**DOKUZ EYLÜL ÜNİVERSİTESİ**

**İZMİR MESLEK YÜSEKOKULU**

**TEKNİK PROGRAMLAR BÖLÜMÜ**

**BİLGİSAYAR PROGRAMCILIĞI PROGRAMI**

**2017-2018 ÖĞRETİM YILI BİTİRME PROJESİ**

**DSD**

**OTEL YÖNETİM PROĞRAMI**

**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-

* Analiz Detayları ……………………………………………………………………………
* Kullanılan Programlama Dili ve Veritabanı ……………………………………………….
* Görev Dağılımları ………………………………………………………………………….
* Rakip Programların Değerlendirmesi ……………………………………………………...
* Proje Setup Dosyası Hazırlanışı ve Kuruluşu ……………………………………………...
* Projenin Proje SWOT Analizi ……………………………………………………………..
* Rakip SWOT Analizi ………………………………………………………………………
* Çalışma Takvimi …………………………………………………………………………...
* Projenin GANTT Şeması …………………………………………………………………..

Akış Diyagramı……………………………………………………………………………………......

Veritabanı Normalizasyonu……………………………………………………………………….......

Entity Relationship Diyagramı ………………………………………………………………………...

Proğram Kodları ………………………………………………………………………………………..

Login.cs ………………………………………………………………………………………...

Admin.cs ……………………………………………………………………………………….

OtelArayüz.cs ………………………………………………………………………..................

Rezervasyon.cs ………………………………………………………………………………....

R\_musteri\_ekle.cs ………………………………………………………………………...........

Rez\_sil.cs ………………………………………………………………………………………

RezRapor.cs ……………………………………………………………………………………

Resepsiyon.cs ………………………………………………………………………………......

Odaicerik.cs ………………………………………………………………………………….....

R\_konak.cs ……………………………………………………………………………………..

Rez\_tikla.cs ………………………………………………………………………………….....

Hatirlatmaekle.cs …………………………………………………………………………….....

Profiller.cs ………………………………………………………………………………………

Konaklamalar.cs ……………………………………………………………………………......

OdaSiparis.cs ……………………………………………………………………………………

UrunEkle.cs ……………………………………………………………………………………

KatHizmetleri.cs ………………………………………………………………………………

Oda\_Durum.cs ………………………………………………………………………………..

YoneticiPanel.cs ………………………………………………………………………………

Personel.cs …………………………………………………………………………………….

Odalar.cs ………………………………………………………………………………………

Istatistikler.cs ………………………………………………………………………………….

Log.cs ………………………………………………………………………………………….

Örnek Yazıcı Çıktıları …………………………………………………………………………………

Sonsöz …………………………………………………………………………………………………

Notlar ………………………………………………………………………………………………….

**DEĞERLENDİRME** **KRİTERLERİ** **TABLOSU**

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

**ÖNSÖZ**

**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 TUNÇ

* 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 Teknoloşi 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

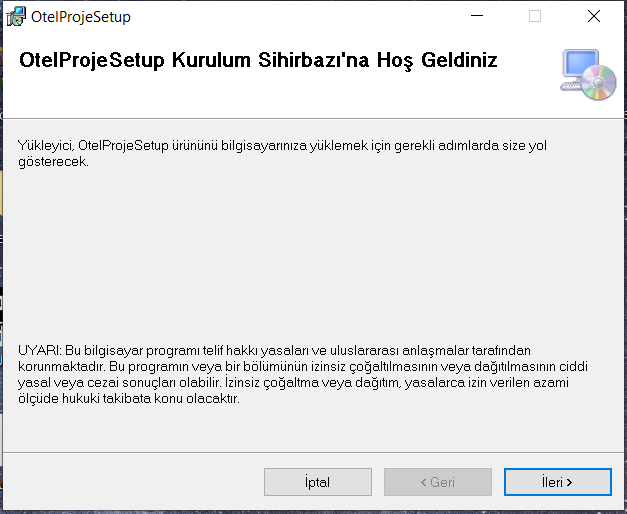
* Ana ekranda tüm odaların bilgilerinin görüntülenmesi misafirlerin gizlilik haklarına aykırıdır.
* Muhasebe sistemleri ile entegre değildir.
* Santral entegrasyonu bulunmamakta.
* Bulut tabanlı değildir, kurulum gerekir.
* Tüm cihazlarda kullanılabilir uyumluluğu bulunmamakta.
* Grafikler yüksek değerlerde taşma yaşamakta ve değerler üst üste binmekte.
* 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 anlaşılırdır.
* Departman bazında ana sayfalara ve sekmelere sahiptir.
* Odalara arıza bilgisi girilebilir.
* Yönetici panelinde çeşitli istatistikler için grafikler sergilenir.
* Her işlem log olarak tutulur ve yönetici bile silemeyecek şekilde kaydedilir.
* Mesaj gönderimi bulunmaktadır.
* Web paketi ile tesise ait entegre bir web sitesi oluşturulabilinir.
* 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 tesisi cep telefonu üzerinden takip edebilirsiniz.
* Otomatik EGM rapor gönderimi özelliği bulunmaktadır.

Basit Otel Yönetim Programı

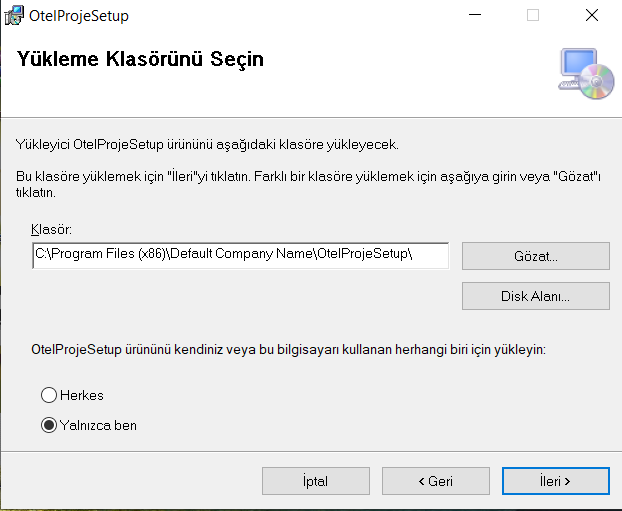
* Polis ve Jandarma sistemlerine bilgiler otomatik olarak gönderilir.
* Kanal Yönetim Sistemi ile bulunmamakta
* Bulut tabanlıdır, kurulum gerekmez.
* Tüm cihazlarda kullanılabilir uyumluluktadır.

**Proje Setup Dosyası Hazırlanışı ve Kuruluşu**

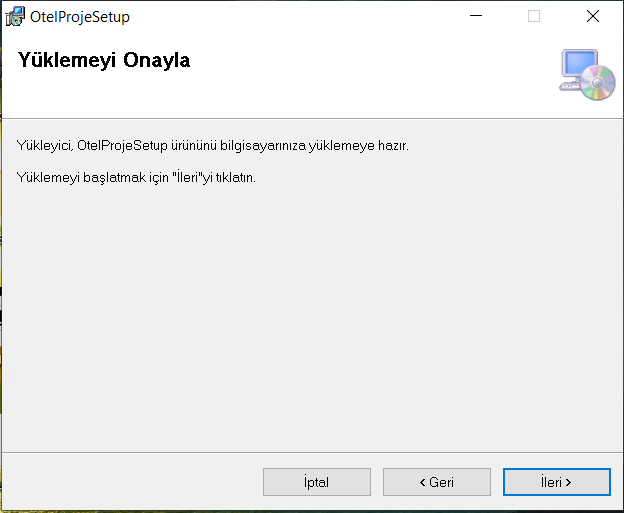
Setup ekran ile başlıyor aşağıda belirli uyarılarımız var. Kullanıcı ileri butonuna basarak ilerler



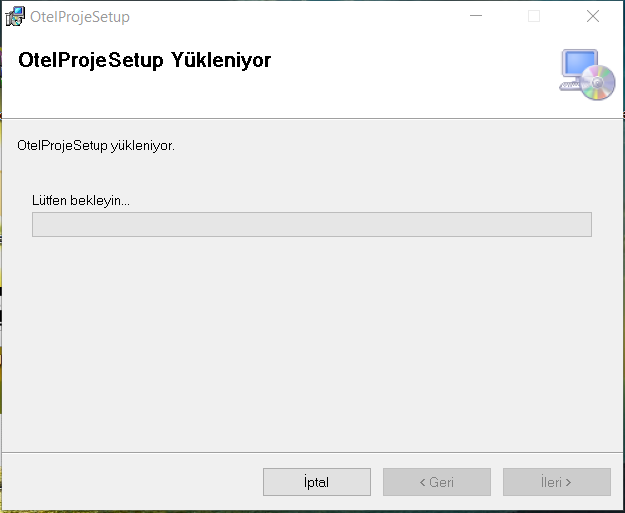
Burada kullanıcı yükleyeceği alanı seçebilir.



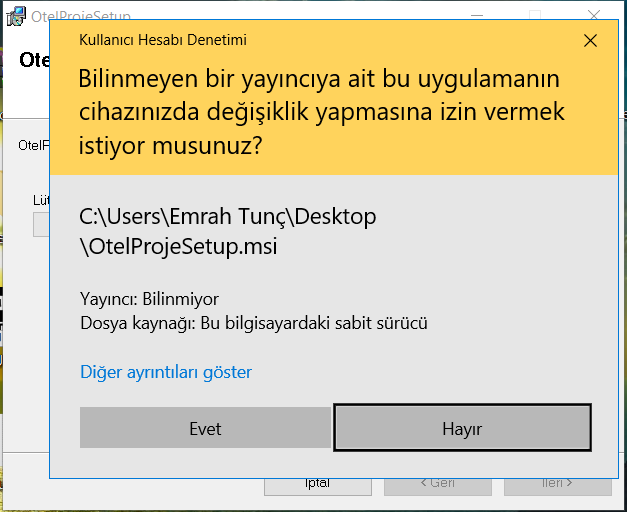
İleri diyerek programı yüklemeye izin veriririz.



Programımız yüklenmeye başlar.



Yükleme sırasında kullanıcıdan izin ister, Evet diyip devam edilir.



Son olarakta kurulum tamamlanır ve masaüstünde programımızın kısayolu oluşur.



**Analiz Aşamasında Görüşülen Kişiler**

Unvan : Müdür

Adı ve Soyadı : Murat AKIN

Çalıştığı Kurum : Alsancak İbis Hotel

Görevi : Müdür

Telefon numarası : 0532 709 00 10

**Projenin Swot Analizi**

**Rakip Swot Analizi**

**Çalışma Takvimi**

**27.02.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

Yapılanlar : Programın taslak özellikleri, menüler, menüler içerisinde ki 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 işleyişi, 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 danıştık 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 Birliğine 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 konuşuldu, çeşitli özellik eklemeleri ve eksiltmeleri not alındı. Proje hedefi birtakım değişmelere 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 tasarımı 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 düzenlendi 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

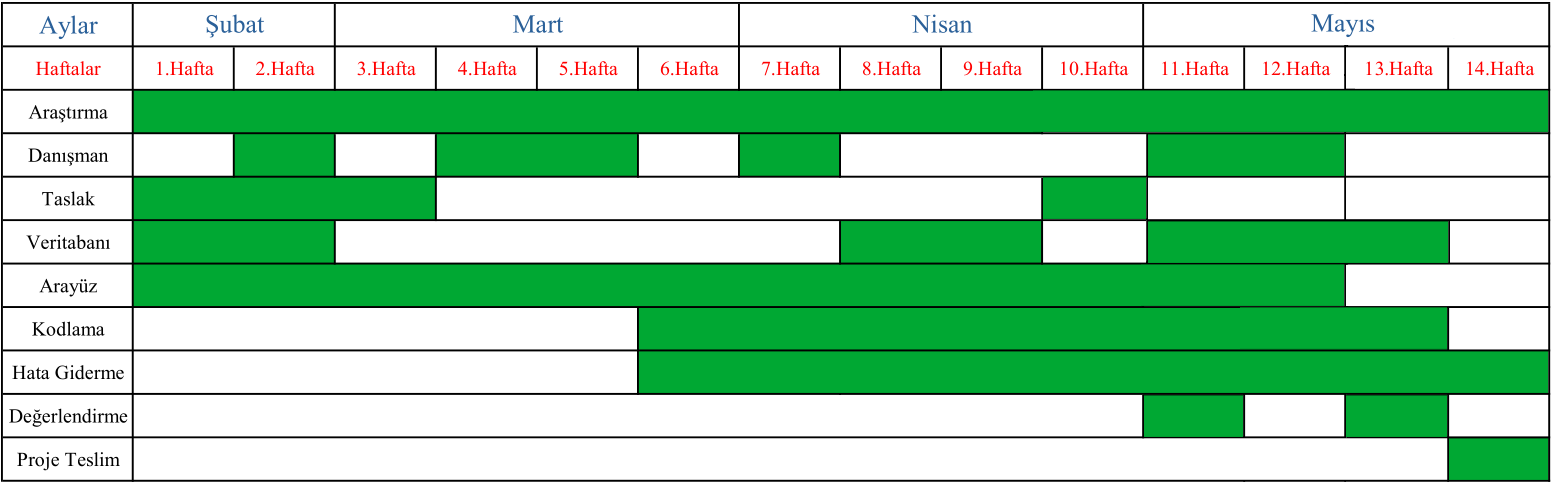
Yapılanlar : Yapılan formlar birbirine entegre edildi ve hata düzeltmeleri yapıldı.

**26.05.2019**

Katılanlar : Melih Can AKGÜNEŞ, Mustafa DOĞAN, Emrah TUNÇ

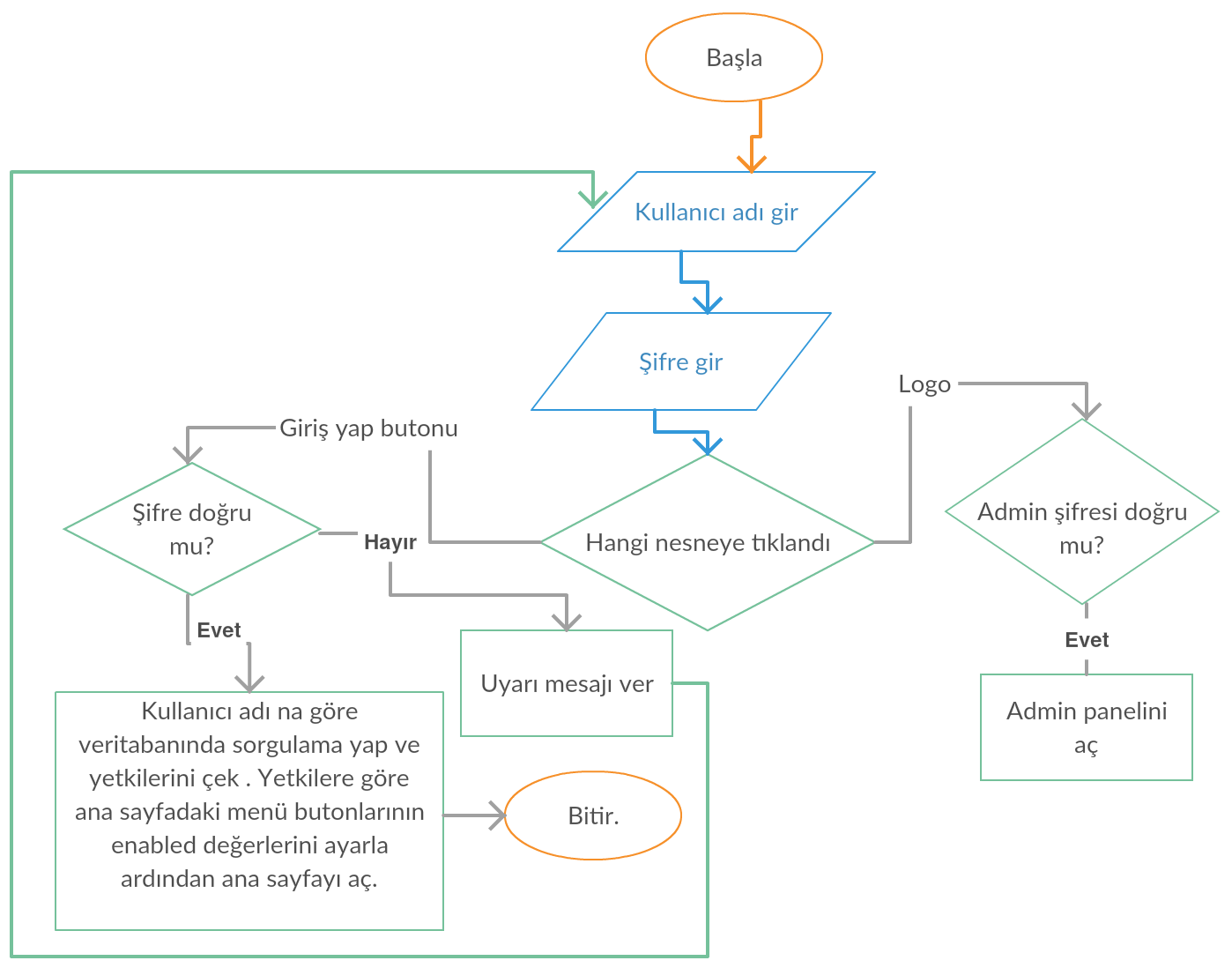
Yapılanlar : Programa veriler girildi. Sunum için her şey tamamlandı.

**Proje GANTT Şeması**

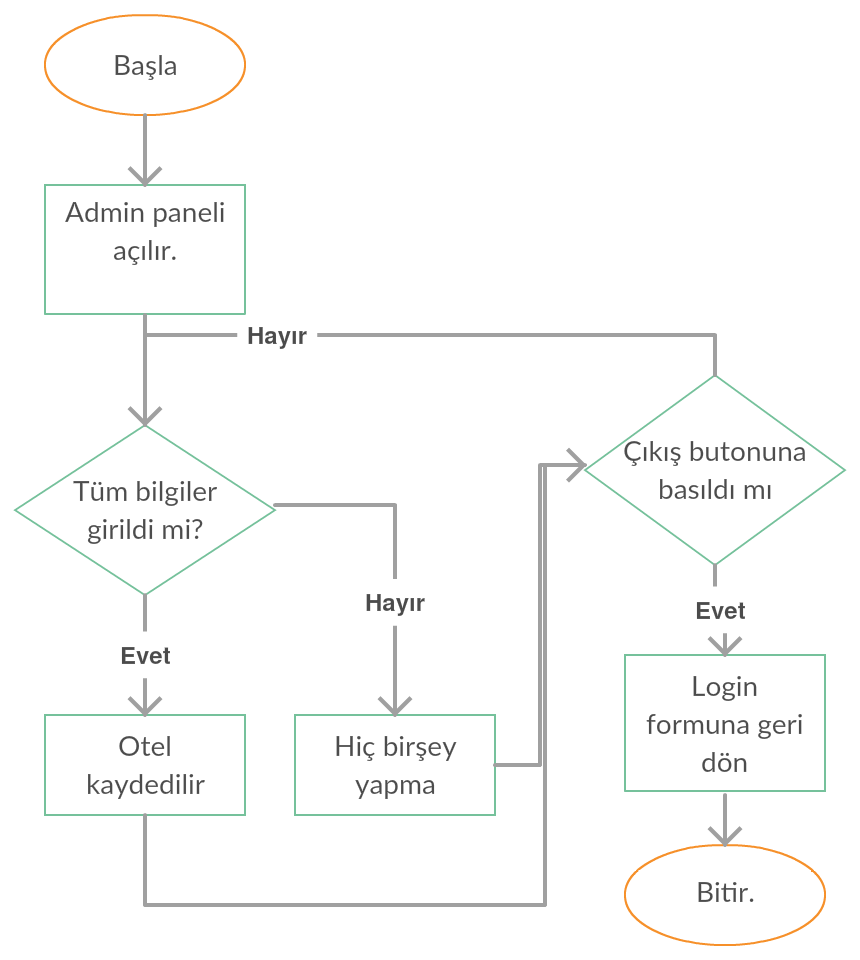
****

**AKIŞ DİYAGRAMI**

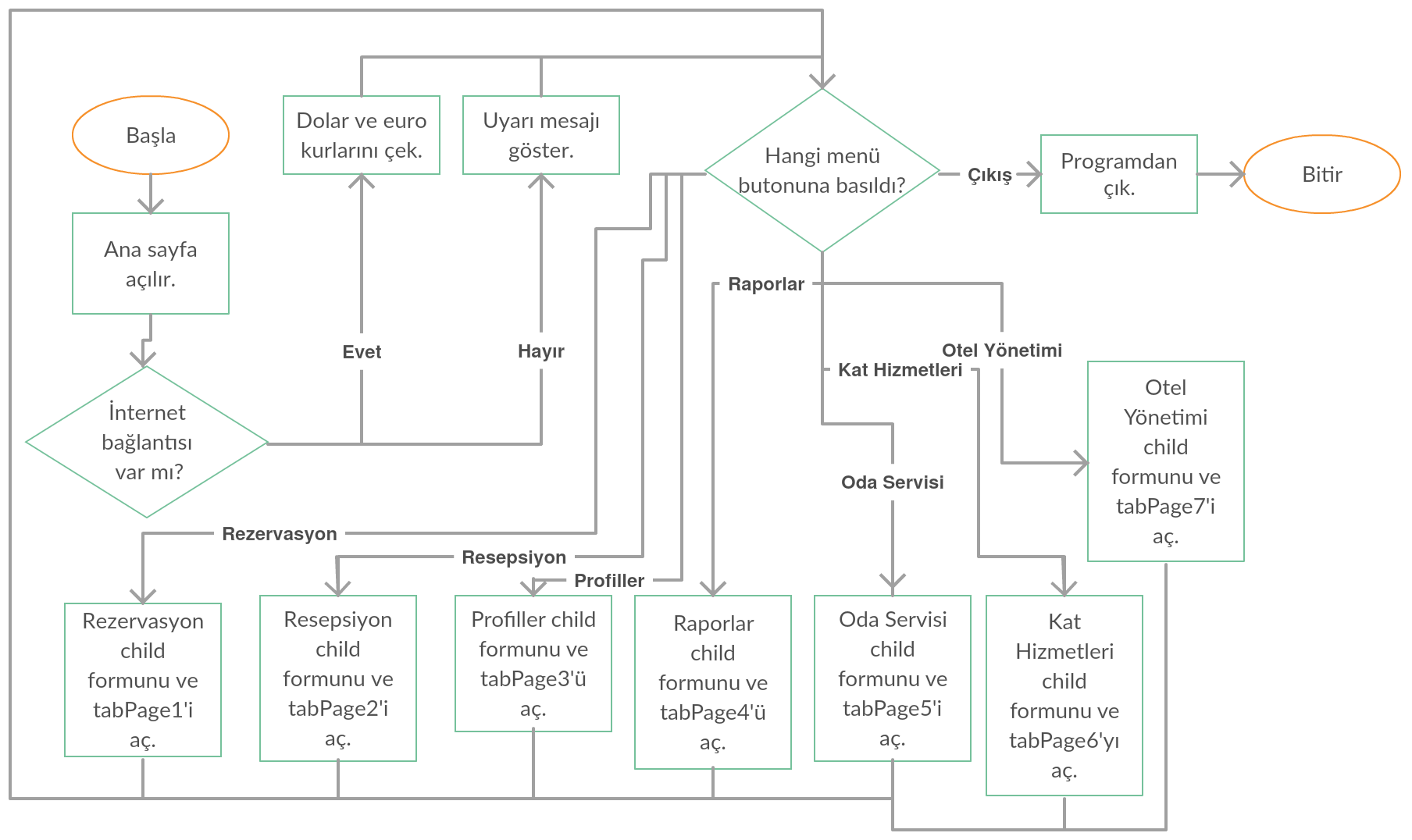
**LOGİN.CS**

****

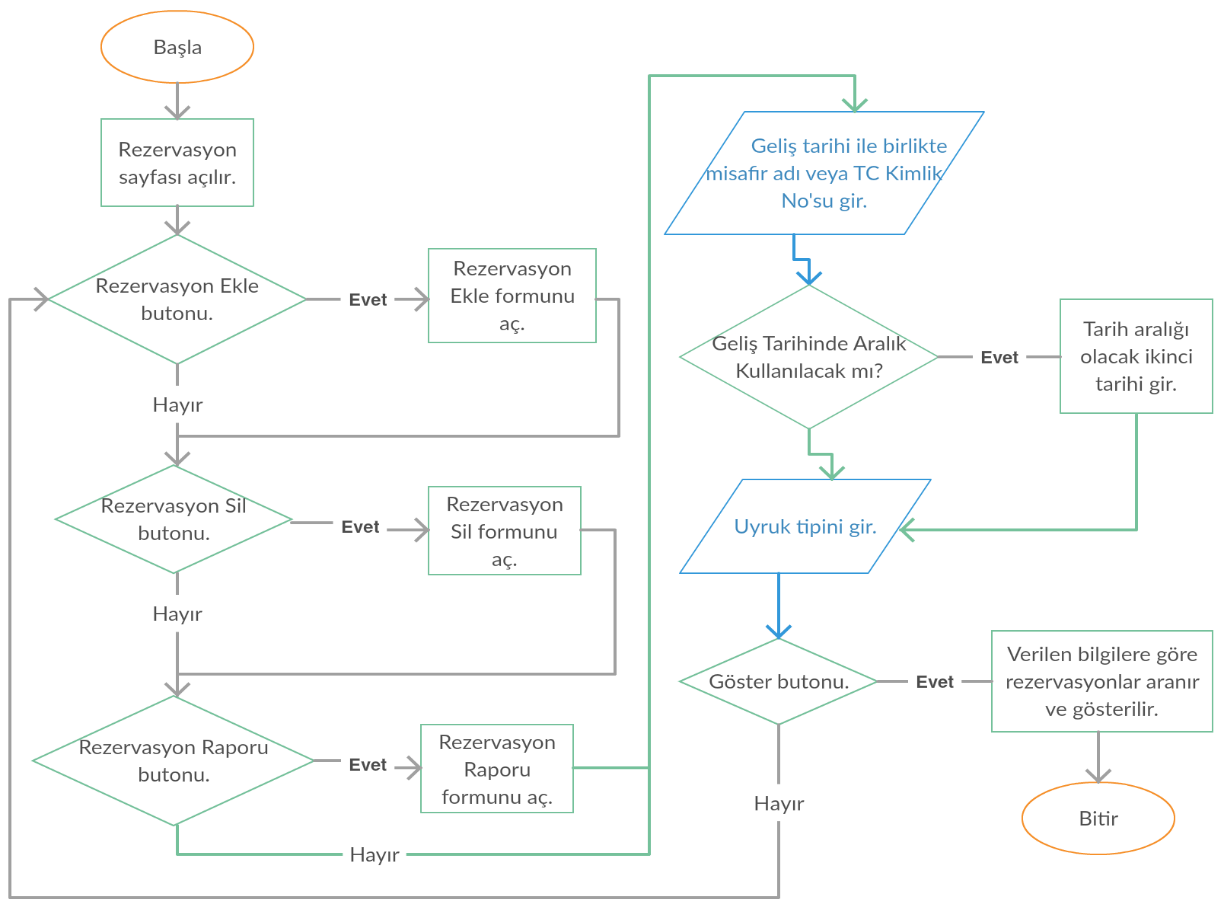
**ADMİN.CS**

****

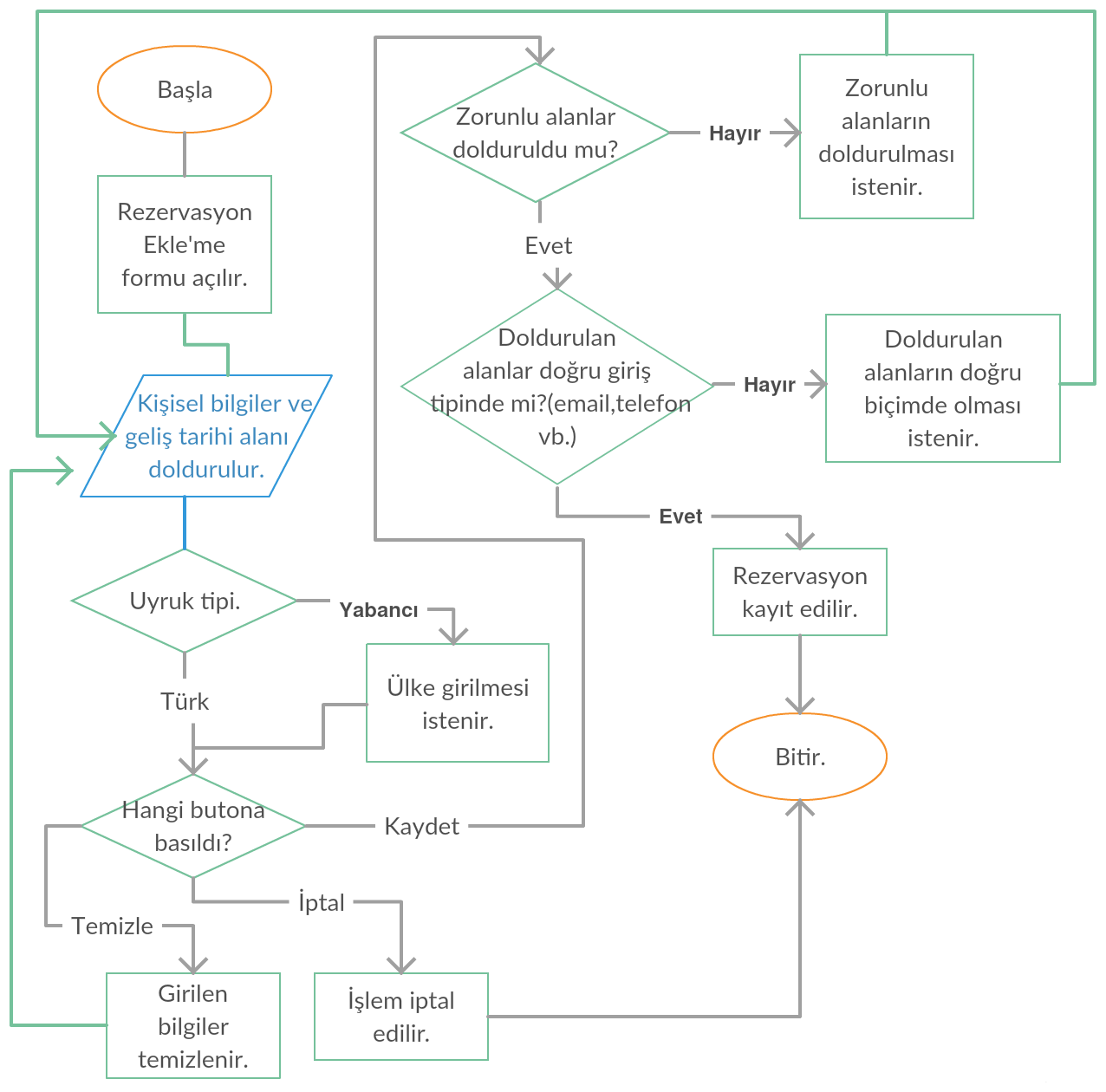
**OTELARAYÜZ.CS**

****

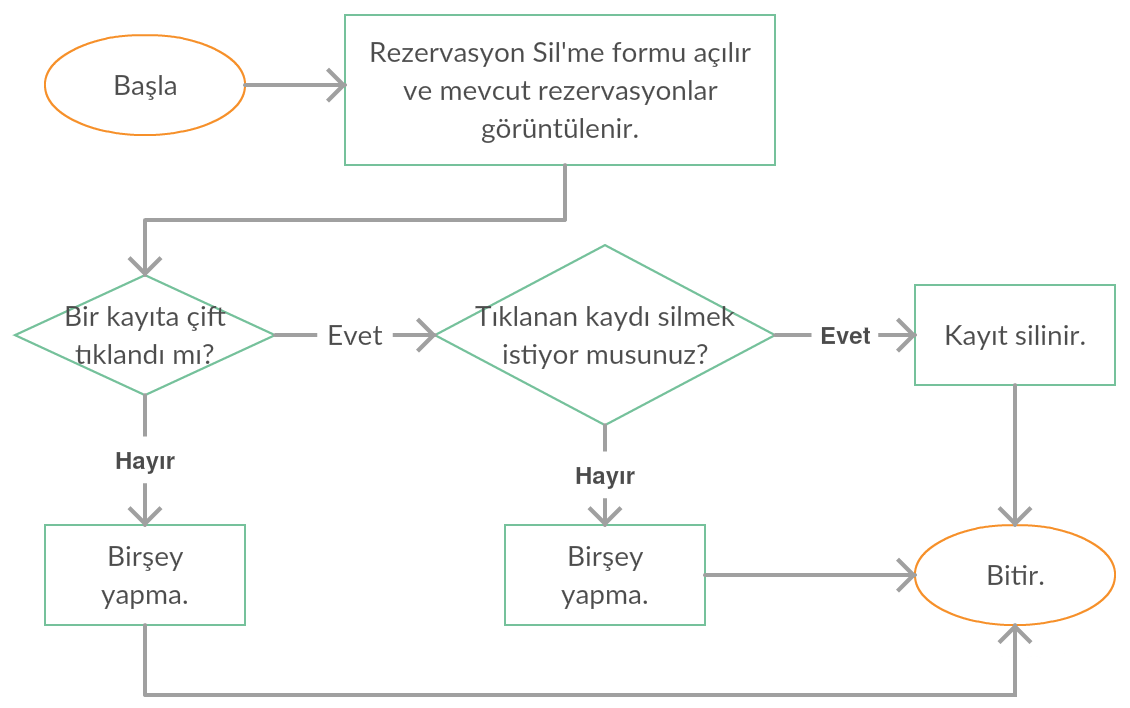
**REZERVASYON.CS**

****

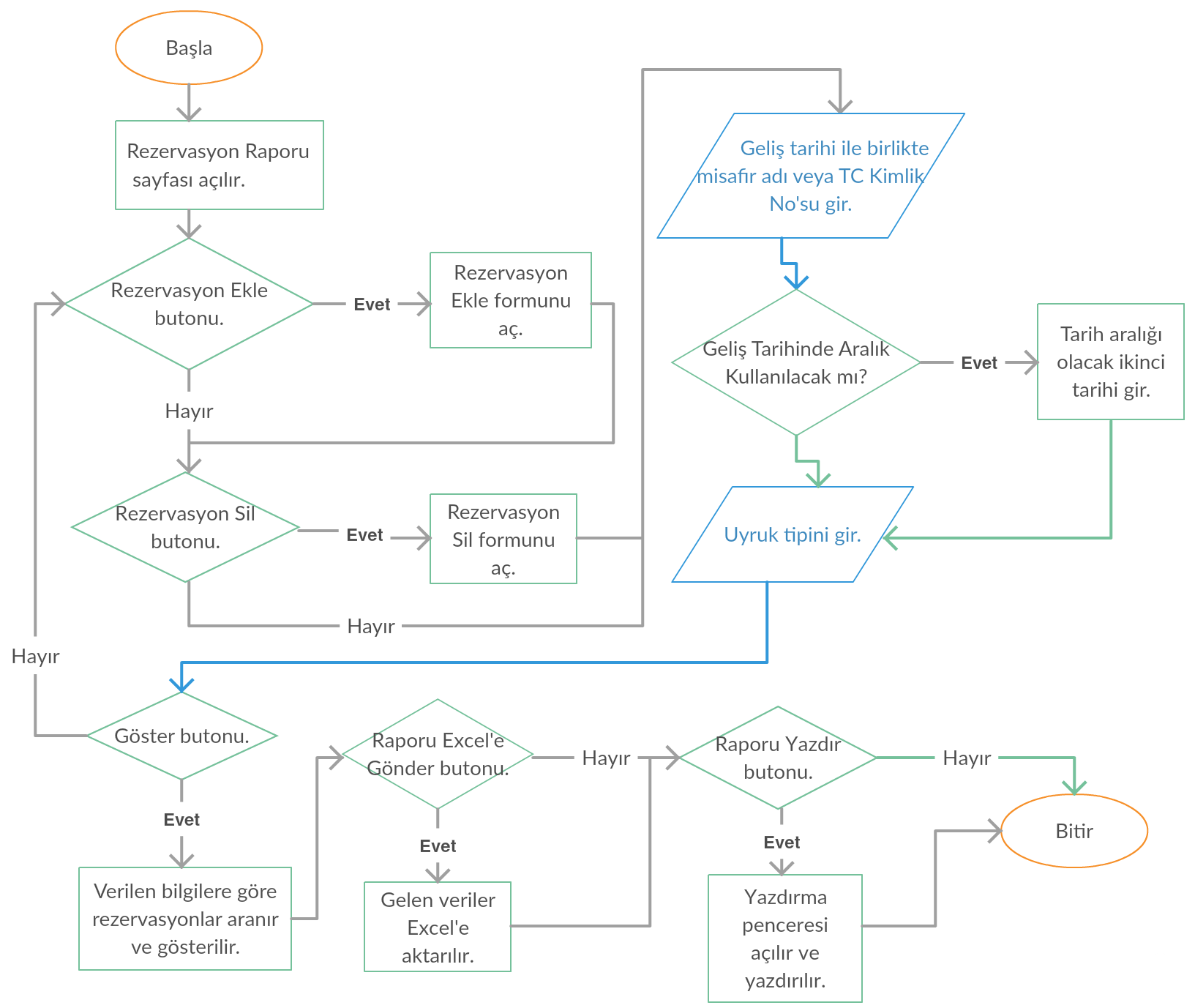
**R\_MUSTERİ\_EKLE.CS**

****

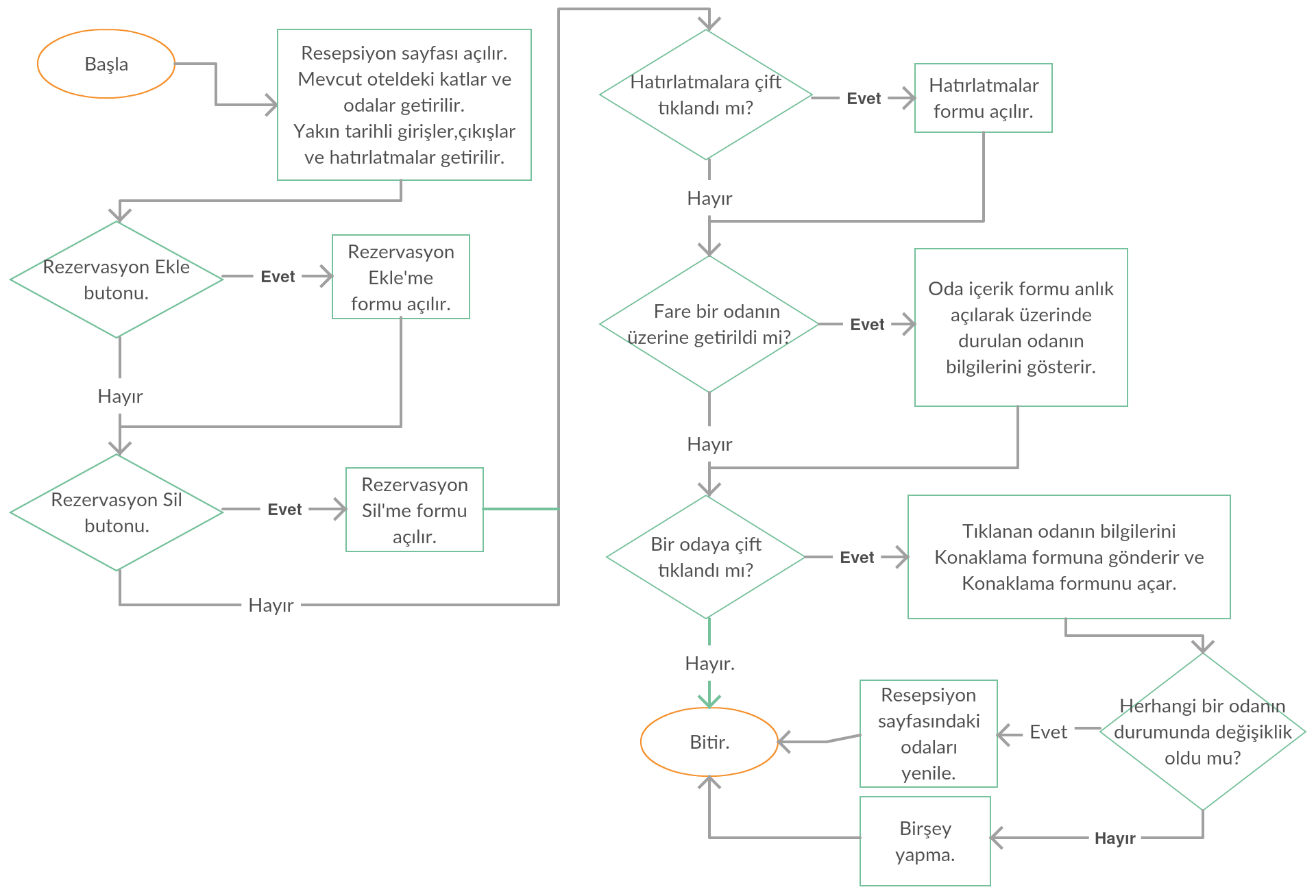
**REZ\_SİL.CS**

****

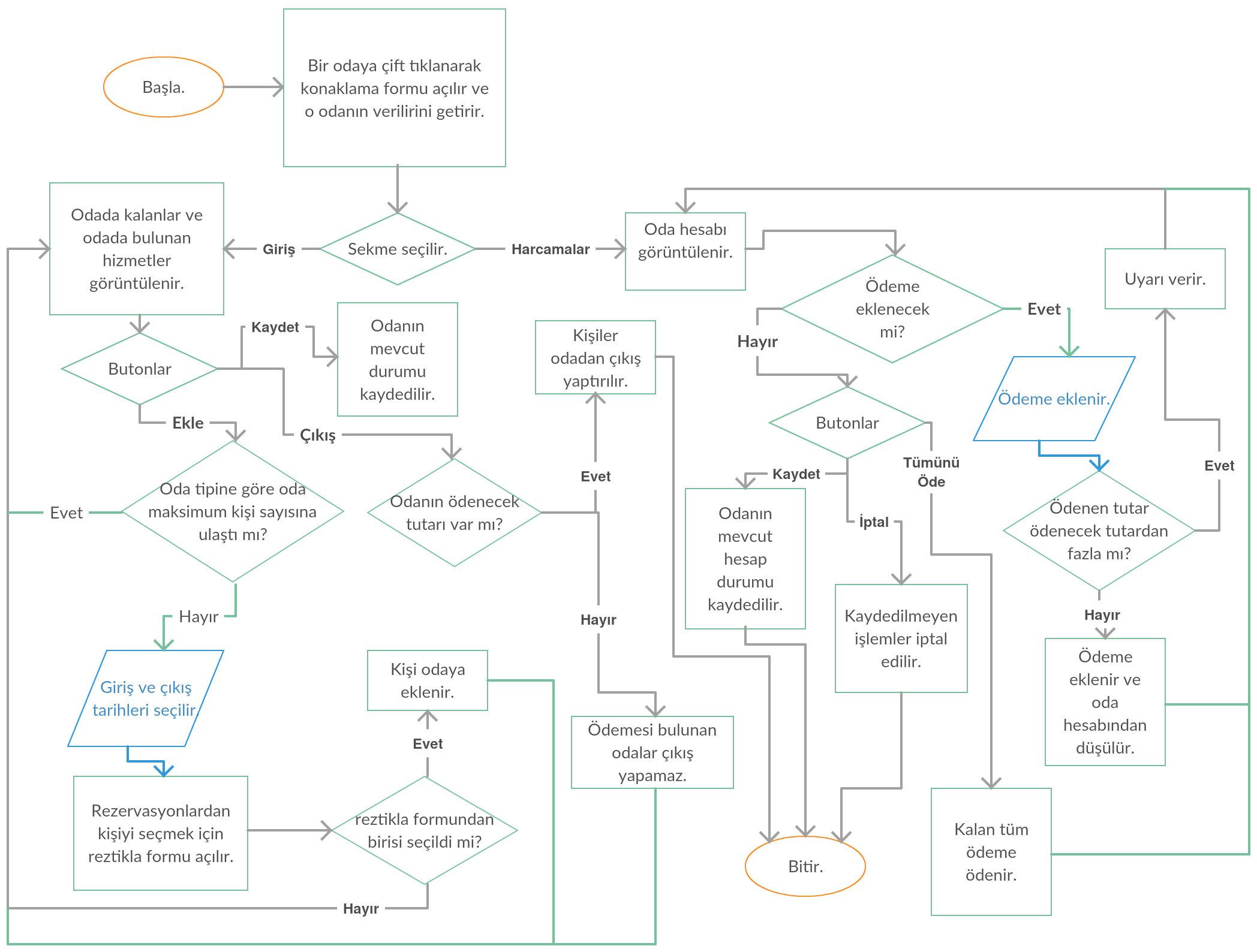
**REZRAPOR.CS**

****

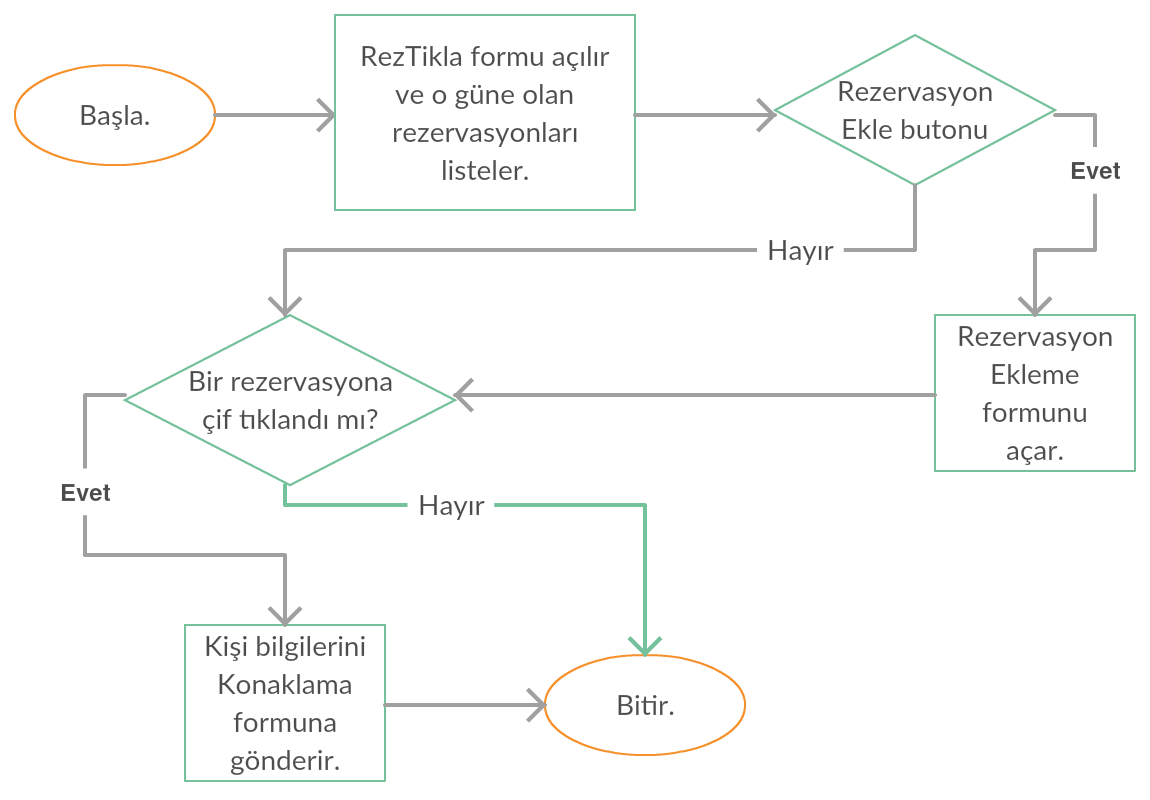
**RESEPSİYON.CS**

****

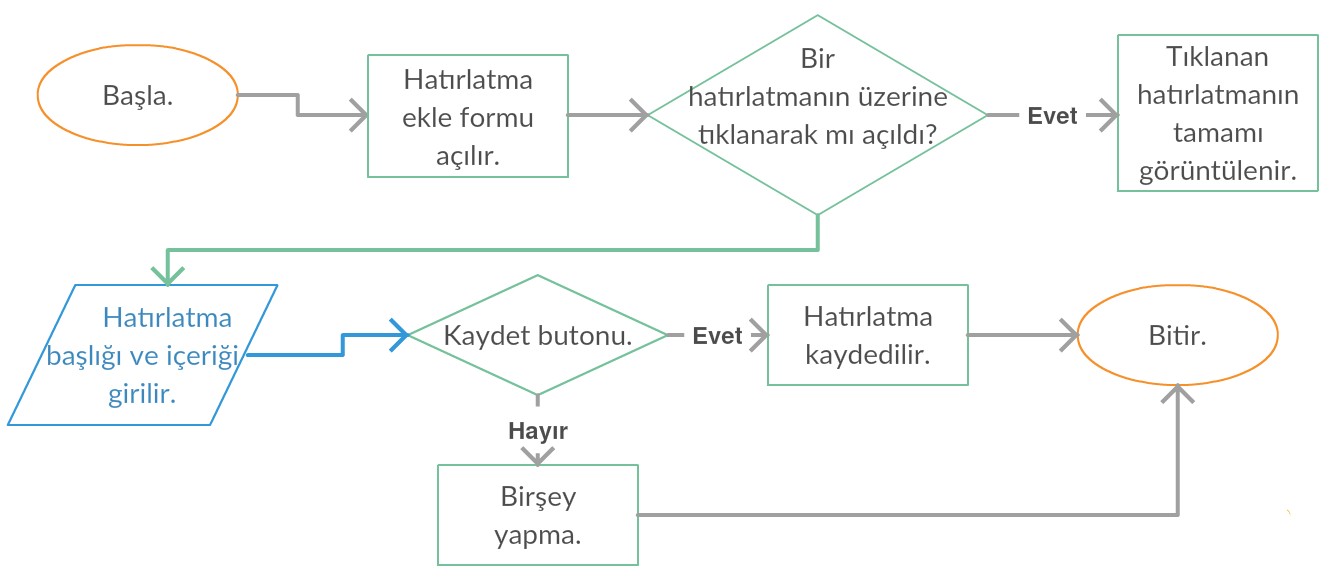
**R\_KONAK.CS**

****

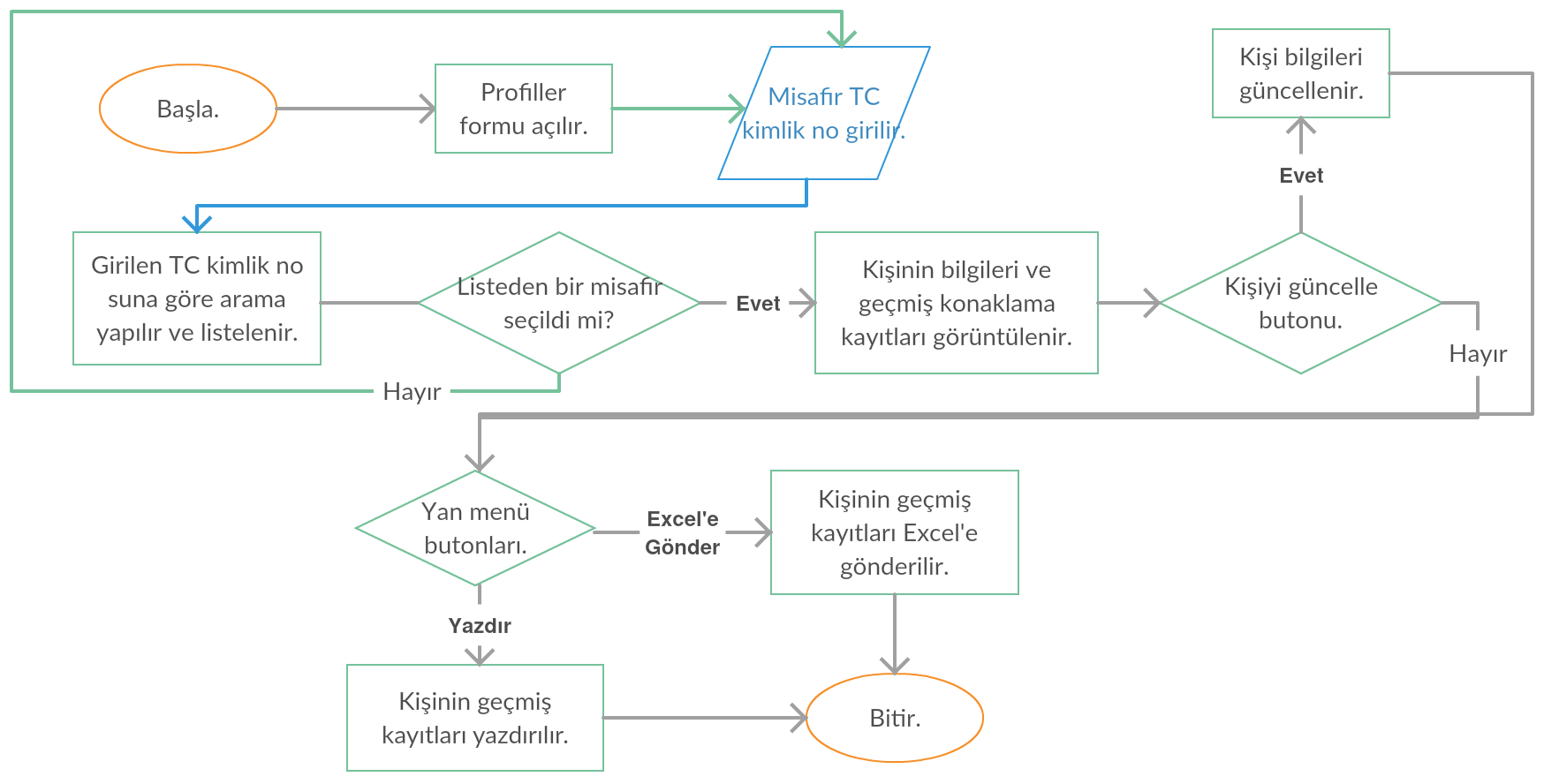
**REZ\_TİKLA.CS**

****

**HATIRLATMAEKLE.CS**

****

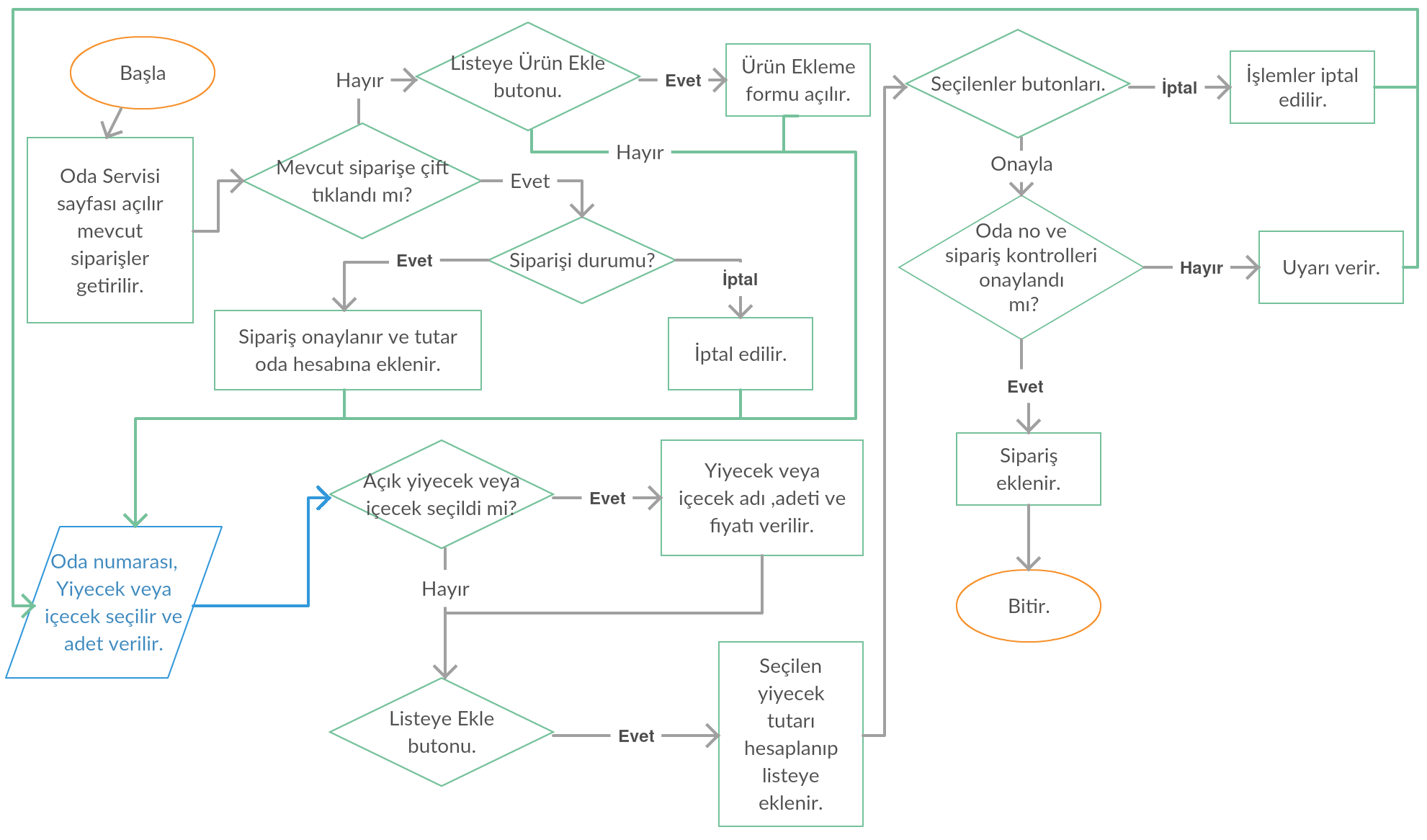
**PROFİLLER.CS**

****

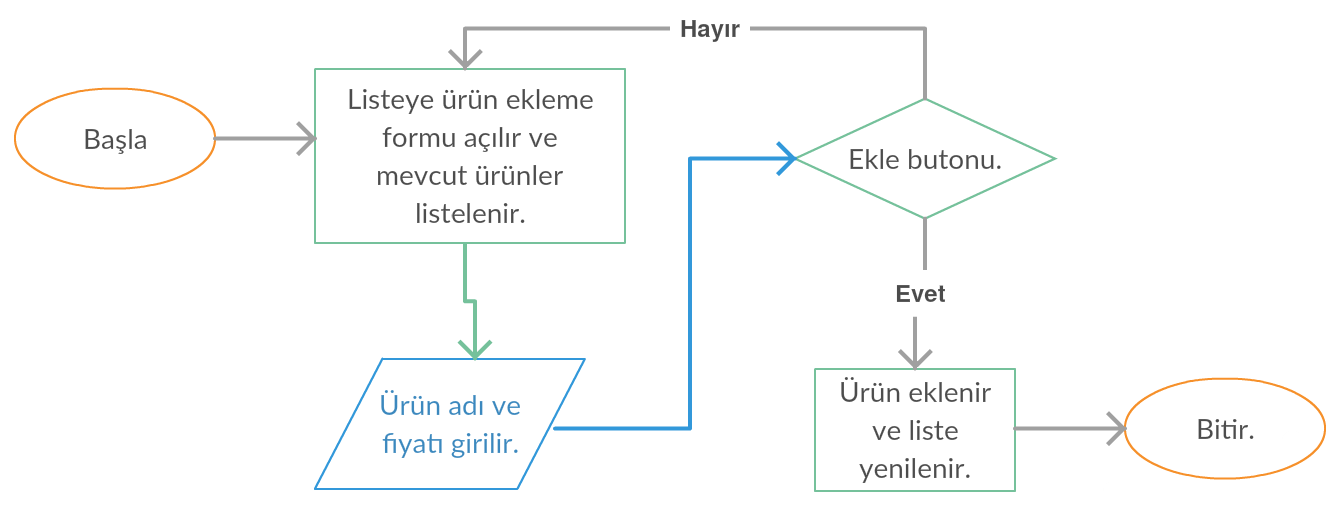
**KONAKLAMALAR.CS**

****

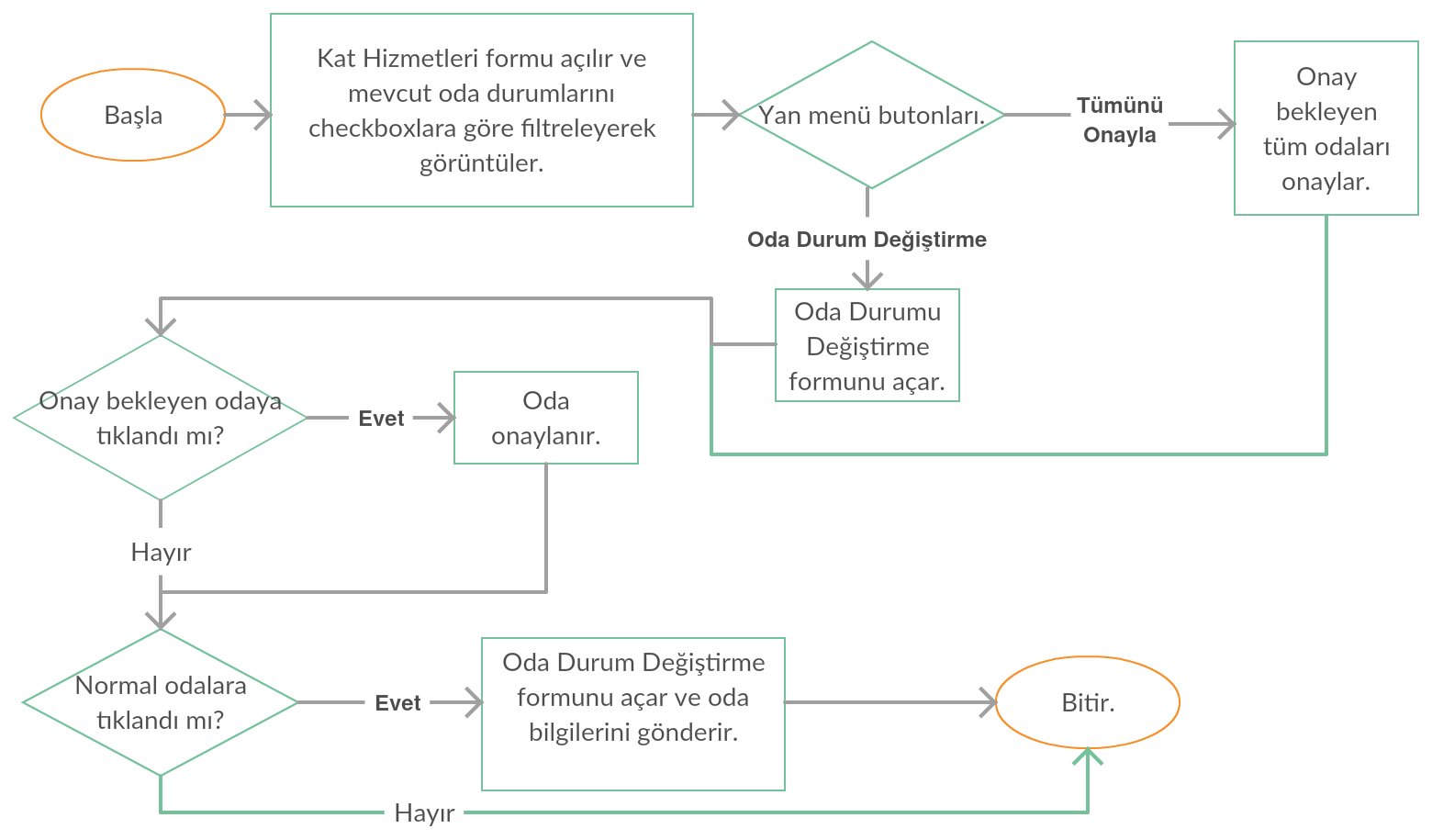
**ODASİPARİS.CS**

****

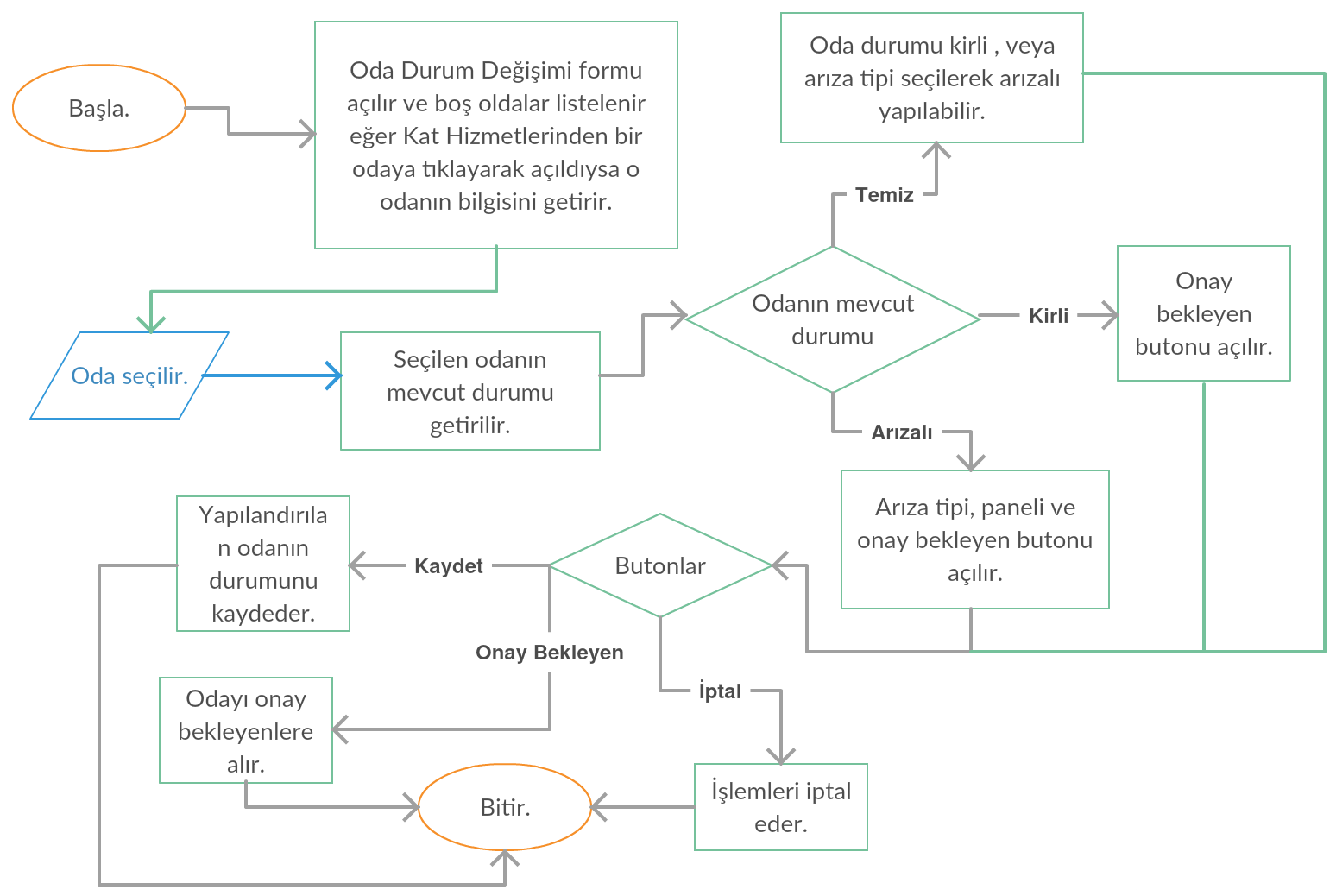
**URUNEKLE.CS**

****

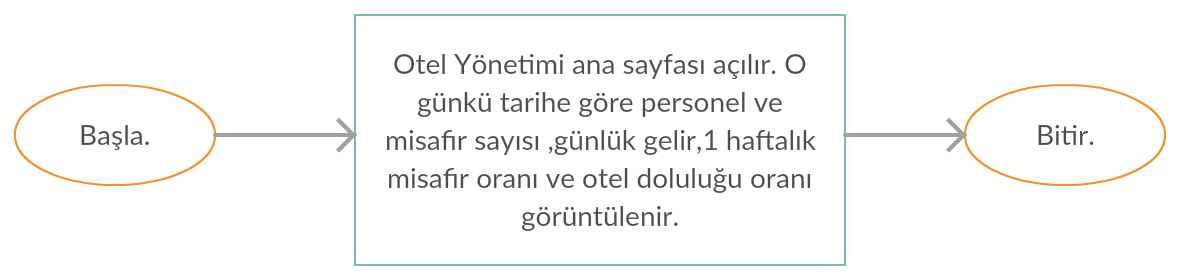
**KATHİZMETLERİ.CS**

****

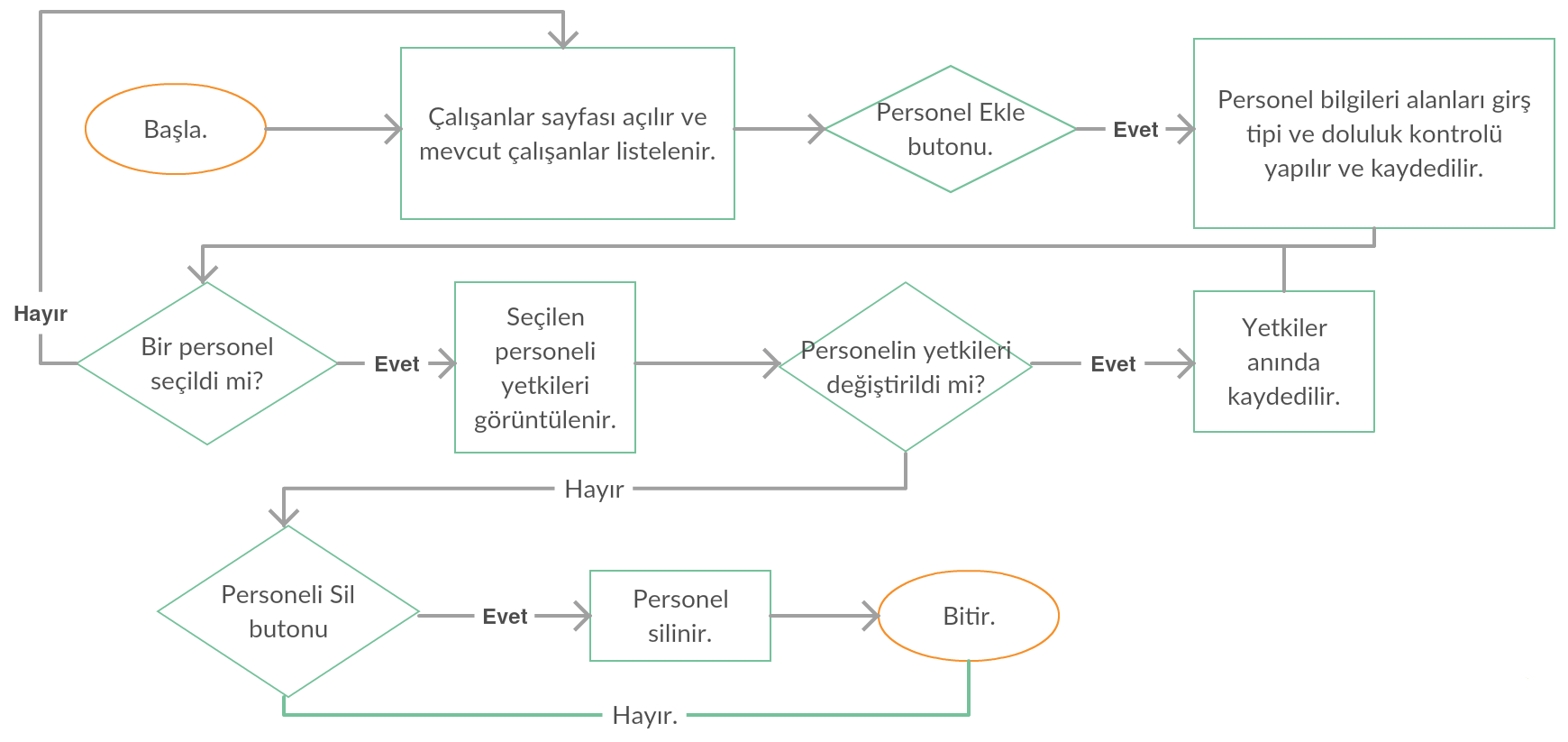
**ODADURUM.CS**

****

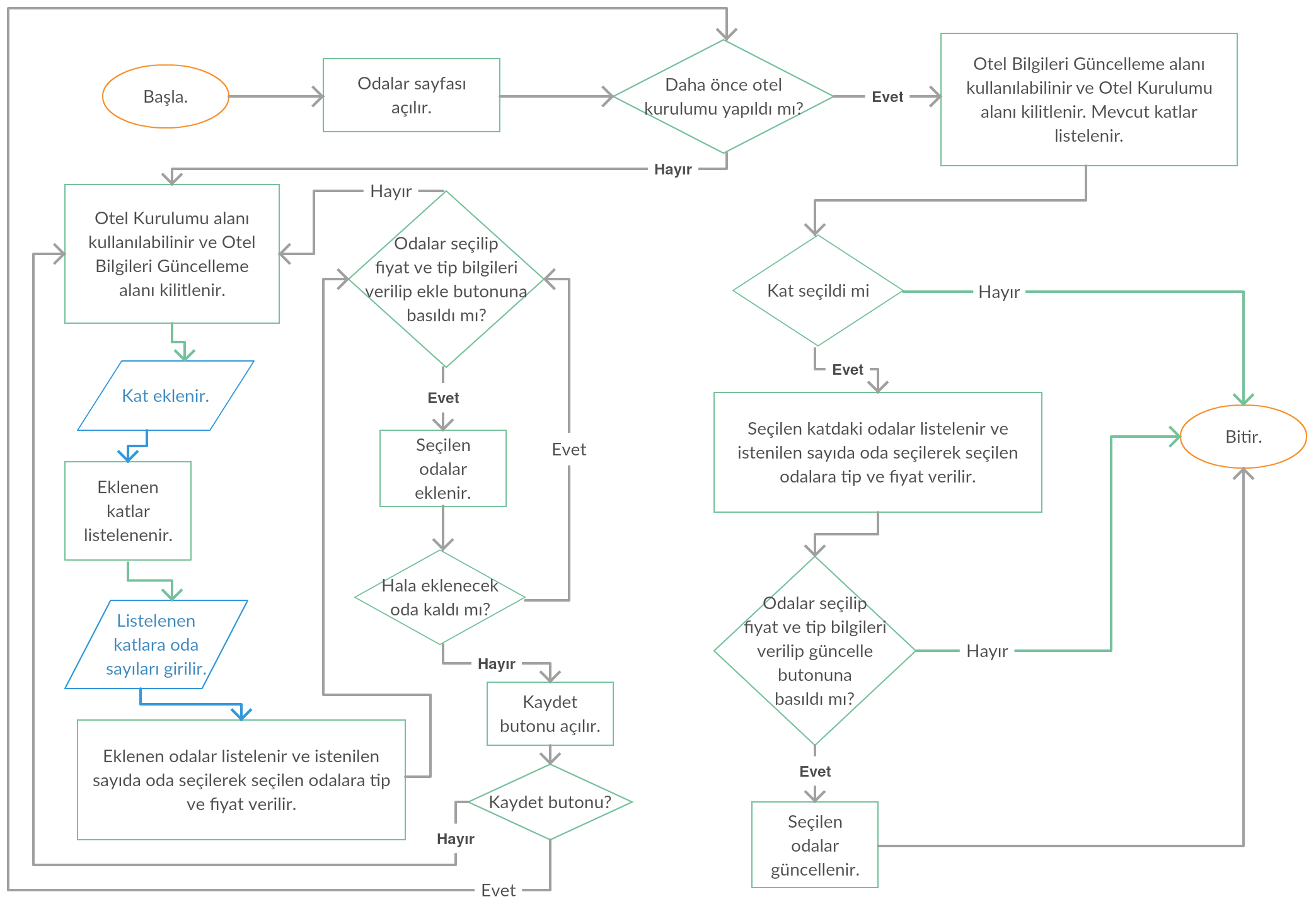
**YÖNETİCİPANEL.CS**

****

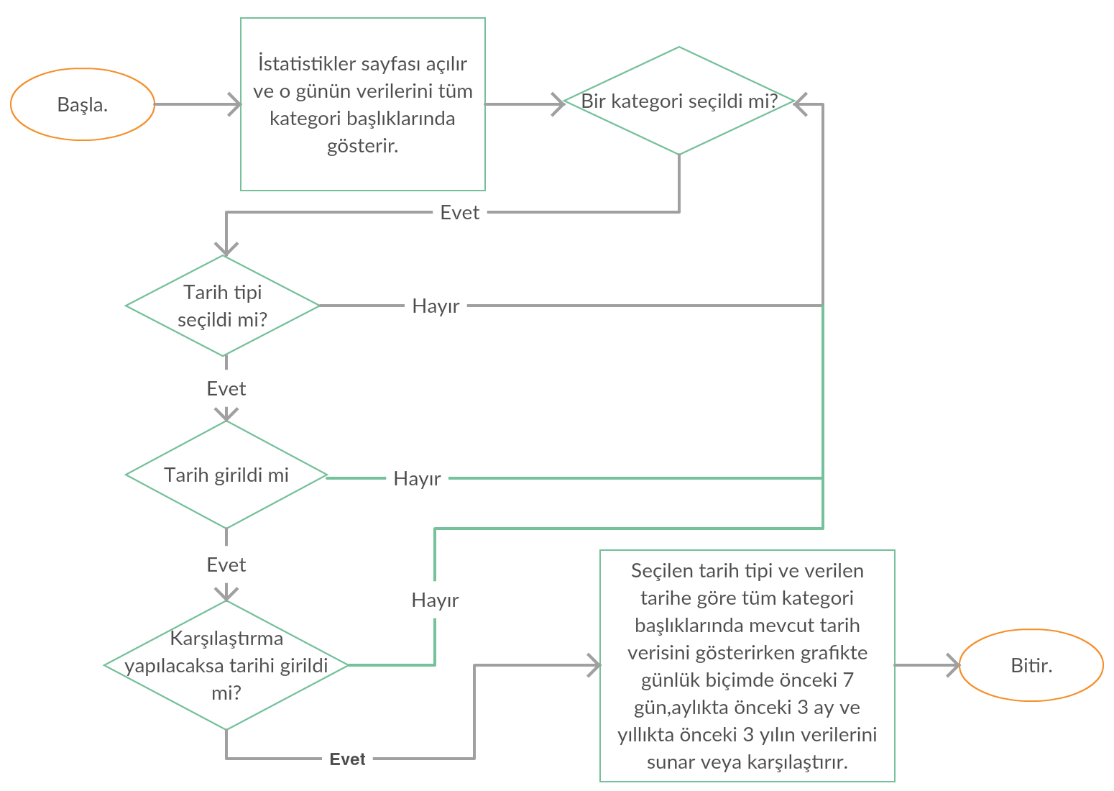
**PERSONEL.CS**

****

**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**

admin

**Çıkış Geçmiş**



**Giriş Geçmiş**



**Hatırlatmalar**

hatırlatma

**İçecekler**



**Yiyecekler**



**İlçe**



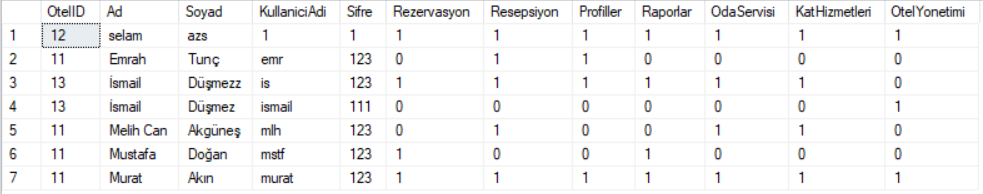
**Kişi Çıkış**



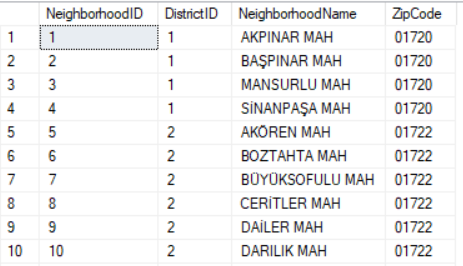
**Kişi Giriş**



**Kullanıcılar**



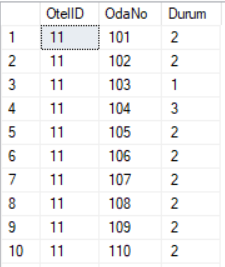
**Mahalle**



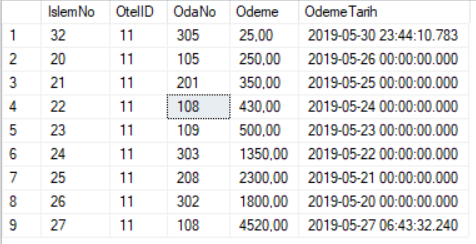
**Odalar**



**OdalarYenileme**



**Ödemeler**



**ÖdemelerGeçmiş**

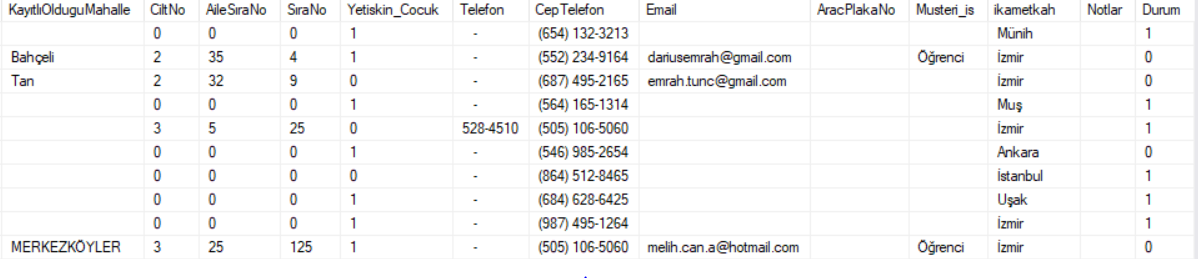


**Oteller**



**R\_musteriekle**





**Rez\_Rapor**



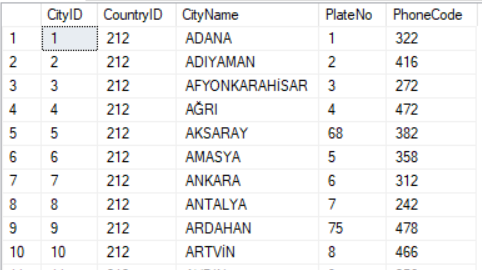
**SiparisGecmis**



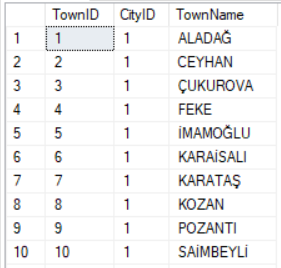
**Siparisler**



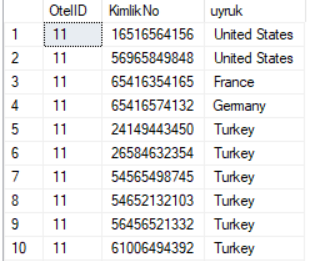
**Şehir**



**Kasaba**



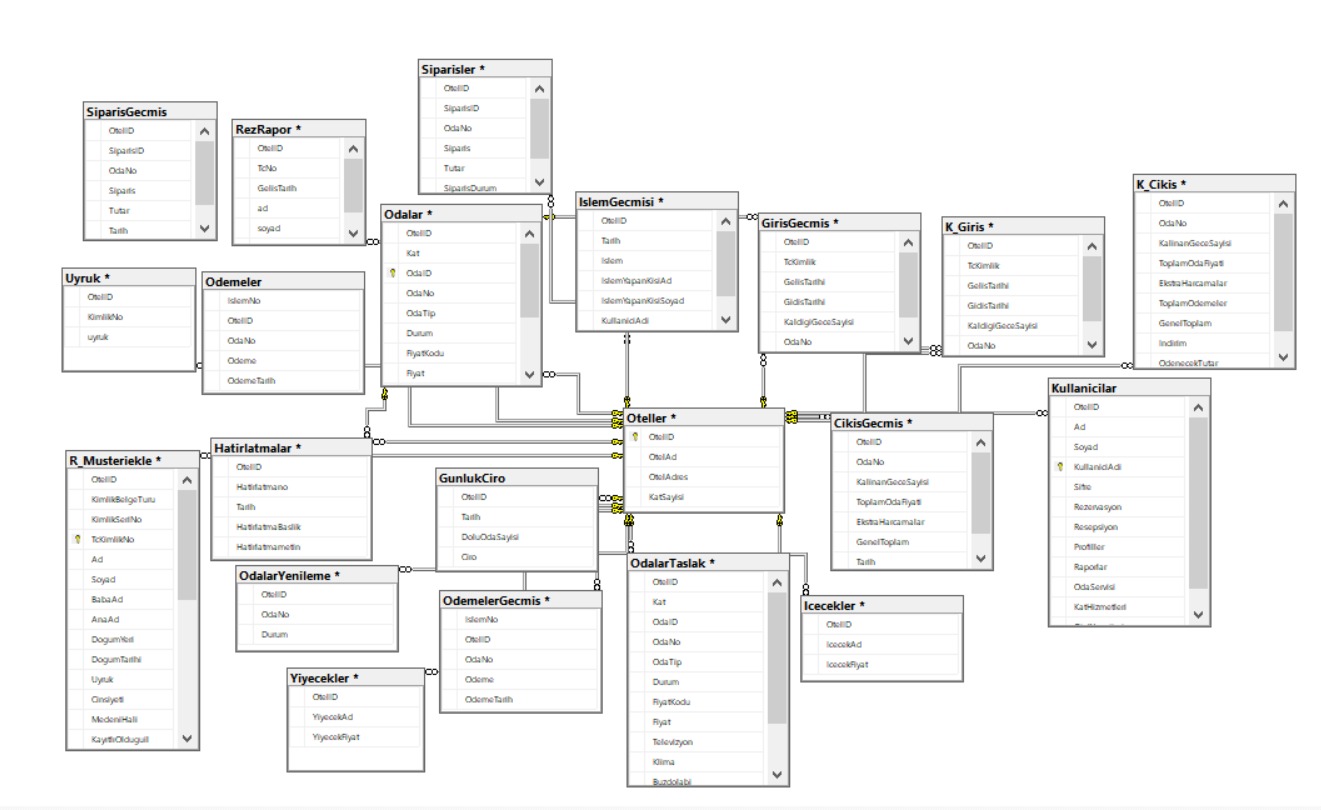
**Uyruk**



**Ülke**



**ENTİTY RELATİONSHİP DİYAGRAMI**

****

**PROGRAM KODLARI**

**GİRİS.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 Kullanicilar where KullaniciAdi='" + bunifuMetroTextbox1.Text + "' and Sifre='" + bunifuMetroTextbox2.Text + "'", con);

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

OtelArayüz oa = new OtelArayüz();

Resepsiyon resepsiyon = new Resepsiyon();

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["Resepsiyon"].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.btnotelyonetimi.Enabled = true;

}

else

{

oa.btnotelyonetimi.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\_OnValueChanged(object sender, EventArgs e)

{

}

}

}

**ADMİN.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\_OnValueChanged(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 + "','" + 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 iTotalWidth = 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 OtelArayüz()

{

InitializeComponent();

}

private void OtelArayüz\_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.ToLongTimeString();

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\_1(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.bunifuCustomDataGrid1.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.bunifuCustomDataGrid1.ColumnCount; j++)

{

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

myRange.Value2 = form.bunifuCustomDataGrid1.Columns[j].HeaderText;

myRange.ColumnWidth = 15;

}

StartRow++;

for (int i = 0; i < form.bunifuCustomDataGrid1.Rows.Count; i++)

{

for (int j = 0; j < form.bunifuCustomDataGrid1.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-yazdrma?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\_musterı\_ekle musterı\_Ekle = new r\_musterı\_ekle();

musterı\_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;

iTotalWidth = 0;

foreach (DataGridViewColumn dgvGridCol in form.bunifuCustomDataGrid1.Columns)

{

iTotalWidth += 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").InnerXml;

string EURO\_Satis = xmlDoc.SelectSingleNode("Tarih\_Date/Currency[@Kod='EUR']/BanknoteSelling").InnerXml;

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.ToLongDateString() + " " + DateTime.Now.ToShortTimeString();

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\_musterı\_ekle f = new r\_musterı\_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.ToLongTimeString();

bunifuFlatButton6.Text = " - " + DateTime.Now.Day.ToString() + "." + DateTime.Now.Month.ToString() + "." + DateTime.Now.Year.ToString();

// 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\_musterı\_ekle : Form

{

public int OtelID = Login.id;

public r\_musterı\_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 + "',KayıtlıOlduguil='" + il.Text + "',KayıtlıOlduguilce='" + ilce.Text + "',KayıtlıOlduguMahalle='" + mah.Text + "',CiltNo='" + cilt\_no.Text + "',AileSıraNo='" + aile\_sıra\_no.Text + "',SıraNo='" + int.Parse(sıra\_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);

}

}

else

{

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, KayıtlıOlduguil, KayıtlıOlduguilce, KayıtlıOlduguMahalle, CiltNo, AileSıraNo, SıraNo,Yetiskin\_Cocuk, Telefon, CepTelefon, Email, AracPlakaNo, Musteri\_is, ikametkah, Notlar,Durum) values('" + OtelID + "','" + kim\_bel\_tur.SelectedIndex + "','" + kimlik\_seri\_no.Text + "','" + Convert.ToUInt64(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\_sıra\_no.Text + "','" + sıra\_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\_sıra\_no.Text = "";

sıra\_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\_musterı\_ekle\_Load(object sender, EventArgs e)

{

dogum\_tar.MaxDate = DateTime.Now;

giris\_tar.MinDate = 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.ToUInt64(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\_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 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 iTotalWidth = 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.ToLongDateString() + " " + DateTime.Now.ToShortTimeString();

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;

iTotalWidth = 0;

foreach (DataGridViewColumn dgvGridCol in rapor.Columns)

{

iTotalWidth += 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='" + OtelId + "'", con);

SqlCommand cmd2 = new SqlCommand("Select Count(Ad) as sayi from Odaicerik where OdaNo='" + splitodano + "' and OtelID='" + OtelId + "'", con2);

SqlCommand cmd3 = new SqlCommand("Select OdaNo,OdaTip,Durum from Odalar where OdaNo='" + splitodano + "' and OtelID='" + OtelId + "'", 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 alıyoruz

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);

}

}

}

}

**ODAİCERİK.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 çıkışı 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 kaldigigece = 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,KaldigiGeceSayisi,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;

kaldigigece += 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,@toplamodafiyat,@extra,@toplamodeme,@geneltoplam,@indirim,@odenecektutar)", con2);

insert.Parameters.AddWithValue("@otelid", OtelID);

insert.Parameters.AddWithValue("@odano", cmbodano.Text);

insert.Parameters.AddWithValue("@gece", kaldigigece);

double toplamoda = odafiyat \* kaldigigece;

insert.Parameters.AddWithValue("@toplamodafiyat", toplamoda);

insert.Parameters.AddWithValue("@extra", double.Parse(extraharcamalartxt.Text));

insert.Parameters.AddWithValue("@toplamodeme", 0);

insert.Parameters.AddWithValue("@geneltoplam", ((Convert.ToDouble(odafiyat) \* Convert.ToDouble(kaldigigece))) + Convert.ToDouble(extraharcamalartxt.Text));

insert.Parameters.AddWithValue("@indirim", 0);

insert.Parameters.AddWithValue("@odenecektutar", (((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,ToplamOdemeler=@toplamodeme,GenelToplam=@geneltoplam,OdenecekTutar=@odenecektutar 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("@toplamodeme", double.Parse(toplamodemeler.Text));

update.Parameters.AddWithValue("@geneltoplam", (Convert.ToDouble(odafiyat) \* Convert.ToDouble(kaldigigece)) + Convert.ToDouble(extraharcamalartxt.Text));

update.Parameters.AddWithValue("@odenecektutar", ((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 datetpGelis\_ValueChanged(object sender, EventArgs e) //kalınan gece sayısı hesaplandı

{

DatetpGidis.MinDate = datetpGelis.Value;

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 = 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 kadiklerı 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 tcsiyle 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();

bunifuCustomDataGrid1.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];

bunifuCustomDataGrid1.Rows.Add();

bunifuCustomDataGrid1.Rows[b].Cells[0].Value = adet;

bunifuCustomDataGrid1.Rows[b].Cells[1].Value = urunad;

bunifuCustomDataGrid1.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 Tutardan Fazla Olamaz", "Bilgilendirme", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

private void yenile()

{

SqlCommand odenecektutar = 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 = odenecektutar.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 geneltoplamoku = komut.ExecuteReader();

double geneltoplam = 0;

while (geneltoplamoku.Read())

{

geneltoplam = double.Parse(geneltoplamoku[0].ToString());

}

con.Close();

con.Open();

SqlCommand cikisupdate = new SqlCommand("update K\_Cikis set ToplamOdemeler=@toplamodeme,OdenecekTutar=@odenecektutar where OdaNo=@odano and OtelID=@otelid", con);

cikisupdate.Parameters.AddWithValue("@otelid", OtelID);

cikisupdate.Parameters.AddWithValue("@odano", cmbodano.Text);

cikisupdate.Parameters.AddWithValue("@toplamodeme", toplamodemeupdate);

cikisupdate.Parameters.AddWithValue("@odenecektutar", geneltoplam - toplamodemeupdate);

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)

{

toplamodemeler.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 cmbfiyatkod\_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 (cmbfiyatkod.SelectedIndex == 0)

{

cmbfiyatkod.SelectedIndex = 0;

textBox1.Text = Convert.ToDouble(dr[0]).ToString();

}

else if (cmbfiyatkod.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\_TİKLA.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);

bunifuCustomDataGrid1.Rows.Clear();

int a = 0;

con.Open();

SqlDataReader sdr = cmd.ExecuteReader();

while (sdr.Read())

{

bunifuCustomDataGrid1.Rows.Add();

bunifuCustomDataGrid1.Rows[a].Cells[0].Value = sdr["TcNo"].ToString();

bunifuCustomDataGrid1.Rows[a].Cells[1].Value = sdr["ad"].ToString();

bunifuCustomDataGrid1.Rows[a].Cells[2].Value = sdr["soyad"].ToString();

bunifuCustomDataGrid1.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\_OnValueChanged(object sender, EventArgs e)

{

gridyenile();

}

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

{

r\_musterı\_ekle ekle = new r\_musterı\_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();

}

}

}

**HATİRLATMAEKLE.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\_OnValueChanged(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=="")

{

bunifuCustomDataGrid1.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 == "")

{

bunifuCustomDataGrid1.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,ikametkah,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();

tclabel.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.bunifuCustomDataGrid1.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);

//profiller diye sqlde view oluşturdum onu excel için kullanacağız

DataSet ds = new DataSet();

da.Fill(ds);

bunifuCustomDataGrid1.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,DogumTarihi=@dt,Uyruk=@uyruk,CepTelefon=@cep,Email=@email,Musteri\_is=@is,Notlar=@nott,ikametkah=@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 bunifuCustomDataGrid1\_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\_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 bunifuToggleSwitch2\_OnValuechange(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\_OnValueChanged(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\_ayrilan.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\_ayrilan.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 iTotalWidth = 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.ToLongDateString() + " " + DateTime.Now.ToShortTimeString();

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;

iTotalWidth = 0;

foreach (DataGridViewColumn dgvGridCol in rapor.Columns)

{

iTotalWidth += dgvGridCol.Width;

}

}

catch (Exception ex)

{

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

}

}

private void Gel\_tar2\_ValueChanged(object sender, EventArgs e)

{

}

}

}

**ODASİPARİS.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 içn 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[satır].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[satır].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[satır].Cells[1].Value + "' and OtelID='" + otelid + "'", baglan);

update.Parameters.AddWithValue("@extra", tutar);

update.Parameters.AddWithValue("@genel", odafiyat+tutar);

string isaretcikar = bunifuCustomDataGrid1.Rows[satır].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[satır].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 yiyecekekle = new SqlCommand("insert into Yiyecekler values('" + otelid + "','" + TxtYiyecekAd.Text + "','" + tutargir + "')", baglan2);

baglan2.Open();

yiyecekekle.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 += ayril;

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();

SqlDataAdapter komut = new SqlDataAdapter("select Ad as PersonelAd,Soyad as PersonelSoyad,KullaniciAdi as KullanıcıAdı from Kullanicilar where OtelID='" + OtelID + "'", baglanti);

DataTable dt = new DataTable();

komut.Fill(dt);

personneller.DataSource = dt;

personneller.Columns[0].FillWeight = 100;

personneller.Columns[1].FillWeight = 100;

personneller.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\_OnValuechange(object sender, EventArgs e)

{

try

{

int a = personneller.CurrentRow.Index;

baglanti2.Open();

SqlCommand komut = new SqlCommand("update Kullanicilar set Rezervasyon='" + Convert.ToBoolean((rezsecim.Value)) + "' where Kullaniciadi='" + personneller.Rows[a].Cells[2].Value.ToString() + "' and OtelID='" + OtelID + "'", baglanti2);

komut.ExecuteNonQuery();

baglanti2.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Ressecim\_OnValuechange(object sender, EventArgs e)

{

try

{

int a = personneller.CurrentRow.Index;

baglanti2.Open();

SqlCommand komut = new SqlCommand("update Kullanicilar set Resepsiyon='" + Convert.ToBoolean((ressecim.Value)) + "' where Kullaniciadi='" + personneller.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 = personneller.CurrentRow.Index;

baglanti2.Open();

SqlCommand komut = new SqlCommand("update Kullanicilar set Profiller='" + Convert.ToBoolean((prosecim.Value)) + "' where Kullaniciadi='" + personneller.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 = personneller.CurrentRow.Index;

baglanti2.Open();

SqlCommand komut = new SqlCommand("update Kullanicilar set Raporlar='" + Convert.ToBoolean((raporsecim.Value)) + "' where Kullaniciadi='" + personneller.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 = personneller.CurrentRow.Index;

baglanti2.Open();

SqlCommand komut = new SqlCommand("update Kullanicilar set OdaServisi='" + Convert.ToBoolean((odasecim.Value)) + "' where Kullaniciadi='" + personneller.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 = personneller.CurrentRow.Index;

baglanti2.Open();

SqlCommand komut = new SqlCommand("update Kullanicilar set KatHizmetleri='" + Convert.ToBoolean((katsecim.Value)) + "' where Kullaniciadi='" + personneller.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 = personneller.CurrentRow.Index;

baglanti2.Open();

SqlCommand komut = new SqlCommand("delete from Kullanicilar where Kullaniciadi='" + personneller.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\_OnValueChanged(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 çıkılırsa 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 sayısı 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);

}

}

}

else

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.ToUInt32(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.ToUInt32(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.ToUInt32(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.ToUInt32(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.ToUInt32(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.ToUInt32(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 = guncelleodasecim.SelectedItem.ToString()+ " Nolu oda başarıyla güncellendi.";

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

if (guncelleodasecim.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 (odasecim.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 + "','" + 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-islemleri/

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(oku[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 RezervasyonSayısı");

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 (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;

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-islemleri/

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 gecmissure = 0;

if (RadioGun.Checked == true)

gecmissure = 7;

else if (RadioAy.Checked == true)

gecmissure = 3;

else

gecmissure = 3;

for (int i = 0; i < gecmissure; 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 gecmissure = 0;

if (RadioGun.Checked == true)

gecmissure = 7;

else if (RadioAy.Checked == true)

gecmissure = 3;

else

gecmissure = 3;

for (int i = 0; i < gecmissure; 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 gecmissure = 0;

if (RadioGun.Checked == true)

gecmissure = 7;

else if (RadioAy.Checked == true)

gecmissure = 3;

else

gecmissure = 3;

for (int i = 0; i < gecmissure; 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())

{

karsiurun += double.Parse(oku2[0].ToString());

}

baglan.Close();

chart1.Series["Grafik"].Points.Add(karsiurun);

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 = "TUNÇ";

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)

{

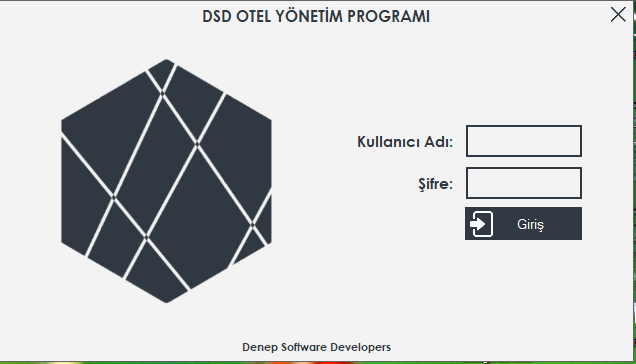
}

}

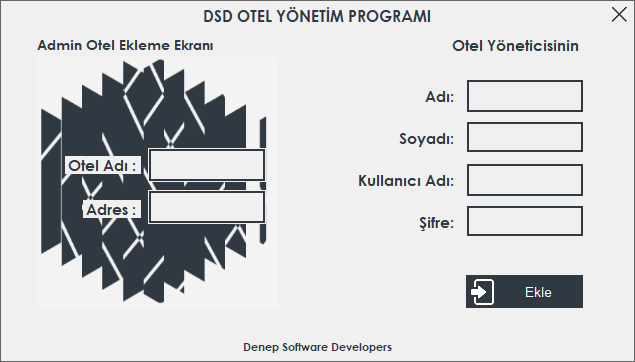
}

**ÖRNEK YAZICI ÇIKTILARI**

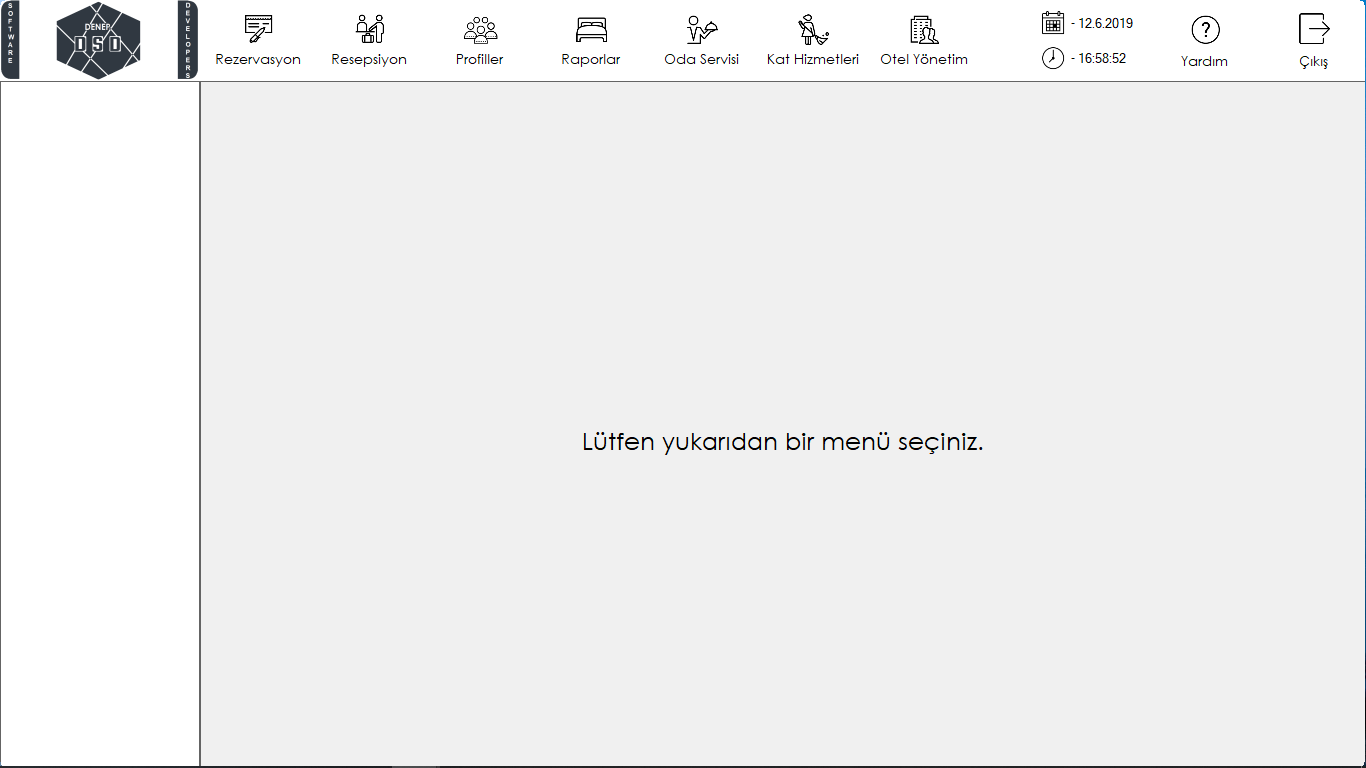
**LOGİN.CS**

****

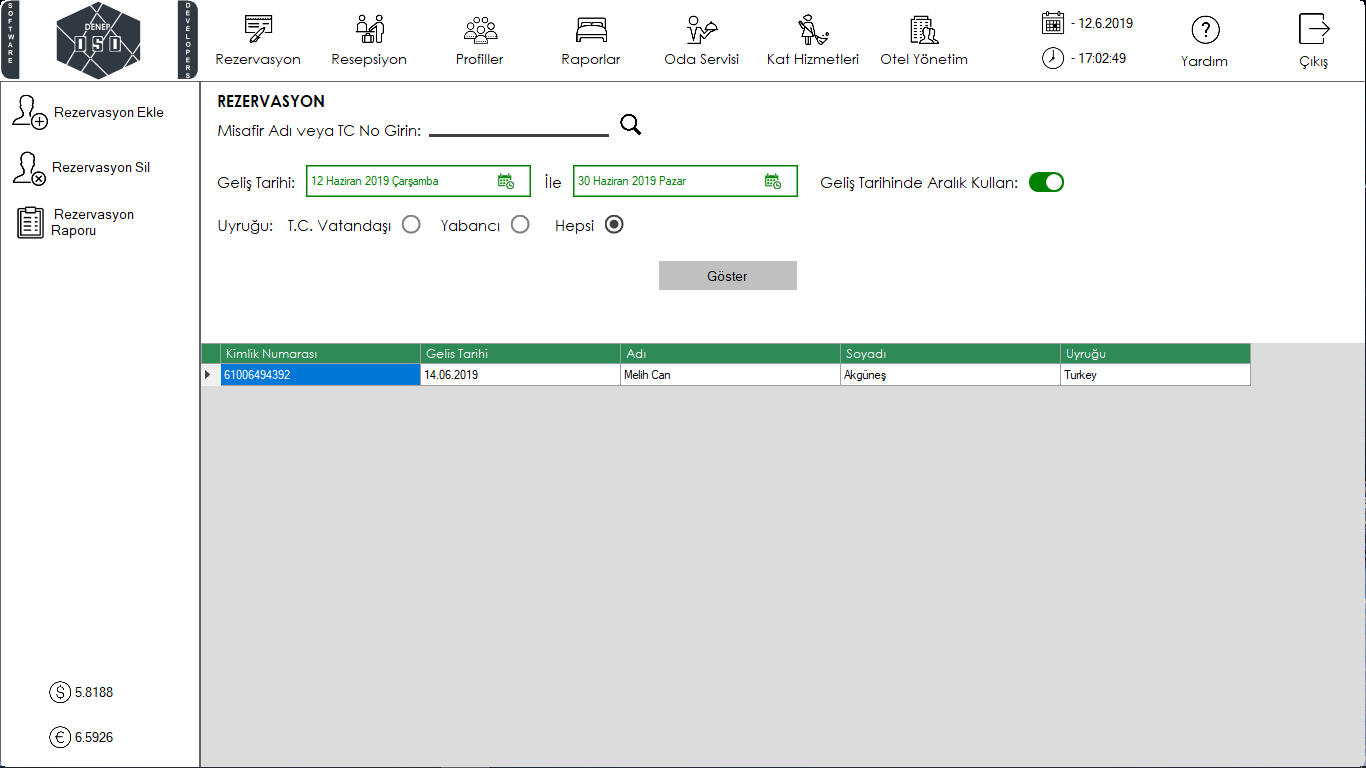
**ADMİN.CS**

****

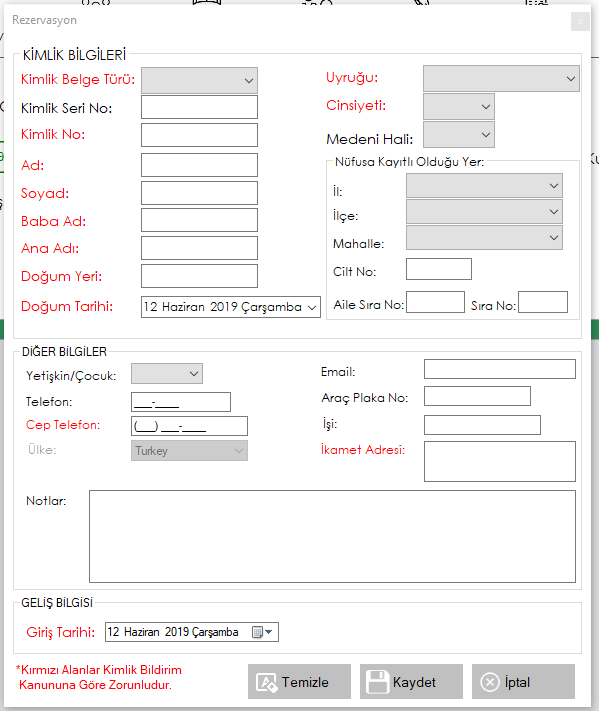
**OTELARAYÜZ.CS**

****

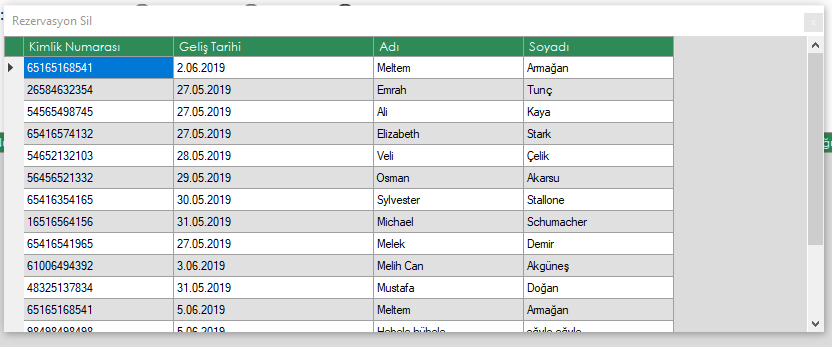
**REZERVASYON.CS**

****

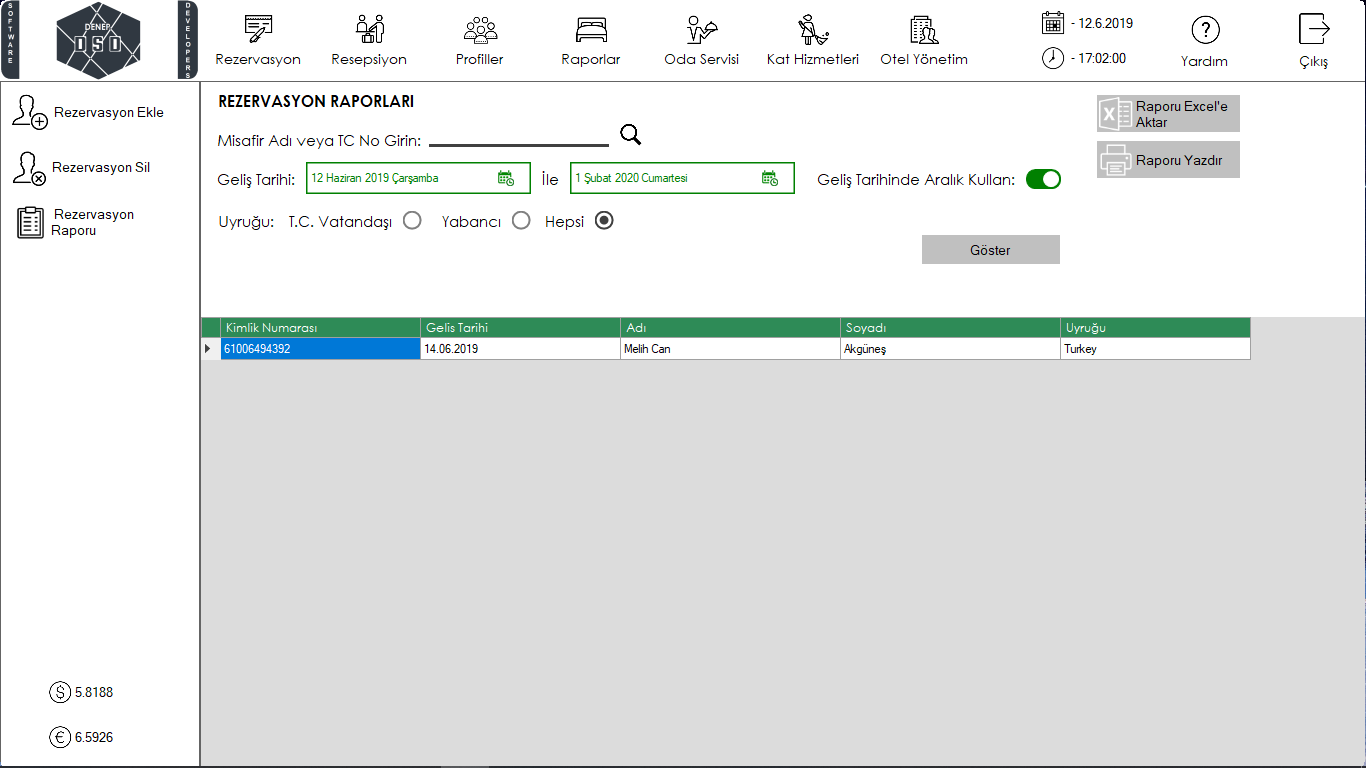
**R\_MUSTERİ\_EKLE.CS**

****

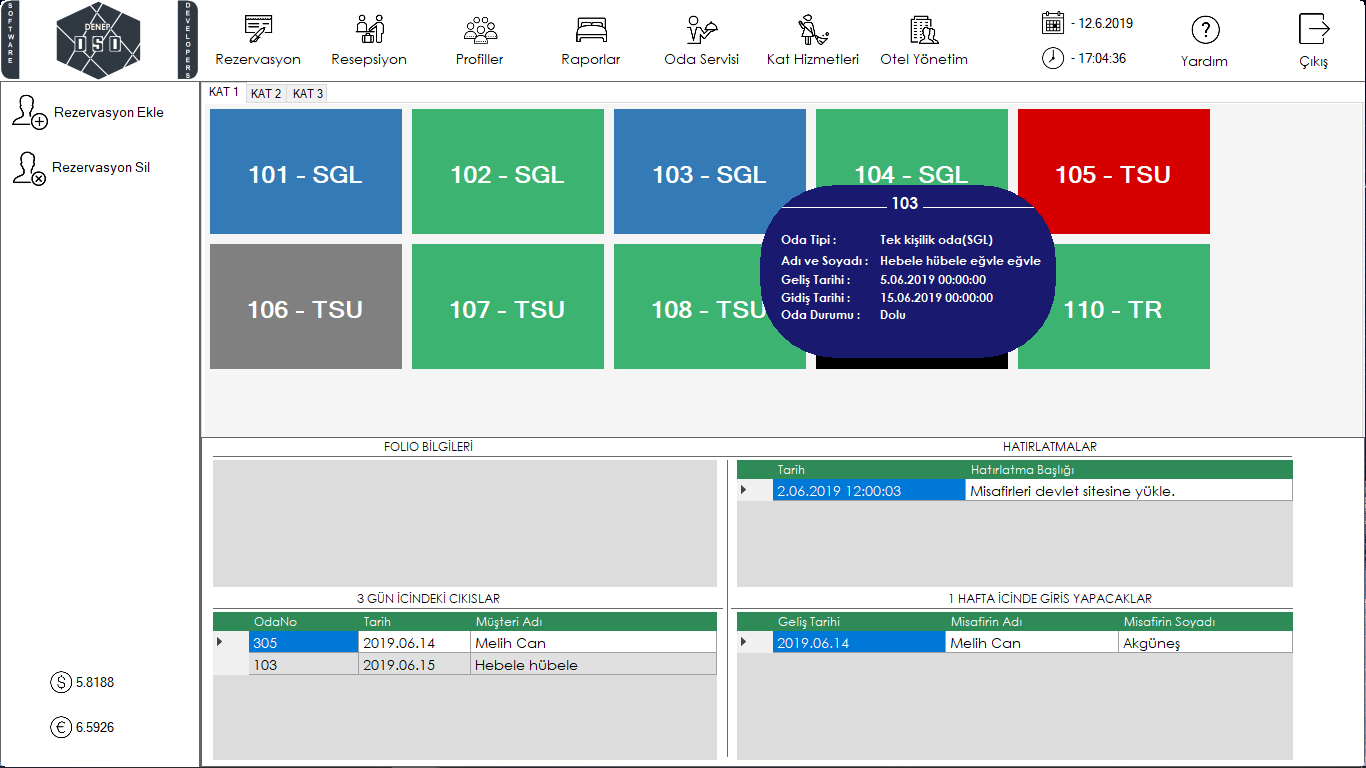
**REZ\_SİL.CS**

****

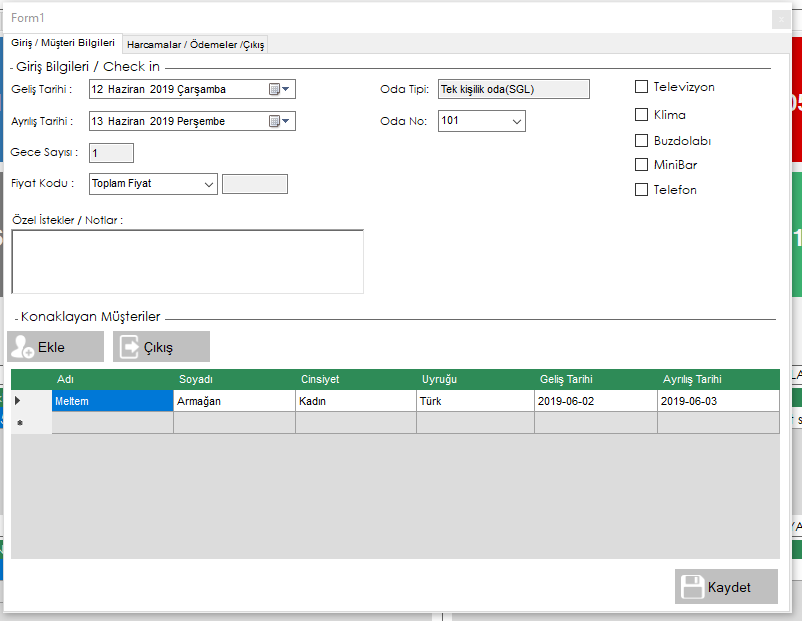
**REZRAPOR.CS**

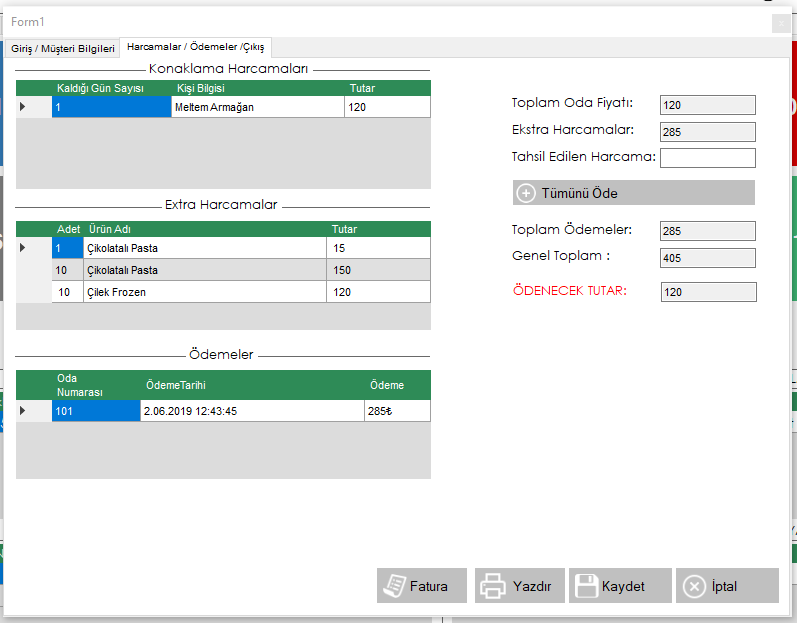
****

**RESEPSİYON.CS**

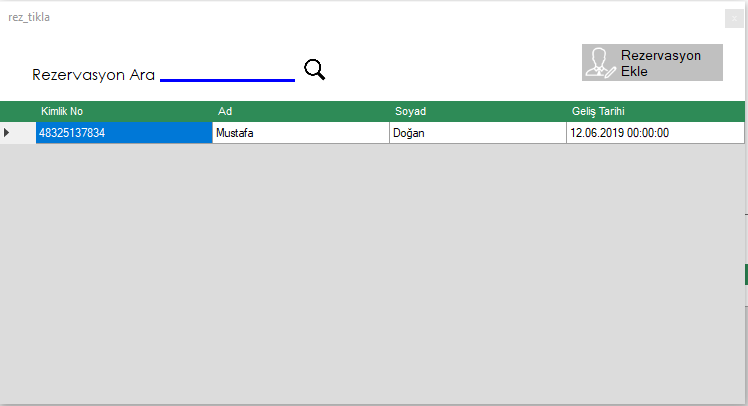
****

**R\_KONAK.CS**

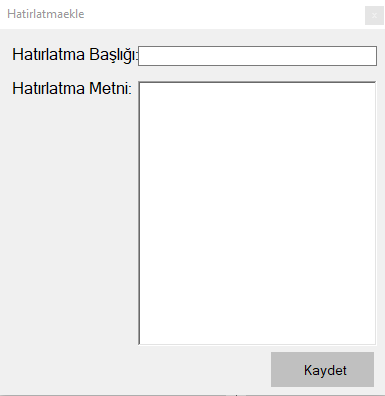
****

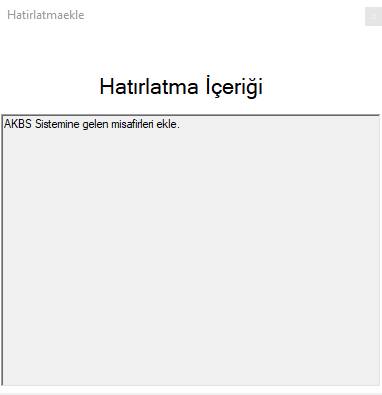
****

**REZ\_TİKLA.CS**

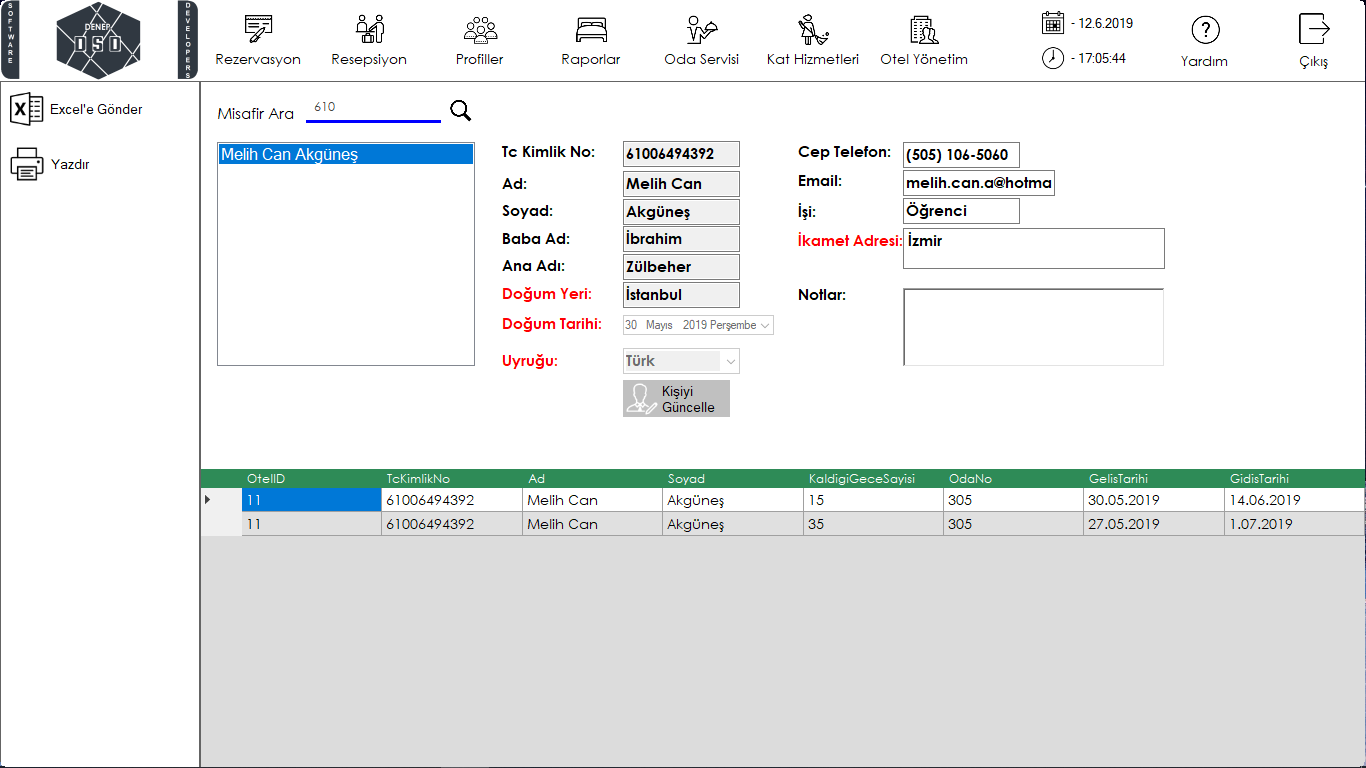
****

**HATİRLATMAEKLE.CS**

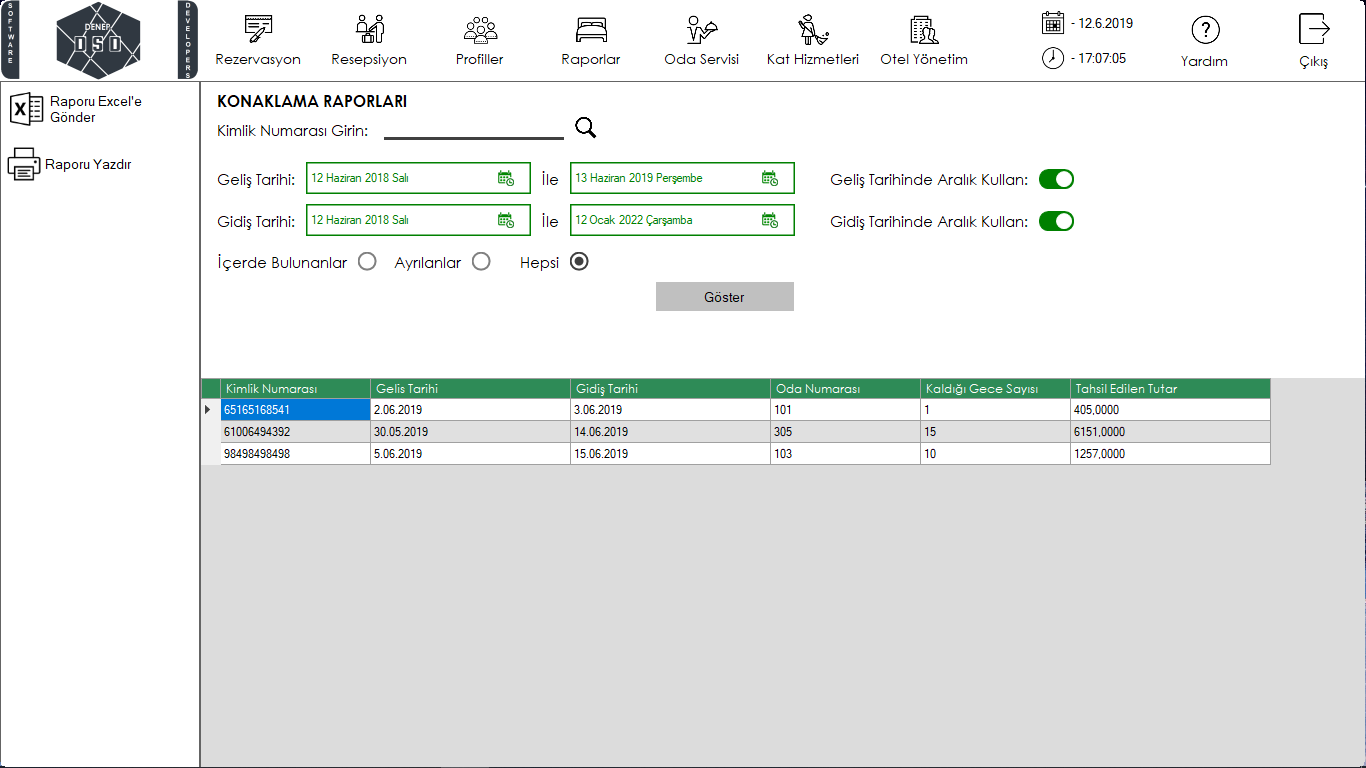
****

****

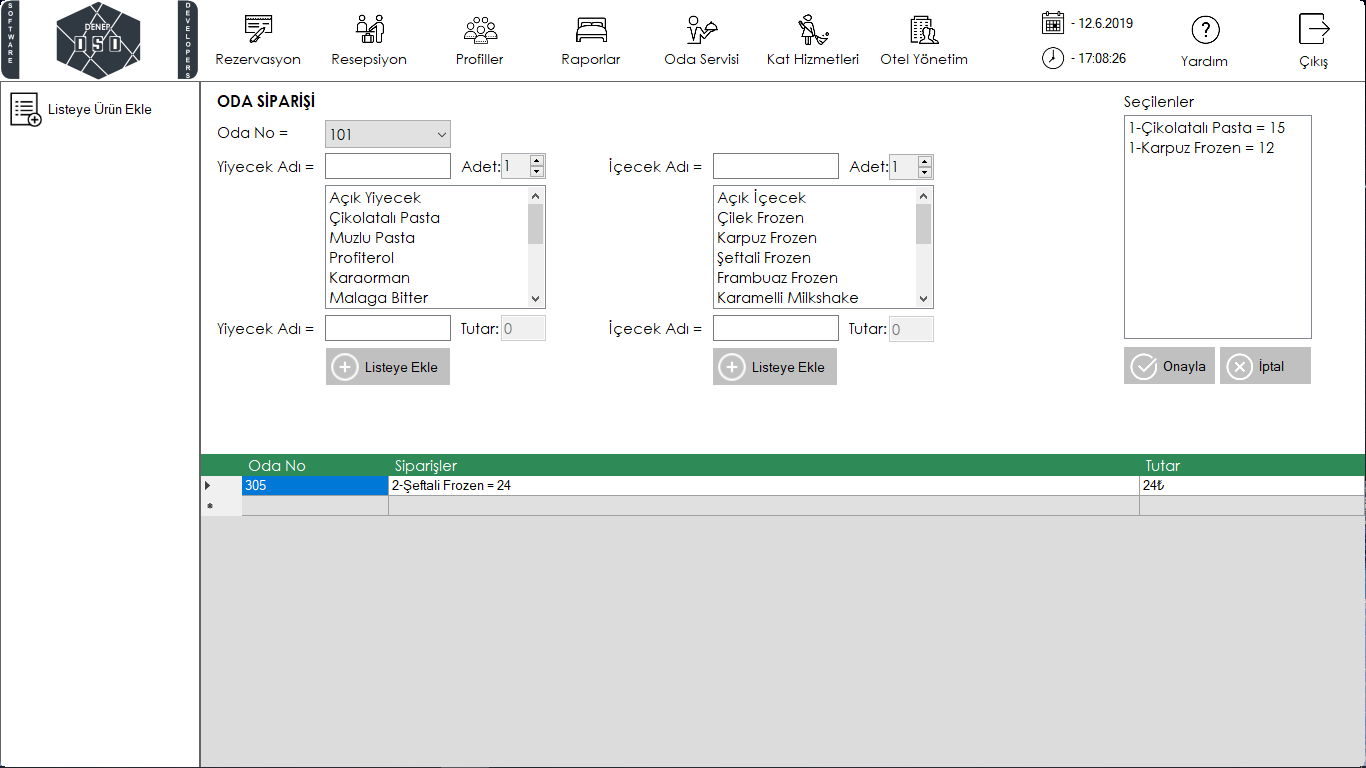
**PROFİLLER.CS**

****

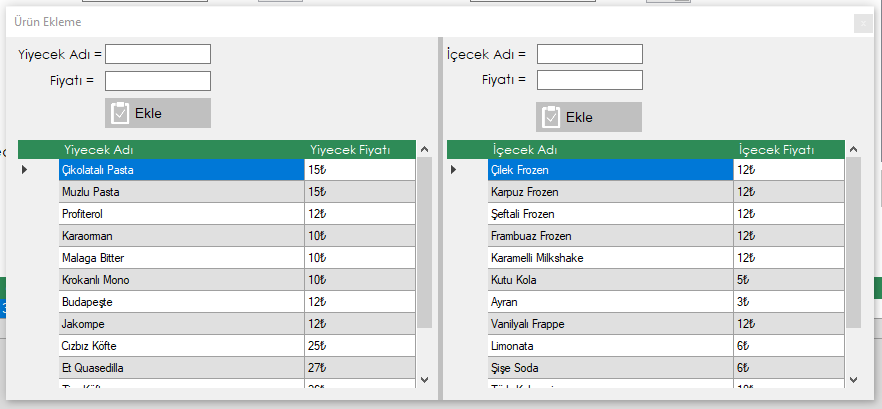
**KONAKLAMALAR.CS**

****

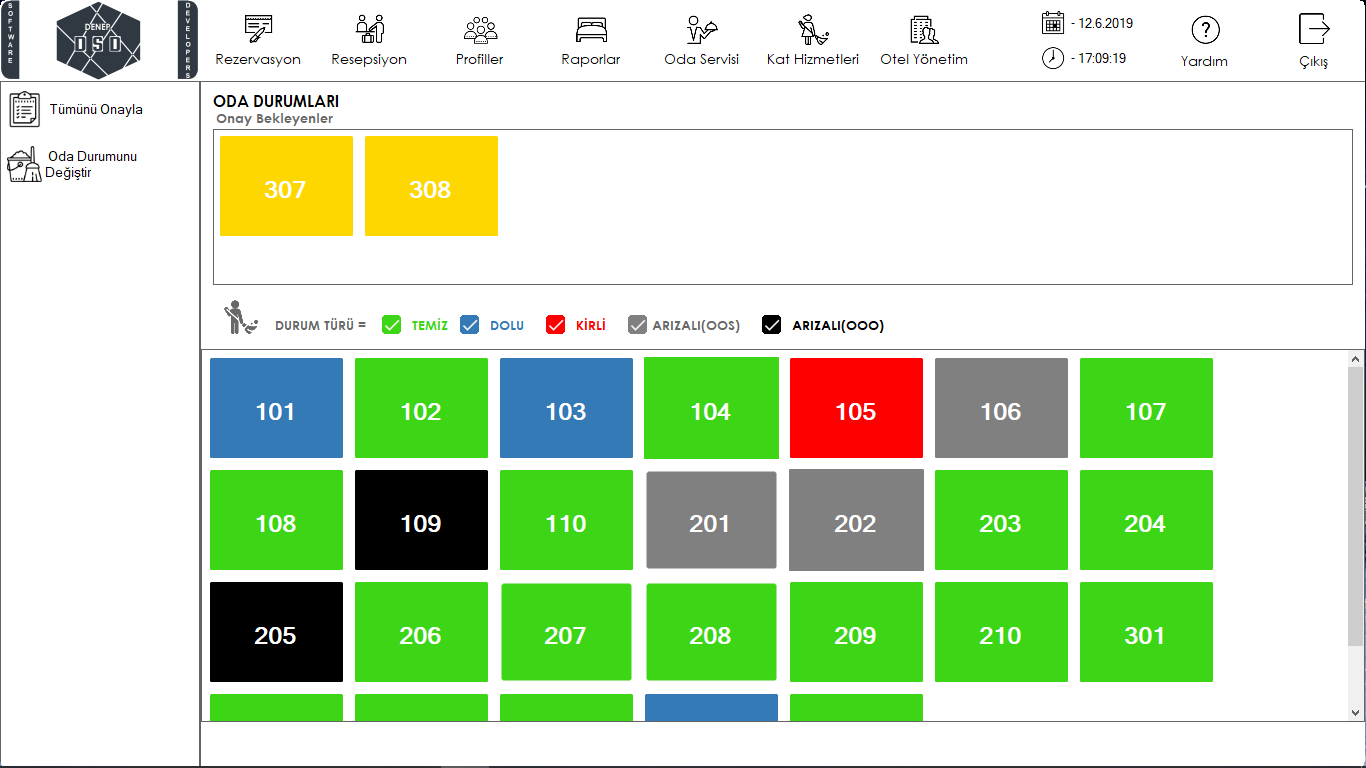
**ODASİPARİS.CS**

****

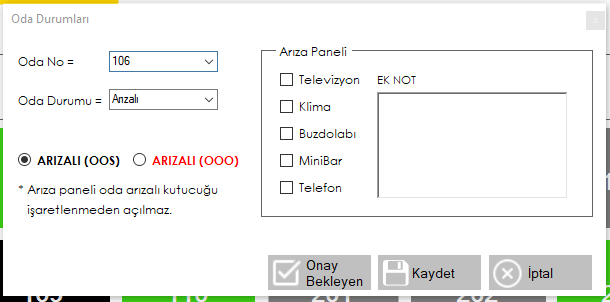
**URUNEKLE.CS**

****

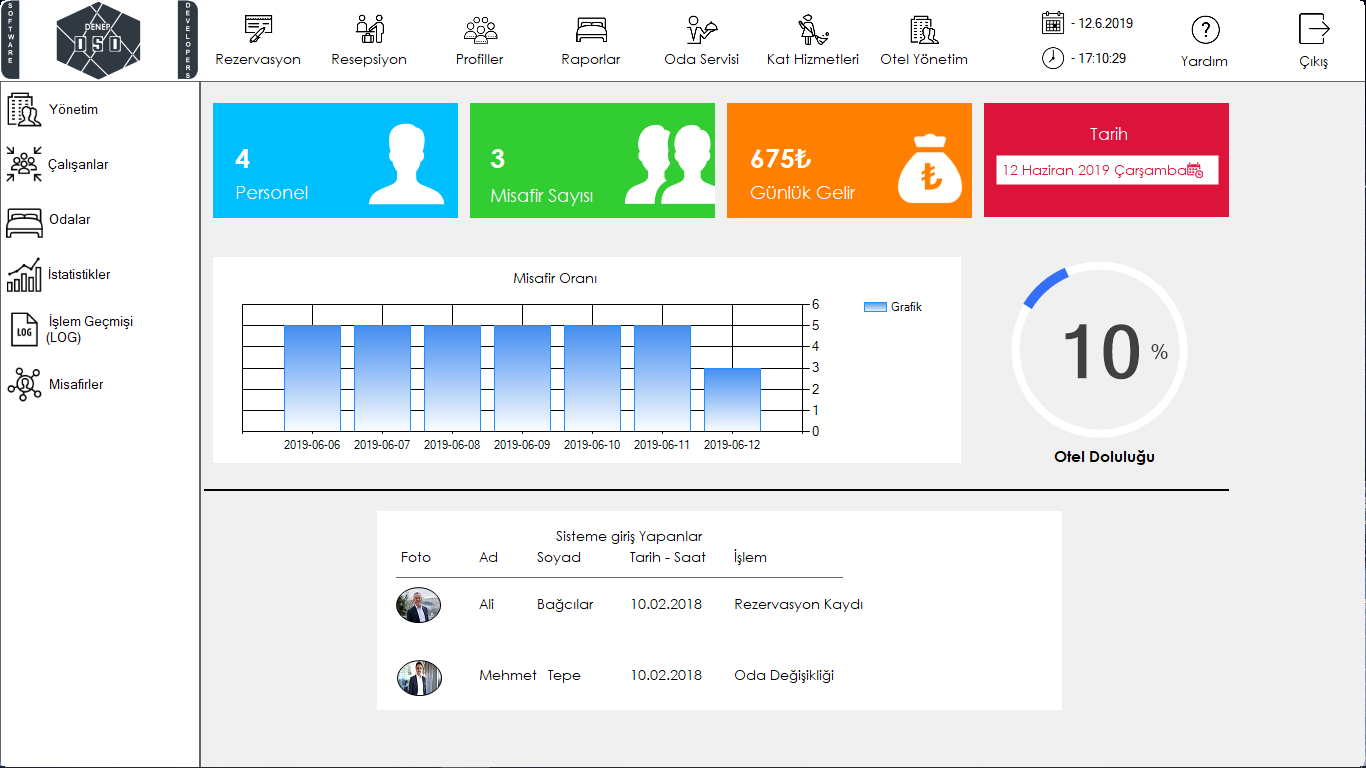
**KATHİZMETLERİ.CS**

****

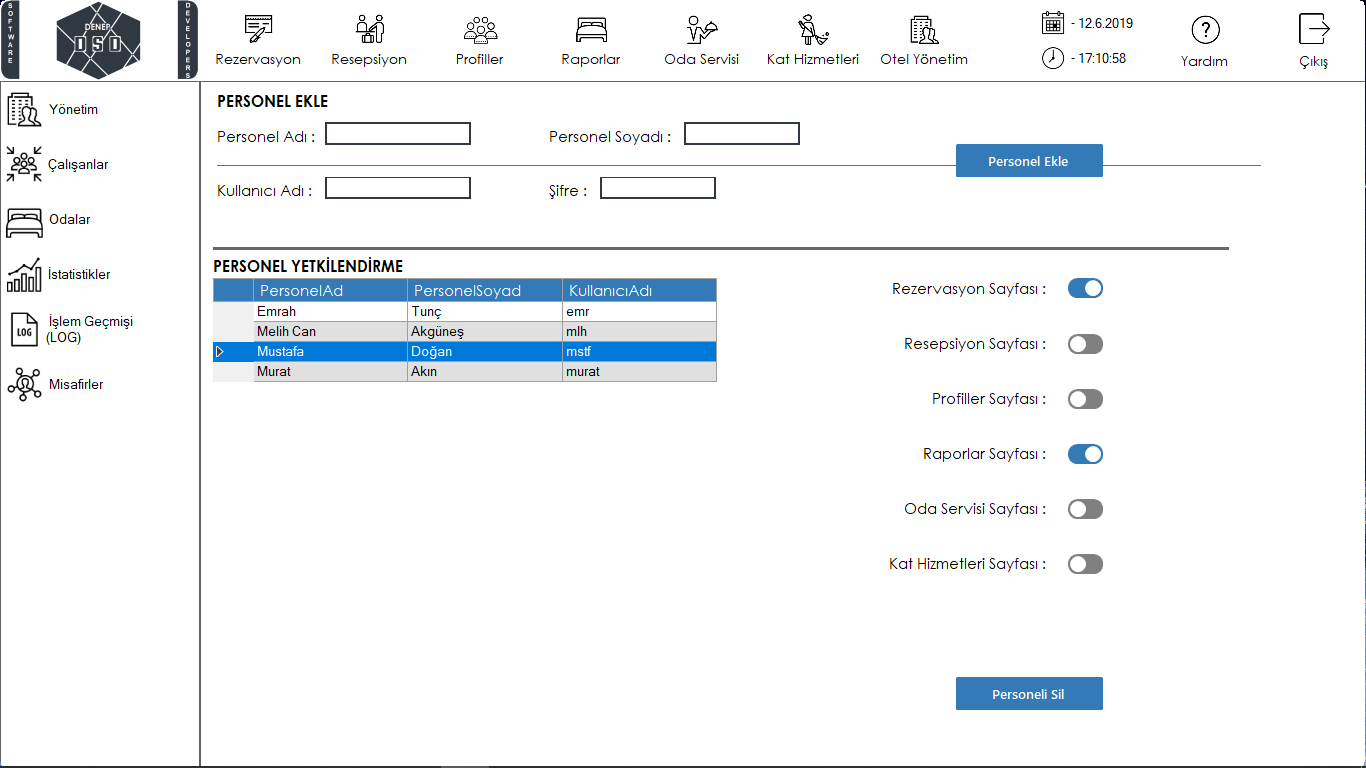
**ODADURUM.CS**

****

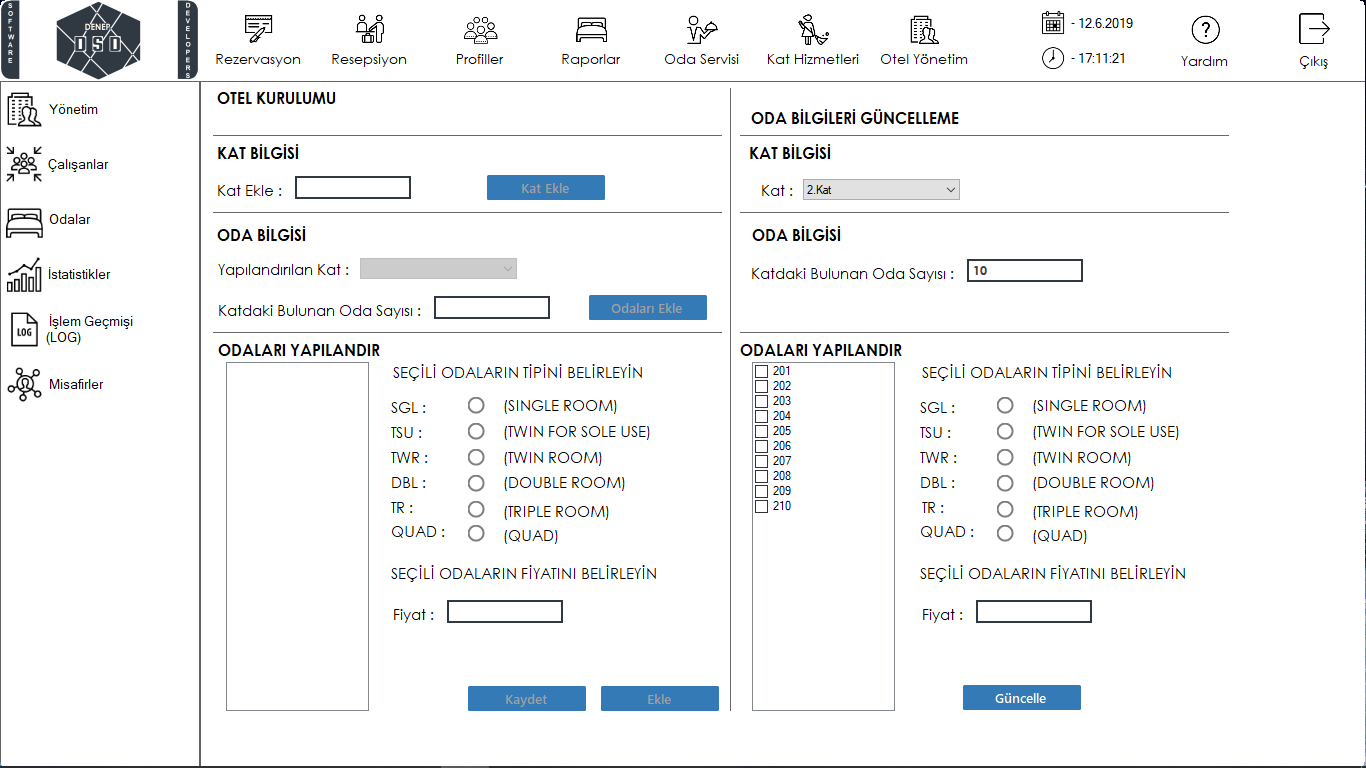
**YÖNETİCİPANELİ.CS**

****

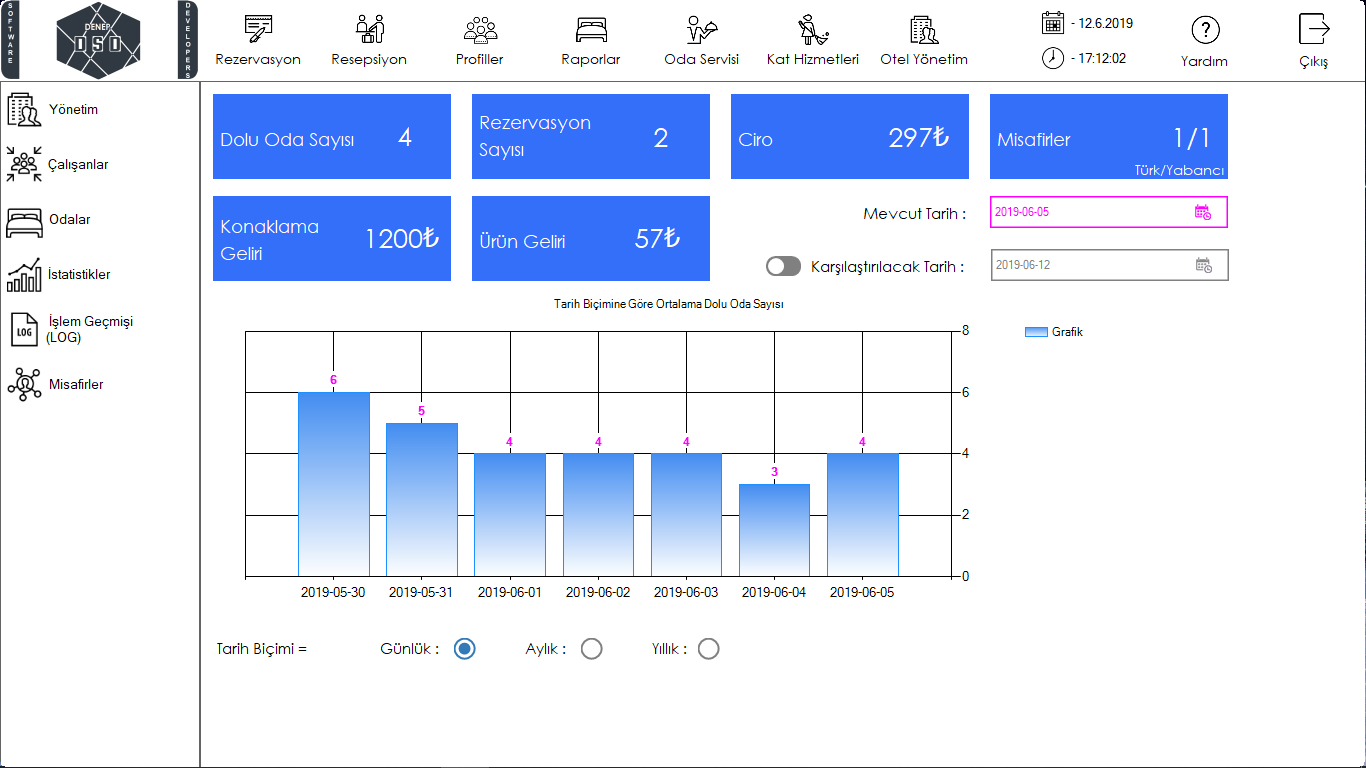
**PERSONEL.CS**

****

**ODALAR.CS**

****

**İSTATİSTİKLER.CS**

****

**SONSÖZ**

**NOTLAR**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_