NETFLIX

Engin Yenice Kocaeli Üniversitesi Bilgisayar Mühendisliği 190201133 enginyenice2626@gmail.com

Özet—C# dili ile Netflix platformunun benzeri geliştirilmiştir. Bu proje sayesinde veritabanı yönetim sistemleri dersinde öğrendiğimiz konular pekiştirilmiştir.

Keywords—C#,NETFLIX,veri tabanı, SQLite,çoktan-çok ilişkisi, ilişkisel veri tabanları

I. TEMEL BILGILER

Programlama Dili: C#

• Kullanılan IDE: Visual Studio 2019

Veritabanı : SQLite

• Veritaban Görüntüleme: DB Browser (SQLite)

II. ÖZET

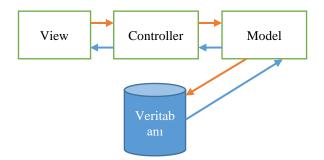
Netflix platformunun bir benzerinin masaüstü platformlara göre geliştirilmesi hedeflenmiştir. Bu proje sayesinde veri tabanı yönetim sistemlerindeki işlenmiş konular pekiştirilmiştir.

III. GIRIŞ

A. Projenin Kodlama Yapısı

Projenin kodlama yapısını tasarlarken 4 farklı katmandan oluşmasına karar verdim. Bu katmanlar: Model, View, Controller ve Datas dır.

- Model: Veri tabanı ile haberleşme işlemlerinin gerçekleşeceği katmandır.
- View: Formların bulunduğu görüntü katmanıdır.
- Controller: Model ve View arasında bulunan kontrollerin yapıldığı ara katmandır.
- Datas: Oluşturulan veri tiplerinin bulunduğu ayrı bir katmandır.



Projenin kodlamasının genel yapsı bu şekilde olmasına karar verildi.

- 1. View katmanı klasör yapısı:
 - a. AccountSetting.cs
 - b. HomePage.cs
 - c. LoadingPage.cs
 - d. LoginPage.cs
 - e. RegisterPage.cs
 - f. RegisterSelectType.cs
 - g. WatchPage.cs
- 2. Controller Katmanı
 - a. HomePageController.cs
 - b. RegisterSelectTypeController.cs
 - c. UserController.cs
 - d. WatchController.cs
- 3. Model
 - a. HomePageModel.cs
 - b. RegisterSelectTypeModel.cs
 - c. UserModel.cs
 - d. WatchModel.cs
- 4. Datas
 - a. DBPath.cs
 - b. Program.cs
 - c. Type.cs
 - d. User.cs
- B. Veritabanı ve Veritabanı Tabloları
 - 1) Veritabanı

Veritabanı olarak ilk başta MS ACCESS kullandım fakat projenin son günlerine geldikten sonra MS ACCESS başka bilgisayarlarda çalışabilmesi için ACCESS Runtime programını bulundurmasını gerektiğini bildiğim için SQLite olarak değiştirdim.

2) Veritabanı Yapısı

• Kullanıcı Tablosu

id	INTEGER (PK)
kullaniciAdi	VARCHAR(255)
kullaniciEmail	VARCHAR(255)
kullaniciParola	VARCHAR(255)
kullaniciDogumTarihi	VARCHAR(255)

• KullanıcıProgram Tablosu

izlemeTarihi	VARCHAR(255)
izlemeSuresi	INTEGER
hangiBolumdeKaldi	INTEGER
verilenPuan	INTEGER
kullaniciID	INTEGER (FK)
programID	INTEGER (FK)

• Program Tablosu

id	INTEGER (PK)
programAdi	VARCHAR(255)
programTipi	INTEGER
programBolumSayisi	INTEGER
programUzunlugu	INTEGER
toplamPuan	INTEGER

• ProgramTur Tablosu

programID	INTEGER (FK)
turID	INTEGER (FK)

IV. PROGRAM NASIL KULLANILIR.

A. Giriş Yapma Ekranı

Giriş yapmak ekranından sisteme giriş yapabilirsiniz. Kullanıcı bilgilerinizi girdikten sonra Beni Hatırla seçeneğini aktif ederseniz. Giriş bilgileriniz otomatik olarak kayıt edilecektir ve bir daha ki girişinizde bu bilgiler otomatik olarak gelecektir.

Bu ekrandan ayrıca hesap oluşturma ekranına da geçiş yapabilirsiniz veya programdan çıkış yapabilirsiniz.



Giriş ekranı

B. Hesap Oluştur Ekranı

Hesap oluşturma ekranından sisteme kayıt olabilirsiniz. Bu ekranda karşılaşabileceğiniz hatalar:

Hesap Kayıt Edildi. Hesap Kayıt Edilemedi. Böyle Bir Eposta Sisteme Kayıtlı

Şeklinde uyarı mesajları bulunmaktadır.



Hesap oluştur



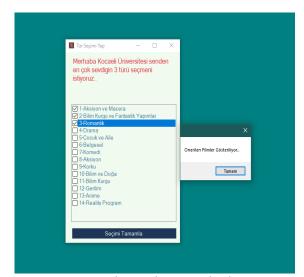
Hesap oluştur(kayıt tamamlandı)

C. Yeni Kayıt Olan Kullanıcı İçin Tür Seçme Ekranı

Yeni hesap oluşturan kullanıcıların sisteme giriş yapmadan önce ilk karşılaştıkları ekran burası olmaktadır. Bu ekranda yeni kullanıcılar için önerilen listesi oluşturulmasına yardımcı olmaktadır.



Tür seçimi yap



Tür seçim. Seçim tamamlandı

D. Program Yükleniyor Ekranı

Bu ekranda program yükleniyormuş hissi veren bir ekran karşılamaktadır. Bu ekranın süresi 2 saniyedir.



Program yükleniyor

E. Önerilenler Ekranı

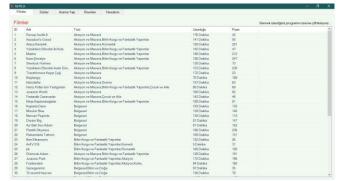
Yeni hesap oluşturan kullanıcıların türleri seçtikten sonra karşılaştığı ekrandır. Kullanıcının seçtiği türler metin olarak sol üst köşede yazmaktadır. Alt tarafta ise önerilen programlar listelenmektedir. Bu ekran sadece yeni kullanıcılara gösterilmektedir. Programları izlemek için izlenmesini izlediğiniz programın üzerine çift tıklamanız gereklidir.



Önerilenler

F. Filmler Ekranı

Tüm filmlerin listelendiği ekrandır. Filmleri izlemek için izlenmesini izlediğiniz filmin üzerine çift tıklamanız gereklidir.



Filmler

G. Diziler Ekranı

Tüm dizilerin listelendiği ekrandır. Dizileri izlemek için izlenmesini izlediğiniz dizinin üzerine çift tıklamanız gereklidir.



Diziler

H. Arama Yap

Tüm dizilerin ve filmlerin listelendiği ekrandır. Bu ekranda: Türe göre arama

Program adına göre arama

Tüm programların listelenmesi (boş arama yapılırsa)

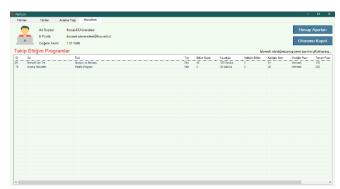
Tür ve program adı beraber arama



Arama

İ. Hesabım

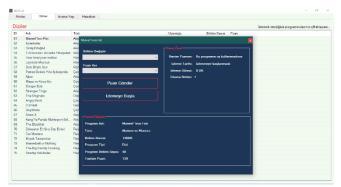
Kullanıcının takip ettiği programların listelendiği ve hesap ayarlarını düzenleyebildiği ekrandır.



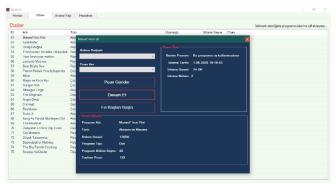
Hesabım

J. WatchPage – Program İzleme Ekranı

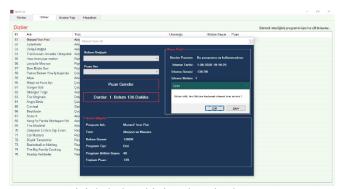
Program izleme, program izlerken bölüm değiştirme, programa yeniden başlama ve puan verme özelliklerinin bulunduğu ekrandır.



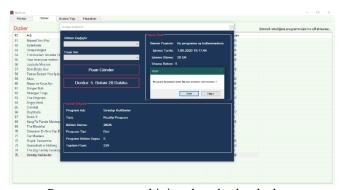
Program hiç izlenmemişse



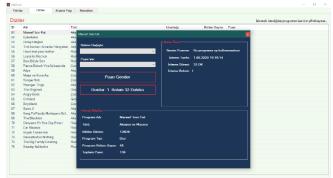
Program izlenmeye başlamış ama tamamlanmamışsa



Dizi de bölüm bittiyse karşılaşılan uyarı



Program tamamen bittiyse karşılaşılacak ekran



Program izlenme anı

K. Hesap Ayarları

Bu ekrandan hesabınızın Adını soyadını, şifresini ve doğum tarihini değiştirebilirsiniz.



Hesap Ayarları

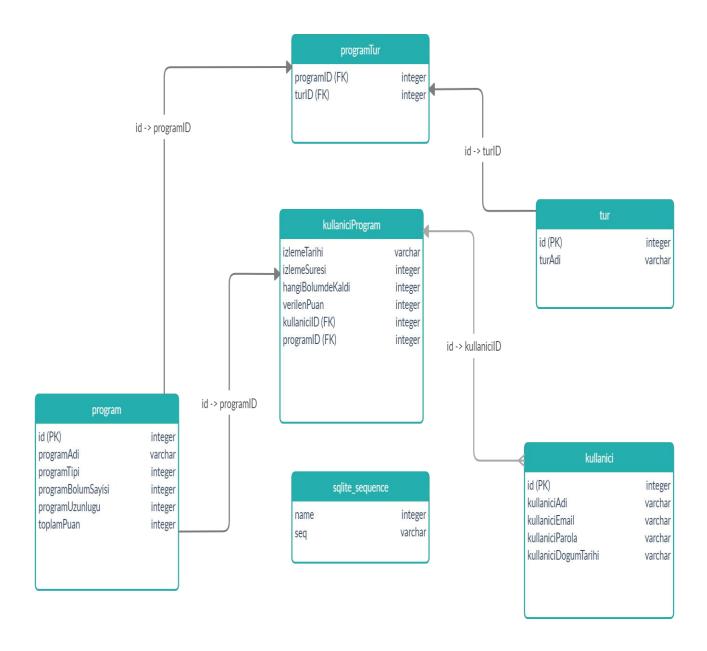
L. Programın çalışması için exe dosyasının yanında hangi dosyalar bulunmalı?

Programın çalışabilmesi için birkaç dosyanın exe dosyasının yanında olması gerekmektedir bunlar:

- 1. Assets klasörü
- 2. X64 klasörü
- 3. X86 klasörü
- 4. System.Data.SQLite.dll

Dosyaları kesinlikle bulunmalıdır.

V. VERİTABANI, VARLIK İLİŞKİ DİYAGRAMI



VI. DENEYSEL SONUÇLAR

Projenin yapım aşamalarında ACCESS Veritabanı ile işlemlerimi sürdürüyordum. Bu süre zarfında Access veri tabanının performansından memnun değildim. Bunun yanında veri tabanının taşınabilir olması gerekiyordu ve Access veri tabanı bilgisayarda Runtime kurulumu olmadan çalışmıyordu ve bu uygulama her bilgisayarda kurulmamış olabiliyordu.

Bu problemlerin değerlendirilmesi sonucunda Veritabanı sistemini değiştirmeye karar verdim. Hem hızlı hem taşınabilir olması gereken bir veri tabanı arıyordum ve SQLite bu konuda çok başarılı olduğunu fark ettim. Fakat SQLite da ise tarih işlemleri için değişken tanımlaması yapılmıyordu. Uygulama üzerindeki tarih verilerini veri tabanına text tipinde göndermek için düzenlemeler projeyi başarılı bir şekilde SQLite veri tabanına geçirdim. Burada kullandığım MVC yapısını doğru kullanmam sonucunda bu geçiş işlemi çok hızlı gerçekleşti.

VII. KABAKOD

Proje teslim dosyaları içerisinde yer alan 1_KabaKod_190201133.pdf dosyasının içerisinde yer almaktadır.

VIII. SONUÇLAR

Bu proje sayesinde:

- 1. MVC yapısının temelini öğrendim
- Bir veritabanından başka bir veritabanına geçişin nasıl sağlandığını öğrendim.
- 3. Foreign key ve Primary key değerlerinin ne anlama geldiğini öğrendim.
- 4. SQL sorguları yazmayı öğrendim.
- 5. Uygulama ile Veritabanı arasında iletişim kurmayı öğrendim.

IX. KAYNAKLAR

- https://www.dbsofts.com/articles/ms_access_to_sqlite/
 Access_deki
 veritabanını sqlite aktarmak için kullanıldı.
- [2] https://www.w3schools.com/sql/sql_select.asp SQL Select
- $[3] \quad \underline{https://www.w3schools.com/sql/sql_update.asp} \ SQL \ Update$
- [4] https://www.w3schools.com/sql/sql_insert.asp SQL Insert
- [5] https://www.w3schools.com/sql/sql_delete.asp SQL Delete
- [6] https://coolors.co/ Programdaki renk seçimlerini karar vermek için