

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет»

Кафедра системного программирования

Васильев Алексей Дмитриевич

Персонафикация мобильных приложений для конечных
пользователей на основе шаблонов

Курсовая работа

Допущена к защите.

Зав. кафедрой:

д. ф.-м. н., профессор Терехов А. Н.

Научный руководитель:

д. ф.-м. н., профессор Терехов А. Н.

Санкт-Петербург

2015

Оглавление

Введение	3
1. Особенности реализации сервиса	6
2. Шаблонизация	6
2.1.Создание файла конфигурации.....	6
2.2.Реализация механизма конфигурационных файлов	7
2.3.Снятие ограничений на количество хранилищ.....	7
2.4.Добавление функционала «живой» очереди.....	8
Заключение	9
Список литературы	10

Введение

Мобилизация программного обеспечения, наряду с технологиями обработки больших данных и облачными технологиями, является одним из основных трендов последних лет.

В последние годы наблюдается тенденция к отказу от использования персональных компьютеров в пользу мобильных решений. В бизнесе возможность принимать решения «в поле» особенно важна. Она позволяет экономить время, выигрывая у менее мобильных конкурентов. Это повышает уровень эффективности бизнеса.

Но все же мобилизация бизнес систем происходит довольно медленно, это обусловлено несколькими факторами. Главным из таких факторов является невозможность использовать массовые, широко тиражируемые решения для бизнеса, которые удовлетворяли бы частным потребностям конкретного пользователя, а индивидуальная разработка программных продуктов влечет довольно большие затраты времени и средств. Так же зачастую оказывается, что во время процесса создания программного обеспечения возникают различные проблемы, такие как трудности в создании начальной спецификации, правки и т.д. Все это приводит к удорожанию его стоимости и увеличению времени разработки.

Одной из альтернатив разработки мобильных приложений является создание мобильных версий сайтов. К сожалению нестабильность работы, ограничения из за отсутствия некоторых функций и сложный процесс разработки, требующий применения сразу нескольких инструментов, также являются крупной проблемой. Вышедший в 2014 году HTML 5, призванный исправить данные проблемы, оказался слишком тяжел и громоздок для реализации мобильных версий сайтов. Кроме того он по прежнему требовал использования целого набора инструментов разработки. Часто такое решение теряет нативность – то, что кажется простым и удобным для десктопной версии, может оказаться непригодным для использования на планшете, не говоря уже о экране смартфона.

В течении нескольких лет на Математико-механическом факультете СПбГУ разрабатывается технология «мобилизации» бизнес приложений, которая может решить описанные выше проблемы - UbiqMobile. Особенностью данной технологии является возможность хранить, выполнять, изменять бизнес логику, графику, данные на стороне сервера. Пользователь при этом получает тонкий клиент, получающий данные от сервера и отображающий их на экран мобильного устройства, при этом данные частично кэшируются в локальной памяти устройства. Это в чем-то делает клиент UbiqMobile похожим на браузер, но для конечного пользователя он выглядит как обычное нативное мобильное приложение. Такой подход позволяет приложению практически ничего не знать о специфике устройства, то есть сделать технологию кроссплатформенной. Обмен данными между серверными приложениями и мобильными клиентами осуществляется через оригинальный двоичный протокол, построенный над TCP/IP. [1]

Использование UbiqMobile позволяет реализовать оригинальный подход, основанный на идее настраиваемых сервисов. Решать эту проблему предлагается путем предоставления мобильного приложения в виде настраиваемого сервиса. Идея состоит в следующем: существует шаблон мобильного приложения, которое решает конкретную бизнес задачу. Мы можем произвести настройку такого приложения как со стороны клиента: графика, данные и т.д., так и со стороны интеграции с базовой информационной системой. Тогда заказчик, который хочет решить бизнес-задачу, может вместо разработки оригинального приложения использовать настраиваемый сервис. Использование технологии UbiqMobile также позволяет производить изменение интерфейса «на лету», при этом не требуется обновлять приложение на телефонах конечных пользователей. Для этого достаточно внести лишь правки на сервере.

Предметом данной курсовой работы является проработка отдельных аспектов выше описанного решения и создание прототипа приложения на

основе настраиваемых шаблонов. За основу для создания прототипа было взято существующее приложение на основе технологии UbiqMobile, которое реализует часть бизнес процесса, связанного с логистикой. Приложение было преобразовано, в прототип настраиваемого шаблона, с целью стать частью выше описанного сервиса, а его функциональность была существенно расширена. Создание такого приложения должно стать первым этапом в реализации прототипа сервиса предоставления мобильных приложений для конечного пользователя на основе настраиваемых шаблонов.

1. Особенности реализации сервиса

В качестве основы для шаблонизации было использовано существующее приложение. Оно предназначено для организации электронной очереди грузовиков, получающих продукцию со склада, и резервации тайм слотов на одном из складов.

Приложение включает в себя три части: пользовательское приложение, приложение «менеджер» и сервисную компоненту, в которой реализована основная бизнес логика. Все серверные компоненты работают на виртуальном сервере («песочнице») в облаке. Доступ к сервису может осуществляться не только через мобильные устройства, но и через web интерфейс, который обращается к той же самой сервисной компоненте.

Архитектура каждого приложения организована следующим образом: для каждого экрана приложения верстка представляется в виде UbiQML – специального графического формата на базе языка XML, поддерживаемого системой UbiqMobile. В приложении для каждого экрана создается объект класса Form, в котором реализуется логика этого экрана. Основной класс приложения представляет собой конечный автомат, управляющий переключениями между формами.

Сервисная компонента предоставляет API для обращения к ней со стороны других приложений, использующий механизм обмена сообщениями (messaging). Для взаимодействия в web интерфейсом компонента обеспечивает SOAP механизм. Данные приложения хранятся в специальной объектной базе данных, являющейся частью платформы UbiqMobile.

«Песочница» разделена на две части: Data – в которой хранятся файлы базы данных и App – содержащей файлы динамически подключаемых библиотек (dll).

2. Шаблонизация

2.1. Создание файла конфигурации

Использование механизма шаблонов предполагает легкую замену одних данных на другие. Для реализации такой замены был создан файл

конфигурации - config.xml. Он был расположен в папке Data. Для хранения в нем информации была разработана объектная, иерархическая структура файла.

Для начальной параметризации был выбран следующий список параметров:

1. Порт;
2. Адрес компании;
3. Телефоны;
4. Имена складов;
5. Адреса складов;
6. Краткие адреса складов;
7. Дополнительную информацию по адресу;

При первом запуске сервисная компонента считывает данные из файла и заполняет базу данных.

2.2. Реализация механизма конфигурационных файлов

При тестировании приложения выяснилось, что на сервере для папки Data установлено ограничение доступа. Эта проблема создавала трудности для дальнейшей автоматизации процесса шаблонизации. Для решения этой проблемы был использован встроенный механизм конфигурационных файлов. Использование этого механизма также позволило повысить гибкость приложения.

Конфигурационный файл должен иметь строгий синтаксис и структуру. Для соответствия этим требованиям был реализован пользовательский раздел конфигураций (custom configuration section). После его реализации, в него были перенесены данные, ранее находившиеся в файле config.xml.

2.3. Снятие ограничений на количество хранилищ

Взятое за основу приложение решало частный случай бизнес задачи. В частности устанавливалось ограничение на количество складов, равное двум.

Для дальнейшей параметризации приложения была реализована возможность использования N хранилищ.

Для реализации этой задачи были внесены изменения в графический дизайн приложений, их логику и логику сервисной компоненты. В частности изменения в логике сервисной компоненты затронули: API, механизм обмена сообщениями, SOAP механизм.

Также эти изменения были внесены в веб интерфейс.

2.4. Добавление функционала «живой» очереди

Для расширения функционала приложения в него был добавлен механизм «живой» очереди. При этом одно приложение получило возможность работы сразу с несколькими очередями.

В результате пользователь клиентского приложения получил возможность занимать позицию в очереди и возможность покинуть очередь. Так же он может видеть свою позицию в очереди и её текущую длину.

В приложение менеджера был добавлен функционал оператора очереди. Оператор может управлять состоянием очереди и двигать ее. В одной очереди могут находиться сразу несколько операторов, которые могут работать параллельно.

Для обеспечения необходимой функциональности была значительно расширена сервисная компонента. Так же были внесены изменения в механизм обмена сообщениями, а так же API.

Заключение

В данной курсовой работе был продемонстрирован пример создания шаблона приложения. В будущем предполагается дальнейшее расширение его функционала и добавление новых возможностей персонификации.

Следующим шагом к сервису предоставления мобильных приложений для конечного пользователя на основе шаблонов должно стать создание окружения и инфраструктуры. Создание такого окружения позволит автоматизировать и упростить процесс персонализации.

Реализация такого сервиса должна позволить конечному пользователю быстро получить необходимое приложение и легко изменять его в дальнейшем.

Список литературы

1. 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.
2. UBIQMobile documentation [Electronic resource]. — 2015. — URL: <http://ubiqmobile.com/developers/> (online; accessed: 17.12.2015).
3. А.Н.Терехов, В.В. Оносовский. Технология разработки мобильных онлайн сервисов. // Конференция CEE-SECR 2011. С. 1-2
4. V. Onossovski, A.Terekhov. Ubiq Mobile – a New Universal Platform for Mobile Online Services // Proceedings of 6th seminar of FRUCT Program, 2009.
5. Джон Скит. С# для профессионалов. Тонкости программирования. – «Вильямс». с. 608, 2014 .