

## **Бизнес-требования**

1. User Story
2. User Case
3. Макет
4. 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

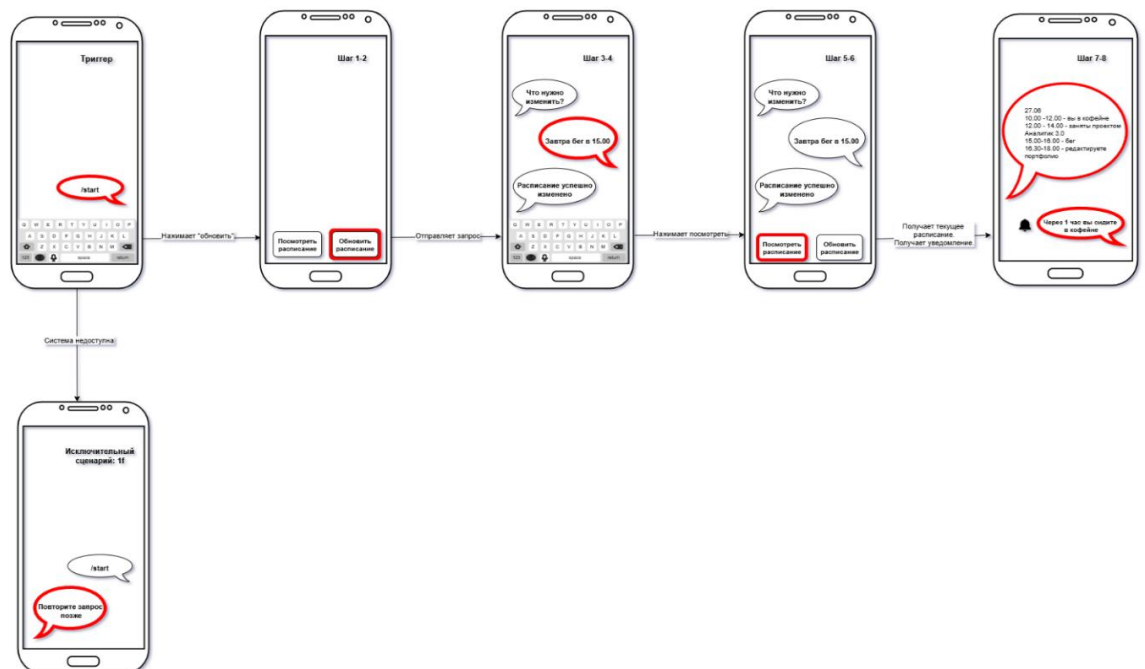
# 1. Название

AI-агент Личный помощник

## 2. User Story

Как пользователь Telegram, я хочу отправлять сообщения на естественном языке, чтобы бот извлекал дату, время и описание, создавал события в моем расписании и отправлял мне напоминания.

## 3. Макеты

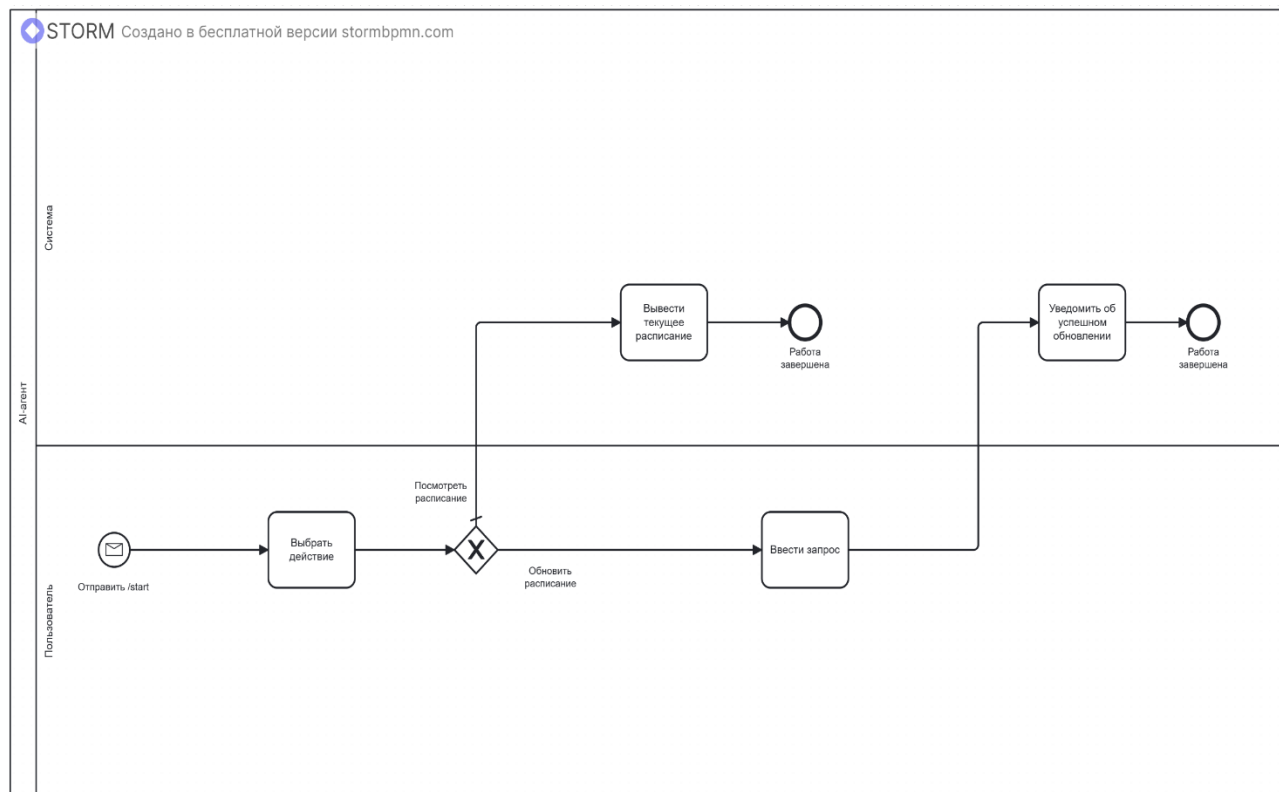


## 4. Use Case

Заголовок	Формирование расписания через AI-агента в telegram
Акторы	Пользователь telegram
Предусловия	<ul style="list-style-type: none"><li>- Пользователь запустил Telegram-бота</li><li>- Система авторизована и подключена к базе данных PostgreSQL</li><li>- AI-модель доступна для обработки запросов</li><li>- В базе данных существуют необходимые таблицы</li></ul>
Ограничения	
Триггер	Пользователь вводит команду /start
Основной сценарий	<ol style="list-style-type: none"><li>1. Система отображает на выбор просмотреть текущее расписание или обновить расписание</li><li>2. Пользователь нажимает обновить расписание</li><li>3. Система отображает уведомление: «Введите событие, которое нужно добавить в расписание»</li><li>4. Пользователь отправляет запрос с обновлением расписания</li><li>5. Система отправляет уведомление о том, что расписание успешно изменено и отображает кнопку посмотреть расписание и обновить.</li><li>6. Пользователь нажимает «посмотреть расписание»</li><li>7. Система выводит текущее расписание</li><li>8. За час до мероприятия бот присылает сообщение с описанием мероприятия.</li><li>9. Каждый день в 10.00 система присылает все расписание на текущий день</li></ol>

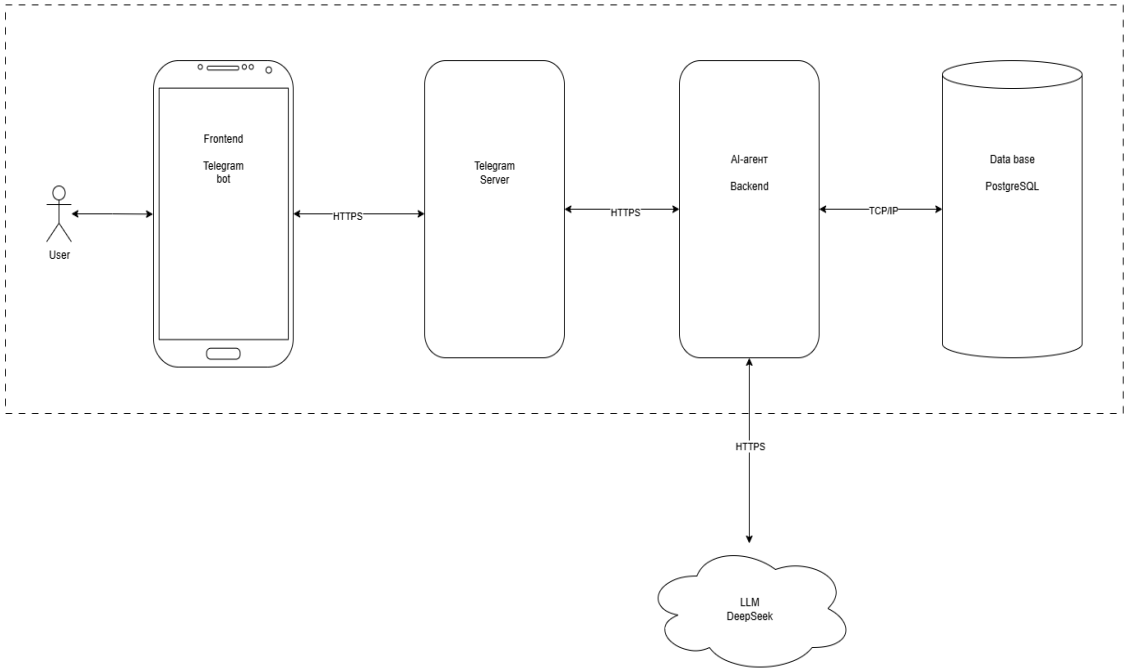
Альтернативный сценарий	<p>1a. Пользователь нажимает посмотреть расписание Переход к основному сценарию шаг 6</p> <p>8f. Пользователь вводит «удалить встреча завтра» 9f. Система находит событие и удаляет его 10f. Система присылает уведомление, об успешном удалении.</p>
Исключительный сценарий	<p>4b. Пользователь вводит запрос с бессмысленным текстом 5b. Система уведомляет, что на эту дату невозможно сформировать расписание и приводит пример корректного запроса</p> <p>8с. Пользователь вводит команду удаления (например, "удали встречу завтра") 9с. Система не находит указанное событие и сообщает, что событие не найдено</p> <p>1d. Система недоступна. Уведомление с просьбой повторить запрос позже.</p>
Результат	<p>Результат: У пользователя формируется актуальное расписание с возможностью добавления, просмотра, удаления событий и получения автоматических уведомлений.</p>

## 5. BPMN



# Пакет функциональных требований

## 1. Архитектура



## 2. Модель данных

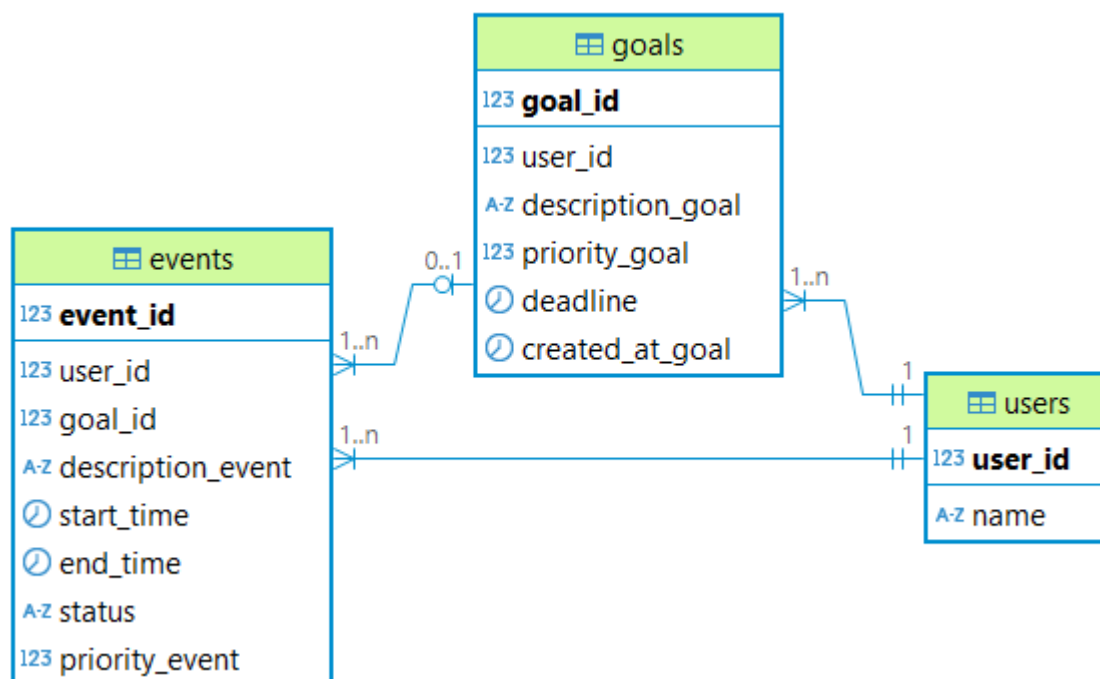
Родительская сущность	Атрибут	Описание
User	—	Аккаунт пользователя в телеграм
	User_id	Уникальный идентификатор пользователя в Telegram
	Name	Имя пользователя

Родительская сущность	Атрибут	Описание
Goal	—	Долгосрочная цель пользователя.(Например: Составь план тренировок для новичка. Цель - к концу месяца пробежать 7 км)
	Goal_id	Уникальный идентификатор цели
	User_id	Аккаунт пользователя в телеграм. Ссылка на пользователя.
	Description_goal	Описание цели
	Deadline	Крайний срок выполнения
	Created_at_goal	Дата создания цели

Родительская сущность	Атрибут	Описание
Event	—	Конкретное мероприятие в расписании
	Event_id	Уникальный идентификатор мероприятия

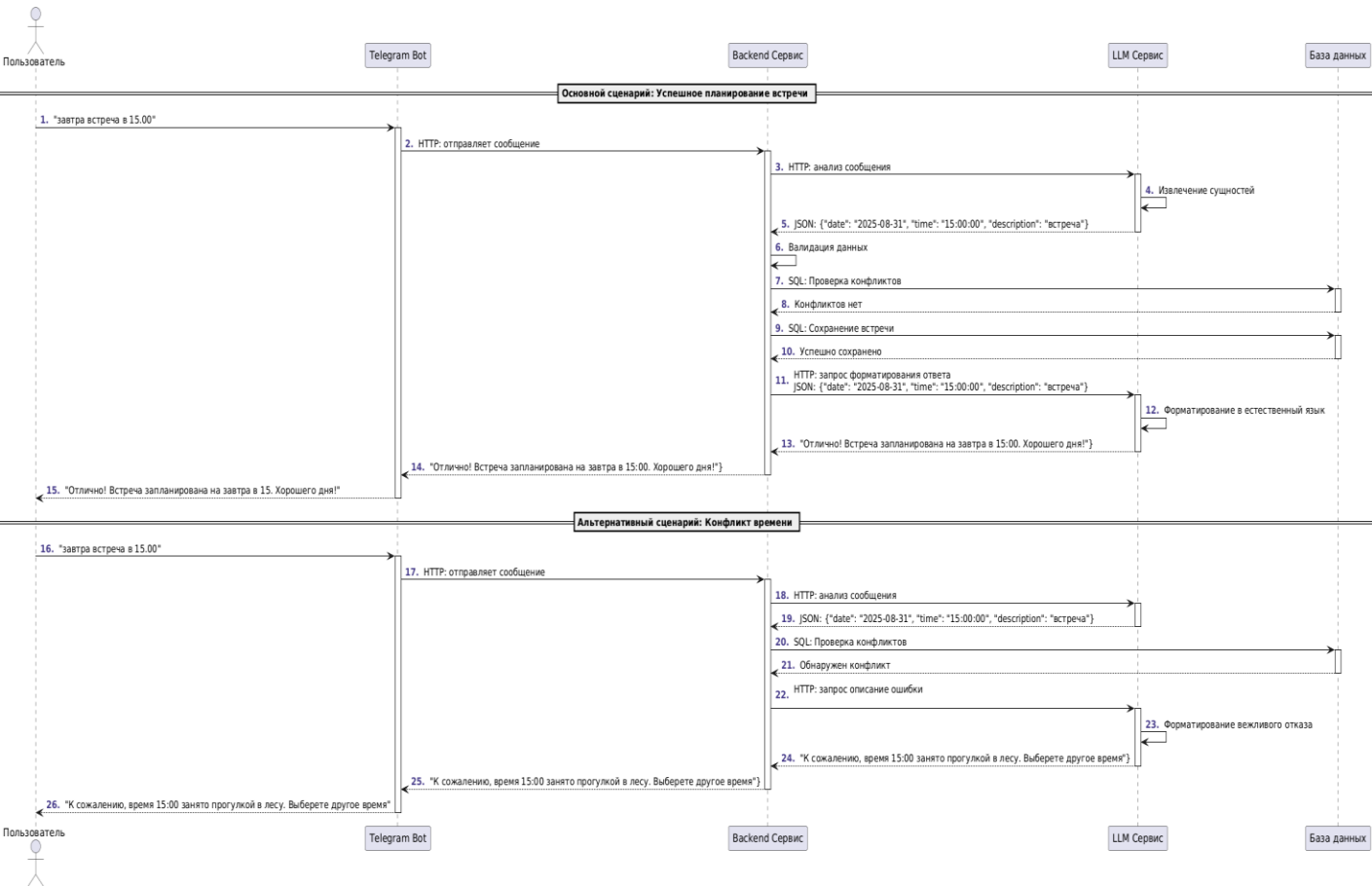
	User_id	Уникальный идентификатор пользователя.
	Goal_id	Ссылка на цель пользователя. Опционально. Если мероприятие связано с целью.
	Description_event	Описание мероприятия
	Start_time	Дата и время начала мероприятия
	End_time	Дата и время завершения мероприятия
	Priority_event	Приоритет события

### 3. ER-диаграмма





# 4. Sequence-диаграммах



## 5. REST API

В проекте нет REST API для основной функциональности. Telegram-бот работает через Telegram Bot API.

Есть только два метода REST для мониторинга состояния сервиса:

### 1. Get /status

Получение статуса работоспособности Telegram-бота

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
-	-	-	-	-

Response Code: 200

Response:

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
status	string	body	Статус работоспособности	да
message	string	body	Сообщение о состоянии	да

### 2. Get /health

Проверка состояния сервиса

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
-	-	-	-	-

**Response Code:** 200

**Response:**

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
status	string	body	Статус состояния	да

## 6. Swagger

GET

/ Проверка статуса работоспособности

📄 ↶ Ⓜ ⤴

Возвращает статус работоспособности Telegram-бота

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Бот работает корректно	No links
<div>Media type</div> <div>application/json</div> <div>Controls Accept header</div> <div>Example Value   Schema</div> <div><pre>{   "status": "healthy",   "message": "Telegram bot is running" }</pre></div>		
500	Внутренняя ошибка сервера	No links
<div>Media type</div> <div>application/json</div> <div>Example Value   Schema</div> <div><pre>{   "status": "error",   "error": "Internal server error" }</pre></div>		

GET /health Проверка статуса

Возвращает детальную информацию о состоянии сервиса

Parameters [Try it out](#)

No parameters

Responses

Code	Description	Links
200	Сервер работает корректно	No links
<p>Media type: <input type="text" value="application/json"/></p> <p>Controls: <a href="#">Accept</a> <a href="#">header</a>.</p> <p>Example Value   Schema</p> <pre>{   "status": "ok" }</pre>		
500	Внутренняя ошибка сервера	No links
<p>Media type: <input type="text" value="application/json"/></p> <p>Example Value   Schema</p> <pre>{   "status": "error",   "error": "Internal server error" }</pre>		

### 3. Критерии приемки и нефункциональные требования

#### 1. Критерии приемки:

АС-1:

- Дано: Пользователь отправляет команду /start
- Когда: Бот получает команду
- Тогда: Отправляется приветственное сообщение с описанием возможностей и клавиатурой

АС-2:

- Дано: Бот отобразил клавиатуру после /start
- Когда: Пользователь нажимает "Обновить расписание"

- Тогда: Бот отправляет "Введите событие, которое нужно добавить в расписание"

#### АС-3:

- Дано: Бот в режиме ожидания нового мероприятия
- Когда: Пользователь пишет "встреча завтра в 15:00"
- Тогда: Бот извлекает дату/время/описание через LLM и сохраняет в базу данных

#### АС-4:

- Дано: Бот отобразил клавиатуру после /start
- Когда: Пользователь нажимает "Посмотреть расписание"
- Тогда: Бот показывает все события из базы данных

#### АС-5:

- Дано: Событие существует в базе с конкретным временем
- Когда: Время подходит к событию (за 1 час)
- Тогда: Бот отправляет уведомление с описанием события

#### АС-7

- Дано: Пользователь отправляет команду /goal
- Когда: Бот получает команду
- Тогда: Бот запрашивает у пользователя глобальную цель и генерирует пошаговый план с помощью LLM
- Тогда: Бот предлагает принять или отклонить сгенерированный план

## АС-8

- Дано: Пользователь пишет сообщение с командой удаления вроде "удали встречу завтра"
- Когда: Бот обрабатывает команду
- Тогда: Бот находит и удаляет соответствующее событие
- Тогда: Бот отправляет подтверждение об успешном удалении

## АС-9

- Дано: Пользователь отправляет команду /clear
- Когда: Бот получает команду
- Тогда: Бот запрашивает подтверждение с кнопками "Да" и "Нет"
- Тогда: При подтверждении бот удаляет все события пользователя

## АС-10

- Дано: Пользователь отправляет сообщение в свободной форме (например "встреча с коллегой 2 октября в 14:00")
- Когда: Бот получает сообщение
- Тогда: Бот использует LLM для извлечения даты, времени и описания
- Тогда: Бот сохраняет событие в базу данных и отправляет подтверждение

## АС-11

- Дано: Пользователь отправляет бессмысленный текст или гиббериш

- Когда: Бот получает сообщение
- Тогда: Бот определяет, что текст не содержит полезной информации о событии
- Тогда: Бот отправляет сообщение с просьбой переформулировать запрос и примером

## 2. Нефункциональные требования

### Требования надежности:

1. Система должна быть доступна 99% времени в течение месячного периода.

### Требования производительности:

1. Время отклика на команды и действия в телеграм-боте (от нажатия кнопки/ввода команды до ответа бота) должно составлять не более 3 секунд в 95% случаев.
2. Время ответа через запрос к LLM не более 30 секунд.