

## Proje Tanımı

Ediz Teker 211002041

Sinema Bilet Sistemi, kullanıcıların sinema biletlerini satın almalarını, koltuk seçimlerini yapmalarını ve bilet bilgilerini e-posta ile alabilmelerini sağlayan bir uygulamadır. Bu sistem, kullanıcı dostu bir arayüze sahip olup, çeşitli filmleri listeleyerek izleyicilerin seans bilgilerini görüntülemelerine olanak tanır.

## Veritabanı Tasarımı

Proje, bir Access veritabanı kullanarak aşağıdaki üç tabloyu içerir:

### 1. Filmler Tablosu

Tablo Adı: Filmler

Alanlar:

Kimlik: Sayı, Film'in benzersiz tanımlayıcısı.

FilmAdi: Kısa Metin, Filmin adı.

FilmTuru: Kısa Metin, Filmin türü. Projenin bu versiyonunda kullanılmamıştır.

SeansZamani: Kısa Metin, Seans zamanı.

FilmSuresi: Kısa Metin, Filmin süresi.

### 2. KoltukDurumları Tablosu

Tablo Adı: KoltukDurumlari

Alanlar:

Kimlik: Otomatik Sayı, Koltuk durumunun benzersiz tanımlayıcısı.

FilmID: Sayı, İlgili filmin tanımlayıcısı.

Tarih: Tarih/Saat, Seans tarihi.

KoltukNumarasi: Sayı, Koltuk numarası.

MusteriID: Kısa Metin, İlgili müşteri kimliği.

### 3. Müşteri Tablosu

Tablo Adı: Musteri

Alanlar:

TcKimlik: Kısa Metin, Müşterinin TC kimlik numarası.

MusteriAdiSoyadi: Kısa Metin, Müşterinin adı ve soyadı.

MusteriEmail: Kısa Metin, Müşterinin e-posta adresi.

## Projenin Yapısı

Proje katmanlı mimariye göre yapılmıştır. Entities katmanı dışında, Data katmanı sadece Business katmanı ile, Business katmanı ise sadece Presentation katmanı ile ilişkili olmuştur. Bu katmandaki fonksiyonlar ve işlevleri aşağıda verilmiştir:

### 1. DataAccess Layer

Data katmanında veritabanına OLEDB ile bağlantı kurularak veri çekilip yazılmıştır.

-List<Film> GetFilmler(): Bu public fonksiyon, veritabanından bütün film bilgilerini (film sınıfı olarak) çeker ve filmler diye bir listeye koyar.

-List<KoltukDurumu> GetKoltuklar(): Bu public fonksiyon, veritabanından bütün koltuk durumlarını çekip, filmid'leri ile beraber bir listeye koyar.

-bool CheckIfMusteriExists(string tcKimlik): Bu public fonksiyon çağırılırken verilen tc kimlik numarasını alıp veritabanında olup olmadığına bakar. Yoksa false döndürür.

-void AddMusteri(string tcKimlik, string musterAdi, string musterEposta): Bu public fonksiyon CheckIfMusteriExists fonksiyonunu çağırıp kontrol sağlar. Eğer müşteri varsa bu fonksiyon atlanır. Yok ise girilen bilgilerle veritabanına bir müşteri eklenir.

-void AddKoltukDurumu(int filmID, DateOnly tarih, int koltukNumarasi, string musterID): Bu public fonksiyonda girilen verilerle, veritabanına bir koltuk durumu eklenir.

-void DeleteFilm(string filmName): bu public fonksiyon seçilmiş olan filmi veritabanından siler.

-void UpdateFilm(string oldFilmName, string newFilmName): Bu public fonksiyon veritabanındaki bir filmin ismini günceller.

## 2. Business Layer

Business katmanında data katmanındaki fonksiyonlar çağırılmıştır.(Presentation katmanına iletilsin diye) Bunlar dışında veritabanı ile alakalı olmayan işlemler burada yapılmıştır. (data katmanından direkt çağırılan fonksiyonlar aşağıda verilmemiştir.)

-bool CheckTcKimlik(string tcKimlik): Bu public fonksiyon tc kimliğin 11 haneden oluşup oluşmadığına ve tamamı sayılardan oluşup oluşmadığına bakar. Buna göre true veya false döndürür.

-int[] GetKoltukDurumlari(Film film, DateOnly tarih): Bu fonksiyon GetKoltuklar() fonksiyonunu çağırıp istenilen film ve tarihi filtreleyerek “dolukoltuklar” dizisi döndürür.

-static string CreatePdf(string tcKimlik, string musteriAdi, string musteriEposta, string filmAdi, string koltukNumarasi, DateOnly tarih, string zaman): Bu public fonksiyon iText kütüphanesini kullanarak verilen bilgiler ile bir pdf oluşturup programın olduğu bilgisayarın masaüstüne koyar.

-void SendEmailWithAttachment(string toEmail, string subject, string bodyText, string attachmentPath): Bu public fonksiyon MailKit kütüphanesini kullanarak pdf dosyasının veri yolunu alarak istenilen kişiye bu dosya ile bir e-posta atar.

## 3. Entities Layer

Bu katmanda her katmanın kullandığı film, müşteri ve koltuk durumu sınıfları oluşturulmuştur. Bu sınıflarda tablolarındaki nesneler mevcuttur.

-Film: Film bilgilerini tutan sınıf.

-Musteri: Müşteri bilgilerini tutan sınıf.

-KoltukDurumu: Koltuk bilgilerini ve ilişkili filmi tutan sınıf.

## 4. Presentation Layer

Projenin kullanıcı arayüzü, Windows Forms kullanılarak geliştirilmiştir.

-Form1: Filmleri listeleyen ana ekran.

-Form2: Seçilen filmin koltuk durumunu gösteren form.

-Form3: Bilet satın alma işlemleri için kullanıcı bilgilerini girdiği form.

-Form4: Admin paneli; film silme ve güncelleme işlemleri için kullanılır.

## İş Akışı

Film Seçimi: Kullanıcı ana ekrandan filmlerden birini seçer.

Koltuk Seçimi: Seçilen film için uygun koltukları görüntüler.

Bilgi Girişi: Müşteri adı, soyadı, TC kimlik numarası ve e-posta adresi bilgileri istenir.

PDF ve E-posta Gönderimi: Bilet bilgileri PDF formatında oluşturulur ve kullanıcının e-posta adresine gönderilir.

Admin İşlemleri: Admin yetkisine sahip kullanıcılar film ekleyebilir, güncelleyebilir veya silebilir.

## Kullanım Senaryoları

Müşteri Senaryosu: Bir kullanıcı, tek sinema salonu olan bir yerde bu programın açık olduğu bir ekrandan film seçer, uygun koltukları görüntüler ve bilet satın alma işlemini gerçekleştirir.

Admin Senaryosu: Admin veritabanından film ekleyebilir, program üzerinden mevcut filmleri güncelleyebilir ve silme işlemleri gerçekleştirebilir.

## Kullanılan Kütüphaneler

-iText 9.0.0

-iText bouncy castle adapter 9.0.0

-MailKit 4.9.0

-OleDb 9.0.0

## Sonuç

Sinema Bilet Sistemi, kullanıcıların sinema biletlerini yönetmelerini ve satın almalarını kolay hale getiren bir uygulamadır. Proje, katmanlı mimari ile veritabanı etkileşimleri, kullanıcı arayüzü geliştirme ve PDF/e-posta işlemleri gibi temel yazılım geliştirme becerilerini içermektedir.

## Ek Bilgiler

Projenin kaynak kodu, kullanışlı yorumlar ile birlikte iyi yapılandırılmıştır. Geliştiriciler, sistemin işleyişini anlamak ve özelleştirmek için kod üzerinde çalışmaya devam edebilirler.