

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programlama Laboratuvar II Proje III

NETFLIX

Ramazan Kaan YARAYAN 190201138

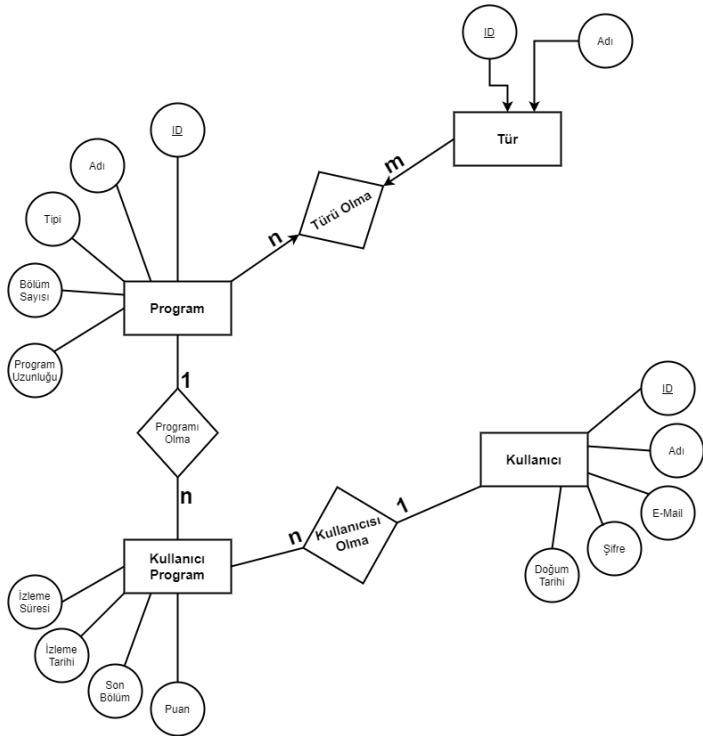
Kocaeli, Türkiye
rknyryn@gmail.com

Özet—Yerel bir veri tabanına bağlı olarak oluşturulan Netflix platformudur.

I. GİRİŞ

Son kullanıcının kullanımına yönelik geliştirilen Netflix platformu. Access uygulaması kullanılarak oluşturulan bir veri tabanına bağlı olup, C# programlama dili kullanılarak geliştirilen bir grafik arayüzü ile kullanıcıların kayıt olup, içerik arayabileceği ve bu içerikleri izleyebileceği bir platformdur.

II. VARLIK İLİŞKİ DİYAGRAMI

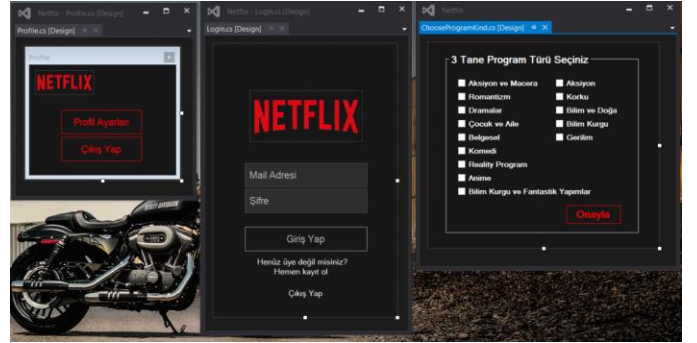


III. YÖNTEM

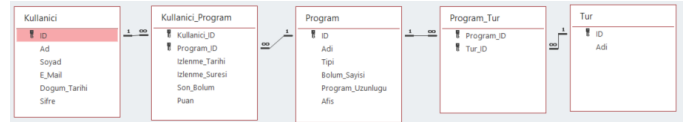
A. Uygulama Analizi ve Veri Tabanı Oluşturulması

Uygulamanın kullanıcının kolaylıkla anlayıp , kullanabileceği bir arayüzü geliştirilmesi ve uygulama için kullanılacak verilerin depolanması için bir veri tabanı oluşturulması ile başladı.

Öncelikle tüm form tasarımları oluşturuldu (Örnek: Şekil1.0) ve ardından kullanılacak veri tabanı oluşturuldu (Örnek: Şekil1.1).



(Şekil:1.0)



(Şekil:1.1)

B. FileOperation Sınıfı

FileOperation sınıfı; kullanıcı uygulamaya giriş yaptıktan sonra giriş yapan kullanıcıyı uygulamanın klasörü içerisine NetflixSet klasörü altına control.txt dosyası oluşturarak bu txt dosyasının içerisine kayıt ediyor. Bu şekilde kullanıcı uygulamayı tekrar başlattığında giriş işlemi yapmasına gerek

kalmadan uygulama erişimi sağlanmış oluyor.

Eğer kullanıcı uygulamadan çıkış yapar ise; control.txt dosyası siliniyor ve tekrar giriş yaptığında kullanıcı giriş sayfasına yönlendiriliyor.

Bu işlemleri gerçekleştirmek için System.IO kütüphanesi eklenerek; FileStream, StreamReader ve StreamWriter sınıflarından türetilen nesneler kullanıldı(Örnek: Şekil1.2).

```
class FileOperation
{
    private static FileStream file;
    private static StreamReader reader;
    private static StreamWriter writer;
    public static int userID;

    public string openingControl()
    public void openingSave(string eMail)
    public static void saveUserID(int ID)
    public static void closeAccount()
}
```

(Şekil1.2)

FileStream sınıfından türetilen nesne ile dosyanın nerde olduğu ve yapılacak işlem (Okuma-Yazma) belirtildi ve reader nesnesi ile okuma işlemi, writer nesnesi ile yazma işlemi gerçekleştirildi.

C. Account Sınıfı

```
class Account
{
    public string firstName;
    public string lastName;
    public string eMail;
    public string date;
    public string password;
}
```

Veri tabanına kayıt edilen kullanıcılar alınırken verilerin tutulması için oluşturulmuştur.

D. Programs Sınıfı

```
class Programs
{
    public int id;
    public string name;
    public string programType;
    public List<string> programKind = new List<string>();
    public int numberOfEpisodes;
    public float programLength;
    public string bannerUrl;
    public double puan;
}
```

Veri tabanından alınan program verilerinin tutulması için oluşturulmuştur.

E. DatabaseOperations Sınıfı

Veri tabanı işlemlerini gerçekleştirmek için oluşturulmuştur.

Access bağlantısı için gerekli olan "System.Data.OleDb" eklenmiştir. Veri tabanına bağlanmak için bir OleDbConnection sınıfından nesne(connection) türetilmiştir. Bağlantı kurulduktan sonra sorgu işlemleri için OleDbCommand sınıfından da bir nesne(command) türetilmiştir. İşletilen sorgu sonucunda verileri alabilmek için; OleDbDataReader sınıfından bir nesne (reader) türetilmiştir. Böylelikle bu nesneleri kullanarak veri tabanı işlemleri gerçekleştirilmiştir.

DatabaseOperation sınıfının kurucu metodu içerisinde, connection için bağlantı yolu belirtilmiştir.

```
public DatabaseOperation()
{
    connection = new OleDbConnection(@"Provider=Microsoft.ACE.OleDB.12.0; Data Source=NetflixBelirliVeriTabani.accdb");
    command = new OleDbCommand();
}
```

1) public bool passwordControl(string password, string password1) Fonksiyonu

Parametre olarak gönderilen değerlerin aynı olup olmadığının kontrolünü yapıyor ve eğer aynı iseler geriye "true" değer farklı iseler "false" değer döndürüyor.

2) public void AddAccount() Fonksiyonu

Veri tabanına kullanıcı eklemek için oluşturulan bu fonksiyon içerisinde;

- Veri tabanı bağlantısını aç
- Yapılması istenilen sorguyu tanımla (insert into [tablo_adi] ([alan1],[alan2],...) values ([değer1],[değer2],...))
- Sorgunun hangi bağlantıyı kullanarak çalışacağını ata
- Sorguyu işle
- Kullanıcıya bilgilendirme mesajı gönder
- Veri tabanı bağlantısını kapat

İşlemleri gerçekleştirilerek veri tabanına kullanıcı ekleniyor.

3) public bool LoginControl(eMail, password) Fonksiyonu

Kullanıcı uygulamaya giriş yapmak istediğinde e-mail adresi ve şifresini kontrol edip doğru ise "true" yanlış ise "false" değeri döndüren fonksiyon.

Fonksiyonun işleyişi:

- Veri tabanı bağlantısını aç
- Yapılması istenilen sorguyu tanımla (select Sifre from Kullanici where E-Mail = eMail)
- Sorgunun hangi bağlantıyı kullanarak çalışacağını ata

- Sorguyu çalıştır ve gelen veriyi oku(*reader*)
- Eğer gelen veri = password ise; veri tabanı bağlantısını kapat ve true değer döndür
- Eğer gelen veri != password ise; veri tabanı bağlantısını kapat ve false değer döndür

4) *public Account getAccountData(eMail) Fonksiyonu*

Kullanıcı kendi bilgilerinde değişiklik yapmak isterse veri tabanından verilerinin alınması için oluşturulmuştur.

Fonksiyonun işleyişi:

- Veri tabanı bağlantısını aç
- Yapılması istenilen sorguyu tanımla (*select * from Kullanici where E_Mail = eMail*)
- Sorgunun hangi bağlantıyı kullanarak çalışacağını ata
- Sorguyu çalıştır ve gelen veriyi oku(*reader*)
- Bir Account nesnesi oluştur
- Account nesnesine gelen değerleri ata
- Veri tabanı bağlantısını kapat
- Geri dönüş değeri olarak account nesnesini gönder

5) *public void UpdateAccount(eMail) Fonksiyonu*

Kullanıcı kendi bilgilerinde değişiklik yaptıktan sonra bilgileri veri tabanına kaydetmek için oluşturulmuş fonksiyon.

Fonksiyonun işleyişi:

- Veri tabanı bağlantısını aç
- Yapılması istenilen sorguyu tanımla (*update sorgusu*)
- Sorgunun hangi bağlantıyı kullanarak çalışacağını ata
- Sorguyu çalıştır
- Kullanıcıya bilgilendirme mesajı gönder
- Veri tabanı bağlantısını kapat

6) *public void getPrograms(type) Fonksiyonu*

Kullanıcı uygulamaya giriş yaptıktan sonra karşısına gelecek olan içeriklerin veri tabanından alınması için oluşturulmuştur. type parametresi ile içeriğin tipi(*Dizi, Film, Tv Show*) alınarak içeriği tipine göre veri tabanından verileri alıyor.

Fonksiyonun işleyişi:

- Veri tabanı bağlantısını aç
- Önceki listeyi temizle(*programs.Clear()*)
- Yapılması istenilen sorguyu tanımla(*select * from Program where tipi = type*)
- Sorgunun hangi bağlantıyı kullanarak çalışacağını ata

- Sorguyu çalıştır ve gelen veriyi oku(*reader*)
- Programs nesnesi oluştur
- Programs nesnesine okunan verileri ata
- programs listesine ekle
- reader okuduğu sürece işlemleri tekrarla
- Veri tabanı bağlantısını kapat

7) *public void getMyList(userID) Fonksiyonu*

Kullanıcının daha önce izlediği dizi, film ve tv showlarını veri tabanından alır.

Fonksiyonun işleyişi:

- Veri tabanı bağlantısını aç
- Önceki listeyi temizle(*programs.Clear()*)
- Kullanici_Program tablosunda Kullanici_ID = userID olan programların bilgilerini al
- Tüm verileri programs listesine ekle
- Veri tabanı bağlantısını kapat

8) *public void Search(search) Fonksiyonu*

Kullanıcının aradığı içerik(*içerik adı, içerik türü, içerik tipi*) ile eşleşen verilerin veri tabanından alınması için oluşturulan fonksiyon.

Fonksiyonun işleyişi:

- Veri tabanı bağlantısını aç
- Önceki listeyi temizle(*programs.Clear()*)
- Programın adı search'e benzer olanları al
- Alınan verileri programs listesine ekle
- Programın tipi search'e benzer olanları al
- Alınan verileri programs listesine ekle
- Programın türü search'e benzer olanları al
- Alınan verileri programs listesine ekle
- Veri tabanı bağlantısını kapat

9) *public void AddList() Fonksiyonu*

Kullanıcının izlediği içeriği, izlediği içerikler arasına(*Kullanici_Program tablosu*) eklenmesi için oluşturulmuş fonksiyon.

Fonksiyon işleyişi:

- Günün tarihini al
- Veri tabanı bağlantısını aç
- Kullanici_Program tablosuna izlenen içeriği ekle
- Veri tabanı bağlantısını kapat

10) *public void AddList() Fonksiyonu*

Uygulamaya kayıt olunurken kullanıcıya gösterilecek olunan en çok puan alınan filmleri alan fonksiyon.

Fonksiyonun işleyişi:

- Veri tabanı bağlantısını aç
- Önceki listeyi temizle(*programs.Clear()*)
- Kullanıcının seçtiği 3 türün, en çok puan alınan film içeriklerini al

- Parametre olarak gönderilen içeriğin verilerini bilgi gösterilecek sayfanın(*pnl_info*) görünürlüğünü açıp içeriğin verilerini ekrana basıyor.

4) Watch(programs) Fonksiyonu

Parametre olarak aldığı içerik eğer filmse izleme sayfasına(*pnl_player*) yönlendiriyor, eğer film değil ise bölüm sayfasına(*pnl_episodes*) yönlendiriyor. Böylelikle kullanıcı istediği bölümü seçip izleyebiliyor.

IV. DENEYSEL SONUÇLAR

Kullanıcı giriş kontrolü yapılırken, en son giriş yapan kullanıcı “.txt” dosyası ile tutuluyordu. Bunun ayrıca sebebi veri tabanının oluşturulmasının uygulama içerisinde gerçekleştirilmesinin planlanmasıdır. Bu sayede hem giriş yapan kullanıcıyı hem de veri tabanının olup olmadığı kontrol edilmesi planlanmıştır ancak veri tabanının proje içerisine yerleştirilmesi ile bu işlemden vazgeçilmiştir.

Dosya ile kontrol işlemi yerine ”Kullanici” tablosunda bu amaç için kullanılmak üzere bir alan oluşturulup giriş yapan kullanıcının kontrolü buradan gerçekleştirilebilirdi. Bu sayede dosya işlemleri ile uğraşılmaya gerek duyulmazdı.

V. SONUÇLAR

- Veri tabanı kullanımı ve amacı hakkında bilgi edinildi.

- C# dili kullanılarak bir veri tabanına bağlanma ve ordaki verilerle işlem yapılması hakkında bilgi sahibi olundu.

KAYNAKLAR

Access Bağlantısı ve İşlemleri

- <https://www.yazilimkodlama.com/programlama/c-access-veri-tabani-baglantisi-select-insert-update-delete-ornek-uygulama/>
- <https://www.youtube.com/watch?v=TxVNMTfOWgc>
- <http://www.ibrahimbayraktar.net/2014/08/sqlde-unique-kullanm.html>

Key Down Özelliği

- <https://seminercikiz.wordpress.com/2015/06/24/cd-a-textboxda-entera-basınca-tetikleme/>

E-Mail Kontrol

- <https://mustafabukulmez.com/2018/07/18/c-e-mail-kontrolu-net-mail/>