

РАЗРАБОТКА КРОССПЛАТФОРМЕННЫХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Прохоренко Артём Сергеевич

*магистрант, Белорусский государственный университет информатики и радиоэлектроники
Беларусь, г. Минск
E-mail: acprohor@gmail.com*

Якимович Алексей Владимирович

*магистрант, Белорусский государственный университет информатики и радиоэлектроники,
Беларусь, г. Минск
E-mail: yaki.alex123@gmail.com*

Коваль Артём Вадимович

*магистрант, Белорусский государственный университет информатики и радиоэлектроники,
Беларусь, г. Минск
E-mail: artyom.koval.139@gmail.com*

Нахратьянц Дмитрий Артемьевич

*магистрант, Белорусский государственный университет информатики и радиоэлектроники,
Беларусь, г. Минск
E-mail: dirmidante@gmail.com*

Разработка мобильных приложений достаточно трудоёмкий процесс. Существует множество аспектов, которые стоит продумать до начала разработки. Структура компонентов и их логическая связь, внешний вид элементов и многое другое. Также необходимо сделать выбор под какую мобильную операционную систему будет разработано приложение. Основными используемыми мобильными операционными системами на сегодняшний день являются всего две: iOS и Android.

iOS — мобильная операционная система для смартфонов, планшетов и носимых проигрывателей, разрабатываемая и выпускаемая американской компанией Apple. С 2014 года включена поддержка автомобильных мультимедийных систем Apple CarPlay [1]. Особенностью iOS является тот факт, что данная система выпускается исключительно для устройств, произведённых компанией Apple [2].

Android — операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, фитнес-браслетов, игровых приставок, ноутбуков, телевизоров и других устройств. В 2015 году также появилась поддержка автомобильных мультимедийных систем [3].

У каждой системы есть свои положительные и отрицательные стороны. Одним из наиболее ярких отличий является процент использования системы в мире. Android на сегодняшний день используют около 72% от общего числа пользователей, тогда как у iOS всего 27% [4].

Основные преимущества системы Android: открытость системы, большое количество встроенных сервисов Google, система может работать на большом количестве устройств, современный интерфейс, регулярные обновления системы, множество функций, которые отсутствуют у конкурентов, возможность тонкой настройки системы под каждого пользователя, поддержка многопользовательского режима, возможность устанавливать приложения не только из официального магазина приложение Google Play.

К недостаткам системы можно отнести следующее: излишнее количество собираемой о пользователе информации, возможность установки приложений из неофициальных источни-

ков способствует пиратству и делает систему более уязвимой, медленное распространение новых версий системы, слабая поддержка старых устройств.

Основные преимущества системы iOS: крупный магазин приложений с качественным контентом; оптимизация системы; быстрое исправление ошибок; регулярное обновление системы; поддержка старых устройств; продуманный и красивый дизайн.

К основным недостаткам относится: закрытая операционная система, в связи с чем нельзя свободно обмениваться файлами с устройствами, на которых установлена другая операционная система; отсутствует множество функций, которые есть в Android, например поддержка многооконного режима; сложная работа с файловой системой;

Если разработчики в процессе написания приложения пользуются принятым для конкретной платформы языком программирования, будь то Objective-C и Swift для iOS или Java или Kotlin для Android, такое приложение будет называться нативным.

Даже при том, что система Android имеет большее распространение, магазин приложений в iOS считается более приоритетным так как там сконцентрирована более платежеспособная аудитория. Для максимального охвата аудитории имеет смысл создать приложение под каждую из этих систем. Однако это сильно увеличит объём работ при создании и разработке. Существует возможность создать кроссплатформенное приложение, которое будет одинаково выглядеть и работать как на iOS, так и на Android. Такой подход ускорит разработку и может значительно увеличить охват аудитории разрабатываемого продукта.

Cordova — мобильная среда разработки приложений, которая позволяет создавать приложения для мобильных устройств с помощью CSS3, HTML5 и JavaScript, вместо того, чтобы использовать конкретные платформы Android и iOS [5].

Одно из самых популярных направлений в кроссплатформенном программировании, которое часто называют PhoneGap. Фактически создаётся мобильный сайт, который «обращивается» небольшим платформенным кодом, транслирующим вызовы от системы к приложению и обратно.

PWA — технология от Google. Это те же самые веб-приложения, но за счёт использования определённых технологий (в первую очередь это так называемые Service Worker — работающие в фоновом режиме скрипты, и Web App Manifest — описание веб-приложения в понятном для мобильной системы виде) они без обёртки из PhoneGap могут работать как нативные. Они могут устанавливаться на домашний экран в обход магазина приложений, работать без подключения к интернету, работать с уведомлениями в системе и другими нативными функциями.

Xamarin — платформа компании Microsoft. Для разработки приложений на этой платформе применяется язык программирования C# и среда разработки Visual Studio. В результате можно получить нативные приложения.

React Native — платформа разработанная компанией Facebook. Приложения пишутся на JavaScript и используются CSS-подобные стили. Интерфейс приложения используют нативные элементы, а код интерпретируется уже в платформе что придаёт ему гибкость.

Немного подробнее остановимся на React Native. Данная платформа достаточно молодая, первый релиз состоялся в марте 2015 года [6]. Приложения, создаваемые на платформе, строятся из компонентов платформы – это нативные модули, которые завернуты в React-компоненты. К сожалению, в данном случае полной кроссплатформенности не достигается. Некоторые компоненты специфичные для своей системы. Например, для iOS это TabBarIOS а для Android ToolBarAndroid. Такие компоненты, которые имеют специфичное для своей системы поведение приходится использовать с уточнением на какой операционной системе запущено приложение либо необходимо избегать использование таких компонентов.

Преимуществом при использовании React Native является тот факт, что можно использовать возможности самого React такие как принцип взаимодействия компонентов между собой. Также в некоторой степени положительным фактором является то, что вся логическая часть приложения пишется на JavaScript и стилизуется CSS-подобными стилями.

Так как платформа относительно молодая ещё не все нативные модули поддерживаются и реализованы на должном уровне. Но при этом есть возможность подключать эти самые модули к приложению.

Заключение:

Проанализировав рынок мобильных операционных систем, можно сказать, что разрабатывать приложение часто необходимо для двух самых популярных систем для охвата максимальной аудитории. Для того чтобы создать приложение для каждой системы приходится затрачивать большие ресурсы. Разумеется, такие приложения имеют большие преимущества. Они более отказоустойчивые, имеют лучшую скорость работы и быстрый отклик интерфейса, внешний вид более подходящий под концепцию системы, под которую разрабатывалось приложение. На сегодняшний день существует несколько платформ которые позволяют создавать кроссплатформенные приложения. С их помощью можно сократить время разработки и количество затраченных на создание приложения ресурсов, но при этом в некоторой степени страдает качество. В результате можно сделать вывод что во многих случаях создание кроссплатформенного приложения это неплохая практика, которая значительно упростит процесс разработки, но применение данного подхода подойдёт не во всех случаях.

Список литературы:

1. iOS 13 [Электронный ресурс]. — Режим доступа: www.url:https://www.apple.com/ru/ios/ios-13/. — 06.04.2020.
2. Apple Developer Documentation [Электронный ресурс]. — Режим доступа: www.url:https://developer.apple.com/documentation/. — 06.04.2020.
3. Documentation for app developers [Электронный ресурс]. — Режим доступа: www.url:https://developer.android.com/docs. — 06.04.2020.
4. Browser Market Share Worldwide Mar 2019 – Mar 2020 [Электронный ресурс]. — Режим доступа: www.url:https://gs.statcounter.com/. — 06.04.2020.
5. Apache Cordova 2020 [Электронный ресурс]. — Режим доступа: www.url:https://cordova.apache.org/. — 06.04.2020.
6. React Native [Электронный ресурс]. — Режим доступа: www.url:https://reactnative.dev/. — 06.04.2020