

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

07.08.2023

idvlabs

Asp.Net Core API Task Projesi

Several thin, curved lines in dark blue and light gray originate from the bottom left and curve upwards and to the right.

OĞUZHAN CİHAN

İçindekiler

1. Proje Konusu	2
2. Proje Mimarisi	2
3. Proje Er Diagramı	4
4. API Endpoint Tanımlamaları	5

1. Proje Konusu

Uygulama konusu öğrenci kurs işlemlerini yapmayı amaçlamaktadır. Bu anlamda bazı gereksinimler aşağıdaki gibi olacaktır:

- Öğrenci birden fazla kurs seçebilecektir.
- Öğrencinin kurs seçmesi için departman ve kurs verilerinin hazır olması gerekecektir.
- Uygulamada rol olarak Standart ve Admin roller kullanılacaktır.
- Standart role sahip kullanıcılar departman,kurs senaryolarında CRUD işlemlerinin hepsini yapabileceklerdir.
- Standart role sahip kullanıcılar öğrenci senaryolarında kullanıcı Id'sine göre listeleme yapabileceklerdir.
- Standart role sahip kullanıcılar öğrenci senaryolarında kayıt işlemi yapabileceklerdir.
- Admin role sahip kullanıcılar tüm senaryolardaki işlemlerinin hepsini yapabileceklerdir.
- Uygulamada bir kullanıcının oturum süresi 30 dk olarak çalışacaktır.
- Uygulama in-memory veri tabanı ile çalışabilecektir. Bu ayar konfigürasyon bağımlıdır.
- In-Memory çalışmak için konfigürasyon içerisinde ilgili alan 'true' yapılarak aktifleştirilebilecektir.
- Proje mimarisi olarak N-Layer kullanılacaktır.

2. Proje Mimarisi

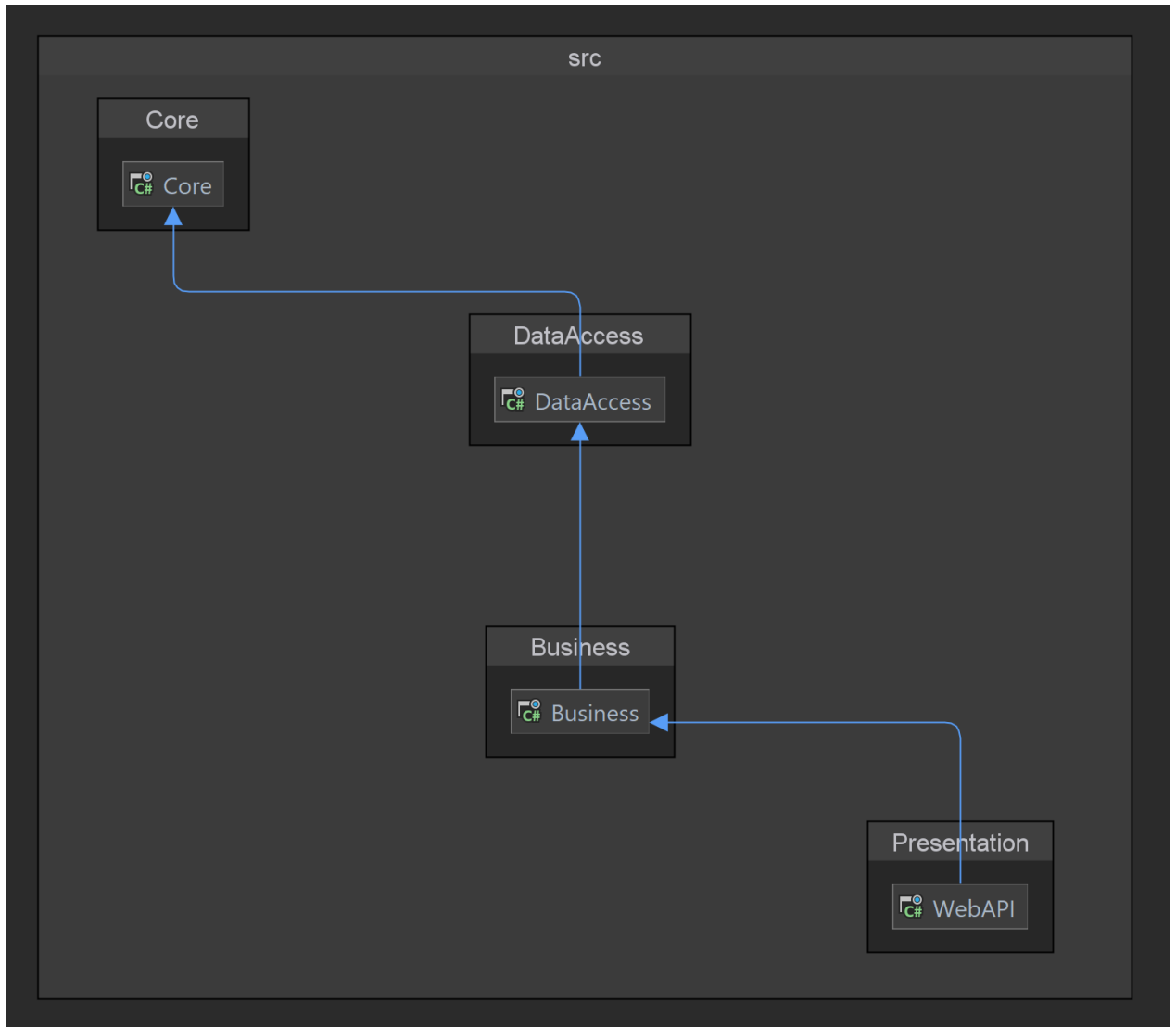
Uygulama için N-Layer mimarisi kullanıldı. Kullanılan katmanlar şu şekildedir:

- Business: İş kodlarının ve uygulama işlevselliğinin yer aldığı katmandır. Veritabanı işlemlerinden alınan verileri işlemek, iş kurallarını uygulamak ve uygulamanın ana mantığı burada yer alır.
- Core: Bu katmanda, genel iş mantığının ve bağımsız işlemleri içeren kodlar bulunur. Bu uygulamanın temel yapı taşlarının ve çekirdek işlevselliğini içerir.
- DataAccess: Bu katman, veritabanı işlemlerinin yapıldığı yerdir. Veri tabanına erişim ve veri işlemleri(CRUD işlemleri gibi) burada gerçekleştirilir.
- Presentation: Bu katman, kullanıcı arayüzünün oluşturulduğu ve uygulamanın ayağa kaldırıldığı yerdir. Kullanıcıların uygulama ile etkileşimde bulunabileceği arayüz elemanları(örneğin, web sayfaları, kullanıcı formları vb.) burada oluşturulur. Bu katmanda kullanıcı ile uygulama arasındaki etkileşimler yönetilir.

Projede çalıştıktan sonra test verileri eklenecektir. Bu test verilerinde kullanıcı girişi yapmak için bilgiler şöyle olacaktır:

- Standart kullanıcısı için:
 - Kullanıcı adı: standart
 - Şifre: 1
- Admin kullanıcısı için:
 - Kullanıcı adı: admin
 - Şifre: 1

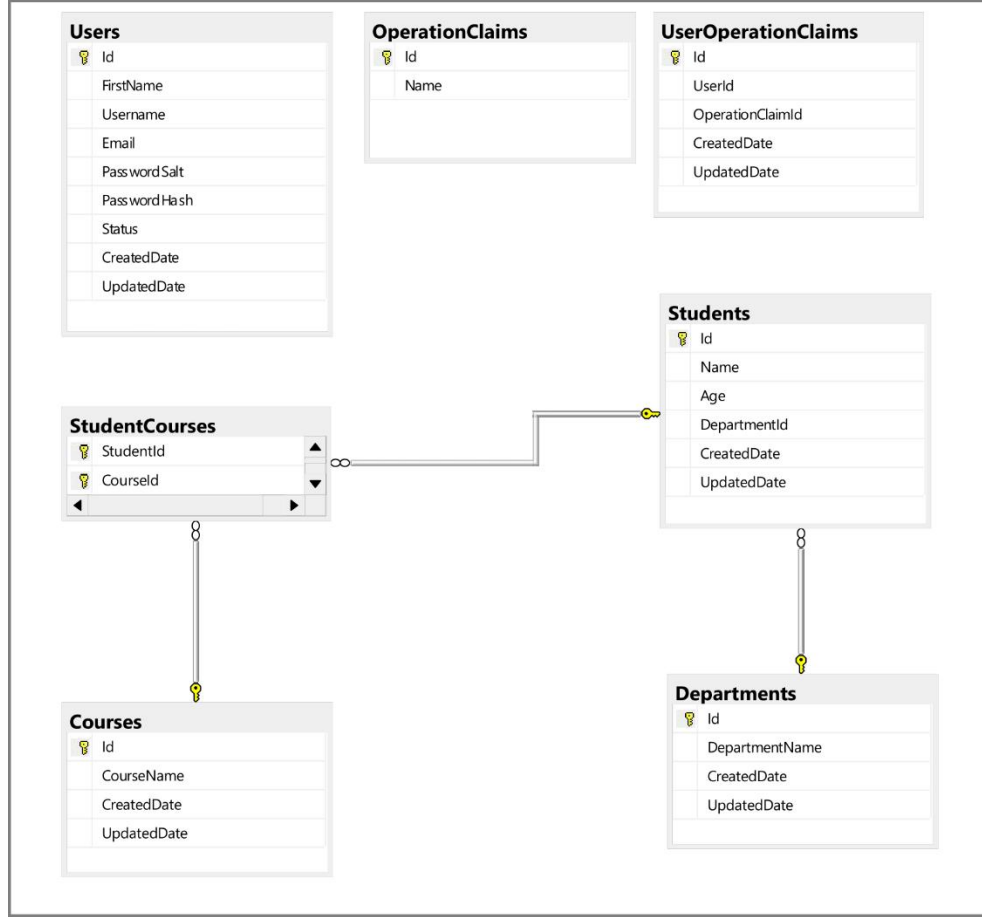
Uygulama tasarım yapısıyla ilgili örnek ekran görüntüsü Şekil 2.1’de verilmiştir.



Şekil 2.1 Uygulama Dosya Yapısı

3. Proje Er Diagramı

Projeye ait veri tabanı ilişkisellik diagramı Şekil 3.1’de verilmiştir.Öğrenci ile kurs arasında ‘Many to Many’ bir ilişki vardır. Öğrenci ile departman arasında ise ‘One to Many’ bir ilişki olacak şekilde tasarlanmıştır.



Şekil 3.1 Veri tabanı İlişkisellik Diagramı

4. API Endpoint Tanımlamaları

- Öğrenci Listesi Alma (GET /api/students)
Açıklama: Tüm öğrenci bilgilerini almak için kullanılacak endpoint.
Parametreler: Yok.
Dönen Veri: Öğrenci listesi (örneğin, JSON formatında).
- Belirli Bir Öğrenci Alma (GET /api/students/{id})
Açıklama: Belirli bir öğrencinin bilgilerini almak için kullanılacak endpoint.
Parametreler: {id} ile belirtilen öğrenci kimliği.
Dönen Veri: Belirtilen öğrenciye ait bilgiler (örneğin, JSON formatında).
- Yeni Öğrenci Ekleme (POST /api/students)
Açıklama: Yeni bir öğrenci eklemek için kullanılacak endpoint.
Gönderilen Veri: Yeni öğrencinin bilgileri (örneğin, JSON formatında).
Dönen Veri: Eklenen öğrenci bilgileri ve öğrenci kimliği (örneğin, JSON formatında).
- Öğrenci Bilgilerini Güncelleme (PUT /api/students/{id})
Açıklama: Belirli bir öğrencinin bilgilerini güncellemek için kullanılacak endpoint.
Parametreler: {id} ile belirtilen öğrenci kimliği.
Gönderilen Veri: Güncellenecek öğrencinin yeni bilgileri (örneğin, JSON formatında).
Dönen Veri: 204 HttpStatusCode.
- Öğrenci Silme (DELETE /api/students/{id})
Açıklama: Belirli bir öğrenciyi silmek için kullanılacak endpoint.
Parametreler: {id} ile belirtilen öğrenci kimliği.
Dönen Veri: 204 HttpStatusCode.

- Tüm Kursları Alma (GET /api/courses)
Açıklama: Tüm kurs bilgilerini almak için kullanılacak endpoint.
Parametreler: Yok.
Dönen Veri: Kurs listesi (örneğin, JSON formatında).
- Yeni Kurs Ekleme (POST /api/courses)
Açıklama: Yeni bir kurs eklemek için kullanılacak endpoint.
Gönderilen Veri: Yeni kursun bilgileri (örneğin, JSON formatında).
Dönen Veri: Eklenen kursun bilgileri ve kurs kimliği (örneğin, JSON formatında).
- Kurs Bilgilerini Güncelleme (PUT /api/courses/{id})
Açıklama: Belirli bir kursun bilgilerini güncellemek için kullanılacak endpoint.
Parametreler: {id} ile belirtilen kurs kimliği.
Gönderilen Veri: Güncellenecek kursun yeni bilgileri (örneğin, JSON formatında).
Dönen Veri: 204 HttpStatusCode.
- Kurs Silme (DELETE /api/courses/{id})
Açıklama: Belirli bir kursu silmek için kullanılacak endpoint.
Parametreler: {id} ile belirtilen kurs kimliği.
Dönen Veri: 204 HttpStatusCode.
- Departman Listesi Alma (GET /api/departments)
Açıklama: Tüm departman bilgilerini almak için kullanılacak endpoint.
Parametreler: Yok.
Dönen Veri: Departman listesi (örneğin, JSON formatında).
- Belirli Bir Departman Alma (GET /api/departments/{id})
Açıklama: Belirli bir departmanın bilgilerini almak için kullanılacak endpoint.
Parametreler: {id} ile belirtilen departman kimliği.
Dönen Veri: Belirtilen departmanın bilgileri (örneğin, JSON formatında).

- Yeni Departman Ekleme (POST /api/departments)
Açıklama: Yeni bir departman eklemek için kullanılacak endpoint.
Gönderilen Veri: Yeni departmanın bilgileri (örneğin, JSON formatında).
Dönen Veri: Eklenen departmanın bilgileri ve departman kimliği (örneğin, JSON formatında).
- Departman Bilgilerini Güncelleme (PUT /api/departments/{id})
Açıklama: Belirli bir departmanın bilgilerini güncellemek için kullanılacak endpoint.
Parametreler: {id} ile belirtilen departman kimliği.
Gönderilen Veri: Güncellenecek departmanın yeni bilgileri (örneğin, JSON formatında).
Dönen Veri: 204 HttpStatusCode.
- Departman Silme (DELETE /api/departments/{id})
Açıklama: Belirli bir departmanı silmek için kullanılacak endpoint.
Parametreler: {id} ile belirtilen departman kimliği.
Dönen Veri: 204 HttpStatusCode.
- Kullanıcı Kayıt (POST /api/users/register)
Açıklama: Yeni bir kullanıcı kaydı oluşturmak için kullanılacak endpoint.
Gönderilen Veri: Yeni kullanıcının kayıt bilgileri
Dönen Veri: Kullanıcı girişi için token.
- Kullanıcı Giriş (POST /api/users/login)
Açıklama: Kullanıcı girişi yapmak için kullanılacak endpoint.
Gönderilen Veri: Kullanıcının giriş bilgileri (örneğin, kullanıcı adı veya e-posta ve parola).
Dönen Veri: Kullanıcı girişi için token.