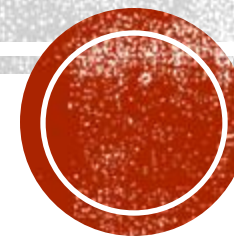


# **ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА КРОС- ПЛАТФОРМНИХ РІШЕНЬ ДЛЯ ПОБУДОВИ МОБІЛЬНИХ ЗАСТОСУНКІВ**

Підготував: Джос Олексій Кирилович

Науковий керівник: Франків Олександр Олександрович



# МЕТА

Рекомендації у виборі технології на основі порівняння та аналізу зважаючи на особливості, переваги, недоліки для певних кейсів використання



# ЕТАПИ ДОСЛІДЖЕННЯ

- Детальний опис технологій
- Опис критеріїв порівняння
- Порівняння технологій за критеріями
- Рекомендації щодо використання



# ТРОХИ СТАТИСТИКИ

Статистика росту мобільних пристроїв



Розподіл мобільних платформ



# ІСТОРІЯ КРОС-ПЛАТФОРМИ

- Web
- Progressive Web Application
- ... React Native / Flutter
- Kotlin Multiplatform



# ПРАКТИЧНА ЧАСТИНА

Створення трьох додатків з використанням **React Native/Flutter/Kotlin Multiplatform**

Функціонал:

- **CRUD** операції
- Отримання геолокації з отриманням дозволів
- Жести
- Анімація
- Вкладеність інтерфейсу



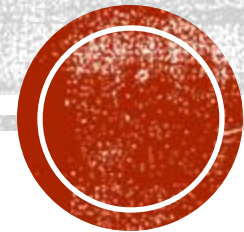
# КРИТЕРІЇ ПОРІВНЯННЯ

- Ініціалізація проекту
- Мова програмування
- Бібліотеки
- Допоміжні інструменти
- Користувацький інтерфейс
- Повторне використання коду
- Апаратні можливості
- Спільнота
- Пошук розробників
- Бюджет на розробку
- Швидкість розробки



	React Native	Flutter	Kotlin Multiplatform
Ініціалізація додатку	CLI команда	CLI команда або IDE плагін	IDE плагін або сайт з вибором залежностей
Мова програмування	JavaScript, але краще TypeScript	Dart та багато шаблонного коду	Kotlin, що дає змогу писати спільний код Android розробникам
Бібліотеки	Велика кількість, але складно знайти з підтримкою фреймворку; конфлікти версій	Багато доступних, але мала кількість для не мобільних платформ	Мала кількість, проте дозволяється використовувати платформні бібліотеки
Допоміжні інструменти	Плагіни IDE для React/Web розробки, Hot Reload, Hot Push	Плагін для IDE, Hot Reload, дерево Widgets, дебаг	Плагін для Android Studio, дебаг, запуск усіх платформ з одного місця
Користувацький інтерфейс	Перетинається нативний досвід користування Android та iOS	Власний двигун – будь який UI	Вільний вибір
Повторне використання коду	Бізнес логіка, якщо відсутні нативні залежності	Тільки в межах Flutter додатків	Немає обмежень, підключається як зовнішня бібліотека
Апаратні функції	Потрібно вивчати функціонал бібліотеки, або писати нативні виклики для кожної платформи		Нативні виклики
Пошук розробників	Велика кількість вакансій; Одна людина на проект, для апаратних функцій ще по одній людині на платформу, тобто 3 в сумі		Залучення Android/iOS команди – 2 людини

# ПОРІВНЯННЯ





# РЕКОМЕНДАЦІЇ ПО ВИКОРИСТАННЮ ТЕХНОЛОГІЙ

## React Native

- Не потрібні апаратній функції
- Велика кількість вакансій
- Hot Push
- Тільки мобільна платформа

## Flutter

- Єдиний UI
- Велика кількість готових бібліотек

## Kotlin Multiplatform

- Інтеграція у існуюче рішення
- Підтримка нових платформ
- Бізнес логіка



# ВИСНОВКИ

- Дослідив такі технології як **React Native, Flutter, Kotlin Multiplatform**
- Реалізував додатки з використанням цих технологій
- Проаналізував та описав рекомендації для застосування технологій для конкретних кейсів



ДЯКУЮ ЗА УВАГУ

