

помощью Angular, TypeScript или JavaScript; Electron – фреймворк для создания кроссплатформенных десктопных приложений на JavaScript, HTML, и CSS.

На наш взгляд перечисленный набор технологий обеспечивает выполнение всех требований, предоставляемых к системе как изначально, так и после анализа первой версии.

Код из Web приложения на Angular оборачивается с помощью Electron и на выходе мы имеем Desktop приложение. Используя NativeScript мы привязываем всю логику описанную с помощью Angular к кроссплатформенным компонентам интерфейса для Android и iOS. Sails.js позволяет, используя сокет поддерживать данные во всех приложениях в актуальном состоянии и моментально синхронизировать их.

Таким образом разрабатываемая в настоящее время система позволит исключить перечисленные выше недостатки первой версии системы. Используя перечисленные технологии, система обеспечивает выполнение предъявляемых к ней требований, обладает высоким функционалом и является востребованной.

УДК 004.42

Ивашко С.П., Капуста Е.В.

РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА NATIVESCRIPT И ANGULAR

БНТУ, Минск

Научный руководитель Дробыш А.А.

В настоящее время почти у каждого человека в кармане лежит смартфон с доступом к интернету. И в каждом из этих миллионов устройств находится множество предустановленных производителем или скаченных и установленных самим пользователем из специализированного магазина приложений.

Двумя самыми популярными мобильными платформами являются Android и iOS. Каждая организация или сервис стараются помимо сайта и приложения под персональный компьютер, предоставить еще и приложения под эти платформы. Для разработки приложений под Android используется язык

программирования Java, а под iOS язык Swift. Использование различных языков для разработки под разные платформы подразумевает написание двух совершенно не похожих друг на друга приложений, двумя командами разработчиков, использующих различные средства для этого. Это повышает затраты на разработку и разделяет сам процесс на две почти независимые части. Именно поэтому есть технологии, позволяющие разрабатывать приложения сразу под несколько мобильных платформ, естественно накладывающие ряд ограничений.

Мы рассмотрим набор технологий для разработки мобильных приложений под Android и iOS, основанный на Web-технологиях, а именно связку Nativescript и Angular. Изначально Angular – JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений с использованием языка программирования TypeScript. Использование Angular совместно с открытым фреймворком Nativescript позволяет разрабатывать мобильные приложения с использованием языка TypeScript и принципов как мобильной, так и веб разработки. Использование подобного решения дает ряд преимуществ: большинство кода используется сразу двумя платформами; почти идентичный интерфейс для обеих платформ; HTML как язык разметки интерфейса, вместо стандартных HTML тегов используются теги одноименные стандартным элементам интерфейса Android и iOS; CSS как язык описания стилей; возможность использования CSS препроцессоров; TypeScript как основной язык разработки; использование преимуществ и принципов разработки Angular; быстрый старт для разработчиков знакомых с мобильной или веб разработкой; удобные инструменты разработчика; автоматическая настройка рабочей среды; множество плагинов для расширения возможностей.

Как и у каждой технологии у Nativescript есть свои недостатки: автоматическая настройка среды работает не всегда корректно, иногда приходится устанавливать некоторые компоненты вручную; часть кода остается платформо-зависимой, её приходится либо оборачивать в специальные теги, либо выносить в отдельные файлы; из-за особенностей платформ приходится дорабатывать отдельные части интерфейса под них; не все плагины доступные для NativeScript обновляются вовремя и корректно работают в новых версиях

NativeScript под обеими платформами; для отладки iOS приложений необходимо устройство с macOS; не самые удобные средства отладки программ. Рассматривая преимущества и недостатки данного набора технологий можно сделать вывод: использование NativeScript имеет смысл при разработке приложений малой и средней сложности, не использующих множество аппаратных возможностей. Если у вас небольшая компания с малым штатом разработчиков и вам нужен быстрый старт с мобильными приложениями небольшой сложности для обеих популярных платформ, то такой подход может стать отличным выбором.

УДК 371.68/69

Игнатовец В.М.

ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ ПРИ ПРОЕКТИРОВАНИИ И СОЗДАНИИ УЧЕБНЫХ ЭЛЕКТРОННЫХ ПОСОБИЙ

БНТУ, Минск

Научный руководитель Кравченя Э.М.

В современной системе образования возрастает роль информационных технологий, которые охватывают дополнительные возможности как для повышения качества и эффективности процесса обучения, так и для расширения сфер его применения. Быстрыми темпами развивается новая прогрессивная форма организации учебного процесса на основе принципа самостоятельного обучения ученика с помощью различных информационных ресурсов – дистанционное образование.

Стремительный процесс компьютеризации образования на основе современных компьютерных систем, поступающих в учебные заведения страны, открывает в образовании путь электронным учебникам (ЭУ). Этот термин в настоящее время наиболее устойчив, и к этому типу разработок относятся все в большей или меньшей степени целостные компьютерные курсы учебного назначения.

Учебник, в классическом понимании, это книга для учащихся или студентов, в которой систематически излагается материал в