

**DOKUZ EYLÜL ÜNİVERSİTESİ  
İZMİR MESLEK YÜSEKOKULU  
TEKNİK PROGRAMLAR BÖLÜMÜ  
BİLGİSAYAR PROGRAMCILIĞI PROGRAMI  
2018-2019 ÖĞRETİM YILI BİTİRME PROJESİ**

**DSD  
OTEL YÖNETİM PROGRAMI**



**2017707018 Mustafa DOĞAN 2017707002 Melih Can AKGÜNEŞ 2017707048 Emrah TUNÇ**

**Proje Danışmanı  
Öğr. Gör. İsmail DÜŞMEZ**

**2018-2019, İZMİR**

## **İÇİNDEKİLER**

Değerlendirme Kriterleri Tablosu .....	4
Önsöz .....	5
Analiz Bilgileri .....	6-18
- Analiz Detayları .....	6
- Kullanılan Programlama Dili ve Veritabanı .....	6
- Görev Dağılımları .....	6
- Rakip Programların Değerlendirmesi .....	7
- Proje Setup Dosyası Hazırlanışı ve Kuruluşu .....	9
- Projenin Proje SWOT Analizi .....	12
- Rakip SWOT Analizi .....	13
- Çalışma Takvimi .....	14
- Projenin GANTT Şeması .....	18
Akış Diyagramı .....	19-29
Veritabanı Normalizasyonu .....	30-36
Entity Relationship Diyagramı .....	37
Program Kodları .....	38-249
Login.cs .....	38
Admin.cs .....	42
OtelArayüz.cs .....	44
Rezervasyon.cs .....	60
R_musteri_ekle.cs .....	66
Rez_sil.cs .....	80
RezRapor.cs .....	82
Resepsiyon.cs .....	95
Odaicerik.cs .....	117
R_konak.cs .....	118

Rez_tikla.cs .....	142
Hatirlatmaekle.cs .....	147
Profiller.cs .....	149
Konaklamalar.cs .....	159
OdaSiparis.cs .....	171
UrunEkle.cs .....	186
KatHizmetleri.cs .....	191
Oda_Durum.cs .....	199
YoneticiPanel.cs .....	205
Personel.cs .....	208
Odalar.cs .....	216
Istatistikler.cs .....	228
Log.cs .....	248
Örnek Yazıcı Çıktıları .....	250-261
Sonsöz .....	262
Notlar .....	263

## DEĞERLENDİRME KRİTERLERİ TABLOSU

<b>Adı Soyadı</b>				
<b>Okul No</b>				
Proje Konusu		10 Puan	5 Puan	0 Puan
		Evet	Kısmen	Hayır
1	Projeyi sunmak için gerekli kişisel hazırlığını yapmış mı? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Setup/Install dosyası hazırlanmış mı? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Programı Install ettiği bilgisayarda çalıştırabildi mi? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	Projenin analizi süresince uzman biri ile çalışılmış mı, Adını verdi mi? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	Analiz ile ilgili kısa bilgi verdi mi? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Projede veri tabanı kullanılmış mı? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Öğrenci; Projeyi sunmak için yeteri bilgiye sahip mi? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Öğrenci; Açık ve anlaşılır bir sunuş yaptı mı? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	Öğrenci; Projede yazılmış olan kodlara hâkim mi? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Proje konusuna ilişkin ticari piyasada benzeri programları araştırmış mı? <input type="checkbox"/> EVET <input type="checkbox"/> HAYIR	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PUAN HESAP		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
TOPLAM PUAN				
Sunuyu Yapan Öğrenci ve Çalışması hakkında Yorum:				

## ÖNSÖZ

Zincir otellerin kullandığı geniş kapsamlı oteller için yapılmış olan programlar ve butik oteller yani tek kişinin yönettiği otellerin kullandığı programlar olarak 2 ayrı başlıkta inceleyecek olursak bizim yapmış olduğumuz program şuan ki haliyle butik otellerin kullanacağı düzeydedir.

Bu butik oteller sizin programınızı neden kullansın sorusunu duyar gibiyiz. Cevap olarak;

Çünkü Butik otellerin gelişmiş programlarda kullanmayacağı bir çok özellik dolu( otelinin bulunduğu yerdeki havaalanından uçakların kalkış geliş saatini çeken, diğer otellerde oda fiyatları değişmişse odalara ek ücret veya bilgilendirme mesela yandaki otelin şu tip teki odanın fiyatı bu oldu şu kadar zarar ediyoruz gibi mesajlar vermesi.) veya bu programları alacak bütçeye sahip degillerdir.

Bununla beraber butik otellere yönelik programların incelediğimiz kadarıyla kullanıcı dostluğu açısından yetersiz ve tasarımsal eksikliklerinin bulunduğu gördük.

Bizim ulaşmaya çalıştığımız amaç ise programımız kullanıcı dostu ve estetik bir arayüz ile beraber sade bir kullanım ama istenildiğinde de kullanıcıya detaylı veriler verebilicek yapıya getirmektir.

Son olarak projemizin yapımında ve gelişmesinde bize katkılarını esirgemeyen ek olarak danışmanımız olan İbis HOTEL Genel Müdürü Murat AKIN ‘ a , İzmir Meslek Yüksekokulu Öğretim Görevlisi Özer KESTANE ‘ ye , İzmir Meslek Yüksekokulu Öğretim Görevlisi İsmail DÜŞMEZ ‘ e ve Murat AKIN ‘ a ulaşmamızı sağlayan Ege Turistik İşletmeler ve Konaklamalar Birliği(EPİK) Başkan Asistanı Leman Hanım ‘ a sonsuz minnet ve teşekkürlerimizi sunarız.

# **ANALİZ BİLGİLERİ**

## **Programın Adı**

DSD OTEL YÖNETİM PROGRAMI

## **Hizmet Ettiği Sektör**

Otelcilik ve Konaklama Sektörü

## **Kullanılan Programlama Dili ve Veritabanı**

Kullanılan Programa Dili : C#

Kullanılan Framework : Bunifu Framework, DevExpress Framework

Kullanılan Veri Tabanı : Microsoft SQL Server

## **Projeye Dahil Olan Öğrenciler ve Bölümleri**

Numarası : 2017707018      Adı ve Soyadı : Mustafa DOĞAN

- Rezervasyon
- Raporlar
- Otel Yönetim

Numarası : 2017707002      Adı ve Soyadı : Melih Can AKGÜNEŞ

- Resepsiyon
- Oda Siparis
- Kat Hizmetleri
- Otel Yönetim

Numarası : 2017707048      Adı ve Soyadı : Emrah TUNC

- Resepsiyon
- Profiller
- Giriş

## **Piyasadaki Yazmış Olduğunuz Programa Alternatif Programlar**

### **Adonis Otel Yönetim Programı**

- Üretim Yılı : 2004
- Versiyon Numaraları : 5.2.354
- Program Adı : Adonis Otel Yönetim Programı
- Üretici Firma Adı ve Adresi : 3D Bilişim Teknolojileri A.Ş / Esenyurt-İstanbul

### **Roommate Otel Yönetim Programı**

- Üretim Yılı : 2014
- Versiyon Numaraları :
- Program Adı : Roommate Otel Yönetim Programı
- Üretici Firma Adı ve Adresi : İleri Teknoloji Yazılım ve Bilişim Hizmetleri / Beyoğlu-İstanbul

### **Basit Otel Yönetim Programı**

- Üretim Yılı : 2019
- Versiyon Numaraları :
- Program Adı : Basit Otel Yönetim Programı
- Üretici Firma Adı ve Adresi : SaaS Otel Yönetim Bilişim Sistemleri A.Ş / Çankaya – İzmir

## **Alternatif Programların Eksiklikleri ve Fazlalıkları**

### **Roommate Otel Yönetim Programı**

- Çoklu Tesis Yönetimi ile birden fazla tesise sahip kullanıcılar tüm tesislerini aynı anda aynı ekran üzerinde yönetebilir.
- Muhasebe sistemleri ile entegre değildir.
- Santral entegrasyonu bulunmamakta.

### **Adonis Otel Yönetim Programı**

- Ana ekranda tüm odaların bilgilerinin görüntülenmesi misafirlerin gizlilik haklarına aykırıdır.
- Muhasebe sistemleri ile entegre değildir.
- Santral entegrasyonu bulunmamakta.
- Bulut tabanlı değildir, kurulum gereklidir.
- Tüm cihazlarda kullanılabilir uyumluluğu bulunmamakta.
- Grafikler yüksek değerlerde taşıma yaşamakta ve değerler üst üste binmektedir.
- Web Paketi ile tesise ait entegre bir web sitesi oluşturulabilir.
- Mobil Otel Yönetimi ile tesis cep telefonu üzerinden takip edilebilir.

- Polis ve Jandarma sistemlerine bilgiler otomatik olarak gönderilir.
- Kullanımı pratik ve anlaşılır.
- Departman bazında ana sayfalara ve sekmelere sahiptir.
- Odalara arıza bilgisi girilebilir.
- Yönetici panelinde çeşitli istatistikler için grafikler sergilenebilir.
- Her işlem log olarak tutulur ve yönetici bile silemeyecek şekilde kaydedilir.
- Mesaj gönderimi bulunmaktadır.
- Web paketi ile tesise ait entegre bir web sitesi oluşturulabilir.
- Rezervasyon motoru ile web siteleri Adonis'e bağlanabilir.
- Kanal yönetimi modülü ile farklı kanallarda tesis yönetilebilir.
- SambaPOS restoran yönetimi ile entegrasyonu bulunmaktadır.
- Mobil otel yönetimi ile tesisin cep telefonu üzerinden takip edebilirsiniz.
- Otomatik EGM rapor gönderimi özelliği bulunmaktadır.

#### Basit Otel Yönetim Programı

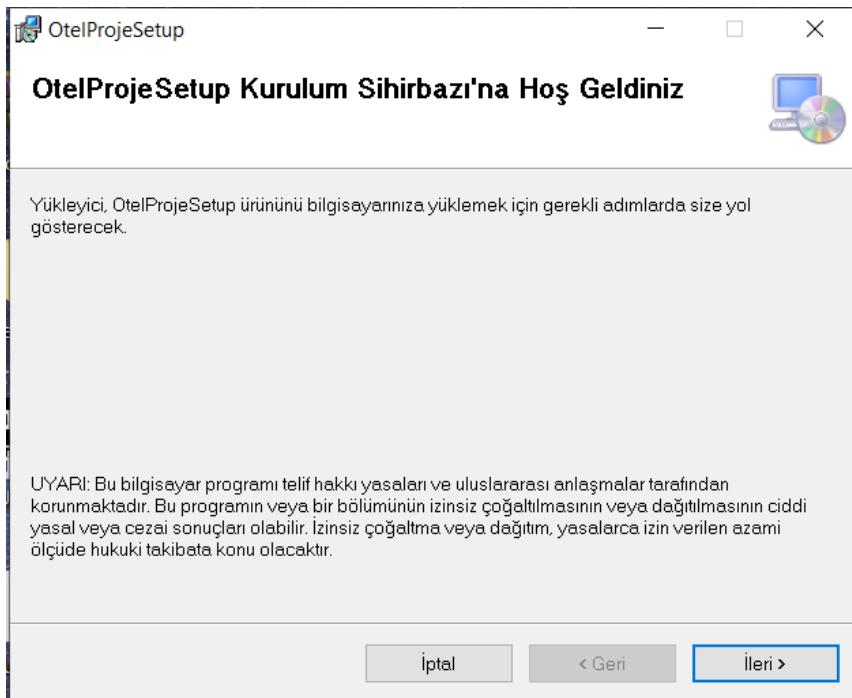
- Polis ve Jandarma sistemlerine bilgiler otomatik olarak gönderilir.
- Kanal Yönetim Sistemi ile bulunmamakta
- Bulut tabanlıdır, kurulum gerekmeyen.
- Tüm cihazlarda kullanılabilir uyumluluktadır.

#### Analiz Aşamasında Görüşülen Kişiler

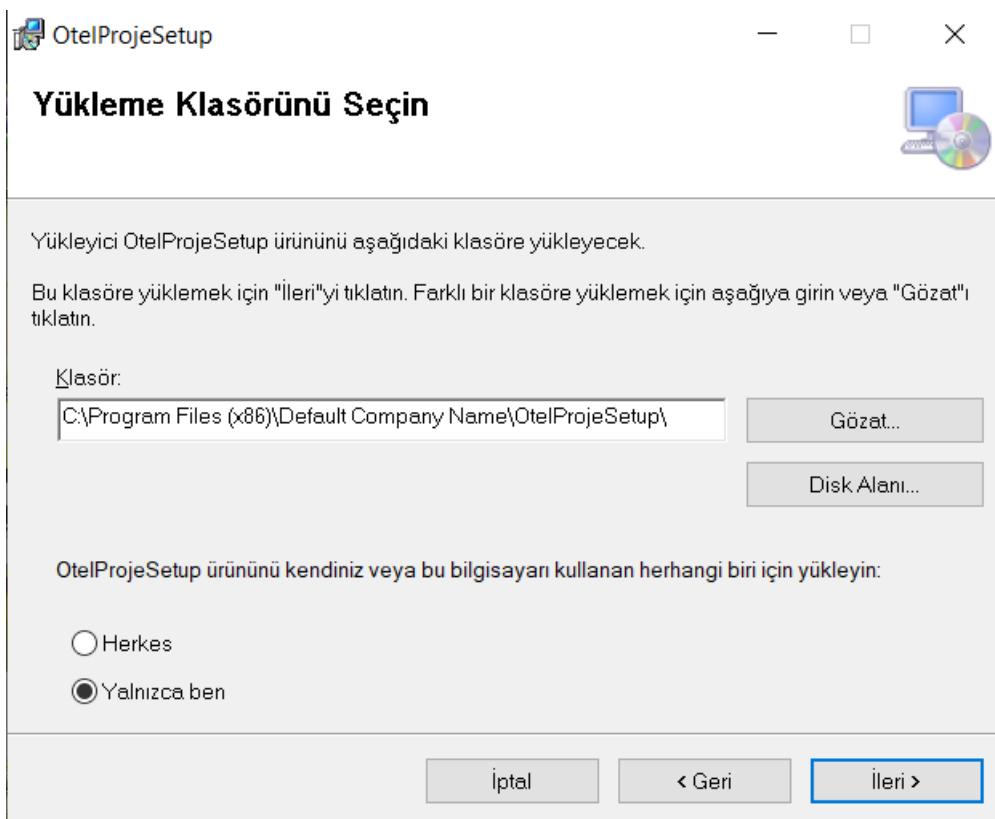
- Unvan : Müdür
- Adı ve Soyadı : Murat AKIN
- Çalıştığı Kurum : Alsancak İbis Hotel
- Görevi : Müdür
- Telefon numarası : 0532 709 00 10

## Proje Setup Dosyası Hazırlanışı ve Kuruluşu

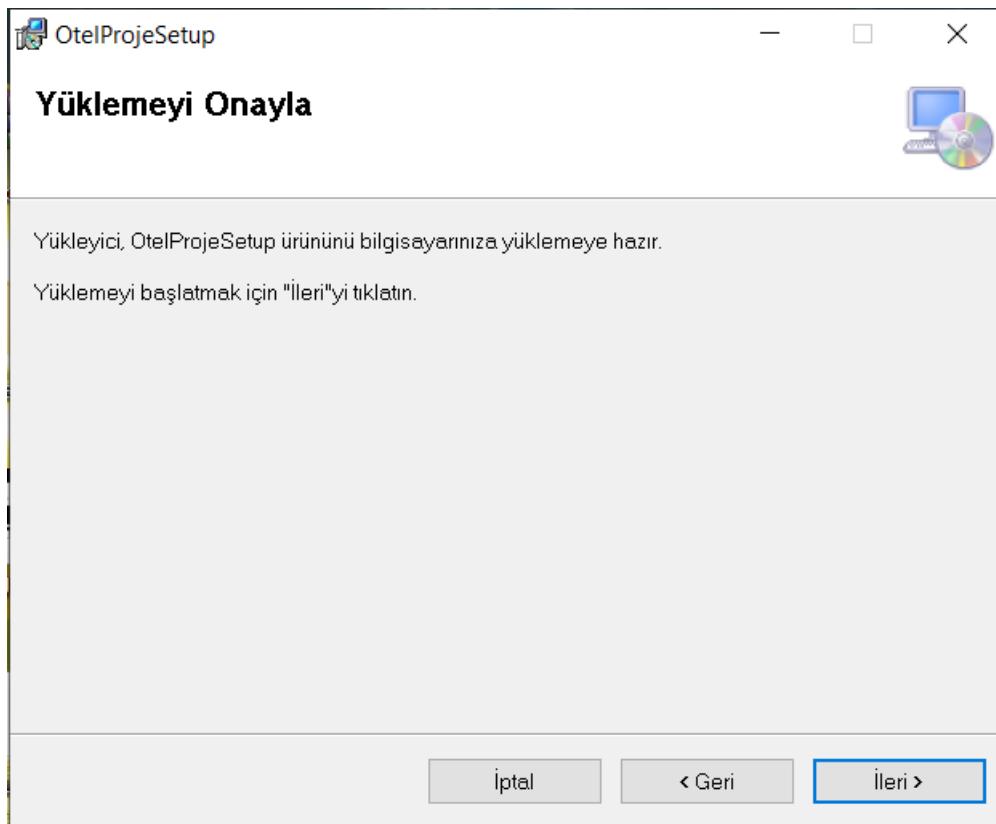
Setup ekran ile başlıyor aşağıda belirli uyarılarımız var. Kullanıcı ileri butonuna basarak ilerler



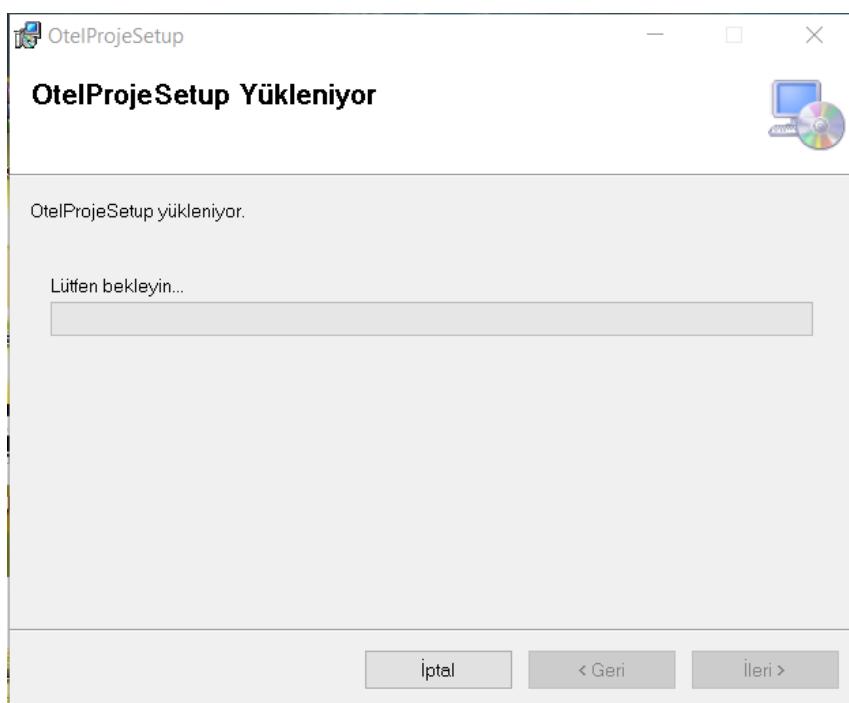
Burada kullanıcı yükleyeceği alanı seçebilir.



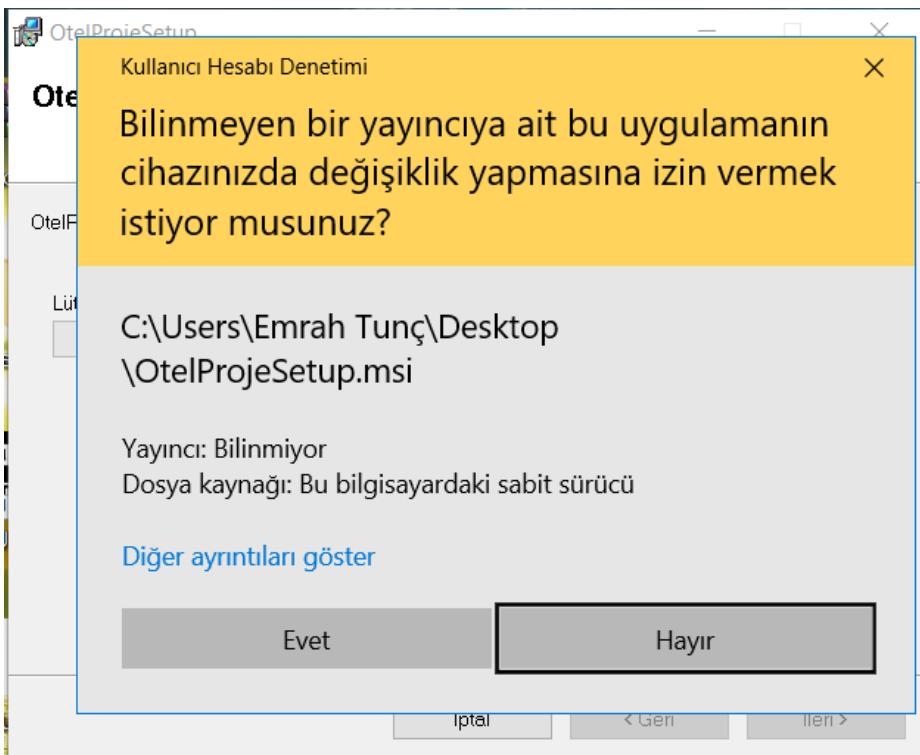
İleri diyerek programı yüklemeye izin veririz.



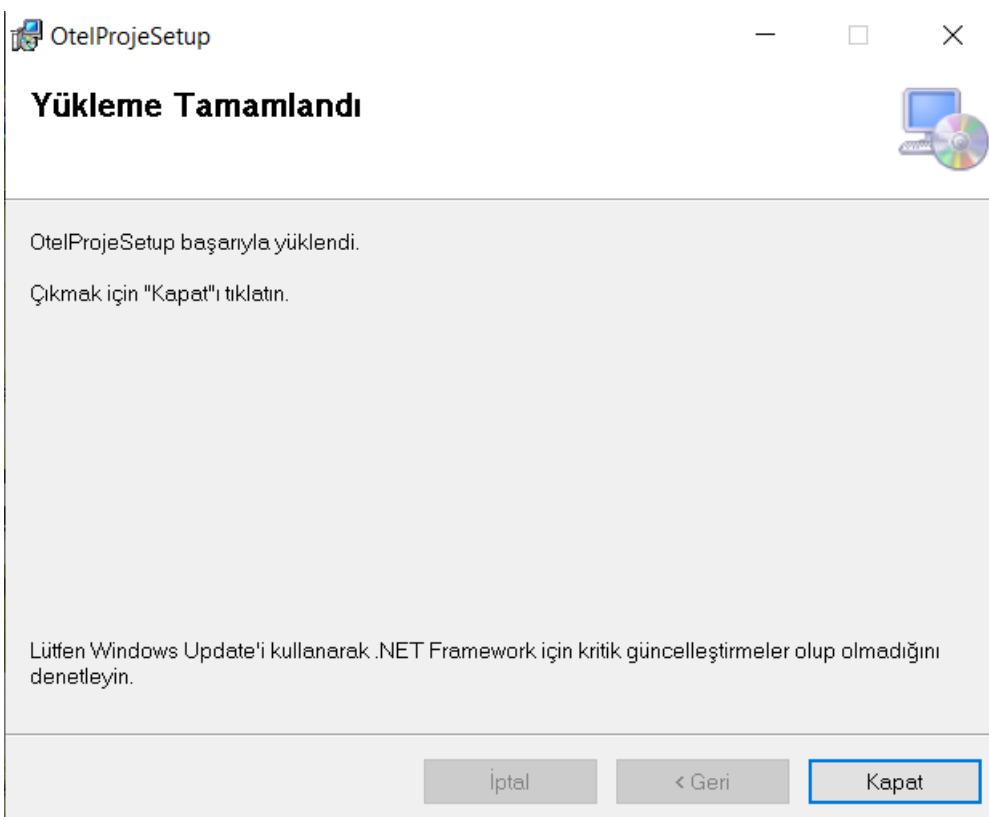
Programımız yüklenmeye başlar.



Yükleme sırasında kullanıcıdan izin ister, Evet diyip devam edilir.



Son olarak kurulum tamamlanır ve masaüstünde programımızın kısayolu oluşur.



## Projenin Swot Analizi

<b>Strengths</b>	<b>Weaknesses</b>	<b>Opportunities</b>	<b>Threats</b>
<ul style="list-style-type: none"><li>• Kapasitesine göre açılış sonrası hızlidir.</li><li>• Tasarımsal olarak üstündür.</li><li>• Kullanımı pratik ve anlaşılırıdır.</li><li>• Pop-up benzeri anlık açılır pencerelerle hızlı erişime sahiptir.</li><li>• Buton değişimleri ile kontrol etme düşüncesini rahatlatır.</li><li>• Departman bazında ana sayfalara ve sekmelere sahiptir.</li><li>• Her kattaki oda sayısı özelleştirilebilinir.</li><li>• Oda sipariş ve istekleri sırası ile listelenir.</li><li>• Odalar görevliler tarafından yetkileri içerisinde sadece oda durumları görülebilinir.</li><li>• Odalara arıza bilgisi girilebilinir.</li><li>• Yönetici panelinde çeşitli istatistikler için renkli ve anlaşılır grafikler sergilenebilir.</li><li>• Çalışanlar ve yetkileri yönetilir.</li><li>• Her işlem log olarak tutulur ve yönetici bile silemeyecek şekilde kaydedilir.</li><li>• Mesaj gönderimi bulunmaktadır.</li></ul>	<ul style="list-style-type: none"><li>• Yüksek kapasiteli otellerde açılış 2-3 saniye uzayabilir.</li><li>• Ana ekranın tüm odaların bilgilerinin görüntülenmemesi bazı durumlarda dezavantaj olabilir.(A)</li><li>• Çoklu tesis yönetimi bulunmamakta.(R)</li><li>• Kanal yönetim sistemi ve rezervasyon motoru bulunmamakta.(A,S)</li><li>• Mobil otel yönetimi bulunmamakta.(A)</li><li>• Muhasebe sistemleri ile entegre değildir.(S)</li><li>• EGM sistemi ile entegre değildir. (A,S)</li><li>• Santral entegrasyonu bulunmamakta(S)</li><li>• Bulut tabanlı değildir, kurulum gereklidir.(B)</li><li>• Tüm cihazlarda kullanılabilir uyumluluğu bulunmamakta.(B)</li></ul>	<ul style="list-style-type: none"><li>• Yeni frameworklar kullanıyoruz ve kullanıcılaraya iyi bir arayüz sunuyoruz.</li><li>• Butik otel programlarının genel olarak standart yapısı olması.</li><li>• Gelişmiş programların yüksek maaliyetleri.</li><li>• Sektörde uzman danışmanımızdan eleştiriler almamız.</li><li>• Öğrenci olmamız ve yüksek maddi gelir kazanma bekłentisinde olmamamız.</li><li>• Küçük butik otellerin bile sistemlerini programlara taşımaları.</li></ul>	<ul style="list-style-type: none"><li>• Piyasada hali hazırda çok fazla otel yönetim otomasyonları bulunması ve hala geliştirilmeye devam edilmeleri.</li><li>• Bazı otel yönetim sistemlerin bulut sisteme geçmeleri.</li><li>• Yapılış süresinin kısıtlı olması.</li><li>• Ana server olarak kullanacak ayrı cihazımızın bulunmaması.</li><li>• Bazı frameworkleri tam sürüm kullanacak durumda olmamamız.</li><li>• Herhangi bir destek ve tanınmışlığımızın bulunmaması.</li></ul>

## Rakip Swot Analizi

<b>Strengths</b>	<b>Weaknesses</b>	<b>Opportunities</b>	<b>Threats</b>
<ul style="list-style-type: none"><li>Kullanımı pratik ve anlaşılırıdır.</li><li>Departman bazında ana sayfalar ve sekmelere sahiptir.</li><li>Odalara ariza bilgisi girilebilinir.</li><li>Yönetici panelinde çeşitli istatistikler için grafikler sergilendirir.</li><li>Her işlem log olarak tutulur ve yönetici bile silemeyecek şekilde kaydedilir.</li><li>Mesaj gönderimi bulunmaktadır.</li><li>Web paketi ile tesise ait entegre bir web sitesi oluşturulabilir.</li><li>Rezervasyon motoru ile web siteleri Adonis'e bağlanabilinir.</li><li>Kanal yönetimi modülü ile farklı kanllarda tesis yönetilebilinir.</li><li>SambaPOS restoran yönetimi ile entegrasyonu bulunmaktadır.</li><li>Mobil otel yönetimi ile tesisin cep telefonu üzerinden takip edebilirsiniz.</li><li>Otomatik EGM rapor gönderimi özelliği bulunmaktadır.</li></ul>	<ul style="list-style-type: none"><li>Ana ekranada tüm odaların bilgilerinin görüntülenmesi misafirlerin gizlilik haklarına aykırıdır.</li><li>Muhasebe sistemleri ile entegre değildir.</li><li>Santral entegrasyonu bulunmamakta.</li><li>Bulut tabanlı değildir, kurulum gereklidir.</li><li>Tüm cihazlarda kullanılabilir uyumluluğu bulunmamakta.</li><li>Grafikler yüksek değerlerde taşıma yapamakta ve değerler üst üste binmektedir.</li></ul>	<ul style="list-style-type: none"><li>Hali hazırda sektörde yol almış olması.</li><li>Butik otel programlarının genel olarak standart yapısı olması.</li><li>Gelişmiş programların yüksek maaliyetleri.</li><li>Web paketi ile tesislerin kendi sitesini yapabilmesinin küçük tesisler için cezbedici olması.</li><li>Otellerin kendi sitelerini direkt sisteme entegre edebilmelerinin verdiği kolaylık.</li><li>Küçük butik otellerin bile sistemlerini programlara taşımaları.</li></ul>	<ul style="list-style-type: none"><li>Piyasada hali hazırda çok fazla otel yönetim otomasyonları bulunmaktadır ve hala geliştirilmeye devam edilmeleri.</li><li>Bazı otel yönetim sistemlerin bulut sistemine geçmeleri.</li></ul>

## **Çalışma Takvimi**

### **27.02.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Programın taslak özellikleri, menüler, menüler içerisindeki alt başlıklar belirlendi.

Eklenebilir ekstra fonksiyonlar tartışıldı.

### **29.02.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Programımız için bize yardım edebilecek otel araştırmaları yapıldı. 4 farklı otel için görüşme hazırlığı yapıldı. Ek olarak programın işleyışı, ne yapacağımız, nasıl bir yol inceleyeceğiz bu konular hakkında konuşuldu.

### **01.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Üst maddede belirlenen 4 otele gidildi. Gidilen otellerden olumsuz cevap alındı, bunun üzerine 15+ farklı otele mail atıldı ve hiçbir geri dönüş alınamadı.

### **03.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Projenin mevcut taslağı düşünülerek veri tabanı yapısı hazırlandı.

### **05.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Proje veri tabanının taslağı göz önüne alınarak muhtemel senaryolar düşünüldü ve veri tabanı, projenin mevcut haline göre tamamlandı.

### **07.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Olumsuz geri dönüşler sonucu İsmail DÜŞMEZ hocamıza danışık ve tavsiyeler aldık.

Projenin taslak görüntülerinin, mevcut amaç ve planlanan özelliklerinin anlatıldığı bir dosya hazırladık.

### **11.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Ege Turistik İşletmeler ve Konaklamalar Birligine görüşmeye gidildi.

Sekreter Leman Hanıma projemiz anlatıldı ve bir danışman istendi. Bunun sonucunda 13.03.2019 da Alsancak İbis Hotel'e bir randevu ayarlandı.

### **13.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Alsancak İbis Hotel'e gidildi ve Otel müdürü Murat Akın Bey danışmanımız olmayı kabul etti. Projenin ilk taslakları ve yapısı üzerine konusuldu, çeşitli özellik eklemeleri ve eksiltmeleri not alındı. Proje hedefi birtakım değişimelere maruz kaldı.

### **18.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Program taslağı görüşmeler sonucunda yeniden oluşturuldu. Planlanan frameworklar projeye dahil edilmeye başlandı.

### **19.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Arayüz tasarımını yapılmaya başlandı, ana sayfa yapısı hazırlandı.

### **26.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Ana sayfa genel olarak tamamlandı ve rezervasyon, müşteri ekleme formları hazırlandı.

### **27.03.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Konaklama, oda durumu, oda siparişi ve oda temizlik formları hazırlandı.

### **01.04.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Ana sayfanın yapısı düzenlenendi ve tasarımı geliştirildi. Ana sayfada gözükecek alt formlar child form olarak entegre edildi. Odaların içeriklerini gösteren bir pop-up bilgilendirme özelliği tasarlandı.

### **03.04.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Yönetici paneli taslağı tasarlandı ve bir sonraki danışman görüşmesi için dosya hazırlandı.

### **04.04.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Danışman ile görüşmeye gidildi ve bilgisayar ortamında projenin mevcut durumu gösterildi.

Yapılan görüşme sonucunda iyileştirme notları alındı.

## **08.04.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Alınan notlar doğrultusunda oda durumu formu ayrılarak ayrı formlar olarak yeniden tasarlandı.

## **11.04.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Danışmanın tavsiyeleri doğrultusunda konaklama formunda bazı değişiklikler yapıldı ve profil formu tasarlandı.

## **20.04.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Profil formu tamamlandı ve genel işleyişte iyileştirmeler yapıldı.Daha sonra görev dağılımı yapıldı.

## **27.04.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Görev dağılımından sonra nelerin yapıldığı , bilgi alışverişi ve proje gidişatı hakkında konuşuldu.

## **03.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Yapılan formlar birleştirildi ve iyileştirmeler yapıldı.

## **04.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Resepsiyon kısmında iyileştirme yapıldı.

## **05.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Resepsiyon kısmında iyileştirme yapıldı.

## **06.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Resepsiyon kısmında iyileştirme yapıldı.

## **07.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Resepsiyon kısmında iyileştirme yapıldı.

**08.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : Resepsiyon kısmında iyileştirme yapıldı.

**09.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : Resepsiyon kısmında iyileştirme yapıldı.

**10.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : Resepsiyon kısmında iyileştirme yapıldı.

**11.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : R\_Konak formunda iyileştirme yapıldı.

**12.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : R\_Konak formunda iyileştirme yapıldı.

**13.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : R\_Konak formunda iyileştirme yapıldı.

**14.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : R\_Konak formunda iyileştirme yapıldı.

**15.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : Otel Arayüz formu tasarlandı , yan menü ve bu menülere ait formlar oluşturuldu.Daha Sonra görev dağılımı yapıldı.

**23.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : Yapılan formlar birbirine entegre edildi ve hata düzeltmeleri yapıldı.

**26.05.2019**

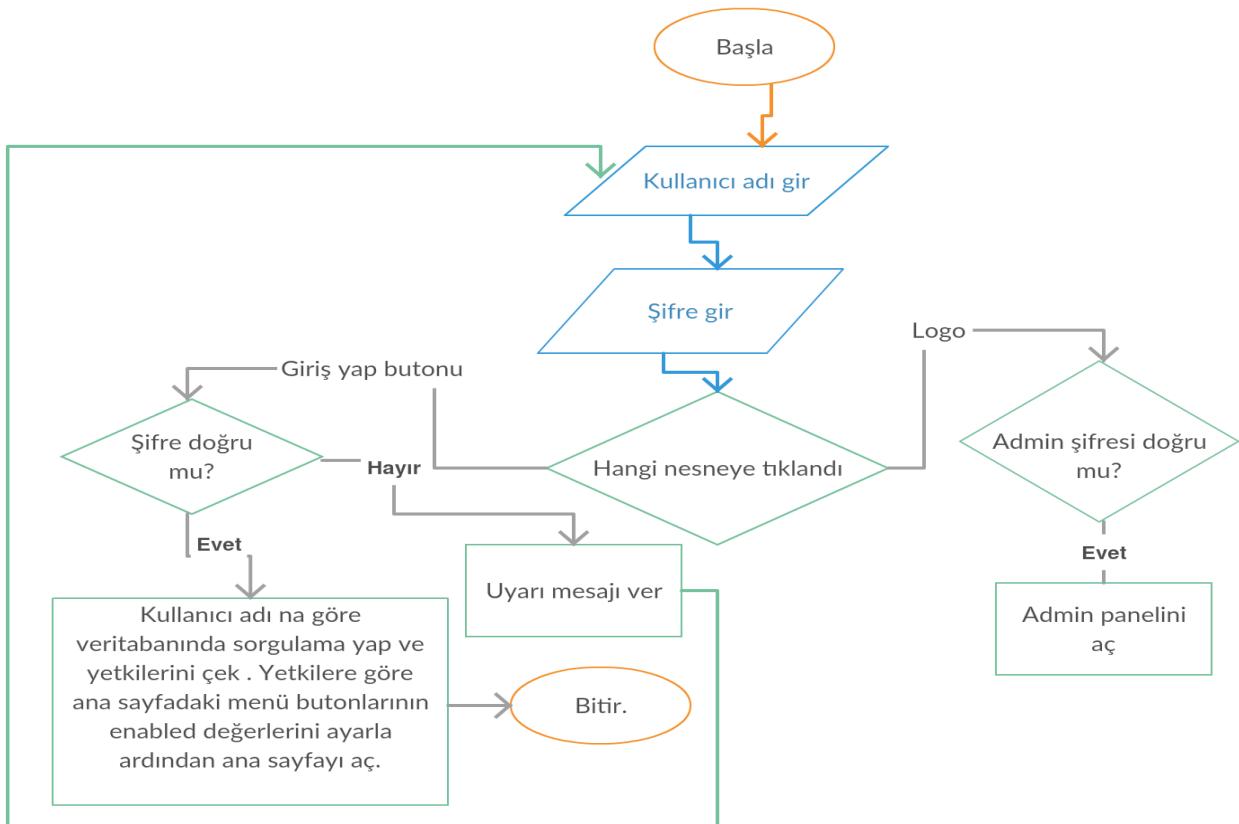
Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ  
Yapılanlar : Programa veriler girildi. Sunum için her şey tamamlandı.

## Proje GANTT Şeması

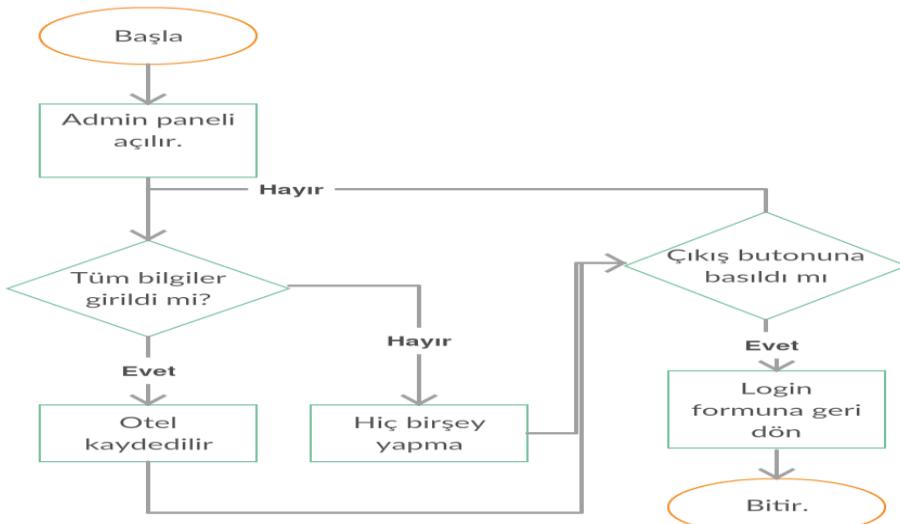
Aylar	Şubat		Mart					Nisan					Mayıs				
Haftalar	1.Hafta	2.Hafta	3.Hafta	4.Hafta	5.Hafta	6.Hafta	7.Hafta	8.Hafta	9.Hafta	10.Hafta	11.Hafta	12.Hafta	13.Hafta	14.Hafta			
Araştırma																	
Danışman																	
Taslak																	
Veritabanı																	
Arayüz																	
Kodlama																	
Hata Giderme																	
Değerlendirme																	
Proje Teslim																	

## AKIŞ DİYAGRAMI

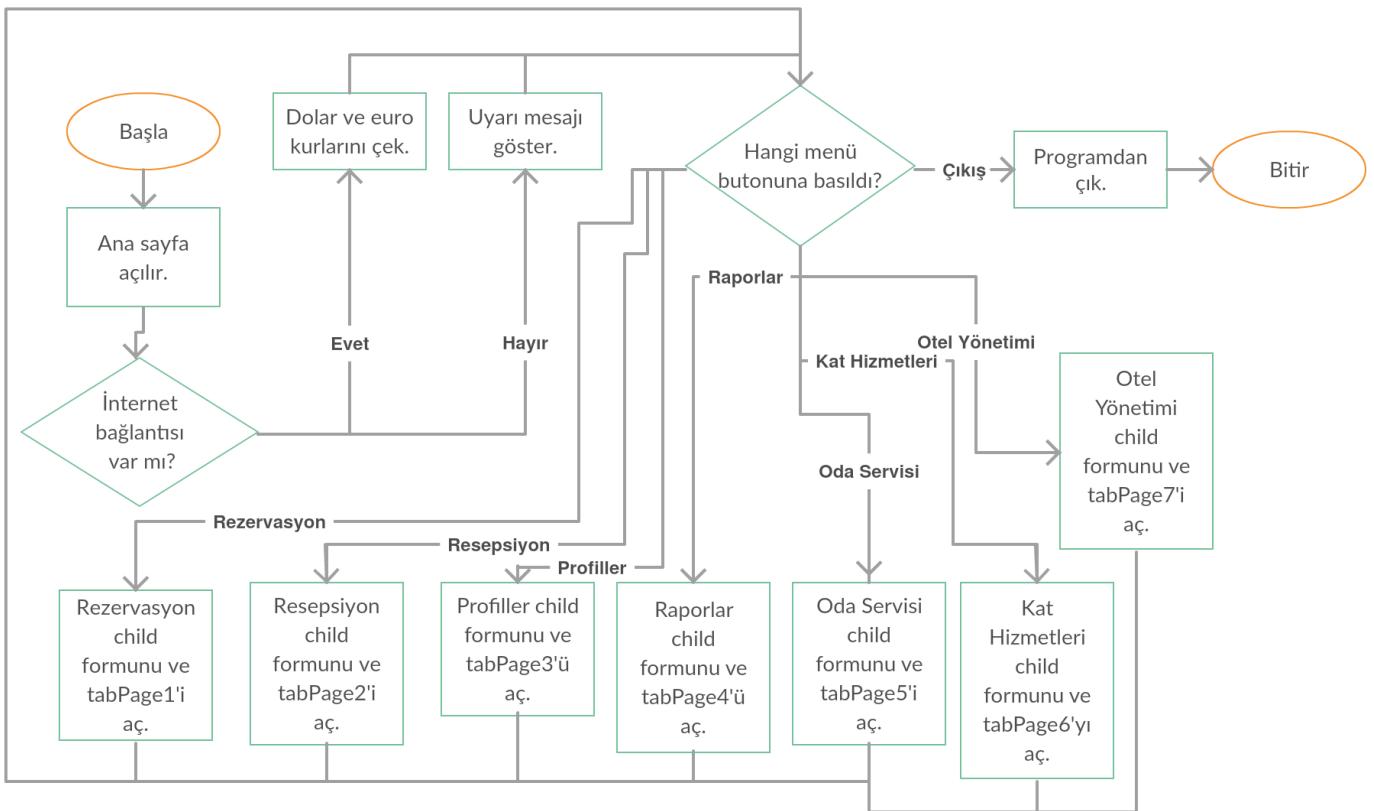
### LOGİN.CS



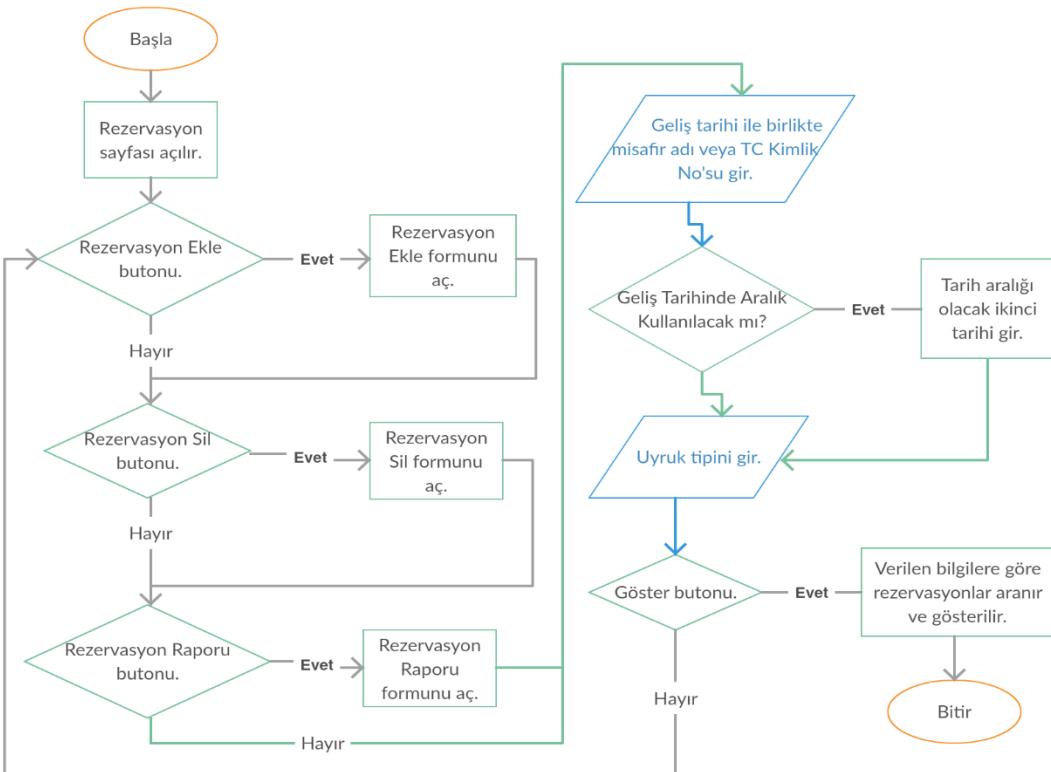
### ADMİN.CS



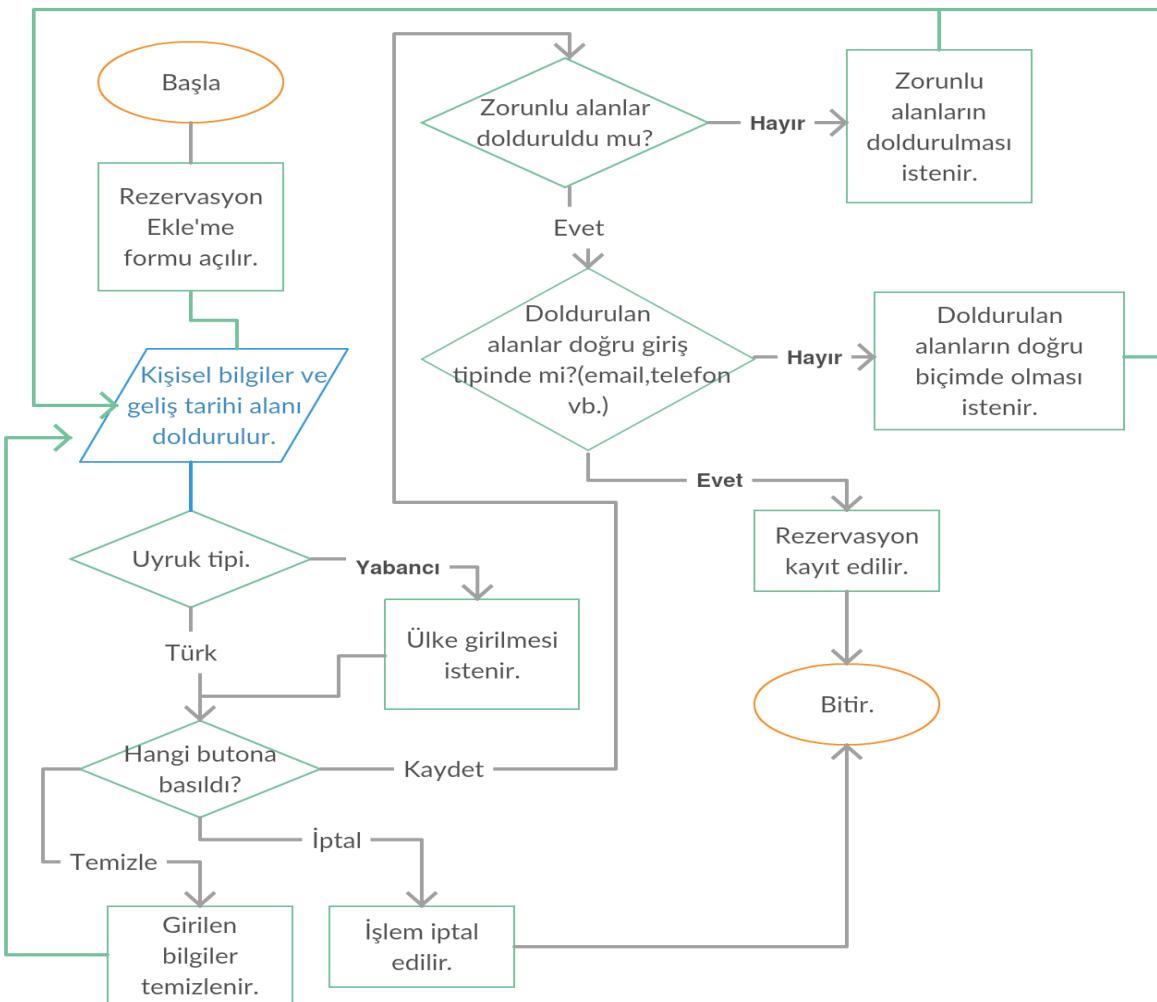
## OTELARAYÜZ.CS



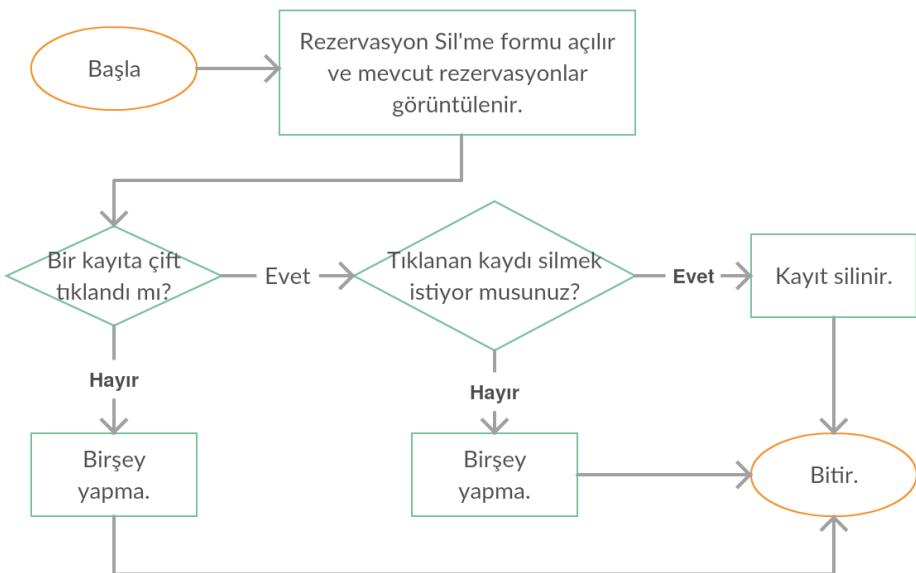
## REZERVASYON.CS



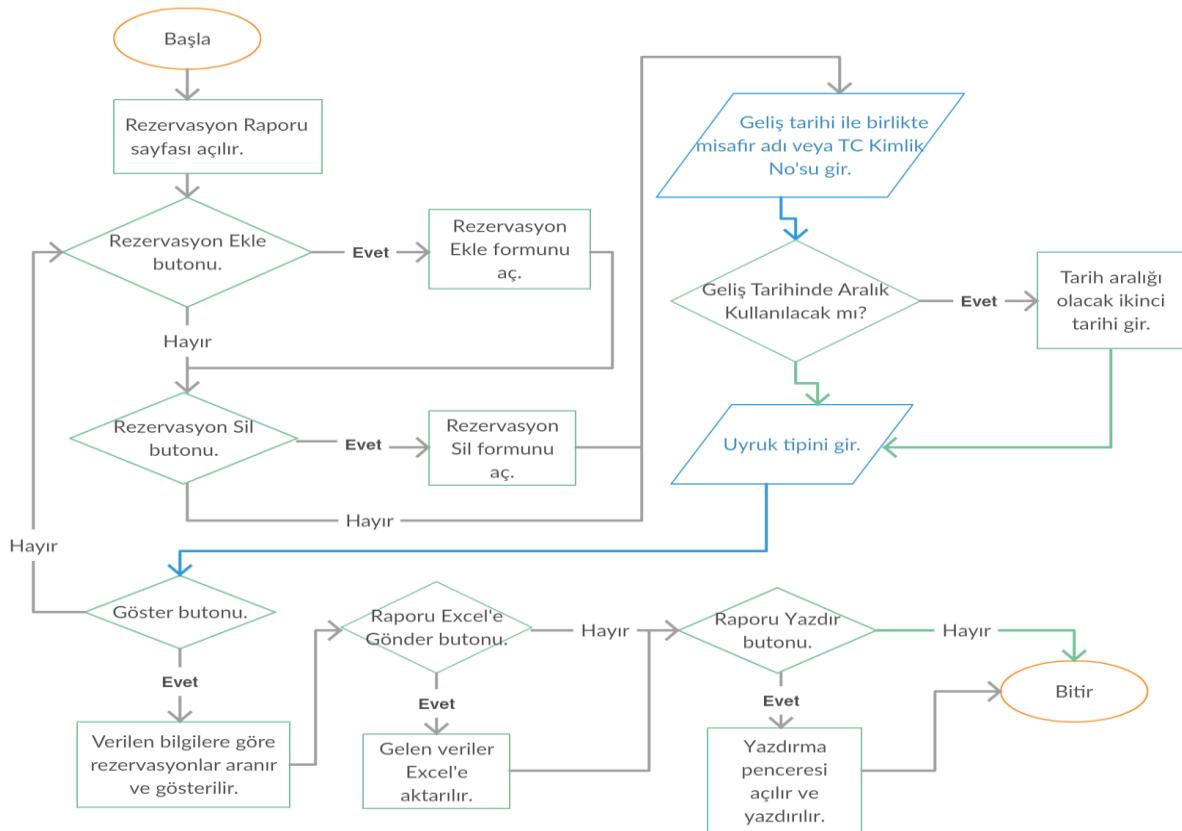
## R\_MUSTERİ\_EKLE.cs



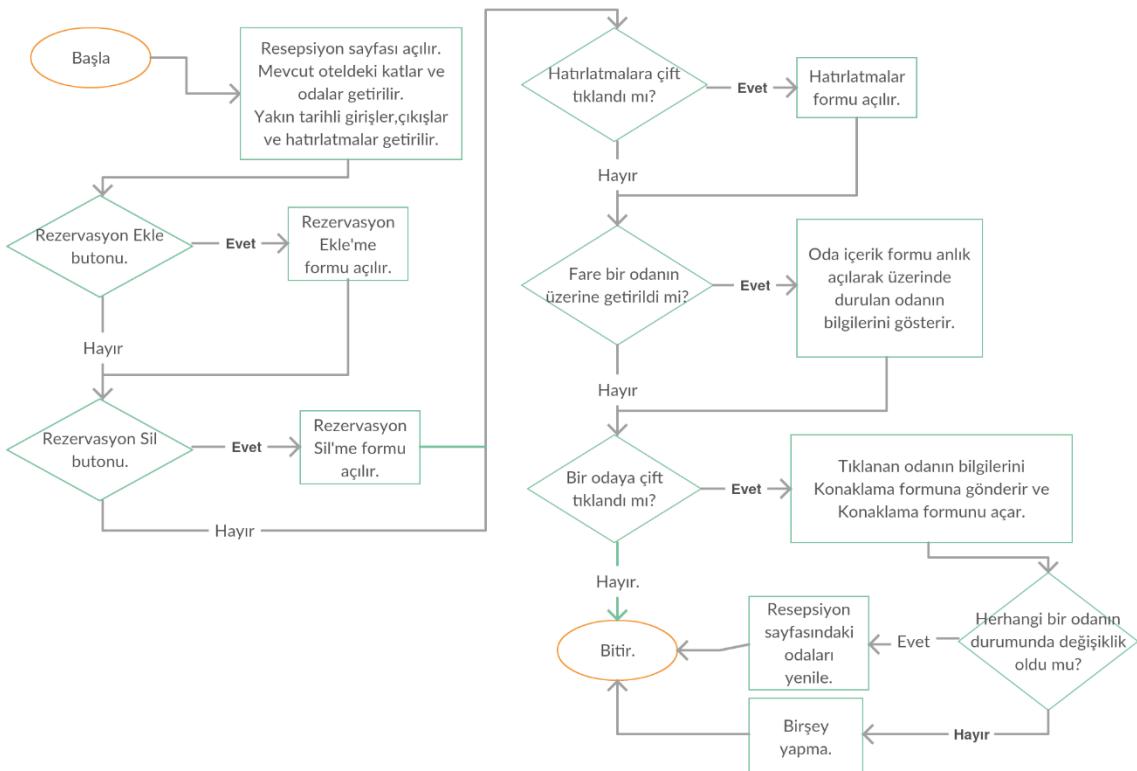
## REZ\_SİL.cs



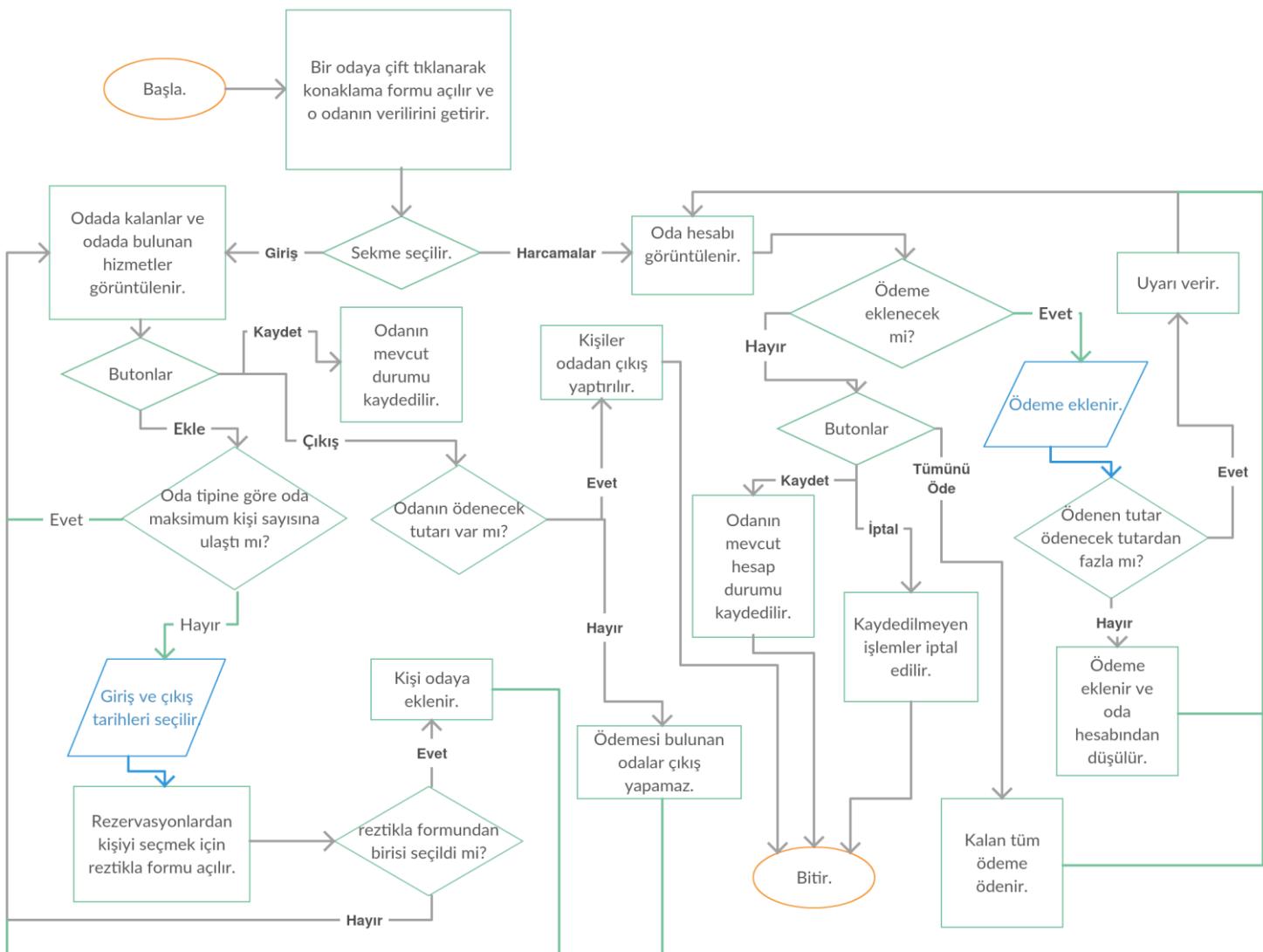
## REZRAPOR.CS



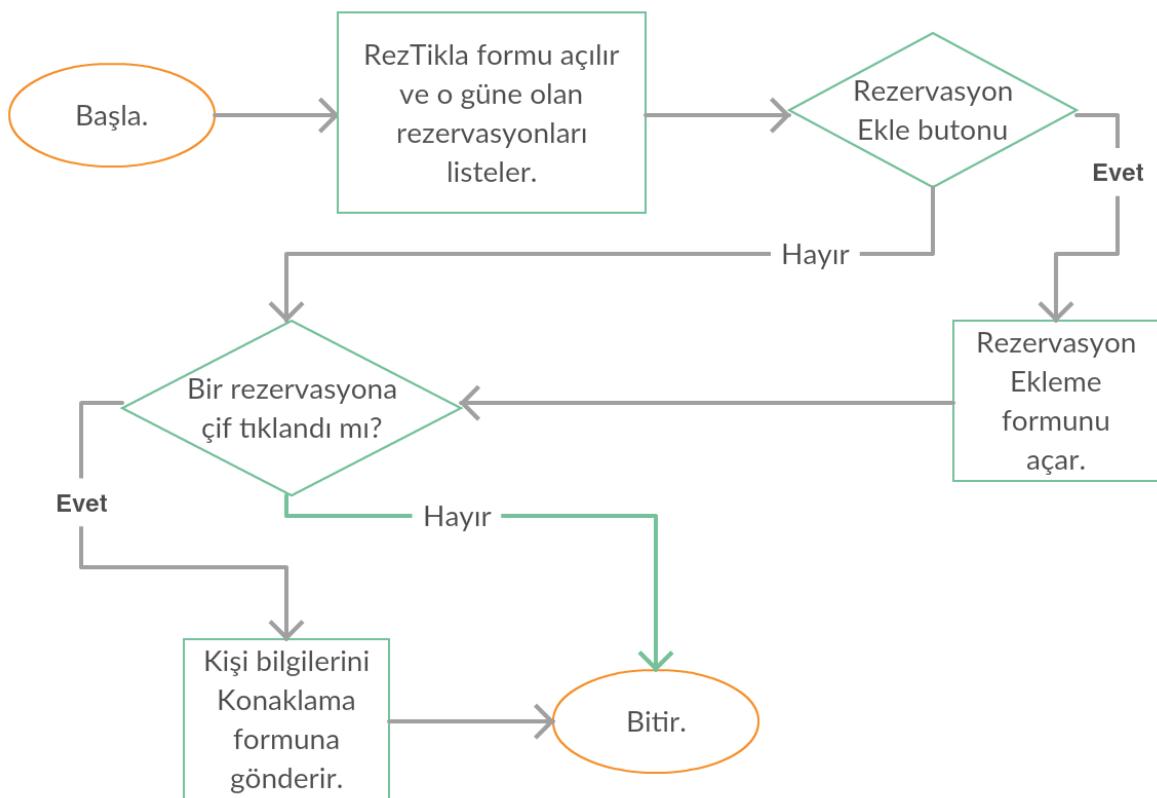
## RESEPSİYON.CS



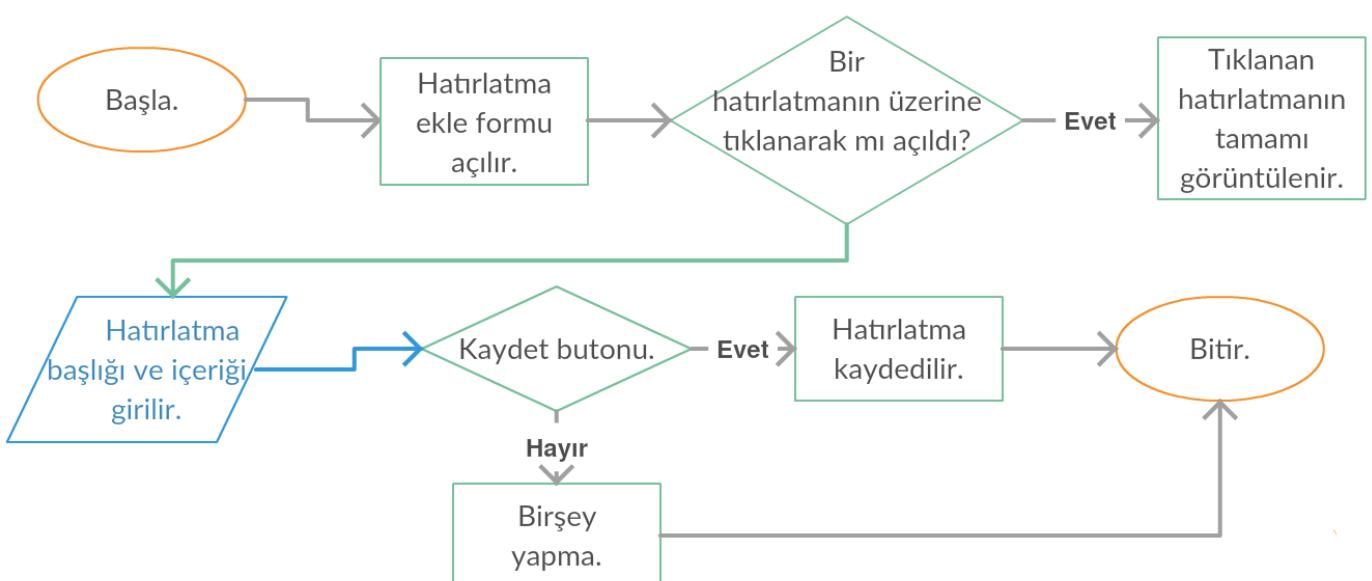
## R\_KONAK.cs



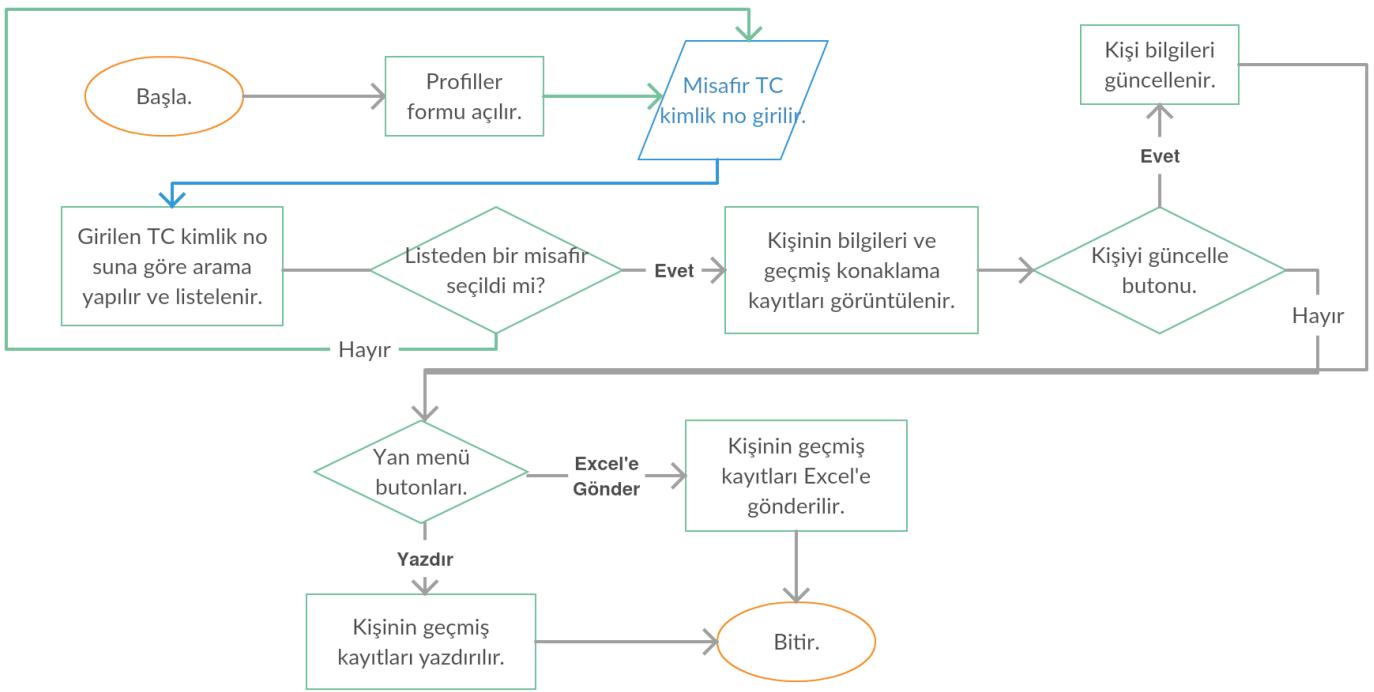
## REZ\_TİKLA.CS



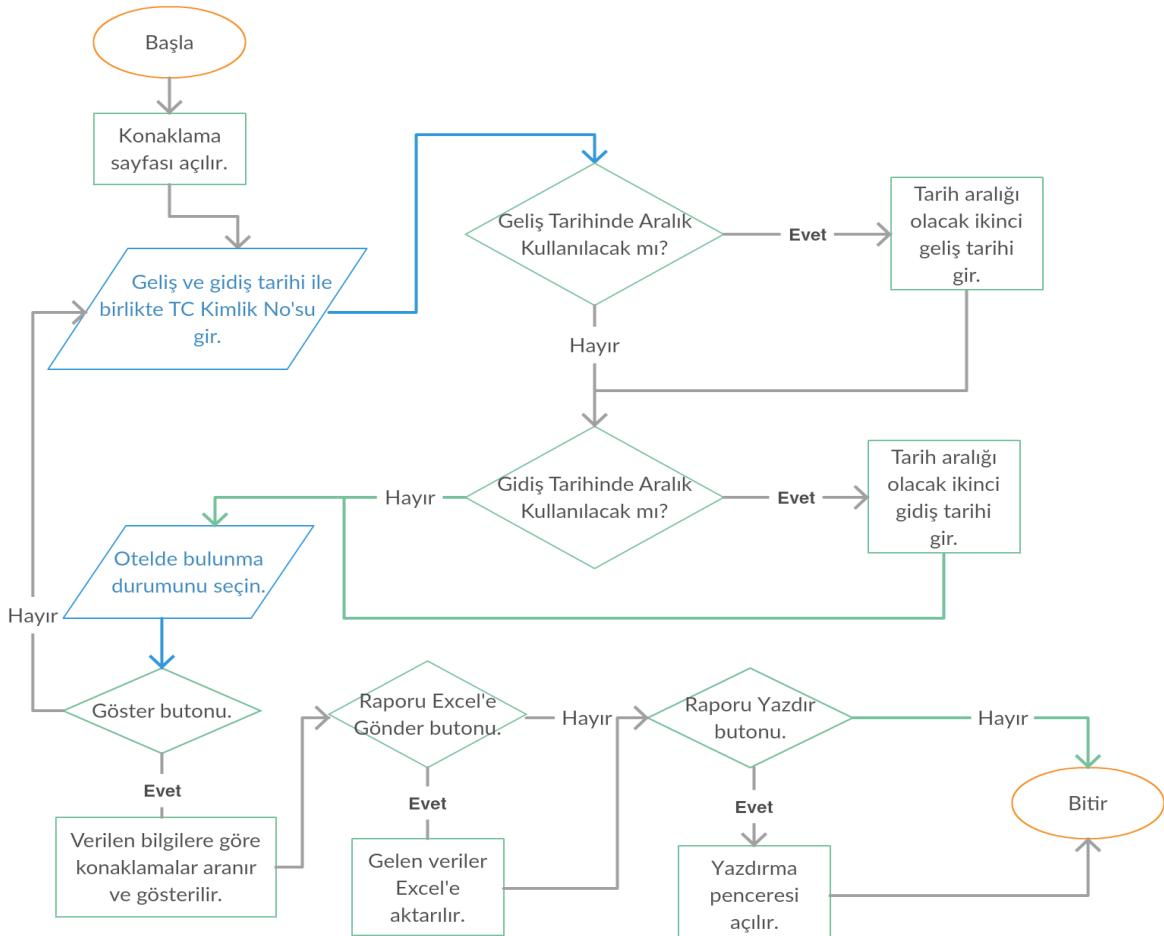
## HATIRLATMAEKLE.CS



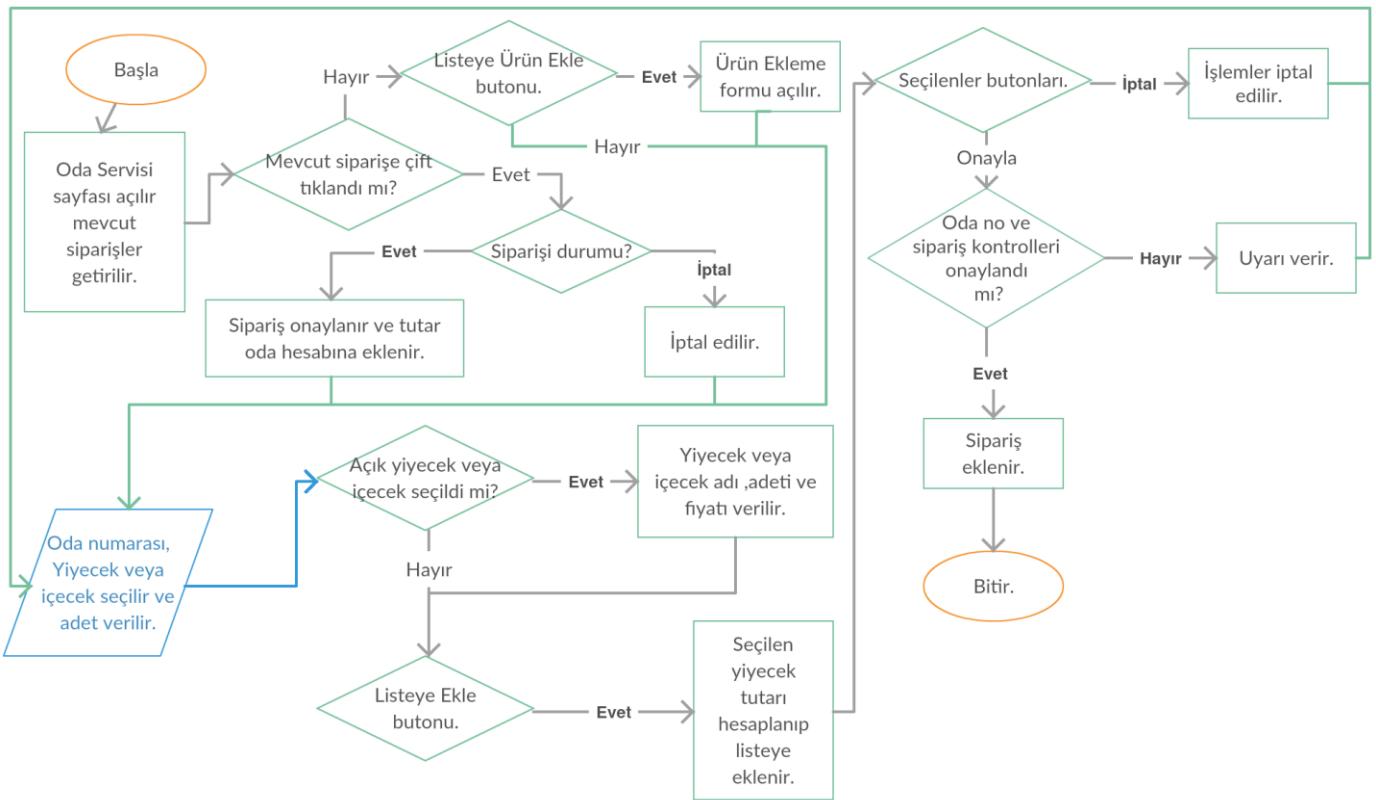
PROFILLER.CS



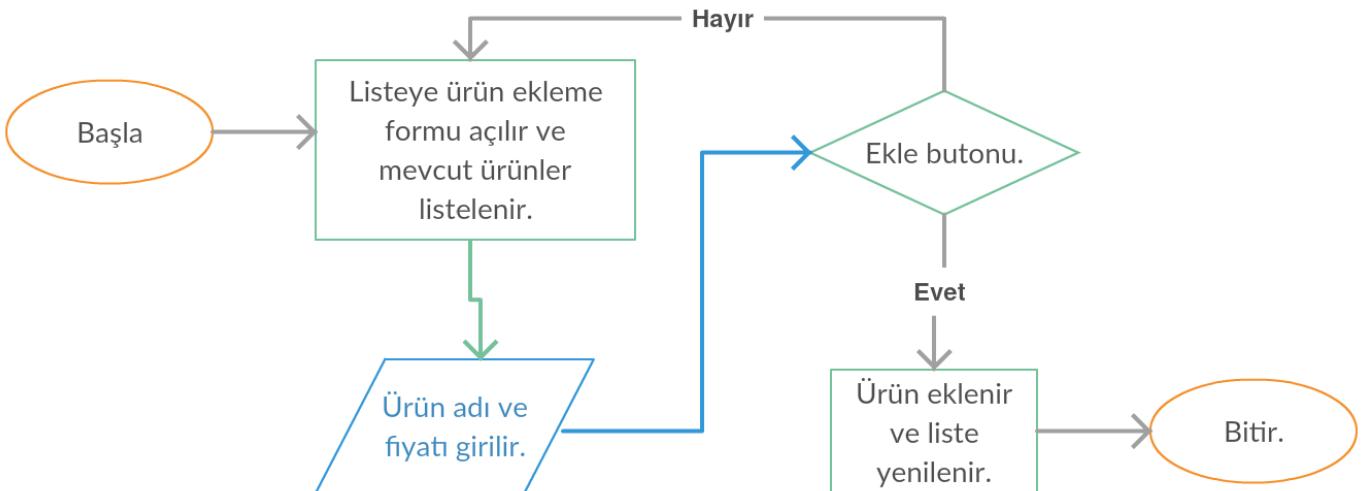
KONAKLAMALAR.CS



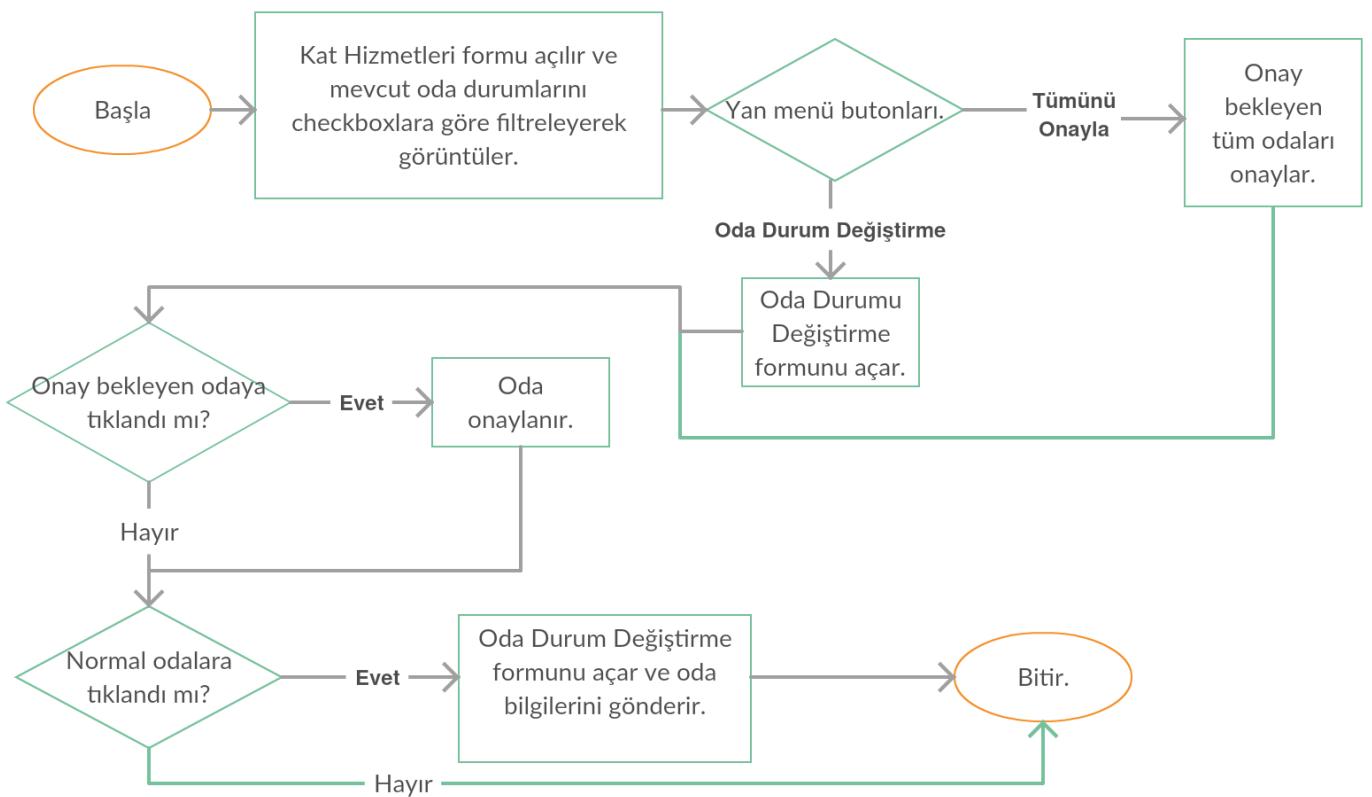
## ODASİPARİŞ.CS



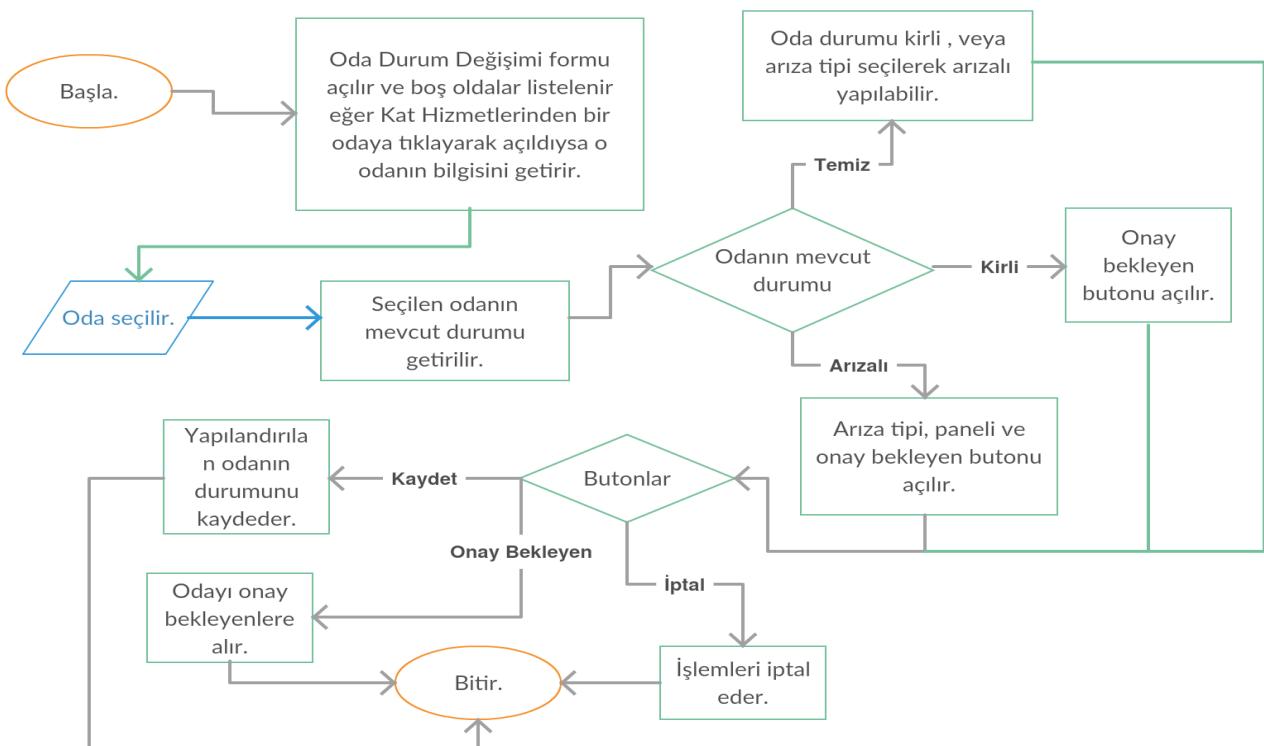
## URUNEKLE.CS



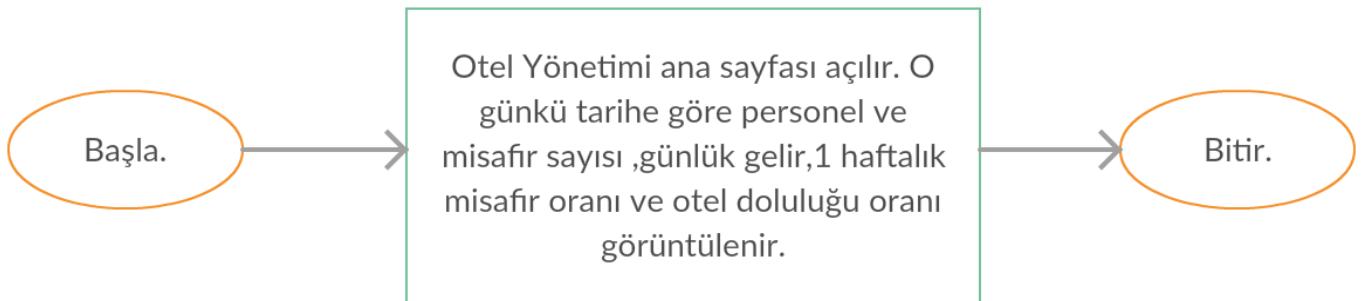
## KATHİZMETLERİ.CS



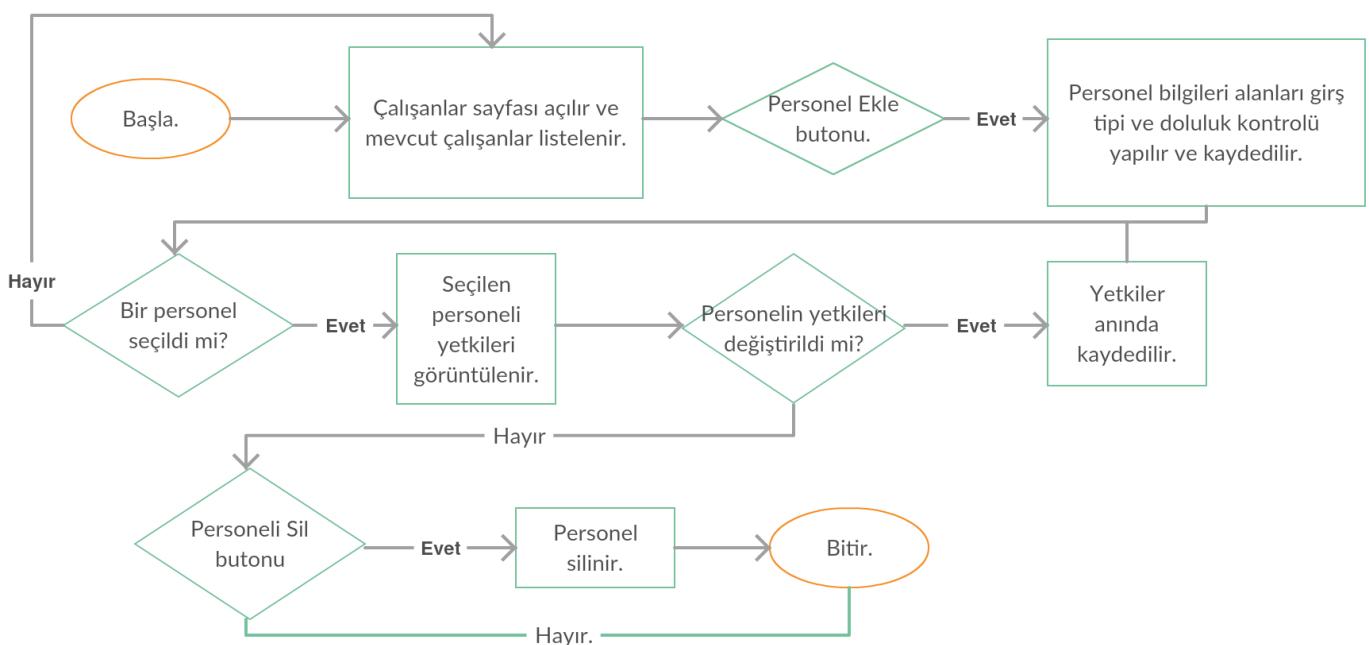
## ODADURUM.CS



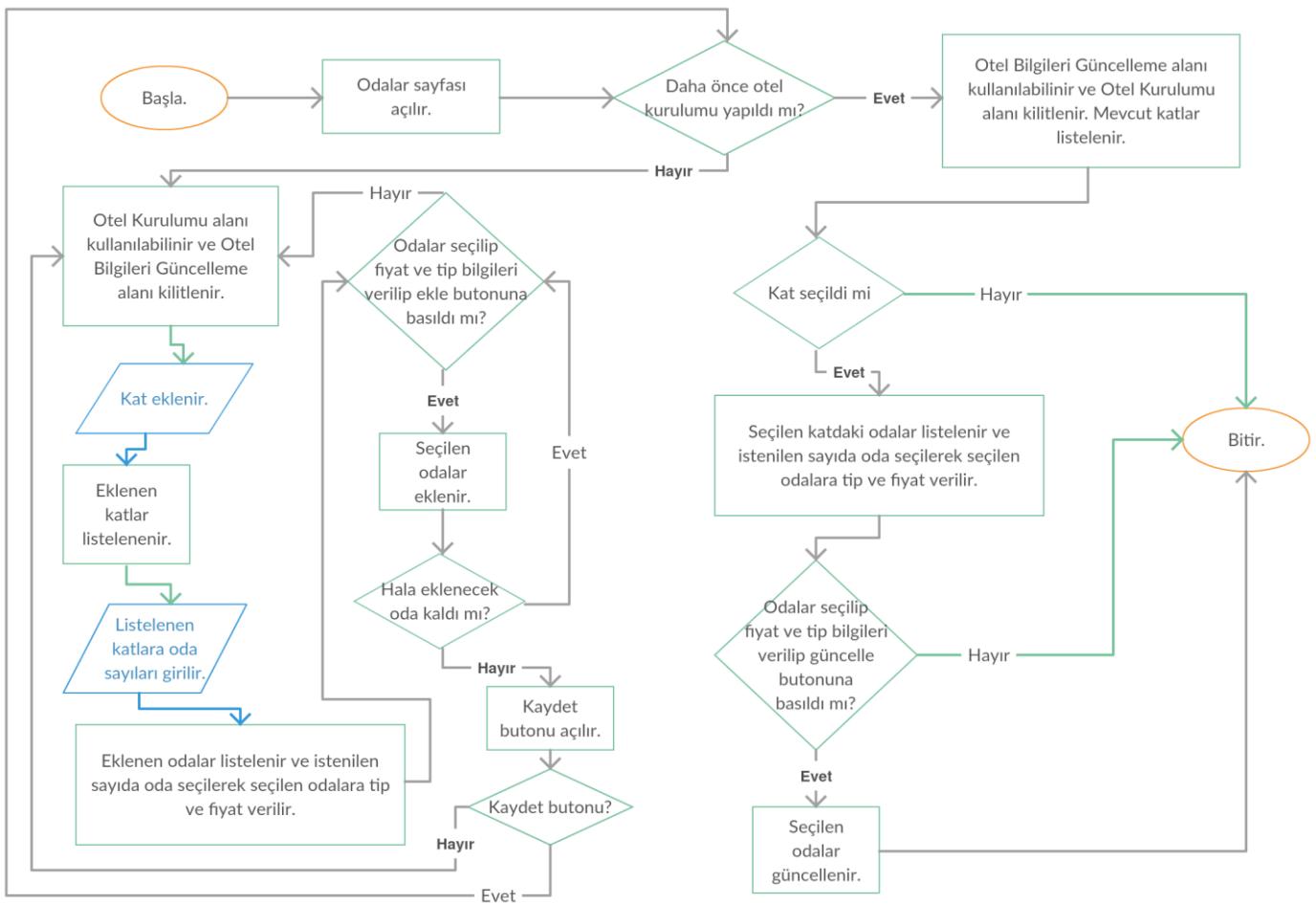
## YÖNETİCİPANEL.CS



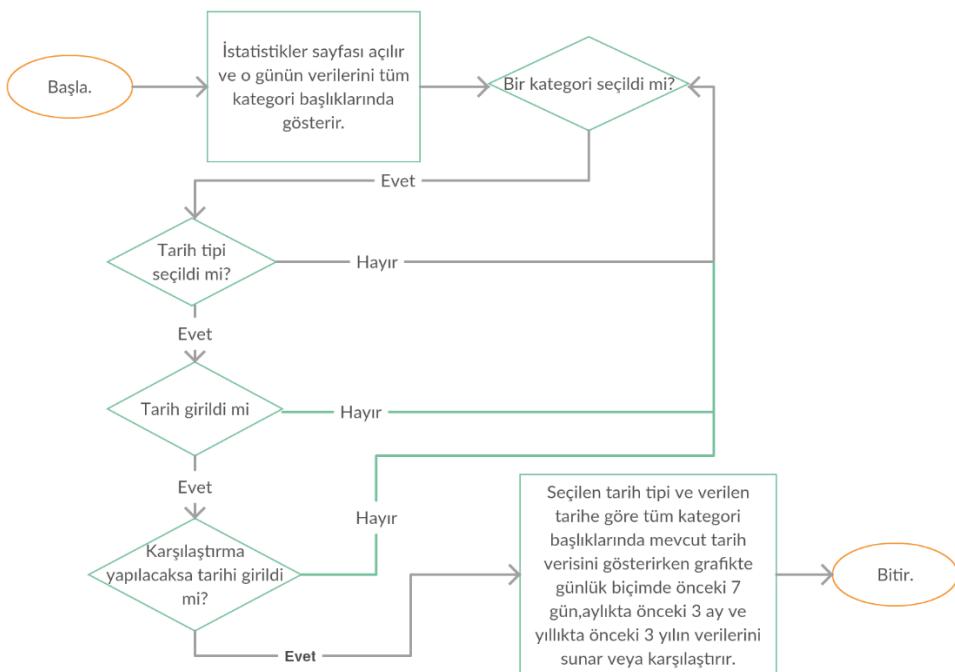
## PERSONEL.CS



## ODALAR.CS



## İSTATİSTİKLER.CS



## VERİTABANI NORMALİZASYONU

Projemizde kullandığımız veritabanı MsSQL ve genel olarak veritabanımızda veri tekrarından, veri çakışmasından muzdarip olmamak için gerek tablolarımız, gerek kullandığımız trigger ve view'larla destekledik.

Veritabanımız BCNF/NF(Boyce-Codd Normal Form) Statüsündedir.

Veritabanımız aşağıdaki tablo görsellerinde görebilirsiniz;

### Admin

Kullanici	Sifre
denep	dsd

### Çıkış Geçmiş

	OtelID	OdaNo	KalinanGeceSayısı	ToplamOdaFiyati	EkstraHarcamalar	GenelToplam	Tarih
1	11	101	4	480,00	0,00	480,00	2019-05-27
2	11	203	2	400,00	0,00	400,00	2019-05-27
3	11	305	35	14000,00	0,00	14000,00	2019-05-27
4	11	305	0	0,00	58,00	58,00	2019-05-27
5	11	305	0	0,00	0,00	0,00	2019-05-27
6	11	203	0	0,00	27,00	27,00	2019-05-27
7	11	102	1	120,00	0,00	120,00	2019-05-27
8	11	102	0	0,00	0,00	0,00	2019-05-27
9	11	102	0	0,00	0,00	0,00	2019-05-27
10	11	203	0	0,00	0,00	0,00	2019-05-29

### Giriş Geçmiş

	OtelID	TcKimlik	GelisTarihi	GidisTarihi	KaldiGeceSayisi	OdaNo
1	11	68746321654	2019-05-27	2019-05-31	4	101
2	11	65416574132	2019-05-27	2019-06-11	15	108
3	11	56965849848	2019-05-27	2019-06-11	15	108
4	11	54565498745	2019-05-27	2019-05-28	1	203
5	11	65416541965	2019-05-27	2019-05-28	1	102
6	11	98745641324	2019-03-01	2019-03-30	29	105
7	11	95659865498	2019-03-01	2019-03-25	24	201
8	11	98756256545	2019-03-03	2019-03-28	25	303
9	11	98751465421	2019-03-06	2019-03-15	9	108
10	11	20215268462	2019-03-10	2019-03-29	19	203

### Hatırlatmalar

OtelID	Hatırlatmano	Tarih	HatırlatmaBaslık	Hatırlatmametin
1	11	1	2019-06-02 12:00:03.510	Misafirleri devlet sitesine yükle. AKBS Sistemine gelen misafirleri ekle.

## İçecekler

	OtelID	İçecekAd	İçecekFiyat
1	11	Çilek Frozen	12,00
2	11	Karpuz Frozen	12,00
3	11	Şeftali Frozen	12,00
4	11	Frambuaz Frozen	12,00
5	11	Karamelli Milkshake	12,00
6	11	Kutu Kola	5,00
7	11	Ayran	3,00
8	11	Vanilyalı Frappe	12,00
9	11	Limonata	6,00
10	11	Şıxe Soda	6,00
11	11	Türk Kahvesi	10,00
12	11	Çikolatalı Espresso	10,00

## Yiyecekler

	OtelID	YiyecekAd	YiyecekFiyat
1	11	Çikolatalı Pasta	15,00
2	11	Muzlu Pasta	15,00
3	11	Profiterol	12,00
4	11	Karaorman	10,00
5	11	Malaga Bitter	10,00
6	11	Krokanlı Mono	10,00
7	11	Budapeşte	12,00
8	11	Jakompe	12,00
9	11	Cızbız Köfte	25,00
10	11	Et Quasedilla	27,00
11	11	Tire Köfte	26,00
12	11	Şefin Tavası	25,00

## İlçe

	TownID	CityID	TownName
1	1	1	ALADÄG
2	2	1	CEYHAN
3	3	1	ÇUKUROVA
4	4	1	FEKE
5	5	1	İMAMOĞLU
6	6	1	KARAİSALI
7	7	1	KARATAŞ
8	8	1	KOZAN
9	9	1	POZANTI
10	10	1	SAIMBEYLİ

## Kişi Çıkış

	OtelID	OdaNo	KalanGeceSayisi	ToplamOdaFiyati	EkstraHarcamalar	ToplamOdemeler	GenelToplam	Indirim	OdenecekTutar
1	11	305	15	6000,00	79,00	25,00	6079,00	0,00	6054,00
2	11	103	2	240,00	0,00	0,00	240,00	0,00	240,00

## Kişi Giriş

	OtelID	TcKimlik	GelisTarihi	GidisTarihi	KaldigiGeceSayisi	OdaNo
1	11	61006494392	2019-05-30	2019-06-14	15	305
2	11	65416354165	2019-05-30	2019-06-01	2	103

## Kullanıcılar

	OtelID	Ad	Soyad	KullaniciAdı	Sifre	Rezervasyon	Resepsyon	Profiller	Raporlar	OdaServisi	KatHizmetler	OtelYonetimi
1	12	selam	azs	1	1	1	1	1	1	1	1	1
2	11	Emrah	Tunç	emr	123	0	1	1	0	0	0	0
3	13	İsmail	Düğmez	is	123	1	1	1	1	1	1	0
4	13	İsmail	Düğmez	ismail	111	0	0	0	0	0	0	1
5	11	Melih Can	Akgünəş	mlh	123	0	1	0	0	1	1	0
6	11	Mustafa	Doğan	mstf	123	1	0	0	1	0	0	0
7	11	Murat	Akın	murat	123	1	1	1	1	1	1	1

## Mahalle

	NeighborhoodID	DistrictID	NeighborhoodName	ZipCode
1	1	1	AKPINAR MAH	01720
2	2	1	BAŞPINAR MAH	01720
3	3	1	MANSURLU MAH	01720
4	4	1	SİNANPAŞA MAH	01720
5	5	2	AKÖREN MAH	01722
6	6	2	BOZTAHTA MAH	01722
7	7	2	BÜYÜKSOFULU MAH	01722
8	8	2	CERİTLER MAH	01722
9	9	2	DAİLER MAH	01722
10	10	2	DARLIK MAH	01722

## Odalar

	OtelID	Kat	OdaID	OdaNo	OdaTip	Durum	FiyatKodu	Fiyat	Televizyon	Klima	Buzdolabi	MiniBar	Telefon
1	11	1	11101	101	1	2	1	120,00	0	0	0	0	0
2	11	1	11102	102	1	2	1	120,00	0	0	0	0	0
3	11	1	11103	103	1	1	1	120,00	1	1	1	1	1
4	11	1	11104	104	1	3	1	120,00	0	0	0	0	0
5	11	1	11105	105	2	2	1	150,00	0	0	0	0	0
6	11	1	11106	106	2	2	1	150,00	0	0	0	0	0
7	11	1	11107	107	2	2	1	150,00	1	1	1	1	1
8	11	1	11108	108	2	2	1	150,00	1	1	1	1	1
9	11	1	11109	109	5	2	1	180,00	1	1	1	1	1
10	11	1	11110	110	5	2	1	180,00	1	1	1	1	1

## Odalar Yenileme

	OtelID	OdaNo	Durum
1	11	101	2
2	11	102	2
3	11	103	1
4	11	104	3
5	11	105	2
6	11	106	2
7	11	107	2
8	11	108	2
9	11	109	2
10	11	110	2

## Ödemeler

	IslemNo	OtelID	OdaNo	Odeme	OdemeTarih
1	32	11	305	25.00	2019-05-30 23:44:10.783
2	20	11	105	250.00	2019-05-26 00:00:00.000
3	21	11	201	350.00	2019-05-25 00:00:00.000
4	22	11	108	430.00	2019-05-24 00:00:00.000
5	23	11	109	500.00	2019-05-23 00:00:00.000
6	24	11	303	1350.00	2019-05-22 00:00:00.000
7	25	11	208	2300.00	2019-05-21 00:00:00.000
8	26	11	302	1800.00	2019-05-20 00:00:00.000
9	27	11	108	4520.00	2019-05-27 06:43:32.240

## Ödemeler Geçmiş

	IslemNo	OtelID	OdaNo	Odeme	OdemeTarih
1	17	11	305	5000.00	2019-05-27
2	18	11	102	156.00	2019-05-27
3	1	11	103	200.00	2019-05-26
4	30	11	102	360.00	2019-05-27
5	31	11	203	427.00	2019-05-29
6	32	11	305	25.00	2019-05-30
7	33	11	101	115.00	2019-05-31
8	20	11	105	250.00	2019-05-26
9	21	11	201	350.00	2019-05-25
10	22	11	108	430.00	2019-05-24

## Oteller

	OtelID	OtelAd	OtelAdres	KatSayisi
1	11	İbis Hotel	Alsancak	3
2	12	su	izmir	1
3	13	Sular	İzmir	0

## R\_musteriekle

	OtelID	KmlkBelgeTuru	KmlkSeriNo	TcKmlkNo	Ad	Soyad	BabaAd	AnaAd	DogumYeri	DogumTarihi	Uyruk	Cinsiyeti	Medenihali	KaytildOlduguil	KaytildOldugulce
1	11	0	496874946	16516564156	Michael	Schumacher	Daniel	Amanda	Münih	2019-05-27	0	1	1		
2	11	1	465462135	24149443450	Emrah	Tunç	Osman	Şerife	Açgöl	1998-08-18	1	1	1	Nevşehir	Açgöl
3	11	1	546263452	26584632354	Emrah	Tunç	Mehmet	Şule	Nevşehir	1998-01-08	1	1	1	Nevşehir	Salçalı
4	11	1	sdgr35163	36054892666	Mustafa	Karaca	Osman	Merve	Muş	1994-06-16	1	1	1		
5	11	1	64z6vf514	48325137834	Mustafa	Doğan	Ömer	Zelha	Kahramanmaraş	1998-01-17	1	1	1		
6	11	1	496854865	54565498745	Ali	Kaya	Abdullah	Merve	Sinop	1997-12-30	1	1	0		
7	11	1	846546516	54652132103	Veli	Çelik	Murat	Ayşe	Gaziantep	1991-07-05	1	1	1		
8	11	1	658465269	56456521332	Osman	Akarsu	Zeki	Canan	Rize	1992-06-01	1	1	1		
9	11	0	584965468	56965849848	Archie	Qwenthur	Tony	Isabella	Paris	1997-10-16	0	1	1		
10	11	1	646984165	61006494392	Melih Can	Akgüneş	Ibrahim	Zülbeher	İstanbul	2019-05-30	1	1	0	OSMANİYE	BAHÇE

KaytildOlduguMahalle	CitNo	AileSiraNo	SiraNo	Yetiskin_Cocuk	Telefon	CepTelefon	Email	AracPlakaNo	Musteri_is	İkametkah	Notlar	Durum
	0	0	0	1	-	(554) 132-3213				Münih	1	
Bahçeli	2	35	4	1	-	(552) 234-9164	dariusemrah@gmail.com		Öğrenci	İzmir	0	
Tan	2	32	9	0	-	(687) 495-2165	emrah.tunc@gmail.com			İzmir	0	
	0	0	0	1	-	(564) 165-1314				Muş	1	
	3	5	25	0	528-4510	(505) 106-5060				İzmir	1	
	0	0	0	1	-	(546) 985-2654				Ankara	0	
	0	0	0	0	-	(864) 512-8465				İstanbul	1	
	0	0	0	1	-	(684) 628-6425				Uşak	1	
	0	0	0	1	-	(987) 495-1264				İzmir	1	
MERKEZKÖYLER	3	25	125	1	-	(505) 106-5060	melih.can.a@hotmail.com		Öğrenci	İzmir	0	

## Rez\_Rapor

	OtelID	TcNo	GelisTarih	ad	soyad	uyruk
1	11	61006494392	2019-05-27	Melih Can	Akgüneş	1
2	11	68746321654	2019-05-27	Mustafa	Doğan	1
3	11	26584632354	2019-05-27	Emrah	Tunç	1
4	11	54565498745	2019-05-27	Ali	Kaya	1
5	11	65416574132	2019-05-27	Elizabeth	Stark	0
6	11	54652132103	2019-05-28	Veli	Çelik	1
7	11	56456521332	2019-05-29	Osman	Akarsu	1
8	11	65416354165	2019-05-30	Sylvester	Stallone	0
9	11	16516564156	2019-05-31	Michael	Schumacher	0
10	11	65416541965	2019-05-27	Melek	Demir	1

## SiparisGecmis

	OtelID	SiparisID	OdaNo	Siparis	Tutar	Tarih
1	11	16	305	1-Çikolatalı Pasta = 15 / 1-Profiterol = 12 / 1-Cızb...	58,00	2019-05-27
2	11	17	203	1-Muzlu Pasta = 15 / 1-Karamelli Milkshake = 12	27,00	2019-05-27
3	11	18	101	1-Jakompe = 12	12,00	2019-05-27
4	11	20	102	1-Profiterol = 12 / 2-Çilek Frozen = 24	36,00	2019-05-27
5	11	21	305	1-Muzlu Pasta = 15 / 1-Malaga Bitter = 10 / 1-Et ...	76,00	2019-05-30
6	11	23	305	1-Ayran = 3	3,00	2019-05-30
7	11	24	101	1-Çikolatalı Pasta = 15	15,00	2019-05-31
8	11	25	101	1-Domuz Bonfile = 100	100,00	2019-05-31
9	11	19	108	2-Çikolatalı Espresso = 20	20,00	2019-05-27

## Siparisler

	OtelID	SiparisID	OdaNo	Siparis	Tutar	SiparisDurum
1	11	21	305	1-Muzlu Pasta = 15 / 1-Malaga Bitter = 10 / 1-Et ...	76,00	0
2	11	22	305	1-Et Quasedilla = 27	27,00	1
3	11	19	108	2-Çikolatalı Espresso = 20	20,00	0
4	11	23	305	1-Ayran = 3	3,00	0

## Sehir

	CityID	CountryID	CityName	PlateNo	PhoneCode
1	1	212	ADANA	1	322
2	2	212	ADİYAMAN	2	416
3	3	212	AFYONKARAHİSAR	3	272
4	4	212	AĞRI	4	472
5	5	212	AKSARAY	68	382
6	6	212	AMASYA	5	358
7	7	212	ANKARA	6	312
8	8	212	ANTALYA	7	242
9	9	212	ARDAHAN	75	478
10	10	212	ARTVİN	8	466

## Kasaba

	TownID	CityID	TownName
1	1	1	ALADAĞ
2	2	1	CEYHAN
3	3	1	ÇUKUROVA
4	4	1	FEKE
5	5	1	İMAMOĞLU
6	6	1	KARAİSALI
7	7	1	KARATAŞ
8	8	1	KOZAN
9	9	1	POZANTI
10	10	1	SAİMBEYLİ

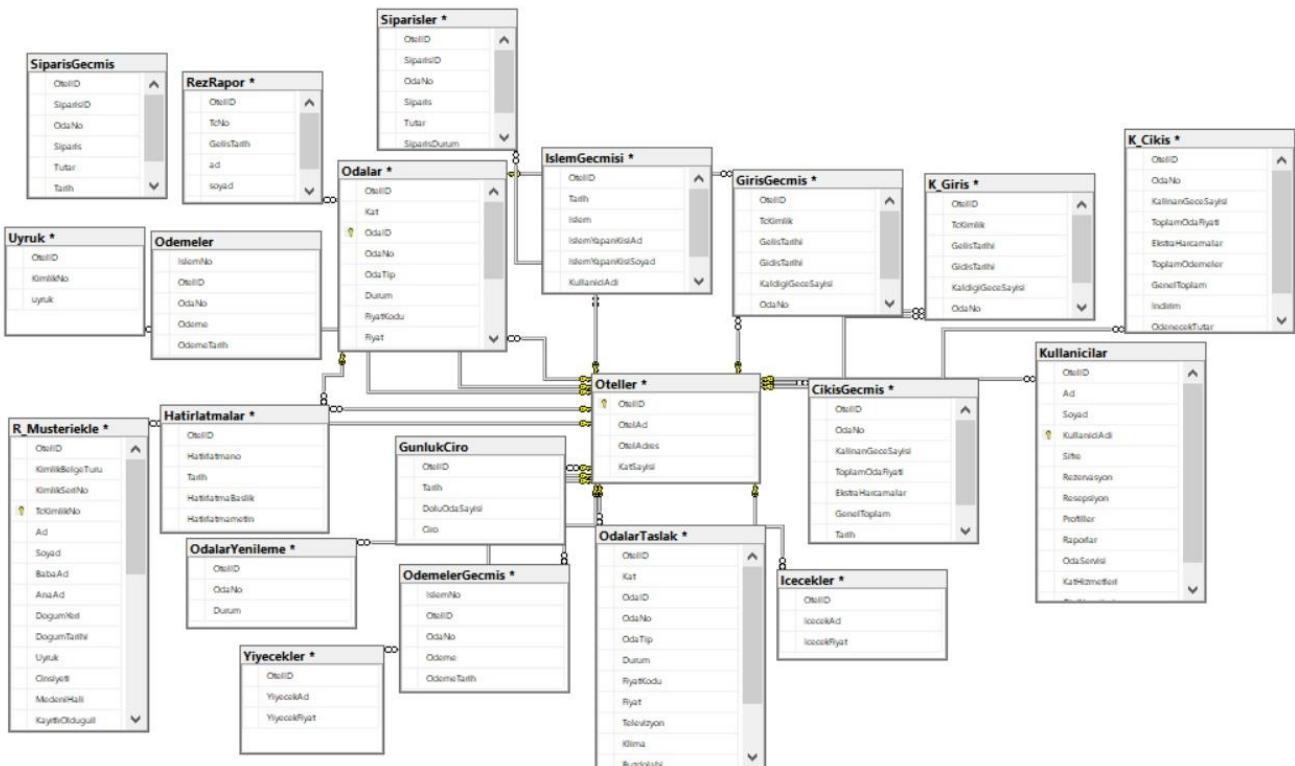
## Uyruk

	OtelID	KimlikNo	uyruk
1	11	16516564156	United States
2	11	56965849848	United States
3	11	65416354165	France
4	11	65416574132	Germany
5	11	24149443450	Turkey
6	11	26584632354	Turkey
7	11	54565498745	Turkey
8	11	54652132103	Turkey
9	11	56456521332	Turkey
10	11	61006494392	Turkey

## Ülke

	CountryID	BinaryCode	TripleCode	CountryName	PhoneCode
1	1	AF	AFG	Afghanistan	93
2	2	AL	ALB	Albania	355
3	3	DZ	DZA	Algeria	213
4	4	AS	ASM	American Samoa	684
5	5	AD	AND	Andorra	376
6	6	AO	AGO	Angola	244
7	7	AI	AIA	Anguilla	264
8	8	AQ	ATA	Antarctica	672
9	9	AG	ATG	Antigua and Barbuda	268
10	10	AR	ARG	Argentina	54

## ENTİTY RELATIONSHIP DİYAGRAMI



## PROGRAM KODLARI

### GİRİŞ.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{
    public partial class Login : Form
    {
        public Login()
        {
            InitializeComponent();
        }

        public static string veritabani = "Data Source=DESKTOP-7JCKATF\MELIH;Integrated Security=SSPI;Initial Catalog=OtelProje";
        SqlConnection con = new SqlConnection(veritabani);
        SqlConnection con2 = new SqlConnection(veritabani);

        public static int id;

        private void bunifuTileButton1_Click_1(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void bunifuFlatButton1_Click(object sender, EventArgs e)
        {
            try
            {
```

```

SqlCommand cmd = new SqlCommand("Select * from Kullanilar where KullaniciAdi='" +
bunifuMetroTextbox1.Text + "' and Sifre='" + bunifuMetroTextbox2.Text + "'", con);
con.Open();
SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())
{
    OtelArayuz oa = new OtelArayuz();
    Resepsyon resepsiyon = new Resepsyon();
    oa.Show();
    this.Hide();
    id=int.Parse(dr["OtelID"].ToString());

    if (Boolean.Parse(dr["Rezervasyon"].ToString()) == true)
    {
        oa.btnrezervasyon.Enabled = true;
    }
    else
    {
        oa.btnrezervasyon.Enabled = false;
    }
    if (Boolean.Parse(dr["Resepsyon"].ToString()) == true)
    {
        oa.btnresepsiyon.Enabled = true;
    }
    else
    {
        oa.btnresepsiyon.Enabled = false;
    }
    if (Boolean.Parse(dr["Profiller"].ToString()) == true)
    {
        oa.btnprofiller.Enabled = true;
    }
    else
    {
        oa.btnprofiller.Enabled = false;
    }
    if (Boolean.Parse(dr["Raporlar"].ToString()) == true)

```

```

{
    oa.btnraporlar.Enabled = true;
}
else
{
    oa.btnraporlar.Enabled = false;
}
if (Boolean.Parse(dr["OdaServisi"].ToString()) == true)
{
    oa.btnodaservisi.Enabled = true;
}
else
{
    oa.btnodaservisi.Enabled = false;
}
if (Boolean.Parse(dr["KatHizmetleri"].ToString()) == true)
{
    oa.btnkathizmetleri.Enabled = true;
}
else
{
    oa.btnkathizmetleri.Enabled = false;
}
if (Boolean.Parse(dr["OtelYonetimi"].ToString()) == true)
{
    oa.bnnotelyonetimi.Enabled = true;
}
else
{
    oa.bnnotelyonetimi.Enabled = false;
}

}

else
{
    MessageBox.Show("KullanıcıAdı Veya Şifre Yanlış");
}
con.Close();

```

```
        }

        catch(Exception ex)
        {
            MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

private void pictureBox1_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("Select * from Admin where Kullanici=" +
bunifuMetroTextbox1.Text + " and Sifre=" + bunifuMetroTextbox2.Text + "", con);
        con.Open();
        SqlDataReader dr = cmd.ExecuteReader();

        if (dr.Read())
        {
            Admin admin = new Admin();
            admin.Show();
        }

        con.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void bunifuMetroTextbox2_ValueChanged(object sender, EventArgs e)
{
}
```

## ADMIN.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{
    public partial class Admin : Form
    {
        SqlConnection baglanti = new SqlConnection(Login.veritabani);

        public Admin()
        {
            InitializeComponent();
        }

        private void bunifuMetroTextbox2_ValueChanged(object sender, EventArgs e)
        {

        }

        private void bunifuFlatButton1_Click(object sender, EventArgs e)
        {
            if(otelad.Text!="" && oteladres.Text!="" && ad.Text!="" && soyad.Text!="" && kad.Text!="" && ksifre.Text!="")
            {
                try
                {
                    SqlCommand ekle = new SqlCommand("insert into Oteller(OtelAd,OtelAdres,KatSayisi) values('" + otelad.Text + "','" + oteladres.Text + "',0)", baglanti);
                    baglanti.Open();
                }
            }
        }
}
```

```
ekle.ExecuteNonQuery();
baglanti.Close();

baglanti.Open();
int otelid = 0;
SqlCommand oku = new SqlCommand("select OtelID from Oteller where OtelAd='' +
otelad.Text + "" and OtelAdres='' + oteladres.Text + "" and KatSayisi=0", baglanti);
SqlDataReader getir = oku.ExecuteReader();
while (getir.Read())
{
    otelid = int.Parse(getir[0].ToString());
}
baglanti.Close();
SqlCommand ekle2 = new SqlCommand("insert into Kullanicilar values('"+ otelid + "','" +
ad.Text + "','" + soyad.Text + "','" + kad.Text + "','" + ksifre.Text + "','" + true + "','" + true + "','" + true + "','" + true + "','" + true + "','" + true + "')", baglanti);
baglanti.Open();
ekle2.ExecuteNonQuery();
baglanti.Close();

MessageBox.Show("Otel başarıyla eklendi.", "Bilgilendirme", MessageBoxButtons.OK,
MessageBoxIcon.Information);

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Bilgilendirme", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}

}

private void bunifuTileButton1_Click(object sender, EventArgs e)
{
    this.Close();
}
}
```

## OTELARAYÜZ.CS

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Bunifu.UI.WinForms.BunifuButton;
using System.Data.SqlClient;
using Excel = Microsoft.Office.Interop.Excel;
using System.Xml;
```

```
namespace Login
```

```
{
```

```
    public partial class OtelArayüz : Form
    {
        Form kapa = new Form();
        StringFormat strFormat;

        ArrayList arrColumnLefts = new ArrayList();
        ArrayList arrColumnWidths = new ArrayList();
        int iCellHeight = 0;
        int iTotWidth = 0;
        int iRow = 0;
        bool bFirstPage = false;
        bool bNewPage = false;
        int iHeaderHeight = 0;
        int OtelId = Login.id;
```

```
    public void Fomrdgsme(int a)
```

```

{
    bunifuLabel1.Visible = false;
    Form child=new Form();

    try
    {
        kapa.Close();
    }
    catch (Exception)
    {

        throw;
    }

    if (a == 1)
    { child = new Rezervasyon();
        kapa = child;
    }
    else if (a == 2)
    { child = new Resepsiyon();
        kapa = child;
    }
    else if (a == 3)
    { child = new Profiller();
        kapa = child;
    }
    else if (a == 4)
    { child = new Konaklamalar();
        kapa = child; }
    else if (a == 5)
    { child = new OdaSiparis();
        kapa = child;
    }
    else if (a == 6)
    { child = new KatHizmetleri();
        kapa = child;
    }
    else if (a == 7)
    { child = new RezRapor();

```

```

    kapa = child;
}
else if(a==8)
{
    child = new YoneticiPanel();
    kapa = child;
}
else if (a == 9)
{
    child = new YoneticiPanel();
    kapa = child;
}
else if (a == 10)
{
    child = new Personel();
    kapa = child;
}
else if (a == 11)
{
    child = new Odalar();
    kapa = child;
}
else if (a == 12)
{
    child = new Istatistikler();
    kapa = child;
}
else if (a == 13)
{
    child = new Log();
    kapa = child;
}
else
{
    child = new Profiller();
    kapa = child;
}

child.TopLevel = false; //set it's TopLevel to false
Controls.Add(child); //and add it to the parent Form

```

```

        child.Show(); //finally display it
        child.Location = new Point(200, 80);
        child.Dock = DockStyle.Fill;
        panel5.Controls.Add(child);
        child.BringToFront();

    }

public OtelArayuz()
{
    InitializeComponent();
}

private void OtelArayuz_Load(object sender, EventArgs e)
{
    timer1.Enabled = true;
    Rez_Ekle.IconVisible = true; Rez_Sil.IconVisible = true; bunifuFlatButton3.IconVisible = true;
    bunifuFlatButton4.IconVisible = true; bunifuFlatButton5.IconVisible = true;
    bunifuFlatButton6.IconVisible = true; Rez_Raporu.IconVisible = true; bunifuFlatButton8.IconVisible =
    true; bunifuFlatButton9.IconVisible = true; bunifuFlatButton10.IconVisible = true;
    bunifuFlatButton11.IconVisible = true; btnexcelprofil.IconVisible = true; btnyazdirprofil.IconVisible =
    true; Excel_Gonder.IconVisible = true; Yazdir.IconVisible = true; bunifuFlatButton16.IconVisible = true;
    bunifuFlatButton17.IconVisible = true; bunifuFlatButton18.IconVisible = true;
    bunifuFlatButton19.IconVisible = true; bunifuFlatButton20.IconVisible = true;
    bunifuFlatButton21.IconVisible = true; bunifuFlatButton22.IconVisible = true;
    bunifuFlatButton23.IconVisible = true; bunifuFlatButton24.IconVisible = true;
    bunifuFlatButton5.Text = " - "+DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss");
    bunifuFlatButton6.Text = " - " +
    DateTime.Now.Day.ToString()+"."+DateTime.Now.Month.ToString()+"."+DateTime.Now.Year.ToString());
    bunifuPages1.SelectedTab = tabPage8;
    bunifuLabel1.Location = new Point(panel5.Size.Width/2-202,panel5.Size.Height/2);

    TCMBDovizKuruAl();
}

```

```

private void bunifuTileButton6_Click(object sender, EventArgs e)
{
    Resepsiyon resepsiyon = new Resepsiyon();
    resepsiyon.Dispose();
    resepsiyon.Close();

    Application.Exit();
}

private void bunifuTileButton2_Click(object sender, EventArgs e)
{
    bunifuPages1.SelectedTab = tabPage4;
    Fomrdgsme(4);

}

private void bunifuTileButton1_Click(object sender, EventArgs e)
{
    bunifuPages1.SelectedTab = tabPage2;
    Fomrdgsme(2);
}

private void bunifuTileButton8_Click(object sender, EventArgs e)
{
    bunifuPages1.SelectedTab = tabPage1;
    Fomrdgsme(1);
}

private void bunifuTileButton7_Click(object sender, EventArgs e)
{
    bunifuPages1.SelectedTab = tabPage3;
    Fomrdgsme(3);
}

private void bunifuTileButton3_Click(object sender, EventArgs e)
{
}

```

```

        bunifuPages1.SelectedTab = tabPage5;
        Fomrdgsme(5);
    }

private void bunifuTileButton9_Click(object sender, EventArgs e)
{
    bunifuPages1.SelectedTab = tabPage6;
    Fomrdgsme(6);
}

private void bunifuTileButton4_Click(object sender, EventArgs e)
{
    bunifuPages1.SelectedTab = tabPage7;
    Fomrdgsme(8);
}

private void bunifuFlatButton7_Click(object sender, EventArgs e)
{
    Fomrdgsme(7);
}

private void bunifuFlatButton20_Click(object sender, EventArgs e)
{
    Fomrdgsme(9);
}

private void bunifuFlatButton19_Click(object sender, EventArgs e)
{
    Fomrdgsme(10);
}

private void bunifuFlatButton22_Click(object sender, EventArgs e)
{
    Fomrdgsme(11);
}

private void bunifuFlatButton21_Click(object sender, EventArgs e)
{
    Fomrdgsme(12);
}

```

```

}

private void bunifuFlatButton24_Click(object sender, EventArgs e)
{
    Fomrdgsme(13);
}

private void bunifuFlatButton23_Click(object sender, EventArgs e)
{
    Fomrdgsme(14);
}

private void bunifuFlatButton18_Click(object sender, EventArgs e)
{
    Oda_Durum pencere = new Oda_Durum();
    pencere.Show();
}

private void bunifuFlatButton17_Click(object sender, EventArgs e)
{
    var pencere = (KatHizmetleri)Application.OpenForms["KatHizmetleri"];
    if (pencere != null)
    {
        pencere.TumOnay();
        pencere.OdaDurumDegisimi();
        pencere.OdaOnaylanacaklar();
    }
}

private void bunifuFlatButton16_Click(object sender, EventArgs e)
{
    UrunEkle urunekle = new UrunEkle();
    urunekle.Show();
}

private void btnexcelprofil_Click(object sender, EventArgs e)//excele veri aktarma işlemleri aşağıda
{
}

```

```

Profiller form = Application.OpenForms["Profiller"] as Profiller;
try
{
    if (form.bunifuCustomDataGridView1.Rows.Count != 0)
    {
        Microsoft.Office.Interop.Excel.Application uyg = new
Microsoft.Office.Interop.Excel.Application();
        uyg.Visible = true;
        Microsoft.Office.Interop.Excel.Workbook kitap =
uyg.Workbooks.Add(System.Reflection.Missing.Value);
        Microsoft.Office.Interop.Excel.Worksheet sheet1 =
(Microsoft.Office.Interop.Excel.Worksheet)kitap.Sheets[1];

        int StartCol = 1;

        int StartRow = 1;

        for (int j = 0; j < form.bunifuCustomDataGridView1.ColumnCount; j++)
        {
            Microsoft.Office.Interop.Excel.Range myRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow, StartCol + j];
            myRange.Value2 = form.bunifuCustomDataGridView1.Columns[j].HeaderText;
            myRange.ColumnWidth = 15;
        }

        StartRow++;

        for (int i = 0; i < form.bunifuCustomDataGridView1.Rows.Count; i++)
        {
            for (int j = 0; j < form.bunifuCustomDataGridView1.Columns.Count; j++)
            {

```

```

        Microsoft.Office.Interop.Excel.Range myRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow + i, StartCol + j];

        myRange.Value2 = form.bunifuCustomDataGrid1[j, i].Value == null ? "" :
form.bunifuCustomDataGrid1[j, i].Value;

    }

}

else
{
    MessageBox.Show("Veri olmadan excel çıktısı alamazsınız!", "Uyarı",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
}

catch (Exception)
{
    MessageBox.Show("Bir Hata Oluştu ", "Uyarı", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
}

}

private void btnyazdirprofil_Click(object sender, EventArgs e) //Profil formu yazdırma işlemleri
{
    try
    {
        Profiller form = Application.OpenForms["Profiller"] as Profiller;
        if (form.bunifuCustomDataGrid1.Rows.Count > 0)
        {
            PrintPreviewDialog onizleme = new PrintPreviewDialog();
            onizleme.Document = printDocument1;
            onizleme.ShowDialog();
        }
    }
}

```

```

        }
    else
    {
        MessageBox.Show("Profil Bilgileri Olmadan Çıktı Alamazsınız", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    // https://social.msdn.microsoft.com/Forums/tr-TR/2d97993e-96a0-4524-ad2c-
b0e82a0a9238/datagrid-deki-bilgileri-yazdirma?forum=csharptr
    //yazdırma kısmını yapabilmek için yukarıdaki makaleden yardım aldım
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

```

private void bunifuFlatButton8_Click(object sender, EventArgs e)
{
    r_musteri_ekle musteri_Ekle = new r_musteri_ekle();
    musteri_Ekle.Show();
}

```

```

private void printDocument1_BeginPrint(object sender, System.Drawing.Printing.PrintEventArgs e)
{
    Profiller form = Application.OpenForms["Profiller"] as Profiller;
    try
    {
        strFormat = new StringFormat();
        strFormat.Alignment = StringAlignment.Near;
        strFormat.LineAlignment = StringAlignment.Center;
        strFormat.Trimming = StringTrimming.EllipsisCharacter;

        arrColumnLefts.Clear();
        arrColumnWidths.Clear();
        iCellHeight = 0;
        iRow = 0;
        bFirstPage = true;
        bNewPage = true;
    }
}

```

```

iTTotalWidth = 0;
foreach (DataGridViewColumn dgvGridCol in form.bunifuCustomDataGrid1.Columns)
{
    iTTotalWidth += dgvGridCol.Width;
}
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

} /// Profil Formu yazdırma İşlemleri

void TCMBDovizKuruAl()
{
    try
    {
        string today = "http://www.tcmb.gov.tr/kurlar/today.xml";

        var xmlDoc = new XmlDocument();
        xmlDoc.Load(today);
        string USD_Satis =
xmlDoc.SelectSingleNode("Tarih_Date/Currency[@Kod='USD']/BanknoteSelling").InnerText;
        string EURO_Satis =
xmlDoc.SelectSingleNode("Tarih_Date/Currency[@Kod='EUR']/BanknoteSelling").InnerText;

        bunifuFlatButton11.Text = USD_Satis;
        bunifuFlatButton10.Text = EURO_Satis;
        bunifuFlatButton3.Text = USD_Satis;
        bunifuFlatButton4.Text = EURO_Satis;
    }
    catch (Exception)
    {
        MessageBox.Show("İnternet bağlantınız bulunmadığı için döviz kurları alınamadı!",
"Bilgilendirme", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

```

private void bunifuFlatButton9_Click(object sender, EventArgs e)
{
    Rez_Sil rez_Sil = new Rez_Sil();
    rez_Sil.Show();
}

private void printDocument1_PrintPage_1(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
{
    Profiller form = Application.OpenForms["Profiller"] as Profiller;
    try
    {
        int iLeftMargin = e.MarginBounds.Left;
        int iTopMargin = e.MarginBounds.Top;
        bool bMorePagesToPrint = false;
        int iTmpWidth = 0;
        bFirstPage = true;

        if (bFirstPage)
        {
            foreach (DataGridViewColumn GridCol in form.bunifuCustomDataGrid1.Columns)
            {
                iTmpWidth = (int)(Math.Floor((double)((double)GridCol.Width /
                    (double)iTotalWidth * (double)iTotalWidth *
                    (double)e.MarginBounds.Width / (double)iTotalWidth)));
                iHeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,
                    GridCol.InheritedStyle.Font, iTmpWidth).Height) + 11;

                arrColumnLefts.Add(iLeftMargin);
                arrColumnWidths.Add(iTmpWidth);
                iLeftMargin += iTmpWidth;
            }
        }

        while (iRow <= form.bunifuCustomDataGrid1.Rows.Count - 1)
        {
            DataGridViewRow GridRow = form.bunifuCustomDataGrid1.Rows[iRow];

```

```

iCellHeight = GridRow.Height + 5;
int iCount = 0;

if (iTopMargin + iCellHeight >= e.MarginBounds.Height + e.MarginBounds.Top)
{
    bNewPage = true;
    bFirstPage = false;
    bMorePagesToPrint = true;
    break;
}
else
{
    if (bNewPage)
    {

        e.Graphics.DrawString("Profil Bilgileri", new Font(form.bunifuCustomDataGrid1.Font,
FontStyle.Bold),
            Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -
            e.Graphics.MeasureString("Profil Bilgileri", new
Font(form.bunifuCustomDataGrid1.Font,
FontStyle.Bold), e.MarginBounds.Width).Height - 13);

        String strDate = DateTime.Now.ToString(" " +
DateTime.Now.ToString(" "));

        e.Graphics.DrawString(strDate, new Font(form.bunifuCustomDataGrid1.Font,
FontStyle.Bold),
            Brushes.Black, e.MarginBounds.Left + (e.MarginBounds.Width -
e.Graphics.MeasureString(strDate, new Font(form.bunifuCustomDataGrid1.Font,
FontStyle.Bold), e.MarginBounds.Width).Width), e.MarginBounds.Top -
e.Graphics.MeasureString("Profil Bilgileri", new Font(new
Font(form.bunifuCustomDataGrid1.Font,
FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);

    }
}

iTopMargin = e.MarginBounds.Top;
foreach (DataGridViewColumn GridCol in form.bunifuCustomDataGrid1.Columns)
{
}

```

```

e.Graphics.FillRectangle(new SolidBrush(Color.LightGray),
    new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
    (int)arrColumnWidths[iCount], iHeaderHeight));

e.Graphics.DrawRectangle(Pens.Black,
    new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
    (int)arrColumnWidths[iCount], iHeaderHeight));

e.Graphics.DrawString(GridCol.HeaderText, GridCol.InheritedStyle.Font,
    new SolidBrush(GridCol.InheritedStyle.ForeColor),
    new RectangleF((int)arrColumnLefts[iCount], iTopMargin,
    (int)arrColumnWidths[iCount], iHeaderHeight), strFormat);
iCount++;
}

bNewPage = false;
iTopMargin += iHeaderHeight;
}

iCount = 0;

foreach (DataGridViewCell Cel in GridRow.Cells)
{
    if (Cel.Value != null)
    {
        e.Graphics.DrawString(Cel.Value.ToString(), Cel.InheritedStyle.Font,
            new SolidBrush(Cel.InheritedStyle.ForeColor),
            new RectangleF((int)arrColumnLefts[iCount], (float)iTopMargin,
            (int)arrColumnWidths[iCount], (float)iCellHeight), strFormat);
    }
}

e.Graphics.DrawRectangle(Pens.Black, new Rectangle((int)arrColumnLefts[iCount],
    iTopMargin, (int)arrColumnWidths[iCount], iCellHeight));

iCount++;
}

}

iRow++;
iTopMargin += iCellHeight;
}

```

```

if (bMorePagesToPrint)
    e.HasMorePages = true;
else
    e.HasMorePages = false;
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void Rez_Ekle_Click(object sender, EventArgs e)
{
    r_musteri_ekle f = new r_musteri_ekle();
    f.Show();
}

private void Rez_Sil_Click(object sender, EventArgs e)
{
    Rez_Sil f = new Rez_Sil();
    f.Show();
}

private void Excel_Gonder_Click(object sender, EventArgs e)
{
    Konaklamalar form = Application.OpenForms["Konaklamalar"] as Konaklamalar;
    try
    {

        if (form.rapor.Rows.Count != 0)
        {
            Microsoft.Office.Interop.Excel.Application uyg = new
Microsoft.Office.Interop.Excel.Application();
            uyg.Visible = true;
            Microsoft.Office.Interop.Excel.Workbook kitap =
uyg.Workbooks.Add(System.Reflection.Missing.Value);
            Microsoft.Office.Interop.Excel.Worksheet sheet1 =
(Microsoft.Office.Interop.Excel.Worksheet)kitap.Sheets[1];
    }
}

```

```

int StartCol = 1;

int StartRow = 1;

for (int j = 0; j < form.rapor.ColumnCount; j++)
{
    Microsoft.Office.Interop.Excel.Range myRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow, StartCol + j];

    myRange.Value2 = form.rapor.Columns[j].HeaderText;
    myRange.ColumnWidth = 20;
}

StartRow++;

for (int i = 0; i < form.rapor.Rows.Count; i++)
{
    for (int j = 0; j < form.rapor.Columns.Count; j++)
    {

        Microsoft.Office.Interop.Excel.Range myRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow + i, StartCol + j];

        myRange.Value2 = form.rapor[j, i].Value == null ? "" : form.rapor[j, i].Value;

    }
}

}
else
{

```

```

        MessageBox.Show("Veri olmadan excel çıktısı alamazsınız!", "Uyarı",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
    }
}
catch (Exception)
{
    MessageBox.Show("Bir Hata Oluştu ", "Uyarı", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
}

private void Yazdir_Click(object sender, EventArgs e)
{
    Konaklamalar form = Application.OpenForms["Konaklamalar"] as Konaklamalar;

    PrintPreviewDialog onizleme = new PrintPreviewDialog();
    onizleme.Document = form.printDocument1;
    onizleme.ShowDialog();
}

private void timer1_Tick(object sender, EventArgs e)
{
    bunifuFlatButton5.Text = " - " + DateTime.Now.ToString("yyyy-MM-dd");
    bunifuFlatButton6.Text = " - " + DateTime.Now.ToString("dd/MM/yyyy");
    bunifuFlatButton7.Text = " - " + DateTime.Now.ToString("HH:mm:ss");
}

// o günün verisi
}

}

}

```

## **REZERVASYON.CS**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;

```

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{
    public partial class Rezervasyon : Form
    {
        public int OtelID = Login.id;
        public Rezervasyon()
        {
            InitializeComponent();
        }
        SqlConnection baglanti = new SqlConnection(Login.veritabani);

        private void Ara_Click(object sender, EventArgs e)
        {
            rapor.Rows.Clear();
            try
            {
                baglanti.Open();
                SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from RezRaporV where (TcNo='" + tc_noveyaad.Text + "' or ad='" + tc_noveyaad.Text + "') and OtelID='"
                + OtelID + " ", baglanti);
                SqlDataReader dr = komut.ExecuteReader();
                int syc = 0;
                while (dr.Read())
                {
                    rapor.Rows.Add();
                    rapor.Rows[syc].Cells[0].Value = dr[0].ToString();
                    rapor.Rows[syc].Cells[1].Value = dr[1].ToString();
                    rapor.Rows[syc].Cells[2].Value = dr[2].ToString();
                    rapor.Rows[syc].Cells[3].Value = dr[3].ToString();
                    rapor.Rows[syc].Cells[4].Value = dr[4].ToString();
                    syc++;
                }
            }
        }
    }
}

```

```

        }

        baglanti.Close();
    }

    catch (Exception)
    {
        baglanti.Open();

        SqlCommand da = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from RezRaporV
where ad=''' + tc_noveyaad.Text + ''' and OtelID=''' + OtelID + ''' ", baglanti);

        SqlDataReader sa = da.ExecuteReader();

        int a = 0;

        while (sa.Read())
        {
            rapor.Rows.Add();

            rapor.Rows[a].Cells[0].Value = sa[0].ToString();
            rapor.Rows[a].Cells[1].Value = sa[1].ToString();
            rapor.Rows[a].Cells[2].Value = sa[2].ToString();
            rapor.Rows[a].Cells[3].Value = sa[3].ToString();
            rapor.Rows[a].Cells[4].Value = sa[4].ToString();

            a++;
        }

        baglanti.Close();
    }

}

```

```

private void Tc_noveyaad_OnValueChanged(object sender, EventArgs e)
{
    if (tc_noveyaad.Text.Length > 0)
    {
        gel_aralik.Value = false;
        radiotc.Enabled = false;
        radiotc.Checked = false;
        radioyabanci.Enabled = false;
        radioyabanci.Checked = false;
        gel_tar.Enabled = false;
        gel_tar.IconColor = Color.Gray;
        gel_tar2.Enabled = false;
        gel_tar2.IconColor = Color.Gray;
    }
}

```

```

        gel_tar2.Enabled = false;
    }
    else
    {
        radiotc.Enabled = true;
        radioyabanci.Enabled = true;
        gel_tar.Enabled = true;
        gel_tar.IconColor = Color.Green;
        rapor.Rows.Clear();
    }
}

private void Rapor_Goster_Click(object sender, EventArgs e)
{
    try
    {
        gel_tar.CustomFormat = "MM/dd/yyyy";
        gel_tar2.CustomFormat = "MM/dd/yyyy";
        gel_tar.Format = DateTimePickerFormat.Custom;
        gel_tar2.Format = DateTimePickerFormat.Custom;

        if (radiotc.Checked == true)
        {
            SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where (GelisTarih BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and
uyruk='Turkey' and OtelID='" + OtelID + "'", baglanti);
            verigetir(komut);
        }
        else if (radioyabanci.Checked == true)
        {
            SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where (GelisTarih BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and
uyruk<>'Turkey' and OtelID='" + OtelID + "'", baglanti);
            verigetir(komut);
        }
        else if (radio_hepsi.Checked == true)
        {
    }
}

```

```

        SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where (GelisTarih BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and
(uyruk='Turkey' or uyruk<>'Turkey') and OtelID='" + OtelID + "'", baglanti);
        verigetir(komut);
    }
    gel_tar.Format = DateTimePickerFormat.Long;
    gel_tar2.Format = DateTimePickerFormat.Long;

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}
void verigetir(SqlCommand a)
{
    rapor.Rows.Clear();
    baglanti.Open();

    if (gel_aralik.Value == true)
    {
        SqlDataReader oku = a.ExecuteReader();
        int sayac = 0;
        while (oku.Read())
        {
            rapor.Rows.Add();
            rapor.Rows[sayac].Cells[0].Value = oku[0].ToString();
            rapor.Rows[sayac].Cells[1].Value = oku[1].ToString().Substring(0, 10);
            rapor.Rows[sayac].Cells[2].Value = oku[2].ToString();
            rapor.Rows[sayac].Cells[3].Value = oku[3].ToString();
            rapor.Rows[sayac].Cells[4].Value = oku[4].ToString();
            sayac++;
        }
    }
    else
    {

```

```

gel_tar.CustomFormat = "MM/dd/yyyy";
gel_tar2.CustomFormat = "MM/dd/yyyy";
gel_tar.Format = DateTimePickerFormat.Custom;
gel_tar2.Format = DateTimePickerFormat.Custom;
SqlDataReader oku = a.ExecuteReader();
gel_tar.Format = DateTimePickerFormat.Long;
gel_tar2.Format = DateTimePickerFormat.Long;
int sayac = 0;
while (oku.Read())
{
    rapor.Rows.Add();
    rapor.Rows[sayac].Cells[0].Value = oku[0].ToString();
    rapor.Rows[sayac].Cells[1].Value = oku[1].ToString().Substring(0, 10);
    rapor.Rows[sayac].Cells[2].Value = oku[2].ToString();
    rapor.Rows[sayac].Cells[3].Value = oku[3].ToString();
    rapor.Rows[sayac].Cells[4].Value = oku[4].ToString();
    sayac++;
}

}
baglanti.Close();
}

private void Rezervasyon_Load(object sender, EventArgs e)
{
    gel_aralik.Value = false;
    radiotc.Checked = false;
    radioyabanci.Checked = false;
    radio_hepsi.Checked = true;
    gel_tar.MinDate = DateTime.Now;
    gel_tar2.MinDate = DateTime.Now;
    baglanti.Open();
    gel_tar.CustomFormat = "MM/dd/yyyy";
    gel_tar2.CustomFormat = "MM/dd/yyyy";
    gel_tar.Format = DateTimePickerFormat.Custom;
    gel_tar2.Format = DateTimePickerFormat.Custom;
    SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,Uyruk from
RezRaporV where GelisTarih=''' + gel_tar.Text + ''' and OtelID=''' + OtelID + ''' ", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
}

```

```

gel_tar.Format = DateTimePickerFormat.Long;
gel_tar2.Format = DateTimePickerFormat.Long;
int sayac = 0;
while (oku.Read())
{
    rapor.Rows.Add();
    rapor.Rows[sayac].Cells[0].Value = oku[0].ToString();
    rapor.Rows[sayac].Cells[1].Value = oku[1].ToString().Substring(0, 10);
    rapor.Rows[sayac].Cells[2].Value = oku[2].ToString();
    rapor.Rows[sayac].Cells[3].Value = oku[3].ToString();
    rapor.Rows[sayac].Cells[4].Value = oku[4].ToString();
    sayac++;
}
baglanti.Close();

}

private void Gel_aralik_OnValuechange(object sender, EventArgs e)
{
    if (gel_aralik.Value == true)
    {
        gel_tar2.Enabled = true;
        gel_tar2.IconColor = Color.Green;
    }
    else
    {
        gel_tar2.Enabled = false;
        gel_tar2.IconColor = Color.Gray;
    }
}

private void Gel_tar_ValueChanged(object sender, EventArgs e)
{
    gel_tar2.MinDate = gel_tar.MinDate;
}

}
}

```

## R\_MUSTERİ\_EKLE.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Text.RegularExpressions;

namespace Login
{
    public partial class r_musteri_ekle : Form
    {
        public int OtelID = Login.id;
        public r_musteri_ekle()
        {
            InitializeComponent();
        }

        SqlConnection baglanti = new SqlConnection(Login.veritabani);
        SqlConnection baglanti2 = new SqlConnection(Login.veritabani);
        SqlConnection baglanti3 = new SqlConnection(Login.veritabani);
        SqlConnection baglanti4 = new SqlConnection(Login.veritabani);
        SqlConnection baglanti5 = new SqlConnection(Login.veritabani);
        SqlConnection baglanti6 = new SqlConnection(Login.veritabani);
        SqlConnection baglanti7 = new SqlConnection(Login.veritabani);
        SqlConnection baglanti8 = new SqlConnection(Login.veritabani);

        private void Kaydet_Click(object sender, EventArgs e)
        {
            //try
            //{
            if (kim_bel_tur.Text == "" || cep_tel.Text == "" || tcno.Text == "" || ad.Text == "" || soyad.Text ==
                "" || babaad.Text == "" || anaad.Text == "" || dogum_yer.Text == "" || dogum_tar.Text == "" || uyruk.Text
                == "" || cins.Text == "" || ikamet_adres.Text == "" || giris_tar.Text == "")
```

```

{
    MessageBox.Show("Lütfen kırmızı alanları boş geçmeyin", "Bilgilendirme",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
    UInt64 tc = UInt64.Parse(tcno.Text);
    baglanti2.Open();
    baglanti3.Open();
    baglanti4.Open();
    SqlCommand komut = new SqlCommand("SELECT Count(TcKimlikNo) From R_Musteriekle
where TcKimlikNo=" + tc + " and OtelID=" + OtelID + "", baglanti4);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        if (int.Parse(oku[0].ToString()) > 0)
        {
            bool mail = mailkontrol(e_mail.Text);
            if (mail == true || e_mail.Text == "")
            {
                dogum_tar.CustomFormat = "MM/dd/yyyy";
                dogum_tar.Format = DateTimePickerFormat.Custom;
                if (tcno.Text.Length > 10)
                {
                    SqlCommand cmd = new SqlCommand("update R_Musteriekle set
KimlikBelgeTuru=" + kim_bel_tur.SelectedIndex + ",KimlikSeriNo=" + kimlik_seri_no.Text +
",TcKimlikNo=" + tcno.Text + ",Ad=" + ad.Text + ",Soyad=" + soyad.Text + ",BabaAd=" +
babaad.Text + ",AnaAd=" + anaad.Text + ",Dogumyeri=" + dogum_yer.Text + ",DogumTarihi=" +
dogum_tar.Text + ",Uyruk=" + uyruk.SelectedIndex + ",Cinsiyeti=" + cins.SelectedIndex +
",MedeniHali=" + med_hal.SelectedIndex + ",KayitliOlduguil=" + il.Text + ",KayitliOlduguilce=" +
ilce.Text + ",KayitliOlduguMahalle=" + mah.Text + ",CiltNo=" + cilt_no.Text + ",AileSiraNo=" +
aile_sira_no.Text + ",SiraNo=" + int.Parse(sira_no.Text) + ",Yetiskin_Cocuk=" +
yet_cocuk.SelectedIndex + ",Telefon=" + tel.Text + ",CepTelefon=" + cep_tel.Text + ",Email=" +
e_mail.Text + ",AracPlakaNo=" + arac_plaka_no.Text + ",Musteri_is=" + isi.Text + ",ikametkah=" +
ikamet_adres.Text + ",Notlar=" + not.Text + " where TcKimlikNo=" + tc + " and OtelID=" + OtelID +
"", baglanti3);
                    dogum_tar.Format = DateTimePickerFormat.Long;
                    cmd.ExecuteNonQuery();
                    cmd.Dispose();
                }
            }
        }
    }
}

```

```

SqlCommand cmd2 = new SqlCommand("update Uyruk set uyruk='' +
comboBox1.Text + '' where KimlikNo='' + tc + '' and OtelID='' + OtelID + """", baglanti2);
cmd2.ExecuteNonQuery();
cmd2.Dispose();

giris_tar.CustomFormat = "MM/dd/yyyy";
giris_tar.Format = DateTimePickerFormat.Custom;

SqlCommand Rez_rapor = new SqlCommand("insert into RezRapor values('' + OtelID
+ "','" + tcno.Text + "','" + giris_tar.Text + "','" + ad.Text + "','" + soyad.Text + "','" + uyruk.SelectedIndex
+ "')", baglanti2);
Rez_rapor.ExecuteNonQuery();
baglanti2.Close();

giris_tar.Format = DateTimePickerFormat.Long;

MessageBox.Show("Bilgiler güncellenmiştir.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
yenile();
baglanti2.Close();
baglanti3.Close();
}

else
    MessageBox.Show("Kimlik numarası hatalıdır.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}

else
{
    MessageBox.Show("Mail adresi hatalıdır. Lütfen kontrol ediniz.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}

bool mail = mailkontrol(e_mail.Text);
if (mail == true || e_mail.Text == "")
```

```

{
    baglanti2.Close();
    baglanti.Open();
    baglanti2.Open();
    giris_tar.CustomFormat = "MM/dd/yyyy";
    dogum_tar.CustomFormat = "MM/dd/yyyy";
    giris_tar.Format = DateTimePickerFormat.Custom;
    dogum_tar.Format = DateTimePickerFormat.Custom;
    if (tcno.Text.Length > 10)
    {
        SqlCommand Rez_ekle = new SqlCommand("insert into R_Musteriekle(OtelID,
KimlikBelgeTuru, KimlikSeriNo, TcKimlikNo, Ad, Soyad, BabaAd, AnaAd, DogumYeri, DogumTarihi,
Uyruk, Cinsiyeti, MedeniHali, KayitliOlduguil, KayitliOlduguilce, KayitliOlduguMahalle, CiltNo,
AileSiraNo, SiraNo, Yetiskin_Cocuk, Telefon, CepTelefon, Email, AracPlakaNo, Musteri_is, ikametkah,
Notlar,Durum) values('" + OtelID + "','" + kim_bel_tur.SelectedIndex + "','" + kimlik_seri_no.Text + "','" +
Convert.ToInt64(tcno.Text) + "','" + ad.Text + "','" + soyad.Text + "','" + babaad.Text + "','" + anaad.Text +
"','" + dogum_yer.Text + "','" + dogum_tar.Text + "','" + uyruk.SelectedIndex + "','" + cins.SelectedIndex +
"','" + med_hal.SelectedIndex + "','" + il.Text + "','" + ilce.Text + "','" + mah.Text + "','" + cilt_no.Text +
"','" + aile_sira_no.Text + "','" + sira_no.Text + "','" + yet_cocuk.SelectedIndex + "','" + tel.Text + "','" +
cep_tel.Text + "','" + e_mail.Text + "','" + arac_plaka_no.Text + "','" + isi.Text + "','" + ikamet_adres.Text +
"','" + not.Text + "',1)", baglanti);
        Rez_ekle.ExecuteNonQuery();
        baglanti.Close();

        SqlCommand Rez_rapor = new SqlCommand("insert into RezRapor values('" + OtelID +
+ "','" + tcno.Text + "','" + giris_tar.Text + "','" + ad.Text + "','" + soyad.Text + "','" + uyruk.SelectedIndex +
+ "')", baglanti2);
        Rez_rapor.ExecuteNonQuery();
        baglanti2.Close();

        baglanti8.Open();
        SqlCommand uyrukekle = new SqlCommand("insert into Uyruk values('" + OtelID +
"','" + tcno.Text + "','" + comboBox1.Text + "')", baglanti8);
        uyrukekle.ExecuteNonQuery();
        baglanti8.Close();

        dogum_tar.Format = DateTimePickerFormat.Long;
        giris_tar.Format = DateTimePickerFormat.Long;
}

```

```

        MessageBox.Show("Rezervasyon başarıyla oluşturulmuştur.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    yenile();
}
else
    MessageBox.Show("Kimlik numarası hatalıdır.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    baglanti.Close();
}
else
{
    MessageBox.Show("Mail adresi hatalıdır. Lütfen kontrol ediniz.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

baglanti2.Close();
baglanti3.Close();
baglanti4.Close();
}
// }
//catch (Exception ex)
//{
//    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
//}
}

```

```

void yenile()
{
    kim_bel_tur.SelectedItem = null;
    kimlik_seri_no.Text = "";
    tcno.Text = "";
    ad.Text = "";
    soyad.Text = "";
    babaad.Text = "";
    anaad.Text = "";
}

```

```

dogum_tar.Text = "";
dogum_yer.Text = "";
uyruk.SelectedItem = null;
cins.SelectedItem = null;
med_hal.SelectedItem = null;
il.SelectedItem = null;
ilce.SelectedItem = null;
mah.SelectedItem = null;
ilce.Items.Clear();
mah.Items.Clear();
cilt_no.Text = "";
aile_sira_no.Text = "";
sira_no.Text = "";
yet_cocuk.SelectedItem = null;
tel.Text = "";
cep_tel.Text = "";
e_mail.Text = "";
arac_plaka_no.Text = "";
isi.Text = "";
ikamet_adres.Text = "";
not.Text = "";

comboBox1.Enabled = false;
comboBox1.SelectedItem = "Turkey";
}

public static bool mailkontrol(string Telefon)
{
    string mailvalidation = @"^([\w\.-]+)@([\w\.-]+)(\.(.\w){2,3})+$";
    Match Eslesme = Regex.Match(Telefon, mailvalidation, RegexOptions.IgnoreCase);
    return Eslesme.Success; // bool değer döner
}

private void Temizle_Click(object sender, EventArgs e)
{
    yenile();
}

private void R_musteri_ekle_Load(object sender, EventArgs e)
{
}

```

```

dogum_tar.MaxValue = DateTime.Now;
giris_tar.MinValue = DateTime.Now;
tcno.MaxLength = 11;
kimlik_seri_no.MaxLength = 9;

baglanti.Open();
SqlCommand illiste = new SqlCommand("select CityName from City", baglanti);
SqlDataReader iller = illiste.ExecuteReader();
while(iller.Read())
{
    il.Items.Add(iller[0].ToString());
}
baglanti.Close();

baglanti.Open();
SqlCommand ulkeliste = new SqlCommand("select CountryName from Country", baglanti);
SqlDataReader ulkeler = ulkeliste.ExecuteReader();
while (ulkeler.Read())
{
    comboBox1.Items.Add(ulkeler[0].ToString());
}
baglanti.Close();

comboBox1.SelectedItem = "Turkey";
}

private void Iptal_Click(object sender, EventArgs e)
{
    this.Hide();
}

private void Tcno_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

private void Cilt_no_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

```

```

void sayi_KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

void harf_KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);
}

private void Aile_sıra_no_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

private void Sıra_no_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

private void Tel_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

private void Cep_tel_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

private void Ad_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}

private void Soyad_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}

```

```
private void Babaad_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}
```

```
private void Anaad_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}
```

```
private void İl_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}
```

```
private void İlce_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}
```

```
private void Mah_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}
```

```
private void E_mail_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}
```

```
private void İsi_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}
```

```
private void İkamet_adres_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}
```

```

}

private void Not_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}

private void Dogum_yer_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}

private void Tcno_TextChanged(object sender, EventArgs e)
{
    if (tcno.Text.Length == 11)
    {
        UInt64 tc = Convert.ToInt64(tcno.Text);
        baglanti.Open();
        SqlCommand komut = new SqlCommand("SELECT * From R_Musteriekle where
TcKimlikNo=" + tc + " and OtelID=" + OtelID + "", baglanti);
        SqlDataReader oku = komut.ExecuteReader();
        while (oku.Read())
        {
            Boolean a = Boolean.Parse(oku["KimlikBelgeTuru"].ToString());
            if (a == true)
                kim_bel_tur.Text = "T.C Kimlik Belgesi";
            else if (a == false)
                kim_bel_tur.Text = "Yabancı Kimlik";

            kimlik_seri_no.Text = oku["KimlikSeriNo"].ToString();
            tcno.Text = oku["TcKimlikNo"].ToString();
            ad.Text = oku["Ad"].ToString();
            soyad.Text = oku["Soyad"].ToString();
            babaad.Text = oku["BabaAd"].ToString();
            anaad.Text = oku["AnaAd"].ToString();
            dogum_yer.Text = oku["DogumYeri"].ToString();
            dogum_tar.Value = Convert.ToDateTime(oku["DogumTarihi"].ToString());

            Boolean b = Boolean.Parse(oku["Uyruk"].ToString());
        }
    }
}

```

```

if (b == true)
    uyruk.Text = "Türk";
else if (b == false)
    uyruk.Text = "Yabancı";

Boolean c = Boolean.Parse(oku["Cinsiyeti"].ToString());
if (c == true)
    cins.Text = "Erkek";
else if (c == false)
    cins.Text = "Kadın";

Boolean d = Boolean.Parse(oku["MedeniHali"].ToString());
if (d == true)
    med_hal.Text = "Evli";
else if (d == false)
    med_hal.Text = "Bekar";

il.Text = oku["KayıtlıOlduguil"].ToString();
ilce.Text = oku["KayıtlıOlduguilce"].ToString();
mah.Text = oku["KayıtlıOlduguMahalle"].ToString();
cilt_no.Text = oku["CiltNo"].ToString();
aile_sıra_no.Text = oku["AileSıraNo"].ToString();
sıra_no.Text = oku["SıraNo"].ToString();

Boolean h = Boolean.Parse(oku["Yetiskin_Cocuk"].ToString());
if (h == true)
    cins.Text = "Yetişkin";
else if (h == false)
    cins.Text = "Çocuk";

tel.Text = oku["Telefon"].ToString();
cep_tel.Text = oku["CepTelefon"].ToString();
e_mail.Text = oku["Email"].ToString();
arac_plaka_no.Text = oku["AracPlakaNo"].ToString();
isi.Text = oku["Musteri_is"].ToString();
ikamet_adres.Text = oku["ikametkah"].ToString();
not.Text = oku["Notlar"].ToString();
}

```

```

        baglanti7.Open();
        SqlCommand komut2 = new SqlCommand("SELECT uyruk From Uyruk where KimlikNo='"
+ tc + "' and OtelID='"
+ OtelID + "'", baglanti7);
        SqlDataReader oku2 = komut2.ExecuteReader();
        while (oku2.Read())
        {
            comboBox1.SelectedItem = oku2[0].ToString();
        }
        baglanti7.Close();

    }
    baglanti.Close();
}

```

```

private void il_SelectedIndexChanged(object sender, EventArgs e)
{
    ilce.Items.Clear();

    int ilid=0;
    baglanti5.Open();
    SqlCommand ilidal = new SqlCommand("select CityID from City where CityName='"
+ il.Text +
"', baglanti5);
    SqlDataReader ilidgetir = ilidal.ExecuteReader();
    while(ilidgetir.Read())
    {
        ilid = int.Parse(ilidgetir[0].ToString());
    }
    baglanti5.Close();
    baglanti5.Open();
    SqlCommand ilceliste = new SqlCommand("select TownName from Town where CityID='"
+ ilid.ToString() + "'", baglanti5);
    SqlDataReader ilcedoldur = ilceliste.ExecuteReader();
    while(ilcedoldur.Read())
    {
        ilce.Items.Add(ilcedoldur[0].ToString());
    }
    baglanti5.Close();
}

```

```

}

private void ilce_SelectedIndexChanged(object sender, EventArgs e)
{
    mah.Items.Clear();
    int ilceid = 0;
    int ilid = 0;
    baglanti6.Open();
    SqlCommand ilidal = new SqlCommand("select CityID from City where CityName='" + il.Text +
    "'", baglanti6);
    SqlDataReader ilidgetir = ilidal.ExecuteReader();
    while (ilidgetir.Read())
    {
        ilid = int.Parse(ilidgetir[0].ToString());
    }
    baglanti6.Close();
    baglanti6.Open();
    SqlCommand ilceliste = new SqlCommand("select TownID from Town where CityID=" +
    ilid.ToString() + " and TownName='"++ilce.Text+"'", baglanti6);
    SqlDataReader ilcedoldur = ilceliste.ExecuteReader();
    while (ilcedoldur.Read())
    {
        ilceid=int.Parse(ilcedoldur[0].ToString());
    }
    baglanti6.Close();
    baglanti6.Open();
    SqlCommand mahalleal = new SqlCommand("select DistrictName from District where
    TownID='"++ilceid.ToString()+"'", baglanti6);
    SqlDataReader mahallegetir = mahalleal.ExecuteReader();
    while (mahallegetir.Read())
    {
        mah.Items.Add(mahallegetir[0].ToString());
    }
    baglanti6.Close();
}

private void kim_bel_tur_SelectedIndexChanged(object sender, EventArgs e)
{
    if(kim_bel_tur.SelectedIndex==0)

```

```

{
    label3.Text = "Pasaport No:";
    label2.Text = "Pasaport Seri No:";
}
else
{
    label3.Text = "Kimlik No:";
    label2.Text = "Kimlik Seri No:";
}
}

private void uyruk_SelectedIndexChanged(object sender, EventArgs e)
{
    if (uyruk.SelectedIndex == 0)
    {
        comboBox1.Enabled = true;
    }
    else
    {
        comboBox1.Enabled = false;
        comboBox1.SelectedItem = "Turkey";
    }
}
}

REZ_SİL.CS
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{

```

```

public partial class Rez_Sil : Form
{
    public int OtelID = Login.id;
    public Rez_Sil()
    {
        InitializeComponent();
    }
    SqlConnection baglanti = new SqlConnection(Login.veritabani);
    private void Rez_Sil_Load(object sender, EventArgs e)
    {
        yenile();
    }

    private void Veri_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
    {
        try
        {
            baglanti.Open();
            int a = veri.CurrentRow.Index;
            string b = veri.Rows[a].Cells[0].Value.ToString();
            DialogResult cevap = MessageBox.Show("Rezervasyonu Silmek İstediğinizden Emin Misiniz ?",
                "Bilgilendirme", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (cevap == DialogResult.Yes)
            {
                SqlCommand komut = new SqlCommand("delete from RezRapor where TcNo='" + b + "' and
OtelID=" + OtelID + "", baglanti);
                komut.ExecuteNonQuery();
                MessageBox.Show("Rezervasyon Silinmiştir.");
                baglanti.Close();
                yenile();
            }
            else if (cevap == DialogResult.No)
            {
                baglanti.Close();
            }
        }
        catch (Exception ex)
        {
    }
}

```

```

        MessageBox.Show(ex.Message,"Hata",MessageBoxButtons.OK,MessageBoxIcon.Error);
    }

}

void yenile()
{
    baglanti.Open();
    SqlDataAdapter da = new SqlDataAdapter("select TcNo,GelisTarih,ad,soyad from RezRapor where
OtelID=''' + OtelID + '''", baglanti);
    DataTable dt = new DataTable();
    da.Fill(dt);
    veri.DataSource = dt;
    veri.Columns[0].HeaderText="Kimlik Numarası";
    veri.Columns[0].Width = 150;
    veri.Columns[1].HeaderText = "Geliş Tarihi";
    veri.Columns[1].Width = 200;
    veri.Columns[2].HeaderText = "Adı";
    veri.Columns[2].Width = 150;
    veri.Columns[3].HeaderText = "Soyadı";
    veri.Columns[3].Width = 150;
    baglanti.Close();
}
}
}

```

## REZRAPOR.CS

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Collections;

```

```

namespace Login
{
    public partial class RezRapor : Form
    {
        public int OtelID = Login.id;
        public RezRapor()
        {
            InitializeComponent();
        }
        SqlConnection baglanti = new SqlConnection(Login.veritabani);
        SqlConnection baglanti2 = new SqlConnection(Login.veritabani);

        private void bunifuToggleSwitch1_ValueChanged(object sender, EventArgs e)
        {
            if (gel_aralik.Value == true)
            {
                gel_tar2.Enabled = true;
                gel_tar2.IconColor = Color.Green;
            }
            else
            {
                gel_tar2.Enabled = false;
                gel_tar2.IconColor = Color.Gray;
            }
        }

        private void RezRapor_Load(object sender, EventArgs e)
        {
            gel_tar.MinDate = DateTime.Now;
            gel_tar2.MinDate = gel_tar.Value;
            gel_aralik.Value = false;
            radio_hepsi.Checked = true;
            radio_tc.Checked = false;
            baglanti.Open();
            gel_tar.CustomFormat = "MM/dd/yyyy";
            gel_tar2.CustomFormat = "MM/dd/yyyy";
            gel_tar.Format = DateTimePickerFormat.Custom;
            gel_tar2.Format = DateTimePickerFormat.Custom;
        }
    }
}

```

```

SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,Uyruk from
RezRaporV where GelisTarih=''' + gel_tar.Text + ''' and OtelID=''' + OtelID + ''' ", baglanti);
SqlDataReader oku = komut.ExecuteReader();
gel_tar.Format = DateTimePickerFormat.Long;
gel_tar2.Format = DateTimePickerFormat.Long;
int sayac = 0;
while (oku.Read())
{
    rapor.Rows.Add();
    rapor.Rows[sayac].Cells[0].Value = oku[0].ToString();
    rapor.Rows[sayac].Cells[1].Value = oku[1].ToString().Substring(0, 10);
    rapor.Rows[sayac].Cells[2].Value = oku[2].ToString();
    rapor.Rows[sayac].Cells[3].Value = oku[3].ToString();
    rapor.Rows[sayac].Cells[4].Value = oku[4].ToString();
    sayac++;
}
baglanti.Close();
}

private void PictureBox1_Click(object sender, EventArgs e)
{
    rapor.Rows.Clear();
    try
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where (TcNo=''' + tc_noveyaad.Text + ''' or ad=''' + tc_noveyaad.Text + ''') and OtelID=''' +
OtelID + ''' ", baglanti);
        SqlDataReader dr = komut.ExecuteReader();
        int syc = 0;
        while (dr.Read())
        {
            rapor.Rows.Add();
            rapor.Rows[syc].Cells[0].Value = dr[0].ToString();
            rapor.Rows[syc].Cells[1].Value = dr[1].ToString();
            rapor.Rows[syc].Cells[2].Value = dr[2].ToString();
            rapor.Rows[syc].Cells[3].Value = dr[3].ToString();
            rapor.Rows[syc].Cells[4].Value = dr[4].ToString();
            syc++;
        }
    }
}
```

```

        }

        baglanti.Close();
    }

    catch (Exception)
    {
        baglanti2.Open();

        SqlCommand da = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from RezRaporV
where ad=''' + tc_noveyaad.Text + ''' and OtelID=''' + OtelID + ''' ", baglanti2);

        SqlDataReader sa = da.ExecuteReader();

        int a = 0;

        while (sa.Read())
        {
            rapor.Rows.Add();

            rapor.Rows[a].Cells[0].Value = sa[0].ToString();
            rapor.Rows[a].Cells[1].Value = sa[1].ToString();
            rapor.Rows[a].Cells[2].Value = sa[2].ToString();
            rapor.Rows[a].Cells[3].Value = sa[3].ToString();
            rapor.Rows[a].Cells[4].Value = sa[4].ToString();

            a++;
        }

        baglanti2.Close();
    }

}

```

```

private void Tc_noveyaad_OnValueChanged(object sender, EventArgs e)
{
    if (tc_noveyaad.Text.Length > 0)
    {
        gel_aralik.Value = false;
        radio_tc.Enabled = false;
        radio_tc.Checked = false;
        radio_yabanci.Enabled = false;
        radio_yabanci.Checked = false;
        gel_tar.Enabled = false;
        gel_tar.IconColor = Color.Gray;
        gel_tar2.Enabled = false;
    }
}

```

```

        gel_tar2.IconColor = Color.Gray;
        gel_tar2.Enabled = false;
    }
    else
    {
        gel_aralik.Value = true;
        radio_tc.Enabled = true;
        radio_yabanci.Enabled = true;
        gel_tar.Enabled = true;
        gel_tar.IconColor = Color.Green;
        gel_tar2.Enabled = true;
        gel_tar2.IconColor = Color.Green;
        gel_tar2.Enabled = true;
    }
}

private void Gel_tar_ValueChanged(object sender, EventArgs e)
{
    gel_tar2.MinDate = gel_tar.MinDate;
}

private void Rapor_Goster_Click(object sender, EventArgs e)
{
    try
    {
        if (gel_aralik.Value == true)
        {
            gel_tar.CustomFormat = "MM/dd/yyyy";
            gel_tar2.CustomFormat = "MM/dd/yyyy";
            gel_tar.Format = DateTimePickerFormat.Custom;
            gel_tar2.Format = DateTimePickerFormat.Custom;

            if (radio_tc.Checked == true)
            {
                SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from RezRaporV where (GelisTarih BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and uyruk='Turkey' and OtelID='" + OtelID + "'", baglanti);
                getir1(komut);
            }
            else if (radio_yabanci.Checked == true)

```

```

{
    SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where (GelisTarih BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and
uyruk<>'Turkey' and OtelID=''' + OtelID + '''', baglanti);
    getir1(komut);
}
else if (radio_hepsi.Checked == true)
{
    SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where (GelisTarih BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and
(uyruk='Turkey' or uyruk<>'Turkey') and OtelID=''' + OtelID + '''', baglanti);
    getir1(komut);
}
gel_tar.Format = DateTimePickerFormat.Long;
gel_tar2.Format = DateTimePickerFormat.Long;
}
else
{
    gel_tar.CustomFormat = "MM/dd/yyyy";
    gel_tar.Format = DateTimePickerFormat.Custom;
    if (radio_tc.Checked == true)
    {
        SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where GelisTarih=''' + gel_tar.Text + ''' and uyruk='Turkey' and OtelID=''' + OtelID + '''',
baglanti);
        getir2(komut);
    }
    else if (radio_yabanci.Checked == true)
    {
        SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where GelisTarih=''' + gel_tar.Text + ''' and uyruk<>'Turkey' and OtelID=''' + OtelID + '''',
baglanti);
        getir2(komut);
    }
    else if (radio_hepsi.Checked == true)
    {
        SqlCommand komut = new SqlCommand("select TcNo,GelisTarih,ad,soyad,uyruk from
RezRaporV where GelisTarih=''' + gel_tar.Text + ''' and (uyruk='Turkey' or uyruk<>'Turkey') and
OtelID=''' + OtelID + '''', baglanti);
    }
}

```

```

        getir2(komut);
    }
    gel_tar.Format = DateTimePickerFormat.Long;

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

void getir1(SqlCommand a)
{
    rapor.Rows.Clear();
    baglanti.Open();
    gel_tar.CustomFormat = "MM/dd/yyyy";
    gel_tar2.CustomFormat = "MM/dd/yyyy";
    gel_tar.Format = DateTimePickerFormat.Custom;
    gel_tar2.Format = DateTimePickerFormat.Custom;
    SqlDataReader oku = a.ExecuteReader();
    gel_tar.Format = DateTimePickerFormat.Long;
    gel_tar2.Format = DateTimePickerFormat.Long;
    int sayac = 0;
    while (oku.Read())
    {
        rapor.Rows.Add();
        rapor.Rows[sayac].Cells[0].Value = oku[0].ToString();
        rapor.Rows[sayac].Cells[1].Value = oku[1].ToString().Substring(0, 10);
        rapor.Rows[sayac].Cells[2].Value = oku[2].ToString();
        rapor.Rows[sayac].Cells[3].Value = oku[3].ToString();
        rapor.Rows[sayac].Cells[4].Value = oku[4].ToString();
        sayac++;
    }
    baglanti.Close();
}

void getir2(SqlCommand b)
{

```

```

rapor.Rows.Clear();
baglanti.Open();
gel_tar.CustomFormat = "MM/dd/yyyy";
gel_tar.Format = DateTimePickerFormat.Custom;
SqlDataReader oku = b.ExecuteReader();
gel_tar.Format = DateTimePickerFormat.Long;
int sayac = 0;
while (oku.Read())
{
    rapor.Rows.Add();
    rapor.Rows[sayac].Cells[0].Value = oku[0].ToString();
    rapor.Rows[sayac].Cells[1].Value = oku[1].ToString().Substring(0, 10);
    rapor.Rows[sayac].Cells[2].Value = oku[2].ToString();
    rapor.Rows[sayac].Cells[3].Value = oku[3].ToString();
    rapor.Rows[sayac].Cells[4].Value = oku[4].ToString();
    sayac++;
}
baglanti.Close();
}

```

```

private void Excel_Aktar_Click(object sender, EventArgs e)
{
    try
    {
        if (rapor.Rows.Count != 0)
        {
            Microsoft.Office.Interop.Excel.Application uyg = new
Microsoft.Office.Interop.Excel.Application();
            uyg.Visible = true;
            Microsoft.Office.Interop.Excel.Workbook kitap =
uyg.Workbooks.Add(System.Reflection.Missing.Value);
            Microsoft.Office.Interop.Excel.Worksheet sheet1 =
(Microsoft.Office.Interop.Excel.Worksheet)kitap.Sheets[1];
            int StartCol = 1;
            int StartRow = 1;
            for (int j = 0; j < rapor.ColumnCount; j++)
            {

```

```

        Microsoft.Office.Interop.Excel.Range myRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow, StartCol + j];
        myRange.Value2 = rapor.Columns[j].HeaderText;
        myRange.ColumnWidth = 25;
    }
    StartRow++;
    for (int i = 0; i < rapor.Rows.Count; i++)
    {
        for (int j = 0; j < rapor.Columns.Count; j++)
        {
            Microsoft.Office.Interop.Excel.Range myRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow + i, StartCol + j];
            myRange.Value2 = rapor[j, i].Value == null ? "" : rapor[j, i].Value;
        }
    }
}
else
    MessageBox.Show("Veri olmadan excel çıktısı alamazsınız!", "Uyarı",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
}
catch (Exception)
{
    MessageBox.Show("Bir Hata Oluştu ", "Uyarı", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
}
}

//YAZDIRMAK İÇİN
//https://social.msdn.microsoft.com/Forums/tr-TR/2d97993e-96a0-4524-ad2c-b0e82a0a9238/datagrid-deki-bilgileri-yazdrma?forum=csharptr
```

```

StringFormat strFormat;
ArrayList arrColumnLefts = new ArrayList();
ArrayList arrColumnWidths = new ArrayList();
int iCellHeight = 0;
int iTotWidth = 0;
int iRow = 0;
bool bFirstPage = false;
bool bNewPage = false;
```

```

int iHeaderHeight = 0;
private void Yazdir_Click(object sender, EventArgs e)
{
    PrintPreviewDialog onizleme = new PrintPreviewDialog();
    onizleme.Document = printDocument1;
    onizleme.ShowDialog();
}
private void PrintDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
{
    try
    {
        int iLeftMargin = e.MarginBounds.Left;
        int iTopMargin = e.MarginBounds.Top;
        bool bMorePagesToPrint = false;
        int iTmpWidth = 0;
        bFirstPage = true;

        if (bFirstPage)
        {
            foreach (DataGridViewColumn GridCol in rapor.Columns)
            {
                iTmpWidth = (int)(Math.Floor((double)((double)GridCol.Width /
                    (double)iTotalWidth * (double)iTotalWidth *
                    (double)e.MarginBounds.Width / (double)iTotalWidth)));
                iHeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,
                    GridCol.InheritedStyle.Font, iTmpWidth).Height) + 11;

                arrColumnLefts.Add(iLeftMargin);
                arrColumnWidths.Add(iTmpWidth);
                iLeftMargin += iTmpWidth;
            }
        }

        while (iRow <= rapor.Rows.Count - 1)
        {
            DataGridViewRow GridRow = rapor.Rows[iRow];

```

```

iCellHeight = GridRow.Height + 5;
int iCount = 0;

if (iTopMargin + iCellHeight >= e.MarginBounds.Height + e.MarginBounds.Top)
{
    bNewPage = true;
    bFirstPage = false;
    bMorePagesToPrint = true;
    break;
}
else
{
    if (bNewPage)
    {

        e.Graphics.DrawString("Rezervasyon Raporu", new Font(rapor.Font, FontStyle.Bold),
            Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -
            e.Graphics.MeasureString("Rezervasyon Raporu", new Font(rapor.Font,
            FontStyle.Bold), e.MarginBounds.Width).Height - 13);

        String strDate = DateTime.Now.ToString(" " +
DateTime.Now.ToString(""));

        e.Graphics.DrawString(strDate, new Font(rapor.Font, FontStyle.Bold),
            Brushes.Black, e.MarginBounds.Left + (e.MarginBounds.Width -
            e.Graphics.MeasureString(strDate, new Font(rapor.Font,
            FontStyle.Bold), e.MarginBounds.Width).Width), e.MarginBounds.Top -
            e.Graphics.MeasureString("Rezervasyon Raporu", new Font(new Font(rapor.Font,
            FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);

    }
}

iTopMargin = e.MarginBounds.Top;
foreach (DataGridViewColumn GridCol in rapor.Columns)
{
    e.Graphics.FillRectangle(new SolidBrush(Color.LightGray),
        new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
        (int)arrColumnWidths[iCount], iHeaderHeight));
}

```

```

e.Graphics.DrawRectangle(Pens.Black,
    new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
    (int)arrColumnWidths[iCount], iHeaderHeight));

e.Graphics.DrawString(GridCol.HeaderText, GridCol.InheritedStyle.Font,
    new SolidBrush(GridCol.InheritedStyle.ForeColor),
    new RectangleF((int)arrColumnLefts[iCount], iTopMargin,
    (int)arrColumnWidths[iCount], iHeaderHeight), strFormat);
iCount++;
}
bNewPage = false;
iTopMargin += iHeaderHeight;
}
iCount = 0;

foreach (DataGridViewCell Cel in GridRow.Cells)
{
    if (Cel.Value != null)
    {
        e.Graphics.DrawString(Cel.Value.ToString(), Cel.InheritedStyle.Font,
            new SolidBrush(Cel.InheritedStyle.ForeColor),
            new RectangleF((int)arrColumnLefts[iCount], (float)iTopMargin,
            (int)arrColumnWidths[iCount], (float)iCellHeight), strFormat);
    }
}

e.Graphics.DrawRectangle(Pens.Black, new Rectangle((int)arrColumnLefts[iCount],
    iTopMargin, (int)arrColumnWidths[iCount], iCellHeight));

iCount++;
}
}
iRow++;
iTopMargin += iCellHeight;
}

if (bMorePagesToPrint)
    e.HasMorePages = true;
else

```

```

        e.HasMorePages = false;
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void PrintDocument1_BeginPrint(object sender, System.Drawing.Printing.PrintEventArgs e)
{
    try
    {
        strFormat = new StringFormat();
        strFormat.Alignment = StringAlignment.Near;
        strFormat.LineAlignment = StringAlignment.Center;
        strFormat.Trimming = StringTrimming.EllipsisCharacter;

        arrColumnLefts.Clear();
        arrColumnWidths.Clear();
        iCellHeight = 0;
        iRow = 0;
        bFirstPage = true;
        bNewPage = true;

        iTotWidth = 0;
        foreach (DataGridViewColumn dgvGridCol in rapor.Columns)
        {
            iTotWidth += dgvGridCol.Width;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void radio_tc_Click(object sender, EventArgs e)
{
}

```

```
    }  
}
```

## RESEPSİYON.CS

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using Bunifu.UI.WinForms.BunifuButton;  
using Microsoft.VisualBasic.PowerPacks;  
using System.Data.SqlClient;  
using System.Threading;  
  
namespace Login  
{  
    public partial class Resepsiyon : Form  
    {  
  
        public Resepsiyon()  
        {  
            InitializeComponent();  
        }  
        SqlConnection con = new SqlConnection(Login.veritabani);  
        SqlConnection con2 = new SqlConnection(Login.veritabani);  
        SqlConnection con3 = new SqlConnection(Login.veritabani);  
        SqlConnection con4 = new SqlConnection(Login.veritabani);  
        SqlConnection con5 = new SqlConnection(Login.veritabani);  
        SqlConnection con6 = new SqlConnection(Login.veritabani);  
        SqlConnection con7 = new SqlConnection(Login.veritabani);  
  
        int OtelId = Login.id;
```

```

odaicerik f = new odaicerik();

private void tikla(object sender, EventArgs e)
{
    f.Hide();
}

public static Point GetMousePositionWindowsForms()
{
    System.Drawing.Point point = Control.MousePosition;
    return new Point(point.X, point.Y);
}

private void Btndn_MouseLeave(object sender, EventArgs e)
{
    f.Hide();
}

private void Btndn_DoubleClick(object sender, EventArgs e)
{

    BunifuButton btndn = sender as BunifuButton;
    string metin = btndn.ButtonText;
    string[] parca = metin.Split('-');
    int splitodano = int.Parse(parca[0]);
    try
    {
        SqlCommand cmd = new SqlCommand("Select Durum From Odalar where OdaNo=" +
splitodano + " and OtelID=" + OtelId + "", con);
        con.Open();
        SqlDataReader dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            if (Convert.ToInt32(dr[0].ToString()) == 1 && btndn.BackColor ==
System.Drawing.Color.FromArgb(51, 122, 183))
            {
                goto git;
            }
        }
    }
}

```

```

        }

        else if (Convert.ToInt32(dr[0].ToString()) == 2 && btndn.BackColor ==
Color.MediumSeaGreen)
{
    goto git;

}

else
{
    MessageBox.Show("Kirli veya arızalı odaya giriş yapamazsınız", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
    goto git2;
}
}

git:

r_konak r_Konak = new r_konak();
r_Konak.Show();
r_konak r_Konak2 = Application.OpenForms["r_konak"] as r_konak;
r_Konak2.cmbodano.Text = splitodano.ToString();

git2::
con.Close();

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

void gorunum()
{
    this.DataGridViewFolio.DefaultCellStyle.Font = new Font("Century Gothic", 10);
    this.DataGridView3Gun.DefaultCellStyle.Font = new Font("Century Gothic", 10);
}

```

```

this.DataGridView1Hafta.DefaultCellStyle.Font = new Font("Century Gothic", 10);
this.DataGridViewHatirlatmalar.DefaultCellStyle.Font = new Font("Century Gothic", 10);

}//bunifudatagridviewlerin font tipini belirledim
void folioyenileme()
{
    SqlDataAdapter da = new SqlDataAdapter("select TOP 0 TcKimlikNo as
TcKimlikNumarası,Ad,Soyad,GelisTarihi as GelişTarihi from folyo ", con2);
    DataSet ds = new DataSet();
    da.Fill(ds);
    DataGridViewFolio.DataSource = ds.Tables[0];
}

private void Resepsiyon_Load(object sender, EventArgs e)
{

// int OtelId = Convert.ToInt32(label8.Text.ToString());

    Control.CheckForIllegalCrossThreadCalls = false;
    try
    {

        DateTime myDateTime = DateTime.Now.Date;
        string sqlFormattedDate = myDateTime.ToString("yyyy-MM-dd");

        DateTime myDateTime2 = DateTime.Now.Date.AddDays(+3);
        string sqlFormattedDate2 = myDateTime2.ToString("yyyy-MM-dd");

        int a = 0;

        SqlCommand gun = new SqlCommand("Select OdaNo,GidisTarihi,Ad from gun3 where
(GidisTarihi Between '" + sqlFormattedDate + "' and '" + sqlFormattedDate2 + "') and
OtelID='"++OtelId+"'", con);
        con.Open();
        SqlDataReader gunreader = gun.ExecuteReader();
        while (gunreader.Read())
        {

```

```

        DataGridView3Gun.Rows.Add();
        DataGridView3Gun.Rows[a].Cells[0].Value = gunreader[0].ToString();
        DataGridView3Gun.Rows[a].Cells[1].Value =
Convert.ToDateTime(gunreader[1].ToString()).ToString("yyyy.MM.dd");
        DataGridView3Gun.Rows[a].Cells[2].Value = gunreader[2].ToString();

        a++;
    }
    con.Close();

```

```

DateTime date = DateTime.Now.Date;
string tarih = date.ToString("yyyy-MM-dd");
DateTime date2 = DateTime.Now.Date.AddDays(+7);
string tarih2= date2.ToString("yyyy-MM-dd");
int b = 0;
SqlCommand hafta = new SqlCommand("Select GelisTarih,ad,soyad from RezRapor where
(GelisTarih Between '" + tarih + "' and '" + tarih2 + "') and OtelID=''" + OtelId + "", con2);
con2.Open();
SqlDataReader haftareader = hafta.ExecuteReader();
while (haftareader.Read())
{
    DataGridView1Hafta.Rows.Add();
    DataGridView1Hafta.Rows[b].Cells[0].Value =
Convert.ToDateTime(haftareader[0].ToString()).ToString("yyyy.MM.dd");
    DataGridView1Hafta.Rows[b].Cells[1].Value = haftareader[1].ToString();
    DataGridView1Hafta.Rows[b].Cells[2].Value = haftareader[2].ToString();

    b++;
}
con2.Close();

gorunum();

```

```

SqlCommand hatirlatma = new SqlCommand("Select Tarih,HatirlatmaBaslik,HatirlatmaNo from
Hatirlatmalar where OtelID=''" + OtelId + "", con);

```

```

con.Open();
SqlDataReader hatirladr = hatirlatma.ExecuteReader();
int k = 0;
while (hatirladr.Read())
{
    DataGridViewHatirlatmalar.Rows.Add();
    DataGridViewHatirlatmalar.Rows[k].Cells[0].Value =
Convert.ToDateTime(hatirladr[0].ToString());
    DataGridViewHatirlatmalar.Rows[k].Cells[1].Value = hatirladr[1].ToString();
    DataGridViewHatirlatmalar.Rows[k].Cells[2].Value = hatirladr[2].ToString();
    k++;
}
con.Close();

con.Open();
SqlCommand cmd = new SqlCommand("select DISTINCT(Kat) from Odalar where OtelId=" + 
OtelId + "", con); // kat bilgisini buradan alıyoruz

SqlDataReader sdr = cmd.ExecuteReader();

while (sdr.Read())
{
    TabPage tabPage1 = new TabPage();
    tabPage1.Name = "tabPage" + sdr[0].ToString(); //tabpagelere kat ismini atıyarak
    tabPage1.Text = "KAT " + sdr[0].ToString();
    tabPage1.Font = new Font("Verdana", 12);
    tabControl1.TabPages.Add(tabPage1);

    FlowLayoutPanel flp1 = new FlowLayoutPanel();
    flp1.BackColor = Color.WhiteSmoke;
    Controls.Add(flp1);
    tabPage1.Controls.Add(flp1);
    flp1.Dock = DockStyle.Fill;
}

```

```

flp1.AutoScroll = true;

SqlCommand cmd3 = new SqlCommand("select OdaNo,OdaTip,Durum from Odalar where
OtelID=' + OtelId + '' and Kat=' + sdr[0].ToString() + '' ORDER BY OdaNo ASC", con2);
//kata göre odaları ekleyen sorgumuz
con2.Open();
SqlDataReader sdr3 = cmd3.ExecuteReader();
int i = 1;
while (sdr3.Read())
{
    BunifuButton btndn = new BunifuButton();

    btndn.Margin = new System.Windows.Forms.Padding(5);
    btndn.IdleBorderThickness = 1;
    if (i == 1)
    {
        Thread.Sleep(50); i++;
    }
    if (int.Parse(sdr3[1].ToString()) == 1)
    {
        btndn.ButtonText = sdr3[0].ToString() + " - " + "SGL";
    }
    else if (int.Parse(sdr3[1].ToString()) == 2)
    {
        btndn.ButtonText = sdr3[0].ToString() + " - " + "TSU";
    }
    else if (int.Parse(sdr3[1].ToString()) == 3)
    {
        btndn.ButtonText = sdr3[0].ToString() + " - " + "TWR";
    }
    else if (int.Parse(sdr3[1].ToString()) == 4)
    {
        btndn.ButtonText = sdr3[0].ToString() + " - " + "DBL";
    }
    else if (int.Parse(sdr3[1].ToString()) == 5)
    {
        btndn.ButtonText = sdr3[0].ToString() + " - " + "TR";
    }
    else if (int.Parse(sdr3[1].ToString()) == 6)
    {
}

```

```

btndn.ButtonText = sdr3[0].ToString() + " - " + "QUAD";
}

btndn.Size = new Size(192, 125);
btndn.Font = new Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.Bold);
btndn.ForeColor = System.Drawing.Color.White;
btndn.Click += new EventHandler(Btndn_MouseClick);
btndn.MouseEnter += new EventHandler(Btndn_MouseEnter);
btndn.MouseLeave += new EventHandler(Btndn_MouseLeave);
btndn.MouseDoubleClick += new MouseEventHandler(Btndn_DoubleClick);

if (sdr3[2].ToString() == "1") //dolu oda için
{
    btndn.IdleBorderColor = System.Drawing.Color.FromArgb(51, 122, 183);
    btndn.IdleFillColor = System.Drawing.Color.FromArgb(51, 122, 183);
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = System.Drawing.Color.FromArgb(40, 96, 144);
    btndn.onHoverState.FillColor = System.Drawing.Color.FromArgb(40, 96, 144);
}

else if (sdr3[2].ToString() == "2") //boş oda için
{
    btndn.IdleBorderColor = Color.MediumSeaGreen;
    btndn.IdleFillColor = Color.MediumSeaGreen;
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = Color.SeaGreen;
    btndn.onHoverState.FillColor = Color.SeaGreen;
}

else if (sdr3[2].ToString() == "3") //kirli oda için
{
    btndn.IdleBorderColor = System.Drawing.Color.FromArgb(215, 0, 0);
    btndn.IdleFillColor = System.Drawing.Color.FromArgb(215, 0, 0);
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = System.Drawing.Color.FromArgb(174, 0, 0);
    btndn.onHoverState.FillColor = System.Drawing.Color.FromArgb(174, 0, 0);
}

```

```

else if (sdr3[2].ToString() == "4") // arızalı oda için
{
    btndn.IdleBorderColor = Color.Gray;
    btndn.IdleFillColor = Color.Gray;
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = Color.DimGray;
    btndn.onHoverState.FillColor = Color.DimGray;
}

else
{
    //herhangi bir sorun olursa diye default renk
    btndn.IdleBorderColor = Color.Black;
    btndn.IdleFillColor = Color.Black;
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = Color.Black;
    btndn.onHoverState.FillColor = Color.Black;
}

flp1.Controls.Add(btndn);

}

con2.Close();

}

con.Close();

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

th1 = new Thread(new ThreadStart(Yenile));
th1.Start();

}//buton oluşturma,oda verisi çekme

```

```

public Thread th1;
private void Btndn_MouseEnter(object sender, EventArgs e)
{
    try
    {

        f.Show();
        f.Visible = false;
        f = Application.OpenForms["odaicerik"] as odaicerik;
        BunifuButton btndn = sender as BunifuButton;

        string metin = btndn.ButtonText;
        string[] parca = metin.Split('-');
        int splitodano = int.Parse(parca[0]);

        Point a = new Point();
        a = GetMousePositionWindowsForms();

        if (a.X < 1000)
            f.Location = new Point(a.X - 10, a.Y);
        else
            f.Location = new Point(a.X - 250, a.Y);

        f.Visible = true;
        SqlCommand cmd = new SqlCommand("Select
OdaNo,Ad,Soyad,GelisTarihi,GidisTarihi,OdaTip from Odaicerik where OdaNo='"
+ splitodano + "' and
OteliD='"
+ OteliId + "''", con);
        SqlCommand cmd2 = new SqlCommand("Select Count(Ad) as sayi from Odaicerik where
OdaNo='"
+ splitodano + "' and OteliID='"
+ OteliId + "''", con2);
        SqlCommand cmd3 = new SqlCommand("Select OdaNo,OdaTip,Durum from Odalar where
OdaNo='"
+ splitodano + "' and OteliID='"
+ OteliId + "''", con);
        con2.Close();
        con2.Open();
        SqlDataReader sdr2 = cmd2.ExecuteReader();

        while (sdr2.Read())
        {

```

```

if (Convert.ToInt32(sdr2[0].ToString()) > 0)
{
    con.Close();
    con.Open();
    SqlDataReader sdr = cmd.ExecuteReader();
    while (sdr.Read())
    {
        f.lblodano.Text = sdr["OdaNo"].ToString();
        f.lbladsoyad.Text = sdr["Ad"].ToString() + " " + sdr["Soyad"].ToString();
        int b = Convert.ToInt32(sdr["OdaTip"].ToString());
        if (b == 1)
        {
            f.lblodatip.Text = "Tek kişilik oda(SGL)";
        }
        else if (b == 2)
        {
            f.lblodatip.Text = "Standart iki yataklı oda(TFSU)";

        }
        else if (b == 3)
        {
            f.lblodatip.Text = "İki kişilik oda(TR) ";
        }
        else if (b == 4)
        {
            f.lblodatip.Text = "Standart çift kişilik oda(DBL)";
        }
        else if (b == 5)
        {
            f.lblodatip.Text = "Üç kişilik oda(TR)";
        }
        else if (b == 6)
        {
            f.lblodatip.Text = "Dört kişilik oda(Quad)";
        }
        f.lblgelistarihi.Text = sdr["GelisTarihi"].ToString();
        f.lblgidistarihi.Text = sdr["GidisTarihi"].ToString();
        f.lblodadurumu.Text = "Dolu";
    }
}

```

```

        con.Close();

    }
else
{
    con.Close();
    con.Open();
    SqlDataReader sdr3 = cmd3.ExecuteReader();
    while (sdr3.Read())
    {
        f.lblodano.Text = sdr3["OdaNo"].ToString();
        int i = int.Parse(sdr3["Durum"].ToString());

        if (i == 2)
        {
            f.lblodadurumu.Text = "Boş";
        }
        else if (i == 3)
        {
            f.lblodadurumu.Text = "Kirli";
        }
        else if (i == 4)
        {
            f.lblodadurumu.Text = "Arızalı";
        }

        int b = Convert.ToInt32(sdr3["OdaTip"].ToString());
        if (b == 1)
        {
            f.lblodatip.Text = "Tek kişilik oda(SGL)";
        }
        else if (b == 2)
        {
            f.lblodatip.Text = "Standart iki yataklı oda(TFSU)";
        }

        else if (b == 3)
        {
            f.lblodatip.Text = "İki kişilik oda(TR) ";
        }
    }
}

```

```

        }
        else if (b == 4)
        {
            f.lblodatip.Text = "Standart çift kişilik oda(DBL)";
        }
        else if (b == 5)
        {
            f.lblodatip.Text = "Üç kişilik oda(TR)";
        }
        else if (b == 6)
        {
            f.lblodatip.Text = "Dört kişilik oda(Quad)";
        }
        f.lbladsoyad.Text = "-----";
        f.lblgelistarihi.Text = "-----";
        f.lblgidistarihi.Text = "-----";
    }
    con.Close();
}

}

con2.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void Btndn_MouseClick(object sender, EventArgs e)
{
    try
    {
        BunifuButton btndn = sender as BunifuButton;
        string metin = btndn.ButtonText;

```

```

string[] parca = metin.Split('-');
int odano = int.Parse(parca[0]);

SqlCommand cmd = new SqlCommand("select DISTINCT(Durum) as sayi from Odalar", con);
con.Open();
SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())
{
    if (dr["sayi"].ToString() == "1" && btndn.BackColor ==
System.Drawing.Color.FromArgb(51, 122, 183)) //Dolu Oda için
    {
        SqlDataAdapter da = new SqlDataAdapter("select TcKimlikNo as
KimlikNo,Ad,Soyad,GelisTarihi as GelişTarihi from folyo where OdaNo=''' + odano + ''' and OtelID=''' +
OtelId + ''' ORDER BY GelisTarihi ASC ", con2);
        DataSet ds = new DataSet();
        da.Fill(ds);
        DataGridViewFolio.DataSource = ds.Tables[0];
    }
    else if (dr["sayi"].ToString() == "2" && btndn.BackColor == Color.MediumSeaGreen) // Boş
oda için
    {
        folioyenileme();
        //MessageBox.Show("Oda boş olduğu için Folio bilgileri getirilemiyor", "Uyarı",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else if (dr["sayi"].ToString() == "3" && btndn.BackColor ==
System.Drawing.Color.FromArgb(215, 0, 0)) //Kirli Oda için
    {
        folioyenileme();
        //MessageBox.Show("Oda boş ve kirli olduğu için Folio bilgileri getirilemiyor", "Uyarı",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else if (dr["sayi"].ToString() == "4" && btndn.BackColor == Color.Gray) // Arızalı Oda için
    {
        folioyenileme();
        // MessageBox.Show("Oda Arızalı olduğu için Folio bilgileri getirilemiyor", "Uyarı",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

```

        }

    }

    con.Close();

}//folio bilgileri
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

public void Yenile()
{
    try
    {
        while (true)
        {
            git:
            con3.Open();

            Thread.Sleep(500);
            SqlCommand oda = new SqlCommand(" select OdaNo,Durum from Odalar where OtelID='"
+ OtelId + "'", con3);
            SqlDataReader dr = oda.ExecuteReader();
            while (dr.Read())
            {
                con4.Open();
                SqlCommand karsilastir = new SqlCommand("select Durum from OdalarYenileme where
OdaNo='"
+ dr[0].ToString() + "' and OtelID='"
+ OtelId + "'", con4);
                SqlDataReader dr2 = karsilastir.ExecuteReader();
                while (dr2.Read())
                {
                    if (dr[1].ToString() == dr2[0].ToString())

```

```

{
}

else
{

    if (this.InvokeRequired)
    {
        this.Invoke((MethodInvoker)delegate ()
        {

            tabControl1.TabPages.Clear();
        });
    }

    con.Open();
    SqlCommand cmd = new SqlCommand("select DISTINCT(Kat) from Odalar where
OtelId=" + OtelId + "", con); // kat bilgisini buradan aliyoruz

    SqlDataReader calislütfen = cmd.ExecuteReader();
    while (calislütfen.Read())
    {
        TabPage tabPage1 = new TabPage();
        tabPage1.Name = "tabPage" + calislütfen[0].ToString(); //tabpagelere kat ismini
        atıyarak
        tabPage1.Text = "KAT " + calislütfen[0].ToString();
        tabPage1.Font = new Font("Verdana", 12);

        if (this.InvokeRequired)
        {
            this.Invoke((MethodInvoker)delegate ()
            {

                tabControl1.TabPages.Add(tabPage1);
            });
        }
    }
}

```

```

FlowLayoutPanel flp1 = new FlowLayoutPanel();
if (this.InvokeRequired)
{
    this.Invoke((MethodInvoker)delegate ()
    {
        Controls.Add(flp1);
        tabPage1.Controls.Add(flp1);

    });
}
flp1.BackColor = Color.WhiteSmoke;

flp1.Dock = DockStyle.Fill;
flp1.AutoScroll = true;

SqlCommand cmd3 = new SqlCommand("select OdaNo,OdaTip,Durum from
Odalar where OtelID=' + OtelId + '' and Kat=' + calislütfen[0].ToString() + '' ORDER BY OdaNo
ASC", con7);
//kata göre odaları ekleyen sorgumuz
//con2.Close();
con7.Open();
SqlDataReader sdr3 = cmd3.ExecuteReader();
int i = 1;
while (sdr3.Read())
{
    BunifuButton btndn = new BunifuButton();

    btndn.Margin = new System.Windows.Forms.Padding(5);
    btndn.IdleBorderThickness = 1;
    if (i == 1)
    { Thread.Sleep(100); i++; }
    if (int.Parse(sdr3[1].ToString()) == 1)
    {
        btndn.ButtonText = sdr3[0].ToString() + " - " + "SGL";
    }
}

```

```

else if (int.Parse(sdr3[1].ToString()) == 2)
{
    btndn.ButtonText = sdr3[0].ToString() + " - " + "TSU";
}
else if (int.Parse(sdr3[1].ToString()) == 3)
{
    btndn.ButtonText = sdr3[0].ToString() + " - " + "TWR";
}
else if (int.Parse(sdr3[1].ToString()) == 4)
{
    btndn.ButtonText = sdr3[0].ToString() + " - " + "DBL";
}
else if (int.Parse(sdr3[1].ToString()) == 5)
{
    btndn.ButtonText = sdr3[0].ToString() + " - " + "TR";
}
else if (int.Parse(sdr3[1].ToString()) == 6)
{
    btndn.ButtonText = sdr3[0].ToString() + " - " + "QUAD";
}

btndn.Size = new Size(192, 125);
btndn.Font = new Font("Microsoft Sans Serif", 18F,
System.Drawing.FontStyle.Bold);

btndn.ForeColor = System.Drawing.Color.White;
btndn.Click += new EventHandler(Btndn_MouseClick);
btndn.MouseEnter += new EventHandler(Btndn_MouseEnter);
btndn.MouseLeave += new EventHandler(Btndn_MouseLeave);
btndn.MouseDoubleClick += new MouseEventHandler(Btndn_DoubleClick);

if (sdr3[2].ToString() == "1") //dolu oda için
{
    btndn.IdleBorderColor = System.Drawing.Color.FromArgb(51, 122, 183);
    btndn.IdleFillColor = System.Drawing.Color.FromArgb(51, 122, 183);
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = System.Drawing.Color.FromArgb(40, 96,
144);
    btndn.onHoverState.FillColor = System.Drawing.Color.FromArgb(40, 96,
144);
}

```

```

    }

else if (sdr3[2].ToString() == "2") //boş oda için
{
    btndn.IdleBorderColor = Color.MediumSeaGreen;
    btndn.IdleFillColor = Color.MediumSeaGreen;
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = Color.SeaGreen;
    btndn.onHoverState.FillColor = Color.SeaGreen;
}

else if (sdr3[2].ToString() == "3") //kirli oda için
{
    btndn.IdleBorderColor = System.Drawing.Color.FromArgb(215, 0, 0);
    btndn.IdleFillColor = System.Drawing.Color.FromArgb(215, 0, 0);
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = System.Drawing.Color.FromArgb(174, 0,
0);
    btndn.onHoverState.FillColor = System.Drawing.Color.FromArgb(174, 0, 0);
}

else if (sdr3[2].ToString() == "4") // arızalı oda için
{
    btndn.IdleBorderColor = Color.Gray;
    btndn.IdleFillColor = Color.Gray;
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = Color.DimGray;
    btndn.onHoverState.FillColor = Color.DimGray;
}

else
{
    //herhangi bir sorun olursa diye default renk
    btndn.IdleBorderColor = Color.Black;
    btndn.IdleFillColor = Color.Black;
    btndn.BackColor = btndn.IdleFillColor;
    btndn.onHoverState.BorderColor = Color.Black;
    btndn.onHoverState.FillColor = Color.Black;
}

```

```

        }

        if (this.InvokeRequired)
        {
            this.Invoke((MethodInvoker)delegate ()
            {
                flp1.Controls.Add(btndn);
            });
        }

    }

    con7.Close();

}

con.Close();

con5.Open();
SqlCommand odalis = new SqlCommand(" select OdaNo,Durum from Odalar where
OtelID=''' + OtelId + '''", con5);
SqlDataReader dr3 = odalis.ExecuteReader();
while (dr3.Read())
{
    con6.Open();
    SqlCommand guncelle = new SqlCommand("update OdalarYenileme set
Durum=@durum where OdaNo=''' + dr3[0].ToString() + ''' and OtelID=''' + OtelId + '''", con6);
    guncelle.Parameters.AddWithValue("@durum", dr3[1].ToString());
    guncelle.ExecuteNonQuery();
    con6.Close();
}

con5.Close();
con3.Close();
con4.Close();

```

```

        goto git;
    }
}
con4.Close();
}

con3.Close();

}

}

catch (Exception ex)
{
    //MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void Resepsiyon_FormClosed(object sender, FormClosedEventArgs e)
{
    try
    {
        th1.Abort();
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

}

private void DataGridViewHatirlatmalar_DoubleClick(object sender, EventArgs e)

```

```

{
    Hatirlatmaekle hatirlatmaekle = new Hatirlatmaekle();
    hatirlatmaekle.Show();
}

private void DataGridViewHatirlatmalar_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {

        Hatirlatmaekle hatirlatmaekle = new Hatirlatmaekle();
        hatirlatmaekle.Show();
        hatirlatmaekle.panel1.Visible = true;
        con.Open();

        SqlCommand cmd = new SqlCommand("Select Hatirlatmametin From Hatirlatmalar Where
        Hatirlatmano=@hatirlatmano And OtelID='' + OtelId + "", con);
        cmd.Parameters.AddWithValue("@hatirlatmano",
        DataGridViewHatirlatmalar.CurrentRow.Cells[2].Value.ToString());

        SqlDataReader dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            hatirlatmaekle.richTextBox2.Text = dr[0].ToString();
        }

        con.Close();
    }
    catch (Exception ex)
    {
}

```

```
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

}
```

## ODAİCİRİK.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{
    public partial class odaicerik : Form
    {
        public odaicerik()
        {
            InitializeComponent();
        }

        private void odaicerik_Load(object sender, EventArgs e)
        {
        }
    }
}
```

## R\_KONAK.CS

```
using System;
using System.Collections.Generic;
```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using Bunifu.UI.WinForms.BunifuButton;
namespace Login
{
    public partial class r_konak : Form
    {
        public r_konak()
        {
            InitializeComponent();
        }

        SqlConnection con = new SqlConnection(Login.veritabani);
        SqlConnection con2 = new SqlConnection(Login.veritabani);
        SqlConnection con3 = new SqlConnection(Login.veritabani);
        SqlConnection con4 = new SqlConnection(Login.veritabani);
        int OtelID = Login.id;

        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                datetpGelis.MinDate = DateTime.Now.Date;
                datetpGelis.MaxDate = DateTime.Now.Date;
                btn_cik.IconVisible = true; btn_ekle.IconVisible = true; btn_ekleEx.IconVisible = true;
                btn_fatura.IconVisible = true; btn_iptal.IconVisible = true; btn_kay.IconVisible = true;
                btn_kayder.IconVisible = true; btn_yazdir.IconVisible = true;
                SqlCommand cmd = new SqlCommand("Select OdaNo from Odalar where OtelID=" + OtelID +
                "'''", con);
                con.Open();
                SqlDataReader sdr = cmd.ExecuteReader();

```

```

        while (sdr.Read())
    {
        cmbodano.Items.Add(sdr[0]).ToString();

    }
    con.Close();

cmbfiyatkod.Items.Add("Toplam Fiyat");
cmbfiyatkod.Items.Add("Kişi Başı Fiyat");
cmbfiyatkod.SelectedIndex = 0;

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

DatetpGidis.MinDate = datetpGelis.Value.AddDays(+1);

}

private void tabPage1_Click(object sender, EventArgs e)
{

}

private void cmbodano_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        toplamodafiyat.Text = "";
        geneltxt.Text = "";
        odenecektxt.Text = "";
        r_konakdatagrid.Rows.Clear();
        SqlCommand cmd = new SqlCommand("Select
GelisTarihi,GidisTarihi,KaldigiGeceSayisi,OdaTip,Televizyon,Klima,Buzdolabi,MiniBar,Telefon,Notlar
from R_Konaklama where OdaNo=''' + cmbodano.Text + ''' and OtelID=''' + OtelID + '''", con);
    }
}

```

```

SqlCommand cmd2 = new SqlCommand("Select Durum from Odalar Where OdaNo=" +
cmbodano.Text + " and OtelID=" + OtelID + "", con2);
SqlCommand cmd3 = new SqlCommand("Select
OdaTip,Televizyon,Klima,Buzdolabi,Minibar,Telefon from Odalar Where OdaNo=" + cmbodano.Text + ""
and OtelID=" + OtelID + "", con);
SqlCommand cmd4 = new SqlCommand("Select
TcKimlikNo,Ad,Soyad,Cinsiyeti,Uyruk,GelisTarihi,GidisTarihi from r_konakgriddoldurma where
OdaNo=" + cmbodano.Text + " and OtelID=" + OtelID + "", con); ;

con.Open();
SqlDataReader readgrid = cmd4.ExecuteReader();
int i = 0;
while (readgrid.Read())
{
    r_konakdatagrid.Rows.Add();
    r_konakdatagrid.Rows[i].Cells[0].Value = readgrid["TcKimlikNo"].ToString();
    r_konakdatagrid.Rows[i].Cells[1].Value = readgrid["Ad"].ToString();
    r_konakdatagrid.Rows[i].Cells[2].Value = readgrid["Soyad"].ToString();

    if (Boolean.Parse(readgrid["Cinsiyeti"].ToString()) == true)
    { r_konakdatagrid.Rows[i].Cells[3].Value = "Erkek"; }
    else if (Boolean.Parse(readgrid["Cinsiyeti"].ToString()) == false)
    { r_konakdatagrid.Rows[i].Cells[3].Value = "Kadın"; }

    if (Boolean.Parse(readgrid["Uyruk"].ToString()) == true)
    { r_konakdatagrid.Rows[i].Cells[4].Value = "Türk"; }
    else if (Boolean.Parse(readgrid["Uyruk"].ToString()) == false)
    { r_konakdatagrid.Rows[i].Cells[4].Value = "Yabancı"; }

    r_konakdatagrid.Rows[i].Cells[5].Value = readgrid["GelisTarihi"].ToString();
    r_konakdatagrid.Rows[i].Cells[5].Value =
Convert.ToDateTime(r_konakdatagrid.Rows[i].Cells[5].Value).ToString("yyyy-MM-dd");
    r_konakdatagrid.Rows[i].Cells[6].Value = readgrid["GidisTarihi"].ToString();
    r_konakdatagrid.Rows[i].Cells[6].Value =
Convert.ToDateTime(r_konakdatagrid.Rows[i].Cells[6].Value).ToString("yyyy-MM-dd");
    i += 1;
}
con.Close();

```

```

try
{
    con2.Open();
    SqlDataReader sdr2 = cmd2.ExecuteReader();
    sdr2.Read();
    if (sdr2[0].ToString() == "1")
    {
        con.Open();
        SqlDataReader sdr = cmd.ExecuteReader();
        while (sdr.Read())
        {
            //datetpGelis.Value = Convert.ToDateTime(sdr["GelisTarihi"].ToString());
            //DatetpGidis.Value = Convert.ToDateTime(sdr["GidisTarihi"].ToString());
            datetpGelis.MinDate = DateTime.Now.Date;
            datetpGelis.MaxDate = DateTime.Now.Date;
            txtGecesayisi.Text = sdr["KaldigiGeceSayisi"].ToString();
            int b = Convert.ToInt32(sdr["OdaTip"].ToString());
            if (b == 1)
            {
                txtodatip.Text = ("Tek kişilik oda(SGL)");
            }
            else if (b == 2)
            {
                txtodatip.Text = ("Standart iki yataklı oda(TFSU)");
            }
            else if (b == 3)
            {
                txtodatip.Text = ("İki kişilik oda(TR) ");
            }
            else if (b == 4)
            {
                txtodatip.Text = ("Standart çift kişilik oda(DBL)");
            }
            else if (b == 5)
            {
                txtodatip.Text = ("Üç kişilik oda(TR)");
            }
        }
    }
}

```

```

else if (b == 6)
{
    txtodatip.Text = ("Dört kişilik oda(Quad)");
}
if (Convert.ToBoolean(sdr["Televizyon"].ToString()) == true)
{
    chcktelevizyon.Checked = true;
}
else if (Convert.ToBoolean(sdr["Televizyon"].ToString()) == false)
{
    chcktelevizyon.Checked = false;
}
if (Convert.ToBoolean(sdr["Klima"].ToString()) == true)
{
    chckklima.Checked = true;
}
else
{
    chckklima.Checked = false;
}
if (Convert.ToBoolean(sdr["Buzdolabi"].ToString()) == true)
{
    chckBuzdolabi.Checked = true;
}
else
{
    chckBuzdolabi.Checked = false;
}
if (Convert.ToBoolean(sdr["MiniBar"].ToString()) == true)
{
    chckMiniBar.Checked = true;
}
else
{
    chckMiniBar.Checked = false;
}
if (Convert.ToBoolean(sdr["Telefon"].ToString()) == true)
{
    ChckTelefon.Checked = true;
}

```

```

        }

        else
        {
            ChckTelefon.Checked = false;
        }

        richTextBox2.Text = sdr["Notlar"].ToString();

    }

    con.Close();
}

else
{
    con.Open();
    SqlDataReader sdr3 = cmd3.ExecuteReader();
    while (sdr3.Read())
    {
        int b = Convert.ToInt32(sdr3["OdaTip"].ToString());
        if (b == 1)
        {
            txtodatip.Text = ("Tek kişilik oda(SGL)");
        }
        else if (b == 2)
        {
            txtodatip.Text = ("Standart iki yataklı oda(TFSU)");
        }

        else if (b == 3)
        {
            txtodatip.Text = ("İki kişilik oda(TR) ");
        }
        else if (b == 4)
        {
            txtodatip.Text = ("Standart çift kişilik oda(DBL)");
        }
        else if (b == 5)
        {
    }
}

```

```

        txtodatip.Text = ("Üç kişilik oda(TR)");
    }
    else if (b == 6)
    {
        txtodatip.Text = ("Dört kişilik oda(Quad)");
    }
    if (Convert.ToBoolean(sdr3["Televizyon"].ToString()) == true)
    {
        chcktelevizyon.Checked = true;
    }
    else if (Convert.ToBoolean(sdr3["Televizyon"].ToString()) == false)
    {
        chcktelevizyon.Checked = false;
    }
    if (Convert.ToBoolean(sdr3["Klima"].ToString()) == true)
    {
        chckklima.Checked = true;
    }
    else
    {
        chckklima.Checked = false;
    }
    if (Convert.ToBoolean(sdr3["Buzdolabi"].ToString()) == true)
    {
        chckBuzdolabi.Checked = true;
    }
    else
    {
        chckBuzdolabi.Checked = false;
    }
    if (Convert.ToBoolean(sdr3["MiniBar"].ToString()) == true)
    {
        chckMiniBar.Checked = true;
    }
    else
    {
        chckMiniBar.Checked = false;
    }
    if (Convert.ToBoolean(sdr3["Telefon"].ToString()) == true)

```

```

        {
            ChckTelefon.Checked = true;
        }
        else
        {
            ChckTelefon.Checked = false;
        }

        richTextBox2.Text = "";
    }
    con.Close();
}

con2.Close();

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void btn_ekle_Click(object sender, EventArgs e)
{
    try
    {
        int a = 0;
        SqlCommand cmd3 = new SqlCommand("Select OdaTip from Odalar Where OdaNo=" + 
cmbodano.Text + " and OtelID=" + OtelID + "", con);
        con.Open();
    }
}

```

```

SqlDataReader sdr3 = cmd3.ExecuteReader();
while (sdr3.Read())
{
    int b = Convert.ToInt32(sdr3["OdaTip"].ToString());
    if (b == 1 && (r_konakdatagrid.RowCount) <= 1 && a==0)
    {
        a++;
        rez_tikla rez_Tikla = new rez_tikla();
        rez_Tikla.Show();
    }
    else if (b == 2 && (r_konakdatagrid.RowCount) <= 2 && a == 0)
    {
        a++;
        rez_tikla rez_Tikla = new rez_tikla();
        rez_Tikla.Show();
    }
    else if (b == 3 && (r_konakdatagrid.RowCount) <= 2 && a == 0)
    {
        a++;
        rez_tikla rez_Tikla = new rez_tikla();
        rez_Tikla.Show();
    }
    else if (b == 4 && (r_konakdatagrid.RowCount) <= 2 && a == 0)
    {
        a++;
        rez_tikla rez_Tikla = new rez_tikla();
        rez_Tikla.Show();
    }
    else if (b == 5 && (r_konakdatagrid.RowCount) <= 3 && a == 0)
    {
        a++;
        rez_tikla rez_Tikla = new rez_tikla();
        rez_Tikla.Show();
    }
    else if (b == 6 && (r_konakdatagrid.RowCount) <= 4 && a == 0)
    {
        a++;
        rez_tikla rez_Tikla = new rez_tikla();
        rez_Tikla.Show();
    }
}

```

```

        }

        else
            MessageBox.Show("Mevcut oda tipinde maksimum kişi sayısına ulaşıldı.", "Uyarı",
    MessageBoxButtons.OK, MessageBoxIcon.Stop);

    }

    con.Close();
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void btn_cik_Click(object sender, EventArgs e)
{
    try
    {

        SqlCommand cmd = new SqlCommand("Select OdenecekTutar From K_Cikis Where OdaNo='"
+ cmbodano.Text + "' and OtelID=''" + OtelID + "'", con);

        SqlCommand cmd2 = new SqlCommand("Select TcKimlik from K_Giris Where OdaNo='"
+ cmbodano.Text + "' and OtelID=''" + OtelID + "'", con3);

        con.Open();
        con3.Open();

        SqlDataReader sdr = cmd.ExecuteReader();

        while (sdr.Read())
        {
            if (Convert.ToDouble(sdr[0].ToString()) == 0)
            {
                con2.Open();

                SqlCommand update = new SqlCommand("update Odalar set Durum=3 where OdaNo='"
+ cmbodano.Text + "' and OtelID=''" + OtelID + "'", con2);

                update.ExecuteNonQuery();

                SqlCommand deletesiparis = new SqlCommand("delete from Siparisler where OdaNo='"
+ cmbodano.Text + "' and OtelID=''" + OtelID + "'", con2);

                deletesiparis.ExecuteNonQuery();
            }
        }
    }
}

```

```

SqlCommand deleteodemeler = new SqlCommand("delete from Odemeler Where OdaNo='"
+ cmbodano.Text + "' and OtelID=''" + OtelID + "", con2);
    deleteodemeler.ExecuteNonQuery();

SqlDataReader sdr2 = cmd2.ExecuteReader();
while (sdr2.Read())
{
    SqlCommand updatetc = new SqlCommand("Update r_Musteriekle set Durum=0 where
TcKimlikNo=''" + sdr2[0].ToString() + "' and OtelID=''" + OtelID + "", con2);
        updatetc.ExecuteNonQuery();
    SqlCommand kgrissil = new SqlCommand("delete from K_Giris Where TcKimlik='"
+sdr2[0].ToString() + "' and OtelID=''" + OtelID + "", con2);
        kgrissil.ExecuteNonQuery();
    SqlCommand kcikissil = new SqlCommand("delete from K_Cikis where OdaNo='"
+cmbodano.Text + "' and OtelID=''" + OtelID + "", con2);
        kcikissil.ExecuteNonQuery();

}
con2.Close();
MessageBox.Show("İstediğiniz başarı ile çıkış yapılmıştır.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    r_konakdatagrid.Rows.Clear();
    datagridodemeler.Rows.Clear();

}
else
{
    MessageBox.Show("Oda ödemesi tamamlanmadan çıkış yapılamaz");
}

}
con.Close();
con2.Close();
con3.Close();
}

```

```

        catch(Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

}

private void btn_kay_Click(object sender, EventArgs e)
{
    try
    {
        int kaldiigece = 0;
        if (r_konakdatagrid.RowCount > 0)
        {
            for (int j = 0; j < r_konakdatagrid.RowCount - 1; j++)
            {
                con.Open();
                con3.Open();
                SqlCommand sorgula = new SqlCommand("Select Count(TcKimlik) from K_Giris where
TcKimlik=@tc and OtelID=""" + OtelID + """", con3);
                sorgula.Parameters.AddWithValue("@tc",
r_konakdatagrid.Rows[j].Cells[0].Value.ToString());

                SqlDataReader dr = sorgula.ExecuteReader();
                while (dr.Read())
                {
                    if (Convert.ToInt32(dr[0].ToString()) == 0)
                    {

                        SqlCommand cmd = new SqlCommand("insert into
K_Giris(OtelID,TcKimlik,GelisTarihi,GidisTarihi,KaldiGeceSayisi,OdaNo) Values(@otelid
,@tc,@gelis,@gidis,@gece,@odano)", con);
                        cmd.Parameters.AddWithValue("@otelid", OtelID);
                        string a = r_konakdatagrid.Rows[j].Cells[0].Value.ToString();
                        cmd.Parameters.AddWithValue("@tc", a);
                        cmd.Parameters.AddWithValue("@gelis",
r_konakdatagrid.Rows[j].Cells[5].Value.ToString());
                    }
                }
            }
        }
    }
}

```

```

        cmd.Parameters.AddWithValue("@gidis",
r_konakdatagrid.Rows[j].Cells[6].Value.ToString());
        cmd.Parameters.AddWithValue("@gece", txtGecesayisi.Text);
        cmd.Parameters.AddWithValue("@odano", cmbodano.Text);
        cmd.ExecuteNonQuery();
        SqlCommand update = new SqlCommand("update Odalar set Durum=1 where
OdaNo=@odano", con4);
        update.Parameters.AddWithValue("@odano", cmbodano.Text);
        con4.Open();
        update.ExecuteNonQuery();
        con4.Close();

    }

    else
    {
    }
}

con.Close();
con3.Close();
}

for (int j = 0; j < r_konakdatagrid.RowCount - 1; j++)
{
    DateTime gelis =
Convert.ToDateTime(Convert.ToDateTime(r_konakdatagrid.Rows[j].Cells[5].Value).ToString("yyyy-
MM-dd"));
    DateTime gidis =
Convert.ToDateTime(Convert.ToDateTime(r_konakdatagrid.Rows[j].Cells[6].Value).ToString("yyyy-
MM-dd"));
    TimeSpan sonuc = gidis - gelis;
    kaldiigece += int.Parse(sonuc.TotalDays.ToString());
}

SqlCommand bak = new SqlCommand("Select COUNT(OdaNo) from K_Cikis where
OdaNo=@odano and OtelID=@otelid", con3);
SqlCommand odalar = new SqlCommand("Select Fiyat from Odalar where OdaNo=@odano and
OtelID=@otelid", con);
odalar.Parameters.AddWithValue("@odano", cmbodano.Text);

```

```

odalar.Parameters.AddWithValue("@otelid", OtelID);
con.Open();
SqlDataReader odalaroku = odalar.ExecuteReader();
double odafiyat = 0;
if (odalaroku.Read())
{
    odafiyat += Convert.ToDouble(odalaroku[0].ToString());
}
con.Close();
bak.Parameters.AddWithValue("@odano", cmbodano.Text);
bak.Parameters.AddWithValue("@otelid", OtelID);
con3.Open();

SqlDataReader dene = bak.ExecuteReader();
while (dene.Read())
{
    if (Convert.ToInt32(dene[0].ToString()) == 0)
    {
        con2.Open();
        SqlCommand insert = new SqlCommand("insert into
K_Cikis(OtelID,OdaNo,KalinanGeceSayisi,ToplamOdaFiyati,EkstraHarcamalar,ToplamOdemeler,GenelToplam,Indirim,OdenecekTutar)
Values(@otelid,@odano,@gece,@toplalomodafiyat,@extra,@toplalomodeme,@geneltoplam,@indirim,@odeneciktutar)", con2);
        insert.Parameters.AddWithValue("@otelid", OtelID);
        insert.Parameters.AddWithValue("@odano", cmbodano.Text);
        insert.Parameters.AddWithValue("@gece", kaldigigece);
        double toplamoda = odafiyat * kaldigigece;
        insert.Parameters.AddWithValue("@toplalomodafiyat", toplamoda);
        insert.Parameters.AddWithValue("@extra", double.Parse(extraharcamalartxt.Text));
        insert.Parameters.AddWithValue("@toplalomodeme", 0);
        insert.Parameters.AddWithValue("@geneltoplam", ((Convert.ToDouble(odafiyat) *
Convert.ToDouble(kaldigigece))) + Convert.ToDouble(extraharcamalartxt.Text));
        insert.Parameters.AddWithValue("@indirim", 0);
        insert.Parameters.AddWithValue("@odeneciktutar", (((Convert.ToDouble(odafiyat) *
Convert.ToDouble(kaldigigece))) + Convert.ToDouble(extraharcamalartxt.Text)));
        insert.ExecuteNonQuery();
        con2.Close();
    }
}

```

```

        con.Close();
        yenile();
        con.Open();

        MessageBox.Show("Girdiğiniz Veriler Başarı İle Eklendi", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    else
    {
        con.Close();
        yenile();
        con.Open();
        con2.Open();

        SqlCommand update = new SqlCommand("update K_Cikis set
KalinanGeceSayisi=@gece,ToplamOdaFiyati=@toplamodafiyat,EkstraHarcamalar=@extra,ToplamOdeme
ler=@toplalomdeme,GenelToplam=@geneltoplam,OdencekTutar=@odeneciktutar where OdaNo='"
+cmbodano.Text + "" and OtelID='"
+ OtelID + "", con2);

        update.Parameters.AddWithValue("@gece", kaldigigece);
        update.Parameters.AddWithValue("@toplamodafiyat", odafiyat * kaldigigece);
        update.Parameters.AddWithValue("@extra", double.Parse(extraharcamalartxt.Text));
        update.Parameters.AddWithValue("@toplalomdeme", double.Parse(toplamodemeler.Text));
        update.Parameters.AddWithValue("@geneltoplam", (Convert.ToDouble(odafiyat) *
Convert.ToDouble(kaldigigece)) + Convert.ToDouble(extraharcamalartxt.Text));

        update.Parameters.AddWithValue("@odeneciktutar", ((Convert.ToDouble(odafiyat) *
Convert.ToDouble(kaldigigece)) + Convert.ToDouble(extraharcamalartxt.Text) -
double.Parse(toplamodemeler.Text)));

        update.ExecuteNonQuery();

        MessageBox.Show("Girdiğiniz Veriler Başarı İle Güncellendi", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        con2.Close();
    }

    con.Close();
    con3.Close();

```

```

        }
    else
    {
        MessageBox.Show("Kaydetmeden Önce Bazı Kişiler Eklemelisiniz", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        con.Close();
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

private void datntpGelis\_ValueChanged(object sender, EventArgs e) //kalınan gece sayısı hesaplandı

```

{
    DatntpGidis.MinDate = datntpGelis.Value;
    DateTime gelis = Convert.ToDateTime(datntpGelis.Value.ToString("yyyy-MM-dd"));
    DateTime gidis = Convert.ToDateTime(DatntpGidis.Value.ToString("yyyy-MM-dd"));
    TimeSpan sonuc = gidis - gelis;
    txtGecesayisi.Text = sonuc.TotalDays.ToString(); ;
}

```

private void btn\_ip\_Click(object sender, EventArgs e)

```

{
    this.Hide();
    r_konak konak = new r_konak();
    konak.Dispose();
}

```

private void tabControl1\_SelectedIndexChanged(object sender, EventArgs e)//harcamaları aldığımız datagridleri dolduruyoruz

```

{
    yenile();
    con.Open();
    DataGridViewHarcama.Rows.Clear();
    int i = 0;
}

```

```

SqlCommand cmd = new SqlCommand("select TcKimlik,KaldigiGeceSayisi from K_Giris where
OdaNo=''' + cmbodano.Text + ''' and OtelID=''' + OtelID + '''", con); //odadaki tüm kişilerin tc noları ve
kadikleri gece sayısını alıyoruz

SqlDataReader sdr = cmd.ExecuteReader();
while (sdr.Read()) // aldığımız herkesi sırayla döndürüyoruz
{
    DataGridViewHarcama.Rows.Add();
    DataGridViewHarcama.Rows[i].Cells[0].Value = sdr[1].ToString();
    con2.Open();
    SqlCommand cmd2 = new SqlCommand("select Ad,Soyad from R_Musteriekle where
TcKimlikNo=''' + sdr[0] + ''' and OtelID=''' + OtelID + '''", con2); //sırasıyla gelen kişinin tcsile
sorgulayıp ad ve soyadını birleştirip yazıyoruz

    SqlDataReader sdr2 = cmd2.ExecuteReader();
    while (sdr2.Read())
    {
        DataGridViewHarcama.Rows[i].Cells[1].Value = sdr2[0].ToString() + " " +
sdr2[1].ToString();
    }
    con2.Close();
    con3.Open();
    SqlCommand cmd3 = new SqlCommand("select Fiyat from Odalar where OdaNo=''' +
cmbodano.Text + ''' and OtelID=''' + OtelID + '''", con3); //bulunan odanın fiyatını alıyoruz

    SqlDataReader sdr3 = cmd3.ExecuteReader();
    while (sdr3.Read())
    {
        DataGridViewHarcama.Rows[i].Cells[2].Value = (double.Parse(sdr3[0].ToString()) *
double.Parse(sdr[1].ToString())).ToString(); //kişinin kaldığı gün sayısı ile oda fiyatının çarpıp tutar elde
ediyoruz
    }
    con3.Close();
    i++;
}
con.Close();
bunifuCustomDataGridView1.Rows.Clear();

SqlCommand siparis = new SqlCommand("Select Siparis from Siparisler Where OdaNo=@odano
and OtelID=@otelid and SiparisDurum=@siparis", con4);
siparis.Parameters.AddWithValue("@odano", cmbodano.Text);
siparis.Parameters.AddWithValue("@otelid", OtelID);
siparis.Parameters.AddWithValue("@siparis", false);

```

```

con4.Open();
int b = 0;
SqlDataReader dr = siparis.ExecuteReader();
while (dr.Read())
{
    string metin = dr["Siparis"].ToString();
    string[] parca = metin.Split('/');
    for (int a = 0; a < parca.Length; a++)
    {
        string[] metinadet = parca[a].Split('-');
        string adet = metinadet[0];
        string[] urun = metinadet[1].Split('=');
        string urunad = urun[0];
        string tutar = urun[1];

        bunifuCustomDataGridView1.Rows.Add();
        bunifuCustomDataGridView1.Rows[b].Cells[0].Value = adet;
        bunifuCustomDataGridView1.Rows[b].Cells[1].Value = urunad;
        bunifuCustomDataGridView1.Rows[b].Cells[2].Value = tutar;
        b++;
    }
}
con4.Close();

}

private void DatetpGidis_ValueChanged(object sender, EventArgs e)
{
    DateTime gelis = Convert.ToDateTime(datetpGelis.Value.ToString("yyyy-MM-dd"));
    DateTime gidis = Convert.ToDateTime(DatetpGidis.Value.ToString("yyyy-MM-dd"));
    TimeSpan sonuc = gidis - gelis;
    txtGecesayisi.Text = int.Parse(sonuc.TotalDays.ToString()).ToString();
}

```

```

private void tahsiltxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
    if (e.KeyChar == (char)Keys.Enter)
    {
        if (Convert.ToInt32(tahsiltxt.Text) <= Convert.ToInt32(odenecektxt.Text))
        {
            toplamodemeler.Text = (Convert.ToDouble(toplamodemeler.Text) +
Convert.ToDouble(tahsiltxt.Text)).ToString();
            int i = datagridodemeler.RowCount;
            datagridodemeler.Rows.Add();
            datagridodemeler.Rows[i].Cells[0].Value = cmbodano.Text;
            datagridodemeler.Rows[i].Cells[1].Value = DateTime.Now;
            datagridodemeler.Rows[i].Cells[2].Value = tahsiltxt.Text;
            datagridodemeler.Rows[i].Cells[3].Value = "ekle";
            tahsiltxt.Text = null;
            odenecektxt.Text = (Convert.ToDouble(geneltxt.Text) -
Convert.ToDouble(toplamodemeler.Text)).ToString();
        }
        else
        {
            MessageBox.Show("Tahsil Edilen Tutar Ödenecek Tutardan Fazla Olamaz", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}

private void extraharcamalartxt_TextChanged(object sender, EventArgs e)
{
    geneltxt.Text = (Convert.ToDouble(toplamodafiyat.Text) +
Convert.ToDouble(extraharcamalartxt.Text)).ToString();
}

private void toplamodemeler_TextChanged(object sender, EventArgs e)
{
}

```

```

}

private void textBox11_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
    if (e.KeyChar == (char)Keys.Enter)
    {

        if (Convert.ToDouble(indirimtxt.Text) <= Convert.ToDouble(odenecektxt.Text))
        {
            odenecektxt.Text = Convert.ToDouble((Convert.ToDouble(odenecektxt.Text) -
Convert.ToDouble(indirimtxt.Text))).ToString();
        }
        else
        {
            MessageBox.Show("İndirim Ödenecek Tutarın Fazla Olamaz", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}

private void yenile()
{
    SqlCommand odenecktutar = new SqlCommand("select
ToplamOdaFiyati,EkstraHarcamalar,ToplamOdemeler,GenelToplam,Indirim,OdenecekTutar from K_Cikis
where OdaNo=" + cmbodano.Text + " and OtelID=" + OtelID + "", con);
    SqlCommand odemeler = new SqlCommand("Select OdaNo,Odeme,OdemeTarih,IslemNo from
Odemeler where OdaNo=@odano and OtelID=@otelid", con2);
    odemeler.Parameters.AddWithValue("@odano", cmbodano.Text);
    odemeler.Parameters.AddWithValue("@otelid", OtelID);

    con.Open();
    SqlDataReader oku = odenecktutar.ExecuteReader();
}

```

```

while (oku.Read())
{
    toplamodafiyat.Text = (Convert.ToDouble(oku[0].ToString())).ToString();
    extraharcamalartxt.Text = (Convert.ToDouble(oku[1].ToString())).ToString();
    toplamodemeler.Text = (Convert.ToDouble(oku[2].ToString())).ToString();
    geneltxt.Text = (Convert.ToDouble(oku[3].ToString())).ToString();
    indirimtxt.Text = (Convert.ToDouble(oku[4].ToString())).ToString();
    odenecektxt.Text = (Convert.ToDouble(oku[5].ToString())).ToString();

}

con.Close();
con2.Open();
SqlDataReader dr = odemeler.ExecuteReader();
int i = 0;
datagridodemeler.Rows.Clear();
while (dr.Read())
{
    datagridodemeler.Rows.Add();
    datagridodemeler.Rows[i].Cells[0].Value = dr["OdaNo"].ToString();
    datagridodemeler.Rows[i].Cells[1].Value = dr["OdemeTarih"].ToString();
    datagridodemeler.Rows[i].Cells[2].Value = Convert.ToDouble(dr["Odeme"].ToString()) + "₺";
    datagridodemeler.Rows[i].Cells[3].Value = dr["IslemNo"].ToString();
    i++;
}

}
con2.Close();
con.Close();
}

```

```

private void btn_kayder_Click(object sender, EventArgs e)
{
    try
    {
        con.Open();
        for (int i = 0; i < datagridodemeler.RowCount; i++)

```

```

{
    if (datagridodemeler.Rows[i].Cells[3].Value.ToString() == "ekle")
    {
        SqlCommand tahsilekle = new SqlCommand("insert into
Odemeler(OtelID,OdaNo,Odeme,OdemeTarih) values(@otelid,@odano,@odeme,@odemetarih)", con);
        tahsilekle.Parameters.AddWithValue("@otelid", OtelID);
        tahsilekle.Parameters.AddWithValue("@odano", cmbodano.Text);
        tahsilekle.Parameters.AddWithValue("@odeme",
datagridodemeler.Rows[i].Cells[2].Value);
        tahsilekle.Parameters.AddWithValue("@odemetarih",
datagridodemeler.Rows[i].Cells[1].Value);

        tahsilekle.ExecuteNonQuery();
        MessageBox.Show("Değişiklikler Başarı İle Kaydedildi", "Bilgi", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        datagridodemeler.Rows[i].Cells[3].Value = "0";
    }

}

con.Close();
SqlCommand toplamodeme = new SqlCommand("Select Odeme from Odemeler where
OdaNo=" + cmbodano.Text + " and OtelID=" + OtelID + "", con);
con.Open();
double toplamodemeupdate = 0;
SqlDataReader topla = toplamodeme.ExecuteReader();
while (topla.Read())
{
    toplamodemeupdate += double.Parse(topla[0].ToString());
}
con.Close();

SqlCommand komut = new SqlCommand("Select GenelToplam from K_Cikis where OdaNo="
+ cmbodano.Text + " and OtelID=" + OtelID + "", con);
con.Open();
SqlDataReader geneltoplamu = komut.ExecuteReader();
double geneltoplam = 0;
while (geneltoplamu.Read())
{
    geneltoplam = double.Parse(geneltoplamu[0].ToString());
}

```

```

        con.Close();
        con.Open();
        SqlCommand cikisupdate = new SqlCommand("update K_Cikis set
ToplamOdemeler=@toplalomdeme,OdencekTutar=@odeneciktutar where OdaNo=@odano and
OtelID=@otelid", con);
        cikisupdate.Parameters.AddWithValue("@otelid", OtelID);
        cikisupdate.Parameters.AddWithValue("@odano", cmbodano.Text);
        cikisupdate.Parameters.AddWithValue("@toplalomdeme", toplalomdemeupdate);
        cikisupdate.Parameters.AddWithValue("@odeneciktutar", geneltoplam - toplalomdemeupdate);
        cikisupdate.ExecuteNonQuery();
        MessageBox.Show("Değişiklikler Başarı İle Güncellendi", "Bilgi", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        con.Close();
        yenile();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```
private void tahsiltxt_TextChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void btn_ekleEx_Click(object sender, EventArgs e)
```

```
{
```

```

toplalomdemeler.Text = geneltxt.Text;
int i = datagridodemeler.RowCount;
datagridodemeler.Rows.Add();
datagridodemeler.Rows[i].Cells[0].Value = cmbodano.Text;
datagridodemeler.Rows[i].Cells[1].Value = DateTime.Now;
datagridodemeler.Rows[i].Cells[2].Value = Convert.ToDouble(odenecektxt.Text);
datagridodemeler.Rows[i].Cells[3].Value = "ekle";
tahsiltxt.Text = null;
odenecektxt.Text = 0.ToString();

```

```

}

private void cmbfiyatkode_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        SqlCommand cmd = new SqlCommand("Select ToplamOdaFiyati From K_Cikis where
OdaNo=" + cmbodano.Text + " and OtelID=" + OtelID + "", con);
        con.Open();
        SqlDataReader dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            if (cmbfiyatkode.SelectedIndex == 0)
            {
                cmbfiyatkode.SelectedIndex = 0;
                textBox1.Text = Convert.ToDouble(dr[0]).ToString();
            }
            else if (cmbfiyatkode.SelectedIndex == 1)
            {
                textBox1.Text = (Convert.ToDouble(dr[0].ToString()) / (r_konakdatagrid.RowCount -
1)).ToString();
            }
        }

        con.Close();
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}

```

```
}
```

## REZ\_TIKLA.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{
    public partial class rez_tikla : Form
    {
        public rez_tikla()
        {
            InitializeComponent();
        }

        int OtelID = Login.id;
        SqlConnection con = new SqlConnection(Login.veritabani);
        SqlConnection con2 = new SqlConnection(Login.veritabani);
        private void pictureBox1_Click(object sender, EventArgs e)
        {

    }

    public void gridyenile()//rezrapor tablosundan verileri çekiyor
    {
        try
        {
            DateTime mydatetime = DateTime.Now.Date;
            string sqlformat = mydatetime.ToString("yyyy-MM-dd");

```

```
SqlCommand cmd = new SqlCommand("Select TcNo,ad,soyad,GelisTarih from RezRapor where  
TcNo like '" + bunifuMaterialTextbox1.Text + "%' and OtelID=' " + OtelID + " ' and  
GelisTarih=' "+sqlformat+"'", con);
```

```
bunifuCustomDataGridView1.Rows.Clear();  
int a = 0;  
con.Open();  
SqlDataReader sdr = cmd.ExecuteReader();  
while (sdr.Read())  
{  
    bunifuCustomDataGridView1.Rows.Add();  
    bunifuCustomDataGridView1.Rows[a].Cells[0].Value = sdr["TcNo"].ToString();  
    bunifuCustomDataGridView1.Rows[a].Cells[1].Value = sdr["ad"].ToString();  
    bunifuCustomDataGridView1.Rows[a].Cells[2].Value = sdr["soyad"].ToString();  
    bunifuCustomDataGridView1.Rows[a].Cells[3].Value = sdr["GelisTarih"].ToString();  
    a++;  
}  
con.Close();  
}  
catch (Exception ex)  
{  
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);  
}  
  
}  
private void bunifuMaterialTextbox1_ValueChanged(object sender, EventArgs e)  
{  
    gridyenile();  
}  
  
private void btnrezekle_Click(object sender, EventArgs e)  
{  
    r_musteri_ekle ekle = new r_musteri_ekle();  
    ekle.Show();  
}  
  
private void rez_tikla_Load(object sender, EventArgs e)
```

```

{
    gridyenile();
}

private void bunifuCustomDataGrid1_CellClick(object sender, DataGridViewCellEventArgs e)
{
}

private void bunifuCustomDataGrid1_CellDoubleClick(object sender, DataGridViewCellEventArgs e)//hücreye 2 kere tıkladığımızda verileri r_konak datagridine ekliyor
{
    /// burada kaldık tc kimliğe göre datagrid sınırla hem oda gridi hem k_giris selecti yap
    try
    {

        int hata = 0; //kişi bu odada hatası
        int hata2 = 0; //kişi bu otelde hatası
        r_konak konak = Application.OpenForms["r_konak"] as r_konak;
        UInt64 tc =
        UInt64.Parse(bunifuCustomDataGrid1.Rows[bunifuCustomDataGrid1.CurrentRow.Index].Cells[0].Value.
        ToString());
        con.Open();
        con2.Open();

        SqlCommand datagonder = new SqlCommand("select TcKimlikNo,Ad,Soyad,Cinsiyeti,Uyruk
from R_Musteriekle where TcKimlikNo=" + tc + " and OtelID=" + OtelID + "", con2);
        SqlDataReader getir = datagonder.ExecuteReader();

        if (konak.r_konakdatagrid.RowCount - 1 > 0)
        {
            for (int satır = 0; satır < konak.r_konakdatagrid.RowCount - 1; satır++)
            {
                if (konak.r_konakdatagrid.Rows[satır].Cells[0].Value.ToString() == tc.ToString())
                    hata++;
            }
        }
    }
}

```

```

SqlCommand sorgu = new SqlCommand("select COUNT(TcKimlik) from K_Giris where
TcKimlik=" + tc + " and OtelID=" + OtelID + "", con);
SqlDataReader varmi = sorgu.ExecuteReader();
while (varmi.Read())
{
    if (int.Parse(varmi[0].ToString()) >= 1)
        hata2++;
}

if (hata == 0 && hata2 == 0)
{
    while (getir.Read())
    {
        konak.r_konakdatagrid.Rows.Add();
        konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[0].Value =
getir[0];
        konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[1].Value =
getir[1];
        konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[2].Value =
getir[2];
        if(Boolean.Parse(getir[3].ToString())==true)
        {
            konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[3].Value =
"Erkek";
        }
        else if (Boolean.Parse(getir[3].ToString()) == false)
        {
            konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[3].Value =
"Kadın";
        }

        if(Boolean.Parse(getir[4].ToString())== true)
        {
            konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[4].Value =
"Türk";
        }
        if (Boolean.Parse(getir[4].ToString()) == false)
        {

```

```

        konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[4].Value =
    "Yabancı";
}

        konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[5].Value =
konak.datetpGelis.Value.ToString("yyyy-MM-dd");
        konak.r_konakdatagrid.Rows[konak.r_konakdatagrid.RowCount - 2].Cells[6].Value =
konak.DatetpGidis.Value.ToString("yyyy-MM-dd");
        MessageBox.Show("Kişi odaya eklendi", "Bilgilendirme", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
}

else if (hata > 0)
    MessageBox.Show("Bu odada böyle bir kişi zaten mevcut.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
else if (hata2 > 0)
    MessageBox.Show("Bu otelde böyle bir kişi zaten mevcut.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);

con2.Close();
con.Close();

this.Hide();
konak.Focus();

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void rez_tikla_FormClosed(object sender, FormClosedEventArgs e)
{
    r_konak konak = new r_konak();
    konak.BringToFront();
}

```

```
    }  
}
```

## HATIRLATMAEKLE.CS

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Data.SqlClient;  
  
namespace Login  
{  
    public partial class Hatirlatmaekle : Form  
    {  
        public Hatirlatmaekle()  
        {  
            InitializeComponent();  
        }  
        SqlConnection con = new SqlConnection(Login.veritabani);  
        int OtelID = Login.id;  
        private void btn_kay_Click(object sender, EventArgs e)  
        {  
            try  
            {  
                SqlCommand hatirlatmaekle = new SqlCommand("insert into  
Hatirlatmalar(OtelID,HatirlatmaBaslik,HatirlatmaMetin) values(@otelid,@baslik,@metin)", con);  
                hatirlatmaekle.Parameters.AddWithValue("@otelid", OtelID);  
                hatirlatmaekle.Parameters.AddWithValue("@baslik", textBox1.Text);  
                hatirlatmaekle.Parameters.AddWithValue("@metin", richTextBox1.Text);  
                con.Open();  
                hatirlatmaekle.ExecuteNonQuery();  
                MessageBox.Show("Hatırlatma başarı ile eklenmiştir", "Bilgilendirme",  
MessageBoxButtons.OK, MessageBoxIcon.Information);  
            }  
        }  
    }  
}
```

```

        con.Close();
        yenile();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
void yenile()
{
    try
    {
        Resepsiyon resepsiyon = Application.OpenForms["Resepsiyon"] as Resepsiyon;
        resepsiyon.DataGridViewHatirlatmalar.Rows.Clear();
        SqlCommand cmd = new SqlCommand("Select Tarih,HatirlatmaBaslik,Hatirlatmano From
        Hatirlatmalar where OtelID=''' + OtelID + '''", con);
        con.Open();
        SqlDataReader dr = cmd.ExecuteReader();
        int i = 0;
        while (dr.Read())
        {
            resepsiyon.DataGridViewHatirlatmalar.Rows.Add();
            resepsiyon.DataGridViewHatirlatmalar.Rows[i].Cells[0].Value =
Convert.ToDateTime(dr[0].ToString());
            resepsiyon.DataGridViewHatirlatmalar.Rows[i].Cells[1].Value = dr[1].ToString();
            resepsiyon.DataGridViewHatirlatmalar.Rows[i].Cells[2].Value = dr[2].ToString();

            i++;
        }
        con.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void panel1_Paint(object sender, PaintEventArgs e)

```

```
{  
}  
}  
}  
}
```

## PROFİLLER.CS

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Data.SqlClient;
```

```
namespace Login
```

```
{  
    public partial class Profiller : Form
```

```
{  
    public Profiller()
```

```
{  
    InitializeComponent();
```

```
}
```

```
    SqlConnection con = new SqlConnection(Login.veritabani);
```

```
    SqlConnection con2= new SqlConnection(Login.veritabani);
```

```
    int otelid = Login.id;
```

```
    private void bunifuMaterialTextbox1_ValueChanged(object sender, EventArgs e)
```

```
{  
    SqlCommand cmd = new SqlCommand("Select Ad,Soyad From R_Musteriekle where TcKimlikNo  
like "+bunifuMaterialTextbox1.Text+"%' and OtelID='"+otelid+"'", con);
```

```
    try
```

```
{  
  
    if (con.State != ConnectionState.Open)  
        con.Open();  
  
    SqlDataReader dr = cmd.ExecuteReader();  
    listBox1.Items.Clear();  
    while (dr.Read())  
    {  
  
        listBox1.Items.Add(dr["Ad"].ToString() + " " + dr["Soyad"].ToString());  
  
    }  
    if (bunifuMaterialTextbox1.Text=="")  
    {  
        bunifuCustomDataGridView1.DataSource = null;  
        listBox1.Items.Clear();  
        temizle();  
    }  
}  
  
catch (Exception ex)  
{  
  
    MessageBox.Show("HATA" + ex.Message);  
  
}  
finally  
{  
    con.Close();  
  
}  
}
```

```

void listBox()
{
    SqlCommand cmd = new SqlCommand("Select Ad,Soyad From R_Musteriekle where TcKimlikNo
like '" + bunifuMaterialTextbox1.Text + "%' and OtelID='" + otelid + "'", con);

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

        SqlDataReader dr = cmd.ExecuteReader();
        listBox1.Items.Clear();
        while (dr.Read())
        {
            listBox1.Items.Add(dr["Ad"].ToString() + " " + dr["Soyad"].ToString());
        }

        if (bunifuMaterialTextbox1.Text == "")
        {
            bunifuCustomDataGridView1.DataSource = null;
            listBox1.Items.Clear();
            temizle();
        }
    }

    catch (Exception ex)
    {
        MessageBox.Show("HATA" + ex.Message);
    }

    finally
}

```

```
{  
    con.Close();  
  
}  
}//listboxtaki veriyi güncellemek için yaptım
```

```
void temizle()  
{  
    Tctxt.Clear();  
    Adtxt.Clear();  
    Soyadtxt.Clear();  
    Anatxt.Clear();  
    Babatxt.Clear();  
    Dogumyertxt.Clear();  
    dateTimePicker1.ResetText();  
    cmbuyruk.ResetText();  
    ceptxt.Clear();  
    Emailtxt.Clear();  
    istxt.Clear();  
    ikamettxt.Clear();  
    txtNotlar.ResetText();
```

```
}//textbox ve diğer nesneleri clearaladım
```

```
public void listBox1_SelectedIndexChanged(object sender, EventArgs e)  
{
```

```
    temizle();  
    if(listBox1.SelectedItem.ToString() != null)  
    {  
        string metin = listBox1.SelectedItem.ToString();
```

```
        SqlCommand cmd = new SqlCommand("Select  
TcKimlikNo,Ad,Soyad,BabaAd,AnaAd,DogumYeri,DogumTarihi,Uyruk,CepTelefon,Email,Musteri_is,ik
```

```

ametkah,Notlar from R_Musteriekle where TcKimlikNo like "+@tc+'%' and Ad+' '+Soyad=""+metin+""
and OtelID=""+otelid+"", con);
cmd.Parameters.AddWithValue("@tc", bunifuMaterialTextbox1.Text);

try
{

if (con.State != ConnectionState.Open) //bağlantıyı kontrol edip açık değilse açtım
    con.Open();

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read()) // datareaderden gerekli verileri aldım
{


Tctxt.Text = dr["TcKimlikNo"].ToString();
tlabel.Text = Tctxt.Text;
Adtxt.Text = dr["Ad"].ToString();
Soyadtxt.Text = dr["Soyad"].ToString();
Babatxt.Text = dr["BabaAd"].ToString();
Anatxt.Text = dr["AnaAd"].ToString();
Dogumyertxt.Text = dr["Dogumyeri"].ToString();
dateTimePicker1.Value = Convert.ToDateTime(dr["DogumTarihi"].ToString());


Boolean a =Convert.ToBoolean(dr["Uyruk"].ToString());
if(a==true)
{
    cmbuyruk.Items.Clear();
    cmbuyruk.Items.Add("Türk");
    cmbuyruk.SelectedIndex = 0;
}
else if(a==false)
{
    cmbuyruk.Items.Clear();
    cmbuyruk.Items.Add("Yabancı");
    cmbuyruk.SelectedIndex = 0;
}
}

```

```

ceptxt.Text = dr["CepTelefon"].ToString();
Emailtxt.Text = dr["Email"].ToString();
istxt.Text = dr["Musteri_is"].ToString();
ikamettxt.Text = dr["ikametkah"].ToString();
txtNotlar.Text = dr["Notlar"].ToString();

}

yenile();

}

catch (Exception ex)
{
    MessageBox.Show("HATA " + ex.Message);

}
finally
{
    con.Close();

}
}

private void bunifuMaterialTextbox1_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); //string veri girişini engelliyor
}

private void Profiller_Load(object sender, EventArgs e)
{

```

```

        this.bunifuCustomDataGridView1.DefaultCellStyle.Font = new Font("Century Gothic", 10);

    }

void yenile()
{
    try
    {

        SqlDataAdapter da = new SqlDataAdapter("select * from profiller where TcKimlikNo='"
Tctxt.Text + "' and OtelID='"
+ otelid + "'", con2);
        //profiler diye sqlde view oluşturдум onu excel için kullanacağız
        DataSet ds = new DataSet();
        da.Fill(ds);

        bunifuCustomDataGridView1.DataSource = ds.Tables[0];
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void btnkisiguncelle_Click(object sender, EventArgs e)
{
    try
    {

        if (Tctxt.Text != "" && Tctxt.TextLength == 11)
        {

            DialogResult cevap = MessageBox.Show("Kişinin Verileri Güncellenecek Emin misiniz?", "Güncelleme", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (cevap == DialogResult.Yes)
            {
                SqlCommand cmd = new SqlCommand("Update R_Musteriekle set
TcKimlikNo=@tc,Ad=@ad,Soyad=@soyad,BabaAd=@babaad,AnaAd=@anaad,Dogumyeri=@dy,Dogu

```

```
mTarihi=@dt,Uyruk=@uyruk,CepTelefon=@cep,Email=@email,Musteri_is=@is,Notlar=@nott,ikametka  
h=@ikametkah where TcKimlikNo=@aratc and OtelID="" + otelid + "", con);
```

```
cmd.Parameters.AddWithValue("@tc", Tctxt.Text);  
cmd.Parameters.AddWithValue("@ad", Adtxt.Text);  
cmd.Parameters.AddWithValue("@soyad", Soyadtxt.Text);  
cmd.Parameters.AddWithValue("@babaad", Babatxt.Text);  
cmd.Parameters.AddWithValue("@anaad", Anatxt.Text);  
// cmd.Parameters.AddWithValue("@dy", dateTimePicker1.Value.Date);  
cmd.Parameters.AddWithValue("@dy", Dogumyertxt.Text);  
cmd.Parameters.Add("@dt", SqlDbType.DateTime).Value =  
dateTimePicker1.Value.ToString("yyyy-MM-dd");
```

```
//cmd.Parameters["@dy"].Value = dateTimePicker1.Value;
```

```
if (cmbuyruk.Text == "Türk")  
{  
    cmd.Parameters.AddWithValue("@uyruk", true);  
}  
else if (cmbuyruk.Text == "Yabancı")  
{  
    cmd.Parameters.AddWithValue("@uyruk", false);  
}  
cmd.Parameters.AddWithValue("@cep", ceptxt.Text.ToString());  
cmd.Parameters.AddWithValue("@email", Emailtxt.Text);  
cmd.Parameters.AddWithValue("@is", istxt.Text);  
cmd.Parameters.AddWithValue("@nott", txtNotlar.Text);  
cmd.Parameters.AddWithValue("@aratc", tclabel.Text);  
cmd.Parameters.AddWithValue("@ikametkah", ikamettxt.Text);  
  
con.Open();  
cmd.ExecuteNonQuery();  
listbox();  
yenile();  
con.Close();
```

```
MessageBox.Show("Kişi başarı ile güncellendi", "Bilgi", MessageBoxButtons.OK,  
MessageBoxIcon.Information);
```

```

        }

        else
        {
            MessageBox.Show("Güncelleme işlemi iptal edildi", "Uyarı", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
    }

    else
    {
        MessageBox.Show("Kişi Bilgileri boş olduğu için güncelleme yapamazsınız!", "Uyarı",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
    }
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void cmbuyruk_MouseClick(object sender, MouseEventArgs e)
{
    if (cmbuyruk.Items.Count < 2)
    {
        if (cmbuyruk.Text == "Türk")
        {
            cmbuyruk.Items.Add("Yabancı");
        }

        else if (cmbuyruk.Text == "Yabancı")

```

```

    {
        cmbuyruk.Items.Add("Türk");
    }

}

private void Tctxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void ceptxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void Adtxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);
}

private void Soyadtxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);
}

private void Babatxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);
}

private void Anatxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);
}

```

```

}

private void istxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);
}

private void Dogumyertxt_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);
}

private void bunifuCustomDataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    }

}
}
}

```

## **KONAKLAMALAR.CS**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Collections;

namespace Login
{
    public partial class Konaklamalar : Form
    {
        public int OtelID = Login.id;
    }
}

```

```

public Konaklamalar()
{
    InitializeComponent();
}

SqlConnection baglanti = new SqlConnection(Login.veritabani);

private void Konaklama_Load(object sender, EventArgs e)
{
    gel_aralik.Value = false;
    gid_aralik.Value = false;
    gel_tar2.MinDate = gel_tar.Value;
    gid_tar2.MinDate = gel_tar.Value;
    radio_ayrilan.Checked = false;
    radio_icerde_blnn.Checked = false;
    radio_hepsi.Checked = true;
    tc.MaxLength = 11;
}

private void bunifuToggleSwitch1_ValueChanged(object sender, EventArgs e)
{
    if (gel_aralik.Value == true)
    {
        gel_tar2.Enabled = true;
        gel_tar2.IconColor = Color.Green;
    }
    else
    {
        gel_tar2.Enabled = false;
        gel_tar2.IconColor = Color.Gray;
    }
}

private void bunifuToggleSwitch2_ValueChanged(object sender, EventArgs e)
{
    if (gid_aralik.Value == true)
    {
        gid_tar2.Enabled = true;
        gid_tar2.IconColor = Color.Green;
    }
}

```

```

else
{
    gid_tar2.Enabled = false;
    gid_tar2.IconColor = Color.Gray;
}
}

private void PictureBox1_Click(object sender, EventArgs e)
{
    try
    {
        rapor.Rows.Clear();
        baglanti.Open();
        SqlCommand komut = new SqlCommand("SELECT * from KonaklamaRapor where
TcKimlik=" + Convert.ToInt64(tc.Text) + " and OtelID=" + OtelID + "", baglanti);
        SqlDataReader sa = komut.ExecuteReader();
        int a = 0;
        while (sa.Read())
        {
            rapor.Rows.Add();
            rapor.Rows[a].Cells[0].Value = sa[0].ToString();
            rapor.Rows[a].Cells[1].Value = sa[1].ToString().Substring(0, 10);
            rapor.Rows[a].Cells[2].Value = sa[2].ToString().Substring(0, 10);
            rapor.Rows[a].Cells[3].Value = sa[3].ToString();
            rapor.Rows[a].Cells[4].Value = sa[4].ToString();
            rapor.Rows[a].Cells[5].Value = sa[5].ToString();
            a++;
        }
        baglanti.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void Tc_ValueChanged(object sender, EventArgs e)

```

```

{
    if (tc.Text.Length > 0)
    {
        radio_ayrilan.Enabled = false;
        radio_icerde_blnn.Enabled = false;
        radio_hepsi.Enabled = false;

        gel_tar.Enabled = false;
        gel_tar.IconColor = Color.Gray;
        gel_tar2.Enabled = false;
        gel_tar2.IconColor = Color.Gray;
        gel_aralik.Value = false;
        gid_tar.Enabled = false;
        gid_tar.IconColor = Color.Gray;
        gid_tar2.Enabled = false;
        gid_tar2.IconColor = Color.Gray;
        gid_aralik.Value = false;
    }

    else
    {
        gel_tar.Enabled = true;
        gel_tar.IconColor = Color.Green;
        gid_tar.Enabled = true;
        gid_tar.IconColor = Color.Green;
        rapor.Rows.Clear();
    }
}

private void Rapor_Goster_Click(object sender, EventArgs e)
{
    if (gel_aralik.Value == true && gid_aralik.Value == true)
    {
        aralik();
        if (radio_ayrilan.Checked == true)
        {

```

```

        SqlCommand komut = new SqlCommand("SELECT * from KonaklamaRapor where
(GelisTarihi BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and (GidisTarihi BETWEEN '" +
gid_tar.Text + "' and '" + gid_tar2.Text + "') and Durum=0 and OtelID=''' + OtelID + '''", baglanti);
        getir1(komut);
    }
    else if (radio_icerde_blnn.Checked == true)
    {
        SqlCommand komut = new SqlCommand("SELECT * from KonaklamaRapor where
(GelisTarihi BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and (GidisTarihi BETWEEN '" +
gid_tar.Text + "' and '" + gid_tar2.Text + "') and Durum=1 and OtelID=''' + OtelID + '''", baglanti);
        getir1(komut);
    }
    else if (radio_hepsi.Checked == true)
    {
        SqlCommand komut = new SqlCommand("SELECT * from KonaklamaRapor where
(GelisTarihi BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and (GidisTarihi BETWEEN '" +
gid_tar.Text + "' and '" + gid_tar2.Text + "') and (Durum=0 or Durum=1) and OtelID=''' + OtelID + '''",
baglanti);
        getir1(komut);
    }
    aralik2();
}
else if (gel_aralik.Value == true && gid_aralik.Value == false)
{
    aralik();
    if (radio_ayrilan.Checked == true)
    {
        SqlCommand komut = new SqlCommand("SELECT * from KonaklamaRapor where
(GelisTarihi BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and (GidisTarihi<=''' +
gid_tar.Text + '') and Durum=0 and OtelID=''' + OtelID + ''', baglanti");
        getir1(komut);
    }
    else if (radio_icerde_blnn.Checked == true)
    {
        SqlCommand komut = new SqlCommand("SELECT * from KonaklamaRapor where
(GelisTarihi BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and (GidisTarihi<=''' +
gid_tar.Text + '') and Durum=1 and OtelID=''' + OtelID + '''", baglanti);
        getir1(komut);
    }
}

```

```

else if (radio_hepsi.Checked == true)
{
    SqlCommand komut = new SqlCommand("SELECT * from KonaklamaRapor where
(GelisTarihi BETWEEN '" + gel_tar.Text + "' and '" + gel_tar2.Text + "') and (GidisTarihi<=''" +
gid_tar.Text + "') and (Durum=0 or Durum=1) and OtelID=''' + OtelID + '''", baglanti);
    getir1(komut);
}
aralik2();
}

else if (gel_aralik.Value == false && gid_aralik.Value == true)
{
    aralik();
    if (radio_ayrulan.Checked == true)
    {
        SqlCommand komut = new SqlCommand("SELECT * from Konaklamarapor where
GelisTarihi>=''' + gel_tar.Text + ''' and (GidisTarihi BETWEEN '' + gid_tar.Text + '' and '' +
gid_tar2.Text + '') and Durum=0 and OtelID=''' + OtelID + '''", baglanti);
        getir1(komut);
    }
    else if (radio_icerde_blnn.Checked == true)
    {
        SqlCommand komut = new SqlCommand("SELECT * from Konaklamarapor where
GelisTarihi>=''' + gel_tar.Text + ''' and (GidisTarihi BETWEEN '' + gid_tar.Text + '' and '' +
gid_tar2.Text + '') and Durum=1 and OtelID=''' + OtelID + '''", baglanti);
        getir1(komut);
    }
    else if (radio_hepsi.Checked == true)
    {
        SqlCommand komut = new SqlCommand("SELECT * from Konaklamarapor where
GelisTarihi>=''' + gel_tar.Text + ''' and (GidisTarihi BETWEEN '' + gid_tar.Text + '' and '' +
gid_tar2.Text + '') and (Durum=0 or Durum=1) and OtelID=''' + OtelID + '''", baglanti);
        getir1(komut);
    }
    aralik2();
}
else if(gel_aralik.Value == false && gid_aralik.Value == false)
{
    aralik();
    if (radio_ayrulan.Checked == true)

```

```

{
    SqlCommand komut = new SqlCommand("SELECT * from Konaklamarapor where
(GelisTarihi>="" + gel_tar.Text + "" and GidisTarihi<="" + gid_tar.Text + "") and Durum=0 and OtelID="" +
OtelID + "", baglanti);
    getir1(komut);
}
else if (radio_icerde_blnn.Checked == true)
{
    SqlCommand komut = new SqlCommand("SELECT * from Konaklamarapor where
(GelisTarihi>="" + gel_tar.Text + "" and GidisTarihi<="" + gid_tar.Text + "") and Durum=1 and OtelID="" +
OtelID + "", baglanti);
    getir1(komut);
}
else if (radio_hepsi.Checked == true)
{
    SqlCommand komut = new SqlCommand("SELECT * from Konaklamarapor where
(GelisTarihi>="" + gel_tar.Text + "" and GidisTarihi<="" + gid_tar.Text + "") and (Durum=0 or Durum=1)
and OtelID="" + OtelID + "", baglanti);
    getir1(komut);
}
aralik2();
}

void aralik()
{
    gel_tar.CustomFormat = "MM/dd/yyyy";
    gel_tar2.CustomFormat = "MM/dd/yyyy";
    gid_tar.CustomFormat = "MM/dd/yyyy";
    gid_tar2.CustomFormat = "MM/dd/yyyy";
    gel_tar.Format = DateTimePickerFormat.Custom;
    gel_tar2.Format = DateTimePickerFormat.Custom;
    gid_tar.Format = DateTimePickerFormat.Custom;
    gid_tar2.Format = DateTimePickerFormat.Custom;
}

void aralik2()
{

```

```

gel_tar.Format = DateTimePickerFormat.Long;
gel_tar2.Format = DateTimePickerFormat.Long;
gid_tar.Format = DateTimePickerFormat.Long;
gid_tar2.Format = DateTimePickerFormat.Long;
}
void getir1(SqlCommand a)
{
    rapor.Rows.Clear();
    baglanti.Open();
    SqlDataReader oku = a.ExecuteReader();
    aralik2();
    int sayac = 0;
    while (oku.Read())
    {
        rapor.Rows.Add();
        rapor.Rows[sayac].Cells[0].Value = oku[0].ToString();
        rapor.Rows[sayac].Cells[1].Value = oku[1].ToString().Substring(0, 10);
        rapor.Rows[sayac].Cells[2].Value = oku[2].ToString().Substring(0, 10);
        rapor.Rows[sayac].Cells[3].Value = oku[3].ToString();
        rapor.Rows[sayac].Cells[4].Value = oku[4].ToString();
        rapor.Rows[sayac].Cells[5].Value = oku[5].ToString();
        sayac++;
    }
    baglanti.Close();
}

private void Gel_tar_ValueChanged(object sender, EventArgs e)
{
    gel_tar2.MinDate = gel_tar.Value;
    gel_tar2.Value = gel_tar.Value;

    gid_tar.MinDate = gel_tar.Value;
    gid_tar.Value = gel_tar.Value;
}

private void Gid_tar_ValueChanged(object sender, EventArgs e)
{
    gid_tar2.MinDate = gid_tar.MinDate;
}

```

```

        gid_tar2.Value = gid_tar.Value;

    }

//Yazdırma İşlemi
StringFormat strFormat;
ArrayList arrColumnLefts = new ArrayList();
ArrayList arrColumnWidths = new ArrayList();
int iCellHeight = 0;
int iTotWidth = 0;
int iRow = 0;
bool bFirstPage = false;
bool bNewPage = false;
int iHeaderHeight = 0;
private void PrintDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
{
    try
    {
        int iLeftMargin = e.MarginBounds.Left;
        int iTopMargin = e.MarginBounds.Top;
        bool bMorePagesToPrint = false;
        int iTmpWidth = 0;
        bFirstPage = true;

        if (bFirstPage)
        {
            foreach (DataGridViewColumn GridCol in rapor.Columns)
            {
                iTmpWidth = (int)(Math.Floor((double)((double)GridCol.Width /
                    (double)iTotalWidth * (double)iTotalWidth *
                    (double)e.MarginBounds.Width / (double)iTotalWidth)));
                iHeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,
                    GridCol.InheritedStyle.Font, iTmpWidth).Height) + 11;
                arrColumnLefts.Add(iLeftMargin);

```

```

        arrColumnWidths.Add(iTmpWidth);
        iLeftMargin += iTmpWidth;
    }
}

while (iRow <= rapor.Rows.Count - 1)
{
    DataGridViewRow GridRow = rapor.Rows[iRow];

    iCellHeight = GridRow.Height + 5;
    int iCount = 0;

    if (iTopMargin + iCellHeight >= e.MarginBounds.Height + e.MarginBounds.Top)
    {
        bNewPage = true;
        bFirstPage = false;
        bMorePagesToPrint = true;
        break;
    }
    else
    {
        if (bNewPage)
        {

e.Graphics.DrawString("Konaklama Raporu", new Font(rapor.Font, FontStyle.Bold),
                    Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -
                    e.Graphics.MeasureString("Konaklama Raporu", new Font(rapor.Font,
                    FontStyle.Bold), e.MarginBounds.Width).Height - 13);

        String strDate = DateTime.Now.ToString(" " +
DateTime.Now.ToString("MM/dd/yyyy HH:mm:ss");

        e.Graphics.DrawString(strDate, new Font(rapor.Font, FontStyle.Bold),
                    Brushes.Black, e.MarginBounds.Left + (e.MarginBounds.Width -
                    e.Graphics.MeasureString(strDate, new Font(rapor.Font,
                    FontStyle.Bold), e.MarginBounds.Width).Width), e.MarginBounds.Top -
                    e.Graphics.MeasureString("Konaklama Raporu", new Font(new Font(rapor.Font,
                    FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);
    }
}
}

```

```

iTopMargin = e.MarginBounds.Top;
foreach (DataGridViewColumn GridCol in rapor.Columns)
{
    e.Graphics.FillRectangle(new SolidBrush(Color.LightGray),
        new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
        (int)arrColumnWidths[iCount], iHeaderHeight));

    e.Graphics.DrawRectangle(Pens.Black,
        new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
        (int)arrColumnWidths[iCount], iHeaderHeight));

    e.Graphics.DrawString(GridCol.HeaderText, GridCol.InheritedStyle.Font,
        new SolidBrush(GridCol.InheritedStyle.ForeColor),
        new RectangleF((int)arrColumnLefts[iCount], iTopMargin,
        (int)arrColumnWidths[iCount], iHeaderHeight), strFormat);
    iCount++;
}
bNewPage = false;
iTopMargin += iHeaderHeight;
}

iCount = 0;

foreach (DataGridViewCell Cel in GridRow.Cells)
{
    if (Cel.Value != null)
    {
        e.Graphics.DrawString(Cel.Value.ToString(), Cel.InheritedStyle.Font,
            new SolidBrush(Cel.InheritedStyle.ForeColor),
            new RectangleF((int)arrColumnLefts[iCount], (float)iTopMargin,
            (int)arrColumnWidths[iCount], (float)iCellHeight), strFormat);
    }
}

e.Graphics.DrawRectangle(Pens.Black, new Rectangle((int)arrColumnLefts[iCount],
    iTopMargin, (int)arrColumnWidths[iCount], iCellHeight));

iCount++;
}
}

```

```

        iRow++;
        iTopMargin += iCellHeight;
    }

    if (bMorePagesToPrint)
        e.HasMorePages = true;
    else
        e.HasMorePages = false;
    }

    catch (Exception exc)
    {
        MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void PrintDocument1_BeginPrint(object sender, System.Drawing.Printing.PrintEventArgs e)
{
    try
    {
        strFormat = new StringFormat();
        strFormat.Alignment = StringAlignment.Near;
        strFormat.LineAlignment = StringAlignment.Center;
        strFormat.Trimming = StringTrimming.EllipsisCharacter;

        arrColumnLefts.Clear();
        arrColumnWidths.Clear();
        iCellHeight = 0;
        iRow = 0;
        bFirstPage = true;
        bNewPage = true;

        iTotWidth = 0;
        foreach (DataGridViewColumn dgvGridCol in rapor.Columns)
        {
            iTotWidth += dgvGridCol.Width;
        }
    }

    catch (Exception ex)
}

```

```
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void Gel_tar2_ValueChanged(object sender, EventArgs e)
    {
        }
    }
}
```

ODASİPARİŞ.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using Microsoft.VisualBasic;

namespace Login
{
    public partial class OdaSiparis : Form
    {
        SqlConnection baglan = new SqlConnection(Login.veritabani);
        SqlConnection baglan2 = new SqlConnection(Login.veritabani);

        int otelid = Login.id;

        public OdaSiparis()
        {
            InitializeComponent();
        }
}
```

```

private void OdaSiparis_Load(object sender, EventArgs e)
{
    ListleriYenile();
}

public void ListleriYenile()
{
    try
    {
        listBox2.Items.Clear();
        listBox3.Items.Clear();
        listBox2.Items.Add("Açık Yiyecek");
        listBox3.Items.Add("Açık İçecek");
        TxtIcecekFiyat.Enabled = false;
        TxtYiyecekFiyat.Enabled = false;
        CmbOda.Items.Clear();
        SqlCommand odalist = new SqlCommand("select OdaNo from Odalar where OtelId=" + otelid +
        " and Durum=1", baglan);
        baglan.Open();
        SqlDataReader getir = odalist.ExecuteReader();
        while (getir.Read())
        {
            CmbOda.Items.Add(getir[0]);
        }
        baglan.Close();

        SqlCommand yiyecek = new SqlCommand("select YiyecekAd from Yiyecekler where OtelID=" +
        + otelid + "", baglan);
        baglan.Open();
        SqlDataReader getiry = yiyecek.ExecuteReader();
        while (getiry.Read())
        {
            listBox2.Items.Add(getiry[0]);
        }
        baglan.Close();

        SqlCommand icecek = new SqlCommand("select IcecekAd from Icecekler where OtelID=" +
        + otelid + "", baglan);
    }
}

```

```

        baglan.Open();
        SqlDataReader getiri = icecek.ExecuteReader();
        while (getiri.Read())
        {
            listBox3.Items.Add(getiri[0]);
        }
        baglan.Close();
        DataGridYenile();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        TxtYiyecekAd.Text = listBox2.SelectedItem.ToString();
    }
    catch (Exception)
    {
    }
}

private void listBox3_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        TxtIcecekAd.Text = listBox3.SelectedItem.ToString();
    }
    catch (Exception)
    {
    }
}

```

```

private void TxtYiyecekAd_TextChanged(object sender, EventArgs e)
{
    if (TxtYiyecekAd.Text == "Açık Yiyecek")
    {
        numericUpDown1.Text = "1";
        TxtYiyecekFiyat.Text = "0";
        TxtYiyecekFiyat.Enabled = true;
        listBox2.Items.Clear();
    }
    else
    {
        TxtYiyecekFiyat.Enabled = false;
        listBox2.Items.Clear();
        try
        {
            SqlCommand yiyecek = new SqlCommand("select YiyecekAd from Yiyecekler where
OtelID=" + otelid + " and YiyecekAd like '%" + TxtYiyecekAd.Text + "%'", baglan);
            baglan.Open();
            SqlDataReader getiry = yiyecek.ExecuteReader();
            while (getiry.Read())
            {
                listBox2.Items.Add(getiry[0]);
            }
            baglan.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

try
{
    if (TxtYiyecekAd.Text == listBox2.Items[0].ToString())
    {
        numericUpDown1.Text = "2";
        numericUpDown1.Text = "1";
    }
}

```

```

else if (TxtYiyecekAd.Text == "")
{
    listBox2.Items.Insert(0, "Açık Yiyecek");
    numericUpDown1.Text = "1";
    TxtYiyecekFiyat.Text = "0";
}
}

catch (Exception)
{
}

private void TxtIcecekAd_TextChanged(object sender, EventArgs e)
{
    if (TxtIcecekAd.Text == "Açık İçecek")
    {
        numericUpDown2.Text = "1";
        TxtIcecekFiyat.Text = "0";
        TxtIcecekFiyat.Enabled = true;
        listBox3.Items.Clear();
    }
    else
    {
        TxtIcecekFiyat.Enabled = false;
        listBox3.Items.Clear();
        try
        {
            SqlCommand icecek = new SqlCommand("select IcecekAd from Icecekler where OtelID=" + 
otelid + " and IcecekAd like '%" + TxtIcecekAd.Text + "%'", baglan);
            baglan.Open();
            SqlDataReader getiri = icecek.ExecuteReader();
            while (getiri.Read())
            {
                listBox3.Items.Add(getiri[0]);
            }
            baglan.Close();
        }
    }
}

```

```

        catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

    try
    {
        if (TxtIcecekAd.Text == listBox3.Items[0].ToString())
        {
            numericUpDown2.Text = "2";
            numericUpDown2.Text = "1";
        }
        else if (TxtIcecekAd.Text == "")
        {
            listBox3.Items.Insert(0, "Açık İçecek");
            numericUpDown2.Text = "1";
            TxtIcecekFiyat.Text = "0";
        }
    }
    catch (Exception)
    {
    }
}
}

```

```

private void numericUpDown1_ValueChanged(object sender, EventArgs e)
{
    try
    {
        SqlCommand yiyecek = new SqlCommand("select YiyecekFiyat from Yiyecekler where
OtelID=" + otelid + " and YiyecekAd=" + TxtYiyecekAd.Text + "", baglan);
        baglan.Open();
        SqlDataReader getiry = yiyecek.ExecuteReader();
        while (getiry.Read())
        {

```

```

        TxtYiyecekFiyat.Text = (double.Parse(getiry[0].ToString()) *
double.Parse(numericUpDown1.Value.ToString())).ToString();
    }
    baglan.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void bunifuFlatButton1_Click(object sender, EventArgs e)
{
    if (TxtYiyecekAd.Text == "Açık Yiyecek" && TxtAcikYiyecekAd.Text != "")
    {
        if (TxtYiyecekFiyat.Text != "")
        {
            if (numericUpDown1.Value > 0 && double.Parse(TxtYiyecekFiyat.Text) > 0)
            {
                listBox1.Items.Add(numericUpDown1.Text + " - " + TxtAcikYiyecekAd.Text + " = " +
TxtYiyecekFiyat.Text);
                TxtYiyecekAd.Text = "";
                TxtAcikYiyecekAd.Text = "";
            }
            else
                MessageBox.Show("Lütfen açık yiyeceğe adet, ve tutar giriniz.", "UYARI",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
        }
        else
            MessageBox.Show("Tutar alanı boş geçilemez.", "UYARI", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
    }

}
else
{
    SqlCommand yiyecek = new SqlCommand("select YiyecekAd from Yiyecekler where OtelID=@"
+ otelid + "" and YiyecekAd=@" + TxtYiyecekAd.Text + "", baglan);
}

```

```

baglan.Open();
SqlDataReader onay = yiyecek.ExecuteReader();
onay.Read();
try {
    if (onay[0].ToString() != "")
    {
        if (numericUpDown1.Value > 0 && double.Parse(TxtYiyecekFiyat.Text) > 0 &&
TxtYiyecekFiyat.Text != "")
        {
            listBox1.Items.Add(numericUpDown1.Text + "-" + TxtYiyecekAd.Text + " = " +
TxtYiyecekFiyat.Text);
        }
        else
            MessageBox.Show("Lütfen açık yiyeceğe adet, ve tutar giriniz.", "UYARI",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
    }
    catch (Exception)
    {
        MessageBox.Show("Böyle bir ürün
bulunmamakta.", "UYARI", MessageBoxButtons.OK, MessageBoxIcon.Stop);
    }
    baglan.Close();
    TxtYiyecekAd.Text = "";
    TxtAcikYiyecekAd.Text = "";
}

```

```

private void bunifuFlatButton5_Click(object sender, EventArgs e)
{
    if (TxtIcecekAd.Text == "Açık İçecek" && TxtAcikIcecekAd.Text != "")
    {
        if (TxtIcecekFiyat.Text != "")
        {
            if (numericUpDown2.Value > 0 && double.Parse(TxtIcecekFiyat.Text) > 0)
            {
                listBox1.Items.Add(numericUpDown2.Text + "-" + TxtAcikIcecekAd.Text + " = " +
TxtIcecekFiyat.Text);
                TxtIcecekAd.Text = "";
            }
        }
    }
}

```

```

        TxtAcikIcecekAd.Text = "";
    }
    else
        MessageBox.Show("Lütfen açık içeceğe adet,ve tutar giriniz.", "UYARI",
    MessageBoxButtons.OK, MessageBoxIcon.Stop);
}
else
    MessageBox.Show("Tutar alanı boş geçilemez.", "UYARI", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
}

else
{
SqlCommand icecek = new SqlCommand("select IcecekAd from Icecekler where OtelID=" +
otelid + " and IcecekAd='" + TxtIcecekAd.Text + "'", baglan);
baglan.Open();
SqlDataReader onay = icecek.ExecuteReader();
onay.Read();
try
{
    if (onay[0].ToString() != "")
    {
        if (numericUpDown2.Value > 0 && double.Parse(TxtIcecekFiyat.Text) > 0 &&
TxtIcecekFiyat.Text != "")
        {
            listBox1.Items.Add(numericUpDown2.Text + "-" + TxtIcecekAd.Text + " = " +
TxtIcecekFiyat.Text);
        }
        else
            MessageBox.Show("Lütfen açık içeceğe adet,ve tutar giriniz.", "UYARI",
MessageBoxButtons.OK, MessageBoxIcon.Stop);
    }
}
catch (Exception)
{
    MessageBox.Show("Böyle bir ürün bulunmamakta.", "UYARI", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
}
baglan.Close();
TxtIcecekAd.Text = "";

```

```
        TxtAcikIcecekAd.Text = "";
    }
}
```

```
private void bunifuFlatButton4_Click(object sender, EventArgs e)
{
    CmbOda.Text = "";
    TxtYiyecekAd.Text = "";
    TxtIcecekAd.Text = "";
    numericUpDown1.Text = "";
    numericUpDown2.Text = "";
    TxtIcecekFiyat.Text = "";
    TxtYiyecekFiyat.Text = "";
    TxtAcikIcecekAd.Text = "";
    TxtAcikYiyecekAd.Text = "";
    listBox1.Items.Clear();
}
```

```
private void bunifuFlatButton2_Click(object sender, EventArgs e)
{
    SqlCommand odakontrol = new SqlCommand("select OdaNo from Odalar where
OtelID='"+otelid+"' and OdaNo='"+CmbOda.Text+"'",baglan);
    baglan.Open();
    SqlDataReader kontroloku = odakontrol.ExecuteReader();
    kontroloku.Read();
    try
    {
        if (kontroloku[0].ToString() != "" && listBox1.Items.Count > 0)
        {
            string siparis = "";
            float tutar=0;
            for (int a = 0; a < listBox1.Items.Count; a++)
            {
                string ayir = listBox1.Items[a].ToString();
                string[] tutarayir = ayir.Split('=');
                tutar = tutar + float.Parse(tutarayir[1]);
                siparis += listBox1.Items[a];
                if (a != listBox1.Items.Count - 1)
                { siparis += " / "; }
            }
        }
    }
}
```

```

        }

        baglan2.Open();
        string tutargir = tutar.ToString().Replace(",",".");
        SqlCommand ekle = new SqlCommand("insert into
Siparisler(OtelID,OdaNo,Siparis,Tutar,SiparisDurum) values(" + otelid + "," + CmbOda.Text + "," +
siparis + "','" + tutargir + ",1)", baglan2);
        ekle.ExecuteNonQuery();
        MessageBox.Show("Siparişler başarıyla eklendi.", "BİLGİLENDİRME",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        baglan2.Close();
        listBox1.Items.Clear();
        CmbOda.Text = null;
    }

    else if (kontroloku[0].ToString() != "" && listBox1.Items.Count == 0)
    {
        MessageBox.Show("Lütfen sipariş için ürünler seçiniz.", "UYARI", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
    }

}

catch (Exception)
{
    MessageBox.Show("Lütfen sipariş için oda ve ürünler seçiniz.", "UYARI",
MessageBoxButtons.OK, MessageBoxIcon.Stop);

}

baglan.Close();
DataGridYenile();
}

private void numericUpDown2_ValueChanged(object sender, EventArgs e)
{
    try
    {
        SqlCommand icecek = new SqlCommand("select IcecekFiyat from Icecekler where OtelID='"
+ otelid + "' and IcecekAd='" + TxtIcecekAd.Text + "'", baglan);
        baglan.Open();
        SqlDataReader getiri = icecek.ExecuteReader();
        while (getiri.Read())

```

```

    {
        TxtIcecekFiyat.Text = (double.Parse(getiri[0].ToString()) *
double.Parse(numericUpDown2.Value.ToString())).ToString();
    }
    baglan.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void numericUpDown1_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void numericUpDown2_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void TxtYiyecekFiyat_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar) && e.KeyChar != ',';
}

private void TxtIcecekFiyat_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar) && e.KeyChar != ',';
}

```

```

private void listBox1_DoubleClick(object sender, EventArgs e)
{
    try
    {if(listBox1.SelectedItem.ToString()!="")
    {
        DialogResult cevap = MessageBox.Show("Bu sipariş listeden silinecek, emin misiniz",
"UYARI", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (cevap == DialogResult.Yes)
            listBox1.Items.RemoveAt(listBox1.SelectedIndex);
    }
}
catch (Exception)
{
}

private void TxtAcikYiyecekAd_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsWhiteSpace(e.KeyChar) && !char.IsPunctuation(e.KeyChar);
}

private void TxtAcikIcecekAd_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsWhiteSpace(e.KeyChar) && !char.IsPunctuation(e.KeyChar);
}

private void DataGridYenile()
{
    try
    {
        int a = 0;
        bunifuCustomDataGrid1.Rows.Clear();
        SqlCommand yenile = new SqlCommand("select SiparisID,OdaNo,Siparis,Tutar from Siparisler
where OtelID=''' + otelid + ''' and SiparisDurum=1", baglan);
        baglan.Open();
    }
}

```

```

SqlDataReader getir = yenile.ExecuteReader();
while (getir.Read())
{
    bunifuCustomDataGrid1.Rows.Add();
    bunifuCustomDataGrid1.Rows[a].Cells[0].Value = getir[0];
    bunifuCustomDataGrid1.Rows[a].Cells[1].Value = getir[1];
    bunifuCustomDataGrid1.Rows[a].Cells[2].Value = getir[2];
    bunifuCustomDataGrid1.Rows[a].Cells[3].Value = float.Parse(getir[3].ToString()) + "₺";
    a++;
}

}
baglan.Close();
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void bunifuCustomDataGrid1_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        int satır = bunifuCustomDataGrid1.CurrentRow.Index;
        DialogResult cevap = MessageBox.Show("Siparişi tamamlandığını onaylıyor musunuz?" + "\n" +
+ "Onaylamak için EVET," + "\n" + "Siparişten çıkmak için HAYIR" + "\n" + "Siparişi iptal etmek için İPTAL'e basınız.", "ODA " + bunifuCustomDataGrid1.Rows[satır].Cells[0].Value.ToString() + " SİPARİŞİ", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
        baglan.Open();
        if (cevap == DialogResult.Yes)
        {
            SqlCommand onay = new SqlCommand("update Siparisler set SiparisDurum=0 where SiparisID=" + bunifuCustomDataGrid1.Rows[satır].Cells[0].Value + "", baglan);
            onay.ExecuteNonQuery();
            MessageBox.Show("Sipariş tamamlandı.", "BİLGİLENDİRME", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}

```

```

        double tutar=0;
        SqlCommand tutartopla = new SqlCommand("select Tutar from Siparisler where
SiparisDurum=0 and OdaNo=""" + bunifuCustomDataGrid1.Rows[satir].Cells[1].Value + "" and OtelID=""" +
otelid + """", baglan2);
        baglan2.Open();
        SqlDataReader getir = tutartopla.ExecuteReader();
        while(getir.Read())
        {
            tutar+=double.Parse(getir[0].ToString());
        }
        baglan2.Close();
        double odafiyat = 0;
        double odenecek = 0;
        SqlCommand odafiyattopla = new SqlCommand("select ToplamOdaFiyati,OdenecekTutar
from K_Cikis where OdaNo=""" + bunifuCustomDataGrid1.Rows[satir].Cells[1].Value + "" and OtelID=""" +
otelid + """", baglan2);
        baglan2.Open();
        SqlDataReader topla = odafiyattopla.ExecuteReader();
        while (topla.Read())
        {
            odafiyat = double.Parse(topla[0].ToString());
            odenecek = double.Parse(topla[1].ToString());
        }
        baglan2.Close();
        SqlCommand update = new SqlCommand("update K_Cikis set
EkstraHarcamalar=@extra,GenelToplam=@genel,OdenecekTutar=@odenecek where OdaNo=""" +
bunifuCustomDataGrid1.Rows[satir].Cells[1].Value + "" and OtelID=""" + otelid + """", baglan);
        update.Parameters.AddWithValue("@extra", tutar);
        update.Parameters.AddWithValue("@genel", odafiyat+tutar);
        string isaretcikar = bunifuCustomDataGrid1.Rows[satir].Cells[3].Value.ToString();
        isaretcikar = isaretcikar.Replace("₺", "");
        update.Parameters.AddWithValue("@odenecek", odenecek+double.Parse(isaretcikar));
        update.ExecuteNonQuery();
    }
    else if (cevap == DialogResult.Cancel)
    {
        SqlCommand sil = new SqlCommand("delete from Siparisler where SiparisID=" +
bunifuCustomDataGrid1.Rows[satir].Cells[0].Value + "", baglan);
        sil.ExecuteNonQuery();
    }

```

```

        MessageBox.Show("Sipariş iptal edildi.", "UYARI", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    baglan.Close();
    DataGridYenile();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}
}
}
}

```

## **URUNEKLE.CS**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{
    public partial class UrunEkle : Form
    {
        SqlConnection baglan = new SqlConnection(Login.veritabani);
        SqlConnection baglan2 = new SqlConnection(Login.veritabani);

        int otelid = Login.id;

        public UrunEkle()
    }
}

```

```

{
    InitializeComponent();
}

private void TxtYiyecekFiyat_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar) && e.KeyChar != ',';
}

private void TxtIcecekFiyat_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar) && e.KeyChar != ',';
}

private void TxtYiyecekAd_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsWhiteSpace(e.KeyChar) && !char.IsPunctuation(e.KeyChar);

}

private void TxtIcecekAd_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsWhiteSpace(e.KeyChar) && !char.IsPunctuation(e.KeyChar);

}

private void DataGridYenile1()
{
    try
    {
        int a = 0;
        bunifuCustomDataGrid1.Rows.Clear();
        SqlCommand yenile = new SqlCommand("select YiyecekAd,YiyecekFiyat from Yiyecekler
where OtelID=''' + otelid + """", baglan);
        baglan.Open();
        SqlDataReader getir = yenile.ExecuteReader();
        while (getir.Read())
        {

```

```

        bunifuCustomDataGrid1.Rows.Add();
        bunifuCustomDataGrid1.Rows[a].Cells[0].Value = getir[0];
        bunifuCustomDataGrid1.Rows[a].Cells[1].Value = double.Parse(getir[1].ToString()) + "₺";
        a++;
    }
    baglan.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void DataGridYenile2()
{
    try
    {
        int a = 0;
        bunifuCustomDataGrid2.Rows.Clear();
        SqlCommand yenile = new SqlCommand("select IcecekAd,IcecekFiyat from Icecekler where
OtelID='' + otelid + """", baglan);
        baglan.Open();
        SqlDataReader getir = yenile.ExecuteReader();
        while (getir.Read())
        {
            bunifuCustomDataGrid2.Rows.Add();
            bunifuCustomDataGrid2.Rows[a].Cells[0].Value = getir[0];
            bunifuCustomDataGrid2.Rows[a].Cells[1].Value = double.Parse(getir[1].ToString()) + "₺";
            a++;
        }
        baglan.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

}

private void UrunEkle_Load(object sender, EventArgs e)
{
    DataGridYenile1();
    DataGridYenile2();
}

private void bunifuFlatButton1_Click(object sender, EventArgs e)
{
    try
    {
        if (TxtYiyecekAd.Text != "" && TxtYiyecekFiyat.Text != "")
        {
            SqlCommand sayi = new SqlCommand("select count(YiyecekAd) from Yiyecekler where
YiyecekAd=" + TxtYiyecekAd.Text + " and OtelId=" + otelid + "", baglan);
            baglan.Open();
            SqlDataReader oku = sayi.ExecuteReader();
            while (oku.Read())
            {
                if (int.Parse(oku[0].ToString()) == 0)
                {
                    string tutargir = TxtYiyecekFiyat.Text.ToString().Replace(",", ".");
                    SqlCommand yiyecekekkle = new SqlCommand("insert into Yiyecekler values(" + otelid
+ "," + TxtYiyecekAd.Text + "," + tutargir + ")", baglan2);
                    baglan2.Open();
                    yiyecekekkle.ExecuteNonQuery();
                    baglan2.Close();
                    MessageBox.Show("Yiyecek eklendi.", "BİLGİLENDİRME", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                }
                else
                    MessageBox.Show("Aynı isimde yiyecek mevcut.", "BİLGİLENDİRME",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            baglan.Close();
            DataGridYenile1();
        }
    }
}

```

```

        }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void bunifuFlatButton2_Click(object sender, EventArgs e)
{
    try
    {
        if (TxtIcecekAd.Text != "" && TxtIcecekFiyat.Text != "")
        {
            SqlCommand sayi = new SqlCommand("select count(IcecekAd) from Icecekler where
IcecekAd=" + TxtIcecekAd.Text + " and OtelId=" + otelid + "", baglan);
            baglan.Open();
            SqlDataReader oku = sayi.ExecuteReader();
            while (oku.Read())
            {
                if (int.Parse(oku[0].ToString()) == 0)
                {
                    string tutargir = TxtIcecekFiyat.Text.ToString().Replace(",",".");
                    SqlCommand icecekekle = new SqlCommand("insert into Icecekler values('" + otelid +
",'" + TxtIcecekAd.Text + "','" + tutargir + "')", baglan2);
                    baglan2.Open();
                    icecekekle.ExecuteNonQuery();
                    baglan2.Close();
                    MessageBox.Show("İçecek eklendi.", "BİLGİLENDİRME", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                }
            }
        }
        else
            MessageBox.Show("Aynı isimde yiyecek mevcut.", "BİLGİLENDİRME",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    baglan.Close();
    DataGridYenile2();
}

```

```

        }

    }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void UrunEkle_FormClosing(object sender, FormClosingEventArgs e)
{
    var yenilet = (OdaSiparis)Application.OpenForms["OdaSiparis"];
    if (yenilet != null)
        yenilet.ListleriYenile();
}

}
}

```

## KATHİZMETLERİ.CS

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using Bunifu.UI.WinForms.BunifuButton;

namespace Login
{
    public partial class KatHizmetleri : Form
    {"Server=DESKTOP-7JCKATF\MELIH;Integrated Security=SSPI;Initial Catalog=OtelProje"
        SqlConnection baglan = new SqlConnection(Login.veritabani);

```

```

int otelid = Login.id;

public KatHizmetleri()
{
    InitializeComponent();
}

private void OdaServisi_Load(object sender, EventArgs e)
{
    OdaDurumDegisimi();
    OdaOnaylanacaklar();
}

public void OdaOnaylanacaklar()
{
    try
    {
        Yenile2();
        baglan.Open();
        SqlCommand odalist = new SqlCommand("select OdaNo,Durum from Odalar where OtelId=" + 
otelid + " and Durum=6", baglan);

        SqlDataReader getir = odalist.ExecuteReader();
        while (getir.Read())
        {
            BunifuButton btndn = new BunifuButton();
            btndn.Size = new Size(135, 102);
            btndn.Margin = new System.Windows.Forms.Padding(5);
            btndn.IdleBorderThickness = 1;
            btndn.ButtonText = getir[0].ToString();
            btndn.Font = new Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.Bold);

            btndn.ForeColor = System.Drawing.Color.White;

            btndn.IdleBorderColor = Color.Gold;
            btndn.IdleFillColor = Color.Gold;

            flowLayoutPanel1.Controls.Add(btndn);
            btndn.onHoverState.BorderColor = btndn.IdleBorderColor;
    }
}

```

```

        btndn.onHoverState.FillColor = btndn.IdleFillColor;
        btndn.MouseEnter += belirtec;
        btndn.MouseLeave += ayirl;
        btndn.Click += Onay;
    }
    baglan.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void Onay(object sender, EventArgs e)
{
    try
    {
        BunifuButton btn = sender as BunifuButton;
        DialogResult cevap = MessageBox.Show("Odanın temizlendiğini / onarıldığını onaylıyor musunuz ?", "ONAY EKRANI", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (cevap == DialogResult.Yes)
        {
            baglan.Open();
            SqlCommand guncelle = new SqlCommand("update Odalar set Durum=2 where OdaNo='"
+ btn.ButtonText + "' and OtelId='"
+ otelid + "'", baglan);

            guncelle.ExecuteNonQuery();
            baglan.Close();
        }
        OdaOnaylanacaklar();
        OdaDurumDegisimi();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

}

public void TumOnay()
{
    try
    {
        if(flowLayoutPanel1.Controls.Count>0)
        {
            DialogResult cevap = MessageBox.Show("Tüm odaların temizlendiğini / onarıldığını onaylıyor musunuz?", "ONAY EKRANI", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (cevap == DialogResult.Yes)
            {
                baglan.Open();
                SqlCommand guncelle = new SqlCommand("update Odalar set Durum=2 where Durum=6 and OtelId='' + otelid + """", baglan);

                guncelle.ExecuteNonQuery();
                baglan.Close();
            }
            OdaOnaylanacaklar();
            OdaDurumDegisimi();
        }
        else
            MessageBox.Show("Onaylanacak oda mevcut değil.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception)
    {
    }
}

private void ayril(object sender, EventArgs e)
{
    BunifuButton btn = sender as BunifuButton;
    if(btn.IdleBorderThickness==3)

```

```
    btn.IdleBorderThickness = 1;
}

private void belirtec(object sender, EventArgs e)
{
    BunifuButton btn = sender as BunifuButton;
    btn.IdleBorderThickness = 4;
}

private void bunifuCheckbox5_OnChange(object sender, EventArgs e)
{
    OdaDurumDegisimi();
}

private void bunifuCheckbox1_OnChange(object sender, EventArgs e)
{
    OdaDurumDegisimi();
}

private void bunifuCheckbox2_OnChange(object sender, EventArgs e)
{
    OdaDurumDegisimi();
}

private void bunifuCheckbox3_OnChange(object sender, EventArgs e)
{
    OdaDurumDegisimi();
}

private void bunifuCheckbox4_OnChange(object sender, EventArgs e)
{
    OdaDurumDegisimi();
}

public void OdaDurumDegisimi()
{
    try
    {
        Yenile();
    }
}
```

```

String sqlsorgu = "";
if (bunifuCheckbox5.Checked == true)
    sqlsorgu = sqlsorgu + "Durum=2 or ";
if (bunifuCheckbox1.Checked == true)
    sqlsorgu = sqlsorgu + "Durum=1 or ";
if (bunifuCheckbox2.Checked == true)
    sqlsorgu = sqlsorgu + "Durum=3 or ";
if (bunifuCheckbox3.Checked == true)
    sqlsorgu = sqlsorgu + "Durum=4 or ";
if (bunifuCheckbox4.Checked == true)
    sqlsorgu = sqlsorgu + "Durum=5 or ";
sqlsorgu = sqlsorgu.Remove(sqlsorgu.Length - 4, 3);
baglan.Open();
SqlCommand odalist = new SqlCommand("select OdaNo,Durum from Odalar where OtelId=" + 
otelid + "" and (" + sqlsorgu + ") ", baglan);

SqlDataReader getir = odalist.ExecuteReader();
while (getir.Read())
{
    BunifuButton btndn = new BunifuButton();
    btndn.Size = new Size(135, 102);
    btndn.Margin = new System.Windows.Forms.Padding(5);
    btndn.IdleBorderThickness = 1;
    btndn.ButtonText = getir[0].ToString();
    btndn.Font = new Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.Bold);

    btndn.ForeColor = System.Drawing.Color.White;
    if (getir[1].ToString() == "1")
    {
        btndn.IdleBorderColor = System.Drawing.Color.FromArgb(51, 122, 183);
        btndn.IdleFillColor = System.Drawing.Color.FromArgb(51, 122, 183);
    }

    else if (getir[1].ToString() == "2")
    {
        btndn.IdleBorderColor = System.Drawing.Color.FromArgb(61, 214, 22);
        btndn.IdleFillColor = System.Drawing.Color.FromArgb(61, 214, 22);
    }
}

```

```

else if (getir[1].ToString() == "3")
{
    btndn.IdleBorderColor = Color.Red;
    btndn.IdleFillColor = Color.Red;
}

else if (getir[1].ToString() == "4")
{
    btndn.IdleBorderColor = Color.Gray;
    btndn.IdleFillColor = Color.Gray;
}

else
{
    btndn.IdleBorderColor = Color.Black;
    btndn.IdleFillColor = Color.Black;
}

flowLayoutPanel2.Controls.Add(btndn);
btndn.onHoverState.BorderColor = btndn.IdleBorderColor;
btndn.onHoverState.FillColor = btndn.IdleFillColor;
btndn.MouseEnter += belirtec;
btndn.MouseLeave += ayril;
btndn.Click += DurumPencere;
}

baglan.Close();
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void DurumPencere(object sender, EventArgs e)
{
    BunifuButton btn = sender as BunifuButton;
}

```

```

if (btn.IdleFillColor == System.Drawing.Color.FromArgb(51, 122, 183))
{
    MessageBox.Show("Dolu oda durumu değiştirilemez", "Uyarı", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
}
else
{
    Oda_Durum pencere = new Oda_Durum();
    pencere.Show();
    pencere.Controls[11].Text = btn.ButtonText;
}

private void Yenile()
{
    int temizle = flowLayoutPanel2.Controls.Count;
    for (int a = 0; a < temizle; a++)
    {
        flowLayoutPanel2.Controls.Remove(flowLayoutPanel2.Controls[0]);
    }
}

private void Yenile2()
{
    int temizle = flowLayoutPanel1.Controls.Count;
    for (int a = 0; a < temizle; a++)
    {
        flowLayoutPanel1.Controls.Remove(flowLayoutPanel1.Controls[0]);
    }
}
}

```

## **ODA\_DURUM.CS**

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{
    public partial class Oda_Durum : Form
    {
        SqlConnection baglan = new SqlConnection(Login.veritabani);
        int otelid = Login.id;

        public Oda_Durum()
        {
            InitializeComponent();
        }

        private void Oda_Durum_Load(object sender, EventArgs e)
        {
            try
            {
                SqlCommand odalist = new SqlCommand("select OdaNo from Odalar where OtelId=" + otelid +
                "", baglan);
                baglan.Open();
                SqlDataReader getir = odalist.ExecuteReader();
                while (getir.Read())
                {
                    comboBox1.Items.Add(getir[0]);
                }
                baglan.Close();
                radioButton1.Enabled = false;
                radioButton2.Enabled = false;
                comboBox2.Enabled = false;
            }
            catch (Exception ex)

```

```

{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        comboBox2.Enabled = true;
        SqlCommand odalist = new SqlCommand("select
Durum,Televizyon,Klima,Buzdolabi,MiniBar,Telefon from Odalar where OtelId=''" + otelid + "" and
OdaNo=''" + comboBox1.Text + "", baglan);
        baglan.Open();
        SqlDataReader getir = odalist.ExecuteReader();
        getir.Read();
        if (getir[0].ToString() == "2")
        {
            comboBox2.Text = "Temiz";
            radioButton1.Checked = false;
            radioButton2.Checked = false;
            PanelSifirla();
        }
        else if (getir[0].ToString() == "3")
        {
            comboBox2.SelectedItem = comboBox2.Items[0];
            radioButton1.Checked = false;
            radioButton2.Checked = false;
            PanelSifirla();
        }
        else if (getir[0].ToString() == "4")
        {
            comboBox2.SelectedItem = comboBox2.Items[1];
            radioButton1.Checked = true;
            radioButton2.Checked = false;
            panel1.Enabled = true;
            checkBox1.Checked = Convert.ToBoolean(getir[1]);
        }
    }
}

```

```

checkBox2.Checked = Convert.ToBoolean(getir[2]);
checkBox3.Checked = Convert.ToBoolean(getir[3]);
checkBox4.Checked = Convert.ToBoolean(getir[4]);
checkBox5.Checked = Convert.ToBoolean(getir[5]);
}
else if (getir[0].ToString() == "5")
{
    comboBox2.SelectedItem = comboBox2.Items[1];
    radioButton1.Checked = false;
    radioButton2.Checked = true;
    PanelSifirla();
}
baglan.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    if(radioButton2.Checked==true)
    {
        PanelSifirla();
    }
}

private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton1.Checked == true)
    {
        panel1.Enabled = true;
        checkBox1.Checked = false;
        checkBox2.Checked = false;
        checkBox3.Checked = false;
        checkBox4.Checked = false;
        checkBox5.Checked = false;
    }
}

```

```

        }
    }

private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    if(comboBox2.SelectedIndex==1)
    { radioButton1.Enabled = true;
        radioButton2.Enabled = true;
    }
    else
    { radioButton1.Enabled = false;radioButton1.Checked = false;
        radioButton2.Enabled = false;radioButton2.Checked = false;
        PanelSifirla();
    }
    if (comboBox2.SelectedIndex == 0 || comboBox2.SelectedIndex == 1)
    {
        bunifuFlatButton1.Visible = true;
    }
    else
        bunifuFlatButton1.Visible = false;
}
private void PanelSifirla()
{
    panel1.Enabled = false;
    checkBox1.Checked = false;
    checkBox2.Checked = false;
    checkBox3.Checked = false;
    checkBox4.Checked = false;
    checkBox5.Checked = false;
}

private void bunifuFlatButton13_Click(object sender, EventArgs e)
{
    if(comboBox1.Text=="")
        MessageBox.Show("Lütfen bir oda
seçiniz","UYARI",MessageBoxButtons.OK,MessageBoxIcon.Information);

    else {
        if (comboBox2.SelectedIndex == 0)

```

```

        Durum(3);
    else if (comboBox2.SelectedIndex == 1 && radioButton1.Checked == true)
        Durum(4);
    else if (comboBox2.SelectedIndex == 1 && radioButton2.Checked == true)
        Durum(5);
    else if (comboBox2.SelectedIndex == 1 && (radioButton1.Checked == false ||
radioButton2.Checked == false))
        MessageBox.Show("Lütfen arıza tipini seçiniz.", "UYARI", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }

    this.Hide();
    try
    {
        var pencere = (KatHizmetleri)Application.OpenForms["KatHizmetleri"];
        if (pencere != null)
        {
            pencere.OdaDurumDegisimi();
            pencere.OdaOnaylanacaklar();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void Durum(int a)
{
    try
    {
        SqlCommand guncelle = new SqlCommand("update Odalar set Durum=" + a + ",Televizyon=" +
+ checkBox1.Checked + "",Klima=" + checkBox2.Checked + "",Buzdolabi=" + checkBox3.Checked +
",MiniBar=" + checkBox4.Checked + "",Telefon=" + checkBox5.Checked + "" where OdaNo=" +
comboBox1.Text + " and OtelId=" + otelid + "", baglan);
        baglan.Open();
        guncelle.ExecuteNonQuery();
        baglan.Close();
    }
}

```

```

        }

        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

private void bunifuFlatButton12_Click(object sender, EventArgs e)
{
    this.Hide();
}

private void bunifuFlatButton1_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand guncelle = new SqlCommand("update Odalar set Durum=6,Televizyon=""" +
checkBox1.Checked + "",Klima=""" + checkBox2.Checked + "",Buzdolabi=""" + checkBox3.Checked +
""" ,MiniBar=""" + checkBox4.Checked + "",Telefon=""" + checkBox5.Checked + "" where OdaNo=""" +
comboBox1.Text + "" and OtelId=""" + otelid + """", baglan);

        baglan.Open();
        guncelle.ExecuteNonQuery();
        baglan.Close();
        this.Hide();
        var pencere = (KatHizmetleri)Application.OpenForms["KatHizmetleri"];
        if (pencere != null)
        {
            pencere.OdaDurumDegisimi();
            pencere.OdaOnaylanacaklar();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```
    }
}
}
```

## YÖNETİCİPANEL.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
```

```
namespace Login
{
    public partial class YoneticiPanel : Form
    {
        int otelid = Login.id;
        SqlConnection baglan = new SqlConnection(Login.veritabani);
        SqlCommand grafik;
```

```
public YoneticiPanel()
{
    InitializeComponent();
}
```

```
private void YoneticiPanel_Load(object sender, EventArgs e)
{
    Veriler();
}
```

```
private void Veriler()
{
```

```

chart1.ChartAreas[0].AxisX.IsReversed = true;
Tarih.Format = DateTimePickerFormat.Custom;
DateTime mevcutasil = Tarih.Value;

for (int i = 0; i < 7; i++)
{
    int gunsayi = 0;
    int misafirsayi = 0;
    grafik = new SqlCommand("SELECT DATEDIFF(day,GelisTarihi,GidisTarihi) from
GirisGecmis where OtelID=" + otelid + " and GelisTarihi <= " + Tarih.Text + " and GidisTarihi >= " +
Tarih.Text + "", baglan);
    baglan.Open();
    SqlDataReader oku = grafik.ExecuteReader();
    while (oku.Read())
    {
        gunsayi += 1;

    }
    if (i == 0)
    {
        Misafir.Text = gunsayi.ToString();
    }
    chart1.Series["Grafik"].Points.Add(int.Parse(gunsayi.ToString()));
    chart1.Series["Grafik"].Points[i].AxisLabel = Tarih.Text;
    baglan.Close();

}

DateTime oncekigun = Tarih.Value;
oncekigun = oncekigun.AddDays(-1);
Tarih.Value = oncekigun;
}

DateTime oncekigun2 = Tarih.Value;
oncekigun2 = oncekigun2.AddDays(7);
Tarih.Value = oncekigun2;

grafik = new SqlCommand("select COUNT(*) from Kullanicilar where OtelID=" + otelid + "", baglan);
baglan.Open();

```

```

SqlDataReader oku2 = grafik.ExecuteReader();
while (oku2.Read())
{
    Personel.Text = oku2[0].ToString();
}
baglan.Close();

double gelir = 0;
grafik = new SqlCommand("select Odeme from OdemelerGecmis where OtelID=" + otelid + "" and
OdemeTarih like "" + Tarih.Text + "%", baglan);
baglan.Open();
SqlDataReader oku3 = grafik.ExecuteReader();
while (oku3.Read())
{
    gelir += double.Parse(oku3[0].ToString());
}
baglan.Close();
Gelir.Text = gelir.ToString() + "₺";

int odasayisi = 0;
grafik = new SqlCommand("select Count(*) from Odalar where OtelID=" + otelid + "''", baglan);
baglan.Open();
SqlDataReader oku4 = grafik.ExecuteReader();
while (oku4.Read())
{
    odasayisi = int.Parse(oku4[0].ToString());
}
baglan.Close();

int doluoda = 0;
grafik = new SqlCommand("select Count(*) from Odalar where OtelID=" + otelid + "" and
Durum=1", baglan);
baglan.Open();
SqlDataReader oku5 = grafik.ExecuteReader();
while (oku5.Read())
{
    doluoda = int.Parse(oku5[0].ToString());
}

```

```

baglan.Close();

try
{
    int yuzde = doluoda * 100 / odasayisi;
    bunifuCircleProgress2.Value = yuzde;
}
catch (Exception)
{
    bunifuCircleProgress2.Value = 0;
}

Tarih.Value = mevcutasil;
Tarih.Format = DateTimePickerFormat.Long;

}
}

```

## **PERSONEL.CS**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

```

namespace Login

```

{
public partial class Personel : Form
{
    public Personel()
    {
        InitializeComponent();
    }

    int OtelID = Login.id;
    SqlConnection baglanti = new SqlConnection(Login.veritabani);
    SqlConnection baglanti2 = new SqlConnection(Login.veritabani);
    private void Personel_Load(object sender, EventArgs e)
    {
        try
        {
            yenile();
        }
        catch (Exception ex)
        {

            MessageBox.Show(ex.Message);
        }
    }

    void yenile()
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("select Ad as PersonelAd,Soyad as PersonelSoyad,KullaniciAdı as KullanıcıAdı from Kullanicilar where OtelID='" + OtelID + "'", baglanti);
        DataTable dt = new DataTable();
        komut.Fill(dt);
        personeller.DataSource = dt;
        personeller.Columns[0].FillWeight = 100;
        personeller.Columns[1].FillWeight = 100;
        personeller.Columns[2].FillWeight = 100;
        baglanti.Close();
    }

    private void PersonelEkle_Click(object sender, EventArgs e)
    {
}

```

```

if (pad.Text == "" || psoyad.Text == "" || ksifre.Text == "" || kad.Text == "")
    MessageBox.Show("Lütfen bütün bilgileri doldurun.", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
else
{
    try
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("insert into Kullanicilar values('" + OtelID + "','" +
pad.Text + "','" + psoyad.Text + "','" + kad.Text + "','" + ksifre.Text + "','" + false + "','" + false + "','" +
false + "','" + false + "','" + false + "','" + false + "','" + false + "')", baglanti);
        komut.ExecuteNonQuery();
        baglanti.Close();
        pad.Text = "";
        psoyad.Text = "";
        kad.Text = "";
        ksifre.Text = "";
        MessageBox.Show("Personel başarıyla eklenmiştir", "Bilgilendirme",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        yenile();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Personneller_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        int a = personneller.CurrentRow.Index;
        baglanti.Open();
        SqlCommand komut = new SqlCommand("select * from Kullanicilar where KullaniciAdi='" +
personneller.Rows[a].Cells[2].Value.ToString() + "' and OtelID=''" + OtelID + "'", baglanti);
        SqlDataReader dr = komut.ExecuteReader();
        while (dr.Read())

```

```

{
    rezsecim.Value = Convert.ToBoolean(dr[5].ToString());
    ressecim.Value = Convert.ToBoolean(dr[6].ToString());
    prosecim.Value = Convert.ToBoolean(dr[7].ToString());
    raporsecim.Value = Convert.ToBoolean(dr[8].ToString());
    odasecim.Value = Convert.ToBoolean(dr[9].ToString());
    katsecim.Value = Convert.ToBoolean(dr[10].ToString());
}
baglanti.Close();
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

}

private void Rezsecim_ValueChanged(object sender, EventArgs e)
{
    try
    {
        int a = personeller.CurrentRow.Index;
        baglanti2.Open();
        SqlCommand komut = new SqlCommand("update Kullanicilar set Rezervasyon=" +
Convert.ToBoolean((rezsecim.Value)) + " where Kullaniciadi=" + 
personeller.Rows[a].Cells[2].Value.ToString() + " and OtelID=" + OtelID + "", baglanti2);
        komut.ExecuteNonQuery();
        baglanti2.Close();
    }
    catch (Exception ex)
    {

        MessageBox.Show(ex.Message);
    }
}

private void Ressecim_ValueChanged(object sender, EventArgs e)

```

```

{
try
{
    int a = personeller.CurrentRow.Index;
    baglanti2.Open();
    SqlCommand komut = new SqlCommand("update Kullanicilar set Resepsyon='' +
Convert.ToBoolean((ressecim.Value)) + '' where Kullaniciadi='' +
personeller.Rows[a].Cells[2].Value.ToString() + '' and OtelID='' + OtelID + """", baglanti2);
    komut.ExecuteNonQuery();
    baglanti2.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

}

private void Prosecim_OnValuechange(object sender, EventArgs e)
{
try
{
    int a = personeller.CurrentRow.Index;
    baglanti2.Open();
    SqlCommand komut = new SqlCommand("update Kullanicilar set Profiller='' +
Convert.ToBoolean((prosecim.Value)) + '' where Kullaniciadi='' +
personeller.Rows[a].Cells[2].Value.ToString() + '' and OtelID='' + OtelID + """", baglanti2);
    komut.ExecuteNonQuery();
    baglanti2.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

}

```

```

private void Raporsecim_OnValuechange(object sender, EventArgs e)
{
    try
    {
        int a = personeller.CurrentRow.Index;
        baglanti2.Open();
        SqlCommand komut = new SqlCommand("update Kullanicilar set Raporlar="" +
Convert.ToBoolean((raporsecim.Value)) + "" where Kullaniciadi="" +
personeller.Rows[a].Cells[2].Value.ToString() + "" and OtelID="" + OtelID + """", baglanti2);
        komut.ExecuteNonQuery();
        baglanti2.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Odasecim_OnValuechange(object sender, EventArgs e)
{
    try
    {
        int a = personeller.CurrentRow.Index;
        baglanti2.Open();
        SqlCommand komut = new SqlCommand("update Kullanicilar set OdaServisi="" +
Convert.ToBoolean((odasecim.Value)) + "" where Kullaniciadi="" +
personeller.Rows[a].Cells[2].Value.ToString() + "" and OtelID="" + OtelID + """", baglanti2);
        komut.ExecuteNonQuery();
        baglanti2.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

private void Katsecim_OnValuechange(object sender, EventArgs e)
{
    try
    {
        int a = personeller.CurrentRow.Index;
        baglanti2.Open();
        SqlCommand komut = new SqlCommand("update Kullanicilar set KatHizmetleri='" +
Convert.ToBoolean((katsecim.Value)) + "' where Kullaniciadi='" +
personeller.Rows[a].Cells[2].Value.ToString() + "' and OtelID='" + OtelID + "'", baglanti2);
        komut.ExecuteNonQuery();
        baglanti2.Close();
    }
    catch (Exception ex)
    {

        MessageBox.Show(ex.Message);
    }
}

private void PersonelSil_Click(object sender, EventArgs e)
{
    try
    {
        DialogResult result = MessageBox.Show("Personeli silmek üzeresiniz ,onaylıyor musunuz?", "Bilgilendirme", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
        if (result == DialogResult.OK)
        {
            int a = personeller.CurrentRow.Index;
            baglanti2.Open();
            SqlCommand komut = new SqlCommand("delete from Kullanicilar where Kullaniciadi='" +
personeller.Rows[a].Cells[2].Value.ToString() + "' and OtelID='" + OtelID + "'", baglanti2);
            komut.ExecuteNonQuery();
            baglanti2.Close();
            MessageBox.Show("Personel başarıyla silinmiştir", "Bilgilendirme", MessageBoxButtons.OK, MessageBoxIcon.Information);
            yenile();
        }
    }
}

```

```
        }

        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

private void Pad_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}

void sayi.KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

void harf.KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);
}

private void Psoyad_KeyPress(object sender, KeyPressEventArgs e)
{
    harf(e);
}

private void Ksifre_ValueChanged(object sender, EventArgs e)
{
    if (ksifre.Text.Length == 8)
    {
        hata.Text = "Şifre 8 karakterden uzun olamaz";
    }
    else
        hata.Text = "";
}

}
```

## ODALAR.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Login
{
    public partial class Odalar : Form
    {
        public Odalar()
        {
            InitializeComponent();
        }

        int OtelID = Login.id;
        SqlConnection baglanti = new SqlConnection(Login.veritabani);
        SqlConnection baglanti2 = new SqlConnection(Login.veritabani);
        public int taslakonay = 1;
        public int Katsayisi;
        private void Katekle_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Eğer otel kurulumu(kat ve kattaki odaların içerikleri doldurulmadan) tamamlanmadan bu sayfadan çıkışlırsa mevcut veriler silinecektir ve tekrardan kurulumu yapmanız gerekecektir.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            try
            {
                baglanti.Open();
                SqlCommand komut = new SqlCommand("delete from OdalarTaslak where OtelID=" + OtelID + "'''", baglanti);
                komut.ExecuteNonQuery();
            }
        }
    }
}
```

```

        baglanti.Close();
    }
    catch (Exception)
    {
    }
    if(kat.Text.Length>0)
    {
        ykat.Items.Clear();
        ykat.Text = "";
        for (int i = 1; i <= int.Parse(kat.Text); i++)
        {
            ykat.Items.Add(i+".Kat");
        }
        Katsayisi = int.Parse(kat.Text);
    }
    else
        MessageBox.Show("Lütfen kat sayısını girin.", "Bilgilendirme", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
private void OdaEkle_Click(object sender, EventArgs e)
{
    if (kat.Text.Length > 0)
    {
        if (katoda.Text.Length > 0)
        {
            odasecim.Items.Clear();
            int oda = int.Parse(katoda.Text);
            for (int i = 1; i <= oda; i++)
            {
                if(ykat.SelectedItem!=null)
                {
                    ykat.Enabled = false;
                    katoda.Enabled = false;
                    odasecim.Items.Add(((int.Parse(ykat.SelectedItem.ToString().Substring(0, 1)) * 100) +
i).ToString(), false);
                }
            }
        }
    }
}

```

```

        MessageBox.Show("Lütfen oda sayısı girin.", "Bilgilendirme", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("Lütfen kat sayısı girin.", "Bilgilendirme", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }

int odatip;
void sayi_KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void Kat_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

private void Katoda_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

private void Fiyat_KeyPress(object sender, KeyPressEventArgs e)
{
    sayi(e);
}

private void Odalar_Load(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut5 = new SqlCommand("select Count(*) From Odalar where
OtelID='"+OtelID+"'",baglanti);
    SqlDataReader dr1 = komut5.ExecuteReader();
    while (dr1.Read())
    {
        if (int.Parse(dr1[0].ToString()) == 0)
        {
        }
        else
        {
            guncellekat.Enabled = true;
            guncelleoda.Enabled = true;
        }
    }
}

```

```

guncelletr.Enabled = true;
guncelletsu.Enabled = true;
guncelletwr.Enabled = true;
guncellesgl.Enabled = true;
guncellequad.Enabled = true;
guncelledbl.Enabled = true;
guncellefiyat.Enabled = true;
guncelleodasecim.Enabled = true;
Güncelle.Enabled = true;
kat.Enabled = false;
Katekle.Enabled = false;
ykat.Enabled = false;
katoda.Enabled = false;
OdaEkle.Enabled = false;
tr.Enabled = false;
tsu.Enabled = false;
twr.Enabled = false;
sgl.Enabled = false;
quad.Enabled = false;
dbl.Enabled = false;
fiyat.Enabled = false;
odasecim.Enabled = false;
Ekle.Enabled = false;
Kaydet.Enabled = false;
kat.Enabled = false;
Katekle.Enabled = false;
}

}

baglanti.Close();
guncelletr.Checked = false;
guncelletsu.Checked = false;
guncelletwr.Checked = false;
guncellesgl.Checked = false;
guncellequad.Checked = false;
guncelledbl.Checked = false;
tr.Checked = false;
tsu.Checked = false;
twr.Checked = false;
sgl.Checked = false;

```

```

quad.Checked = false;
dbl.Checked = false;
baglanti.Open();
SqlCommand komut = new SqlCommand("select KatSayisi from Oteller where
OtelID="""+OtelID+"",baglanti);
SqlDataReader dr = komut.ExecuteReader();
while (dr.Read())
{
    for (int i = 1; i <= int.Parse(dr[0].ToString()); i++)
    {
        guncellekat.Items.Add(i+".Kat");
    }
}
baglanti.Close();
}

private void Guncellekat_SelectedIndexChanged(object sender, EventArgs e)
{
    guncelleodasecim.Items.Clear();
    guncelleoda.Text = "";
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select Count(OdaNo) from Odalar where
Kat="""+guncellekat.SelectedItem.ToString().Substring(0,1)+ "" and OtelID=""" + OtelID + """", baglanti);
    SqlDataReader dr = komut.ExecuteReader();
    while (dr.Read())
    {
        guncelleoda.Text = dr[0].ToString();
    }
    baglanti.Close();
    baglanti2.Open();
    SqlCommand komut2 = new SqlCommand("select OdaNo from Odalar where Kat=""" +
guncellekat.SelectedItem.ToString().Substring(0, 1) + "" and OtelID="""+OtelID+"", baglanti2);
    SqlDataReader dr2 = komut2.ExecuteReader();
    while (dr2.Read())
    {
        int b = int.Parse(guncelleoda.Text);
        b= int.Parse(dr2[0].ToString());
        guncelleodasecim.Items.Add(dr2[0].ToString(),false);
    }
    baglanti2.Close();
}

```

```

}

int odatip2;
private void Guncelleodasecim_SelectedIndexChanged(object sender, EventArgs e)
{
    if (guncellesgl.Checked == true)
        odatip2 = 1;
    else if (guncelletsu.Checked == true)
        odatip2 = 2;
    else if (guncelletwr.Checked == true)
        odatip2 = 3;
    else if (guncelledbl.Checked == true)
        odatip2 = 4;
    else if (guncelletr.Checked == true)
        odatip2 = 5;
    else if (guncellequad.Checked == true)
        odatip2 = 6;
    if (guncelleodasecim.CheckedItems.Count == 0)
    {
        guncellesgl.Checked = false;
        guncelletr.Checked = false;
        guncelletsu.Checked = false;
        guncelletwr.Checked = false;
        guncellequad.Checked = false;
        guncelledbl.Checked = false;
        guncellefiyat.Text = "";
        baglanti.Open();
        SqlCommand komut = new SqlCommand("select OdaTip,Fiyat from Odalar where OdaNo=" +
guncelleodasecim.SelectedItem + " and OtelID=" + OtelID + "", baglanti);
        SqlDataReader dr = komut.ExecuteReader();
        while (dr.Read())
        {
            if (Convert.ToInt32(dr[0].ToString()) == 1)
            {
                guncellesgl.Checked = true;
                guncelletr.Checked = false;
                guncelletsu.Checked = false;
                guncelletwr.Checked = false;
                guncellequad.Checked = false;
                guncelledbl.Checked = false;
            }
        }
    }
}

```

```

        }

else if (Convert.ToInt32(dr[0].ToString()) == 2)
{
    guncelletsu.Checked = true;
    guncelletr.Checked = false;
    guncelletwr.Checked = false;
    guncellesgl.Checked = false;
    guncellequad.Checked = false;
    guncelledbl.Checked = false;
}

else if (Convert.ToInt32(dr[0].ToString()) == 3)
{
    guncelletwr.Checked = true;
    guncelletr.Checked = false;
    guncelletsu.Checked = false;
    guncellesgl.Checked = false;
    guncellequad.Checked = false;
    guncelledbl.Checked = false;
}

else if (Convert.ToInt32(dr[0].ToString()) == 4)
{
    guncelledbl.Checked = true;
    guncelletr.Checked = false;
    guncelletsu.Checked = false;
    guncelletwr.Checked = false;
    guncellesgl.Checked = false;
    guncellequad.Checked = false;
}

else if (Convert.ToInt32(dr[0].ToString()) == 5)
{
    guncelletr.Checked = true;
    guncelletsu.Checked = false;
    guncelletwr.Checked = false;
    guncellesgl.Checked = false;
    guncellequad.Checked = false;
    guncelledbl.Checked = false;
}

else if (Convert.ToInt32(dr[0].ToString()) == 6)
{

```

```

        guncellequad.Checked = true;
        guncelletr.Checked = false;
        guncelletsu.Checked = false;
        guncelletwr.Checked = false;
        guncellesgl.Checked = false;
        guncelledbl.Checked = false;
    }
    guncellefiyat.Text = dr[1].ToString();

}

baglanti.Close();
}

else
{
    guncellequad.Checked = false;
    guncelletr.Checked = false;
    guncelletsu.Checked = false;
    guncelletwr.Checked = false;
    guncellesgl.Checked = false;
    guncelledbl.Checked = false;
    guncellefiyat.Text = "";
}

}

int odatip3;
private void Güncelle_Click(object sender, EventArgs e)
{
    if (guncellesgl.Checked == false && guncelletsu.Checked == false && guncelletwr.Checked == false && guncelledbl.Checked == false && guncelletr.Checked == false && guncellequad.Checked == false)
        MessageBox.Show("Lütfen oda tipini seçiniz.", "Bilgilendirme", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    else if (guncellefiyat.Text == "")
        MessageBox.Show("Lütfen oda fiyatını giriniz.", "Bilgilendirme", MessageBoxButtons.OK,
        MessageBoxIcon.Information);      else
    {
        if (guncellesgl.Checked == true)
            odatip3 = 1;
        else if (guncelletsu.Checked == true)

```

```

odatip3 = 2;
else if (guncelletwr.Checked == true)
    odatip3 = 3;
else if (guncelledbl.Checked == true)
    odatip3 = 4;
else if (guncelletr.Checked == true)
    odatip3 = 5;
else if (guncellequad.Checked == true)
    odatip3 = 6;
for (int a = 0; a < guncelleodasecim.CheckedItems.Count; a++)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("update Odalar set OdaTip='' + odatip3 + '' ,
Fiyat='' + Convert.ToDouble(guncellefiyat.Text) + '' where OdaNo='' +
guncelleodasecim.CheckedItems[a] + '' and OtelID='' + OtelID + """", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
}
if(guncelleodasecim.SelectedItems.Count==1)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("update Odalar set OdaTip='' + odatip3 + '' ,
Fiyat='' + Convert.ToDouble(guncellefiyat.Text) + '' where OdaNo='' + guncelleodasecim.SelectedItem +
"" and OtelID='' + OtelID + """", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
}
string degisenodalar = "";
if (guncelleodasecim.CheckedItems.Count > 0)
{
    for (int i = 0; i <= guncelleodasecim.CheckedItems.Count; i++)
    {
        degisenodalar = degisenodalar + " " + guncelleodasecim.CheckedItems[0].ToString();
        guncelleodasecim.SelectedItem = guncelleodasecim.CheckedItems[0];
        guncelleodasecim.SetItemChecked(guncelleodasecim.SelectedIndex, false);
    }
    degisenodalar = degisenodalar + " " + guncelleodasecim.CheckedItems[0].ToString();
    guncelleodasecim.SelectedItem = guncelleodasecim.CheckedItems[0];
    guncelleodasecim.SetItemChecked(guncelleodasecim.SelectedIndex, false);
}

```

```

        degisenodalar = degisenodalar + " Nolu odalar başarıyla güncellendi.";
    }
    else
        degisenodalar = guncelleodasesecim.SelectedItem.ToString() + " Nolu oda başarıyla
        güncellendi.";

    MessageBox.Show(degisenodalar, "Bilgilendirme", MessageBoxButtons.OK,
    MessageBoxIcon.Information);

    if (guncelleodasesecim.Items.Count == 0)
    {
        guncelleoda.Text = "";
    }
    guncelletsu.Checked = false;
    guncelletr.Checked = false;
    guncelletwr.Checked = false;
    guncellesgl.Checked = false;
    guncellequad.Checked = false;
    guncelledbl.Checked = false;
    guncellefiyat.Text = "";
}

private void Kaydet_Click(object sender, EventArgs e)
{
    if (odasesecim.CheckedItems.Count == 0)
        MessageBox.Show("Lütfen oda seçiniz.", "Bilgilendirme", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    else if (sgl.Checked == false && tsu.Checked == false && twr.Checked == false && dbl.Checked
    == false && tr.Checked == false && quad.Checked == false)
        MessageBox.Show("Lütfen oda tipini seçiniz.", "Bilgilendirme", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    else if (fiyat.Text == "")
        MessageBox.Show("Lütfen oda fiyatını giriniz.", "Bilgilendirme", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    else
    {
        if (sgl.Checked == true)
            odatip = 1;
        else if (tsu.Checked == true)
            odatip = 2;
        else if (twr.Checked == true)

```

```

odatip = 3;
else if (dbl.Checked == true)
    odatip = 4;
else if (tr.Checked == true)
    odatip = 5;
else if (quad.Checked == true)
    odatip = 6;
for (int i = 0; i < odasecim.CheckedItems.Count; i++)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("insert into OdalarTaslak values('" + OtelID + "','" +
int.Parse(ykat.SelectedItem.ToString().Substring(0, 1)) + "','" + int.Parse(Convert.ToString(OtelID +
odasecim.CheckedItems[i].ToString())) + "','" + int.Parse(odasecim.CheckedItems[i].ToString()) + "','" +
odatip + "','" + 2 + "','" + 1 + "','" + double.Parse(fiyat.Text) + "','" + true + "','" + true + "','" +
true + "','" + true + "')", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
}
MessageBox.Show("Odalar eklendi");
taslakonay++;
tr.Checked = false;
tsu.Checked = false;
twr.Checked = false;
sgl.Checked = false;
quad.Checked = false;
dbl.Checked = false;
fiyat.Text = "";
for (int i = 0; i < odasecim.CheckedItems.Count * 100; i++)
{
    odasecim.Items.Remove(odasecim.CheckedItems[0]);
}
if (odasecim.Items.Count == 0)
{
    ykat.Items.RemoveAt(ykat.SelectedIndex);
    katoda.Text = "";
    ykat.Enabled = true;
    katoda.Enabled = true;
}
if (ykat.Items.Count == 0)

```

```

        kat.Text = "";
        if (odasecim.Items.Count == 0 && ykat.Items.Count == 0 && taslakonay!=1)
        {
            MessageBox.Show("Otel kurulum taslağı tamamlandı. İşlenen mevcut odaları eklemek için 'Ekle' butonuna basın.", "Bilgilendirme", MessageBoxButtons.OK, MessageBoxIcon.Information);
            ykat.Enabled = false;
            katoda.Enabled = false;
            OdaEkle.Enabled = false;
            tr.Enabled = false;
            tsu.Enabled = false;
            twr.Enabled = false;
            sgl.Enabled = false;
            quad.Enabled = false;
            dbl.Enabled = false;
            fiyat.Enabled = false;
            odasecim.Enabled = false;
            Ekle.Enabled = true;
            Kaydet.Enabled = false;
            Katekle.Enabled = false;
            kat.Enabled = false;
        }
    }
}

private void Ekle_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select * from OdalarTaslak where OtelID='"++OteliD+"'",baglanti);
    SqlDataReader dr = komut.ExecuteReader();
    while (dr.Read())
    {
        baglanti2.Open();
        SqlCommand komut2 = new SqlCommand("insert into Odalar values('"+dr[0]+ "','" + dr[1] + "','" + dr[2] + "','" + dr[3] + "','" + dr[4] + "','" + dr[5] + "','" + dr[6] + "','" + double.Parse(dr[7].ToString()) + "','" + dr[8] + "','" + dr[9] + "','" + dr[10] + "','" + dr[11] + "','" + dr[12] + "')", baglanti2);
        komut2.ExecuteNonQuery();
        baglanti2.Close();
        baglanti2.Open();
    }
}

```

```

SqlCommand komut5 = new SqlCommand("insert into OdalarYenileme values('" + OtelID + "','" +
+ dr[3] + "','" + dr[5] + "')", baglanti2);
komut5.ExecuteNonQuery();
baglanti2.Close();
}
baglanti.Close();
baglanti.Open();
SqlCommand komut3 = new SqlCommand("delete from OdalarTaslak where OtelID='" + OtelID +
"''", baglanti);
komut3.ExecuteNonQuery();
baglanti.Close();
baglanti.Open();
SqlCommand komut4 = new SqlCommand("update Oteller set KatSayisi='"+Katsayisi+"' where
OtelID='"+OtelID+"'",baglanti);
komut4.ExecuteNonQuery();
baglanti.Close();
MessageBox.Show("Otel kurulumu tamamlandı.", "Bilgilendirme", MessageBoxButtons.OK,
MessageBoxIcon.Information);
Ekle.Enabled = false;
}
}
}

```

## İSTATİSTİKLER.CS

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace Login
{
    public partial class Istatistikler : Form
    {

```

```

int otelid = Login.id;
SqlConnection baglan = new SqlConnection(Login.veritabani);
int sorgutipi = 0;
SqlCommand grafik;
Boolean datepickerchange = true;

public Istatistikler()
{
    InitializeComponent();
}

private void RadioGun_Click(object sender, EventArgs e)
{
    DateMevcut.CustomFormat = "yyyy-MM-dd";
    DateKarsi.CustomFormat = "yyyy-MM-dd";
    Grafik();
    IstatistikYazi();
}

private void RadioAy_Click(object sender, EventArgs e)
{
    DateMevcut.CustomFormat = "yyyy-MM";
    DateKarsi.CustomFormat = "yyyy-MM";
    Grafik();
    IstatistikYazi();
}

private void RadioYil_Click(object sender, EventArgs e)
{
    DateMevcut.CustomFormat = "yyyy";
    DateKarsi.CustomFormat = "yyyy";
    Grafik();
    IstatistikYazi();
}

private void Grafik()
{
    try
    {
        datepickerchange = false;
        DateTime mevcutasil = DateMevcut.Value;
        DateTime karsiasil = DateKarsi.Value;
        chart1.Series["Grafik"].IsValueShownAsLabel = true;
    }
}

```

```

chart1.Series["Grafik"].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Column;
chart1.Series["Grafik"].BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.TopBottom;
foreach (var series in chart1.Series)
{
    series.Points.Clear();
}
chart1.Titles.Clear();
if (bunifuToggleSwitch1.Value == true)
{
    chart1.ChartAreas[0].AxisX.IsReversed = false;
}
else
{
    chart1.ChartAreas[0].AxisX.IsReversed = true; //chartı ters çevirmek için buradan yardım
aldım : https://stackoverflow.com/questions/29932431/reverse-the-direction-of-line-chart
}
if (sorgutipi == 1)
{
    chart1.Titles.Add("Tarih Biçimine Göre Ortalama Dolu Oda Sayısı");
    if (RadioAy.Checked == true)
    {
        for (int i = 0; i < 3; i++)
        {
            int gunsayi = 0;
            grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID=" + otelid + " and GelisTarihi
like '" + DateMevcut.Text + "%' and GidisTarihi like '" + DateMevcut.Text + "%'", baglan);
            baglan.Open();
            SqlDataReader oku = grafik.ExecuteReader();
            while (oku.Read())
            {
                gunsayi += int.Parse(oku[3].ToString());
            }
            int aydakigun = DateTime.DaysInMonth(DateMevcut.Value.Year,
DateMevcut.Value.Month);
            double a = gunsayi / aydakigun;
            baglan.Close();
        }
    }
}

```

```

chart1.Series["Grafik"].Points.Add(a);
chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
if (bunifuToggleSwitch1.Value == true)
{
    int gunsayi2 = 0;
    grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID=" + otelid + "" and GelisTarihi
like "" + DateKarsi.Text + '%' and GidisTarihi like "" + DateKarsi.Text + "%'", baglan);
    baglan.Open();
    SqlDataReader oku2 = grafik.ExecuteReader();
    while (oku2.Read())
    {
        gunsayi2 += int.Parse(oku2[3].ToString());
    }
    int aydakigun2 = DateTime.DaysInMonth(DateKarsi.Value.Year,
DateKarsi.Value.Month);
    double a2 = gunsayi2 / aydakigun2;
    baglan.Close();
    chart1.Series["Grafik"].Points.Add(a2);
    chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
    break;
}
else
{
    DateTime oncekigun = DateMevcut.Value;
    oncekigun = oncekigun.AddMonths(-1); //tarih işlemleri için buradan yardım aldım :
http://web.bilecik.edu.tr/ugur-talas/2014/10/14/cta-datetimea-zaman-ekleme-yada-cikarma-islemeleri/
    DateMevcut.Value = oncekigun;
}
}
}
else if (RadioYil.Checked == true)
{
    for (int i = 0; i < 3; i++)
    {
        int gunsayi = 0;
        grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID=" + otelid + "" and GelisTarihi
like "" + DateMevcut.Text + '%' and GidisTarihi like "" + DateMevcut.Text + "%'", baglan);
    }
}

```

```

baglan.Open();
SqlDataReader oku = grafik.ExecuteReader();
while (oku.Read())
{
    gunsayi += int.Parse(oku[3].ToString());
}
int a = gunsayi / 365;
baglan.Close();
chart1.Series["Grafik"].Points.Add(a);
chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
if (bunifuToggleSwitch1.Value == true)
{
    int gunsayi2 = 0;
    grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID='" + otelid + "' and GelisTarihi
like '" + DateKarsi.Text + "%' and GidisTarihi like '" + DateKarsi.Text + "%'", baglan);
    baglan.Open();
    SqlDataReader oku2 = grafik.ExecuteReader();
    while (oku2.Read())
    {
        gunsayi2 += int.Parse(oku2[3].ToString());
    }
    int a2 = gunsayi2 / 365;
    baglan.Close();
    chart1.Series["Grafik"].Points.Add(a2);
    chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
    break;
}
else
{
    DateTime oncekigun = DateMevcut.Value;
    oncekigun = oncekigun.AddYears(-1);
    DateMevcut.Value = oncekigun;
}
}
}
else
{
for (int i = 0; i < 7; i++)

```

```

{
    int gunsayi = 0;
    grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID=''' + otelid + ''' and GelisTarihi
<= ''' + DateMevcut.Text + ''' and GidisTarihi >= ''' + DateMevcut.Text + """', baglan);
    baglan.Open();
    SqlDataReader oku = grafik.ExecuteReader();
    while (oku.Read())
    {
        gunsayi += 1;
    }
    baglan.Close();
    chart1.Series["Grafik"].Points.Add(gunsayi);
    chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
    if (bunifuToggleSwitch1.Value == true)
    {
        int gunsayi2 = 0;
        grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID=''' + otelid + ''' and GelisTarihi
<= ''' + DateKarsi.Text + ''' and GidisTarihi >= ''' + DateKarsi.Text + """', baglan);
        baglan.Open();
        SqlDataReader oku2 = grafik.ExecuteReader();
        while (oku2.Read())
        {
            gunsayi2 += 1;
        }
        baglan.Close();
        chart1.Series["Grafik"].Points.Add(gunsayi2);
        chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
        break;
    }
    else
    {
        DateTime oncekigun = DateMevcut.Value;
        oncekigun = oncekigun.AddDays(-1);
        DateMevcut.Value = oncekigun;
    }
}
}

```

```

    }

else if (sorgutipi == 2)
{
    chart1.Titles.Add("Tarih Biçimine Göre Yapılan Rezervasyon Sayısı");
    if (RadioAy.Checked == true)
    {
        for (int i = 0; i < 3; i++)
        {
            grafik = new SqlCommand("select COUNT(*) from RezRapor where OtelID=" + otelid
+ " and GelisTarih like '" + DateMevcut.Text + "%'", baglan);
            baglan.Open();
            SqlDataReader oku = grafik.ExecuteReader();
            while (oku.Read())
            {
                chart1.Series["Grafik"].Points.Add(int.Parse(oku[0].ToString()));
                chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
            }
            baglan.Close();
            if (bunifurtoggleSwitch1.Value == true)
            {
                grafik = new SqlCommand("select COUNT(*) from RezRapor where OtelID=" + otelid
+ " and GelisTarih like '" + DateKarsi.Text + "%'", baglan);
                baglan.Open();
                SqlDataReader oku2 = grafik.ExecuteReader();
                while (oku2.Read())
                {
                    chart1.Series["Grafik"].Points.Add(int.Parse(oku2[0].ToString()));
                    chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
                    chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
                }
                baglan.Close();
                break;
            }
        }
    }
}
else
{
    DateTime oncekigun = DateMevcut.Value;
}

```

```

oncekigun = oncekigun.AddMonths(-1); //tarih işlemleri için buradan yardım aldım :
http://web.bilecik.edu.tr/ugur-talas/2014/10/14/cta-datetimea-zaman-ekleme-yada-cikarma-islemeleri/
DateMevcut.Value = oncekigun;
}
}
}
else if (RadioYil.Checked == true)
{
for (int i = 0; i < 3; i++)
{
grafik = new SqlCommand("select COUNT(*) from RezRapor where OtelID=" + otelid
+ " and GelisTarih like '" + DateMevcut.Text + "%'", baglan);
baglan.Open();
SqlDataReader oku = grafik.ExecuteReader();
while (oku.Read())
{
chart1.Series["Grafik"].Points.Add(int.Parse(oku[0].ToString()));
chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
}
baglan.Close();
if (bunifuToggleSwitch1.Value == true)
{
grafik = new SqlCommand("select COUNT(*) from RezRapor where OtelID=" + 
otelid + " and GelisTarih like '" + DateKarsi.Text + "%'", baglan);
baglan.Open();
SqlDataReader oku2 = grafik.ExecuteReader();
while (oku2.Read())
{
chart1.Series["Grafik"].Points.Add(int.Parse(oku2[0].ToString()));
chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
}
baglan.Close();
break;
}
else
{
DateTime oncekigun = DateMevcut.Value;
oncekigun = oncekigun.AddYears(-1);
DateMevcut.Value = oncekigun;
}
}
}

```

```

        }
    }
}
else
{
    for (int i = 0; i < 7; i++)
    {
        grafik = new SqlCommand("select COUNT(*) from RezRapor where OtelID='" + otelid
+ "' and GelisTarih like '" + DateMevcut.Text + "%'", baglan);
        baglan.Open();
        SqlDataReader oku = grafik.ExecuteReader();
        while (oku.Read())
        {
            chart1.Series["Grafik"].Points.Add(int.Parse(oku[0].ToString()));
            chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
        }
        baglan.Close();
        if (bunifuToggleSwitch1.Value == true)
        {
            grafik = new SqlCommand("select COUNT(*) from RezRapor where OtelID='" +
otelid + "' and GelisTarih like '" + DateKarsi.Text + "%'", baglan);
            baglan.Open();
            SqlDataReader oku2 = grafik.ExecuteReader();
            while (oku2.Read())
            {
                chart1.Series["Grafik"].Points.Add(int.Parse(oku2[0].ToString()));
                chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
            }
            baglan.Close();
            break;
        }
    else
    {
        DateTime oncekigun = DateMevcut.Value;
        oncekigun = oncekigun.AddDays(-1);
        DateMevcut.Value = oncekigun;
    }
}
}

```

```

        }

        if (sorgutipi == 3)
        {
            chart1.Titles.Add("Tarihlere Göre Toplam Ciro (₺)");
            int gecmisure = 0;
            if (RadioGun.Checked == true)
                gecmisure = 7;
            else if (RadioAy.Checked == true)
                gecmisure = 3;
            else
                gecmisure = 3;
            for (int i = 0; i < gecmisure; i++)
            {
                double asilciro = 0;
                double karsiciro = 0;
                grafik = new SqlCommand("select Odeme from OdemelerGecmis where OtelID=" + otelid
                + " and OdemeTarih like '" + DateMevcut.Text + "%'", baglan);
                baglan.Open();
                SqlDataReader oku = grafik.ExecuteReader();
                while (oku.Read())
                {
                    asilciro += double.Parse(oku[0].ToString());
                }
                baglan.Close();
                chart1.Series["Grafik"].Points.Add(asilciro);
                chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
                if (bunifuToggleSwitch1.Value == true)
                {
                    grafik = new SqlCommand("select Odeme from OdemelerGecmis where OtelID=" + otelid
                    + " and OdemeTarih like '" + DateKarsi.Text + "%'", baglan);
                    baglan.Open();
                    SqlDataReader oku2 = grafik.ExecuteReader();
                    while (oku2.Read())
                    {
                        karsiciro += double.Parse(oku2[0].ToString());
                    }
                    baglan.Close();
                    chart1.Series["Grafik"].Points.Add(karsiciro);
                    chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
                }
            }
        }
    }
}

```

```

        break;
    }
    else
    {
        if (RadioGun.Checked == true)
        {
            DateTime oncekigun = DateMevcut.Value;
            oncekigun = oncekigun.AddDays(-1);
            DateMevcut.Value = oncekigun;
        }
        else if (RadioAy.Checked == true)
        {
            DateTime oncekigun = DateMevcut.Value;
            oncekigun = oncekigun.AddMonths(-1);
            DateMevcut.Value = oncekigun;
        }
        else
        {
            DateTime oncekigun = DateMevcut.Value;
            oncekigun = oncekigun.AddYears(-1);
            DateMevcut.Value = oncekigun;
        }
    }
}

if (sorgutipi == 4)
{
    chart1.Titles.Add("Tarihlere Göre Misafirlerin Uyruk Oranı");
    chart1.Series["Grafik"].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Pie;
    chart1.Series["Grafik"].BackGradientStyle =
System.Windows.Forms.DataVisualization.Charting.GradientStyle.None;
    grafik = new SqlCommand("select Count(*) from RezRapor where OtelID='" + otelid + "' and
GelisTarih like '" + DateMevcut.Text + "%' and uyruk=1", baglan);
    baglan.Open();
    SqlDataReader oku = grafik.ExecuteReader();
    while (oku.Read())
    {
        chart1.Series["Grafik"].Points.AddXY("Türk", int.Parse(oku[0].ToString()));
    }
}

```

```

        }

        baglan.Close();

        grafik = new SqlCommand("select Count(*) from RezRapor where OtelID='' + otelid + '' and
GelisTarih like '' + DateMevcut.Text + '%' and uyruk=0", baglan);

        baglan.Open();

        SqlDataReader oku2 = grafik.ExecuteReader();

        while (oku2.Read())
        {
            chart1.Series["Grafik"].Points.AddXY("Yabancı", int.Parse(oku2[0].ToString()));

        }

        baglan.Close();

        if (bunifuToggleSwitch1.Value == true)
        {
            grafik = new SqlCommand("select Count(*) from RezRapor where OtelID='' + otelid + '' and GelisTarih like '' + DateKarsi.Text + '%' and uyruk=1", baglan);

            baglan.Open();

            SqlDataReader oku3 = grafik.ExecuteReader();

            while (oku3.Read())
            {
                chart1.Series["Grafik"].Points.AddXY("Karşılaştırılan Türk",
int.Parse(oku3[0].ToString()));

            }

            baglan.Close();

            grafik = new SqlCommand("select Count(*) from RezRapor where OtelID='' + otelid + '' and GelisTarih like '' + DateKarsi.Text + '%' and uyruk=0", baglan);

            baglan.Open();

            SqlDataReader oku4 = grafik.ExecuteReader();

            while (oku4.Read())
            {
                chart1.Series["Grafik"].Points.AddXY("Karşılaştırılan Yabancı",
int.Parse(oku4[0].ToString()));

            }

            baglan.Close();
        }
    }
}
if (sorgutipi == 5)
{

```

```

chart1.Titles.Add("Tarihlere Göre Toplam Konaklama Geliri (₺)");
int gecmisure = 0;
if (RadioGun.Checked == true)
    gecmisure = 7;
else if (RadioAy.Checked == true)
    gecmisure = 3;
else
    gecmisure = 3;
for (int i = 0; i < gecmisure; i++)
{
    double asilkonak = 0;
    double karsikonak = 0;
    grafik = new SqlCommand("select ToplamOdaFiyati from CikisGecmis where OtelID=" +
otelid + " and Tarih like '" + DateMevcut.Text + "%'", baglan);
    baglan.Open();
    SqlDataReader oku = grafik.ExecuteReader();
    while (oku.Read())
    {
        asilkonak += double.Parse(oku[0].ToString());
    }
    baglan.Close();
    chart1.Series["Grafik"].Points.Add(asilkonak);
    chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
    if (bunifuToggleSwitch1.Value == true)
    {
        grafik = new SqlCommand("select ToplamOdaFiyati from CikisGecmis where OtelID=" +
+ otelid + " and Tarih like '" + DateKarsi.Text + "%'", baglan);
        baglan.Open();
        SqlDataReader oku2 = grafik.ExecuteReader();
        while (oku2.Read())
        {
            karsikonak += double.Parse(oku2[0].ToString());
        }
        baglan.Close();
        chart1.Series["Grafik"].Points.Add(karsikonak);
        chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
        break;
    }
}

```

```

else
{
    if (RadioGun.Checked == true)
    {
        DateTime oncekigun = DateMevcut.Value;
        oncekigun = oncekigun.AddDays(-1);
        DateMevcut.Value = oncekigun;
    }
    else if (RadioAy.Checked == true)
    {
        DateTime oncekigun = DateMevcut.Value;
        oncekigun = oncekigun.AddMonths(-1);
        DateMevcut.Value = oncekigun;
    }
    else
    {
        DateTime oncekigun = DateMevcut.Value;
        oncekigun = oncekigun.AddYears(-1);
        DateMevcut.Value = oncekigun;
    }
}
}

if (sorgutipi == 6)
{
    chart1.Titles.Add("Tarihlere Göre Toplam Ürün Geliri (₺)");
    int gecmisure = 0;
    if (RadioGun.Checked == true)
        gecmisure = 7;
    else if (RadioAy.Checked == true)
        gecmisure = 3;
    else
        gecmisure = 3;
    for (int i = 0; i < gecmisure; i++)
    {
        double asilurun = 0;
        double karsiurun = 0;
        grafik = new SqlCommand("select EkstraHarcamalar from CikisGecmis where OtelID=" + otelid + " and Tarih like '" + DateMevcut.Text + "%'", baglan);

```

```

baglan.Open();
SqlDataReader oku = grafik.ExecuteReader();
while (oku.Read())
{
    asilurun += double.Parse(oku[0].ToString());
}
baglan.Close();
chart1.Series["Grafik"].Points.Add(asilurun);
chart1.Series["Grafik"].Points[i].AxisLabel = DateMevcut.Text;
if (bunifuToggleSwitch1.Value == true)
{
    grafik = new SqlCommand("select EkstraHarcamalar from CikisGecmis where OtelID="""
+ otelid + "" and Tarih like "" + DateKarsi.Text + "%\"", baglan);
    baglan.Open();
    SqlDataReader oku2 = grafik.ExecuteReader();
    while (oku2.Read())
    {
        karsiuруn += double.Parse(oku2[0].ToString());
    }
    baglan.Close();
    chart1.Series["Grafik"].Points.Add(karsiuруn);
    chart1.Series["Grafik"].Points[i + 1].AxisLabel = DateKarsi.Text;
    break;
}
else
{
    if (RadioGun.Checked == true)
    {
        DateTime oncekigun = DateMevcut.Value;
        oncekigun = oncekigun.AddDays(-1);
        DateMevcut.Value = oncekigun;
    }
    else if (RadioAy.Checked == true)
    {
        DateTime oncekigun = DateMevcut.Value;
        oncekigun = oncekigun.AddMonths(-1);
        DateMevcut.Value = oncekigun;
    }
    else

```

```

        {
            DateTime oncekigun = DateMevcut.Value;
            oncekigun = oncekigun.AddYears(-1);
            DateMevcut.Value = oncekigun;
        }
    }
}

DateMevcut.Value = mevcutasil;
DateKarsi.Value = karsiasil;
datepickerchange = true;
if (bunifuToggleSwitch1.Value == true)
{
    chart1.Series["Grafik"].Points[1].Color = Color.Orange;
    chart1.Series["Grafik"].Points[1].BorderColor = Color.Orange;
}
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void IstatistikYazi()
{
    try
    {
        if (RadioGun.Checked == true)
        {
            grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID=" + otelid + "" and GelisTarihi
<= " + DateMevcut.Text + "" and GidisTarihi >= " + DateMevcut.Text + "", baglan);
            int gunsayi = 0;
            baglan.Open();
            SqlDataReader oku = grafik.ExecuteReader();
            while (oku.Read())
            {
                gunsayi += 1;
            }
            baglan.Close();
        }
    }
}

```

```

        DoluOda.Text = gunsayi.ToString();
    }
    else if (RadioAy.Checked == true)
    {
        grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID=" + otelid + " and GelisTarihi
like '" + DateMevcut.Text + "%' and GidisTarihi like '" + DateMevcut.Text + "%'", baglan);
        int gunsayi = 0;
        baglan.Open();
        SqlDataReader oku = grafik.ExecuteReader();
        while (oku.Read())
        {
            gunsayi += int.Parse(oku[3].ToString());
        }
        baglan.Close();
        int aydakigun = DateTime.DaysInMonth(DateMevcut.Value.Year, DateMevcut.Value.Month);
        int a = gunsayi / aydakigun;
        DoluOda.Text = a.ToString();
    }
    else
    {
        grafik = new SqlCommand("SELECT Distinct GelisTarihi,GidisTarihi,OdaNo
,DATEDIFF(day,GelisTarihi,GidisTarihi) from GirisGecmis where OtelID=" + otelid + " and GelisTarihi
like '" + DateMevcut.Text + "%' and GidisTarihi like '" + DateMevcut.Text + "%'", baglan);
        int gunsayi = 0;
        baglan.Open();
        SqlDataReader oku = grafik.ExecuteReader();
        while (oku.Read())
        {
            gunsayi += int.Parse(oku[3].ToString());
        }
        baglan.Close();
        int a = gunsayi / 365;
        DoluOda.Text = a.ToString();
    }
    grafik = new SqlCommand("select COUNT(*) from RezRapor where OtelID=" + otelid + " and
GelisTarih like '" + DateMevcut.Text + "%'", baglan);
    baglan.Open();
    SqlDataReader oku2 = grafik.ExecuteReader();

```

```

while (oku2.Read())
{
    RezSayi.Text = oku2[0].ToString();
}
baglan.Close();
double asilciro = 0;
grafik = new SqlCommand("select Odeme from OdemelerGecmis where OtelID='" + otelid + "'"
and OdemeTarih like '" + DateMevcut.Text + "%'", baglan);
baglan.Open();
SqlDataReader oku3 = grafik.ExecuteReader();
while (oku3.Read())
{
    asilciro += double.Parse(oku3[0].ToString());
}
baglan.Close();
Ciro.Text = asilciro.ToString() + "₺";
string misafir = "";
grafik = new SqlCommand("select Count(*) from RezRapor where OtelID='" + otelid + "' and"
GelisTarih like '" + DateMevcut.Text + "%' and uyruk=1", baglan);
baglan.Open();
SqlDataReader oku4 = grafik.ExecuteReader();
while (oku4.Read())
{
    misafir = oku4[0].ToString();
}
baglan.Close();
misafir += "/";
grafik = new SqlCommand("select Count(*) from RezRapor where OtelID='" + otelid + "' and"
GelisTarih like '" + DateMevcut.Text + "%' and uyruk=0", baglan);
baglan.Open();
SqlDataReader oku5 = grafik.ExecuteReader();
while (oku5.Read())
{
    misafir += oku5[0].ToString();
}
baglan.Close();
Misafirler.Text = misafir;
double asilkonak = 0;

```

```

grafik = new SqlCommand("select ToplamOdaFiyati from CikisGecmis where OtelID=''' + otelid
+ ''' and Tarih like '''+ DateMevcut.Text + "%'', baglan);
baglan.Open();
SqlDataReader oku6 = grafik.ExecuteReader();
while (oku6.Read())
{
    asilkonak += double.Parse(oku6[0].ToString());
}
baglan.Close();
KonaklamaGelir.Text = asilkonak.ToString() + "₺";
double asilurun = 0;
grafik = new SqlCommand("select EkstraHarcamalar from CikisGecmis where OtelID=''' +
otelid + ''' and Tarih like '''+ DateMevcut.Text + "%'', baglan);
baglan.Open();
SqlDataReader oku7 = grafik.ExecuteReader();
while (oku7.Read())
{
    asilurun += double.Parse(oku7[0].ToString());
}
baglan.Close();
UrunGelir.Text = asilurun.ToString() + "₺";
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void PanelOdaSayi_Click(object sender, EventArgs e)
{
    sorgutipi = 1;
    Grafik();
}

private void PanelRezervasyon_Click(object sender, EventArgs e)
{
    sorgutipi = 2;
    Grafik();
}

private void PanelCiro_Click(object sender, EventArgs e)
{

```

```

sorgutipi = 3;
Grafik();
}
private void PanelMisafir_Click(object sender, EventArgs e)
{
    sorgutipi = 4;
    Grafik();
}
private void PanelKonaklama_Click(object sender, EventArgs e)
{
    sorgutipi = 5;
    Grafik();
}
private void PanelUrun_Click(object sender, EventArgs e)
{
    sorgutipi = 6;
    Grafik();
}
private void DateMevcut_ValueChanged(object sender, EventArgs e)
{
    if(datepickerchange==true)
        Grafik();
    IstatistikYazi();
}
private void DateKarsi_ValueChanged(object sender, EventArgs e)
{
    if(datepickerchange==true)
        Grafik();
}
private void bunifuToggleSwitch1_OnValuechange(object sender, EventArgs e)
{
    Grafik();
    if (bunifuToggleSwitch1.Value == true)
    {
        DateKarsi.ForeColor=Color.Magenta;
        DateKarsi.Color = Color.Magenta;
        DateKarsi.IconColor = Color.Magenta;
    }
    else

```

```

    {
        DateKarsi.Color = Color.Gray;
        DateKarsi.ForeColor= Color.Gray;
        DateKarsi.IconColor = Color.Gray;
    }
}

private void Istatistikler_Load(object sender, EventArgs e)
{
    bunifuToggleSwitch1.Value = false;
    IstatistikYazi();
    sorgutipi = 1;
    Grafik();
}

}
}

```

## **LOG.CS**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Login
{
    public partial class Log : Form
    {
        public Log()
        {
            InitializeComponent();
        }

        private void Log_Load(object sender, EventArgs e)

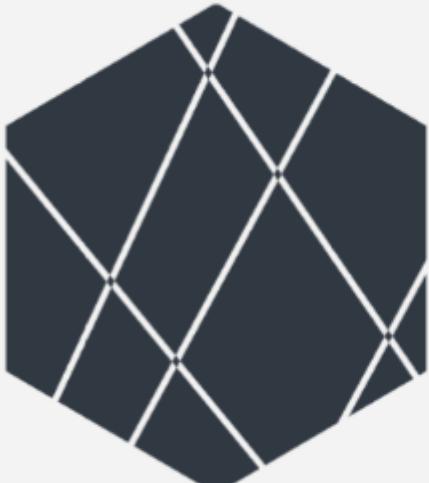
```

```
{  
    bunifuCustomDataGrid1.Rows.Add();  
    bunifuCustomDataGrid1.Rows[0].Cells[0].Value = "04.05.2019";  
    bunifuCustomDataGrid1.Rows[0].Cells[1].Value = "Oda Durumu Değişikliği";  
    bunifuCustomDataGrid1.Rows[0].Cells[2].Value = "Mustafa";  
    bunifuCustomDataGrid1.Rows[0].Cells[3].Value = "DOĞAN";  
    bunifuCustomDataGrid1.Rows[0].Cells[4].Value = "jollydogn";  
    bunifuCustomDataGrid1.Rows.Add();  
    bunifuCustomDataGrid1.Rows[1].Cells[0].Value = "06.04.2018";  
    bunifuCustomDataGrid1.Rows[1].Cells[1].Value = "Rezervasyon Kaydı";  
    bunifuCustomDataGrid1.Rows[1].Cells[2].Value = "Emrah";  
    bunifuCustomDataGrid1.Rows[1].Cells[3].Value = "TUNC";  
    bunifuCustomDataGrid1.Rows[1].Cells[4].Value = "emrah.png";  
    bunifuCustomDataGrid1.Rows.Add();  
    bunifuCustomDataGrid1.Rows[2].Cells[0].Value = "02.01.2016";  
    bunifuCustomDataGrid1.Rows[2].Cells[1].Value = "Rezervasyon Silme";  
    bunifuCustomDataGrid1.Rows[2].Cells[2].Value = "Melih Can";  
    bunifuCustomDataGrid1.Rows[2].Cells[3].Value = "AKGÜNEŞ";  
    bunifuCustomDataGrid1.Rows[2].Cells[4].Value = "melihcanakgunes";  
}  
private void PictureBox1_Click(object sender, EventArgs e)  
{  
}  
}  
}  
}
```

## ÖRNEK YAZICI ÇIKTILARI

LOGİN.CS

DSD OTEL YÖNETİM PROGRAMI X



Kullanıcı Adı:

Şifre:

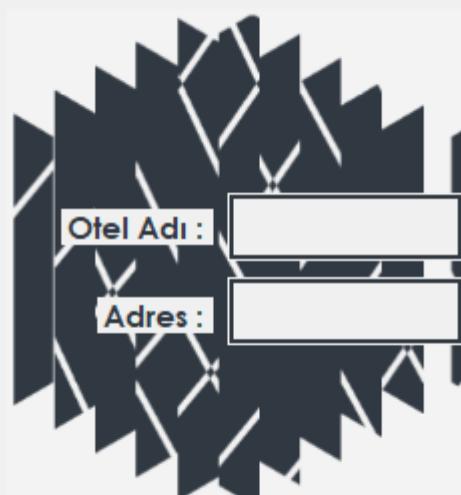
 Giriş

Denep Software Developers

ADMİN.CS

DSD OTEL YÖNETİM PROGRAMI X

Admin Otel Ekleme Ekranı



Otel Yöneticisinin

Otel Adı :

Adres :

Adı:

Soyadı:

Kullanıcı Adı:

Şifre:

 Ekle

Denep Software Developers

## OTELARAYÜZ.CS

Lütfen yukarıdan bir menü seçiniz.

## REZERVASYON.CS

**REZERVASYON**

Misafir Adı veya TC No Girin:

Geliş Tarihi:   ile   Geliş Tararihinde Aralık Kullan:

Uyruğu:  T.C. Vatandaşı  Yabancı  Hepsi

Kimlik Numarası	Geliş Tarihi	Adı	Soyadı	Uyruğu
61006494392	14.06.2019	Melih Can	Akgünəş	Turkey

(\$) 5.8188  
(€) 6.5926

## R\_MUSTERİ\_EKLE.CS

Rezervasyon

**KİMLİK BİLGİLERİ**

Kimlik Belge Türü:	Uyruğu:
Kimlik Seri No:	Cinsiyeti:
Kimlik No:	Medeni Hali:
Ad:	Nüfusa Kayıtlı Olduğu Yer:
Soyad:	İl:
Baba Ad:	İlçe:
Ana Adı:	Mahalle:
Doğum Yeri:	Cilt No:
Doğum Tarihi:	Aile Sıra No: Sıra No:

**DIĞER BİLGİLER**

Yetişkin/Çocuk:	Email:
Telefon:	Araç Plaka No:
Cep Telefon:	İş:
Ülke:	İkamet Adresi:

Notlar:

**GELİŞ BİLGİSİ**

Giriş Tarihi: 12 Haziran 2019 Çarşamba

\*Kırmızı Alanlar Kimlik Bildirim Kanununa Göre Zorunludur.

## REZ\_SİL.CS

Rezervasyon Sil

Kimlik Numarası	Geliş Tarihi	Adı	Soyadı
65165168541	2.06.2019	Meltem	Amağan
26584632354	27.05.2019	Emrah	Tunç
54565498745	27.05.2019	Ali	Kaya
65416574132	27.05.2019	Elizabeth	Stark
54652132103	28.05.2019	Veli	Çelik
56456521332	29.05.2019	Osman	Akarsu
65416354165	30.05.2019	Sylvester	Stallone
16516564156	31.05.2019	Michael	Schumacher
65416541965	27.05.2019	Melek	Demir
61006494392	3.06.2019	Melih Can	Akgüneş
48325137834	31.05.2019	Mustafa	Doğan
65165168541	5.06.2019	Meltem	Amağan
00000000000	5.06.2019	Habibe Bübalo	Özüloğlu

## REZRAPOR.CS

The screenshot shows the REZRAPOR.CS software interface. At the top, there are navigation links: Rezervasyon, Resepsyon, Profiller, Raporlar, Oda Servisi, Kat Hizmetleri, Otel Yönetim, and several status indicators (date, time, help, log out). On the left, a sidebar has icons for adding reservations, deleting reservations, viewing reservation reports, and currency exchange rates (\$ 5.8188, € 6.5926). The main area is titled "REZERVASYON RAPORLARI" and includes search fields for guest name/TC number, arrival date (12 June 2019), departure date (1 February 2020), and a toggle for using arrival dates only. It also shows guest type options (Turkish Citizen, Foreigner, All). A "Göster" (Show) button and a table with one row of data are present. The table columns are: Kimlik Numarası (ID Number), Geliş Tarihi (Arrival Date), Adı (Name), Soyadı (Surname), and Uyruğu (Nationality). The data row is: 61006494392, 14.06.2019, Melih Can, Akgünəş, Turkey.

## RESEPSİYON.CS

The screenshot shows the RESEPSİYON.CS software interface. At the top, there are navigation links: Rezervasyon, Resepsyon, Profiller, Raporlar, Oda Servisi, Kat Hizmetleri, Otel Yönetim, and several status indicators (date, time, help, log out). On the left, a sidebar has icons for adding reservations, deleting reservations, and currency exchange rates (\$ 5.8188, € 6.5926). The main area displays a room status board with rooms 101-SGL, 102-SGL, 103-SGL, 104-SGL (highlighted in blue circle), and 105-TSU. Room 103 is highlighted with a blue circle. Below the board, a callout box provides detailed information about room 103: Oda Tipi: Tek kişilik oda(SGL), Adı ve Soyadı: Hebele hübele egypte egypte, Geliş Tarihi: 06.2019 00:00:00, Gidiş Tarihi: 15.06.2019 00:00:00, Oda Durumu: Dolu. Below the board, sections for "FOLIO BİLGİLERİ" and "HATIRLATMALAR" (with a history entry for 2019-06-06 at 12:00:03) are shown. At the bottom, sections for "3 GÜN İÇİNDE ÇIKISLAR" and "1 HAFTA İÇİNDE GİRİŞ YAPACAKLAR" show reservation details for room 305 and 103 respectively.

## R\_KONAK.CS

Form1

Giriş / Müşteri Bilgileri Harcamalar / Ödemeler /Çıkış

- Giriş Bilgileri / Check in -

Geliş Tarihi :	12 Haziran 2019 Çarşamba	Oda Tipi:	Tek kişilik oda(SGL)	<input type="checkbox"/> Televizyon
Ayrılış Tarihi :	13 Haziran 2019 Perşembe	Oda No:	101	<input type="checkbox"/> Klima
Gece Sayısı :	1			<input type="checkbox"/> Buzdolabı
Fiyat Kodu :	Toplam Fiyat			<input type="checkbox"/> MiniBar
				<input type="checkbox"/> Telefon

Özel İstekler / Notlar :

- Konaklayan Müşteriler -

Adı	Soyadı	Cinsiyet	Uyruğu	Geliş Tarihi	Ayrılış Tarihi
Meltem	Armağan	Kadın	Türk	2019-06-02	2019-06-03
*					



Form1

Giriş / Müşteri Bilgileri Harcamalar / Ödemeler /Çıkış

Konaklama Harcamaları -

Kaldığı Gün Sayısı	Kişi Bilgisi	Tutar
1	Meltem Armağan	120

Extra Harcamalar -

Adet	Ürün Adı	Tutar
1	Çikolatalı Pasta	15
10	Çikolatalı Pasta	150
10	Çilek Frozen	120

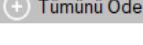
Ödemeler -

Oda Numarası	Ödeme Tarihi	Ödeme
101	2.06.2019 12:43:45	285₺

Toplam Oda Fiyatı:

Ekstra Harcamalar:

Tahsil Edilen Harcama:



Toplam Ödemeler:

Genel Toplam :

ÖDENECEK TUTAR:

## REZ\_TİKLA.CS

The screenshot shows a Windows application window titled "rez\_tikla". At the top left is a search bar labeled "Rezervasyon Ara" with a magnifying glass icon. On the right is a button labeled "Rezervasyon Ekle" with a user icon. Below the search bar is a table with four columns: "Kimlik No", "Ad", "Soyad", and "Geliş Tarihi". A single row is visible, containing the values: Kimlik No - 48325137834, Ad - Mustafa, Soyad - Doğan, and Geliş Tarihi - 12.06.2019 00:00:00.

Kimlik No	Ad	Soyad	Geliş Tarihi
48325137834	Mustafa	Doğan	12.06.2019 00:00:00

## HATIRLATMAEKLE.CS

The screenshot shows a Windows application window titled "Hatirlatmaekle". It contains two input fields: "Hatırlatma Başlığı:" and "Hatırlatma Metni:". Below these fields is a "Kaydet" (Save) button.

Hatırlatma Başlığı:

Hatırlatma Metni:

**Kaydet**



## Hatırlatma İçeriği

AKBS Sistemine gelen misafirleri ekle.

## PROFİLLER.CS

OFFICE DENEYİS DEVELOPERS
 Rezervasyon
 Resepsiyon
 Profiller
 Raporlar
 Oda Servisi
 Kat Hizmetleri
 Otel Yönetim
- 12.6.2019
(?) - 17:05:44
Yardım
Çıkış

Excel'e Gönder
  
 Yazdır

Misafir Ara

Tc Kimlik No:

Ad:

Soyad:

Baba Adı:

Ana Adı:

Doğum Yeri:

Doğum Tarihi:

Uyruğu:

Kişiyi Güncelle

Cep Telefon:

Email:

İş:

İkamet Adresi:

Notlar:

OfelID	TcKimlikNo	Ad	Soyad	KaldığıGeceSayısı	OdaNo	GirişTarihi	GidisiTarihi
11	61006494392	Melih Can	Akgüneş	15	305	30.05.2019	14.06.2019
11	61006494392	Melih Can	Akgüneş	35	305	27.05.2019	1.07.2019

256

## KONAKLAMALAR.CS

The screenshot shows the KONAKLAMALAR.CS application interface. At the top, there is a header bar with the logo "DENEK İŞ İŞLETİMLERİ", the title "KONAKLAMA RAPORLARI", and various menu items: Rezervasyon, Resepsyon, Profiller, Raporlar, Oda Servisi, Kat Hizmetleri, Otel Yönetim, Yardım, and Çıkış. Below the header, there is a search form with fields for "Kimlik Numarası Girin:" (Enter ID Number), "Giriş Tarihi:" (Arrival Date) set to 12 Haziran 2018 Salı (Wednesday, June 12, 2018) and "Gidiş Tarihi:" (Departure Date) set to 13 Haziran 2019 Perşembe (Thursday, June 13, 2019), and "Giriş Tarihinde Aralık Kullan:" (Use Arrival Date Range) and "Gidiş Tarihinde Aralık Kullan:" (Use Departure Date Range) both checked. There are also radio buttons for "İçerde Bulunanlar" (In room), "Ayrılanlar" (Leaving), and "Hepsi" (All). A "Göster" (Show) button is located below the search form. The main area displays a table with the following data:

Kimlik Numarası	Giriş Tarihi	Gidiş Tarihi	Oda Numarası	Kaldığı Gece Sayısı	Tahsil Edilen Tutar
65165168541	2.06.2019	3.06.2019	101	1	405,0000
61006494392	30.05.2019	14.06.2019	305	15	6151,0000
98498498498	5.06.2019	15.06.2019	103	10	1257,0000

## ODASİPARİŞ.CS

The screenshot shows the ODASİPARİŞ.CS application interface. At the top, there is a header bar with the logo "DENEK İŞ İŞLETİMLERİ", the title "ODA SİPARİŞİ", and various menu items: Rezervasyon, Resepsyon, Profiller, Raporlar, Oda Servisi, Kat Hizmetleri, Otel Yönetim, Yardım, and Çıkış. Below the header, there is a form for entering an order. It includes fields for "Oda No =" (Room No) set to 101, "Yiyecek Adı =" (Food Item) with a dropdown menu containing "Açık Yiyecek", "Çikolatalı Pasta", "Muzlu Pasta", "Profiterol", "Karaorman", and "Malaga Bitter", and an "Adet:" (Quantity) input field set to 1; there is also a "Listeye Ekle" (Add to list) button. Another section shows "İçecek Adı =" (Drink Item) with a dropdown menu containing "Açık İçecek", "Çilek Frozen", "Karpuz Frozen", "Şeftali Frozen", "Frambuaz Frozen", and "Karamelli Milkshake", and an "Adet:" input field set to 1; there is also a "Listeye Ekle" button. To the right, a "Seçenler" (Selected Items) box displays "1-Çikolatalı Pasta = 15" and "1-Karpuz Frozen = 12". At the bottom, there is a table with the following data:

Oda No	Siparişler	Tutar
305	2-Şeftali Frozen = 24	24₺

## URUNEKLE.CS

Ürün Ekleme

Yiyecek Adı = <input type="text"/>	İçecek Adı = <input type="text"/>
Fiyatı = <input type="text"/>	Fiyatı = <input type="text"/>
<input checked="" type="checkbox"/> Ekle	<input checked="" type="checkbox"/> Ekle

Yiyecek Adı	Yiyecek Fiyatı
Çikolatalı Pasta	15₺
Muzlu Pasta	15₺
Profiterol	12₺
Karaorman	10₺
Malaga Bitter	10₺
Krokanlı Mono	10₺
Budapeşte	12₺
Jakompe	12₺
Czibiz Köfte	25₺
Eti Quasedilla	27₺

İçecek Adı	İçecek Fiyatı
Çilek Frozen	12₺
Karpuz Frozen	12₺
Şeftali Frozen	12₺
Frambuaz Frozen	12₺
Karamelli Milkshake	12₺
Kutu Kola	5₺
Ayran	3₺
Vanilyalı Frappe	12₺
Limonata	6₺
Şışe Soda	6₺

## KATHİZMETLERİ.CS

Software Developers

Rezervasyon Resepsyon Profiller Raporlar Oda Servisi Kat Hizmetleri Otel Yönetim - 12.6.2019 - 17.09.19 Yardım Çıkış

<input type="checkbox"/> Tümünü Onayla	ODA DURUMLARI Onay Bekleyenler																						
<input type="checkbox"/> Oda Durumunu Değiştir	307	308																					
<input type="checkbox"/> DURUM TÜRÜ = <input checked="" type="checkbox"/> TEMİZ <input checked="" type="checkbox"/> DOLU <input checked="" type="checkbox"/> KIRLI <input checked="" type="checkbox"/> ARIZALI(OOS) <input checked="" type="checkbox"/> ARIZALI(OOO)																							
<table border="1"> <tr><td>101</td><td>102</td><td>103</td><td>104</td><td>105</td><td>106</td><td>107</td></tr> <tr><td>108</td><td>109</td><td>110</td><td>201</td><td>202</td><td>203</td><td>204</td></tr> <tr><td>205</td><td>206</td><td>207</td><td>208</td><td>209</td><td>210</td><td>301</td></tr> </table>			101	102	103	104	105	106	107	108	109	110	201	202	203	204	205	206	207	208	209	210	301
101	102	103	104	105	106	107																	
108	109	110	201	202	203	204																	
205	206	207	208	209	210	301																	

## ODADURUM.CS

Oda Durumları

Oda No =

Oda Durumu =

ARIZALI (OOS)  ARIZALI (OO)

\* Ariza paneli oda arızalı kutucuğu işaretlenmeden açılmaz.

Ariza Paneli

Televizyon EK NOT

Klima

Buzdolabı

MiniBar

Telefon

Onay Bekleyen

## YÖNETİCİPANELİ.CS

The screenshot displays the Manager Panel interface with the following elements:

- Top Bar:** Includes links for "DEVELOPERS", "Rezervasyon", "Resepsyon", "Profiller", "Raporlar", "Oda Servisi", "Kat Hizmetleri", "Otel Yönetim", a date/time indicator ("12.6.2019 - 17:02"), "Yardım", and "Çıkış".
- Left Sidebar:** Features icons for "Yönetim", "Çalışanlar", "Odalar", "İstatistikler", "İşlem Geçimi (LOG)", and "Misafirler".
- Dashboard Metrics:** Shows "4 Personel", "3 Misafir Sayısı", "675₺ Gündük Gelir", and a "Tarih" box indicating "12 Haziran 2019 Çarşamba".
- Guest Logins:** A chart titled "Misafir Oranı" (Guest Ratio) showing guest counts from June 6 to June 12, 2019. The chart shows a peak on June 6 and a dip on June 11. A circular gauge indicates "Otel Doluluğu" (Hotel Occupancy) at 10%.
- Guest Logins Table:** A table listing guest logins with columns: Foto, Ad, Soyad, Tarih - Saat, and İşlem. It includes entries for "Ali Bağcılar" (Login Record) and "Mehmet Tepe" (Room Change).

## PERSONEL.CS



PERSONEL EKLE

Personel Adı :	Personel Soyadı :
Kullanıcı Adı :	Şifre :

**PERSONEL YETKILENDİRME**

PersonelAd	PersonelSoyad	KullanıcıAdı
Emrah	Tunc	emr
Melih Can	Akgünəş	mih
Mustafa	Doğan	msf
Murat	Akin	murat

Rezervasyon Sayfası :

Resepsiyon Sayfası :

Profiller Sayfası :

Raporlar Sayfası :

Oda Servisi Sayfası :

Kat Hizmetleri Sayfası :

**Personeli Sil**

## ODALAR.CS



**OTEL KURULUMU**

**KAT BİLGİSİ**

Kat Ekle :	<input type="button" value="Kat Ekle"/>
------------	---

**ODA BİLGİSİ**

Yapilandırılan Kat :	<input type="button" value="Kat Ekle"/>
Katdaki Bulunan Oda Sayısı :	<input type="button" value="Odaları Ekle"/>

**ODALARI YAPILANDIR**

SEÇİLİ ODALARIN TİPİNİ BELİRLEYİN

SGL :	<input type="radio"/> (SINGLE ROOM)
TSU :	<input type="radio"/> (TWIN FOR SOLE USE)
TWR :	<input type="radio"/> (TWIN ROOM)
DBL :	<input type="radio"/> (DOUBLE ROOM)
TR :	<input type="radio"/> (TRIPLE ROOM)
QUAD :	<input type="radio"/> (QUAD)

SEÇİLİ ODALARIN FİYATINI BELİRLEYİN

Fiyat :	<input type="button" value="Fiyat Ekle"/>
---------	---

**ODA BİLGİLERİ GÜNCELLEME**

**KAT BİLGİSİ**

Kat :	<input type="button" value="Kat Ekle"/>
-------	---

**ODA BİLGİSİ**

Katdaki Bulunan Oda Sayısı :	<input type="button" value="10"/>
------------------------------	-----------------------------------

**ODALARI YAPILANDIR**

SEÇİLİ ODALARIN TİPİNİ BELİRLEYİN

201	<input type="checkbox"/>
202	<input type="checkbox"/>
203	<input type="checkbox"/>
204	<input type="checkbox"/>
205	<input type="checkbox"/>
206	<input type="checkbox"/>
207	<input type="checkbox"/>
208	<input type="checkbox"/>
209	<input type="checkbox"/>
210	<input type="checkbox"/>

SEÇİLİ ODALARIN FİYATINI BELİRLEYİN

Fiyat :	<input type="button" value="Fiyat Ekle"/>
---------	---

**Güncelle**

# İSTATİSTİKLER.CS

SW
DENEYİ
DE
VELOPERS

Rezervasyon
 Resepsiyon
 Profiller
 Raporlar
 Oda Servisi
 Kat Hizmetleri
 Otel Yönetim

- 12.6.2019
 - 17:12:02
 Yardım
 Çıkış

Yönetim
 Çalışanlar
 Odalar
 İstatistikler
 İşlem Geçimi (LOG)
 Misafirler

Dolu Oda Sayısı **4**

Konaklama Geliri **1200₺**

Rezervasyon Sayısı **2**

Ürün Geliri **57₺**

Ciro **297₺**

Misafirler **1/1**  
Türk/Yabancı

Mevcut Tarih : **2019-06-05**
 Karşılaştırılacak Tarih : **2019-06-12**

Tarih Biçimine Göre Ortalama Dolu Oda Sayısı

Grafik

Tarih Biçimi =
Günlük : 
Aylık : 
Yıllık :

## **SONSÖZ**

“İllerlemenin sırrı, başlamaktan geçer. Başlamanın sırrı ise; Tek bir seferde yapılması zor olan karmaşık işleri, üstesinden gelinebilir daha küçük işlere bölmekten.” **Mark Twain.**

## NOTLAR

