

2. Макет

Главный экран

Баланс 152320 р

Все операции

Вклады Платежи

Открыть новый счет

Перевести деньги

Экран 1

Расходы 9600 р Доходы 17000 р

История операций

Сегодня

Четвергочка - 1200 р

28 октября

Интернет - 600 р

Полноценное + 3200 р

Кино - 500 р

Экран 2

Расходы -9600 р

Сегодня

Четвергочка - 1200 р

28 октября

Интернет - 600 р

Метро - 1000 р

Кино - 500 р

Экран 3

Супермаркет -3000 р

Сегодня

Четвергочка - 1200 р

28 октября

Укусоло - 300 р

Черное-белое - 800 р

Писки - 700 р

Экран 4

Писки

29 августа -1200 р

20 августа - 300 р

16 августа - 150 р

Писки - 100 р

Экран 5

Вернуться назад

Период: 1 авг

Период можно выбирать

Расходы 0 р

Нет трат за выбранный период

Экран 3а

3. Use case

Заголовок	Просмотр категорий трат за определенный промежуток времени
Актеры	Клиент
Предусловие	Пользователь является клиентом банка и авторизован в мобильном приложении банка
Ограничения	Наличие истории трат
Триггер	Клиент нажимает кнопку "Расходы" (Экран 2)
Основной сценарий	<ol style="list-style-type: none"> 1. Система отображает экран с разбивкой на категории, список и сумму расходов за текущий месяц (по умолчанию) (Экран 3) 2. Клиент выбирает категорию 3. Система отображает экран с разбивкой на подкатегории, список и сумму расходов соответствующей выбранной категории за период (Экран 4) 4. Клиент выбирает подкатегорию 5. Система отображает экран со списком и суммой расходов для выбранной подкатегории за период (Экран 5)
Критерий успеха:	Клиенту доступна история расходов с навигацией по

Альтернативный сценарий	2а. Система отображает экран с разбивкой на категории, список и сумму расходов за выбранный период → Переход к шагу 2 основного сценария с назначенным периодом
Исключительный сценарий	2б. Клиент выбирает период, за который не было произведено трат 3а. Система выдает экран, сообщающий об отсутствии трат за выбранный период (Экран 3а)

A - Фронтенд (мобильное приложение)
B - Бэкэнд / сервер
C - Реляционная БД PostgreSQL

1 - https
2 - протокол PostgreSQL

6. Модель данных

Объект Account (Счет)

Атрибут	Описание
balance	Текущий баланс счета
accountNumber	Номер счета
currency	Валюта счета

Account (Счет)

Содержит

7. ER-диаграмма

The ER diagram illustrates the relationships between three main entities: **Accounts**, **Counterparties**, and **Categories**.

- Accounts Entity:**
 - Primary Key: `account_id INT (PK)`
 - Attributes: `account_number INT`, `balance DECIMAL`, `currency VARCHAR`
- Counterparties Entity:**
 - Primary Key: `counterparty_id INT (PK)`
 - Attributes: `title VARCHAR`, `type VARCHAR`, `msc INT`
- Categories Entity:**
 - Primary Key: `category_id INT (PK)`
 - Attributes: `name VARCHAR`, `description TEXT`

Relationships:

- содержит (contains):** Connects **Accounts** and **Counterparties**.
- определяет (determines):** Connects **Accounts** and **Categories**.
- определяет (determines):** Connects **Counterparties** and **Categories**.

Table Content:

определяет	
Category (Категория)	
date	Дата транзакции
amount	Сумма транзакции
type	Тип операции
counterparty	Название контрагента
msc	MCC-код операции

Объект Category (Категория)	
Атрибут	Описание
name	Название категории (например, "Супермаркет")
description	Описание категории
msc	Массив MCC-кодов, относящихся к категории

Объект Counterparty (Контрагент)	
Атрибут	Описание
title	Отображаемое название контрагента (например, "Четверочка")
names	Массив возможных названий контрагента ("Четверочка", "Четверочка дистрибут")
type	Тип контрагента (организация/физлицо/банк/смит)
msc	MCC-код контрагента

```
sequenceDiagram
    actor User
    participant Frontend
    participant DB as БД
    User->>Frontend: 1 Пользователь нажимает кнопку "Расходы"
    Frontend->>DB: 2 Запрос на список расходов и категории расходов
    DB-->>Frontend: 3 Возвращает данные
    Frontend-->>User: 5 Возвращает ответ со списком расходов и категориями
    User-->>Frontend: 6 Отображает категории и список расходов
```

2	Frontend -> Backend	Запрос на список расходов и категории расходов
3	Backend -> DB	SQL-запрос на формирование списка расходов и категорий
4	DB -> Backend	Возвращает список расходов и категорий
5	Backend ->> Frontend	Возвращает ответ со списком расходов и категорий
6	Frontend ->> User	Отображает категории и список расходов

9. REST. Табличный вид

Request

Название параметра	Принадлежность	Тип данных	Описание	Обязательность параметра
date_from	query	date	Выбор от какой даты	нет
date_to	query	date	Выбор до какой даты	нет
page	query	int	Номер текущей страницы	нет
page_size	query	int	Количество записей на странице	нет

Response code 200

Название параметра	Принадлежность	Тип данных	Описание	Обязательность параметра
transactions	body	array	Массив транзакций	да
transaction	body	object	объект содержащий данные о транзакции	да
transaction_id	body	int	id транзакции	да

amount	body	decimal	сумма транзакции	да
type	body	string	тип транзакции	да
name_id	body	int	id названия контрагента	да
mcc_id	body	int	id mcc кода	да
pagination	body	object	объект, содержащий параметры пагинации	да
page	body	int	текущая страница	да
page_size	body	int	количество записей на странице	да
total_pages	body	int	общее количество страниц	да
total_items	body	int	общее количество записей	да

10. Swagger

Servers

<https://bank.ru/api/v1>

Transactions

Операции для получения данных о транзакциях

GET /transactions

Запрос списка транзакций за период времени

Parameters

Name	Description
------	-------------

date_to

string(date)

(query)

Выбор до какой даты (ГГГГ-ММ-ДД). Если не указано, используется текущая дата.

Example : 2024-12-27

2024-12-27

page

integer

(query)

Номер текущей страницы

Default value : 1

Example : 1

1

page_size

integer

(query)

Количество записей на странице

Default value : 10

Example : 10

10

Responses

Code	Description	Links
200	Успешное выполнение операции. Media type <div>application/json</div> Content Accept Header: Example Value Schema	No links

```
{
  "items": [
    {
      "transaction_id": 123,
      "account_id": 456,
      "date": "2024-12-27",
      "amount": 500,
      "type": "omnata",
      "name_id": 789,
      "mcc_id": 9411
    }
  ],
  "pagination": {
    "page": 1,
    "page_size": 10,
    "total_pages": 5,
    "total_items": 50
  }
}
```

11. Критерии приемки

User story	
Я, как пользователь банковского приложения, хочу видеть категории своих трат за определенный промежуток времени для того, чтобы более осознанно тратить деньги.	
Use case Основной сценарий	<ol style="list-style-type: none"> Система отображает экран с разбивкой на категорию, список и сумму расходов за текущий месяц (по умолчанию) (Экран 3) Клиент выбирает категорию Система отображает экран с разбивкой на подкатегорию, список и сумму расходов соответствующих выбранной категории за период (Экран 4) Клиент выбирает подкатегорию Система отображает экран со списком и суммой расходов для выбранной подкатегории за период (Экран 5).

← Период: октябрь → Период можно выбирать

Среднемесячные расходы Среднедневные расходы
Текущие расходы Среднедневные расходы

Расходы Сегодня -9600 p

28 октября

- Чайварочка - 1200 p
- Интернет - 600 p
- Метро - 1000 p
- Кино - 500 p

Экран 3

Функциональность: Просмотр истории расходов
Дано: пользователь просматривает список расходов (экран 3)
Сюжет: пользователь выбирает период за который дала бы ему статистику

отсутствует соединение с интернетом"

12. Нефункциональные требования:

Требования надежности:

- 1. Система должна быть доступна 99% времени.
- 2. В случае потери связи с интернетом, система должна автоматически продлить работу после восстановления соединения

Требования производительности:

- 1. История данных загружаться не более чем 0,5 секунды
- 2. Запрос на получение списка транзакции должен выдерживать нагрузку не менее 2 грс.