МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Рубцовский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования
«Алтайский государственный университет»

Кафедра математики и прикладной информатики

Выпускная квалификационная работа

Тема: Разработка мобильного приложения для организации доступа к сервису VoIP коммуникации «Русский телефон»

Выпускную квалификационную работу выполнил студент 4 курса, группы 1235
Погоняев Д.А.
_____
(подпись)
Научный руководитель:
доцент, к.ф.–м.н.
Шевченко А. С.
_____
(подпись)

Допустить к защите

Зав. кафедрой
к.т.н., доцент
Жданова Е.А.
_____
(подпись)
«___» _____ 2017 г.

Выпускная квалификационная работа защищена

«___» _____ 2017 г.
Оценка _____
Председатель ГЭК
д.т.н., профессор
Пятковский О.И.
_____
(подпись)

Рубцовск 2017

Реферат

Отчет 102 страницы, 36 рисунков, 15 таблиц, 39 источников.

Ключевые слова: системный анализ, VoIP телефония, мобильное приложение, ОС Андроид.

Целью выпускной квалификационной работы является разработка мобильного приложения для организации доступа к сервису VoIP коммуникации «Русский телефон», которое позволит обеспечить пользователей удобным средством коммуникации, а также привлечет новых клиентов сервиса.

Объектом выпускной квалификационной работы является сервис VoIP коммуникации «Русский телефон» компании ЗАО «Рубцовск».

Предметом – организация доступа к функционалу VoIP сервиса.

Методы решения поставленных задач: системный анализ, моделирование предметной области, функционально-ориентированная методология.

Результаты квалификационной работы: проанализированы данные о предметной области, проведен системный анализ предметной области, на основе функциональной модели выявлены цели создания проекта, требования к нему и разработано мобильное приложение.

Содержание

Введение

Неотъемлемой частью развития человечества является развитие систем связи и способов коммуникации. Однако, не смотря на развитие технологий, индустрия коммуникационных услуг развивается медленно и не успевает за потребностями рынка. Следствием медленного изменения фундаментальной структуры коммуникационных сетей являются дорогие, как для оператора, так и для клиента, звонки между абонентами разных сетей. Роуминг вне зоны обслуживания своего оператора и изменение тарифов в разных регионах в пределах обслуживания, также являются следствием низкой гибкости сотовых сетей. При этом с каждым годом растет доля не голосового трафика.

При этом, в следствии бурного роста потребности в широкополосном доступе к сети Интернет, активно внедряются 3G, 4G сети, распространены публичные Wi-Fi точки доступа. Благодаря этому, широкое распространение получили сервисы обмена контентом по интернет протоколам.

Одним из таких сервисов, является РУССКИЙ ТЕЛЕФОН™ – это сервис, который дает возможность по экономным тарифам звонить в любые страны или города России с компьютера, ноутбука, планшета и т.д. Звонки между абонентами сервиса по присвоенным им 6-значным номерам бесплатные.

По статистике, примерна половина интернет-пользователей в России выходят в Сеть с помощью смартфонов. Очевидно, что наличие мобильного приложения является очень важным для такого сервиса. Без собственного мобильного приложения, любая рекламная компания была бы малоэффективна, его создание, напротив, должно вызвать не только приток новых пользователей сервиса, но и побудить уже имеющихся, на более активное использование.

Отсутствие мобильного приложения для этого сервиса, делает эту тему актуальной. Появление удобного средства совершения выгодных звонков, крайне важно для желающих сэкономить на звонках. Наиболее остро, данная проблема стоит для компаний, оплачивающих услуги связи для работников по корпоративным тарифам операторов связи, что является значительной статьей расходов если совершать звонки за рубеж или находясь в роуминге.

Объектом выпускной квалификационной работы является сервис VoIP коммуникации «Русский телефон» компании ЗАО «Рубцовск».

Предметом является организация доступа к функционалу VoIP сервиса.

Целью выпускной квалификационной работы является разработка мобильного приложения для организации доступа к сервису VoIP коммуникации «Русский телефон», которое позволит обеспечить пользователей удобным средством коммуникации, а также привлечет новых клиентов сервиса.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области;
- построить модель предметной области «как есть», с целью выявления недостатков;
- построить модель предметной области «как должно быть»;
- выполнить обзор и анализ существующих разработок;
- разработать мобильное приложение;
- оценить эффективность от внедрения мобильного приложения.

Методы, используемые при написании работы: системный анализ, моделирование предметной области. Ресурсами информации, используемой в данной работе, является Интернет, современные книги и т.д.

Средства, используемые при проектировании: Visio – графический инструментарий для визуального моделирования баз данных и хранилищ данных, Android Studio – инструмент разработки мобильного приложения.

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

Закрытое Акционерное Общество «Рубцовск» было основано 14 февраля 1991 года. В данный момент, компания предоставляет широкий спектр услуг, связанных с информационными технологиями.

Статус закрытого акционерного общества подразумевает, что акции распределяются только среди его учредителей или иного, заранее определенного круга лиц.

Территориально предприятие расположено в городе Рубцовске Алтайского края, однако его услугами пользуются по всей России и за рубежом.

Работники ЗАО «Рубцовск», впервые в городе:

– автоматизировали справочную службу «09», банк, все рабочие места в своем предприятии;

– разработали многотерминальную систему на IBM PC под MS-DOS;

– поставили модемы на IBM PC, сделали локальную сеть;

– установили правовую справочную систему КонсультантПлюс;

– получили лицензии Министерства связи на телематические услуги и передачу данных;

– разработали банковскую систему пятого поколения;

– стали продавать свои программы в России и за рубежом;

– сгенерировали Unix на IBM PC;

– сделали узел Internet, установили Web-сервер и разработали сайт;

– создали работающую систему IP-телефонии;

– разработали биллинговую систему Internet-провайдера;

– получили Большую Золотую медаль Сибирского форума

информационных технологий и телекоммуникаций;

– перевели свое ПО под технологию SaaS.

Закрытое акционерное общество, создается, с целью получения прибыли и может заниматься любой не запрещенной законом деятельностью.

Основной целью деятельности Общества является наиболее полное и качественное удовлетворение потребностей российских и иностранных предприятий, организаций и граждан в продукции (работах, услугах) производимой Обществом.

Главным направлением деятельности ЗАО «Рубцовск» является разработка программного обеспечения. Примером разрабатываемых решений являются Бухгалтерия предприятий, учет товарных и материальных ценностей, расчет заработной платы, автоматизированные банковские системы, управление предприятием, биллинговые системы операторов связи и программно-аппаратный комплекс обеспечивающий работу IP-телефонии.

Отличительной чертой предприятия является внедрение и использование новейших сетевых и телекоммуникационных технологий. Профессиональная работа с оборудованием CISCO, модемами, серверами и каналами связи позволила стать мощным региональным провайдером Интернет и IP-телефонии. Собственная биллинговая система позволяет решить все вопросы учета и статистики.

Компания одной из первых за Уралом, стала предоставлять услуги выхода в интернет. И по сей день является интернет-провайдером.

ЗАО «Рубцовск» является разработчиком и владельцем старейшего информационного портала города Рубцовска - RUBTSOVSK.RU. Сайт является средством массовой информации согласно свидетельству «Эл N ФС77-66851».

Фирма оказывает информационные услуги в качестве регионального информационного центра сети «КонсультантПлюс». Также, обеспечивает работу многих клиентов с ЕГАИС и средствами цифровой подписи.

Обслуживание компьютерной техники, сетевого оборудования и операционных систем для обеспечения работы программных разработок фирмы

позволяет оказывать соответствующие услуги ремонта и настройки на высочайшем уровне качества.

В соответствующем отделе можно сделать ксерокопии, отправить факс, электронное письмо, оплатить услуги Интернет, IP-телефонии, отсканировать, распечатать документы, дать объявление в газету «Экспресс» или на сайт RUBTSOVSK.RU.

Для ведения описанной выше деятельности, ЗАО «Рубцовск» обладает целым рядом лицензий.

В частности, лицензии Федеральной службы по надзору в сфере связи:

1. Услуги связи по передаче данных для целей передачи голосовой информации № 145970 (полная версия) от 30.08.2016 г.
2. Телематические услуги связи № 145971(полная версия) от 30.08.2016г.
3. Услуги связи по передаче данных, за исключением услуг связи по передаче данных для целей передачи голосовой информации № 145969(полная версия) от 30.08.2016 г.
4. Услуги связи по предоставлению каналов связи № 100757 (полная версия) от 16.10.2012 г.
5. Решение о выделении ресурса нумерации единой сети электросвязи РФ № 17855 от 19.12.2007 г.

Согласно лицензии УФСБ №134 от 31.08.2012 г. (ЛСЗ №0000210), предприятию разрешается разработка, производство, распространение шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, выполнение работ, оказание услуг в области шифрования информации, техническое обслуживание шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств (за исключением случая, если техническое обслуживание шифровальных (криптографических) средств, информационных

систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, осуществляется для обеспечения собственных нужд юридического лица или индивидуального предпринимателя)

Организационная структура ЗАО «Рубцовск» представлена на рисунке 1.1.

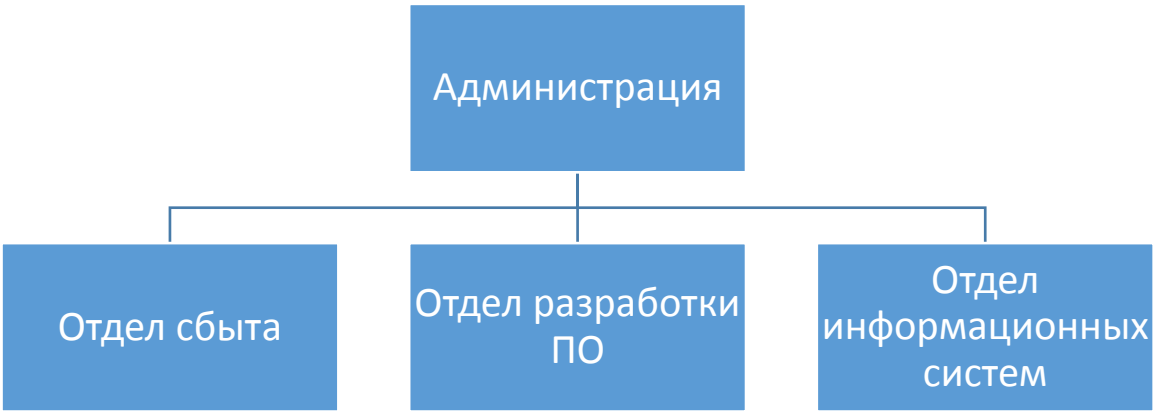

Рисунок 1.1 – Организационная структура ЗАО «Рубцовск»

Администрация выполняет руководящую функцию, отвечает за бухгалтерский учет и кадровую работу.

Состав администрации включает директора и главного бухгалтера.

Отдел сбыта отвечает за работу с клиентами, сопровождение ПО и общее обслуживание.

Состав отдела сбыта включает начальника отдела, трех ведущих специалистов и оператора.

Отдел разработки ПО отвечает непосредственно за разработку, внедрение и обслуживание программных и аппаратных средств.

Состав отдела разработки ПО включает начальника отдела, программиста первой категории, двух программистов и ведущего специалиста.

Отдел информационных систем отвечает за системное администрирование, техобслуживание, системное программирование.

Состав отдела информационных технологий включает начальника отдела, ведущего программиста, системного программиста, ведущего инженера по

оборудованию.

Архитектура информационной системы данных включает: БД и хранилища данных, системы управления БД или хранилищами данных, правила и средства санкционирования доступа к данным. Техническая архитектура состоит из сетевой архитектуры и архитектуры платформ. Сетевая архитектура включает: локальные и территориальные вычислительные сети, используемые в сетях коммуникационные протоколы, сервисы и системы адресации, аварийные планы по обеспечению бесперебойной работы сетей в условиях чрезвычайных обстоятельств.

Архитектура платформ включает: аппаратные средства вычислительной техники, серверы, рабочие станции, накопители и другое компьютерное оборудование, операционные и управляющие системы, утилиты и офисные программные системы, аварийные планы по обеспечению бесперебойной работы аппаратуры (главным образом, серверов) и БД в условиях чрезвычайных обстоятельств. Выход в Интернет осуществляется через прокси-сервер, который обеспечивает распределение прав доступа пользователей и обеспечивает защиту от внешних атак.

Рабочие места сотрудников предприятия оснащены необходимым набором оборудования (компьютер, факс, телефон, принтер, коммутатор).

Одним из перспективных направлений развития предприятия является IP-телефония.

VoIP (Voice over Internet Protocol) – или IP-телефония, это телефонная связь по протоколу IP. Подразумевается набор коммуникационных протоколов, технологий и методов, обеспечивающих традиционные для телефонии набор номера, дозвон и двустороннее голосовое общение, а также видео общение по сети Интернет или любым другим IP-сетям. Сигнал по каналу связи передаётся в цифровом виде и, как правило, перед передачей преобразовывается (сжимается) с тем, чтобы удалить избыток информации и снизить нагрузку на сеть передачи данных.

В качестве АТС используется оборудование CISCO. А в качестве сервера

IP телефонии программное решение «Астериск», на базе мощного сервера.

Asterisk – свободное решение компьютерной телефонии (в том числе, VoIP) с открытым исходным кодом от компании Digium, первоначально разработанное Марком Спенсером. Приложение работает на операционных системах Linux, FreeBSD, OpenBSD и Solaris и др. Имя проекта произошло от названия символа «*» (англ. asterisk — «звездочка»). Asterisk в комплексе с необходимым оборудованием обладает всеми возможностями классической АТС, поддерживает множество VoIP-протоколов и предоставляет богатые функции управления звонками.

С помощью этого оборудования обеспечивается бесперебойная работа многочисленных IP телефонов предприятия, а также предоставляются услуги связи для многих других.

Но ЗАО «Рубцовск» не остановилось на достигнутом. Используя возможности SIP протокола, сервера предприятия позволяют связываться абонентам по всему миру.

SIP (англ. Session Initiation Protocol – протокол установления сеанса) – протокол передачи данных, описывающий способ установления и завершения пользовательского интернет-сеанса, включающего обмен мультимедийным содержимым (IP-телефония, видео- и аудиоконференции, мгновенные сообщения, онлайн-игры). Протокол описывает, каким образом клиентское приложение (например, софтфон) может запросить начало соединения у другого, возможно, физически удалённого клиента, находящегося в той же сети, используя его уникальное имя.

Пользователи, могут делать звонки используя стационарный телефон в Рубцовске или целый набор разнообразных средств по всему миру. Например, можно сделать звонок через браузер, благодаря Flash-софтофону.

Софтфон (англ. software telephone, программный телефон) – класс программного обеспечения для совершения телефонных (голосовых) или видеозвонков через Интернет (в общем случае через любую IP-сеть) без использования дополнительного аппаратного обеспечения, за исключением

средств ввода-вывода звукового и/или видео сигнала.

Или использовать стационарный IP-телефон, настроенный на связь с сервером.

Или один из мобильных клиентов для совершения звонков по SIP протоколу.

Мобильное приложение (англ. «Mobile app») – программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах. Интерфейс и логика работы ориентированы на небольшой, обычно сенсорный, экран.

РУССКИЙ ТЕЛЕФОН™ – это сервис, который дает возможность по экономным тарифам звонить в любые страны и города России напрямую с компьютера, ноутбука, планшета и т.д. Звонки между абонентами сервиса по присвоенным им 6-значным номерам бесплатные.

Сервис развился из услуги IP телефонии, для стационарных номеров. Звонки клиентов, проходили через сервер Астериск. Стоимость вызова в IP-телефонии определяется по так называемой «системе с минимальной стоимостью маршрутизации звонка» (LCR, Least Cost Routing System), которая основана на том, что осуществляется проверка пункта назначения каждого телефонного звонка, как только он сделан внутри сети, что даёт потребителю самую низкую цену. Использовались как обычные телефонные аппараты поддерживающие тональный набор, так и специализированные аппараты.

Благодаря SIP протоколу, и использующим его софтофонам, удалось создать сервис звонков из браузера, софтофона установленного на персональном компьютере. Для пользователя это означает возможность голосового общения между любыми распространенными устройствами, начиная с телефона, заканчивая браузером.

Специфической особенностью IP телефонии, а значит и сервиса, является «система с минимальной стоимостью маршрутизации звонка», которая позволяет сэкономить на международной связи, а также на связи во время путешествия.

Ну а в случае звонков между абонентами РУССКОГО ТЕЛЕФОНА разговор бесплатен. Этой особенностью можно воспользоваться, разместив на сайте «кнопку бесплатного звонка», которая используя встроенный в страницу нашего сайта софтофон, позволит установить соединение.

1.2 Анализ функционирования объекта исследования

Системный анализ – научный метод познания, представляющий собой последовательность действий по установлению структурных связей между переменными или элементами исследуемой системы.

Моделирование бизнес-процессов необходимо для выявления текущих проблем на предприятии и предвидения будущих. Моделирование деловых процессов, как правило, выполняется с помощью CASE-средств. Для моделирования воспользовался программой Visio, построив диаграмму IDEF0.

IDEF0 (Function Modeling) – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.

Одним из основных понятий стандарта IDEF0 является декомпозиция (Decomposition). Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции.

Модель IDEF0 всегда начинается с представления системы как единого целого – одного функционального блока с интерфейсными дугами, уходящими за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется контекстной диаграммой.

В пояснительном тексте к контекстной диаграмме должна быть указана цель (Purpose) построения диаграммы в виде краткого описания и зафиксирована точка зрения (Viewpoint). Фактически цель определяет соответствующие области в исследуемой системе, на которых необходимо

фокусироваться в первую очередь. Точка зрения определяет основное направление развития модели и уровень необходимой детализации.

В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется дочерней (Child Diagram) по отношению к нему, а каждый из функциональных блоков, принадлежащих дочерней диаграмме, соответственно называется дочерним блоком (Child Box).

В основе функционального моделирования лежат четыре основных понятия: функциональный блок, интерфейсная дуга, декомпозиция, глоссарий.

Функциональный блок (Activity Box) представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении. На диаграмме функциональный блок изображается прямоугольником. Каждая из четырех сторон функционального блока имеет свое определенное значение (роль), при этом:

- верхняя сторона имеет значение «Управление» (Control);
- левая сторона имеет значение «Вход» (Input);
- правая сторона имеет значение «Выход» (Output);
- нижняя сторона имеет значение «Механизм» (Mechanism).

Интерфейсная дуга (Arrow) отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, представленную данным функциональным блоком. Интерфейсные дуги часто называют потоками или стрелками.

С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Такими объектами могут быть элементы реального мира или потоки данных и информации.

В зависимости от того, к какой из сторон функционального блока

подходит данная интерфейсная дуга, она носит название «входящей», «исходящей» или «управляющей».

Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (Data Flow Diagram) и WFD (Work Flow Diagram).

В ЗАО «Рубцовск» реализован сервис VoIP коммуникации «Русский телефон» который предоставляет возможность совершать звонки между абонентами и на номера телефонных сетей.

Для доступа к сервису можно использовать:

– браузер компьютера (ноутбук или стационарный);
– мобильное устройство (смартфон или планшет);
– телефон IP-телефонии.

В диаграммах IDEF0 в качестве «механизмов» указаны способы доступа к функционалу сервиса на момент анализа.

На вход поступают необходимые данные, в результате удовлетворяется та или иная потребность клиента, в зависимости от требуемого функционала.

Деятельность сервиса управляется законами и нормативными документами, а также пользовательским соглашением.

На рисунке 1.2 представлена контекстная диаграмма «КАК-ЕСТЬ»

Рисунок 1.2 – Контекстная диаграмма IDEF0 AS-IS «Организация доступа к сервису VoIP коммуникации «Русский телефон»»

Главная функция делится на подфункции доступа к:

– регистрации новой учетной записи;

– функции совершения звонков;

– информации о текущем балансе;

– пополнению лицевого счета.

Подфункции сервиса представлены на рисунке 1.3 в виде детализации диаграммы AS-IS.

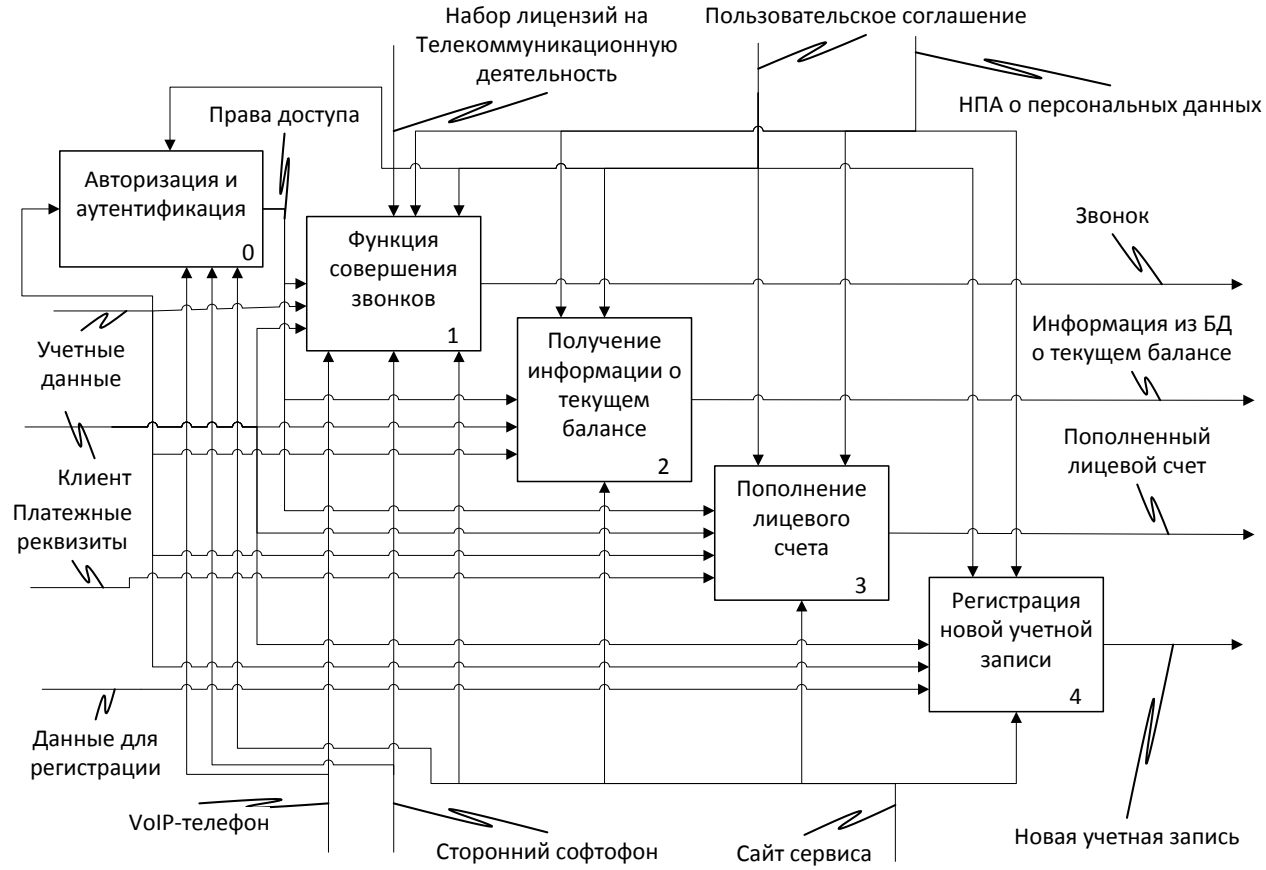

Рисунок 1.3 – Детализированная диаграмма IDEF0 AS-IS «Организация доступа к сервису VoIP коммуникации «Русский телефон»»

После того, как модель системы построена ее необходимо верифицировать. Все ее объекты должны быть тщательно описаны и детализированы.

Способ распространения информационных ресурсов посредством браузера имеет ряд недостатков:

– отсутствие способа для хранения полученной информации в памяти

устройства;

– неудобный пользовательский интерфейс;

– отсутствие интерактивного взаимодействия с пользователем;

– не все браузеры поддерживают технологии необходимые для совершения звонка.

Еще одним способом доступа к ресурсам информационно-образовательного портала является использование мобильного приложения, данный способ очень важен, так как люди привыкли совершать звонки используя смартфон. Мобильные приложения в смартфоне имеют доступ к той-же телефонной книге, что используется и для обычных звонков.

Исходя из этого, было принято разработать мобильное приложение.

1.3 Определение цели и задач проектирования

Мобильное приложение – последовательность инструкций, предназначенных для исполнения операционной системой мобильного устройства. Возможность звонить из любого мета, а не только рядом со стационарным устройством, трудно переоценить. Совершение звонков через мобильное приложение позволяет использовать имеющуюся телефонную книгу устройства. Приложение гораздо менее требовательно к производительности устройства, чем мобильный браузер. Данные факторы обеспечивают удобство от использования мобильного приложения.

В связи с этим можно выделить цель: разработать мобильное приложение для организации доступа к сервису VoIP коммуникации «Русский телефон», которое позволит обеспечить пользователей удобным средством коммуникации, а также привлечет новых клиентов сервиса.

Исходя из цели, можно выделить задачи, решаемые мобильным приложением:

– возможность совершения аудио звонков;

– хранение истории звонков;

– обеспечение доступа к контактам мобильного устройства с возможностью позвонить на мобильные номера и номера «Русского телефона»;

– предоставление доступа к изменению телефонной книги (добавление, удаление и изменение контактов);

– регистрация в сервисе непосредственно из приложения;

– пополнение баланса в сервисе из приложения;

– предоставление возможности узнать баланс счета;

– возможность обращения в службу поддержки непосредственно из приложения;

– возможность пригласить, воспользоваться сервисом, людей из списка контактов.

Приложение «Русский телефон» будет интегрировано в уже работающий одноименный сервис и являться дополнительным, более удобным, способом доступа к широкому спектру функций.

1.4 Обзор и анализ существующих разработок, выбор технологии проектирования

1.4.1 Обзор и анализ существующих программных разработок

VoIP вовсе не молодая технология. Интернет пестрит рекламой «бесплатных» звонков по всему миру. Многие из предложений действительно заслуживают внимания, но зачастую от использования удерживает важный нюанс, например, невозможность совершения звонка на определенные направления или запутанная система ценообразования.

Рассмотрим несколько сервисов для звонков через интернет. Примем, что звонки между абонентами в нижеперечисленных сервисах – бесплатны и не лимитированы. Звонки на мобильные и стационарные номера зачастую являются платными или требуют некий непрямой способ оплаты, например,

демонстрируя рекламу.

Самыми популярными сервисами для совершения звонков через интернет являются сервисы «Skype» и «Viber Out» (рисунок 1.4). Высокое качество звонков, возможность звонить во многие страны, как на мобильные, так и на стационарные номера. Однако, в ряде стран данные сервисы запрещены или ограничены в использовании. Главной отрицательной особенностью являются цены непосредственно связанные с курсом валюты США.

Рисунок 1.4– Сервисы «Skype» и «Viber Out»

«Viber» лидирует среди мобильных пользователей, есть приложения для всех популярных мобильных платформ. Также существуют клиенты под все популярные операционные системы персональных компьютеров. Отсутствует возможность использования сервиса через браузер. «Viber» не совместим с офисными АТС и не может быть интегрирован в Asterisk, что ограничивает его использование в бизнесе.

Основной причиной быстрого роста популярности «Viber» стало удобное мобильное приложение. Рассмотрим принятые в сервисе проектные решения.

В качестве способа регистрации «Viber» сразу-же выбрал номер телефона с подтверждением по SMS. Причем номер стал одновременно уникальным идентификатором, позволяющим искать других пользователей сервиса. «Viber» жестко привязывается к номеру телефона и мобильному устройству на котором установлен. Для переустановки требуются дополнительные действия, подтверждения по SMS. После установки и регистрации, телефонная книга мобильного устройства пересылается на сервера «Viber» для поиска других абонентов, которым можно делать бесплатные звонки. Сервис звонков на мобильные и стационарные телефоны называется «Viber Out».

«Skype», пожалуй, самый известный сервис для звонков через интернет, однако не многие знают, что с его помощью можно звонить на стационарные и мобильные телефоны. «Skype» стал на столько популярен, что во многих странах его использование ограничено законодательством. В любом случае, цены на звонки в большинстве случаев не позволяют рассматривать его как замену сотовой связи. Есть клиенты для всех популярных платформ. Однако различные версии часто не совместимы между собой. Мобильная версия приложения для ОС «Андроид» занимает очень много места в памяти по сравнению с другими подобными приложениями. На слабых устройствах есть проблемы с производительностью, вплоть до отказа в работе приложения. Есть возможность совершать звонки в браузере. Есть специальные предложения для бизнеса, позволяющие интегрироваться в офисную телефонную сеть с помощью специального дорогостоящего оборудования. Возможна интеграция с сервером Asterisk, однако способ подключения часто меняется, регулярно требуя вмешательства специалистов, для восстановления работоспособности.

Из сообщения информационного агенства lenta.ru рынок VoIP-телефонии в России составляет около восьми миллиардов рублей, причем звонки на обычные телефоны являются основным методом монетизации. Подобные звонки осуществляются благодаря терминации «приземления» звонка на локальную телефонную сеть, что обеспечивает наименьшую стоимость телефонного соединения. Обычно сеть, на которую «приземляется» звонок, берет за это деньги. Для Российских компаний, эта цена обычно ниже, что позволяет успешно конкурировать с описанными выше гигантами.

Но популярности VoIP сервисов оказываются не довольны телекоммуникационные компании. Во многих странах подаются жалобы, что эти сервисы используют ресурсы операторов связи, но не несут никаких расходов. В некоторых случаях, это приводит к запрету или ограничению для вида услуги или для отдельных компаний.

В Мае 2016 года шумиху в прессе вызвал опубликованный Минкомсвязью проект, фактически делающим невозможность «приземление

звонка» в России. Однако позже Минкомсвязи опровергло информацию о том, что в России будет введен запрет на звонки из Skype, Viber и других подобных приложений на стационарные и мобильные номера.

Одновременно, операторы связи разрабатывают свои сервисы VoIP телефонии. К таким сервисам относятся «МТС Connect» от МТС, «МультиФон» от Мегафон, «Домашний телефон» от Билайн, «Аллё» от Ростелекома. Однако, ввиду того, что успех этих сервисов уменьшат прибыли от основной деятельности – развиваются эти сервисы весьма медленно.

Самым популярным из них является «МультиФон» от Мегафон (рисунок 1.5).

Рисунок 1.5– Сервис «МультиФон» от «Мегафон»

«МультиФон» –продукт выпущенный еще в 2008 году. Это первый в мире VoIP-сервис от оператора связи.

Такое происхождение дает многие преимущества. Например, приложение «МультиФон» может принимать звонки идущие на обычный мобильный номер Мегафон, если он находится вне зоны действия сети.

Очень выгодны звонки на мобильные номера Мегафон и относительно дешевы на номера других операторов. Причем, деньги списываются с баланса телефона.

«МультиФон» работает по стандартному протоколу SIP, поэтому есть возможность подключать к «МультиФону» любые устройства, поддерживающие SIP-телефонию – от SIP-телефонов и SIP-клиентов на ПК и смартфонах до IP-PBX, например, Asterisk.

Сервис популярен в России, однако пользователи отмечают неудобный интерфейс в приложениях, а также их нестабильную работу и многочисленные

«баги».

В Москве известен сервис «МТС Connect» от МТС (рисунок 1.6).

Рисунок 1.6– Сервис «МТС Connect» от «МТС»

Звонки через интернет, чаты, передача файлов.

Отличительной особенностью является возможность звонить куда угодно по ценам тарифного плана своего номера МТС. Что решает проблемы оплаты роуминга в поездках.

Для некоторых моделей мобильных телефонов (Samsung Galaxy S7/S7 EDGE) доступны звонки по технологии Wi-Fi Calling, позволяющей использовать Wi-Fi роутеры в качестве «базовой станции МТС».

Интернет трафик при звонках не учитывается, что позволяет не беспокоится о ограничениях тарифа на доступ в Интернет. Бесплатные минуты тарифа МТС не только учитываются в стоимости разговора, для «МТС Connect» их число удваивается.

Однако есть серьезный недостаток. Сервис доступен только для абонентов Москвы и Московской области.

Для использования нужен клиент от МТС, стандартный SIP-софтофон не подойдет. Интеграция с офисными АТС и Asterisk не предусмотрена.

Теперь абонентам «Домашнего Интернета Билайн» доступна новая услуга – «Домашний Цифровой телефон». Отличительными особенностями услуги являются выгодные тарифы на междугородние и международные звонки и единый счет с услугой «Домашнего Интернета Билайн» (рисунок 1.7).

Рисунок 1.7– Сервис «Домашний Цифровой телефон» от «Билайн»

В рамках продукта предоставляются услуги местной телефонной связи сети общего пользования с выделением в пользование местных (городских) номеров телефонной сети общего пользования. Также предоставляется доступ к услугам внутризоновой, междугородной и международной телефонной связи через сеть «Билайн».

В сервисе можно использовать как обыкновенный кнопочный телефон, так и цифровые аппараты VoIP, а также совершать звонки с компьютера или ноутбука при помощи программы «Софтофон» и использовать функцию видеозвонков.

Работа по протоколу SIP подразумевает возможность интеграции с Asterisk или другим PBX сервером.

Основным ограничением является необходимость быть абонентом «Домашнего Интернета Билайн». Причем для звонка нужно пользоваться именно сетью домашнего интернета Билайн. Эта услуга доступна только в Москве, Санкт-Петербурге, Екатеринбурге, Красноярске и Уфе.

Немногие знают, что Ростелеком предоставляет не только связь по стационарным телефонам, но и по мобильным (своя SIM-карта). По заявлению президента Ростелекома Сергея Калугина осенью 2016 года должен был быть запущен мессенджер от Ростелеком – «Аллё» (рисунок 1.8) [17].

Рисунок 1.8– Сервис «Аллё» от «Ростелеком»

В «Ростелекоме» заявили, что планируют запустить новый сервис по обмену звонками и сообщениями. Работать он будет не только на сети «Ростелекома», но и на сети других операторов, в том числе мобильных. Но сервис будет платным – компания не планирует конкурировать на рынке бесплатных мессенджеров и VoIP приложений. [28]

Позже Сергей Калугин объявил о переносе даты запуска на декабрь, также сообщил, что сервис будет примером классической IP-телефонии.

Открытой информации по поводу сервиса и приложения крайне мало, но можно предположить, что услуга будет схожей с приложением «Телефон Ростелеком». «Телефон Ростелеком» – это SIP телефон, предназначенный только для пользователей услуги «облачная АТС Ростелеком». Его отличительной особенностью является возможность приема звонков с «домашнего» номера телефона и с номеров «8 800». Однако, для использования «домашнего» номера в SIP придется лично прийти в офис «Ростелеком» для заключения соответствующего договора.

Большинство описанных выше сервисов используют свои собственные протоколы работы. Исходный код приложений, серверов и протоколов передачи данных – закрыт.

В качестве примера сервиса, работающего по протоколу SIP, стоит упомянуть популярный в России сервис «ZADARMA» (рисунок 1.9).

Рисунок 1.9 – Сервис «ZADARMA»

Имеет собственные мобильные приложения, но может использоваться практически любой SIP-софтофон. Работая по протоколу SIP может быть легко интегрирован с офисными АТС и сервером Asterisk. Тарификация имеет множество условий, которые могут оказать радикальное влияние на выгодность звонка.

Кроме этих сервисов существуют сотни других, работающих преимущественно по протоколу SIP.

Перечислять их все неконструктивно, поэтому выделим их основные особенности.

Объединяющей особенностью подавляющего большинства подобных сервисов, является непрозрачная тарификация или скрытые способы заработка на пользователях. Во многих случаях заявляется о бесплатных звонках, но накладываются разнообразные ограничения:

1. Бесплатны только первые минуты разговора.
2. Число бесплатных минут ограничено.
3. Бесплатные минуты делятся между всеми абонентами сервиса по принципу «кто первый позвонил»
4. Бесплатные минуты доступны только при положительном балансе.
5. Бесплатные минуты можно получить за просмотр рекламных видео, установку определенных приложений или «лайки» в социальных сетях.
6. Бесплатные звонки становятся доступны только после пополнения счета на значительную сумму.

Платные тарифы также чаще всего отличаются запутанностью и зависимостью от множества факторов.

В краткосрочной перспективе подобные сервисы могут оказаться полезными, но их использование несет большой риск, так как бесплатные звонки могут внезапно привести к значительным тратам при нарушении какого-либо лимита или условия. Более того, мошенники часто используют подобные сервисы для наживы. Нередки случаи, когда сервисы внезапно пропадали с деньгами пользователей или изымали деньги внезапным изменением тарифов. Возможны случаи хищения данных банковской карточки, вставка рекламы, сбор конфиденциальной информации о пользователе.

Для доступа к этим сервисам чаще всего используется браузер с установленным «Флеш-плеером». В качестве мобильного клиента используется софтофон с ограниченной функциональностью, часто не работающий на многих устройствах. Интеграция с сервером Asterisk может быть сопряжена с трудностями ввиду отсутствия поддержки со стороны сервиса.

Отсутствие лицензии и российского представительства, ограничивает пользователей в возможности защиты своих прав.

1.4.2 Выбор технологии проектирования

В мобильной разработке существует множество подходов к проектированию и реализации программных продуктов.

На сегодняшний день, подавляющее большинство смартфонов и планшетов находятся под управлением операционных систем Андроид и IOS. Ситуация на рынке не позволяет игнорировать ни одну из этих платформ, поэтому, приступая к проектированию стоит рассмотреть возможность кроссплатформенной разработки.

Разработка данного типа ведётся при помощи различных фреймворков (Kony Platform, Adobe AIR, PhoneGap, Appcelerator Titanium, IBM Worklight и др.) и технологий HTML5/CSS3 нередко с использованием сценариев Java.

Плюсы данного типа приложений:

1. Для работы на разных операционных системах используется один и тот же исходный код.
2. Дешевая стоимость разработки.
3. Самые низкие сроки разработки.

Возникает вопрос, зачем тогда использовать нативные приложения если кроссплатформенные решения быстрее разрабатываются и дешевле стоят? Дело в том, что, такие приложения зачастую подходят лишь для разработки пилотного продукта (макета или прототипа) или разработки несложных и стандартизированных решений.

Минусы кроссплатформенного типа:

1. Сильно ограниченные возможности.
2. Высокая ресурсоёмкость, что значительно понижает производительность приложений.
3. Невозможность адаптации под новые версии ОС до тех пор, пока используемый фреймворк не обновит свою среду разработки.
4. Достаточно долгий отклик приложений (опять же из-за высокой ресурсоёмкости).

5. Низкая стабильность приложений (возможны ошибки на некоторых платформах).

6. Реализация сложных сценариев практически невозможна.

Но главным ограничением для использования подобных решений при создании софтофона, является необходимость работы с протоколом SIP. Поддержка работы с этим протоколом при кроссплатформенной разработке в большинстве случаев затруднена или вовсе невозможна.

Нативные приложения (от англ. Native – «родной») – это тип приложений, разрабатываемый на «родном» для каждой платформы языке программирования.

Разработка мобильных приложений для iPhone, iPad, iPod велась на языке программирования Objective-C. Потом компания Apple в 2014 году вместе с выпуском iOS 8 и iPhone 6/6+ выпустила новый язык программирования Swift для Xcode 6. На сегодняшний день большинство новых разработок для этой платформы ведутся именно на нем.

Разработка мобильных приложений под Андроид осуществляется при помощи объектно-ориентированного языка программирования Java.

Именно нативный способ разработки приложений решает самый широкий спектр задач, и позволяет разрабатывать наиболее сложные продукты.

Плюсы нативных приложений:

1. Используя нативный метод можно реализовать задачу любой сложности.

2. Производительность гораздо выше, чем у кроссплатформенных и web-приложений.

3. Возможность разрабатывать приложения с использованием «дополненной реальности».

4. Реализация сложных игровых сценариев.

5. Обработка значительного количества данных на стороне клиента.

6. Использование расширенного функционала мобильного устройства (геолокация, камера, адресная книга и др.).

Помимо платформозависимости, недостатком такого подхода являются:

1. Трудоёмкость разработки.
2. Сравнительно большой срок разработки.

В данном случае, необходимость диктует выбор пути нативной разработки для каждой платформы в отдельности. Как говорилось ранее, платформа Андроид имеет преимущество в большей распространенности. Разумно начать именно с этой операционной системы, написав приложение на языке Java.

Ввиду быстро изменяющейся экономической ситуации и требовании бизнеса в наличии рабочего прототипа, в качестве технологии проектирования использовалась технология прототипного проектирования.

Прототипное проектирование – эта технология позволяет создавать на первой стадии реализации команд интерактивной модели системы, т.е. системы прототипа, который позволяет пользователю увидеть будущую систему.

Получаемые в результате прототипы могут быть использованы в ходе цикла Agile-разработки и могут быть внедрены в эксплуатацию с последующим обновлением после каждой итерации разработки.

1.5 Выбор и обоснование проектных решений

Для обоснованного выбора платформ разработки мобильных приложений рассмотрим распространённые операционные системы.

Современные операционные системы для мобильных устройств: Android, CyanogenMod, Cyanogen OS, Fire OS, Flyme OS, iOS, Windows Phone, BlackBerry OS, Firefox OS, Sailfish OS, Tizen, Ubuntu Touch. Устаревшие, ныне не поддерживаемые программные платформы: Symbian, Windows Mobile, Palm OS, webOS, Maemo, MeeGo, LiMo.

По данным «Net Applications» и «StatCounter» состоянию на сентябрь 2016 года, почти девяносто пять процентов рынка мобильных операционных систем занимают «Android» и «IOS».

На рисунке 1.9 видна статистика от «Net Applications» – фирмы, которая занимается веб-аналитикой. Компания широко известна тем, что ведёт исследования и глобальную статистику доли мирового рынка для веб-браузеров и операционных систем [30].

MONTH	ANDROID	IOS	WINDOWS PHONE	JAVA ME	SYMBIAN	OTHER
November, 2015	57.10%	34.88%	3.14%	1.76%	1.89%	1.23%
December, 2015	57.29%	35.43%	2.58%	1.85%	1.70%	1.14%
January, 2016	58.75%	32.93%	2.86%	2.14%	1.73%	1.60%
February, 2016	59.65%	32.28%	2.57%	2.40%	1.57%	1.53%
March, 2016	60.99%	31.76%	2.54%	2.07%	1.40%	1.24%
April, 2016	61.92%	28.42%	4.03%	2.05%	1.02%	2.57%
May, 2016	70.85%	23.10%	2.57%	1.50%	0.86%	1.12%
June, 2016	65.58%	27.24%	3.26%	1.81%	1.08%	1.03%
July, 2016	66.01%	27.84%	2.79%	1.44%	1.03%	0.88%
August, 2016	66.87%	27.20%	2.30%	1.62%	1.03%	0.98%
September, 2016	69.18%	25.02%	2.35%	1.43%	1.11%	0.91%

Рисунок 1.9 – Статистика от «Net Applications»

На рисунке 1.10 видна статистика от «StatCounter» – веб-сайт, который представляет собой инструмент для анализа веб-трафика [3].

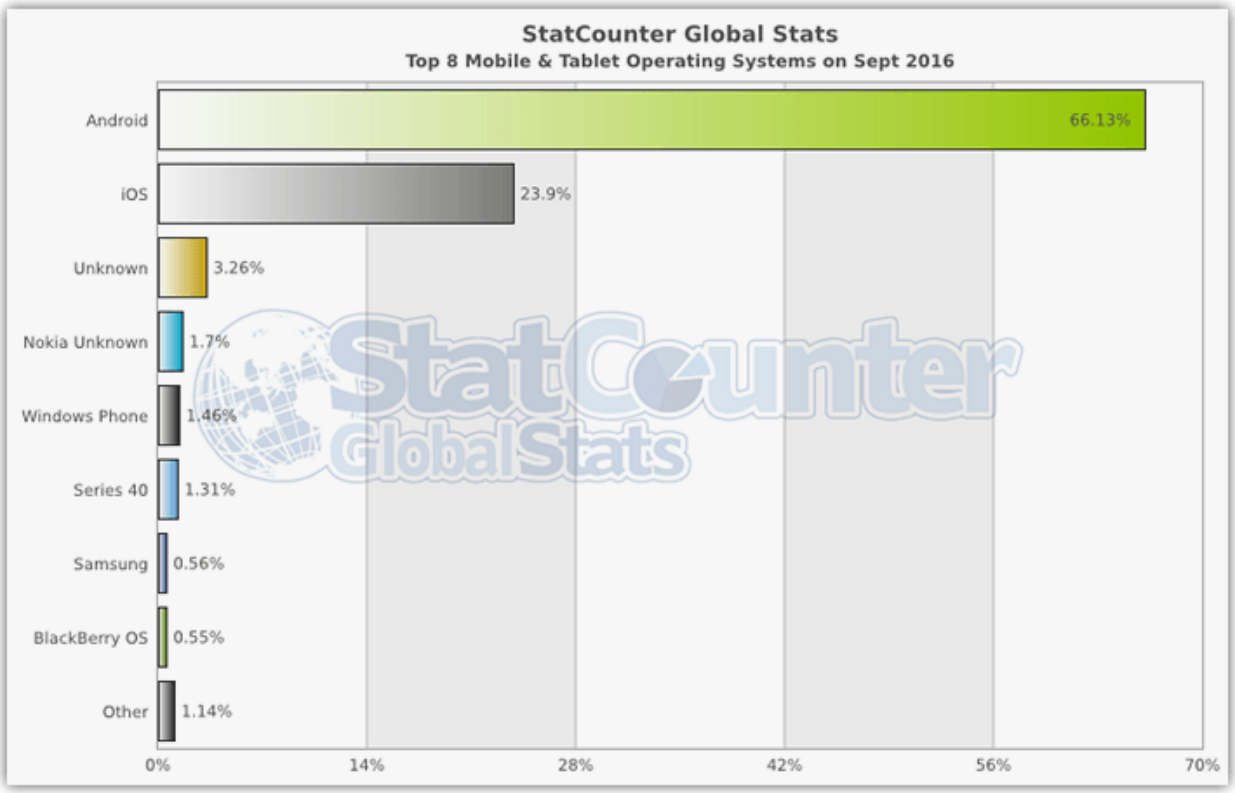

Рисунок 1.10– Статистика от «StatCounter»

Из этой статистики можно сделать вывод, что имеет смысл сосредоточится только на Android и IOS. Кроме того, стоит обратить внимание, что устройства под управлением IOS, предназначены в первую очередь для состоятельных пользователей. Андроид имеет более широкое маркетинговое позиционирование и почти в три раза более распространен.

Таким образом, выбираем операционную систему Андроид в качестве первоначальной платформы для мобильного приложения сервиса.

Существует множество способов разработки на ОС Андроид, в том числе с частичной кроссплатформенностью на другие ОС. Однако необходимость плотного взаимодействия с мобильным устройством, работа с протоколом SIP, а также требование к высокой производительности не позволяют выбрать кроссплатформенную разработку.

Но даже для «нативной» разработки есть возможность в выборе языка и фреймворков.

Основным языком программирования для операционной системы Андроид является «Java».

Java – строго типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска – 23 мая 1995 года [36].

На сегодняшний момент язык Java является одним из самых распространенных и популярных языков программирования. Java задумывался как универсальный язык программирования, который можно применять для различного рода задач. И к настоящему времени язык Java проделал большой путь, было издано множество различных версий. Текущей версией является Java 8, официальный релиз которой произошел в марте 2014 года. А Java превратился из просто универсального языка в целую платформу и экосистему, которая объединяет различные технологии, используемые в целом ряде задач:

от создания серверных приложений до написания крупных веб-порталов и сервисов. Кроме того, язык Java активно применяется для создания программного обеспечения для целого ряда устройств: обычных ПК, планшетов, смартфонов и мобильных телефонов и даже бытовой техники. Большинство программ для ОС Android пишутся именно на Java.

Ключевой особенностью языка Java является то, что его код сначала транслируется в специальный байт-код, независимый от платформы. А затем этот байт-код выполняется виртуальной машиной JVM (Java Virtual Machine). В этом плане Java отличается от стандартных интерпретируемых языков как PHP или Perl, код которых сразу же выполняется интерпретатором. В то же время Java не является и чисто компилируемым языком, как C или C++.

Подобная архитектура обеспечивает кроссплатформенность и аппаратную переносимость программ на Java, благодаря чему подобные программы без перекомпиляции могут выполняться на различных платформах. Для каждой из платформ может быть своя реализация виртуальной машины JVM, но каждая из них может выполнять один и тот же код.

Java является языком с Си-подобным синтаксисом и близок в этом отношении к C/C++ и C#.

Еще одной ключевой особенностью Java является то, что она поддерживает автоматическую сборку мусора. А это значит, что не нужно освобождать вручную память от ранее использовавшихся объектов, как в C++, так как сборщик мусора это сделает автоматически.

Java является объектно-ориентированным языком. Он поддерживает полиморфизм, наследование, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений.

В последние годы набирает популярность язык программирования «Kotlin».

Kotlin – статически типизированный язык программирования, работающий поверх JVM и разрабатываемый компанией JetBrains.

Компилируется также в JavaScript. Язык назван в честь острова Котлин в Финском заливе, на котором расположена часть Кронштадта.

Из отличительных черт языка Kotlin можно выделить ориентацию на обеспечение более высокой безопасности за счет реализации статических проверок для выявления таких проблем как разыменование NULL-указателей. Среди других особенностей, имеющих отношение к безопасности, отмечается отсутствие raw-типов, полное сохранение информации о типах в процессе выполнения и реализация массивов в виде инварианта. Из расширенных языковых возможностей отмечается поддержка функций высшего порядка, вывода типов значений на основании выражения, использование уточняющих «примесей» (mixin) и делегирования.

Однако язык пока недостаточно распространен и его использование сопряжено с увеличением сроков реализации, если у разработчика нет опыта в его использовании.

Исходя из описанных выше фактов и наличия, в качестве библиотеки примеров, большого количества «открытого» исходного кода на Java, именно этот язык выбран для разработки программного продукта для ОС Андроид.

Важнейшим проектным решением при разработке мобильного приложения для SIP-телефонии является выбор библиотек для соединения по протоколу SIP, для работы с аудиопотоком и прочими задачами необходимыми для работы мобильного клиента как средства VoIP связи.

Разработка своих протоколов и библиотек обработки звука и взаимодействием с мобильными устройствами на аппаратном уровне не соответствует целям и задачам, описанным в пункте 1.3.

Для наиболее быстрого и экономически выгодного внедрения приложения было принято решение о выборе в качестве платформы, уже имеющийся SIP-клиент, и адаптировать его под свои нужды.

К сожалению, примеров проектов с открытым исходным кодом, реализующим функцию софтофона для ОС Андроид, оказалось относительно мало и практически все они обладают двумя значительными недостатками:

1. Перестали развиваться.
2. Имеют устаревший интерфейс.

Рассмотрим связки библиотек, реализующих SIP-стэк и открытых проектов, основанных на их базе.

Csipsimple (рисунок 1.11) использующий PjSip распространяется по лицензии GNU GPL v3.

Рисунок 1.11 – SIP-клиент Csipsimple

Отличается большим количеством настроек. Работает по протоколам UDP, TCP, TLS для SIP и SRTP/ZRTP для медиа. Поддерживает широкоформатные кодеки. Однако его интерфейс устарел, а последний коммит в репозиторий датируется 20-ым марта 2015 года. Это значит, что проект вероятно больше не будет развиваться усилиями сообщества, не будут исправляться ошибки и код не будет адаптироваться под новые версии операционной системы.

Проект основан на PjSip, бесплатной библиотеке для мультимедиа коммуникаций с открытым исходным кодом. Положительными особенностями является малый размер и наличие документации.

Sipdroid (рисунок 1.12) использующий MjSip распространяется по лицензии GNU GPL v3.

Рисунок 1.12 – SIP-клиент Sipdroid

Поддерживает широкоформатные кодеки. Однако его интерфейс устарел, а последний коммит в репозиторий датируется 14-ым июля 2015 года. Это значит, что проект вероятно больше не будет развиваться усилиями сообщества, не будут исправляться ошибки и код не будет адаптироваться под новые версии операционной системы.

Проект основан на MjSip, бесплатной библиотеке для мультимедиа коммуникаций с открытым исходным кодом. Положительными особенностями является простота настройки и подключения.

Imsdroid (рисунок 1.13), использующий MjSip распространяется по лицензии GNU GPL v3.

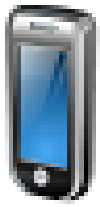

Рисунок 1.13– SIP-клиент Imsdroid

Позиционировался как SIP-клиент для профессионалов. Однако его интерфейс устарел, а последний коммит в репозиторий датируется 30ым июля 2015 года. Это значит, что проект вероятно больше не будет развиваться усилиями сообщества, не будут исправляться ошибки и код не будет адаптироваться под новые версии операционной системы.

Проект основан на doubango, бесплатной библиотеке для мультимедиа коммуникаций с открытым исходным кодом. По сути, является фрэймверком предназначенным как для встраиваемых и настольных систем. Используется на стороне сервера в таком популярном проекте WebRTC-телефонии как sipML5.

На фоне описанных SIP-клиентов выделяется Linphone (рисунок 1.14).

Рисунок 1.14 – SIP-клиент Linphone

Linphone – кроссплатформенный программный клиент IP-телефонии в стандарте SIP с открытым исходным кодом, распространяемый по лицензии GNU GPL v3. Существует клиент для Android, IOS, Windows, Linux. Недавно был улучшен интерфейс, еженедельные коммиты. Это значит, что проект, вероятно, будет развиваться, будут исправляться ошибки и код будет адаптироваться под новые версии операционной системы.

Приложение для Андроид написано на Java, с использованием библиотек на СИ.

Проект основан на belle-sip, бесплатной библиотеке для имплементации SIP. Написан на языке СИ и является продуктом компании BelledonneCommunications.

Так же для работы с медиа используется mediastreamer2 – бесплатная библиотека для мультимедиа коммуникаций с открытым исходным кодом, написан на языке СИ и является продуктом компании BelledonneCommunications.

Linphone имеет широкий функционал, использующий возможности SIP. Доступны аудио и видео звонки и передача мгновенных сообщений.

В 2015 году «ЗАО Рубцовск» уже пробовали данный sip-клиент для работы со своим сервисом.

Исходя из всего вышесказанного именно этот проект был выбран в качестве основы мобильного приложения «Русский телефон». Это основополагающее проектное решение окажет влияние на все элементы системы.

Помимо работы с SIP протоколом и медиа, приложению потребуется интенсивный обмен с сервером по API сервиса «Русский телефон».

API сервиса «Русский телефон» разработано по стандарту RESTfull.

REST (сокр. от англ. Representational State Transfer – «передача состояния представления») – архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой

гипермедиа-системы.

Данные передаются в формате JSON, поэтому фреймворк предоставляющий удобный способ взаимодействия значительно сократил бы время разработки, снизил бы затраты на поддержку и повысил надежность сетевого обмена.

Как ни странно, несмотря на то, что операционная система Андроид известна с 2008 года, выбор способа отправки сетевых запросов до сих пор актуален. Этому вопросу до сих пор посвящаются статьи и выступления. Рассмотрим имеющиеся возможности для последующего принятия проектного решения.

Сегодня почти все приложения используют HTTP/HTTPS запросы как своеобразный транспорт для своих данных.

У Android-разработчиков есть много причин, чтобы сделать выбор в пользу сторонних библиотек, вместо уже встроенных API, таких как HttpURLConnection или Apache Client, например:

1. Возможность отмены сетевого вызова.
2. Параллельное исполнение запросов.
3. Пул соединений и переиспользование уже существующих сокет-соединений.
4. Локальное кэширование ответов сервера.
5. Простой интерфейс асинхронной работы, чтобы избежать блокировки главного или UI потоков.
6. Удобная обертка над REST API.
7. Политика повторов и задержек.
8. Эффективная загрузка и трансформация изображений.
9. Сериализация в виде JSON.
10. Поддержка SPDY, http/2.

В самом начале в Android имелись только 2 HTTP клиента: HttpURLConnection и Apache HTTP Client. Согласно официальному посту в блоге Google, HttpURLConnection имел несколько багов в ранних версиях

Android. В то же время, Google не хотел развивать и переходить на Apache HTTP Client.

До «Android Froyo» большинство разработчиков предпочитали использовать различные клиенты, основываясь на версии ОС.

Это классический пример фрагментации Android. В 2013 году Square обратила внимание на эту проблему, когда выпускала OkHttp. OkHttp была создана для прямой работы с верхним уровнем сокетов Java, при этом не используя какие-либо дополнительные зависимости. Она поставляется в виде JAR-файла, так что разработчики могут использовать ее на любых устройствах с JVM. Для упрощения перехода на их библиотеку, Square имплементировали OkHttp используя интерфейсы HttpUrlConnection и Apache client. Схема имплементации показана на рисунке (рисунок 1.15) [18].

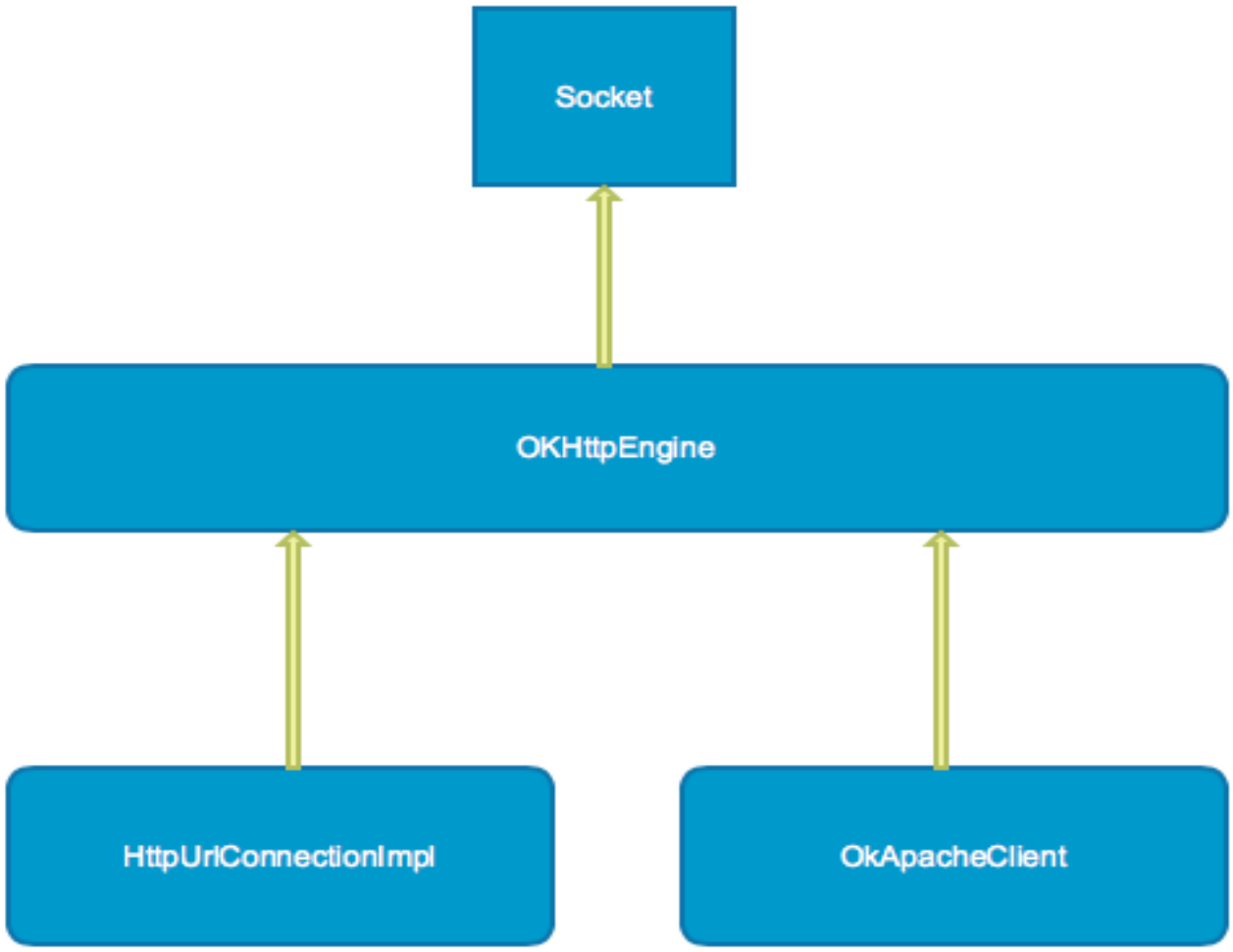

Рисунок 1.15 – Схема OkHttp

OkHttp получила большое распространение и поддержку сообществом, и, в конце-концов, Google решили использовать версию 1.5 в Android 4.4 (KitKat). В июле 2015 Google официально признала AndroidHttpClient, основанный на Apache, устаревшим, вместе с выходом Android 5.1 (Lolipop).

Сегодня OkHttp поставляется со следующим огромным набором функций:

1. Поддержка HTTP/2 и SPDY позволяет всем запросам, идущим к одному хосту, делиться сокетом.
2. Объединение запросов уменьшает время ожидания (если SPDY не доступен).
3. Прозрачный GZIP позволяет уменьшить вес загружаемой информации.
4. Кэширование ответов позволяет избежать работу с сетью при повторных запросах.
5. Поддержка как и синхронизированных блокирующих вызовов, так и асинхронных вызовов с обратным вызовом (callback).

HTTP библиотека от Google под названием Volley, предоставляет нам следующие преимущества:

6. Автоматическое планирование сетевых запросов.
7. Множество параллельных сетевых соединений.
8. Прозрачное кэширование в памяти и на диске, в соответствии со стандартной кэш-согласованностью.
9. Поддержка приоритизации запросов.
10. Отмена API запросов. Вы можете отменить как один запрос, так и целый блок.
11. Простота настройки, например, для повторов и отсрочек.
12. Строгая очередность, которая делает легким корректное заполнение данными, полученными асинхронно из сети, интерфейса пользователя.
13. Инструменты отладки и трассировки

Все, что ни есть в Volley, находится на вершине HttpUrlConnection. Для

получения JSON или изображения, Volley имеет специальные абстракции, такие как ImageRequest и JsonObjectRequest, которые помогают в автоматическом режиме конвертировать полезную нагрузку HTTP.

Архитектуру Volly демонстрирует следующая схема включающая разбиение на потоки. (рисунок 1.16).

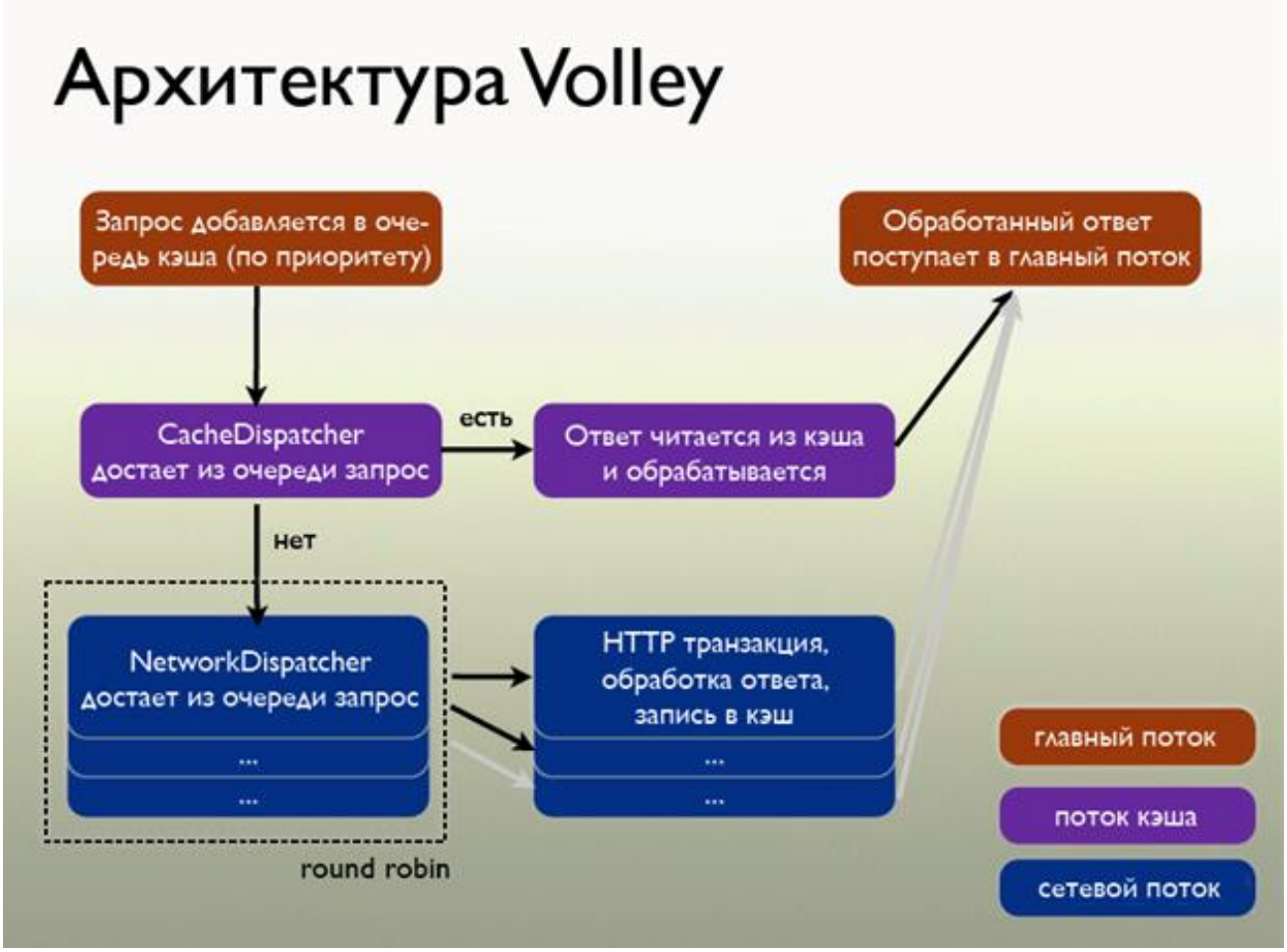

Рисунок 1.16 – Архитектура Volly.

В большинстве случаев OkHttp будет действовать быстрее за счет использования большего количества потоков.

HTTP клиенты продолжили развиваться для поддержки приложений с большим количеством картинок, особенно тех, кто поддерживает бесконечную прокрутку и трансформацию изображений. В то же время, REST API стал стандартом в индустрии, и каждый разработчик имел дело с такими типовыми задачами как сериализация в/из JSON и преобразование REST-вызовов в

интерфейсы Java. Не прошло много времени, как появились библиотеки, решающие эти задачи.

Retrofit – типобезопасный HTTP Android клиент для взаимодействия с REST-интерфейсами.

Retrofit предоставляет некий мост между Java кодом и REST-интерфейсом. Он позволяет быстро включить в ваш проект HTTP API интерфейсы, и генерирует самодокументирующуюся реализацию.

В дополнение ко всему, Retrofit поддерживает конвертацию в JSON, XML, протокол буферов (protocol buffers).

Проанализировав преимущества, мощность и лаконичность получающегося в результате кода, было принято решения использовать Retrofit, базирующийся на OkHttp.

Так как в задачи дипломной квалификационной работы не входит изменение серверной структуры работы сервиса, было решено использовать имеющуюся инфраструктуру и работающий программный комплекс.

Для обоснования этого решения приведу краткое описание уже работающего аппаратного и программного обеспечения.

Сервис «Русский телефон» развился из сервиса IP телефонии. Важной частью системы IP телефонии являются модульные маршрутизаторы, обеспечивающие связь с городской и офисной АТС.

В ЗАО Рубцовск маршрутизаторы представлены оборудованием «Cisco 1760» и «Cisco 1750» отличающиеся выдающейся надёжностью.

Cisco – американская транснациональная компания, разрабатывающая и продающая сетевое оборудование, предназначенное в основном для крупных организаций и телекоммуникационных предприятий.

Одна из крупнейших в мире компаний, специализирующихся в области высоких технологий. Изначально занималась только корпоративными маршрутизаторами.

HTTP запросы обслуживаются сервером Apache работающим на базе двухъядерного процессора Intel Pentium 2.66GHz с тремя гигабайтами

оперативной памяти. На этом же аппаратном обеспечении работает база данных.

Основой сервиса является сервер Астериск, работающий на базе Intel Celeron CPU 1.70GHz с 265 мегабайтами оперативной памяти.

Оптимизация операционной системы и самого сервера позволяет получить должную производительность, на экономичном аппаратном обеспечении.

Терминация звонка, о которой говорилось выше и биллинг обрабатываются на сервере, базирующемся на Intel Core2 Duo CPU E6750 2.66GHz с двумя гигабайтами оперативной памяти.

Этот программно-аппаратный комплекс хорошо зарекомендовал себя в работе сервиса и имеет запас производительной мощности.

Возможность модернизации позволяет масштабировать систему на большее число пользователей. Это является доказательством, что правильным проектным решением, будет использование имеющейся инфраструктуры, для дальнейшей интеграции мобильного приложения в качестве клиента.

2 Проектная часть

2.1 Разработка функционального обеспечения

Построенная функциональная модель «как есть» («AS-IS») в подразделе «Анализ функционирования объекта исследования» аналитической части и её декомпозиция приводят к необходимости построения модели «как должно быть» (TO-BE).

Задачей описания модели TO-BE является нахождение мер блокирования отрицательного влияния плохих бизнес-факторов, полученных при анализе данных. Полученная модель представлена на рисунке 2.1.

На данной модели видно, что мобильное приложение будет содержать функционал совершения всех видов звонков, будет предоставлять доступ к информации о балансе, позволять пополнить счет и зарегистрировать нового пользователя.

На рисунке 2.2 представлена декомпозиция функций «Как должно быть».

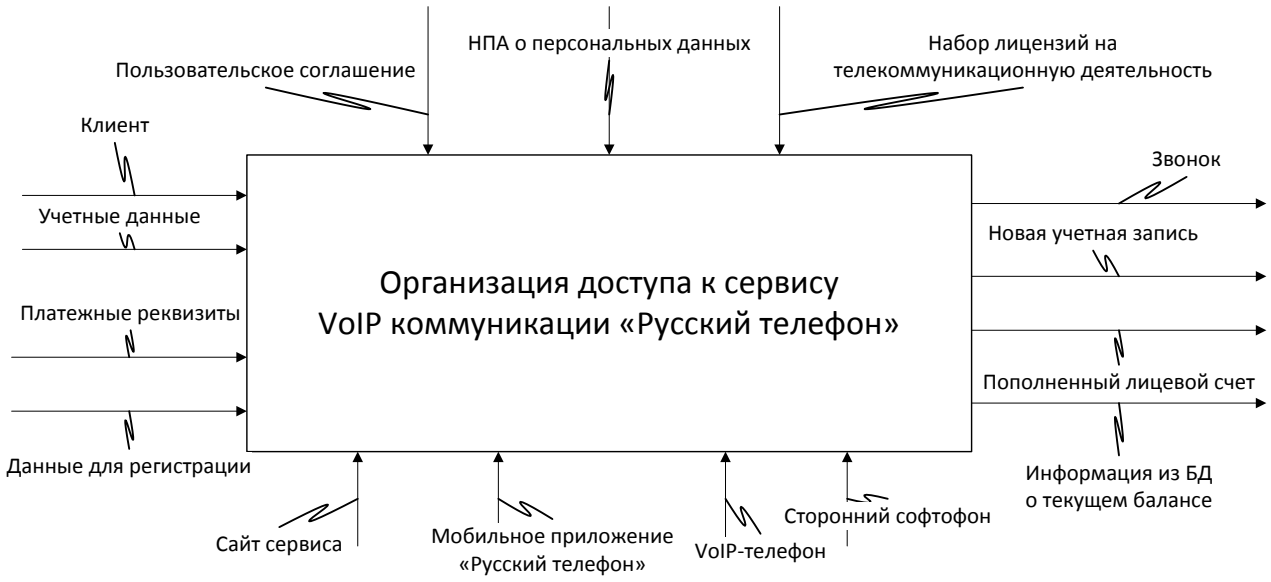

Рисунок 2.1 – Контекстная диаграмма IDEF0 TO-BE «Организация доступа к сервису VoIP коммуникации «Русский телефон»»

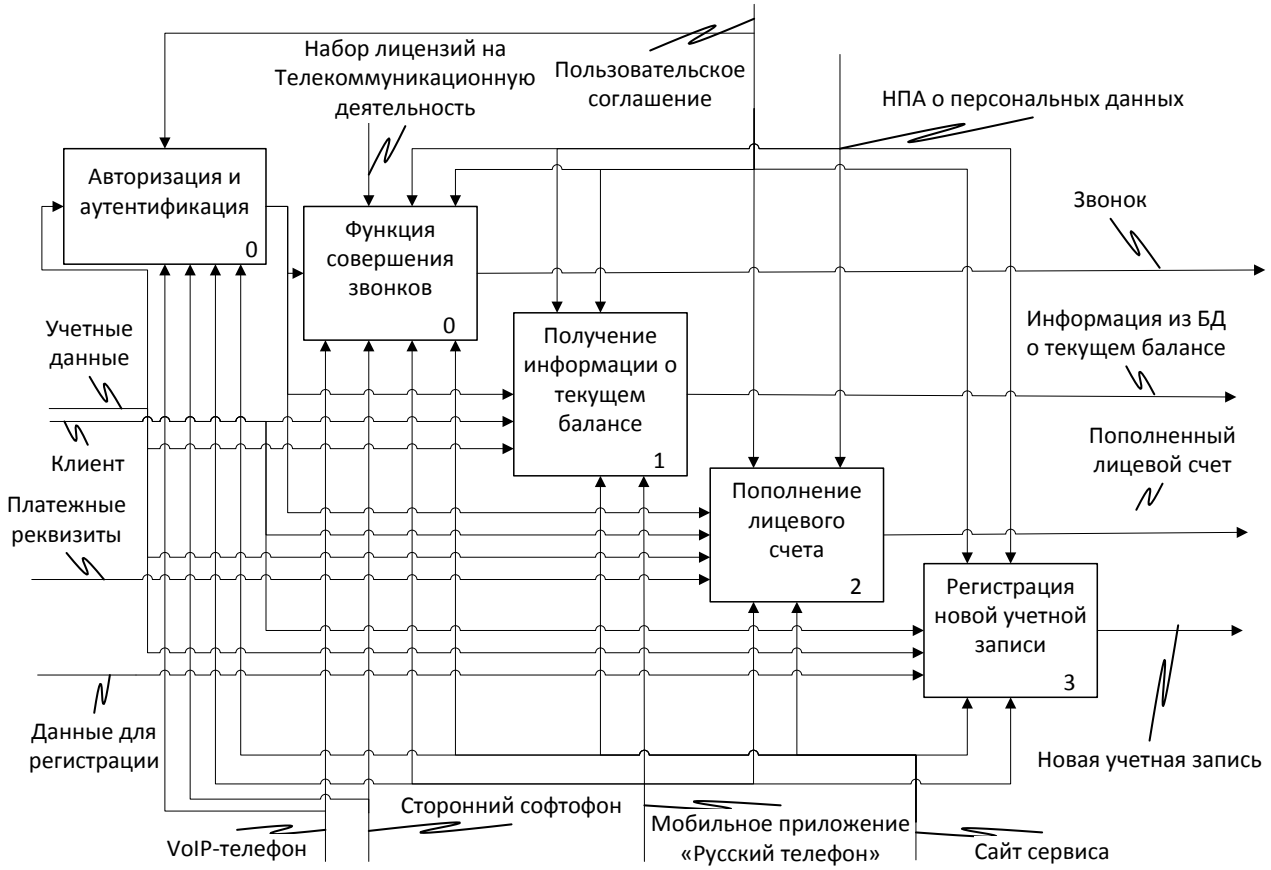

Рисунок 2.2 – Детализированная диаграмма IDEF0 TO-BE «Организация доступа к сервису VoIP коммуникации «Русский телефон»»

2.2 Разработка информационного обеспечения

2.2.1 Используемые классификаторы и системы кодирования

Классификатор, или классификационная схема (от лат. Classis – разряд и facere – делать) – систематизированный перечень наименований объектов, каждому из которых в соответствие дан уникальный код. Классификация объектов производится согласно правилам распределения заданного множества объектов на подмножества (классификационные группировки) в соответствии с установленными признаками их различия или сходства. Классификатор является стандартным кодовым языком документов, финансовых отчётов и автоматизированных систем.

Классификаторы разрабатываются как на уровне отдельных предприятий (организаций), так и на уровне государств. Существуют следующие уровни

классификаторов:

– международные – стандартные классификаторы, используемые по всему миру;

– межгосударственные – классификаторы, используемые в рамках экономических союзов и других межгосударственных объединений: например, классификаторы, используемые в ЕС, СНГ и т. д.;

– национальные, или межотраслевые – классификаторы, используемые в пределах государства;

– отраслевые – классификаторы, используемые в рамках одной отрасли;

– системные – классификаторы, принятые отдельным предприятием (организацией) для применения в рамках своей автоматизированной системы. Они содержат информацию, необходимую для решения задач в конкретной АС и отсутствующую в национальном или отраслевом классификаторе.

В ходе работы используются классификаторы компании Google в отношении стандартного для ОС Андроид способа хранения информации в реляционной базе данных SQLite, а также используемая на серверах ЗАО «Рубцовск» база «Информикс».

При реализации проекта для обеспечения уникальности объектов в пределах класса применяется порядковая система кодирования объектов. При этом все множество объектов предварительно не упорядочивается. В результате чего всю работу по обеспечению максимальной скорости поиска объекта и определения его принадлежности во время сложной логической обработки с использованием имеющихся данных берет на себя СУБД [9].

Таким образом, некоторые таблицы имеют в своем составе автоинкрементное поле – идентификатор. Это позволяет не контролировать процессы генерации и присвоения уникальных кодов для записей программно, а полностью предоставить этот процесс под контроль СУБД.

Все даты представляются в стандартном для России формате «ДД.ММ.ГГГГ», где ДД – день месяца от 0 до 31, ММ – месяц от 0 до 12, ГГГГ – год в четырехзначном представлении.

Вся информация в базе данных представлена в кодировке UTF-8 – распространённой кодировке, реализующей представление Юникода, совместимое с 8-битным кодированием текста, позволяющая представить знаки практически всех письменных языков.

2.2.2 Характеристика нормативно-справочной и входной оперативной информации

Вся справочная информация единого информационного пространства может быть представлена в виде объектов, представленных на основе классов, со следующими характеристиками: идентификатор объекта, собственно справочную информацию и ряд конкретных дополнительных свойств. Методами у таких объектов могут быть добавление новой справочной информации, изменение или удаление существующей.

Все методы объектов реализованы с помощью программных средств организации, а также с использованием существующей системы управления содержимым сайта компании.

Для исследуемой предметной области можно выделить следующие основные справочники:

- абоненты;
- звонки;
- стоимость направлений.

2.2.3 Характеристика результатной информации

Так как приложение будет иметь клиент-серверную архитектуру, в качестве результатной информации работы сервиса «Русский телефон» (сервера) для мобильного приложения (клиента) будут выступать:

- информация об аутентификации пользователя;

– информация о балансе;

– информация о текущем звонке (статистическая информация состояния соединения, во время совершения звонка);

– аудио поток как компонент разговора.

Все перечисленные результатные сведения будут передаваться мобильному устройству для дальнейшей обработки и представления в удобном для пользователя виде.

Не будем рассматривать технологию и формат передачи аудио информации. Эта работа осуществляется ядром выбранной платформы и в данном проекте не меняется.

В качестве формата для передачи данных мобильному устройству могут быть такие форматы, как: JSON, XML, YAML и INI.

Основными требованиями к формату стали: удобство редактирования, скорость парсинга, скорость сериализации, размер в сериализованном виде, распространённость и поддержка редакторами.

Результаты сравнения форматов представлены в таблице 2.1.

В итоге, исходя из показателей, в качестве формата для передачи мобильному устройству выбран – JSON.

Мобильное приложение, посредством POST-запроса, будет передавать необходимые параметры серверу, а в качестве результатной информации будет получать ответ в формате JSON.

Использование формата данных JSON позволит сократить мобильный трафик и упростить обработку результатной информации за счет того, что передаваемый JSON-объект содержит только необходимые данные, в отличии от использования например, веб-сайта, когда клиенту передается весь исходный код страницы, включая данные разметки, картинки, компоненты и т.д.

Таблица 2.1 – Сравнение форматов данных

Свойство	JSON	XML	YAML	INI
Человекочетаемость	3	1	4	5
Удобство редактирования	3	1	4	5
Произвольная иерархия	3	3	3	1
Простота реализации	3	2	1	5
Скорость парсинга/сериализации	3	1	1	5
Размер в сериализованном виде	3	1	4	5
Поддержка поточной обработки	0	0	5	5
Бинарная безопасность	3	0	0	0
Распространённость	5	5	3	3
Поддержка редакторами	5	5	3	5
Поддержка языками программирования	5	5	3	5

2.2.4 Структура результатной информации

В качестве результатной информации на мобильное устройство будет отправляться JSON-объект.

JSON – текстовый формат обмена данными, основанный на JavaScript.

Несмотря на происхождение от JavaScript (точнее, от подмножества языка стандарта ECMA-262 1999 года), формат считается языконезависимым и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON.

За счёт своей лаконичности по сравнению с XML, формат JSON может быть более подходящим для сериализации сложных структур. Если говорить о веб-приложениях, в таком ключе он уместен в задачах обмена данными как между браузером и сервером (AJAX), так и между самими серверами (программные HTTP-интерфейсы).

JSON-текст представляет собой (в закодированном виде) одну из двух структур:

– набор пар ключ: значение. В различных языках это реализовано как объект, запись, структура, словарь, хэш-таблица, список с ключом или ассоциативный массив. Ключом может быть только строка, значением – любая форма;

– упорядоченный набор значений. Во многих языках это реализовано как массив, вектор, список или последовательность.

Это универсальные структуры данных: как правило, любой современный язык программирования поддерживает их в той или иной форме. Они легли в основу JSON, так как он используется для обмена данными между различными языками программирования.

В качестве значений в JSON используются структуры:

– объект – это неупорядоченное множество пар ключ:значение, заключённое в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары ключ-значение отделяются друг от друга запятыми;

– массив (одномерный) – это упорядоченное множество значений. Массив заключается в квадратные скобки «[ ]». Значения разделяются запятыми;

– значение может быть строкой в двойных кавычках, числом, объектом, массивом, одним из литералов: true, false или null. Т.о. структуры могут быть вложены друг в друга;

– строка – это упорядоченное множество из нуля или более символов юникода, заключенное в двойные кавычки. Символы могут быть указаны с использованием escape-последовательностей, начинающихся с обратной косой черты «\» [6].

Структура JSON должна удовлетворять заданным требованиям: человекопонятность, удобство редактирования, удобство для парсинга.

Передаваемых данных должно быть достаточно для выполнения текущей задачи. Например, запрос к web-серверу для инициации процедуры регистрации нового пользователя должен содержать:

– идентификатор капчи (индекс картинки с текстом);
– введенный пользователем проверочный код капчи;
– промо-код (необязательное поле);
– номер мобильного телефона;
– адрес электронной почты (необязательное поле);
– фамилию;
– имя;
– отчество (необязательное поле).

В конце процедуры регистрации сервер отправит ответ содержащий логин и пароль новой учетной записи. Приватность этих данных сохраняется благодаря протоколу HTTPS.

2.2.5 Информационная модель и ее описание

Проблемная область – взаимосвязанная совокупность управляемых объектов предприятия (предметная область), субъектов управления, автоматизируемых функций управления и программно-технических средств их реализации.

Модель – некоторая система, имитирующая структуру или функционирование исследуемой проблемной области, отвечающей основному требованию – адекватности этой области.

С точки зрения объектов моделирования необходимо различать модели предметной области и модели базы данных. Эти модели взаимосвязаны, поскольку представляют собой образы одного и того же оригинала – некоторого множества предметов реального мира, информацию о которых предполагается хранить и обрабатывать с помощью проектируемой БД.

Модель предметной области скорее ассоциируется с неформальным уровнем семантического моделирования, а модель базы данных – с формализованным уровнем системы (и в частности, СУБД). В идеале целью семантического моделирования является формирование систематического

основания для хорошо формализованного процесса проектирования базы данных.

Требования, предъявляемые к базам данных, и, в частности, к способам описания данных:

– описания должны быть понятны пользователю, не проектировавшему базу;

– однажды принятые способы представления данных должны допускать присоединение новых элементов данных без изменения существующих схем данных и прикладных программ;

– СУБД должны позволять эффективно обрабатывать произвольные запросы к базе данных.

Эти требования отражают, с одной стороны, точку зрения пользователя, для которой характерны требования высокой степени общности и широты представления, позволяющих ему получить достаточно сведений без затраты значительных временных или интеллектуальных ресурсов. С другой стороны – точку зрения администратора, выполняющего проектирование и оптимизацию системы баз данных, что предполагает высокую степень детализации и формализации, обеспечивающих обоснованность технических решений, а также возможность автоматизации проектирования [8].

Начальной стадией проектирования системы баз данных является построение семантической модели предметной области, которая базируется на анализе свойств и природы объектов предметной области и информационных потребностей будущих пользователей разрабатываемой системы. Эту стадию принято называть концептуальным проектированием системы, а ее результат – концептуальной моделью предметной области. Объектом моделирования здесь является предметная область будущей системы. Этой стадии соответствуют также термины «инфологическое проектирование» и «инфологическая модель» [10].

Такие модели обобщенно представляют информационные потребности пользователей создаваемой системы в части использования хранимых данных и

по существу являются средством коммуникации как разработчиков, так и пользователей на разных стадиях жизненного цикла базы данных.

К инфологическим моделям относятся различные компоненты, по-разному и разными средствами отражающие предметную область. Помимо наиболее известного описания объектов и связей между ними (модель «сущность-связь») к инфологическому уровню описания предметной области можно отнести следующие компоненты:

– систему атрибутов и средств описания предметной области. Например, логические (алгоритмические) связи между показателями или лингвистические свойства языка (синонимию, синтаксис и т. д.), используемого для вербального представления объектов;

– ограничения целостности, определяющие допустимость значения отдельных полей и взаимосвязей как на уровне семантики содержимого БД, так и ее физической структуры (отдельных файлов данных и взаимосвязей между ними);

– описание информационных потребностей пользователей, например, в виде типовых запросов, отражающих процедурные особенности обращения к данным.

Моделирование предметной области на основе модели «Сущность-связь» также называется ER-моделирование. Оно базируется на использовании графических диаграмм, как простого (привычного), наглядного и в то же время информативного и многоаспектного способа отображения компонентов проекта.

Сущность, с помощью которой моделируется класс однотипных объектов, определяется как «предмет, который может быть четко идентифицирован». Так же как каждый объект уникально характеризуется набором значений свойств, сущность должна определяться таким набором атрибутов, который позволял бы различать отдельные экземпляры сущности.

Каждый экземпляр сущности должен быть отличим от любого другого экземпляра той же сущности. Уникальным идентификатором сущности может

являться атрибут, комбинация атрибутов, комбинация связей или комбинация связей и атрибутов, однозначно отличающая любой экземпляр сущности от других экземпляров сущности того же типа.

Сущность имеет имя, уникальное в пределах модели. При этом имя сущности – это имя типа, а не некоторого конкретного экземпляра.

Сущности подразделяются на сильные и слабые. Сущность является слабой, если ее существование зависит от другой сущности – сильной по отношению к ней.

Семантическую основу ER-модели составляют следующие предположения:

– часть реального мира (совокупность взаимосвязанных объектов), сведения о которых должны быть помещены в базу данных, может быть представлена как совокупность сущностей;

– каждая сущность обладает характеристическими свойствами (атрибутами), отличающими ее от других сущностей и позволяющими ее идентифицировать;

– сущности можно классифицировать по типам сущностей: каждый экземпляр сущности (представляющий некоторый объект) может быть отнесен к классу – типу сущностей, каждый экземпляр которого обладает общими для них и отличающими их от сущностей других классов свойствами;

– систематизация представления, основанная на классах, в общем случае предполагает иерархическую зависимость типов сущность типа A является подтипом сущности B, если каждый экземпляр типа A является экземпляром сущности типа B;

– взаимосвязи объектов могут быть представлены как связи – сущности, которые служат для фиксирования (представления) взаимозависимости двух или нескольких сущностей.

Одна из основных целей семантического моделирования состоит в том, чтобы результаты анализа предметной области были отражены в достаточно простом, наглядном, но в то же время формализованном и достаточно

информативном виде. В этом смысле ER-диаграмма является очень удачным решением. В ней сочетаются функциональный и информационный подходы, что позволяет представлять, как совокупность выполняемых функций, так и отношения между элементами системы, задаваемые структурами данных. При этом графическая форма позволяет отобразить в компактном виде типологию и свойства сущностей и связей за счет наглядных условных обозначений, а формализмы, положенные в основу ER-диаграмм, позволяют использовать на следующем шаге проектирования логической структуры базы данных строгий аппарат нормализации.

Сущности. Каждый тип сущности в ER-диаграммах представляется в виде прямоугольника, содержащего имя сущности. В качестве имени обычно используются существительные (или обороты существительного) в единственном числе.

Связь – графически изображаемая ассоциация, устанавливаемая между сущностями. Каждый тип связи на ER-диаграмме отображается в виде ромба с именем связи внутри.

Как и в реляционных схемах баз данных, в ER-диаграммах вводится понятие нормальных форм, причем их смысл очень близок смыслу реляционных нормальных форм. Приведем краткие и неформальные определения трех первых нормальных форм.

В первой нормальной форме ER-диаграммы устраняются повторяющиеся атрибуты или группы атрибутов, т. е. производится выявление неявных сущностей, «замаскированных» под атрибуты.

Во второй нормальной форме устраняются атрибуты, зависящие только от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

В третьей нормальной форме устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности [21].

Задачей следующей стадии проектирования системы базы данных

является выбор подходящей СУБД и отображение в ее среду (структуру данных) спецификаций инфологической модели предметной области. Другими словами, модель предметной области разрабатываемой системы должна быть представлена в терминах модели данных концептуального уровня выбранной конкретной СУБД. Эту стадию называют логическим (или даталогическим) проектированием базы данных, а ее результатом является концептуальная схема базы данных, включающая определение всех информационных элементов (единиц) и связей, в том числе задание типов, характеристик и имен.

Существует много вариантов отображения инфологической модели предметной области в даталогическую модель базы. Здесь следует учитывать влияние двух следующих значимых факторов, связанных с практикой разработки базы данных.

Во-первых, связи предметной области могут отображаться двумя путями: как декларативным – в логической схеме, так и процедурным – отработкой связей через программные модули, обрабатывающие (связывающие) соответствующие хранимые данные.

Во-вторых, существенным фактором может оказаться характер обработки информации. Например, частые обращения к совместно обрабатываемым данным, очевидно, предполагают их совместное хранение, а данные (особенно большого объема), к которым обращаются редко, целесообразно хранить отдельно от часто используемых.

Стадия физического проектирования базы данных в общем случае включает:

– выбор способа организации базы данных;

– разработку спецификации внутренней схемы средствами модели данных ее внутреннего уровня;

– описание отображения концептуальной схемы во внутреннюю.

Важно заметить, что в отличие от ранних СУБД, многие современные системы не предоставляют разработчику какого-либо выбора на этой стадии.

Способ хранения базы данных определяется механизмами СУБД

автоматически «по умолчанию» на основе спецификаций концептуальной схемы базы данных, и внутренняя схема в явном виде в таких системах не используется. Следует также отметить, что внешние схемы базы данных обычно конструируются на стадии разработки приложений [20].

На основе проведенного анализа, изученной теории создания информационных систем и поставленных целей и задач можно перейти непосредственно к проектированию.

Данные в основном хранятся на сервере и выдаются клиентским приложениям в минимальном необходимом объеме при необходимости.

Самым большим объемом данных, требующих структурированного доступа, в приложении является телефонная книга.

В операционной системе Андроид, эта информация хранится в реляционной базе данных SQLite и имеет следующую структуру (рисунок 2.3).

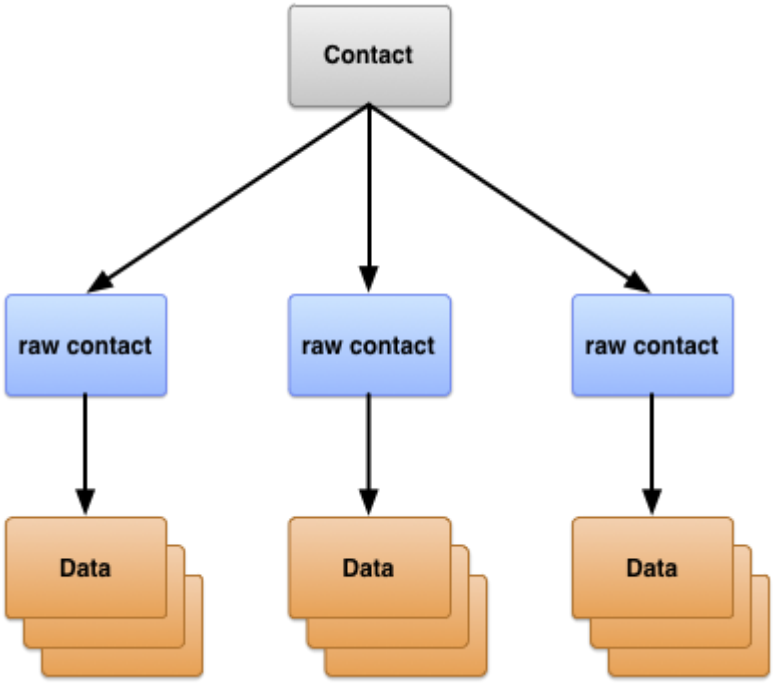

Рисунок 2.3. – Структура таблицы поставщика контактов.

В качестве имен этих трех таблиц обычно используются названия соответствующих классов-контрактов. Эти классы определяют константы для URI контента, названий столбцов и значений в столбцах этих таблиц.

Таблица ContactsContract.Contacts - строки в этой таблице содержат данные о разных пользователях, полученные путем агрегации строк необработанных контактов.

Таблица ContactsContract.RawContacts - строки в этой таблице содержат сводные данные о пользователе, относящиеся к пользовательскому аккаунту и его типу.

Таблица ContactsContract.Data - строки в этой таблице содержат сведения о необработанных контактах, такие как адреса эл. почты или номера телефонов.

Другие таблицы, представленные классами-контрактами в ContactsContract, представляют собой вспомогательные таблицы, которые поставщик контактов использует для управления своими операциями или поддержки определенных функций, имеющихся в приложении устройства «Контакты» или приложениях для телефонной связи.

Рассмотрим подробнее структуру ContactsContract.Data представленную в таблице 2.2.

Таблица 2.2 – ContactsContract.Data

Тип данных	Наименования столбца	Доступ пользователя	Описание
long	_ID	read-only	Идентификатор для связи таблиц (контакт)
String	MIMETYPE	read/write-once	Иденификатор типа хранящихся данных. Определяет, что должно хранится в столбцах DATA1-DATA15
long	RAW_CONTACT_ID	read/write-once	Идентификатор для связи таблиц (RAW-контакт)
int	IS_PRIMARY	read/write	Является ли первичной записью

Продолжение таблицы 2.2.

Тип данных	Наименования столбца	Доступ пользователя	Описание
int	IS_SUPER_PRIMARY	read/write	Является ли записью «по умолчанию»
int	DATA_VERSION	read-only	Номер версии этой записи
Any type	DATA1	read/write	В этих столбцах могут хранится любые данные. Тип данных зависит от MIMETYPE, может быть задан собственный MIMETYPE со своим набором столбцов.
	DATA2		
	DATA3		
	DATA4		
	DATA5		
	DATA6		
	DATA7		
	DATA8		
	DATA9		
	DATA10		
	DATA11		
	DATA12		
	DATA13		
	DATA14		
	DATA15		
Any type	SYNC1	read/write	Данные для синхронизации.
	SYNC2		
	SYNC3		
	SYNC4		

Как видно из приведенной схемы и таблицы, организация хранения данных контактов в Андроид имеет сложную структуру, которую необходимо учитывать при разработке приложения.

Остальные данные, мобильное приложение будет получать от серверной части. Данные в серверной части хранятся в реляционной базе данных. База данных была создана для работы IP телефонии, затем изменена для нужд VoIP сервера. База данных построена на базе СУБД «Информикс» от IBM, удовлетворяет всем требованиям сервиса и не требует модиффикации.

На рисунке 2.4 представлена логическая структура базы данных, взятая из документации.

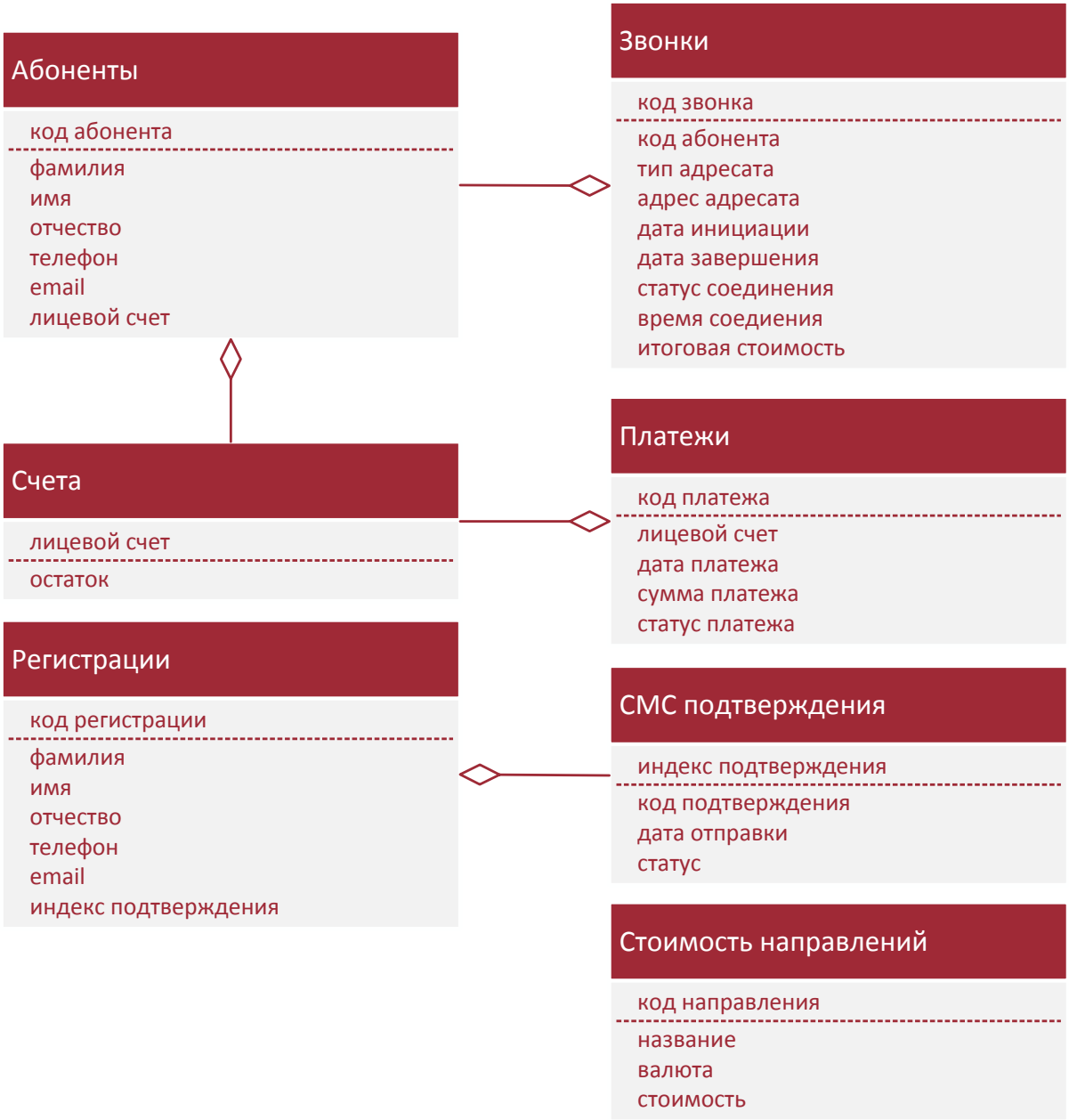

Рисунок 2.4 – Логическая модель базы данных

Таблица абонентов, содержит информацию о клиентах. Ключевым полем

является «код абонента», позволяющий однозначно идентифицировать учетную запись.

Таблица 2.3 – Таблица Абоненты/ABONENTS из БД

Псевдоним	Имя	Тип
код абонента	ID	INT
фамилия	LAST_NAME	TEXT
имя	FIRST_NAME	TEXT
отчество	MIDDLE_NAME	TEXT
телефон	PHONE	TEXT
email	EMAIL	TEXT
лицевой счет	SETLEMENT_ACCOUNT	INT

Поле «лицевой счет» агрегирует (1-1) элемент таблицы «Счета»

Таблица звонков, содержит информацию о совершенных через сервис вызовах. Ключевым полем является «код регистрации», позволяющий однозначно идентифицировать звонок.

Таблица 2.4 – Таблица Звонки/CALLS из БД

Псевдоним	Имя	Тип
код звонка	ID	INT
код абонента	ABONENT_ID	INT
тип адресата	TYPE	TEXT
адрес адресата	ADDRESS	TEXT
дата инициации	DATE_START	DATE
дата завершения	DATE_FINISH	DATE
статус соединения	STATUS	TEXT
время соединения	TIME	INTERVAL
итоговая стоимость	COST	FLOAT

Поле «код абонента» агрегирует (1-1) элемент таблицы «Абоненты».

Таблица лицевых счетов, содержит информацию о текущем балансе. Ключевым полем является «лицевой счет», позволяющий однозначно идентифицировать элемент.

Таблица 2.5 – Таблица Счета/ACCOUNTS из БД

Псевдоним	Имя	Тип
лицевой счет	ID	INT
остаток	BALANCE	FLOAT

Таблица платежей, содержит информацию о пополнениях лицевого счета. Ключевым полем является «код платежа», позволяющий однозначно идентифицировать платеж.

Таблица 2.6 – Таблица Платежи/PAYMENTS из БД

Псевдоним	Имя	Тип
код платежа	ID	INT
лицевой счет	SETLEMENT_ACCOUNT	INT
дата платежа	DATE	DATE
сумма платежа	PAYMENT	FLOAT
статус платежа	STATUS	TEXT

Поле «лицевой счет» агрегирует (1-1) элемент таблицы «Счета».

Таблица регистраций, содержит информацию о попытках зарегистрироваться в сервисе. При подтверждении телефона, эти данные станут основой записи в таблице абонентов. Ключевым полем является «код регистрации», позволяющий однозначно идентифицировать элемент.

Таблица 2.7 – Таблица Регистрации/REG_TEMP из БД

Псевдоним	Имя	Тип
код регистрации	ID	INT

Продолжение таблицы 2.7

Псевдоним	Имя	Тип
фамилия	LAST_NAME	TEXT
имя	FIRST_NAME	TEXT
отчество	MIDDLE_NAME	TEXT
Телефон	PHONE	TEXT
email	EMAIL	TEXT
индекс подтверждения	CONFIRMATION_ID	INT

Поле «индекс подтверждения» агрегирует (1-1) элемент таблицы «СМС подтверждения».

Таблица кодов подтверждения, содержит информацию о попытках подтвердить телефон, указанный при регистрации. Ключевым полем является «индекс подтверждения».

Таблица 2.8 – Таблица СМС подтверждения/PHONE_CONFIRM из БД

Псевдоним	Имя	Тип
индекс подтверждения	ID	INT
код подтверждения	CONFIRMATION_CODE	TEXT
дата отправки	DATE	DATE
статус	STATUS	TEXT

Таблица стоимости направлений, содержит информацию о цене минуты разговора. Ключевым полем является «код направления», позволяющий однозначно идентифицировать направление звонка.

Таблица 2.9 – Таблица Стоимость направлений/COST_LIST из БД

Псевдоним	Имя	Тип
код направления	ID	INT

Продолжение таблицы 2.9

Псевдоним	Имя	Тип
название	NAME	TEXT
валюта	CURRENCY	TEXT
стоимость	COST	FLOAT

2.3 Разработка программного обеспечения

Проектируемое мобильное приложение «Русский телефон», позволит:

1. Совершать аудио звонки.
2. Хранить историю звонков.
3. Иметь доступ к контактам мобильного устройства с возможностью позвонить на мобильные номера и номера «Русского телефона».
4. Предоставлять доступ к изменению телефонной книги (добавление, удаление и изменение контактов).
5. Регистрироваться в сервисе непосредственно из приложения.
6. Пополнять баланс в сервисе.
7. Узнавать баланс счета.
8. Обратиться непосредственно из приложения в службу поддержки.
9. Возможность пригласить, воспользоваться сервисом, людей из списка контактов.

2.3.1 Описание серверных программных модулей

В серверном обеспечении можно выделить два основных модуля, с которыми напрямую взаимодействует клиентская часть разработанного приложения.

Важнейшим элементом, обеспечивающим функционирование VoIP, является сервер Астериск.

Asterisk — свободное решение компьютерной телефонии (в том числе,

VoIP) с открытым исходным кодом от компании Digium, первоначально разработанное Марком Спенсером. Приложение работает на операционных системах Linux, FreeBSD, OpenBSD и Solaris и др. Имя проекта произошло от названия символа «*» (англ. asterisk — «звёздочка»).

Asterisk в комплексе с необходимым оборудованием обладает всеми возможностями классической АТС, поддерживает множество VoIP-протоколов и предоставляет богатые функции управления звонками.

Является популярнейшим программным сервером IP-телефонии, уже используется в работе сервиса «Русский телефон».

Приложение обращается к серверу Астериск для аутентификации и авторизации. С этим сервером приложение поддерживает соединение по протоколу SIP, и осуществляет передачу голосовой информации.

Астериск взаимодействует с биллинговой системой и АТС.

Другим ключевым элементом серверной архитектуры является web-сервер, предоставляющий доступ к информации о балансе и осуществляющий создание новых пользователей по запросу приложения. Его разработка велась на языке PHP.

PHP – скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

2.3.2 Описание клиентских программных модулей

В качестве клиента в системе выступает разработанное на языке Java мобильное приложение.

Java – объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Программы на Java транслируются в байт-код,

выполняемый виртуальной машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание [37].

Особенностью архитектуры мобильных приложений для Андроид, является разделение реализации на структурные элементы – «Activity», являющиеся частью жизненного цикла программы.

Activity (Активность, Деятельность) – это компонент приложения, который выдает экран, и с которым пользователи могут взаимодействовать для выполнения каких-либо действий, например, набрать номер телефона, сделать фото, отправить письмо или просмотреть карту. Каждой операции присваивается окно для прорисовки соответствующего пользовательского интерфейса. Обычно окно отображается во весь экран, однако его размер может быть меньше, и оно может размещаться поверх других окон.

Как правило, приложение состоит из нескольких операций, которые слабо связаны друг с другом. Обычно одна из операций в приложении обозначается как «основная», предлагаемая пользователю при первом запуске приложения. В свою очередь, каждая операция может запустить другую операцию для выполнения различных действий. Каждый раз, когда запускается новая операция, предыдущая операция останавливается, однако система сохраняет ее в стеке («стек переходов назад»). При запуске новой операции она помещается в стек переходов назад и отображается для пользователя. Стек переходов назад

работает по принципу «последним вошёл — первым вышел», поэтому после того как пользователь завершил текущую операцию и нажал кнопку «Назад», текущая операция удаляется из стека (и уничтожается), и возобновляется предыдущая операция. (Подробные сведения о стеке переходов назад представлены в статье Задачи и стек переходов назад.)

Когда операция останавливается по причине запуска новой операции, для уведомления об изменении ее состояния используются методы обратного вызова жизненного цикла операции. Существует несколько таких методов, которые может принимать операция вследствие изменения своего состояния — создание операции, ее остановка, возобновление или уничтожение системой; также каждый обратный вызов представляет возможность выполнить определенное действие, подходящее для соответствующего изменения состояния. Например, в случае остановки операция должна освободить любые крупные объекты, например, подключение к сети или базе данных. При возобновлении операции вы можете повторно получить необходимые ресурсы и возобновить выполнение прерванных действий. Такие изменения состояния являются частью жизненного цикла операции. [1].

«Activity» можно выделить по выполняемым в нем функциям и интерфейсам пользователя, которые она включает:

- загрузка приложения и подключение сервисных модулей;
- вход и регистрация в приложении;
- менеджмент звонка;
- исходящий вызов;
- входящий вызов;
- пополнение счета.

Отдельно стоит выделить главную «Activity», инкапсулирующую взаимодействие с телефонной книгой, историей звонков, номеронабирателем, меню настроек приложения.

2.3.3 Компоненты пользовательского интерфейса

Интерфейс пользователя – коммуникационный канал, реализующий взаимодействие пользователя с программным обеспечением с помощью элементов управления.

Графический интерфейс пользователя – среда организации взаимодействия пользователя с вычислительной системой.

Стандартный графический интерфейс пользователя должен отвечать ряду требований:

– поддерживать информационную технологию работы пользователя с программным продуктом – содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки, расположенные в естественной последовательности использования;

– ориентироваться на конечного пользователя, который общается с программой на внешнем уровне взаимодействия;

– удовлетворять правилу «шести» – в одну линейку меню включать не более шести понятий, каждое из которых содержит не более шести операций;

– графические объекты сохраняют свое стандартизованное назначение и, по возможности, местоположение на экране [11].

Разработка пользовательского интерфейса – одна из самых сложных и ответственных задач проектирования. Это объясняется тем, что пользователя интересует в первую очередь удобство, эргономичность и наглядность [11].

Графический интерфейс Android приложений строится на иерархии из элементов двух типов: View и ViewGroup. Элементами типа View являются любые дочерние элементы пользовательского интерфейса (UI widgets), такие как кнопка, поле для ввода и т.д. ViewGroup – это невидимый контейнер графических элементов, определяющий их размещение на экране. Это может быть вертикальный, горизонтальный список или таблицы элементов. Один ViewGroup может содержать множество view и другие ViewGroup. Оперируя различными графическими элементами и контейнерами элементов можно

создавать интерфейсы. Иерархия View и ViewGroup представлена на рисунке 2.5.

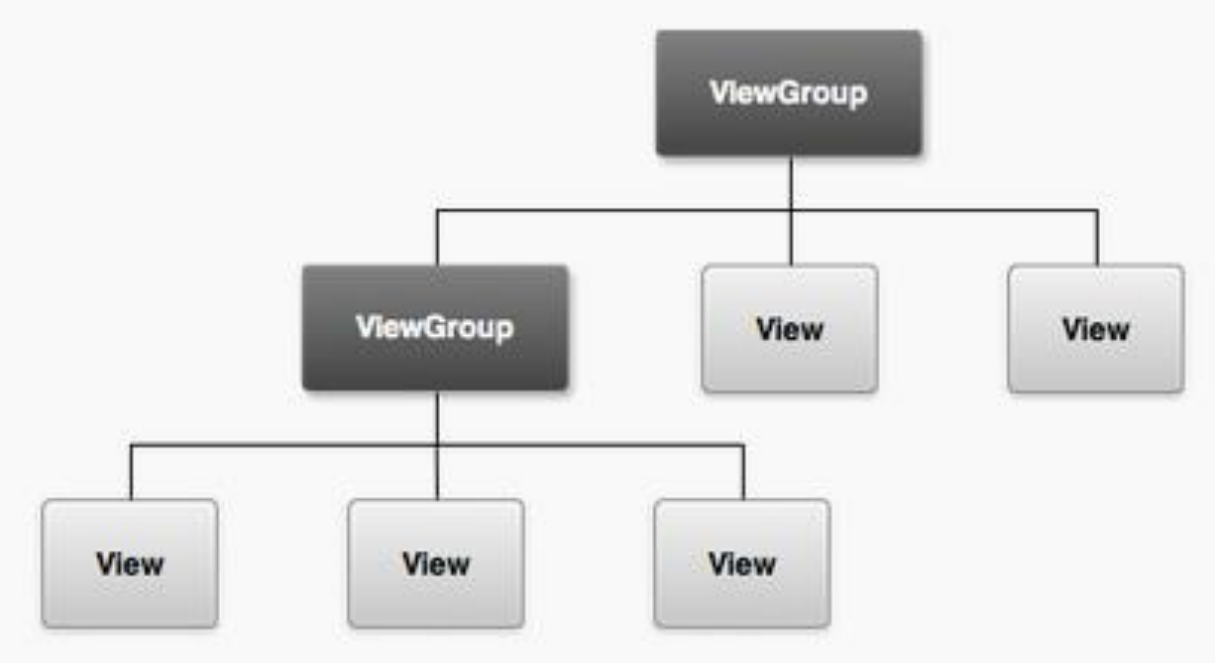

Рисунок 2.5 – Иерархия View и ViewGroup

Доступны следующие виды ViewGroup:

– FrameLayout – самая простая разметка, прикрепляет каждое новое дочернее представление к левому верхнему углу экрана, накладывая новый элемент на предыдущий, заслоняя его;

– LinearLayout – помещает дочерние представления в горизонтальный или вертикальный ряд. Вертикальная разметка представляет собой колонку, а горизонтальная – строку с элементами. Данная разметка позволяет задавать не только размеры, но и «относительный вес» дочерних элементов, благодаря чему можно гибко контролировать их размещение на экране;

– RelativeLayout – наиболее гибкий среди стандартных видов разметки. Позволяет указывать позиции дочерних Представлений относительно границ свободного пространства и других представлений;

– TableLayout – позволяет размещать дочерние представления внутри ячеек «сетки», состоящей из строк и столбцов. Размеры ячеек могут оставаться постоянными или автоматически растягиваться при необходимости;

– GridLayout – отображает прокручиваемую сетку со строками и столбцами.

Для создания пользовательского интерфейса в Android приложениях используется расширяемы язык разметки XML. Для создания окна или группы элементов необходимо создать отдельный файл с расширением .xml.

Каждый созданный .xml файл должен иметь хотя бы одну ViewGroup, которая будет описывать то, каким образом будут располагаться элементы на форме.

Приложение выполнено опираясь на методологии «material design» и «flat design».

Material design – единая концепция построения логики работы и внешнего вида сервисов и приложений, унифицирующая все продукты Google с целью их максимально лёгкого и интуитивного восприятия пользователями. Является рекомендованной методологией для реализации интерфейсов в приложении для платформы Андроид.

Flat design – минималистичный подход к дизайну объектов, который подчеркивает удобство использования, он в большей степени ориентирован на конечного пользователя. Этот подход используется при создании приложений для платформы IOS, однако для того что бы приложение на разных платформах выглядели одинаково, стоит использовать оба подхода.

Важнейшим компонентом удобного интерфейса является навигация. Для упрощения навигации и создания «легкого» интерфейса используется система из трех «вкладок» для основных действий и «гамбургер-меню» для остальных действий.

На рисунке 2.6. представлен главный экран приложения. В верху экрана можно увидеть строку с логином пользователя и цветным индикатором состояния подключения к серверу. Внизу экрана находятся переключатели вкладок, (слева на право) отвечающие за переключение между основными экранами «история звонков», «телефонная книга», «набор номера».

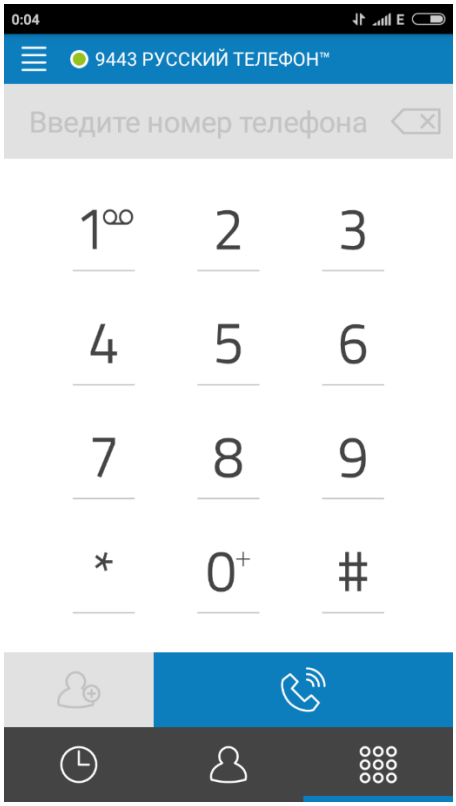

Рисунок 2.6 – Экран «набора номера»

Доступ к остальным функциям сокрыт, в так называемом «гамбургер-меню», названном так из-за иконки нескольких горизонтальных полосок. Кнопка вызова меню расположена в верхнем левом углу. Это-же меню вызывается жестом «смахнуть», если провести с левого края экрана на право.

Вид открытого меню и его содержание можно увидеть на рисунке 2.7.

Видно, что меню выезжающее из-за левого края, причем, остается зазор, в котором затенен предыдущий экран. Пользователь может нажать в это место, чтобы закрыть меню или воспользоваться кнопкой «Назад».

В верху экрана виден номер пользователя в сервисе, статус подключения к серверу «Астериск» и текущий баланс. Нажатие данной кнопки приведет на экран настроек аккаунта.

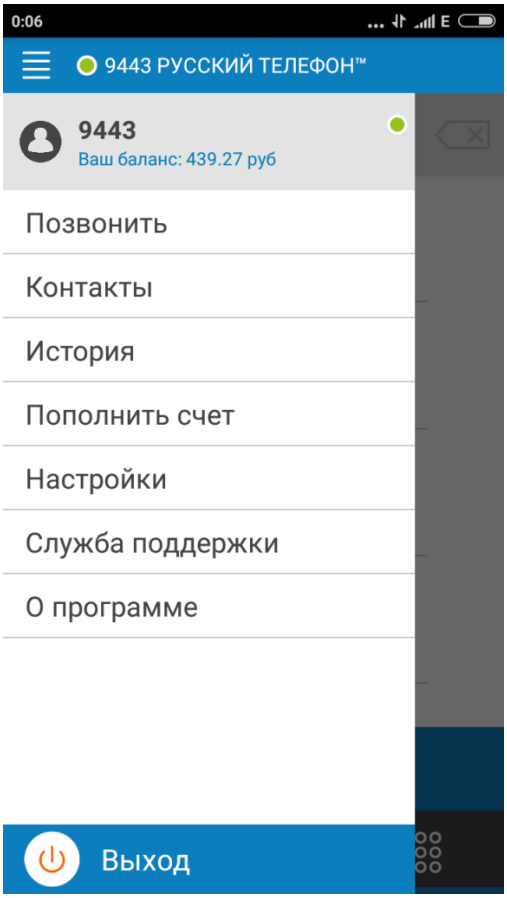

Рисунок 2.7– Экран «меню-гамбургер»

Ниже расположены три пункта, позволяющие сделать переход между «вкладками», аналогично кнопкам на главном экране.

Еще ниже расположен пункт меню, ведущий на экран пополнения счета.

Затем, пункт перехода к настройкам приложения, там же находится возможность создания новой учетной записи или подключение еще одной имеющейся.

Следующим пунктом, идет возможность обратиться в службу поддержки через email.

Заключительный пункт меню отображает экран информации о приложении и сервисе.

Кнопка «Выход» полностью закрывает приложение без возможности приема вызовов.

Однако, после установки приложения, пользователя будет ждать экран входа и регистрации (рисунок 2.8.), и только потом он сможет воспользоваться

функционалом.

Если пользователь уже зарегистрирован в сервисе «Русский телефон», достаточно нажать кнопку «Войти», ввести логин и пароль своей учетной записи (рисунок 2.8).

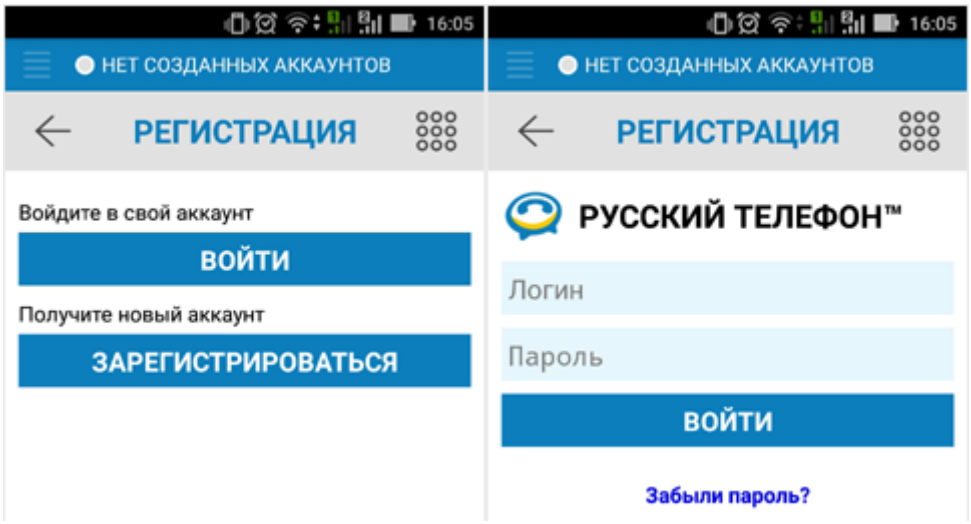

Рисунок 2.8– Экран «Вход и регистрация» и экран «Ввода логина и пароля»

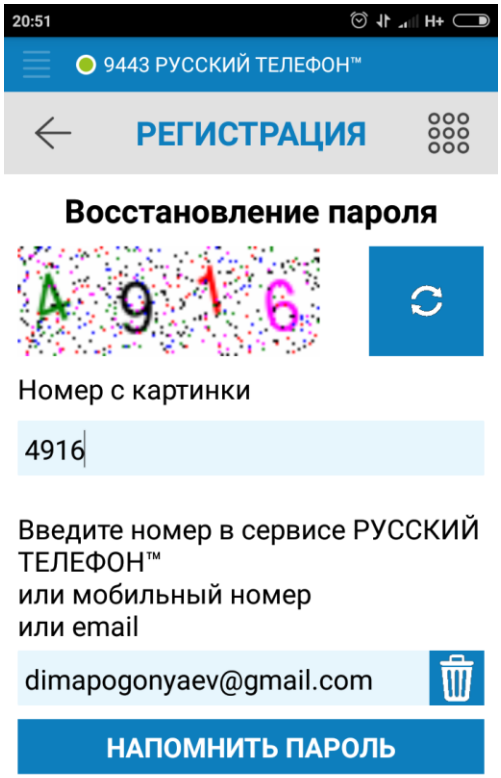

Рисунок 2.9– Экран «Восстановление пароля учетной записи»

Если пользователь забыл пароль, может воспользоваться формой восстановления пароля на сотовый телефон или email (рисунок 2.9).

Чтобы избежать злонамеренного использования этой функции, добавлена проверка вводом капчи.

Для восстановления нужно ввести номер мобильного телефона или email, указанные при регистрации. Так же можно восстановить пароль, указав номер в сервисе русский телефон.

Чтобы перейти на форму создания новой учетной записи новый пользователь должен выбрать кнопку «Зарегистрироваться». Поскольку форма не помещается на одном экране, то она представлена в виде прокручиваемого списка полей. Форма регистрации, представленая на рисунок 2.10, имеет обязательные поля и в случае их не заполнения, пользователю эти поля будут выделены в более красивой форме.

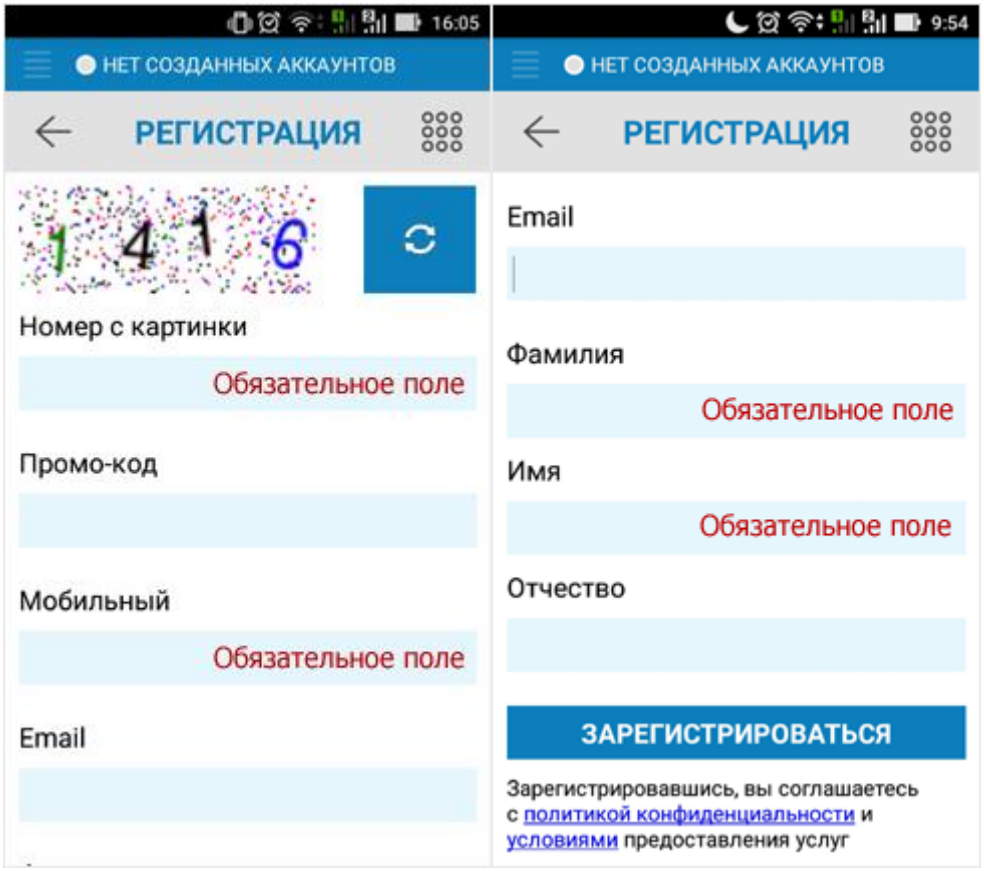

Рисунок 2.10– Экран «Ввод учетных данных для регистрации»

После регистрации на мобильный телефон придет смс-сообщение с кодом регистрации. Нужно ввести его в соответствующее поле и нажать «Отправить» (рисунок 2.11).

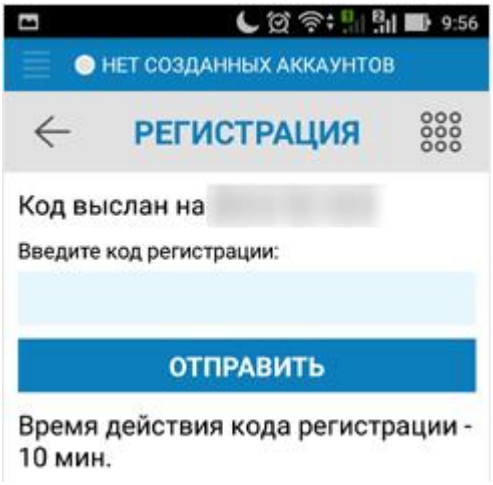

Рисунок 2.11– Экран «Ввод кода подтверждения»

Код подтверждения отправляется на сервер. Если все условия выполнены, то на сервере будет создана учетная запись. Регистрационные данные высылаются пользователю на email и на экране мобильного устройства появляется сообщение «Регистрация успешно завершена» (рисунок 2.12).

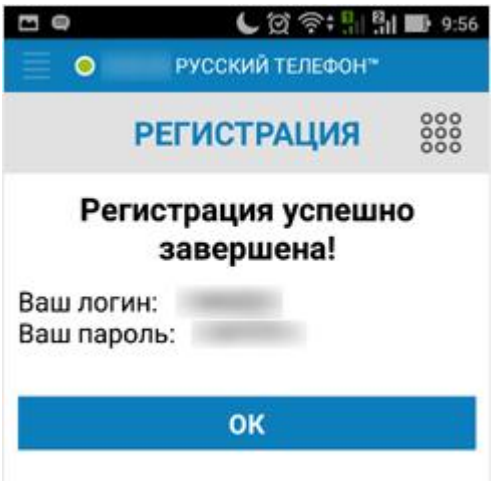

Рисунок 2.12– Экран «Сообщение об успешной регистрации»

Теперь пользователь может пользоваться бесплатными звонками между абонентами сервиса «Русский телефон». Для совершения звонков на сотовые и стационарные телефоны операторов связи, необходимо пополнить баланс счета.

По статистике использования программ для совершения коммуникаций, самым популярным действием является повторный вызов абонента из истории звонков.

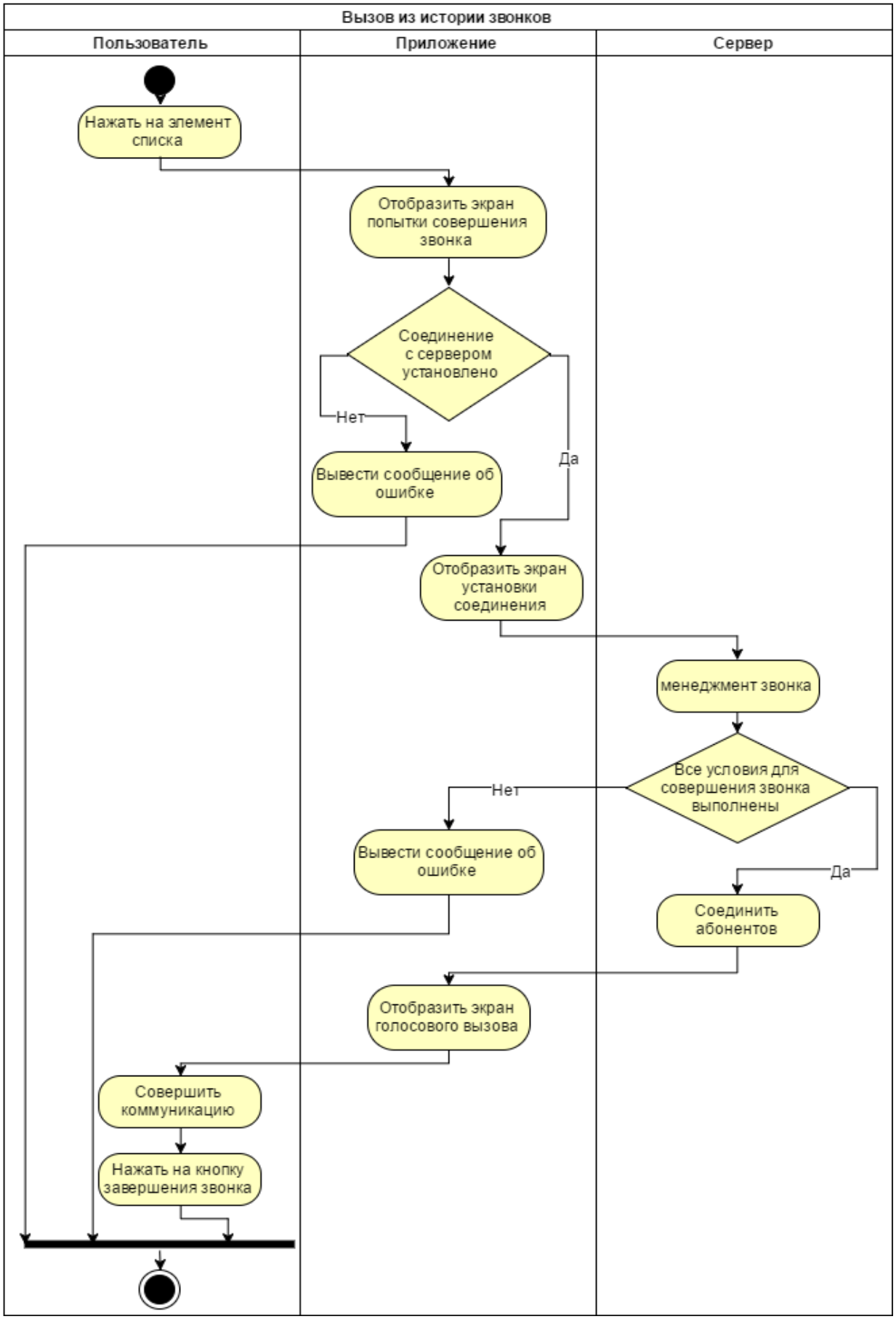

Рисунок 2.13– Диаграмма действий «Голосовой вызов из истории звонков».

Рассмотрим рисунок 2.13 действий для сценария, когда пользователь уже делал звонки с помощью приложения русский телефон и хочет совершить повторный вызов, при этом все условия, необходимые для звонка, выполнены. Еще примем допущение, что экран «История звонков» уже открыт.

Из диаграммы наглядно отображен процесс взаимодействия пользователя, приложения и сервера, однако детали реализации лучше рассмотрим на рисунках.

Первым этапом является взаимодействие с экраном истории звонков. На рисунке 2.14 отображены совершенные пользователем звонки. Нажатие на один из элементов списка приведет к вызову данного абонента.

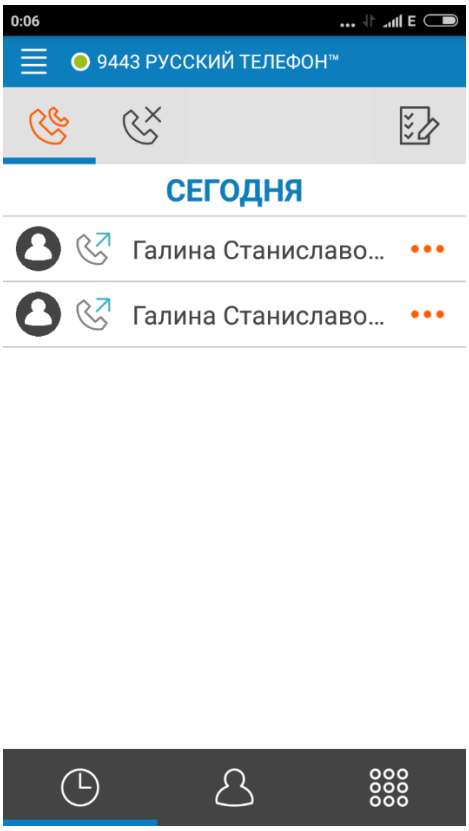

Рисунок 2.14– Экран «История звонков»

Затем осуществляется соединение с сервером и проверка необходимых условий (например, наличие положительного баланса и установка соединения с другим абонентом).

Результатом является разговор абонентов. В это время отображается соответствующий экран (рисунок 2.15).

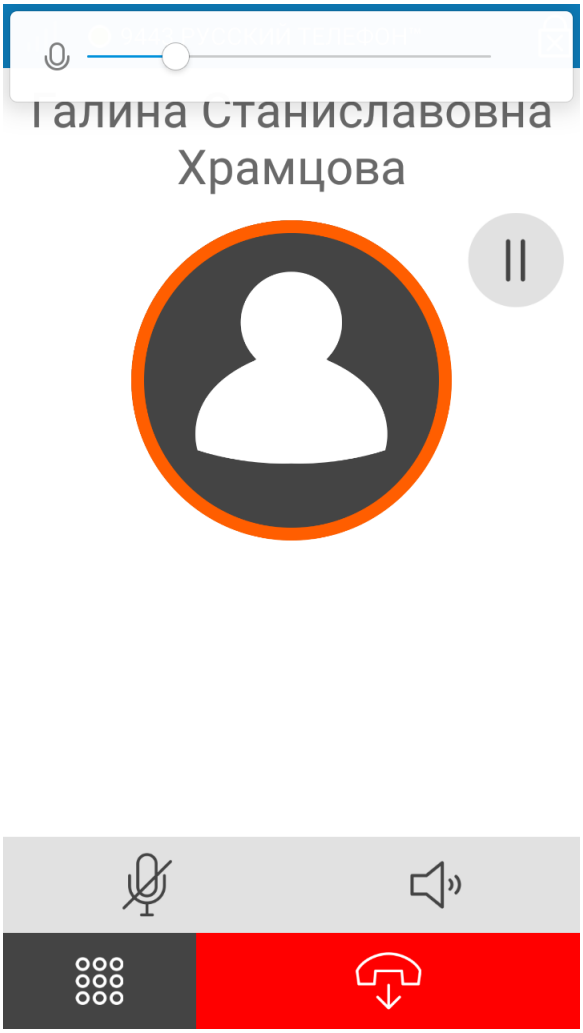

Рисунок 2.15– Экран «Текущий звонок»

На экране «Текущий звонок» расположены элементы управления текущим вызовом.

Вверху расположен телефон другого участника разговора или данные сохраненного контакта.

Ниже есть кнопка паузы, позволяющая временно прервать обмен аудиоинформацией.

Внизу экрана расположены основные элементы управления.

Большая красная кнопка отвечает за прекращение вызова.

Слева от неё расположена кнопка вызова клавиатуры для отправки DTMF сигнала.

Немного выше находятся кнопки отключения микрофона и включения громкой связи.

Регулировка громкости осуществляется аппаратными кнопками.

Другим способом совершения звонка является прямой набор номера, этот процесс можно увидеть на рисунке 2.16.

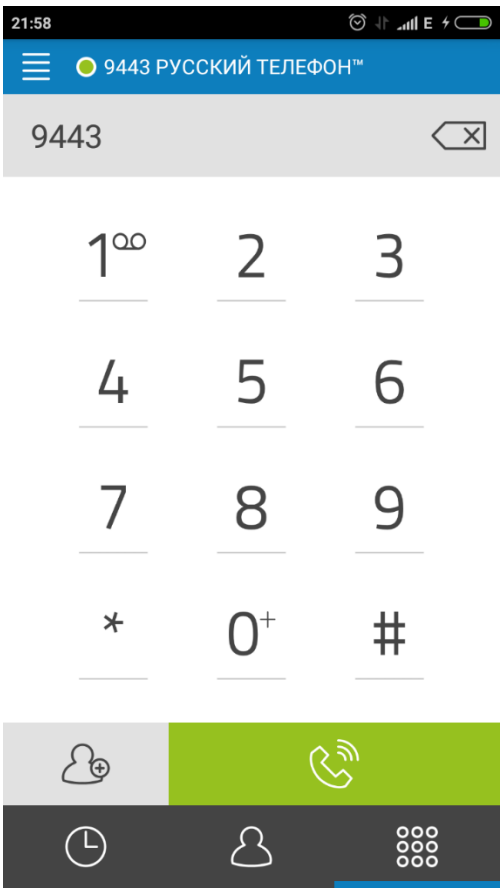

Рисунок 2.16– Экран «Текущий звонок»

В середине экрана расположены кнопки набора номера. Сам номер отображается в строке выше, там-же видна кнопка, позволяющая удалить ошибочно введенные символы. Под областью набора номера расположены кнопка добавления номера в список контактов и кнопка инициации вызова. Если номер не введен, то кнопка добавления номера в список контактов недоступна, а кнопка вызова заполняет строку номера согласно последнему сделанному звонку.

Третьим способом сделать звонок, является выбор номера из списка контактов. Приложение «Русский телефон» имеет доступ к списку контактов мобильного устройства. Есть возможность изменять контакты или создавать новые. Экран отображающий список контактов можно увидеть на рисунке 2.17.

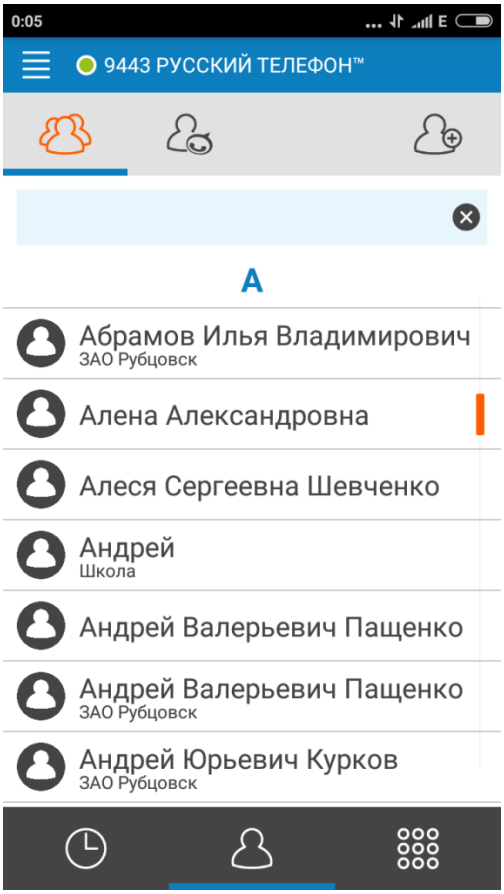

Рисунок 2.17– Экран «Контакты»

Сверху слева расположены кнопки переключения между всеми контактами и контактами, имеющими номер в сервисе «Русский телефон».

Сверху справа расположена кнопка создания нового контакта.

Ниже расположена строка поиска по телефонной книге и собственно сам список контактов.

Однако, для совершения платных звонков на мобильные и стационарные номера, нужно иметь положительный остаток. За пополнение счета из приложения, отвечает соответствующий экран, представленный на рисунке 2.18.

В начале экрана представлена информация, на какой номер в сервисе будет сделан платеж, и баланс соответствующего счета.

Далее пользователю предлагается выбрать сумму, на которую будет пополнен счет.

Рисунок 2.18– Экран «Пополнение счета»

Ниже идет список способов оплаты. Этот список загружается с сервера, поэтому всегда актуален и может быть мгновенно изменен при необходимости.

Непосредственная оплата происходит в браузере пользователя, на сайте партнеров. В случае банковской карты, это будет сайт банка.

Настройки программы представлены в привычном для платформы Андроид виде (рисунок 2.19).

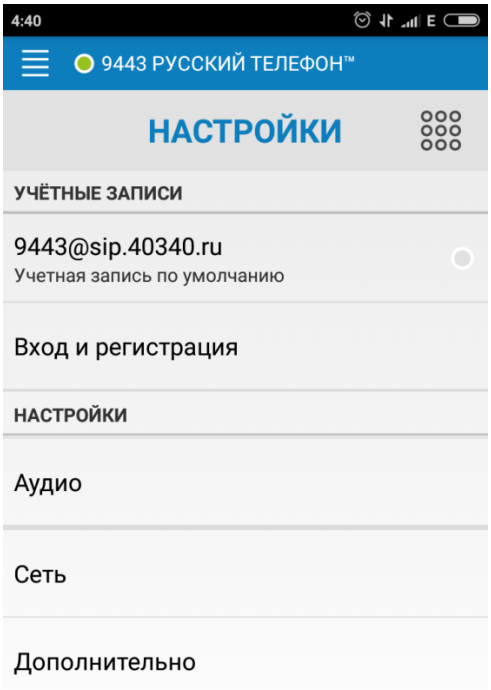

Рисунок 2.19– Экран «Настройки приложения»

Настройки разделены на группы. Сначала идут учетные записи, с возможностью настроить каждую из них отдельно. Здесь же находится пункт, позволяющий создать новую учетную запись или подключить имеющуюся.

Настройки самого приложения разделены на настройки аудио, сети и дополнительные настройки. В аудио настройках отображены используемые кодеки, а также параметры эхоподавления. В настройках сети можно задать использование только Wi-Fi подключения для работы. В разделе «дополнительно» можно настроить автозапуск и работу в фоне.

Номер текущей версии приложения можно узнать на экране «О программе» (рисунок 2.20).

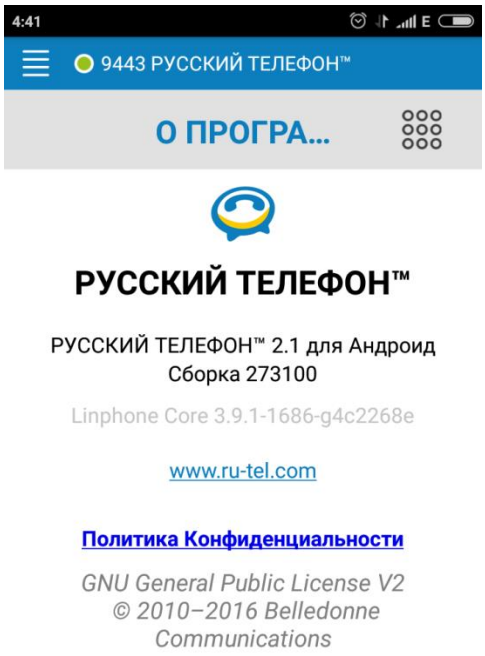

Рисунок 2.20– Экран «О программе»

Также на этом экране отображен номер сборки приложения и номер сборки ядра Linphone, на котором оно основано.

Пользователь может воспользоваться этим экраном для перехода на главную страницу сайта сервиса или на страницу с политикой конфиденциальности.

Здесь же указана лицензия, по которой распространяется приложение.

2.4 Компьютерно-сетевое обеспечение

Мобильное приложение имеет архитектуру клиент-сервер.

Серверная часть приложения использует существующее сетевое оборудование ЗАО «Рубцовск» без изменений. Структура сети также не подвергается изменениям, т.к. ее характеристики полностью удовлетворяют предъявленным требованиям.

Сетевое оборудование и структура сети были подробно разобраны в пункте 1.5, где было обосновано проектное решение по использованию существующей архитектуры.

2.5 Обеспечение информационной безопасности

Информационная безопасность (ИБ) – состояние защищённости информационной среды общества, обеспечивающее её формирование, использование и развитие в интересах отдельных граждан, организаций и государства [25].

Персональные данные – любая информация, относящаяся к определенному или определяемому на основании такой информации физическому лицу (субъекту персональных данных), в том числе его фамилия, имя, отчество, год, месяц, дата и место рождения, адрес, семейное, социальное, имущественное положение, образование, профессия, доходы, другая информация.

Информационная система персональных данных – информационная система, представляющая собой совокупность персональных данных, содержащихся в базе данных, а также информационных технологий и технических средств, позволяющих осуществлять обработку таких персональных данных с использованием средств автоматизации или без использования таких средств [22].

Классификация информационной системы персональных данных.

Определение категории обрабатываемых персональных данных:

– первая категория – персональные данные, касающиеся расовой, национальной принадлежности, политических взглядов, религиозных и философских убеждений, состояния здоровья, интимной жизни;

– вторая категория – персональные данные, позволяющие идентифицировать субъекта персональных данных и получить о нём дополнительную информацию, за исключением персональных данных, относящихся к категории 1;

– третья категория – персональные данные, позволяющие идентифицировать субъекта персональных данных;

– четвертая категория – обезличенные и (или) общедоступные персональные данные.

По результатам анализа исходных данных ИС присваивается один из следующих классов:

– класс 1 (К1) – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к значительным негативным последствиям для субъектов персональных данных;

– класс 2 (К2) – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к негативным последствиям для субъектов персональных данных;

– класс 3 (К3) – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, может привести к незначительным негативным последствиям для субъектов персональных данных;

– класс 4 (К4) – информационные системы, для которых нарушение заданной характеристики безопасности персональных данных, обрабатываемых в них, не приводит к негативным последствиям для субъектов персональных данных [22, 8].

Для защиты от постороннего вторжения предусматриваются определенные меры безопасности. В ИС это осуществляется программными средствами, которые выполняют следящие функции:

- идентификация субъектов и объектов;
- разграничение доступа к ресурсам и информации;
- контроль и регистрация действий.

Процедура идентификации и подтверждения подлинности предполагает проверку является ли субъект, осуществляющий доступ, или объект, к которому осуществляется доступ, тем за кого себя выдает. Здесь используются следующие методы:

- простые, сложные или одноразовые пароли;
- средства анализа индивидуальных характеристик субъекта (геометрических данных);
- ключи, жетоны, магнитные карты;
- обмен вопросами и ответами с администратором системы;
- специальные идентификаторы и контрольные суммы для программ и данных.

После процедуры идентификации пользователь получает доступ к системе, где защита от несанкционированного доступа реализуется на трех уровнях: на уровне аппаратуры, на уровне программного обеспечения, на уровне данных.

Защита информации на 1 и 2 уровнях предусматривает управление доступом к различным вычислительным ресурсам (отдельным устройствам, операционной системы, служебным или личным программам пользователя).

Защита информации на уровне данных направлена на защиту информации в процессе обращения к ней, в процессе работы с файловой структурой ЭВМ, в процессе передачи информации по каналам связи.

В общем, комплекс программно-технических средств и организованных решений по защите информации от несанкционированного доступа характеризуется следующими действиями:

– логическое управление доступом;
– регистрацией, контролем и учетом работы;
– применением криптографических средств;
– обеспечением целостности информации.

Выделяют следующие формы контроля и управления доступом:

– предотвращение доступа к жесткому диску, каталогам и файлам;
– установка привилегий к группам файлов;
– защита от модификаций и изменений;
– защита от уничтожения;
– предотвращение копирования.

Под средствами защиты от копирования понимается такие средства, которые обеспечивают выполнение программой своих функций только при опознании некоторого уникального не копированного элемента.

Защита от копирования реализуется выполнением ряда функций:

– идентификация среды, из которой запускается система;
– проверка подлинности среды, из которой произошел запуск;
– немедленная реакция на запуск из несанкционированной среды;
– обязательная регистрация санкционированного копирования;
– противодействие изучению алгоритмов работы системы [27].

Защита информации БД на программном уровне состороны мобильного клиента осуществляется стандартными методами, такими как: проверка введенной пользователем информации и проверка корректности принимаемых от клиента данных.

Доступ к API web-сервера осуществляется по зашифрованному HTTPS соединению. Для работы алгоритма используются сертификаты Comodo – старейшего сертификационного цента, по умолчанию входящего в число доверенных источников сертификации в том числе и на ОС Андроид.

Защита от совершения несанкционированных звонков осществляется на основе диалплана и биллинга.

3 Оценка эффективности от внедрения мобильного приложения

3.1 Общие положения

Эффективность процессов характеризуется системой показателей, отражающих соотношение их затрат и результатов. Эффективность процесса тем выше, чем выше результаты и ниже приложенные усилия.

Эффективность ИС – это свойство системы выполнять поставленную цель в заданных условиях использования и с определенным качеством. Эта характеристика отражает:

– действенность системы, т.е. степень соответствия своему назначению (прагматическая эффективность);

– техническое совершенство (техническая эффективность);

– простоту, технологичность разработки и создание системы (технологическая эффективность);

– удобство использования и обслуживания (эксплуатационная эффективность);

– улучшение и облегчение условий труда, изменение его содержания, развитие творческих функций, способностей и потребностей людей, преодоление существенных различий в труде и др. (социальная эффективность);

– экономическую целесообразность внедрения, т.е. целесообразность произведенных на создание и функционирование системы затрат (экономическая эффективность).

Понятие эффективности связано с получением некоторого полезного результата – эффекта использования.

В соответствии с ГОСТ Р ИСО 9000-2001, эффективность функционирования определяется соотношением результата (эффекта) и

затраченными ресурсами. Приведенной оценкой затрат ресурсов выступает их стоимость. Затраты на функционирование состоят из:

- стоимости приобретения программной платформы;
- стоимости доработки;
- стоимости внедрения;
- стоимости системного и вспомогательного программного обеспечения;
- стоимости аппаратного и сетевого обеспечения;
- количества циклов (лет) эксплуатации;
- стоимости эксплуатации.

Основные задачи, стоящие при создании – минимизация стоимости и обеспечение требуемого качества.

Качество – это совокупность свойств системы, обусловливающих возможность ее использования для удовлетворения определенных потребностей пользователей в соответствии с ее назначением.

Основными показателями качества являются:

- надежность;
- достоверность;
- безопасность.

Надежность – свойство системы сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных условиях применения.

Надежность является средством обеспечения актуальной и достоверной информации на выходе системы.

Достоверность функционирования – свойство системы, обусловливающее безошибочность производимых ею преобразований информации. Достоверность функционирования полностью определяется и измеряется достоверностью ее результатной информации.

В любой сфере человеческой деятельности оценка эффективности внедрения любой новой техники, технологий и осуществляется с помощью множества показателей. К ним относятся показатели прагматической,

технической, эксплуатационной, социальной и экономической эффективности.

3.2 Показатели эффективности

В качестве показателей прагматической эффективности для мобильного приложения могут выступать:

– показатели безопасности информационной системы (защита сервера от различных атак, уменьшение уязвимости портала за счет контроля вводимой пользователем информации);

– показатели оперативности (использование созданного приложения позволяет снизить затраты ресурсов на доступ к информации, значительно снижены временные затраты на поиск необходимой информации).

Показатели технической эффективности должны оценивать техническое совершенство, научно-технический уровень организации и функционирования этой системы.

Показатели эксплуатационной эффективности:

– показатели надежности – обрабатываемые данные зависят от надежности компьютера и спроектированной системы;

– функциональные возможности – появилась возможность использования портала с помощью мобильных устройств;

– технология обслуживания – система не требует вмешательства программиста во время работы.

Среди показателей социальной эффективности можно выделить предоставление постоянного и удобного доступа к необходимому расписанию, а также повышение престижности ВУЗа, конкурентоспособности.

Обобщающими показателями эффективности любой ИС являются показатели экономической эффективности. Часто прибыль определяется путем экспертной оценки или по аналогии с другими подобными системами.

Для оценки эффективности могут использоваться две группы показателей: интегральные традиционные показатели и частные показатели.

Обычно в качестве экономических показателей используются:

- годовой экономический эффект;
- коэффициент экономической эффективности капитальных вложений;
- срок окупаемости капитальных вложений;
- трудоемкость обработки информации;
- эксплуатационная стоимость затрат;
- расчет текущих затрат пользователя;
- экономия текущих затрат при автоматизации;
- годовая экономия затрат на материалы.

Экономический эффект – в случае внедрения мобильного приложения, экономическим эффектом является получение прибыли от привлечения новых клиентов.

Предварительный экономический эффект рассчитывается до выполнения разработки на основе данных технических предложений и прогноза использования. Предварительный эффект является элементом технико-экономического обоснования (ТЭО) разработки проекта.

Потенциальный экономический эффект рассчитывается по окончании разработки на основе достигнутых технико-экономических характеристик и прогнозных данных о максимальных объёмах использования программного изделия.

Коэффициент экономической эффективности капитальных вложений показывает величину годового прироста прибыли, образующуюся в результате производства или эксплуатации программного изделия на один рубль капитальных единовременных вложений.

Срок окупаемости (величина, обратная коэффициенту эффективности) – показатель эффективности использования капиталовложений, представляет собой период времени, в течение которого произведённые затраты на программные изделия окупаются полученным эффектом.

Для оценки экономической эффективности внедрения мобильного приложения, можно взять прибыль от привлечения новых клиентов.

3.3 Расчет экономической эффективности

PP (Payback Period) – Метод расчета срока окупаемости инвестиций – один из самых простых и широко распространенных в мировой учетно-аналитической практике. Его алгоритм зависит от равномерности распределения прогнозируемых доходов от инвестиций: если доход распределен по годам равномерно, то срок окупаемости рассчитывается делением единовременных затрат на величину годового дохода, обусловленного ими; если прибыль распределена неравномерно, то срок окупаемости рассчитывается прямым подсчетом числа лет, в течение которых инвестиция будет погашена кумулятивным доходом. По сути, PP представляет собой анализ возврата средств исходя из принятых в компании максимальных сроков окупаемости вложений.

Для достижения конкретной управленческой цели возможно применение различных решений, имеющих, как правило, разные экономические результаты. Поэтому предпочтительно разрабатывать многовариантные предложения, в которых сравниваются различные подходы к решению проблемы.

Сравнение при этом возможно по двум направлениям:

– существующий вариант сравнивается с предлагаемым вариантом;

– один вариант сравнивается с другим (или несколькими) одинаково не применяемыми до этого на объекте исследования.

3.3.1 График выполнения работ

График выполнения работ представлен в таблице 3.1.

Таким образом, на разработку мобильного приложения под операционную систему Андроид для сервиса «Русский телефон» было затрачено 80 дней, или 560 человеко-часов.

Таблица 3.1 – График выполнения работ по разработке приложения

№ п/п	Наименование работ	Длительность работы	
		в днях	в часах
1	Разработка технического задания	2	14
2	Планирование приложения	10	70
3	Рабочее проектирование приложения	48	336
4	Отладка и тестирование приложения	16	112
5	Обобщение и оценка результатов	4	28
6	Итого	80	560

3.3.2 Расчет стоимости разработки приложения

При расчете стоимости (составлении сметы затрат) разработки мобильного приложения учитываются следующие виды расходов:

- стоимость материалов и покупных изделий;
- основная заработная плата;
- дополнительная заработная плата;
- страховые взносы;
- накладные расходы;
- затраты на машинное время (затраты на электроэнергию).

Перечень затрат на материалы и покупные изделия приведен в таблице 3.2.

Транспортные расходы учитываются в объеме 10% от суммы затрат на материалы и покупные изделия, что составляет 63 руб. Таким образом, затраты на материалы и покупные изделия равны:

$$Зм = 630 + 630*10/100= 693 \text{ руб.} \quad (3.1)$$

Таблица 3.2 – Затраты на материалы и покупные изделия

№ п/п	Наименование	Единица измерения	Количество	Цена за единицу, руб.	Стоимость, руб.
1	Доступ в Internet	шт.	1	350	350
2	Канцтовары	шт.	2	50	100
3	Бумага формата А4	упаковка	1	180	180
4	Итого				630
5	Транспортные расходы (10 % от п.5)				63
6	Итого				693

Ниже приведен расчет фонда заработной платы (основной и дополнительной заработной платы).

К этой статье относится основная и дополнительная заработная плата разработчика (программиста). Результаты расчета фонда заработной платы представлены в таблице 3.3.

Таблица 3.3 – Расчет фонда заработной платы

№ п/п	Должность: техник	Кол-во рабочих дней	Кол-во проработанных дней	Размер дневной оплаты	Заработная плата руб.
1	Основная заработная плата	80	80	568,75	45500
2	Дополнительная заработная плата (10% от основной заработной платы)				4550
3	Итого фонд заработной платы				50050

В статью «Дополнительная заработная плата» входят выплаты, предусмотренные трудовым договором. Размер дополнительной заработной платы разработчика определяется в размере 10 процентов от основной заработной платы.

$$Здоп = Зосн * 10/100 = 45500*10/100 = 4550 \text{ руб.} \quad (3.2)$$

Следовательно, разработчику всего начислено:

$$Знач = (Зосн + Здоп) * = 45500 + 4550 = 50050 \text{ руб.} \quad (3.3)$$

Таким образом, фонд заработной платы разработчика составляет 50050 руб.

К отчислениям на социальные нужды относят страховые взносы в ПФР, ФСС, ФФОМС и взносы на страхование от несчастных случаев на производстве и профзаболеваний.

Страховые взносы рассчитываются в размере 32,2 процентов от фонда заработной платы, что составит:

$$СВ = Знач * 32,2/100 = 50050 * 32,2/100 = 16116,1 \text{ руб.} \quad (3.4)$$

Тарифы страховых взносов приведены в таблице 3.4.

Отчисления в пенсионный фонд ЗПФ составляют 24 процента от фонда заработной платы и равны:

$$ЗПФ = Знач * 24/100 = 50050 * 24/100 = 12012 \text{ руб.} \quad (3.5)$$

Отчисления в фонд обязательного медицинского страхования Змс равны:

$$Змс = Знач * 5,1/100 = 50050 * 5,1/100 = 2552,55 \text{ руб.} \quad (3.6)$$

Отчисления на социальное страхование Зсс равны:

$$Зсс = Знач * 2,9/100 = 50050 * 2,9/100 = 1451,45 \text{ руб.} \quad (3.7)$$

Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний равны:

$$Знс = Знач * 0,2/100 = 50050 * 0,2/100 = 100,1 \text{ руб.} \quad (3.8)$$

Численные значения отчислений на социальные нужды (страховые взносы), представлены в таблице 3.4.

Таблица 3.4 – Расчет отчислений на социальные нужды (страховые взносы)

№ п/п	Отчисления на социальные взносы (страховые нужды)	Тарифы страховых взносов, в %	Суммы страховых взносов, руб.
1	Отчисления в ПФР	24,00	12012
2	Отчисления В ФОМС	5,10	2552,55
3	Отчисления в ФСС	2,90	1451,450
4	Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний	0,20	100,1
5	Итого	32,20	16116,1

Размеры тарифов страховых взносов устанавливаются Федеральными законами. На момент разработки проекта необходимо руководствоваться действующим законодательством.

Далее приведены накладные расходы.

Накладные расходы, косвенные затраты – расходы, затраты, сопровождающие, сопутствующие основному производству, но не связанные с

ним напрямую, не входящие в стоимость труда и материалов - дополнительные к основным затратам расходы для обеспечения процессов производства и обращения.

Накладные расходы Зн фирмы составляют 20 процентов (условно) от суммы основной и дополнительной заработной платы:

$$Зн = (Зосн + Здоп) * 20/100 = 50050 * 20/100 = 2502.5 \text{ руб.} \quad (3.9)$$

Рассчитаем затраты на машинное время.

Как следует из данных таблицы 3.1, на разработку и тестирование мобильного приложения «Русский телефон» для Андроид потребовалось 80 рабочих дней (Дн).

В среднем с учетом перерывов программист работает за компьютером 7 часов в день. Себестоимость одного кВт/ч электроэнергии (С1квт/ч) для организаций составляет 8 рублей 50 копеек. При проведении расчетов в проекте необходимо в расчеты брать существующие на дату расчета тарифы. Суммарная мощность энергопотребителей для программиста складывается из мощности, потребляемой системным блоком персонального компьютера, монитором, принтером и другим периферийным оборудованием, которая составляет 1,2 кВт. Следовательно, за 7 часов работы программиста суммарное энергопотребление за день составит:

$$Р= 1,2 * 7 = 8.4 \text{ кВт/ч} \quad (3.10)$$

Таким образом, стоимость машинного времени Змаш, необходимого для разработки мобильного приложения «Русский телефон» для Андроид, составит:

$$Змаш = Р*Дн* С1квт/ч = 8.4 \text{ кВт/ч} *80 *8,5 \text{ руб./кВт/ч} = 5712 \text{ руб.} \quad (3.11)$$

Затраты на машинное время учитываются как затраты на электроэнергию.

В результате выше произведенных расчетов мы получили итоговые затраты на разработку (Таблице 3.5.).

В результате цена программного продукта Ц определяется итоговыми затратами и прибылью, которая, в свою очередь, составляет 30 процентов (условно) от фонда заработной платы:

$$Ц = 164929,1+50050 * 30/100 = 179944,1 \text{ руб.} \quad (3.12)$$

Таблица 3.5 – Итоговая смета затрат

№ п/п	Наименование статей расхода	Сумма, руб.
1	Стоимость материалов и покупных изделий	693,00
2	Основная заработная плата	45500,00
3	Дополнительная заработная плата	45500
4	Отчисления за социальные нужды (32, 2 % от п. 2 и п. 3)	16116,1
5	Накладные расходы (20 % от п. 2 и п. 3)	2502.50
6	Затраты на машинное время (затраты на электроэнергию)	57120
7	Итого	164929,1

3.3.3 Оценка экономической эффективности

Одним из основных методов определения экономической эффективности служит метод сравнения результатов работы до и после проведения мероприятия.

Расчет экономической эффективности по этому методу проведен на основе данных о суммарном количестве платных минут, совершённых через сервис до и после публикации приложения.

Анализ роста прибыли от платных звонков показал, что появление собственного мобильного приложения под ОС Андроид привело к притоку

новых клиентов. Кроме того, уже пользующиеся сервисом, оценив удобство нового способа, стали активнее пользоваться услугой.

В связи с неравномерностью распределения суммарной стоимости звонков во времени, для анализа был взят период в 4 месяца до внедрения и 4 месяца после.

В расчет брались только звонки с мобильных устройств, через софтофоны, так как структура данных позволяет отделить звонки, сделанные через сайт и стационарные телефоны IP-телефонии.

За расчётный период, получен доход 452167,22 рублей до внедрения и доход 678250,83 рублей после.

Из приведенных данных можно сделать выводы о том, что за 4 месяца дополнительные доходы от реализации программного продукта в результате выпуска мобильной версии составили:

$$Ддоп= 678250,83 – 452167,22 = 226083,61 \text{ руб.} \quad (3.13)$$

Расходы на разработку программного продукта равны 179944,1 рублей, поэтому экономическая эффективность Э(П) публикации приложения составляет:

$$Э (П)= Ддоп – Ц= 226083,61 – 179944,1 = 46139,51 \text{ руб.} \quad (3.14)$$

Таким образом, экономический эффект Э в 0,2564 раза (Э(П)/Ц=46139,51 /179944,1) превышает затраты на разработку.

Рентабельность проекта (Р) составила:

$$Р = Э(П)/Ц*100 = 46139,51/179944,1 * 100 = 25,64\% \quad (3.15)$$

Срок окупаемости проекта равен:

Ток =Ц/ Э(П)=179944,1/46139,51 = 3,9 месяца или 117 дня. (3.16)

Следовательно, разработка и публикация мобильного приложения «Русский телефон» для Андроид было экономически оправдано.

Наличие собственного мобильного приложения с удобной регистрацией открывает возможности для дальнейшего продвижения приложения, рекламная компания без этой важной детали вероятно провалилась бы.

Не стоит также забывать и о социальной значимости нововведения. Мобильное приложение «Русский телефон» бесплатно и звонки между абонентами тоже бесплатны. Таким образом «Русский телефон» объединяет людей, делая их ближе друг к другу.

Заключение

Целью исследования была разработка мобильного приложения для организации доступа к сервису VoIP коммуникации «Русский телефон» фирмы ЗАО «Рубцовск».

Для достижения цели были решены следующие задачи:

- проведен анализ предметной области;
- построена модель предметной области «как есть», с целью выявления недостатков;
- построена модель предметной области «как должно быть»;
- выполнен обзор и анализ существующих разработок;
- разработано мобильное приложение;
- подтверждена эффективность от внедрения мобильного приложения.

Результатом выпускной квалификационной работы стало мобильное приложение для организации доступа к сервису VoIP коммуникации «Русский телефон», обеспечившее пользователей удобным средством коммуникации. Приложение привлекло новых пользователей сервиса и стало причиной увеличения числа звонков, что подтвердило экономическую эффективность.

В дальнейшем планируется выход подобного мобильного приложения для IOS.

Развитие приложения для ОС Андроид также не будет остановлено, планируется демонстрация стоимости разговора непосредственно на экране звонка.

Приложение «Русский телефон» доступно для загрузки пользователям платформы Android в магазине приложений Google Play.

Список использованных источников

1. Activity: Сайт для разработчиков Андроид [Электронный источник] // https://developer.android.com/guide/components/activities.html?hl=ru (дата обращения 08.11.2016)
2. Meet Android Studio: Сайт для разработчиков Андроид [Электронный источник] // https://developer.android.com/studio/intro/index.html (дата обращения 08.11.2016)
3. Анализ мобильного веб-трафика: Сервис веб-статистики «StatCounter» [Электронный источник] // gs.statcounter.com/ (дата обращения 08.11.2016)
4. Башмаков, А.И. Теория систем и системный анализ: методические указания по выполнению курсовой работы по курсу «Теория систем и системный анализ» по направлениям «Прикладная математика и информатика» и «Экономика» / А.И. Башмаков. – М.: Изд. дом МЭИ, 2010. – 52 с.
5. В Минкомсвязи решили запретить звонки со Skype на мобильные: Информационное агенство lenta.ru [Электронный источник] // https://lenta.ru/news/2016/05/20/skype_over/ (дата обращения 08.11.2016)
6. Введение в JSON: Сайт посвященный JSON [Электронный источник] // http://www.json.org/json-ru.html (дата обращения 08.05.2015)
7. Волкова, В.Н. Теория систем и системный анализ в управлении организациями: Справочник: Учебное пособие / В.Н. Волкова, А.А. Емельянова. – М.: Финансы и статистика, 2006. – 848 с.
8. Гвоздева, Т.В. Проектирование информационных систем: учебное пособие / Т.В. Гвоздева, Б.А. Баллод. – Ростов-н/Д: Феникс, 2009. – 508 с.
9. Глумаков, С.В. Базы данных: Учебный курс / С.В. Глумаков, Д.В. Ломотько. – М.:АСТ, 2002. – 504 с.
10. Голицына, О. Л. Базы данных: Учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – Форум, Москва – 2009. – 400 с.
11. Головач, В.В. Дизайн пользовательского интерфейса: учебное пособие / В.В. Головач. – М.: Наука, – 132 с.

12. Гультяев, А.К. Проектирование и дизайн пользовательского интерфейса / А.К. Гультяев, В.А. Машин. – СПб: КОРОНА принт, 2010. – 352 с.

13. Дженифер Тидвелл. Разработка пользовательских интерфейсов / книга / Переводчик Е. Шикарева. – Питер 2008. – 416 с.

14. Диго, С.М. Базы данных. Проектирование и создание: учебно-методический комплекс / С.М. Диго. – М:. Изд. центр ЕАОИ, 2008. – 172 с.

15. Дубейковский, В. И. Эффективное моделирование с AllFusion Process Modeler 4.1.4 и AllFusion PM.: Книга / В.И. Дубейковский. – Диалог-МИФИ, 2007. – 464 с.

16. Дэйтел, П. Android для разработчиков / Пол Дейтел, Харви Дэйтел, Александр Уолд. – СПб: Питер, 2016. – 512 с.

17. Запуск мессенджера библ: Новостной сайт macdigger.ru [Электронный источник] // http://www.macdigger.ru/iphone-ipod/rostelekom-osenyu-zapustit-messendzher-allyo-platnyj-analog-skype-dlya-ios-i-android.html (дата обращения 08.11.2016)

18. Какую библиотеку работы с HTTP в Android выбрать: IT ресурс Хабрахабр [Электронный источник] // https://habrahabr.ru/post/281965/ (дата обращения 08.11.2016)

19. Кириллов, В.В. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб: БХВ – Петербург, 2009. – 464 с.

20. Корнеев, В.В. Базы данных. Интеллектуальная обработка информации / В.В. Корнеев, А.Ф. Гареев, С.В. Васютин. – М.: Нолидж, 2004. – 496 с.

21. Кулябов, Д.С. Введение в формальные методы описания бизнес-процессов: Учебное пособие / Д.С. Кулябов, А.В. Королькова, 2008. – 173 с.

22. Маклаков, С.В. Создание информационных систем с AllFusionModelingSuite / С.В. Маклаков. – М.: Диалог-МИФИ, 2004. – 432 с.

23. Мартин, Р. Чистый код / Роберт Мартин. – СПб: Питер, 2010. – 464 с.

24. Маховикова, Г.А. МЕНЕДЖМЕНТ: УЧЕБНЫЙ КУРС / Г.А.

Маховикова, В.Е. Кантор. – Эксмо, 2013. – 312 с.

25. Мельников, В.П. Информационная безопасность и защита информации / В.П. Мельников, А.М. Петраков, С.А. Клейменов. – М.: ACADEMA, 2009. – 336 с.

26. Мишенин, А.И. Теория экономических информационных систем / А.И. Мишенин. – М.: Финансы и статистика, 2005. – 240 с.

27. Родичев, Ю.А. Информационная безопасность: Нормативно-правовые аспекты: Учебное пособие / Ю.А. Родичев. – СПб: Питер, 2008. – 272 с.

28. Ростелеком в декабре запустит собственный мессенджер Алле: Информационное агенство РИА НОВОСТИ [Электронный источник] // https://ria.ru/technology/20161110/1481093291.html (дата обращения 08.11.2016)

29. Сериализация в Java: IT ресурс «Хабрахабр» [Электронный источник] // https://habrahabr.ru/post/60317/ (дата обращения 08.11.2016)

30. Статистика популярности мобильных ОС: Сервис веб-аналитики [Электронный источник] // www.netapplications.com/ (дата обращения 08.11.2016)

31. Тарасенко, Ф.П. Прикладной системный анализ / Ф.П. Тарасенко. – М.: Изд. КноРус, 2010.- 454 с.

32. Титоренко, Г.А. Информационные системы в экономике: учебник / Г.А. Титоренко. – Юнити-Дана, 2012. – 463 с.

33. Харди, Б. Android. Программирование для профессионалов / Брайн Харди, Билл Филлипс. – СПб: Питер, 2016. – 640 с.

34. Хомоненко, А.Д. Базы данных: Учебник для высших учебных заведений / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев. – СПб: КОРОНА принт, 2011. – 672 с.

35. Черемных, С.А. Моделирование и анализ систем. IDEF-технологии. / С.А. Черемных, И.В. Семенов, В.Г. Ручкин. - М.: Финансы и статистика, 2006. – 192 с.

36. Что такое технология Java и каково ее применение: Официальный

сайт JAVA [Электронный источник] // https://www.java.com/ru/whatis_java.xml (дата обращения 08.11.2016)

37. Эккель, Б. Философия Java / Брюс Эккель. – СПб: Питер, 2016. – 1168 с.

38. Элементы модели «сущность-связь»: IP-блог citforum.ru [Электронный источник] // www.citforum.ru/database/ (дата обращения 08.11.2016)

39. Ярочкин, В.И. Информационная безопасность: Учебник для вузов / В.И. Ярочкин. – М.: Мир, 2008. – 640 с.