

# **ÇİFTLİK YÖNETİM SİSTEMİ (ÇYS)**

**Ferhat ASLAN**

Mezuniyet Projesi Komitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Lefke Avrupa Üniversitesi  
Ağustos 2020

## Approval of the Computer Engineering Department

---

Assoc. Prof. Dr. Hüseyin Ademgil  
Chairman/Director

This is to certify that we have read this graduation project and that in our opinion it is fully adequate, in cope and quality, as an Undergraduate Project.

Assist. Prof. Dr. Zafer Erenel  
Supervisor

## Members of the examining committee

<u>Name</u>	<u>Signature</u>
1. Assoc. Prof. Dr. Hüseyin Ademgil	.....
2. Assist. Prof. Dr. Cem B. Kalyoncu	.....
3. Assist. Prof. Dr. Ferhun Yorgancıoğlu	.....
4. Assist. Prof. Dr. Vesile Evrim	.....
5. Dr. Ersin Çağlar	.....

Date: .....

## Önsöz

### ÇİFTLİK YÖNETİM SİSTEMİ

Ferhat Aslan

Bilgisayar Mühendisliği Bölümü  
Lefke Avrupa Üniversitesi

Danışman: Assist. Prof. Dr. Zafer Erenel

Gelişen teknoloji ile beraber hemen hemen tüm işlemlerimiz bilgisayarlar ve bunlara bağlı olarak yazılımlar tarafından yapılmaktadır. Masaüstü uygulamalar, web uygulamaları ve mobil uygulamalar hayatımızın bir parçası haline gelmiştir.

Eğitimden, sağlığa, hizmet sektöründen, turizm ve diğer bir çok sektörde kullanılan yazılımlar bizlere büyük kolaylıklar sağlamaktadır. Her ne kadar mobil uygulamalar son zamanlarda yoğun olarak kullanılmaya başlansada önümüzdeki bir kaç yıl daha masaüstü ve web uygulamalarının etkinliği sürecektir gibi görünmektedir.

Bu proje ile masaüstü uygulaması ve projenin sonraki aşamalarında ilave edilecek modüllerle mobil uygulama, PDA uygulamaları ve daha fazlası projenin gelişmesine katkı koyacaktır.

## Teşekkürler,

Proje hazırlama esnasında yardımlarını ve desteğini esirgemeyen, proje danışmanım Sayın Asst.Prof.Dr. Zafer Erenel'e teşekkürlerimi sunuyorum.

Ayrıca, Lefke Avrupa Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyeleri ve okutmanlarına göstermiş oldukları ilgi ve paylaşımları için ayrıca teşekkürlerimi sunuyorum.

## İÇİNDEKİLER

Önsöz.....	3
Teşekkürler.....	4
1.Giriş.....	6
1.1 Proje Hakkında.....	6
1.2 Hedefler.....	7
1.3 Sistem Gereksinimleri.....	8
2. ÇYS Çalışma Prensipleri.....	10
3. Tasarım.....	11
3.1. Veritabanı.....	11
3.2. Ara Yüz Tasarımı.....	16
4. Kullanım Klavuzu.....	17
5. Sonuç ve Öneriler.....	25
6. Referanslar.....	26
7. EK – Kaynak Kodlar.....	27

# 1. Giriş

## 1.1. Proje Hakkında

Çiftlik Yönetim Sistemi, daha sonra bu raporda CYS olarak anılacaktır, çiftliklerde yere alan büyükbaş hayvanların takibi, aşılama süreçleri, doğum işlemleri, yemleme bu süt verimliliklerini kontrol eden ve bunları bağlı gerekli raporların alınmasına olanak sağlayan bir yönetim sistemi yazılımıdır.

Hayvan sayısının özellikle fazla olduğu çiftliklerde, yukarıda belirttiğimiz başlıkları takip etmek için çok fazla insan gücüne, efora ve zamana ihtiyaç duyulmaktadır. İnsan gücünün fazlalığı olası hatalara da davetiye çıkarmaktadır. Verilerin toplanıp işlenmesi de ayrıca zaman almaktadır.

Çiftlik çalışanları (hayvan bakıcıları, satın alma departmanı, muhasebe departmanı vb) CYS ile verilere hızlı ve doğru bir şekilde ulaşabilmektedir. Ayrıca, hayvan kontrollerini yapan veterinerler için de büyük kolaylıklar sağlamaktadır.

PDA ve/veya mobil uygulama entegrasyonu ile hayvanların boyun ve/veya kulak kartları okutularak hayvan ile ilgili tüm verilere (doğum, sağım, aşı, ilaç vb) başka bir şeye ihtiyaç duymadan ulaşabilmektedir. Yapılan muayene, aşı ve ilaç kayıtları yine bu cihazlar vasıtasıyla alınarak tüm bilgiler eş zamanlı olarak ana sunuya gönderilmektedir.

## 1.2. Hedefler

Bu projenin amacı iftliklerde hayvanları takip etmek için basit, anlaşılır ve güvenilir bir kurgu uygulayarak zaman kayıplarını önlemeye çalışmaktır. Kullanıcı dostu ara yüz, hata olasılığını yok denecek kadar aza indiren bir alt yapı, windows işletim sistemi tabanlı bir uygulama ve ilerleyen zamanlarda farklı programlama dillerinin de katılarak projenin daha da gelişmesine olanak sağlama düşüncesi en büyük hedefimizdir.

### 1.3. Sistem Gereksinimleri

#### A. Fonksiyonel Gereklikler

Geliştirilen yazılım Andorid İşletim Sisteminde çalışan mobil bir uygulamadır.

##### - Kullanıcı Gereksinimleri

##### Yazılım Geliştirici Adayı

1. Uygulamayı İndir ve Yükle
2. Testi yap

##### - Yazılım Gereksinimleri

- İşletim Sistemi
  - Windows 7 veya daha üstü işletim sistemi
- Uygulama Derleyicisi
  - Visual Studio .NET 2019
- Veritabanı
  - SQL Server Express Edition

#### B. Fonksiyonel Olmayan Gereksinimleri

**Güvenlik:** Uygulama windows tabanlı olduğu için sadece kurulan windows işletim sistemine sahip bilgisayarlarda çalışabilmektedir.

**Performans:** Kullanıcı sayısından bağımsız olmak üzere, istenilen sayıda kullanıcının kullanması herhangi bir performans kaybı yaratmayacaktır.

**Kullanılabilirlik:** Kullanıcı dostu, basit anlaşılabilir bir ara yüz.

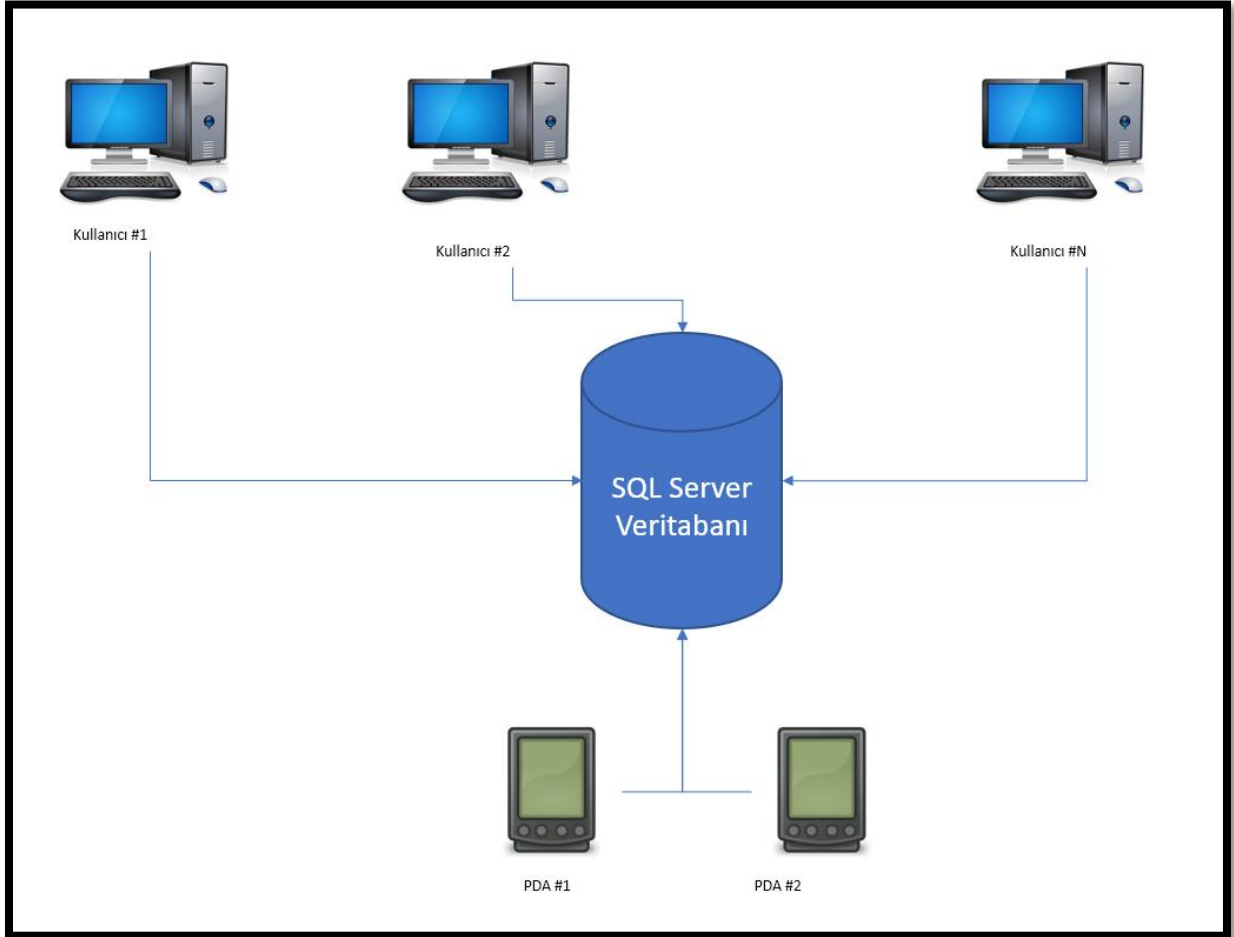
**Geçerlilik:** Uygulama bilgisayarınızda olduğu sürece kullanılabilir, ayrıca güncellemeler otomatik olarak yüklenir.



**Tařınabilirlik:** Uygulama bu ařamada sadece Windows İřletim Sistemi olan (Windows 7.0 ve üzeri) cihazlarda kullanılabilir.

**Geliřtirilebilirlik:** Altyapısı itibari ile geliřtirilmeye açıktır.

## 2. ÇYS Çalışma Prensibi



Şekil 1 – Çiftlik Yönetim Sistemi Genel Yapısı

Şekil 1’de de gösterildiği üzere, projenin ilk aşaması tek kullanıcılı bir bilgisayarda sistem çalışır hale getirilecek, daha sonra da eklenecek modüller ile daha da geliştirilerek son halini alacaktır.



Veritabanımızda ilk etapta 14 adet tablo vardır. Veritabanı açıklamalarında kullanılan PK (Primary Key – Birinci Anahtar), FK (Foreign Key – İkincil Anahtar) anlamına gelmektedir.

**Tablo 1: HAYVAN\_CINS**

Hayvan türlerinin kaydedildiği tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Hayvan Cinslerine verilen otomatik numara, her yeni kayıta yeni bir numara verilir.
CINS_ADI	NVARCHAR(50)		Hayvan tür adı

**Tablo 2: ILAC\_TURU**

Veteriner tarafından hayvanlara verilen ilaçların saklandığı tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Her yeni kayıta verilen otomatik numaradır
ILAC_ADI	NVARCHAR(50)		İlacın adı
KULLANIM_TURU	NVARCHAR(100)		İlacın kullanım bilgisi

**Tablo 3: ASI\_TURU**

Hayvan yapılan aşı türlerinin saklandığı tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Aşı türüne verilen otomatik numara, her yeni kayıta yeni bir numara verilir.
ASI_ADI	NVARCHAR(50)		Aşının adı

**Tablo 4: VETERINER**

Hayvanlara kontrol yapan veteriner bilgilerinin tutulduğu tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Her veterinere verilen otomatik kayıt numarası
VETERINER_ADI	NVARCHAR(100)		Veterinerin adı
ADRES	NVARCHAR(100)		Veterinerin adres bilgisi
TELEFON1	NVARCHAR(20)		Veterinerin 1.telefon numarası
TELEFON2	NVARCHAR(20)		Veterinerin 2.telefon numarası
EPOSTA	NVARCHAR(100)		Veterinerin E-Posta adresi
NOTLAR	NVARCHAR(4000)		Veteriner ile ilgili özel notlar

**Tablo 5: DOGUM\_TURU**

Hayvanların doğum kayıt türlerinin saklandığı tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Her yeni kayıta otomatik yeni bir numara verilir.
DOGUMTURU_ACIKLAMA	NVARCHAR(50)		Doğum türü açıklaması

**Tablo 6: YEM\_TURU**

Hayvanlara verilen yem türlerinin saklandığı tablodur

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Her yeni kayıta yeni bir numara verilir.
YEM_TURU_ACIKLAMA	NVARCHAR(50)		Yem türünün adı

**Tablo 7: YEM\_KARISIM**

1 ya da daha fazla yem türleri kullanılarak oluşturulan yem reçetelerinin saklandığı tablodur

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
KARISIM_SIRA_NO	INT (AUTO)	PK	Her yem karışımı için verilen otomatik artan sayı
ACIKLAMA	NVARCHAR(100)		Yem karışımı açıklaması

**Tablo 8: YEM\_KARISIM\_YEM\_TURU**

Bir yem karışımını oluşturan yem türleri ve miktarlarının saklandığı tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
KARISIM_SIRA_NO	INT	PK + FK	Yem karışım sıra no'su
YEMTURU_SIRANO	INT	PK + FK	Yem türü sıra no'su
ORAN	DECIMAL(10,2)		Karışımda kullanılacak yem türünün oranı

**Tablo 9: HAYVAN\_KAYIT\_TURU**

Yeni hayvan kartların nasıl açıldığının tutulduğu tablodur. (Örnek: Satın Alma, Doğum, vb)

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Her yeni kayıt için verilen otomatik artan sayı
KAYITTURU_ADI	NVARCHAR(100)		Kayıt türü açıklaması

**Tablo 10: HAYVAN**

Hayvan bilgilerinin saklandığı tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
KAYIT_NO	INT (AUTO)	PK	Her yeni kayıt için verilen otomatik artan sayı
KAYIT_TARIHI	DATETIME		Hayvan kartının açıldığı tarih
DEVLET_NO	NVARCHAR(50)		Devlet tarafından verilen kayıt numarası
BOYUN_NO	NVARCHAR(50)		Her hayvan için verilen boyun numarası
KULAK_NO	NVARCHAR(50)		Her hayvan için verilen kulak numarası
CINSIYETI	INT		Hayvanın cinsiyeti
ANNE_KAYIT_NO		FK	Eğer doğum kaydı ise hayvanın anne kayıt numarası
BABA_KAYIT_NO		FK	Eğer doğum kaydı ise hayvanın baba kayıt numarası
HAYVANCINSI_SIRANO		FK	Hayvan cinsi numarası
HAYVANKAYITTURU_SIRANO		FK	Hayvan kayıt türü numarası
DOGUM_TARIHI			Hayvanın doğum tarihi

**Tablo 11: SAGLIK**

Hayvanlar için veteriner tarafından yapılan sağlık muayenelerinin saklandığı tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Her yeni kayıt için verilen otomatik artan sayı
HAYVANKAYIT_NO	INT	FK	Hayvan kayıt numarası
VETERINER_SIRANO	INT	FK	Veteriner kayıt numarası
ISLEM_TARIHI	DATETIME		İşlem tarihi
ISLEM_ACIKLAMA	NVARCHAR(255)		Muayene açıklaması
NOTLAR	NVARCHAR(4000)		Muayene ile ilgili notlar

**Tablo 12: SAGLIK\_ILAC**

Muayene esnasında hayvana verilen ilaç kayıtlarının tutulduğu tablodur

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SAGLIK_SIRA_NO	INT	PK+FK	Muayene numarası
ILACTURU_SIRANO	INT	PK+FK	İlaç türü numarası

**Tablo 13: SUT**

Hayvan süt sağımlarının kayıtlarının tutulduğu tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
SIRA_NO	INT (AUTO)	PK	Her yeni kayıt için verilen otomatik artan sayı
HAYVAN_KAYITNO	INT	FK	Hayvan kayıt numarası
SAGIM_TARIHI	DATETIME		Sağım yapılan tarih
ALINAN_SUT	DECIMAL(10,2)		Alınan süt miktarı

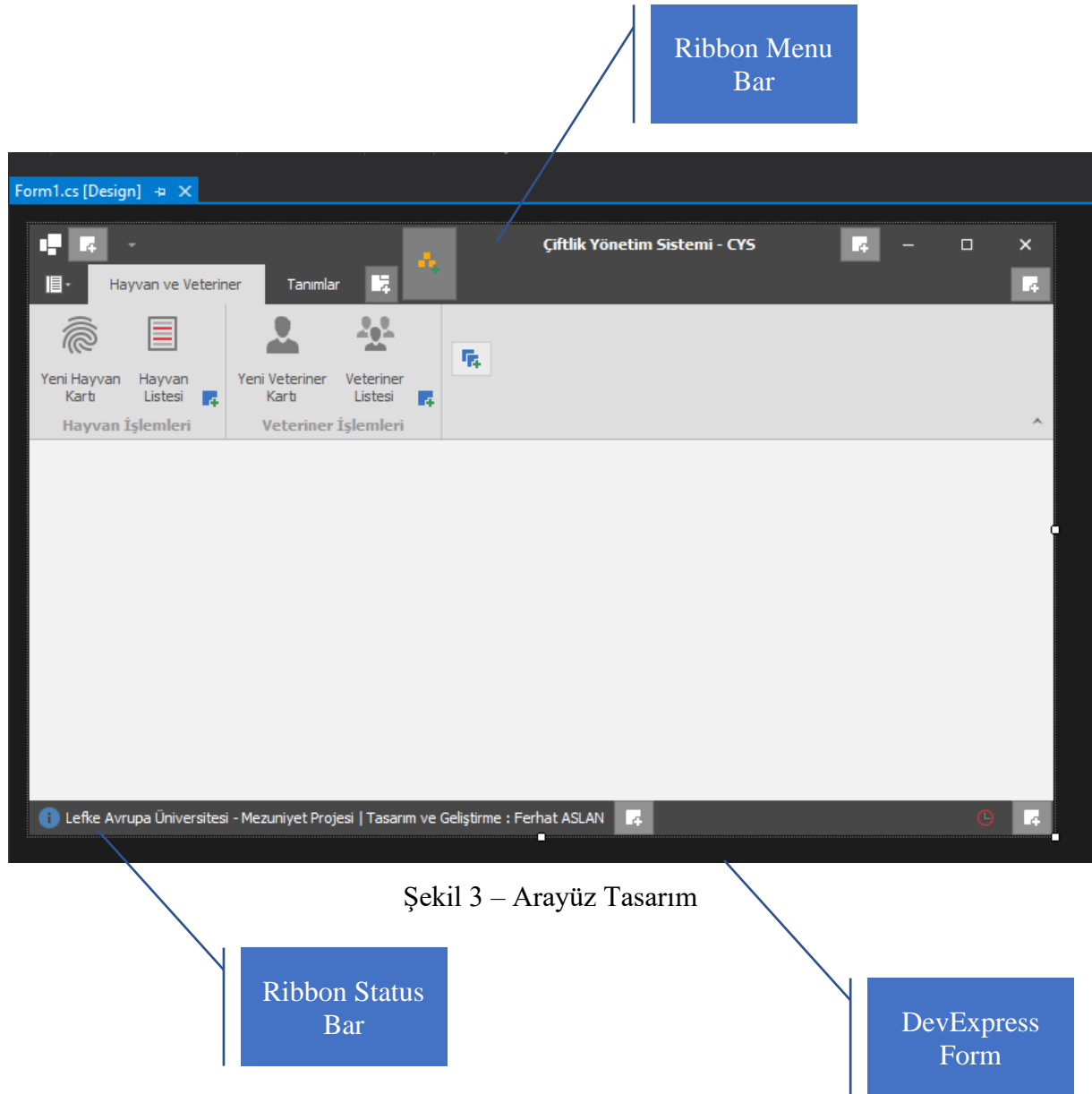
**Tablo 14: HAYVAN\_DOGUM\_TURU**

Doğum yapan hayvanların doğum detaylarının tutulduğu tablodur.

Alan Adı	Alan Tipi	Anahtar Türü	Açıklama
HAYVAN_KAYITNO	INT	PK	Doğum yapan hayvanın kayıt numarası
DOGUMTURU_SIRANO	INT		Doğum türü numarası
ACIKLAMA	NVARCHAR(255)		Doğum ile ilgili açıklamalar
VETERINER_SIRANO	INT		Doğum işlemini yapan veterinerin kayıt numarası

### 3.2.Arayüz Tasarımı

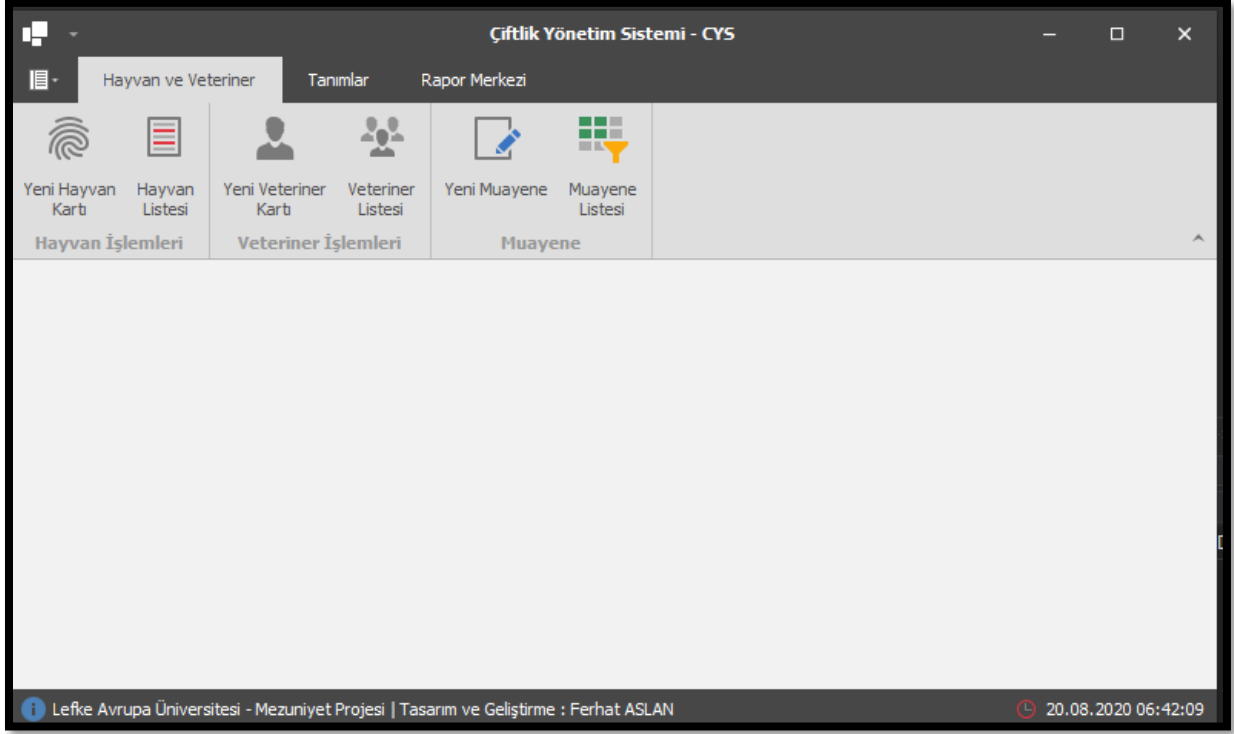
Projenin ara yüzü, Visual Studio 2019 .NET ve DevExpress uygulama geliştirme aracı kullanılarak hazırlanmıştır. Tüm kodlar C# dili ile yazılmıştır. Ayrıca, veritabanı ile ilgili haberleşmede .NET Entity Framework teknolojisi kullanılmıştır.



Şekil 3 – Arayüz Tasarım



## 4. Kullanım Klavuzu



Şekil 4 – ÇYS Ana Ekranı

Şekil 5 – Yeni Hayvan Kartı



**Veteriner Listesi**

Veteriner Adı	Adres	Telefon #1	Telefon #2	E-Posta	Notlar
ABC	ABC	ABC	ABC	ABC	ABC

Yeni Kartı Aç Sil Muayene İşlemleri Listeyi Yazdır Kapat

Şekil 8 – Veteriner Listesi

**Muayene İşlemleri**

Veteriner:

Hayvan Seçiniz:

Muayene Tarihi:

Muayene Açıklama:

Notlar:

Kaydet İlaç Girişi Aşı Girişi Muayene Raporunu Yazdır Kapat

Şekil 9 – Yeni Muayene Kartı

Muayene Listesi

Kayıt No	Muayene Tarihi	Açıklama	Notlar	Hayvan	Veteriner
=	=	ABC	ABC	=	ABC

Yeni Muayene

Muayene Kartını Aç

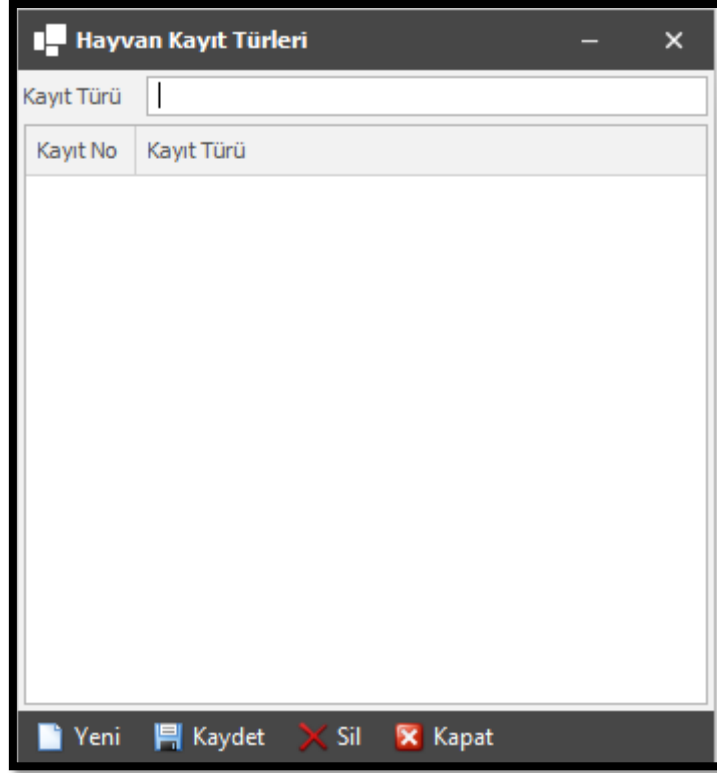
İlaç İşlemleri

Aşı İşlemleri

Muayene Kaydını Sil

Kapat

Şekil 10 – Muayene Listesi



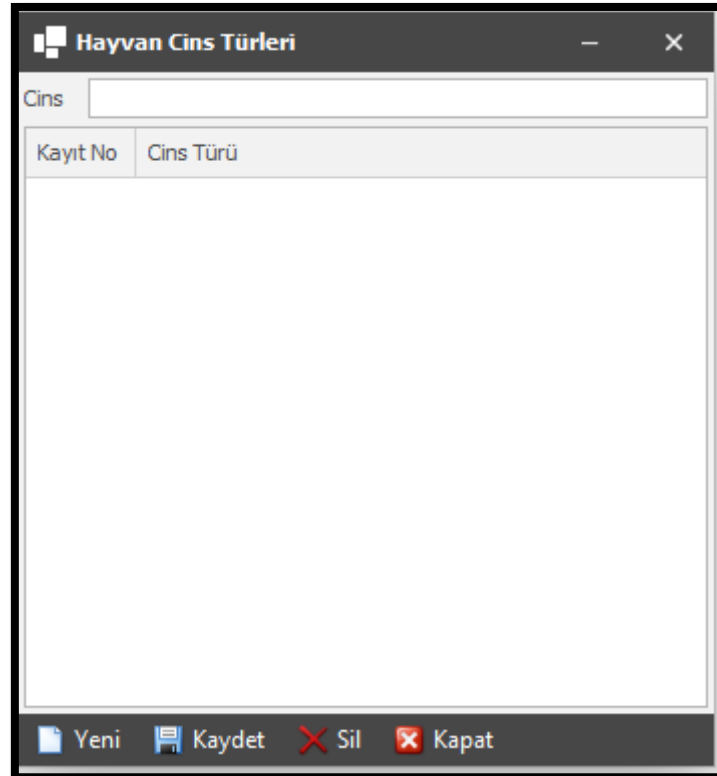
**Hayvan Kayıt Türleri**

Kayıt Türü

Kayıt No	Kayıt Türü
----------	------------

Yeni Kaydet Sil Kapat

Şekil 11 – Hayvan Kayıt Türleri



**Hayvan Cins Türleri**

Cins

Kayıt No	Cins Türü
----------	-----------

Yeni Kaydet Sil Kapat

Şekil 12 – Hayvan Cinsleri

**Aşı Türleri**

Aşı Adı

Kayıt No	Aşı Adı
----------	---------

Yeni Kaydet Sil Kapat

Şekil 13 – Aşı Türleri

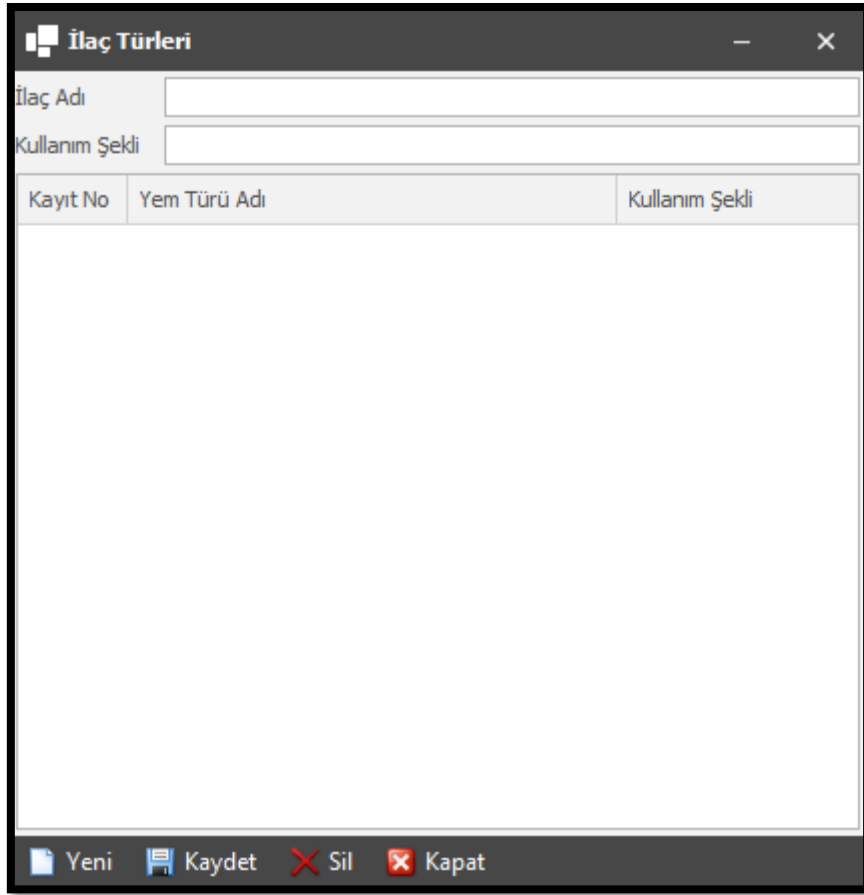
**Hayvan Doğum Türleri**

Doğum Türü

Kayıt No	Doğum Türü Adı
----------	----------------

Yeni Kaydet Sil Kapat

Şekil 14 – Doğum Türleri



**İlaç Türleri**

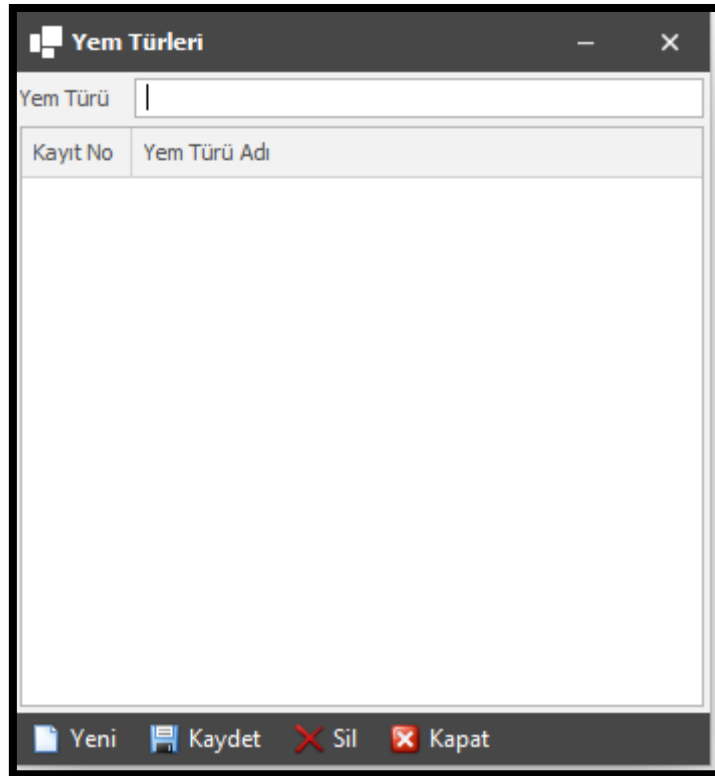
İlaç Adı

Kullanım Şekli

Kayıt No	Yem Türü Adı	Kullanım Şekli
----------	--------------	----------------

Yeni Kaydet Sil Kapat

Şekil 15 – İlaç Türleri



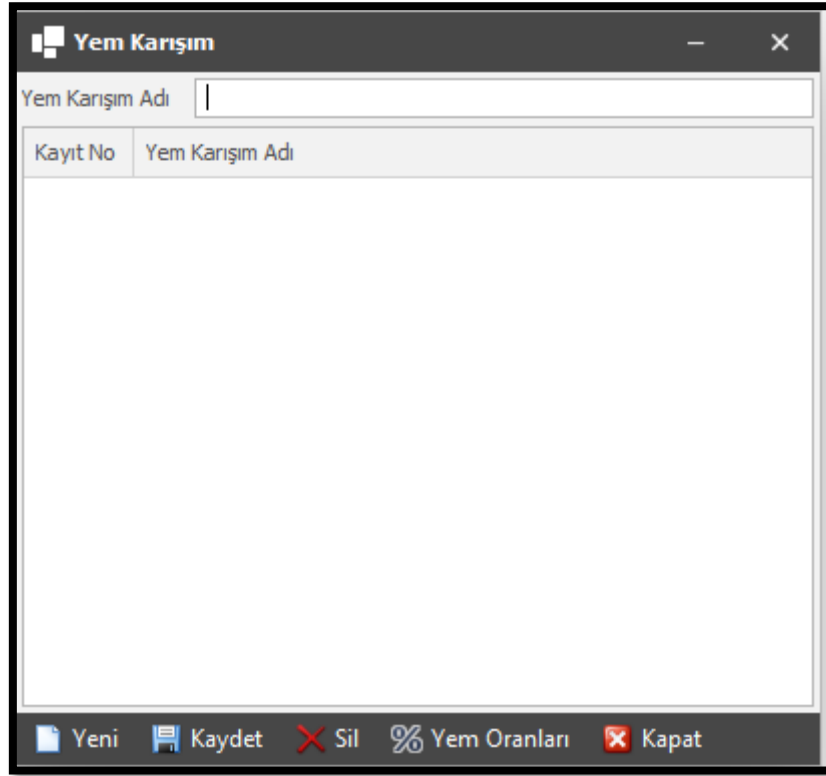
**Yem Türleri**

Yem Türü

Kayıt No	Yem Türü Adı
----------	--------------

Yeni Kaydet Sil Kapat

Şekil 16 – Yem Türleri



Yem Karışım

Yem Karışım Adı

Kayıt No	Yem Karışım Adı
----------	-----------------

Yeni Kaydet Sil Yem Oranları Kapat

Şekil 17 – Yem Karışım Türü



## 5. Sonuç ve Öneriler

Geliştirmiş olduğumuz masaüstü uygulaması Çiftlik yönetenlerinin bir çok işini hızlı, güvenilir ve hatasız yapmasına olanak sağlayacaktır. Bu uygulama ile beraber, daha az insan gücü, hayvan bilgilerine daha hızlı erişim olanağı vardır.

Aynı zamanda, uygulamaya eklenecek ilave modül ya da modüllere proje dah da büyüyecek farklı alanlardaki talepler de bu sayaele değerelecektir.

## 6. Referanslar

<https://tr.wikipedia.org/>

<https://blog.koddit.com/>

<https://devepxress.com>

## 7. EK – Kaynak Kodlar

### Form1.cs

```

using CYSLAU.Formlar;
using CYSLAU.Formlar.Hayvan;
using CYSLAU.Formlar.Veteriner;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace CYSLAU
{
    public partial class Form1 : DevExpress.XtraBars.Ribbon.RibbonForm
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            bsiTarih.Caption = DateTime.Now.ToString();
        }

        private void barButtonItem3_ItemClick(object sender,
        DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmAsiTur frm = new frmAsiTur();
            frm.ShowDialog();
        }

        private void barButtonItem4_ItemClick(object sender,
        DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmHayvanKayitTur frm = new frmHayvanKayitTur();
            frm.ShowDialog();
        }

        private void barButtonItem5_ItemClick(object sender,
        DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmHayvanCins frm = new frmHayvanCins();
            frm.ShowDialog();
        }
    }
}

```

```

        private void barButtonItem6_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmDogumTuru frm = new frmDogumTuru();
            frm.ShowDialog();
        }

```

```

        private void barButtonItem7_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmYemTuru frm = new frmYemTuru();
            frm.ShowDialog();
        }

```

```

        private void barButtonItem8_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmIlacTuru frm = new frmIlacTuru();
            frm.ShowDialog();
        }

```

```

        private void barButtonItem9_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmYemKarisim frm = new frmYemKarisim();
            frm.ShowDialog();
        }

```

```

        private void barButtonItem1_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmHayvan frm = new frmHayvan();
            frm.ShowDialog();
        }

```

```

        private void barButtonItem2_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmHayvanListesi frm = new frmHayvanListesi();
            frm.ShowDialog();
        }

```

```

        private void barButtonItem10_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmVeteriner frm = new frmVeteriner(null, null);
            frm.ShowDialog();
        }

```

```

        private void barButtonItem11_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
        {
            frmVeterinerList frm = new frmVeterinerList();
            frm.ShowDialog();
        }
    }
}

```

### **frmAsiTur.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;

namespace CYSLAU.Formlar
{
    public partial class frmAsiTur : DevExpress.XtraEditors.XtraForm
    {
        ASI_TURU asiTuru;

        public frmAsiTur()
        {
            InitializeComponent();

            Listele();
        }
        private void EkraniTemizle()
        {
            /* Ekranı Temizlemek için kullanılacak fonksiyon */
            txtAsiAdi.Text = string.Empty;
            asiTuru = null; /* Yeni kayıt için asiTuru nesnesini null yapar */
        }

        private void Listele()
        {
            /* Veritabanından kayıtları okur ve listeler */
            gcAsiTuru.DataSource= CYS.model.ASI_TURU.Select(x => x).ToList();
        }
    }
}

```

```

private void barButtonItem1_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    EkraniTemizle();
}

private void barButtonItem2_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    /* Eğer yeni bir kayıt ise, kayıt işlemi yap
    * var olan bir kayıt ise güncelleme yap */
    if (asiTuru == null)
    {
        /* Yeni Kayıt */
        asiTuru = new ASI_TURU();
        asiTuru.ASI_ADI = txtAsiAdi.Text;
        CYS.model.ASI_TURU.Add(asiTuru);
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Aşı türü kaydı yapıldı", "Bilgi", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    else
    {
        /* Var olan bir kayıt */
        asiTuru.ASI_ADI = txtAsiAdi.Text;
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Aşı türü kaydı güncellendi", "Bilgi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    Listele();
    EkraniTemizle();
}

private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    /* Seçilen kaydı sil */
    if (asiTuru == null) return;
    CYS.model.ASI_TURU.Remove(asiTuru);
    CYS.model.SaveChanges();
    XtraMessageBox.Show("Aşı türü kaydı silindi", "Bilgi", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
    Listele();
    EkraniTemizle();
}

private void gvAsiTuru_Click(object sender, EventArgs e)
{
    asiTuru = gvAsiTuru.GetFocusedRow() as ASI_TURU;
    if (asiTuru != null)

```

```

        {
            txtAsiAdi.Text = asiTuru.ASI_ADI;
        }
    }

    private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        Close();
    }
}

```

### **frmDogumTuru.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;

namespace CYSLAU.Formlar
{
    public partial class frmDogumTuru : DevExpress.XtraEditors.XtraForm
    {
        DOGUM_TURU kayıtTuru;

        public frmDogumTuru()
        {
            InitializeComponent();
            Listele();
        }

        private void EkraniTemizle()
        {
            /* Ekranı Temizlemek için kullanılacak fonksiyon */
            txtAsiAdi.Text = string.Empty;
            kayıtTuru = null; /* Yeni kayıt için kayıtTuru nesnesini null yapar */
        }

        private void Listele()
        {
            /* Veritabanından kayıtları okur ve listeler */
            gc.DataSource = CYS.model.DOGUM_TURU.Select(x => x).ToList();
        }
    }
}

```

```

    }

    private void barButtonItem1_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        EkraniTemizle();
    }

    private void barButtonItem2_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        /* Eğer yeni bir kayıt ise, kayıt işlemi yap
        * var olan bir kayıt ise güncelleme yap */
        if (kayitTuru == null)
        {
            /* Yeni Kayıt */
            kayitTuru = new DOGUM_TURU();
            kayitTuru.DOGUMTURU_ACIKLAMA = txtAsiAdi.Text;
            CYS.model.DOGUM_TURU.Add(kayitTuru);
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Doğum türü kaydı yapıldı", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            /* Var olan bir kayıt */
            kayitTuru.DOGUMTURU_ACIKLAMA = txtAsiAdi.Text;
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Doğum türü kaydı güncellendi", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        Listele();
        EkraniTemizle();
    }

    private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        /* Seçilen kaydı sil */
        if (kayitTuru == null) return;
        CYS.model.DOGUM_TURU.Remove(kayitTuru);
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Doğum türü kaydı silindi", "Bilgi", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        Listele();
        EkraniTemizle();
    }

    private void gvKayitTuru_Click(object sender, EventArgs e)
    {

```



```

        kayitTuru = gv.GetFocusedRow() as DOGUM_TURU;
        if (kayitTuru != null)
        {
            txtAsiAdi.Text = kayitTuru.DOGUMTURU_ACIKLAMA;
        }
    }

    private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        Close();
    }
}

```

### **frmHayvanCins.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;
using CYSLAU.Formlar.Hayvan;

namespace CYSLAU.Formlar
{
    public partial class frmHayvanCins : DevExpress.XtraEditors.XtraForm
    {
        HAYVAN_CINSI kayitTuru;
        object caller;
        public frmHayvanCins(object caller=null)
        {
            InitializeComponent();
            this.caller = caller;
           >Listele();
        }
        private void EkraniTemizle()
        {
            /* Ekrani Temizlemek için kullanılacak fonksiyon */
            txtAsiAdi.Text = string.Empty;
            kayitTuru = null; /* Yeni kayıt için kayitTuru nesnesini null yapar */
        }
    }
}

```

```

private void Listele()
{
    /* Veritabanından kayıtları okur ve listeler */
    gc.DataSource= CYS.model.HAYVAN_CINSI.Select(x => x).ToList();
}

private void barButtonItem1_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    EkraniTemizle();
}

private void barButtonItem2_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    /* Eğer yeni bir kayıt ise, kayıt işlemi yap
    * var olan bir kayıt ise güncelleme yap */
    if (kayitTuru == null)
    {
        /* Yeni Kayıt */
        kayitTuru = new HAYVAN_CINSI();
        kayitTuru.CINS_ADI = txtAsiAdi.Text;
        CYS.model.HAYVAN_CINSI.Add(kayitTuru);
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Hayvan Cins türü kaydı yapıldı", "Bilgi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        /* Var olan bir kayıt */
        kayitTuru.CINS_ADI = txtAsiAdi.Text;
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Hayvan Cinsi türü kaydı güncellendi", "Bilgi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    Guncelle();
    Listele();
    EkraniTemizle();
}

private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    /* Seçilen kaydı sil */
    if (kayitTuru == null) return;
    CYS.model.HAYVAN_CINSI.Remove(kayitTuru);
    CYS.model.SaveChanges();
    XtraMessageBox.Show("Hayvan Cins türü kaydı silindi", "Bilgi",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    Guncelle();
}

```

```

       >Listele();
       >EkraniTemizle();
    }

    private void gvKayitTuru_Click(object sender, EventArgs e)
    {
       >kayitTuru = gv.GetFocusedRow() as HAYVAN_CINSI;
       >if (kayitTuru != null)
        {
           >>txtAsiAdi.Text = kayitTuru.CINS_ADI;
        }
    }

    private void barButtonItem4_ItemClick(object sender,
    DevExpress.XtraBars.ItemClickEventArgs e)
    {
       >>Close();
    }

    private void Guncelle()
    {
       >>/* Diğer formları güncellemek için kullanılır */
       >>if (caller == null) return;

       >>frmHayvan frm = caller as frmHayvan;
       >>if (frm != null) frm.HayvanCinsleriniGetir();
    }
}

```

### **frmHayvanKayitTur.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;
using CYSLAU.Formlar.Hayvan;

namespace CYSLAU.Formlar
{
    public partial class frmHayvanKayitTur : DevExpress.XtraEditors.XtraForm

```

```

{
    HAYVAN_KAYIT_TURU kayitTuru;
    object caller;
    public frmHayvanKayitTur(object caller=null)
    {
        InitializeComponent();
        this.caller = caller;
        Listele();
    }
    private void EkraniTemizle()
    {
        /* Ekrani Temizlemek için kullanılacak fonksiyon */
        txtAsiAdi.Text = string.Empty;
        kayitTuru = null; /* Yeni kayıt için kayitTuru nesnesini null yapar */
    }

    private void Listele()
    {
        /* Veritabanından kayıtları okur ve listeler */
        gc.DataSource= CYS.model.HAYVAN_KAYIT_TURU.Select(x => x).ToList();
    }

    private void barButtonItem1_ItemClick(object sender,
    DevExpress.XtraBars.ItemClickEventArgs e)
    {
        EkraniTemizle();
    }

    private void barButtonItem2_ItemClick(object sender,
    DevExpress.XtraBars.ItemClickEventArgs e)
    {
        /* Eğer yeni bir kayıt ise, kayıt işlemi yap
        * var olan bir kayıt ise güncelleme yap */
        if (kayitTuru == null)
        {
            /* Yeni Kayıt */
            kayitTuru = new HAYVAN_KAYIT_TURU();
            kayitTuru.KAYIT_TURU_ADU = txtAsiAdi.Text;
            CYS.model.HAYVAN_KAYIT_TURU.Add(kayitTuru);
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Kayıt türü kaydı yapıldı", "Bilgi",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            /* Var olan bir kayıt */
            kayitTuru.KAYIT_TURU_ADU = txtAsiAdi.Text;
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Kayıt türü kaydı güncellendi", "Bilgi",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}

```

```

    }
    Guncelle();
    Listele();
    EkraniTemizle();

}

private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    /* Seçilen kaydı sil */
    if (kayitTuru == null) return;
    CYS.model.HAYVAN_KAYIT_TURU.Remove(kayitTuru);
    CYS.model.SaveChanges();
    XtraMessageBox.Show("Kayıt türü kaydı silindi", "Bilgi", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
    Guncelle();
    Listele();
    EkraniTemizle();
}

private void gvKayitTuru_Click(object sender, EventArgs e)
{
    kayitTuru = gv.GetFocusedRow() as HAYVAN_KAYIT_TURU;
    if (kayitTuru != null)
    {
        txtAsiAdi.Text = kayitTuru.KAYIT_TURU_ADU;
    }
}

private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    Close();
}
private void Guncelle()
{
    /* Diğer formları güncellemek için kullanılır */
    if (caller == null) return;

    frmHayvan frm = caller as frmHayvan;
    if (frm != null) frm.HayvanKayitTurleriniGetir();
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;

namespace CYSLAU.Formlar
{
    public partial class frmIlacTuru : DevExpress.XtraEditors.XtraForm
    {
        ILAC_TURU kayitTuru;

        public frmIlacTuru()
        {
            InitializeComponent();
            Listele();
        }
        private void EkraniTemizle()
        {
            /* Ekrani Temizlemek için kullanılacak fonksiyon */
            txtAdi.Text = string.Empty;
            txtKullanimSekli.Text = string.Empty;
            kayitTuru = null; /* Yeni kayıt için kayitTuru nesnesini null yapar */
        }

        private void Listele()
        {
            /* Veritabanından kayıtları okur ve listeler */
            gc.DataSource = CYS.model.ILAC_TURU.Select(x => x).ToList();
        }

        private void barButtonItem1_ItemClick(object sender,
        DevExpress.XtraBars.ItemClickEventArgs e)
        {
            EkraniTemizle();
        }

        private void barButtonItem2_ItemClick(object sender,
        DevExpress.XtraBars.ItemClickEventArgs e)
        {
            /* Eğer yeni bir kayıt ise, kayıt işlemi yap
            * var olan bir kayıt ise güncelleme yap */
            if (kayitTuru == null)

```

```

{
    /* Yeni Kayıt */
    kayitTuru = new ILAC_TURU();
    kayitTuru.ILAC_ADI = txtAdi.Text;
    kayitTuru.KULLANIM_TURU = txtKullanımSekli.Text;
    CYS.model.ILAC_TURU.Add(kayitTuru);
    CYS.model.SaveChanges();
    XtraMessageBox.Show("İlaç Türü kaydı yapıldı", "Bilgi",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
    /* Var olan bir kayıt */
    kayitTuru.ILAC_ADI = txtAdi.Text;
    kayitTuru.KULLANIM_TURU = txtKullanımSekli.Text;
    CYS.model.SaveChanges();
    XtraMessageBox.Show("İlaç Türü kaydı güncellendi", "Bilgi",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
Listele();
EkraniTemizle();
}

private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    /* Seçilen kaydı sil */
    if (kayitTuru == null) return;
    CYS.model.ILAC_TURU.Remove(kayitTuru);
    CYS.model.SaveChanges();
    XtraMessageBox.Show("İlaç Türü kaydı silindi", "Bilgi", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
    Listele();
    EkraniTemizle();
}

private void gvKayitTuru_Click(object sender, EventArgs e)
{
    kayitTuru = gv.GetFocusedRow() as ILAC_TURU;
    if (kayitTuru != null)
    {
        txtAdi.Text = kayitTuru.ILAC_ADI;
        txtKullanımSekli.Text = kayitTuru.KULLANIM_TURU;
    }
}

private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    Close();
}

```

```

    }

    private void gc_Click(object sender, EventArgs e)
    {

    }
}

```

### **frmYemKarisim.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;

namespace CYSLAU.Formlar
{
    public partial class frmYemKarisim : DevExpress.XtraEditors.XtraForm
    {
        YEM_KARISIM kayıtTuru;
        public frmYemKarisim()
        {
            InitializeComponent();
            Listele();
        }
        private void EkraniTemizle()
        {
            /* Ekranı Temizlemek için kullanılacak fonksiyon */
            txtAsiAdi.Text = string.Empty;
            kayıtTuru = null; /* Yeni kayıt için kayıtTuru nesnesini null yapar */
        }

        private void Listele()
        {
            /* Veritabanından kayıtları okur ve listeler */
            gc.DataSource = CYS.model.YEM_KARISIM.Select(x => x).ToList();
        }

        private void barButtonItem1_ItemClick(object sender,
        DevExpress.XtraBars.ItemClickEventArgs e)
        {

```



```

        EkraniTemizle();
    }

    private void barButtonItem2_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        /* Eğer yeni bir kayıt ise, kayıt işlemi yap
        * var olan bir kayıt ise güncelleme yap */
        if (kayitTuru == null)
        {
            /* Yeni Kayıt */
            kayitTuru = new YEM_KARISIM();
            kayitTuru.ACIKLAMA = txtAsiAdi.Text;
            CYS.model.YEM_KARISIM.Add(kayitTuru);
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Yem Karışım kaydı yapıldı", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            /* Var olan bir kayıt */
            kayitTuru.ACIKLAMA = txtAsiAdi.Text;
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Yem Karışım kaydı güncellendi", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        Listele();
        EkraniTemizle();
    }

    private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        /* Seçilen kaydı sil */
        if (kayitTuru == null) return;
        CYS.model.YEM_KARISIM.Remove(kayitTuru);
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Yem Karışım kaydı silindi", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        Listele();
        EkraniTemizle();
    }

    private void gvKayitTuru_Click(object sender, EventArgs e)
    {
        kayitTuru = gv.GetFocusedRow() as YEM_KARISIM;
        if (kayitTuru != null)
        {
            txtAsiAdi.Text = kayitTuru.ACIKLAMA;
        }
    }

```

```

    }

    private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        Close();
    }

    private void gc_Click(object sender, EventArgs e)
    {

    }

    private void barButtonItem5_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        kayitTuru = gv.GetFocusedRow() as YEM_KARISIM;
        if (kayitTuru != null)
        {
            frmYemKarisimOran frm = new frmYemKarisimOran(kayitTuru);
            frm.ShowDialog();
        }
    }
}

```

### **frmYemKarisimOran.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;

namespace CYSLAU.Formlar
{
    public partial class frmYemKarisimOran : DevExpress.XtraEditors.XtraForm
    {
        YEM_KARISIM_YEM_TURU kayitTuru;
        YEM_KARISIM karisim;
        public frmYemKarisimOran(YEM_KARISIM karisim=null)
        {
            InitializeComponent();

```

```

        YemTurleriniGetir();
        this.karisim = karisim;
        txtKarisimAdi.Text = karisim.ACIKLAMA;
        Listele();
    }
    private void EkraniTemizle()
    {
        /* Ekrani Temizlemek için kullanılacak fonksiyon */
        txtKarisimAdi.Text = karisim.ACIKLAMA;
        seOran.Value = 0;
        cmbYemTuru.ReadOnly = false;
        cmbYemTuru.EditValue = null;
        kayitTuru = null; /* Yeni kayıt için kayitTuru nesnesini null yapar */
    }

    private void Listele()
    {
        /* Veritabanından kayıtları okur ve listeler */
        gc.DataSource = CYS.model.YEM_KARISIM_YEM_TURU.Select(x => x).ToList();
    }

    private void barButtonItem1_ItemClick(object sender,
    DevExpress.XtraBars.ItemClickEventArgs e)
    {
        EkraniTemizle();
    }

    private void YemTurleriniGetir()
    {
        List<YEM_TURU> YemTurleri = CYS.model.YEM_TURU.Select(x => x).ToList();
        cmbYemTuru.Properties.Items.Clear();
        cmbYemTuru.Properties.Items.AddRange(YemTurleri.ToArray());
    }

    private void barButtonItem2_ItemClick(object sender,
    DevExpress.XtraBars.ItemClickEventArgs e)
    {
        /* Eğer yeni bir kayıt ise, kayıt işlemi yap
        * var olan bir kayıt ise güncelleme yap */
        if (kayitTuru == null)
        {
            /* Yeni Kayıt */
            kayitTuru = new YEM_KARISIM_YEM_TURU();
            kayitTuru.KARISIM_SIRANO = karisim.KARISIM_SIRANO;
            kayitTuru.ORAN = seOran.Value;
            kayitTuru.YEMTURU_SIRANO = (cmbYemTuru.SelectedItem as
            YEM_TURU).SIRANO;
            // kayitTuru.ACIKLAMA = txtAsiAdi.Text;
            CYS.model.YEM_KARISIM_YEM_TURU.Add(kayitTuru);
            CYS.model.SaveChanges();
        }
    }

```

```

        XtraMessageBox.Show("Yem Oran kaydı yapıldı", "Bilgi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        /* Var olan bir kayıt */
        kayitTuru.ORAN = seOran.Value;
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Yem Oran kaydı güncellendi", "Bilgi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    Listele();
    EkraniTemizle();
}

private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    /* Seçilen kaydı sil */
    if (kayitTuru == null) return;
    CYS.model.YEM_KARISIM_YEM_TURU.Remove(kayitTuru);
    CYS.model.SaveChanges();
    XtraMessageBox.Show("Yem Oran kaydı silindi", "Bilgi", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
    Listele();
    EkraniTemizle();
}

private void gvKayitTuru_Click(object sender, EventArgs e)
{
    kayitTuru = gv.GetFocusedRow() as YEM_KARISIM_YEM_TURU;
    if (kayitTuru != null)
    {
        cmbYemTuru.EditValue = kayitTuru.YEM_TURU;
        seOran.Value = kayitTuru.ORAN;
        cmbYemTuru.ReadOnly = true;
    }
}

private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    Close();
}

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

```

```
}
```

### **frmYemTuru.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;

namespace CYSLAU.Formlar
{
    public partial class frmYemTuru : DevExpress.XtraEditors.XtraForm
    {
        YEM_TURU kayıtTuru;

        public frmYemTuru()
        {
            InitializeComponent();
            Listele();
        }

        private void EkraniTemizle()
        {
            /* Ekranı Temizlemek için kullanılacak fonksiyon */
            txtAsiAdi.Text = string.Empty;
            kayıtTuru = null; /* Yeni kayıt için kayıtTuru nesnesini null yapar */
        }

        private void Listele()
        {
            /* Veritabanından kayıtları okur ve listeler */
            gc.DataSource = CYS.model.YEM_TURU.Select(x => x).ToList();
        }

        private void barButtonItem1_ItemClick(object sender,
        DevExpress.XtraBars.ItemClickEventArgs e)
        {
            EkraniTemizle();
        }

        private void barButtonItem2_ItemClick(object sender,
        DevExpress.XtraBars.ItemClickEventArgs e)
        {

```

```

/* Eğer yeni bir kayıt ise, kayıt işlemi yap
* var olan bir kayıt ise güncelleme yap */
if (kayitTuru == null)
{
    /* Yeni Kayıt */
    kayitTuru = new YEM_TURU();
    kayitTuru.YEM_TURU_ACIKLAMA = txtAsiAdi.Text;
    CYS.model.YEM_TURU.Add(kayitTuru);
    CYS.model.SaveChanges();
    XtraMessageBox.Show("Yem Türü kaydı yapıldı", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
    /* Var olan bir kayıt */
    kayitTuru.YEM_TURU_ACIKLAMA = txtAsiAdi.Text;
    CYS.model.SaveChanges();
    XtraMessageBox.Show("Yem Türü kaydı güncellendi", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
Listele();
EkraniTemizle();
}

private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    /* Seçilen kaydı sil */
    if (kayitTuru == null) return;
    CYS.model.YEM_TURU.Remove(kayitTuru);
    CYS.model.SaveChanges();
    XtraMessageBox.Show("Yem Türü kaydı silindi", "Bilgi", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    Listele();
    EkraniTemizle();
}

private void gvKayitTuru_Click(object sender, EventArgs e)
{
    kayitTuru = gv.GetFocusedRow() as YEM_TURU;
    if (kayitTuru != null)
    {
        txtAsiAdi.Text = kayitTuru.YEM_TURU_ACIKLAMA;
    }
}

private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    Close();
}

```

```

    }

    private void gc_Click(object sender, EventArgs e)
    {

    }
}

```

### **frmHayvan.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;

namespace CYSLAU.Formlar.Hayvan
{
    public partial class frmHayvan : DevExpress.XtraEditors.XtraForm
    {
        private HAYVAN hayvan;
        private object caller;
        public frmHayvan(HAYVAN hayvan=null,object caller=null)
        {
            InitializeComponent();
            this.hayvan = hayvan;
            this.caller = caller;
            AnneBilgileriniGetir();
            BabaBilgileriniGetir();
            HayvanCinsleriniGetir();
            HayvanKayitTurleriniGetir();
            if (hayvan == null) FormuTemizle();
            else BilgileriGoster();
        }

        public void AnneBilgileriniGetir()
        {
            List<HAYVAN> AnneListesi= CYS.model.HAYVANs.Where(x =>
x.CINSIYETI.Value==1).ToList();
            cmbAnneKayitNo.Properties.Items.Clear();
            cmbAnneKayitNo.Properties.Items.AddRange(ArneListesi.ToArray());
        }
    }
}

```

```

public void BabaBilgileriniGetir()
{
    List<HAYVAN> BabaListesi = CYS.model.HAYVANs.Where(x =>
x.CINSIYETI.Value == 0).ToList();
    cmbBabaKayitNo.Properties.Items.Clear();
    cmbBabaKayitNo.Properties.Items.AddRange(BabaListesi.ToArray());
}

public void HayvanCinsleriniGetir()
{
    cmbHayvanCinsi.Properties.Items.Clear();
    cmbHayvanCinsi.Properties.Items.AddRange(CYS.model.HAYVAN_CINSI.Select(x
=> x).ToArray());
}

public void HayvanKayitTurleriniGetir()
{
    cmbHayvanKayitTuru.Properties.Items.Clear();

cmbHayvanKayitTuru.Properties.Items.AddRange(CYS.model.HAYVAN_KAYIT_TURU.S
elect(x => x).ToArray());
}
private void FormuTemizle()
{
    hayvan = null;
    txtBoyunNo.Text = string.Empty;
    txtDevletNo.Text = string.Empty;
    txtKulakNo.Text = string.Empty;
    cmbAnneKayitNo.EditValue = null;
    cmbBabaKayitNo.EditValue = null;
    cmbCinsiyeti.SelectedIndex = -1;
    cmbHayvanCinsi.EditValue = null;
    cmbHayvanKayitTuru.EditValue = null;
    deKayitTarihi.EditValue = DateTime.Now;
}

private void BilgileriGoster()
{
    txtBoyunNo.Text = hayvan.BOYUN_NO;
    txtDevletNo.Text = hayvan.DEVLET_NO;
    txtKulakNo.Text = hayvan.KULAK_NO;
    deDogumTarihi.EditValue = hayvan.DOGUM_TARIHI;
    deKayitTarihi.EditValue = hayvan.KAYIT_TARIHI;
    cmbCinsiyeti.SelectedIndex = hayvan.CINSIYETI.Value;
    if (hayvan.ANNE_KAYITNO.HasValue)
    {
        cmbAnneKayitNo.EditValue = CYS.model.HAYVANs.Where(x => x.KAYITNO
== hayvan.ANNE_KAYITNO.Value).FirstOrDefault();
    }
}

```



```

    }

    if (hayvan.BABA_KAYITNO.HasValue)
    {
        cmbBabaKayitNo.EditValue = CYS.model.HAYVANs.Where(x => x.KAYITNO
== hayvan.BABA_KAYITNO.Value).FirstOrDefault();
    }
    cmbHayvanCinsi.EditValue = hayvan.HAYVAN_CINSI;
    cmbHayvanKayitTuru.EditValue = hayvan.HAYVAN_KAYIT_TURU;
}

private void cmbAnneKayitNo_Properties_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    if (e.Button.Kind==DevExpress.XtraEditors.Controls.ButtonPredefines.Clear)
    {
        cmbAnneKayitNo.EditValue = null;
    }
}

private void cmbBabaKayitNo_Properties_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    if (e.Button.Kind == DevExpress.XtraEditors.Controls.ButtonPredefines.Clear)
    {
        cmbBabaKayitNo.EditValue = null;
    }
}

private void cmbHayvanCinsi_Properties_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    if (e.Button.Kind == DevExpress.XtraEditors.Controls.ButtonPredefines.Clear)
    {
        cmbHayvanCinsi.EditValue = null;
    }
    else if (e.Button.Kind==DevExpress.XtraEditors.Controls.ButtonPredefines.Plus)
    {
        frmHayvanCins frm = new frmHayvanCins(this);
        frm.ShowDialog();
    }
}

private void cmbHayvanKayitTuru_Properties_ButtonClick(object sender,
DevExpress.XtraEditors.Controls.ButtonPressedEventArgs e)
{
    if (e.Button.Kind == DevExpress.XtraEditors.Controls.ButtonPredefines.Clear)
    {
        cmbHayvanKayitTuru.EditValue = null;
    }
}

```

```

    }
    else if (e.Button.Kind == DevExpress.XtraEditors.Controls.ButtonPredefines.Plus)
    {
        frmHayvanKayitTur frm = new frmHayvanKayitTur(this);
        frm.ShowDialog();
    }
}

private void barButtonItem1_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    FormuTemizle();
}

private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    Close();
}

private void barButtonItem2_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    try
    {
        if (!dxValidationProvider1.Validate()) return;
        if (hayvan==null)
        {
            hayvan = new HAYVAN();
            HayvanBilgisiniDoldur();
            CYS.model.HAYVANs.Add(hayvan);
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Hayvan kartı açıldı", "Bilgi", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
        else
        {
            HayvanBilgisiniDoldur();
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Hayvan kartı güncellendi", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        Guncelle();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

```

private void Guncelle()
{
    if (caller == null) return;

    frmHayvanListesi frm = caller as frmHayvanListesi;
    if (frm != null) frm.HayvanlariGetir();
}
private void HayvanBilgisiniDoldur()
{
    hayvan.KAYIT_TARIHI = (DateTime)deKayitTarihi.EditValue;
    hayvan.DOGUM_TARIHI = (DateTime)deDogumTarihi.EditValue;
    hayvan.DEVLET_NO = txtDevletNo.Text;
    hayvan.BOYUN_NO = txtBoyunNo.Text;
    hayvan.KULAK_NO = txtKulakNo.Text;
    hayvan.CINSIYETI = cmbCinsiyeti.SelectedIndex;
    hayvan.ANNE_KAYITNO = cmbAnneKayitNo.EditValue != null ?
(cmbAnneKayitNo.EditValue as HAYVAN).KAYITNO : (int?)null;
    hayvan.BABA_KAYITNO = cmbBabaKayitNo.EditValue != null ?
(cmbBabaKayitNo.EditValue as HAYVAN).KAYITNO : (int?)null;
    hayvan.HAYVANCINSI_SIRANO = (cmbHayvanCinsi.EditValue as
HAYVAN_CINSI).SIRANO;
    hayvan.HAYVANKAYITTURU_SIRANO = (cmbHayvanKayitTuru.EditValue as
HAYVAN_KAYIT_TURU).SIRANO;
}
}
}

```

### **frmHayvanListesi.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.Genel;
using CYSLAU.DB;

namespace CYSLAU.Formlar.Hayvan
{
    public partial class frmHayvanListesi : DevExpress.XtraEditors.XtraForm
    {
        public frmHayvanListesi()
        {
            InitializeComponent();

```

```

        HayvanlariGetir();
    }

    public void HayvanlariGetir()
    {
        gridControl1.DataSource = null;
        gridControl1.DataSource = CYS.model.HAYVANs.Select(x => x).ToList();
    }

    private void barButtonItem1_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        frmHayvan frm = new frmHayvan(null, this);
        frm.ShowDialog();
    }

    private void barButtonItem2_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        frmHayvan frm = new frmHayvan(gridView1.GetFocusedRow() as DB.HAYVAN,
this);
        frm.ShowDialog();
    }

    private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        try
        {
            if (XtraMessageBox.Show("Hayvan kartını silmek istediğinizden emin
misiniz?", "Onay", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Ye
s)
            {
                HAYVAN hayvan = gridView1.GetFocusedRow() as HAYVAN;
                CYS.model.HAYVANs.Remove(hayvan);
                XtraMessageBox.Show("Hayvan kartı silindi", "Bilgi", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                HayvanlariGetir();
            }
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    private void barButtonItem8_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        Close();
    }

```

```

    }

    private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        HAYVAN hayvan = gridView1.GetFocusedRow() as HAYVAN;
        if (hayvan == null) return;

        if (hayvan.CINSIYETI.Value==0)
        {
            XtraMessageBox.Show("Sağım işlemi için Dişi hayvan seçilmelidir", "Uyarı",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        frmSutSagim frm = new frmSutSagim(hayvan);
        frm.ShowDialog();
    }

    private void gridView1_CustomDrawCell(object sender,
DevExpress.XtraGrid.Views.Base.RowCellCustomDrawEventArgs e)
    {
        if (e.Column.FieldName=="CINSIYETI")
        {
            if (e.CellValue != null)
            {
                if (e.CellValue.ToString() == "0") e.DisplayText = "ERKEK";
                else e.DisplayText = "DİŞİ";
            }
        }
    }
}
}

```

### **frmSutSagim.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.Genel;
using CYSLAU.DB;

namespace CYSLAU.Formlar.Hayvan
{

```

```

public partial class frmSutSagim : DevExpress.XtraEditors.XtraForm
{
    DB.HAYVAN hayvan;
    SUT sut;
    public frmSutSagim(DB.HAYVAN hayvan=null)
    {
        InitializeComponent();
        this.hayvan = hayvan;
        txtHayvan.Text = "Devlet No:" + hayvan.DEVLET_NO + ", Boyun No:" +
hayvan.BOYUN_NO;
        SutSagimBilgileriniGetir();
        FormuTemizle();
    }

    private void FormuTemizle()
    {
        deSagimTarihi.EditValue = DateTime.Now;
        seSutMiktar.Value = 0;
        sut = null;
    }

    private void SutSagimBilgileriniGetir()
    {
        gridControl1.DataSource = null;
        gridControl1.DataSource = CYS.model.SUTs.Where(x => x.HAYVAN_KAYITNO
== hayvan.KAYITNO).ToList();
    }
    private void barButtonItem4_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        FormuTemizle();
    }

    private void barButtonItem1_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        if (!dxValidationProvider1.Validate()) return;
        try
        {
            if (sut==null)
            {
                sut = new SUT();
                sut.SAGIM_TARIHI = (DateTime)deSagimTarihi.EditValue;
                sut.ALINAN_SUT = seSutMiktar.Value;
                sut.HAYVAN_KAYITNO = hayvan.KAYITNO;
                CYS.model.SUTs.Add(sut);
                CYS.model.SaveChanges();
                XtraMessageBox.Show("Sağım kaydı yapıldı", "Bilgi", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
        }
    }
}

```

```

        else
        {
            sut.SAGIM_TARIHI = (DateTime)deSagimTarihi.EditValue;
            sut.ALINAN_SUT = seSutMiktar.Value;
            CYS.model.SaveChanges();
            XtraMessageBox.Show("Sağım kaydı güncellendi", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        }
        SutSagimBilgileriniGetir();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

private void barButtonItem2_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    sut = gridView1.GetFocusedRow() as SUT;
    if (sut == null) return;
    if (XtraMessageBox.Show("Sağım kaydını silmek istediğinizden emin
misiniz?", "Onay", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        CYS.model.SUTs.Remove(sut);
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Sağım kaydı silindi", "Bilgi", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        SutSagimBilgileriniGetir();
    }
}

private void barButtonItem3_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
{
    Close();
}
}
}

```

### **frmMuayene.cs**

```

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

```

```

using System.Drawing;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using DevExpress.XtraEditors;
using CYSLAU.DB;
using CYSLAU.Genel;

namespace CYSLAU.Formlar.Veteriner
{
    public partial class frmMuayene : DevExpress.XtraEditors.XtraForm
    {
        VETERINER veteriner;
        HAYVAN hayvan;
        SAGLIK muayene;
        object caller;

        public frmMuayene(VETERINER veterine=null,HAYVAN hayvan=null,SAGLIK
muayene=null,object caller=null)
        {
            InitializeComponent();
            this.veteriner = veterine;
            this.hayvan = hayvan;
            this.muayene = muayene;
            this.caller = caller;
            HayvanlariGetir();
            VeterinerleriGetir();
            if (muayene==null)
            {
                btnAsi.Enabled = false;
                btnIlac.Enabled = false;
                btnPrint.Enabled = false;
            }
            else
            {
                btnAsi.Enabled = true;
                btnIlac.Enabled = true;
                btnPrint.Enabled = true;

                cmbHayvan.EditValue = muayene.HAYVAN;
                cmbVeteriner.EditValue = muayene.VETERINER;
            }
        }

        private void HayvanlariGetir()
        {
            cmbHayvan.Properties.Items.Clear();

```



```

        cmbHayvan.Properties.Items.AddRange(CYS.model.HAYVANs.Select(x =>
x).ToList());

        if (hayvan!=null)
        {
            cmbHayvan.EditValue = CYS.model.HAYVANs.Where(x => x.KAYITNO ==
hayvan.KAYITNO).Select(x => x).FirstOrDefault();
        }
    }

    private void VeterinerleriGetir()
    {
        cmbVeteriner.Properties.Items.Clear();
        cmbVeteriner.Properties.Items.AddRange(CYS.model.VETERINERs.Select(x =>
x).ToList());
        if (veteriner != null)
        {
            cmbVeteriner.EditValue = CYS.model.VETERINERs.Where(x => x.SIRANO ==
veteriner.SIRANO).Select(x => x).FirstOrDefault();
        }
    }

    private void barButtonItem1_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        if (!dxValidationProvider1.Validate()) return;

        try
        {
            if (muayene == null)
            {
                muayene = new SAGLIK();
                muayene.VETERINER = cmbVeteriner.EditValue as VETERINER;
                muayene.HAYVAN = cmbHayvan.EditValue as HAYVAN;
                muayene.ISLEM_TARIHI = (DateTime)deMuayeneTarihi.DateTime;
                muayene.ISLEM_ACIKLAMA = txtAciklama.Text;
                muayene.NOTLAR = txtNotlar.Text;

                CYS.model.SAGLIKs.Add(muayene);
                CYS.model.SaveChanges();
                XtraMessageBox.Show("Muayene kaydı yapıldı", "Bilgi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                btnAsi.Enabled = true;
                btnIlac.Enabled = true;
                btnPrint.Enabled = true;
            }
            else
            {
                muayene.ISLEM_TARIHI = (DateTime)deMuayeneTarihi.DateTime;
                muayene.ISLEM_ACIKLAMA = txtAciklama.Text;
            }
        }
    }

```

```

        muayene.NOTLAR = txtNotlar.Text;
        CYS.model.SaveChanges();
        XtraMessageBox.Show("Muayene kaydı güncellendi", "Bilgi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
catch (Exception ex)
{
    throw ex;
}
}
}
}

```

## Veritabanı Sorguları

```

/***** Object: Table [dbo].[HAYVAN_KAYIT_TURU]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[HAYVAN_KAYIT_TURU] (
    [SIRANO] [int] IDENTITY(1,1) NOT NULL,
    [KAYIT_TURU_ADU] [nvarchar](40) NOT NULL,
    CONSTRAINT [PK_HAYVAN_KAYIT_TURU] PRIMARY KEY CLUSTERED
(
    [SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[ILAC_TURU]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ILAC_TURU] (
    [SIRANO] [int] IDENTITY(1,1) NOT NULL,
    [ILAC_ADI] [nvarchar](40) NOT NULL,
    [KULLANIM_TURU] [nvarchar](40) NULL,
    CONSTRAINT [PK_ILAC_TURU] PRIMARY KEY CLUSTERED
(
    [SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[YEM_TURU]      Script Date: 08/20/2020 01:56:07
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[YEM_TURU] (

```

```

        [SIRANO] [int] IDENTITY(1,1) NOT NULL,
        [YEM_TURU_ACIKLAMA] [nvarchar](40) NOT NULL,
        CONSTRAINT [PK_YEM_TURU] PRIMARY KEY CLUSTERED
    (
        [SIRANO] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[YEM_KARISIM]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[YEM_KARISIM] (
    [KARISIM_SIRANO] [int] IDENTITY(1,1) NOT NULL,
    [ACIKLAMA] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_YEM_KARISIM] PRIMARY KEY CLUSTERED
    (
        [KARISIM_SIRANO] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[VETERINER]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[VETERINER] (
    [SIRANO] [int] IDENTITY(1,1) NOT NULL,
    [VETERINER_ADI] [nvarchar](100) NOT NULL,
    [ADRES] [nvarchar](100) NULL,
    [TELEFON1] [nvarchar](20) NULL,
    [TELEFON2] [nvarchar](20) NULL,
    [EPOSTA] [nvarchar](200) NULL,
    [NOTLAR] [nvarchar](255) NULL,
    CONSTRAINT [PK_VETERINER] PRIMARY KEY CLUSTERED
    (
        [SIRANO] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[DOGUM_TURU]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DOGUM_TURU] (
    [SIRANO] [int] IDENTITY(1,1) NOT NULL,
    [DOGUMTURU_ACIKLAMA] [nvarchar](40) NOT NULL,
    CONSTRAINT [PK_DOGUM_TURU] PRIMARY KEY CLUSTERED
    (
        [SIRANO] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

```

```

GO
/***** Object:  Table [dbo].[ASI_TURU]      Script Date: 08/20/2020 01:56:07
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ASI_TURU] (
    [SIRANO] [int] IDENTITY(1,1) NOT NULL,
    [ASI_ADI] [nvarchar](40) NOT NULL,
    CONSTRAINT [PK_ASI_TURU] PRIMARY KEY CLUSTERED
(
    [SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object:  Table [dbo].[HAYVAN_CINSI]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[HAYVAN_CINSI] (
    [SIRANO] [int] IDENTITY(1,1) NOT NULL,
    [CINS_ADI] [nvarchar](40) NOT NULL,
    CONSTRAINT [PK_HAYVAN_CINSI] PRIMARY KEY CLUSTERED
(
    [SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object:  Table [dbo].[HAYVAN]      Script Date: 08/20/2020 01:56:07
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[HAYVAN] (
    [KAYITNO] [int] IDENTITY(1,1) NOT NULL,
    [KAYIT_TARIHI] [datetime] NOT NULL,
    [DEVLET_NO] [nvarchar](40) NOT NULL,
    [BOYUN_NO] [nvarchar](40) NOT NULL,
    [KULAK_NO] [varchar](40) NOT NULL,
    [CINSIYETI] [int] NULL,
    [ANNE_KAYITNO] [int] NULL,
    [BABA_KAYITNO] [int] NULL,
    [HAYVANCINSI_SIRANO] [int] NOT NULL,
    [HAYVANKAYITTURU_SIRANO] [int] NULL,
    [DOGUM_TARIHI] [smalldatetime] NULL,
    CONSTRAINT [PK_HAYVAN] PRIMARY KEY CLUSTERED
(
    [KAYITNO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF

```

```

GO
/***** Object: Table [dbo].[YEM_KARISIM_YEM_TURU]      Script Date:
08/20/2020 01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[YEM_KARISIM_YEM_TURU] (
    [KARISIM_SIRANO] [int] NOT NULL,
    [YEMTURU_SIRANO] [int] NOT NULL,
    [ORAN] [decimal] (10, 2) NOT NULL,
    CONSTRAINT [PK_YEM_KARISIM_YEM_TURU] PRIMARY KEY CLUSTERED
(
    [KARISIM_SIRANO] ASC,
    [YEMTURU_SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[SAGLIK]      Script Date: 08/20/2020 01:56:07
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SAGLIK] (
    [SIRANO] [int] NOT NULL,
    [HAYVAN_KAYITNO] [int] NOT NULL,
    [VETERINER_SIRANO] [int] NOT NULL,
    [ISLEM_TARIHI] [datetime] NOT NULL,
    [ISLEM_ACIKLAMA] [nvarchar] (255) NULL,
    [NOTLAR] [nvarchar] (4000) NULL,
    CONSTRAINT [PK_SAGLIK] PRIMARY KEY CLUSTERED
(
    [SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[HAYVAN_YEM_KARISIM_YEM_TURU]      Script Date:
08/20/2020 01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[HAYVAN_YEM_KARISIM_YEM_TURU] (
    [KAYITNO] [int] NOT NULL,
    [HAYVAN_KAYITNO] [int] NOT NULL,
    [KARISIM_SIRANO] [int] NOT NULL,
    [YEM_SAATI] [datetime] NOT NULL,
    [ACIKLAMA] [nvarchar] (100) NULL,
    CONSTRAINT [PK_HAYVAN_YEM_KARISIM_YEM_TURU] PRIMARY KEY CLUSTERED
(
    [KAYITNO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[HAYVAN_DOGUM_TURU]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[HAYVAN_DOGUM_TURU] (
    [HAYVAN_KAYITNO] [int] NOT NULL,
    [DOGUMTURU_SIRANO] [int] NOT NULL,
    [ACIKLAMA] [nvarchar](40) NULL,
    [VETERINER_SIRANO] [int] NULL,
    CONSTRAINT [PK_HAYVAN_DOGUM_TURU] PRIMARY KEY CLUSTERED
(
    [HAYVAN_KAYITNO] ASC,
    [DOGUMTURU_SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[SUT]      Script Date: 08/20/2020 01:56:07
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SUT] (
    [SIRANO] [int] IDENTITY(1,1) NOT NULL,
    [HAYVAN_KAYITNO] [int] NULL,
    [SAGIM_TARIHI] [datetime] NOT NULL,
    [ALINAN_SUT] [decimal](10, 2) NULL,
    CONSTRAINT [PK_SUT] PRIMARY KEY CLUSTERED
(
    [SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[SAGLIK_ILAC]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SAGLIK_ILAC] (
    [SAGLIK_IRANO] [int] NOT NULL,
    [ILACTURU_SIRANO] [int] NOT NULL,
    [ACIKLAMA] [nvarchar](50) NULL,
    CONSTRAINT [PK_SAGLIK_ILAC] PRIMARY KEY CLUSTERED
(
    [SAGLIK_IRANO] ASC,
    [ILACTURU_SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[SAGLIK_ASI_TURU]      Script Date: 08/20/2020
01:56:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SAGLIK_ASI_TURU] (
    [SAGLIK_SIRANO] [int] NOT NULL,
    [ASITURU_SIRANO] [int] NOT NULL,

```

```

        [ACIKLAMA] [nvarchar](50) NULL,
    CONSTRAINT [PK_SAGLIK_ASI_TURU] PRIMARY KEY CLUSTERED
(
    [SAGLIK_SIRANO] ASC,
    [ASITURU_SIRANO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Default [DEF_SUT_ALINAN_SUT]      Script Date: 08/20/2020
01:56:07 *****/
ALTER TABLE [dbo].[SUT] ADD CONSTRAINT [DEF_SUT_ALINAN_SUT] DEFAULT ((0))
FOR [ALINAN_SUT]
GO
/***** Object: Default [DEF_YEM_KARISIM_YEM_TURU_ORAN]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[YEM_KARISIM_YEM_TURU] ADD CONSTRAINT
[DEF_YEM_KARISIM_YEM_TURU_ORAN] DEFAULT ((0)) FOR [ORAN]
GO
/***** Object: ForeignKey [HAYVAN_CINSI_HAYVAN]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[HAYVAN] WITH CHECK ADD CONSTRAINT
[HAYVAN_CINSI_HAYVAN] FOREIGN KEY ([HAYVANCINSI_SIRANO])
REFERENCES [dbo].[HAYVAN_CINSI] ([SIRANO])
GO
ALTER TABLE [dbo].[HAYVAN] CHECK CONSTRAINT [HAYVAN_CINSI_HAYVAN]
GO
/***** Object: ForeignKey [HAYVAN_KAYIT_TURU_HAYVAN]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[HAYVAN] WITH CHECK ADD CONSTRAINT
[HAYVAN_KAYIT_TURU_HAYVAN] FOREIGN KEY ([HAYVANKAYITTURU_SIRANO])
REFERENCES [dbo].[HAYVAN_KAYIT_TURU] ([SIRANO])
GO
ALTER TABLE [dbo].[HAYVAN] CHECK CONSTRAINT [HAYVAN_KAYIT_TURU_HAYVAN]
GO
/***** Object: ForeignKey [DOGUM_TURU_HAYVAN_DOGUM_TURU]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[HAYVAN_DOGUM_TURU] WITH CHECK ADD CONSTRAINT
[DOGUM_TURU_HAYVAN_DOGUM_TURU] FOREIGN KEY ([DOGUMTURU_SIRANO])
REFERENCES [dbo].[DOGUM_TURU] ([SIRANO])
GO
ALTER TABLE [dbo].[HAYVAN_DOGUM_TURU] CHECK CONSTRAINT
[DOGUM_TURU_HAYVAN_DOGUM_TURU]
GO
/***** Object: ForeignKey [HAYVAN_HAYVAN_DOGUM_TURU]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[HAYVAN_DOGUM_TURU] WITH CHECK ADD CONSTRAINT
[HAYVAN_HAYVAN_DOGUM_TURU] FOREIGN KEY ([HAYVAN_KAYITNO])
REFERENCES [dbo].[HAYVAN] ([KAYITNO])
GO
ALTER TABLE [dbo].[HAYVAN_DOGUM_TURU] CHECK CONSTRAINT
[HAYVAN_HAYVAN_DOGUM_TURU]
GO
/***** Object: ForeignKey [VETERINER_HAYVAN_DOGUM_TURU]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[HAYVAN_DOGUM_TURU] WITH CHECK ADD CONSTRAINT
[VETERINER_HAYVAN_DOGUM_TURU] FOREIGN KEY ([VETERINER_SIRANO])
REFERENCES [dbo].[VETERINER] ([SIRANO])
GO
ALTER TABLE [dbo].[HAYVAN_DOGUM_TURU] CHECK CONSTRAINT
[VETERINER_HAYVAN_DOGUM_TURU]

```

```

GO
/***** Object:  ForeignKey [FK_HAYVAN_YEM_KARISIM_YEM_TURU_YEM_KARISIM]
Script Date: 08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[HAYVAN_YEM_KARISIM_YEM_TURU] WITH CHECK ADD CONSTRAINT
[FK_HAYVAN_YEM_KARISIM_YEM_TURU_YEM_KARISIM] FOREIGN KEY([KARISIM_SIRANO])
REFERENCES [dbo].[YEM_KARISIM] ([KARISIM_SIRANO])
GO
ALTER TABLE [dbo].[HAYVAN_YEM_KARISIM_YEM_TURU] CHECK CONSTRAINT
[FK_HAYVAN_YEM_KARISIM_YEM_TURU_YEM_KARISIM]
GO
/***** Object:  ForeignKey [HAYVAN_HAYVAN_YEM_KARISIM_YEM_TURU]      Script
Date: 08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[HAYVAN_YEM_KARISIM_YEM_TURU] WITH CHECK ADD CONSTRAINT
[HAYVAN_HAYVAN_YEM_KARISIM_YEM_TURU] FOREIGN KEY([HAYVAN_KAYITNO])
REFERENCES [dbo].[HAYVAN] ([KAYITNO])
GO
ALTER TABLE [dbo].[HAYVAN_YEM_KARISIM_YEM_TURU] CHECK CONSTRAINT
[HAYVAN_HAYVAN_YEM_KARISIM_YEM_TURU]
GO
/***** Object:  ForeignKey [HAYVAN_SAGLIK]      Script Date: 08/20/2020
01:56:07 *****/
ALTER TABLE [dbo].[SAGLIK] WITH CHECK ADD CONSTRAINT [HAYVAN_SAGLIK]
FOREIGN KEY([HAYVAN_KAYITNO])
REFERENCES [dbo].[HAYVAN] ([KAYITNO])
GO
ALTER TABLE [dbo].[SAGLIK] CHECK CONSTRAINT [HAYVAN_SAGLIK]
GO
/***** Object:  ForeignKey [VETERINER_SAGLIK]      Script Date: 08/20/2020
01:56:07 *****/
ALTER TABLE [dbo].[SAGLIK] WITH CHECK ADD CONSTRAINT [VETERINER_SAGLIK]
FOREIGN KEY([VETERINER_SIRANO])
REFERENCES [dbo].[VETERINER] ([SIRANO])
GO
ALTER TABLE [dbo].[SAGLIK] CHECK CONSTRAINT [VETERINER_SAGLIK]
GO
/***** Object:  ForeignKey [ASI_TURU_SAGLIK_ASI_TURU]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[SAGLIK_ASI_TURU] WITH CHECK ADD CONSTRAINT
[ASI_TURU_SAGLIK_ASI_TURU] FOREIGN KEY([ASITURU_SIRANO])
REFERENCES [dbo].[ASI_TURU] ([SIRANO])
GO
ALTER TABLE [dbo].[SAGLIK_ASI_TURU] CHECK CONSTRAINT
[ASI_TURU_SAGLIK_ASI_TURU]
GO
/***** Object:  ForeignKey [SAGLIK_SAGLIK_ASI_TURU]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[SAGLIK_ASI_TURU] WITH CHECK ADD CONSTRAINT
[SAGLIK_SAGLIK_ASI_TURU] FOREIGN KEY([SAGLIK_SIRANO])
REFERENCES [dbo].[SAGLIK] ([SIRANO])
GO
ALTER TABLE [dbo].[SAGLIK_ASI_TURU] CHECK CONSTRAINT
[SAGLIK_SAGLIK_ASI_TURU]
GO
/***** Object:  ForeignKey [ILAC_TURU_SAGLIK_ILAC]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[SAGLIK_ILAC] WITH CHECK ADD CONSTRAINT
[ILAC_TURU_SAGLIK_ILAC] FOREIGN KEY([ILACTURU_SIRANO])
REFERENCES [dbo].[ILAC_TURU] ([SIRANO])
GO
ALTER TABLE [dbo].[SAGLIK_ILAC] CHECK CONSTRAINT [ILAC_TURU_SAGLIK_ILAC]
GO

```



```

/***** Object: ForeignKey [SAGLIK_SAGLIK_ILAC]      Script Date: 08/20/2020
01:56:07 *****/
ALTER TABLE [dbo].[SAGLIK_ILAC] WITH CHECK ADD CONSTRAINT
[SAGLIK_SAGLIK_ILAC] FOREIGN KEY ([SAGLIK_IRANO])
REFERENCES [dbo].[SAGLIK] ([SIRANO])
GO
ALTER TABLE [dbo].[SAGLIK_ILAC] CHECK CONSTRAINT [SAGLIK_SAGLIK_ILAC]
GO
/***** Object: ForeignKey [HAYVAN_SUT]      Script Date: 08/20/2020
01:56:07 *****/
ALTER TABLE [dbo].[SUT] WITH CHECK ADD CONSTRAINT [HAYVAN_SUT] FOREIGN
KEY ([HAYVAN_KAYITNO])
REFERENCES [dbo].[HAYVAN] ([KAYITNO])
GO
ALTER TABLE [dbo].[SUT] CHECK CONSTRAINT [HAYVAN_SUT]
GO
/***** Object: ForeignKey [YEM_KARISIM_YEM_KARISIM_YEM_TURU]      Script
Date: 08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[YEM_KARISIM_YEM_TURU] WITH CHECK ADD CONSTRAINT
[YEM_KARISIM_YEM_KARISIM_YEM_TURU] FOREIGN KEY ([KARISIM_SIRANO])
REFERENCES [dbo].[YEM_KARISIM] ([KARISIM_SIRANO])
GO
ALTER TABLE [dbo].[YEM_KARISIM_YEM_TURU] CHECK CONSTRAINT
[YEM_KARISIM_YEM_KARISIM_YEM_TURU]
GO
/***** Object: ForeignKey [YEM_TURU_YEM_KARISIM_YEM_TURU]      Script Date:
08/20/2020 01:56:07 *****/
ALTER TABLE [dbo].[YEM_KARISIM_YEM_TURU] WITH CHECK ADD CONSTRAINT
[YEM_TURU_YEM_KARISIM_YEM_TURU] FOREIGN KEY ([YEMTURU_SIRANO])
REFERENCES [dbo].[YEM_TURU] ([SIRANO])
GO
ALTER TABLE [dbo].[YEM_KARISIM_YEM_TURU] CHECK CONSTRAINT
[YEM_TURU_YEM_KARISIM_YEM_TURU]
GO

```