

## Use Case: Виртуальный Финансовый Консультант

### Веб-интерфейс + Telegram-бот

#### User Story:

*"Как пользователь, я хочу видеть наглядную диаграмму своих расходов за месяц, чтобы оптимизировать бюджет."*

## Заголовок

Пользователь анализирует расходы через визуализацию и получает рекомендации по оптимизации бюджета с помощью ИИ.

## Акторы

- **Пользователь** (основной актор).
- **ИИ-аналитик** (ML-модуль для категоризации и рекомендаций).
- **Банковская система** (источник данных через API).
- **Telegram-бот** (альтернативный интерфейс).

## Предусловия

- Пользователь зарегистрирован в системе и подключил минимум одну банковскую карту.
- ИИ-аналитик обучен на исторических данных пользователя (если есть) или общих шаблонах трат.
- Доступен веб-интерфейс или Telegram-бот.

## Постусловия

- Пользователь видит наглядную диаграмму расходов с категориями.
- Рекомендации по оптимизации сохранены в профиле и учтены для будущего анализа.
- Данные синхронизированы между веб-интерфейсом и Telegram-ботом.

## Ограничения

- Максимум **10 категорий** для автоматической классификации.
- Фильтрация расходов возможна только за последние **12 месяцев**.
- Telegram-бот обрабатывает запросы только в формате текста/кнопок (без сложных графиков).

## Триггер

Пользователь:

- Открывает раздел **«Анализ расходов»** в веб-интерфейсе;
- Или отправляет команду /статистика в Telegram-боте.

# Основной сценарий

## Веб-интерфейс

### 1. Веб-интерфейс

#### Шаг 1. Автоматический анализ расходов

- Пользователь открывает раздел «Анализ расходов» → система загружает транзакции через банковский API.
- ИИ категоризирует траты и показывает диаграмму.

#### Шаг 2. Ручное добавление расходов

- Пользователь нажимает кнопку «Добавить расход»:
  - Вводит сумму (например, 5 000 Р).
  - Выбирает категорию из списка (Еда, Транспорт, Другое).
  - Указывает дату (по умолчанию — текущая).
  - Опционально: добавляет комментарий («Обед с клиентом»).
- Система сохраняет запись и обновляет диаграмму в реальном времени.

#### Шаг 3. Редактирование/удаление

- Пользователь может:
  - Исправить категорию ошибочной транзакции (например, перевести «Такси» из «Транспорт» в «Бизнес»).
  - Удалить дубликаты вручную.

## Telegram-бот

1. Пользователь пишет: /статистика\_август.

2. Бот:

- Запрашивает данные из облака (синхронизация с веб-версией).
- Отправляет упрощённую диаграмму (текст/изображение).

3. Пользователь выбирает: «Оптимизировать» → бот предлагает:

- Установить лимит.
- Перейти в веб-интерфейс для деталей.

#### Шаг 1. Команда для ручного ввода

- Пользователь отправляет:

/add\_расход 5000 еда "Продукты на неделю"

- Бот:

- Парсит сумму (5000), категорию (еда), описание.
- Подтверждает:

✓ Добавлено: 5 000 Р (Еда)

Комментарий: Продукты на неделю

#### Шаг 2. Интерактивный режим

- Пользователь пишет: «Добавить расход» → бот задаёт вопросы:

Бот: Введите сумму:

Пользователь: 3000

Бот: Выберите категорию:

1. ☐ Еда
2. ☐ Транспорт
3. ☐ Прочее

Пользователь: 1

Бот: Готово! 3 000 Р (Еда).

## Интеграция ручного ввода

- **Синхронизация:** Ручные расходы из бота и веб-интерфейса объединяются в одном отчёте.
- **Обработка конфликтов:** Если транзакция добавлена вручную и автоматически (например, по чеку), ИИ предлагает **объединить дубликаты**.

## Альтернативные сценарии

### 1. Нет данных за месяц

- Веб-интерфейс: «Подключите карту или введите траты вручную».
- Бот: «Не найдено транзакций. Используйте /add\_расход [сумма] [категория]».

### 2. Низкая точность категоризации

- ИИ помечает транзакцию как «Неизвестно» → предлагает пользователю выбрать категорию вручную.

### 3. Превышен лимит категории

- Бот присылает уведомление:  
\*«☐ Лимит на транспорт превышен! (5 000/4 000 Р)»\*.

### 4. Примеры для ручного ввода

#### 1. Неверный формат ввода в боте

- Пользователь: /add\_расход 5000 (нет категории).
- Бот: «Укажите категорию: /add\_расход [сумма] [категория] [описание]».

#### 2. Добавление расхода без интернета

- Веб-интерфейс: Данные сохраняются локально → синхронизируются при восстановлении соединения.
- Бот: «Оффлайн-режим. Запись сохранена временно. Отправьте /sync при подключении».

## Исключительные сценарии

### 1. Ошибка банковского API

- Веб: «Не удалось загрузить данные. Попробуйте позже».
- Бот: «Ошибка синхронизации с банком. Проверьте подключение».

## 2. Сбой ИИ-аналитика

- Рекомендации заменяются на шаблонные советы (например, «Попробуйте вести бюджет вручную»).

## 3. Неверная команда в боте

- Бот: «Не понимаю. Доступные команды: /help».

## Критерии успеха

- Точность категоризации транзакций  $\geq 90\%$ .
- Время формирования отчёта  $\leq 3$  секунд.
- $\geq 70\%$  пользователей применяют хотя бы одну рекомендацию.
- Синхронизация веб-бота и Telegram происходит за  $\leq 1$  секунды.

### Критерии успеха для ручного ввода

- Время добавления расхода:  $\leq 5$  секунд (бот),  $\leq 10$  секунд (веб).
- Поддержка **10+** категорий и пользовательских тегов.
- Возможность импорта/экспорта в CSV (для веба).

## Технические детали

Для веб-интерфейса:

- **Frontend:** React + Chart.js.
- **Backend:** Python (FastAPI) + ML (CatBoost для классификации).
- **Хранилище:** PostgreSQL + Redis (кэш).

Для Telegram-бота:

- **Библиотеки:** aiogram (Python).
- **Команды:**
  - /start — подключение банка.
  - /статистика — отчёт за текущий месяц.
  - /лимит [категория] [сумма] — установка лимита.

### Диаграмма взаимодействия



