



Міністерство освіти й науки України

КПІ ім. Ігоря Сікорського

Звіт

із дисципліни «Сучасні технології розробки WEB-застосувань на

платформі Microsoft.Net»

студентки III курсу ФІОТ

групи ІК-12

Петканич Жанни Федорівни

Перевірив:
Бардін В.

Київ 2023

Тема роботи: Проектування REST веб-API

Мета 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.

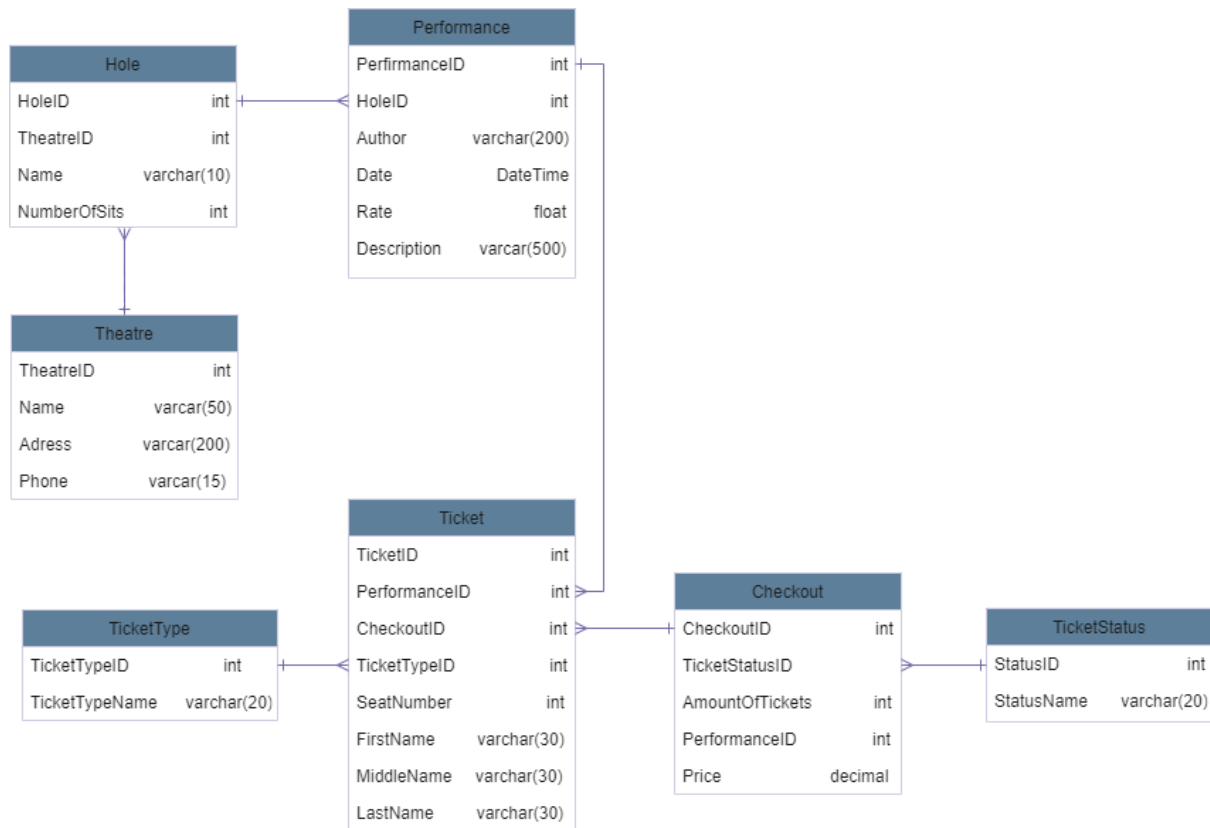
Завдання:

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.

Варіант 3:

3	Театральна каса. Розповсюдження квитків	<p>1. Афіша вистав надає функції пошуку потрібної вистави за автором, назвою, жанром та датою.</p> <p>2. Кожній виставі відповідає кількість квитків різної вартості.</p> <p>3. Квитки можна безпосередньо продати чи попередньо забронювати та згодом перевести їх у стан проданих.</p> <p>Функціональні вимоги:</p> <p>1. Ведення афіши вистав;</p> <p>2. Продаж квитків</p>
----------	---	---

Хід роботи ER-діаграма



Таблиця: Performance

Поля:

- PerformanceID: GUID - унікальний ідентифікатор вистави.
- HoleID: GUID - унікальний ідентифікатор залу.
- Author: varchar(200) - ім'я автора.
- Rate: int - оцінка вистави.
- Description: varchar(500) - опис вистави.

Таблиця: Hole

Поля:

- HoleId: GUID - унікальний ідентифікатор залу.
- TheatreID: GUID - унікальний ідентифікатор театру.
- Name: varchar(10) - назва залу.
- NumberOfSits: int - кількість місць.

Таблиця: Theatre

Поля:

- TheatreID: GUID - унікальний ідентифікатор

- Name: varchar(50) - назва театру.
- Address: varchar(100) - адреса театру.
- Phone: varchar(15) - телефон театру.

Таблиця: TicketType

Поля:

- TicketTypeID: GUID - унікальний ідентифікатор типу квитка.
- TicketTypeName: varchar(20) - тип квитка.

Таблиця: Ticket

Поля:

- TicketID: GUID - унікальний ідентифікатор квитка.
- PerformanceID: GUID - унікальний ідентифікатор вистави.
- CheckoutID: GUID - унікальний ідентифікатор каси.
- TicketTypeID: GUID - унікальний ідентифікатор типу квитка.
- SeatNumber: int - номер місця.
- FirstName: varchar(30) - Ім'я покупця.
- Middle Name: varchar(30) - По-батькові покупця.
- Last Name: varchar(30) - Прізвище покупця.

Таблиця: Checkout

Поля:

- CheckoutID: GUID - унікальний ідентифікатор каси.
- TicketStatusID: GUID - унікальний ідентифікатор статусу квитка.
- AmountOfTickets: int - кількість квитків.
- Price: decimal - ціна квитка.

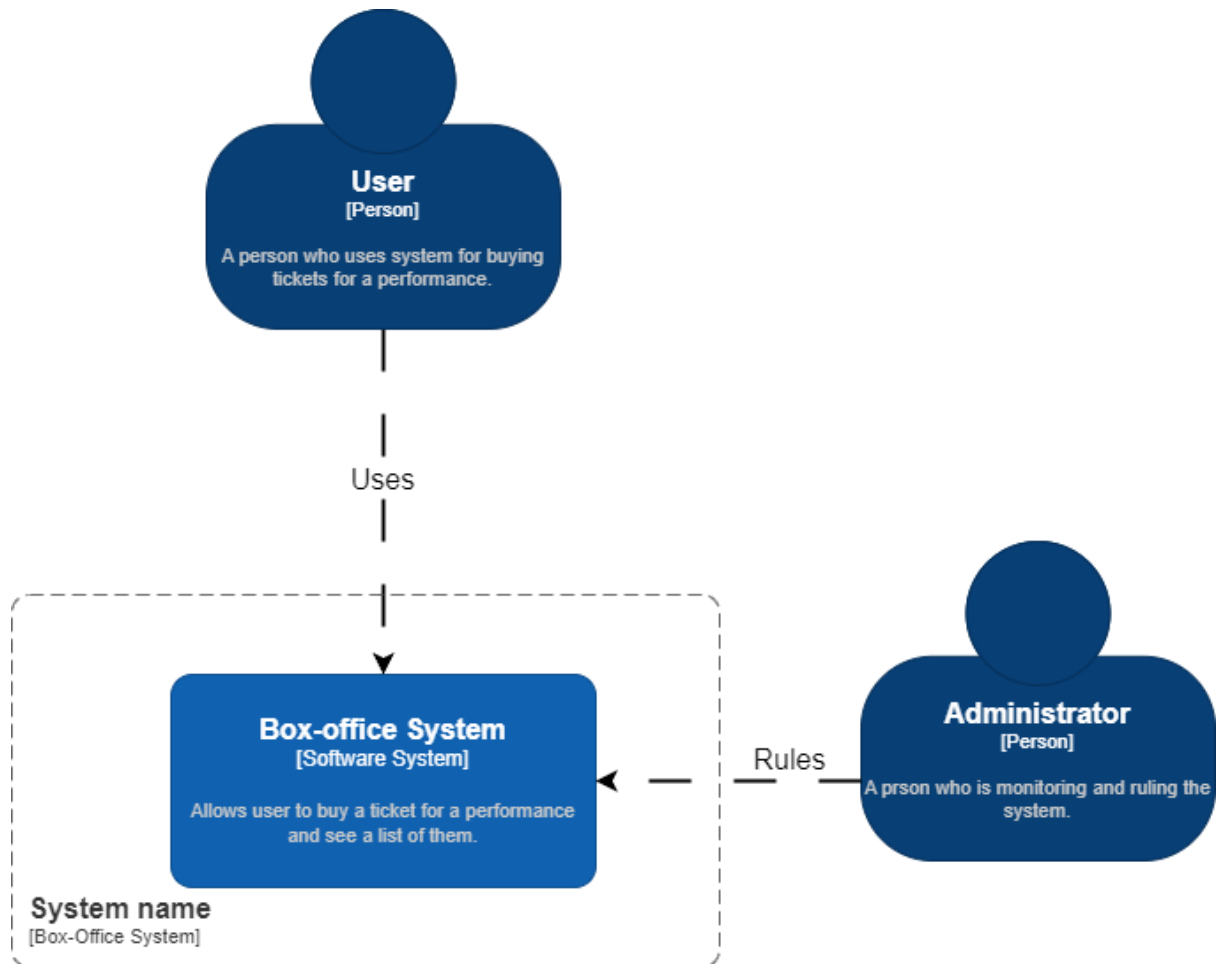
Таблиця: TicketStatus

Поля:

- StatusID: GUID - унікальний ідентифікатор статусу квитка.
- StatusName: varchar(20) - статус квитка.

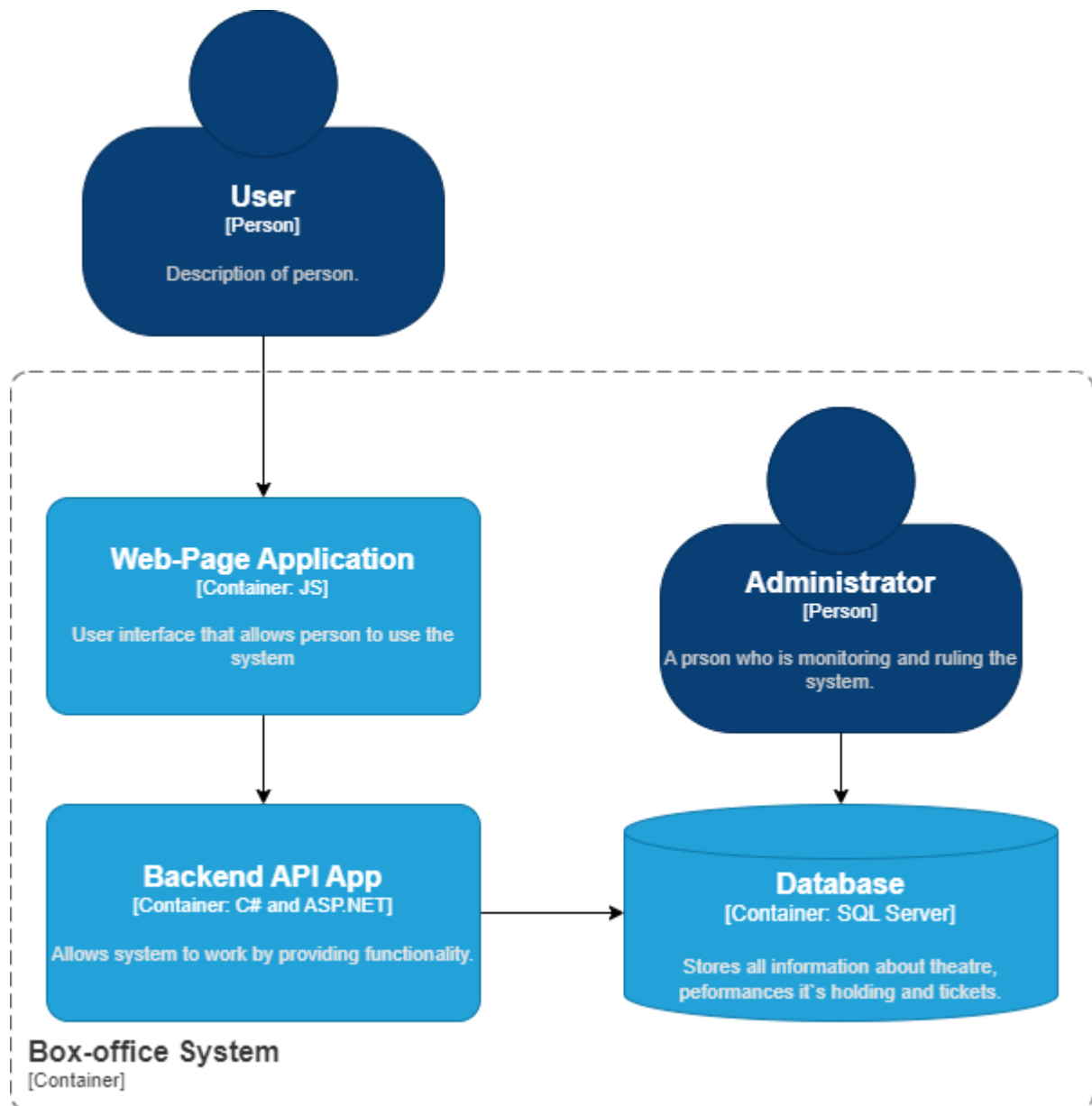
Діаграма C4

Рівень 1:



У системі є неавторизований користувач, який може переглядати театральні вистави, шукати їх за назвою, автором, ідентифікатором і датою. Система надає можливість бронювання та купівлі квитків. Також у системі є адміністратор, який може додавати та видаляти вистави.

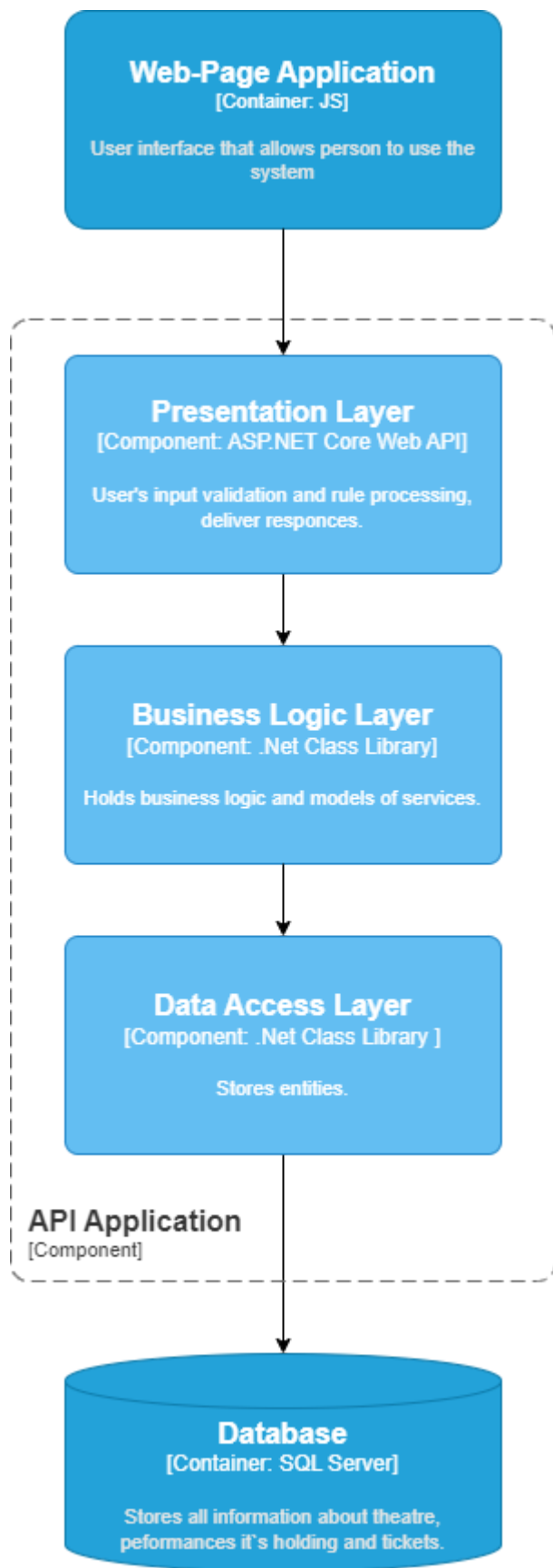
Рівень 2:



Система поділяється на такі компоненти:

1. Веб-сторінка, що надає можливість користувачу взаємодіяти з системою.
2. Серверний застосунок, що виконує запити користувача.
3. База даних, що зберігає інформацію яка необхідна для функціонування системи.

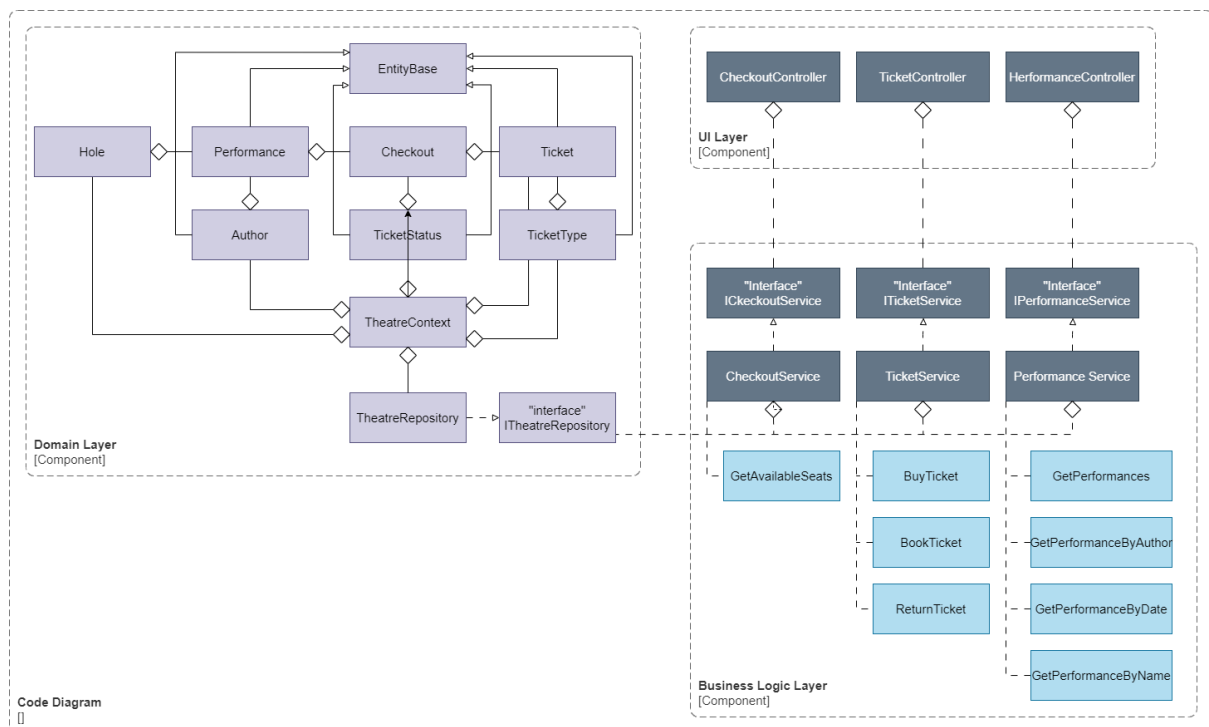
Рівень 3:



Для розробки програми було використано N-Layer архітектуру, що складається з 3 шарів:

1. Presentation Layer (UI Layer): Цей шар відповідає за представлення інформації користувачу та обробку його введення за допомогою графічного інтерфейсу.
2. Application Layer (Business Logic Layer): Цей шар містить функції, які визначають, як дані обробляються та бізнес-операції виконуються.
3. Data Access Layer (DAL): Шар доступу до даних відповідає за зберігання та отримання даних з бази даних.

Рівень 4:



DLL:

- TheatreContext - контекст БД.
- Performance, Checkout, Ticket, Author, TicketStatus, TicketType - сутності БД.

UIL:

- CheckoutController, TicketController, PerformanceController - контролери.

BLL:

- ICheckoutService, ITicketService, IPerformanceService - інтерфейси сервісів.

- CheckoutService, TicketService, PerformanceService - сервіси та нижче їх основні функції.

Кінцеві точки REST API (Endpoints):

1. Отримати всі вистави:
 - URL: api/performances
 - Метод: GET
2. Отримати виставу за ід:
 - URL: api/performances/{id}
 - Метод: GET
3. Пошук вистави за датою:
 - URL: api/performances/searchByDate/{Date}
 - Метод: GET
4. Пошук вистави за автором:
 - URL: api/performances/searchByAuthor/{Author}
 - Метод: GET
5. Бронювання квитків:
 - URL:
api/tickets/booking/{CheckoutID}/{FirstName}/{MiddleName}

/{LastName}
 - Метод: POST
6. Продаж квитків:
 - URL: api/tickets/sell/{CheckoutID}/{FirstName}/{MiddleName}

/{LastName}
 - Метод: POST
7. Отримати стан квитка за ід:
 - URL: api/tickets/{TicketID}/{StatusName}
 - Метод: GET
8. Скасування бронювання квитка
 - URL: api/tickets/{TicketID}
 - Метод: DELETE