

User Story

HairCare AI агент

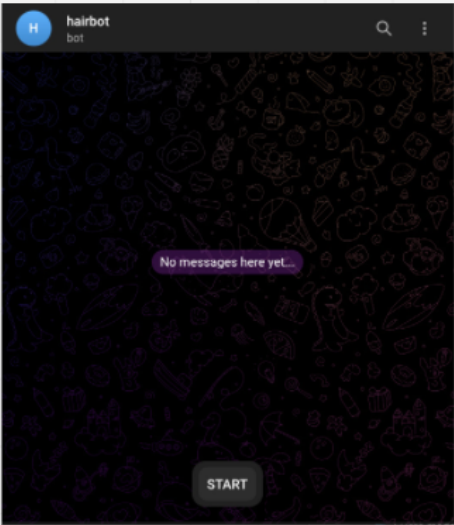
Как покупатель средств для волос,
я хочу отправить текст состава средства для волос в Telegram-бот,
чтобы получить анализ компонентов и понять эффективность средства

Use Case

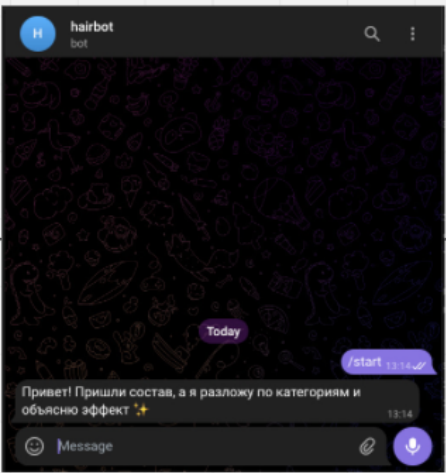
UC	UC-001
Название	Анализ состава средства для волос
User Story	US-001: Как покупатель средств для волос, я хочу отправить текст состава средства для волос в Telegram-бот, чтобы получить анализ компонентов и понять эффективность средства
Участники	Основной: Пользователь (покупатель) Вторичные: AI-агент (Telegram Bot, Backend API, LLM)
Предусловия	1. Пользователь имеет доступ к Telegram боту 2. AI-агент доступен и работает корректно. 3. У пользователя есть текст состава
Ограничения	1. Максимальный размер входного описания – 4096 символов 2. Поддерживаются только текст 3. Время ответа ≤ 60 секунд 4. Если компонент отсутствует в справочнике, он возвращается как неизвестный
Триггер	Пользователь отправляет состав на анализ
Основной сценарий	1. Пользователь отправляет состав боту 2. Бот передаёт текст в Backend API 3. API вызывает LLM для анализа состава 4. LLM выделяет активные компоненты, классифицирует их (по категориям: уход, лечение, косметический эффект) 5. API возвращает результат в JSON 6. Бот показывает пользователю структурированный ответ (по категориям, с описанием действия каждого ингредиента)
Альтернативные сценарии	п.3а: Если состав слишком короткий или содержит менее 2 ингредиентов <ul style="list-style-type: none">AI-агент запрашивает у пользователя уточнения.Telegram-бот отображает запрос пользователю.Пользователь дополняет текст состава. Переход к шагу 4 основного сценария. п.4а: Если AI-агент не распознал часть компонентов <ul style="list-style-type: none">Используется частичный результат анализа.Telegram-бот уведомляет пользователя, что некоторые компоненты не найдены. Переход к шагу 6 основного сценария. п.6а: Если результат анализа недостаточно понятен пользователю <ul style="list-style-type: none">Пользователь задаёт уточняющий вопрос.AI-агент формирует пояснение (например: «Этот компонент отвечает за увлажнение»).Telegram-бот возвращает дополненный ответ. Переход к шагу 6 основного сценария.
Исключительный сценарий	п.5и: Если время анализа превышено (таймаут) <ul style="list-style-type: none">Система прерывает процесс анализа.Telegram-бот уведомляет пользователя об ошибке.Предлагается повторить запрос позже.
Постусловия	1. Результат анализа сохранён в истории переписки 2. Пользователь может вернуться и просмотреть прошлые анализы

3 Wireframe

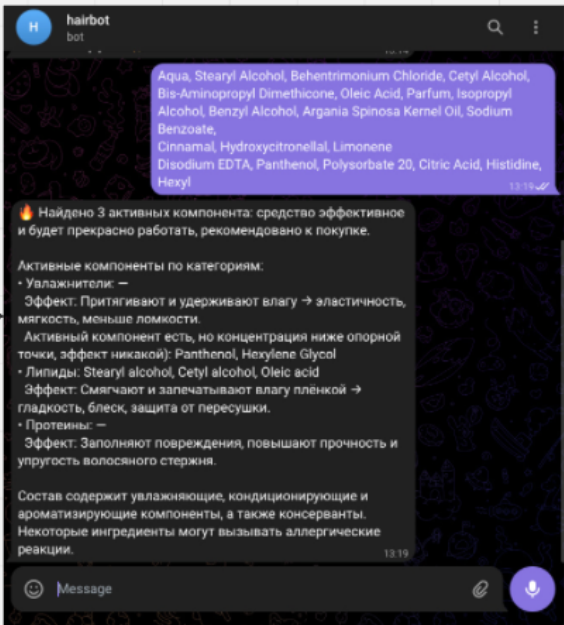
Экран 1



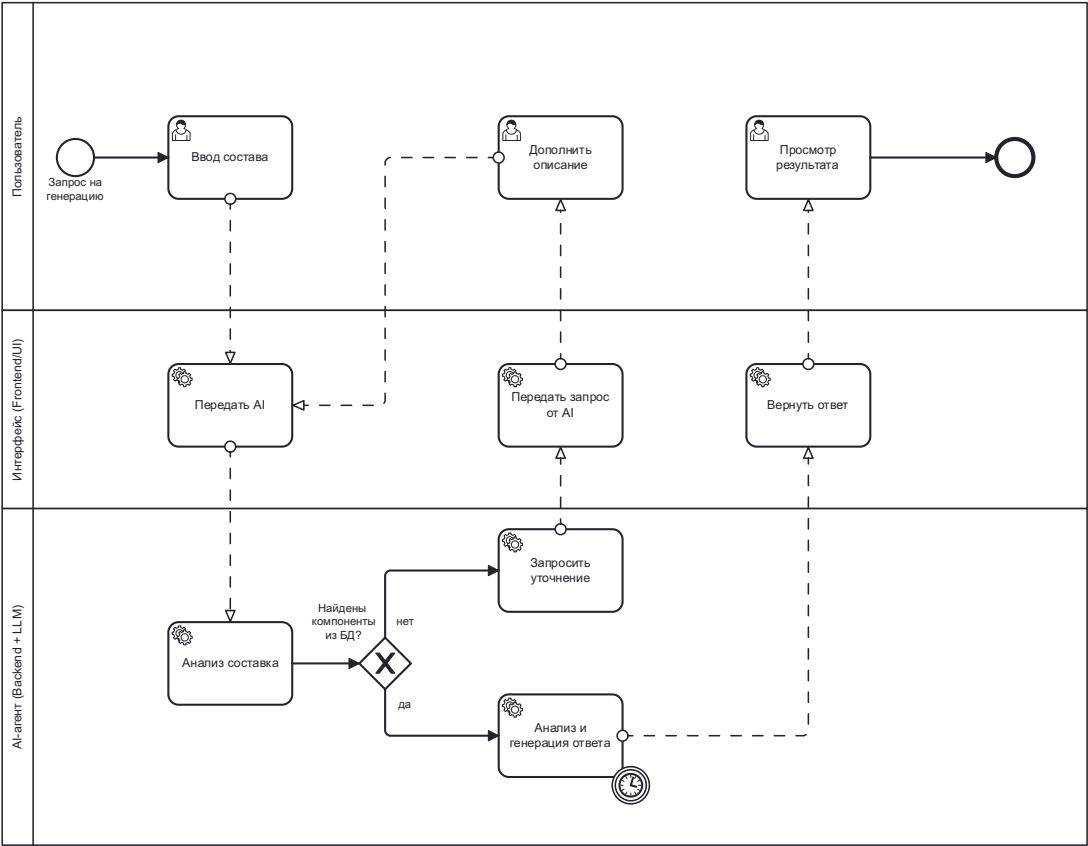
Экран 2



Экран 3

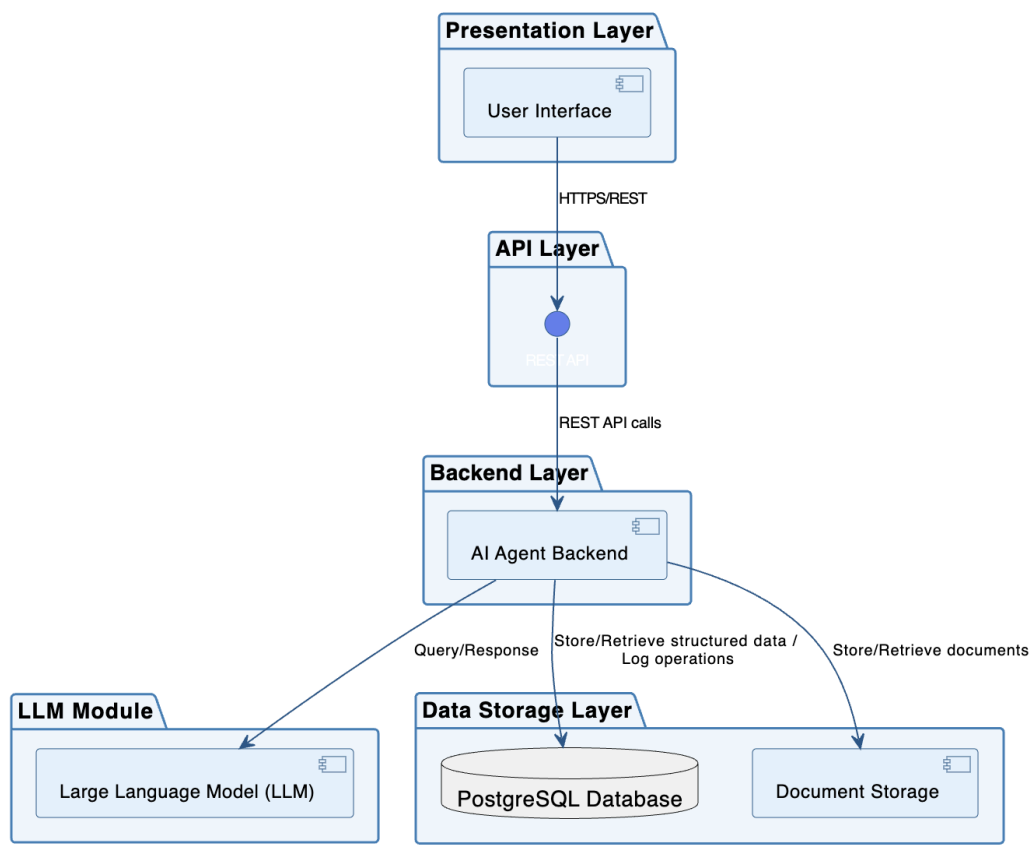


4 BPMN

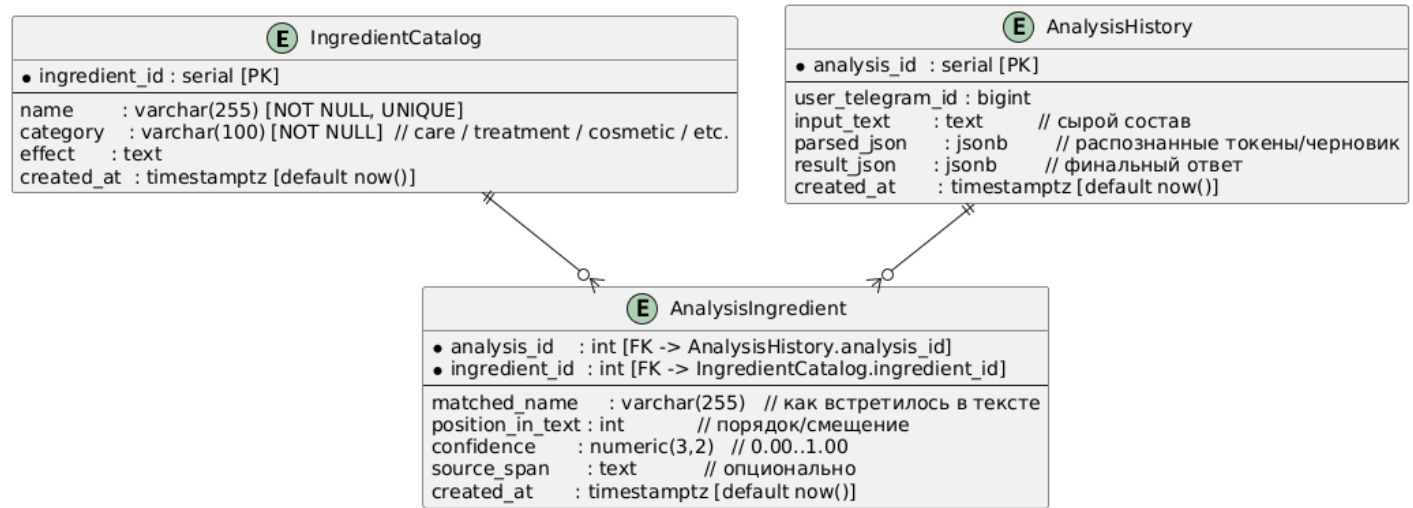


5 Architecture

Компонентная диаграмма - AI-агент для генерации черновика документации



6 ERD



Data model

IngredientCatalog

Атрибут	Тип данных	Обязательность	Описание/ограничения
ingredient_id	SERIAL	PK	Идентификатор ингредиента
name	VARCHAR(255)	NOT NULL, UNIQUE	Название ингредиента
category	VARCHAR(100)	NOT NULL	Категория: уход / лечение / косметика и др.
effect	TEXT	NULL	Описание эффекта компонента
created_at	TIMESTAMPTZ	NOT NULL, DEFAULT now()	Дата добавления в справочник

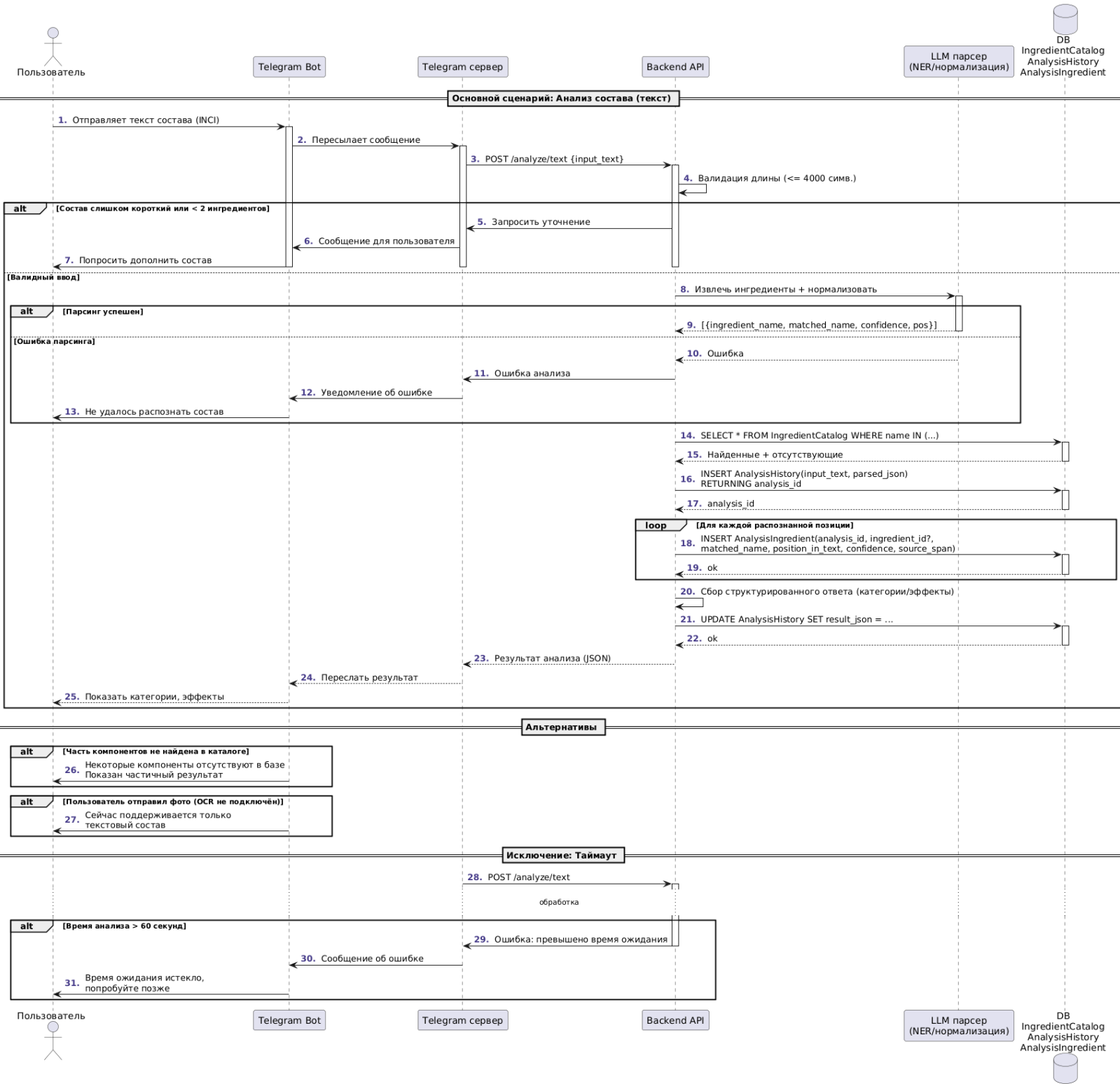
AnalysisHistory

Атрибут	Тип данных	Обязательность	Описание/ограничения
analysis_id	SERIAL	PK	Идентификатор анализа
user_telegram_id	BIGINT	NULL	Telegram ID пользователя
input_text	TEXT	NOT NULL	Сырой состав от пользователя
parsed_json	JSONB	NULL	Распознанные компоненты (черновик)
result_json	JSONB	NULL	Финальный результат анализа
created_at	TIMESTAMPTZ	NOT NULL, DEFAULT now()	Дата анализа

AnalysisIngredient

Атрибут	Тип данных	Обязательность	Описание/ограничения
analysis_id	INT	FK→AnalysisHistory(analysis_id), NOT NULL	К какому запросу относится
ingredient_id	INT	FK→IngredientCatalog(ingredient_id), NOT NULL	Какой ингредиент найден
matched_name	VARCHAR(255)	NULL	Как встретился в тексте (синоним/орфография)
position_in_text	INT	NULL	Позиция в составе
confidence	NUMERIC(3,2)	NULL	Уверенность (0.00–1.00)
source_span	TEXT	NULL	Фрагмент исходного текста
created_at	TIMESTAMPTZ	NOT NULL, DEFAULT now()	Дата фиксации

7 Sequence diagram



8 REST API

GET /ping

Проверка доступности сервиса

Request — параметры не требуются

Response <200>

Название параметра	Тип данных	Описание	Обязательность
ok	boolean	Признак доступности сервиса	да

POST /analyze/text

Анализ состава по текстовому описанию

Request

Название параметра	Тип данных	Описание	Обязательность
text	string	Текст с перечнем ингредиентов или описанием средства	да

Response <200>

По умолчанию возвращается JSON со схемой:

Название параметра	Тип данных	Описание	Обязательность
parsed_ingredients	array<object>	Распознанные ингредиенты	да
parsed_ingredients[].name	string	Название ингредиента	да
parsed_ingredients[].category	string	Категория (например, surfactant, preservative)	да
parsed_ingredients[].stated_conc_pct	number null	Указанный % концентрации (или null)	да
parsed_ingredients[].sufficiency	string	Оценка достаточности: enough low unknown	да
parsed_ingredients[].effects	array<string>	Выявленные эффекты	да
parsed_ingredients[].notes	array<string>	Дополнительные примечания	да
active_count	number	Количество активных компонентов	да
warnings	array<string>	Замечания/риски	да
overall_verdict	string	Итоговый вывод	да
summary	string	Краткое резюме анализа	да

Response code <422> Validation Error

Название параметра	Тип данных	Находится в	Описание
detail	array<object>	body	Детали ошибки валидации входных данных

Response code <400> Bad Request

Название параметра	Тип данных	Находится в	Описание
detail	string	body	Сообщение об ошибке входного текста

Response code <500> Internal Server Error

Название параметра	Тип данных	Находится в	Описание
detail	string	body	Сообщение о внутренней ошибке

Response code <503> Service Unavailable

Название параметра	Тип данных	Находится в	Описание
detail	string	body	Сервис временно недоступен

9 Swagger

```
openapi: 3.0.3
info:
  title: Hair-Components Analyzer API
  version: "1.0.0"
  description: >
    Сервис проверки доступности и анализа состава по текстовому описанию.
    Спецификация составлена по документу "8 REST.docx".
servers:
  - url: https://api.example.com
    description: Production
  - url: http://localhost:8000
    description: Local

paths:
  /ping:
    get:
      summary: Проверка доступности сервиса
      operationId: ping
      responses:
        "200":
          description: OK
          content:
            application/json:
              schema:
                $ref: "/components/schemas/PingResponse"
              examples:
                ok:
                  value: { ok: true }

  /analyze/text:
    post:
      summary: Анализ состава по текстовому описанию
      operationId: analyzeText
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              required: [text]
              properties:
                text:
                  type: string
                  description: Текст с перечнем ингредиентов или описанием средства
      responses:
```

```
"200":
  description: Успешный анализ (JSON-схема по умолчанию)
  content:
    application/json:
      schema:
        $ref: "/components/schemas/AnalysisResponse"
"400":
  description: Bad Request – ошибка входного текста
  content:
    application/json:
      schema:
        $ref: "/components/schemas/ErrorMessage"
"422":
  description: Validation Error – ошибки валидации входных данных
  content:
    application/json:
      schema:
        $ref: "/components/schemas/ValidationError"
"500":
  description: Internal Server Error – внутренняя ошибка
  content:
    application/json:
      schema:
        $ref: "/components/schemas/ErrorMessage"
"503":
  description: Service Unavailable – сервис временно недоступен
  content:
    application/json:
      schema:
        $ref: "/components/schemas/ErrorMessage"

components:
  schemas:
    PingResponse:
      type: object
      required: [ok]
      properties:
        ok:
          type: boolean
          description: Признак доступности сервиса

    Ingredient:
      type: object
      required:
        [ name, category, stated_conc_pct, sufficiency, effects, notes ]
      properties:
        name:
          type: string
```



```
    description: Название ингредиента
category:
  type: string
  description: Категория (например, surfactant, preservative)
stated_conc_pct:
  type: number
  nullable: true
  description: Указанный % концентрации (или null)
sufficiency:
  type: string
  description: Оценка достаточности
  enum: [enough, low, unknown]
effects:
  type: array
  items:
    type: string
  description: Выявленные эффекты
notes:
  type: array
  items:
    type: string
  description: Дополнительные примечания

AnalysisResponse:
  type: object
  required:
    [ parsed_ingredients, active_count, warnings, overall_verdict, summary ]
  properties:
    parsed_ingredients:
      type: array
      items:
        $ref: "/components/schemas/Ingredient"
      description: Распознанные ингредиенты
    active_count:
      type: number
      description: Количество активных компонентов
    warnings:
      type: array
      items:
        type: string
      description: Замечания/риски
    overall_verdict:
      type: string
      description: Итоговый вывод
    summary:
      type: string
      description: Краткое резюме анализа
```

```
ValidationError:
  type: object
  properties:
    detail:
      type: array
      description: Детали ошибки валидации входных данных
      items:
        type: object
        additionalProperties: true

ErrorMessage:
  type: object
  properties:
    detail:
      type: string
      description: Сообщение об ошибке
```

10 Критерии приемки

Общие критерии

- Пользователь может отправить текст состава средства для волос в Telegram-бот.
- Текст состава должен быть длиной не более 4000 символов.
- При успешной отправке текста бот возвращает анализ состава с категориями и эффектами ингредиентов.
- Анализ включает распознанные ингредиенты, их категории, эффекты.
- Если часть компонентов не найдена в каталоге, бот уведомляет пользователя о частичном результате.
- Время обработки запроса не превышает 60 секунд.

Технические критерии

- API принимает POST запросы с текстом состава по endpoint `/analyze/text`.
- Валидация входных данных проверяет длину и минимальное количество ингредиентов.
- Результат анализа возвращается в формате JSON с четкой структурой.

10 Нефункциональные требования

NFR-PERF-001: Производительность обработки запросов

Описание: Система должна обеспечивать быстрый отклик на запросы анализа состава.

Критерии измерения:

- Время отклика API: не более 60 секунд (99-й перцентиль)
- Максимальная длина текста: 4000 символов
- Обработка не менее 100 запросов в час

Условия измерения:

- Среда: сервер с 4 CPU, 8 GB RAM
- Нагрузка: до 100 одновременных пользователей

Инструменты: Apache JMeter, Prometheus

Приоритет: Высокий

Обоснование: Быстрая обработка повышает удовлетворенность пользователей

NFR-SEC-001: Безопасность данных пользователей

Описание: Система должна обеспечивать защиту персональных данных пользователей.

Критерии измерения:

- Хранение user_telegram_id и данных анализа в зашифрованном виде
- Доступ к данным только авторизованным сервисам
- Использование HTTPS для всех внешних API вызовов
- Логирование доступа и изменений данных

Условия тестирования:

- Инструменты: OWASP ZAP, аудит безопасности

Приоритет: Критический

Обоснование: Защита персональных данных обязательна по законодательству

NFR-REL-001: Надежность и отказоустойчивость

Описание: Система должна обеспечивать высокую доступность и устойчивость к сбоям.

Критерии измерения:

- Доступность сервиса не менее 99.9% в месяц
- Время восстановления после сбоя не более 15 минут
- Обработка ошибок с информированием пользователя

Условия тестирования:

- Мониторинг uptime, тесты отказоустойчивости

Приоритет: Критический

Обоснование: Стабильная работа важна для доверия пользователей

NFR-SCAL-001: Масштабируемость

Описание: Система должна масштабироваться для поддержки роста числа пользователей.

Критерии измерения:

- Горизонтальное масштабирование backend сервисов
- Автоматическое добавление ресурсов при нагрузке CPU > 70%
- Балансировка нагрузки с отклонением не более 10%

Условия тестирования:

- Нагрузочное тестирование с постепенным увеличением пользователей

Приоритет: Средний

Обоснование: Поддержка роста аудитории без деградации качества

NFR-USAB-001: Удобство использования

Описание: Взаимодействие с ботом должно быть интуитивным и понятным.

Критерии измерения:

- Время отклика бота на сообщения не более 5 секунд (за исключением анализа)
- Понятные сообщения об ошибках и подсказки
- Поддержка основных сценариев без сбоев

Условия тестирования:

- Тестирование с реальными пользователями

Приоритет: Высокий

Обоснование: Удобство повышает вовлеченность и лояльность

NFR-COMP-001: Совместимость и интеграция

Описание: Система должна корректно работать с Telegram API и внешними сервисами.

Критерии измерения:

- Поддержка актуальной версии Telegram Bot API
- Корректная обработка всех типов сообщений, кроме фото (текст только)

Условия тестирования:

- Тестирование интеграции с Telegram API

Приоритет: Высокий

Обоснование: Корректная интеграция обеспечивает стабильность работы