

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

**«Мультиплатформений програмний комплекс для
організації розважальних заходів»**

Виконав студент групи ІПЗ-22-2
МИКОЛАЙЧУК Вадим Юрійович

Керівник роботи:
ЛОКТИКОВА Тамара Миколаївна

Рецензент:
ЛОБАНЧИКОВА Надія Миколаївна

Житомир – 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ "ЖИТОМИРСЬКА ПОЛІТЕХНІКА"
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. каф. ПЗ, д.п.н., проф.

Тетяна ВАКАЛЮК

«28» лютого 2026 р.

ЗАВДАННЯ

на кваліфікаційну роботу

Здобувач вищої освіти: **МИКОЛАЙЧУК Вадим Юрійович**

Керівник роботи: **ЛОКТИКОВА Тамара Миколаївна**

Тема роботи: **«Мультиплатформений програмний комплекс для організації розважальних заходів»** , затверджена Наказом закладу вищої освіти від **«27» лютого 2026р., №92/с**

Вихідні дані для роботи: **об'єктом дослідження є процеси цифрової організації та адміністрування розважальних заходів у багатоклієнтському програмному середовищі, а предметом дослідження — методи і засоби розробки мультиплатформеного програмного комплексу для створення, управління та відвідування подій.**

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Локтікова Т.М	02.03.2026	02.03.2026
2	Локтікова Т.М	02.03.2026	02.03.2026
3	Локтікова Т.М	02.03.2026	02.03.2026

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебсайту, мобільного застосунку, серверної частини та пояснювальної записки. Пояснювальна записка до випускної кваліфікаційної роботи містить текстову частину, викладену на 95 сторінках друкованого тексту, 44 ілюстрацій, 34 таблиць. Список використаних джерел містить 31 найменування і займає 3 сторінки. Також у роботі є 20 сторінки додатків. Загальний обсяг роботи складає 115 сторінок.

Мета роботи — створити систему управління розважальними заходами (вебсайт, мобільний застосунок для Android та iOS, серверна підсистема) з можливостями створення й редагування подій, пошуку та фільтрації, реєстрації користувачів, участі в заходах, оплати квитків, гарантії організатора, просування подій, відгуків і статистики.

Застосовано методи об'єктно-орієнтованого проектування, клієнт–серверної архітектури та мультиплатформеної розробки; реалізовано REST API, інтеграцію платіжного сервісу та модулі аналітики.

Результатом роботи є програмний комплекс із серверною частиною, вебінтерфейсом і мобільним застосунком на Kotlin Multiplatform. Система забезпечує узгоджену роботу з даними та зручний доступ до функцій на різних пристроях.

КЛЮЧОВІ СЛОВА: ВЕБСАЙТ, ІВЕНТ-МЕНЕДЖМЕНТ, МОБІЛЬНИЙ ЗАСТОСУНОК, МУЛЬТИПЛАТФОРМЕНІСТЬ, ОПЛАТА КВИТКІВ, СТАТИСТИКА ЗАХОДІВ, УПРАВЛІННЯ ПОДІЯМИ, REST API.

					ІПЗ.КР.Б-121-26 - ПЗ			
Змн	Арк.	№ докум.	Підпис	Дата				
Розроб.		Миколайчук В.Ю.			Мультиплатформений комплекс для організації розважальних заходів	Літ.	Арк.	Аркуші
Перевір.		Локтікова Т.М.					3	115
						ФІКТ, ар. ІПЗ-22-2		
Н. Контр.		Тичина О.П.						
Зав.каф.		Вакалюк Т.А.						

ABSTRACT

The bachelor's graduation thesis consists of a website, a mobile application, a server component. The explanatory report contains a textual part presented on 95 pages of printed text, 44 illustrations, and 34 tables. The list of references includes 31 sources and occupies 3 pages. The thesis also includes 20 pages of appendices. The total volume of the work is 115 pages.

The object of the study is the processes of digital organization of entertainment events; the subject is the development of a multiplatform system with a single data source and synchronized information across clients.

The aim is to develop a system for managing entertainment events (website, Android/iOS mobile application, server subsystem) supporting event creation and editing, search and filtering, user registration, participation, ticket payment, an organizer wallet, event promotion, reviews, and statistics.

The work applies object-oriented design, client-server architecture, and multiplatform development; it implements a REST API, payment service integration, and analytics modules.

The result is a software system with a server component, web interface, and Kotlin Multiplatform mobile application that ensures consistent data handling and convenient access on different devices.

KEYWORDS: EVENT MANAGEMENT, EVENT STATISTICS, MOBILE APPLICATION, MULTIPLATFORMNESS, REST API, TICKET PAYMENT, WEBSITE.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ МУЛЬТИПЛАТФОРМЕННОГО ПРОГРАМНОГО КОМПЛЕКСУ	9
1.1 Постановка задачі.....	9
1.2 Аналіз аналогів програмного продукту.....	11
1.3 Вибір архітектури програмного комплексу	23
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.....	27
Висновки до розділу 1	30
РОЗДІЛ 2 ПРОЕКТУВАННЯ МУЛЬТИПЛАТФОРМЕННОГО ПРОГРАМНОГО КОМПЛЕКСУ	32
2.1 Визначення варіантів використання	32
2.2 Розробка бази даних	41
2.3 Розробка функціональних алгоритмів роботи програми	48
2.4 Реалізація мультиплатформеного комплексу	52
Висновки до розділу 2.....	58
РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З МУЛЬТИПЛАТФОРМЕНИМ КОМПЛЕКСОМ.....	60
3.1 Структура інтерфейсу мультиплатформеного комплексу	60
3.2 Тестування мультиплатформеного комплексу	83
Висновки до розділу 3.....	88
ВИСНОВКИ	90
ДОДАТКИ	95

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

МЗ – мобільний застосунок

ПЗ – програмне забезпечення

БД – база даних

API – програмний інтерфейс застосунку (Application Programming Interface)

Compose – декларативний UI-фреймворк Jetpack Compose

CRUD – базові операції роботи з даними (Create, Read, Update, Delete)

HTTPS – захищений протокол HTTP (HyperText Transfer Protocol Secure)

IDE – інтегроване середовище розробки (Integrated Development Environment)

JSON – формат обміну даними (JavaScript Object Notation)

KMP – Kotlin Multiplatform

Ktor – HTTP-клієнт для мобільного застосунку

MVVM – архітектурний шаблон (Model–View–ViewModel)

REST API – взаємодія мобільного клієнта та сервера через HTTP (Representational State Transfer Application Programming Interface)

UI – інтерфейс користувача (User Interface)

UX – досвід користувача (User Experience)

ViewModel – компонент управління станом екрана

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми зумовлена стрімким розвитком цифрових сервісів у сфері організації подій і зростанням очікувань користувачів щодо швидкого доступу до інформації про розважальні заходи. Більшість пошуку подій, реєстрації на них і взаємодії між організаторами та учасниками відбувається через Інтернет, причому значна частина запитів формується саме з мобільних пристроїв. Тому актуальним є створення інформаційних систем, які поєднують централізоване зберігання даних на сервері з зручними клієнтськими застосунками для різних платформ. Важливим аспектом є забезпечення високої швидкості роботи таких сервісів та зручного інтерфейсу, який дозволяє користувачам оперативно знаходити необхідні події без зайвих дій. Окрему роль відіграє підтримка персоналізації контенту та гнучких механізмів фільтрації, що підвищує релевантність інформації для кожного користувача. Це сприяє підвищенню ефективності організації заходів і загальному покращенню взаємодії між усіма учасниками системи. Додатково варто враховувати інтеграцію таких систем із зовнішніми сервісами, зокрема платіжними системами та календарями, що підвищує зручність використання для кінцевих користувачів. Також важливим є забезпечення стабільної роботи системи при високих навантаженнях у періоди активної реєстрації на популярні події. Це робить розробку подібних інформаційних платформ особливо актуальною в сучасних умовах цифровізації.

Метою випускної кваліфікаційної роботи є розробка мультиплатформеного комплексу для організації розважальних заходів, який включає серверну підсистему, вебінтерфейс і мобільний застосунок для Android та iOS. Програмний продукт має забезпечувати створення та редагування подій, пошук і фільтрацію інформації, реєстрацію користувачів, участь у заходах, оплату квитків, роботу з гаманцем організатора, просування подій, коментарі й відгуки, сповіщення в межах застосунку, а також перегляд статистики організатора.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

- 1) проаналізувати існуючі сервіси для організації подій;
 - 2) спроектувати структуру системи та бази даних;
 - 3) налаштувати систему входу для користувачів;
 - 4) розробити функціонал для створення заходів та роботи з відгуками
- інтегрувати оплату квитків;
- 5) створити мобільний застосунок;
 - 6) впровадити систему сповіщень;
 - 7) автоматизувати створення текстів за допомогою штучного інтелекту та реалізувати використання QR-кодів для швидкого доступу до квитків.

Об’єктом дослідження є процеси цифрової організації та адміністрування розважальних заходів у багатоклієнтському програмному середовищі.

Предметом дослідження є методи і засоби розробки мультиплатформеного програмного комплексу для створення, управління та відвідування подій, зокрема проектування мобільного застосунку, його архітектури, користувацького інтерфейсу та взаємодії з серверною частиною через REST API.

За темою випускної кваліфікаційної роботи бакалавра були подані та прийняті до публікації тези: Миколайчук В.Ю., Лімінович І.Д., Локтікова Т.М. Архітектура та реалізація клієнтської частини платформи для організації розважальних заходів на базі Kotlin Multiplatform. Всеукраїнська науково-практична он-лайн конференція здобувачів вищої освіти і молодих учених, присвячена Дню науки (15 травня 2026 року) [1].

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ МУЛЬТИПЛАТФОРМЕННОГО ПРОГРАМНОГО КОМПЛЕКСУ

1.1 Постановка задачі

Мультиплатформений комплекс проектується як єдине середовище для планування, публікації та відвідування розважальних заходів. Він має об'єднати серверну логіку, вебдоступ і мобільного клієнта так, щоб організатори могли вести свої події, а відвідувачі — швидко знаходити цікаві заходи, реєструватися на них і отримувати актуальну інформацію без зайвих дій. Окремою вимогою є безпечна робота з обліковими записами, коректне розмежування можливостей звичайного користувача та автора події, а також стабільний обмін даними між усіма частинами системи.

Головним завданням роботи є створення мобільного застосунку для Android і iOS, який стане основним інструментом взаємодії з платформою. Для реалізації підтримки обох операційних систем було обрано концепцію кросплатформеної розробки [2]. У застосунку потрібно реалізувати перегляд стрічки подій і добірок, авторизацію з підтвердженням електронної пошти, детальний перегляд заходу, пошук із фільтрами за категоріями, містом і локацією, додавання подій до обраного, коментування, формування відгуків після завершення заходу, перегляд профілю організатора та власної статистики. Для організаторів передбачається створення й редагування подій, керування учасниками, оплата квитків через зовнішній платіжний сервіс, робота з внутрішнім балансом, запуск рекламного просування, перегляд сповіщень у межах застосунку, а також використання інструментів автоматичного формування опису й обкладинки події.

Серверна підсистема виступає центральним вузлом комплексу: саме на ній зберігаються дані про користувачів, події, реєстрації, платежі, промокампанії та системні повідомлення. Логіка роботи з цими сутностями подана у вигляді

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

окремих програмних блоків, які через REST API [3] надають клієнтам однаковий набір операцій незалежно від типу пристрою. Блок облікових записів забезпечує реєстрацію, вхід у систему й редагування профілю; блок подій — повний цикл роботи з заходами від публікації до видалення; блок реєстрацій — облік учасників і контроль ліміту місць; блок оплат — проведення транзакцій за квитки та супутні послуги; блок промоцій — показ рекламних матеріалів у стрічці; блок сповіщень — формування повідомлень про дії, що стосуються активності користувача.

Зв'язок мобільного застосунку з сервером має будуватися на HTTP-запитах до REST API: клієнт передає параметри дії, сервер виконує перевірку прав доступу, змінює або повертає потрібні записи з бази даних і віддає результат у структурованому вигляді. Такий підхід дозволяє тримати бізнес-правила в одному місці, а в мобільному застосунку зосередитися на відображенні даних, навігації між екранами та зручності виконання типових сценаріїв — від пошуку події до її оплати або скасування участі.

Під час виконання роботи етапи розробки проходять один за одним: спочатку формуються вимоги до функціоналу та структури даних, далі проектується архітектура мобільного застосунку, реалізуються його екрани та інтеграція з API, після чого перевіряється коректність основних сценаріїв використання. Очікуваним результатом є працездатний мобільний клієнт, який разом із серверною частиною та вебінтерфейсом утворює цілісну систему для організації розважальних заходів і може бути розширений новими можливостями без переробки базової архітектури.

Важливою вимогою до системи є забезпечення зручного та інтуїтивно зрозумілого користувацького інтерфейсу, який дозволить швидко виконувати основні дії незалежно від рівня підготовки користувача. Крім того, архітектура програмного комплексу повинна підтримувати масштабування та подальше розширення функціональних можливостей без суттєвих змін у вже реалізованих компонентах. Це забезпечить можливість адаптації системи до нових вимог та збільшення кількості користувачів у майбутньому.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

1.2 Аналіз аналогів програмного продукту

Розробка програмного комплексу для організації розважальних заходів потребує попереднього вивчення рішень, які уже застосовуються на ринку. Сучасні event-платформи, зазвичай, поєднують вебінтерфейс для організатора та мобільний застосунок для учасників: через браузер створюють і налаштовують подію, а з телефона переглядають деталі, проходять реєстрацію, отримують квитки й сповіщення. Саме тому порівняння доцільно проводити за обома каналами взаємодії.

Luma [4] — платформа для створення, публікації та відвідування подій, орієнтована на культурні, освітні та розважальні заходи. Організатор через браузер формує сторінку події, налаштовує реєстрацію, обмеження місць, платні квитки та розсилку запрошень. Учасники можуть знайти захід у публічному календарі, зареєструватися через веб або мобільний застосунок Luma, додати подію до власного календаря й отримувати нагадування.

На головній сторінці організатора (рис. 1.1) відображається перелік створених і майбутніх подій, календарні сторінки спільнот і швидкий доступ до створення нового заходу. Інтерфейс мінімалістичний, без перевантаження корпоративними модулями.

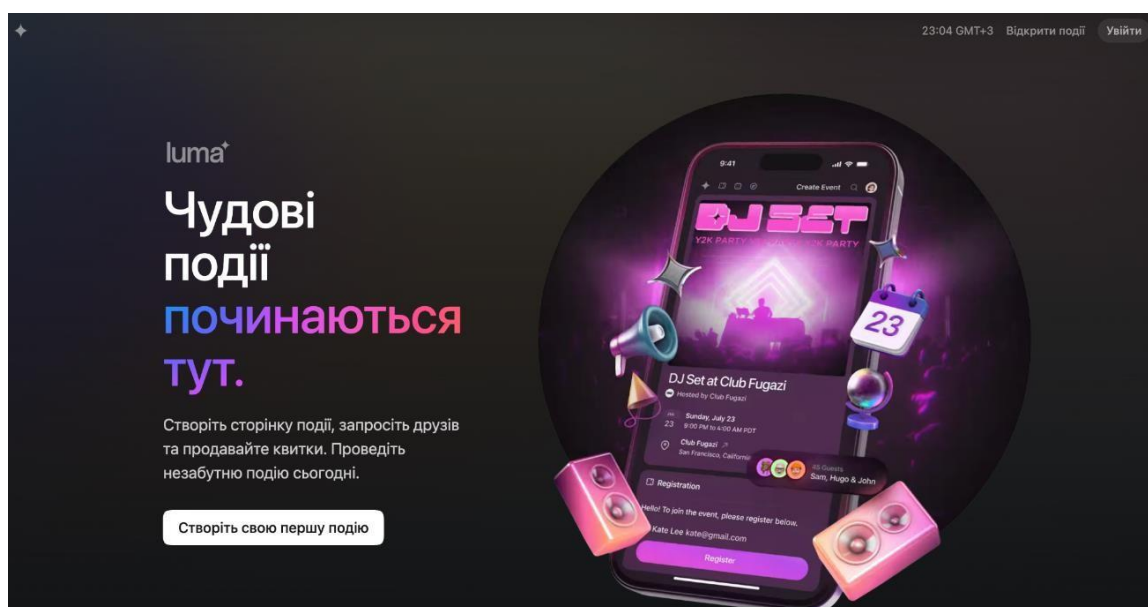


Рис. 1.1. Головна сторінка організатора Luma

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтерфейс створення події забезпечує комплексне налаштування заходу, де користувач вказує назву, дати проведення та деталі локації (офлайн-адреса або віртуальне посилання). У формі передбачено завантаження обкладинки для візуального оформлення картки події та додавання розгорнутого опису. Окремий блок параметрів дозволяє гнучко керувати ціноутворенням, налаштовувати необхідність попереднього схвалення заявок та встановлювати ліміти на кількість учасників. Таке поєднання інструментів, наведене на рис. 1.2, дає змогу швидко створювати події з професійним виглядом та необхідними обмеженнями, балансуючи між функціональністю та простотою налаштування.

The screenshot shows the 'Create Event' screen in the Luma app. At the top, there are navigation icons for 'Події' (Events), 'Календарі' (Calendars), and 'Відкрийте' (Open). The time '23:06 GMT' is shown in the top right. The main form is titled 'Назва події' (Event Name). Below this, there are sections for 'Початок' (Start) and 'Кінець' (End) times, a 'Додати місце проведення події' (Add event location) field, and a 'Додати опис' (Add description) field. The 'Параметри події' (Event parameters) section includes 'Ціна квитка' (Ticket price) set to 'Безкоштовно' (Free), 'Потребує схвалення' (Requires approval) with a toggle switch, and 'Місткість' (Capacity) set to 'Необмежений' (Unlimited). A large purple button at the bottom says 'Створити подію' (Create event). On the left, there is a preview image of confetti and a 'Тема' (Theme) selector set to 'Мінімальний' (Minimalist).

Рис. 1.2. Форма створення події в Luma

У мобільному застосунку користувач переглядає стрічку подій і календарі спільнот (рис. 1.3). Події можна фільтрувати за містом і датою, зберігати в обране та переходити на сторінку реєстрації. Додатково реалізовано навігацію за тематичними категоріями та пошук заходів у конкретних локаціях, що спрощує доступ до релевантного контенту. Також користувачам доступний розділ персоналізованих рекомендацій, який формується на основі підписок на

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

календарі спільнот та інтересів до певних типів активностей. Завдяки такій структурі інтерфейсу забезпечується швидка взаємодія з платформою та інтуїтивно зрозуміле вивчення актуальних пропозицій. Реалізація навігаційної панелі у нижній частині екрана забезпечує миттєвий доступ до основних модулів системи, включаючи створення нових подій та перегляд сповіщень, що оптимізує досвід користувача.

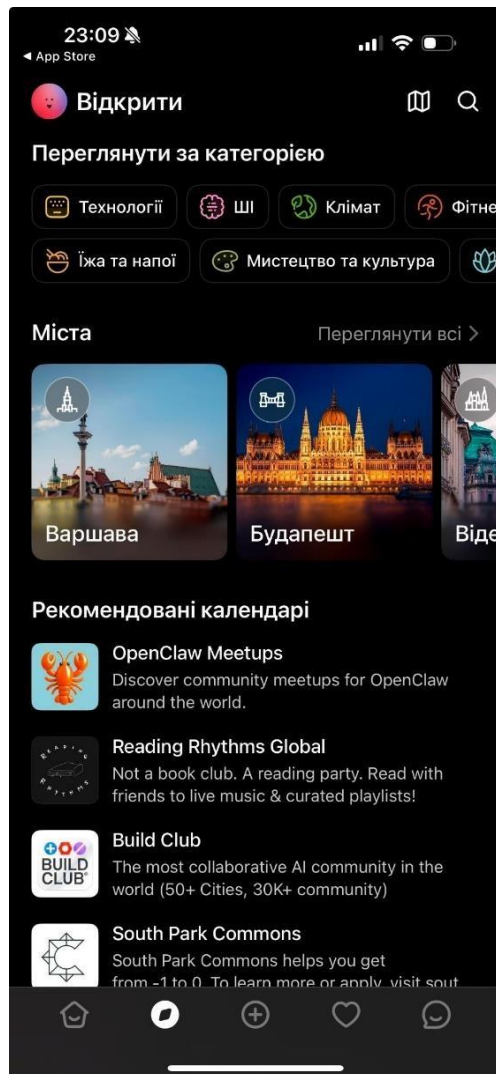


Рис. 1.3. Стрічка подій у мобільному застосунку Luma

Сторінка деталей події в застосунку (рис. 1.4) містить опис, карту локації, список учасників і кнопку реєстрації або покупки квитка. Після підтвердження участі подія з'являється в особистому календарі користувача. Така організація інформаційного простору дозволяє користувачеві отримати вичерпні відомості про захід без потреби переходу до зовнішніх ресурсів, що значно мінімізує когнітивне навантаження. Крім того, інтеграція з календарними системами

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

пристрою забезпечує своєчасне нагадування про майбутню подію, сприяючи підвищенню рівня відвідуваності та забезпечуючи комфортний супровід користувача на всіх етапах взаємодії із сервісом. Цей функціональний підхід демонструє високий рівень автоматизації процесів планування, що є критично важливим для створення позитивного користувацького досвіду у сучасних мультиплатформних івент-застосунках.

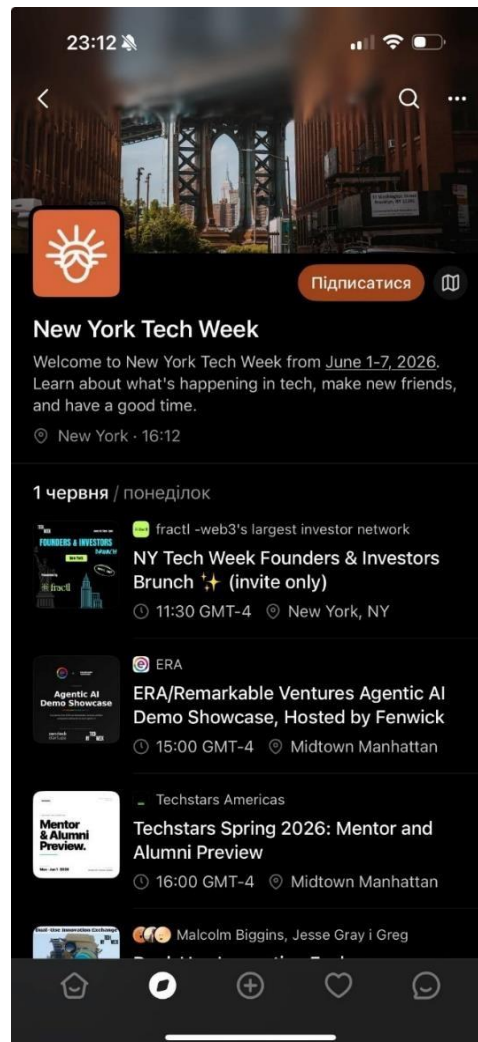


Рис. 1.4. Сторінка події в мобільному застосунку Luma

Peatix [5] — це платформа для публікації та продажу квитків на заходи різного масштабу: від воркшопів і концертів до фестивалів і спортивних подій. Базовий функціонал доступний безкоштовно: організатор створює подію на вебсайті, налаштовує типи квитків (безкоштовні або платні), керує списком учасників і переглядає базову статистику продажів. Учасники знаходять події

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

через сайт або мобільний застосунок Peatix, оформлюють реєстрацію й отримують електронний квиток для входу на захід.

У вебкабінеті організатора (рис. 1.5) зібрано список активних і завершених подій, показники реєстрацій і швидкий перехід до редагування заходу. Інтерфейс орієнтований на практичне керування квитками, а не на побудову багатосторінкового event-сайту.

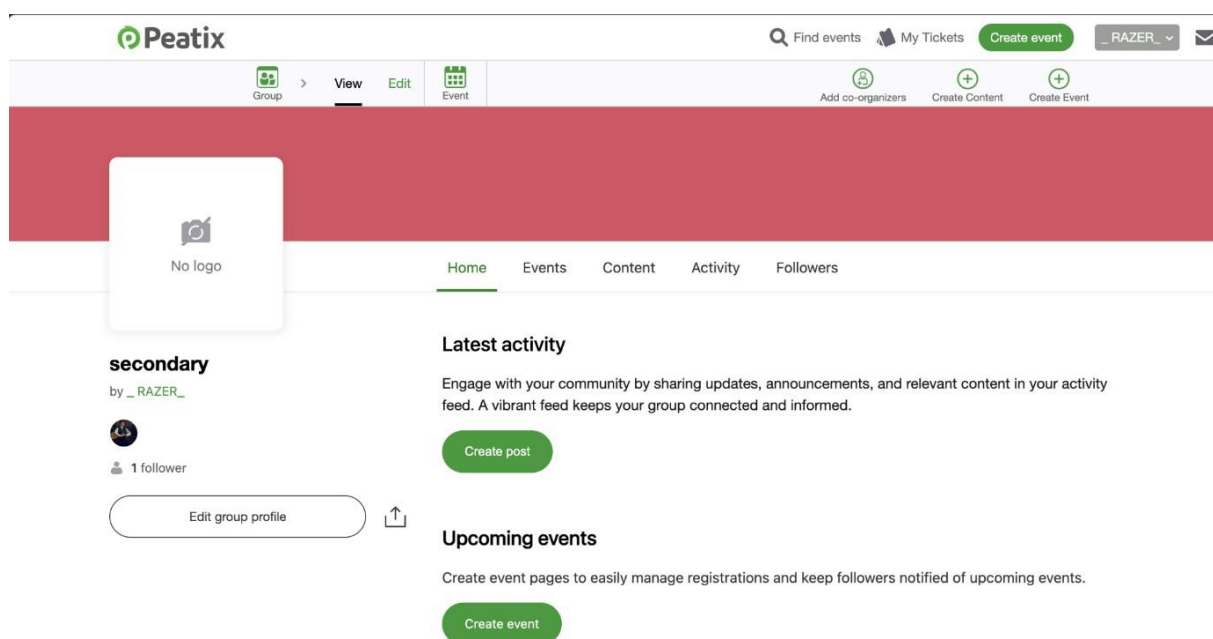


Рис. 1.5. Кабінет організатора Peatix

Під час створення події (рис. 1.6) організатор задає назву, дату, локацію, опис, обкладинку та параметри квитків. Передбачено обмеження кількості місць, кілька цінових категорій і налаштування публічності заходу. Для безкоштовних подій комісія платформи не стягується, що робить Peatix доступним для невеликих організаторів розважальних заходів. Інтерфейс створення події реалізований у вигляді покрокової форми, що дозволяє організатору послідовно вводити всі необхідні дані без перевантаження екрана. Такий підхід зменшує ймовірність помилок під час заповнення та підвищує зручність використання системи. Додатково передбачено валідацію введених даних, зокрема перевірку коректності дат, часу та обов'язкових полів. Це забезпечує цілісність інформації та запобігає створенню некоректних або неповних подій у системі.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

Рис. 1.6. Створення та налаштування події в Peatix

У мобільному застосунку користувач переглядає структурований каталог подій, що детально відображено на рис. 1.7. Інтерфейс забезпечує швидку навігацію завдяки використанню фільтрів за категоріями, датами та типами заходів, що мінімізує час на пошук потрібної інформації. Кожна картка заходу містить візуальний супровід, назву та інформацію про організатора, що значно покращує візуальне сприйняття контенту. Після обрання події користувач переходить на окрему сторінку, де представлені деталі заходу, актуальна ціна квитка та кнопка миттєвої реєстрації. Додатково система пропонує список пов'язаних категорій, дозволяючи користувачеві глибше досліджувати тематичні напрямки захоплень. Завдяки нижній навігаційній панелі забезпечується безперешкодний перехід між головним екраном, пошуком, квитками та профілем. Загалом сервіс максимально оптимізований для реалізації сценарію швидкого пошуку події та зручного оформлення квитка. Така організація інтерфейсу дозволяє користувачеві ефективно керувати своїм часом, маючи під рукою весь необхідний інструментарій для вибору та бронювання розважальних заходів. Забезпечена в такий спосіб інтерактивність значно підвищує залученість аудиторії та стимулює активну взаємодію з платформою.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

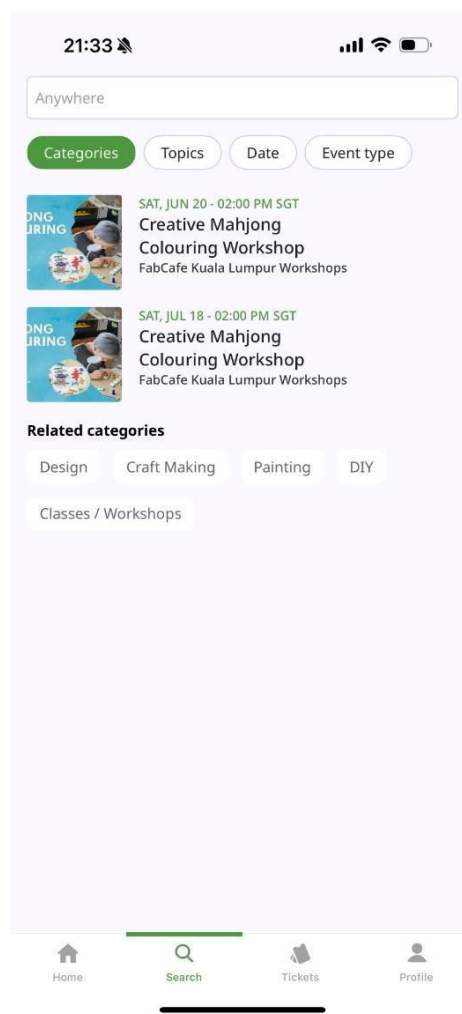


Рис. 1.7. Каталог подій у мобільному застосунку Peatix

Окремою можливістю Peatix є групи організаторів — постійні сторінки спільнот, на яких зібрано опис, підписники та майбутні заходи одного організатора або команди (рис. 1.8). Користувач може підписатися на групу й отримувати оновлення про нові події, не переходячи щоразу в загальний каталог. Для організатора це спрощує формування постійної аудиторії, проте можливості груп обмежуються переліком подій і базовою взаємодією з підписниками — без розширеної аналітики, внутрішнього гаманця чи вбудованих інструментів просування. Таким чином, функціональність груп має переважно інформаційний характер і слугує інструментом для підтримки зв'язку з аудиторією.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

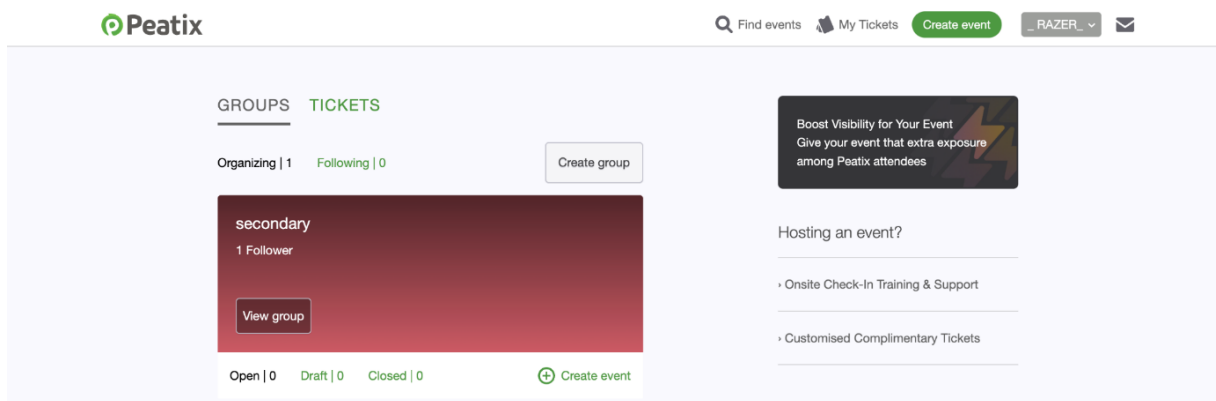


Рис. 1.8. Сторінка групи організатора на Peatix

Partiful [6] є платформою, орієнтованою на неформальні зустрічі, вечірки, дні народження та невеликі приватні події. Сервіс доступний через браузер і мобільний застосунок; гості можуть переглядати запрошення та підтверджувати участь навіть без встановлення програми — достатньо посилання. Основний акцент у системі зроблено на простоті взаємодії, де інтегровані інструменти для створення опитувань та обговорень допомагають організаторам швидко узгоджувати деталі заходу з учасниками. Завдяки такому підходу користувачі отримують зручний інструмент для соціальної координації, що спрощує процес планування без надмірної формалізації подій.

Сторінка події в браузері (рис. 1.9) виглядає як інтерактивне запрошення: назва, дата, локація, опис, список гостей із статусами Going / Maybe / Can't go, коментарі та реакції.

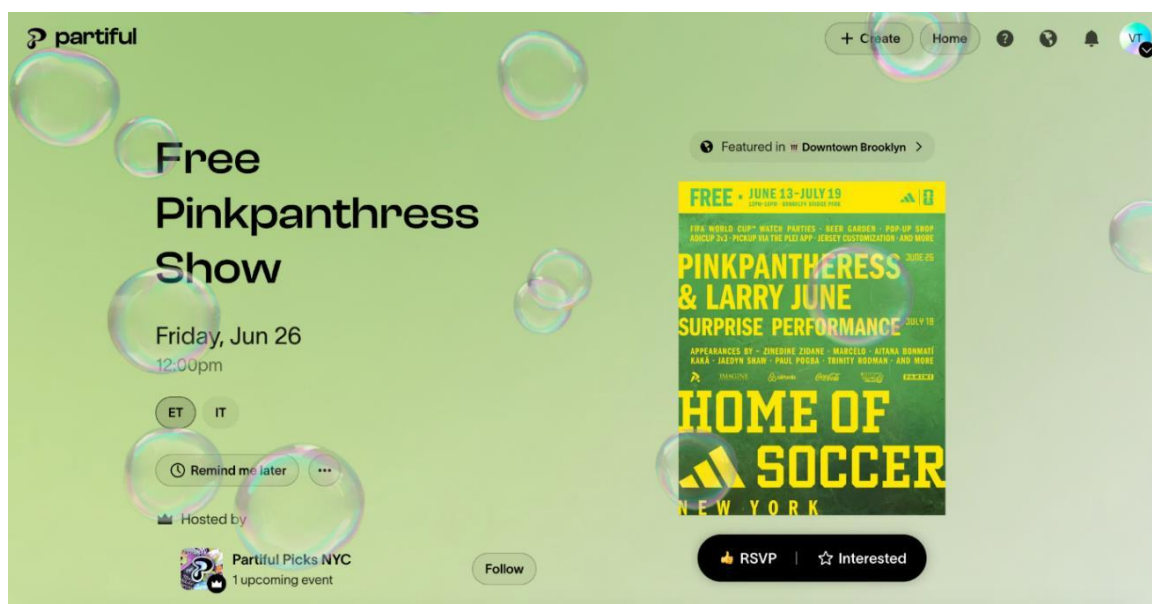


Рис. 1.9. Сторінка події Partiful

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

Створення нової події (рис. 1.10) займає кілька хвилин: обирається постер або тема, вводяться дата, час і місце. Акцент на швидкості, а не на розширеному наборі полів для публічного каталогу.

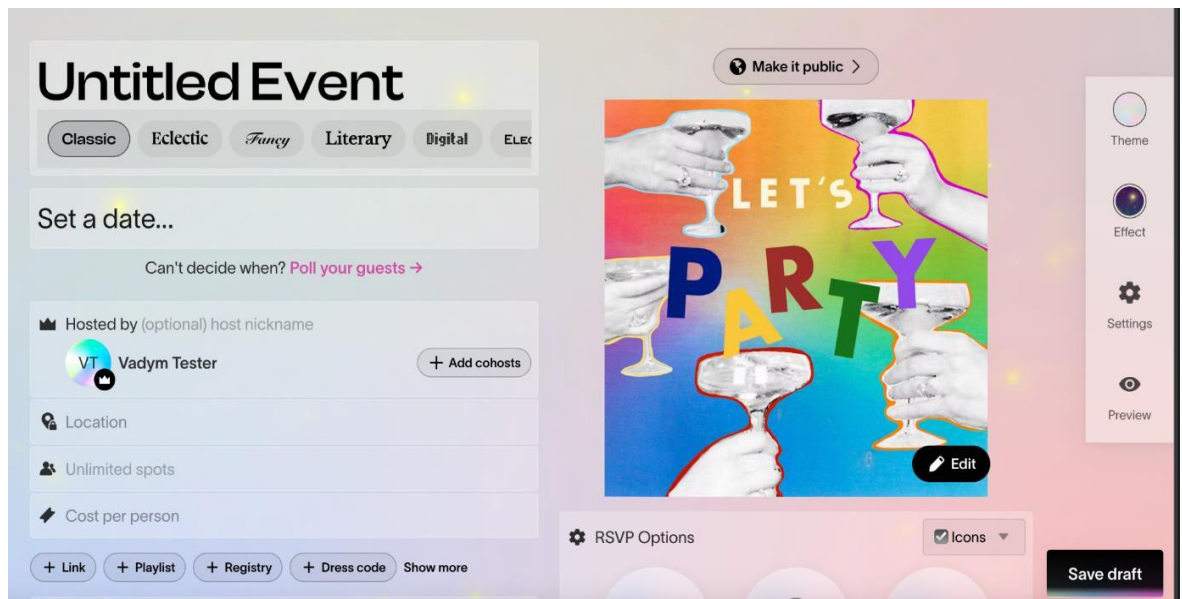


Рис. 1.10. Створення події в Partiful

Механізм запрошень і RSVP (рис. 1.11) — ключові переваги сервісу. Рис. 2.4Повноцінної інтеграції з платіжним шлюзом, аналітики організатора чи рекламного просування Partiful не пропонує. Інтерфейс відповіді на запрошення дозволяє користувачеві швидко вказати статус присутності, додати супровідну особу та залишити коментар із використанням GIF-анімацій. Така гнучкість у спілкуванні суттєво полегшує процес збору підтверджень для організаторів неформальних заходів. Окрім основних функцій підтвердження, система забезпечує інтуїтивно зрозумілий доступ до керування запрошеннями навіть для незареєстрованих користувачів. Це мінімізує бар'єри при вході в систему, гарантуючи високу конверсію відповідей від запрошених гостей. Завдяки такому лаконічному дизайну процес зворотного зв'язку стає майже миттєвим, що сприяє кращій динаміці взаємодії між усіма учасниками події.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

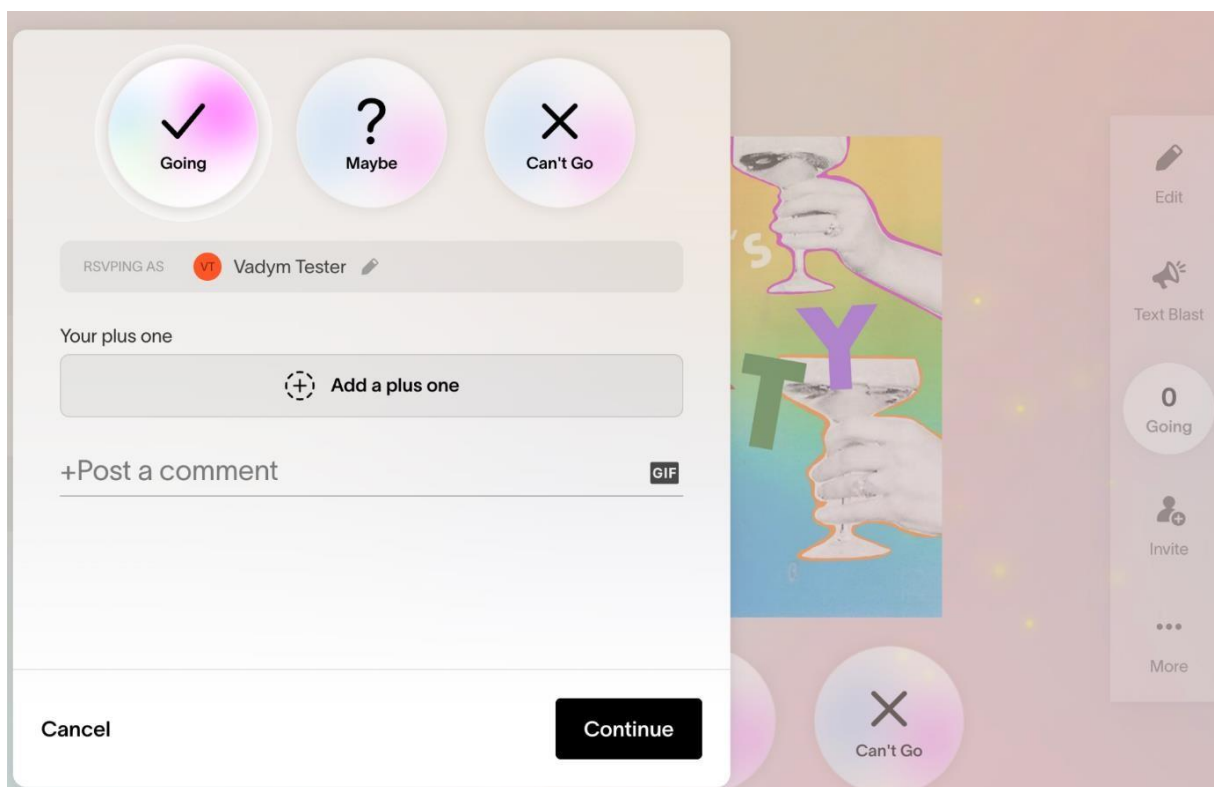


Рис. 1.11. Запрошення гостей і відстеження RSVP у Partiful

У мобільному застосунку (рис. 1.12) користувач бачить список майбутніх і минулих подій, може створити нову або переглянути публічні заходи поблизу. На головному екрані реалізовано зручну навігацію за категоріями, що дозволяє швидко знаходити тематичні концерти чи інтерактивні виставки. Інтегрований функціонал «Top 10 in São Paulo» надає актуальний рейтинг найпопулярніших заходів, допомагаючи користувачам орієнтуватися у культурному житті міста. Важливою особливістю системи є інноваційний розділ «Pick your vibe», де за допомогою технологій штучного інтелекту користувач може підібрати персоналізовані рекомендації відповідно до свого настрою або інтересів. Також передбачено можливість переходу до перегляду подій на карті, що значно спрощує планування маршрутів до місць проведення. Завдяки продуманій архітектурі панелі навігації доступ до квитків та профілю залишається миттєвим, що забезпечує комфортний користувацький досвід у будь-якій ситуації. Такий комплексний підхід до організації інтерфейсу перетворює застосунок з простого каталогу подій на повноцінний інтелектуальний помічник для планування дозвілля.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

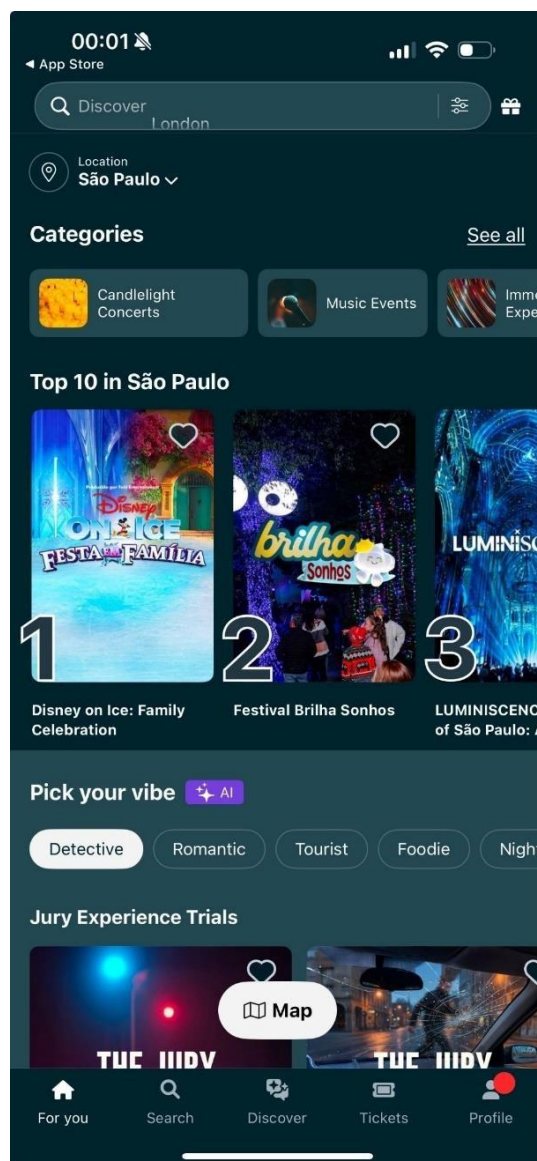


Рис. 1.12. Головний екран мобільного застосунку Partiful

Після детального розгляду функціональних можливостей досліджуваних платформ доцільно узагальнити результати у вигляді порівняльної таблиці. У табл. 1.1 наведено основні критерії, за якими оцінюються Luma, Peatix і Partiful, а також цільовий набір функцій програмного комплексу. Таке порівняння дозволяє наочно визначити спільні параметри аналогів, їхні обмеження та функціональні недоліки, які обґрунтовують доцільність розробки власного рішення. Це структурування дає змогу чітко окреслити конкурентні переваги розроблюваної системи та зосередитися на впровадженні інструментів, що найбільш затребувані користувачами в умовах сучасного ринку івент-менеджменту.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

Таблиця 1.1

Порівняльний огляд і аналіз аналогів

Параметр	Luma	Peatix	Partiful
Вебінтерфейс організатора	+	+	+
Мобільний застосунок	+	+	+
Публічний каталог подій	+	+	+
Створення та редагування події	+	+	+
Пошук і фільтрація	+	+	+
Реєстрація учасників	+	+	+
Електронні квитки	+	+	+
Онлайн оплата	+	+	частково
Групи авторів	+	+	-
Аналітика організатора	базова	базова	-
Просування подій	-	-	-
Відгуки	-	-	-
Коментарі	-	-	+
AI-генерація контенту	-	-	-
Складність інтерфейсу	низька	середня	низька

Проведений аналіз показує, що Luma, Peatix і Partiful вирішують різні задачі організації заходів. Luma зручна для публічних подій із реєстрацією через веб і мобільний застосунок, але не пропонує розвинених інструментів просування, аналітики та зворотного зв'язку після заходу. Peatix добре підходить для продажу квитків і роботи з постійною аудиторією через групи організаторів, проте її можливості зосереджені переважно на квитковому сценарії. Partiful виграє у швидкому створенні неформальних подій і комунікації між гостями, однак слабо покриває публічний каталог, платні заходи та інструменти для організаторів.

Жодна з проаналізованих платформ не поєднує в одному рішенні пошук подій, оплату квитків, монетизацію, рекламне просування, відгуки учасників і

автоматизацію підготовки контенту. Через це доцільним є створення— мультиплатформеного комплексу, який об’єднує окремі переваги аналогів і доповнює їх функціями, що в них відсутні або реалізовані обмежено.

У межах проекту передбачено роботу через веб- і мобільний клієнти, онлайн-оплату, внутрішній гаманець, просування подій, коментарі й відгуки, статистику організатора, in-app сповіщення та AI-генерацію матеріалів для подій. Такий підхід надає можливість зменшити функціональні недоліки існуючих сервісів і забезпечити цілісну взаємодію між організаторами та учасниками розважальних заходів.

1.3 Вибір архітектури програмного комплексу

Вибір архітектури визначає, наскільки стабільно працюватиме мультиплатформений комплекс при зростанні кількості користувачів, наскільки просто буде розвивати окремі частини системи та чи зможуть веб- і мобільний клієнти працювати за однаковими правилами. Оскільки комплекс передбачає одночасну роботу кількох клієнтських застосунків із спільними даними про події, квитки, оплату та профілі користувачів, на етапі проектування необхідно було обрати модель, яка централізує бізнес-логіку та не дублює її на кожному клієнті.

Розглянуто три базові підходи. Повністю монолітний застосунок не підходить через необхідність окремої підтримки вебінтерфейсу та мобільної версії. Мікросервісна архітектура забезпечує масштабованість, але для поточного масштабу мультиплатформений комплекс створює надмірну складність розгортання, моніторингу та узгодження даних між сервісами. Тому обрано клієнт-серверну архітектуру [7]: серверна частина виконує бізнес-логіку і роботу з даними, а клієнти відповідають лише за відображення інформації та введення команд користувача. Загальне представлення обраної моделі наведено на рис. 1.13.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

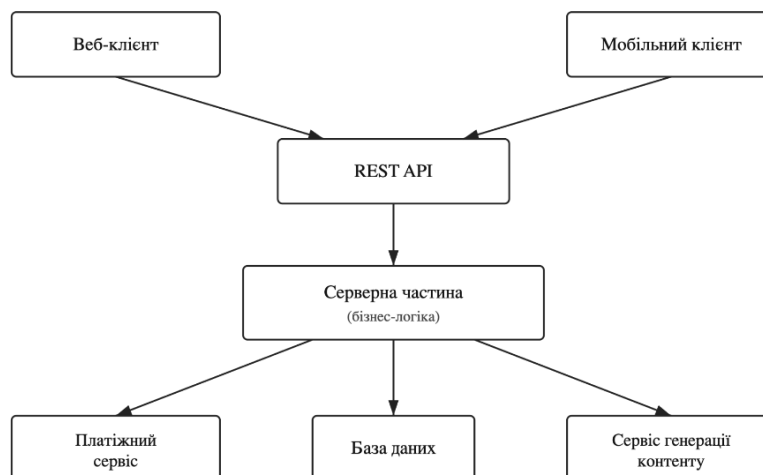


Рис. 1.13. Архітектура програмного комплексу

У межах обраної моделі система структурована за трирівневою схемою (рис. 1.14). На рівні представлення розміщуються веб- і мобільний клієнти. На рівні застосунків — серверна частина з REST API, яка обробляє запити, перевіряє права доступу та формує відповіді. На рівні даних — сховище інформації про користувачів, події, квитки, платежі та пов'язані сутності. Таке розділення дозволяє змінювати інтерфейс, серверну логіку або організацію даних відносно незалежно одне від одного.

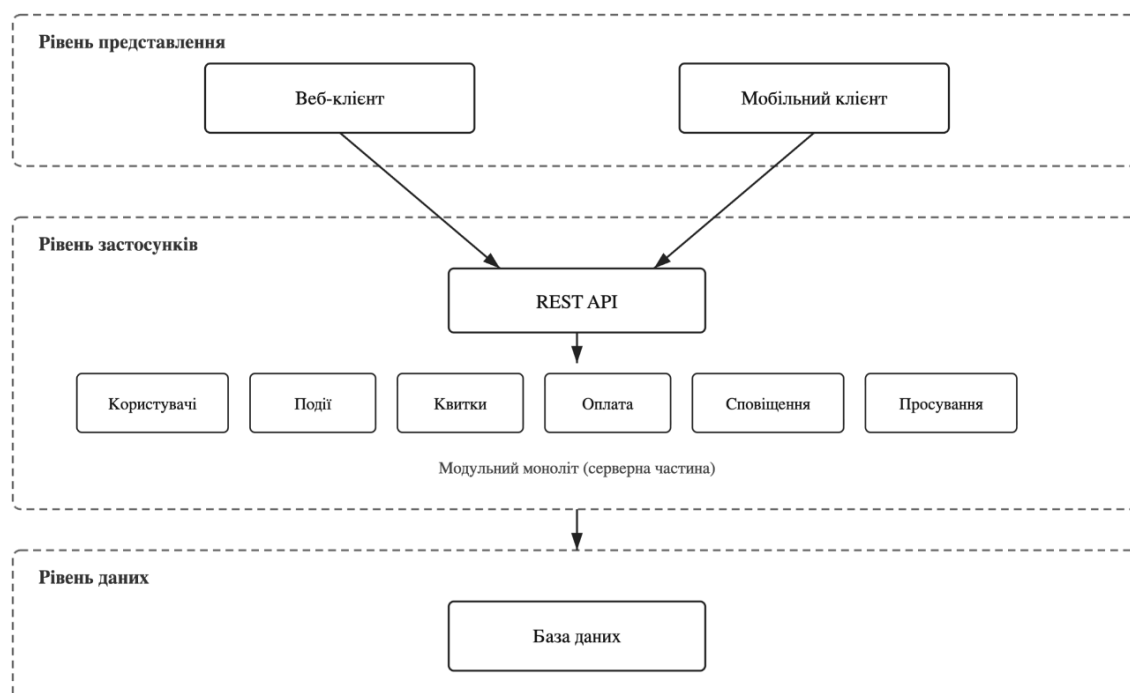


Рис. 1.14. Трирівнева архітектура програмного комплексу

Для зв'язку між рівнями обрано REST API як архітектурний стиль взаємодії. Він передбачає доступ до ресурсів через уніфіковані операції і

забезпечує слабке зв'язування між клієнтами та сервером: обидва типи застосунків можуть користуватися одним програмним інтерфейсом без перенесення бізнес-правил у клієнтський код. Порівняно з тісно пов'язаними протоколами REST простіший у підтримці та краще підходить для системи з кількома незалежними клієнтами.

Окремим архітектурним рішенням є централізація бізнес-логіки на сервері. Реєстрація на події, оплата, формування сповіщень, управління профілем і просування заходів мають виконуватися за єдиними правилами незалежно від того, через який клієнт звернувся користувач. Це зменшує ризик розбіжностей між веб- і мобільною версіями та спрощує контроль коректності операцій.

Серверну частину доцільно проектувати як модульний моноліт [8]: функціональні області виділені в окремі логічні модулі — користувачі, події, квитки, оплата, сповіщення, просування, — але працюють у складі одного серверного застосунку. Такий варіант простіший у розгортанні та супроводі, ніж набір незалежних мікросервісів, і водночас дозволяє локалізувати зміни в межах окремих блоків.

Архітектура комплексу також передбачає підключення зовнішніх сервісів через серверну частину — зокрема платіжних систем і сервісів генерації контенту (рис. 1.13). Це не дає клієнтам напряду залежати від сторонньої логіки і зосереджує контроль інтеграцій у серверному шарі. Такий підхід забезпечує гнучкість при масштабуванні системи, оскільки заміна одного зовнішнього провайдера на іншого потребує коригування лише на серверному рівні, без необхідності оновлення клієнтських застосунків. Це значно підвищує стабільність роботи всього програмного комплексу та спрощує процес впровадження нових функціональних можливостей.

Окремо варто обрати внутрішню архітектуру мобільного клієнта, який у загальній схемі комплексу належить до рівня представлення. Для нього доцільно застосувати шаблон MVVM [9], оскільки він розділяє інтерфейс, стан екрана та роботу з даними: UI реагує на зміни стану, ViewModel обробляє дії користувача, а звернення до сервера та локального сховища винесені в нижні шари. Такий

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

поділ спрощує підтримку великої кількості екранів — стрічка подій, деталі заходу, профіль, гаманець, сповіщення — без змішування мережевої логіки з кодом інтерфейсу. Застосування MVVM у поєднанні з реактивним підходом до управління станом дозволяє уникнути типових проблем синхронізації даних, забезпечуючи високу чуйність інтерфейсу навіть при складних фонових обчисленнях.

Мобільний застосунок доцільно будувати за багатошаровою схемою (рис. 1.15): рівень UI, рівень ViewModel, рівень репозиторіїв і доменної логіки, а також окремі шари мережевої взаємодії з REST API і локального зберігання даних авторизації та налаштувань. Оскільки застосунок має працювати на Android і iOS, доцільно винести спільну логіку в один модуль, а платформові особливості — в окремі реалізації. Це узгоджується з обраною архітектурою комплексу: мобільний клієнт не дублює серверну бізнес-логіку, а використовує єдиний REST API.

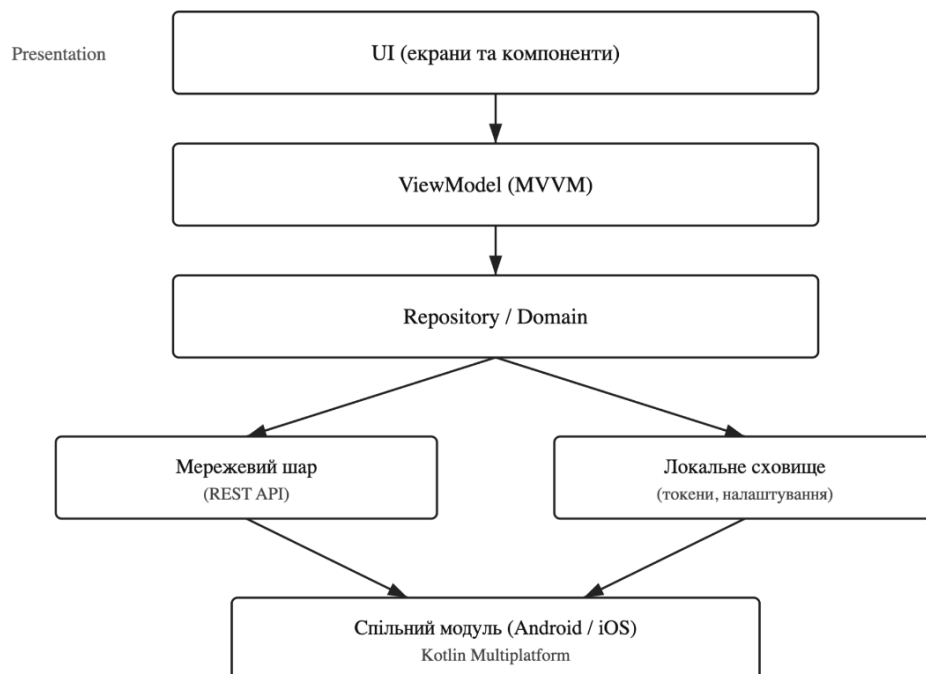


Рис. 1.15. Архітектура мобільного застосунку

Отже, для мультиплатформеного комплексу обрано клієнт-серверну багатошарову архітектуру з REST API, централізованою серверною логікою, модульним монолітом, серверною інтеграцією зовнішніх сервісів і

багатошаровою MVVM-структурою мобільного клієнта. Це рішення відповідає вимогам мультиплатформеного комплексу та створює основу для обґрунтування інструментальних засобів у наступному підрозділі. Такий розподіл відповідальності забезпечує повну ізоляцію бізнес-логіки від особливостей інтерфейсу, дозволяючи оновлювати клієнтські додатки без модифікації серверного ядра. Використання MVVM у мобільному клієнті спрощує підтримку реактивного стану екранів, що критично для відображення динамічних даних, таких як статуси реєстрації або зміни в гаманці. Це рішення повністю відповідає вимогам мультиплатформеного комплексу та створює основу для обґрунтування вибору інструментальних засобів.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Реалізація мультиплатформеного комплексу охоплює серверну підсистему, вебінтерфейс і мобільний застосунок для Android та iOS. Для кожного з цих компонентів потрібен власний набір програмних засобів, який забезпечує функціонал, визначений у постановці задачі, і водночас відповідає обраній клієнт-серверній моделі, REST API, модульному моноліту та багатошаровій структурі мобільного клієнта. При виборі технологій враховувалися зручність супроводу, наявність документації, сумісність компонентів між собою та можливість подальшого розвитку системи.

Серверну підсистему комплексу доцільно реалізувати мовою Python [10] з використанням Django [11] і Django REST Framework [12]. Python має розвинену екосистему бібліотек для веброзробки, а Django надає ORM, міграції, вбудовані механізми роботи з користувачами та адміністрування даних. Django REST Framework доповнює це засобами побудови REST API: серіалізація, валідація запитів, пагінація, права доступу та формування структурованих відповідей у форматі JSON. Такий стек дозволяє реалізувати програмні блоки системи — облікові записи, події, реєстрації, оплати, промоції, сповіщення — як окремі модулі в межах одного серверного проекту і забезпечити однакову взаємодію з

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

веб- і мобільним клієнтами через HTTP-запити до REST API. У серверній частині також передбачено модуль формування QR-кодів для квитків; структура бази даних формувалася з урахуванням основних сутностей системи.

Для автентифікації користувачів у системі обрано JSON Web Token [13] на базі Simple JWT. Access-токен використовується для доступу до захищених ресурсів, refresh-токен — для поновлення сесії без повторного входу. Токенний механізм є безстанним, однаково підходить для браузера та мобільного застосунку і дозволяє серверу контролювати права звичайного користувача та організатора події.

Структуровані дані системи доцільно зберігати в PostgreSQL [14]. Для мультиплатформеного комплексу характерні взаємопов'язані сутності — користувачі, події, квитки, транзакції, сповіщення, промокампанії — тому реляційна модель даних забезпечує цілісність записів, транзакційність операцій і коректність зв'язків між таблицями. Окремо від табличних даних у системі передбачено зберігання медіафайлів — обкладинок подій, зображень профілів, QR-кодів квитків.

Вебінтерфейс комплексу доцільно реалізувати на React [15] із маршрутизацією React Router [16]. React дозволяє будувати інтерфейс із компонентів, що спрощує розробку сторінок каталогу подій, деталей заходу, профілю, квитків і форм організатора. Взаємодія з сервером здійснюється через HTTP-запити до REST API; для авторизованих операцій використовується JWT-токен. Решта вебінтерфейсу сформована в рамках спільної розробки програмного комплексу.

Для мобільного застосунку на Android і iOS обрано Kotlin Multiplatform [17] і Compose Multiplatform. KMP [18] дає змогу винести спільну логіку — мережевий шар, репозиторії, ViewModel, моделі даних — у модуль commonMain, а платформові особливості залишити в androidMain і iosMain. Архітектура мобільного клієнта будується за шаблоном MVVM: екрани на Compose працюють із ViewModel, які формують стан інтерфейсу та обробляють дії користувача. Для звернення до REST API використовується Ktor Client [19] із

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

kotlinx.serialization [20] для роботи з JSON. Залежності між компонентами керуються через Koin [21]. Локальне зберігання токенів авторизації та налаштувань — через DataStore Preferences [22]. Завантаження і показ зображень — через Coil [23]. Навігація між екранами — засобами Navigation Compose [24]. Цей стек покриває сценарії mobile-клієнта: перегляд стрічки подій, пошук і фільтрацію, роботу з профілем, гаманцем, сповіщеннями в межах застосунку, коментарі, відгуки, оплату, просування заходів і перегляд статистики організатора.

Для розробки та перевірки роботи комплексу достатньо персонального комп'ютера з обсягом оперативної пам'яті не менше 8 ГБ, накопичувачем SSD і доступом до мережі Internet. На робочому місці використовується Android Studio з JDK не нижче 17-ї версії для збирання мобільного застосунку; перевірку інтерфейсу, навігації та взаємодії з API зручно проводити на емуляторі або фізичному смартфоні з Android 7.0 і новіше чи iPhone з актуальною версією iOS. Кінцевим користувачам вебінтерфейсу достатньо сучасного браузера з підтримкою JavaScript [25]; для mobile-клієнта — пристрою з встановленим застосунком і доступом до мережі.

Додатково слід відзначити, що обраний стек технологій забезпечує високу масштабованість системи, що є критично важливим при збільшенні кількості користувачів та подій у майбутньому. Використання модульної архітектури дозволяє незалежно розвивати окремі компоненти системи без необхідності значних змін у всій інфраструктурі. Це також спрощує процес тестування та впровадження нових функціональних можливостей.

Застосування REST API як єдиного каналу взаємодії між клієнтами та сервером забезпечує уніфікованість обміну даними та спрощує інтеграцію сторонніх сервісів. Окремо важливим є використання сучасних механізмів кешування та оптимізації запитів до бази даних, що позитивно впливає на продуктивність системи.

Крім того, вибрані інструментальні засоби мають широку спільноту розробників, що забезпечує швидке вирішення потенційних проблем та доступ

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

до оновлень. Це підвищує надійність і довгострокову підтримуваність програмного комплексу.

Важливим є також те, що використані технології підтримують кросплатформену розробку, що дозволяє зменшити витрати часу на створення та підтримку окремих клієнтів для різних операційних систем. Це особливо актуально для мобільної частини системи, де спільна бізнес-логіка значно скорочує дублювання коду.

Окрему увагу приділено безпеці передачі даних, зокрема використанню HTTPS та захищеного зберігання токенів авторизації. Такий підхід мінімізує ризики несанкціонованого доступу до персональної інформації користувачів.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи сформовано постановку задачі щодо програмного комплексу для організації розважальних заходів. Визначено, що система має поєднувати серверну підсистему, вебінтерфейс і мобільний застосунок для Android та iOS, а ключовим завданням є створення mobile-клієнта з повним набором сценаріїв взаємодії користувача з платформою — від перегляду та пошуку подій до реєстрації, оплати, роботи з профілем і функцій організатора. Установлено, що всі клієнти повинні працювати з єдиним джерелом даних через REST API, забезпечуючи узгодженість інформації про події, квитки та пов'язані операції. Актуальність реалізації саме такого підходу зумовлена постійним зростанням вимог користувачів до оперативності отримання інформації у мобільному форматі. У сучасних умовах ринку івент-менеджменту швидкість доступу до сервісів стає визначальним фактором конкурентоспроможності будь-якого цифрового продукту. Відтак, дослідження базувалося на принципах системного аналізу, що дозволило комплексно оцінити взаємозв'язки між усіма компонентами майбутньої архітектури. Процес проектування передбачав ретельне вивчення поведінкових патернів потенційних користувачів, а також визначення технічних обмежень, які можуть вплинути на

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

продуктивність системи. Саме завдяки такому всебічному охопленню поставлених завдань було забезпечено фундамент для подальшого вибору технологічного стека та інструментарію розробки. Такий підхід дозволив врахувати не лише функціональні вимоги, а й критерії надійності та безпеки, що є критично важливими при обробці персональних даних користувачів та проведенні транзакцій. Особливу увагу було приділено гнучкості інтерфейсу, щоб адаптувати його до різних розмірів екранів мобільних пристроїв та специфіки взаємодії в умовах динамічного середовища. Окремим етапом роботи стала розробка прототипу, що дало змогу верифікувати обрані концептуальні рішення ще до початку етапу активного програмування. Таким чином, вдалося виявити та мінімізувати потенційні ризики, пов'язані зі складністю навігації, забезпечивши високу якість кінцевого програмного продукту.

Проведено аналіз програмних аналогів — Luma, Peatix і Partiful. Порівняння показало, що сучасні event-платформи, як правило, поєднують вебінструменти для організатора з мобільним доступом для учасників, але різняться за масштабом, складністю інтерфейсу та набором функцій. Частина сервісів орієнтована на вузькі сценарії або перевантажена можливостями, що ускладнює щоденне користування; інші не покривають окремі потреби, зокрема розширену роботу з оплатою, внутрішнім обліком коштів, промоцією подій чи генерацією контенту. На основі порівняльної таблиці функціональних можливостей визначено цільовий набір функцій для мультиплатформеного комплексу і обґрунтовано доцільність власної розробки комплексу, який поєднує зручний mobile-клієнт із необхідним серверним функціоналом.

Окремо підкреслено важливість можливості подальшого розширення функціональності без суттєвих змін у базовій структурі системи. Отримані результати аналізу стали основою для формування вимог до програмного комплексу та вибору оптимальних технологічних рішень.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

РОЗДІЛ 2 ПРОЕКТУВАННЯ МУЛЬТИПЛАТФОРМЕННОГО ПРОГРАМНОГО КОМПЛЕКСУ

2.1 Визначення варіантів використання

Функціональна модель описує, які дії може виконувати користувач у межах програмного комплексу. Вона формується на основі вимог, визначених у постановці задачі, та результатів аналізу аналогів і охоплює серверну підсистему, вебінтерфейс і мобільний застосунок для Android та iOS. Основним каналом доступу до функцій передбачено мобільний застосунок; вебінтерфейс доповнює комплекс, зокрема роботою з профілем користувача.

Єдиним актором системи є Користувач. Він може виступати як учасник подій або як організатор власних заходів — окремий тип облікового запису для цього не передбачено, роль визначається конкретною дією. У межах комплексу користувач проходить реєстрацію та автентифікацію, переглядає каталог заходів і виконує пошук із фільтрами, реєструється на події та оплачує квитки, керує профілем, створює та редагує власні події, залишає коментарі й відгуки, працює з гаманцем, оформлює просування заходів і переглядає сповіщення в межах застосунку. Окремо передбачено формування електронного квитка з QR-кодом після успішної реєстрації або оплати. Сповіщення відображаються лише всередині застосунку; окремі сповіщення поза застосунком не передбачені.

На діаграмі варіантів використання (рис. 2.1) зображено актора «Користувач» і двадцять два прецеденти, що відповідають основним функціям системи. Прецеденти верхньої частини діаграми стосуються облікового запису та перегляду каталогу подій. Центральну частину займають сценарії участі в заході — реєстрація, оплата квитка, скасування реєстрації. Нижче розміщено прецеденти організатора та додаткові можливості: створення і редагування події, перегляд учасників, управління профілем, робота з гаманцем, просування заходів і перегляд сповіщень. Усі прецеденти пов'язані з одним актором, що відображає єдину точку взаємодії людини з системою незалежно від типу застосунку. Така

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

концепція єдиного актора спрощує архітектуру системи, оскільки дозволяє динамічно змінювати рівень прав доступу залежно від контексту поточної операції без необхідності створення окремих ієрархій користувачів.

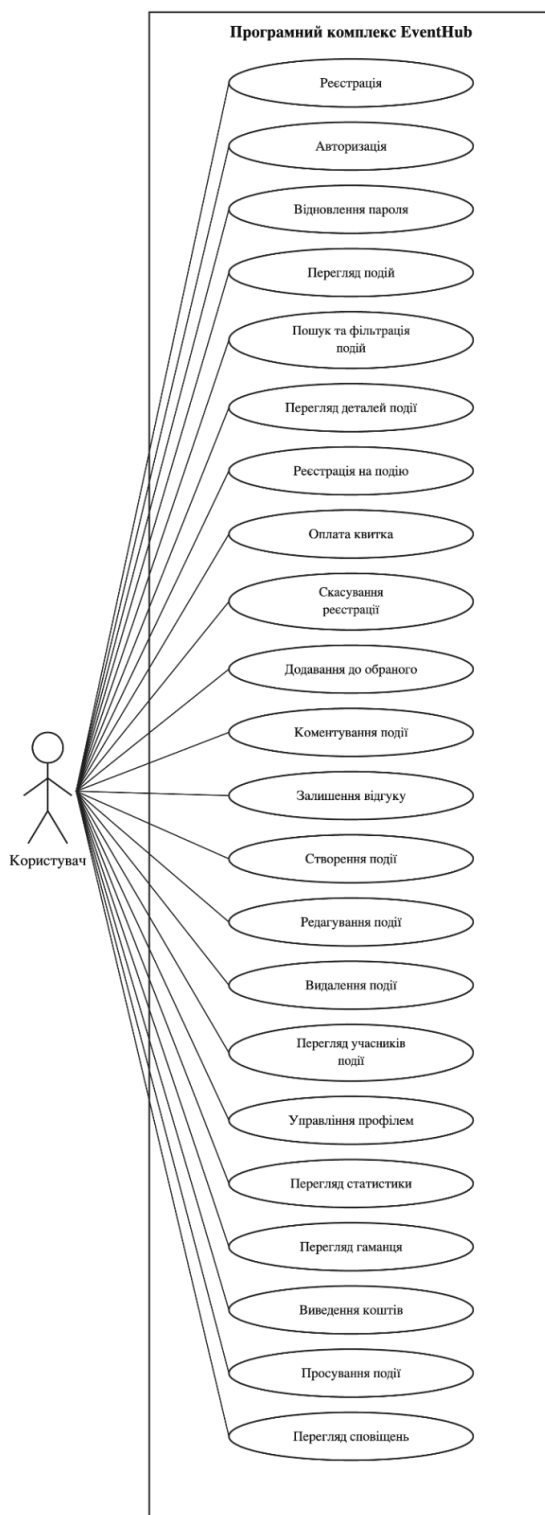


Рис. 2.1. Діаграма варіантів використання програмного комплексу

Прецеденти на діаграмі охоплюють повний цикл роботи користувача з комплексом: від створення облікового запису та входу в систему до перегляду

заходів, участі в подіях, соціальної взаємодії, фінансових операцій і перегляду сповіщень. Частина з них стосується дій будь-якого користувача як учасника подій, інша — функцій організатора власних заходів; межа між цими ролями визначається не окремим типом профілю, а конкретним сценарієм використання. Для кожного варіанту використання нижче наведено назву прецеденту, основного актора та короткий опис відповідних можливостей системи. Опис кожного прецеденту наведено в табл. 2.1–2.22. Такий структурований підхід до класифікації дозволяє уникнути дублювання функціональності та забезпечує логічну цілісність архітектури програмного продукту, що значно спрощує подальше тестування та масштабування системи.

Таблиця 2.1

Опис прецеденту «Реєстрація»

Назва варіанту використання	Реєстрація
Основні актори	Користувач
Короткий опис	Користувач створює обліковий запис, вводить адресу електронної пошти та пароль, підтверджує реєстрацію через електронну пошту

Таблиця 2.2

Опис прецеденту «Авторизація»

Назва варіанту використання	Авторизація
Основні актори	Користувач
Короткий опис	Користувач входить у систему за адресою електронної пошти і паролем, отримує токени доступу для подальшої роботи з комплексом, що забезпечує захищений перехід до основної користувацької зони застосунку та персоналізацію його інтерфейсу

Таблиця 2.3

Опис прецеденту «Відновлення пароля»

Назва варіанту використання	Відновлення пароля
Основні актори	Користувач
Короткий опис	Користувач ініціює скидання пароля, отримує посилання на електронну пошту та задає новий пароль для облікового запису

Таблиця 2.4

Опис прецеденту «Перегляд подій»

Назва варіанту використання	Перегляд події
Основні актори	Користувач
Короткий опис	Користувач переглядає стрічку подій, актуальні та рекомендовані добірки, каталог за категоріями, просунуті заходи

Таблиця 2.5

Опис прецеденту «Пошук та фільтрація подій»

Назва варіанту використання	Пошук та фільтрація подій
Основні актори	Користувач
Короткий опис	Користувач виконує пошук і фільтрацію за ключовими словами, категоріями, містом, локацією та датою проведення, що дозволяє швидко знаходити релевантні заходи та отримувати актуальну інформацію про них у зручному форматі

Таблиця 2.6

Опис прецеденту «Перегляд деталей події»

Назва варіанту використання	Перегляд деталей подій
Основні актори	Користувач
Короткий опис	Користувач відкриває сторінку обраної події та переглядає повну інформацію про захід: опис, дату, локацію, вартість, організатора, коментарі

Таблиця 2.7

Опис прецеденту «Реєстрація на подію»

Назва варіанту використання	Реєстрація на подію
Основні актори	Користувач
Короткий опис	Користувач реєструється на безкоштовну подію; після підтвердження отримує електронний квиток із QR-кодом; може додати подію до локального календаря пристрою

Таблиця 2.8

Опис прецеденту «Коментування події»

Назва варіанту використання	Коментування події
Основні актори	Користувач
Короткий опис	Користувач залишає коментар під подією та переглядає коментарі інших учасників. Написаний текст автоматично додається до загального списку під описом заходу із зазначенням імені автора та часу публікації

Таблиця 2.9

Опис прецеденту «Оплата квитка»

Назва варіанту використання	Оплата квитка
Основні актори	Користувач
Короткий опис	Користувач оплачує квиток на платну подію; після успішної оплати отримує електронний квиток із QR-кодом

Таблиця 2.10

Опис прецеденту «Створення події»

Назва варіанту використання	Створення події
Основні актори	Користувач
Короткий опис	Користувач створює нову подію, задає назву, опис, дату проведення, категорії, місто, локацію, ліміт учасників і вартість квитка; за потреби генерує опис або зображення заходу

Таблиця 2.11

Опис прецеденту «Перегляд статистики»

Назва варіанту використання	Перегляд статистики
Основні актори	Користувач
Короткий опис	Користувач переглядає статистику власної активності: кількість створених подій, відвіданих заходів, отриманих відгуків, що дозволяє оцінити рівень залученості аудиторії, аналізувати динаміку популярності заходів та ефективно планувати майбутню діяльність

Таблиця 2.12

Опис прецеденту «Скасування реєстрації»

Назва варіанту використання	Скасування реєстрації
Основні актори	Користувач
Короткий опис	Користувач скасовує попередню реєстрацію на подію з урахуванням доступних варіантів повернення коштів

Таблиця 2.13

Опис прецеденту «Додавання до обраного»

Назва варіанту використання	Додавання до обраного
Основні актори	Користувач
Короткий опис	Користувач додає подію до списку обраних або видаляє її з цього списку для подальшого перегляду

Таблиця 2.14

Опис прецеденту «Управління профілем»

Назва варіанту використання	Управління профілем
Основні актори	Користувач
Короткий опис	Користувач переглядає та редагує власний профіль: ім'я, опис, аватар, електронну пошту і пароль; переглядає календар збережених подій, перелік створених і відвіданих заходів; за потреби видаляє обліковий запис. Аналогічні операції доступні у вебінтерфейсі, що гарантує повну синхронізацію внесених змін між усіма платформами в режимі реального часу

Таблиця 2.15

Опис прецеденту «Редагування події»

Назва варіанту використання	Редагування події
Основні актори	Користувач
Короткий опис	Користувач змінює параметри власної події: опис, дату, локацію, вартість, ліміт учасників та інші атрибути

Таблиця 2.16

Опис прецеденту «Залишення відгуку»

Назва варіанту використання	Залишення відгуку
Основні актори	Користувач
Короткий опис	Користувач публікує або видаляє відгук після завершення події, за якою брав участь

Таблиця 2.17

Опис прецеденту «Видалення події»

Назва варіанту використання	Видалення події
Основні актори	Користувач
Короткий опис	Користувач видаляє власну подію, яку більше не планує проводити. Після підтвердження видалення інформація про подію вилучається із системи та більше не відображається в каталозі заходів для інших користувачів, що забезпечує миттєве оновлення бази даних та підтримання актуальності загального списку доступних подій у реальному часі

Таблиця 2.18

Опис прецеденту «Перегляд учасників події»

Назва варіанту використання	Перегляд учасників події
Основні актори	Користувач
Короткий опис	Користувач-організатор переглядає список зареєстрованих учасників і статистику реєстрацій на власну подію, що дозволяє оцінити рівень зацікавленості заходом та контролювати процес підготовки до його проведення

Таблиця 2.19

Опис прецеденту «Перегляд гаманця»

Назва варіанту використання	Перегляд гаманця
Основні актори	Користувач
Короткий опис	Користувач переглядає баланс гаманця та історію транзакцій, пов'язаних з оплатою квитків і просуванням подій, що дає змогу контролювати рух коштів, відстежувати виконані операції та перевіряти фінансову активність у системі

Таблиця 2.20

Опис прецеденту «Виведення коштів»

Назва варіанту використання	Виведення коштів
Основні актори	Користувач
Короткий опис	Користувач-організатор подає запит на виведення коштів, отриманих від продажу квитків

Таблиця 2.21

Опис прецеденту «Просування події»

Назва варіанту використання	Просування події
Основні актори	Користувач
Короткий опис	Користувач-організатор оформлює просування власної події для підвищення її видимості в каталозі заходів

Таблиця 2.22

Опис прецеденту «Перегляд сповіщень»

Назва варіанту використання	Перегляд сповіщень
Основні актори	Користувач
Короткий опис	Користувач переглядає сповіщення в межах застосунку про дії, пов'язані з його подіями та активністю

Наведені таблиці описують повний набір дій користувача в мультиплатформеному комплексі: від реєстрації та входу в систему до перегляду заходів, участі в подіях, керування профілем, створення власних заходів, соціальної взаємодії, роботи з гаманцем і сповіщеннями у межах застосунку.

2.2 Розробка бази даних

Інформаційна модель мультиплатформеного комплексу формується на основі функціональної моделі, і визначає дані, необхідні для реалізації прецедентів користувача. Для зберігання даних обрано реляційну модель і систему управління базами даних PostgreSQL [26], що відповідає вимогам до цілісності, масштабованості та підтримки складних зв'язків між сутностями.

Структура бази даних побудована за принципами нормалізації [27]: атрибути згруповано в окремі сутності так, щоб уникнути надмірного дублювання даних і забезпечити однозначну ідентифікацію записів. Для

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

критичних операцій передбачено обмеження унікальності — зокрема, один користувач може залишити лише один відгук на подію, одну реєстрацію на захід і одну позначку «обране» для конкретної події.

Концептуальну структуру бази даних представлено ER-діаграмами (рис. 2.2, 2.3). На них кожна сутність зображена прямокутником із переліком атрибутів, первинний ключ позначено підкресленням, зовнішній ключ — позначкою FK, зв'язок між сутностями — ромбом, а кардинальність — значеннями 1, N або M. Через великий обсяг моделі її подано двома рисунками: на рис. 2.2 відображено сутності облікового запису та каталогу подій, на рис. 2.3 — сутності участі користувачів, соціальної взаємодії, фінансових операцій і просування.

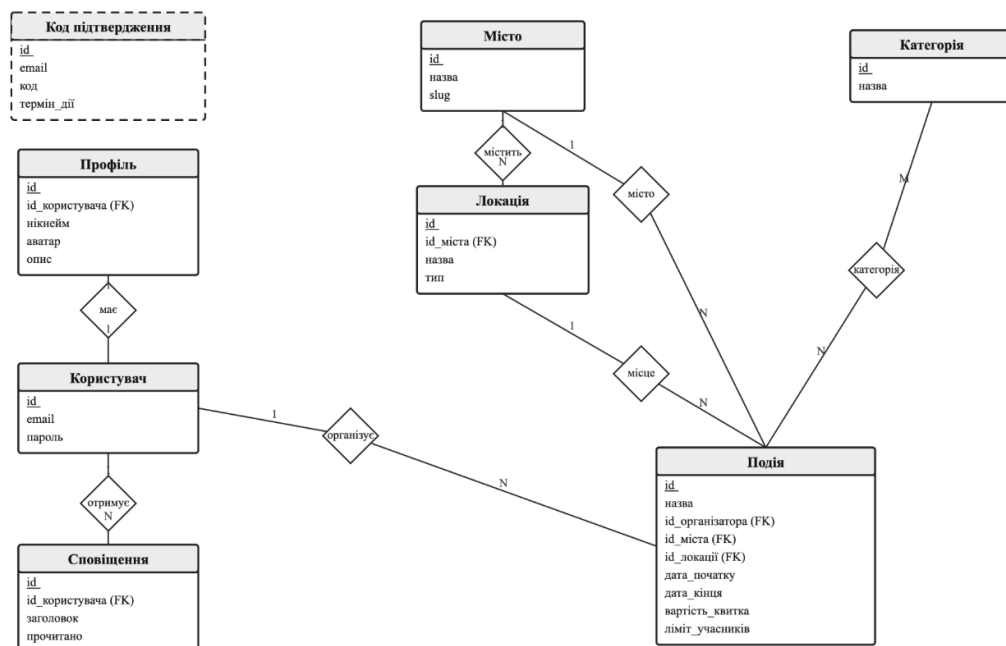


Рис. 2.2. ER-діаграма бази даних програмного комплексу. Обліковий запис і каталог подій

На рис. 2.2 зображено сутності, що забезпечують реєстрацію, автентифікацію та роботу з профілем користувача, а також формування каталогу заходів. Сутність Користувач пов'язана з Профілем зв'язком «один до одного» і містить дані для входу в систему. Код підтвердження зберігає тимчасові дані для реєстрації та відновлення пароля до створення облікового запису. Сповіщення

пов'язані з користувачем зв'язком «один до багатьох» і призначені для відображення повідомлень у межах застосунку.

Центральною сутністю каталогу є Подія, яка пов'язана з Користувачем як організатором, із довідниками Місто та Локація, а також із Категорією зв'язком «багато до багатьох». Довідник Місто пов'язаний з Локацією зв'язком «один до багатьох», що дозволяє задавати місце проведення заходу в межах обраного міста. На рис. 2.3 зображено сутності участі користувачів у заходах, соціальної взаємодії, фінансових операцій та просування подій. Сутності Користувач та Подія наведено також на рис. 2.3 як опорні для описаних на цьому рисунку зв'язків.

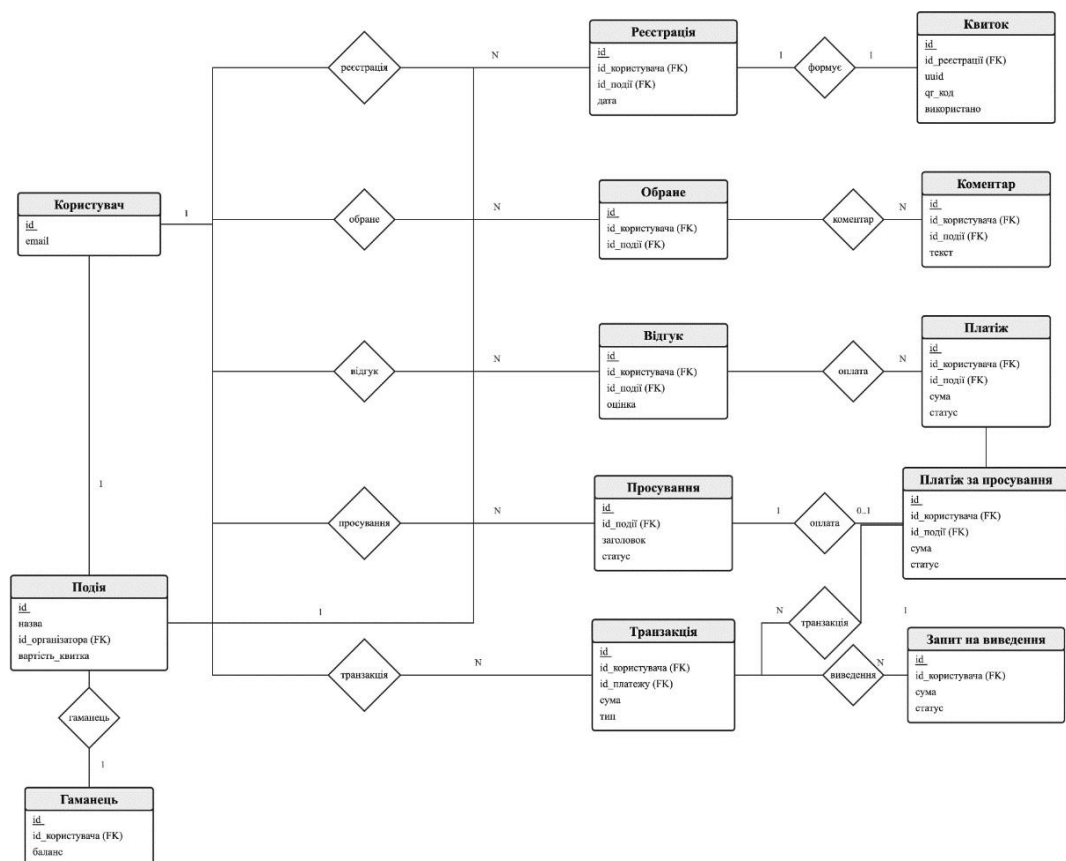


Рис. 2.3. ER-діаграма бази даних програмного комплексу. Участь користувачів і фінансові операції

Реєстрація фіксує факт участі користувача в заході і пов'язана з Квитком зв'язком «один до одного»; останній містить унікальний ідентифікатор та зображення QR-коду для підтвердження участі. Обране, Коментар і Відгук пов'язують користувача з подією.

Фінансовий блок включає сутності Платіж, Гаманець, Транзакція та Запит на виведення. Платіж зберігає дані про оплату квитка, Гаманець — поточний баланс організатора, Транзакція — історію змін балансу, Запит на виведення — заявки на виведення коштів.

Опис основних сутностей і їх призначення наведено в табл. 2.23–2.28.

Таблиця 2.23

Сутності облікового запису та профілю користувача

Сутність	Призначення	Основні атрибути	Зв'язки
Користувач	Обліковий запис для входу в систему	id, email, пароль	один до одного з Профілем; один до багатьох з Подією, Сповіщенням, Реєстрацією, Обраним, Коментарем, Відгуком, Платежем, Просуванням, Транзакцією, Запитом на виведення, Платежем за просування
Профіль	Публічні дані користувача в застосунку	id, id_користувача в(FK), нікнейм, аватар, опис	один до одного з Користувачем
Код підтвердження	Тимчасове зберігання коду під час реєстрації або відновлення пароля	id, email, код, термін дії	окрема службова сутність; зв'язок із Користувачем встановлюється за email після підтвердження коду

Таблиця 2.24

Сутності каталогу подій та довідника

Сутність	Призначення	Основні атрибути	Зв'язки
Подія	Опис розважального заходу	id, назва, id_організатора (FK), id_міста (FK), id_локації (FK), дата_початку, дата_кінця, вартість_квитка, ліміт_учасників	багато до одного з Користувачем (організатор), Містом, Локацією; багато до багатьох з Категорією; один до багатьох з Реєстрацією, Обраним, Коментарем, Відгуком, Платежем, Просуванням, Платежем за просування
Категорія	Класифікація заходів для перегляду та фільтрації	id, назва	багато до багатьох з Подією
Місто	Довідник міст	id, назва, slug	один до багатьох з Локацією та Подією
Локація	Місце проведення заходу в межах міста	id, id_міста (FK), назва, тип	багато до одного з Містом; один до багатьох з Подією

Таблиця 2.25

Сутності участі користувачів у подіях

Сутність	Призначення	Основні атрибути	Зв'язки
Реєстрація	Факт участі користувача в події	id, id_користувача (FK), id_події (FK), дата	багато до одного з Користувачем і Подією; один до одного з Квитком
Обране	Збережена користувачем подія	id, id_користувача (FK), id_події (FK)	багато до одного з Користувачем і Подією
Коментар	Текстовий коментар до події	id, id_користувача (FK), id_події (FK), текст	багато до одного з Користувачем і Подією
Відгук	Оцінка завершені події	id, id_користувача (FK), id_події (FK), оцінка	багато до одного з Користувачем і Подією

Таблиця 2.26

Сутності платежів та гарантія

Сутність	Призначення	Основні атрибути	Зв'язки
Платіж	Оплата квитка на платну подію	id, id_користувача (FK), id_події (FK), сума, статус	багато до одного з Користувачем (платник) і Подією; один до багатьох з Транзакцією
Гаранець	Баланс коштів організатора	id, id_користувача (FK), баланс	один до одного з Користувачем

продовж. табл. 2.26

Транзакція	Запис про зміну балансу гаманця	id, id_користувача (FK), id_платежу (FK), сума, тип	багато до одного з Користувачем; багато до одного з Платежем (за наявності зв'язку)
Запит на виведення	Заявка на виведення коштів із гаманця	id, id_користувача (FK), сума, статус	багато до одного з Користувачем

Таблиця 2.27

Сутності просування та сповіщень

Сутність	Призначення	Основні атрибути	Зв'язки
Просування	Рекламне просування події	id, id_події (FK), заголовок, статус	багато до одного з Подією і Користувачем (організатор); один до нуля або одного з Платежем за просування
Платіж за просування	Оплата послуги просування	id, id_користувача (FK), id_події (FK), сума, статус	багато до одного з Користувачем і Подією; нуль або один до одного з Просуванням
Сповіщення	Повідомлення користувачу в застосунку	id, id_користувача (FK), заголовок, прочитано	багато до одного з Користувачем

Сутність електронного квитка

Сутність	Призначення	Основні атрибути	Зв'язки
Квиток	Електронний квиток із QR-кодом	id, id_реєстрації (FK), uuid, qr_код, використано	один до одного з Реєстрацією

2.3 Розробка функціональних алгоритмів роботи програми

Функціонування мультиплатформеного програмного комплексу базується на взаємодії серверної та клієнтської підсистем, які реалізують узгоджені алгоритми обробки даних та виконання користувацьких сценаріїв. Детально розглянемо логіку виконання ключових операцій системи, зокрема автентифікацію користувачів, роботу з подіями, платежами та формування електронних квитків.

Серверна частина відповідає за обробку бізнес-логіки, збереження даних та забезпечення цілісності інформації, тоді як клієнтські застосунки ініціюють запити до REST API та отримують результати у форматі JSON. Така взаємодія забезпечує централізоване виконання алгоритмів на стороні сервера та уніфіковану поведінку системи для всіх платформ. Усі серверні компоненти реалізовано на основі фреймворку Django [28], який забезпечує маршрутизацію запитів, ORM-взаємодію з базою даних та побудову REST API.

Запити від клієнтських застосунків проходять через єдину систему контролерів, де виконується перевірка автентифікації, валідація даних та запуск відповідних бізнес-операцій. У результаті формується стандартизована відповідь у форматі JSON, що забезпечує однакову обробку даних на всіх клієнтських платформах.

Декомпозиція серверної логіки реалізована через набір функціональних модулів, кожен з яких відповідає окремій предметній області системи.

Узагальнена структура серверних модулів, їх призначення та основні операції наведені в табл. 2.29.

Таблиця 2.29

Функціональні модулі серверної підсистеми

Модуль	Призначення	Сутності бази даних	Основні операції на сервері
Облікові записи	Реєстрація, автентифікація, профіль	Користувач, Профіль, Код підтвердження	надсилання коду підтвердження, вхід у систему, оновлення токена, редагування профілю
Події та довідники	Каталог заходів, пошук, соціальна взаємодія	Подія, Категорія, Місто, Локація, Реєстрація, Обране, Коментар, Відгук	перегляд і фільтрація подій, створення та редагування подій, реєстрація, коментарі, відгуки, обране
Квитки	Електронний квиток після реєстрації або оплати	Квиток, Реєстрація	формування квитка, генерація QR-коду, перевірка дійсності

Платежі	Оплата платних подій	Платіж, Подія, Користувач	створення платежу, взаємодія із зовнішнім платіжним сервісом, оновлення статусу
Гаманець	Фінанси організатора	Гаманець, Транзакція, Запит на виведення	нарахування коштів, історія операцій, заявки на виведення
Просування	Реклама подій	Просування, Платіж за просування, Подія	створення банера, оплата просування, активація та деактивація
Сповіщення	Повідомлення в застосунку	Сповіщення, Користувач	формування in-app сповіщень, позначення прочитаним

Такий підхід дозволяє чітко розділити відповідальність між компонентами системи та структурувати бізнес-логіку відповідно до предметних областей. Кожен модуль реалізує визначений набір операцій через REST API, що забезпечує логічну ізоляцію функціоналу та спрощує масштабування системи.

Окремі сценарії системи реалізують багатокрокові алгоритми взаємодії між модулями. Наприклад, процес реєстрації на подію та формування електронного квитка включає послідовність серверних операцій, які відображено на рис. 2.3. Кожен із таких сценаріїв виконується як послідовність взаємопов'язаних дій, де результат одного модуля є вхідними даними для

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

наступного. Це дозволяє забезпечити узгодженість стану системи та виключити дублювання логіки між різними частинами серверної підсистеми. Додатково така організація процесів спрощує тестування та налагодження окремих етапів алгоритмів. У результаті підвищується надійність виконання критичних операцій, зокрема реєстрації користувачів та обробки платежів.

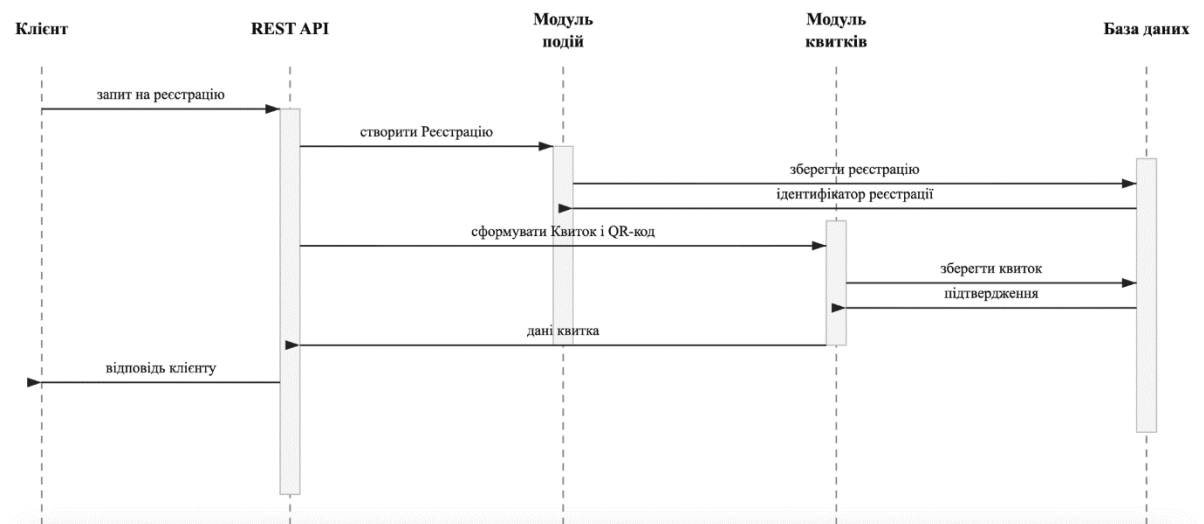


Рис. 2.3. Послідовність обробки реєстрації на подію та формування електронного квитка на сервері

У межах цього алгоритму модуль подій створює запис реєстрації, після чого модуль квитків формує електронний квиток із QR-кодом. Для платних подій додатково залучається модуль платежів, який взаємодіє із зовнішнім платіжним сервісом та підтверджує транзакцію перед завершенням сценарію.

Алгоритм автентифікації базується на використанні JWT-токенів. Після входу користувач отримує access-токен і refresh-токен, що дозволяє реалізувати безперервну сесію без повторного введення даних. Для захищених операцій використовується перевірка токена на кожному запиті.

Алгоритми роботи з подіями включають фільтрацію, пошук та посторінкову видачу даних. Клієнтський застосунок формує запити з параметрами фільтрації, а сервер повертає відповідний набір результатів із підтримкою пагінації та довантаження.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

На клієнтській стороні мобільний застосунок реалізує алгоритми відображення та оновлення даних, отриманих із серверної частини. Кожен екран взаємодіє зі своєю моделлю подання, яка ініціює запити до API та обробляє відповіді у форматі JSON. Отримані дані перетворюються у внутрішні моделі та використовуються для формування стану інтерфейсу користувача. Перехід між екранами здійснюється відповідно до визначеної навігаційної структури, що забезпечує послідовність користувацьких сценаріїв. Такий підхід дозволяє розділити логіку представлення та бізнес-логіку, що підвищує підтримуваність застосунку.

2.4 Реалізація мультиплатформеного комплексу

Реалізація мультиплатформеного програмного комплексу охоплює три взаємопов'язані складові: серверну підсистему, мобільний застосунок для Android та iOS, а також вебклієнт для роботи з профілем користувача. Усі компоненти функціонують у межах єдиної клієнт–серверної архітектури та взаємодіють через уніфікований програмний інтерфейс застосунку, що використовує формат JSON для обміну даними. Такий підхід забезпечує узгодженість інформації між усіма платформами та дозволяє централізовано керувати бізнес-логікою системи.

Загальна структура програмного комплексу відображає принцип розподіленої модульності, де сервер виконує роль єдиного джерела даних і бізнес-логіки, а клієнтські застосунки відповідають за інтерфейс користувача та виконання сценаріїв взаємодії. На рис. 2.4 наведено узагальнену архітектуру системи, яка ілюструє взаємодію між серверною підсистемою, мобільним застосунком та вебклієнтом через єдиний API-шар. Діаграма демонструє централізований характер обробки даних на сервері та розподіл функціональності між різними типами клієнтів. Також вона відображає принцип відокремлення бізнес-логіки від представлення, що підвищує масштабованість і підтримуваність системи. Архітектура системи передбачає чітке розмежування відповідальності між серверною та клієнтською частинами,

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

що дозволяє незалежно розвивати кожен компонент. Завдяки використанню єдиного API забезпечується уніфікований формат взаємодії для всіх платформ, що спрощує інтеграцію нових клієнтів. Такий підхід зменшує зв'язаність між модулями та підвищує гнучкість системи при внесенні змін у бізнес-логіку. Крім того, централізована обробка даних на сервері дозволяє забезпечити узгодженість інформації в режимі реального часу для всіх користувачів.

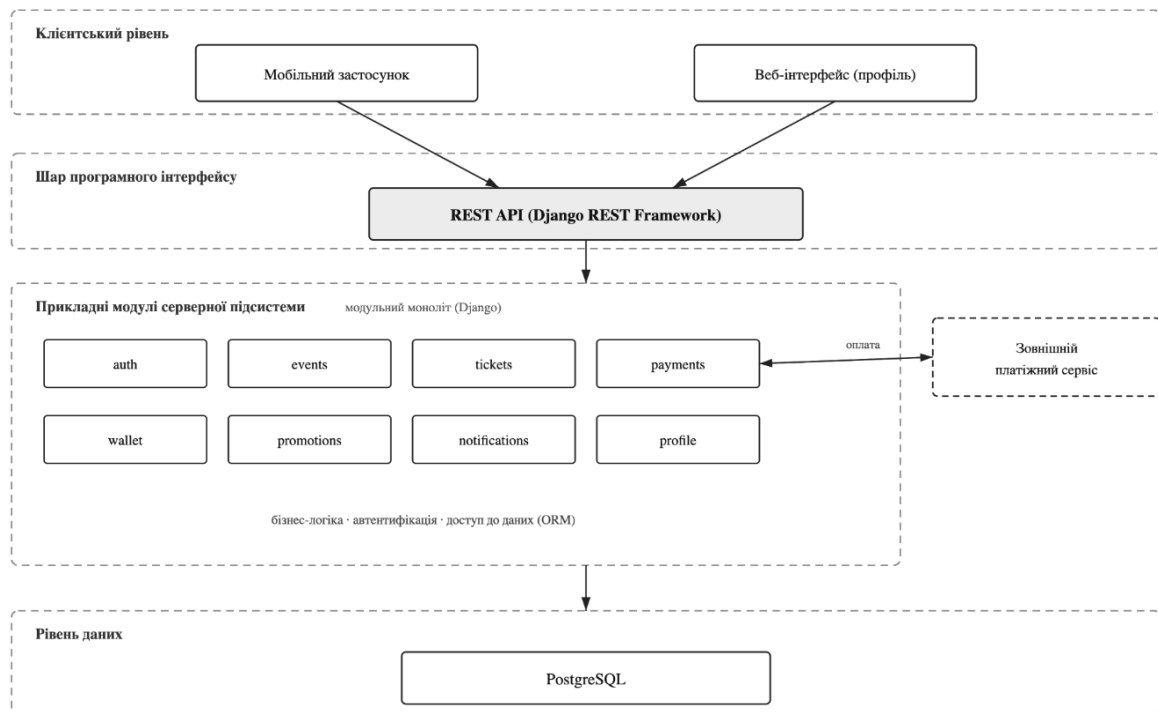


Рис. 2.4. Загальна архітектура програмного комплексу

Серверна частина реалізує централізовану обробку запитів, управління даними та забезпечення цілісності інформації. Вона побудована на основі модульного принципу, де кожен функціональний блок відповідає окремій предметній області (облікові записи, події, платежі, квитки, гаманець, просування та сповіщення). Така структура дозволяє ізолювати бізнес-логіку та забезпечує масштабованість системи при подальшому розширенні функціональності. Для забезпечення контролю доступу до заходів у системі реалізовано механізм генерації та перевірки електронних квитків із використанням QR-кодів (Додаток А).

Мобільний застосунок є основним інтерфейсом користувача та реалізує ключові сценарії взаємодії з системою. Його архітектура побудована за багат шаровим принципом, що відображає чіткий поділ відповідальності між компонентами. Структура багат шарової архітектури мобільного застосунку наведена на рис. 2.5, де показано взаємодію між рівнями інтерфейсу, моделі подання, доменної логіки, репозиторіїв та мережевого шару. Реалізація інтерфейсу мобільного застосунку здійснена з використанням підходу Jetpack Compose Multiplatform [29], що дозволяє створювати єдину декларативну UI-логіку для Android та iOS та забезпечує повторне використання компонентів між платформами. Взаємодія між рівнем інтерфейсу, бізнес-логікою та джерелами даних у мобільному застосунку реалізована за шаблоном MVVM, що включає отримання даних з API, їх трансформацію у внутрішні моделі та керування навігацією між екранами (Додаток Б).

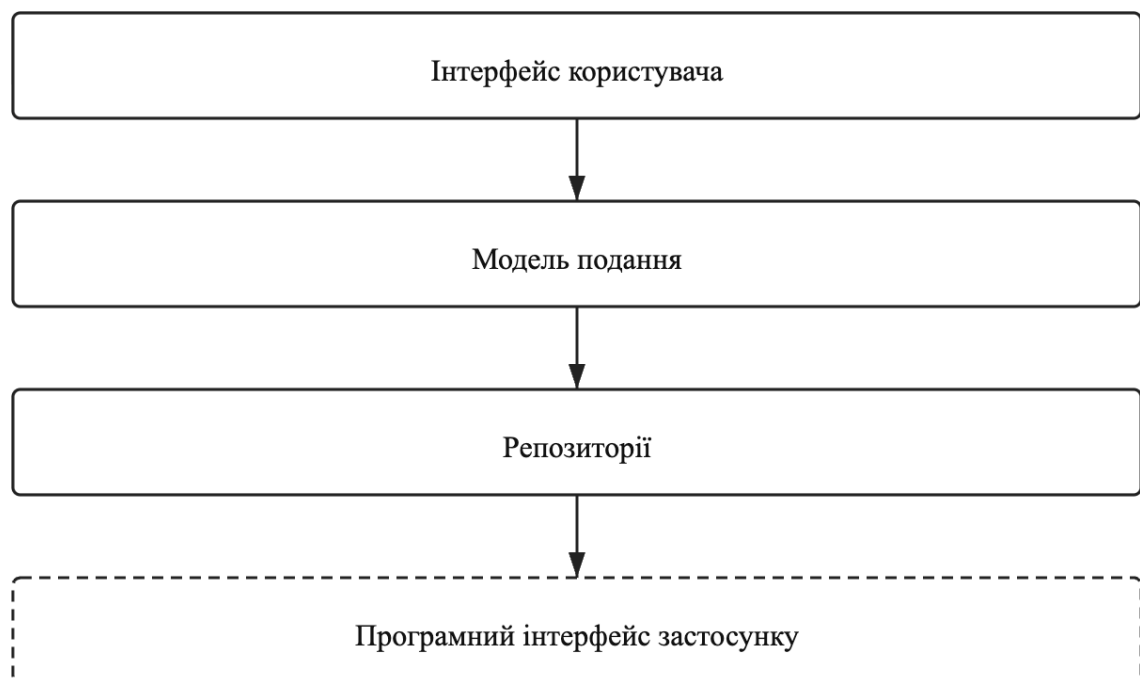


Рис. 2.5. Багат шарова архітектура мобільного застосунку

На рівні представлення знаходиться інтерфейс користувача, який відповідає за відображення даних і взаємодію з користувачем. Нижче розташовано модель подання, що керує станом екранів і координує бізнес-логіку. Доменний рівень інкапсулює правила предметної області, включаючи сценарії

автентифікації та перевірки даних. Доступ до серверної частини здійснюється через репозиторії, які формують запити до API та перетворюють відповіді у внутрішні моделі застосунку. Мережевий шар забезпечує HTTP-взаємодію, додавання токенів авторизації та обробку помилок. Локальне сховище використовується для збереження сесійних даних та налаштувань користувача. Платформозалежний модуль реалізує функції, що залежать від операційної системи, зокрема оплати через зовнішні сервіси, роботу з камерою, календарем та системою поширення контенту. Для оптимізації роботи зі списками подій у застосунку реалізовано механізм пагінації та поступового довантаження даних під час прокручування сторінок (Додаток В). Узагальнена структура компонентів мобільного застосунку наведена в табл. 2.30.

Таблиця 2.30

Компоненти архітектури мобільного застосунку

Компонент	Призначення	Що входить до складу
Інтерфейс користувача	відображення даних і прийом дій користувача	екрани, елементи інтерфейсу, навігація між екранами
Модель подання	управління станом екрана	окрема модель подання для кожного екрана
Доменна логіка	правила, не залежні від інтерфейсу	сценарії реєстрації та входу, перевірка облікових даних
Репозиторії	доступ до даних для моделі подання	звернення до сервера, перетворення відповідей

продовж. табл. 2.30

Мережевий шар	обмін даними з сервером	формування запитів, передача токена, обробка помилок
Локальне сховище	збереження даних на пристрої	токени сесії, окремі налаштування користувача
Платформозалежний модуль	можливості операційної системи	оплата через зовнішній сервіс, вибір зображення, календар, поширення

Застосунок реалізує ієрархічну навігацію, яка поділяється на дві основні зони: доавтентифікаційну та основну користувацьку. Доавтентифікаційна зона включає екрани входу, реєстрації та відновлення доступу. Після успішної автентифікації користувач переходить до основної частини системи з вкладковою навігацією. Структура навігації мобільного застосунку наведена на рис. 2.6, де показано розподіл на зони, вкладки та ієрархію переходів між екранами (Додаток Г). Така організація навігації забезпечує логічний розподіл функціональності та спрощує доступ користувача до основних можливостей системи. Використання вкладкової структури дозволяє швидко перемикатися між ключовими розділами застосунку без втрати поточного контексту роботи. Ієрархічні переходи між екранами забезпечують зручну навігацію до детальної інформації про події, профіль користувача та інші функціональні модулі системи. У разі виходу з облікового запису система автоматично скидає стек навігації та повертає користувача до початкового екрана авторизації. Такий підхід гарантує коректне керування станом застосунку та оптимізує використання ресурсів мобільного пристрою.

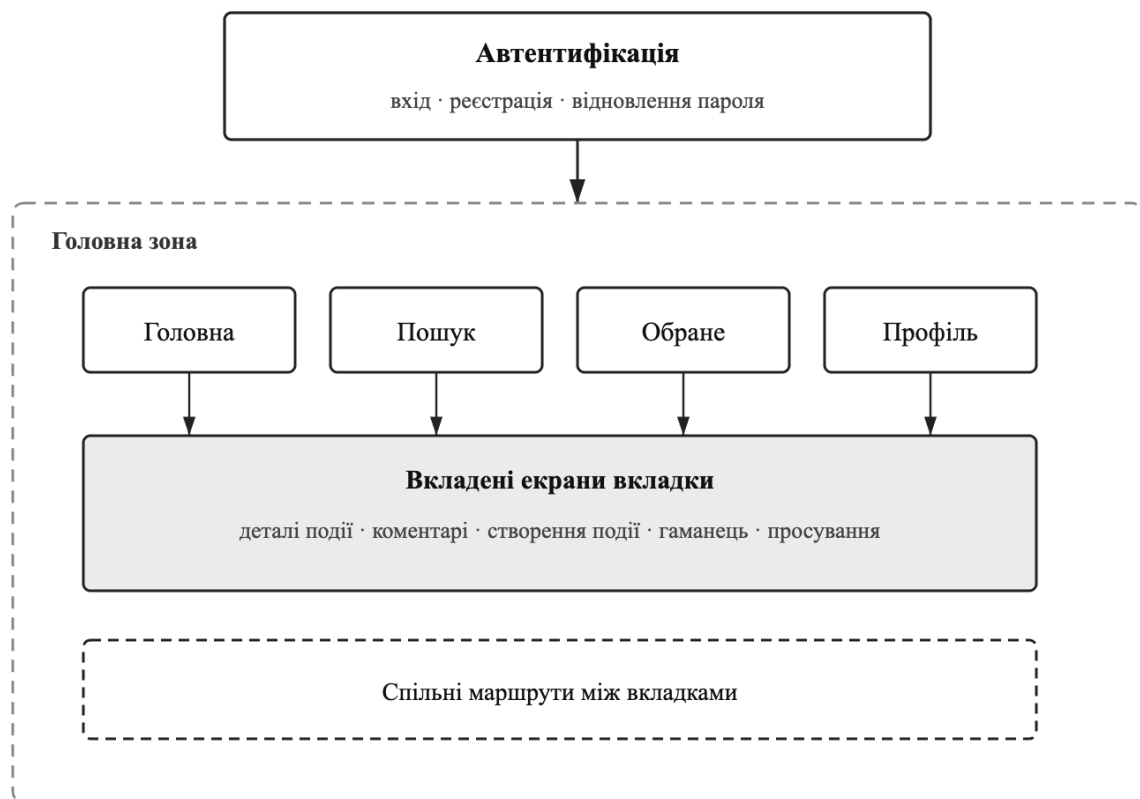


Рис. 2.6. Структура навігації мобільного застосунку

Основна зона містить чотири ключові розділи: головна сторінка, пошук подій, обрані заходи та профіль користувача. Кожен розділ має власний стек навігації, що дозволяє зберігати стан екранів при переходах між функціональними модулями. Такий підхід забезпечує зручність користувацької взаємодії та зменшує кількість повторних завантажень даних.

Серверна підсистема інтегрується з мобільним застосунком через REST API, що забезпечує уніфікований формат обміну даними. Усі запити виконуються асинхронно, а відповіді передаються у форматі JSON. Для захищених операцій використовується токенна автентифікація, що дозволяє контролювати доступ до ресурсів системи.

Вебклієнт профілю користувача реалізує додатковий спосіб взаємодії з системою через браузер. Він дублює основні можливості мобільного застосунку в частині керування профілем, налаштуваннями, фінансовими операціями та переглядом статистики. Такий підхід забезпечує кросплатформену доступність системи без прив'язки до конкретного пристрою.

Клієнтська частина вебзастосунку взаємодіє з сервером за тим самим API, що і мобільний застосунок, що гарантує єдину логіку обробки даних. Інтерфейс побудовано як односторінковий застосунок [30], де стан компонентів синхронізується через централізоване керування даними.

Для забезпечення фінансових операцій у системі реалізовано інтеграцію з міжнародним платіжним сервісом Stripe [31]. Процес оплати платного квитка побудовано за моделлю перенаправлення користувача на захищену сторінку оплати (Stripe Checkout), що забезпечує високий рівень безпеки даних платіжних карток, оскільки обробка чутливої інформації відбувається безпосередньо на стороні платіжного провайдера.

Висновки до розділу 2

У другому розділі було виконано проектування мультиплатформеного програмного комплексу, що охоплює серверну підсистему, мобільний застосунок та вебклієнт. На основі функціональної моделі визначено основні варіанти використання системи, які описують повний цикл взаємодії користувача з подіями, включаючи реєстрацію, оплату, створення заходів та керування профілем.

Розроблено інформаційну модель системи, що базується на реляційній базі даних PostgreSQL, із забезпеченням нормалізації даних та визначенням основних сутностей і зв'язків предметної області. Це дозволяє реалізувати ключові бізнес-процеси системи без дублювання інформації та з дотриманням цілісності даних. Окремо спроектовано та описано функціональні алгоритми роботи системи, включаючи автентифікацію користувачів, обробку подій, платежів та формування електронних квитків. Встановлено, що серверна підсистема виконує централізовану бізнес-логіку, тоді як клієнтські застосунки працюють через уніфікований REST API та обмін даними у форматі JSON.

Важливим результатом є розроблення мобільного застосунку для Android та iOS, архітектура якого побудована за багатошаровим принципом. Для

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

реалізації інтерфейсу використано підхід Jetpack Compose Multiplatform, що забезпечує єдину декларативну UI-логіку для різних платформ і підвищує повторне використання компонентів.

Також реалізовано вебклієнт, який доповнює систему та забезпечує доступ до функціональності через браузер, зокрема для керування профілем, фінансовими операціями та перегляду статистики. Усі клієнтські застосунки взаємодіють із сервером через єдиний API, що гарантує узгодженість даних у межах всієї системи.

У результаті сформовано цілісну архітектуру мультиплатформеного програмного комплексу, яка забезпечує масштабованість, підтримуваність та кросплатформену доступність, а також здійснено його реалізацію у вигляді серверної підсистеми, мобільного застосунку та вебклієнта.

Серверна підсистема спроектована як модульний моноліт із REST API: окремі модулі відповідають за облікові записи, події, квитки, платежі, гаманець, просування та сповіщення. Визначено автентифікацію на основі JSON Web Token, взаємодію із зовнішнім платіжним сервісом і сценарії, що потребують узгодженої роботи кількох модулів, зокрема реєстрації на подію з формуванням електронного квитка. Веб-інтерфейс профілю користувача використовує той самий програмний інтерфейс, що й мобільний застосунок.

Мобільний застосунок для Android та iOS спроектовано на базі Kotlin Multiplatform і шаблону MVVM. Визначено багатошарову архітектуру, склад репозитаріїв, відповідність функціональних блоків модулям серверної підсистеми та ієрархію навігації з зоною автентифікації і головною зоною з чотирма вкладками. Застосунок взаємодіє з сервером лише через програмний інтерфейс застосунку; локально зберігаються переважно параметри сесії та окремі налаштування.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З МУЛЬТИПЛАТФОРМЕНИМ КОМПЛЕКСОМ

3.1 Структура інтерфейсу мультиплатформеного комплексу

Структура інтерфейсу мультиплатформеного програмного комплексу для організації розважальних заходів побудована з урахуванням принципів зручності використання, логічного групування функціональних можливостей та забезпечення швидкого доступу до основних сценаріїв взаємодії користувача із системою. Інтерфейс орієнтований на різні категорії користувачів, зокрема звичайних учасників подій та організаторів, і забезпечує для кожної ролі відповідний набір екранів та функцій.

Перший сценарій взаємодії користувача із системою — це реєстрація та вхід. На екрані автентифікації користувач вводить електронну пошту та пароль для доступу до облікового запису. Також передбачено можливість створення нового акаунта для нових користувачів та відновлення доступу до системи у випадку втрати пароля через електронну пошту. Інтерфейс цього етапу є максимально простим та орієнтованим на швидке виконання базових дій користувача. Він забезпечує зрозумілу структуру введення даних без зайвих елементів, що знижує ймовірність помилок. Додатково передбачено логічний поділ між входом і реєстрацією, що спрощує первинну взаємодію з системою. Такий підхід підвищує загальну зручність використання та швидкість доступу до функціоналу. Після успішного входу користувач потрапляє до головного інтерфейсу застосунку, який містить нижню панель навігації для доступу до основних розділів системи. Така структура забезпечує швидкий перехід між функціональними модулями та зручну навігацію в межах застосунку. Екрани входу та реєстрації наведено на рис. 3.1. Інтерфейс цього етапу є базовою точкою входу до всієї системи та визначає подальший користувацький сценарій взаємодії із платформою.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

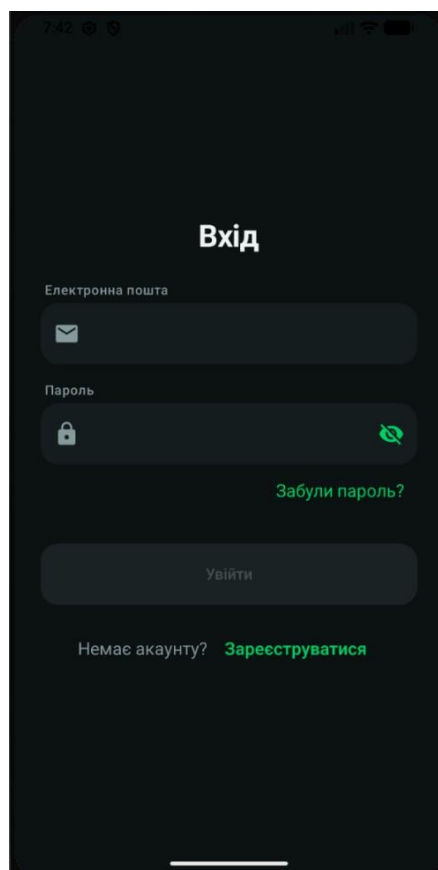


Рис. 3.1. Екрани входу в мобільному застосунку

Основний інтерфейс організовано через нижню панель з чотирма вкладками: головна, пошук, обране та профіль. Кожна вкладка відповідає окремому функціональному розділу системи та забезпечує доступ до ключових сценаріїв взаємодії користувача із застосунком. Перехід між вкладками відбувається миттєво та не порушує загальний стан інтерфейсу, що забезпечує безперервність користувацького досвіду.

На вкладці «Головна» відображається основна стрічка подій, яка формує персоналізоване уявлення про доступні та рекомендовані заходи. Вміст цієї вкладки структуровано у вигляді окремих інформаційних блоків, що дозволяє користувачу швидко орієнтуватися у великій кількості контенту. Окремі секції можуть бути згруповані за різними критеріями, такими як популярність або актуальність подій.

Користувач має можливість переходити до детальної інформації про події, переглядати категорії та взаємодіяти з організаторами безпосередньо зі стрічки. Також передбачено швидкий доступ до додаткових розділів, пов'язаних із

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

сповіщеннями та персональними рекомендаціями. Інтерфейс головної сторінки наведено на рис. 3.2. Така організація забезпечує зручну навігацію та швидкий доступ до найбільш важливої інформації.

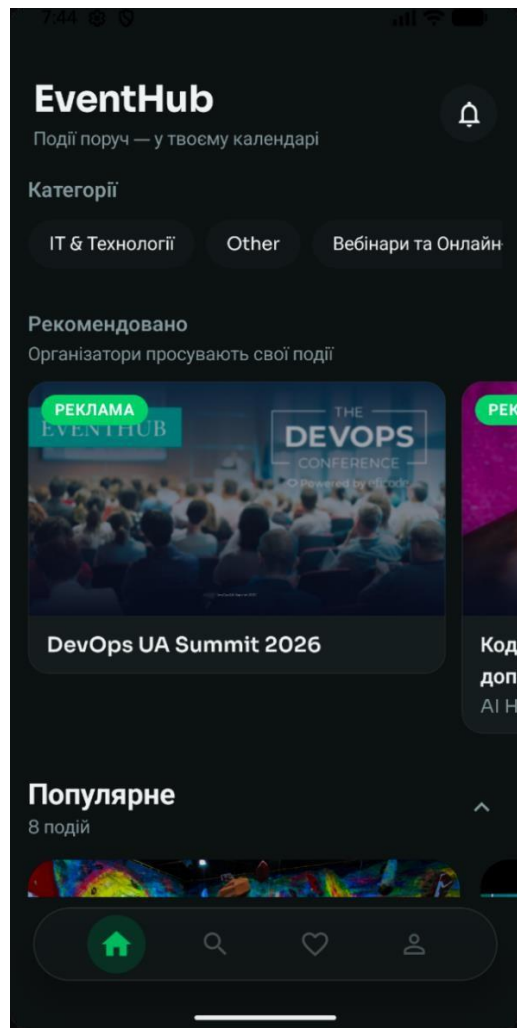


Рис. 3.2. Головний екран мобільному застосунку

На вкладці «Пошук» реалізовано інтерфейс для знаходження подій за заданими критеріями. Користувач може вводити текстовий запит та застосовувати різноманітні фільтри для уточнення результатів пошуку. Серед доступних параметрів передбачено вибір міст, категорій подій, а також часових діапазонів, таких як сьогодні, завтра, поточний тиждень або довільний період.

Результати пошуку відображаються у вигляді списку подій, який оновлюється відповідно до змін параметрів фільтрації. Інтерфейс підтримує поступове завантаження додаткових результатів під час прокручування списку, що дозволяє працювати з великим обсягом даних без перевантаження екрана.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

У випадку відсутності результатів передбачено відображення відповідного повідомлення для користувача, що покращує зрозумілість взаємодії із системою. Екран пошуку наведено на рис. 3.3. Такий підхід забезпечує зручну та швидку навігацію серед великої кількості подій і підвищує ефективність пошукового процесу.

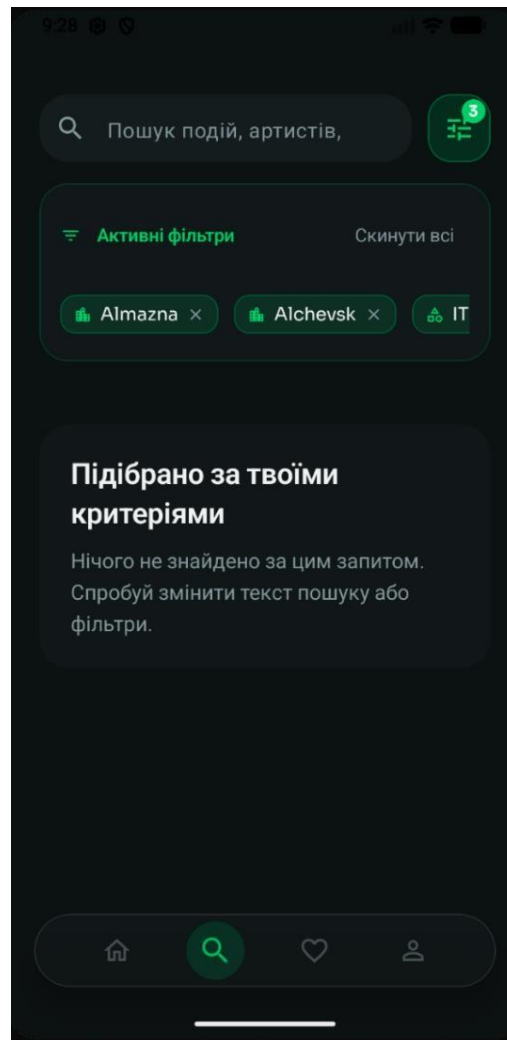


Рис. 3.3. Пошук та фільтрація подій у мобільному застосунку

Вкладка «Обране» призначена для відображення подій, у яких користувач вже бере участь або на які він записався. Вона забезпечує швидкий доступ до персонального списку майбутніх заходів, що дозволяє користувачу легко орієнтуватися у власному розкладі. Інтерфейс цього розділу орієнтований на зручний перегляд основної інформації про події без зайвих дій.

Кожна подія у списку містить ключові відомості, такі як дата, час проведення та місце, що дозволяє швидко оцінити деталі майбутнього заходу.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

Оновлення списку виконується в інтерактивному режимі, що забезпечує актуальність відображуваних даних.

Додатково користувач може позначати події як вподобані або прибирати їх зі списку вподобань безпосередньо з екрана деталей події. Екран обраних подій наведено на рис. 3.4. Такий підхід дозволяє розмежувати участь у подіях та особисті вподобання, що робить структуру інтерфейсу більш логічною та зручною для використання.

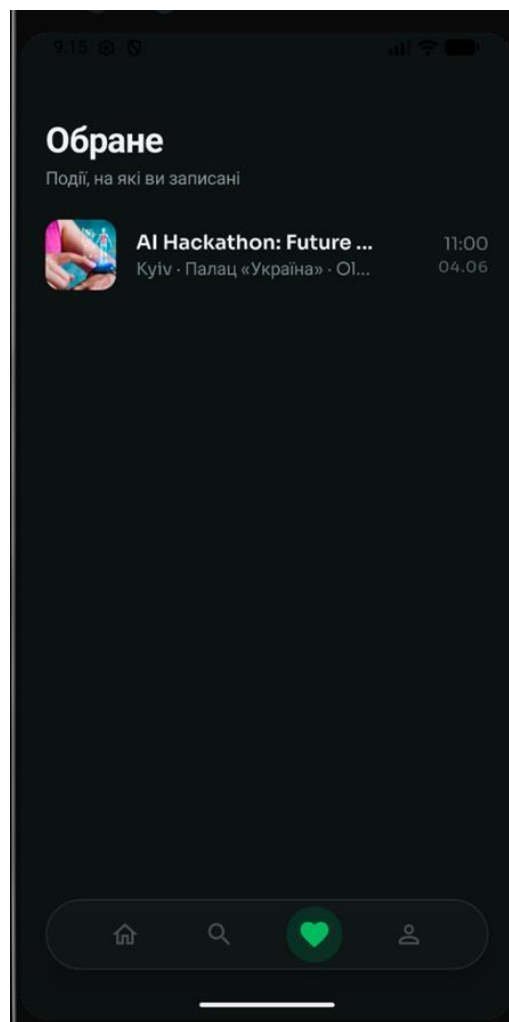


Рис. 3.4. Список подій, на які користувач записаний, у мобільному застосунку

Екран деталей події відображає повну інформацію про захід, включно з описом, датою, часом, місцем проведення та статусом реєстрації поточного користувача. Тут також доступні опції перегляду додаткової інформації, зокрема відгуків, а для завершених подій — можливість їх оцінювання. Інтерфейс дозволяє користувачу оцінити подію та прийняти рішення щодо участі.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Користувач може виконувати основні дії: зареєструватися на безкоштовну подію, ініціювати оплату платного квитка, скасувати участь, взаємодіяти з подією через вподобання, а також переходити до коментарів і списку учасників. Також передбачено можливість додавання події до системного календаря пристрою для зручного планування.

Для платних заходів реалізовано процес переходу до зовнішнього платіжного сервісу з подальшою обробкою результату оплати та оновленням статусу події. Екран деталей події наведено на рис. 3.5.

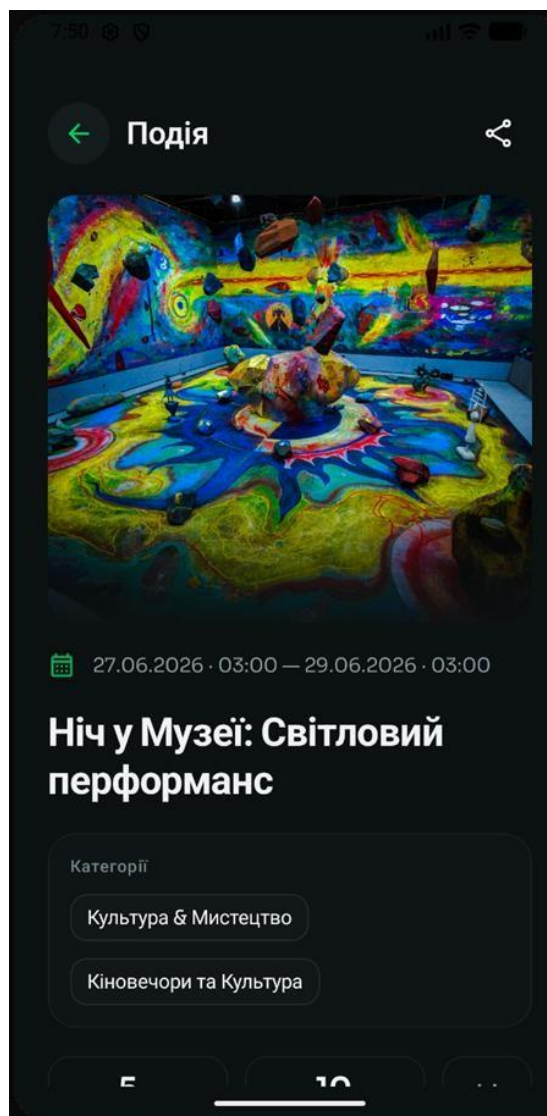


Рис. 3.5. Екран деталей події в мобільному застосунку

Оплата платного квитка ініціюється з екрана деталей події. Користувач переходить на сторінку зовнішнього платіжного сервісу; після повернення

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

застосунок оновлює статус реєстрації, і користувач отримує доступ до електронного квитка на сервері. Екран оплати наведено на рис. 3.6.

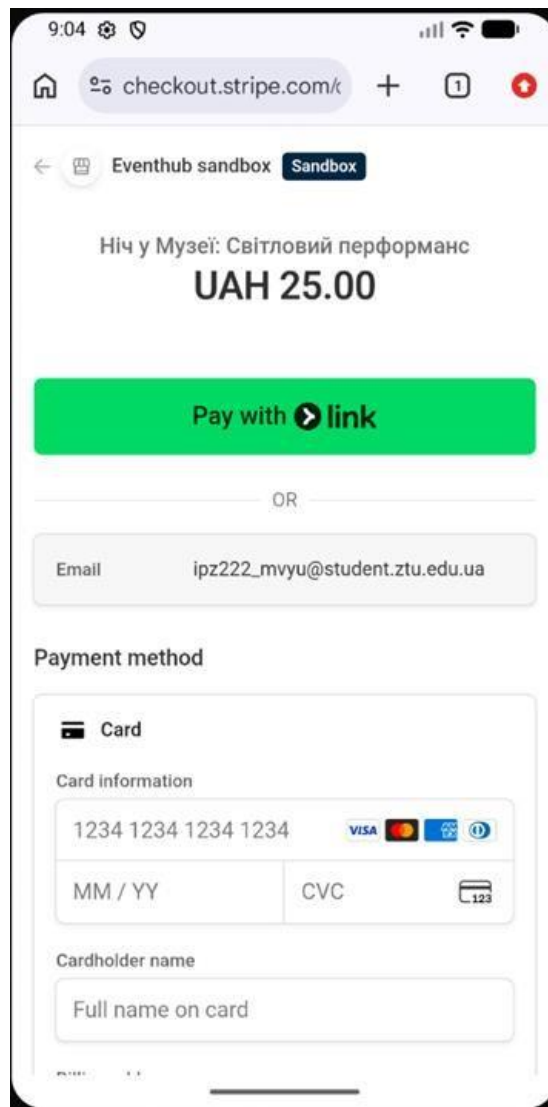


Рис. 3.6. Оплата квитка в мобільному застосунку

Після успішної реєстрації на подію або оплати платного квитка серверна підсистема автоматично створює запис електронного квитка, генерує QR-код із посиланням на сторінку квитка та зберігає зображення коду. У мобільному застосунку користувач отримує доступ до квитків через розділ «Мої квитки» в налаштуваннях профілю; також після підтвердження участі на екрані деталей події доступна дія «Показати мій квиток», яка відкриває відповідний запис у списку. Інтерфейс оформлено у вигляді цифрової картки з акцентом на QR-код, що відповідає сценарію контролю доступу на захід. Екран електронного квитка в мобільному застосунку наведено на рис. 3.7.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		66



Рис. 3.7. Електронний квиток із QR-кодом у мобільному застосунку

Створення та редагування подій здійснюється через окремий екран редактора події. Інтерфейс форми містить усі необхідні поля для заповнення інформації про захід: назву, опис, дату та час проведення, місце проведення, категорію, кількість доступних місць, обкладинку події та параметри квитків. Така структура дозволяє організатору зручно керувати всіма основними характеристиками заходу в межах одного екрана.

Для спрощення заповнення форми передбачено окремі вікна вибору міста та локації, а також можливість завантаження зображення для оформлення картки події. Крім того, в інтерфейс інтегровано інструменти штучного інтелекту, які допомагають автоматично генерувати опис події та рекламне зображення на основі короткого текстового запиту користувача. Отриманий результат можна переглянути та за потреби відредагувати перед збереженням.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

Екран редактора події наведено на рис. 3.8. Використання таких елементів інтерфейсу спрощує процес створення заходів та дозволяє організаторам швидше підготувати інформацію для публікації події на платформі.

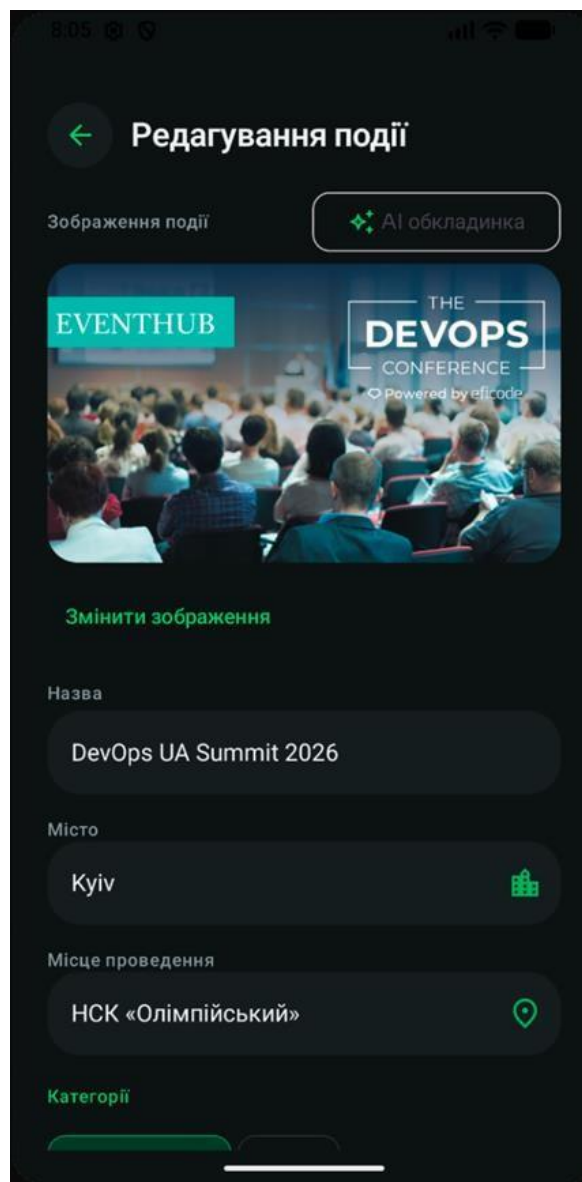


Рис. 3.8. Редактор події в мобільному застосунку

Розділ профілю об'єднує персональні дані користувача та додаткові функції, пов'язані з його активністю в системі. З цього розділу користувач отримує доступ до власного профілю, статистики, гаманця, календаря збережених подій, списку просувань, а також до сторінок налаштувань облікового запису. Окремо передбачено доступ до політики конфіденційності та умов використання платформи.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

На головному екрані профілю відображається основна інформація про користувача, зокрема аватар, ім'я та відомості про створені ним події. Інтерфейс також надає можливість редагувати особисті дані та керувати параметрами облікового запису через відповідні екрани налаштувань.

Екран профілю наведено на рис. 3.9. Така структура дозволяє зосередити всі персональні налаштування та інструменти керування обліковим записом в одному розділі, забезпечуючи зручний доступ до них під час використання застосунку.

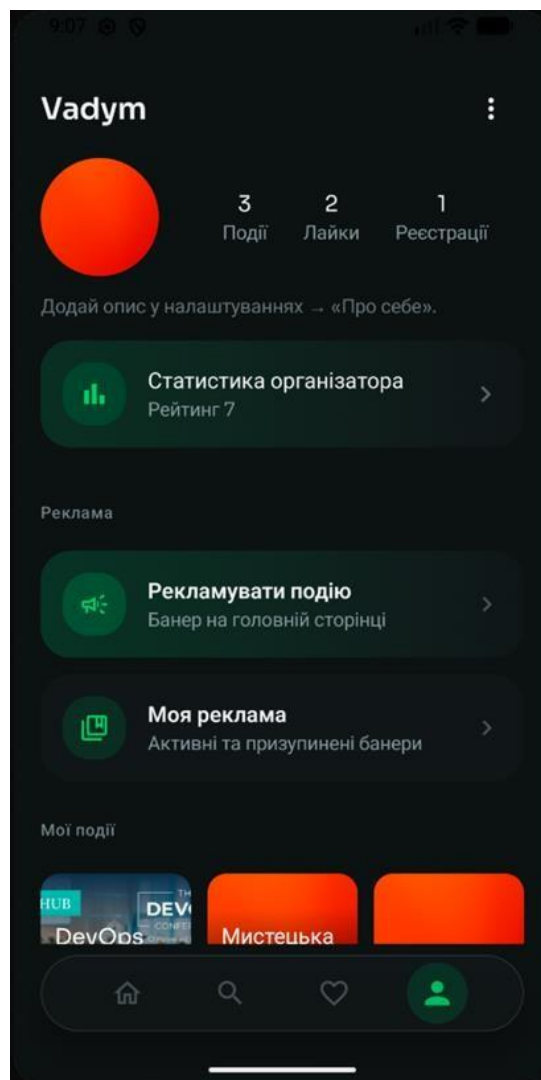


Рис. 3.9. Профіль користувача в мобільному застосунку

Розділ статистики призначений для відображення аналітичної інформації про діяльність організатора на платформі. На екрані представлено різноманітні показники, пов'язані зі створеними подіями, кількістю реєстрацій, активністю учасників та загальною популярністю заходів.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

Інформація відображається у зручному візуальному вигляді за допомогою інформаційних карток, графіків і діаграм, що дозволяє швидко оцінити результати діяльності та виявити найбільш успішні події. Також користувач може переглядати статистику за різними напрямками, зокрема активність аудиторії, популярні категорії та динаміку проведення заходів.

Екран статистики наведено на рис. 3.10. Такий інтерфейс надає організатору наочне уявлення про ефективність його діяльності та сприяє прийняттю обґрунтованих рішень щодо подальшого розвитку та просування власних заходів.

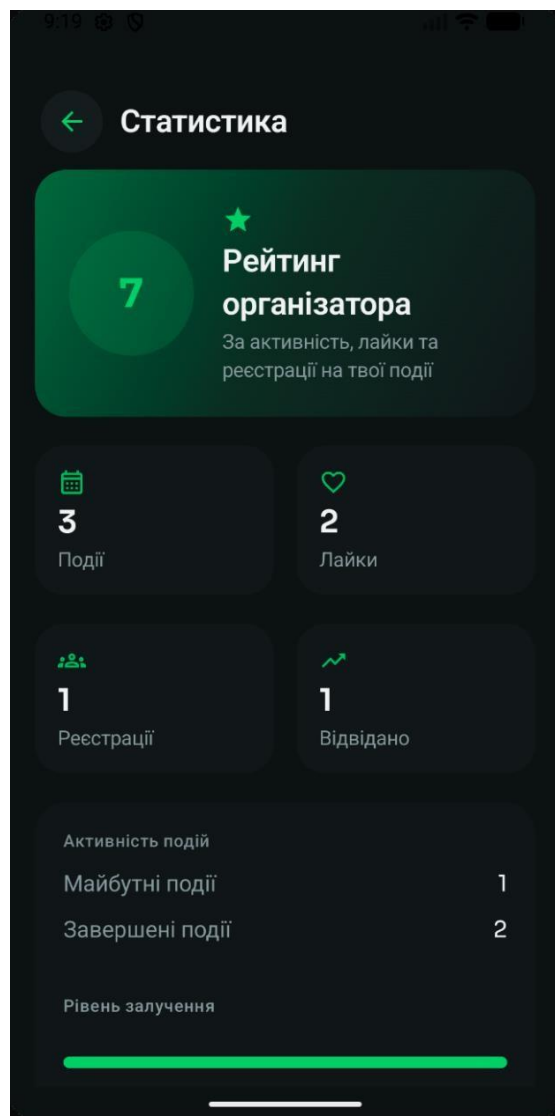


Рис. 3.10. Статистика користувача у мобільному застосунку

Розділ гаманця призначений для керування фінансовими операціями організатора безпосередньо в мобільному застосунку. На екрані відображаються

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

поточний баланс користувача, історія фінансових надходжень і витрат, а також інформація про подані заявки на виведення коштів. Інтерфейс забезпечує швидкий доступ до основних фінансових показників та дозволяє контролювати рух коштів у межах платформи. Усі дані подаються у зрозумілому вигляді, що спрощує аналіз доходів від проведених заходів та відстеження статусів фінансових операцій. Екран гаманця наведено на рис. 3.11.

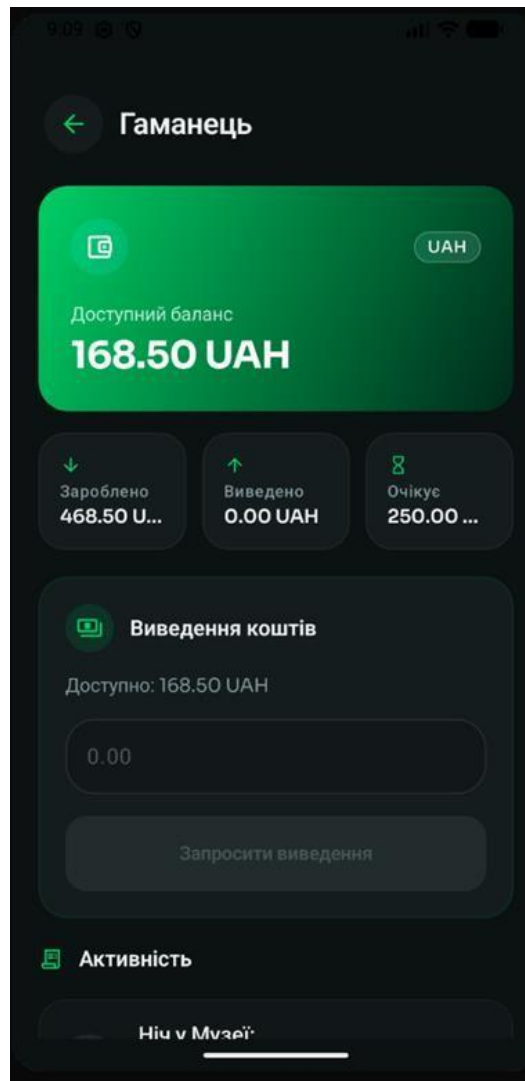


Рис. 3.11. Гаманець користувача в мобільному застосунку

Соціальна взаємодія в застосунку реалізована через екрани коментарів, відгуків та перегляду учасників події. Користувачі можуть обговорювати заходи, залишати власні коментарі та ознайомлюватися з думками інших учасників. Після завершення події доступна можливість залишити відгук та оцінити захід за п'ятибальною шкалою. Такий підхід сприяє формуванню активної спільноти користувачів, підвищує рівень взаємодії між учасниками та допомагає майбутнім

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

відвідувачам отримати додаткову інформацію про якість заходів. Інтерфейс коментарів і відгуків показано на рис. 3.12.

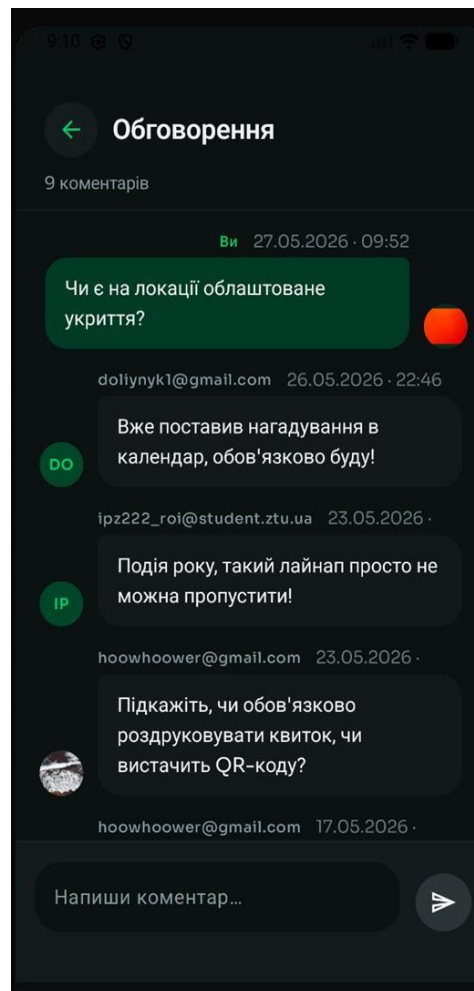


Рис. 3.12. Коментарі та відгуки в мобільному застосунку

Центр сповіщень призначений для інформування користувача про всі важливі події в системі, пов'язані з його активністю. У цьому розділі відображаються повідомлення про нові лайки, коментарі, реєстрації на події та інші системні дії. Сповіщення згруповані за часовими категоріями, що дозволяє швидко орієнтуватися в їхній актуальності. Також передбачено можливість перегляду лише непрочитаних повідомлень для швидкого доступу до нової інформації. Інтерфейс сповіщень наведено на рис. 3.13.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		72

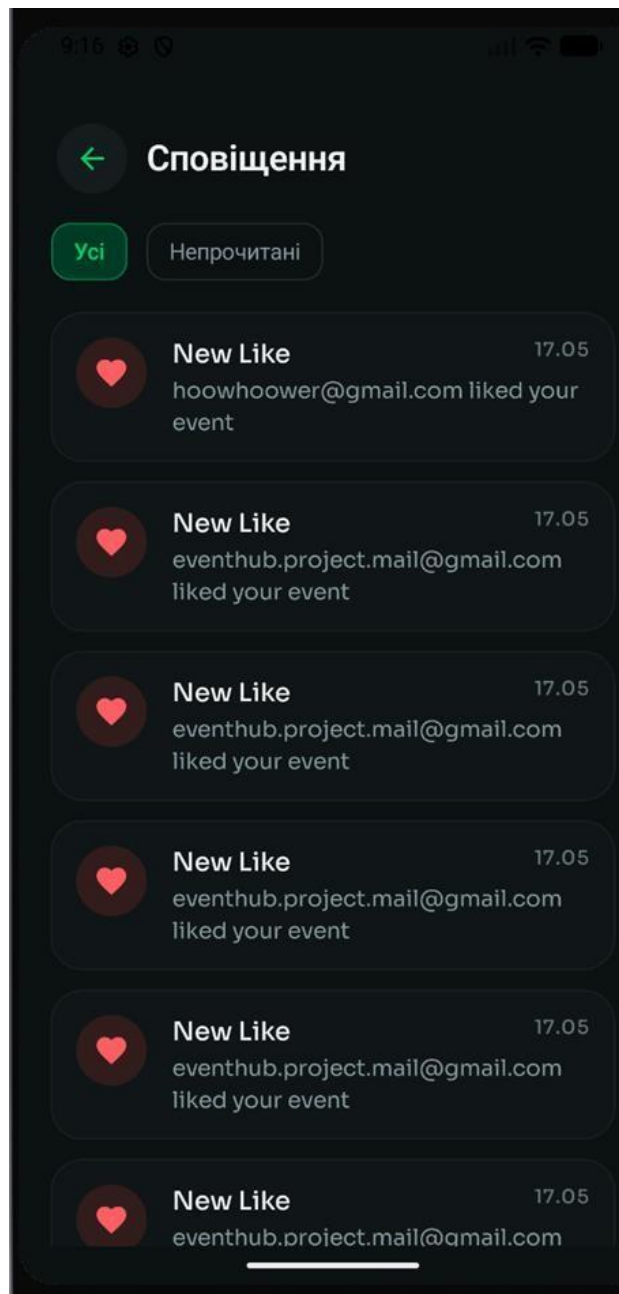


Рис. 3.13. Центр сповіщень в мобільному застосунку

Розділ просування подій призначений для підвищення видимості створених користувачем заходів у межах платформи. У цьому інтерфейсі користувач може переглядати власні опубліковані події та обирати ті, які планується просувати. Також відображається інформація про доступні варіанти просування та їхні умови. Окрему частину екрана займає попередній перегляд того, як рекламний блок буде виглядати в стрічці подій.

Система дозволяє ініціювати процес просування безпосередньо з цього розділу інтерфейсу, забезпечуючи інтегровану взаємодію з платіжними

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

механізмами. Після завершення процесу відповідний статус відображається у списку активних просувань користувача. Така організація інтерфейсу забезпечує зручне керування рекламними кампаніями в межах платформи. Екран просування наведено на рис. 3.14.

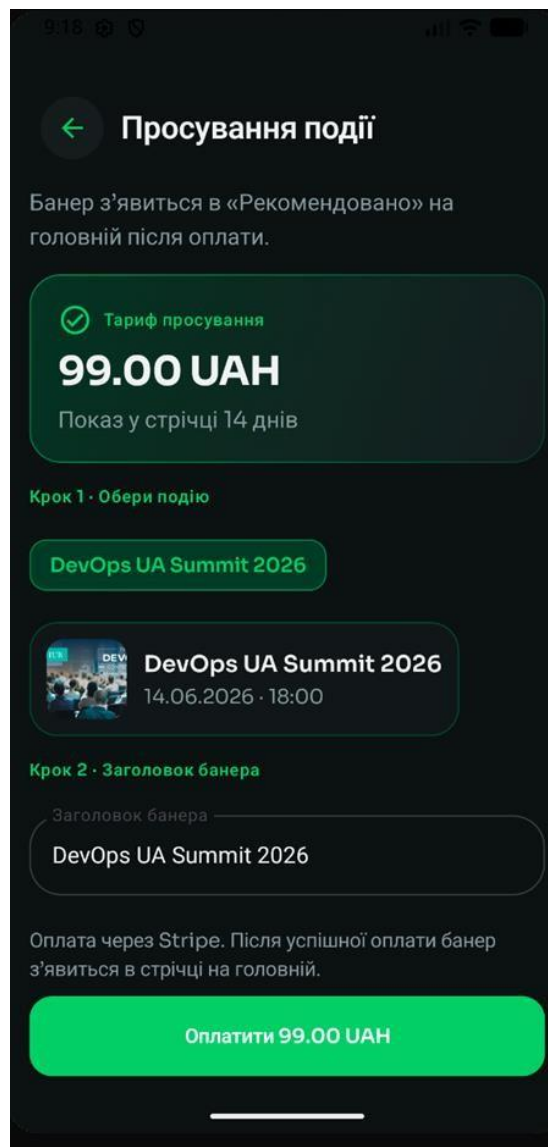


Рис. 3.14. Просування події в мобільному застосунку

З екрана деталей події користувач має можливість перейти до профілю організатора. У цьому розділі відображається базова інформація про автора, включно з ім'ям та коротким описом діяльності. Також користувач може ознайомитися зі списком подій, які були створені цим організатором, не покидаючи межі застосунку. Це дозволяє отримати додатковий контекст про автора подій перед прийняттям рішення щодо участі. Профіль організатора показано на рис. 3.15.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		74

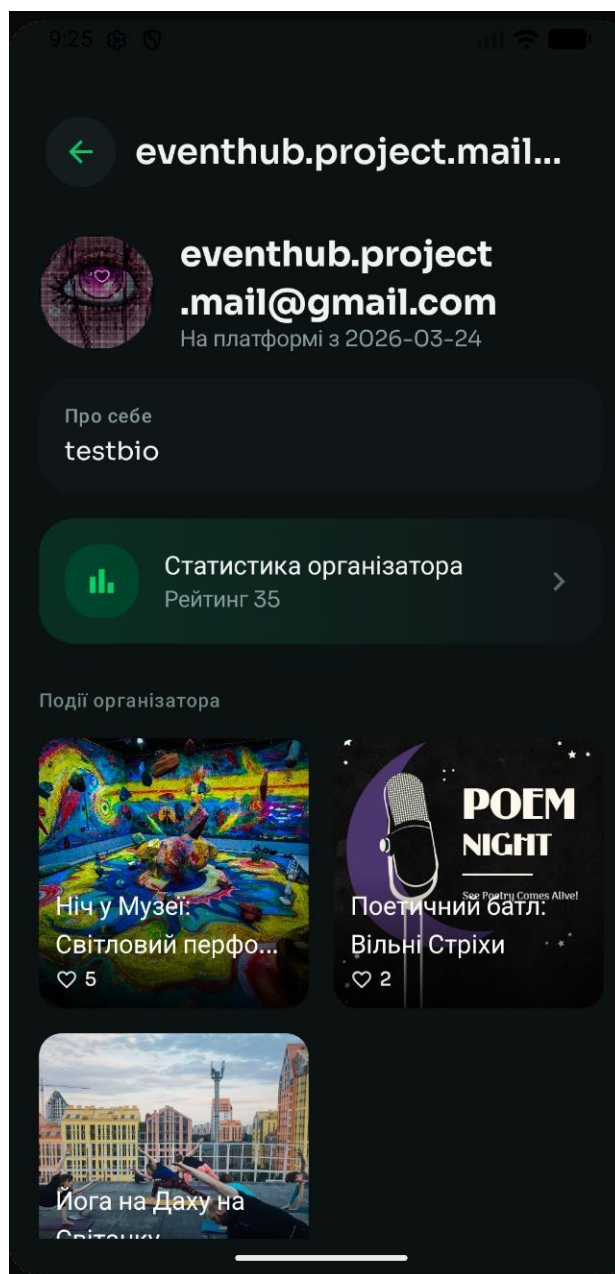


Рис. 3.15. Профіль організатора події в мобільному застосунку

Екран перегляду подій за категоріями забезпечує зручну навігацію та структурований перегляд заходів у межах обраної тематики, наприклад «ІТ & Технології». На цьому екрані користувач бачить список подій, згрупованих за категоріями, що дозволяє швидко переходити до цікавого контенту без перегляду загальної стрічки. Візуальне подання реалізовано у вигляді адаптивної сітки, де кожна подія представлена карткою з зображенням, назвою та додатковою короткою інформацією.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

Такий формат відображення забезпечує більш ефективний пошук релевантних заходів відповідно до інтересів користувача та спрощує навігацію в межах каталогу подій. Екран показано на рис. 3.16.

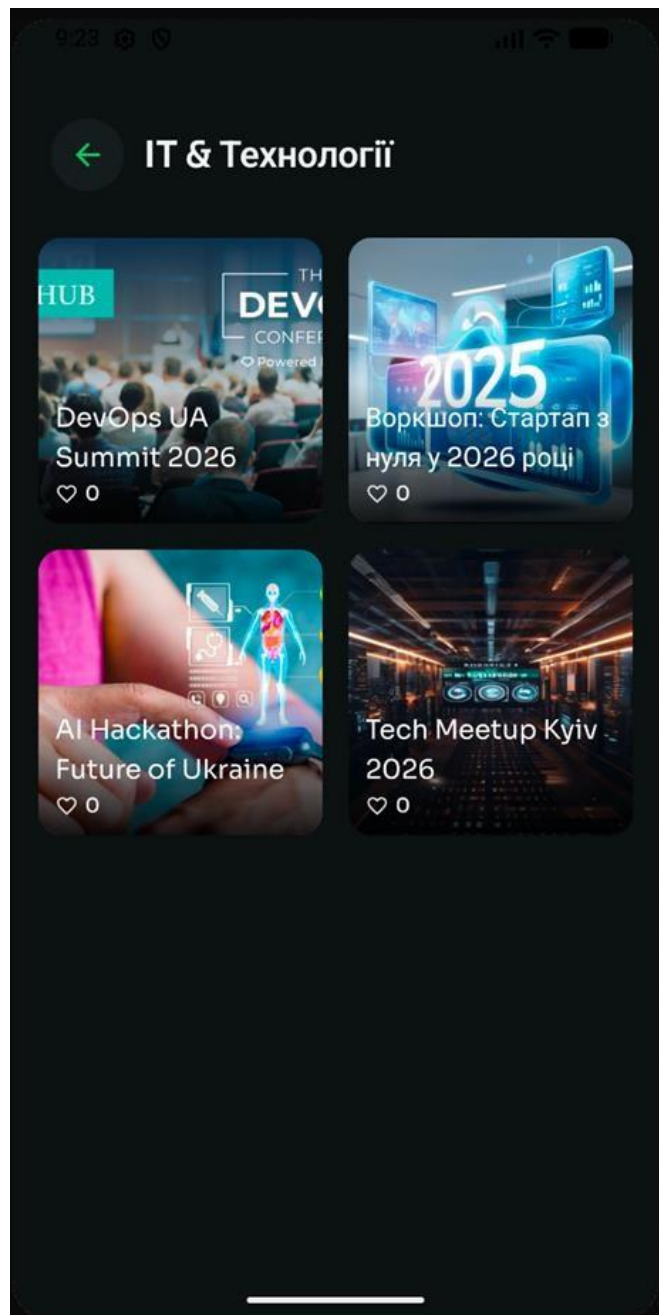


Рис. 3.16. Пошук і фільтрація подій у мобільному застосунку

Модуль профілю користувача у вебклієнті мультиплатформеного комплексу реалізовано як частину сучасного односторінкового застосунку з інтерактивним оновленням інтерфейсу без повного перезавантаження сторінки.

Вебінтерфейс забезпечує доступ до основних функцій профілю та доповнює мобільний застосунок, надаючи користувачу альтернативний спосіб

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

взаємодії з платформою через браузер. Завдяки єдиному підходу до роботи з даними всі зміни, виконані у вебклієнті, синхронізуються із серверною частиною системи.

Після входу в обліковий запис користувач відкриває особистий профіль. На екрані відображається аватар користувача, ім'я, короткий опис та базові показники активності організатора. Також доступний список створених подій, що дозволяє швидко переглядати власний контент. З цього розділу користувач може переходити до інших частин профілю, зокрема до налаштувань та статистики. Екран особистого профілю наведено на рис. 3.17.

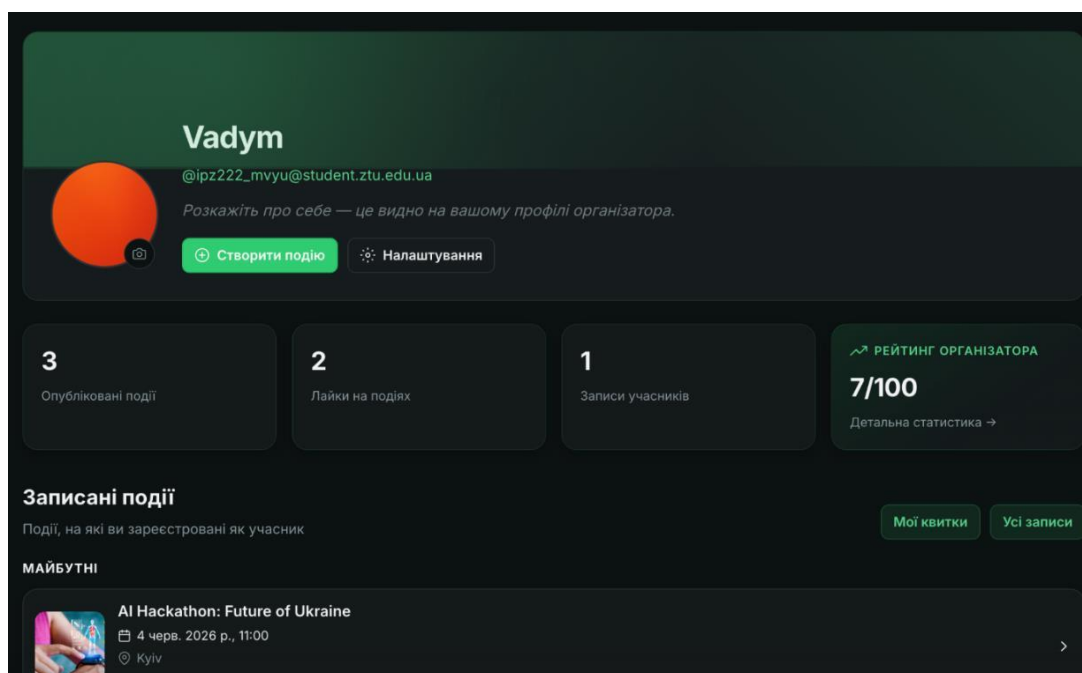


Рис. 3.17. Особистий профіль користувача

Сторінка налаштувань об'єднує основні функції керування обліковим записом у логічно згруповані розділи, що забезпечує зручну навігацію та швидкий доступ до необхідних параметрів. Інтерфейс поділений на блоки, пов'язані з даними профілю, безпекою, активністю користувача та додатковою інформацією про використання платформи. Кожен елемент меню відкриває окремий екран для перегляду або редагування відповідних даних. Така структура дозволяє користувачу швидко орієнтуватися у великій кількості параметрів без перевантаження інтерфейсу.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

Окремо в інтерфейсі передбачено доступ до ключових функцій системи, пов'язаних із профілем користувача та його активністю. Меню налаштувань показано на рис. 3.18.

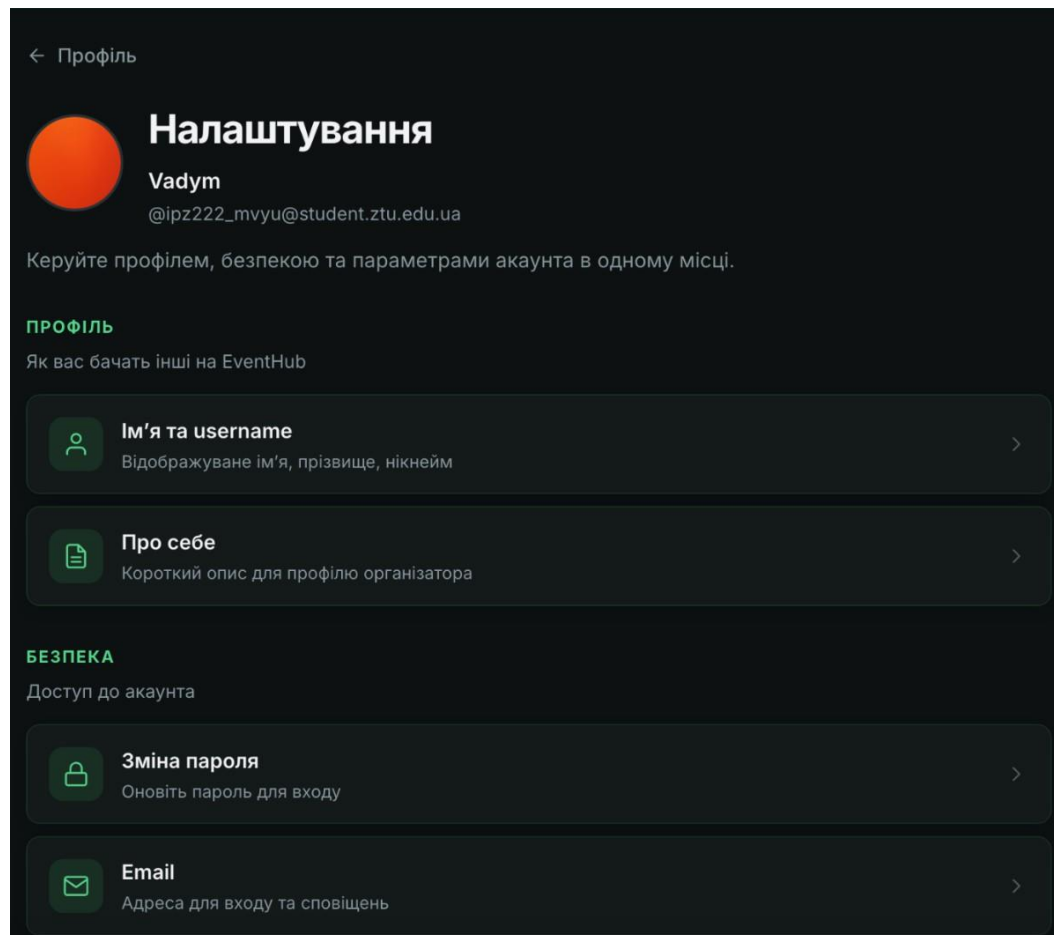


Рис. 3.18. Налаштування профілю користувача

На сторінці зміни імені користувач має можливість переглянути та редагувати основні дані профілю, зокрема ім'я, прізвище та нікнейм. Інтерфейс побудований таким чином, щоб забезпечити просте та зрозуміле внесення змін без зайвих дій. У верхній частині екрана розміщено блок попереднього перегляду, де відображаються поточні або вже змінені дані користувача. Це дозволяє одразу бачити результат редагування перед збереженням.

Кожне поле введення має зручний формат для редагування текстової інформації, а для нікнейму передбачено візуальний контроль довжини. Унікальний ідентифікатор користувача відображається з префіксом @ для кращої ідентифікації. Після внесення змін користувач може або зберегти

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

оновлені дані, або скасувати редагування без застосування змін. Сторінку зміни імені наведено на рис. 3.19.

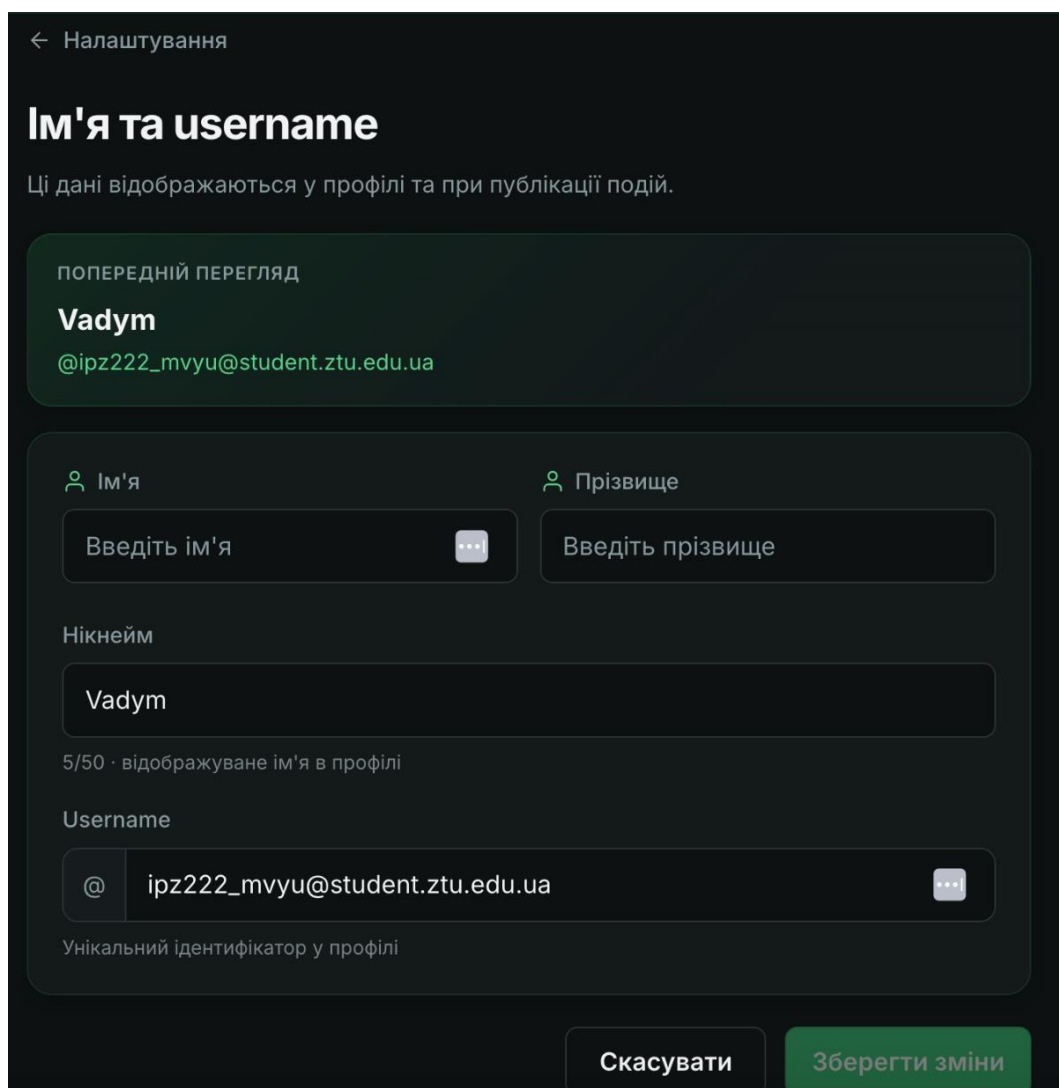


Рис. 3.19. Редагування імені користувача

Сторінка «Про себе» призначена для короткого опису організатора. Користувач вводить текст у поле з обмеженням довжини; поруч показано попередній перегляд того, як опис виглядатиме на профілі. Сторінку редагування опису наведено на рис. 3.20. У нижньому правому куті поля введення розміщено динамічний лічильник, який відображає кількість використаних символів з максимально доступних 500. Кнопки «Скасувати» та «Зберегти зміни» забезпечують зручне керування процесом редагування та фіксацію оновленої інформації в профілі.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				79
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.20. Сторінка редагування опису профілю

Зміна пароля виконується на окремій сторінці з полями поточного, нового та підтвердження пароля. Інтерфейс показує вимоги до довжини пароля та дозволяє переглянути введені символи. Сторінку зміни пароля наведено на рис. 3.21. У верхній частині сторінки розміщено блок рекомендацій щодо безпеки, який допомагає користувачеві створити надійний пароль. Поруч із кожним полем введення розташована кнопка «Показати», що надає можливість перевірити правильність написання символів, а нижня панель містить кнопки для збереження змін або відмови від дії. Така структура екрана мінімізує ризик помилок під час введення нових даних та забезпечує швидкий доступ до необхідних функцій керування акаунтом. Додатково виділена зона з правилами безпеки сприяє дотриманню користувачем стандартів захисту облікового запису від несанкціонованого доступу.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		80

Рис. 3.21. Сторінка зміни пароля

На сторінці зміни електронної пошти відображається поточна адреса; користувач вводить нову та підтверджує її повторним введенням. Перед збереженням перевіряється коректність формату адреси. Сторінку зміни email наведено на рис. 3.22. У верхній частині екрана передбачено блок із рекомендаціями щодо безпеки, який чітко окреслює вимоги до надійності пароля, включаючи мінімальну кількість символів та використання комбінації літер і цифр. Кожне з трьох полів введення обладнане функціональною кнопкою «Показати», що дозволяє користувачеві активувати відображення прихованих символів для перевірки правильності вводу. Такий підхід мінімізує ризик помилок під час набору складних паролів та сприяє дотриманню правил інформаційної безпеки. У нижній частині інтерфейсу розміщено кнопки «Скасувати» та «Оновити пароль», які дозволяють або відмовитися від операції, або надіслати запит на оновлення облікових даних на сервер. Ця сторінка реалізована з урахуванням принципів зручності взаємодії, що дозволяє

користувачеві швидко та без зайвих зусиль змінити пароль у межах системи керування акаунтом.

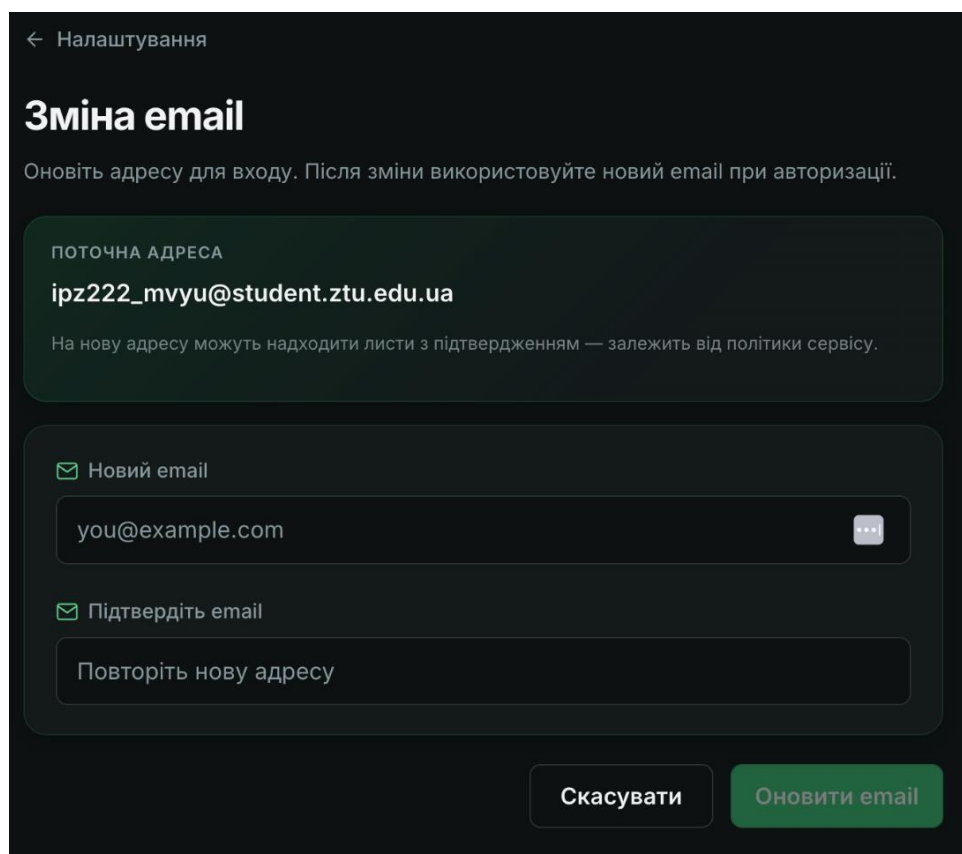


Рис. 3.22. Сторінка зміни електронної пошти

Сторінка гаманця організатора містить дві вкладки: історію оплат за квитки та операції гаманця. На вкладці «Гаманець» показано баланс, перелік нарахунків і виведень, а також форму подання заявки на виведення коштів. Сторінку гаманця наведено на рис. 3.23. У верхній частині інтерфейсу реалізовано навігаційний елемент (таби), який дозволяє користувачеві швидко перемикатися між деталізованим списком оплат та станом рахунку гаманця. Для кожного запису в історії оплат відображається назва події, вартість квитка, дата та поточний статус транзакції, наприклад «Очікує» або «Повернуто». Таке візуальне відображення статусів за допомогою кольорових міток допомагає організатору миттєво оцінити фінансові результати кожного заходу. Завдяки структурованому представленню даних, сторінка забезпечує прозорий контроль над усіма грошовими потоками, що є критично важливим для професійного управління подіями.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

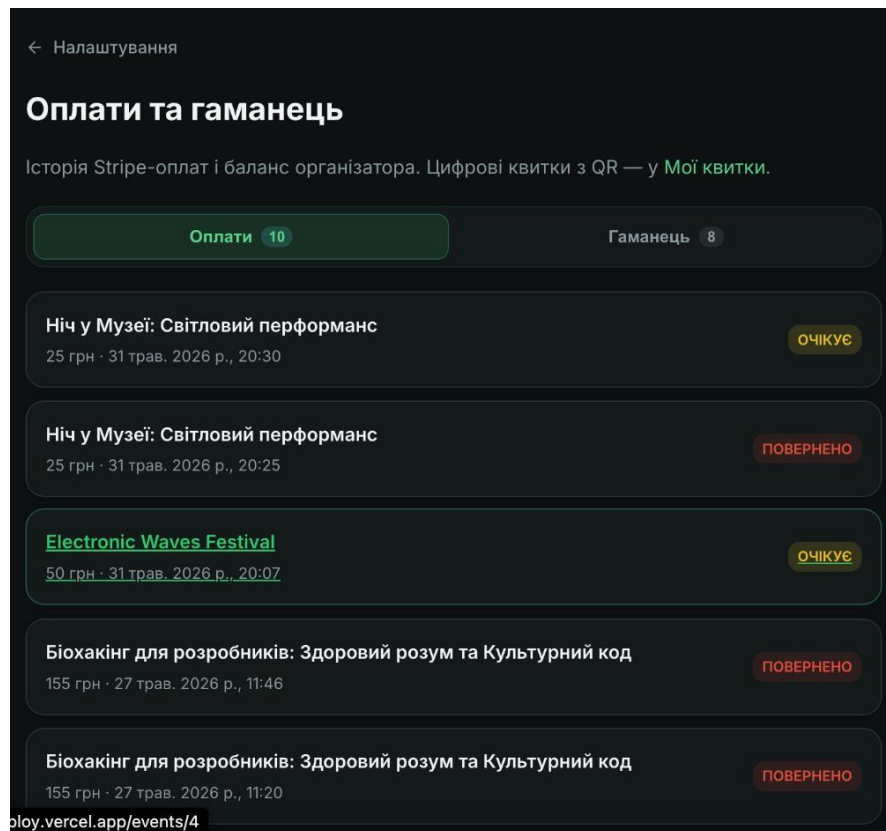


Рис. 3.23. Сторінка гаманця організатора

3.2 Тестування мультиплатформеного комплексу

Після завершення розробки мультиплатформеного програмного комплексу було проведено тестування основних функціональних можливостей системи. Метою тестування було перевірити коректність роботи мобільного застосунку для Android і iOS, модуля профілю веб-клієнта та їхньої взаємодії з серверною підсистемою через REST API, відповідність реалізованого функціоналу вимогам постановки задачі, а також стабільність обміну даними між клієнтськими і серверною частинами.

Особливу увагу приділено перевірці процесів авторизації та реєстрації користувачів, доступу до захищених розділів профілю, створенню та редагуванню подій, реєстрації на заходи й скасуванню участі. Окремо протестовано сценарії оплати квитків через платіжну систему Stripe, роботу внутрішнього гаманця організатора, автоматичне створення цифрових квитків із QR-кодом і їхню перевірку при вході на подію. Додатково перевірено перегляд

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

каталогу подій із фільтрацією та пошуком, детальну інформацію про події та організаторів, функціонал календаря, коментарів і відгуків,рекламування подій, сповіщення, а також генерацію опису та обкладинки події за допомогою AI.

Тестування виконувалось вручну на розгорнутому серверному середовищі з використанням мобільного застосунку на емуляторі. Android і симуляторі iOS, а також вебклієнта в браузері Google Chrome. Результати тестування авторизації, реєстрації та відновлення доступу наведено в табл. 3.1.

Таблиця 3.1

Тестування авторизації і базових сценарії

Операція	Умови використання	Очікуваний результат	Фактичний результат	Статус
Вхід у мобільний застосунок	Зареєстрований користувач, валідні email і пароль	Відкриття головного екрану застосунку	Після входу відображається головний екран Home зі списком подій	Пройдено
Вхід через вебінтерфейс	Зареєстрований користувач, валідні email і пароль	Відкриття профілю користувача	Відкривається профіль користувача	Пройдено
Багатокрокова реєстрація	Новий email, код із листа, пароль від 8 символів	Створення облікового запису	Акаунт створено, виконано автоматичний вхід	Пройдено
Відновлення доступу до акаунта	Email зареєстрованого користувача, код і новий пароль	Оновлення пароля на сервері	Вхід можливий лише з новим паролем	Пройдено

Результати тестування підтвердили коректну роботу механізмів автентифікації та управління обліковими записами користувачів. Усі перевірені сценарії виконуються без помилок, а система стабільно обробляє як мобільні, так і веб-запити до серверної підсистеми. Особливо важливо, що процес реєстрації та відновлення доступу працює узгоджено між усіма платформами, забезпечуючи синхронний стан користувача в межах єдиного акаунта. Це свідчить про правильну інтеграцію клієнтської частини з серверним API та коректну реалізацію логіки автентифікації.

Далі було проведено тестування основних сценаріїв взаємодії користувача з подіями, включаючи пошук, перегляд детальної інформації, додавання до обраного та реєстрацію на заходи. Результати цього етапу наведено в табл. 3.2.

Таблиця 3.2

Тестування сценаріїв роботи з подіями

Операція	Умови використання	Очікуваний результат	Фактичний результат	Статус
Пошук подій на вкладці Search в мобільному застосунку	Категорія, місто, текстовий запит	Список подій відфільтровано за заданими параметрами	Результати пошуку відповідають обраним фільтрам	Пройдено
Додавання події до обраного	Натискання іконки «лайк» на сторінці події	Подія з'являється на вкладці Favorites	Подія з'являється на вкладці Favorites	Пройдено
Перегляд деталей події	Відкриття події зі стрічки або пошуку	Показ назви, дати, локації, ціни, опису, організатора	Усі поля відповідають даним API	Пройдено

Запис на безкоштовний захід	Кнопка реєстрації на події з нульовою вартістю	Фіксація участі на сервері, зміна статусу на екрані	Користувач відображається серед учасників	Пройдено
-----------------------------------	--	---	---	----------

Отримані результати показали, що функціонал роботи з подіями реалізовано коректно та відповідає вимогам системи. Користувачі можуть швидко знаходити події за заданими параметрами, переглядати актуальну інформацію та взаємодіяти з контентом без затримок. Особливо важливо, що оновлення стану (наприклад, додавання в обране або реєстрація на подію) миттєво відображається в інтерфейсі, що підтверджує коректну роботу синхронізації між клієнтом і сервером.

Окремо було перевірено розширені функції системи, включаючи оплату квитків, генерацію цифрових квитків із QR-кодом, створення подій та використання AI-функцій. Результати тестування цього функціонального блоку наведено в табл. 3.3.

За результатами перевірок усі сценарії виконано успішно. Під час тестування перевірялась робота мобільного застосунку на Android і iOS, модуля профілю вебклієнта та їхня взаємодія з серверною підсистемою через REST API.

Окремо контролювалась синхронізація даних профілю, квитків і результатів оплати між вебклієнтом і мобільним застосунком. Виявлені в процесі розробки недоліки (затримка відображення стану завантаження, обробка помилок мережі, перенаправлення після оплати через Stripe) усувались ітеративно до повторної перевірки відповідних сценаріїв. Успішне усунення цих зауважень дозволило досягти безперебійної та синхронної роботи застосунку на різних платформах і пристроях користувачів. Таким чином, забезпечено високу надійність збереження даних та комфортний користувацький досвід під час взаємодії з усіма сервісами системи.

Таблиця 3.3

Тестування платіжних та додаткових функцій

Операція	Умови використання	Очікуваний результат	Фактичний результат	Статус
Оплата платного квитка з мобільного застосунку	Подія з ціною > 0	Перехід на вебсторінку Stripe для оплати	Користувача успішно оплатив квиток	Пройдено
Відображення цифрового квитка	Наявність купленого квитка	QR-код, назва події, дата, локація, статус квитка	Квиток відображається однаково на мобільному та вебклієнті	Пройдено
Публікація нової події з мобільного застосунку	Заповнення даних події (дата, час, опис, назва, картинка)	Дані відправляються на бекенд і зберігаються	Подія успішно створена	Пройдено
Генерація опису події через AI	Назва і категорія в редакторі, кнопка AI-опису	Текст підставляється в поле опису	Згенерований опис відображається і доступний для редагування	Пройдено

Представлені дані свідчать про успішну інтеграцію зовнішніх сервісів, зокрема платіжного шлюзу Stripe та моделей штучного інтелекту для генерації контенту. Усі протестовані транзакції та операції з формування цифрових квитків із QR-кодами продемонстрували повну відповідність закладеній бізнес-логіці. Відсутність відхилень між очікуваними та фактичними результатами

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		87

підтверджує надійність архітектурних рішень на етапі критичних користувацьких сценаріїв.

Висновки до розділу 3

У третьому розділі проведено комплексний опис інтерфейсної частини та результатів функціонального тестування мультиплатформеного програмного комплексу. Завдяки впровадженню інтуїтивно зрозумілої структури навігації в мобільних застосунках та вебклієнті, забезпечено зручний доступ до всіх ключових сценаріїв користувача, від швидкого пошуку та фільтрації заходів до керування власним розкладом і фінансовими операціями. Детально проаналізовано організацію екранів, що охоплюють взаємодію з подіями, роботу гаманця організатора, систему сповіщень та інтеграцію інструментів ІІІ для автоматизації контенту. Проведене ручне тестування ключових функціональних вузлів системи, результати якого наведено у вигляді таблиці, підтвердило коректність роботи всіх інтерфейсних елементів та надійність взаємодії клієнтської частини з серверною підсистемою через REST API. Виявлені під час перевірки недоліки були оперативно усунуті, що гарантує стабільність відображення даних та безперебійність користувацьких сценаріїв. Описані методи візуалізації даних, зокрема використання інтерактивних графіків і діаграм у розділі статистики, дозволяють організаторам отримувати наочну інформацію про ефективність їхньої діяльності, що значно підвищує аналітичну цінність платформи. Архітектурні рішення, закладені в основу інтерфейсу, забезпечують високу швидкість відгуку системи та однакову якість відображення контенту на різних пристроях. Додатково, уніфікація елементів керування сприяє швидкій адаптації користувачів до інтерфейсу та зменшує кількість помилкових дій під час роботи з платформою. Таким чином, розроблений інтерфейс повністю відповідає вимогам ергономіки та функціональності, а результати тестування засвідчують відповідність комплексу поставленим завданням та його повну готовність до подальшої експлуатації.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		88

Отримані результати також підтверджують доцільність використаних архітектурних підходів, які забезпечили стабільну роботу системи в умовах багатоплатформенного середовища. Крім того, реалізовані рішення дозволили досягти узгодженості між усіма компонентами комплексу та мінімізувати ризики виникнення помилок під час взаємодії клієнтських частин із сервером.

Результати тестування також підтвердили коректність реалізації ключових сценаріїв взаємодії користувача із системою, включаючи авторизацію, роботу з подіями та фінансові операції. Перевірені сценарії охоплювали як типові користувацькі дії, так і граничні випадки, що дозволило оцінити стійкість системи до різних умов використання. Окремо було зафіксовано швидку реакцію інтерфейсу на дії користувача, що позитивно впливає на загальний користувацький досвід. Отримані дані тестування узгоджуються між усіма платформами та не виявляють розбіжностей у стані даних. Особливу увагу було приділено перевірці синхронізації даних між мобільним застосунком, вебклієнтом та серверною частиною, яка продемонструвала стабільну та передбачувану роботу.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		89

ВИСНОВКИ

У кваліфікаційній роботі розглянуто процеси цифрової організації розважальних заходів і обґрунтовано доцільність створення власного мультиплатформеного комплексу. Метою роботи було розробити систему для керування подіями, яка поєднує серверну підсистему, вебсайт і мобільний застосунок для Android та iOS з єдиним джерелом даних. Поставлену мету досягнуто: спроектовано та реалізовано працездатний комплекс, що підтримує основні сценарії користувача — від перегляду афіші до участі в заході та роботи організатора з подіями й фінансами.

У першому розділі сформульовано вимоги до системи, проаналізовано аналоги Luma, Peatix і Partiful і показано, що жоден із них не поєднує в одному рішенні повний цикл роботи з подіями, оплатою, гаманцем організатора та просуванням. Обґрунтовано клієнт-серверну архітектуру з REST API, модульну організацію серверної частини та багатоплатформну модель із окремими застосунками для вебу й мобільних платформ. Визначено інструментальні засоби: Django і PostgreSQL на сервері, React для вебклієнта, Kotlin Multiplatform і Jetpack Compose Multiplatform для мобільного застосунку.

У другому розділі побудовано функціональну модель системи з прецедентами користувача та спроектовано базу даних із сутностями облікових записів, каталогу подій, реєстрацій, коментарів, відгуків, платежів, гаманця, просування та сповіщень. Описано архітектуру серверної підсистеми у вигляді модульного моноліту, структуру мобільного застосунку з розділенням на шари інтерфейсу, моделей подання, репозитаріїв і мережевої взаємодії, а також реалізовано модель даних засобами Django ORM, визначено принципи побудови REST API та механізми автентифікації на основі JSON Web Token, які стали фундаментом для технічної реалізації системи.

У третьому розділі проведено комплексний опис інтерфейсної частини та результатів функціонального тестування мультиплатформеного програмного комплексу. Проаналізовано організацію навігації, екранів взаємодії з подіями, гаманця організатора та інтеграцію інструментів ШІ для автоматизації контенту.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		90

Проведене ручне тестування ключових функціональних вузлів системи підтвердило коректність роботи всіх інтерфейсних елементів та надійність взаємодії клієнтської частини із сервером. Виявлені під час перевірки недоліки були ітеративно усунуті, що гарантує стабільність відображення даних та безперебійність користувацьких сценаріїв, таких як реєстрація, оплата через Stripe, генерація цифрових квитків та робота з профілем.

Практична цінність роботи полягає в тому, що створений програмний комплекс дає змогу організаторам публікувати події, приймати реєстрації та оплату, вести облік коштів і просувати заходи, а учасникам — швидко знаходити події, записуватися на них, отримувати квитки та взаємодіяти з контентом. Використання Kotlin Multiplatform зменшує дублювання логіки між Android і iOS, а модульна серверна архітектура спрощує подальший розвиток окремих частин системи.

Подальший розвиток мультиплатформеного комплексу може включати розширення функціональності вебклієнта, удосконалення аналітики для організаторів, додаткові способи сповіщення користувачів і оптимізацію продуктивності при зростанні навантаження. Отримані результати підтверджують доцільність обраного архітектурного підходу та успішну реалізацію всіх запланованих функціональних можливостей.

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		91

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Миколайчук В. Ю., Лімінович І. Д., Локтікова Т. М. Архітектура та реалізація клієнтської частини платформи для організації розважальних заходів на базі Kotlin Multiplatform. Всеукраїнська науково-практична онлайн конференція здобувачів вищої освіти і молодих учених, присвячена Дню науки (15 травня 2026 року).

2. Ічанська Н. В., Улько С. І. Основні аспекти створення мобільних додатків та вибір інструментів їх розробки. Системи управління, навігації та зв'язку. 2020. № 1(59). С. 74–83.

3. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures : dis doctor of philosophy. Irvine : University of California, 2000. 180 р. URL: <https://ics.uci.edu/~fielding/pubs/dissertation/top.htm> (дата звернення: 30.05.2026).

4. Luma : платформа для організації подій. URL: <https://luma.com/> (дата звернення: 30.05.2026).

5. Peatix : платформа для організації та продажу квитків на події. URL: <https://peatix.com/> (дата звернення: 30.05.2026).

6. Partiful : платформа для організації заходів. URL: <https://partiful.com/> (дата звернення: 30.05.2026).

7. Посвістак В. С., Демківська Т. І. Клієнт-серверна архітектура та її використання при розробці програмного забезпечення. Інформаційні технології в науці, виробництві та підприємництві : зб. наук. праць / за заг. наук. ред. В. Ю. Щербаня. Київ : Освіта України, 2020. С. 78–81.

8. Технології розробки програмного забезпечення : навч. посіб. / В. В. Соколов, В. В. Рябцев, А. В. Микитюк. Київ : КПІ ім. Ігоря Сікорського, 2025. 212 с. URL: <https://ela.kpi.ua/> (дата звернення: 30.05.2026).

9. Guide to app architecture // Android Developers. URL: <https://developer.android.com/topic/architecture> (дата звернення: 30.05.2026).

10. Python : офіційна документація. URL: <https://docs.python.org/3/> (дата звернення: 30.05.2026).

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				92
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Django : офіційна документація. URL: <https://docs.djangoproject.com/> (дата звернення: 30.05.2026).
12. Django REST framework : офіційна документація. URL: <https://www.django-rest-framework.org/> (дата звернення: 30.05.2026).
13. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT) : RFC 7519. IETF, 2015. URL: <https://datatracker.ietf.org/doc/html/rfc7519> (дата звернення: 30.05.2026).
14. PostgreSQL : офіційна документація. URL: <https://www.postgresql.org/docs/> (дата звернення: 30.05.2026).
15. React : офіційна документація. URL: <https://react.dev/> (дата звернення: 30.05.2026).
16. React Router : офіційна документація. URL: <https://reactrouter.com/> (дата звернення: 30.05.2026).
17. Kotlin Multiplatform : офіційна документація. URL: <https://kotlinlang.org/docs/multiplatform.html> (дата звернення: 30.05.2026).
18. Compose Multiplatform: A modern UI framework for shared development : JetBrains. URL: <https://www.jetbrains.com/lp/compose-multiplatform/> (дата звернення: 03.06.2026).
19. Ktor : офіційна документація. URL: <https://ktor.io/docs/> (дата звернення: 30.05.2026).
20. Kotlin Serialization : офіційна документація. URL: <https://kotlinlang.org/docs/serialization.html> (дата звернення: 30.05.2026).
21. Koin : офіційна документація. URL: <https://insert-koin.io/> (дата звернення: 30.05.2026).
22. DataStore : офіційна документація // Android Developers. URL: <https://developer.android.com/topic/libraries/architecture/datastore> (дата звернення: 30.05.2026).
23. Coil : офіційна документація. URL: <https://coil-kt.github.io/coil/> (дата звернення: 30.05.2026).
24. Navigation with Compose : офіційна документація // Android

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				93
Змн.	Арк.	№ докум.	Підпис	Дата		

Developers. URL: <https://developer.android.com/develop/ui/compose/navigation> (дата звернення: 30.05.2026).

25. JavaScript : офіційна документація. URL: <https://developer.mozilla.org/uk/docs/Web/JavaScript> (дата звернення: 03.06.2026).

26. Гайдаржи В. І., Ізварін І. В. Бази даних в інформаційних системах : підруч. Київ : Університет «Україна», 2018. 418 с.

27. Організація баз даних : навч. посіб. / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, І. М. Копитчук. 2-ге вид., виправл. і доповн. Одеса : Фенікс, 2019. 246 с.

28. Django : Using the ORM. URL: <https://docs.djangoproject.com/en/stable/topics/db/> (дата звернення: 30.05.2026).

29. Jetpack Compose Multiplatform : офіційна документація. URL: <https://www.jetbrains.com/compose-multiplatform/> (дата звернення: 30.05.2026).

30. Web-програмування. Частина 1 (frontend) : навч. посіб. / В. В. Босько, Л. В. Константинова, К. М. Марченко, О. С. Улічев. Кропивницький : ЦНТУ, 2022. 208 с.

31. Stripe : документація API. URL: <https://docs.stripe.com/> (дата звернення: 30.05.2026).

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		94

ДОДАТКИ

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		95

Модель і сервіс електронного квитка з генерацією QR-коду

```
# backend/tickets/models.py
```

```
import uuid
```

```
from django.db import models
```

```
class EventTicket(models.Model): registration = models.OneToOneField(
    "events.EventRegistration", on_delete=models.CASCADE, related_name="ticket",
)
```

```
    uuid = models.UUIDField(default=uuid.uuid4, unique=True,
                             editable=False, db_index=True)
```

```
    qr_code = models.ImageField(upload_to="tickets/qr/", blank=True) is_used =
    models.BooleanField(default=False)
```

```
    used_at = models.DateTimeField(null=True, blank=True) created_at =
    models.DateTimeField(auto_now_add=True) updated_at =
    models.DateTimeField(auto_now=True)
```

```
# backend/tickets/services/ticket_service.py import qrcode
```

```
from django.conf import settings
```

```
from django.core.files.base import ContentFile from io import BytesIO
```

```
from ..models import EventTicket
```

```
def ticket_public_url(ticket: EventTicket) -> str:
```

```
    base = getattr(settings, "FRONTEND_URL", "").rstrip("/") return
```

```
    f"{base}/tickets/{ticket.uuid}"
```

```
        _uiState.value = _uiState.value.copy(
```

```
            paymentActionPending = false,
```

```
            paymentUrlToOpen = result.data.paymentUrl,
```

```
        )
```

```
    }
```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		96

```

is NetworkResult.Error -> {
    _uiState.value = _uiState.value.copy(
        paymentActionPending = false,
        paymentActionError = result.errorType,
    )
}
}
}

def generate_ticket_qr(ticket: EventTicket) -> None:
qr = qrcode.QRCode(version=1, box_size=8, border=2)
qr.add_data(ticket_public_url(ticket)) qr.make(fit=True)
image = qr.make_image(fill_color="black", back_color="white") buffer = BytesIO()
image.save(buffer, format="PNG") buffer.seek(0)
ticket.qr_code.save(f"{ticket.uuid}.png", ContentFile(buffer.read()), save=False)
ticket.save(update_fields=["qr_code", "updated_at"])

def create_ticket_for_registration(registration) -> EventTicket | None:
ticket, created = EventTicket.objects.get_or_create(registration=registration) if
created or not ticket.qr_code:
    generate_ticket_qr(ticket) return ticket

```

Додаток Б

Реалізація шаблону MVVM: отримання даних з API, перетворення моделей і
навігація між екранами мобільного застосунку

```

data class EventFeedUiModel(
val id: Int,
val title: String,
val description: String?, val categoryLine: String, val startLabel: String, val endLabel:
String,
val authorLine: String?, val likesCount: Int,

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				97
Змн.	Арк.	№ докум.	Підпис	Дата		

```

val isLiked: Boolean = false,
val isRegistered: Boolean = false, val isPaid: Boolean = false,
val ticketPrice: String? = null,
// ...
)

fun EventDto.toFeedUiModel(
    timeZone: TimeZone = TimeZone.currentSystemDefault(),
): EventFeedUiModel {
val refs = categories.zip(categoryNames).mapNotNull { (catId, rawName) -> val name =
    rawName.trim()
    if (name.isEmpty()) null else EventCategoryRef(id = catId, name = name)
}
return EventFeedUiModel( id = id,
    title = title.trim().takeIf { it.isNotEmpty() } ?: "Подія #$id", description =
    description?.takeIf { it.isNotBlank() }, categoryLine = refs.joinToString(" · ") {
    it.name }, startLabel = formatIsoDateTime(startDate, timeZone), endLabel =
    formatIsoDateTime(endDate, timeZone),
    authorLine = authorName?.trim()?.takeIf { it.isNotEmpty() }, likesCount =
    likesCount,
    isLiked = isLiked, isRegistered = isRegistered, isPaid = isPaid,
    ticketPrice = ticketPrice,
    // ...
)
}

data class EventDetailState(
    val loading: Boolean = true,
    val event: EventFeedUiModel? = null, val error: NetworkError? = null,
    val likeActionPending: Boolean = false,
    val registrationActionPending: Boolean = false, val paymentUrlToOpen: String? =

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		98

```

null,
val reviews: List<EventReviewUiModel> = emptyList(),
// ...
)

```

```

class EventDetailViewModel( private val eventId: Int,
    private val eventsRepository: EventsRepository, private val paymentsRepository:
    PaymentsRepository, private val ticketsRepository: TicketsRepository,
    // ...
) : ViewModel() {

```

```

    private val _uiState = MutableStateFlow(EventDetailState())
    val uiState: StateFlow<EventDetailState> = _uiState.asStateFlow()

```

```

    init {
        refresh(silent = false)
    }

```

```

fun refresh(silent: Boolean = false) { viewModelScope.launch {
    val previous = _uiState.value if (!silent) {
        _uiState.value = EventDetailState(loading = true, event = null, error = null)

    }

    when (val result = eventsRepository.getEventById(eventId)) { is NetworkResult.Success
        -> {
            val event = result.data.toFeedUiModel()
            _uiState.value = previous.copy( loading = false,
                event = event, error = null,

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				99
Змн.	Арк.	№ докум.	Підпис	Дата		


```

uiState.loading -> LoadingContent()
uiState.error != null -> ErrorContent(onRetry = viewModel::load) event != null ->
EventDetailContent(
event = event,
onLikeClick = viewModel::toggleLike, onRegisterClick = viewModel::register,
onCommentsClick = onOpenComments,
// ...
)
}
}

```

@Serializable

data object HomeFeedRoute

@Serializable

data class EventDetailRoute(val eventId: Int)

@Serializable

data class EventCommentsRoute(val eventId: Int)

@Serializable

data class TicketDetailRoute(val ticketUuid: String) fun

NavGraphBuilder.eventDetailDestination(

navController: NavHostController, onLeaveEventDetail: () -> Unit = {},

onOpenTicket: (String) -> Unit = {},

) {

composable<EventDetailRoute> { entry ->

val route = entry.toRoute<EventDetailRoute>()

val viewModel: EventDetailViewModel = koinViewModel {

parametersOf(route.eventId)

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		101

```

    }
EventDetailScreen( eventId = route.eventId,
    viewModel = viewModel, onBack = {
        if (navController.popBackStack()) onLeaveEventDetail()
    },
onOpenComments = { navController.navigate(EventCommentsRoute(route.eventId))
    },
    onOpenTicket = onOpenTicket,
    // ...
)
}
}

@Composable
fun HomeTabNavHost( navController: NavHostController,
    homeTabViewModel: HomeTabViewModel,
) {
    NavHost(
        navController = navController, startDestination = HomeFeedRoute,
    ) {
composable<HomeFeedRoute> { HomeTabScreen(
    viewModel = homeTabViewModel, onOpenEvent = { id ->
        navController.navigate(EventDetailRoute(id))
    },
onOpenCategory = { categoryId, name ->
        navController.navigate(CategoryEventsRoute(categoryId, name))
    },
)
}

eventDetailDestination( navController = navController, onOpenTicket = { uuid ->
    navController.navigate(TicketDetailRoute(uuid))

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		102

```

    },
    )
    ticketsDestinations(navController = navController)
    // ...
    }
    }

    @Composable
    internal fun EventHubNavHost() {
        val navController = rememberNavController()
        val startRoute = if (hasSession) HomeRoute else SignInRoute

        NavHost(
            navController = navController, startDestination = startRoute,
        ) {
            signInScreen( onSignInSuccess = {
                navController.navigate(HomeRoute) { popUpTo(SignInRoute) { inclusive = true }
            }
        },
        )
            signUpScreen( onRegistrationComplete = {
                navController.navigate(HomeRoute) { popUpTo(SignInRoute) { inclusive = true }
            }
        },
        )
        homeScreen()
    }
}

```

Додаток В

Реалізація пошуку, фільтрації та посторінкової видачі подій у мобільному

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				103
Змн.	Арк.	№ докум.	Підпис	Дата		

застосунок

@Serializable

```
data class EventPagedDto( val count: Int,
    val next: String? = null,
    val previous: String? = null,
    val results: List<EventDto> = emptyList(),
)

suspend fun getEvents( page: Int = 1, pageSize: Int = 20, search: String? = null,
    categoryIds: Collection<Int> = emptyList(), citySlug: String? = null,
    locationSlug: String? = null, authorId: Int? = null,
    ordering: String? = "-start_date",
): NetworkResult<EventPagedDto> = network.get("events/events") {
    parameter("page", page) parameter("page_size", pageSize) val q =
    search?.trim().orEmpty()
    if (q.isNotEmpty()) parameter("search", q) categoryIds.forEach {
    parameter("categories", it) }
    citySlug?.trim()?.takeIf { it.isNotEmpty() }?.let { parameter("city", it) }
    locationSlug?.trim()?.takeIf { it.isNotEmpty() }?.let { parameter("location", it) }
    authorId?.let { parameter("author", it) }
    val ord = ordering?.trim().orEmpty()
    if (ord.isNotEmpty()) parameter("ordering", ord)
    }

    enum class SearchDatePreset {
    Any, Today, ThisWeek, Custom,
    }

    data class SearchFilters(
    val cities: Set<CityRef> = emptySet(),
```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		104

```

val categories: Set<EventCategorySelection> = emptySet(), val datePreset:
SearchDatePreset = SearchDatePreset.Any, val customDate: LocalDate? = null,
) {
    val hasAny: Boolean
get() = cities.isNotEmpty() || categories.isNotEmpty() || datePreset !=
    SearchDatePreset.Any
}
fun loadForSearch(query: String, filters: SearchFilters) { lastQuery = query
    lastFilters = filters viewModelScope.launch {
        _uiState.update { it.copy(resultsLoading = true, resultsError = null) } val categoryIds
        = filters.categories.map { it.id }
        val citySlugs = filters.cities.map { it.slug } val eventsResult = fetchEventsForFilters(
            query = query, categoryIds = categoryIds, citySlugs = citySlugs,
        )
        when (eventsResult) {
is NetworkResult.Success -> { val preset = filters.datePreset val filtered =
            eventsResult.data
                .asSequence()
                .filter { event ->
                    citySlugs.isEmpty() || event.city in citySlugs
                }
                .filter { it.matchesDatePreset(preset, filters.customDate) }
                .map { it.toFeedUiModel() }
                .toList()
            _uiState.update { it.copy(
                results = filtered, resultsLoading = false, totalCount = filtered.size,
            )
        }
    }
}

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		105

```

        is NetworkResult.Error -> {
    _uiState.update { it.copy(
        resultsLoading = false,
        resultsError = eventsResult.errorType,
    )
    }
    }
    }
    }
    }
} private suspend fun fetchEventsForFilters( query: String,
    categoryIds: List<Int>, citySlugs: List<String>,
    ): NetworkResult<List<EventDto>> {
    val trimmedQuery = query.trim().takeIf { it.isNotEmpty() } return when {
citySlugs.isEmpty() -> { when (
val r = eventsRepository.getEvents( page = 1,
    pageSize = 50,
    search = trimmedQuery, categoryIds = categoryIds, ordering = "-start_date",
    )
    ) {
        is NetworkResult.Success -> NetworkResult.Success(r.data.results) is
        NetworkResult.Error -> r
    }
    }
citySlugs.size == 1 -> { when (
val r = eventsRepository.getEvents( page = 1,
    pageSize = 50,
    search = trimmedQuery, categoryIds = categoryIds, citySlug = citySlugs.first(),
    ordering = "-start_date",
    )
    ) {

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		106

```

is NetworkResult.Success -> NetworkResult.Success(r.data.results) is
NetworkResult.Error -> r
}
}
else -> {
    val merged = LinkedHashMap<Int, EventDto>() for (slug in citySlugs) {
        when (
val r = eventsRepository.getEvents( page = 1,
    pageSize = 50,
    search = trimmedQuery, categoryIds = categoryIds, citySlug = slug,
    ordering = "-start_date",
                                )
                                ){
is NetworkResult.Success ->
    r.data.results.forEach { event -> merged[event.id] = event } is NetworkResult.Error
-> return r
}
}
NetworkResult.Success(merged.values.toList())
}
}
}
suspend fun getCities( search: String? = null,
    pageSize: Int = 100,
): NetworkResult<List<CityDto>> {
    val term = search?.trim()?.takeIf { it.isNotEmpty() } val aggregated =
    mutableListOf<CityDto>()
    var page = 1 while (true) {
        when (
val result = network.get<CityPagedDto>("events/cities") { parameter("page", page)

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		107

```

parameter("page_size", pageSize) term?.let { parameter("search", it) }
}
){
is NetworkResult.Success -> { aggregated.addAll(result.data.results) if
(result.data.next.isNullOrBlank()) {
return NetworkResult.Success(aggregated.toList())
}
page++
}
is NetworkResult.Error -> return result
}
}
}

fun EventDto.matchesDatePreset( preset: SearchDatePreset, customDate: LocalDate? =
null,
): Boolean {
val zone = TimeZone.currentSystemDefault() val eventDate =
runCatching {
Instant.parse(startDate).toLocalDateTime(zone).date
}.getOrNull()
?: return true
val today = EventHubTime.todayIn(zone) return when (preset) {
SearchDatePreset.Any -> true
SearchDatePreset.Custom -> customDate?.let { eventDate == it } ?: false
SearchDatePreset.Today -> eventDate == today SearchDatePreset.ThisWeek -> {
val end = today.plus(7, DateTimeUnit.DAY) eventDate >= today && eventDate <=
end
}
}
}
}

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		108

Реалізація ієрархічної навігації мобільного застосунку

@Serializable

data object SignInRoute

@Serializable

data object SignUpRoute

@Serializable

data object ForgotPasscodeRoute

@Serializable

data object ForgotPasscodeSuccessRoute @Serializable

data object HomeRoute

fun NavGraphBuilder.homeScreen() { composable<HomeRoute> {

HomeScreen()

}

}

@Composable

internal fun EventHubNavHost() {

val sessionManager = koinInject<AuthSessionManager>() val navController =

rememberNavController()

var sessionResolved by remember

{ mutableStateOf(AppBootstrapState.cachedHasSession) }

LaunchedEffect(Unit) {

if (sessionResolved == null) {

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		109

```

sessionResolved = sessionManager.hasStoredSession()
}
}

```

```

LaunchedEffect(sessionResolved) { sessionManager.sessionExpired.collect {
    sessionResolved = false navController.navigate(SignInRoute) {
        popUpTo(navController.graph.id) { inclusive = true } launchSingleTop = true
    }
}
}
}

```

```

val startRoute = if (sessionResolved == true) HomeRoute else SignInRoute

```

```

NavHost(
    navController = navController, startDestination = startRoute,
) {
    signInScreen(
        onSignUpClick = { navController.navigate(SignUpRoute) }, onSignInSuccess = {
navController.navigate(HomeRoute) { popUpTo(SignInRoute) { inclusive = true }
        },
        onForgotPasswordClick = { navController.navigate(ForgotPasscodeRoute) },
    )

    signUpScreen(
        onBack = { navController.popBackStack() }, onRegistrationComplete = {
navController.navigate(HomeRoute) { popUpTo(SignInRoute) { inclusive = true }
        },
    )
}

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		110

```

forgotPasscodeScreen(
  onBack = { navController.popBackStack() },
  onRecoveryCompleted = {
    navController.navigate(ForgotPasscodeSuccessRoute) },
) forgotPasscodeSuccessScreen( onUnderstood = {
  navController.navigate(SignInRoute) { popUpTo(SignInRoute) { inclusive = false }
    launchSingleTop = true
  }
},
)

homeScreen()
}
}

internal enum class MainTab { Home,
  Search, Favorites, Profile,
}

@Composable
fun HomeScreen() {
  var selectedTab by remember { mutableStateOf(MainTab.Home) }

  val homeNavController = rememberNavController() val searchNavController =
  rememberNavController() val favoritesNavController = rememberNavController() val
  profileNavController = rememberNavController()

```

Scaffold(bottomBar = {

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		111

```

    EventHubBottomBar(
        selected = selectedTab, onTabSelected = { selectedTab = it },
    )
},
) { innerPadding -> when (selectedTab) {
MainTab.Home -> HomeTabNavHost( navController = homeNavController, modifier =
    Modifier.padding(innerPadding),
)
MainTab.Search -> SearchTabNavHost( navController = searchNavController, modifier
    = Modifier.padding(innerPadding),
)
MainTab.Favorites -> FavoritesTabNavHost( navController = favoritesNavController,
    modifier = Modifier.padding(innerPadding),
)
MainTab.Profile -> ProfileTabNavHost( navController = profileNavController, modifier
    = Modifier.padding(innerPadding),
)
}
}
}

@Serializable
data object HomeFeedRoute

@Serializable
data object NotificationsRoute

@Serializable
data class EventDetailRoute(val eventId: Int)

@Serializable

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		112

data class EventCommentsRoute(val eventId: Int)

@Serializable

data class CategoryEventsRoute(val categoryId: Int, val categoryName: String)

@Serializable

data object EventCreateRoute

@Serializable

data class UserProfileRoute(val userId: Int, val displayName: String)

@Serializable

data object ProfileMainRoute

@Serializable

data object ProfileAccountSettingsRoute

@Serializable

data object SettingsEditProfileNameRoute

@Serializable

data object SettingsEditProfileBioRoute

@Serializable

data object SettingsChangePasswordRoute

@Serializable

data object SettingsChangeEmailRoute

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		113

@Serializable

data object SettingsWalletRoute

@Serializable

data object SettingsCalendarEventsRoute

@Serializable

data object SettingsDeleteAccountRoute @Composable

```
fun ProfileTabNavHost(navController: NavHostController) { NavHost(
    navController = navController, startDestination = ProfileMainRoute,
) {
    composable<ProfileMainRoute> { ProfileMainScreen(
        onOpenAccountSettings = { navController.navigate(ProfileAccountSettingsRoute)
        },
        onOpenStatistics = { navController.navigate(ProfileStatisticsRoute)
        },
        onOpenEvent = { id -> navController.navigate(EventDetailRoute(id))
        },
    )
}
```

```
composable<ProfileAccountSettingsRoute> { ProfileSettingsListScreen(
    onBack = { navController.popBackStack() }, onEditProfileName = {
    navController.navigate(SettingsEditProfileNameRoute)
    },
    onEditProfileBio = { navController.navigate(SettingsEditProfileBioRoute)
    },
    onChangePassword = { navController.navigate(SettingsChangePasswordRoute)
```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				114
Змн.	Арк.	№ докум.	Підпис	Дата		

```

},
onWallet = { navController.navigate(SettingsWalletRoute) }, onMyTickets = {
navController.navigate(MyTicketsRoute) }, onDeleteAccount = {
navController.navigate(SettingsDeleteAccountRoute)
},
)
}

```

```

eventDetailDestination(navController = navController)
ticketsDestinations(navController = navController)
// ...
}
}

```

		Миколайчук В.Ю.			ІПЗ.КР.Б –121–26–ПЗ	Арк.
		Локтікова Т.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		115