BİLGİSAYAR PROGRAMCILIĞI BİTİRME PROJESİ

Hazırlayan

Öğrenci Adı : Nursena Şevval Candan

Öğrenci Numarası : 19MY03002

# giriş

## içindekiler

[GİRİŞ 1](#_Toc40041896)

[İÇİNDEKİLER 1](#_Toc40041897)

[PROJE KONUSU 1](#_Toc40041898)

[İNCELENEN PROJELERİN ÖZETİ 2](#_Toc40041899)

[3)KULLANILAN ARAÇLAR 2](#_Toc40041900)

[PROJE HAZIRLARKEN KULLANILAN ARAÇ VE TEKNOLOJİLER 2](#_Toc40041901)

[PROJE GELİŞTİRME ORTAMI VE ARAÇLARIN HAZIRLANMASI 3](#_Toc40041902)

[4)PROJENİN HAZIRLANMASI 3](#_Toc40041903)

[PROJENİN PLANLANMASI 3](#_Toc40041904)

[1)KULLANICI TÜRLERİNİN TANIMLANMASI 3](#_Toc40041905)

[2)KULLANICI BAZLI İŞ AKIŞ PLANI 3](#_Toc40041906)

[3)KULLANILACAK NESNE(CLASS), ARABİRİM(İNTERFACE), KOLEKSİYON VE DİĞER ARAÇLARIN PLANLANMASI 4](#_Toc40041907)

[4)EKRANLARIN TASARIMI 5](#_Toc40041908)

[5)VERİTABANININ TASARIMI 5](#_Toc40041909)

[PROJENİN BİRLEŞTİRİLMESİ, VARSA SUNUCUYA YÜKLENMESİ 6](#_Toc40041910)

[PROJENİN TEST EDİLMESİ 6](#_Toc40041911)

[4)SONUÇ 6](#_Toc40041912)

[BAŞLANGIÇTAKİ HEDEFLERE ULAŞMA DURUMU 6](#_Toc40041913)

[ELDE EDİLEN PERFORMANSIN DEĞERLENDİRİLMESİ 7](#_Toc40041914)

[İYİLEŞTİRME VE GELİŞTİRME HEDEFLERİ 7](#_Toc40041915)

[6)KAYNAKLAR 7](#_Toc40041916)

[7)EKLER 7](#_Toc40041917)

[GITHUB ADRESİ 7](#_Toc40041918)

[VERİTABANI YAPISI 7](#_Toc40041919)

[PROGRAM ARAYÜZLERİ 8](#_Toc40041920)

[PROGRAM KODLARI 10](#_Toc40041921)

[KULLANIM KILAVUZU 33](#_Toc40041922)

[TANITIM BROŞÜRÜ 34](#_Toc40041923)

# ÖZET

Projem bir şirketin müşteri kayıt,işlem kaydı ve ürün stok takibi yapabilmeleri için yapılmış bir uygulamadır.

## Proje konusu

Bu proje,bir Otogaz Dönüşüm Servisinin müşteri kayıtlarını tutmaları ve stok takibi yapabilmeleri için yapılmıştır.

Proje, 5 ayrı formdan oluşmakta :

Müşteri Kayıt sayfasından,yeni gelen müşterinin kaydı yapılırken aynı zamanda müşterinin satın almış olduğu ürünlerin girişleri de yapılıyor.

Böylece müşterinin satın almış olduğu ürünlerin kendi stoğundan 1 ‘er adet eksiliyor.

(Otogaz dönüşüm servisinde bir araç dönüşümü için 1 adet LPG Kit ve 1 adet Tank kullanılıyor.)

Ürün Kayıt sayfasında, yeni gelen ürünler aşağıda belirttiğim bilgilerle beraber sistemden eklenilen tarih bilgisiyle stoğa eklenir ;

-Ürün Adı (LPG Kit veya Tank)

-Ürün Markası

-Ürün Modeli

-Ürün Miktarı

Bu proje kullanıcılar tarafından :

* Müşteri kaydı yapabilmek
* Müşteri kaydı yaparken satın alınan ürünü stoktan düşürmek
* Müşteri kaydı yaparken ürün stoğunu güncel tutabilmek
* Kayıtlı müşterileri listeleyebilmek
* Ürün stoğuna kayıt ekleyebilmek
* Stok sorgulayabilmek
* Güncel stok kaydını görüntüleyebilmek

Amacıyla yapılmıştır.

## İNCELEnen PROJELERİN ÖZETİ

1)Derste hocamızın yapmış olduğu Telefon Rehberi Projesini inceledim

Bir telefon rehberi oluşturdu ve bu rehbere yeni kişi ekleyebilirsin,kişileri listeleyebilirsin ve silebilirsin.

2)İnternetten bulduğum Stok Takip Otomasyonu Projesini inceledim.

Bir market için stok takibi yapabildiği bir satış otomasyonu oluşturdu.Satılan ürünlerin bilgilerini ve miktarlarını girdikten sonra satış yapınca stoktan düşüyor.

3)İnternette bulduğum Araç Kiralama Otomasyonu Projesini inceledim.

Müşterilerin listelendiği ve kayıt yapıldığı daha sonra uygulama üzerinden araç kiralayabildikleri bir otomasyon oluşturdu.

4)İnternette bulduğum Barkodlu Satış Otomasyonu Projesini inceledim.

Ürünlerin barkod numaralarına göre satış yapıldığı bir otomasyon oluşturdu.

# 3)kullanılan araçlar

## proje hazırlarken kullanılan araç ve teknolojiler

ASUS 8 GB RAM 120 GB SSD i5 64 bit işletim sistemi

Windows 10 Pro

Microsoft Visual Studio 2017

## Proje geliştirme ortamı ve araçların hazırlanması

Projemi Visual Studio 2017 programı üzerinde oluşturdum.Visual Studio’yu daha öncesinden kullandığım için program bilgisayarımda kuruluydu.

Projeyi oluşturmadan önce planlama yaptım.Projeyi daha önceden hazırladığım plana göre oluşturmaya çalıştım.

Projede kullanmak için uygun fotoğrafları bulup indirdim.

Projeyi planlarken bir kağıda :

* Projemde kullanacağım formları ,
* Formlarımın nasıl görüneceğini,
* Formlarımda kullanacağım textbox’ları,
* Kodlama yaparken kolay olması için textbox’lara vereceğim textname’leri,
* Formlarımda kullanacağım combobox’ları,
* Kodlama yaparken kolaylık olması için textbox’larıma ve combobox’larıma vereceğim textname’leri ve comboname’leri,
* Formların nasıl işleyeceğini

yazdım ve tasarladım.

# 4)Projenin hazırlanması

## pROJENİN PLANLANMASI

### 1)kullanıcı türlerinin tanımlanması

Projem de kullanıcı türleri tanımlamadım.Uygulamaya sahip olan kişiler istedikleri gibi giriş yapabilirler.

### 2)KULLANICI BAZLI İŞ AKIŞ PLANI

1. Proje konusunu belirlendi
2. Proje planlaması yapıldı
3. Projenin içerikleri,sayfaları tasarlandı
4. Proje için gerekli fotoğraflar bulundu
5. Anasayfanın tasarımı ve kodlaması yapıldı
6. SQL’de proje için veritabanı oluşturuldu
7. Veritabanına tablolar eklendi
8. Class oluşturuldu
9. SQL’deki veritabanı Visual’daki projede oluşturulan class’ın içinde bağlandı
10. Müşteri Kayıt Sayfasının tasarımı ve kodlaması yapıldı
11. Müşteri Kayıt Sayfasına veriler yazıldıktan sonra Kaydet butonuna tıklandığında, veritabanındaki Müşteri Kayıt Tablosuna verilerin eklenmesi için kodlaması yapıldı
12. Müşteri Kayıt Sayfasında Müşterinin satın aldığı ürünleri seçtikten ve Kaydet butonuna tıklandıktan sonra veritabanındaki Ürün tablosundan seçilen ürünün miktarından 1’er adet eksilmesi için gerekli kodlama yapıldı
13. Oluşturduğum class’ın içinde verileri eklemek,silmek ve güncellemek için bir nesne kullanıldı
14. Müşteri Arama Sayfasının tasarımı ve kodlaması yapıldı
15. Müşteri Arama Sayfasında, Müşteri Ad Soyad’ına göre arama yapılabilmesi için kodlama yapıldı
16. Müşteri Arama Sayfasındaki Güncelle butonuna tıklandığında veritabanında güncelleme yapması için kodlama yapıldı
17. Ürün Kayıt Sayfasında,verileri girilen ürünlerin Ekle butonuna tıklandığında veritabanındaki Ürün tablosuna eklenmesi için kodlaması yapıldı
18. Ürün Stok Sorgula Sayfasında, Ürün Modeli’ne göre arama yapılabilmek için kodlaması yapıldı
19. Ürün Stok Sayfasındaki Güncelle butonuna tıklandığında veritabanında güncelleme yapması için kodlama yapıldı
20. Proje test edildi.

### 3)KULLANILACAK NESNE(CLASS), ARABİRİM(İNTERFACE), KOLEKSİYON VE DİĞER ARAÇLARIN PLANLANMASI

* ‘candan\_muhendıslık’ adında bir adet Class oluşturdum.
* Class’ımın içerisinde SQL bağlantısı oluşturdum.
* ‘MüşteriKayıtSayfası.cs’,’MüşteriAramaSayfası.cs’,’ÜrünKayıtSayfası.cs’ ve ‘ÜrünStokSorgulamaSayfası.cs’ sayfalarından ‘candan\_muhendıslık’ adlı class’ımı çektim.
* ‘candan\_muhendıslık’ adlı Class’ın içinde ‘ekle\_sil\_güncelle’ adında bir nesne oluşturdum.
* ‘MüşteriKayıtSayfası.cs’ ve ‘ÜrünKayıtSayfası.cs’ sayfalarından oluşturmuş olduğum ‘ekle\_sil\_güncelle’ adlı nesnemi çektim.

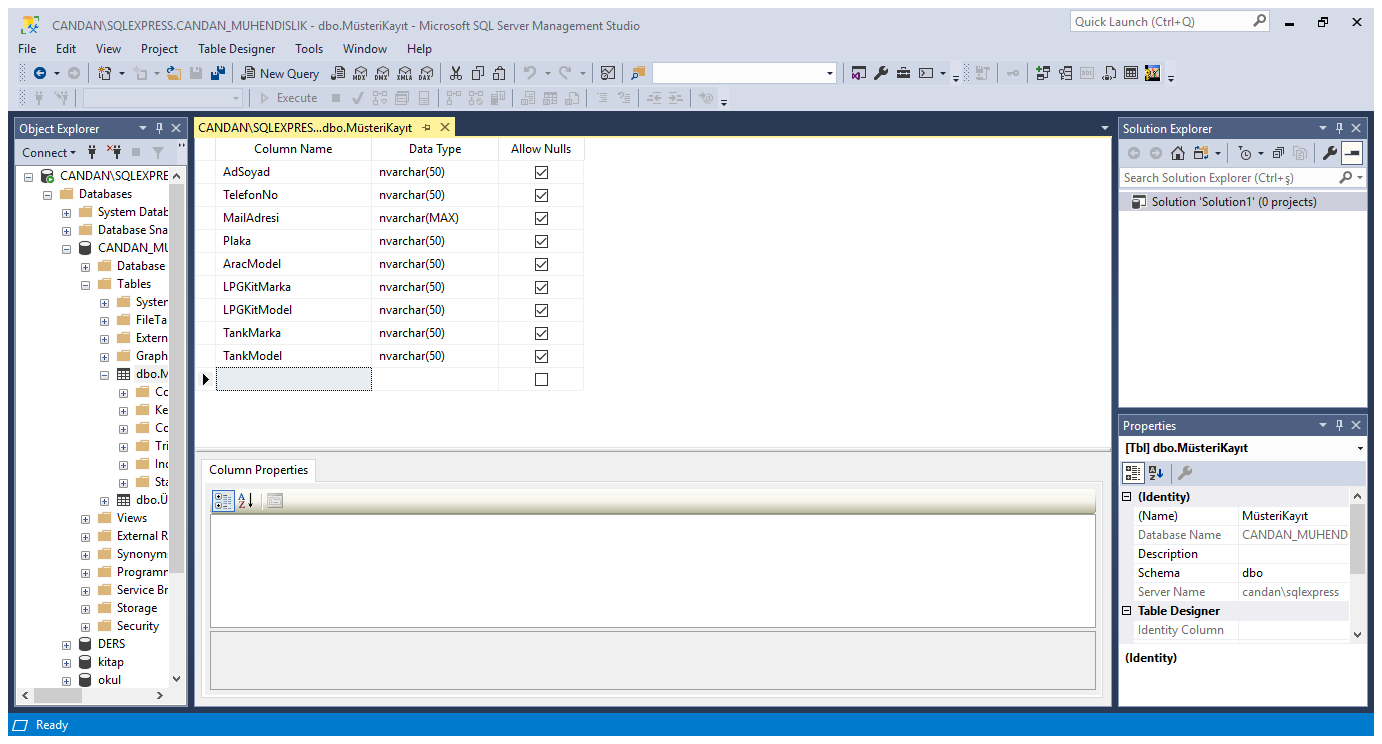
### 4)ekranların tasarımı

* 5 adet ekran tasarımı yaptım.
* Ekranların arka plan tasarımlarını ve butonlarını aynı yapmaya özen gösterdim
* Aynı yazı tipini kullanmaya çalıştım.
* Butonlar için simgeler buldum.
* Tasarım olarak göz yormamasına, sade olmasına özen gösterdim
* Anasayfa için işyerinin uygun bir fotoğrafını kullanarak sayfayı renklendirdim
* Genel olarak önemli verilere yer verdim, kolay ve pratik olmasına özen gösterdim

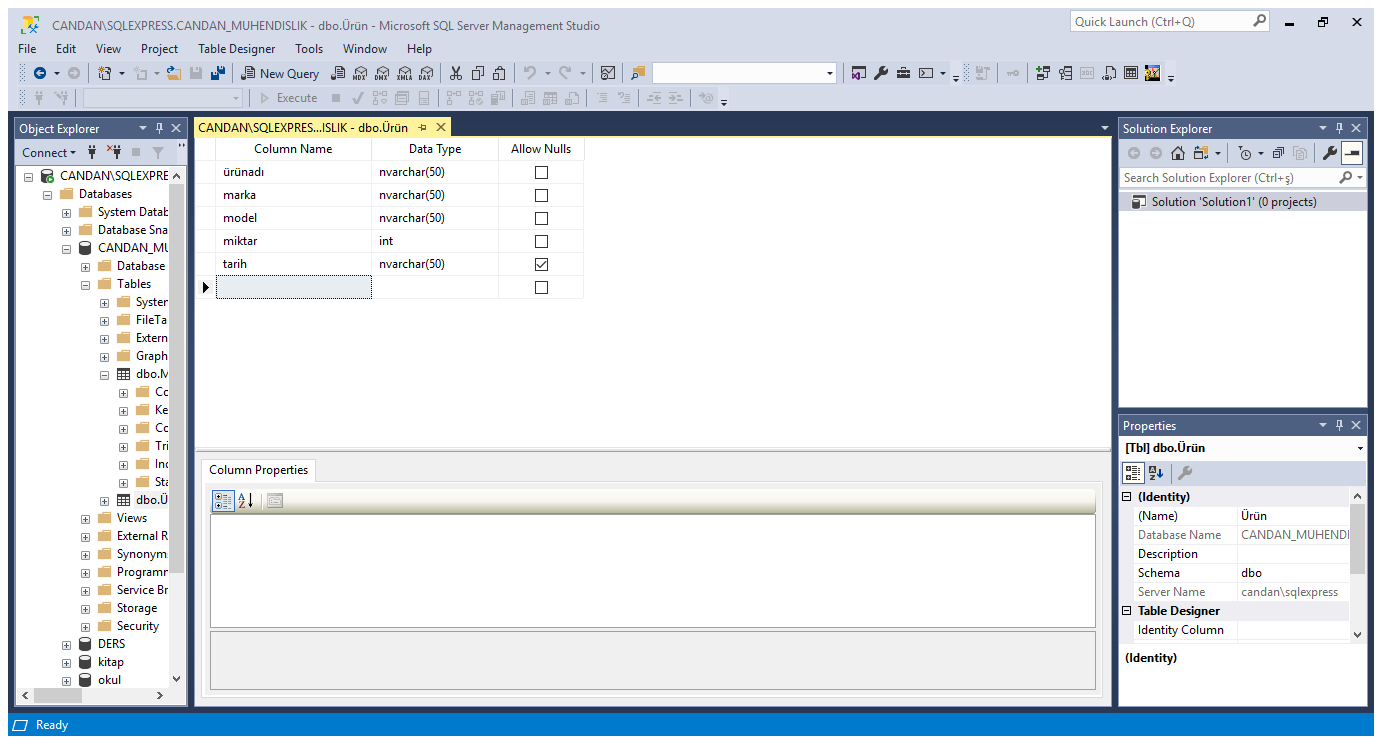
### 5)VERİTABANININ TASARIMI

* Veritabanında ihtiyacım olan iki adet tablo var
* Bir adet müşteri kayıtlarını tutmamız için Müşteri Tablosu gerekli
* Müşteri Tablosuna müşteri bilgileri ve satın aldığı ürünlerin bilgilerini ekledim
* Bir adet ürün stoklarının tutulduğu Ürün Tablosu gerekli
* Ürün Tablosuna ürün bilgilerini ekledim

Müşteri Kayıt Tablosu



Ürün Tablosu



## projenin iş katmanın hazırlanması

* Anasayfanın tasarımı ve kodlaması yapıldı
* SQL’de proje için veritabanı oluşturuldu
* Veritabanına tablolar eklendi
* Class oluşturuldu
* SQL’deki veritabanı Visual’daki projede oluşturulan class’ın içinde bağlandı
* Müşteri Kayıt Sayfasının tasarımı ve kodlaması yapıldı
* Müşteri Kayıt Sayfasına veriler yazıldıktan sonra Kaydet butonuna tıklandığında, veritabanındaki Müşteri Kayıt Tablosuna verilerin eklenmesi için kodlaması yapıldı
* Müşteri Kayıt Sayfasında Müşterinin satın aldığı ürünleri seçtikten ve Kaydet butonuna tıklandıktan sonra veritabanındaki Ürün tablosundan seçilen ürünün miktarından 1’er adet eksilmesi için gerekli kodlama yapıldı
* Oluşturduğum class’ın içinde verileri eklemek,silmek ve güncellemek için bir nesne kullanıldı
* Müşteri Arama Sayfasının tasarımı ve kodlaması yapıldı
* Müşteri Arama Sayfasında, Müşteri Ad Soyad’ına göre arama yapılabilmesi için kodlama yapıldı
* Müşteri Arama Sayfasındaki Güncelle butonuna tıklandığında veritabanında güncelleme yapması için kodlama yapıldı
* Ürün Kayıt Sayfasında,verileri girilen ürünlerin Ekle butonuna tıklandığında veritabanındaki Ürün tablosuna eklenmesi için kodlaması yapıldı
* Ürün Stok Sorgula Sayfasında, Ürün Modeli’ne göre arama yapılabilmek için kodlaması yapıldı
* Ürün Stok Sayfasındaki Güncelle butonuna tıklandığında veritabanında güncelleme yapması için kodlama yapıldı

## Projenin birleştirilmesi, varsa sunucuya yüklenmesi

Bütün sayfalarımı bitirdikten sonra projeme ait bütün dosyaları bir dosyada topladım.Daha sonra Github hesabıma yükledim.

## Projenin test edilmesi

Projeyi yaparken her aşamasında test ederek devam ettim.Her sayfada kodları yazdıktan sonra test ettim.Sql bağlarken birçok kez hata aldım.Hatta Visual’ın içinde veritabanı oluşturmayı denedim.Sorunu bulmak için uğraştım en sonunda buldum.Sql sorununu çözdükten sonra daha kolay oldu ödevin geri kalanı.Ancak yine bir çok kez hata aldım.İlk iki sayfayı yaptıktan sonra tasarımlarım silindi ve sayfaları tekrar oluşturmak,kodlamak zorunda kaldım.Bu yüzden proje dosyamda bir çok visual dosyam oluştu.Projeyi test ederek devam ettim.Bittikten sonra da tümünü test ettim ve hatalı yerleri not ederek düzeltmeye çalıştım.Bazı yerleri düzeltebildim ancak bazı sorunlarıma çözüm bulamadım.

# 4)sonuç

## Başlangıçtaki hedeflere ulaşma durumu

İstediğim gibi çalışmayan kısımlar :

* Müşteri Arama ve Ürün Stok Sorgulama sayfalarında veritabanında arama yapmak için kodlamayı yaptım fakat istediğim gibi arama yapamıyorum.
* Müşteri Arama ve Ürün Stok Sorgulama sayfalarında veritabanındaki bir veri ile ilgili değişiklik yapmak için güncelle butonuna kodlama yaptım fakat güncelle butonuna bastığım zaman tablodaki bütün veriler değişip o veriye dönüşüyor.

Diğer kısımlar tam da planladığım ve istediğim gibi çalışıyor .

## elde edilen performansın değerlendirilmesi

Kısıtlı bir süre içinde ilk defa böyle bir proje yapmama rağmen bana göre kullanışlı bir proje oldu.Üzerine eklenerek,sorunları giderilerek kullanışlı bir program olacağını düşünüyorum.Projemi var olan bir işyeri üzerinden yaptığım için tam olarak ihtiyaçlar üzerine oluşturdum.Bu da varsayım yapmak yerine gerçekçi olmasına olanak sağladı.

## İyileştirme ve geliştirme hedefleri

Bu proje için ileriye dönük hedeflerim şöyle :

* Projemin içerisine randevu oluşturma kısmı ekleyebilirim böylelikle işyerinde işler daha düzenli ilerler.
* Bu projeye birkaç farklı kullanıcı girişi ekleyip,işyerindeki alanlardaki kişilere göre özelleştirebilirim.Mesela montajı yapacak kişi sadece montaj hakkındaki bilgileri görebilir yada müşteri ilişkileri müşteri bilgilerini ve ödeme bilgilerini görebilir gibi.
* Kullanıcılar için şifreli giriş yapabilecekleri giriş sayfası ekleyebilirim.

# 6)kaynaklar

.Net Programlama ders notları.

<https://www.youtube.com/watch?v=w_cdeeF4qMY>

<https://www.youtube.com/watch?v=YSQd20gKZJs&list=PL22TK-6CtDafQkb1bMdoRyL7yhnMZwxDy>

<https://www.youtube.com/watch?v=AYI_yA5vpYI>

# 7)ekler

## GITHUB ADRESİ

<https://github.com/nursenacndnn/Bitirme-Projesi>

## VERİTABANI YAPISI

USE [master]

GO

/\*\*\*\*\*\* Object: Database [CANDAN\_MUHENDISLIK] Script Date: 10.05.2020 22:39:03 \*\*\*\*\*\*/

CREATE DATABASE [CANDAN\_MUHENDISLIK]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'CANDAN\_MUHENDISLIK', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\CANDAN\_MUHENDISLIK.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'CANDAN\_MUHENDISLIK\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\CANDAN\_MUHENDISLIK\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET COMPATIBILITY\_LEVEL = 140

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [CANDAN\_MUHENDISLIK].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET ARITHABORT OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET DISABLE\_BROKER

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET RECOVERY SIMPLE

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET MULTI\_USER

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET DB\_CHAINING OFF

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [CANDAN\_MUHENDISLIK] SET QUERY\_STORE = OFF

GO

USE [CANDAN\_MUHENDISLIK]

GO

/\*\*\*\*\*\* Object: Table [dbo].[MüsteriKayıt] Script Date: 10.05.2020 22:39:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[MüsteriKayıt](

[AdSoyad] [nvarchar](50) NULL,

[TelefonNo] [nvarchar](50) NULL,

[MailAdresi] [nvarchar](max) NULL,

[Plaka] [nvarchar](50) NULL,

[AracModel] [nvarchar](50) NULL,

[LPGKitMarka] [nvarchar](50) NULL,

[LPGKitModel] [nvarchar](50) NULL,

[TankMarka] [nvarchar](50) NULL,

[TankModel] [nvarchar](50) NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Ürün] Script Date: 10.05.2020 22:39:04 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Ürün](

[ürünadı] [nvarchar](50) NOT NULL,

[marka] [nvarchar](50) NOT NULL,

[model] [nvarchar](50) NOT NULL,

[miktar] [int] NOT NULL,

[tarih] [nvarchar](50) NULL

) ON [PRIMARY]

GO

INSERT [dbo].[MüsteriKayıt] ([AdSoyad], [TelefonNo], [MailAdresi], [Plaka], [AracModel], [LPGKitMarka], [LPGKitModel], [TankMarka], [TankModel]) VALUES (N'SADIK PASLI', N'05466678900', N'sadık12@gmail.com', N'34 ED 099', N'SUBARU XV', N'Prins', N'Ecomax', N'Step Tank', N'48 LT')

INSERT [dbo].[MüsteriKayıt] ([AdSoyad], [TelefonNo], [MailAdresi], [Plaka], [AracModel], [LPGKitMarka], [LPGKitModel], [TankMarka], [TankModel]) VALUES (N'AHMET AKAR', N'054666672780', N'akar@gmail.com', N'34 EB 112', N'CİTROEN C3', N'Atiker', N'Gold', N'Atiker Tank', N'32 LT')

INSERT [dbo].[MüsteriKayıt] ([AdSoyad], [TelefonNo], [MailAdresi], [Plaka], [AracModel], [LPGKitMarka], [LPGKitModel], [TankMarka], [TankModel]) VALUES (N'SELİM OKUT', N'056438734882', N'slmokut@gmail.com', N'34 TH 111', N'OPEL ASTRA', N'Lovato', N'OBD 2', N'Step Tank', N'42 LT')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'LPG KİT', N'Prins', N'Ecomax', 18, N'19.04.2020 23:40:54')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'TANK', N'Step Tank', N'48 LT', 15, N'23.04.2020 14:09:34')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'TANK', N'Step Tank', N'55 LT', 15, N'23.04.2020 14:09:57')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'TANK', N'Atiker Tank', N'48 LT', 15, N'23.04.2020 14:10:17')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'TANK', N'Step Tank', N'42 LT', 15, N'23.04.2020 14:12:35')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'TANK', N'Atiker Tank', N'55 LT', 20, N'23.04.2020 14:13:38')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'TANK', N'Atiker Tank', N'42 LT', 15, N'23.04.2020 14:14:38')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'LPG KİT', N'Atiker', N'Atikfast', 48, N'23.04.2020 14:15:34')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'LPG KİT', N'Atiker', N'Atikfast', 3, N'23.04.2020 14:15:49')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'LPG KİT', N'Atiker', N'Grand', 50, N'23.04.2020 14:16:24')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'LPG KİT', N'BRC', N'Platinum', 50, N'23.04.2020 14:16:40')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'LPG KİT', N'Lovato', N'OBD 2', 57, N'23.04.2020 14:17:13')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'LPG KİT', N'Lovato', N'Smart', 25, N'23.04.2020 14:17:37')

INSERT [dbo].[Ürün] ([ürünadı], [marka], [model], [miktar], [tarih]) VALUES (N'LPG KİT', N'BRC', N'Comfort', 8, N'19.04.2020 23:49:15')

USE [master]

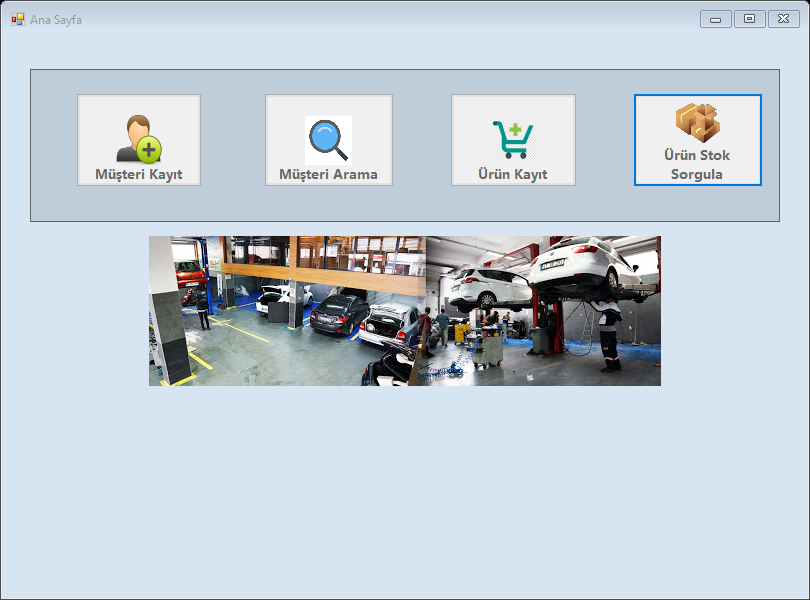
GO

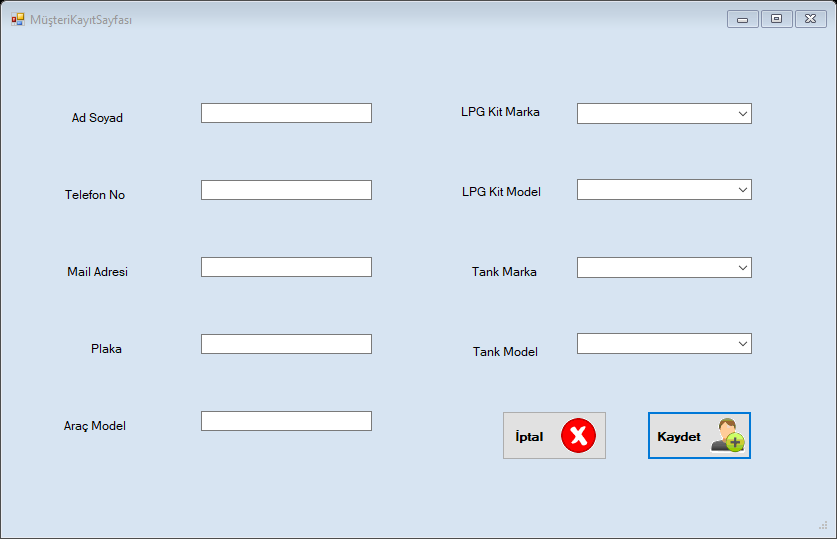
ALTER DATABASE [CANDAN\_MUHENDISLIK] SET READ\_WRITE

GO

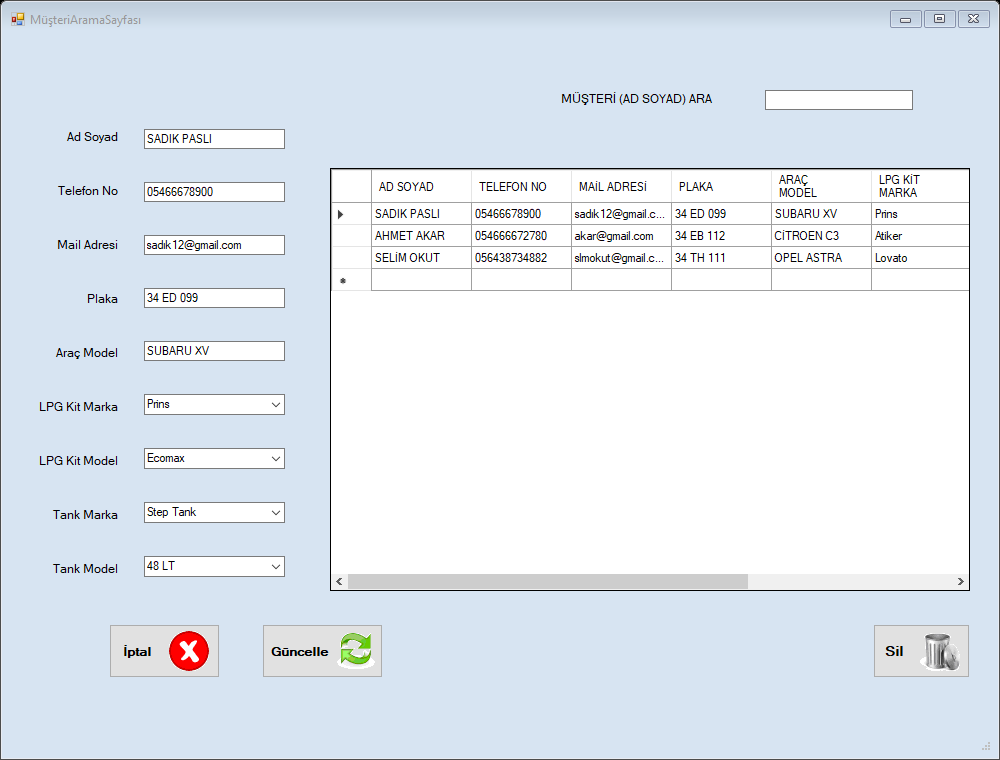
## PROGRAM ARAYÜZLERİ

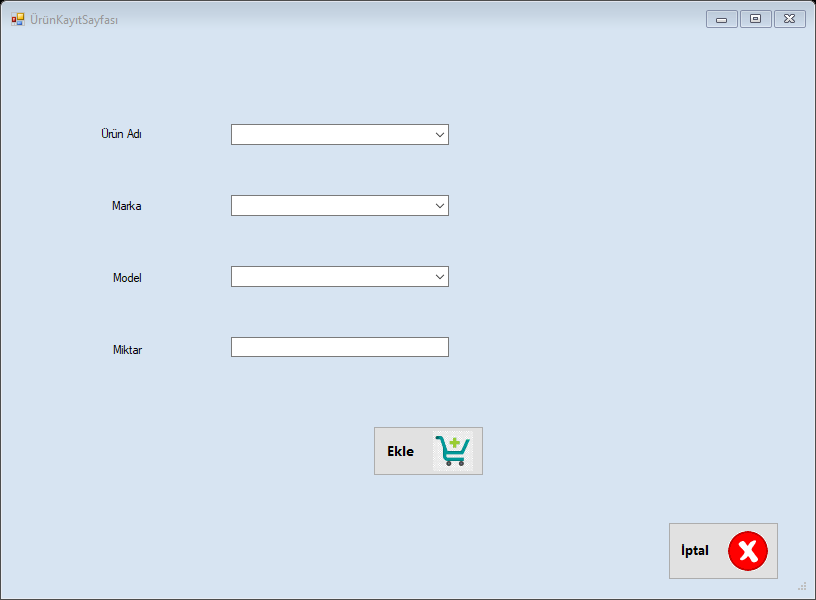
ANASAYFA



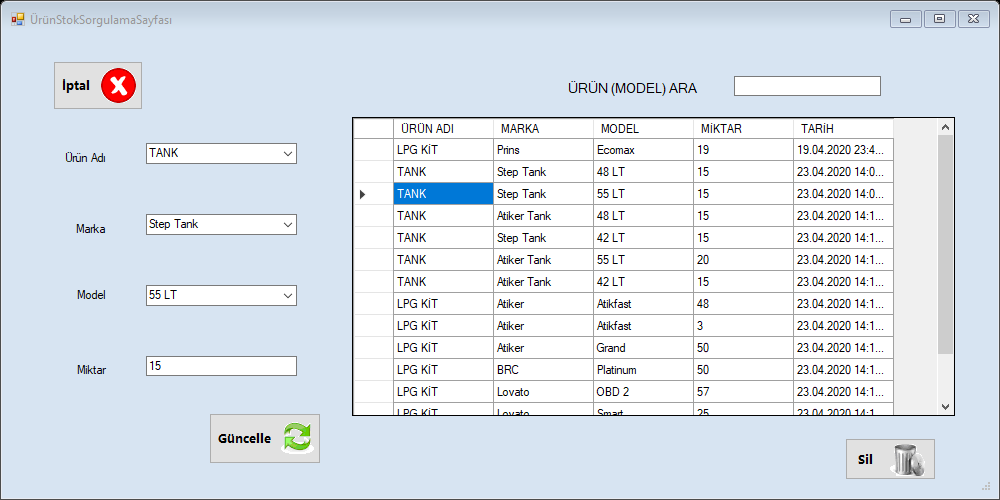
MÜŞTERİ KAYIT SAYFASI

MÜŞTERİ ARAMA SAYFASI



ÜRÜN KAYIT SAYFASI

ÜRÜN STOK SORGULAMA SAYFASI



## PROGRAM KODLARI

**candan\_muhedıslık.cs**

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WindowsFormsApp1

{

class candan\_muhendıslık

{

SqlConnection baglantı = new SqlConnection("Data Source=CANDAN\\SQLEXPRESS;Initial Catalog=CANDAN\_MUHENDISLIK;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

DataTable tablo;

public void Ekle\_sil\_güncelle(SqlCommand komut,string sorgu)

{

baglantı.Open();

komut.Connection = baglantı;

komut.CommandText = sorgu;

komut.ExecuteNonQuery();

baglantı.Close();

}

public DataTable Listele(SqlDataAdapter adtr,string sorgu)

{

tablo = new DataTable();

adtr = new SqlDataAdapter(sorgu,baglantı);

adtr.Fill(tablo);

baglantı.Close();

return tablo;

}

}

}

**Anasayfa.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 WindowsFormsApp1

{

public partial class Anasayfa : Form

{

public Anasayfa()

{

InitializeComponent();

}

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

{

}

private void button2\_Click(object sender, EventArgs e)

{

MüşteriAramaSayfası listele = new MüşteriAramaSayfası();

listele.ShowDialog();

}

private void button1\_Click(object sender, EventArgs e)

{

MüşteriKayıtSayfası ekle = new MüşteriKayıtSayfası();

ekle.ShowDialog();

}

private void button4\_Click(object sender, EventArgs e)

{

ÜrünStokSorgulamaSayfası listele = new ÜrünStokSorgulamaSayfası();

listele.ShowDialog();

}

private void button3\_Click(object sender, EventArgs e)

{

ÜrünKayıtSayfası kayıt = new ÜrünKayıtSayfası();

kayıt.ShowDialog();

}

private void button5\_Click(object sender, EventArgs e)

{

}

}

}

**MüşteriKayıtSayfası.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 WindowsFormsApp1

{

public partial class MüşteriKayıtSayfası : Form

{

candan\_muhendıslık cnd\_muh = new candan\_muhendıslık();

public MüşteriKayıtSayfası()

{

InitializeComponent();

}

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

{

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

string cümle = "insert into MüsteriKayıt(Adsoyad, TelefonNo, MailAdresi, Plaka, AracModel, LPGKitMarka,LPGKitModel,TankMarka,TankModel) values(@AdSoyad, @TelefonNo, @MailAdresi, @Plaka, @AracModel, @LPGKitMarka,@LPGKitModel,@TankMarka,@TankModel)";

SqlCommand komut2 = new SqlCommand();

komut2.Parameters.AddWithValue("@AdSoyad", textad.Text);

komut2.Parameters.AddWithValue("@TelefonNo", texttelefon.Text);

komut2.Parameters.AddWithValue("@MailAdresi", textmail.Text);

komut2.Parameters.AddWithValue("@Plaka", textplaka.Text);

komut2.Parameters.AddWithValue("@AracModel", textmodel.Text);

komut2.Parameters.AddWithValue("@LPGKitMarka", combokitmarka.Text);

komut2.Parameters.AddWithValue("@LPGKitModel", combokitmodel.Text);

komut2.Parameters.AddWithValue("@TankMarka", combotmarka.Text);

komut2.Parameters.AddWithValue("@TankModel", combotmodel.Text);

cnd\_muh.Ekle\_sil\_güncelle(komut2, cümle);

foreach (Control item in Controls) if (item is TextBox) item.Text = "";

string cümle2 = "update Ürün set miktar=miktar-1 where model='" + combokitmodel.Text + "'";

SqlCommand komut3 = new SqlCommand();

cnd\_muh.Ekle\_sil\_güncelle(komut3, cümle2);

foreach (Control item in Controls) if (item is ComboBox) item.Text = "";

string cümle3 = "update Ürün set miktar=miktar-1 where model='" + combotmodel.Text + "'";

SqlCommand komut4 = new SqlCommand();

cnd\_muh.Ekle\_sil\_güncelle(komut4, cümle3);

foreach (Control item in Controls) if (item is ComboBox) item.Text = "";

}

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

{

if (combokitmarka.Text == "Atiker")

{

combokitmodel.Items.Clear();

combokitmodel.Items.Add("Atikfast");

combokitmodel.Items.Add("Gold");

combokitmodel.Items.Add("Grand");

}

else if (combokitmarka.Text == "BRC")

{

combokitmodel.Items.Clear();

combokitmodel.Items.Add("Comfort");

combokitmodel.Items.Add("Platinum");

}

else if (combokitmarka.Text == "Lovato")

{

combokitmodel.Items.Clear();

combokitmodel.Items.Add("OBD 2");

combokitmodel.Items.Add("Smart");

}

else if (combokitmarka.Text == "Prins")

{

combokitmodel.Items.Clear();

combokitmodel.Items.Add("Ecomax");

combokitmodel.Items.Add("Silverline");

combokitmodel.Items.Add("tecnomax");

combokitmodel.Items.Add("VSI 2");

}

}

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

{

if (combotmarka.Text == "Atiker Tank")

{

combotmodel.Items.Clear();

combotmodel.Items.Add("32 LT");

combotmodel.Items.Add("42 LT");

combotmodel.Items.Add("48 LT");

combotmodel.Items.Add("55 LT");

}

else if (combotmodel.Text == "Step Tank")

{

combotmodel.Items.Clear();

combotmodel.Items.Add("32 LT");

combotmodel.Items.Add("42 LT");

combotmodel.Items.Add("48 LT");

combotmodel.Items.Add("55 LT");

}

}

}

}

**MüşteriAramaSayfası.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 WindowsFormsApp1

{

public partial class MüşteriAramaSayfası : Form

{

candan\_muhendıslık candanmuh = new candan\_muhendıslık();

public MüşteriAramaSayfası()

{

InitializeComponent();

}

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

{

YenileListele();

}

private void YenileListele()

{

string cümle = "select \*from MüsteriKayıt";

SqlDataAdapter adtr2 = new SqlDataAdapter();

dataGridView1.DataSource = candanmuh.Listele(adtr2,cümle);

dataGridView1.Columns[0].HeaderText = "AD SOYAD";

dataGridView1.Columns[1].HeaderText = "TELEFON NO";

dataGridView1.Columns[2].HeaderText = "MAİL ADRESİ";

dataGridView1.Columns[3].HeaderText = "PLAKA";

dataGridView1.Columns[4].HeaderText = "ARAÇ MODEL";

dataGridView1.Columns[5].HeaderText = "LPG KİT MARKA";

dataGridView1.Columns[6].HeaderText = "LPG KİT MODEL";

dataGridView1.Columns[7].HeaderText = "TANK MARKA";

dataGridView1.Columns[8].HeaderText = "TANK MODEL";

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

string cümle = "update MüsteriKayıt set Adsoyad=@AdSoyad, TelefonNo=@TelefonNo, MailAdresi=@MailAdresi, Plaka=@Plaka, AracModel=@AracModel, LPGKitMarka=@LPGKitMarka, LPGKitModel=@LPGKitModel, TankMarka=@TankMarka, TankModel=@TankModel";

SqlCommand komut2 = new SqlCommand();

komut2.Parameters.AddWithValue("@AdSoyad", textad.Text);

komut2.Parameters.AddWithValue("@TelefonNo", texttelefon.Text);

komut2.Parameters.AddWithValue("@MailAdresi", textmail.Text);

komut2.Parameters.AddWithValue("@Plaka", textplaka.Text);

komut2.Parameters.AddWithValue("@AracModel", textmodel.Text);

komut2.Parameters.AddWithValue("@LPGKitMarka", combokitmarka.Text);

komut2.Parameters.AddWithValue("@LPGKitModel", combokitmodel.Text);

komut2.Parameters.AddWithValue("@TankMarka", combotmarka.Text);

komut2.Parameters.AddWithValue("@TankModel", combotmodel.Text);

candanmuh.Ekle\_sil\_güncelle(komut2, cümle);

foreach (Control item in Controls) if (item is TextBox) item.Text = "";

foreach (Control item in Controls) if (item is ComboBox) item.Text = "";

}

private void textBox9\_TextChanged(object sender, EventArgs e)

{

string cümle = "select \*from MüsteriKayıt where AdSoyad like '%" + textad.Text + "%'";

SqlDataAdapter adtr2 = new SqlDataAdapter();

dataGridView1.DataSource = candanmuh.Listele(adtr2, cümle);

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

DataGridViewRow satır = dataGridView1.CurrentRow;

textad.Text = satır.Cells[0].Value.ToString();

texttelefon.Text = satır.Cells[1].Value.ToString();

textmail.Text = satır.Cells[2].Value.ToString();

textplaka.Text = satır.Cells[3].Value.ToString();

textmodel.Text = satır.Cells[4].Value.ToString();

combokitmarka.Text = satır.Cells[5].Value.ToString();

combokitmodel.Text = satır.Cells[6].Value.ToString();

combotmarka.Text = satır.Cells[7].Value.ToString();

combotmodel.Text = satır.Cells[8].Value.ToString();

}

private void button3\_Click(object sender, EventArgs e)

{

DataGridViewRow satır = dataGridView1.CurrentRow;

string cümle = "delete from MüsteriKayıt where Plaka='" + satır.Cells["Plaka"].Value.ToString() + "'";

SqlCommand komut2 = new SqlCommand();

candanmuh.Ekle\_sil\_güncelle(komut2, cümle);

foreach (Control item in Controls) if (item is TextBox) item.Text = "";

YenileListele();

}

private void label6\_Click(object sender, EventArgs e)

{

}

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

{

}

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

{ try

{

if (combokitmarka.Text == "Atiker")

{

combokitmodel.Items.Clear();

combokitmodel.Items.Add("Atikfast");

combokitmodel.Items.Add("Gold");

combokitmodel.Items.Add("Grand");

}

else if (combokitmarka.Text == "BRC")

{

combokitmodel.Items.Clear();

combokitmodel.Items.Add("Comfort");

combokitmodel.Items.Add("Platinum");

}

else if (combokitmarka.Text == "Lovato")

{

combokitmodel.Items.Clear();

combokitmodel.Items.Add("OBD 2");

combokitmodel.Items.Add("Smart");

}

else if (combokitmarka.Text == "Prins")

{

combokitmodel.Items.Clear();

combokitmodel.Items.Add("Ecomax");

combokitmodel.Items.Add("Silverline");

combokitmodel.Items.Add("tecnomax");

combokitmodel.Items.Add("VSI 2");

}

}

catch

{

;

}

}

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

{ try {

if (combotmarka.Text == "Atiker Tank")

{

combotmodel.Items.Clear();

combotmodel.Items.Add("32 LT");

combotmodel.Items.Add("42 LT");

combotmodel.Items.Add("48 LT");

combotmodel.Items.Add("55 LT");

}

else if (combotmarka.Text == "Step Tank")

{

combotmodel.Items.Clear();

combotmodel.Items.Add("32 LT");

combotmodel.Items.Add("42 LT");

combotmodel.Items.Add("48 LT");

combotmodel.Items.Add("55 LT");

}

}

catch

{

;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void label8\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

string cümle = "select \*from MüsteriKayıt where AdSoyad like '%" + textad.Text + "%'";

SqlDataAdapter adtr2 = new SqlDataAdapter();

dataGridView1.DataSource = candanmuh.Listele(adtr2, cümle);

}

}

}

**ÜrünKayıtSayfası.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 WindowsFormsApp1

{

public partial class ÜrünKayıtSayfası : Form

{

candan\_muhendıslık candanmuh = new candan\_muhendıslık();

public ÜrünKayıtSayfası()

{

InitializeComponent();

}

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

{

if (comboürün.Text == "LPG KİT")

{

combomarka.Items.Clear();

combomarka.Items.Add("Atiker");

combomarka.Items.Add("BRC");

combomarka.Items.Add("Lovato");

combomarka.Items.Add("Prins");

}

else if (comboürün.Text == "TANK")

{

combomarka.Items.Clear();

combomarka.Items.Add("Atiker Tank");

combomarka.Items.Add("Step Tank");

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

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

{

if (combomarka.Text == "Atiker")

{

combomodel.Items.Clear();

combomodel.Items.Add("Atikfast");

combomodel.Items.Add("Gold");

combomodel.Items.Add("Grand");

}

else if (combomarka.Text == "BRC")

{

combomodel.Items.Clear();

combomodel.Items.Add("Comfort");

combomodel.Items.Add("Platinum");

}

else if (combomarka.Text == "Lovato")

{

combomodel.Items.Clear();

combomodel.Items.Add("OBD 2");

combomodel.Items.Add("Smart");

}

else if (combomarka.Text == "Prins")

{

combomodel.Items.Clear();

combomodel.Items.Add("Ecomax");

combomodel.Items.Add("Silverline");

combomodel.Items.Add("tecnomax");

combomodel.Items.Add("VSI 2");

}

if (combomarka.Text == "Atiker Tank")

{

combomodel.Items.Clear();

combomodel.Items.Add("32 LT");

combomodel.Items.Add("42 LT");

combomodel.Items.Add("48 LT");

combomodel.Items.Add("55 LT");

}

else if (combomarka.Text == "Step Tank")

{

combomodel.Items.Clear();

combomodel.Items.Add("32 LT");

combomodel.Items.Add("42 LT");

combomodel.Items.Add("48 LT");

combomodel.Items.Add("55 LT");

}

}

private void button1\_Click(object sender, EventArgs e)

{

string cümle = "insert into Ürün(ürünadı,marka,model,miktar,tarih) values(@ürünadı,@marka,@model,@miktar,@tarih)";

SqlCommand komut2 = new SqlCommand();

komut2.Parameters.AddWithValue("@ürünadı", comboürün.Text);

komut2.Parameters.AddWithValue("@marka", combomarka.Text);

komut2.Parameters.AddWithValue("@model", combomodel.Text);

komut2.Parameters.AddWithValue("@miktar",int.Parse(textmiktar.Text));

komut2.Parameters.AddWithValue("@tarih",DateTime.Now.ToString());

candanmuh.Ekle\_sil\_güncelle(komut2, cümle);

comboürün.Items.Clear();

combomodel.Items.Clear();

combomarka.Items.Clear();

foreach (Control item in Controls) if (item is TextBox) item.Text = "";

foreach (Control item in Controls) if (item is ComboBox) item.Text = "";

}

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

{

}

}

}

**ÜrünStokSorgulamaSayfası.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 WindowsFormsApp1

{

public partial class ÜrünStokSorgulamaSayfası : Form

{

candan\_muhendıslık candanmuh = new candan\_muhendıslık();

public ÜrünStokSorgulamaSayfası()

{

InitializeComponent();

}

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

{

YenileÜrün();

}

private void YenileÜrün()

{

String cümle = "select \*from Ürün";

SqlDataAdapter adtr2 = new SqlDataAdapter();

dataGridView1.DataSource=candanmuh.Listele(adtr2, cümle);

dataGridView1.Columns[0].HeaderText = "ÜRÜN ADI";

dataGridView1.Columns[1].HeaderText = "MARKA";

dataGridView1.Columns[2].HeaderText = "MODEL";

dataGridView1.Columns[3].HeaderText = "MİKTAR";

dataGridView1.Columns[4].HeaderText = "TARİH";

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

DataGridViewRow satır = dataGridView1.CurrentRow;

comboürün.Text = satır.Cells["ürünadı"].Value.ToString();

combomarka.Text = satır.Cells["marka"].Value.ToString();

combomodel.Text = satır.Cells["model"].Value.ToString();

textmiktar.Text = satır.Cells["miktar"].Value.ToString();

}

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

{

if (comboürün.Text == "LPG KİT")

{

combomarka.Items.Clear();

combomarka.Items.Add("Atiker");

combomarka.Items.Add("BRC");

combomarka.Items.Add("Lovato");

combomarka.Items.Add("Lovato");

}

else if (comboürün.Text == "TANK")

{

combomarka.Items.Clear();

combomarka.Items.Add("Atiker Tank");

combomarka.Items.Add("Step Tank");

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

string cümle = "update Ürün set ürünadı=@ürünadı,marka=@marka,model=@model,miktar=@miktar";

SqlCommand komut2 = new SqlCommand();

komut2.Parameters.AddWithValue("@ürünadı", comboürün.Text);

komut2.Parameters.AddWithValue("@marka", combomarka.Text);

komut2.Parameters.AddWithValue("@model", combomodel.Text);

komut2.Parameters.AddWithValue("@miktar", textmiktar.Text);

candanmuh.Ekle\_sil\_güncelle(komut2, cümle);

foreach (Control item in Controls) if (item is TextBox) item.Text = "";

foreach (Control item in Controls) if (item is ComboBox) item.Text = "";

YenileÜrün();

}

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

{

if (combomarka.Text == "Atiker")

{

combomodel.Items.Clear();

combomodel.Items.Add("Atikfast");

combomodel.Items.Add("Gold");

combomodel.Items.Add("Grand");

}

else if (combomarka.Text == "BRC")

{

combomodel.Items.Clear();

combomodel.Items.Add("Comfort");

combomodel.Items.Add("Platinum");

}

else if (combomarka.Text == "Lovato")

{

combomodel.Items.Clear();

combomodel.Items.Add("OBD 2");

combomodel.Items.Add("Smart");

}

else if (combomarka.Text == "Prins")

{

combomodel.Items.Clear();

combomodel.Items.Add("Ecomax");

combomodel.Items.Add("Silverline");

combomodel.Items.Add("tecnomax");

combomodel.Items.Add("VSI 2");

}

if (combomarka.Text == "Atiker Tank")

{

combomodel.Items.Clear();

combomodel.Items.Add("32 LT");

combomodel.Items.Add("42 LT");

combomodel.Items.Add("48 LT");

combomodel.Items.Add("55 LT");

}

else if (combomarka.Text == "Step Tank")

{

combomodel.Items.Clear();

combomodel.Items.Add("32 LT");

combomodel.Items.Add("42 LT");

combomodel.Items.Add("48 LT");

combomodel.Items.Add("55 LT");

}

}

private void button3\_Click(object sender, EventArgs e)

{

DataGridViewRow satır = dataGridView1.CurrentRow;

string cümle = "delete from Ürün where model='" + satır.Cells["model"].Value.ToString() + "'";

SqlCommand komut2 = new SqlCommand();

candanmuh.Ekle\_sil\_güncelle(komut2, cümle);

foreach (Control item in Controls) if (item is TextBox) item.Text = "";

foreach (Control item in Controls) if (item is ComboBox) item.Text = "";

YenileÜrün();

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

string cümle = "select \*from Ürün where model like '%" + combomodel.Text + "%'";

SqlDataAdapter adtr3 = new SqlDataAdapter();

dataGridView1.DataSource = candanmuh.Listele(adtr3, cümle);

}

}

}

## kULLANIM KILAVUZU

Uygulamamızı açtığımız zaman bizi Anasayfada bulunan 4 adet buton karşılamakta.

Yapmak istediğimiz işleme göre butonlara tıklayarak ilgili sayfaya ulaşabiliriz.

1. Müşteri Kayıt Butonu : Yeni bir müşteri kaydı için butona tıklayınız.

* Satışı gerçekleştirilen her işlem için ayrı kayıt açınız.
* Aynı kişi birden çok araç için kayıt oluşturabilir.
* Satışı gerçekleştirilen bir işlemde değişiklik yapmak için Müşteri Arama Sayfasından güncelleme yapınız.

1. Müşteri Arama Butonu : Daha önceden kayıtlı olan bir müşterinin bilgilerine ulaşmak için butona tıklayınız.Buradan kayıtlı müşteriyi silebilir,müşteri ile ilgili değişiklik yapabilirsiniz.

* Müşteri aramak için arama kısmına müşteri Ad Soyad yazınız
* Müşteriyi silmek için müşteri bilgisine iki kez tıklayın ve sil butonuna tıklayınız
* Değişiklik yapmak için müşteri bilgisinin üzerine iki defa tıklayınız ve güncelle butonuna tıklayarak değişikliği kaydediniz.

1. Ürün Kayıt Butonu : Yeni gelen ürünleri stok bilgilerine eklemek için butonu tıklayınız.

* Ürünün geliş tarihi sistemden o anki tarih olarak otomatik yazılır.
* Stok veya ürünlerle ilgili bir değişiklik yapmak için Stok Sorgulama Sayfasını kullanınız.

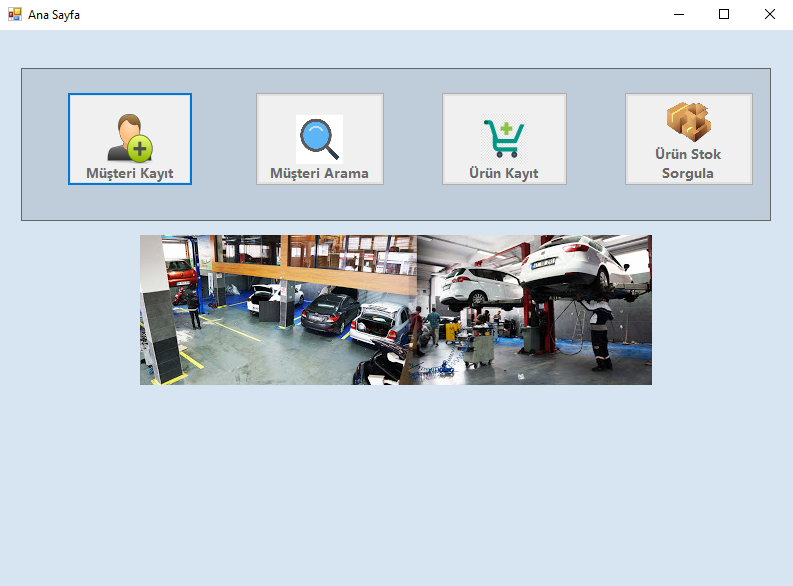
1. Ürün Stok Sorgulama Butonu : Ürünlerin stok durumunu görmek için butona tıklayınız.

* Değişiklik yapmak için ürün bilgisinin üzerine iki defa tıklayınız ve güncelle butonuna tıklayarak değişikliği kaydediniz.
* Ürün silmek için ürünün üzerine iki kere tıklayınız ve sil butonuna tıklayınız.
* Ürün aramak için arama kısmına aradığınız Model’i yazınız

## TANITIM BROŞÜRÜ

**Bir şirket için müşteri ve işlem kaydı yapabilecekleri bir uygulama tasarladık**





**Bu uygulamada yeni gelen müşteri için kayıt oluşturabilir,kayıtları güncelleyebilir,ürün stok takibi yapılabilmekte.**

**Uygulamanın kullanımı kolay olduğundan kullanıcılar rahatlıkla uygulamayı kullanabilirler.İstedikleri işlemleri rahatlıkla yapabilirler.**