

ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СПІЛЬНОГО ВІДТВОРЕННЯ МЕДІАРЕСУРСІВ

Все більше аспектів соціально життя переходить в онлайн формат: планування та проведення зустрічей, побудова та контроль власної справи та, що не менш важливо, організація дозвілля. Враховуючи запропоновану велику кількість опцій щодо останнього, не усі користувацькі потреби задоволено.

Серед таких незайнятих ніш можна виокремити спільне відтворення відео та аудіо контенту, для чого ще не було представлено застосунку, який в повній мірі вирішує проблеми користувача та реалізує потенціал ідеї. Наявні на ринку пропозиції або першочергово спрямовані на вирішення іншої специфічної задачі і тому реалізують спільне відтворення медіаресурсів лише побічним чином, або виконують поставлену задачу лише частково.

Однією з центральних задач при розробці застосунку для відтворення медіаресурсів є інтеграція інформаційних джерел, з яких система буде відтворювати контент. Зважаючи на серйозну інфраструктурну логіку для гнучкого механізму інтеграції, насиченість предметної області та потенціальне розширення функціональності системи, з'являється вимога розділення відповідальності між частинами програми за видами логіки. Тому було вирішено будувати систему на основі багаторівневої архітектури із наступними шарами:

- шар представлення, який надає інтерфейс для взаємодії с програмою;
- шар прикладної логіки, який описує сценарії використання системи;
- шар предметної області, в якому сконцентровані поняття і правила предметної області;
- шар інфраструктури, в якому виконується взаємодія або використання зовнішніх залежностей.

Такий варіант розшарування системи чітко формує межі та відповідальність кожного із рівнів, завдяки чому отримано ізольовані шари предметної області та інфраструктури, що значно спрощує підтримку та розвиток системи у перспективі.

Через насиченість предметної області та наявність підобластей з пересічними поняттями рівень домену було спроектовано за допомогою основних принципів предметно-орієнтовного проєктування (Domain Driven Design [1]). Цей підхід допомагає контролювати складність предметної

області завдяки використанню спільної мови із бізнесом (Ubiquitous Language [1]), розділенню відповідного шару на контексти (Bounded Contexts [1]) та шаблонам їх взаємодії.

Предметну область розроблюваної системи було розділено наступним чином:

- контекст авторизації;
- контекст пошуку контенту;
- контекст спільного відтворення контенту;
- контекст інформаційного джерела (є представленням сторонніх систем, які надають медіаресурси).

Щодо прикладного рівня, було вирішено будувати його за базовими принципами розділення відповідальності «команд» та «запитів» (CQRS [2]). Основними перевагами цього підходу у порівнянні із використанням служб (сервісів) прикладного рівня є:

- більш структурований прикладний рівень, завдяки виділенню сценаріїв використання системи у відповідні модулі;
- можливість розділення моделей «запису» та «читання», яка дозволяє винести бізнес-логіку до частини, що модифікує стан застосунку, що значно сприяє спрощенню отримання даних і позитивно впливає на продуктивність системи.

Для першої версії продукту повноцінне розділення моделей «запису» та «читання» є надмірним, тому його не було впроваджено, але завдяки слідуванню базовим принципам CQRS це буде просто реалізувати у разі зміни рішення у майбутньому.

Із урахуванням вимог до технічного аспекту архітектури та компетенції розробника було обрано наступні технології для реалізації системи:

- .NET Core 3.1;
- C# 8 для серверної частини;
- HTML/CSS та JavaScript для клієнтської частини;
- MS SQL Server у якості бази даних та EF Core для взаємодії із нею.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Еванс, Е. Предметно-ориентированное проектирование (DDD) [Текст] : структуризация сложных программных систем. : Пер. с англ. / Е. Еванс.— М. : Вильямс, 2011. – 448 с.
2. CQRS [Електронний ресурс]. – Режим доступу : <https://martinfowler.com/bliki/CQRS.html>.