

УДК 004.415.2:004.051:37.018.43

ПРОЄКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО РІШЕННЯ ДЛЯ МОНІТОРИНГУ ТА ЛОГУВАННЯ LMS

Грибенко І.М., студент

Київський національний університет технологій та дизайну

Мительська О.В., кандидат технічних наук, доцент

Київський національний університет технологій та дизайну

Афтанділянц В.Є., кандидат економічних наук, доцент

Київський національний університет технологій та дизайну

Ключові слова: архітектура, програмне рішення, інформаційна система, LMS, моніторинг системи, логування подій.

У сучасних інформаційних системах, зокрема платформах управління навчанням (LMS), зростає потреба у побудові ефективних механізмів моніторингу, логування та аналізу подій [1-2]. Це обумовлено необхідністю забезпечення стабільності роботи системи, контролю виконання бізнес-процесів, а також оперативного виявлення помилок і аномалій. У даній роботі запропоновано архітектурний підхід [3] до побудови системи моніторингу та логування LMS на основі багатошарової моделі та принципів розділення відповідальності. На рис. 1 наведено архітектуру програмного рішення для моніторингу та логування LMS.

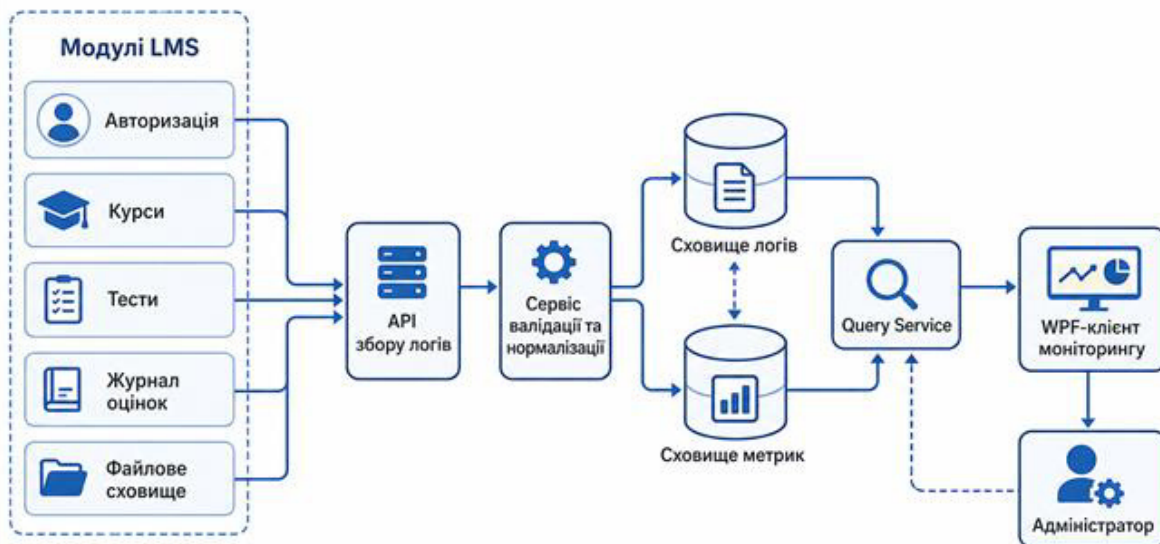


Рисунок 1 – Архітектура програмного рішення для моніторингу та логування LMS

Загальна архітектура рішення базується на поділі системи на чотири основні компоненти: джерела подій, серверний контур обробки, сховище даних і клієнтський застосунок. Джерелами подій виступають модулі LMS, такі як авторизація користувачів, управління курсами, тестування, журнал оцінок, файлове сховище та адміністративні операції. Кожен із цих модулів генерує телеметричні події, які містять інформацію про стан системи та дії користувачів.

Серверна частина виконує ключову роль у процесі обробки подій. Вона приймає вхідні повідомлення, перевіряє їх структуру на відповідність визначеному формату, виконує нормалізацію даних та зберігає їх у централізованому сховищі. Додатково сервер формує агреговані показники, які використовуються для аналітики та побудови звітів. Клієнтська частина системи забезпечує взаємодію користувача з даними моніторингу, включаючи перегляд стану системи, фільтрацію логів, візуалізацію у вигляді графіків та експорт даних у зовнішні формати.

Архітектурна модель системи поділяється на чотири логічні шари: Presentation, Application, Domain та Infrastructure. Такий підхід відповідає концепції багатошарової архітектури (Layered Architecture) і дозволяє чітко розмежувати відповідальність між компонентами системи. Основною перевагою даної моделі є зменшення зв'язності між модулями, підвищення рівня підтримуваності коду та можливість незалежного розвитку окремих підсистем.

Presentation-шар відповідає за взаємодію із користувачем і зовнішніми системами. Він включає API-контролери серверної частини та клієнтський WPF-інтерфейс оператора. На цьому рівні реалізується прийом HTTP-запитів, передача параметрів, базова валідація вхідних даних та відображення результатів у вигляді таблиць, графіків і журналів подій. Важливо, що Presentation-рівень не містить бізнес-логіки, виконуючи лише функцію адаптера між користувачем і внутрішніми сервісами системи. У випадку WPF-клієнта використовується патерн MVVM, що дозволяє розділити логіку представлення та керування станом інтерфейсу.

Application-шар реалізує прикладну логіку системи та координує виконання основних сценаріїв (use-case). Саме тут відбувається обробка телеметричних подій, агрегація даних, формування звітів та експорт інформації. Цей рівень не залежить від деталей зберігання або відображення даних, що забезпечує гнучкість архітектури. Наприклад, процес обробки події включає перевірку її структури, нормалізацію полів та передачу у репозиторій для подальшого збереження.

Domain-шар є ядром системи та містить основні доменні сутності: журнали подій, модулі LMS, рівні критичності, типи подій та метрики системи. У цьому шарі визначаються ключові бізнес-інваріанти, які

повинні залишатися незмінними за будь-яких сценаріїв виконання. Наприклад, кожна подія повинна містити часову мітку, ідентифікатор модуля, рівень критичності та контекст виконання. Такий підхід відповідає принципам Domain-Driven Design і забезпечує чітку формалізацію предметної області.

Infrastructure-шар відповідає за технічну реалізацію взаємодії із зовнішніми ресурсами. До його функцій належать робота з базою даних, реалізація репозиторіїв, файловий експорт, кешування та інтеграція з зовнішніми сервісами. У межах цього рівня реалізується збереження даних у PostgreSQL, серіалізація у форматі JSON та CSV, а також технічне логування роботи самої системи. Важливо, що Infrastructure залежить від технологічної реалізації, тоді як Domain та Application залишаються незалежними від конкретних інструментів.

Окрему увагу приділено принципу структурованості телеметричних даних. Кожна подія LMS подається у стандартизованому вигляді, що включає тип події, модуль системи, рівень критичності, часову мітку та контекст виконання. Це дозволяє перетворити журнали подій із неструктурованого тексту на структурований набір даних, придатний для аналітичної обробки. Завдяки цьому можливе виконання складних операцій фільтрації, агрегації та виявлення аномалій, зокрема аналіз частоти критичних помилок або оцінка стабільності окремих модулів LMS.

Таким чином, запропонована архітектура забезпечує високу масштабованість, гнучкість та підтримуваність системи моніторингу LMS. Поєднання багатошарової структури з принципами структурованого логування дозволяє створити ефективну платформу для збору, обробки та аналізу телеметричних даних у реальному часі.

Список використаних джерел

1. Du Plooy E., Casteleijn D., Franzsen D. Personalized adaptive learning in higher education: A scoping review of key characteristics and impact on academic performance and engagement. *Heliyon*. 2024. Vol. 10, № 21. DOI: 10.1016/j.heliyon.2024.e39630
2. Аксак Н., Татарников А. Огляд систем управління електронним навчанням. Традиційні та інноваційні підходи до наукових досліджень: матеріали IV Міжнар. наук. конф. Житомир, 2023. С. 131-136. DOI: 10.36074/mcnd-10.02.2023
3. Аксак Н. Г. Багатошарова мікросервісно-агентна архітектура системи персоналізованого навчання / Аксак Н. Г., Татарников А. О. // Системи та технології, № 2 (70), 2025. – с. 74-83. DOI:10.32782/2521-6643-2025-2-70.8