

УДК 004.42

ТЕХНОЛОГИИ ДЛЯ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ УНИВЕРСИТЕТА

Тюкало К.А., студент гр. ПИБ-162, IV курс
Научный руководитель: Пимонов А.Г., д.т.н., профессор,
Кузбасский государственный технический университет
имени Т.Ф. Горбачева, г. Кемерово

Существует два основных направления разработки мобильных приложений: нативное и кроссплатформенное [1]. Это два основных направления, у которых есть свои плюсы и минусы. О них поговорим дальше.

- Нативная разработка:
 - приложение работает быстрее;
 - доступ ко всем возможностям платформы;
 - новые возможности для разработки появляются сразу после появления нового функционала в операционной системе;
 - возможность гибко тестировать свои решения благодаря огромному количеству инструментов
 - использует уникальные привычные для пользователя особенности платформы.
- Кроссплатформенная разработка:
 - позволяет разрабатывать приложение одновременно под IOS и Android не требует при этом знания языков нативной разработки;
 - большое комьюнити ;
 - исправление ошибок происходит одновременно на двух платформах;
 - одинаковая логика приложения для всех платформ.

В мобильном приложении для работы с основной информацией, такой как расписание, не требуется больших вычислительных мощностей, поэтому кроссплатформенная разработка отлично подходит для решения данной задачи.

Крупные компании видят будущее в такой разработке и активно вкладывают в нее, поэтому на данный момент есть множество решений, позволяющих разрабатывать кроссплатформенные приложения. Каждое решение имеет свои особенности [2] (рис. 1). У меня вызывает интерес React Native за его производительность, близкую к нативной, очень большое сообщество и используемый язык программирования JavaScript (JavaScript – мультипарадигменный язык программирования, поддерживает объектно-ориентированный, императивный и функциональный стили, является реализацией языка ECMAScript). Данный фреймворк разрабатывается на протяжении 5 лет и поддерживается всеми популярными IDE (интегрированная среда разработки),

а также имеет огромное количество готовых решений, которые легко подключаются к проекту.

	React Native	Flutter	Ionic	Xamarin	PhoneGap
Язык	JavaScript + React	Dart	JavaScript, HTML, CSS + Angular, React, Vue	C# + .NET	JavaScript, HTML, CSS
Приложения	Кроссплатформенные	Кроссплатформенные	Кроссплатформенные гибридные	Кроссплатформенные	Кроссплатформенные гибридные
Первый релиз	2015	2017	2013	2011	2009
Кто разрабатывает	Facebook + сообщество	Google + сообщество	Drifty Co.	Microsoft	Adobe
Сообщество	Очень большое	Небольшое, но активно развивается	Большое	Большое	Большое
Платформы	Android, iOS, UWP	Android, iOS, Google Fuchsia, Web, Desktop	Android, iOS, Web	Android, iOS, UWP	Android, iOS, Windows Phone 8
Open source	Да	Да	Да + платные пакеты	Да + платные пакеты	Да
Инструменты фронтенда	Компоненты Native + Declarative UI	Встроенные виджеты	HTML, CSS + виджеты	Xamarin.iOS/Android или Xamarin.Forms	HTML, CSS
Производительность	Высокая, близка к нативной	Очень высокая	Средняя из-за веб-технологий	iOS/Android: высокая, близка к нативной Forms: средняя	Средняя из-за веб-технологий

Рисунок 1 – Сравнительная таблица инструментов разработки мобильных приложений

Список литературы:

1. Нативная или кроссплатформенная разработка – что лучше? [Электронный ресурс]. – Режим доступа: <http://wnfx.ru/nativnaya-ili-krossplatformennaya-razrabotka-cto-luchshe/>, свободный (дата обращения: 14.03.2020).
2. Сравнение фреймворков для кроссплатформенной мобильной разработки: React Native, Flutter, Ionic, Xamarin и PhoneGap [Электронный ресурс]. – Режим доступа: <https://tproger.ru/translations/cross-platform-frameworks-for-mobile-development/#7>, свободный (дата обращения: 14.03.2020).