

## **Система бронирования отелей**

Веб-приложение, предназначено для поиска, выбора и бронирования номеров в отелях. Оно объединяет информацию о доступных отелях, номерах, ценах и условиях проживания, предоставляя пользователям простой интерфейс для быстрого оформления брони. Система автоматизирует процесс бронирования, минимизирует ручную работу администраторов и обеспечивает быстрое подтверждение бронирований. Пользователи могут сравнивать варианты размещения, просматривать детали номеров, проверять доступность на нужные даты и безопасно оплачивать бронирование онлайн.

### **Основные функции системы:**

1. Поиск и сравнение вариантов размещения по заданным параметрам: географическому расположению, датам проживания и количеству гостей.
2. Управление бронированиями.
3. Управление профилями пользователей.
4. Оплата брони.
5. Уведомления о подтверждении брони, напоминания о предстоящем заезде, а также уведомления об изменениях в условиях бронирования.

### **Описание сервисов**

#### **Фронтенд**

React. Работает в браузере пользователя, отвечает за отображение интерфейса, сбор пользовательского ввода и отправку запросов на бэкенд. Все данные фронтенд получает только через REST API, не зная о существовании других сервисов.

#### **Бэкенд**

Node.js, Express. Принимает запросы от фронтенда, проверяет авторизацию, валидирует данные и распределяет задачи между специализированными сервисами. Бэкенд знает обо всех сервисах и управляет их взаимодействием.

#### **Сервис пользователей**

Node.js, PostgreSQL. Сервис пользователей хранит учетные записи и управляет аутентификацией. Когда пользователь входит в систему, бэкенд обращается к этому сервису для проверки логина и пароля. Сервис создает JWT-токен для

последующей авторизации и хранит основную информацию о пользователе (ФИО, email, настройки). Осуществляется взаимодействие с БД и бэкендом.

### **Сервис отелей**

Node.js, MongoDB. Содержит всю информацию об отелях и номерах. При получении запроса от бэкенда он возвращает данные об отелях, проверяет доступность номеров на определенные даты и обновляет информацию о бронированиях. Это главный источник «справочной» информации в системе. Осуществляется взаимодействие с БД и бэкендом.

### **Сервис бронирований**

Node.js, PostgreSQL. Сервис отвечает за жизненный цикл бронирования. Когда пользователь хочет забронировать номер, бэкенд обращается к этому сервису. Сервис проверяет доступность номера через сервис отелей, создает предварительное бронирование, затем после успешной оплаты подтверждает его. Осуществляется взаимодействие с БД и бэкендом.

### **Платежный сервис**

Node.js, Stripe API. преобразует внутренний формат заказа в структуру, ожидаемую платежным шлюзом, и обрабатывает ответы (успех/отказ платежа). Все подтвержденные транзакции получают уникальный идентификатор Stripe, который вместе с основными метаданными (сумма, валюта, статус, timestamp) передается на бэкенд.

### **Сервис поиска**

PostgreSQL. Реализован на базе PostgreSQL и использует его полнотекстовые возможности для поиска отелей. Когда пользователь ищет отели, бэкенд формирует SQL-запросы с использованием полнотекстового поиска (по названию, описанию, удобствам), географического поиска (расстояние до центра города, достопримечательностей), фильтрации по параметрам (даты, цена, рейтинг, тип номера).

### **Сервис уведомлений**

Node.js. Сервис получает события только от сервиса отелей (например, о новом бронировании). Его задача — отправить email или SMS пользователю. Сервис не имеет API и работает полностью асинхронно, получая сообщения через простую очередь.

### **БД**

В PostgreSQL содержатся все пользовательские профили с их учетными данными, история бронирований с детализированным статусом каждого заказа, а также защищенная платежная информация в виде специальных защищенных кодов. СУБД обеспечивает целостность транзакций при обработке бронирований и поддерживает сложные запросы для генерации отчетов.

В MongoDB содержатся полные описания отелей с фотографиями, динамически изменяемыми параметрами номеров и мультязычными переводами.

