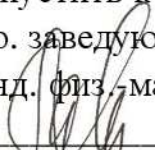


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра информационных технологий

Допустить к защите
И.о. заведующего кафедрой
канд. физ.-мат. наук, доц.
 О. В. Гаркуша
(подпись)

_____ 2019 г.


ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)


МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИНТЕРАКТИВНЫХ
ВЗАИМОДЕЙСТВИЙ ПОЛЬЗОВАТЕЛЕЙ С ОРГАНИЗАЦИЯМИ
ГОРОДСКОЙ ИНФРАСТРУКТУРЫ

Работу выполнил _____  В. А. Ардышев
(подпись)

Направление подготовки 01.03.02 «Прикладная математика и информатика»

Направленность (профиль) «Системное программирование и компьютерные технологии (Математическое и программное обеспечение вычислительных машин)»

Научный руководитель
канд. физ.-мат. наук, доц. _____  Е. П. Лукащик
(подпись)

Нормоконтролер
ст. преп. _____  А. В. Харченко
(подпись)

Краснодар
2019

РЕФЕРАТ

Выпускная квалификационная работа 47 с., 3ч., 20 рис., 17 источников.

ОРГАНИЗАЦИЯ, ПРИЛОЖЕНИЕ, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ПРИЛОЖЕНИЕ РЕАЛЬНОГО ВРЕМЕНИ, МОБИЛЬНЫЙ СПРАВОЧНИК

Объектом исследования данной работы является сфера взаимодействия клиентов и организаций.

Предмет исследования – возможности мобильной разработки для организации взаимодействия клиентов и организаций.

Цель работы – разработка трехзвенного приложения с клиентским мобильным приложением, предоставляющего справочные данные об организациях города Краснодара с возможностью оставления отзывов и оценок, содержащее настраиваемую новостную ленту организаций и позволяющее напрямую взаимодействовать с представителями организаций.

В результате проведенной работы было разработано мобильное приложение «Организации Краснодара», которое предоставляет справочные данные об организациях города Краснодара, позволяет оставлять отзывы и оценки, предоставляет настраиваемую новостную ленту организаций, а так же открывает возможность взаимодействия с представителями городских организаций.

СОДЕРЖАНИЕ

Введение.....	4
1 Рынок мобильных приложений	7
1.1 История рынка мобильных приложений.....	7
1.2 Городские мобильные приложения	9
1.3 Анализ мобильного рынка города Краснодара.....	10
1.4 Итоги анализа рынка. Постановка задачи.....	11
2 Клиент-серверное взаимодействие.....	14
2.1 Архитектура клиент-сервер	14
2.2 Способы и форматы передачи данных	17
2.3 Системы и приложения реального времени.....	19
2.4 Особенности реализации мобильных приложений реального времени.....	22
3 Реализация приложения	27
3.1 Средства разработки	27
3.2 Разделение сервера приложения на два самостоятельных сервера.....	29
3.3 Пользовательский интерфейс приложения	30
Заключение	45
Список использованных источников	46

ВВЕДЕНИЕ

Первые мобильные устройства не были компактных размеров, их использование не представлялось возможным без доступа к электросети, потому что они шли в комплекте с проводом, а высокая стоимость такого устройства сильно ограничивала его распространение среди населения.

Мобильные технологии с тех пор сделали большой шаг вперед, однако качественный рост технологий в этой сфере произошел за последнее десятилетие. Сегодня мобильные устройства компактны, удобны, производительны и их роль в повседневной жизни сильно выросла. Такое распространение смартфонов открыло широкие возможности для мобильной разработки.

Популярными направлениями мобильной разработки на текущий момент времени является сфера развлекательных мобильных технологий и социальные сети. Но такое бурное развитие мобильной индустрии и распространение смартфонов не могло не затронуть сферу хранения данных. Современный обладатель смартфона привык к возможности быстрого поиска информации в сети, к ощущению общности с информационным сообществом и коммуникации через Интернет. Таким образом, существуют перспективы развития мобильных приложений, открывающих пользователю доступ к ежедневно востребованным данным, предоставляющих возможность коммуникации посредством сети Интернет и др.

Жители любого города в ходе своей жизнедеятельности постоянно взаимодействуют с компаниями и организациями городской инфраструктуры. Помимо поиска справочной информации, такой как местоположение организации, контакты, отзывы и прочее, они вынуждены искать удобные современные способы взаимодействия и коммуникации. Организации в свою очередь заинтересованы в привлечении новых клиентов и используют для этого разнообразную рекламу. Однако многие виды прямой рекламы на сегодняшний день не дают существенного эффекта [1].

Разрабатываемое приложение должно реализовать новые подходы к привлечению клиентов, позволит горожанам взаимодействовать с организациями, а так же самостоятельно находить для себя интересующую рекламную информацию в виде акций и предложений, анализировать ее и выбирать наиболее подходящие компании, предоставляющие необходимые услуги.

Объектом исследования данной работы является сфера взаимодействия клиентов и организаций.

Предмет исследования – возможности мобильной разработки для организации взаимодействия клиентов и организаций.

Целью работы является разработка трехзвенного приложения с клиентским мобильным приложением, которое:

- 1)предоставляет справочные данные об организациях города Краснодара;
- 2)дает возможность оставлять отзывы и оценки;
- 3)содержит настраиваемую новостную ленту организаций;
- 4)позволяет напрямую взаимодействовать с представителями организаций.

Для достижения поставленной цели, необходимо решить следующие задачи:

- 1)исследовать рынок и определить направления развития приложения;
- 2)изучить подходы к реализации мобильных приложений, работающих в режиме реального времени;
- 3)произвести оценку возможных нагрузок на систему и в соответствии с этим выбрать инструменты разработки мобильного приложения и сервера;
- 4)разработать структуру базы данных;
- 5)реализовать сервер и клиентское мобильное приложение, позволяющее работать с базой предприятий города Краснодара.

Результаты поиска решений поставленных задач будут полезны в области разработки приложений-справочников, в частности мобильных

приложений, приложений от городской администрации, а так же начинающим разработчикам сферы мобильных технологий в качестве набора основных подходов к мобильной разработке подобных приложений.

1 Рынок мобильных приложений

1.1 История рынка мобильных приложений

Мобильное приложение – это компьютерная программа, созданная специально для использования в мобильном телефоне, смартфоне или коммуникаторе, которая предназначена для выполнения той или иной задачи. На первых мобильных устройствах было установлено малое количество приложений, поставляющихся совместно с программным обеспечением устройства. Первым мобильным приложением можно считать телефонный справочник – та часть программного обеспечения аппарата, которая упорядочивала контакты пользователя [2].

Время появления первого мобильного приложения, установленного на телефон поверх уже имеющегося программного обеспечения, можно отнести к концу 90-х годов прошлого века, когда сотовая связь стала постепенно входить в жизнь миллионов людей во всем мире. С появлением в 1997 году на рынке сотовой связи технологии WAP, позволяющей выходить в Интернет с помощью мобильного телефона, количество программных приложений для мобильных устройств стало расти. Дело в том, что ранее установка любого стороннего приложения на устройство осуществлялась через DATA-кабель, который шел в комплекте не с каждым телефоном. Сегодня поставить на мобильное устройство какую-либо программу стало гораздо проще.

К 2000 году началось бурное развитие рынка мобильного контента в целом и мобильных приложений в частности. Появление новых технологий передачи данных с помощью сотовой связи (GPRS, EDGE) позволило удешевить мобильный интернет-трафик, что являлось определенным сдерживающим фактором роста рынка мобильных технологий.

В 2000-х годах рынок мобильных устройств сотовой связи стали постепенно завоевывать смартфоны и коммуникаторы. Обладая более широкими возможностями и производительностью, они отличались от

обычных мобильных телефонов наличием достаточно развитой операционной системы (Windows Mobile, Symbian OS, RIM, Android, Mac OS) открытой для разработки программного обеспечения сторонними разработчиками [2].

Начиная с 2000-х годов распространение мобильных устройств и приложений стремительно росло, следуя за технологическим прогрессом в данной области.

2007 год стал одним из самых важных в истории развития карманных компьютеров и смартфонов. 9 января 2007 года на выставке Macworld Conference & Expo был представлен iPhone, перевернувший все представления пользователей о смартфонах. 2007-й стал также годом анонса Android, который явно произошел под давлением стремительно набирающего популярность продукта от Apple. Тогда Android фигурировал только в виде бета-версии комплекта для разработчиков (SDK), оснащенного эмулятором.

2009 год можно по праву считать годом расцвета Android как мобильной ОС. Производители мобильной техники начали присматриваться к Android и анонсировать свои первые устройства на его основе, Google продолжала спешно дорабатывать ОС, залатывая множественные пробелы в ее дизайне и функциональности.

Развитие операционных мобильных систем открывало возможности развития в области мобильных приложений. Операционные системы предоставляли все больший функционал разработчикам мобильных приложений. Мобильная разработка встала по востребованности рядом с компьютерной и нашла свою многочисленную аудиторию.

AppAnnie – крупная платформа мобильной аналитики подвела итоги развития мобильного рынка на 2015 год. Стало ясно, что в будущем веб-сайты уже не смогут конкурировать с приложениями. С каждым днем люди тратят на приложения все больше времени и средств, используя их в разных сферах жизни. Внедрение таких устройств и платформ как Apple TV, AppleWatch и аналогичных им разработок на базе Android, в будущем

способствует тому, что пользователи будут становиться все более «мобильными».

1.2 Городские мобильные приложения

Сегодня каждый человек достаточно сильно зависит от используемых им мобильных устройств. Многие ежедневно используют смартфон для проверки электронной почты, чтения новостной ленты, общения в социальных сетях и т.п. Трудно представить каким длинным может быть данный список востребованных мобильных ресурсов. В нем обязательно можно отыскать ресурсы, связанные с накоплением справочной информации в тех или иных сферах жизни. Мобильные устройства заметно упрощают жизнь пользователям, предоставляя возможность получить в любом месте и в любое время быструю справку, сэкономить время на поиск информации в Интернете с использованием любой поисковой системы. Для получения необходимой информации пользователю не требуется совершать сложных действий.

Городские мобильные приложения связаны с конкретным городом своим контентом и не вызывают интереса у чужой аудитории. Но для граждан конкретного города такое приложение может быть весьма удобным и полезным. Как правило, в нем объединены городские информационные порталы, выведены городские службы, ведется новостная лента и т.п. Однако широкий функционал приложения для некоторых пользователей может быть излишним, поэтому возможны разбиения городских мобильных приложений на самостоятельные приложения, предназначенные для конкретных задач. Так, например, обычно выносятся отдельным приложением тематика муниципального транспорта, новостная тематика и т.д.

1.3 Анализ мобильного рынка города Краснодара

На рынке городских мобильных приложений Краснодара имеются такие приложения, как «Краснодар» (администрация МО г. Краснодар), «Краснодар+» и «Мой Краснодар». Не будем включать в рассмотрение приложения связанные с муниципальным транспортом, так как эта тема слишком далека от поставленной задачи. Рассмотрим каждое приложение в отдельности и определим, какие функции в них реализованы, и на каком уровне они предоставляются.

Приложение «Краснодар» направлено на оповещение пользователей об основных городских мероприятиях, содержание фотогалереи событий и достопримечательностей города, предоставляет некоторые данные о городских службах и содержит вкладку с городским транспортом. Так же предоставляется информация о структуре и составе администрации города. В целом приложение полезно в большей степени для приезжего из другого города человека, который только знакомится с Краснодаром.

Приложение «Краснодар+» появилось на рынке в конце 2017 года и еще не набрало своей аудитории, хотя и предоставляет огромный набор разнообразных функций. Здесь из интересующего нас направления имеется вкладка «Компании и услуги». Реализована возможность добавления организаций и отзывов. Однако взаимодействие клиента и организации реализовано плохо. Нет возможности вести хотя бы с некоторыми компаниями активный диалог по поводу бизнес сотрудничества, получения услуги и т.п. Лента акций и предложений имеется, но в ней нет никакого фильтра, поэтому пользователь не может быстро получить интересующую его информацию. Ему необходимо пролистывать много лишнего контента, тратить мобильный трафик, терять время, при этом может оказаться, что интересующих его акций на данный момент вообще нет.

«Мой Краснодар» – еще одно развивающееся приложение, в котором реализованы пока еще не все возможности. Новости города, справочник

городских служб и разнообразный развлекательный контент предоставляется пользователям данного приложения. Здесь не реализовано добавление организаций и отзывов.

Представленные выше приложения, которые имеют справочники, не имеют гибкого выбора сохранения данных на устройстве. Либо это происходит автоматически с полным переносом данных на устройство пользователя, либо возможность доступа к справочникам без доступа к сети Интернет.

Нельзя опустить рассмотрение предоставляемого функционала приложения «2ГИС». Оно не специализируется только лишь на одном городе и связано больше с тематикой карт. Однако оно содержит в себе справочник организаций и поддерживает некоторые функции по взаимодействию с ними. Рассмотрим их подробнее.

В «2ГИС» пользователь может добавлять организацию, оставляя любой из ее контактов. Модераторы приложения проверяют существование такой организации и в случае успеха могут добавить ее в справочник. По каждой организации есть возможность посмотреть отзывы и рейтинг, а так же оставить свое мнение. Справочник работает и без доступа к Интернету, однако за это пользователь платит хранением на своем устройстве большого блока данных, а так же вынужден его своевременно обновлять.

1.4 Итоги анализа рынка. Постановка задачи

В ходе анализа рынка мобильных приложений с соответствующей тематикой были выявлены возможности модернизации некоторых предоставляемых ими функций, а так же добавление новых более гибких инструментов настройки работы приложения.

Одной из основных идей работы станет удобная категоризация контента. Пользователь должен легко «находить путь» к интересующим его

данным за несколько движений пальцев по экрану устройства. Это не только экономит время пользователя, но и экономит трафик приложения.

Безусловно, важной деталью проекта должна являться гибкая настройка сохранения данных на устройстве. У пользователя должен быть выбор: сохранять все возможные данные на его устройство, либо же сохранять лишь некоторые из них. Существуют пользователи, которые вообще не нуждаются в сохранении данных на своем устройстве по причинам экономии памяти или каким-либо другим.

Благодаря социальным сетям пользователи мобильных устройств привыкли к такому формату представления информации, как новостная лента. В разрабатываемом приложении новостная лента будет наполнена новостями и акциями организаций, то есть рекламой. В данном случае под новостной лентой приложения будет пониматься набор карточек от разных организаций, содержащих изображение и текст, в котором представитель компании сообщает о проводимой акции, ее продолжительности и пр. Карточки выстраиваются в вертикальный список, который можно пролистывать. Идея ленты состоит в том, что реклама не навязывается пользователю, он сам является инициатором просмотра рекламы исходя из своих потребностей на текущий момент времени. Если все акции и новости всех организации будут перемешаны в новостной ленте, то пользователь будет вынужден просматривать лишнюю не интересующую его информацию. Для решения данной проблемы в приложении будет реализована возможность выбора категории организаций, что заметно упростит пользователю работу с контентом.

Дополнительной удобной функцией приложения станет возможность прямого взаимодействия пользователя с организациями города. Взаимодействие с организацией будет представлять собой чат с представителями организации, которые могут ответить на интересующие пользователя вопросы, оформить заказ, уточнить наличие товара. Данные функции доступны и по телефону. Однако, передвигаясь по городу на

общественном транспорте в шумной пробке, или находясь на многолюдной улице, где нет возможности разговаривать по телефону, востребованным будет именно чат.

Таким образом, можно сформировать следующий список функций разрабатываемого приложения:

- 1) просмотр предлагаемых предложений и акций;
- 2) предоставление структурированных списков организаций города Краснодара;
- 3) поиск организаций;
- 4) просмотр подробной информации по каждой организации;
- 5) сохранение полной базы организаций на устройстве клиента;
- 6) сохранение списка организаций отдельных категорий на устройстве клиента;
- 7) регистрация и авторизация пользователя;
- 8) создание чата с организацией;
- 9) возможность просматривать и оставлять отзывы;
- 10) добавление своей организации;
- 11) возможность сообщить о существовании новой организации;
- 12) возможность оставить заявку владельца на передачу профиля организации пользователю;
- 13) редактирование профиля пользователя и организации;
- 14) режимы модератора для работы с заявками;

Перед разработкой приложения с заданным функционалом необходимо определиться с форматом передаваемых данных, изучить проблематику систем реального времени и подходы к реализации таких систем в мобильных приложениях.

2 Клиент-серверное взаимодействие

2.1 Архитектура клиент-сервер

Как правило, устройства и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам. Устройство (или программу), управляющее ресурсом, называют сервером этого ресурса. Клиент и сервер какого-либо ресурса могут находиться как на одном компьютере, так и на различных компьютерах, связанных сетью.

Появление архитектуры клиент-сервер, как и многих других новых технологий, сопровождалось рождением новой терминологии:

–Прикладной программный интерфейс (Application Programming Interface, API) – это набор функций и подпрограмм, обеспечивающих взаимодействие клиентов и серверов;

–Клиент – объект, запрашивающий информацию по сети. Как правило, это персональный компьютер или рабочая станция, запрашивающая информацию у сервера;

–Промежуточное программное обеспечение – набор драйверов, прикладных программных интерфейсов и прочего программного обеспечения, позволяющего улучшить взаимодействие между клиентским приложением и сервером;

–Реляционная база данных – база данных, в которой доступ к информации ограничен выбором строк, удовлетворяющих определенным критериям поиска;

– Сервер – компьютер (как правило, высокопроизводительная рабочая станция, мини- компьютер или мэйнфрейм), хранящий информацию, с которой работают сетевые клиенты;

–Язык структурированных запросов (Structured Query Language, SQL) – язык для создания, управления и изменения баз данных, разработанный корпорацией IBM и стандартизованный институтом ANSI;

–Архитектура «клиент-сервер» – сетевое окружение, в котором управление данными осуществляется на серверном узле, а другим узлам предоставляется доступ к данным;

–Вычислительная модель клиент-сервер – совместная с клиентом обработка запросов клиента сервером и возвращение результатов клиенту. В этой модели обработка данных приложением распределена (не обязательно поровну) между клиентом и сервером. Обработка данных инициируется и частично управляется клиентом, но не в режиме «ведущий-ведомый», а в режиме сотрудничества.

Архитектура «клиент – сервер» используется для организации сетевых приложений. Далее будет рассмотрена структура сетевого приложения.

В рамках многоуровневого представления вычислительных систем можно выделить три группы функций, ориентированных на решение различных подзадач:

1) функции ввода и отображения данных (обеспечивают взаимодействие с пользователем);

2) прикладные функции, характерные для данной предметной области;

3) функции управления ресурсами (файловой системой, базой данных и т.д.).

Выполнение этих функций в основном обеспечивается программными средствами, которые можно представить в виде взаимосвязанных компонентов (Рисунок 1), где:

1) компонент представления отвечает за пользовательский интерфейс;

2) прикладной компонент реализует алгоритм решения конкретной задачи;

3) компонент управления ресурсом обеспечивает доступ к необходимым ресурсам.

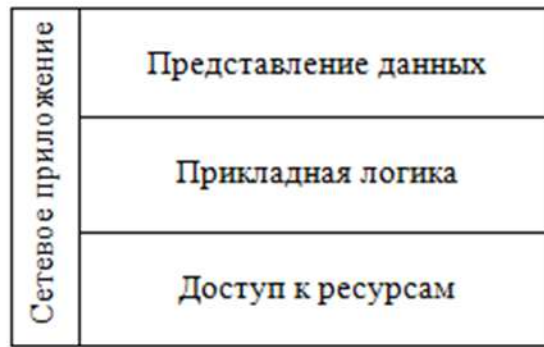


Рисунок 1 – Компоненты сетевого приложения

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты, потребители этих функций. Практические реализации такой архитектуры называются клиент-серверными технологиями. Каждая технология определяет собственные или использует имеющиеся правила взаимодействия между клиентом и сервером, которые называются протоколом обмена (протоколом взаимодействия).

Схема взаимодействия клиентского приложения и сервера представлена на рисунке 2:

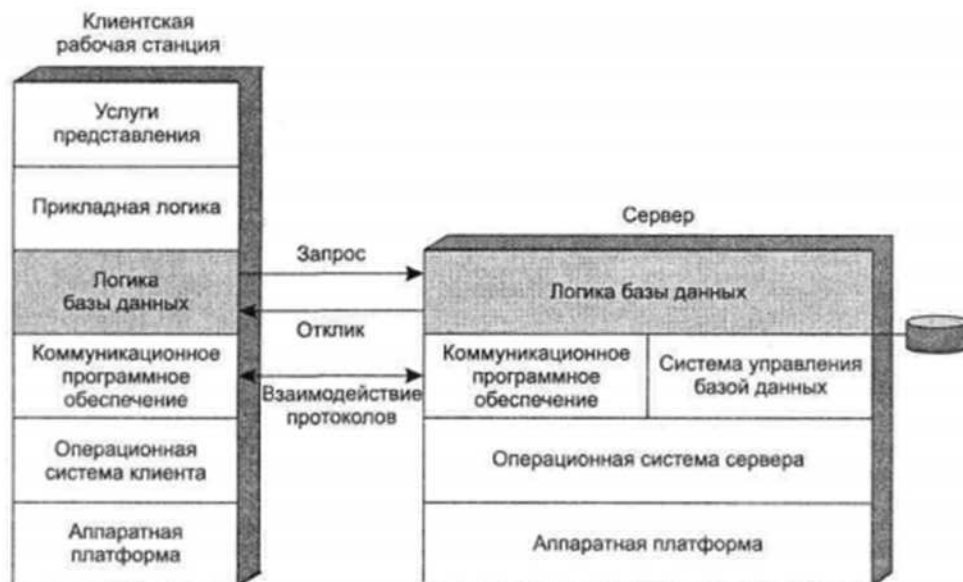


Рисунок 2 – Взаимодействие «клиент – сервер»

Существует два основных типа клиент-серверной архитектуры: двухзвенная («two-tier», «2-tier») и трехзвенная («three-tier», «3-tier»). Двухзвенная архитектура получила свое название из-за необходимости распределения трех базовых компонентов между двумя узлами (клиентом и сервером). Такая архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы самостоятельно и в полном объеме, при этом используя только собственные ресурсы, не взаимодействуя со сторонними сетевыми приложениями. Отличие трехзвенной архитектуры связано с использованием распределенных вычислений, которые реализуются на основе модели сервера приложений, разделенного на части, каждая из которых может выполняться на отдельном компьютере. Как правило, третьим звеном в такой архитектуре становится сервер приложений, реализующий в себе основную бизнес-логику системы и предоставляющий клиентским приложениям некоторый функционал [3].

2.2 Способы и форматы передачи данных

Для передачи данных между клиентом и веб-сервером используется HTTP-протокол – протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных). HTTP используется также в качестве «транспорта» для других протоколов прикладного уровня, таких как SOAP.

В ходе работы было выделено два возможных подхода к организации клиент-серверного взаимодействия – это SOAP и REST-подходы.

SOAP активно использует XML для кодирования запросов и ответов, а также строгую типизацию данных, гарантирующую их целостность при передаче между клиентом и сервером, используя для этого формат XML документа. XML (Extensible Markup Language) –расширяемый язык разметки. Предназначен для разметки текста при помощи обрамляющих тегов, что

разбивает его на уникально идентифицируемые, сортируемые фрагменты, что позволяет качественно обрабатывать документ программными методами [4].

REST предоставляет большую свободу действий, поскольку запросы и ответы могут передаваться в ASCII, XML, JSON или любых других форматах, распознаваемых одновременно и клиентом, и сервером. Кроме того, в модели REST отсутствуют встроенные требования к типизации данных. В результате пакеты запросов и ответов в REST имеют намного меньшие размеры, чем соответствующие им пакеты SOAP.

JSON (JavaScript Object Notation) – это простой формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Он основан на подмножестве языка программирования JavaScript, определенного в стандарте ECMA-262 3rd Edition - December 1999. JSON - текстовый формат, полностью независимый от языка реализации [5]. Этот формат основан на двух структурах данных:

–коллекция пар ключ-значение (в разных языках, эта концепция реализована как объект, запись, структура, словарь, именованный список или ассоциативный массив);

–упорядоченный список значений (в большинстве языков это реализовано как массив, вектор, список или последовательность).

В модели SOAP уровень передачи данных протокола HTTP является «пассивным наблюдателем», и его роль ограничивается передачей запросов SOAP от клиента серверу с использованием метода POST. Детали сервисного запроса, такие как имя удаленной процедуры и входные аргументы, кодируются в теле запроса. Архитектура REST, напротив, рассматривает уровень передачи данных HTTP как активного участника взаимодействия, используя существующие методы HTTP, такие как GET, POST, PUT и DELETE, для обозначения типа запрашиваемого сервиса. Поэтому запросы REST в общем случае более просты для формулирования и понимания, так как они используют существующие и хорошо понятные интерфейсы HTTP.

Модель SOAP поддерживает определенную степень интроспекции, позволяя разработчикам сервиса описывать его API в файле формата Web Service Description Language (WSDL, язык описания веб-сервисов). Клиенты SOAP могут автоматически получать из этих файлов подробную информацию об именах и сигнатурах методов, типах входных и выходных данных и возвращаемых значениях. С другой стороны, модель REST избегает сложностей WSDL в угоду более интуитивно понятному интерфейсу, основанному на стандартных методах HTTP, описанных выше.

В основе REST лежит концепция ресурсов, в то время как SOAP использует интерфейсы, основанные на объектах и методах. Интерфейс SOAP может содержать практически неограниченное количество методов, интерфейс REST, напротив, ограничен четырьмя возможными операциями, соответствующими четырем методам HTTP.

Необходимо отметить, что REST является архитектурным стилем, в то время как SOAP является протоколом. Активное использование формата XML протоколом SOAP и указание типизации в тексте запросов порождает увеличение данных, передаваемых по сети, что делает его менее востребованным в сфере разработки мобильных приложений в связи с повышенным использованием мобильного интернет-трафика.

2.3 Системы и приложения реального времени

Система реального времени (СРВ) – это система, которая должна реагировать на события во внешней по отношению к системе среде или воздействовать на среду в рамках требуемых временных ограничений. В такой системе важно время получения результата, другими словами время обработки информации системой не должно превышать некоторого фиксированного значения.

СРВ делятся на системы мягкого и жесткого реального времени, последние из которых в большинстве русскоязычной литературы называют

системами с детерминированным временем. Отличаются эти СРВ последствиями невыполнения единичной операции за фиксированный ограниченный интервал времени. В системе жесткого реального времени такая задержка при выполнении операции с большой долей вероятности приведет к выходу из строя всей системы. Для мягкой СРВ возможны некоторые дополнительные издержки по отношению к средним расходам, например финансовые потери, повышенная трата ресурсов и прочее. Однако превышение заданного времени обработки запроса в системе мягкого реального времени не приводит к серьезным последствиям. Таким образом, жесткие СРВ должны обеспечивать работу с заданными временными ограничениями на выполнение единичной операции в самом худшем случае, в то время как мягкие СРВ обеспечивают такую работу в среднем.

На сегодняшний день оба вида систем реального времени получили широкое распространение. Многие конвейерные роботизированные линии работают в логике жестких СРВ. Так, с момента получения сигнала от датчиков у робота есть очень короткий интервал времени для действия. Задержка в обработке данных датчика в данном случае может привести к остановке всей линии. Примером применения мягкой СРВ может послужить финансовая организация, которой приходится иметь дело с миллионом транзакций в секунду. Каждая задержка может привести к потере денежных средств, при этом задержки допустимы, но их количество должно быть по возможности минимальным.

Рядом с системами реального времени тесно стоит определение операционной системы реального времени. Операционные системы реального времени (ОСРВ) предназначены для обеспечения интерфейса к ресурсам критических по времени систем реального времени [6]. Основной задачей в таких системах является своевременность выполнения обработки данных. Такие операционные системы предоставляют разработчику расширенный функционал многопоточной разработки. ОСРВ не являются

привычными операционными системами, они представляют собой прикладной пакет, встраиваемый в проект разработчика.

Приложение реального времени – это система реального времени, особенностью которой является обеспечение приемлемого времени передачи данных между разными клиентскими устройствами.

Общие требования к ОСРВ следующие: ОС должна быть многозадачной и допускающей вытеснение, обладать понятием приоритета для потоков, поддерживать предсказуемые механизмы синхронизации, обеспечивать механизм наследования приоритетов, поведение ОС должно быть известным и предсказуемым, это значит, что должно быть определено максимальное время отклика во всех сценариях рабочей нагрузки системы [6]. Из вышесказанного следует, что операционные системы реального времени предоставляют гибкие готовые инструменты для параллельного программирования, что делает их очень востребованными при разработке жестких систем реального времени. Для мягких СРВ это не всегда так. Иногда включение ОСРВ в проект мягкой системы реального времени не оправдано.

Применение операционных систем реального времени вносит дополнительную сложность в разработку проекта. Все многопоточные приложения сложно отлаживать, а логика организации параллельных операций сильно усложняется. Так же накладываются ограничения на использование только тех языков программирования, которые позволяют работать с данной ОСРВ. Таким образом, включение в проект операционной системы реального времени должно быть обоснованным.

Для принятия решения о внедрении в разработку приложения ОСРВ требуется оценить необходимость такого решения. Во-первых, необходимо определить насколько будет нагружена разрабатываемая система, оценить максимальное количество пользователей. Во-вторых, оценить вероятность появления пиковых нагрузок на приложение, а именно часы, когда пользователи будут проявлять наибольшую активность в приложении. В-

третьих, проанализировать объем обработки данных при каждой операции, чем он больше, тем выше вероятность переполнения очереди операций в однопоточном сервере приложений.

В тех случаях, когда количество пользователей ограничено в обозримом будущем, пиковые нагрузки не будут явно выражены, а траты времени на обработку данных каждой операции минимальны, внедрение операционной системы реального времени будет являться избыточным.

Для данной дипломной работы интерес представляет приложение для взаимодействия пользователей посредством сообщений в режиме реального времени. Когда между двумя пользователями ведется переписка, они хотят постоянно получать актуальную информацию, то есть приложение должно обеспечить передачу сообщений с наименьшей возможной задержкой. Однако очень редкие задержки в пределах нескольких секунд для пользователя будут приемлемыми, поэтому данная система реального времени будет являться мягкой. Если разработчик не ожидает большого числа пользователей данного приложения и уверен, что нагрузка на приложение будет равномерно распределена в течение всего дня, то он может обойтись однопоточными вариантами организации серверной части системы, избежав излишних сложностей работы с ОСРВ.

2.4 Особенности реализации мобильных приложений реального времени

В разработке мобильных приложений получающих некоторые данные по сети обычно используется подход, при котором клиент запрашивает нужные данные, а затем получает ответ. То есть помимо клиентского мобильного приложения имеется сервер приложения. Такой сервер принимает и обрабатывает запросы, получает все необходимые данные, обращаясь к серверу баз данных, файловой системе и т.д., формирует ответ и посылает его клиенту.

Такое решение отлично подходит для редко изменяемых данных и в тех случаях, когда актуальность запрашиваемой информации не критична.

Однако мобильные приложения реального времени должны своевременно реагировать на изменения в состоянии сервера: получать новые данные, обновлять измененные данные и т.д.

Самым очевидным решением данной проблемы может стать написание так называемого сервиса (выделенного отдельного процесса в памяти устройства) для мобильного приложения, который будет с некоторым заданным интервалом времени опрашивать сервер на предмет обновленных данных. В случае получения положительного ответа сервис будет информировать приложение. Самыми главными недостатками такого решения являются повышенная трата мобильного трафика и заряда аккумуляторной батареи устройства, излишняя нагрузка на сервер приложения, порционное получение обновлений данных от сервера.

Другой подход к проверке обновлений данных на сервере клиентским приложением – обновление данных при следующем обращении клиента к серверу, что получило название «Long pulling». Очевидно такой подход так же не подходит для приложений реального времени.

Самым подходящим решением данной проблемы будет использование технологии WebSocket [7]. WebSocket – это протокол полнодуплексной (двусторонней) связи, располагающийся над TCP-соединением и предназначенный для обмена сообщениями между клиентским приложением и сервером в режиме реального времени. Используя указанную технологию, клиентское приложение один раз открывает соединение, после чего сервер становится равноправным участником обмена сообщениями, то есть при изменении данных он в любой момент может посылать сообщения клиентскому приложению, используя ранее открытое и постоянно поддерживаемое соединение. Схема взаимодействия клиентского приложения с сервером по технологии WebSocket изображена на рисунке 3.

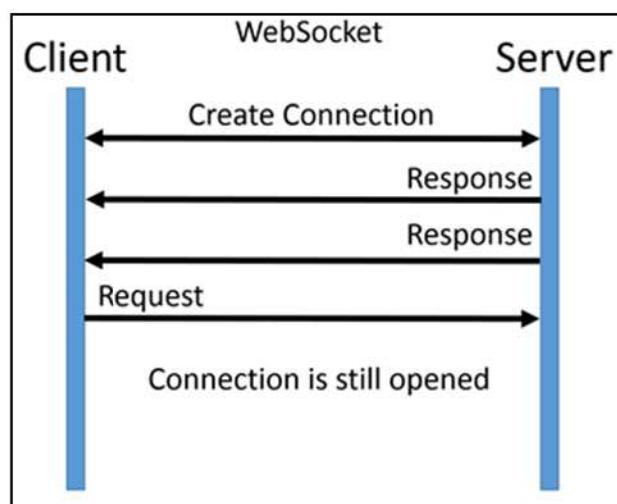


Рисунок 3 – Организация процесса передачи данных через WebSocket

Протокол WebSocket работает над TCP-соединением – это означает, что при попытке установки соединения клиентское приложение отправляет по HTTP специальные заголовки, целью которых является подтверждение, что сервер поддерживает WebSocket соединение [8]. Если это так, то дальнейшее взаимодействие сторон происходит по протоколу WebSocket, который не имеет ничего общего с протоколом HTTP.

В протокол WebSocket встроена проверка связи при помощи управляющих фреймов типа PING и PONG [8]. Сторона, желающая проверить состояние соединения, отправляет фрейм PING с произвольным телом. Противоположная сторона в этом случае должна в приемлемое время ответить на данный запрос фреймом PONG с тем же телом, что будет подтверждением того, что с соединением все в порядке. Для корректного завершения сеанса обмена сообщениями одна из сторон посылает фрейм закрывающий соединение. В теле данного фрейма содержится причина закрытия соединения.

На практике для использования данной технологии существуют библиотеки, предоставляющие простой интерфейс. Они позволяют вести разработку на более высоком уровне программирования.

Таким образом, использование технологии WebSocket для мобильных приложений реального времени является лучшим решением. Однако поддержание постоянно открытого соединения не является оптимальным. Такое решение приведет к повышенной трате ресурсов устройства, к тому же оно сложно реализуемо с программной точки зрения. Когда пользователь закрывает мобильное приложение с функциями системы реального времени, он рассчитывает, что приложение продолжит работать в фоне и уведомит его о существующих изменениях. Чтобы реализовать такой функционал только с помощью технологии WebSocket требуется написание существующего отдельно от приложения фонового сервиса, постоянно поддерживающего соединение с сервером.

Для уведомления пользователя о важных событиях в приложении даже когда оно находится в закрытом состоянии, существует сервис Firebase Cloud Messaging (FCM) – это решение которое облегчает обмен сообщениями между мобильными и серверными приложениями [9]. Чтобы использовать данный сервис разработчику необходимо зарегистрировать свое мобильное приложение и реализовать требуемый программный интерфейс в коде своего приложения. Сервис позволяет производить полную и выборочную рассылку push-уведомлений как из личного кабинета, так и через программный интерфейс приложения (англ. application programming interface, API). Сервис присваивает каждому клиентскому устройству уникальный регистрационный ключ, а так же предоставляет уникальный ключ сервера. Эти данные позволяют, используя API сервиса, отправлять уведомления на конкретные устройства пользователей. Push-уведомления – это небольшие всплывающие окна на экране мобильного устройства. Они могут появляться на экране любого устройства, где есть область оповещений, или есть возможность вывода на экран данных принятых из сети Интернет [10]. Такие уведомления позволяют вовремя уведомить пользователя о происходящих изменениях, если приложение находится в закрытом состоянии.

Так же сервис собирает некоторые статистические данные по использованию приложения и предоставляет наглядные графики. На рисунке 4 приведен результат сбора статистики, касаемо частоты посещения тех или иных страниц приложения.

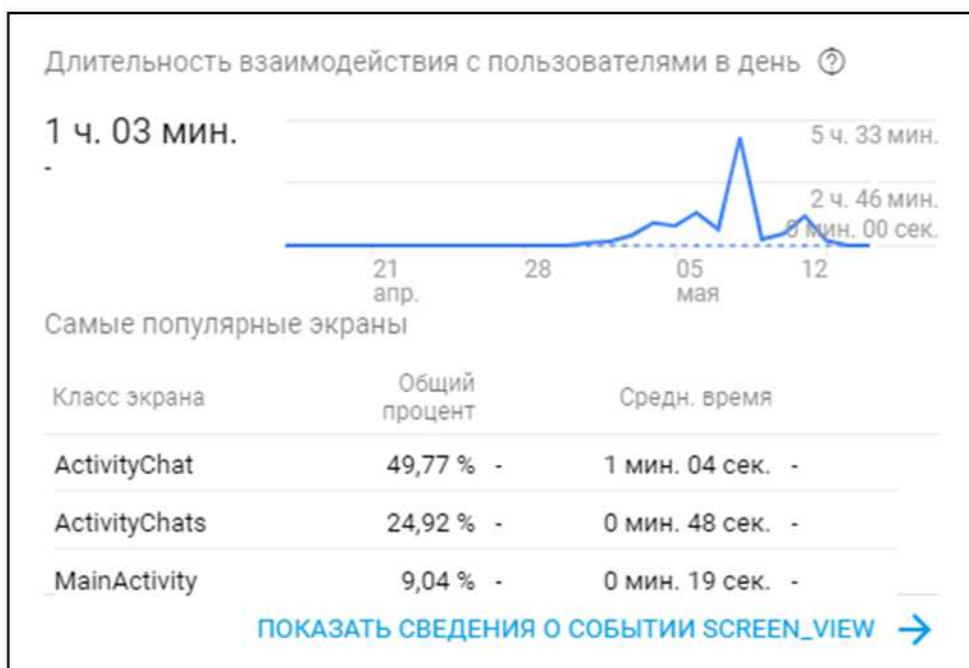


Рисунок 4 – Статистика частоты посещения страниц приложения

Таким образом, наилучшими решениями в организации мобильного приложения реального времени будет:

- 1) использование привычных HTTP-запросов для получения архивных (не изменяемых) данных.
- 2) использование протокола WebSocket для отслеживания изменений данных в режиме реального времени;
- 3) удержание открытого соединения лишь на некоторых страницах приложения для экономии ресурсов;
- 4) использование сервиса Firebase Cloud Messaging для уведомления об изменении данных на сервере в случае закрытого WebSocket-соединения.

3 Реализация приложения

3.1 Средства разработки

Согласно последним статистическим данным наиболее распространенной операционной системой для мобильных устройств в России является Android [11]. Исходя из этого для разработки мобильного приложения «Организации Краснодара» была выбрана ОС Android с официальной средой разработки Android Studio. Разработка велась на языке программирования Java. Шаблоны страниц android приложений создаются автоматически при работе с графическим редактором среды. Макет изображения задается программным кодом на языке XML и генерируется автоматически. Однако для более гибкой работы со стилями визуальных объектов приложения xml-код дорабатывается вручную.

В ходе работы была проведена оценка возможных нагрузок на приложение. Так как область использования приложения ограничивается городом Краснодаром, то можно говорить о том, что непредвиденно большой рост количества пользователей данного приложения невозможен. Приложение «Организации Краснодара» должно предоставлять справочную информацию об организациях города, поддерживая возможность переписки с представителями организаций в режиме реального времени. На основании этого можно сделать вывод, что основная нагрузка на приложение будет ложиться в светлое время суток. Специфика приложения позволяет рассчитывать на отсутствие явно выраженных пиковых нагрузок. С учетом всего вышесказанного был выбран достаточный инструментарий разработки, перечисляемый ниже по тексту работы.

В качестве хранилища всех глобальных данных приложения была выбрана реляционная СУБД MySQL – открытая распространённая СУБД, предоставляемая на любых хостингах. Для разработки использовался

локальный сервер из так называемого WAMP комплекса под названием Open Server. Он содержит следующие инструменты:

- 1) web-сервер Apache;
- 2) СУБД MySQL;
- 3) интерпретатор серверного языка программирования PHP.

Для реализации серверной части был выбран скриптовый язык PHP как самый распространенный и простой. Для передачи данных между мобильным приложением и сервером был выбран аналог REST-подхода с форматом обмена данными JSON, что позволяет экономить мобильный трафик.

Для хранения информации на устройстве клиента была выбрана компактная встраиваемая реляционная база данных SQLite. В Java имеются встроенные библиотеки для работы с этой базой данных, что значительно упрощает задачу хранения данных на клиентском устройстве. В качестве протокола обмена данными используются вызовы функций (API) соответствующих библиотек. База данных SQLite позволяет сохранять часть данных с сервера на устройстве клиента и позволяет использовать приложение без доступа в Интернет.

Для реализации взаимодействия приложения с сервером в режиме реального времени в обе стороны была использована специальная библиотека Ratchet. Ratchet - это PHP-библиотека, предоставляющая разработчикам инструменты для создания двунаправленных приложений в реальном времени между клиентами и серверами с помощью технологии WebSocket [12].

При разработке серверной части велось активное использование документации по языку PHP [13]. Большую помощь при написании мобильного приложения оказали официальная документация по языку Java [13] и главный ресурс android-разработчиков – Android Developers [14]. Базовые идеи и понятия языка Java были изучены по книгам «Java 8: руководство для начинающих» [16] и «Философия Java» [17].

3.2 Разделение сервера приложения на два самостоятельных сервера

В приложении явным образом прослеживается разделение операций получения данных на два вида: получение редко изменяемых справочных данных об организациях, неизменных ранее отосланных сообщений чатов и т.д. и динамичное получение обновлений на странице чатов и сообщений.

Было принято решение разделить сервер приложения на два самостоятельных сервера. Первый будем называть сервером данных, он работает с HTTP-запросами и возвращает редко обновляемые данные. Второй сервер имеет свое подключение к базе данных и работает с соединениями по протоколу WebSocket, его будем называть сервером чата. Сервер чата предназначен исключительно для обмена сообщениями и рассылки уведомлений. Такое решение повышает отказоустойчивость всей системы, так как позволяет сохранить рабочим часть функционала приложения при выходе из строя одного из серверов.

Для динамического обновления данных на странице чатов приложения устанавливается соединение с сервером чата. При этом мобильное приложение посылает на сервер регистрационные данные, позволяющие уникальным образом идентифицировать пользователя, который при этом добавляется в списки активных подключений. Таким образом, при получении сообщений от одного из подключенных пользователей сервер чата сохраняет обновления в базу данных, там же находит пользователей, которым нужно сообщить об изменениях. Далее происходит поиск открытых подключений с этими пользователями и отправка данных. Если имеются пользователи, которые по некоторым причинам не имеют открытого соединения с сервером чата в данный момент, то происходит рассылка push-уведомлений через сервис Firebase Cloud Messaging. Данный сервис назначает устройству уникальный идентификатор, который устройство передает серверу при входе

в профиль пользователя. Таким образом сервер чата может отправлять push-уведомления на конкретные устройства.

3.3 Пользовательский интерфейс приложения

Рассмотрим интерфейс пользователя в разработанном городском мобильном приложении «Организации Краснодара».

После запуска приложения пользователем, на экране устройства отображается окно новостной ленты (рисунок 5). Здесь представлены карточки акций и новостей, выбранные в соответствии с разделом и категорией в верхней части окна, оформленные в виде выпадающего списка. При нажатии на карточку происходит переход к соответствующей организации. Меню предоставляет возможность авторизации и перехода к организациям.

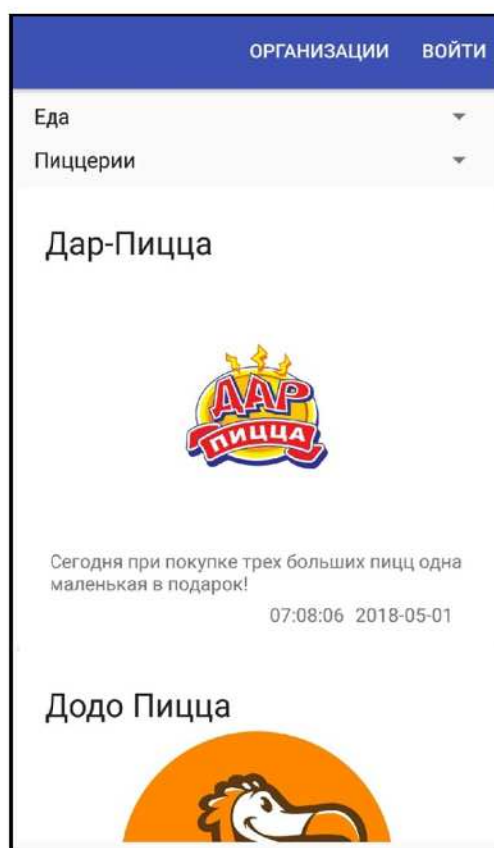


Рисунок 5 – Новостная лента приложения

На рисунке 6 изображено окно авторизации. В данном окне необходимо ввести данные, указанные при регистрации в приложении. Если пользователь оставляет выбранным пункт «Сохранять данные профиля», то приложение в последующем не будет требовать авторизации пользователя при каждом запуске, запомнив текущего пользователя. Принудительный выход из профиля сбрасывает данную настройку.

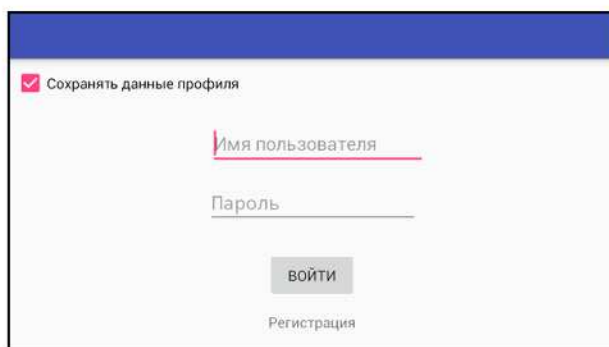


Рисунок 6 – Окно авторизации

Если же у пользователя еще нет аккаунта, то перейти к регистрации можно нажав на ссылку «Регистрация».

В окне регистрации, представленном на рисунке 7, пользователь, руководствуясь подсказками, должен ввести все необходимые данные.

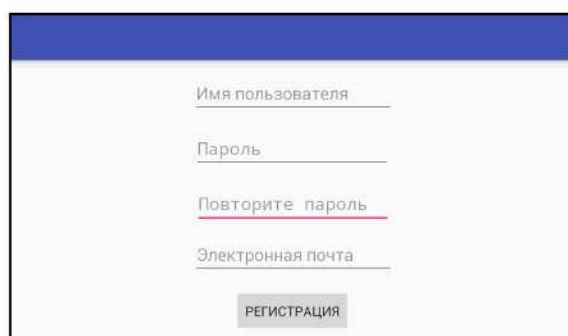


Рисунок 7 – Окно регистрации

Здесь организована проверка совпадения паролей. Если пароли совпадают, то текст контрольного (второго) поля пароля выделяется

зеленым, в противном случае красным. Для повышения безопасности в базе данных хранятся и сравниваются не пароли, а их хеш-коды.

При попытке регистрации без указания некоторых полей, подсказки пустых полей выделяются красным цветом, сигнализируя пользователю, что данные поля обязательны для заполнения. Адрес электронной почты проходит простую проверку корректности его записи. После успешной регистрации приложение вернет пользователя в окно авторизации. После авторизации пользователь может перейти в свой профиль из основных окон приложения, выбрав в меню соответствующий пункт.

На рисунке 8 представлен профиль пользователя.



Рисунок 8 – Профиль пользователя

Здесь отображаются все данные профиля и изображение, перечислен список организаций пользователя. Если список не пустой, то пользователь является владельцем указанных в нем организаций. Это означает, что ему предоставляется возможность редактирования данных указанных организаций в профиле организации, в который можно перейти, выбрав в списке соответствующую позицию. При этом все ранее определенные данные будут автоматически внесены в форму редактирования организации.

Может существовать не один пользователь, с доступом к редактированию данных компании.

Устроенная аналогичным образом форма появляется на экране при нажатии по ссылкам «Добавить свою организацию» и «Сообщить об организации».

Выбрав пункт меню «Сменить пароль» пользователь попадает в окно смены пароля, где может, введя корректный старый пароль, указать новый и подтвердить изменения. Нажатие пункта «Сообщения» откроет страницу с чатами обращений, которая изображена на рисунке 16. Нажатие на изображение пользователя запустит диалог загрузки нового изображения для профиля.

Если в профиле пользователя или фирмы произошли изменения полей, то при переходе к любому другому окну на экране появится диалоговое окно подтверждения изменений. Пользователю при этом будет предложено подтвердить изменения и покинуть окно, отменить изменения, либо же отменить действие и остаться в текущем окне.

На рисунке 9 изображено окно профиля организации.

The screenshot shows a form titled "Добавить новость или акцию" (Add news or promotion) with a sub-section "Текущие новости" (Current news). The form contains the following fields and options:

- Text: "Вы можете изменять информацию ниже" (You can change the information below).
- Checkbox: "Принимать обращения клиентов в приложении" (Accept client requests in the app).
- Text: "Настроить менеджеров по работе с клиентами" (Set up managers for client work).
- Dropdown: "Раздел: Еда" (Section: Food).
- Dropdown: "Категория: Пиццерии" (Category: Pizzerias).
- Text: "Наименование организации: Дар-Пицца" (Organization name: Dar-Pizza).
- Text: "Адрес: Чекистов проспект, 42/1" (Address: Chekistov Prospekt, 42/1).
- Text: "Режим работы: пн-вс: 10:00-23:00" (Operating hours: Mon-Sun: 10:00-23:00).
- Text: "Электронный ресурс: darpizza.com" (Electronic resource: darpizza.com).
- Text: "Номера телефонов: +79182536657" (Phone numbers: +79182536657).
- Text: "Описание: «Дар Пицца» - одни из первых пиццерий Краснодара. Приходите в «Дар Пиццу»!" (Description: "Dar Pizza" - one of the first pizzerias in Krasnodar. Come to "Dar Pizza"!).
- Text: "Здесь вы можете оставить заметку по данной организации. Она будет видна только вам" (Here you can leave a note about this organization. It will only be visible to you).
- Image: A small photo of a pizza restaurant interior.
- Image: The "ДАР ПИЦЦА" logo.
- Button: "ДОБАВИТЬ ФОТОГРАФИЮ" (Add photo).

Рисунок 9 – Профиль организации

В данном окне реализовано динамическое создание полей для ввода телефонных номеров. Для добавления нового поля необходимо нажать по строке «Добавить поле». Список фотографий оформлен в виде горизонтальной ленты прокрутки. Кнопка добавить фотографию запускает диалог выбора изображений из памяти устройства. Длительное нажатие на изображение в полосе прокрутки удаляет его. Так же редактированию поддаются все текстовые поля. Если владелец организации указывает пункт «Принимать обращения клиентов в приложении», то всем пользователям будет открыта возможность создания чата с этой организацией по заданной теме обращения. Отвечать на обращения пользователя должны менеджеры по работе с клиентами, выбрать которых можно перейдя по ссылке «Настроить менеджеров по работе с клиентами». За операции с менеджерами организации отвечает окно приложения, изображенное на рисунке 10. Осуществляется поиск пользователей на должность менеджера по имени, возможно добавление и удаление пользователей из списка менеджеров, что сопровождается рассылкой соответствующих push-уведомлений.

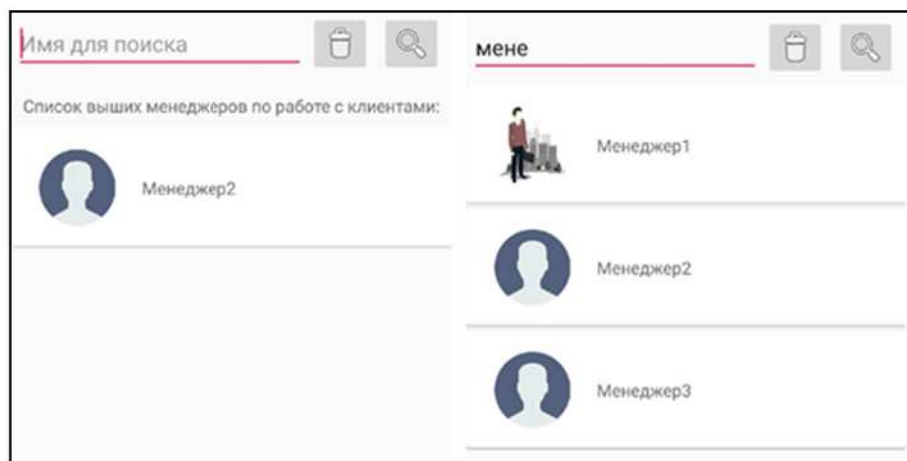


Рисунок 10 – Окно выбора менеджеров

Выбор ссылки «Добавить новость или акцию» запускает окно создания новости. В этом окне можно редактировать фотографии организации и

прикреплять текстовую информацию. После нажатия кнопки «Опубликовать» новость появится в новостной ленте.

Ссылка «Текущие новости» переводит владельца организации к списку активных новостей, представленных теми же карточками, что и в ленте новостей. При длительном нажатии на карточку появляется диалоговое окно удаления новости.

Все основные окна приложения снабжены меню, в каждом из которых есть пункты «Организации» и «Новости» для быстрой навигации в приложении. Чтобы перейти к просмотру организаций города пользователю необходимо выбрать пункт меню «Организации». В этом случае будет запущено окно, представленное на рисунке 11.

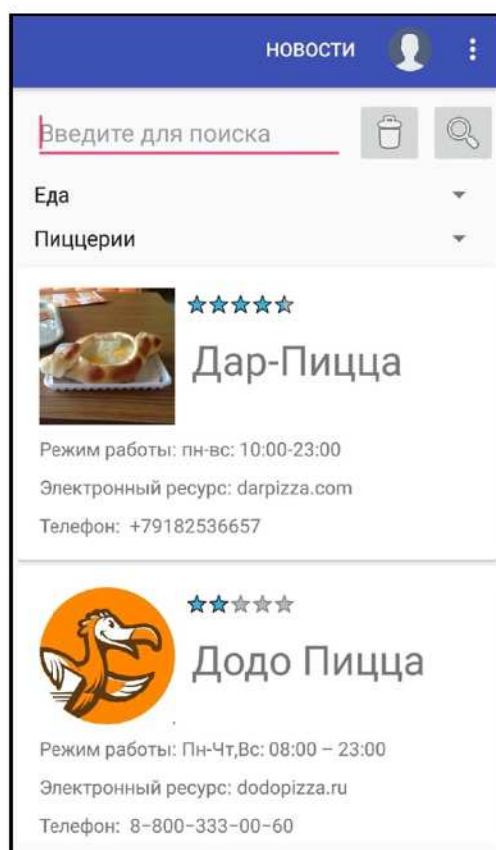


Рисунок 11 – Список организаций

Аналогично окну новостей осуществляется выбор категории, после чего в списке отображаются результаты поиска. Так же здесь реализована

возможность текстового поиска организаций. В меню имеется пункт «Скачать справочник», по нажатию на который появляется диалоговое окно подтверждения операции. При положительном ответе происходит загрузка полной базы данных на мобильное устройство, при этом до окончания загрузки приложение выводит уведомление о фоновом процессе загрузки, как это показано на рисунке 12. После выполнения данной операции пользователь сможет работать со списками организаций без доступа к сети. Для скачивания отдельной категории организаций пользователю необходимо выбрать пункт меню «Скачать категорию».

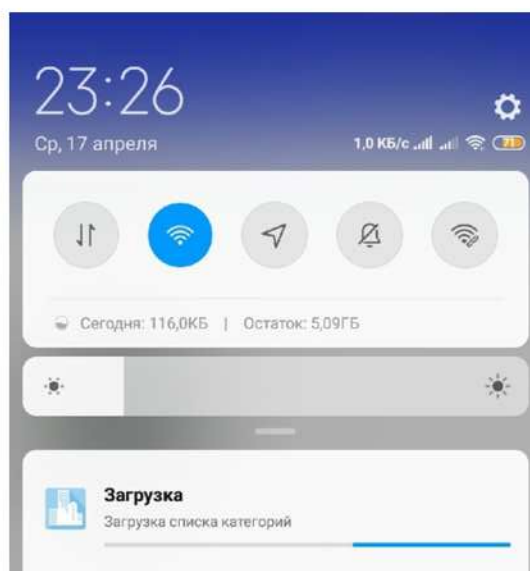


Рисунок 12 – Уведомление о процессе загрузки

Список организаций реализован таким образом, что загрузка информации происходит постепенно, по мере пролистывания списка. Такое решение позволяет экономить трафик на мобильном устройстве. Подобный подход применяется во всех крупных списках приложения. Список состоит из карточек с информацией. Нажатие на номер или электронный ресурс, указанные на карточках, запускает приложение «Телефон» или браузер соответственно. Короткое нажатие по карточке открывает окно с более подробной информацией по данной организации.

Это окно изображено на рисунке 13. Здесь так же организованы ссылки по номерам телефонов и электронных ресурсов. Фотографии фирмы располагаются, как и в профиле фирмы, в горизонтальной ленте прокрутки. Рейтинг фирмы вычисляется на основании оценок пользователей и отображается в виде закрашиваемых звездочек. Ссылка «Отзывы» открывает окно отзывов по данной организации. При нажатии на ссылку «я владелец» открывается диалог, позволяющий пользователю оставить контактный номер телефона для связи с менеджером с целью подтверждения причастности к организации.

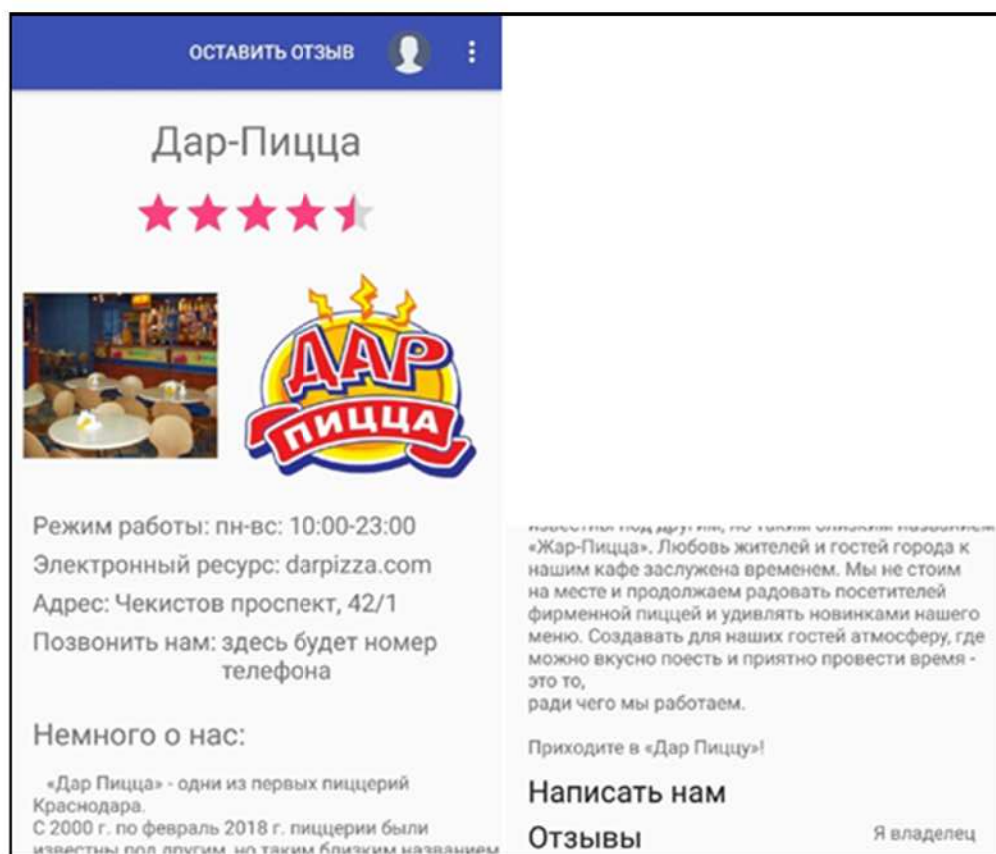


Рисунок 13 – Информация об организации

По нажатию пункта меню «оставить отзыв» открывается окно, где можно указать рейтинг и прикрепить текстовое сообщение.

В окне просмотра отзывов по выбранной организации, представленном на рисунке 14, так же имеется возможность добавления отзыва. Кроме

просмотра списка отзывов здесь реализована функция удаления своего отзыва по короткому нажатию на карточку с подтверждением действия в диалоговом окне. При нажатии на чужие отзывы никаких действий не происходит.

Если пользователь в данный момент не имеет доступа к сети Интернет, то список отзывов загружен не будет. Оставить отзыв при этом можно. Если в момент отправки отзыва приложение обнаружит, что отсутствует подключение к сети, то оно предложит пользователю сохранить отзыв в памяти устройства. При следующем запуске приложения и наличии сети сохраненные отзывы будут отправлены на сервер.

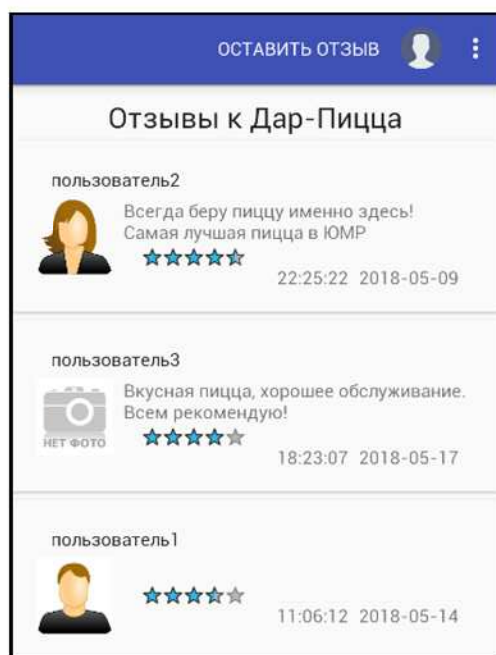


Рисунок 14 – Отзывы

В окне с информацией об организации (рисунок 13) может присутствовать ссылка «Написать нам», при нажатии на которую запускается диалог создания обращения, который представлен на рисунке 15.

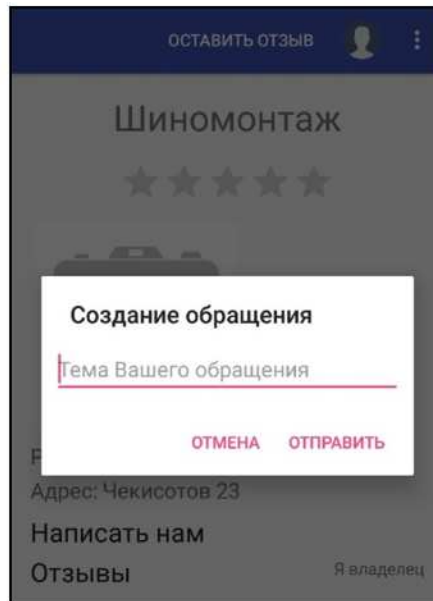


Рисунок 15 – Диалог создания обращения

В этом диалоге пользователю необходимо указать тему своего обращения и подтвердить действие. В случае успешного создания обращения будет автоматически запущено окно сообщений по данному обращению, которое изображено на рисунке 18. Также в окне обращений пользователя (рисунок 16) появится новый чат.

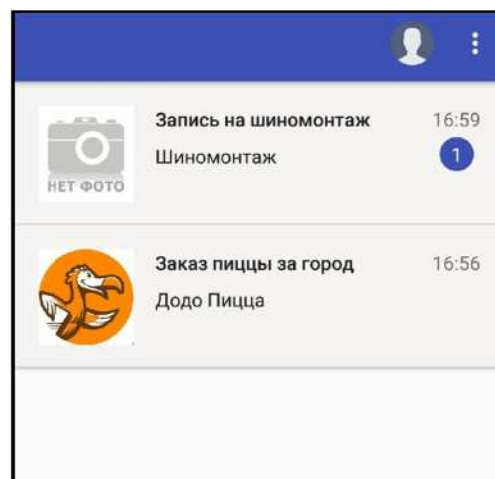


Рисунок 16 – Окно обращений пользователя

Окно обращений пользователя отображает все обращения в виде списка чатов. В том случае, когда пользователь отмечен некоторым владельцем организации как менеджер по работе с клиентами, данное окно будет иметь несколько другой вид. По умолчанию вверху списка будут отображены все обращения пользователей к организации, такие элементы списка выделяются цветом до тех пор, пока один из менеджеров не взял данное обращение в обработку, поэтому отмеченные цветом обращения видят все менеджеры одной организации. Считается, что менеджер взял обращение в обработку, если он написал в данный чат хотя бы одно сообщение. В этот момент это обращение пропадает из списков обращений других менеджеров. Так же менеджеру предоставляется возможность настройки выборки чатов по их типу посредством выбора пунктов меню (рисунок 17).

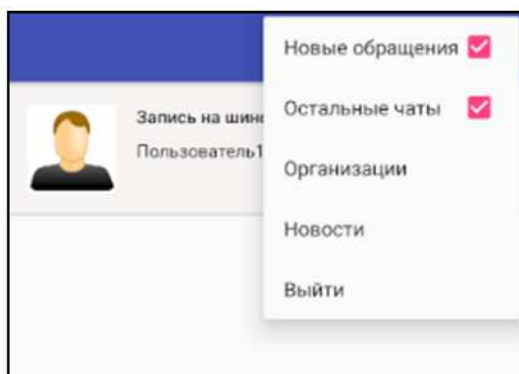


Рисунок 17 – Настройка отображаемых чатов менеджера

Будем называть создателя обращения стороной вопроса, а менеджера, взявшегося за обработку данного обращения – стороной ответа. Отображение одного и того же чата у сторон вопроса и ответа отличается, как это можно заметить на рисунках 16, 17. Тема обращения отображается у обеих сторон, однако сторона вопроса видит изображение и наименование организации, а противоположная сторона видит изображение пользователя и его имя. Такое

решение является интуитивно понятным и удобным для обеих сторон. Нажатие на чат переводит в окно сообщений данного чата (рисунок 18).

Окно обращений работает в режиме реального времени. Когда в один из чатов приходит сообщение, приложение оповещает пользователя звуковым сигналом, чат поднимается на первую строку и на нем появляется индикатор количества непрочитанных сообщений (если индикатор уже отображался, то он обновляется). Когда некоторый пользователь создает обращение к организации, то у всех менеджеров этой организации, находящихся в окне обращений, появляется новый выделенный цветом чат. Это происходит, если менеджер не сбрасывал выбор пункта меню «Новые обращения». В противном случае и в случаях, когда менеджер находится в другом окне приложения либо приложение закрыто, на устройство приходит соответствующее push-уведомление.

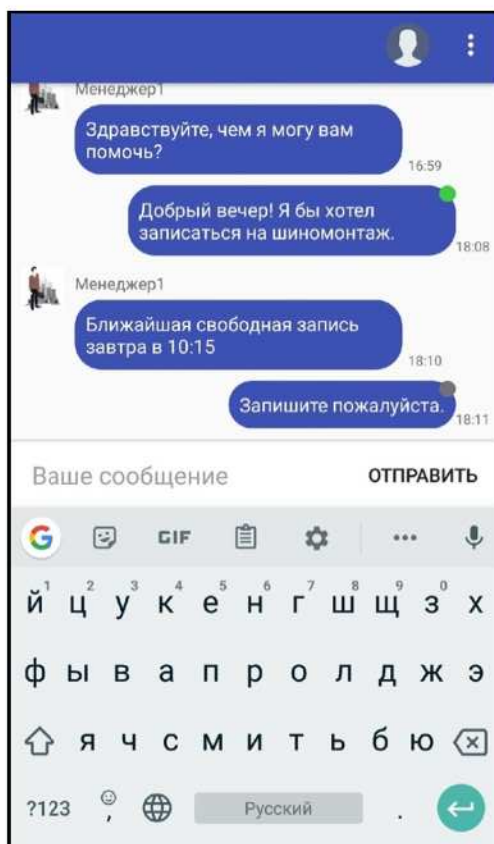


Рисунок 18 – Окно сообщений

В окне сообщений приложения отображаются все сообщения чата в хронологическом порядке от старых к новым. Исходящие сообщения расположены справа и снабжены индикатором прочтения. Когда сообщение будет прочитано получателем, цвет индикатора изменится с серого на зеленый. Индикатор не будет менять свой цвет в случае ознакомления менеджеров с обращением до принятия обращения в работу. Для входящих сообщений, расположенных слева дополнительно выводятся изображение и имя отправителя. Перед отправкой сообщение сохраняется в локальной памяти устройства, а в список отправленных сообщений первоначально добавляется сообщение с индикатором отправки, что показано на рисунке 19.

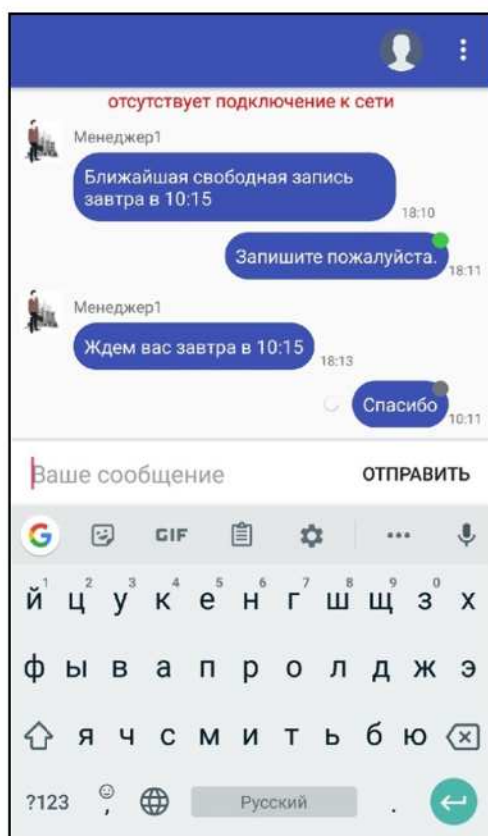


Рисунок 19 – Отправка сообщения без доступа к сети

После подтверждения факта отправки сервером чата, сообщение будет обновлено: исчезнет индикатор отправки и установится точное время приема сообщения сервером. В случае потери подключения к сети Интернет в

верхней части экрана будет выведено сообщение (рисунок 19). Аналогичным образом выводится сообщение о потере связи с сервером чата по иным причинам, при этом дополнительно всплывает текст, сообщающий пользователю, что функции чата временно недоступны. В случаях потери соединения с сервером чата запускается фоновый процесс ожидания соединения. При этом если имеются не отосланные сообщения, появляется уведомлением о процессе отправки сообщений в фоне. Сообщения будут отправлены при установлении соединения, после чего страница автоматически перезагрузится во избежание пропуска входящих сообщений. Этот же фоновый процесс запускается при проблемах с соединением в окне обращений пользователя.

Для доступа пользователя к чатам в режиме *offline* реализовано сохранение всех чатов в локальной памяти устройства, при этом количество хранящихся сообщений по каждому чату ограничено с целью экономии памяти устройства. Просмотр всех сообщений возможен только с доступом к сети Интернет.

Меню основных окон приложения содержат схожие пункты, предназначенные для быстрой навигации. Все уникальные пункты меню основных окон были оговорены отдельно, кроме случаев, когда авторизованный пользователь является модератором. При этом в профиле пользователя появляется дополнительный пункт меню «Модератор», при нажатии на который открывается окно, представленное на рисунке 20.

Выбирая пункты в верхней части окна, пользователь производит выборку карточек по типу заявки. Короткое нажатие по карточке переводит модератора в профиль организации, где он может исправить ошибки, помочь пользователю внести нужную информацию и т.п. Длинное нажатие по карточке вызывает диалоговое окно, в котором модератору предлагается подтвердить данную заявку, игнорировать, либо же отменить действие. После подтверждения заявки организация появляется у ее владельца и в списках организаций.

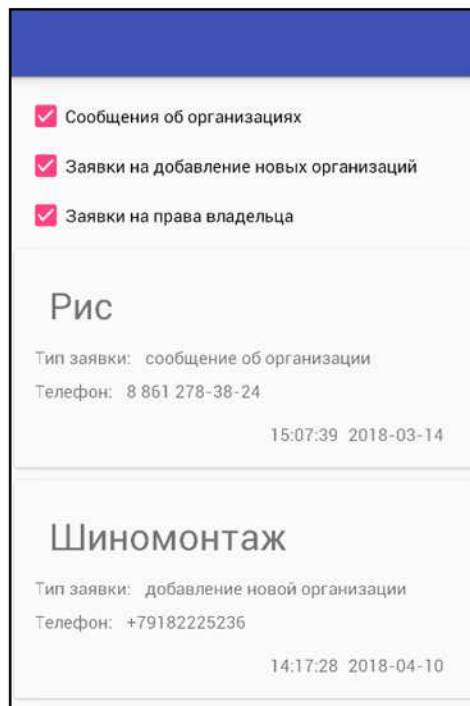


Рисунок 20 – Страница модератора

ЗАКЛЮЧЕНИЕ

Цель работы – разработка трехзвенного приложения с клиентским мобильным приложением, которое предоставляет справочные данные об организациях города Краснодара, дает возможность оставлять отзывы и оценки, содержит настраиваемую новостную ленту организаций и предоставляет возможность взаимодействия пользователей с организациями городской инфраструктуры – достигнута.

В рамках данной работы проведен анализ рынка, изучены подходы к реализации мобильных приложений реального времени, проведена оценка возможных нагрузок на приложение, разработана база данных. На основании оценки возможной нагрузки был выбран подходящий инструментарий разработки и реализовано мобильное приложение «Организации Краснодара». Данное приложение предоставляет справочные данные об организациях города Краснодара, позволяет оставлять отзывы и оценки, предоставляет настраиваемую новостную ленту организаций, а также открывает возможность взаимодействия с представителями городских организаций.

При реализации приложения применялась среда разработки Android Studio, разработка велась на языке Java. Для написания сервера был использован язык PHP. База данных клиентского приложения реализована в SQLite, сервера – MySQL. Разработанное приложение в достаточной степени реализует функционал модератора, позволяет работать над профилем организаций, публиковать новости и акции. Однако использовать данный функционал через мобильное устройство не всегда удобно. Эти функции приложения вероятнее станут вспомогательными инструментами пользователя в конечной системе. Поэтому возможным направлением дальнейшего развития системы может стать написание web-клиента, частично использующего уже готовый программный интерфейс сервера и реализующего указанный функционал.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Филия А.С. Падение эффективности ТВ-рекламы и современные технологии ее повышения / А. С. Филия // Российское предпринимательство. –2007. – Т. 6. – № 9. – С. 99–104.
- 2 Электронная энциклопедия «TAdviser» [Электронный ресурс]. – Режим доступа: http://www.tadviser.ru/index.php/Статья:Мобильные_приложения (дата обращения: 5.05.18).
- 3 Информатика, информационные технологии [Электронный ресурс].– Режим доступа: <http://csaa.ru/dva-osnovnyh-tipa-klient-servernoj-arhitektury/> (дата обращения: 21.03.19).
- 4 DeveloperWorks [Электронный ресурс].– <https://www.ibm.com/developerworks/ru/library/x-newxml/index.html> (дата обращения: 16.04.19)
- 5 Введение в JSON [Электронный ресурс]. – Режим доступа: <https://www.json.org/json-ru.html> (дата обращения: 5.05.18)
- 6 Операционные системы реального времени / И. Б. Бурдонов, А. С. Косачев, В. Н. Пономаренко // Институт системного программирования РАН, 2006. – 98с.: ил.
- 7 Android WebSocket. Обратная связь, или когда важна каждая доля секунды [Электронный ресурс]. – Режим доступа: <https://stfalcon.com/ru/blog/post/android-websocket> (дата обращения: 18.01.19).
- 8 Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/websockets> (дата обращения: 18.01.19).
- 9 Firebase Cloud Messaging [Электронный ресурс]. – Режим доступа: <https://firebase.google.com/docs/cloud-messaging> (дата обращения: 20.01.19)
- 10 PushAll [Электронный ресурс]. – Режим доступа: <https://pushall.ru/blog/whatispushnotifications> (дата обращения: 20.01.19)

11 Насколько популярна iOS в России и других странах. AppleInsider.ru [Электронный ресурс]. – Режим доступа: <https://pushall.ru/blog/whatispushnotifications> (дата обращения: 10.05.18)

12 Документация Ratchet [Электронный ресурс]. – Режим доступа: <http://socketo.me/> (дата обращения: 21.01.19)

13 Документация PHP [Электронный ресурс]. – Режим доступа: <http://php.net/manual/ru/tutorial.php> (дата обращения: 12.03.19)

14 Документация Java [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/javase/8/docs/> (дата обращения: 15.04.19)

15 Android Developers [Электронный ресурс]. – Режим доступа: <https://developer.android.com/> (дата обращения: 15.04.19)

16 Шилдт, Герберт. Ш57 Java 8: руководство для начинающих, 6-е изд. : Пер. с англ. – М. : ООО "И.Д. Вильямс", 2015. - 720 с.: ил. - Парал. тит. англ.

17 Эккель Б. Философия Java. 4-е полное изд — СПб.: Питер, 2015 — 1168 с.: ил. – (Серия «Классика computer science»).