**11/15/2023Міністерство 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)**

*ІC-11 Бацан І. О.*

(шифр, прізвище, ім'я, по батькові)

**Перевірив**

*Бардін В.*

(прізвище, ім'я, по батькові)

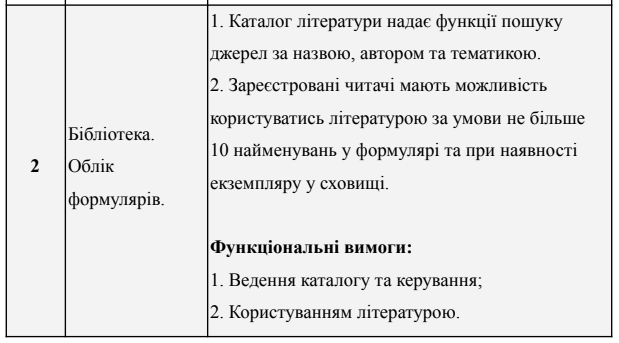
Київ 2023

# Лабораторна робота 3

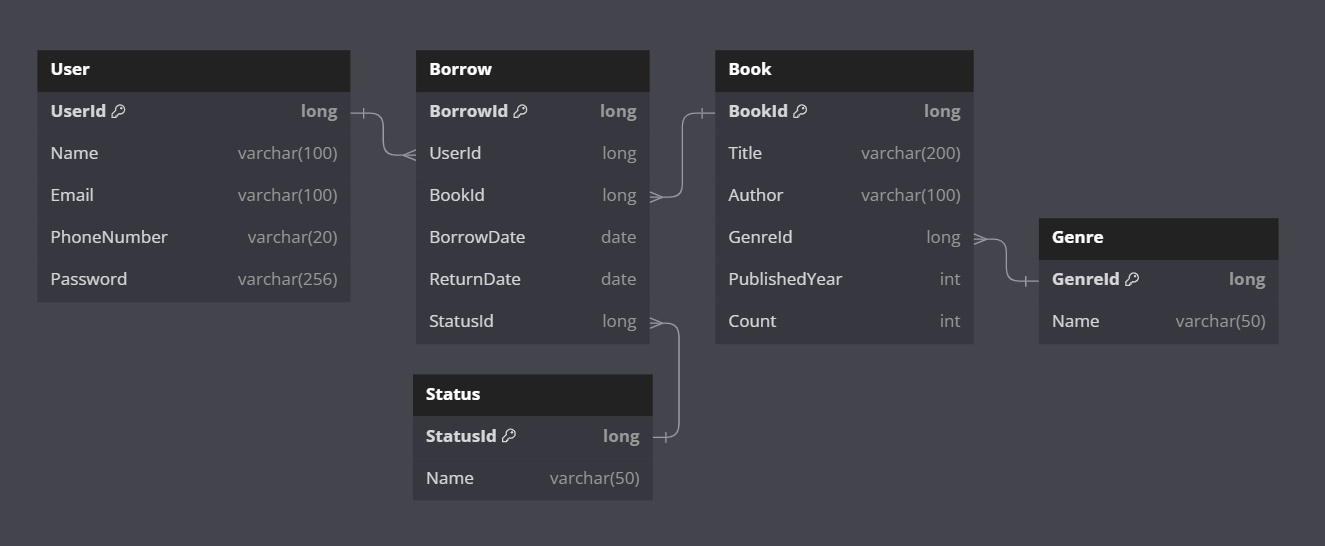
Проектування REST веб-API Завдання

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 спроєктоване рішення у вигляді звіту до лабораторної роботи.

Доменна область:



ER-діаграма



**Таблиця:** User

**Призначення:** зберігання інформації про читача.

# Поля:

* UserId: long – унікальний ідентифікатор користувача.
* Name: varchar(100) – ім’я
* Email: varchar(100) – електронна пошта читача
* PhoneNumber: varchar(20) – номер телефону
* Password: varchar(256) – хешовані дані паролю

**Таблиця:** Borrow

**Призначення:** Облік та відстежувати інформацію про видачу та повернення книг користувачами в бібліотеці.

# Поля:

* BorrowId: long – унікальний ідентифікатор видачі книжки
* UserId: long – унікальний ідентифікатор користувача
* BookId: long – унікальний ідентифікатор книжки
* BorrowDate: date – дата видачі книжки
* ReturnDate: date – дата повернення книжки
* Status: guid – статус видачі книжки

**Таблиця:** Book

**Призначення:** Облікова таблиця книжок в бібліотеці.

# Поля:

* BookId: long – унікальний ідентифікатор
* Title: varchar(200) – назва
* Author: varchar(100) – автор
* GenreId: long – унікальний ідентифікатор жанру
* PublishedYear: int – рік публікації
* Count: int – кількість екземплярів

**Таблиця:** Genre

**Призначення:** Таблиця для зберігання жанрів книг

# Поля:

* GenreId: long – унікальний ідентифікатор жанру
* Name: varchar(50) – назва жанру

**Таблиця:** Status

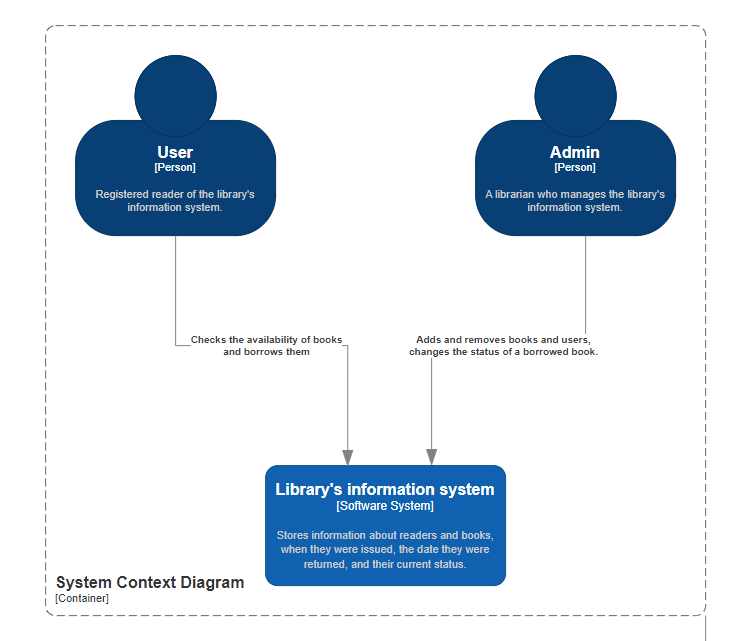
**Призначення:** статус виданої книги

# Поля:

* StatusId: long – унікальний ідентифікатор статусу
* Name: varchar(50) – назва статусу

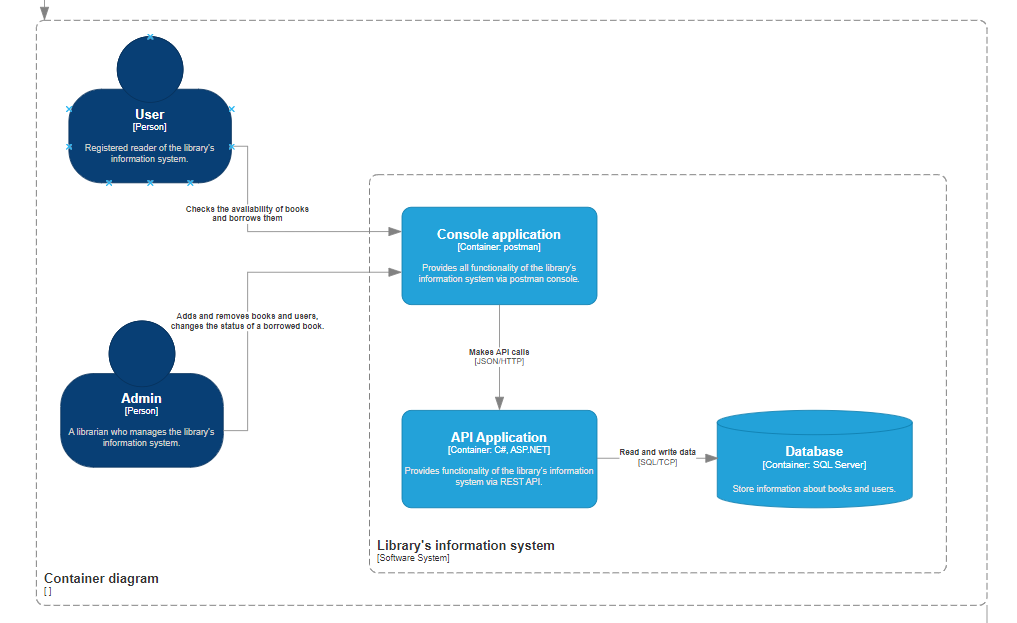
C4-діаграма

System Context Diagram



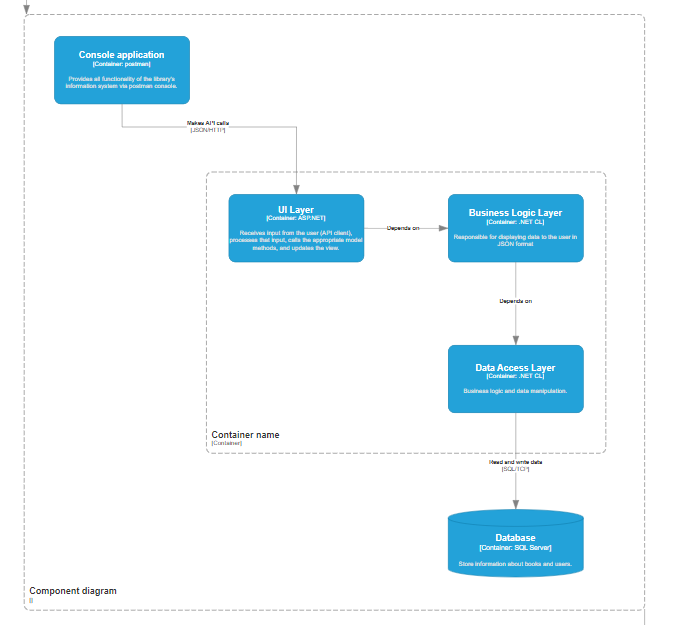
У інформаційної системи бібліотеки будуть 2 типи користувачів: Адмін – це бібліотекар який керує даними про звичайний юзерів, книжки та видачу книг. Користувач – це звичайний читач, який може переглядати наявність книжок у бібліотеці , а також фільтрувати їх.

Container Diagram



Інформаційна система бібліотеки розділяється на три основні компоненти: консольний застосунок, який надає функціонал бібліотеки засобами Postman, який реалізує функціонал через API, та база даних – яка зберігає усю інформацію про читачів, книжки та їх видачу.

Component Diagram



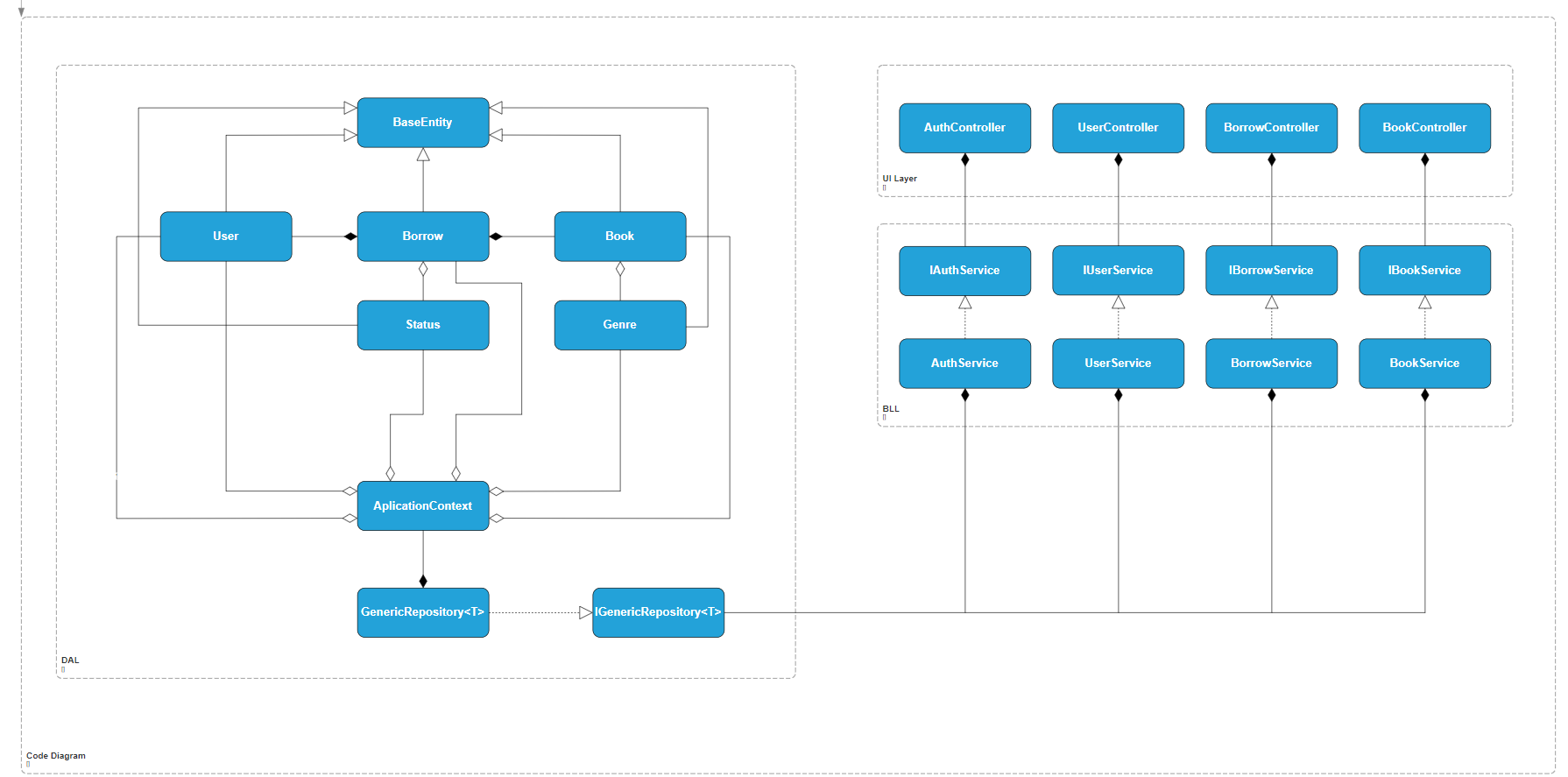
Для розробки програми .Net буде використовуватися типова трирівнева архітектура, яка розділить програму на такі рівні:

1. Рівень доступу до даних (DAL) - забезпечує простий доступ до даних, що зберігаються в SQL Server. Містить усі сутності домену та контекст бази даних.

2. Рівень бізнес-логіки (BLL) – містить всю бізнес-логіку програми. Залежить від DAL.

3. Рівень інтерфейсу користувача (UI Layer) – рівень, відповідальний за обробку запитів користувачів. Містить усі контролери. Залежить від BLL.

Code Diagram



Останнім кроком є ​​створення діаграми коду. Клас домену буде реалізовано на основі сутностей бази даних, які успадковуються від класу BaseEntity.

Він також містить реалізацію патерну GenericRepositoryBase та контексту бази даних ApplicationContext, який, у свою чергу, містить набір даних усіх сутностей. Entity Framework Core використовується для зв’язку з базою даних.

Рівень бізнес-логіки містить інтерфейси та їх реалізації.

За допомогою контролерів обробляються такі базові запити, які користувачі можуть відправляти в додаток:

1. **Додавання нового користувача**Метод: POSTURL: /api/usersПараметри: ім'я, номер телефону, електронная адреса.
2. **Видалення користувача**Метод: DELETEURL: /api/users/{id} Параметри: ідентифікатор користувача
3. **Видача книжки**Метод: POSTURL: /api/borrows Параметри: ідентифікатор користувача, ідентифікатор книжки, дата видачі.
4. **Зміна статусу виданої книжки**Метод: PATCHURL: /api/borrows/{id}Параметри: ідентифікатор видачі книжки, новий статус, дата повернення.
5. **Додавання нової книжки**Метод: POSTURL: /api/books Параметри: ідентифікатор книжки, назва, автор, жанр, рік публікації, кількість штук.
6. **Зміна кількості книжок в наявності**Метод: PATCHURL: /api/books/{id}Параметри: ідентифікатор книжки, кількість штук.
7. **Перегляд доступних книжок**Метод: GETURL: /api/booksПараметри: параметри фільтрації (за назвою, автором, жанром, роком публікації )