

# KUTUPHANE OTOMASYONU PROJE RAPORU

29.08.2024

## PROJENİN AMACI VE FAYDALARI

### 1. Proje Amacı

Bu proje, kütüphane personelleri için işlevsel ve kullanıcı dostu bir web uygulaması geliştirmeyi amaçlamaktadır. Projenin temel hedefi, kütüphane içinde bulunan kitapların, kitapları emanet alan öğrencilerin, kitapların yazarlarının bilgilerinin ve emanet işlemlerinin veritabanından alınarak arayüzde kolayca yönetilebilmesini sağlamaktır. Bu uygulama, kütüphane personellerinin kitap ve kullanıcı bilgilerini etkin bir şekilde takip etmelerine, emanet işlemlerini hızlı ve güvenli bir şekilde gerçekleştirmelerine olanak tanır.

### 2. Proje Kapsamı

Bu proje, kütüphane yönetim süreçlerini dijitalleştirerek, kütüphane personellerinin iş yükünü hafifletmek ve operasyonları daha verimli hale getirmek amacıyla geliştirilmiştir. Proje kapsamı aşağıdaki ana bileşenleri içermektedir:

- Kitap Yönetimi:
  - Kütüphanede mevcut olan kitapların veri tabanında saklanması. Kütüphaneden çıkarılan kitapların silinmesi. Kütüphaneye yeni kitap eklenmesi.
- Yazar Yönetimi:
  - Kütüphanedeki kitapların yazarlarının bilgilerinin veri tabanında saklanması. Yeni kitap ekleneceği zaman yazar seçimi. Veri tabanına yazar eklenmesi. Kitabı bulunmayan yazarın silinmesi.
- Kitap Türü Yönetimi:
  - Kütüphanedeki kitapların türlerinin bilgilerinin veri tabanında saklanması. Yeni kitap ekleneceği zaman tür seçimi. Veri tabanına tür eklenmesi.
- Öğrenci Yönetimi:
  - Kütüphaneye üye öğrencilerin veri tabanında saklanması. Yeni öğrenci eklenmesi. Öğrencilerin kitap emaneti işlemlerinin saklanması. Öğrenci silme.
- Emanet Yönetimi:
  - Kütüphaneden öğrencilere emanet edilen kitapların teslim durumu, teslim tarihi. Teslim alan Öğrenci ve teslim edilen kitap gibi bilgilerin veri tabanında saklanması. Arayüzde geciken teslim gün sayısı gibi bilgilerin sunumu. Kitap teslim alma, teslim etme gibi işlemlerin gerçekleştirilmesi.
- Personel Yönetimi:
  - Personellerin isim, soyadı, mail adresi, yeki durumu gibi bilgilerin veri tabanında saklanması. Admin iznli personellerin personel eklemesi, güncellemesi, silmesi ve yetkilendirmesi.

#### 7. Login Yönetimi:

- Personellerin şifre ve mail adresi ile uygulamaya giriş/çıkış işlemleri.

### 3. Projede Kullanılan Metod Ve Yazılım Dilleri

Bu projede, katmanlı mimari yaklaşımı benimsenmiş olup, uygulamanın sürdürülebilirliğini, ölçeklenebilirliğini ve bakım kolaylığını sağlamak amacıyla çeşitli yazılım dilleri ve teknolojiler kullanılmıştır.

Yeni sistem şunları içermelidir:

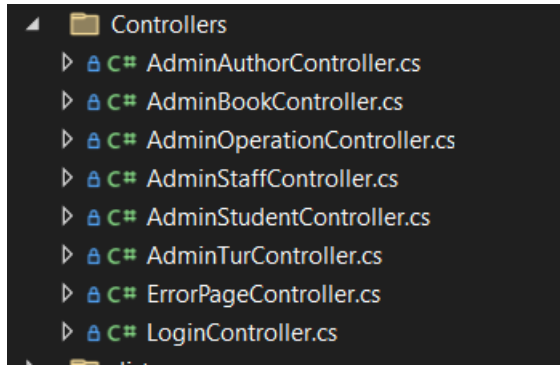
- ASP.NET MVC (Model-View-Controller)
- C#
- Entity Framework
- MsSQL
- Fluent Validation
- Authentication ve Authorization
- Bootstrap HTML/CSS
- JavaScript

### 4. Proje Katmanları

#### 1. Presentation Layer ( UI, Sunum Katmanı ):

Bu katman, kullanıcı arayüzünü oluşturan ve kullanıcı ile etkileşimi sağlayan katmandır. Bu katmanda MVC (Model View Controller) yapısı kullanılmıştır. Proje yapısı 8 controllerdan oluşmaktadır.

Controller:



##### 1. AdminAuthorController:

- Bu controllerda Yazar ekleme, silme ve güncelleme işlemleri yapılmaktadır. 3 adet view i çalıştırır. Bunlar; AdminAuthor / Index, AddAuthor, UpdateAuthor.

##### 2. AdminBookController:

- Bu controllerda Kitap ekleme, silme ,güncelleme ve yazara göre listeleme işlemleri yapılmaktadır. 4 adet view i çalıştırır. Bunlar; AdminBook / Index, AddBook, UpdateBook, AuthorBook.

3. AdminOperationController:

- Bu controllerda emanet kitap verme, alma, teslim tarihi geçenleri listeleme, teslim tarihi yaklaşan, giriş yapan personele göre listeleme gibi işlemler bulunmaktadır. 7 adet view i çalıştırır. Bunlar; AdminOperation/Index, AddOperation, MyTotalOperation, NoReturnlOperation, StudentOperation, TotalOperations, WaitOperation .

4. AdminStaffOperation:

- Bu controllerda Personel işlemleri yer almaktadır. Bu Controllerdaki methodlara, personeller arasında yalnızca Admin yetkisine sahip personeller programa giriş yaptığıında kullanabilmektedir. Personel listeleme, ekleme, silme, güncelleme gibi işlemler yer almaktadır.
- Admin yetkisine sahip olup olmadığı controller içerisinde tanımlı IsAdmin methodu ile kontrol edilmektedir.
- 3 adet viewi çalıştırır. Bunlar; AdminStaff/Index, AddStaff, UpdateStaff.

5. AdminStudentController:

- Bu controller da; Öğrenci listeleme, ekleme, silme ve güncelleme işlemleri yapılmaktadır.
- 3 adet Viewi çalıştırır. Bunlar; AdminStudent/Index, AddStudent, UpdateStudent.
- AddStudent ve UpdateStudent sayfaları içerisinde bulunan cinsiyet seçimi için DropDownList kullanılmıştır. DropDownList içersine SelectListItem ile veri gönderilmiştir. Bu verileri oluşturmak için SetGenderViewbag adında bir method eklenmiştir. Veri örneği ;

```
{Text="Kadın",Value="false"},  
{Text="Erkek",Value="true"}
```

6. AdminTurController:

- Bu controller da; Kitapların türlerinin kontrolü sağlanmaktadır. Tür ekleme, silme güncelleme ve türe göre kitapları listeleme işlemleri yapmaktadır.
- 4 adet viewi çalıştırır. Bunlar; AdminTur/Index, AddTur, UpdateTur, GenreBook.

7. ErrorPageController:

- Bu controller da; Hata yönetimi yapılmaktadır. Program içerisindeki hata türüne göre bu controllerdaki methodu çalıştırarak hata viewlerinin yani hata sayfalarının açılmasını sağlar.
- 2 adet viewi kontrol eder. Bunlar; Page403, Page404.

8. LoginController:

- Bu controller da; Personellerin uygulamaya giriş ve çıkış işlemleri yapılmaktadır.
- Kullanıcının mail ve şifresini veri tabanından kontrol eder ve programın diğer sayfalarının çalıştırılması için gerekli işlemleri yapar.
- Personel girişi olmadan çalışan tek controllerler.
- Tek viewi çalıştırır, o da giriş sayfasıdır (login)

## 2. Entity Layer ( Varlık Katmanı ):

Bu katman, veri tabanındaki nesneleri temsil eden sınıfları içerir.

Sınıflar:

```
▷ Author.cs
▷ Book.cs
▷ Genre.cs
▷ Operation.cs
▷ Staff.cs
▷ Student.cs
```

1. Author.cs : Yazar tablosunu oluşturur.
2. Book.cs : Kitap tablosunu oluşturur.
3. Genre.cs : Tür tablosunu oluşturur.
4. Operation.cs : İşlemler tablosunu oluşturur.
5. Staff.cs : Yazar tablosunu oluşturur.
6. Student.cs : Öğrenciler tablosunu oluşturur.

## 3. Data Access Layer (Veri Sunum Katmanı):

Bu katman, veritabanıyla etkileşimi sağlayan sınıfları ve işlemleri içerir. CRUD (Create, Read, Update, Delete) işlemlerinin gerçekleştirildiği katmandır. Bu katmandaki sınıflar, Entity Framework gibi ORM araçları kullanılarak veritabanı işlemlerini gerçekleştirir.

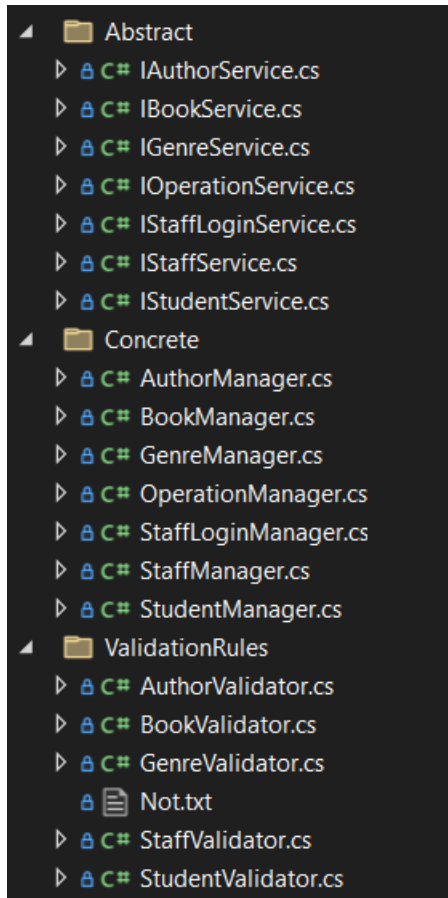
DataAccessLayer katmanında bulunan sınıflar ve yöntemler:

```
└─ Abstract
  │   ├── IAuthorDal.cs
  │   ├── IBookDal.cs
  │   ├── IGenreDal.cs
  │   ├── IOperationDal.cs
  │   ├── IRepository.cs
  │   ├── IStaffDal.cs
  │   └── IStudentDal.cs
  │
  └─ bin
  │
  └─ Concrete
    │   └─ Repositories
    │       ├── GenericRepository.cs
    │       └── Context.cs
    │
    └─ EntityFramework
        ├── EfAuthorDal.cs
        ├── EfBookDal.cs
        ├── EfGenreDal.cs
        ├── EfOperationDal.cs
        ├── EfStaffDal.cs
        └── EfStudentDal.cs
    │
    └─ Migrations
```

1. Context Sınıfı :
  - Entity Layer katmanında bulunan sınıfları, mssql de tablo olarak yansımalarını sağlar.
  - Kalıtım yoluyla DbContext sınıfından özellik almıştır.
2. Abstract Sınıflar :
  - Belirli veri tabanı işlemlerini standart hale getirmek ve kod tekrarını önlemek için abstract sınıflar kullanılır.
  - IRepository.cs sınıfı temel özellikleri içerir Diğer sınıflar (IBookDal.cs, IStaffDal gibi) IRepository sınıfından kalıtım yoluyla özelliklerini alır.
3. GenericRepository Sınıfı :
  - Bu sınıf temel IRepository.cs sınıfını kendine Implement eder ve IRepository de bulunan özelliklerin içini doldurur.
  - Temel Crud işlemlerinin gerçekleştiği veri tabanına yansıtıldığı yerdir.
4. EntityFramework Sınıfları :
  - Bu sınıfın temel amacı bağımlılıklardan kurtulmaktır.
  - Proje genelinde veri erişim işlemlerinin standartlaştırılmasını sağlar.

#### 4. Bussiness Layer (İş Katmanı) :

Bu katman, uygulamanın temel iş mantığını içerir. İş kuralları, doğrulamalar ve iş süreçleri burada tanımlanır. Bu katman, uygulamanın çekirdeğini oluşturur ve diğer katmanlar arasında bağlantı noktasıdır.

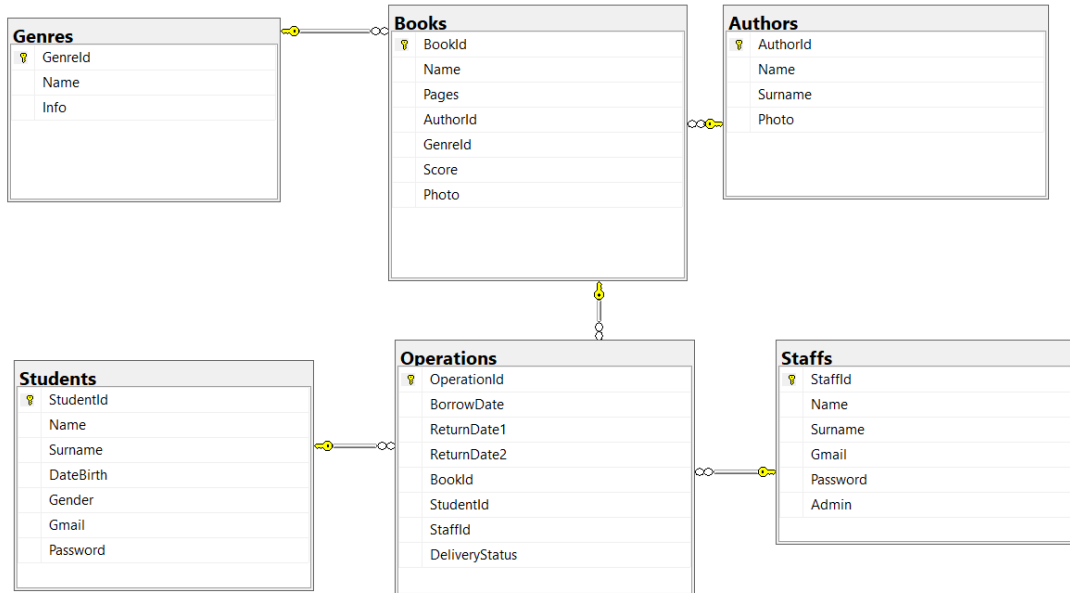


1. Abstract Sınıflar :
  - Bussines Layer da abstract sınıfların amacı, yapılacak işin arayüzünü oluşturmaktır.
  - İnterface sınıflardır.
2. Concrete Sınıfı :
  - Bu sınıflar Abstract sınıflarından miras alır. Implament işlemi yapılarak Abstract sınıfındaki methodların işlemlerini gerçekleştirir.
3. ValidationRules Sınıfları :
  - Bu sınıfın temel amacı verileri istenilen formattamı olduğunu kontrol etmektir.
  - Fluent validation eklentisi kullanılmıştır.

## 5. Database ( Veri Tabanı ) :

Veri tabanı için ilişkisel bir veri tabanı yönetim sistemi olan, MsSQL(Microsoft Sql Server) kullanılmıştır.

Veri tabanı ilişkiler şu şekildedir:



1. Genres Tablosu:
  - Genres tablosu, Kitap türlerini depolayan tablodur.
  - GenrelId, Int türünde Primary Key (Birincil Anahtar) olarak tanımlanmıştır.
  - Name ve Info String türündedir.
2. Books Tablosu:
  - Kitap bilgilerini depolar.
  - BookId, Int türünde Primary Key (Birincil Anahtar) olarak tanımlanmıştır.
  - Name ve Photo string türünde tanımlanmıştır.
  - Pages ve score int türünde tanımlanmıştır.
  - AuthorId ve GenrelId Foreign Key (Yabancı Anahtar) dır.
3. Authors Tablosu:
  - Yazar bilgilerini depolar.

- AuthorId Int türünde Primary Key (Birincil Anahtar) olarak tanımlanmıştır.
  - Name, Surname ve Photo string türünde tanımlanmıştır.
4. Students Tablosu:
- Öğrenci bilgilerini depolar.
  - StudentId Int türünde Primary Key (Birincil Anahtar) olarak tanımlanmıştır.
  - Name, Surname, Email ve Password string türünde tanımlanmıştır.
  - DateBirth, DateTime türünde tanımlanmıştır.
  - Gender, Bool türünde tanımlanmıştır.
5. Operations Tablosu:
- Kitapların emanet durum bilgilerini depolar.
  - OperationId Int türünde Primary Key (Birincil Anahtar) olarak tanımlanmıştır.
  - BorrowDate, ReturnDate1 ve ReturnDate2 DateTime türünde tanımlanmıştır. ReturnDate1 tahmini teslim tarihini kaydeder. ReturnDate2 ise Teslim edilen tarihi almaktadır.
  - BookId, StudentId ve StaffId Foreign Key (Yabancı Anahtar) dir.
  - DeliveryStatus Bool türünde tanımlanmıştır. Amacı işlemdeki kitabın teslim durumudur.
6. Staffs Tablosu:
- Personel bilgilerini depolar.
  - StaffId Int türünde Primary Key (Birincil Anahtar) olarak tanımlanmıştır.
  - Name, Surname, Email ve Password string türünde tanımlanmıştır.
  - Admin Bool türünde tanımlanmıştır. Personelin yetki düzeyi için kullanılır.

## İletişim Bilgileri

Bahadır AKKAŞOĞLU

Github : <https://github.com/18bahadir06/KatmanliMimariMVCKutuphane>

LinkedIn : <https://www.linkedin.com/in/bahad%C4%B1r-akka%C5%9Fo%C4%9Flu-b9b1b1233/>

Gmail : bahadirakkas06@gmail.com