**1. GİRİŞ**

**1.1.Proje Başlığı:** GÜRBÜZ OTEL OTOMASYONU

**1.2.Proje Amacı**: Otel personelleri ve müşterilerinin müşteri kaydı konusunda işini kolaylaştırmak.

**1.3.Projenin Yapılma Nedeni ve Yanıt Verdiği İhtiyaç:** Otel müşterilerinin bireysel olarak kayıt gerçekleştirememesi ve personellerin otelde kalan ya da giden müşterilerin kayıtlarına ulaşamaması, bu kayıtları düzenleyememesi.

**2.PROJE TANIMI:**

Otel müşterilerinin kayıt oluşturması, çalışanlarının ise admin girişi ile kullanıcı adı ve şifre kullanarak müşterilerin kaydını alması, müşterileri güncellemesi, silmesi bu kayıtları listelenmiş olarak görmesidir.

**3.PROJEDE KULLANILAN TEKNOLOJİLER VE UYGULAMALAR:**

Proje Visual Studio ,Windows Forms App(.NET Framework) ile oluşturulmuş C# dili ile yazılmış bir masaüstü uygulamasıdır. Veri Tabanı SQL üzerinden sağlanmıştır.

**4.PROJENİN ARAYÜZÜ VE KULLANIMI:**

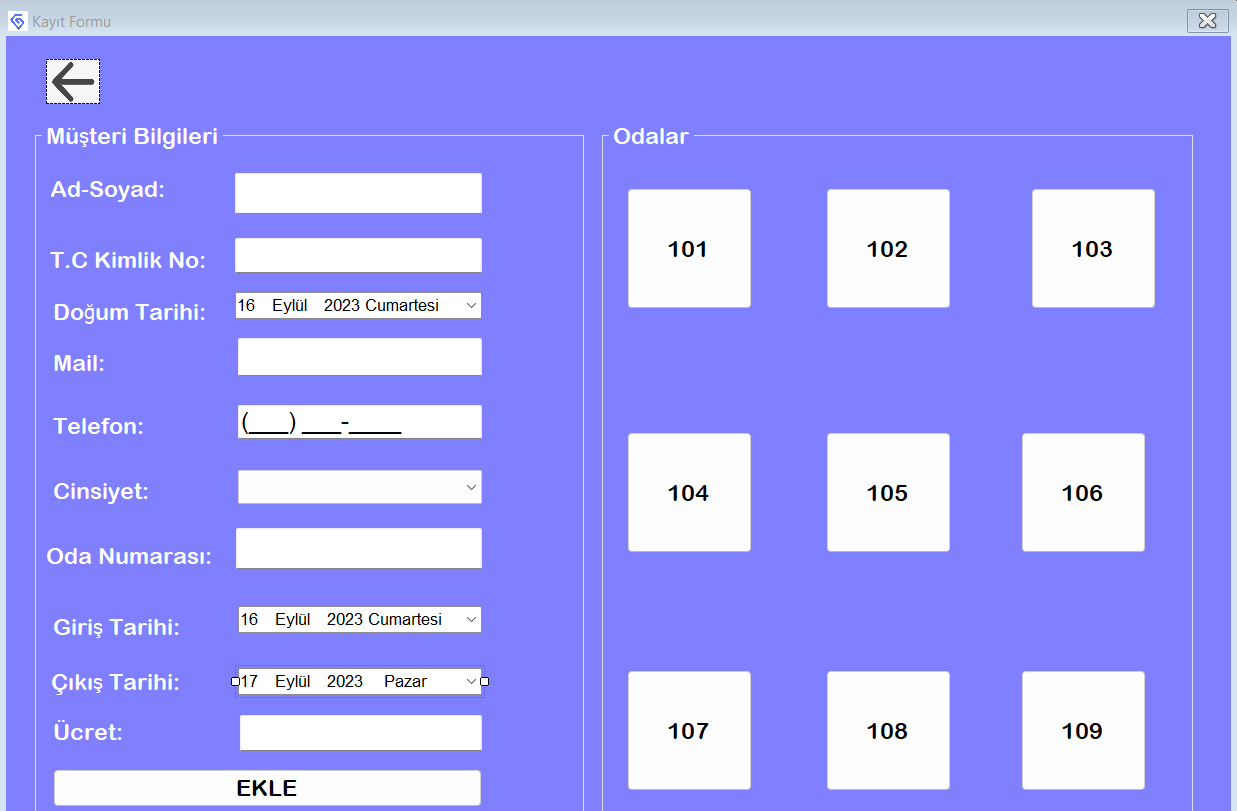
ANA SAYFA:

Proje ilk olarak Ana Sayfa görüntüsüyle açılır burada kullanıcı tipi belirlenir. Kullanıcı müşteri ise; Müşteri Kaydı butonuna tıklayarak müşteri kaydı sayfasına, admin ise; Admin Girişi butonuna tıklayarak Admin giriş sayfasına ulaşır.



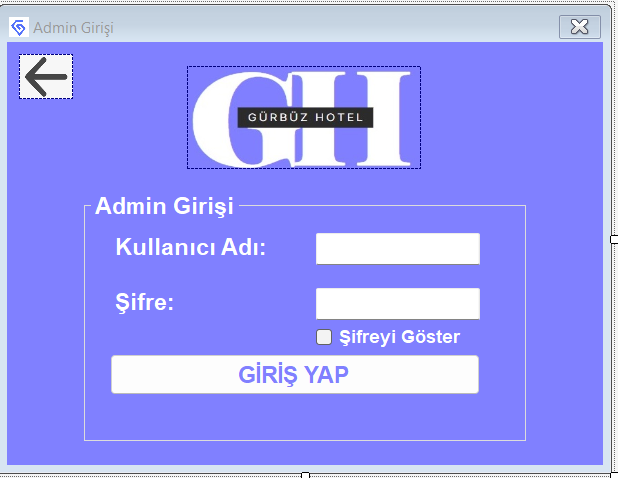
KAYIT FORMU:

Ana Sayfadaki Müşteri Kaydı isimli butona tıklanıldığında açılan sayfa ile Müşterinin Ad-Soyad, T.C Kimlik No, Doğum Tarihi , Mail, Telefon, Cinsiyet, Giriş Tarihi, Çıkış Tarihi gibi bilgileri alınır. Ayrıca Giriş- Çıkış Tarihi üzerinden Doğum Tarihiyle belirlenen yaşı 18 in altında olan kişilere %50 indirim yapılıp 18 yaşından büyük kişilere günlük 200 tl üzerinden ücretlendirme yapılır. Ücret kısmı kullanıcıya kapalıdır. Sisteme girilen doğum tarihi, giriş ve çıkış Tarihi bilgileriyle program tarafından oluşturulur. Odalar kısmında otelde olan odaların butonu bulunur bu butonlardan birine tıklayarak oda seçimi yapılır, yapılan seçim oda numarası kısmına aktarılır. Ana Sayfaya geri dönülmesini sağlamak için sola bakan bir oku gösteren image’e sahip geri butonu vardır. Son olarak ekle butonuna tıklanılarak kayıt gerçekleştirilebilir.

****

ADMİN GİRİŞİ:

Otel personellerinin programa girişini sağlamak için oluşturulmuş bir formdur. Kullanıcı Adı ve Şifre girilir. Şifre görünürlüğü degişikliğinin yapılması için Şifreyi göster ismindeki checkbox işaretlenir. Giriş Yap butonuna veya Enter’a basıldığında kullanıcı adı ve şifrenin doğruluğunun konrolü yapılır hatalı ise ekrana hata mesajı verilir ve tekrar denenmesi beklenir. Doğru ise Admin Sayfası formuna giriş yapılır.



ADMİN SAYFASI:

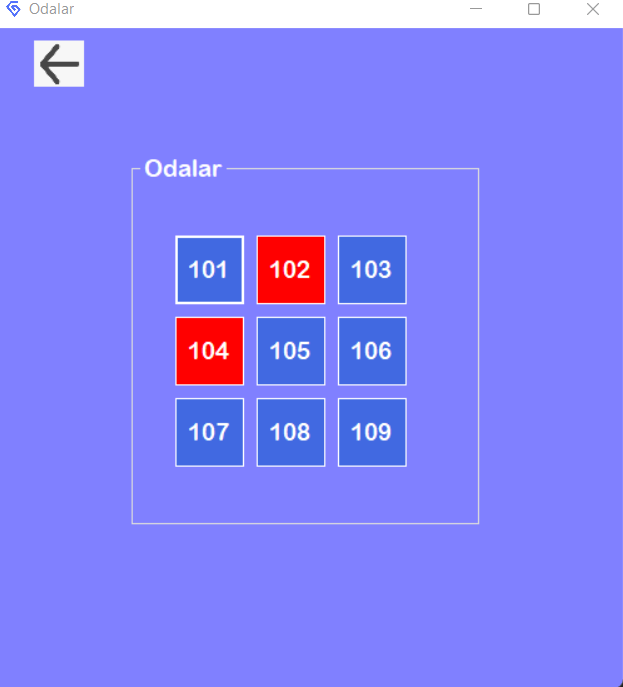
Müsteriler ve Odalar formlarına giriş yapılması için butonlar bulunmaktadır. Müşteri listesini görmek ve bu listede düzenlemeler yapmak için müşteri butonuna, gün içindeki oda doluluk durumunu görmek için ise odalar butonuna tıklanılır.Ayrıca tıklandığında Admin Girişi formuna geri dönülmesi amacı taşıyan sola bakan bir oku gösteren image’e sahip geri butonu bulunur.

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

ODALAR:

Odalar kısmında oteldeki dolu ve boş olan odalar görülmektedir. Kırmızı renkli odalar o gün dolu, mavi renkteki odalar ise doludur. Bu sayfaya bakılarak oda personeli bugün temizlik ya da oda servisi yapıp yapmayacağına karar verebilir. Ayrıca tıklandığında Admin Girişi formuna geri dönülmesi amacı taşıyan sola bakan bir oku gösteren image’e sahip geri butonu bulunur.



MÜŞTERİLER:

Sol tarafta bulunan groupbox’a Ad-Soyad, T.C Kimlik No, Doğum Tarihi , Mail, Telefon, Cinsiyet, Giriş Tarihi, Çıkış Tarihi gibi bilgilerin girilmesi ve sağ taraftaki oda butonlarına tıklanarak oda seçilmesi durumunda kaydet butonuyla müşteri kaydedilebilir ve bu aşağıdaki listeye yansır. Temizle butonuyla textboxlar temizlenir. Listede bulununan kaydın id’sine çift tıklanıldığında üst sol kısımda Ad-Soyad, T.C Kimlik No, Doğum Tarihi , Mail, Telefon, Cinsiyet, Giriş Tarihi, Çıkış Tarihi bilgileri gösterilir çift tıklama durumunda kaydet buttonu güncelle butonuna dönüşür ve buradan müşteri bilgilerinde değişiklik yapılabilir veya sil butonuna tıklanılarak kayıt silinebilir. Arama çubuğuna yazılan büyüteç butonuyla aratılabilir ve çarpı butonuyla silinebilir. Ayrıca tıklandığında Admin Girişi formuna geri dönülmesi amacı taşıyan sola bakan bir oku gösteren image’e sahip geri butonu bulunur.

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**5.PROJENİN KODLARI**

ANA SAYFA (home.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 otel\_otomasyonu

{

public partial class home : Form

{

public home()

{

InitializeComponent();

}

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

{

adminGiris AdminGiris = new adminGiris();

AdminGiris.Show();

this.Hide();

}

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

{

kayitForm KayitForm = new kayitForm();

KayitForm.Show();

this.Hide();

}

}

}

KAYIT FORMU (kayitForm.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.Sql;

using System.Data.SqlClient;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.Collections;

using otel\_otomasyonu;

using System.Reflection.Emit;

namespace otel\_otomasyonu

{

public partial class kayitForm : Form

{

public kayitForm()

{

InitializeComponent();

}

SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-UTB1M8P\\SQLEXPRESS;Initial Catalog=GURBUZHOTEL;Integrated Security=True");

// Diğer değişkenler burada tanımlanır

int price;

int yas;

bool kontrolonsekiz;

DateTime check\_in;

DateTime check\_out;

string secilenOda = "";

// Geri gitme düğmesine tıklanınca çağrılır

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

{

home = new home();

Home.Show();

this.Hide();

}

// Sayısal bir değer olup olmadığını kontrol etmek için yardımcı bir fonksiyon

private bool IsNumeric(string metin)

{

double sayi;

return double.TryParse(metin, out sayi);

}

// Metin içinde sayı olup olmadığını kontrol etmek için yardımcı bir fonksiyon

private bool ContainsNumber(string metinisim)

{

foreach (char karakter in metinisim)

{

if (char.IsDigit(karakter))

{

return true; // Sayı bulundu, true döndür

}

}

return false; // Sayı bulunamadı, false döndür

}

// Doğum tarihi seçildiğinde çağrılır

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

// Doğum tarihini al

DateTime birthdate = dateTimePicker1.Value;

if (dateTimePicker1.Value > DateTime.Today)

{

dateTimePicker1.Value = DateTime.Today;

}

// Bugünün tarihi

DateTime bugun = DateTime.Today;

// Yaşı hesapla

yas = bugun.Year - birthdate.Year;

if (bugun < birthdate.AddYears(yas))

{

yas--;

}

label12.Text = yas.ToString();

if (yas < 18)

{

label13.Text = "18 yaşından küçük";

kontrolonsekiz = true;

}

else

{

label13.Text = "18 yaşında veya 18 yaşından büyük";

kontrolonsekiz = false;

}

txtcikisTarihi\_ValueChanged\_1(sender, e);

}

// Oda 101 seçildiğinde çağrılır

private void button101\_Click(object sender, EventArgs e)

{

baglanti.Open();

SqlCommand komut = new SqlCommand("SELECT \* FROM Customer WHERE room\_number=101 ", baglanti);

int rezervasyonSayisi = 0;

using (SqlDataReader reader = komut.ExecuteReader())

{

while (reader.Read())

{

check\_in = Convert.ToDateTime(reader["check\_in"]);

check\_out = Convert.ToDateTime(reader["check\_out"]);

secilenOda = "101";

rezervasyonSayisi = 1;

if (secilenOda != "" && check\_out != null && check\_in != null && txtcheck\_in.Value < check\_out && txtcheck\_in.Value >= check\_in)

{

DialogResult result = MessageBox.Show(secilenOda + " numaralı oda seçilen tarihler arasında dolu.Yine de seçmek istiyor musunuz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == DialogResult.Yes)

{

txtOdaNo.Text = secilenOda;

}

else

{

secilenOda = "";

MessageBox.Show("Lütfen başka bir oda seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtcheck\_in.Value = check\_out.AddDays(1);

txtcikisTarihi.Value = check\_out.AddDays(2);

txtOdaNo.Clear();

}

}

}

reader.Close();

}

baglanti.Close();

if (rezervasyonSayisi == 0)

{

txtOdaNo.Text = "101";

}

}

//Oda 102,103,104,105,106,107,108,109 için aynı çağrılar yapılır.

private void txtMusteriIsim\_TextChanged(object sender, EventArgs e)

{

string metinisim = txtMusteriIsim.Text;

// Metin içerisinde sayı varsa uyarı ver

if (ContainsNumber(metinisim))

{

MessageBox.Show("Lütfen sadece harf girişi yapın.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMusteriIsim.Clear(); // TextBox'ı temizle

}

}

// MusteritcNo TextBox'ına metin girildiğinde çağrılır

private void txtMusteritcNo\_TextChanged(object sender, EventArgs e)

{

string metintc = txtMusteritcNo.Text;

if (!IsNumeric(metintc))

{

MessageBox.Show("Lütfen sadece sayı girişi yapın.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMusteritcNo.Clear();

}

}

// MusteriTel TextBox'ına metin girildiğinde çağrılır

private void txtMusteriTel\_MaskInputRejected(object sender, MaskInputRejectedEventArgs e)

{

string metintel = txtMusteriTel.Text;

if (!IsNumeric(metintel) && !(metintel.Length > 10))

{

MessageBox.Show("Lütfen sadece sayı girişi yapın.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMusteriTel.Clear();

}

}

// Çıkış tarihi değiştiğinde çağrılır

private void txtcikisTarihi\_ValueChanged\_1(object sender, EventArgs e)

{

DateTime ilktarih = Convert.ToDateTime(txtcheck\_in.Text);

DateTime sontarih = Convert.ToDateTime(txtcikisTarihi.Text);

TimeSpan fark = sontarih - ilktarih;

// Kalınacak gün sayısı

label10.Text = fark.TotalDays.ToString();

if (kontrolonsekiz == true)

{

price = ((Convert.ToInt32(label10.Text)) \* 200) - 100;

txtprice.Text = price.ToString();

}

else

{

price = ((Convert.ToInt32(label10.Text)) \* 200);

txtprice.Text = price.ToString();

}

if (sontarih < ilktarih)

{

MessageBox.Show("Giriş tarihi, çıkış tarihinden sonra olamaz!!!", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtprice.Text = "";

}

if (secilenOda != "" && check\_out != null && check\_in != null && txtcikisTarihi.Value <= check\_out && txtcikisTarihi.Value > check\_in)

{

DialogResult result = MessageBox.Show(secilenOda + " numaralı oda seçilen tarihler arasında dolu.Yine de seçmek istiyor musunuz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == DialogResult.Yes)

{

txtOdaNo.Text = secilenOda;

txtcheck\_in.Value = check\_out.AddDays(1);

txtcikisTarihi.Value = check\_out.AddDays(2);

}

else

{

secilenOda = "";

MessageBox.Show("Lütfen başka bir oda seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtOdaNo.Clear();

}

}

}

// Giriş tarihi değiştiğinde çağrılır

private void txtcheck\_in\_ValueChanged(object sender, EventArgs e)

{

DateTime ilktarih = Convert.ToDateTime(txtcheck\_in.Text);

DateTime sontarih = Convert.ToDateTime(txtcikisTarihi.Text);

TimeSpan fark = sontarih - ilktarih;

// Kalınacak gün sayısı

label10.Text = fark.TotalDays.ToString();

if (kontrolonsekiz == true)

{

price = ((Convert.ToInt32(label10.Text)) \* 200) - 100;

txtprice.Text = price.ToString();

}

else

{

price = ((Convert.ToInt32(label10.Text)) \* 200);

txtprice.Text = price.ToString();

}

if (sontarih < ilktarih)

{

MessageBox.Show("Giriş tarihi, çıkış tarihinden sonra olamaz!!!", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtprice.Text = "";

}

if (secilenOda != "" && check\_out != null && check\_in != null && txtcheck\_in.Value < check\_out && txtcheck\_in.Value >= check\_in)

{

DialogResult result = MessageBox.Show(secilenOda + " numaralı oda seçilen tarihler arasında dolu.Yine de seçmek istiyor musunuz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == DialogResult.Yes)

{

txtOdaNo.Text = secilenOda;

txtcheck\_in.Value = check\_out.AddDays(1);

txtcikisTarihi.Value = check\_out.AddDays(2);

}

else

{

secilenOda = "";

MessageBox.Show("Lütfen başka bir oda seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtOdaNo.Clear();

}

}

}

// Kayıt ekle butonuna tıklanınca çağrılır

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

{

MessageBox.Show("kayıt yapıldı", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

baglanti.Open();

SqlCommand komut = new SqlCommand("insert into Customer(full\_name,identy\_number,email,phone,gender,birthdate,room\_number,price,check\_in,check\_out) values('" + txtMusteriIsim.Text + "','" + txtMusteritcNo.Text + "','" + txtMusteriemail.Text + "','" + txtMusteriTel.Text + "','" + txtMusterigender.Text + "','" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + "','" + txtOdaNo.Text + "','" + txtprice.Text + "','" + txtcheck\_in.Value.ToString("yyyy-MM-dd") + "','" + txtcikisTarihi.Value.ToString("yyyy-MM-dd") + "')", baglanti);

komut.ExecuteNonQuery();

baglanti.Close();

}

private void kayitForm\_Load(object sender, EventArgs e)

{

DateTime bugun = DateTime.Today;

// Yarının tarihini hesapla

DateTime yarin = bugun.AddDays(1);

string yrn = Convert.ToString(yarin);

txtcikisTarihi.Text = yrn;

} }

}

ADMİN GİRİŞİ (adminGiris.cs)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace otel\_otomasyonu

{

public partial class adminGiris : Form

{

public adminGiris()

{

InitializeComponent();

}

SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-UTB1M8P\\SQLEXPRESS;Initial Catalog=GURBUZHOTEL;Integrated Security=True");

private void passwordCheck\_CheckedChanged(object sender, EventArgs e)

{

if (passwordCheck.Checked == true)

{

txtpassword.UseSystemPasswordChar = true;

}

else

{

txtpassword.UseSystemPasswordChar = false;

}

}

private void adminGiris\_Load(object sender, EventArgs e)

{

txtpassword.UseSystemPasswordChar = false;

// Enter olayını tanımla

txtKullanıcıAdı.KeyDown += new KeyEventHandler(txtKullanıcıAdı\_KeyDown);

txtpassword.KeyDown += new KeyEventHandler(txtKullanıcıAdı\_KeyDown);

}

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

{

try

{

baglanti.Open();

string sql = "select \* from [dbo].[User] where user\_name=@user\_name AND password=@password";

SqlParameter prm1 = new SqlParameter("user\_name", txtKullanıcıAdı.Text.Trim());

SqlParameter prm2 = new SqlParameter("password", txtpassword.Text.Trim());

SqlCommand komut = new SqlCommand(sql, baglanti);

komut.Parameters.Add(prm1);

komut.Parameters.Add(prm2);

DataTable dt = new DataTable();

SqlDataAdapter da = new SqlDataAdapter(komut);

da.Fill(dt);

if (dt.Rows.Count > 0)

{

frmadmin frmadmin = new frmadmin();

frmadmin.Show();

this.Hide();

}

else

{

MessageBox.Show("Hatalı giriş! Kullanıcı Adınız veya Şifreniz yanlış girildi.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtKullanıcıAdı.Clear();

txtpassword.Clear();

}

baglanti.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

private void txtKullanıcıAdı\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

buttonGirisYap\_Click(sender, e);

}

}

private void txtpassword\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

buttonGirisYap\_Click(sender, e);

}

}

private void adminGirisiBack\_Click\_1(object sender, EventArgs e)

{

home Home = new home();

Home.Show();

this.Hide();

}

}

}

ADMİN SAYFASI (frmadmin.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 otel\_otomasyonu

{

public partial class frmadmin : Form

{

public frmadmin()

{

InitializeComponent();

}

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

{

musteriler musteriler = new musteriler();

musteriler.Show();

this.Hide();

}

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

{

odalar odalar = new odalar();

odalar.Show();

this.Hide();

}

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

{

adminGiris adminGiris = new adminGiris();

adminGiris.Show();

this.Hide();

}

}

}

MÜŞTERİLER (musteriler.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.Sql;

using System.Data.SqlClient;

using System.Reflection.Emit;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.TextBox;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.Diagnostics;

namespace otel\_otomasyonu

{

public partial class musteriler : Form

{

int id = 0;

int yas;

Boolean kontrolonsekiz;

int price;

DateTime check\_in;

DateTime check\_out;

string secilenOda = "";

public musteriler()

{

InitializeComponent();

}

SqlConnection baglanti = new SqlConnection(@"Data Source=DESKTOP-UTB1M8P\SQLEXPRESS;Initial Catalog=GURBUZHOTEL;Integrated Security=True");

#region Event

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

{

listView1.Items.Clear();

baglanti.Open();

SqlCommand komut = new SqlCommand("select \* from Customer where full\_name like '%" + textBox1.Text + "%'", baglanti);

SqlDataReader oku = komut.ExecuteReader();

while (oku.Read())

{

ListViewItem ekle = new ListViewItem();

ekle.Text = oku["id"].ToString();

ekle.SubItems.Add(oku["full\_name"].ToString());

ekle.SubItems.Add(oku["identy\_number"].ToString());

ekle.SubItems.Add(oku["email"].ToString());

ekle.SubItems.Add(oku["phone"].ToString());

ekle.SubItems.Add(oku["gender"].ToString());

ekle.SubItems.Add(oku["room\_number"].ToString());

ekle.SubItems.Add(oku["birthdate"].ToString());

ekle.SubItems.Add(oku["price"].ToString());

ekle.SubItems.Add(oku["check\_in"].ToString());

ekle.SubItems.Add(oku["check\_out"].ToString());

listView1.Items.Add(ekle);

}

oku.Close();

baglanti.Close();

}

Boolean kontrol;

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

{

if (kontrol = true)

{

txtMusteriIsim.Clear();

//txtMusteritcNo.Clear();

txtMusteritcNo.Text = "";

txtMusteriemail.Clear();

txtMusteriTel.Clear();

txtMusterigender.SelectedItem = null;

txtOdaNo.Clear();

dateTimePicker1.Text = "";

txtprice.Clear();

txtcheck\_in.Text = "";

txtcikisTarihi.Text = "";

txtprice.Clear();

kontrol = true;

}

DateTime bugun = DateTime.Today;

// Yarının tarihini hesapla

DateTime yarin = bugun.AddDays(1);

string yrn = Convert.ToString(yarin);

txtcikisTarihi.Text = yrn;

MessageBox.Show("Textbox lar Temizlendi");

}

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

{

try

{

baglanti.Open();

SqlCommand komut = new SqlCommand("DELETE FROM Customer WHERE identy\_number = @identy\_number", baglanti);

komut.Parameters.AddWithValue("@identy\_number", txtMusteritcNo.Text);

int affectedRows = komut.ExecuteNonQuery();

if (affectedRows > 0)

{

MessageBox.Show("Müşteri başarıyla silindi.");

btnTemizle\_Click(sender, e);

}

else

{

MessageBox.Show("Müşteri bulunamadı.");

}

}

catch (Exception ex)

{

MessageBox.Show("Hata oluştu: " + ex.Message);

}

finally

{

baglanti.Close();

}

verilerigoster();

}

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

{

if (secilenOda != "" && check\_out != null && check\_in != null && ((txtcikisTarihi.Value <= check\_out && txtcikisTarihi.Value > check\_in) || (txtcheck\_in.Value < check\_out && txtcheck\_in.Value >= check\_in)))

{

DialogResult result = MessageBox.Show(secilenOda + " numaralı oda seçilen tarihler arasında dolu.Yine de seçmek istiyor musunuz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == DialogResult.Yes)

{

txtOdaNo.Text = secilenOda;

}

else

{

secilenOda = "";

MessageBox.Show("Lütfen başka bir oda seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtOdaNo.Clear();

}

}

if (btnKaydet.Text == "Kaydet")

{

baglanti.Open();

SqlCommand komut = new SqlCommand("insert into Customer(full\_name,identy\_number,email,phone,gender,birthdate,room\_number,price,check\_in,check\_out) values('" + txtMusteriIsim.Text + "','" + txtMusteritcNo.Text + "','" + txtMusteriemail.Text + "','" + txtMusteriTel.Text + "','" + txtMusterigender.Text + "','" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + "','" + txtOdaNo.Text + "','" + txtprice.Text + "','" + txtcheck\_in.Value.ToString("yyyy-MM-dd") + "','" + txtcikisTarihi.Value.ToString("yyyy-MM-dd") + "')", baglanti);

komut.ExecuteNonQuery();

baglanti.Close();

}

else if (btnKaydet.Text == "Güncelle")

{

baglanti.Open();

int price;

DateTime ilktarih = Convert.ToDateTime(txtcheck\_in.Text);

DateTime sontarih = Convert.ToDateTime(txtcikisTarihi.Text);

if (sontarih < ilktarih)

{

MessageBox.Show("Çıkış tarihi,giriş tarihinden sonra olmalıdır.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

TimeSpan fark = sontarih - ilktarih; //KALINACAK GÜN SAYISI

label10.Text = fark.TotalDays.ToString();

price = (Convert.ToInt32(label10.Text)) \* 200;// günlük 200 tl

txtprice.Text = price.ToString();

SqlCommand komut = new SqlCommand("update Customer set full\_name= '" + txtMusteriIsim.Text + "',identy\_number='" + txtMusteritcNo.Text + "',email='" + txtMusteriemail.Text + "',phone='" + txtMusteriTel.Text + "',gender='" + txtMusterigender.Text + "',room\_number='" + txtOdaNo.Text + "',birthdate='" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + "',price='" + txtprice.Text + "',check\_in='" + txtcheck\_in.Value.ToString("yyyy-MM-dd") + "',check\_out='" + txtcikisTarihi.Value.ToString("yyyy-MM-dd") + "' where id=" + id + "", baglanti);

komut.ExecuteNonQuery();

baglanti.Close();

MessageBox.Show("Kayıt Güncellendi", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

btnTemizle\_Click(sender, e);

}

btnKaydet.Text = "Kaydet";

verilerigoster();

}

#endregion

private void verilerigoster()

{

listView1.Items.Clear();

baglanti.Open();

using (SqlCommand komut = new SqlCommand("select \* from [dbo].[Customer] ORDER BY id ASC", baglanti))

{

SqlDataReader oku = komut.ExecuteReader();

while (oku.Read())

{

ListViewItem ekle = new ListViewItem();

ekle.Text = oku["id"].ToString();

ekle.SubItems.Add(oku["full\_name"].ToString());

ekle.SubItems.Add(oku["identy\_number"].ToString());

ekle.SubItems.Add(oku["email"].ToString());

ekle.SubItems.Add(oku["phone"].ToString());

ekle.SubItems.Add(oku["gender"].ToString());

ekle.SubItems.Add(oku["room\_number"].ToString());

ekle.SubItems.Add(oku["birthdate"].ToString());

ekle.SubItems.Add(oku["price"].ToString());

ekle.SubItems.Add(oku["check\_in"].ToString());

ekle.SubItems.Add(oku["check\_out"].ToString());

listView1.Items.Add(ekle);

}

oku.Close();

}

baglanti.Close();

}

private void listView1\_DoubleClick(object sender, EventArgs e)

{

List<ListViewItem> listViewItems = new List<ListViewItem>();

foreach (ListViewItem item in listView1.Items)

{

listViewItems.Add(item);

}

// ID'sine göre sırala

listViewItems.Sort((x, y) => int.Parse(x.SubItems[0].Text).CompareTo(int.Parse(y.SubItems[0].Text)));

listView1.Items.Clear();

foreach (ListViewItem item in listViewItems)

{

listView1.Items.Add(item);

}

id = int.Parse(listView1.SelectedItems[0].SubItems[0].Text);

txtMusteriIsim.Text = listView1.SelectedItems[0].SubItems[1].Text;

txtMusteritcNo.Text = listView1.SelectedItems[0].SubItems[2].Text;

txtMusteriemail.Text = listView1.SelectedItems[0].SubItems[3].Text;

txtMusteriTel.Text = listView1.SelectedItems[0].SubItems[4].Text;

txtMusterigender.Text = listView1.SelectedItems[0].SubItems[5].Text;

txtOdaNo.Text = listView1.SelectedItems[0].SubItems[6].Text;

dateTimePicker1.Text = listView1.SelectedItems[0].SubItems[7].Text;

txtprice.Text = listView1.SelectedItems[0].SubItems[8].Text;

txtcheck\_in.Text = listView1.SelectedItems[0].SubItems[9].Text;

txtcikisTarihi.Text = listView1.SelectedItems[0].SubItems[10].Text;

btnKaydet.Text = "Güncelle";

}

private void musterilerBack\_Click\_1(object sender, EventArgs e)

{

frmadmin frmadmin = new frmadmin();

frmadmin.Show();

this.Hide();

}

private void musteriler\_Load(object sender, EventArgs e)

{

verilerigoster();

DateTime bugun = DateTime.Today;

// Yarının tarihini hesapla

DateTime yarin = bugun.AddDays(1);

string yrn = Convert.ToString(yarin);

txtcikisTarihi.Text = yrn;

}

private bool ContainsNumber(string metinisim)

{

foreach (char karakter in metinisim)

{

if (char.IsDigit(karakter))

{

return true; // Sayı bulundu, true döndür

}

}

return false; // Sayı bulunamadı, false döndür

}

private bool IsNumeric(string metin)

{

double sayi;

return double.TryParse(metin, out sayi);

}

private void txtMusteriTel\_MaskInputRejected\_1(object sender, MaskInputRejectedEventArgs e)

{

string metintel = txtMusteriTel.Text;

if (kontrol != true)

{

if (!IsNumeric(metintel))

{

MessageBox.Show("Lütfen sadece sayı girişi yapın.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMusteriTel.Clear();

}

}

}

private void dt\_birth\_date\_ValueChanged(object sender, EventArgs e)

{

DateTime birthdate = dateTimePicker1.Value;

if (dateTimePicker1.Value > DateTime.Today)

{

dateTimePicker1.Value = DateTime.Today;

}

// Bugünün tarihi

DateTime bugun = DateTime.Today;

// yıl farkı

yas = bugun.Year - birthdate.Year;

if (bugun < birthdate.AddYears(yas))

{

yas--;

}

label12.Text = yas.ToString();

if (yas < 18)

{

label13.Text = "18 yaşından küçük";

kontrolonsekiz = true;

}

else

{

label13.Text = "18yaşında veya 18 yaşından büyük";

kontrolonsekiz = false;

}

//txtcikisTarihi\_ValueChanged\_1(sender, e);

}

private void txtMusteriIsim\_TextChanged\_1(object sender, EventArgs e)

{

string metinisim = txtMusteriIsim.Text;

if (ContainsNumber(metinisim))

{

MessageBox.Show("Lütfen sadece harf girişi yapın.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void txtMusteritcNo\_TextChanged\_1(object sender, EventArgs e)

{

string metintc = txtMusteritcNo.Text;

if (kontrol != true)

{

if (!IsNumeric(metintc))

{

MessageBox.Show("Lütfen sadece sayı girişi yapın.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

private void txtMusteriTel\_MaskInputRejected(object sender, MaskInputRejectedEventArgs e)

{

string metintel = txtMusteriTel.Text;

if (kontrol != true)

{

if (!IsNumeric(metintel) && !(metintel.Length > 10))

{

MessageBox.Show("Lütfen sadece sayı girişi yapın.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

private void txtcikisTarihi\_ValueChanged\_1(object sender, EventArgs e)

{

DateTime ilktarih = Convert.ToDateTime(txtcheck\_in.Text);

DateTime sontarih = Convert.ToDateTime(txtcikisTarihi.Text);

TimeSpan fark = sontarih - ilktarih;

//KALINACAK GÜN SAYISI

label10.Text = fark.TotalDays.ToString();

if (kontrolonsekiz == true)

{

price = ((Convert.ToInt32(label10.Text)) \* 200) - 100;

txtprice.Text = price.ToString();

}

else

{

price = ((Convert.ToInt32(label10.Text)) \* 200);

txtprice.Text = price.ToString();

}

if (sontarih < ilktarih)

{

// MessageBox.Show("Giriş tarihi, çıkış tarihinden sonra olamaz!!!");

txtprice.Text = "";

}

if (secilenOda != "" && check\_out != null && check\_in != null && txtcikisTarihi.Value <= check\_out && txtcikisTarihi.Value > check\_in)

{

DialogResult result = MessageBox.Show(secilenOda + " numaralı oda seçilen tarihler arasında dolu.Yine de seçmek istiyor musunuz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == DialogResult.Yes)

{

txtOdaNo.Text = secilenOda;

txtcheck\_in.Value = check\_out.AddDays(1);

txtcikisTarihi.Value = check\_out.AddDays(2);

}

else

{

secilenOda = "";

MessageBox.Show("Lütfen başka bir oda seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtOdaNo.Clear();

}

}

}

private void button101\_Click(object sender, EventArgs e)

{

baglanti.Open();

SqlCommand komut = new SqlCommand("SELECT \* FROM Customer WHERE room\_number=101 ", baglanti);

int rezervasyonSayisi = 0;

using (SqlDataReader reader = komut.ExecuteReader())

{

while (reader.Read())

{

check\_in = Convert.ToDateTime(reader["check\_in"]);

check\_out = Convert.ToDateTime(reader["check\_out"]);

secilenOda = "101";

rezervasyonSayisi = 1;

if (secilenOda != "" && check\_out != null && check\_in != null && txtcheck\_in.Value < check\_out && txtcheck\_in.Value >= check\_in)

{

DialogResult result = MessageBox.Show(secilenOda + " numaralı oda seçilen tarihler arasında dolu.Yine de seçmek istiyor musunuz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == DialogResult.Yes)

{

txtOdaNo.Text = secilenOda;

}

else

{

secilenOda = "";

MessageBox.Show("Lütfen başka bir oda seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtcheck\_in.Value = check\_out.AddDays(1);

txtcikisTarihi.Value = check\_out.AddDays(2);

txtOdaNo.Clear();

}

}

}

reader.Close();

}

baglanti.Close();

if (rezervasyonSayisi == 0)

{

txtOdaNo.Text = "101";

}

}

//Oda 102,103,104,105,106,107,108,109 için aynı çağrılar yapılır.

private void txtcheck\_in\_ValueChanged(object sender, EventArgs e)

{

DateTime ilktarih = Convert.ToDateTime(txtcheck\_in.Text);

DateTime sontarih = Convert.ToDateTime(txtcikisTarihi.Text);

TimeSpan fark = sontarih - ilktarih;

//KALINACAK GÜN SAYISI

label10.Text = fark.TotalDays.ToString();

if (kontrolonsekiz == true)

{

price = ((Convert.ToInt32(label10.Text)) \* 200) - 100;// günlük 200 tl -100

txtprice.Text = price.ToString();

}

else

{

price = ((Convert.ToInt32(label10.Text)) \* 200);// günlük 200 tl

txtprice.Text = price.ToString();

}

if (sontarih < ilktarih)

{

// MessageBox.Show("Giriş tarihi, çıkış tarihinden sonra olamaz!!!");

txtprice.Text = "";

}

if (secilenOda != "" && check\_out != null && check\_in != null && txtcheck\_in.Value < check\_out && txtcheck\_in.Value >= check\_in)

{

DialogResult result = MessageBox.Show(secilenOda + " numaralı oda seçilen tarihler arasında dolu.Yine de seçmek istiyor musunuz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == DialogResult.Yes)

{

txtOdaNo.Text = secilenOda;

txtcheck\_in.Value = check\_out.AddDays(1);

txtcikisTarihi.Value = check\_out.AddDays(2);

}

else

{

secilenOda = "";

MessageBox.Show("Lütfen başka bir oda seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtOdaNo.Clear();

}

}

}

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

{

textBox1.Text = "";

verilerigoster();

}

private void listView1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}

ODALAR (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.Sql;

using System.Data.SqlClient;

namespace otel\_otomasyonu

{

public partial class odalar : Form

{

public odalar()

{

InitializeComponent();

}

// Veritabanı bağlantısı için SqlConnection nesnesi

SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-UTB1M8P\\SQLEXPRESS;Initial Catalog=GURBUZHOTEL;Integrated Security=True");

// Geri gitme düğmesine tıklanınca çağrılır

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

{

// frmadmin formunu oluşturup gösterir, bu formu gizler

frmadmin frmadmin = new frmadmin();

frmadmin.Show();

this.Hide();

}

// Form yüklendiğinde çağrılır

private void odalar\_Load(object sender, EventArgs e)

{

// Odaları oluşturan fonksiyonu çağırır

odaolustur();

}

// Odaları oluşturan fonksiyon

private void odaolustur()

{

// Başlangıç oda numarası

int oda = 100;

// Butonların dikey mesafesi

int mesafe = 0;

// Butonların yatay mesafesi

int x = -30;

// 3x3'lük döngü ile butonlar oluşturulur

for (int i = 1; i <= 3; i++)

{

mesafe = mesafe + 65;

for (int j = 1; j <= 3; j++)

{

// Yeni bir buton oluşturulur

Button button = new Button

{

Text = (oda + (i - 1) \* 3 + j).ToString(),

Location = new System.Drawing.Point(j \* 65 + x, mesafe),

Height = 55,

Width = 55,

ForeColor = Color.White,

FlatStyle = FlatStyle.Flat

};

// Veritabanı bağlantısı açılır

using (SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-UTB1M8P\\SQLEXPRESS;Initial Catalog=GURBUZHOTEL;Integrated Security=True"))

{

baglanti.Open();

// Veritabanından oda durumları kontrol edilir

string sorgu = "SELECT [check\_in],[check\_out] FROM Customer WHERE [room\_number] = @deger";

// Parametreli sorgu çalıştırılır

using (SqlCommand command = new SqlCommand(sorgu, baglanti))

{

command.Parameters.AddWithValue("@deger", (i - 1) \* 3 + j + oda);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

DateTime today = DateTime.Now;

DateTime check\_in = Convert.ToDateTime(reader["check\_in"]);

DateTime check\_out = Convert.ToDateTime(reader["check\_out"]);

check\_in = check\_in.Date.AddHours(12);

check\_out = check\_out.Date.AddHours(12);

// Oda doluysa arka plan rengi kırmızı yapılır, boşsa mavi yapılır

if ((today >= check\_in && today <= check\_out))

{

button.BackColor = Color.Red;

}

else

{

button.BackColor = Color.RoyalBlue;

}

}

else

{

button.BackColor = Color.RoyalBlue;

}

// Butona tıklanma eventi atanır

button.Click += button\_Click;

// Buton groupBox kontrolüne eklenir

groupBox1.Controls.Add(button);

// Reader kapatılır

reader.Close();

}

// Veritabanı bağlantısı kapatılır

baglanti.Close();

}

}

}

}

// Bir oda butonuna tıklandığında çağrılır

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

{

Button btn = (Button)sender;

odalar rez = new odalar(btn);

}

public odalar(Button btn) : this()

{

// Yapılandırıcı kodu buraya eklenir

}

}

}