**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**

(шифр, прізвище, ім'я, по батькові)

*ІП-15, Тонконог В.В.*

**Перевірив**

(прізвище, ім'я, по батькові)

*Бардін В.*

Київ 2023

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.

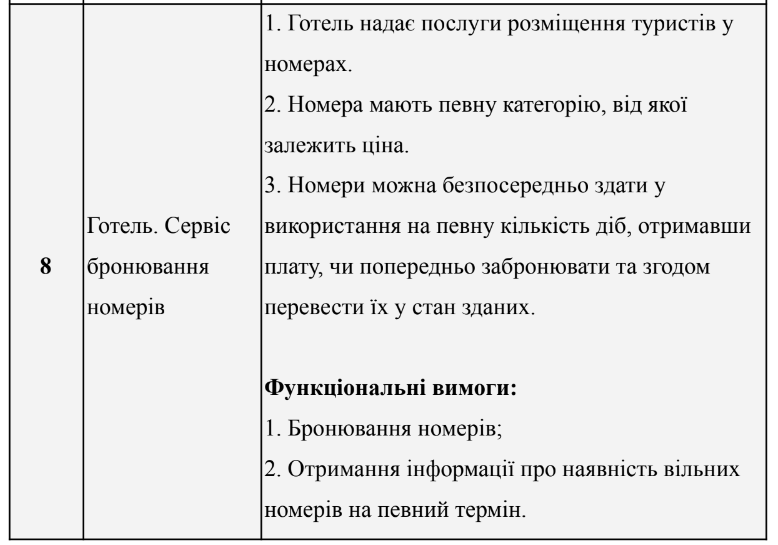
3. Оформити спроєктоване рішення у вигляді звіту до лабораторної

роботи.

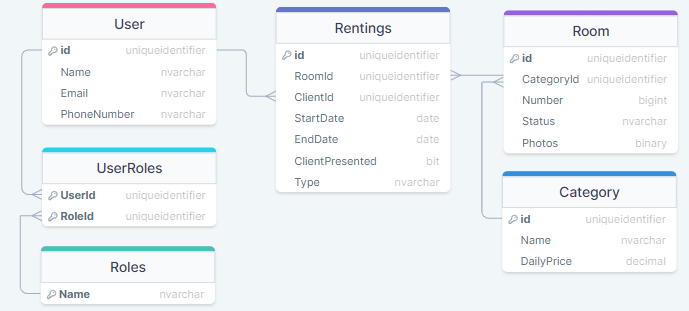
Документація:

1. Підготувати документацію(звіт до ЛР), яка включатиме опис

веб-API, а також структуру бази даних з урахуванням ER-діаграми.

Варіант

ER діаграма



**Таблиця:** Roles

**Призначення:** ролі користувачів.

**Поля:**

* Name (PK): NVARCHAR(255) Назва ролі

**Таблиця:** UserRoles

**Призначення:** Реалізація зв’язку багато до багатьох між користувачами та ролями.

**Поля:**

* RoleName (PK): NVARCHAR(255) Назва ролі .
* UserId (PK): GUID Ідентифікатор користувача.

**Таблиця:** User

**Призначення:** Користувач системи

**Поля:**

* Id (PK): GUID Ідентифікатор користувача.
* Name: NVARCHAR(255) Ім’я.
* Email: NVARCHAR(255) Електронна адреса.
* PhoneNumber: NVARCHAR(12) Номер телефону.

**Таблиця:** Ratings

**Призначення:** Здачі та бронювання номерів

**Поля:**

* Id (PK): GUID Ідентифікатор.
* RoomId: GUID Ім’я.
* ClientId: GUID Ідентифікатор клієнта.
* StartDate: Date Дата початку.
* EndDate: Date Дата кінця.
* ClentPresented: Bit Чи з’явився клієнт.
* Type: NVARCHAR(255) Тип.

**Таблиця:** Room

**Призначення:** Номери готелю

**Поля:**

* Id (PK): GUID Ідентифікатор.
* CategoryId: GUID Ідентифікатор категорії.
* Status: NVARCHAR(255) Статус кімнати.
* Number: INT Номер.
* Photos: BINARY Масив фотографій.

**Таблиця:** Category

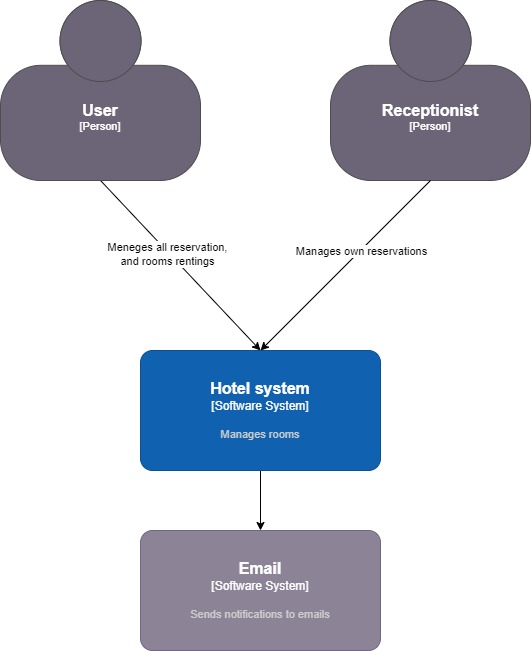
**Призначення:** Номери готелю

**Поля:**

* Id (PK): GUID Ідентифікатор.
* Name: NVARCHAR(255) Назва.
* DailyPrice: DECIMAL Вартість за один день.

# С4-діаграма

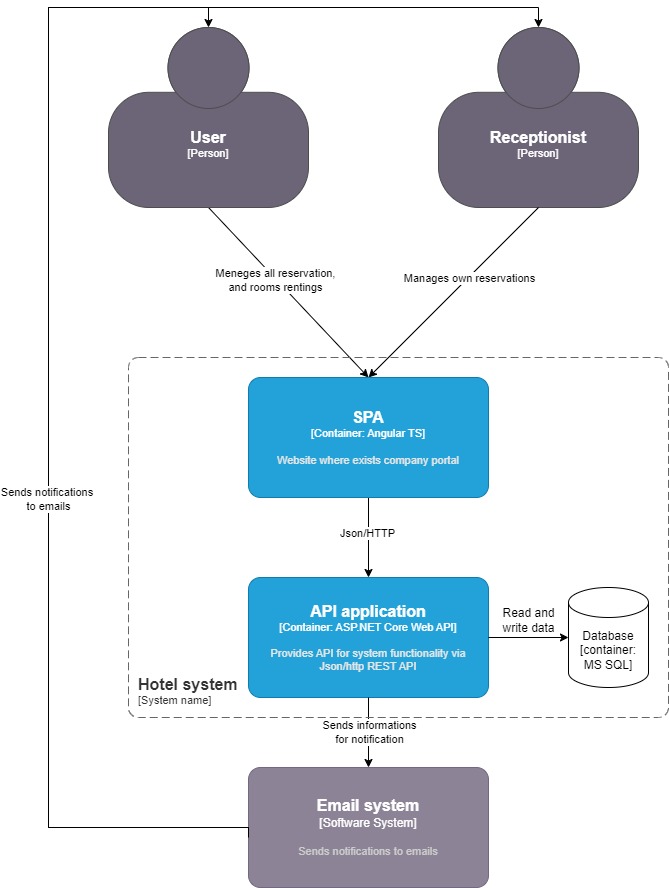
## System Context Diagram



У системи буде 2 типи користувачі:

* Клієнт – може забронювати номер та переглянути список доступних номерів
* Адміністратор – Керує всіма даними які зберігаються в системі.

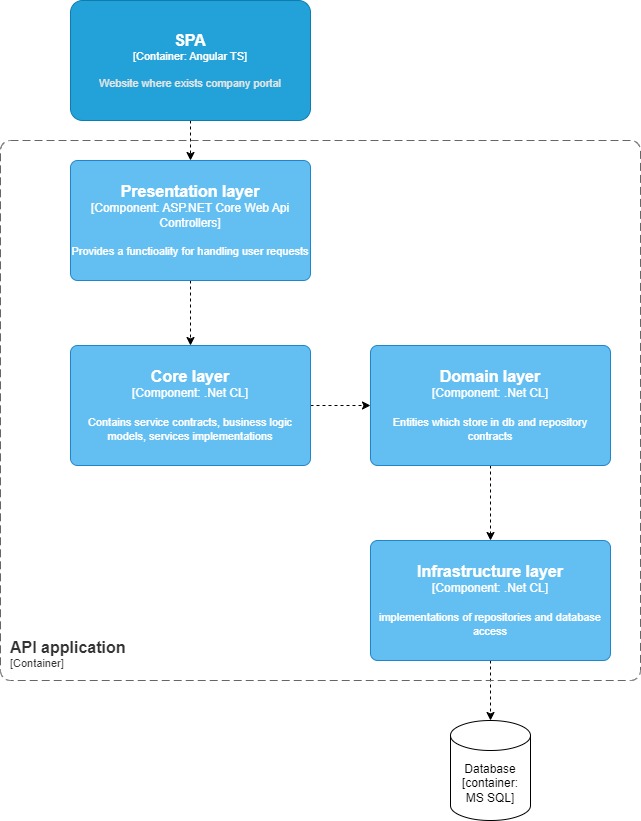
## Container Diagram



Система розділена на три основні контейнери:

* Single Page Application (SPA) – Веб застосунок.
* API application – серверна частина додатку.
* Database – База даних застосунку.

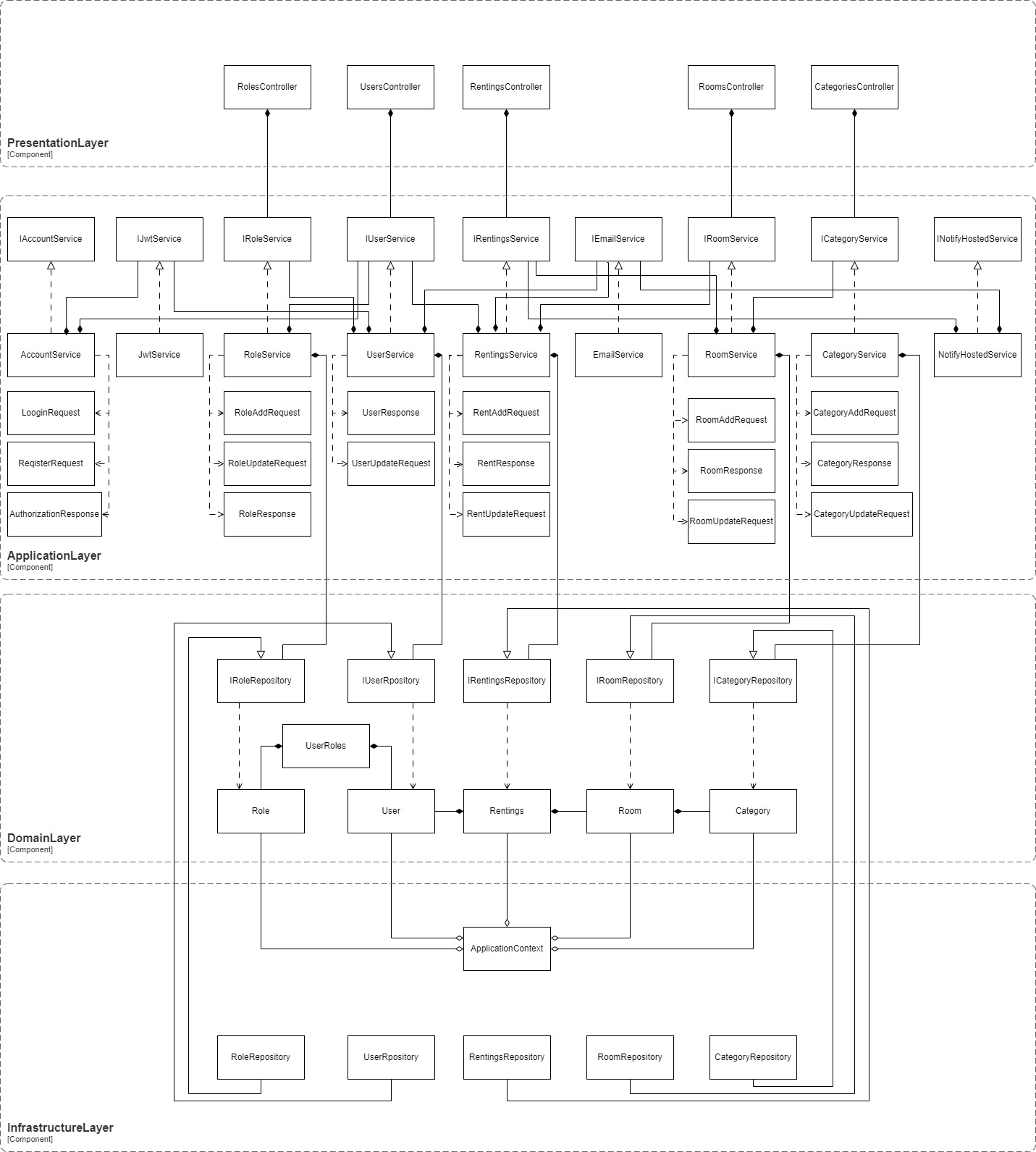
## Component Diagram



Для того аби розробити серверну частину застосунку було взято за основу архітектурний паттерн “Clean Architecture”:

* Core – бізнес логіка застосунку (сервіси, DTO…)
* Domain – Сутності бази даних та контракти репозиторіїв.
* Infrastructure – DAL, містить контекст бази даних та реалізації репозиторіїв.
* Presentation – надає API застосунку.

## Code Diagram



Всі компоненти застосунку розробляються у вигляді програмних бібліотек, окрім Presentation layer, який є Web API.

# API Endpoints

**Roles**

URL: /api/v1/roles

Метод: GET

Параметри: пагінація

Призначення: список ролей системи

Відповідь: RoleResponse[]

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

Метод: GET

Параметри: role id

Призначення: Отримання ролі за її id

Відповідь: RoleResponse

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

Метод: Delete

Параметри: role id

Призначення: Видалення ролі за її id

Відповідь: -

URL: /api/v1/roles

Метод: Post

Параметри: RoleAddRequest

Призначення: Створення нової системи

Відповідь: RoleResponse

URL: /api/v1/roles

Метод: Put

Параметри: RoleUpdateRequest

Призначення: оновлення існуючої ролі

Відповідь: RoleResponse

**Users**

URL: /api/v1/users

Метод: GET

Параметри: пагінація

Призначення: список користувачів системи

Відповідь: UserResponse[]

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

Метод: get

Параметри: User id

Призначення: Отримання даних користувача за id

Відповідь: UserResponse

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

Метод: Delete

Параметри: User id

Призначення: Видалення користувача за id

Відповідь: -

URL: /api/v1/users

Метод: Put

Параметри: UserUpdateRequest

Призначення: Оновлення данних користувача системи

Відповідь: UserResponse

**Account**

URL: /api/v1/login

Метод: Post

Параметри: LoginRequest

Призначення: Логін користувача в системі

Відповідь: AuthorizationResponse

URL: /api/v1/register

Метод: Post

Параметри: RegisterRequest

Призначення: реєстрація користувача в системі

Відповідь: AuthorizationResponse

**Categories**

URL: /api/v1/categories

Метод: Get

Параметри: пагінація

Призначення: Отримання всіх категорій кімнат

Відповідь: CategoryResponse[]

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

Метод: get

Параметри: Category id

Призначення: Отримання даних категорії за id

Відповідь: UserResponse

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

Метод: Delete

Параметри: Category id

Призначення: Видалення категорії за id

Відповідь: -

URL: /api/v1/categories

Метод: Post

Параметри: CategoryAddRequest

Призначення: Створення нової категорії кімнат

Відповідь: CategoryResponse

URL: /api/v1/categories

Метод: Put

Параметри: CategoryUpdateRequest

Призначення: Оновлення категорії кімнат

Відповідь: CategoryResponse

**Rooms**

URL: /api/v1/rooms

Метод: Get

Параметри: фільтри (категорія, дата, стан), пагінація

Призначення: Отримання всіх кімнат

Відповідь: RoomResponse[]

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

Метод: get

Параметри: Room id

Призначення: Отримання даних кімнати за id

Відповідь: RoomResponse

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

Метод: Delete

Параметри: Room id

Призначення: Видалення кімнати за id

Відповідь: -

URL: /api/v1/rooms

Метод: Post

Параметри: RoomAddRequest

Призначення: Створення нової кімнати

Відповідь: RoomResponse

URL: /api/v1/rooms

Метод: Put

Параметри: RoomUpdateRequest

Призначення: Оновлення кімнати

Відповідь: RoomResponse

**Rantings**

URL: /api/v1/rantings

Метод: Get

Параметри: фільтри (користувач, номер, проміжок дат), пагінація

Призначення: Отримання всіх бронювань

Відповідь: RantingResponse[]

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

Метод: get

Параметри: Ranting id

Призначення: Отримання даних бронювання за id

Відповідь: RantingResponse

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

Метод: Delete

Параметри: Ranting id

Призначення: Видалення бронювання за id

Відповідь: -

URL: /api/v1/rantings

Метод: Post

Параметри: RantingAddRequest

Призначення: Створення бронювання

Відповідь: RantingResponse

URL: /api/v1/rantings

Метод: Put

Параметри: RantingUpdateRequest

Призначення: Оновлення бронювання

Відповідь: RantingResponse