



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.4'2

## РАЗРАБОТКА МУЛЬТИПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ВЕБ-ТЕХНОЛОГИЙ

**Сидоров И.А.**

Филиал ФГБОУ ВО «Национальный исследовательский университет «МЭИ» в г. Смоленске, Россия, (214013, г. Смоленск, Энергетический проезд, 1), e-mail: igorsidorov770@gmail.com.

Статья посвящена разработке мультиплатформенных приложений (веб-приложения, мобильные приложения и десктоп-приложения). Рассмотрены категории разработчиков, а также технологии, которые они используют. Предложен комплекс технологий для разработки мультиплатформенных приложений усилиями веб-разработчиков.

Ключевые слова: веб-приложения, мобильные приложения, десктоп-приложения.

## DEVELOPMENT OF MULTIPLATFORM APPLICATIONS USING WEB TECHNOLOGIES

**Sidorov I.A.**

Smolensk Branch of the National Research University "Moscow Power Engineering Institute", Smolensk, Russia (214013, Smolensk, Energeticheskyy proezd, 1), e-mail: igorsidorov770@gmail.com.

The article is devoted to the development of multi-platform applications, that is, applications that work as web applications, mobile applications and desktop applications. The categories of developers, as well as the technologies that they use, are examined. Technologies for developing multi-platform applications by the efforts of web developers are proposed.

Keywords: web applications, mobile applications, desktop applications.

### Введение

В настоящее время большинство приложений являются мультиплатформенными, имеющими одновременно веб-мобильную и десктоп-версии приложения.

Базовым сценарием разработки приложения, как правило, является разработка веб-версии приложения, при этом, в какой-то момент появляется необходимость в разработке мобильной версии приложения и десктоп-версии приложения. Такой переход осложняется рядом факторов. Во-первых, построение слоя для общения с базой данных (БД) иногда определяется требованиями конкретной разрабатываемой фронт-части. Во-вторых, итеративно-инкрементальный подход к разработке ПО, зачастую, обуславливает появление

неоптимальных решений в коде, требующих реализации особенных алгоритмов взаимодействия с БД. В-третьих, поиск и подбор специалистов для десктоп- или мобильной разработки может быть длительным, а найденные специалисты – недостаточно квалифицированными для выполнения поставленных задач. В-четвертых, разработка отдельных приложений предполагает не только разработку UI, но и разработку слоя бизнес-логики, а также его тестирования на соответствие бизнес-логике уже реализованных приложений. Эти операции займут значительный ресурс разработчиков и могут сильно отодвинуть вывод в продуктив искомого функционала.

Одним из вариантов решения обозначенной проблемы может стать разработка мобильной и десктоп-версий приложения усилиями веб-разработчиков.

### **1. Категории разработчиков**

Для начала необходимо рассмотреть какие категории разработчиков вообще есть.

На данный момент разработчиков программного обеспечения можно разделить на следующие категории:

- веб-разработчики;
- мобильные разработчики;
- разработчики десктоп-приложений.

### **2. Веб-разработчики**

Веб-разработчиков можно разделить на два типа:

- Front-end;
- Back-end.

Front-end разработчики создают клиентскую сторону пользовательского интерфейса, предназначенную для программно-аппаратной части сервиса.

Front-end разработчики в основном используют следующие технологии:

- HTML;
- CSS;
- JavaScript.

HTML (от английского HyperText Markup Language – язык гипертекстовой разметки) – это стандартизированный язык разметки документов в интернете. Описание большинства веб-страниц состоит из разметки на языке HTML. Язык разметки HTML сначала интерпретируется браузером, а затем полученный в результате интерпретации результат отображается на экране монитора компьютера или мобильного устройства [1].

CSS (от английского Cascading Style Sheets – каскадные таблицы стилей) – это формальный язык описания внешнего вида документов, которые написаны с использованием языка разметки. Преимущественно используется как средство оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к XML подобным документам, например XUL или SVG [2-5].

JavaScript (JS) – это мультипарадигменный язык программирования. Он поддерживает императивный, объектно-ориентированный и функциональный стили. Является реализацией языка ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам [6-8].

В свою очередь Back-end разработчики создают программно-аппаратную часть сервиса.

Выбор технологий у Back-end разработчиков велик, это обусловлено тем, что любой язык программирования, который поддерживается CGI, может быть использован для разработки.

CGI (Common Gateway Interface) – это стандарт интерфейса, который служит для связи внешней программы с веб-сервером (Apache или Nginx). Программу, которая работает по такому интерфейсу совместно с веб-сервером, обычно называют шлюзом. Сам протокол разработан таким образом, чтобы можно было использовать любой язык программирования, который может работать со стандартными устройствами ввода / вывода. А это умеет практически любой язык программирования и даже сама операционная система [9].

Список технологий поддерживаемый CGI, которые часто используются Back-end разработчиками:

- PHP совместно с фреймворком Laravel;
- Python совместно с фреймворком Django;
- JavaScript совместно с фреймворком Express;
- Rube совместно с фреймворком Ruby on Rails;
- C# совместно с фреймворком ASP.NET;
- Java совместно с фреймворком Spring и др.

Помимо этого, Back-end разработчики работают с базами данных за счет языка SQL и одной из систем управления базами данных, например: MySQL, PostgreSQL и др.

### **3. Мобильные разработчики**

Мобильные разработчики – это специалисты, разрабатывающие программные приложения для различных мобильных устройств: планшетов и смартфонов.

Мобильных разработчиков можно разделить на два типа:

- разрабатывают под операционную систему Android;
- разрабатывают под операционную систему iOS.

Мобильные разработчики, которые разрабатывают под операционную систему Android используют языки программирования Java и Kotlin.

Java – это строго типизированный объектно-ориентированный язык программирования. Язык был разработан компанией Sun Microsystems (в последующем купленной компанией Oracle). Приложения на Java обычно транслируются в байт-код. Этот байт-код позволяет работать приложениям на любой компьютерной архитектуре с помощью виртуальной Java-машины (JVM).

Kotlin – это статически типизированный язык программирования, работающий поверх виртуальной Java-машины (JVM). Язык разрабатывается компанией JetBrains. Язык также может компилироваться в JavaScript, и в исполняемый код ряда платформ через специальную инфраструктуру LLVM. Язык был назван в честь острова Котлин расположенном в Финском заливе, на котором располагается город Кронштадт. Авторы ставили целью создать язык более типобезопасный и лаконичный, чем Java, и более простым, чем Scala.

Мобильные разработчики, которые разрабатывают под операционную систему iOS используют языки программирования Objective-C и Swift.

Objective-C – это компилируемый объектно-ориентированный язык программирования, в основном используемый корпорацией Apple, построенный на основе языка C и парадигм

заложенных Smalltalk. В частности, объектная модель построена в стиле Smalltalk – то есть объектам посылаются специальные сообщения. Язык Objective-C является надмножеством языка C, это приводит к тому, что код на языке программирования C полностью понятен компилятору Objective-C. Компилятор Objective-C входит в состав GCC и доступен на большинстве основных платформ. Язык используется в первую очередь для GNUstep и Mac OS X (Cocoa). GNUstep – это реализация объектно-ориентированного интерфейса OpenStep. Также язык программирования используется для iOS (Cocoa Touch).

Swift – это открытый мультипарадигмальный компилируемый язык программирования общего назначения. Создан компанией Apple в первую очередь для разработчиков под macOS и iOS. Swift может работать с фреймворками Cocoa и Cocoa Touch. Также язык совместим с кодовой базой Apple, которая была написана на Objective-C. Swift задумывался быть более легким для чтения и устойчивый к ошибкам допускаемыми программистами, нежели предшествовавший ему язык программирования Objective-C. Программы на Swift компилируются при помощи LLVM, входящей в интегрированную среду разработки Xcode 6 и выше. Swift может использовать райнтам Objective-C, что делает возможным одновременное использование обоих языков (а также языка C) в рамках одной программы.

#### **4. Веб-разработчики в роли мобильных разработчиков**

Современные веб сильно изменился, сейчас есть куча инструментов основанных на JavaScript, которые позволяют веб-разработчикам создавать мобильные приложения:

- PhoneGap;
- Apache Cordova;
- Ionic;
- Meteor;
- NativeScript;
- Flutter;
- React Native.

PhoneGap – это бесплатный open-source фреймворк для создания мобильных приложений, созданный компанией Nitobi Software. Позволяет создать мобильные приложения используя HTML, CSS и JavaScript без необходимости знания «родных» языков программирования, например Java и Kotlin для Android, и языков Objective-C и Swift для iOS. Готовое приложение компилируется в виде установочных пакетов для каждой мобильной операционной системы.

Apache Cordova – это мобильная среда разработки приложений, первоначально разработанная компанией Nitobi. Компания Adobe Systems приобрела Nitobi в 2011 году и провела ребрендинг PhoneGap, после чего выпустила специальную версию с открытым исходным кодом программного обеспечения под названием Apache Cordova. Apache Cordova позволяет создавать мобильные приложения с помощью HTML, CSS и JavaScript вместо того, чтобы использовать конкретные платформы API, такие как IOS, Windows Phone или Android. Это обеспечивается за счет преобразования из HTML, CSS и JavaScript в код, который платформа воспринимает как элемент web. Это расширяет возможности HTML и JavaScript для работы с различными устройствами. В результате приложения являются гибридными, то есть они ни являются по-настоящему мобильными приложениями (Так как вся генерация

макета приложения осуществляются с помощью WebView). Также приложения имеют доступ к API устройств.

Ionic – это полный набор средств для разработки (SDK) с открытым исходным кодом. Предназначенные для разработки гибридных мобильных приложений, созданный Беном Сперри, Максом Линчем и Адамом Брэдли из Drifty Co. в 2013 году. Оригинальная версия Ionic была выпущена в 2013 году и построена на основе Apache Cordova и JavaScript-фреймворка AngularJS.

Meteor – это веб-платформа на языке JavaScript, предназначенная для разработки веб-приложений, а также мобильных приложений. Данная веб-платформа для связи с браузерами использует протокол Distributed Data Protocol, работающий на WebSocket.

NativeScript – это фреймворк с открытым исходным кодом, разрабатываемый Telerik. Фреймворк предназначен для разработки приложений под Android и iOS. Приложения NativeScript разрабатываются с помощью Javascript или TypeScript. В NativeScript реализована полная поддержка JavaScript-фреймворка Angular. Мобильные приложения, построенные с NativeScript, имеют полный доступ к API платформы так, будто они были разработаны в Android Studio или в XCode. Также разработчики могут использовать сторонние библиотеки с таких ресурсов, как Maven, Cocoapods, Android Arsenal и npm.js, без использования дополнительных прослоек.

Flutter – это полный набор средств для разработки (SDK) с открытым исходным кодом для создания мобильных приложений от компании Google. Он используется для разработки мобильных приложений под Android и iOS, а также это единственный способ разработать приложение для Google Fuchsia.

React Native – это JavaScript-библиотека для разработки кроссплатформенных приложений, в частности для разработки приложений для Android и iOS. Она была разработана программистами из компании Facebook и открыта для всех желающих. Она базируется на JavaScript-библиотеке React. Это простая и производительная библиотека, построенная на основе компонентов. Вместо того чтобы предоставлять полный набор инструментов для создания приложения, она позволяет выбирать, что и как реализовать в моделях данных, серверных вызовах и других функциях приложений. Но самое большое преимущественно – это статическая ментальная модель, которую она обеспечивает. Для оформления внешнего вида приложений вместо CSS она использует Yoga [10-13].

PhoneGap, Apache Cordova, Ionic и Meteor используют WebView. WebView – это мини-браузер, который позволяет запускать внутри себя HTML, CSS и JavaScript. Но так как это всего лишь браузер, приложения в нем работают хуже, чем «родные». Работают они хуже потому, что приложения, работающие в WebView, не экономят ресурсы мобильного устройства, что приводит к его быстрой разрядке и перегреву батареи.

В свою очередь NativeScript, Flutter и React Native не работают в WebView. Но NativeScript не очень популярный, а Flutter использует язык программирования Dart созданный Google, который позиционируется как замена JavaScript. На деле веб-разработчикам, знающим JavaScript не хочется переучиваться на Dart.

Поэтому React Native является лучшим инструментом веб-разработчика для разработки мобильных приложений.

С помощью React Native созданы такие мобильные приложения как Facebook, Instagram, Skype, Discord, Uber и другие известные мобильные приложения.

## **5. Разработчики десктоп-приложений**

Разработчики десктоп-приложений – это специалисты, разрабатывающие программные приложения для различных десктоп устройств: ноутбуков и персональных компьютеров.

Разработчиков десктоп-приложений можно разделить на три типа:

- разрабатывают под операционную систему Windows;
- разрабатывают под операционную систему MacOS
- разрабатывают под различные дистрибутивы операционной системы Linux.

Разработчики десктоп-приложений имеют большое количество языков программирования для разработки программного обеспечения. Например:

- C#;
- Java;
- C++;
- Python;
- Delphi.

## **6. Веб-разработчики в роли разработчиков десктоп-приложений**

Но сейчас есть куча инструментов основанных на JavaScript, которые позволяют веб-разработчикам создавать десктоп-приложения:

- Electron;
- NW.js;
- AppJS.

NW.js (также известен как node-webkit) – это JavaScript-фреймворк, который позволяет создавать кроссплатформенные десктоп-приложения для MacOS, Linux и Windows при помощи HTML, CSS, JavaScript, Node.js и V8.

AppJS – это движок Node.js, к которому в качестве одного из модулей пристроен движок WebKit из Chromium. AppJS позволяет создавать кроссплатформенные десктоп-приложения для операционных систем MacOS, Linux и Windows при помощи HTML, CSS и JavaScript.

Electron – это библиотека с открытым исходным кодом, разработанная компанией GitHub, для создания кроссплатформенных приложений с помощью следующих веб-технологий: HTML, CSS и JavaScript. Electron достигает этого путем объединения Node.js и Chromium в единую среду выполнения, а приложения могут быть собраны для работы под операционные системы Mac, Linux и Windows [14].

Но NW.js и AppJS сильно отстают по возможностям от Electron. Поэтому Electron является лучшим инструментом веб-разработчика для разработки десктоп-приложений.

С помощью Electron созданы такие десктоп-приложения как Skype, WhatsApp, Discord, Slack, Visual Studio Code и другие известные десктоп-приложения.

## **7. Подбор технологий для мультиплатформенной разработке программного обеспечения**

Рассмотрев категории разработчиков и инструменты, которые они используют мы приходим к выводу, что для разработки мобильных приложений с использованием веб-технологий лучше подходит React Native. А для разработки десктоп-приложений с использованием веб-технологий лучше подходит Electron.

Допустим, что перед нами стоит задача разработать веб-приложение, мобильное приложение и десктоп приложение, которые будут работать с программно-аппаратной частью сервиса за счет взаимодействия с API. Программно-аппаратная часть сервиса будет реализована на PHP с использованием PHP-фреймворка Laravel, а в качестве системы управления базами данных будет взята MySQL.

PHP – это скриптовый язык общего назначения, в основном применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Laravel – это бесплатный PHP-фреймворк с открытым кодом, предназначенный для разработки динамических веб-сайтов с использованием архитектурной модели MVC [15].

MySQL – это свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, но также и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Благодаря одному из таких заказов почти в самых ранних версиях MySQL появился механизм репликации. Гибкость MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, которые поддерживают полнотекстовый поиск, так и таблицы InnoDB, которые поддерживают транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря своей открытой архитектуре и лицензии GPL, в СУБД MySQL постоянно появляются новые типы таблиц.

Веб-приложение будет разработано с использованием HTML, CSS и JavaScript, а также библиотеки React. Но для управления состоянием данных также потребуется Redux и React Redux. Также для работы с асинхронными операциями потребуются Redux Thunk и Redux Saga. А для навигации по приложению React Router.

React – это JavaScript-библиотека, используемая совместно с HTML и CSS, для создания пользовательских интерфейсов на разных платформах, которая была разработана программистами из компании Facebook и открыта для всех желающих. Это простая и производительная библиотека, построенная на основе компонентов. Вместо того чтобы предоставлять полный набор инструментов для создания приложения, она позволяет выбирать, что и как реализовать в моделях данных, серверных вызовах и других функциях приложений. Также она реализует виртуальную DOM – прослойку между вашей программой и фактической DOM браузера. Виртуальная DOM позволяет эффективно обновлять фактическую DOM с использованием алгоритма быстрой сверки. Виртуальная DOM обеспечивает отличную производительность, но самое большое преимущественно – это статическая ментальная модель, которую она обеспечивает [16-20].

Redux – это JavaScript-библиотека с открытым исходным кодом, предназначенная для управления состоянием приложения. Чаще всего используется в связке с JavaScript-библиотекой React или JavaScript-фреймворком Angular для разработки клиентской части. Содержит ряд инструментов, позволяющих значительно упростить передачу данных хранилища через контекст [21].

React Redux – это JavaScript-библиотека, предоставляющая удобный API для работы с Redux совместно с React [22].

Redux Thunk – это библиотека для удобной работы с асинхронными операциями при использовании Redux.

Redux Saga – это библиотека для удобной работы с асинхронными операциями, основанная на JavaScript-генераторах [23].

React Router – это JavaScript-библиотека, предназначенная для навигации приложений, написанных с использованием React [24].

Для взаимодействия веб-приложения с API программно-аппаратной частью сервиса можно воспользоваться либо REST, либо GraphQL.

REST (от английского Representational State Transfer – передача состояния представления) – это архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, которые учитываются при проектировании распределённой гипермедиа-системы. В определённых случаях это приводит к повышению производительности и упрощению архитектуры. В широком смысле компоненты в REST взаимодействуют наподобие взаимодействия клиентов и серверов в сети Интернет. REST является альтернативой RPC. В сети Интернет вызов удалённой процедуры может представлять собой обычный HTTP-запрос (обычно «POST» или «GET»; такой запрос называют «REST-запрос»), а необходимые данные передаются в качестве параметров запроса. Для веб-служб, построенных с помощью REST (то есть не нарушающих накладываемых им ограничений), применяют термин «RESTful». В отличие от веб-сервисов (веб-служб) на основе протокола SOAP, не существует «официального» стандарта для RESTful веб-API. Дело в том, что REST является архитектурным стилем, в то время как SOAP является протоколом. Несмотря на то, что REST не является стандартом сам по себе, большинство RESTful-реализаций используют такие стандарты, как HTTP, XML, URL и JSON.

GraphQL – это язык запросов и манипулирования данными с открытым исходным кодом для API. Также есть среда GraphQL для выполнения запросов с существующими данными. GraphQL был разработан внутри компании Facebook в 2012 году, но опубликован был только в 2015 году. 7 ноября 2018 года проект GraphQL был перенесен из Facebook в GraphQL Foundation, размещённый некоммерческим Linux Foundation. С 2012 года рост GraphQL следовал срокам принятия, как это было установлено Ли Байроном, создателем GraphQL, с удивительной точностью. Целью Байрона было сделать GraphQL вездесущим на всех веб-платформах. GraphQL обеспечивает мощный, эффективный и гибкий подход к разработке API и был сравнен и сопоставлен с REST и другими архитектурами веб-сервисов, например SOAP. Это позволяет клиентам определять структуру требуемых данных, и с сервера возвращается точно такая же структура данных, что предотвращает возврат чрезмерно больших объемов данных, но это влияет на эффективность веб-кэширования результатов запроса. Гибкость и богатство языка запросов имеет и минусы, это сложность языка, которая может не иметь смысла для простых API. Он состоит из системы типов, языка запросов и семантики выполнения, самоанализа и статической проверки типа. GraphQL поддерживает такие операции как чтение, запись (изменение) и подписку на изменения данных (обновления в реальном времени). Основные клиенты GraphQL включают в себя Apollo Client и Relay.

Серверы GraphQL доступны для нескольких языков, включая Haskell, Clojure, JavaScript, Perl, Python, Go, Ruby, Java, C#, Scala, Elixir, Erlang, PHP, и R [25].

Если взаимодействия веб-приложения с API программно-аппаратной частью сервиса будет осуществляться за счет GraphQL, то потребуется использовать GraphQL клиент, например Apollo Client. Также при использовании GraphQL использование Redux не обязательно.

Допустим, что для разработки мобильного приложения будет использоваться React Native. Так как это библиотека, в ней не реализованы такие элементарные вещи как навигация по приложению и работа с состоянием данных.

Для работы с навигацией в React Native отлично подойдет React Navigation, а для работы с состоянием данных отлично подойдет Redux, React Redux, Redux Thunk и Redux Saga.

React Navigation – это JavaScript-библиотека для маршрутизации мобильных приложений, написанных с помощью React Native [26].

Так как в мобильных приложениях нет CSS, то необходимо использовать Yoga.

Yoga – это кроссплатформенный движок для построения макетов, который реализует CSS Flexbox [27].

Для разработки десктоп-приложения будет использоваться Electron совместно с React, Redux, React Redux, Redux Thunk и Redux Saga, React Router и Apollo Client, если взаимодействие будет за счет GraphQL, то использование Redux не обязательно.

Пример мультиплатформенной разработке программного обеспечения усилиями веб-разработчиков представлен на рисунке 1.

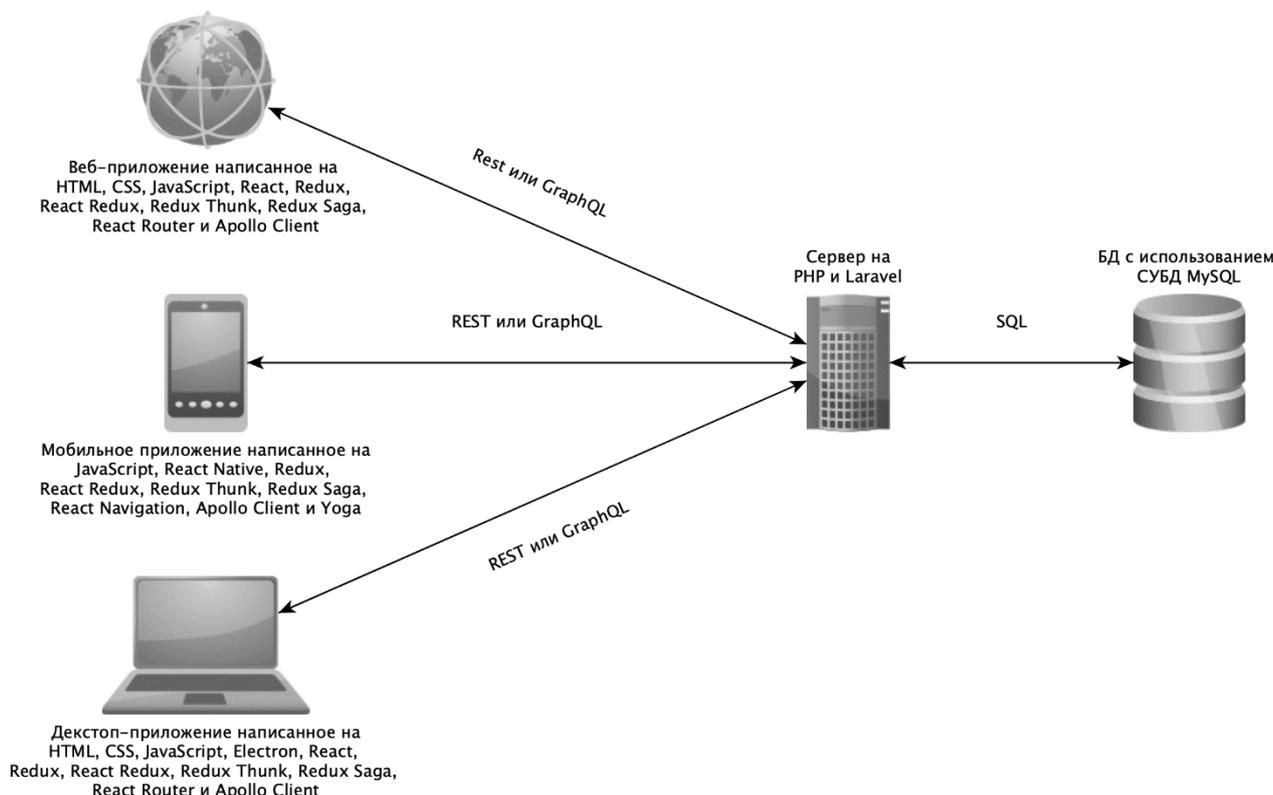


Рисунок 1 - Мультиплатформенная разработка программного обеспечения усилиями веб-разработчиков

В качестве инструментов для разработки может использоваться операционная система MacOS, набор необходимых программ MAMP, PhpMyAdmin, а также редакторы кода Visual Studio Code и PhpStorm.

Таким образом, можно сделать вывод что разработка мобильной и десктоп- версий приложения усилиями веб-разработчиков возможна при изначальном выборе следующего стека разработки: HTML, CSS, Yoga, JavaScript, React, React Native, Electron, Redux, React Redux, Redux Thunk, Redux Saga, React Router, React Navigation и Apollo Client.

## Список литературы

1. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт; [пер. с англ. М. А. Райтмана]. – М.: Эксмо, 2017. – 480 с.: ил. + 1 CD. – (Мировой компьютерный бестселлер);
2. Макфарланд Д. Новая большая книга CSS. – СПб.: Питер, 2016. – 720 с.: ил. – (Серия «Бестселлеры O'Reilly»);
3. Веру Л. Секреты CSS. Идеальные решения ежедневных задач. — СПб.: Питер, 2016. — 336 с.: ил. — (Серия «Бестселлеры O'Reilly»);
4. Грант Кит. CSS для профи. — СПб.: Питер, 2019. — 496 с.: ил. — (Серия «Библиотека про-граммиста»);
5. Мейер, Эрик А. CSS. Карманный справочник, 4-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2017. – 288 с.: ил. – Парал. тит. англ.;
6. Флэнага, Дэвид. JavaScript. Подробное руководство. – Пер. с англ. – СПб: Символ-Плюс, 2016. – 992 с., ил.;
7. Дакетт Д. JavaScript и jQuery. Интерактивная веб-разработка / Джон Дакетт; [пер. с англ. М. А. Райтмана]. – Москва: Издательство «Э», 2017. – 640 с.: ил. – (Мировой компьютерный бестселлер);
8. Флэнаган, Дэвид. JavaScript: карманный справочник, 3-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2016. - 320 с.: ил – Парал. тит. англ.;
9. Котеров, Д. В. PHP 7 / Д. В. Котеров, И. В. Симдянов. – СПб.: БХВ-Петербург, 2016. – 1088 с.: ил. – (В подлиннике);
10. Официальный сайт React Native [Электронный ресурс] – Режим доступа: URL <https://facebook.github.io/react-native/>;
11. Anthony Accomazzo, Ari Lerner, Nate Murray, Clay Allsopp, David Guttman, and Tyler McGinnis. Fullstack React. 2019, pp. 834 (на Английском);
12. Written by Devin Abbott, Houssein Djirdeh, Anthony Accomazzo, and Sophia Shoemaker. Fullstack React Native. 2018, pp. 681 (на Английском);
13. Nader Dabit. React Native in Action. 2019, pp. 321 (на Английском);
14. Официальный сайт Electron [Электронный ресурс] – Режим доступа: URL <https://electronjs.org/>;
15. Официальный сайт Laravel [Электронный ресурс] – Режим доступа: URL <https://laravel.com/>;
16. Официальный сайт React [Электронный ресурс] – Режим доступа: URL <https://ru.reactjs.org/>;

17. Томас Марк Тиленс. React в действии. - СПб.: Питер, 2019. - 368 с.: ил. - (Серия "Для профессионалов");
18. Мардан Азат. React быстро. Веб-приложения на React, JSX, Redux и GraphQL. - СПб.: Питер, 2019. - 560 с.: ил. - (Серия "Библиотека программиста");
19. Бэнкс Алекс, Порселло Ева. React и Redux: функциональная веб-разработка. — СПб.: Питер, 2018. — 336 с.: ил. — (Серия «Бестселлеры O'Reilly»);
20. Чиннатамби Кирупа. Изучаем React. [пер. с англ. М. А. Райтмана]. – М.: Эксмо, 2019. – 368 с.: ил. – (Мировой компьютерный бестселлер);
21. Официальный сайт Redux [Электронный ресурс] – Режим доступа: URL <https://redux.js.org>;
22. Официальный сайт React Redux [Электронный ресурс] – Режим доступа: URL <https://react-redux.js.org>;
23. Официальный сайт Redux Saga [Электронный ресурс] – Режим доступа: URL <https://redux-saga.js.org/>;
24. Официальный сайт React Router [Электронный ресурс] – Режим доступа: URL <https://reacttraining.com/react-router/>;
25. Официальный сайт GraphQL [Электронный ресурс] – Режим доступа: URL <https://graphql.org/learn/>;
26. Официальный сайт React Navigation [Электронный ресурс] – Режим доступа: URL <https://reactnavigation.org/>;
27. Официальный сайт Yoga [Электронный ресурс] – Режим доступа: URL <https://yogalayout.com>.

## References

1. Duckett D. HTML and CSS. Website development and design / John Duckett; [trans. from English M. A. Wrightman]. - М .: Eksmo, 2017 .-- 480 p .: ill. + 1 CD. - (World Computer Bestseller) (in Russian);
2. McFarland D. The New Big Book of CSS. - SPb .: Peter, 2016 .-- 720 p .: ill. - (O'Reilly Best Sellers Series) (in Russian);
3. Faith L. Secrets of CSS. Ideal solutions for everyday tasks. - SPb .: Peter, 2016 .-- 336 p .: ill. - (O'Reilly Best Sellers Series) (in Russian);
4. Grant Kit. CSS for the pros. - SPb .: Peter, 2019 .-- 496 p.: Ill. - (Series "Library about the programmer") (in Russian);
5. Meyer, Eric A. CSS. Pocket Guide, 4th ed .: Per. from English - М .: LLC "I.D. Williams ", 2017. - 288 p.: Ill. - Paral. tit. English (in Russian);
6. Flanagan, David. Javascript Detailed guide. - Per. from English - St. Petersburg: Symbol-Plus, 2016 .-- 992 p., Ill . (in Russian);
7. Duckett D. JavaScript and jQuery. Interactive Web Development / John Da-kett; [trans. from English M. A. Wrightman]. - Moscow: Publishing House "E", 2017. - 640 p.: Ill. - (World Computer Bestseller) (in Russian);
8. Flanagan, David. JavaScript: A Pocket Guide, 3rd ed .: Per. from English - М .: LLC "I.D. Williams ", 2016. - 320 с.: Il - Paral. tit. English (in Russian);
9. Koterov, D.V. PHP 7 / D.V. Koterov, I.V. Simdyanov. - SPB .: BHV-Petersburg, 2016 .-- 1088 p.: Ill. - (In the original) (in Russian);

10. Official React Native website [Electronic resource] - Access mode: URL <https://facebook.github.io/react-native/>;
  11. Anthony Accomazzo, Ari Lerner, Nate Murray, Clay Allsopp, David Guttman, and Tyler McGinnis. Fullstack React. 2019, pp. 834;
  12. Written by Devin Abbott, Houssein Djirdeh, Anthony Accomazzo, and Sophia Shoemaker. Fullstack React Native. 2018, pp. 681;
  13. Nader Dabit. React Native in Action. 2019, pp. 321;
  14. Official site of Electron [Electronic resource] - Access mode: URL <https://electronjs.org/>;
  15. Official site Laravel [Electronic resource] - Access mode: URL <https://laravel.com/>;
  16. Official React website [Electronic resource] - Access mode: URL <https://ru.reactjs.org/>;
  17. Thomas Mark Tilens. React in action. - SPb .: Peter, 2019 .-- 368 с .: ill. - (Series "For Professionals");
  18. Mardan Azat. React fast. Web applications in React, JSX, Redux, and GraphQL. - SPb .: Peter, 2019 .-- 560 с .: ill. - (Series "Programmer's Library") (in Russian);
  19. Banks Alex, Porsello Eve. React and Redux: functional web development. - SPb .: Peter, 2018 .-- 336 p .: ill. - (O'Reilly Best Sellers Series) (in Russian);
  20. Chinnatambi Kirupa. Learn React. [trans. from English M. A. Wrightman]. - M .: Eksmo, 2019 .-- 368 p.: Ill. - (World Computer Bestseller) (in Russian);
  21. Official Redux website [Electronic resource] - Access mode: URL <https://redux.js.org/>;
  22. Official site of React Redux [Electronic resource] - Access mode: URL <https://react-redux.js.org/>;
  23. Official site of Redux Saga [Electronic resource] - Access mode: URL <https://redux-saga.js.org/>;
  24. Official site of React Router [Electronic resource] - Access mode: URL <https://reacttraining.com/react-router/>;
  25. GraphQL official website [Electronic resource] - Access mode: URL <https://graphql.org/learn/>;
  26. Official site of React Navigation [Electronic resource] - Access mode: URL <https://reactnavigation.org/>;
  27. Official site Yoga [Electronic resource] - Access mode: URL <https://yogalayout.com>.
-