

Національний технічний університет України «КПІ»
Факультет інформатики та обчислювальної техніки
Кафедра Інформаційних систем та технологій

Лабораторна робота №3

з дисципліни « Сучасні технології розробки WEB-застосувань на платформі
Microsoft.NET»

на тему: « Проектування REST веб-API»

Виконав:
студент групи ІК-13
Хрисанфов Дмитро

Перевірив:
Бардін В.

2023 рік

Завдання:

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.

Варіант:

6	Соціальна мережа. Комунікації між членами мережі	<ol style="list-style-type: none">1. Соціальну мережу складають групи її членів, пов'язаних між собою відносинами дружності.2. Кожний її член може керувати цими відносинами, а саме додавати до своєї мережі друзів: запрошувати до своїх друзів та просити запрошення для себе.3. Друзі можуть обмінюватись повідомленнями та передивлятися свої розмови у мережі. <p>Функціональні вимоги:</p> <ol style="list-style-type: none">1. Створення соціальної мережі;2. Забезпечення спілкування в ній.
---	---	---

Хід виконання роботи:

С4 модель - це спосіб уявити архітектуру програмного забезпечення у вигляді ієрархії абстракцій. Ця модель складається з чотирьох видів діаграм:

- Діаграма контексту (Context diagram) - це загальний огляд архітектури, який показує взаємодію між програмною системою та її зовнішнім середовищем.
- Діаграма контейнерів (Container diagram) - це більш детальний огляд архітектури, який показує взаємодію між контейнерами, які є фізичними або логічними одиницями, що розміщують компоненти.
- Діаграма компонентів (Component diagram) - це ще більш детальний огляд архітектури, який показує взаємодію між компонентами, які є логічними одиницями, що реалізують певні функції.
- Діаграма коду (Code diagram) - це найдетальніший огляд архітектури, який показує взаємодію між елементами коду, які є найменшою одиницею архітектури.

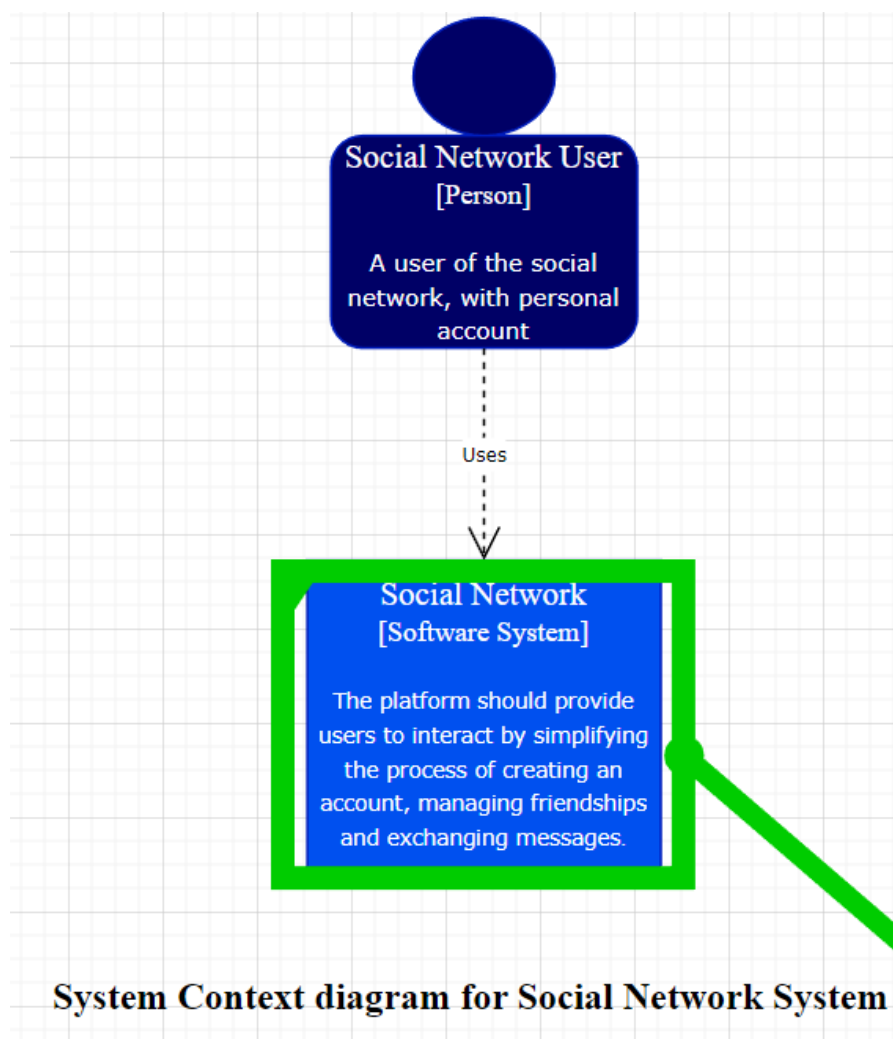


Рис.1 – Context diagram

У системі "Соціальна мережа" існує тип користувача "Користувач", який має можливість зареєструватись у системі та використовувати її для спілкування та обміну повідомленнями з іншими користувачами. Передбачено можливість створення діалогів та обмін повідомленнями між користувачами у межах соціальної мережі.

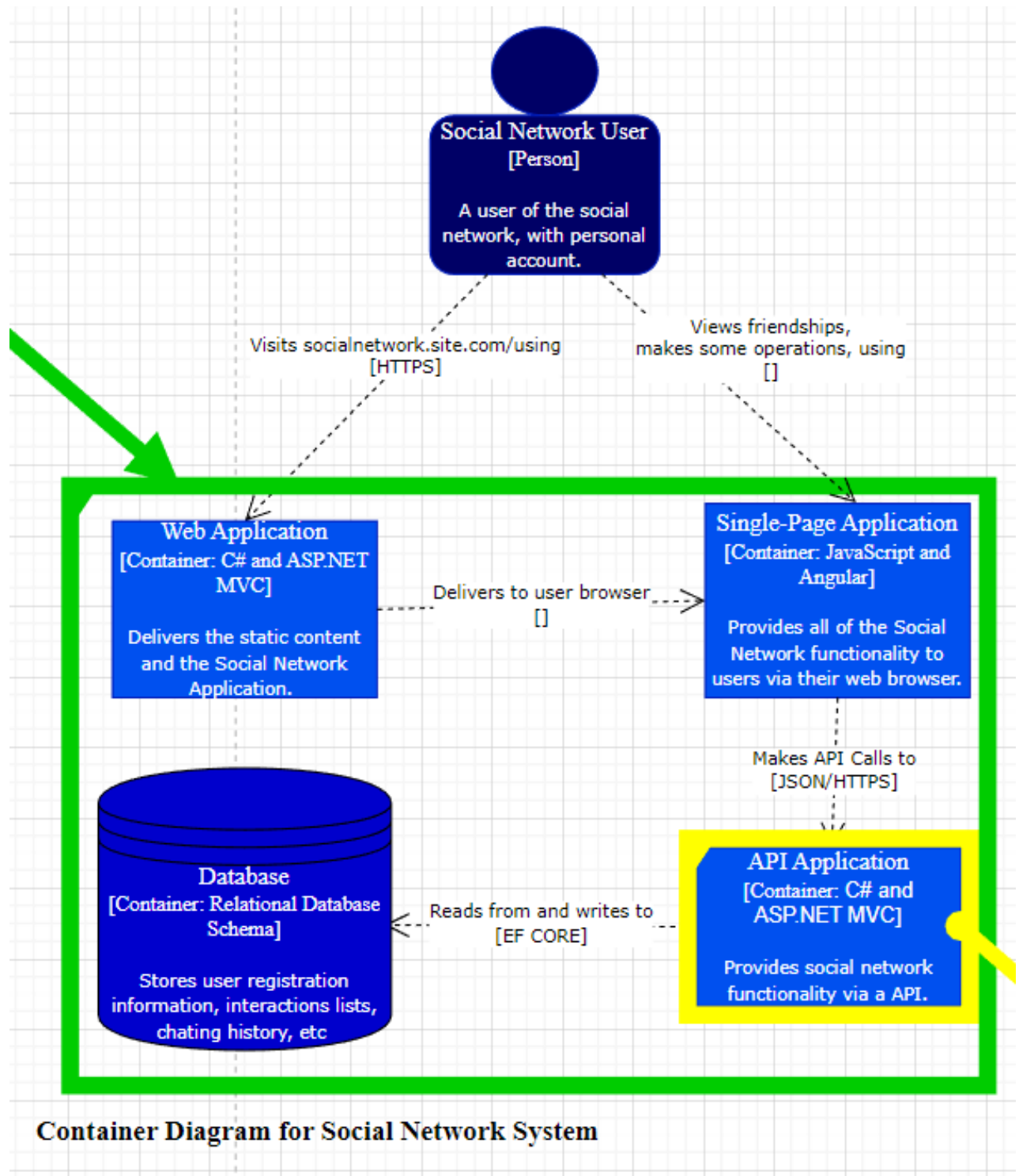


Рис.2 – Container diagram

Щоб отримати більш детальний огляд архітектури, я розбиваю кожен контейнер на його складові частини. Я визначаю, які компоненти містяться в кожному контейнері, що вони роблять і як вони взаємодіють один з одним. Діаграма компонентів показує, як контейнери складаються з компонентів. Вона також показує, які технічні деталі є важливими для кожного компонента.

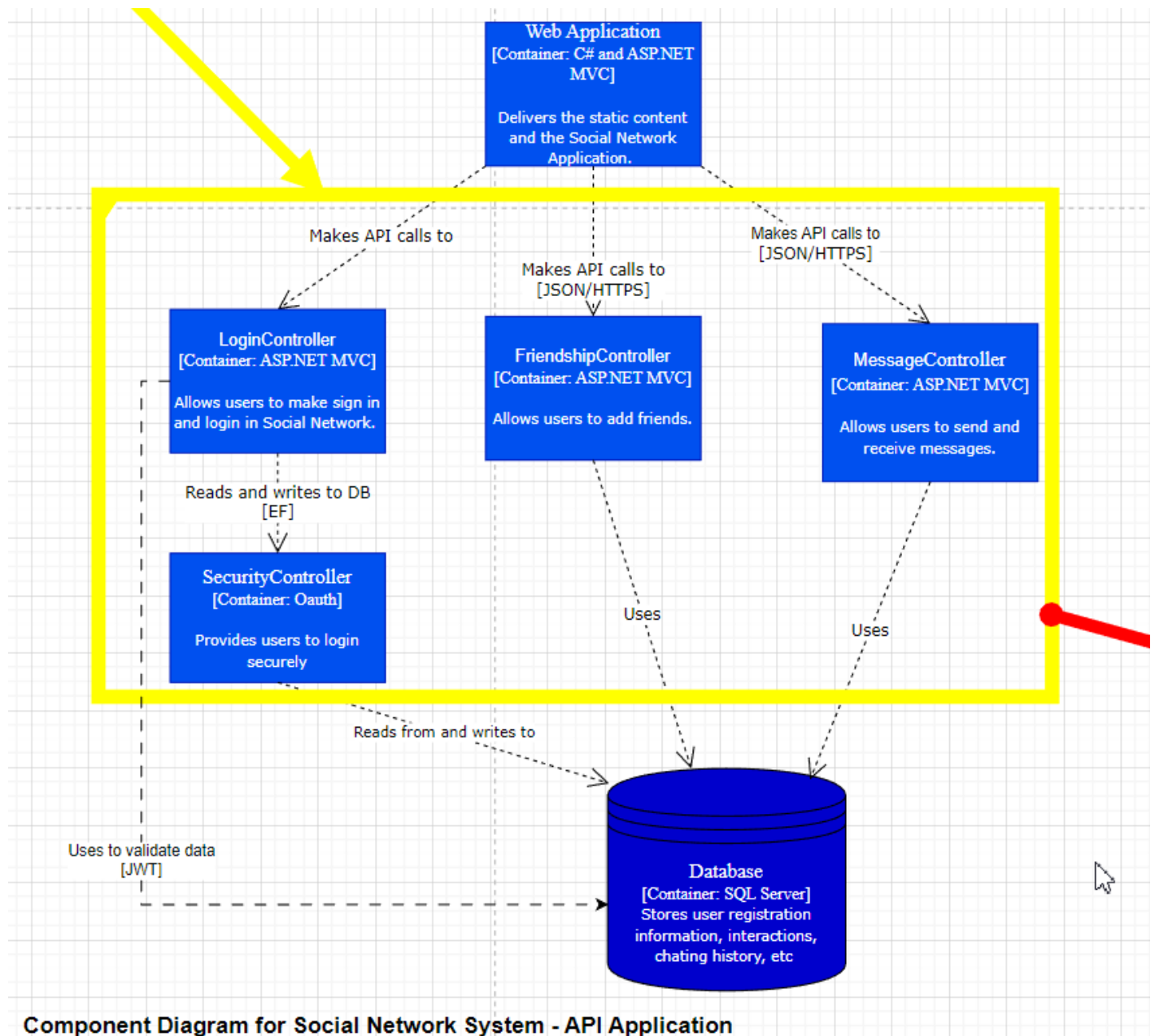


Рис.3 – Component diagram

Для розробки бекенду для застосунку буде використаний архітектурний шаблон MVC. Цей шаблон дозволяє розділити програму на три основні частини, які відповідають за різні завдання:

- **Модель(Model)** відповідає за збереження даних і обробку бізнес-логіки. У цьому випадку модель буде відповідати за зберігання інформації про бюджети та рахунки, а також за обчислення балансу.
- **Вид(View)** відповідає за відображення даних користувачеві. У цьому випадку вид буде відповідати за відображення інтерфейсу користувача для роботи з бюджетом та звітами.
- **Контролер(Controller)** відповідає за обробку вхідних запитів користувача та передачу їх моделі. У цьому випадку контролер буде відповідати за обробку запитів користувача для створення рахунків, додавання транзакцій, переказу коштів тощо.

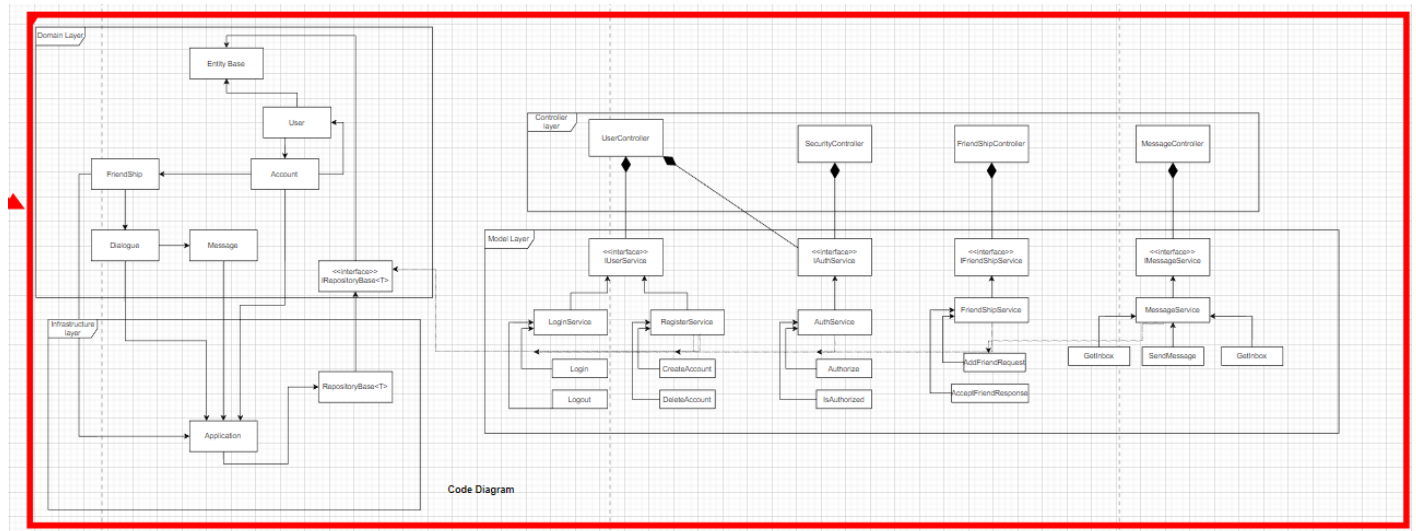


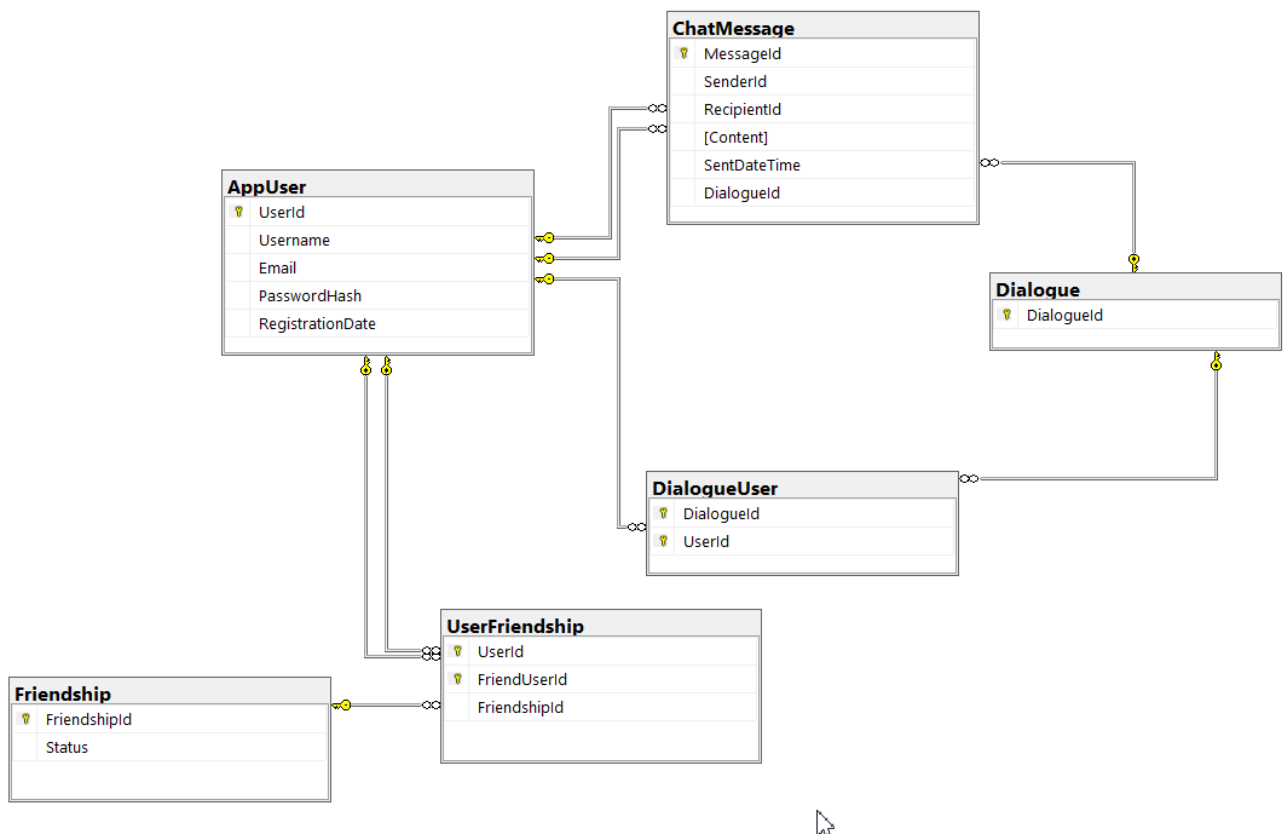
Рис.4 – Code diagram

1. ER-діаграма для DAL має наступні сутності:

- `AppUser`: Представляє користувача соціальної мережі.
- `Friendship`: Відображає дружбу між користувачами.
- `UserFriendship`: Таблиця-посередник для відношення багато-до-багатьох між користувачами та їх друзями.
- `Dialogue`: Представляє діалог між користувачами.
- `DialogueUser`: Таблиця-посередник для відношення багато-до-багатьох між діалогами та користувачами.
- `ChatMessage`: Зберігає інформацію про повідомлення між користувачами.

2. Зв'язки між сутностями:

- AppUser → Friendship: Користувач може мати багато друзів (Friendship).
- Friendship → UserFriendship: Таблиця-посередник для відношення багато-до-багатьох між користувачами та їх друзями.
- AppUser → DialogueUser: Користувач може брати участь в багатьох діалогах.
- Dialogue → DialogueUser: Таблиця-посередник для відношення багато-до-багатьох між діалогами та користувачами.
- AppUser → ChatMessage: Користувач може надсилати та отримувати багато повідомлень.



1. Таблиця "AppUser" (Користувач)

- UserId (INT): Унікальний ідентифікатор користувача.
- Username (VARCHAR(255)): Ім'я користувача.
- Email (VARCHAR(255)): Електронна пошта користувача.

- PasswordHash (VARCHAR(255)): Хеш пароля користувача.
- RegistrationDate (DATETIME): Дата реєстрації користувача.

2. Таблиця "Friendship" (Дружба)

- FriendshipId (INT): Унікальний ідентифікатор дружби.
- Status (VARCHAR(50)): Статус дружби (наприклад, "pending", "accepted", тощо).

3. Таблиця "UserFriendship" (Друзі_Користувач)

- UserId (INT): Зовнішній ключ для AppUser, ідентифікатор користувача.
- FriendUserId (INT): Зовнішній ключ для AppUser, ідентифікатор друга.
- FriendshipId (INT): Зовнішній ключ для Friendship, ідентифікатор дружби.

4. Таблиця "Dialogue" (Діалог)

- DialogueId (INT): Унікальний ідентифікатор діалогу.

5. Таблиця "DialogueUser" (Діалог_Користувач)

- DialogueId (INT): Зовнішній ключ для Dialogue, ідентифікатор діалогу.
- UserId (INT): Зовнішній ключ для AppUser, ідентифікатор користувача.

6. Таблиця "ChatMessage" (Повідомлення)

- MessageId (INT): Унікальний ідентифікатор повідомлення.
- SenderId (INT): Зовнішній ключ для AppUser, ідентифікатор відправника.
- RecipientId (INT): Зовнішній ключ для AppUser, ідентифікатор отримувача.
- Content (TEXT): Текст повідомлення.
- SentDateTime (DATETIME): Дата та час відправлення повідомлення.
- DialogueId (INT): Зовнішній ключ для Dialogue, ідентифікатор діалогу. Цей опис містить інформацію про кожне поле кожної таблиці разом із їхніми типами даних.