

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

Звіт

з лабораторної роботи № 3 з дисципліни
«Сучасні технології розробки WEB-застосувань на платформі
Microsoft.NET»

“Проектування REST веб-API”

Виконав: Щербацький Антон, ІП-14

Перевірив: Бардін В.

Київ 2023

Лабораторна робота №3

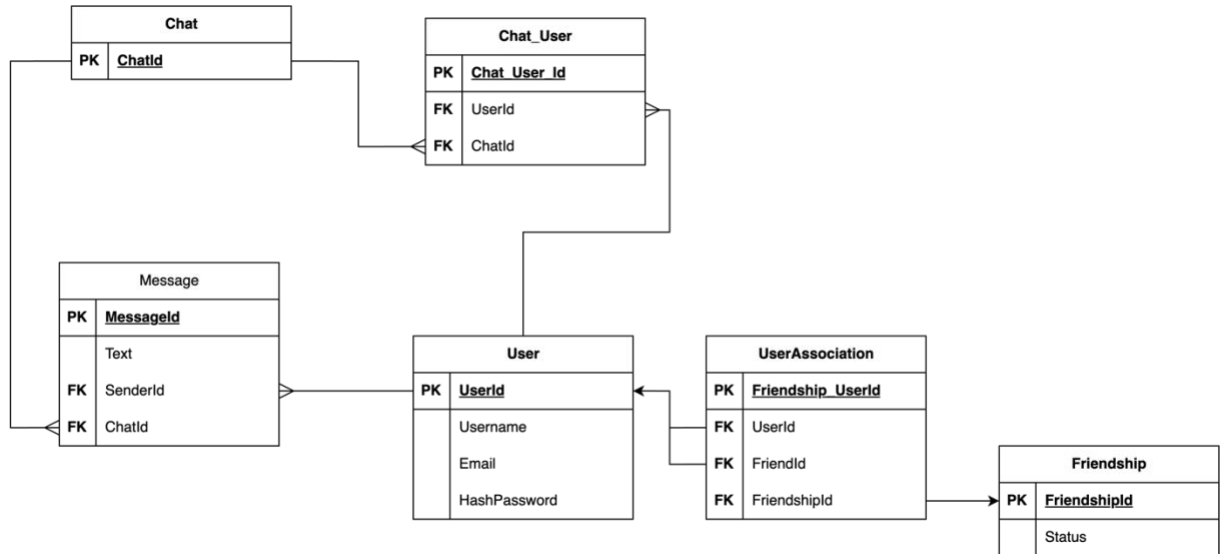
Завдання

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.

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>
---	---	--

ER diagram



Дана ER діаграма містить наступні сутності:

1. User – таблиця користувачів:
 - 1.1.UserId: int, унікальний ідентифікатор користувача
 - 1.2.Username: varchar, ім'я користувача у системі
 - 1.3.Email: varchar, електронна пошта користувача
 - 1.4.HashPassword: varchar, пароль користувача у хешованому вигляді
2. Friendship – таблиця статусів дружби:
 - 2.1.FriendshipId: int, унікальний ідентифікатор статусу дружби користувача
 - 2.2.Status: varchar, статус дружби користувача
3. UserAssociation – проміжна таблиця для зв'язування таблиці користувачів із статусами дружби:
 - 3.1.UserId: int, ідентифікатор користувача
 - 3.2.FriendId: int, ідентифікатор друга користувача

3.3.FriendshipId: int, ідентифікатор статусу дружби користувача

4. Message: таблиця повідомлень:

4.1.Text: varchar, текст повідомлення

4.2.SenderId: int, ідентифікатор відправника повідомлення

4.3.ChatId: int, ідентифікатор чату у якому це повідомлення було відправлено

5. Chat - таблиця чатів:

5.1.ChatId: int, ідентифікатор чату

6. Chat_User – проміжна таблиця для зв'язування таблиці User та Chat, оскільки відношення багато до багатьох:

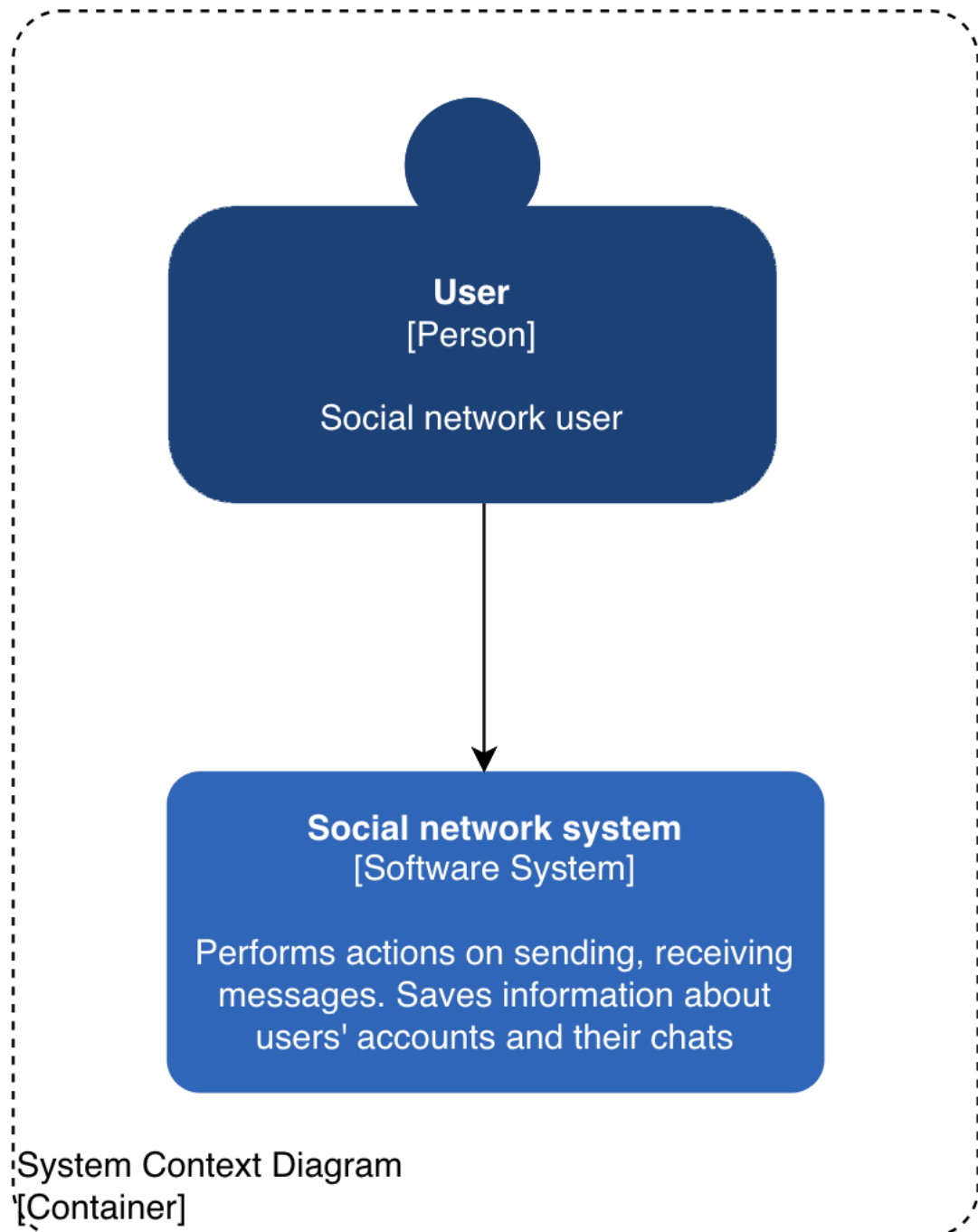
6.1.Chat_User_Id: int, ідентифікатор запису

6.2.UserId: int, ідентифікатор користувача

6.3.ChatId: int, ідентифікатор чату

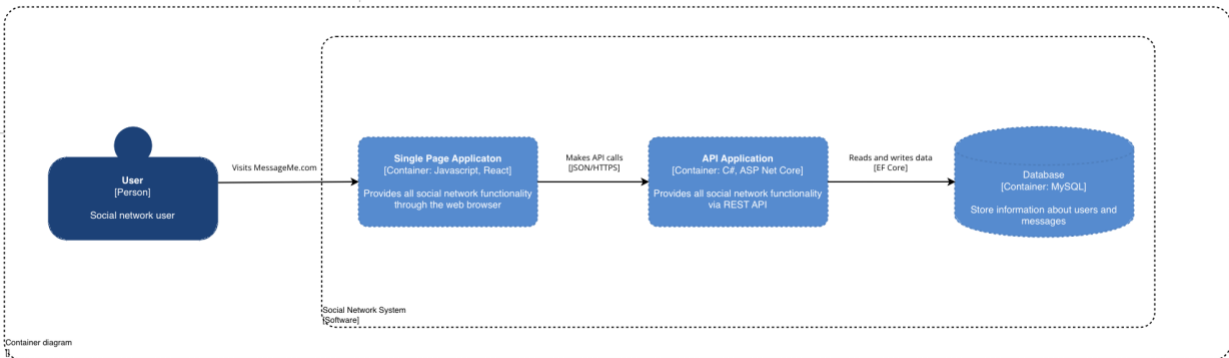
C4 Diagram

System context diagram



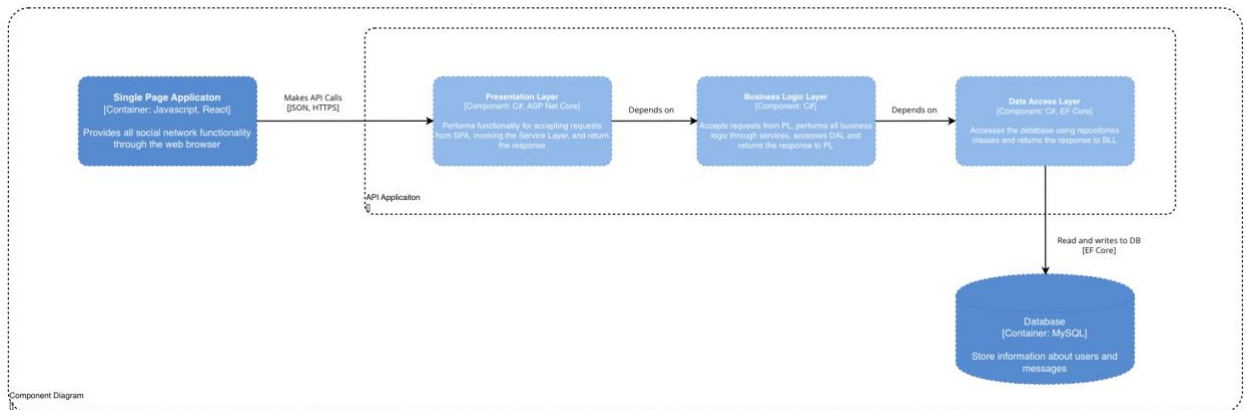
У системі наявний лише один тип користувача який може обмінюватися повідомленнями та додавати/видаляти друзів.

Container Diagram



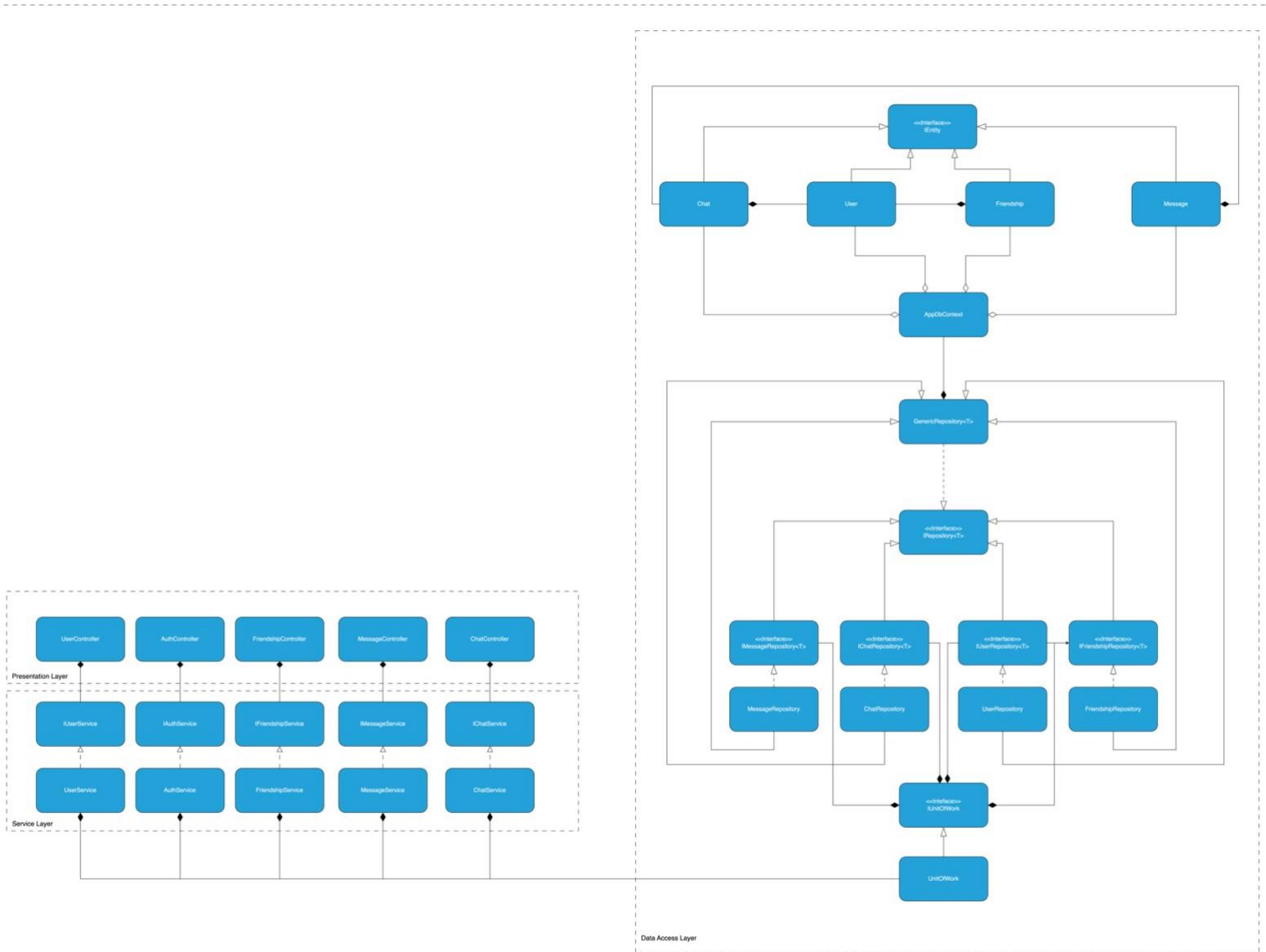
Container diagram у даній реалізації містить користувача системою та 3 наступні компоненти: веб-сайт – містить всю фронтенд логіку, через який користувач може робити запити до системи, надає графічну візуалізацію отриманих даних у браузері; компонент API – головний компонент системи, який містить бекенд логіку застосунку та надає дані у форматі JSON для зовнішніх запитів; компонент бази даних, який зберігає інформацію про користувачів та їх повідомлення у базі даних MySQL.

Component Diagram



Component Diagram у даному варіанті складається із 3 основних компонентів(рівнів архітектури): презентаційний рівень – містить контролери для отримання та відправки запитів клієнту, рівень бізнес логіки – містить необхідні сервіси для керування користувачами та повідомленнями із всією бізнес логікою, рівень доступу до даних – містить необхідні класи репозиторії для зручного доступу до бази даних.

Code diagram



Діаграма коду містить вище описані 3 рівні архітектури. На рівні Presentation Layer знаходяться контролери запитів, на рівні Service Layer знаходяться необхідні сервіси для проведення всієї бекенд бізнес-логіки, на рівні Data Access Layer знаходяться класи репозиторії, які були спроектовані використовуючи патерни Repository та Unit Of Work.

API endpoints

1. GET

1.1. URL: /api/users/{filtrationModel}

Description: отримати користувача по ID

Параметри: filtrationModel – модель фільтрації запиту, може містити ID, Email, Username користувача.

1.2. URL: /api/users/{id}/friends

Description: отримати список друзів для користувача, визначеного по ID

Параметри: id – ідентифікатор користувача

1.3. URL: /api/messages/user/{id}

Description: отримати всі повідомлення для користувача, визначеного по ID

Параметри: id – ідентифікатор користувача

1.4. URL: /api/messages/chat/{id}

Description: отримати список всіх повідомлень по ID чату

Параметри: id – ідентифікатор чату

2. POST

2.1. URL: /api/users/{userModel}

Description: додавання користувача у систему

Параметри: userModel – модель користувача, яка містить Username, Email користувача

2.2. URL: /api/chats/{id}/{messageModel}

Description: додавання повідомлення у певний чат визначений за ID

Параметри: id – унікальний ідентифікатор чату, messageModel – модель повідомлення, містить відправника та текст повідомлення

2.3. URL: api/friendship/add/{id}/{friendId}

Description: Додавання друга до користувача

Параметри: id – ідентифікатор користувача, friendId – ідентифікатор друга користувача

2.4. URL: api/friendship/request/{senderId}/{receiverId}

Description: Запрошення до мережі друзів певного користувача

Параметри: senderId – ідентифікатор відправника запрошення, receiverId – ідентифікатор отримувача запрошення

3. DELETE

3.1. URL: /api/users/{id}

Description: видалення користувача за ID

Параметри: id – унікальний ідентифікатор користувача

3.2. URL: /api/friendship/{id}/{friendId}

Description: видалення друга користувача за ID

Параметри: id – унікальний ідентифікатор користувача, friendId – ідентифікатор друга користувача

4. PATCH

4.1. URL: /api/users/{id}/{userModel}

Description: оновлення даних користувача

Параметри: userModel - модель користувача, яка містить Username, Email користувача

Висновок

На даній лабораторній роботі були спроектовані наступні діаграми для реалізації соціальної мережі: context diagram, container diagram, component diagram, code diagram.

Для визначення інтерфейсу API були побудовані API Endpoints, які підтримують запити GET, POST, PATCH, DELETE.