

Use case

Заголовок	Пользователь анализирует расходы через визуализацию и получает рекомендации по оптимизации бюджета с помощью ИИ.
Акторы	<ul style="list-style-type: none">- Пользователь (основной актор)- ИИ-аналитик (ML-модуль)- Банковская система (API)- Telegram-бот
Предусловия	<ul style="list-style-type: none">- Пользователь зарегистрирован и подключил минимум одну банковскую карту.- ИИ-аналитик обучен на исторических данных.- Доступен веб-интерфейс или Telegram-бот.
Постусловия	<ul style="list-style-type: none">- Пользователь видит диаграмму расходов.- Рекомендации сохранены в профиле.- Данные синхронизированы между веб-интерфейсом и 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: <ul style="list-style-type: none">- Открывает раздел «Анализ расходов» в веб-интерфейсе;- Или отправляет команду /статистика в Telegram-боте.
Основной сценарий	Веб-интерфейс 1. Пользователь открывает платформу и авторизуется (логин/пароль или через Google/банк-клиент) → система проверяет данные и загружает личный кабинет. 2. ИИ-агент автоматически подключает банковские карты (через Open Banking API) и загружает историю транзакций → Отображается уведомление: "Подключено 2 карты. Загружено 45 транзакций за август". 3. Пользователь выбирает раздел «Анализ расходов» → ИИ категоризирует траты (еда, транспорт, подписки) и строит круговую диаграмму. 4. ИИ-агент предлагает «Оптимизировать бюджет» и уточняет цель: "Хотите копить или сократить долги?" → Пользователь выбирает "Накопить 50 000 ₽ на отпуск". 5. Пользователь вводит параметры: срок (3 месяца), приоритетные категории для сокращения (развлечения, доставка еды) → ИИ сохраняет настройки. 6. ИИ-агент анализирует данные: Сравнивает с прошлыми месяцами, Выявляет избыточные траты (например, 12 000 ₽ на кафе) → Показывает тренды: "Расходы на транспорт выросли на 20%". 7. ИИ формирует рекомендации: "Сократите кафе до 8 000 ₽/мес (+4 000 ₽ к накоплениям)", "Откройте вклад под 7% годовых" → Отчёт включает графики и кнопку «Применить».

	<p>8. Пользователь корректирует рекомендации (например, увеличивает лимит на кафе) → ИИ пересчитывает план.</p> <p>9. Пользователь нажимает «Сохранить план» → ИИ устанавливает лимиты, создаёт цель в разделе «Накопления».</p> <p>10. ИИ-агент запускает еженедельный мониторинг: Присылает уведомления о приближении к лимиту, Обновляет статистику Пример: "На этой неделе сэкономили 1 500 ₽".</p> <p>Telegram-бот</p> <p>1. Пользователь пишет /start → Бот отправляет меню: "Выберите действие: /статистика, /добавить_расход, /оптимизировать".</p> <p>2. Пользователь выбирает /оптимизировать → Бот уточняет: "Какую цель преследуете? 1. Накопления 2. Сокращение долгов".</p> <p>3. Пользователь выбирает "Накопления" и вводит сумму/срок (50 000 ₽ за 3 месяца) → Бот запрашивает данные из облака (синхронизация с веб-версией).</p> <p>4. ИИ анализирует расходы и предлагает: - *"Сократите развлечения до 10 000 ₽/мес"*, "Переведите 15 000 ₽ на накопительный счёт" → Отправляет краткий отчёт (текст + кнопки «Принять», «Изменить»).</p> <p>5. Пользователь нажимает «Изменить» и корректирует лимиты через интерактивные кнопки → Бот обновляет рекомендации.</p> <p>6. Пользователь подтверждает план кнопкой «Сохранить» → Бот сообщает: *"План сохранён! Лимит на кафе: 8 000 ₽/мес"*.</p> <p>7. ИИ-агент ежедневно проверяет траты: - При превышении лимита отправляет: *"Лимит на транспорт исчерпан (5 000/4 000 ₽)"*, - Раз в неделю присылает сводку: Пример: "За неделю накопили 3 200 ₽ из 50 000 ₽".</p>
Альтернативные сценарии	<p>1. Нет данных за месяц → предложение ввести вручную.</p> <p>2. Низкая точность категоризации → запрос ручного выбора.</p> <p>3. Превышен лимит категории → уведомление.</p> <p>4. Оффлайн-режим → локальное сохранение данных.</p>
Исключительные сценарии	<p>1. Ошибка банковского API → сообщение об ошибке.</p> <p>2. Сбой ИИ → шаблонные рекомендации.</p> <p>3. Неверная команда в боте → подсказка /help.</p>
Критерии успеха	<ul style="list-style-type: none"> - 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

