****

**GÖMÜLÜ SİSTEM TASARIM+PROJE**

**Dr. Öğr. Üyesi Fırat AYDEMİR**

**Mobil Film-Dizi Tanıtma Uygulaması**

**2022-2023 Bahar Dönemi**

**09/04/2023 - 23/04/2023**

**4.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 |  |  |  |
| 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 |
| Detaylı arama ve filtreleme |  |  | X | X |  |  |  |
| Dizilerin eklenmesi |  | X | X |  |  |  |  |
| İçerik yorumlarının gösterilmesi |  | X | X |  |  |  |  |
| Yükleme Animasyonları |  |  |  |  | X |  |  |

(Farklı raporlara ayrılmış işler güncellenecek veya değiştirilecektir.)

**Yapılan Çalışmalar**

Bu rapor döneminde uygulama içerisinde favorilere eklenen içeriklerin tutulabilmesi için kullanılacak olan “hive” paketi kullanımı araştırıldı. Diğer lokal veri depolama paketleri ile karşılaştırılmalar yapıldı ve kullanılması gereken paketin “hive” olmasına karar verildi. Uygulama içerisine paket kullanılarak favorilere ekleme kısmı bitirildi.

**Hive NoSql Veritabanı**

Hive, Dart ile yazılmış hızlı bir veritabanıdır. Anahtar-değer sistemi ile çalışarak kullanım kolaylığı sağlar. Uygulama içerisinde favorilere eklenecek olan içeriğin “id” numarası tutulacağı için bu tarz bir veritabanı işimizi oldukça kolaylaştırcaktır.

Örnek kullanım:

metin, yazı tipi, beyaz, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Hive paketini tıpkı bir “map” veri tipi gibi kullanırken ayrıca zaman alarak gerçekleşen “future” yapısında da kullanabiliriz. Örnek kullanım:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Favoriler**

Kullanıcın seçtiği içerikleri favoriler kısmına eklemesi ve sonrasında internet bağlantısı olmasa dahi bu içeriklere ulaşabilmesi için içerikler lokal depolamada “Hive” paketi kullanılarak tutuldu. Hive içerisinde tutulacak veriler birer box (kutu) içerisinde tutulur ve verilere bu kutular aracılığı ile erişilebilir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu Uygulamamızın ana sınıfı içerisinde Hive paketi başlatıldı ve gerekli konfigürasyonlar yapıldı. Film ve diziler için ayrı kutular oluşturuldu ve veri tabanına kayıtları yapıldı.

metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Film ve dizi bilgilerinin ekleme veya silme gibi işlemlerinin yapılacağı sınıfların her birisinde bu metodların bulunmasını zorunlu kılmak amacı ile bir abstract (soyut) sınıf oluşturuldu.

Bu soyut sınıfta bulunan metotların hepsini soyut sınıfı “extends” anahtar kelimesi ile kendisini bağlayan sınıf, kendi içerisinde bu metotların üzerine yazarak yeni metotlarla değiştirmek zorundadır.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu Buradaki gibi HiveTvManager sınıfımız bu soyut sınıfı “extend” ettiği için soyut sınıftaki “add” metodunun üzerine yeni bir metot yazmıştır.

**HiveManager**

metin, ekran görüntüsü, yazılım, işletim sistemi içeren bir resim

Açıklama otomatik olarak oluşturulduSoyut sınıftaki ekleme, silme, getirme ve hepsini getirme metotlarının bu sınıfta üzerinde yazılma zorunluluğunu sağlamak için soyut sınıf “extends” ediliyor.

Film içeriklerini bir “box” değişkeni kullanarak veri tabanına yazdırıp okuyacağımız için bu değişken oluşturuluyor fakat değer atanmıyor.

Sınıfımızın kurucu metodunda “box” değişkenine değer ataması yapılıyor. Bu sınıf ile bir işlem yapılacağı zaman sınıftan bir nesne oluşturulduğunda bu atama yapılıyor. “box” değişkeni ile veri tabanında yapılacak işlemlerin “DetailMovie” veri tipi kullanılarak yapılacağı belirtiliyor.

Ekleme, silme, hepsini getirme ve tek bir tane getirme metotları soyut sınıftaki tanımlamaların üzerine yazılarak yeniden oluşturuluyor. Hive veri tabanına yazma ve okuma gibi işlemler zaman alan işlemler olduğu için “await” anahtar kelimesi ile işin bitmesi bekleniyor. Bu sebeple metotlar da asenkron metotlar olarak tanımlanıyor. Veri tabanına eklenecek veya silinecek nesnenin veri tipi olan DetailMovie tipindeki nesneler ile işlemler yapılıyor. DetailMovie ve DetailTv sınıfları film ve dizi bilgilerinin bulunduğu json dosyalarının Dart sınıfları şeklinde tutulduğu ve json’dan Dart sınıfına dönüşümlerin yapıldığı sınıflardır.

metin, ekran görüntüsü, yazılım, işletim sistemi içeren bir resim

Açıklama otomatik olarak oluşturulduHive ile lokal depolamada tutulacak film veya dizi bilgilerinin her ayrıntısını depolamada tutmak zaman ve kaynak harcamasını artırıyor. Sadece ihtiyacımız olan alanları cihaz depolamasına yazdırmak ve geri okumak için gerekli alanları @HiveField olarak işaretledik.

Şekil - Film ile ilgili tüm bilgiler

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil - Sadece hafızada tutulacak alanlar

Hive ile depolayacağımız film veya dizi bilgilerini @HiveField olarak işaretledikten sonra “flutter packages pub run build\_runner build” komutunu kullanarak bir adaptör sınıfı oluşturuyoruz. Ana sayfa içerisinde Hive paketinin kurulumunu yaparken bu adaptör sınıfını registerAdapter metodunu kullanarak kaydediyoruz. Bu sayede Hive paketi ile yapılacak olan işlemlerde kullanılacak nesneyi tanımlıyoruz.

**SnackBar Diyalog**

metin, ekran görüntüsü, yazılım, işletim sistemi içeren bir resim

Açıklama otomatik olarak oluşturulduUygulama içerisinde kullanıcının etkinliklerinde kullanıcıyı bilgilendirmek için mesaj diyalog kutuları oluşturuldu. Kullanıcın uygulama ile iletişimini güçlendirmek için favorilere ekleme veya çıkarma işlemleri için kullanıcıya mesajlar gösteriliyor.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil - Örnek mesaj kutusu çağırımı

Kullanıcı etkileşimlerde mesaj diyalog kutuları ekranın alt kısmından kayan bir animasyonla ekrana geliyor ve tekrar kaydırılarak kullanıcı tarafından kapatılabiliyorlar. Ekranda 2 saniyelik bir zamandan sonra kendiliğinden kayboluyorlar. Film ve dizi sayfalarında favorilere ekleme:

****

****

**Favori İşlemleri**

metin, yazı tipi, ekran görüntüsü, tipografi içeren bir resim

Açıklama otomatik olarak oluşturulduSoyut sınıflardan ekleme, silme gibi işlemler için nesnelerimiz oluşturuyoruz. Ayrıca tüm favorilere ulaşmak için de listeler oluşturuyoruz.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturulduSınıfımız oluşturulurken soyut sınıf nesnelerinin ve diğer değişkenlerin atamasını yapıp “\_getItems()” metodu ile tüm içerikleri depolamadan alıyoruz.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu Tüm film ve diziler kullanıcıya gösterileceği için cihaz depolamasından okunuyor.

metin, ekran görüntüsü, yazı tipi içeren bir resim

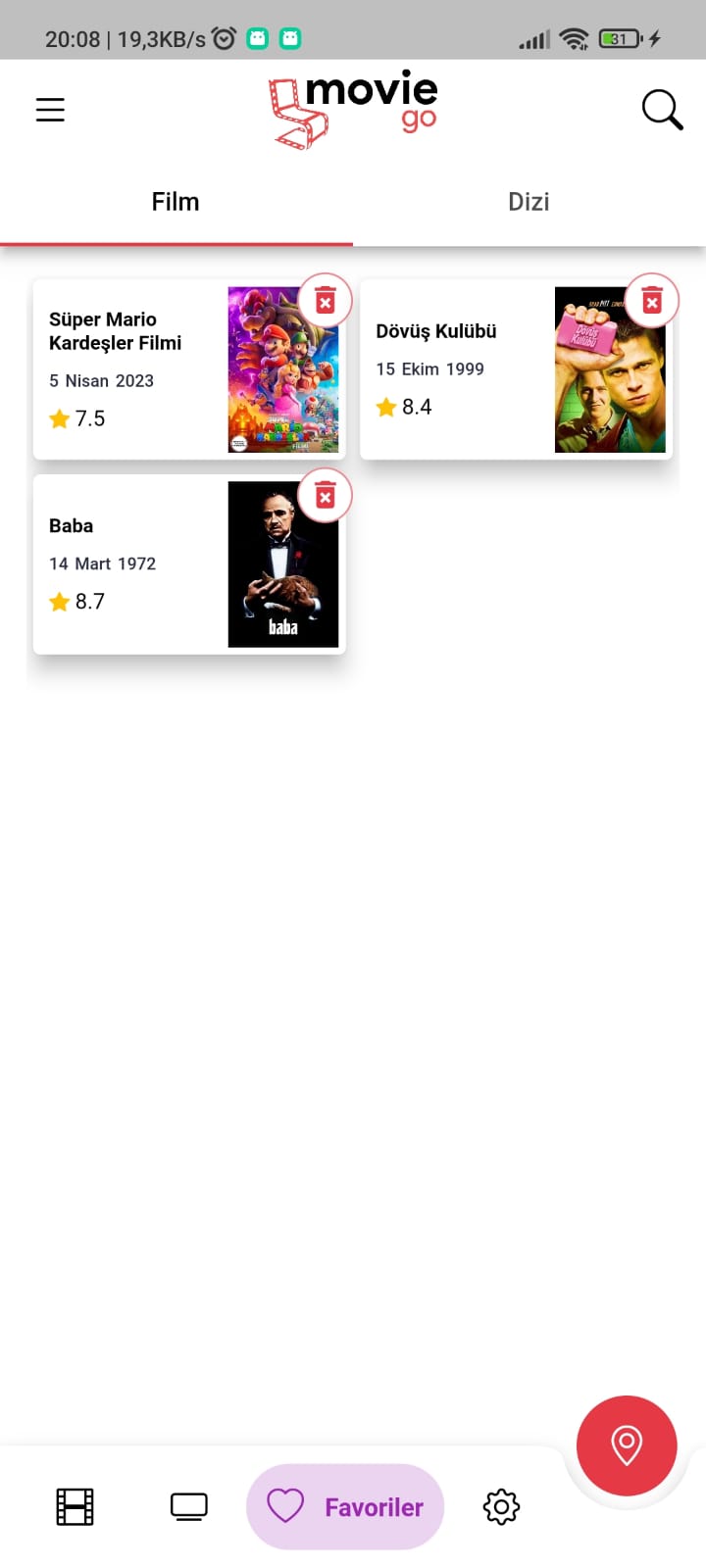
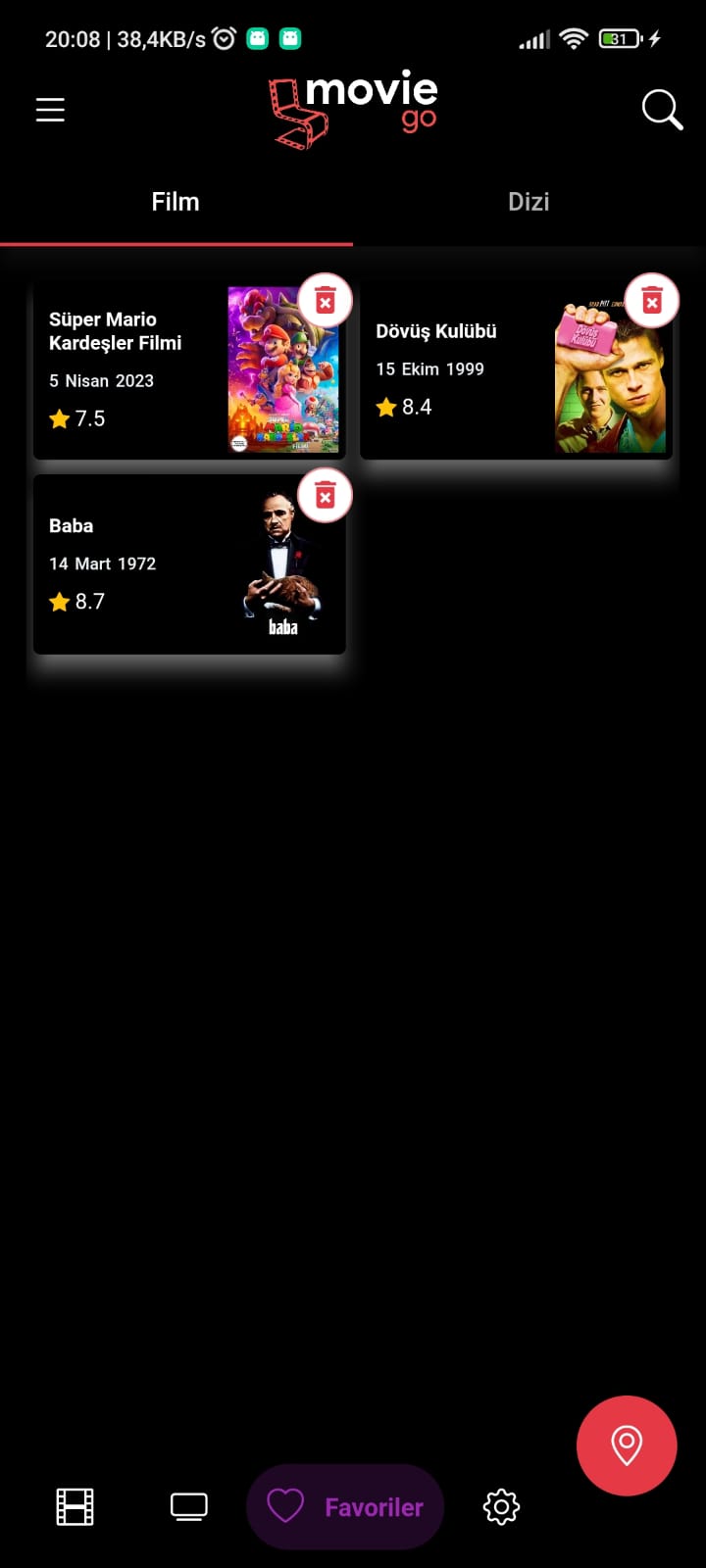
Açıklama otomatik olarak oluşturulduFavori butonuna tıklanıldığında içeriği favorilere ekliyoruz. Favorilere eklerken soyut sınıfımız üzerinden ekleme metodunu, soyut sınıfımızı “extends” eden sınıftan alıyoruz. Burada soyut sınıfımızı DetailMovie veri tipin tutacak şekilde oluşturduğumuz için filmler kısmına ekleme yapıyoruz. Sonrasında bilgi mesaj kutusu gösteriliyor.

metin, ekran görüntüsü, yazı tipi, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

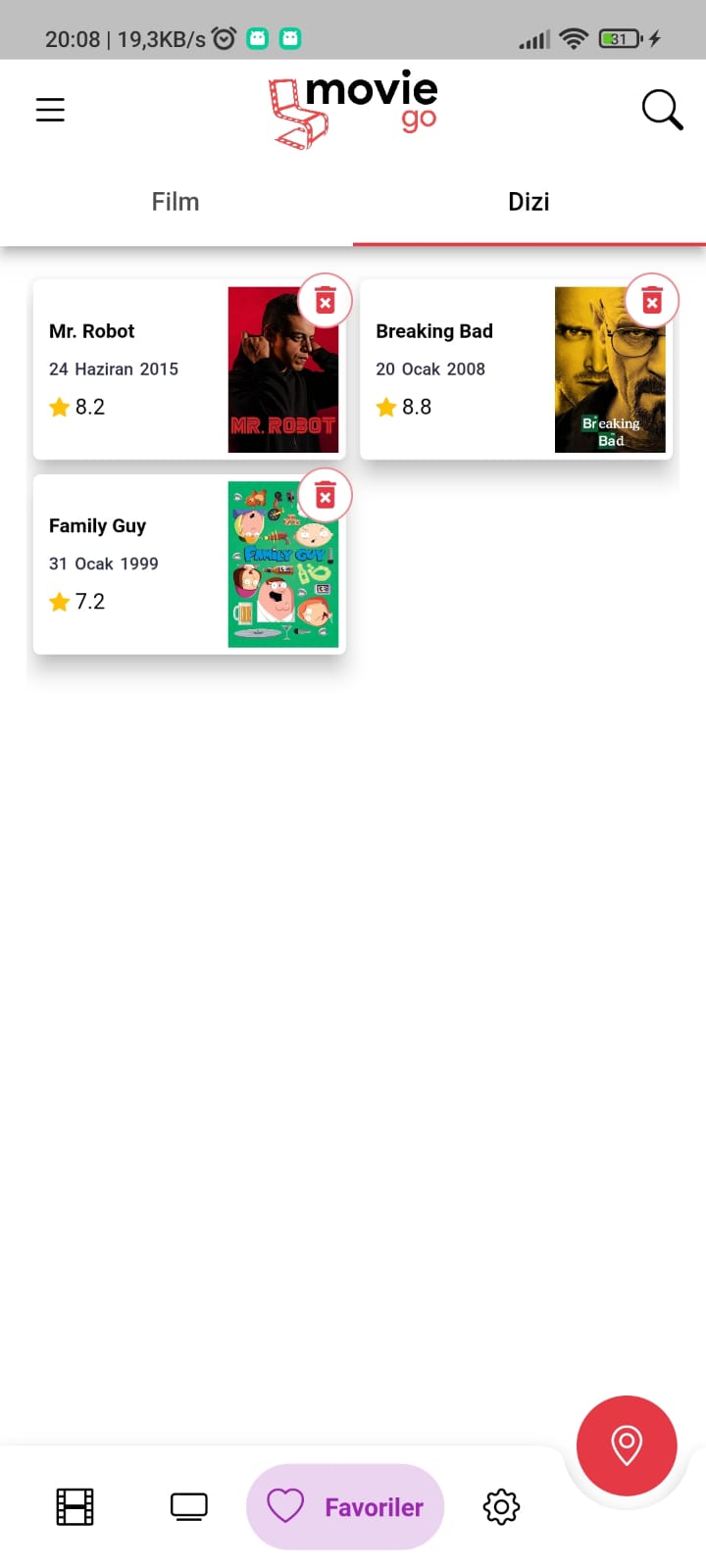
İçerik silineceği zaman oluşturulduğu soyut sınıf nesnesi kullanılarak ve içeriğin index değeri verilerek lokal depolamadan silme işlemi yapılıyor. Sonrasında bilgi mesaj kutusu gösteriliyor.

**Favoriler Arayüzü**

****

Şekil - Favori filmler sekmesi karanlık mod

Şekil - Favori filmler sekmesi

****

Şekil - Favori diziler sekmesi

**Yapılacak Olan Çalışmalar**

Bir sonraki rapor döneminde uygulama içerisindeki kullanılan tema bilgisini de lokal depolamada tutmayı ve kullanıcıya daha akıcı bir arayüz sağlamak için yükleme animasyonları tamamlamayı 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>

[21] <https://pub.dev/packages/hive>