

# Техническое задание на разработку

## Содержание

### Пакет Бизнес-требований

1. Название продукта
2. User Story
3. Use Case
4. Wireframe
5. Activity диаграмма

### Пакет функциональных требований

1. Архитектура
2. Модель данных
3. ER диаграмма
4. Sequence диаграмма
5. REST. Табличный вид
6. Swagger

### Критерии приемки и нефункциональные требования

1. Критерии приемки
2. Нефункциональные требования

## Пакет Бизнес-требований

### 1. Название продукта

Telegram-бот “@medcoder\_bot”

### 2. User Story

Как врач, я хочу вводить клинический диагноз, симптомы и результаты исследований, чтобы получать список релевантных кодов МКБ-10 для документирования случая.

### 3. Use Case

ID: UC-MEDCODER-01

Версия: 2.0

Автор: Алексей Голицын

#### 1. Краткое описание

Система автоматически преобразует текстовые медицинские данные (жалобы, диагнозы, процедуры), введенные пользователем в Telegram-боте, в соответствующие коды МКБ-10 с помощью LLM DeepSeek. Результат возвращается в формате структурированного списка с указанием точности соответствия.

#### 2. Бизнес-контекст

**Проблема:**

Врачи тратят до 30% времени на ручное кодирование медицинских данных. Ошибки в кодах МКБ-10 приводят к отказам в страховых выплатах (до 15% случаев).

**Решение:**

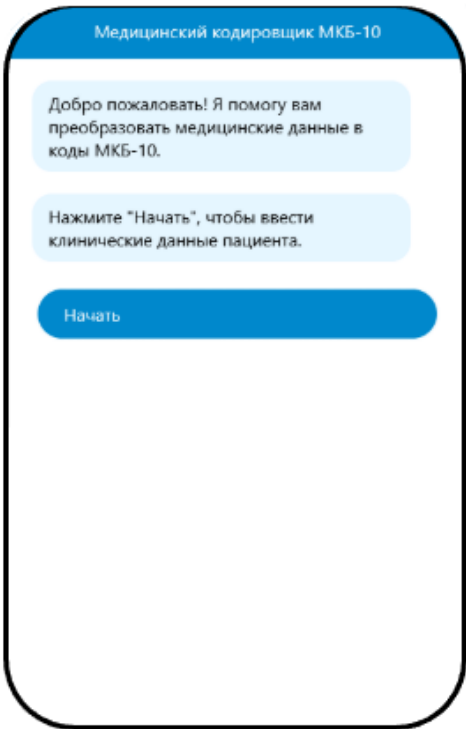
Автоматизация процесса через Telegram-бота с точностью >90%.  
Интеграция с DeepSeek LLM для обработки естественного языка.

Заголовок	Определение кодов МКБ в присланных данных
Акторы	<div>1. Пользователь</div> <div>2. Telegram-bot</div> <div>3. База МКБ-10 с RAG</div> <div>4. DeepSeek LLM</div>
Предусловие	<div>1. Пользователь авторизован в Telegram.</div> <div>2. Бот активирован и подключен к API DeepSeek.</div> <div>3. База данных МКБ-10 инициализирована</div>
Ограничения	<div>1. Точность зависит от полноты входных данных</div> <div>2. Время обработки не более 30 секунд</div>
Триггер	Пользователь запускает бота командой /start или нажав на кнопку «Старт» приветственного экрана бота
Основной сценарий	<div>1. Система отображает приветственный экран</div> <div>2. Пользователь отправляет текстовое сообщение с медицинским описанием</div> <div>3. Система присылает ответ в виде найденного в базе</div>

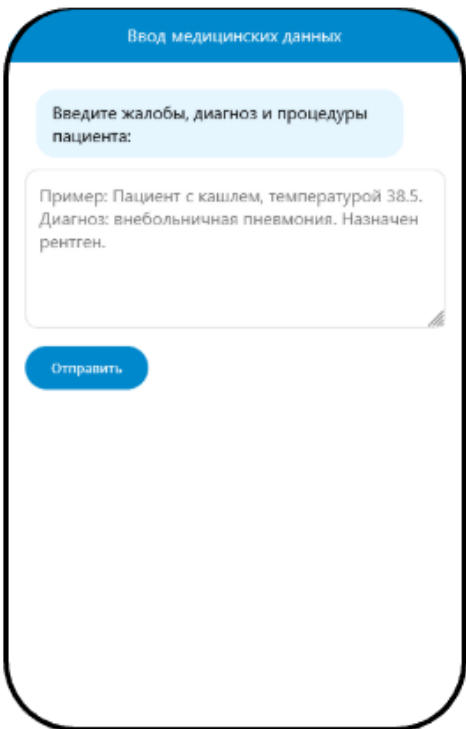
	<p>4. кода МКБ-10 и Ответа на основе анализа LLM Deepseek</p> <p>5. <b>Критерий успеха:</b> Система успешно выполняет шаг 3</p>
<b>Альтернативные сценарии</b>	<p><b>A.</b> Поиск запросов через команду</p> <ol style="list-style-type: none"> <li>1. Пользователь отправляет запрос через команду /search</li> <li>2. Система повторяет шаги 2-3 основного сценария</li> </ol> <p><b>B.</b> Просмотр истории запросов</p> <ol style="list-style-type: none"> <li>1. Пользователь использует команду /history или нажимает кнопку “История запросов”</li> <li>2. Система: Формирует интерактивное меню с пагинацией</li> <li>3. Пользователь: Выбирает конкретный запрос для просмотра деталей</li> <li>4. Система: Отображает детальную информацию о запросе и найденных кодах</li> </ol> <p><b>C.</b> Просмотр статистики</p> <ol style="list-style-type: none"> <li>1. Пользователь использует команду /stats или нажимает кнопку “Статистика”</li> <li>2. Система отображает статистику использования бота</li> <li>4.</li> </ol>
<b>Исключительный сценарий</b>	<p><b>D.</b> Некорректный запрос</p> <ol style="list-style-type: none"> <li>1. Пользователь вводит немедицинские данные/бессмысленный набор символов/эмодзи</li> <li>2. Система отвечает: «Релевантные коды МКБ не найдены. Пожалуйста, пришлите медицинские данные»</li> </ol>

## 4. Wireframe

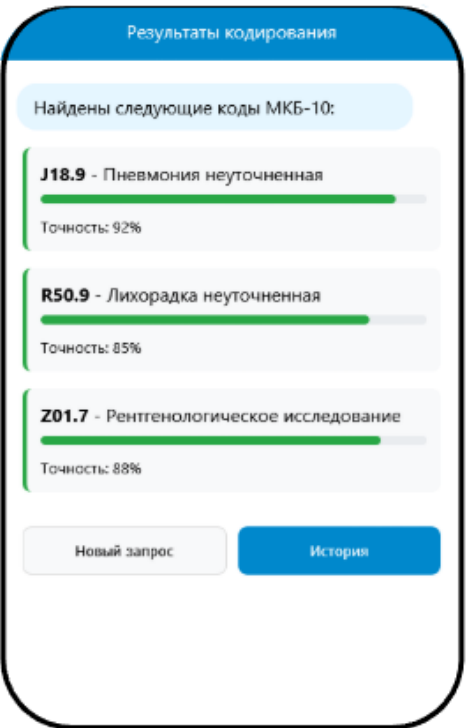
1. Стартовый экран бота



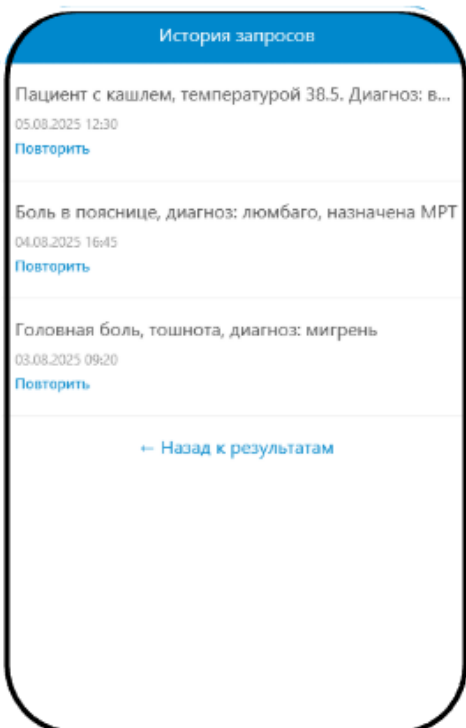
2. Экран с полем ввода запроса



3. Экран ответа на запрос, переход в историю

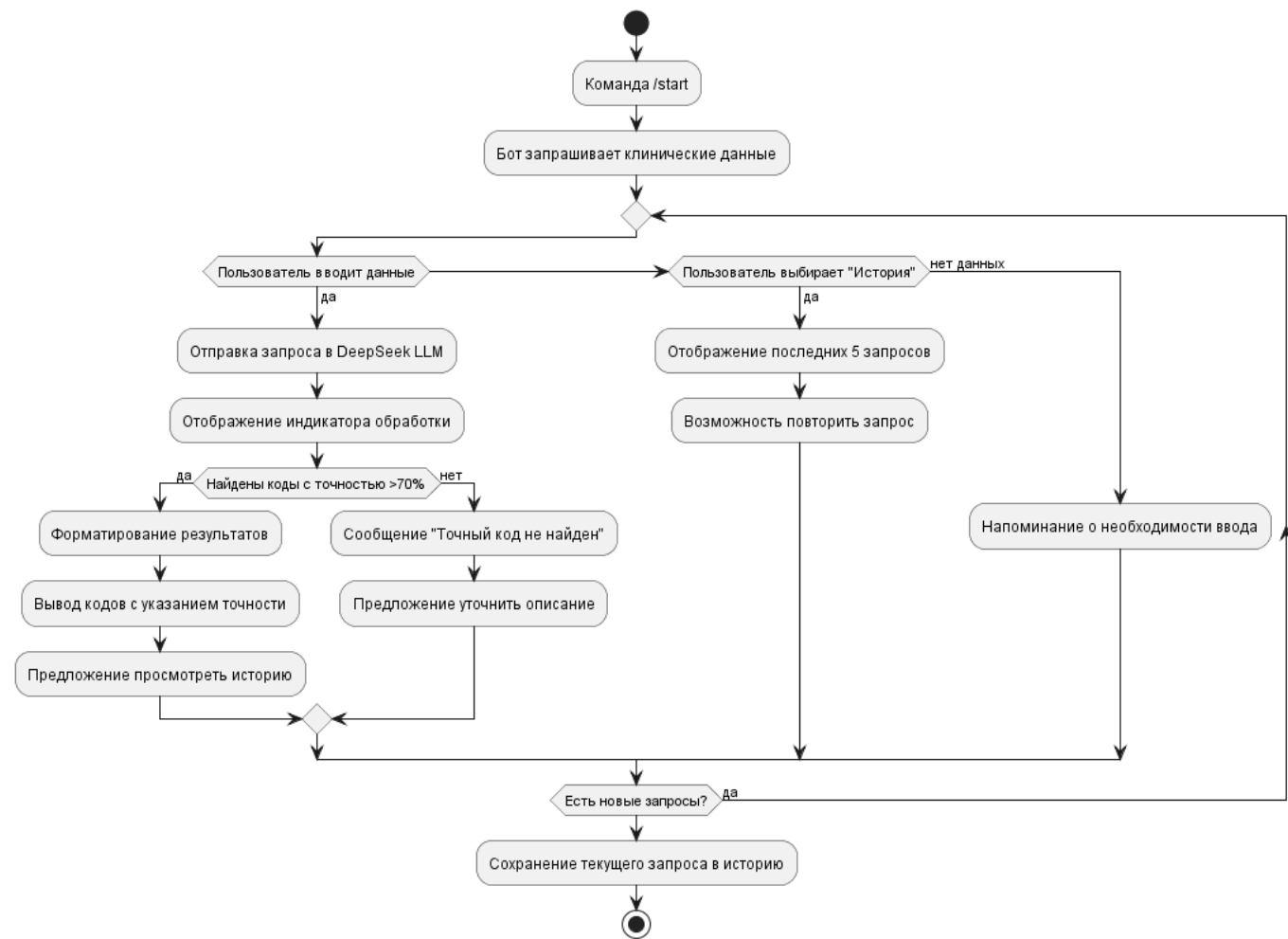


4. Экран истории запросов



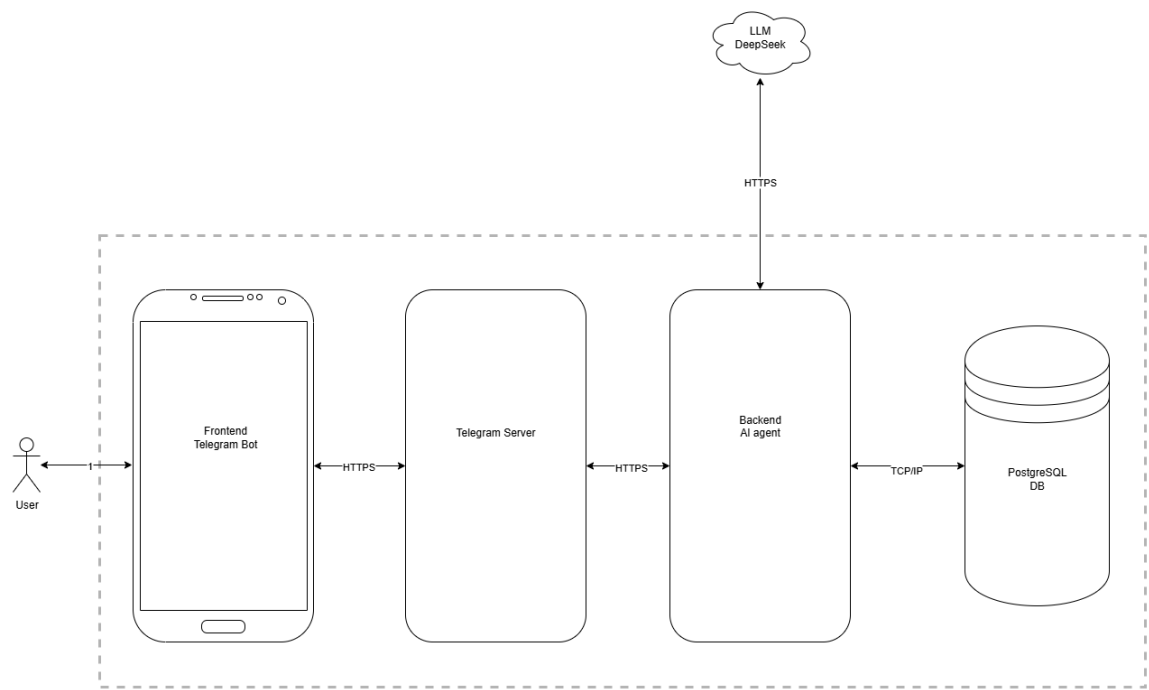
# 5. Activity-диаграмма

Диаграмма активности для медицинского кодировщика МКБ-10



# Пакет функциональных требований

## 1. Архитектура



## 2. Модель данных

### Объект Users

Родительская сущность	Атрибут	Описание
User	---	Объект пользователя
	telegram_id	Уникальный идентификатор в Телеграм
	telegram_name	Ник пользователя в Телеграмм

### Объект Requests

Родительская сущность	Атрибут	Описание
Requests	---	Объект запросов пользователя
	user_id	Связь с пользователем
	input_text	Запрос в свободной форме
	processed_text	Предобработанный бэком текст
	request_data	Дата создания

	status	Статус запроса (создан, обработан, выполнен)
--	--------	--

#### Объект Answers

Родительская сущность	Атрибут	Описание
Answers	---	Объект ответов для пользователя
	request_id	Связь с запросом
	code_id	Связь с кодом
	desription	Описание
	confidence	Точность кода в %
	status	статус ответа (создан, обработан, доставлен)
	completion_date	Дата создания

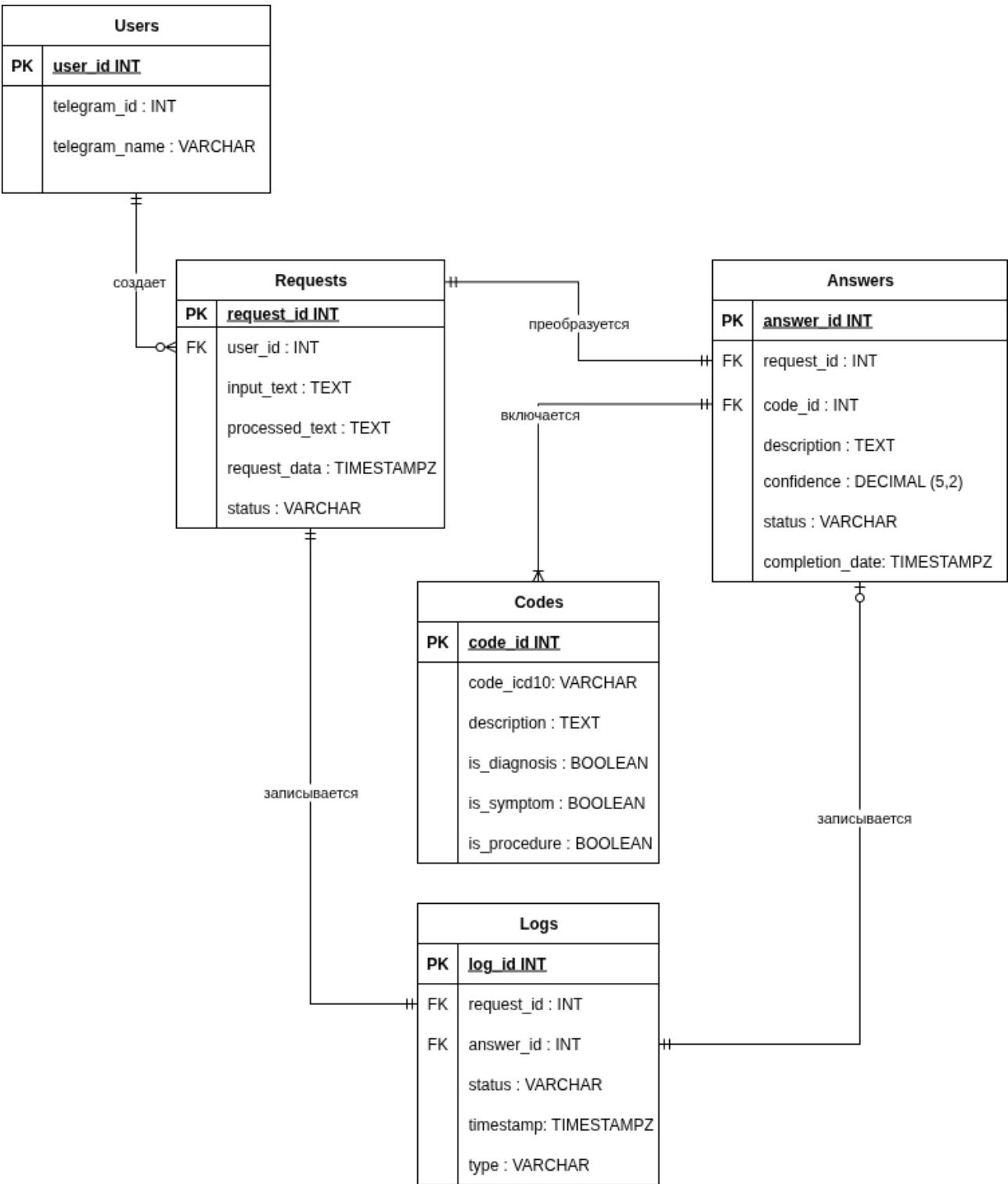
#### Объект Codes

Родительская сущность	Атрибут	Описание
Logs	---	Объект логов запросов
	request_id	Связь с запросом
	answer_id	Связь с ответом
	status	Статус : создан, обработан, доставлен
	timestamp	Время записи лога
	type	Тип: информационный, предупреждение, ошибка

#### Объект Logs

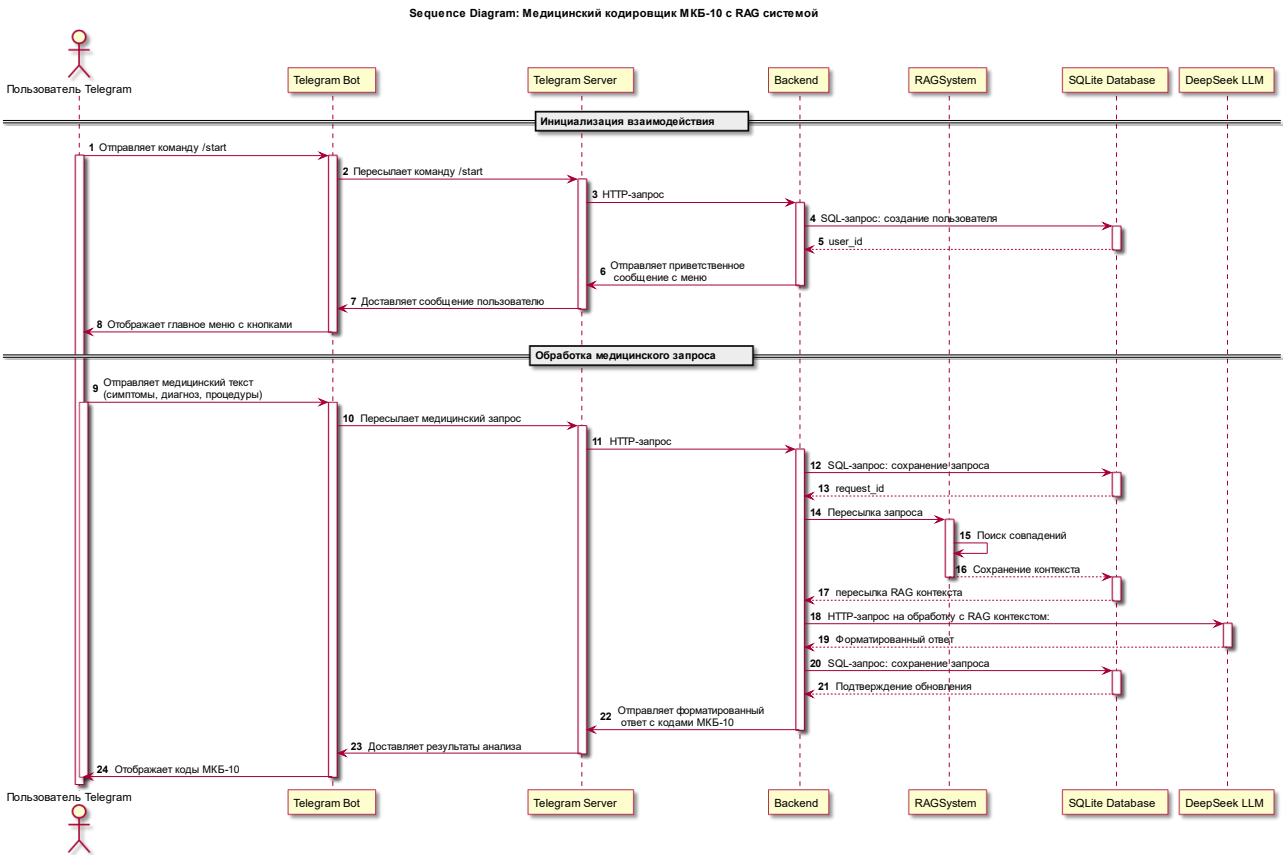
Родительская сущность	Атрибут	Описание
Logs	---	Объект логов запросов
	request_id	Связь с запросом
	answer_id	Связь с ответом
	status	Статус : создан, обработан, доставлен
	timestamp	Время записи лога
	type	Тип: информационный, предупреждение, ошибка

### 3. ER диаграмма





4. Sequence-диаграмма



№	Описание
1	Открытие пользователем бота в ТГ и нажатие на кнопку "Старт" в интерфейсе бота или написание пользователем сообщения "/start"
2	Бот направляет команду ТГ серверу
3	HTTP Запрос наличие пользователя
4	SQL-запрос создания нового пользователя
5	БД возвращает id пользователя
6	Backend отправляет приветственное сообщение и меню
7	ТГ сервер доставляет сообщение боту
8	Бот отображает сообщение и меню с кнопками
9	Пользователь вводит и отправляет медицинский текст
10	Пересылка запроса пользователя на сервер ТГ
11	HTTP POST запрос, содержащий запрос пользователя

12	SQL-запрос на сохранение запроса в БД
13	БД возвращает id запроса на Backend
14	Backend направляет запрос в RAG
15	Поиск совпадений по базе данных в RAG
16	SQL запрос для сохранения RAG контекста
17	Пересылка RAG контекста на Backend
18	HTTP-запрос в LLM с указанием RAG контекста
19	LLM выделяет сущности (жалобы, диагноз, процедура) и ищет коды в базе МКБ-10, формирует и направляет ответ на Backend
20	SQL-запрос на сохранение текущего запроса пользователя и ответа в историю
21	БД подтверждает обновление
22	Отправка сформированного ответа на ТГ сервер
23	ТГ сервер пересылает ответ боту
24	Бот отображает сообщение пользователю

## 5. REST. Табличный вид

REST API подход использует HTTP-методы (GET, POST, PUT, DELETE и т.д.) для управления сущностями с уникальными URL.

*POST /chat/completions*

Запрос к LLM

### Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
Authorization: Bearer <your_llm_api_key>	string	headers	Обязательный заголовок для аутентификации в LLM сервисе	да
Content-Type: application/json	string	headers	Указывает, что тело запроса представлено в формате JSON	да
model	string	body	Идентификатор модели LLM ("DeepSeek-V3")	да
messages	array	body	Массив сообщений, представляющих историю диалога	да
role	string	body	Роль отправителя сообщения (user, assistant, system)	да

<i>content</i>	string	body	Текст сообщения или промпт для генерации	да
<i>temperature</i>	number	body	Параметр, контролирующий случайность ответов (0.0-2.0)	нет
max_tokens	integer	body	Максимальное количество токенов для генерации в ответе. Ограничивает длину выходного текста	нет
stream	boolean	body	Указание, будет ли ответ приходить частями (поток)	нет

## Response

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
id	string	body	Уникальный идентификатор ответа от API (например: "chatcmpl-123")	да
model	string	body	Идентификатор модели, сгенерировавшей ответ	да
choices	array	body	Массив с результатами генерации	да
message	object	body	Объект, содержащий сгенерированный ответ	да
role	string	body	Роль создателя сообщения. Всегда "assistant" для сгенерированного ответа.	да
content	string	body	Основной результат запроса — сгенерированный LLM ответ. Содержит коды МКБ-10 с указанием точности.	да
finish_reason	string	body	Причина завершения генерации ("stop", "length", "content_filter")	нет
usage	object	body	Информация об использовании токенов	нет
prompt_tokens	integer	body	Количество токенов в промте	нет
completion_tokens	integer	body	Количество токенов в ответе	нет
total_tokens	integer	body	Общее количество токенов	нет

Response code <200>

GET /getUpdates

Запрос новых сообщений от Telegram Bot API

## Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
Authorization: Bearer <your_bot_token>	string	headers	Обязательный заголовок для аутентификации в Telegram Bot API	да
offset	integer	query	Идентификатор первого обновления, которое должно быть возвращено	нет
limit	integer	query	Максимальное количество обновлений для возврата (1-100)	нет
timeout	integer	query	Таймаут в секундах для long polling	нет
allowed_updates	array	query	Типы обновлений для получения	нет

**Response**

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
ok	boolean	body	Статус выполнения запроса	да
result	array	body	Массив объектов Update	да
update_id	integer	body	Уникальный идентификатор обновления	да
message	object	body	Объект сообщения (если есть)	нет
message_id	integer	body	Уникальный идентификатор сообщения	да (если есть message)
from	object	body	Информация об отправителе	да (если есть message)
id	integer	body	Уникальный идентификатор пользователя	да
is_bot	boolean	body	Является ли отправитель ботом	да
first_name	string	body	Имя пользователя	да
chat	object	body	Информация о чате	да (если есть message)
id	integer	body	Уникальный идентификатор чата	да
type	string	body	Тип чата ("private", "group", "supergroup", "channel")	да
date	integer	body	Дата отправки сообщения (Unix timestamp)	да (если есть message)
text	string	body	Текст сообщения	нет

Response code <200>

GET /users/{telegram\_id}/logs

Запрос истории по конкретному пользователю

**Request**

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
Authorization: Bearer <your_api_key>	string	headers	Обязательный заголовок для аутентификации	да
telegram_id	integer	path	Уникальный идентификатор пользователя в Telegram	да
page	integer	query	Номер страницы для пагинации (начиная с 1)	нет
limit	integer	query	Количество записей на странице (1-100)	нет
from_date	string	query	Начальная дата для фильтрации (формат: YYYY-MM-DD)	нет
to_date	string	query	Конечная дата для фильтрации (формат: YYYY-MM-DD)	нет

**Response**

Название	Тип	Находится	Описание	Обязательность
----------	-----	-----------	----------	----------------

параметра	данных	в		параметра
telegram_id	integer	body	Уникальный идентификатор пользователя в Telegram	да
logs	array	body	Массив объектов истории пользователя	да
log_id	integer	body	Уникальный идентификатор записи в истории	да
request_id	integer	body	Уникальный идентификатор запроса пользователя	да
answer_id	integer	body	Уникальный идентификатор ответа пользователю	да
request_text	string	body	Текст исходного запроса пользователя	да
response_text	string	body	Текст ответа системы	да
status	string	body	Статус обработки запроса ("success", "partial", "error")	да
accuracy	number	body	Точность определения кода МКБ-10 (0-100)	нет
codes	array	body	Массив найденных кодов МКБ-10 с точностью	нет
code	string	body	Код МКБ-10 (например: "J18.9")	да (если есть codes)
accuracy	number	body	Точность соответствия кода (0-100)	да (если есть codes)
timestamp	string	body	Временная метка создания записи (ISO 8601 format: YYYY-MM-DDTHH:MM:SSZ)	да
type	string	body	Тип медицинской информации ("diagnosis", "procedure", "symptom")	да
pagination	object	body	Информация о пагинации	да
total_records	integer	body	Общее количество записей	да
current_page	integer	body	Текущая страница	да
total_pages	integer	body	Общее количество страниц	да
has_next	boolean	body	Есть ли следующая страница	да
has_prev	boolean	body	Есть ли предыдущая страница	да

Response code <200>

POST /sendMessage

Отправка сообщения через Telegram Bot API

#### Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
Authorization: Bearer <your_bot_token>	string	headers	Обязательный заголовок для аутентификации	да
Content-Type: application/json	string	headers	Указывает JSON формат тела запроса	да
chat_id	integer	body	Уникальный идентификатор чата	да

text	string	body	Текст сообщения для отправки	да
parse_mode	string	body	Режим парсинга текста ("Markdown", "HTML", "MarkdownV2")	нет
reply_markup	object	body	Объект для кнопок	нет
disable_web_page_preview	boolean	body	Отключить предпросмотр веб-страниц	нет

## Response

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
ok	boolean	body	Статус выполнения запроса	да
result	object	body	Объект отправленного сообщения	да
message_id	integer	body	Уникальный идентификатор сообщения	да
from	object	body	Информация об отправителе (бота)	да
id	integer	body	Уникальный идентификатор бота	да
is_bot	boolean	body	Всегда true	да
first_name	string	body	Имя бота	да
chat	object	body	Информация о чате	да
id	integer	body	Уникальный идентификатор чата	да
type	string	body	Тип чата	да
date	integer	body	Дата отправки сообщения (Unix timestamp)	да
text	string	body	Текст отправленного сообщения	да

Response code <200>

# 6. Swagger

Yaml

## Medical ICD-10 Coding Bot API 1.0.0 OAS 3.0

<https://gorbe.io/api/openapi.yaml>

Telegram bot for medical text analysis and ICD-10 code classification

Servers

[https://api.telegram.org/bot{BOT\\_TOKEN}](https://api.telegram.org/bot{BOT_TOKEN}) - Telegram Bot API

### Medical Coding ICD-10 code classification endpoints

### User Management User data and history management

### Search ICD-10 database search functionality

### default

POST

/start

Start bot and show main menu

Initializes user registration and displays main navigation menu

Parameters

Try it out

No parameters

Responses

Curl

curl -X 'POST' \n'https://api.telegram.org/bot578807\_TOKEN637D/start' \n-H 'accept: \*/\*' \n-d ''

Request URL

https://api.telegram.org/bot578807\_TOKEN637D/start

Server response

Code	Details
404	Error: response status is 404

Undocumented

Response body

{\n "ok": false,\n "error\_code": 404,\n "description": "Not Found"\n}

Response headers

content-length: 55\ncontent-type: application/json\ndate: Tue, 30 Sep 2025 03:20:33 GMT\nserver: nginx/1.18.0

Responses

Code	Description	Links
200	Welcome message with inline keyboard	No links

POST

/help

Show help information

Displays bot usage instructions and examples

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Help text with navigation buttons	No links

POST

/history

Get user request history

^

Retrieves user's previous medical coding requests with pagination

Parameters

Try it out

Name	Description
offset	Pagination offset
integer	
(query)	<input type="text" value="offset"/>


Responses

Code	Description	Links
200	List of previous requests with details	No links

POST

/stats

Get user statistics

 ← ^

Shows user activity statistics and coding metrics

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Statistics including request count, codes found, average confidence	No links

POST

/search

Search ICD-10 codes by keywords

^

Direct search in ICD-10 database using RAG system

Parameters

Try it out

Name	Description
query	Medical keywords to search
string	
(query)	<input type="text" value="query"/>

Responses

Code	Description	Links
200	List of relevant ICD-10 codes with similarity scores	No links



POST

/handle\_message

Process medical text and return ICD-10 codes

^

Main endpoint for medical text analysis using AI and RAG

Parameters

Try it out

No parameters

Request body

required

text/plain

^

Example Value

Schema

пациент с кашлем и температурой

Responses

Code	Description	Links
200	AI analysis results with relevant ICD-10 codes and confidence percentages	No links

Schemas

^

User

user\_id

>

[...]

telegram\_id

>

[...]

telegram\_name

>

[...]

registration\_date

>

[...]

}

Request

request\_id

>

[...]

user\_id

>

[...]

input\_text

>

[...]

processed\_text

>

[...]

status

>

[...]

request\_date

>

[...]

}

ICD10Code

code\_icd10

>

[...]

description

>

[...]

confidence

>

[...]

}

RAGResult

code

>

[...]

description

>

[...]

similarity

>

[...]

}

AIResponse

original\_query

>

[...]

rag\_context

>

[...]

ai\_analysis

>

[...]

formatted\_codes

>

[...]

}

# Критерии приемки и нефункциональные требования

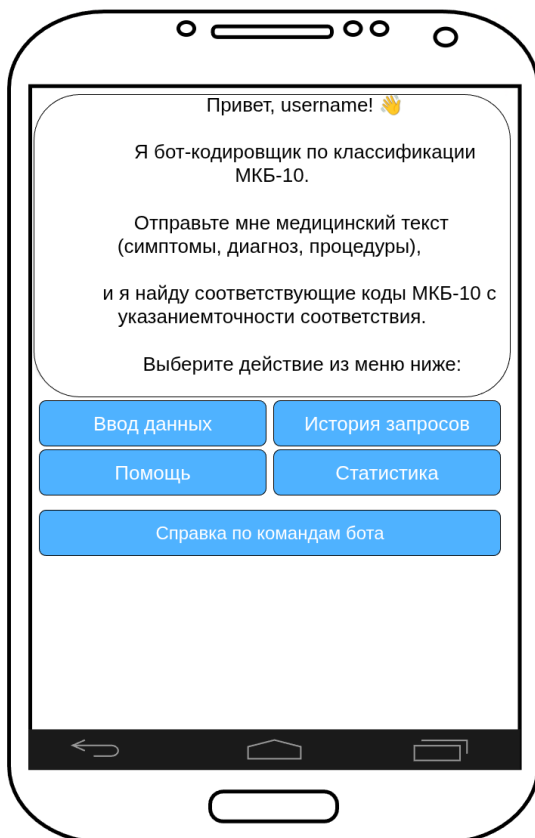
## Критерии приемки

### User story:

Как врач, я хочу вводить клинический диагноз, симптомы и результаты исследований, чтобы получать список релевантных кодов МКБ-10 для точного документирования случая.

### Номер кейса: 1.

#### 1. Стартовый экран бота



**Функциональность:** Просмотр истории запросов

**Дано:** Пользователь находится на Экране 1: Стартовый экран

**Когда:** Пользователь нажимает кнопку "История", но запросов еще не было

**Тогда:** Получает ответ: "У вас еще не было запросов. Отправьте медицинский текст, чтобы начать"

## Номер кейса: 2.

### 1. Стартовый экран бота



**Функциональность:** Просмотр истории запросов

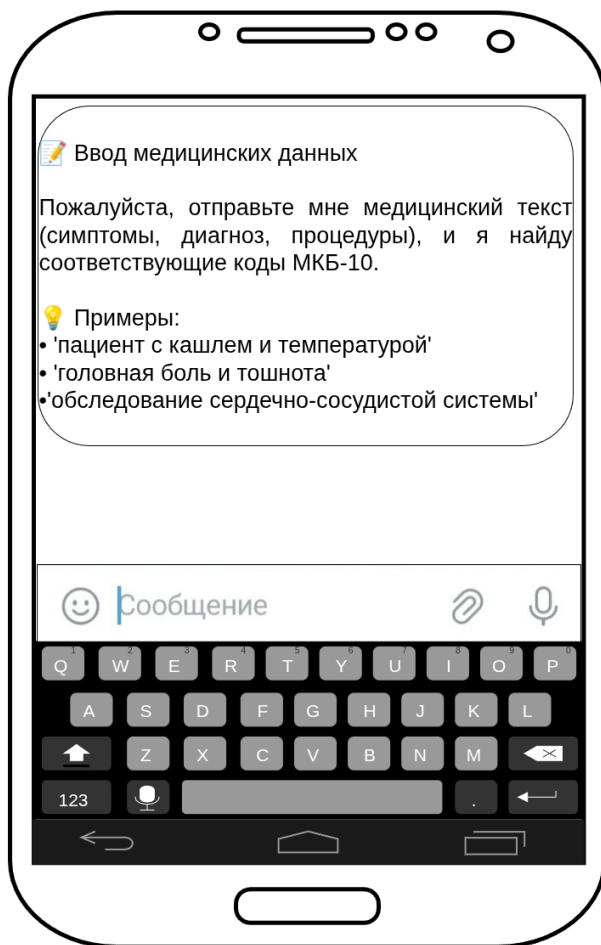
**Дано:** Пользователь находится на Экране 1: Стартовый экран

**Когда:** Пользователь нажимает кнопку "Статистика", но запросов еще не было

**Тогда:** Получает ответ: "У вас еще не было запросов. Отправьте медицинский текст, чтобы начать"

### Номер кейса: 3.

#### 2. Экран с полем ввода запроса



**Функциональность:** Ввод медицинских данных.

**Дано:** Пользователь находится на Экране 2: "Экран с полем ввода запроса"

**Когда:** Пользователь вносит некорректные данные (не медицинские или набор символов)

**Тогда:** Система отвечает: "Пожалуйста, пришлите медицинские данные для кодирования"

## **Нефункциональные требования**

### **Требования производительности:**

1. Время обработки запроса: не более 30 секунд (включая вызов AI API)
2. Время отклика Telegram бота: не более 2 секунд для командных операций
3. Время загрузки истории запросов: не более 3 секунд при 1000 записей

### **Требования надёжности:**

1. Система должна быть доступна 99% времени в месяц
2. Обработка ошибок API: автоматические повторные попытки (до 3 раз)

### **Требования безопасности:**

1. Защита API ключей: хранение в переменных окружения
2. Административный доступ только для указанных Telegram ID

### **Удобство использования:**

1. Простота использования: не более 3 шагов для основных операций
2. Обработка длинных сообщений: автоматическое разбиение на части
3. Подсказки и справка: команда /help с подробной информацией