**📘 Sofra - Restoran Yönetim Sistemi**

**🧩 Proje Özeti**

**Sofra**, restoranların dijital dönüşümünü kolaylaştırmak amacıyla geliştirilen bir **tam kapsamlı yönetim sistemi**dir. Uygulama hem **mobil (React Native + Expo Go)** hem de **web tabanlı (planlı)** olarak çalışmakta ve restoran müşterilerinden çalışanlara, yöneticilerden restoran sahiplerine kadar tüm paydaşları kapsayan bir deneyim sunmaktadır.

**🎯 Amaç ve Faydaları**

* **Restoran sahipleri**: Menü yönetimi, sipariş takibi, personel kontrolü ve satış raporları.
* **Çalışanlar**: Masa yönetimi, çağrı talepleri ve stok takibi.
* **Müşteriler**: Sipariş verme, çağrı gönderme ve geçmiş siparişleri görme.
* **Yöneticiler**: Kullanıcı yönetimi, sistem denetimi ve kullanıcı etkinliklerinin analizi.

**👨‍💻 Kullanıcı Rolleri ve Senaryoları**

| **Rol** | **Temel Yetkiler** |
| --- | --- |
| **Admin** | Tüm kullanıcıları ve restoranları yönetir. |
| **Restoran Sahibi (OWNER)** | Restoran oluşturur, menü ve çalışan yönetimi yapar. |
| **Müşteri (CUSTOMER)** | Sipariş verir, çağrı gönderir. |
| **Çalışan (WAITER)** | Masaları yönetir, çağrılara yanıt verir. |

**📌 Örnek Kullanım Senaryosu**

1. **Bir restoran sahibi** kayıt olur → restoranını oluşturur.
2. Restoranına **masa (table)** ve **menü öğeleri (menu item)** ekler.
3. Müşteriler mobil uygulama üzerinden giriş yapar → restorana ait menüyü görüntüleyip sipariş verir.
4. Masadan **garson çağrısı (call request)** gönderir.
5. Çalışanlar gelen çağrıları ve siparişleri yönetir.
6. Sahipler veya yöneticiler **stokları (inventory)** günceller, **raporları (sales report)** kontrol eder.

**🧱 Backend Katmanları (Spring Boot)**

**1. Model Katmanı (model/)**

* User: Kullanıcı bilgilerini içerir. (admin, owner, customer, waiter)
* Restaurant: Restoran bilgileri (adı, konum, açıklama, koordinatlar vs.)
* Table: Masa bilgileri (numara, restoran ilişkisi vs.)
* MenuItem: Menüye ait ürünler (isim, açıklama, fiyat, kategori)
* Order: Sipariş bilgileri
* CallRequest: Garson çağrı verisi
* Inventory: Stok yönetimi
* Sale: Satış kaydı
* Role: Kullanıcı rollerinin ENUM tanımı

**2. Repository Katmanı (repository/)**

MongoDB ile bağlantılı interface yapılarıdır. CRUD işlemlerini yönetir.  
Örnekler:

* UserRepository
* RestaurantRepository
* OrderRepository

**3. Service Katmanı (service/)**

Tüm iş mantığı burada yazılır. Controller’lara hizmet eder.  
Örnek:

* RestaurantService: Restoran oluşturma, getirme, silme işlemleri.
* OrderService: Sipariş yönetimi.
* CallRequestService: Garson çağrıları için mantık.

**4. Controller Katmanı (controller/)**

API endpointlerinin yer aldığı sınıflardır.  
Örnekler:

* AuthController: Kayıt & Giriş
* RestaurantController: /api/restaurants
* CallRequestController: /api/calls

**5. Config (config/)**

* SecurityConfig: JWT ile güvenlik ayarları
* JwtAuthenticationFilter, JwtService: Token doğrulama ve üretim işlemleri

**6. Exception Handling**

* GlobalExceptionHandler: 400, 404, 500 gibi durumları yönetir
* ResourceNotFoundException: İstenen kaynak bulunamadığında fırlatılır

**📱 Frontend (Mobil - React Native + Expo)**

**🔧 Kurulum Adımları**

npx create-expo-app sofra-mobile --template

cd sofra-mobile

npm install

npx expo start

**💡 Planlanan Sayfalar**

* Giriş / Kayıt ekranı
* Kullanıcı dashboard’ları (rol bazlı yönlendirme)
* Restoran listesi ve detayları
* Sipariş oluşturma ve geçmiş siparişler
* Garson çağırma butonu
* Menü yönetimi (sahipler için)
* Envanter (stok) ve satış görüntüleme

**🧾 Veritabanı (MongoDB) Yapısı**

**Koleksiyonlar:**

* users
* restaurants
* tables
* menu\_items
* orders
* call\_requests
* inventories
* sales

**Örnek User Belgesi**

{

"email": "owner@mail.com",

"name": "Ali Veli",

"role": "OWNER",

"password": "hashed\_pw",

"enabled": true,

...

}

**📈 Raporlama Özelliği (Sales Reporting)**

* Günlük / aylık satış görüntüleme
* Toplam sipariş sayısı
* Kazanç analizleri (Dashboard'a taşınacak)

**🛡️ JWT ile Güvenlik**

* Her kullanıcı giriş yaptıktan sonra token alır.
* Bu token ile yetkili olduğu endpoint’lere erişir.
* @PreAuthorize("hasAuthority('ADMIN')") gibi anotasyonlar ile erişim sınırlandırılır.

**✅ Sonuç ve Gelecek Geliştirmeler**

Bu proje, restoran yönetimini dijitalleştirmek isteyen işletmelere mobil ve web tabanlı modern bir çözüm sunmaktadır. Gelecekte aşağıdaki özellikler entegre edilebilir:

* QR kodla menü görüntüleme ve sipariş verme
* Google Maps API ile canlı konum takibi
* Stripe veya Iyzico ile ödeme sistemi
* Bildirim sistemi (sipariş onaylandı vs.)
* Çoklu dil desteği (i18n)