

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»



Кафедра інформаційних систем та технологій

Лабораторна робота №3

з дисципліни «Розробка програмного забезпечення на платформі .Net»

на тему:

«Проектування REST веб-API»

Викладачк:
Бардін В.

Виконав:
Студент групи ІС-11

Петраков Назар

Завдання:

Практична частина:

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.

Теоретична частина:

1. Ознайомитися з основами створення REST веб-API та методологією С4 для відображення архітектури системи.
2. Ознайомитися з основами створення ER-діаграм для представлення структури бази даних.

Документація:

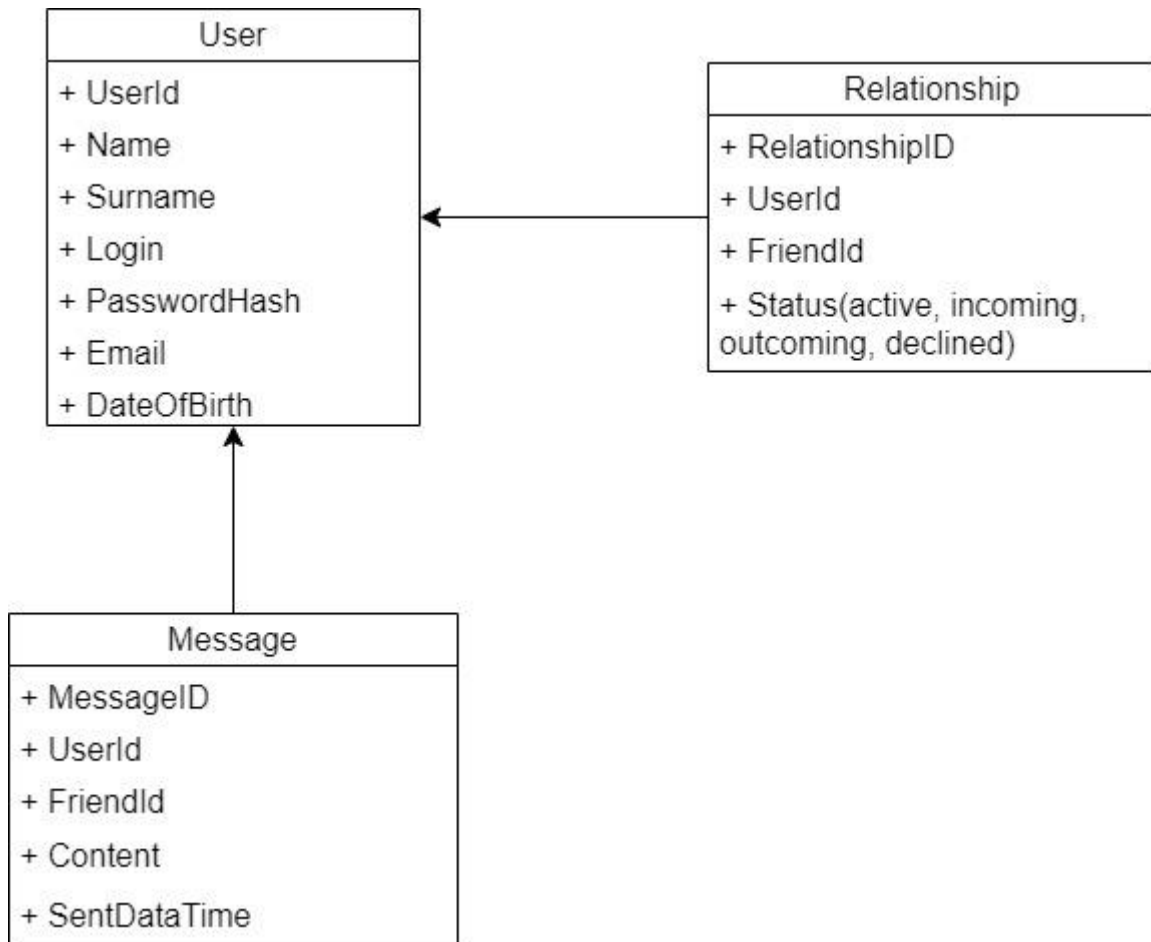
1. Підготувати документацію(звіт до ЛР), яка включатиме опис веб-API, а також структуру бази даних з урахуванням ER-діаграми.

Варіант:

6	Соціальна мережа. Комунікації між членами мережі	<p>1. Соціальну мережу складають групи її членів, пов'язаних між собою відносинами дружності.</p> <p>2. Кожний її член може керувати цими відносинами, а саме додавати до своєї мережі друзів: запрошувати до своїх друзів та просити запрошення для себе.</p> <p>3. Друзі можуть обмінюватись повідомленнями та передивлятися свої розмови у мережі.</p> <p>Функціональні вимоги:</p> <p>1. Створення соціальної мережі;</p> <p>2. Забезпечення спілкування в ній.</p>
---	---	--

Хід роботи

ER-Діаграма:



Тип та опис сутностей

User:

UserId (bigint) – Ідентифікатор користувача

Name(nvarchar(40)) – Ім'я користувача

Surname(nvarchar(40)) – Прізвище користувача

Login(varchar(32)) – Логін користувача

PasswordHash(varchar(255)) – Хешований пароль користувача

Email(nvarchar(255)) – Адреса електронної пошти користувача

DateOfBirth(datetime) – дата народження користувача

Relationship:

RelationshipId (bigint) – Ідентифікатор відносин

UserId (bigint) – Ідентифікатор користувача

FriendId(bigint) – Ідентифікатор друга користувача

Status(nvarchar(10)) – Статус дружби (active(дружба), incoming(прошення в друзі), outgoing(запрошення в друзі), declined(відхилено))

Message:

MessageId (bigint) – Ідентифікатор повідомлення

UserId (bigint) – Ідентифікатор користувача

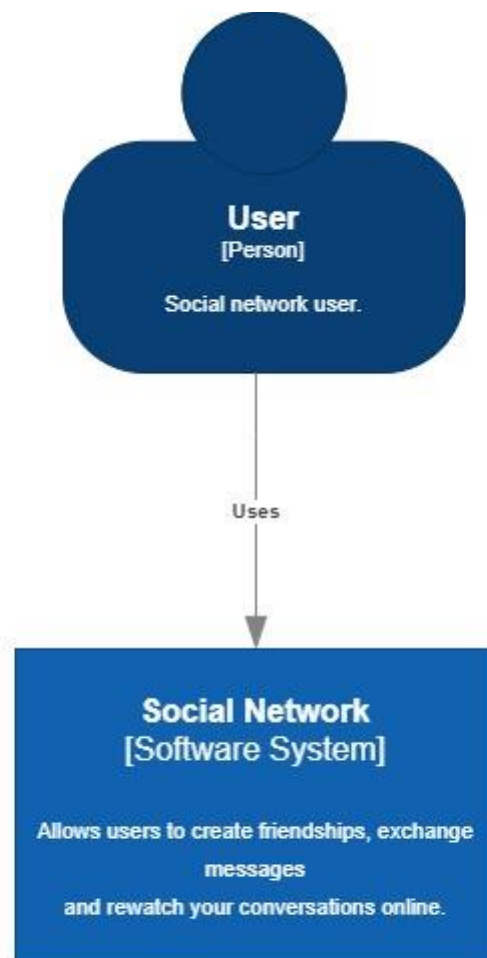
FriendId(bigint) – Ідентифікатор друга користувача

Content(nvarchar(max)) – Зміст повідомлення

SentDateTime(datetime) – Дата та час надсилання повідомлення

C4Model:

Context



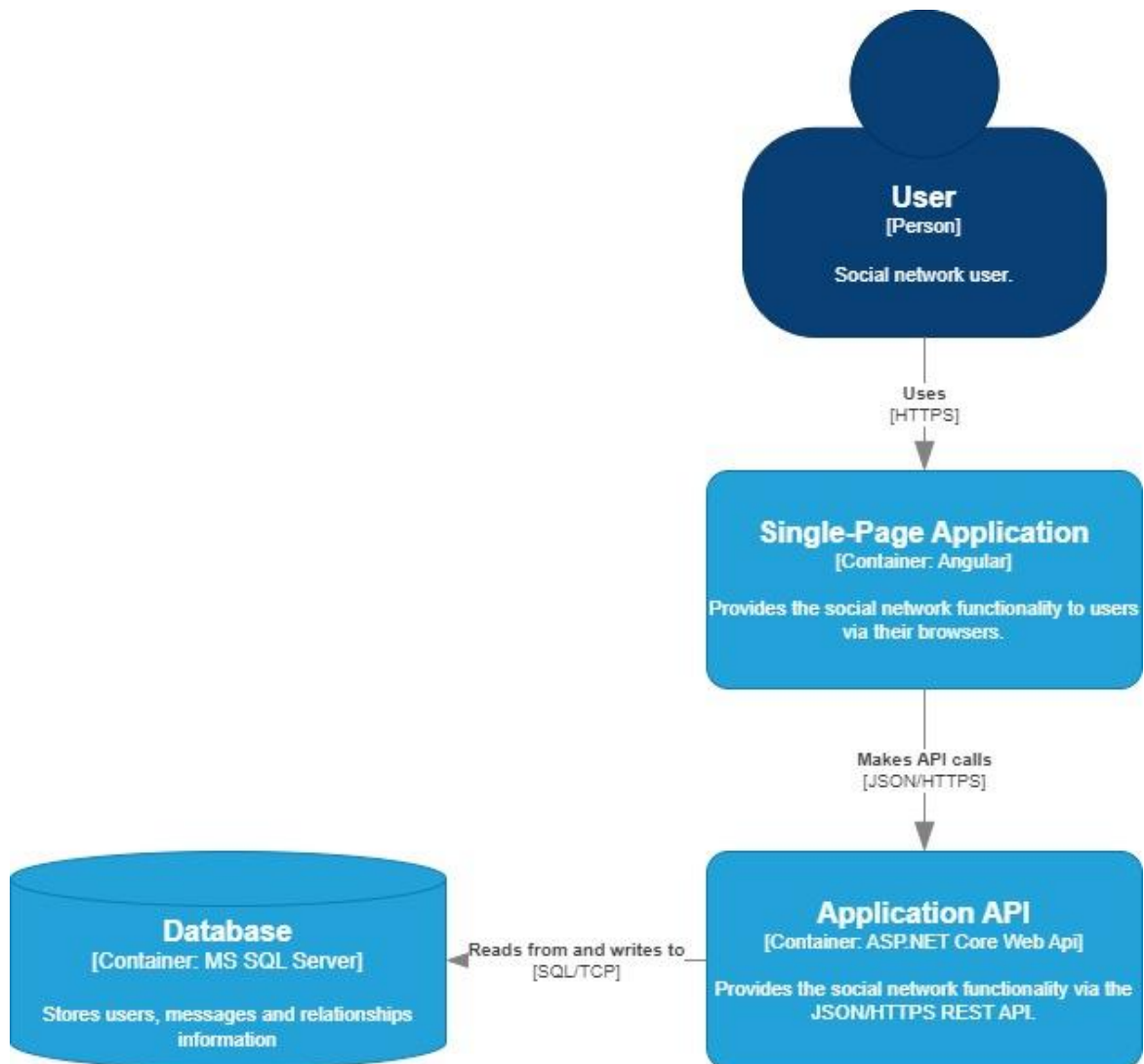
[System Context] Social Network

The System Context diagram for Social Network

Система «Соціальна мережа» має один тип користувачів:

User – користувач, який може зареєструватись та увійти в систему, використовувати її для створення відносин дружби з іншими користувачами, обмінюватись повідомленнями та переглядати їх.

Container



[Containers] Social Network

The container diagram for the Social Network

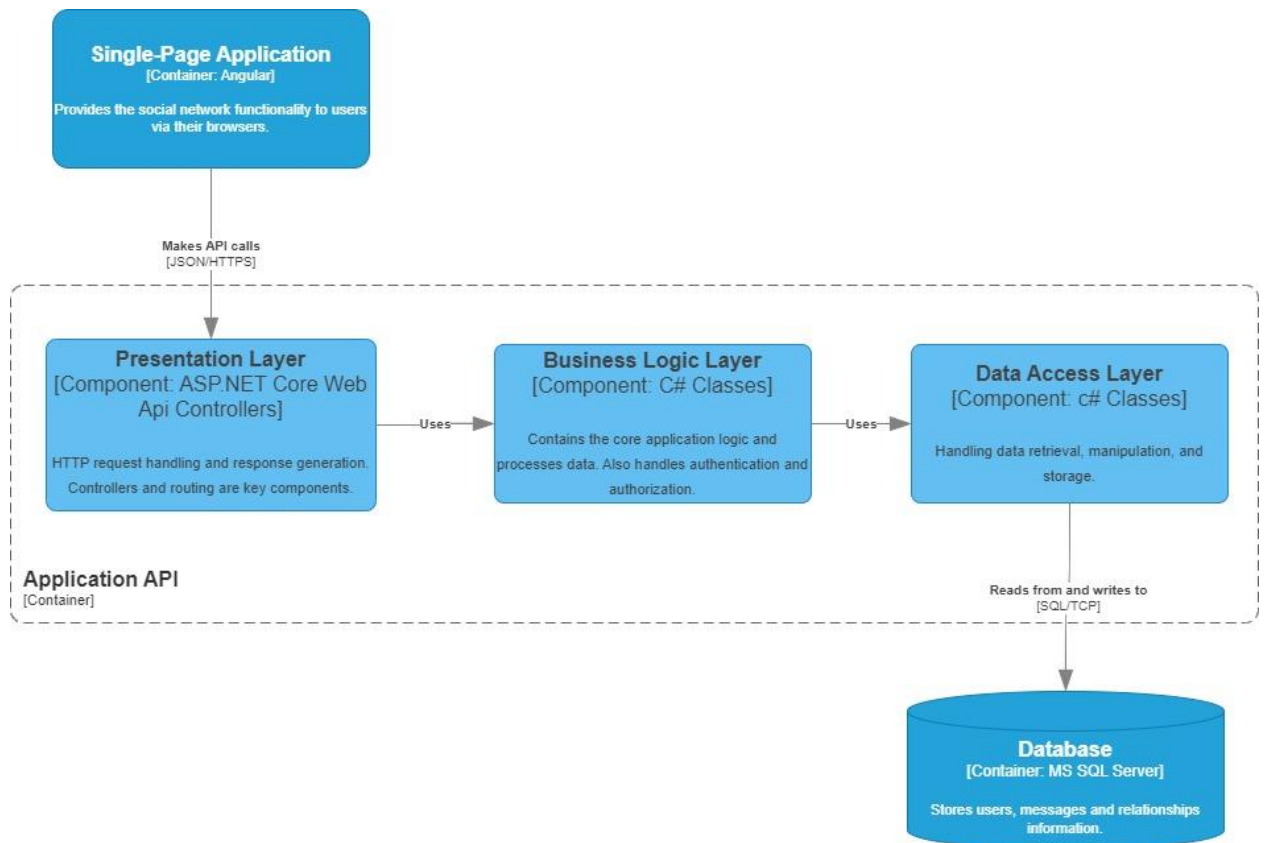
Система складається з 3 контейнерів:

Single-Page Application – надає функціонал системи через браузер.

Application API – надає функціонал системи, використовуючи сутності у форматі JSON та HTTP.

Database – зберігає інформацію про користувачів, повідомлення, відносини.

Component



[Components] Social Network

The Component diagram for Social Network

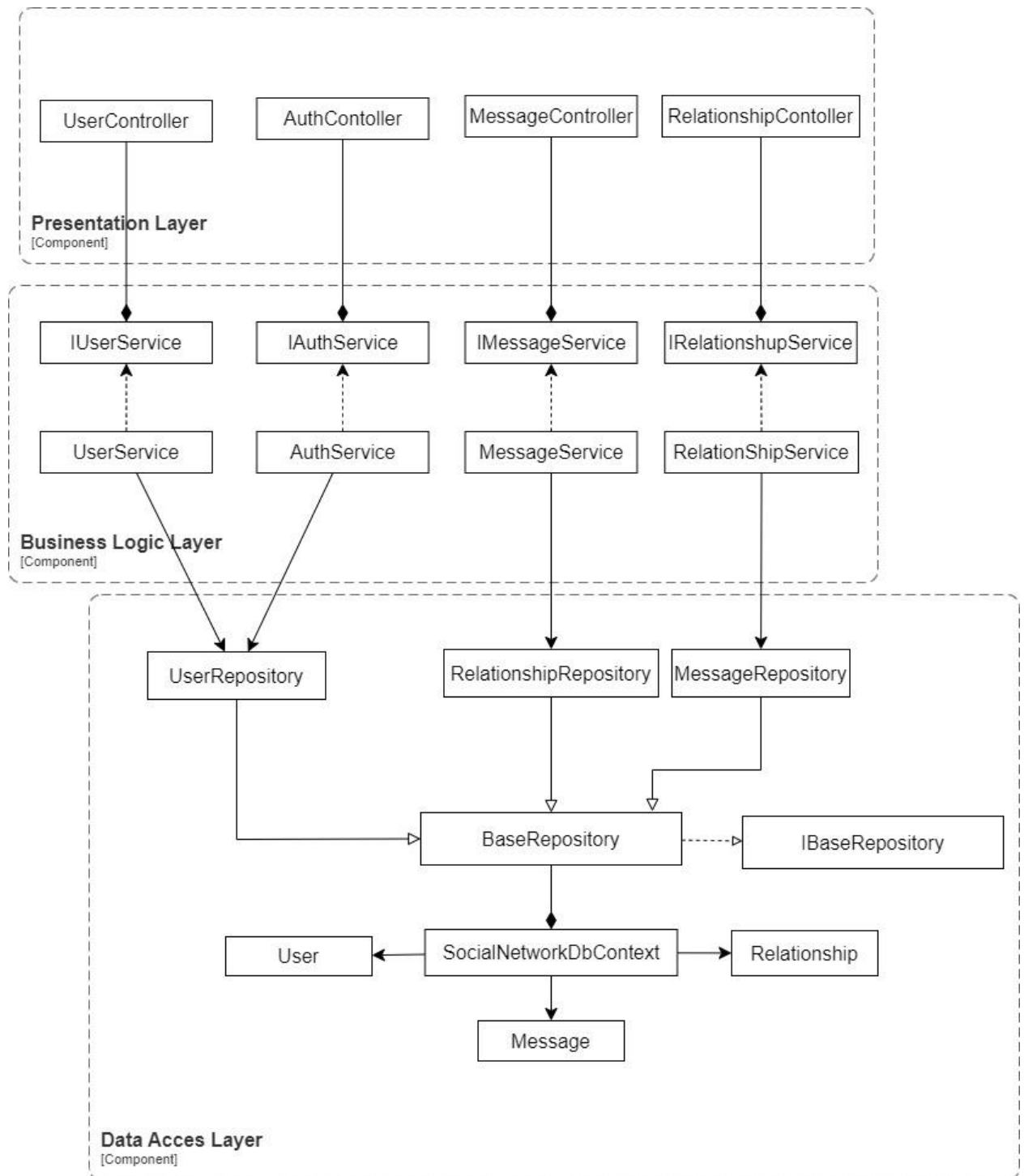
Була застосована багатошарова архітектура, яка складається з 3 шарів:

Presentation Layer – відповідає за взаємодію з користувачем. Включає обробку HTTP-запитів і генерацію відповідей. Містить контролери.

Business Logic Layer – містить основну логіку і обробляє дані на основі бізнес-правил.

Data Access Layer – забезпечує зберігання сутностей та надає до них доступ.

Code



Endpoints:

- Реєстрація нового користувача – **POST** /auth/register
- Запит для аутентифікації користувача(логін) – **POST** /auth/login
- Вихід користувача з системи – **POST** /auth/logout
- Зміна паролю – **POST**/auth/change-password
- Отримати список користувачів – **GET** /users

- Отримати інформацію про конкретного користувача за ID – **GET** /users/{id}
- Отримати список відносин конкретного користувача – **GET** /users/{id}/relationships
- Отримати список повідомлень конкретного користувача від друга – **GET** /users/{id}/friends/{id}/messages
- Отримати список користувачів, з якими залучений конкретний користувач у діалоги – **GET** /users/{id}/dialogs
- Створити запит на запрошення/прошення в друзі – **POST** /relationships
- Прийняти або відхилити запрошення/прошення в друзі – **PUT** /relationships/{id}
- Видалити відносини дружби – **DELETE** /relationships/{id}
- Створити нове повідомлення – **POST** /messages
- Видалити повідомлення – **DELETE** /messages/{id}