Elbette! Aşağıda verdiğiniz analiz içeriğini profesyonel, düzgün ve sistematik bir **Türkçe rapor** formatında yeniden yapılandırdım. Bu haliyle hem teknik dökümantasyon olarak hem de proje içi planlama ve geliştirme sürecinde kullanılabilir:

**📊 MasaMenu – Kapsamlı Model ve Servis Yapısı Analizi Raporu**

**🏗️ 1. Sistem Mimarisi Genel Bakış**

MasaMenu uygulaması, modüler ve rol bazlı bir yapıya sahiptir. Üç ana kullanıcı tipi desteklenmektedir:

* 👤 **Admin (Yönetici)** – Tüm sistemin denetimi
* 🏢 **İşletme (Business)** – Kendi işletmesini ve menülerini yöneten kullanıcı
* 👥 **Müşteri (Customer)** – Sipariş veren ve uygulamayı son kullanıcı olarak kullanan kişi

Her kullanıcı tipi için ayrı klasörler, modeller ve servisler tanımlanmış olup, ortak yapıların kullanımı merkezî core/ dizininde toplanmıştır.

**📋 2. Veri Modeli Analizi**

**🎯 Ana Modeller ve Amaçları**

**🧩 User Modeli**

* **Konum:** lib/data/models/user.dart
* **Amaç:** Tüm kullanıcı tipleri için ortak bir yapı sunar.
* **İlişkiler:**
  + customerData, businessData, adminData alanları ile ilgili modüllere bağlanır.
* **İşlevler:**
  + JSON dönüşümleri, factory constructor'lar, kullanıcı tipi kontrolü (isCustomer gibi).

**🧩 Business Modeli**

* **Konum:** lib/data/models/business.dart
* **Amaç:** İşletmeye ait tüm bilgiler, ayarlar ve ilişkili veriler burada tutulur.
* **İlişkiler:**
  + User (ownerId)
  + Category, Product, Order

**🧩 CustomerUser ve CustomerPreferences Modelleri**

* **Konum:** lib/customer/models/
* **Amaç:** Müşteriye özel tercihler, geçmiş veriler ve favoriler burada tutulur.

**🧩 BusinessUser Modeli**

* **Konum:** lib/business/models/business\_user.dart
* **Amaç:** İşletme personel bilgileri ve yetkilendirmeleri.

**🧩 AdminUser Modeli**

* **Konum:** lib/admin/models/admin\_user.dart
* **Amaç:** Admin kimlik ve yetki bilgilerinin yönetimi.

**🔄 3. Model İlişkileri (Mermaid diyagramı temelli)**

* User → CustomerData, BusinessData, AdminData
* Business → Category, Product, Order, BusinessUser
* Customer → Order, Cart, Preferences
* Product → Category, Discount

**🛠️ 4. Servis Yapısı Analizi**

**🌐 Core Servisler (lib/core/services/)**

| **Servis** | **Görev** |
| --- | --- |
| **FirestoreService** | Firebase üzerinden tüm veri işlemleri (CRUD, senkronizasyon) |
| **AuthService** | Giriş, çıkış, kullanıcı kimliği yönetimi |
| **DataService** | Yerel veri (cache, shared preferences) yönetimi |
| **CartService** | Sepet kontrolü ve ürün yönetimi |
| **OrderService** | Sipariş işlemleri, durum güncellemeleri |
| **QRService** | QR kod üretimi ve yönetimi |
| **StorageService** | Dosya yükleme, silme, Firebase Storage entegrasyonu |

**🧩 Modüle Özgü Servisler**

| **Servis** | **Açıklama** |
| --- | --- |
| **AdminService** | Admin login, işletme onayları, sistem analitiği |
| **BusinessService** | Personel yetkilendirme, işletme ayarları, operasyonlar |
| **CustomerService** | Müşteri profili, tercihleri, geçmiş işlemler |

**✅ 5. Mimarinin Güçlü Yanları**

* 📁 **Modülerlik:** Admin, Business ve Customer modülleri açıkça ayrılmıştır.
* 🔐 **Güvenlik:** SHA-256 hash, session kontrolü, yetki yönetimi.
* ⚙️ **Sorumluluk Ayrımı:** Her servis tek bir işlevde uzmanlaşmıştır.
* 🔄 **Veri Akışı:** Net, okunabilir ve merkezi kontrol edilen veri yapıları.