

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

Лабораторна робота №3
з курсу «Сучасні технології розробки WEB-застосунків на платформі
Microsoft.NET»
на тему: «Проектування REST веб-API»

Викладач:

Бардін В.

Виконав студент:

Лазюта Олексій

групи ІІІ-15 ФІОТ

Київ-2023

Завдання:

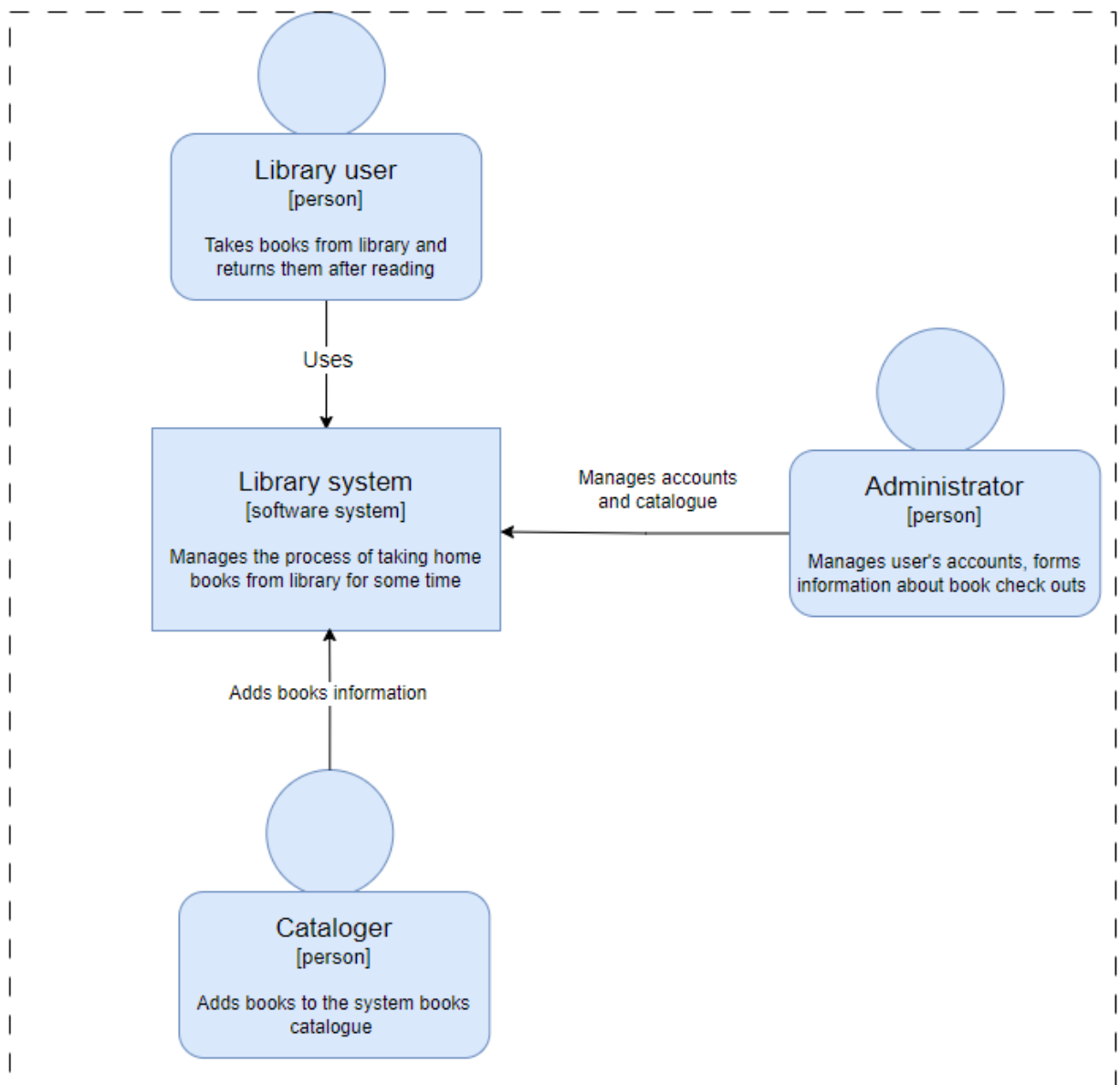
1. З дотриманням вимог REST-у спроектувати веб-API для обраної(згідно варіанту) доменної області, використовуючи методологію C4 для створення діаграми архітектури системи.
2. Створити ER-діаграму для DAL (Data Access Layer), яка відображатиме структуру бази даних веб-API.
3. Оформити спроектоване рішення у вигляді звіту до лабораторної роботи.

Варіант 2

| | | |
|---|-------------------------------------|--|
| 2 | Бібліотека. Облік формулярів. | <p>1. Каталог літератури надає функції пошуку джерел за назвою, автором та тематикою.</p> <p>2. Зареєстровані читачі мають можливість користуватись літературою за умови не більше 10 найменувань у формулярі та при наявності екземпляру у сховищі.</p> <p>Функціональні вимоги:</p> <p>1. Ведення каталогу та керування;</p> <p>2. Користуванням літературою.</p> |
|---|-------------------------------------|--|

1.
 - а. Контекстна діаграма:

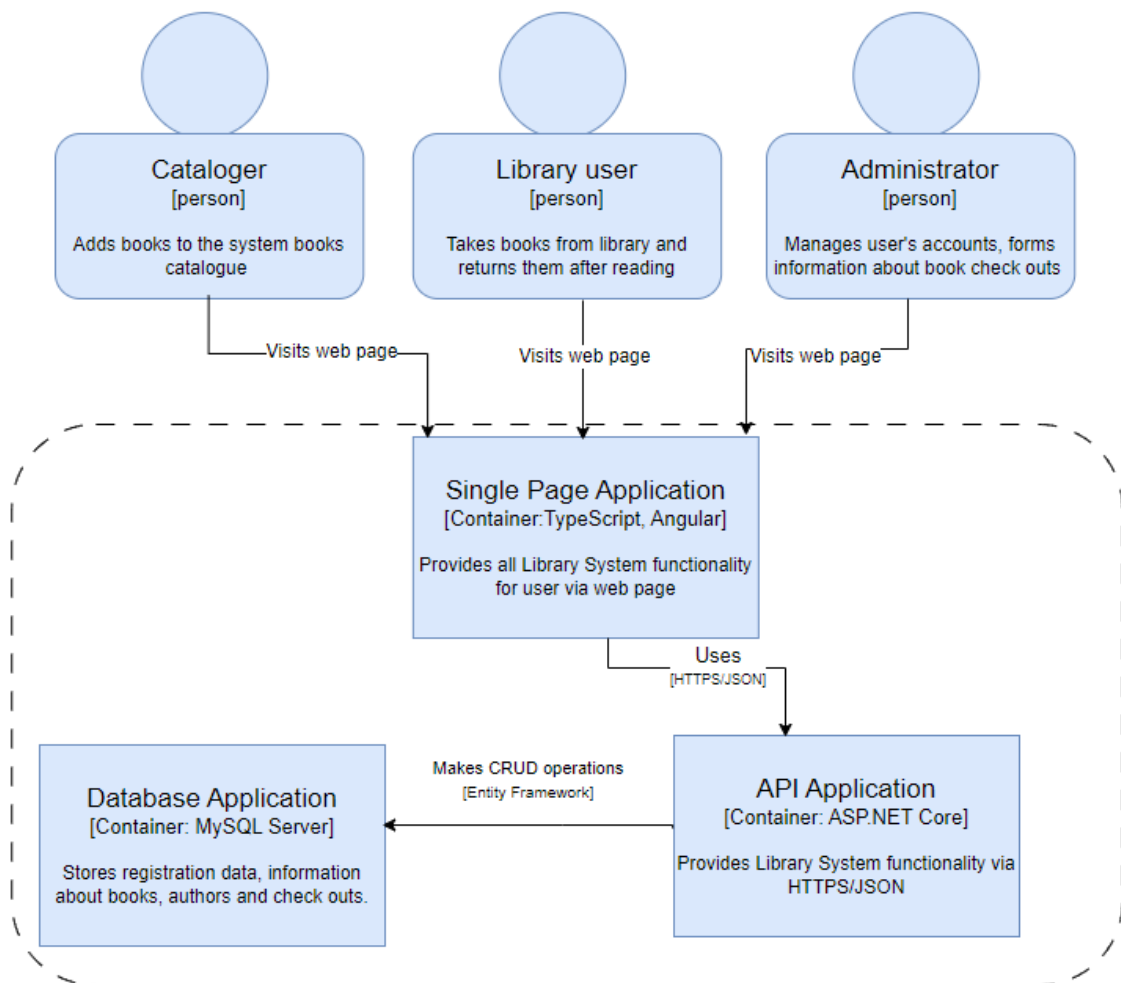
Context Diagram for Library System



Контекстна діаграма складається з системи бібліотеки, користувача бібліотеки, адміністратора та менеджера каталогу. Менеджер каталогу вносить в систему нові книги, які з'являються у бібліотеці. Адміністратор керує резервуванням книг: проводить операції резервування, слідкує за кількістю наявних примірників, керує користувачами в системі.

б. Діаграма контейнерів:

Container Diagram for Library System

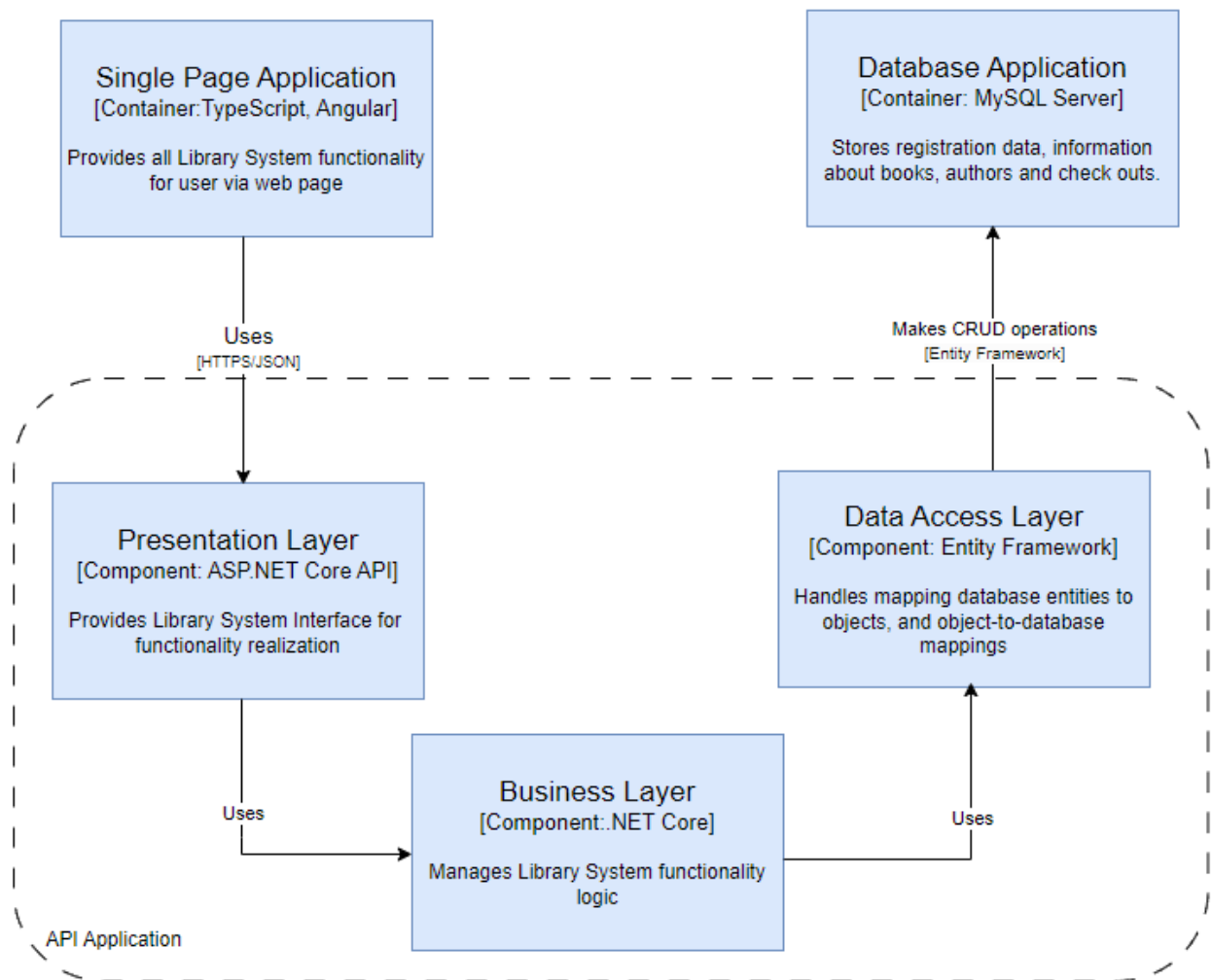


Діаграма контейнерів складається із трьох видів користувачів: менеджера каталогу, користувача бібліотеки, адміністратора, і трьох контейнерів в системі бібліотеки: веб сторінка, апі системи та база даних.

Для створення веб сторінки використовуватиметься мова TypeScript та Angular. Для комунікації із апі системи використовуватиметься HTTPS протокол та JSON. В свою чергу апі буде отримувати дані для операції із бази даних за допомогою Entity Framework.

с. Діаграма компонентів:

Component Diagram for Library System - API Application



Дана апі система містить три рівні відображення – Presentation layer, Business layer та Data access layer.

Presentation layer міститиме ендпоінти арі, які використовуватимуться для виклику відповідних операцій.

Business layer міститиме логіку проведення операцій над отриманими даними
Data access layer міститиме засоби для підключення до бази даних, отримання даних, маніпуляції з даними.

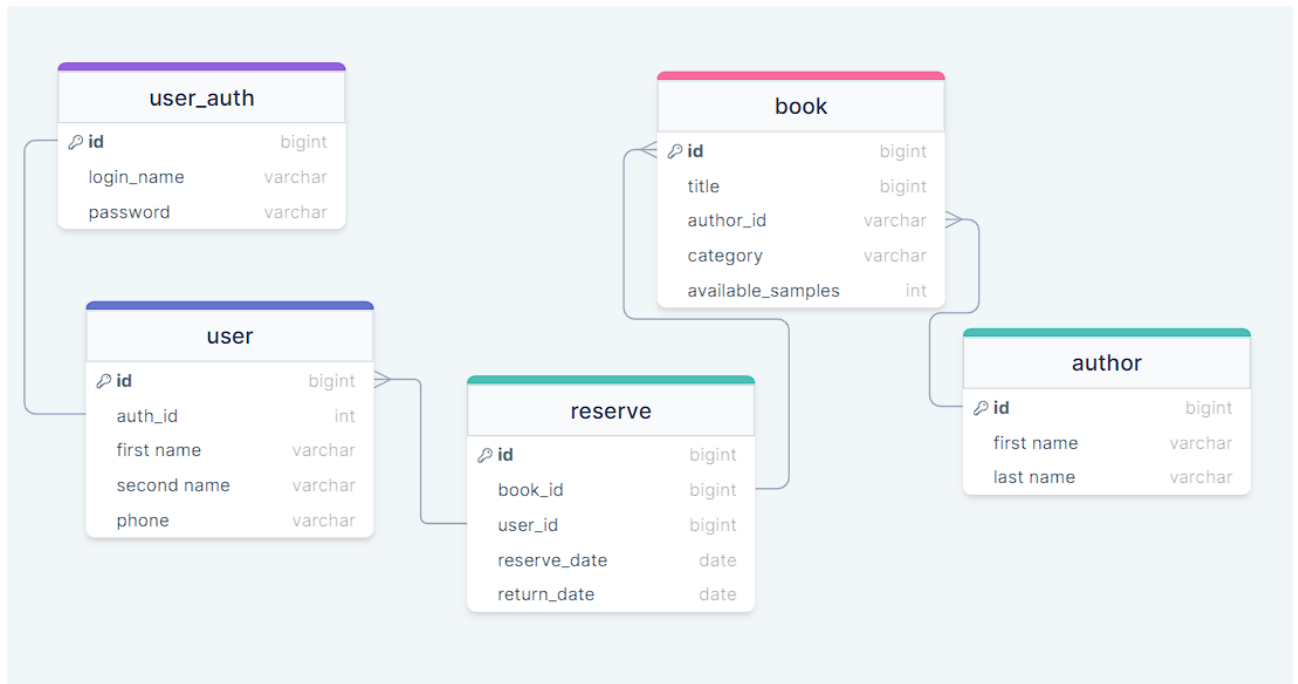
d. Діаграма коду:

```

classDiagram
    class PL {
        UserController
        AuthController
        BookController
        ReserveController
        AuthorController
    }
    class BBL {
        interface IUserService
        interface IBookService
        interface IReserveService
        interface IAuthService
        UserService
        BookService
        ReserveService
        AuthService
        BaseService
    }
    class DAL {
        interface IBaseRepository
        UserRepository
        BookRepository
        AuthorRepository
        ReserveRepository
        DbContext
        User
        UserAuth
        Book
        Author
        Reserve
    }
    PL --> BBL : Controller to Interface
    BBL --> BBL : Interface to Service
    BBL --> BBL : Service to BaseService
    BBL --> DAL : BaseService to IBaseRepository
    DAL --> DAL : Repositories to DbContext
    DAL --> DAL : DbContext to Entities
  
```

Сервіси реалізують відповідні інтерфейси для реалізації бізнес правил з використанням відповідних репозиторіїв, проведення операції із даними. Усі сервіси наслідуватимуть базовий сервіс який матиме логіку валідації даних, обробки помилок та інші спільні для усіх сервісів операції.

2. ER – діаграма:



Дана ER-діаграма має сутності user_auth, user, reserve, book, author

- User_auth – сутність реєстрації

Id – ідентифікатор

Login_name – логін користувача

Password – пароль користувача

- User – сутність користувача

Id – ідентифікатор

Auth_id – ідентифікатор інформації реєстрації

First name – ім'я користувача

Second name – прізвище користувача

Phone – номер телефону користувача

- Book – сутність книги

Id – ідентифікатор

Title – назва книги

Author_id – ідентифікатор автора

Category – тематика

Available samples – кількість доступних примірників

- Author – сутність автора

Id – ідентифікатор

First name – Ім'я автора

Second name – прізвище автора

- Reserve – сутність процесу резервування книги

Id – ідентифікатор

Book_id – ідентифікатор книги

User_id – ідентифікатор користувача

Reserve_date – дата резервування книги

Return_date – дата повернення книги

3. API Endpoints

POST:

POST /api/auth/

Створення користувача

POST /api/login/

Логін

POST /api/reserve/

Створення операції взяття книги з бібліотеки

POST /api/books/

Занесення нової книги у список

GET:

GET /api/user/{id}/

Отримання інформації про юзера

GET /api/user/{id}/reserves

Отримання списку резервувань книги

GET /api/user/{id}/reserves/{id}

Отримання інформації по резервуванню

GET /api/book?name={name}

Пошук книги по імені

GET /api/book?author={author}

Пошук книги по автору

GET /api/book?category={category}

Пошук книги по темі

GET /api/book?name={name}&author={author}&category={category}

Пошук книги по назві, автору, темі

GET /api/book?name={name}&author={author}

Пошук книги по назві, автору

GET /api/book?name={name}&category={category}

Пошук книги по назві, темі

GET /api/book?author={author}&category={category}

Пошук книги по автору, темі

PUT:

PUT /api/book/{id}/

Оновлення інформації по книзі

PUT /api/author/{id}/

Зміна ім'я автора

PUT /api/user/{id}/

Оновлення інформації по користувачу

DELETE:

DELETE /api/book/{id}

Видалення книги із системи

DELETE /api/author/{id}

Видалення автора із системи

DELETE /api/reserve/{id}

Видалення резерву із системи

DELETE /api/user/{id}

Видалення користувача із системи