

**AFYON KOCATEPE ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**



**NESNE TABANLI PROGRAMLAMA / DÖNEM SONU PROJE RAPORU**

**Afyon Fitness Center**

**Gürkan ŞAHİN**

**212923058**

**ÖĞRETİM ÜYESİ**

**NAİM KARASEKRETER**

**12.04.2023**

## **ÖZET**

### **NESNE TABANLI PROGRAMLAMA / DÖNEM SONU PROJE RAPORU**

**Afyon Fitness Center**

**Gürkan ŞAHİN**

**212923058**

**Afyon Kocatepe Üniversitesi**

**Mühendislik Fakültesi**

**Öğretim Üyesi: Naim Karasekreter**

**Spor Salonu Yönetim Otomasyonu projesinin temel hedefi, spor salonlarının daha düzenli ve sistematik bir çalışma düzenine kavuşmalarını sağlamaktır. Bu otomasyon sistemi, özellikle üye takibi ve ödeme takibi gibi kritik alanlarda daha etkili bir yönetim imkanı sunmayı amaçlamaktadır.**

**Aşağıda projenin temel özelliklerine kısaca yer verilmiştir:**

**\*Üye Kayıt ve Yönetim:**

**\*Yeni Üye kaydı ekleyebilme.**

**\*Üye güncelleme.**

**\*Üye ödeme takibi.**

**\*Duyuru paneli .**

**\*Bireysel olarak spor programı oluşturma**

### **Proje Sunumu Detayları:**

**1.Projenin Amaçları ve Hedefleri: Spor Salonu Otomasyonu projesinin genel amacı daha planlı ve sistematik bir çalışma düzeni kazandırması**

2.Projenin geliştirilmesinde hangi programlama dili ve platformlar kullanıldı? (C#, .NET, Visual Studio gibi) Veri yönetimi için hangi veritabanı teknolojisi tercih edildi? (VS veritabanı kullanıldı )

3.Hangi veritabanı mimarisi ve yapıları kullanıldı? Kullanılan veri modelleri nelerdir? (Üyeler, ödemeler, duyurular)

4.Projenin kod yapısı nasıl düzenlendi? (modülerlik, sınıf yapıları, vb.) Hangi sınıflar ve nesneler kullanıldı? (Üye sınıfı, Ödeme sınıfı, vs.) Bu sınıflar arasındaki ilişkiler nasıl kuruldu? (miras alma, arayüzler, vs.)

## İÇİNDEKİLER

	Sayfa
1)ÖZET .....	2
2)PROJE MİMARİSİ VE SINIF YAPISI.....	4
3)VERİTABANI YAPISI .....	6
4)PROJE DETAYLARI VE TEKNİK BİLGİLER .....	7
5)KULLANICI ARAYÜZ ÖRNEKLERİ.....	9
6)FORMLAR VE İŞLEVSELLİKLER .....	11
7)OOP KULLANIMI .....	13
8)TARTIŞMA VE SONUÇ .....	14
9) TEŞEKKÜRLER .....	15

## Proje Mimarisi ve Sınıf Yapısı

### Proje Genel Bakışı

Spor Salonu Yönetim Otomasyonu projesi, C# programlama dili ve Visual Studio'nun .NET Framework Form Application kullanılarak geliştirilmiştir. Proje, spor salonlarının yönetimi, üye kayıtları, ödeme takibi sistemi ve genel üye bilgilerinin tutulması gibi temel özelliklere odaklanmaktadır.

### Sınıf Yapısı

Proje, aşağıda belirtilen ana sınıflar etrafında şekillenmiştir:

#### 1. Giriş Sınıfı

Amaç:

Giriş sınıfı, kullanıcı adı ve şifre gibi kimlik bilgileriyle sisteme giriş yapma işlevselliğini sağlamak amacıyla tasarlanmıştır.

"Giris" sınıfı, sisteme giriş yapan kullanıcının doğruluğunu kontrol eder ve gerekli aksiyonları gerçekleştirir.

Metotlar:

**Login**(string username, string password, string baglanti): Kullanıcının giriş yapmasını sağlayan metodur. Kullanıcı adı ve şifre kontrolü yapar, doğrulandığında giriş işlemine izin verir. Ayrıca, bağlantı parametresi aracılığıyla veritabanı işlemlerini başlatır.

**PerformDatabaseAction**(string baglanti): Veritabanı işlemlerini gerçekleştirmek üzere kullanılan bir metodur. Bu metodun içeriği, veritabanı bağlantısı üzerinden gerekli işlemleri yapacak şekilde uyarlanabilir.

Bu sınıf aynı zamanda bir başka sınıf olan "trigger" sınıfından türetilmiştir.

#### 2. Kayıt Sınıfı

Amaç:

"kaydol" sınıfı, spor salonu üyelerinin kayıt işlemlerini gerçekleştirmek ve bu kayıtları veritabanına eklemek amacıyla tasarlanmıştır.

Yeni üye kaydı yapılırken, gerekli bilgiler kontrol edilir ve eksik bilgi durumunda kullanıcıya bilgi verilir.

Metotlar:

**HesaplaGunSayisi**(DateTimePicker baslangicDateTimePicker, DateTimePicker bitisDateTimePicker): İki tarih arasındaki gün sayısını hesaplayan bir metottur. Bu metot, üye kaydının başlama ve bitiş tarihleri arasındaki gün sayısını hesaplamak için kullanılabilir.

**KayitOl**(string adSoyad, string telefon, string cinsiyet, string yas, string zamanlama, string odeme, DateTime baslangic, DateTime bitis): Yeni üye kaydını oluşturan ve veritabanına ekleyen ana metottur. Bilgiler kontrol edilir, eksik bilgi varsa kullanıcıya uyarı verilir. Veritabanına eklenen kayıt başarıyla tamamlandığında kullanıcıya bilgi verilir.

### 3. SporProgrami Sınıfı

Amaç:

"SporProgrami" sınıfı, spor salonu üyeleri için kişisel bir antrenman programı oluşturmayı hedefler.

Kullanıcının cinsiyet, çalışılacak bölge, gelişim süresi ve kilo aralığına göre özelleştirilmiş bir spor programı sağlar.

Metotlar:

**ProgramOlustur()**: Kullanıcının belirlediği parametrelere göre genel bir spor programı oluşturur. Cinsiyet, çalışılacak bölge, gelişim süresi ve kilo aralığı gibi bilgileri içeren genel bir program listesi döndürür.

**SporProgramiOlustur()**: Cinsiyet bilgisine göre spesifik bir spor programını oluşturan metottur. Bu metod, "Bay" veya "Bayan" olarak belirlenen cinsiyetlere göre farklı spor programlarını çağırır.

**SporProgramiBay() ve SporProgramiBayan()**: Belirli bir cinsiyete göre çalışılacak bölgeye özel spor programlarını oluşturan metotlardır. Hangi bölgelerde çalışma yapılacağına göre farklı egzersizler içerir.

**BeslenmeMenuOlusturBay() ve BeslenmeMenuOlusturKadin()**: Belirli bir cinsiyete göre kilo aralığına özel beslenme menüsü oluşturan metotlardır. Kullanıcıya önerilen beslenme düzenini içerir.

### 4. Yazdir Sınıfı

Amaç: Çeşitli raporları ve formları yazdırmak için kullanılır.

Metotlar:

**YazdirSayfa**: Belirli bir DataGridView'ı ve başlığı kullanarak sayfa yazdırma işlemini yönetir.

## 5. TarihSeçimi Sınıfı

Amaç:

"TarihSeçimi" sınıfı, kullanıcıya sunulan iki tarih arasındaki gün farkını hesaplamak amacıyla oluşturulmuştur.

Genellikle bir başlangıç tarihi ve bitiş tarihi arasındaki gün sayısını hesaplamak için kullanılır.

Metotlar:

**HesaplaGunFarki(DateTimePicker baslangicDatePicker, DateTimePicker bitisDatePicker):** Bu metod, iki tarih arasındaki gün farkını hesaplamak için kullanılır. Başlangıç ve bitiş tarihleri, DateTimePicker nesneleri aracılığıyla alınır. Ardından, bu tarihler arasındaki gün farkı hesaplanır ve bir tam sayı olarak geri döndürülür.

## 6. textboxAyarlar Sınıfı

Amaç:

"textboxAyarlar" sınıfı, bir Windows Forms uygulamasında bulunan TextBox ve ComboBox gibi kontrol elemanlarını temizlemek amacıyla oluşturulmuştur.

Kullanıcıdan alınan giriş verilerini temizlemek ve başlangıç durumuna getirmek için kullanılır.

Metotlar:

**TemizleTumControls(Control.ControlCollection controls):** Bu metod, bir Control koleksiyonunu alır ve içindeki tüm kontrol elemanlarını temizler. Kontrol koleksiyonu içinde yer alan TextBox'ların metin içeriklerini temizler, ComboBox'ların seçilen indeksini sıfırlar. Ayrıca, eğer bir kontrol elemanının altında başka kontrol elemanları varsa, bu metod recursive olarak çağrılarak altındaki tüm kontrol elemanları da temizlenir.

## 7. trigger Sınıfı

Amaç:

"trigger" sınıfı, bir veritabanı bağlantısı üzerinde belirli bir eylemi gerçekleştirmek amacıyla oluşturulmuştur.

Genellikle bu sınıf, bağlantı açma, belirli bir SQL sorgusu gerçekleştirme gibi işlemleri içerir.

Metotlar:

**PerformDatabaseAction(string connectionString):** Bu metod, bir veritabanı bağlantı dizesini parametre olarak alır. Bu bağlantı dizesi ile bir SqlConnection nesnesi oluşturularak veritabanına bağlanır. Bağlantı başarılı olduğunda kullanıcıya bir bilgi mesajı gösterilir. Daha sonra, belirli bir SQL sorgusu (örneğin,

üyelik bitiş tarihi geçmiş olan üyeleri silme) kullanılarak veritabanında bir işlem gerçekleştirilir. Bağlantı açılırken veya sorgu çalıştırılırken bir hata oluşursa kullanıcıya bir hata mesajı gösterilir.

## 8. UyeGuncelleSil Sınıfı

Amaç:

Bu sınıf, üye bilgilerini güncelleme ve silme işlemlerini gerçekleştirmek amacıyla tasarlanmıştır.

Genellikle kullanıcıdan alınan girişleri kullanarak veritabanındaki üye bilgilerini güncelleme veya silme işlemlerini gerçekleştirir.

Metotlar:

**Sil**(int key, DataGridView dataGridView, string tableName): Bu metot, belirli bir üyenin silinmesini sağlar. key parametresi, silinecek üyenin benzersiz kimliğini belirtir. Eğer key değeri 0 ise, silinecek üye seçilmemiş demektir ve kullanıcıya bir uyarı mesajı gösterilir. Aksi takdirde, veritabanından ilgili üye silinir ve kullanıcıya işlemin başarıyla gerçekleştirildiği bir bilgi mesajı gösterilir. Son olarak, UyelerMetodu çağrılarak verilerin güncel halini yükler.

**Guncelle**(int key, string adSoyad, string telefon, string cinsiyet, string yas, string zamanlama, string odeme, DataGridView dataGridView, DateTime baslangic, DateTime bitis): Bu metot, belirli bir üyenin bilgilerini günceller. Parametre olarak alınan değerlere göre ilgili üye veritabanında güncellenir. Eğer gerekli bilgiler eksikse veya üye seçilmemişse, kullanıcıya bir uyarı mesajı gösterilir. Güncelleme işlemi başarıyla tamamlandığında kullanıcıya bir bilgi mesajı gösterilir ve UyelerMetodu çağrılarak verilerin güncel halini yükler.

## 9. uye işlemleri Sınıfı

Amaç:

Bu sınıf, üye işlemleri (üyeleri görüntüleme, filtreleme, ödeme yapma, güncelleme ve silme) ve genel veri işleme görevlerini yerine getirmek üzere tasarlanmıştır.

Veritabanı bağlantısı üzerinden gerekli sorguları çalıştırarak üye işlemleri ve diğer veri işlemlerini gerçekleştirir.

Metotlar:

**UyelerMetodu**(SqlConnection baglanti, DataGridView dataGridView, string tabloAdi): Bu metot, belirli bir tablodaki tüm üyeleri DataGridView'e yükler. Bağlantı açılır, sorgu çalıştırılır ve veriler DataGridView'e atanır.

**FiltrelemeMetodu**(SqlConnection baglanti, DataGridView dataGridView, string tabloAdi, string arananMetin): Bu metot, belirli bir tablodaki verileri belirli bir sütuna göre filtreler. Filtreleme işlemi için LIKE sorgusu kullanılmıştır.

FillName(SqlConnection baglanti, ComboBox AdSoyadCb): Bu metot, belirli bir tablodaki üye adlarını bir ComboBox'a doldurur. Veritabanından üye adlarını alır ve ComboBox'a atar.

MakePayment(SqlConnection baglanti, ComboBox AdSoyadCb, string OdemeTb, DateTimePicker PeriyotDt): Bu metot, belirli bir üyenin ödeme yapmasını sağlar. Ödeme bilgileri güncellenir ve ilgili ödeme tablosuna yeni bir kayıt eklenir.

DuyuruEkle(SqlConnection baglanti, DateTimePicker tarihPicker, TextBox konuTextBox, RichTextBox aciklamaRichTextBox): Bu metot, belirli bir tabloya (Duyurular tablosu) yeni bir duyuru ekler. Kullanıcı tarafından girilen tarih, konu ve açıklama bilgileri ile yeni bir kayıt oluşturur.

## Veritabanı Yapısı

Projede kullanılan veritabanı, Fitness Center otomasyonu için üç temel tabloyu içermektedir: UyeTbl, odemeTbl ve duyurular.

### 1. UyeTbl Tablosu

Alanlar:

**Uid** (Üye ID): Üyenin benzersiz bir kimlik numarasını temsil eder. Otomatik artan bir sayıdır (IDENTITY).

**UAdsoyad** (Üye Adı Soyadı): Üyenin adı ve soyadını saklar.

**Utelefon** (Üye Telefon): Üyenin telefon numarasını saklar.

**Ucinsiyet** (Üye Cinsiyeti): Üyenin cinsiyetini temsil eder.

**UYas** (Üye Yaşı): Üyenin yaşı.

**UOdeme** (Üye Ödeme): Üyenin ödeme bilgilerini saklar.

**UZamanlama** (Üye Zamanlama): Üyenin katılım zamanlamasını belirtir.

**UBaslangic** (Üye Başlangıç Tarihi): Üyenin başlangıç tarihini saklar.

**UBitis** (Üye Bitiş Tarihi): Üyenin bitiş tarihini saklar.

**UAciklama** (Üye Açıklama): Üye ile ilgili açıklamaları içerir.

### 2. OddemeTbl Tablosu

Alanlar:

**Oid** (Odeme ID): Bu alan, benzersiz bir kimlik numarasını temsil eder ve otomatik artan bir sayıdır (IDENTITY). Her yeni ödeme kaydı eklenirken, bu alan otomatik olarak bir artar.



**OAy** (Ödeme Ayl): Bu alan, ödemenin hangi aya ait olduğunu belirtir. VARCHAR (50) veri türü, en fazla 50 karakter uzunluğunda metin bilgisi saklayabilir.

**OUye** (Üye Adı): Bu alan, ödemeyi yapan üyenin adını temsil eder. VARCHAR (50) veri türü, en fazla 50 karakter uzunluğunda metin bilgisi saklayabilir.

**OMiktar** (Ödeme Miktarı): Bu alan, ödemenin miktarını (tutarını) temsil eder. INT veri türü, tam sayı değerleri saklar.

### 3. Duyurular Tablosu

Alanlar:

**Uid** (User ID): Bu alan, benzersiz bir kimlik numarasını temsil eder ve otomatik artan bir sayıdır (IDENTITY). Her yeni kayıt eklenirken, bu alan otomatik olarak bir artar.

**tarih** (Tarih): Bu alan, duyurunun tarih bilgisini depolar. DATETIME veri türü, hem tarih hem de saat bilgisini içerebilir.

**konu** (Konu): Bu alan, duyurunun konusunu temsil eder. VARCHAR (30) veri türü, en fazla 30 karakter uzunluğunda metin bilgisi saklayabilir.

**aciklama** (Açıklama): Bu alan, duyurunun detaylı açıklamasını içerir. VARCHAR (550) veri türü, en fazla 550 karakter uzunluğunda metin bilgisi saklayabilir.

## Proje Detayları ve Teknik Bilgiler

### Kullanılan Kütüphaneler

Projede, aşağıdaki önemli kütüphaneler kullanılmıştır:

**System.Data.SqlClient:** Veritabanı işlemleri için kullanılmıştır. SqlConnection, SqlCommand, SqlDataAdapter vb. sınıfları içerir.

**System.Windows.Forms:** Windows Forms uygulaması oluşturmak ve kullanıcı arayüzü işlemleri için kullanılmıştır. Formlar, kontroller ve diğer GUI elemanlarını içerir.

**System.Drawing:** Görsel elemanlar ve resim işlemleri için kullanılmıştır. PictureBox, Bitmap vb. sınıfları içerir.

## Proje Akışı

**Giriş Formu:** Yönetimden biri, yönetici olarak sisteme giriş yapar.

**Ana Sayfa:** Genel bir bakış sağlar ve temel işlemlere yönlendirir.

**Üye Kayıt Formu:** Üye kayıtlarıyla ilgili işlemleri gerçekleştirir.

**Üye Güncelleme Formu** Üye güncellemeleri ilgili işlemleri gerçekleştirir.

**Ödeme Formu** Üye ödemeleri ilgili işlemleri gerçekleştirir.

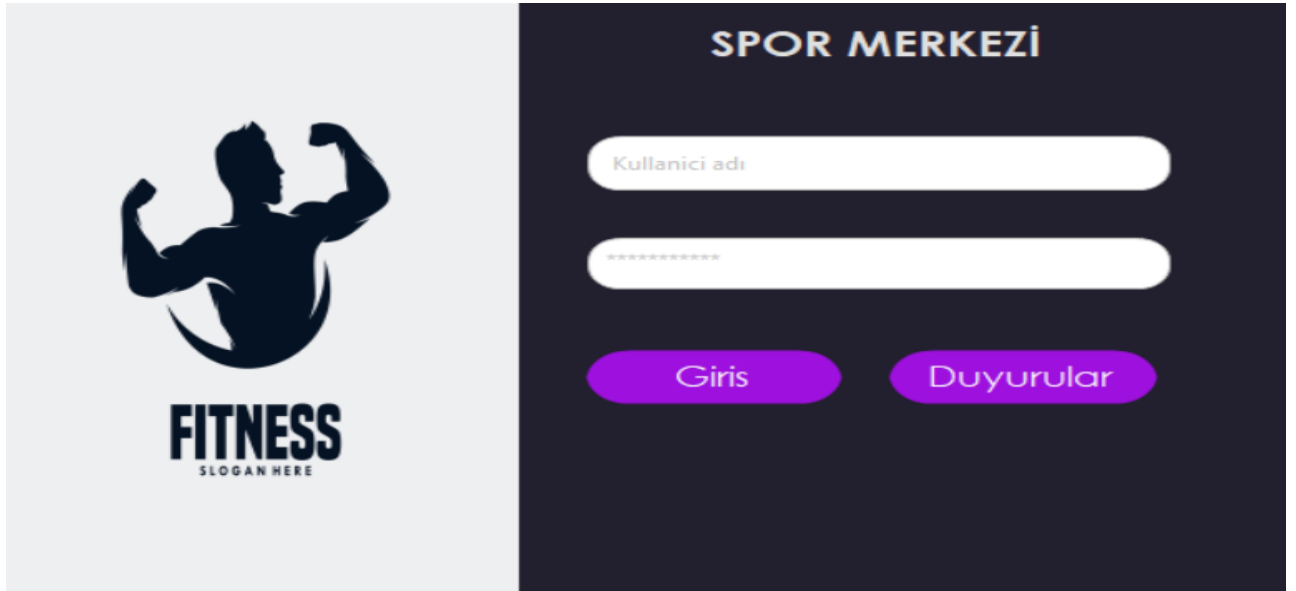
**Planlama Formu** Üyeye ait bireysel spor programı ilgili işlemleri gerçekleştirir.

**Duyuru Organize Formu** Duyuru ilgili işlemleri gerçekleştirir.

## Kullanıcı Arayüzü Örnekleri Formlar ve İşlevsellik

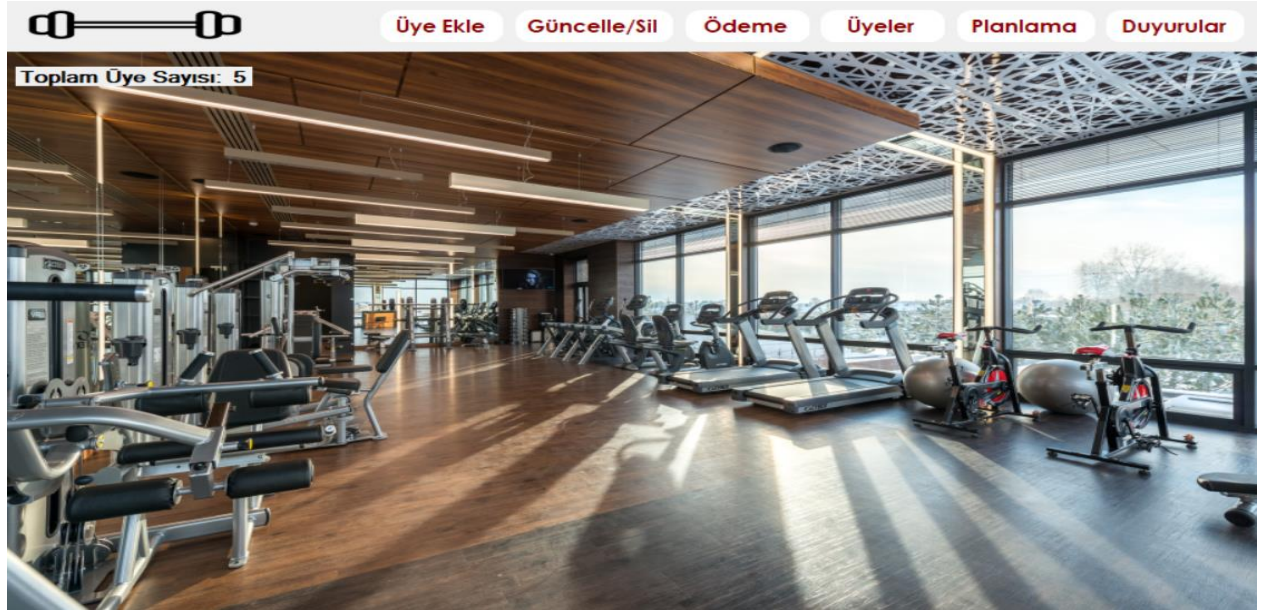
### 1.Giriş Ekranı ve Kullanıcı Yetkilendirme (Login Paneli):

Kullanıcıların kimlik doğrulaması için bir giriş ekranı. Kullanıcı adı ve şifre ile oturum açma. Yetkilendirme işlemiyle erişim kontrolü.



## 2.Menü ve Üst Navbar:

Açıklama: Kullanıcıya erişimi kolaylaştırmak için menü ve navigasyon çubuğu



## 3.Üye Ekle: Formu

Açıklama: Yeni üye bilgilerinin kaydedilmesi. Kişisel bilgiler, iletişim detayları, abonelik tipi gibi bilgilerin alınması

## SPOR MERKEZİ

### Yeni Üye Ekle

X

Üye Adı Soyadı

Enter text

Telefon

Enter text

Yaş

Enter text

Cinsiyet

▼

Aylık Tutar

Enter text

Zamanlama

▼

Minimum 30 günlük yapabilirsiniz!

Spor Yapılacak Ay

7.01.2024

Spor Yapılacak Ay

7.01.2024

Ekle

Reset

Geri

## 4. Üye Güncelleme:

Açıklama: Varolan üye bilgilerinin güncellenmesi. Üyenin bilgilerinin değiştirilebilmesi (adres, telefon numarası vb.).

## SPOR MERKEZİ

### Uye Güncelleme Ve Silme

X

Üye Adı Soyadı

Enter text

Cinsiyet

▼

Telefon

Enter text

Aylık Tutar

Enter text

Yaş

Enter text

Zamanlama

▼

Spor Başlangıç Ay

7.01.2024

Bitiş Tarihi

7.01.2024

Silinecek Üyeye tıklayınız

UId	UAdsoyad	Utelefon	UCinsiyet	UYas	UOdeme	UZamanlam	UBaslangic	UBitis	UAcklama
27	Ali Can	053654789	Bay	23	220	12.00-14....	17.12.20...	19.02.20...	Odeme Y...
28	Selime Ca...	0536456...	Bayan	27	2000	18.00-20....	17.12.20...	1.03.202...	Odenen ...
29	Mehmet ...	0555555...	Bay	21	250	16.00-18....	20.12.20...	24.01.20...	Odenen ...
30	Asaf Elbol	5364789...	Bay	28	180	16.00-18....	2.01.202...	23.02.20...	Odeme Y...

Güncell

Reset

Sil

Geri

## 5. Üyeleri Görüntüleme:

Açıklama: Tüm üyelerin listesinin görüntülenmesi :Filtreleme seçenekleri (abonelik türüne göre, isme göre vb.).



**SPOR MERKEZİ**  
Üyeleri Listele

<-- Geri X

Enter text Ara Reset

UId	UAdsoyad	Utelefon	Ucinsiyet	UYas	UOdeme	UZamanlama	UBaslangic	UBitis	UAciklama
55	Ali Can	053654789	Bay	23	220	12.00-14.00	17.12.2023 ...	19.02.2024 ...	Odeme Yap...
28	Selime Can...	05364569875	Bayan	27	2000	18.00-20.00	17.12.2023 ...	1.03.2024 0...	Odenen Mi...
29	Mehmet Vural	05555555555	Bay	21	250	16.00-18.00	20.12.2023 ...	24.01.2024 ...	Odenen Mi...
30	Asaf Elbol	53647896546	Bay	28	180	16.00-18.00	2.01.2024 1...	23.02.2024 ...	Odeme Yap...

## 6. Ödeme Ekranı Formu

Açıklama: Bu form, :Üyelerin ödeme geçmişini görüntüleme ve yönetmeyi sağlar. Borç durumu, ödeme tarihleri, ödeme miktarları gibi bilgilerin takibi

SPOR MERKEZİ

Ödemeler

<-- Geri X

Ödeme Ayı

7.01.2024

Üye Adı Soyadı

Ali Can

Tutar

Enter text

Reset

Ödeme

Enter text

Ara

Yenile

Oid	OAy	OUye	OMiktar
12	122023	Bilgehan Kayim	250
13	122023	Ali can	660
14	122023	Mert Örs	620
15	122023	Ali Can	600
16	122023	Ali Can	200
17	122023	Ali Can	220
18	122023	Selime Canseven	-500
19	122023	Mehmet Vural	450
20	12024	Asaf Elibol	450
21	12024	Asaf Elibol	450

## 7. Program Oluşturma Ekranı Formu

Açıklama: Bu form, Üyelere özel spor programlarının oluşturulması.Antrenman rutinleri, süreçlerin planlanması ve güncellenmesi.



SPOR MERKEZİ

Planlama

<-- Geri X

Cinsiyet

Bay

Çalışacağın Bölge

Göğüs

Kaç günde istiyorsun

8-10

Reset

Oluştur

Yazdır

**Sporcu Programı:**  
**Cinsiyet: Bay**  
**Çalışılacak Bölge: Göğüs**  
**Gelişim Süresi: 8-10**  
**Bay için Göğüs Programı:**  
- Isınma: 10 dakika koşu, 5 dakika göğüs germe  
- Bench press: 4 set x 10 tekrar  
- Dumbbell flys: 3 set x 12 tekrar  
- Push-ups: 3 set x 15 tekrar  
**Geçersiz süre seçimi!**

## 8. Duyurular Ekranı Formu

Spor salonuyla ilgili duyuruların yayınlanması. Etkinlikler, kapanış saatleri, özel teklifler gibi bilgilerin paylaşılması.



### **3. Inheritance (Miras):**

Proje içerisinde örneğin miras konusunu açıkça kullanmış gibi görünmeyebilir. Ancak, bu proje küçük ölçekli olduğu için geniş bir miras yapısına gerek olmayabilir. Mirasın kullanımı, genellikle sınıflar arasında ortak olan özelliklerin ve metodların paylaşılmasını sağlar. Daha büyük ölçekli projelerde miras kullanımına daha fazla rastlanabilir.

### **4. Polymorphism (Çok Biçimlilik):**

Projede, bir nesnenin farklı durumlarını veya formlarını yönetmek amacıyla çok biçimlilik kullanılabilir. Örneğin, yazdırma işlevselliğini sağlayan Yazdır sınıfı, farklı veri türleri üzerinde çalışabilir ve bu sayede çeşitli raporları yazdırabilir.

Bu OOP prensipleri sayesinde, proje daha okunabilir, bakımı daha kolay ve genişletilebilir bir yapıya kavuşur. Modüler ve düzenli bir programlama yaklaşımı benimsenmiştir, bu da kodun sürdürülebilirliğini artırır.

## **Tartışma ve Sonuç**

### **Tartışma:**

#### **Modüler ve Organize Yapı:**

Proje, OOP prensipleri kullanılarak modüler bir yapıya sahiptir. Bu, her sınıfın belirli bir işlevselliği içermesini ve bu sayede projenin genelinde düzenin korunmasını sağlar.

#### **Veritabanı Yapısı:**

Veritabanı tasarımı, kurs yönetimi, öğrenci kayıtları ve puanlama sistemini başarılı bir şekilde destekler. Bu sayede sistem, verileri düzenli bir şekilde depolar ve yönetir.

#### **Kullanıcı Arayüzü ve İşlevselliği:**

Kullanıcı arayüzü, temiz ve anlaşılır bir yapıya sahiptir. Kullanıcıların öğrenci kayıtları, kurs yönetimi ve puanlama gibi işlemleri kolayca gerçekleştirebilmeleri sağlanmıştır.

#### **OOP Prensipleri:**

Sınıflar arasında kapsülleme ve miras prensipleri başarıyla uygulanmıştır. Bu, her sınıfın kendi sorumluluklarına odaklanmasını ve kodun daha sürdürülebilir olmasını sağlar.



### **Geniřletilebilirlik ve Okunabilirlik:**

OOP sayesinde, projenin geniřletilmesi kolaydır. Yeni özellikler eklenmesi veya mevcut özelliklerin deęiřtirilmesi, kodun anlaşılabilirliğini ve bakımını artırır.

### **Sonuç:**

Bu projede, Nesne Yönelimli Programlama prensiplerinin etkin bir şekilde uygulanmasıyla bir dil kursu otomasyon sistemi başarıyla geliştirilmiştir. Projenin başarılı bir şekilde tamamlanması, OOP'nin yazılım geliştirme süreçlerindeki önemini vurgular.

OOP prensipleri, kodun düzenli ve modüler olmasını sağlayarak projenin sürdürülebilirliğini artırır. Ayrıca, gelecekteki deęişikliklere daha iyi uyum sağlar ve yeni özelliklerin eklenmesini kolaylaştırır.

**Sonuç olarak, Afyon Kocatepe Üniversitesi Mühendislik Fakóltesi Nesne Tabanlı Programlama dönem sonu projesi olan FİTNESS CENTER otomasyon sistemi, hem kullanıcı dostu arayüzüyle hem de OOP prensiplerinin başarılı bir şekilde uygulanmasıyla dikkat çekmektedir. Bu proje, öğrencinin OOP konseptlerini anlamasını ve uygulamasını sağlamak adına etkili bir örnek teşkil etmektedir.**

## **Teşekkürler**

Bu projenin tamamlanmasında katkıda bulunan herkese içten teşekkürlerimi sunarım.

Özellikle öğretim görevlimiz Sayın Naim Karasekreter'e bu projenin her aşamasında gösterdiği rehberlik ve destek için teşekkür ederim. Deęerli görüşleri ve yönlendirmeleri, projenin kalitesini artırmama büyük katkı sağladı

Proje sürecinde bana yardımcı olan tüm arkadaşlarıma ve öğrenci dostlarıma teşekkür ederim. Birlikte geçirdiğimiz zaman, yaşadığımız tartışmalar ve sağladığınız geri bildirimler, projenin gelişimine önemli katkılarda bulundu.

Aileme ve arkadaşlarıma, bu süreçteki destekleri, anlayışları ve motivasyonları için teşekkür ederim. Sizlerin desteęi olmadan bu projeyi başarıyla tamamlamak mümkün olmazdı.

Her birinize içten teşekkürler.

Gürkan Şahin

