

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( Н И У « Б е л Г У » )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ СНЯТИЯ  
ПОКАЗАНИЙ СЧЕТЧИКОВ ВОДОСНАБЖЕНИЯ И  
ЭЛЕКТРОЭНЕРГИИ**

Выпускная квалификационная работа  
обучающегося по направлению подготовки 09.03.02. Информационные  
системы и технологии  
очной формы обучения, группы 07001408  
Петрова Ярослава Васильевича

Научный руководитель  
ст. преподаватель  
Удовенко И.В.

БЕЛГОРОД 2018

## РЕФЕРАТ

Разработка мобильного приложения для снятия показаний счетчиков водоснабжения и электроэнергии. – Петрова Ярослава Васильевича, выпускная квалификационная работа бакалавра, Белгород, ФГАОУ ВО "Белгородский государственный национальный исследовательский университет" (НИУ «БелГУ»), количество страниц 46, включая приложения 50, количество рисунков 17, количество использованных источников 30.

**КЛЮЧЕВЫЕ СЛОВА:** мобильное приложение, электроэнергия, программное обеспечение, снятие показаний счетчиков.

**ОБЪЕКТ ИССЛЕДОВАНИЯ:** энергосбытовая деятельность.

**ПРЕДМЕТ ИССЛЕДОВАНИЯ:** методика проведения контрольных считываний приборов учета.

**ЦЕЛЬ РАБОТЫ:** повышение эффективности работы сотрудников энергосбытовой компании за счет разработки мобильного приложения для снятия показаний счетчиков электроэнергии и водоснабжения.

**ЗАДАЧИ ИССЛЕДОВАНИЯ:** изучение предметной области и уже существующий программных и аппаратных решений; разработка спецификации создаваемого программного обеспечения (ПО); выбор аппаратно-программной платформы, а также языка программирования, СУБД и среды программирования; описание структуры разрабатываемого ПО; моделирование алгоритмов работы разрабатываемого ПО;

**МЕТОДЫ ИССЛЕДОВАНИЯ:** метод лексико-синтаксических шаблонов; статистические методы анализа текстов на естественном языке.

**ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ:** в результате работы было спроектировано и разработано приложение для снятия показаний счетчиков электроэнергии и водоснабжения.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Аналитическая часть.....	7
1.1 Общая характеристика энергосбытовой деятельности.....	7
1.2 Существующий метод проведения контрольно-измерительных мероприятий .....	12
1.3 Существующие решения .....	14
1.4 Постановка задачи .....	16
2 Моделирование программного обеспечения.....	18
2.1 Декомпозиция моделируемого решения .....	18
2.2 Проектирование базы данных .....	20
3 Реализация программного обеспечения .....	24
3.1 Выбор программной платформы.....	24
3.1.1 Выбор языка программирования.....	25
3.1.2 Выбор среды программирования .....	27
3.2 Разработка ПО .....	29
3.2.1 Создание структуры проекта .....	29
3.2.2 Создание структуры экранов приложения .....	30
3.3 Тестирование приложения .....	35
3.4 Минимальные технические требования .....	38
ЗАКЛЮЧЕНИЕ .....	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	41
ПРИЛОЖЕНИЕ А .....	44
ПРИЛОЖЕНИЕ Б.....	46

## ВВЕДЕНИЕ

На сегодняшний день во всех развитых странах мира экономика имеет рыночный характер. В связи с этим спрос конечного потребителя полностью формирует предложение рынка, что в свою очередь порождает конкуренцию между производителями товаров и услуг. Постоянная конкуренция стимулирует производителей к выработке все более экономически выгодных решений, которые помогут увеличить спрос на их продукцию и увеличить прибыль. Несомненно, понижение конечной стоимости продукта либо услуги является одним из наиболее привлекательных аргументов при выборе продукта в глазах потребителя, однако зачастую понижение конечной стоимости может иметь и негативные последствия. В стремлении к снижению цен на конечный продукт или услугу, производитель зачастую вынужден жертвовать качеством конечного продукта, либо увеличивать время на производство, что в конечном счете так же скажется на общей прибыли и как итог на возможности к последующему присутствию производителя на рынке. Данный факт вынуждает производителей всячески минимизировать производственные издержки, и здесь основным помощником является автоматизация.

Развитие ЭВМ позволило человеку переложить массу трудоемких задач на «плечи» машин и автоматизировать бесчисленное количество процессов. В данных реалиях бизнес стал одной из основных сфер применения ЭВМ, использование которых позволяет сократить временные издержки, уменьшить процент брака на производстве и улучшить сервис при оказании услуг.

Наряду с развитием электротехники и ЭВМ в частности, неизменно растет и потребление энергии. В связи с этим всё большие и большие объемы энергии ежеминутно генерируются и предоставляются конечному пользователю. Растущие объемы используемой клиентами энергии, требуют от производителей оперативного и точного учета, который позволит

эффективно поставлять, отслеживать и планировать выработку энергетических ресурсов.

Для осуществления контроля за предоставлением энергоресурсов конечному пользователю, поставщикам необходимо регулярно осуществлять контрольные считывания со счетчиков, которые в подавляющем своем большинстве не оснащены автоматизированными системами коммерческого учета электроэнергии (далее АСКУЭ). Низкий уровень использования счетчиков с АСКУЭ продиктован большой ценой таких устройств, а также необходимостью наличия техники для считывания показаний и интернет соединения. Вышеописанное делает применение счетчиков с АСКУЭ весьма затруднительным, и вынуждает поставщик энергии к использованию устаревших методов контроля потребления энергии путем привлечения дополнительных человеческих ресурсов, что приводит к дополнительным финансовым расходам. В свою очередь сам процесс контрольных считываний зачастую производится представителями компаний поставщиков энергии без использования средств ЭВМ. Такой подход к контрольно-измерительным мероприятиям накладывает дополнительные риски возникновения неточностей, а также ведет к появлению временных издержек на транспортировку и внесение контрольных измерений в базу данных для дальнейшей обработки. Косвенным образом, вышеперечисленные проблемы в конечном счете ведут к снижению общей прибыли предприятия, а значит и ухудшению положения на рынке. Исходя из вышесказанного можно сделать вывод о существовании проблемы учета энергии, решение которой поможет повысить рентабельность энергосбытовой деятельности предприятий.

Данная выпускная квалификационная работа призвана проанализировать и разработать решение вышеописанной проблемы путем автоматизации процесса контрольных замеров с помощью использования ЭВМ и специализированного программного обеспечения (ПО).

Таким образом, целью работы является повышение эффективности работы сотрудников энергосбытовой компании за счет разработки мобильного

приложения для снятия показаний счетчиков электроэнергии и водоснабжения.

В качестве задач данной работы можно выделить следующие:

- изучение предметной области;
- анализ существующих программных и аппаратных решений;
- разработка спецификации создаваемого ПО;
- выбор аппаратно-программной платформы, а также языка программирования, СУБД и среды программирования;
- описание структуры разрабатываемого ПО;
- моделирование алгоритмов работы разрабатываемого ПО;

Пояснительная записка выполнена на 46 страницах без приложения, и 50 с приложением, содержит 17 рисунков и приложение.

## 1 Аналитическая часть

### 1.1 Общая характеристика энергосбытовой деятельности

Энергосбытовая деятельность – это деятельность по реализации электрической и тепловой энергии потребителям. Компании, продающие электрическую и тепловую энергию, могут как быть посредниками, реализующими купленную энергию, так и иметь собственные производственные мощности [2].

Основными функциями энергосбытовых компаний являются:

- реализация электрической и тепловой энергии и мощности по тарифам, установленным постановлениями органов государственного регулирования тарифов в соответствии с действующим законодательством о государственном регулировании тарифов на электрическую и тепловую энергию;

- изучение состояния и перспективы развития рынка сбыта электрической и тепловой энергии и мощности, прогноз платежеспособного спроса на электрическую и тепловую энергию и мощность;

- заключение с потребителями электрической и тепловой энергии договоров энергоснабжения и контроль выполнения договоров в части режимов энергопотребления и выполнения потребителями своих договорных обязательств;

- разработку и реализацию мероприятий по побуждению потребителей к выполнению договорных обязательств, по ограничению и прекращению подачи энергоресурсов потребителям-неплательщикам, по выявлению фактов нарушения правил учета и потребления электрической и тепловой энергии, а также по пресечению безучетного потребления и хищения энергии;

- участие в разработке тарифов для потребителей электрической и тепловой энергии;

– обеспечение жесткого контроля энергопотребления организаций, финансируемых из бюджетов всех уровней.

По состоянию на 2017 г. в России функционируют более сотни энергосбытовых организаций. Энергосбытовая деятельность потенциально является одним из наиболее конкурентных сегментов электроэнергетической отрасли [4].

На сегодняшний день энергосбытовые компании являются связующим звеном в движении финансовых потоков между субъектами системы электроэнергетики и напрямую влияют на реализацию энергетических ресурсов конечному потребителю. Четкая организация энергосбытовыми компаниями расчетов с потребителями электроэнергии и своевременная оплата за неё напрямую влияет на финансовое состояние всех субъектов электроэнергетики, как электросетевых компаний, так и генерирующих компаний. Именно поэтому эффективность и результативность функционирования энергосбытовых компаний является важнейшей задачей для успешного функционирования всей отрасли.

Для энергосбытовых компаний, через которые проходят денежные потоки всех участников рынка, особую важность имеет четкая организация бизнеса. По оценке экспертов, средний уровень оплаты электроэнергии физическими лицами по стране составляет пока 65-70%, а с учетом того, что физические лица составляют значительную часть в отпуске электроэнергии, неоплаты населения представляют серьезную проблему для энергосбытов [5].

К числу неплательщиков относятся бюджетные потребители с недостаточными лимитами финансирования, предприятия сферы ЖКХ, управляющие компании многоквартирных домов, отдельные граждане должники, крупные промышленные предприятия, оказавшиеся в тяжелом финансовом положении из-за кризиса.

Низкая платёжная дисциплина потребителей является одной из критичных проблем, которая отражается в том числе, и на стоимости электроэнергии. По словам экспертов, в настоящий момент задолженность за



электроэнергию на розничном рынке составляет порядка 207 млрд. рублей, на оптовом – 47 млрд. рублей. В результате почти треть сбытовых компаний вынуждена работать в условиях недостатка кредитных ресурсов для покрытия кассовых разрывов. В целях решения этой проблемы Минэнерго РФ разработало законопроект, предусматривающий повышение размера пени за просрочку платежей, введение института обеспечения платежей для категорийных потребителей (включая финансовые гарантии), введение административной и уголовной ответственности, а также обязательство расторжения договоров аренды в случае неоплаты арендатором платежей за ресурсы.

На сегодняшний день отсутствует реальная возможность отпускать энергию только добросовестным потребителям, а также существует проблема дифференциации платежеспособности потребителей, препятствующая эффективной работе энергосбытовой компании на основе равных возможностей и справедливой конкуренции [6].

Таким образом, низкая платежная дисциплина в условиях неопределенности нормативно-правовой базы может привести к потере финансовой устойчивости энергосбытовых компаний, надежности на розничном и неплатежам на оптовых рынках электроэнергии.

Данный фактор оказывают прямое воздействие на эффективность функционирования энергосбытовых компаний и их развитие. Поэтому его нужно не только учитывать при осуществлении деятельности энергосбытовой компании, но и уметь минимизировать его влияния, что, несомненно, может привести к укреплению конкурентных позиций на рынке для каждой энергосбытовой компании.

В соответствии с концепцией реформирования электроэнергетической отрасли России, проводимой в целях привлечения инвестиций и улучшения финансово-экономического положения субъектов рынка, региональные энергетические компании с регулируемым государством ценообразованием на поставляемую потребителям электроэнергию разделены по видам

деятельности на генерирующие, сетевые и сбытовые. При этом генерирующие и энергосбытовые компании (ЭСК) рассматриваются в качестве конкурентных как на оптовом, так и на розничных рынках электроэнергии. Конкуренция как форма взаимодействия субъектов розничного рынка электроэнергии закреплена в федеральном законе «Об электроэнергетике» от 26 марта 2003 г. № 35-ФЗ: «потребитель свободен в выборе контрагента по договору купли-продажи, договору поставки электрической энергии» (статья 37, п.2)[7].

Вследствие обеспечения недискриминационного доступа к сетевой инфраструктуре, на розничных рынках электроэнергии образовалось несколько видов энергосбытовых организаций, подразделяющихся на 2 основные категории:

- гарантирующие поставщики (ГП) — сбытовые организации с регулируемым государством ценообразованием и обязанностью заключения публичных договоров с потребителями, расположенными в соответствующей зоне обслуживания;

- независимые сбытовые компании (НСК), осуществляющие продажу другим лицам электрической энергии по свободным, нерегулируемым ценам.

Таким образом, потребитель получает право выбора энергосбытовой компании, но при этом всегда должна быть такая сбытовая компания, которая обязана его обслужить. Законодательством предусмотрено, что статус гарантирующего поставщика должен присваиваться по результатам открытого конкурса, который необходимо проводить один раз в 3 года.

Альтернативой деятельности энергосбытовых организаций являются самостоятельный выход и последующая работа потребителей на оптовом рынке электроэнергии.

Обособление сбытового бизнеса в отдельные компании обусловлено рядом следующих факторов: во-первых, специализированные сбытовые компании более четко выполняют функцию сбора денег с потребителей. Во-вторых, энергосбытовая компания берет на себя огромное количество рисков

(изменение цен на рынке, неплатежеспособность или банкротство отдельных потребителей), «очищая» от них более консервативные генерацию и передачу электрической энергии. В-третьих, сбытовой бизнес, отделенный от сетевого, перестает быть местным, локальным, привязанным к определенному собственнику сети, позволяет выходить на новые рынки, как в части покупки электроэнергии, так и в части сбыта (это возможно только в объединенной высоковольтной сети) [4].

Энергосбытовая деятельность с потребителями любой категории включает в себя:

- договорную работу (заключение, переоформление и расторжение договоров);
- организацию расчетов за потребленную энергию;
- деятельность по снижению коммерческих потерь (хищений);
- ограничение и прекращение подачи энергии потребителям-неплательщикам;
- организацию работ по замене и модернизации приборов учета электрической энергии;
- юридическую работу.

Важным отличием энергосбытовой компании от других участников — генерирующих и сетевых компаний, является отсутствие на балансе материальных активов. Устойчивость конкурентных преимуществ компаний, не обладающих материальными активами, определяется их внутренним потенциалом и способностями, включая эффективную систему управления и социальный капитал. Для российских ЭСК, особенно, выделившихся из АО-энерго, в большинстве случаев характерна невысокая финансовая устойчивость и ограниченные возможности по привлечению инвестиций в свое развитие. Одновременно эти компании обладают накопленными ранее знаниями приоритетов и особенностей конечных региональных потребителей и социальным капиталом в виде сложившихся отношений, связей с потребителями и другими стейкхолдерами на региональном рынке, которые

свидетельствуют о том, что у ЭСК имеется потенциал для развития и источник конкурентных преимуществ.

## 1.2 Существующий метод проведения контрольно-измерительных мероприятий

На сегодняшний день мероприятия по снятию показаний счетчиков, зачастую, мало чем отличаются от процедуры, проводимой несколько десятилетий назад. Контролер энергонадзора всё так же, как и ранее вынужден совершать обходы по заранее утвержденным маршрутам, вручную снимая показания счетчиков и внося их в соответствующие бумажные формы [1].

На рисунке 1 можно увидеть контекстную диаграмму, отображающую описанный выше метод проведения контрольно-измерительных мероприятий.

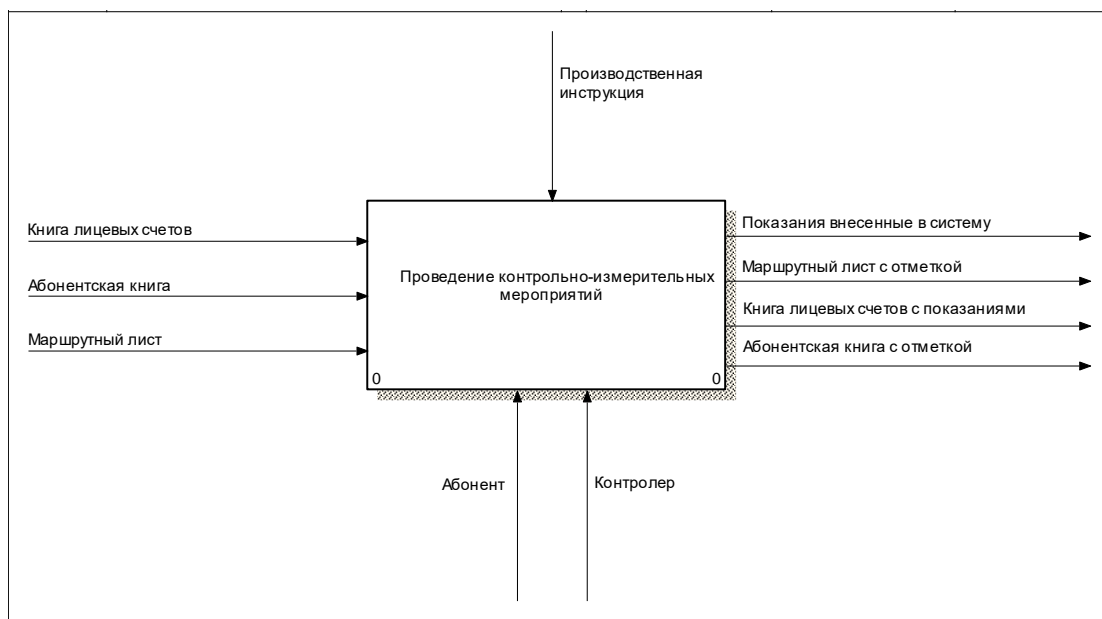


Рисунок 1 – Контекстная диаграмма AS\_IS

На приведенной выше контекстной диаграмме, можно увидеть, что в процессе проведения контрольно-измерительных мероприятий, на данный момент всё ещё участвует большое количество бумажных носителей [8].

Ознакомиться с более детальной декомпозицией приведенной диаграммы можно на рисунке 2.

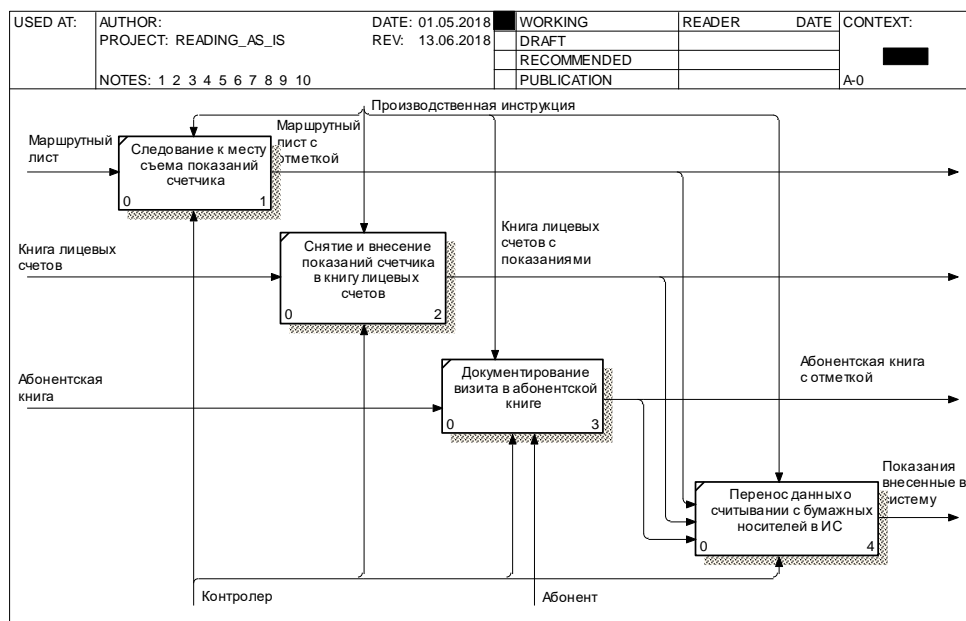


Рисунок 2 – Декомпозиция контекстной диаграммы AS\_IS

Из рисунка 2 видно, что помимо использования бумаги в качестве носителя, контролеру так же приходится осуществлять ручной перенос собранных данных в информационную систему (ИС) сбытовой компании.

Приведенный метод работы контролеров имеет как явные преимущества и не менее явные недостатки. К достоинствам данного подхода несомненно можно отнести относительную дешевизну главного расходного материала – бумаги. Так же устоявшийся подход весьма нетребователен к оснастке самого контролера, которому зачастую будет достаточно обычной шариковой ручки. Однако устоявшийся подход имеет и ряд следующих недостатков:

- отсутствие возможности оперативной передачи информации в сбытовую компанию;
- возможные искажения информации, вызванные ошибками контролера при их снятии;
- отсутствие возможности контроля оперативного персонала;

– возможность потери данных при переносе с бумажных носителей в электронный вид [3].

Перечисленные выше недостатки ведут к появлению таких проблем как нерациональное использование рабочих возможностей сотрудников энергосбытовой компании и понижению качества предоставляемых услуг, что так же косвенно ведет к падению прибыли предприятия.

### 1.3 Существующие решения

На данный момент проблема устаревшего подхода к сбору данных с приборов учета решается одновременно несколькими компаниями.

Одним из главных действующих лиц в данной области является компания Motorola Solutions которая представляет целую линейку приборов для снятия показаний счетчиков. Данные приборы обладают всеми необходимыми для работы функциями, однако стоит заметить, что ПО для снятия показаний продается только вкупе с самим устройством, которое зачастую является малопроизводительным и стоит довольно дорого. В качестве примера можно привести модельный ряд Motorola Symbol. Все устройства данной линейки представляют из себя КПК различных размеров с предустановленным специализированным ПО. Одним из наиболее распространенных из данной линейки является модель ТС70. Данное устройство размером 161 мм (Д) x 84 мм (Ш) x 28 мм (Г) с экраном размером 4,7 дюйма защищено по стандарту IP67, располагает 1ГБ RAM и 8 ГБ ROM, работает под управлением специальной версии ANDROID AOSP 4.4.2 с предустановленным ПО от Motorola. Данная модель так же располагает довольно маломощным 2х ядерным процессором с частотой 1,7 ГГц. Внешний ТС70 представлен на рисунке 3.



Рисунок 3 – Motorola Symbol TC70

С одной стороны, данное устройство является полноценным продуктом, готовым к использованию, однако стоимость данного устройства составляет около 120 000 руб. Устройства, схожие по характеристикам с данным стоят в районе 15 000 руб., что в среднем меньше на 100 000 руб. Отсюда следует вывод что цена данного решения является несколько завышенной.

Рассматривая ПО для снятия показаний, распространяемое свободно от устройств следует в первую очередь говорить о ПО от компании Syclo. Данная компания достигла определенных успехов в области контроля измерительных приборов и на данный момент может предложить такой продукт как Smart Work Manager For Energy Delivery. Данное ПО обладает функционалом для снятия показаний с любых видов счетчиков, а также механизмами передачи данных прямо в БД ИС. Однако приобретая данное решение стоит помнить, что оно требует наличия лицензии SAP, а также использования самой системы SAP R/3 в качестве ИС предприятия. Здесь же стоит упомянуть что стоимость внедрения SAP R/3 на предприятии весьма высока и начинается от 400 000

долларов. Необходимо так же упомянуть что компания SAP самолично производит партнерское обучение сотрудников, стоимость которого начинается от 2500 долларов. Внешний вид Smart Work Manager For Energy Delivery на различных устройствах продемонстрирован на рисунке 4.



Рисунок 4 – Smart Work Manager For Energy Delivery

Как можно заметить приложение работает на устройствах как под управлением Android так и iOS.

#### 1.4 Постановка задачи

Исходя из недостатков существующего метода проведения контрольно-измерительных мероприятий, описанных в подразделе 1.2 данного раздела можно сделать вывод о существующей необходимости модернизации метода. В качестве основного инструмента модернизации существующего подхода является использование ЭВМ для повышения эффективности работы сотрудников энергосбыта и уменьшения количества ошибок при контрольных



считываниях. Предлагается разработать ПО, для достижения следующих целей:

- снижения объемов ручного ввода показаний и обработки бумажных носителей и, как следствие, повышение эффективности работников энергосбыта;
- повышения достоверности данных;
- повышение оперативности перерасчета объемов потребления;
- контроль оперативного персонала.

Разрабатываемое ПО должно обладать следующими возможностями:

- возможность ручного ввода показаний;
- возможность фотофиксации показаний приборов учета с регистрацией геолокации;
- возможность автоматической передачи показаний в ИС сбытовой компании;
- возможность загрузки и отображения маршрутных листов;
- возможность внесения информации о составленных в ходе контрольно-измерительных мероприятий актах.

#### Вывод по первому разделу

В первом разделе была рассмотрена предметная область энергосбыта. В рамках исследования предметной области было установлено что контроль показаний приборов является одной из главных задач энергосбытовых компаний. Так же было установлено что в существующей методике снятия показаний существуют определенные проблемы. Были рассмотрены существующие аппаратно-программные решения являются весьма дорогостоящими. Заключительным же этапом главы стала постановка задачи ВКР.

## 2 Моделирование программного обеспечения

### 2.1 Декомпозиция моделируемого решения

Первичным заданием при проектировании решения является моделирование целевого состояния метода проведения контрольно-измерительных мероприятий после модернизации. Это позволит наиболее точно рассмотреть объект модернизации и выделить модернизируемые процессы. Контекстную диаграмму декомпозиции можно увидеть на рисунке 5.

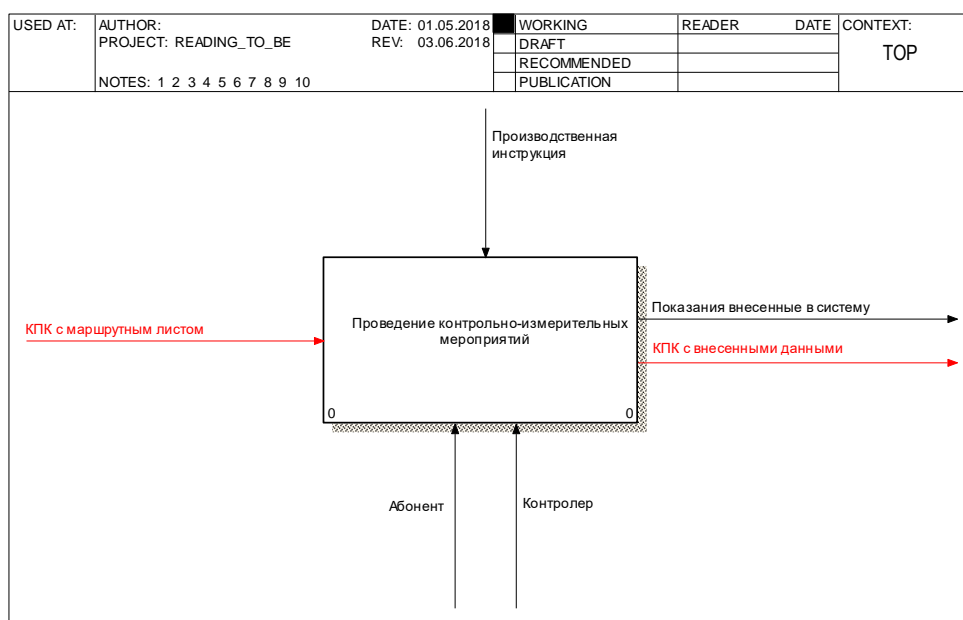


Рисунок 5 – Контекстная диаграмма ТО\_ВЕ

Проанализировав контекстную диаграмму решения на рисунке 5 можно заметить отсутствие бумажных носителей в модернизируемом процессе. В остальном же данный процесс не претерпел значительных изменений. Декомпозиция контекстной диаграммы, приведенная на рисунке 6 позволяет увидеть изменения более детально [9].

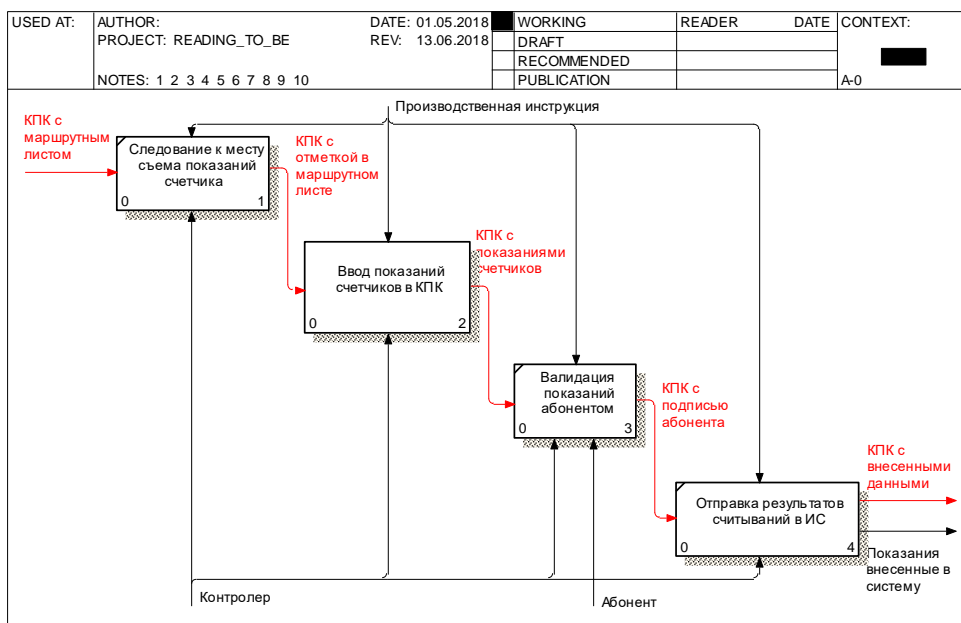


Рисунок 6 – Декомпозиция контекстной диаграммы TO\_BE

Первый функциональный блок «Следование к месту съема показаний» не претерпел изменений, однако теперь на вход вместо бумажного носителя подается «КПК с маршрутным листом». Это позволило исключить маршрутный лист в бумажном виде из процесса [10]. Соответственно следующий по счету блок «Ввод показаний в КПК» так же был освобожден от бумажных носителей и далее будет рассмотрен подробнее. Следующий блок «Валидация показаний абонентом» по сути остался неизменен, однако теперь роспись абонента будет приниматься так же посредством КПК, без использования бумажных носителей. Заключительный блок «Отправка результатов считываний в ИС» позволяет избежать ручного переноса показаний в систему тем самым повысив достоверность и скорость передачи данных в ИС. Поскольку наиболее изменившимся в результате модернизации процесса функциональным блоком является «Ввод показаний в КПК», будет целесообразно рассмотреть его детальнее. Декомпозиция данного блока приведена на рисунке 7.

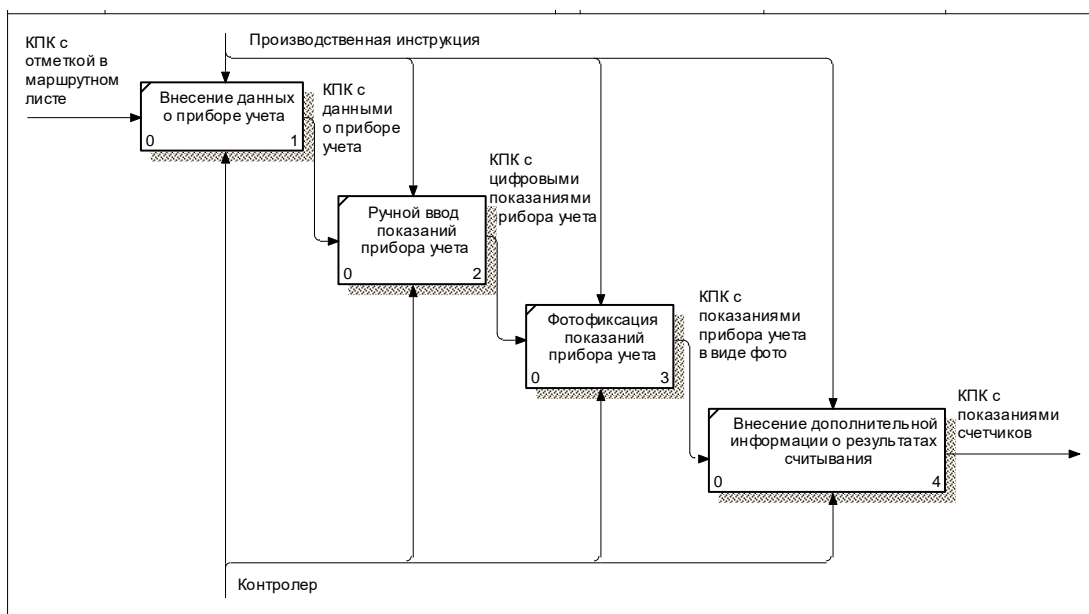


Рисунок 7 – Декомпозиция блока «Ввод оказаний счетчиков» в КПК

Первым функциональным блоком данной декомпозиции является «Внесение данных о приборе учета». На данном этапе контролер вносит в КПК такие данные как тип прибора учета, его номер и разрядность. Следующий блок «Ручной ввод показаний прибора учета» описывает одноименное действие, в результате которого в КПК сохраняются данные для последующей передачи в ИС. Следующий блок «Фотофиксация показаний прибора учета» направлен на добавление фото с показаниями к перечню передаваемых в ИС данных. Последний блок данной диаграммы «Внесение дополнительной информации о результатах считывания» направлен, к примеру, на внесение таких данных, как информация о неисправности прибора [11].

## 2.2 Проектирование базы данных

Поскольку проектируемое ПО для снятия показаний счетчиков предназначено для оперативной отправки показаний в ИС, может показаться логичным хранение отказ от использования БД [12]. Однако данный подход имеет ряд недостатков и ограничений. К примеру, если работа устройства

будет аварийно завершена, то все данные будут утеряны. Так же устройство может оказаться отрезано от сети интернет и не сможет своевременно передать данные, а в последствии увеличится и шанс потери данных[13]. Во избежание возникновения подобных ситуаций необходимо использование локальной БД, которая будет хранить все данные на устройстве[14]. Структура спроектированной БД приведена на рисунке 8.

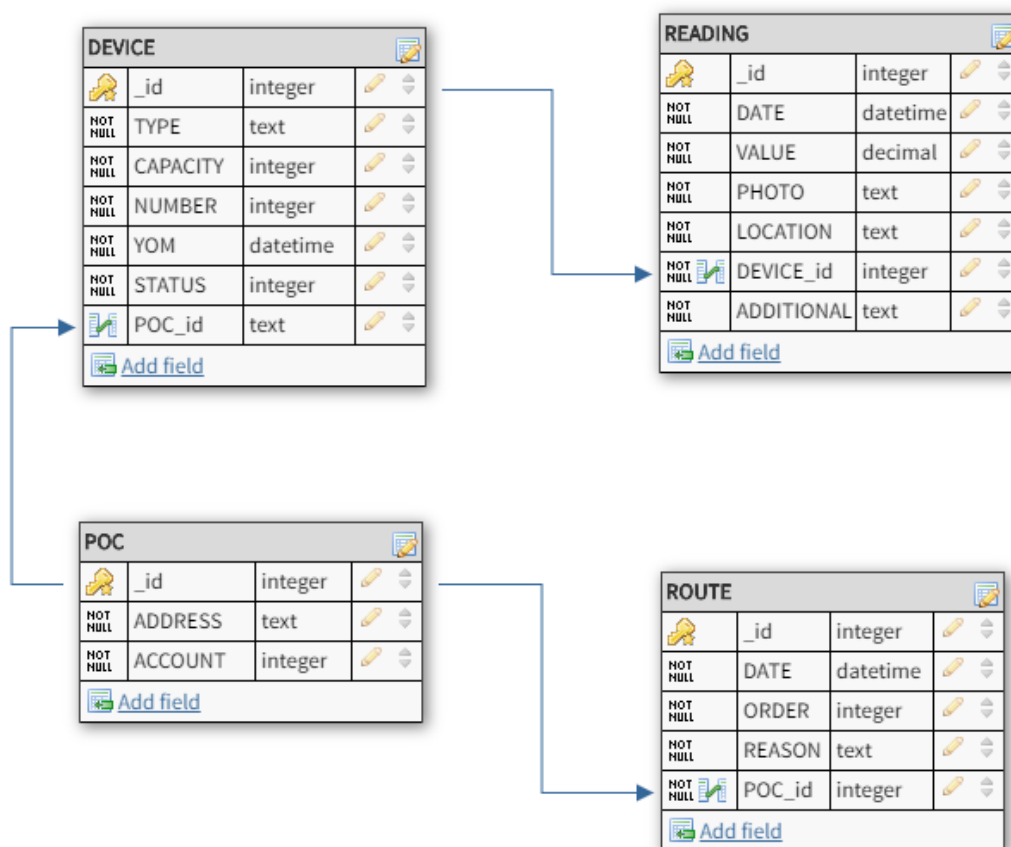


Рисунок 8 – Структура БД

Первая таблица «READING» предназначена для хранения показаний счетчиков и содержит следующие поля:

- **\_id** – данное поле является первичным ключом таблицы и предназначено для хранения целочисленного значения;
- **DATE** – назначением данного поля является хранение даты контрольного считывания в формате «ДДММГГГ:ЧЧММСС»;

- VALUE – данное поле предназначено для хранения непосредственно значения счетчика в формате числа с заданной точностью;
- PHOTO – поле, предназначенное для хранения расположения файла изображения, которое хранится в текстовом виде;
- LOCATION – данное поле предназначено для хранения геоданных в текстовом формате;
- DEVICE\_id – данное поле является внешним ключом с целочисленным форматом, по которому данная таблица связывается с таблицей «DEVICE»;
- ADDITIONAL – данное поле предназначено для хранения дополнительной информации, которая хранится в текстовом виде.

Вторая таблица «DEVICE» отвечает за хранение информации о счетчиках и содержит следующие поля:

- \_id – данное поле является первичным ключом таблицы и хранится в целочисленном формате;
- TYPE – назначением данного поля является хранения информации о типе счетчика в текстовом виде;
- CAPACITY – данное поле хранит разрядность счетчика в формате целого числа;
- NUMBER – поле, предназначенное для хранения номера счетчика в целочисленном виде;
- YOM – данное поле призвано хранить дату выпуска счетчика в формате «ДДММГГГ:ЧЧММСС»;
- STATUS – данное поле предназначено для хранения статуса счетчика в формате целого числа;
- ROC\_id – данное поле является внешним ключом целочисленного формата, по которому данная таблица связывается с таблицей «РОС».

Третья таблица «РОС» предназначена для хранения информации о месте расположения счетчика и контокоррентном счете к которому он привязан [14]. Содержит следующие поля:

- `_id` – поле, являющееся первичным ключом таблицы и хранящееся в формате целого числа;
- `ADDRESS` – данное поле предназначено для хранения текстовой информации об адресе по которому находится счетчик;
- `ACCOUNT` – данное поле содержит целочисленную информацию о номере контокоррентного счета абонента.

Четвертая таблица «`ROUTE`» предназначена для хранения информации о маршрутном листе и содержит следующие поля:

- `_id` – поле, которое является первичным ключом таблицы и хранится в целочисленном виде;
- `DATE` – поле в котором в формате `ДДММГГГ:ЧЧММСС` хранится дата маршрутного задания;
- `ORDER` – данное поле хранит порядковый номер задания в формате целого числа;
- `REASON` – данное поле предназначено для хранения причины включения прибора учета в маршрутный лист;
- `POC_id` – данное поле является внешним ключом целочисленного формата, которое используется для связи с таблицей «`POC`».

#### Вывод по второму разделу

Подходя к логическому завершению данного раздела стоит отметить, что в нем был рассмотрен вариант улучшения методики снятия показаний счетчиков, в рамках которого был описан новый алгоритм работф – ввод показаний в КПК. В целях повышения надежности ПО была смоделирована локальная база данных, которая позволяет временно хранить информацию о считываниях, для дальнейшей их передачи в ИС сбытовой компании.

### 3 Реализация программного обеспечения

#### 3.1 Выбор программной платформы

Сегодняшний рынок мобильных программных платформ возглавляют два гиганта Android и iOS. Данные операционные системы практически полностью формируют весь рынок мобильных платформ во всем мире и суммарно занимают по разным оценкам от 97% до 99% всего рынка [16]. Согласно исследованию Gartner, по состоянию на конец 2016 года около 88% всех продаваемых в мире смартфонов это доля Android [17]. Доля же продаваемых устройств основного конкурента под управлением iOS составляет порядка около 11%. Остальные же представители занимают весьма незначительный процент рынка, который к тому же ежегодно сокращается. В данных реалиях наиболее рациональным станет выбор одной из двух лидирующих систем.

Выбирая между Android и iOS следует учитывать высокий темп развития обеих платформ, и как результат довольно схожие показатели в таких областях как производительность, безопасность и удобство использования. Говоря об iOS нужно понимать, что Apple в первую очередь пытается предоставить пользователю цельный продукт, как с аппаратной, так и с программной точки зрения. Apple располагает собственными производственными мощностями в аппаратном сегменте и стремится распространять свою платформу iOS исключительно на устройствах собственного производства. Сама же iOS является частично открытой мобильной платформой и пресекает изменение внутреннего устройства системы. Android же напротив является платформой с открытым исходным кодом, и лояльно относится к модификациям системы. Это позволяет подстраивать платформу под свои нужды, реализуя, либо наоборот упраздняя тот или иной функционал. К тому же Android является аппаратно-независимой платформой и повсеместно распространяется не только на устройствах от Google, но и сторонних компаний [18].



Сравнивая стоимости устройств под управлением Android и iOS можно утверждать, что устройства с ОС от Google зачастую стоят на порядок меньше своих аппаратных аналогов от Apple. В данном вопросе свою роль играет огромный рынок устройств от сторонних производителей под управлением Android, в то время как Apple является единственным производителем устройств под управлением iOS [19]. Исходя из данного факта можно утверждать, что устройства под управлением Android могут покрывать более обширные потребности покупателей, в виду большего количества устройств на данной платформе.

Исходя из вышеописанного можно сделать вывод о преимуществах платформы Android над iOS в разрезе текущей предметной области, а именно:

- открытость кода и возможность модификации платформы;
- широкий выбор устройств под управлением Android;
- низкая относительно устройств от Apple стоимость устройств;
- большой рынок используемых устройств на базе Android.

### 3.1.1 Выбор языка программирования

Подходя к выбору языка программирования для создания приложений на платформе Android стоит учитывать, что официально Google поддерживаются 4 языка программирования для разработки Android приложений.

Первый, он же основной язык разработки Android приложений это Java. Данный язык программирования был выбран Google в качестве основного не случайно. Изначально Google преследовала идею независимости программного обеспечения от аппаратной архитектуры, и язык Java стал идеальным решением воплощения этой идеи в жизнь. Язык Java позволяет создавать приложения, которые можно запускать на любой аппаратно-программной платформе где установлена JVM, независимо от ОС или устройства что делает его по настоящему мульти-платформенным [27].

Данный язык заслуженно является одним из самых популярных в мире, и тому есть несколько причин:

- Java строго типизирован, что избавляет его от смешивания в выражениях различных типов и не позволяет производить неявных преобразований[28];
- обладает лаконичным и понятным синтаксисом;
- позволяет создавать аппаратно-независимые приложения;
- является мультипарадигмальным языком, что позволяет писать на нем в различных стилях[29];
- сопровождается обширной документацией и обладает одним из самых больших сообществ разработчиков;
- поддерживает вставки кода, написанного на других языках программирования;
- обладает богатой стандартной библиотекой;
- исходный код Java является открытым[30].

Вторым и третьим языком для написания Android приложений являются C и C++ [20]. Однако стоит отметить что компания Google предоставляет для разработки на данных языках только NDK. Поэтому применение данных языков не подразумевает создания целостного приложения только с использованием C/C++. Несмотря на это, с помощью данных языков возможно написание подключаемых библиотек, которые могут быть использованы при помощи Java и JNI. Такое с виду запутанное использование C/C++ способно дать значительный выигрыш в производительности и позволяет грамотнее распорядиться ресурсами устройства.

Последним же официально поддерживаемым Google языком является относительно новый высокоуровневый язык программирования Kotlin. Прежде всего Kotlin это статически типизированный язык программирования, который, как и Java работает поверх JVM. Данный язык весьма схож, а также полностью совместим с уже упомянутым Java. Несмотря на свою молодость

язык может похвастать растущей популярностью, которая вызвана прежде всего следующими положительными качествами:

- Kotlin компилируется не только в байт код Java, но и в код JavaScript;
- полностью совместим с Java, что делает возможным использование обширных библиотек и фреймворков Java;
- имеет открытый исходный код, поэтому можно легко найти, обозначить проблему в случае её возникновения и сообщить об этом разработчикам языка;
- небольшой объем репозитория, который сопоставим по размерам со стандартными вспомогательными библиотеками от Google;
- обладает иммутабельностью по умолчанию;
- является языком с повышенным вниманием к null объектам, что позволяет избежать ошибок обращения в «пустоту».

Однако разработка для платформы Android не ограничивается только лишь четырьмя вышеперечисленными языками. Существуют библиотеки и фреймворки, позволяющие использовать для разработки такие языки как Python, C#, Lua.

Исходя из вышеперечисленного наиболее целесообразным является использование языка Java в силу его положительных качеств.

### 3.1.2 Выбор среды программирования

В мире Java разработки существует множество действительно хороших сред разработки, однако официально используемой и наиболее распространенной для платформы Android является официальная среда разработки от Google – Android Studio.

Данная среда разработки обладает внушительным объемом полезных функций, среди которых в первую очередь стоит упомянуть помощник создания графической составляющей приложения – Visual layout editor. VLE

позволяет создавать визуальные представления любых стилей и направленностей, при этом позволяя разработчику пользоваться как текстовым редактором так и визуальным, используя drag'n'drop. Данный помощник позволяет просматривать моделируемое визуальное представление на экранах любой с любым разрешением и ориентацией, при помощи гибких инструментов предварительного просмотра[21].

Следующим не менее значительным преимуществом Android Studio является наличие встроенного эмулятора Android устройств. Данный эмулятор позволяет смоделировать работу абсолютно любого Android устройства с различными аппаратными и программными характеристиками[22]. Это позволяет разработчику отладить приложение для любой версии платформы Android, что делает разработку гораздо удобнее и позволяет избежать проблем с совместимостью на разных устройствах.

Android Studio обладает технологией Intelligent code editor, которая позволяет разработчику меньше внимания уделять проблемам синтаксиса и ускоряет разработку[23]. Данная технология поддерживает все четыре официальных языка разработки для Android и позволяет писать хорошо структурированный код, помогая разработчику не только в исправлении синтаксических ошибок, но и помогает управлять подключениями различных библиотек.

Данная среда разработки так же обладает гибкой системой сборки проектов – Gradle. Данная система позволяет управлять зависимостями проекта, подключать сторонние библиотеки и фреймворки, а также настраивать параметры запуска и работы приложений. Gradle способен генерировать APK файлы полностью готовые к установке на устройстве[24].

В Android Studio встроена система автоматического отслеживания нагрузки на аппаратную часть устройства. Данная система позволяет в режиме реального времени отслеживать нагрузки на центральный процессор устройства, его оперативную и постоянную память, а также на модули

передачи данных. Это позволяет избежать узких мест в работе приложения, а также служит дополнительным средством оптимизации.

## 3.2 Разработка ПО

### 3.2.1 Создание структуры проекта

Первым этапом создания приложения является создание проекта в Android Studio. Данный этап предполагает указание целевой версии платформы Android на которой будет возможно использование разрабатываемого приложения. Так же на данном этапе указываются название проекта и возможность поддержки модулей, написанных на других языках программирования.

После создания нового проекта, средой разработки в автоматическом режиме генерируется строгая структура проекта, которая является необходимой для работы любого Android приложения. Конечная структура разрабатываемого ПО приведена на рисунке 9.

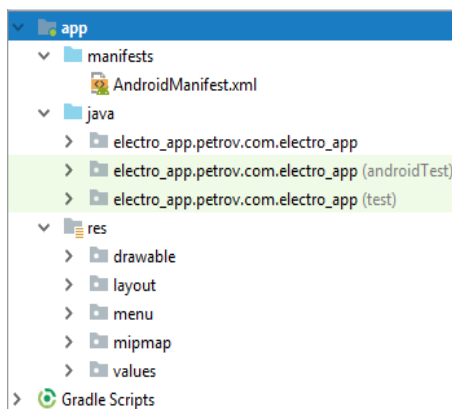


Рисунок 9 – Структура проекта

Корневой папкой проекта является папка app, в которой содержатся все ресурсы и бизнес логика приложения.

Первой папкой, располагающейся непосредственно в каталоге `app` является папка `manifests`. В данном каталоге хранятся файлы манифестов, в которых содержатся основные сведения о приложении, такие как целевая версия Android, `intent` фильтры для главного `activity` приложения и многое другое. Стоит отметить что работа приложения невозможна без наличия `manifest` файла.

Следующим каталогом, входящим в состав `app` является подкаталог `java`. В данном каталоге хранится исходный код, написанный на `java`. Собственно, все классы и интерфейсы приложения хранятся в подкаталоге `electro_app.petrov.com.electro_app`. Два каталога со схожим именем но дополнительными префиксами `android_test` и `test` служат для хранения описанных автоматических тестов.

Последней же папкой в каталоге `app` является подпапка `res`. Данная папка предназначена для хранения ресурсов приложения, таких как XML файлы разметки, константы и изображения.

Вторым каталогом, находящимся на уровне с каталогом `app` является `Gradle Scripts`. В данном каталоге хранятся параметры запуска приложения, а также файлы сборки проекта.

Подробное представление структуры проекта представлено в приложении А.

### 3.2.2 Создание структуры экранов приложения

В Android приложениях логика построена таки образом, что каждый отдельный экран является независимым модулем программы, который отвечает за исполнения возложенной на него бизнес логики. Главным же связующим звеном, а также точкой входа в приложения служит отдельный экран – главная активность (англ. `MainActivity`). В данной активности описана логика вызовов других компонентов приложения и именно данную активность видит перед собой пользователь при запуске приложения. Графическое

представление главной активности, разрабатываемого приложения представлено на рисунке 10.

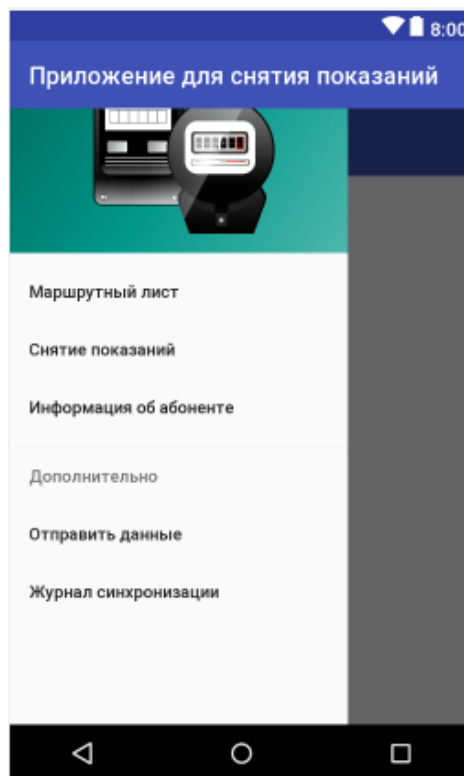


Рисунок 10 – Главная активность

Экран главной активности представляет собой совокупность из зоны для отображения результатов работы модулей, а также выдвижного меню, каждый пункт которого вызывает требуемый модуль приложения. Само же выдвижное меню расположено над экраном отображения и предстает перед пользователем в открытом виде сразу после запуска приложения. Основная логика главного экрана состоит в том, чтобы при нажатии пользователем на элемент меню, вызвать соответствующий модуль (Fragment, применительно к Android) и отобразить его в зоне под меню. Таким образом, главный экран служит своего рода контейнером для отображения других экранов внутри себя. Логика работы главного экрана описана в java классе «MainActivity», в то время как графическое представление описано в «activity\_main.xml».

Первым элементом главного меню программы является «Маршрутный лист», который вызывает одноименный модуль (Fragment), представленный на рисунке 11.

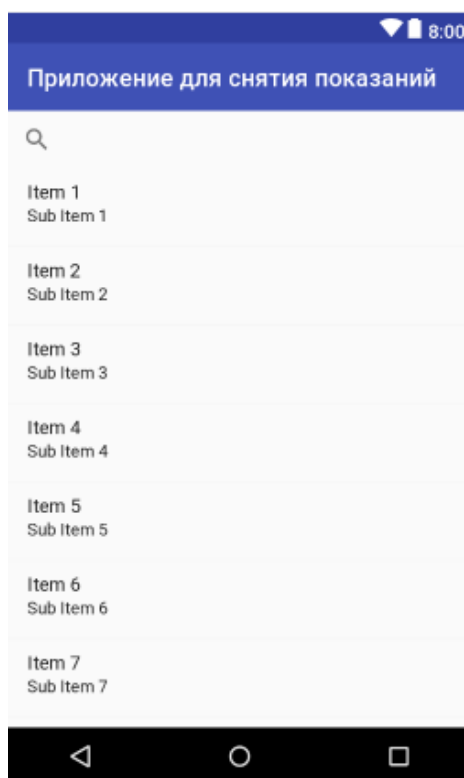


Рисунок 11 – Экран «Маршрутный лист»

Данный модуль предназначен для отображения маршрутного листа контролера, в соответствии с которым будет осуществляться плановый обход абонентов. Данный экран представлен списком, в котором отображаются адреса абонентов для посещения, а также строкой поиска. Помимо своей основной функции отображения адресов абонентов, данный экран так же служит точкой перехода к снятию показаний прибора и получению информации об абоненте. Логика данного экрана заключается в следующем – пользователь выбирает адрес из предложенного в списке, после чего ему предлагается либо перейти непосредственно к снятию показаний, либо просмотреть информацию об абоненте.



Следующим пунктом главного меню, а также вариантом перехода с экрана «Маршрутный лист» является экран «Снятие показаний». Логика данного экрана описана в java классе «ReadingFragment» а его графическое представление в «fragment\_reading.xml», которое представлено на рисунке 12.

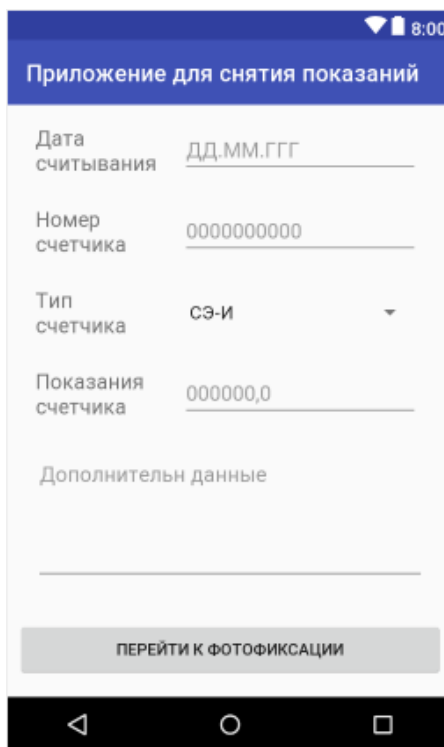


Рисунок 12 – Экран «Снятие показаний»

Данный экран предназначен для внесения показаний счетчика сотрудником сбытовой компании. Для ввода информации на экране представлены соответствующие поля, которые снабжены описанием, а также подсказками о формате ввода информации. Логика данного экрана заключается непосредственно в вводе показаний счетчика, с предварительным указанием типа и номера прибора. После осуществления ввода, пользователь, нажатием на кнопку «Перейти к фотофиксации» подтверждает ввод данных и переходит к использованию камеры. После осуществления фотосъемки, данные введенные на экране, а также фото и данные о текущем местоположении сохраняются в локальную базу данных устройства для последующей отправки [25].

Третьим пунктом главного меню является «Информация об абоненте», который описан java классом «InfoFragment» и XML файлом «fragment\_info.xml». Графическое представление данного модуля представлено на рисунке 13.



Рисунок 13 – Экран «Информация об абоненте»

Данный экран предназначен для просмотра информации об абоненте и представлен полями для ее отображения, а также кнопкой для их получения. Логика данного экрана весьма проста – после нажатия на кнопку «Обновить», модулем формируется запрос к БД, который передается в ИС энергосбытовой компании. После получения данных по абоненту, который был выбран по адресу из маршрутного листа, экран отобразит данные об абоненте.

Дополнительными элементами главного меню являются «Отправить данные» и «Журнал синхронизации». Первый служит для отправки данных о считывании в ИС предприятия, в то время как второй служит формой, для регистрации операции отправки на сервер.

### 3.3 Тестирование приложения

В ходе тестирования будут основные экраны приложения на предмет исполнения заложенной в них бизнес логики.

Первым шагом будет непосредственно запуск приложения, после которого перед пользователем открывается главное меню программы. Данный шаг представлен на рисунке 14.

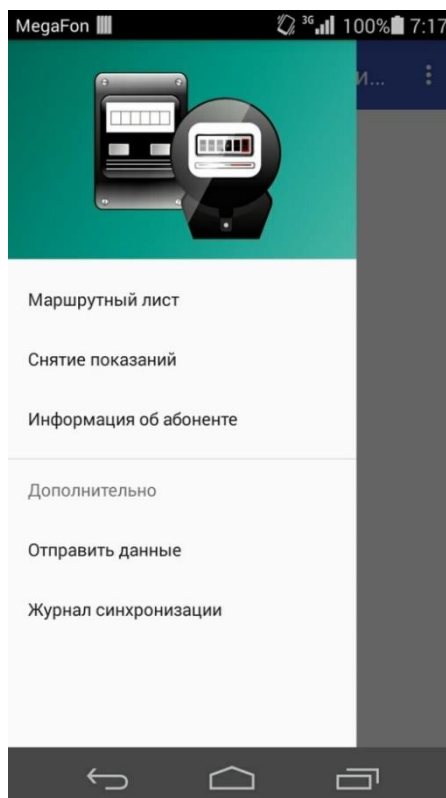


Рисунок 14 – Тестирование экрана «Главное меню»

Как можно заметить из рисунка 14, ранее смоделированное меню отображается правильно. Единственным отличием является положение самого меню относительно зоны отображения фрагментов. При работе приложения верхняя часть меню полностью занимает зону отображения фрагментов.

Далее необходимо запустить модуль «Маршрутный лист» нажав на соответствующем пункте главного меню. После нажатия перед пользователем открывается экран маршрутного листа, что проиллюстрировано на рисунке 15.



Рисунок 15 – Тестирование экрана «Маршрутный лист»

Проанализировав рисунок 15 можно заметить, что адреса маршрутного листа, как и название самого экрана отображаются корректно.

Для перехода к следующему экрану «Снятие показаний» необходимо выбрать адрес из списка, находящегося в маршрутном листе. После выбора адреса, будет открыт экран снятия показаний счетчика, что отображено на рисунке 16.

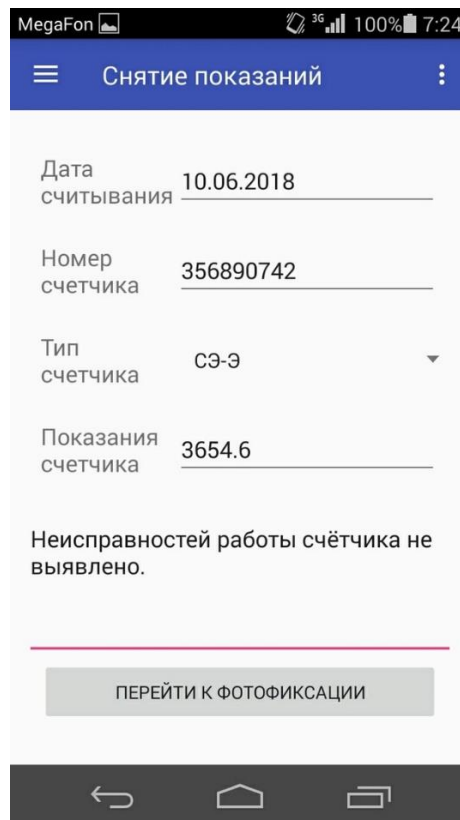


Рисунок 16 – Тестирование экрана «Снятие показаний»

На приведенном выше рисунке изображен экран снятия показаний с данными, введенными пользователем [26]. При вводе информации не было выявлено дефектов, а кнопка правильно реагирует на воздействие. В связи с вышесказанным можно считать работу данного экрана корректной.

Последним же тестируемым экраном является «Информация об абоненте». К данному экрану можно перейти как с экрана «Маршрутный лист», так и из главного меню программы. Результат вызова данного экрана изображен на рисунке 17.

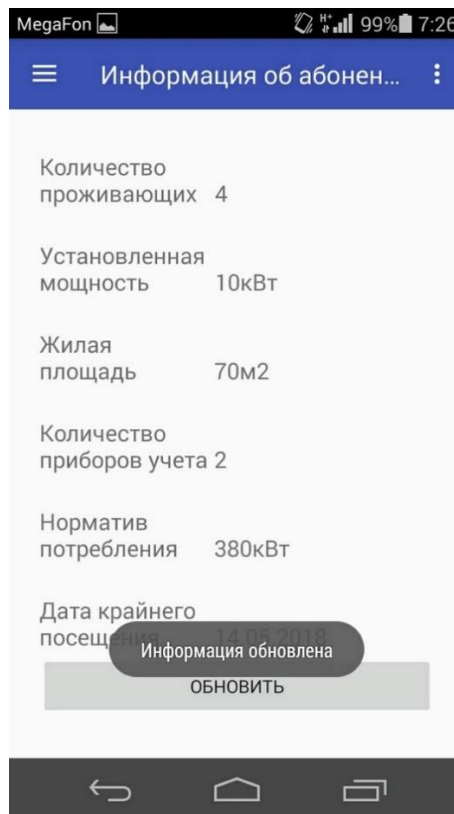


Рисунок 17 – Тестирование экрана «Информация об абоненте»

На приведенном выше рисунке можно увидеть уже заполненные данные об абоненте. Данные были получены при нажатии на кнопку «Обновить». На данном экране так же не наблюдается графических искажений.

Исходя из вышесказанного можно судить о корректности работы разработанного приложения. Исходный код модулей приложения представлен в приложении Б.

### 3.4 Минимальные технические требования

Для корректной работы приложения необходимо устройство, обладающее следующими характеристиками:

- экран диагональю 4.7” или больше;
- наличие модуля GPS;
- наличие модуля 3g;

- объем оперативной памяти 512 Мб и более;
- объем постоянной памяти 8 Гб и более;
- операционная система Android 4.2.\* и выше.

#### Вывод по третьему разделу

В данном разделе был описан процесс разработки требуемого ПО. Также были представлены структура конечного проекта, а также связь элементов между собой. Была описана логика работы каждого из основных экранов приложения, а также проиллюстрирован их внешний вид. В заключении было осуществлено тестирование созданного приложения, а также приведены минимальные технические требования для его работы.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была изучена предметная область энергоснабжения. Был изучен существующий метод снятия показаний счетчиков и их переноса в ИС энергосбытовой компании. Были выявлены существующие проблемы данного подхода и смоделировано ее решение. Были выполнены следующие задачи:

- изучена предметная область и уже существующие программные и аппаратные решения;
- спроектированы модели текущего и целевого состояния подхода к съему и передаче данных счетчиков;
- выбраны аппаратно-программная платформа, а также язык программирования, СУБД и среда программирования;
- описана структура разрабатываемого ПО;
- смоделированы алгоритмы работы разрабатываемого ПО;

В результате была достигнута основная цель работы: разработано ПО, предназначенное для внесения показаний контрольных измерений счетчиков водоснабжения и электроэнергии и их передачи в ИС.

Разработанное ПО позволяет:

- сократить временные издержки на передачу показаний в ИС;
- снизить риски потери информации за счет отказа от бумажных носителей;
- повысить надежность передачи данных в ИС;
- уменьшить возможность ошибки при считывании показаний;
- контролировать отклонения контролеров от маршрутов следования.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Должностная инструкция контролера абонентского отдела (энергосбыта) [Электронный ресурс] Режим доступа: <http://prom-nadzor.ru/content/dolzhnostnaya-instrukciya-kontrolera-abonentskogo-otdela-energobyta>
2. Анчарова, Т.В. Электроснабжение и электрооборудование зданий и сооружений: 2016. - 416 с.
3. Производственная инструкция для контролеров энергонадзора 1-й квалификационной группы [Электронный ресурс] Режим доступа: <http://leg.co.ua/instrukcii/energonadzor/proizvodstvennaya-instrukciya-dlya-kontrolerov-energonadzora-1-y-kvalifikacionnoy-gruppy.html>
4. Конюхова, Е.А. Электроснабжение объектов: Учебное пособие для среднего профессионального образования: ИЦ Академия, 2013. - 320 с.
5. Г.В. Коробов, В.В. Картавцев, Н.А. Черемисинова. Электроснабжение. Курсовое проектирование: Учебное пособие . - СПб.: Лань, 2011. - 192 с.
6. Кудрин, Б.И. Электроснабжение: Учебник для студентов учреждений высшего профессионального образования: ИЦ Академия, 2012. - 352 с.
7. Ю.Д. Сибикин, М.Ю. Сибикин. Электроснабжение - Вологда: Инфра-Инженерия, 2013. - 328 с.
8. А.И. Громов, А. Фляйшман, В. Шмидт. Управление бизнес-процессами: современные методы. монография Люберцы: Юрайт, 2016. - 367 с.
9. Елиферов, В.Г. Бизнес-процессы: Регламентация и управление: Учебник: НИЦ ИНФРА-М, 2013. - 319 с.
10. Ю.Ф. Тельнов, И.Г. Фёдоров. Инжиниринг предприятия и управление бизнес-процессами. Методология и технология: Учебное пособие: ЮНИТИ, 2015. - 176 с.

11. Чукарин, А.В. Бизнес-процессы и информационные технологии в управлении современной инфокоммуникационной компанией: Альпина Паблишер, 2016. - 512 с.
12. Лукин, В.Н. Введение в проектирование баз данных: Вузовская книга, 2015. - 144 с.
13. Мюллер, Р.Д. Проектирование баз данных и UML : Лори, 2013. - 420 с.
14. Шпак, Ю.А. Проектирование баз данных. Просто как дважды два: Эксмо, 2007. - 304 с.
15. С. Эмблер, П. Садаладж. Рефакторинг баз данных: эволюционное проектирование: Вильямс, 2007. - 672 с.
16. Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов: Питер, 2017, 688 с.
17. Дейтел П., Дейтел Х. Android для разработчиков: Питер 2016, 512 с.
18. Дон Гриффитс, Дэвид Гриффитс. Head First. Программирование для Android: Питер, 2016, 704 с.
19. Колисниченко Д.Н. Программирование для Android 5. Самоучитель: БХВ-Петербург, 2015, 303 с.
20. Пол Дейтел Харви Дейтел Эби Дейтел Майкл Моргано. Android для разработчиков: Питер, 2015, 384 с.
21. Голощاپов А.Л. Google Android. Создание приложений для смартфонов и планшетных ПК: БХВ-Петербург, 2014, 928 с.
22. Медникс З., Дорнин Л. Программирование под Android: Питер, 2013, 560 с.
23. Голощاپов А.Л. Google Android. Системные компоненты и сетевые коммуникации: БХВ-Петербург, 2012, 384 с.
24. Э. Бурнет. Привет, Android! Разработка мобильных приложений: Питер, 2012, 236 с.
25. Протокол ODATA [Электронный ресурс] Режим доступа: [https://ru.wikipedia.org/wiki/Open\\_Data\\_Protocol](https://ru.wikipedia.org/wiki/Open_Data_Protocol)

26. ODATA protocol [Электронный ресурс] – Режим доступа:  
<http://www.odata.org/>
27. Гонсалвес Д, Энтони К. Изучаем Java EE 7: Питер, 2016. - 640 с.
28. Шилдт, Герберт. Java 8. Руководство для начинающих: Вильямс, 2015. - 720 с.
29. Эккель, Брюс Философия Java : Питер, 2016. - 809 с.
30. Блох Д. Эффективное программирование: Питер, 2014 – 453 с.

## ПРИЛОЖЕНИЕ А

Расширенное представление иерархии разработанного приложения:

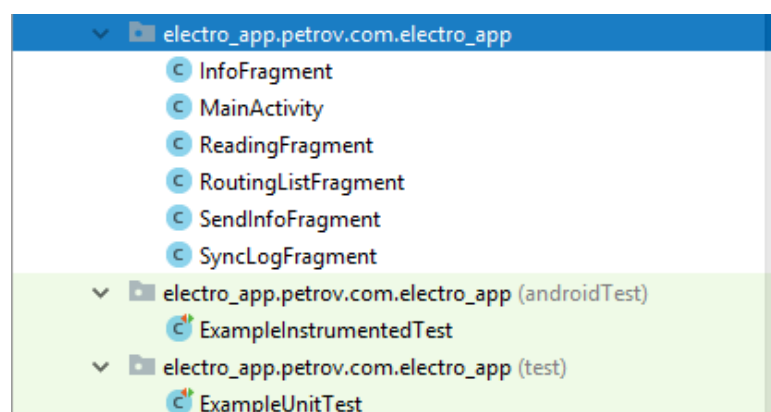


Рисунок 1А – Структура java классов

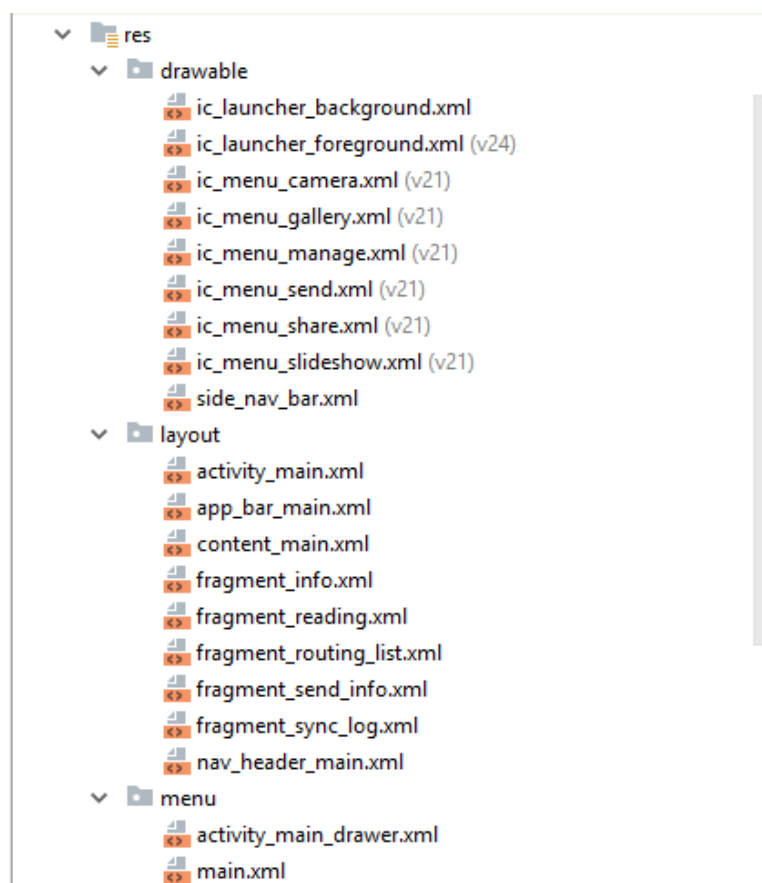


Рисунок А2 – Структура XML файлов

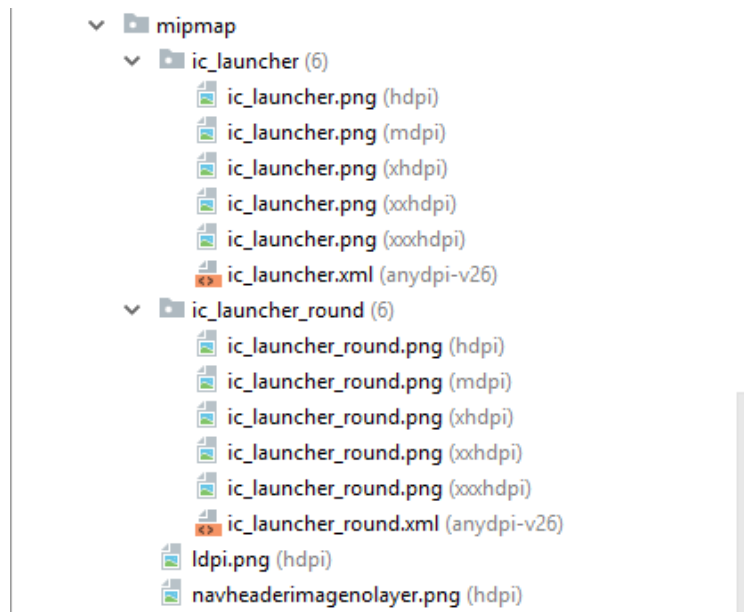


Рисунок 3А – Структура иконок приложения

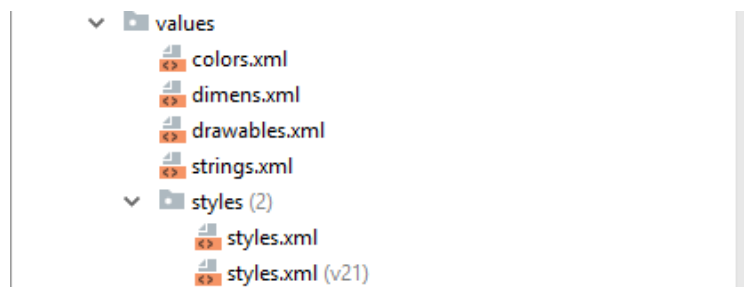


Рисунок 4А – Структура текстовых переменных

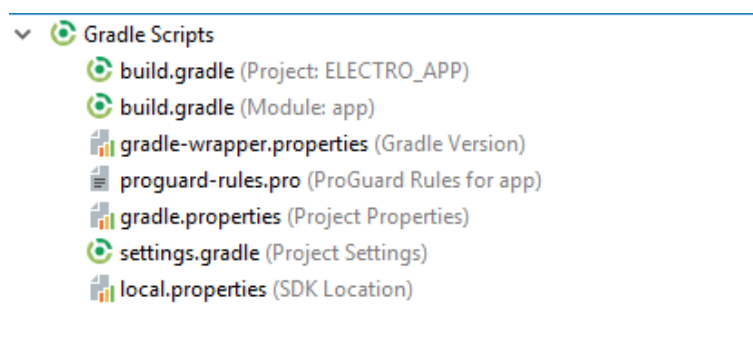


Рисунок 5А – Структура скриптов запуска

## ПРИЛОЖЕНИЕ Б

Исходный код java классов:

```
package electro_app.petrov.com.electro_app;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener,
    ReadingFragment.OnSelectedButtonListener, RoutingListFragment.AddressListListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            this, drawer, toolbar, R.string.navigation_drawer_open,
            R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();
```

```

        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        drawer.openDrawer(GravityCompat.START);
    }
    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
        } else {
            super.onBackPressed();
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        Fragment fragment = null;
        Class fragmentClass = null;
        int id = item.getItemId();

```

```

if (id == R.id.nav_routing) {
    fragmentClass = RoutingListFragment.class;
} else if (id == R.id.nav_reading) {
    fragmentClass = ReadingFragment.class;
} else if (id == R.id.nav_info) {
    fragmentClass = InfoFragment.class;
} else if (id == R.id.nav_journal) {
    fragmentClass = SyncLogFragment.class;
} else if (id == R.id.nav_send) {
    fragmentClass = SendInfoFragment.class;
}
try {
    fragment = (Fragment) fragmentClass.newInstance();
} catch (Exception e) {
    e.printStackTrace();
}
FragmentManager fragmentManager = getSupportFragmentManager();
fragmentManager.beginTransaction().replace(R.id.container, fragment).commit();
item.setChecked(true);
setTitle(item.getTitle());
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
return true;
}
@Override
public void onButtonSelected(int buttonIndex) {
}
@Override
public void itemClicked(long id) {
}
}

```

```

public class RoutingListFragment extends Fragment{

```



```

static interface AddressListListener {
    void itemClicked(long id);
};
private AddressListListener listener;
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";
private String mParam1;
private String mParam2;
public RoutingListFragment() {
}
public static RoutingListFragment newInstance(String param1, String param2) {
    RoutingListFragment fragment = new RoutingListFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    return inflater.inflate(R.layout.fragment_routing_list, container, false);
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    this.listener = (AddressListListener)activity;
    View rootView = getView();
    ListView mainView = (ListView) rootView.findViewById(R.id.ListView);
    mainView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View itemClicked, int position,
long id) {

```

```

        makeChoice();
    }
});
}
}

public class ReadingFragment extends Fragment implements View.OnClickListener {
    public ReadingFragment() {
    }
    public interface OnSelectedButtonListener {
        void onButtonSelected(int buttonIndex);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_reading, container, false);
        Button submitButton = (Button) rootView.findViewById(R.id.submitButton);

        submitButton.setOnClickListener(this);

        return rootView;
    }

    @Override
    public void onClick(View view) {
        makePhoto();
        getLocation();
    }
    public void makePhoto() {
        Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(takePictureIntent, 1);
    }
}
}

```

## Ведомость выпускной квалификационной работы

Обозначение	Наименование	Дополнительные сведения				
	Текстовые документы					
1. 110700026.09.03.02. 757.ПЗВКР	Пояснительная записка	с.				
	Графические документы					
2. 110700026.09.03.02. 757.ДМВКР	Демонстрационные материалы (презентация)	сл.				
	Демонстрационные материалы (пл. ф. А4)	__ лист.				
	Другие документы					
3.	Документы на компакт-диске	1 CD				
110700026.09.03.02. 757.ПЗВКР						
Изм.	Лист.	Номер докум.	Подп.	Дата	Разработка мобильного приложения для снятия показаний счетчиков водоснабжения и электроэнергии Ведомость ВКР	Лит.    Лист    Листов НИУ «БелГУ» гр. _____
Разработал						
Проверил						
Н.контр.	Нестерова Е.В.					
Утвердил	Ивашук О.А.					

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« \_\_\_ » \_\_\_\_\_ Г.

---

*(подпись)*

---

*(Ф.И.О.)*