**TC**

**KOCAELİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Ders Adı : Programlama Labaratuvarı II**

**Proje : KÜTÜPHANE OTOMASYONU**

**Proje Ortamı: C#**

**Proje Danışmanı: Arş. Gör. Mehmet ALİ ALTUNCU**

**Hazırlayanlar:**

**110201035 Burak UTKUCU**

**110201053 Feyyaz ÖZHAN**

**110201001 Ramazan GÜÇLÜ**

İlk olarak Microsoft Visual Stdio çalışma ortamında form application oluşturduk. C# dilini

seçtik ve programa başladık.

Access veri tabanını kullanacağımızdan dolayı “using System.Data.OleDb” kodunu ekledik.

Veri tabanıyla bağlantı yapabilmek için OleDbConnection oluşturduk.

Şimdi bu programımızda seçtiğimiz acsess veri tabanından bahsedelim.Neden bu veritabanını seçriğimize göz atalım.

Access İlişkisel veritabanı yönetim sistemi ile çalışan bir veritabanı oluşturma programıdır.

Bir veri tabanını oluşturmak ve kullanmak Access ile diğer veri tabanı uygulamalarına göre çok daha kolaydır. Bunun nedeni Access’in Windows ortamının Grafiksel Kullanıcı Arabiriminin sağladığı avantajların tümünden yararlanma imkânı vermesidir. Grafiksel Kullanıcı Arabirimi karmaşık komut dizilerini öğrenmeyi gerektirmeden ekran üzerindeki nesneler ve simgeler yardımıyla fare desteğinden de yararlanarak kullanıcının çalışmasına olanak verir.

Geleneksel veri tabanı uygulamalarında iki tablo arasında bağlantı kurmak için oldukça karmaşık komut dizileri yazmak gerekirken Access’te bu iş basit bir fare hareketiyle gerçekleştirilebilir.

MS Access ile kolaylıkla yapabileceğiniz bazı işlemler;

Birbiriyle bağlantılı yada bağlantısız çok sayıda tablo veya veri tabanı yaratılabilir.

Veri tabanlarını güncel tutabilmek için gereken işlemler istenildiğinde kullanıcı tarafından tasarlanabilen formlar vasıtasıyla yapılabilir.

Değişik yöntemlerle istenilen verilere ulaşılabilir ve çıktılar ekran yada yazıcıya yönlendirilebilir.

Değişik biçim ve boyutlarda rapor ve etiket basımı gerçekleştirilebilir.

Veri tabanında bulunan sayısal veriler üzerinde matematiksel hesaplamalar yaptırılabilir ve sonuçlar istenildiğinde raporlanabilir veya veri tabanında saklanabilir.

Verilerin grafikler yardımıyla analizi yapılabilir ve raporlanabilir.

Kayıtlar üzerinde filtreleme ve sorgulama işlemleri yapılarak istenilen kayıtların görüntülenmesi ve işleme konulması sağlanabilir.

Excel, Word, Paradox, Dbase vb. tabloları Access’te kullanılabilir ya da Access tabloları bu uygulamalarda kullanmak üzere kaydedilebilir.

Yukarıda sıralanan tüm bu özellikler Microsoft Access’in esnek yapısının sonucudur.

Örneğin; oluşturulan ve hatta bilgi girilen bir tablonun yapısı üzerinde tekrar değişiklik yapmak oldukça kolay bir şekilde gerçekleştirilebilir. Oysa ki geleneksel veri tabanı sistemlerinde böyle bir düzenleme oldukça zahmetli ve karmaşık bir çalışma gerektirir.

FONKSİYONLAR

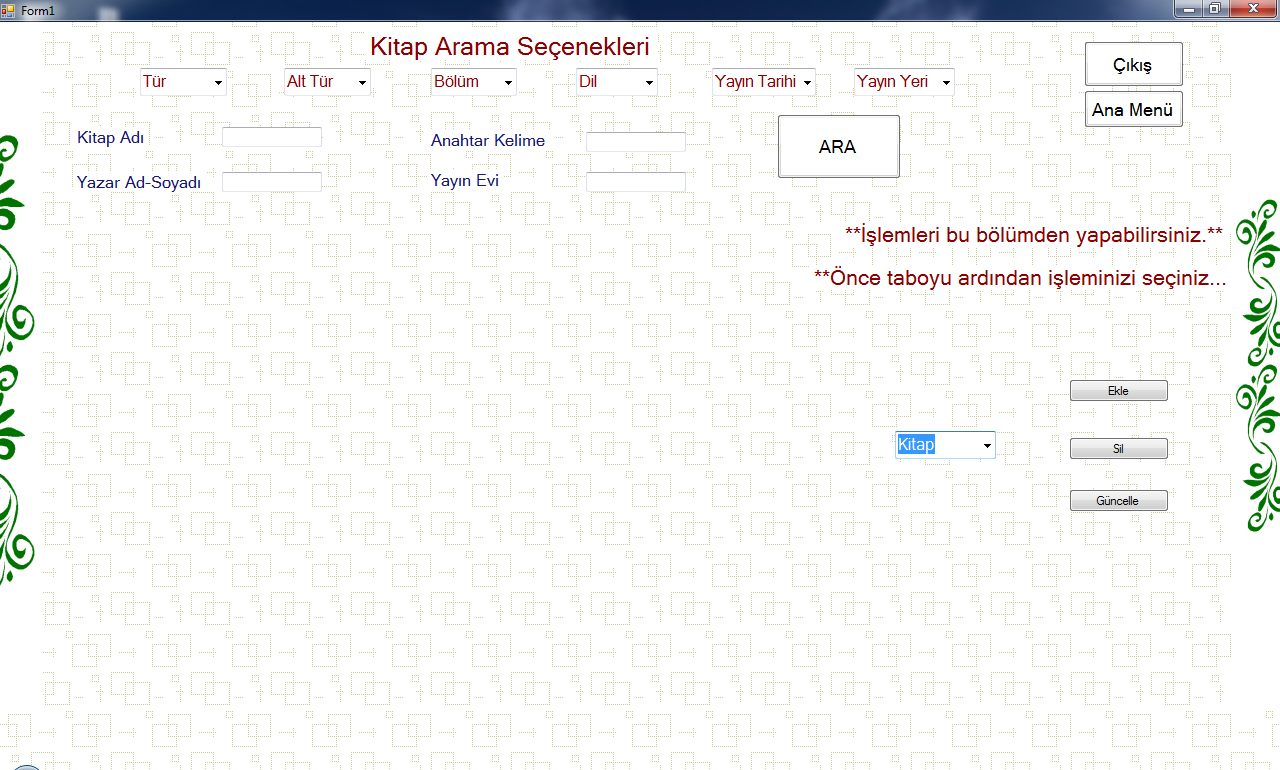
baglan() fonksiyonu:

Veri tabanıyla bağlantı durumunu kontrol eder. Eğer bağlantı yoksa tekrar veri tabanına

bağlanma işlemini gerçekleştirir.

e\_kitap\_goster() fonksiyonu:

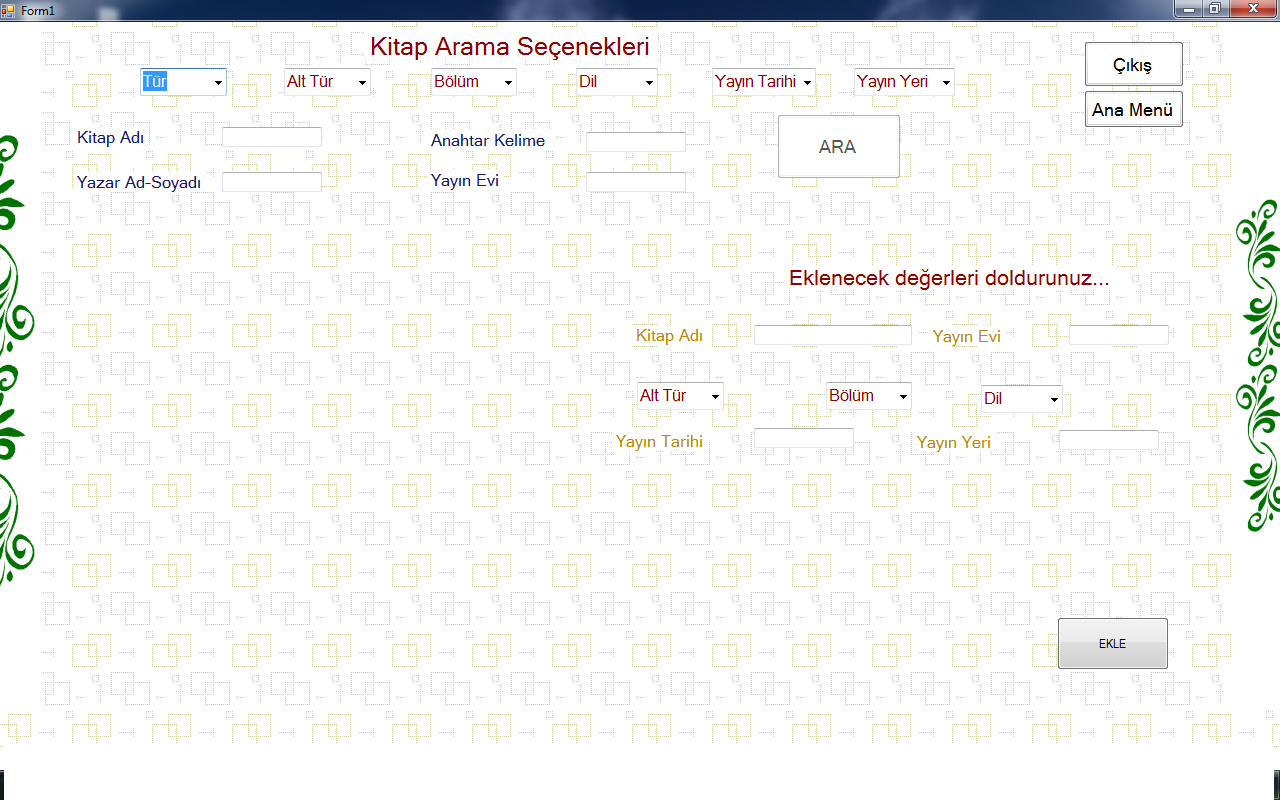
Yönetici giriş yaptıktan sonra karşısına çıkan ekran bu olacaktır.



Ardından ekle butonuna basarsa bu fonksiyon sayesinde butonları, labelleri vs. gösterir.

Örneğin “Kitap Adı” labeli , “Alt Tür” comboboxının başlangıçtaki false olan visible

özelliğini true yaptık.



e\_yazar\_goster(), e\_özet\_goster(), e\_kullanıcı\_goster() fonksiyonları da yöneticinin eklemek

istediği tabloların eklenecek seçeneklerini yukarıdaki fonksiyon gibi ekrana getirir.

e\_kitap\_kaybet(),e\_yazar\_kaybet(),e\_özet\_kaybet(),e\_kullanıcı\_kaybet()

fonksiyonları , yönetici ekleme işlemini yaptıktan sonra 'tamam' butonuna

basarsa, eklemek istediği tablonun seçeneklerini belirleyen labellerin,

butonların, textboxların vs. visible özelliğini kullanarak ekrandan kaybolmasını

sağlar.

arama\_goster() fonksiyonu yönetici veya kullanıcı giriş yaptıysa kitap arama

seçeneklerini ekrana getirir.

private void button4\_Click(object sender, EventArgs e) {}: kullanıcı öğrenci

girişine tıklarsa yapılacak işlemlerin olduğu yerdir. İlk olarak baglan()

fonksiyonuyla veri tabanına bağlantıyı kontrol eder. Ardından dataGridView1

de kullanmak üzere geçici bir datatable oluşturduk. OleDbDataAdapter

kullanarak kullanıcı tablosundan daha önce böyle bir kayıt olup olmadığını

kontrol ettirdik. if (dt.Rows.Count > 0) kodu tablodan kayıtlı öğrenciyi bulma

işlemini yapar. Eğer kayıtlı öğrenci varsa tekrar OleDbDataAdapter oluşturduk

ve kitap tablosundan öğrenciye kayıtlı kitapları gösterdik.

private void button5\_Click(object sender, EventArgs e){}: kullanıcı yeni kayıt

girişine tıklarsa yapılacak işlemlerin olduğu yerdir. Tabloya ekleme işlemi

yapacağımızdan dolayı şu kodu kullandık:

OleDbCommand kom = new OleDbCommand("INSERT INTO Kullanıcı(Öğrenci\_Numara,Şifre) VALUES('" + textBox1.Text + "', '" + textBox2.Text + "')", burak);

Bu kod sayesinde kullanıcı tablosuna sırayla kullanıcının girmiş olduğu öğrenci

numarasını ve şifresini ekledik. Ardından “kom.ExecuteNonQuery();” koduyla

da ekleme işlemini tamamladık.

private void button7\_Click(object sender, EventArgs e) {}: Yönetici veya

öğrenci sisteme giriş yaptığında, istediği kitabı ara seçeneğine basarak bulabilme

imkanını sağlar. Combobox tan istediği kitabın özelliklerini doldurarak ara

butonuna bastığında kitap tablosundan istediği özellikteki kitapları bulur.

Ardından dataGridView elemanını kullanarak ekrana basar.

private void button10\_Click(object sender, EventArgs e) {} : Veri tabanına join işlemiyle

istek adında bir tablo oluşturduk. Bu tablo kullanıcının istediği kitabın adını ve kullanıcının

öğrenci numarasını barındırmaktadır. Öğrenci kitap isteğinde bulunmak için istediği kitabın

adını yazıp “istek gönder” butonuna basması gerekir. Yönetici sisteme giriş yaptığında

OleDbDataAdapter kullanarak istek tablosundaki kullanıcıları ve istediği kitapları gösterir.

\*\*\*Yöneticinin silme işlemini gerçekleştirmesi için de şu kodu kullandık:

**OleDbCommand kom = new OleDbCommand("DELETE FROM Kitap WHERE Kitap\_Adı='" + textBox11.Text + "' AND Alt\_Tür='" + selectItem9.ToString() + "' AND Dil='" + selectItem12.ToString() + "' ", burak);**

Bu kod sayesinde Kitap tablosundan yöneticinin girmiş olduğu özellikteki kitabı bulur.

Ardından da “kom.ExecuteNonQuery()” koduyla da silme işlemini tamamlar.

\*\*\*Yazar, Kullanıcı, Özet tablosundaki ekleme ve silme işlemlerini de yukarıdaki bağlantıların birkaç özelliğini değiştirerek yapmış olduk.

KAYNAKÇA

<http://office.microsoft.com/tr-tr/access>

<http://www.pdfindir.com/access-veri-taban%C4%B1-ornekler-pdf>

access.cahilim.com

ramazanapaydın.com

http://www.youtube.com/watch?v=1TDGSGjoU9c