МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Курсовой проект

Приложение для ведения домашнего бюджета и рекомендаций по сокращению расходов

09.03.04 Программная инженерия

Информационные системы и сетевые технологии

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *А.А. Мясоедов, 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *А.А. Донцов, 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *А.А. Золотова, 3 курс, д/о*

Воронеж 2022

Содержание

[Содержание 2](#_Toc102213862)

[1. Анализ предметной области 3](#_Toc102213863)

[2. Постановка задачи 4](#_Toc102213864)

[2.1. Функциональные требования 4](#_Toc102213865)

[2.2. Технические требования 4](#_Toc102213866)

[2.3. Требования к интерфейсу 4](#_Toc102213867)

[3. Макеты интерфейса 5](#_Toc102213868)

[4. Анализ аналогов 13](#_Toc102213869)

[4.1. CoinKeeper 13](#_Toc102213870)

[4.2. 1Money: расходы финансы бюджет 14](#_Toc102213871)

[5. Анализ задачи 16](#_Toc102213872)

[5.1. Средства реализации 16](#_Toc102213873)

[5.2. Входные-выходные данные(IDEF0) 16](#_Toc102213874)

[5.3. Схема БД 17](#_Toc102213875)

[5.4. Диаграммы классов 18](#_Toc102213876)

[5.5. Диаграммы вариантов использования 21](#_Toc102213877)

[5.6. Диаграмма потоков данных 22](#_Toc102213878)

[5.7. Диаграмма состояний 23](#_Toc102213879)

[5.8. Структурная схема приложения 23](#_Toc102213880)

[5.9. Диаграммы последовательностей 24](#_Toc102213881)

[6. Сценарии воронок 26](#_Toc102213882)

[7. Список планируемых работ 27](#_Toc102213883)

[8. Календарный план 28](#_Toc102213884)

1. Анализ предметной области

Человек каждый день совершает покупки, оплачивает счета или услуги. Число таких расходов ежедневно может переваливать за несколько десятков. Запоминать все расходы и держать их в голове не представляется возможным. Вести учёт на бумаге не всегда удобно, так как это не очень быстро и записной книжки может не оказаться под рукой. Вдобавок считать расходы и составлять статистику по ним придётся вручную, что повышает вероятность ошибок. Также у человека не всегда остаются время и силы на анализ своих расходов и составление плана по их перераспределению или сокращению. Автоматизация ведения домашнего бюджета, учёта расходов, составления по ним статистики и рекомендаций по их сокращению – это экономия времени и финансовых средств. В современном мире почти у каждого есть смартфон, который всегда находится под рукой. В связи с этим ведение домашнего бюджета и учёт расходов в мобильном приложении становятся более доступными и актуальными.

Цель данной работы – разработка мобильного приложения для ведения домашнего бюджета и рекомендаций по сокращению расходов. Мобильное приложение будет использоваться для учёта расходов на различные категории(сферы жизни), просмотра статистики по расходам и рекомендаций по их сокращению.

1. Постановка задачи

Разработать мобильное приложение для ведения домашнего бюджета и рекомендаций по сокращению расходов.

Приложение предназначено для:

* Упрощения ведения домашнего бюджета, учёта расходов
* Распределения расходов по категориям
* Предоставления рекомендаций по сокращению расходов
  1. Функциональные требования

1. Добавление собственных категорий расходов
2. Добавление записей о расходах на различные категории и записей о пополнении баланса
3. Просмотр списка категорий с суммой расходов на категорию за текущий месяц
4. Просмотр списка расходов и пополнений
5. Просмотр списка расходов на конкретную категорию
6. Редактирование категории
7. Редактирование записи о расходе или пополнении
8. Просмотр статистики за текущий и предыдущий месяц и рекомендаций
   1. Технические требования
9. Авторизация пользователей посредством адреса электронной почты и пароля
10. Запрет на доступ к ресурсам неидентифицированным пользователям
11. Удаление транзакций, относящихся к категории при удалении этой категории у пользователя
    1. Требования к интерфейсу
12. Все страницы должны быть оформлены в одинаковом стиле в соответствии с принципами Flat Design
13. Все элементы управления должны быть выделены на фоне основного содержимого страницы
14. Макеты интерфейса



Рисунок 1 – Макет страницы входа

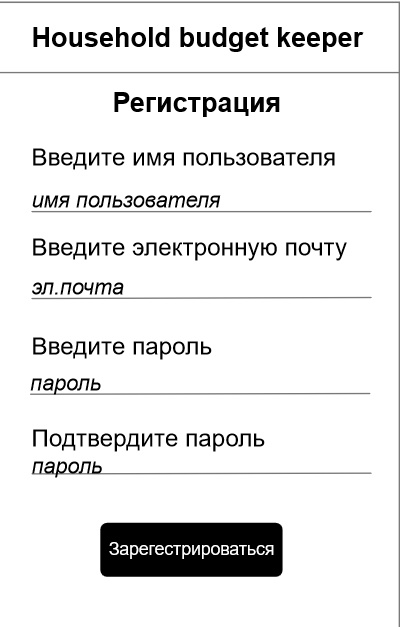


Рисунок 2 – Макет страницы регистрации

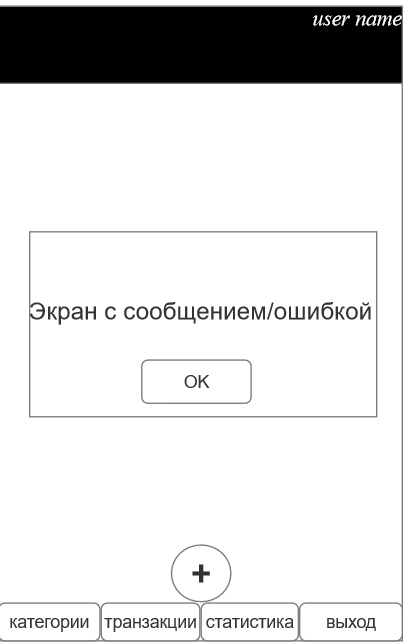


Рисунок 3 – Макет страницы с сообщением об ошибке/уведомлением

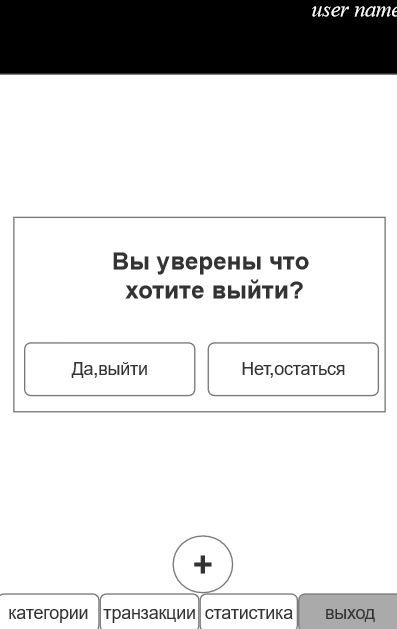


Рисунок 4 - Макет страницы выхода

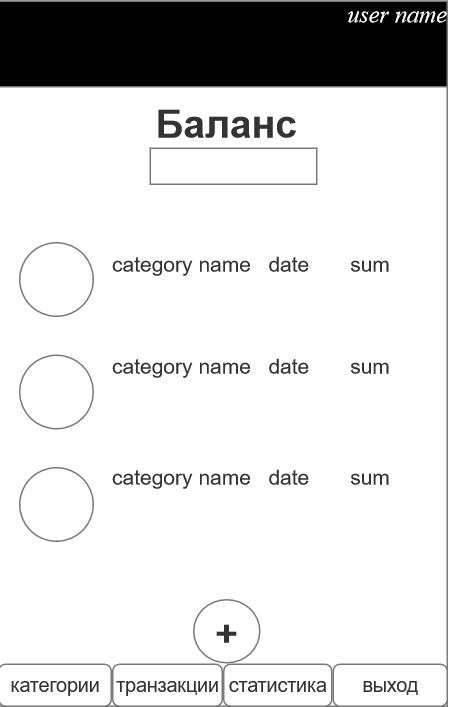


Рисунок 5 – Макет страницы с транзакциями

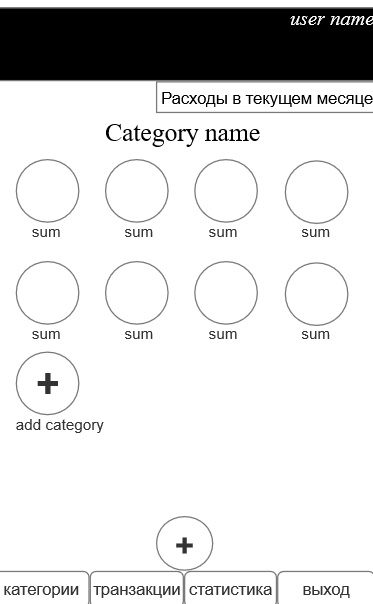


Рисунок 6 – Макет страницы с категориями



Рисунок 7 – Макет страницы с транзакциями категории

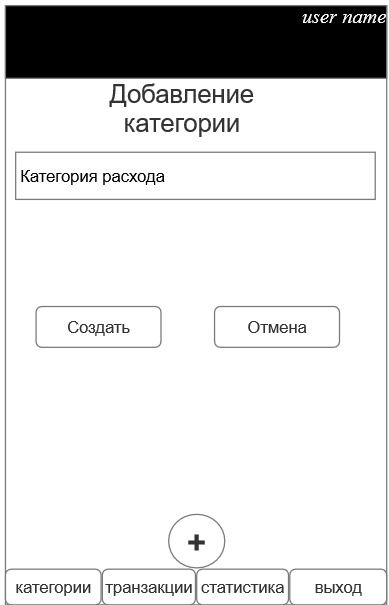


Рисунок 8 – Макет страницы добавления категории

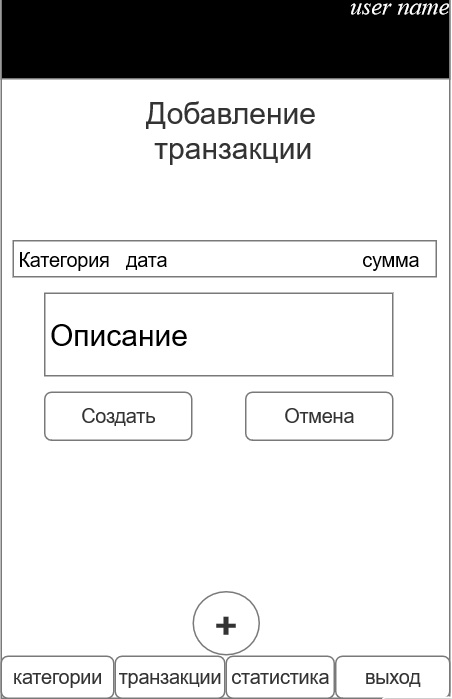


Рисунок 9 – Макет страницы добавления транзакции

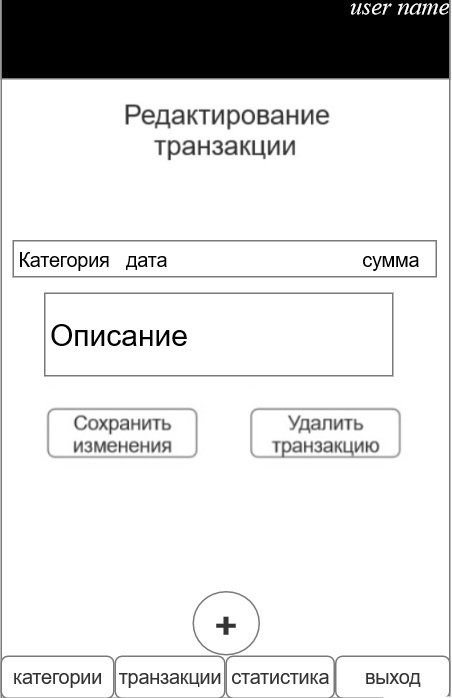


Рисунок 10 – Макет страницы редактирования транзакции

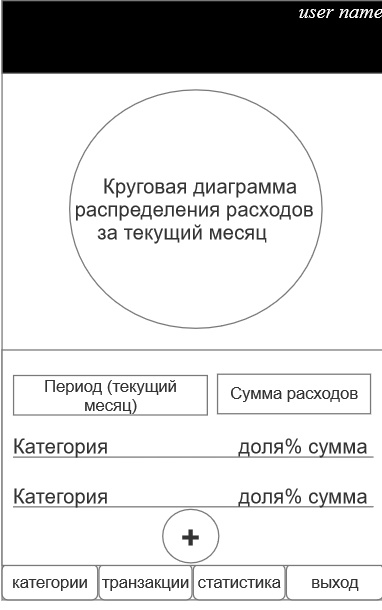


Рисунок 11 – Макет страницы со статистикой для пользователя(расходы в текущем месяце)

Пользователь может «проскролить» страницу со статистикой, на ней появятся статистика за предыдущий месяц и рекомендации по сокращению расходов(Рисунки 12, 13).



Рисунок 12 – Макет страницы со статистикой за предыдущий месяц

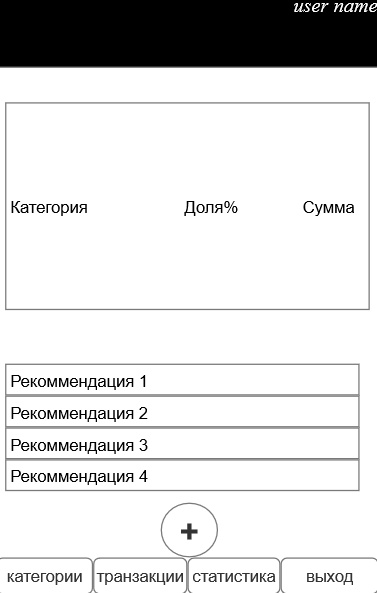


Рисунок 13 – Макет страницы с рекомендациями



Рисунок 14 – Макет страницы с общей статистикой по расходам пользователей за текущий месяц

«Проскролив» вниз пользователь увидит статистику за предыдущий месяц



Рисунок 15 – Макет страницы со статистикой по расходам пользователей за предыдущий месяц

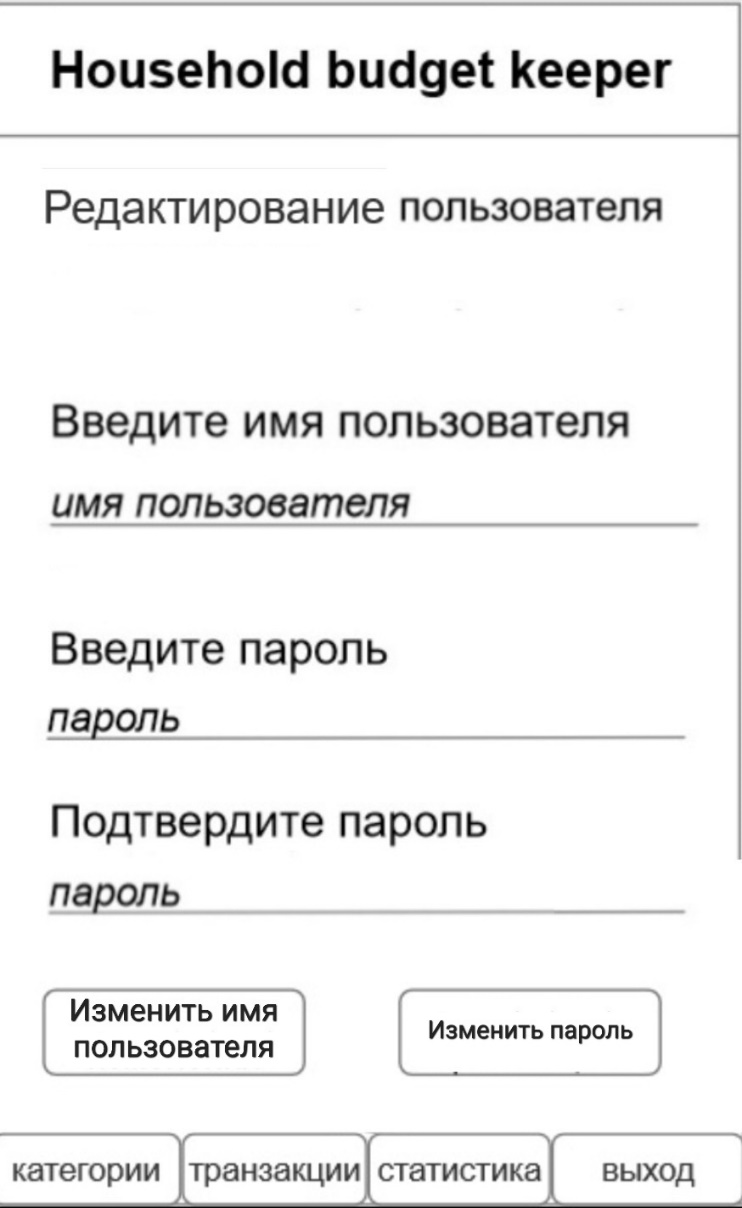


Рисунок 16 – Макет страницы редактирования пользователя

1. Анализ аналогов
   1. CoinKeeper

Функциональность:

* Кастомизация статистики и построение графиков и диаграмм по заданным критериям
* Интеграция с смс уведомлениями от банков и автоматическое создание записей о зачислениях
* Создание и редактирование категорий расходов
* Создание и редактирование транзакций
* Просмотр статистики по расходам пользователя за выбранный период
* Создание и редактирование счетов для пополнения баланса
* Просмотр расходов на отдельные категории



Рисунок 17 – Главный экран приложения CoinKeeper

Преимущества:

* Создание нескольких счетов для пополнения баланса
* Создание счетов и записей о расходах и пополнениях в разных валютах

Недостатки:

* Перегруженный интерфейс
* Отсутствие возможности кастомизировать статистику в бесплатной версии приложения
* Ограниченное количество счетов и категорий расходов в бесплатной версии приложения
  1. 1Money: расходы финансы бюджет

Функциональность:

* Создание и редактирование категорий расходов
* Создание и редактирование транзакций
* Просмотр статистики по расходам пользователя за выбранный период
* Создание и редактирование счетов для пополнения баланса
* Просмотр расходов на отдельные категории

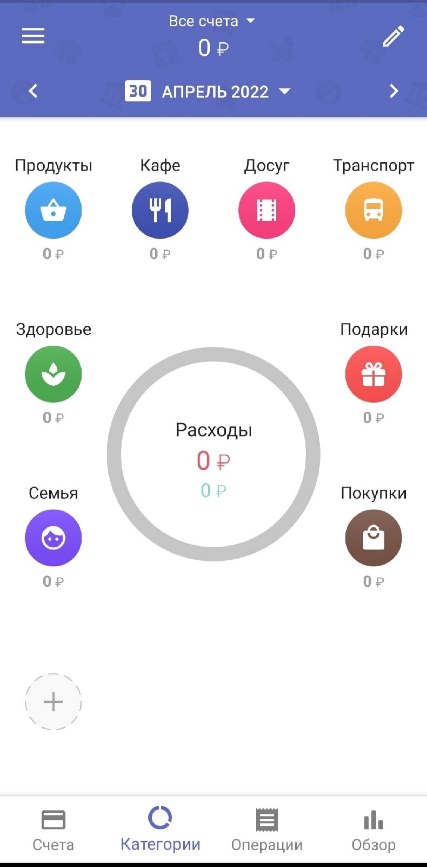


Рисунок 18 – Страница с категориями расходов приложения 1Money

Преимущества:

* Создание несколько счетов для пополнения баланса
* Просмотр статистики расходов за выбранный пользователем период времени

Недостатки:

* В бесплатной версии присутствует реклама, отвлекающая от ведения учёта расходов
* Возможность синхронизации на нескольких устройствах только при наличии платной подписки
* Отсутствие рекомендаций

1. Анализ задачи
   1. Средства реализации

Backend:

* Язык программирования Java 11
* Framework SpringBoot 2.5
* База данных PostgreSQL
* Система сборки Gradle
* IDE для разработки Intellij idea

Android app:

* Язык программирования Java 11
* Средство разработки Android SDK
* Система сборки Gradle
* IDE для разработки Android studio
  1. Входные-выходные данные(IDEF0)

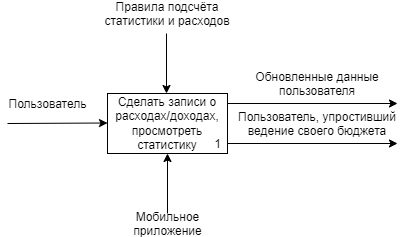


Рисунок 19 – IDEF0

* 1. Схема БД

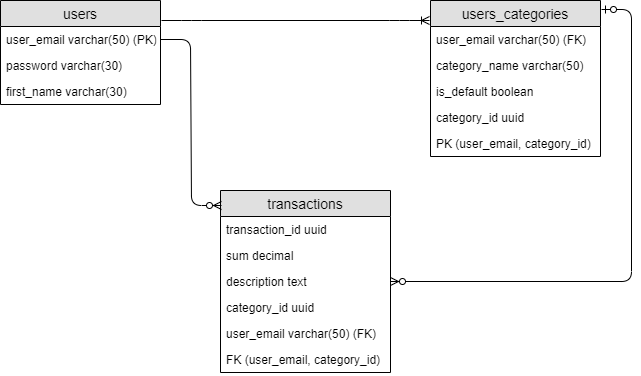


Рисунок 20 – ER диаграмма

users – таблица с пользователями, transactions – таблица с пополнениями баланса и с расходами, users\_categories – таблица с категориями пользователей.

* 1. Диаграммы классов

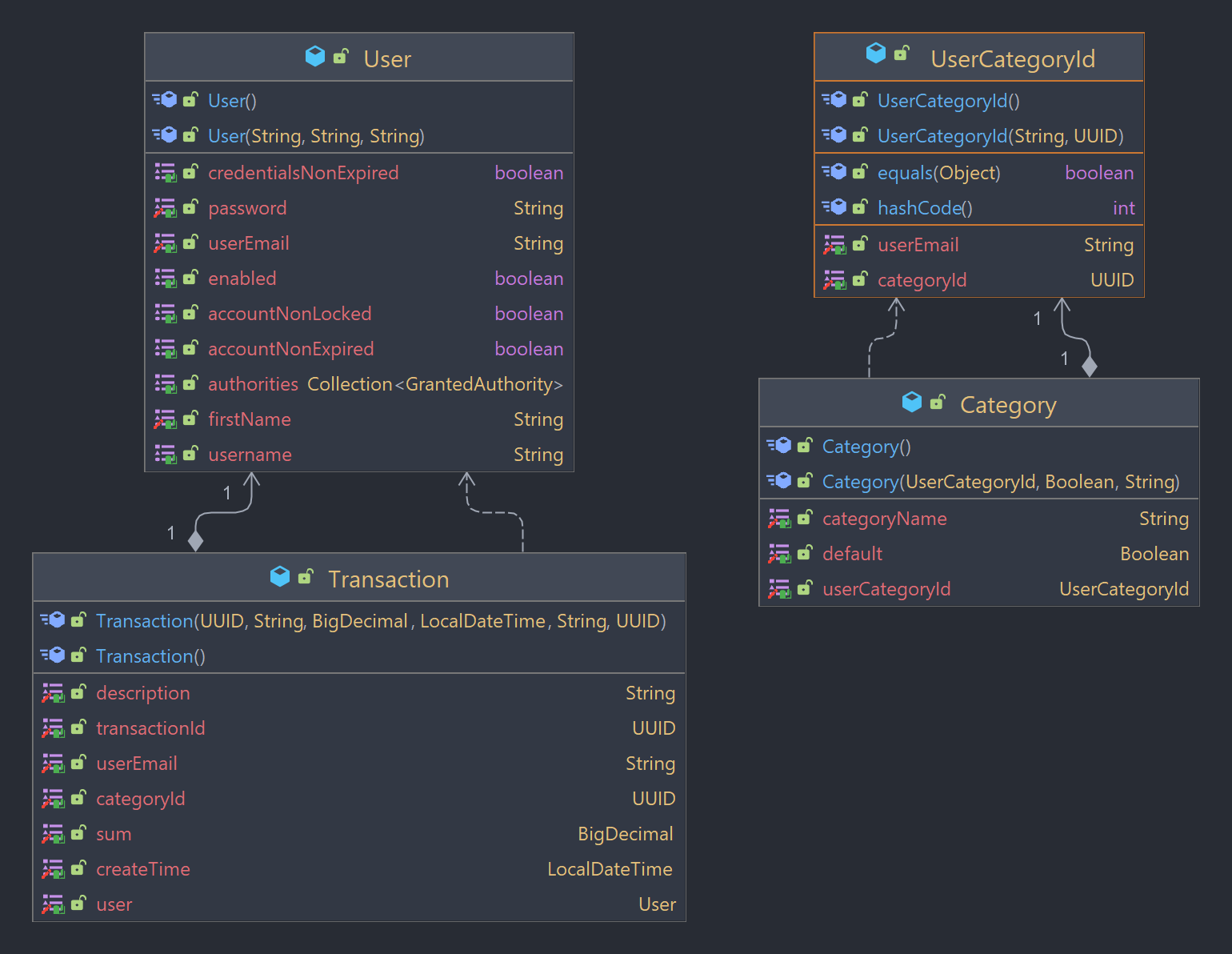


Рисунок 21 – Диаграмма классов сущностей БД

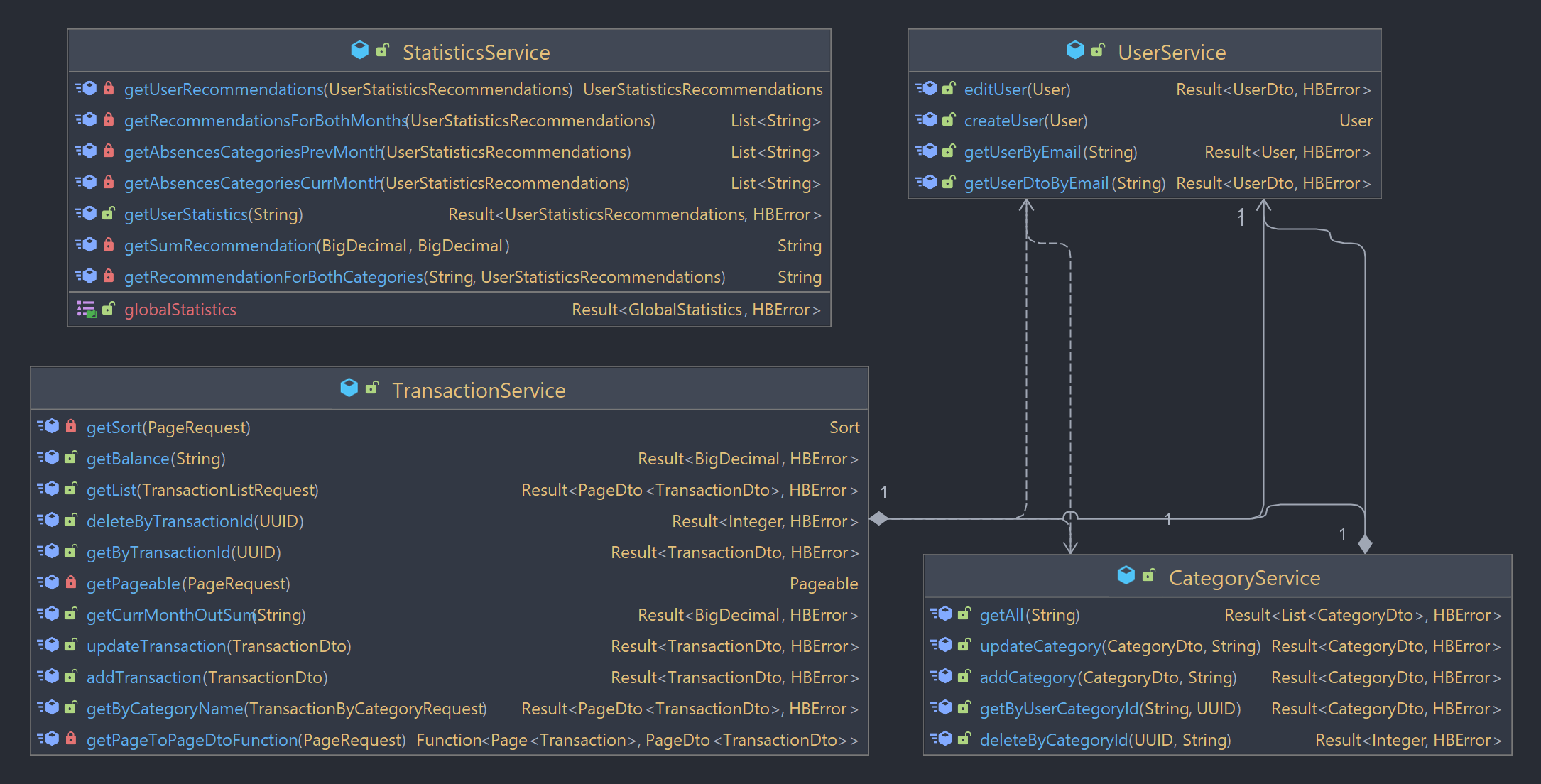


Рисунок 22 – Диаграмма классов сервисов

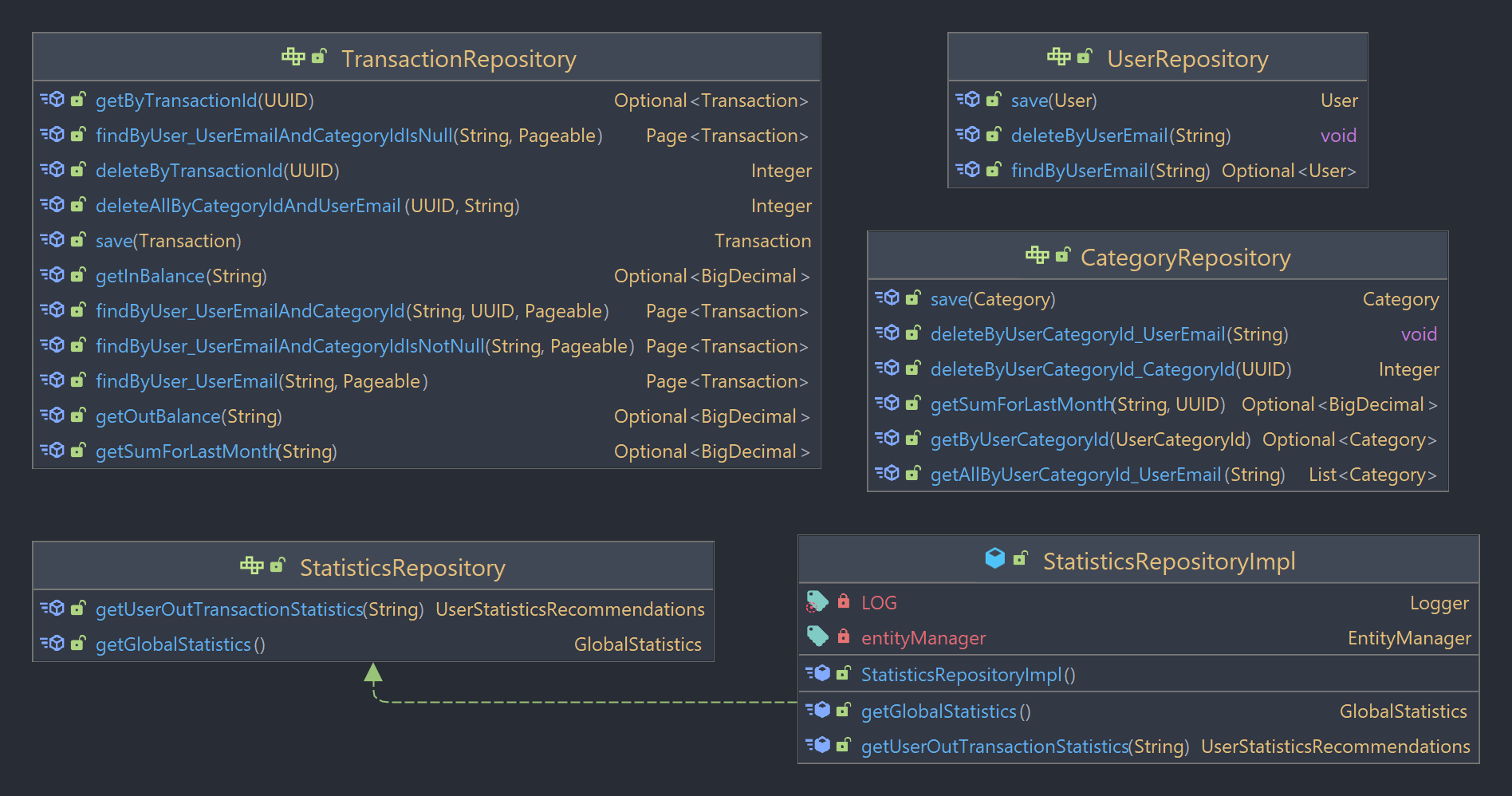


Рисунок 23 – Диаграмма классов репозиториев

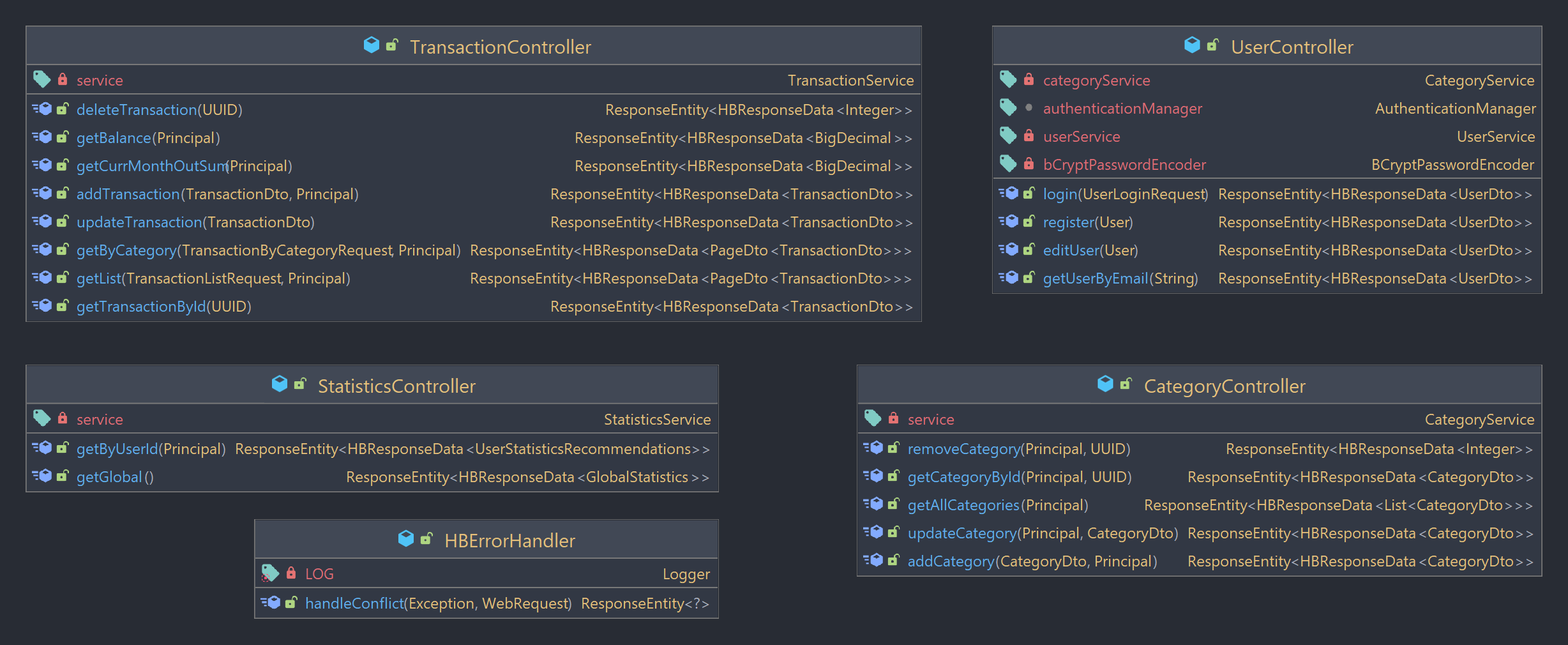


Рисунок 24 – Диаграмма классов контроллеров

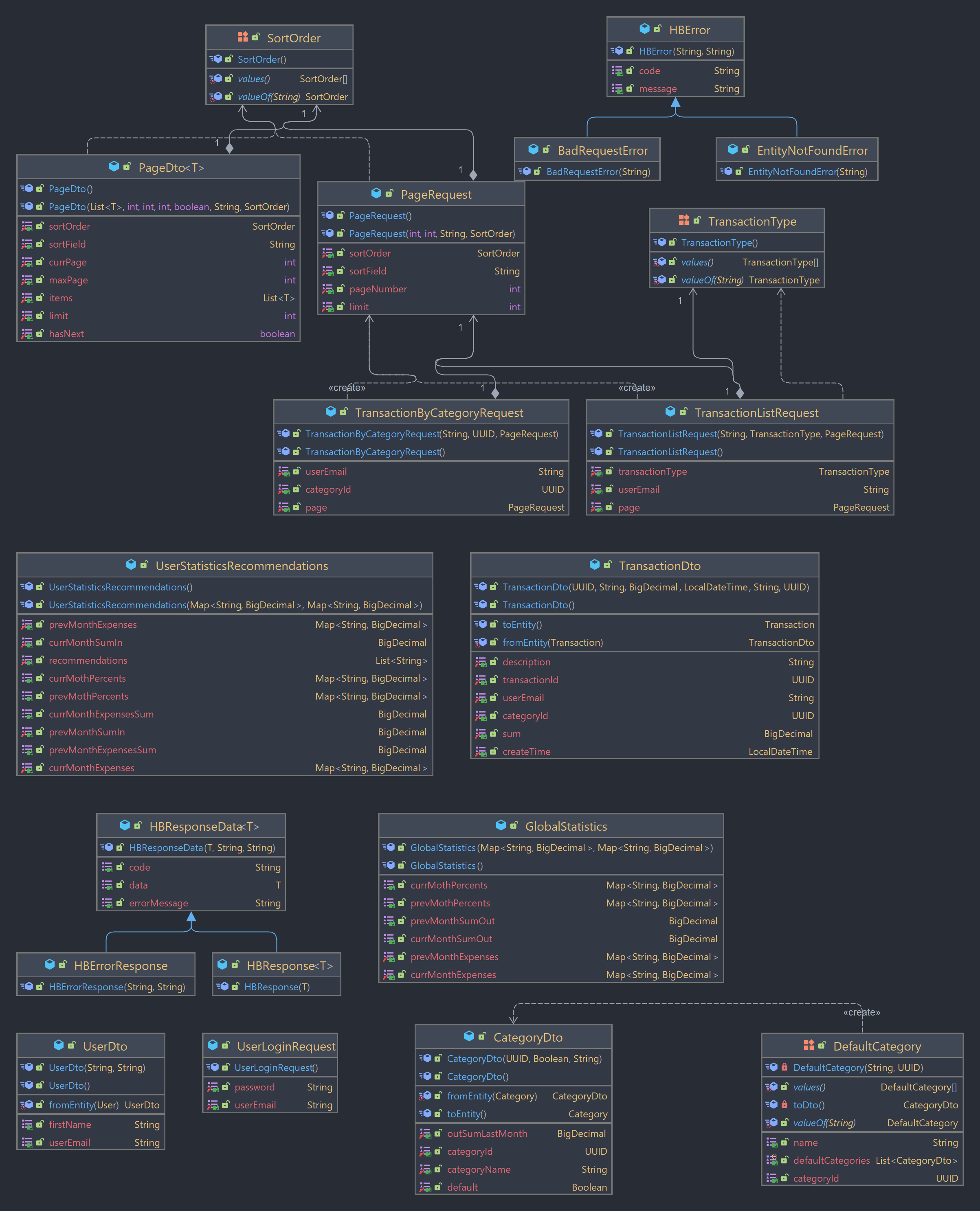


Рисунок 25 – Диаграмма классов Dto

* 1. Диаграммы вариантов использования



Рисунок 26 – Диаграмма вариантов использования гостя

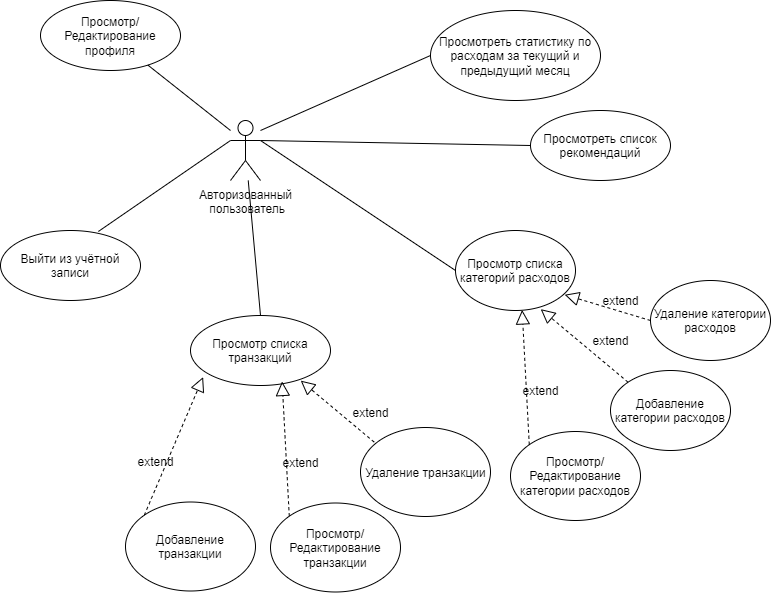


Рисунок 27 – Диаграмма вариантов использования авторизованного пользователя

* 1. Диаграмма потоков данных

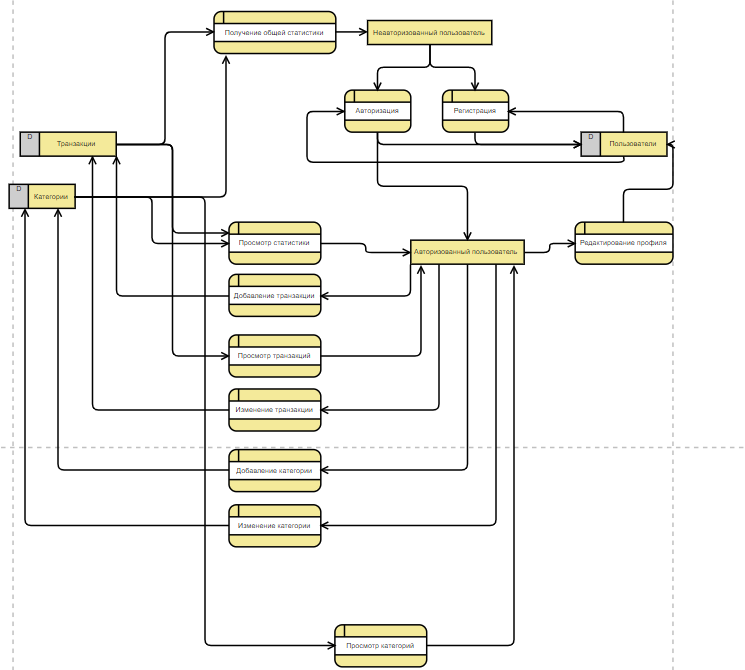


Рисунок 28 – Диаграмма потоков данных

* 1. Диаграмма состояний

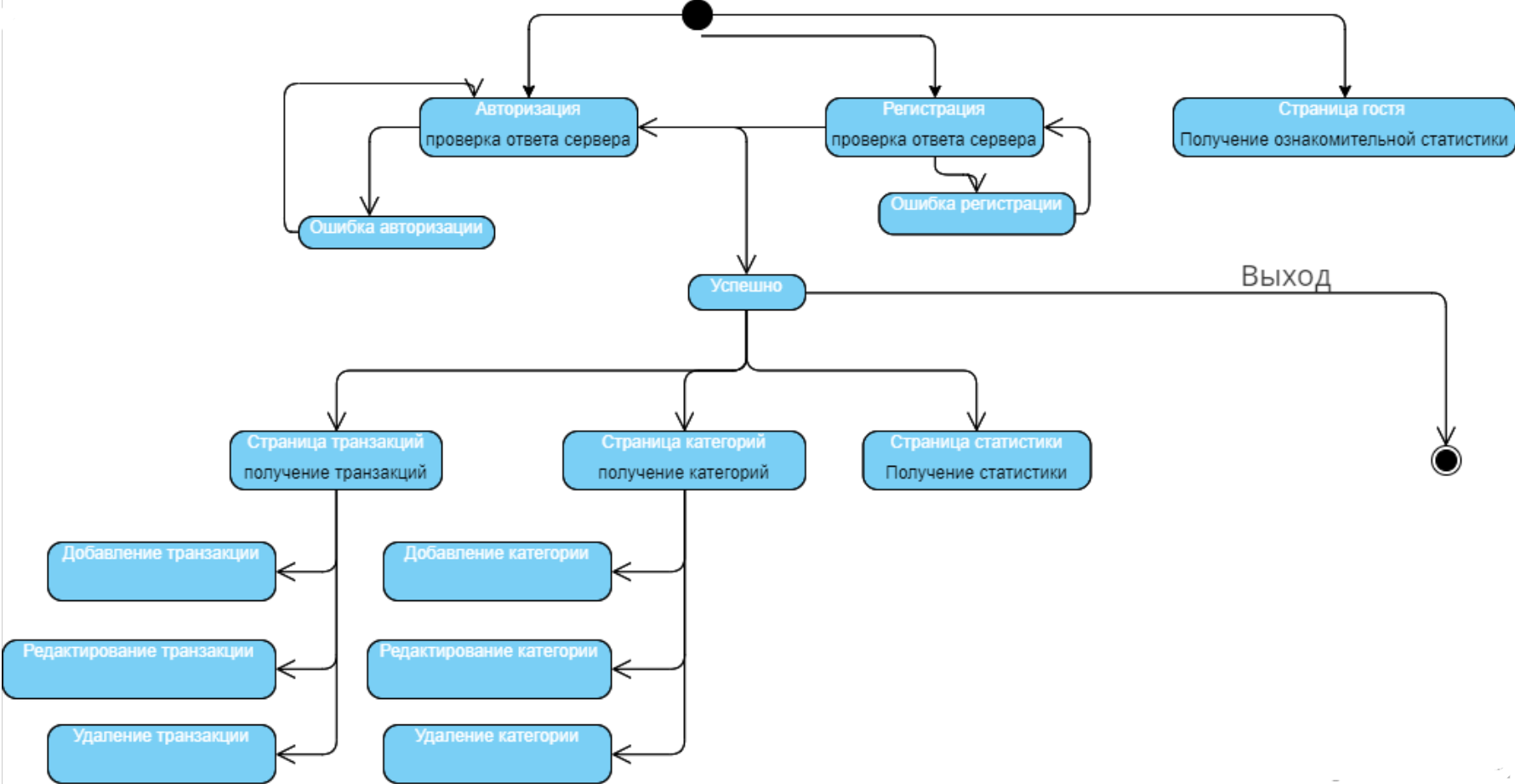


Рисунок 29 – Диаграмма состояний

* 1. Структурная схема приложения

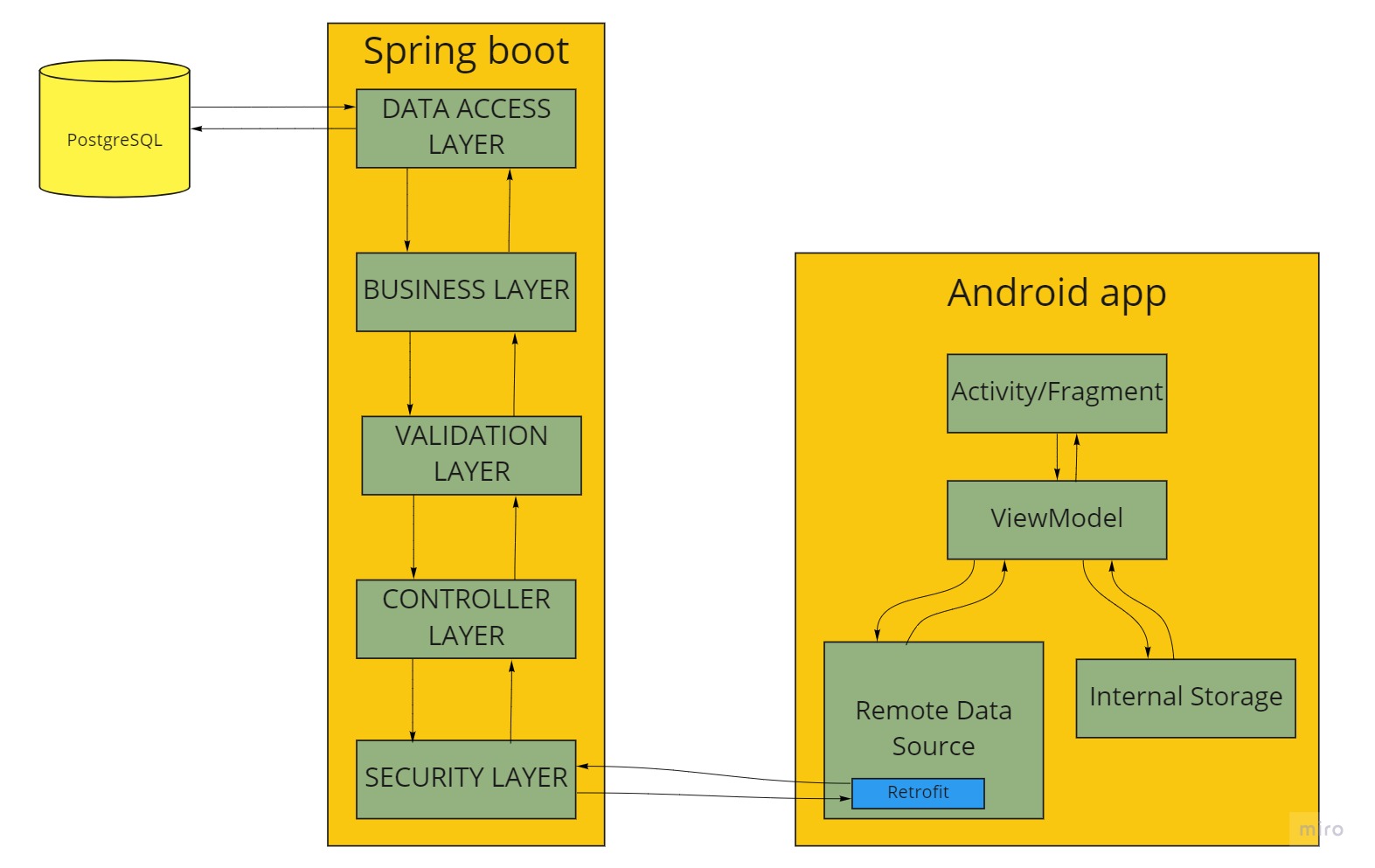


Рисунок 30 – Структурная схема приложения

* 1. Диаграммы последовательностей

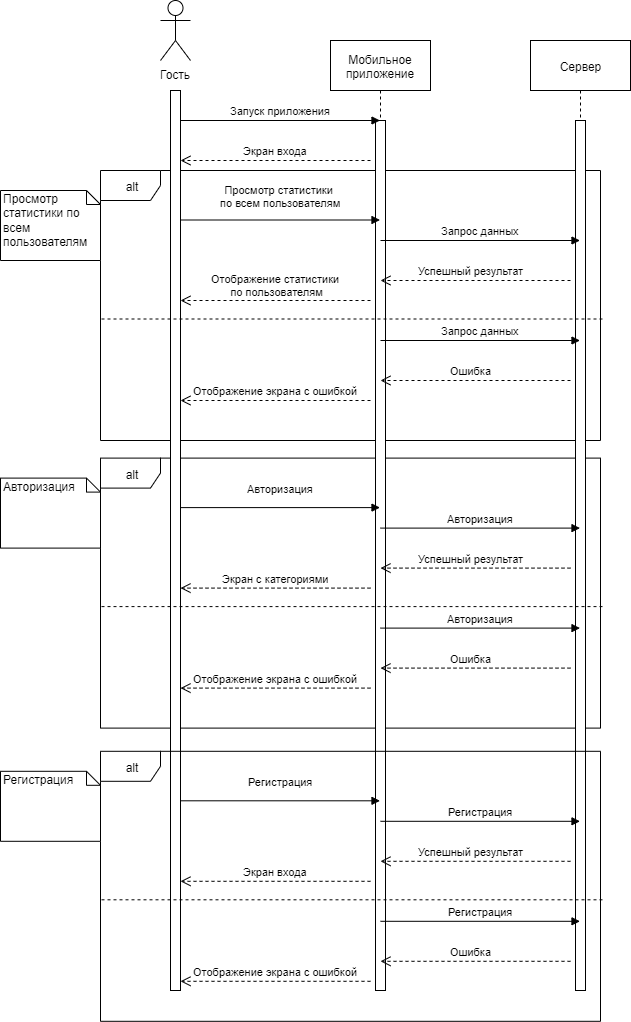


Рисунок 31 – Диаграмма последовательностей гостя

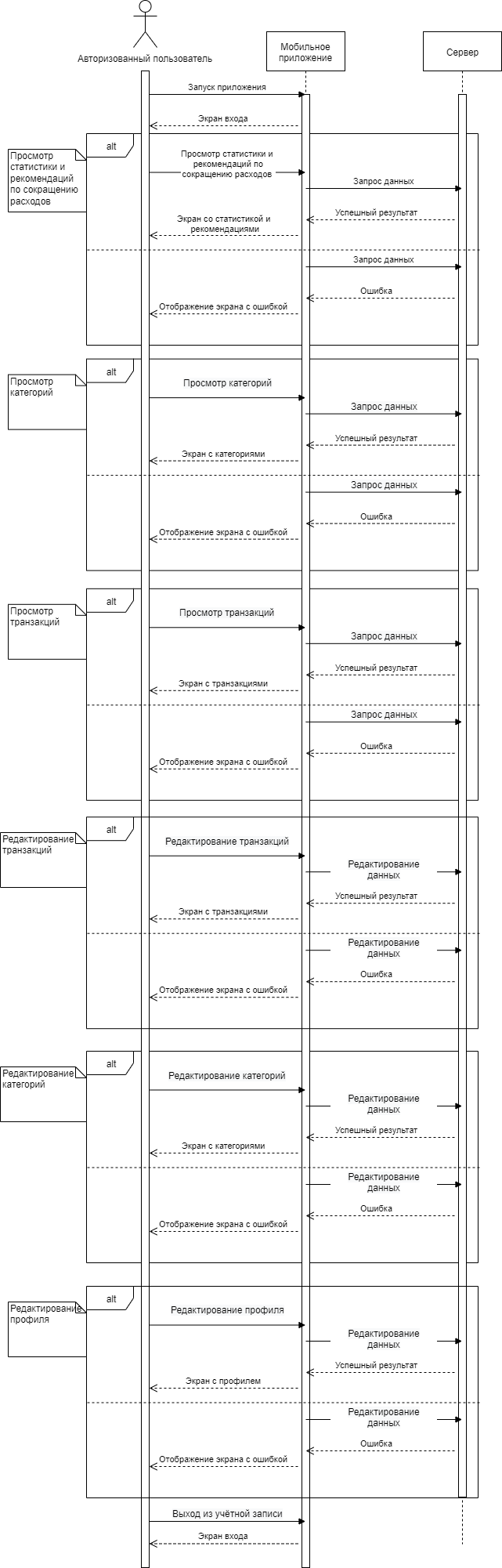


Рисунок 32 – Диаграмма последовательностей авторизованного пользователя

1. Сценарии воронок

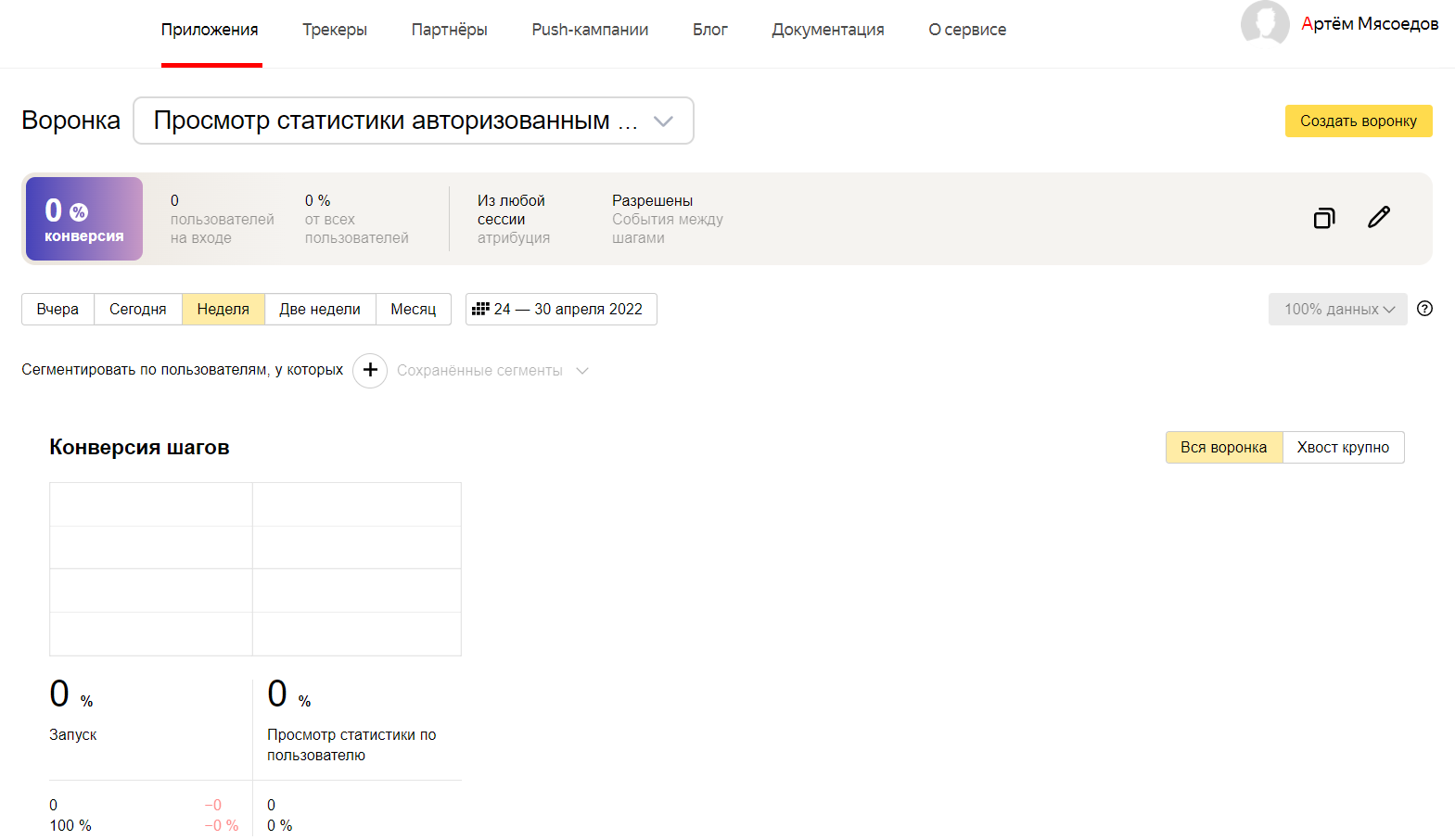


Рисунок 33 – Воронка просмотра статистики по пользователю

Запуск приложения – Переход на страницу со статистикой авторизованным пользователем

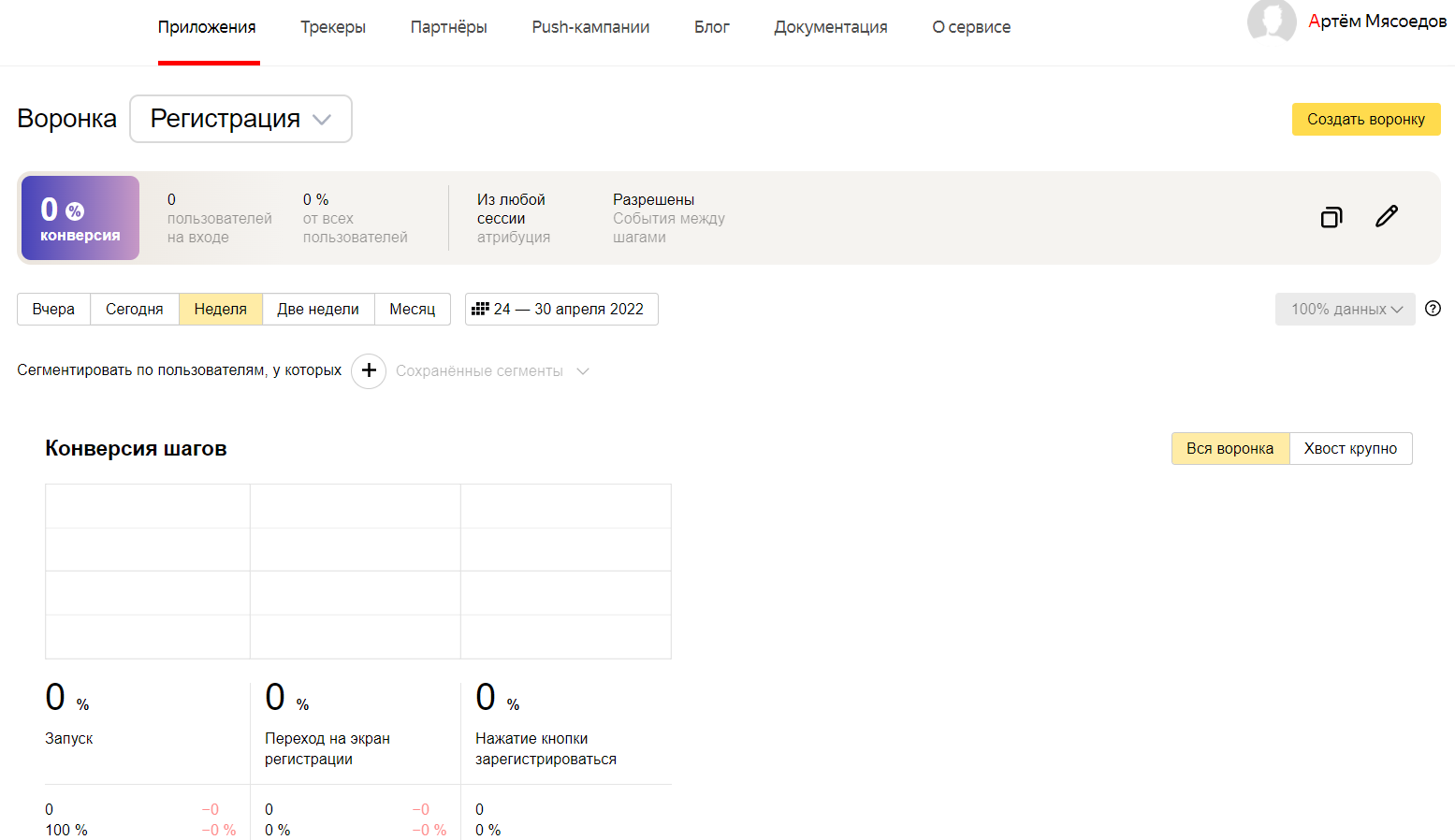


Рисунок 34 – Воронка регистрации

Запуск приложения – Переход на страницу регистрации – Нажатие кнопки «Зарегистрироваться»

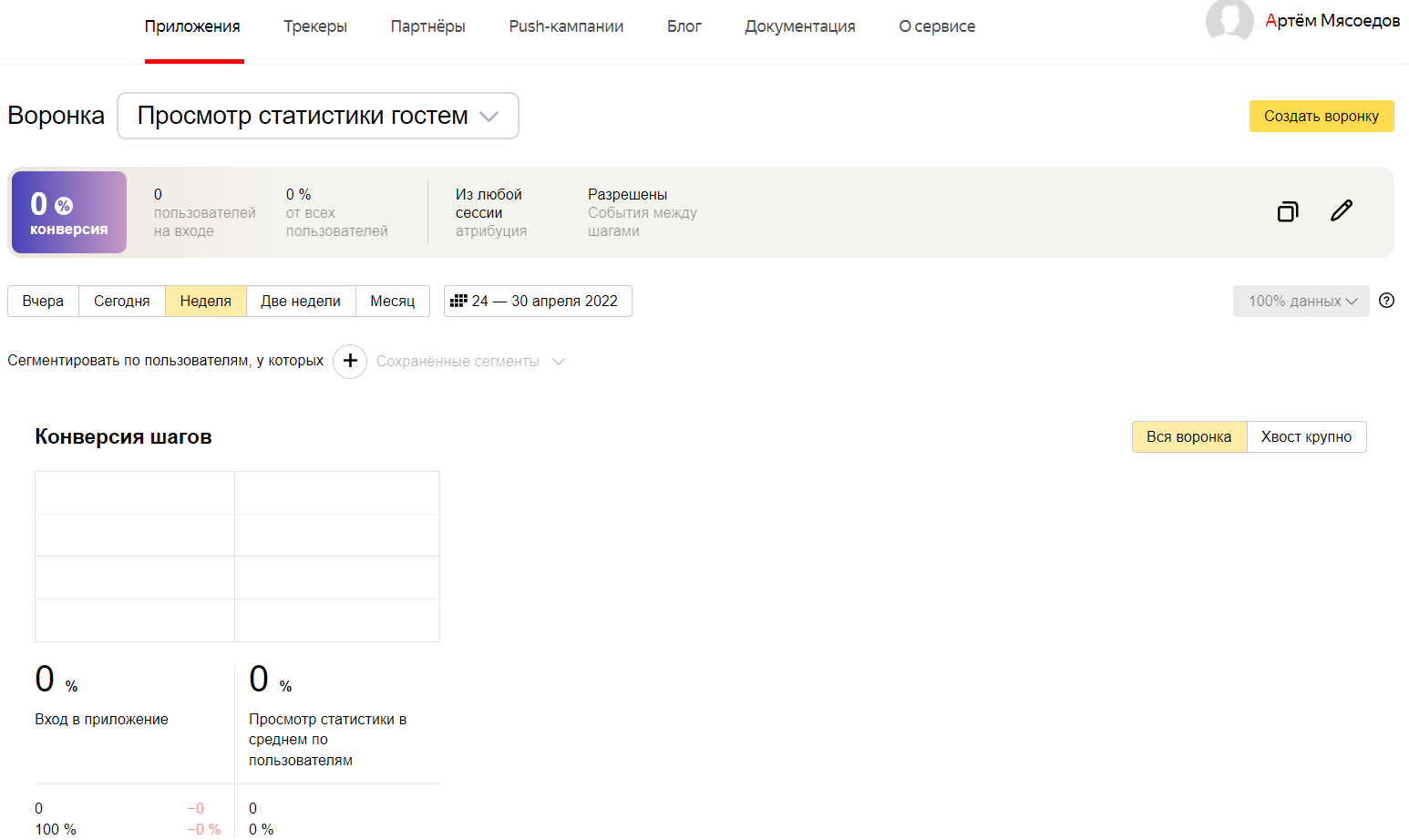


Рисунок 35 – Воронка просмотра статистики гостем

Запуск – Переход на страницу со статистикой по всем пользователям в среднем

1. Список планируемых работ

* Составление ТЗ
* Разработка сценариев работы системы
* Разработка схемы архитектуры приложения
* Разработка модуля хранения данных
* Разработка серверной части
* Разработка клиентской части
* Составление курсового проекта:
  + Анализ предметной области
  + Постановка задачи
  + Анализ аналогов
  + Составление UML диаграмм
* Тестирование приложения
* Доработка и устранение недостатков

1. Календарный план

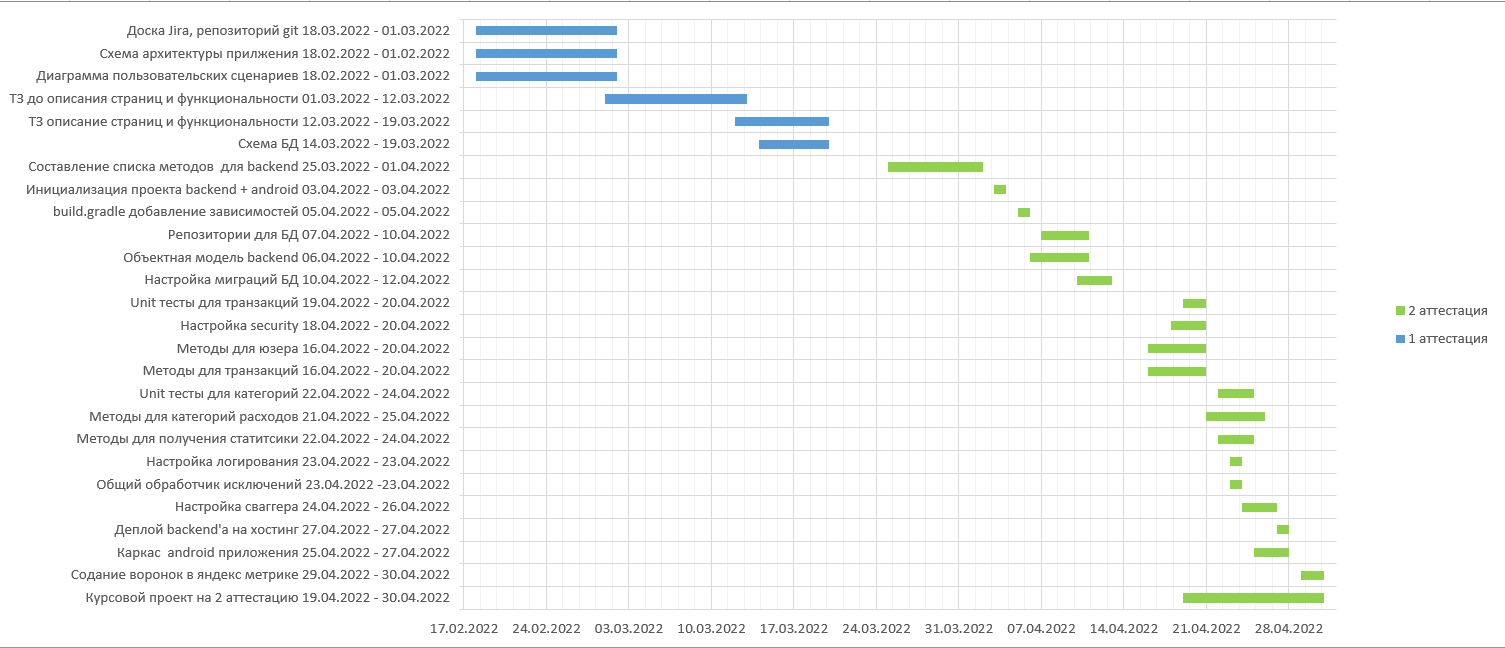


Рисунок 36 – Календарный план