Планирование поездок

**1. Описание проекта:**  
**1.1 Цель проекта:** Разработка веб-приложения для планирования поездок, позволяющего пользователям создавать, редактировать и управлять поездками, включая поиск доступных вариантов транспорта, бронирование билетов и отелей, а также управление расходами.

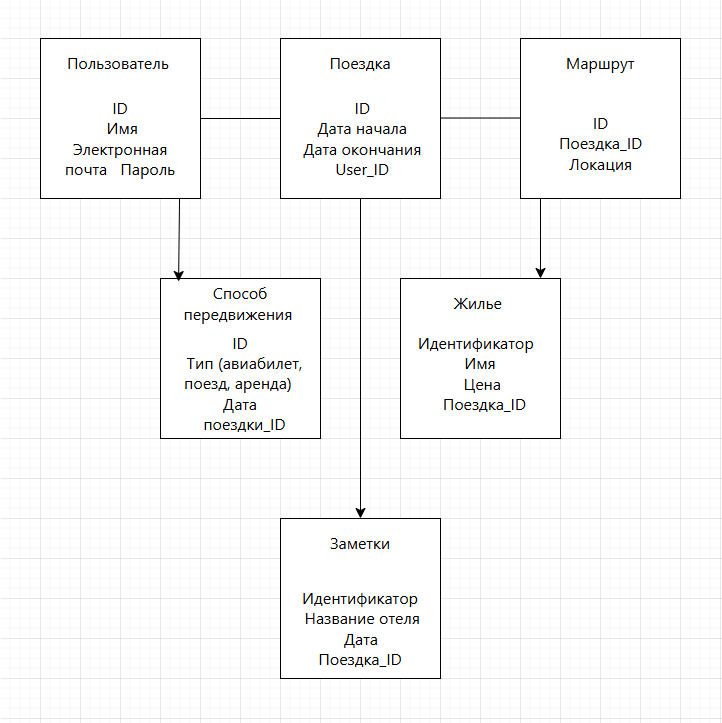
**1.2 Функциональные требования:**

* **Создание поездок:** Пользователи должны иметь возможность создавать новые поездки, добавляя даты, места назначения, маршруты и предполагаемые расходы.
* **Добавление точек:** Пользователи могут добавлять промежуточные точки и останавливаться на маршруте.
* **Поиск транспорта:** Интеграция с базами данных о транспорте (авиакомпании, ж/д перевозчики, автобусные компании) для поиска доступных вариантов, включая сравнение цен и расписаний.
* **Бронирование:** Возможность бронирования билетов и отелей, получение подтверждающих документов. Интеграция с API сервисов бронирования (Booking, Expedia, и др.).
* **Управление расходами:** Запись предполагаемых расходов, их категоризация и возможность отслеживания затрат в процессе поездки.
* **Отчеты:** Возможность генерации отчетов о поездках, включая подробные списки расходов, маршруты и бронирование.
* **Автоматизация:** Возможность автоматического планирования маршрута (в зависимости от выбранного транспорта).

**1.3 Нефункциональные требования:**

* **Удобство использования:** Приложение должно быть интуитивно понятным и простым в использовании.
* **Безопасность:** Данные пользователей должны быть защищены.
* **Масштабируемость:** Приложение должно быть способно обрабатывать большое количество пользователей и данных.
* **Производительность:** Приложение должно быстро обрабатывать запросы пользователей.

2.



3. Языки и технологии  
  
- **JavaScript/TypeScript**: Используется как на серверной (Node.js), так и на клиентской (React/Vue.js) стороне разработки. Это позволяет использовать одни и те же языки и подходы для разработки всего приложения.  
- **Node.js**: Выбор для создания серверной части, так как он позволяет обрабатывать множество параллельных соединений и обеспечивает высокую производительность.  
- **Express.js**: Упрощает разработку API и маршрутизацию на стороне сервера.  
- **MongoDB**: Носит документно-ориентированный характер, что удобно для хранения данных о поездках и пользователях, что позволяет быстро изменять и добавлять новые поля.  
- **React/Vue.js**: Для построения интерактивного пользовательского интерфейса, который легко поддерживать благодаря компонентному подходу.  
- **API Google Maps/OpenStreetMap**: Чтобы интегрировать карты и маршруты в приложение.