

Модель информационной системы на базе push-уведомлений

И.О. Федченко, Д.Е. Намиот

Аннотация—В настоящее время тренд, связанный с отказом от присущих телекоммуникационной отрасли методов передачи сообщений (SMS, MMS) и заменой их другими способами обмена информацией является вполне очевидным. Короткие сообщения долгое время были (и являются сейчас) одним из основных способов информирования мобильных абонентов. В данной работе мы рассматриваем возможность построения информационной системы для мобильных пользователей, основанной на так называемых push-уведомлениях. Речь идет о создании инструмента, который позволил бы без программирования создавать системы рассылок уведомлений мобильным пользователям в самых разных областях.

Ключевые слова—уведомления, push, короткие сообщения, рассылка.

I. ВВЕДЕНИЕ

Уже давно в мире существует задача донесения информации от продавца потенциальному покупателю. Помимо массовой рекламы, в последнее время популярность приобрела персонализированная передача информации. Персонализированная передача есть, по сути, доставка определенной информации конкретным потребителям. В нашей работе мы говорим об информировании мобильных пользователей (мобильных абонентов). Долгое время основой такого рода систем доставки информации были короткие сообщения (SMS) и их продвинутый вариант MMS [1]. Именно SMS использовались (используются сейчас) как транспортный механизм в информационных системах для мобильных пользователей. В настоящее же время, например, вполне очевидно смещение интересов мобильных пользователей от SMS к так называемому Instant Messaging [2]. Очевидно, что этот тренд должен как-то отражаться и в информационных системах. Если мобильные пользователи массово уходят из телекоммуникационных сервисов (SMS) в Интернет-приложения (те же упомянутые выше мессенджеры), то должны эволюционировать и системы уведомления (информирования). Другим побудительным мотивом может быть стоимость общения (доставки информации).

Статья получена 20 июля 2015.

И. О. Федченко выпускник факультета ВМК МГУ имени М.В. Ломоносова (e-mail: igor@fedchenko.pro).

Д.Е. Намиот старший научный сотрудник лаборатории ОИТ факультета ВМК МГУ имени М.В. Ломоносова (e-mail: dnamiot@gmail.com).

SMS являются существенным источником дохода телекоммуникационных компаний, и цены на эту услугу не будут сильно снижаться. Наконец, в эпоху развития программных интерфейсов (API) интеграция Интернет-сообщений в приложения будет проще (гибче), чем программные интерфейсы телекоммуникационных операторов. Управление сообщениями может быть более удобно для пользователей, чем операции с SMS.

Все это приводит к тому, что вопрос о замене транспортного уровня в информационных системах для мобильных абонентов становится весьма актуальным. Как один из примеров, можно привести работу [3], где рассматриваются информационные системы на базе Twitter. В настоящей работе изложены результаты дипломной работы, выполненной на факультете ВМК МГУ имени М.В. Ломоносова. Здесь речь идет о построении информационных систем на базе так называемых push-уведомлений.

Это не первая работа, выполняемая в рамках исследования телекоммуникаций в лаборатории ОИТ [4], посвященная push-уведомлениям. Здесь уместно упомянуть работы [5-7]. Основным результатом данной работы является модель автоматического конфигурирования мобильного приложения для получения рассылок, о чем будет рассказано ниже.

II. ЧТО ТАКОЕ PUSH-УВЕДОМЛЕНИЯ

Push-уведомления – это краткие всплывающие уведомления, которые появляются на экране мобильного телефона и сообщают о важных событиях и обновлениях. [8] Они поддерживаются практически в каждой мобильной операционной системе (ОС). Сами по себе они являются средством представления информации пользователю и могут являться интерактивным элементом, позволяющим просмотреть информацию подробнее после осуществления клика по элементу. Они не зависят от наличия или отсутствия интернета, и могут отображаться, например, в зависимости от положения телефона в пространстве (например, используя GPS), или просто уведомлять пользователя телефона о текущем времени суток. Однако мы в данной работе рассматриваем push-уведомления в связке с интернет сервисом, обеспечивающим ведение интернет-рассылки.

Доставка сообщений в системах Push-уведомлений осуществляется по сети интернет, используя клиент-серверную модель. Мобильный клиент (subscriber) при помощи установленного приложения подписывается на

рассылку push-уведомлений, а сервер публикаций (publisher), по наступлению определенного события, осуществляет рассылку подписавшимся клиентам. Таким образом, адресатом сообщения всегда является некоторое приложение. Важно отметить, что push-уведомления позволяют серверам рассылки уведомлять пользователей о наступлениях событий, даже если приложение в данный конкретный момент неактивно. Достаточно, чтобы приложение было установлено, и действовала подписка.

Отправка на сервере, как правило, осуществляется при наступлении некоторого события. Само событие отслеживается некоторым отдельным приложением, и может быть любым: получение сообщения от администратора, или нахождение некоторого количества клиентских устройств поблизости (может использоваться пассивный Wi-Fi или Bluetooth мониторинг) [9].

Системы, позволяющие некоторым образом обрабатывать информацию, в совокупности с системами доставки этой информации и связанные с этим аппаратные, программные и человеческие ресурсы, называются информационными системами [10].

III. СИСТЕМЫ НА ОСНОВЕ PUSH-УВЕДОМЛЕНИЙ

Изначально системы создавались автономными: инициатором события мог быть либо администратор системы, либо выполнение предопределённого условия. Однако следующим этапом в развитии концепции рассылок на базе push-уведомлений явилось появление сервисов, предоставляющих интерфейс программирования приложений (application programming interface, API) для настройки собственных рассылок. Сервисы предоставляют своим пользователям возможность настройки контента рассылки, а так же загрузки клиентских приложений, которые будут являться подписчиками рассылки, реализованных для различных мобильных платформ (iOS, Android и другие).

Эти сервисы являются довольно популярными, однако у них есть один недостаток с точки зрения большинства коммерческих предприятий – это необходимость обладания навыками программирования. Когда у коммерческой организации или, тем более, у некоммерческого предприятия, возникает потребность в организации рассылки (в том числе это может быть рассылка для узкого круга лиц – например, для сотрудников компании), то необходимость нанимать программиста, имеющего навыки создания мобильных приложений, к тому же для нескольких мобильных операционных систем – это если не проблема, то по крайней мере издержки, которых хотелось бы избежать.

Поэтому возникает потребность в следующей ступени развития сервисов – это сервисы, которые, помимо представления API для создания и ведения рассылки, предоставляют мобильное приложение, работающее с данным API и устанавливаемое на мобильное устройство клиента-подписчика, а также

панель управления рассылкой push-уведомлений. Такие сервисы являются уже готовым бизнес-решением для организаций, упомянутых выше: при необходимости вести рассылку определенному кругу лиц, нужно просто зарегистрироваться, далее при необходимости произвести оплату, скачать клиентское приложение и настроить рассылку в личном кабинете панели управления. Такие сервисы являются уже можно назвать реализацией информационной системы. Более подробно перечень основных сервисов этого рода и взаимодействие с ними также рассматривается в следующих разделах.

Очевидно, что раз субъектом подписки на push-уведомления является приложение, то такая информационная система должна предлагать не только систему управления рассылками, но и мобильное приложение для приема сообщений. При этом возникает один важный момент. Раз мы решили отказаться от программирования, мобильное приложение должно быть универсальным. Но это может противоречить политике конкретных компаний (не использовать те же решения, что и конкуренты). Соответственно, универсальное мобильное приложение должно как-то конфигурироваться. Тогда каждый бизнес-пользователь сможет создать “собственную” версию мобильного приложения и распространять ее уже от своего имени, как свое собственное приложение. В случае создания такого мобильного приложения для подписчиков это решается программированием [11]. В нашем же случае идея состоит именно в отказе от программирования. Мы хотим представить некоторый аналог системы управления контентом (CMS) для веб, который охватывает все этапы, включая мобильное приложение.

IV. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Существует довольно большое количество информационных систем, которые так или иначе пытаются предоставить инструмент одним пользователям производить рассылку информации другим пользователям посредством push-уведомлений. В целом структура и функционал таких систем сильно пересекаются, поэтому в данном разделе мы опишем, как эти системы устроены, какие именно возможности могут быть предоставлены пользователям, а также проведём сравнение некоторых популярных сервисов.

Для доставки push-уведомлений от сервера-отправителя к клиенту-получателю в целях экономии ресурсов клиентских устройств все существующие системы использует промежуточное звено. Этим звеном является служба уведомлений (Push Service Provider). Вот несколько примеров основных служб:

GCM (Google Cloud Messaging) для мобильных устройств под ОС Android;

APNS (Apple Push Notification Service) для мобильных устройств под iOS;

MPNS (Microsoft Push Notification Service) для мобильных устройств под ОС Windows.

Эти службы поддерживают очень экономичное соединение с мобильными устройствами и осуществляют доставку push-уведомлений. ОС мобильного устройства перенаправляет уведомление соответствующему приложению. В приложениях под Windows Phone уведомления появляются в верхней части экрана. В приложениях под Android и iOS уведомления появляются в верхней части экрана в панели уведомлений.

Однако эти сервисы предоставляют именно API, под которое можно запрограммировать приложение, с ним взаимодействующее. Также необходимо самостоятельно создать сервис по управлению рассылкой. Как мы уже говорили, это не всегда удобно, что и обусловило появление информационных систем, рассматриваемых далее.

В первую очередь, во всех рассматриваемых решениях пользователю предоставляется возможность осуществлять 3 вида рассылок:

1) Широковещательная рассылка.

После указания контента рассылки, сервер запрашивает список зарегистрированных клиентов, после чего им осуществляется рассылка

2) Персонализированная (индивидуальная) рассылка.

При указании контента рассылки публикатор также указывает уникальный номер получателя (из списка имеющихся у зарегистрированных пользователей), и по этому номеру осуществляется доставка

3) Сегментированная рассылка.

Нечто среднее между первыми двумя видами: публикатор должен указать условие на данные получателя. Сервер выбирает тех подписчиков, данные которых удовлетворяют условию, после чего им осуществляется доставка.

Информационные системы в этой области ставят перед собой следующие основные задачи:

Предоставить пользователю инструмент для создания тематической рассылки

Предоставить возможность осуществлять любой из 3-х видов рассылок (см. выше), а для сегментированной рассылки предоставить как можно более гибкий способ выборки подписчиков для отправки

Предоставить как можно более прозрачный способ подписи клиентского устройства на рассылку. В частности, нас будет интересовать возможность поместить в мобильное устройство пользователей приложение-подписчик на рассылку

Для решения задач из этого списка, пользователям предоставляются следующие инструменты:

Каждому пользователю-публикатору информационной системы выделяется виртуальный личный кабинет, в котором пользователь имеет консоль управления рассылками. Наиболее типичные возможности консоли: создание рассылки по определённой тематике, отправка сообщения (одним из трёх видов push-рассылки), просмотр статистики

открытых push-уведомлений и т.п.

Для осуществления различных видов рассылок используются метки, которые присваиваются каждому отдельному получателю. Это могут быть как произвольные тэги, так и универсальные характеристики, такие как версия операционной системы мобильного телефона. Для персонализированной рассылки могут использоваться уникальные идентификаторы, генерируемые самим мобильным приложением, или же персональные параметры, настраиваемые пользователем – к примеру, день рождения владельца телефона.

Как правило, также есть возможность указать заголовок сообщения и ссылку для перехода при нажатии получателем на сообщение, когда оно появляется в области уведомлений. Может указываться и изображение, появляющееся вместе с сообщением, и даже звуковой сигнал.

Для того, чтобы получатель вообще мог каким-то образом подписаться на рассылку, необходимо, во-первых, создать приложение, которое бы являлось подписчиком в техническом смысле, и во-вторых доставить это приложение конечному потребителю. Для этого применяются самые разнообразные решения (одно из которых предлагается в данной работе и является уникальным). В целом, возможные подходы можно разбить на следующие категории:

Предоставить программный код, который нужно внедрить в собственное мобильное приложение. Такую возможность предоставляют практически все сервисы, и это является наиболее гибким способом доставки, однако в то же время наиболее трудоёмким.

Предоставить готовое к загрузке «тестовое» приложение. В противовес предыдущему подходу, этот способ практически не требует усилий со стороны пользователя-публикатора информационной системы, однако также подход лишён гибкости в настройке приложения

Предоставить более или менее настраиваемый в личном кабинете шаблон мобильного приложения, который пользователь-публикатор может загрузить себе на компьютер. После этого, доработав при необходимости, приложение может быть собрано в подходящей для шаблона среде разработки и выложено как своё собственное

Основной способ донести так или иначе созданное готовое мобильное приложение до пользователя – это выложить его в основной, специализированный для каждой мобильной операционной системы сборник приложений. Такие сборники всегда находятся в публичном доступе и дают разработчикам возможность загружать туда свои приложения. Также публикатор, желающий распространить своё приложение, может отправить его по электронной почте или выложить где-то в публичном доступе ссылку для загрузки приложения.

Встречаются и другие способы решения перечисленных задач, однако нет смысла перечислять их в общем обзоре. Более подробно мы их опишем в сравнительной таблице наиболее популярных

информационных систем на основе push-уведомлений.

V. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Здесь мы представим таблицу сравнения некоторых известных информационных систем, которые используются для проведения информационных

Название сервиса	Поддерживаемые устройства	Консоль управления рассылкой	Поддержка видов рассылок	Возможности настройки содержимого уведомлений	Средства создания мобильного приложения и доставки его пользователям-подписчикам
Amazon Simple Notification Service	iOS, Android, Fire OS, Windows и другие	Есть, стандартная	Все виды	Можно указать тему сообщения, текст, звук, ссылку для перехода и изображение	Готовое приложение и программный код для встраивания в собственное приложение
Windows Azure Notification Hubs	Windows 8, iOS, Windows Phone и Android	Есть, стандартная	Все виды	Аналогично	Программный код для встраивания и тестовое приложение
Urban Airship Push-notifications	Аналогичный набор	Стандартная	Все виды	Аналогично	Программный код, встраиваемый в приложение

Для примера перечислим ещё несколько систем, не углубляясь в их функционал, поскольку он в большей мере аналогичен перечисленным ранее:

- Сервис PUSH-уведомлений Parse
- Сервис PUSH-уведомлений Uniqush – сервис с открытым исходным кодом
- Сервис PUSH-уведомлений PushOver –

рассылок на основе push-уведомлений. Выборка сервисов для сравнения – субъективный выбор автора, однако она достаточно репрезентативна, чтобы показать, какие задачи существующие системы решают, а какие нет.

Таблица 1

имеются мобильное и настольное приложение, ориентированные на распространение среди пользователей подписчиков

Таблица 2

Jeapie	Аналогично	Стандартная	Все виды	Отличительной особенностью является визуальный конструктор уведомлений, который позволяет акцентировать внимание публикатора на визуальной составляющей уведомления, что является очень важным	Готовый исходный код для встраивания в приложения. Обещается шаблон приложения.
Push2Pres s	iOS, Android	Как отдельная CMS, так и плагин для Word Press.	Все виды	Стандартные	В отличие от большинства остальных, используя Titanium SDK, сервис позволяет сконструировать приложение практически полностью онлайн, после чего скачать готовый к сборке шаблон, и собрать его в настольной среде разработки
OpenPush	Всё как у большинства, но для распространения предоставляется готовый шаблон приложения. Мобильные ОС: iPhone, Android, Blackberry, Windows Phone и другие				
Nudge	Аналогично предыдущему				

Таким образом, можно подвести итоги. Для начала отметим те задачи, которые перечисленные сервисы позволяют решить:

Для создания рассылки на основе push-уведомлений необходимо создать одну-две учётные записи, после чего контент рассылки можно задать в консоли управления.

Есть готовые решения для интеграции сервиса с мобильным приложением: от библиотеки, встраиваемой в исходный код приложения, до готового приложения,

настроенного на взаимодействие с конкретным сервисом.

Сопровождение рассылки: будь то платный веб-сервис или сервис с открытым исходным кодом, всё хранение сообщений, гарантию их доставки и защиту информации он берёт на себя.

В целом, не нужно изобретать велосипед: уже имеется высокоуровневая надстройка рассылки, которую лишь нужно настроить пользователю.

Теперь отметим проблему, которая осталась

нерешённой:

Со стороны клиента приложение может быть либо стандартным (готовое решение от сервиса), либо написанным пользователем сервиса самостоятельно с использованием библиотек сервиса или шаблонов. А если пользователь хочет уникальное приложение, однако он далёк от программирования? Если он не хочет или не имеет возможности устанавливать программное обеспечение для сборки мобильного приложения? Придётся искать специалиста. Эта проблема и решалась в рамках дипломной работы.

Веб-сервис, предоставляющий пользователю-публикатору взаимодействовать с системой. Это личный кабинет, включающий в себя как визуальный конструктор мобильного приложения, так и консоль управления рассылкой

Шаблон мобильного приложения, настраиваемый в визуальном конструкторе, а также система сборки приложения из этого шаблона

Система доставки уведомлений. Нужна система, которая позволяла бы осуществлять отправку уведомлений на конкретные собранные приложения.

VI МОДЕЛЬ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Вообще говоря, информационную систему, которая позволила бы решить поставленную задачу, можно разделить на три составляющие:

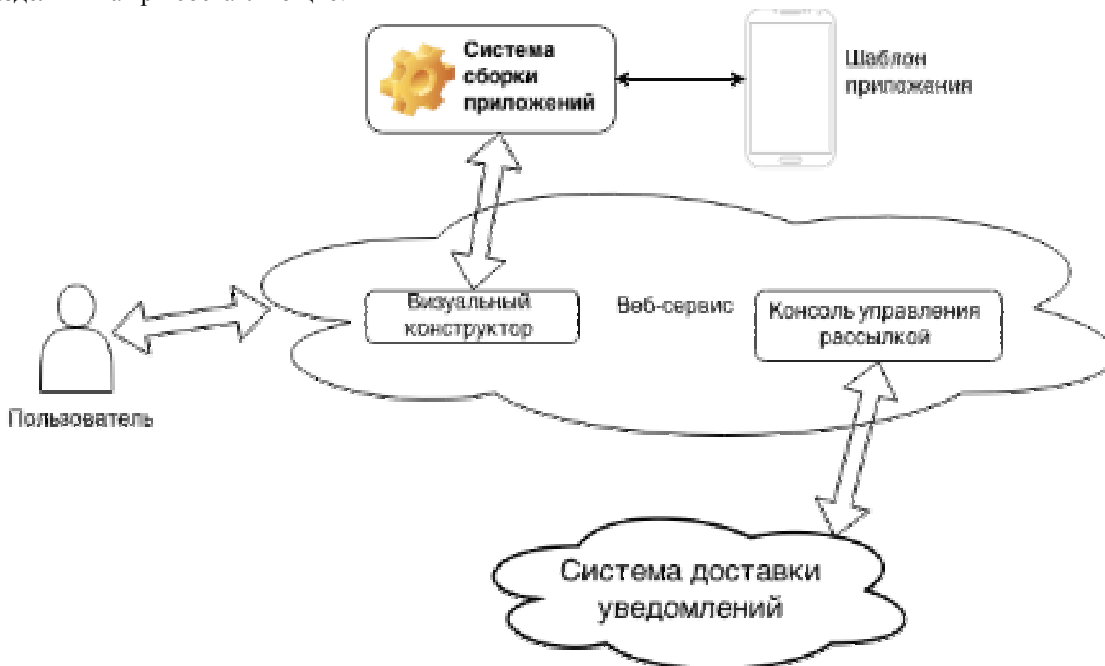


Рис. 1 модель информационной системы

Далее мы опишем модель каждого из этих компонентов.

А. Описание веб-сервиса информационной системы

Веб-сервис нашей информационной системы будет построен следующим образом:

Новый пользователь, желающий стать публикатором рассылки, должен зарегистрироваться, чтобы получить собственный аккаунт и войти в личный кабинет

Отношения между мобильными приложениями, создаваемыми пользователем, и рассылками, которые он будет публиковать, характеризуются связями один-многим, то есть сначала пользователь создаёт сущность приложения, которое будет распространяться среди

пользователей, а затем для этого приложения он создаёт рассылки, на которые это приложение будет подписано.

В каждой рассылке может отправляться, вообще говоря, не ограниченное число уведомлений.

Для того, чтобы созданное приложение можно было публиковать в свободном доступе, оно должно быть подписано уникальным ключом разработчика. Ключи также могут создаваться пользователем в личном кабинете и привязываться к приложениям связью один ключ ко многим приложениям.

Таким образом, связка ключ_разработчика -> приложение -> рассылки -> сообщения_в_рассылках образуют древовидную структуру:

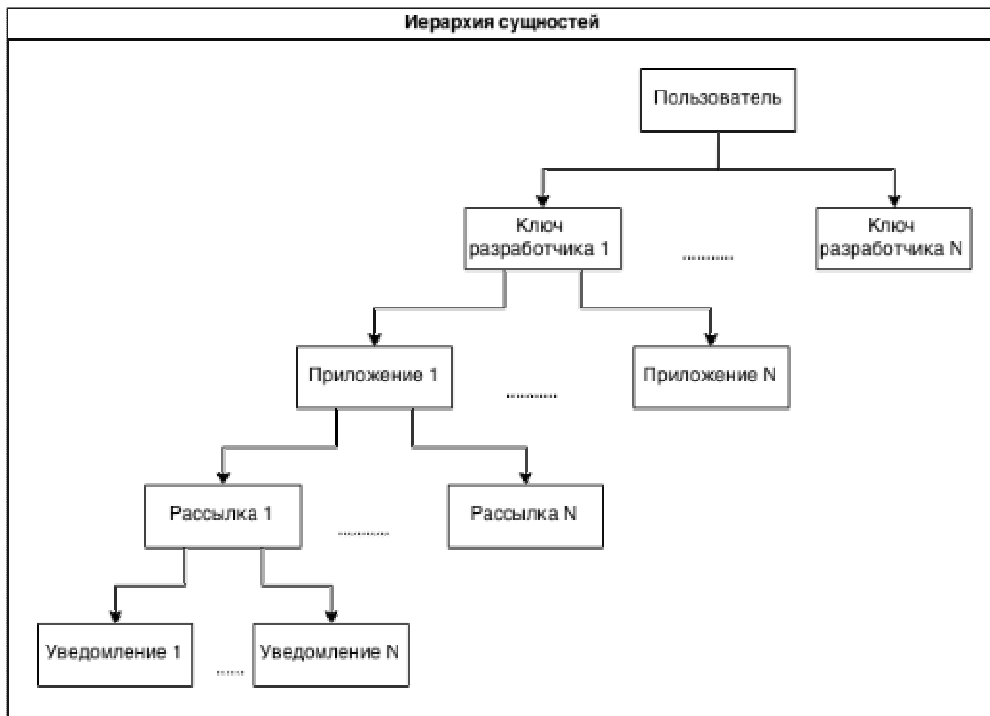


Рис. 2 Структура системы

Стандартный алгоритм действий рядового пользователя-публикатора после регистрации можно разделить на две фазы:

- Создание мобильного приложения для распространения между подписчиками
- Создание и управление рассылкой

Описание действий пользователя для создания приложения можно описать следующим образом:

1) Пользователь в визуальном конструкторе создаёт сущность мобильного приложения. Целью данной работы не является демонстрация возможностей по визуальному построению мобильных приложений, так что для обозначения уникальности приложения достаточно указать логотип приложения (подробнее мы это рассмотрим в подразделе, посвящённом шаблону мобильного приложения).

2) Как во время создания приложения, так и после этого, перейдя на страницу управления этим приложением, пользователь может привязать к приложению один из ключей разработчика (или перейти к мастеру создания ключей для создания такового).

3) Приложение может быть загружено как с привязанным ключом, так и без него. Без привязки ключа оно может быть загружено с целью отладки приложения – тогда оно будет подписано специальным, отладочным ключом, запрещающим публикацию. Если приложение скачано после привязки к нему ключа разработчика, оно готово к публикации, например, в Google Play Market.

Более строго процесс создания приложения

пользователем описывается следующей блок-схемой «Алгоритм создания приложения» (см. ниже).

Далее, после создания приложения, пользователь может создать в нём рассылку. Действия необходимы следующие:

1) В управлении приложением необходимо создать новую рассылку – для этого нужно просто указать тему рассылки. Отметим, что тема может вводиться кириллицей – проблем с кодировкой не возникнет. Указанная тема рассылки будет указываться как заголовок всех уведомлений, отправленных в рамках этой рассылки.

2) В управлении созданной рассылкой можно отправлять уведомления. Для достижения целей, поставленных в данной работе, достаточно задавать в уведомлении текст уведомления и опциональную ссылку для перехода при открытии уведомления на устройстве получателя. Ссылка указывается, если необходимо, чтобы при клике по уведомлению на устройстве пользователя открылся браузер и загрузил страницу по указанному адресу. Это удобно, когда нужно вставить в уведомление графический или звуковой контент. Если ссылка не указывается, при клике на уведомлении будет открыто наше приложение с демонстрацией логотипа при старте. Уведомление будет отправлено на все устройства с установленным в этот момент приложением сразу же после его отправки с этой страницы.

Блок-схема алгоритма создания рассылки и отправки уведомлений представлена ниже.

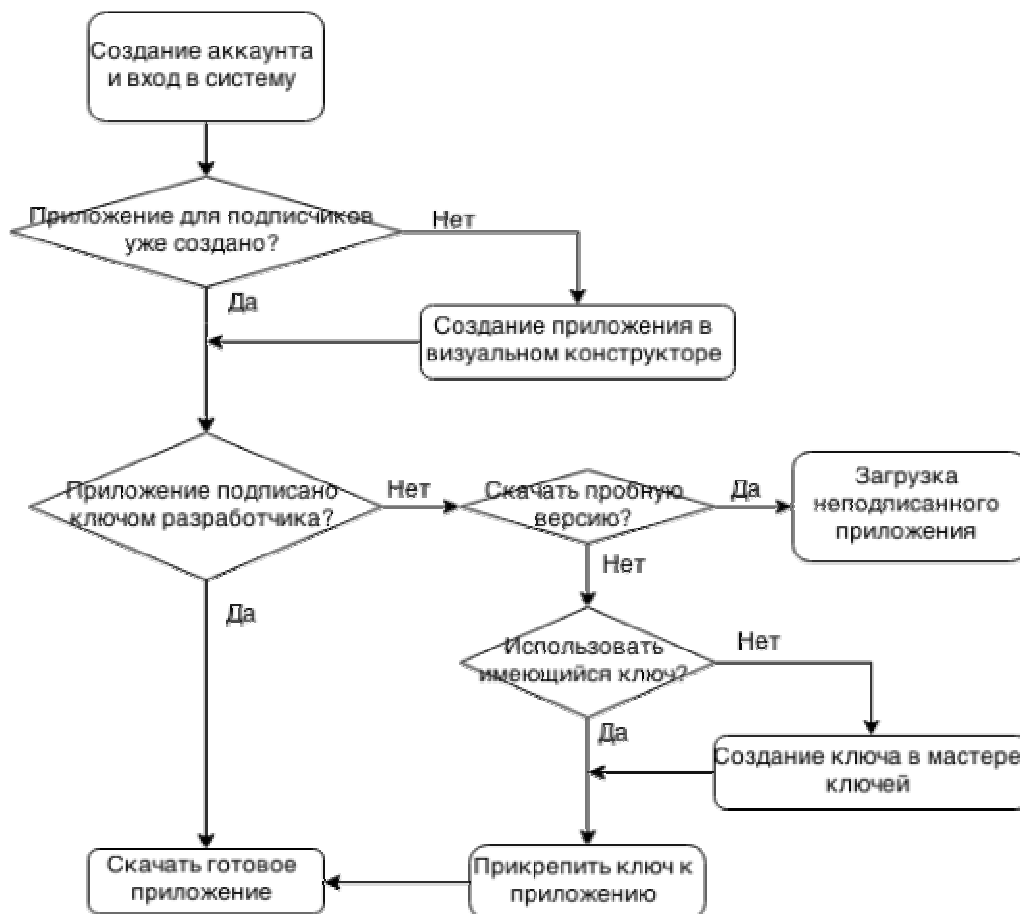


Рис. 3 Алгоритм создания приложения

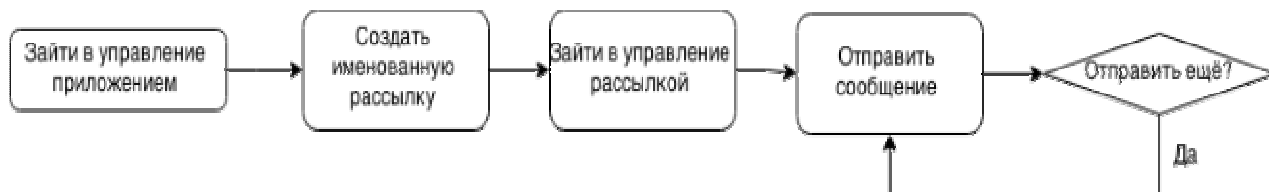


Рис. 4 Алгоритм создания рассылки и отправки уведомления

Далее мы подробнее рассмотрим собираемое мобильное приложение и модель доставки сообщения после того, как оно было опубликовано в одной из рассылок.

В. Описание мобильного приложения

Поскольку построение приложения, предоставляющего весь необходимый функционал для любого публикатора, будь то интернет-магазин или некоммерческая организация, заведомо является очень тяжёлой (если вообще выполнимой) задачей, а создание визуального конструктора, позволяющего достаточно гибко определить входящий в приложение функционал, выходит за рамки данной дипломной работы, мы ограничимся простым приложением, которое было бы уникальным для целей публикатора, и позволяло бы оповещать пользователей о новостях рассылкой push-уведомлений.

Для этого достаточно приложения, которое:

1) При запуске отображает уникальный логотип, задаваемый публикатором в визуальном конструкторе

2) В основной части работы просто отображает историю приходивших уведомлений – чтобы подписчик мог посмотреть последние уведомления и при необходимости перейти по соответствующим ссылкам

Такое приложение вполне подойдёт публикаторам, которые хотели бы просто держать своих подписчиков в курсе последних новостей, и все подробности (в том числе графический и звуковой контент) размещали бы по связанной с уведомлением ссылке.

Для сборки приложения мы будем использовать метод программных тегов. Он заключается в том, что в программном коде приложения фрагменты, которые должны быть настроены непосредственно перед сборкой, помечаются двойными фигурными скобками и некоторым ключевым словом. Например, `<image="{{путь_к_логотипу}}">`, где «путь_к_логотипу» – ключевое слово, которое отличает этот тег от всех остальных.

Для того, чтобы пользователь всегда мог просмотреть полученные новости, вне зависимости от наличия подключения к интернету, история уведомлений будет храниться в локальном хранилище, расположенном в файловой системе операционной системы устройства.

Сама система сборки осуществляет построение приложения в два этапа:

1) Нахождение всех тегов и подстановка вместо них конкретных значений. В нашем случае необходимо для каждого приложения указать его уникальный номер, по которому можно будет адресовать уведомления, а также все параметры, определяемые в визуальном конструкторе. То есть путь к изображению логотипа. Номер приложения используется при запуске процедуры его на сервере-публикаторе, рассылающем уведомления.

2) Когда исходный код приложения готов к сборке,

система собирает из него готовое приложение, при необходимости подключая ключ разработчика, чтобы приложение получилось с возможностью публикации.

Для того, чтобы экономить место на жёстком диске файловой системы сервера, приложения в собранном виде не хранятся – достаточно хранить в базе данных все необходимые для сборки данные. Это также позволяет гарантировать, что пользователь всегда загружает актуальную версию своего приложения – в начале загрузки пользователем оно заново собирается, а по окончании удаляется с сервера.

Таким образом, схематично сборка осуществляется так:

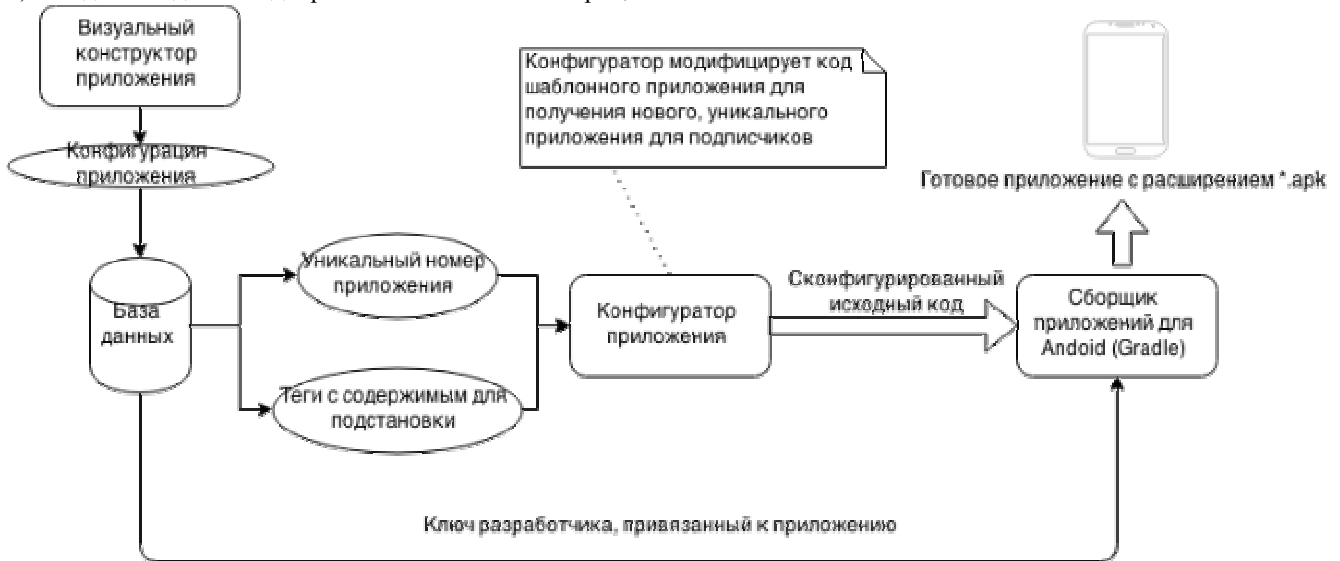


Рис. 4. Модель сборки

Для того, чтобы доставить собранному приложению уведомление, отправленное через личный кабинет, необходима некоторая служба-посредник, гарантирующая доставку уведомлений (то есть его хранение, отслеживание статуса доставки и т.п.). Для этого нужно либо реализовывать систему самостоятельно, либо воспользоваться одним из существующих решений, которые были перечислены в разделе обзора существующих решений.

В качестве посредника было взято одно из существующих решений, наилучшим образом подходящее по тарифному плану – это сервис push-уведомлений Parse. Если рассматривать проект с коммерческой точки зрения, то при превышении лимита бесплатной отправки уведомлений с пользователей

сервиса может взиматься плата, отвечающая тарифным планам выбранного нами сервиса.

Модель работы службы доставки уведомлений следующая: после того, как собранное приложение устанавливается на устройстве подписчика, оно регистрируется в сервисе Parse, указывая свой уникальный идентификатор. Далее, при отправке уведомления через наш веб-сервис, посредством внешнего API оно перенаправляется в сервис посредника с указанием уникального идентификатора приложения, которому его необходимо отправить. Далее сервис-посредник просматривает все зарегистрированные экземпляры приложений, выбирает из них те, которые соответствуют полученному идентификатору, и рассылает уведомление этим приложениям. Схема представлена ниже:

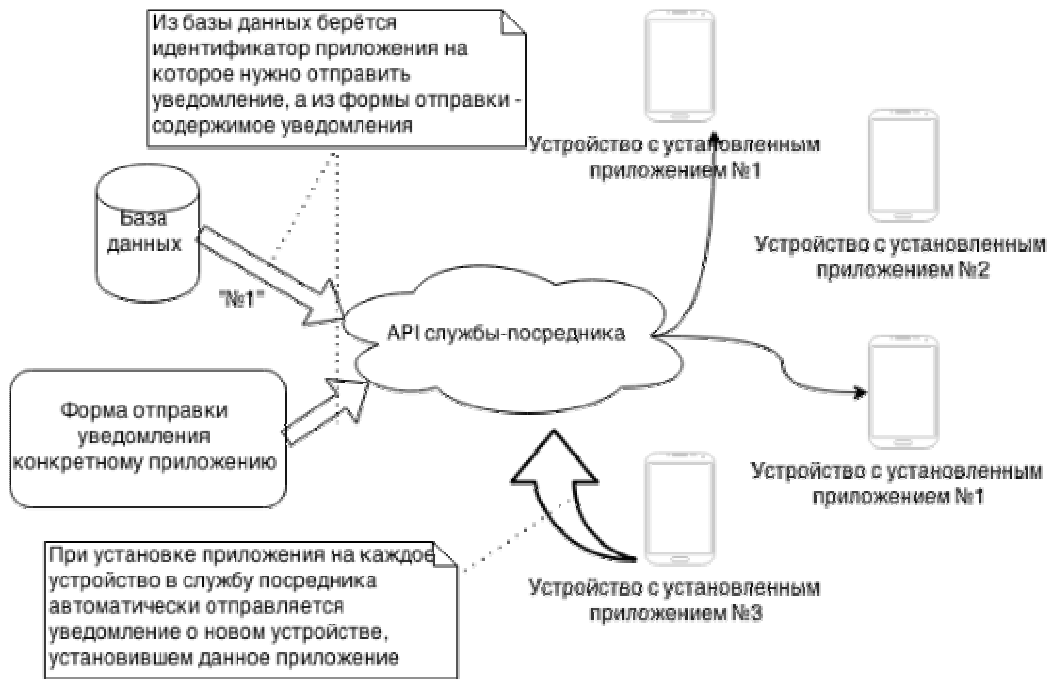


Рис. 5. Схема рассылки

С. Описание возможных модификаций и расширений

Здесь мы перечислим примеры модификаций и расширений, которые могут быть позже реализованы:

1) Увеличение возможностей визуального конструктора. Используя методологию программных тегов, можно фактически встраивать любой код в мобильное приложение.

2) Добавление полей при отправке уведомления. Можно расширить перечень полей до соответствия набору сервиса посредника. Это может быть звуковой сигнал, или графический контент

3) По аналогии с ОС Android, можно создавать приложения для ряда других мобильных операционных систем. Нужно лишь доработать совместимые системы сборки.

4) Можно добавить автоматическое выкладывание приложений в публичный доступ, рассылку ссылок на приложение на заданный список электронных почт и так далее

VII ПЕРСОНАЛИЗАЦИЯ УВЕДОМЛЕНИЙ

Персонализация уведомлений подразумевает использование в содержимом уведомления персональной информации получателя. Подобное часто обусловлено, например, маркетинговыми стратегиями.

Для реализации такого функционала необходимо, во-первых, получить средство сбора таковой информации от пользователя, и во-вторых средство указания в тексте уведомления некоторой переменной, принимающей значение, зависящее от полученной персональной информации.

Наилучшим источником персональной информации является приложение, устанавливаемое пользователем. В первую очередь, если при установке пользователь подтвердил разрешение приложению собирать

персональную информацию, приложение способно собирать такую информацию в фоновом режиме и отсылать на сервер. Альтернативой является встраивание в приложение некоторой формы пользовательского ввода, в которой необходимую нам информацию пользователь укажет самостоятельно. Для хранения персональной информации можно использовать хранилище сервиса Parse и его API для загрузки/получения из него информации.

Для внедрения персональной информации в текст уведомления можно использовать уже рассмотренный нами метод тегов, заменяемых на некоторое содержимое. Например, `{{name}}` в тексте уведомления может автоматически заменяться на имя получателя, если таковое имеется, или некоторое предустановленное значение.

Не смотря на естественное желание публикатора донести информацию до как можно большего количества людей, не все обладатели приложения могут хотеть быть подписчиками той или иной рассылки. Напомним, что изначально для приложения создаётся набор рассылок, все уведомления из которых принимаются этим приложением.

Для того, чтобы пользователь мог отписаться от конкретной рассылки, достаточно дать пользователю возможность сообщить об этом из своего приложения – например, при длительном нажатии на уведомление высветить опцию «отписаться от данной рассылки». После этого можно воспользоваться вызовом к API сервиса Parse, добавив данную тему рассылки в список рассылок, игнорируемых данным приложением.

В данный момент все уведомления одного пользователя отправляются синхронным образом – пока не отправится первое уведомление, невозможно отправить второе, и так далее. Это неизбежно влияет на пропускную способность сервиса. В данный момент такая реализация использовалась для того, чтобы

затруднить быстрое исчерпание лимита бесплатных уведомлений, которые предоставляются сервисом Parse, однако в будущем можно запускать отправку уведомлений в асинхронном режиме, например с помощью технологии AJAX, что позволит значительно повысить скорость отклика сервиса.

Для целей данной работы достаточно управления рассылки из личного кабинета пользователя, однако нередко подобные действия удобнее производить из некоторого собственного приложения, или стороннего веб-сайта, к которому у публикатора есть доступ. На этот случай имеет смысл предоставить возможность вызова некоторых функций API для, например, отправки уведомления в рамках уже имеющейся рассылки.

Поскольку наш веб-сервис построен по принципу REST, он уже полностью готов к интеграции в него REST API – то есть отправки специальных типов запросов по протоколу HTTP для изменения состояния сервиса. К примеру, можно указать, что при получении запроса HTTP PUT, содержащем приложение, рассылку и текст уведомления, должна производиться отправка соответствующего уведомления. Однако для того, чтобы авторизация пользователя, отправляющего подобный запрос, была надёжной, есть смысл изменить протокол взаимодействия с API сервиса на HTTPS, предоставляющий защищённое соединение.

VIII СРАВНЕНИЕ ПОЛУЧЕННОГО РЕШЕНИЯ С СЕРВИСОМ SMS-РАССЫЛКИ

Как уже отмечалось ранее, предшественником и в некотором смысле конкурентом информационных систем на базе push-уведомлений являются системы, основанные на SMS-рассылках. Помимо преимуществ полученного нами решения перед другими аналогичными информационными системами (преимущества вытекают из формулировки поставленной в данной работе задачи), логично было бы провести сравнение с сервисом, осуществляющим SMS-рассылку.

В сравнении будем рассматривать пункты, описанные в предыдущем подразделе, реализованными – это позволит сделать акцент в первую очередь не на функциональности (которую в таком случае можно реализовать приблизительно одинакового уровня), а на различии технологий push и SMS самих по себе.

Отметим, что перечисленные достоинства одной системы являются недостатками другой системы. Общие достоинства и недостатки указывать в сравнении нет необходимости.

Достоинства SMS-рассылок:

- 1) Нет необходимости ставить специальное приложение
- 2) Нет ограничения на операционную систему устройства – поддерживаются абсолютно все модели мобильных телефонов
- 3) Для того, чтобы подписать владельца устройства к рассылке, нужен только его номер телефона
- 4) Можно отправить уведомление «вслепую» по

случайному номеру – иногда это бывает полезным

5) На многих моделях телефонов для того, чтобы добавить публикатора в чёрный список, требуется установка отдельного приложения. А на некоторых это вообще не возможно (сомнительное достоинство, но всё же)

6) Нет зависимости от интернета

7) Простота организации – кроме подписки на сервисе рассылок вообще ни о чём больше не надо думать

Достоинства рассылок на основе push-уведомлений:

1) Простой доступ к персональным данным пользователя

2) Понятный способ подписаться/отписаться от рассылок как внутри приложения, так и посредством установки/удаления приложения

3) Возможность организовать обратную связь средствами приложения (например, упомянутые ранее формы пользовательского ввода)

4) Потенциально более широкие возможности по отображению уведомлений - например, иконка, предустановленная публикатором

5) Требование к развитой операционной системе мобильного телефона позволяет гарантировать наличие таких данных, как, например, GPS-координаты. Отсюда всевозможные преимущества вроде возможности проведения гео-позиционирования и отправки уведомлений в зависимости от местонахождения пользователя.

6) Дешёвая стоимость отправки уведомления (в тысячи раз дешевле SMS)

7) В целом, с помощью технологии push-уведомлений можно построить полностью интерактивный диалог с пользователем

Как видно из этих перечней, каждая технология всё же имеет право на жизнь, обладая своими уникальными возможностями. Выбор должен производиться в зависимости от конкретной отрасли, в которой необходимо использовать рассылку уведомлений.

Однако хочется отметить, что используя в качестве основы результаты, полученные в данной работе, можно решить комплекс проблем, что позволит использовать информационные системы на основе push-уведомлений гораздо более широкому кругу пользователей.

IX ОПИСАНИЕ ПРАКТИЧЕСКОЙ РЕАЛИЗАЦИИ

Все исходные коды программы выложены в публичном репозитории сервиса GitHub (ссылку для клонирования репозитория см. в [12]).

Для разработки проекта была выбрана среда Android Studio. На данный момент оно состоит из двух частей: демонстрации логотипа при запуске и перечня полученных ранее уведомлений. Для примера, нами было собрано приложение с логотипом – изображением пингвина, а также отправлено на него 4 уведомления: 3 с обычным текстом, и 4ое с привязанной ссылкой для перехода.

Ниже показаны скриншоты мобильного экрана,

демонстрирующие отображение логотипа, перечень полученных уведомлений, историю уведомлений и результат клика по уведомлению с ссылкой (мы указали <http://ya.ru>). Заметим, что при нажатии на уведомление без привязанной ссылки просто открывается история уведомлений с демонстрацией логотипа приложения:

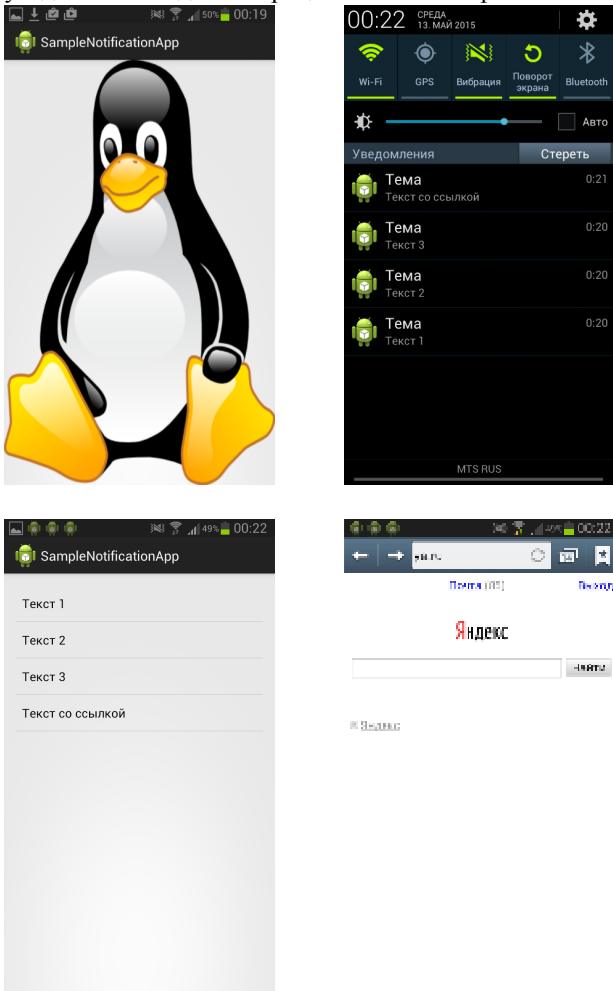


Рис. 6. Мобильное приложение

Для написания самой программы веб-сервиса, разумно использовать некоторый framework - программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта [13]. Учитывая то, что веб-сервис может потенциально расширять свой функционал, а также выбор серверной операционной системы, удобно использовать в качестве основного языка разработки язык программирования Python. Обилие библиотек для этого языка оставляет широкие возможности для расширения функционала сервиса без значительных усилий. Для данного языка одним из самых популярных фреймворков является Django Framework – его и было решено использовать. Помимо популярности, он обладает тем преимуществом, что активно поддерживает принцип DRY – don't repeat yourself (в переводе: не надо повторять самостоятельно, то есть, нет смысла реализовывать то, что до вас реализовывали другие). На практике это означает широкий выбор плагинов, которые позволят в будущем расширять функциональность сервера без значительных усилий

(или по крайней мере усилия будут меньшими, нежели для большинства других фреймворков).

Для хранения данных пользователей используется реляционная база данных под управлением MySQL. Веб-сервис предоставляет в публичный доступ вебсайт, на котором пользователи-публикаторы создают приложения и отправляют уведомления, а также в целом управляют личным кабинетом. Для управления используются обычные формы пользовательского ввода, передаваемые методом HTTP POST. Для отправки уведомлений, как уже было описано в разделе описания решения, используется API веб-сервиса Parse – в нашем случае это просто отправка POST запроса с соответствующими параметрами:

```
import json, httplib
connection =
httplib.HTTPSConnection('api.parse.com',
443)
connection.connect()
request = {
    "where": {
        "deviceType": "android",
        "appID":
"%i"%message.theme.application.id,
    },
    "data": {
        "alert": message.text,
        "title": message.theme.name,
    }
}
if message.url is not None and
len(message.url) > 0:
    request["data"]["uri"] =
message.url

connection.request('POST', '/1/push',
json.dumps(request), {
    "X-Parse-Application-Id":
"SOME_ID", #hidden
    "X-Parse-REST-API-Key":
"SOME_KEY", #hidden
    "Content-Type":
"application/json"
})
result =
json.loads(connection.getresponse().read
())
return result
```

Для сборки приложения создаётся копия исходного кода шаблона, модифицируется согласно уникальным настройкам приложения, заданным пользователем в визуальном конструкторе, после чего производится сборка приложения и выдача готового *.APK файла пользователю для загрузки.

X ЗАКЛЮЧЕНИЕ

В работе представлены модель и практическая реализация информационной системы для мобильных пользователей, основанной на push-уведомлениях. Созданная система позволяет создавать групповые и

индивидуальные системы рассылки информации, управлять данными рассылок и, главное, снабжать получателей рассылок собственными мобильными приложениями, которые конфигурируются непосредственно самими авторами рассылок. При этом персонализация мобильных приложений не требует навыков программирования.

БИБЛИОГРАФИЯ

- [1] Rittippant, N., Witthayawarakul, J., Limpiti, P., & Lertdejdech, N. (2009). Consumers' perception of the effectiveness of Short Message Service (SMS) and Multimedia Message Service (MMS) as marketing tools. *Proceedings of World Academy of Science: Engineering & Technology*, 53, p815.
- [2] Grinter, R. E., & Palen, L. (2002, November). Instant messaging in teen life. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work* (pp. 21-30). ACM.
- [3] Намиот Д. Е. Twitter как транспорт в информационных системах // *International Journal of Open Information Technologies*. – 2014. – Т. 2. – №. 1. – С. 42-46.
- [4] Гурьев Д. Е., Намиот Д. Е., Шнепс М. А. О телекоммуникационных сервисах // *International Journal of Open Information Technologies*. – 2014. – Т. 2. – №. 4. – С. 13-17.
- [5] Павлов А. Д., Намиот Д. Е. Системы для поддержки push-уведомлений // *International Journal of Open Information Technologies*. – 2014. – Т. 2. – №. 7. – С. 37-44.
- [6] Павлов А. Д., Намиот Д. Е. Информационные системы на основе push-уведомлений // *International Journal of Open Information Technologies*. – 2014. – Т. 2. – №. 8. – С. 11-19.
- [7] Павлов В., Намиот Д. Анализ и Разработка Системы Push-уведомлений с Использованием Технологий Google // *International Journal of Open Information Technologies*. – 2013. – Т. 1. – №. 3. – С. 20-24.
- [8] Chetan, S., Kumar, G., Dinesh, K., Mathew, K., & Abhimanyu, M. A. (2010). Cloud computing for mobile world. available at chetan. ueuo. com.
- [9] Sneps-Sneppe, M., & Namiot, D. (2013). Spotique: A new approach to local messaging. In *Wired/Wireless Internet Communication* (pp. 192-203). Springer Berlin Heidelberg.
- [10] William S. Davis, David C. Yen. // *The Information System Consultant's Handbook. Systems Analysis and Design.*: CRC Press, 1998.
- [11] Vorona T. Ukraine startup Jeapie got 50 000\$ of investments from Digital Future [URL] // (<http://ain.ua/2014/11/18/550918>)
- [12] Федченко И.О. Репозиторий с исходными кодами программ дипломной работы «Информационные системы на базе push-уведомлений» <https://github.com/IgorFedchenko/NotificationService.git>
- [13] Фаронов В. Создание приложений с помощью С# // *Руководство программиста*. М.: Эксмо. – 2008.
- [14] Holovaty, A., & Kaplan-Moss, J. (2009). *The definitive guide to Django: Web development done right*. Apress.

On Model of Information System Based on push-notifications

Igor Fedchenko, Dmitry Namiot

Abstract— Currently, the trend associated with the rejection of the telecom-related methods for messaging (Short Messaging Service, Multimedia Messaging Service) is obvious. Telecom-related approaches for messaging are replaced by other methods of exchange of information. Short messages have long been (and are now) one of the main ways of informing the mobile subscribers. In this paper, we consider the possibility of building an information system for mobile users, based on the so-called push-notifications. It is about creating a tool that would create a system without programming mailing notifications to mobile users in a variety of areas.

Keywords—notifications, push, short messages, message delivery.