

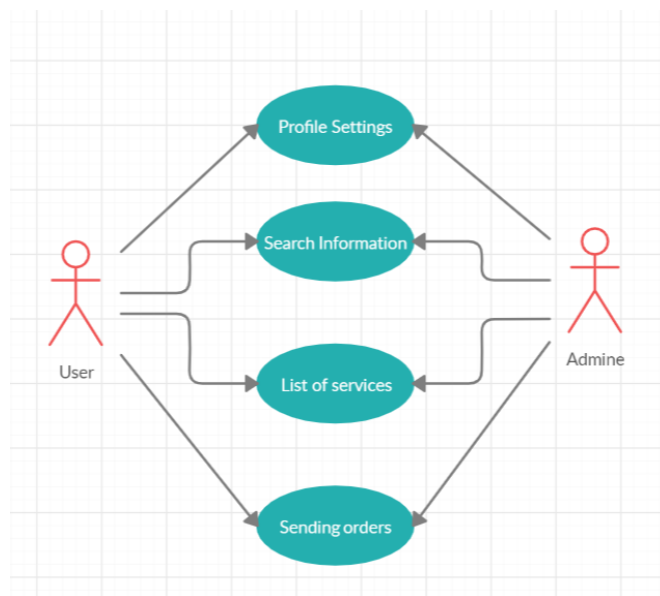


Рисунок 2 – Другая схема использования

Архитектура проекта состоит из нескольких частей: административной, пользовательской и серверной.

Административная часть отвечает за управление проектом, а именно - после регистрации заявки администратор должен проверить информацию о заявке и принять решение о разрешении или отклонении пользователя и заявки.

Пользовательская часть отвечает за выполнение заказов, проверку списка компаний, настройку их параметров, оставление отзывов, проверку информации о заказах.

Сервер содержит всю бизнес-логику сайта, а также логику для авторизации, обработки входных запросов и операций с базой данных.

Н. В. Федин, Д. А. Костюченко (ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **Е. А. Дей**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ «АБИТУРИЕНТ ГГУ»

Центром тестирования УО ГГУ имени Ф. Скорины была поставлена задача разработки мобильного приложения для удобного доступа абитуриентов к справочным материалам центра и личному кабинету абитуриента. В ходе создания полноценного мобильного приложения возникает ряд проблем, связанных с разработкой и публикацией приложения в короткий срок, а также предоставлением полного функционала, аналогич-

ного сайту университета. На помощь приходит временное решение, называемое программной оболочкой («оберткой»).

Такой подход приобрёл большую популярность с возникновением потребности в наличии собственных мобильных приложений у большинства малых и средних предприятий. Следует отметить, что стоимость полной разработки даже на одной из двух лидирующих платформ Android и iOS становится нереальной для большинства предприятий. Описанная проблема имеет отношение к полноценным мобильным приложениям, созданным полностью с применением инструментов в родной среде. Значительно более дешёвым, но далеко не худшим вариантом выступает приложение-оболочка. Это решение было неоднократно опробовано компаниями по разработке мобильных приложений и показало себя довольно жизнеспособным.

Специализированное мобильное приложение, нацеленное на отображение одного адаптивного сайта, – так вкратце можно описать общее представление относящиеся к этому подходу. Внутри такого приложения находится веб-сайт, переделанный под мобильные устройства. С помощью родных компонентов операционной системы он может делать запросы к пуш-уведомлениям, геолокации и камере, и как приложение его можно просто распространять через магазины приложений.

Что необходимо для начала работы – качественный адаптивный сайт, который хорошо отображается на мобильных устройствах. Однако бывают и такие случаи, когда сайт не является полностью адаптивным и его отображение на мобильных устройствах получается достаточно неинформативным, тогда в ход вступает ручная обработка со стороны мобильного приложения. Необходимо заранее:

- Предусмотреть хранение данных в памяти устройства.
- Продумать варианты отображение при переключении ориентации.
- Решить проблему поведения приложения в случае недоступности или потери интернета.
- Проверить адекватность работы функционала сайта и собственный JavaScript-код.

Для реализации данного решения были использованы интегрированные среды разработки от создателей платформ: Android Studio (платформа Android) и язык Java, а также Xcode (платформа iOS) и язык Swift.

Software Development Kit (SDK) этих платформ имеют компоненты WebView, позволяющие создавать специальное окно для отображения веб-страниц или реализовать полноценную копию браузера.

Данный компонент взаимодействует с движком WebKit в наличии которого имеются различные свойства и методы. Последние версии WebView для Android используют движок от Chromium, но для обыкновенных задач большой разницы в этом нет.

В разметке интерфейса приложения основное место выделено под элемент WebView и затем дополнено нативными компонентами, такими как всевозможные виды навигационных меню, окна настроек, заголовки и отдельные компоненты интерфейса, заменяющие элементы веб-страницы, которые, в свою очередь, удаляются оттуда при помощи JavaScript или CSS инъекций. Под инъекцией подразумевается внедрение собственного кода в структуру сайта с целью корректировки, улучшения и оптимизации веб-страницы. В разделы сайта, в зависимости от его подразделов, были внесены индивидуальные технические и косметические изменения, улучшающие пользовательский опыт работы с приложением. Результат работы представлен на рисунках 1 и 2.

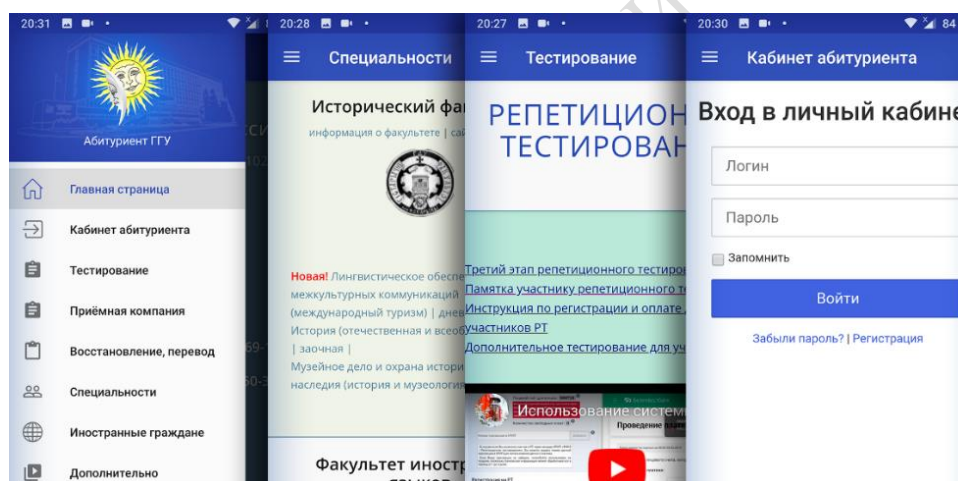


Рисунок 1 – Окна созданного приложения-оболочки

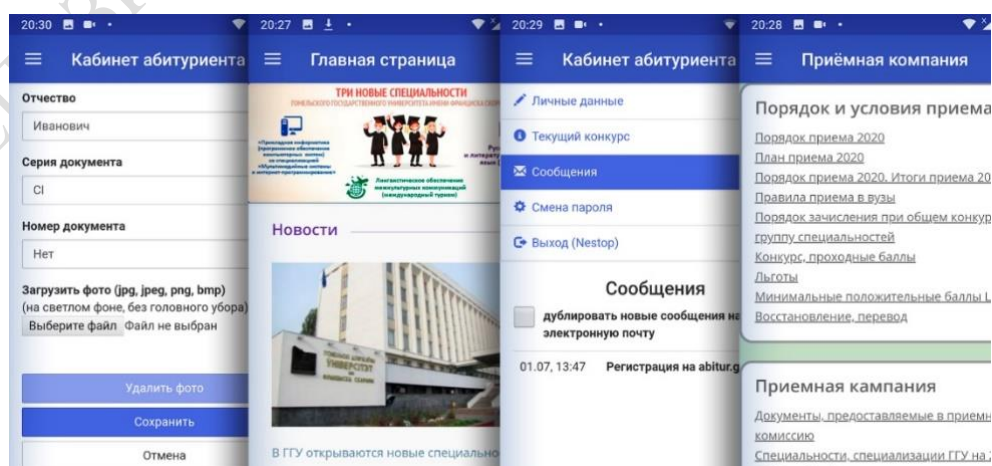


Рисунок 2 – Окна созданного приложения-оболочки

За время разработки приложения-оболочки был проанализирован полученный опыт и выделены следующие преимущества:

- Быстрота разработки, достаточно буквально трёх-пяти недель при должном опыте в используемых технологиях.
- Невысокая стоимость. Во много раз меньше в отличие от приложения, написанного с использованием полностью родных компонентов.
- Контроль содержания приложения через веб-страницу, без необходимости доработки.
- Отсутствует потребность реализации программного интерфейса на сервере.
- Отсутствует необходимость в значительной переделке дизайна.
- Тестирование сводится к минимуму, так как приложение работает со встроенным браузером, возможности которого везде практически однотипны.

Как и в любом решении, были так же выявлены недостатки приложений-оболочек:

- Полноценная адаптация дизайна для достижения высокого качества требует немалых усилий.
- Нестандартный интерфейс, работающий по другим принципам.
- Интерфейс является менее оптимизированным, чем у приложений, использующих родные компоненты среды.
- Присутствует необходимость писать нативный код платформы для полной поддержки сайта, что увеличивает время на разработку.
- Устройства с более старыми версиями операционных систем имеют риск некорректного отображения приложения.

Литература

1. Медникс, З. Программирование под Android. 2-е изд. / З. Медникс, Л. Дорнин, Б. Мик, М. Накамура – СПб.: Питер, 2013. – 560 с.
2. Apple Developer – Документация [Электронный ресурс] – Режим доступа: <https://developer.apple.com>. – Дата доступа: 15.04.2020.
3. Android Developers – Документация [Электронный ресурс] – Режим доступа: <https://developer.android.com/docs>. – Дата доступа: 19.04.2020.
4. Swift.org – Документация [Электронный ресурс] – Режим доступа: <https://swift.org>. – Дата доступа: 20.04.2020.
5. Дейтел, П. Android для разработчиков. 3-е изд. / П. Дейтел, Х. Дейтел, А. Уолд – СПб.: Питер, 2016. – 512 с.