



**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПОЛТАВСЬКА ПОЛІТЕХНІКА
ІМЕНІ ЮРІЯ КОНДРАТЮКА**

ЗБІРНИК МАТЕРІАЛІВ

**77-ї НАУКОВОЇ КОНФЕРЕНЦІЇ ПРОФЕСОРІВ,
ВИКЛАДАЧІВ, НАУКОВИХ ПРАЦІВНИКІВ,
АСПІРАНТІВ ТА СТУДЕНТІВ УНІВЕРСИТЕТУ**

16 травня – 22 травня 2025 р.

Сучасні підходи до реалізації мобільних додатків умовно поділяються на три категорії: нативні, гібридні та веб-додатки. Кожен з них має свої переваги й недоліки. Нативні додатки відзначаються високою продуктивністю, повним доступом до функцій пристрою, але вимагають окремої розробки для кожної ОС. Гібридні додатки дозволяють одночасно підтримувати Android та iOS, зменшуючи час та вартість розробки. Веб-додатки, у свою чергу, забезпечують швидкий доступ через браузер, однак мають обмеження у продуктивності й доступі до системних ресурсів пристрою.

Таким чином, вибір залежить від вимог до функціональності, бюджету, термінів реалізації та цільової аудиторії, що робить етап аналізу програмного забезпечення критично важливим у процесі розробки мобільних додатків. При цьому встановлено, перспективними напрямками є використання: кросплатформної розробки як економічно ефективною стратегією; сучасних мов (наприклад, Swift замість Objective-C); гібридних технологій (HTML5, C# + Xamarin) для швидкого виведення продукту на ринок.

Отже, у роботі проведено комплексний аналіз сучасних інструментів і методів розробки мобільних додатків із акцентом на актуальні мови програмування, середовища розробки та архітектурні підходи.

Література :

1. Вікіпедія: вільна енциклопедія. URL: https://uk.wikipedia.org/wiki/Мобільний_додаток
2. Тарнавський Ю.А. Java Програмування навч. посіб. Київ; КПІ. 2021, 95 с.

УДК 004.9

*Я.В. Падусенко, студент групи 401-ТК,
О.В. Скакаліна, к.т.н., доцент
Національний університет
«Полтавська політехніка імені Юрія Кондратюка»*

РОЗРОБКА АДАПТИВНОГО ПРОТОТИПУ ДЕЦЕНТРАЛІЗОВАНОЇ КРИПТОВАЛЮТНОЇ БІРЖІ: АНАЛІЗ, РЕАЛІЗАЦІЯ ТА ПЕРСПЕКТИВИ

Децентралізовані криптовалютні біржі (DEX) набувають популярності у сфері фінансів, надаючи користувачам безпеку, конфіденційність та контроль над їх активами. Створення адаптивного прототипу такої біржі,

що застосовує новітні блокчейн-технології, є важливим завданням, що допомагає інтегрувати криптовалюти в світ фінансів.

Першим етапом дослідження було вивчення наявних продуктів на ринку децентралізованих бірж, як-от Uniswap, PancakeSwap та SushiSwap. Здійснено порівняльний аналіз централізованих (CEX) та децентралізованих (DEX) бірж, в результаті якого було визначено основні переваги останніх, зокрема відсутність централізованого контролю, застосування смарт-контрактів для автоматизації операцій обміну та зменшення ризиків злому. Крім того, було розглянуто сучасні технології для розробки DEX, включаючи блокчейн-платформи Ethereum та Binance Smart Chain, мову програмування Solidity для створення смарт-контрактів, фреймворком React.js для розробки адаптивного інтерфейсу та бібліотекою Ethers.js для інтеграції з блокчейном.

Другий етап розробки передбачав проектування архітектури майбутньої біржі. В основу функціонування прототипу було покладено механізм автоматизованого маркет-мейкера (АММ). Зібрано структуру смарт-контрактів, що відповідають за втілення логіки пулів ліквідності та обмін токенів. Для клієнтської частини спроектовано адаптивний веб-інтерфейс, що гарантує комфортну взаємодію користувачів із системою. Важливим аспектом архітектури стала інтеграція з популярним Web3-гаманцем MetaMask для забезпечення безпечної автентифікації користувачів та підпису транзакцій.

Третій етап концентрувався на практичному втіленні прототипу. Для створення смарт-контрактів обрали мову Solidity та платформу Hardhat, яку використано для випробування коду. Фронтенд розроблено на основі React.js. Бібліотеку Ethers.js застосовано для забезпечення взаємодії з блокчейном. Основні функції прототипу включають механізм обміну токенів, що використовує пули ліквідності, показ історії операцій, а також адаптивний інтерфейс, який бездоганно працює на різних пристроях з різними розмірами екранів.

Фінальний етап роботи передбачав тестування та аналіз отриманих даних. Використовуючи Hardhat, було здійснено функціональне тестування смарт-контрактів. Юзабіліті інтерфейсу перевірено шляхом відповідних тестувань його адаптивності. Підсумкові результати засвідчили продуктивність прийнятих рішень, а також підтвердили придатність прототипу до подальшого розширення функціоналу.

Майбутнє проекту передбачає інтеграцію з іншими блокчейн-мережами, наприклад Solana. Також передбачається введення нових фінансових інструментів, зокрема стейкінгу та фармінгу. Важливим моментом є подальше вдосконалення безпеки та здатності до масштабування.

Розроблений прототип децентралізованої криптовалютної біржі, здатний адаптуватися, показав свою продуктивність та практичну користь.

Застосування передових технологій блокчейну у поєднанні з ретельно розробленим UX/UI дизайном дозволило створити діючий прототип. Він має потенціал стати підґрунтям для повноцінного кінцевого продукту.

Література:

1. *Uniswap v2 Core Whitepaper. URL: <https://uniswap.org/whitepaper.pdf> (дата звернення – 07.05.2025 р.)*

2. *Ethereum Whitepaper / Vitalik Buterin. URL: <https://ethereum.org/en/whitepaper> (дата звернення – 07.05.2025 р.)*

УДК 004.415

*В.В. Загнойко, асистент
Національний університет
«Полтавська політехніка імені Юрія Кондратюка»*

РИЗИКИ І ПОМИЛКИ ПРИ ВПРОВАДЖЕННІ АВТОМАТИЧНОГО ТЕСТУВАННЯ В РЕАЛЬНОМУ ПРОЄКТІ

Автоматичне тестування є критично важливою частиною життєвого циклу програмного забезпечення, особливо в умовах сучасної розробки, що орієнтована на Agile-методології та безперервну інтеграцію і доставку (CI/CD). Його головна мета — забезпечити швидку зворотну реакцію на зміни в коді, зменшити кількість помилок у продуктиві та оптимізувати ресурси тестувальників. Проте, незважаючи на очевидні переваги, автоматизація тестування часто стикається з рядом складнощів під час впровадження в реальних проєктах.

Однією з найпоширеніших помилок є початок автоматизації без чітко сформованої стратегії. Часто команда просто починає «писати тести», не маючи відповіді на базові питання:

- Що саме має бути автоматизовано?
- У якій послідовності?
- Який рівень покриття є цільовим?
- Як тести будуть інтегровані у загальний процес розробки?

В результаті створюється велика кількість неузгоджених, дубльованих або надмірно складних тестів, які важко підтримувати. Іншою проблемою є неправильний вибір інструментів. Інструменти для автоматизації мають відповідати технологічному стеку, типу продукту, а також рівню компетенції команди. Наприклад, вибір Selenium WebDriver для