

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем  
(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии  
(направленность (профиль) / специализация)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка мобильной версии сервиса компании «ООО Квартплата 24»

Обучающийся

В.С. Ильин

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н., В.С. Климов

(ученая степень, ученое звание, Инициалы Фамилия)

Консультант

И.Ю. Усатова

(ученая степень, ученое звание, Инициалы Фамилия)

Тольятти 2022

## Аннотация

Тема выпускной квалификационной работы — «Разработка мобильной версии сервиса компании «ООО Квартплата 24»».

Выпускная работа состоит из введения, трех глав, заключения, таблиц, списка литературы из 25 источников, включая 5 зарубежных источников и 6 приложений.

Ключевым вопросом дипломной работы является проектирование и создание мобильного приложения, имеющего функционал личного кабинета жителя компании «ООО Квартплата 24».

Целью выпускной квалификационной работы является разработка мобильного приложения личного кабинета жителя компании «ООО Квартплата 24».

Работа может быть разделена на следующие логически взаимосвязанные главы: анализ предметной области, проектирование мобильного приложения и реализация мобильного приложения. Каждая из глав включает в себя подглавы, в которых подробно рассматриваются вопросы об анализе компании и сферы ЖКХ, проектировании приложения в соответствии с выбранными технологиями и поставленными требованиями, и разработке приложения, учитывая используемые компанией технологии.

В конце исследования представляется работа об успешной разработке работающего мобильного приложения, которое позволяет пользоваться функционалом личного кабинета «ООО Квартплата 24».

Подводя итоги, хотелось бы подчеркнуть, что данная работа актуальна не только в решении проблемы отсутствия мобильной версии сервиса, а также подобные конструктивные и технологические решения могут быть применены для внедрения в других компаниях, нуждающихся в подобном приложении.

## **Abstract**

The title of the graduation work is «Development of a mobile version of the service of the company «Kvartplata 24 LLC»».

The senior paper consists of an introduction, three parts, a conclusion, a list of twenty five references including five foreign sources and six appendices.

The key issue of the thesis is design and creation of a mobile application that has the functionality of a personal account for a resident of the «Kvartplata 24 LLC» company.

The aim of work is the development of a mobile application for a personal account of a user of the company «Kvartplata 24 LLC».

The graduation work may be divided into several logically connected parts which are: domain analysis, mobile application design and mobile application implementation. Each of the chapters includes sub-chapters that deal in detail with the analysis of the company and the housing and communal services sector, designing the application in accordance with the selected technologies and requirements, and developing the application, taking into account the technologies used by the company.

Finally, we present the work on the successful development of a working mobile application that allows you to use the functionality of the personal account «Kvartplata 24 LLC».

In conclusion I would like to stress this work is relevant in solving the problem of the problems of the lack of a mobile version of the service specifically for this company, as well as similar design and technological solutions can be applied for implementation in other companies that need a similar application.

## Оглавление

Введение .....	5
Глава 1 Анализ предметной области .....	7
1.1 Общая характеристика ЖКХ .....	7
1.2 Общая характеристика компании «ООО Квартплата 24».....	9
1.3 Описание сервисов экосистемы .....	11
1.4 Постановка задачи на разработку мобильной версии сервиса .....	13
Глава 2 Проектирование мобильного приложения .....	14
2.1 Формирование требований для разрабатываемого приложения .....	14
2.2 Выбор технологий разработки мобильного приложения .....	15
2.3 Логическая модель разрабатываемого приложения.....	17
2.4 Разработка пользовательского интерфейса приложения .....	19
Глава 3 Реализация мобильного приложения.....	27
3.1 Спецификация выбранных технологий.....	27
3.2 Разработка алгоритма авторизации.....	29
3.3 Разработка пользовательских сценариев приложения.....	31
3.4 Тестирование приложения.....	36
Заключение .....	41
Список используемой литературы .....	43
Приложение А Диаграмма сервисов «ООО Квартплата 24» .....	47
Приложение Б_Диаграмма процесса регистрации показаний».....	48
Приложение В_Развитие рынка мобильных ОС .....	49
Приложение Г_Карта переходов и состояний приложения .....	50
Приложение Д_Разработанный прототип пользовательского интерфейса .....	51
Приложение Е_Позитивные тест-кейсы для тестирования .....	52

## Введение

В двадцать первом веке сфера жилищно-коммунального хозяйства стала всеохватывающей частью любого развитого государства, где, фактически, каждый человек является её потребителем. Как и для любой другой отрасли экономики, очень важно применять современные IT – технологии для достижения надлежащего оказания услуг.

Зачастую, услуги оплаты, внесения показаний и другие услуги, связанные с ЖКХ, доступны только на веб-сайтах, что является неудобным для пользователей мобильных устройств. Для удобства управления услугами ЖКХ сегодня не хватает некой единой системы, объединяющая главные аспекты ЖКХ.

IT-компания «ООО Квартплата 24» решает задачи расчета и учета платы за ЖКХ, а также прочие задачи по управлению услугами ЖКХ. Компанией была разработана большая экосистема облачных сервисов, которые тесно интегрированы между собой.

Целью выпускной квалификационной работы является разработка мобильного приложения личного кабинета жителя компании «ООО Квартплата 24».

Для достижения поставленной цели необходимо составить список задач:

- изучить предметную область;
- провести анализ предметной области;
- спроектировать мобильное приложение;
- реализовать мобильное приложение;
- провести тестирование приложения.

Объект исследования – экосистема сервисов компании «ООО Квартплата 24».

Предметом исследования является разработка мобильного приложения, включающее в себя функционал личного кабинета жителя компании «ООО

Квартплата 24».

Методы исследования:

- анализ;
- моделирование;
- проектирование;
- применение навыков программирования;
- изучение электронных источников.

В выпускной квалификационной работе разбираются вопросы по проектированию и разработке мобильной версии личного кабинета жителя.

В первой главе будет изучена и проанализирована деятельность сферы ЖКХ, описана деятельность компании «ООО Квартплата 24» и ее сервисов.

Во второй главе будут сформированы требования к мобильному приложению, построены логические модели приложения, а также будет разрабатываться пользовательский интерфейс мобильного приложения.

В третьей главе будет происходить разработка алгоритмов авторизации и пользовательских сценариев. Для этого также будут рассмотрены программный интерфейс компании и используемые технологии.

## Глава 1 Анализ предметной области

### 1.1 Общая характеристика ЖКХ

Для анализа предметной области, необходимо понять, что такое ЖКХ и на какие направления оно делится.

«Жилищно-коммунальное хозяйство – часть экономики, обеспечивающая функционирование зданий для безопасного, удобного и комфортабельного проживания и нахождения в них людей» [4, с. 4]. Частью ЖКХ являются также объекты социальной инфраструктуры для обслуживания жителей [4].

Для успешного функционирования современного населенного пункта деятельность ЖКХ – одна из основных и сложных задач городского управления. Всех их можно классифицировать по типу проводимой работы.

Основные классификации это:

- подача,
- отвод,
- поддержание.

Подача – это снабжение и обеспечение населения необходимыми ресурсами для его бытовых, а также производственных нужд. Примером является водоснабжение, которое предоставляет жителям возможность пользования пресной водой.

Под понятием отвода подразумевается сбор и вывоз бытовых отходов, а также отходов природного производства.

Поддержание – более сложная и разноплановая классификация деятельности ЖКХ. Её основа заключается в обеспечении граждан комфортного проживания. В данную классификацию можно внести такие задачи, как ремонт конструкций и прочих придомовых компонентов (домофон, дворные лавочки), так и обеспечение чистоты и порядка.

Исходя из вышеописанных классификаций, рассмотрим каждую из них

более подробно, описав состав этих классификаций.

В классификацию «Подача» включается:

- подача в объекты ЖКХ электроэнергии;
- подача в объекты ЖКХ холодной и горячей воды.

В классификацию «Отвод» включается:

- отвод жидких ЖБО (канализация);
- отвод твёрдых бытовых отходов;
- функционирование ливневой канализации.

В классификацию «Поддержание» включается:

- поддержание чистоты и порядка независимо от погодных и климатических условий (уборка территорий и дорог)
- поддержание в жилых и иных зданиях установленной температуры воздуха;
- устранение неполадок и недочетов жилых домов, а также социальной инфраструктуры;
- обеспечение жителей охраной и защитой.

Таким образом, после описания деятельности ЖКХ, представим её общий вид на рисунке 1.

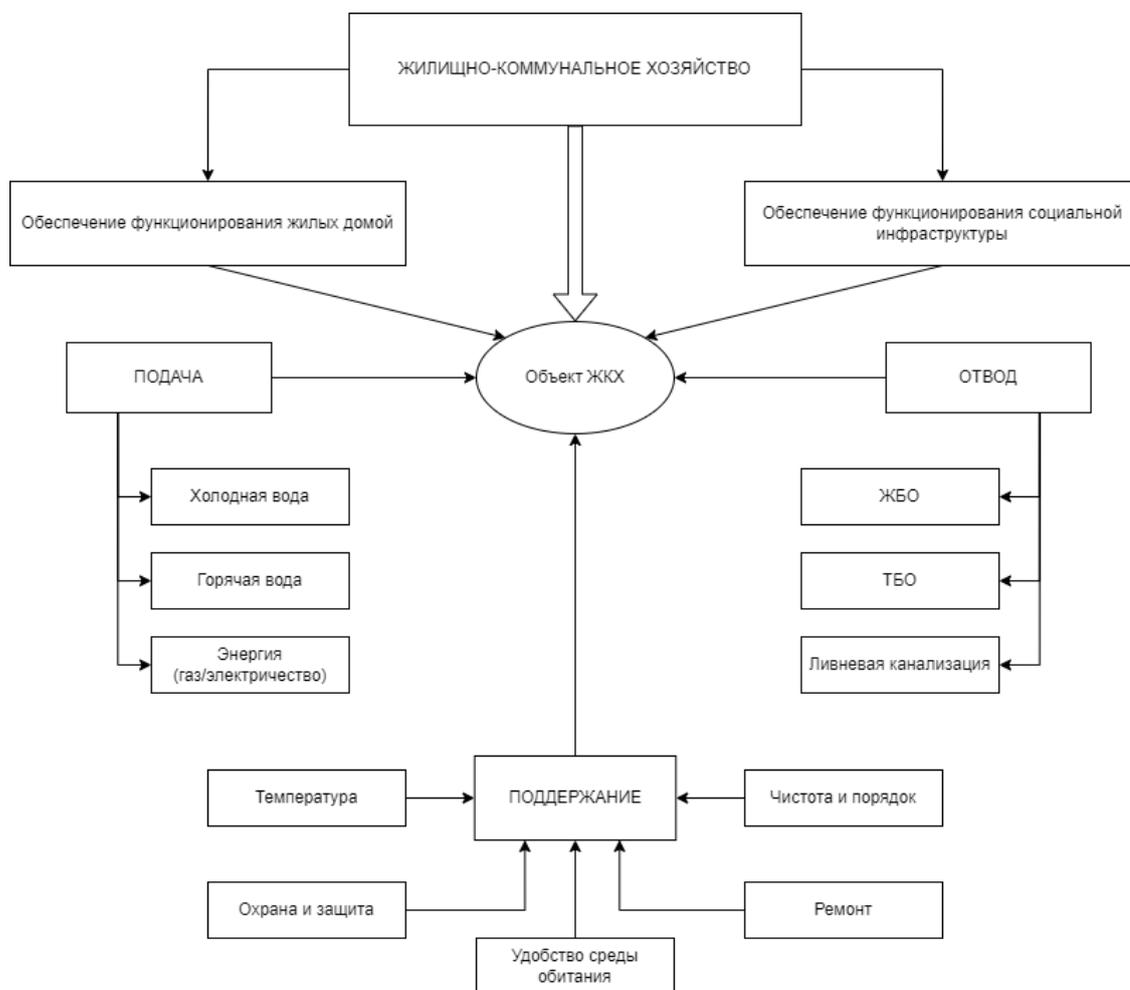


Рисунок 1– Схема деятельности ЖКХ

Решение всех вышеперечисленных задач возложено на ЖКХ и выделять из них основные сложно, т.к. все они в совокупности формируют качество жизни людей.

## 1.2 Общая характеристика компании «ООО Квартплата 24»

«ООО Квартплата 24» имеет большой опыт работы в области информационных технологий. Организация начала свою деятельность в 1996 году.

Форма организации компании – коммерческая, а значит ее деятельность, как и деятельность любой другой коммерческой организации, - получение

прибыли. В нашем случае получение прибыли идет за счет предоставления лицензии на программное обеспечение. ПО предоставляется по всей стране, а также за ее пределами.

На 2021 год компания имеет более 800 клиентов ТСЖ, УО, РСО в 63 субъектах РФ, а также за рубежом. Клиентура компании представлена в таблице 1.

Таблица 1 – Клиенты компании «ООО Квартплата 24»

Тип исполнителя ЖКУ	Количество клиентов	Примеры
ТСЖ (товарищество собственников жилья)	более 500	ТСЖ «Ленина, 74» (Нижегородская область), ТСЖ «Тетьково» (Тверская область), ТСЖ «Мечта» (Оренбургская область), ТСЖ «Промышленная 19» (Краснодарский край).
УО (управляющие организации)	более 200	ООО «УК №2 ЖКХ» (Самарская область), ООО «Новый Город» (Ивановская область), ООО «УК «Восточный» (Тюменская область), ООО «БашЖилСервис» (Республика Башкортостан).
РСО (ресурсоснабжающие организации)	более 20	Энергосбыт+ (ПАО «Т-плюс»), Российский коммунальные системы (РКС), ПАО Самараэнерго, ПАО Газпром, Фонд капитального ремонта.

Применение эффективных инструментов управления качеством работы экспертов, а также его многоуровневый контроль позволяют клиентам гарантировано получать поддержку вовремя и на должном уровне. Эксперты "ООО Квартплата 24" предоставляют клиентам в день обращения полную,

развернутую информацию о:

- последних изменениях законодательства;
- работе сервисов;
- проведении работ по передаче данных о жилом фонде и настройке системы;
- актуализации данных по плательщикам, проживающим и собственникам;
- работе личного кабинета жителя;
- проведении перерасчета;
- расчетах стоимости ЖКУ;
- действиях клиента при изменении тарифов, нормативов и правилах расчета ЖКУ.

### **1.3 Описание сервисов экосистемы**

Экосистема «ООО Квартплата 24» состоит более чем из шести тесно интегрированных сервисов, позволяющих жителю управлять услугами, которые предоставляет ТСЖ или УО, рассчитывать ЖКУ и прочее.

Сервисы экосистемы позволяют УО осуществлять обслуживание до 6 тысяч лицевых счетов, а РСО – до 12 тысяч лицевых счетов. Как показывает практика, другие сервисы, в том числе на базе 1С, не способны привести к подобным результатам.

В общем виде существующие сервисы можно представить на диаграмме, представленной на рисунке А.1:

Исходя из построенной диаграммы, можно заявить о 6 основных сервисах, которые являются базой работы компании, а именно:

- расчет ЖКУ (биллинговая система),
- процессинг платежей,
- личный кабинет жителя,

- сервис аналитики,
- сервис по работе с должниками,
- паспортный стол.

Из перечисленных сервисов можно особо выделить личный кабинет жителя, поскольку он объединяет в себя функции остальных, и именно им пользуется житель. В свою очередь, основная возможность личного кабинета – ввод показаний ПУ.

Попробуем разобрать процесс ввода показаний используя концептуальное моделирование. Из существующих технологий концептуального моделирования наиболее популярны следующие:

- SADT,
- UML,
- ARIS.

Из данных методологий стоит обратить внимание на ARIS.

ARIS представляет из себя современный подход к структурированному описанию деятельности компании, представлению её в виде взаимосвязанных графических диаграмм, удобных для понимания и анализа [5].

Методология ARIS включает в себя много различных нотаций, но самой популярной остается BPMN. Главная задача BPMN — это предоставление возможности описать и смоделировать бизнес-процессы таким образом, чтобы они были понятны аналитикам, программистам и обычным бизнес-пользователям, поэтому модели, созданные с помощью BPMN, могут использоваться в рамках выполнения анализа бизнес-процессов [5].

Проводить моделирование процессов с использованием нотации BPMN будем в среде разработки Microsoft Visio. Visio имеет готовые шаблоны BPMN, а также удобное управление элементами диаграммы.

Разработанная диаграмма изображена на рисунке Б.1.

Можно заметить, что единственное действие, требуемое от жителя – это корректный ввод показаний. Однако, отсутствие мобильного приложения для личного кабинета – большое неудобство для пользователей.

## **1.4 Постановка задачи на разработку мобильной версии сервиса**

Рассмотрев деятельность компании и экосистему сервисов, можно сделать вывод, что основной сервис, используемый жителем – личный кабинет, поскольку личный кабинет предоставляет функционал многих других сервисов.

Однако, наличие только веб-версии сервиса может сказаться на опыте пользования, поэтому возникает необходимость в создании мобильного приложения, имеющего основной функционал личного кабинета. Имея такой вариант сервиса, повысится удобство пользования жителем, что только положительно скажется на развитие экосистемы сервисов и компании в целом.

### **Выводы по главе 1**

Анализ предметной области показал, что были разработаны автоматизированные системы для более удобного управления жителем услугами ЖКХ. Такая система решает основные задачи расчета-учета ЖКХ, прием и распределение платежей и другие.

Также был рассмотрен основной связующий компонент компании «ООО Квартплата 24», а именно – сфера жилищно-коммунальных услуг. Были описаны задачи, решаемые этой части экономики и показана важность такой сферы в жизни общества.

В процессе изучения был выявлен наиболее важный для жителей сервис, а именно – личный кабинет жителя, который позволяет без обращения в ТСЖ, УО или РСО, а также без обращения в расчетный банк исполнителя ЖКУ оплачивать услуги ЖКХ.

Помимо этого, была поставлена задача на разработку мобильной версии личного кабинета.

## **Глава 2 Проектирование мобильного приложения**

### **2.1 Формирование требований для разрабатываемого приложения**

На основе полученных результатов первой главы образовалась необходимость в разработке мобильного приложения – личного кабинета жителя. Требования для приложения будут формироваться также, исходя из результатов первой главы.

Требования к разрабатываемым приложениям можно подразделить на две категории – функциональные и нефункциональные.

Функциональные требования — это требования к поведению системы, они определяют функциональность разрабатываемого программного обеспечения, т.е. описывают возможности, которые предоставляет система [14]. Функциональные требования определяют «как» необходимо реализовать проект, включают в себя бизнес-требования и пользовательские требования [14].

На основе результатов первой главы можно выявить следующие функциональные требования:

- мобильное приложение должно иметь возможность авторизации в личный кабинет жителя;
- мобильное приложение должно иметь возможность выхода из учетной записи пользователя;
- мобильное приложение должно отображать информацию личного кабинета жителя, а именно – лицевой счет, наименование управляющей компании, отображение адреса жилого помещения
- мобильное приложение должно иметь возможность массового ввода показаний приборов учета. По всем ПУ данные должны отправляться одним запросом;

Также, как и для любого другого приложения должны быть поставлены и нефункциональные требования.

Нефункциональные требования – это требования к характеру поведения системы, описывают как должна работать система, свойства и ограничения, накладываемые на разрабатываемое программное обеспечение [14]. Были выявлены следующие нефункциональные требования:

- мобильное приложение должно быть кроссплатформенным – под платформы iOS и Android;
- мобильное приложение должно содержать как элементы управления, так и графические элементы;
- мобильное приложение должно быть интуитивно понятным и удобным;
- мобильное приложение должно работать только в режиме портретной ориентации экрана;

## **2.2 Выбор технологий разработки мобильного приложения**

Разработка приложений для мобильных устройств на сегодняшний день феноменально развилась. Рынок мобильных приложений получил особый рост в 2009 году с пришествием смартфонов с тачскрином, а также с популярностью и распространенностью таких девайсов [16]. Сравнивая современную разработку мобильных приложения и разработку 2010-х годов, можно с уверенностью заявить, что условия и возможности разработки значительно улучшились [3].

Во-первых, определились лидеры мобильных операционных систем. Самые популярные мобильные операционные системы за период с января 2009 года по март 2022 года, а также рост рынка мобильных ОС отображены на рисунке В.1.

Основываясь на статистику Statcounter, к марту 2022 года актуальная доля ОС Android на рынке мобильных операционных систем составляет около 71%, iOS – около 27%. Остальные операционные системы, а именно –

Samsung, KaiOS, Windows и прочие в общей сложности составляют менее одного процента. Статистика с данными о долях мобильных операционных систем на мировом рынке представлена в таблице 2. Таким образом, «главные» мобильные операционные системы на сегодня – Android и iOS, и только они имеют актуальность разработки приложений.

Таблица 2 – Доля рынка мобильных ОС к марту 2022 года

Название мобильной ОС	Доля мирового рынка
Android	71.7%
iOS	27.57%
Samsung	0.42%
KaiOS	0.14%
Windows	0.01%
Остальные	0.1%

Кроме того, на фоне общего развития IT-технологий менялись и улучшались средства разработки. К примеру, появилась возможность добавления в платформу уже готовых компонентов, а значит ускорился процесс разработки.

Итак, как уже было сказано выше, средства мобильной разработки всегда дополнялись и улучшались. Это коснулось и языков программирования.

Поскольку невозможно выделить операционные системы на мобильных устройствах жителей, разработка будет вестись одновременно для Android и iOS.

Современные средства разработки мобильных приложений дают большой выбор инструментов. Один из самых популярных — фреймворк React Native.

React Native — очень востребованный среди разработчиков кроссплатформенный фреймворк. Это быстро растущая платформа начиналась как хакатон-проект в 2013 году. С использованием этой технологии были разрабатываются разного уровня – от социальных сетей до

управления функционалам электрокара. Одни из популярных приложений, разработанных на React Native:

- Bloomberg;
- Airbnb;
- Tesla;
- Microsoft OneDrive.

Благодаря единому воркфлоу запуск на разных операционных системах проходит, используя одни и те же инструменты. Это дает разрабатываемому приложению высокую скорость и простоту разработки. Поэтому, в качестве разработки приложения будем использовать фреймворк React Native.

### **2.3 Логическая модель разрабатываемого приложения**

В первую очередь стоит напомнить, что мобильное приложение это в первую очередь пользовательский интерфейс, поэтому проектирование лучше начать со схем экранов и последовательности переходов между ними. Это необходимо для того, чтобы определить набор шагов, которые предстоит пройти пользователю для получения желаемого результата, ведь бизнес-приложение создается для определенного набора ключевых сценариев (последовательности действий пользователя), с помощью которых человек может решить свои задачи [11].

Для того, чтобы разбить приложение на части, необходимо выделить экраны, которые связаны между собой, обычно они должны идти друг за другом в пользовательских сценариях [11]. Как правило, в приложениях с персональным профилем пользователя, можно отметить следующие экраны:

- экран авторизации
- экран главной страницы приложения,
- экран пользовательского сценария.

Экран авторизации – это страница, позволяющая осуществить вход в

профиль пользователя путем подтверждения прав на совершение определенных операций. Список возможных операций определяется поставщиком – сервером или разработчиком.

Экран главной страницы приложения отображает основную информацию приложения. Эта информация, в свою очередь зависит от самого приложения. К примеру, для приложений социальных сетей характерно частое взаимодействие с другими пользователями, а значит для такого приложения основной информацией может быть как лента новостей, так и страница чата.

Экран пользовательского сценария помещает в себя компоненты, которые позволяют выполнять набор определенных функций и задач. Примером такого экрана может служить страница написания электронного письма в почтовом сервисе. Функцией в данном случае будет являться написание текста и отправка его получателю.

Итак, базируясь на составленных требованиях и описанных частях(экранах) приложения, составим карту переходов и состояний, чтобы лучше понять основные пользовательские сценарии, а также обозначить связи между экранами [15]. Плюсами карты являются ее компактность и наглядность приборов.

Карта переходов начинается с точки старта – момента запуска приложения пользователем и таких точек старта может быть несколько, например, один вариант запуска для авторизованного пользователя, второй – для неавторизованного [15].

Разрабатывать такую карту будем, применяя диаграмму состояний State & Transition Diagram (сокращенно S&T). Такая техника визуализации ТЗ наглядно показывает, как некий объект переходит из одного состояния в другое [9]. Данная диаграмма изображена на рисунке Г.1.

На данной диаграмме красным цветом изображены причины возможных прерываний сценариев, к таким можно отнести отсутствие интернета. Рассматривая переход из состояния «1.2 Авторизация» в состояние «1.1 Главная страница» наличие доступа в интернет обязательно, поскольку без

него не будет возможности связи с сервером, а значит не будут предоставлены права на совершение операций.

Также помимо ситуации с отсутствием доступа в интернет отображена возможность некорректности введенных данных. Такие ситуации видны на переходах из состояний «1.2 Авторизация» в «1.1 Главная страница», а также из «2.1 Регистрация показаний» в «2.2 Результат отправки данных».

## **2.4 Разработка пользовательского интерфейса приложения**

Разрабатывая интерфейс пользователя, необходимо придерживаться некоторых принципов, определяющие взаимодействие с пользователем и саму структуру приложения в целом [6]. Для реализации каждого принципа должен учитываться ряд требований.

Рассмотрим основные принципы:

а) принцип интуитивно понятного и доступного интерфейса:

1) краткое описание: все элементы интерфейса должны быть понятны пользователю для того, чтобы он смог последовательно разобраться, как выполнить тот или иной запрос и получить необходимую информацию [12];

2) требования:

– интерфейс должен быть простым и лаконичным, то есть использовать оптимальное количество кнопок и предоставлять оптимальное количество вариантов выбора

– кнопки команд должны иметь интуитивно понятное графическое или текстовое обозначение. Для этого кнопкам с графическими обозначениями должна быть присуща наглядность, изображения должны быть понятны пользователю и не вызывать сомнений;

б) принцип структурности интерфейса:

1) краткое описание: определяет последовательность представления информации пользователю в удобной для него форме [12];

2) требования:

- структура интерфейса должна предусматривать последовательность и логичность операций. Это означает, что команды внутри приложения должны быть выстроены с учетом элементарной логики, чтобы пользователь смог последовательно выполнить необходимую операцию;
- элементы интерфейса должны стремиться к стандартной структуре размещения, что предполагает привычное расположение кнопок и прочих элементов. Это поможет пользователю лучше ориентироваться в возможностях приложения;

в) принцип привлекательности (эстетичности) интерфейса:

1) краткое описание: подразумевает, что пользователю приятнее иметь дело с интерфейсом, который гармонично воспринимается, то есть построен с учетом рационального использования законов психологии, композиции и восприятия изображения [12];

2) требования:

- цветовые сочетания элементов интерфейса должны иметь единый стиль оформления, не затруднять читаемость, соответствовать смысловому содержанию раздела приложения, при этом все детали должны быть видны;
- элементы интерфейса должны быть взаимосвязаны между

собой, выделяться только главные. Они должны иметь разные стили для кнопок с разными типами действия в зависимости от их назначения. Выделение элементов интерфейса может осуществляться с использованием цвета, шрифта, стиля, тени, размера, контура, заливки;

Среди современных средств разработки дизайна сегодня очень выделяется онлайн-сервис Figma. Сервис условно-бесплатный и дает большой и удобный функционал для проектирования внешнего вида приложений, в том числе и мобильных.

Разрабатывать интерфейс необходимо учитывая спроектированные части приложения, а именно, экраны. Важно понимать, что реализовывается именно прототип, поэтому конечный вид приложения может отличаться от задуманного [10].

В соответствии с запланированными экранами, используя сервис Figma составим прототип экрана авторизации в личный кабинет. Экран авторизации представлен на рисунке 2.

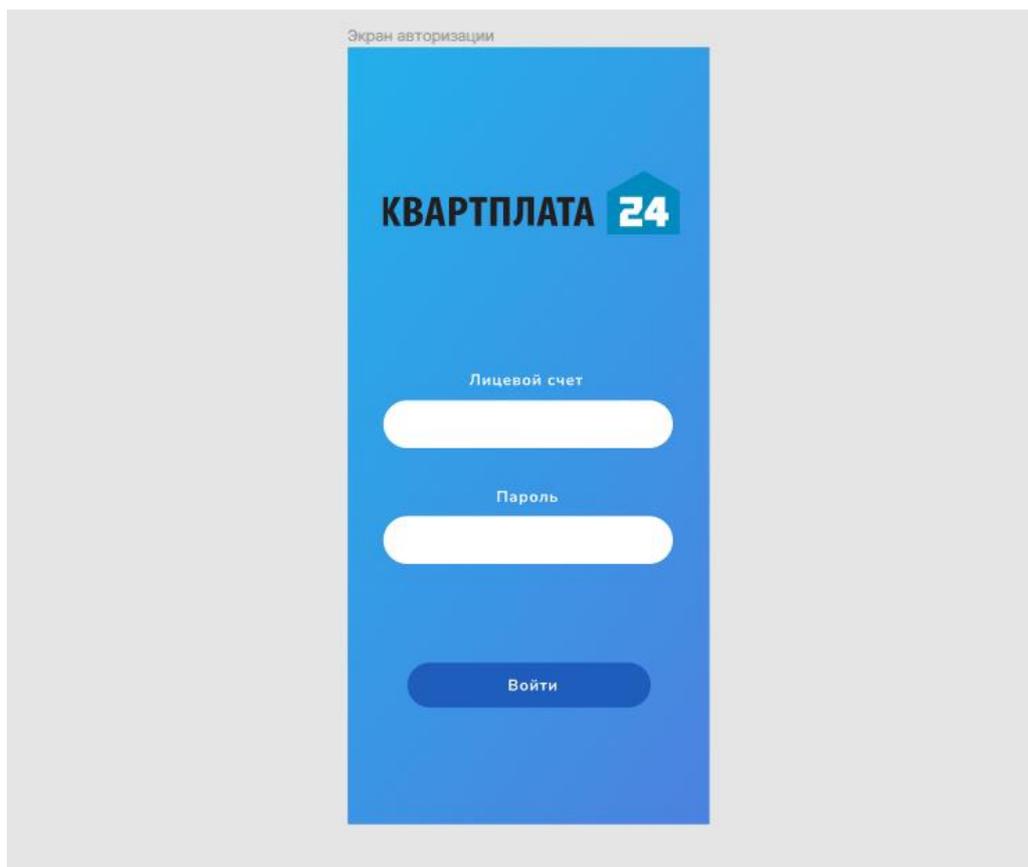


Рисунок 2 – Экран авторизации в личный кабинет

Экран содержит следующие компоненты:

- логотип компании «ООО Квартплата 24». Это помогает пользователю сориентироваться при входе в приложение;
- поля «Лицевой счет» и «Пароль». В них через клавиатуру вводятся данные для входа;
- кнопка «Войти». Обрабатывает введенные данные. Является основным ключом для перехода на следующий экран;

На карте переходов и состояний данный экран отображен как состояние «1.2 Авторизация».

Предполагаемый алгоритм действий клиента таков. Пользователь вводит данные для входа – лицевой счет и пароль. После нажатия на кнопку «Войти» данные отправляются на сервер и, если они корректны, происходит переход на экран главной страницы. Экран главной страницы отображен на рисунке 3.

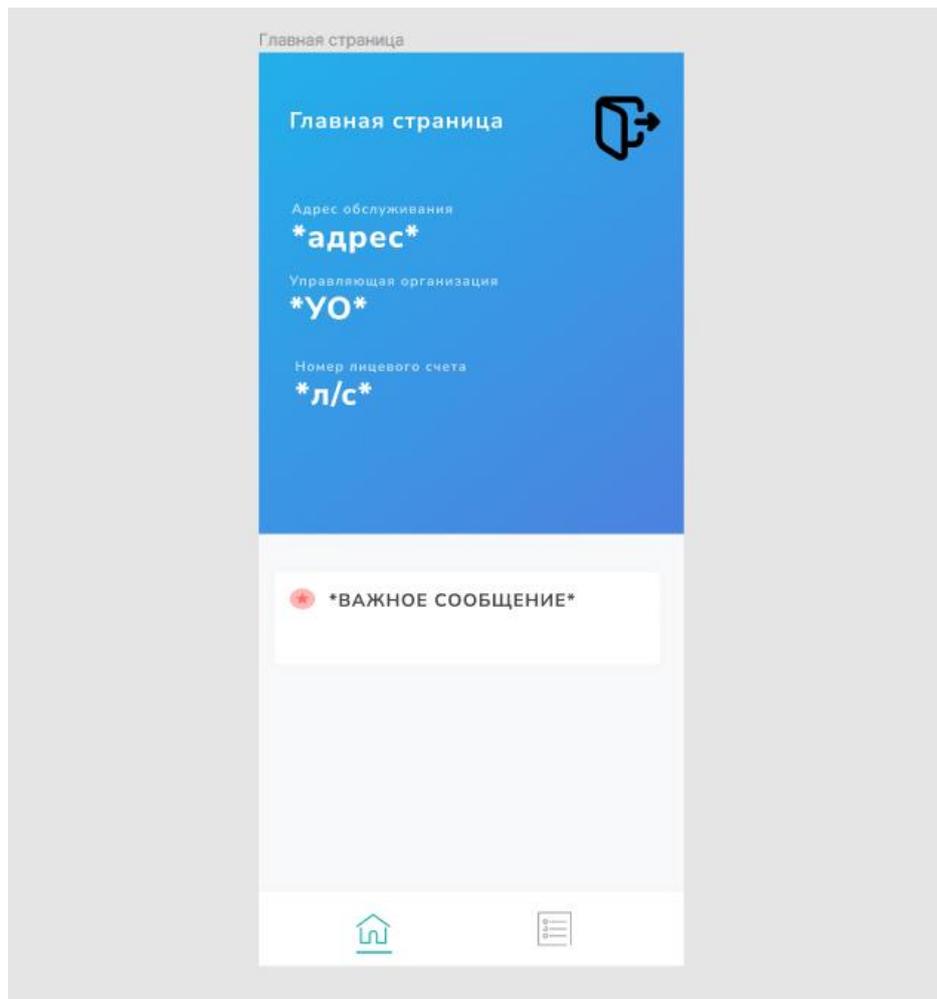


Рисунок 3 – Экран главной страницы приложения.

Данный экран приложения также можно отнести к группе пользовательских сценариев, поскольку здесь житель тоже взаимодействует с отображенными данными.

Экран содержит следующие компоненты:

- а) область с данными, которые необходимо отображать пользователю. В такие данные включены:
  - 1) адрес обслуживаемого дома,
  - 2) название управляющей организации,
  - 3) номер лицевого счета абонента;
- б) область с текстом, необходимым для отображения всем пользователям. Обычно, это сообщение о дате регистрации

показаний;

в) навигационная панель. Она позволяет переключаться между состояниями «1.1 Главная страница» и «2.1 Регистрация показаний»;

г) кнопка выхода из учетной записи.

На карте переходов и состояний данный экран обозначен как состояние «1.1 Главная страница».

При нажатии на значок документа в навигационной панели происходит переключение экранов. В нашем случае активным экраном станет экран регистрации показаний. Интерфейс данного экрана изображен на рисунке 4.

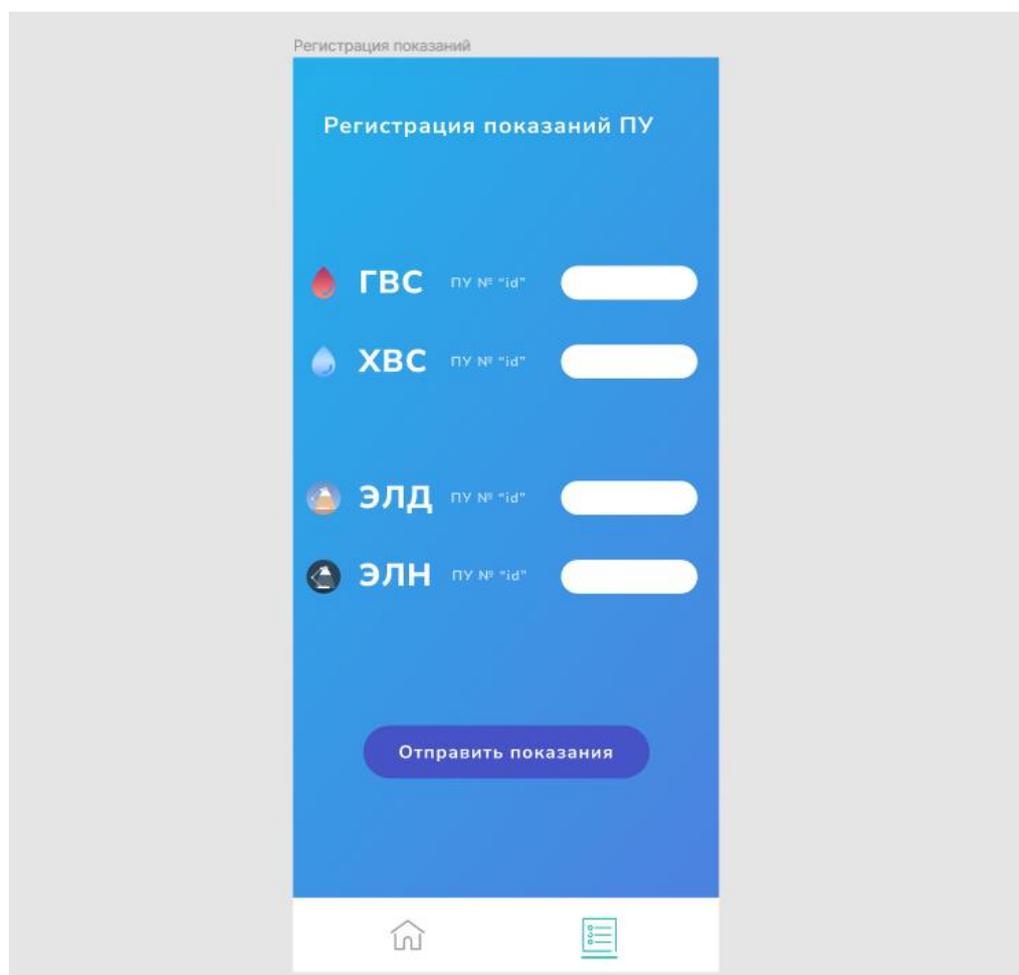


Рисунок 4 – Экран регистрации показаний ПУ.

Экран регистрации показаний ПУ является основным пользовательским сценарием, именно здесь осуществляется ввод и отправка показаний приборов

учета.

Экран содержит следующие компоненты:

- а) область ввода показаний. Она, в свою очередь, состоит из 4 строк, каждая из которых включает в себя:
  - 1) иконка, выбранная в соответствии с типом прибора учета,
  - 2) информация о типе прибора учета и его номер,
  - 3) поле для ввода данных;
- б) кнопка «Войти». Обрабатывает введенные данные. Является основным ключом для перехода на следующий экран;
- в) навигационная панель. Она позволяет переключаться между состояниями «1.1 Главная страница» и «2.1 Регистрация показаний».

Для наглядности изобразим разработанные экраны на одном изображении. Экраны представлены на рисунке Д.1.

Спроектированные экраны, составляющие пользовательский интерфейс соответствуют принципам понятности, структурности и привлекательности интерфейса. Они дают ориентир для процесса разработки, упрощая его и давая целостность будущему приложению [13].

## Выводы по главе 2

В ходе данной главы были рассмотрены основные требования, используемые для разработки современных мобильных приложений. Требования были составлены и разделены на две категории – функциональные и нефункциональные. Составленные требования дали первоначальное понимание того, как должно функционировать приложение, а также характер поведения системы.

Исходя из требования кроссплатформенности был проанализирован

рынок современных мобильных приложения и рынок мобильных операционных систем. Анализ выявил два лидера мобильных операционных систем – Android и iOS. Также был выбран основной инструмент для разработки приложения – React Native.

Кроме того, приложение было разбито на части — экраны, каждый из которых отвечает за свой пользовательский сценарий и на их основе с использованием диаграммы переходов и состояний была построена логическая модель разрабатываемого приложения.

Также в ходе данной главы был разработан пользовательский интерфейс, который, в свою очередь, соблюдает принципы взаимодействия пользователя с приложением.

## Глава 3 Реализация мобильного приложения

### 3.1 Спецификация выбранных технологий

Для лучшего понимания фреймворка React Native, необходимо разобрать важные аспекты, описывающие принцип работы таких приложений.

React Native – популярная платформа мобильных приложений на основе JavaScript. Платформа позволяет разрабатывать приложения для iOS и Android, используя одну и ту же кодовую базу по принципу наследования.

При запуске одного и того же приложения на двух разных ОС, для каждой ОС, комплектация и выполнение кода будет происходить по-разному [20]. Унаследовав общие свойства, «уникальные» компоненты, такие Text, View и другие будут выполняться в соответствии с архитектурой операционной системы.

React Native содержит некоторый JavaScript API над нативными компонентами. Это значит, что нативные компоненты имеют некоторый «биндинг» в JavaScript-компоненты на ReactJS, а взаимосвязь между нативным и JavaScript-бандлом осуществляется через bridge с помощью JavaScript API [25].

В общем виде принцип запуска React Native приложения изображен на рисунке 5.

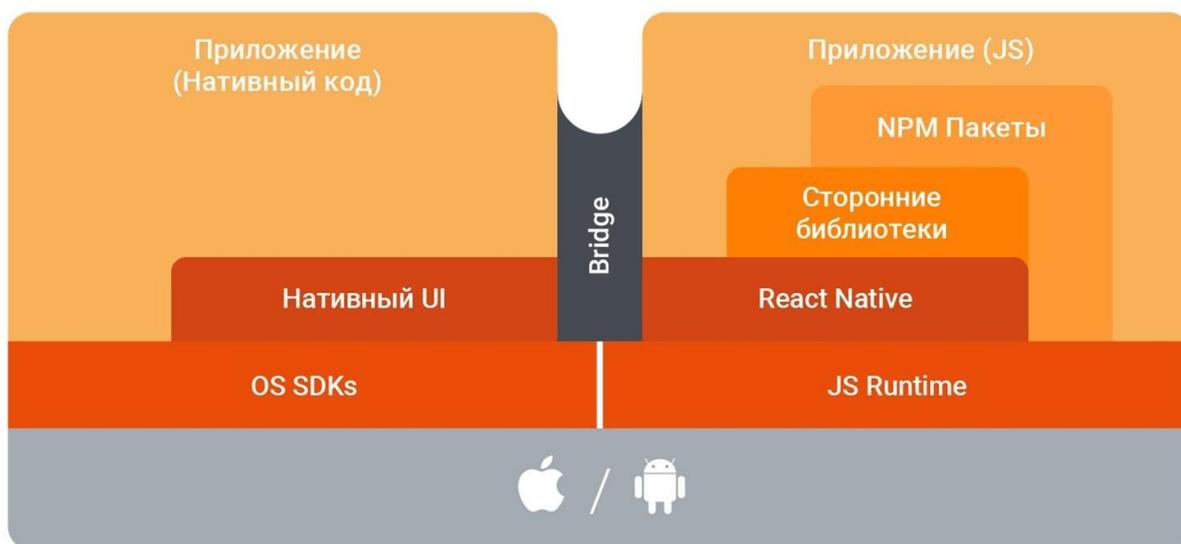


Рисунок 5 – Принцип работы RN приложения на iOS и Android

Существует два способа разработки на React Native — Expo и RN CLI.

Expo — некий слой абстракций, набор инструментов, библиотек и сервисов для быстрого запуска. В Expo присутствуют инструменты отладки и тестирования приложения.

Тестировать приложение можно прямо в браузере, создав локальный хост на ЭВМ, и даже на своем мобильном устройстве, подключенном к клиенту Expo Go. Кроме того, отсутствует потребность в наличии Android Studio для эмуляции мобильных устройств, а функция горячей загрузки проходит в разы быстрее чем в RN CLI.

Из минусов можно выделить неудобная документация, понимание которой у новичка может занять немало времени

Такой инструмент подходит для начинающих и средних специалистов в области разработки мобильных приложений благодаря ее простоте настройки, скорости и более легкому пониманию технологии [25].

React Native CLI — более сложная платформа для крупных и сложно-структурированных приложений [24]. Наличие Android Studio наряду с Android Studio SDK обязательно, что является не очень практичным решением [7]. К тому же отсутствует функция горячей загрузки, поэтому выберем Expo.

### 3.2 Разработка алгоритма авторизации

Компания «ООО Квартплата 24» для авторизации своих пользователей использует протокол OAuth2.

OAuth 2.0 — протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе [1]. Протокол избавляет от необходимости доверять приложению логин и пароль, а также позволяет выдавать ограниченный набор прав, а не все сразу

Рассмотрим алгоритм получения прав с использованием протокола OAuth2.

- клиент запрашивает авторизацию у владельца ресурса;
- клиент получает грант авторизации;
- клиент запрашивает токен доступа путем аутентификации с помощью сервера авторизации и предоставление гранта авторизации;
- сервер проверяет грант авторизации и если он действителен, выдает токен доступа (Access Token) и рефреш токен (Refresh Token);
- клиент запрашивает защищенный ресурс у провайдера и аутентифицируется, представляя токен доступа;
- провайдер проверяет токен доступа и если он действителен, обслуживает запрос.

Принцип работы такой авторизации изображен на рисунке 6.



Рисунок 6 – Авторизация через протокол OAuth

Можно заметить, что основные выходные данные – это Access и Refresh токены. Такие токены поставляются в JWT формате – открытом стандарте для создания токена. Access Token обычно используется при самой авторизации и имеет небольшой срок годности, а Refresh Token – токен для обновления пары Access и Refresh, который живет гораздо дольше [2]. Это обеспечивает безопасность в системе аутентификации, поскольку время действий злоумышленника уменьшается и с этим уменьшаются шансы получения несанкционированного доступа к данным.

Очень важно отслеживать состояние компонента авторизации, поскольку при выполнении пользовательских сценариев REST API запросы могут быть отклонены с ошибкой «401 Unauthorized», означающая, что пользователь не авторизован и не имеет прав на выполнение тех или иных действий [23].

Для отправки данных на сервер сначала необходимо составить

конфигурацию, состоящую из следующих компонентов:

- URL-адрес, на который отправляются HTTP запросы;
- `clientId`, уникальный идентификатор, используемый для идентификации. Является аналогом логина в паре «логин-пароль»;
- `clientSecret` – ключ доступа для получения пары токенов. Используется совместно с `clientId`. Является аналогом пароля в паре «логин-пароль»;

Из соображений информационной безопасности `clientId` и `clientSecret` не могут быть переданы третьим лицам, поэтому особое внимание должно быть сосредоточено на безопасности приложения.

### **3.3 Разработка пользовательских сценариев приложения**

Поскольку связь осуществляется через API, для совершения HTTP-запросов будет использоваться метод `Fetch`. Использование `Fetch API` в React-приложениях – это стандартный способ использования `Fetch API`, синтаксис не изменится, однако проблема заключается в выборе места расположения такого запроса [8]. Большинство запросов на выборку или HTTP-запросы любого рода обычно выполняются в `React Component`.

Сами данные, которые будут включать в себя информацию об управляющей организации, адрес обслуживания и номер лицевого счета хранятся в JSON формате- удобном текстовом формате обмена данных [17].

`RN Expo` предлагает размещать такие запросы в хуках. Хуки – это функции, с помощью которых можно «подцепиться к состоянию и методам жизненного цикла `React` из функциональных компонентов [21].

Пример алгоритма получения данных о пользователе из API приведен на рисунке 7.

```

const [data, setData] = useState([]);
console.log(data);

const getDataUsingGet = () => {
  //GET запрос
  fetch('https://lk.kvp24.ru/v1/profile', {
    method: 'GET',
  })
  .then((response) => response.json())
  //Если ответ в json формате
  .then((json) => setData(json)) //Сохранить данные

  //Если ответ не в json формате, уведомление об ошибке
  .catch((error) => {
    //Ошибка
    alert(JSON.stringify(error));
    console.error(error);
  })
  .finally(()=>setLoading(false));
};

```

Рисунок 7 – Получение данных о пользователе из REST API с применением Fetch API

Главным параметром для Fetch метода является URL-адрес, который задает владелец сервера ресурсов. Такой подход упрощает способ получения данных в JSON-формате [19]. Данные сохраняются в переменную data и отслеживаются состоянием Loading. Если состояние Loading равно True, скорее всего, происходит ошибка в получении данных.

После удачного сохранения остается перенести это на страницу приложения. Для выборки и поиска данных в Expo присутствует функция renderItem. Пример выборки данных с использованием renderItem приведен на рисунке 8.

```

<View style={{marginTop:10}} profileInfo={data.linkedAccounts} executor={data.overhaulContract}>
  <Text style={styles.infoListTop}>
    | | Адрес обслуживания
  </Text>
  <View renderItem={({ profileInfo }) => (
    <Text style={styles.infoListObject} >{profileInfo.address}</Text>
  )}
  />
  <Text style={styles.infoListTop}>
    | | Управляющая организация
  </Text>

  <View keyExtractor={({ overhaulContract }, id) => id renderItem={({ executor }) => (
    <Text style={styles.infoListObject} >{executor.managerialCompanyName}</Text>
  )}
  />

  <Text style={styles.infoListTop}>
    | | Номер лицевого счета
  </Text>
  <View renderItem={({ profileInfo }) => (
    <Text style={styles.infoListObject} >{profileInfo.contractNumber}</Text>
  )}
  />
</View>

```

Рисунок 8 – Выборка данных с использованием renderItem

Также на данной странице необходимо разработать кнопку для выхода из учетной записи жителя. Обработка нажатия на кнопку в Expo проходит аналогично JavaScript, за нажатие отвечает параметр onPress [22]. Он вызывает функцию doUserLogout, которая в свою очередь вызывает аннулирование токена доступа [18]. После успешного Revoke Token состояние приложение переносится на страницу авторизации.

Пример программного кода с выходом из учетной записи представлен на рисунке 9.

```

const doUserLogout = async function () {
  return await Parse.User.logout()
    .then(async () => {
      const currentUser = await AppAuth.revokeAsync(config, options());
      if (currentUser === null) {
        Alert.alert('Успешно!', 'Выход из учетной записи!');
      }
      // Ошибка вернет пользователя на экран
      navigation.dispatch(StackActions.popToTop());
      return true;
    })
    .catch((error) => {
      Alert.alert('Ошибка!', error.message);
      return false;
    });
};

```

Рисунок 9 – Аннулирование токена доступа

Отправка данных происходит аналогично получению, с использованием POST запросов. Однако, проверка корректности введенных данных происходит на сервере владельца ресурсов. При получении положительного ответа появляется уведомление об успешной регистрации показаний. Кроме того, такой запрос отправляется не асинхронно, в отличии от GET запроса, поскольку она не отслеживает состояние авторизации.

Экран регистрации показаний представлен на рисунке 10.

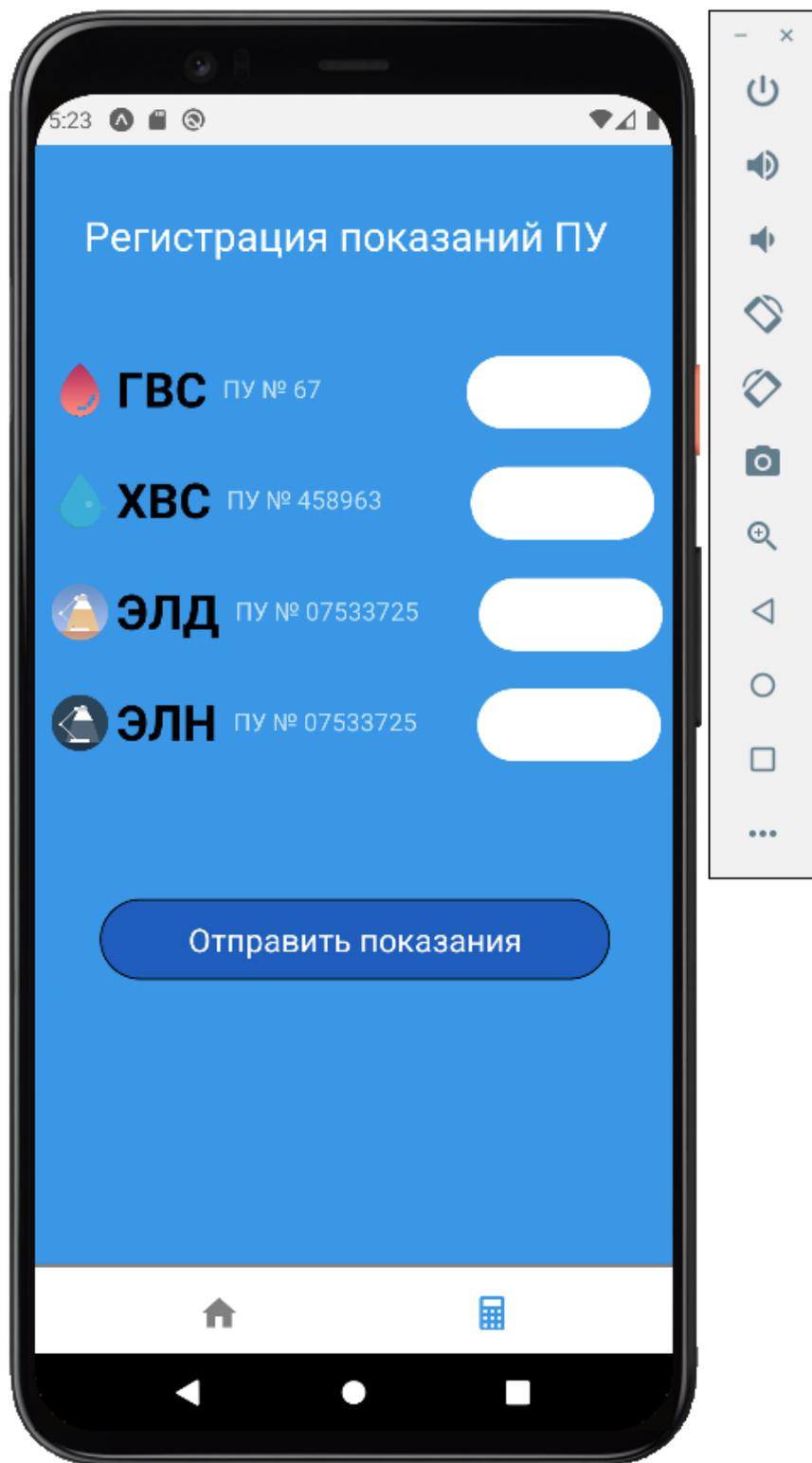


Рисунок 10 – Экран регистрации показаний

### 3.4 Тестирование приложения

Исходя из требований, составленных во 2 главе, приложения должно работать на платформах Android и iOS. Все дальнейшие тесты будут проводиться с использованием виртуального эмулятора Android устройства, а также реальным физическим iOS устройством – Apple Iphone.

Кроссплатформенность приложения можно подтвердить после успешного проведения ряда тест-кейсов, касаемых функциональных требований.

Были составлены позитивные и негативные тест-кейсы, которые, в зависимости от входных данных должны правильно выполнять сценарий. Примером позитивного тест-кейса являются корректные данные для авторизации, в то время как примером негативного тест-кейса будут некорректные данные. Список позитивных тест-кейсов приведен в таблице Е.1.

Все позитивные тест-кейсы проведены успешно, данные получаются и отправляются успешно, навигация приложения также имеет правильную работоспособность. Правильность работы отправки показаний через приложение на платформе Android изображена на рисунке 11.

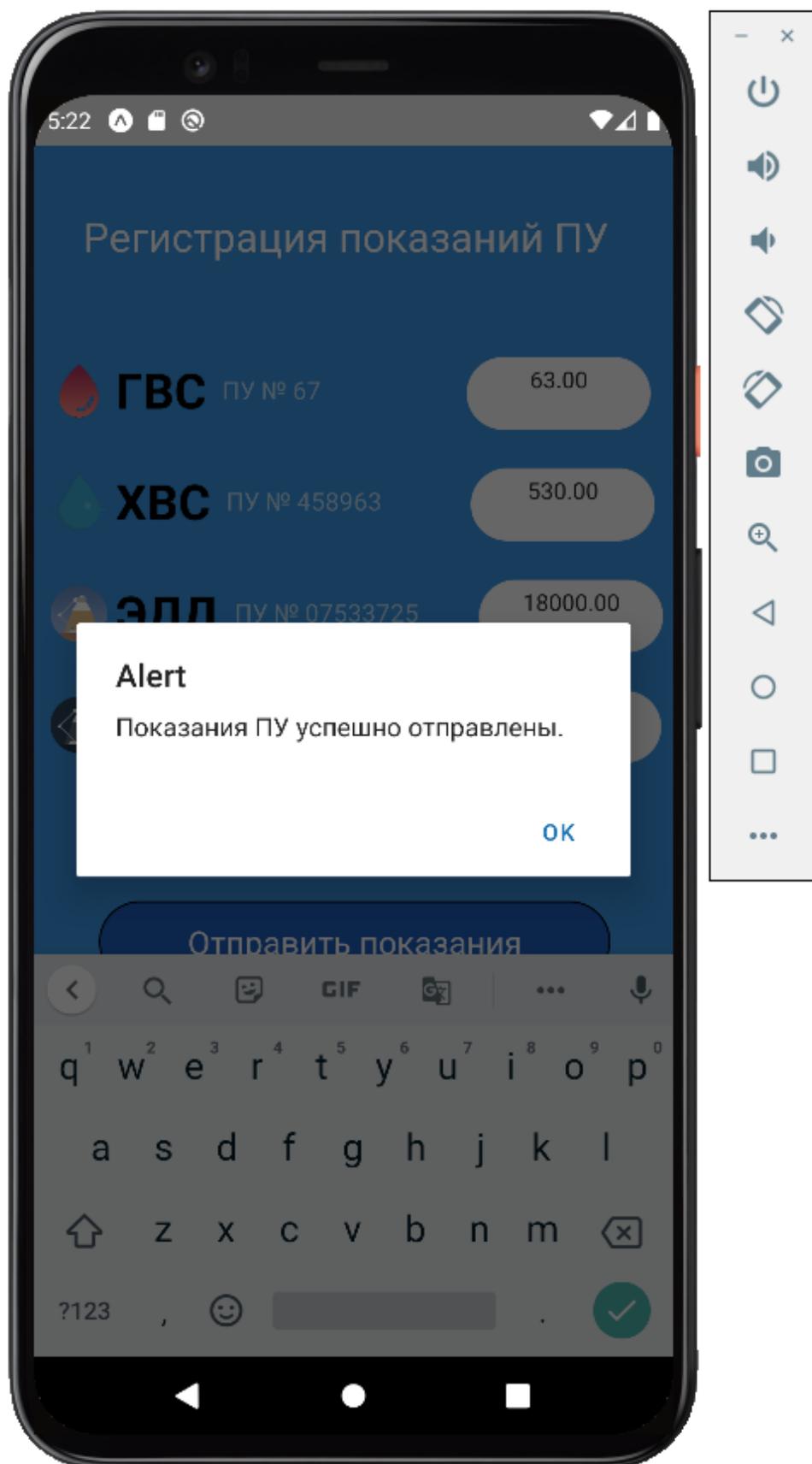


Рисунок 11 – Результат позитивного тест-кейса с корректными данными

Также были составлены некоторые негативные тест-кейсы, при которых приложение должно правильно обработать некорректность данных и показать об этом уведомление. Список позитивных тест-кейсов приведен в таблице 3.

Таблица 3 – Негативные тест-кейсы для тестирования

Название	Инструкции	Ожидаемый результат
Некорректность данных авторизации	Открыть приложение от лица не аутентифицированного пользователя, ввести неверные данные	Уведомление о некорректности введенных данных
Некорректность отправляемых данных	Открыть экран регистрации показаний от лица аутентифицированного пользователя, ввести некорректные данные 4 приборов учета, нажать кнопку «отправить показания»	Уведомление о неверности показаний

Правильность проверки введенных данных о показаниях ПУ представлена на рисунке 12.

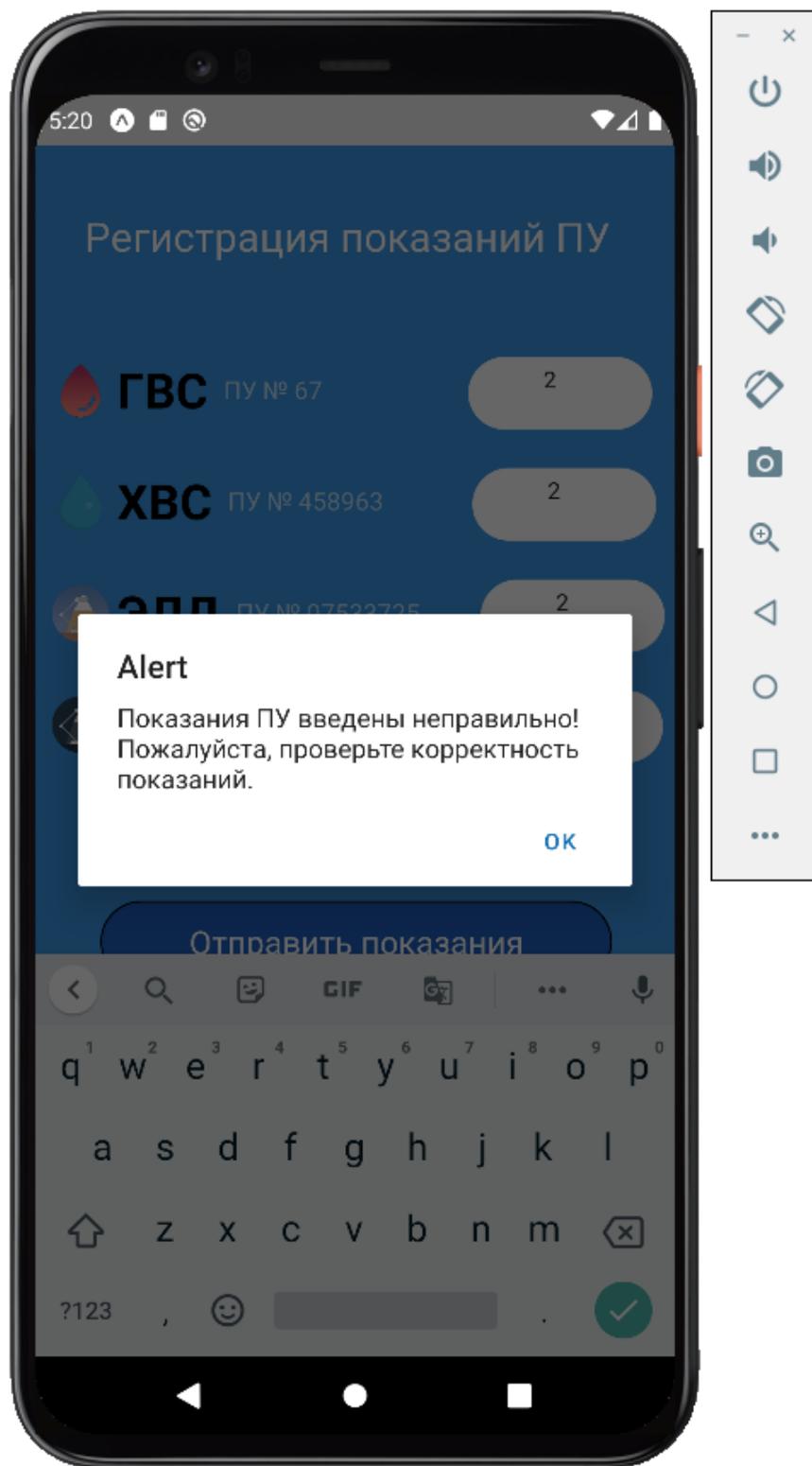


Рисунок 12 – Результат негативного тест-кейсы с некорректными данными

Негативные тест-кейсы также были выполнены успешно, данные обрабатываются и проверяются успешно.

Описанные выше тест-кейсы показывают правильность работы

приложения, оно удовлетворяет всем составленным функциональным и нефункциональным требованиям. Кроме того, все тест-кейсы были проведены в обеих операционных системах, поэтому можно сделать вывод, что кроссплатформенность присутствует, и приложение одинаково работает на ОС Android и iOS.

Для наглядности пользовательский интерфейс представлен в приложении.

### Выводы по главе 3

В результате данной главы были описаны выбранные средства разработки, изучены способы авторизации через протокол OAuth2. Также было разработано мобильное приложение с использованием технологии React Native Expo.

Тестирование показало, что приложение удовлетворяет всем поставленным требованиям.

## Заключение

В процессе выполнения выпускной квалификационной работы был проведен анализ сферы жилищно-коммунальных услуг, а также компании «ООО Квартплата 24». Анализ выявил проблему в отсутствии мобильной версии основного сервиса – личного кабинета жителя. Была выделена актуальность исследуемой темы, определены объект, предмет исследования, цели и задачи работы.

При изучении деятельности компании «ООО Квартплата 24» была разработана логическая модель основных бизнес-процессов, а также составлена диаграмма сервисов компании «ООО Квартплата 24», отображающие основу их работы.

Были рассмотрены стандарты требований для современных мобильных приложений и на их основе были составлены требования к разрабатываемому приложению. Требования были подробно разобраны и были описаны принципы разработки компонентов приложения, дающие приложению современный вид.

Анализ мобильного рынка и мобильных приложений показал, что приложение лучше делать кроссплатформенным и поэтому были рассмотрены основные средства для реализации таких приложений. Исходя из составленных требований были выбраны технологии, применяемые нами для разработки приложения.

Основываясь на требованиях и учитывая особенности выбранных технологий, был разработан прототип пользовательского интерфейса и базируясь на нем была реализована внешняя оболочка приложения.

Была проведена разработка мобильного приложения, в которую входило:

- описание спецификаций выбранных технологий,
- сравнение узконаправленных технологий для разработки,
- изучение программного интерфейса API компании

- написание программного кода,
- тестирование.

Результатом выпускной квалификационной работы является работающее мобильное приложение для личного кабинета жителя компании которое позволяет:

- управлять авторизацией в систему,
- собирать и отображать необходимые данные,
- совершать регистрацию показаний ПУ,

После проведения тестирования можно сделать вывод, что разработанное мобильное приложение работает корректно, выполняет основные функции личного кабинета и соответствует всем установленными требованиям.

## Список используемой литературы

1. Базовое понимание OAuth 2.0 [Электронный ресурс] // Stfalcon. – Режим доступа: <https://stfalcon.com/ru/blog/post/oauth-2.0>
2. Вишневский Д.Н. OAUTH И ЕГО ИСПОЛЬЗОВАНИЕ В СОВРЕМЕННЫХ СИСТЕМАХ // Вестник магистратуры. 2019. №6-2 (93). URL: <https://cyberleninka.ru/article/n/oauth-i-ego-ispolzovanie-v-sovremennyh-sistemah> (дата обращения: 16.02.2022).
3. Воробьев Н.А., Бурмин Л.Н., Степанов Ю.А. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕДСТВ ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ // Евразийский Союз Ученых. 2020. №6-1 (75). URL: <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-sredstv-testirovaniya-mobilnyh-prilozheniy> (дата обращения: 15.02.2022).
4. Все о ЖКХ на 2019 год: услуги, тарифы, платежи и сборы, основания не платить или платить меньше / сост. Е. Давыденко. – Москва: Издательство АСТ, 2018. – 224 с. (Справочник для населения)
5. Долганова, О. И. Моделирование бизнес-процессов : учебник и практикум для академического бакалавриата / О. И. Долганова, Е. В. Виноградова, А. М. Лобанова ; под редакцией О. И. Долгановой. — Москва : Издательство Юрайт, 2019. — 289 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-00866-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/433143> (дата обращения: 20.03.2022).
6. Нужный, А. М. Разработка мобильных приложений : учебное пособие для СПО / А. М. Нужный, Н. И. Гребенникова, В. В. Сафронов. — Саратов : Профобразование, 2022. — 92 с. — ISBN 978-5-4488-1494-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/121301.html> (дата обращения: 19.03.2022). — Режим доступа: <https://www.iprbookshop.ru/epd-reader?publicationId=121301>
7. Пирская, Л. В. Разработка мобильных приложений в среде Android

Studio : учебное пособие / Л. В. Пирская. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2019. — 123 с. — ISBN 978-5-9275-3346-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/100196.html> (дата обращения: 23.02.2022). — Режим доступа: <https://www.iprbookshop.ru/epd-reader?publicationId=100196>

8. Применение в React REST API с помощью Fetch и Axios [Электронный ресурс] // Initskill. - Режим доступа: <https://www.internet-technologies.ru/articles/primenenie-rest-api-s-pom-fetch-i-axios.html>

9. Простое руководство по UML-диаграммам и моделированию баз данных [Электронный ресурс]: Сайт с преимуществами и типами UML-диаграмм. URL: <https://www.microsoft.com/ru-ru/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling> (Дата обращения: 02.01.2021)

10. Соколова, В. В. Разработка мобильных приложений : учебное пособие / В. В. Соколова. — Томск : Томский политехнический университет, 2014. — 176 с. — ISBN 978-5-4387-0369-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/34706.html> (дата обращения: 06.06.2022). — Режим доступа: <https://www.iprbookshop.ru/epd-reader?publicationId=34706>

11. Техническое проектирование мобильных приложений [Электронный ресурс] // Gitbooks. - Режим доступа: <https://slavachernikoff.gitbooks.io/technical-design-of-mobile-applications/content/chapter2.html>

12. Утробина Елена Степановна, Кокорина Ирина Петровна Принципы разработки пользовательского интерфейса для мобильных картографических приложений // Интерэкспо Гео-Сибирь. 2019. №2. URL: <https://cyberleninka.ru/article/n/printsipy-razrabotki-polzovatel'skogo-interfeysa-dlya-mobilnyh-kartograficheskikh-prilozheniy> (дата обращения: 20.03.2022).

13. Федотенко, М. А. Разработка мобильных приложений. Первые

шаги / М. А. Федотенко ; под редакцией В. В. Тарапаты. — Москва : Лаборатория знаний, 2019. — 336 с. — ISBN 978-5-00101-640-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89067.html> (дата обращения: 03.04.2022). — Режим доступа: <https://www.iprbookshop.ru/89067.html>

14. Формирование требований и классификация требований [Электронный ресурс] // Бизнес-анализ в России - Режим доступа: <https://analytics.infozone.pro/formation-requirements-and-classification-requirements/>

15. Черников В.Н., Подвальный С.Л., Барабанов В.Ф., Нужный А.М. ФОРМАЛИЗАЦИЯ ПРОЦЕССА РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО КРОССПЛАТФОРМЕННОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ // Вестник ВГТУ. 2018. №5. URL: <https://cyberleninka.ru/article/n/formalizatsiya-protsessa-razrabotki-polzovatelskogo-krossplatformennogo-mobilnogo-prilozheniya> (дата обращения: 06.04.2022).

16. Чубенко М.Г., Смирнова Ю.А. АНАЛИЗ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ И МОБИЛЬНОГО РЫНКА // Вестник науки. 2020. №6 (27). URL: <https://cyberleninka.ru/article/n/analiz-mobilnyh-prilozheniy-i-mobilnogo-rynka> (дата обращения: 28.01.2022).

17. Шор Александр Михайлович Сравнительный анализ подходов в разработке api веб-приложений // StudNet. 2020. №9. URL: <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-podhodov-v-razrabotke-api-veb-prilozheniy> (дата обращения: 16.01.2022).

18. Эспозито, Д. Разработка современных веб-приложений: анализ предметных областей и технологий / Д. Эспозито. - М.: Вильямс И.Д., 2017. - 464 с.

19. Янишевская А.Г., Чурсин М.А. Использование сторонних интерфейсов программирования приложений на примере интерфейсов прикладного программирования социальных сетей Facebook и twitter // ИВД. 2015. №2-2. URL: <https://cyberleninka.ru/article/n/ispolzovanie-storonnih->

interfeysov-programmirovaniya-prilozheniy-na-primere-interfeysov-prikladnogo-programmirovaniya-sotsialnyh (дата обращения: 15.02.2022).

20. Android Studio [Электронный ресурс]: Android Studio Documentation. URL: <https://developer.android.com/docs> (Дата обращения: 20.02.2022)

21. Expo [Электронный ресурс]: Expo Documentation. URL: <https://docs.expo.dev/> (Дата обращения: 02.04.2021)

22. JavaScript [Электронный ресурс]: Android Studio Documentation. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (Дата обращения: 20.02.2022)

23. Node.js [Электронный ресурс]: Node.js Documentation. URL: <https://nodejs.org/ru/> (Дата обращения: 20.02.2022)

24. React Native [Электронный ресурс]: React Native Documentation. URL: <https://reactnative.dev/> (Дата обращения: 20.01.2021)

25. React Native для самых маленьких. Опыт мобильной разработки [Электронный ресурс] // Хабр. - Режим доступа: <https://habr.com/ru/company/oleg-bunin/blog/499636/>

## Приложение А

### Диаграмма сервисов «ООО Квартплата 24»

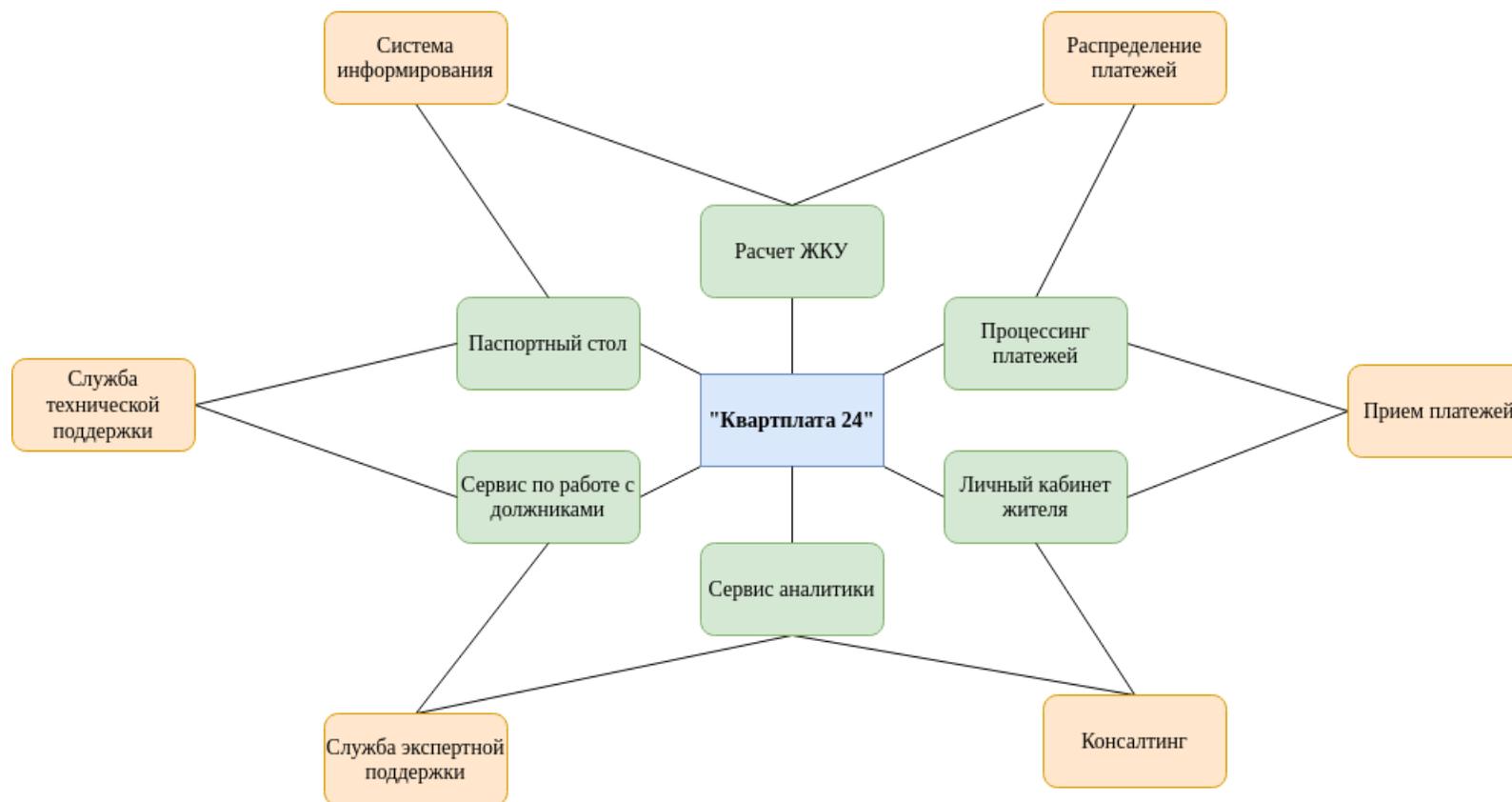


Рисунок А.1 – Диаграмма сервисов «ООО Квартплата 24»

## Приложение Б

### «**Диаграмма процесса регистрации показаний**»

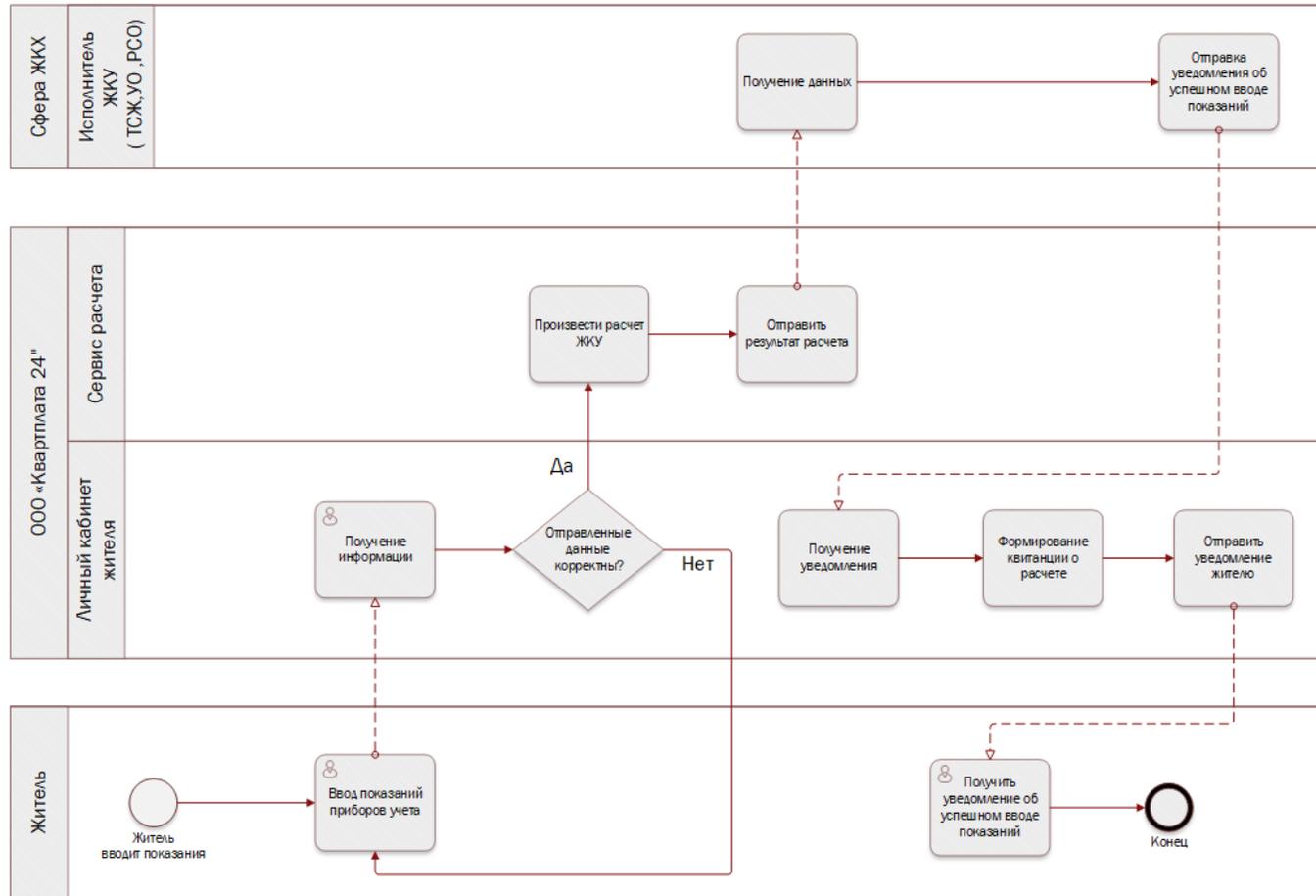


Рисунок Б.1 – Диаграмма процесса регистрации показаний

## Приложение В

### Развитие рынка мобильных ОС

StatCounter Global Stats

Mobile Operating System Market Share Worldwide from Jan 2010 - Mar 2022

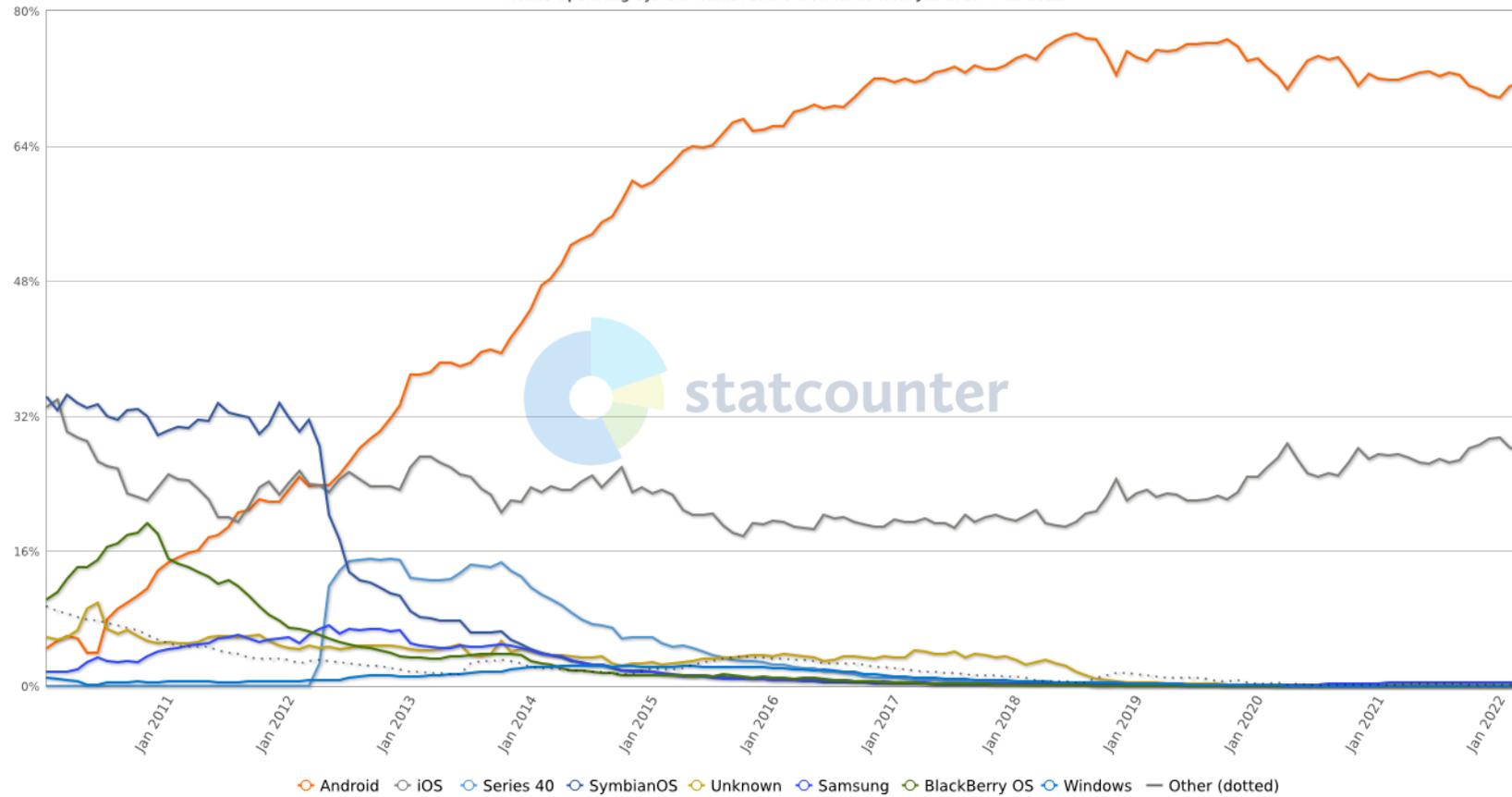


Рисунок В.1 – Развитие рынка мобильных ОС

## Приложение Г

### Карта переходов и состояний приложения

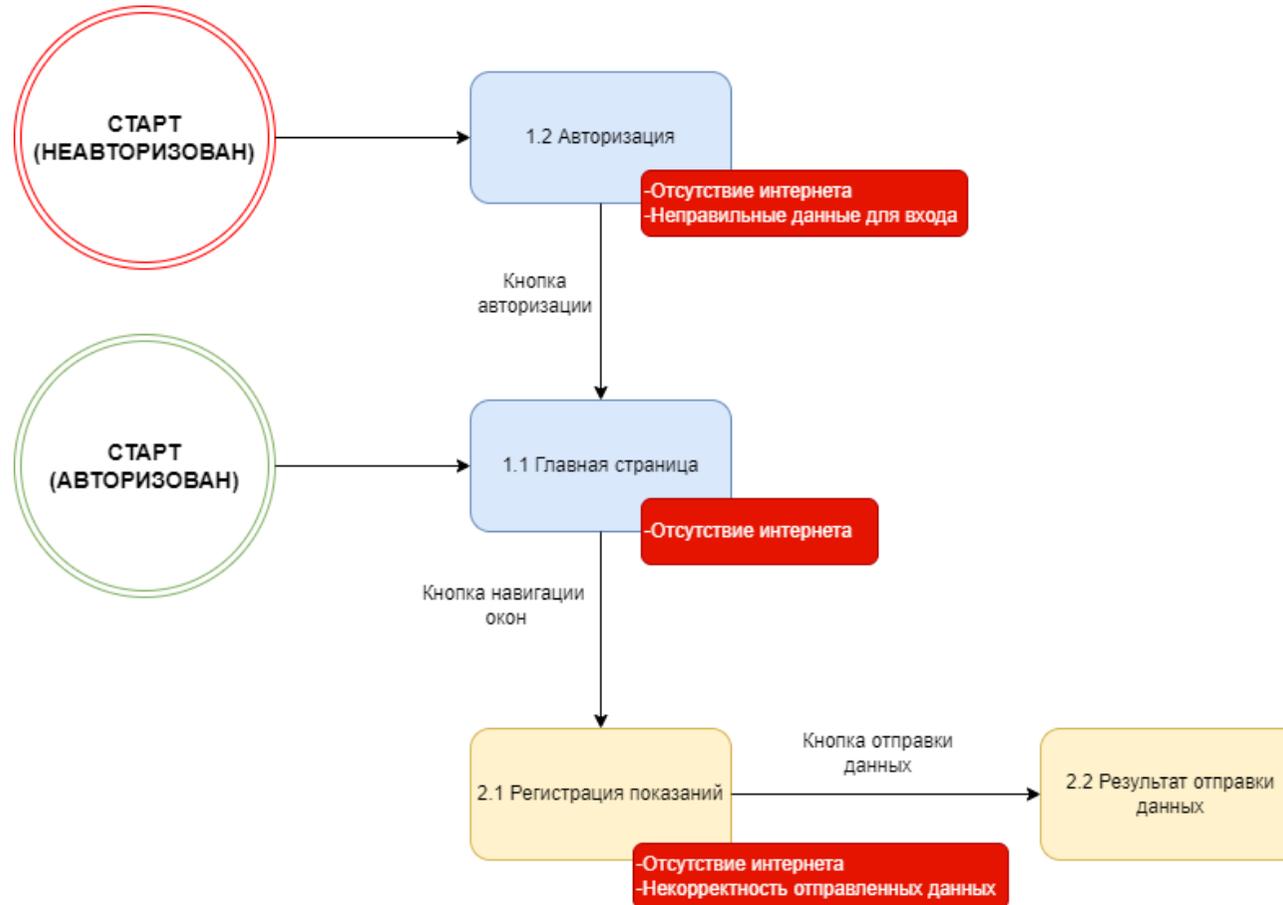


Рисунок Г.1 – Карта переходов и состояний приложения

## Приложение Д

### Разработанный прототип пользовательского интерфейса

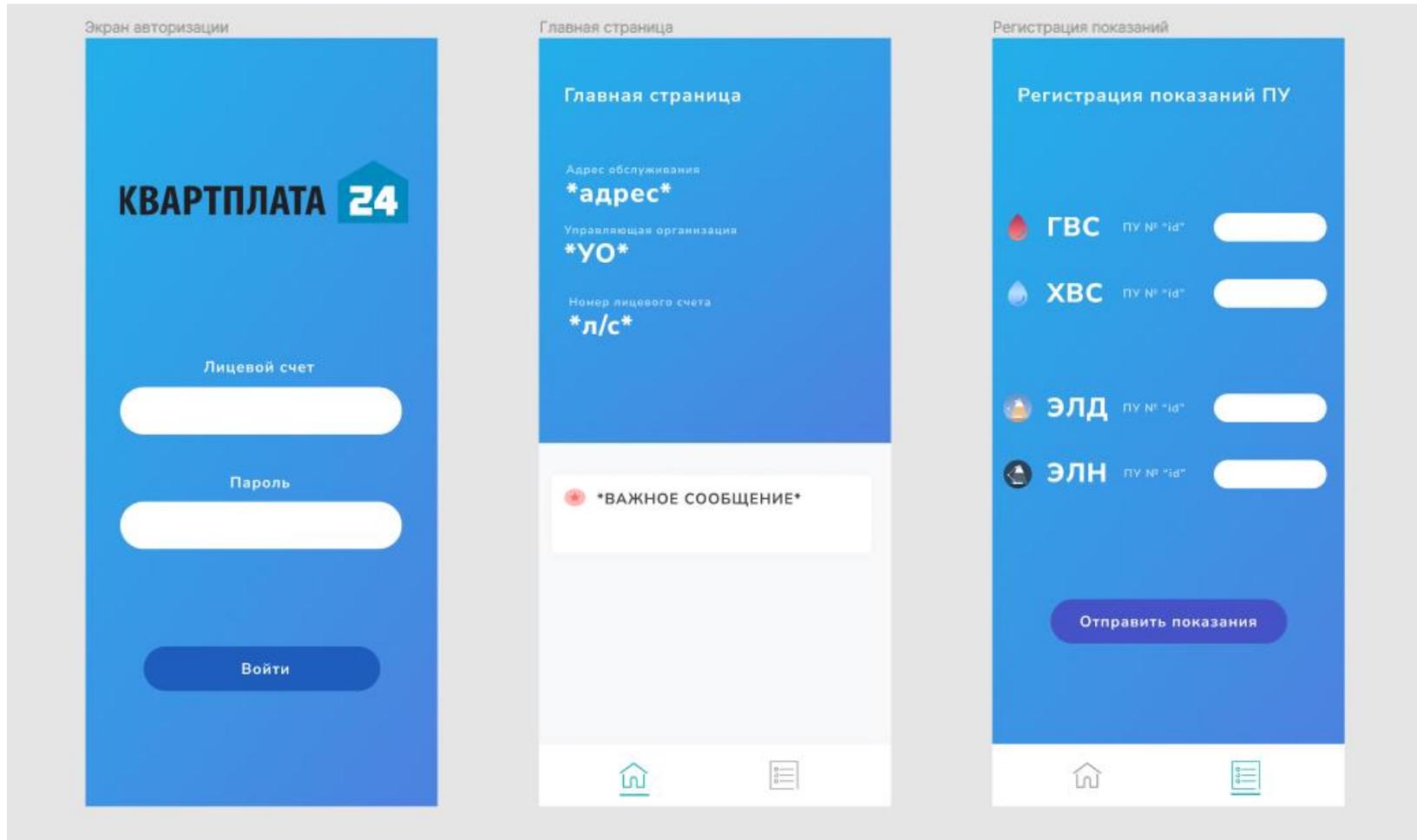


Рисунок Д.1 – Прототип пользовательского интерфейса приложения

## Приложение Е

### Позитивные тест-кейсы для тестирования

Таблица Е.1 – Позитивные тест-кейсы для тестирования

Название	Инструкции	Ожидаемый результат
Отображение пользовательского интерфейса	Открыть необходимую вкладку приложения	Графические элементы и элементы управления отображаются корректно
Навигация приложения	Переключить экраны приложения через нижнюю навигационную панель	Экраны приложения корректно переключаются, введенные данные не теряются
Отсутствие авторизации	Открыть приложение от лица не аутентифицированного пользователя	Отображение экрана с авторизацией
Наличие авторизации	Открыть приложение от лица аутентифицированного пользователя	Отображение главной страницы экрана
Корректность данных авторизации	Открыть приложение от лица не аутентифицированного пользователя, ввести корректные данные	Отображение главной страницы экрана
Корректность информации о пользователе	Открыть экран главной страницы после успешной авторизации	Отображаемые данные корректны и сходятся с реальными
Корректность отправляемых данных	Открыть экран регистрации показаний от лица аутентифицированного пользователя, ввести корректные данные 4 приборов учета, нажать кнопку «отправить показания»	Уведомление об успешной регистрации показаний