: список вільних слотів для запису пацієнта до лікаря

URL: /api/v1/appointments/{id}

Метод: DELETE

Параметри: ідентифікатор запису (path)

Призначення: скасування запису пацієнта до лікаря

URL: /api/v1/appointments/{id}/recover

Метод: POST

Параметри: ідентифікатор запису (path)

Призначення: відновлення скасованого запису пацієнта до лікаря

URL: /api/v1/appointments/canceled

Метод: DELETE

Призначення: видалення всіх скасованих записів пацієнта до лікаря

Diagnoses

URL: /api/v1/diagnoses

Метод: GET

Параметри: специфікації фільтрування, сортування, пошуку та пагінації (query string)

Призначення: список діагнозів, які лікуються в лікарні

URL: /api/v1/diagnoses/{id}

Метод: GET

Параметри: ідентифікатор діагнозу (path)

Призначення: інформація про діагноз вказаним ідентифікатором

URL: /api/v1/diagnoses

Метод: POST

Параметри: об'єкт з назвою діагнозу (request body)

Призначення: створення нового діагнозу

URL: /api/v1/diagnoses

Метод: PUT

Параметри: об'єкт з ідентифікатором та новою назвою діагнозу (request body)

Призначення: оновлення назви існуючого діагнозу

URL: /api/v1/diagnoses

Метод: PUT

Параметри: об'єкт з ідентифікатором та новою назвою діагнозу (request body)

Призначення: оновлення назви існуючого діагнозу

URL: /api/v1/diagnoses/{id}

Метод: DELETE

Параметри: ідентифікатор діагнозу (path)

Призначення: програмне видалення існуючого діагнозу

URL: /api/v1/diagnoses/{id}/recover

Метод: POST

Параметри: ідентифікатор діагнозу (path)

Призначення: відновлення видаленого діагнозу

Doctors

URL: /api/v1/doctors

Метод: GET

Параметри: специфікації фільтрування, сортування, пошуку та пагінації (query string)

Призначення: список докторів лікарні

URL: /api/v1/doctors/{id}

Метод: GET

Параметри: ідентифікатор доктору (path)

Призначення: інформація про доктора за вказаним ідентифікатором

URL: /api/v1/doctors

Метод: POST

Параметри: об'єкт з інформацією про доктора(ім'я, дата народження, email, номер телефону, спеціальність) (request body)

Призначення: створення нового запису про доктора

URL: /api/v1/doctors

Метод: PUT

Параметри: об'єкт з інформацією про доктора(ім'я, дата народження, email, номер телефону, спеціальність) (request body)

Призначення: оновлення інформації про існуючого доктора

URL: /api/v1/doctors/{id}

Метод: DELETE

Параметри: ідентифікатор доктору (path)

Призначення: програмне видалення існуючого доктору (для звільнення)

URL: /api/v1/doctors/{id}/recover

Метод: POST

Параметри: ідентифікатор доктору (path)

Призначення: відновлення видаленого доктору

URL: /api/v1/doctors/{id}/schedule

Метод: GET

Параметри: ідентифікатор доктору (path)

Призначення: інформація про робочій графік доктору

URL: /api/v1/doctors/schedule

Метод: POST

Параметри: об'єкт з ідентифікатором доктору та інформацією про новий робочий графік (request body)

Призначення: встановлення нового робочого графіку для доктору

URL: /api/v1/doctors/{id}/appointments/history

Метод: GET

Параметри: ідентифікатор доктору (path)

Призначення: історія записів до вказаного лікаря

URL: /api/v1/doctors/{id}/appointments/scheduled

Метод: GET

Параметри: ідентифікатор доктору (path) та дата (query string)

Призначення: список запланованих записів до вказаного лікаря на вказану дату

Patients

URL: /api/v1/patients

Метод: GET

Параметри: специфікації фільтрування, сортування, пошуку та пагінації (query string)

Призначення: список пацієнтів лікарні

URL: /api/v1/patients/{id}

Метод: GET

Параметри: ідентифікатор доктору (path)

Призначення: інформація про пацієнта за вказаним ідентифікатором

URL: /api/v1/patients

Метод: POST

Параметри: об'єкт з інформацією про пацієнта (ім'я, дата народження, email, номер телефону) (request body)

Призначення: створення нового запису про пацієнта

URL: /api/v1/patients

Метод: PUT

Параметри: об'єкт з інформацією про пацієнта (ім'я, дата народження, email, номер телефону) (request body)

Призначення: оновлення інформації про існуючого пацієнта

URL: /api/v1/patients/{id}

Метод: DELETE

Параметри: ідентифікатор пацієнта (path)

Призначення: програмне видалення існуючого пацієнта

URL: /api/v1/patients/{id}/recover

Метод: POST

Параметри: ідентифікатор пацієнта (path)

Призначення: відновлення видаленого пацієнта

URL: /api/v1/patients/{id}/appointments/history

Метод: GET

Параметри: ідентифікатор пацієнта (path)

Призначення: історія записів вказаного пацієнта

URL: /api/v1/patients/{id}/appointments/scheduled

Метод: GET

Параметри: ідентифікатор пацієнта (path) та дата (query string)

Призначення: список запланованих записів вказаного пацієнта на вказану дату