

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук  
Кафедра информационных технологий управления

**Курсовой проект**

по дисциплине: «Технологии программирования»

на тему: «Разработка мобильного приложения «Сервис для управления  
финансами и бюджетом семьи с категориями расходов, отчетами и  
планированием бюджета»»

Зав. Кафедрой \_\_\_\_\_ М.Г. Матвеев, д.т.н., профессор

Обучающийся \_\_\_\_\_ Д.А. Жилиев

Обучающийся \_\_\_\_\_ А.А. Соколов

Обучающийся \_\_\_\_\_ А.А. Волобуева

Руководитель \_\_\_\_\_ В.А. Ушаков, преподаватель

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

Воронеж 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
ГЛОССАРИЙ .....	4
1 Постановка задачи.....	6
1.1 Цели создания системы.....	6
1.2 Функциональные требования к разрабатываемой системе .....	6
1.3 Задачи системы.....	7
2 Анализ предметной области .....	8
2.1 Обзор аналогов .....	8
2.1.1 PocketGuard .....	8
2.1.2 Дзен-мани.....	9
2.1.3 YNAB .....	10
2.2 Моделирование системы.....	11
2.2.1 Use Case (диаграмма прецедентов).....	11
2.2.2 Sequence (Диаграмма последовательностей) .....	13
2.2.3 State Chart (Диаграмма состояний) .....	17
2.2.4 Deployment диаграмма (развертывания) .....	20
2.2.5 ER-диаграмма .....	20
3 Реализация .....	21
3.1 Реализация клиентской части приложения.....	21
3.2 Реализация серверной части приложения .....	26
ЗАКЛЮЧЕНИЕ .....	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	29

## **ВВЕДЕНИЕ**

Грамотное планирование бюджета – неотъемлемая и очень важная часть жизни каждого взрослого человека. Особенно в современном мире, в условиях существования множества счетов в разных банках, проблема централизованного учёта всех трат становится всё более актуальной. И если даже одному человеку сложно контролировать свои доходы и расходы, еще сложнее распланировать семейный бюджет.

В рамках данной курсовой работы предлагается разработка мобильного приложения для контроля семейного бюджета с возможностью планирования и аналитики как по личному бюджету, так и по семейному счету. Целью данного проекта является создание удобного и функционального инструмента, который поможет семьям четко и структурированно управлять своими финансами, распределять и планировать траты по категориям, анализировать, куда уходит большая часть бюджета, оптимизировать свои расходы и достигать финансовых целей.

Одной из ключевых особенностей разрабатываемого приложения будет возможность совместного анализа и планирования семейных расходов, что поможет семьям планировать, отслеживать и анализировать свои финансы в удобном и эффективном формате, способствуя более осознанному управлению семейными финансами и созданию финансовой гармонии в семье.

## ГЛОССАРИЙ

Транзакция – любая денежная операция над бюджетом пользователя, которая вводится им в приложение вручную;

Операция – трата или приход денежных средств пользователя, которые он считает нужным записать в приложение (записать транзакцию).

Счёт – это раздел хранения информации о сумме денег, к которому применяются транзакции. Примером может служить счет в банке, счет для учета наличных денег, счет для учета средств на брокерском счету и так далее (в зависимости от желания пользователя);

Личный счёт – счёт, заведенный текущим пользователем, к которому есть доступ только у одного человека (текущего пользователя, создателя счёта);

Семейный счёт – счёт, заведенный одним пользователем, доступ к просмотру и редактированию транзакций по которому создатель счёта может предоставить другим пользователям;

Категоризация – разделение транзакций по группам (категориям) по принципу схожести объекта трат/зачислений для удобства анализа и планирования бюджета.

Редактирование транзакций/счетов/категорий – изменение поля с названием или суммой в случае неправильного заполнения пользователем (человеческий фактор при вводе пользователем в приложение).

Flutter — фреймворк для создания мобильных приложений под Android и iOS, веб-приложений, а также настольных приложений под Windows, macOS и Linux с использованием языка программирования Dart;

UUID (universally unique IDentifier) — это 128-битное число, которое в разработке ПО используется в качестве уникального идентификатора элементов;

JSON Web Token (JWT) — это открытый стандарт (RFC 7519) для создания токенов доступа, основанный на формате JSON. Используется для передачи данных для аутентификации в клиент-серверных приложениях;

FastAPI - это фреймворк для создания API на основе стандартных подсказок по типу Python;

SQLAlchemy — это программное обеспечение для работы с базами данных при помощи языка SQL. Оно реализует технологию программирования ORM (Object-Relational Mapping), которая связывает базы данных с концепциями объектно-ориентированных языков программирования.

## **1 Постановка задачи**

### **1.1 Цели создания системы**

Целями создания приложения являются:

- Реализация системы, которая позволит учитывать транзакции с разных счетов в одном месте;
- Возможность создания общего семейного счёта с подключением к нему пользователей (с возможностью подключения личных счетов);
- Возможность анализа и планирования доходов и расходов на будущие периоды как по личным счетам, так и отдельно по семейному счёту.

### **1.2 Функциональные требования к разрабатываемой системе**

Для неавторизованных пользователей:

- Создание аккаунта;
- Создание, удаление, редактирование личного денежного счёта;
- Создание, удаление, редактирование транзакции;
- Категоризация транзакций.

Для авторизованных пользователей:

- Авторизация;
- Восстановление пароля;

- Редактирование, удаление аккаунта;
- Создание, редактирование, удаление личного счёта;
- Создание, редактирование, удаление семейного счёта;
- Подключение других пользователей к семейному счёту, отключение от него;
- Подключение личного счёта к семейному счёту, отключение от него;
- Создание, редактирование, удаление транзакции;
- Категоризация транзакций;
- Создание, редактирование, удаление новой категории;
- Построение графиков расходов за определенный период или по семейному счёту;
- Расчёт % расходов от общей суммы трат по категориям;
- Установление лимита трат на месяц по каждой категории;
- Расчёт потраченной и оставшейся суммы в рамках плана для каждой категории.

### **1.3 Задачи системы**

- создавать/редактировать/удалять транзакции;
- категоризировать транзакции;
- создавать/редактировать/удалять категории транзакций;

- создавать/удалять семейный счёт, подключать/удалять к нему других пользователей;
- подключать/удалять личный счёт к семейному счёту;
- просматривать графики расходов за день/месяц/год/по семейному счёту;
- просматривать, какая часть бюджета потрачена и какой % трат составляет каждая категория;
- задавать ограничение трат на месяц по каждой категории.

## **2 Анализ предметной области**

### **2.1 Обзор аналогов**

#### **2.1.1 PocketGuard**

PocketGuard - это приложение для составления бюджета, которое синхронизируется с банковскими и кредитными счетами и может автоматически обновлять общий доход и последние транзакции, которые упорядочиваются по мере их поступления.

Еще одно преимущество данного приложения – функция "В моем кармане". Функция рассчитывает доходы и расходы пользователя, чтобы мгновенно оценить, сколько денег он может безопасно потратить.

Также в PocketGuard можно получать персональные отчёты о своих расходах, составлять бюджеты и ставить цели по экономии.

Интерфейс приложения представлен на Рисунке 1.



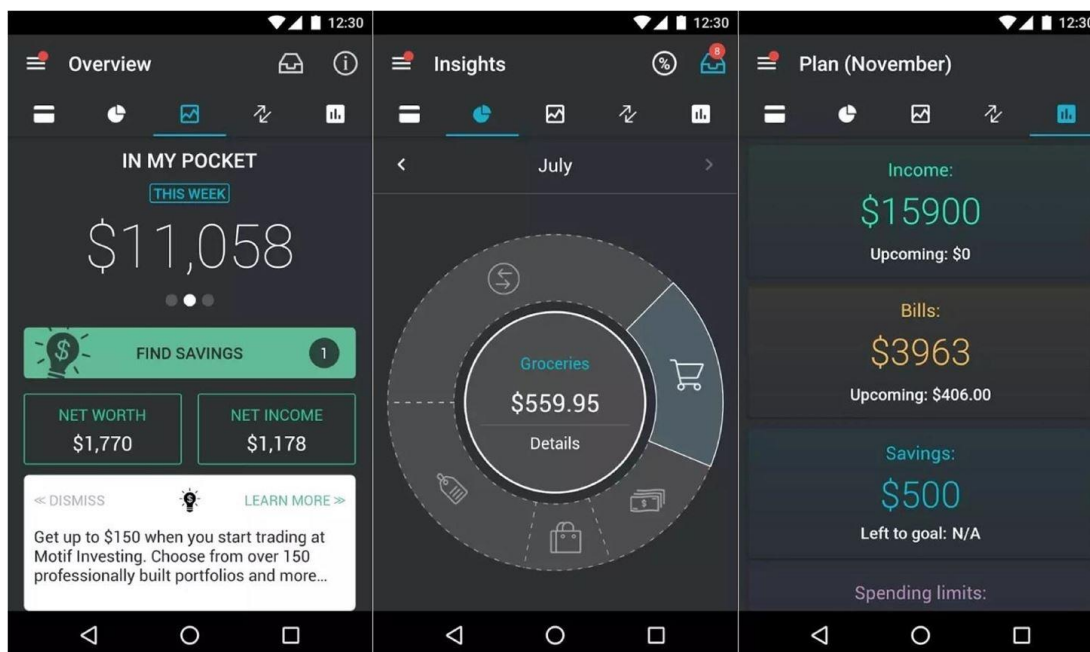


Рисунок 1 - Интерфейс PocketGuard

Однако данное приложение имеет ряд недостатков:

- Отсутствие возможности создания семейного счёта на нескольких пользователей;
- Ограниченная географическая поддержка: работа приложения может быть ограничена определенными регионами и банками.

### 2.1.2 Дзен-мани

Мобильное приложение «Дзен-мани: учет расходов» является продвинутым инструментом для учета личных финансов. Оно позволяет управлять категориями и счетами, записывать расходы, доходы и переводы между счетами в разных валютах, учитывать долги, устанавливать бюджеты и рассчитывать бюджет на день исходя из доступных средств на месяц.

«Дзен-мани» поддерживает автоматический импорт транзакций из онлайн-банков, распознавание СМС и сканирование чеков.

Интерфейс приложения представлен на Рисунке 2.



Рисунок 2 - Интерфейс «Дзен-мани»

Недостатки данного приложения:

- Отсутствие напоминаний об операциях;
- Отсутствие возможности создания семейного счёта на нескольких пользователей;
- Необходимость регистрации.

### 2.1.3 YNAB

YNAB (You Need a Budget) - это популярное приложение для учета личных финансов и составления своего бюджета.

YNAB подключается к банкам и кредитным картам пользователи, автоматически импортируя информацию о счетах и транзакциях. Эти данные также синхронизируются на всех устройствах, на которых установлено приложение.

Одним из его главных преимуществ является возможность поделиться своей подпиской до пяти человек, что позволяет всей семье участвовать в составлении бюджета.

Также есть возможность создавать отчеты и диаграммы, в которых описываются тенденции расходов и доходов с течением времени.

Интерфейс приложения представлен на Рисунке 3.

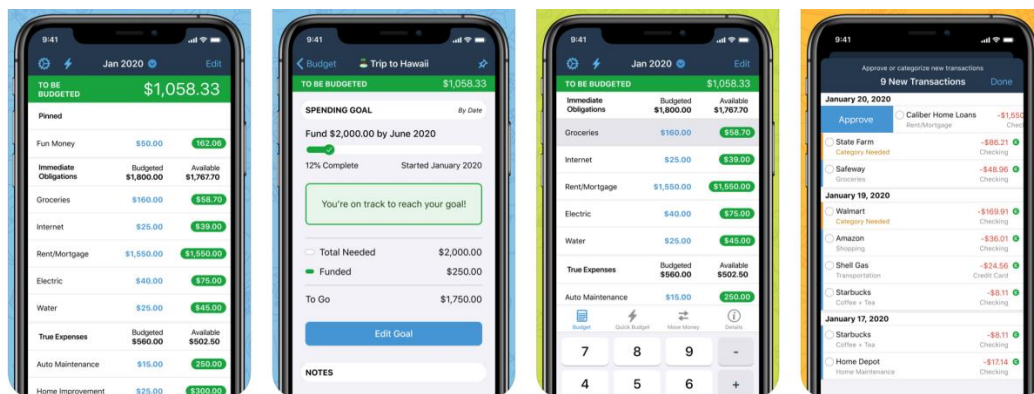


Рисунок 3 - Интерфейс YNAB

Недостатки приложения YNAB:

- Неумение работать с кредитами как с видом долга;
- Высокая цена подписки (выше, чем у многих приложений для составления бюджета) и отсутствие возможности отмены подписки.

## 2.2 Моделирование системы

### 2.2.1 Use Case (диаграмма прецедентов)

Для неавторизованного пользователя:



Рисунок 4 - Use Case диаграмма для неавторизованного пользователя

Для авторизованного пользователя:

Часть 1:

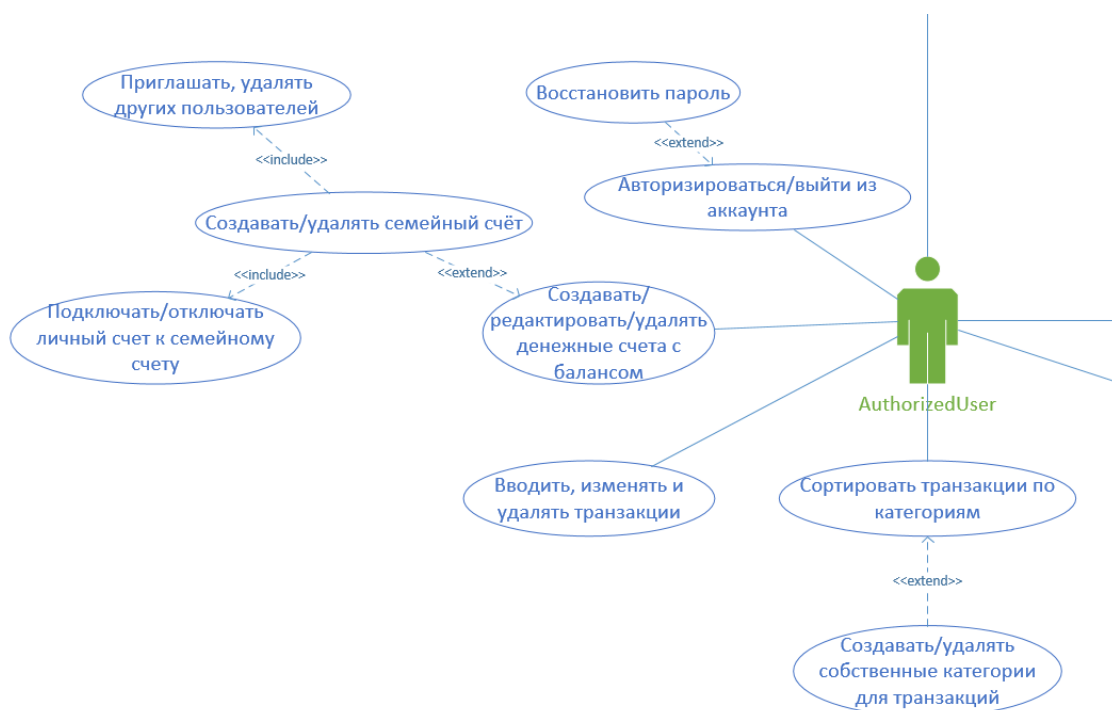


Рисунок 5 - Use Case диаграмма для авторизованного пользователя.

Часть 1

Часть 2:

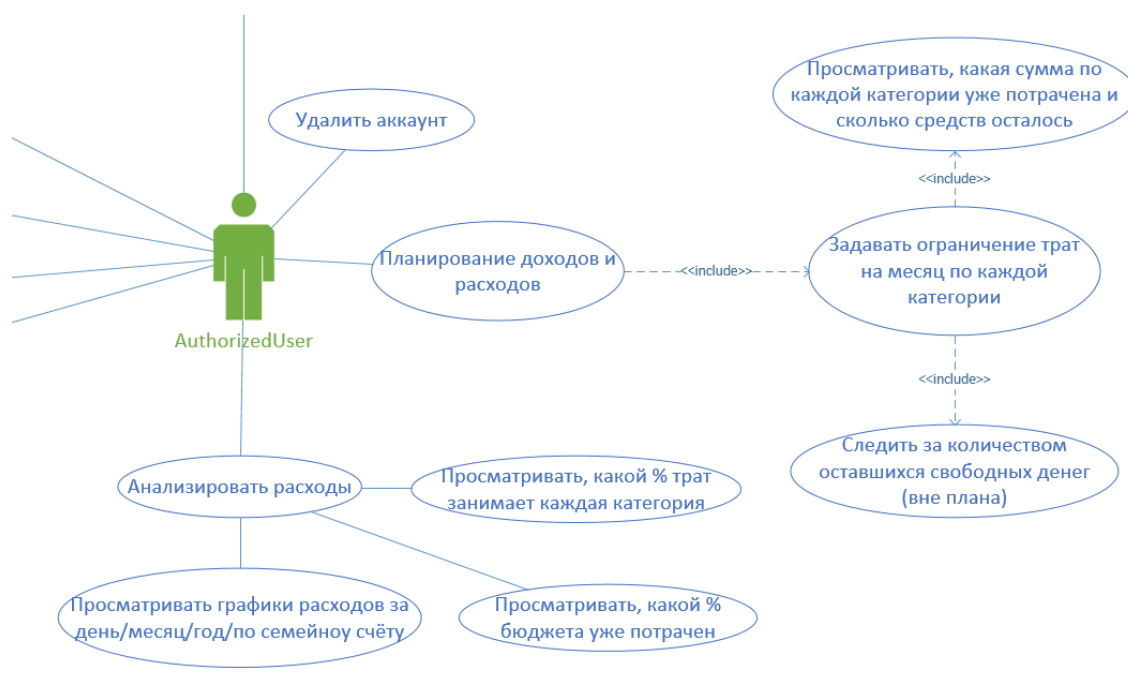


Рисунок 6 - Use Case диаграмма для авторизованного пользователя.

Часть 2

### 2.2.2 Sequence (Диаграмма последовательностей)

Для неавторизованного пользователя:

Часть 1:

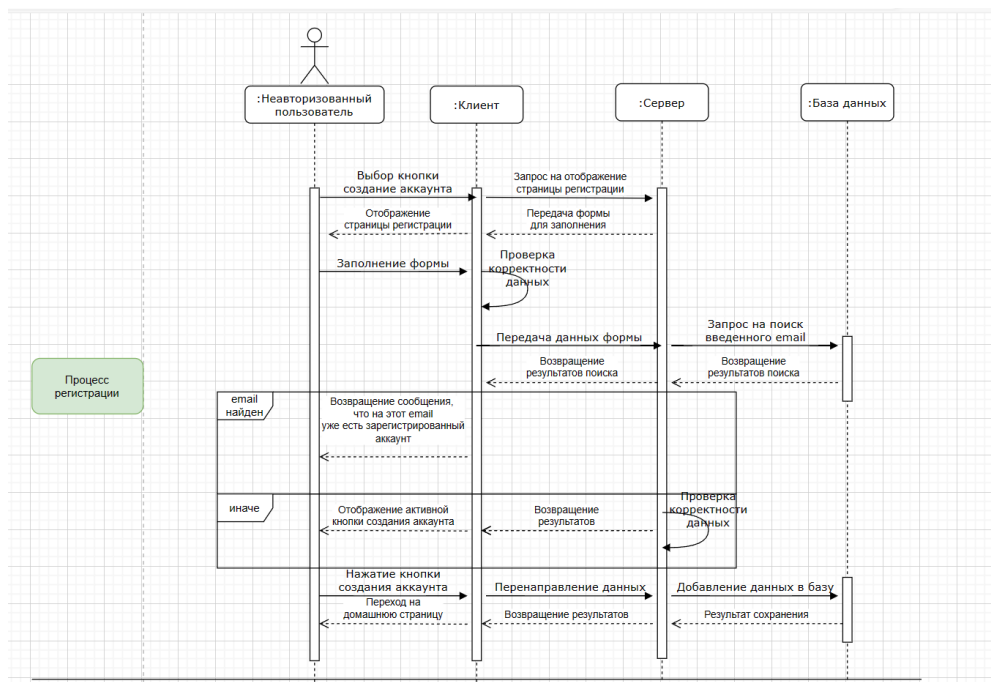


Рисунок 7 - Sequence диаграмма для неавторизованного пользователя.  
Часть 1

Часть 2:

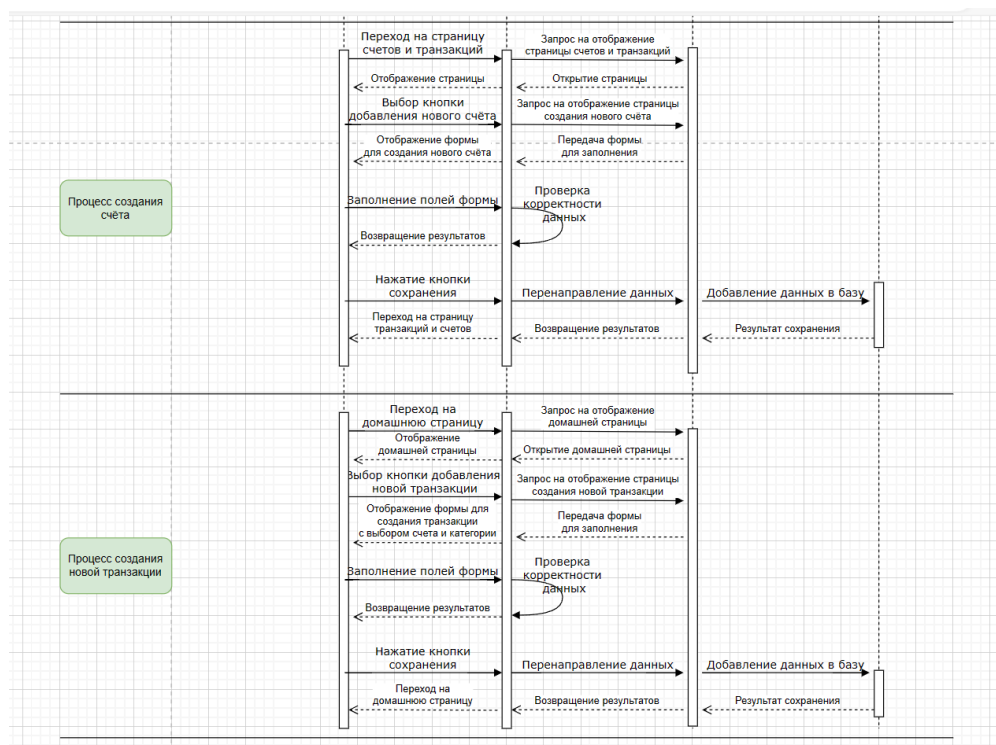


Рисунок 8 - Sequence диаграмма для неавторизованного пользователя.  
Часть 2

Для авторизованного пользователя:

Часть 1:

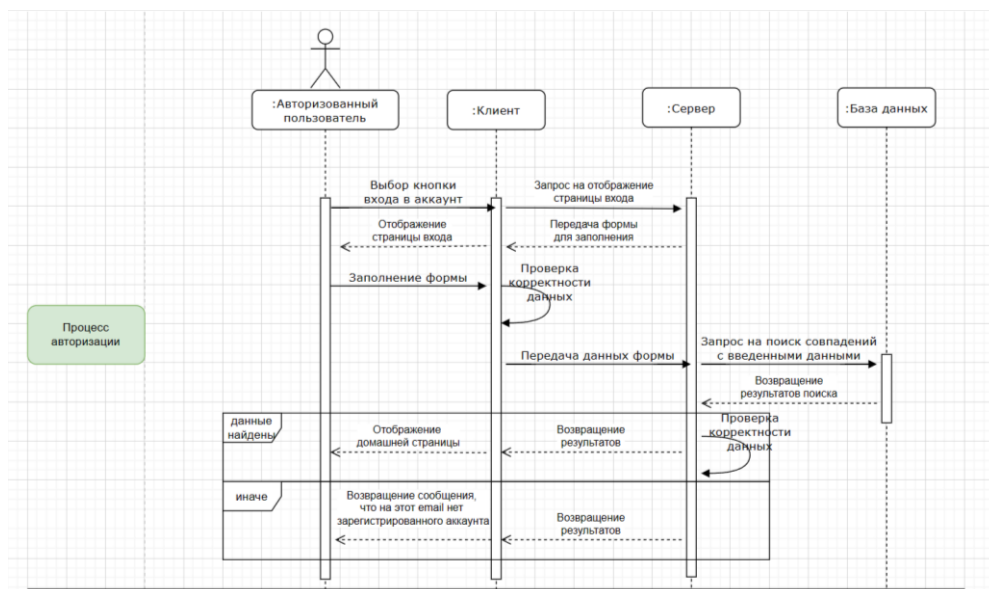


Рисунок 9 - Sequence диаграмма для авторизованного пользователя.

Часть 1

Часть 2:

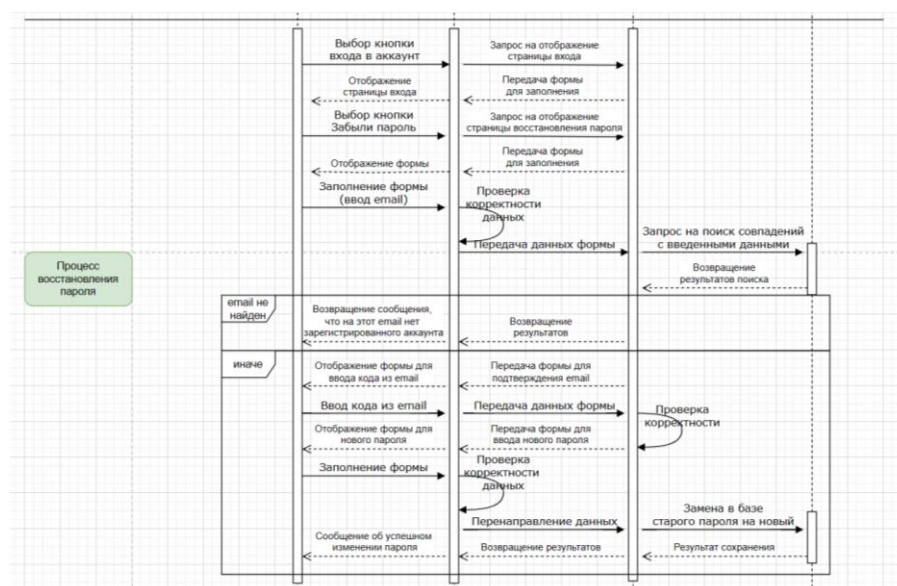


Рисунок 10 - Sequence диаграмма для авторизованного пользователя.

Часть 2



### Часть 3:

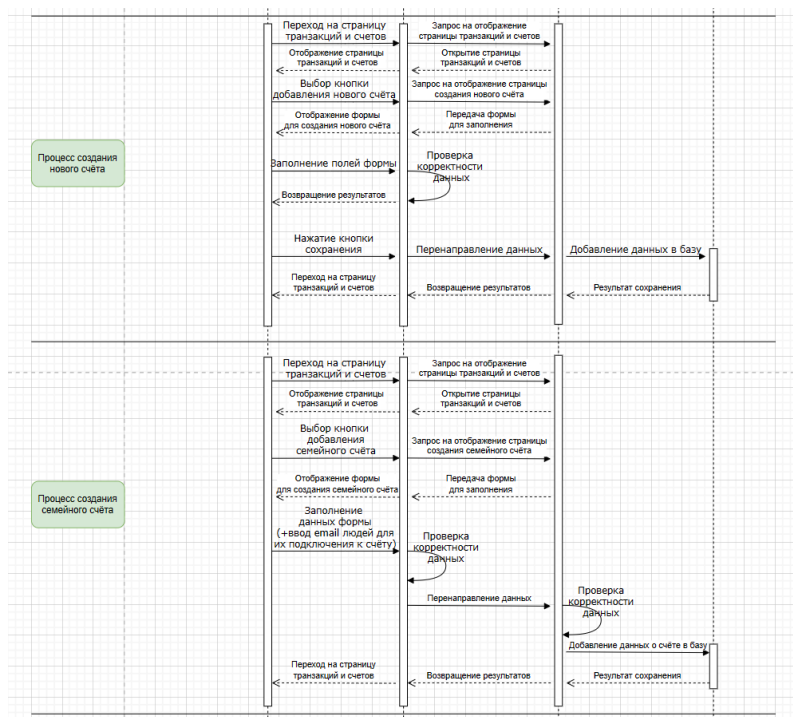


Рисунок 11 - Sequence диаграмма для авторизованного пользователя.

### Часть 3

### Часть 4:

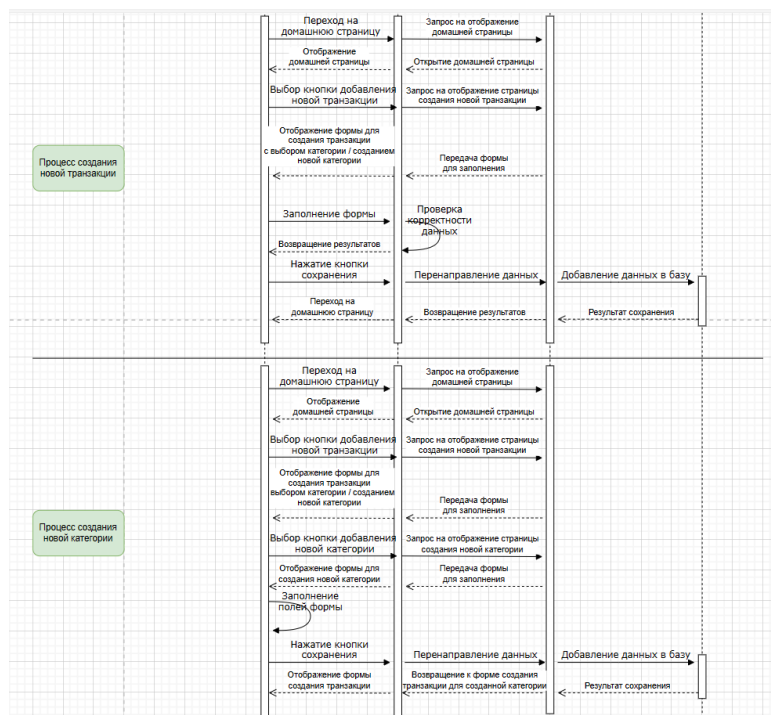


Рисунок 12 - Sequence диаграмма для авторизованного пользователя.

### Часть 4



## Часть 5:

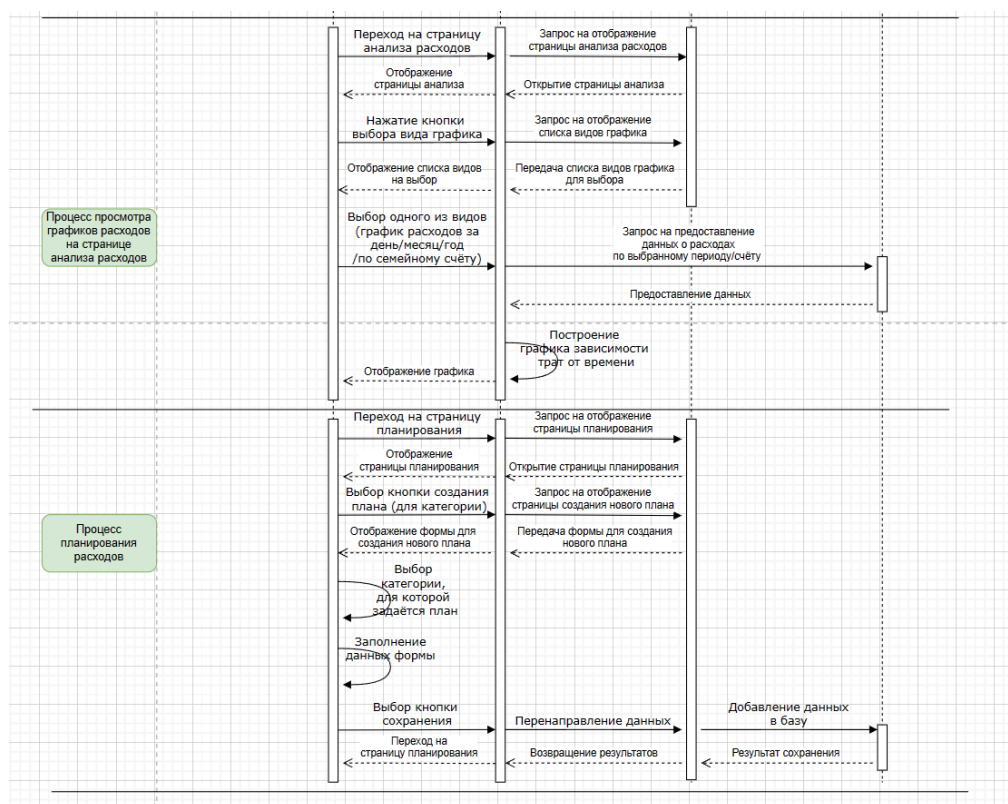


Рисунок 13 - Sequence диаграмма для авторизованного пользователя.

## Часть 5

### 2.2.3 State Chart (Диаграмма состояний)

Для неавторизованного пользователя:

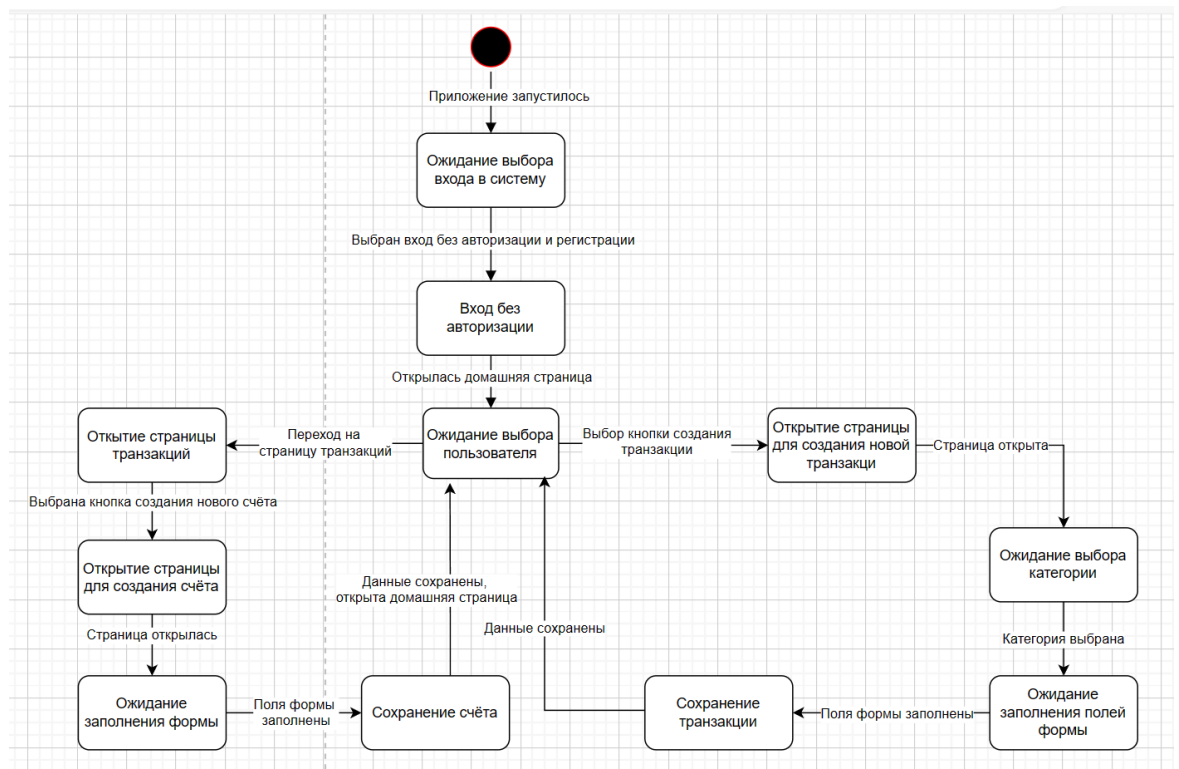


Рисунок 14 - State Chart диаграмма для неавторизованного пользователя

Для авторизованного пользователя:

Часть 1:

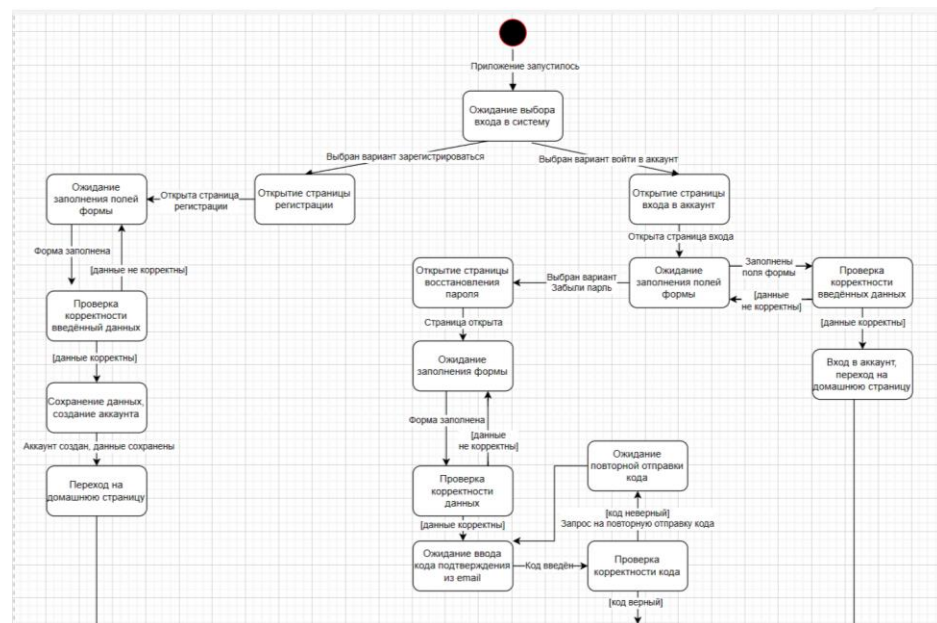


Рисунок 15 - State Chart диаграмма для авторизованного пользователя.

Часть 1

## Часть 2:

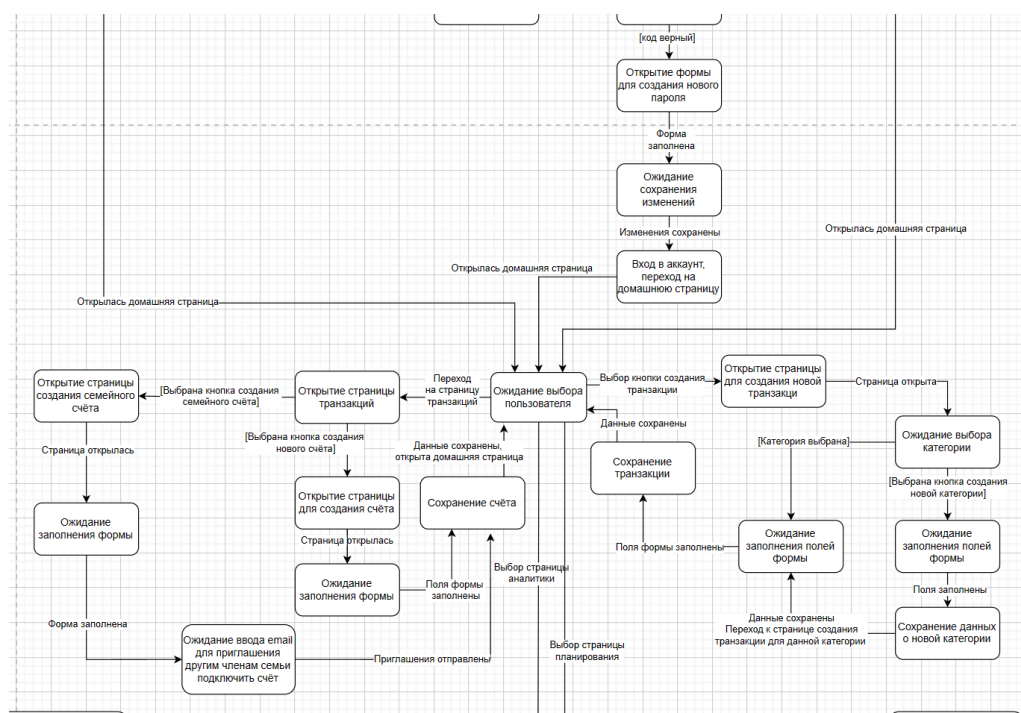


Рисунок 16 - State Chart диаграмма для авторизованного пользователя.

## Часть 2

### Часть 3:

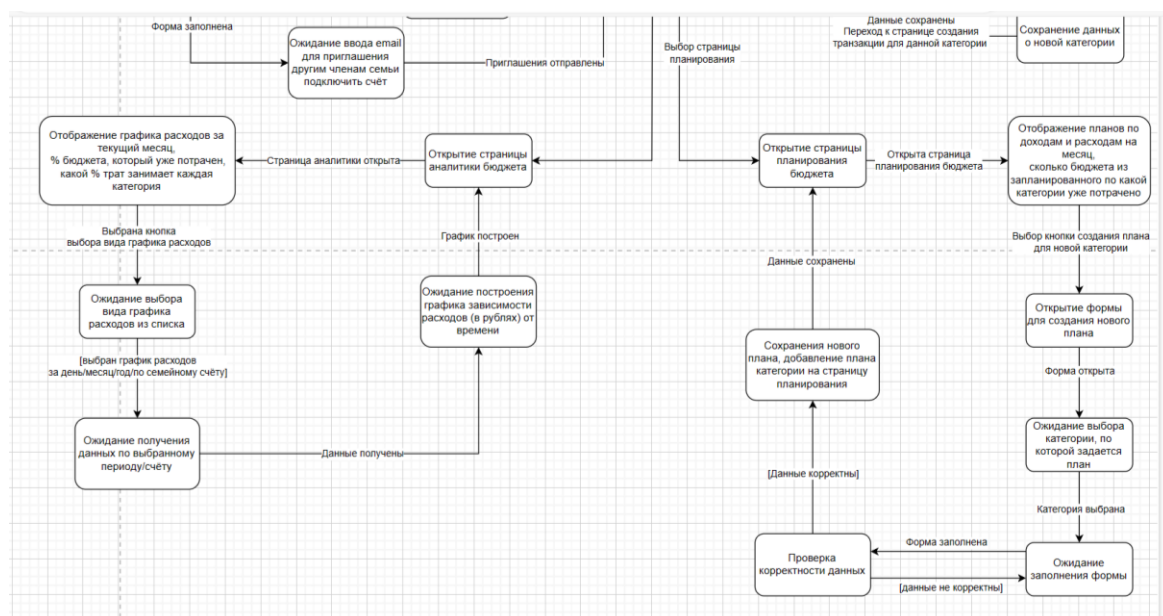


Рисунок 17 - State Chart диаграмма для авторизованного пользователя.

## Часть 3

## 2.2.4 Deployment диаграмма (развертывания)

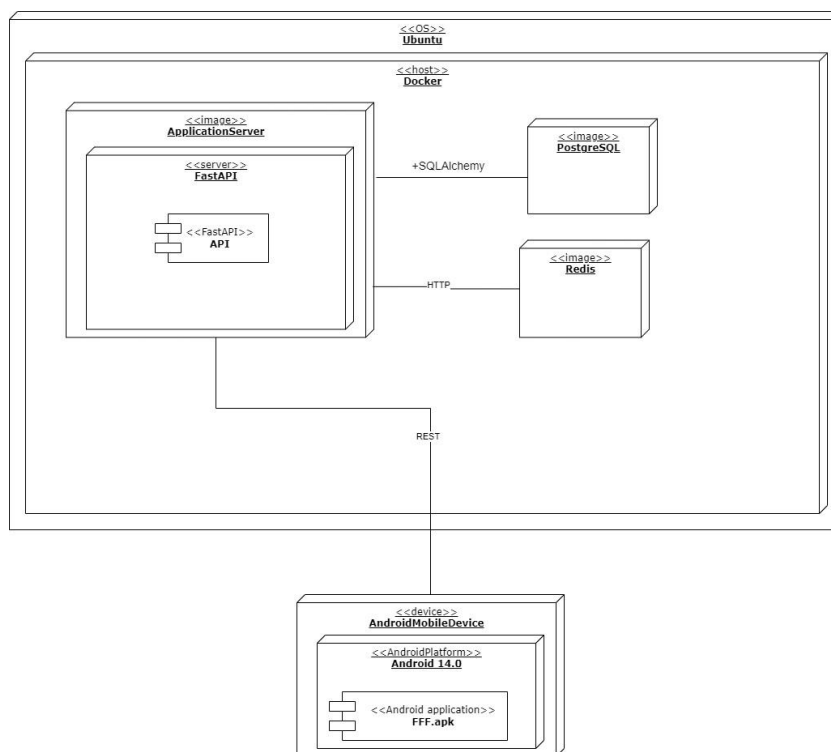


Рисунок 18 - Deployment диаграмма

## 2.2.5 ER-диаграмма

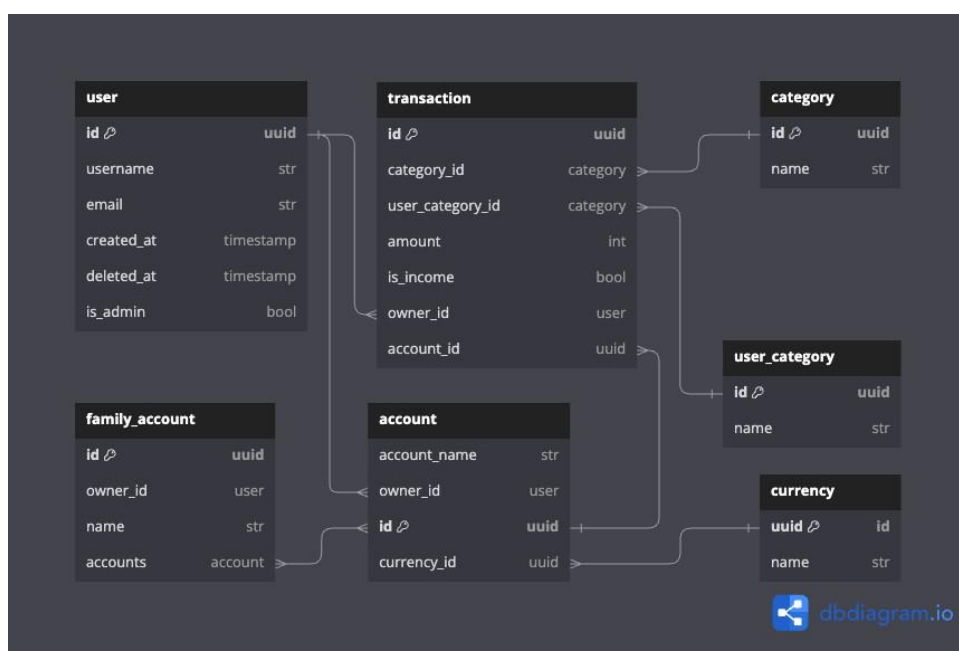


Рисунок 19 - ER-диаграмма

## 3 Реализация

### 3.1 Реализация клиентской части приложения

Для создания основных сценариев мобильного приложения клиентская часть делится на экраны. Каждый экран описывается языком Flutter и Dart. За основу были взяты заранее разработанные и согласованные командой макеты.

Архитектура разработки была организована согласно бизнес-логике проекта на основании модульного подхода. В процессе реализации для каждого экрана был создан отдельный файл для простоты расширения и разработки.

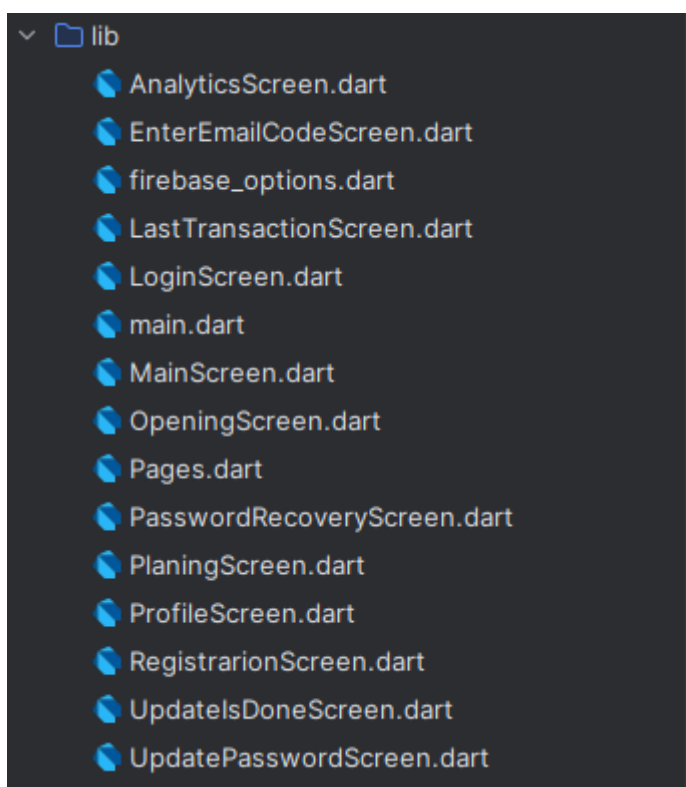


Рисунок 20 - Файлы (Классы) для экранов

Все экраны были реализованы и представлены командой разработчиков в соответствии с заявленным дизайном и соответствующими правками, внесенными в процессе разработки системы.

Основные экраны:

1. Начальный экран. Здесь пользователь может добавить, удалить, изменить транзакцию, также можно видеть траты за месяц по всем категориям и счетам.

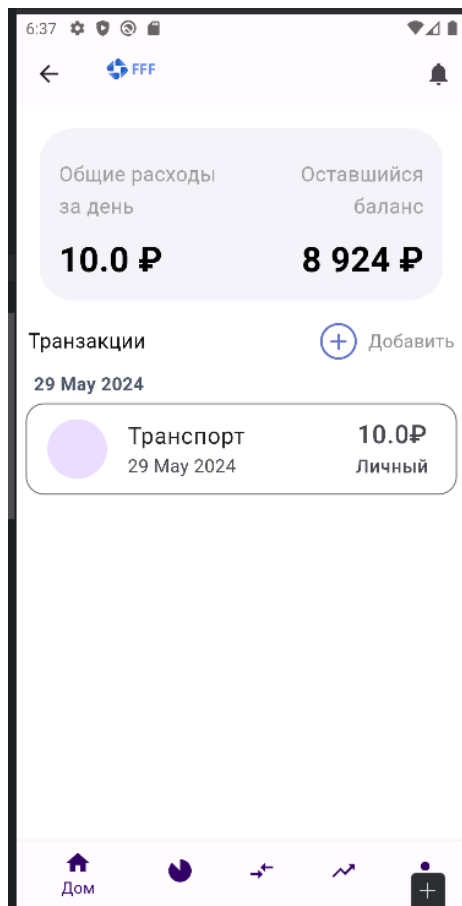


Рисунок 21 - Начальный экран

2. Экран планирования. На данном экране пользователь может задать ожидаемые расходы по каждой категории на месяц.

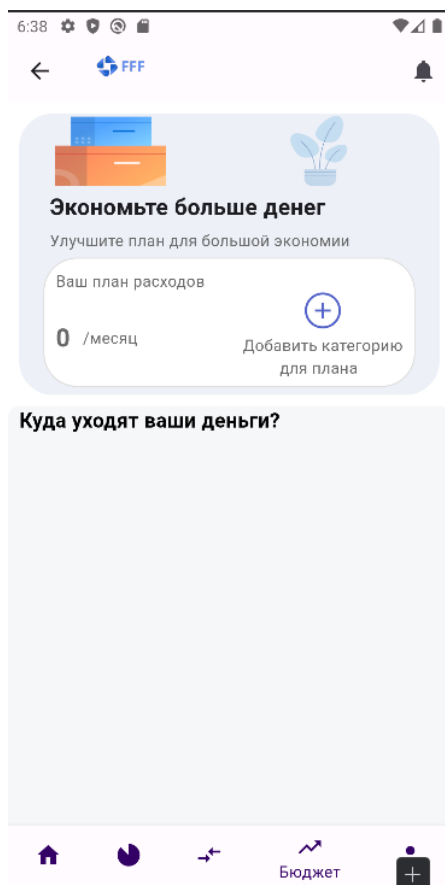


Рисунок 22 - Экран планирования

3. Профиль. Здесь можно изменить имя пользователя, адрес электронной почты, а также пароль. Для включения режима редактирования необходимо нажать на кнопку «Редактировать».

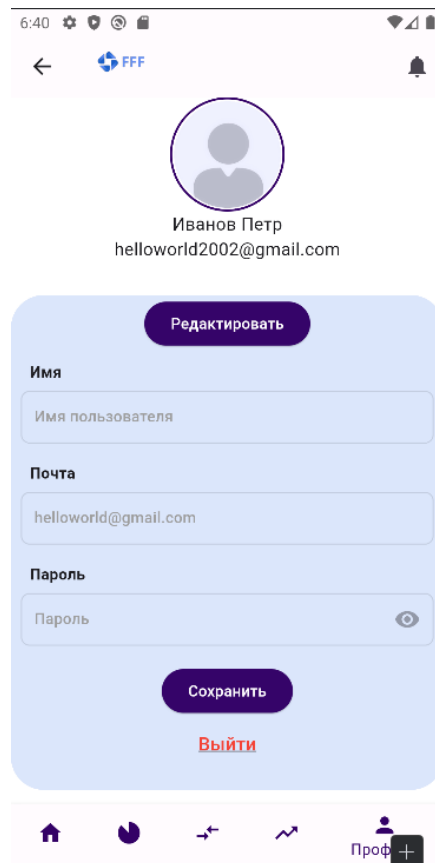


Рисунок 23 - Экран профиль

4. Расходы. На этом экране можно посмотреть счета пользователя, создать новый, отредактировать или удалить старый. При создании возможен выбор: личный или семейный счет. Различие заключается в том, что в семейный можно добавлять других пользователей. И каждый из них сможет видеть этот счет у себя в приложении.



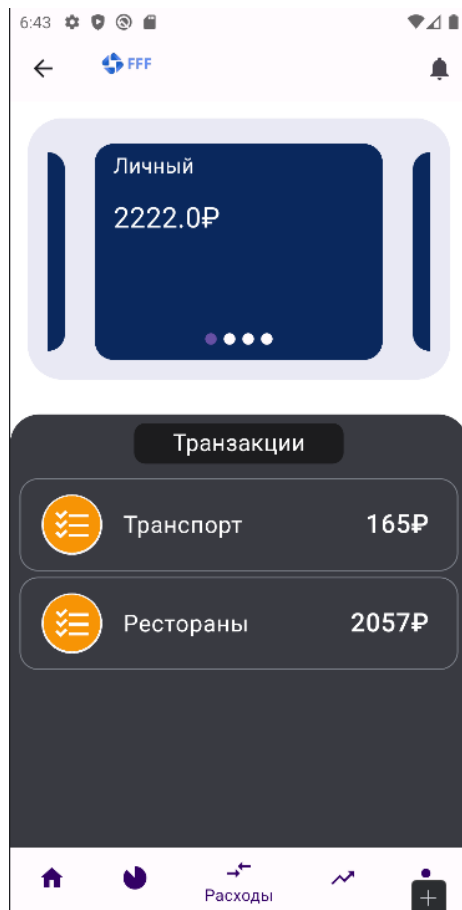


Рисунок 24 - Экран расходы

5. Экран аналитики. Здесь можно посмотреть расходы по каждой категории в месяц и их процентное распределение, а также на графике увидеть расходы по каждому месяцу.

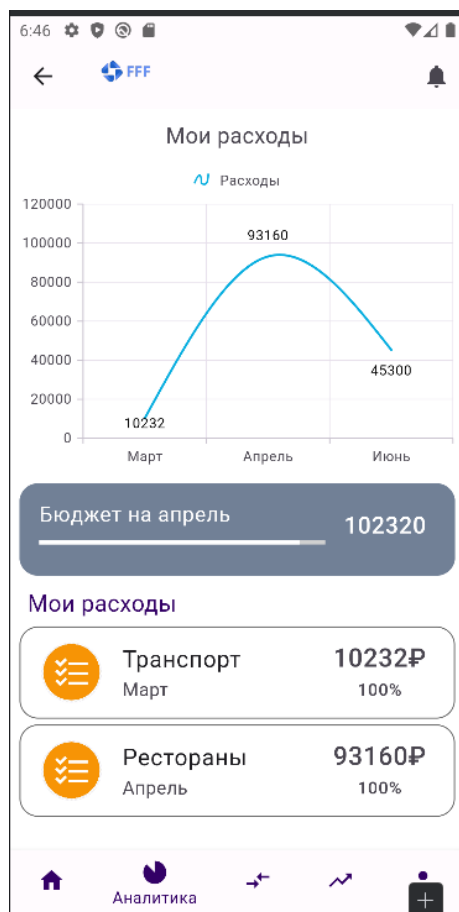


Рисунок 25 - Экран аналитика

### 3.2 Реализация серверной части приложения

Для осуществления основных сценариев работы приложения требуется авторизация, регистрация пользователей, хранение и синхронизация данных, за что отвечает серверная часть (backend).

При первичной регистрации пользователь задает свои основные данные, такие как логин, пароль, уникальный идентификатор пользователя в виде UUID4 присваивается автоматически и далее не будет возможности его изменить.

После прохождения авторизации на клиентскую часть приложения будет отправлен специальный токен для доступа в формате JWT (Json Web Token), в котором закодирована основная информация о пользователе. Токен

требуется отправлять при каждом запросе к серверной части приложения для идентификации пользователя и разграничения прав доступа к данным.

Серверная часть приложения написана с помощью языка Python и фреймворка FastAPI. Структура проекта представляет собой корневую папку, в которой хранятся различные вспомогательные файлы, такие как переменные окружения, файлы для работы с зависимостями проекта и файлы для миграций. В основном модуле, который называется “backend” храниться исполняемый код серверной части приложения. В ней также присутствуют различные модули, такие как модуль с кодом для обеспечения работы авторизации, код для работы с ORM (Object Relational Mapping), в качестве которой используется SQLAlchemy, модуль, который хранит настройки приложения, которые считываются из конфигурационного файла.

Хранение данных происходит в реляционной базе данных PostgreSQL. Для работы с ней, как уже было сказано выше, используется ORM библиотека SQLAlchemy, которая позволила описать все нужные модели данных и спроецировать их в сущности для реляционной базы данных. Для обеспечения версионирования базы данных используются миграции. Миграции сделаны с помощью библиотеки Alembic, которая де-факто является стандартом для работы с SQLAlchemy. Сериализация данных в json формат для отправки их в клиентскую часть приложения осуществляется с помощью моделей, которые построены на основе библиотеки Pedantic.

Взаимодействие с клиентской частью приложения происходит с применением RESTful API. Для удобства реализации которого и был выбран FastAPI, как фреймворк заточенный для создания программных интерфейсов. Документация реализована с помощью встроенных в фреймворк механизмов, которые используют Pedantic модели для описания принимаемых и возвращаемых данных.

## **ЗАКЛЮЧЕНИЕ**

В заключении хочется подвести итог всему, что было рассмотрено в данной работе выше.

С каждым годом проблема учёта и планирования бюджета становится всё более актуальной. С каждым годом банки получают всё больше запросов на открытие счетов, люди имеют большое количество категорий доходов и расходов. Сложность контроля своего бюджета растёт, и люди все больше испытывают потребность в удобном инструменте для централизации своих расходов, анализе и планировании своего бюджета.

Реализованное мобильное приложение удовлетворяет потребность пользователей в хранении всех своих транзакций в одном месте, анализе трат и планировании расходов на будущие периоды. Так же приложение предоставляет возможность создания семейного счёта, что способствует более успешному ведению семейного бюджета.

Таким образом, разработанное приложение при дальнейшем своем развитии сможет оказать существенную помощь пользователям в сохранении своего бюджета, минимизации излишних трат, планировании покупок на долгосрочные периоды и совместном учете расходов семьи.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Тинькофф [электронный ресурс]. – Режим доступа: [Ведение бюджета: как тратить деньги правильно \(tinkoff.ru\)](https://tinkoff.ru) – Как составлять семейный бюджет. – (Дата обращения: 15.04.2024).
2. The Ascent [электронный ресурс]. – Режим доступа: [PocketGuard Review: Pros, Cons, and More | The Motley Fool \(turbopages.org\)](https://turbopages.org). – Обзор PocketGuard: плюсы, минусы и многое другое. – (Дата обращения: 15.04.2024).
3. Статьи Дзен [электронный ресурс]. – Режим доступа: [Дзен-мани. Обзор мобильного приложения для учета доходов и расходов | Экономист экономит | Дзен \(dzen.ru\)](https://dzen.ru). – Дзен-мани. Обзор мобильного приложения для учета доходов и расходов. – (Дата обращения: 15.04.2024).
4. Статьи Дзен [электронный ресурс]. – Режим доступа: [Дзен-мани. Обзор мобильного приложения для учета доходов и расходов | Экономист экономит | Дзен \(dzen.ru\)](https://dzen.ru). – Дзен-мани. Обзор мобильного приложения для учета доходов и расходов. – (Дата обращения: 15.04.2024).
5. Business Incider [электронный ресурс]. – Режим доступа: [YNAB Review 2024: You Need a Budget App for Tracking Your Finances \(turbopages.org\)](https://turbopages.org). – Обзор YNAB 2024. – (Дата обращения: 15.04.2024).
6. Вести Экономика [электронный ресурс]. – Режим доступа: [Опрос: 51% россиян ведут учет своих доходов и расходов \(vesti.ru\)](https://vesti.ru). – Опрос россиян по учету своих доходов и расходов. – (Дата обращения: 15.04.2024).