

NETFLİX VERİTABANI PROJESİ

Müberra ÇELİK – Taha Batuhan TÜRK
180202102 - 180202007

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
180202102@kocaeli.edu.tr - 180202007@kocaeli.edu.tr

Özet-Proje , Veritabanı Yönetim Sistemlerinden yararlanarak Netflix dizi veya film izleme platformunu geliştirmeyi amaçlayan, nesneye yönelik programlamaya dayalı, Java dilinde yazılmış görsel bir arayüze sahip masaüstü uygulamasıdır.

Anahtar Kelimeler : Nesneye yönelik programlama , grafik arayüz, veritabanı, JAVA, SQL, Veritabanı Yönetim Sistemleri .

I. GİRİŞ

Program, kullanıcı giriş sayfası, kullanıcı kayıt sayfası ve kullanıcı yönetim sayfası bulunan bir arayüzden oluşmaktadır. Programların bilgileri “Netflix DB.xlsx” dosyasından çekilir ve “netflix.db” isimli veritabanına kaydedilir. Giriş sayfasında, kullanıcı e-posta adresini ve parolasını yazar, bu bilgilerin doğruluğu veritabanından kontrol edilir. Kullanıcı kayıtlı ise hesabına giriş yapabilir ve içerik izleyebileceği veya arayabileceği yönetim sayfasına yönlendirilir. Eğer kullanıcı yanlış veya eksik bilgi girdiyse ya da kayıtlı değil ise kullanıcıya bir uyarı mesajı verilir. Kullanıcı kayıt olmak isterse “Şimdi Aramıza Katıl” butonuna tıklayarak kayıt sayfasına yönlendirilir. Kayıt sayfasında kullanıcı ad-soyad, doğum tarihi, e-posta, parola ve parola onay alanlarını doldurarak kayıt olur. Girilen bu bilgiler veritabanında tutulur. Kullanıcı eğer doldurulmayan alan bırakmışsa veya parolalar uyuşmuyorsa bunun uyarı bilgisi kullanıcıya verilir. Kullanıcının karşısına kayıt olduktan sonra en yüksek puan alan 6 dizi veya film listeleyebilmek için 3 tane tür seçebileceği bir panel çıkar. Kullanıcı istediği 3 türü seçer ve “GÖSTER” butonuna tıkladıktan sonra dizi veya filmler listelenir. “Giriş yap” butonuna tıkladıktan sonra tekrar giriş sayfasına yönlendirilir e-posta ve parolasını kullanarak yönetim sayfasına giriş yapar. Yönetim sayfasında kullanıcı dizi veya filmin türüne ya da ismine göre arama yapabilir, programların bilgilerini görebilir ve izleyebilir.

II. YÖNTEM

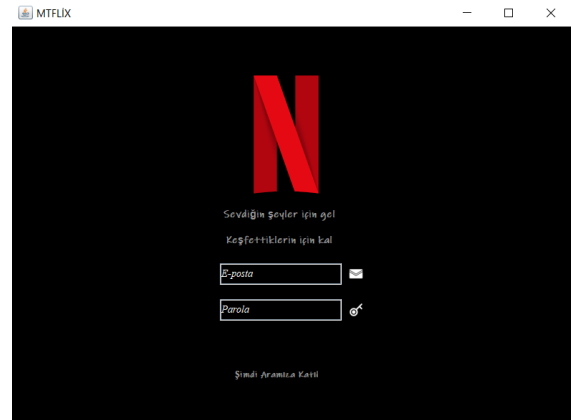
Program NetBeans ortamında Java programlama dili ile yazılmıştır. Projede Inheritance , Polymorphism, Veritabanı Yönetim Sistemleri, SQL ve Swing kavramları kullanılmıştır. Kodda Constructor (Yapıcı) metotlar tanımlanmıştır. Kodun içeriği threadler, action listener’lar, metotlar, classlar, döngüler ve if – else koşul durumlarından oluşmaktadır.

III. DENEYSEL SONUÇLAR

Programı çalıştırdığımızda ilk olarak kullanıcının karşısına gif olarak (Şekil.1) deki gibi bir karşılama ekranı çıkar ve hemen ardından giriş sayfası ekrana gelir. (Şekil.2)



Şekil.1 Kullanıcıyı karşılayan giriş ekranı



Şekil.2 Kullanıcı Giriş Sayfası

Kullanıcı e-posta adresini ve parolasını kullanarak sağ alttaki ok tuşuna tıklar ve hesabına giriş yapar. Eğer kullanıcı e-postasını (Şekil.2.1) ya da parolasını (Şekil.2.2) yanlış veya eksik yazarsa ya da kullanıcının yazdığı bilgilerde bir hesap yoksa (Şekil.2.3) deki gibi kullanıcıya bir uyarı mesajı verilir.

The screenshot shows a login interface with a large red 'N' logo at the top. Below the logo, the text 'Sevdiğin şeyler için gel' and 'Keşfettiklerin için kal' is displayed. There are two input fields: the first contains 'muberra@' and the second is empty. A red error message 'Geçerli bir e-posta girmelisin' is shown below the first field. At the bottom, there is a button labeled 'Şimdi Aramıza Katıl'.

Şekil.2.1 Kullanıcı e-postayı eksik yazdığımda kullanıcıya verilen uyarı mesajı

The screenshot shows the same login interface as Şekil.2.1. The first input field contains 'muberra@gmail.com' and the second field contains '***'. A red error message 'Doğru yazdığından emin misin?' is shown below the second field. At the bottom, there is a button labeled 'Şimdi Aramıza Katıl'.

Şekil.2.2 Kullanıcı parolayı yanlış yazdığımda kullanıcıya verilen uyarı mesajı

The screenshot shows the same login interface. The first input field contains 'muberracelik@gmail.com' and the second field contains '***'. A red error message 'Üzgünüm böyle biri aramızda yok' is shown below the second field. At the bottom, there is a button labeled 'Şimdi Aramıza Katıl'.

Şekil.2.3 Kullanıcının girdiği bilgilere dair bir hesap yoksa kullanıcıya verilen uyarı mesajı

Kullanıcı eğer kayıt olmak isterse “Şimdi Aramıza Katıl” butonuna tıklayarak kayıt sayfasına yönlendirilir. (Şekil.3) Kullanıcı sol üstte bulunan ok tuşuna tıklarsa giriş sayfasına tekrar geri dönebilir. Kullanıcı gerekli alanları doldurduktan sonra kayıt olur. Boş alan bırakılırsa (Şekil.3.1) veya parolalar uyuşmazsa (Şekil.3.2) ya da böyle bir kullanıcı mevcut ise (Şekil.3.3) deki gibi kullanıcıya bir uyarı mesajı verilir.

The screenshot shows a registration page with a large red 'N' logo at the top. Below the logo, there are five input fields: 'Ad-Soyad', 'Doğum Tarihi', 'E-posta', 'Parola', and 'Parola Onay'. Each field has a corresponding icon to its right. At the bottom, there is a button labeled 'Şimdi Aramıza Katıl'.

Şekil.3 Kullanıcının karşısına çıkan kayıt sayfası

Şekil.3.1 Herhangi bir alan boş bırakıldığında kullanıcıya verilen uyarı mesajı

Şekil.3.3 Girilen bilgilerde bir hesap varsa kullanıcıya verilen uyarı mesajı

Şekil.3.2 Girilen Parola ve Parola Onay alanları uyuşmuyorsa kullanıcıya verilen uyarı mesajı

Kullanıcı sağ alttaki ok tuşuna tıklayıp kayıt olduktan sonra en yüksek puanı alan 6 tane dizi veya film önerilerini listelemek için kullanıcıyı 3 farklı program türü seçebileceği bir panel karşılar. Program kullanıcının 3 taneden az veya fazla tür seçmesine izin vermez. Kullanıcı seçim yapmadan giriş yapmak isterse sol üstteki ok tuşuna basarak bu panelden de tekrar giriş sayfasına geri dönebilir. (Şekil.4)

Şekil.4 Kullanıcının program türü seçebileceği panel

Kullanıcı sevdiği 3 türü seçtikten sonra “GÖSTER” butonuna tıklayarak seçtiği türlere göre her türden ikişer tane olacak şekilde 6 tane dizi veya filmin isim, bölüm sayısı, puanı ve süresi olmak üzere tüm bilgileri listelenir. (Şekil.5)

Belgesel	Anime	Drama
Aydaki Son Adam	Sonic X	Yerçekimi
Bölüm Sayısı: 1	Bölüm Sayısı: 54	Bölüm Sayısı: 1
Puan: 10	Puan: 7	Puan: 8
95.0dk.	60.0dk.	88.0dk.
72 sevimli hayvan	Beyblade	Dangal
Bölüm Sayısı: 1	Bölüm Sayısı: 23	Bölüm Sayısı: 1
Puan: 10	Puan: 6	Puan: 7
95.0dk.	54.0dk.	145.0dk.

Şekil.5 Kullanıcının seçtiği türlere göre listelenen programlar ve programların bilgilerini gösteren panel.

Kullanıcı Şekil.5 deki panelde sağ altta bulunan “Giriş Yap” butonuna tıklar ve tekrar giriş sayfasına yönlendirilir. Giriş sayfasında kullanıcı e-posta ve parolasını kullanarak giriş yapabilir ve yönetim sayfasına erişebilir. Yönetim sayfasında kullanıcı isme ve türe göre arama yapabilir, arama yaptığı programların bilgilerini görebilir ve bu programları izleyebilir. (Şekil.6)

Belgesel	Anime	Drama
Aydaki Son Adam	Sonic X	Yerçekimi
Bölüm Sayısı: 1	Bölüm Sayısı: 54	Bölüm Sayısı: 1
Puan: 10	Puan: 7	Puan: 8
95.0dk.	60.0dk.	88.0dk.
72 sevimli hayvan	Beyblade	Dangal
Bölüm Sayısı: 1	Bölüm Sayısı: 23	Bölüm Sayısı: 1
Puan: 10	Puan: 6	Puan: 7
95.0dk.	54.0dk.	145.0dk.

Şekil.6 Kullanıcı yönetim sayfasının görüntüsü

Türe göre arama yapmak için kullanıcı ekranın sol tarafındaki (Şekil.7) türlerden birini seçerek o türlere ait bütün dizi ve filmleri bilgileri ile birlikte (Şekil.7.1) deki gibi panelin ortasında listeleyebilir.

Aksiyon ve Macera
Belgesel
Bilim Kurgu ve Fantastik Yapımlar
Romantik
Bilim ve Doga
Cocuk ve Aile
Drama
Gerilim
Komedi
Korku
Anime
Reality Program

Şekil.7 Ekranın sol üst tarafında bulunan kullanıcının seçebileceği türler.

Belgesel	Anime	Drama
Aydaki Son Adam	Sonic X	Yerçekimi
Bölüm Sayısı: 1	Bölüm Sayısı: 54	Bölüm Sayısı: 1
Puan: 10	Puan: 7	Puan: 8
95.0dk.	60.0dk.	88.0dk.
72 sevimli hayvan	Beyblade	Dangal
Bölüm Sayısı: 1	Bölüm Sayısı: 23	Bölüm Sayısı: 1
Puan: 10	Puan: 6	Puan: 7
95.0dk.	54.0dk.	145.0dk.

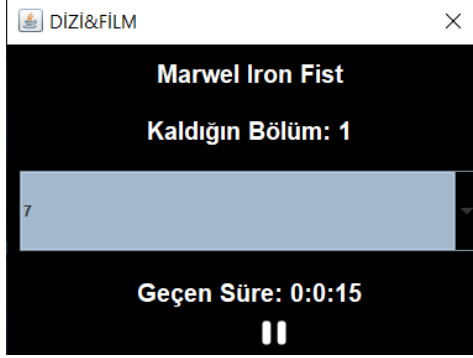
Şekil.7.1 Seçilen türe göre o türe ait tüm programların bilgileri ile birlikte listelenmesi

Kullanıcı Şekil.7.1 de görülen izle butonuna tıkladıktan sonra karşısına çıkan panelde programın ismini, hangi bölümde kaldığını, ne kadar süre izlediğini görebilir ve programa puan verebilir. (Şekil.8) Bu bilgiler veritabanında tutulur.



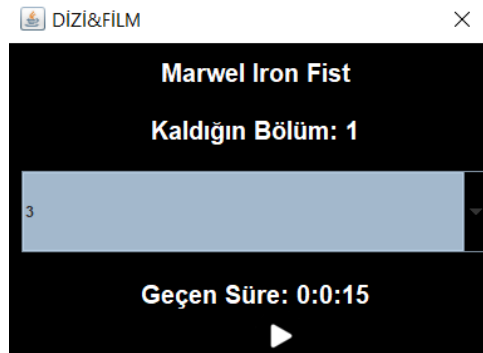
Şekil.8 Kullanıcının karşısına izlemek istediği programı seçtikten sonra çıkan panel

Kullanıcı Şekil.8 deki panelin en altında bulunan “Play” tuşuna tıklayarak programı izlemeye başlar ve izlediği süre “Geçen Süre” kısmında gösterilir. (Şekil.8.1)



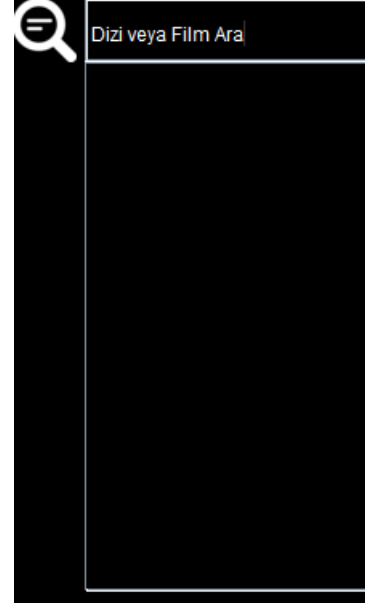
Şekil.8.1 Play tuşuna basıldıktan sonra izlendikçe artan sürenin ve verilen puanın görüntüsü.

Kullanıcı Şekil.8.1 deki “Stop” tuşuna basarak programı izlemeyi durdurabilir. Durdurduktan sonra izleme ekranını kapatsa veya uygulamanın çalışmasını durdursa dahi kaldığı süre veritabanına kaydedilir ve her defasında güncellenir. Böylelikle kullanıcı tekrar kaldığı yerden izlemeye devam edebilir. Ayrıca kullanıcı aynı programa tekrar tekrar puan verebilir ve verdiği puanlarda veritabanında güncellenir.

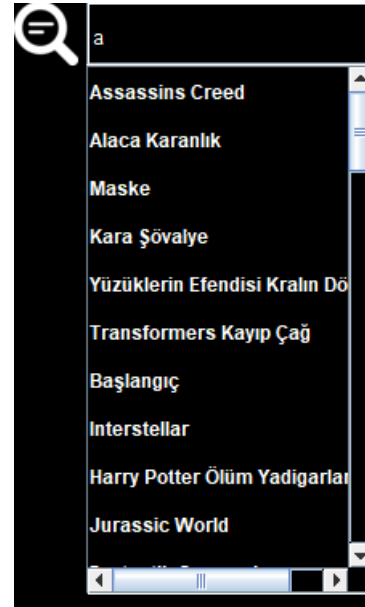


Şekil.8.2 Stop tuşuna basıldıktan sonra tekrar puan verilmesi ve kalınan süreden devam edildiğinin görüntüsü

Kullanıcı eğer isme göre arama yapmak isterse ekranın sağ üst köşesinde yer alan (Şekil.9) arama butonundan programın ismine göre arama yapabilir. Aranani dizi veya filmler kullanıcı yazdıkça yazdığı harfleri de içinde bulunduran bütün dizi, film sırası ile alt alta listeler. (Şekil.9.1)



Şekil.9 Ekranın sağ üst köşesinde isme göre arama yapılan “Dizi veya Film Ara” butonu



Şekil.9.1 İsme göre aranan dizi veya filmlerin listelenmesi

Kullanıcı aradığı programa tıkladıktan sonra karşısına Şekil.8 deki panel çıkar ve Şekil.8.1 ve Şekil.8.2 deki gösterilen gibi seçtiği programı izleyebilir.

IV. YALANCI KOD

Programlar ile türler arasındaki ilişkiyi oluşturan ve progtur tablosuna kaydeden kod parçası:

```
Main.programTurTablosuOlustur() {
    String turSorgu = "SELECT tur,pid FROM program";
    sonu k mesi1 = databaseSorgu(turSorgu);
    while(database'de program satırı bulduk a){
        sonu k mesi++;
        sonu k mesi tur ve pid al,
        birden fazla tur varsa turleri string dizisine ata;
        for(string dizisinin boyu kadar
d n,sonu k mesi2++) {
            tidSorgu = "SELECT tid FROM tur
where tname=" + "\"" + t radı + "\"";
            sonu k mesi2 = databaseSorgu(tidSorgu);
            sonu k mesi2 t rid al;
            String dbkayit = "INSERT INTO
progtur(pid,tid)"
            + "VALUES(" + pid + "," + t rid + ")";
            databaseEkle(progtur tablosuna pid ve tid);
        }
    }
}
```

Verileri Netflix DB.xlsx dosyasından  eken ve veritabanına ekleyen kod par ası:

```
Main.programTablosuOlustur(){
    Netflix DB.xlsx dosyasını a ;
    while(dosyada satır var){
        Netflix DB.xlsx dosyasından oku;
        Satırdaki programa ait verileri bellekte tut;
        Bellekteki verileri databasedeki program tablosuna
ekle;
        satır.next;
    }
}
```

Tabloların i indeki  zellikleri silen kod par ası:

```
Main.tabloSifirlama(tabloİsmi){
    veritabanında "tabloİsmi" ile e leşen tabloya ait verileri
veri tabanından sil.
}
```

Veritabanını jdbc'ye ba layan kod par ası:

```
veritabaniBaglama() {
    "netflix.db" veri tabanına ba lan;
    sorgu yapabilmek i in global statement olu tur;
    kisiSayisi= (database'de kayıtlı toplam ki i sayısını al);
    kisiSayısını global eri ilebilen de i kене ata; }
```

Kullanıcının girdi i e-postanın regex kontrol n n yapıldı ı kod par ası:

```
Login.dogrulama(String mail) {
    if(mail regexe uygun){
        return true;
    }else{
        return false;
    }
}
```

Kullanıcının hesabına giri  kontrol n n yapıldı ı kod par ası:

```
Login.girisYap() {
    String e-posta;
    String parola;
    if (do rulama(e-posta)) {
        if(e-posta e leşiyor database && parola e leşiyor
database){
            oturumA ;
            Management sayfasına y nlendir;
        }else{
            hataMesaji("E-posta veya parola hatalı");
        }
    }else {
        hataMesaji("Ge erli bir e-posta girmelisin");
    }
}
```

Kullanıcının 3'ten fazla t r se mesini engelleyen kod par ası:

```
Register.turKontrol(JCheckBox t rKutucu u) {
    if (t rKutucu u.isSelected()) {
        if (secilenTurSayisi < 3) {
            secilenTurSayisi++;
            secilenTurler ekle (t rKutucu u.getText());
        }
    } else {
        secilenTurler sil (t rKutucu u.getText());
        secilenTurSayisi--;
    }
}
```

Kullanıcının girdiği bilgileri kontrol edip veritabanına kullanıcıyı ekleyen kod parçacığı:

```
Register.kayitYap() {  
    if(tüm bilgiler girilmiş && dogrulama(e-posta) &&  
        kullanıcı veritabanında kayıtlı değil){  
        kullanıcıyı bilgileriyle birlikte veritabanına kayıt et;  
    }else{  
        hataMesajı("Girdiğiniz bilgileri kontrol ederek  
yeniden deneyin");  
    }  
}
```

Kullanıcının seçtiği türe göre öneri program çıkaran kod parçacığı:

```
Register.tureGoreProgramAtama(seçilen tür ismi,JLabel  
program1İsim, program1Süre, program1Puan,  
program1BölümSayısı, JLabel program2İsim...){  
    seçilen türe ait en yüksek puanlı 2 dizinin bilgilerini  
veritabanından al;  
    program1 in bilgileirini ata;  
    ...  
    program2 nin bilgilerini ata;  
    ...  
}
```

Veritabanından turlerin isimlerini çeken ve bunları turListModel'e ekleyen kod parçacığı:

```
Management.dbTurCek() {  
    for(veri tabanındaki tür sayısı kadar dolaş){  
        tür ismini veritabanından çek;  
        tür ismini turListModel'e ekle;  
    }  
}
```

Kullanıcının ProgramAramaText'e yazdığı stringi veritabanında program ismi olarak sorgulayan ve arama sonuçlarını aramaSonucModel'ekleyen kod parçacığı:

```
Management.dbProgAra(String progname) {  
    İçinde progname geçen tüm programları  
aramaSonucModel'e ekle;  
}
```

Kullanıcının bilgisayarının ekran değerlerine gore özellik atayan kod parçacığı:

```
ozellikAta(JPanel panel, String deger) {  
    kullanıcın ekranının boyutlarını al;  
    kullanıcının ekran boyutuna göre fontSize belirle;  
    panelin Layoutunu BorderLayout yap;  
    JLabel label = new JLabel(deger);  
    labelin fontunu, rengini, konumunu ve boyutlarını  
ata;  
    labeli panele ekle;  
}
```

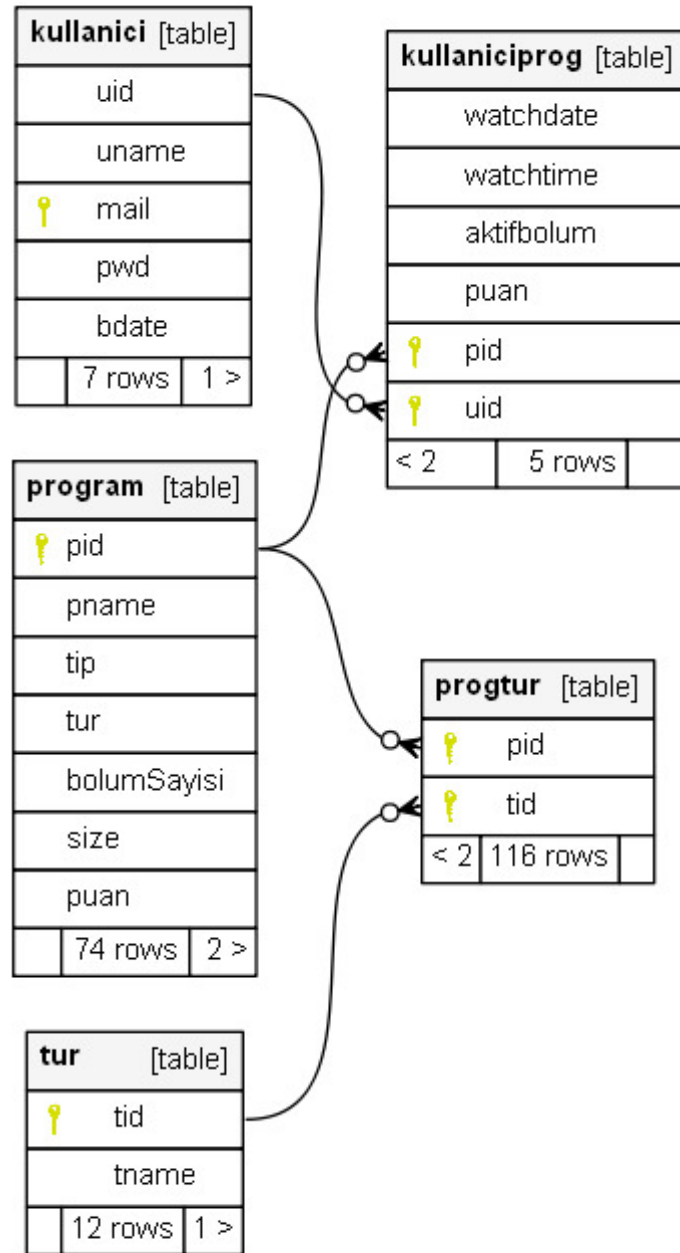
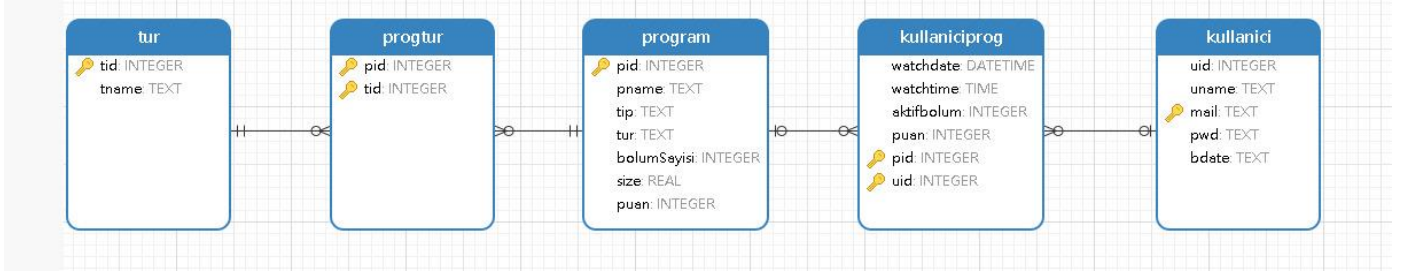
Kullanıcının turlerListde seçtiği türe göre programPanele programları yerleştirip, güncelleyen kod parçacığı:

```
Management.turProgramAlanOlustur(String turName) {  
    for(turName adlı türe ait program sayısı kadar dolaş){  
        JPanel panel = new JPanel();  
        ozellikAta(panel);  
        program bilgilerini al;  
        program bilgilerini panel'e ekle;  
        programizle butonunu oluştur;  
        programizle butonuna  
mauseClickListener ekle;  
        programizle yi panel' e ekle;  
        JPaneli programların listelendiği  
programlarPanel'e ekle  
    }  
}
```

Kullanıcının isme göre program aramasını gerçekleştiren kod parçacığı:

```
programAlanOlustur(String progName) {  
    kullanıcının arama yaptıktan sonra seçtiği  
programa(progName) ait bilgileri veritabanından al;  
    kullanıcının o programla ilişkisini veritabanından al;  
    bilgileri eşleştir;  
    kullanıcının karşısına program izleme panelini çıkart;  
}
```

VERİTABANI VARLIK İLİŞKİ DİYAGRAMI



V. SONUÇ

Projeyi geliştirirken arayüz ve veritabanı için bir çok kaynaktan araştırma yaptık. Ms Access, MySQL, MsSQL, SQLite ve Apache Derby gibi bir çok Veritabanı Yönetim Sistemlerini araştırdık. Projemizin başka bilgisayarda çalışmasını sağlamak için localhost üzerinden erişmek yerine VTYS'yi projemizin içine gömmek istedik. Bu yüzden bize en uygun olan SQLite VTYS'ini kullanmaya karar verdik.

Projede en zorlandığımız kısım arayüzü oluşturmak ve Thread kavramını kullanmak oldu. Arayüzde kullandığımız JLabel, JPanel, JList JScrollPane, JTextfield gibi Swing materyallerinin bazılarını sürükleyip bırak yaparak bazılarını kendimiz yazarak oluşturduk. Kullanıcının izleme ekranını kapattığında veya programı sonlandığında kaldığı süreden devam edebilmesi için thread sınıfından ve bu sınıfın birçok fonksiyonundan yararlandık.

Gerekli verileri bize verilen “Netflix DB.xlsx” dosyasından okuma yaparak bu verileri ve kullanıcıdan alınan bilgileri “netflix.db” isimli veritabanımıza kaydettik. Fonksiyonlarımızı, kullanıcıyı karşılayan bir gif panelini (“Karsilama.java”), kullanıcı girişi, kayıt ve yönetim panellerini(“Login.java”, “Register.java”, “Management.java”), tüm pencerelerin güncellendiği “AnaEkran.java” isimli classlarda oluşturduk. Excel dosyasından verileri okuyan, veritabanını JDBC'ye bağlayan, program-tür tablosunu oluşturan ve güncellenebilecek veriler için tabloları sıfırlayan fonksiyonları “Main.java” classında oluşturup projemizi tek bir JFrame üzerinden çalıştırarak programımızı tamamladık.

VI. KAYNAKÇA

- [1] w3Schools SQL Tutorial (n.d.). Retrieved from <https://www.w3schools.com/sql/>
- [2] Kodlab JAVA-10 Mehmet Kirazlı – Sezer Tanrıverdioğlu
- [3] Youtube Sylvain Saurel (n.d). Retrieved from <https://www.youtube.com/watch?v=nnum2jEwOUA>
- [4] Yazılım Bilimi Java programlama
- [5] ISMAIL GURSOY (n.d.). Retrieved from <https://www.ismailgursoy.com.tr/thread-sinifi-thread-class/>
- [6] Oracle (n.d.). Retrieved from <https://docs.oracle.com/javase/8/docs/api/java/time/format/DateFormatter.html>