T.C.

KIRIKKALE ÜNİVERSİTESİ

MESLEK YÜKSEKOKULU

BİLGİSAYAR PROGRAMCILIĞI

SİSTEM ANALİZİ ve TASARIMI DERSİ

SPOR MERKEZİ PROGRAMI PROJESİ

Dersin Öğretim Elemanı

Dr. Öğr. Üyesi Aşır ÖZBEK

ZEYNEL ŞAHİN

180705009

KIRIKKALE-2020

**İÇİNDEKİLER**

[1. GİRİŞ 1](#_Toc42602181)

[2. VERİTABANI TASARIMI 2](#_Toc42602182)

[2.1. Gereklilik Analizi 2](#_Toc42602183)

[2.2. Kavramsal Tasarım 2](#_Toc42602184)

[2.2.1. Üyeler Varlık Kümesi 3](#_Toc42602185)

[2.2.2. Kullanıcılar Varlık Kümesi 4](#_Toc42602186)

[2.2.3. Sınıflar Varlık Kümesi 5](#_Toc42602187)

[2.2.4. Satışlar Varlık Kümesi 6](#_Toc42602188)

[2.2.5. Tahsilatlar Varlık Kümesi 7](#_Toc42602189)

[2.2.6. Vücut Ölçüleri Varlık Kümesi 8](#_Toc42602190)

[2.2.7. Ölçü Türleri Varlık Kümesi 9](#_Toc42602191)

[2.3. Mantıksal Tasarım 10](#_Toc42602192)

[2.3.1. Üyeler\_Sınıflar İlişkisi 14](#_Toc42602193)

[2.3.2. Üyeler\_Ölçü Türleri İlişkisi 15](#_Toc42602194)

[2.3.3. Üyeler\_Kullanıcılar İlişkisi 16](#_Toc42602195)

[2.3.4. Genel Gösterim 17](#_Toc42602196)

[2.4. Fiziksel Tasarım 18](#_Toc42602197)

[2.4.1. uyeler Tablosu: 19](#_Toc42602198)

[2.4.2. satislar Tablosu 20](#_Toc42602199)

[2.4.3. siniflar Tablosu 21](#_Toc42602200)

[2.4.4. vucut\_olculeri Tablosu 22](#_Toc42602201)

[2.4.5. olcu\_turleri Tablosu 23](#_Toc42602202)

[2.4.6. tahsilatlar Tablosu 24](#_Toc42602203)

[2.4.7. kullanicilar Tablosu 24](#_Toc42602204)

[2.4.8. Veri Tabanı Diyagramı 25](#_Toc42602205)

[2.4.9. Tabloların SQL Kodları 26](#_Toc42602206)

[2.5. Programın Akış Şeması 29](#_Toc42602207)

[2.5.1. Giriş Ekranı Akış Şeması 29](#_Toc42602208)

[2.5.2. Arama Akış Şeması 30](#_Toc42602209)

[2.5.3. Ekleme Akış Şeması 31](#_Toc42602210)

[2.5.4. Silme İşlemi Akış Diyagramı 32](#_Toc42602211)

[2.5.5. Güncelleme İşlemi Akış Diyagramı 33](#_Toc42602212)

[2.5.6. Üye Bilgilerinin Görüntüleme Akış Diyagramı 34](#_Toc42602213)

[2.5.7. Kullanıcı Bilgilerinin Görüntüleme Akış Diyagramı 35](#_Toc42602214)

[2.6. Proje Uygulama Kısmı 36](#_Toc42602215)

[2.6.1. Spor Merkezi Otomasyonu Giriş Ekranı 36](#_Toc42602216)

[2.6.2. Spor Merkezi Otomasyonu Ana Sayfası 36](#_Toc42602217)

[2.7. Projenin Kodları 58](#_Toc42602218)

[2.7.1. Giriş Ekranı Kodları 58](#_Toc42602219)

[2.7.2. Ana Sayfa Kodları 59](#_Toc42602220)

[2.7.3. Ölçü Türü Tanımlamaları 64](#_Toc42602221)

[2.7.4. Kullanıcı Tanımlamaları 72](#_Toc42602222)

[2.7.5. Sınıflar 78](#_Toc42602223)

[2.7.6. Ön Kayıt 84](#_Toc42602224)

[2.7.7. Üye Ara 85](#_Toc42602225)

[2.7.8. Üye Detayları 88](#_Toc42602226)

[2.7.9. Çöp Kutusu 98](#_Toc42602227)

[2.7.10. Silinecekler 102](#_Toc42602228)

[KAYNAKÇA 105](#_Toc42602229)

**ŞEKİL LİSTESİ**

[**Şekil 1**: Üye Varlık Kümesi 9](#_Toc38133423)

[**Şekil 2**: Kullanıcılar Varlık Kümesi 10](#_Toc38133424)

[**Şekil 3**: Sınıflar Varlık Kümesi 11](#_Toc38133425)

[**Şekil 4**: Satışlar Varlık Kümesi 12](#_Toc38133426)

[**Şekil 5**: Tahsilatlar Varlık Kümesi 13](#_Toc38133427)

[**Şekil 6**: Vücut Ölçüleri Varlık Kümesi 14](#_Toc38133428)

[**Şekil 7**: Ölçü Türleri Varlık Kümesi 15](#_Toc38133429)

[**Şekil 8**: Üyeler\_Sınıflar Türleri İlişkisi 19](#_Toc38133430)

[**Şekil 9**: Üyeler\_Ölçü Türleri İlişkisi 20](#_Toc38133431)

[**Şekil 10**: Üyeler\_ Kullanıcılar İlişkisi 21](#_Toc38133432)

**TABLO LİSTESİ**

[**Tablo 1**: Üyeler Tablosu 10](#_Toc42602230)

[**Tablo 2**: Kullanıcılar Tablosu 11](#_Toc42602231)

[**Tablo 3**: Sınıflar Tablosu 12](#_Toc42602232)

[**Tablo 4**: Satışlar Tablosu 12](#_Toc42602233)

[**Tablo 5**: Tahsilatlar Tablosu 13](#_Toc42602234)

[**Tablo 6**: Vücut Ölçüleri Tablosu 13](#_Toc42602235)

[**Tablo 7**: Ölçü Türleri Tablosu 14](#_Toc42602236)

[**Tablo 8**: uyeler Tablosu 20](#_Toc42602237)

[**Tablo 9**: satislar Tablosu 21](#_Toc42602238)

[**Tablo 10**: siniflar Tablosu 22](#_Toc42602239)

[**Tablo 11**: vücut\_olculeri Tablosu 23](#_Toc42602240)

[**Tablo 12**: olcu\_turleri Tablosu 23](#_Toc42602241)

[**Tablo 13**: tahsilatlar Tablosu 24](#_Toc42602242)

[**Tablo 14**: kullanicilar Tablosu 25](#_Toc42602243)

[**Tablo 15**: Veritabanı Diyagramı 25](#_Toc42602244)

[**Tablo 16:**Giriş Ekranı 36](#_Toc42602245)

[**Tablo 17:** Ana Sayfa 37](#_Toc42602246)

[**Tablo 18:** Ölçü Türü Tanımlamaları 38](#_Toc42602247)

[**Tablo 19:** Ölçü Türü Arama Sonucu 38](#_Toc42602248)

[**Tablo 20:** Çöp Kutusu Ölçü Türü 39](#_Toc42602249)

[**Tablo 21:** Çöp Kutusu Ölçü Türü (Yakından) 39](#_Toc42602250)

[**Tablo 22:** Silinecekler Ölçü Türü 40](#_Toc42602251)

[**Tablo 23:** Silinecekler Ölçü Türü (Yakından) 40](#_Toc42602252)

[**Tablo 24:** Ölçü Türü Güncelle 41](#_Toc42602253)

[**Tablo 25:** Kullanıcı Tanımlamaları 42](#_Toc42602254)

[**Tablo 26:** Kullanıcı Arama Sonucu 42](#_Toc42602255)

[**Tablo 27:** Çöp Kutusu Kullanıcı 43](#_Toc42602256)

[**Tablo 28:** Silinecekler Kullanıcı 43](#_Toc42602257)

[**Tablo 29:** Kullanıcı Güncelle 44](#_Toc42602258)

[**Tablo 30:** Sınıflar 45](#_Toc42602259)

[**Tablo 31:** Sınıf Arama Sonucu 45](#_Toc42602260)

[**Tablo 32:** Çöp Kutusu Sınıf 46](#_Toc42602261)

[**Tablo 33:** Silinecekler Sınıf 46](#_Toc42602262)

[**Tablo 34:** Sınıf Güncelle 47](#_Toc42602263)

[**Tablo 35:** Ön Kayıt 48](#_Toc42602264)

[**Tablo 36:** Üye Ara 49](#_Toc42602265)

[**Tablo 37:** Üye Arama Sonucu 50](#_Toc42602266)

[**Tablo 38:** Çöp Kutusu Üye 50](#_Toc42602267)

[**Tablo 39:** Silinecekler Üye 51](#_Toc42602268)

[**Tablo 40:** Üye Detayları 52](#_Toc42602269)

[**Tablo 41:** Üye Sil 52](#_Toc42602270)

[**Tablo 42:** Vücut Ölçüleri 53](#_Toc42602271)

[**Tablo 43:** Vücut Ölçüleri Arama Sonucu 54](#_Toc42602272)

[**Tablo 44:** Satış 55](#_Toc42602273)

[**Tablo 45:** Satış İşlemi 55](#_Toc42602274)

[**Tablo 46:** Tahsilatlar 56](#_Toc42602275)

[**Tablo 47:** Oturum Kapatma 57](#_Toc42602276)

[**Tablo 48:** Kullanıcı Bilgileri 58](#_Toc42602277)

# GİRİŞ

Spor merkezi otomasyonun amacı; spor merkezlerinde spor yapan üyelerin bilgilerinin kayıt altına almak, gerektiğinde verilere hızlı ve kullanıcı giriş sistemi ile güvenli bir şekilde ulaşabilmek, üyelerin kayıtlı oldukları sınıfları, satış işlemlerini ve tahsilat işlemlerinin takibini sağlamaktır.

# VERİTABANI TASARIMI

## Gereklilik Analizi

Program gereksinimleri ihtiyaçları yapılan araştırma sonucunda aşağıda belirtildiği şekilde tespit edilmiştir.

* Üyelerin kayıt altına alınması,
* Personellerin kayıt alınması,
* Üye ve personel giriş çıkış bilgilerinin kayıt altına alınması,
* İşletme (spor salonu) bilgilerinin kayıt altına alınması,
* Spor salonu üyelik ve hizmet bilgilerinin kayıt altına alınması,
* Yönetici bilgilerinin kayıt altına alınması,
* Spor salonu sınıf bilgilerinin kayıt altına alınması,
* Satış işlemlerinin kayıt altına alınması,
* Tahsilat işlemlerinin kayıt altına alınması,
* Uyarı, mesaj, duyuru sisteminin oluşturulması.

## Kavramsal Tasarım

Kavramsal tasarım aşağıdaki ihtiyaçların giderilmesi amacıyla tasarlanmıştır.

* Spor salonu programı yöneticisi ve yetkilendirdiği kişiler tarafından kullanılması,
* Üyelerin ve bilgilerinin anlaşılır bir şekilde listelenmesi,
* Üye bilgilerinin kolay ve hızlı girilebilmesini sağlamak,
* Üye bilgilerinin kayıt altına alınabilmesi,
* Üye bilgilerinin takibini sağlanması,
* Üye ders programının anlık olarak gösterilebilmesi,
* Üye sınıf bilgilerinin kayıt altına alınabilmesi,
* Üye vücut ölçüleri, satış, tahsilat bilgilerinin takibinin yapılması,
* Tanımlamaların yapılabilmesi (ölçü türü, hizmet),
* Her türlü bilgilere ait aktiflik, ekleme, düzeltme, güncelleme işlemleri yapılması,
* Üye duyurularının kaydedilebilmesi ve takip edilebilmesi,
* Veritabanı güvenilirliğinin sağlanması,
* Tasarımın gözü yormaması ve kullanıcı dostu olması amaçlanmıştır.

Bu çerçevede; gereklilik analizinden yola çıkılarak yapılan çalışma sonucunda aşağıdaki varlıkların planlanması gerekli bulunmuş planlama da buna yönelik olarak yapılmıştır. Buna göre Varlık kümelerinin aşağıdaki maddelerden oluşması planlanmıştır.

* Üyeler Varlık Kümesi
* Kullanıcılar Varlık Kümesi
* Sınıflar Varlık Kümesi
* Satışlar Varlık Kümesi
* Tahsilatlar Varlık Kümesi
* Vücut Ölçüleri Varlık Kümesi
* Ölçü Türleri Varlık Kümesi

### Üyeler Varlık Kümesi

Üye Varlık Kümesi tasarlanırken bir bu varlık kümesine ait nitelikler diğer varlıklar arasındaki ilişkileri belirleyen nitelikler ile birlikte tasarlanmıştır.

Buna göre Üyeler Varlık Kümesi;

* Üye İd,
* T.C. No,
* Adı,
* Soyadı,
* Cinsiyet,
* Doğum Tarihi,
* İl,
* İlçe,
* Adres,
* Silme Durumu,
* Telefon,
* Telefon2,
* E Posta,
* Kan Grubu,
* Resim nitelikleri olarak tasarlanmıştır.

Buna göre Üye Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Üye İd** anahtar nitelik olarak tanımlanmıştır.

ÜYELER

**Şekil 1**: Üye Varlık Kümesi

### Kullanıcılar Varlık Kümesi

Kullanıcılar Varlık Kümesi tasarlanırken bir bu varlık kümesine ait nitelikler diğer varlıklar arasındaki ilişkileri belirleyen nitelikler ile birlikte tasarlanmıştır.

Buna göre Kullanıcılar Varlık Kümesi;

* Kullanıcı İd,
* Kullanıcı Adı,
* Şifre,
* Adı,
* Soyadı
* Silme Durumu nitelikleri olarak tasarlanmıştır.

Buna göre Danışman Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. **Kullanıcı İd** anahtar nitelik olarak tanımlanmıştır.

KULLANICILAR

**Şekil 2**: Kullanıcılar Varlık Kümesi

### Sınıflar Varlık Kümesi

Sınıflar Varlık Kümesi tasarlanırken bir bu varlık kümesine ait nitelikler diğer varlıklar arasındaki ilişkileri belirleyen nitelikler ile birlikte tasarlanmıştır.

Buna göre Sınıflar Varlık Kümesi;

* Sınıf İd,
* Adı,
* Kapasite,
* Durum
* Fiyatı
* Silme Durumu nitelikleri olarak tasarlanmıştır.

Buna göre Öğrenci Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Sınıf İd** anahtar nitelik olarak tanımlanmıştır.

SINIFLAR

**Şekil 3**: Sınıflar Varlık Kümesi

### Satışlar Varlık Kümesi

Satışlar Varlık Kümesi tasarlanırken bir bu varlık kümesine ait nitelikler diğer varlıklar arasındaki ilişkileri belirleyen nitelikler ile birlikte tasarlanmıştır.

Buna göre Satışlar Varlık Kümesi;

* Satış İd,
* Başlangıç Tarihi,
* Bitiş Tarihi,
* Satış Fiyatı,
* Uye,
* Sınıf
* Durum nitelikleri olarak tasarlanmıştır.

Buna göre Öğrenci Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Satış İd** anahtar nitelik olarak tanımlanmıştır.

SATIŞLAR

**Şekil 4**: Satışlar Varlık Kümesi

### Tahsilatlar Varlık Kümesi

Tahsilatlar Varlık Kümesi tasarlanırken bir bu varlık kümesine ait nitelikler diğer varlıklar arasındaki ilişkileri belirleyen nitelikler ile birlikte tasarlanmıştır.

Buna göre Tahsilatlar Varlık Kümesi;

* Tahsilat İd,
* Tarih,
* Saat,
* Tutar,
* Ödeme Türü Adı,
* Uye,
* Kullanıcı
* Durum nitelikleri olarak tasarlanmıştır.

Buna göre Öğrenci Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Tahsilat İd** anahtar nitelik olarak tanımlanmıştır.

TAHSİLATLAR

**Şekil 5**: Tahsilatlar Varlık Kümesi

### Vücut Ölçüleri Varlık Kümesi

Vücut Ölçüleri Varlık Kümesi tasarlanırken bir bu varlık kümesine ait nitelikler diğer varlıklar arasındaki ilişkileri belirleyen nitelikler ile birlikte tasarlanmıştır.

Buna göre Vücut Ölçüleri Varlık Kümesi;

* Vücut Ölçü İd,
* Tarih,
* Değer,
* Ölçü Türü,
* Üye nitelikleri olarak tasarlanmıştır.

Buna göre Öğrenci Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Vücut Ölçü İd** anahtar nitelik olarak tanımlanmıştır.

VÜCUT ÖLÇÜLERİ

**Şekil 6**: Vücut Ölçüleri Varlık Kümesi

### Ölçü Türleri Varlık Kümesi

Ölçü Türleri Varlık Kümesi tasarlanırken bir bu varlık kümesine ait nitelikler diğer varlıklar arasındaki ilişkileri belirleyen nitelikler ile birlikte tasarlanmıştır.

Buna göre Ölçü Türleri Varlık Kümesi;

* Ölçü Türü İd,
* Adı,
* Birimi,
* Durum
* Silme Durumu nitelikleri olarak tasarlanmıştır.

Buna göre Öğrenci Varlık Kümesi aşağıdaki şekilde tasarlanmıştır. Tasarımda **Ölçü Türleri İd** anahtar nitelik olarak tanımlanmıştır.

VÜCUT ÖLÇÜLERİ

**Şekil 7**: Ölçü Türleri Varlık Kümesi

## Mantıksal Tasarım

Spor merkezi veri tabanı tasarımında varlık kümeleri arasındaki ilişkilerin tasarlanmasından sonra mantıksal tasarım aşamasına geçilmiştir. Bu aşamada

**Üyeler Tablosu;**

* Üye İd Birincil Anahtar,
* Şubeler tablosuna bağlantı için Şube İd Yabancı Anahtar,

Bu durumda Üyeler Tablosu aşağıdaki şekilde tasarlanmıştır.

**Tablo 1**: Üyeler Tablosu

|  |
| --- |
| Üye İd |
| T.C. No |
| Adı |
| Soyadı |
| Cinsiyet |
| Doğum Tarihi |
| İl |
| İlçe |
| Adres |
| Telefon |
| Telefon2 |
| E Posta |
| Kan Grubu |
| Resim |
| Silme Durumu |

**Kullanıcılar Tablosu;**

* Kullanıcı İd Birincil Anahtar,

Bu durumda Kullanıcılar Tablosu aşağıdaki şekilde tasarlanmıştır.

**Tablo 2**: Kullanıcılar Tablosu

|  |
| --- |
| Kullanıcı İd |
| Kullanıcı Adı |
| Şifre |
| Adı |
| Soyadı |
| Silme Durumu |

**Sınıflar Tablosu;**

* Sınıf İd Birincil Anahtar,

Bu durumda Sınıflar Tablosu aşağıdaki şekilde tasarlanmıştır

**Tablo 3**: Sınıflar Tablosu

|  |
| --- |
| Sınıf İd |
| Adı |
| Kapasite |
| Durum |
| Fiyatı |
| Silme Durumu |

**Satışlar Tablosu;**

* Satış İd Birincil Anahtar,
* Üyeler Tablosuna bağlantı için Üye İd Yabancı Anahtar
* Sınıflar Tablosuna bağlantı için Sınıf İd Yabancı Anahtar

Bu durumda Satışlar Tablosu aşağıdaki şekilde tasarlanmıştı

**Tablo 4**: Satışlar Tablosu

|  |
| --- |
| Satış İd |
| Başlangıç Tarihi |
| Bitiş Tarihi |
| Satış Fiyatı |
| Üye İd |
| Sınıf İd |
| Durum |

**Tahsilatlar Tablosu;**

* Tahsilat İd Birincil Anahtar,
* Üyeler Tablosuna bağlantı için Üye İd Yabancı Anahtar
* Kullanıcılar Tablosuna bağlantı için Kullanıcı İd Yabancı Anahtar

Bu durumda Tahsilatlar Tablosu aşağıdaki şekilde tasarlanmıştı

**Tablo 5**: Tahsilatlar Tablosu

|  |
| --- |
| Tahsilat İd |
| Tarih |
| Saat |
| Tutar |
| Ödeme Türü Adı |
| Üye İd |
| Kullanıcı İd |
| Silme Durumu |

**Vücut Ölçüleri Tablosu;**

* Vücut Ölçü İd Birincil Anahtar,
* Üyeler Tablosuna bağlantı için Üye İd Yabancı Anahtar
* Ölçü Türleri Tablosuna bağlantı için Ölçü Türü İd Yabancı Anahtar

Bu durumda Vücut Ölçüleri Tablosu aşağıdaki şekilde tasarlanmıştı

**Tablo 6**: Vücut Ölçüleri Tablosu

|  |
| --- |
| Vücut Ölçü İd |
| Tarih |
| Değer |
| Ölçü Türü İd |
| Üye İd |

**Ölçü Türleri Tablosu;**

* Ölçü Türü İd Birincil Anahtar,

Bu durumda Ölçü Türleri Tablosu aşağıdaki şekilde tasarlanmıştır

**Tablo 7**: Ölçü Türleri Tablosu

|  |
| --- |
| Ölçü Türü İd |
| Adı |
| Birimi |
| Durum |
| Silme Durumu |

### Üyeler\_Sınıflar İlişkisi

Bir üye birden fazla sınıfa kayıt yaptırabilirken, bir sınıfa birden fazla üye kayıt yaptırabilir. Bu nedenle Üyeler\_Sınıflar varlık kümeleri arasındaki ilişki n:m ilişkidir. Üyeler\_Sınıflar varlık kümeleri arasındaki ilişkinin er şeması şekil 8’de gösterilmiştir

m

n

ÜYELER

SINIFLAR

OLUR

**Şekil 8**: Üyeler\_Sınıflar Türleri İlişkisi

### Üyeler\_Ölçü Türleri İlişkisi

Bir üye birden çök ölçü türünden ölçüm yapabilirken, bir ölçü türünden birden çok üye ölçüm yapar. Bu nedenle Üyeler\_Ölçü Türleri varlık kümeleri arasındaki ilişki n:m ilişkidir. Üyeler\_Ölçü Türleri varlık kümeleri arasındaki ilişkinin er şeması şekil 9’da gösterilmiştir.

m

n

ÜYELER

ÖLÇÜ TÜRLERİ

OLUR

**Şekil 9**: Üyeler\_Ölçü Türleri İlişkisi

### Üyeler\_Kullanıcılar İlişkisi

Bir üye birden fazla kullanıcıya tahsilat verebilirken, bir kullanıcıya birden fazla üye tahsilat verebilir. Bu nedenle Üyeler\_Kullanıcılar varlık kümeleri arasındaki ilişki n:m ilişkidir. Üyeler\_Kullanıcılar varlık kümeleri arasındaki ilişkinin er şeması şekil 10’da gösterilmiştir.

m

n

ÜYELER

KULLANICILAR

OLUR

**Şekil 10**: Üyeler\_ Kullanıcılar İlişkisi

### Genel Gösterim

Varlık kümelerinin ve kümeler arasındaki var olan ilişkilerin birlikte gösterilmesi şekil 11’de verilmiştir.

ÜYELER

ALIR

SATIŞLAR

1

n

OLUR

SINIFLAR

n

1

OLUR

VÜCUT ÖLÇÜLERİ

ÖLÇÜ TÜRLERİ

OLUR

n

1

1

n

OLUR

TAHSİLATLAR

ALIR

KULLANICILAR

1

n

n

1

**Şekil 11:** Genel Gösterim

## Fiziksel Tasarım

Proje MSSQL programı ile tasarlanmıştır. Üye Kayıt tablosu tüm tabloların bağlı olduğu ana tablodur. Projede toplamda 7 adet tablo bulunmaktadır. Bulunan tablo adları aşağıda listelenmiştir.

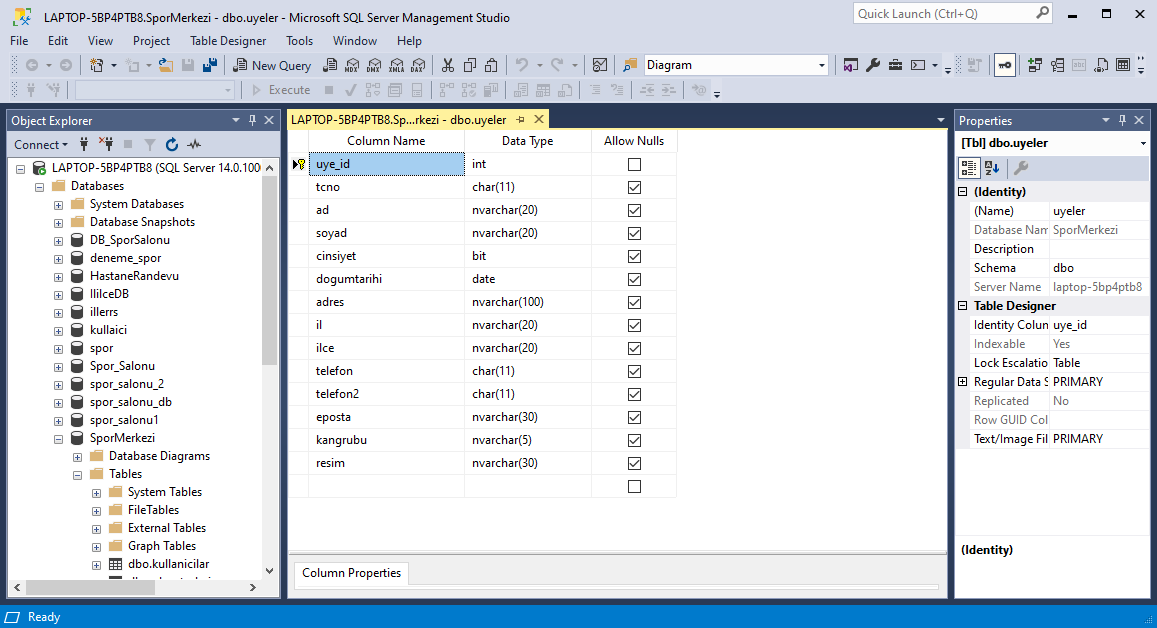
* **uyeler**: Üye bilgilerinin kayıt altına alındığı tablo adıdır.
* **satislar**: Satış bilgilerinin kayıt altına alındığı tablo adıdır.
* **siniflar**: Sınıf bilgilerinin kayıt altına alındığı tablo adıdır.
* **vucut\_olculeri**: Vücut ölçülerinin kayıt altına alındığı tablo adıdır.
* **olcu\_turleri**: Ölçü türlerinin kayıt altına alındığı tablo adıdır.
* **tahsilatlar**: Tahsilat bilgilerinin kayıt altına alındığı tablo adıdır.
* **kullanicilar**: Kullanıcı bilgilerinin kayıt altına alındığı tablo adıdır.

### uyeler Tablosu:

Üyeler Tablosu Alanları değişkenleri aşağıda belirtildiği şekilde tespit edilmiştir.

* uye\_id, int veri tipi,
* tcno, char(11) veri tipi,
* ad, nvarchar(20) veri tipi,
* soyad, nvarchar(20) veri tipi,
* cinsiyet, bit veri tipi,
* dogumtarihi, date veri tipi,
* adres, nvarchar(100) veri tipi,
* il, nvarchar(20) veri tipi,
* ilce, nvarchar(20) veri tipi,
* telefon, char(11) veri tipi,
* telefon2, char(11) veri tipi,
* eposta, nvarchar(30) veri tipi,
* kangrubu, nvarchar(5) veri tipi,
* resim, nvarchar(30) veri tipi
* silme\_durumu, bit veri tipi olarak tasarlanmıştır.

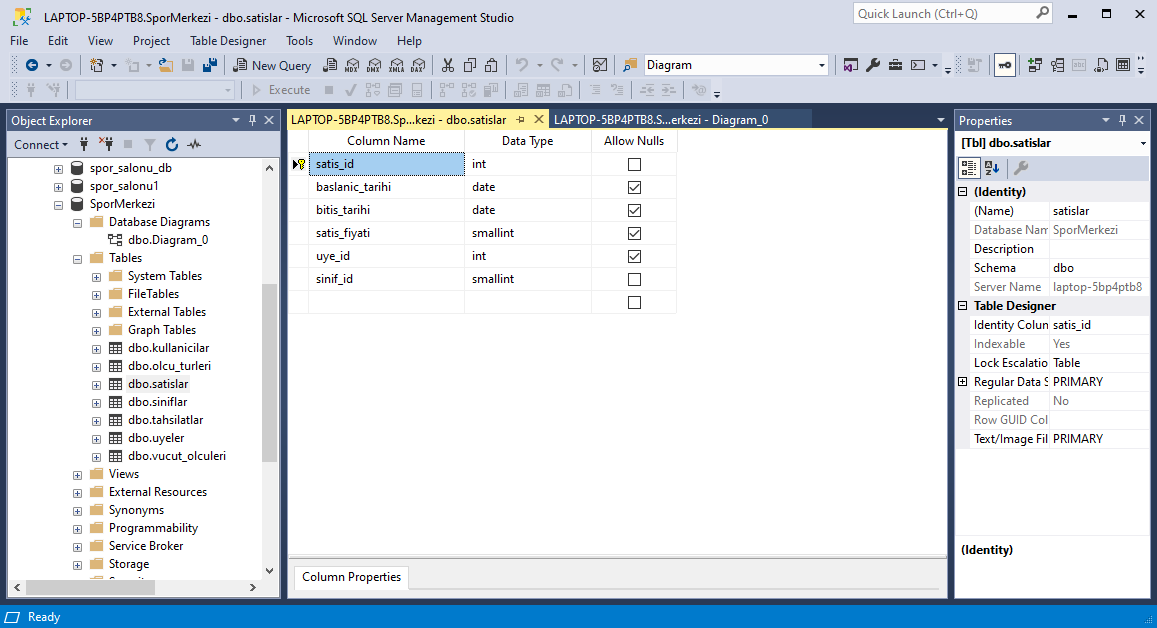
**Tablo 8**: uyeler Tablosu



### satislar Tablosu

* satis\_id, int veri tipi,
* baslanic\_tarihi, date veri tipi,
* bitis\_tarihi, date veri tipi,
* satis\_fiyati, smallint veri tipi,
* uye\_id, int veri tipi ikincil anahtar,
* sinif\_id, smallint veri tip,
* durum, bit veri tipi ikincil anahtar olarak tasarlanmıştır.

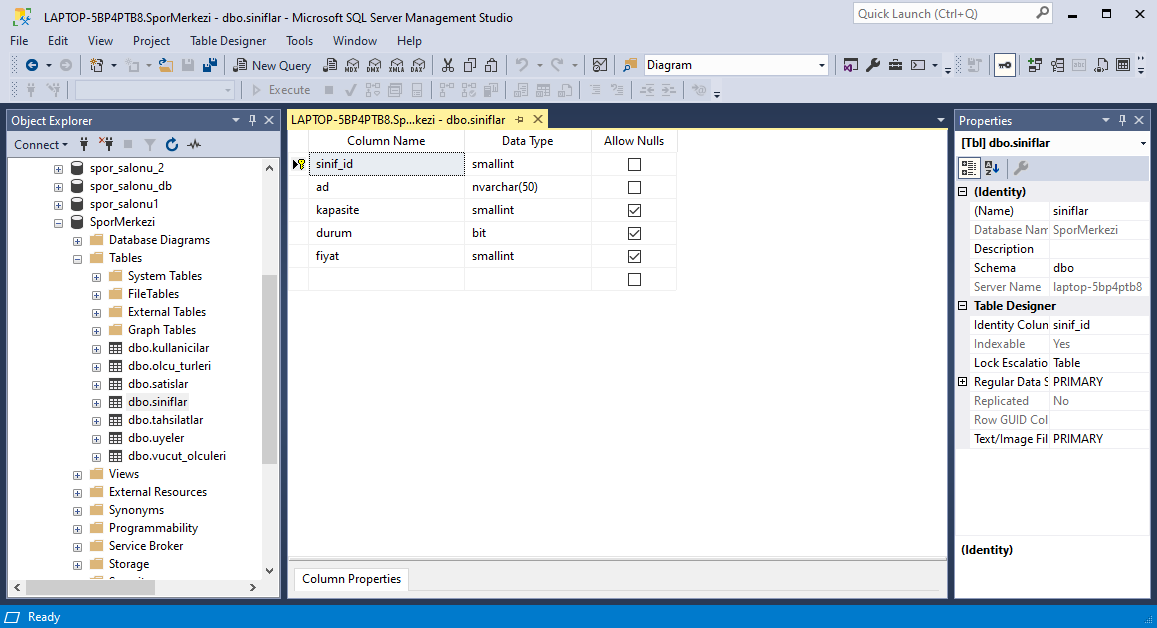
**Tablo 9**: satislar Tablosu



### siniflar Tablosu

* sinif\_id, smallint veri tipi,
* ad, nvarchar(50) veri tipi,
* kapasite, smallint veri tipi,
* durum, bit veri tipi,
* fiyat, smallint veri tipi,
* silme\_durumu, bit veri tipi olarak tasarlanmıştır

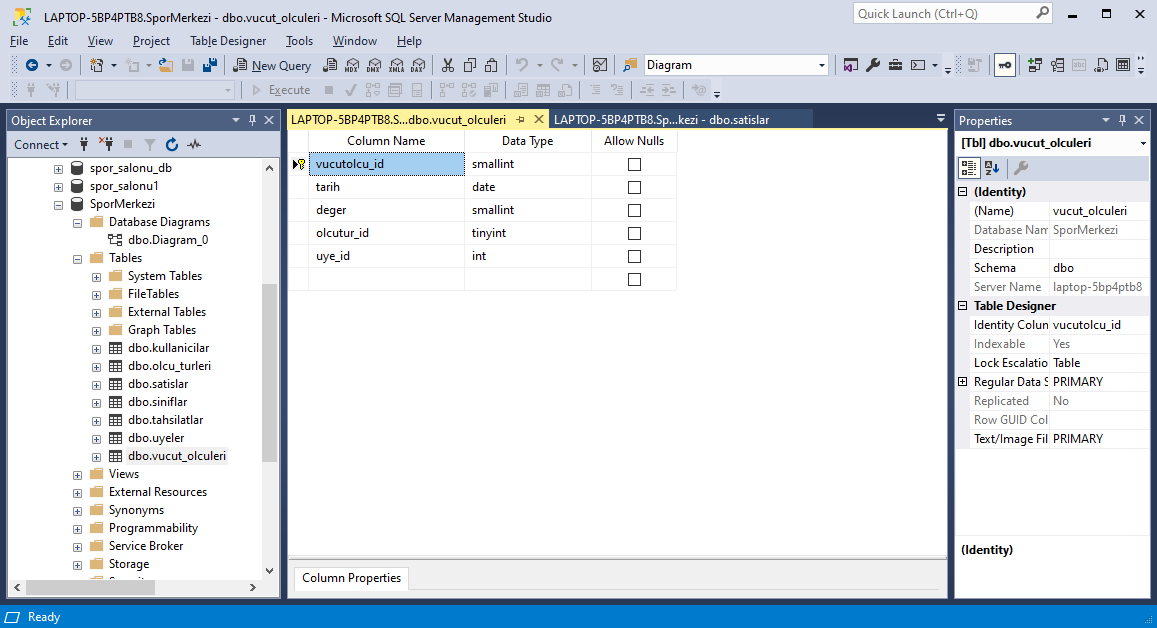
**Tablo 10**: siniflar Tablosu



### vucut\_olculeri Tablosu

* vucutolcu\_id, smallint veri tipi,
* tarih, date veri tipi,
* değer, smallint veri tipi,
* tur\_id, tinyint veri tipi ikincil anahtar,
* uye\_id, int veri tipi ikincil anahtar olarak tasarlanmıştır.

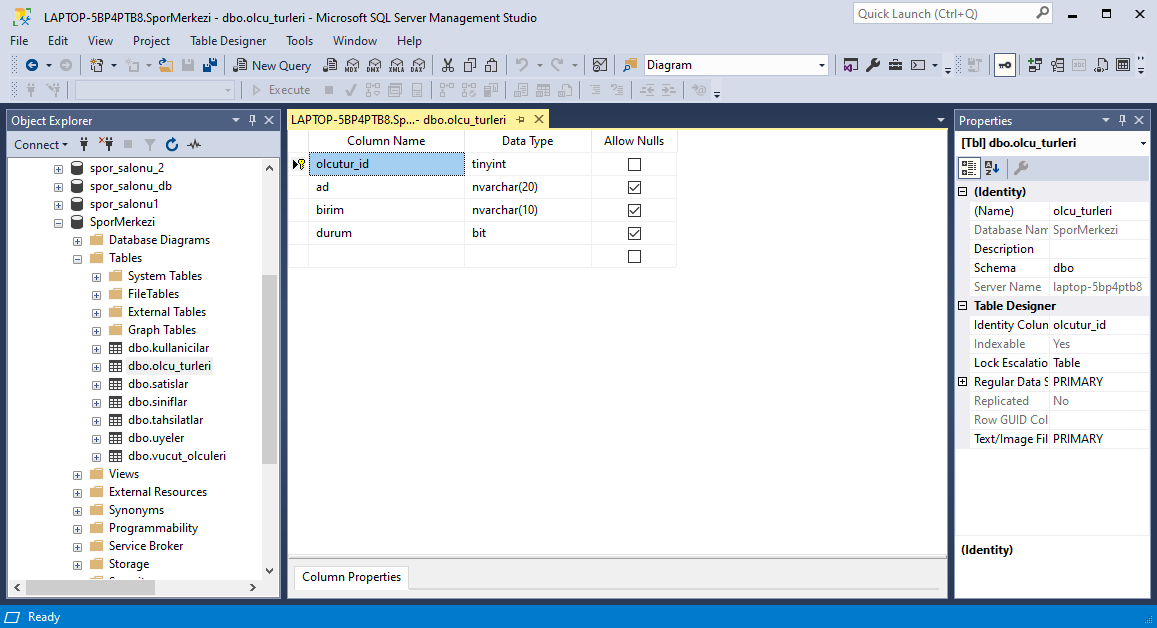
**Tablo 11**: vücut\_olculeri Tablosu



### olcu\_turleri Tablosu

* olcutur\_id, tinyint veri tipi,
* ad, nvarchar(20) veri tipi,
* birim, nvarchar(10) veri tipi,
* durum, bit veri tipi,
* silme\_durumu, bit veri tipi olarak tasarlanmıştır.

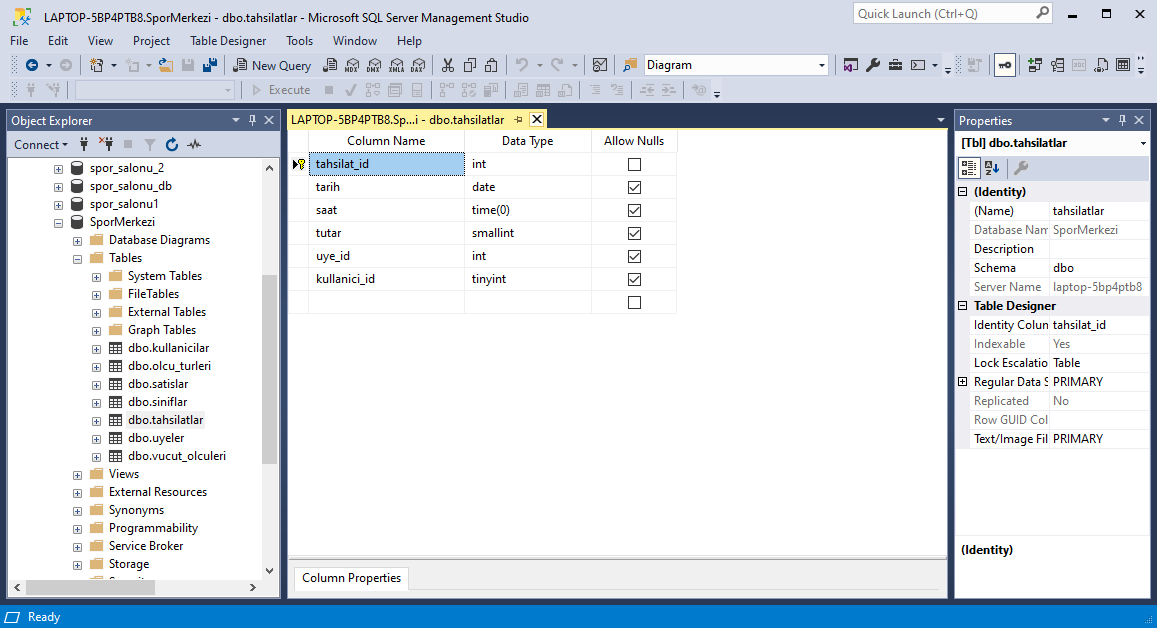
**Tablo 12**: olcu\_turleri Tablosu



### tahsilatlar Tablosu

* tahsilat\_id, int veri tipi,
* tarih, date veri tipi,
* saat, time(0) veri tipi,
* tutar, smallint veri tipi,
* uye\_id, int veri tipi ikincil anahtar,
* kullanici\_id, tinyint veri tipi
* durum, bit veri tipi ikincil anahtar olarak tasarlanmıştır.

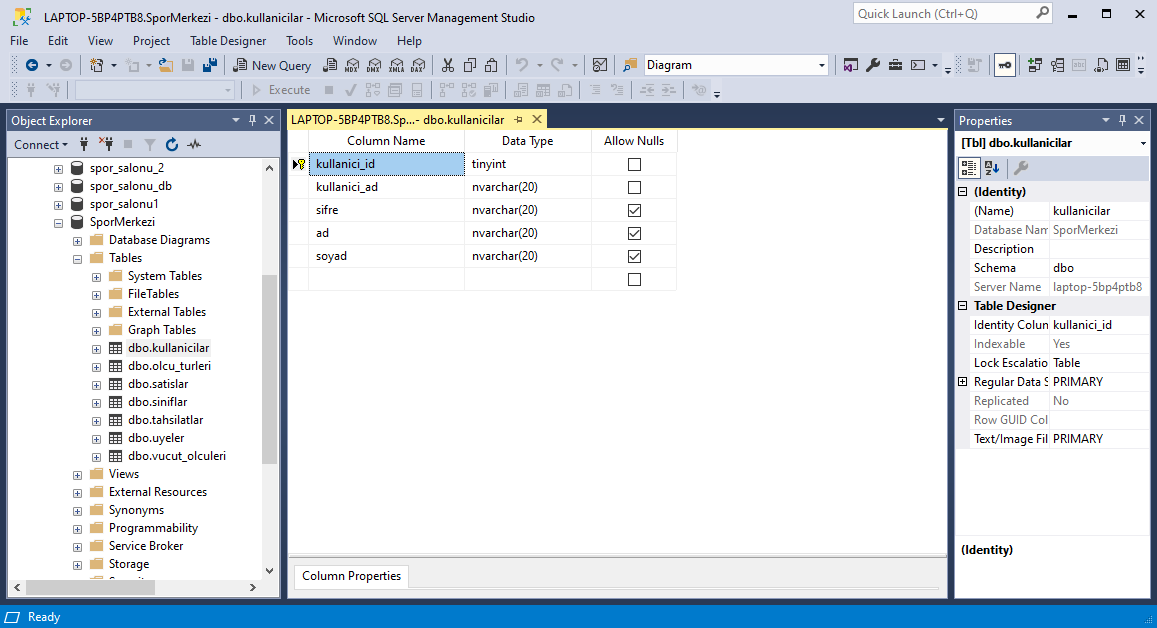
**Tablo 13**: tahsilatlar Tablosu



### kullanicilar Tablosu

* kullanici\_id, tinyint veri tipi,
* kullanici\_ad, nvarchar(20) veri tipi,
* sifre, nvarchar(20) veri tipi,
* ad, nvarchar(20) veri tipi,
* soyad, nvarchar(20) veri tipi,
* silme\_durumu, bit veri tipi olarak tasarlanmıştır.

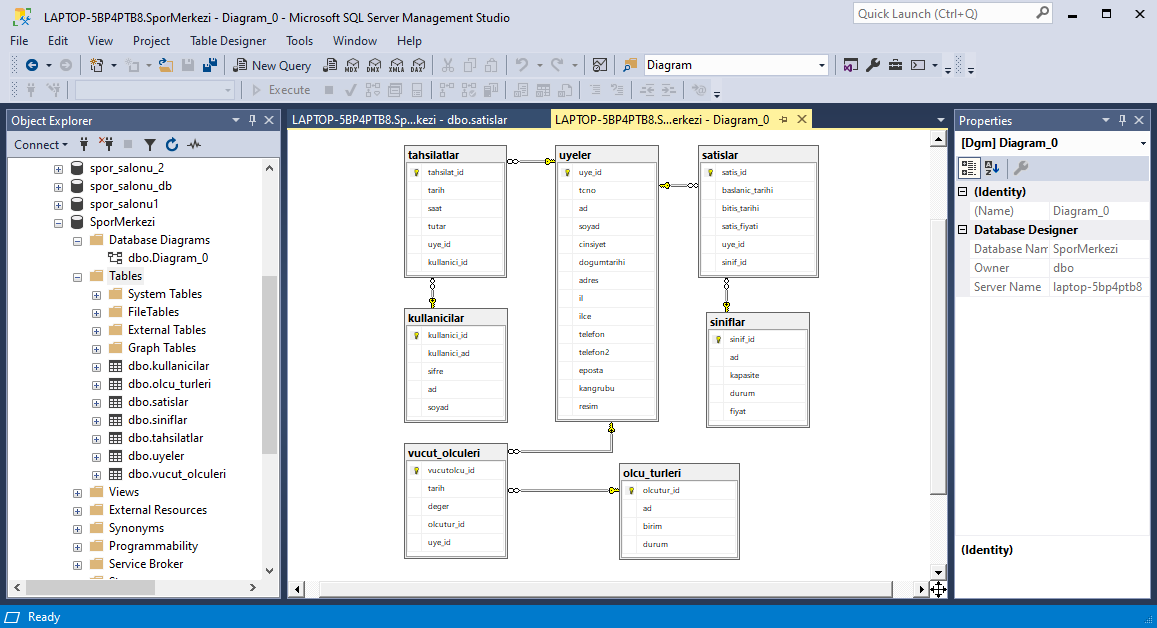
**Tablo 14**: kullanicilar Tablosu



### Veri Tabanı Diyagramı

Tablolar yukarıda belirtilen analizler ışığında aşağıda belirtildiği şekilde birbirleri ile ilişkilendirilmiş ve diyagram oluşturulmuştur.

**Tablo 15**: Veritabanı Diyagramı



### Tabloların SQL Kodları

Oluşturulan tablolara ait SQL kodları aşağıda sunulmuştur.

USE [SporMerkezi]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[kullanicilar](

[kullanici\_id] [tinyint] IDENTITY(1,1) NOT NULL,

[kullanici\_ad] [nvarchar](20) NOT NULL,

[sifre] [nvarchar](20) NULL,

[ad] [nvarchar](20) NULL,

[soyad] [nvarchar](20) NULL,

[silme\_durumu] [bit] NULL,

CONSTRAINT [PK\_kullanicilar] PRIMARY KEY CLUSTERED

(

[kullanici\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[olcu\_turleri](

[olcutur\_id] [tinyint] IDENTITY(1,1) NOT NULL,

[ad] [nvarchar](20) NULL,

[birim] [nvarchar](10) NULL,

[durum] [bit] NULL,

[silme\_durumu] [bit] NULL,

CONSTRAINT [PK\_olcu\_turleri] PRIMARY KEY CLUSTERED

(

[olcutur\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[satislar](

[satis\_id] [int] IDENTITY(1,1) NOT NULL,

[baslanic\_tarihi] [date] NULL,

[bitis\_tarihi] [date] NULL,

[satis\_fiyati] [smallint] NULL,

[uye\_id] [int] NULL,

[sinif\_id] [smallint] NOT NULL,

[durum] [bit] NULL,

CONSTRAINT [PK\_satislar] PRIMARY KEY CLUSTERED

(

[satis\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[siniflar](

[sinif\_id] [smallint] IDENTITY(1,1) NOT NULL,

[ad] [nvarchar](50) NOT NULL,

[kapasite] [smallint] NULL,

[durum] [bit] NULL,

[fiyat] [smallint] NULL,

CONSTRAINT [PK\_siniflar] PRIMARY KEY CLUSTERED

(

[sinif\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[tahsilatlar](

[tahsilat\_id] [int] IDENTITY(1,1) NOT NULL,

[tarih] [date] NULL,

[saat] [time](0) NULL,

[tutar] [smallint] NULL,

[uye\_id] [int] NULL,

[kullanici\_id] [tinyint] NULL,

[silme\_durumu] [bit] NULL,

CONSTRAINT [PK\_tahsilatlar] PRIMARY KEY CLUSTERED

(

[tahsilat\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[uyeler](

[uye\_id] [int] IDENTITY(1,1) NOT NULL,

[tcno] [char](11) NULL,

[ad] [nvarchar](20) NULL,

[soyad] [nvarchar](20) NULL,

[cinsiyet] [bit] NULL,

[dogumtarihi] [date] NULL,

[adres] [nvarchar](100) NULL,

[il] [nvarchar](20) NULL,

[ilce] [nvarchar](20) NULL,

[telefon] [char](11) NULL,

[telefon2] [char](11) NULL,

[eposta] [nvarchar](30) NULL,

[kangrubu] [nvarchar](5) NULL,

[resim] [nvarchar](30) NULL,

[silme\_durumu] [bit] NULL,

CONSTRAINT [PK\_uyeler] PRIMARY KEY CLUSTERED

(

[uye\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[vucut\_olculeri](

[vucutolcu\_id] [smallint] IDENTITY(1,1) NOT NULL,

[tarih] [date] NOT NULL,

[deger] [smallint] NOT NULL,

[olcutur\_id] [tinyint] NOT NULL,

[uye\_id] [int] NOT NULL,

CONSTRAINT [PK\_vucut\_olculeri] PRIMARY KEY CLUSTERED

(

[vucutolcu\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[satislar] WITH CHECK ADD CONSTRAINT [FK\_satislar\_siniflar] FOREIGN KEY([sinif\_id])

REFERENCES [dbo].[siniflar] ([sinif\_id])

GO

ALTER TABLE [dbo].[satislar] CHECK CONSTRAINT [FK\_satislar\_siniflar]

GO

ALTER TABLE [dbo].[satislar] WITH CHECK ADD CONSTRAINT [FK\_satislar\_uyeler] FOREIGN KEY([uye\_id])

REFERENCES [dbo].[uyeler] ([uye\_id])

GO

ALTER TABLE [dbo].[satislar] CHECK CONSTRAINT [FK\_satislar\_uyeler]

GO

ALTER TABLE [dbo].[tahsilatlar] WITH CHECK ADD CONSTRAINT [FK\_tahsilatlar\_kullanicilar] FOREIGN KEY([kullanici\_id])

REFERENCES [dbo].[kullanicilar] ([kullanici\_id])

GO

ALTER TABLE [dbo].[tahsilatlar] CHECK CONSTRAINT [FK\_tahsilatlar\_kullanicilar]

GO

ALTER TABLE [dbo].[tahsilatlar] WITH CHECK ADD CONSTRAINT [FK\_tahsilatlar\_uyeler] FOREIGN KEY([uye\_id])

REFERENCES [dbo].[uyeler] ([uye\_id])

GO

ALTER TABLE [dbo].[tahsilatlar] CHECK CONSTRAINT [FK\_tahsilatlar\_uyeler]

GO

ALTER TABLE [dbo].[vucut\_olculeri] WITH CHECK ADD CONSTRAINT [FK\_vucut\_olculeri\_olcu\_turleri] FOREIGN KEY([olcutur\_id])

REFERENCES [dbo].[olcu\_turleri] ([olcutur\_id])

GO

ALTER TABLE [dbo].[vucut\_olculeri] CHECK CONSTRAINT [FK\_vucut\_olculeri\_olcu\_turleri]

GO

ALTER TABLE [dbo].[vucut\_olculeri] WITH CHECK ADD CONSTRAINT [FK\_vucut\_olculeri\_uyeler] FOREIGN KEY([uye\_id])

REFERENCES [dbo].[uyeler] ([uye\_id])

GO

ALTER TABLE [dbo].[vucut\_olculeri] CHECK CONSTRAINT [FK\_vucut\_olculeri\_uyeler]

GO

## Programın Akış Şeması

### Giriş Ekranı Akış Şeması

**Veri Tabanı ile Bağlantı var mı? mı?**

**Şifre**

**Uyarı  
Veri Tabanı ile Bağlantı Kurulamadı.**

Hayır

Evet

**Kullanıcı Adı veya Şifresi Hatalı**

Hayır

**Kullanıcı Adı ve Şifre Ekranı**

Tekrar Dene

**Kullanıcı ve Şifre Doğru Evet ise**

Evet

**Ana Sayfa**

**Veri Tabanı Bağlantısını  
Kapat  
  
  
MySqlCommand**

### Arama Akış Şeması

**Programda bulunan verilerin İd No gir**

**Uyarı**

**Veri Tabanı ile Bağlantı Kurulamadı.**

**Veri Tabanı Bağlantı aç**

Hayır

**Veri Tabanı ile Bağlantı var mı? mı?**

**Şifre**

**Bilgileri Listele**

**Bilgileri Aktar**

**Veri Tabanı Sorgusu**

### Ekleme Akış Şeması

**Bilgileri Al**

**Bilgi Girişini Onayla**

**Veritabanına Bağlan**

Hayır

**Uyarı**

**Veri Tabanı ile Bağlantı Kurulamadı.**

**Bilgileri Verilere Yükle**

**Verileri Command Nesnesine Ekle**

**Kaydet**

**Öğrenci Bilgileri Kayıt Edildi.**

### Silme İşlemi Akış Diyagramı

**Programda bulunan verilerin İd No gir**

**Silme İşlemi İptal Edildi**

Hayır

**Silme İşemini Onayla**

**Şifre**

**Uyarı**

**Veri Tabanı ile Bağlantı Kurulamadı.**

**Veri Tabanı Bağlantı aç**

Hayır

**Veri Tabanı ile Bağlantı var mı? mı?**

**Şifre**

**Veri Tabanı Sorgusu**

### Güncelleme İşlemi Akış Diyagramı

**Programda bulunan verileri al**

**Güncelleme İşlemi İptal Edildi**

Hayır

**Güncelleme İşemini Onayla**

**Şifre**

**Uyarı**

**Veri Tabanı ile Bağlantı Kurulamadı.**

**Veri Tabanı Bağlantı aç**

Hayır

**Veri Tabanı ile Bağlantı var mı? mı?**

**Şifre**

**Veri Tabanı Sorgusu**

### Üye Bilgilerinin Görüntüleme Akış Diyagramı

**Üyenin İd No Al**

**Üye Bilgileri Görüntülendi**

**Uyarı**

**Veri Tabanı ile Bağlantı Kurulamadı.**

**Veri Tabanı Bağlantı aç**

Hayır

**Veri Tabanı ile Bağlantı var mı? mı?**

**Şifre**

**Veri Tabanı Sorgusu**

### Kullanıcı Bilgilerinin Görüntüleme Akış Diyagramı

**Kullanıcının İd No Al**

**Kullanıcı Bilgileri Görüntülendi**

**Uyarı**

**Veri Tabanı ile Bağlantı Kurulamadı.**

**Veri Tabanı Bağlantı aç**

Hayır

**Veri Tabanı ile Bağlantı var mı? mı?**

**Şifre**

**Veri Tabanı Sorgusu**

## Proje Uygulama Kısmı

### Spor Merkezi Otomasyonu Giriş Ekranı

İlk olarak MSSQL veri tabanı ile bağlantı kurulması sağlanmıştır. Giriş ekranında kullanıcı adı ve şifre doğru girildiğinde **Ana Sayfa** gelmektedir.

**Tablo 16:**Giriş Ekranı

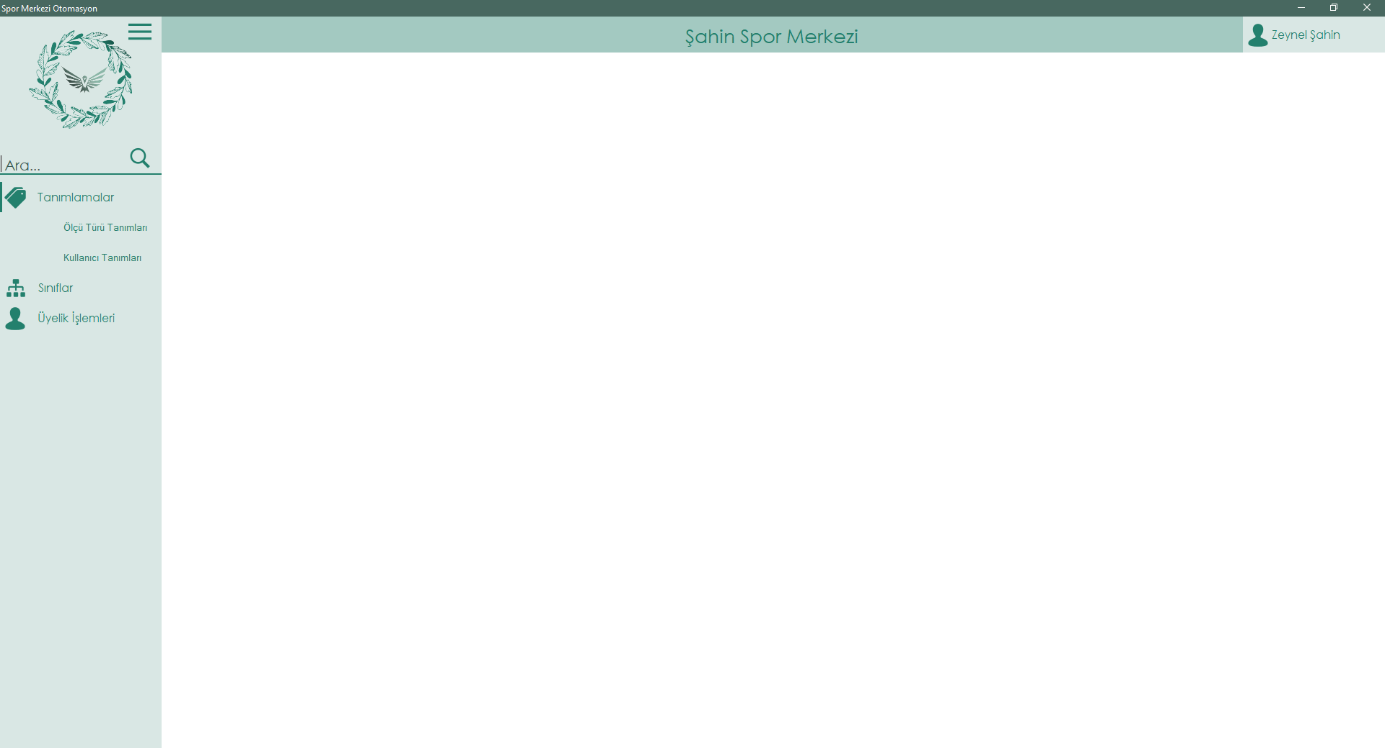


### Spor Merkezi Otomasyonu Ana Sayfası

Giriş yapıldığında sırasıyla sekme olarak;

* Tanımlamalar(Ölçü Türü ve Kullanıcı)
* Sınıflar
* Üyelik İşlemleri(Üye Ara ve Ön kayıt)

**Tablo 17:** Ana Sayfa



**Tanımlamalar(Ölçü Türü ve Kullanıcı) Sekmesinde:**

Tanımlamalar sekmesinde iki menü bulunuyor. İlk menü **Ölçü Türü Tanımlamaları** menüsünde;

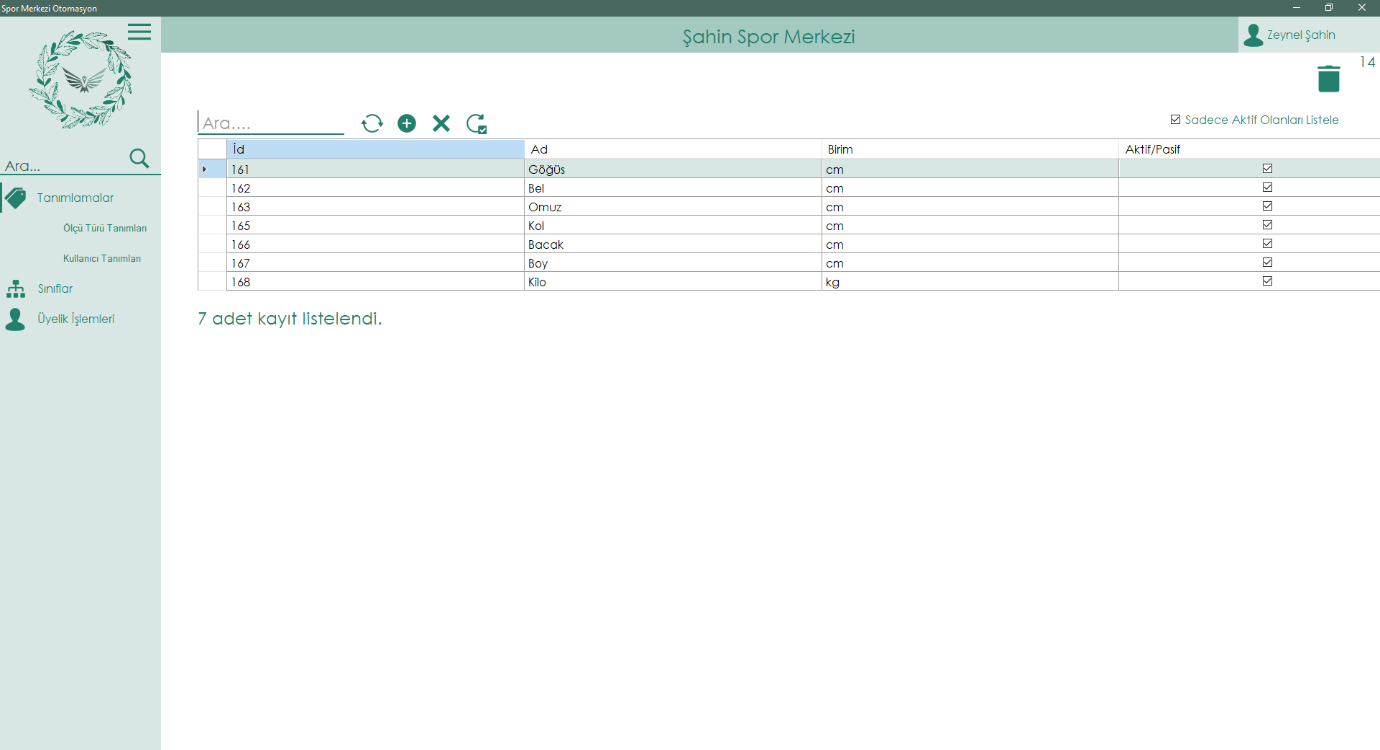
* Ara
* Ekle
* Sil
* Güncelle
* Yeni kayıt işlemleri yapılmaktadır.

Bu işlemlere ait veriler olcu\_turleri tablosundan getirilip DataGridView’ de gösterilmektedir. Bu bilgiler;

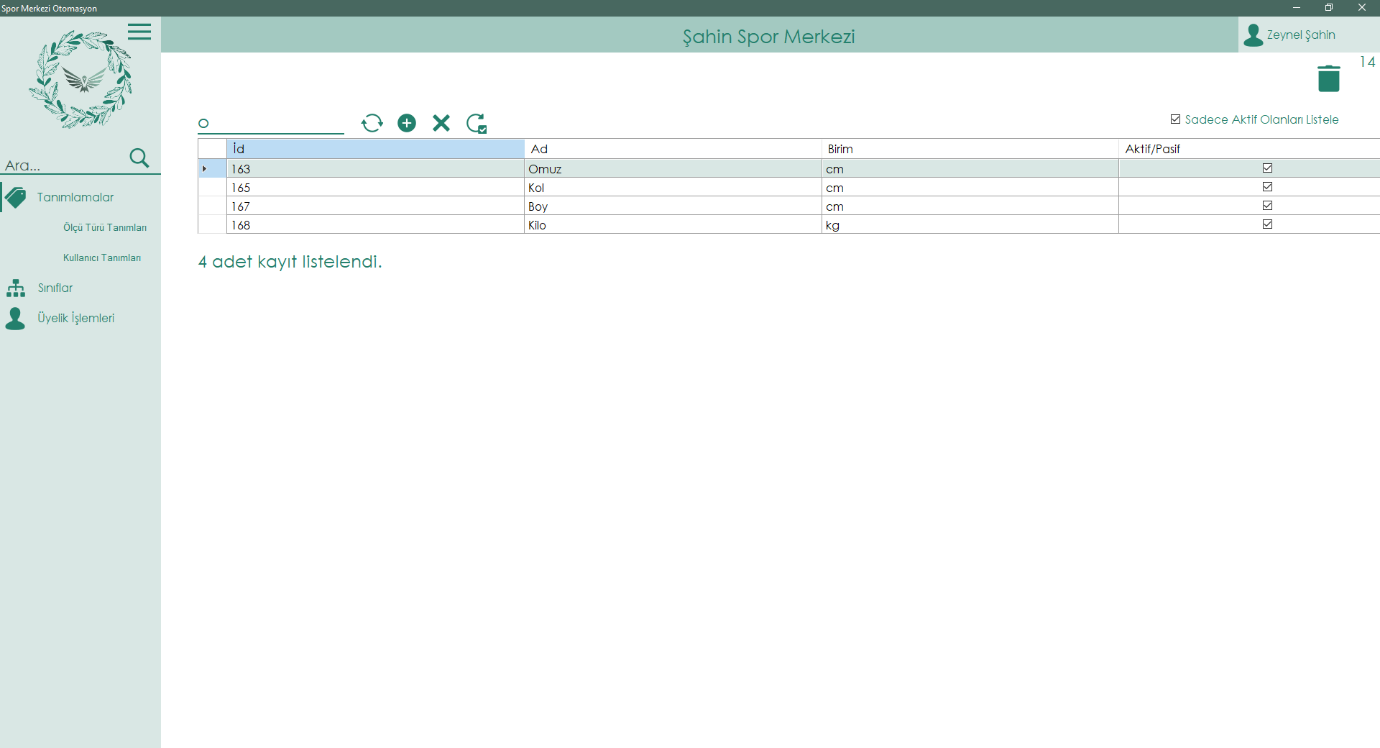
* İd
* Ad
* Birim
* Durum bilgileridir.

Ara yazılan yere yazılan Ölçü Türü adına göre arama yapılmaktadır.

**Tablo 18:** Ölçü Türü Tanımlamaları

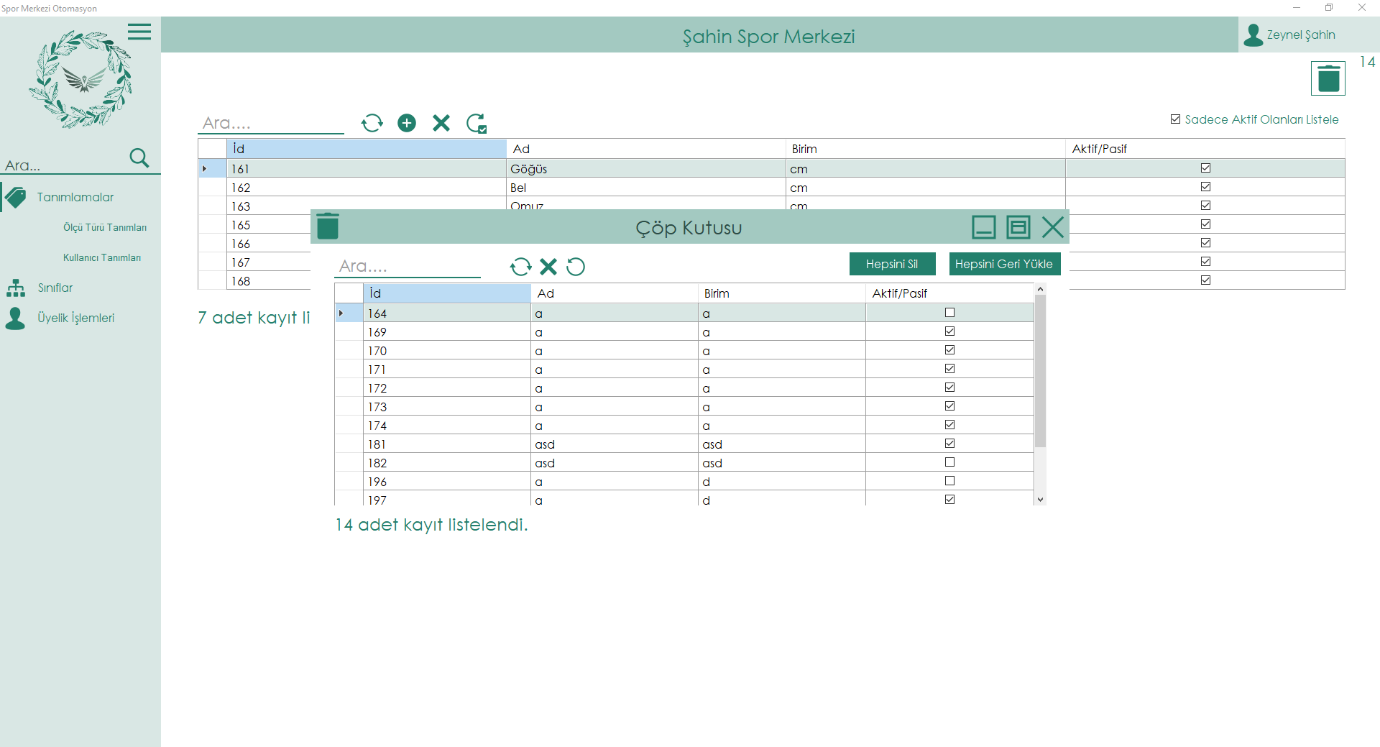


**Tablo 19:** Ölçü Türü Arama Sonucu

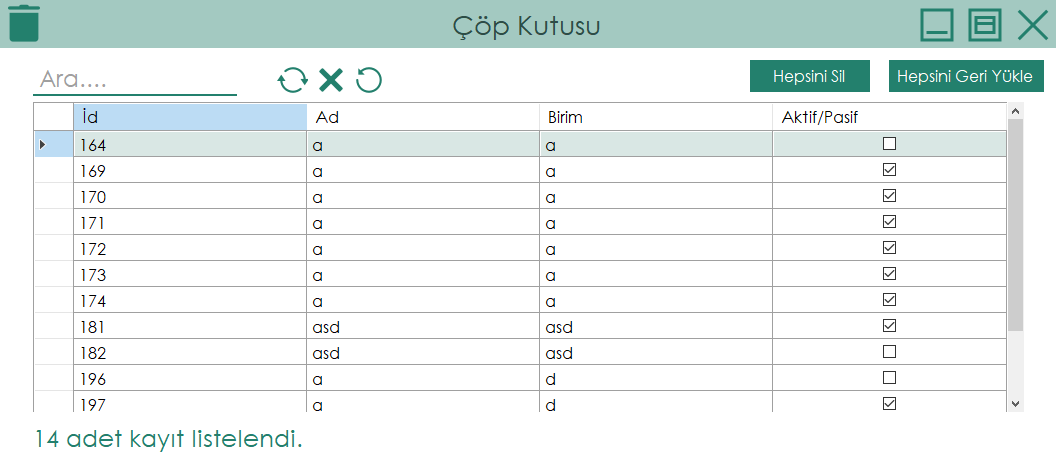


Silme İşlemi için silme butonuna basılarak seçilmiş kayıtları çöp kutusuna gönderir. Temelli silme işlemi için Çöp Kutusuna butonuna basılarak seçilen öğeleri silmek için yine silme butonuna basılır. Karşınıza gelen Silinecekler ekranından Tamam butonuna basılarak silme işlemi gerçekleştirilir.

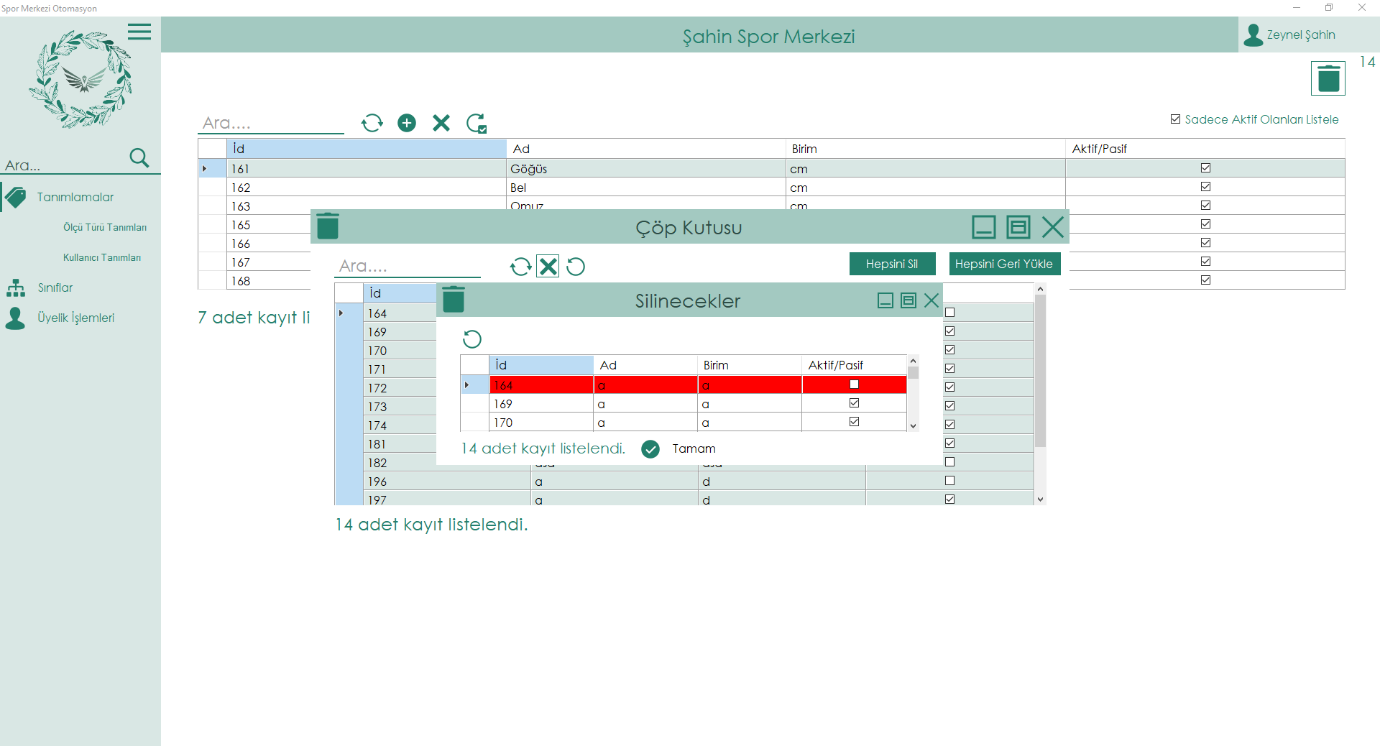
**Tablo 20:** Çöp Kutusu Ölçü Türü



**Tablo 21:** Çöp Kutusu Ölçü Türü (Yakından)



**Tablo 22:** Silinecekler Ölçü Türü

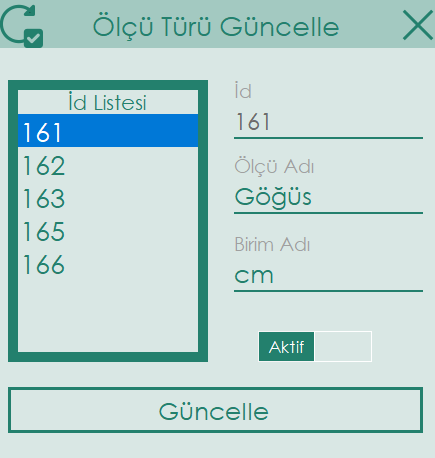


**Tablo 23:** Silinecekler Ölçü Türü (Yakından)



Güncelleme İşlemi için güncelleme butonuna basılarak seçilmiş kayıtları Güncelleme Ekranına İd olarak taşır. Taşınan idler arasından güncellenmek istenilene tıklanır ve ekrandan veriler değiştirilerek güncelle butonuna basılarak güncellenir.

**Tablo 24:** Ölçü Türü Güncelle



Son olarak **Kullanıcı Tanımlamaları** menüsünde;

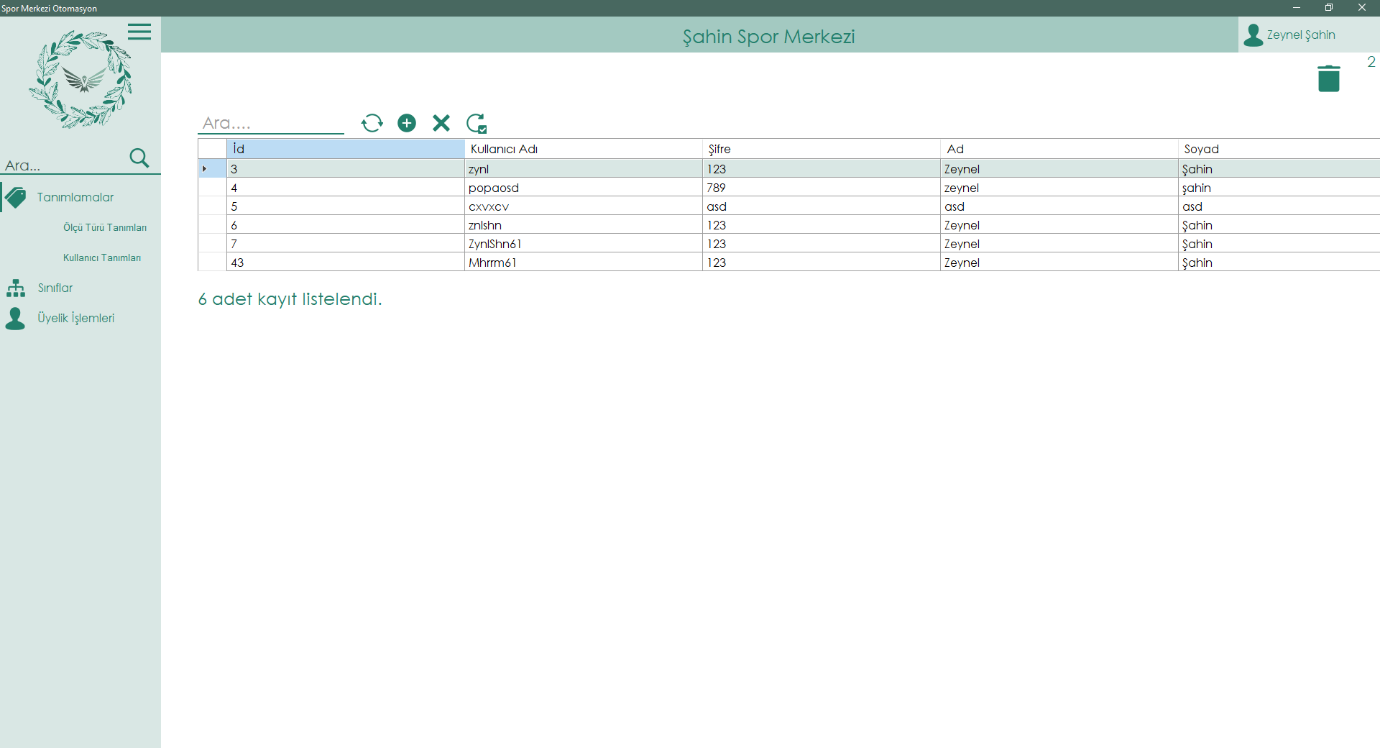
* Ara
* Ekle
* Sil
* Güncelle
* Yeni kayıt işlemleri yapılmaktadır

Bu işlemlere ait veriler olcu\_turleri tablosundan getirilip DataGridView’ de gösterilmektedir. Bu bilgiler;

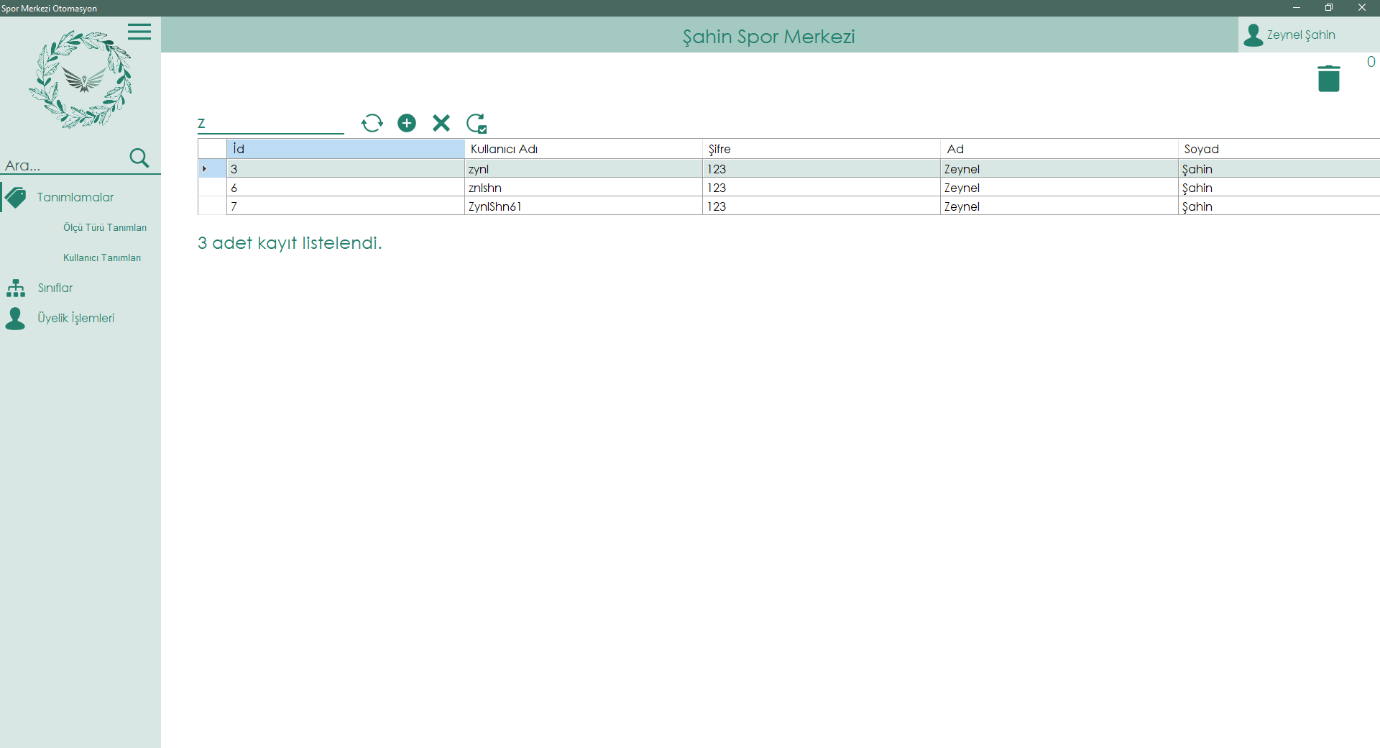
* İd
* Kullanıcı Adı
* Şifre
* Ad
* Soyad

Ara yazılan yere yazılan Ölçü Türü adına göre arama yapılmaktadır.

**Tablo 25:** Kullanıcı Tanımlamaları

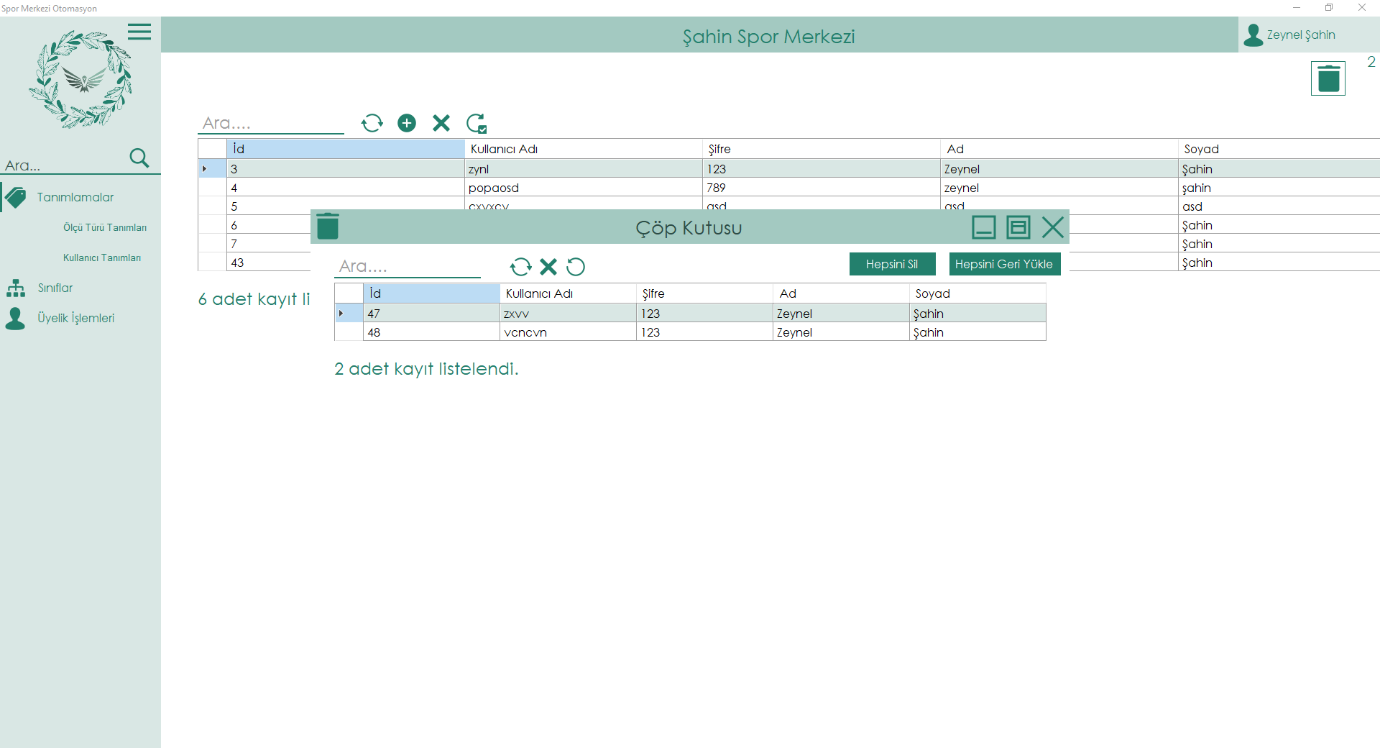


**Tablo 26:** Kullanıcı Arama Sonucu

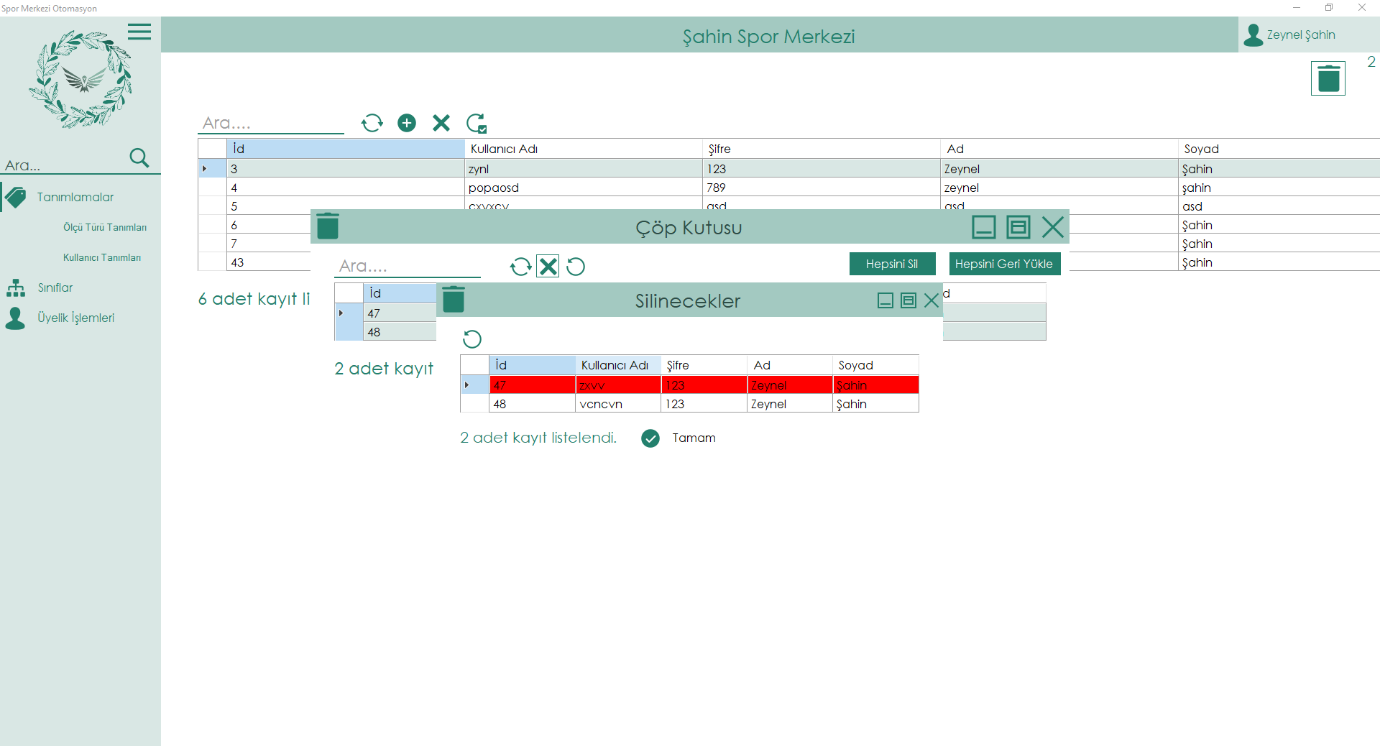


Silme İşlemi için silme butonuna basılarak seçilmiş kayıtları çöp kutusuna gönderir. Temelli silme işlemi için Çöp Kutusuna butonuna basılarak seçilen öğeleri silmek için yine silme butonuna basılır. Karşınıza Silinecekler gelen ekranından Tamam butonuna basılarak silme işlemi gerçekleştirilir.

**Tablo 27:** Çöp Kutusu Kullanıcı

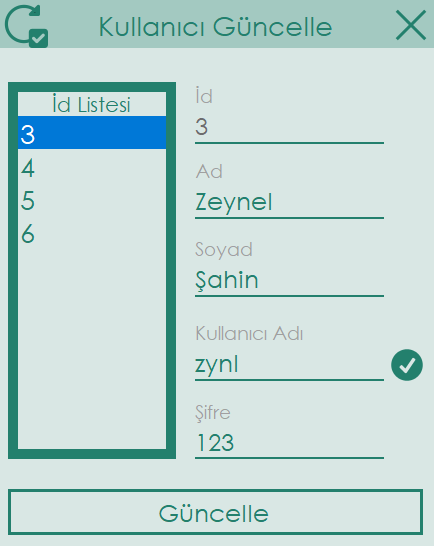


**Tablo 28:** Silinecekler Kullanıcı



Güncelleme İşlemi için güncelleme butonuna basılarak seçilmiş kayıtları Güncelleme Ekranına İd olarak taşır. Taşınan idler arasından güncellenmek istenilene tıklanır ve ekrandan veriler değiştirilerek güncelle butonuna basılarak güncellenir.

**Tablo 29:** Kullanıcı Güncelle



**Sınıflar Sekmesinde:**

Sınıflar Sekmesinde;

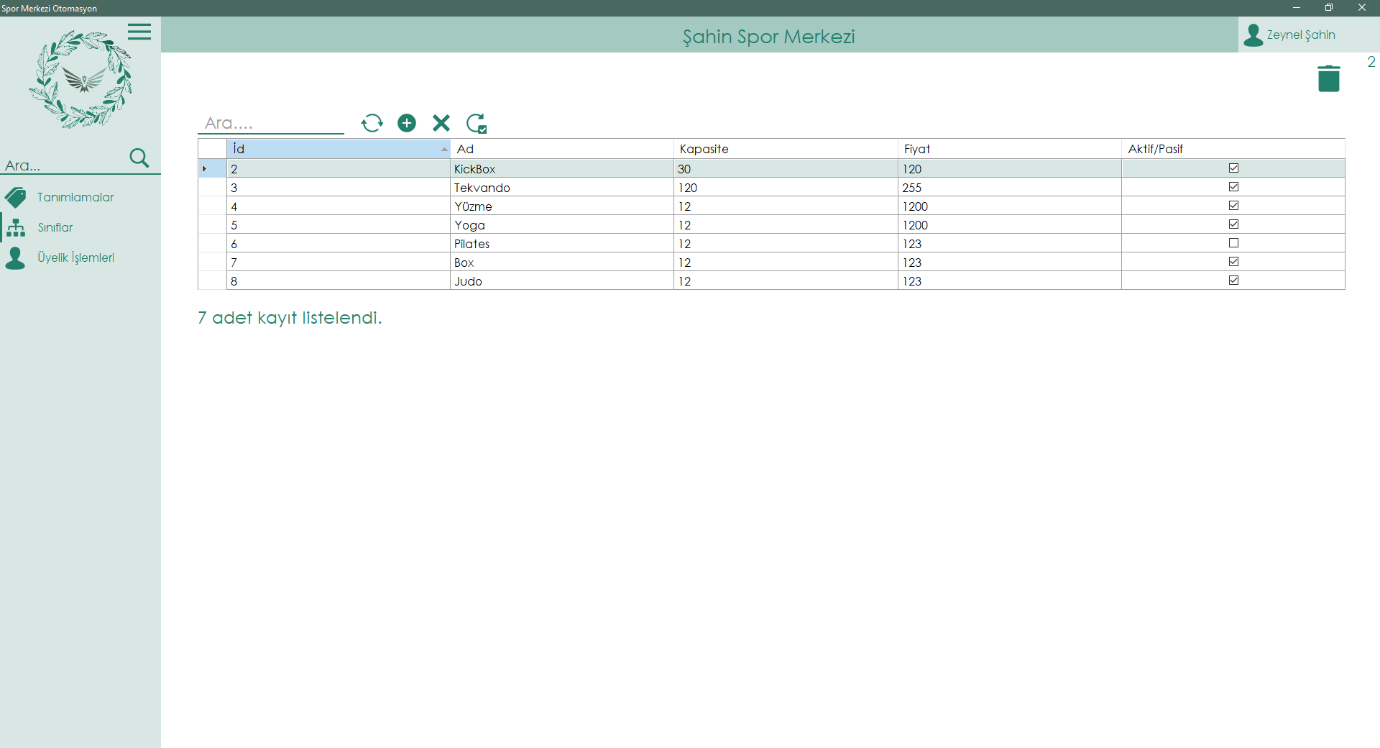
* Ara
* Ekle
* Sil
* Güncelle
* Yeni kayıt işlemleri yapılmaktadır

Bu işlemlere ait veriler sınıflar tablosundan getirilip DataGridView’ de gösterilmektedir. Bu bilgiler;

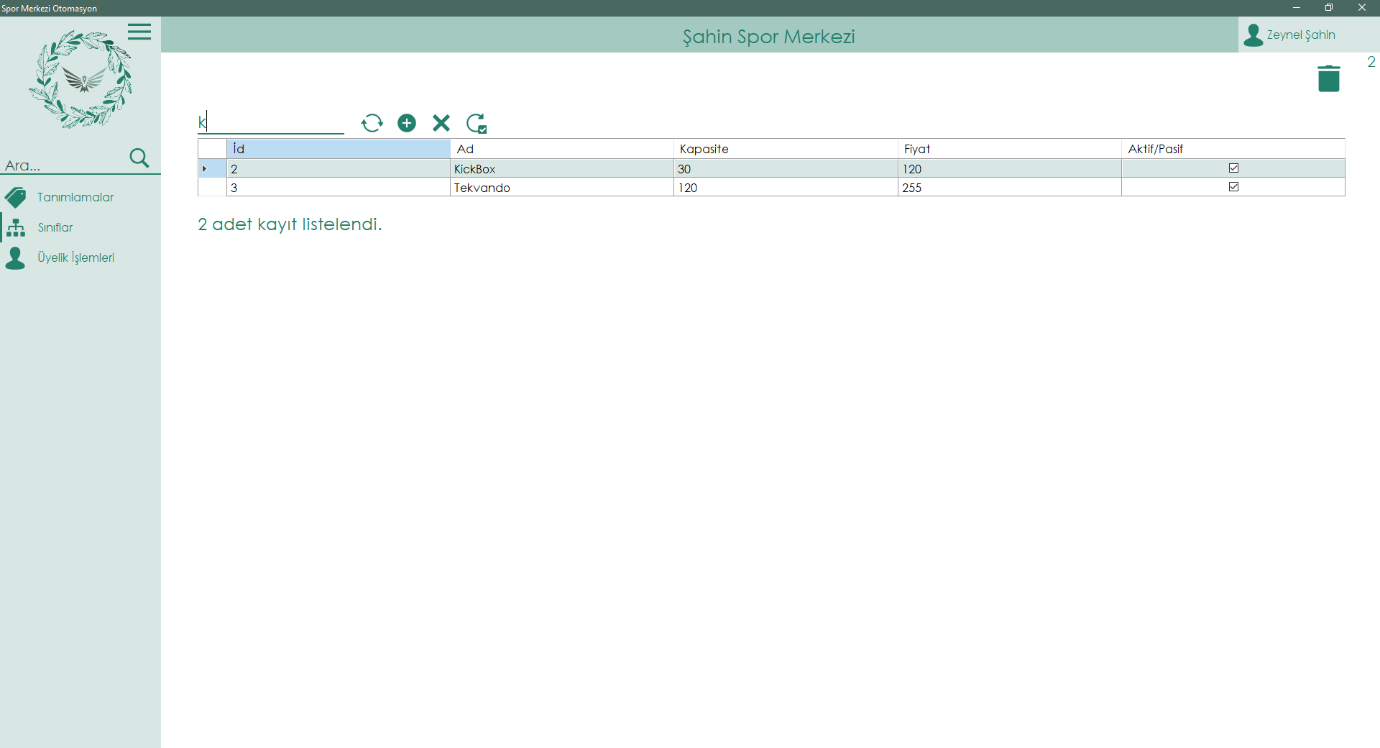
* İd
* Ad
* Kapasite
* Fiyat
* Durum

Ara yazılan yere yazılan Sınıf adına göre arama yapılmaktadır.

**Tablo 30:** Sınıflar

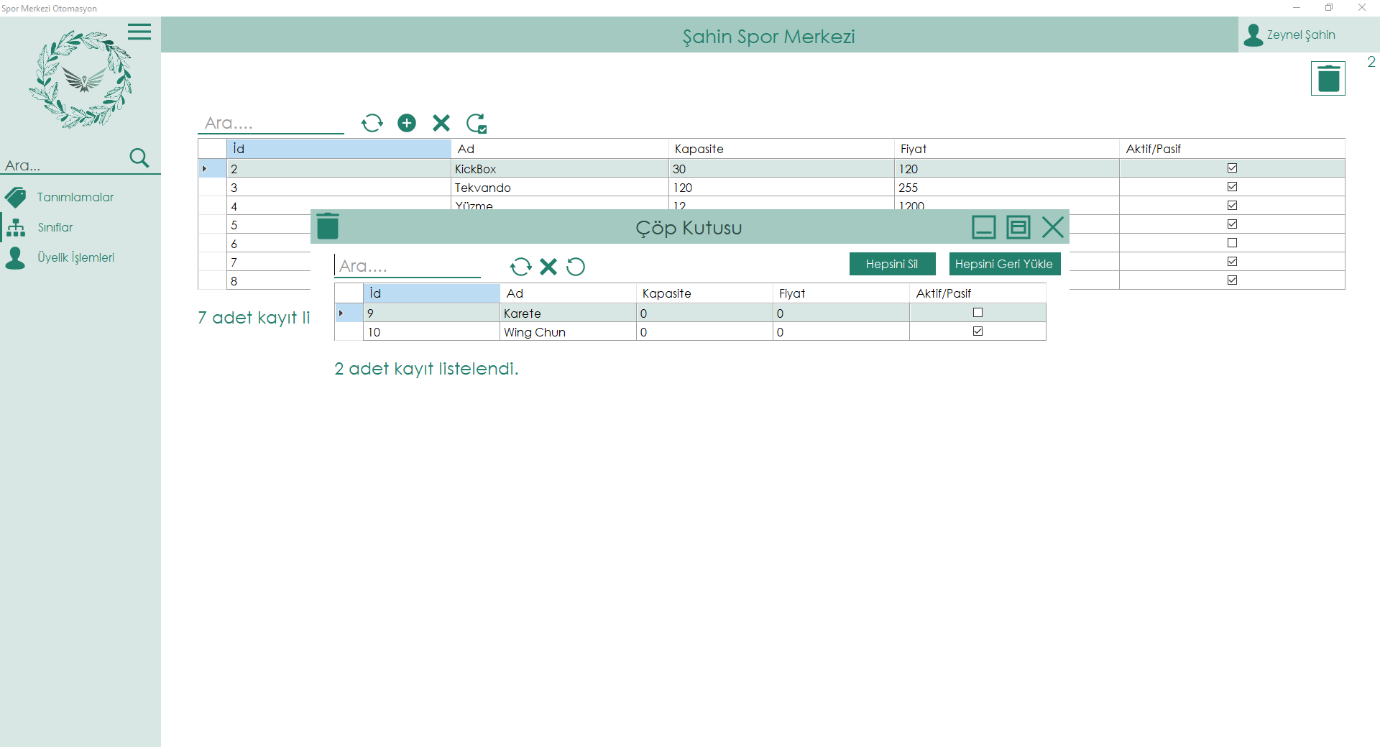
****

**Tablo 31:** Sınıf Arama Sonucu

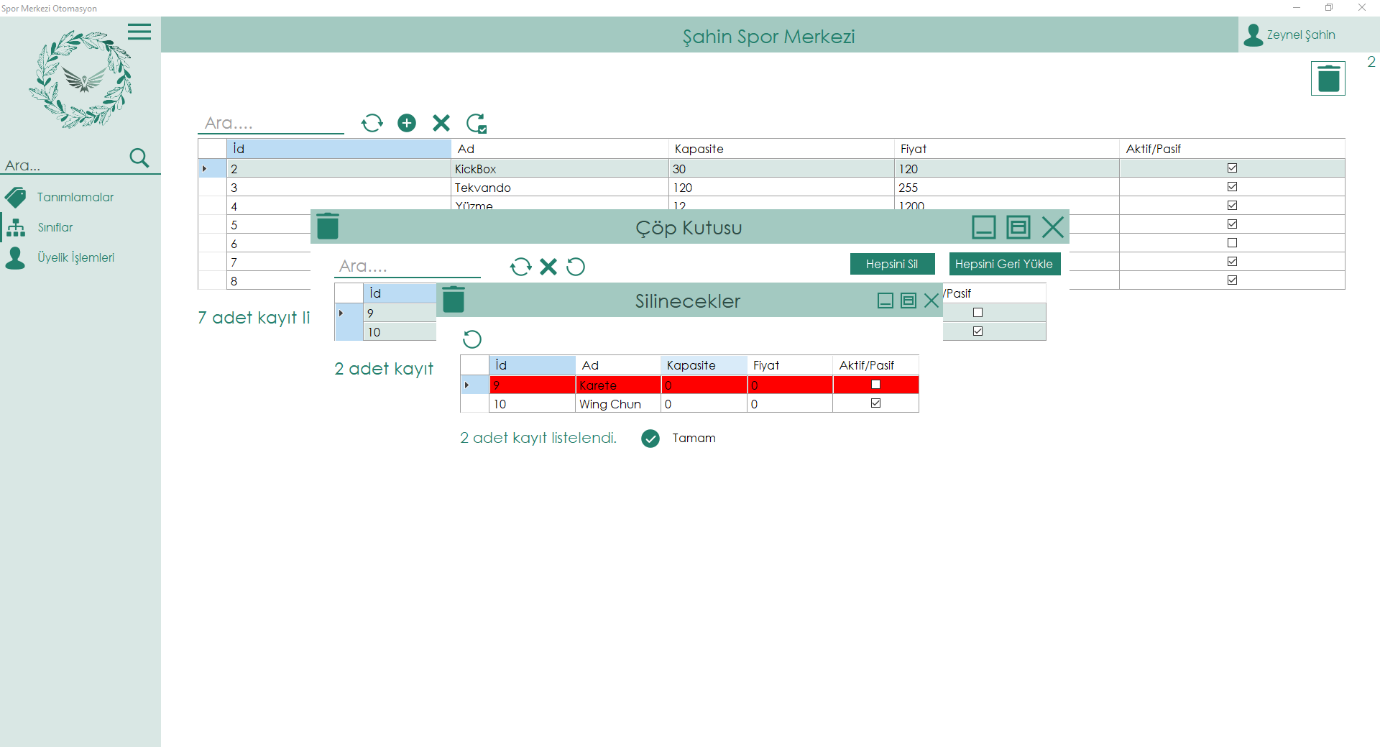


Silme İşlemi için silme butonuna basılarak seçilmiş kayıtları çöp kutusuna gönderir. Temelli silme işlemi için Çöp Kutusuna butonuna basılarak seçilen öğeleri silmek için yine silme butonuna basılır. Karşınıza gelen Silinecekler ekranından Tamam butonuna basılarak silme işlemi gerçekleştirilir.

**Tablo 32:** Çöp Kutusu Sınıf

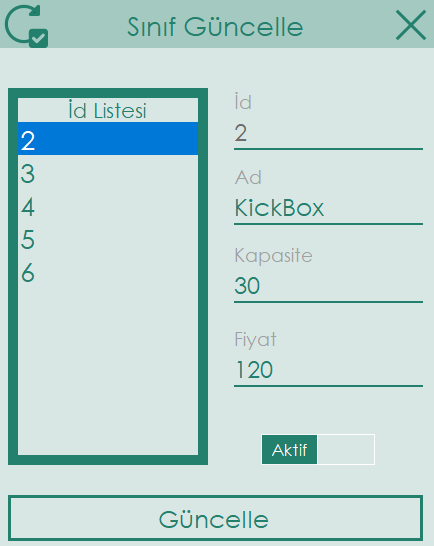


**Tablo 33:** Silinecekler Sınıf



Güncelleme İşlemi için güncelleme butonuna basılarak seçilmiş kayıtları Güncelleme Ekranına İd olarak taşır. Taşınan idler arasından güncellenmek istenilene tıklanır ve ekrandan veriler değiştirilerek güncelle butonuna basılarak güncellenir.

**Tablo 34:** Sınıf Güncelle



**Üyelik İşlemleri (Ön Kayıt ve Üye Ara) Sekmesinde:**

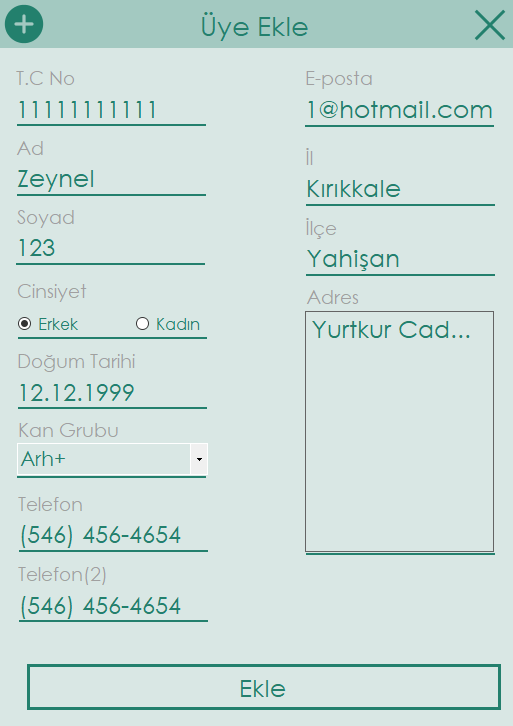
Üyelik İşlemleri sekmesinde iki menü bulunuyor. İlk menü **Ön Kayıt** menüsünde;

* Yeni üye kayıt işlemi yapılmaktadır.

Bu bilgiler kayıt altına alınır;

* T.C. No,
* Adı,
* Soyadı,
* Cinsiyet,
* Doğum Tarihi,
* İl,
* İlçe,
* Adres,
* Silme Durumu,
* Telefon,
* Telefon2,
* E Posta,
* Kan Grubu olarak bilgiler kayıt altına alınır.

**Tablo 35:** Ön Kayıt



Son olarak **Üye Ara** menüsünde;

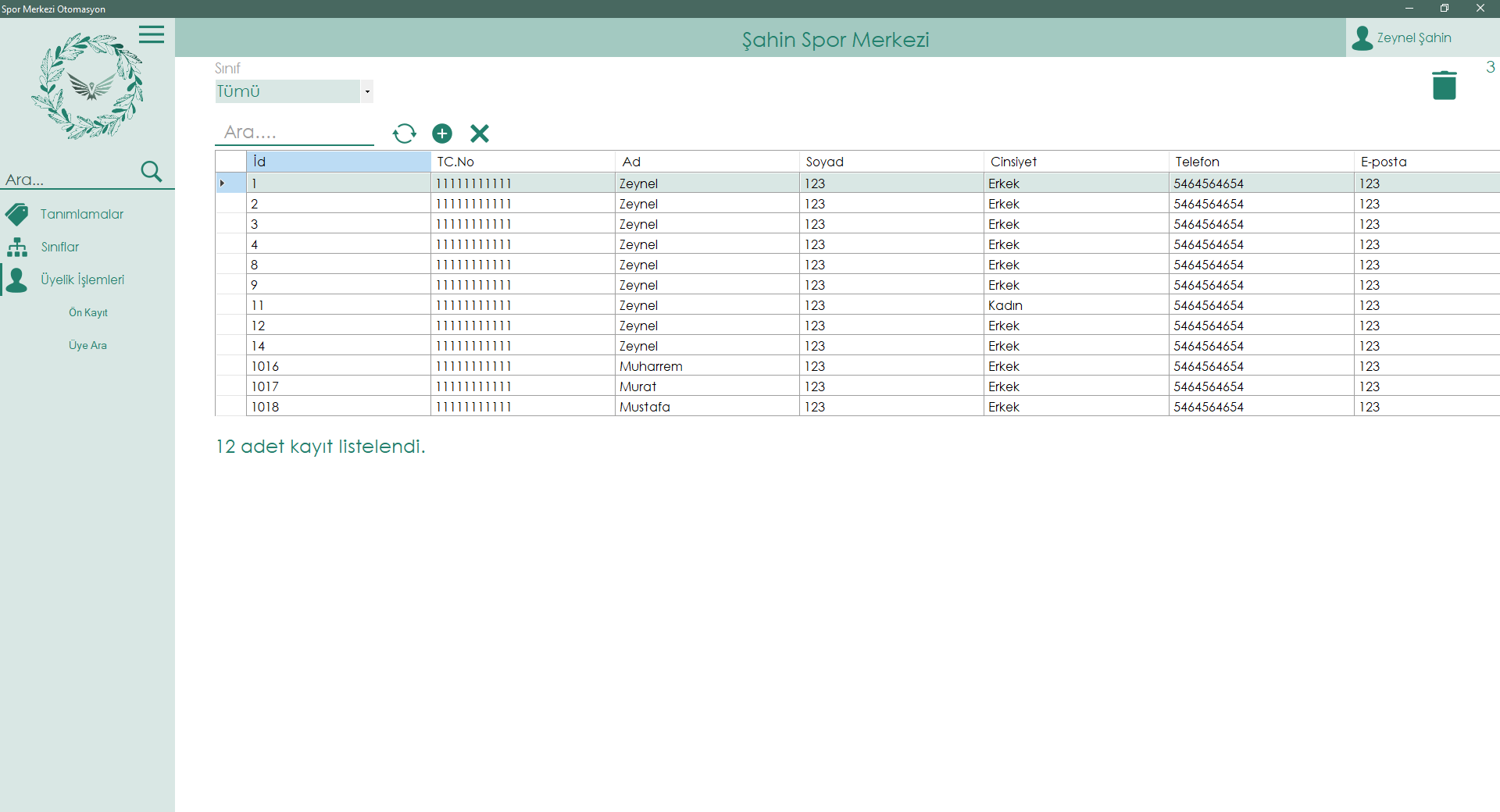
* Ara
* Ekle
* Sil
* Üye Detaylarına erişme
* Yeni kayıt işlemleri yapılmaktadır

Bu işlemlere ait veriler olcu\_turleri tablosundan getirilip DataGridView’ de gösterilmektedir. Bu bilgiler;

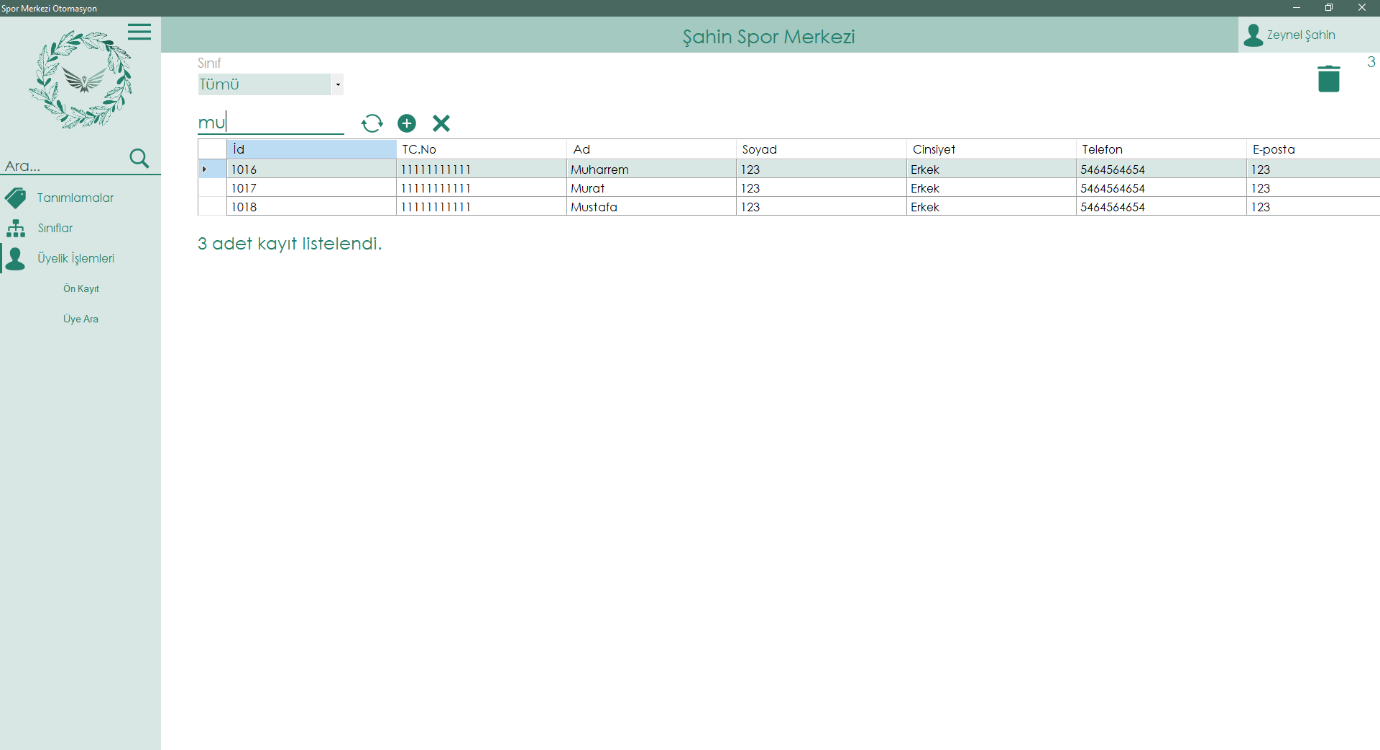
* İd
* TC No
* Ad
* Soyad
* Cinsiyet
* Telefon
* E-posta

Ara yazılan yere yazılan Üye adına göre arama yapılmaktadır.

**Tablo 36:** Üye Ara

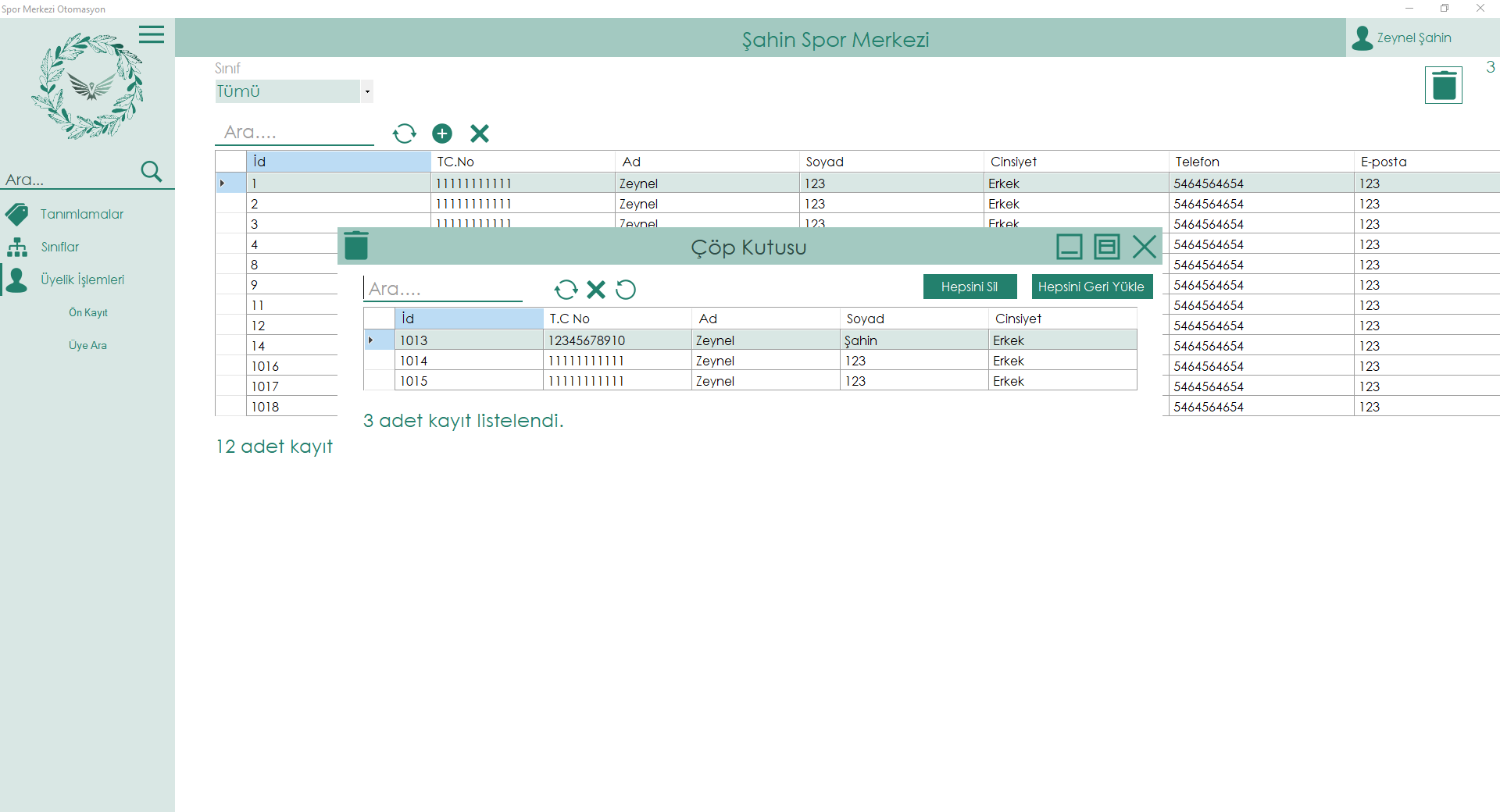


**Tablo 37:** Üye Arama Sonucu

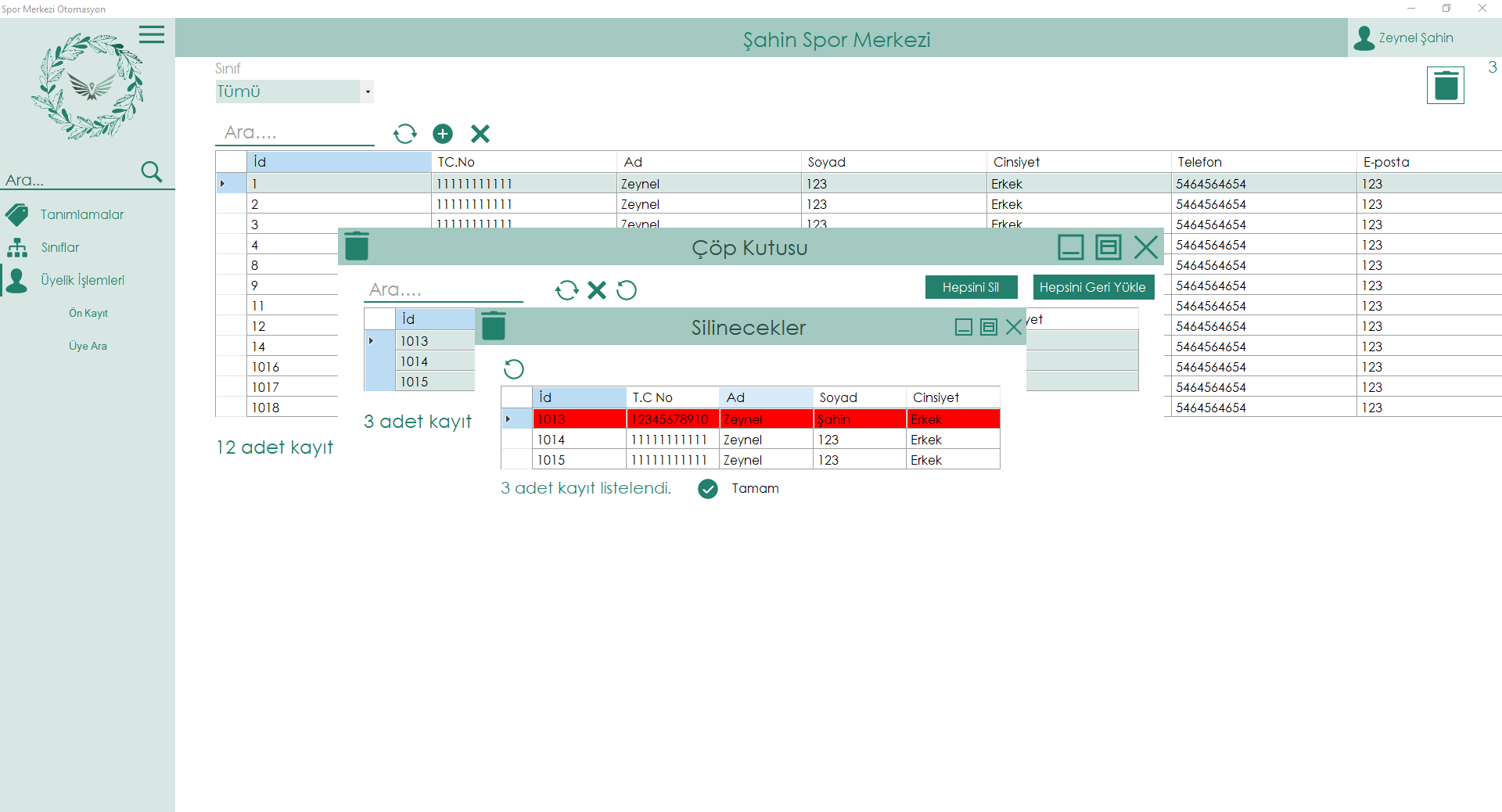


Silme İşlemi için silme butonuna basılarak seçilmiş kayıtları çöp kutusuna gönderir. Temelli silme işlemi için Çöp Kutusuna butonuna basılarak seçilen öğeleri silmek için yine silme butonuna basılır. Karşınıza gelen Silinecekler ekranından Tamam butonuna basılarak silme işlemi gerçekleştirilir.

**Tablo 38:** Çöp Kutusu Üye



**Tablo 39:** Silinecekler Üye



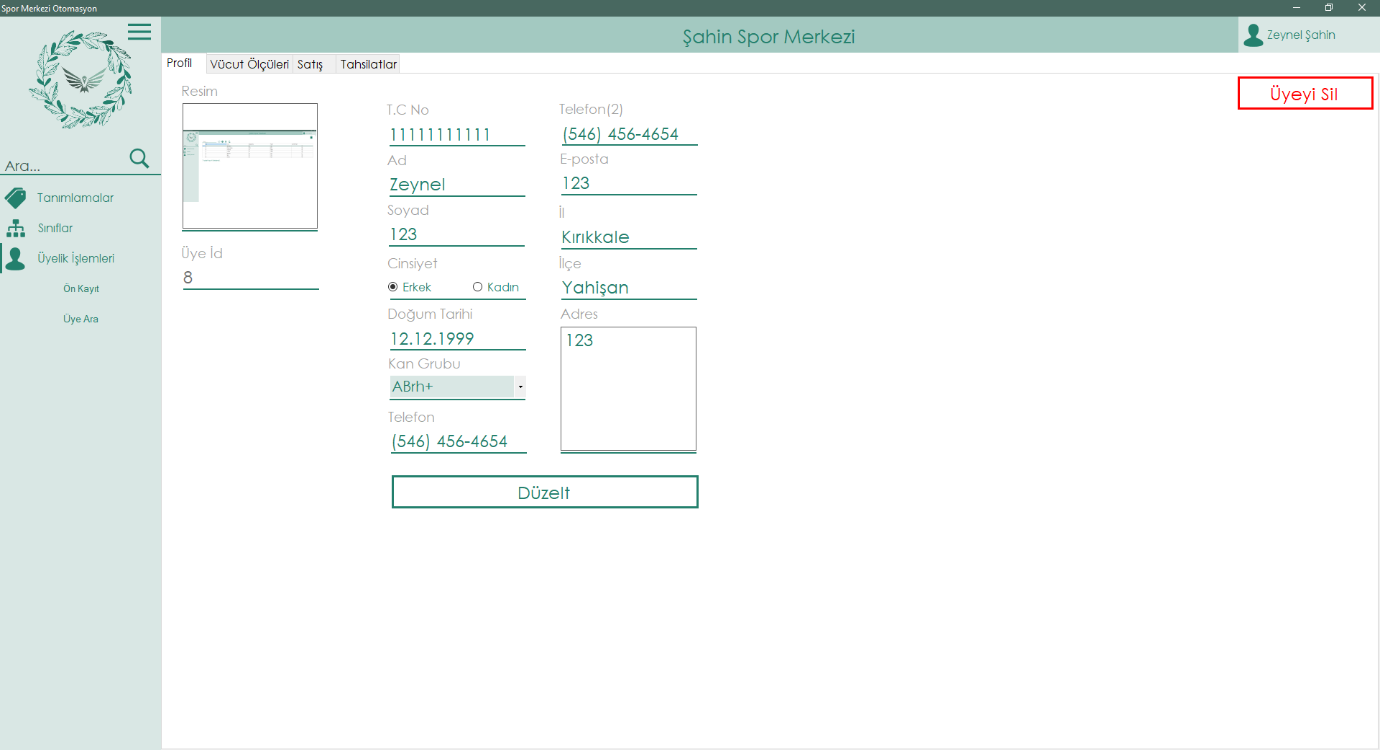
Üyenin detaylı bilgilerine erişmek için DataGridView de üyenin bulunduğu satırı seçip iki kere tıklamanız yeterlidir. Karşınıza çıkan **Üye Detay** ekranında 4 adet sekme bulunur. İlk olarak **Profil sekmesinde**;

* Sil
* Güncelle işlemleri yapılmaktadır.

Bu işlemlere ait veriler uyeler tablosundan getirilip gerekli alanlarda gösterilmektedir. Bu bilgiler;

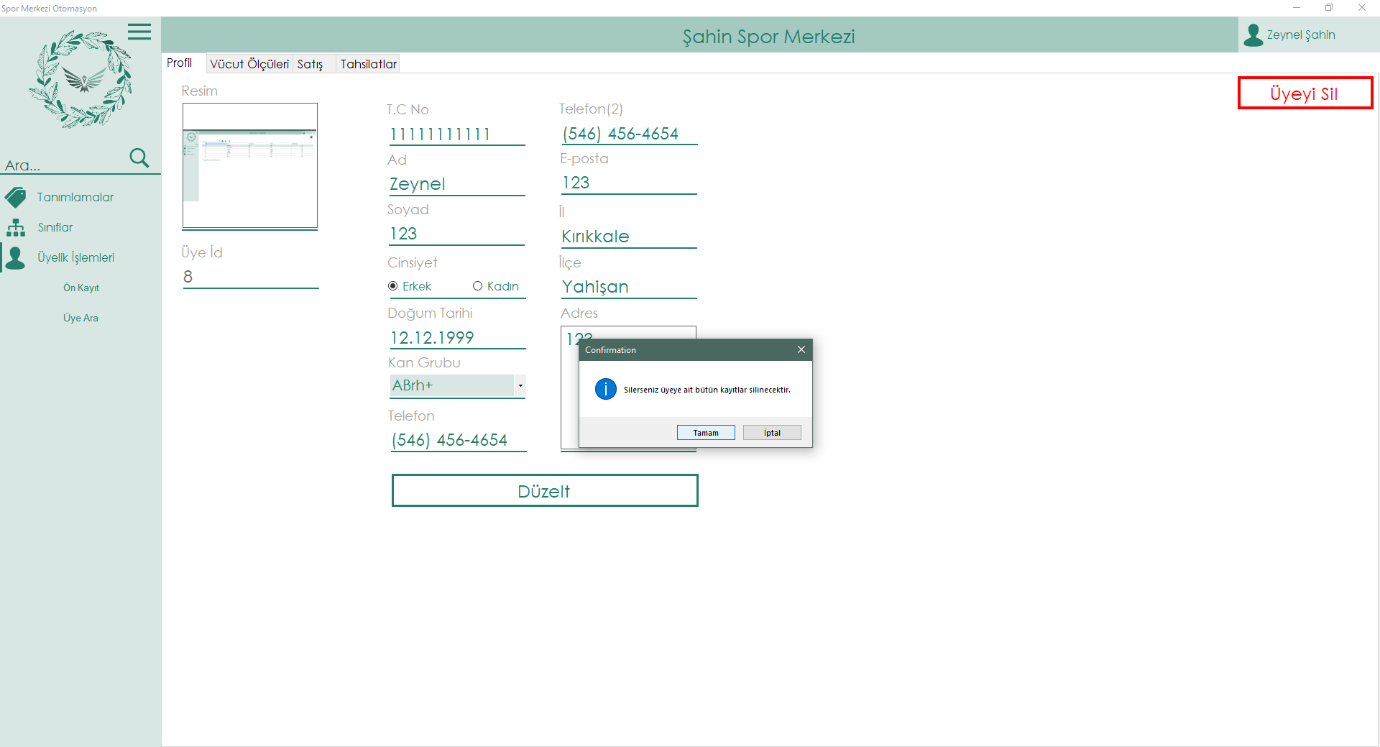
* Üye İd,
* T.C. No,
* Adı,
* Soyadı,
* Cinsiyet,
* Doğum Tarihi,
* İl,
* İlçe,
* Adres,
* Telefon,
* Telefon2,
* E Posta,
* Kan Grubu,
* Resim.

**Tablo 40:** Üye Detayları



Silme İşlemi için Üyeyi Sil butonuna basılır. Karşınıza gelen ekranda onaylama işlemini yaptığınız taktirde üye ve üyeye ait tüm bilgiler kalıcı olarak silinir.

**Tablo 41:** Üye Sil



İkinci olarak **Vücut Ölçüleri sekmesinde**;

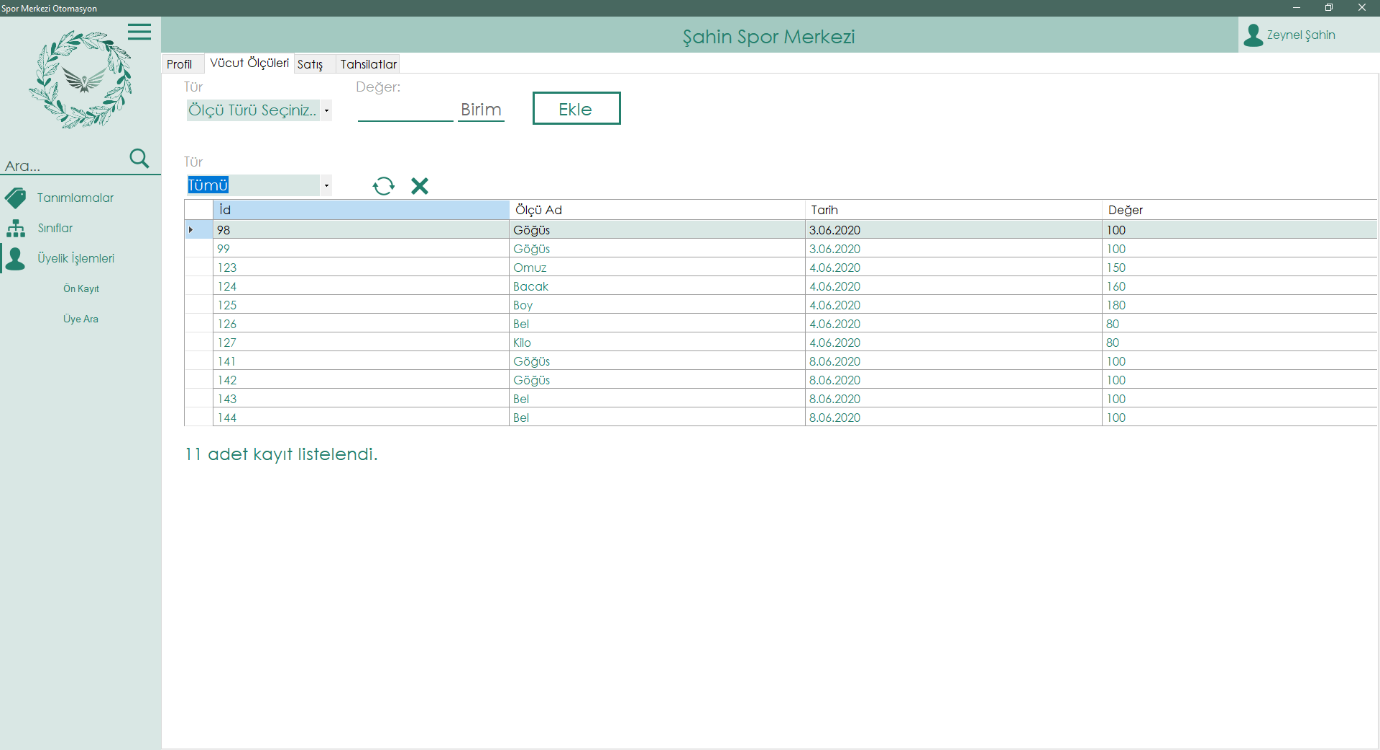
* Ara
* Sil
* Yeni kayıt işlemleri yapılmaktadır

Bu işlemlere ait veriler vucut\_olculeri tablosundan getirilip DataGridView’ de gösterilmektedir. Bu bilgiler;

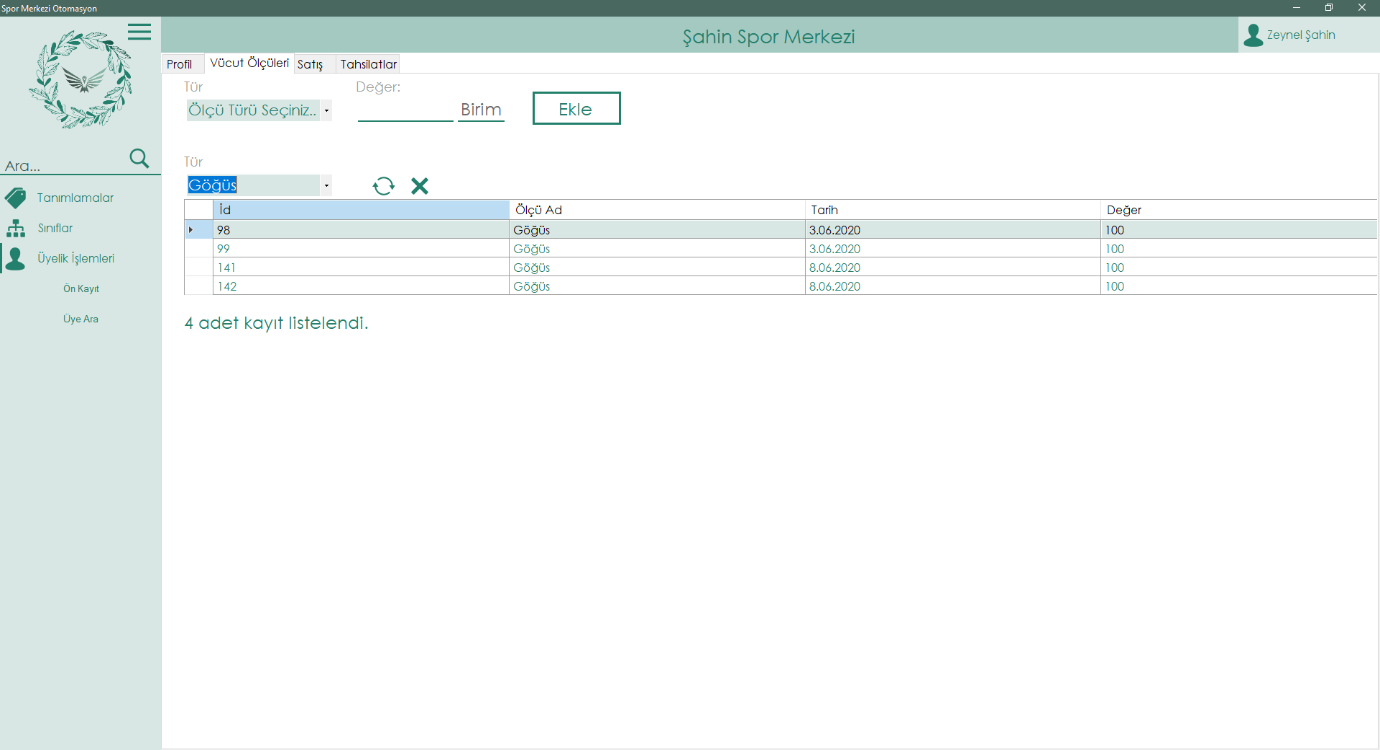
* İd
* Ölçü Adı
* Tarih
* Değer

Ara işlemi için sunulan sadece Ölçü Tür Ad biçiminde aramadır. Tür seçeneklerinden bir tanesin seçilir ve kayıtlar o türe göre listelenir.

**Tablo 42:** Vücut Ölçüleri



**Tablo 43:** Vücut Ölçüleri Arama Sonucu



Silme İşlemi için silme butonuna(X) basılarak silme işlemi gerçekleştirilir.

Ekleme İşlemi için ilk olarak tür seçilir ve değer girilir sonra Ekle butonuna basılarak vücut ölçüsü girilmiş olur.

Üçüncü olarak **Satış sekmesinde**;

* Sil
* Yeni kayıt işlemleri yapılmaktadır

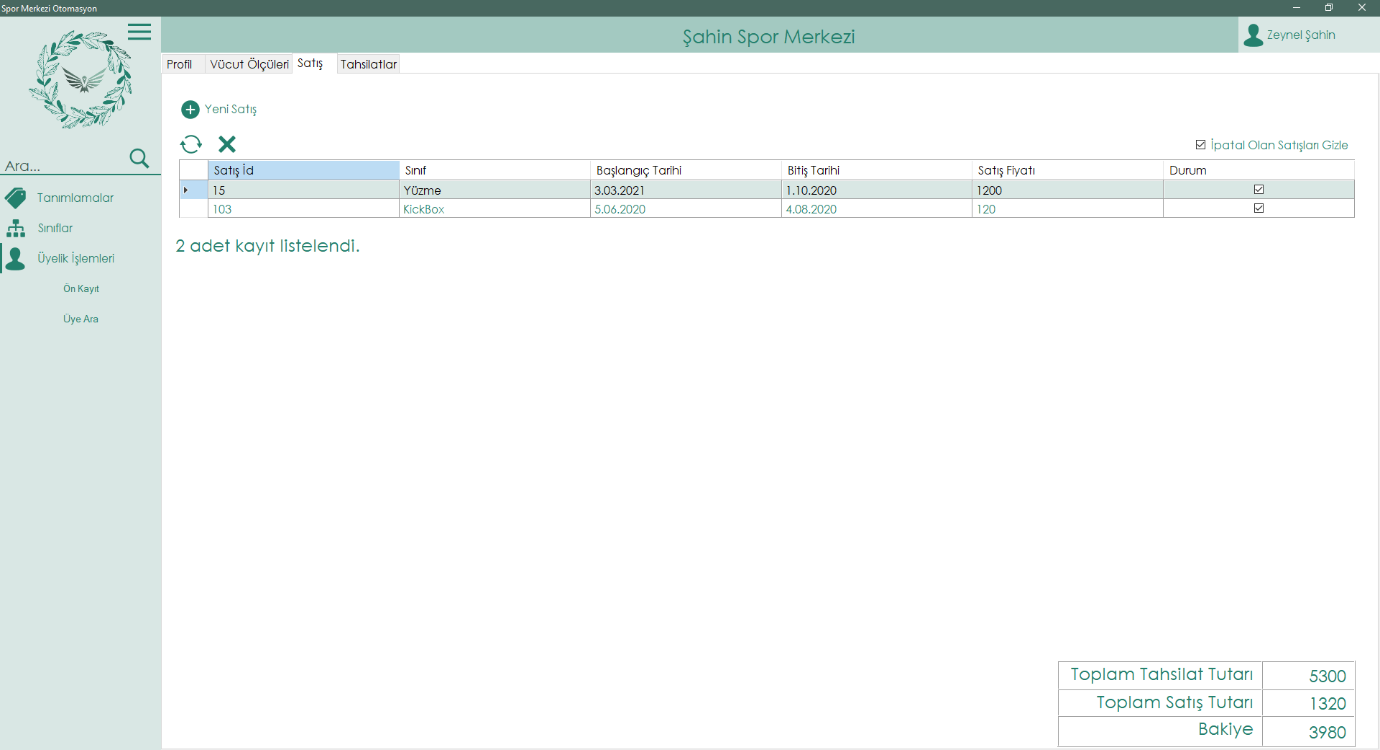
Bu işlemlere ait veriler satislar tablosundan getirilip DataGridView’ de gösterilmektedir. Bu bilgiler;

* Satış İd
* Sınıf
* Başlangıç Tarihi
* Bitiş Tarihi
* Satış Fiyatı
* Durum

Artı olarak bur sekmede satış, tahsilat tutarları ve bakiye gösterilmektedir.

İptal Olan Satışları Gizle kutusunu ister aktif ister pasif yapılarak veriler listelenebilir.

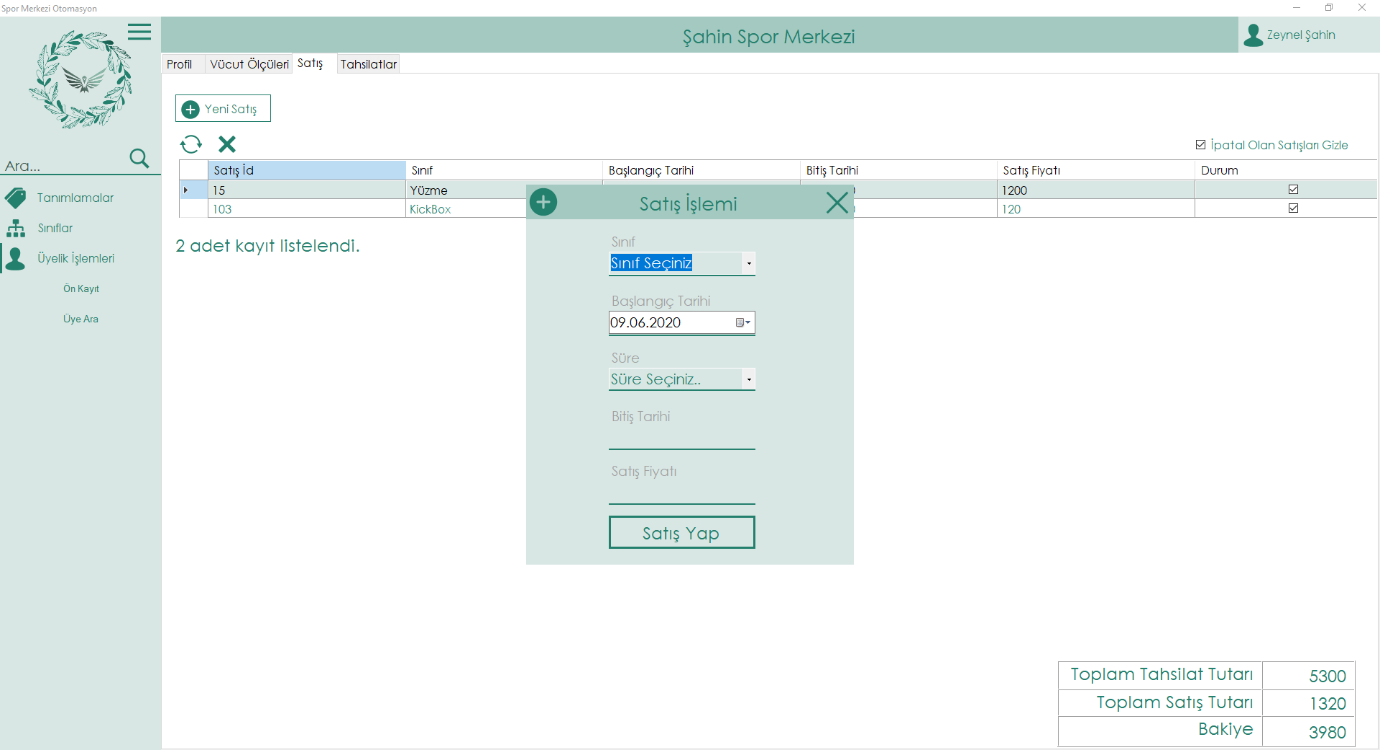
**Tablo 44:** Satış



Silme İşlemi için silme butonuna(X) basılarak silme işlemi gerçekleştirilir.

Ekleme İşlemi için Yeni Satış butonuna basılır. Açılan **Satış İşlemi** formundan tüm bilgileri doldurup satış yap butonuna basmanız yeterlidir.

**Tablo 45:** Satış İşlemi



Son olarak **Tahsilatlar sekmesinde**;

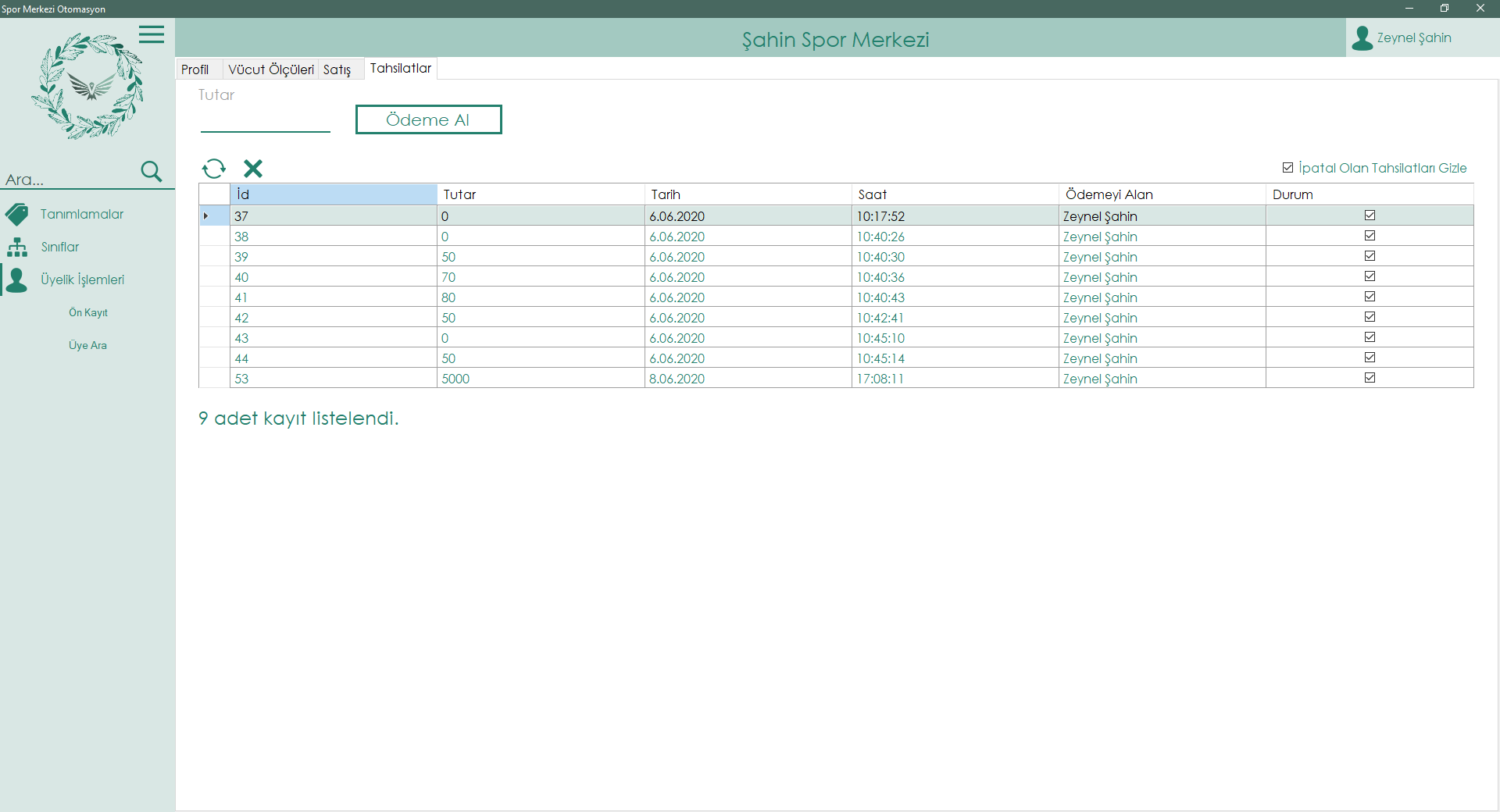
* Sil
* Yeni kayıt işlemleri yapılmaktadır

Bu işlemlere ait veriler tahsilatlar tablosundan getirilip DataGridView’ de gösterilmektedir. Bu bilgiler;

* İd
* Tutar
* Tarih
* Saat
* Ödemeyi Alan
* Durum

İptal Olan Tahsilatları Gizle kutusunu ister aktif ister pasif yapılarak veriler listelenebilir.

**Tablo 46:** Tahsilatlar



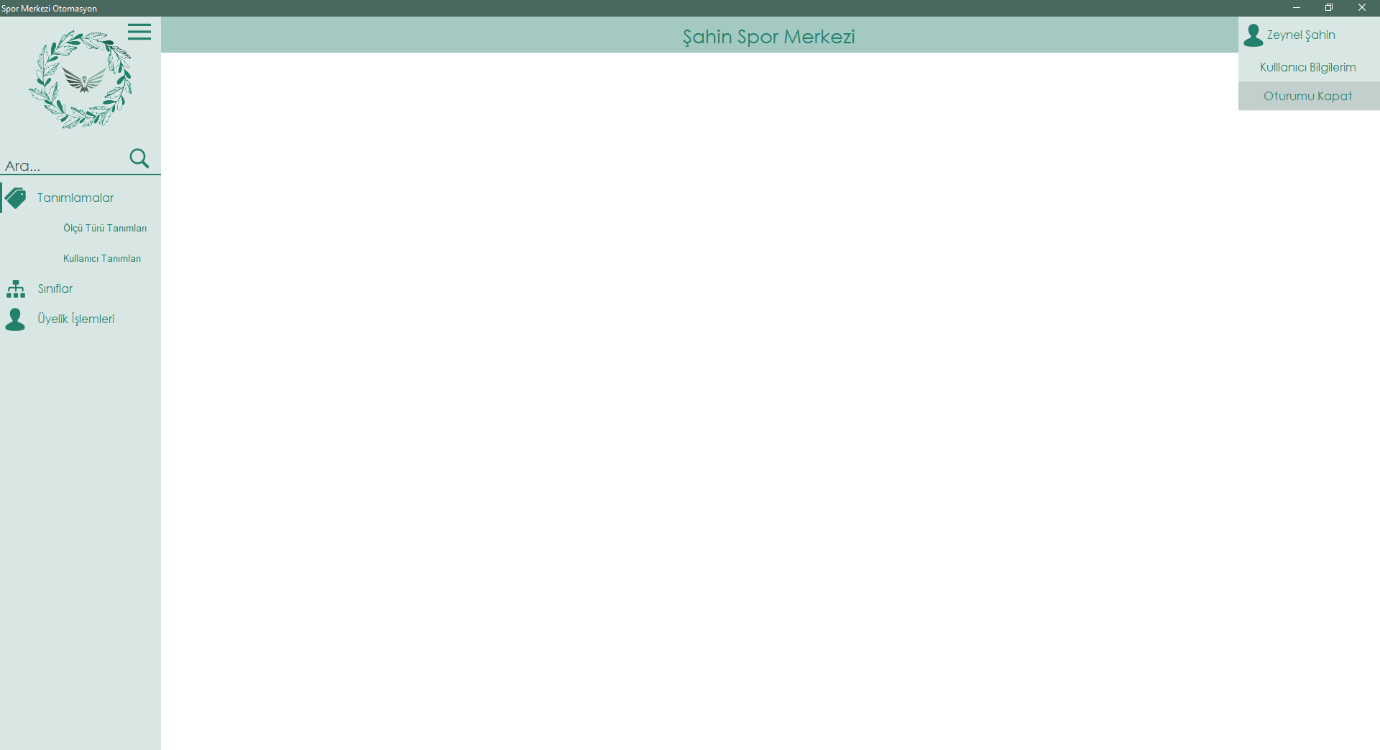
Silme İşlemi için silme butonuna(X) basılarak silme işlemi gerçekleştirilir.

Ekleme İşlemi için değer girilir ve Ödeme Al butonuna basılarak tahsilat girilmiş olur. Alan kişi oturum sahibi olan kişi olarak kaydedilir.

**Oturum Kapatma İşlemi;**

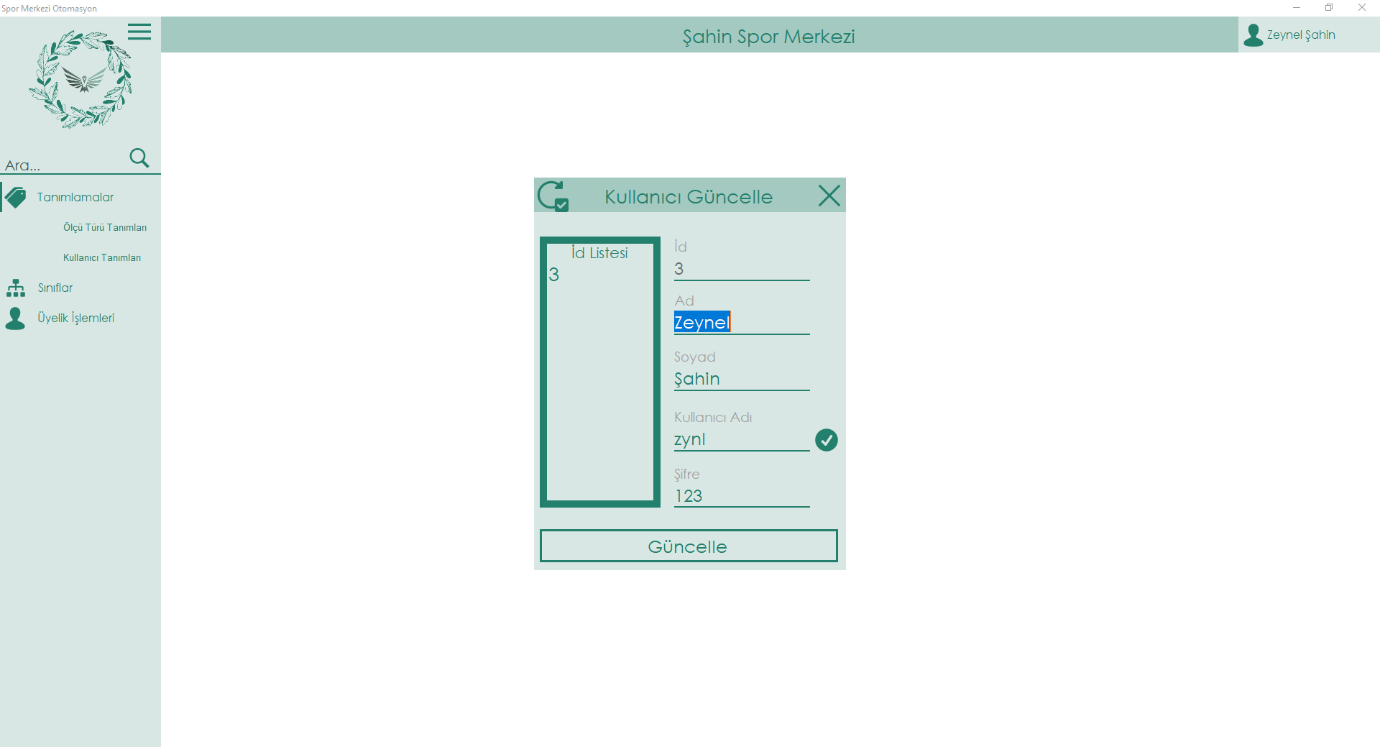
Oturum kapatma işlemi için sağ üstte bulunan Kullanıcı butonuna bir kez tıklayıp açılan menüden Oturumu Kapat butonuna tıklanarak oturum kapatılır.

**Tablo 47:** Oturum Kapatma



Ek olarak Kullanıcı Bilgilerine tıklanarak oturum sahibi kendi bilgilerine ulaşabilir.

**Tablo 48:** Kullanıcı Bilgileri



## Projenin Kodları

### Giriş Ekranı Kodları

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class giris : Form

{

public giris()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

private void button2\_Click(object sender, EventArgs e)

{

string kmt = "select \*from kullanicilar where kullanici\_ad=@p1 and sifre=@p2 ";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", textBox\_kullanici.Text);

komut.Parameters.AddWithValue("@p2", textBox\_sifre.Text);

SqlDataReader da = komut.ExecuteReader();

if (da.Read())

{

this.Hide();

anaSayfa anaSayfa = new anaSayfa();

anaSayfa.kullaniciId = Convert.ToInt32(da["kullanici\_id"]);

anaSayfa.Show();

}

else

{

label\_hata.Show();

}

sqlSinif.Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (textBox\_kullanici.Text != "")

{

label\_kullanici.Hide();

}

else

{

label\_kullanici.Show();

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

if (textBox\_kullanici.Text != "")

{

label\_sifre.Hide();

}

else

{

label\_sifre.Show();

}

}

private void button1\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

### Ana Sayfa Kodları

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class anaSayfa : Form

{

public anaSayfa()

{

InitializeComponent();

}

public int uyeId { get; set; }

public int kullaniciId { get; set; }

private void Panelsifirla()

{

if (panel\_isletme.Height != 0)

{

panel\_isletme.Height = 0;

}

if (panel\_tanımlamalar.Height != 0)

{

panel\_tanımlamalar.Height = 0;

}

if (panel\_uyelikislem.Height != 0)

{

panel\_uyelikislem.Height = 0;

}

}

private void Panelhide()

{

panel\_tanımlamalar.Visible = false;

panel\_uyelikislem.Visible = false;

panel\_isletme.Visible = false;

panel\_cekmece.Visible = false;

}

private void Panelshow()

{

panel\_tanımlamalar.Visible = true;

panel\_uyelikislem.Visible = true;

panel\_isletme.Visible = true;

panel\_cekmece.Visible = true;

}

private void Panelgoster(Panel panel)

{

int sayac = 0;

int panelyuksekligi = 0;

if (panel.Height == 0)

{

Panelsifirla();

foreach (Button item in panel.Controls)//paneldeki butonların yüksekliği alınıyor

{

if (item is Button)

{

sayac++;

}

panelyuksekligi = sayac \* 42;

panel.Height = panelyuksekligi;

}

}

else

{

panel.Height = 0;

}

}

private void Slider(Control buton)

{

panel\_slider.Top = buton.Top;

}

private void Panelesunumekle(Control c)

{

c.Dock = DockStyle.Fill;

panel\_sunum.Controls.Clear();

panel\_sunum.Controls.Add(c);

}

private void Ayar\_kucukmenu()

{

//butonların isimleri küçük menüye hazırlanıyor

button\_tanımlamalar.Text = null;

button\_sınıflar.Text = null;

button\_uyelikislemleri.Text = null;

//butunların resimleri küçük menüye hazırlanıyor

button\_tanımlamalar.ImageAlign = ContentAlignment.MiddleCenter;

button\_sınıflar.ImageAlign = ContentAlignment.MiddleCenter;

button\_uyelikislemleri.ImageAlign = ContentAlignment.MiddleCenter;

}

private void Ayar\_buyukmenu()

{

//butonların isimleri büyük menü tipine hazırlanıyor

button\_tanımlamalar.Text = " Tanımlamalar";

button\_sınıflar.Text = " Sınıflar";

button\_uyelikislemleri.Text = " Üyelik İşlemleri";

//butonların iresimleri sola hizalanıyyor

button\_tanımlamalar.ImageAlign = ContentAlignment.MiddleLeft;

button\_sınıflar.ImageAlign = ContentAlignment.MiddleLeft;

button\_uyelikislemleri.ImageAlign = ContentAlignment.MiddleLeft;

}

private void menukodlar(bool deger)

{

if (deger == true)//

{

panel\_logo.Height = 50;

pictureBox\_logo.SizeMode = PictureBoxSizeMode.Zoom;

Slidertop = panel\_slider.Top - 84;

panel\_slider.Top = Slidertop;

Ayar\_kucukmenu();

Panelhide();

panel\_cekmece.Visible = true;

button\_menualt.Visible = true;

button\_menu.Visible = false;

panel\_solarka.Width = 60;

label\_ara.Hide();

textBox\_ara.Visible = false;

label\_ara.Visible = false;

}

else

{

panel\_logo.Height = 176;

pictureBox\_logo.SizeMode = PictureBoxSizeMode.CenterImage;

panel\_slider.Top = Slidertop + 84;

Panelshow();

Ayar\_buyukmenu();

button\_menualt.Visible = false;

button\_menu.Visible = true;

panel\_cekmece.Visible = false;

panel\_solarka.Width = 224;

textBox\_ara.Visible = true;

label\_ara.Visible = true;

textBox\_ara.Focus();

}

}

public void Acilacakform(Form frm)

{

panel\_sunum.Controls.Clear();

frm.TopLevel = false;

panel\_sunum.Controls.Add(frm);

frm.Show();

frm.Dock = DockStyle.Fill;

frm.BringToFront();

}

private void button\_menualt\_Click(object sender, EventArgs e)

{

menukodlar(false);

}

private void button\_menu\_Click(object sender, EventArgs e)

{

menukodlar(true);

}

public int Slidertop { get; set; }

private void anaSayfa\_Load(object sender, EventArgs e)

{

string kmt = "select ad +' '+soyad from kullanicilar where kullanici\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", kullaniciId);

string adSoyad = komut.ExecuteScalar().ToString();

button\_oturum.Text = adSoyad;

panel\_cekmece.Hide();

Slider(button\_tanımlamalar);

textBox\_ara.Focus();

}

private void textBox\_ara\_TextChanged(object sender, EventArgs e)

{

if (textBox\_ara.Text != "")

{

label\_ara.Visible = false;

}

else

{

label\_ara.Visible = true;

}

}

private void Kucukmenu\_goster(Panel p, string deger)

{

p.Visible = true;

int a;

if (deger == "işletme")

{

a = 138;

p.Location = new Point(0, a + 42);

}

else if (deger == "tanımlamalar")

{

a = 180;

p.Location = new Point(0, a + 42);

}

else if (deger == "üyelikişlemleri")

{

a = 348;

p.Location = new Point(0, a + 42);

}

else if (deger == "raporlar")

{

a = 390;

p.Location = new Point(3, a + 42);

}

}

private void button\_tanımlamalar\_Click(object sender, EventArgs e)

{

Panelgoster(panel\_tanımlamalar);

Slider(button\_tanımlamalar);

Slidertop = button\_tanımlamalar.Top;

}

private void button\_uyelikislemleri\_Click(object sender, EventArgs e)

{

Panelgoster(panel\_uyelikislem);

Slidertop = button\_uyelikislemleri.Top;

Slider(button\_uyelikislemleri);

}

private void button\_sınıflar\_Click(object sender, EventArgs e)

{

Panelsifirla();

Slidertop = button\_sınıflar.Top;

Slider(button\_sınıflar);

siniflar siniflar = new siniflar();

Acilacakform(siniflar);

}

private void anaSayfa\_ResizeBegin(object sender, EventArgs e)

{

panel\_sunum.Hide();

}

private void anaSayfa\_ResizeEnd(object sender, EventArgs e)

{

panel\_sunum.Show();

}

private void button\_olcuturu\_Click(object sender, EventArgs e)

{

tanimlama\_olcu\_tur frm = new tanimlama\_olcu\_tur();

Acilacakform(frm);

}

private void button\_kullanici\_Click(object sender, EventArgs e)

{

tanimlama\_kullanici tanimlama\_Kullanici = new tanimlama\_kullanici();

Acilacakform(tanimlama\_Kullanici);

}

private void button22\_Click(object sender, EventArgs e)

{

uyeEkle uyeEkle = new uyeEkle();

uyeEkle.ShowDialog();

}

private void button21\_Click(object sender, EventArgs e)

{

uyeler uyeler = new uyeler();

uyeler.kullaniciId = kullaniciId;

Acilacakform(uyeler);

}

sqlSinif sqlSinif = new sqlSinif();

private void button\_ara\_Click(object sender, EventArgs e)

{

if (textBox\_ara.Text != "")

{

uyeler uyeler = new uyeler();

uyeler.aranacakAd = textBox\_ara.Text;

uyeler.kullaniciId = kullaniciId;

Acilacakform(uyeler);

}

}

bool durum = true;

private void button2\_Click(object sender, EventArgs e)

{

if (durum == true)

{

panel\_kullanici.Height = 130;

durum = false;

}

else

{

panel\_kullanici.Height = 50;

durum = true;

}

}

private void button\_oturumKapat\_Click(object sender, EventArgs e)

{

this.Hide();

giris giris = new giris();

giris.Show();

}

private void button\_kullaniciBilgi\_Click(object sender, EventArgs e)

{

durum = true;

panel\_kullanici.Height = 50;

sqlSinif.secilenlerListe.Clear();

sqlSinif.secilenlerListe.Add(kullaniciId);

kullaniciGuncelle kullaniciGuncelle = new kullaniciGuncelle();

kullaniciGuncelle.ShowDialog();

}

private void anaSayfa\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

}

}

### Ölçü Türü Tanımlamaları

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class tanimlama\_olcu\_tur : Form

{

public tanimlama\_olcu\_tur()

{

InitializeComponent();

}

string komut = "select olcutur\_id as İd,ad as Ad ,birim as Birim ,durum as'" + "Aktif/Pasif" + "' from olcu\_turleri where durum=1 and silme\_durumu=0";

string komut1 = "select olcutur\_id as İd,ad as Ad ,birim as Birim ,durum as '" + "Aktif/Pasif" + "' from olcu\_turleri where silme\_durumu=0";

string copkutuAdet = "select count(\*) from olcu\_turleri where silme\_durumu=1";

sqlSinif sqlSinif = new sqlSinif();

private void tanımlama\_olcu\_tur\_Load(object sender, EventArgs e)

{

VerilerVeAdet();

textBox\_ara.Focus();

}

void VerilerVeAdet()

{

if (checkBox\_aktif.Checked == true)

{

sqlSinif.veriListeleRenkli(komut, veriListesi, label\_adet, 3);

}

else

{

sqlSinif.veriListeleRenkli(komut1, veriListesi, label\_adet, 3);

}

sqlSinif.copkutuAdet(copkutuAdet, label\_copadet);

}

private void checkBox\_aktif\_CheckedChanged(object sender, EventArgs e)

{

VerilerVeAdet();

string ad = textBox\_ara.Text;

textBox\_ara.Text = "";

textBox\_ara.Text = ad;

}

private void veriListesi\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void button\_ekle\_Click(object sender, EventArgs e)

{

olcuturuEkle frm = new olcuturuEkle();

frm.ShowDialog();

VerilerVeAdet();

}

private void button\_sil\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);//DataGridView deki seçilenler

foreach (var item in sqlSinif.secilenlerListe)

{

int id = Convert.ToInt32(item);

string kmt = "update olcu\_turleri set silme\_durumu=1 where olcutur\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", id);

komut.ExecuteNonQuery();

sqlSinif.Close();

//tekrardan listeyi yenelemek yerine secilenler listeden siliniyor

}

VerilerVeAdet();

}

private void textBox\_ara\_TextChanged(object sender, EventArgs e)

{

string komut = "select olcutur\_id as İd,ad as Ad ,birim as Birim ,durum as'" + "Aktif/Pasif" + "' from olcu\_turleri where durum=1 and silme\_durumu=0";

string komut1 = "select olcutur\_id as İd,ad as Ad ,birim as Birim ,durum as '" + "Aktif/Pasif" + "' from olcu\_turleri where silme\_durumu=0";

if (textBox\_ara.Text != "")

{

label\_ara.Visible = false;

}

else

{

label\_ara.Visible = true;

}

if (checkBox\_aktif.Checked == true)

{

komut += "and ad Like '%" + textBox\_ara.Text + "%'";

sqlSinif.veriListeleRenkli(komut, veriListesi, label\_adet, 3);

}

else

{

komut1 += "and ad Like '%" + textBox\_ara.Text + "%'";

sqlSinif.veriListeleRenkli(komut1, veriListesi, label\_adet, 3);

}

}

private void button\_yenile\_Click(object sender, EventArgs e)

{

VerilerVeAdet();

}

private void button\_copkutusu\_Click(object sender, EventArgs e)

{

copKutusu copKutusu = new copKutusu();

copKutusu.gelenForm = 1;

copKutusu.ShowDialog();

VerilerVeAdet();

}

private void panel\_sunum\_SizeChanged(object sender, EventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void button\_güncelle\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);

olcuTurGuncelle olcuTurGuncelle = new olcuTurGuncelle();

olcuTurGuncelle.ShowDialog();

VerilerVeAdet();

}

private void tanimlama\_olcu\_tur\_Resize(object sender, EventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

}

}

**Ölçü Türü Ekle;**

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class olcuturuEkle : Form

{

public olcuturuEkle()

{

InitializeComponent();

this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;

}

sqlSinif sqlSinif = new sqlSinif();

private void olcuturu\_Ekle\_Load(object sender, EventArgs e)

{

pasifbuttonfgizle();

}

private void textBox\_olcuadi\_TextChanged(object sender, EventArgs e)

{

if (textBox\_olcuadi.Text != "")

{

label\_olcuadi.Visible = false;

}

else

{

label\_olcuadi.Visible = true;

}

}

private bool formboyut = true;

private void button3\_Click(object sender, EventArgs e)

{

if (formboyut == true)

{

this.WindowState = FormWindowState.Maximized;

formboyut = false;

}

else

{

this.WindowState = FormWindowState.Normal;

formboyut = true;

}

}

private void button2\_Click(object sender, EventArgs e)

{

timer\_sonuc.Stop();

this.Close();

}

private void button5\_Click(object sender, EventArgs e)

{

this.WindowState = FormWindowState.Minimized;

}

private void button\_aktif\_Click(object sender, EventArgs e)

{

aktifbuttongizle();

pasifbuttongöster();

}

private void aktifbuttongizle()

{

button\_aktif.BackColor = Color.Transparent;

button\_aktif.Text = "";

button\_aktif.Enabled = false;

}

private void pasifbuttongöster()

{

button\_pasif.Enabled = true;

button\_pasif.BackColor = Color.FromArgb(108, 170, 157);

button\_pasif.Text = "Pasif";

}

private void pasifbuttonfgizle()

{

button\_pasif.Enabled = false;

button\_pasif.BackColor = Color.Transparent;

button\_pasif.Text = "";

}

private void button\_pasif\_Click(object sender, EventArgs e)

{

pasifbuttonfgizle();

aktifbuttongöster();

}

private void aktifbuttongöster()

{

button\_aktif.BackColor = Color.FromArgb(35, 128, 109);

button\_aktif.Text = "Aktif";

button\_aktif.Enabled = true;

}

private void textBox\_birimi\_TextChanged\_1(object sender, EventArgs e)

{

if (textBox\_birimi.Text != "")

{

label\_birimi.Visible = false;

}

else

{

label\_birimi.Visible = true;

}

}

private void button\_ekle\_Click(object sender, EventArgs e)

{

if (textBox\_birimi.Text != "" || textBox\_birimi.Text != null && textBox\_olcuadi.Text != "" || textBox\_olcuadi.Text != null)

{

string kmt = "insert into olcu\_turleri (ad,birim,durum) values(@p1,@p2,@p3) ";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", textBox\_olcuadi.Text);

komut.Parameters.AddWithValue("@p2", textBox\_birimi.Text);

if (button\_aktif.Enabled == true)

{

komut.Parameters.AddWithValue("@p3", 1);

}

else

{

komut.Parameters.AddWithValue("@p3", 0);

}

komut.ExecuteNonQuery();

sqlSinif.Close();

timer\_sonuc.Start();

timer\_sonuc.Interval = 10;

textBox\_birimi.Text = "";

textBox\_olcuadi.Text = "";

}

textBox\_olcuadi.Focus();

}

private int sayac = 0;

private void timer\_sonuc\_Tick(object sender, EventArgs e)

{

if (sayac == 20)

{

label\_sonuc.Text = "";

timer\_sonuc.Stop();

sayac = 0;

}

else

{

sayac++;

label\_sonuc.Text = "Kayıt İşlemi Gerçekleşti";

}

}

}

}

**Ölçü Türü Güncelle;**

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class olcuTurGuncelle : Form

{

public olcuTurGuncelle()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

private void olcuTurGuncelle\_Load(object sender, EventArgs e)

{

foreach (var item in sqlSinif.secilenlerListe)

{

listBox\_veriler.Items.Add(item);

}

if (listBox\_veriler.Items.Count > 0)

{

string komut = "";

komut = "select olcutur\_id , ad,birim,durum from olcu\_turleri where olcutur\_id= ('" + listBox\_veriler.Items[0] + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac()); ;

SqlDataReader dr = kmt.ExecuteReader();

if (dr.Read())

{

textBox\_id.Text = listBox\_veriler.Items[0].ToString();

textBox\_olcu.Text = dr["ad"].ToString();

textBox\_birim.Text = dr["birim"].ToString();

if (dr["durum"].ToString() == "False")

{

aktifbuttongizle();

pasifbuttongöster();

}

else

{

pasifbuttonfgizle();

aktifbuttongöster();

}

}

sqlSinif.Close();

}

}

private void aktifbuttongizle()

{

button\_aktif.BackColor = Color.Transparent;

button\_aktif.Text = "";

button\_aktif.Enabled = false;

}

private void pasifbuttongöster()

{

button\_pasif.Enabled = true;

button\_pasif.BackColor = Color.FromArgb(108, 170, 157);

button\_pasif.Text = "Pasif";

}

private void pasifbuttonfgizle()

{

button\_pasif.Enabled = false;

button\_pasif.BackColor = Color.Transparent;

button\_pasif.Text = "";

}

private void button\_pasif\_Click(object sender, EventArgs e)

{

pasifbuttonfgizle();

aktifbuttongöster();

}

private void aktifbuttongöster()

{

button\_aktif.BackColor = Color.FromArgb(35, 128, 109);

button\_aktif.Text = "Aktif";

button\_aktif.Enabled = true;

}

private void button\_guncelle\_Click(object sender, EventArgs e)

{

if (listBox\_veriler.Items.Count > 0)

{

string komut = "update olcu\_turleri set ad=@p1, birim=@p2 ,durum=@p3 where olcutur\_id= ('" + textBox\_id.Text + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac());

kmt.Parameters.AddWithValue("@p1", textBox\_olcu.Text);

kmt.Parameters.AddWithValue("@p2", textBox\_birim.Text);

if (button\_aktif.Enabled == true)

{

kmt.Parameters.AddWithValue("@p3", "True");

}

else

{

kmt.Parameters.AddWithValue("@p3", "False");

}

kmt.ExecuteNonQuery();

sqlSinif.Close();

MessageBox.Show("Kayıt Güncellendi");

}

}

private void button\_aktif\_Click(object sender, EventArgs e)

{

aktifbuttongizle();

pasifbuttongöster();

}

private void button\_pasif\_Click\_1(object sender, EventArgs e)

{

pasifbuttonfgizle();

aktifbuttongöster();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void listBox\_veriler\_SelectedIndexChanged(object sender, EventArgs e)

{

if (listBox\_veriler.Items.Count > 0)

{

string komut = "";

komut = "select olcutur\_id , ad,birim,durum from olcu\_turleri where olcutur\_id= ('" + listBox\_veriler.SelectedItem + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac()); ;

SqlDataReader dr = kmt.ExecuteReader();

if (dr.Read())

{

textBox\_id.Text = listBox\_veriler.SelectedItem.ToString();

textBox\_olcu.Text = dr["ad"].ToString();

textBox\_birim.Text = dr["birim"].ToString();

if (dr["durum"].ToString() == "False")

{

aktifbuttongizle();

pasifbuttongöster();

}

else

{

pasifbuttonfgizle();

aktifbuttongöster();

}

}

sqlSinif.Close();

}

}

}

}

### Kullanıcı Tanımlamaları

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class tanimlama\_kullanici : Form

{

public tanimlama\_kullanici()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

string komut = "select kullanici\_id as İd, kullanici\_ad as '" + "Kullanıcı Adı" + "', sifre as Şifre, ad as Ad,soyad as Soyad from kullanicilar where silme\_durumu=0";

string copkutuAdet = "select count(\*) from kullanicilar where silme\_durumu=1";

void VerilerVeAdet()

{

sqlSinif.veriListele(komut, veriListesi, label\_adet);

sqlSinif.copkutuAdet(copkutuAdet, label\_copadet);

}

private void veriListesi\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void button\_ekle\_Click(object sender, EventArgs e)

{

kullaniciEkle kullaniciEkle = new kullaniciEkle();

kullaniciEkle.ShowDialog();

sqlSinif.veriListele(komut, veriListesi, label\_adet);

}

private void button\_yenile\_Click(object sender, EventArgs e)

{

VerilerVeAdet();

}

private void button\_copkutusu\_Click(object sender, EventArgs e)

{

copKutusu copKutusu = new copKutusu();

copKutusu.gelenForm = 2;

copKutusu.ShowDialog();

VerilerVeAdet();

}

private void tanimlama\_kullanici\_Load(object sender, EventArgs e)

{

VerilerVeAdet();

textBox\_ara.Focus();

}

private void textBox\_ara\_TextChanged(object sender, EventArgs e)

{

string komut = "select kullanici\_id as İd, kullanici\_ad as '" + "Kullanıcı Adı" + "', sifre as Şifre, ad as Ad,soyad as Soyad from kullanicilar where silme\_durumu=0";

if (textBox\_ara.Text != "")

{

label\_ara.Visible = false;

}

else

{

label\_ara.Visible = true;

}

komut += "and kullanici\_ad Like '%" + textBox\_ara.Text + "%'";

sqlSinif.veriListele(komut, veriListesi, label\_adet);

}

private void button\_sil\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);//DataGridView deki seçilenler

foreach (var item in sqlSinif.secilenlerListe)

{

int id = Convert.ToInt32(item);

string kmt = "update kullanicilar set silme\_durumu=1 where kullanici\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", id);

komut.ExecuteNonQuery();

sqlSinif.Close();

//tekrardan listeyi yenelemek yerine secilenler listeden siliniyor

}

sqlSinif.secilenleriListedenSil(veriListesi, label\_adet);

sqlSinif.copkutuAdet(copkutuAdet, label\_copadet);

}

private void button\_güncelle\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);

kullaniciGuncelle kullaniciGuncelle = new kullaniciGuncelle();

kullaniciGuncelle.ShowDialog();

VerilerVeAdet();

}

private void tanimlama\_kullanici\_Resize(object sender, EventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void panel\_sunum\_SizeChanged(object sender, EventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

}

}

**Kullanıcı Ekle;**

using sporsalonu.siniflar;

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class kullaniciEkle : Form

{

public kullaniciEkle()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

bool durum = false;

private void button\_ekle\_Click(object sender, EventArgs e)

{

if (textBox\_ad.Text != "" || textBox\_ad.Text != null && textBox\_soyad.Text != "" || textBox\_soyad.Text != null && textBox\_sifre.Text != "" || textBox\_sifre.Text != null && textBox\_kullanici.Text != "" || textBox\_kullanici.Text != null)

{

if (durum == true)

{

string kmt = "insert into kullanicilar (kullanici\_ad,sifre,ad,soyad) values(@p1,@p2,@p3,@p4) ";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", textBox\_kullanici.Text);

komut.Parameters.AddWithValue("@p2", textBox\_sifre.Text);

komut.Parameters.AddWithValue("@p3", textBox\_ad.Text);

komut.Parameters.AddWithValue("@p4", textBox\_soyad.Text);

komut.ExecuteNonQuery();

MessageBox.Show("Kullanıcı Kaydedildi");

sqlSinif.Close();

}

else

{

textBox\_kullanici.Focus();

}

}

else

{

MessageBox.Show("Boş alanlar var");

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void kullaniciEkle\_Load(object sender, EventArgs e)

{

label\_ad.Visible = false;

label\_kullanici.Visible = false;

label\_sifre.Visible = false;

label\_soyad.Visible = false;

}

private void textBox\_kullanici\_TextChanged(object sender, EventArgs e)

{

string kmt = "select kullanici\_ad from kullanicilar where kullanici\_ad='" + textBox\_kullanici.Text + "'";

SqlCommand komut2 = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

SqlDataAdapter da = new SqlDataAdapter(komut2);

DataTable dt = new DataTable();

da.Fill(dt);

sqlSinif.Close();

if (dt.Rows.Count > 0)

{

pictureBox\_durum.Image = global::sporsalonu.Properties.Resources.yanlis;

durum = false;

}

else

{

pictureBox\_durum.Image = global::sporsalonu.Properties.Resources.dogru;

durum = true;

}

}

}

}

**Kullanıcı Güncelle;**

using sporsalonu.siniflar;

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class kullaniciGuncelle : Form

{

public kullaniciGuncelle()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

private void listBox\_veriler\_SelectedIndexChanged(object sender, EventArgs e)

{

ilkDurum = false;

if (listBox\_veriler.Items.Count > 0)

{

string komut = "";

komut = "select kullanici\_id , kullanici\_ad, sifre , ad,soyad from kullanicilar where kullanici\_id= ('" + listBox\_veriler.SelectedItem + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac()); ;

SqlDataReader dr = kmt.ExecuteReader();

if (dr.Read())

{

textBox\_id.Text = listBox\_veriler.SelectedItem.ToString();

textBox\_kullanici.Text = dr["kullanici\_ad"].ToString();

textBox\_sifre.Text = dr["sifre"].ToString();

textBox\_soyad.Text = dr["soyad"].ToString();

textBox\_ad.Text = dr["ad"].ToString();

}

sqlSinif.Close();

ilkDurum = true;

}

}

private void button\_guncelle\_Click(object sender, EventArgs e)

{

if (listBox\_veriler.Items.Count > 0)

{

if (durum == true)

{

string komut = "update kullanicilar set kullanici\_ad=@p1, sifre=@p2 ,ad=@p3,soyad=@p4 where kullanici\_id= ('" + textBox\_id.Text + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac());

kmt.Parameters.AddWithValue("@p1", textBox\_kullanici.Text);

kmt.Parameters.AddWithValue("@p2", textBox\_sifre.Text);

kmt.Parameters.AddWithValue("@p3", textBox\_ad.Text);

kmt.Parameters.AddWithValue("@p4", textBox\_soyad.Text);

kmt.ExecuteNonQuery();

sqlSinif.Close();

MessageBox.Show("Kayıt Güncellendi");

}

else

{

textBox\_kullanici.Focus();

}

}

}

public string kullaniciAdi { get; set; }

private void kullaniciGuncell\_Load(object sender, EventArgs e)

{

foreach (var item in sqlSinif.secilenlerListe)

{

listBox\_veriler.Items.Add(item);

}

if (listBox\_veriler.Items.Count > 0)

{

string komut = "";

komut = "select kullanici\_id , kullanici\_ad, sifre , ad,soyad from kullanicilar where kullanici\_id= ('" + listBox\_veriler.Items[0] + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac()); ;

SqlDataReader dr = kmt.ExecuteReader();

if (dr.Read())

{

textBox\_id.Text = listBox\_veriler.Items[0].ToString();

textBox\_kullanici.Text = dr["kullanici\_ad"].ToString();

textBox\_sifre.Text = dr["sifre"].ToString();

textBox\_soyad.Text = dr["soyad"].ToString();

textBox\_ad.Text = dr["ad"].ToString();

kullaniciAdi = dr["kullanici\_ad"].ToString();

}

ilkDurum = true;

sqlSinif.Close();

pictureBox\_durum.Image = global::sporsalonu.Properties.Resources.dogru;

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

bool durum = false;

bool ilkDurum = false;

private void textBox\_kullanici\_TextChanged(object sender, EventArgs e)

{

if (ilkDurum == true)

{

if (textBox\_kullanici.Text == kullaniciAdi)

{

pictureBox\_durum.Image = global::sporsalonu.Properties.Resources.dogru;

durum = true;

}

else

{

string kmt = "select kullanici\_ad from kullanicilar where kullanici\_ad='" + textBox\_kullanici.Text + "'";

SqlCommand komut2 = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

SqlDataAdapter da = new SqlDataAdapter(komut2);

DataTable dt = new DataTable();

da.Fill(dt);

sqlSinif.Close();

if (dt.Rows.Count > 0)

{

pictureBox\_durum.Image = global::sporsalonu.Properties.Resources.yanlis;

durum = false;

}

else

{

pictureBox\_durum.Image = global::sporsalonu.Properties.Resources.dogru;

durum = true;

}

}

}

}

}

}

### Sınıflar

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class siniflar : Form

{

public siniflar()

{

InitializeComponent();

}

string komut = "select sinif\_id as İd, ad as '" + "Ad" + "', kapasite as Kapasite,fiyat as Fiyat, durum '" + "Aktif/Pasif" + "' from siniflar where silme\_durumu=0";

string copkutuAdet = "select count(\*) from siniflar where silme\_durumu=1";

sqlSinif sqlSinif = new sqlSinif();

void VerilerVeAdet()

{

sqlSinif.veriListele(komut, veriListesi, label\_adet);

sqlSinif.copkutuAdet(copkutuAdet, label\_copadet);

}

private void siniflar\_Load(object sender, EventArgs e)

{

VerilerVeAdet();

}

private void veriListesi\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void button\_ekle\_Click(object sender, EventArgs e)

{

sinifEkle sinifEkle = new sinifEkle();

sinifEkle.ShowDialog();

VerilerVeAdet();

}

private void button\_yenile\_Click(object sender, EventArgs e)

{

VerilerVeAdet();

}

private void button\_copkutusu\_Click(object sender, EventArgs e)

{

copKutusu copKutusu = new copKutusu();

copKutusu.gelenForm = 3;

copKutusu.ShowDialog();

VerilerVeAdet();

}

private void button\_sil\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);//DataGridView deki seçilenler

foreach (var item in sqlSinif.secilenlerListe)

{

int id = Convert.ToInt32(item);

string kmt = "update siniflar set silme\_durumu=1 where sinif\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", id);

komut.ExecuteNonQuery();

sqlSinif.Close();

//tekrardan listeyi yenelemek yerine secilenler listeden siliniyor

}

sqlSinif.secilenleriListedenSil(veriListesi, label\_adet);

sqlSinif.copkutuAdet(copkutuAdet, label\_copadet);

}

private void button\_güncelle\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);

sinifGuncelle sinifGuncelle = new sinifGuncelle();

sinifGuncelle.ShowDialog();

VerilerVeAdet();

}

private void textBox\_ara\_TextChanged(object sender, EventArgs e)

{

if (textBox\_ara.Text != "")

{

label\_ara.Visible = false;

}

else

{

label\_ara.Visible = true;

}

string komut = "select sinif\_id as İd, ad as '" + "Ad" + "', kapasite as Kapasite,fiyat as Fiyat, durum '" + "Aktif/Pasif" + "' from siniflar where silme\_durumu=0";

komut += "and ad Like '%" + textBox\_ara.Text + "%'";

sqlSinif.veriListele(komut, veriListesi, label\_adet);

}

}

}

**Sınıf Ekle;**

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class sinifEkle : Form

{

public sinifEkle()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

private void button\_ekle\_Click(object sender, EventArgs e)

{

if (textBox\_ad.Text != "" || textBox\_ad.Text != null && textBox\_kapasite.Text != "" || textBox\_kapasite.Text != null && textBox\_fiyat.Text != "" || textBox\_fiyat.Text != null)

{

string kmt = "insert into siniflar (ad,kapasite,durum,fiyat) values(@p1,@p2,@p3,@p4) ";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", textBox\_ad.Text);

komut.Parameters.AddWithValue("@p2", textBox\_kapasite.Text);

if (button\_aktif.Enabled == true)

{

komut.Parameters.AddWithValue("@p3", 1);

}

else

{

komut.Parameters.AddWithValue("@p3", 0);

}

komut.Parameters.AddWithValue("@p4", textBox\_fiyat.Text); ;

komut.ExecuteNonQuery();

sqlSinif.Close();

textBox\_ad.Text = "";

textBox\_kapasite.Text = "";

textBox\_fiyat.Text = "";

}

textBox\_ad.Focus();

}

private void aktifbuttongöster()

{

button\_aktif.BackColor = Color.FromArgb(35, 128, 109);

button\_aktif.Text = "Aktif";

button\_aktif.Enabled = true;

}

private void aktifbuttongizle()

{

button\_aktif.BackColor = Color.Transparent;

button\_aktif.Text = "";

button\_aktif.Enabled = false;

}

private void pasifbuttongöster()

{

button\_pasif.Enabled = true;

button\_pasif.BackColor = Color.FromArgb(108, 170, 157);

button\_pasif.Text = "Pasif";

}

private void pasifbuttonfgizle()

{

button\_pasif.Enabled = false;

button\_pasif.BackColor = Color.Transparent;

button\_pasif.Text = "";

}

private int sayac = 0;

private void timer\_sonuc\_Tick(object sender, EventArgs e)

{

if (sayac == 20)

{

label\_sonuc.Text = "";

timer\_sonuc.Stop();

sayac = 0;

}

else

{

sayac++;

label\_sonuc.Text = "Kayıt İşlemi Gerçekleşti";

}

}

private void sinifEkle\_Load(object sender, EventArgs e)

{

pasifbuttonfgizle();

label\_ad.Visible = false;

label\_fiyat.Visible = false;

label\_kapasite.Visible = false;

}

private void button\_aktif\_Click(object sender, EventArgs e)

{

aktifbuttongizle();

pasifbuttongöster();

}

private void button\_pasif\_Click(object sender, EventArgs e)

{

pasifbuttonfgizle();

aktifbuttongöster();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

**Sınıf Güncelle;**

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class sinifGuncelle : Form

{

public sinifGuncelle()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

private void listBox\_veriler\_SelectedIndexChanged(object sender, EventArgs e)

{

if (listBox\_veriler.Items.Count > 0)

{

string komut = "";

komut = "select ad ,kapasite,fiyat,durum from siniflar where sinif\_id= ('" + listBox\_veriler.SelectedItem + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac()); ;

SqlDataReader dr = kmt.ExecuteReader();

if (dr.Read())

{

textBox\_id.Text = listBox\_veriler.SelectedItem.ToString();

textBox\_ad.Text = dr["ad"].ToString();

textBox\_kapasite.Text = dr["kapasite"].ToString();

textBox\_fiyat.Text = dr["fiyat"].ToString();

if (dr["durum"].ToString() == "False")

{

aktifbuttongizle();

pasifbuttongöster();

}

else

{

pasifbuttonfgizle();

aktifbuttongöster();

}

}

sqlSinif.Close();

}

}

private void button\_guncelle\_Click(object sender, EventArgs e)

{

if (listBox\_veriler.Items.Count > 0)

{

string komut = "update siniflar set ad=@p1, kapasite=@p2 ,durum=@p3,fiyat=@p4 where sinif\_id= ('" + textBox\_id.Text + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac());

kmt.Parameters.AddWithValue("@p1", textBox\_ad.Text);

kmt.Parameters.AddWithValue("@p2", textBox\_kapasite.Text);

if (button\_aktif.Enabled == true)

{

kmt.Parameters.AddWithValue("@p3", "True");

}

else

{

kmt.Parameters.AddWithValue("@p3", "False");

}

kmt.Parameters.AddWithValue("@p4", textBox\_fiyat.Text);

kmt.ExecuteNonQuery();

sqlSinif.Close();

MessageBox.Show("Kayıt Güncellendi");

}

}

private void aktifbuttongizle()

{

button\_aktif.BackColor = Color.Transparent;

button\_aktif.Text = "";

button\_aktif.Enabled = false;

}

private void pasifbuttongöster()

{

button\_pasif.Enabled = true;

button\_pasif.BackColor = Color.FromArgb(108, 170, 157);

button\_pasif.Text = "Pasif";

}

private void pasifbuttonfgizle()

{

button\_pasif.Enabled = false;

button\_pasif.BackColor = Color.Transparent;

button\_pasif.Text = "";

}

private void button\_pasif\_Click(object sender, EventArgs e)

{

pasifbuttonfgizle();

aktifbuttongöster();

}

private void aktifbuttongöster()

{

button\_aktif.BackColor = Color.FromArgb(35, 128, 109);

button\_aktif.Text = "Aktif";

button\_aktif.Enabled = true;

}

private void sinifGuncell\_Load(object sender, EventArgs e)

{

foreach (var item in sqlSinif.secilenlerListe)

{

listBox\_veriler.Items.Add(item);

}

if (listBox\_veriler.Items.Count > 0)

{

string komut = "";

komut = "select ad ,kapasite,fiyat,durum from siniflar where sinif\_id= ('" + listBox\_veriler.Items[0] + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac()); ;

SqlDataReader dr = kmt.ExecuteReader();

if (dr.Read())

{

textBox\_id.Text = listBox\_veriler.Items[0].ToString();

textBox\_ad.Text = dr["ad"].ToString();

textBox\_kapasite.Text = dr["kapasite"].ToString();

textBox\_fiyat.Text = dr["fiyat"].ToString();

if (dr["durum"].ToString() == "False")

{

aktifbuttongizle();

pasifbuttongöster();

}

else

{

pasifbuttonfgizle();

aktifbuttongöster();

}

}

sqlSinif.Close();

}

}

private void button\_aktif\_Click(object sender, EventArgs e)

{

aktifbuttongizle();

pasifbuttongöster();

}

private void button\_pasif\_Click\_1(object sender, EventArgs e)

{

pasifbuttonfgizle();

aktifbuttongöster();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

### Ön Kayıt

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class uyeEkle : Form

{

public uyeEkle()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

sqlSinif sqlSinif = new sqlSinif();

public string resim { get; set; }

private void button\_ekle\_Click(object sender, EventArgs e)

{

string kmt = "insert into uyeler (tcno,ad,soyad,cinsiyet,dogumtarihi,adres,il,ilce,telefon,telefon2,eposta,kangrubu) values (@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8,@p9,@p10,@p11,@p12)";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", maskedTextBox\_tcno.Text);

komut.Parameters.AddWithValue("@p2", textBox\_ad.Text);

komut.Parameters.AddWithValue("@p3", textBox\_soyad.Text);

if (radioButton\_erkek.Checked)

{

komut.Parameters.AddWithValue("@p4", "Erkek");

}

else

{

komut.Parameters.AddWithValue("@p4", "Kadın");

}

komut.Parameters.AddWithValue("@p5", maskedTextBox\_dogumta.Text);

komut.Parameters.AddWithValue("@p6", textBox\_adres.Text);

komut.Parameters.AddWithValue("@p7", textBox\_il.Text);

komut.Parameters.AddWithValue("@p8", textBox\_ilce.Text);

komut.Parameters.AddWithValue("@p9", maskedTextBox1\_telefon.Text);

komut.Parameters.AddWithValue("@p10", maskedTextBox\_telefon2.Text);

komut.Parameters.AddWithValue("@p11", textBox4\_eposta.Text);

komut.Parameters.AddWithValue("@p12", comboBox\_kangrub.Text);

komut.ExecuteNonQuery();

sqlSinif.Close();

MessageBox.Show("Üye Eklendi");

textBox\_ad.Focus();

}

}

}

### Üye Ara

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class uyeAra : Form

{

public uyeAra()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

string komut = "select uye\_id as İd,tcno as '" + "TC.No" + "',ad as Ad,soyad as Soyad,cinsiyet as Cinsiet,telefon as Telefon ,eposta as '" + "E-posta" + "' from uyeler where silme\_durumu=0";

string copkutuAdet = "select count(\*) from uyeler where silme\_durumu=1";

void VerilerVeAdet()

{

sqlSinif.veriListele(komut, veriListesi, label\_adet);

sqlSinif.copkutuAdet(copkutuAdet, label\_copadet);

}

public string aranacakAd { get; set; }

private void uyeAra\_Load(object sender, EventArgs e)

{

int id;

comboBox\_sinif.SelectedIndex = 0;

sqlSinif.siniflar.Clear();

string kmt = "select \* from siniflar where silme\_durumu=0";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

SqlDataReader dr = komut.ExecuteReader();

while (dr.Read())

{

comboBox\_sinif.Items.Add(dr["ad"]);

id = Convert.ToInt32(dr["sinif\_id"]);

sqlSinif.siniflar.Add(id);

}

sqlSinif.Close();

if (aranacakAd == "")

{

VerilerVeAdet();

}

else

{

textBox\_ara.Text = aranacakAd;

arama();

}

}

public void arama()

{

string komut = "select uye\_id as İd,tcno as '" + "TC.No" + "',ad as Ad,soyad as Soyad,cinsiyet as Cinsiyet,telefon as Telefon ,eposta as '" + "E-posta" + "' from uyeler where silme\_durumu=0";

if (comboBox\_sinif.SelectedIndex == 0)

{

komut += "and ad Like '%" + textBox\_ara.Text + "%'";

}

else

{

komut = "select uyeler.uye\_id as İd,tcno as '" + "TC.No" + "',ad as Ad,soyad as Soyad,cinsiyet Cinsiyet ,telefon as Telefon,eposta as '" + "E-posta" + "' from uyeler inner join satislar on satislar.uye\_id=uyeler.uye\_id where satislar.sinif\_id= '" + sqlSinif.siniflar[comboBox\_sinif.SelectedIndex - 1] + "' and uyeler.ad like '%" + textBox\_ara.Text + "%' and durum=1 ";

}

sqlSinif.veriListele(komut, veriListesi, label\_adet);

sqlSinif.copkutuAdet(copkutuAdet, label\_copadet);

}

private void button\_copkutusu\_Click(object sender, EventArgs e)

{

copKutusu copKutusu = new copKutusu();

copKutusu.gelenForm = 4;

copKutusu.ShowDialog();

arama();

}

private void button\_ekle\_Click(object sender, EventArgs e)

{

uyeEkle uyeEkle = new uyeEkle();

uyeEkle.ShowDialog();

VerilerVeAdet();

}

private void button\_yenile\_Click(object sender, EventArgs e)

{

arama();

textBox\_ara.Text = "";

}

private void button\_sil\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);//DataGridView deki seçilenler

foreach (var item in sqlSinif.secilenlerListe)

{

int id = Convert.ToInt32(item);

string kmt = "update uyeler set silme\_durumu=1 where uye\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", id);

komut.ExecuteNonQuery();

sqlSinif.Close();

//tekrardan listeyi yenelemek yerine secilenler listeden siliniyor

}

sqlSinif.secilenleriListedenSil(veriListesi, label\_adet);

sqlSinif.copkutuAdet(copkutuAdet, label\_copadet);

}

private void veriListesi\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void uyeAra\_Resize(object sender, EventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void textBox\_ara\_TextChanged(object sender, EventArgs e)

{

if (textBox\_ara.Text != "")

{

label\_ara.Visible = false;

}

else

{

label\_ara.Visible = true;

}

arama();

}

private void comboBox\_sinif\_SelectedIndexChanged(object sender, EventArgs e)

{

arama();

}

private void veriListesi\_DoubleClick(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);

this.Close();

}

}

}

**Üyeler;**

using System;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class uyeler : Form

{

public uyeler()

{

InitializeComponent();

}

public void Acilacakform(Form frm)

{

panel\_sunum.Controls.Clear();

frm.TopLevel = false;

panel\_sunum.Controls.Add(frm);

frm.Show();

frm.Dock = DockStyle.Fill;

frm.BringToFront();

}

public string aranacakAd { get; set; }

public int kullaniciId { get; set; }

private void uyeler\_Load(object sender, EventArgs e)

{

uyeAra uyeAra = new uyeAra();

uyeAra.aranacakAd = aranacakAd;

Acilacakform(uyeAra);

}

private bool deger = true;

private void panel\_sunum\_ControlRemoved(object sender, ControlEventArgs e)

{

if (deger == true)

{

uyeDetaylari uyeDetaylari = new uyeDetaylari();

uyeDetaylari.kullaniciId = kullaniciId;

Acilacakform(uyeDetaylari);

deger = false;

}

else

{

uyeAra uyeAra = new uyeAra();

Acilacakform(uyeAra);

deger = true;

}

}

}

}

### Üye Detayları

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.IO;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class uyeDetaylari : Form

{

public uyeDetaylari()

{

InitializeComponent();

}

sqlSinif sqlSinif = new sqlSinif();

string kmt;

public string resim { get; set; }

public string gelenResim { get; set; }

private void uyeDetaylari\_Load(object sender, EventArgs e)

{

//Profil

int uyeId = sqlSinif.secilenlerListe[0];

string komut = "select \*from uyeler where uye\_id= ('" + uyeId + "') ";

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac()); ;

SqlDataReader dr = kmt.ExecuteReader();

if (dr.Read())

{

textBox\_id.Text = uyeId.ToString();

maskedTextBox\_tcno.Text = dr["tcno"].ToString();

textBox\_ad.Text = dr["ad"].ToString();

textBox\_soyad.Text = dr["soyad"].ToString();

maskedTextBox\_dogumta.Text = dr["dogumtarihi"].ToString();

textBox\_adres.Text = dr["adres"].ToString();

textBox\_il.Text = dr["il"].ToString();

textBox\_ilce.Text = dr["ilce"].ToString();

maskedTextBox1\_telefon.Text = dr["telefon"].ToString();

maskedTextBox\_telefon2.Text = dr["telefon2"].ToString();

textBox4\_eposta.Text = dr["eposta"].ToString();

int a = comboBox\_kangrub.FindStringExact(dr["kangrubu"].ToString());

comboBox\_kangrub.SelectedIndex = a;

if (dr["cinsiyet"].ToString() == "Erkek")

{

radioButton\_erkek.Checked = true;

}

else

{

radioButton\_kadın.Checked = true;

}

gelenResim = dr["resim"].ToString();

if (gelenResim != "")

{

pictureBox\_resim.ImageLocation = hedef + dr["resim"].ToString();

} }

sqlSinif.Close();

//Vücut Ölçüleri

sqlSinif.secilenlerListe.Clear();

int olcutur\_id;

string kmt1 = "select \* from olcu\_turleri where durum=1 and silme\_durumu=0";

SqlCommand komut1 = new SqlCommand(kmt1, sqlSinif.Baglanti\_ac());

dr = komut1.ExecuteReader();

while (dr.Read())

{

comboBox\_tur.Items.Add(dr["ad"].ToString());

comboBox\_turAra.Items.Add(dr["ad"].ToString());

olcutur\_id = Convert.ToInt32(dr["olcutur\_id"]);

sqlSinif.secilenlerListe.Add(olcutur\_id);

}

sqlSinif.Close();

verilerVucut();

comboBox\_turAra.SelectedIndex = 0;

//Satışlar

verilerSatis();

//Tahislatlar

verilerTahsilat();

}

void verilerVucut()

{

kmt = "select vucutolcu\_id as İd,olcu\_turleri.ad as '" + "Ölçü Ad" + "',tarih as Tarih,vucut\_olculeri.deger as Değer from vucut\_olculeri join olcu\_turleri on olcu\_turleri.olcutur\_id=vucut\_olculeri.olcutur\_id where vucut\_olculeri.uye\_id='" + textBox\_id.Text + "'";

sqlSinif.veriListele(kmt, veriListesi, label\_adet);

}

void verilerTahsilat()

{

if (checkBox\_tahislat.Checked)

{

kmt = "select tahsilatlar.tahsilat\_id as İd,tahsilatlar.tutar as Tutar,tahsilatlar.tarih as Tarih,tahsilatlar.saat as Saat, ad+' '+soyad as '" + "Ödemeyi Alan" + "', tahsilatlar.durum as Durum from tahsilatlar inner join kullanicilar on kullanicilar.kullanici\_id=tahsilatlar.kullanici\_id where tahsilatlar.durum=1 and tahsilatlar.uye\_id='" + textBox\_id.Text + "'";

}

else

{

kmt = "selec tahsilatlar.tahsilat\_id as İd,tahsilatlar.tutar as Tutar,tahsilatlar.tarih as Tarih,tahsilatlar.saat as Saat,ad+' '+soyad as'" + "Ödemeyi Alan" + "' ,tahsilatlar.durum as Durum from tahsilatlar inner join kullanicilar on kullanicilar.kullanici\_id=tahsilatlar.kullanici\_id where tahsilatlar.uye\_id='" + textBox\_id.Text + "' ";

}

sqlSinif.veriListeleRenkli(kmt, veriListesiTahsilat, label\_adetTahsilat, 5);

}

void verilerSatis()

{

if (checkBox\_iptal.Checked)

{

kmt = "select satislar.satis\_id '" + "Satış İd" + "',siniflar.ad as Sınıf,satislar.baslanic\_tarihi as '" + "Başlangıç Tarihi" + "',satislar.bitis\_tarihi '" + "Bitiş Tarihi" + "',satislar.satis\_fiyati as '" + "Satış Fiyatı" + "', satislar.durum as Durum from satislar join siniflar on satislar.sinif\_id=siniflar.sinif\_id where satislar.durum=1 and satislar.uye\_id='" + textBox\_id.Text + "'";

}

else

{

kmt = "select satislar.satis\_id '" + "Satış İd" + "',siniflar.ad as Sınıf,satislar.baslanic\_tarihi as '" + "Başlangıç Tarihi" + "',satislar.bitis\_tarihi '" + "Bitiş Tarihi" + "',satislar.satis\_fiyati as '" + "Satış Fiyatı" + "', satislar.durum as Durum from satislar join siniflar on satislar.sinif\_id=siniflar.sinif\_id where satislar.uye\_id='" + textBox\_id.Text + "' ";

}

sqlSinif.veriListeleRenkli(kmt, veriListesiSatis, label\_adetSatis, 5);

}

void AramaVucut()

{

if (comboBox\_turAra.SelectedIndex != -1)

{

if (comboBox\_turAra.SelectedIndex == 0)

{

verilerVucut();

}

else

{

int deger = sqlSinif.secilenlerListe[comboBox\_turAra.SelectedIndex - 1];

string kmt = "select vucutolcu\_id as İd,olcu\_turleri.ad as '" + "Ölçü Ad" + "',vucut\_olculeri.tarih as Tarih,vucut\_olculeri.deger as Değer from vucut\_olculeri join olcu\_turleri on olcu\_turleri.olcutur\_id=vucut\_olculeri.olcutur\_id where vucut\_olculeri.uye\_id='" + textBox\_id.Text + "' and vucut\_olculeri.olcutur\_id='" + deger + "' ";

sqlSinif.veriListele(kmt, veriListesi, label\_adet);

}

}

}

private void button\_duzelt\_Click(object sender, EventArgs e)

{

//Profil

string kmt = "update uyeler set tcno=@p1,ad=@p2,soyad=@p3,cinsiyet=@p4,dogumtarihi=@p5,adres=@p6,il=@p7,ilce=@p8,telefon=@p9,telefon2=@p10,eposta=@p11,kangrubu=@p12,resim=@p13 where uye\_id='" + textBox\_id.Text + "' ";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", maskedTextBox\_tcno.Text);

komut.Parameters.AddWithValue("@p2", textBox\_ad.Text);

komut.Parameters.AddWithValue("@p3", textBox\_soyad.Text);

if (radioButton\_erkek.Checked)

{

komut.Parameters.AddWithValue("@p4", "Erkek");

}

else

{

komut.Parameters.AddWithValue("@p4", "Kadın");

}

komut.Parameters.AddWithValue("@p5", maskedTextBox\_dogumta.Text);

komut.Parameters.AddWithValue("@p6", textBox\_adres.Text);

komut.Parameters.AddWithValue("@p7", textBox\_il.Text);

komut.Parameters.AddWithValue("@p8", textBox\_ilce.Text);

komut.Parameters.AddWithValue("@p9", maskedTextBox1\_telefon.Text);

komut.Parameters.AddWithValue("@p10", maskedTextBox\_telefon2.Text);

komut.Parameters.AddWithValue("@p11", textBox4\_eposta.Text);

komut.Parameters.AddWithValue("@p12", comboBox\_kangrub.Text);

komut.Parameters.AddWithValue("@p13", resim);

komut.ExecuteNonQuery();

sqlSinif.Close();

}

private void button\_sil\_Click(object sender, EventArgs e)

{

//Profil

DialogResult res = MessageBox.Show("Silerseniz üyeye ait bütün kayıtlar silinecektir.", "Confirmation", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

if (res == DialogResult.OK)

{

string kmt = "Delete from uyeler where uye\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", textBox\_id.Text);

komut.ExecuteNonQuery();

this.Close();

}

}

private void comboBox\_tur\_SelectedIndexChanged(object sender, EventArgs e)

{

int deger = sqlSinif.secilenlerListe[comboBox\_tur.SelectedIndex];

if (comboBox\_tur.SelectedIndex != -1)

{

string kmt = "Select birim,olcutur\_id from olcu\_turleri where olcutur\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", deger);

SqlDataReader dr = komut.ExecuteReader();

if (dr.Read())

{

textBox\_birim.Text = dr["birim"].ToString();

}

sqlSinif.Close();

}

}

private void button\_ekle\_Click(object sender, EventArgs e)

{

int deger = sqlSinif.secilenlerListe[comboBox\_tur.SelectedIndex];

if (comboBox\_tur.SelectedIndex != -1)

{

string kmt = "insert into vucut\_olculeri (tarih,deger,olcutur\_id,uye\_id) values (@p1,@p2,@p3,@p4)";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", DateTime.Now.ToString("yyyy.MM.dd"));

komut.Parameters.AddWithValue("@p2", textBox\_deger.Text);

komut.Parameters.AddWithValue("@p3", deger);

komut.Parameters.AddWithValue("@p4", textBox\_id.Text);

komut.ExecuteNonQuery();

sqlSinif.Close();

AramaVucut();

}

}

private void button\_yenile\_Click(object sender, EventArgs e)

{

AramaVucut();

}

private void veriListesi\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, tabPage\_vucut.Height - 220, label\_adet);

}

private void tabControl1\_SizeChanged(object sender, EventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, tabPage\_vucut.Height - 220, label\_adet);

sqlSinif.veriListesiBoyutu(veriListesiSatis, panel\_ust.Height - 170, label\_adetSatis);

sqlSinif.veriListesiBoyutu(veriListesiTahsilat, tabPage\_tahsilat.Height - 180, label\_adetTahsilat);

}

private void button\_Vsil\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow item in veriListesi.SelectedRows)

{

int id = Convert.ToInt32(item.Cells[0].Value);

string kmt = "delete from vucut\_olculeri where vucutolcu\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", id);

komut.ExecuteNonQuery();

sqlSinif.Close();

}

sqlSinif.secilenleriListedenSil(veriListesi, label\_adet);

}

private void comboBox\_turAra\_SelectedIndexChanged(object sender, EventArgs e)

{

AramaVucut();

}

private void tabPage3\_Click(object sender, EventArgs e)

{

//DateTime.Now.ToString("yyyy.MM.dd"));

}

private void button\_satis\_Click(object sender, EventArgs e)

{

satis satis = new satis();

satis.uyeId = Convert.ToInt32(textBox\_id.Text);

satis.ShowDialog();

verilerSatis();

}

private void checkBox\_iptal\_CheckedChanged(object sender, EventArgs e)

{

verilerSatis();

}

private void button\_yenileSatis\_Click(object sender, EventArgs e)

{

verilerSatis();

}

private void button\_silSatis\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesiSatis);//DataGridView deki seçilenler

foreach (int item in sqlSinif.secilenlerListe)

{

int id = Convert.ToInt32(item);

string kmt = "update satislar set durum=0 where satis\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", item);

komut.ExecuteNonQuery();

sqlSinif.Close();

//tekrardan listeyi yenelemek yerine secilenler listeden siliniyor

}

verilerSatis();

}

private void veriListesiSatis\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesiSatis, panel\_ust.Height - 170, label\_adetSatis);

int toplamSatis = 0;

if (veriListesiSatis.RowCount != 0)

{

for (int i = 0; i < veriListesiSatis.RowCount; i++)

{

if (Convert.ToBoolean(veriListesiSatis.Rows[i].Cells[5].Value) == true)

{

toplamSatis += Convert.ToInt32(veriListesiSatis.Rows[i].Cells[4].Value);

}

}

label\_satisTutar.Text = toplamSatis.ToString();

}

else

{

label\_satisTutar.Text = 0.ToString();

}

label\_bakiye.Text = (Convert.ToInt32(label\_tahsilatTutar.Text) - toplamSatis).ToString();

}

public int kullaniciId { get; set; }

private void button\_odemeAl\_Click(object sender, EventArgs e)

{

anaSayfa anaSayfa = new anaSayfa();

if (textBox\_tutar.Text != "")

{

string kmt = "insert into tahsilatlar(tarih,saat,tutar,uye\_id,kullanici\_id) values (@p1,@p2,@p3,@p4,@p5)";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", DateTime.Now.ToString("yyyy.MM.dd"));

komut.Parameters.AddWithValue("@p2", DateTime.Now.ToLongTimeString());

komut.Parameters.AddWithValue("@p3", textBox\_tutar.Text);

komut.Parameters.AddWithValue("@p4", textBox\_id.Text);

komut.Parameters.AddWithValue("@p5", kullaniciId);

komut.ExecuteNonQuery();

sqlSinif.Close();

verilerTahsilat();

}

}

private void button3\_Click(object sender, EventArgs e)

{

verilerTahsilat();

}

private void veriListesiTahsilat\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesiTahsilat, tabPage\_tahsilat.Height - 180, label\_adetTahsilat);

int toplamTahsilat = 0;

if (veriListesiTahsilat.RowCount != 0)

{

for (int i = 0; i < veriListesiTahsilat.RowCount; i++)

{

if (Convert.ToBoolean(veriListesiTahsilat.Rows[i].Cells[5].Value) == true)

{

toplamTahsilat += Convert.ToInt32(veriListesiTahsilat.Rows[i].Cells[1].Value);

}

}

label\_tahsilatTutar.Text = toplamTahsilat.ToString();

}

else

{

label\_tahsilatTutar.Text = 0.ToString();

}

label\_bakiye.Text = (toplamTahsilat - Convert.ToInt32(label\_satisTutar.Text)).ToString();

}

private void button\_silTahsilat\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesiTahsilat);//DataGridView deki seçilenler

foreach (int item in sqlSinif.secilenlerListe)

{

int id = Convert.ToInt32(item);

string kmt = "update tahsilatlar set durum=0 where tahsilat\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", item);

komut.ExecuteNonQuery();

sqlSinif.Close();

//tekrardan listeyi yenelemek yerine secilenler listeden silinmek istensede pasif olanları tekrardan listeden sildiğin için liste yenilenmek zorunda kalınıyor

}

verilerTahsilat();

}

private void checkBox\_tahislat\_CheckedChanged(object sender, EventArgs e)

{

verilerTahsilat();

}

string hedef = @"C:\Users\zynls\source\repos\sporsalonu\sporsalonu\uyeResimleri\";

private void pictureBox1\_Click(object sender, EventArgs e)

{

string kayit = "";

openFileDialog\_resim.Title = "Resim Dosyası Seçiniz";

openFileDialog\_resim.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);

openFileDialog\_resim.Filter = "Resim Dosyası (\*.jpg, \*.jpeg, \*.png) | \*.jpg; \*.jpeg; \*.png";

if (openFileDialog\_resim.ShowDialog() == DialogResult.OK)

{

pictureBox\_resim.ImageLocation = openFileDialog\_resim.FileName;

string resim1 = openFileDialog\_resim.FileName;

if (openFileDialog\_resim.SafeFileName.EndsWith(".jpg"))

{

kayit = "uye" + textBox\_id.Text + ".jpg";

}

else if (openFileDialog\_resim.SafeFileName.EndsWith(".png"))

{

kayit = "uye" + textBox\_id.Text + ".png";

}

else if (openFileDialog\_resim.SafeFileName.EndsWith(".jpeg"))

{

kayit = "uye" + textBox\_id.Text + ".jpeg";

}

if (File.Exists(hedef + kayit))

{

File.Delete(hedef + kayit);

}

File.Copy(resim1, hedef + kayit);

resim = kayit;

}

}

}

}

**Satış Yap;**

using sporsalonu.siniflar;

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class satis : Form

{

public satis()

{

InitializeComponent();

}

public int uyeId { get; set; }

sqlSinif sqlSinif = new sqlSinif();

private void button\_satisyap\_Click(object sender, EventArgs e)

{

if (durumBos == true)

{

DateTime baslangic = dateTimePicker\_baslangic.Value;

DateTime bitis = Convert.ToDateTime(textBox\_bitistar.Text);

string kmt = "insert into satislar(baslanic\_tarihi,bitis\_tarihi,satis\_fiyati,uye\_id,sinif\_id) values (@p1,@p2,@p3,@p4,@p5)";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", baslangic.ToString("yyyy.MM.dd"));

komut.Parameters.AddWithValue("@p2", bitis.ToString("yyyy.MM.dd"));

komut.Parameters.AddWithValue("@p3", textBox\_fiyat.Text);

komut.Parameters.AddWithValue("@p4", uyeId);

komut.Parameters.AddWithValue("@p5", comboBox\_sinif.SelectedValue);

komut.ExecuteNonQuery();

MessageBox.Show("Satış işlemi başarıyla gerçekleştirildi.");

int secili = comboBox\_sinif.SelectedIndex;

comboBox\_sinif.SelectedIndex++; ;

comboBox\_sinif.SelectedIndex = secili;

}

else

{

MessageBox.Show("Seçtiğiniz sınıf doludur.");

}

}

private void satis\_Load(object sender, EventArgs e)

{

string kmt = "select \* from siniflar";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

SqlDataAdapter da = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

da.Fill(dt);

comboBox\_sinif.DataSource = dt;

comboBox\_sinif.ValueMember = "sinif\_id";

comboBox\_sinif.DisplayMember = "ad";

comboBox\_sinif.Text = "Sınıf Seçiniz";

}

int fi;

private void comboBox\_sure\_SelectedIndexChanged(object sender, EventArgs e)

{

string kmt = "select fiyat siniflar where sinif\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

//komut.Parameters.AddWithValue("@p1",)

DateTime baslangic = DateTime.Parse(dateTimePicker\_baslangic.Text);

if (comboBox\_sure.SelectedIndex < 12)

{

baslangic = baslangic.AddDays((comboBox\_sure.SelectedIndex + 1) \* 30);

textBox\_fiyat.Text = (fi \* (comboBox\_sure.SelectedIndex + 1)).ToString();

}

else

{

baslangic = baslangic.AddDays((comboBox\_sure.SelectedIndex - 10) \* 30 \* 12);

textBox\_fiyat.Text = (fi \* ((comboBox\_sure.SelectedIndex - 10) \* 12)).ToString();

}

textBox\_bitistar.Text = baslangic.ToShortDateString();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

int sayac = 0;

bool durumBos = true;

private void comboBox\_sinif\_SelectedIndexChanged(object sender, EventArgs e)

{

if (sayac > 2)

{

//Seçilen sınıfın kapasitesi

string kmt1 = "select kapasite from siniflar where sinif\_id=@p1";

SqlCommand komut1 = new SqlCommand(kmt1, sqlSinif.Baglanti\_ac());

komut1.Parameters.AddWithValue("@p1", comboBox\_sinif.SelectedValue);

string kapasite = komut1.ExecuteScalar().ToString();

//Seçilen sınıfın toplam üye sayısı

string kmt2 = "select count(\*) from satislar where sinif\_id=@p1 and durum=1";

SqlCommand komut2 = new SqlCommand(kmt2, sqlSinif.Baglanti\_ac());

komut2.Parameters.AddWithValue("@p1", comboBox\_sinif.SelectedValue);

string toplamUye = komut2.ExecuteScalar().ToString();

if (Convert.ToInt32(toplamUye) <= Convert.ToInt32(kapasite))

{

string kmt = "select fiyat from siniflar where sinif\_id=@p1";

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", comboBox\_sinif.SelectedValue);

string deger = komut.ExecuteScalar().ToString();

fi = Convert.ToInt32(deger);

textBox\_fiyat.Text = fi.ToString();

durumBos = true;

}

else

{

MessageBox.Show("Sınıf kapasitesi doludur.");

durumBos = false;

}

}

else

{

sayac++;

}

}

private void textBox\_fiyat\_Leave(object sender, EventArgs e)

{

if (textBox\_fiyat.Text == "" || textBox\_fiyat.Text == null)

{

}

else

{

fi = Convert.ToInt32(textBox\_fiyat.Text);

}

}

private void textBox\_fiyat\_Enter(object sender, EventArgs e)

{

comboBox\_sure.SelectedIndex = 0;

}

}

}

### Çöp Kutusu

using sporsalonu.siniflar;

using System;

using System.Data.SqlClient;

using System.Runtime.InteropServices;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class copKutusu : Form

{

public copKutusu()

{

InitializeComponent();

this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;

}

sqlSinif sqlSinif = new sqlSinif();

string kmt;

string komut;

public byte gelenForm { get; set; }

[DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]

private extern static void ReleaseCapture();

[DllImport("user32.DLL", EntryPoint = "SendMessage")]

private extern static void SendMessage(System.IntPtr hWnd, int wMsg, int wParam, int lParam);

void veriListele()

{

switch (gelenForm)

{

case 1: komut = "select olcutur\_id as İd,ad as Ad ,birim as Birim ,durum as '" + "Aktif/Pasif" + "' from olcu\_turleri where silme\_durumu=1"; break;

case 2: komut = "select kullanici\_id as İd, kullanici\_ad as '" + "Kullanıcı Adı" + "', sifre as Şifre, ad as Ad,soyad as Soyad from kullanicilar where silme\_durumu = 1"; break;

case 3: komut = "select sinif\_id as İd, ad as '" + "Ad" + "', kapasite as Kapasite,fiyat as Fiyat, durum '" + "Aktif/Pasif" + "' from siniflar where silme\_durumu=1"; break;

case 4: komut = "select uye\_id as İd,tcno as '" + "T.C No" + "',ad as Ad,soyad as Soyad,cinsiyet as Cinsiyet from uyeler where silme\_durumu=1"; break;

}

sqlSinif.veriListele(komut, veriListesi, label\_adet);

}

private void cop\_kutusu\_Load(object sender, EventArgs e)

{

veriListele();

}

private bool formboyut = true;

private void button3\_Click(object sender, EventArgs e)

{

if (formboyut == true)

{

this.WindowState = FormWindowState.Maximized;

formboyut = false;

}

else

{

this.WindowState = FormWindowState.Normal;

formboyut = true;

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.WindowState = FormWindowState.Minimized;

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void panel\_üst\_MouseDown(object sender, MouseEventArgs e)

{

ReleaseCapture();

SendMessage(this.Handle, 0x112, 0xf012, 0);

formboyut = true;

}

private void button\_geriyukle\_Click(object sender, EventArgs e)

{

sqlSinif.secilenler(veriListesi);

foreach (var item in sqlSinif.secilenlerListe)

{

int id = Convert.ToInt32(item);

switch (gelenForm)

{

case 1: kmt = "update olcu\_turleri set silme\_durumu=0 where olcutur\_id=@p1"; break;

case 2: kmt = "update kullanicilar set silme\_durumu=0 where kullanici\_id=@p1"; break;

case 3: kmt = "update siniflar set silme\_durumu=0 where sinif\_id=@p1"; break;

case 4: kmt = "update uyeler set silme\_durumu=0 where uye\_id=@p1"; break;

}

SqlCommand komut = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

komut.Parameters.AddWithValue("@p1", id);

komut.ExecuteNonQuery();

}

sqlSinif.secilenleriListedenSil(veriListesi, label\_adet);

sqlSinif.Close();

}

private void button\_yenile\_Click(object sender, EventArgs e)

{

veriListele();

}

private void veriListesi\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void button\_sil\_Click(object sender, EventArgs e)

{

silinecekler silinecekler = new silinecekler();

sqlSinif.secilenler(veriListesi);

switch (gelenForm)

{

case 1: silinecekler.gelenForm = 1; break;

case 2: silinecekler.gelenForm = 2; break;

case 3: silinecekler.gelenForm = 3; break;

case 4: silinecekler.gelenForm = 4; break;

}

silinecekler.ShowDialog();

veriListele();

}

private void cop\_kutusu\_SizeChanged(object sender, EventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

}

private void button\_hepsiniSil\_Click(object sender, EventArgs e)

{

sqlSinif.secilenlerListe.Clear();

int id;

switch (gelenForm)

{

case 1: komut = "select olcutur\_id from olcu\_turleri where silme\_durumu=1"; break;

case 2: komut = "select kullanici\_id from kullanicilar where silme\_durumu = 1"; break;

case 3: komut = "select sinif\_id from siniflar where silme\_durumu=1"; break;

case 4: komut = "select uye\_id from uyeler where silme\_durumu=1"; break;

}

SqlCommand kmt = new SqlCommand(komut, sqlSinif.Baglanti\_ac());

SqlDataReader dr = kmt.ExecuteReader();

while (dr.Read())

{

id = Convert.ToInt32(dr[0]);

sqlSinif.secilenlerListe.Add(id);

}

sqlSinif.Close();

silinecekler silinecekler = new silinecekler();

switch (gelenForm)

{

case 1: silinecekler.gelenForm = 1; break;

case 2: silinecekler.gelenForm = 2; break;

case 3: silinecekler.gelenForm = 3; break;

case 4: silinecekler.gelenForm = 4; break;

}

silinecekler.ShowDialog();

veriListele();

}

private void button\_hepsiniGeri\_Click(object sender, EventArgs e)

{

sqlSinif.secilenlerListe.Clear();

int id;

switch (gelenForm)

{

case 1: komut = "select olcutur\_id from olcu\_turleri where silme\_durumu=1"; break;

case 2: komut = "select kullanici\_id from kullanicilar where silme\_durumu = 1"; break;

case 3: komut = "select sinif\_id from siniflar where silme\_durumu=1"; break;

case 4: komut = "select uye\_id from uyeler where silme\_durumu=1"; break;

}

SqlCommand kmt1 = new SqlCommand(komut, sqlSinif.Baglanti\_ac());

SqlDataReader dr = kmt1.ExecuteReader();

while (dr.Read())

{

id = Convert.ToInt32(dr[0]);

sqlSinif.secilenlerListe.Add(id);

}

sqlSinif.Close();

switch (gelenForm)

{

case 1: kmt = "update olcu\_turleri set silme\_durumu=0 where olcutur\_id=@p1"; break;

case 2: kmt = "update kullanicilar set silme\_durumu=0 where kullanici\_id=@p1"; break;

case 3: kmt = "update siniflar set silme\_durumu=0 where sinif\_id=@p1"; break;

case 4: kmt = "update uyeler set silme\_durumu=0 where uye\_id=@p1"; break;

}

foreach (int item in sqlSinif.secilenlerListe)

{

SqlCommand komut2 = new SqlCommand(kmt, sqlSinif.Baglanti\_ac());

int id2 = Convert.ToInt32(item);

komut2.Parameters.AddWithValue("@p1", id2);

komut2.ExecuteNonQuery();

}

veriListele();

}

private void textBox\_ara\_TextChanged(object sender, EventArgs e)

{

if (textBox\_ara.Text != "")

{

label\_ara.Visible = false;

}

else

{

label\_ara.Visible = true;

}

}

}

}

### Silinecekler

using sporsalonu.siniflar;

using System;

using System.Data;

using System.Data.SqlClient;

using System.Runtime.InteropServices;

using System.Windows.Forms;

namespace sporsalonu.formlar

{

public partial class silinecekler : Form

{

public silinecekler()

{

InitializeComponent();

this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;

}

sqlSinif sqlSinif = new sqlSinif();

[DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]

private extern static void ReleaseCapture();

[DllImport("user32.DLL", EntryPoint = "SendMessage")]

private extern static void SendMessage(System.IntPtr hWnd, int wMsg, int wParam, int lParam);

public byte gelenForm { get; set; }

private void silinecekler\_Load(object sender, EventArgs e)

{

string komut = "";

DataTable dt = new DataTable();

foreach (var item in sqlSinif.secilenlerListe)

{

switch (gelenForm)

{

case 1: komut = "select olcutur\_id as İd,ad as Ad ,birim as Birim ,durum as '" + "Aktif/Pasif" + "' from olcu\_turleri where silme\_durumu=1 and olcutur\_id = ('" + item + "') "; break;

case 2: komut = "select kullanici\_id as İd, kullanici\_ad as '" + "Kullanıcı Adı" + "', sifre as Şifre, ad as Ad,soyad as Soyad from kullanicilar where silme\_durumu=1 and kullanici\_id= ('" + item + "') "; break;

case 3: komut = "select sinif\_id as İd, ad as '" + "Ad" + "', kapasite as Kapasite,fiyat as Fiyat, durum '" + "Aktif/Pasif" + "' from siniflar where silme\_durumu=1 and sinif\_id=('" + item + "')"; break;

case 4: komut = "select uye\_id as İd,tcno as '" + "T.C No" + "',ad as Ad,soyad as Soyad,cinsiyet as Cinsiyet from uyeler where silme\_durumu=1 and uye\_id=('" + item + "')"; break;

}

SqlDataAdapter da = new SqlDataAdapter(komut, sqlSinif.Baglanti\_ac());

da.Fill(dt);

}

veriListesi.DataSource = dt;

sqlSinif.Close();

sqlSinif.veriAdeti(veriListesi, label\_adet);

sqlSinif.veriAdeti(veriListesi, label\_adet);

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button\_tamam\_Click(object sender, EventArgs e)

{

string komut = "";

switch (gelenForm)

{

case 1: komut = "delete from olcu\_turleri where olcutur\_id=@p1"; break;

case 2: komut = "delete from kullanicilar where kullanici\_id=@p1"; break;

case 3: komut = "delete from siniflar where sinif\_id=@p1"; break;

case 4: komut = "delete from uyeler where uye\_id=@p1"; break;

}

try

{

sqlSinif.veriSilTumu(komut, veriListesi);

label\_adet.Text = "Hiç kayıt bulunamadı.";

}

catch (Exception)

{

MessageBox.Show("Kayda ait veriler bulunmakta.");

}

}

private bool formboyut = true;

private void button\_buyut\_Click(object sender, EventArgs e)

{

if (formboyut == true)

{

this.WindowState = FormWindowState.Maximized;

formboyut = false;

}

else

{

this.WindowState = FormWindowState.Normal;

formboyut = true;

}

}

private void button\_kucult\_Click(object sender, EventArgs e)

{

this.WindowState = FormWindowState.Minimized;

}

private void button\_geriyukle\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow item in veriListesi.SelectedRows)//gelen verileri Listeden siler

{

if (veriListesi.Rows.Count != 0)

{

int deger = (Convert.ToInt32(item.Cells[0].Value));

sqlSinif.secilenlerListe.Remove(deger);

}

}

sqlSinif.secilenleriListedenSil(veriListesi, label\_adet);

}

private void veriListesi\_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

button\_tamam.Top = label\_adet.Top - 8;

}

private void silinecekler\_SizeChanged(object sender, EventArgs e)

{

sqlSinif.veriListesiBoyutu(veriListesi, panel\_sunum.Height - 50, label\_adet);

button\_tamam.Top = label\_adet.Top - 8;

}

}

}

# KAYNAKÇA

<http://www.acıbadem.com>

<https://spormerkezi.usak.edu.tr/>

<http://www.sporsalonum.net>

<http://www.perkotek.com>

<http://www.gymsoft.com>

<http://www.gymturk.com>

<https://www.interbu.com/tr/spor-salonu-otomasyonu>

<http://spormerkezi.pau.edu.tr/>