

РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА FLUTTER НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ DART

Шингерей А.Н.

*Белорусский национальный технический университет,
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент

Аннотация. Рассматривается и обосновывается разработка мобильных приложений с использованием фреймворка FLUTTER на языке программирования DART для повышения качества, комфортности и сокращения времени получения необходимых информационных ресурсов пользователями.

В наше время современными мобильными телефонами владеют большинство людей, от чего возникает острая необходимость разработки мобильного представления для уже существующих проектов. Поскольку это за собой влечёт более быстрый способ получения информации, удобства и привлечения новых пользователей за счёт быстрого распространения.

В современном мире у многих день начинается с того, что мы выключаем будильник на нашем мобильном телефоне и затем просматриваем уведомления, которые пришли за ночь. Мы пользуемся телефонами для любых повседневных дел: занимаемся поиском быстрой и доступной информации в сети Интернет, ищем нужный адрес, ведём расписание тренировок, планируем семейный бюджет. Мобильные приложения – неотъемлемая часть нашей жизни. По данным компании Nielsen, пользователи проводят в них в среднем 30 часов в месяц. Приложения стали нашими постоянными помощниками в микромоментах «хочу узнать», «хочу пойти», «хочу сделать» и «хочу купить».

Приложения помогают компаниям развивать взаимоотношения с клиентами. Эффективная стратегия по продвижению приложений может способствовать развитию электронной торговли, повысить эффективность офлайн мероприятий (например, акций в магазине) и укрепить взаимоотношения с покупателями. Потребители все чаще используют приложения и сайты для мобильных устройств, и маркетологам необходимо это учитывать. На сегодняшний день, основную классификацию мобильных телефонов делят на: BlackBerry, SymbianOS, Windows Mobile, Android, IOS и иные. ОС Android и IOS — считается наиболее популярными поскольку занимают 98 % всего рынка мобильных устройств. В процессе проектирования выбор пал на платформу Android. Поскольку при анализировании рынка стало понятно, что фактически, пользователей Android больше из-за дороговизны продуктов на ОС IOS.

Пользовательский интерфейс мобильного приложения является важной частью системы. Ландшафт разработки приложений кишит технологиями и решениями для создания пользовательских интерфейсов. Наряду с собственными разработками, существует множество кроссплатформенных, веб-, гибридных и даже не кодированных инструментов для создания интерфейсов приложений. Хотелось рассмотреть конкретный инструмент для этого это фреймворк Flutter.

Flutter — это комплект разработки пользовательского интерфейса с открытым исходным кодом (UIToolkit) от Google для разработки скомпилированных в собственном коде приложений. В настоящее время он предлагает пользователям возможность компилировать приложения для веб-платформ, платформ iOS, Android, Linux, Windows, Mac и Google Fuschia. Первая версия Flutter, известная как «Sky», работала в операционной системе Android. Он был представлен на саммите разработчиков Dart в 2015 году как решение, способное обеспечить рендеринг со скоростью 120 кадров в секунду. Предварительная версия Flutter Release Preview 2 была анонсирована компанией Google во время презентации Google Developer Days в Шанхае. Это была последняя крупная версия выпуска Flutter перед Flutter 1.0. На мероприятии Flutter Live 4 декабря 2018 года была запущена Flutter 1.0. Flutter 1.12 был запущен на мероприятии Flutter Interactive 11 декабря 2019 года. 6 мая 2020 года были выпущены Flutter версии 1.17.0 и Dart SFK версии 2.8. В этой версии была добавлена поддержка Metal API. Эта функция значительно повысила производительность устройств iOS, предлагая новые виджеты материалов, инструменты отслеживания сети и многое другое [1-4].

Основные компоненты Flutter включают язык программирования Dart, движок Flutter и библиотеку Foundation.

Приложения Flutter разрабатываются на языке программирования Dart. Он использует многие из наиболее продвинутых функций этого языка. Flutter работает на виртуальной машине Dart, которая имеет механизм сборки во время выполнения, в macOS, Linux, Windows и в рамках проекта Flutter Desktop Embedding. Flutter использует JIT-сравнение во время написания и отладки приложения. Это позволяет выполнять горячую перезагрузку, позволяя пользователям вносить изменения исходного файла в работающее приложение. Flutter расширяет поддержку виджетов с отслеживанием состояния при горячей перезагрузке. Это позволяет отражать изменения исходного кода в запущенных приложениях без потери состояния или перезапуска. Релизные версии приложения

Flutter скомпилированы с использованием ранней сборки (AOT) на iOS и Android. Это обеспечивает высокую производительность платформы на мобильных устройствах.

Движок Flutter, который в основном был написан на C ++, предлагает поддержку низкоуровневого рендеринга графической библиотекой Skia от Google. Кроме того, он может взаимодействовать с SDK для конкретных платформ, например, с теми, которые предоставляются iOS и Android. Flutter Engine — это портативная среда выполнения для размещения приложений Flutter. Движок реализует библиотеки Flutter, такие как файловый и сетевой ввод-вывод, анимацию и графику, архитектуру, поддержку специальных возможностей, а также несколько сред выполнения Dart и компонентов сборки. Большинство разработчиков используют Flutter Framework для взаимодействия с Flutter. Фреймворк Flutter предлагает гибкую структуру, а также виджеты макета, основы и платформы.

Библиотека Foundation, написанная на Dart, предоставляет множество фундаментальных функций и классов, которые используются для создания приложений Flutter, включая API-интерфейсы, которые облегчают взаимодействие с движком.

Библиотека также содержит виджеты для конкретного дизайна с реализациями Android и iOS.

Фреймворк Flutter имеет два набора виджетов, которые соответствуют определенным языкам дизайна. Виджеты материального дизайна создают реализации визуальной идентификации Google, а виджеты Cupertino реализуют рекомендации Apple по интерфейсу человека для iOS.

В качестве примера написанных приложений на Flutter можно рассмотреть такие приложения как: Alibaba; Tencent; Google Ads.

Доступ к встроенным функциям. Функции, специфичные для платформы, такие как геолокация и камера, требуют доступа к встроенным функциям. Эти функции должны быть реализованы на родных языках. Flutter дает своим пользователям ощущение разработки собственной платформы. Они могут повторно использовать свой код Swift, Java и Objective-C для доступа к SDK и встроенным функциям на Android и iOS.

Подобно Native App Performance — производительность приложения является важным фактором в обеспечении благоприятного взаимодействия с пользователем. Уровень производительности, предлагаемый Flutter, в основном соответствует уровню собственных приложений и сложной анимации пользовательского интерфейса. Flutter не полагается на промежуточное представление или интерпретацию кода. Приложение Flutter встроено непосредственно в машинный код, что сокращает количество ошибок производительности в процессе интерпретации.

Горячая перезагрузка — горячая перезагрузка — одна из самых важных функций Flutter. Это позволяет разработчикам незамедлительно наблюдать за всеми изменениями кода, которые они внесли. Изменения видны за считанные секунды, что позволяет разработчикам с легкостью расширять функции, исправлять ошибки и экспериментировать. HotReload упрощает совместную работу разработчиков и дизайнеров приложений при мгновенном тестировании внешнего вида приложений.

Меньше кода — Flutter был написан в основном на языке программирования Dart. Dart — объектно-ориентированный язык со строгой типизацией. Flutter похож на React Native и имеет реактивный и декларативный стиль программирования. Flutter не требует моста JavaScript для увеличения времени запуска и производительности приложения. Dart предлагает компиляцию AOT вместе с компиляцией Just-in-Time. Flutter использует JIT-компиляцию для улучшения рабочего процесса разработки. Это достигается за счет облегчения горячей перезагрузки и обновления пользовательского интерфейса во время разработки без создания полной новой сборки.

Собственный движок рендеринга — Flutter позволяет разработчикам достигать большего, чем любая другая платформа. Он дает хорошие результаты с мощными фреймворками, такими как мощный кроссплатформенный механизм рендеринга. Flutter использует Skia для самостоятельного рендеринга в инфраструктуру платформы. Его движок позволяет запускать пользовательский интерфейс, созданный с помощью Flutter, практически на любой платформе. Пользователям Flutter не нужно настраивать пользовательский интерфейс для переноса платформы для упрощения разработки.

Были проанализированы технологии создания мобильного приложения, а также некоторые шаги, которые необходимо учитывать при разработке. Проведён подробный анализ существующих платформ и языков программирования. Рассмотрен язык программирования Dart и Фреймворк Flutter.

Платформа Android взята за базис, исходя из текущего количества пользователей мобильных телефонов на её основе, что делает данную систему актуальной.

Список использованных источников:

1. Заметти Франк FLUTTER на практике / Редактор – Мовчан Д.А. / Переводчик – Тищенко А.С. Фрэнк Заметти – ДМК Пресс, 2020 – 328с.
2. Официальный сайт с документацией Flutter [Электронный ресурс]. – Режим доступа : <https://flutter.dev/>.
3. Dart&Flutter — статьи, примеры, уроки. [Электронный ресурс]. – Режим доступа : <https://dartflutter.ru/>.
4. Данные компании Nielsen [Электронный ресурс]. – Режим доступа: <https://www.nielsen.com/ru/ru/>.