Concurency Patterns usage

1. Обробка кількох запитів одночасно (Thread Pool)

Учні одночасно запускають тест. Сервер обробляє їхні запити через пул потоків, що дозволяє уникнути перевантаження.

Учень1 Система тестування Учень2 Учень3 База даних Запит на тест Запит на тест Запит на тест Запуск у потоці (Thread Pool) Отримати питання Повертає питання Відправляє питання Відправляє питання Відправляє питання Учень1 Система тестування Учень2 Учень3 База даних

Thread Pool: Одночасне проходження тесту

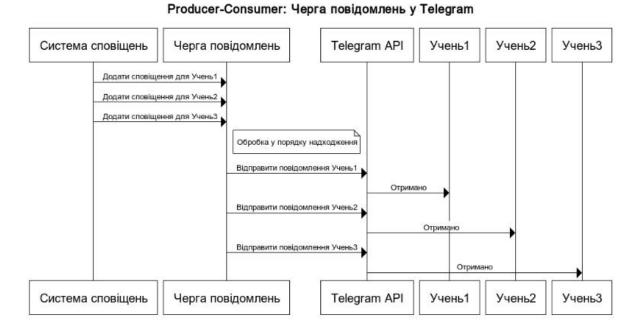
2. Асинхронне оновлення статистики (Future/Promise)

Учень завершує тест, а оновлення статистики запускається у фоновому режимі. Це дозволяє швидко повертати результат тесту, не чекаючи обробки статистики

Future/Promise: Асинхронне оновлення статистики

3. Черга повідомлень у Telegram-боті (Producer-Consumer)

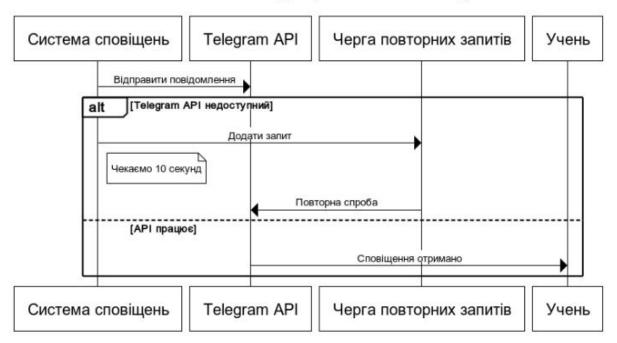
Щоб не перевантажувати Telegram API, повідомлення додаються у чергу і відправляються поступово.



4. Захист від перевантаження Telegram API (Circuit Breaker)

Якщо Telegram API не працює, система перериває запити і повторює спробу через певний час, щоб уникнути перевантаження.

Circuit Breaker: Захист від перевантаження Telegram API



5. Асинхронна обробка домашніх завдань (Asynchronous Messaging)

Коли вчитель призначає завдання, воно відправляється учням асинхронно, щоб не блокувати інші операції.

Учень1 Учень2 Учень3 Вчитель Система управління завданнями Черга повідомлень База даних Призначити завдання Додати завдання для учнів Обробка у фоновому режим Отримати завдання Учень3 Учень1 Вчитель Система управління завданнями Черга повідомлень Учень2 База даних

Asynchronous Messaging: Призначення домашнього завдання