

NETFLİX

Furkan Aydoğan
Bilgisayar Mühendisliği, Kocaeli Üniversitesi
180202085
180202085@kocaeli.edu.tr

Sefa Berke Kara
Bilgisayar Mühendisliği, Kocaeli Üniversitesi
180202086
180202086@kocaeli.edu.tr

Giriş

Netflix uygulaması “.exe” uzantılı dosyanın çalıştırmasıyla sorunsuz bir şekilde başlar. Eğer kayıtlı bir kullanıcı var ise filmlerin bulunduğu ana sayfa ekranına bilgiler girilerek giriş yapılır. Eğer kayıtlı herhangi bir bilgi yok ise kayıt ol butonuna tıklanarak kayıt olma sayfasından işlem gerçekleştirilir.

Tasarım

Netflix Uygulaması programlanma aşamaları altta belirtilen başlıklar altında açıklanmıştır

Özet

Netflix uygulaması 3 tane ara yüzden oluşmaktadır. Bunlar Kullanıcı Giriş Sayfası, Kullanıcı kayıt sayfası ve Kullanıcı Yönetim sayfasıdır. Kullanıcının kayıt kontrolü Kullanıcı Kayıt sayfasında yapılır eğer kullanıcı kayıtlı değil ise Kullanıcı Kayıt sayfasından kayıt olur. Kayıt olunan veriler SQLİTE veri tabanına kayıt edilir. Kullanıcı Yönetim sayfasında filmler JButton dizisi ile buton olarak bastırılır. Kullanıcı hem kategoriye göre hem de isme göre arama yapabilmektedir.

Temel Bilgiler

Windows 10 işletim sistemine sahip bilgisayar ile geliştirme yapılmıştır. Program Java programlama dilinde geliştirilmiş olup, tümleşik geliştirme ortamı olarak “Netbeans” kullanılmıştır

Algoritma

Netflix uygulaması ilk olarak run() fonksiyonun içinde panel class'ında nesne üretilerek panelin parametresiz constructor'una girer. Girilen bu fonksiyondaki initComponents() fonksiyonu çağrılarak Kullanıcı giriş sayfası paneli ekrana bastırılır. Ardından bağlan() fonksiyonuna giderek Sqlite ile java arasında bağlantı kurulur. Kullanıcı email ve şifresini girer ve giriş yap butonuna tıklar. Tıklamasıyla birlikte giriş_yap() fonksiyonu çağrılır. giriş sırasında alınan email ve şifre değişkenlerde tutulur Girilen bilgilerin doğru olup olmadığı program tarafından Sqlite kullanıcı tablosundaki kayıtlı veriler çekilerek kontrol edilir. Eğer kayıtlı değil ise yanlış bilgi olduğuna dair ekrana mesaj bastırılır. Kullanıcı kayıtlı değil ise kullanıcı kayıt sayfasına gitmesi beklenir. Kayıt ol butonuna tıklanarak Kullanıcı Kayıt Sayfasına geçiş yapılır. Kullanıcının kayıt olması için gerekli bilgiler alınır. Kayıt_ekle() fonksiyonu ile bilgiler alınarak girilen bilgiler değişkenlerde tutulur. Sqlite veri tabanındaki kullanıcı tablosuna alınan veriler kayıt edilir. Ekrana kayıt başarılı bir şekilde gerçekleştirildi mesajı bastırılır. Her kullanıcının birbirinden farklı kullanıcı id'si bulunmaktadır. Bu id kullanıcı tablosunda Primary Key olarak eklenmiştir ve kullanıcı tablosundaki tek primary key budur. Kullanıcı kayıt sayfası içerisinde kullanıcının kategori tercihlerine göre film önerisi alması için “kategori seçimi için tıklayınız ” butonuna tıklanır. Kullanıcının karşısına 10 tane farklı kategorinin olduğu sayfa bastırılır. Kullanıcıdan bu kategoriler arasından 3 tane

kategoriye seçmesi istenir. Kullanıcının seçtiği kategorilere göre her kategori için 2 tane olmak üzere toplam 6 tane film önerisi kullanıcıya sunulması için puan_sirala() fonksiyonu çağrılır. Bu fonksiyon parametre olarak tür tablosunda her tür için belirlenmiş olan tür_id'sını alır .Bu id tür tablosundaki tek Primary Keydir.puan_sirala() fonksiyonuna giriş yaptıktan sonra sorgu ve sorgu1 olmak üzere 2 tane String tipinde sql sorgusu oluşturulur. Ve bu iki sorgu execute edilerek ResultSet tipinden tanımlanmış res ve res1 değişkenlerine atanır. Daha sonra while döngüsü içine girerek eğer kullanıcın izlediği film daha önceden puanlanmış ise res değişkeni if içinde ilerletilerek ArrayList tipindeki list değişkenine eklenir. Eğer kullanıcının izlediği dizi yada film daha önceden izlenmedi ise res1 değişkeni if içinde ilerletilerek seçilen kategorilerden sırayla ilk iki film yada dizi alınarak list değişkenine eklenir.Bu list değişkenine eklenen isimler puan_sirala() fonksiyonundan return edilerek fonksiyonun çağırıldığı yerdeki başka bir ArrayList değişkenine atanır. Bu adımlar bittikten sonra seçilen film ya da dizi isimleri tablo halinde kullanıcıya bastırılması için tablo_ekle() fonksiyonu çağrılır. Bu fonksiyon film isimlerinin atandığı list değişkeni seçilen kategori ismi ve sqlite'de tur_tablosunda her kategori için belirtilmiş id'leri alır.tablo_ekle() fonksiyona çağırıldıktan sonra ilk olarak Object tipinde satırların ve sütunların oluşması için 6 uzunluğunda bir rowData değişkeni oluşturulur. Ardından for döngüsü içinde fonksiyon da parametre olarak alınan list değişkeninin içinde film isimleri satir dizisine atanır.3 tane kategori için bu işlemler yapıldığında ve if koşulu içindeki şart sağlandığı zaman alınan bu altı isim rowData değişkenine atılarak kullanıcıya öneriler sunulur. Kullanıcı bu önerileri gördükten sonra kayıt ol butonuna sonra da geri dön butonuna tıklayarak giriş sayfasına geri döner. Ardından kayıt olduğu mail ve şifre ile giriş yapar. Giriş yap butonuna tıklanmasıyla Kullanıcı Yönetim sayfasının bastırılabilmesi için initilaze() fonksiyonu çağrılır ve film adeti kadar JButton dizisi buton oluşturulur. veri_cek() fonksiyonuna giderek Sqlite veri tabanındaki program tablosundaki film isimleri return edilir. Bu film isimleri buton üzerine yazdırılarak kullanıcıya bastırılır. Kullanıcı istediği bir filme tıklayarak filmi izle butonu ile izlemeye başlayabilir ve bu esnada tıkladığı zamanın bilgileri bir değişkende tutulur. İzle butonuna tıklandıktan sonra kullanıcının bu filmi daha önce izleyip izlemediğini anlamak için kontrol() fonksiyonu çağrılır. Eğer kullanıcının tıkladığı butondaki isim Dizi türüne ait ise if bloğuna girerek daha önce giriş_id , program_id ,

bolum_index verileri ile kullanıcıProgram tablosunda arama işlemi yapar duruma göre kontrol() fonksiyonu da True ya da False değerini döndürür. Eğer kullanıcının seçtiği seçeneğin türü Film ise bu seferde giriş_id ve program_id'ye göre arama yaparak olup olmadığını kontrol ederek True ya da False değerini döndürür. Eğer fonksiyondan False değeri dönse ve kullanıcı bu filmi veya diziyi ilk defa izliyorsa ekle() fonksiyonu çağrılır. ekle() fonksiyonu çalıştığı zaman ilk önce SimpleDateFormat tipinde biçim değişkeni oluşturulur. Ardından Date tipinde tarih değişkeni oluşturulur. Daha önceden alınan ve program_tur değişkeninde tutulan içeriğe göre eğer kullanıcı Film izliyor ise if bloğu içine girerek kullanıcı için belirlenmiş kullanıcı_id , program_id ,izlediği tarih,ve bölüm sayısı sqlite'daki kullanıcıProgram tablosuna kayıt edilir. Eğer kullanıcı Dizi izliyor ise Film bloğuna ek olarak izlediği dizinin bölümü kullanıcıProgram tablosuna eklenir. Kullanıcı dilediği vakit duraklat butonuna tıklayarak izlediği filmi ya da diziyi durdurabilir. Bu butona tıkladığı an zaman() fonksiyonu çalıştırılır ve tekrardan zaman alınır ve bir değişkene atanır. Kullanıcı eğer daha önce filmi izlenmediyse saat, dakika, saniye değişkenlerine sıfır atanır. Ardından son tutulan zaman ile ilk tutulan zaman çıkarılarak ve belli işlemler yaparak ekle() fonksiyonunda eklenmiş veriler burada süresi Update edilerek kullanıcıProgram tablosuna yazdırılır. Kullanıcı bu filmi eğer daha önce izleyip ve tekrardan izlemek istiyorsa film yani kontrol() fonksiyonundan True değeri döner ise bu adımda filmdevam() fonksiyonu çağrılır. Burada yapılan ilk işlem kullanıcının izlediği tür kontrol ettirilir. Kullanıcının izlediği tür Dizi ise giriş_id ,program_id ve bolum_index'e göre arama yapılır. KullanıcıProgram tablosunda bu kayıta erişildiği vakit izleme_suresindeki veri saat ,dakika ,saniye olarak parçalanarak bu değişkenlere atanır. Ardından kullanıcı izlediği film ya da diziyi duraklatma butonuna tıklayarak durdurur ve zaman() fonksiyonu tekrar çağrılır. Bu çağırılma aşamasında kullanıcı filmi daha önce izlediği için saat ,dakika ,saniye değişkenine atılan süreler ile izle ve duraklat arasındaki zaman farkı ile toplanarak yeni süre o filmin izlenme süresine atanır. Kullanıcı dilerse izle butonuna ve ardından duraklat butonuna tıkladıktan sonra film için uygun olan puanı verebilir.1-10 arasında seçtiği değerlerden dilediğini verdiği takdir de seçilen değer program_puan değişkenine atanır sonra giriş_id ve program_id'ye göre arama yapılarak o satıra gelmesiyle satırın puanı güncellenir. Kullanıcı bu işlemleri yaptıktan sonra dilerse kategoriye ya da isme göre arama yapabilir. Eğer kategoriye arama yapmak isterse JComboBox ile 10 tane kategori kullanıcıya sunulur. secActionPerformed() fonksiyonu ile seçilen kategori tur_arama() fonksiyonuna gönderilir. Tur_arama() fonksiyonu String tipinde

parametre alır. Bu parametrede kullanıcının seçtiği kategori tutulur. Bu değişken için de tutulan kategori ilk olarak tür tablosunda where koşulu ile tür aranır ve tur_id'si tutulur. Tutulan bu id önceden oluşturulmuş program_tur (Filmlerin veya dizilerin hangi kategoriye sahip olduklarını tutan sql tablosudur İki tane sütünü vardır. Bu tablodaki program_id(Foreign Key), program tablosundaki program_id'den referans almıştır. Yine bu tablodaki tur_id(Foreign Key) de, tur_tablosundan referans almıştır(Foreign Key).Ve bu program_tur tablosundaki 2 değişken Primary Key olarak atanmıştır.) tablosundaki tur_id ile eşitlenir. Eşitleme işleminden sonra döndürülen satırların program_id'leri tutulur.Tutulan bu program_id'ler "IN " arama koşulu ile program_tablosundaki program_id ile eşleşen verilerin tamamı döndürülür. Bu döndürülen program adları butonlara atanarak her bir buton setVisible(True) 'ya çevirilir. Bu sayede filmlerin ya da dizilerin ekrana bastırılması sağlanır. Eğer kullanıcı film veya dizi ismine göre arama yapmak isterse arama yapılacak alana filmin veya dizinin ismini girerek arama yap butonuna tıklarsa şu işlemler gerçekleşir ;

İlk olarak girilen metin String tipinde bir değişkende tutulur. Ardından buton fonksiyonuna giderek ekranda halihazırda bastırılmış butonları setVisible(false) yaparak ekran da gözükmesini engeller. Ardından arama_yap() fonksiyonu çağrılır. String tipinde tutulan bu değişken arama_yap() fonksiyonuna parametre olarak yollanır. Gönderilen bu parametre Sql program tablosundan program_adi adlı sütun da alınan bu parametre "LIKE" Sql komutu ile aranır. Döndürülen bu değerler daha önceden False olarak tanımlanan butonlar döndürülen satır kadar True yapılarak program adları da butonlara atanır. Böylece kullanıcıya program bir bütün olarak sunulur. Kullanıcı isterse çıkış yap butonu ile çıkış yapıp Kullanıcı Giriş sayfasına geçiş yaparak tekrardan başka bir hesaba dilerse giriş yapabilir ya da çarpı butonuna tıklayarak programdan çıkış yapabilir. Program böylece son bulur.

Sözde Kod

1. run() fonksiyonun içinde panel class'ında nesne üretetek panelin parametresiz constructor'una gir.
2. Girilen bu fonksiyondaki initComponents() fonksiyonu çağır ve Kullanıcı giriş sayfası paneli ekrana bastır.

3. bağlan() fonksiyonuna giderek Sqlite ile java arasında bağlantı kur.
4. Kullanıcı email ve şifresini girer ve giriş yap butonuna tıklar
5. Tıklamasıyla birlikte giriş_yap() fonksiyonunu çağır.
6. Giriş sırasında alınan email ve şifre değişkenlerde tut.
7. Girilen bilgilerin doğru olup olmadığını Sqlite kullanıcı tablosundaki kayıtlı veriler çekilerek kontrol et.
8. Eğer kayıtlı değil ise yanlış bilgi olduğuna dair ekrana mesaj bastır.
9. Kullanıcı kayıtlı değil ise kullanıcı kayıt sayfasına gitmesi bekle.
10. Kayıt ol butonuna tıklanmasıyla Kullanıcı Kayıt Sayfasına geçiş yap.
11. Kullanıcının kayıt olması için gerekli bilgileri al.
12. Kayıt_ekle() fonksiyonu ile bilgiler alınarak girilen bilgiler değişkenlerde tut.
13. Sqlite veri tabanındaki kullanıcı tablosuna alınan veriler kayıt ed.
14. Ekrana kayıt başarılı bir şekilde gerçekleşti mesajı bastır.
15. Kullanıcı kayıt sayfası içerisinde kullanıcının kategori tercihlerine göre film önerisi alması için "kategori seçimi için tıklayınız " butonuna tıklanmasını bekle.
16. Kullanıcının karşısına 10 tane farklı kategorinin olduğu sayfayı bastır.
17. Kullanıcıdan bu kategoriler arasından 3 tane kategoriye seçmesi iste.
18. Kullanıcının seçtiği kategorilere göre her kategori için 2 tane olmak üzere toplam 6 tane film önerisi kullanıcıya sunulması için puan_sirala() fonksiyonunu çağır.
19. Bu fonksiyon parametre olarak tür tablosunda her tür için belirlenmiş olan tür_id'sini al.
20. Bu id tür tablosundaki tek Primary Key olarak ata.p
21. uan_sirala() fonksiyonuna giriş yap.
22. sorgu ve sorgu1 olmak üzere 2 tane String tipinde sql sorgusu oluştur.
23. Bu iki sorgu execute edilerek ResultSet tipinden tanımlanmış res ve res1 değişkenlerine ata.
24. while döngüsü içine girerek eğer kullanıcın izlediği film daha önceden puanlanmış ise res değişkeni if içinde ilerletilerek ArrayList tipindeki list değişkenine ekle.
25. Bu list değişkenine eklenen isimler puan_sirala() fonksiyonundan return edilerek fonksiyonun çağırıldığı yerdeki başka bir ArrayList değişkenine ata.

26. Seçilen film ya da dizi isimleri tablo halinde kullanıcıya bastırılması için `tablo_ekle()` fonksiyonu çağır.
27. .Bu fonksiyon film isimlerinin atandığı list değişkeni seçilen kategori ismi ve `sqlite`'de `tur_tablosunda` her kategori için belirtilmiş `id`'leri alır.`tablo_ekle()` fonksiyona çağırıldıktan sonra ilk olarak Object tipinde satırların ve sütunların oluşması için 6 uzunluğunda bir `rowData` değişkeni oluştur.
28. `for` döngüsü içinde fonksiyon da parametre olarak alınan list değişkeninin içinde film isimleri satır dizisine ata.
29. Üç tane kategori için bu işlemler yapıldığında ve `if` koşulu içindeki şart sağlandığı zaman alınan bu altı isim `rowData` değişkenine atılarak kullanıcıya önerileri sun.
30. Kullanıcı bu önerileri gördükten sonra kayıt ol butonuna sonra da geri dön butonuna tıklayarak giriş sayfasına geri dön.
31. Kayıt olduğu mail ve şifre ile giriş yap.
32. Giriş yap butonuna tıklanmasıyla Kullanıcı Yönetim sayfasının bastırılabilmesi için `initilaze()` fonksiyonu çağır.
33. Film veya dizi adeti kadar `JButton` dizisi buton oluştur.
34. `veri_cek()` fonksiyonuna giderek `Sqlite` veri tabanındaki program tablosundaki film isimleri return et.
35. . Bu film isimleri buton üzerine yazdırılarak kullanıcıya bastırılır
36. Kullanıcı istediği bir filme tıklayarak filmi izle butonu ile izlemeye başlayabilir ve bu esnada tıkladığı zamanın bilgileri bir değişkende tut.
37. İzle butonuna tıklandıktan sonra kullanıcının bu filmi daha önce izleyip izlemediğini anlamak için `kontrol()` fonksiyonunu çağır.
38. Eğer kullanıcının tıkladığı butondaki isim Dizi türüne ait ise `if` bloğuna girerek daha önce `giris_id` , `program_id` , `bolum_index` verileri ile `kullaniciProgram` tablosunda arama işlemi yap.
39. Duruma göre `kontrol()` fonksiyonu da `True` ya da `False` değerini döndür.
40. Eğer kullanıcının seçtiği seçeneğin türü Film ise bu seferde `giris_id` ve `program_id`'ye göre arama yaparak olup olmadığını kontrol ederek `True` ya da `False` değerini döndür.
41. Eğer fonksiyondan `False` değeri dönse ve kullanıcı bu filmi veya diziyi ilk defa izliyorsa `ekle()` fonksiyonu çağır.
42. `ekle()` fonksiyonu çalıştığı zaman ilk önce `SimpleDateFormat` tipinde biçim değişkeni oluştur.
43. Daha önceden alınan ve `program_tur` değişkeninde tutulan içeriğe göre eğer kullanıcı Film izliyor ise `if` bloğu içine girerek kullanıcı için belirlenmiş `kullanici_id` , `program_id` ,izlediği tarih,ve bölüm sayısı `sqlite`'daki `kullaniciProgram` tablosuna kayıt et.
44. Eğer kullanıcı Dizi izliyor ise Film bloğuna ek olarak izlediği dizinin bölümü `kullaniciProgram` tablosuna ekle.
45. Kullanıcı dilediği vakit duraklat butonuna tıklayarak izlediği filmi ya da diziyi durdurabilir. Bu butona tıkladığı an `zaman()` fonksiyonu çalıştır ve tekrardan zamanı al ve bir değişkene ata.
46. Kullanıcı eğer daha önce filmi izlenmediyse saat, dakika, saniye değişkenlerine sıfır ata.
47. Son tutulan zaman ile ilk tutulan zaman çıkarılarak ve belli işlemler yap.
48. `ekle()` fonksiyonunda eklenmiş veriler burada süresi Update et. `kullaniciProgram` tablosuna yazdır.
49. Kullanıcı bu filmi eğer daha önce izleyip ve tekrardan izlemek istiyorsa film yani `kontrol()` fonksiyonundan `True` değeri döner ise bu adımda `filmdevam()` fonksiyonu çağır.
50. Kullanıcının izlediği tür Dizi ise `giris_id` ,`program_id` ve `bolum_index`'e göre arama yap.
51. `kullaniciProgram` tablosunda bu kayıta erişildiği vakit `izleme_suresindeki` veri saat ,dakika ,saniye olarak parçalanarak bu değişkenlere ata.
52. Kullanıcı izlediği film ya da diziyi duraklatma butonuna tıklayarak durdurur ve `zaman()` fonksiyonu tekrar çağır.
53. Bu çağırılma aşamasında kullanıcı filmi daha önce izlediği için saat,dakika ,saniye değişkenine atılan süreler ile izle ve duraklat arasındaki zaman farkı ile toplanarak yeni süre o filmin izlenme süresine ata.
54. Kullanıcının puan vermesini bekle.
55. Seçilen değer `program_puan` değişkenine ata.
56. `giris_id` ve `program_id`'ye göre arama yap.
57. O satıra gelmesiyle satırın puanını güncellenir
58. Eğer kategoriye arama yapmak isterse `JComboBox` ile 10 tane kategori kullanıcıya sun.
59. `secActionPerformed()` fonksiyonu ile seçilen kategori `tur_arama()` fonksiyonuna gönder.
60. `Tur_arama()` fonksiyonu String tipinde parametre alır. Bu parametrede kullanıcının seçtiği kategori tut.
61. Tutulan bu id önceden oluşturulmuş

62. program_tur'ü tur_id ile eşitle.
63. Eşitleme işleminden sonra döndürülen satırların program_id'leri tut.
64. Tutulan bu program_id'ler "IN " arama koşulu ile program_tablosundaki program_id ile eşleşen verilerin tamamı döndür.
65. Bu döndürülen program adları butonlara atanarak her bir buton setVisible(True) 'ya çevir.
66. Bu sayede filmlerin ya da dizilerin ekrana bastırılması sağla.
67. Eğer kullanıcı film veya dizi ismine göre arama yapmak isterse arama yapılacak alana filmin veya dizinin ismini girerek arama yap butonuna tıklamasını bekle.
68. İlk olarak girilen metin String tipinde bir değişkende tut.
69. Ardından buton fonksiyonuna giderek ekranda halihazırda bastırılmış butonları setVisible(false) yap.
70. Ekran da gözükmesini engelle ve arama_yap() fonksiyonunu çağır.
71. String tipinde tutulan bu değişken arama_yap() fonksiyonuna parametre olarak yolla.
72. Gönderilen bu parametre Sql program tablosundan program_adi adlı sütun da alınan bu parametre "LIKE" Sql komutu ile ara.
73. Döndüren bu değerler daha önceden False olarak tanımlanan butonlar döndürülen satır kadar True yapılarak program adları da butonlara atanır.
74. Döndüren bu değerler daha önceden False olarak tanımlanan butonlar döndürülen satır kadar True yapılarak program adları da butonlara ata.
75. Program son.

Karşılaştığımız Sorunlar

Sqlite bağlantısında bağlama konusunda sorun yaşadık.

Filmler için oluşturulan butonların yerlerini tayin etmekte zorlandık.

Veri tabanının birden fazla bağlanma isteği yollandığı için kilitlenmesi sorunu ile karşılaştık.

Projenin Bize Kattığı Yararlar:

Veri tabanı kullanımını gerçek bir proje üzerinde deneme fırsatı edindik ve Sql bilgimizi daha da geliştirdik.

Bir veri tabanı ile programlama dili arasındaki bağlantının nasıl gerçekleştirildiğini öğrendik.

Java ile birden fazla Arayüz yaparak bu konuda kendimizi geliştirdik.

Kullanılan fonksiyonlar

public void actionPerformed():

Bu fonksiyon ekranda oluşturulan butonların hangisini tıkladığı ve tıklanan butonun üstündeki ismi alarak izle() fonksiyonuna gönderir.

public void arama_yap():

Bu fonksiyon aldığı String tipindeki parametre ile s sql'de arama yaparak dönen satırları buton ile ekrana bastırır.

public static void baglan():

Java ile SQLITE arasındaki bağlantıyı kurmayı sağlar.

public static void baglanti_kes():

Java ile SQLITE arasındaki bağlantıyı keser.

public void buton():

Ekranda oluşan butonları setVisible(false) ile görünmez yapar.

public void ekle():

İzlenilen filmin yada dizinin verilerini kullanıcıProgram tablosuna ekler.

public void filmdevam():

Database de kayıtlı olan filmin süresini alarak saat,dakika,saniye değişkenlerine atama yapar.

public void girisYap():

Kullanıcının e-mail ve şifresini kontrol ederek Database de kayıtlı ise bu fonksiyon ile giriş yapar.

```
private void initialize():
```

Film ya da Dizi sayısı kadar buton oluşturur ve panele ekler.

```
public void izle(String s):
```

Daha önceden Database'ye kayıt edilmiş verilere çekerek yeni bir pencere açar ve çektiği verileri gerekli yerlere bastırarak kullanıcının izleceği bir ortam sunar.

```
public void kayit_ekle():
```

Kullanıcı kayıtlı değil ise bu fonksiyon sayesinde programa kayıt olur.

```
public boolean kontrol():
```

Film ya da Dizin daha önce izlenip izlenilmediği kontrol eder.

```
public ArrayList puan_sirala(int id):
```

Aldığı değişken ile sorgu oluşturularak istenilen kategorideki filmlerin yada dizilerin en yüksek puan alan iki ismini döndürür.

```
private void secActionPerformed():
```

Seçilen filmin adını alarak tur_arama() fonksiyonuna bu değişkeni yollar.

```
public void tablo_ekle(ArrayList list, String ad, int temp):
```

Aldığı değişkenler ile kullanıcıya en çok puan alan film yada dizileri seçilen kategori altında tabloya ekleme yapar.

```
public void temizle():
```

Kayıt ekranında girilen verileri silmeye yarar.

```
public void tur_arama(String s):
```

Aldığı değişken ile sql sorgusu çalıştırılarak dönen satırlar buton ile ekrana bastırılır.

```
public ResultSet veri_cek():
```

Database'ye daha önceden kaydedilmiş

program tablosundan verileri çekerek ResultSet tipinde veriyi döndürür.

```
public void zaman():
```

Kullanıcının izlediği dizi yada filmin süresini hesaplayıp Databasesde yeni süreyi güncellemeye yarar.

```
private void jButton16ActionPerformed():
```

Arama yap butonuna basıldığı anda arama_yap() fonksiyonunu çalıştırır.

Sonuçlar

Netflix uygulaması veri tabanına daha önceden girilmiş verileri çekerek üzerinde gerekli işlemleri yaparak kullanıcıya istediği film veya dizileri sunar .Aynı zaman da kullanıcı kayıt olma sırasında alınan veriler de veri tabanına kayıt edilir.

Kaynakça

Java GridLayout

<https://www.javatpoint.com/GridLayout>

Array that Holds JButton Objects

<https://stackoverflow.com/questions/9814566/array-that-holds-jbutton-objects>

SQLite Java

<https://yazdoldur.com/programlama/java/sqlite-java/>

Veri tabanında İlişki Türleri

<https://teknikakil.com/veritabani/veritabani-tasarimi/veritabaninda-iliski-turleri-bire-bir-bire-cok-coka-cok/>

Ekran Görüntüleri



Uygulamaya Hosgeldiniz
Lutfen giriş yapınız...

NETFLIX

E-mail :
Sifre :

Kullanıcı e-mail ve şifresini girer.Kayıtlı değil ise Kayıt ol butonuna tıklar.



FİLM TÜRLERİ

Lutfen 3 tane secim yapınız

Kullanıcı Kayıt Ol sayfasındaki ” Kategori seçimi için tıklayınız.” Butonuna tıkladığında bu sayfa karşısına gelir.



Kayıt ekranına Hosgeldiniz

E mail adresi giriniz :

Adınızı Giriniz :

Sifre Giriniz :

Dogum tarihi giriniz :

Örn: 1999.11.01

[Geri dön](#)

Kayıt işlemi için bilgiler doldurulur.Geri dön butonuna tıklanarak kullanıcı giriş sayfasına geçiş yapılır.

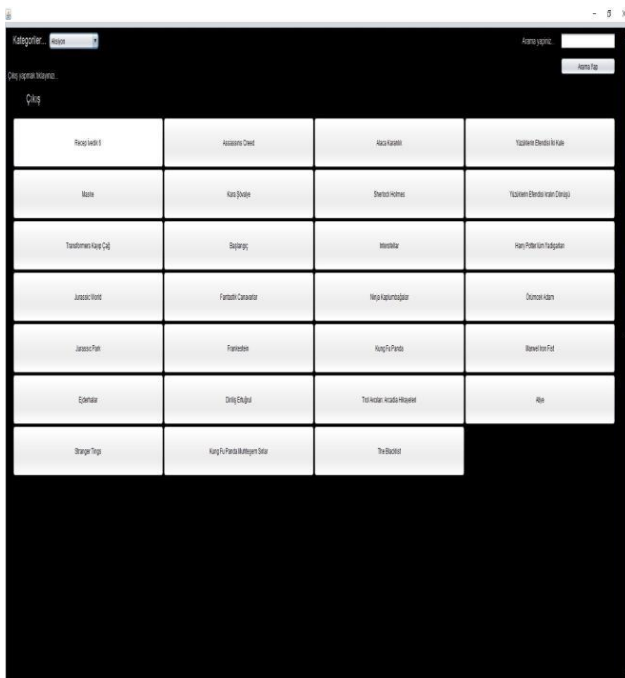


| İzleyebileceğiniz En İyi Filmler | | |
|----------------------------------|--------------------------------------|----------------------------------|
| aksiyon | Belgesel | Bilim ve Doğa |
| Maske The Blacklist | Aydaki Son Adam Rakamlarla Tahmin | Gezegelimiz 72 sevimli hayvan |

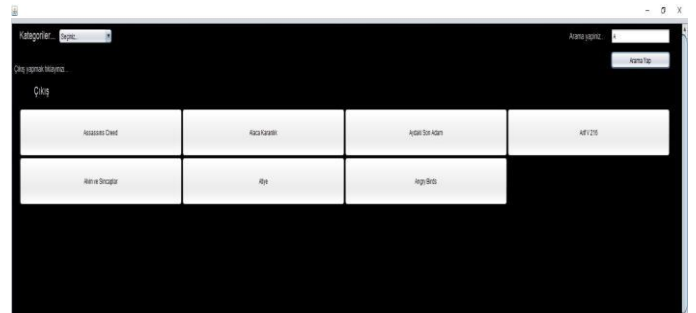
Kullanıcı seçtiği kategorilerden sonra karşısına bu pencere gelir. Eğer film veya dizilere hiç puan verilmediyse program tablosundaki kategorilere göre ilk iki film veya dizi atanır. Kullanıcı eğer kayıt olduktan sonra puan verirse tekrar bu tabloya dönüş yapıldığında bu tablo güncellenerek en yüksek puan alan ilk iki film veya dizil bastırılır.



Kullanıcın giriş işlemini başarılı bir şekilde gerçekleştirmesi durumunda bu sayfa karşısına gelir.



Kullanıcı sol üst köşedeki Kategori seçim butonundan istediği herhangi bir kategoriye seçmesi durumunda o kategoriye ait film veya diziler listelenir.(örn: Aksiyon kategorisi seçilidir.)

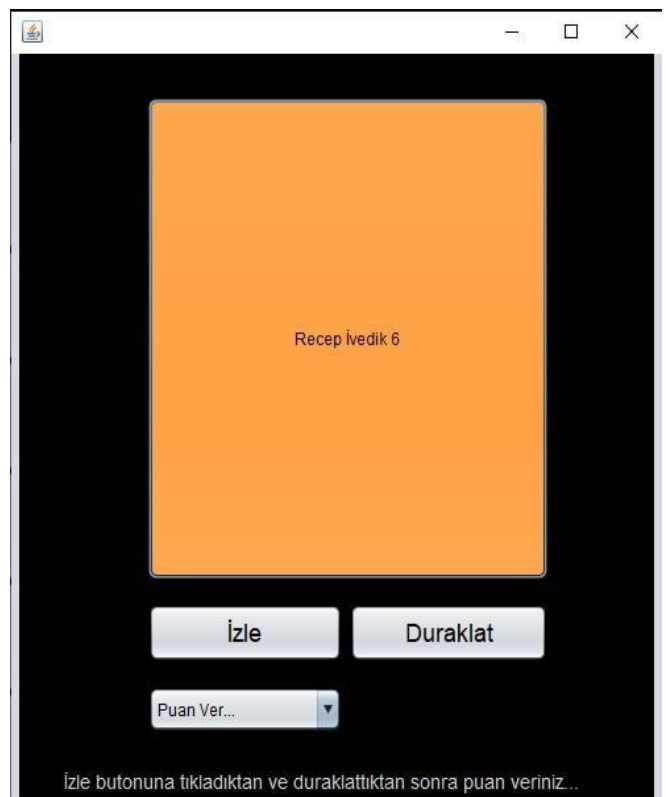


Örnek :A ile başlayan film veya diziler aranmıştır.

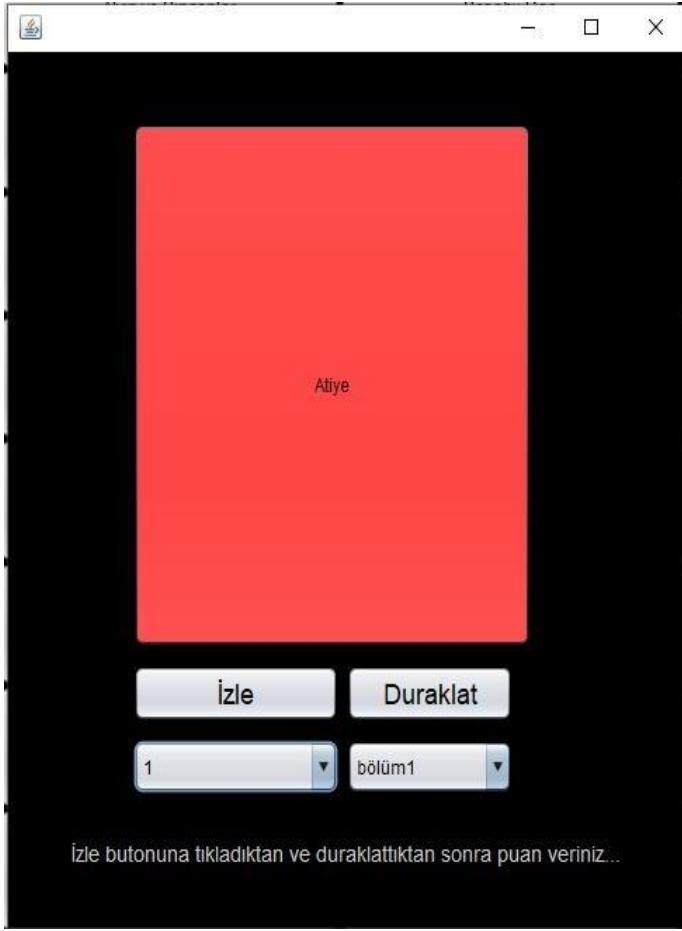


Örnek : Atiye dizisi aranmıştır.

Arama kısmına herhangi bir veri yazılmadan arama yap butonuna tıklandığında bütün film veya diziler ekrana bastırılır.



Eğer kullanıcın seçtiği tür film ise bu pencere karşısına gelir.



Kullanıcının seçtiği tür dizi ise bu pencere karşısına gelir .Bölüm seçimi yapılmadan dizi izlenemez. İzle butonuna tıkladıktan ve duraklattıktan sonra puan verilir ve puan veri tabanına kayıt edilir. İzlenmeden puan verilemez.

ER Diyagramı

