

Calendar Bot

AI-ассистент на базе телеграм бота, который может управлять календарем: добавлять события в календарь, редактировать и удалять события из календаря, а также показывать события пользователю по дате либо по дате и времени.

Основные функции:

- Внесение событий и задач в календарь на основании текстового сообщения в свободной форме.
 - Изменение и удаление событий в календаре.
 - Предоставление информации о событиях и на определенный день и/или определенное время.
-

User Stories

Шаблон:

Как, <роль>, я хочу <функцию>, чтобы <ценность>.

US001. Как пользователь, я хочу вносить события в календарь, отправляя в телеграм бот сообщения в свободной форме, чтобы сократить время на добавление событий.

- *Критерии приемлемости (Acceptance Criteria):*
 - Пользователь отправляет боту сообщение, содержащее описание события, которое нужно добавить в календарь (сообщение написано в свободной форме и содержит необходимую информацию (дата, время, название) и иную информацию, необходимая информация расположена хаотично).
 - Система добавляет событие или задачу в календарь, добавив краткое описание события, и отправляет пользователю сообщение о том, что событие успешно добавлено.

US002. Как пользователь, я хочу получать информацию о событиях и на определенный день и определенное время (либо только на определенный день) через телеграм бота, чтобы продуктивнее управлять своим временем и не пропустить важные события.

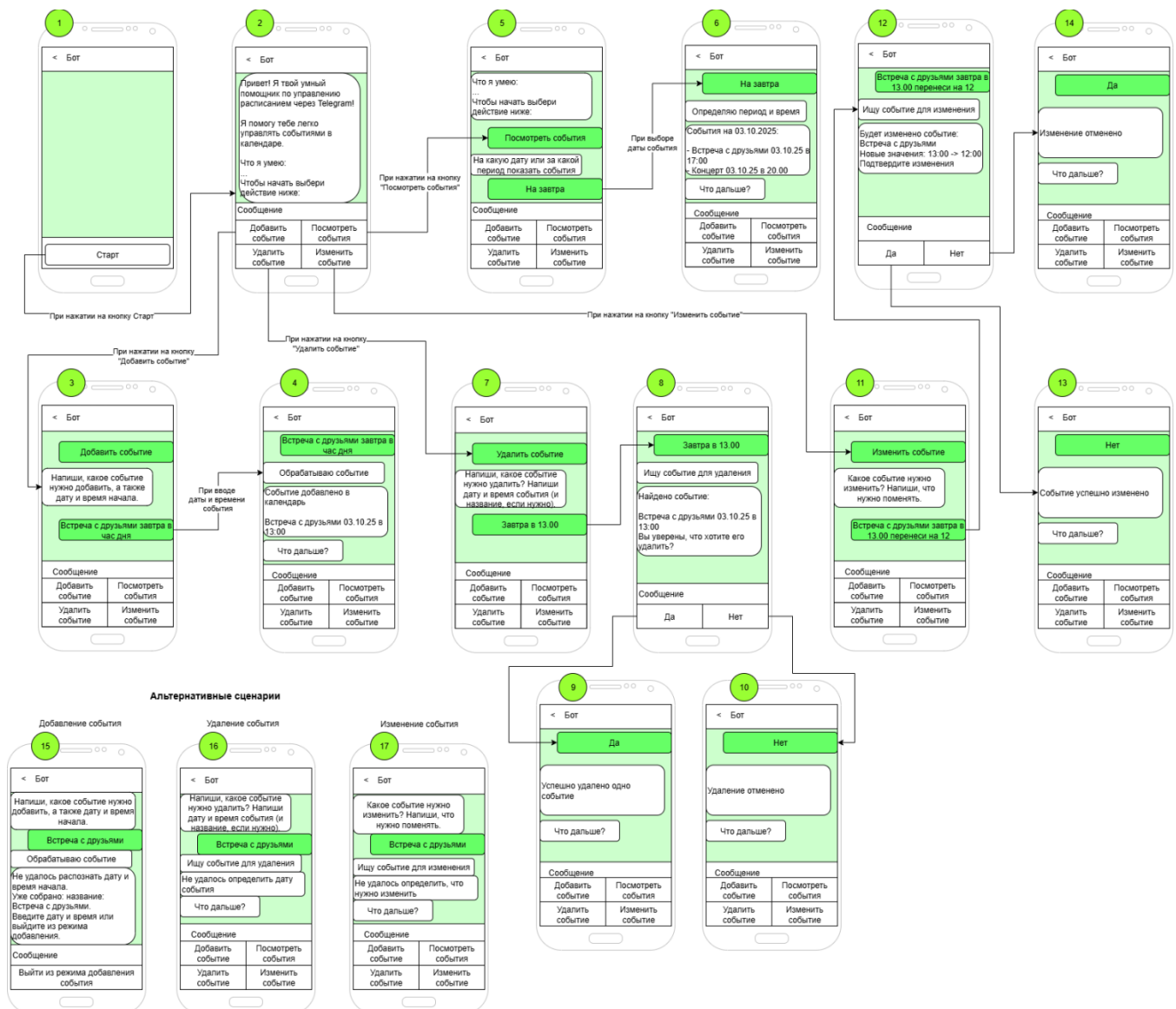
- *Критерии приемлемости (Acceptance Criteria):*
 - Пользователь отправляет боту сообщение с информацией о том, за какой период ему необходимо получить события из календаря (дата и время, либо только дата, либо временной промежуток).
 - Система отправляет пользователю сообщение со списком событий, запланированных на это время, либо сообщает, что события не запланированы на это время.

US003. Как пользователь, я хочу иметь возможность изменять / удалять события в календаре через телеграм бота, чтобы поддерживать свое расписание в актуальном состоянии.

- *Критерии приемлемости (Acceptance Criteria):*

- Пользователь описывает событие, которое нужно изменить / удалить (в случае изменения описывается также то, что нужно изменить).
- Система находит событие по описанию и запрашивает у пользователя подтверждение его изменения / удаления.
- Пользователь подтверждает изменение / удаление события.
- Система отправляет пользователю сообщение об изменении / удалении события.

Макеты



Use Case: Добавление нового события

Идентификатор: UC-001

Название

Добавление нового события в базу через Telegram-бота

Цель

Позволить пользователю сохранить событие в базе данных, отправив сообщение в свободной форме, с автоматическим извлечением данных с помощью ИИ и сохранением в базу данных.

Акторы

- **Пользователь** — пользователь Telegram, создающий событие
- **Telegram-бот** — бот, принимающий сообщения и взаимодействующий с пользователем
- **DeepSeek API** — модель ИИ для анализа естественного языка
- **Supabase** — база данных для хранения пользователей и событий

Предусловия

1. Пользователь запустил бота в Telegram.
2. Бот запущен и работает 24/7.
3. Настроены переменные окружения:
 - TELEGRAM_BOT_TOKEN
 - SUPABASE_URL
 - SUPABASE_SERVICE_ROLE_KEY
 - DEEPSEEK_API_KEY
4. Таблицы `users` и `events` существуют в Supabase.

Основной сценарий (успешный поток)

1. Пользователь отправляет сообщение в свободной форме:
"Завтра в 18:00 встреча с командой в Zoom. Обсудим презентацию."
2. Бот получает сообщение и отвечает:
▢ Обработка события...
3. Бот отправляет текст в **DeepSeek API** с запросом извлечь:
 - Название события
 - Описание
 - Дату и время начала
 - Дату и время окончания (если указано)
 - Место проведения (если указано)
4. DeepSeek возвращает JSON:

```
5. {
6.   "event_title": "Встреча с командой",
7.   "event_description": "Обсудим презентацию",
8.   "start_datetime": "2025-04-06 18:00",
9.   "end_datetime": null,
10.  "event_place": "Zoom"
11. }
```
12. Бот проверяет, что есть:
 - event_title
 - start_datetime
13. Бот проверяет, есть ли пользователь в таблице `users`:
 - Если нет — создаёт новую запись с `telegram_id`

- Если есть — использует существующий user_id
14. Бот сохраняет событие в таблицу events в Supabase.
15. Бот отправляет пользователю подтверждение:
- ☐ Событие добавлено:
 - Встреча с командой
 - ☐ 2025-04-06 18:00
 - ☐ Zoom
 - ☐ Обсудим презентацию

Альтернативные сценарии

4А. Нет названия события:

4А-1. DeepSeek не смог извлечь event_title и вернул event_title = null.

4А-2. Бот отвечает, что не удалось распознать название события, сообщает, какие данные уже собраны и предлагает ввести название события или выйти из режима добавления.

4А-3 Пользователь вводит название события. (возврат к шагу 4 основного сценария)

4В. Нет даты и времени начала:

4В-1. DeepSeek вернул start_datetime = null.

4В-2. Бот отвечает, что не удалось распознать дату начала события, сообщает, какие данные уже собраны и предлагает ввести дату начала события или выйти из режима добавления.

4В-3 Пользователь вводит время начала события. (возврат к шагу 4 основного сценария)

4А-3А/4В-3А. Пользователь выходит из режима добавления:

4А-3А/4В-3А-1. Пользователь нажимает кнопку "Выйти из режима добавления".

4А-3А/4В-3А-2. Бот предлагает пользователю выбрать следующее действие.

7А. Ошибка при сохранении в базу

7А-1. Произошла ошибка при работе с Supabase.

7А-2. Бот отвечает:

☐ Ошибка при сохранении в базу.

7А-3. Ошибка записывается в логи. (Возврат к шагу 1 основного сценария.)

Постусловия

- Событие сохранено в базе данных.
- Пользователь получил подтверждение.
- Логи операции записаны (при необходимости).

Примечания

- Входные данные: Текст сообщения от пользователя.
 - Выходные данные: Подтверждение о создании события.
 - Используемые API: Telegram Bot API, DeepSeek API
 - База данных: Supabase (PostgreSQL)
 - Формат даты: YYYY-MM-DD HH:MM
 - Бот поддерживает неявные даты: "завтра", "в понедельник", "через неделю".
-

Use Case: Просмотр событий из календаря

Идентификатор: UC-002

Название

Просмотр событий из календаря за указанный период

Цель

Позволить пользователю получить список событий из календаря за указанный период времени с возможностью фильтрации по времени.

Акторы

- **Пользователь** — пользователь Telegram, запрашивающий события
- **Telegram-бот** — бот, обрабатывающий запрос и возвращающий события
- **DeepSeek API** — модель ИИ для анализа естественного языка
- **Supabase** — база данных для получения событий

Предусловия

1. Пользователь запустил бота в Telegram.
2. Бот запущен и работает 24/7.
3. Настроены переменные окружения.
4. Таблицы users и events существуют в Supabase.
5. Пользователь имеет сохраненные события в календаре.

Основной сценарий (успешный поток)

1. Пользователь выбирает "📅 Посмотреть события" в меню бота.
2. Бот переходит в состояние ожидания периода и спрашивает:
"На какую дату или за какой период показать события?"
3. Пользователь отправляет сообщение в свободной форме:
"Какие у меня дела на завтра?"
4. Бот отвечает:
🕒 Определяю период и время...
5. Бот отправляет текст в **DeepSeek API** с запросом извлечь:
 - Начальную дату
 - Конечную дату
 - Время начала (если указано)
 - Время окончания (если указано)
 - Точное время (если указано)
6. DeepSeek возвращает JSON:
7.

```
{
```
8.

```
  "start_date": "2025-04-06",
```
9.

```
  "end_date": "2025-04-06",
```
10.

```
  "start_time": null,
```
11.

```
  "end_time": null,
```
12.

```
  "exact_time": null
```
13.

```
}
```
14. Бот выполняет запрос к Supabase для получения событий пользователя за указанный период.
15. Бот находит события и формирует ответ:
📅 События на 06.04.2025:\n• **Встреча с командой** — 06.04.2025 18:00\n📅 Zoom\n📅 Обсудим презентацию\n\n• **Презентация проекта** — 06.04.2025 14:00"

16. Бот возвращается в главное меню.

Альтернативные сценарии

3А. Пользователь указывает точное время:

3А-1. Пользователь пишет: "на завтра в 18:00"

3А-2. DeepSeek возвращает exact_time = "18:00"

3А-3. Бот ищет события только на указанное время

3А-4. Бот отвечает: "На 06.04.2025 в 18:00 нет запланированных событий."

3В. Пользователь указывает диапазон времени:

3В-1. Пользователь пишет: "после 15:00 сегодня"

3В-2. DeepSeek возвращает start_time = "15:00"

3В-3. Бот ищет события после указанного времени

3В-4. Бот показывает соответствующие события

7А. События не найдены:

7А-1. Бот не находит событий за указанный период

7А-2. Бот отвечает: "На 06.04.2025 нет запланированных событий."

7В. Ошибка при получении событий:

7В-1. Произошла ошибка при работе с Supabase

7В-2. Бот отвечает: "❌ Ошибка при получении событий."

7В-3. Ошибка записывается в логи

Постусловия

- Пользователь получил список событий за указанный период.
- Бот вернулся в главное меню.
- Логи операции записаны (при необходимости).

Примечания

- Входные данные: Текст сообщения с периодом/датой
- Выходные данные: Список событий в читаемом формате
- Поддерживаются сложные запросы: "на этой неделе", "в пятницу вечером", "до 12:00 завтра"

Use Case: Удаление события из календаря

Идентификатор: UC-003

Название

Удаление события из календаря с подтверждением

Цель

Позволить пользователю безопасно удалить событие из календаря с предварительным подтверждением.

Акторы

- **Пользователь** — пользователь Telegram, удаляющий событие
- **Telegram-бот** — бот, обрабатывающий запрос на удаление
- **DeepSeek API** — модель ИИ для идентификации события
- **Supabase** — база данных для удаления события

Предусловия

1. Пользователь запустил бота в Telegram.
2. Бот запущен и работает 24/7.
3. Настроены переменные окружения.
4. Таблицы users и events существуют в Supabase.
5. Событие, которое нужно удалить, существует в базе данных.

Основной сценарий (успешный поток)

1. Пользователь выбирает "❏ Удалить событие" в меню бота.
2. Бот переходит в состояние ожидания удаления и спрашивает:
"Какое событие нужно удалить?\nНапиши название или упомяни дату."
3. Пользователь отправляет сообщение в свободной форме:
"Удали встречу с командой завтра в 18:00"
4. Бот отвечает:
❏ Ищу событие для удаления...
5. Бот отправляет текст в **DeepSeek API** с запросом извлечь:
 - Название события
 - Дату события
 - Точное время
6. DeepSeek возвращает JSON:
7.

```
{
```
8.

```
  "event_title": "встреча с командой",
```
9.

```
  "start_date": "2025-04-06",
```
10.

```
  "exact_time": "18:00"
```
11.

```
}
```
12. Бот выполняет запрос к Supabase для поиска события по критериям.
13. Бот находит событие и показывает его пользователю:
"Найдено событие:\n\n• **Встреча с командой** — 06.04.2025 18:00\n❏ Zoom\n❏ Обсудим презентацию\n\nВы уверены, что хотите его удалить?"
14. Бот предлагает клавиатуру подтверждения с кнопками "❏ Да" и "❏ Нет".
15. Пользователь нажимает "❏ Да".
16. Бот удаляет событие из базы данных.
17. Бот подтверждает удаление:
"❏ Успешно удалено 1 событие."
18. Бот возвращается в главное меню.

Альтернативные сценарии

3А. Пользователь указывает только дату:

- 3А-1. Пользователь пишет: "удали события на завтра"
- 3А-2. DeepSeek возвращает только start_date
- 3А-3. Бот ищет все события на указанную дату
- 3А-4. Если найдено несколько событий, бот удаляет их все

8А. Событие не найдено:

- 8А-1. Бот не находит события по указанным критериям
- 8А-2. Бот отвечает: "❏ Событие на 06.04.2025 не найдено."
- 8А-3. Бот возвращается в главное меню

10А. Пользователь отменяет удаление:

10А-1. Пользователь нажимает "❑ Нет"

10А-2. Бот отвечает: "❑ Удаление отменено."

10А-3. Бот возвращается в главное меню

11А. Ошибка при удалении:

11А-1. Произошла ошибка при работе с Supabase

11А-2. Бот отвечает: "❑ Ошибка при удалении события."

11А-3. Ошибка записывается в логи

Постусловия

- Событие удалено из базы данных (при подтверждении).
- Пользователь получил подтверждение удаления.
- Бот вернулся в главное меню.
- Логи операции записаны (при необходимости).

Примечания

- Входные данные: Текст сообщения с описанием события для удаления
 - Выходные данные: Подтверждение удаления
 - Обязательное подтверждение перед удалением для предотвращения случайных удалений
-

Use Case: Изменение события в календаре

Идентификатор: UC-004

Название

Изменение существующего события в календаре

Цель

Позволить пользователю изменить параметры существующего события (название, описание, время, место) с предварительным просмотром изменений.

Акторы

- **Пользователь** — пользователь Telegram, изменяющий событие
- **Telegram-бот** — бот, обрабатывающий запрос на изменение
- **DeepSeek API** — модель ИИ для идентификации события и изменений
- **Supabase** — база данных для обновления события

Предусловия

1. Пользователь запустил бота в Telegram.
2. Бот запущен и работает 24/7.
3. Настроены переменные окружения.
4. Таблицы users и events существуют в Supabase.
5. Событие, которое нужно изменить, существует в базе данных.

Основной сценарий (успешный поток)

1. Пользователь выбирает "☰ Изменить событие" в меню бота.
2. Бот переходит в состояние ожидания изменения и спрашивает:
"Какое событие нужно изменить?\nНапиши, что нужно поменять.\n\nПримеры:\n• Перенеси встречу с командой на завтра в 19:00\n• Измени название презентации на 'Демо продукта'\n• Перенеси созвон в Zoom"
3. Пользователь отправляет сообщение в свободной форме:
"Перенеси встречу с командой на завтра в 19:00"
4. Бот отвечает:
🔍 Ищу событие для изменения...
5. Бот отправляет текст в **DeepSeek API** с двумя запросами:
 - Извлечь изменения (extract_edit_data)
 - Идентифицировать событие (extract_event_to_delete)
6. DeepSeek возвращает данные изменений:
7.

```
{
```
8.

```
  "event_title": null,
```
9.

```
  "event_description": null,
```
10.

```
  "start_datetime": "2025-04-06T19:00:00",
```
11.

```
  "end_datetime": null,
```
12.

```
  "event_place": null
```
13.

```
}
```
14. Бот находит событие "Встреча с командой" по контексту.
15. Бот показывает изменения пользователю:
"Будет изменено:\n\nСтарые данные:\n• **Встреча с командой** — 18:00\n\nНовые значения:\n☐ Время: 18:00 → 19:00\n\nПодтвердите изменение:"
16. Бот предлагает клавиатуру подтверждения с кнопками "☐ Да" и "☐ Нет".
17. Пользователь нажимает "☐ Да".
18. Бот обновляет событие в базе данных.
19. Бот подтверждает изменение:
"☐ Событие успешно изменено!"
20. Бот возвращается в главное меню.

Альтернативные сценарии

3А. Пользователь изменяет название:

- 3А-1. Пользователь пишет: "Измени название 'Встреча с командой' на 'Совещание по проекту'"
- 3А-2. Бот находит событие и показывает изменения названия
- 3А-3. Процесс продолжается как в основном сценарии

3В. Пользователь изменяет несколько параметров:

- 3В-1. Пользователь пишет: "Перенеси презентацию на завтра в 15:00 в конференц-зал"
- 3В-2. Бот извлекает изменения времени и места
- 3В-3. Бот показывает все изменения

7А. Событие не найдено:

- 7А-1. Бот не может идентифицировать какое событие изменить
- 7А-2. Бот отвечает: "☐ Событие не найдено."
- 7А-3. Бот возвращается в главное меню

7В. Нечего изменять:

- 7В-1. DeepSeek не смог извлечь изменения
- 7В-2. Бот отвечает: "☐ Нечего изменять."
- 7В-3. Бот возвращается в главное меню

10A. Пользователь отменяет изменение:

10A-1. Пользователь нажимает "❑ Нет"

10A-2. Бот отвечает: "❑ Изменение отменено."

10A-3. Бот возвращается в главное меню

11A. Ошибка при сохранении:

11A-1. Произошла ошибка при работе с Supabase

11A-2. Бот отвечает: "❑ Ошибка при сохранении изменений."

11A-3. Ошибка записывается в логи

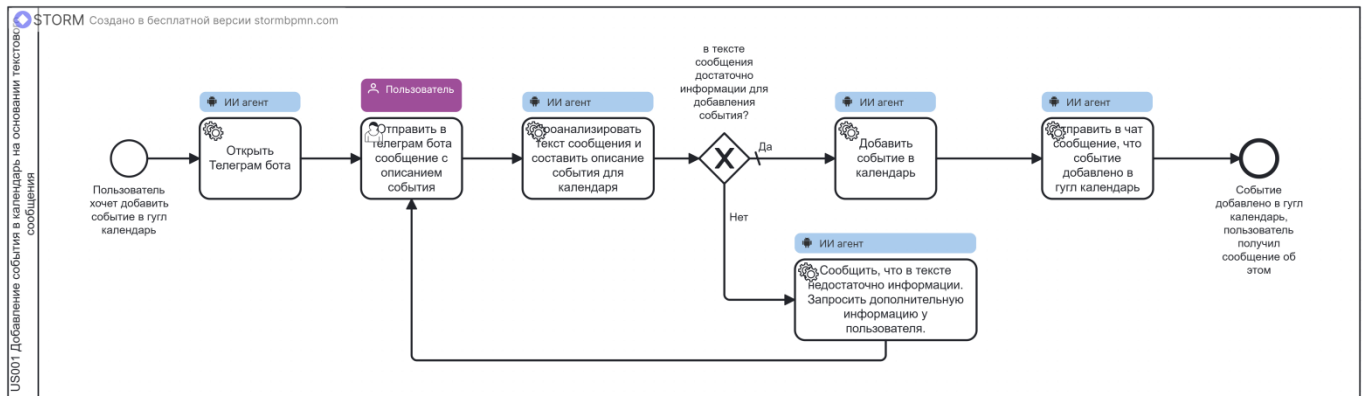
Постусловия

- Событие обновлено в базе данных (при подтверждении).
- Пользователь получил подтверждение изменения.
- Бот вернулся в главное меню.
- Логи операции записаны (при необходимости).

Примечания

- Входные данные: Текст сообщения с описанием изменений
- Выходные данные: Подтверждение изменения
- Поддерживается изменение любого параметра события: название, описание, время, место
- Обязательный предварительный просмотр изменений перед сохранением

BPMN



Архитектура

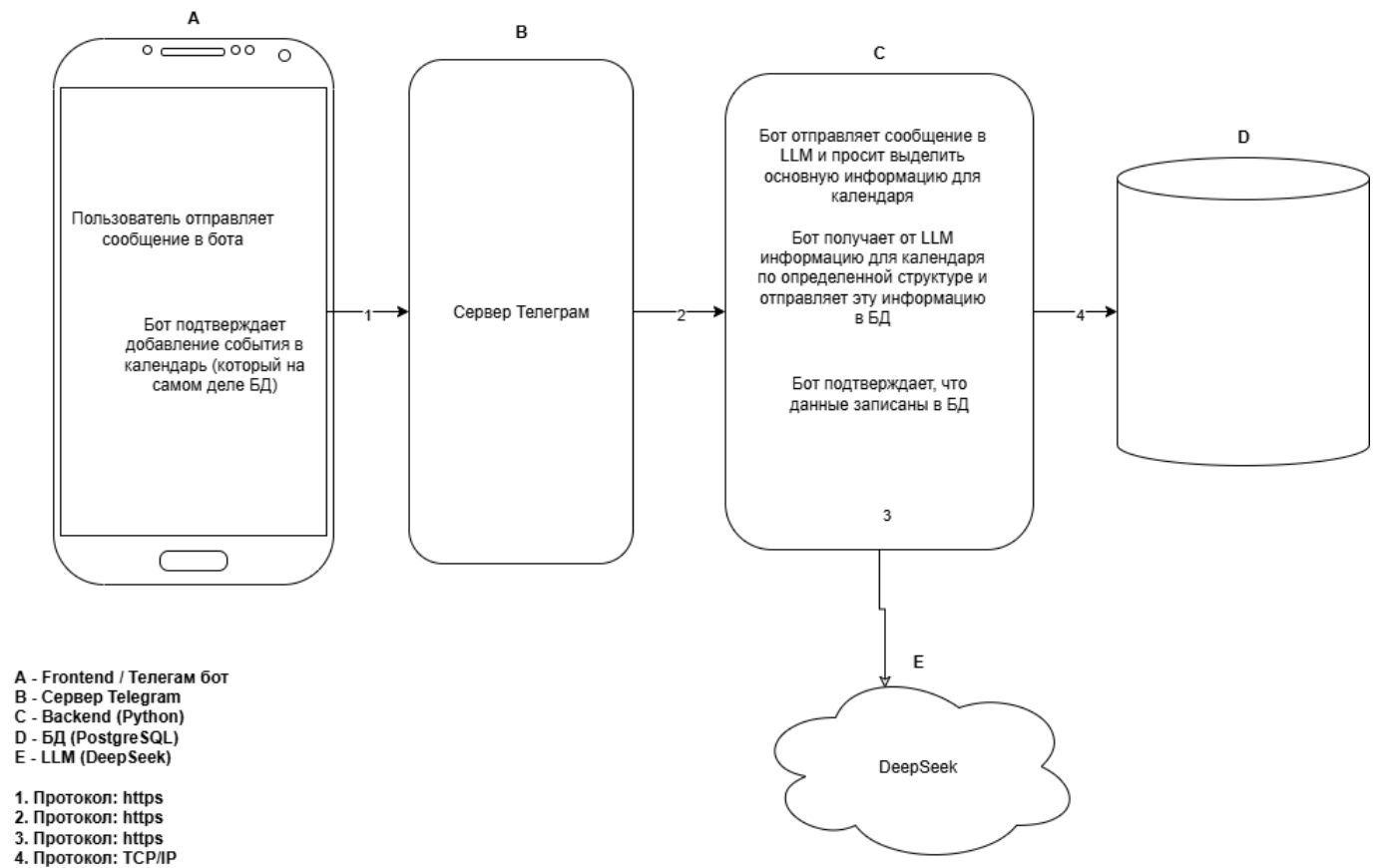
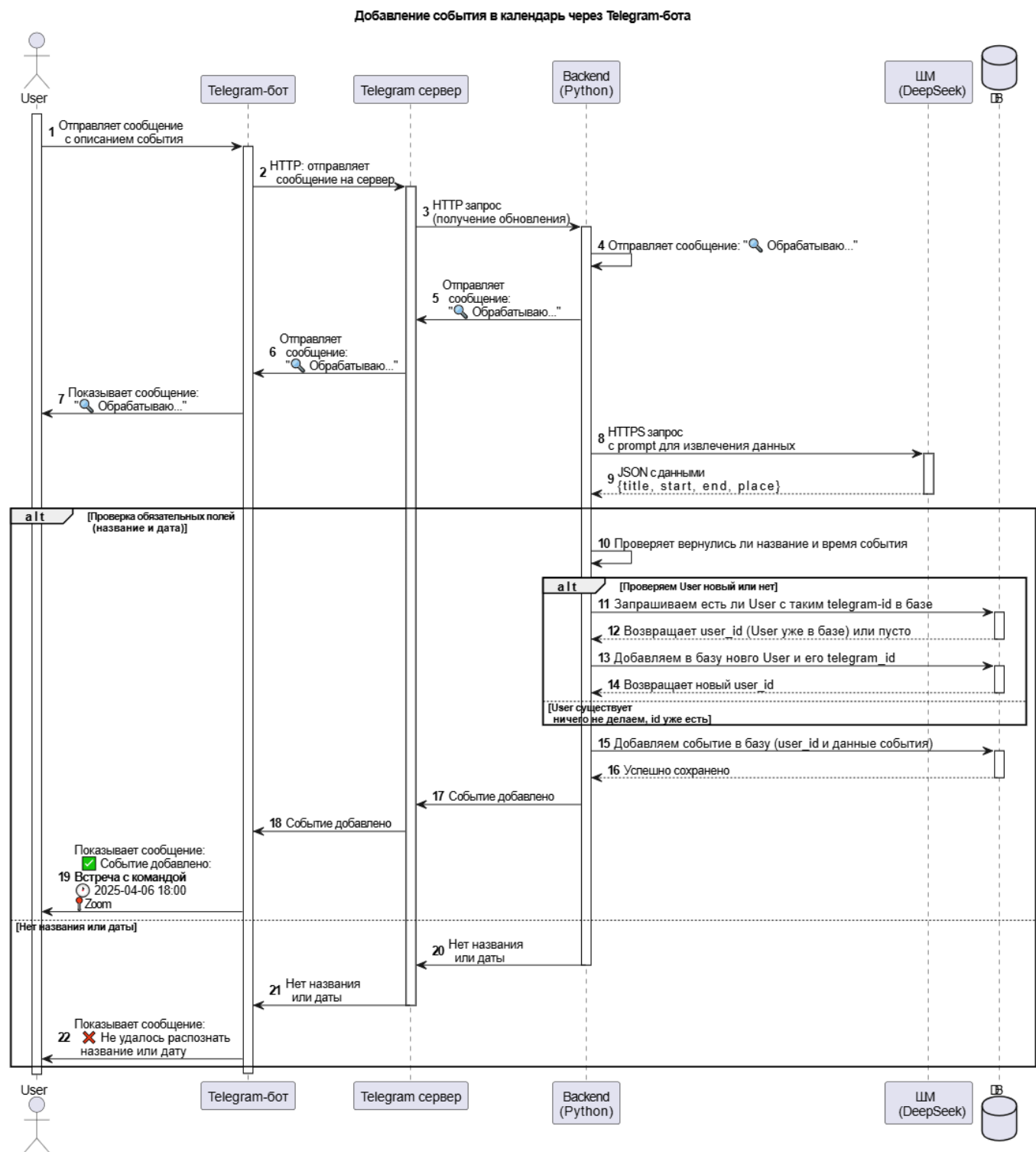
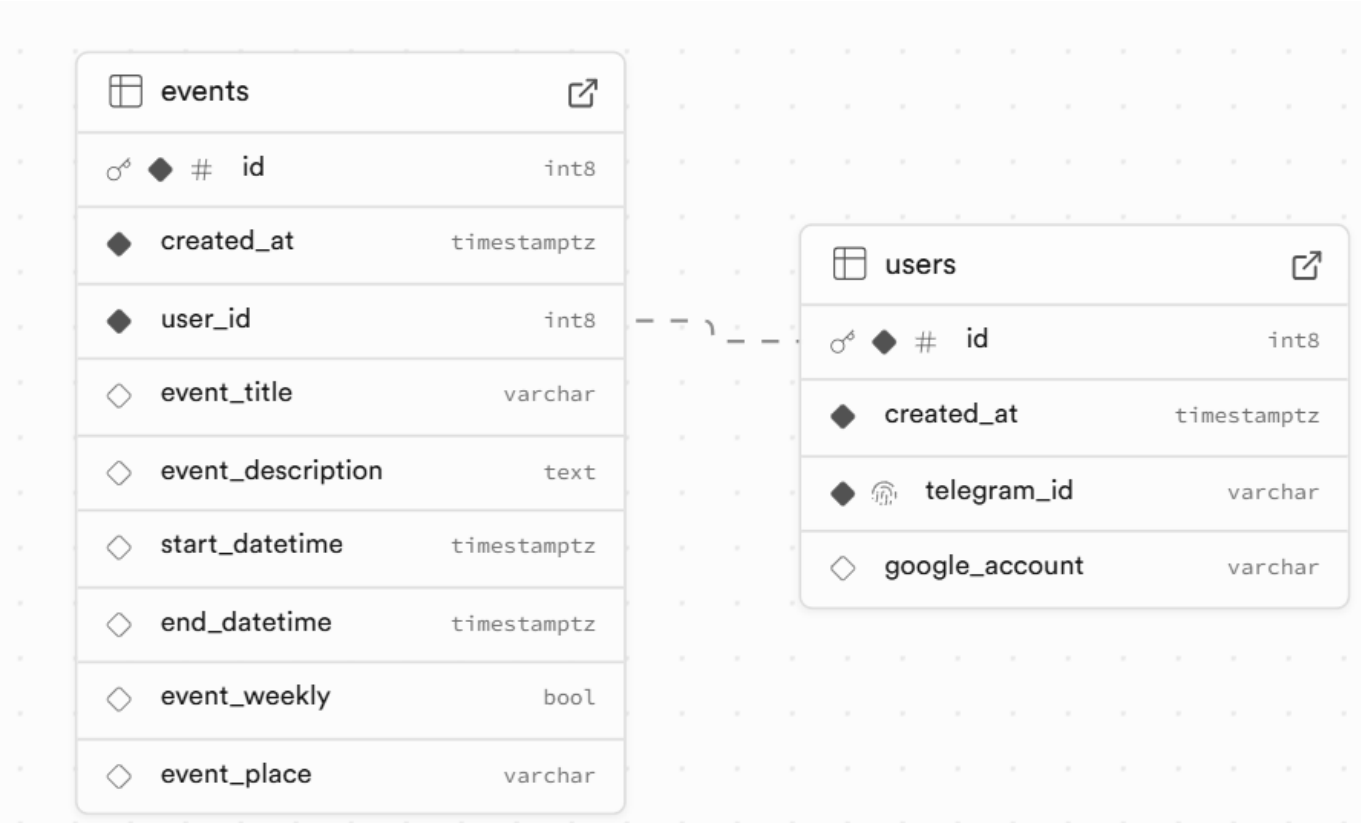


Диаграмма последовательности



ERD-диаграмма



REST. Табличный вид

1.

POST/events

Создание события

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
event_title	string	body	Название события	Да
event_description	string	body	Описание	Нет
start_datetime	string	body	Дата и время начала в формате YYYY-MM-DD HH:MM	Да
end_datetime	string	body	Дата и время окончания	Нет
event_place	string	body	Место проведения	Нет
telegram_id	string	header	ID пользователя в Telegram	Да

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность
id	string	body	Уникальный ID события	Да
event_title	string	body	Название события	Да
event_description	string	body	Описание	Нет
start_datetime	string	body	Дата и время начала в формате YYYY-MM-DD HH:MM	Да
end_datetime	string	body	Дата и время окончания	Нет
event_place	string	body	Место проведения	Нет
Created_at	string	body	Время создания	Да

Response code <201> - Событие создано

Пример запроса

```
POST /events HTTP/1.1
Host: api.eventbot.example
Content-Type: application/json
telegram_id: 7894615329

{
  "event_title": "Встреча с командой",
  "event_description": "Обсудим релиз",
  "start_datetime": "2025-09-14T18:00:00",
  "end_datetime": "2025-09-14T19:30:00",
  "event_place": "Zoom"
}
```

Пример успешного ответа

```
HTTP/1.1 201 Created

{
  "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "event_title": "Встреча с командой",
  "event_description": "Обсудим релиз",
  "start_datetime": "2025-09-14T18:00:00",
  "end_datetime": "2025-09-14T19:30:00",
  "event_place": "Zoom",
  "created_at": "2025-09-13T15:30:00Z"
}
```

2.

GET/events

Получение событий по периоду времени

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
start_date	string	query	Начальная дата в формате YYYY-MM-DD	Нет
end_date	string	query	Конечная дата в формате YYYY-MM-DD	Нет
start_time	string	query	Время начала в формате HH:MM	Нет
end_time	string	query	Время окончания в формате HH:MM	Нет
telegram_id	string	header	ID пользователя в Telegram	Да

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность
id	string	body	Уникальный ID события	Да
event_title	string	body	Название события	Да
event_description	string	body	Описание	Нет
start_datetime	string	body	Дата и время начала в формате	Да
end_datetime	string	body	Дата и время окончания	Нет
event_place	string	body	Место проведения	Нет
Created_at	string	body	Время создания	Да

Response code <200> - Ок

Пример запроса

GET /events?start_date=2025-09-14&end_date=2025-09-14 HTTP/1.1 Host: api.eventbot.example telegram_id: 7894615329

Пример успешного ответа

HTTP/1.1 200 OK [{ "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890", "event_title": "Встреча с командой", "event_description": "Обсудим релиз", "start_datetime": "2025-09-14T18:00:00", "end_datetime": "2025-09-14T19:30:00",

```
"event_place": "Zoom",
"created_at": "2025-09-13T15:30:00Z"
}
]
```

3.

PUT/events/{id}

Изменение события

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
event_title	string	body	Новое название события	Да
event_description	string	body	Новое описание	Нет
start_datetime	string	body	Новая дата и время начала	Да
end_datetime	string	body	Новая дата и время окончания	Нет
event_place	string	body	Новое место проведения	Нет
telegram_id	string	header	ID пользователя в Telegram	Да

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность
id	string	body	Уникальный ID события	Да
event_title	string	body	Новое название события	Да
event_description	string	body	Новое описание	Нет
start_datetime	string	body	Новая дата и время начала в формате	Да
end_datetime	string	body	Новая дата и время окончания	Нет
event_place	string	body	Новое место проведения	Нет
updated_at	string	body	Время обновления	Да

Response code <200> - Ок

Пример запроса

```
PUT /events/a1b2c3d4-e5f6-7890-abcd-ef1234567890 HTTP/1.1
Host: api.eventbot.example
Content-Type: application/json
telegram_id: 7894615329

{
```



```
"start_datetime": "2025-09-14T19:00:00",
"event_place": "Google Meet"
}
```

Пример успешного ответа

```
HTTP/1.1 200 OK

{
  "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "event_title": "Встреча с командой",
  "event_description": "Обсудим релиз",
  "start_datetime": "2025-09-14T19:00:00",
  "end_datetime": "2025-09-14T19:30:00",
  "event_place": "Google Meet",
  "created_at": "2025-09-13T15:30:00Z",
  "updated_at": "2025-09-14T10:00:00Z"
}
```

4.

DELETE/events/{id}

Удаление события

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
telegram_id	string	header	ID пользователя в Telegram	Да

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность
id	string	body	Уникальный ID события	Да
deleted_at	string	body	Время обновления	Да

Response code <200> - Ок

Пример запроса

```
DELETE /events/a1b2c3d4-e5f6-7890-abcd-ef1234567890 HTTP/1.1
Host: api.eventbot.example
telegram_id: 7894615329
```

Пример успешного ответа

```
HTTP/1.1 200 OK

{
  "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "deleted_at": "2025-09-14T10:15:00Z"
}
```

Коды ошибок (HTTP Status Codes)

Код	Сообщение	Описание
400 Bad Request	Invalid input	Неверный формат даты, времени или отсутствуют обязательные поля
401 Unauthorized	Missing or invalid telegram_id	Не передан telegram_id или он не существует
404 Not Found	Event not found	Событие с таким ID не найдено
500 Internal Server Error	Server error	Ошибка на стороне Supabase или бэкенда

Пример ошибки

```
HTTP/1.1 400 Bad Request

{
  "error": "Invalid datetime format",
  "details": "Field 'start_datetime' must be in format YYYY-MM-DDTHH:MM:SS"
}
```

CalendarBot Events API 1.0.0 OAS 3.0

<http://openapi-preview.example/>

REST API для управления событиями через Telegram бота. Все операции выполняются от имени пользователя (telegram_id).

Основные возможности:

- Создание событий с автоматическим парсингом из текста
- Просмотр событий по заданному периоду
- Изменение существующих событий
- Удаление событий

Аутентификация:

- Через заголовок `telegram_id` в формате строки

Contact Development Team
MIT

Servers

https://api.calendar-bot.example/v1 - Production server

events

Операции с событиями календаря

^

POST	/events	Создание нового события		
GET	/events	Получение событий по периоду		
PUT	/events/{eventId}	Изменение события		
DELETE	/events/{eventId}	Удаление события		

Schemas

^

EventCreateRequest >

EventUpdateRequest >

EventResponse >

DeleteResponse >

Error >

```
openapi: 3.0.3
info:
  title: CalendarBot Events API
  description: |
    REST API для управления событиями через Telegram бота.
    Все операции выполняются от имени пользователя (telegram_id).
```

```
## Основные возможности:
```

- Создание событий с автоматическим парсингом из текста
- Просмотр событий по заданному периоду
- Изменение существующих событий
- Удаление событий

```
## Аутентификация:
```

- Через заголовок `telegram_id` в формате строки

```
version: '1.0.0'
```

```
contact:
```

```
  name: Development Team
  email: dev@calendar-bot.example
```

```
license:
```

```
  name: MIT
  url: https://opensource.org/licenses/MIT
```

```
servers:
```

- url: https://api.calendar-bot.example/v1
 description: Production server
- url: https://staging.api.calendar-bot.example/v1
 description: Staging server

```
tags:
```

- name: events
 description: Операции с событиями календаря

```
paths:
```

```
  /events:
```

```
    post:
```

```
      tags: [events]
```

```
      summary: Создание нового события
```

```
      description: |
```

```
        Создает новое событие в календаре пользователя.
```

```
        Поддерживает автоматический парсинг дат и времени из естественного языка.
```

```
operationId: createEvent
parameters:
  - $ref: '#/components/parameters/TelegramIdHeader'
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/EventCreateRequest'
      examples:
        meetingExample:
          summary: Пример создания встречи
          value:
            event_title: "Встреча с командой"
            event_description: "Обсуждение релиза продукта"
            start_datetime: "2025-09-15T14:00:00"
            end_datetime: "2025-09-15T15:30:00"
            event_place: "Zoom конференция"
        simpleEvent:
          summary: Простое событие
          value:
            event_title: "Обед"
            start_datetime: "2025-09-15T13:00:00"
responses:
  '201':
    description: Событие успешно создано
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/EventResponse'
        examples:
          createdEvent:
            summary: Созданное событие
            value:
              id: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
              event_title: "Встреча с командой"
              event_description: "Обсуждение релиза продукта"
              start_datetime: "2025-09-15T14:00:00"
              end_datetime: "2025-09-15T15:30:00"
              event_place: "Zoom конференция"
              created_at: "2025-09-15T10:30:00Z"
              updated_at: "2025-09-15T10:30:00Z"
```

```
'400':
  $ref: '#/components/responses/BadRequest'
'401':
  $ref: '#/components/responses/Unauthorized'
'500':
  $ref: '#/components/responses/ServerError'

get:
  tags: [events]
  summary: Получение событий по периоду
  description: |
    Возвращает список событий пользователя за указанный период.
    Поддерживает фильтрацию по дате и времени.
  operationId: getEvents
  parameters:
    - $ref: '#/components/parameters/TelegramIdHeader'
    - name: start_date
      in: query
      description: Начальная дата периода (YYYY-MM-DD)
      schema:
        type: string
        format: date
        example: "2025-09-15"
    - name: end_date
      in: query
      description: Конечная дата периода (YYYY-MM-DD)
      schema:
        type: string
        format: date
        example: "2025-09-20"
    - name: start_time
      in: query
      description: Время начала (HH:MM)
      schema:
        type: string
        pattern: '^([0-1][0-9]|2[0-3]):[0-5][0-9]$$'
        example: "09:00"
    - name: end_time
      in: query
      description: Время окончания (HH:MM)
      schema:
        type: string
```

```
    pattern: '^([0-1][0-9]|2[0-3]):[0-5][0-9]$\nexample: "18:00"\n- name: exact_time\n  in: query\n  description: Точное время события (HH:MM)\n  schema:\n    type: string\n    pattern: '^([0-1][0-9]|2[0-3]):[0-5][0-9]$\nexample: "14:30"\nresponses:\n  '200':\n    description: Список событий получен\n    content:\n      application/json:\n        schema:\n          type: array\n          items:\n            $ref: '#/components/schemas/EventResponse'\n        examples:\n          eventsList:\n            summary: Список событий\n            value:\n              - id: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"\n                event_title: "Встреча с командой"\n                start_datetime: "2025-09-15T14:00:00"\n                end_datetime: "2025-09-15T15:30:00"\n                event_place: "Zoom"\n                created_at: "2025-09-14T10:30:00Z"\n                updated_at: "2025-09-14T10:30:00Z"\n              - id: "b2c3d4e5-f6g7-8901-bcde-f23456789012"\n                event_title: "Обед"\n                start_datetime: "2025-09-15T13:00:00"\n                created_at: "2025-09-14T11:45:00Z"\n                updated_at: "2025-09-14T11:45:00Z"\n\n  '400':\n    $ref: '#/components/responses/BadRequest'\n  '401':\n    $ref: '#/components/responses/Unauthorized'\n  '500':\n    $ref: '#/components/responses/ServerError'
```

/events/{eventId}:

```
put:
  tags: [events]
  summary: Изменение события
  description: Обновляет данные существующего события
  operationId: updateEvent
  parameters:
    - $ref: '#/components/parameters/TelegramIdHeader'
    - name: eventId
      in: path
      required: true
      description: UUID события
      schema:
        type: string
        format: uuid
        example: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/EventUpdateRequest'
        examples:
          updateExample:
            summary: Обновление времени и места
            value:
              start_datetime: "2025-09-15T15:00:00"
              event_place: "Google Meet"
  responses:
    '200':
      description: Событие успешно обновлено
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/EventResponse'
    '400':
      $ref: '#/components/responses/BadRequest'
    '401':
      $ref: '#/components/responses/Unauthorized'
    '404':
      $ref: '#/components/responses/NotFound'
    '500':
      $ref: '#/components/responses/ServerError'
```



```
delete:
  tags: [events]
  summary: Удаление события
  description: Удаляет событие по идентификатору
  operationId: deleteEvent
  parameters:
    - $ref: '#/components/parameters/TelegramIdHeader'
    - name: eventId
      in: path
      required: true
      description: UUID события
      schema:
        type: string
        format: uuid
        example: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
  responses:
    '200':
      description: Событие успешно удалено
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/DeleteResponse'
          examples:
            deleteSuccess:
              summary: Успешное удаление
              value:
                id: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
                deleted_at: "2025-09-15T12:30:00Z"
    '401':
      $ref: '#/components/responses/Unauthorized'
    '404':
      $ref: '#/components/responses/NotFound'
    '500':
      $ref: '#/components/responses/ServerError'
```

```
components:
  schemas:
    EventCreateRequest:
      type: object
      required:
        - event_title
```

```
- start_datetime
properties:
  event_title:
    type: string
    description: Название события
    maxLength: 255
    example: "Встреча с командой"
  event_description:
    type: string
    description: Описание события
    maxLength: 1000
    example: "Обсуждение релиза продукта"
  start_datetime:
    type: string
    format: date-time
    description: Дата и время начала события (ISO 8601)
    example: "2025-09-15T14:00:00"
  end_datetime:
    type: string
    format: date-time
    description: Дата и время окончания события
    example: "2025-09-15T15:30:00"
  event_place:
    type: string
    description: Место проведения события
    maxLength: 255
    example: "Zoom конференция"

EventUpdateRequest:
  type: object
  properties:
    event_title:
      type: string
      description: Новое название события
      maxLength: 255
      example: "Встреча с командой"
    event_description:
      type: string
      description: Новое описание события
      maxLength: 1000
      example: "Обсуждение релиза продукта"
    start_datetime:
```

```
    type: string
    format: date-time
    description: Новая дата и время начала
    example: "2025-09-15T15:00:00"
  end_datetime:
    type: string
    format: date-time
    description: Новая дата и время окончания
    example: "2025-09-15T16:30:00"
  event_place:
    type: string
    description: Новое место проведения
    maxLength: 255
    example: "Google Meet"

EventResponse:
  type: object
  properties:
    id:
      type: string
      format: uuid
      description: Уникальный идентификатор события
      example: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
    event_title:
      type: string
      description: Название события
      example: "Встреча с командой"
    event_description:
      type: string
      description: Описание события
      example: "Обсуждение релиза продукта"
    start_datetime:
      type: string
      format: date-time
      description: Дата и время начала
      example: "2025-09-15T14:00:00"
    end_datetime:
      type: string
      format: date-time
      description: Дата и время окончания
      example: "2025-09-15T15:30:00"
    event_place:
```

```
    type: string
    description: Место проведения
    example: "Zoom конференция"
  created_at:
    type: string
    format: date-time
    description: Время создания записи
    example: "2025-09-14T10:30:00Z"
  updated_at:
    type: string
    format: date-time
    description: Время последнего обновления
    example: "2025-09-14T10:30:00Z"
```

DeleteResponse:

```
  type: object
  properties:
    id:
      type: string
      format: uuid
      description: ID удаленного события
      example: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
    deleted_at:
      type: string
      format: date-time
      description: Время удаления
      example: "2025-09-15T12:30:00Z"
```

Error:

```
  type: object
  properties:
    error:
      type: string
      description: Тип ошибки
      example: "VALIDATION_ERROR"
    message:
      type: string
      description: Сообщение об ошибке
      example: "Неверный формат даты"
    details:
      type: object
      description: Детали ошибки
```

```
        additionalProperties: true

parameters:
  TelegramIdHeader:
    name: telegram_id
    in: header
    required: true
    description: Идентификатор пользователя в Telegram
    schema:
      type: string
      example: "1234567890"

responses:
  BadRequest:
    description: Неверный запрос
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error'
        examples:
          validationError:
            summary: Ошибка валидации
            value:
              error: "VALIDATION_ERROR"
              message: "Неверный формат даты"
              details:
                field: "start_datetime"
                expected_format: "YYYY-MM-DDTHH:MM:SS"

  Unauthorized:
    description: Не авторизован
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error'
        examples:
          unauthorized:
            summary: Пользователь не найден
            value:
              error: "UNAUTHORIZED"
              message: "Пользователь с указанным telegram_id не найден"
```

```
NotFound:
  description: Ресурс не найден
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Error'
      examples:
        eventNotFound:
          summary: Событие не найдено
          value:
            error: "NOT_FOUND"
            message: "Событие с указанным ID не найдено"
```

```
ServerError:
  description: Внутренняя ошибка сервера
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Error'
      examples:
        databaseError:
          summary: Ошибка базы данных
          value:
            error: "SERVER_ERROR"
            message: "Ошибка при обращении к базе данных"
```

```
security:
  - ApiKeyAuth: []
```

Критерии приемки

Общие критерии приемки системы

AC-SYS-001: Старт бота

- **Дано:** Пользователь запускает бота командой `/start`
- **Когда:** Бот получает команду
- **Тогда:** Бот отправляет приветственное сообщение с описанием функционала и главным меню с кнопками:
 - ☐ Добавить событие
 - ☐ Посмотреть события
 - ☐ Удалить событие
 - ☒ Изменить событие

AC-SYS-002: Наличие переменных окружения

- **Дано:** Бот запускается
- **Когда:** Проверяются переменные окружения
- **Тогда:** Все обязательные переменные присутствуют:
 - TELEGRAM_BOT_TOKEN
 - SUPABASE_URL
 - SUPABASE_SERVICE_ROLE_KEY
 - DEEPSEEK_API_KEY

AC-SYS-003: Доступность внешних сервисов

- **Дано:** Бот работает
 - **Когда:** Выполняется операция с внешними сервисами
 - **Тогда:** Все сервисы доступны:
 - Telegram Bot API
 - DeepSeek API
 - Supabase (база данных)
-

Критерии приемки для добавления событий (US001/UC-001)

AC-ADD-001: Успешное добавление события

- **Дано:** Пользователь нажимает "□ Добавить событие"
- **Когда:** Пользователь отправляет сообщение в свободной форме с полной информацией о событии
- **Тогда:**
 - Бот обрабатывает сообщение через DeepSeek API
 - Извлекаются: название, описание(необязательно), дата/время начала, дата/время окончания (необязательно), место (необязательно)
 - Событие сохраняется в базу данных
 - Пользователь получает подтверждение с деталями события

AC-ADD-002: Добавление с неявными датами

- **Дано:** Пользователь в режиме добавления события
- **Когда:** Пользователь использует относительные даты ("завтра", "в понедельник", "через неделю")
- **Тогда:**
 - DeepSeek API корректно преобразует относительные даты в абсолютные
 - Событие сохраняется с правильной датой
 - Пользователь видит корректную дату в подтверждении

AC-ADD-003: Частичное извлечение данных

- **Дано:** Пользователь отправляет сообщение с неполной информацией
- **Когда:** DeepSeek не может извлечь название события или дату начала
- **Тогда:**
 - Бот сообщает о недостающих данных
 - Показывает уже собранную информацию
 - Предлагает ввести недостающие данные или выйти
 - Сохраняет частичные данные для следующего сообщения

AC-ADD-004: Обработка ошибок при сохранении

- **Дано:** Пользователь отправляет корректное событие
- **Когда:** Происходит ошибка при сохранении в базу данных
- **Тогда:**
 - Бот сообщает об ошибке сохранения
 - Предлагает повторить попытку или выйти
 - Сохраняет данные события для повторной попытки
 - Ошибка логируется

AC-ADD-005: Создание нового пользователя

- **Дано:** Пользователь впервые использует бота
 - **Когда:** Пользователь добавляет первое событие
 - **Тогда:**
 - Бот создает запись пользователя в таблице `users`
 - Присваивает `telegram_id` пользователя
 - Событие связывается с созданным пользователем
-

Критерии приемки для просмотра событий (US002/UC-002)

AC-VIEW-001: Просмотр событий за день

- **Дано:** Пользователь нажимает "☐ Посмотреть события"
- **Когда:** Пользователь запрашивает события на конкретную дату
- **Тогда:**
 - Бот определяет дату через DeepSeek API
 - Получает события пользователя за указанную дату
 - Форматирует список событий в читаемом виде
 - Показывает название, время, место и описание каждого события

AC-VIEW-002: Просмотр событий по времени

- **Дано:** Пользователь запрашивает события
- **Когда:** Пользователь указывает точное время или диапазон
- **Тогда:**
 - Бот фильтрует события по указанному времени
 - Показывает только соответствующие события
 - Форматирует ответ с учетом временных ограничений

AC-VIEW-003: Отсутствие событий

- **Дано:** Пользователь запрашивает события
- **Когда:** На указанный период нет событий
- **Тогда:**
 - Бот сообщает об отсутствии событий
 - Указывает период, по которому искал
 - Возвращает в главное меню

AC-VIEW-004: Сложные временные запросы

- **Дано:** Пользователь запрашивает события
- **Когда:** Пользователь использует сложные формулировки ("на этой неделе", "в пятницу вечером", "до 12:00 завтра")
- **Тогда:**
 - DeepSeek API корректно определяет временной диапазон

- Бот получает события за правильный период
- Форматирует ответ с учетом сложного запроса

AC-VIEW-005: Ошибка получения событий

- **Дано:** Пользователь запрашивает события
 - **Когда:** Происходит ошибка при работе с базой данных
 - **Тогда:**
 - Бот сообщает об ошибке получения
 - Ошибка логируется
 - Возвращает в главное меню
-

Критерии приемки для удаления событий (US003/UC-003-MVP)

AC-DEL-001: Успешное удаление с подтверждением

- **Дано:** Пользователь нажимает "☐ Удалить событие"
- **Когда:** Пользователь описывает событие для удаления
- **Тогда:**
 - Бот находит событие по описанию
 - Показывает детали найденного события
 - Запрашивает подтверждение удаления
 - После подтверждения удаляет событие
 - Подтверждает успешное удаление

AC-DEL-002: Удаление по дате

- **Дано:** Пользователь описывает событие для удаления
- **Когда:** Пользователь указывает только дату без названия
- **Тогда:**
 - Бот находит все события на указанную дату
 - Показывает первое найденное событие
 - Удаляет все события на дату при подтверждении
 - Сообщает количество удаленных событий

AC-DEL-003: Событие не найдено

- **Дано:** Пользователь описывает событие для удаления
- **Когда:** Событие по указанным критериям не найдено
- **Тогда:**
 - Бот сообщает, что событие не найдено
 - Указывает дату, по которой искал
 - Возвращает в главное меню

AC-DEL-004: Отмена удаления

- **Дано:** Бот запрашивает подтверждение удаления
- **Когда:** Пользователь нажимает "☐ Нет"
- **Тогда:**
 - Бот отменяет операцию удаления
 - Сообщает об отмене
 - Возвращает в главное меню

AC-DEL-005: Ошибка при удалении

- **Дано:** Пользователь подтверждает удаление
 - **Когда:** Происходит ошибка при удалении из базы
 - **Тогда:**
 - Бот сообщает об ошибке удаления
 - Ошибка логируется
 - Возвращает в главное меню
-

Критерии приемки для изменения событий (US003/UC-004-MVP)

AC-EDIT-001: Успешное изменение с предпросмотром

- **Дано:** Пользователь нажимает "⇒ □ Изменить событие"
- **Когда:** Пользователь описывает изменения
- **Тогда:**
 - Бот находит событие для изменения
 - Извлекает изменения через DeepSeek API
 - Показывает старые и новые значения
 - Запрашивает подтверждение
 - После подтверждения сохраняет изменения

AC-EDIT-002: Изменение названия события

- **Дано:** Пользователь изменяет событие
- **Когда:** Пользователь изменяет только название
- **Тогда:**
 - Бот находит событие по контексту
 - Показывает изменение названия
 - Сохраняет новое название после подтверждения

AC-EDIT-003: Изменение времени события

- **Дано:** Пользователь изменяет событие
- **Когда:** Пользователь переносит событие на другое время
- **Тогда:**
 - Бот определяет новое время через DeepSeek API
 - Показывает изменение времени
 - Сохраняет новое время после подтверждения

AC-EDIT-004: Изменение нескольких параметров

- **Дано:** Пользователь изменяет событие
- **Когда:** Пользователь изменяет несколько параметров одновременно
- **Тогда:**
 - Бот извлекает все изменения
 - Показывает все изменения в предпросмотре
 - Сохраняет все изменения после подтверждения

AC-EDIT-005: Событие не найдено для изменения

- **Дано:** Пользователь описывает изменения
- **Когда:** Бот не может идентифицировать какое событие изменить
- **Тогда:**
 - Бот сообщает, что событие не найдено
 - Возвращает в главное меню

AC-EDIT-006: Нечего изменять

- **Дано:** Пользователь описывает изменения
- **Когда:** DeepSeek не может извлечь изменения
- **Тогда:**
 - Бот сообщает, что нечего изменять
 - Возвращает в главное меню

AC-EDIT-007: Отмена изменения

- **Дано:** Бот запрашивает подтверждение изменения
- **Когда:** Пользователь нажимает "❑ Нет"
- **Тогда:**
 - Бот отменяет операцию изменения
 - Сообщает об отмене
 - Возвращает в главное меню

AC-EDIT-008: Ошибка при сохранении изменений

- **Дано:** Пользователь подтверждает изменения
 - **Когда:** Происходит ошибка при сохранении в базу
 - **Тогда:**
 - Бот сообщает об ошибке сохранения
 - Ошибка логируется
 - Возвращает в главное меню
-

Форматы данных и ограничения

AC-FORMAT-001: Формат даты и времени

- **Дано:** Любая операция с датой/временем
- **Когда:** Бот работает с датами
- **Тогда:** Используются форматы:
 - Входные данные: свободный текст
 - Внутренний формат: YYYY-MM-DD HH:MM
 - ISO формат для базы: YYYY-MM-DDTHH:MM:SS
 - Отображение пользователю: DD.MM.YYYY HH:MM

AC-FORMAT-002: Ограничения полей

- **Дано:** Создание или изменение события
- **Когда:** Пользователь вводит данные
- **Тогда:** Применяются ограничения:
 - Описание: до 500 символов
 - Место: до 200 символов

AC-FORMAT-003: Обработка null значений

- **Дано:** Любая операция с данными
- **Когда:** DeepSeek возвращает null значения
- **Тогда:**
 - Бот корректно обрабатывает строковые "null"
 - Преобразует их в Python None

- Не сохраняет null значения в обязательные поля

Нефункциональные требования

Требования производительности

NFR-PERF-001: Время отклика бота

- **Описание:** Бот должен быстро обрабатывать пользовательские запросы
- **Метрика:** Время отклика на команды пользователя
- **Целевое значение:** ≤ 3 секунды для 95% запросов
- **Критическое значение:** ≤ 5 секунд для 99% запросов
- **Измерение:** От момента получения сообщения до отправки ответа

NFR-PERF-002: Время обработки DeepSeek API

- **Описание:** Время обработки запросов к DeepSeek API
- **Метрика:** Время выполнения API вызова
- **Целевое значение:** ≤ 2 секунды
- **Критическое значение:** ≤ 5 секунд
- **Измерение:** От отправки запроса до получения ответа

NFR-PERF-003: Производительность базы данных

- **Описание:** Время выполнения операций с базой данных
- **Метрика:** Время выполнения CRUD операций
- **Целевое значение:** ≤ 500 мс
- **Критическое значение:** ≤ 1 секунда
- **Измерение:** От отправки запроса до получения ответа от Supabase

NFR-PERF-004: Пропускная способность

- **Описание:** Количество одновременных пользователей
- **Метрика:** Максимальное количество параллельных запросов
- **Целевое значение:** Поддержка 1000 активных пользователей
- **Критическое значение:** Поддержка 5000 активных пользователей
- **Измерение:** Количество одновременных сессий

NFR-PERF-005: Время запуска бота

- **Описание:** Время запуска и инициализации бота
- **Метрика:** Время от запуска до готовности принимать сообщения
- **Целевое значение:** ≤ 10 секунд
- **Критическое значение:** ≤ 30 секунд
- **Измерение:** От запуска приложения до готовности

Требования надежности

NFR-REL-001: Доступность системы

- **Описание:** Общая доступность бота для пользователей
- **Метрика:** Uptime (время доступности)
- **Целевое значение:** 99.5% в месяц
- **Критическое значение:** 99.9% в месяц
- **Измерение:** Отношение времени работы к общему времени

NFR-REL-002: Восстановление после сбоев

- **Описание:** Время восстановления после сбоев
- **Метрика:** Mean Time To Recovery (MTTR)
- **Целевое значение:** ≤ 15 минут
- **Критическое значение:** ≤ 1 час
- **Измерение:** От момента сбоя до полного восстановления

NFR-REL-003: Обработка ошибок внешних сервисов

- **Описание:** Поведение при недоступности внешних API
- **Метрика:** Время ожидания и повторных попыток
- **Целевое значение:**
 - Таймаут запросов: 15 секунд
 - Количество повторных попыток: 3
 - Экспоненциальная задержка между попытками
- **Измерение:** Логирование ошибок и статистика повторных попыток

NFR-REL-004: Целостность данных

- **Описание:** Сохранность данных при сбоях
- **Метрика:** Потеря данных при сбоях
- **Целевое значение:** 0% потерь данных
- **Критическое значение:** $\leq 0.1\%$ потерь данных
- **Измерение:** Аудит операций с базой данных

NFR-REL-005: Мониторинг и логирование

- **Описание:** Полнота мониторинга и логирования
- **Метрика:** Покрытие логированием критических операций
- **Целевое значение:** 100% критических операций
- **Измерение:** Наличие логов для всех операций с событиями

Требования безопасности

NFR-SEC-001: Защита данных пользователей

- **Описание:** Безопасное хранение пользовательских данных
- **Метрика:** Шифрование данных в базе
- **Целевое значение:**
 - Шифрование данных в rest (Supabase)
 - HTTPS для всех API вызовов
 - Безопасное хранение токенов
- **Измерение:** Аудит безопасности конфигурации

NFR-SEC-002: Аутентификация и авторизация

- **Описание:** Проверка прав доступа пользователей

- **Метрика:** Механизмы контроля доступа
- **Целевое значение:**
 - Аутентификация через Telegram
 - Проверка telegram_id для всех операций
 - Изоляция данных между пользователями
- **Измерение:** Тестирование несанкционированного доступа

NFR-SEC-003: Защита от атак

- **Описание:** Защита от распространенных веб-атак
- **Метрика:** Наличие защитных механизмов
- **Целевое значение:**
 - Валидация входных данных
 - Защита от SQL-инъекций (ORM Supabase)
 - Ограничение частоты запросов
- **Измерение:** Пентест и security audit

NFR-SEC-004: Безопасность API ключей

- **Описание:** Защита конфиденциальных данных
 - **Метрика:** Способ хранения API ключей
 - **Целевое значение:**
 - Хранение в переменных окружения
 - Отсутствие ключей в коде
 - Регулярная ротация ключей
 - **Измерение:** Аудит конфигурации
-

Требования удобства использования

NFR-USAB-001: Простота взаимодействия

- **Описание:** Интуитивно понятный интерфейс
- **Метрика:** Количество шагов для выполнения основных операций
- **Целевое значение:**
 - Добавление события: ≤ 3 шага
 - Просмотр событий: ≤ 2 шага
 - Удаление события: ≤ 4 шага
- **Измерение:** User testing и аналитика

NFR-USAB-002: Качество обработки естественного языка

- **Описание:** Точность распознавания пользовательских сообщений
- **Метрика:** Точность извлечения данных из текста
- **Целевое значение:** $\geq 90\%$ точности для основных полей
- **Критическое значение:** $\geq 80\%$ точности
- **Измерение:** Тестирование на различных формулировках

NFR-USAB-003: Обратная связь пользователю

- **Описание:** Информативность сообщений бота
- **Метрика:** Понятность сообщений об ошибках и статусах
- **Целевое значение:** 100% понятных сообщений
- **Измерение:** User testing и опросы

NFR-USAB-004: Время обучения

- **Описание:** Время освоения функционала новым пользователем
 - **Метрика:** Время до первого успешного использования
 - **Целевое значение:** ≤ 2 минуты
 - **Критическое значение:** ≤ 5 минут
 - **Измерение:** User testing с новыми пользователями
-

Требования совместимости

NFR-COMP-001: Совместимость с Telegram

- **Описание:** Поддержка различных версий Telegram
- **Метрика:** Совместимость с клиентами Telegram
- **Целевое значение:**
 - Поддержка Telegram Web/Desktop/Mobile
 - Совместимость с последними 3 версиями Telegram API
- **Измерение:** Тестирование на различных платформах

NFR-COMP-002: Совместимость с Python

- **Описание:** Поддержка версий Python
- **Метрика:** Совместимость с Python версиями
- **Целевое значение:** Python 3.8+
- **Критическое значение:** Python 3.9+
- **Измерение:** Тестирование на целевых версиях

NFR-COMP-003: Совместимость с операционными системами

- **Описание:** Поддержка различных ОС для развертывания
 - **Метрика:** Совместимость с популярными ОС
 - **Целевое значение:**
 - Linux (Ubuntu 20.04+)
 - Windows Server 2019+
 - Docker контейнеризация
 - **Измерение:** Тестирование развертывания
-

Требования масштабируемости

NFR-SCAL-001: Масштабируемость пользовательской базы

- **Описание:** Возможность обслуживания растущего числа пользователей
- **Метрика:** Максимальное количество пользователей
- **Целевое значение:** Поддержка 10,000 пользователей
- **Критическое значение:** Поддержка 50,000 пользователей
- **Измерение:** Load testing и мониторинг производительности

NFR-SCAL-002: Масштабируемость базы данных

- **Описание:** Производительность при росте объема данных
- **Метрика:** Время выполнения запросов при росте данных
- **Целевое значение:**
 - ≤ 1 секунда при 100,000 событий
 - ≤ 2 секунды при 1,000,000 событий
- **Измерение:** Тестирование с большими объемами данных

NFR-SCAL-003: Горизонтальное масштабирование

- **Описание:** Возможность масштабирования путем добавления инстансов
- **Метрика:** Поддержка горизонтального масштабирования
- **Целевое значение:**
 - Stateless архитектура
 - Поддержка нескольких инстансов бота
 - Балансировка нагрузки
- **Измерение:** Тестирование с несколькими инстансами

NFR-SCAL-004: Эффективность использования ресурсов

- **Описание:** Оптимальное использование вычислительных ресурсов
 - **Метрика:** Потребление памяти и CPU
 - **Целевое значение:**
 - Память: ≤ 512 MB на инстанс
 - CPU: ≤ 1 vCPU на инстанс
 - Сеть: ≤ 100 MB/час на 1000 пользователей
 - **Измерение:** Мониторинг ресурсов в production
-

Требования сопровождения

NFR-MAIN-001: Логируемость

- **Описание:** Полнота и структурированность логов
- **Метрика:** Качество логов для отладки
- **Целевое значение:**
 - Структурированные логи (JSON)
 - Логирование всех критических операций
 - Уровни логирования (DEBUG, INFO, ERROR)
 - Контекст в логах (user_id, operation_id)
- **Измерение:** Аудит логов

NFR-MAIN-002: Мониторинг

- **Описание:** Наличие системы мониторинга
- **Метрика:** Покрытие метриками
- **Целевое значение:**
 - Мониторинг доступности
 - Мониторинг производительности
 - Мониторинг ошибок
 - Алертинг при проблемах
- **Измерение:** Наличие дашбордов и алертов

NFR-MAIN-003: Документация

- **Описание:** Полнота технической документации

- **Метрика:** Качество документации
- **Целевое значение:**
 - Документация развертывания
 - Документация API
 - Документация мониторинга
 - Руководство по устранению неисправностей
- **Измерение:** Проверка документации

NFR-MAIN-004: Простота развертывания

- **Описание:** Время и сложность развертывания
- **Метрика:** Время развертывания
- **Целевое значение:** ≤ 30 минут для нового окружения
- **Критическое значение:** ≤ 1 час
- **Измерение:** Время от клонирования репозитория до работающего бота