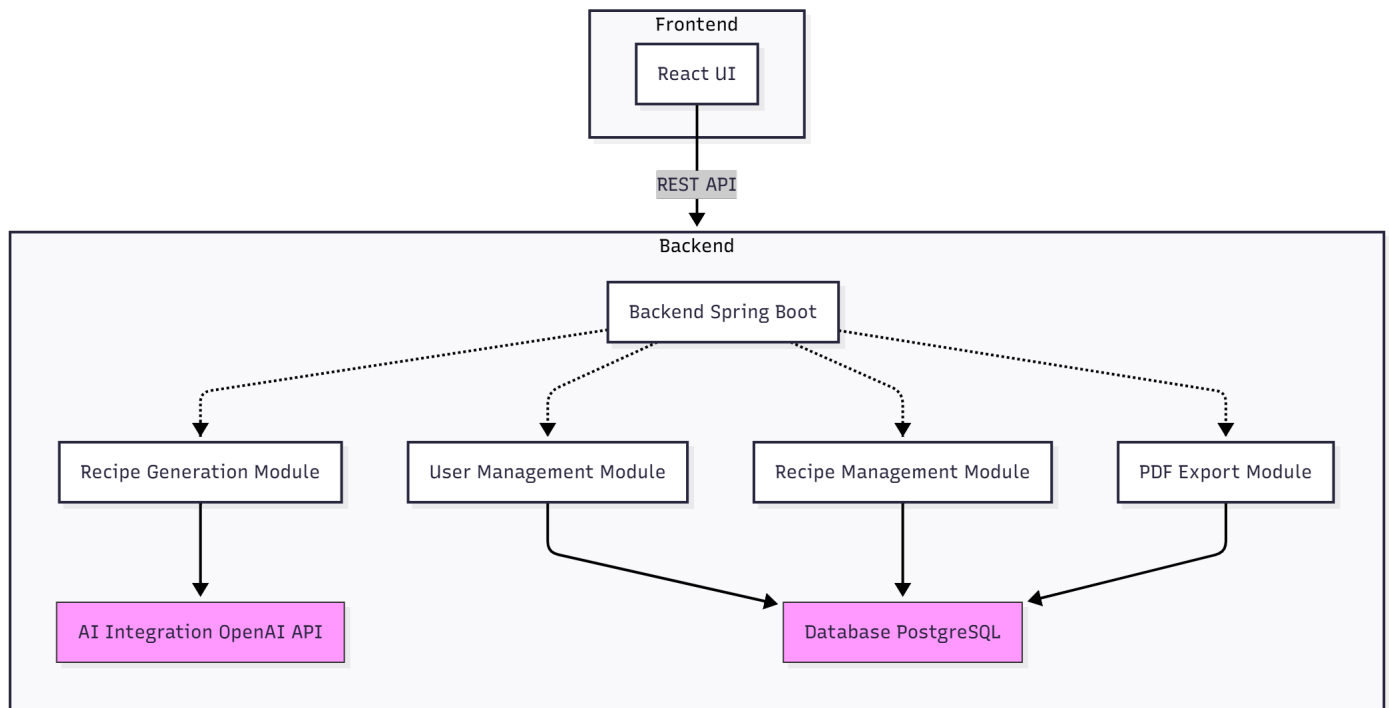


Архітектура

Для системи "**Personalized Recipe Generator**" обрана **монолітна архітектура** на базі **Spring Boot**, оскільки вона найбільш оптимально відповідає вимогам проекту щодо продуктивності, інтеграції та підтримки цілісності даних. Оскільки більшість функцій системи взаємопов'язані (користувач ↔ рецепти ↔ історія ↔ AI-генерація), моноліт дозволяє швидко та ефективно передавати дані між модулями без міжсервісних викликів і додаткових мережевих затримок.



Компоненти

Frontend (React)

- Відповідає за інтерфейс користувача.
- Надає можливість вводити інгредієнти, переглядати та фільтрувати рецепти, додавати улюблені, переглядати історію.
- Взаємодіє з бекендом через REST API.

Backend (Spring Boot)

- Інкапсулює всю бізнес-логіку системи.
- Обробляє запити з фронтенду та взаємодіє з базою даних і OpenAI API.
- Основні модулі:
 - o **Recipe Generation Module**

- Інкапсулює всю логіку взаємодії з OpenAI API.
 - Формує запити (prompts) до OpenAI API на основі введених інгредієнтів, дієтичних фільтрів та історії користувача.
 - Отримує структуровані рецепти від ШІ та передає їх фронтенду.
- o **User Management Module**
 - Відповідає за реєстрацію, аутентифікацію (включаючи OAuth2 з Google) та авторизацію.
 - Управління профілями користувачів, їхніми дієтичними обмеженнями та налаштуваннями.
 - o **Recipe Management Module**
 - Реалізує CRUD-операції з рецептами (створеними ШІ та доданими адміністратором).
 - Керує колекціями "Улюблених" рецептів, історією генерацій та коментарями.
 - o **PDF Export Module**
 - Відповідає за генерацію рецептів у формат PDF.

AI Integration (OpenAI API)

- Генерує персоналізовані рецепти на основі введених інгредієнтів і дієтичних фільтрів.
- Spring Boot формує запит, отримує текст від ШІ та повертає структурований рецепт.

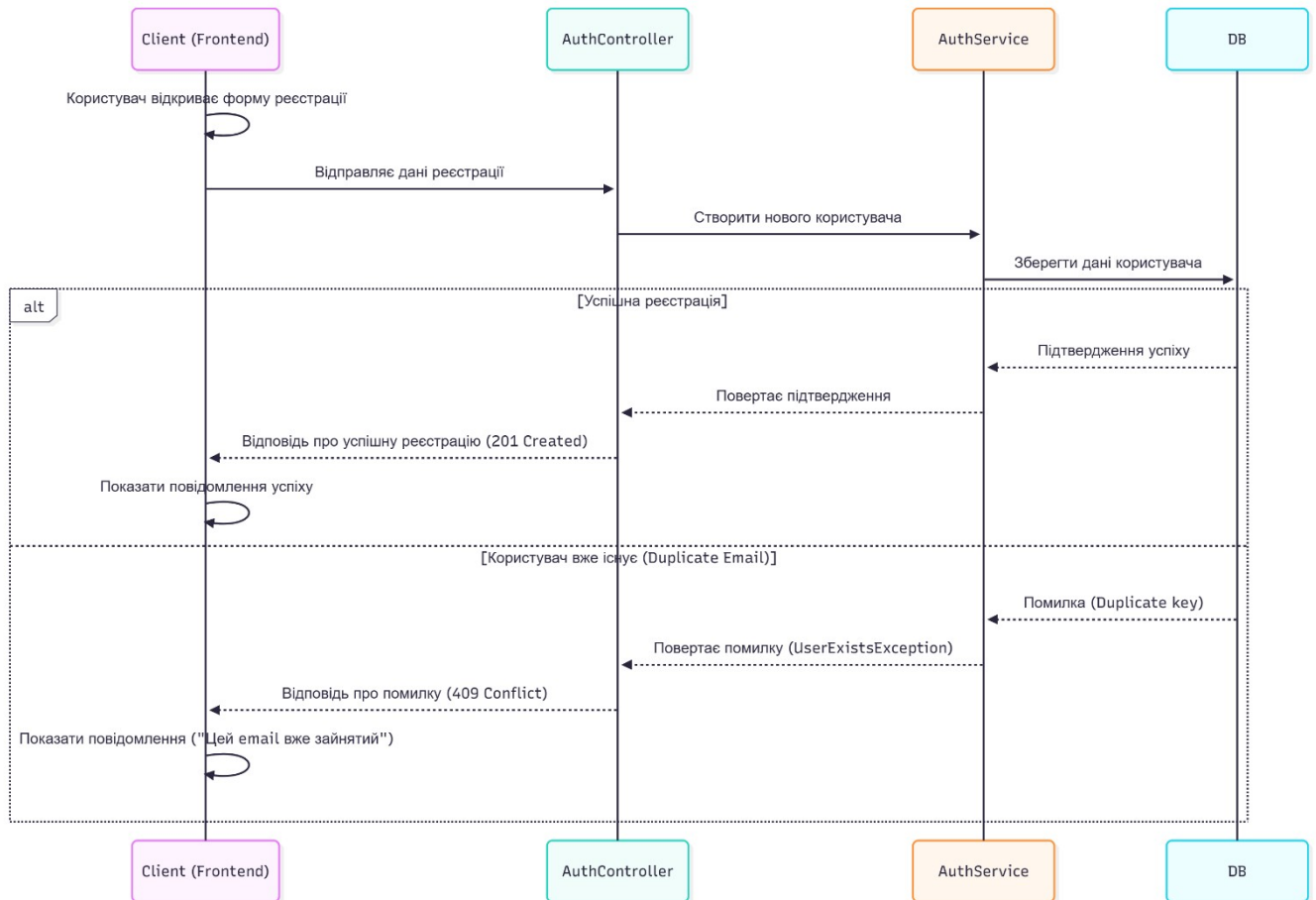
Database (PostgreSQL)

- Зберігає всі дані користувачів, рецепти, улюблені, історію генерацій, нотатки.
- Взаємодіє безпосередньо лише з бекендом через Spring Data JPA.

Діаграми послідовностей

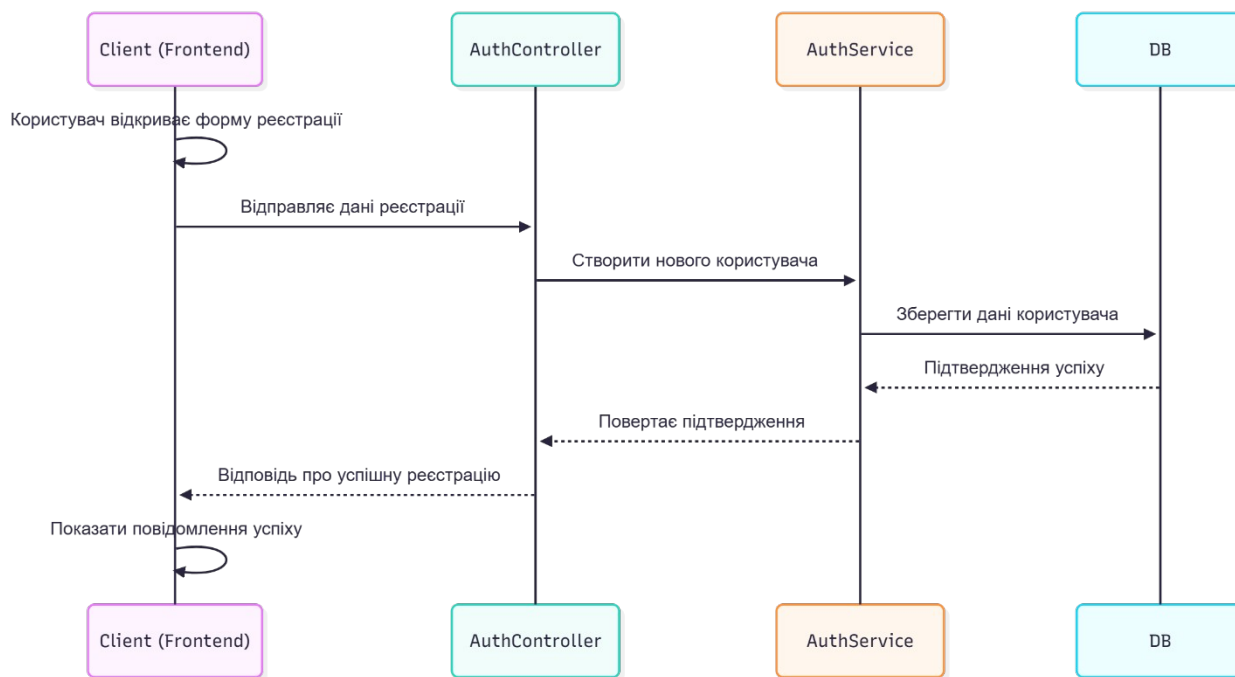
1. Діаграма реєстрації

Під час реєстрації Client надсилає дані на AuthController. AuthService намагається зберегти нового користувача в DB. Якщо email унікальний, користувач успішно зберігається, і Client отримує відповідь 201 Created. Однак, якщо такий email вже існує, DB повертає помилку, і Client отримує 400.



2. Автентифікація

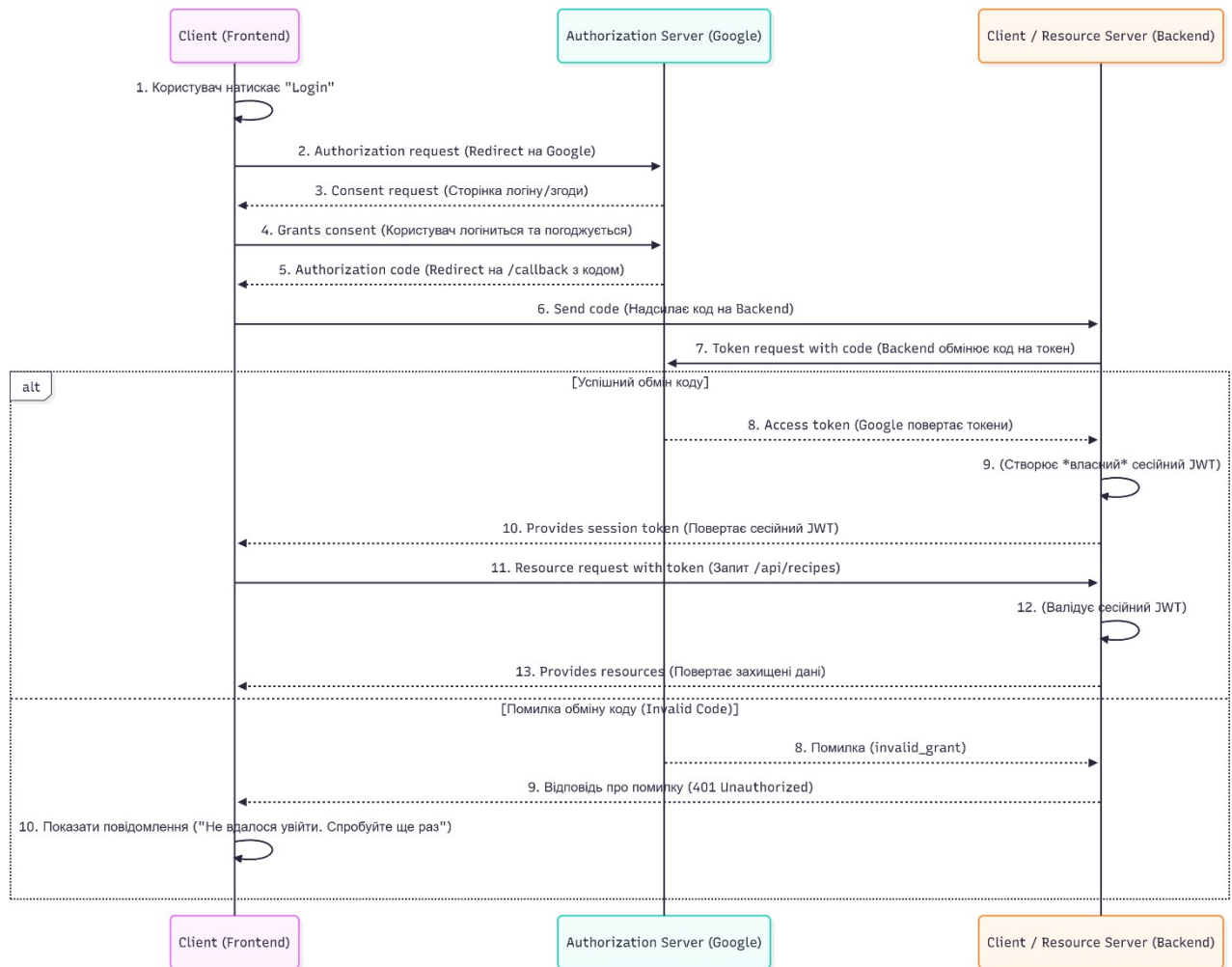
Client надсилає дані реєстрації на AuthController. Контролер передає їх AuthService, який зберігає користувача в DB. DB повертає успіх AuthService, той повертає підтвердження AuthController, і AuthController надсилає фінальну відповідь про успіх на Client.



OAuth2.0:

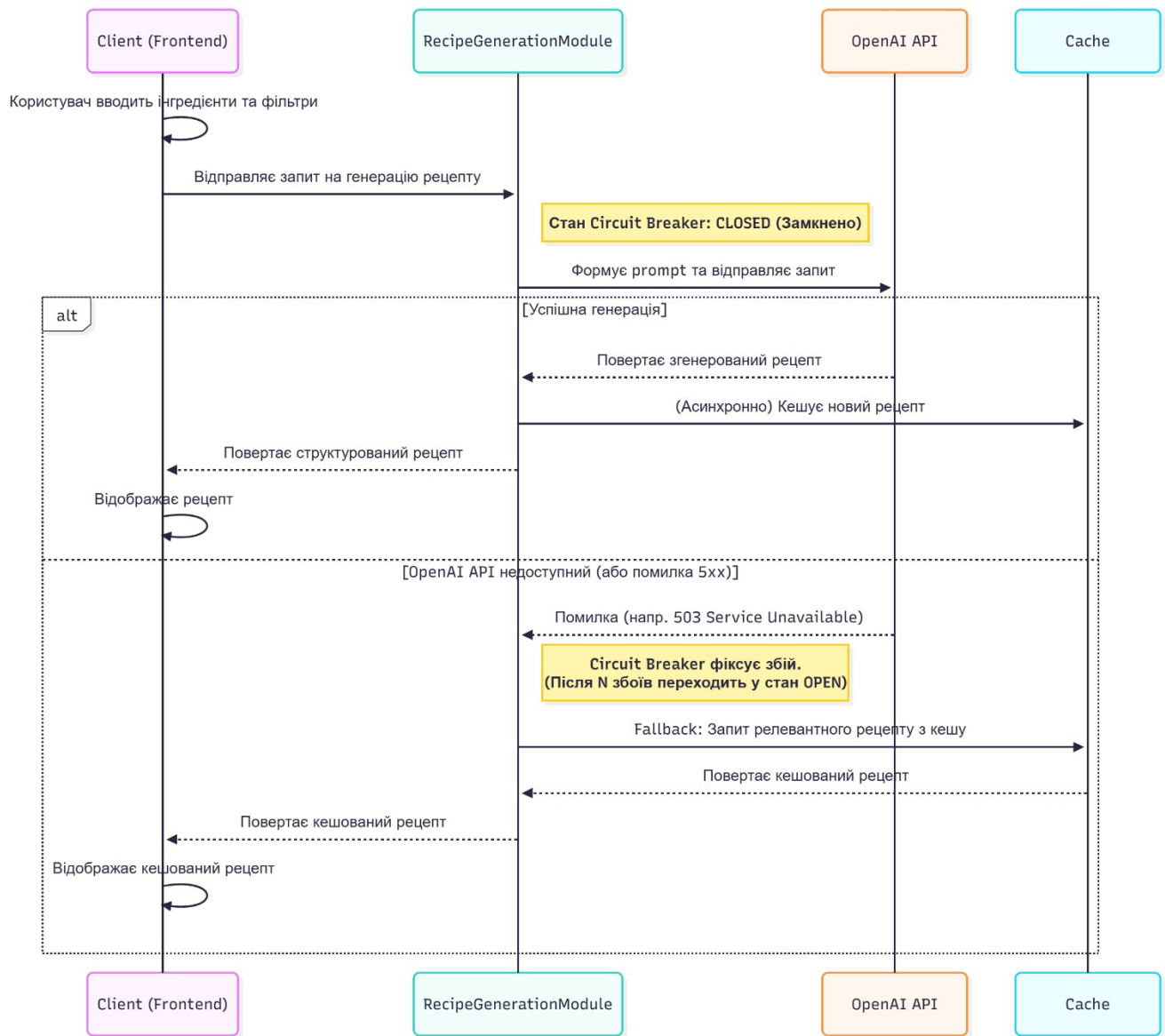
Цей потік починається, коли Client ініціює логін, перенаправляючи користувача на Authorization Server (Google). Після того, як користувач логінується і дає згоду, Google повертає його на Client з Authorization code. Client негайно надсилає цей code на Resource Server (Backend).

Далі потік розгалужується: в успішному випадку Backend обмінює code у Google на Access token, створює власний сесійний JWT і повертає його Client. Client використовує цей JWT для подальших запитів до захищених ресурсів (наприклад, /api/recipes), Backend валідує JWT і повертає дані. В альтернативному сценарії, якщо Backend не може обміняти code і отримує від Google помилку (invalid_grant), він повертає Client помилку 401 Unauthorized, і той показує повідомлення про невдалий вхід.



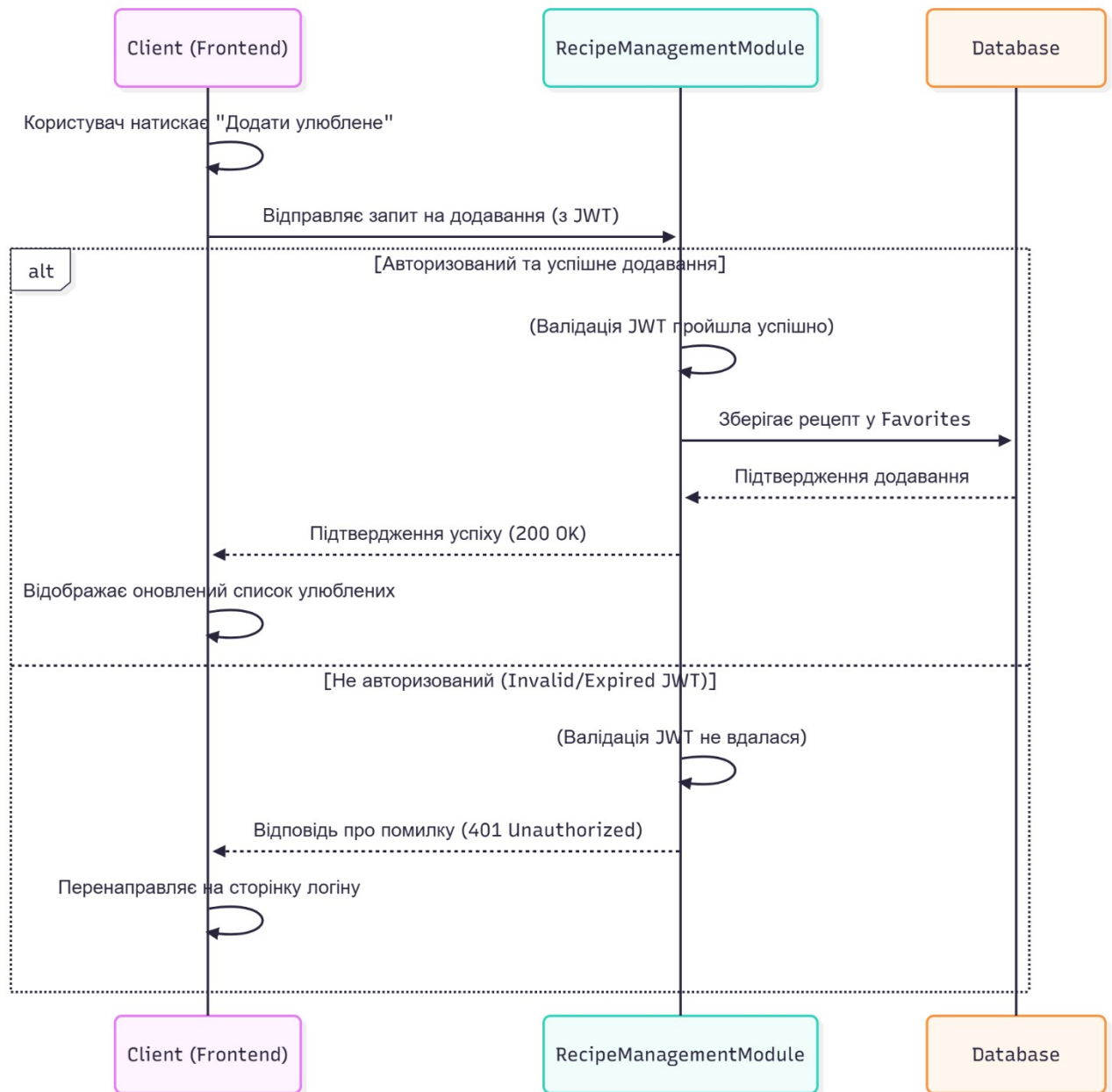
3. Створення рецепту з використанням OpenAPI

Для створення рецепту Client надсилає інгредієнти до RecipeGenerationModule. Цей модуль намагається викликати OpenAI API. Якщо OpenAI API успішно генерує та повертає рецепт, модуль структурує його і віддає Client. Якщо OpenAI API недоступний, спрацьовує Circuit Breaker, і модуль активує fallback-механізм, дістаючи релевантний рецепт з кешу і повертаючи його Client.



4. Додавання рецептів до "Улюблених"

Client надсилає запит із сесійним JWT до RecipeManagementModule. Якщо JWT валідний, модуль успішно зберігає дані в Database і повертає 200 OK. Якщо JWT відсутній або невалідний, модуль негайно повертає 401 Unauthorized, що змушує Client перенаправити користувача на сторінку логіну.



5. Експорт рецепту у форматі PDF

Client запитує PDF-файл у RecipeController. В успішному випадку PDFExportService генерує файл і повертає його для завантаження. Якщо ж PDFExportService стикається з помилкою під час генерації (наприклад, IOException), контролер повертає Client помилку 500 Internal Server Error.

