

УДК 004.7

*Залозний О.А., здобувач,
Савицький Р.С., аспірант, ст. викладач
Державний університет «Житомирська політехніка»*

ОРГАНІЗАЦІЯ ЗБЕРІГАННЯ ПОВІДОМЛЕНЬ, СПОВІЩЕНЬ ТА ДІЙ КОРИСТУВАЧІВ У ВЕБПЛАТФОРМІ АНІМЕ Й МАНГИ

Вебплатформи з активною взаємодією користувачів потребують ефективного зберігання різних типів комунікації та подій системи. Використання єдиної структури для повідомлень і системних сповіщень призводить до ускладнення обробки даних, зниження продуктивності та складності масштабування. Відсутність механізму фіксації дій користувачів ускладнює аналіз поведінки та контроль змін у системі. Додатково виникає проблема узгодженості даних при великій кількості одночасних запитів, що характерно для вебплатформ із високою активністю користувачів.

У вебзастосунках підсистеми комунікації реалізуються з урахуванням розділення типів взаємодії. Повідомлення між користувачами та системні сповіщення розглядаються як різні за призначенням сутності, що мають окремі структури зберігання та обробки. Повідомлення забезпечують двосторонню комунікацію та потребують збереження історії діалогів, тоді як сповіщення формуються системою та відображають події, пов'язані з активністю користувачів або змінами в системі.

У сучасних підходах до розробки вебзастосунків використовується розділення даних за функціональним призначенням та застосування ORM-інструментів для роботи з реляційними базами даних [1–3]. Також застосовується збереження історії дій користувачів у вигляді журналів подій, що дозволяє виконувати аналіз активності та забезпечувати контроль змін у системі.

Потрібно визначити підхід до організації зберігання повідомлень, сповіщень та історії дій користувачів у вебплатформі аніме й манги. Необхідно розмежувати типи комунікації, визначити структуру таблиць бази даних та забезпечити фіксацію активності користувачів із урахуванням можливості масштабування системи.

Повідомлення між користувачами мають двосторонній характер і використовуються для передачі текстової інформації. Вони містять ідентифікатори відправника та отримувача, текст повідомлення, часову мітку та статус прочитання. Зберігання повідомлень реалізується в

окремій таблиці, що забезпечує ведення історії переписки та ефективну обробку діалогів.

Системні сповіщення призначені для інформування користувачів про події платформи. Вони формуються автоматично та мають односторонній характер. Сповіщення містять тип події, ідентифікатор користувача, посилання на пов'язаний об'єкт та статус прочитання. Використання окремої таблиці дозволяє оптимізувати обробку подій та спростити реалізацію відповідної функціональності.

Розділення повідомлень і сповіщень на різні сутності дозволяє уникнути змішування різнорідних даних, зменшити складність запитів та забезпечити гнучкість архітектури системи.

Для фіксації активності користувачів використовується таблиця історії дій, яка містить тип дії, ідентифікатор користувача, пов'язаний об'єкт та часову мітку. Збереження історії дій дозволяє виконувати аналіз поведінки користувачів та забезпечувати контроль змін у системі.

При реалізації історії дій доцільно враховувати обсяг даних, що накопичуються. Для цього може застосовуватися архівація або обмеження глибини збереження історії. Також можливе використання окремих механізмів обробки подій, що дозволить зменшити навантаження на основну базу даних.

Реалізація здійснюється із використанням фреймворку Next.js, який забезпечує інтеграцію клієнтської та серверної частин застосунку [1]. Для взаємодії з базою даних використовується Prisma ORM, що забезпечує типобезпечні запити [2]. Зберігання даних реалізовано у PostgreSQL, що підтримує складні зв'язки між сутностями [3].

Розмежування повідомлень, сповіщень та історії дій користувачів забезпечує структуроване зберігання даних, спрощує обробку запитів та підвищує ефективність роботи вебплатформи.

Список використаних джерел:

1. Next.js Documentation. URL: <https://nextjs.org/docs> (дата звернення: 02.03.2026)
2. Prisma ORM Documentation. URL: <https://www.prisma.io/docs> (дата звернення: 02.03.2026)
3. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs> (дата звернення: 02.03.2026)