OTO SERVİS PROGRAMI RAPORU

Denizhan Çil   
231307104  
denizcilcd@gmail.com

Hamza Akdemir  
 231307025  
 hamzaakademie@gmail.comMeliha Damla Coşkun  
231307113  
damlala33@gmail.com Enes Şenyurt  
231307032  
enes.senyurt11@gmail.com

1. Problem

Otomotiv servis işletmelerinde müşteri randevu yönetimi, servis personeli ataması ve parça stok takibi çoğunlukla manuel yapılmakta; bu da hatalara, çakışmalara ve verimsizliğe neden olmaktadır. Bu proje, aşağıdaki sorunlara çözüm getirmeyi amaçlamakta:

-Randevu Yönetimi: Müşterilerin araçları için uygun tarih-saatte randevu oluşturabilmesi, servisin müsait personeline otomatik yönlendirilmesi.

-Kullanıcı Rolleri: Farklı yetkilerle çalışan “admin” ve “user” kullanıcı tiplerinin ayrı ayrı işlevlere sahip olması.

-Parça Stok Takibi: Servis parçalarının stoklarının yönetilmesi, kullanıcıların stokta mevcut parçalara sipariş verebilmesi ve stoktan düşülmesi.

-Veri Tutarlılığı: Tüm işlemlerde veri tabanı bütünlüğünün (foreign key, 5NF) korunması.

2. Yapılan Araştırmalar

1. Flutter & Next.js Kimlik doğrulama

\* JWT tabanlı kimlik doğrulama ve flutter\_secure\_storage kullanımını inceledik.

\* Next.js API Routes’ta dinamik [id].ts dosya yapısı ve token doğrulama metotlarını araştırdık.

2. Veri Tabanı Tasarımı

\* 5NF’ye uygun, ilişkisel modelde araç, servis, personel, parça ve sipariş tablolarını oluşturduk.

\* MySQL foreign key kısıtlamaları (ON DELETE CASCADE) hakkında dökümantasyonlardan yararlandık.

3. Durum Yönetimi

\* Flutter’da FutureBuilder ve setState ile liste yenileme çözümlerini inceledik.

4. Akış Şeması

\* Lucidchart ve mermaid.js kullanarak akış diyagramını tasarladık.

3. Akış Şeması

A[Uygulama Başlatılır] --> B{Kullanıcı Giriş mi?}

B -- Hayır --> C[Login/Register Ekranı]

C --> B

B -- Evet --> D{Role=Admin mi?}

D -- Evet --> E[Admin Dashboard]

D -- Hayır --> F[User Ana Sayfa]

subgraph Admin

E --> E1[Servis Yönetimi]

E --> E2[Personel Yönetimi]

E --> E3[Parça Yönetimi]

E --> E4[Sipariş Onay/Red]

end

subgraph User

F --> F1[Hizmet Listesi]

F --> F2[Randevu Oluşturma]

F --> F3[Randevularım]

F --> F4[Araçlarım]

F --> F5[Parça Listesi ve Sipariş]

F --> F6[Siparişlerim]

end

4. Yazılım Mimarisi

├── backend/

│ ├── lib/db.ts

│ ├── api/

│ │ ├── register.ts

│ │ ├── login.ts

│ │ ├── vehicles/

│ │ ├── services/

│ │ ├── personnel/

│ │ ├── part-orders/

│ │ │ ├── index.ts

│ │ │ └── [id].ts

│ │ └── create-record.ts

│ └── models/

├── frontend/

│ ├── lib/

│ │ ├── main.dart

│ │ ├── constants/theme.dart

│ │ ├── services/api.dart

│ │ ├── models/

│ │ └── screens/

│ │ ├── login\_page.dart

│ │ ├── home\_page.dart

│ │ ├── home\_admin\_page.dart

│ │ ├── user/

│ │ │ ├── services\_page.dart

│ │ │ ├── create\_record\_page.dart

│ │ │ ├── my\_records\_page.dart

│ │ │ ├── vehicles\_page.dart

│ │ │ ├── vehicle\_parts\_page.dart

│ │ │ └── part\_orders\_page.dart

│ │ └── admin/

│ │ ├── services\_admin\_page.dart

│ │ ├── personnel\_admin\_page.dart

│ │ ├── service\_records\_admin\_page.dart

│ │ ├── vehicle\_parts\_admin\_page.dart

│ │ └── part\_orders\_admin\_page.dart

│ └── pubspec.yaml

└── README.md

--Geliştirme Aşamaları--

1. Auth\*: Register/login, JWT saklama.

2. Veri Tabanı: Tabloların oluşturulması ve ERD onayı.

3. CRUD API: Next.js ile her tablo için GET/POST/PUT/DELETE endpoint’leri.

4. Flutter UI: Ekranların temel form ve liste yapıları.

5. State Yönetimi: FutureBuilder, setState ile dinamik veri güncelleme.

6. Tema & UX: Siyah-mavi tema, modern görünümler

5. ER Diyagramı

USERS ||--o{ VEHICLES : owns

USERS ||--o{ SERVICE\_RECORDS : creates

VEHICLES ||--o{ SERVICE\_RECORDS : has

SERVICES ||--o{ SERVICE\_RECORDS : scheduled

SERVICES }o--|| PERSONNEL : handled\_by

USERS ||--o{ PART\_ORDERS : requests

VEHICLE\_PARTS ||--o{ PART\_ORDERS : ordered

PART\_ORDERS }o--|| VEHICLE\_PARTS : reduces\_stock

USERS {

INT id PK

VARCHAR name

VARCHAR email

VARCHAR password\_hash

ENUM role

}

VEHICLES {

INT id PK

INT owner\_id FK

VARCHAR make

VARCHAR model

VARCHAR plate\_number

INT year

}

SERVICES {

INT id PK

VARCHAR name

TEXT description

DECIMAL standard\_price

INT personnel\_id FK

}

PERSONNEL {

INT id PK

VARCHAR name

VARCHAR position

VARCHAR contact

}

SERVICE\_RECORDS {

INT id PK

INT vehicle\_id FK

INT service\_id FK

INT personnel\_id FK

DATETIME service\_date

INT created\_by FK

TEXT notes

}

VEHICLE\_PARTS {

INT id PK

VARCHAR part\_name

VARCHAR part\_code

DECIMAL cost

INT stock

}

PART\_ORDERS {

INT id PK

INT user\_id FK

INT part\_id FK

INT quantity

DATETIME order\_date

ENUM status

}

6. Genel Yapı

-Modüler Mimari: Her işlev (araçlar, servisler, parçalar, kayıtlar, siparişler) ayrı API klasöründe ve Flutter ekranda modül halinde.

-Temiz Ayrım: Frontend/Backend net ayırımı. API katmanı yalnızca JSON, Flutter yalnızca HTTP client.

-Güvenlik: JWT ile endpoint bazlı rol kontrolü.

-Veri Bütünlüğü: 5NF uyumlu, tüm foreign key ilişkileri açıkça tanımlı.

-UI/UX: Kolay navigasyon, modern siyah-mavi tema, responsive listeler.

7. Referanslar

[1] Flutter: “Building layouts in Flutter,” Flutter Docs.

[2] Next.js API Routes: Next.js Resmi Dokümantasyonları.

[3] MySQL Foreign Keys: MySQL 8.0 Reference Manual.

[4] JWT: jwt.io Documentation.

[5] Mermaid.js: mermaid-js.github.io/mermaid.