ANKARA UNIVERSITESI

Muhendislik Fakultesi

Bilgisayar Muhendisligi Bolumu

KUTUPHANE YONETIM SISTEMI

BLM4537 - IOS İle Mobil Uygulama Geliştirme I [B]

Ogrenci: Ahmet Akgun
Ogrenci No: 22290602
Danisman: Enver Bağcı

Ocak 2026

Contents

1 Ozet

Bu proje, Flutter framework kullanilarak gelistirilen cross-platform bir kutuphane yonetim sistemi mobil uygulamasidir. Uygulama, .NET Core Web API backend ile iletisim kurarak kitap yonetimi, odunc alma/iade islemleri ve istatistik goruntuleme gibi islevleri saglamaktadir.

Kullanilan Teknolojiler:

- **Framework:** Flutter 3.x, Dart
- **State Management:** Riverpod
- **HTTP Client:** Dio
- **Navigation:** go_router
- **Grafikler:** fl_chart
- **Code Generation:** Freezed, json_serializable
- **Secure Storage:** flutter_secure_storage

2 Giris ve Amac

2.1 Projenin Amaci

Bu mobil uygulamanin temel amaci, kutuphane yonetim sistemine mobil cihazlardan erisim saglamaktir. Uygulama su ozellikleri sunmaktadir:

1. Kitap kataloguna goz atma ve arama
2. Kullanici kayit ve giris islemleri
3. Kitap odunc talebi olusturma
4. Odunc gecmisini goruntuleme
5. Admin paneli ile yonetim islemleri
6. Istatistik ve raporlama (grafikler)

2.2 Kapsam

Uygulama iki kullanici rolunu desteklemektedir:

- **Uye (Member):** Kitap goruntuleme, odunc talep etme, profil yonetimi
- **Admin:** Tum islemler + kitap/odunc yonetimi, istatistik goruntuleme

3 Uygulama Mimarisi

3.1 Katmanli Yapi

Uygulama, Clean Architecture prensiplerine uygun katmanli bir yapida tasarlanmistir:

Screens (UI Layer) Auth, User, Admin ekranlari
Widgets (Reusable Components) BookCard, LoadingWidget, vs.
Providers (State Management) AuthProvider, BookProvider, LoanProvider, StatisticsProvider
Services (Business Logic) ApiClient, AuthService, BookService, LoanService
Models (Data Layer) BookModel, UserModel, LoanModel, StatisticsModel

Figure 1: Flutter Uygulama Mimarisi

3.2 Proje Dosya Yapisi

```
lib/
|-- main.dart                    # Uygulama giris noktasi
|-- core/
|   |-- constants/               # API sabitleri
|   |-- theme/                   # Tema ve renkler
|   |-- utils/                   # Router, yardimci fonksiyonlar
|-- models/
|   |-- book_model.dart          # Kitap modeli
|   |-- user_model.dart          # Kullanici modeli
|   |-- loan_model.dart          # Odunc modeli
|   |-- statistics_model.dart    # Istatistik modeli
|   |-- *.freezed.dart           # Freezed generated
|   |-- *.g.dart                 # JSON serializable
|-- services/
|   |-- api_client.dart          # Dio HTTP client
|   |-- auth_service.dart        # Kimlik dogrulama
|   |-- book_service.dart        # Kitap islemleri
|   |-- loan_service.dart        # Odunc islemleri
|   |-- statistics_service.dart  # Istatistik API
|-- providers/
|   |-- auth_provider.dart       # Auth state
|   |-- book_provider.dart       # Book state
|   |-- loan_provider.dart       # Loan state
|   |-- statistics_provider.dart # Statistics state
|-- screens/
|   |-- auth/                    # Giris, Kayit, Splash
|   |-- user/                    # Ana sayfa, Profil, Kitap detay
|   |-- admin/                   # Dashboard, Yonetim ekranlari
|-- widgets/                     # Yeniden kullanilabilir widgetlar
```

4 State Management - Riverpod

4.1 Provider Yapisi

Uygulama, Riverpod kutuphanesi ile state yonetimi yapmaktadir:

```
// Ornek: Book Provider
final allBooksProvider = FutureProvider<List<BookModel>>((ref)
   async {
  final service = ref.watch(bookServiceProvider);
  return await service.getAllBooks();
});

// Ornek: Auth Provider
final authNotifierProvider =
  StateNotifierProvider<AuthNotifier, AuthState>((ref) {
    return AuthNotifier(ref);
});
```

4.2 Kullanilan Providerlar

Provider	Aciklama
authNotifierProvider	Kullanici oturum durumu
currentUserProvider	Mevcut kullanici bilgisi
allBooksProvider	Tum kitaplar listesi
allLoansProvider	Tum odunc kayitlari
pendingLoansProvider	Bekleyen talepler
libraryStatisticsProvider	Kutuphane istatistikleri
categoryStatisticsProvider	Kategori istatistikleri
topBooksProvider	Populer kitaplar

Table 1: Riverpod Provider Listesi

5 API Entegrasyonu

5.1 Dio HTTP Client

API istekleri Dio kutuphanesi ile yapilmaktadir:

```
class ApiClient {
  late final Dio _dio;
  final FlutterSecureStorage _secureStorage;

  ApiClient._internal() {
    _dio = Dio(BaseOptions(
      baseUrl: ApiConstants.baseUrl,
      connectTimeout: Duration(seconds: 30),
    ));

    // JWT Token interceptor
```

```
    _dio.interceptors.add(InterceptorsWrapper(
      onRequest: (options, handler) async {
        final token = await getToken();
        if (token != null) {
          options.headers['Authorization'] = 'Bearer $token';
        }
        return handler.next(options);
      },
    ));
  }
}
```

5.2 API Servisleri

Servis	Islevler
AuthService	login, register, getCurrentUser
BookService	getAllBooks, getBookById, search
LoanService	createLoan, returnBook, getMyLoans
CategoryService	getCategories, getActiveCategories
StatisticsService	getStatistics, getCategoryStats, getTopBooks

Table 2: API Servisleri

6 Uygulama Ekranlari

6.1 Auth Ekranlari

- **SplashScreen:** Uygulama acilis, token kontrolu
- **LoginScreen:** Kullanici girisi (email/sifre)
- **RegisterScreen:** Yeni kullanici kaydi

6.2 User Ekranlari

- **HomeScreen:** Kitap listesi, arama, kategori filtre
- **BookDetailScreen:** Kitap detayi, odunc alma butonu
- **ProfileScreen:** Profil bilgileri, odunc gecmisi

6.3 Admin Ekranlari

- **AdminDashboardScreen:** Ozet istatistikler, hizli erisim
- **BookManagementScreen:** Kitap CRUD islemleri
- **LoanManagementScreen:** Odunc onay/red islemleri
- **StatisticsScreen:** Grafikli istatistik raporu

7 Istatistik Grafikleri

7.1 fl_chart Entegrasyonu

Uygulama, fl_chart kutuphanesi ile grafik gosterimi yapmaktadir:

- **PieChart:** Kategori bazli kitap dagilimi
- **LineChart:** Aylik odunc trendi
- **Stat Cards:** Ozet istatistik kartlari

```
// Ornek: PieChart kullanimi
PieChart(
  PieChartData(
    sections: categories.map((cat) =>
      PieChartSectionData(
        value: cat.bookCount.toDouble(),
        title: cat.categoryName,
        color: colors[index],
      )
    ).toList(),
  ),
)
```

8 Model Generation - Freezed

8.1 Immutable Data Classes

Freezed kutuphanesi ile immutable ve type-safe modeller olusturulmaktadir:

```
@freezed
class BookModel with _$BookModel {
  const factory BookModel({
    required int id,
    required String title,
    required String author,
    required int categoryId,
    required String categoryName,
    String? isbn,
    int? publishYear,
    String? imageUrl,
    @Default(true) bool isAvailable,
  }) = _BookModel;

  factory BookModel.fromJson(Map<String, dynamic> json) =>
      _$BookModelFromJson(json);
}
```

8.2 Modeller

- BookModel - Kitap verisi
- UserModel - Kullanici verisi
- LoanModel - Odunc verisi
- CategoryModel - Kategori verisi
- LibraryStatistics - Genel istatistikler
- CategoryStatistics - Kategori istatistikleri
- TopBook - Populer kitaplar

9 Navigation - go_router

9.1 Route Yapisi

```
final routerProvider = Provider<GoRouter>((ref) {
  return GoRouter(
    initialLocation: '/splash',
    routes: [
      GoRoute(path: '/splash', builder: (_, __) => SplashScreen()
        ),
      GoRoute(path: '/auth/login', builder: (_, __) =>
        LoginScreen()),
      GoRoute(path: '/auth/register', builder: (_, __) =>
        RegisterScreen()),
      GoRoute(path: '/home', builder: (_, __) => HomeScreen()),
      GoRoute(path: '/book/:id', builder: (_, state) =>
        BookDetailScreen(bookId: state.pathParameters['id']!)),
      GoRoute(path: '/admin', builder: (_, __) =>
        AdminDashboardScreen()),
      GoRoute(path: '/admin/statistics', builder: (_, __) =>
        StatisticsScreen()),
    ],
  );
});
```

10 Sonuc ve Degerlendirme

10.1 Tamamlanan Ozellikler

- JWT tabanli kimlik dogrulama
- Kitap listeleme, arama, filtreleme
- Kitap odunc talebi olusturma ve takip

- Admin paneli ile tam yonetim
- Grafikli istatistik goruntuleme
- Responsive ve modern UI tasarimi
- Cross-platform destek (Android/iOS)

11 Kaynaklar

1. Flutter Documentation - https://flutter.dev/docs
2. Riverpod Documentation - https://riverpod.dev
3. Freezed Package - https://pub.dev/packages/freezed
4. Dio Package - https://pub.dev/packages/dio
5. fl_chart Package - https://pub.dev/packages/fl_chart
6. go_router Package - https://pub.dev/packages/go_router