

УДК 004.45

## **ПОРІВНЯЛЬНИЙ АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ**

Кислий Олександр

Науковий керівник: канд. ф.-м. доцент Паращук С.Д.

Центральноукраїнський державний педагогічний університет імені Володимира  
Винниченка, м. Кропивницький, Україна

*Стаття присвячена дослідженню питання визначення підходів та інструментальних засобів кросплатформної розробки мобільних застосунків, які дозволяють прискорити процес написання програмного коду застосунка. Проведено аналіз існуючих та популярних фреймворків, що використовуються для написання мобільних застосунків за певними критеріями. У кожного фреймворку є свої переваги та недоліки. При виборі технології слід керуватися необхідними потребами, бюджетом, наявністю команди розробників та іншими чинниками. Загалом React Native ефективно задовольняє більшість потреб.*

*Ключові слова: мобільний застосунок, кросплатформна розробка, нативна розробка, React Native, Flutter, Ionic.*

## **COMPARATIVE ANALYSIS OF TOOLS FOR DEVELOPMENT OF MOBILE APPLICATIONS**

A. Kisliy

Scientific supervisor: Candidate of Physics and Mathematics Docent Paraschuk S.D.

Volodymyr Vynnychenko Central Ukrainian State Pedagogical University,  
Kropyvnytsky, Ukraine

*The article is devoted to the study of the definition of approaches and tools for cross-platform development of mobile applications, which can speed up the process of writing application code. An analysis of existing and popular frameworks used to write mobile applications according to certain criteria. Each framework has its advantages and disadvantages. The choice of technology should be guided by the necessary needs, budget, availability of the development team and other factors. In general, React Native effectively meets most needs.*

*Key words: mobile application, cross-platform development, native development, React Native, Flutter, Ionic.*

Мобільність стає невід'ємною частиною всіх сфер життя. Мобільність це гнучкість, мінливість та здатність адаптуватися до безмежних можливостей для нових продуктів та розробок.

На даний момент мобільні програми є невід'ємною частиною нашого повсякденного життя. На ринку багато різних мобільних застосунків. За сферами застосування мобільні програми можна класифікувати наступним чином: ігрові програми, бізнес-програми, освітні програми, соціальні мобільні програми, новинні програми та ін.

Бізнес-програми займають одне із головних місць. З їх допомогою можна: зробити оплату, купівлю, бронювання, відправити електронного листа, відстежити робочий процес тощо. Бізнес-програми підвищують продуктивність та мінімізують витрати. Бізнес-програми – це перспективний інструмент мобільного маркетингу.

Мобільні застосунки для бізнесу розвиваються швидкими темпами та допомагають стати ближче до клієнта. Вони потрібні як продавцями послуг, так і клієнтами. Для перших це додатковий канал просування та продажу, для других – зручний, швидкий та доступний спосіб отримання актуальної інформації та послуги. Мобільні застосунки мають величезний потенціал у плані залучення та утримання клієнтів.

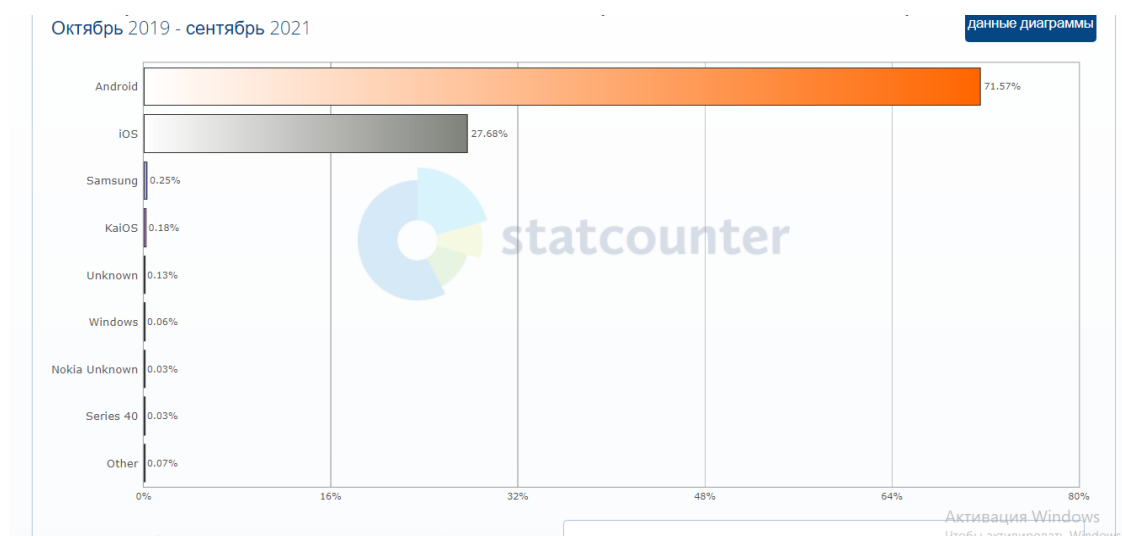
Не всі компанії мають змогу дозволити собі найняти дві команди для розробки власного мобільного застосунка для двох основних платформ: Android та iOS. У пошуках вирішення цієї проблеми було знайдено прогресивне рішення: розробка кросплатформних мобільних застосунків, які дозволяють домогтися якості продукту, який можна порівняти з нативним аналогом, і навіть краще.

Використання кросплатформних інструментів дозволяє реалізувати лише один застосунок, який потім розгортається або експортується на всі платформи. Такий підхід економить час та дозволяє розробникам використовувати загальну базу коду на одній мові програмування для кількох мобільних чи десктопних операційних систем. Незважаючи на цю перевагу, кросплатформні підходи

також можуть мати свої недоліки, наприклад, обмежений доступ до апаратних ресурсів пристрою, високий використання пам'яті, підвищене навантаження на обчислювальні пристрої.

В даний час розробка застосунків для смартфонів є популярним напрямком. Щодня активується дедалі більше мобільних пристроїв.

Ринок мобільних операційних систем різноманітний. Однак, згідно зі статистикою StatCounter [1] за період з жовтня 2019 року по вересень 2021 року, більшу частину ринку займають дві операційні системи: Android та iOS (рис. 1)

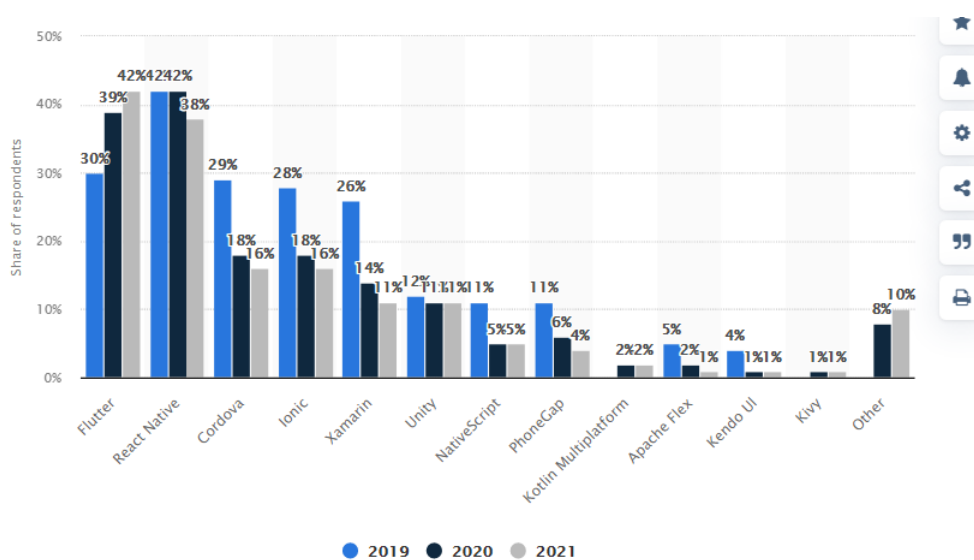


*Рис. 1 Доля ринку мобільних операційних систем*

Кожна мобільна платформа має власні засоби для розробки додатків. Для Android основною мовою розробки є Java (з травня 2017 року також Kotlin), для iOS мова Swift і раніше мова Objective C. Таким чином, для покриття більшої частини ринку мобільних пристроїв розробникам програмного забезпечення необхідно розробляти дві версії програм. Створення програм для кожної із мобільних платформ має свої нативні засоби розробки. Але ми не будемо розглядати нативні засоби розробки з очевидних на наш погляд причин: відсутність кросплатформності, а наслідком є збільшення штату розробників і об'єму роботи та економічно не вигідна вартість для замовника програмного продукту.

Однак існують кросплатформні рішення, що дозволяють писати єдину кодову базу, яка компілюватиметься у виконуючі файли для різних платформ. Це дозволяє знизити витрати на розробку та підтримку додатків.

Вибір технологій для проектування мобільних застосунків заснований на їхній популярності серед розробників. Аналіз існуючих на ринку рішень показав, що є декілька актуальних інструментів для кросплатформної розробки мобільних додатків. За даними сайту Statista на рис. 2 представлено діаграму фреймворків, які найбільше використовують розробники по всьому світу за остання 2 роки [2].



© Statista 2021 r.

*Рис. 2 Кросплатформні мобільні фреймворки, що використовуються розробниками по всьому світу, 2019-2021 рр.*

Для порівняння розглянемо фреймворки Ionic, Flutter и React Native.

Ionic – це інструментарій інтерфейсу користувача з відкритим вихідним кодом, який фокусується на зовнішньому інтерфейсі користувача та користувацькому інтерфейсі програми [3]. Дозволяє створювати програми для всіх основних платформ із єдиної кодової бази. Програми Ionic можуть мати нативний зовнішній вигляд, оскільки мають можливості адаптивного стилю на основі конкретної платформи.

Перевагою Ionic є можливість створювати гібридні мобільні програми, що позбавляє від складності наявності кількох команд на різних платформах, що підвищує продуктивність.

Компоненти побудовані за допомогою вебстандартів з використанням HTML, CSS та JavaScript. Кожен Ionic компонент адаптує свій зовнішній вигляд до платформи, на якій працює програма.

Програма Ionic, завантажена з Apple App Store, отримує тему iOS, а програма Ionic, завантажена з Play Store Android, отримує тему Material Design. Для Progressive Web App Ionic за замовчуванням використовує тему Material Design.

Ionic поставляється разом з новим середовищем виконання гібридних додатків з відкритим вихідним кодом Capacitor, який дозволяє запускати сучасні веб-застосунки для iOS, Android (з використанням технології Progressive Web App), надаючи потужний та простий у використанні інтерфейс для доступу до нативних пакетів SDK та API на кожній платформі [3].

Ionic програми створюються з використанням веб-технологій та візуалізуються з використанням Web Views, які являють собою повноекранний повнофункціональний веб-браузер.

Flutter – мобільний SDK з відкритим вихідним кодом для створення кросплатформних мобільних застосунків, представлений компанією Google у 2017 році. Основою для розробки Flutter-застосунків є об'єктно-орієнтована мова програмування Dart, розроблена в Google. Flutter має: широкий вибір нативних компонентів, які використовуються при побудові інтерфейсу застосунка; можливість швидкого перезавантаження програми для перегляду внесених змін; високу продуктивність.

Програми побудовані з єдиної кодової бази, скомпільовані в нативний код ARM, використовують графічний процесор (GPU) і можуть отримувати доступ до певних API-інтерфейсів iOS та Android (наприклад, розташування GPS, бібліотека зображень), спілкуючись по каналах платформи [4].

Під час розробки Flutter використовує гаряче перезавантаження, щоб оновити працюючий додаток за мілісекунди, коли змінюється вихідний код для додавання нових функцій або зміни уже існуючих. Гаряче перезавантаження – це спосіб побачити зміни, які вносяться до коду на симуляторі або пристрої, зберігаючи на екрані стан програми та дані. Інтерфейс Flutter реалізований з використанням віджетів на основі сучасних реактивних фреймворків та використовує власний механізм рендерингу. Віджети є будівельними блоками програми Flutter, і кожен є незмінним оголошенням інтерфейсу користувача.

React Native – open source фреймворк JavaScript, який використовується для розробки кросплатформних мобільних застосунків, вперше був представлений Facebook у 2015 році. Він підходить для нативних та гібридних проєктів.

В React Native інтерфейс користувача створюється декларативно. React Native дозволяє перетворити стандартні елементи VDOM на нативні візуальні елементи для кожної окремої мобільної платформи.

Розробка на React Native відбувається у модульному режимі. Програма фактично збирається з «цеглинок». Окремі модулі адаптуються та інтегруються один з одним.

Запуск програми відбувається за допомогою проміжних бібліотек. Тому створені програми однаково добре працюють на базі iOS та Android. Якщо виникає потреба внести якісь зміни в той чи інший компонент, він буде оновлений відразу для обох платформ. З функцією «Гаряче перезавантаження» (Hot Reloading), розробка та тестування інтерфейсу користувача – це легко. Завдяки цій функції програма перезавантажується щоразу, коли JS-файл зберігається.

У React Native виклик API, рендеринг зображень з URL та інші процеси дуже прості. Набагато менше часу витрачається на налаштування. Коли дані надходять з API на платформі Android, вони спочатку перетворюються на POJO-модель і лише потім використовуються в елементах UI. А ось дані JSON, отримані в React Native, зручні для JavaScript і можуть використовуватися

безпосередньо для перегляду UI. Це дозволяє полегшити веб-інтерфейс GET або POST-запитів від REST API [5].

React Native скорочує цикл розробки, дозволяє розробляти програми дуже швидко і дозволяє доставляти програми максимально швидко. Він використовує бібліотеку ReactJS UI, розроблену Facebook для інтерфейсів користувача.

Додавання апаратних можливостей пристрою у застосунок є однією з найпоширеніших вимог у наші дні. React Native дозволяє безпосередньо пов'язати плагін із власним модулем через інфраструктуру. Це призводить до більш плавного запуску програми, більш швидкого завантаження та менших вимог до пам'яті.

Огляд технологій показав відмінності між кросплатформними технологіями. Кожна технологія має унікальний набір властивостей та має свою сферу застосування.

Вибір відповідної технології стає не тривіальним завданням для розробників. Щоб показати переваги та недоліки певної технології, було визначено критерії порівняння, після чого проведено порівняльний аналіз.

Результати порівняльного аналізу відображені у таблиці 1.

Таблиця 1

### Критерії порівняння

<i>Критерії</i>	<i>Ionic</i>	<i>Flutter</i>	<i>React Native</i>
<i>Розробник та час представлення</i>	Drift Co. 2013 р.	Googl травень 2017 р.	Facebook березень 2015 р.
<i>Підтримка спільнотою</i>	сильна	Відсутня, оскільки занадто новий	Дуже сильна
<i>Мова розробки</i>	HTML, CSS, JavaScript	Dart	JavaScript + React
<i>Відкритість ресурсу</i>	Так + платні пакети	Так	Так
<i>Тип фреймворка</i>	Кросплатформний, гібридна web-розробка	Кросплатформний, Нативно-гібридна розробка	Кросплатформний, Нативно-гібридна розробка
<i>Платформи</i>	Android, IOS, WEB	Android, IOS, WEB, Googl, Fuchsia, Desktop	Android, IOS, UWP (універсальна платформа Windows)
<i>Продуктивність</i>	Середня	Висока	Висока, близька до нативної
<i>Популярні програмні продукти</i>	MarketWotch, NHS, Swokit	Alibaba, AppTre, Google Ads, Reflectly,	Facebook, Instagram, Tesla, Uber, Walmart, Airbnb

Для кросплатформної розробки мобільних застосунків усі розглянуті фреймворки встигли зарекомендувати себе серед компаній, такими, які прагнуть скоротити час та витрати на розробку. Кожен із фреймворків може бути корисним залежно від пріоритетів та потреб бізнесу.

У підсумку, для написання мобільного застосунку був обраний фреймворк React Native. Цей фреймворк поширюється безкоштовно і має відкритий вихідний код. Основною його перевагою перед аналогами є можливість випускати оновлення програми без повторного складання настановного файлу та проходження процедури публікації в магазині додатків. Крім цього, плюсами є велика та активна спільнота розробників, а також використання одного з найпопулярніших мов програмування – JavaScript.

### **Список використаної літератури**

1. Mobile & Tablet Operating System Market Share Worldwide - October 2021 [Електронний ресурс] – Режим доступу: <https://gs.statcounter.com/os-market-share/mobile-tablet/worldwide/#monthly-201910-202109-bar>.
2. Statista [Електронний ресурс] – Режим доступу: <https://www.statista.com/>.
3. Cheng, F. Build Mobile Apps with Ionic 4 and Firebase: Hybrid Mobile App Development / F. Cheng – Apress, 2018 – P.238.
4. Windmill, E. Flutter in Action / E. Windmill – Manning Publications, 2020 – P.310.
5. Nalwaya, A., Paul, A. React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Applications / A. Nalwaya, A. Paul – Apress, 2019 – P.119.