

ОГЛАВЛЕНИЕ

Техническое задание на разработку

Бизнес-требования

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

Функциональные требования

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

Критерии приемки и нефункциональные требования

1. Критерии приемки
2. Нефункциональные требования

Бизнес-требования:

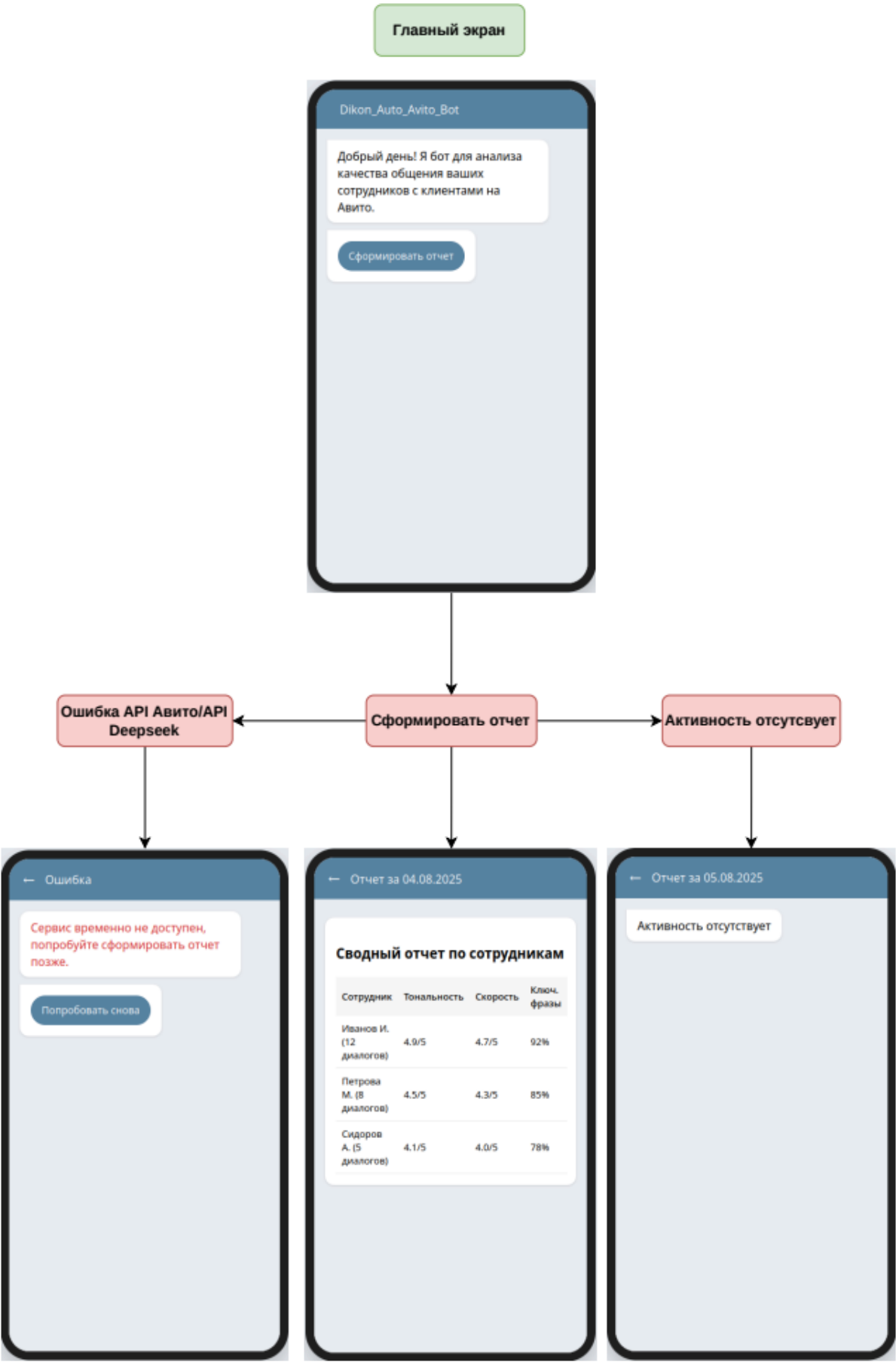
1. Название продукта

Мониторинг качества общения с клиентами на Авито

2. User Story

Как владелец компании, продающей товары на Авито,
я хочу получать ежедневные отчеты в Telegram-боте с оценкой общения сотрудников с клиентами, чтобы контролировать качество обслуживания, тем самым повышая лояльность клиентов к компании

3. Макеты

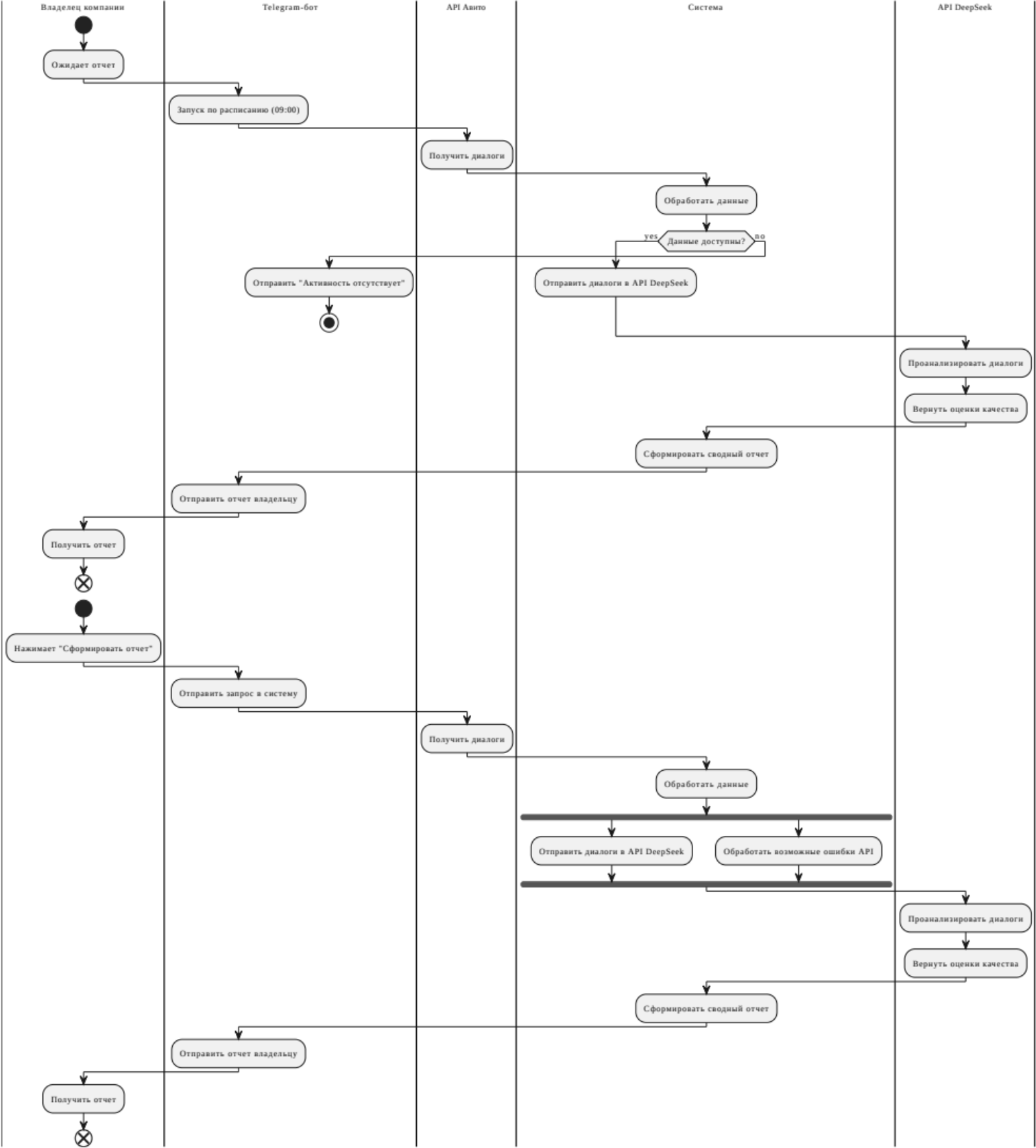


4. User Case

User_Case - 001	
Заголовок	Получение отчета о качестве общения с клиентами
Актор	Владелец компании (Пользователь)
Предусловие	Пользователь подписан на Telegram-бота
Ограничения	
Триггер	1. Наступает время отправки отчета (Автоматический) 2. Пользователь нажимает кнопку "Сформировать отчет" в Telegram-боте (Ручной)
Основной сценарий	1. Система запускается по расписанию (раз в 24 часа) 2. Система получает данные диалогов сотрудников с клиентами через API Авито 3. Система отправляет тексты диалогов на анализ через API DeepSeek 4. API DeepSeek возвращает оценки качества общения по каждому диалогу 5. Система формирует сводный отчет с оценками по сотрудникам 6. Telegram-бот отправляет отчет владельцу компании 7. Владелец компании получает отчет в установленное время
Альтернативный сценарий	п.2: Если не было диалогов за период → отчет содержит сообщение "Активность отсутствует" п.3: Если API Авито недоступно → отчет содержит сообщение "Сервис временно не доступен, попробуйте сформировать отчет позже" п.4: Если API Deepseek недоступно → отчет содержит сообщение "Сервис временно не доступен, попробуйте сформировать отчет позже"
Сценарий по требованию	1. Владелец компании нажимает кнопку "Сформировать отчет" в Telegram-боте 2. Бот отправляет запрос в систему 3. Система выполняет шаги 2-6 основного сценария 4. Владелец компании получает отчет по требованию
Постусловие	Отчет успешно доставлен пользователю

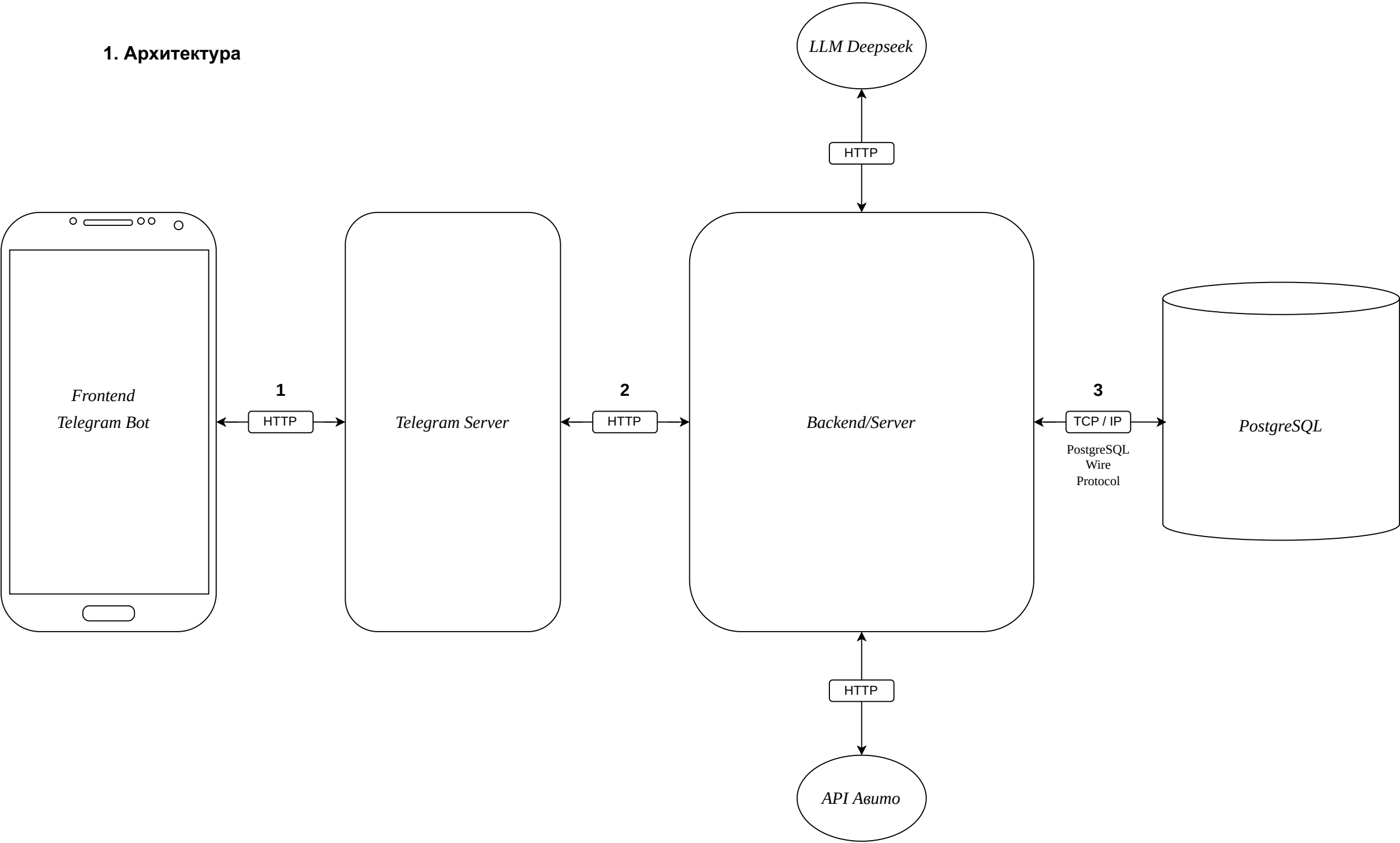
5.Activity

Процесс формирования отчета о качестве общения с клиентами (US-001, UC-001)



Функциональные требования:

1. Архитектура



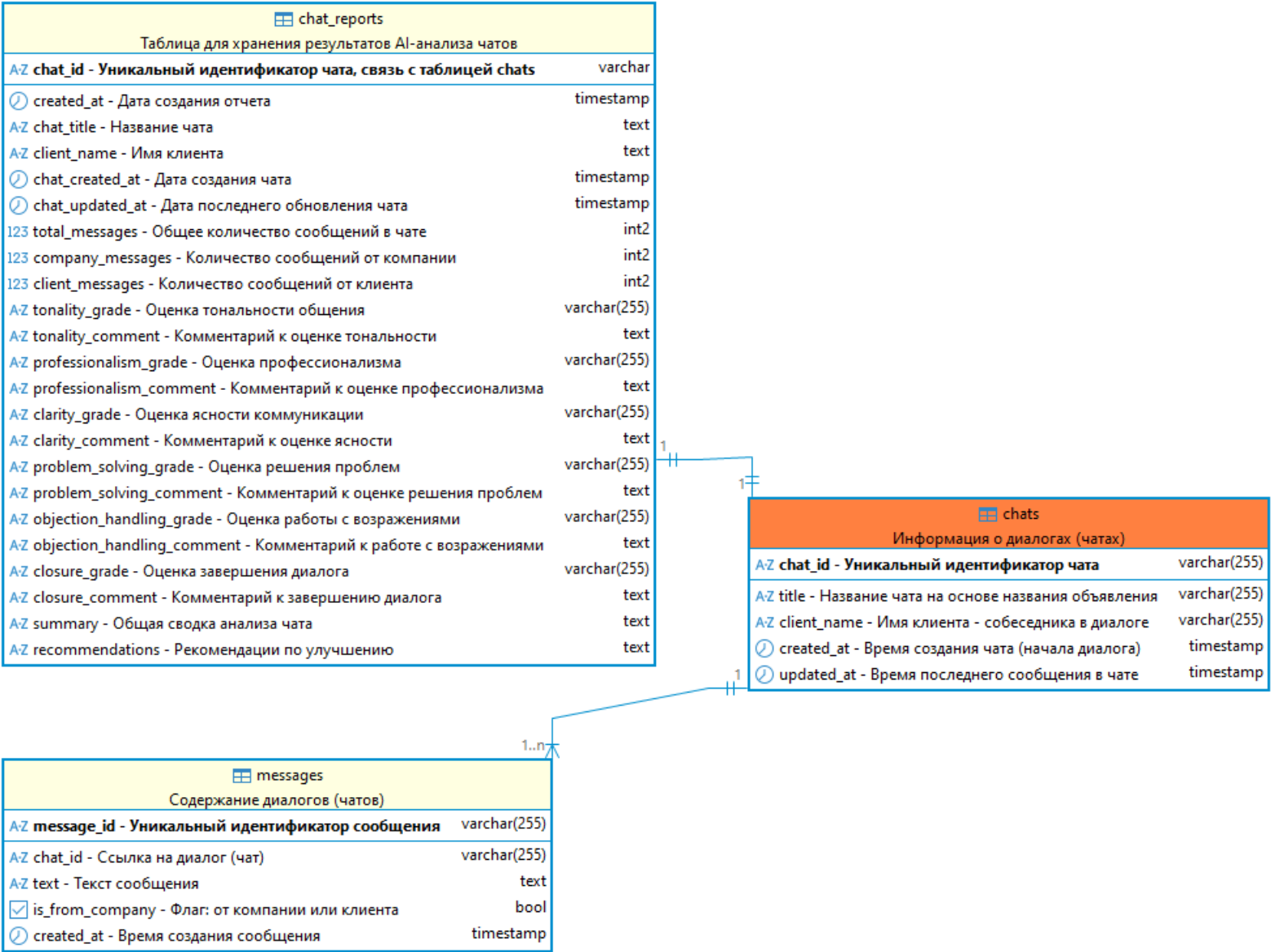
2. Модель данных

Сущность	Атрибут	Тип данных	Описание
chats	-	-	Информация о диалогах (чатах).
	chat_id	VARCHAR (PK)	Уникальный идентификатор чата.
	title	VARCHAR	Название чата на основе названия объявления.
	client_name	VARCHAR	Имя клиента - собеседника в диалоге.
	created_at	TIMESTAMP	Время создания чата (начала диалога).
	updated_at	TIMESTAMP	Время последнего сообщения в чате.

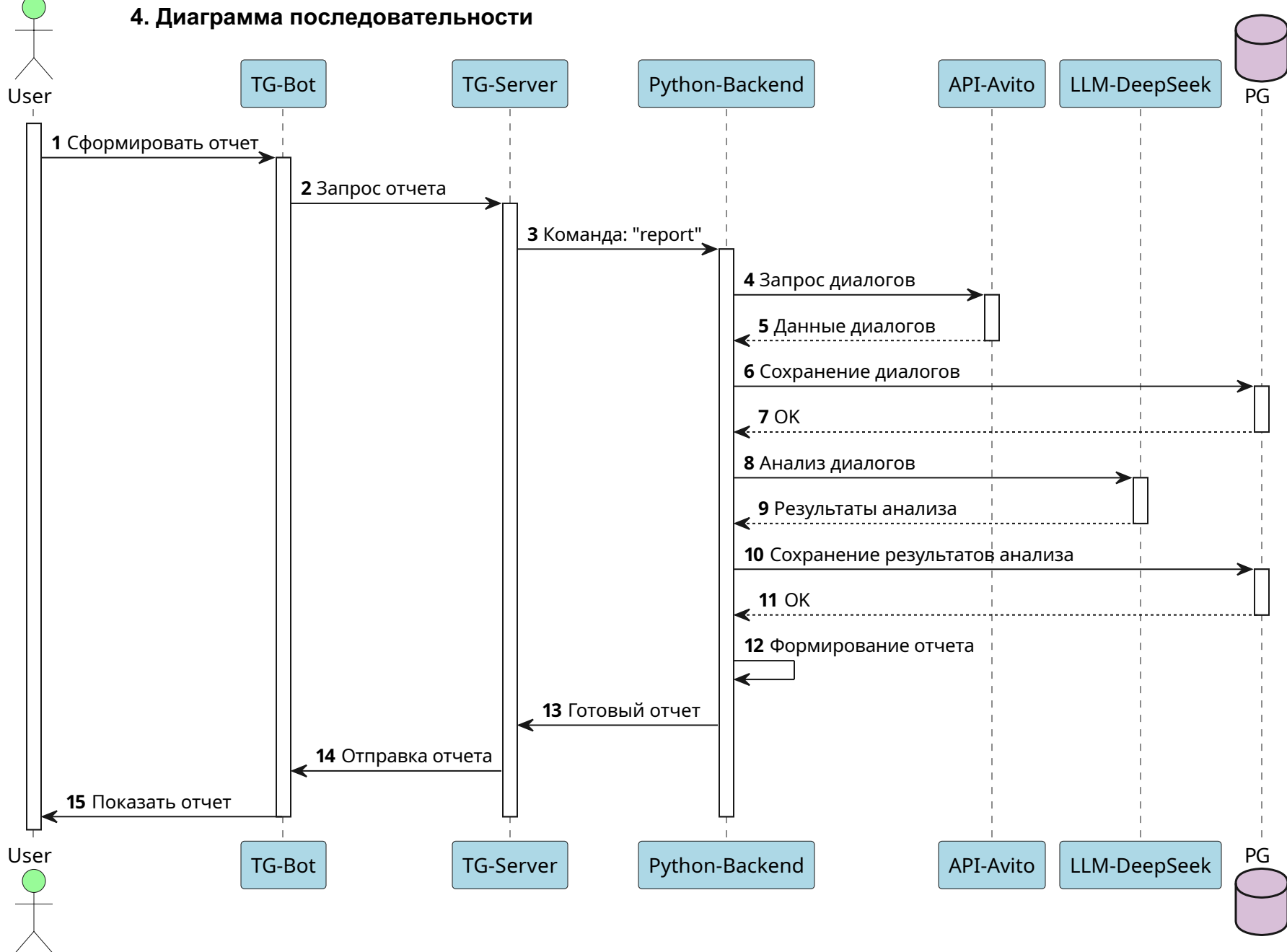
Сущность	Атрибут	Тип данных	Описание
messages	-	-	Содержание диалогов (чатов).
	message_id	VARCHAR (PK)	Уникальный идентификатор сообщения
	chat_id	VARCHAR (FK)	Ссылка на диалог (чат).
	text	TEXT	Текст сообщения
	is_from_company	BOOL	Флаг: от компании или клиента.

Сущность	Атрибут	Тип данных	Описание
chat_reports	-	-	Данные для отчета.
	chat_report_id	INTEGER (PK)	Уникальный идентификатор отчета.
	chat_id	VARCHAR (FK)	Ссылка на диалог (чат).
	report_date	TIMESTAMP	Дата генерации отчета
	total_messages	INTEGER	Общее количество сообщений.
	company_messages	INTEGER	Кол-во сообщений от компании.
	client_messages	INTEGER	Кол-во сообщений от клиента.
	positive_count	INTEGER	Кол-во позитивных сообщений.
	neutral_count	INTEGER	Кол-во нейтральных сообщений.
	negative_count	INTEGER	Кол-во негативных сообщений.
	sentiment_score	FLOAT	Общая оценка тональности (0-10).
	professionalism_score	FLOAT	Оценка профессионализма (0-10).
	clarity_score	FLOAT	Оценка ясности коммуникации (0-10).
	problem_solving_score	FLOAT	Оценка решения проблем (0-10).
	closure_score	FLOAT	Оценка закрытия диалога (0-10).

3. ER Диаграмма



4. Диаграмма последовательности



5. REST API Табличный вид

Endpoint: POST /webhook

Описание: Получение Webhok-ов от Telegram

Request:

Название параметра	Тип данных	Находится в	Описание	Обязательность
Content-Type	string	headers	Указывает JSON формат тела запроса (“application/json”)	Да

Response:

Название параметра	Тип данных	Находится в	Описание	Обязательность
status	string	body	Статус выполнения (“ok”)	Да

Response code: 200 OK

Endpoint: POST /sync/avito

Описание: Запуск синхронизации данных с Авито

Request:

Название параметра	Тип данных	Находится в	Описание	Обязательность
Apikey	string	headers	API Key для аутентификации	Да
Content-Type	string	headers	Указывает JSON формат тела запроса (“application/json”)	Да

Response:

Название параметра	Тип данных	Находится в	Описание	Обязательность
status	string	body	Статус выполнения (“ok”)	Да

Response code: 200 OK

Endpoint: POST /llm/analyze

Описание: Запуск анализа чатов через DeepSeek AI

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
Apikey	string	headers	API Key для аутентификации	Да
Content-Type	string	headers	Указывает JSON формат тела запроса (“application/json”)	Да

Response:

Название параметра	Тип данных	Находится в	Описание	Обязательность
status	string	body	Статус выполнения (“ok”)	Да

Response code: 200 OK

Endpoint: POST /timer/reports

Описание: Отправка отчетов за прошлый день

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
Apikey	string	headers	API Key для аутентификации	Да
Content-Type	string	headers	Указывает JSON формат тела запроса (“application/json”)	Да

Response:

Название параметра	Тип данных	Находится в	Описание	Обязательность
status	string	body	Статус выполнения (“ok”)	Да

Response code: 200 OK

AI-Agent 1.0.0 OAS 3.0

API для анализа чатов Авито с использованием ИИ

Servers

https://ai-agent-zelenkow.amvera.io/docs

▼

User

^

POST	/webhook	Telegram Webhook	▼
POST	/sync/avito	Trigger Avito Sync	▼
POST	/llm/analyze	Trigger Llm Analyze	▼
POST	/timer/reports	Trigger Timer Reports	▼

Schemas

^

HTTPValidationError	>	↑
ValidationError	>	↑

Критерии приемки

User Story:

Как владелец компании, продающей товары на Авито, я хочу получать ежедневные отчеты в Telegram-боте с оценкой общения сотрудников с клиентами, чтобы контролировать качество обслуживания, тем самым повышая лояльность клиентов к компании.

User Case:

Основной сценарий (автоматический режим):

1. Система запускается по расписанию (раз в 24 часа)
2. Система получает данные диалогов сотрудников с клиентами через API Авито
3. Система отправляет тексты диалогов на анализ через API DeepSeek
4. API DeepSeek возвращает оценки качества общения по каждому диалогу
5. Система формирует сводный отчет с оценками по сотрудникам
6. Telegram-бот отправляет отчет владельцу компании
7. Владелец компании получает отчет в установленное время

Case 1:

Функциональность:

Система автоматически формирует и отправляет ежедневный отчет о качестве общения сотрудников

Дано:

Настроено время ежедневной отправки отчетов (09:00), в системе есть данные диалогов за предыдущий день, id владельца telegram добавлен в систему

Когда:

Наступает установленное время отправки отчетов

Тогда:

Владелец получает отчет в установленное время

Case 2:

Функциональность:

Формирование отчета по требованию через кнопку в боте

Дано:

Владелец компании имеет активную сессию в Telegram, бот активирован, id владельца telegram добавлен в систему

Когда:

Владелец нажимает кнопку «Сформировать отчет» в Telegram-боте, указывает начальную и конечную даты

Тогда:

Владелец получает отчет в Telegram

Нефункциональные требования

Производительность системы:

- Время отклика Telegram бота: < 2 секунд для всех команд пользователя
- Время обработки LLM запроса: < 60 секунд на один чат
- Время синхронизации Авито данных: < 5 минут для 100 чатов
- Пропускная способность API: поддержка до 10 одновременных LLM запросов

Доступность системы:

- 99.0% доступности в рабочее время (9:00-18:00 по МСК)

Защита данных:

- Только авторизованные пользователи имеют доступ к отчетам