****

**GÖMÜLÜ SİSTEM TASARIM+PROJE**

**Dr. Öğr. Üyesi Fırat AYDEMİR**

**Mobil Film-Dizi Tanıtma Uygulaması**

**2022-2023 Bahar Dönemi**

**27/03/2023 - 09/04/2023**

**3.Rapor**

**İsmail Keyvan – 201913171030**

**Muhammed Emir Gözcü – 201913171053**

**Giriş**

Filmler ve diziler gibi dijital içeriklerin üretimi ve tüketimi günümüzde oldukça yükselmiş durumda. Netflix, Amazon Prime, HBO, Disney+ ve daha birçok film ve dizi platformunun oluşturduğu bu içerik havuzunda hangi filmin veya dizinin izlenmesine karar kılmak da kullanıcıyı yoran ve düşündüren bir duruma dönüşebilir. Bizim amacımız ise bu tarz durumlarda veya arkadaşlarla yapılan film gecelerinde hangi içeriğin izlenmesinde karar kılmayı kolay bir hale getirmek ve kullanıcının izlemek istediği film veya dizi tipine en yakın içeriği kullanıcın karşısına çıkartmaktır. Bu durumda kategori türleri, öneriler ve filtreleme, yorumlar ile kullanıcıya en çok hitap eden içeriğin kullanıcı karşısına çıkartmayı planlıyoruz. Ayrıca bu içeriği izleyebileceği sanal platformları kullanıcıya sunmak ve çevresindeki sinemalarda hangi filmlerin gösterildiğini de kullanıcıya göstermek istiyoruz. Projenin gidişatına göre gerek duyulan farklı özelliklerin farkına varılırsa bunlar da eklenebilir.

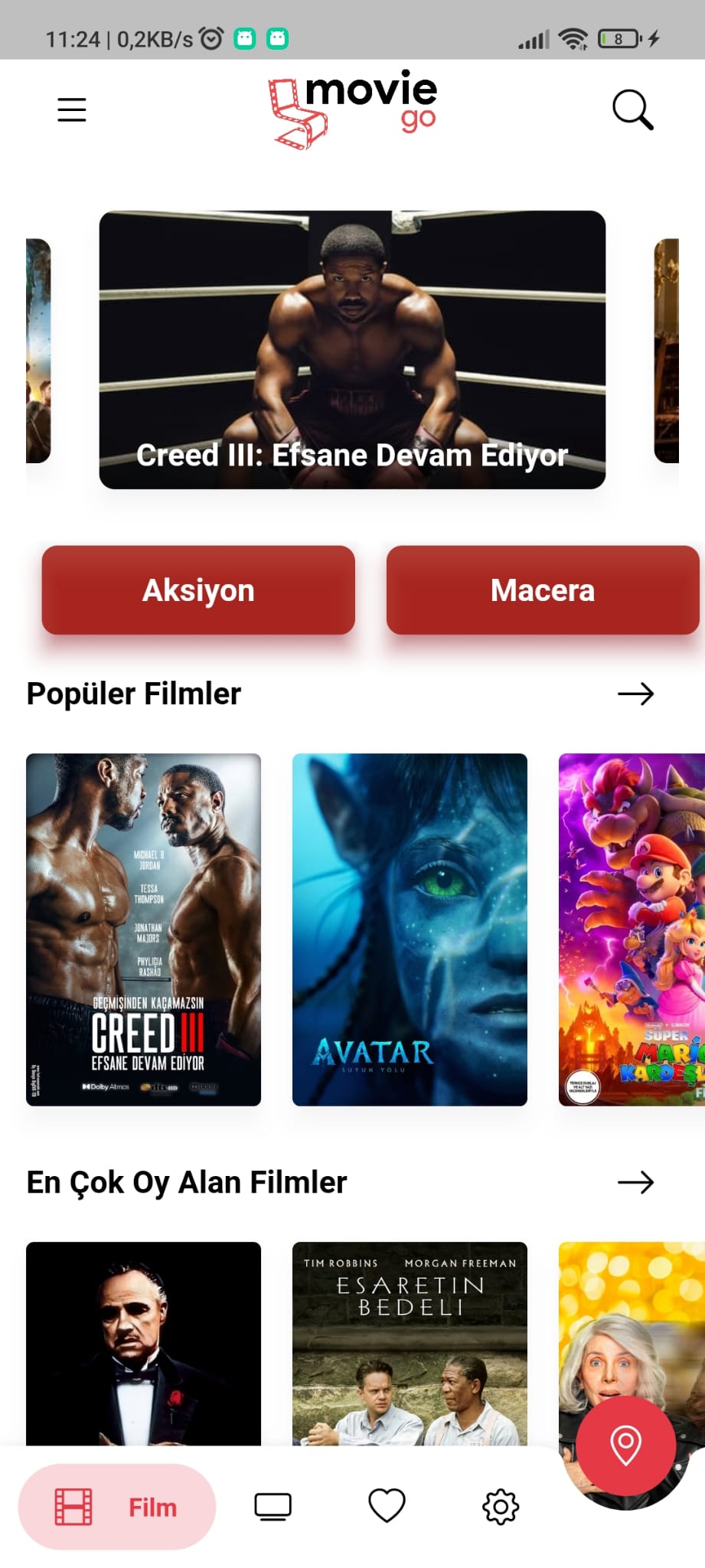
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Yapılacak İş** | **Rapor 1** | **Rapor 2** | **Rapor 3** | **Rapor 4** | **Rapor 5** | **Rapor 6** | **Rapor 7** |
| Yeni detay sayfası | X |  |  |  |  |  |  |
| Film liste sayfaları | X |  |  |  |  |  |  |
| Film arama sayfası | X |  |  | X | X | X | X |
| Fragman gösterimi | X |  | X |  |  |  |  |
| Kategori filtreleme | X |  |  |  |  |  |  |
| Drawer menü | X | X | X |  |  |  |  |
| Alt tab navigasyon barı | X | X | X |  |  |  |  |
| Oyuncu filmleri sayfası | X |  |  |  |  |  |  |
| İçeriğe göre film önerisi | X |  |  |  |  |  |  |
| İçeriği favorilere ekleme |  |  |  | X |  |  |  |
| Yakındaki sinema ve oynatılan filmlerin listelenmesi |  |  |  | X | X | X | X |
| Detaylı arama ve filtreleme |  |  | X | X | X | X | X |
| Dizilerin eklenmesi |  | X | X |  |  |  |  |
| İçerik yorumlarının gösterilmesi |  | X | X |  |  |  |  |

(Farklı raporlara ayrılmış işler güncellenecek veya değiştirilecektir.)

**Yapılan Çalışmalar**

Bu rapor döneminde öncelikle uygulama içerisinde kullanılan alt kısımdaki navigasyon barını değiştirildi ve harita üzerinde sinema lokasyonları göstermek için gerekli olan 5. butonu da eklendi. Daha hızlı ve anlaşılabilir bir kodlama için sabit değişkenlerle “enum” yapıları içerisinde tutuldu ve karşılaştırma işlemleri de bu yapılar üzerinden yapıldı. Uygulamaya karanlık mod desteği eklendi ve bu sebeple kullanılan yazı veya metin renkleri gibi temaya bağlı olan renkler “ThemeData” değişkenleri halinde kullanılmaya başlandı, uygulamada baştan aşağı bir yenilemeye gidilerek bu yapı kullanılmaya başlandı. Uygulamaya Türkçe’nin yanı sıra İngilizce dil desteği eklendi. Uygulama içerisinde kullanılan ikonlar değiştirildi. Arama kısmı çok daha detaylı hale getirildi. Uygulama arayüzünde küçük değişiklikler yapıldı.

**Navigasyon Barı**



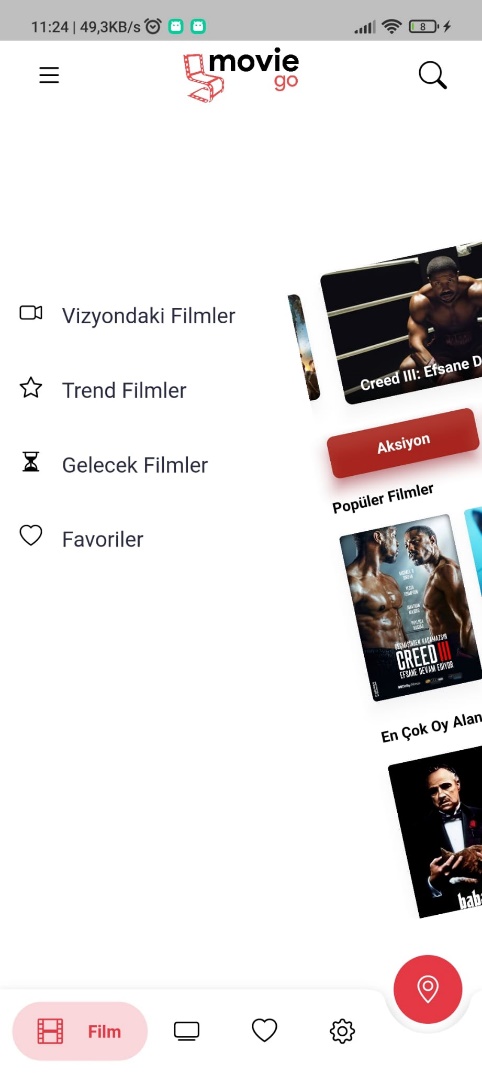
Uygulama içerisinde ekranlar arasında geçiş yapmayı daha kolay hale getirmek için ekranın alt kısmına bir navigasyon barı eklemiştik.Bu barı özelleştirerek daha sonradan ekleyeceğimiz haritalarda sinema lokasyonları gösterecek sayfa için yeni bir lokasyon ikonu ekledik. Sağdan başlayarak diğer 4 ikon ise sırasıyla film sayfası, dizi sayfası, favoriler ve ayarlar sayfaları arasında yönlendirme yapıyor.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu Navigasyon barı içerisinde “setstate” kullanıldığı için bu durumlarda sayfa yeniden oluşturulur. Bu durumda tüm sayfanın değil de sadece navigasyon barının yeniden oluşturulmasını sağlamak için bu barı farklı bir sınıf olarak kullandık ve o anki bulunulan bar indeksini “ValueSetter” kullanarak üst sınıfa ilettik.

Bu kısımda bar üzerinde herhangi bir elemana tıklanıldığında o anki indeksi tıklanılan indekse eşitleniyor ve üst sınıfa yani ana sayfaya bu bilgi “setIndex” metodu kullanılarak iletiliyor.

**İkonlar**

****

Uygulama içerisinde kullanılan ikonlarda görsel bir değişiklik yapıldı ve uygulama arayüzü ile daha uyumlu olan ikonlar kullanıldı.

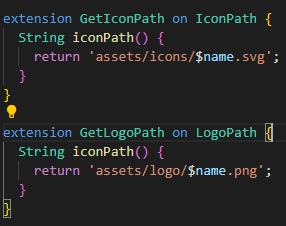
metin, ekran, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

İkonlar “assets” klasörü içerisinde “svg” formatında tutuldu.

metin içeren bir resim

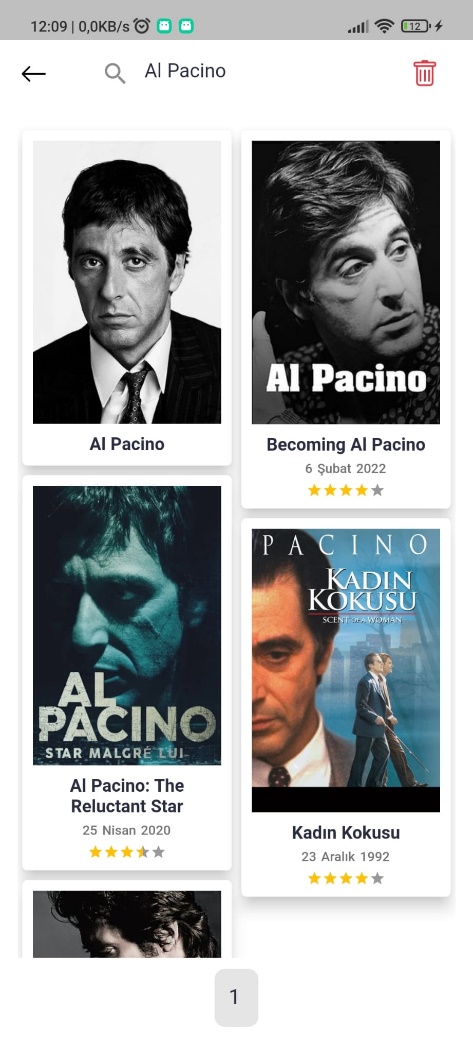
Açıklama otomatik olarak oluşturulduİkonlara erişim içim “enum” yapısında sabit değişkenler kullanıldı. Bu değişkenler ile ikonun dosya içerisinde bulunduğu konuma erişim sağlandı.

 İkonlara ulaşım için “extension” anahtar kelimesi oluşturulan uzantı metotları kullanıldı ve bu metotlar kendilerini çağırmak için kullanılan “enum” değişkeninin ismini parametre olarak alıp ikonun uygulama içerisindeki dosya konumunu geriye döndürdü.

metin içeren bir resim

Açıklama otomatik olarak oluşturulduHerhangi bir “widget” içerisinde bu yapının kullanımı ise yandaki şekilde görülmektedir.

**Detaylı Arama**

****

Arama sayfasında güncelleme yaparak sadece filmlerin değil aynı zamanda dizi ve kişilerin de listelenmesi sağlandı. Kullanıcın aradığı anahtar kelime ile eşleşen bütün sonuçlar ekranda listelendi.

metin içeren bir resim

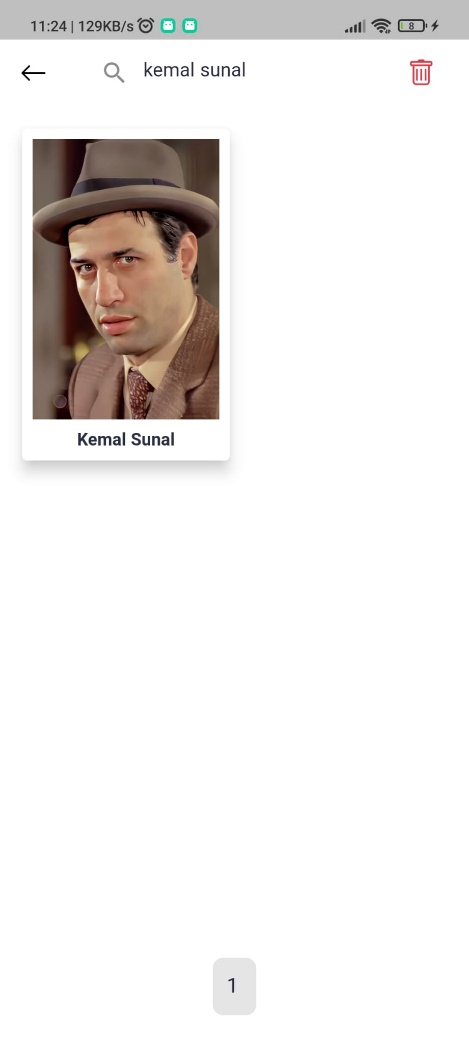
Açıklama otomatik olarak oluşturulduAPI üzerinden bize dönen içeriğin “mediaType” özelliğine göre film, dizi veya kişi bilgisi olduğu anlaşılıyor ve bu durumlara göre gösterilecek içerik belirleniyor. Tekrardan “enum” yapıları kullanılarak medya tiplerine ulaşım ve karşılaştırma daha kolay bir hale getirildi. Ayrıca tıklanılınca olarak farklı fonksiyonlar da bu tiplere göre düzenlendi.

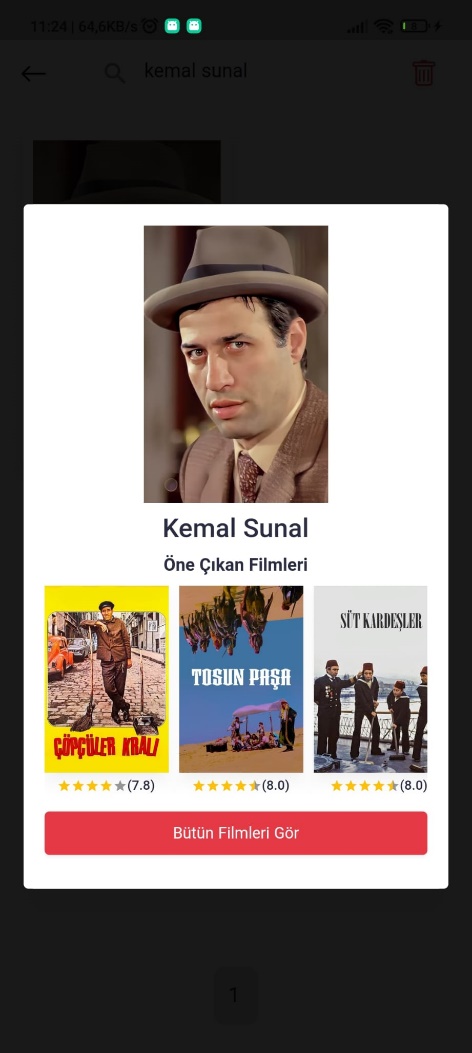
metin içeren bir resim

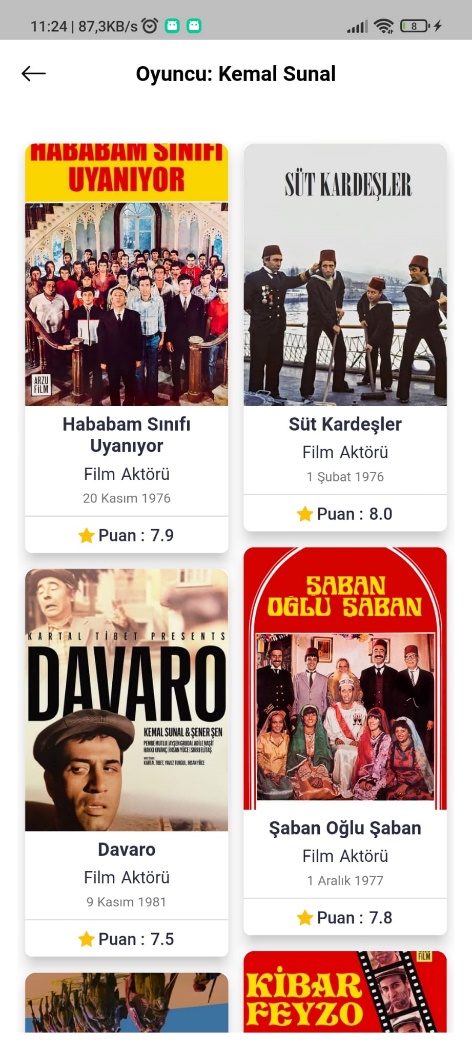
Açıklama otomatik olarak oluşturulduFarklı medya tiplerine göre işlemlerin ayrımı yandaki gibi yapıldı.

API üzerinden farklı sonuçların gelmesini sağlamak için “multisearch” özelliği kullanıldı ve istek metinleri bu durumlara göre oluşturuldu.

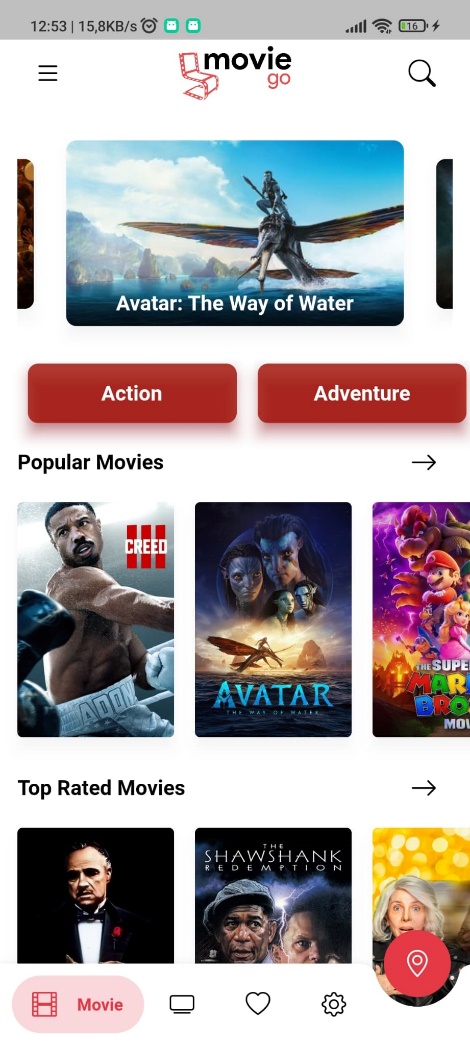
**Kullanılan istek metni:** '$\_baseuRL/search/multi?api\_key=$\_apiKey&language=$\_languageKey&query=$query&page=$page&include\_adult=false'

 Kullandığımız TheMovieDB API bize çok büyük bir veri havuzu sağlıyor bunu etkili kullanmak için kişi kısmını detaylandırdık. Arama metni her türlü içerik arasından aranıyor ve sonuçlar kullanıcı arayüzünde listeleniyor. Kişilerde, kişi resmi ve isminin bulunduğu bir kutu gösteriliyor.

 Kişi profil kutusuna tıklanıldığında ekranda mesaj diyaloğu kutusu şeklinde kişi ile ilgili filmlerin olduğu bir kutu gösteriliyor. Bu kutu içerisinde kişinin öne çıkan filmleri içerisinden 3 tanesi ve bu filmlerin puanlaması gösteriliyor.

 Profil kutusu üzerinden “Bütün Filmleri Gör” butonuna tıklanıldığında oyuncunun oynadığı diğer filmlerin listelendiği oyuncu sayfasına yönlendirme yapılıyor.

**İngilizce Dil Desteği**

**** Kullandığımız API bize yolladığımız isteklerde kullanacağımız bir “language” parametresi de sunuyor. Bu sayede içeriğin metin dilini değiştirebiliyoruz. Flutter “easy\_localization” paketini kullanarak uygulama içerisinde genel bir dil değişim sistemi kurduk. Bu sayede hem uygulama içi metinler hem de gösterilen içerikler farklı dillerde gösterilebiliyor.

metin içeren bir resim

Açıklama otomatik olarak oluşturulduUygulama içerisinde kullanılan metinlerin Türkçe ve İngilizce çevirileri “translations” klasörü içerisinde “json” dosyaları olarak tutuldu.

metin içeren bir resim

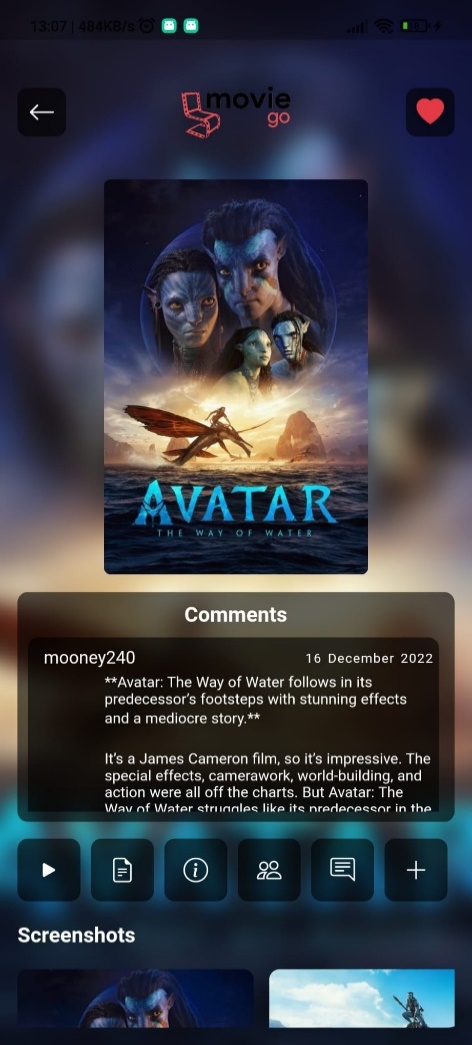
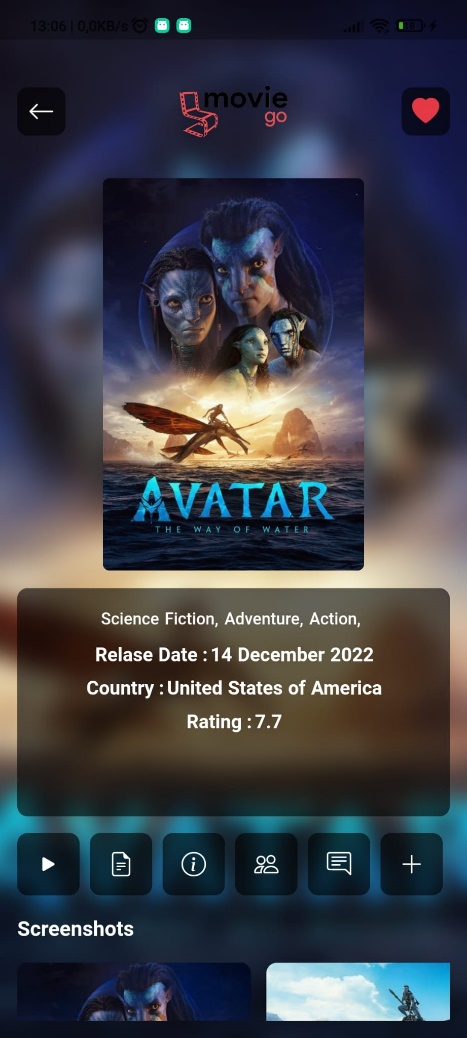
Açıklama otomatik olarak oluşturuldu Çeviri dosyaları içerisinde metne erişmek için bir değer ve bu değere karşılık gelen metin Türkçe ve İngilizce halleri ile farklı dosyalarda tutuldu.

metin içeren bir resim

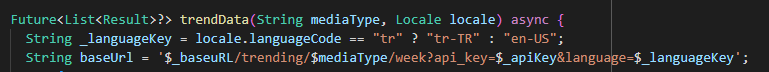
Açıklama otomatik olarak oluşturuldu Kullanılacak metine ulaşım için otomatik oluşturulmuş “LocaleKeys” sınıf kullanıldı ve “tr()” metodu ile metnin çevrilmesi sağlandı.

web sitesi içeren bir resim

Açıklama otomatik olarak oluşturuldu Uygulama içi metinlerin yanı sıra API ile bağlantılı olan film, dizi gibi içeriklerin de seçili dilde gösterilmesi sağlandı.



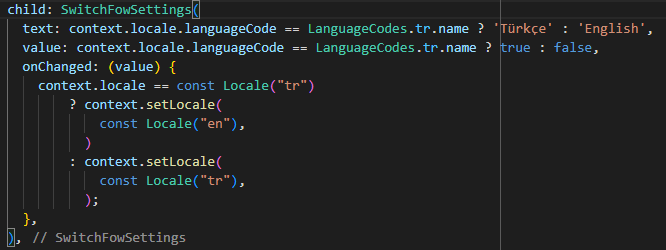
Bu sistem için o anki seçili olan dil seçeneğine erişmek için kullanılan “Locale” değişkeni kullanıldı. Bu değişkene ulaşmak için bir “BuildContext” nesnesine ihtiyacımız olduğu için bu değişken istek yaptığımız metoda parametre olarak gönderildi.



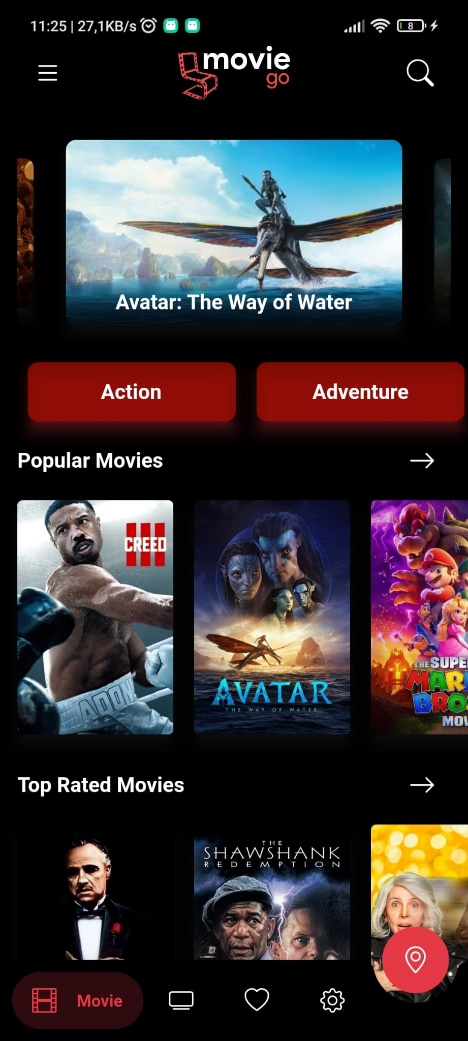
Örnek kullanım:



Uygulama içerisinde canlı olarak dil değişimi ise aşağıdaki gibi lokal dilin değiştirilmesi şeklinde yapıldı.



**Karanlık Mod**

**** Uygulamanın karanlık ortamlarda göz yormaması ve aşrı parlaklığı engellemek için karanlık mod seçeneği eklendi. Karanlık veya aydınlık modlar arasında geçiş yapıldığında değişimler canlı olarak ekrana yansıtılıyor. Kullanılan widget’lardaki değişiklikler canlı olarak uygulamaya yansıtılıyor. Uygulama içerisindeki bileşenler tema bilgisini “ThemeDataProvider” isimli bir sınıftan alıyor ve değişimleri canlı olarak uyguluyor. Ayrıca kullanılan temayı değiştirmek için de bu sınıftaki “setThemeData” metodu kullanılıyor. Kullanılan karanlık ve aydınlık temalar tanımlandıkları sınıflar ile ve sağlayıcı sınıf ile “theme” klasörü içerisinde bulunuyor.

metin, ekran görüntüsü, ekran, kapatma içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu Güncel olarak kullanılan temayı öğrenmek ve temayı değiştirmek için kullanılan “provider” sınıfı.

Uygulama içerisinde canlı olarak tema değiştirmesi aşağıdaki şeklide yapıldı.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

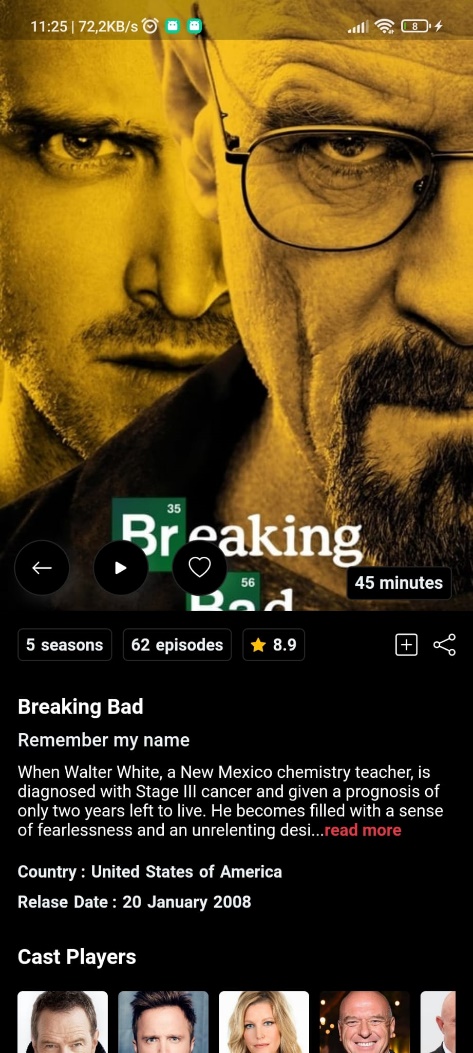
Açıklama otomatik olarak oluşturuldumetin içeren bir resim

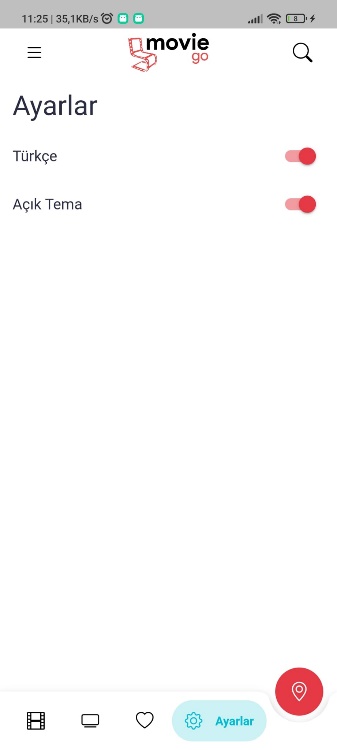
Açıklama otomatik olarak oluşturuldu

İkonlara ulaşım kısmında olduğu gibi kullanılan temaya ulaşmak için de “extension” anahtar kelimesi oluşturulan uzantı metotları kullanıldı. Temalara ulaşım için ilgili sınıfın “BuildContext” nesnesi kullanıldı.

metin içeren bir resim

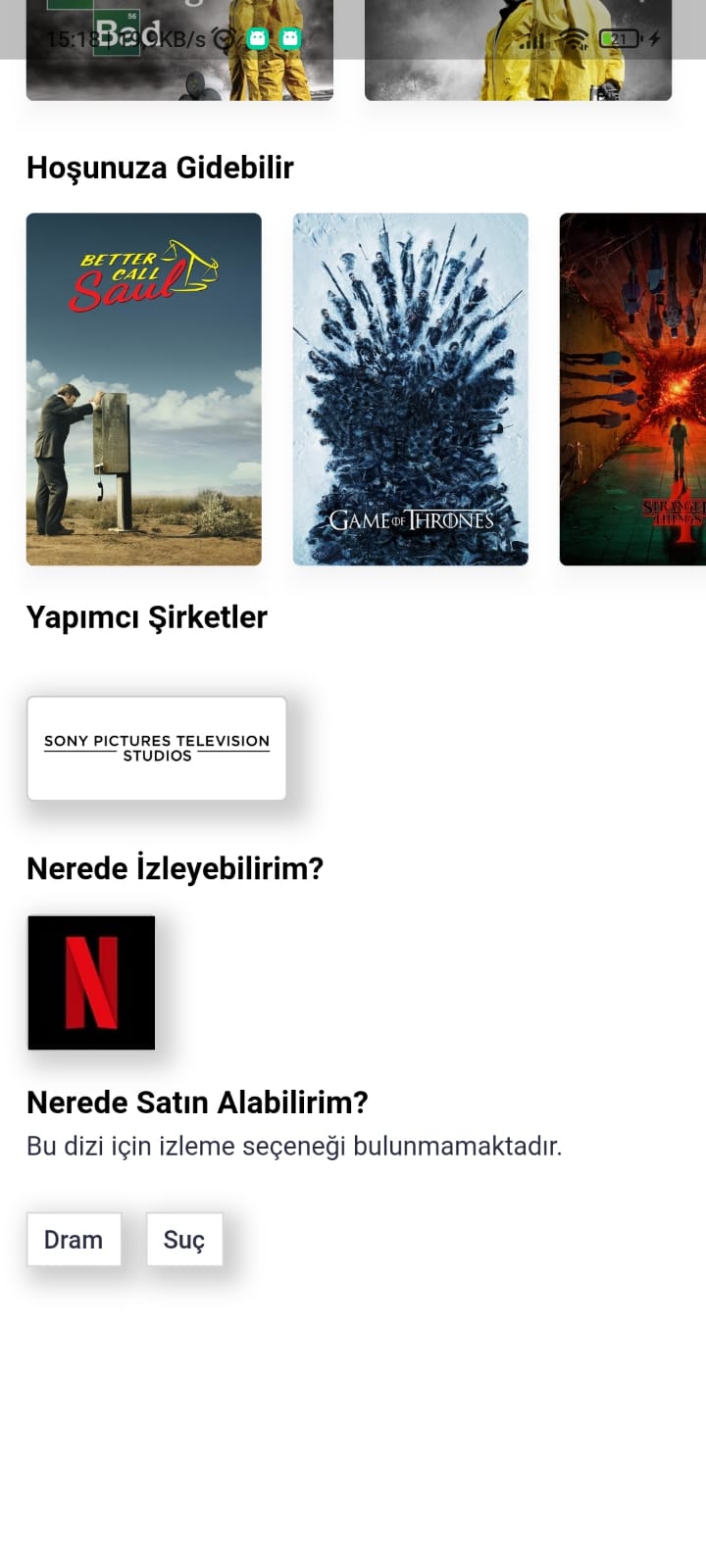
Açıklama otomatik olarak oluşturuldu

 Karanlık temaya geçildiğinde metin renkleri, arka plan rengi veya ikon renkleri gibi renkler koyu haller ile değiştiriliyor. Karanlık moddaki dizi detay sayfası soldaki gibi gözükmektedir.



Karanlık mod ve dil değiştirme kısımları ayarlar sayfasındaki “switch” butonlar ile yapılabilmektedir. Bu sayfa henüz son haline gelmedi, sayfa üzerinde görsel değişiklikler yapılacaktır.

**Detay Sayfası**



**** Film ve dizi detay sayfalarında içerik ile ilgili gösterilen bilgilerin yanı sıra içeriklerin nereden izlenebileceği veya nerelerden satın alınabileceği bilgileri de gösterildi. API kaynaklı olarak şuanda bu platformlara yönlendirme yapılamıyor.

**Yapılacak Olan Çalışmalar**

Bir sonraki rapor döneminde uygulama içerisinde küçük arayüz güncellemeleri ile favorilere ekleme kısmının ve haritalarda sinemaların listelenmesi kısmını planlamayı ve uygulama içerisine eklemeyi planlıyoruz.

**Kaynakça**

**Github**: <https://github.com/emirgzc/movie_app>

[1] <https://app.quicktype.io/>

[2]<https://www.themoviedb.org/documentation/api>

[3] <https://jsonbeautifier.org/>

[4] <https://dribbble.com/shots/5026483-Netflix-Mobile-App-Redesign>

[5] <https://developers.themoviedb.org/3/movies/get-movie-details>

[6] <https://pub.dev/packages/flutter_rating_bar>

[7] [https://pub.dev/packages/carousel\_slider](https://pub.dev/packages/carousel_slider%20)

[8] [https://pub.dev/packages/cached\_network\_image](https://pub.dev/packages/cached_network_image%20)

[9] [https://pub.dev/packages/youtube\_player\_flutter](https://pub.dev/packages/youtube_player_flutter%20)

[10] [https://pub.dev/packages/flutter\_staggered\_grid\_view](https://pub.dev/packages/flutter_staggered_grid_view%20)

[11] [https://pub.dev/packages/masonry\_grid](https://pub.dev/packages/masonry_grid%20)

[12] [https://pub.dev/packages/google\_nav\_bar](https://pub.dev/packages/google_nav_bar%20)

[13] [https://pub.dev/packages/flutter\_zoom\_drawer](https://pub.dev/packages/flutter_zoom_drawer%20)

[15] [https://pub.dev/packages/http](https://pub.dev/packages/http%20)

[16] <https://pub.dev/packages/easy_localization>

[17] <https://pub.dev/packages/bubble_bottom_bar>

[18] <https://pub.dev/packages/provider>

[19] <https://pub.dev/packages/flutter_svg>

[20] <https://pub.dev/packages/flutter_native_splash>