

**T.C.  
İSTANBUL AYDIN ÜNİVERSİTESİ**



**CAFE OTOMASYONU PROJESİ**

**ÖN LİSANS BİTİRME PROJESİ**

**Enes Talip İşbilen**

Bilgisayar Programcılığı

HAZİRAN, 2022



**T.C.  
İSTANBUL AYDIN ÜNİVERSİTESİ  
ANADOLU BİL MESLEK YÜKSEKOKULU**

**CAFE OTOMASYONU PROJESİ**

**ÖN LİSANS BİTİRME PROJESİ**

**Enes Talip İşbilen**

U2011.040036

Bilgisayar Programcılığı

**Tez Danışmanı: Dr. Öğr. Üyesi Hakan Burak EMEKLİ**

HAZİRAN, 2022

## ÖNSÖZ

Bu çalışmanın konusunun belirlenmesinde, hazırlanma sürecinin her aşamasında değerli bilgilerini ve zamanını benden esirgemeyerek her fırsatta çalışmamda yakından ilgilenen, eleştirileriyle yol gösteren danışman hocam Dr. Öğr. Üyesi Hakan Burak EMEKLİ, aynı zamanda teşekkürlerin az kalacağı diğer üniversite hocalarımın da üniversite hayatım boyunca bana kattıkları tüm değerler ve tecrübeler için hepsine teker teker sonsuz teşekkürlerimi sunuyorum.

Beni bu günlere kadar getiren, hayatın her alanında yanımda olup sevgi, saygı ve büyük bir emekle beni yetiştiren sevgili anne ve babama teşekkürlerimi sunuyorum.

## ÖZET

Günümüzde manuel olarak yapılan her iş dijitalleşme ile otomatikleştirilmeye başlanmıştır. Bu dönüşüm sürecinde işletmeler dijital dönüşümü yakalayabilmek adına bilişim teknolojileri adına yatırımlarını arttırmıştır. Projesini hazırladığım cafe otomasyonu uygulaması yeme içme sektöründe hizmet veren butik cafe ve restoranlar için önemli bir yer tutmaktadır. Bu otomasyon sayesinde kâğıt adisyonları bırakacak, verileri bilgisayar ortamında tutabilecek ve bu verilerden raporlar çekilebilecektir. Bu sayede kâğıt israfı önlenecek, işletmenin gelir ve giderler otomatik bir şekilde kaydedilecek, müşteriye daha gelişmiş bir hizmet sunulacaktır. Proje yapımında Visual Studio 2019 ve SQL Server kullanılmıştır. Kullanılan teknolojiler; .Net Framework, ADO.Net, C# programlama dili ve Windows forms uygulamalarıdır.

**Anahtar Kelimeler:** Visual Studio, Otomasyon, ADO.NET, C Sharp, .NET Framework

## **ABSTRACT**

Today, every work done manually has started to be automated with digitalization. In this transformation process, businesses have increased their investments in information technologies in order to achieve digital transformation. The cafe automation application for which I have prepared the project has an important place for boutique cafes and restaurants serving in the food and beverage sector. Thanks to this automation, it will be possible to leave paper tickets, keep the data in the computer environment and draw reports from these data. In this way, wastage of paper will be prevented, the income and expenses of the business will be recorded automatically, and a more advanced service will be offered to the customer. Visual Studio 2019 and SQL Server were used in the project construction. Used technologies; .Net Framework, ADO.Net, C# programming language and Windows forms applications.

**Keywords:** Visual Studio, Automation, ADO.NET, C Sharp, .NET Framework

# İÇİNDEKİLER

ÖNSÖZ .....	iv
ÖZET .....	v
ABSTRACT.....	vi
İÇİNDEKİLER .....	vii
ŞEKİLLER LİSTESİ.....	xi
KISALTMALAR LİSTESİ.....	xiv
GİRİŞ .....	15
<b>I. KULLANILAN TEKNOLOJİLER.....</b>	<b>16</b>
A. Microsoft Visual Studio 2019 .....	16
B. .NET Framework .....	18
C. ADO .NET .....	20
D. Windows Forms .....	22
<b>II. PROJE KLASÖR YAPISI .....</b>	<b>23</b>
<b>III. MODÜLLER VE MODÜLLERİN AÇIKLANMASI .....</b>	<b>25</b>
A. Genel Class .....	25
B. Adisyon Class .....	26
1. Adisyon Class Metotları.....	26
2. SetByAdditionNew Metodu.....	27
3. GetbyAddition Metodu .....	27
4. Adisyon Kapat .....	28
5. ID'ye Göre Adisyon Bul Metodu.....	28
6. Açık Paket Sipariş Adisyon Metodu .....	29
7. Müşterinin Son Adisyon ID'sini Getirme Metodu.....	29
8. Müşterinin Adisyon Detayları Metodu .....	30
9. Rezervasyona İstinaden Adisyon Aç Metodu .....	30
C. Masalar Classı.....	31
1. Masalar Class Metotları .....	31
2. Sessionsum Metodu .....	32
3. TableGetbyNumber Metodu .....	32
4. Masa Durumunu Getir Metodu .....	33
5. Masa Durumunu Değiştir Metodu .....	33
6. Masa Kapasitesi ve Durumu Getir Metodu.....	34
D. Müşteriler Class .....	35
1. Müşteriler Class Metotları .....	35

2.	Müşteri Var Mı Metodu .....	36
3.	Müşteri Ekleme Metodu.....	36
4.	Müşteri Bilgilerini Güncelleme Metodu .....	37
5.	Müşterileri ListView'e Getirme Metodu .....	37
6.	Müşterileri ID'ye Göre Getirme Metodu .....	38
7.	Müşterileri İsme Göre Listeleme Metodu .....	39
8.	Müşterileri Soyada Göre Listeleme Metodu .....	40
9.	Müşterileri Telefona Göre Listeleme Metodu.....	40
E.	Ödeme Class .....	41
1.	Ödeme Class Metotları.....	41
2.	Hesap Kapatma Metodu.....	42
3.	Müşteri ID'sine Göre Toplam Tutar Metodu.....	42
F.	Paketler Classı.....	43
1.	Paket Servis Class Metotları .....	43
2.	Paket Sipariş Aç Metodu.....	44
3.	Paket Sipariş Kapat Metodu.....	44
4.	Ödeme Türünü Getir Metodu.....	45
5.	Müşterinin Son Adisyonunu Getir Metodu.....	45
6.	Açık Adisyonun ID'sinin Getir Metodu .....	46
G.	Personel Görev Classı .....	46
1.	Personel Görev Metotları .....	47
2.	Personel Görevlerini ComboBox'a Getir Metodu .....	47
3.	Personel Görev Tanım Metodu .....	48
H.	Personel Hareketleri Classı .....	48
1.	Personel Hareketleri Class Metotları .....	49
2.	Personel Hareketleri Kayıt Metodu.....	49
İ.	Personeller Class .....	50
1.	Personeller Class Metotları .....	50
2.	Personel Giriş Kontrol Metodu .....	51
3.	Personel Bilgilerini ComboBox'a Getir Metodu .....	51
4.	Personel Bilgilerini ListView'e Getir Metodu.....	52
5.	Personel Bilgilerini ID'ye Göre ListView'e Getir Metodu.....	52
6.	Personel Bilgilerini İsme Göre Getir Metodu .....	52
7.	Personel Şifre Değiştir Metodu.....	53
8.	Personel Ekle Metodu .....	53
9.	Personel Güncelle Metodu .....	54



10.	Personel Sil Metodu .....	54
J.	Rezervasyon Class .....	55
1.	Rezervasyon Class Metotları .....	55
2.	Müşteri ID'sine Göre Rezervasyon Metodu .....	56
3.	Rezervasyon Kapatma Metodu .....	56
4.	Rezervasyonlardan müşteri ID'sini Getir Metodu .....	57
5.	Eski Rezervasyonları Getir Metodu .....	57
6.	En Son Rezervasyon Tarihini Getir Metodu .....	57
7.	Açık Rezervasyon Sayısını Say Metodu .....	58
8.	Rezervasyon Durum=1 Kontrol Metodu .....	58
9.	Rezervasyon Açma Metodu .....	58
10.	Rezerve Masa ID'sini Getir Metodu .....	59
K.	Sipariş Class .....	59
1.	Sipariş Class Metotları .....	60
2.	Siparişleri ListView'e Atama Metodu .....	60
3.	Siparişleri Kaydetme Metodu .....	61
4.	Siparişi Silme Metodu .....	61
5.	Sipariş Genel Toplam Metodu .....	62
6.	Paket Sipariş Adisyon Detayları ListView Metodu .....	62
L.	Ürün Çeşitleri Class .....	63
1.	Ürün Çeşitleri Class Metotları .....	63
2.	Ürün Çeşitlerini ListView'e Getir Metodu .....	64
3.	Ürün Arama Metodu .....	64
4.	Ürün Çeşitlerini ComboBox'a Getir Metodu .....	65
5.	Ürün Çeşitlerini ListView'e Getir Metodu .....	65
6.	Ürün Çeşitlerini Getir Metodu .....	66
7.	Ürün Kategori Ekle Metodu .....	66
8.	Ürün Kategori Güncelle Metodu .....	67
9.	Ürün Kategori Sil Metodu .....	67
M.	Ürünler Class .....	68
1.	Ürünler Class Metotları .....	68
2.	Ürün Adına Göre Ürünleri Listele Metodu .....	69
3.	Ürün Ekle Metodu .....	69
4.	Ürünleri ListView'e Ekle Metodu .....	70
5.	Ürün Güncelle Metodu .....	70
6.	Ürün Sil Metodu .....	71

7. Ürün Adına Göre Ürünleri Listele Metodu .....	71
8. Ürünleri İstatistiklere Göre Listele Metodu .....	72
9. Ürün ID'ye Göre Ürün İstatistiklerini Listele Metodu .....	72
<b>IV. WINDOWS FORMS .....</b>	<b>73</b>
A. Giriş Formu .....	73
B. Menü Formu .....	73
C. Masalar Formu .....	74
D. Rezervasyon Formu .....	75
E. Sipariş Formu .....	76
F. Mutfak Formu .....	77
G. Kasa İşlemleri Formu .....	78
H. Müşteri Formu .....	78
İ. Raporlar Formu .....	79
J. Ayarlar Formu .....	80
K. Kilit Formu .....	81
L. Ödeme Formu .....	81
<b>V. KAYNAKÇA .....</b>	<b>83</b>
.....	

## ŞEKİLLER LİSTESİ

Şekil 1: Cafe Otomasyonu Proje Klasör Yapısı.....	23
Şekil 2: Cafe Otomasyonu Class Yapısı .....	23
Şekil 3: Cafe Otomasyonu Form Yapısı .....	24
Şekil 4: Genel Classı Fields .....	25
Şekil 5: Adisyon Class Fields ve Properties .....	26
Şekil 6: Adisyon Class Metotları .....	26
Şekil 7: Yeni Adisyon Metodu .....	27
Şekil 8: Adisyon Seçme Metodu.....	27
Şekil 9: Adisyon Kapatma Metodu.....	28
Şekil 10: ID'ye Göre Adisyon Bul Metodu .....	28
Şekil 11:Açık Paket Sipariş Adisyon Metodu.....	29
Şekil 12:Müşterinin Son Adisyon ID'sini Getirme Metodu .....	29
Şekil 13: Müşterinin Adisyon Detayları Metodu.....	30
Şekil 14:Rezervasyona İstinaden Adisyon Aç Metodu.....	30
Şekil 15: Masalar Class Fields ve Properties .....	31
Şekil 16: Masalar Class Metotları.....	31
Şekil 17: Masanın Açık Kalma Süresi Metodu.....	32
Şekil 18: TableGetbyNumber Metodu .....	32
Şekil 19: TableGetbyState Metodu .....	33
Şekil 20: setChangeTableState Metodu .....	33
Şekil 21: MasaKapasitesiveDurumuGetir Metodu .....	34
Şekil 22: Müşteriler Class Fields ve Properties .....	35
Şekil 23: Müşteriler Class Metotları .....	35
Şekil 24: MusteriVarmi Metodu .....	36
Şekil 25: Müşteri Ekleme Metodu .....	36
Şekil 26: Müşteri Bilgilerini Güncelleme Metodu.....	37
Şekil 27: Müşterileri ListView'e Getirme Metodu .....	37
Şekil 28: Müşterileri ID'ye Göre Getirme Metodu.....	38
Şekil 29: Müşterileri İsme Göre Listeleme Metodu.....	39
Şekil 30: Müşterileri Soyada Göre Listeleme Metodu.....	40
Şekil 31: Müşterileri Telefona Göre Listeleme Metodu .....	40
Şekil 32: Ödeme Class Fields ve Properties .....	41
Şekil 33: Ödeme Class Metotları .....	41
Şekil 34: Hesap Kapatma Metodu .....	42
Şekil 35: Müşteri ID'sine Göre Toplam Tutar Metodu .....	42
Şekil 36: Paketler Class Fields ve Properties .....	43
Şekil 37: Paket Servis Class Metotları .....	43
Şekil 38: Paket Sipariş Aç Metodu .....	44
Şekil 39: Paket Sipariş Kapat Metodu .....	44
Şekil 40: Ödeme Türünü Getir Metodu .....	45
Şekil 41: Müşterinin Son Adisyonunu Getir Metodu .....	45
Şekil 42: Açık Adisyonun ID'sinin Getir Metodu .....	46
Şekil 43: Personel Görev Class Fields ve Properties .....	46
Şekil 44: Personel Görev Metotları.....	47
Şekil 45: Personel Görevlerini ComboBox'a Getir Metodu .....	47

Şekil 46: Personel Görev Tanım Metodu.....	48
Şekil 47: Personel Hareket Class Fields ve Properties.....	48
Şekil 48: Personel Hareketleri Class Metotları .....	49
Şekil 49: Personel Hareketleri Kayıt Metodu .....	49
Şekil 50: Personeller Class Fields ve Properties .....	50
Şekil 51: Personeller Class Metotları.....	50
Şekil 52: Personel Giriş Kontrol Metodu.....	51
Şekil 53: Personelleri ComboBox'a Getir Metodu .....	51
Şekil 54: Personel Bilgilerini ListView'e Getir Metodu.....	52
Şekil 55: Personel Bilgilerini ID'ye Göre ListView'e Getir Metodu .....	52
Şekil 56: Personel Bilgilerini İsme Göre Getir Metodu.....	52
Şekil 57: Personel Şifre Değiştir Metodu .....	53
Şekil 58: Personel Ekle Metodu.....	53
Şekil 59: Personel Güncelle Metodu.....	54
Şekil 60: Personel Sil Metodu.....	54
Şekil 61: Rezervasyon Class Fields ve Properties .....	55
Şekil 62: Rezervasyon Class Metotları .....	55
Şekil 63: Müşteri ID'sine Göre Rezervasyon Metodu.....	56
Şekil 64: Rezervasyon Kapatma Metodu.....	56
Şekil 65: Rezervasyonlardan müşteri ID'sini Getir Metodu.....	57
Şekil 66: Eski Rezervasyonları Getir Metodu.....	57
Şekil 67: Rezervasyon Tarihini Getir Metodu .....	57
Şekil 68: Açık Rezervasyon Sayısını Say Metodu.....	58
Şekil 69: Rezervasyon Durum=1 Kontrol Metodu .....	58
Şekil 70: Rezervasyon Açma Metodu.....	58
Şekil 71: Rezerve Masa ID'sini Getir Metodu.....	59
Şekil 72: Sipariş Class Fields ve Properties.....	59
Şekil 73: Sipariş Class Metotları.....	60
Şekil 74: Siparişleri ListView'e Atama Metodu.....	60
Şekil 75: Siparişleri Kaydetme Metodu .....	61
Şekil 76: Siparişi Silme Metodu .....	61
Şekil 77: Sipariş Genel Toplam Metodu.....	62
Şekil 78: Paket Sipariş Adisyon Detayları ListView Metodu.....	62
Şekil 79: Ürün Çeşitleri Class Fields ve Properties .....	63
Şekil 80: Ürün Çeşitleri Class Metotları.....	63
Şekil 81: Ürün Çeşitlerini ListView'e Getir Metodu.....	64
Şekil 82: Ürün Arama Metodu.....	64
Şekil 83: Ürün Çeşitlerini ComboBox'a Getir Metodu .....	65
Şekil 84: Ürün Çeşitlerini ListView'e Getir Metodu.....	65
Şekil 85: Ürün Çeşitlerini Getir Metodu.....	66
Şekil 86: Ürün Kategori Ekle Metodu .....	66
Şekil 87: Ürün Kategori Güncelle Metodu .....	67
Şekil 88: Ürün Kategori Sil Metodu .....	67
Şekil 89: Ürünler Class Fields ve Properties .....	68
Şekil 90: Ürünler Class Metotları .....	68
Şekil 91: Ürün Adına Göre Ürünleri Listele Metodu.....	69
Şekil 92: Ürün Ekle Metodu .....	69
Şekil 93: Ürünleri ListView'e Ekle Metodu .....	70

Şekil 94: Ürün Güncelle Metodu .....	70
Şekil 95: Ürün Sil Metodu .....	71
Şekil 96: Ürün Adına Göre Ürünleri Listele Metodu.....	71
Şekil 97: Ürünleri İstatistiklere Göre Listele Metodu.....	72
Şekil 98: Ürün ID'ye Göre Ürün İstatistiklerini Listele Metodu .....	72

## KISALTMALAR LİSTESİ

<b>SQL</b>	: Structured Query Language
<b>LINQ</b>	: Language Integrated Query
<b>IDE</b>	: Integrated Development Environment
<b>API</b>	: Application Programming Interface
<b>CLR</b>	: Common Language Runtime
<b>FCL</b>	: Framework Class Library
<b>CLI</b>	: Command-line interface
<b>KDV</b>	: Katma Değer Vergisi

## GİRİŞ

Geçmişte tüm belgelerin el yazısı ile tutulması işleri yavaşlatmaktaydı ve saklanabilirlik açısından zorlukları mevcuttu. Bu zorluklar bilişim dünyasına geçişi hızla arttırmıştır ve dijitalleşmeye verilen önem artmıştır.

Kaktüs cafe otomasyonu butik bir cafenin ihtiyacı olabilecek tüm araçları içermektedir. Visual studio IDE'sinde, C# programlama dili kullanılarak yazılmıştır. Veri tabanı bağlantısı ADO.NET teknolojisi kullanılarak yapılmıştır. Windows form araçlarıyla birlikte ekranlar oluşturulmuş ve kesintisiz bir çalışma zamanı sunulmuştur.

Bu otomasyonda, masaya istinaden adisyon ve satış açılması, siparişlerin girilmesi, siparişlerin ödeme yapılması ile kapatılması, günlük ve aylık rapor çekilmesi, kullanıcı işlemlerinin yapılması, geliştirilebilir zengin bir mutfak kısmının oluşturulması gibi birçok özelliği vardır.

Bu otomasyon geliştirilmeye açıktır. Farklı formlar ve özellikler eklenerek daha zengin bir otomasyon haline getirilebilir.

## **I. KULLANILAN TEKNOLOJİLER**

### **A. Microsoft Visual Studio 2019**

Microsoft Visual Studio, bir geliştirme ortamıdır (IDE). Bilgisayar programlarının yanı sıra web siteleri, web uygulamaları, web hizmetleri ve mobil uygulamalar geliştirmek için kullanılır. Visual Studio, IntelliSense'i destekleyen bir kod düzenleyicisinin yanı sıra kod yeniden düzenlemeyi içerir. Entegre hata ayıklayıcı hem kaynak düzeyinde hata ayıklayıcı hem de makine düzeyinde hata ayıklayıcı olarak çalışır.

Visual Studio, 36 farklı programlama dilini destekler ve dile özgü bir hizmetin mevcut olması koşuluyla, kod düzenleyicinin ve hata ayıklayıcının hemen hemen tüm programlama dillerini desteklemesine olanak tanır. Kullanılabilir diller arasında C, C++, C++/CLI, Visual Basic .NET, C#, F#,JavaScript, TypeScript, XML, XSLT, HTML ve CSS bulunur. Eklentiler aracılığıyla Python, Ruby, Node.js ve M gibi diğer diller için destek sağlanır. Visual Studio'nun en temel sürümü olan topluluk sürümü ücretsizdir. Güncel Visual Studio sürümü 2022'dir.

### **Özellikler**

#### **Kod düzenleyici**

Visual Studio, değişkenler, döngüler ve LINQ sorguları için IntelliSense kullanarak sözdizimi vurgulamayı ve kod tamamlamayı destekleyen bir kod düzenleyici içerir. IntelliSense, web siteleri ve web uygulamaları geliştirirken XML ve JavaScript'in yanı sıra dahil edilen diller için desteklenir. Otomatik tamamlama önerileri, kod düzenleyici penceresinin üzerinde, düzenleme imlecinin yakınında, modelsiz bir liste kutusunda görünür. Visual Studio 2008'den itibaren, kodun engellediğini görmek için geçici olarak yarı saydam yapılabilir. Kod düzenleyici, desteklenen tüm diller için kullanılır.

**Visual Studio Code Editor**, hızlı gezinme için koddaki yer imlerinin ayarlanmasını da destekler. Gezinme yardımcıları, metin arama ve ifade aramasına ek olarak, hata veren kod bloklarını ve artımlı aramayı içerir. Kod düzenleyici ayrıca çok öğeli bir pano ve bir görev listesi içerir. Kod düzenleyici, tekrarlanan kodlar için kaydedilmiş şablonlar olan ve koda eklenebilen ve üzerinde çalışılan proje için özelleştirilebilen kod parçacıklarını destekler.



**Hata ayıklayıcı**, hem kaynak düzeyinde hata ayıklayıcı hem de makine düzeyinde hata ayıklayıcı olarak çalışan bir hata ayıklayıcı içerir. Visual Studio tarafından desteklenen herhangi bir dilde yazılmış uygulamalarda hata ayıklamak için kullanılabilir. Ayrıca, çalışan süreçlere eklenebilir, bu süreçleri izleyebilir ve hata ayıklayabilir. Visual Studio Hata Ayıklayıcı, break point kullanımını sunar. Hata ayıklayıcı, düzenleme ve devam et özelliklerine izin verir. Hata ayıklanırken kodun düzenlenmesine izin verir.

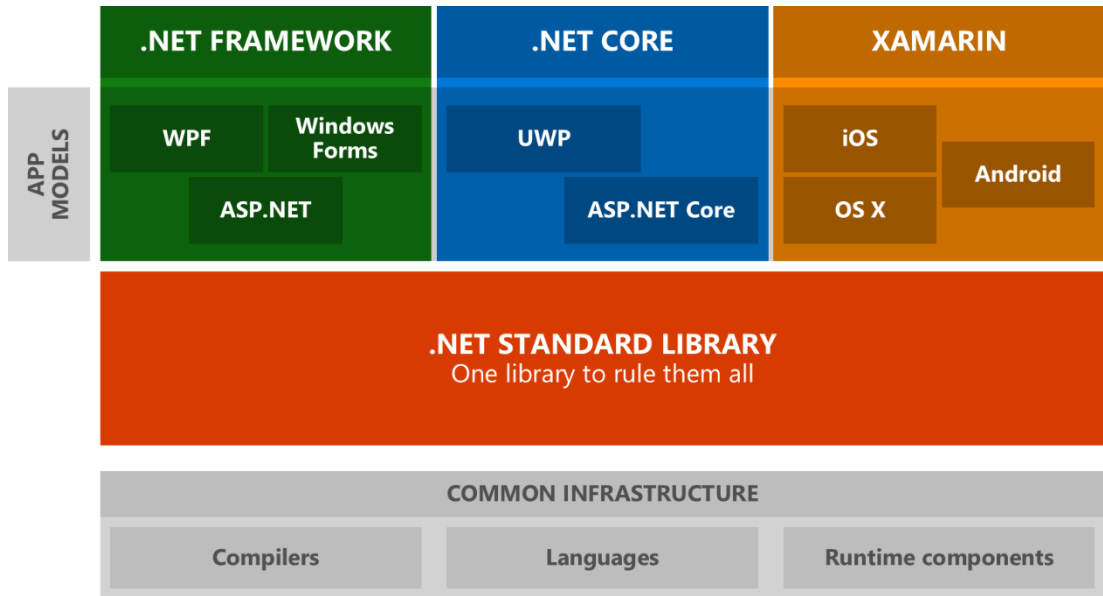
Aşağıdaki gibi Microsoft Visual Studio'nun 3 sürümü vardır:

1. **Topluluk:** 2014 yılında duyurulan ücretsiz bir sürümdür. Diğer tüm sürümler ücretlidir. Professional sürümüne benzer özellikleri içerir. Bu sürümü kullanarak, herhangi bir bireysel geliştirici, .Net uygulamaları, Web uygulamaları ve daha fazlası gibi kendi ücretsiz veya ücretli uygulamalarını geliştirebilir. Kurumsal bir kuruluşta bu sürümün bazı sınırlamaları vardır. Örneğin, kuruluşunuzun 250'den fazla PC'si varsa ve yıllık geliri 1 Milyon dolardan fazlaysa, bu sürümü kullanmanıza izin verilmez. Kurumsal olmayan bir kuruluşta, bu sürümü en fazla beş kullanıcı kullanabilir.
2. **Professional:** Visual Studio'nun ticari sürümüdür. Visual Studio 2010 ve sonraki sürümlerinde gelir. XML ve XSLT düzenleme desteği sağlar ve Server Explorer gibi araçları ve Microsoft SQL Server ile entegrasyon içerir. Microsoft, bu sürümün ücretsiz deneme sürümünü sağlar ve deneme süresinden sonra kullanmaya devam etmek için kullanıcının ödeme yapması gerekir.
3. **Enterprise:** Microsoft, bu sürümün 90 günlük ücretsiz deneme sürümünü sağlar ve deneme süresinden sonra kullanıcının kullanmaya devam etmek için ödeme yapması gerekir. Bu sürümün ana avantajı, yüksek düzeyde ölçeklenebilir olması ve yüksek kaliteli yazılım sunmasıdır.

## B. .NET Framework

.Net Framework, Windows uygulamaları oluşturmak ve çalıştırmak için Microsoft tarafından geliştirilmiş bir yazılım geliştirme platformudur. .Net çerçevesi, masaüstü ve web uygulamaları oluşturmak için geliştirici araçları, programlama dilleri ve kitaplıklardan oluşur. Ayrıca web siteleri, web hizmetleri ve oyunlar oluşturmak için kullanılır. .Net Framework, Windows Platformunda çalışacak uygulamalar yaratmayı amaçlıyordu. .Net çerçevesinin ilk sürümü 2002 yılında yayınlandı. Microsoft .Net Framework hem Form tabanlı hem de Web tabanlı uygulamalar oluşturmak için kullanılabilir. Web servisleri .Net Framework kullanılarak da geliştirilebilir.

Framework ayrıca Visual Basic ve C# gibi çeşitli programlama dillerini de destekler. Böylece geliştiriciler, gerekli uygulamayı geliştirmek için dili seçip seçebilir. .NET Framework için yazılan programlar, Common Language Runtime (CLR) adlı bir yazılım ortamında yürütülür. CLR; güvenlik, bellek yönetimi ve özel durum işleme gibi hizmetler sağlayan bir uygulama sanal makinesidir. Bu nedenle .NET Framework kullanılarak yazılan bilgisayar koduna "yönetilen kod" denir. FCL ve CLR birlikte .NET Framework'ü oluşturur.



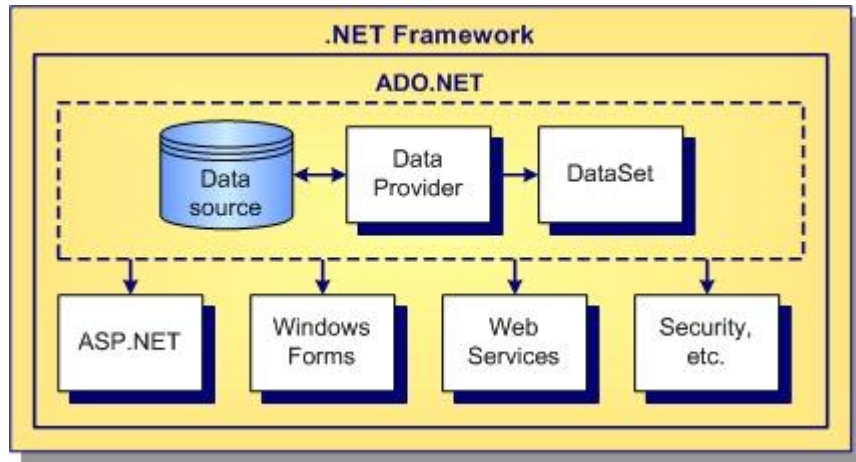
Şekil 1 . NET Framework

FCL, kullanıcı arayüzü, veri erişimi, veritabanı bağlantısı, kriptografi, web uygulaması geliştirme, sayısal algoritmalar ve ağ iletişimi sağlar. Programcılar kaynak kodlarını .NET Framework ve diğer kütüphanelerle birleştirerek yazılım üretirler.

Framework'ün, Windows platformu için oluşturulan çoğu yeni uygulama tarafından kullanılması amaçlanmıştır. Microsoft ayrıca Visual Studio adlı .NET yazılımı için tümleşik bir geliştirme ortamı da üretir. Nisan 2019'da Microsoft, çerçevenin son sürümü olan .NET Framework 4.8'i tescilli bir teklif olarak yayınladı. O zamandan beri yalnızca bu sürüme yönelik aylık güvenlik ve güvenilirlik hata düzeltmeleri yayınlandı. Bu sürümde başka bir değişiklik planlanmamaktadır.

## C. ADO .NET

ADO.NET, ilişkisel ve ilişkisel olmayan sistemler arasında iletişim sağlayan Microsoft .NET Framework'ten bir veri erişim teknolojisidir. ADO.NET, programcıların bir veri tabanından verilere ve veri hizmetlerine erişmek için kullanabilecekleri bir dizi bilgisayar yazılımı bileşenidir. Microsoft .NET Framework'te bulunan temel sınıf kitaplığının bir parçasıdır. İlişkisel olmayan veri kaynaklarındaki verilere de erişebilmesine rağmen, programcılar tarafından ilişkisel veri tabanı sistemlerinde depolanan verilere erişmek ve bunları değiştirmek için yaygın olarak kullanılır. ADO.NET bazen ActiveX Data Objects (ADO) teknolojisinin bir evrimi olarak kabul edilir, ancak o kadar kapsamlı bir şekilde değiştirildi ki tamamen yeni bir ürün olarak kabul edilebilir.



Şekil 2: ADO.NET Mimarisi

### ADO.NET Bileşenleri

ADO.NET'in verilere erişmek ve bunları işlemek için kullanılan iki ana bileşeni, .NET Framework Data Providers ve DataSet'tir.

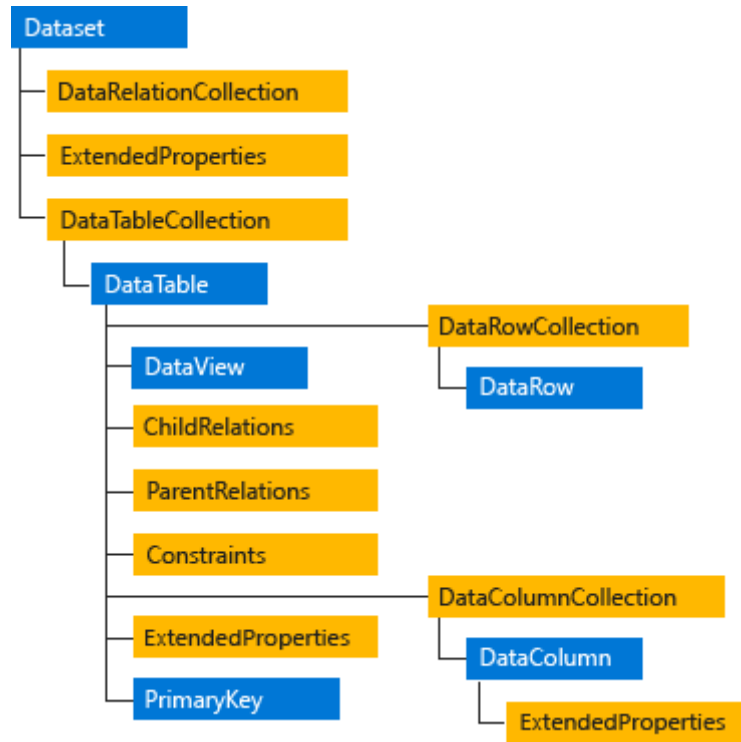
- .NET Framework Data Providers

.NET Framework Data Providers, veri işleme ve verilere hızlı, salt okunur erişim için özel olarak tasarlanmış bileşenlerdir. Connection nesnesi, bir veri kaynağına bağlantı sağlar. Command nesnesi, verileri döndürmek, verileri değiştirmek ve parametre bilgilerini göndermek veya almak için veri tabanı komutlarına erişim sağlar. DataReader, veri kaynağından yüksek performanslı bir veri akışı sağlar. Son olarak, DataAdapter, DataSet nesnesi ile veri kaynağı arasındaki köprüyü sağlar. DataAdapter, hem DataSet'i verilerle yüklemek hem de DataSet'teki verilerde yapılan

değişiklikleri veri kaynağına geri döndürmek için veri kaynağında SQL komutlarını yürütmek için Command nesnelerini kullanır.

- DataSet

ADO.NET DataSet, herhangi bir veri kaynağından bağımsız olarak veri erişimi için açıkça tasarlanmıştır. Sonuç olarak, birden fazla ve farklı veri kaynağıyla kullanılabilir, XML verileriyle kullanılabilir veya uygulamaya yerel verileri yönetmek için kullanılabilir. DataSet, veri satırlarından ve sütunlarından oluşan bir veya daha fazla DataTable nesnesi koleksiyonunu ve ayrıca DataTable nesnelerindeki verilerle ilgili birincil anahtar, yabancı anahtar, kısıtlama ve ilişki bilgilerini içerir.



Şekil 3: ADO.NET Bileşenleri

## D. Windows Forms

Windows Forms (WinForms), Microsoft .NET, .NET Framework veya Mono Framework'tün bir parçası olarak dahil edilen ücretsiz ve açık kaynaklı bir grafik (GUI) sınıf kitaplığıdır ve masaüstü, dizüstü bilgisayar ve tablet PC'ler için istemci uygulamaları yazmak için bir platform sağlar. Daha önceki ve daha karmaşık olan C++ tabanlı Microsoft Foundation Class Library'nin yerine geçtiği görülse de karşılaştırılabilir bir paradigma sunmaz ve yalnızca çok katmanlı bir çözümde kullanıcı arabirimi katmanı için bir platform görevi görür.

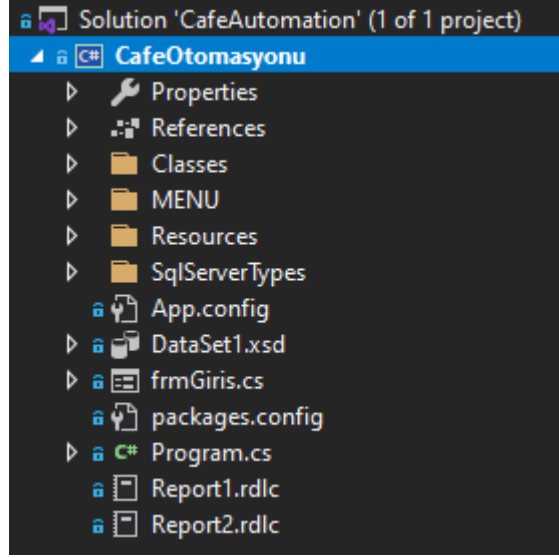
4 Aralık 2018'deki Microsoft Connect etkinliğinde Microsoft, Windows Forms'u GitHub'da açık kaynaklı bir proje olarak yayınladığını duyurdu

### Windows Forms Componentleri

Bu projede kullanılan Windows forms bileşenleri aşağıda açıklamalarıyla birlikte verilmiştir.

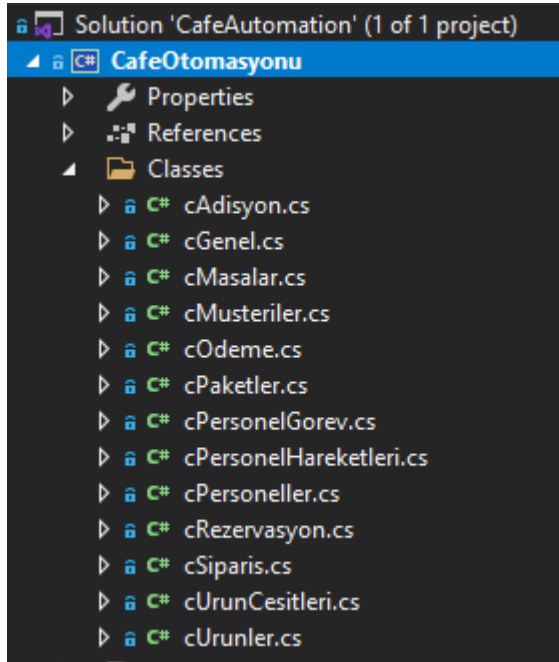
- **Button:** Butonlar tıklama işlemiyle yeni form açmak, işlemleri gerçekleştirmek amacıyla kullanılmıştır.
- **Label:** Bileşenlere isim vermek, numaraları tutmak için kullanılmıştır.
- **TextBox:** Kullanıcıdan bilgi almak için kullanılmıştır.
- **CheckBox:** Bilgileri checkbox'a atıp seçmek için kullanılmıştır.
- **ListView:** Veri tabanındaki bilgileri yansıtmak için kullanılmıştır.
- **GroupBox:** Menüleri gruplara bölmek için kullanılmıştır.
- **DataGridView:** Raporlama yapmak için kullanılmıştır.
- **DateTimePicker:** Tarih eklemek için kullanılmıştır.
- **DataSet:** Verileri düzenlemek için kullanılmıştır.
- **ReportViewer:** Raporlama çizelgesi için kullanılmıştır.

## II. PROJE KLASÖR YAPISI



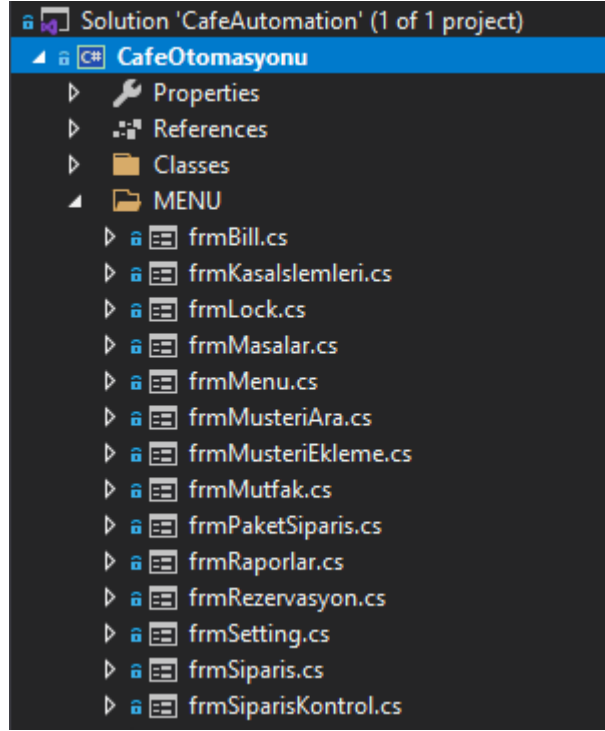
Şekil 4: Cafe Otomasyonu Proje Klasör Yapısı

Cafe Otomasyonu projesi çözüm dosyası içerisinde Classes, MENU, Resources, SqlServerTypes klasörleri bulunmaktadır. DataSet1, Report1 ve Report2 dosyaları; raporlar modülünde kullanılmaktadır.



Şekil 5: Cafe Otomasyonu Class Yapısı

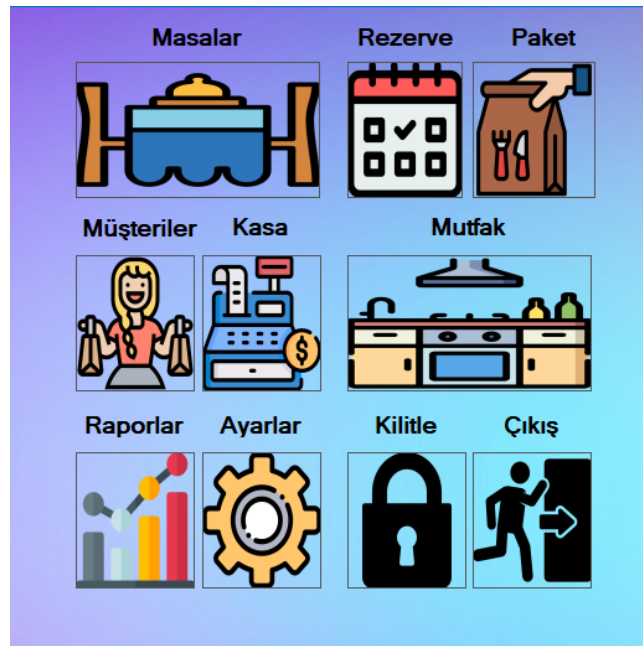
Cafe Otomasyonu projesinde metotların yazıldığı tüm classlar SQL veri tabanına göre ayrı ayrı sınıflandırılmıştır. Tüm class yapısı “Classes” klasörü içerisinde yer almaktadır. Bir tanesi SQL bağlantısı ve genel propertylerin bulunduğu genel class olmakla birlikte toplamda 13 adet classımız bulunmaktadır.



Şekil 6: Cafe Otomasyonu Form Yapısı

Otomasyon projesindeki modüllerin bulunduğu formlar MENU klasörü içerisinde bulunmaktadır. Toplamda bir tanesi kullanıcı giriş ekranı olmakla birlikte 15 adet form yapısı vardır.

Giriş formu frmGiris olarak adlandırılmış olup tanımlanan kullanıcı ve şifreleriyle giriş yapmayı sağlamaktadır. Giriş yapıldıktan sonra frmMenu olarak adlandırdığımız ve tüm formlara bu ekrandan ulaştığımız menü ekranı açılmaktadır.



Şekil 7: Menü Formu



### III. MODÜLLER VE MODÜLLERİN AÇIKLANMASI

#### A. Genel Class

```
91 references
public class cGenel
{
    public string conString = ("Server=DESKTOP-FLOADJ7\\SQLEXPRESS;Initial Catalog=CafeOtomasyonu;Integrated Security=True");
    public static int _personelId;
    public static int _gorevId;
    public static int _musteriEkleme;
    public static string _buttonValue;
    public static int _musteriId;
    public static string _buttonName;
    public static int _servisTurNo;
    public static string _adisyonId;
}
```

Şekil 8: Genel Classı Fields

Genel Classı otomasyon projesinin ana class yapısını oluşturmaktadır. Bu class içerisinde Microsoft SQL Server bağlantısı bulunmaktadır. Genel classının oluşturulma amacı veri tabanına ulaşmak istenildiğinde sürekli olarak aynı kodları yazmamak ve kod tekrarını önlemektir.

## B. Adisyon Class

```
class cAdisyon
{
    cGenel gnl = new cGenel();

    #region Fields
    private int _ID;
    private int _ServisTurNo;
    private decimal _Tutar;
    private DateTime _Tarih;
    private int _PersonelId;
    private int _Durum;
    private int _MasaId;
    #endregion

    #region Properties
    0 references
    public int ID { get => _ID; set => _ID = value; }
    4 references
    public int ServisTurNo { get => _ServisTurNo; set => _ServisTurNo = value; }
    0 references
    public decimal Tutar { get => _Tutar; set => _Tutar = value; }
    4 references
    public DateTime Tarih { get => _Tarih; set => _Tarih = value; }
    4 references
    public int PersonelId { get => _PersonelId; set => _PersonelId = value; }
    0 references
    public int Durum { get => _Durum; set => _Durum = value; }
    4 references
    public int MasaId { get => _MasaId; set => _MasaId = value; }
    #endregion
}
```

Şekil 9: Adisyon Class Fields ve Properties

Adisyon class yapısı adisyona ait ID, servis türü, adisyon tutarı, adisyon tarihi, işlemi yapan personelin ID'sini, adisyonun durumunu ve adisyonun hangi masaya ait olduğu gibi bilgileri tutmaktadır. Bu tanımlanan bu bilgilerin kullanılabilmesi için aynı zamanda propertylerini de tanımlamak gerekmektedir. Şekil 6'da görüldüğü gibi tüm veri tanımlayıcıları adisyon classının içerisinde belirtilmiştir.

### 1. Adisyon Class Metotları

```
1 reference
public bool setByAdditionNew(cAdisyon Bilgiler)...

4 references
public int getByAddition(int MasaId)...

2 references
public void adisyonkapat(int adisyonID, int durum)...

1 reference
public int paketAdisyonIdbulAdedi()...

1 reference
public void acikPaketAdisyonlar(ListView lv)...

2 references
public int musterininsonadisyonId(int musterId)...

1 reference
public void musteridetaylar(ListView lv, int musterId)...

1 reference
public int RezervasyonAdisyonAc(cAdisyon bilgiler)...
```

Şekil 10: Adisyon Class Metotları

Adisyon classının içerisinde toplamda 8 adet metot bulunmaktadır. Bu metotlar müşterilerin adisyon bilgilerinin getirilmesi, masalara göre adisyonların tutulması,

ödeme gerçekleştiğinde adisyonların kapatılması, rezervasyona istinaden adisyon açılması, müşterilerin hangi adisyonları olduğunun incelenmesi ve rapor çekilmesi gibi işlemleri yapmaktadır.

## 2. SetByAdditionNew Metodu

```
public bool setByAdditionNew(cAdisyon Bilgiler)
{
    bool sonuc = false;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Insert into ADISYON(SERVISTURNO,TARIH,PERSONELID,MASAIID,DURUM) values (@ServisTurNo,@Tarih,@PersonelID,@MasaId,@Durum)", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@ServisTurNo", SqlDbType.Int).Value = Bilgiler.ServisTurNo;
        cmd.Parameters.Add("@Tarih", SqlDbType.DateTime).Value = Bilgiler.Tarih;
        cmd.Parameters.Add("@PersonelID", SqlDbType.Int).Value = Bilgiler.PersonelID;
        cmd.Parameters.Add("@MasaId", SqlDbType.Int).Value = Bilgiler.MasaId;
        cmd.Parameters.Add("@Durum", SqlDbType.Bit).Value = 0;
        sonuc = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 11: Yeni Adisyon Metodu

Yeni adisyon açılması için kullandığımız bu metod, masa içerisinden ürünlerin eklenmesi ve sipariş butonun tıklanmasıyla aktif hale gelmektedir. Yeni adisyon açarken servis türünü, tarihi, personeli, masa numarasını ve adisyonun durumu verilerini tutmaktayız. Veri tabanımıza ise aldığımız tüm bu bilgileri otomatik bir numara atayarak adisyon numarası olarak tutmaktayız.

## 3. GetbyAddition Metodu

```
4 references
public int getbyAddition(int MasaId)
{
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select top 1 ID from ADISYON where MASAIID=@MasaId order by ID desc", con);
    //parametreyi gönderiyoruz
    cmd.Parameters.Add("MasaId", SqlDbType.Int).Value = MasaId;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        MasaId = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Close();
    }
    return MasaId;
}
```

Şekil 12: Adisyon Seçme Metodu

Hali hazırda açılmış olan adisyonu seçmek için kullandığımız metottur. Bu metotta sipariş alınıp adisyonu açılmış siparişlere tekrardan sipariş girerken yeni

adisyon açılmaması için kullanılmaktadır. Dışarıdan gönderdiğimiz masa numarasına göre adisyon tablosundaki adisyon numarasını seçmektedir.

#### 4. Adisyon Kapat

```
2 references
public void AdisyonKapat(int adisyonID, int durum)
{
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("update ADISYON set DURUM=@durum where ID=@adisyonId", con);

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@adisyonId", SqlDbType.Int).Value = adisyonID;
        cmd.Parameters.Add("@durum", SqlDbType.Int).Value = durum;
        cmd.ExecuteNonQuery();
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
}
```

Şekil 13: Adisyon Kapatma Metodu

Açık olan adisyonun kapatılmasını sağlayan metottur. Bu metotta adisyonun numarası ve durumunu veri tabanındaki adisyon tablosundan güncellemektedir. Parametre olarak adisyonID ve durum verilerini almaktadır.

#### 5. ID'ye Göre Adisyon Bul Metodu

```
1 reference
public int PaketAdisyonIdBulAdedi()
{
    int miktar = 0;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select count(*) Sayi from ADISYON where (DURUM=0) and (SERVISTURNO=2)", con);

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        miktar = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    return miktar;
}
```

Şekil 14: ID'ye Göre Adisyon Bul Metodu

Servis türü olarak veri tabanımızda masa servisi ve paket servisi olarak iki tür tutmaktayız. Bu metot paket sipariş servisi için adisyon bulmamızı sağlamaktadır.

## 6. Açık Paket Sipariş Adisyon Metodu

```
public void açıkPaketAdisyonlar(listview lv)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select PAKETSIPARIS.MUSTERIID,MUSTERILER.AD+ ' '+ MUSTERILER.SOYAD as Musteri, ADISYON.ID as AdisyonID FROM PAKETSIPARIS inner join MUSTERILER on MUSTERILER.ID=PAKETSIPARIS.MUSTERIID inner join ADISYON on ADISYON.ID=PAKETSIPARIS.ADISYONID where ADISYON.DURUM=0", con);

    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();

        int sayac = 0;

        while (dr.Read())
        {
            lv.Items.Add(dr["MUSTERIID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["Musteri"].ToString());
            lv.Items[sayac].SubItems.Add(dr["AdisyonID"].ToString());

            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 15: Açık Paket Sipariş Adisyon Metodu

Açık paket siparişler için adisyonu bulan metottur. Parametre olarak dışarıdan müşteri numarası, müşteri adı ve adisyon numarasını almaktadır. SQL veri tabanında paket sipariş, müşteriler ve adisyon tablolarının ilişkilendirilmesi ile bulunmaktadır.

## 7. Müşterinin Son Adisyon ID'sini Getirme Metodu

```
2 references
public int musterininsonadisyonId(int musterId)
{
    int sonuc = 0;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select ADISYON.ID from ADISYON inner join PAKETSIPARIS on PAKETSIPARIS.ADISYONID=ADISYON.ID where PAKETSIPARIS.DURUM=0 and ADISYON.DURUM=0 and PAKETSIPARIS.MUSTERIID=@musterId", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@musterId", SqlDbType.Int).value = musterId;
        sonuc = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 16:Müşterinin Son Adisyon ID'sini Getirme Metodu

Siparişleri kontrol ettiğimiz formda müşterinin en son adisyonunu seçen metottur. Adisyon tablosu ile paket sipariş tablosunu birleştirip müşteri numarasına göre adisyonunu getirir.

## 8. Müşterinin Adisyon Detayları Metodu

```
1 reference
public void müşteriDetaylar(ListView lv, int müşteriId)
{
    lv.Items.Clear();

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select PAKETSIPARIS.MUSTERIID,PAKETSIPARIS.ADISYONID,MUSTERILER.AD,MUSTERILER.SOYAD,CONVERT(varchar(10),ADISYON.TARİH,104) as tarih from ADISYON inner join
    PAKETSIPARIS on PAKETSIPARIS.ADISYONID=ADISYON.ID inner join MUSTERILER on MUSTERILER.ID=PAKETSIPARIS.MUSTERIID where ADISYON.SERVISTURNO=2 and PAKETSIPARIS.MUSTERIID=@musteriId", con);

    cmd.Parameters.Add("@musteriId", SqlDbType.Int).Value = müşteriId;
    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        int sayac = 0;
        dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            lv.Items.Add(dr["MUSTERIID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["AD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["SOYAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["tarih"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ADISYONID"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
}
```

Şekil 17: Müşterinin Adisyon Detayları Metodu

Müşterinin ad, soyad, tarih ve adisyon numarasına göre paket sipariş adisyon detaylarını getiren metottur. Aldığı bilgileri listview'e atar ve müşteri numarasına göre işlem yapar.

## 9. Rezervasyona İstinaden Adisyon Aç Metodu

```
1 reference
public int RezervasyonAdisyonAc(CAdisyon bilgiler)
{
    int sonuc = 0;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Insert into ADISYON(SERVISTURNO,TARİH,PERSONELID,MASAIID) values (@ServisTurNo,@Tarih,@PersonelID,@MasaId); select scope_IDENTITY()", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        cmd.Parameters.Add("@ServisTurNo", SqlDbType.Int).Value = bilgiler.ServisTurNo;
        cmd.Parameters.Add("@Tarih", SqlDbType.Datetime).Value = bilgiler.Tarih;
        cmd.Parameters.Add("@PersonelID", SqlDbType.Int).Value = bilgiler.PersonelID;
        cmd.Parameters.Add("@MasaId", SqlDbType.Int).Value = bilgiler.MasaId;

        sonuc = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 18: Rezervasyona İstinaden Adisyon Aç Metodu

Form ekranımızdaki rezervasyon modülünden açılan rezervasyona istinaden adisyon açan metottur. Adisyon tablomuza servis türünü, tarihini, personel numarasını ve rezervasyon numarasını tutar.

## C. Masalar Classı

```
18 references
class cMasalar
{
    #region Fields
    private int _ID;
    private int _KAPASITE;
    private int _SERVISTURU;
    private int _DURUM;
    private int _ONAY;
    private string _MASABILGI;
    #endregion

    #region Properties

    public int ID { get => _ID; set => _ID = value; }
    1 reference
    public int KAPASITE { get => _KAPASITE; set => _KAPASITE = value; }
    0 references
    public int SERVISTURU { get => _SERVISTURU; set => _SERVISTURU = value; }
    0 references
    public int DURUM { get => _DURUM; set => _DURUM = value; }
    0 references
    public int ONAY { get => _ONAY; set => _ONAY = value; }
    1 reference
    public string MASABILGI { get => _MASABILGI; set => _MASABILGI = value; }
    #endregion
}
```

Şekil 19: Masalar Class Fields ve Properties

Masalar classı içerisinde masa numaraları, kapasiteleri, servis türü, durumu, masa bilgisi gibi bilgiler bulunmaktadır.

### 1. Masalar Class Metotları

```
18 references
class cMasalar
{
    #region Fields
    #endregion

    #region Properties
    #endregion

    cGenel gen1 = new cGenel();
    private string masaNo;
    2 references
    public string SessionSum(int state)...
    2 references
    public int TableGetbyNumber(string TableValue)...
    8 references
    public bool TableGetbyState(int ButtonName, int state)...
    4 references
    public void setChangeTableState(string ButonName, int state)...
    1 reference
    public void MasaKapasitesiveDurumuGetir(ComboBox cm)...
    0 references
    public override string ToString()...
}
```

Şekil 20: Masalar Class Metotları

Bu class içerisinde beş adet metot bulunmaktadır. Masaların açık kalma bilgisini gösteren metot, masa numarasını ve durumunu gösteren metot, masa durumunu

değiştiren metod ve masa kapasitesi ile durumunu combobox içerisine getiren metotlar vardır.

## 2. Sessionsum Metodu

```
cGenel gnl = new cGenel();
private string masaNo;
2 references
public string Sessionsum(int state)
{
    string dt = "";
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select TARIH,MASAIID from ADISYON Join MASALAR on ADISYON.MASAIID=MASALAR.ID where MASALAR.DURUM =" + state + " and ADISYON.Durum = 0", con);

    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            dt = Convert.ToDateTime(dr["TARIH"]).ToString("yyyy-MM-dd HH:mm:ss");
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
    return dt;
}
```

Şekil 21: Masanın Açık Kalma Süresi Metodu

Masanın dolu durumuna geçmesi ile müşterilerin masada kalma süresini açıklayan metottur. Tarih parametresi gönderilerek yapılır.

## 3. TableGetbyNumber Metodu

```
2 references
public int TableGetbyNumber(string TableValue)
{
    string aa = TableValue;
    int length = aa.Length;
    if (length > 8)
    {
        return Convert.ToInt32(aa.Substring(length - 2, 2));
    }
    else
    {
        return Convert.ToInt32(aa.Substring(length - 1, 1));
    }
}
```

Şekil 22: TableGetbyNumber Metodu

Masanın numarasını döndüren metottur. On sayısından küçük masa numaraları için ilgili if bloğunu döndürür, masa numarası iki haneli olduğunda else bloğunu döndürür.



#### 4. Masa Durumunu Getir Metodu

```
8 references
public bool TableGetbyState(int ButtonName, int state)
{
    bool result = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select DURUM from MASALAR where ID=@TableId and DURUM=@state", con);

    cmd.Parameters.Add("@TableId", SqlDbType.Int).Value = ButtonName;
    cmd.Parameters.Add("@state", SqlDbType.Int).Value = state;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        result = Convert.ToBoolean(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return result;
}
```

Şekil 23: TableGetbyState Metodu

Masanın dolu, boş, rezerve, açık rezerve olma durumlarını seçer. Parametre olarak masa numarası ve durumu gönderilir. İlgili tablodan masanın durumunu seçer.

#### 5. Masa Durumunu Değiştir Metodu

```
4 references
public void setChangeTableState(string ButonName, int state)//masa durumunu değiştir
{
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("update MASALAR set DURUM=@Durum where ID=@MasaNo", con);
    string masaNo = "";

    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }
    string aa = ButonName;
    int uzunluk = aa.Length;

    cmd.Parameters.Add("@Durum", SqlDbType.Int).Value = state;
    cmd.Parameters.Add("@MasaNo", SqlDbType.Int).Value = aa.Substring(uzunluk - 1, 1);

    if (uzunluk > 8)
    {
        masaNo = aa.Substring(uzunluk - 2, 2);
    }
    else
    {
        masaNo = aa.Substring(uzunluk - 2, 1);
    }
    cmd.ExecuteNonQuery();
    con.Dispose();
    con.Close();
}
```

Şekil 24: setChangeTableState Metodu

Masanın durumunu dört durum içerisinde yeniden değiştirmek için kullanılır. Parametre olarak masanın durumu ve masa numarası gönderilir. Masanın numarası tek haneli ise else bloğu, çift haneli ise if bloğu çalıştırılır.

## 6. Masa Kapasitesi ve Durumu Getir Metodu

```
public void MasaKapasitesiveDurumuGetir(ComboBox cm)
{
    cm.Items.Clear();
    string durum = "";

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from masalar", con);

    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }

    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        cMasalar c = new cMasalar();
        if (c._DURUM == 2)
            durum = "DOLU";
        else if (c._DURUM == 3)
            durum = "REZERVE";
        c._KAPASITE = Convert.ToInt32(dr["KAPASITE"]);
        c._MASABILGI = "MASA NO" + dr["ID"].ToString() + "KAPASİTESİ : " + dr["KAPASITE"].ToString();
        c._ID = Convert.ToInt32(dr["ID"]);
        cm.Items.Add(c);
    }
    dr.Close();
    con.Dispose();
    con.Close();
}
```

Şekil 25: MasaKapasitesiveDurumuGetir Metodu

Masanın kapasitesini ve masa durumunu getiren metottur. İlgili verileri masalar tablosu içerisinde seçer. Masa durumu ikiye eşit ise “dolmuş”, üçe eşit ise “rezerve” durumlarını döndürür.

## D. Müşteriler Class

```
20 references
class cMusteriler
{
    cGenel gnl = new cGenel();
    #region Fields
    private int _musteriid;
    private string _musteriad;
    private string _musterisoyad;
    private string _telefon;
    private string _adres;
    private string _email;
    #endregion
    #region Properties
    2 references
    public int Musteriid { get => _musteriid; set => _musteriid = value; }
    2 references
    public string Musteriad { get => _musteriad; set => _musteriad = value; }
    2 references
    public string Musterisoyad { get => _musterisoyad; set => _musterisoyad = value; }
    2 references
    public string Telefon { get => _telefon; set => _telefon = value; }
    2 references
    public string Adres { get => _adres; set => _adres = value; }
    2 references
    public string Email { get => _email; set => _email = value; }
    #endregion
}
```

Şekil 26: Müşteriler Class Fields ve Properties

Müşteriler classı içerisinde müşteriye ait bilgiler yer almaktadır. Bu bilgiler müşteri numarası, müşteri adı, soyadı, telefonu, adresi ve e-mail adresidir.

### 1. Müşteriler Class Metotları

```
20 references
class cMusteriler
{
    cGenel gnl = new cGenel();
    Fields
    Properties

    1 reference
    public bool MusteriVarmi(string tlf)...

    1 reference
    public int musteriekle(cMusteriler m)...

    1 reference
    public bool musteribilgileriGuncelle(cMusteriler m)...

    2 references
    public void musterileriGetir(ListView lv)...

    1 reference
    public void musterigetirID(int musterIID, TextBox ad, TextBox soyad, TextBox tlf, TextBox adres, TextBox email)...

    3 references
    public void musterigetirAd(ListView lv, string musteriad)...

    0 references
    public void musterigetirSoyad(ListView lv, string musterisoyad)...

    1 reference
    public void musterigetirTlf(ListView lv, string tlf)...

}
```

Şekil 27: Müşteriler Class Metotları

Bu class içerisinde yedi adet metot bulunmaktadır. Müşteri ekle metodu, güncelle metodu, müşterileri listview içerisine getirme metodu, müşteri numarasına göre arama metodu, soyadına göre ve telefonuna göre arama metotları bulunmaktadır.

## 2. Müşteri Var Mı Metodu

```
1 reference
public bool MusteriVarmi(string tlf)
{
    bool sonuc = false;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand();

    cmd.Connection = con;
    cmd.CommandText = "MusteriVarmi";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add("@telefon", SqlDbType.VarChar).Value = tlf;
    cmd.Parameters.Add("@sonuc", SqlDbType.Int);
    cmd.Parameters["@sonuc"].Direction = ParameterDirection.Output;

    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }
    try
    {
        cmd.ExecuteNonQuery();
        sonuc = Convert.ToBoolean(cmd.Parameters["@sonuc"].Value);
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    return sonuc;
}
```

Şekil 28: MusteriVarmi Metodu

Dışarıdan parametre olarak gönderdiğimi telefon string ifadesine istinaden müşteri olup olmadığını kontrol eden metottur. Geriye sonuç olarak değer döndürür.

## 3. Müşteri Ekleme Metodu

```
1 reference
public int musteriekle(cMusteriler m)
{
    int sonuc = 0;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("insert into MUSTERILER (AD,SOYAD,TELEFON,ADRES,EMAIL) values (@ad,@soyad,@telefon,@adres,@email); select SCOPE_IDENTITY()", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        cmd.Parameters.Add("@ad", SqlDbType.VarChar).Value = m._musteriad;
        cmd.Parameters.Add("@soyad", SqlDbType.VarChar).Value = m._musterisoyad;
        cmd.Parameters.Add("@telefon", SqlDbType.VarChar).Value = m._telefon;
        cmd.Parameters.Add("@adres", SqlDbType.VarChar).Value = m._adres;
        cmd.Parameters.Add("@email", SqlDbType.VarChar).Value = m._email;
        sonuc = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 29: Müşteri Ekleme Metodu

Müşteriler classını göndererek veri tabanına yeni müşteri eklemek için kullandığımız metottur. Kullanıcıdan; ad, soyad, telefon, adres ve email adresi bilgilerini alırız.

#### 4. Müşteri Bilgilerini Güncelleme Metodu

```
1 reference
public bool musteribilgileriguncelle(cMusteriler m)
{
    bool sonuc = false;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Update MUSTERILER set AD=@ad,SOYAD=@soyad,TELEFON=@telefon,ADRES=@adres,EMAIL=@email where ID=@musteriid", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        cmd.Parameters.Add("@ad", SqlDbType.VarChar).Value = m._musteriad;
        cmd.Parameters.Add("@soyad", SqlDbType.VarChar).Value = m._musterisoyad;
        cmd.Parameters.Add("@telefon", SqlDbType.VarChar).Value = m._telefon;
        cmd.Parameters.Add("@adres", SqlDbType.VarChar).Value = m._adres;
        cmd.Parameters.Add("@email", SqlDbType.VarChar).Value = m._email;
        cmd.Parameters.Add("@musteriid", SqlDbType.VarChar).Value = m.Musteriid;
        sonuc = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 30: Müşteri Bilgilerini Güncelleme Metodu

Müşteri bilgilerinde güncelleme yapacağımız zaman çağırdığımız metottur. Müşterinin güncelleyeceğimiz bilgilerini update komutu ile veri tabanından güncelleriz. Dışarıdan aldığımız parametreler; ad,soyad, telefon, adres, email ve müşteri numarasıdır.

#### 5. Müşterileri ListView'e Getirme Metodu

```
2 references
public void musterilerigetir(ListView lv)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select*from MUSTERILER", con);
    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["ID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["AD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["SOYAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["TELEFON"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ADRES"].ToString());
            lv.Items[sayac].SubItems.Add(dr["EMAIL"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 31: Müşterileri ListView'e Getirme Metodu

Herhangi bir listview içerisine müşteriler atadığımız metottur. Veri tabanından müşteriler tablosundan tüm verileri çekeriz ve listview içerisine atarız.

## 6. Müşterileri ID'ye Göre Getirme Metodu

```
1 reference
public void musterigetir(int musterIID, TextBox ad, TextBox soyad, TextBox tlf, TextBox adres, TextBox email)
{
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from MUSTERILER where ID=@musterIID", con);
    SqlDataReader dr = null;
    cmd.Parameters.Add("@musterIID", SqlDbType.Int).Value = musterIID;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            ad.Text = dr["AD"].ToString();
            soyad.Text = dr["SOYAD"].ToString();
            tlf.Text = dr["TELEFON"].ToString();
            adres.Text = dr["ADRES"].ToString();
            email.Text = dr["EMAIL"].ToString();
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 32: Müşterileri ID'ye Göre Getirme Metodu

Dışarıdan müşteri numarası, ad, soyad, telefon, adres ve email parametrelerini alırız. Müşteriler tablosundan müşteri numarasını eşleyip ilgili müşteriye çağırabildiğimiz metottur.

## 7. Müşterileri İsme Göre Listeleme Metodu

```
3 references
public void musterigetirAd(ListView lv, string musteriad)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from MUSTERILER where AD like @musteriad + '%'", con);

    SqlDataReader dr = null;
    cmd.Parameters.Add("@musteriad", SqlDbType.VarChar).Value = musteriad;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;

        while (dr.Read())
        {
            lv.Items.Add(Convert.ToInt32(dr["ID"]).ToString());
            lv.Items[sayac].SubItems.Add(dr["AD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["SOYAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["TELEFON"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ADRES"].ToString());
            lv.Items[sayac].SubItems.Add(dr["EMAIL"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 33: Müşterileri İsme Göre Listeleme Metodu

Müşteriler içerisinde ada göre arama yaparken kullandığımız metottur. Müşteriler bir Listview'e atarız ve müşteri adına göre arama yapabiliriz. Müşteriye ait tüm özellikleri burada tutarız.

## 8. Müşterileri Soyada Göre Listeleme Metodu

```
public void musterigetirSoyad(ListView lv, string musterisoyad)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from MUSTERILER where SOYAD like @musterisoyad + '%'", con);
    SqlDataReader dr = null;
    cmd.Parameters.Add("@musterisoyad", SqlDbType.VarChar).Value = musterisoyad;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(Convert.ToInt32(dr["ID"]).ToString());
            lv.Items[sayac].SubItems.Add(dr["AD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["SOYAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["TELEFON"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ADRES"].ToString());
            lv.Items[sayac].SubItems.Add(dr["EMAIL"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 34: Müşterileri Soyada Göre Listeleme Metodu

Müşteriler içerisinde soyada göre arama yaparken kullandığımız metottur. Müşteriler bir ListView'e atarız ve müşteri adına göre arama yapabiliriz. Müşteriye ait tüm özellikleri burada tutarız.

## 9. Müşterileri Telefona Göre Listeleme Metodu

```
1 reference
public void musterigetirTlf(ListView lv, string tlf)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from MUSTERILER where TELEFON like @tlf + '%'", con);
    SqlDataReader dr = null;
    cmd.Parameters.Add("@tlf", SqlDbType.VarChar).Value = tlf;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(Convert.ToInt32(dr["ID"]).ToString());
            lv.Items[sayac].SubItems.Add(dr["AD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["SOYAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["TELEFON"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ADRES"].ToString());
            lv.Items[sayac].SubItems.Add(dr["EMAIL"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 35: Müşterileri Telefona Göre Listeleme Metodu

Müşteriler içerisinde telefona göre arama yaparken kullandığımız metottur. Müşteriler bir ListView'e atarız ve müşteri adına göre arama yapabiliriz. Müşteriye ait tüm özellikleri burada tutarız.



## E. Ödeme Class

```
5 references
class cOdeme
{
    cGenel gnl = new cGenel();
    #region Fields
    private int _OdemeID;
    private int _AdisyonID;
    private int _OdemeTurId;
    private decimal _AraToplam;
    private decimal _Indirim;
    private decimal _Kdvtutari;
    private decimal _GenelToplam;
    private DateTime _Tarih;
    private int _MusteriId;
    #endregion

    #region Properties
    0 references
    public int OdemeID { get => _OdemeID; set => _OdemeID = value; }
    2 references
    public int AdisyonID { get => _AdisyonID; set => _AdisyonID = value; }
    2 references
    public int OdemeTurId { get => _OdemeTurId; set => _OdemeTurId = value; }
    2 references
    public decimal AraToplam { get => _AraToplam; set => _AraToplam = value; }
    2 references
    public decimal Indirim { get => _Indirim; set => _Indirim = value; }
    3 references
    public decimal Kdvtutari { get => _Kdvtutari; set => _Kdvtutari = value; }
    2 references
    public decimal GenelToplam { get => _GenelToplam; set => _GenelToplam = value; }
    0 references
    public DateTime Tarih { get => _Tarih; set => _Tarih = value; }
    2 references
    public int MusteriId { get => _MusteriId; set => _MusteriId = value; }
    #endregion
}
```

Şekil 36: Ödeme Class Fields ve Properties

Ödeme classı müşterinin verdiği tüm siparişleri ödemesiyle yapılacak işlemleri kapsamaktadır. Burada veri tabanımızda ödemenin numarasını, adisyon numarasını, ödeme tür numarasını, ara toplamı, indirim tutarını, KDV tutarını, genel toplamı, tarihi ve müşteri numarasını tutmaktayız.

### 1. Ödeme Class Metotları

```
5 references
class cOdeme
{
    cGenel gnl = new cGenel();
    Fields
    Properties

    2 references
    public bool billClose(cOdeme bill)...

    0 references
    public decimal sumTotalforClientId(int clientId)...
}
```

Şekil 37: Ödeme Class Metotları

Ödeme classının içinde toplam iki adet metodumuz bulunmaktadır. Bunlar hesabı kapatmak için kullandığımız “billClose” metodu ve müşteri numarasına göre toplam tutarı hesapladığımız “sumTotalforClientId” metotlarımızdır.

## 2. Hesap Kapatma Metodu

```
2 references
public bool billClose(codeme bill)
{
    bool result = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("insert into HESAPODEMELERI (ADISYONID,ODEMETURID,MUSTERIID,ARATOPLAM,KDVTUTARI,TOPLAMTUTAR,INDIRIM) values (@ADISYONID,@ODEMETURID,@MUSTERIID,@ARATOPLAM,@KDVTUTARI,@TOPLAMTUTAR,@INDIRIM)", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@ADISYONID", SqlDbType.Int).Value = bill.AdisyonID;
        cmd.Parameters.Add("@ODEMETURID", SqlDbType.Int).Value = bill.OdemeTurId;
        cmd.Parameters.Add("@MUSTERIID", SqlDbType.Int).Value = bill.MusteriId;
        cmd.Parameters.Add("@ARATOPLAM", SqlDbType.Money).Value = bill.AraToplam;
        cmd.Parameters.Add("@KDVTUTARI", SqlDbType.Money).Value = bill.Kdvtutari;
        cmd.Parameters.Add("@INDIRIM", SqlDbType.Money).Value = bill.Indirim;
        cmd.Parameters.Add("@TOPLAMTUTAR", SqlDbType.Money).Value = bill._GenelToplam;
        result = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return result;
}
```

Şekil 38: Hesap Kapatma Metodu

Hesap kapatma olarak adlandırdığımız metotta müşterinin ödeme yapmasıyla ilgili işlemleri tamamlar.

## 3. Müşteri ID'sine Göre Toplam Tutar Metodu

```
2 references
public bool billClose(codeme bill)...
0 references
public decimal sumTotalforClientId(int clientId)
{
    decimal total = 0;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select sum(TOPLAMTUTAR) as total from HESAPODEMELERI where MUSTERIID=@clientId", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@clientId", SqlDbType.Int).Value = clientId;
        total = Convert.ToDecimal(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return total;
}
```

Şekil 39: Müşteri ID'sine Göre Toplam Tutar Metodu

Müşteri numarasına göre toplam tutarı yazdıran metodumuz dışarıdan parametre olarak müşteri numarasını almaktadır.

## F. Paketler Classı

```
7 references
class cPaketler
{
    cGenel gnl = new cGenel();
    #region Fields
    private int _ID;
    private int _AdditionID;
    private int _ClientId;
    private string _Description;
    private int _State;
    private int _Paytypeid;
    #endregion

    #region Properties
    0 references
    public int ID { get => _ID; set => _ID = value; }
    1 reference
    public int AdditionID { get => _AdditionID; set => _AdditionID = value; }
    1 reference
    public int ClientId { get => _ClientId; set => _ClientId = value; }
    1 reference
    public string Description { get => _Description; set => _Description = value; }
    0 references
    public int State { get => _State; set => _State = value; }
    1 reference
    public int Paytypeid { get => _Paytypeid; set => _Paytypeid = value; }
    #endregion
}
```

Şekil 40: Paketler Class Fields ve Properties

Paketler classımız içerisinde ID, adisyon numarası, müşteri numarası, açıklama, durumu ve ödeme tür numarası yer almaktadır.

### 1. Paket Servis Class Metotları

```
7 references
class cPaketler
{
    cGenel gnl = new cGenel();
    Fields
    Properties

    0 references
    public bool OrderServiceOpen(cPaketler order) ...
    1 reference
    public void OrderServiceClose(int AdditionID) ...
    1 reference
    public int OdemeTurIdGetir(int adisyonId) ...
    0 references
    public int musterisonAdisyonIdGetir(int musteriID) ...
    1 reference
    public bool getCheckOpenAdditionID(int additionID) ...
}
```

Şekil 41: Paket Servis Class Metotları

Paket servis classımızın içerisinde toplam beş adet metodumuz bulunmaktadır. Bu metotlar paket servis siparişinin açılması, kapanması, müşterinin son adisyon

numarasını getirme, açık adisyonu kontrol etme ve ödeme tür numarasını getirme olarak adlandırılmaktadır.

## 2. Paket Sipariş Aç Metodu

```
1 reference
public bool OrderServiceOpen(cPaketler order)
{
    bool result = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("insert into PAKETSIPARIS (ADISYONID,MUSTERIID,ODEMETURID,ACIKLAMA) values (@ADISYONID,@MUSTERIID,@ODEMETURID,@ACIKLAMA)", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@ADISYONID", SqlDbType.Int).Value = order.AdditionID;
        cmd.Parameters.Add("@MUSTERIID", SqlDbType.Int).Value = order.ClientId;
        cmd.Parameters.Add("@ODEMETURID", SqlDbType.Int).Value = order.PaytypeId;
        cmd.Parameters.Add("@ACIKLAMA", SqlDbType.VarChar).Value = order.Description;
        result = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return result;
}
```

Şekil 42: Paket Sipariş Aç Metodu

Veri tabanından paket sipariş tablosuna kullanıcıdan gelen adisyon numarası, müşteri numarası, ödeme tür numarası ve açıklamaya istinaden ekleme yapan yani sipariş oluşturan metottur. Geriye bool türünde değer döndürür.

## 3. Paket Sipariş Kapat Metodu

```
1 reference
public void OrderServiceClose(int AdditionID)
{
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("update PAKETSIPARIS set PAKETSIPARIS.DURUM=1 where PAKETSIPARIS.ADISYONID=@AdditionID", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@AdditionID", SqlDbType.Int).Value = AdditionID;
        Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
}
```

Şekil 43: Paket Sipariş Kapat Metodu

Veri tabanından paket sipariş tablosuna kullanıcıdan gelen adisyon numarasına göre ilgili siparişi kapatan, veri tabanından siparişi kapatan metottur. Geriye değer döndürmez.

#### 4. Ödeme Türünü Getir Metodu

```
1 reference
public int OdemeTurIdGetir(int adisyonId)
{
    int odemeTurID = 0;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select PAKETSIPARIS.ODEMETURID from PAKETSIPARIS inner join ADISYON on PAKETSIPARIS.ADISYONID=ADISYON.ID where ADISYON.ID=@adisyonId", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@adisyonId", SqlDbType.Int).Value = adisyonId;
        odemeTurID = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return odemeTurID;
}
```

Şekil 44: Ödeme Türünü Getir Metodu

Kullanıcıdan gelen adisyon numarasına göre ilgili siparişin ödeme türünü seçen metottur.

#### 5. Müşterinin Son Adisyonunu Getir Metodu

```
6 references
public int musterisonAdisyonIdGetir(int musterIID)
{
    int no = 0;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select ADISYON.ID from ADISYON inner join PAKETSIPARIS on PAKETSIPARIS.ADISYONID=ADISYON.ID where (ADISYON.DURUM=0) and (PAKETSIPARIS.DURUM=0) and PAKETSIPARIS.MUSTERIID=@musterIID", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@musterIID", SqlDbType.Int).Value = musterIID;
        no = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return no;
}
```

Şekil 45: Müşterinin Son Adisyonunu Getir Metodu

Müşteri numarasına göre adisyon tablosundan müşteriye ait olan son adisyonu getiren metottur. Geriye int türünde adisyon numarasını döndürür.

## 6. Açık Adisyonun ID'sinin Getir Metodu

```
1 reference
public bool getCheckOpenAdditionID(int additionID)
{
    bool result = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from ADISYON where (DURUM=0) and (ID=@additionID)", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@additionID", SqlDbType.Int).Value = additionID;
        result = Convert.ToBoolean(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return result;
}
```

Şekil 46: Açık Adisyonun ID'sinin Getir Metodu

Kullanıcıdan gelen adisyon numarasına göre adisyon tablosundan ilgili adisyonu bulan ve durumu sıfıra eşit koşulunu sağlayan metottur. Geriye bool türünde bir sonuç döndürür.

## G. Personel Görev Classı

```
6 references
class cPersonelGorev
{
    cGenel gnl = new cGenel();

    #region Fields

    private int personelGorevId;
    private string _tanım;
    #endregion

    #region Properties
1 reference
    public int PersonelGorevId { get => personelGorevId; set => personelGorevId = value; }
0 references
    public string Tanım { get => _tanım; set => _tanım = value; }
    #endregion
}
```

Şekil 47: Personel Görev Class Fields ve Properties

Personel görev classı ilgili işletmede çalışan personellerin görev tanımlarını içermektedir.

## 1. Personel Görev Metotları

```
6 references
class cPersonelGorev
{
    cGenel gnl = new cGenel();

    Fields

    Properties

    1 reference
    public void PersonelGorevGetir(ComboBox cb) ...
    1 reference
    public string PersonelGorevTanim(int per) ...
    0 references
    public override string ToString() ...
}
```

Şekil 48: Personel Görev Metotları

Personel görev metotlarımız iki tanedir. Personel görevlerini ve personel görev tanımlarını getirdiğimiz metotlardır.

## 2. Personel Görevlerini ComboBox'a Getir Metodu

```
1 reference
public void PersonelGorevGetir(ComboBox cb)
{
    cb.Items.Clear();

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from PERSONELGOREVLERI", con);
    SqlDataReader dr = null;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            cPersonelGorev c = new cPersonelGorev();
            c.personelGorevId = Convert.ToInt32(dr["ID"].ToString());
            c._tanim = dr["GOREV"].ToString();
            cb.Items.Add(c);
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    dr.Close();
    con.Close();
}
```

Şekil 49: Personel Görevlerini ComboBox'a Getir Metodu

### 3. Personel Görev Tanım Metodu

```
1 reference
public string PersonelGorevTanım(int per)
{
    string sonuc = "";
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select GOREV from PERSONELGOREVLERI where ID=@perId", con);
    cmd.Parameters.Add("@perId", SqlDbType.Int).Value = per;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        sonuc = cmd.ExecuteScalar().ToString();
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    con.Close();
    return sonuc;
}
```

Şekil 50: Personel Görev Tanım Metodu

## H. Personel Hareketleri Classı

```
3 references
class cPersonelHareketleri
{
    cGenel gnl = new cGenel();
    #region Field
    private int _ID;
    private int _PersonelId;
    private string _Islem;
    private DateTime _Tarih;
    private bool _Durum;
    #endregion
    #region Properties
    0 references
    public int ID { get => _ID; set => _ID = value; }
    1 reference
    public int PersonelId { get => _PersonelId; set => _PersonelId = value; }
    1 reference
    public string Islem { get => _Islem; set => _Islem = value; }
    1 reference
    public DateTime Tarih { get => _Tarih; set => _Tarih = value; }
    0 references
    public bool Durum { get => _Durum; set => _Durum = value; }
    #endregion
}
```

Şekil 51: Personel Hareket Class Fields ve Properties

Personel hareket classımızın içerisinde personelin sisteme giriş çıkışını takip eden classtır. Bu class içerisinde hareket numarası, personel numarası, işlemler, tarih ve durum bilgileri tutulmaktadır.



## 1. Personel Hareketleri Class Metotları

```
3 references
class cPersonelHareketleri
{
    cGenel gnl = new cGenel();
    Field
    Properties

    1 reference
    public bool PersonelActionSave(cPersonelHareketleri ph)...
```

Şekil 52: Personel Hareketleri Class Metotları

Bu class içerisinde bir adet metot bulunmaktadır. Bu metot personelin otomasyon içerisinde yaptığı hareketleri kaydetme işlemini yapar.

## 2. Personel Hareketleri Kayıt Metodu

```
1 reference
public bool PersonelActionSave(cPersonelHareketleri ph)
{
    bool result = false;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Insert into PERSONELHAREKETLERI(PERSONELID,ISLEM,TARİH) values(@personelId,@islem,@tarih)", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@personelId", SqlDbType.Int).Value = ph._PersonelId;
        cmd.Parameters.Add("@islem", SqlDbType.VarChar).Value = ph._Islem;
        cmd.Parameters.Add("@tarih", SqlDbType.Date).Value = ph._Tarih;

        result = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    return result;
}
```

Şekil 53: Personel Hareketleri Kayıt Metodu

Personellerin giriş yaptıktan sonra otomasyon içerisinde yaptığı hareketleri kaydettiği metottur. Personel hareketleri tablosunda personel numarası, yapılan işlem ve işlemin yapıldığı tarih olarak veriler tutulmaktadır.

## İ. Personeller Class

```
24 references
class cPersoneller
{
    cGenel gnl = new cGenel();
    #region Field
    private int _PersonelId;
    private int _PersonelGorevId;
    private string _PersonelAd;
    private string _PersonelSoyad;
    private string _PersonelParola;
    private string _PersonelKullaniciAdi;
    private bool _PersonelDurum;
    #endregion
    #region Properties
    2 references
    public int PersonelId { get => _PersonelId; set => _PersonelId = value; }
    3 references
    public int PersonelGorevId { get => _PersonelGorevId; set => _PersonelGorevId = value; }
    1 reference
    public string PersonelAd { get => _PersonelAd; set => _PersonelAd = value; }
    3 references
    public string PersonelSoyad { get => _PersonelSoyad; set => _PersonelSoyad = value; }
    2 references
    public string PersonelParola { get => _PersonelParola; set => _PersonelParola = value; }
    0 references
    public string PersonelKullaniciAdi { get => _PersonelKullaniciAdi; set => _PersonelKullaniciAdi = value; }
    0 references
    public bool PersonelDurum { get => _PersonelDurum; set => _PersonelDurum = value; }
    #endregion
}
```

Şekil 54: Personeller Class Fields ve Properties

Personeller class içerisinde personel numarası, personel görev numarası, personel adı, soyadı, parolası, kullanıcı adı ve personelin durumu bilgileri tutulmaktadır.

### 1. Personeller Class Metotları

```
24 references
class cPersoneller
{
    cGenel gnl = new cGenel();
    #region Field
    #region Properties

    1 reference
    public bool personelEntryControl(string password, int UserId)...
    2 references
    public void personelGetbyInformation(ComboBox cb)...
    0 references
    public override string ToString()...
    4 references
    public void personelBilgileriniGetirLV(ListView lv)...
    0 references
    public void personelBilgileriniGetirfromIDLV(ListView lv, int perId)...
    2 references
    public string personelBilgiGetirIsim(int perId)...
    2 references
    public bool personelsifreDegistir(int personelID, string pass)...
    1 reference
    public bool personelEkle(cPersoneller cp)...
    1 reference
    public bool personelGuncelle(cPersoneller cp, int perId)...
    1 reference
    public bool personelsil(int perId)...
    }
}
```

Şekil 55: Personeller Class Metotları

Personeller classının içerisinde dokuz adet metot bulunmaktadır. Bu metotlar personeli silmek, güncellemek ve eklemek; personel şifresini değiştirmek, personel giriş çıkışlarını kontrol etmek, personel bilgilerini comboBox içerisine, listview'e getirmek, personel bilgilerini isme göre aramak gibi metotları içerir.

## 2. Personel Giriş Kontrol Metodu

```
1 reference
public bool personelEntryControl(string password, int UserId)
{
    bool result = false;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select*from PERSONELLER where ID=@Id and PAROLA=@password", con);
    cmd.Parameters.Add("@Id", SqlDbType.VarChar).Value = UserId;
    cmd.Parameters.Add("@password", SqlDbType.VarChar).Value = password;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        result = Convert.ToBoolean(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    return result;
}
```

Şekil 56: Personel Giriş Kontrol Metodu

Personeller tablosu içerisinde önceden tanımlanmış veya personel ekle metodu ile eklenmiş müşterilerin kullanıcı adı ve şifrelerini kontrol ederek otomasyona giriş yapılmasını sağlayan metottur.

## 3. Personel Bilgilerini ComboBox'a Getir Metodu

```
2 references
public void personelGetbyInformation(ComboBox cb)
{
    cb.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select*from PERSONELLER", con);

    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }

    SqlDataReader dr = cmd.ExecuteReader();

    while (dr.Read())
    {
        cPersoneller p = new cPersoneller();
        p._PersonelId = Convert.ToInt32(dr["ID"]);
        p._PersonelGorevId = Convert.ToInt32(dr["GOREVID"]);
        p._PersonelAd = Convert.ToString(dr["AD"]);
        p._PersonelSoyad = Convert.ToString(dr["SOYAD"]);
        p._PersonelParola = Convert.ToString(dr["PAROLA"]);
        p._PersonelKullaniciAdi = Convert.ToString(dr["KULLANICIADI"]);
        p._PersonelDurum = Convert.ToBoolean(dr["DURUM"]);
        cb.Items.Add(p);
    }
    dr.Close();
    con.Close();
}
```

Şekil 57: Personelleri ComboBox'a Getir Metodu

Sisteme kayıtlı olan tüm personellerin bilgilerini çağırıldığı combobox'a getiren metottur. Personelleri, veri tabanındaki personeller tablosundan çekmektedir.

#### 4. Personel Bilgilerini ListView'e Getir Metodu

```
public void personelBilgileriniGetirV(ListView lv)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select PERSONELLER.*,PERSONELGOREVLERI.GOREV from PERSONELLER inner join PERSONELGOREVLERI on PERSONELGOREVLERI.ID= PERSONELLER.GOREVID where PERSONELLER.DURUM=0", con);

    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }

    SqlDataReader dr = cmd.ExecuteReader();
    int i = 0;
    while (dr.Read())
    {
        lv.Items.Add(dr["ID"].ToString());
        lv.Items[i].SubItems.Add(dr["GOREVID"].ToString());
        lv.Items[i].SubItems.Add(dr["GOREV"].ToString());
        lv.Items[i].SubItems.Add(dr["AD"].ToString());
        lv.Items[i].SubItems.Add(dr["SOYAD"].ToString());
        lv.Items[i].SubItems.Add(dr["KULLANICIADI"].ToString());
        i++;
    }
    dr.Close();
    con.Close();
}
```

Şekil 58: Personel Bilgilerini ListView'e Getir Metodu

Sisteme kayıtlı olan tüm personellerin bilgilerini çağırıldığı listview'e getiren metottur. Personelleri, veri tabanındaki personeller tablosundan çekmektedir.

#### 5. Personel Bilgilerini ID'ye Göre ListView'e Getir Metodu

```
public void personelBilgileriniGetirIDV(ListView lv, int perId)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select PERSONELLER.*,PERSONELGOREVLERI.GOREV from PERSONELLER inner join PERSONELGOREVLERI on PERSONELGOREVLERI.ID= PERSONELLER.GOREVID where PERSONELLER.DURUM=0 and PERSONELLER.ID=@perId", con);

    cmd.Parameters.Add("@perId", SqlDbType.Int).Value = perId;

    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }

    SqlDataReader dr = cmd.ExecuteReader();
    int i = 0;
    while (dr.Read())
    {
        lv.Items.Add(dr["ID"].ToString());
        lv.Items[i].SubItems.Add(dr["GOREVID"].ToString());
        lv.Items[i].SubItems.Add(dr["GOREV"].ToString());
        lv.Items[i].SubItems.Add(dr["AD"].ToString());
        lv.Items[i].SubItems.Add(dr["SOYAD"].ToString());
        lv.Items[i].SubItems.Add(dr["KULLANICIADI"].ToString());
        i++;
    }
    dr.Close();
    con.Close();
}
```

Şekil 59: Personel Bilgilerini ID'ye Göre ListView'e Getir Metodu

Personel bilgilerini personel numarasına göre çağırıldığı listview'e getiren metottur. Personeller tablosundan personel numarası çağırılarak yapılır.

#### 6. Personel Bilgilerini İsme Göre Getir Metodu

```
public string personelBilgiGetirIsim(int perId)
{
    string sonuc = "";
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select (AD + SOYAD) from PERSONELLER where PERSONELLER.DURUM=0 and PERSONELLER.ID=@perId", con);

    cmd.Parameters.Add("@perId", SqlDbType.Int).Value = perId;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        sonuc = Convert.ToString(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    con.Close();
    return sonuc;
}
```

Şekil 60: Personel Bilgilerini İsme Göre Getir Metodu

Personel bilgilerini personel adına göre çağırıldığı listview'e getiren metottur. Personeller tablosundan personel adı seçilerek yapılır.

## 7. Personel Şifre Değiştir Metodu

```
2 references
public bool personelSifreDegistir(int personelID, string pass)
{
    bool sonuc = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("update PERSONELLER set PAROLA=@pass where ID=@perId", con);

    cmd.Parameters.Add("@perId", SqlDbType.Int).Value = personelID;
    cmd.Parameters.Add("@pass", SqlDbType.VarChar).Value = pass;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        sonuc = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    con.Close();
    return sonuc;
}
```

Şekil 61: Personel Şifre Değiştir Metodu

Personellerin şifrelerini değiştirmek için kullanılan metottur. Veri tabanından personeller tablosundan şifre hücreğini işlem yapılan personel numarasına göre günceller.

## 8. Personel Ekle Metodu

```
1 reference
public bool personelEkle(cPersoneller cp)
{
    bool sonuc = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("insert into PERSONELLER(AD,SOYAD,PAROLA,GOREVID) values (@AD,@SOYAD,@PAROLA,@GOREVID)", con);

    cmd.Parameters.Add("@AD", SqlDbType.VarChar).Value = _PersonelAd;
    cmd.Parameters.Add("@SOYAD", SqlDbType.VarChar).Value = _PersonelSoyad;
    cmd.Parameters.Add("@PAROLA", SqlDbType.VarChar).Value = _PersonelParola;
    cmd.Parameters.Add("@GOREVID", SqlDbType.Int).Value = _PersonelGorevid;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        sonuc = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    con.Close();
    return sonuc;
}
```

Şekil 62: Personel Ekle Metodu

Personel eklenmesi durumunda çağırılacak metottur. Personellerin ad, soyad, parola, görev numarası gibi bilgileri alınarak, personeller tablosuna eklenerek oluşturulur.

## 9. Personel Güncelle Metodu

```
1 reference
public bool personelGuncelle(cPersoneller cp, int perId)
{
    bool sonuc = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("update PERSONELLER set (AD=@AD,SOYAD=@SOYAD,PAROLA=@PAROLA,GOREVID=@GOREVID) where ID=@perID", con);

    cmd.Parameters.Add("@perID", SqlDbType.Int).Value = perId;
    cmd.Parameters.Add("@AD", SqlDbType.VarChar).Value = _PersonelAd;
    cmd.Parameters.Add("@SOYAD", SqlDbType.VarChar).Value = _PersonelSoyad;
    cmd.Parameters.Add("@PAROLA", SqlDbType.VarChar).Value = _PersonelParola;
    cmd.Parameters.Add("@GOREVID", SqlDbType.Int).Value = _PersonelGorevid;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        sonuc = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    con.Close();
    return sonuc;
}
```

Şekil 63: Personel Güncelle Metodu

Personel güncellenmesi durumunda çağırılacak metottur. Personellerin ad, soyad, parola, görev numarası gibi bilgileri alınarak, personeller tablosunda güncellenmesiyle oluşturulur.

## 10. Personel Sil Metodu

```
1 reference
public bool personelsil(int perId)
{
    bool sonuc = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("update PERSONELLER set DURUM=1 where ID=@perID", con);

    cmd.Parameters.Add("@perID", SqlDbType.Int).Value = perId;

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        sonuc = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    con.Close();
    return sonuc;
}
```

Şekil 64: Personel Sil Metodu

Personel silinmesi durumunda çağırılacak metottur. Personellerin ad, soyad, parola, görev numarası gibi bilgileri personeller tablosundan silinmesiyle oluşturulur.

## J. Rezervasyon Class

```
7 references
class cRezervasyon
{
    #region Fields
    private int _ID;
    private int _TableId;
    private int _ClientId;
    private DateTime _Date;
    private int _ClientCount;
    private string _Description;
    private int _AdditionId;
    #endregion
    #region Properties
    0 references
    public int ID { get => _ID; set => _ID = value; }
    2 references
    public int TableId { get => _TableId; set => _TableId = value; }
    2 references
    public int ClientId { get => _ClientId; set => _ClientId = value; }
    2 references
    public DateTime Date { get => _Date; set => _Date = value; }
    2 references
    public int ClientCount { get => _ClientCount; set => _ClientCount = value; }
    2 references
    public string Description { get => _Description; set => _Description = value; }
    2 references
    public int AdditionId { get => _AdditionId; set => _AdditionId = value; }
    #endregion
}
```

Şekil 65: Rezervasyon Class Fields ve Properties

Rezervasyon classı, rezervasyon ile ilgili bilgilerin tutulması için hazırlanmıştır. Rezervasyon numarası, masa numarası, müşteri numarası, tarih, müşteri adeti, açıklama ve adisyon numarası gibi bilgileri tutmaktadır.

### 1. Rezervasyon Class Metotları

```
7 references
class cRezervasyon
{
    Fields
    Properties

    cGenel gn1 = new cGenel();
    1 reference
    public int getByClientIdFromRezervasyon(int tableId)...
    1 reference
    public bool rezervasyonclose(int adisyonID)...
    0 references
    public void musterIdGetirFromRezervasyon(ListView lv)...
    0 references
    public void eskiRezervasyonlariGetir(ListView lv, int mId)...
    0 references
    public DateTime EnSonRezervasyonTarihi(int mId)...
    0 references
    public int acikRezervasyonSayisi()...
    1 reference
    public bool RezervasyonAcikmiKontrol(int mId)...
    1 reference
    public bool RezervasyonAc(cRezervasyon r)...
    0 references
    public int RezerverMasaIdGetir(int mId)...
}
```

Şekil 66: Rezervasyon Class Metotları

Rezervasyon class metotları dokuz adettir. Rezervasyonu açmak ve kapatmak, kontrol etmek, açık rezervasyon sayısını bulmak, eski rezervasyonları ve en son rezervasyon tarihini bulmak, müşteri numarasına göre rezervasyonu ListView'e getirmek gibi metotlar vardır.

## 2. Müşteri ID'sine Göre Rezervasyon Metodu

```
1 reference
public int getByClientIdFromRezervasyon(int tableId)
{
    int clientId = 0;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select top 1 MUSTERIID from REZERVASYONLAR where MASAIID=@masaid order by MUSTERIID desc", con);

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@masaid", SqlDbType.Int).Value = tableId;
        clientId = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }

    return clientId;
}
```

Şekil 67: Müşteri ID'sine Göre Rezervasyon Metodu

Seçilen müşteri numarasına göre ilgili masa numarasına ilgili müşteriyi atayan metottur. Rezervasyon tablosundan masa numarasını dışarıdan aldığı parametre ile atayarak rezervasyonu yapmaktadır.

## 3. Rezervasyon Kapatma Metodu

```
1 reference
public bool rezervasyonclose(int adisyonID)
{
    bool result = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("update REZERVASYONLAR set DURUM=1 where ADISYONID=@adisyonId", con);

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@adisyonId", SqlDbType.Int).Value = adisyonID;
        result = Convert.ToBoolean(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }

    return result;
}
```

Şekil 68: Rezervasyon Kapatma Metodu

Rezervasyonlar tablosundan adisyon numarasına göre rezervasyonu kapatan ve veri tabanında güncelleyen metottur. Dışarıdan parametre olarak adisyon numarasını almaktadır.



## 4. Rezervasyonlardan müşteri ID'sini Getir Metodu

```
0 references
public void musterIdGetirFromRezervasyon(ListView lv)
{
    lv.Items.Clear();
    SqlConnection conn = new SqlConnection(gnl.conString);
    SqlCommand comm = new SqlCommand("select REZERVASYONLAR.MUSTERIID,( AD+SOYAD) AS MUSTERI FROM REZERVASYONLAR INNER JOIN MUSTERILER ON REZERVASYONLAR.MUSTERIID=MUSTERILER.ID WHERE REZERVASYONLAR.DURUM=0", conn);

    if (conn.State == ConnectionState.Closed)
    {
        conn.Open();
    }
    SqlDataReader dr = comm.ExecuteReader();

    int i = 0;
    while (dr.Read())
    {
        lv.Items.Add(dr["MUSTERIID"].ToString());
        lv.Items[i].SubItems.Add(dr["MUSTERI"].ToString());
        i++;
    }
    dr.Close();
    conn.Dispose();
    conn.Close();
}
```

Şekil 69: Rezervasyonlardan müşteri ID'sini Getir Metodu

## 5. Eski Rezervasyonları Getir Metodu

```
0 references
public void eskiRezervasyonlariGetir(ListView lv, int mid)
{
    lv.Items.Clear();
    SqlConnection conn = new SqlConnection(gnl.conString);
    SqlCommand comm = new SqlCommand("select REZERVASYONLAR.MUSTERIID,AD,SOYAD,ADISYONID,TARİH FROM REZERVASYONLAR INNER JOIN MUSTERILER ON REZERVASYONLAR.MUSTERIID=MUSTERILER.ID WHERE REZERVASYONLAR.MUSTERIID=@mid and REZERVASYONLAR.DURUM=0 order by REZERVASYONLAR.ID DESC", conn);

    comm.Parameters.Add("mid", SqlDbType.Int).Value = mid;

    if (conn.State == ConnectionState.Closed)
    {
        conn.Open();
    }
    SqlDataReader dr = comm.ExecuteReader();

    int i = 0;
    while (dr.Read())
    {
        lv.Items.Add(dr["MUSTERIID"].ToString());
        lv.Items[i].SubItems.Add(dr["AD"].ToString());
        lv.Items[i].SubItems.Add(dr["SOYAD"].ToString());
        lv.Items[i].SubItems.Add(dr["TARİH"].ToString());
        lv.Items[i].SubItems.Add(dr["ADISYONID"].ToString());
        i++;
    }
    dr.Close();
    conn.Dispose();
    conn.Close();
}
```

Şekil 70: Eski Rezervasyonları Getir Metodu

## 6. En Son Rezervasyon Tarihini Getir Metodu

```
0 references
public DateTime EnSonRezervasyonTarihi(int mid)
{
    DateTime tar = new DateTime();
    tar = DateTime.Now;

    SqlConnection conn = new SqlConnection(gnl.conString);
    SqlCommand comm = new SqlCommand("select TARİH FROM REZERVASYONLAR WHERE REZERVASYONLAR.MUSTERIID=@mid and REZERVASYONLAR.DURUM=1 order by REZERVASYONLAR.ID DESC", conn);

    comm.Parameters.Add("@mid", SqlDbType.Int).Value = mid;

    if (conn.State == ConnectionState.Closed)
    {
        conn.Open();
    }
    tar = Convert.ToDateTime(comm.ExecuteScalar());

    conn.Dispose();
    conn.Close();

    return tar;
}
```

Şekil 71: Rezervasyon Tarihini Getir Metodu

## 7. Açık Rezervasyon Sayısını Say Metodu

```
0 references
public int açıkRezervasyonSayisi()
{
    int sonuc = 0;
    SqlConnection conn = new SqlConnection(gnl.conString);
    SqlCommand comm = new SqlCommand("select count(*) FROM REZERVASYONLAR WHERE REZERVASYONLAR.DURUM=0", conn);

    if (conn.State == ConnectionState.Closed)
    {
        conn.Open();
    }

    try
    {
        sonuc = Convert.ToInt32(comm.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    conn.Dispose();
    conn.Close();

    return sonuc;
}
```

Şekil 72: Açık Rezervasyon Sayısını Say Metodu

## 8. Rezervasyon Durum=1 Kontrol Metodu

```
1 reference
public bool RezervasyonAcikmiKontrol(int mId)
{
    bool result = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select top 1 REZERVASYONLAR.ID FROM REZERVASYONLAR WHERE MUSTERIID=@mId and DURUM=1 ORDER BY ID DESC", con);

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@mId", SqlDbType.Int).Value = mId;
        result = Convert.ToBoolean(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }

    return result;
}
```

Şekil 73: Rezervasyon Durum=1 Kontrol Metodu

## 9. Rezervasyon Açma Metodu

```
1 reference
public bool RezervasyonAc(CRezervasyon r)
{
    bool result = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("insert into REZERVASYONLAR (MUSTERIID,MASAIID,ADISYONID,KISISAYISI,TARİH,ACIKLAMA,DURUM) VALUES (@MUSTERIID,@MASAIID,@ADISYONID,@KISISAYISI,@TARİH,@ACIKLAMA,1)", con);

    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@MUSTERIID", SqlDbType.Int).Value = r.ClientId;
        cmd.Parameters.Add("@MASAIID", SqlDbType.Int).Value = r.TableId;
        cmd.Parameters.Add("@ADISYONID", SqlDbType.Int).Value = r.AdditionId;
        cmd.Parameters.Add("@KISISAYISI", SqlDbType.Int).Value = r.ClientCount;
        cmd.Parameters.Add("@TARİH", SqlDbType.Date).Value = r.Date;
        cmd.Parameters.Add("@ACIKLAMA", SqlDbType.VarChar).Value = r.Description;

        result = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }

    return result;
}
```

Şekil 74: Rezervasyon Açma Metodu

## 10. Rezerve Masa ID'sini Getir Metodu

```
0 references
public int RezerveMasaIdGetir(int mid)
{
    int sonuc = 0;
    SqlConnection conn = new SqlConnection(gnl.conString);
    SqlCommand comm = new SqlCommand("select REZERVASYONLAR.MASAID FROM REZERVASYONLAR INNER JOIN ADISYON ON REZERVASYONLAR.ADISYONID = ADISYON.ID WHERE REZERVASYONLAR.DURUM=1 AND ADISYON.DURUM=0 AND REZERVASYONLAR.MUSTERIID=@mid ", conn);

    if (conn.State == ConnectionState.Closed)
    {
        conn.Open();
    }

    try
    {
        comm.Parameters.Add("@mid", SqlDbType.Int).Value = mid;
        sonuc = Convert.ToInt32(comm.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
        throw;
    }

    conn.Dispose();
    conn.Close();

    return sonuc;
}
```

Şekil 75: Rezerve Masa ID'sini Getir Metodu

## K. Sipariş Class

```
13 references
class Siparis
{
    cGenel gnl = new cGenel();

    #region Fields
    private int _Id;
    private int _adisyonID;
    private int _urunId;
    private int _adet;
    private int _masaId;
    private int _durum;
    #endregion
    #region Properties
    0 references
    public int Id { get => _Id; set => _Id = value; }
    3 references
    public int AdisyonID { get => _adisyonID; set => _adisyonID = value; }
    3 references
    public int UrunId { get => _urunId; set => _urunId = value; }
    3 references
    public int Adet { get => _adet; set => _adet = value; }
    3 references
    public int MasaId { get => _masaId; set => _masaId = value; }
    0 references
    public int Durum { get => _durum; set => _durum = value; }
    #endregion
}
```

Şekil 76: Sipariş Class Fields ve Properties

Siparişlerle ilgili bilgilerin bulunduğu classtır. Sipariş numarası, adisyon numarası, ürün numarası, adet, masa numarası ve siparişin durumu gibi bilgiler bu class içerisinde tutulur.

## 1. Sipariş Class Metotları

```
13 references
class cSiparis
{
    cGenel gnl = new cGenel();

    Fields
    Properties

    3 references
    public void getByOrder(ListView lv, int AdisyonId)...
    3 references
    public bool setSaveOrder(cSiparis Bilgiler)...
    2 references
    public void setDeleteOrder(int satisId)...
    2 references
    public decimal GenelToplamBul(int musterId)...
    1 reference
    public void adisyonpaketsiparisDetaylari(ListView lv, int adisyonID)...
}
```

Şekil 77: Sipariş Class Metotları

Bu classın içerisinde toplam beş adet metot bulunmaktadır. Bu metotlar siparişi açmak, kaydetmek, silmek, genel toplam bulmak ve adisyon numarasına göre paket sipariş detaylarını listviewe göndermek gibi içerikleri yapar.

## 2. Siparişleri ListView'e Atama Metodu

```
1 references
public void getByOrder(ListView lv, int AdisyonId)
{
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Select URUNAD,FIYAT, SATISLAR.ID,URUNID,SATISLAR.ADET from SATISLAR inner join URUNLER on SATISLAR.URUNID=URUNLER.ID where ADISYONID=@AdisyonID", con);
    SqlDataReader dr = null;
    cmd.Parameters.Add("@AdisyonID", SqlDbType.Int).Value = AdisyonId;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["URUNAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ADET"].ToString());
            lv.Items[sayac].SubItems.Add(dr["URUNID"].ToString());
            lv.Items[sayac].SubItems.Add(Convert.ToString(Convert.ToDecimal(dr["FIYAT"]) * Convert.ToDecimal(dr["ADET"])));
            lv.Items[sayac].SubItems.Add(dr["ID"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        dr.Dispose();
        con.Close();
    }
}
```

Şekil 78: Siparişleri ListView'e Atama Metodu

Girilen siparişleri ürün adı, adeti, ürün numarası, fiyatı ve numarasına göre listview'e atar. Metodu kullanabilmek için listview ve adisyon numarası gönderilmelidir.

### 3. Siparişleri Kaydetme Metodu

```
3 references
public bool setSaveOrder(cSiparis Bilgiler)
{
    bool sonuc = false;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("insert into SATISLAR(ADISYONID,URUNID,ADET,MASAIID) values (@AdisyonNo,@UrunId,@Adet,@masaId", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        cmd.Parameters.Add("@AdisyonNo", SqlDbType.Int).Value = Bilgiler._adisyonID;
        cmd.Parameters.Add("@UrunId", SqlDbType.Int).Value = Bilgiler._urunId;
        cmd.Parameters.Add("@Adet", SqlDbType.Int).Value = Bilgiler._adet;
        cmd.Parameters.Add("@masaId", SqlDbType.Int).Value = Bilgiler._masaId;
        sonuc = Convert.ToBoolean(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 79: Siparişleri Kaydetme Metodu

Girilen siparişlerin veri tabanında satışlar tablosuna adisyon numarası, ürün numarası, adeti ve masa numarasına göre kaydedilmesini sağlar.

### 4. Siparişi Silme Metodu

```
2 references
public void setDeleteOrder(int satisId)
{
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Delete from SATISLAR where ID=@SatisID", con);

    cmd.Parameters.Add("@SatisID", SqlDbType.Int).Value = satisId;

    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }
    cmd.ExecuteNonQuery();
    con.Dispose();
    con.Close();
}
```

Şekil 80: Siparişi Silme Metodu

Girilen siparişlerin satışlar tablosundan satış numarası gönderilerek silinmesini sağlar. Parametre olarak satış numarasını göndeririz.

## 5. Sipariş Genel Toplam Metodu

```
2 references
public decimal GenelToplamBul(int musteriiId)
{
    decimal geneltoplam = 0;
    SqlConnection con = new SqlConnection(gnl.conString);

    SqlCommand cmd = new SqlCommand("select SUM(TOPLAMTUTAR) from HESAPODEMELERI where MUSTERIID=@musteriiId", con);
    cmd.Parameters.Add("@musteriiId", SqlDbType.Int).Value = musteriiId;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        geneltoplam = Convert.ToDecimal(cmd.ExecuteScalar());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return geneltoplam;
}
```

Şekil 81: Sipariş Genel Toplam Metodu

Müşteri numarasına göre toplam tutarı hesaplayan metottur. Parametre olarak müşteri numarası kullanılır ve veri tabanı hesap ödemeleri tablosundaki verileri seçer.

## 6. Paket Sipariş Adisyon Detayları ListView Metodu

```
1 reference
public void gdisyonpaketsiparisDetaylari(ListView lv, int adisyonID)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select SATISLAR.ID as satisID, URUNLER.URUNAD, URUNLER.FIYAT, SATISLAR.ADET from SATISLAR inner join ADISYON on ADISYON.ID=SATISLAR.ADISYONID inner join URUNLER on URUNLER.ID=SATISLAR.URUNID where SATISLAR.ADISYONID=@adisyonID", con);
    cmd.Parameters.Add("@adisyonID", SqlDbType.Int).Value = adisyonID;
    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        int i = 0;
        dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            lv.Items.Add(dr["satisID"].ToString());
            lv.Items[i].SubItems.Add(dr["URUNAD"].ToString());
            lv.Items[i].SubItems.Add(dr["ADET"].ToString());
            lv.Items[i].SubItems.Add(dr["FIYAT"].ToString());
            i++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
}
```

Şekil 82: Paket Sipariş Adisyon Detayları ListView Metodu

Listview ve adisyon numarası kullanılarak satışlar tablosundan verileri seçer. Liste içerisine satış numarası, ürün adı, adeti, fiyatı yansıtılır.

## L. Ürün Çeşitleri Class

```
24 references
class cUrunCesitleri
{
    cGenel gn1 = new cGenel();

    #region Fields
    private int _UrunTurNo;
    private string _KategoriAd;
    private string _Aciklama;
    #endregion
    #region properties
    2 references
    public int UrunTurNo { get => _UrunTurNo; set => _UrunTurNo = value; }
    3 references
    public string KategoriAd { get => _KategoriAd; set => _KategoriAd = value; }
    2 references
    public string Aciklama { get => _Aciklama; set => _Aciklama = value; }
    #endregion
}
```

Şekil 83: Ürün Çeşitleri Class Fields ve Properties

Ürün çeşitleri classı içerisinde ürün kategori numarasını, kategori adını ve açıklama bilgilerini tutar.

### 1. Ürün Çeşitleri Class Metotları

```
24 references
class cUrunCesitleri
{
    cGenel gn1 = new cGenel();

    Fields
    properties
    8 references
    public void getByProductTypes(ListView Cesitler, Button btn)...
    1 reference
    public void getByProductSearch(ListView Cesitler, int txt)...
    2 references
    public void urunCesitleriniGetir(ComboBox cb)...
    2 references
    public void urunCesitleriniGetir(ListView lv)...
    1 reference
    public void urunCesitleriniGetir(ListView lv, string source)...
    1 reference
    public int urunKategoriEkle(cUrunCesitleri uc)...
    1 reference
    public int urunKategoriGuncelle(cUrunCesitleri uc)...
    2 references
    public int urunKategoriSil(int id)...
    0 references
    public override string ToString()...
}
```

Şekil 84: Ürün Çeşitleri Class Metotları

Ürün çeşitleri metotları sekiz adettir. Ürün çeşitlerini text içerisinde aratmak için yazılan iki adet metot vardır. Aynı zamanda ürün çeşitlerini combobox, list viewe göre getiren üç adet metot vardır. Ürünlere kategori eklenmesi, güncellenmesi ve silinmesi ile ilgili üç adet metot vardır.

## 2. Ürün Çeşitlerini ListView'e Getir Metodu

```
8 reference
public void getByProductTypes(ListView Cesitler, Button btn)
{
    Cesitler.Items.Clear();
    SqlConnection conn = new SqlConnection(gnl.conString);
    SqlCommand comm = new SqlCommand("Select URUNAD,FIYAT,URUNLER.ID from KATEGORILER inner join URUNLER on KATEGORILER.ID=URUNLER.KATEGORIID where URUNLER.KATEGORIID=@KATEGORIID", conn);
    string aa = btn.Name;
    int uzunluk = aa.Length;

    comm.Parameters.Add("@KATEGORIID", SqlDbType.Int).Value = aa.Substring(uzunluk - 1, 1);
    if (conn.State == ConnectionState.Closed)
    {
        conn.Open();
    }
    SqlDataReader dr = comm.ExecuteReader();
    int i = 0;
    while (dr.Read())
    {
        Cesitler.Items.Add(dr["URUNAD"].ToString());
        Cesitler.Items[i].SubItems.Add(dr["FIYAT"].ToString());
        Cesitler.Items[i].SubItems.Add(dr["ID"].ToString());
        i++;
    }
    dr.Close();
    conn.Dispose();
    conn.Close();
}
```

Şekil 85: Ürün Çeşitlerini ListView'e Getir Metodu

Veri tabanına eklenmiş olan ürün çeşitlerini listview'de göstermeye yarayan metottur. Parametre olarak kategori numarası verilir. Veri tabanından kategoriler tablosundan ürün adı ve fiyatı seçilir.

## 3. Ürün Arama Metodu

```
1 reference
public void getByProductSearch(ListView Cesitler, int txt)
{
    Cesitler.Items.Clear();
    SqlConnection conn = new SqlConnection(gnl.conString);
    SqlCommand comm = new SqlCommand("Select * from URUNLER where ID=@ID", conn);

    comm.Parameters.Add("@ID", SqlDbType.Int).Value = txt;
    if (conn.State == ConnectionState.Closed)
    {
        conn.Open();
    }
    SqlDataReader dr = comm.ExecuteReader();
    int i = 0;
    while (dr.Read())
    {
        Cesitler.Items.Add(dr["URUNAD"].ToString());
        Cesitler.Items[i].SubItems.Add(dr["FIYAT"].ToString());
        Cesitler.Items[i].SubItems.Add(dr["ID"].ToString());
        i++;
    }
    dr.Close();
    conn.Dispose();
    conn.Close();
}
```

Şekil 86: Ürün Arama Metodu

Ürünler tablosundan ürün numarasına göre arama yapan metottur. Bu metot parametre olarak ürün adını, fiyatını ve numarasını alır ve ilgili bilgileri listview içerisine getirir.



#### 4. Ürün Çeşitlerini ComboBox'a Getir Metodu

```
2 references
public void urunCesitleriniGetir(ComboBox cb)
{
    cb.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from KATEGORILER where DURUM=0", con);
    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            cUrunCesitleri uc = new cUrunCesitleri();
            uc._UrunTurNo = Convert.ToInt32(dr["ID"]);
            uc._KategoriAd = dr["KATEGORIADI"].ToString();
            uc._Aciklama = dr["ACIKLAMA"].ToString();
            cb.Items.Add(uc);
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 87: Ürün Çeşitlerini ComboBox'a Getir Metodu

Kategoriler tablosundan kategori numarasına göre arama yapan metottur. Bu metod parametre olarak kategori adını, açıklamasını ve numarasını alır ve ilgili bilgileri combobox içerisine getirir.

#### 5. Ürün Çeşitlerini ListView'e Getir Metodu

```
2 references
public void urunCesitleriniGetir(ListView lv)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from KATEGORILER where DURUM=0", con);
    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["ID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["KATEGORIADI"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ACIKLAMA"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 88: Ürün Çeşitlerini ListView'e Getir Metodu

Kategoriler tablosundan kategori numarasına göre ürün çeşitlerini listview'e getiren metottur. Bu metot parametre olarak kategori adını, açıklamasını ve numarasını alır ve ilgili bilgileri listview içerisine getirir.

## 6. Ürün Çeşitlerini Getir Metodu

```
1 reference
public void urunCesitleriniGetir(ListView lv, string source)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select * from KATEGORILER where DURUM=0 and KATEGORIADI like '%'+@source+'%'", con);
    cmd.Parameters.Add("@source", SqlDbType.VarChar).Value = source;
    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["ID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["KATEGORIADI"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ACIKLAMA"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 89: Ürün Çeşitlerini Getir Metodu

## 7. Ürün Kategori Ekle Metodu

```
1 reference
public int urunKategoriEkle(cUrunCesitleri uc)
{
    int sonuc = 0;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("insert into KATEGORILER (KATEGORIADI,ACIKLAMA) values (@KATEGORIADI,@ACIKLAMA)", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        cmd.Parameters.Add("@KATEGORIADI", SqlDbType.VarChar).Value = uc._kategoriAd;
        cmd.Parameters.Add("@ACIKLAMA", SqlDbType.VarChar).Value = uc._aciklama;
        sonuc = Convert.ToInt32(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 90: Ürün Kategori Ekle Metodu

Ürün kategorisi eklemek amacıyla yazılmıştır.

## 8. Ürün Kategori Güncelle Metodu

```
1 reference
public int urunKategoriGuncelle(urunCesitleri uc)
{
    int sonuc = 0;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Update KATEGORILER set KATEGORIADI=@KATEGORIADI,ACIKLAMA=@ACIKLAMA where ID=@KATID", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        cmd.Parameters.Add("@KATEGORIADI", SqlDbType.VarChar).Value = uc._KategoriAd;
        cmd.Parameters.Add("@ACIKLAMA", SqlDbType.VarChar).Value = uc._Aciklama;
        cmd.Parameters.Add("@KATID", SqlDbType.Int).Value = uc._UrunTurNo;
        sonuc = Convert.ToInt32(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 91: Ürün Kategori Güncelle Metodu

Seçilen ürün kategorisinin güncellenmesi amacıyla yazılmıştır. Parametre olarak kategori adı, açıklama ve kategori numarası gönderilmiştir.

## 9. Ürün Kategori Sil Metodu

```
2 references
public int urunKategoriSil(int id)
{
    int sonuc = 0;

    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Update KATEGORILER set DURUM=1 where ID=@KATID", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        cmd.Parameters.Add("@KATID", SqlDbType.Int).Value = id;
        sonuc = Convert.ToInt32(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 92: Ürün Kategori Sil Metodu

Seçilen ürün kategorisinin veri tabanından silinmesi amacıyla yazılmıştır.

## M. Ürünler Class

```
21 references
class cUrunler
{
    cGenel gnl = new cGenel();
    #region Fields
    private int _urunid;
    private int _urunturno;
    private string _urunad;
    private decimal _fiyat;
    private string _aciklama;
    #endregion
    #region Properties
    public int Urunid { get => _urunid; set => _urunid = value; }
    2 references
    public int Urunturno { get => _urunturno; set => _urunturno = value; }
    2 references
    public string Urunad { get => _urunad; set => _urunad = value; }
    2 references
    public decimal Fiyat { get => _fiyat; set => _fiyat = value; }
    2 references
    public string Aciklama { get => _aciklama; set => _aciklama = value; }
    #endregion
}
```

Şekil 93: Ürünler Class Fields ve Properties

Ürünler classımızda toplam beş adet fields bulunmaktadır. Bunlar ürün numarası, ürün kategori numarası, ürün adı, fiyat ve açıklamadır.

### 1. Ürünler Class Metotları

```
21 references
class cUrunler
{
    cGenel gnl = new cGenel();
    Fields
    Properties
    1 reference
    public void urunleriListeleByUrunAdi(ListView lv, string urunadi)...
    1 reference
    public int urunEkle(cUrunler u)...
    3 references
    public void urunleriListele(ListView lv)...
    1 reference
    public int urunGuncelle(cUrunler u)...
    1 reference
    public int urunSil(cUrunler u, int kat)...
    1 reference
    public void urunleriListeleByUrunAdi(ListView lv, int urunId)...
    1 reference
    public void urunleriListeleIstatisklereGore(ListView lv, DateTimePicker Baslangic, DateTimePicker Bitis)...
    1 reference
    public void urunleriListeleIstatisklereGoreUrunId(ListView lv, DateTimePicker Baslangic, DateTimePicker Bitis, int urunkatId)...
}
```

Şekil 94: Ürünler Class Metotları

Ürünler classımız içerisinde 8 adet metotumuz vardır. Bunlardan dört tanesi ürünleri silmek, güncellemek, listelemek ve ürün eklemek için kullanılmıştır. Kalan metotlar ise ürünleri farklı parametrelere istinaden listelemek amacıyla yazılmıştır. Bunlar sırasıyla; ürün adına göre, ürün numarasına göre, ürün tarihine göre ve ürün kategori numarasına göre.

## 2. Ürün Adına Göre Ürünleri Listele Metodu

```
1 reference
public void ürünleriListeleByÜrünAdi(ListView lv, string ürünadi)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select*from URUNLER where DURUM=0 and URUNAD like '&' + @ürünAdi + '%'", con);

    SqlDataReader dr = null;
    cmd.Parameters.Add("@ürünAdi", SqlDbType.VarChar).Value = ürünadi;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["ID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["KATEGORIID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["URUNAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["ACIKLAMA"].ToString());
            lv.Items[sayac].SubItems.Add(string.Format("{0:0#00.0}", dr["FIYAT"].ToString()));
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 95: Ürün Adına Göre Ürünleri Listele Metodu

Ürünler tablosundan tüm ürünleri ürün adına göre listeleyerek çağıran metottur.

## 3. Ürün Ekle Metodu

```
1 reference
public int ürünEkle(cÜrünler u)
{
    int sonuc = 0;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("insert into URUNLER (URUNAD,KATEGORIID,ACIKLAMA,FIYAT) values (@ürünAd,@katId,@aciklama,@fiyat)", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }

        cmd.Parameters.Add("@ürünAd", SqlDbType.VarChar).Value = u._ürünAd;
        cmd.Parameters.Add("@katId", SqlDbType.Int).Value = u._ürünTurnu;
        cmd.Parameters.Add("@aciklama", SqlDbType.VarChar).Value = u._aciklama;
        cmd.Parameters.Add("@fiyat", SqlDbType.Money).Value = u._fiyat;
        sonuc = Convert.ToInt32(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 96: Ürün Ekle Metodu

Ürünler tablomuza yeni bir ürün eklemek istediğimizde çağıracağım metottur. Parametre olarak ürün adı, kategori numarası, açıklaması ve fiyatı gönderilmiştir.

#### 4. Ürünleri ListView'e Ekle Metodu

```
3 references
public void urunleriListele(ListView lv)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select URUNLER.*, KATEGORIADI from URUNLER inner join KATEGORILER on KATEGORILER.ID=URUNLER.KATEGORIID where URUNLER.DURUM=0", con);
    SqlDataReader dr = null;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["ID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["KATEGORIID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["KATEGORIADI"].ToString());
            lv.Items[sayac].SubItems.Add(dr["URUNAD"].ToString());
            lv.Items[sayac].SubItems.Add(string.Format("{0:0000.00}", dr["FIYAT"].ToString()));
            lv.Items[sayac].SubItems.Add(dr["ACIKLAMA"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 97: Ürünleri ListView'e Ekle Metodu

Ürünleri listelemek için yazılmıştır. Ürünün kategori adı, id'si, fiyatı, açıklaması listview'e gönderilir.

#### 5. Ürün Güncelle Metodu

```
1 reference
public int urunGuncelle(urunler u)
{
    int sonuc = 0;
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("Update URUNLER set URUNAD=@urunad,KATEGORIID=@katID,ACIKLAMA=@aciklama,FIYAT=@fiyat where ID=@urunID", con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@urunad", SqlDbType.VarChar).Value = u._urunad;
        cmd.Parameters.Add("@katID", SqlDbType.Int).Value = u._urunturno;
        cmd.Parameters.Add("@aciklama", SqlDbType.VarChar).Value = u._aciklama;
        cmd.Parameters.Add("@fiyat", SqlDbType.Money).Value = u._fiyat;
        cmd.Parameters.Add("@urunID", SqlDbType.Int).Value = u._urunid;
        sonuc = Convert.ToInt32(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 98: Ürün Güncelle Metodu

Ürünler listesindeki ürünün adını, kategori adını, açıklamasını, fiyatını veri tabanında güncellemek amacıyla yazılmıştır. Try catch kod blokları kullanılarak hata yakalama amaçlanmıştır.

## 6. Ürün Sil Metodu

```
1 reference
public int urunSil(cUrunler u, int kat)
{
    int sonuc = 0;
    SqlConnection con = new SqlConnection(gnl.conString);

    string sql = "Update URUNLER set DURUM=1 WHERE ";
    if (kat == 0)
    {
        sql += "KATEGORIID=@urunID";
    }
    else
    {
        sql += "ID=@urunID";
    }
    SqlCommand cmd = new SqlCommand(sql, con);
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        cmd.Parameters.Add("@urunID", SqlDbType.Int).Value = u._urunid;
        sonuc = Convert.ToInt32(cmd.ExecuteNonQuery());
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        con.Dispose();
        con.Close();
    }
    return sonuc;
}
```

Şekil 99: Ürün Sil Metodu

Seçilen ürünün veri tabanından silinmesi amacıyla yazılmıştır.

## 7. Ürün Adına Göre Ürünleri Listele Metodu

```
1 reference
public void urunleriListeleByUrunAdi(ListView lv, int urunId)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select URUNLER.*, KATEGORIADI from URUNLER inner join KATEGORILER on KATEGORILER.ID=URUNLER.KATEGORIID where URUNLER.DURUM=0 and URUNLER.KATEGORIID=@urunID",
    con);

    SqlDataReader dr = null;
    cmd.Parameters.Add("@urunID", SqlDbType.Int).Value = urunId;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["ID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["KATEGORIID"].ToString());
            lv.Items[sayac].SubItems.Add(dr["KATEGORIADI"].ToString());
            lv.Items[sayac].SubItems.Add(dr["URUNAD"].ToString());
            lv.Items[sayac].SubItems.Add(string.Format("{0:000.00}", dr["FIYAT"].ToString()));
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 100: Ürün Adına Göre Ürünleri Listele Metodu

Ürünlerin, ürün adına göre listelenmesi amacıyla yazılmıştır.

## 8. Ürünleri İstatistiklere Göre Listele Metodu

```
1 reference
public void urunleriIstistiklereGore(ListView lv, DateTimePicker Baslangic, DateTimePicker Bitis)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select top 10 URUNLER.URUNAD,SUM(SATISLAR.ADET) as adeti from KATEGORILER inner join URUNLER on KATEGORILER.ID=URUNLER.KATEGORIID inner join SATISLAR on URUNLER.ID=SATISLAR.URUNID inner join ADISYON on SATISLAR.ADISYONID=ADISYON.ID where (CONVERT(datetime,TARIH,104) between convert (datetime,@Baslangic ,104) and Convert(datetime,@Bitis,104)) group by URUNLER.URUNAD order by adeti desc", con);

    SqlDataReader dr = null;
    cmd.Parameters.Add("@Baslangic", SqlDbType.VarChar).Value = Baslangic.Value.ToShortDateString();
    cmd.Parameters.Add("@Bitis", SqlDbType.VarChar).Value = Bitis.Value.ToShortDateString();
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["URUNAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["adeti"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

Şekil 101: Ürünleri İstatistiklere Göre Listele Metodu

Ürünleri belirtilen tarihler arasında ürün adetine göre sıralama yaparak listelenmesi amacıyla yazılmıştır.

## 9. Ürün ID'ye Göre Ürün İstatistiklerini Listele Metodu

```
1 reference
public void urunleriIstistiklereGoreUrunId(ListView lv, DateTimePicker Baslangic, DateTimePicker Bitis, int urunkatid)
{
    lv.Items.Clear();
    SqlConnection con = new SqlConnection(gnl.conString);
    SqlCommand cmd = new SqlCommand("select top 10 URUNLER.URUNAD,SUM(SATISLAR.ADET) as adeti from KATEGORILER inner join URUNLER on KATEGORILER.ID=URUNLER.KATEGORIID inner join SATISLAR on URUNLER.ID=SATISLAR.URUNID inner join ADISYON on SATISLAR.ADISYONID=ADISYON.ID where (CONVERT(datetime,TARIH,104) between convert (datetime,@Baslangic ,104) and Convert(datetime,@Bitis,104)) and(URUNLER.KATEGORIID=@katid) group by URUNLER.URUNAD order by adeti desc", con);
    SqlDataReader dr = null;
    cmd.Parameters.Add("@Baslangic", SqlDbType.VarChar).Value = Baslangic.Value.ToShortDateString();
    cmd.Parameters.Add("@Bitis", SqlDbType.VarChar).Value = Bitis.Value.ToShortDateString();
    cmd.Parameters.Add("@katid", SqlDbType.Int).Value = urunkatid;
    try
    {
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        dr = cmd.ExecuteReader();
        int sayac = 0;
        while (dr.Read())
        {
            lv.Items.Add(dr["URUNAD"].ToString());
            lv.Items[sayac].SubItems.Add(dr["adeti"].ToString());
            sayac++;
        }
    }
    catch (SqlException ex)
    {
        string hata = ex.Message;
    }
    finally
    {
        dr.Close();
        con.Dispose();
        con.Close();
    }
}
```

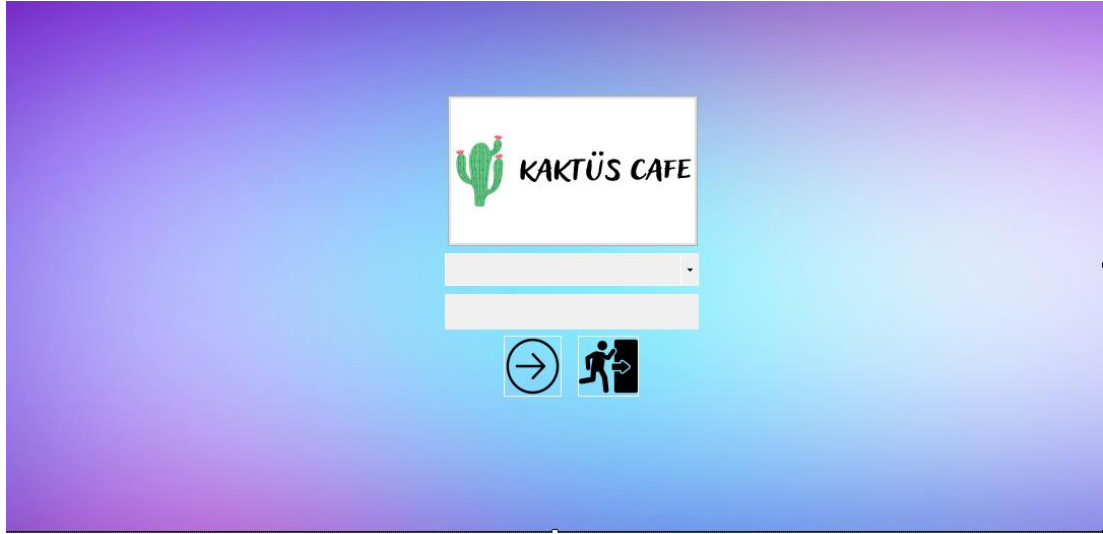
Şekil 102: Ürün ID'ye Göre Ürün İstatistiklerini Listele Metodu

Başlangıç ve bitiş tarihleri girilen ve kategori numarası seçilen ürünleri istatistiksel olarak listelemek amacıyla yazılmıştır.



## IV. WINDOWS FORMS

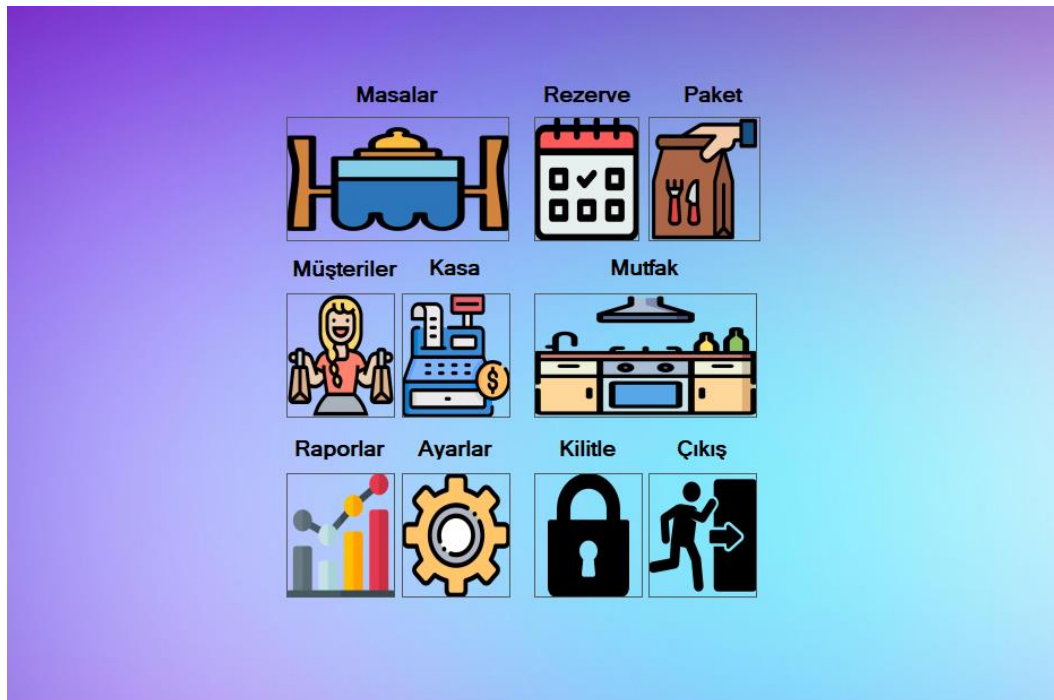
### A. Giriş Formu



Şekil 103: Form Giriş Ekranı

Otomasyon ilk açıldığında bizi karşılayan ekrandır. Bu ekrandan önceden veri tabanına kaydedilmiş kullanıcı adı ve şifreleriyle giriş yapılabilmektedir. Giriş ve çıkış butonları ile oluşturulmuştur. Kullanıcı adının girildiği alan combobox olarak tanımlanmıştır, şifreyi ise textboxa yazılarak alınmaktadır.

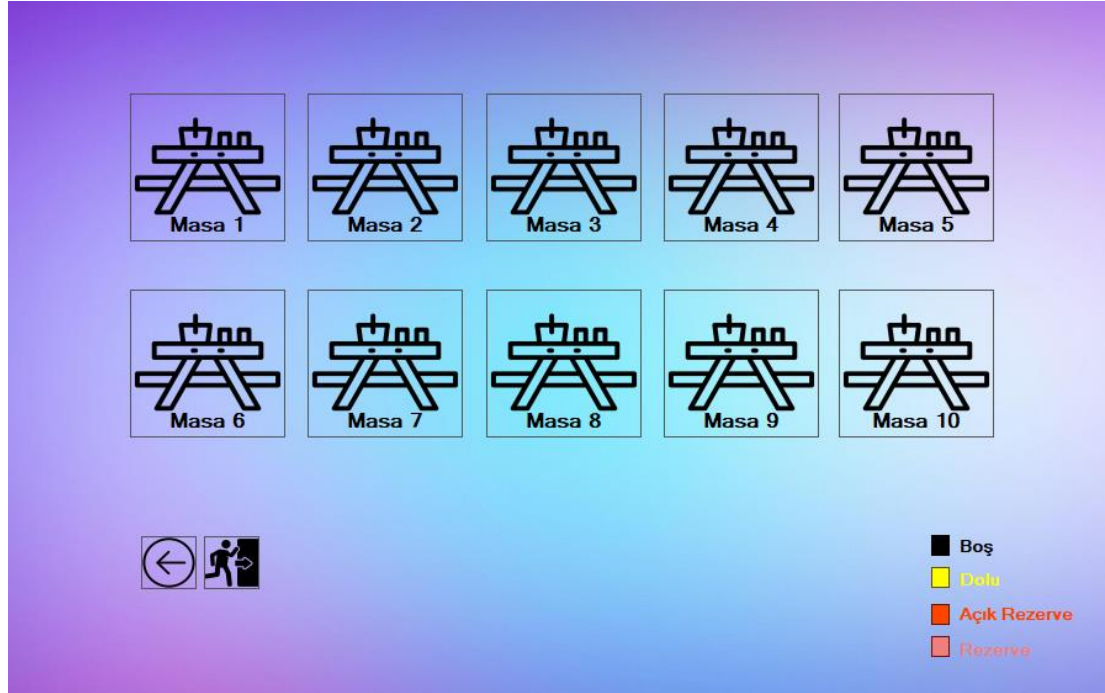
### B. Menü Formu



Şekil 104: Form Menü Ekranı

Giriş yapıldıktan sonra menü ekranı açılmaktadır. Bu ekrandan tüm formlara erişim sağlanmaktadır. Menü ekranında; masalar, rezerve, kasa işlemleri, paket, mutfak, müşteriler, raporlar, ayarlar, kilitle, çıkış ekranları yer almaktadır. Button içerisine image alınarak oluşturulmuştur. Arka planı property kısmından back ground image ile şekillendirilmiştir.

### C. Masalar Formu



Şekil 105: Form Masalar Ekranı

Menüden masalar butonuna tıkladığımızda açılan formdur. Toplam on masa bulunmaktadır. Masalar button içerisine image alınarak oluşturulmuştur. Toplam dört masa durumu vardır. Bunlar boş, dolu, açık rezerve ve rezerve masadır. Her forma eklediğimiz geri ve çıkış butonları formun sol alt köşesinde bulunmaktadır. Bunlar da buton içerisine image alınarak yapılmıştır.

#### D. Rezervasyon Formu

MÜŞTERİ		TELEFON		ADRES	
Ad	Soyad	Telef...	Adres	Email	Must...

Yeni Müşteri Rezervasyon Aç Rezervasyonlar Müşteri Güncelle Geri Dön

Şekil 106: Form Rezervasyon Ekranı

Menü ekranından rezerve butonuna tıklandığında açılan sayfadır. Bu form üzerinden tarih, masa ve kişi sayısı seçilir, girilmek istenen not yazılır, sağ tarafta bulunan müşteriler listesinden müşteri seçilir ve rezervasyon aç butonuna tıklanarak rezervasyon oluşturulur. Eğer müşteri sisteme kayıtlı değil ise yeni müşteri butonuna tıklanarak müşteri oluşturulur. Müşterinin bilgileri yanlış ise müşteri güncelle butonundan seçilen müşterinin bilgileri güncellenebilir. Rezervasyonlar menüsünden tüm rezervasyonlar görülmektedir. Sağ üstte bulunan textboxlar ile müşteri telefona göre ve adrese göre arama işlemi yapılabilir. Bu formda label, textbox, listview, combobox, datetimepicker kullanılmıştır.

## E. Sipariş Formu

MASA 1 Ürün No

Menü

Ana yemek İçecek

Tatlılar Salatalar

Fast food Çorbalar

Spagetti Ara Sıcak

Menü

Ürün Adı	Ürün Fiyatı

Siparişler

Ürün Adı	Ürün Adeti	Ürün Fiyatı

1 2 3

4 5 6

7 8 9

C 0

Sipariş İptal

Ödeme Sipariş

Şekil 107: Form Sipariş Ekranı

Masalar ekranından herhangi bir masaya tıklandığında açılan form ekranıdır. Bu form üzerinden müşteriden aldığımız siparişler girilmektedir. Sekiz adet default olarak eklediğimiz kategoriler bulunmaktadır. Bu kategoriler; ana yemek, içecek, tatlılar, salatalar, fast food, çorbalar, spagetti, ara sıcaktır. Kategoriler içerisinde default olarak veri tabanından eklediğimiz ürünleri barındırmaktadır. Her bir kategoriye ait ürünler ayrı ayrı eklenmiştir. Kategorilerden herhangi birinin üzerine tıklandığında üst tabloda ürünler listelenmektedir. Bu listeden seçilen ürünler siparişler tablosuna aktarılmaktadır. Yanlış seçilen ürün için çift tıklama ile siparişler listesinden silinebilmektedir. Ürün birden fazla olacak ise sağ tarafta bulunan rakamlar ile kaç adet sipariş geçileceği tuşlanarak siparişler ekranına aktarılabilir. Sipariş iptal butonu siparişleri iptal eder, sipariş butonu masanın durumunu dolu yaparak yeni bir adisyon açar ve siparişleri bu adisyona kaydeder. Ödeme butonu ise ödeme ekranına aktararak ödeme işlemlerinin yapılmasını sağlar. Bu formda; label, textbox, listview, button ve group box özellikleri kullanılmıştır.

## F. Mutfak Formu

Ürün Ekle    Ürün Kategori Ekle

Gıda Kategorisi :

Gıda Adı :

Gıda Fiyatı :

Kategori Adı :

Açıklama :

Aramak İstediğiniz Ürün :

Ekle    Değiştir    Bul    Sil

Kategori	Urun Id	UrunT...	Kateg...	Urun ...	Fiyatı

Şekil 108: Form Mutfak Ekranı

Mutfak formundan yeni bir kategori açılabilir veya var olan bir kategoriye ürün ekleme işlemi yapılabilir. En üstte bulunan seçeneklerden biri işaretlenerek diğer işlemlerin görünürlüğü ortadan kaldırılır. Ürün ekle seçili ise; kategori seçilerek gıda adı ve fiyatı belirtilerek ekleme işlemi yapılabilir. Alt kısımda bulunan listview üzerinden ürünler listelenerek ürün silme veya değiştirme işlemi yapılabilir. Bul butonu içerisinde adisyon bulunarak ilgili adisyona ait ürünler listeleme işlemi yapılabilir. Aynı şekilde kategori kısmı seçilerek, aynı özellikler kullanılabilir. Bu form üzerinde label, button, listview, radio button araçları kullanılmıştır.

## G. Kasa İşlemleri Formu

Şekil 109: Form Kasa İşlemleri

Menü ekranından kasa işlemleri tıklandığında açılan ekrandır. Bu form üzerinden günlük bazlı dediğimiz z raporu ve aylık raporlama yapılabilir. Aynı zamanda bu ekranda yapılan raporlamalar yazdırılabilir özellik taşımaktadır. Bu formda button, label ve report viewer araçları kullanılmıştır.

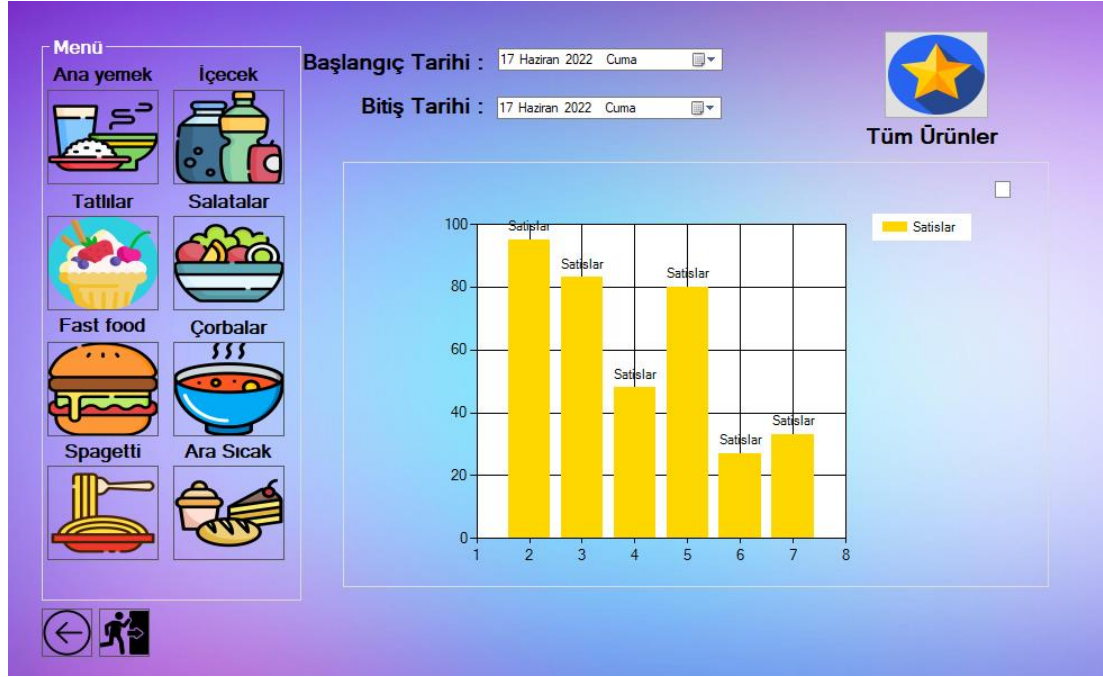
## H. Müşteri Formu

Şekil 110: Form Müşteri Ekranı



Bu form üzerinden müşteri ile ilgili işlemler yapılabilmektedir. Müşteri ekleme, güncelleme, müşteriye ait adisyon bulma, siparişler ekranını açma gibi özellikler eklenmiştir. Üst kısımda bulunan textboxlar ile ad ve soyada, telefona göre müşteri arama işlemleri yapılabilmektedir. Label, textbox, listview, button araçları kullanılmıştır.

## İ. Raporlar Formu



Şekil 111: Form Raporlar Ekranı

Kategori ve tarih baz alınarak raporlama yaptığımız ekrandır. Kategori ve belirtilen tarih aralıkları alınarak alt kısımda bulunan chart alanında ilgili grafik gösterilmektedir. Aynı zamanda tüm ürünler de bu ekranda raporlanabilmektedir. Kullanılan araçlar; label, button, date time picker, chart, group box, listview.

## J. Ayarlar Formu

Giriş yapan

Yeni Şifre

T Yeni Şifre

Değiştir

Adı

Soyadı

Şifre

Şifre Tekrar

Görevi

Yeni

Kaydet

Sil

Değiştir

Görevi	Adı	Soyadı

Şekil 112: Form Ayarlar Ekranı

Ayarlar formu kullanıcıların şifrelerini güncelleyebildiği, yeni kullanıcılar eklenebildiği formdur. Üst kısımda bulunan giriş yapan isimli label giriş yapanın mevkini, adını ve soyadını gösterir. Atanan yetkilere istinaden kullanıcılara ekleme, silme ve değiştirme butonları aktif edilir. Alt kısımda bulunan listview aracına görevlilerin tümü listelenir. Listedenden görevli seçilerek güncelleme işlemi yapılabilir. Bu formda label, button, textbox, listview, combobox ve groupbox araçları kullanılmıştır.



## K. Kilit Formu



Şekil 113: Form Kilit Ekranı

Kilit formu kullanıcıların ekrandan uzaklaştığında kullanabileceği bir özelliktir. Bu ekran menüden kilitle butonuna tıklandığında açılmaktadır. Giriş butonuna tıklandığında tekrardan otomasyon giriş ekranını açmaktadır. Buradan tekrardan kullanıcı ve şifre girilerek menü formuna ulaşılabilir.

## L. Ödeme Formu

The image shows a payment form with a purple header and a light blue background. The header has "Siparişler" and "ID" in red. Below the header is a table with columns: "Ürün Adı", "Adet", "U..", "Fiyat", and "Sat...". To the right of the table are two boxes: "Özel İndirim" with a checkbox labeled "İndirim Uygula" and "Özel Aktivite" with a text input field. Below the table is a box labeled "Ödeme Bilgileri" containing a table with rows: "İndirim", "KDV", "Fiyat", and "Ara Toplam", each with a value of "0" and the unit "TL". To the right of this box is a box labeled "Ödeme Türü" with radio buttons for "Nakit", "Kredi Kartı", and "Ticket". To the right of these boxes are two icons: a red card with a gold coin labeled "Hesap Özeti" and a calculator with a dollar sign labeled "Hesap Kapat". At the bottom left are two icons: a left arrow and a person walking.

Şekil 114: Form Ödeme Ekranı

Ödeme formu siparişler ekranından ödeme butonuna tıklanmasıyla açılmaktadır. Bu ekranda müşterinin verdiği siparişlere istinaden ödeme işleminin yapılması işlemlerini gerçekleştirmektedir. İstenirse indirim uygulanmaktadır ve ödeme indirimli olarak yapılabilmektedir. Üst kısımda kırmızı ile işaretlenen ID kısmında müşterinin adisyon ID'si tutulmaktadır. Bu adisyondaki ürünler alt kısımdaki listview'de listelenmektedir. Ödeme türü kısmında radio buttonlar kullanılmıştır ve ödemenin hangi tür ile yapılacağı seçilmektedir. Hesap özeti ile yazdırılabilir bir dosya alabilmekteyiz. Hesabı kapat butonuna tıklandığında ilgili adisyon ve satış kapatılmış olur, masa durumu boş masa olarak güncellenmektedir. Label, button, listview, checkbox, group box ve button araçları kullanılarak oluşturulmuş bir formdur.

## V. KAYNAKÇA

Visual Studio'ya Giriş. 2022. [www.geeksforgeeks.org](http://www.geeksforgeeks.org). Erişim Tarihi: Mayıs 2022

<https://www.geeksforgeeks.org/introduction-to-visual-studio/>

Visual Studio Nedir? Bileşenleri Nelerdir? Erişim Tarihi: Mayıs 2022

[https://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio)

.Net Framework Nedir? Erişim Tarihi: Mayıs 2022

[https://en.wikipedia.org/wiki/.NET\\_Framework](https://en.wikipedia.org/wiki/.NET_Framework)

.Net Framework Mimarisi Erişim Tarihi: Mayıs 2022

<https://www.geeksforgeeks.org/introduction-to-net-framework/>

ADO.NET nedir Erişim Tarihi: Mayıs 2022

<https://stackoverflow.com/questions/2471198/what-is-ado-net>

ADO.NET ' e Genel Bakış Erişim Tarihi: Mayıs 2022

<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview>

ADO. NET Mimarisi Erişim Tarihi: Mayıs 2022

<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-architecture>

Windows Forms Erişim Tarihi: Haziran 2022

<https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-6.0>

Windows Forms Nedir Erişim Tarihi: Haziran 2022

[https://en.wikipedia.org/wiki/Windows\\_Forms](https://en.wikipedia.org/wiki/Windows_Forms)

Windows Forms Açık Kaynak Kodları Erişim Tarihi: Haziran 2022

<https://github.com/dotnet/winforms>