**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

Name Signature

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 uygulamar son zamanlarda yoğun olarak kullanılmaya başlansada önümüzdeki bir kaç yıl daha masaüstü ve web uygulamalarının etkinliği sürecek 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 Prensibi………………………………………………………………. | 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 sağlamakadır.

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 ulaşılabilmektedir. 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 Çiftliklerde 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 farkllı programlama dillerinin de katılarak projenin daha da gelişmesine olanak sağlama düşüncesi en büyük hedefimizdir.

1.3. Sistem Gereksinimleri

1. Fonksiyonel Gereklilikler

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

1. 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.

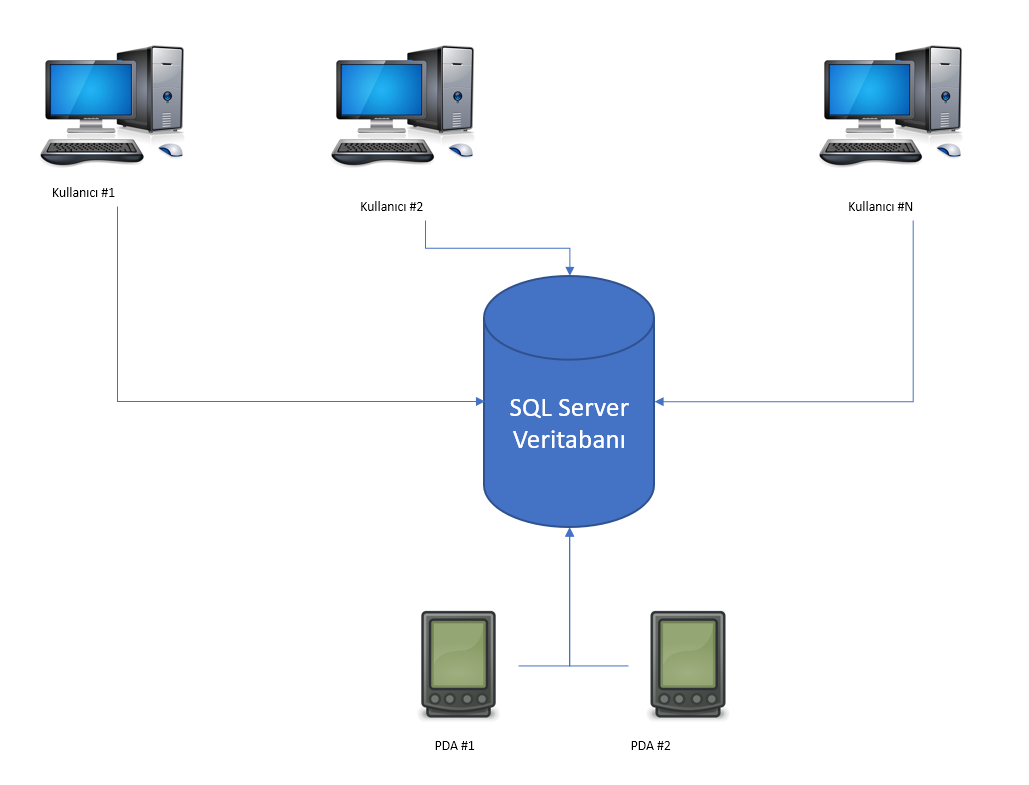
**Kullanılabilirlik:** Kullanıcı dostu, basit anlaşılabilir bir ara yüz.

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

**Taşınabilirlik:** Uygulama bu aşamada sadece Windows İşletim Sistemi olan (Windows 7.0 ve üzeri) cihazlarda kullanılabilir.

**Geliştirilebilirlik:** Altyapısı itibari ile geliştirilmeye 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 geliştilererek son halini alacaktır.

# **3. TASARIM**

3.1.Veritabanı Tasarımı

Datanamic DeZign tool ile tasarımı yapılan veritabanının tablo ve alan açıklamaları bu ara raporda yer almaktadır. Uygulama geliştirme aşamasında ihtiyaca yönelik yeni tablo ve/veya tablo alanları eklenecektir.



Şekil 2 – ÇYS İlişkisel Veritabanı

Şekil 2’de görüldüğü üzere, yapmış olduğumuz çalışma ve analizelerden gerekli tasarım geliştirilmiştir. Tabloların açıklamaları aşağıda detayları olarak yazılmıştır.

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 kayıtta 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 kayıtta 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 kayıtta 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 kayıtta 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 kayıtta 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 havyan 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 veterine tarafından yapılan sağlık muayenelerinn 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 | İlaçtürü 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**

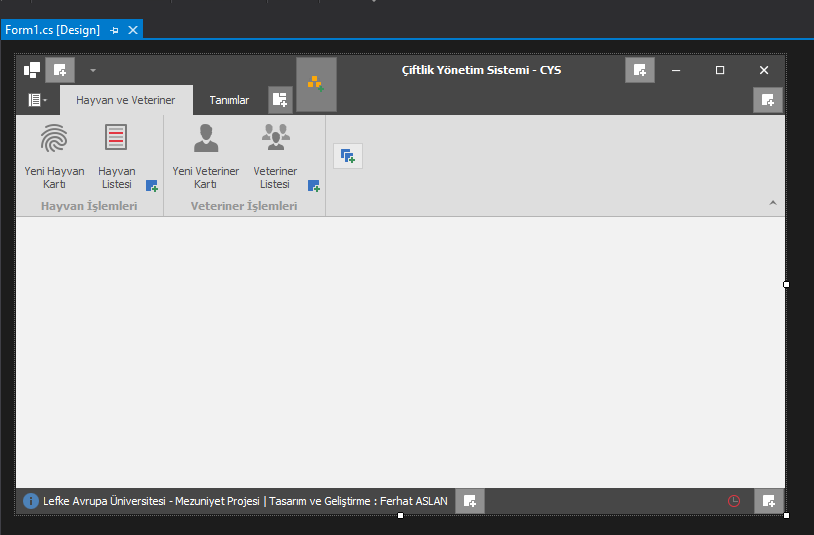
Doğum yapan hayvanların doğum detaylarının tutulduğu tablodur.

|  |  |  |  |
| --- | --- | --- | --- |
| 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.

Ribbon Menu Bar

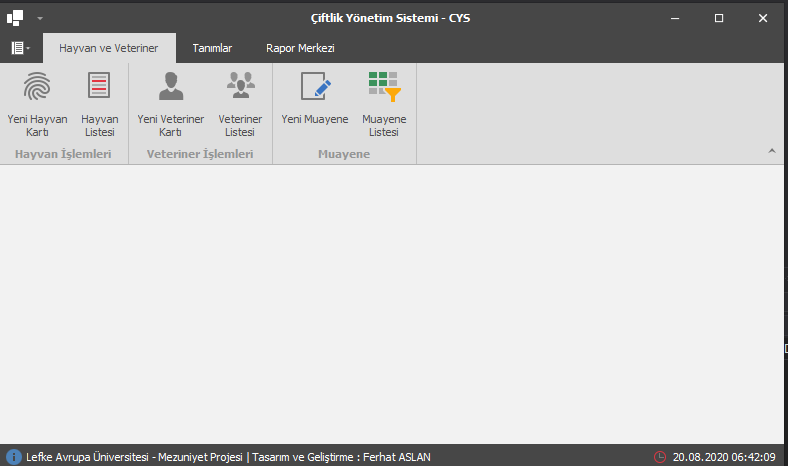


Şekil 3 – Arayüz Tasarım

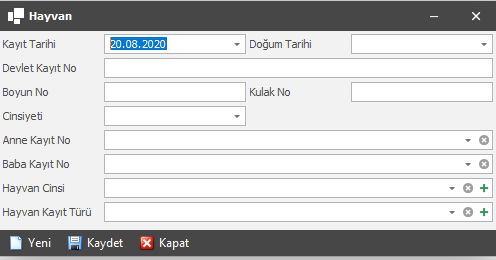
DevExpress Form

Ribbon Status Bar

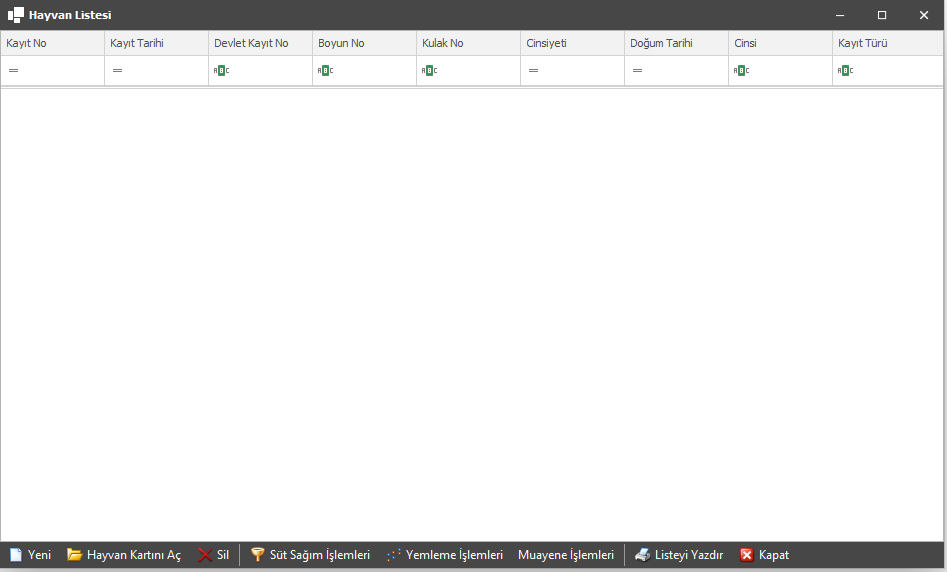
# **4. Kullanım Klavuzu**



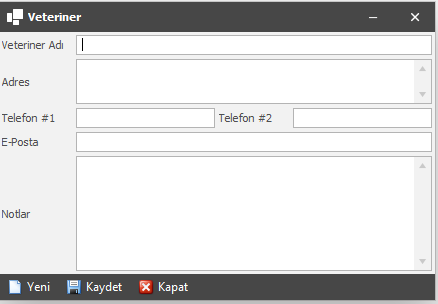
Şekil 4 – ÇYS Ana Ekranı



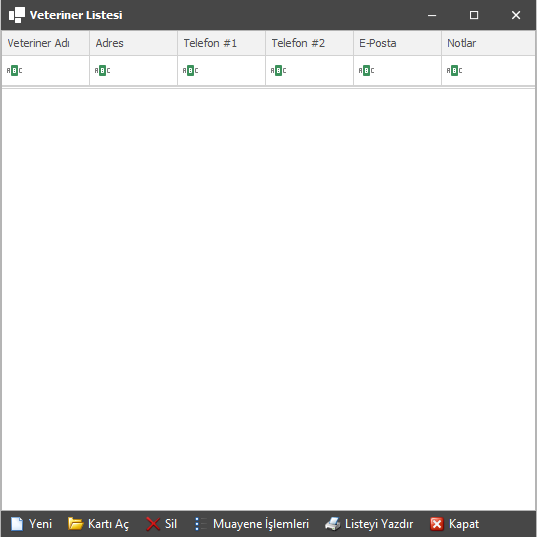
Şekil 5 – Yeni Hayvan Kartı



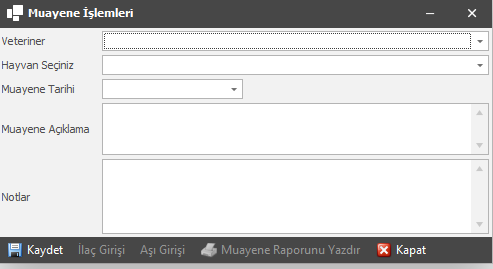
Şekil 6- Kayıtlı Hayvanların Listesi



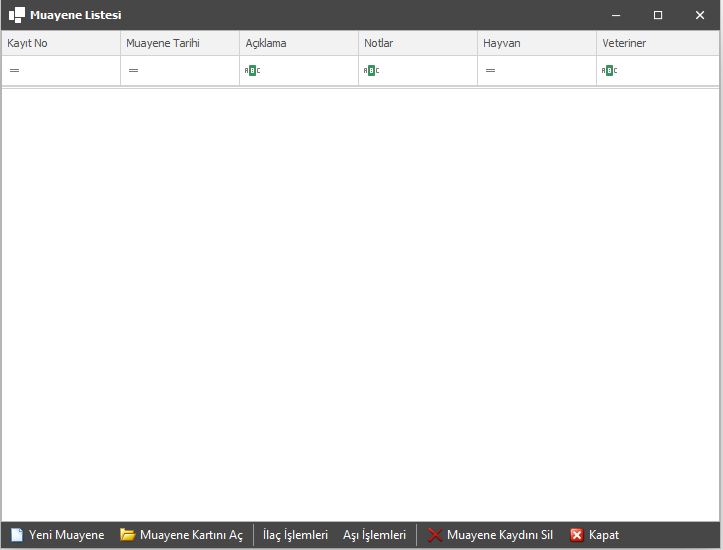
Şekil 7 – Yeni Veteriner Kartı



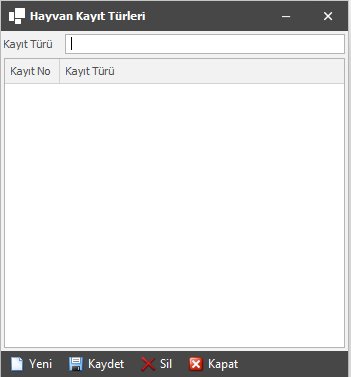
Şekil 8 – Veteriner Listesi



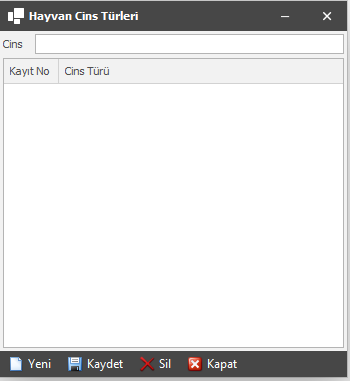
Şekil 9 – Yeni Muayene Kartı



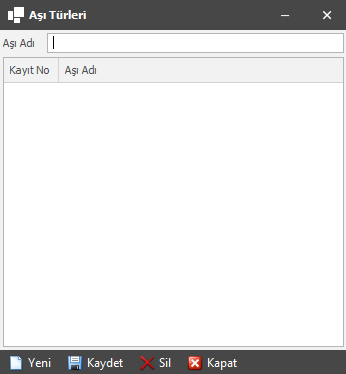
Şekil 10 – Muayene Listesi



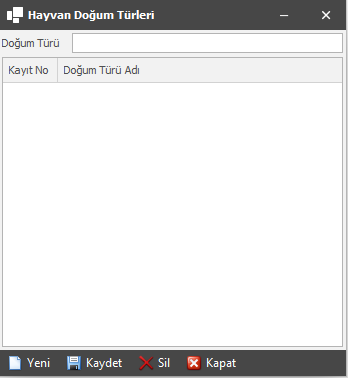
Şekil 11 – Hayvan Kayıt Türleri



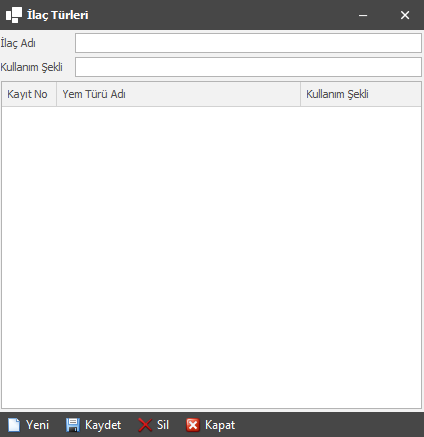
Şekil 12 – Hayvan Cinsleri



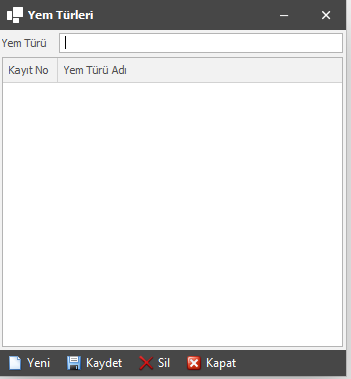
Şekil 13 – Aşı Türleri



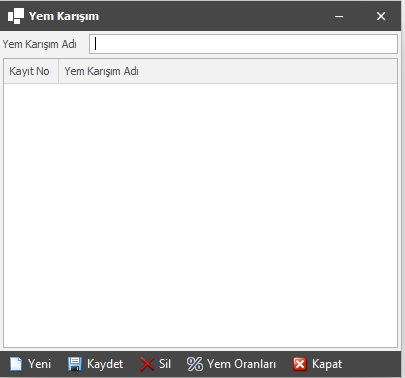
Şekil 14 – Doğum Türleri



Şekil 15 – İlaç Türleri



Şekil 16 – Yem Türleri



Şekil 17 – Yem Karışım Türü

# **5. Sonuç ve Öneriler**

Geliştirmiş olduğuuz 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 sayaede değerlendirelecektir.

# **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 kayitTuru;

public frmDogumTuru()

{

InitializeComponent();

Listele();

}

private void EkraniTemizle()

{

/\* Ekranı 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.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()

{

/\* Ekranı 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();

}

}

}

**frmHayvanKayıtTur.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()

{

/\* Ekranı 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();

}

}

}

**frmIlacTur.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 frmIlacTuru : DevExpress.XtraEditors.XtraForm

{

ILAC\_TURU kayitTuru;

public frmIlacTuru()

{

InitializeComponent();

Listele();

}

private void EkraniTemizle()

{

/\* Ekranı 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 = txtKullanimSekli.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 = txtKullanimSekli.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;

txtKullanimSekli.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 kayitTuru;

public frmYemKarisim()

{

InitializeComponent();

Listele();

}

private void EkraniTemizle()

{

/\* Ekranı 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.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()

{

/\* Ekranı 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 kayitTuru;

public frmYemTuru()

{

InitializeComponent();

Listele();

}

private void EkraniTemizle()

{

/\* Ekranı 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.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(AnneListesi.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.Select(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.Yes)

{

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