



Міністерство освіти і науки України КПІ ім. Ігоря Сікорського

Факультет Інформатики та Обчислювальної Техніки

ЗВІТ

до лабораторної роботи №3 з модуля

**«Сучасні технології розробки WEB-застосунків на платформі
Microsoft.NET»**

Перевірив:

Викладач кафедри ІСТ

ФІОТ

Бардін В.

Виконав:

Поліщук Владислав

гр. ІК-13

Проектування REST веб-API

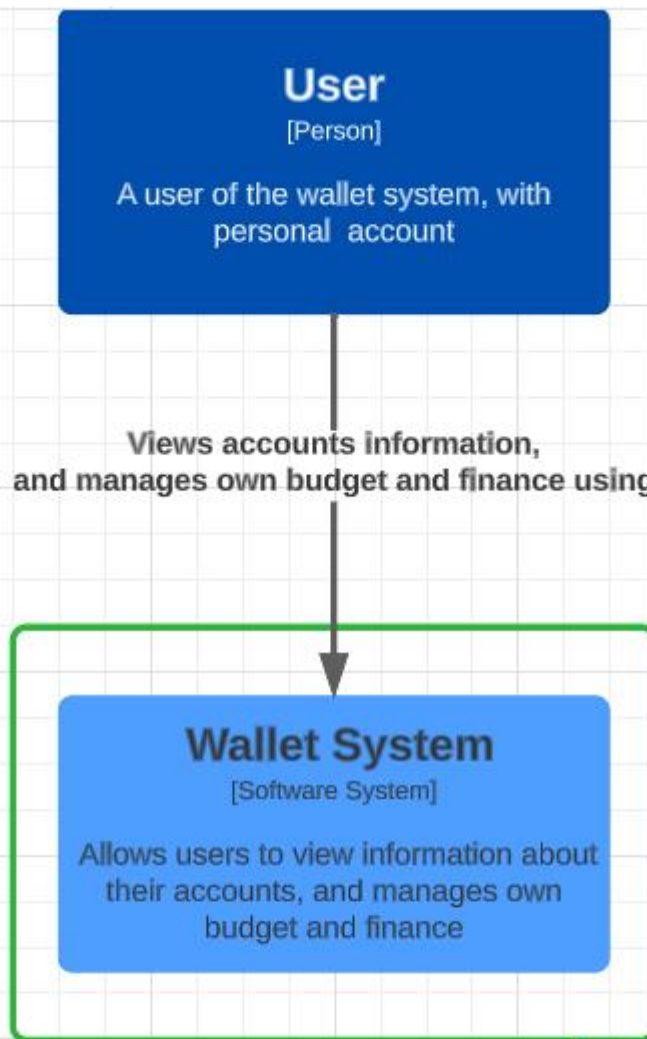
Теоретична частина:

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.

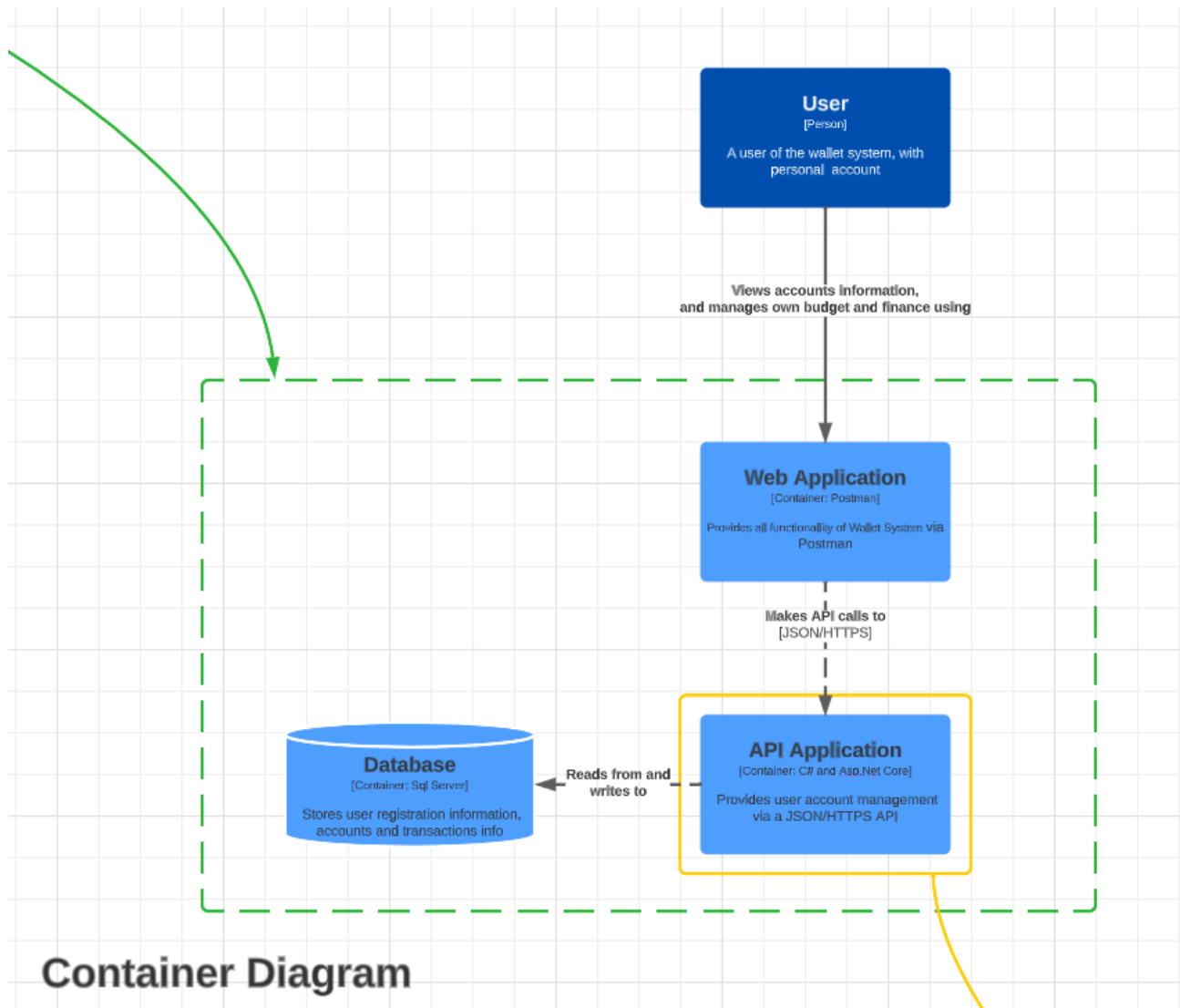
Практична частина:

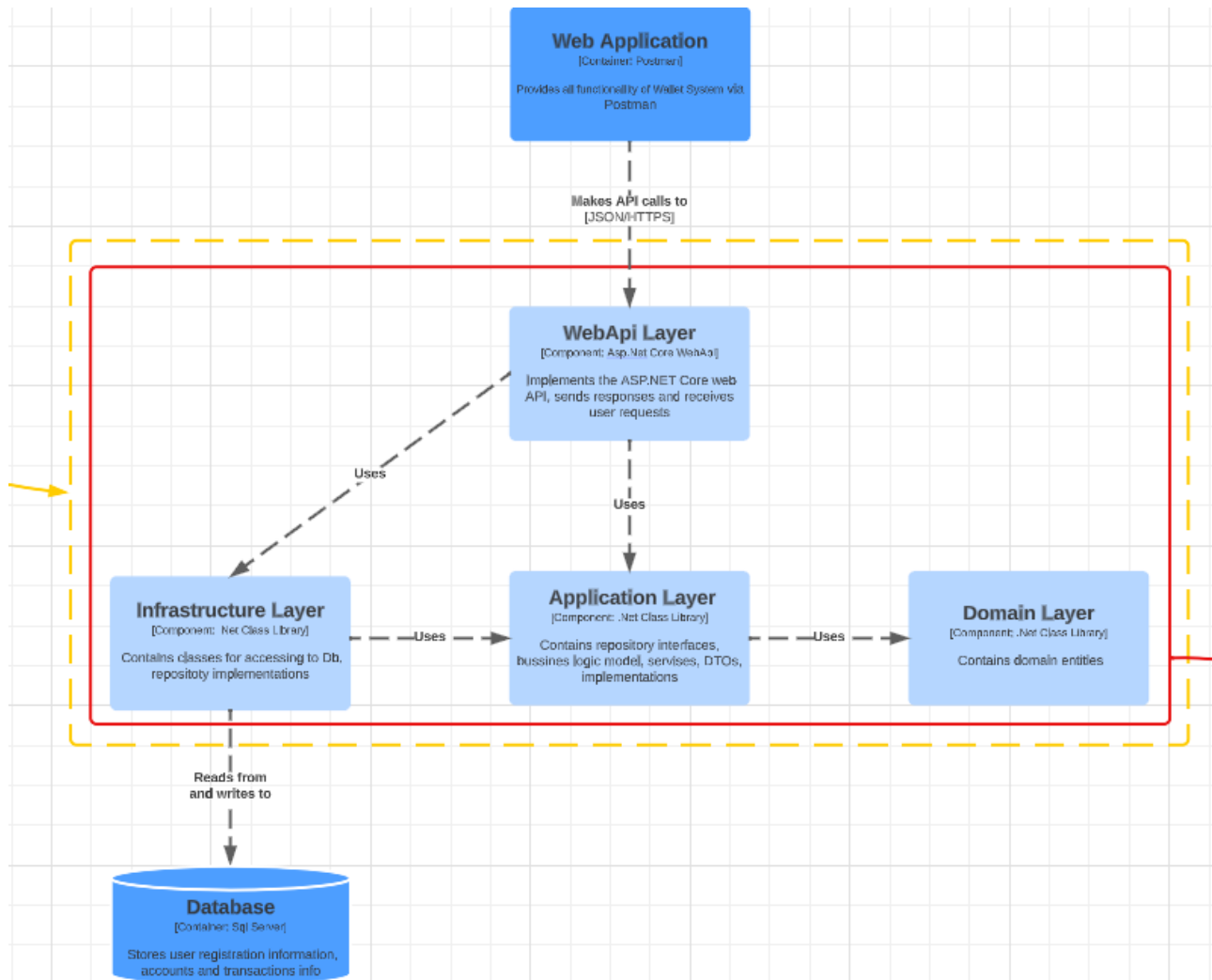
1. З дотриманням вимог REST-у спроектувати веб-API для обраної(згідно варіанту) доменної області, використовуючи методологію C4 для створення діаграми архітектури системи.
2. Створити ER-діаграму для DAL (Data Access Layer), яка відображатиме структуру бази даних веб-API.
3. Оформити спроектоване рішення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

4	Гаманець. Керування власним бюджетом та фінансами	<p>1. Власний бюджет складається з декількох рахунків, які поповнюються за заданими статтями прибутку.</p> <p>2. Гроші цих рахунків можуть бути переведені з одного на інший, можуть витратитись за заданими статтями витрат.</p> <p>3. Підсумовуючи витрати та прибутки, можливо отримати інформацію, скільки було витрачено/отримано загалом/за певною статтею по заданому рахунку.</p> <p>Функціональні вимоги:</p> <p>1. Ведення власного бюджету;</p> <p>2. Отримання звітної інформації по рахунках.</p>
---	---	---

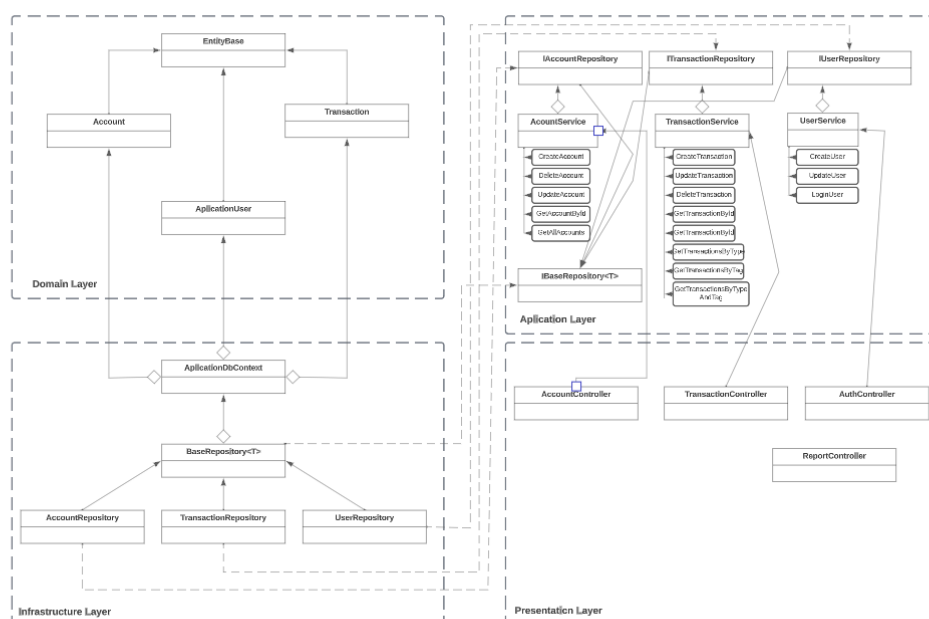


System Context Diagram

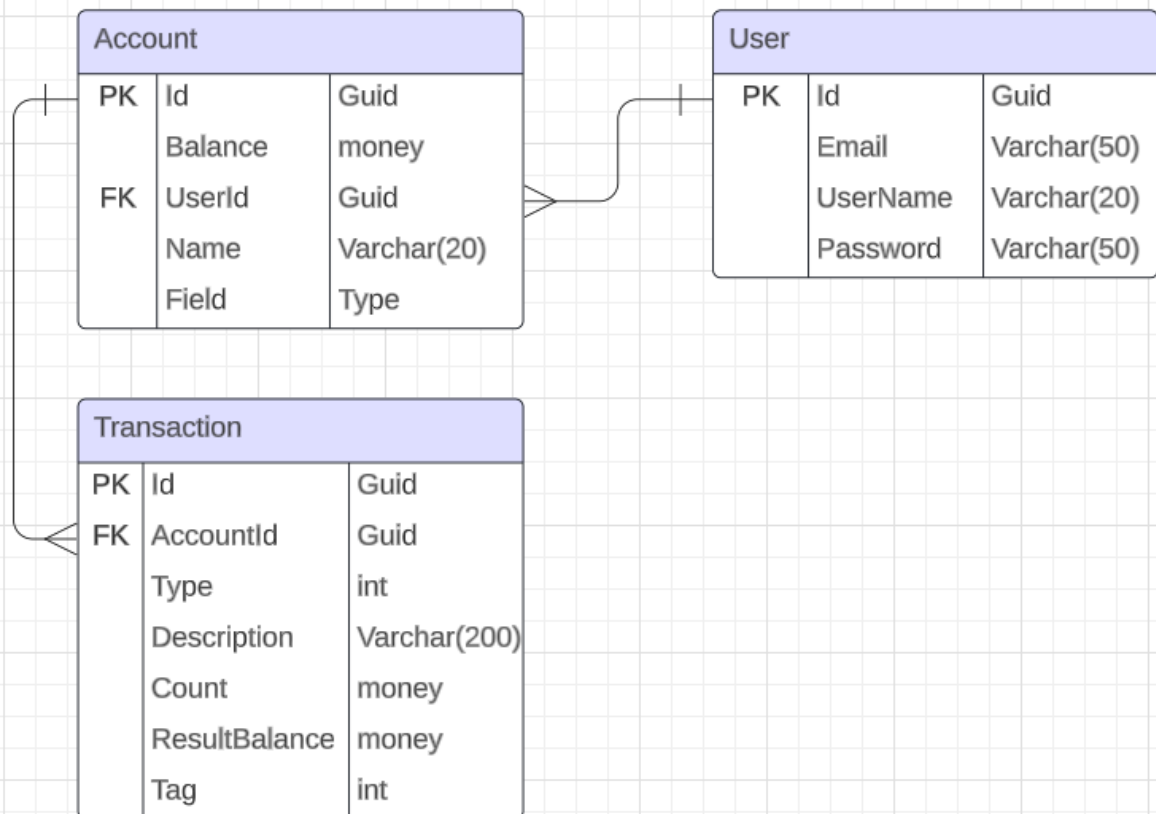




Component Diagram



Code



Опис ендпоінтів

1. Додавання рахунку:

- Метод: POST
- URL: </api/account>
- Параметри запиту: Новий об'єкт рахунку (назва рахунку, баланс тощо)
- Опис: Створення нового рахунку для користувача.

2. Видалення рахунку:

- Метод: DELETE
- URL: </api/account/{id}>
- Параметри запиту: Ідентифікатор рахунку
- Опис: Видалення рахунку користувача.

3. Зміна даних рахунку:

- Метод: PUT
- URL: </api/account/{id}>
- Параметри запиту: дані рахунку
- Опис: Оновлення даних рахунку(назва, баланс).

4. Отримати рахунок за Id:

- Метод: GET
- URL: </api/account>
- Параметри запиту: Ідентифікатор рахунку
- Опис: Отримання рахунку за Id.

5. Отримати всі рахунки юзера:

- Метод: GET
- URL: [/api/account/getAll](#)
- Параметри запиту: Ідентифікатор юзера
- Опис: Отримання всі рахунків юзера.

6. Додавання транзакції:

- Метод: POST
- URL: [/api/transaction](#)
- Параметри запиту: Новий об'єкт транзакції (тип транзакції, сума, опис, рахунок тощо)
- Опис: Додавання нової транзакції до обраного рахунку.

7. Видалення транзакції:

- Метод: DELETE
- URL: [/api/transaction](#)
- Параметри запиту: Ідентифікатор транзакції
- Опис: Видалення транзакції

8. Зміна даних транзакції:

- Метод: PUT
- URL: [/api/transaction](#)
- Параметри запиту: об'єкт
- Опис: Оновлення даних транзакції

9. Переказ коштів між рахунками:

- Метод: POST
- URL: [/api/transaction/transfer](#)
- Параметри запиту: рахунок відправник, рахунок отримувач та сума
- Опис: Переказ коштів з одного рахунку на інший

10. Отримання інформації про всі транзакції за рахунком:

- Метод: GET
- URL: [/api/transaction/{id}](#)
- Параметри запиту: Ідентифікатор рахунку
- Опис: Отримання інформації про транзакції.

11. Отримання інформації про транзакції за рахунком враховуючи тип та тег:

- Метод: GET
- URL: [/api/transaction/{id}&{type}&{tag}](#)
- Параметри запиту: Ідентифікатор рахунку
- Опис: Отримання інформації про транзакції.

12. Реєстрація користувача

- Метод: POST
- URL: [/api/auth/register](#)
- Параметри запиту: дані користувача
- Опис: Реєстрація нового юзера

13. Логін користувача

- Метод: POST
- URL: [/api/auth/login](#)

- Параметри запиту: дані користувача
- Опис: Вхід до свого акаунту