

AI-диетолог 3.0

1. User story

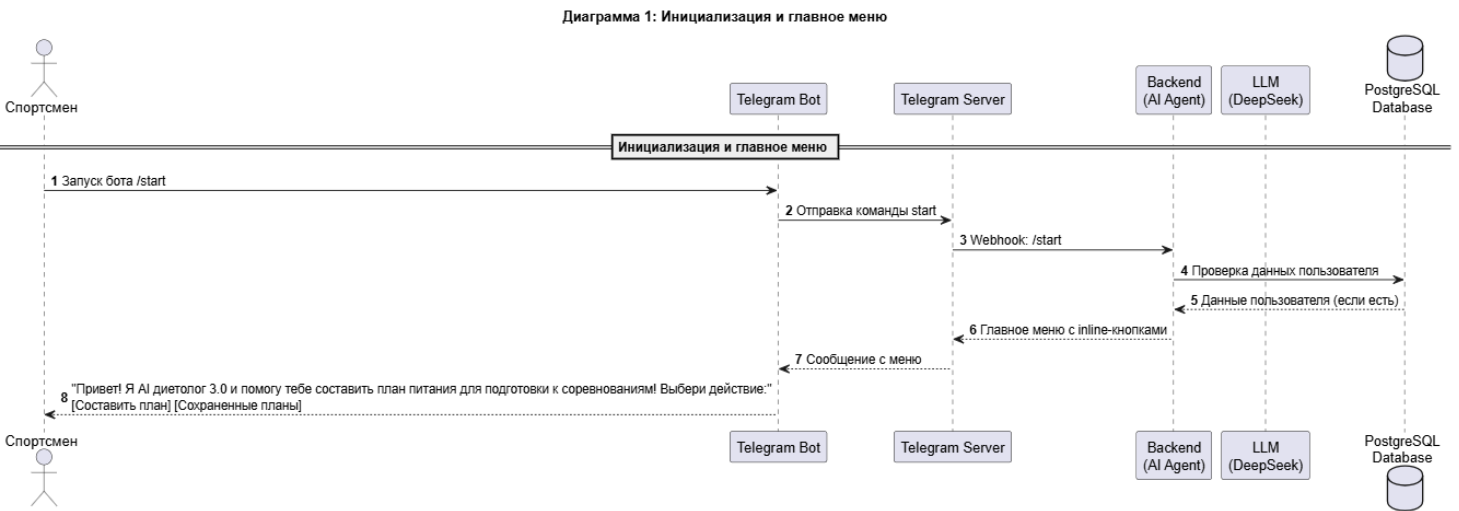
Как спортсмен, я хочу получить от ИИ агента еженедельный план питания, чтобы подготовиться к выступлению на соревнованиях в требуемой весовой категории.

2. Диаграммы последовательностей

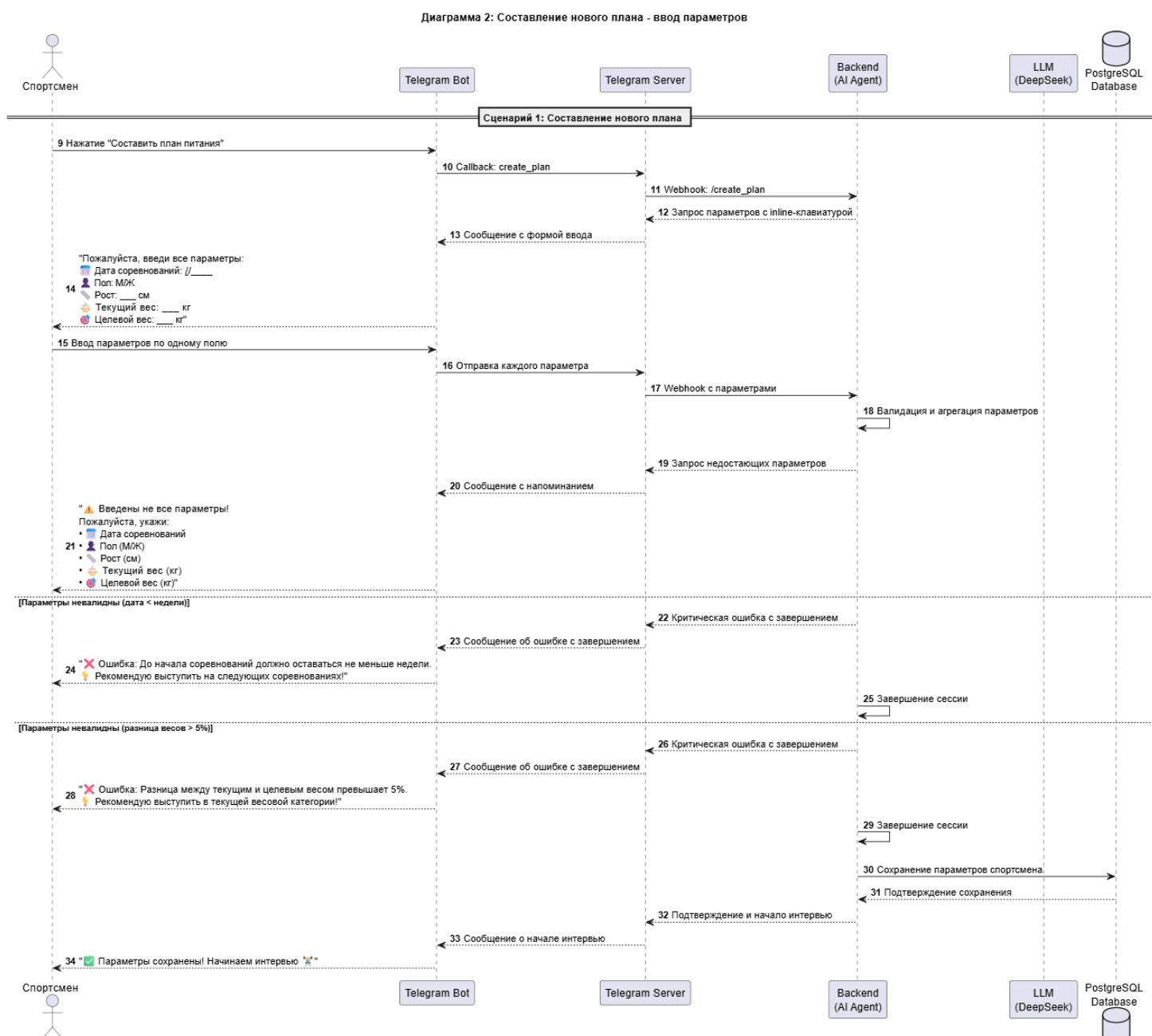
Описание последовательности взаимодействия спортсмена с AI диетологом 3.0

1. Инициализация и главное меню

- Спортсмен запускает бота командой /start
- Telegram Bot отправляет команду start на сервер Telegram
- Сервер Telegram через webhook передает запрос /start в Backend
- Backend проверяет данные пользователя в базе данных PostgreSQL
- Если данные есть, возвращаются данные пользователя
- Backend отправляет главное меню с inline-кнопками
- Пользователь получает приветственное сообщение с выбором действий: [Составить план] [Сохраненные планы]



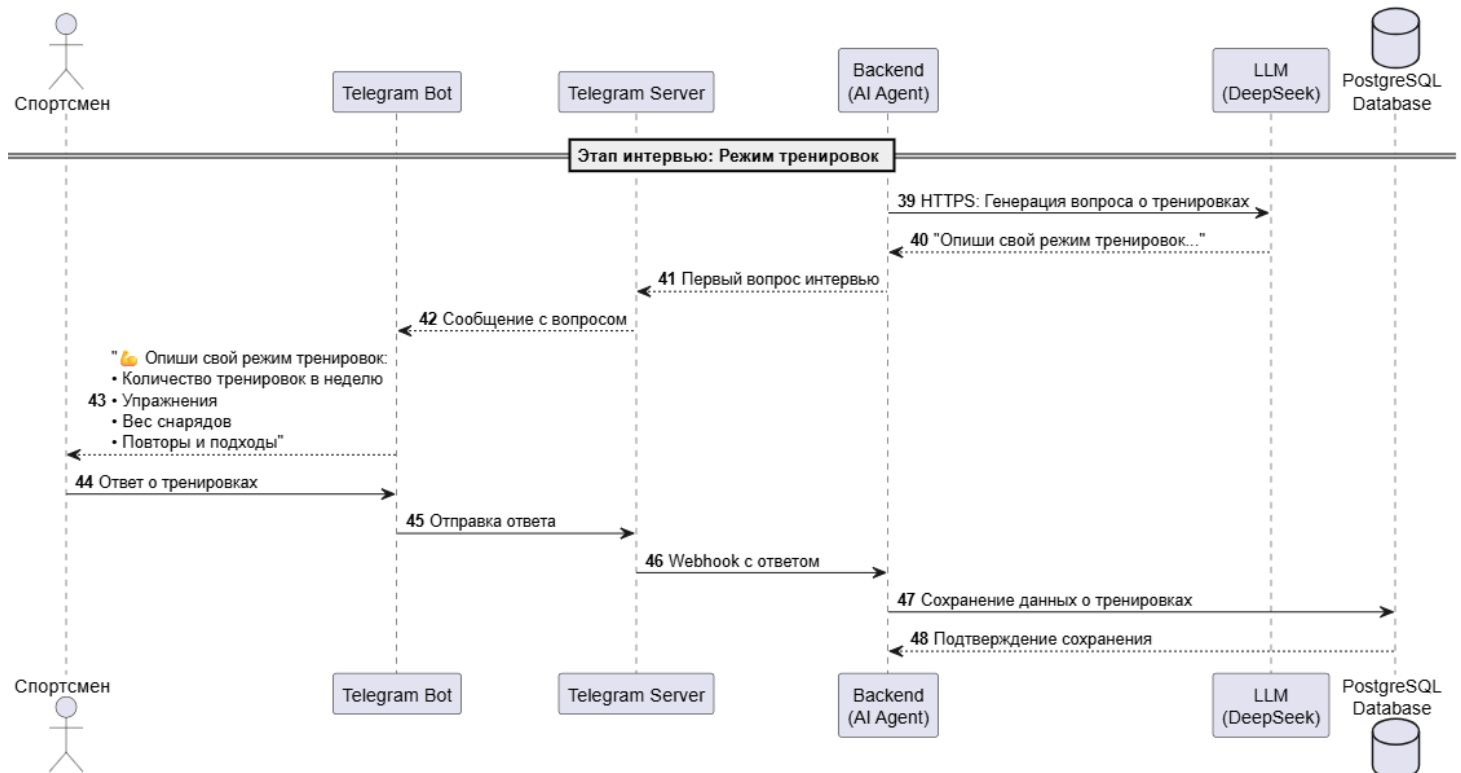
- Пользователь нажимает "Составить план питания"
- Отправляется callback: create_plan через Telegram сервер
- Backend получает webhook: /create_plan
- Backend запрашивает параметры с inline-клавиатурой
- Пользователь вводит параметры по одному полю:
 - Дата соревнований
 - Пол (М/Ж)
 - Рост (см)
 - Текущий вес (кг)
 - Целевой вес (кг)
- Backend валидирует и агрегирует параметры
- Если не все параметры введены - запрос недостающих данных
- Если параметры невалидны (дата < недели или разница весов > 5%) - ошибка с завершением
- Если параметры валидны - сохранение в БД и начало интервью



3. Этап интервью: Режим тренировок

- Backend генерирует вопрос о тренировках через LLM (DeepSeek)
- LLM возвращает сформулированный вопрос
- Пользователю задается вопрос о режиме тренировок:
 - Количество тренировок в неделю
 - Упражнения
 - Вес снарядов
 - Повторы и подходы
- Пользователь отвечает о тренировках
- Ответ сохраняется в базе данных

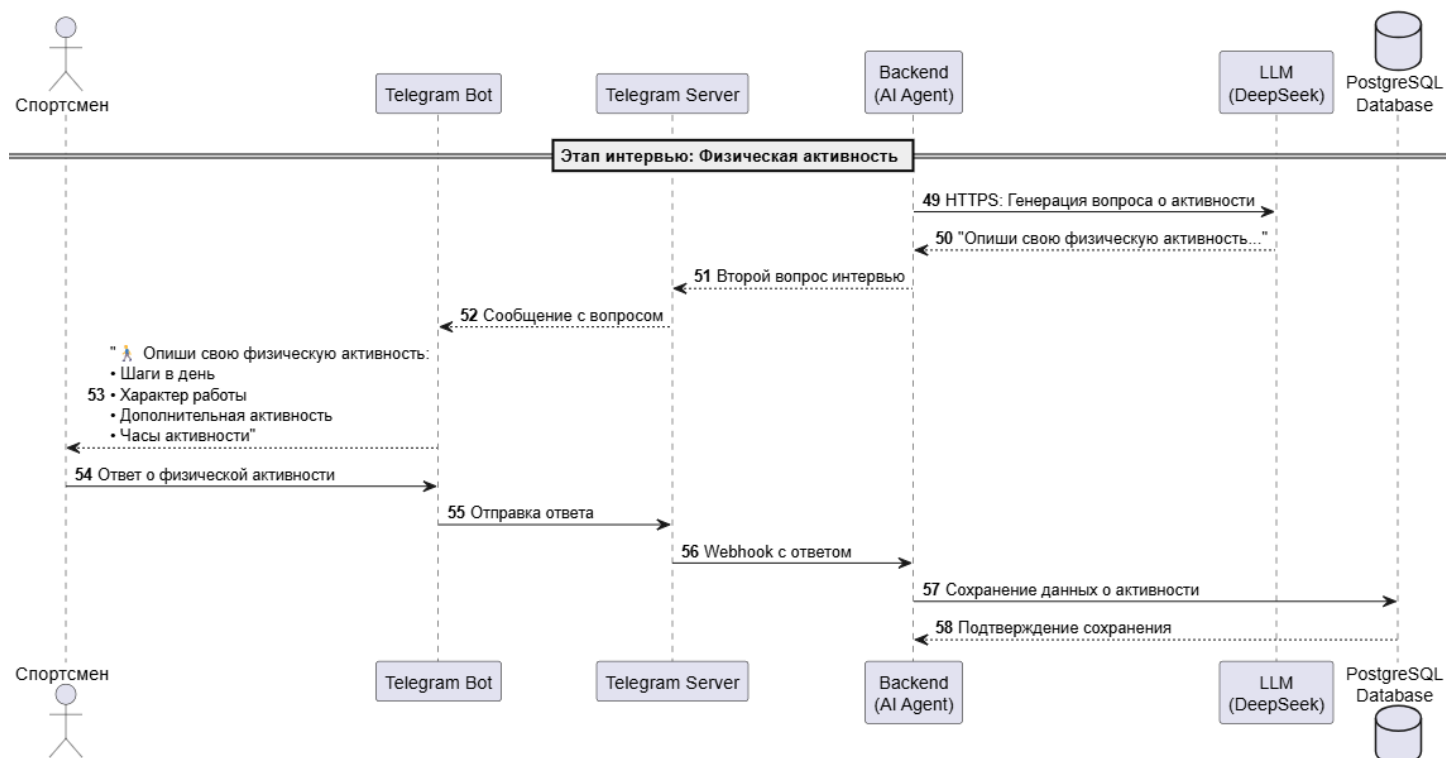
Диаграмма 3: Этап интервью - Режим тренировок



4. Этап интервью: Физическая активность

- Backend генерирует вопрос о физической активности через LLM
- LLM возвращает сформулированный вопрос
- Пользователю задается вопрос о физической активности:
 - Шаги в день
 - Характер работы
 - Дополнительная активность
 - Часы активности
- Пользователь отвечает о физической активности
- Ответ сохраняется в базе данных

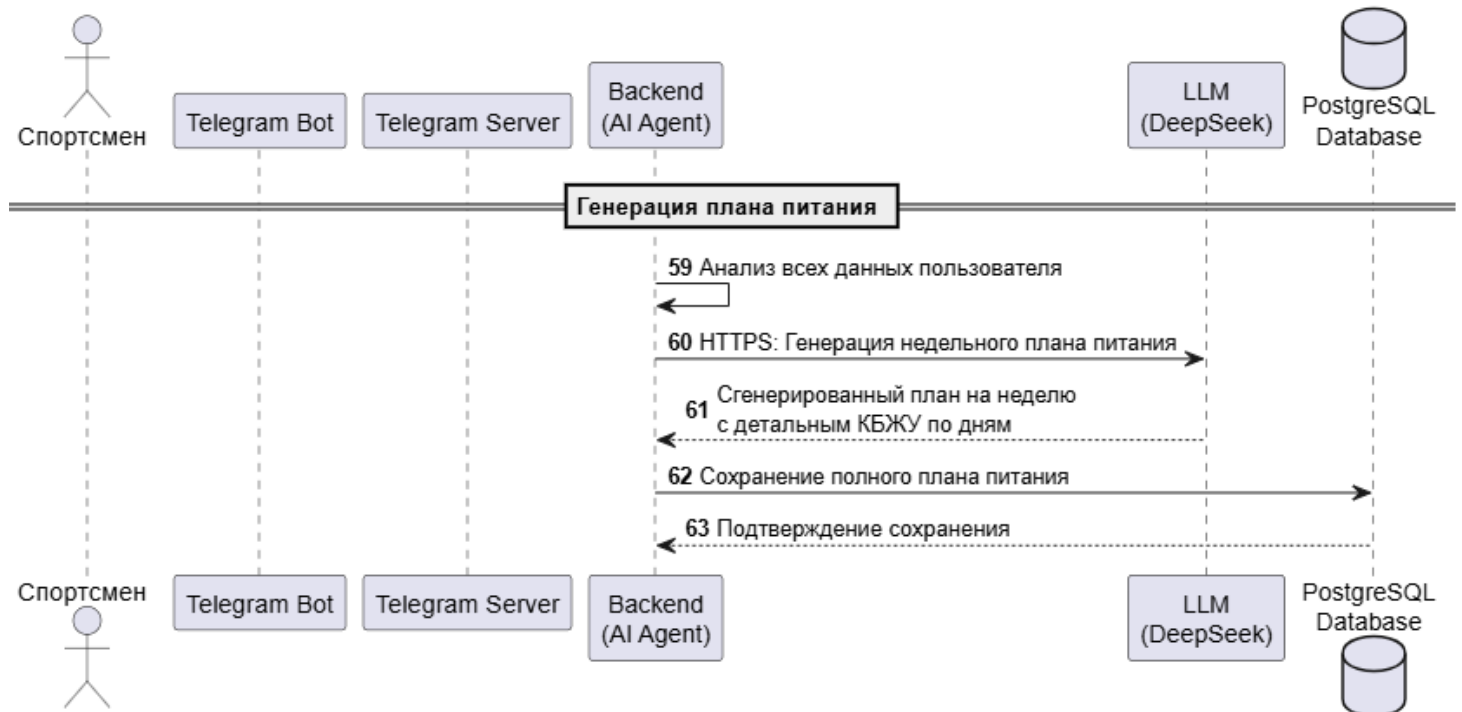
Диаграмма 4: Этап интервью - Физическая активность



5. Генерация плана питания

- Backend анализирует все данные пользователя
- Backend отправляет запрос на генерацию недельного плана питания в LLM
- LLM возвращает сгенерированный план на неделю с детальным КБЖУ по дням
- Полный план питания сохраняется в базе данных

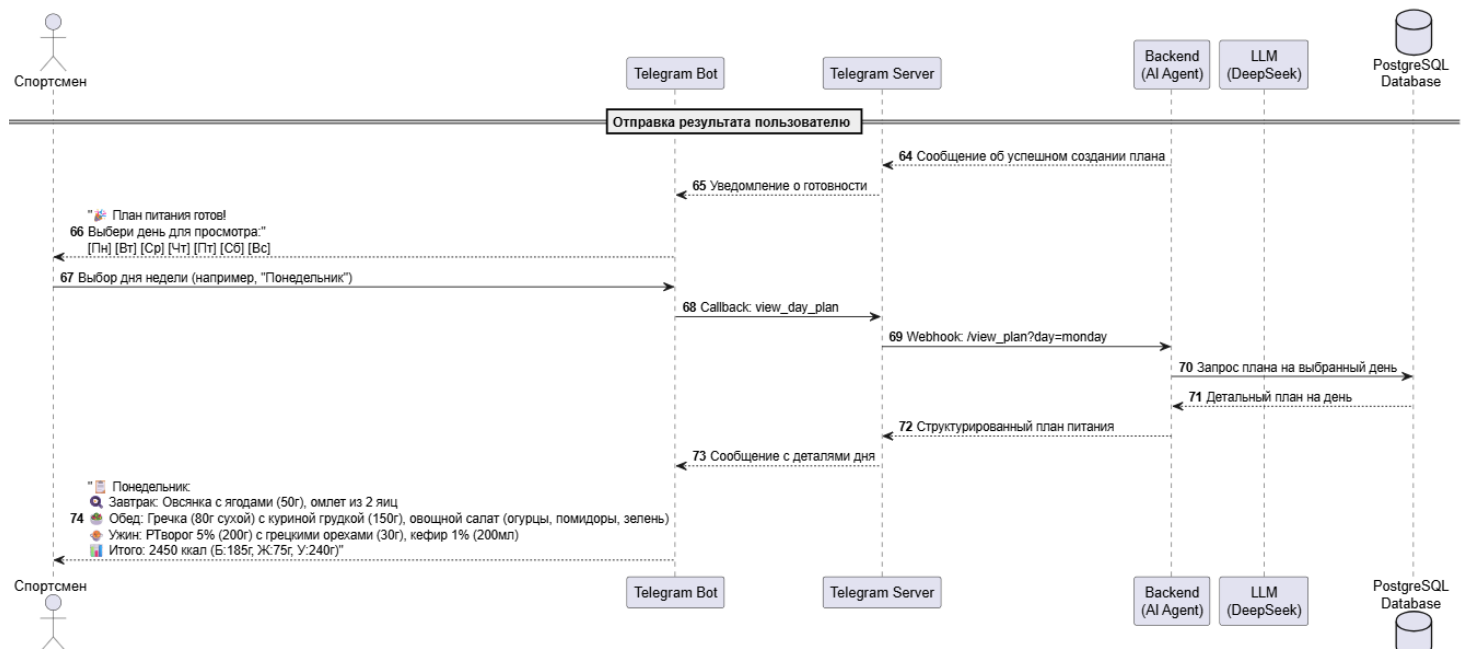
Диаграмма 5: Генерация плана питания



6. Отправка результата пользователю

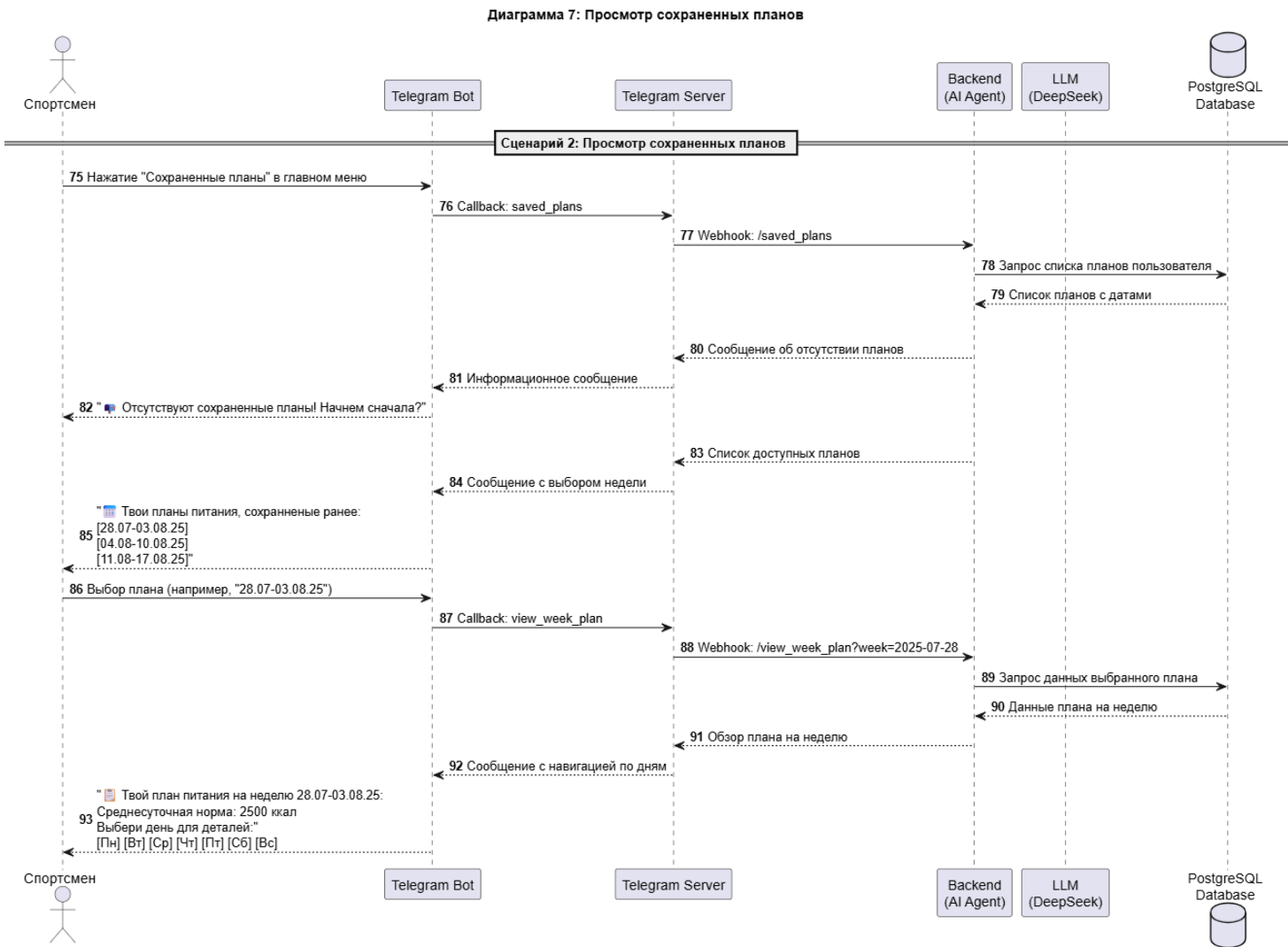
- Backend отправляет сообщение об успешном создании плана
- Пользователь получает уведомление о готовности плана
- Предлагается выбрать день для просмотра: [Пн] [Вт] [Ср] [Чт] [Пт] [Сб] [Вс]
- Пользователь выбирает день недели
- Backend запрашивает план на выбранный день из БД
- Пользователь получает детализированный план питания на выбранный день

Диаграмма 6: Отправка результата пользователю



7. Просмотр сохраненных планов

- Пользователь нажимает "Сохраненные планы" в главном меню
- Backend запрашивает список планов пользователя из БД
- Если список пустой - сообщение об отсутствии планов
- Если список не пустой - отображается список доступных планов с датами
- Пользователь выбирает конкретный план
- Backend запрашивает данные выбранного плана
- Пользователь получает обзор плана на неделю с навигацией по дням



8. Завершение сессии

- Backend очищает временные данные сессии
- Отправляется сообщение о завершении сессии
- Пользователь получает финальное сообщение с пожеланием удачи на соревнованиях

Диаграмма 8: Завершение сессии

