

УДК 004.42

Каленюк Д. – ст. гр. СП-42

Тернопільський національний технічний університет імені Івана Пулюя

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ АРІ ПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ З ВИКОРИСТАННЯМ СУЧАСНИХ ІТ-ТЕХНОЛОГІЙ

Науковий керівник: кандидат технічних наук, доцент Цуприк Г.Б.

Kaleniuk D.

Ternopil Ivan Puluj National Technical University

DESIGN AND IMPLEMENTATION OF AN API FOR A DISTANCE LEARNING PLATFORM USING MODERN IT TECHNOLOGIES

Supervisor: PhD, Associate Professor H. B. Tsupryk

Ключові слова: API, програмна архітектура, веб-сервіси.

Keywords: API, software architecture, web services.

Сучасні платформи дистанційного навчання є складними інформаційними системами, що забезпечують взаємодію між користувачами, навчальними ресурсами та сервісами оцінювання знань. Ключову роль у таких системах відіграють програмні інтерфейси прикладного програмування (API), які реалізують доступ до функціональних можливостей платформи та забезпечують інтеграцію з зовнішніми сервісами. Зі зростанням навантаження та вимог до безпеки виникає потреба у побудові масштабованої, гнучкої та захищеної архітектури.

У сучасних підходах до розробки API значна увага приділяється архітектурним патернам, що дозволяють розділити відповідальності між компонентами системи. Зокрема, Clean Architecture забезпечує незалежність бізнес-логіки від інфраструктурних деталей, що підвищує тестованість та підтримуваність системи [1]. Додатково використання патерну CQRS (Command Query Responsibility Segregation) дозволяє розділити операції читання та запису даних, оптимізуючи продуктивність та спрощуючи масштабування системи [2].

У рамках роботи запропоновано архітектуру API платформи дистанційного навчання, яка включає рівні представлення (Web API), прикладний (Application), доменний (Domain) та інфраструктурний (Infrastructure). Взаємодія між шарами здійснюється через інтерфейси та медіатор-патерн, що забезпечує слабке зв'язування компонентів. Для реалізації API використано REST-архітектурний стиль, який дозволяє стандартизувати обмін даними між клієнтами та сервером [3].

Практична реалізація виконана з використанням платформи ASP.NET Core, що забезпечує високу продуктивність і підтримку сучасних веб-технологій. Для доступу до даних застосовано Entity Framework Core разом із системою керування базами даних PostgreSQL. Для реалізації безпечної автентифікації та авторизації використано Keycloak, що підтримує протокол OAuth 2.0 та дозволяє централізовано керувати користувачами і ролями.

Важливою складовою розробки API є забезпечення якості програмного забезпечення шляхом тестування. Для перевірки коректності роботи системи застосовуються різні рівні тестування, зокрема модульне та інтеграційне. Модульні

тести дозволяють перевіряти окремі компоненти бізнес-логіки ізольовано від зовнішніх залежностей, тоді як інтеграційні тести забезпечують перевірку взаємодії між шарами системи та коректність роботи API-ендпоінтів. Використання принципів Clean Architecture та інверсії залежностей дозволяє ефективно застосовувати мок-об'єкти, що спрощує автоматизоване тестування без необхідності підключення реальної бази даних. Це підвищує швидкість виконання тестів і надійність програмного забезпечення.

Експериментальна перевірка показала, що застосування CQRS у поєднанні з багаторівневою архітектурою дозволяє підвищити продуктивність системи при обробці запитів та зменшити навантаження на базу даних. Використання зовнішнього сервісу ідентифікації забезпечує підвищений рівень безпеки та спрощує інтеграцію з іншими системами. Запропонована архітектура характеризується високою модульністю, що дозволяє легко розширювати функціонал платформи, зокрема шляхом додавання нових освітніх сервісів або аналітичних модулів.

Таким чином, розробка архітектури API платформи дистанційного навчання з використанням сучасних підходів, таких як Clean Architecture, CQRS та OAuth 2.0, дозволяє створити масштабоване, безпечне та ефективне програмне забезпечення. Подальші дослідження можуть бути спрямовані на впровадження мікросервісної архітектури, оптимізацію продуктивності та використання інтелектуальних алгоритмів для персоналізації навчального процесу.

Література:

1. Robert C. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston: Prentice Hall, 2017. 432 p.
2. Greg Young. Domain-Driven Design and Command Query Responsibility Segregation (CQRS). — Електронний ресурс. — Режим доступу: https://cqrs.files.wordpress.com/2010/11/cqrs_documents.pdf (дата звернення: 10.04.2026)
3. Fielding R. T. Architectural Styles and the Design of Network-Based Software Architectures: Doctoral dissertation. University of California, Irvine, 2000. 162 p.
4. Internet Engineering Task Force. OAuth 2.0 Authorization Framework (RFC 6749), 2012. 76 p.
5. Олянін, Д., Цуприк, Г. (2025) Transformer Neural Networks in Industry 4.0 / Д. Олянін, Г. Цуприк, Т. Говорущенко, О. Багрій-Заяць, І. Андрущак // Computer Information Technologies in Industry 4.0: proceedings of the 3rd International Workshop (CITI-2025), Ternopil, Ukraine, 11–12 June 2025. – Ternopil : Ternopil Ivan Puluj National Technical University, 2025 (Scopus) <https://ceur-ws.org/Vol-4057/>
6. Tsupryk, H., Olianin, D. (2025). Vydobuvannia danyh z tekstu vykorystovuiuchy transformerni neironni merezhi [Data extraction from text using Transformer Neural Networks]. Information Technology: Computer Science, Software Engineering and Cyber Security, 125–130, DOI: <https://doi.org/10.32782/IT/2025-2-13>
7. ОЛЯНІН Д., & ЦУПРИК Н. (2025). Огляд ролі трансформерних нейронних мереж у видобуванні інформації із неструктурованих даних. Measuring and computing devices in technological processes, 82(2), 360–364. <https://doi.org/10.31891/2219-9365-2025-82-52>