2025

Erciyes Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği

Mobile Application Development
Dr. Öğr. Üyesi Fehim KÖYLÜ

IMDb Top 250 Film Listeleme ve Reklam gösterimi

VİZE PROJESİ – UYGULAMA ÖDEVİ

BETÜL MOLLAMUSA - 1030510334

IMDb Top 250 Film Listeleme Uygulaması: Teknik Rapor

1. GİRİŞ

Mobil uygulama geliştirme, günümüzün en hızlı büyüyen yazılım alanlarından biridir. Flutter, bu alandaki en güçlü araçlardan biridir ve tek bir kod tabanı ile hem Android hem de iOS platformlarında uygulama geliştirmeye olanak sağlar. Bu projede, Flutter kullanılarak IMDb'nin Top 250 film listesini görüntüleyen ve kullanıcıların favori filmleri seçebileceği bir mobil uygulama geliştirilmiştir. Uygulama, film listesini dinamik olarak yükler ve her 5. filmde reklam gösterimi sunarak, kullanıcı deneyimini zenginleştirir. Ayrıca, kullanıcıların favori filmleri görüntüleyebileceği ve favori listesini yönetebileceği ayrı bir ekran sunulmaktadır.

2. AMAÇ VE HEDEFLER

Projenin amacı, kullanıcılara IMDb Top 250 filmlerini göstermek, filmleri favorilerine eklemelerine olanak sağlamak ve her 5. filmde bir reklam kutusu göstererek gelir elde etmektir. Ayrıca, uygulama içerisinde kullanıcı etkileşimini geliştiren ve navigasyonu kolaylaştıran birkaç önemli özellik bulunmaktadır. Uygulama, kullanıcıların favori filmleri kaydedebilmesi, favorilerini güncelleyebilmesi ve favori filmleri başka bir ekranda görmesi gibi temel işlevlere sahiptir.

Projenin hedefleri şunlardır:

- IMDb Top 250 filmlerinin kullanıcıya liste halinde sunulması.
- Kullanıcıların film favorisi ekleyip çıkarabilmesi.
- Favori filmleri izleme ve yönetme ekranlarının sağlanması.
- Her 5. filmde reklam gösterilmesi.
- Temel mobil uygulama geliştirme tekniklerinin ve Flutter framework'ünün uygulanması.

3. KULLANILAN TEKNOLOJİLER VE ARAÇLAR

Bu projede kullanılan teknolojiler, uygulamanın işlevselliği ve performansı açısından büyük öneme sahiptir. Aşağıda kullanılan teknolojiler detaylı bir şekilde açıklanmıştır:

3.1. Flutter

Flutter, Google tarafından geliştirilen açık kaynaklı bir UI toolkit'idir. Mobil uygulama geliştirme süreçlerinde, tek bir kod tabanı ile hem Android hem de iOS platformlarında çalışabilen uygulamalar geliştirilmesine olanak sağlar. Flutter, zengin kullanıcı arayüzü

(UI) bileşenleri ve hızlı geliştirme imkânı sunar. Bu projede, UI tasarımı, veri yönetimi ve animasyonlar gibi birçok temel özellik Flutter ile gerçekleştirilmiştir.

3.2. Dart Programlama Dili

Flutter ile geliştirilmiş uygulamalar Dart dilinde yazılır. Dart, özellikle Flutter ile uyumlu çalışan ve mobil uygulamalarda yüksek performans sağlayan bir dil olarak dikkat çekmektedir. Dart'ın "just-in-time" derleyicisi (JIT) geliştirme sürecini hızlandırırken, "ahead-of-time" (AOT) derleyicisi ise uygulamanın performansını optimize eder. Bu projede, filmler ve favoriler arasında veri akışı, kullanıcı etkileşimi ve UI güncellemeleri Dart diliyle yönetilmiştir.

3.3. JSON Veri Formatı

Veriler, JSON (JavaScript Object Notation) formatında saklanmaktadır. JSON, verileri saklamak ve taşımak için yaygın olarak kullanılan, insan tarafından okunabilir bir veri formatıdır. Uygulama, film verilerini JSON formatında bir dosyadan alır ve bu veriler, Flutter'ın rootBundle.loadString fonksiyonu kullanılarak yüklenir. JSON, veri aktarımında hızlı ve hafif olması nedeniyle tercih edilmiştir.

3.4. Firebase Analytics (Potansiyel Gelecek Güncellemeleri için)

Gelecek güncellemelerinde, kullanıcı etkileşimlerini izlemek için Firebase Analytics entegre edilebilir. Bu, uygulamanın kullanıcı davranışları hakkında bilgi edinmek ve kullanıcı deneyimini iyileştirmek adına faydalı olacaktır.

3.5. Flutter Widgets ve UI Tasarımı

Flutter'ın hazır widget'ları, uygulamanın temel yapı taşlarını oluşturur. ListView.builder gibi widget'lar, dinamik listeleme işlevini yerine getirirken, Column, Row, Container gibi widget'lar ise temel yapılandırma ve düzenleme işlevini üstlenir. Bu sayede, kullanıcı arayüzü (UI) temiz ve etkileşimli bir şekilde tasarlanmıştır.

4. YAZILIMSAL MİMARİSİ VE TEKNİK DETAYLAR

Uygulamanın mimarisi, Flutter'ın önerdiği temel yapı taşlarına dayanır. Uygulama, dinamik bir veri yükleme, kullanıcı etkileşimi ve favori yönetimi gibi önemli işlevleri yerine getirir. Aşağıda, uygulamanın yazılımsal yapısı ve kullanılan teknik detaylar açıklanacaktır.

4.1. Veri Yükleme ve İşlenmesi

Film verileri, uygulama başlatıldığında JSON formatında bir dosyadan yüklenir. Bu işlem, rootBundle.loadString fonksiyonu ile yapılır ve asenkron bir işlem olarak gerçekleşir. Yüklenen veriler JSON formatında bir listeye dönüştürülür ve ardından her bir film, Movie.fromJson() fonksiyonu ile Movie sınıfına dönüştürülür.

- **Veri Yükleme ve JSON İşleme Örneği:**

```
Future<void> loadMovies() async {
  final data = await rootBundle.loadString('assets/movies.json');
  final List decoded = json.decode(data); // JSON verisini decode et
  setState(() {
    allMovies = decoded.map((json) => Movie.fromJson(json)).toList();
//Filmleri modelle
  });
}
```

Bu işlem sırasında fromJson() fonksiyonu, JSON objesindeki verileri uygun model formatına dönüştürmek için kullanılır. Bu sayede her film bir Movie objesi haline gelir.

4.2. Favori Filmler ve Yönetimi

Favori filmler, Set<String> veri yapısında saklanır. Set kullanımı, her filmin yalnızca bir kez kaydedilmesini sağlar. Kullanıcı bir filme favori eklemek ya da favoriden çıkarmak için toggleFavorite() fonksiyonunu kullanır. Bu fonksiyon, kullanıcı etkileşimini anlık olarak yönetir ve UI'yi günceller.

- **Favori Yönetimi Kodu:**

```
void toggleFavorite(String title) {
  setState(() {
    if (favorites.contains(title)) {
      favorites.remove(title); // Favoriden çıkar
    } else {
      favorites.add(title); // Favoriye ekle
    }
  });
}
```

Bu fonksiyon, film başlıklarını favorites kümesine ekler ya da çıkarır ve kullanıcı etkileşiminden sonra arayüzü günceller.

4.3. Reklam Yönetimi

Her 5. öğede bir reklam kutusu (AdBanner) göstermek için, ListView.builder widget'ı kullanılır. Bu sayede film listesi dinamik bir şekilde yüklenirken, belirli aralıklarla reklam gösterimi yapılır. Reklam kutusu yalnızca her 5. öğede görünür, bu sayede kullanıcı deneyimi bozulmaz.

- **Reklam Gösterimi Kodu:**

```
ListView.builder(
  itemCount: allMovies.length + (allMovies.length ~/ 5),
  itemBuilder: (context, index) {
    if ((index + 1) %6 == 0) return const AdBanner(); // 5. öğe sonrası
reklam
    final realIndex = index - (index ~/ 6);  // Gerçek film index'ini
hesapla
    final movie = allMovies[realIndex];
    return MovieCard(
      movie: movie,
      onFavoriteToggle: () => toggleFavorite(movie.title),
    );
  },
)
```

Bu yöntem, kullanıcı deneyimini olumsuz etkilemeden reklamların yerleştirilmesini sağlar.

4.4. Navigasyon ve Ekranlar Arası Geçiş

Flutter, ekranlar arasında geçiş yapmak için Navigator widget'ını kullanır. Favori filmleri görmek için kullanıcı FavoritesScreen ekranına geçer. Bu geçiş, Navigator.push fonksiyonu ile gerçekleştirilir.

- **Ekranlar Arası Geçiş Kodu:**

```
IconButton(
  icon: const Icon(Icons.star),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => FavoritesScreen(
          favorites: favorites,
          allMovies: allMovies,
          onFavoriteToggle: toggleFavorite,
        ),
      ),
    );
  },
)
```

Bu kod, kullanıcı favori filmlerini görmeye yönlendirilirken, tüm favori filmleri içeren FavoritesScreen ekranına geçiş yapar.

4.5. Film Listeleme Yapısı

Uygulamada en temel işlevlerden biri, IMDb verilerine göre sıralanmış film listesinin kullanıcıya görsel olarak sunulmasıdır. Bu listeleme, Flutter'ın sunduğu performans odaklı ListView.builder yapısı kullanılarak gerçekleştirilmiştir. Bu yapı sayesinde, çok sayıda öğe içeren listeler bellekte sadece görünür öğelerle çalışır, böylece uygulama hafif kalır ve daha akıcı bir kullanıcı deneyimi sunar.

- **Listeleme Algoritması**

Film listesi oluşturulurken, her beş film kartı gösteriminden sonra bir adet reklam kutusu (AdBanner) yerleştirilir. Bu reklam yerleşimi için özel bir indeks hesaplama algoritması kullanılır. Böylece, liste içinde görsel olarak tutarlı bir yapı korunmuş olur.

```
ListView.builder(
  itemCount: allMovies.length + (allMovies.length ~/ 5), // Reklamlar
dahil toplam öğe sayısı
  itemBuilder: (context, index) {
    if ((index + 1) %6 == 0) return const AdBanner(); // Her 6. öğe reklam
    final realIndex = index - (index ~/ 6); // Film index'ini hesapla
    final movie = allMovies[realIndex];
    return MovieCard(
      movie: movie,
      onFavoriteToggle: () => toggleFavorite(movie.title),
    );
  },
)
```

Bu yapı, yalnızca ihtiyaç duyulan veriyi belleğe alarak verimlilik sağlar, reklamları kullanıcıyı rahatsız etmeyecek şekilde entegre eder ve realIndex hesaplamalarıyla kod okunabilirliğini artırır.

- **MovieCard Widget'ı**

Her bir film, özel olarak oluşturulan MovieCard widget'ı ile ekranda gösterilir. Bu widget içerisinde film adı, puanı, görseli ve favori butonu gibi etkileşimli öğeler yer alır. Ayrıca bu yapı modüler bir biçimde tasarlandığı için, her kartın özelleştirilmesi veya geliştirilmesi ilerleyen süreçlerde kolaylıkla yapılabilir.

```
return MovieCard(
  movie: movie,
  isFavorite: favorites.contains(movie.title),
  onFavoriteToggle: () => toggleFavorite(movie.title),
);
```

Bu yapı, kullanıcı dostu bir arayüz oluşturmanın yanında sürdürülebilir bir yazılım mimarisi de sunar. Ayrıca geliştiricinin daha okunabilir ve kontrol edilebilir bir kod altyapısı ile çalışmasını sağlar.

5. Sonuç ve Değerlendirme

Bu proje, Flutter ile mobil uygulama geliştirme sürecini ve temel yazılımsal teknikleri öğretmek amacıyla geliştirilmiştir. Uygulama, kullanıcıların IMDb Top 250 filmlerini görüntülemesine, favori filmleri ekleyip çıkarmasına olanak sağlar. Ayrıca, reklam gösterimi ile gelir elde etme özelliği de entegre edilmiştir.

5.1. Kullanıcı Yorumları ve Testler

Testler sırasında, kullanıcılar favori ekleme ve çıkarma işlevlerini kolayca kullanabildiklerini belirtmişlerdir. Reklam gösterimi, kullanıcı deneyimini olumsuz etkilemeden yapılmış ve uygulama performansı başarılı olmuştur.

5.2. Geliştirilmesi Gereken Alanlar

- **Detaylı Film Bilgisi**: Kullanıcıların her film hakkında daha fazla bilgiye (oyuncular, yapım yılı, IMDb puanı) erişebilmesi sağlanabilir.
- **Offline Destek**: İnternet bağlantısı olmayan kullanıcılar için offline destek eklenebilir.
- **Reklam Çeşitleri**: Daha fazla reklam çeşidi (interstitial, banner) entegre edilebilir.

Kaynaklar

- Flutter Docs: https://flutter.dev/docs
- IMDb API: https://www.imdb.com/interfaces/
- Dart Programming Language: https://dart.dev/guides