

Бизнес-требования

1. Название
2. User Story
3. Use Case
4. Макет
5. BPMN

Функциональные требования

1. Архитектура
2. Модель данных
3. ER-диаграмма
4. Диаграмма последовательности
5. REST
6. Swagger

Критерии приемки и нефункциональные требования

1. Критерии приемки
2. Нефункциональные требования

HedgeAiCare: Умная забота о колючем друге в Telegram🦊

User Story

Я, как владелец ежа, хочу получать от ИИ-агента в Telegram экспертные консультации по содержанию, питанию и гигиене моего питомца, основанные на его индивидуальных данных (порода, возраст, вес), чтобы я мог обеспечить ежу максимально качественный уход в повседневной жизни.

Use Case

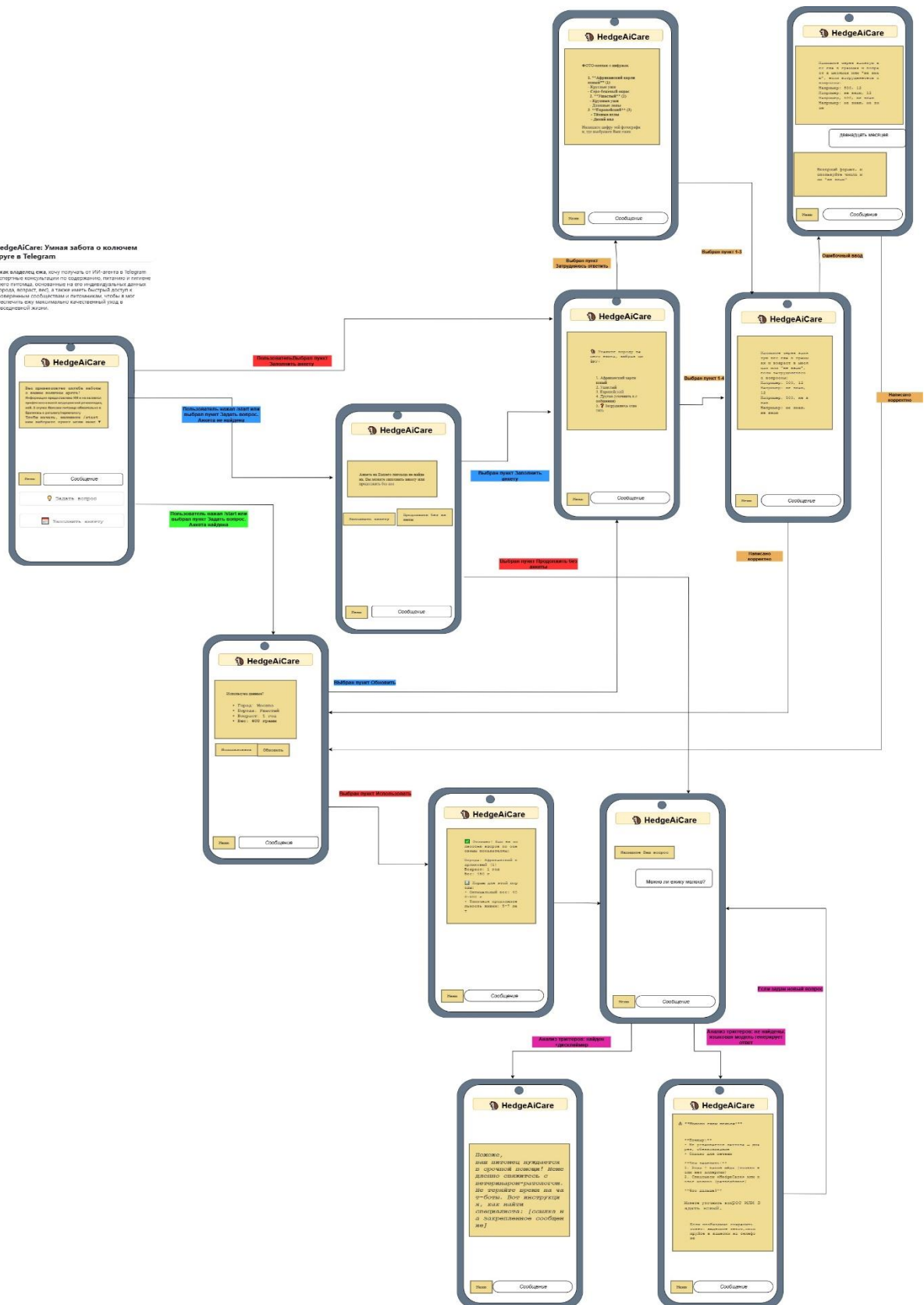
Наименование структуры	Описание
Заголовок	Консультация по уходу за домашним ежом
Акторы	Основной: Пользователь Telegram (владелец ежа); Вспомогательные: 1. ИИ-агент (LLM API);
Предусловие	У клиента установлен Telegram
Ограничение	- Только РФ (акцент на русскоязычные ресурсы); - Язык: русский; - Контент с дисклеймером: <i>"Информация предоставлена ИИ и не является профессиональной медицинской рекомендацией. В случае болезни питомца обязательно обратитесь к ратологу/герпетологу."</i>
Триггер	Пользователь запускает бота командой /start или выбирает один из пунктов меню
Основной сценарий	<div>1. Отображение приветственного сообщения с дисклеймером</div> <div>- Важность обращения к специалисту (ратологу/герпетологу) в случае наличия симптомов нездоровья у ежика.</div> <div>- Краткое объяснение, как искать врача.</div> <div>- Кнопки меню: "Задать вопрос" (аналог команды /start), «Заполнить анкету»</div> <div>2. Пользователь нажимает " Задать вопрос "</div> <div>- Бот проверяет наличие анкеты;</div> <div>- Анкета обнаружена, система показывает ее, уточняет инлайном «Использовать», «Обновить»;</div> <div>-Пользователь выбирает «Использовать»;</div> <div>- Система просит ввести вопрос в свободной форме;</div> <div>- Пользователь пишет вопрос (напр., "как купать ежа?");</div> <div>- Система анализирует вопрос (и данные анкеты, если она заполнена) с помощью LLM формирует и отправляет развернутый, персонализированный ответ.</div> <div>3. Пользователь выбирает "Обновить" или пункт меню "Заполнить анкету"</div> <div>-отображается анкета</div> <div>• 🦊🦊 Порода (выпадающий список: африканский, ушастый,</div>

Наименование структуры	Описание
	<p>европейский и др.);</p> <ul style="list-style-type: none"> • 📅 Возраст (в месяцах); • ⚖️ Вес (в граммах); <p>-Пользователь заполняет анкету;</p> <p>- Система показывает ее, уточняет инлайном «Верно», «Внести изменения»;</p> <p>-Пользователь выбирает «Верно» и далее следует алгоритму из пункта 2 пп.4;</p> <p>-Пользователь выбирает «Внести изменения» и далее следует алгоритму из пункта 3 пп.1;</p>
Альтернативный сценарий	<p>2а. Анкета не заполнена</p> <p>- При первом запросе система может предложить сначала заполнить анкету (порода, возраст, вес) для персонализации ответов, но не блокирует возможность задать вопрос без нее</p>
Исключительный сценарий	<p>2б. Ошибки ввода в анкете</p> <ul style="list-style-type: none"> • Некорректные данные (вес/возраст, город) → Система выдает сообщение: "Проверьте данные и введите повторно" <p>Результат: Пользователь не заполняет анкету, пока не введёт корректный ответ</p> <p>3б. Вопрос содержит описание критических симптомов</p> <p>- Система определяет ключевые слова ("кровь", "судороги", "не дышит").</p> <p>- Система игнорирует запрос на консультацию и показывает экранное сообщение-алерт:</p> <p>"Похоже, ваш питомец нуждается в срочной помощи! Немедленно свяжитесь с ветеринаром-ратологом. Не теряйте время на чат-боты. Вот инструкция, как найти специалиста: [ссылка на закрепленное сообщение]".</p>
Результат	<p>Пользователь получает:</p> <ul style="list-style-type: none"> - Персонализированную консультацию по уходу от агента - Четкое понимание, когда необходимо обращаться к врачу.

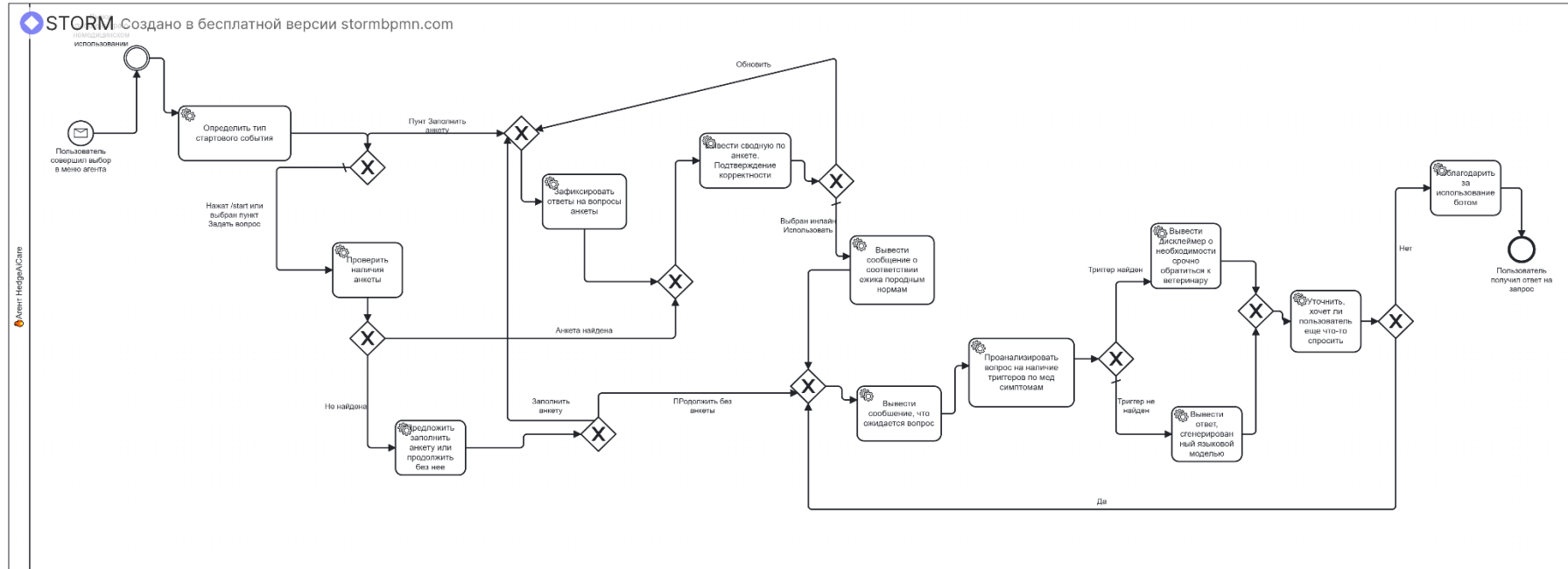
Maker

HedgeAiCare: Умная забота о колючем друге в Telegram

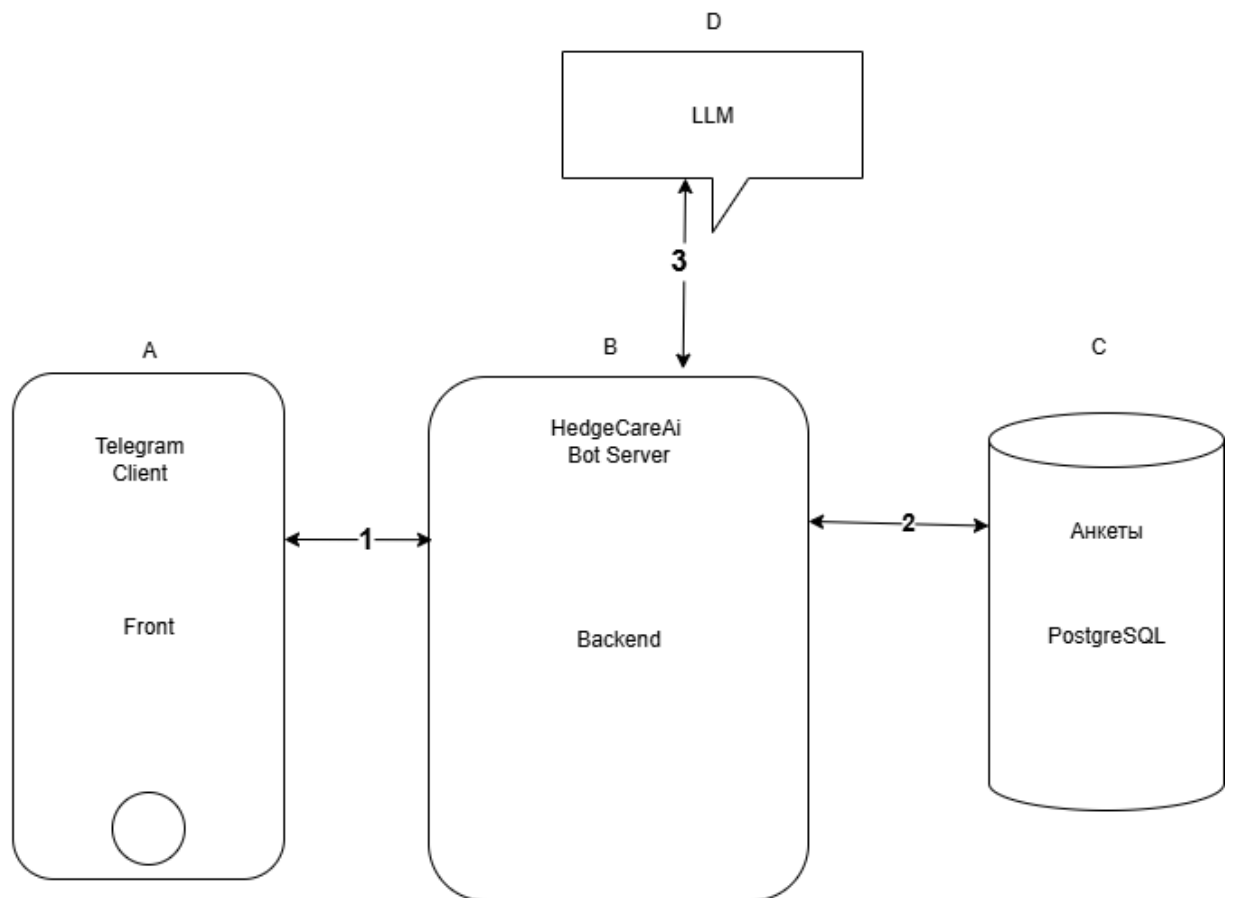
Как владелец ежа, хочу получать от бота-агента в Telegram инструкции по уходу за ежом, советы, рецепты и многое другое. Хотел бы, чтобы бот мог анализировать фотографии ежа и давать рекомендации по уходу, а также иметь доступ к базам данных о ежах и их особенностях, чтобы я мог обеспечить моему ежечеловеку наилучший уход и комфортную жизнь.



BPMN



Архитектура



- A. Frontend Приложение Telegram на устройстве пользователя
B. Backend Программа на Python для обработки запросов, ответа пользователю
C. База данных PostgreSQL. Хранит анкеты ежей
D. API LLM

Протокол взаимодействия:

1. https, telegram bot API
2. PostgreSQL wire protocol
3. https, REST API

Модель данных

Родительская сущность	Атрибут	Описание
Users (Пользователь Telegram)	user_id	Уникальный идентификатор пользователя в Telegram. Главный ключ.
	user_name	@username пользователя.
	first_name	Имя пользователя. Для вежливого обращения.
	created_at	Дата и время регистрации. Для аналитики и аудита.
HedgehogProfiles (Анкеты ежа)	profile_id	Уникальный идентификатор анкеты. Главный ключ.
	user_id	Ссылка на User. Чья это анкета. Внешний ключ.
	hedgehog_breed	Порода ежа (например, "Африканский карликовый").
	hedgehog_age	Возраст в месяцах.
	hedgehog_weight	Вес в граммах.
	last_updated	Дата и время последнего обновления. Чтобы проверять актуальность.

ERD – диаграмма

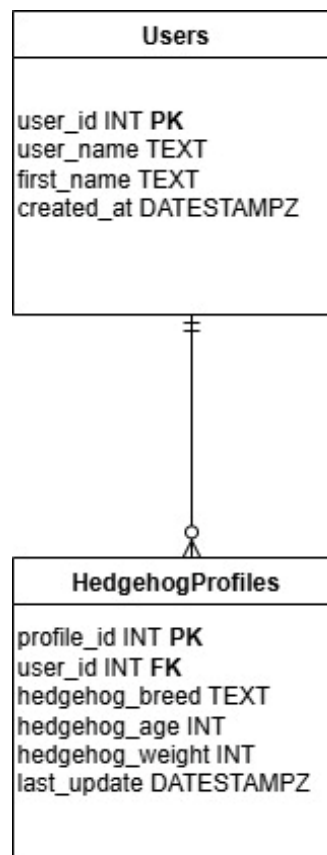
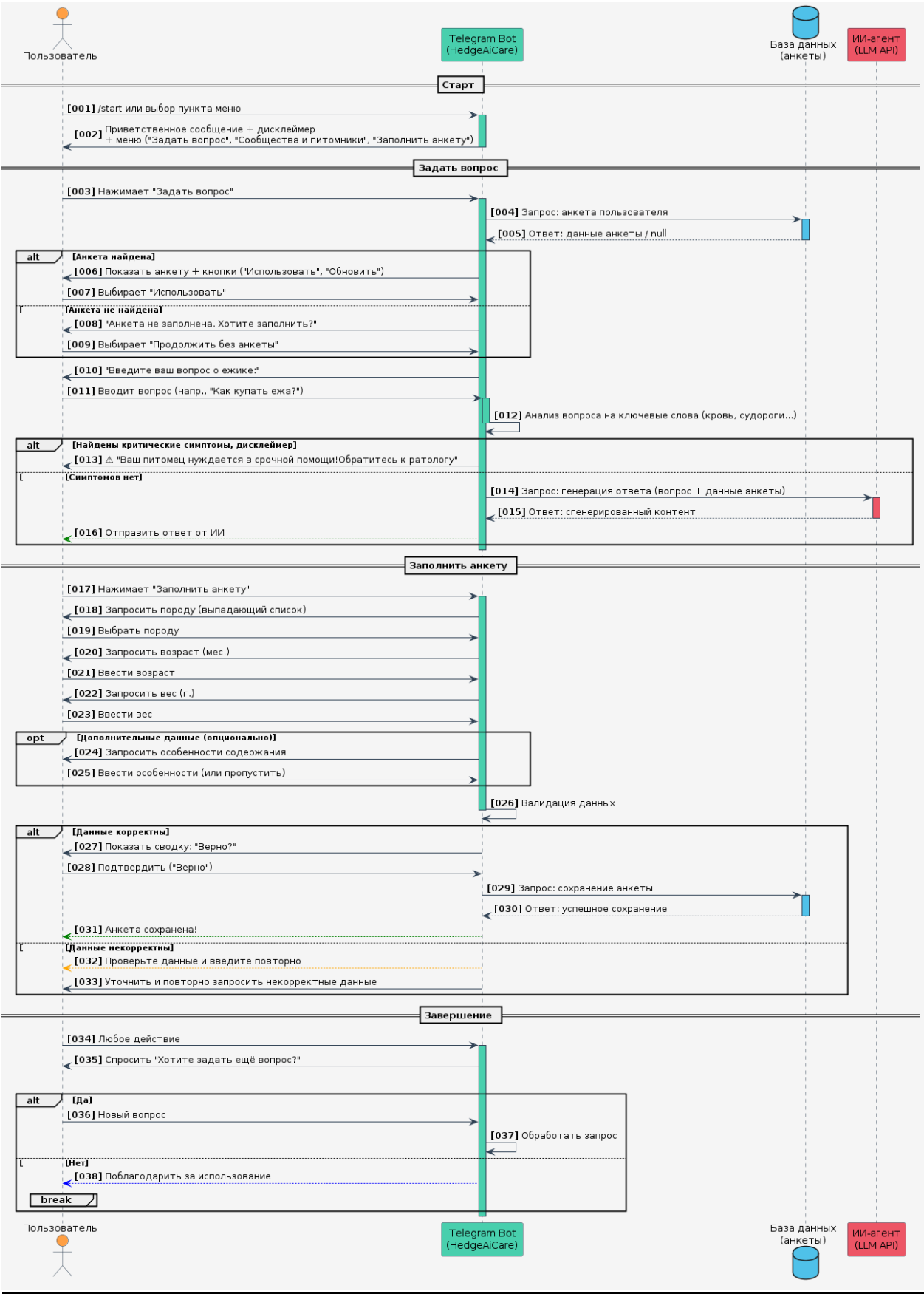


Диаграмма последовательности



REST

1. Получение данных о ежике пользователя (проверка наличия анкеты)

GET /v1/users/{userId}/hedgehog-profile

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
userId	int	path	Уникальный идентификатор пользователя в Telegram	да

Response code <200>

Название параметра	Тип данных	Находится в	Описание	Обязательность
userId	int	body	ID пользователя, которому принадлежит анкета	да
profile	object	body	Объект анкеты. null, если анкета не найдена.	Да
profile.breed	string	body	Порода ежа (напр., "африканский")	Нет
profile.age	int	body	Возраст в месяцах	Нет
profile.weight	int	body	Вес в граммах	Нет
lastUpdatedAt	datetime	body	Дата и время последнего обновления анкеты	Нет

Response code <404> - анкета для данного пользователя не найдена. Body пустое.

2. Создание или обновление анкеты ежика

PUT /v1/users/{userId}/hedgehog-profile

Используем при заполнении анкеты в пункте меню "Заполнить анкету" или выборе "Обновить".

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
userId	int	path	Уникальный идентификатор пользователя в Telegram	да
breed	string	body	Порода ежа	Да
age	int	body	Возраст в месяцах	Да
weight	int	body	Вес в граммах	Да

Response code <200> Успешное сохранение. Возвращает обновлённый профиль.

Название параметра	Тип данных	Находится в	Описание	Обязательность
profile	object	body	Обновлённые данные анкеты	да

Response code <400> Ошибка валидации

Название параметра	Тип данных	Находится в	Описание	Обязательность
error	string	body	Сообщение об ошибке (напр., "Проверьте данные и введите повторно")	Да
invalidField	string	body	Поле, которое не прошло валидацию	Нет

3. Отправка вопроса и получение консультации ИИ

Основной запрос для взаимодействия с LLM. Бот отправляет сюда вопрос пользователя и (если есть) данные анкеты.

POST /v1/users/{userId}/consultations

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
userId	int	path	ID пользователя, задающего вопрос	Да

question	string	body	Текст вопроса от пользователя	Да
hedgehogProfile	object	body	Объект с данными анкеты для контекста	Нет

Response code <200> — Успешный ответ ИИ (Обычный вопрос)

Вопрос не связан с медициной или болезнями (уход, питание, гигиена, поведение).

Название параметра	Тип данных	Находится в	Описание	Обязательность
answer	string	body	Текст ответа, сгенерированный LLM	Да
questionType	string	body	Тип классификации вопроса general	Да

Response code <200> — Успешный ответ ИИ (Обычный вопрос)

Вопрос касается здоровья, симптомов или лечения, но они **НЕ критические** (например, "еж чихает", "плохо ест", "вялый"). Ответ сопровождается дисклеймером

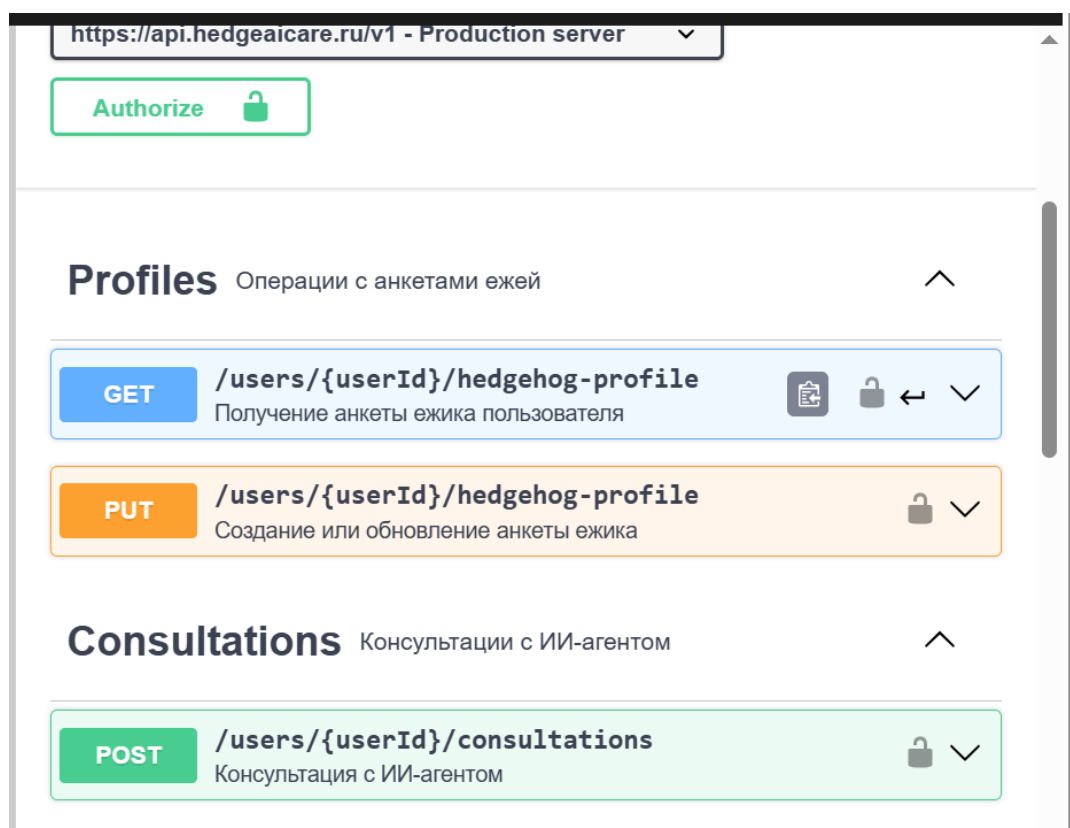
Название параметра	Тип данных	Находится в	Описание	Обязательность
answer	string	body	Текст ответа, сгенерированный LLM, начинающийся с дисклеймера	Да
disclaimer	string	body	Текст дисклеймера, который можно показать отдельно	Да
questionType	string	body	Тип классификации вопроса medical	Да

Response code <422> Критический симптом обнаружен

Вопрос содержит упоминания критических, угрожающих жизни симптомов ("кровь", "судороги", "не дышит", "лежит без движения"). Ответ не генерируется.

Название параметра	Тип данных	Находится в	Описание	Обязательность
error	string	body	Сообщение-алерт для немедленного показа пользователю	Да
questionType	string	body	Тип классификации вопроса critical	Да
code	string	body	Код ошибки: critical_symptoms_detected	Да

Swagger



openapi: 3.0.3

info:

title: HedgeAiCare API

description: API для умного помощника по уходу за ежами в Telegram

version: 1.0.0

contact:

name: HedgeAiCare Team

email: support@hedgeaicare.ru

servers:

- url: https://api.hedgeaicare.ru/v1

description: Production server

tags:

- name: Profiles

description: Операции с анкетами ежей

- name: Consultations

description: Консультации с ИИ-агентом

- name: Communities

description: Сообщества и питомники

paths:

/users/{userId}/hedgehog-profile:

get:

tags:

- Profiles

summary: Получение анкеты ежика пользователя
description: Проверка наличия и получение данных анкеты ежика
parameters:
- name: userId
in: path
required: true
schema:
type: integer
minimum: 1
description: Уникальный идентификатор пользователя в Telegram

responses:
'200':
description: Анкета найдена
content:
application/json:
schema:
\$ref: '#/components/schemas/ProfileResponse'
'404':
description: Анкета не найдена
content:
application/json:
schema:
\$ref: '#/components/schemas/Error'
'500':
description: Internal server error
content:
application/json:
schema:
\$ref: '#/components/schemas/Error'

put:
tags:
- Profiles
summary: Создание или обновление анкеты ежика
description: Сохранение данных анкеты при заполнении или обновлении
parameters:
- name: userId
in: path
required: true
schema:
type: integer
minimum: 1
description: Уникальный идентификатор пользователя в Telegram
requestBody:
required: true
content:
application/json:
schema:
\$ref: '#/components/schemas/HedgehogProfileInput'
responses:
'200':
description: Анкета успешно сохранена

```
content:
  application/json:
    schema:
      $ref: '#/components/schemas/ProfileResponse'
'400':
  description: Ошибка валидации данных
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/ValidationError'
'500':
  description: Internal server error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Error'
```

/users/{userId}/consultations:

post:

tags:

- Consultations

summary: Консультация с ИИ-агентом

description: |

Отправка вопроса и получение консультации с учетом данных анкеты.

Вопросы с критическими симптомами возвращают ошибку 422.

parameters:

- name: userId

in: path

required: true

schema:

type: integer

minimum: 1

description: ID пользователя, задающего вопрос

requestBody:

required: true

content:

application/json:

schema:

type: object

required:

- question

properties:

question:

type: string

description: Текст вопроса от пользователя

example: "как купать ежа?"

hedgehogProfile:

\$ref: '#/components/schemas/HedgehogProfile'

responses:

'200':

description: Успешный ответ ИИ

content:

application/json:
 schema:
 oneOf:
 - \$ref: '#/components/schemas/GeneralResponse'
 - \$ref: '#/components/schemas/MedicalResponse'
 discriminator:
 propertyName: questionType
'422':
 description: Обнаружены критические симптомы
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/CriticalResponse'
'500':
 description: Internal server error
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Error'

/communities:
 get:
 tags:
 - Communities
 summary: Получение списка сообществ и питомников
 description: Возвращает проверенные сообщества и питомники для ежей
 responses:
 '200':
 description: Список сообществ
 content:
 application/json:
 schema:
 type: object
 properties:
 communities:
 type: array
 items:
 \$ref: '#/components/schemas/Community'
 '500':
 description: Internal server error
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Error'

components:
 schemas:
 HedgehogProfileInput:
 type: object
 required:
 - breed
 - age

- weight
properties:
breed:
 type: string
 enum: ["африканский", "ушастый", "европейский", "другой"]
 example: "африканский"
 description: Порода ежа
age:
 type: integer
 minimum: 1
 maximum: 84
 description: Возраст в месяцах (максимум 7 лет)
 example: 12
weight:
 type: integer
 minimum: 100
 maximum: 800
 description: Вес в граммах
 example: 450

HedgehogProfile:
type: object
properties:
 breed:
 type: string
 example: "африканский"
 description: Порода ежа
 age:
 type: integer
 description: Возраст в месяцах
 example: 12
 weight:
 type: integer
 description: Вес в граммах
 example: 450

ProfileResponse:
type: object
properties:
 userId:
 type: integer
 description: ID пользователя, которому принадлежит анкета
 example: 123456789
 profile:
 type: object
 nullable: true
 properties:
 breed:
 type: string
 example: "африканский"
 age:
 type: integer

description: Возраст в месяцах
example: 12
weight:
type: integer
description: Вес в граммах
example: 450
lastUpdatedAt:
type: string
format: date-time
example: "2024-01-15T14:30:00Z"

GeneralResponse:

type: object
properties:
answer:
type: string
description: Текст ответа, сгенерированный LLM для общих вопросов
example: "Для купания ежа используйте теплую воду (около 35°C) и специальный шампунь для грызунов или ежей. Нанесите небольшое количество шампуня, аккуратно помассируйте и тщательно смойте. После купания заверните ежа в мягкое полотенце и осторожно промокните шерсть."
questionType:
type: string
enum: ["general"]
description: Общий вопрос по уходу, питанию, гигиене
example: "general"

MedicalResponse:

type: object
properties:
answer:
type: string
description: |
Текст ответа, сгенерированный LLM. Ответ НАЧИНАЕТСЯ с текста
дисclaimer о том,
что это не медицинская рекомендация, а затем следует основной текст.
example: "Информация предоставлена ИИ и не является профессиональной
медицинской рекомендацией. В случае болезни питомца обязательно обратитесь к
ратологу/герпетологу. Чихание у ежа может быть вызвано различными причинами:
аллергия, простуда или раздражение от наполнителя. Понаблюдайте за общим
состоянием питомца."
disclaimer:
type: string
description: Текст дисclaimer, который можно показать отдельно
example: "Информация предоставлена ИИ и не является профессиональной
медицинской рекомендацией. В случае болезни питомца обязательно обратитесь к
ратологу/герпетологу."
questionType:
type: string
enum: ["medical"]
description: Вопрос о неопасных симптомах и здоровье (с дисclaimerом)
example: "medical"

CriticalResponse:

type: object

properties:

error:

type: string

description: Сообщение-алерт для немедленного показа пользователю

example: "Похоже, ваш питомец нуждается в срочной помощи! Немедленно свяжитесь с ветеринаром-ратологом. Не теряйте время на чат-боты."

questionType:

type: string

enum: ["critical"]

description: Обнаружены критические симптомы (ответ не сгенерирован)

example: "critical"

code:

type: string

example: "critical_symptoms_detected"

Community:

type: object

required:

- name

- link

- type

properties:

name:

type: string

example: "Сообщество любителей ежей"

link:

type: string

format: uri

example: "https://t.me/hedgehog_lovers"

type:

type: string

enum: ["community", "nursery"]

example: "community"

description:

type: string

example: "Крупное сообщество с полезными советами по уходу за ежами"

verified:

type: boolean

example: true

Error:

type: object

properties:

error:

type: string

example: "Произошла внутренняя ошибка сервера"

code:

type: string

example: "internal_server_error"

ValidationError:
type: object
properties:
 error:
 type: string
 example: "Проверьте данные и введите повторно"
 code:
 type: string
 example: "validation_error"
 invalidField:
 type: string
 example: "weight"
 details:
 type: string
 example: "Вес должен быть между 100 и 800 граммами"

securitySchemes:
 ApiKeyAuth:
 type: apiKey
 in: header
 name: X-API-Key

security:
 - ApiKeyAuth: []

Критерии приемки

АС-1: Создание анкеты ежа

- **Дано:** Пользователь нажимает "✚ Добавить ежика"
- **Когда:** Пользователь последовательно заполняет кличку, породу, возраст, вес и подтверждает данные
- **Тогда:** Система сохраняет анкету в БД и показывает подтверждение

АС-2: Редактирование анкеты

- **Дано:** Пользователь просматривает список анкет и выбирает "✎ Редактировать"
- **Когда:** Пользователь изменяет одно или несколько полей и подтверждает изменения
- **Тогда:** Система обновляет анкету в БД и показывает подтверждение

АС-3: Персонализированная консультация

- **Дано:** Пользователь выбирает ежа из списка и вводит вопрос
- **Когда:** Система обрабатывает вопрос через LLM API с данными анкеты
- **Тогда:** Пользователь получает ответ, учитывающий особенности его питомца

АС-4: Обнаружение критических симптомов

- **Дано:** Пользователь вводит вопрос с описанием критических симптомов ("кровь", "судороги" и т.д.)
- **Когда:** Система классифицирует вопрос как "critical"

- **Тогда:** Показывается экстренное предупреждение с рекомендацией немедленно обратиться к ветеринару

АС-5: Медицинский вопрос

- **Дано:** Пользователь вводит вопрос о здоровье ежа
- **Когда:** Система классифицирует вопрос как "medical"
- **Тогда:** Ответ сопровождается дисклеймером о необходимости консультации специалиста

Нефункциональные требования

Производительность

- Время отклика бота: < 3 секунд для 95% запросов
- Время генерации ответа LLM: < 30 секунд
- Время классификации вопроса: < 5 секунд
- Поддержка одновременных пользователей: до 1000

Надежность

- Доступность системы: 99% в рабочее время (09:00-21:00 по МСК)
- Время восстановления: не более 15 минут при сбоях
- Резервное копирование: ежедневное автоматическое

Безопасность

- Аутентификация: через Telegram User ID
- Валидация входных данных: проверка всех пользовательских вводов
- Защита персональных данных: шифрование чувствительной информации
- Ограничение запросов: не более 10 запросов в минуту на пользователя

Масштабируемость

- Архитектура: модульная, с возможностью горизонтального масштабирования
- База данных: репликация для чтения
- Кэширование: кэш частых запросов к LLM API