

Imtihon loyihasi: Hotel booking system

1. Umumiy korinishi

Hotel booking system tizimini Go, Kafka va boshqa texnologiyalar yordamida microservice arxitekturasini ishlab chiqishni o'z ichiga oladi. Tizimda booking service orqali rezervatsiya va bronlash boshqaruvi amalga oshiriladi, API Gateway qo'llab-quvvatlanadi, real vaqtda xabarlar uchun WebSockets va ixtiyoriy email xabarlarini yuborish imkoniyati mavjud bo'ladi.

2. Arxitektura komponentlari

1. **Hotel Service** - Mehmonxona ma'lumotlarini boshqaradi.
2. **Booking Service** - Rezervatsiya va bronlash boshqaruvini amalga oshiradi.
3. **User Service** - Foydalanuvchi ro'yxatdan o'tishi va autentifikatsiyasini boshqaradi.
4. **Notification Service** - Email va WebSocket orqali xabar yuboradi.
5. **API Gateway** - So'rovlarni tegishli microservice larga yo'naltiradi.
6. **WebSocket Service** - Klientlarga real vaqt xabarlarini taqdim etadi.

3. Microservice API Endpoints

3.1 Hotel Service

1. Mehmonxonalar Ro'yxati

- **Endpoint:** GET /api/hotels
- **Tavsif:** Mehmonxonalar ro'yxatini olish.
- **Javob:**

```
[
  {
    "hotelID": "string",
    "name": "string",
    "location": "string",
    "rating": "number",
    "address": "string"
  }
]
```

2. Mehmonxona Tafsilotlari

- **Endpoint:** GET /api/hotels/{hotelID}
- **Tavsif:** Ma'lum bir mehmonxona tafsilotlarini olish.
- **Javob:**

```
{
  "hotelID": "string",
  "name": "string",
  "location": "string",
  "rating": "number",
  "address": "string",
  "rooms": [
    {
      "roomType": "string",
      "pricePerNight": "number",
      "availability": "boolean"
    }
  ]
}
```

3. Xona Mavjudligini Tekshirish

- **Endpoint:** GET /api/hotels/{hotelID}/rooms/availability
- **Tavsif:** Ma'lum bir mehmonxonadagi xona mavjudligini tekshirish.
- **Javob:**

```
[
  {
    "roomType": "string",
    "availableRooms": "number"
  }
]
```

3.2 Booking Service

1. Bronlashni Yaratish va Tasdiqlash

- **Endpoint:** POST /api/bookings
- **Tavsif:** Bronlashni yaratish va tasdiqlash
- **So'rov:**

```
{
  "userID": "string",
  "hotelID": "string",
  "roomType": "string",
  "checkInDate": "date",
  "checkOutDate": "date",
  "totalAmount": "number"
}
```

- **Javob:**

```
{
  "bookingID": "string",
  "userID": "string",
  "hotelID": "string",
  "roomType": "string",
  "checkInDate": "date",
  "checkOutDate": "date",
  "totalAmount": "number",
  "status": "string" // Masalan, Tasdiqlangan
}
```

2. Bronlash Tafsilotlarini Olish

- **Endpoint:** GET /api/bookings/{bookingID}
- **Tavsif:** Ma'lum bir bronlash tafsilotlarini olish.
- **Javob:**

```
{
  "bookingID": "string",
  "userID": "string",
  "hotelID": "string",
  "roomType": "string",
  "checkInDate": "date",
  "checkOutDate": "date",
  "totalAmount": "number",
  "status": "string" // Masalan, Tasdiqlangan, Bekor qilingan
}
```

3. Bronlashni Yangilash

- **Endpoint:** PUT /api/bookings/{bookingID}}
- **Tavsif:** Mavjud bronlash tafsilotlarini yangilash.
- **So'rov:**

```
{
  "checkInDate": "date",
  "checkOutDate": "date",
  "totalAmount": "number",
  "status": "string" // Masalan, Tasdiqlangan, Bekor qilingan
}
```

- **Javob:**

```
{
  "bookingID": "string",
  "userID": "string",
  "hotelID": "string",
  "roomType": "string",
  "checkInDate": "date",
  "checkOutDate": "date",
  "totalAmount": "number",
  "status": "string" // Masalan, Tasdiqlangan, Bekor qilingan
}
```

4. Bronlashni Bekor Qilish

- **Endpoint:** DELETE /api/bookings/{bookingID}
- **Tavsif:** Ma'lum bir bronlashni bekor qilish.
- **Javob:**

```
{
  "message": "Bronlash muvaffaqiyatli bekor qilindi.",
  "bookingID": "string"
}
```

5.Foydalanuvchi Bronlashlarini Ro'yxatga Olish

- **Endpoint:** GET /api/users/{userID}/bookings
- **Tavsif:** Ma'lum bir foydalanuvchi tomonidan amalga oshirilgan bronlashlarni ro'yxatga olish.
- **Javob:**

```
[
  {
    "bookingID": "string",
    "hotelID": "string",
    "roomType": "string",
    "checkInDate": "date",
    "checkOutDate": "date",
    "totalAmount": "number",
    "status": "string" // Masalan, Tasdiqlangan, Bekor qilingan
  }
]
```

5.Foydalanuvchi Bronlashlarini Ro'yxatga Olish

- **Endpoint:** GET /api/users/{userID}/bookings
- **Tavsif:** Ma'lum bir foydalanuvchi tomonidan amalga oshirilgan bronlashlarni ro'yxatga olish.
- **Javob:**

```
[
  {
    "bookingID": "string",
    "hotelID": "string",
    "roomType": "string",
    "checkInDate": "date",
    "checkOutDate": "date",
    "totalAmount": "number",
    "status": "string" // Masalan, Tasdiqlangan, Bekor qilingan
  }
]
```

3.3 User Service

1.Foydalanuvchi Ro'yxatdan O'tishi

- **Endpoint:** POST /api/users
- **Tavsif:** Yangi foydalanuvchini ro'yxatdan o'tkazish.
- **So'rov:**

```
{
  "username": "string",
  "password": "string",
  "email": "string"
}
```

- **Javob:**

```
{
  "userID": "string",
  "username": "string",
  "email": "string"
}
```

2. Foydalanuvchi Kirishi

- **Endpoint:** POST /api/users/login
- **Tavsif:** Foydalanuvchi kirishi va token olish.
- **So'rov:**

```
{
  "username": "string",
  "password": "string"
}
```

- **Javob:**

```
{
  "token": "string",
  "expiresIn": "number" // Tokenning amal qilish muddati (sekundlarda)
}
```

4. Ixtiyoriy Vazifa

Xona Mavjudligi Xabarlari:

- Foydalanuvchilar avval mavjud bo'lmagan xona turlari uchun xabar olishni tanlashlari mumkin. Bu foydalanuvchi qiziqishini saqlash, xona mavjudligini monitoring qilish va `email` hamda `WebSocket` orqali xabar yuborishni o'z ichiga oladi.

5. Tizimning Ishlash O'qimi

1. Foydalanuvchi Ro'yxatdan O'tishi:

- Foydalanuvchi `POST /api/users` endpointi orqali ro'yxatdan o'tadi.
- So'rov muvaffaqiyatli bo'lsa, foydalanuvchi ma'lumotlari saqlanadi va foydalanuvchi `ID` qaytariladi.

2. Foydalanuvchi Kirishi va Token Olish:

- Foydalanuvchi `POST /api/users/login` endpointi orqali tizimga kiradi va token oladi.
- Token foydalanuvchi autentifikatsiyasi uchun ishlatiladi.

3. Mehmonxonalar Ro'yxatini Ko'rish:

- Foydalanuvchi `GET /api/hotels` endpointi orqali mehmonxonalar ro'yxatini ko'radi.

4. Mehmonxona Tanlash:

- Foydalanuvchi mehmonxonani tanlaydi va `GET /api/hotels/{hotelID}/rooms/availability` endpointi orqali xona mavjudligini tekshiradi.

5. Bronlashni Yaratish:

- Foydalanuvchi `POST /api/bookings` endpointi orqali bronlashni yaratadi.
- Bronlash muvaffaqiyatli yaratilsa, bronlash tafsilotlari qaytariladi.

6. Xona Mavjudligi Xabarlari:

- Agar foydalanuvchi xonani buyurtma qila olmasa, xabar olishni tanlagan bo'lsa, tizim xonalar mavjudligi haqida `WebSocket` orqali xabar yuboradi.

7. Email Xabarlari:

- Bronlash tasdiqlangandan so'ng, tizim foydalanuvchiga `email` orqali tasdiqlash xabari yuboradi.

8. API Gateway:

- API Gateway so'rovlarni tegishli microserviselarga yo'naltiradi.

9. WebSocket Servicedan Xabarlar:

- `WebSocket` xizmati real vaqt xabarlari yuboradi, masalan, xona mavjudligi va bronlash holati haqida.

6. Texnologiyalar va Talablar

- Microservicelar Aloqasi: Microservicelar orasida aloqani o'rnatish uchun `gRPC` foydalaniladi.
- Message Brokeri: `Kafka` asinxron aloqa va xabarlar uchun ishlatiladi.

- o Real Vaqtda Xabarlar: Xona mavjudligi va bronlash holati haqidagi real vaqt xabarlari uchun `WebSockets` ishlatiladi.
- o API Gateway: So'rovlarni tegishli `microservicelarga` yo'naltiradi.
- o Swagger: API hujjatlari va testlari uchun foydalaniladi.
- o Email Xabarlar: Bronlash tasdiqlari va yangilanishlarni email orqali yuborish.
- o HTTPS: Xavfsiz aloqa uchun `HTTPS` ishlatiladi.
- o Rate Limiting: Suiste'molni oldini olish uchun `rate limiting` amalga oshiriladi.
- o Graceful Shutdown: `Servicelarni` to'g'ri to'xtatilishini ta'minlash.
- o Konfiguratsiya Boshqaruvi: Konfiguratsiyalarni muhit o'zgaruvchilari yoki konfiguratsiya boshqaruv tizimi yordamida boshqarish.