1. **Use Case**

|  |  |
| --- | --- |
| Заголовок | Формирование расписания через AI-агента в telegram |
| Акторы | Пользователь telegram |
| Предусловия | - Пользователь запустил Telegram-бота  - Система авторизована и подключена к базе данных PostgreSQL  - AI-модель доступна для обработки запросов  - В базе данных существуют необходимые таблицы |
| Ограничения |  |
| Триггер | Пользователь вводит команду /start |
| Основной сценарий | 1. Система отображает на выбор просмотреть текущее расписание или обновить расписание 2. Пользователь нажимает обновить расписание 3. Система отображает уведомление: «Введите событие, которое нужно добавить в расписание» 4. Пользователь отправляет запрос с обновлением расписания 5. Система отправляет уведомление о том, что расписание успешно изменено и отображает кнопку посмотреть расписание и обновить. 6. Пользователь нажимает «посмотреть расписание» 7. Система выводит текущее расписание 8. За час до мероприятия бот присылает сообщение с описанием мероприятия. 9. Каждый день в 10.00 система присылает все расписание на текущий день |
| Альтернативный сценарий | 1а. Пользователь нажимает посмотреть расписание  Переход к основному сценарию шаг 6  8f. Пользователь вводит «удалить встреча завтра»  9f. Система находит событие и удаляет его  10f. Система присылает уведомление, об успешном удалении. |
| Исключительный сценарий | 4b. Пользователь вводит запрос с бессмысленным текстом  5b. Система уведомляет, что на эту дату невозможно сформировать расписание и приводит пример корректного запроса  8c. Пользователь вводит команду удаления (например, "удали встречу завтра")  9c. Система yне находит указанное событие и сообщает, что событие не найдено  1d. Система недоступна. Уведомление с просьбой повторить запрос позже. |
| Результат | Результат: У пользователя формируется актуальное расписание с возможностью добавления, просмотра, удаления событий и получения автоматических уведомлений. |