

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 № 4 з дисципліни
«Сучасні технології розробки WEB-застосувань на платформі Microsoft.NET»

“Імплементація REST 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

Імплементація REST API

Завдання

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.

Доменна область:

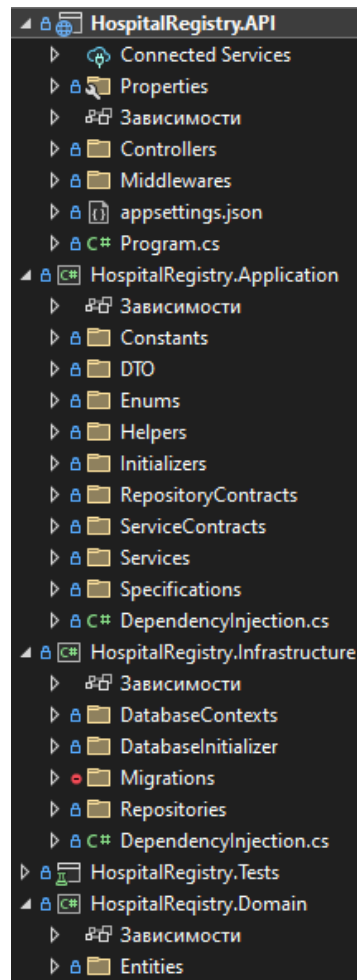
9	Реєстратура лікарні. Запис до лікарів на прийом	<p>1. Реєстратура надає дані стосовно наявності лікарів та розкладу прийому хворих.</p> <p>2. Хворим можливо записатись на прийом до лікаря, якщо є вільний час у розкладі лікаря.</p> <p>3. В реєстратурі ведуться картки відвідування хворими лікарні, в 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>
---	---	--

Код проекту можна побачити за посиланням:

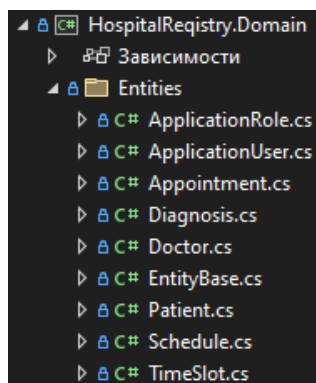
<https://github.com/VadimKostinProg/HospitalRegistryWebApp>

1. Структура проекту

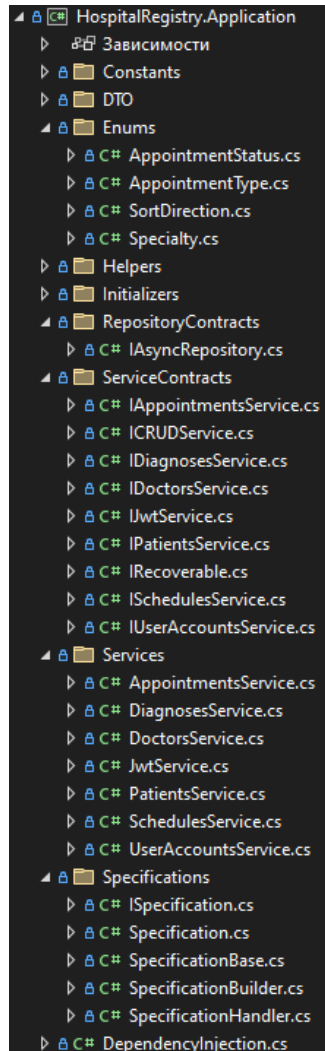
Для реалізації N-Layered архітектури було використано паттерн Clean Architecture. Структура застосунку має наступний вигляд:



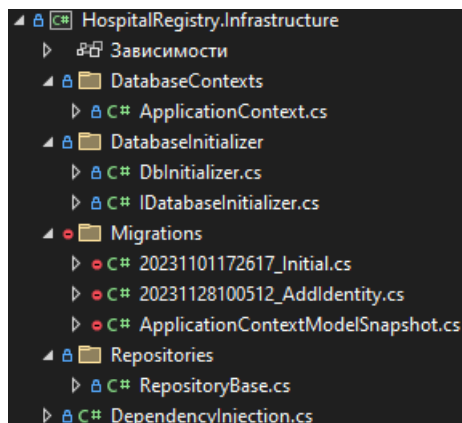
Domain Layer: містить імплементацію сутностей предметної області.



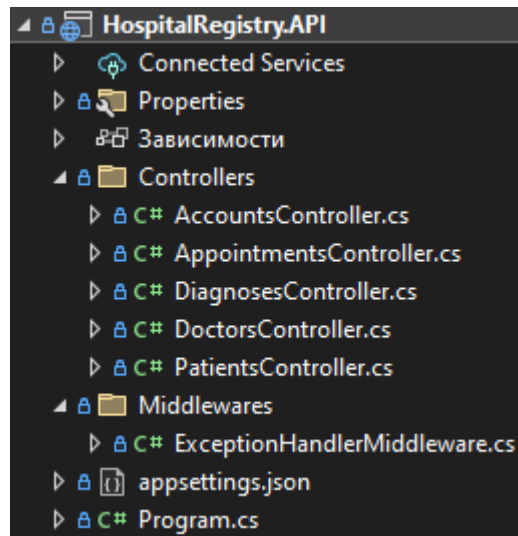
Application Layer: містить імплементацію інтерфейсу репозиторія, інтерфейси та імплементації сервісів, DTO, реалізація Specification Pattern.



Infrastructure Layer: містить імплементацію репозиторію, контекст EF Core бази даних, ініціалізатор та міграції до БД.



Presentation Layer: містить імплементацію контролерів, middlewares та загальну конфігурацію проекту.



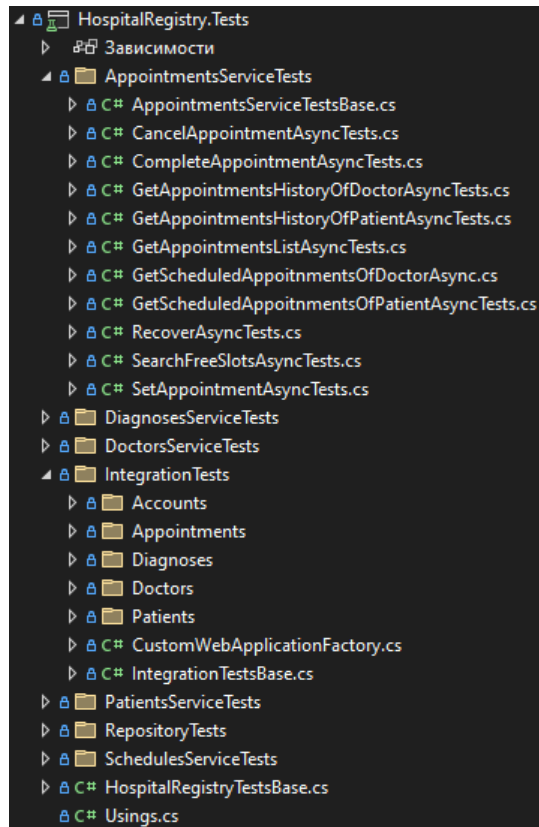
Далі наведено документацію API у вигляді Swagger:

Accounts		^
GET	/api/v1/Accounts	▼
POST	/api/v1/Accounts	▼
POST	/api/v1/Accounts/login	▼
POST	/api/v1/Accounts/register	▼
GET	/api/v1/Accounts/logout	▼
DELETE	/api/v1/Accounts/{id}	▼
Appointments		^
GET	/api/v1/Appointments	▼
POST	/api/v1/Appointments	▼
PUT	/api/v1/Appointments	▼
GET	/api/v1/Appointments/{id}	▼
DELETE	/api/v1/Appointments/{id}	▼
GET	/api/v1/Appointments/free-slots	▼
POST	/api/v1/Appointments/{id}/recover	▼
DELETE	/api/v1/Appointments/canceled	▼

Diagnoses		^
GET	/api/v1/Diagnoses	▼
POST	/api/v1/Diagnoses	▼
PUT	/api/v1/Diagnoses	▼
GET	/api/v1/Diagnoses/{id}	▼
DELETE	/api/v1/Diagnoses/{id}	▼
GET	/api/v1/Diagnoses/deleted	▼
POST	/api/v1/Diagnoses/{id}/recover	▼
Doctors		^
GET	/api/v1/Doctors	▼
POST	/api/v1/Doctors	▼
PUT	/api/v1/Doctors	▼
GET	/api/v1/Doctors/{id}	▼
DELETE	/api/v1/Doctors/{id}	▼
GET	/api/v1/Doctors/deleted	▼
POST	/api/v1/Doctors/{id}/recover	▼
GET	/api/v1/Doctors/{id}/schedule	▼
POST	/api/v1/Doctors/schedule	▼
GET	/api/v1/Doctors/{id}/appointments/history	▼
GET	/api/v1/Doctors/{id}/appointments/scheduled	▼
Patients		^
GET	/api/v1/Patients	▼
POST	/api/v1/Patients	▼
PUT	/api/v1/Patients	▼
GET	/api/v1/Patients/{id}	▼
DELETE	/api/v1/Patients/{id}	▼
GET	/api/v1/Patients/deleted	▼
POST	/api/v1/Patients/{id}/recover	▼
GET	/api/v1/Patients/{id}/appointments/history	▼
GET	/api/v1/Patients/{id}/appointments/scheduled	▼

2. Тестування

В проєкті присутні як модульні, так і інтеграційні тести. Для модульних тестів кожного методу створюється окремий клас. Так само і для інтеграційних методів кожного ендпоінту.



Результат тестування:

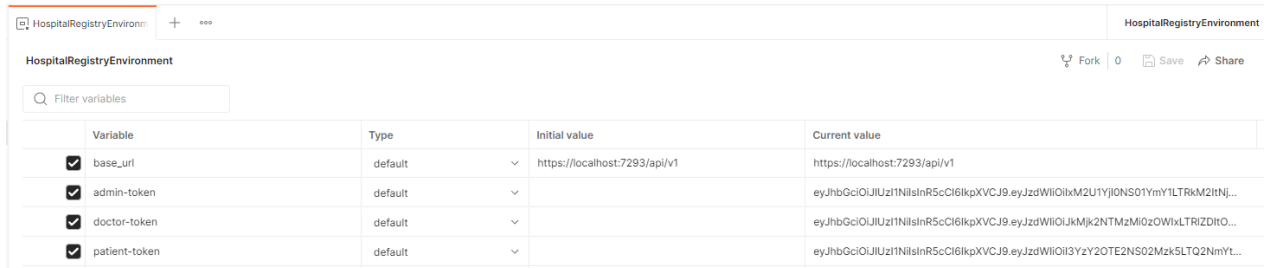
✓ HospitalRegistry.Tests (223)	20,6 с
✓ HospitalRegistry.Tests.AppointmentsSe...	1,2 с
✓ HospitalRegistry.Tests.DiagnosesServic...	566 мс
✓ HospitalRegistry.Tests.DoctorsServiceTe...	544 мс
✓ HospitalRegistry.Tests.IntegrationTests....	2,4 с
✓ HospitalRegistry.Tests.IntegrationTests....	380 мс
✓ HospitalRegistry.Tests.IntegrationTests....	4,5 с
✓ HospitalRegistry.Tests.IntegrationTests....	3,4 с
✓ HospitalRegistry.Tests.IntegrationTests....	2,5 с
✓ HospitalRegistry.Tests.PatientsServiceTe...	149 мс
✓ HospitalRegistry.Tests.RepositoryTests (...)	4,9 с
✓ HospitalRegistry.Tests.SchedulesService...	96 мс

Результат покриття тестами:

Total	84%	191/1226
HospitalRegistry.Application <ul style="list-style-type: none"> HospitalRegistry.Application <ul style="list-style-type: none"> Specifications: 97% (3/90) DTO: 91% (24/264) Services <ul style="list-style-type: none"> JwtService: 100% (0/16) SchedulesService: 97% (2/78) DiagnosesService: 96% (4/96) AppointmentsService: 90% (26/252) PatientsService: 83% (18/108) DoctorsService: 81% (23/122) UserAccountsService: 39% (79/129) HospitalRegistry.Infrastructure <ul style="list-style-type: none"> HospitalRegistry.Infrastructure.Repositories: 83% (12/71) RepositoryBase: 83% (12/71) 	85%	179/1155

3. Експорт розробленого API у Postman

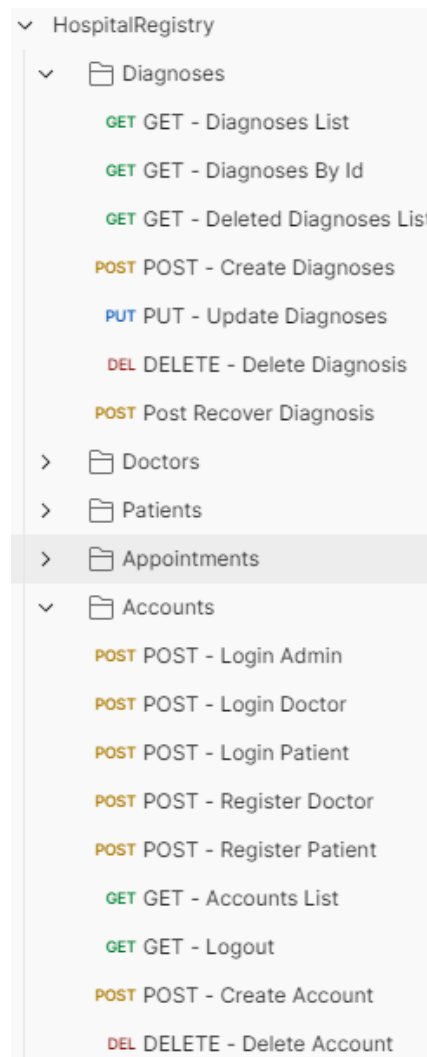
В програмному середовищі Postman було створено Environment для застосунку HospitalRegistry.



The screenshot shows the Postman Environment configuration for 'HospitalRegistryEnvironment'. It includes a search bar for variables and a table of defined variables.

Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/> base_url	default	https://localhost:7293/api/v1	https://localhost:7293/api/v1
<input checked="" type="checkbox"/> admin-token	default		eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjU1YjI0NS01YmY1LTlkM2ItNj...
<input checked="" type="checkbox"/> doctor-token	default		eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJkMjk2NTMzMjUzOTZlOTZlOTZl...
<input checked="" type="checkbox"/> patient-token	default		eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI3YzY2OTE2NS02Mzk5LTQ2NmYt...

До кожного ендпоінту в контролері було створено запит у Postman. Запити розділені по папках за приналежністю до певних ресурсів.



Запит на вхід у систему в якості адміна:

Запит на список усіх наявних записів:

GET `{{base_url}}/appointments?pageSize=2&pageNumber=1`

Parameters • **Authorization** • Headers (8) Body Pre-request Script Tests Settings

Type Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#).

Token {{admin-token}}

Body Cookies (1) Headers (4) Test Results

Pretty Raw Preview Viewsize JSON

```

1 {
2   "list": [
3     {
4       "id": "f9a281cd-10b1-4f21-a55f-1e892ed1e10f",
5       "dateAndTime": "2023-12-11T12:00:00",
6       "doctor": {
7         "id": "29dc783b-b167-4d4e-8a13-b2e3ec27f85",
8         "name": "DoctorName1",
9         "surname": "DoctorSurname1",
10        "patronymic": "DoctorPatronymici",
11        "dateOfBirth": "30.11.2004",
12        "specialty": "Dermatologist",
13        "email": "testDoctorEmail@gmail.com",
14        "phoneNumber": "+380969999999"
15      },
16      "patient": {
17        "id": "ee210537-0cc1-489e-a1dc-dd62c7886dd",
18        "name": "PatientName1",
19        "surname": "PatientSurname1",
20        "patronymic": "PatientPatronymici",
21        "dateOfBirth": "28.11.2003",
22        "email": "patientEmail@gmail.com",
23        "phoneNumber": "+380969999999"
24      },
25      "appointmentType": "HealthVisit",
26      "diagnosis": "Allergy",
27      "extraClinicalData": "test clinic data",
28      "status": "Completed",
29      "conclusion": "test conclusion"
30    }
31  ],
32  "pageInfo": {
33    "totalItems": 2,
34    "currentPage": 1,
35    "pageSize": 2
36  }
37 }

```

Запит на створення нового доктору в системі:

The screenshot shows a REST client interface with a POST request to `{{base_url}}/doctors`. The request body is a JSON object with the following fields: `name`, `surname`, `patronymic`, `dateOfBirth`, `specialty`, `email`, and `phoneNumber`. The response body is a JSON object with the same fields, plus an `id` field.

```
1 {
2   "name": "DoctorName7",
3   "surname": "DoctorSurname7",
4   "patronymic": "DoctorPatronymic7",
5   "dateOfBirth": "2001-01-01",
6   "specialty": "Dermatologist",
7   "email": "testDoctorEmail7@gmail.com",
8   "phoneNumber": "+380959999999"
9 }
```

Body Cookies (1) Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "5f8c223b-4472-44b8-a1e9-4537ff380a5a",
3   "name": "DoctorName7",
4   "surname": "DoctorSurname7",
5   "patronymic": "DoctorPatronymic7",
6   "dateOfBirth": "01.01.2001",
7   "specialty": "Dermatologist",
8   "email": "testDoctorEmail7@gmail.com",
9   "phoneNumber": "+380959999999"
10 }
```

Запит оновлення назви діагнозу:

The screenshot shows a REST client interface with a PUT request to `{{base_url}}/diagnoses`. The request body is a JSON object with the following fields: `id` and `name`. The response body is a JSON object with the same fields.

```
1 {
2   "id": "d8601429-b9c2-45e8-9b24-9fed90a9901d",
3   "name": "new diagnoses"
4 }
```

Body Cookies (1) Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "d8601429-b9c2-45e8-9b24-9fed90a9901d",
3   "name": "new diagnoses"
4 }
```

Запит на історію записів пацієнта:

GET
{{base_url}}/patients/ee210537-0cc1-489e-a1dc-dd62c78860dd/appointments/history

Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings

Type
Bearer Token
Heads up! These parameters hold sensitive data. To keep this data secure, we've replaced them with variables.
Token
{{patient-token}}

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Body
Cookies (1)
Headers (4)
Test Results

Pretty
Raw
Preview
Visualize
JSON

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
{
 "list": [
 {
 "id": "2476af9d-6e1b-43df-896a-6ce0f6e7c914",
 "dateAndTime": "2023-12-11T14:00:00",
 "doctor": {
 "id": "29dc783b-b167-4d4e-8a13-e2e3ec27f85",
 "name": "DoctorName1",
 "surname": "DoctorSurname1",
 "patronymic": "DoctorPatronymic1",
 "dateOfBirth": "30.11.2004",
 "specialty": "Dermatologist",
 "email": "testDoctorEmail1@gmail.com",
 "phoneNumber": "+380959999999"
 },
 "patient": {
 "id": "ee210537-0cc1-489e-a1dc-dd62c78860dd",
 "name": "PatientName1",
 "surname": "PatientSurname1",
 "patronymic": "PatientPatronymic1",
 "dateOfBirth": "28.11.2003",
 "email": "patientEmail1@gmail.com",
 "phoneNumber": "+380959999999"
 },
 "appointmentType": "HealthVisit",
 "diagnosis": "Allergy",
 "extraClinicalData": "test clinica data",
 "status": "Completed",
 "conclusion": "test conclusion"
 }
],
}