

# STAJ TAKİP OTOMASYONU

## Bileşenler

- Programlama Dili : C#
- Ide (Geliştirme Ortamı) : Visual Studio 2015 Pro
- Veri Tabanı : Access 2010-2003

## Yardımcı Araçlar

- Diyagram Çizimi : Draw IO

## Formlar

- Form\_Anasayfa
- Form\_Baglanti
- Form\_Giris
- Form\_IsyeriYonetim
- Form\_OgrenciYonetim
- Form\_OgretmenYonetim
- Form\_Staj
- Form\_VeliYonetim

## Diğer Bileşenler

- Veritabanıİslemleri.cs (class)
- Ayar\_Veritabani.settings (settings)
- Resources (resources)

## Amaç

Staj programına tabi tutulan öğrencilerin takibi işlemlerini otomatize etmek.

## Aşamalar

- Veritabanı Tasarımı
- Veritabanı Bağlantısı
- Sql Sorguları
- Form Dizaynları
- Kodlama

## 1- Veritabanı Tasarımı

Veritabanı ön tasarımı için **Draw IO** adında bir diyagram tasarımcısı kullanılmıştır. Bu sayede **Ms Access** üzerinde işlem yapmadan önce tüm aşamalar gözden geçirilerek sorunlar giderilmiştir.

Ms Access üzerinde veritabanı için tablolar hazırlandıktan sonra ilişkilendirmeleri yapılmıştır.

## 2- Veritabanı Bağlantısı

**Dinamik**(değişken) veritabanı bağlantısı için bir adet **Window Form** oluşturulup gerekli kodlamaları yapılarak projenin taşınması veya veritabanının yerinin değiştirilmesi, yeni veritabanı eklenmesi gibi durumların oluşmasına karşın veritabanı bağlantısı sağlanmış olup gerekli veri **Ayar\_Veritabanı.settings** dosyası içerisinde saklanmıştır.

## 3- Sql Sorguları

Tablolar üzerinde yapılacak sql sorguları için belirli kalıplar oluşturulmuştur. Bu kalıplara örnek verecek olursak Ekleme,Silme,Güncelleme,Arama gibi sorgular her tablo için kalıplaştırılmıştır. Bu sayede kodlama hızı arttırılarak proje süresi kısaltılmıştır. Tüm SQL sorguları tek bir Class(sınıf) olan Veritabanıİşlemleri.cs içerisinde toplanarak düzen sağlanmış olmakla beraber kod tekrarından sakınılmış ve nesne yönelimli yapı oluşturulmuştur. Tek bir class üzerinden veritabanı işlemleri gerçekleştirilerek kod karmaşasından arındırılmış ve tablolar için organizasyon sağlanmıştır.

## 4- Form Dizaynları

Benzer işlemlerin gerçekleştirildiği formların daha hızlı tasarlanması için birçok öge ortak tasarlanmış olup veritabanı işlemleri için kullanılan **dataGridView** nesnelerinin özellikleri eşitlenmiştir. Form nesnelerinin **Dosya Gezgini**nde ayırt edilmeleri için başına **“Form”** ibaresi eklenmiştir. Form tasarımlarında sadeliğe önem verilip hem performans artışı sağlanmış hem de karmaşadan arındırılmıştır.

## 5- Kodlama

Kodlama aşamasında olabildiğinde fonksiyonlar kullanılarak kod tekrarlarının önüne geçilmiş olmakla beraber nesne yönelimli olan **C#** programlama dilinin önemi vurgulanmıştır. Tek bir **class** üzerinde toplanan veritabanı işlemleri pratik olarak gerçekleştirilmiştir. Bazı bölümlerde yorum satırları eklenerek kod karmaşasının önüne geçilmiş ve düzen sağlanmıştır. Oluşabilecek hata durumlarına karşı **try-cath** blokları kullanılmış ve hatalar neticesinde programın kapanması önüne geçilerek hata tespiti kolaylaştırılmıştır. **Form** nesnelerinin üzerinde toplu işlemler yapılması gerektiğinde tek tek yapmak yerine döngüler ile nesneler üzerinde gezilerek özelliklerine göre işlemler yapılmıştır. **SQL** sorguları için fonksiyonlar oluşturularak kod tekrarlarından kaçınılmış ve fonksiyonlar geriye değer döndüren türden yapılarak kısa kodlama sağlanmıştır.