

**2023-2024 AKADEMİK YILI
GÜZ DÖNEMİ**

BİLGİSAYAR MÜHENDİSLİĞİ PROJE TASARIMI

Ders Sorumlusu:
Doç. Serdar BİROĞUL

OKUL YÖNETİM SİSTEMİ

Hazırlayan:
Büşra ÖZDEMİR

Öğrenci No:
201001082

TEŐEKKÖR

Lisans öđrenimimde ve bu tezin hazırlanmasında gösterdiđi her türlü destek ve yardımdan dolayı çok deđerli hocam Doç. Dr. Serdar BİROĐUL'a en içten dileklerimle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

2 Ocak 2024

Büşra ÖZDEMİR

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ	iv
KISALTMALAR.....	v
ÖZET	vi
ABSTRACT	vii
1. GİRİŞ.....	1
2. KULLANILAN ARAÇ VE YÖNTEM.....	2
2.1. ASP.NET CORE WEB UYGULAMASI PROJESİ	2
2.1.1. MVC Şablonu.....	2
2.1.1.1. Model	2
2.1.1.2. View.....	3
2.1.1.3. Controller.....	3
2.2. KATMANLI MİMARİ.....	3
2.3. MS SQL SERVER VERİTABANI.....	4
2.4. PROJEDE KULLANILACAK TEKNOLOJİ VE KÜTÜPHANELER.....	5
2.4.1. Bootstrap	5
2.4.2. Entity Framework Core	5
2.4.2.1. Code First Yaklaşımı	5
2.4.2.2. Kullanılacak Paketler	6
2.4.3. FluentValidation	6
2.4.4. Identity	7
3. OKUL YÖNETİM SİSTEMİ ARAYÜZÜ	8
3.1. GİRİŞ SAYFASI	8
3.2. KAYIT OL SAYFASI.....	9
3.3. ANA SAYFA.....	9
3.4. ÖĞRENCİ-ÖĞRETMEN PANELİ	10
3.5. ADMIN PANELİ	11
4. OKUL YÖNETİM SİSTEMİNİN MODELLENMESİ.....	12
4.1. KULLANIM DURUMU (USE CASE) DİYAGRAMI	12
4.2. SINIF DİYAGRAMI	12
4.3. DERS KAYNAĞI EKLEME ETKİNLİK DİYAGRAMI	13
4.4. MESAJLAŞMA ARDIL ETKİLEŞİM (SEQUENCE) DİYAGRAMI.....	14
5. SONUÇLAR VE ÖNERİLER.....	16
6. KAYNAKLAR.....	17
ÖZGEÇMİŞ.....	xviii

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. Yazılım Geliştirme Aşamaları.....	1
Şekil 2.1.1. Asp.Net Core Web App	2
Şekil 2.1.2. Framework	2
Şekil 2.1.1.1. MVC Tasarım Deseni [3].....	3
Şekil 2.2.1. Katmanlı Mimari Yapısı [4].....	4
Şekil 2.4.2.2.1. Entity Framework Core Paketi.....	6
Şekil 2.4.2.2.2. Entity Framework Core Sql Server Paketi	6
Şekil 2.4.2.2.3. Entity Framework Core Tools Paketi.....	6
Şekil 2.4.2.2.4. Entity Framework Core Design Paketi	6
Şekil 2.4.3.1. FluentValidation Kütüphaneleri.....	7
Şekil 2.4.4.1. Identity Kütüphanesi	7
Şekil 3.1.1. Giriş Sayfası	8
Şekil 3.2.1. Kayıt Ol Sayfası	9
Şekil 3.3.1. Ana Sayfa	9
Şekil 3.4.1. Öğrenci ve Öğretmen Paneli	10
Şekil 3.5.1. Admin Paneli.....	11
Şekil 4.1.1. Kullanım Durumu Diyagramı	12
Şekil 4.2.1. Sınıf Diyagramı.....	13
Şekil 4.3.1. Ders Kaynağı Ekleme Etkinlik Diyagramı	14
Şekil 4.4.1. Öğrenci-Öğretmen Mesajlaşması Sequence Diyagramı	15

KISALTMALAR

CRUD	Create, Read, Update, Delete
EF	Entity Framework
LINQ	Language Integrated Query
MS	Microsoft
MVC	Model-View-Controller
ORM	Object to Relational Mapping
SQL	Structured Query Language
UML	Unified Modelling Language

ÖZET

OKUL YÖNETİM SİSTEMİ

Büşra ÖZDEMİR

Düzce Üniversitesi

Mühendislik Fakültesi Bilgisayar Mühendisliği Proje Tasarımı Raporu

Danışman: Doç. Dr. Serdar BİROĞUL

Ocak 2024, 17 sayfa

Eğitimin de teknoloji ortamına girmesi okul içerisindeki eğitim anlayışına geniş bir yelpaze getirmiş öğrencilere, internet ortamından da ders çalışabilmeyi benimsetmiştir. Bu durum ise internette ders kaynakları arama ihtiyacı doğurmuştur. Okulun verdiği eğitim ile uyuşan kaynaklar bulabilmek, öğretmen-öğrenci iletişimini en hızlı gerçekleştirebilmek ve böylelikle daha başarılı bir nesil yetiştirebilmek amacıyla bu çalışmada, eğitim veren kurumlara yönelik okul yönetim sistemi projesinin analiz ve tasarım aşamasının gerçekleştirilmesi hedeflenmiştir. Tasarım aşaması; kullanım durumu diyagramı, sınıf diyagramı, etkinlik diyagramı, ardıl etkileşim diyagramı gibi bazı UML (Unified Modelling Language) diyagramlarıyla modellenmiştir.

Anahtar sözcükler: Okul yönetim sistemi, UML, UML modellemesi, Uzaktan eğitim

ABSTRACT

SCHOOL MANAGEMENT SYSTEM

Büşra ÖZDEMİR

Düzce University

Faculty of Engineering, Department of Computer Engineering Project Design Report

Supervisor: Assoc. Prof. Dr. Serdar BİROĞUL

January 2024, 17 pages

The integration of education into the technological environment has broadened the scope of educational approaches within schools and has accustomed students to studying through online platforms. This situation has led to the necessity of searching for educational resources on the internet. In order to find resources compatible with the education provided by the school, to realize teacher-student communication as quickly as possible and thus to raise a more successful generation, this study aims to carry out the analysis and design phase of the school management system project for educational institutions. The design phase has been modeled using some UML (Unified Modelling Language) diagrams such as use case diagrams, class diagrams, activity diagrams and sequence diagrams.

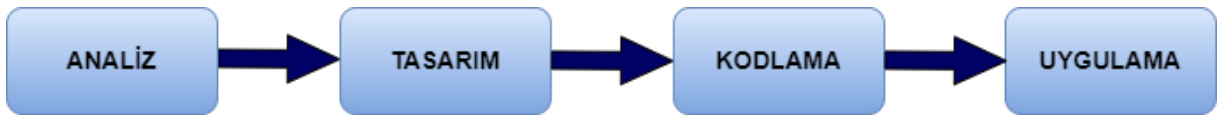
Keywords: School management system, UML, UML modelling, Distance education

1. GİRİŞ

Geçmişten günümüze teknoloji kullanımının artması, birçok işlemin dijital ortamdan yapılabilmesine olanak sağlamıştır. Covid-19 süreciyle beraber eğitimin de dijitalleşmesiyle uzaktan eğitim kavramı hayatımıza entegre edildi. Uzaktan eğitim sayesinde dijital ortama alışan öğrenciler; kendini geliştirmek sebebiyle ya da sorun yaşadığı konuları açıklığa kavuşturabilmek amacıyla internet ortamında doküman, video araştırması yapmaktadır. Öğrencilere yönelik hazırlanan bu kaynaklar her geçen gün artmaktadır. Fakat birçok kaynağın ücretli olması ya da kaynakta Türkçe harici bir dil kullanılması öğrenciler için olumsuzluğa sebebiyet vermektedir. Tasarlamak istediğim Okul Yönetim Sistemindeki platform ile öğrenci-öğretmen iletişiminin hızlı gerçekleşebilmesini, öğrencilere hitap eden ve ders sorumlularının anlattığı içeriklerle bağdaşan kaynaklar bulabilmesini hedefleyerek web tabanlı projemi yapmayı planladım.

Projemde öğretmenler ilgili dersler için öğrencilere yönelik video ya da doküman kaynağı ekleyebilecek, öğrencilerin sınav, performans notlarını girebilecek, öğrencilere hızlı ulaşabilmeleri amacıyla duyurular yapabileceklerdir. Öğrenciler ise panel sayesinde ilgili derslere yüklenen kaynaklara erişim sağlayarak ders tekrarlarını yapabilecek ve notlarını görüntüleyebilecekler. Bu şekilde hem öğrenci hem öğretmen hem de veliler için daha kolay bir okul süreci benimsenmiş olacaktır.

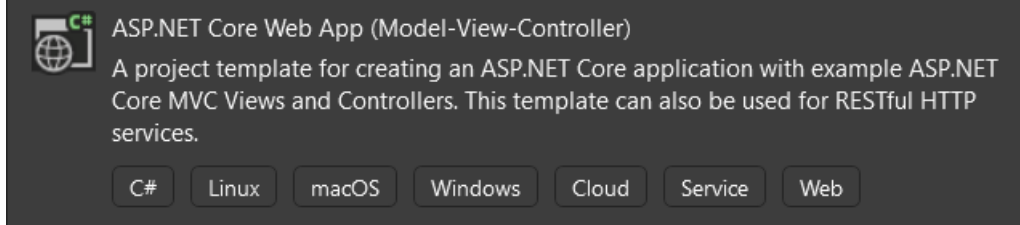
Okul yönetim sisteminin geliştirilmesi süreci temel olarak aşağıdaki aşamalardan geçmektedir. Bu rapor okul yönetim sisteminin analiz ve tasarım aşamasını içermektedir. Kodlama ve Uygulama aşaması ise bahar dönemi mezuniyet tezi dersi kapsamında yapılacaktır.



Şekil 1.1. Yazılım Geliştirme Aşamaları

2. KULLANILAN ARAÇ VE YÖNTEM

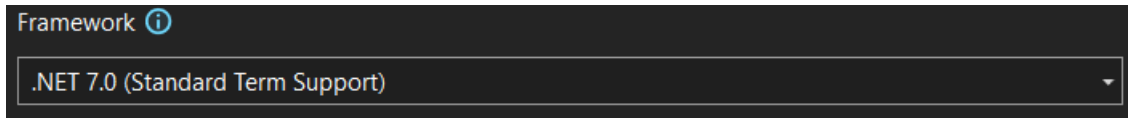
2.1. ASP.NET CORE WEB UYGULAMASI PROJESİ



Şekil 2.1.1. Asp.Net Core Web App

Okul Yönetim Sistemi C# dili vasıtasıyla Asp.Net Core Web Uygulaması MVC şablonu kullanılarak geliştirilecektir. Asp.Net Core, .NET platformunun platform bağımsız halidir ve .NET Core platformunda web tabanlı uygulamalar geliştirmeye olanak sağlayan bir pakettir [1]. .NET web, mobil ve masaüstü uygulamalar geliştirmeye yönelik olup açık kaynak kodlu platformdur. Ayrıca içerisindeki birçok eklenti ve kütüphaneler yardımıyla sürdürülebilir ve performanslı uygulamalar geliştirilebilmekte ve herhangi bir işletim sisteminde yerel olarak çalıştırılabilmesine imkan tanımaktadır.

Projede Framework olarak aşağıdaki görselde de belirtilen ve .NET'in güncel versiyonlarından biri olan 7.0 versiyonu tercih edilecektir.



Şekil 2.1.2. Framework

.NET Framework Windows'ta çalışan hizmetleri, web siteleri, masaüstü uygulamaları ve daha fazlasını desteklemektedir [2].

2.1.1. MVC Şablonu

MVC (Model-View-Controller) yazılım mimari desenlerinin bir parçası haline gelmiştir bu sebeple yazılım sektöründe önemli bir konuma ulaşmıştır. MVC, Model-View-Controller'in baş harflerinin kısaltılmasıyla anılmaktadır. MVC şablonu Model-View-Controller olmak üzere 3 katman barındırmaktadır.

2.1.1.1. Model

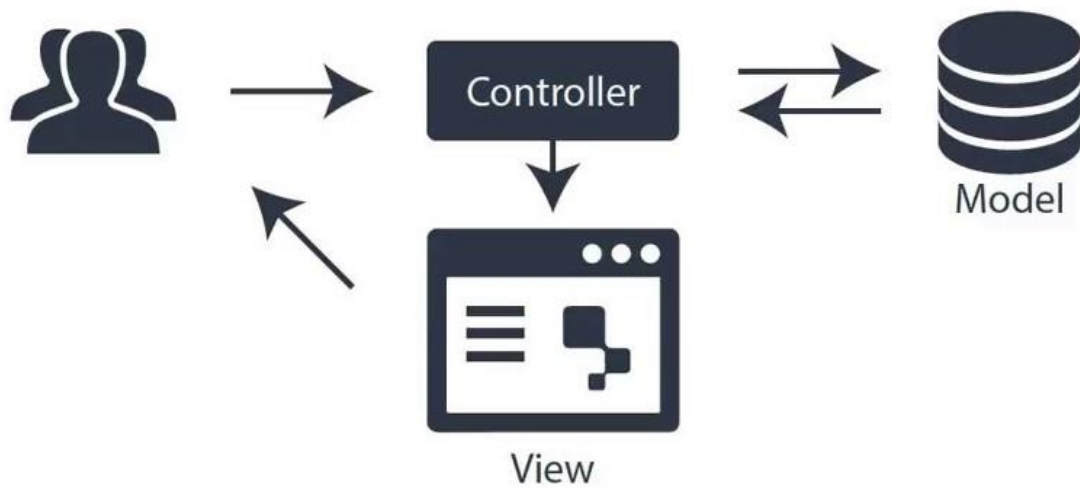
Model katmanı veri ile ilgili işlemlerin yapıldığı katmandır. İçerisinde veri tabanı ilişkileri, veri tabanına erişim gibi işlemleri barındırır.

2.1.1.2. View

Kullanıcıların web sitesinde dolaşırken gördüğü tüm arayüzler View katmanında hazırlanır. View katmanının içerisinde HTML dosyaları barındırılmaktadır. Bir diğer görevi de kullanıcılardan aldığı istekleri Controller katmanına iletmektir.

2.1.1.3. Controller

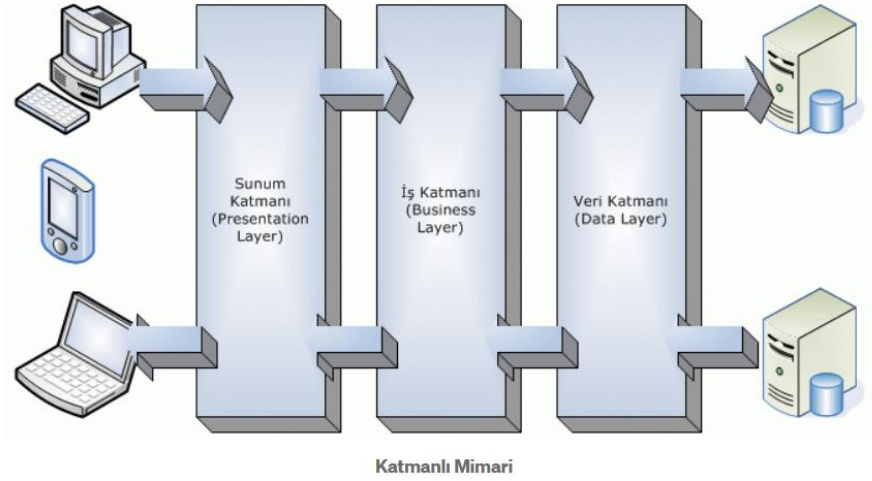
Controller katmanında Model ile View arasında bağlantı kurulur. Kullanıcılardan gelen istekler değerlendirilerek ilgili metoda/fonksiyona gider ve metod işlendikten sonra kullanıcıya hangi view'ı döndüreceğini belirtir.



Şekil 2.1.1.1. MVC Tasarım Deseni [3]

2.2. KATMANLI MİMARİ

Katmanlı mimari kapsamlı ve karmaşık yazılım projelerini daha iyi yönetebilmek için kullanılan bir yapılandırma modelidir. Projenin sürdürülebilirliği ve okunabilirliği kolay olması açısından özellikle de kapsamlı projelerde katmanlı mimari yapısı tercih edilmektedir. Bu sebeple okul yönetim sistemi projesi de katmanlı mimari üzerine kurulacaktır.



Şekil 2.2.1. Katmanlı Mimari Yapısı [4]

Katmanlı mimariyi temel olarak oluşturan katmanlar;

- *Data Access Layer (Veri Erişim Katmanı)*: Bu katman adı altında veri ile ilgili CRUD(Create-Read-Update-Delete) operasyonları yapılmaktadır.
- *Business Layer (İş katmanı)*: İş kuralları bu katmanda yazılır. Bu katman Data Access katmanını kullanmaktadır. Data Access katmanına direkt erişim doğru olmadığı ve literatüre uymadığı için kullanıcılardan gelen veriler önce Business katmanına ardından Data Access katmanına aktarılmaktadır.
- *Presentation Layer (Sunum Katmanı)*: Kullanıcı ile etkileşim yapıldığı katmandır. Buradaki temel amaç kullanıcıya verileri bir web sayfası ya da bir Windows form yardımıyla göstermektir. Kullanıcıdan gelen veriler bu katmandan alınarak önce Business Layer katmanına ardından da Data Access katmanına iletilmektedir.
- *Entity Layer (Varlık Katmanı)*: Bu katmanda projede kullanılacak sınıflar/nesneler belirtilir. İlgili sınıflar içerisinde barındırılmak istenen nitelikler(property) belirlenir. Entity katmanını diğer tüm katmanlar kullanmaktadır.

2.3. MS SQL SERVER VERİTABANI

Projede kullanılacak olan MS (Microsoft) SQL (Structured Query Language) Server veri tabanı, ilişkisel bir veri tabanı olup Microsoft tarafından geliştirilmiştir. Bu veri tabanı aracılığıyla karmaşık ve büyük kapsamlı veriler SQL Server Management Studio'nun sunduğu arayüz sayesinde kolaylıkla depolanabilir, yönetilebilir ve ilgili sorgulamalar yapılabilir.

2.4. PROJEDE KULLANILACAK TEKNOLOJİ VE KÜTÜPHANELER

2.4.1. Bootstrap

Bootstrap html, css, javascript kullanılarak yazılmış açık kaynak kodlu bir kütüphanedir. Bu kütüphane ile akıllı telefonlardan dizüstü/masaüstü bilgisayarlara kadar tüm cihazlar ile uyumlu ve duyarlı tasarım yapmak oldukça kolaydır. Herhangi bir kullanıcı projemizi içeren siteye hangi tür cihazdan erişirse erişsin tasarım bozulmalarının önüne geçmek amacıyla proje arayüzünde ağırlıklı olarak Bootstrap kütüphanesinden yararlanılacaktır.

2.4.2. Entity Framework Core

Entity Framework Microsoft tarafından veri erişim teknolojisi olarak sunulan bir ORM(Object to Relational Mapping) tekniğidir. ORM tekniği ilişkisel veri tabanı ile web tabanlı uygulamamız arasında bir nevi köprü görevi gören, ilişkileri ve nesneleri yönetmek amacıyla kullanılan bir yöntemdir [5]. Daha az SQL sorgusu kullanılarak kolaylık sağlar ve aynı zamanda her türlü veri tabanına uygun çalışabilmektedir.

Yazılım geliştiricilerinin veri ile ilgili işlemleri rahatlıkla yaparak uygulama tarafına odaklanmasına olanak sağlamaktadır. EF (Entity Framework) aracılığıyla bağlantının açılıp kapanmasına gerek kalmadan sadece sorgular ile CRUD operasyonları rahatlıkla yapılabilmektedir.

2.4.2.1. Code First Yaklaşımı

Programlama dili ile veri tabanı arasında bağ kurarak veri tabanının otomatik olarak oluşturulmasını sağlayan tekniktir. Veri tabanına girerek manuel olarak tablolar oluşturmak yerine tüm veri tabanı işlemlerini kod aracılığıyla yapabilmemize olanak sağlar. Geliştirici ile veri tabanı arayüzü etkileşimi minimize edilerek hızlı bir proje süreci sağlar.

Code First tekniğinde programa dili üzerindeki her nesne/class veri tabanı üzerinde bir tabloya, classlar içerisindeki her bir alan/property ise tablo içerisindeki sütun yapılarına denk gelmektedir.

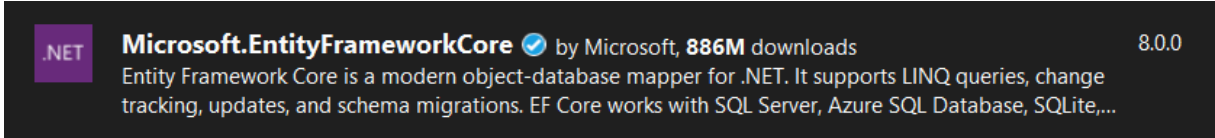
Bahsedilen ilgili sınıflar katmanlı mimari üzerinde Entity katmanına yazılmaktadır. İlgili sınıflar ve bağlantı stringinin, tablo DbSet'lerinin içerdiği Context sınıfı da yazılmasının ardından Package Manage Console açılır. Context sınıfı hangi katmanda yer alıyorsa o katman seçilir ve Console'a Add-Migration MigrationAdı yazılıp enter tuşuna basılır.

Bu kod Migration oluşturmayı sağlayacaktır. Ardından Console ekranına Update-Database

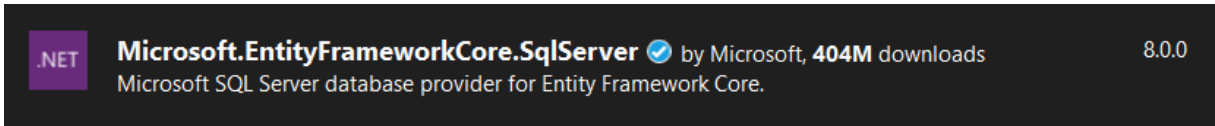
yazılır ve bu işlem ile oluşan Migration şablonuna göre veri tabanının otomatik olarak oluşturulması sağlanır.

2.4.2.2. Kullanılacak Paketler

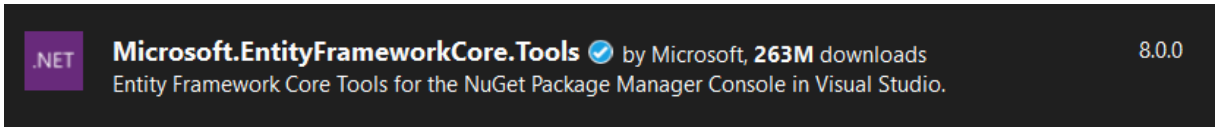
EF yapısını kullanabilmemiz için bazı EF bileşenleri ya da paketleri projeye dahil edilecektir. Bu paketler ORM işlemlerini yapabilmemize olanak sağlar. Veri tabanı işlemlerini kodla yönetme, sorgulama ve veri erişimi gibi işlevleriyle geliştiricilere büyük kolaylık sağlayan 4 ana paket aşağıda belirtilmektedir;



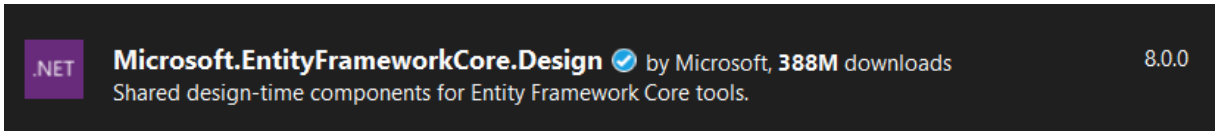
Şekil 2.4.2.2.1. Entity Framework Core Paketi



Şekil 2.4.2.2.2. Entity Framework Core Sql Server Paketi



Şekil 2.4.2.2.3. Entity Framework Core Tools Paketi

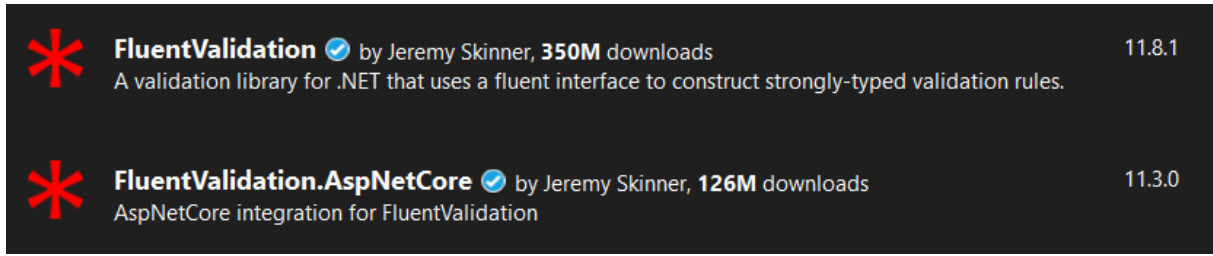


Şekil 2.4.2.2.4. Entity Framework Core Design Paketi

EF'nin yukarıda belirtilen paketleri proje .Net 7.0 versiyonu ile oluşturulacağından ilgili paketlerin de 7.0 versiyonunun kurulması zorunludur. Bu paketler Sql Server bağlantı metnini barındıran DbContext sınıfını ve ilişkili Entity sınıflarını oluşturma, yönetme gibi süreçleri kolaylaştırır. Code first yaklaşımında kullanılacak migration yönetimini yapabilmemize imkan sağlar. Ayrıca LINQ (Language Integrated Query) sorgularını kullanarak veri üzerinde yapılacak CRUD operasyonlarını gerçekleştirmek için gerekli olan yapıları sunar.

2.4.3. FluentValidation

FluentValidation kütüphanesi doğrulama kütüphanesi olup .NET tabanlı platformlarda kullanılmaktadır. Bu kütüphane, kurallarını nesne odaklı bir şekilde belirtmenizi sağlayan bir interface sağlamaktadır [6].



Şekil 2.4.3.1. FluentValidation Kütüphaneleri

Doğrulama kurallarının anlaşılmasının ve tanımlanmasının kolay olması, validasyon kurallarını model sınıflarıyla entegre ederek kodların daha düzenli ve anlaşılır olmasını sağlaması FluentValidation kütüphanesinin avantajları arasında yer almaktadır [6].

2.4.4. Identity

Asp.Net Core Identity kütüphanesi web tabanlı uygulamalarda kimlik doğrulama ve yetkilendirme gereksinimlerini karşılamak için tercih edilmektedir. Bu kütüphane üyelerin, giriş, çıkış, yetkilendirme, token, şifre hatırlatma vs. tüm işlemleri pratik bir şekilde gerçekleştirmemizi sağlayan günümüzde yaygınca kullanılan bir üyelik sistemidir [7]. Web tabanlı projemiz içerisinde Identity kullanabilmek ve Identity nimetlerinden yararlanabilmek için projeye kuracağımız iki paket aşağıda belirtilmiştir;



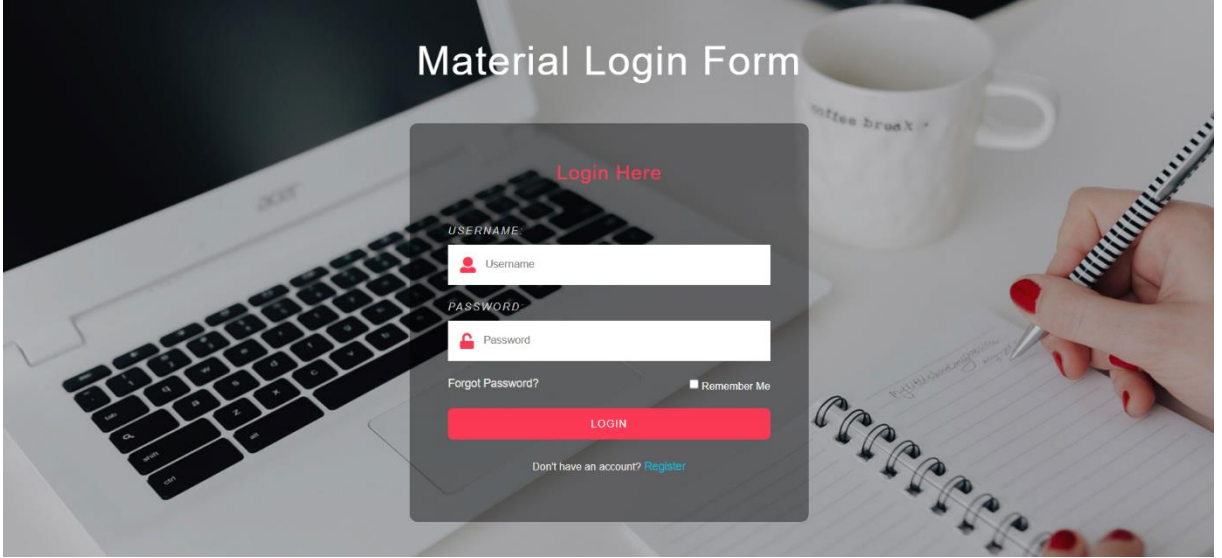
Şekil 2.4.4.1. Identity Kütüphanesi

Identity sayesinde projemizde kullanıcı ve şifre yönetiminin yanı sıra kullanıcılara rollendirmeler yaparak ilgili sayfalara erişimini kısıtlayabiliriz. Örneğin rollendirme sayesinde yetkisi olmayan kullanıcı Admin paneline erişemeyecek. Projemize entegre edeceğimiz bu kütüphane kullanıcıların kaydolması, giriş yapması, şifre değişimi ya da diğer bilgilerin güncellenmesi sırasında oldukça kolaylık sağlayarak güvenlik konusunda da önemli bir katkısı olacaktır.

3. OKUL YÖNETİM SİSTEMİ ARAYÜZÜ

Web tabanlı Okul Yönetim Sistemi projesi için hazır şablonlardan yararlanılacaktır. Bu şablonlar seçilirken okul kavramıyla bağlantılı arayüzler olmasına dikkat edilmiş ve proje içerisinde barındıracak özelliklere uygun şablonlar olmasına özen gösterilmiştir. Arayüzlerde olmayan sayfa veya içerikler proje geliştirilirken ek olarak ilave edilecektir. Şablonlar, istenilmeyen sayfalar ya da eklentilerden arındırılarak proje amacına uygun hale getirilecektir. Böylece renklendirmeler, animasyonlar, efektler, butonlar ve menüler ile kullanıcı dostu arayüzler olması amaçlanmıştır. Ayrıca arayüzler ingilizce olduğu için türkçeleştirilerek kullanıma sunulacaktır.

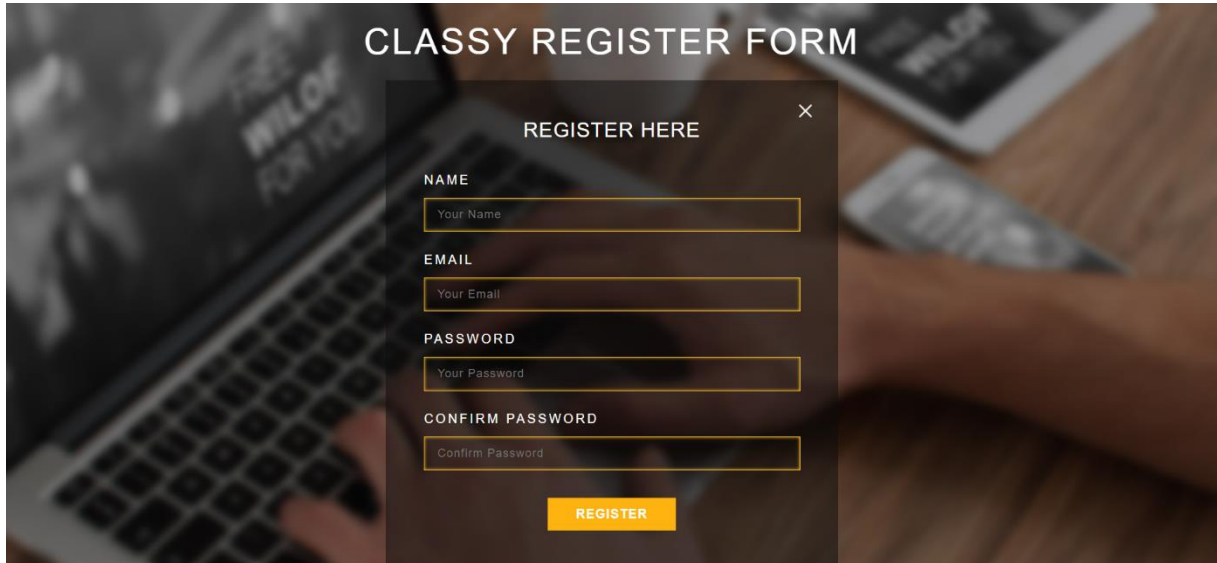
3.1. GİRİŞ SAYFASI



Şekil 3.1.1. Giriş Sayfası

Öğretmen, öğrenci ya da admin rolündeki kullanıcılar yukarıdaki sayfa aracılığıyla sitede oturum açabileceklerdir. Kullanıcı adı ve şifre ile giriş yaparak öğretmen ve öğrenci ilgili panellerine, admin ise kendi paneline yönlendirilecektir. Eğer kullanıcı şifresini unuttuysa mail yardımıyla şifresini güncelleyebilecektir. Kullanıcının hiç hesabı yoksa alt kısımda yer alan Hesap oluştur linki ile hesap oluştur sayfasına yönlendirilecektir.

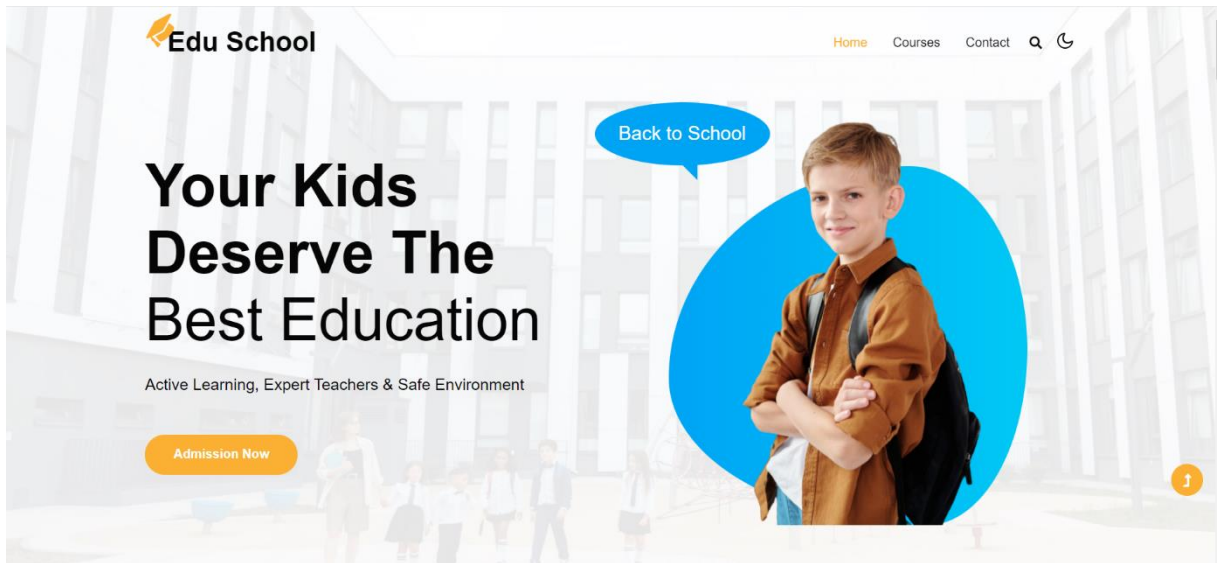
3.2. KAYIT OL SAYFASI

A dark-themed registration form titled "CLASSY REGISTER FORM" is displayed over a background image of a laptop and a hand. The form has a close button (X) in the top right corner. It contains four input fields: "NAME" with placeholder "Your Name", "EMAIL" with placeholder "Your Email", "PASSWORD" with placeholder "Your Password", and "CONFIRM PASSWORD" with placeholder "Confirm Password". A yellow "REGISTER" button is at the bottom.

Şekil 3.2.1. Kayıt Ol Sayfası

Kayıt ol sayfası aracılığıyla kullanıcı kendi bilgilerini girerek sisteme kayıt olabilecek. Öğretmen kayıt olacaksa ilgili açılır menü üzerinden Öğretmen seçeneğini, öğrenci kayıt olacaksa da Öğrenci seçeneğini seçerek kayıt tipini belirtmelidir. Seçmediği ya da zorunlu alanları boş bıraktığı takdirde kullanıcıya anlayabileceği şekilde uyarılarda bulunulacaktır. Kullanıcı ilgili alanlara yanlış değerler girdiğinde projenin hataya düşmesi yerine gerekli doğrulama kuralları vasıtasıyla kullanıcıya mesajlar gösterilecektir.

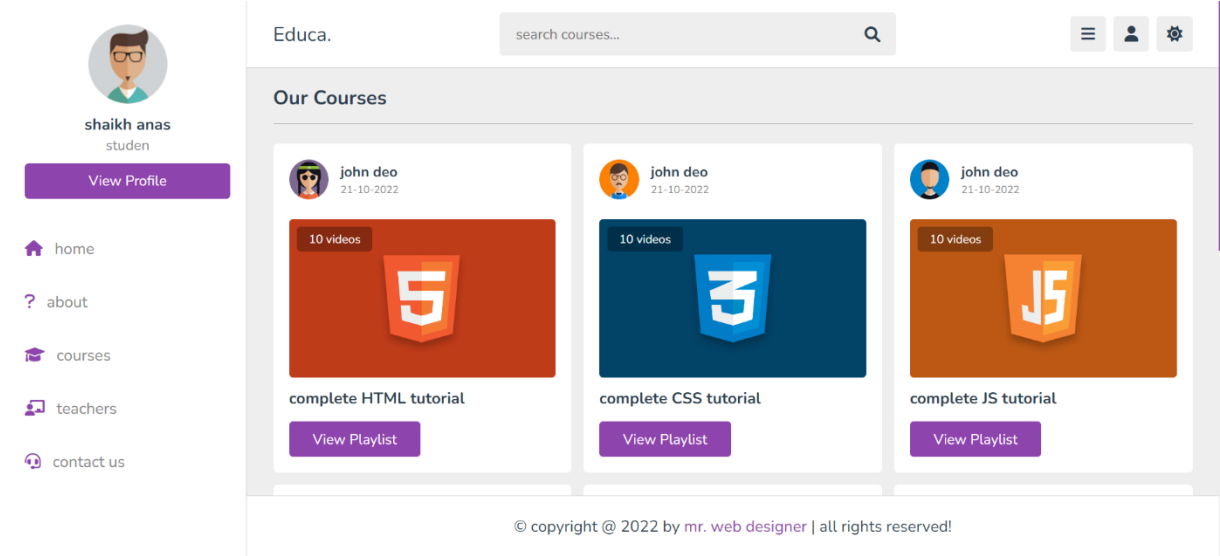
3.3. ANA SAYFA



Şekil 3.3.1. Ana Sayfa

Kullanıcıların karşısına çıkacak ve okulun tanıtımını içerecek olan sayfadır. Bu sayfada okula ait bilgiler yer alacak. Haberler sayfası eklenerek okula yönelik haberler yapılacaktır. İletişim sayfası vasıtasıyla herhangi bir kullanıcı site yöneticisine mesaj atabilecek. Şablondaki ana sayfada kullanıcının giriş yapabileceği ya da kayıt olabileceği menü bulunmadığından bu menü proje yapılırken dahil edilecektir.

3.4. ÖĞRENCİ-ÖĞRETMEN PANELİ



Şekil 3.4.1. Öğrenci ve Öğretmen Paneli

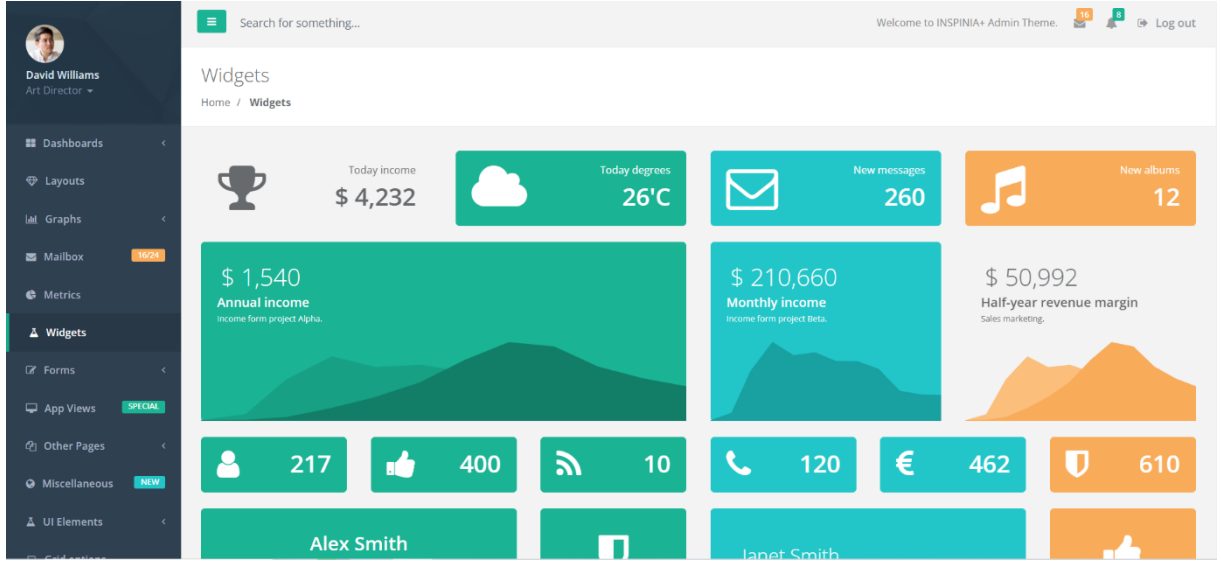
Panel sayesinde öğretmen derslere kaynak yüklemesi yapabilir, öğrencilere duyurular yapabilir ve kendisine gelen mesajları okuyup yanıtlayabilecektir.

Öğrenci rolündeki kullanıcı ise ilgili derslere yüklenen dokümanlara, videolara erişebilecek ve öğretmenlerin yaptığı duyuruları bildirim menüsünden görüntüleyebilecekler.

Sistemde kayıtlı olarak bilgisi tutulan kullanıcılar, sistemde kayıtlı olan diğer kullanıcılar içerisinde herhangi birinin mailini girerek ilgili kişiye mesaj gönderebileceklerdir.

Öğrenci rolündeki bir kullanıcı öğretmen ya da admin paneline ya da öğretmen rolüne sahip kullanıcı diğer panellere erişemez. Erişmeye çalıştığı takdirde kullanıcı, ilgili hata sayfasına yönlendirilecektir. Böylelikle yetkilendirme işlevi projede gerçekleştirilerek yetkisi olmayan kullanıcıların belirli panellere erişimi engellenecektir.

3.5. ADMIN PANELİ



Şekil 3.5.1. Admin Paneli

Yukarıya görseli eklenen admin panelinde de kullanılmayacak özellik ve sayfalar kaldırılarak panelin daha sade bir arayüze sahip olması hedeflenecektir. Arama çubuğu, bildirim menüsü ve mesaj menüsü de hazırlanarak kullanışlı hale getirilecektir.

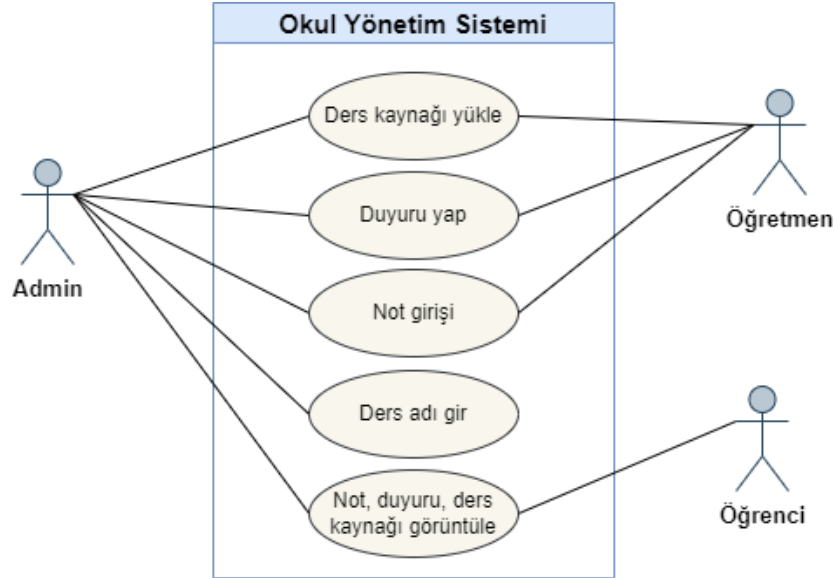
Admin kullanıcısı sistemin her yerine erişebilen role sahiptir. Admin; sistemin istatistiklerini görebilir, ana sayfa üzerinden gönderilen mesajları görüntüleyebilir ve yanıtlayabilir, iletişim sayfasındaki adres bilgilerini güncelleyebilir, ana sayfada görüntülenmesi için haber yayınlatabilir, ders ekleyebilir, yeni kullanıcı tanımlayarak rol atayabilir.

4. OKUL YÖNETİM SİSTEMİNİN MODELLENMESİ

Okul yönetim sistemi web projemiz herhangi bir okul adına öğrenci öğretmen ilişkisini daha hızlı yönetebilmek ve öğrencilerin ders kaynaklarına daha hızlı bir şekilde ulaşabilmeleri için tasarlanmıştır. Bu başlık altında okul yönetim sisteminin senaryosunu bazı UML diyagramlarıyla tasarlayacağız.

4.1. KULLANIM DURUMU (USE CASE) DİYAGRAMI

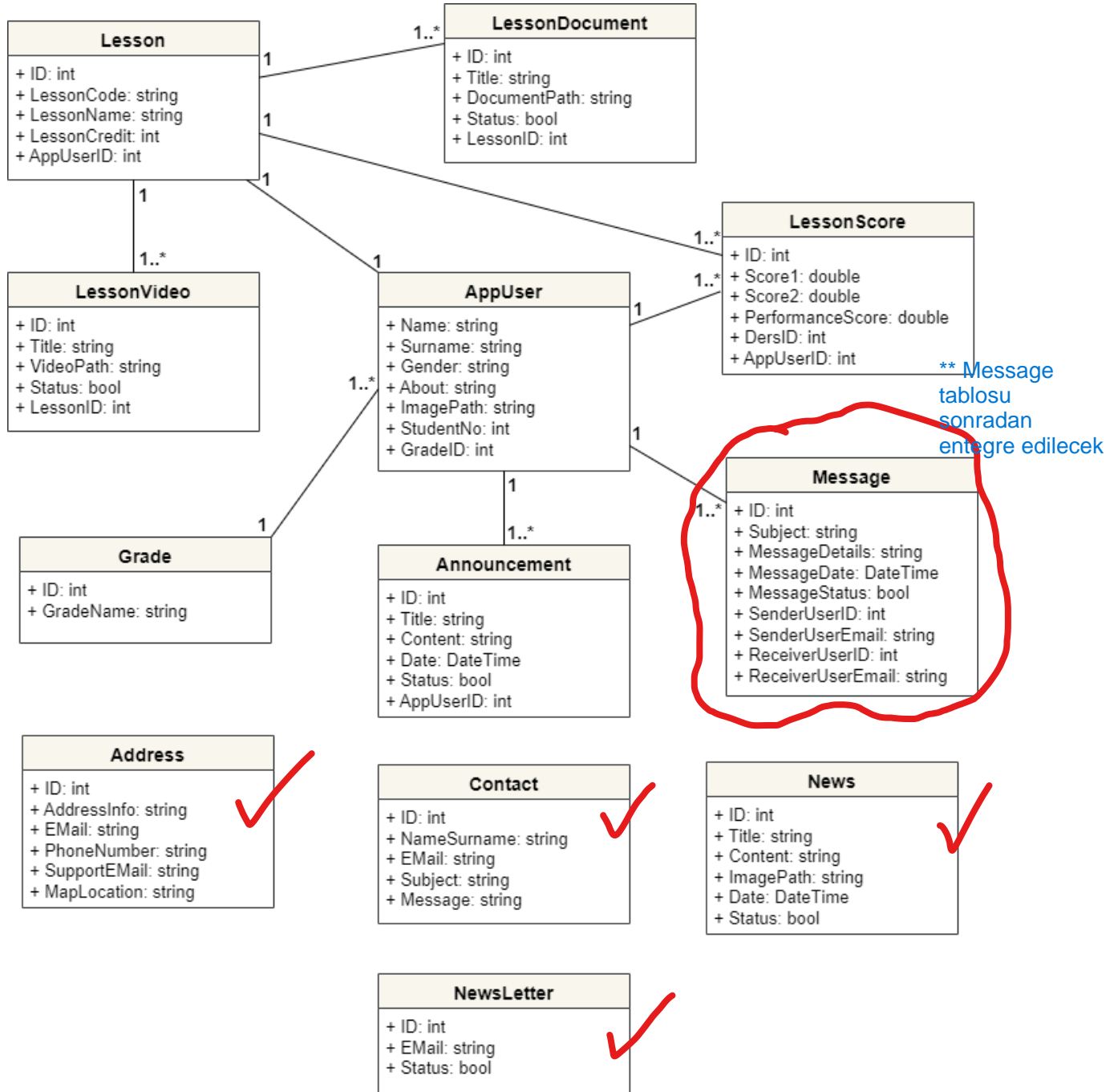
Admin sistemde her şeye erişebilen role sahipken Öğretmen yalnızca ders kaynağı yükleyip duyuru yapabilme ve not girişi yapabilme yetkisine sahiptir. Öğrenci ise Not, duyuru ve ders kaynağını görüntüleyebilmektedir. Bu sistemi modelleyen kullanım durumu diyagramı aşağıdaki gibidir.



Şekil 4.1.1. Kullanım Durumu Diyagramı

4.2. SINIF DİYAGRAMI

Şekil 4.2.1. ile Okul Yönetim Sistemi web tabanlı projesi için kullanılacak olan sınıfların özellikleri ve sınıflar arasındaki ilişkiler UML'nin sınıf diyagramı ile modellenmektedir [8].

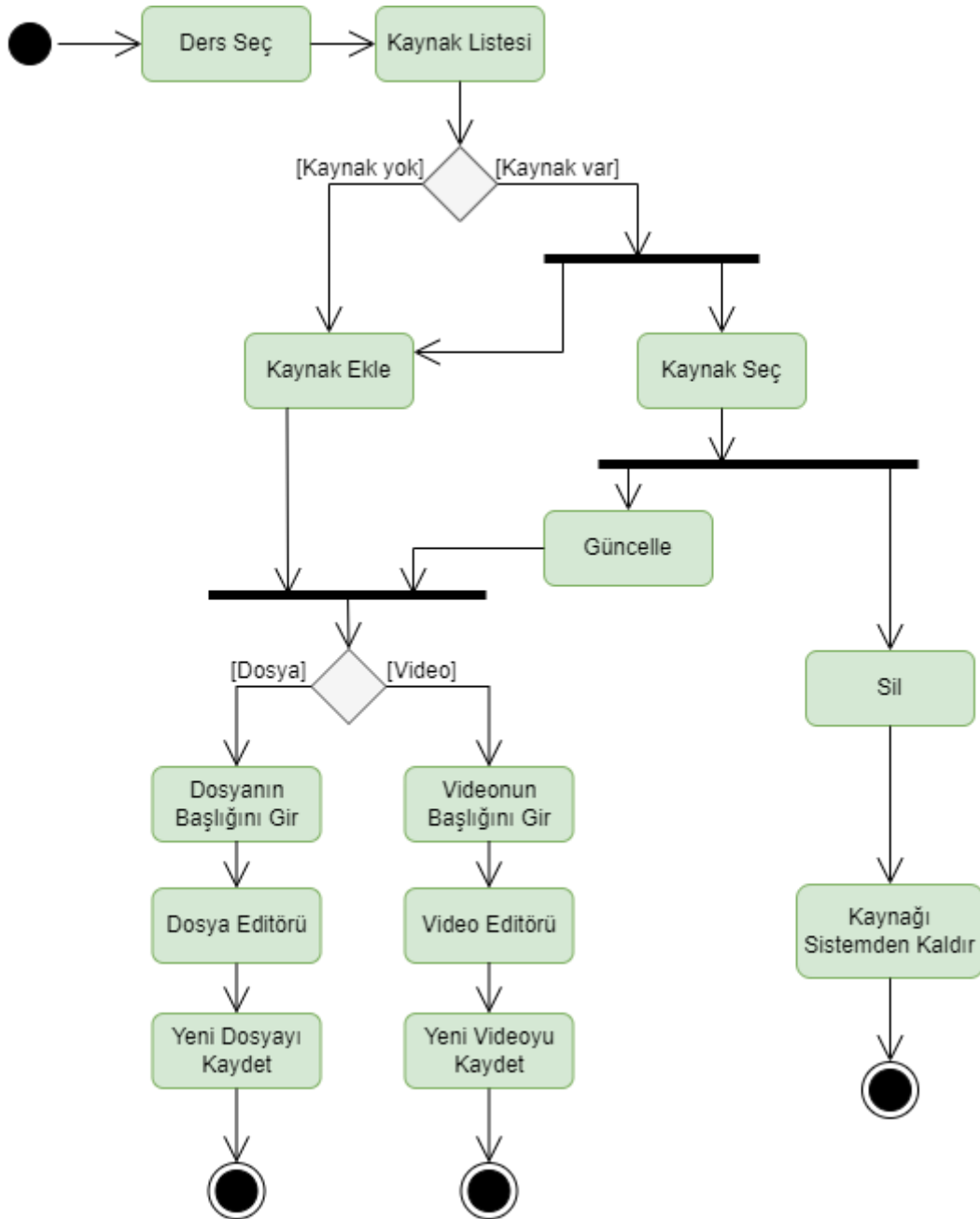


Şekil 4.2.1. Sınıf Diyagramı

4.3. DERS KAYNAĞI EKLEME ETKİNLİK DİYAGRAMI

Ders kaynağı ekleyebilmek için Öğretmen rolündeki kullanıcı sisteme giriş yapması gerekmektedir. Sisteme giriş yapan kullanıcı Öğrenci rolünde ise ders ekleme sayfasına erişememektedir. Öğretmen; sisteme Admin tarafından eklenmiş olan Ders'lerden hangisine kaynak ekleyecekse o dersi seçerek ilgili adımları takip etmelidir. Bu senaryoyu gösteren

etkinlik diyagramı aşağıda verilmiştir.



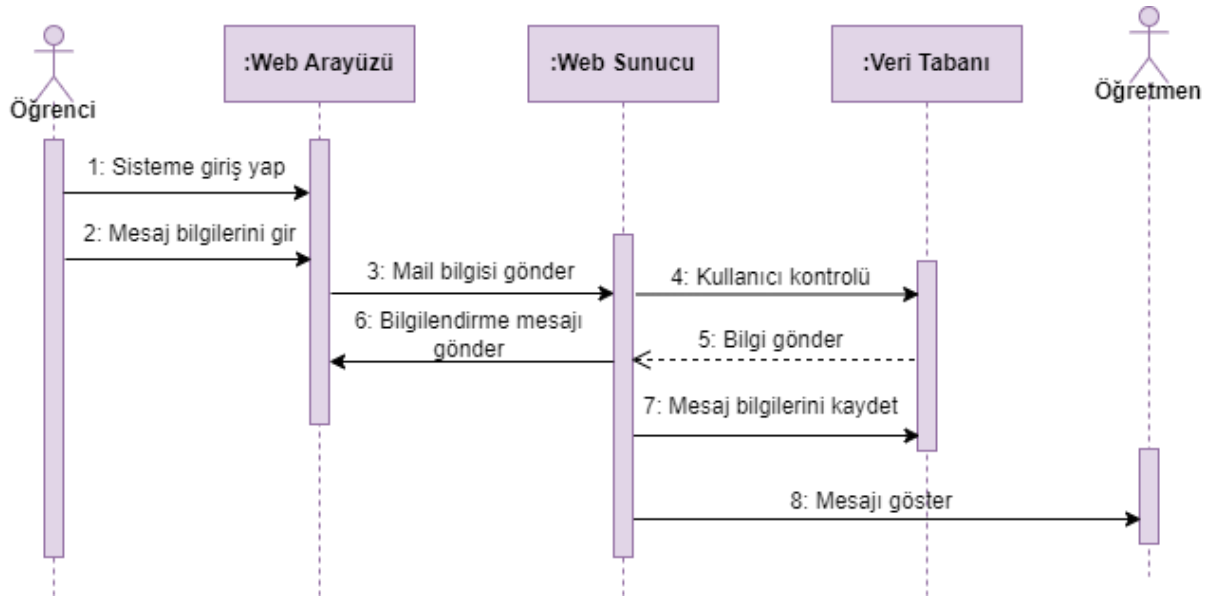
Şekil 4.3.1. Ders Kaynağı Ekleme Etkinlik Diyagramı

4.4. MESAJLAŞMA ARDIL ETKİLEŞİM (SEQUENCE) DİYAGRAMI

Mesaj gönderme sistemi için öğrenci sisteme kullanıcı ad ve şifre bilgisiyle giriş yapması gerekmektedir. Mesaj göndermek istediği öğretmen kullanıcısının mail bilgisini ve ardından mesaj içeriğini sisteme girmelidir. Eğer girilen mail bilgisine ait veri tabanında bir kullanıcı

bulunmuyorsa öğrenciye uyarı mesajı gönderilmelidir. Maile ait kullanıcı veri tabanında var ise mesaj bilgileri veri tabanına kaydedilir ve ilgili mail bilgisine sahip öğretmen kullanıcısının arayüzüne mesaj bildirimi gönderilir. Öğretmen dilerse bu mesajı açarak yanıt verebilir.

Mesajlaşma işlemi veri tabanına kayıtlı olan kullanıcılar arasında gerçekleşebilmektedir. Aşağıda ise Öğrenci-Öğretmen kullanıcıları arasındaki mesajlaşma süreci sequence diyagramı ile modellenmiştir.



Şekil 4.4.1. Öğrenci-Öğretmen Mesajlaşması Sequence Diyagramı

5. SONUÇLAR VE ÖNERİLER

Teknoloji çağı yönelimi, eğitim kurumlarını internet ortamını kullanmaya itmekte ve öğrencilere çevrimiçi imkanlar sunarak konuları pratik bir şekilde öğrenme olanaklarını sağlamaları gerekmektedir.

Okul içerisinde verilen eğitim; öğrencilerin derslerde anlamadığı konular, derste yaşanan herhangi bir olumsuzluktan dolayı dinleyemediği dersler ya da sağlık durumundan ötürü okula gidemeyip derslerinden geri kalan öğrenciler için olumsuzluklar doğurmaktadır. Alternatif bir eğitim sistemi olan ve son yıllarda bazı dönemler zorunlu durumlardan ötürü tercih edilmiş olan uzaktan eğitim ile birlikte bu tip sorunlar ortadan kalkmış, yer ve zaman sınırlaması olmadan istenildiği kadar öğrenciye hizmet verebilen sistem haline gelmiştir [9]. Uzaktan öğretimi benimseyen öğrenciler de elektronik ortamdan ders çalışmaya alışmış ve takıldığı dersler için kaynak araması yapmaya yönelmişlerdir. Fakat bazı kaynaklar öğretmenin müfredatına ya da anlatım tarzına uymamakta bu da öğrenci için olumsuzluğa sebebiyet vermektedir.

Geliştirilecek okul yönetim sistemi ile eğitim süreci daha pratik hale getirilmesi amaçlanmaktadır. Örneğin öğrenci o gün okula gidememişse derslerinden geri kalmamak adına öğretmenlerin ilgili dersler için sisteme yüklediği kaynakları görebilme ya da videoları izleyebilme imkanına sahip olabilmektedir. Bu şekilde diğer kaynaklardan konulara çalışırken ders öğretmenin anlattığı konular ile yaşanabilecek uyumsuzluğu ortadan kaldıracaktır. Ayrıca öğrenci okul yönetim sistemi ile öğretmenlerin ilgili derslere yüklediği kaynaklardan konulara çalışırken sorun yaşadığı kısımlarda öğretmenine panel üzerinden mesaj atarak iletişime geçebilecek ve böylelikle daha kullanışlı bir eğitim modeli benimsetilmeye çalışılacaktır.

6. KAYNAKLAR

- [1] *Yusuf Sezer.* (2023, 11 6). Yusuf Sezer Web sitesi: <https://www.yusufsezer.com.tr/asp-net-core/> adresinden alındı
- [2] *aws Amazon.* (2023, 11 7). aws Amazon Web sitesi: <https://aws.amazon.com/tr/what-is/net/> adresinden alındı
- [3] *Hakan Çamoğlu.* (2023, 11 7). https://www.youtube.com/watch?v=GM_c7zRvFUs adresinden alındı
- [4] *Medium.* (2023, 11 8). <https://medium.com/kodcular/katmanl%C4%B1-mimari-9fb34ef8c376> adresinden alındı
- [5] *Medium.* (2023, 11 8). Medium Web sitesi: <https://medium.com/kodluyoruz/orm-nedir-orm-ara%C3%A7lar%C4%B1-ve-yakla%C5%9F%C4%B1mlar%C4%B1-nelerdir-37af94ee873c#> adresinden alındı
- [6] *Medium.* (2023, 11 27). Medium Web sitesi: <https://medium.com/@ruveydaakcinar/asp-net-core-da-kullanman%C4%B1z-gereken-temel-k%C3%BCt%C3%BCphaneler-fluentvalidation-6cbc73a73e06> adresinden alındı
- [7] *Gençay Yıldız.* (2023, 11 29). Gençay Yıldız Web sitesi: <https://www.gencayyildiz.com/blog/asp-net-core-identity-nedir-ve-temel-kavramlar-i/> adresinden alındı
- [8] UYMAZ, S. A. (2007). *Uml ile Sanal Kampüs Modellemesi*. Bilgisayar Mühendisliği. Konya: Yüksek Lisans Tezi Selçuk Üniversitesi Fen Bilimleri Enstitüsü.
- [9] ÇEKİÇ, U. (2010). *Uzaktan Eğitim Sistemi Tasarımı*. Bilgisayar Mühendisliği. İstanbul: Yüksek Lisans Tezi İstanbul Üniversitesi Fen Bilimleri Enstitüsü.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Büşra ÖZDEMİR
Doğum Tarihi ve Yeri : 16.05.2002 - Düzce
Yabancı Dili : İngilizce
E-posta : busraozdm1@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2024
Lise	Bilişim Teknolojileri	Düzce Adnan Menderes Mesleki ve Teknik Anadolu Lisesi	2020