**Техническое задание на разработку бота-помощника для формирования багажа**

Цель проекта — разработать бота-помощника, который поможет пользователю сформировать список вещей для поездки. Бот будет интегрирован в существующее приложение S7 (super app) и будет предоставлять следующие функции:

- Формирование списка обязательных вещей.

- Рекомендации по вещам в зависимости от длительности поездки, погодных условий и других факторов.

- Помощь в упаковке чемодана с проверкой веса и ограничений на провоз вещей.

**1. Функциональные требования**

1. Основные функции бота

* Сбор информации от пользователя:

- Пункт назначения (город/страна).

- Длительность поездки (количество дней).

- Цель поездки (отпуск, командировка, и т.д.).

- Погодные условия (если пользователь не знает, бот может запросить доступ к геолокации или использовать API погоды).

- Ограничения на провоз вещей (например, ручная кладь или багаж).

* Формирование списков:

- Обязательные вещи: базовый список вещей, которые нужно взять в любую поездку (документы, деньги, телефон, зарядка и т.д.).

- Рекомендации: дополнительные вещи в зависимости от:

- Длительности поездки (например, больше одежды для длительных поездок).

- Погодных условий (зонт, теплая одежда, солнцезащитный крем и т.д.).

- Ограничения: бот должен учитывать ограничения на провоз вещей (например, жидкостей в ручной клади).

* Помощь в упаковке:

- Бот должен проверять, можно ли взять ту или иную вещь (например, запрещенные предметы или превышение веса).

- Расчет примерного веса чемодана и сравнение с ограничениями авиакомпании или другими транспортными правилами.

* Интерактивное взаимодействие:

- Пользователь может добавлять или удалять вещи из списка.

- Бот должен предоставлять подсказки и советы по упаковке.

2. Интеграция с внешними сервисами

1. API погоды: для получения актуальных данных о погоде в пункте назначения.

2. API авиакомпаний/транспортных компаний: для получения информации об ограничениях на багаж.

3. Геолокация: для автоматического определения пункта назначения и погодных условий.

**2. Технические требования**

**Бэкенд**

1. Архитектура:

- Микросервисная архитектура для обеспечения масштабируемости и гибкости.

- RESTful API для взаимодействия с фронтендом и внешними сервисами.

2. Основные модули

- Модуль сбора данных: обработка вводных данных от пользователя (пункт назначения, длительность поездки и т.д.).

- Модуль формирования списков:

- База данных с базовыми списками вещей (обязательные и рекомендации).

- Логика для формирования списков на основе вводных данных.

- Модуль проверки ограничений:

- Интеграция с API авиакомпаний для получения ограничений на багаж.

- Логика для проверки веса и запрещенных предметов.

- Модуль взаимодействия с внешними сервисами:

- Интеграция с API погоды.

- Интеграция с API геолокации.

3. База данных *(предлагаю SQLite или PostgreSQL)*

- Хранение базовых списков вещей.

- Хранение пользовательских данных (данные о поездке, предыдущие списки и т.д.).

4. Логика работы

- Бот должен быть способен обрабатывать запросы в реальном времени.

- Логика формирования списков должна быть гибкой и учитывать множество факторов (погода, длительность поездки и т.д.).

**Фронтенд**

5. Внешняя оболочка

* Интерфейс:

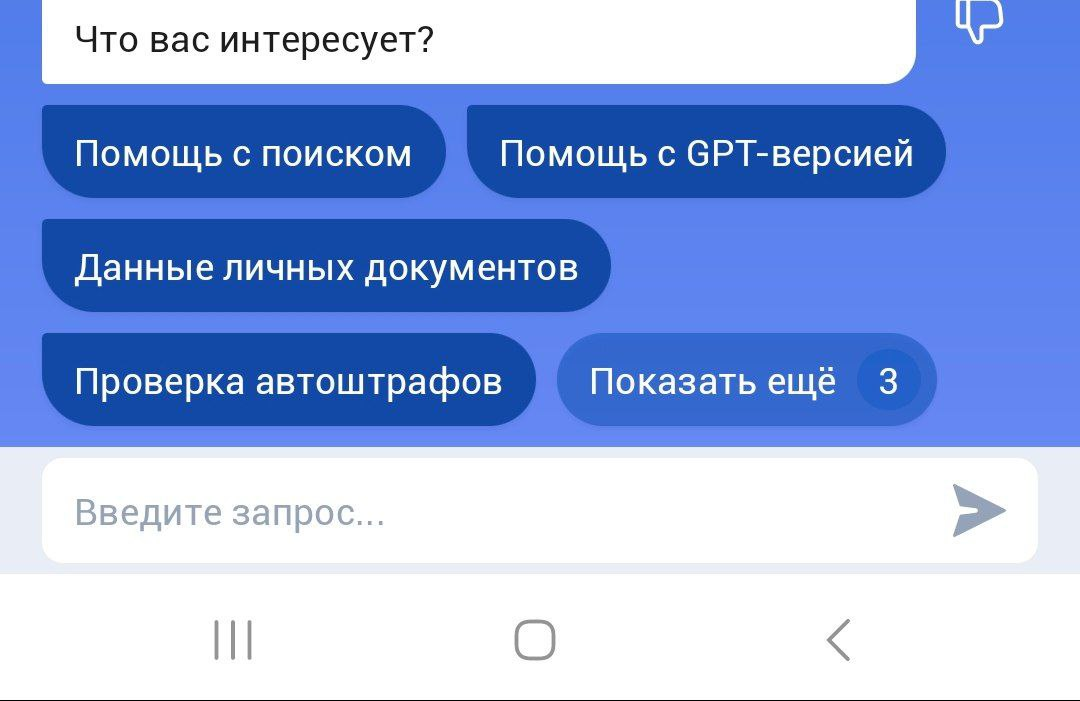
- Чат-интерфейс с возможностью ввода текста и выбора из предложенных вариантов.

- Кнопки для быстрого выбора ("Добавить вещь", "Удалить вещь", "Проверить вес" и тд).

- *Отображение списков вещей с возможностью отметки (галочки?)*.

* Дизайн:

- Минималистичный и интуитивно понятный интерфейс.



**3. Команды и кнопки**

1. Основные команды:

- `/start` — начало взаимодействия с ботом(после этого бот выводит кнопки).

- `/help` — справка по использованию бота.

2. Кнопки:

- "Начать сбор багажа" — запуск процесса формирования списка.

- "Добавить вещь" — открывает поле для ввода новой вещи.

- "Удалить вещь" — позволяет выбрать и удалить вещь из списка.

- "Проверить вес" — расчет веса и проверка ограничений.

- "Рекомендации" — отображение дополнительных рекомендаций.

- "Не забудьте в аэропорт" — отображение списка обязательного (документы, билеты, телефон…)

**4. USE Case**

1. Сценарий 1: Формирование списка обязательных вещей

- Пользователь запускает бота (он выдает свои функции кнопками, как скрин с госуслуг) и выбирает.

- Бот формирует список обязательных вещей и отображает его в чате.

2. Сценарий 2: Добавление рекомендаций

- Бот запрашивает дополнительную информацию *(из БД (по билетам пользователя) или у самого пользователя?)* (погода, длительность поездки).

- На основе этих данных бот добавляет рекомендации в список.

3. Сценарий 3: Проверка веса и ограничений

- Пользователь добавляет вещи в список.

- Бот проверяет вес и ограничения, сообщает о возможных проблемах (нельзя брать эту вещь или перевес).

**5. Этапы разработки?**

Этап 1: Проектирование и разработка бэкенда *(в конструкторе в итоге?).*

Этап 2: Интеграция с внешними API.

Этап 3: Разработка интерфейса и интеграция в super app*(если сможем)*.

Этап 4: Тестирование и доработка.