

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Мобильное приложение для учёта и оптимизации потребления  
коммунальных услуг “Bills Collector”

Курсовая работа

Направление: 09.03.04. Программная инженерия

Зав. Кафедрой \_\_\_\_\_ д. ф.-м. н, доцент С.Д. Махортов

Руководитель \_\_\_\_\_ ст. преподаватель В.С. Тарасов

Руководитель практики \_\_\_\_\_ Е.Д. Проскуряков

Обучающийся \_\_\_\_\_ А.Р. Лысенко, 3 курс, д/о

Обучающийся \_\_\_\_\_ М.С. Леонов, 3 курс, д/о

Обучающийся \_\_\_\_\_ Д. Маатук, 3 курс, д/о

Обучающийся \_\_\_\_\_ М.О.З. Тавфик, 3 курс, д/о

Воронеж 2024

# Содержание

Определения, обозначения и сокращения .....	4
Введение.....	5
1 Постановка задачи.....	6
1.1 Анализ ресторанного рынка России .....	6
1.2 Обзор аналогов.....	7
1.2.1 BillTracker.....	7
1.2.2 Splitwise.....	9
2 Анализ предметной области .....	10
2.1 Глоссарий предметной области.....	10
2.2 Процесс учёта и оптимизации потребления коммунальных услуг .....	10
2.3 Учитываемые сведения о коммунальных услугах .....	11
2.4 Учёт коммунальных услуг .....	11
2.5 Диаграмма сущность-связь .....	12
3 Реализация.....	14
3.1 Средства реализации .....	14
3.1.1 Flutter.....	14
3.1.2 MVVM.....	14
3.1.3 Kotlin .....	15
3.1.4 PostgreSQL.....	15
3.1.5 Keycloak .....	15
3.1.6 Nginx .....	15
3.1.7 Docker.....	16
3.1.8 AppMetrica.....	16
3.2 Логика приложения .....	16
3.3 Функциональные требования к разрабатываемой системе .....	16
3.3.1 Вход в приложение:.....	16
3.3.2 Список коммунальных услуг.....	17
3.3.3 Создание и редактирование коммунальной услуги .....	17
3.3.4 Аналитика и рекомендации потребления коммунальных услуг .....	17
3.3.5 Личный кабинет пользователя .....	18
3.4 Нефункциональные требования к разрабатываемой системе.....	18
3.4.1 Требования по безопасности .....	18

3.4.2 Дополнительные требования.....	18
3.5 Структура классов серверной части приложения .....	18
3.6 Структура классов моделей серверной части .....	19
3.7 Хранение данных .....	20
3.8 Развёртывание приложения .....	20
3.9 Реализация интерфейса .....	21
3.9.1 Реализация интерфейса мобильного приложения.....	21
4 Аналитика .....	31
Заключение .....	32
Список использованных источников .....	33
ПРИЛОЖЕНИЕ А Диаграмма прецедентов.....	34
ПРИЛОЖЕНИЕ Б Диаграммы состояний .....	35

## Определения, обозначения и сокращения

Термин	Определение
Услуга	коммунальная услуга, представляет собой наименование, сводную информацию и набор данных о показаниях этой коммунальной услуги
Визуализация	общее название приёмов представления числовой информации или физического явления в виде, удобном для зрительного наблюдения и анализа
Клиент, клиентская часть	компьютер, использующий ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой
Логин	частный случай идентификатора, который используется для представления конкретного пользователя в системе
Material Design	дизайн-система для создания интерфейсов программного обеспечения и приложений, разработанная компанией Google
Сервер, серверная часть	компьютер, обслуживающий другие компьютеры (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач
Цифровизация	это процесс превращения аналоговых данных и рабочих процессов в цифровой формат
Popup	всплывающее окно, которое появляется внезапно, без запроса пользователя, и отображается поверх экрана, на котором находился пользователь
REST API (REST, Representational State Transfer)	стиль архитектуры программного обеспечения для построения масштабируемых веб-приложений
Сайдменю	боковое меню для навигации в приложении
Onboarding	обучающая функциональность в приложении, появляющаяся при первом запуске для ознакомления пользователя с продуктом
Авторизация	предоставление определённого лицу или группе лиц прав на выполнение определенных действий; а также процесс проверки этих прав при попытке выполнения действия

## Введение

Современные технологии всё больше внедряются в повседневную жизнь, способствуя упрощению различных бытовых процессов и улучшению их эффективности. Одной из областей, где цифровизация может значительно повысить качество жизни, является управление коммунальными услугами. С ростом стоимости коммунальных услуг и увеличением количества потребляемой энергии, воды и других ресурсов, возникает необходимость в оптимизации их потребления [1].

В настоящее время пользователи зачастую сталкиваются с трудностями в отслеживании и управлении своими расходами на коммунальные услуги. Эти трудности включают в себя проблемы с своевременным получением данных о потреблении, неэффективное распределение ресурсов, а также недостаточную осведомлённость о способах снижения затрат.

Целью нашего проекта является создание мобильного приложения "Bills Collector", которое поможет пользователям эффективно управлять и оптимизировать свои расходы на коммунальные услуги. Мы уверены, что цифровизация этой сферы позволит пользователям более рационально использовать ресурсы и снизить свои расходы.

Для достижения этой цели мы планируем проанализировать использование существующих приложений для учёта коммунальных услуг, таких как BillTracker и Splitwise. Мы изучим, какие функции данных приложений наиболее востребованы пользователями, какие проблемы они испытывают при использовании, и какие возможности можно предложить для улучшения процесса управления коммунальными услугами.

На основе полученных данных мы разработаем удобный и интуитивно понятный интерфейс приложения "Bills Collector", который будет выгодно отличаться от конкурентов. Мы уверены, что наше приложение принесёт реальную пользу пользователям и поможет оптимизировать процесс учёта и потребления коммунальных услуг.

## 1 Постановка задачи

Целью данного курсового проекта является создание мобильного приложения для учёта и оптимизации потребления коммунальных услуг. Данное мобильное приложение разрабатывается с целью:

- предоставление конечному пользователю приложения для учёта и оптимизации потребления коммунальных;
- в течение трех месяцев после начала использования приложения и соблюдения рекомендаций по потреблению, уменьшение расходов пользователя на коммунальные услуги должно составить не менее 1 процента в сравнении с периодом до начала использования приложения, при условии превышения заложенной в приложение нормы потребления пользователя не менее чем на 5 процентов.

Приложение должно обладать следующей функциональностью:

- цифровизация процесса учёта потребления коммунальных услуг;
- определение расходов пользователя на коммунальные услуги в зависимости от указанных пользователем данных;
- аналитика и предоставление рекомендаций по потреблению коммунальных услуг.

### 1.1 Анализ ресторанного рынка России

Согласно исследованию BusinesStat, 2018-2022 годах оборот рынка жилищно-коммунальных услуг в России вырос на 28,8%: с 2,88 триллиона рублей до 3,71 триллиона рублей [2] (см. таблица 1).

Таблица 1 — Оборот рынка жилищно-коммунальных услуг в России в 2018-2022 гг (млрд руб)

Параметр	2018	2019	2020	2021	2022
Оборот рынка жилищнокоммунальных услуг (млрд руб)	2 882,0	3 022,3	3 061,0	3 359,2	3 707,0
% к предыдущему году	-	4,9	1,3	9,7	10,4

Также в России отмечается рост тарифов на жилищно-коммунальное обслуживание. Средний размер взносов на капитальный ремонт за 2018-2022 гг вырос на 48,6%, тариф на водоотведение увеличился на 24,8%, на отопление – на 20,6%, на холодное и горячее водоснабжение – на 19,7% и 19,4% соответственно [2].

Аналитики также связывают такой рост с госпрограммами, направленными на обеспечение граждан доступным жильем, расселение аварийного жилого фонда, льготное ипотечное кредитование.

## 1.2 Обзор аналогов

В ходе исследования рынка приложений для учёта и оптимизации потребления коммунальных услуг было выявлено 2 прямых конкурента. Результаты конкурентного исследования приведены в таблица 2.

Таблица 2 — Конкурентное исследование

Характеристика	BillTracker	Splitwise
Учет потребления услуг	+	+
Наличие транзакций и платных подписок	+( \$9.99)	+( от \$2.99 до \$39.99)
Кроссплатформенность	-	+
Наличие перевода на русский язык	-	-

### 1.2.1 BillTracker

Основными функциями приложения являются: отслеживание счетов, оповещения с напоминаниями, история платежей, сводки и анализ счетов, безопасность данных, синхронизация между устройствами, поддержка клиентов. К недостаткам можно отнести доступность только на операционной системе iOS и недоступность русского языка. На рисунках рисунок 1 и рисунок 2 показаны элементы интерфейса приложения:

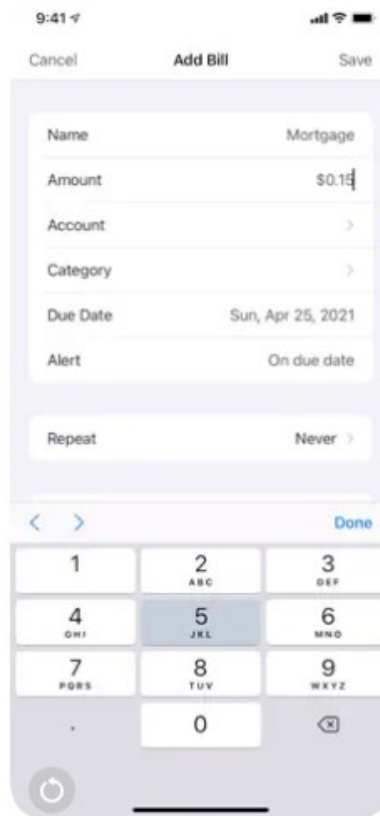


Рисунок 1 — Добавление услуги

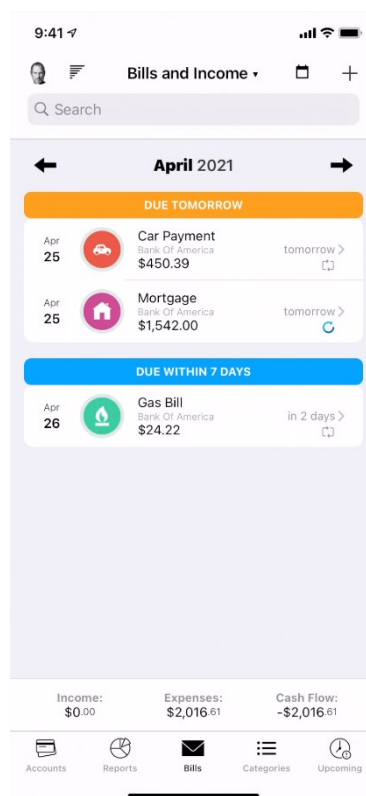


Рисунок 2 — Информация о платежах



### 1.2.2 Splitwise

Основными функциями приложения являются: отслеживание расходов, разделение счетов, управление группами, отслеживание задолженности, интеграция платежей, уведомления, категоризация, лента активности, автономная поддержка, кроссплатформенный доступ. К недостаткам можно отнести недоступность русского языка. Рисунок 3 демонстрирует элементы интерфейса приложения:

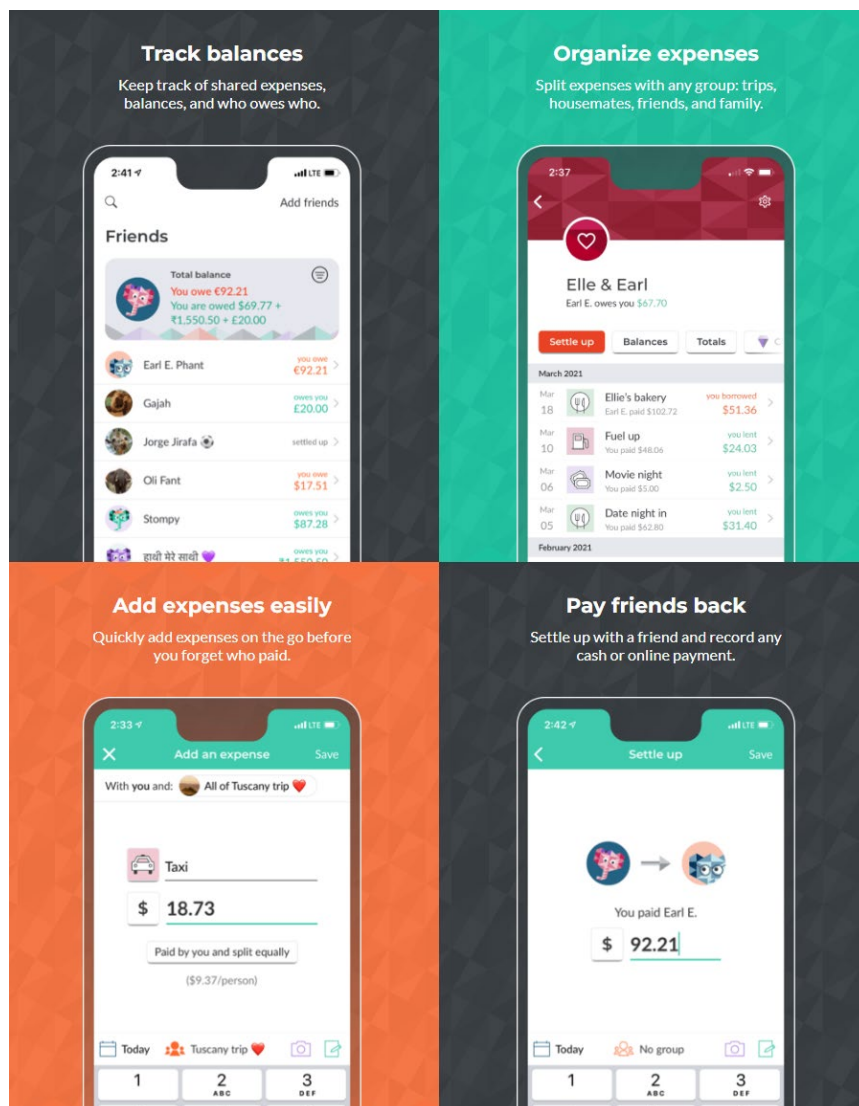


Рисунок 3 — Демонстрация интерфейса приложения Splitwise

## **2 Анализ предметной области**

### **2.1 Глоссарий предметной области**

Пользователь – это лицо, использующее приложение для учёта и оптимизации потребления коммунальных услуг.

Коммунальная услуга – это услуга, предоставляемая коммунальными службами, включая электроэнергию, воду, газ и отопление.

Потребление – это количество использованных коммунальных услуг за определённый период.

Тариф – это стоимость коммунальной услуги, установленная поставщиком за единицу потребления.

Аналитика – это процесс обработки и интерпретации данных о потреблении коммунальных услуг для выявления тенденций и возможностей оптимизации.

Рекомендация – это предложенные действия по уменьшению потребления коммунальных услуг и снижению затрат на них.

Личный кабинет – это персональная страница пользователя в приложении, содержащая информацию о его учетных записях и потреблении коммунальных услуг.

### **2.2 Процесс учёта и оптимизации потребления коммунальных услуг**

Процесс управления коммунальными услугами в приложении "Bills Collector" состоит из следующих шагов (см. рисунок 4):

- а) регистрация пользователя в приложении;
- б) ввод данных о потребляемых коммунальных услугах (название услуги, тариф, данные потребления);
- в) мониторинг и обновление данных о потреблении коммунальных услуг;
- г) получение аналитических данных и рекомендаций по оптимизации потребления;
- д) применение рекомендаций и анализ результатов для снижения затрат.

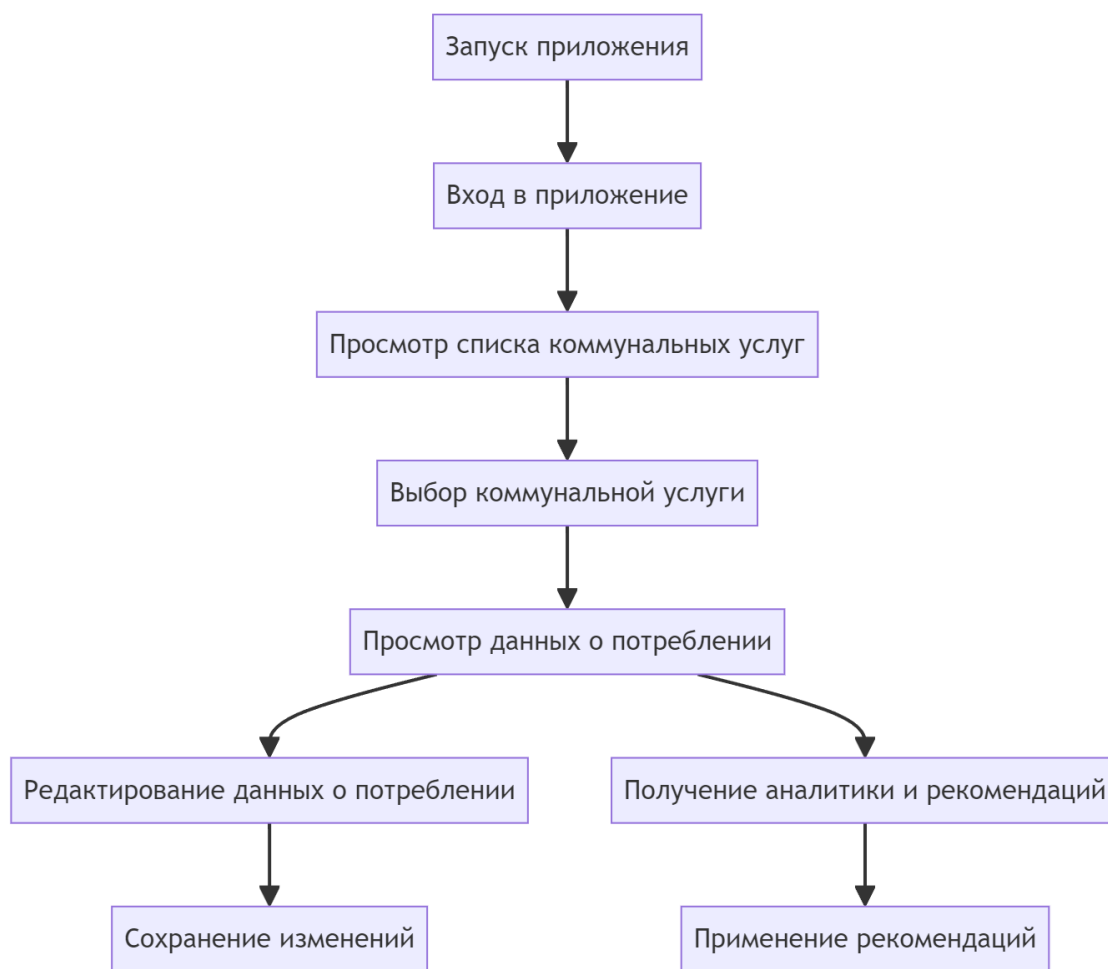


Рисунок 4 — Диаграмма активности работы пользователя с коммунальной услугой

### 2.3 Учитываемые сведения о коммунальных услугах

При резервации брони в программу или книгу резервов вносятся следующие данные:

- название коммунальной услуги;
- дата учёта потребления;
- данные о потреблении коммунальной услуги;
- дополнительную информацию в виде комментария.

### 2.4 Учёт коммунальных услуг

Для авторизованного пользователя приложение должно предоставлять доступ к:

- просмотру списка всех добавленных коммунальных услуг;
- просмотру данных о потреблении за выбранный период;

- созданию, редактированию и удалению информации о коммунальных услугах;
- получению аналитических данных и рекомендаций по оптимизации потребления.

При создании и редактировании данных о коммунальной услуге система должна позволять учитывать следующие сведения:

- название коммунальной услуги;
- дата учёта потребления;
- данные о потреблении коммунальной услуги;
- тариф на коммунальную услугу;
- дополнительную информацию в виде комментария.

Как результат, эта информация должна сохраниться в системе и быть доступной для пользователя для дальнейшего анализа и оптимизации.

## **2.5 Диаграмма сущность-связь**

На основе анализа предметной сущности были выделены сущности «Пользователь», «Коммунальная Услуга», «Аналитика» (см. рисунок 5).

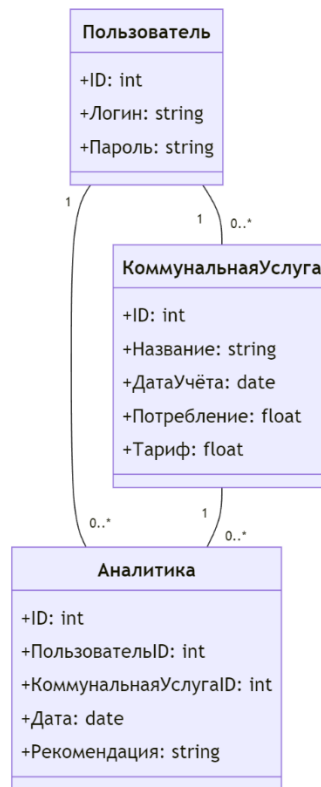


Рисунок 5 — Диаграмма сущностей и связей системы

## **3 Реализация**

### **3.1 Средства реализации**

#### **3.1.1 Flutter**

Для реализации клиентской части веб-приложения использовался фреймворк Flutter версии 3.19.3 и язык программирования Dart 3.3.1.

Flutter – это написанный на Dart кроссплатформенный фреймворк для разработки мобильных, веб- и десктопных приложений. К основным плюсам Flutter можно отнести:

- ускоренную разработку;
- высокую производительность;
- развитое и активное сообщество разработчиков.

Ускорение разработки достигается за счёт использования виджетов, что позволяет применять объектно-ориентированный подход к любому элементу интерфейса.

Высокая производительность достигается за счёт графического движка Skia, обеспечивающего компиляцию исходного кода приложения непосредственно в нативный код платформы, а также за счёт рендеринга пользовательского интерфейса на видеокарте.

Благодаря популярности Flutter существует развитая система сторонних программных пакетов, позволяющих расширить функциональность фреймворка [3].

#### **3.1.2 MVVM**

MVVM (Model-View-ViewModel) – это архитектурный шаблон, используемый при разработке приложений для разделения пользовательского интерфейса и логики выполнения.

За счёт привязки данных модели и представления через viewmodel обеспечивается автоматическое обновление пользовательского интерфейса при изменении данных модели. Это позволяет облегчить и ускорить разработку приложений.

Также за счёт гибкости данного шаблона появляется возможность изменять пользовательский интерфейс без необходимости большого изменения логики выполнения.

### **3.1.3 Kotlin**

Для реализации серверной части приложения использовался язык программирования Kotlin версии 1.9.23 и фреймворк Spring Boot 3.2.3.

Язык программирования Kotlin обладает лаконичным и выразительным синтаксисом, поддержкой функционального программирования, а также совместимостью с Java, что позволяет использовать обширную стандартную библиотеку Java и существующие фреймворки. Удобные инструменты сборки, такие как Gradle, позволяют быстро подключать необходимые зависимости и управлять проектом.

Фреймворк Spring Boot 3 [4] обеспечивает лёгкость создания веб-приложений за счёт развитой системы аннотаций Kotlin.

### **3.1.4 PostgreSQL**

Для хранения данных используется реляционная база данных PostgreSQL 16.2. За счёт производительности в долгосрочной перспективе позволит значительно увеличить клиентскую базу без необходимости смены технологии.

### **3.1.5 Keycloak**

Для аутентификации пользователей используется Keycloak 23.0.7. В будущем этот сервис обеспечит нам возможность лёгкой реализации входа через сторонние сервисы.

### **3.1.6 Nginx**

В качестве веб-сервера для клиентской части использовался Nginx версии 1.25.

Данный веб-сервер обеспечивает высокую скорость обработки запросов при работе со статическим контентом, не требующим постоянного обновления.

### **3.1.7 Docker**

Для развёртывания приложения использовался Docker версии 26.1.1.

Docker – это инструмент контейнеризации приложений, позволяющий обеспечить переносимость приложений и ускорить разработку. За счёт изолированности процессов приложение не сможет затронуть внешнее окружение, что обеспечивает повышенную безопасность.

Инструмент Docker compose позволяет легко настроить запуск нескольких контейнеров за счёт декларативного описания запускаемых сервисов и сетей.

### **3.1.8 AppMetrica**

Для сбора данных о работе приложения, пользовательских событиях и аналитики используется сервис AppMetrica [5]. За счёт удобного SDK и гибкой настройки данный сервис позволяет легко настроить отправку событий пользователей, а также строить на основе полученных данных аналитические отчёты, например, воронки конверсий.

## **3.2 Логика приложения**

В системе пользователь имеет одну из ролей:

- неавторизованный пользователь;
- авторизованный пользователь.

## **3.3 Функциональные требования к разрабатываемой системе**

Обзор функциональных требований описан ниже. Далее приведено подробное описание функциональных требований.

### **3.3.1 Вход в приложение:**

- при первом включении приложения пользователю должен быть показан onboarding;
- приложение должно позволять войти или зарегистрироваться по логину и паролю;
- приложение должно позволять пользователю начать пользоваться приложением с ограниченными возможностями без авторизации.



### **3.3.2 Список коммунальных услуг**

Для пользователя вне зависимости от авторизации система должна предоставлять доступ к:

- просмотру списка добавленных ранее видов коммунальных услуг;
- возможности добавить новый вид коммунальной услуги.

Как результат, эта информация должна быть получена и сохранена на сервере при пройденной авторизации, или на клиенте в иных случаях, и быть доступной для пользователя.

### **3.3.3 Создание и редактирование коммунальной услуги**

При создании и редактировании коммунальной услуги система должна позволять учитывать следующие сведения:

- название коммунальной услуги;
- дата учета для коммунальной услуги;
- данные по потреблению коммунальной услуги;
- данные о тарифе или иные данные для определения расходов на коммунальную услугу.

Как результат, эта информация должна сохраниться в системе и быть доступной для пользователя.

### **3.3.4 Аналитика и рекомендации потребления коммунальных услуг**

Каждому пользователю должна быть предоставлена аналитика потребления коммунальных услуг в произвольной форме (графики, диаграммы или прочие формы визуализации) на основе введенных пользователем данных.

Каждому пользователю должна быть предоставлена как минимум одна рекомендация, учитывающая введенные пользователем данные.

Как результат, эта информация должна быть доступна для пользователя и учтена им для оптимизации потребления коммунальных услуг.

### **3.3.5 Личный кабинет пользователя**

При пройденной авторизации пользователем, система должна предоставлять доступ к:

- Просмотру личного кабинета;
- Возможности смены логина;
- Возможности смены пароля.

Как результат, эта информация должна сохраниться в системе на сервере.

## **3.4 Нефункциональные требования к разрабатываемой системе**

### **3.4.1 Требования по безопасности**

- Обмен данных между клиентом и сервером должен осуществляться по протоколу HTTPS;
- пароли пользователей должны храниться в базе данных в хешированном виде; для хеширования должен использоваться алгоритм PBKDF2;
- пользователи должны иметь возможность просмотра, редактирования и удаления только относящихся к ним данных.

### **3.4.2 Дополнительные требования**

При первом запуске мобильного приложения пользователь должен иметь возможность ознакомиться с основными функциями приложения.

## **3.5 Структура классов серверной части приложения**

Классовая структура серверной части приложения организована по принципу слоистой архитектуры с выделением классов контроллеров, обеспечивающих обработку HTTP запросов; сервисов, содержащих бизнес-логику приложения, и репозитория Spring data JPA, обеспечивающих доступ к данным в базе данных (см. рисунок 6).

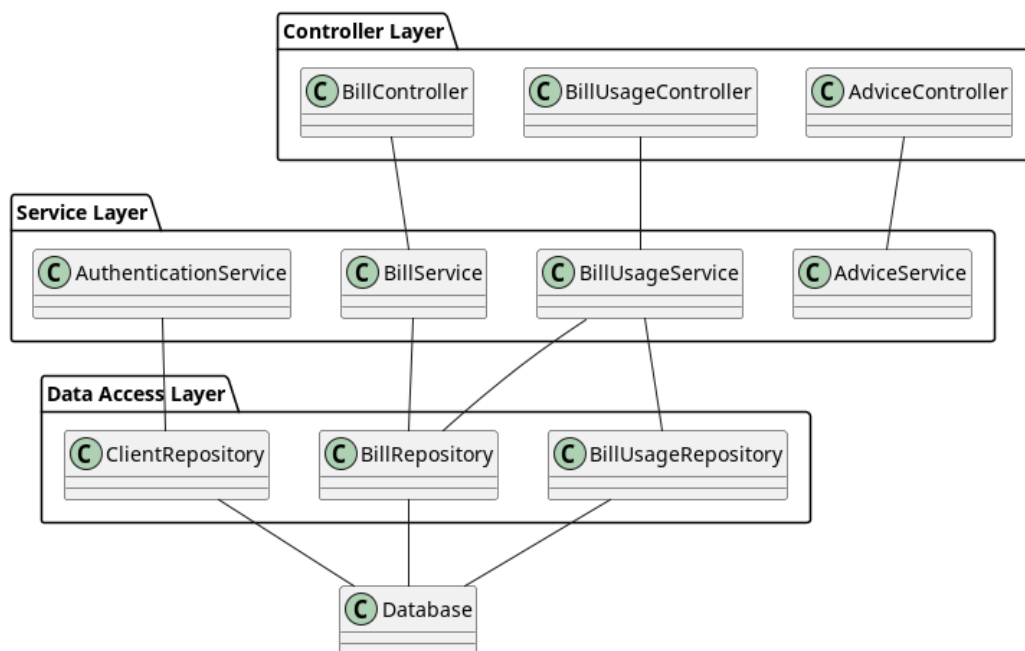


Рисунок 6 — Диаграмма классов серверной части приложения

### 3.6 Структура классов моделей серверной части

Сущности, выявленные в ходе анализа предметной области, представлены в программе серверной части классами моделей (см. рисунок 7).

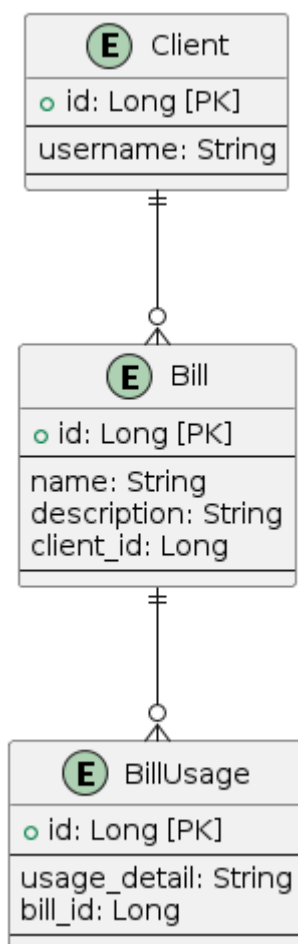


Рисунок 7 — Классы модели

### 3.7 Хранение данных

Данные приложения хранятся в реляционной базе данных PostgreSQL. Физическая схема базы данных соответствует схеме классов моделей. Данные для аутентификации пользователей хранятся в базе данных Keycloak.

### 3.8 Развёртывание приложения

Серверное приложение поставляется в виде контейнеров Docker. Для запуска приложения используется Docker compose. Для обеспечения работы https протокола используется Nginx (см. рисунок 8).

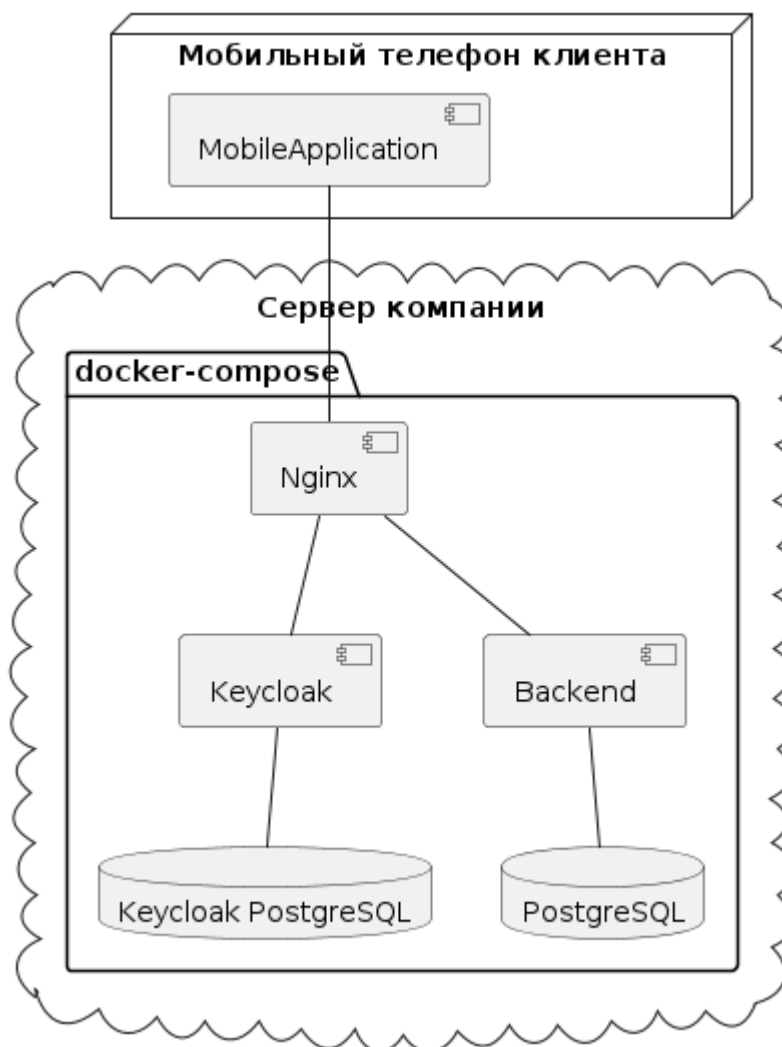


Рисунок 8 — Диаграмма развёртывания приложения

## 3.9 Реализация интерфейса

### 3.9.1 Реализация интерфейса мобильного приложения

Экраны мобильного приложения оформлены в едином стиле, отвечающем нормам Material Design с использованием ограниченного набора шрифтов, одинаково и корректно отображаются на устройствах Google Pixel с операционной системой Android 14 и выше.

При первом заходе в приложение пользователю показывается onboarding, содержащий краткую информацию по использованию приложения (см. рисунок 9)

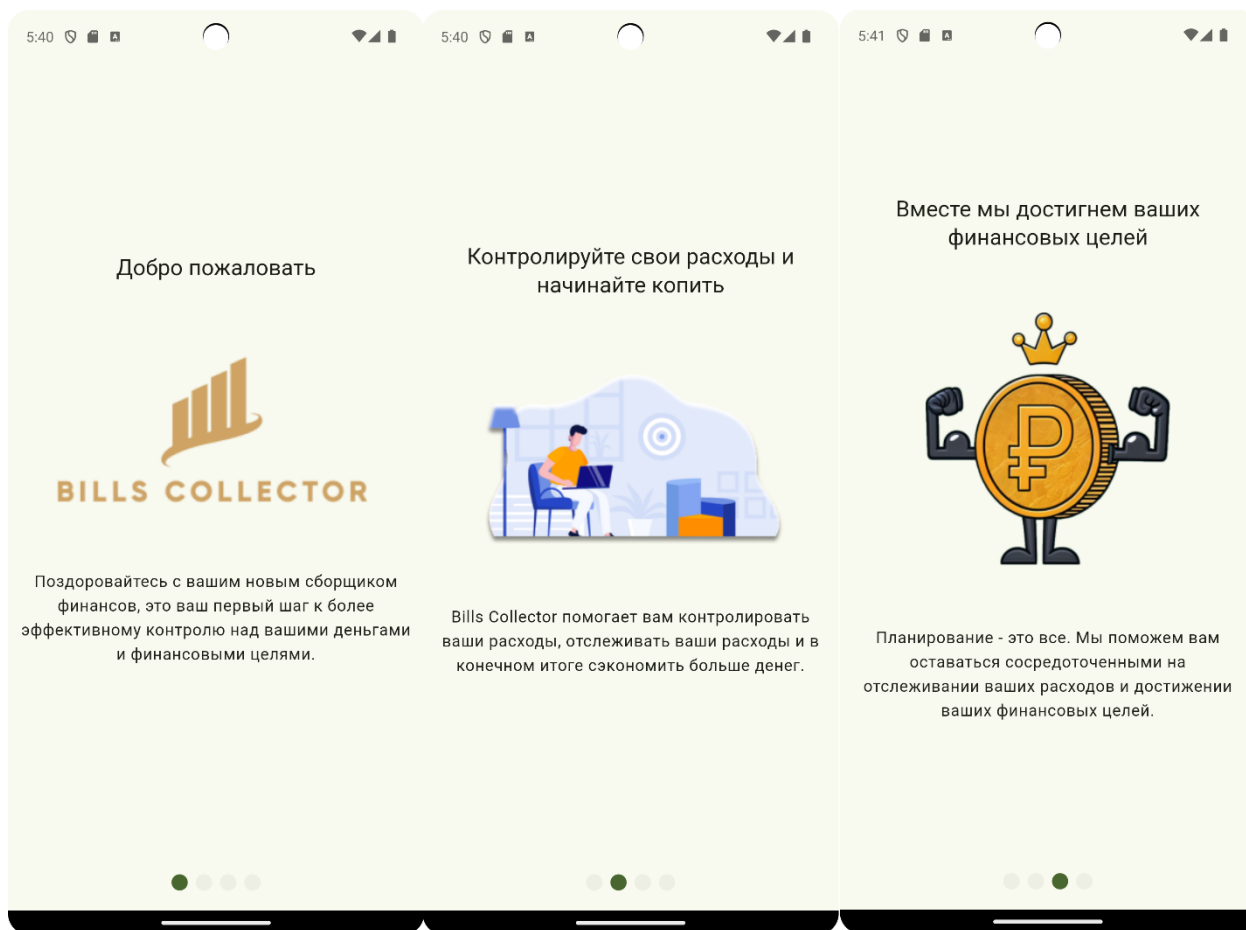


Рисунок 9 — Onboarding при первом входе в приложение

Если пользователь хочет продолжить без авторизации, зарегистрироваться или авторизоваться с помощью логина и пароля, то он может сделать это на соответствующих экранах (см. рисунок 10).

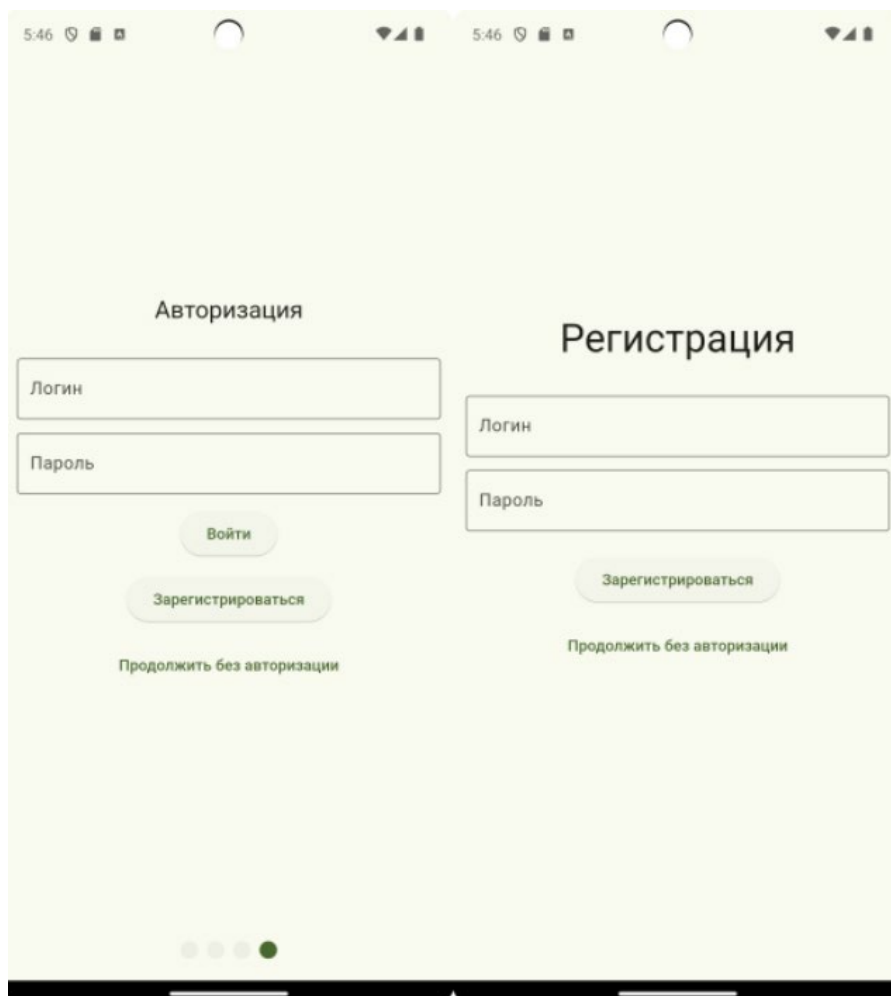


Рисунок 10 — Экран с возможностью продолжения без авторизации, регистрации или авторизации

Первым экраном, который видит пользователь, является экран обзора коммунальных услуг (см. рисунок 11). На нём отображается список коммунальных услуг, добавленных пользователем.

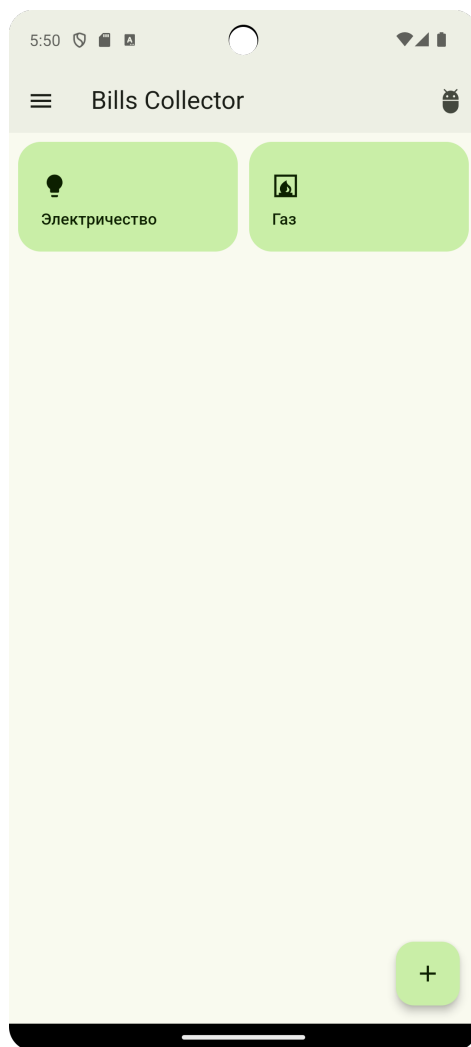


Рисунок 11 — Экран обзора коммунальных услуг пользователя

По нажатию на кнопку «Добавление услуги» открывается форма для добавления услуги (см. рисунок 12).



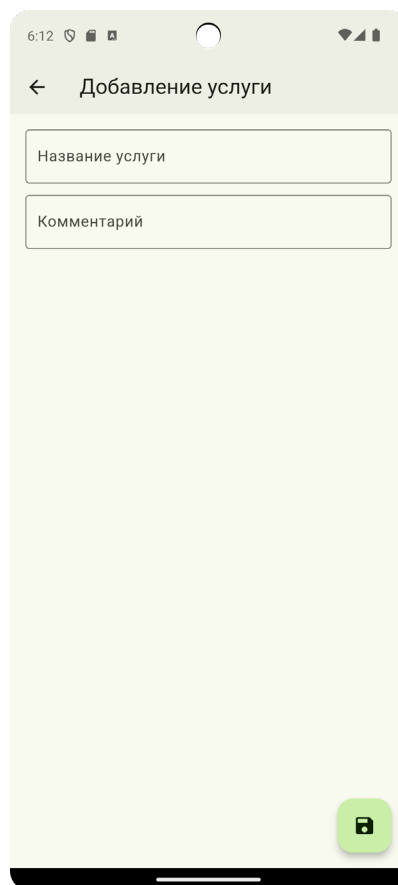


Рисунок 12 — Форма добавления коммунальной услуги

При нажатии на определённую коммунальную услугу открывается экран с подробной информацией о выбранной услуге и возможностью её редактирования (см. рисунок 13).



Рисунок 13 — Экран с информацией о коммунальной услуге

При нажатии на кнопку «Добавить» открывается экран с возможностью добавления информации о потреблении коммунальной услуги, а также возможностью указать расчётный период (дату) (см. рисунок 14), а при нажатии кнопки «Удалить» появляется рорип с предупреждением об удалении (см. рисунок 15).

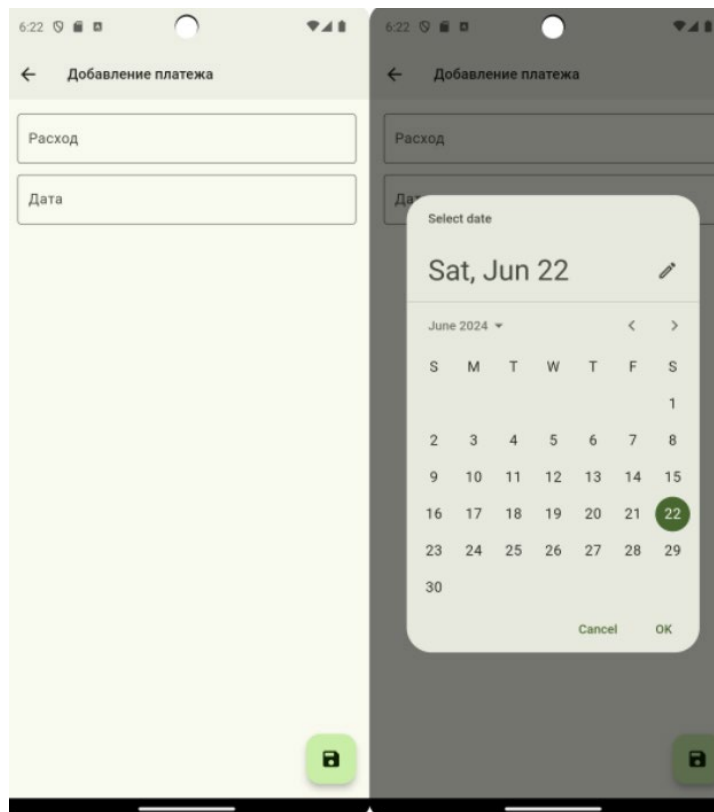


Рисунок 14 — Экраны с добавлением информации о потреблении коммунальной услуги и даты

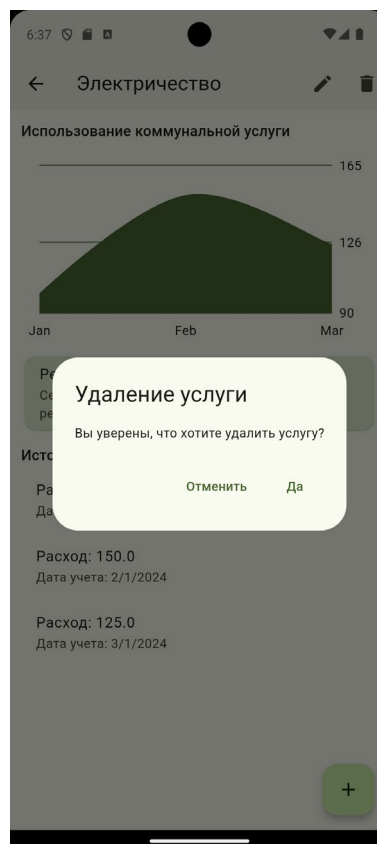


Рисунок 15 — Рорир с предупреждением об удалении коммунальной услуги

При нажатии на кнопку «Редактировать» открывается экран с возможностью редактирования информации о коммунальной услуге (см. рисунок 16).

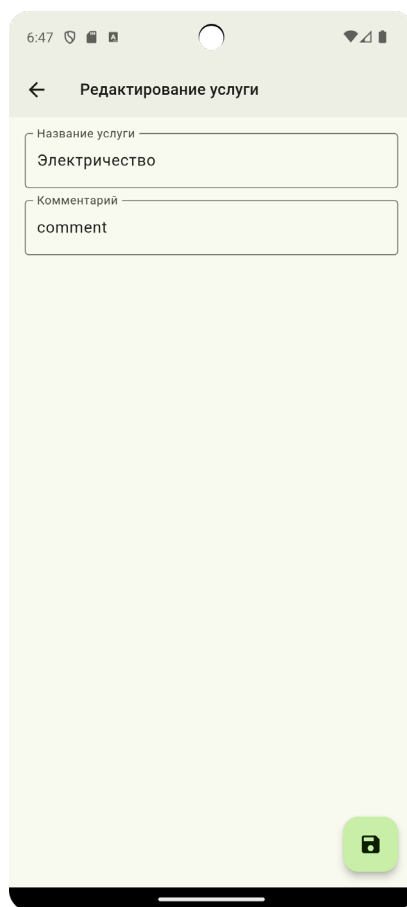


Рисунок 16 — Экран с редактированием коммунальной услуги

В профиле пользователя отображается логин, а также приложение позволяет изменить пароль (см. рисунок 17).

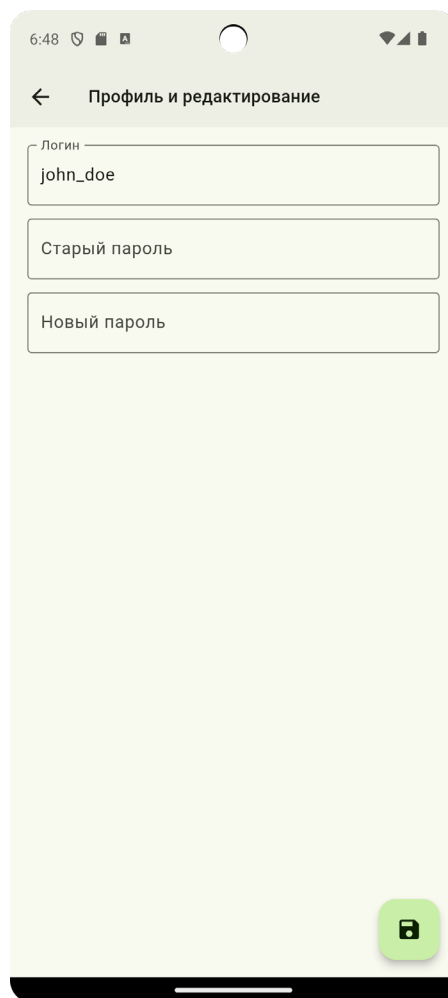


Рисунок 17 — Профиль пользователя

При нажатии на кнопку «Рекомендации по потреблению» открываются аналитика и рекомендации по потреблению коммунальной услуги (см. рисунок 18).

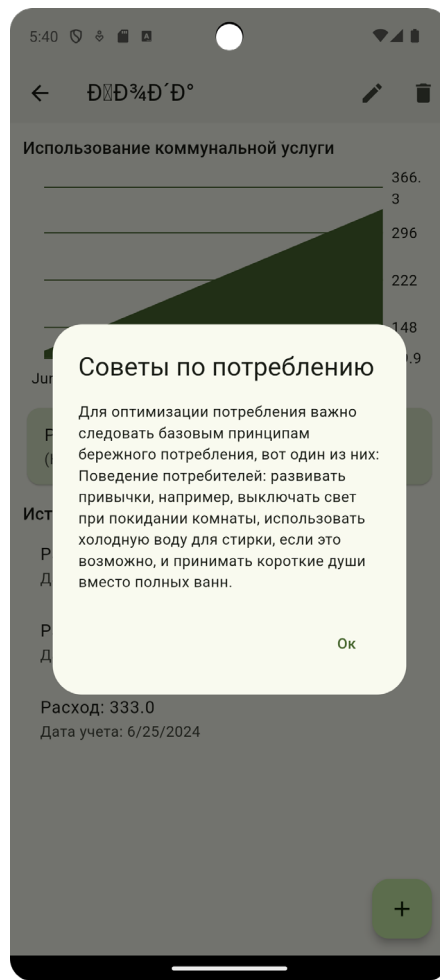


Рисунок 18 — Рекомендации по потреблению

#### 4 Аналитика

Для тестирования работоспособности приложения доступ к нему был предоставлен студентам факультета компьютерных наук. Приложение было установлено 14 людьми. В результате эксплуатации, были получены данные для воронки по основному сценарию – «Создание коммунальной услуги, добавление потребления коммунальной услуги, получение аналитики». Из них видно, что до этапа получения аналитики дошло 7% человек (см. рисунок 19). Из этого можно сделать вывод, что некоторые элементы формы добавления потребления коммунальной услуги стоит сделать более удобными для пользователя.

##### Конверсия шагов

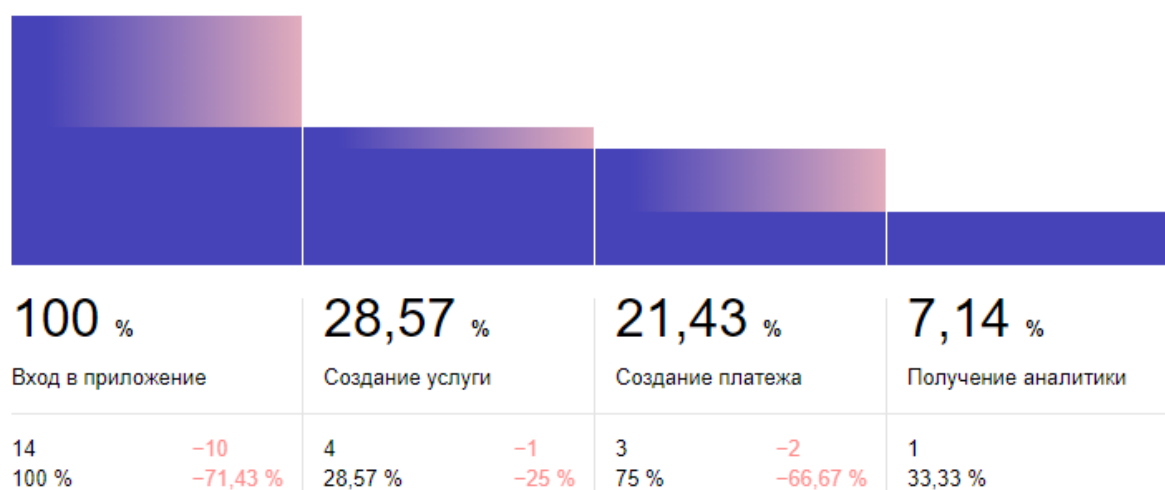


Рисунок 19 — Воронка создания коммунальной услуги, добавления потребления коммунальной услуги и получения аналитики

## **Заключение**

В ходе реализации данного проекта были выполнены поставленные задачи. Разработанное приложение удовлетворяет предъявленным требованиям, а именно:

- обеспечивает возможность создавать и редактировать коммунальную услугу;
- обеспечивает возможность создавать потребление по коммунальной услуге;
- предоставляет аналитику и рекомендации по потреблению коммунальной услуги.

В результате проведённого тестирования было установлено, что готовое приложение успешно прошло все выполненные тесты на различные сценарии использования, тем самым подтвердив свою работоспособность.



## **Список использованных источников**

1. ЖКХ в России [Электронный ресурс]. – Режим доступа: URL: [https://www.tadviser.ru/index.php/Статья:ЖКХ\\_в\\_России](https://www.tadviser.ru/index.php/Статья:ЖКХ_в_России) – заглавие с экрана (Дата обращения 27.05.2024)
2. Анализ рынка жилищно-коммунальных услуг в России в 2019-2023 гг, прогноз на 2024-2028 гг [Электронный ресурс]. – Режим доступа: URL: <https://businessstat.ru/catalog/id11542/> - Заглавие с экрана (Дата обращения 27.05.2024)
3. Pub.dev [Электронный ресурс]. – Режим доступа: URL: <https://pub.dev/> - Заглавие с экрана (Дата обращения 20.05.2024)
4. Документация SpringBoot [Электронный ресурс]. – Режим доступа: URL: <https://docs.spring.io/spring-boot/docs/current/reference/html/> – Заглавие с экрана. – (Дата обращения 23.05.2024).
5. Документация AppMetrica [Электронный ресурс]. – Режим доступа: URL: <https://appmetrica.yandex.ru/docs/ru/> – Заглавие с экрана. – (Дата обращения 25.05.2024).

## ПРИЛОЖЕНИЕ А

### Диаграмма прецедентов



Рисунок А.1 — Диаграмма прецедентов

## ПРИЛОЖЕНИЕ Б

### Диаграммы состояний



Рисунок Б.1 — Диаграмма состояний для услуги