

**2024-2025 AKADEMİK YILI
GÜZ DÖNEMİ**

BMT207 – VERİ YAPILARI

Ders Sorumlusu:
ARŞ. GÖR. DR. TUNAHAN TİMUÇİN

**Dernek Otomasyon
Dokümantasyonu**

Hazırlayan:
Eray OKUKLU

Öğrenci No:
192114043

Veri Yapıları Projesi

Dernek Uygulaması Dökümantasyonu

Proje Amacı

Bu proje, üyelik bazlı bir yönetim sistemi geliştirmek amacıyla oluşturulmuştur. Kullanıcılar, sisteme yeni üyeler ekleyebilir, mevcut üyelerin bilgilerini görüntüleyebilir ve gerekirse güncelleyebilir. Sistem, kullanıcıların temel bilgilerini (ad, soyad, telefon numarası, e-posta adresi, adres gibi) güvenli bir şekilde saklar. Bununla birlikte, üyelerin güncel borç durumlarını takip eder ve üyelikle ilgili işlemleri merkezi bir noktada yönetir. Sistemin ana amacı, üyelik işlemlerini dijital ortamda daha verimli ve düzenli bir şekilde yapabilmektir.

Proje Kapsamı

Bu yazılım, özellikle küçük ve orta ölçekli organizasyonlar, kulüpler veya benzer üyelik tabanlı yapılar için tasarlanmıştır. Sistemin kullanımı, hem yeni üyelerin sisteme eklenmesi hem de mevcut üyelerin verilerinin güncellenmesi işlemlerini kolaylaştırır. Sistem, kullanıcı dostu bir arayüz ile üyelik işlemleri sırasında kullanıcıya yardımcı olur.

Kullanılan Teknolojiler:

- .NET Framework (WinForms)
- C#
- Access database
- Visual Studio 2022

İşlevsel Özellikler

- Üye Ekleme:** Yeni üyelerin sisteme eklenmesini sağlar. Kullanıcı, gerekli bilgileri girerek yeni bir üye kaydı oluşturur.
- Üye Güncelleme:** Mevcut üyelerin bilgilerini güncellemeye olanak tanır. Kullanıcı, üyelerin ad, soyad, telefon gibi bilgilerini düzenleyebilir.
- Üye Silme:** Sistemdeki mevcut bir üyeyi silmeye olanak tanır. Kullanıcı, belirli bir üyeyi sistemden tamamen kaldırabilir.
- Aidat Ödeme:** Üyelerin aidat ödemelerini sisteme kaydetmeyi sağlar. Kullanıcı, borçlu olan üyeler için ödeme işlemi gerçekleştirebilir.
- Aidat Listesi Görüntüleme:** Sistemdeki tüm üyelerin aidat ödeme durumlarını liste halinde görüntüler. Kullanıcı, üyelerin ödenmemiş aidatlarını takip edebilir.

6. **Etkinlik Oluřturma:** Yeni etkinlikler d zenlemek ve kaydetmek i in kullanıcıya imkan tanır. Kullanıcı, etkinlik adı, tarih, açıklama gibi bilgileri girerek etkinlik oluřturabilir.
7. **Duyuru Oluřturma:** Sistemde genel duyurular yapmaya olanak saęlar. Kullanıcı, t m  yelere ulařacak duyurular hazırlayabilir ve bu duyuruları sisteme ekleyebilir.

Kullanıcı Aray z 

1. Giriř Ekranı(Giris.cs):

Bu form sayfası bir kullanıcı adı ve giriřinden oluřuyor.

Textbox1: kullanıcı adı bilgisini tutuyor

Textbox2: kullanıcı řifresini tutuyor.

Ve ek olarak bu sayfaya bir timer ekledim. Bu timer her ayın 1inde otomatik olarak 200 t  t m kullanıcılara aidat eklemektedir.

2. Ana Men (AnaMenu.cs):

Bu form sayfası bir katalog g revi g r yor ve  ye iřlemleri sayfasını, aidat tahsilat sayfasını ve etkinlik duyuru sayfasını ayırarak kullanıcıya sunuyor. Bu kısımda 3 button bulunuyor.

Button1:  ye iřlemleri sayfasına ge iř saęlıyor.

Button2: Aidat Tahsilat sayfasına ge iř saęlıyor.

Button3: Etkinlik ve Duyuru sayfasına ge iř saęlamaktadır.

3.  ye iřlemleri (UyeIslem.cs):

Bu sayfa kendi i inde 3 par aya ayrılıyor ve 3 butona sahip. bu sayfalar;  ye ekle veya g ncelle,  ye kaydı silme ve  yeleri listeleme.

Button1:  ye ekleme veya g ncelleme sayfasına ge iř saęlıyor.

Button2:  ye silme sayfasına ge iř saęlıyor.

Button3:  ye listeleme sayfasına ge iř saęlıyor.

4.  ye Ekleme , G ncelleme(UyeEkleGuncelle.js):

Bu sayfa, kullanıcıların yeni  yeleri sisteme eklemelerine veya mevcut  yelerin bilgilerini g ncellemelerine olanak tanır. i erięinde,  ye eklemek ve g ncellemek i in kullanılan  eřitli iřlevsellikler bulunmaktadır. Bu sayfa ek olarak bir tooltip'e sahip ve bu kullanıcının telefon numarası girerken hangi formata uyması gerektięi hakkında kullanıcıya bilgi vermektedir.

Button1:  ye Ekleme, yeni bir  ye eklemek i in kullanılmaktadır. Kullanıcı, gerekli alanları doldurduktan sonra bu butona tıklayarak yeni bir  ye kaydedebilir.

Button2:  ye G ncelleme, Bu buton, sistemdeki mevcut  yelerin bilgilerini g ncellemek

için kullanılır.

Button3: Üye Bilgilerini Getir, Bu buton, kullanıcının girdiği üye ID'sine göre üyeye ait bilgilerin otomatik olarak doldurulmasını sağlar.

Button5: Alanları Temizle, Bu buton, tüm form alanlarını temizler.

LinkLabel1: Bu link, kullanıcıyı Liste sayfasına yönlendirir.

5. Üye Silme(UyeCikar.js):

Bu sayfa, sisteme kaydedilmiş bir üyenin silinmesi işlemini gerçekleştirir. Kullanıcı, silmek istediği üyenin ID'sini girerek işlem yapabilir. ID' primary key olarak atanmıştır.

Button1: Bu buton, girilen üye ID'sine sahip üyeyi silme işlemine yönlendiren işlemi tetikler. Önce, kullanıcıdan alınan ID'nin geçerliliği kontrol edilir. Eğer geçerli bir ID girilmişse bu ID'ye sahip bir üyenin olup olmadığı kontrol edilir. Eğer üye bulunursa, FonksiyonK.UyeSil(uyeID) fonksiyonu çalıştırılarak üye silinir. Silme işlemi başarılı olduğunda, bir bilgi mesajı gösterilir.

Button4: Bu buton, UyeIslem sayfasına geri dönmeyi sağlar.

LinkLabel1: Bu link, kullanıcıyı Liste sayfasına yönlendirir.

6. Üye Listeleme(UyeListele.js):

Bu sayfa, Üye Listeleme işlemini gerçekleştiren bir formu temsil ediyor. Kullanıcıların bilgilerini veritabanından alarak, bir DataGridView kontrolünde listelemektedir.

Button1: Bu buton veritabanındaki tüm üyeleri alır. Eğer veritabanında üye yoksa, kullanıcıya "Veritabanında üye bulunamadı" mesajı gösterilir. Eğer üyeler varsa, üyelerin bilgileri bir DataGridView'de listelenir.

Button4: Bu buton, UyeIslem sayfasına geri dönmeyi sağlar. Yani, üyeler listelendikten sonra başka bir işlem yapmak için bu sayfaya geçiş yapılabilir.

7. Etkinlik Duyuru Menüsü(EtkinlikDuyuru.cs):

Bu sayfa, Etkinlik Duyuru Ana Sayfası olarak işlev görmektedir. Kullanıcıların etkinlikler ve duyurularla ilgili işlemleri başlatmasına olanak sağlar.

Button1: Bu buton, kullanıcıyı Etkinlik Oluşturma sayfasına yönlendirir. Kullanıcı bu butona tıkladığında, Etkinlik formu açılır ve mevcut sayfa gizlenir. Bu sayfa etkinlik oluşturma işlemleri için kullanılır.

Button2: Bu buton, kullanıcıyı Duyuru Oluşturma sayfasına yönlendirir. Kullanıcı bu butona tıkladığında, Duyuru formu açılır ve mevcut sayfa gizlenir. Bu sayfa duyuru oluşturma işlemleri için kullanılır.

Button4: Bu buton, Ana Menü sayfasına yönlendirir. Kullanıcı bu butona tıklayarak ana menüye geri dönebilir.

8. Etkinlik Oluşturma Sayfası (Etkinlik.cs):

Bu sayfa, kullanıcıların etkinlik oluşturmalarını ve bu etkinlik hakkında e-posta göndermesini sağlar.

button1: Bu buton, etkinlik oluşturulup e-posta gönderme işlemini başlatır. İşlem

adımları şu şekildedir:

1. Etkinlik Adı, Etkinlik Yeri ve Etkinlik Saati bilgileri ilgili metin kutularından ve tarih seçiciden alınır.
2. İş katmanından EtkinlikEpostasiGonder fonksiyonu çağrılır ve etkinlik bilgileri e-posta ile gönderilir.
3. İşlem başarılı olursa, kullanıcıya bir başarı mesajı gösterilir.
4. Eğer işlem sırasında bir hata oluşursa, kullanıcıya hata mesajı gösterilir.

button4: Bu buton, kullanıcıyı Etkinlik Duyuru Ana Sayfasına yönlendirir. Etkinlik oluşturma sayfası gizlenir ve duyuru sayfası gösterilir.

9. Duyuru Oluşturma Sayfası (Duyuru.cs):

Bu sayfa, kullanıcıların duyuru metni girerek, ilgili kişilere e-posta ile duyuru göndermelerini sağlar.

button1: Bu buton, kullanıcıların duyuru metni girip göndermelerini sağlar. İşlem adımları şu şekildedir:

1. Duyuru Metni, kullanıcının girdiği metin kutusundan alınır.
2. Her e-posta adresine duyuru gönderilir.
3. Duyuru gönderme işlemi başarılı olursa, kullanıcıya bir başarı mesajı gösterilir.

10. Aidat Tahsilat Sayfası (AidatTahsilat.cs):

Bu sayfa, aidat tahsilat işlemlerini yönetmek için kullanılan bir menüdür. Kullanıcılar bu menü aracılığıyla aidat ödeme ve borç gösterme işlemlerini gerçekleştirebilirler.

Button1: Aidat ödeme sayfasına geçiş sağlar.

Button2: Borç gösterme sayfasına geçiş sağlar.

11. Aidat Ödeme Sayfası (AidatOde.cs):

Bu sayfa, kullanıcıların üyelik aidatlarını ödeyebilecekleri bir formdur. Kullanıcılar, belirli bir üyenin borç bilgilerini görüntüleyebilir ve ödeme işlemi gerçekleştirebilirler.

Button1: Bu buton, kullanıcıdan ID girmesini ister ve ardından bu ID'ye sahip üyenin bilgilerini getirir. Üye bulunduğunda, adı-soyadı ve güncel borç bilgileri ekrana gelir ve ödeme işlemi yapılabilir hale gelir. Eğer kullanıcı ID girmemişse, hata mesajı gösterilir.

Button2: Bu buton, ödeme işlemi yapar. Kullanıcı, ödeme miktarını girdikten sonra, girilen miktarın üyenin borcunu aşmaması gerektiği kontrol edilir. Ödeme başarılı bir şekilde yapıldığında, ödeme miktarı üyenin borcundan düşülür ve makbuz oluşturulur. Ayrıca, kullanıcıya başarılı bir mesaj gösterilir.

Button4: Bu buton, kullanıcıyı AidatTahsilat formuna yönlendirir. Bu sayfa gizlenir ve AidatTahsilat sayfası açılır.

Linklabel: Bu etiket, Liste formunu açar. Kullanıcı buradan borç listesini görüntüleyebilir.

12.Borç Gösterme Sayfası (BorcGöster.cs):

Bu sayfa, üyelerin borçlarını görüntülemek amacıyla kullanılan bir formdur. Kullanıcılar, ödeme işlemi veya borç bilgilerini görmek için bu sayfayı kullanabilirler. Sayfa, üyelerin ad-soyad, telefon numarası ve güncel borç bilgilerini liste halinde görüntüler.

Button1: Bu buton, üyelerin borçlarını listeleyen işlemi başlatır. Borç bilgileri alınır ve bu bilgiler dataGridView1 kontrolünde tablo olarak gösterilir. Tabloya, üye adı, telefonu ve güncel borcu eklenir. Eğer bir hata oluşursa, kullanıcıya hata mesajı gösterilir.

Button4: Bu buton, kullanıcıyı AidatTahsilat formuna yönlendirir. Borç gösterme sayfası gizlenir ve AidatTahsilat sayfası açılır.

13.Liste Sayfası (Liste.cs):

Bu sayfa, veritabanındaki üyelerin bilgilerini listeleyen bir formdur. Kullanıcı, üyelerin bilgilerine kolayca erişebilir. Bu sayfa ile üyelerin tüm bilgilerini bir DataGridView kontrolünde kullanıcıya gösterir.

VERİ ERİSİM KATMANI

Veri erişim katmanı, uygulamanın veritabanıyla etkileşime giren ve veri işlemlerini gerçekleştiren katmandır. Bu katman, veritabanındaki verilere erişim sağlar, veri ekleme, güncelleme, silme ve okuma gibi temel işlemleri gerçekleştirir. Uygulamanın diğer katmanlarından gelen istekleri alır ve verileri veritabanından alarak veya veritabanına göndererek iş katmanına iletir.

- 1. UyeVarMi fonksiyonu:** Bu fonksiyon, belirli bir ID'ye sahip bir üyenin veritabanında bulunup bulunmadığını kontrol eder. Eğer varsa true, yoksa false döndürür.
- 2. UyeEkle fonksiyonu:** Bu fonksiyon, UyeEkle, yeni bir üye eklemek için kullanılır. Verilen Uye nesnesine ait bilgileri veritabanındaki Uyeler tablosuna ekler.
- 3. UyeSil fonksiyonu:** UyeSil fonksiyonu, verilen uye ID parametresine sahip olan kullanıcıyı veritabanındaki Uyeler tablosundan siler.
- 4. UyeListele fonksiyonu:** Bu fonksiyon, UyeListele, veritabanındaki Uyeler tablosundaki tüm üyeleri listelemek için kullanılır. Veritabanındaki tüm üyeleri alarak bir List<Uye> olarak döndürür. Her bir üye, Uye sınıfına ait bir nesne olarak listeye eklenir.
- 5. UyeGuncelle fonksiyonu:** Veritabanındaki mevcut bir üyenin bilgilerini güncellemek. Uye nesnesi ile gelen yeni verilerle, veritabanındaki kayıt güncellenir.
- 6. UyeIDileGetir fonksiyonu:** belirli bir üyenin ID'sine göre veritabanından kullanıcı bilgilerini getirir.

- 7. UyeTumBilgileriniGetir fonksiyonu:** Bu UyeTumBilgileriniGetir fonksiyonu, belirli bir ID'ye sahip üyenin tüm bilgilerini veritabanından çeker ve Uye sınıfına atayarak geri döndürür.
- 8. UyeBilgileriniGetir fonksiyonu:** Bu UyeBilgileriniGetir fonksiyonu, belirli bir ID'ye sahip üyenin temel bilgilerini (AdSoyad ve GüncelBorc) veritabanından çeker ve bir Uye nesnesi olarak geri döndürür.
- 9. AidatOde fonksiyonu:** Bu AidatOde fonksiyonu, belirli bir ID'ye sahip üyenin GüncelBorc değerini, girilen ödemeMiktari kadar azaltarak günceller. Yani, üye ödeme yaptığı zaman borç tutarını günceller.
- 10. BorclariListele fonksiyonu:** BorclariListele fonksiyonu, veritabanındaki üyelerin sadece AdSoyad, Telefon ve GüncelBorc bilgilerini listelemek amacıyla tasarlanmıştır. Bu fonksiyonun amacı, üyelerin borç bilgilerini kullanıcıya göstermek için veritabanından alınan verilere dayalı bir liste oluşturmak.
- 11. UyeEmailListele fonksiyonu:** veritabanındaki tüm üyelerin Email bilgilerini alıp bir liste halinde döndüren bir fonksiyondur.
- 12. AidatGuncelle fonksiyonu:** belirli bir üyeye ait borcu güncellemek için kullanılan bir fonksiyondur. Bu fonksiyon, üyeye aidat ekler ve borç bilgisini günceller.

İŞ KATMANI

İş katmanı, uygulamanın iş mantığını ve kurallarını yöneten katmandır. Bu katman, uygulamanın kullanıcıdan veya dış sistemlerden aldığı verileri işleyerek uygun sonuçlar üretir. Veritabanı işlemleri ve kullanıcı arayüzü katmanları ile doğrudan etkileşime girmez; bunun yerine, veri erişim katmanını kullanarak gerekli verileri alır, işler ve gerektiğinde bu verileri kullanıcılara veya sistemlere iletir.

MakbuzOlustur fonksiyonu: Bu fonksiyon, ödeme yapılan bir üyeye ait makbuz oluşturur ve PDF formatında kaydeder. PDF dosyası, üyenin bilgilerini, ödenen tutarı, kalan borcu ve ödeme tarihini içerir.

EtkinlikEpostasiGonder fonksiyonu: Bu fonksiyon, belirli bir etkinlik hakkında tüm üyelere e-posta gönderir. Etkinlik adı, yeri ve saati ile ilgili bilgileri içeren bir mesaj oluşturur ve tüm üyelere toplu e-posta gönderir.

MailGonder fonksiyonu: Bu fonksiyon, belirli bir alıcıya e-posta göndermek için kullanılır. E-posta içeriğini HTML formatında belirler ve SMTP üzerinden gönderir.

DuyuruGonder fonksiyonu: Bu fonksiyon, belirli bir alıcıya duyuru amacıyla e-posta göndermek için kullanılır. E-posta içeriğini HTML formatında belirler ve SMTP üzerinden gönderir.

AidatGuncellemeTumu fonksiyonu: Bu fonksiyon, sistemdeki tüm üyelerin aidatlarını günceller. Fonksiyon, her üye için AidatGuncelle fonksiyonunu çağırarak her birinin güncel borcunu belirli bir aidat tutarı kadar artırır.

AidatGuncelle fonksiyonu: Bu fonksiyon, belirli bir üyeye ait borç miktarını günceller. Fonksiyon, üyeyi belirlemek için uyeID parametresini kullanır ve aidat parametresini mevcut borca ekler.

Bu proje, yazılım geliştirme sürecinde katmanlı mimari prensiplerine uygun şekilde tasarlanmış ve geliştirilmiştir. Katmanlı mimari, yazılımın farklı sorumluluklara sahip katmanlara ayrılması prensibine dayanır. Bu sayede, her bir katman belirli bir işlevi yerine getirirken, diğer katmanlarla olan bağımlılıklar minimuma indirgenmiştir. Projemizde Sunum Katmanı , İş katmanı ve Veri erişim katmanı bulunuyor.