

Санкт-Петербургский государственный университет  
Математико–механический факультет  
Кафедра системного программирования

Андреева Мария Александровна

Курсовая работа

Разработка настраиваемых мобильных  
сервисов на платформе Ubiq Mobile на  
примере мобильного интерфейса  
Интернет-магазина

Направление 02.04.03

Математическое обеспечение и администрирование информационных  
систем

Заведующий кафедрой,  
доктор физ.-мат. наук,  
профессор

Терехов А. Н.

Научный руководитель,  
доктор физ.-мат. наук,  
профессор

Терехов А. Н.

Санкт-Петербург  
2016 г.

# Содержание

<b>1. Введение</b>	<b>3</b>
<b>2. Этапы работы и результаты</b>	<b>5</b>
2.1. Работа над приложением . . . . .	5
2.2. Работа над Веб-интерфейсом . . . . .	7
2.3. Результаты . . . . .	8
<b>3. Планы на будущее</b>	<b>10</b>
<b>4. Заключение</b>	<b>10</b>

# 1. Введение

В последнее время наблюдается тенденция к поиску различных путей конвергенции Web и мобильных приложений для того, чтобы одновременно использовать их лучшие стороны: нативность мобильных приложений и гибкость Web (доступ по ссылке, отсутствие необходимости загружать приложение на девайс, мгновенные обновления).

На данный момент существуют несколько направлений решения этой задачи, например Instant Apps и стримминг приложений от Google, карточки приложений от Intercom и боты в мессенджерах. Instant Apps, анонсированные Google I/O 2016 в мае 2016-го года, не нуждаются в установке на девайс. Приложение сразу, без предварительной установки, открывается на устройстве в некотором усеченном виде, а после закрытия автоматически удаляется из системы (но если пользователь желает оставить программу на своем гаджете, то в окне приложения предусмотрена специальная иконка, загружающая полную версию приложения из Play Market). Если пользователь хочет воспользоваться приложением только один раз — это отличный вариант (например, оплатить парковку в чужом городе). А в конце 2015-го года Google представил функцию стримминга приложений: приложение, находящееся в облаке, не скачивается на девайс, а открывается в браузере.

В рамках данной работы предлагается решение более скромной задачи (в рамках этой же тенденции): быстрой и дешевой мобилизации серии схожих информационных систем без необходимости писать и поддерживать множество приложений (например, для документооборота, интернет-магазинов, клиник и т. д.). Таким образом, хочется иметь возможность создавать шаблоны сервисов, настраиваемые без программирования, а только с помощью конфигурирования. Платформа Ubiq Mobile позволяет реализовать такие шаблоны проще, чем с помощью других технологий, в силу архитектурных особенностей:

- Клиент-серверной архитектуры

- Концентрации логики на сервере
- Тонких клиентов
- Возможности загрузки любого UI с сервера на клиент
- Кэширования UI на клиенте и возможности работы offline.

Благодаря вышеперечисленному возможна мгновенная «трансформация» приложения, как веб-страницы, при внесении каких-либо изменений на стороне сервера.

Конечной целью является создание автоматизированного настраиваемого сервиса, в состав которого входят приложение-шаблон (или несколько) и веб-интерфейс для осуществления настройки шаблона под требования информационной системы клиента (с помощью файла конфигурации или UbiqML). Сервис должен позволять модифицировать способ интеграции с внешними системами, пользовательский интерфейс, состав функций из общего набора и генерировать персонифицированное приложение в автоматическом режиме.

На Рис 1. схематически изображена архитектура настраиваемого сервиса.

Целью данной курсовой работы является отработка программных и архитектурных решений для использования при разработке настраиваемых сервисов на основе Ubiq Mobile.

В качестве примера в предыдущей курсовой работе рассматривалось настраиваемое приложение для Интернет-магазина, так как существует множество базовых магазинов, к которым можно подключаться различными способами и их функциональность хорошо поддается стандартизации. В результате было создано приложение для Интернет-магазина, сопряженное с сайтом, содержащее описание магазина, адреса и каталог, разделенный на категории. Приложение получало данные с сайта через YML-файл, настраивалось с помощью файла конфигурации и обновляло каталог по таймеру. В этой работе планируется продолжать отработку программных

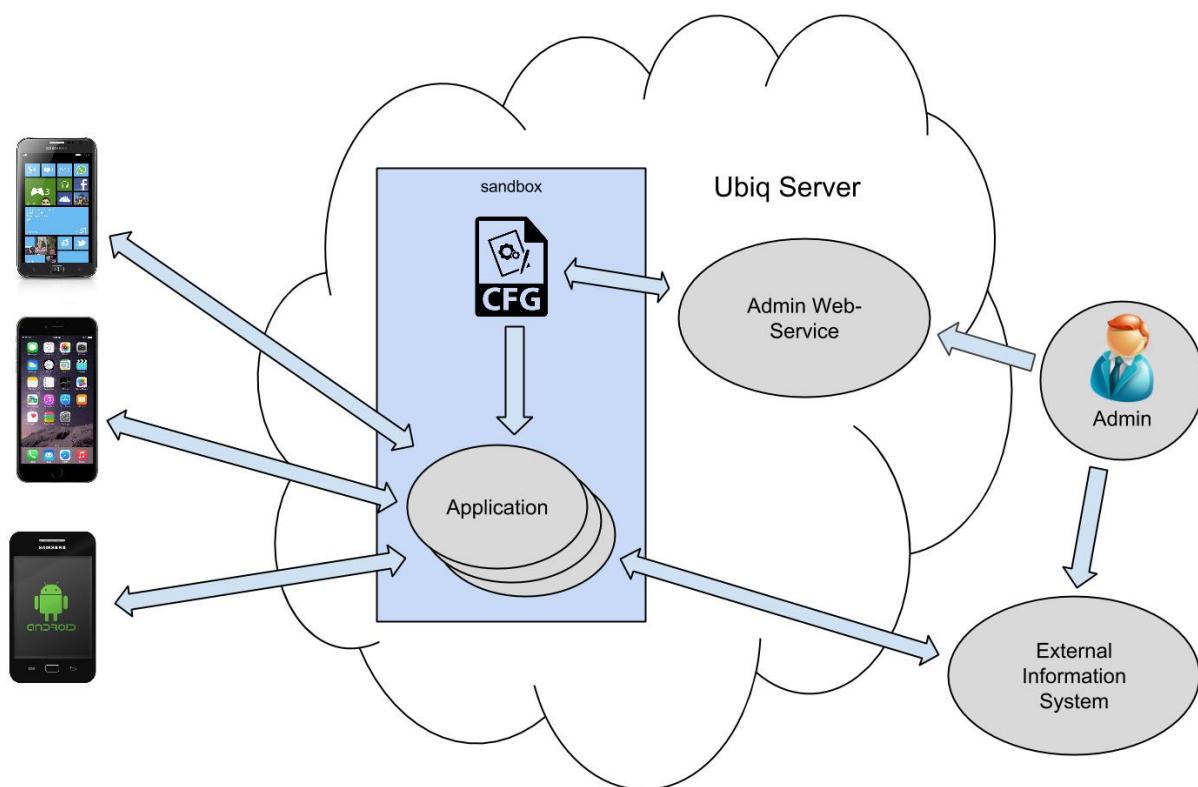


Рис. 1. Архитектура настраиваемого сервиса

и архитектурных решений настраиваемых сервисов на примере Интернет-магазина.

## 2. Этапы работы и результаты

### 2.1. Работа над приложением

В первую очередь, в рамках этой работы, существующее приложение было переписано с использованием оригинального механизма, называемого для удобства контроллерами. Контроллеры — это механизм, позволяющий инкапсулировать отдельные фрагменты большого автомата (большая часть приложений использует автоматную логику), со своими внутренними состояниями, экранами, которые должны отображаться, и т.д., и свободно компоновать такие фрагменты с использованием «общей шины». Контроллеры могут иметь свои внутренние состояния, ожидать события. Существуют разные варианты передачи управления между контроллерами,

возможны переходы между ними. В библиотеке стандартных контроллеров платформы Ubiq Mobile на данный момент существуют такие контроллеры: контроллер авторизации (содержит экраны для авторизации, регистрации, ответы об ошибках и т.п.), контроллер списков (показывает список, имеет встроенный переход на развернутый просмотр элемента списка по клику, умеет искать по списку), контроллер для показа статической информации).

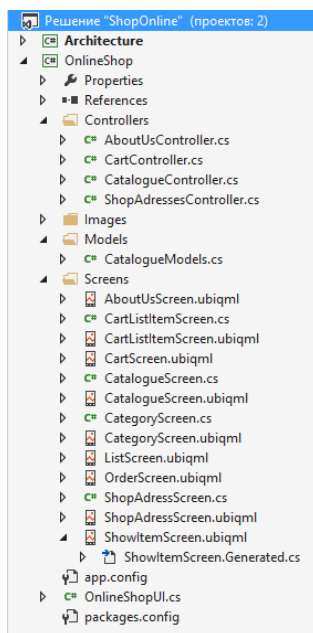


Рис. 2. Решение «ShopOnline»

Приложение для магазина на данный момент включает в себя четыре контроллера (для каждого пункта меню): AboutUsController, CatalogueController, CartController, ShopAdressesController.

Контроллеры AboutUsController и ShopAdressesController содержат по одному состоянию — показа статической страницы. CatalogueController имеет четыре состояния: показ списка категорий, товаров в конкретной категории, найденных с помощью поиска товаров, и страницы с более подробным описанием продуктов.

При создании CartController был выполнен еще один пункт обозначенного в предыдущей работе плана — создание функционирующей корзины. CartController переключается между двумя состояниями: показом элементов корзины (с возможностью увеличивать или уменьшать количество за-

казываемых товаров, удалять их из корзины) и страницы с полями для оформления заказа и кнопкой отправки заказа.

Как и планировалась, был проработан другой вариант получения каталога. В связи с популярностью WordPress было решено научиться получать данные магазинов Woocommerce (бесплатный плагин для WordPress). Для этого был создан магазин-пример с глубокой иерархией каталога (множеством подкатегорий). Каталог с магазинов Woocommerce можно получать как в формате YML с помощью дополнительного плагина Market Exporter (этот вариант уже был реализован в рамках предыдущей курсовой работы), так и с помощью REST API от Woocommerce.

С помощью API можно получить больше информации о магазине и товарах, чем из YML-файла, например, фотографии, соответствующие категориям. А также можно отправить заказ прямо на сайт, как будто заказ был сделан не с мобильного устройства, а с формы на сайте магазина. В приложении вариант получения каталога через REST API был реализован с отправкой заказа на сайт.

## 2.2. Работа над Веб-интерфейсом

Веб-интерфейс нужен для удобства и легкости редактирования файла конфигурации клиентом. Файл конфигурации приложения содержит следующие параметры:

- Имя магазина
- Количество статических страниц, их заголовки и содержимое
- Детали интерфейса: цвета, иконки и логотипы
- Вариант получения каталога (YML-файл или Woocommerce Rest API)
- Ссылка на YML-файл
- Ссылка на магазин на Woocommerce

- Ключи для API
- Параметр, согласно которому включается или выключается показ картинок для категорий

```

1 <?xml version="1.0" encoding="utf-8" ?>
  <configuration>
3   <appSettings>
      <add key="shopname" value="Деталькин"/>
5      <add key="mainpage" value="О нас"/>
      <add key="info" value="Все для вашего автомобиля – по лучшим ценам в Санкт
      –Петербурге и Ленинградской области! Бесплатная доставка и самовывоз,
      гарантия. Широкий ассортимент. Ждем Ваших заказов!"/>
7      <add key="top_panel_color" value="FF3F94FC"/>
      <add key="slide_panel_color" value="FFFDFFFF"/>
9      <add key="CatalogueType" value="2"/>
      <add key="urlToYml" value="http://glazastik.nethouse.ru/pricelist.xml"/>
11     <add key="WooLink" value="http://ubiqmobile.com/OnlineShop"/>
      <add key="ConsumerKey" value="ck_bc718da284e743f950ad4ff117e08885b0600000"
      />
13     <add key="ConsumerSecret" value="
      cs_a91eac42085dc2b711027af9505def5081400000"/>
      <add key="CategoryPicture" value="0"/>
15  </appSettings>
</configuration>

```

Для приложения-магазина на виртуальном сервере («песочнице») в облаке был создан веб-интерфейс для настройки параметров файла конфигурации под конкретных пользователей. Клиент может сконструировать свое приложение, как будто оно целенаправленно писалось под него. А в дальнейшем имеет возможность вносить в него изменения с помощью того же интерфейса без необходимости повторной публикации приложения в магазине.

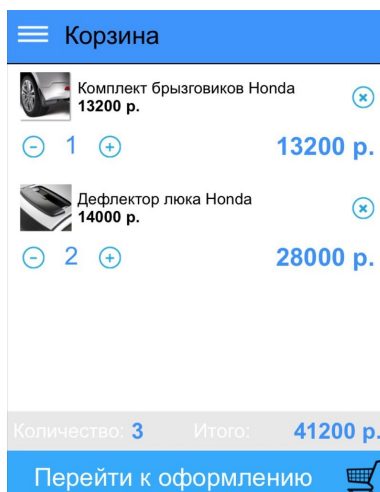
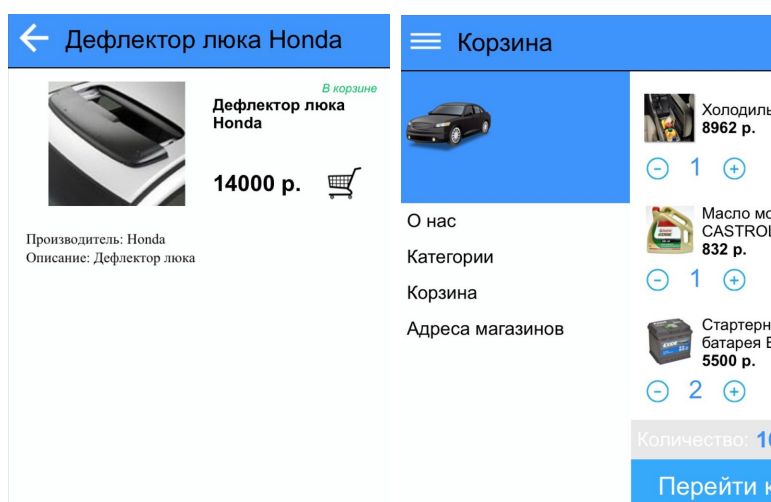
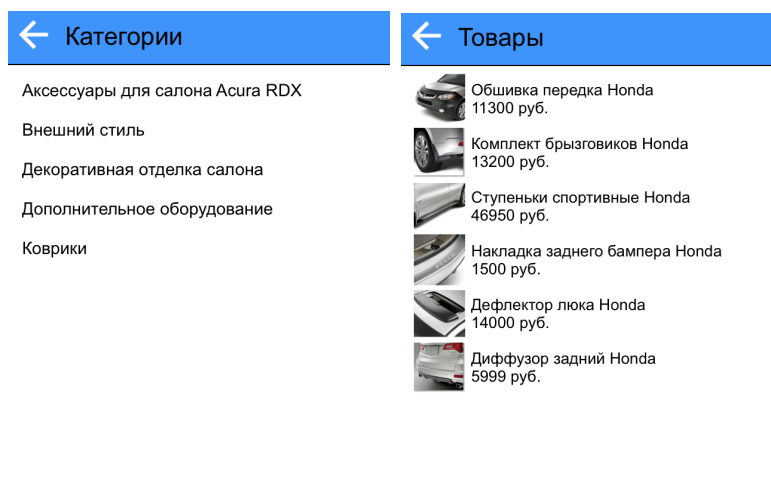
## 2.3. Результаты

Таким образом, на данном этапе имеется настраиваемый сервис для интернет-магазинов, состоящий из приложения и веб-интерфейса, имею-



щий полноценную корзину (для Woocommerce осуществляется отправка заказа на сайт магазина), каталог, несколько статических страниц с информацией и поиск по каталогу.

## Скриншоты приложения



### 3. Планы на будущее

В перспективе планируется:

- Автоматизация инфраструктурных частей сервиса (выделение песочницы для клиента при регистрации, предоставление возможности выбрать шаблон и отредактировать его под себя, сборка персонифицированного приложения).
- Отработка решений и отдельных компонент, которые можно будет использовать в других подобных сервисах на базе Ubiq Mobile.
- Расширение настраиваемости и адаптируемости решения (новые магазины и настраиваемые параметры)
- Расширение функциональности (добавление адресов магазинов на карты, возможность отслеживать статус заказа, push-уведомления)

### 4. Заключение

Таким образом, в курсовой работе был рассмотрен пример настраиваемого сервиса — приложение для Интернет-магазина и веб-интерфейс для его настройки. Запланированные на прошлом этапе задачи выполнены. В дальнейшем планируется автоматизировать процесс персонализации приложения, увеличить количество настраиваемых параметров, расширить функционал приложения, чтобы пользователь, с помощью сервиса, мог за короткое время собрать своё уникальное приложение и обновлять его без повторной публикации в магазине приложений.

## Список литературы

- [1] Валерий Коржов. Многоуровневые системы клиент-сервер // Сети/network world, 1997, №6.
- [2] V. Onossovski, A.Terekhov. Ubiq Mobile – a New Universal Platform for Mobile Online Services // Proceedings of 6th seminar of FRUCT Program, 2009.
- [3] V. Onossovski, A.Terekhov. Modern Interactive Internet Services // Proceedings of 7th Conference of Open Innovations Framework Program FRUCT, 2010.
- [4] А. Н. Терехов, В. В. Оносовский. Платформа для разработки мобильных приложений Ubiq Mobile // Вестник НГУ. Серия: Информационные технологии, 2011, том 9, выпуск 4. С. 60-70.
- [5] А. Н. Терехов, В.В. Оносовский. Технология разработки мобильных онлайн сервисов. // Конференция CEE-SECR 2011. С. 1-2
- [6] <https://yandex.ru/support/webmaster/goods-prices/technical-requirements.xml>