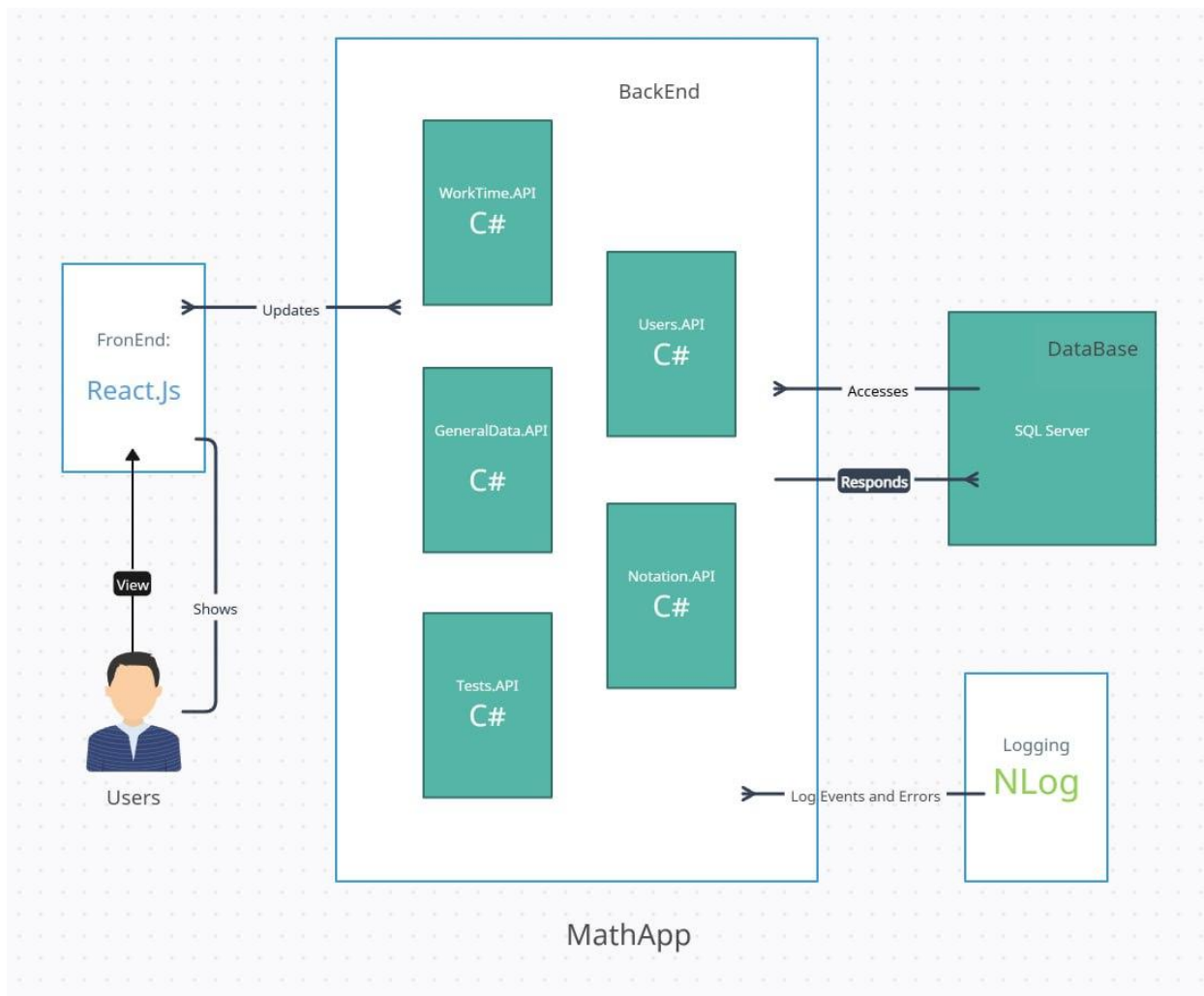


Архітектура

Як архітектуру для проєкту MathTouch обрано мікросервісну архітектуру, оскільки застосунок містить декілька незалежних сервісів, кожен з яких відповідає за окрему функціональність. Це дозволяє легко масштабувати систему, розгортати сервіси окремо та спрощує підтримку й оновлення застосунку.



Компоненти:

Frontend (React.js)

- Відображає інтерфейс користувача
- Отримує дані з бекенду та оновлює свій стан
- Передає оновлення користувача до відповідних API

Backend (C# .NET, набір мікросервісів)

- WorkTime API – керує обліком робочого часу користувачів
- Users API – відповідає за реєстрацію, автентифікацію та управління користувачами
- GeneralData API – обробляє загальні дані застосунку
- Notation API – відповідає за роботу з записами користувачів
- Tests API – керує тестами та їх перевіркою
- Усі сервіси працюють незалежно та взаємодіють через HTTP-запити

Database (SQL Server)

- Зберігає всі дані користувачів, результати тестів, записи робочого часу та інші важливі дані
- База даних напряму взаємодіє тільки з відповідними API

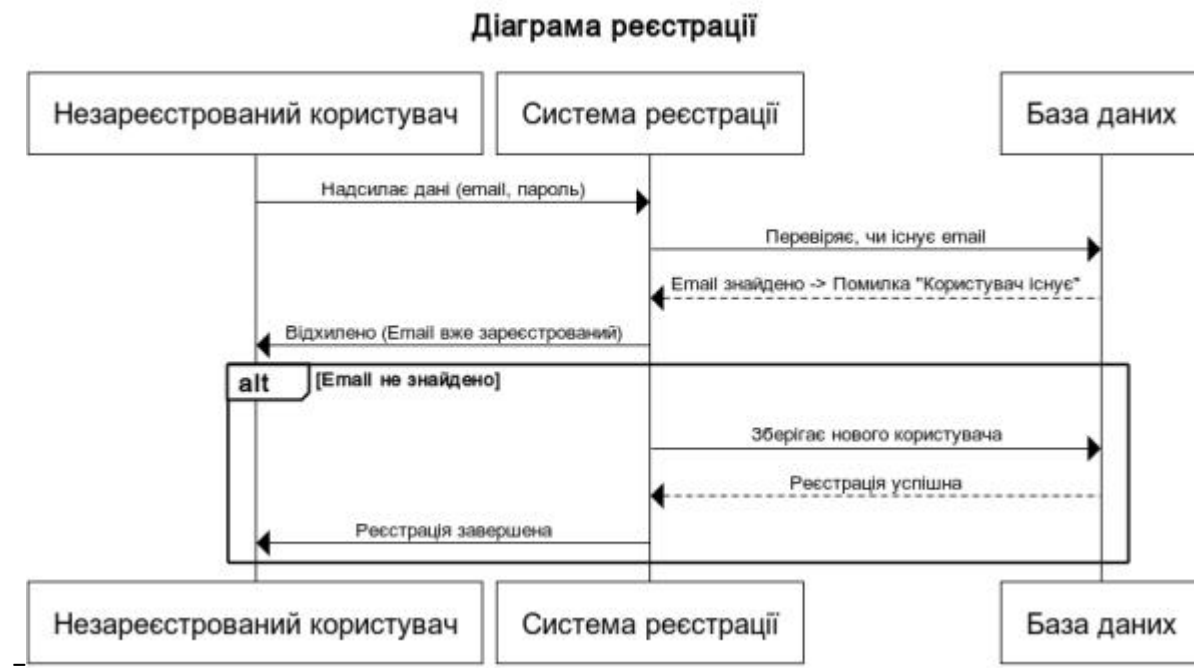
Logging (NLog)

- Відповідає за логування подій та помилок у системі
- Дозволяє відстежувати виконання сервісів та потенційні проблеми

Діаграми послідовностей

1. Діаграма реєстрації

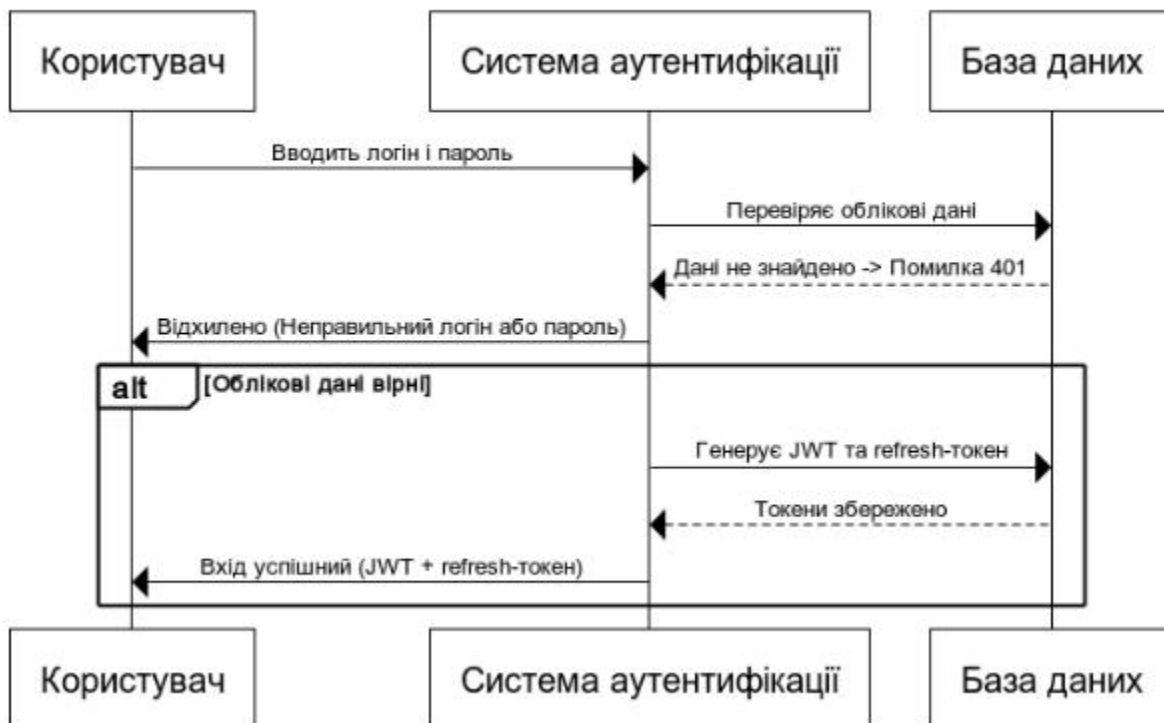
Ця діаграма демонструє процес реєстрації нового користувача. Спочатку система перевіряє, чи email вже існує в базі. Якщо він знайдений, користувач отримує помилку. Якщо email новий, дані зберігаються в базі, і реєстрація успішно завершується.



2. Аутентифікація (вхід у систему)

Ця діаграма описує процес входу користувача. Користувач вводить логін та пароль. Якщо дані невірні, система повертає помилку 401. Якщо все правильно, система створює JWT та refresh-токен, які будуть використовуватися для автентифікації.

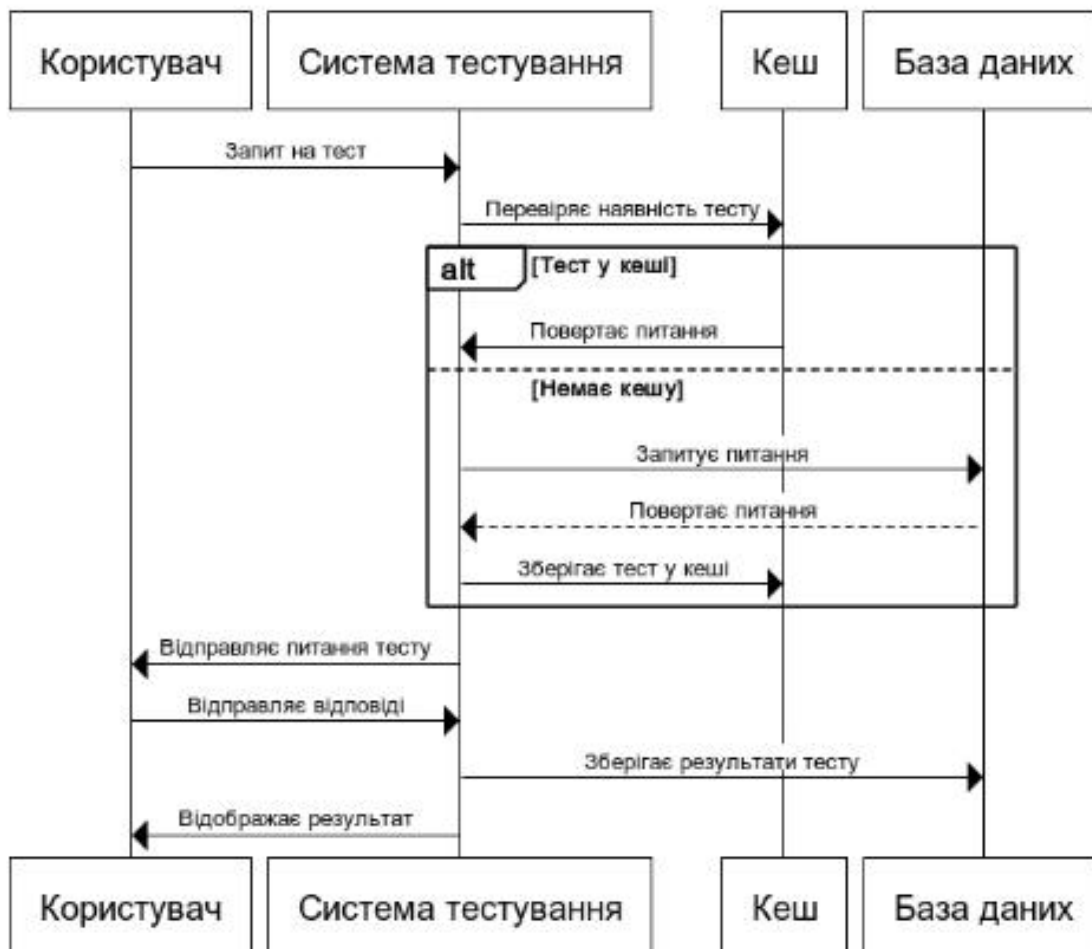
Діаграма входу в систему



3. Проходження тесту

Користувач запитує тест. Система спочатку перевіряє, чи є питання в кеші, щоб зменшити навантаження на базу. Якщо кешу немає, питання беруться з бази, а потім записуються в кеш. Відповіді користувача зберігаються у базі.

Діаграма проходження тесту



4. Отримання статистики

Вчитель запитує статистику успішності класу. Система надсилає запит у базу, отримує агреговані дані та повертає їх вчителю у вигляді звіту.

Діаграма отримання статистики



5. Надсилання сповіщень через Telegram-бот

Система отримує список користувачів, які повинні отримати повідомлення. Щоб уникнути перевантаження, повідомлення додаються в чергу. Потім система відправляє їх через Telegram API. Якщо Telegram API не відповідає, відправка повторюється через 10 секунд.

Діаграма надсилання сповіщень

