

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки
Кафедра ЕОМ

Модель проєктування мобільних застосунків

Автор:
здобувач групи СПм-23-5
Холобок Владислав

Керівник:
доцент каф. ЕОМ, к.т.н.
Філімончук Т.В.

Мета роботи

- визначити особливості та ключові відмінності між нативною та кросплатформною розробкою мобільних застосунків (обґрунтувати доцільність використання нативного або кросплатформного підходу);
- проаналізувати існуючі інструменти, фреймворки та підходи до створення мобільних рішень;
- розробити та обґрунтувати вдосконалену модель архітектури мобільного застосунку, яка враховує сучасні вимоги до UX/UI, безпеки, масштабованості та продуктивності;
- реалізувати практичний приклад нативного застосунку служби доставки з використанням Kotlin (Android) та Swift (iOS);
- на основі аналізу сформулювати висновки та рекомендації для розробників щодо вибору оптимальної стратегії створення мобільного ПЗ.



Актуальність мобільних застосунків



Зростання попиту

цифрова трансформація бізнесу



Вибір архітектури

впливає на якість, вартість, масштабування



Економічний вплив

неправильний вибір призводить до втрат



Ринкова динаміка

Android (110 млрд. завантажень),
iOS (41.5 млрд. завантажень)

3



Основні підходи до розробки

Нативна розробка

- окремо для ОС Android/iOS;
- Kotlin/Java(Android), Swift/Objective-C(iOS);
- висока продуктивність, повний доступ до функцій;
- більше часу та ресурсів.

Кросплатформна розробка

- один код для багатьох платформ;
- Flutter, React Native, Xamarin;
- швидкість розробки, зниження витрат;
- можливі втрати продуктивності.

Гібридна розробка

- вебзастосунок нативній оболонці;
- HTML, CSS, JS(Cordova, Ionic);
- швидка розробка, мінімальні витрати;
- обмежений доступ до API, низька продуктивність;

4



Нативна розробка: переваги та недоліки

- Особливості**
 Окремі застосунки для кожної ОС.
- Переваги**
 Висока продуктивність, доступ до всіх функцій, кращий UX/UI.
- Недоліки**
 Вища вартість, довший час розробки, складність оновлень.

5

Кросплатформна розробка: особливості

Переваги

- одна кодова база;
- швидке оновлення;
- зниження витрат;
- гнучкість (Web/Desktop).

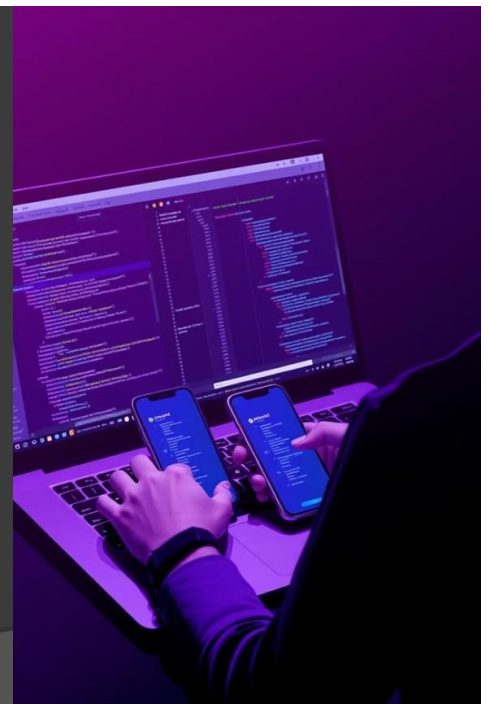
Недоліки

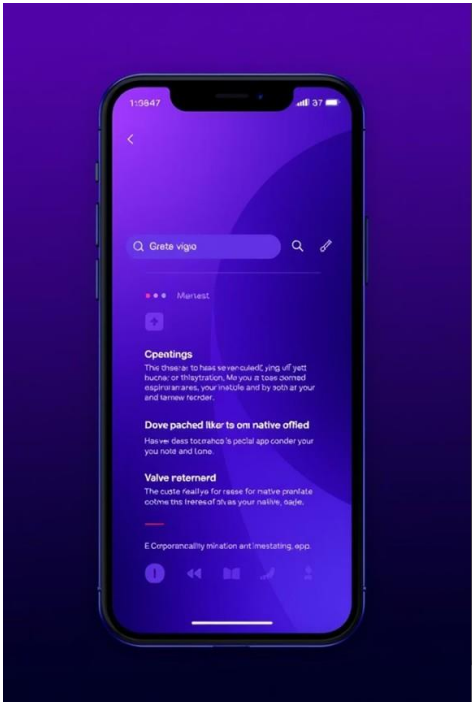
- менша продуктивність;
- обмежений доступ до функцій;
- проблеми зі стабільністю;
- UX не завжди нативний.

Підхід, що використовує один код для Android та iOS.

Фреймворки: Flutter (Dart), React Native (JavaScript), Xamarin (C#).

6





Гібридна розробка: можливості та обмеження

Суть підходу
Вебсайт всередині нативної оболонки (WebView).

- HTML, CSS, JavaScript

Платформи

- Apache Cordova
- Ionic + Capacitor

Переваги

- Єдиний код, швидкий цикл розробки, мінімальні витрати.

Недоліки

- Низька продуктивність, слабкий UX/UI, обмежений доступ до API.

7

Модифікована модель проєктування мобільних застосунків

$$M = \{OS, FE, BE, DS, UX, UI, PS, G, SDE, PN, TT, MAT\}$$

Компоненти	Розшифрування	Призначення
OS	Operating System	Платформа: Android / iOS
FE	Front-End	Клієнтська частина, UI
BE	Back-End	Серверна логіка, база даних
DS	Data Storage	Зберігання даних: локально чи в хмарі
UX	User Experience	Зручність використання
UI	User Interface	Візуальний інтерфейс
PS	Payment Systems	Платіжні системи: Google Pay, Apple Pay
G	Geolocation	Визначення місцезнаходження користувача
SDE	Secure Data Exchange	Шифрування, автентифікація, HTTPS
PN	Push Notifications	Сповіщення у реальному часі
TT	Testing Tools	Інструменти тестування
MAT	Monitoring and Analytics Tools	Аналітика, збір метрик, поведінка користувачів

Приклад нативного застосунку доставки

1

Ціль проекту

Нативний застосунок для служби доставки.

2

Функціонал

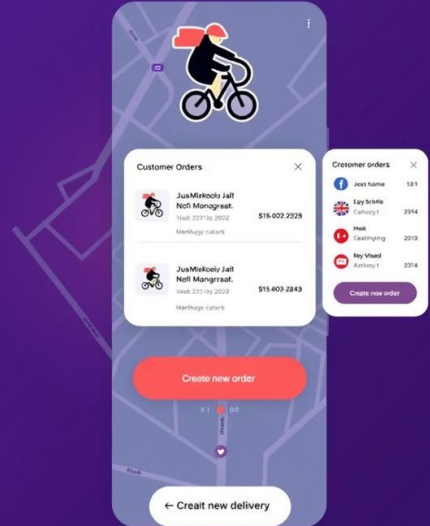
- Створення замовлень
- Перегляд списку
- Відстеження кур'єра

3

Технології

- Android: Kotlin, Jetpack Compose, Room, Google Location Services API.
- iOS: Swift, SwiftUI, Core Data, Core Location Framework.

9



Аналіз реалізації прикладу застосунку

1

Продуктивність

- Android: завантаження списку <1с, додавання <0.5с, геолокація ~10м.
- iOS: плавна анімація, миттєвий доступ, енергоефективність.

2

Зручність (UX/UI)

- Мінімістичний, інтуїтивний інтерфейс.
- Відповідність гайдлайнам Material Design та HIG.

3

Адаптивність

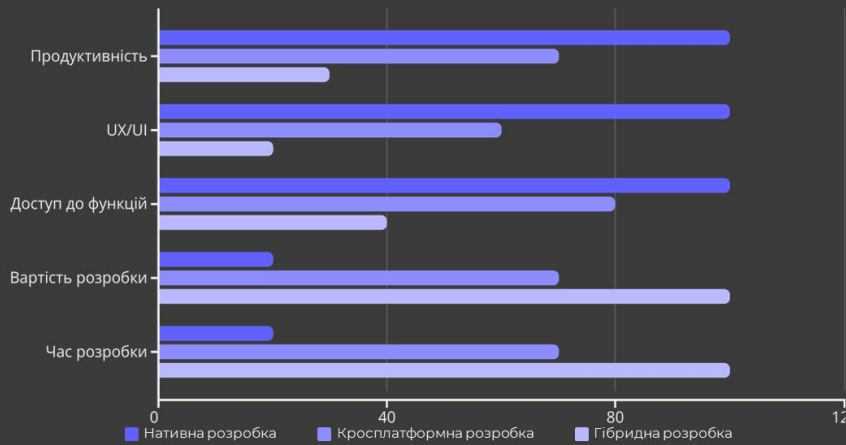
Підтримка темного режиму, різних розмірів екранів, автозаповнення.

Застосунок швидкий, стабільний та енергоефективний на обох платформах.

Висока зручність та простота використання для всіх користувачів.

10

Порівняльний аналіз підходів



Вибір залежить від критеріїв: продуктивність, UX/UI, доступ до функцій, вартість, час розробки.

11

Висновки по роботі

- проаналізовано сучасні підходи до розробки мобільних застосунків: нативний, кросплатформний, гібридний;
- Запропоновано модифіковану модель архітектури, яка включає UX/UI, безпеку, геолокацію, push-сповіщення та аналітику;
- реалізовано нативний застосунок служби доставки з підтримкою ключових функцій (замовлення, трекінг, локальне зберігання);
- нативна розробка довела свою ефективність у проєктах з високими вимогами до продуктивності та взаємодії з пристроєм;
- результати дослідження можуть бути використані для оптимізації процесів розробки у реальних проєктах.

Апробація результатів кваліфікаційної роботи: Філімончук Т.В., Колтун Ю.М., Климова І.М., Холобок В.І. Модель підходу розробки мобільних застосунків // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки, 2025. Том 36 (75) №3 (подано до друку)

12