

Заголовок	<p>Пользователь анализирует расходы через визуализацию и получает рекомендации по оптимизации бюджета с помощью ИИ.</p>
Акторы	<ul style="list-style-type: none"> - Пользователь (основной актор) - ИИ-аналитик (ML-модуль) - Банковская система (API) - Telegram-бот - Пользователь зарегистрирован и подключил минимум одну банковскую карту.
Предусловия	<ul style="list-style-type: none"> - ИИ-аналитик обучен на исторических данных. - Доступен веб-интерфейс или Telegram-бот. - Пользователь видит диаграмму расходов.
Постусловия	<ul style="list-style-type: none"> - Рекомендации сохранены в профиле. - Данные синхронизированы между веб-интерфейсом и ботом. - Максимум 10 категорий для автоматической классификации.
Ограничения	<ul style="list-style-type: none"> - Фильтрация только за последние 12 месяцев. - Бот поддерживает только текстовые команды/кнопки.
Триггер	<ul style="list-style-type: none"> - Пользователь открывает раздел «Анализ расходов» в веб-интерфейсе. - Или отправляет команду /статистика в боте.
Основной сценарий	<p>Веб-интерфейс:</p> <ol style="list-style-type: none"> 1. Автоматический анализ расходов через банковский API. 2. Ручное добавление расходов (сумма, категория, дата). 3. Редактирование/удаление транзакций. <p>Telegram-бот:</p> <ol style="list-style-type: none"> 1. Команда /add_расход 5000 еда "Продукты". 2. Интерактивный режим: запрос суммы и категории.
Альтернативные сценарии	<ol style="list-style-type: none"> 1. Нет данных за месяц → предложение ввести вручную. 2. Низкая точность категоризации → запрос ручного выбора. 3. Превышен лимит категории → уведомление. 4. Оффлайн-режим → локальное сохранение данных.
Исключительные сценарии	<ol style="list-style-type: none"> 1. Ошибка банковского API → сообщение об ошибке. 2. Сбой ИИ → шаблонные рекомендации. 3. Неверная команда в боте → подсказка /help.

Критерии успеха

- Точность категоризации $\geq 90\%$.
- Время формирования отчёта ≤ 3 сек.
- $\geq 70\%$ пользователей применяют рекомендации.
- Синхронизация ≤ 1 сек.

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

Диаграмма взаимодействия

