

**MARMARA ÜNİVERSİTESİ
TEKNİK BİLİMLER MESLEK
YÜKSEKOKULU**

TEST OLUŞTURMA

Bitirme Projesi

**Burak Boz
460115506**

**Bölüm : Bilgisayar Teknolojileri
Program: Bilgisayar Programlama**

Danışman : Fatih Kazdal

2017

**MARMARA ÜNİVERSİTESİ
TEKNİK BİLİMLER MESLEK
YÜKSEKOKULU**

TEST OLUŞTURMA

Bitirme Projesi

**Burak Boz
460115506**

**Bölüm : Bilgisayar Teknolojileri
Program: Bilgisayar Programlama**

**Proje Sınavı :
Doç. Dr. Vedat TOPUZ
Öğr.Görv. Dr. Aysun ALTIKARDEŞ
Öğr.Görv. Fatih KAZDAL**

Özgünlük Bildirisi

1. Bu çalışmada, başka kaynaklardan yapılan tüm alıntıların, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini,
2. Alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın benim tarafımdan yapıldığını bildiririm.

Burak Boz

İÇİNDEKİLER

BÖLÜM I

Özet.....	5
-----------	---

BÖLÜM II

PROJENİN TANITIMI VE PLANI.....	6
2.1 Projenin planlanması ve Gantt şeması.....	6
2.2 Aksaklıklar ve Çözümler.....	6
2.3 Projede Kullanılan Teknolojiler ve Programlar....	6

BÖLÜM III

KURAMSAL BİLGİLER.....	7
3.1 Microsoft Word.....	7
3.2 Microsoft Excel.....	8

BÖLÜM IV

TASARIM ve GERÇEKLEME.....	10
4.1 Veritabanı Tasarımı.....	10
4.2 Form Tasarımı.....	12
4.3 Projenin Kodlanması.....	19

BÖLÜM V

SONUÇ ve ÖNERİLER.....	62
------------------------	----

BÖLÜM VI

KAYNAKÇA.....	63
---------------	----

BÖLÜM VII

ÖZGEÇMİŞ.....	64
---------------	----

TEST OLUŐTURMA

(ÖZET)

Test oluőturma otomasyonunun ana amacı, sınav hazırlanırken ve sonuçları okurken zaman kazanılmasıdır. Birçok sınavda da olduđu gibi konular sabittir ve yıllar içerisinde sorulan sorular belli havuzlarda birikir. Bu otomasyonda amacımız soru havuzunu bir veri tabanında tutmak ve istenildiđi zaman çok çabuk şekilde bu havuzdan soruları çekip sınavın hazır hale gelmesini sađlamaktır. Yönetici kiři soruları derslere ve özel olarak da derslerin içerisinde zorluk derecesine göre guruplandırır.

Proje öncesinde yapılan araőtırmalarda sınav oluőtururken yoğunlukla Microsoft Office programlarının kullanıldıđı ve sınavları okurken de optik okuyucu kullanıldıđı belirlenmiőtir. Optik okuyucudan okutulan sonuçlar bir not defteri dosyasına sistematik bir şekilde kaydedilmektedir. Okutulan bu sonuçlar optik okuyucuya yerleőtirilen öđrenci numarası “00000” olan cevap kađıdı ile karőtılaőtırılır ve sonuçlar okunmuőt olur.

Proje içersinde istenilirse soru sayısı verilip sistemden rastgele sorular seçilebilir ya da sorular özel olarak tek tek seçilebilir. Sınav kađıdı üzerinde olacak olan ders ismi, sınav süresi vb. bilgiler doküman hazırlama formunda sisteme girilir ve onaylandıđında soru kađıdı ve cevap anahtarı kađıdı otomatik olarak oluőturulur.

Böyle deđerli bir proje çalıőtmasında kendisi ile çalıőtmamıza fırsat verdiđi için Öđr. Gör. Fatih KAZDAL hocama, bölüm hocalarımız Doç. Dr. Vedat TOPUZ’a ve tüm Marmara Üniversitesi Tbmyo Bilgisayar Teknolojisi ve Programlama bölümü öđrencilerine desteklerinden dolayı tek tek teőtekkür ederiz.

BÖLÜM II

PROJENİN TANITIMI VE PLANI

2.1 Projenin Planlanması ve Gantt Şeması

1. Mevcut sistem incelenmesi
2. Sistemdeki aksaklıklara çözüm üretilmesi
3. Kullanılacak teknolojilere karar verilmesi
4. Veritabanı tasarımı
5. Form tasarımları
6. Kodlama
7. Danışman onayı
8. Rapor hazırlama

1.Hafta	2.Hafta	3.Hafta	4.Hafta	5.Hafta	6.Hafta	7.Hafta	7.Hafta	8.Hafta	9.Hafta	10.Hafta
1-2										
	3									
		4								
			5							
				6	6	6	6	6		
								7		
									8	

Şekil 2.1: Gantt şeması

2.2 Aksaklıklar ve Çözümler

Mevcut sistem incelemesi tamamlanmış ve aksaklıklara üretilcek çözümler belirlenmiştir. Karşılaşılan aksaklıklardan biri soruların sürekli yeniden yazılması. Üretilen çözüm ise soruları bir veritabanına kaydetmek ve gerektiği zaman buradan tekrar kullanılması. Karşılaşılan ikinci aksaklık sınav sonuçlarının okunması. Üretilen çözüm optik okuyucudan gelen düzenli verinin işlenerek tek seferde okunması ve görüntülenmesi. Karşılaşılan bir diğer aksaklık ise sonuçların görüntülenmesi ve değerlendirilmesi. Bu aksaklığı gidermek için üretilen çözüm ise Microsoft Excel programının kullanılması ve öğrencilerin notlarına göre ayrılması ve grafik ekranında görüntülenmesi.

2.3 Projede Kullanılacak Teknolojiler ve Programlar

- Veritabanı = Microsoft SQL
- Yazılım Dili = C#
- Tasarım = Adobe Photoshop
- Doküman Görütüleme = Microsoft Office, Microsoft Excel

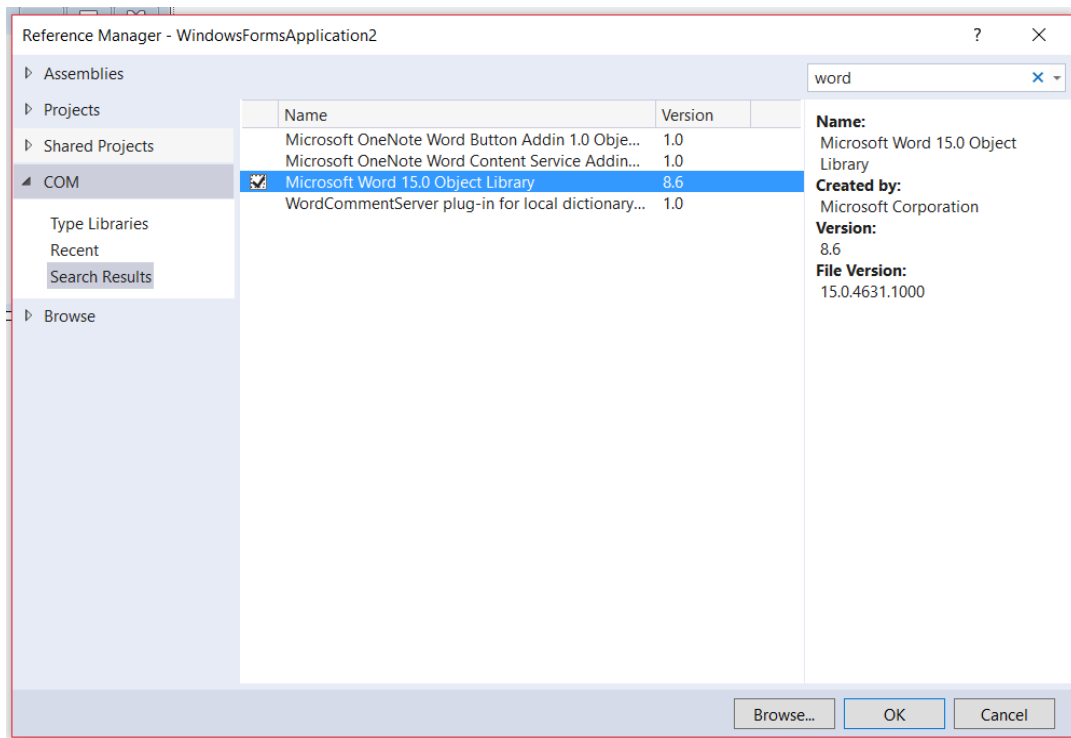
BÖLÜM III

KURAMSAL BİLGİLER

Projeye başlamadan önce yapılan araştırmalar sonucu test hazırlarken ve sonuç listelerken kullanılan programların Microsoft Word ve Microsoft Excel olduğu tespit edilmiştir. Bu programlar detaylı incelenmiş ve hangi alanların kullanılacağına karar verilmiştir. Kullanılan yazılım dili olan C# ile bağlantıları araştırılmıştır ve projeye dahil edilmiştir.

3.1 Mirosoft Word

C# yazılım dili içerisinde Microsoft Word uygulamasının kullanılması için gerekli referans eklenmiştir:



Şekil 3.1: Word referansı ekleme

Kütüphaneler arasına gerekli kod eklenmiştir:

```
using word = Microsoft.Office.Interop.Word;
```

Bu kütüphane sayesinde Microsoft Word programına C# içerisinde aktif şekilde ulaşılabilir. Örneğin;

```
word.Application olustur;  
word.Document icerik;  
olustur = new word.Application();  
olustur.Visible = true;
```

komutları ile yeni bir Word uygulaması oluşturulmuştur ve içerisine yeni bir sayfa eklenmiştir. Daha sonra ise bu uygulamanın görünürlüğü aktif edilmiştir.

Sınav dökümanlarında üst sırada bulunan sınav hakkındaki bilgiler ve ad, soyad girilmesi gereken alanlar Word uygulamasında üst bilgi olarak eklenir ve kullanılır. C# ile Word uygulamasına üst bilgi eklemek için;

```
word.Range headerRangeA =
icerik.Sections[1].Headers[word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;
headerRangeA.Fields.Add(headerRangeA, word.WdFieldType.wdFieldPage);
```

komutları kullanılır.

Soruların ve şıkların düzenli bir formatta listelenmesi için tablo kullanılmalıdır. C# dili ile Word uygulamasına tablo eklemek ve tabloda değişiklik yapmak için;

```
word.Table oTable;
word.Range wrdRng = icerik.Bookmarks.get_Item(ref son).Range;
oTable = icerik.Tables.Add(wrdRng, tabloSatirSayisi, 1, ref omissing, ref omissing);
```

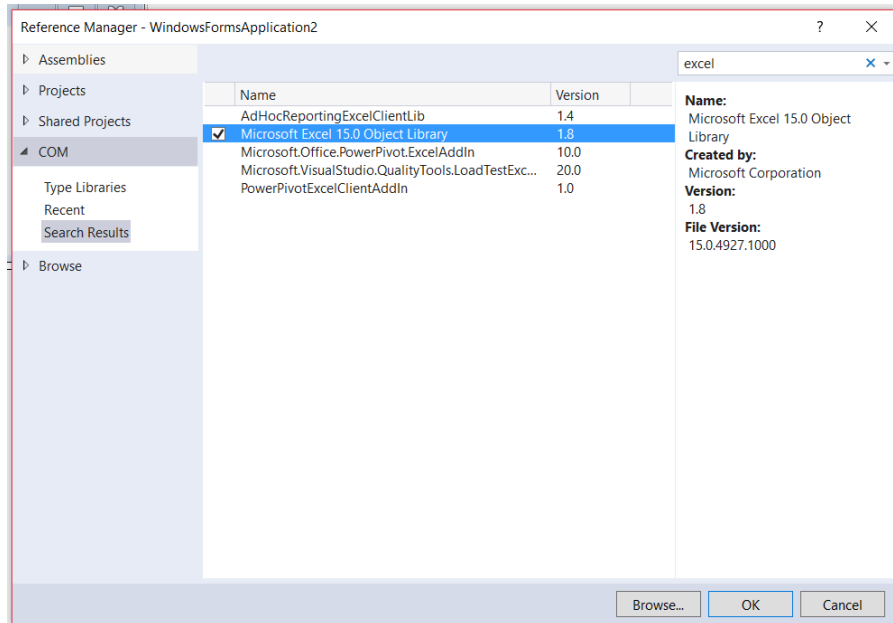
komutları kullanılır. Tablo hücrelerine veri girebilmek için;

```
oTable.Cell(i, 3).Range.Text = "d)";
```

komutu kullanılır.

3.2 Microsoft Excel

C# yazılım dili içerisinde Microsoft Excel uygulamasının kullanılması için gerekli referans eklenmiştir:



Şekil 3.2: Word referansı ekleme

Kütüphaneler arasına gerekli kütüphane eklenmiştir;

```
using Excel = Microsoft.Office.Interop.Excel;
```

Bu kütüphane sayesinde Excel uygulaması C# içersinden kolaylıkla kontrol edilebilmektedir. Yeni bir Excel çalışma sayfası oluşturmak için;

```
Excel.Application excelDosya = new Excel.Application();
object Missing = Type.Missing;
Workbook calismakitabi = excelDosya.Workbooks.Add(Missing);
Worksheet sheet1 = (Worksheet)calismakitabi.Sheets[1];
```

Komutları kullanılır. Bu komutlar sayesinde yeni bir Excel uygulaması oluşturulur ve yeni bir çalışma sayfası açılır.

Hücrelere datatable nesnesinden veri girmek için;

```
Range myrange2 = (Range)sheet2.Cells[analizSatir + i, analizSutun + j];
myrange2.Value2 = dtSoruAnalizA.Rows[i][j] == null ? "" : dtSoruAnalizA.Rows[i][j];
```

komutları kullanılır.

Çalışma sayfasına grafik eklemek için;

```
Excel.Range chartRange;
Excel.ChartObjects xlCharts = (Excel.ChartObjects)sheet2.ChartObjects(Type.Missing);
Excel.ChartObject myChart = (Excel.ChartObject)xlCharts.Add(50, 20, 800, 250);
Excel.Chart chartPage = myChart.Chart;
chartRange = sheet2.get_Range("B:A");
chartPage.SetSourceData(chartRange, Missing);
chartPage.ChartType = Excel.XlChartType.xlColumnClustered;
```

Bu komutlar sütun grafik eklemeye yarar. Komutlar içerisinde bulunan “B:A” seçilen sütunlardır. Grafik bu sütunlara göre oluşturulur.

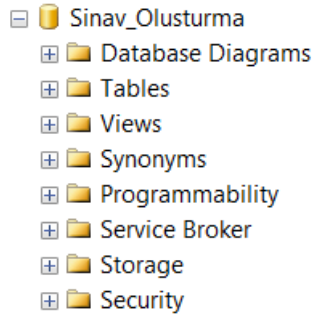
BÖLÜM IV

TASARIM ve GERÇEKLEME

4.1 Veritabanı Tasarımı

Projede Microsoft Sql Sever veritabanı kullanılmıştır. Sql veritabanının seçilme sebebi daha güvenli, daha hızlı ve stabil çalışmasıdır. Projede 3 adet tablo kullanılmıştır.

Öncelikle boş halde Sinav_Olusturma isimli veritabanı oluşturulmuştur.

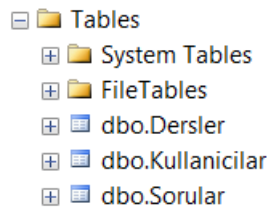


Şekil 4.1.1: Veritabanı oluşturma

Daha sonra içerisine 3 adet tablo eklenmiştir. Bu tablolar sırasıyla

- Dersler
- Kullanıcılar
- Sorular

Tablolarıdır.



Şekil 4.1.2: Tablo ekleme

Tablolara gerekli alanlar düşünülmüş ve eklenmiştir. Tablolara gerekli alanlar eklenmiştir ve birincil anahtarlar belirlenmiştir.

Dersler tablosu:

Column Name	Data Type	Allow Nulls
ders_id	int	<input type="checkbox"/>
ders_adi	nchar(100)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil 4.1.3: Dersler tablosu alanları

Sorular Tablosu:

Column Name	Data Type	Allow Nulls
soru_id	int	<input type="checkbox"/>
ders_id	int	<input checked="" type="checkbox"/>
soru	nvarchar(MAX)	<input checked="" type="checkbox"/>
soru_resim	nvarchar(MAX)	<input checked="" type="checkbox"/>
a_cevap	nvarchar(MAX)	<input checked="" type="checkbox"/>
b_cevap	nvarchar(MAX)	<input checked="" type="checkbox"/>
c_cevap	nvarchar(MAX)	<input checked="" type="checkbox"/>
d_cevap	nvarchar(MAX)	<input checked="" type="checkbox"/>
dogru_cevap	nvarchar(MAX)	<input checked="" type="checkbox"/>
zorluk_derecesi	nchar(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil 4.1.4: Sorular tablosu alanları**Kullanıcılar Tablosu:**

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
kullanici	nchar(10)	<input checked="" type="checkbox"/>
sifre	nchar(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil 4.1.5: Kullanıcılar tablosu alanları

4.2 Form Tasarımları

Daha sonra projede kullanılacak form ekranları tasarımları yapılmıştır. Projede 4 adet form kullanılmıştır. İlk form hoşgeldiniz ekranıdır. Bu formda imza vardır.

Form6.cs: Hoşgeldiniz Formu



Şekil 4.2.1: Hoşgeldiniz formu tasarımı

Bu formda ortaya yazılımcının ikonu konuldu ve formun yüklenme olayına ikonun formun tam ortasında çıkması için gerekli kodlar yazılmıştır. Formun butonları devre dışı bırakılmıştır ve arka plan rengi siyah olarak belirlenmiştir.

Program aktif edildikten sonra bu form açılır ve silinerek kaybolur.

Form5.cs: Sonuç Okuma Formu

Şekil 4.2.2: Sonuç okuma formu tasarımı

Sonuçlar bu formda okunacaktır. Optik okuyucudan alınan not defteri belgesi form üzerine ya da datagridview üzerine sürüklenip bırakılır ve sonuçlar datagridview'e listelenir.

Form1.cs: Giriş Formu

Şekil 4.2.3: Giriş formu tasarımı

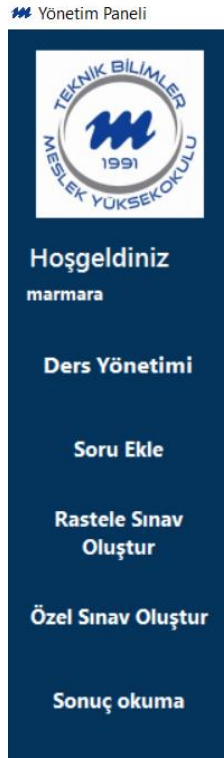
Bu form kullanıcı giriş formudur. Tüm nesnelerin form açıldığı anda ortalanabilmesi için nesneler tek bir panel içerisine alınmıştır. Logo için 1 adet picturebox kullanılmıştır. Ayrıca 3 adet label, 2 adet textbox ve 1 adet buton kullanılmıştır.

Form2.cs: Yönetim Paneli Formu

Şekil 4.2.4: Yönetim paneli formu tasarımı

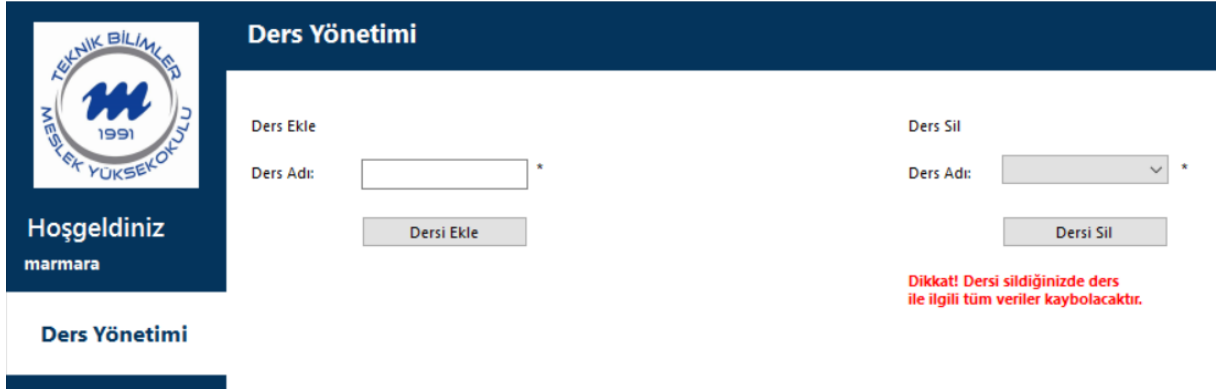
Bu form anaformdur. Panel mantığı kullanılmıştır. Tüm nesneler kategorilendirilmiş ve paneller içerisine eklenmiştir. Kategorilerin ana başlıkları sol tarafa buton olarak eklenmiş ve ilgili butona basıldığı anda ilgili panel ekranda ilgili konuma gider ve Visible özelliği aktif edilir. Geriye kalan panellerinde aynı anda visible özelliği false olarak değiştirilir.

Sol panelde kategoriler bulunmaktadır



Şekil 4.2.5: Kategoriler paneli tasarımı

Ders yönetimi paneli:



Şekil 4.2.6: Ders paneli tasarımı

Bu panelde ders ekleme ve çıkarma işlemleri yapılır.

Soru Ekleme Paneli:

Yönetim Paneli



Hoşgeldiniz
marmara

Ders Yönetimi

Soru Ekle

Rastele Sınav Oluştur

Özel Sınav Oluştur

Sonuç okuma

Soru Ekle

Ders Adı *

Soru *

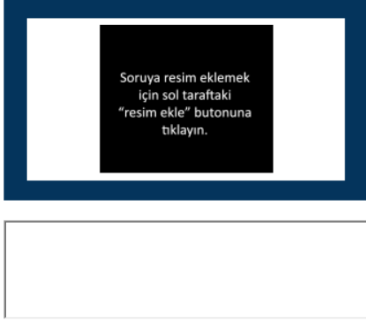
☒ Şıklarda Metin Olacak ☐ Şıklarda Resim Olacak

Şıklar A) * B) * C) * D) * E) *

Doğru Seçenek ☐ A Şıklı ☐ B Şıklı ☐ C Şıklı ☐ D Şıklı ☐ E Şıklı *

Zorluk Derecesi *

Resim Ekle



Şekil 4.2.7: Soru ekleme paneli tasarımı

Bu panelde soru ekleme işlemi yapılır. Sorular için istenilirse resim eklenilebilir. Öncelikle ders seçimi yapılır. Soru yazıldıktan sonra şıklar yazılır. İstenilirse şıklara resimde eklenilebilir. Doğru cevap seçilir. Girilen sorunun zorluk derecesi seçilir ve soru kaydedilir. Soru veritabanına eklenir. Soruyla ilgili eğer resim girildiyse resim c: dizinine açılan DersResimleri isimli klasöre eklenir ve yoluda veritabanında tutulur.

Rastgele Sınav Oluştur Paneli:

Yönetim Paneli

Rastgele Sınav Oluştur

Ders Seçimi * Kolay Soru Sayısı * Orta Soru Sayısı * Zor Soru Sayısı *

Seçilen Sorular

Soru ID	Soru	A Şıkkı	B Şıkkı	C Şıkkı	D Şıkkı	E Şıkkı	Doğru ...	Zorluk ...	Soru R...

Kalan Sorular

Soru ID	Soru	A Şıkkı	B Şıkkı	C Şıkkı	D Şıkkı	E Şıkkı	Doğru ...	Zorluk ...	Soru R...

Toplam Soru Sayısı Kolay Seviye Soru Sayısı Orta Seviye Soru Sayısı Zor Seviye Soru Sayısı

Sayfa Üstbilgisi:

Grup Sayısı * Bölüm Adı *

Vize/Final * Süre *

Soru Resmi

Soru

Şekil 4.2.8: Rastgele sınav oluşturma paneli tasarımı

Bu alan rastgele sınav oluşturma panelidir. Ders seçimi yapıldıktan sonra istenilen miktarda sorular seçilir. Eğer veritabanında istenilen sayı kadar soru yoksa uyarı mesajı olarak verilir. Soru sayısı girildikten sonra soruları getir butonuna basıldığında seçilen sayıda sorular veritabanından rastgele seçilir ve Seçilen sorular alanına listelenir. Buradan sorunun id numarası üzerine tıklanıldığında soru önizleme olarak sağ taraftaki alana gelir. Arta kalan sorular ise alt taraftaki kalan sorular arasına listelenir. İstenilirse buradan seçilen sorular arasına soru girilebilir veya seçilen sorular arasından soru çıkartılabilir. Mevcut soru sayıları ilgili label alanlarında görüntülenir. Textbox lara sınav ile ilgili bilgiler girildikten sonra dokümanı hazırla butonuna basıldığında ilgili doküman Word dosyası olarak hazırlanır ve görüntülenir.

Özel Sınav Oluştur Paneli:

Yönetim Paneli

Özel Sınav Oluştur

Ders Adı: * Zorluk Seviyesi: *

Tüm Sorular

SoruID	Soru	A Şıkkı	B Şıkkı	C Şıkkı	D Şıkkı	E Şıkkı	Doğru ...	Zorluk ...	Resim

Seçilen Sorular

SoruID	Soru	A Şıkkı	B Şıkkı	C Şıkkı	D Şıkkı	E Şıkkı	Doğru ...	Zorluk ...	Resim

Toplam Soru Sayısı: ... Kolay Seviye Soru Sayısı: ... Orta Seviye Soru Sayısı: ... Zor Seviye Soru Sayısı: ...

Grup Sayısı: * Bölüm Adı: *

Vize/Final: * Süre: *

Soru Resm:

Soru:

Şekil 4.2.9: Özel sınav oluşturma paneli tasarımı

Bu alanda eğer sınav soruları tek tek seçilmek istenilirse kullanılır. Ders seçimi ve zorluk seviyesi seçildikten sonra soruları getir butonuna tıklanılır ve sorular Tüm sorular alanında listelenir. Bu liste arasından soru id numarası üzerine tıklanılarak soru önizleme alanında detaylı olarak görüntülenebilir. Soru seçimleri üzerine tıklandıktan sonra ekle butonuna basılarak Seçilen sorular alanına aktarılır. Seçilen soruların sayısına göre label lar üzerinde soru adetleri yazılır.

Sınav hakkındaki gerekli bilgiler girildikten sonra dokümanı hazırla butonuna basılır. Sorular ve cevap anahtarı Word uygulamasında düzenli bir şekilde görüntülenir.

4.3 Projenin Kodlanması

Hoşgeldiniz Ekranı:

Bu formda bir adet timer kullanılmıştır ve bir süre bekledikten sonra formun Opacity değeri timer nesnesiyle azaltılmıştır ve yavaş yavaş kaybolması sağlanmıştır.

```
sayac++;
if (sayac > 100)
{
    this.Opacity -= 0.01;
    if (this.Opacity == 0.00)
    {
        timer1.Stop();
        Form1 frm1 = new Form1();
        frm1.Show();
        this.Hide();
    }
}
```

Giriş Formu:

Bu formda giriş butonuna basıldığında metin kutularına girilen bilgiler alındıktan sonra bu bilgilerin veritabanındaki bilgilerle uyuşup uyuşmadığı kontrol edilir. Eğer bilgiler doğru girildiyse yönetim paneli açılır. Girilen bilgiler veritabanı ile uyuşmazsa yanlış giriş olarak uyarı verilir.

```
SqlCommand cmdkullaniciGetir = new SqlCommand();
cmdkullaniciGetir.Connection = cnn;
cmdkullaniciGetir.CommandText = "SELECT * FROM Kullanicilar WHERE kullanici='" +
kullanici + "' AND sifre='" + sifre + "'";
DataTable dataKullanicilar = new DataTable();
SqlDataAdapter adpKullanicilar = new SqlDataAdapter(cmdkullaniciGetir);
adpKullanicilar.Fill(dataKullanicilar);
if (dataKullanicilar.Rows.Count > 0)
{
    user = kullanici;
    Form2 frm2 = new Form2();
    frm2.Show();
    this.Hide();
}
else
{
    MessageBox.Show("Yanlış Giriş");
}
```

Dersler Paneli:

Burada ders ekle isimli butona basıldığında ilgili metinkutusunda veri olup olmadığını kontrol eder. Eğer veri var ise girilen ders adı veritabanına eklenir. Daha sonra ders ile ilgili girilebilecek soru resimleri için C:\DersResimleri içerisine ders ismi ile yeni bir klasör oluşturur. Son aşama olarak form üzerindeki tüm ders ismi geçen alanlara eklenen dersi ekleyip alanları günceller.

```
private void btnDersEkle_Click(object sender, EventArgs e)
```

```

{
    if (txtDersAdi.Text != "")
    {
        string dersAdi = txtDersAdi.Text;
        SqlCommand cmdDersEkle = new SqlCommand();
        cmdDersEkle.Connection = cnn;
        cmdDersEkle.CommandText = "INSERT INTO Dersler (ders_adi) VALUES ('"
+ dersAdi + "')";
        try
        {
            cmdDersEkle.ExecuteNonQuery();
            if (Directory.Exists(@"C:\DersResimleri\dersler\" + dersAdi))
            {
                MessageBox.Show("Klasör mevcut.");
            }
            else
            {
                Directory.CreateDirectory(@"C:\DersResimleri\dersler\" +
dersAdi);

                MessageBox.Show("Resimler için klasör oluşturuldu.");
            }
            MessageBox.Show(dersAdi + " isimli ders eklendi.");

            //Dersler ilgili comboboxlara yeniden listeleniyor
            SqlCommand cmdDersleriGetir = new SqlCommand();
            cmdDersleriGetir.Connection = cnn;
            cmdDersleriGetir.CommandText = "SELECT * FROM Dersler";
            DataTable dataDersler = new DataTable();
            SqlDataAdapter adpDersler = new
SqlDataAdapter(cmdDersleriGetir);
            adpDersler.Fill(dataDersler);
            cmbDersAdi.Items.Clear();
            cmbSinavDers.Items.Clear();
            cmbOzelDersler.Items.Clear();
            cmbSilinecekDers.Items.Clear();
            for (int i = 0; i < dataDersler.Rows.Count; i++)
            {

                cmbDersAdi.Items.Add(dataDersler.Rows[i]["ders_adi"].ToString().Trim());

                cmbSinavDers.Items.Add(dataDersler.Rows[i]["ders_adi"].ToString().Trim());

                cmbOzelDersler.Items.Add(dataDersler.Rows[i]["ders_adi"].ToString().Trim());

                cmbSilinecekDers.Items.Add(dataDersler.Rows[i]["ders_adi"].ToString().Trim());
            }
            cmbDersAdi.SelectedItem = null;
            cmbSinavDers.SelectedItem = null;
            cmbOzelDersler.SelectedItem = null;
            cmbSilinecekDers.SelectedItem = null;

        }
        catch (Exception hata)
        {
            MessageBox.Show("Ders Eklenemedi. Hata= " + hata.Message);
        }
    }
    else

```

```
{
    MessageBox.Show("Lütfen yıldızlı(*) alanları boş bırakmayınız.");
}
```

Eğer ders sil butonuna tıklanırsa öncelikle uyarı mesajı verilir ve ders ile ilgili tüm verilerin silineceği konusunda kullanıcı uyarılır. Eğer kullanıcı silmeye devam etmek isterse komut ile veritabanında bulunan dersler tablosundan ders silinir. Sorular tablosundan ders ile ilgili tüm sorular silinir ve son aşama olarakta DersResimleri içerisindeki ilgili dersin klasörü silinir.

```
private void btnDersiSil_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ders ile ilgili tüm verileri silmek istediğinizden emin misiniz?", "Ders silme", MessageBoxButtons.YesNo);
    if (dialogResult == DialogResult.Yes)
    {
        int silinecekDersIdsi;
        if (cmbSilinecekDers.SelectedItem != null)
        {
            SqlCommand cmdDersIdsi = new SqlCommand();
            cmdDersIdsi.Connection = cnn;
            cmdDersIdsi.CommandText = "SELECT ders_id FROM Dersler WHERE ders_adi='" + cmbSilinecekDers.SelectedItem.ToString() + "'";
            DataTable dtDersIdsi = new DataTable();
            SqlDataAdapter adpDersIdsi = new SqlDataAdapter(cmdDersIdsi);
            adpDersIdsi.Fill(dtDersIdsi);
            silinecekDersIdsi = Convert.ToInt32(dtDersIdsi.Rows[0]["ders_id"]);
            try
            {
                SqlCommand cmdDerSil = new SqlCommand();
                cmdDerSil.Connection = cnn;
                cmdDerSil.CommandText = "DELETE FROM Dersler WHERE ders_id='" + silinecekDersIdsi + "'";
                cmdDerSil.ExecuteNonQuery();

                SqlCommand cmdDersSoruSil = new SqlCommand();
                cmdDersSoruSil.Connection = cnn;
                cmdDersSoruSil.CommandText = "DELETE FROM Sorular WHERE ders_id='" + silinecekDersIdsi + "'";
                cmdDersSoruSil.ExecuteNonQuery();

                if (Directory.Exists(@"C:\DersResimleri\dersler\" + cmbSilinecekDers.SelectedItem.ToString()))
                {
                    FileInfo fileInfo;
                    string uzanti = ".jpg";

                    foreach (string dosya in Directory.GetFiles(@"C:\DersResimleri\dersler\" + cmbSilinecekDers.SelectedItem.ToString()))
                    {
                        fileInfo = new FileInfo(dosya);
                    }
                }
            }
            catch { }
        }
    }
}
```

```

        if (fileInfo.Extension == uzanti) // Dosya Uzantısı
ve sizin uzantınız aynı mı ??
        {
            fileInfo.Delete();
        }
    }
    Directory.Delete(@"C:\DersResimleri\dersler\" +
cmbSilinecekDers.SelectedItem.ToString());

```

Rastgele Sınav Oluşturma Paneli:

Soruları getir butonuna tıklanıldığında metin kutularında girilen sayılara göre ilgili sorular veritabanından rastgele seçilerek getirilir. Öncelikle tüm sorular çekilir. Sonra rastgele sorular seçilir. Rastgele seçilen sorular tüm sorulardan çıkarılır ve listelenir.

```

SqlCommand cmdDersId = new SqlCommand();
cmdDersId.Connection = cnn;
cmdDersId.CommandText = "SELECT ders_id FROM Dersler WHERE
ders_adi='" + dersinAdi + "'";
DataTable dataDersID = new DataTable();
SqlDataAdapter adpDersID = new SqlDataAdapter(cmdDersId);
adpDersID.Fill(dataDersID);
dersIDsi = Convert.ToInt32(dataDersID.Rows[0]["ders_id"]);

int totalKolay, totalOrta, totalZor;
//Toplam kolay soru sayısı
SqlCommand cmdKolaySorular = new SqlCommand();
cmdKolaySorular.Connection = cnn;
cmdKolaySorular.CommandText = "SELECT * FROM Sorular WHERE
zorluk_derecesi='Kolay' AND ders_id='" + dersIDsi + "'";
DataTable dataKSoruSayisi = new DataTable();
SqlDataAdapter adpKSoruSayisi = new SqlDataAdapter(cmdKolaySorular);
adpKSoruSayisi.Fill(dataKSoruSayisi);
totalKolay = Convert.ToInt32(dataKSoruSayisi.Rows.Count);

//Toplam Orta soru sayısı
SqlCommand cmdOrtaSorular = new SqlCommand();
cmdOrtaSorular.Connection = cnn;
cmdOrtaSorular.CommandText = "SELECT * FROM Sorular WHERE
zorluk_derecesi='Orta' AND ders_id='" + dersIDsi + "'";
DataTable dataOSoruSayisi = new DataTable();
SqlDataAdapter adpOSoruSayisi = new SqlDataAdapter(cmdOrtaSorular);
adpOSoruSayisi.Fill(dataOSoruSayisi);
totalOrta = Convert.ToInt32(dataOSoruSayisi.Rows.Count);

//Toplam Zor soru sayısı
SqlCommand cmdZorSorular = new SqlCommand();
cmdZorSorular.Connection = cnn;
cmdZorSorular.CommandText = "SELECT * FROM Sorular WHERE
zorluk_derecesi='Zor' AND ders_id='" + dersIDsi + "' ";
DataTable dataZSoruSayisi = new DataTable();
SqlDataAdapter adpZSoruSayisi = new SqlDataAdapter(cmdZorSorular);
adpZSoruSayisi.Fill(dataZSoruSayisi);
totalZor = Convert.ToInt32(dataZSoruSayisi.Rows.Count);

```

```

        kolaySayi = Convert.ToInt32(txtSinavKolay.Text);
        ortaSayi = Convert.ToInt32(txtSinavOrta.Text);
        zorSayi = Convert.ToInt32(txtSinavZor.Text);

        if (kolaySayi <= totalKolay && ortaSayi <= totalOrta && zorSayi <=
totalZor)
        {
            /*Form3 frm3 = new Form3();
            frm3.Show();*/
            //Seçilen sayı kadar soruların getirileceği bölüm
            SqlCommand cmdKolayGetir = new SqlCommand();
            cmdKolayGetir.Connection = cnn;
            cmdKolayGetir.CommandText = "SELECT * FROM Sorular WHERE
zorluk_derecesi='Kolay' AND ders_id='" + dersIDsi + "' ORDER By NEWID()";
            DataTable dataKolaylar = new DataTable();
            SqlDataAdapter adpKolaylar = new SqlDataAdapter(cmdKolayGetir);
            adpKolaylar.Fill(dataKolaylar);

            SqlCommand cmdOrtaGetir = new SqlCommand();
            cmdOrtaGetir.Connection = cnn;
            cmdOrtaGetir.CommandText = "SELECT * FROM Sorular WHERE
zorluk_derecesi='Orta' AND ders_id='" + dersIDsi + "' ORDER By NEWID()";
            DataTable dataOrtalar = new DataTable();
            SqlDataAdapter adpOrtalar = new SqlDataAdapter(cmdOrtaGetir);
            adpOrtalar.Fill(dataOrtalar);

            SqlCommand cmdZorGetir = new SqlCommand();
            cmdZorGetir.Connection = cnn;
            cmdZorGetir.CommandText = "SELECT * FROM Sorular WHERE
zorluk_derecesi='Zor' AND ders_id='" + dersIDsi + "' ORDER By NEWID()";
            DataTable dataZorlar = new DataTable();
            SqlDataAdapter adpZorlar = new SqlDataAdapter(cmdZorGetir);
            adpZorlar.Fill(dataZorlar);

            toplamSayi = kolaySayi + ortaSayi + zorSayi;

            for (int i = 0; i < Form2.kolaySayi; i++)
            {
                rndSecilenSorular.Rows.Add(dataKolaylar.Rows[i]["soru_id"].ToString().Trim(),
                dataKolaylar.Rows[i]["soru"].ToString().Trim(),
                dataKolaylar.Rows[i]["a_cevap"].ToString().Trim(),
                dataKolaylar.Rows[i]["b_cevap"].ToString().Trim(),
                dataKolaylar.Rows[i]["c_cevap"].ToString().Trim(),
                dataKolaylar.Rows[i]["d_cevap"].ToString().Trim(),
                dataKolaylar.Rows[i]["e_cevap"].ToString().Trim(),
                dataKolaylar.Rows[i]["dogru_cevap"].ToString().Trim(),
                dataKolaylar.Rows[i]["zorluk_derecesi"].ToString().Trim(),
                dataKolaylar.Rows[i]["soru_resim"].ToString().Trim());
            }
            for (int i = 0; i < Form2.ortaSayi; i++)
            {
                rndSecilenSorular.Rows.Add(dataOrtalar.Rows[i]["soru_id"].ToString().Trim(),
                dataOrtalar.Rows[i]["soru"].ToString().Trim(),

```

```

dataOrtalar.Rows[i]["a_cevap"].ToString().Trim(),
dataOrtalar.Rows[i]["b_cevap"].ToString().Trim(),
dataOrtalar.Rows[i]["c_cevap"].ToString().Trim(),
dataOrtalar.Rows[i]["d_cevap"].ToString().Trim(),
dataOrtalar.Rows[i]["e_cevap"].ToString().Trim(),
dataOrtalar.Rows[i]["dogru_cevap"].ToString().Trim(),
dataOrtalar.Rows[i]["zorluk_derecesi"].ToString().Trim(),
dataOrtalar.Rows[i]["soru_resim"].ToString().Trim());
    }
    for (int i = 0; i < Form2.zorSayi; i++)
    {

rndSecilenSorular.Rows.Add(dataZorlar.Rows[i]["soru_id"].ToString().Trim(),
dataZorlar.Rows[i]["soru"].ToString().Trim(),
dataZorlar.Rows[i]["a_cevap"].ToString().Trim(),
dataZorlar.Rows[i]["b_cevap"].ToString().Trim(),
dataZorlar.Rows[i]["c_cevap"].ToString().Trim(),
dataZorlar.Rows[i]["d_cevap"].ToString().Trim(),
dataZorlar.Rows[i]["e_cevap"].ToString().Trim(),
dataZorlar.Rows[i]["dogru_cevap"].ToString().Trim(),
dataZorlar.Rows[i]["zorluk_derecesi"].ToString().Trim(),
dataZorlar.Rows[i]["soru_resim"].ToString().Trim());
    }

    for (int i = 0; i < rndSecilenSorular.Rows.Count; i++)
    {

1stRndGelen.Items.Add(rndSecilenSorular.Rows[i]["soru_id"].ToString().Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["soru"].ToString().Trim(
));

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["a_cevap"].ToString().Tr
im());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["b_cevap"].ToString().Tr
im());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["c_cevap"].ToString().Tr
im());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["d_cevap"].ToString().Tr
im());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["e_cevap"].ToString().Tr
im());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["dogru_cevap"].ToString(
).Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["zorluk_derecesi"].ToStr
ing().Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["soru_resim"].ToString(
).Trim());
    }
    int[] secilenler = new int[rndSecilenSorular.Rows.Count];
    for (int i = 0; i < rndSecilenSorular.Rows.Count; i++)

```



```

        {
            secilenler[i] =
Convert.ToInt32(rndSecilenSorular.Rows[i]["soru_id"]);
        }

        SqlCommand cmdTumSorular = new SqlCommand();
        cmdTumSorular.Connection = cnn;
        cmdTumSorular.CommandText = "SELECT * FROM Sorular WHERE
ders_id='" + dersIDSi + "'";
        SqlDataAdapter adpTumSorular = new
SqlDataAdapter(cmdTumSorular);
        adpTumSorular.Fill(rndTumSorular);

        for (int i = 0; i < rndSecilenSorular.Rows.Count; i++)
        {
            for (int j = 0; j < rndTumSorular.Rows.Count; j++)
            {
                //MessageBox.Show(secilenler[i].ToString() + " " +
dtKalanSorular.Rows[j]["soru_id"].ToString().Trim());
                if (secilenler[i].ToString() ==
rndTumSorular.Rows[j]["soru_id"].ToString().Trim())
                {
                    rndTumSorular.Rows[j]["soru_id"] = 0;
                    break;
                }
            }
        }

        //Seçme kontrolü
        for (int i = 0; i < rndSecilenSorular.Rows.Count; i++)
        {
            //MessageBox.Show(secilenler[i].ToString());
        }
        //MessageBox.Show("Seçilenler bitti");

        /*for (int i = 0; i < rndTumSorular.Rows.Count; i++)
        {

MessageBox.Show(rndTumSorular.Rows[i]["soru_id"].ToString());
        }*/

        //rndKalanSorular.Rows.Clear();
        for (int i = 0; i < rndTumSorular.Rows.Count; i++)
        {

//MessageBox.Show(rndTumSorular.Rows[i]["soru_id"].ToString());
            if (Convert.ToInt32(rndTumSorular.Rows[i]["soru_id"]) != 0)
            {

//MessageBox.Show(rndTumSorular.Rows[i]["soru_id"].ToString());

rndKalanSorular.Rows.Add(rndTumSorular.Rows[i]["soru_id"].ToString().Trim(),
rndTumSorular.Rows[i]["soru"].ToString().Trim(),
rndTumSorular.Rows[i]["a_cevap"].ToString().Trim(),
rndTumSorular.Rows[i]["b_cevap"].ToString().Trim(),
rndTumSorular.Rows[i]["c_cevap"].ToString().Trim(),

```

```

rndTumSorular.Rows[i]["d_cevap"].ToString().Trim(),
rndTumSorular.Rows[i]["e_cevap"].ToString().Trim(),
rndTumSorular.Rows[i]["dogru_cevap"].ToString().Trim(),
rndTumSorular.Rows[i]["zorluk_derecesi"].ToString().Trim(),
rndTumSorular.Rows[i]["soru_resim"].ToString().Trim());
    }
}

for (int i = 0; i < rndKalanSorular.Rows.Count; i++)
{
    1stRndKalan.Items.Add(rndKalanSorular.Rows[i]["soru_id"].ToString().Trim());

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["soru"].ToString().Trim())
    ;

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["a_cevap"].ToString().Trim
    ());

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["b_cevap"].ToString().Trim
    ());

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["c_cevap"].ToString().Trim
    ());

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["d_cevap"].ToString().Trim
    ());

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["e_cevap"].ToString().Trim
    ());

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["dogru_cevap"].ToString().
    Trim());

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["zorluk_derecesi"].ToStrin
    g().Trim());

    1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i]["soru_resim"].ToString().T
    rim());
}

//MessageBox.Show("Seçilmeyen soru sayısı:" +
rndKalanSorular.Rows.Count.ToString());
//MessageBox.Show("Seçilen soru sayısı:" +
rndSecilenSorular.Rows.Count.ToString());

lblRndToplam.Text = toplamSayi.ToString();
lblRndKolay.Text = kolaySayi.ToString();
lblRndOrta.Text = ortaSayi.ToString();
lblRndZo3r.Text = zorSayi.ToString();
}
else
{
    MessageBox.Show("Veritabanında istediğiniz kadar soru yok.
Mevcut kolay seviye soru sayısı=" + totalKolay.ToString() + " .Mevcut Orta seviye

```

```

soru sayısı=" + totalOrta.ToString() + " .Mevcut zor soru sayısı=" +
totalZor.ToString());
    }

    else
    {
        MessageBox.Show("Lütfen yıldızlı(*) alanları boş bırakmayınız.");
    }

```

Soru çıkar butonuna basıldığında seçilen soru Seçili sorular arasından çıkarılır ve kalan sorular arasına eklenir.

```

private void btnRndCikar_Click(object sender, EventArgs e)
{
    string cikarilan;
    if (rndSeciliId != 0)
    {
        for (int i = 0; i < rndSecilenSorular.Rows.Count; i++)
        {
            if (rndSecilenSorular.Rows[i]["soru_id"].ToString() ==
rndSeciliId.ToString())
            {
                cikarilan =
rndSecilenSorular.Rows[i]["zorluk_derecesi"].ToString().Trim();

rndKalanSorular.Rows.Add(rndSecilenSorular.Rows[i]["soru_id"].ToString().Trim(),
rndSecilenSorular.Rows[i]["soru"].ToString().Trim(),
rndSecilenSorular.Rows[i]["a_cevap"].ToString().Trim(),
rndSecilenSorular.Rows[i]["b_cevap"].ToString().Trim(),
rndSecilenSorular.Rows[i]["c_cevap"].ToString().Trim(),
rndSecilenSorular.Rows[i]["d_cevap"].ToString().Trim(),
rndSecilenSorular.Rows[i]["e_cevap"].ToString().Trim(),
rndSecilenSorular.Rows[i]["dogru_cevap"].ToString().Trim(),
rndSecilenSorular.Rows[i]["zorluk_derecesi"].ToString().Trim(),
rndSecilenSorular.Rows[i]["soru_resim"].ToString().Trim());
                rndSecilenSorular.Rows[i].Delete();
                MessageBox.Show("Seçili soru silindi.");
                if (cikarilan == "Kolay")
                {
                    kolaySayi--;
                }
                else if (cikarilan == "Orta")
                {
                    ortaSayi--;
                }
                else if (cikarilan == "Zor")
                {
                    zorSayi--;
                }
                toplamSayi--;
                lblRndToplam.Text = toplamSayi.ToString();
                lblRndKolay.Text = kolaySayi.ToString();
                lblRndOrta.Text = ortaSayi.ToString();
                lblRndZor.Text = zorSayi.ToString();
            }
        }
    }
}

```

```

        break;
    }

    }

    lstRndGelen.Items.Clear();
    for (int i = 0; i < rndSecilenSorular.Rows.Count; i++)
    {

        lstRndGelen.Items.Add(rndSecilenSorular.Rows[i][ "soru_id" ].ToString().Trim());

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "soru" ].ToString().Trim(
        ));

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "a_cevap" ].ToString().Tr
        im());

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "b_cevap" ].ToString().Tr
        im());

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "c_cevap" ].ToString().Tr
        im());

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "d_cevap" ].ToString().Tr
        im());

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "e_cevap" ].ToString().Tr
        im());

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "dogru_cevap" ].ToString(
        ).Trim());

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "zorluk_derecesi" ].ToStr
        ing().Trim());

        lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "soru_resim" ].ToString()
        .Trim());

    }

    pcbRndOnizleme.Image = null;
    rtxtRndOnizleme.Text = null;
    lstRndKalan.Items.Clear();
    for (int i = 0; i < rndKalanSorular.Rows.Count; i++)
    {

        lstRndKalan.Items.Add(rndKalanSorular.Rows[i][ "soru_id" ].ToString().Trim());

        lstRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "soru" ].ToString().Trim(
        );

        lstRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "a_cevap" ].ToString().Trim
        ());

        lstRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "b_cevap" ].ToString().Trim
        ());

        lstRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "c_cevap" ].ToString().Trim
        ());
    }

```

```

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "d_cevap"].ToString().Trim
());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "e_cevap"].ToString().Trim
());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "dogru_cevap"].ToString().
Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "zorluk_derecesi"].ToStrin
g().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "soru_resim"].ToString().T
rim());
    }
    //MessageBox.Show(rndSecilenSorular.Rows.Count.ToString());
    //MessageBox.Show(rndKalanSorular.Rows.Count.ToString());

    label158.Visible = false;
    label159.Visible = false;
    label160.Visible = false;
    label161.Visible = false;
    label179.Visible = false;
    pcbRndA.Visible = false;
    pcbRndB.Visible = false;
    pcbRndC.Visible = false;
    pcbRndD.Visible = false;
    pcbRndE.Visible = false;
    rndSeciliId = 0;
}
else
{
    MessageBox.Show("Lütfen soru seçin.");
}
}

```

Soru ekle butonuna basıldığı zaman kalan sorulardan seçilen soru Kalan sorular arasından silinir ve Seçilen sorular alanına eklenir.

```

private void btnRndEkle_Click(object sender, EventArgs e)
{
    string eklenen;
    if (rndSeciliIDEkle != 0)
    {
        for (int i = 0; i < rndKalanSorular.Rows.Count; i++)
        {
            if (rndKalanSorular.Rows[i][ "soru_id"].ToString() ==
rndSeciliIDEkle.ToString())
            {
                eklenen =
rndKalanSorular.Rows[i][ "zorluk_derecesi"].ToString().Trim();

```

```

rndSecilenSorular.Rows.Add(rndKalanSorular.Rows[i]["soru_id"].ToString().Trim(),
rndKalanSorular.Rows[i]["soru"].ToString().Trim(),
rndKalanSorular.Rows[i]["a_cevap"].ToString().Trim(),
rndKalanSorular.Rows[i]["b_cevap"].ToString().Trim(),
rndKalanSorular.Rows[i]["c_cevap"].ToString().Trim(),
rndKalanSorular.Rows[i]["d_cevap"].ToString().Trim(),
rndKalanSorular.Rows[i]["e_cevap"].ToString().Trim(),
rndKalanSorular.Rows[i]["dogru_cevap"].ToString().Trim(),
rndKalanSorular.Rows[i]["zorluk_derecesi"].ToString().Trim(),
rndKalanSorular.Rows[i]["soru_resim"].ToString().Trim());
rndKalanSorular.Rows[i].Delete();
//MessageBox.Show("Seçilen soru sınav soruları arasına
eklendi");

if (eklenen == "Kolay")
{
    kolaySayi++;
}
else if (eklenen == "Orta")
{
    ortaSayi++;
}
else if (eklenen == "Zor")
{
    zorSayi++;
}
toplamSayi++;
lblRndToplam.Text = toplamSayi.ToString();
lblRndKolay.Text = kolaySayi.ToString();
lblRndOrta.Text = ortaSayi.ToString();
lblRndZor.Text = zorSayi.ToString();
break;
}
}
label58.Visible = false;
label59.Visible = false;
label60.Visible = false;
label61.Visible = false;
label79.Visible = false;
pcbRndA.Visible = false;
pcbRndB.Visible = false;
pcbRndC.Visible = false;
pcbRndD.Visible = false;
pcbRndE.Visible = false;

lstRndGelen.Items.Clear();
for (int i = 0; i < rndSecilenSorular.Rows.Count; i++)
{
    lstRndGelen.Items.Add(rndSecilenSorular.Rows[i]["soru_id"].ToString().Trim());

    lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["soru"].ToString().Trim());

    lstRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i]["a_cevap"].ToString().Trim());

```

```

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "b_cevap" ].ToString().Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "c_cevap" ].ToString().Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "d_cevap" ].ToString().Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "e_cevap" ].ToString().Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "dogru_cevap" ].ToString().Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "zorluk_derecesi" ].ToString().Trim());

1stRndGelen.Items[i].SubItems.Add(rndSecilenSorular.Rows[i][ "soru_resim" ].ToString().Trim());
    }

    1stRndKalan.Items.Clear();
    for (int i = 0; i < rndKalanSorular.Rows.Count; i++)
    {

1stRndKalan.Items.Add(rndKalanSorular.Rows[i][ "soru_id" ].ToString().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "soru" ].ToString().Trim());
;

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "a_cevap" ].ToString().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "b_cevap" ].ToString().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "c_cevap" ].ToString().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "d_cevap" ].ToString().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "e_cevap" ].ToString().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "dogru_cevap" ].ToString().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "zorluk_derecesi" ].ToString().Trim());

1stRndKalan.Items[i].SubItems.Add(rndKalanSorular.Rows[i][ "soru_resim" ].ToString().Trim());
    }

```

```
        pcbRndOnizleme.ImageLocation = null;
        rtxtRndOnizleme.Text = null;
        rndSeciliIDEkle = 0;
    }
    else
    {
        MessageBox.Show("Lütfen soru seçin.");
    }
}
```


Soruları önizleme ekranında görmek için listelenen sorular arasından görüntülenmek istenilen soru seçilir ve soru önizleme ekranında görüntülenir.

```
pcbRndA.ImageLocation = item.SubItems[2].Text.ToString().Trim();
pcbRndB.ImageLocation = item.SubItems[3].Text.ToString().Trim();
pcbRndC.ImageLocation = item.SubItems[4].Text.ToString().Trim();
pcbRndD.ImageLocation = item.SubItems[5].Text.ToString().Trim();
pcbRndE.ImageLocation = item.SubItems[5].Text.ToString().Trim();
string soru;
soru = "Zorluk Derecesi = " + item.SubItems[8].Text.ToString().Trim() + "\n" + "Soru"
= " + item.SubItems[1].Text.ToString().Trim() + "\n" + "a)" +
item.SubItems[2].Text.ToString().Trim() + "\n" + "b)" +
item.SubItems[3].Text.ToString().Trim() + "\n" + "c)" +
item.SubItems[4].Text.ToString().Trim() + "\n" + "d)" +
item.SubItems[5].Text.ToString().Trim() + "\n" + "e)" +
item.SubItems[6].Text.ToString().Trim() + "\n" + "Doğru Cevap= " +
item.SubItems[7].Text.ToString().Trim();
rtxtRndOnizleme.Text = soru;
```

Sınav hakkında gerekli olan parametreler girildikten sonra(Grup sayısı, Vize/Final, Bölüm Adı, Sınav süresi) Dökümanı hazırla butonuna basıldığında bu verilerle birlikte seçilen sorular Word belgesine aktarılır ve ekranda görüntülenir.

```
private void btnRndWord_Click(object sender, EventArgs e)
{
    int soruSayisi = 0;

    if (rndSecilenSorular.Rows.Count > 0)
    {
        if ((cmbRndGrup.SelectedItem.ToString() != "" ||
cmbRndGrup.SelectedItem != null) && (cmbRndVF.SelectedItem.ToString() != "" ||
cmbRndVF.SelectedItem != null) && txtRndBolum.Text != "" && txtRndSure.Text != "" &&
(cmbSinavDers.SelectedItem.ToString() != "" || cmbSinavDers.SelectedItem != null))
        {
            int tabloSatirSayisi = 0;
            int sikSayisi = 0, resimSayisi = 0;

            soruSayisi = rndSecilenSorular.Rows.Count;
            sikSayisi = soruSayisi;
            for (int i = 0; i < soruSayisi; i++)
            {
                if (rndSecilenSorular.Rows[i]["soru_resim"].ToString() !=
null && rndSecilenSorular.Rows[i]["soru_resim"].ToString() != "")
                {
                    resimSayisi++;
                }
            }
            tabloSatirSayisi = soruSayisi + sikSayisi + resimSayisi;

            if (cmbRndGrup.SelectedItem.ToString() == "Tek Grup")
            {
                object omissing = System.Reflection.Missing.Value;
                object son = "\\endofdoc";
```

```

word.Application olustur;
word.Document icerik;
olustur = new word.Application();
olustur.Visible = true;
icerik = olustur.Documents.Add(ref omissing);

icerik.PageSetup.TextColumns.SetCount(2);// Sayfayı ikiye
bölme

word.Range headerRangeA =
icerik.Sections[1].Headers[word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;
headerRangeA.Fields.Add(headerRangeA,
word.WdFieldType.wdFieldPage);
headerRangeA.ParagraphFormat.Alignment =
word.WdParagraphAlignment.wdAlignParagraphCenter;

headerRangeA.Font.Size = 10;
headerRangeA.Font.Bold = 1;
string sinavKontrol = cmbRndVF.SelectedItem.ToString();
if (sinavKontrol == "Vize")
{
    sinavKontrol = "VİZESİ";
}
else if (sinavKontrol == "Final")
{
    sinavKontrol = "FİNALİ";
}
string ustbilgi = "MARMARA ÜNİVERSİTESİ TEKNİK BİLİMLER
MESLEK YÜKSEKOKULU\n" + txtRndBolum.Text.ToString().ToUpper() + " PROGRAMI " +
cmbSinavDers.SelectedItem.ToString().ToUpper() + " DERSİ " + sinavKontrol.ToString()
+ "\nSINAV SÜRESİ=" + txtRndSure.Text;
headerRangeA.Text = ustbilgi + "\n\nAd:
Soyad: No:
\n";

word.Table oTable;
word.Range wrdRng = icerik.Bookmarks.get_Item(ref
son).Range;
oTable = icerik.Tables.Add(wrdRng, tabloSatirSayisi, 1, ref
omissing, ref omissing);
oTable.Range.ParagraphFormat.SpaceAfter = 6;
oTable.Range.Font.Size = 10;

string soru = "";
string a = "";
string b = "";
string c = "";
string d = "";
string eC = "";
int sayac = 0;
string siklar = "";

```

[illegible]

```

                oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@eC), Type.Missing, Type.Missing, Type.Missing);

                }
                else
                {
                    oTable.Cell(i, 1).Range.Text = siklar;
                }
            }
            else
            {
                oTable.Cell(i, 1).Range.Text = siklar;
            }
        }
    }
    else
    {

        oTable.Cell(i, 1).Range.Font.Bold = 1;

        oTable.Cell(i, 1).Range.Text = tumSoru;
        i++;
        oTable.Cell(i, 1).Range.Font.Bold = 0;
        if
(rndSecilenSorular.Rows[sayac]["a_cevap"].ToString().Length >= 3)
        {
            if
(rndSecilenSorular.Rows[sayac]["a_cevap"].ToString().Trim().Substring(0, 3) ==
@"C:\")
            {
                oTable.Rows[i].Cells[1].Split(3, 4);

                oTable.Cell(i, 1).Range.Text = "a)";
                oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@a), Type.Missing, Type.Missing, Type.Missing);

                oTable.Cell(i, 3).Range.Text = "b)";
                oTable.Cell(i,
4).Range.InlineShapes.AddPicture((@b), Type.Missing, Type.Missing, Type.Missing);
                i++;
                oTable.Cell(i, 1).Range.Text = "c)";
                oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@c), Type.Missing, Type.Missing, Type.Missing);

                oTable.Cell(i, 3).Range.Text = "d)";
                oTable.Cell(i,
4).Range.InlineShapes.AddPicture((@d), Type.Missing, Type.Missing, Type.Missing);
                i++;
                oTable.Cell(i, 1).Range.Text = "e)";
                oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@eC), Type.Missing, Type.Missing, Type.Missing);

                }
                else
                {
                    oTable.Cell(i, 1).Range.Text = siklar;
                }
            }
        }
    }
    else

```

```

        {
            oTable.Cell(i, 1).Range.Text = siklar;
        }

    }
    sayac++;
}

//-----Cevap Anahtarı

object omissing2 = System.Reflection.Missing.Value;
object son2 = "\\endofdoc";

word.Application cevapAnahtari;
word.Document icerik2;
cevapAnahtari = new word.Application();
cevapAnahtari.Visible = true;
icerik2 = cevapAnahtari.Documents.Add(ref omissing2);

icerik2.PageSetup.TextColumns.SetCount(2);// Sayfayı ikiye
bölme

word.Range headerRangeCevaplar =
icerik2.Sections[1].Headers[word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;
headerRangeCevaplar.Fields.Add(headerRangeCevaplar,
word.WdFieldType.wdFieldPage);
headerRangeCevaplar.ParagraphFormat.Alignment =
word.WdParagraphAlignment.wdAlignParagraphCenter;

headerRangeCevaplar.Font.Size = 10;
headerRangeCevaplar.Font.Bold = 1;
if (sinavKontrol == "Vize")
{
    sinavKontrol = "VİZESİ";
}
else if (sinavKontrol == "Final")
{
    sinavKontrol = "FİNALİ";
}
ustbilgi = "MARMARA ÜNİVERSİTESİ TEKNİK BİLİMLER MESLEK
YÜKSEKOKULU\n" + txtRndBolum.Text.ToString().ToUpper() + " PROGRAMI " +
cmbSinavDers.SelectedItem.ToString().ToUpper() + " DERSİ " + sinavKontrol.ToString()
+ "\nSINAV SÜRESİ=" + txtRndSure.Text;
headerRangeCevaplar.Text = ustbilgi + "\nCEVAP ANAHTARI";

word.Table oTableC;
word.Range wrdRngC = icerik2.Bookmarks.get_Item(ref
son2).Range;

oTableC = icerik2.Tables.Add(wrdRngC, tabloSatirSayisi, 1,
ref omissing2, ref omissing2);
oTableC.Range.ParagraphFormat.SpaceAfter = 6;
oTableC.Range.Font.Size = 10;

string soru2 = "";
int sayac2 = 0;

```

```

        int cevapSatirSayisi = soruSayisi * 2;
        string tumSoru2 = "";
        string dogruSik = "";
        string dogruCevap = "";
        for (int i = 1; i <= cevapSatirSayisi; i++)
        {
            soru2 =
rndSecilenSorular.Rows[sayac2]["soru"].ToString().Trim();
            tumSoru2 = (sayac2 + 1).ToString() + "-" + soru2;
            dogruCevap =
rndSecilenSorular.Rows[sayac2]["dogru_cevap"].ToString().Trim();
            if
(rndSecilenSorular.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
rndSecilenSorular.Rows[sayac2]["a_cevap"].ToString().Trim())
            {
                dogruSik = "A-");
            }
            else if
(rndSecilenSorular.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
rndSecilenSorular.Rows[sayac2]["b_cevap"].ToString().Trim())
            {
                dogruSik = "B-");
            }
            else if
(rndSecilenSorular.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
rndSecilenSorular.Rows[sayac2]["c_cevap"].ToString().Trim())
            {
                dogruSik = "C-");
            }
            else if
(rndSecilenSorular.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
rndSecilenSorular.Rows[sayac2]["d_cevap"].ToString().Trim())
            {
                dogruSik = "D-");
            }
            else if
(rndSecilenSorular.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
rndSecilenSorular.Rows[sayac2]["e_cevap"].ToString().Trim())
            {
                dogruSik = "E-");
            }
            oTableC.Cell(i, 1).Range.Font.Bold = 1;
            oTableC.Cell(i, 1).Range.Text = tumSoru2;
            i++;
            oTableC.Cell(i, 1).Range.Font.Bold = 0;
            oTableC.Cell(i, 1).Range.Text = dogruSik + dogruCevap;
            sayac2++;
        }
    }

    if (cmbRndGrp.SelectedItem.ToString() == "İki Grup")
    {
        DataTable dtA = new DataTable();
        DataTable dtB = new DataTable();

        dtA.Columns.Add("soru_id");
        dtA.Columns.Add("soru");
    }

```

```
dtA.Columns.Add("a_cevap");
dtA.Columns.Add("b_cevap");
dtA.Columns.Add("c_cevap");
dtA.Columns.Add("d_cevap");
dtA.Columns.Add("dogru_cevap");
dtA.Columns.Add("zorluk_derecesi");
dtA.Columns.Add("soru_resim");
```

```
dtB.Columns.Add("soru_id");
dtB.Columns.Add("soru");
dtB.Columns.Add("a_cevap");
dtB.Columns.Add("b_cevap");
dtB.Columns.Add("c_cevap");
dtB.Columns.Add("d_cevap");
dtB.Columns.Add("dogru_cevap");
dtB.Columns.Add("zorluk_derecesi");
dtB.Columns.Add("soru_resim");
```

```
dtA.Merge(rndSecilenSorular);
```

```
dtB.Merge(rndSecilenSorular);
DataView dw = dtB.DefaultView;
dw.Sort = "a_cevap";
dtB = dw.ToTable();
```

```
object omissing = System.Reflection.Missing.Value;
object son = "\\endofdoc";
```

```
word.Application olustur;
word.Document icerik;
olustur = new word.Application();
olustur.Visible = true;
icerik = olustur.Documents.Add(ref omissing);
```

```
icerik.PageSetup.TextColumns.SetCount(2);// Sayfayı ikiye
```

bölme

```
word.Range headerRangeA =
icerik.Sections[1].Headers[word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;
headerRangeA.Fields.Add(headerRangeA,
word.WdFieldType.wdFieldPage);
headerRangeA.ParagraphFormat.Alignment =
word.WdParagraphAlignment.wdAlignParagraphCenter;
```

```
headerRangeA.Font.Size = 10;
headerRangeA.Font.Bold = 1;
string sinavKontrol = cmbRndVF.SelectedItem.ToString();
if (sinavKontrol == "Vize")
{
    sinavKontrol = "VİZESİ";
}
else if (sinavKontrol == "Final")
{
    sinavKontrol = "FİNALİ";
}
```

```

        string ustbilgi = "MARMARA ÜNİVERSİTESİ TEKNİK BİLİMLER
MESLEK YÜKSEKOKULU\n" + txtRndBolum.Text.ToString().ToUpper() + " PROGRAMI " +
cmbSinavDers.SelectedItem.ToString().ToUpper() + " DERSİ " + sinavKontrol.ToString()
+ "\nSINAV SÜRESİ=" + txtRndSure.Text.ToString().ToUpper();
        headerRangeA.Text = ustbilgi + "\n\nAd:

```

```

Soyad:
\nA GURUBU";

```

```

No:

```

```

        word.Table oTable;
        word.Range wrdRng = icerik.Bookmarks.get_Item(ref
son).Range;
        oTable = icerik.Tables.Add(wrdRng, tabloSatirSayisi, 1, ref
omissing, ref omising);
        oTable.Range.ParagraphFormat.SpaceAfter = 6;
        oTable.Range.Font.Size = 10;

```

```

        string soru = "";
        string a = "";
        string b = "";
        string c = "";
        string d = "";
        string eC = "";
        int sayac = 0;
        string siklar = "";
        string tumSoru = "";
        string soruResmi = "";

        for (int i = 1; i <= tabloSatirSayisi; i++)
        {
            soru = dtA.Rows[sayac]["soru"].ToString().Trim();
            a = dtA.Rows[sayac]["a_cevap"].ToString().Trim();
            b = dtA.Rows[sayac]["b_cevap"].ToString().Trim();
            c = dtA.Rows[sayac]["c_cevap"].ToString().Trim();
            d = dtA.Rows[sayac]["d_cevap"].ToString().Trim();
            eC = dtA.Rows[sayac]["e_cevap"].ToString().Trim();
            siklar = "a)" + a + "\nb)" + b + "\nc)" + c + "\nd)" + d
+ "\ne)" + eC + "\n";
            tumSoru = (sayac + 1).ToString() + "-" + soru;
            soruResmi =
dtA.Rows[sayac]["soru_resim"].ToString().Trim();
            if (soruResmi != null && soruResmi != "")
            {
                oTable.Cell(i,
1).Range.InlineShapes.AddPicture((@soruResmi), Type.Missing, Type.Missing,
Type.Missing);

```

```

                i++;
                oTable.Cell(i, 1).Range.Font.Bold = 1;

                oTable.Cell(i, 1).Range.Text = tumSoru;
                i++;
                oTable.Cell(i, 1).Range.Font.Bold = 0;

                if (dtA.Rows[sayac]["a_cevap"].ToString().Length >=
3)
                {

```



```

if
(dtA.Rows[sayac]["a_cevap"].ToString().Trim().Substring(0, 3) == @"C:\")
{
    oTable.Rows[i].Cells[1].Split(3, 4);
    oTable.Cell(i, 1).Range.Text = "a";
    oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@a), Type.Missing, Type.Missing, Type.Missing);

    oTable.Cell(i, 3).Range.Text = "b";
    oTable.Cell(i,
4).Range.InlineShapes.AddPicture((@b), Type.Missing, Type.Missing, Type.Missing);
    i++;
    oTable.Cell(i, 1).Range.Text = "c";
    oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@c), Type.Missing, Type.Missing, Type.Missing);

    oTable.Cell(i, 3).Range.Text = "d";
    oTable.Cell(i,
4).Range.InlineShapes.AddPicture((@d), Type.Missing, Type.Missing, Type.Missing);
    i++;
    oTable.Cell(i, 1).Range.Text = "e";
    oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@eC), Type.Missing, Type.Missing, Type.Missing);

}
else
{
    oTable.Cell(i, 1).Range.Text = siklar;
}
}
else
{
    oTable.Cell(i, 1).Range.Text = siklar;
}
}
else
{
    oTable.Cell(i, 1).Range.Font.Bold = 1;

    oTable.Cell(i, 1).Range.Text = tumSoru;
    i++;
    oTable.Cell(i, 1).Range.Font.Bold = 0;

    if (dtA.Rows[sayac]["a_cevap"].ToString().Length >=
3)
    {
        if
(dtA.Rows[sayac]["a_cevap"].ToString().Trim().Substring(0, 3) == @"C:\")
        {
            oTable.Rows[i].Cells[1].Split(3, 4);

            oTable.Cell(i, 1).Range.Text = "a";
            oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@a), Type.Missing, Type.Missing, Type.Missing);

            oTable.Cell(i, 3).Range.Text = "b";

```

```

oTable.Cell(i,
4).Range.InlineShapes.AddPicture((@b), Type.Missing, Type.Missing, Type.Missing);
i++;
oTable.Cell(i, 1).Range.Text = "c)";
oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@c), Type.Missing, Type.Missing, Type.Missing);

oTable.Cell(i, 3).Range.Text = "d)";
oTable.Cell(i,
4).Range.InlineShapes.AddPicture((@d), Type.Missing, Type.Missing, Type.Missing);
i++;
oTable.Cell(i, 1).Range.Text = "e)";
oTable.Cell(i,
2).Range.InlineShapes.AddPicture((@eC), Type.Missing, Type.Missing, Type.Missing);

}
else
{
oTable.Cell(i, 1).Range.Text = siklar;
}
}
else
{
oTable.Cell(i, 1).Range.Text = siklar;
}
}
sayac++;
}

```

//-----A Cevap Anahtarı

```

object omissing2 = System.Reflection.Missing.Value;
object son2 = "\\endofdoc";

```

```

word.Application cevapAnahtari;
word.Document icerik2;
cevapAnahtari = new word.Application();
cevapAnahtari.Visible = true;
icerik2 = cevapAnahtari.Documents.Add(ref omissing2);

```

icerik2.PageSetup.TextColumns.SetCount(2);// Sayfayı ikiye bölme

```

word.Range headerRangeCevaplar =
icerik2.Sections[1].Headers[word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;
headerRangeCevaplar.Fields.Add(headerRangeCevaplar,
word.WdFieldType.wdFieldPage);
headerRangeCevaplar.ParagraphFormat.Alignment =
word.WdParagraphAlignment.wdAlignParagraphCenter;

```

```

headerRangeCevaplar.Font.Size = 10;
headerRangeCevaplar.Font.Bold = 1;
if (sinavKontrol == "Vize")
{
sinavKontrol = "VİZESİ";
}

```

```

else if (sinavKontrol == "Final")
{
    sinavKontrol = "FİNALİ";
}
ustbilgi = "MARMARA ÜNİVERSİTESİ TEKNİK BİLİMLER MESLEK
YÜKSEKOKULU\n" + txtRndBolum.Text.ToString().ToUpper() + " PROGRAMI " +
cmbSinavDers.SelectedItem.ToString().ToUpper() + " DERSİ " + sinavKontrol.ToString()
+ "\nSINAV SÜRESİ=" + txtRndSure.Text.ToString().ToUpper();
headerRangeCevaplar.Text = ustbilgi + "\nA GURUBU CEVAP
ANAHTARI";

word.Table oTableC;
word.Range wrdRngC = icerik2.Bookmarks.get_Item(ref
son2).Range;

oTableC = icerik2.Tables.Add(wrdRngC, tabloSatirSayisi, 1,
ref omissing2, ref omissing2);
oTableC.Range.ParagraphFormat.SpaceAfter = 6;
oTableC.Range.Font.Size = 10;

string soru2 = "";
int sayac2 = 0;
int cevapSatirSayisi = soruSayisi * 2;
string tumSoru2 = "";
string dogruSik = "";
string dogruCevap = "";
for (int i = 1; i <= cevapSatirSayisi; i++)
{
    soru2 = dtA.Rows[sayac2]["soru"].ToString().Trim();
    tumSoru2 = (sayac2 + 1).ToString() + "-" + soru2;
    dogruCevap =
dtA.Rows[sayac2]["dogru_cevap"].ToString().Trim();
    if (dtA.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtA.Rows[sayac2]["a_cevap"].ToString().Trim())
    {
        dogruSik = "A-");
    }
    else if
(dtA.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtA.Rows[sayac2]["b_cevap"].ToString().Trim())
    {
        dogruSik = "B-");
    }
    else if
(dtA.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtA.Rows[sayac2]["c_cevap"].ToString().Trim())
    {
        dogruSik = "C-");
    }
    else if
(dtA.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtA.Rows[sayac2]["d_cevap"].ToString().Trim())
    {
        dogruSik = "D-");
    }
    else if
(dtA.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtA.Rows[sayac2]["e_cevap"].ToString().Trim())

```

```

        {
            dogruSik = "E-");
        }
        oTableC.Cell(i, 1).Range.Font.Bold = 1;
        oTableC.Cell(i, 1).Range.Text = tumSoru2;
        i++;
        oTableC.Cell(i, 1).Range.Font.Bold = 0;
        oTableC.Cell(i, 1).Range.Text = dogruSik + dogruCevap;
        sayac2++;

    }
    // B GRUBU

    object omissingB = System.Reflection.Missing.Value;
    object sonB = "\\endofdoc";

    word.Application olusturB;
    word.Document icerikB;
    olusturB = new word.Application();
    olusturB.Visible = true;
    icerikB = olusturB.Documents.Add(ref omissingB);

    icerikB.PageSetup.TextColumns.SetCount(2);// Sayfayı ikiye
bölme

    word.Range headerRangeB =
icerikB.Sections[1].Headers[word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;
    headerRangeB.Fields.Add(headerRangeB,
word.WdFieldType.wdFieldPage);
    headerRangeB.ParagraphFormat.Alignment =
word.WdParagraphAlignment.wdAlignParagraphCenter;

    headerRangeB.Font.Size = 10;
    headerRangeB.Font.Bold = 1;
    sinavKontrol = cmbRndVF.SelectedItem.ToString();
    if (sinavKontrol == "Vize")
    {
        sinavKontrol = "VİZESİ";
    }
    else if (sinavKontrol == "Final")
    {
        sinavKontrol = "FİNALİ";
    }
    ustbilgi = "MARMARA ÜNİVERSİTESİ TEKNİK BİLİMLER MESLEK
YÜKSEKOKULU\n" + txtRndBolum.Text.ToString().ToUpper() + " PROGRAMI " +
cmbSinavDers.SelectedItem.ToString().ToUpper() + " DERSİ " + sinavKontrol.ToString()
+ "\nSINAV SÜRESİ=" + txtRndSure.Text.ToString().ToUpper();
    headerRangeB.Text = ustbilgi + "\n\nAd:
Soyad:                               No:
\nB GURUBU";

    word.Table oTableB;

```

```

word.Range wrdRngB = icerikB.Bookmarks.get_Item(ref
sonB).Range;
oTableB = icerikB.Tables.Add(wrdRngB, tabloSatirSayisi, 1,
ref omissingB, ref omissingB);
oTableB.Range.ParagraphFormat.SpaceAfter = 6;
oTableB.Range.Font.Size = 10;

soru = "";
a = "";
b = "";
c = "";
d = "";
eC = "";
//dogru = "";
sayac = 0;
siklar = "";
tumSoru = "";
soruResmi = "";

for (int i = 1; i <= tabloSatirSayisi; i++)
{
    soru = dtB.Rows[sayac]["soru"].ToString().Trim();
    a = dtB.Rows[sayac]["a_cevap"].ToString().Trim();
    b = dtB.Rows[sayac]["b_cevap"].ToString().Trim();
    c = dtB.Rows[sayac]["c_cevap"].ToString().Trim();
    d = dtB.Rows[sayac]["d_cevap"].ToString().Trim();
    eC = dtB.Rows[sayac]["e_cevap"].ToString().Trim();
    siklar = "a)" + a + "\nb)" + b + "\nc)" + c + "\nd)" + d
+ "\ne)" + eC + "\n";
    tumSoru = (sayac + 1).ToString() + "-" + soru;
    soruResmi =
dtB.Rows[sayac]["soru_resim"].ToString().Trim();
    if (soruResmi != null && soruResmi != "")
    {
        oTableB.Cell(i,
1).Range.InlineShapes.AddPicture((@soruResmi), Type.Missing, Type.Missing,
Type.Missing);
        i++;
        oTableB.Cell(i, 1).Range.Font.Bold = 1;

        oTableB.Cell(i, 1).Range.Text = tumSoru;
        i++;
        oTableB.Cell(i, 1).Range.Font.Bold = 0;
        if (dtB.Rows[sayac]["a_cevap"].ToString().Length >=
3)
        {
            if
(dtB.Rows[sayac]["a_cevap"].ToString().Trim().Substring(0, 3) == @"C:\")
            {
                oTableB.Rows[i].Cells[1].Split(3, 4);
                oTableB.Cell(i, 1).Range.Text = "a)";
                oTableB.Cell(i,
2).Range.InlineShapes.AddPicture((@a), Type.Missing, Type.Missing, Type.Missing);

                oTableB.Cell(i, 3).Range.Text = "b)";

```

```

        oTableB.Cell(i,
4).Range.InlineShapes.AddPicture((@b), Type.Missing, Type.Missing, Type.Missing);
        i++;
        oTableB.Cell(i, 1).Range.Text = "c";
        oTableB.Cell(i,
2).Range.InlineShapes.AddPicture((@c), Type.Missing, Type.Missing, Type.Missing);

        oTableB.Cell(i, 3).Range.Text = "d";
        oTableB.Cell(i,
4).Range.InlineShapes.AddPicture((@d), Type.Missing, Type.Missing, Type.Missing);
        i++;
        oTableB.Cell(i, 1).Range.Text = "e";
        oTableB.Cell(i,
2).Range.InlineShapes.AddPicture((@eC), Type.Missing, Type.Missing, Type.Missing);

    }
    else
    {
        oTableB.Cell(i, 1).Range.Text = siklar;
    }
}

else
{
    oTableB.Cell(i, 1).Range.Text = siklar;
}
}
else
{

    oTableB.Cell(i, 1).Range.Font.Bold = 1;

    oTableB.Cell(i, 1).Range.Text = tumSoru;
    i++;
    oTableB.Cell(i, 1).Range.Font.Bold = 0;
    if (dtB.Rows[sayac]["a_cevap"].ToString().Length >=
3)
    {
        if
(dtB.Rows[sayac]["a_cevap"].ToString().Trim().Substring(0, 3) == @"C:\")
        {
            oTableB.Rows[i].Cells[1].Split(3, 4);

            oTableB.Cell(i, 1).Range.Text = "a";
            oTableB.Cell(i,
2).Range.InlineShapes.AddPicture((@a), Type.Missing, Type.Missing, Type.Missing);

            oTableB.Cell(i, 3).Range.Text = "b";
            oTableB.Cell(i,
4).Range.InlineShapes.AddPicture((@b), Type.Missing, Type.Missing, Type.Missing);
            i++;
            oTableB.Cell(i, 1).Range.Text = "c";
            oTableB.Cell(i,
2).Range.InlineShapes.AddPicture((@c), Type.Missing, Type.Missing, Type.Missing);

            oTableB.Cell(i, 3).Range.Text = "d";
            oTableB.Cell(i,
4).Range.InlineShapes.AddPicture((@d), Type.Missing, Type.Missing, Type.Missing);

```

```

                i++;
                oTableB.Cell(i, 1).Range.Text = "e");
                oTableB.Cell(i,
2).Range.InlineShapes.AddPicture(@eC), Type.Missing, Type.Missing, Type.Missing);

            }
            else
            {
                oTableB.Cell(i, 1).Range.Text = siklar;
            }
        }
        else
        {
            oTableB.Cell(i, 1).Range.Text = siklar;
        }
    }
    sayac++;
}

```

//B Grubu Cevap Anahtarı

```

object omissing3 = System.Reflection.Missing.Value;
object son3 = "\\endofdoc";

```

```

word.Application cevapAnahtariB;
word.Document icerik3;
cevapAnahtariB = new word.Application();
cevapAnahtariB.Visible = true;
icerik3 = cevapAnahtariB.Documents.Add(ref omissing3);

```

```

icerik3.PageSetup.TextColumns.SetCount(2);// Sayfayı ikiye

```

bölme

```

word.Range headerRangeCevaplarB =
icerik3.Sections[1].Headers[word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;
headerRangeCevaplarB.Fields.Add(headerRangeCevaplarB,
word.WdFieldType.wdFieldPage);
headerRangeCevaplarB.ParagraphFormat.Alignment =
word.WdParagraphAlignment.wdAlignParagraphCenter;

headerRangeCevaplarB.Font.Size = 10;
headerRangeCevaplarB.Font.Bold = 1;
if (sinavKontrol == "Vize")
{
    sinavKontrol = "VİZESİ";
}
else if (sinavKontrol == "Final")
{
    sinavKontrol = "FİNALİ";
}
ustbilgi = "MARMARA ÜNİVERSİTESİ TEKNİK BİLİMLER MESLEK
YÜKSEKOKULU\n" + txtRndBolum.Text.ToString().ToUpper() + " PROGRAMI " +
cmbSinavDers.SelectedItem.ToString().ToUpper() + " DERSİ " + sinavKontrol.ToString()
+ "\nSINAV SÜRESİ=" + txtRndSure.Text.ToString().ToUpper();
headerRangeCevaplarB.Text = ustbilgi + "\nB GURUBU CEVAP
ANAHTARI";

```

```

word.Table oTableBC;
word.Range wrdRngBC = icerik3.Bookmarks.get_Item(ref
son3).Range;

oTableBC = icerik3.Tables.Add(wrdRngBC, tabloSatirSayisi, 1,
ref omissing2, ref omissing2);
oTableBC.Range.ParagraphFormat.SpaceAfter = 6;
oTableBC.Range.Font.Size = 10;

soru2 = "";
sayac2 = 0;
cevapSatirSayisi = soruSayisi * 2;
tumSoru2 = "";
dogruSik = "";
dogruCevap = "";
for (int i = 1; i <= cevapSatirSayisi; i++)
{
    soru2 = dtB.Rows[sayac2]["soru"].ToString().Trim();
    tumSoru2 = (sayac2 + 1).ToString() + "-" + soru2;
    dogruCevap =
dtB.Rows[sayac2]["dogru_cevap"].ToString().Trim();
    if (dtB.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtB.Rows[sayac2]["a_cevap"].ToString().Trim())
    {
        dogruSik = "A-");
    }
    else if
(dtB.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtB.Rows[sayac2]["b_cevap"].ToString().Trim())
    {
        dogruSik = "B-");
    }
    else if
(dtB.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtB.Rows[sayac2]["c_cevap"].ToString().Trim())
    {
        dogruSik = "C-");
    }
    else if
(dtB.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtB.Rows[sayac2]["d_cevap"].ToString().Trim())
    {
        dogruSik = "D-");
    }
    else if
(dtB.Rows[sayac2]["dogru_cevap"].ToString().Trim() ==
dtB.Rows[sayac2]["e_cevap"].ToString().Trim())
    {
        dogruSik = "E-");
    }
    oTableBC.Cell(i, 1).Range.Font.Bold = 1;
    oTableBC.Cell(i, 1).Range.Text = tumSoru2;
    i++;
    oTableBC.Cell(i, 1).Range.Font.Bold = 0;
    oTableBC.Cell(i, 1).Range.Text = dogruSik + dogruCevap;
    sayac2++;
}

```



```

        }
    }
}
else
{
    MessageBox.Show("Lütfen yıldızlı(*) alanları boş bırakmayınız.");
}

}
else
{
    MessageBox.Show("Seçilen sorular listesinde soru olmalıdır.");
}
}
}

```

Özel Sınav Hazırlama Paneli:

Bu panelde sorular özel olarak tek tek seçilir. Seçilen dersin seçilen zorluk seviyesinde tüm sorular listelenir ve sorular arasından tek tek seçi yapılır ve doküman hazırlanır. Yine bu paneldede sorular için önizleme ekranı bulunur ve soruyu önizlemek için soru id numarasına tıklamak yeterlidir.

Soruları listelemek için aşağıdaki kodlar kullanılmıştır.

```

private void btnOzelGetir_Click(object sender, EventArgs e)
{
    if (cmbOzelDersler.SelectedItem == null || cmbOzelZorluk.SelectedItem == null)
    {
        MessageBox.Show("Lütfen yıldızlı(*) alanları boş bırakmayınız.");
    }
    else
    {
        dataGelenSorular.Rows.Clear();
        string seviyeOzel, dersAdiOzel;
        seviyeOzel = cmbOzelZorluk.SelectedItem.ToString();

        int dersIdOzel;
        dersAdiOzel = cmbOzelDersler.SelectedItem.ToString();
        SqlCommand cmdDersID = new SqlCommand();
        cmdDersID.Connection = cnn;
        cmdDersID.CommandText = "SELECT ders_id FROM Dersler WHERE ders_adi
= '" + dersAdiOzel + "'";
        DataTable dataDersID = new DataTable();
        SqlDataAdapter adpDersID = new SqlDataAdapter(cmdDersID);
        adpDersID.Fill(dataDersID);
        dersIdOzel = Convert.ToInt32(dataDersID.Rows[0]["ders_id"]);

        SqlCommand cmdSorular = new SqlCommand();
        cmdSorular.Connection = cnn;
        if (seviyeOzel == "Tümü")
        {

```

```

        cmdSorular.CommandText = "SELECT * FROM Sorular WHERE ders_id='"
+ dersIdOzel + "'";
    }
    else
    {
        cmdSorular.CommandText = "SELECT * FROM Sorular WHERE ders_id='"
+ dersIdOzel + "' AND zorluk_derecesi='" + seviyeOzel + "'";
    }
    SqlDataAdapter adpGelenSorular = new SqlDataAdapter(cmdSorular);
    adpGelenSorular.Fill(dataGelenSorular);

    lstGelenSoruar.Items.Clear();

    for (int i = 0; i < dataGelenSorular.Rows.Count; i++)
    {
        lstGelenSoruar.Items.Add(dataGelenSorular.Rows[i]["soru_id"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["soru"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["a_cevap"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["b_cevap"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["c_cevap"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["d_cevap"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["e_cevap"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["dogru_cevap"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["zorluk_derecesi"].ToString().Trim());

        lstGelenSoruar.Items[i].SubItems.Add(dataGelenSorular.Rows[i]["soru_resim"].ToString().Trim());
    }
}
}

```

Soru Ekleme Paneli:

Bu panelde soru eklemeleri yapılmaktadır. Öncelikle soru girilecek ders seçilir. Soru metni yazılır ve istenilirse soruya resim girilir. Şıklarda metin yada resim kullanılımları seçilir. Şıklar girilir ve doğru seçenek seçilir. Ardından sorulan sorunun zorluk derecesi belirlenir ve soru ekle butonuna basılır. Ekleme kodları aşağıdadır.

```

private void btnSoruEkle_Click(object sender, EventArgs e)
{
    if ((rdbA.Checked == true || rdbB.Checked == true || rdbC.Checked ==
true || rdbD.Checked == true || rdbE.Checked == true) && rtxtSoru.Text != "" &&
/*((txtA.Text != "" && txtB.Text != "" && txtC.Text != "" && txtD.Text != "")||
true) &&*/ cmbDersAdi.SelectedItem != null && cmbZorluk.SelectedItem != null)
{
    string soru = "", cevapA = "", cevapB = "", cevapC = "", cevapD =
"", cevapE = "", dogruCevap = "", zorlukDerecesi = "";
    Random rnd = new Random();
    string yeniisim = (rnd.Next(1, 9999999)).ToString();
    string yeniResimYolu;
    if (resimYolu == "")
    {
        yeniResimYolu = "";
    }
    else
    {
        yeniResimYolu = @"C:\DersResimleri\dersler\" +
cmbDersAdi.SelectedItem.ToString() + @"\\" + yeniisim + ".jpg";
    }
    soru = rtxtSoru.Text;
    if (rdbSikMetin.Checked == true)
    {
        cevapA = txtA.Text;
        cevapB = txtB.Text;
        cevapC = txtC.Text;
        cevapD = txtD.Text;
        cevapE = txtE.Text;
    }

    else if (rdbSikResim.Checked == true)
    {
        aYeniIsim = (rnd.Next(1, 9999999)).ToString();
        bYeniIsim = (rnd.Next(1, 9999999)).ToString();
        cYeniIsim = (rnd.Next(1, 9999999)).ToString();
        dYeniIsim = (rnd.Next(1, 9999999)).ToString();
        eYeniIsim = (rnd.Next(1, 9999999)).ToString();
        //File.Copy(resimYolu, yeniResimYolu);
        aYeniYol = @"C:\DersResimleri\dersler\" +
cmbDersAdi.SelectedItem.ToString() + @"\\" + aYeniIsim + ".jpg";
        bYeniYol = @"C:\DersResimleri\dersler\" +
cmbDersAdi.SelectedItem.ToString() + @"\\" + bYeniIsim + ".jpg";
        cYeniYol = @"C:\DersResimleri\dersler\" +
cmbDersAdi.SelectedItem.ToString() + @"\\" + cYeniIsim + ".jpg";
        dYeniYol = @"C:\DersResimleri\dersler\" +
cmbDersAdi.SelectedItem.ToString() + @"\\" + dYeniIsim + ".jpg";
        eYeniYol = @"C:\DersResimleri\dersler\" +
cmbDersAdi.SelectedItem.ToString() + @"\\" + eYeniIsim + ".jpg";
        cevapA = aYeniYol;
        cevapB = bYeniYol;
        cevapC = cYeniYol;
        cevapD = dYeniYol;
        cevapE = eYeniYol;
    }
}

```

```

    }

    dogruCevap = "";
    if (rdbA.Checked == true)
    {
        dogruCevap = cevapA;
    }
    if (rdbB.Checked == true)
    {
        dogruCevap = cevapB;
    }
    if (rdbC.Checked == true)
    {
        dogruCevap = cevapC;
    }
    if (rdbD.Checked == true)
    {
        dogruCevap = cevapD;
    }
    if (rdbE.Checked == true)
    {
        dogruCevap = cevapE;
    }
    zorlukDerecesi = cmbZorluk.SelectedItem.ToString();
    SqlCommand cmdSoruEkle = new SqlCommand();
    cmdSoruEkle.Connection = cnn;
    try
    {
        cmdSoruEkle.CommandText = "INSERT INTO Sorular
(ders_id,soru,soru_resim,a_cevap,b_cevap,c_cevap,d_cevap,e_cevap,dogru_cevap,zorluk_
derecesi) VALUES
(@ders_id,@soru,@soru_resim,@a_cevap,@b_cevap,@c_cevap,@d_cevap,@e_cevap,@dogru_ceva
p,@zorluk_derecesi)";
        cmdSoruEkle.Parameters.AddWithValue("@ders_id", dersID);
        cmdSoruEkle.Parameters.AddWithValue("@soru", soru);
        cmdSoruEkle.Parameters.AddWithValue("@soru_resim",
yeniResimYolu);
        if (rdbSikResim.Checked == true)
        {
            File.Copy(aResim, aYeniYol);
            File.Copy(bResim, bYeniYol);
            File.Copy(cResim, cYeniYol);
            File.Copy(dResim, dYeniYol);
            File.Copy(eResim, eYeniYol);
        }
        cmdSoruEkle.Parameters.AddWithValue("@a_cevap", cevapA);
        cmdSoruEkle.Parameters.AddWithValue("@b_cevap", cevapB);
        cmdSoruEkle.Parameters.AddWithValue("@c_cevap", cevapC);
        cmdSoruEkle.Parameters.AddWithValue("@d_cevap", cevapD);
        cmdSoruEkle.Parameters.AddWithValue("@e_cevap", cevapE);
        cmdSoruEkle.Parameters.AddWithValue("@dogru_cevap", dogruCevap);
        cmdSoruEkle.Parameters.AddWithValue("@zorluk_derecesi",
zorlukDerecesi);
        if (resimYolu == "")
        {
            yeniResimYolu = "";
        }
    }

```

```

else
{
    File.Copy(resimYolu, yeniResimYolu);
}
//MessageBox.Show(cevapE.ToString());
cmdSoruEkle.ExecuteNonQuery();
MessageBox.Show("Soru eklendi.");

rdbA.Checked = false;
rdbB.Checked = false;
rdbC.Checked = false;
rdbD.Checked = false;
rdbE.Checked = false;
cmbZorluk.SelectedItem = null;
txtA.Text = null;
txtB.Text = null;
txtC.Text = null;
txtD.Text = null;
txtE.Text = null;
rtxtSoru.Text = null;
resimYolu = "";
cmbZorluk.SelectedItem = null;
pcbSoruResmi.ImageLocation = @"resimsiz.jpg";
rtxtOnizleme.Text = null;
pcbOnizleme.Image = null;
pcbA.Image = null;
pcbB.Image = null;
pcbC.Image = null;
pcbD.Image = null;
pcbE.Image = null;
aResim = "";
bResim = "";
cResim = "";
dResim = "";
eResim = "";
}
catch (Exception hata)
{
    MessageBox.Show(hata.Message);
}
}
else
{
    MessageBox.Show("Lütfen yıldızlı(*) alanları boş bırakmayınız.");
}
}

```

Sonuç Okuma Formu:

Bu formda optik okuyucudan alınan not defteri belgesi form üzerindeki datagridwiev üzerine sürüklenip bırakılır. Cevap anahtarları tespit edildikten sonra öğrencilerin cevapları cevap anahtarları ile karşılaştırılır ve veriler hesaplanır.

Not defteri sürüklendikten sonra olan işlemlerin kodları aşağıdadır.

```
private void Form5_DragDrop(object sender, DragEventArgs e)
{
    grupSayisi = 0;
    lblACevaplar.Text = "...";
    lblBCevaplar.Text = "...";
    lblSoruPuani.Text = "...";
    dtSoruAnalizA.Rows.Clear();
    dtSoruAnalizB.Rows.Clear();
    dtAralik.Rows.Clear();
    dataGridView1.Rows.Clear();
    string[] veriler = (string[])e.Data.GetData(DataFormats.FileDrop,
false);
    StreamReader aktar = new StreamReader(veriler[0],
Encoding.GetEncoding("iso-8859-9"), false);
    string yazi;
    //int grupBul;
    int soruSayisi = 0;
    string aCevaplar = "", bCevaplar = "";
    //richTextBox1.Text = aktar.ReadToEnd();
    //int sayac = 0;
    string ogrNo, ogrAd, sinavTur, grup, ogCevaplar;
    int dogru = 0, yanlis = 0, bos = 0;
    double puan = 0;
    while ((yazi = aktar.ReadLine()) != null)
    {
        if (yazi.Substring(0, 5).ToString() == "00000")
        {
            if (yazi.Substring(10, 1).ToString() == "A")
            {
                //MessageBox.Show("A Grubu");
                aCevaplar = yazi.Substring(35, yazi.Length - 35).Trim();
                soruSayisi = aCevaplar.Length;
                soruPuani = 100.00 / soruSayisi;
                lblSoruPuani.Text = soruPuani.ToString();
                lblACevaplar.Text = aCevaplar;
                grupSayisi++;
                //MessageBox.Show("A gurubu cevapları bulundu");
            }
        }
        if (yazi.Substring(0, 5).ToString() == "00000")
        {
            if (yazi.Substring(10, 1).ToString() == "B")
            {
                //MessageBox.Show("B Grubu");
                bCevaplar = yazi.Substring(35, yazi.Length - 35).Trim();
                lblBCevaplar.Text = bCevaplar;
                //MessageBox.Show("B gurubu cevapları bulundu");
                grupSayisi++;
            }
        }
    }
}
```

```

    }
    MessageBox.Show("Sınavdaki grup sayısı=" + grupSayisi + ". Cevap
anahtarı tespit edildi");

    /**/
    if (grupSayisi == 1)
    {
        soruAnalizA = new int[soruSayisi];
        for (int i = 0; i < soruSayisi; i++)
        {
            soruAnalizA[i] = 0;
        }
    }
    if (grupSayisi == 2)
    {
        soruAnalizA = new int[soruSayisi];
        soruAnalizB = new int[soruSayisi];
        for (int i = 0; i < soruSayisi; i++)
        {
            soruAnalizA[i] = 0;
            soruAnalizB[i] = 0;
        }
    }
    /**/
    if (grupSayisi <= 1)
    {
        lblABaslik.Text = "A gurubundaki cevaplar=";
        lblBBaslik.Text = "...";
        lblBCevaplar.Text = "...";
    }
    else
    {
        lblABaslik.Text = "A gurubundaki cevaplar=";
        lblBBaslik.Text = "B gurubundaki cevaplar=";
    }
    string[] veriler2 = (string[])e.Data.GetData(DataFormats.FileDrop,
false);
    StreamReader aktar2 = new StreamReader(veriler[0],
Encoding.GetEncoding("iso-8859-9"), false);

    string yazi2;
    while ((yazi2 = aktar2.ReadLine()) != null)
    {
        //MessageBox.Show("While a girdi");
        if (yazi2.Substring(0, 5).ToString() != "00000")
        {
            //MessageBox.Show("ife girdi");
            ogrNo = yazi2.Substring(0, 8).ToString();
            sinavTur = yazi2.Substring(9, 1).ToString();
            if (sinavTur == "1")
            {
                sinavTur = "Vize";
            }
            else
            {
                sinavTur = "Final";
            }
        }
    }

```

```

grup = yazi2.Substring(10, 1).ToString();
ogrAd = yazi2.Substring(11, 25).ToString().Trim();
ogCevaplar = yazi2.Substring(36, soruSayisi);
//MessageBox.Show("Öğrencinin cevap sayısı=" +
ogCevaplar.Length.ToString());
/*MessageBox.Show(ogCevaplar.Length.ToString());
MessageBox.Show(aCevaplar.Length.ToString());
MessageBox.Show(bCevaplar.Length.ToString());*/
if (grup == "A")
{
    /*MessageBox.Show(ogrAd);
    MessageBox.Show(ogCevaplar.Length.ToString());*/
    for (int i = 0; i < aCevaplar.Length; i++)
    {

        if (ogCevaplar.Substring(i, 1) != " ")
        {
            if (ogCevaplar.Substring(i, 1) ==
aCevaplar.Substring(i, 1))
            {
                dogru++;
                soruAnalizA[i] += 1;
            }
            else
            {
                yanlis++;
            }
        }
        else
        {
            bos++;
        }
    }
    puan = Convert.ToDouble(dogru) * soruPuanı;
    notAraligi(puan);
    dataGridView1.Rows.Add(ogrNo, ogrAd, sinavTur, grup,
dogru.ToString(), yanlis.ToString(), bos.ToString(), puan.ToString());
    //richTextBox1.Text = richTextBox1.Text + ogCevaplar;
    dogru = 0;
    yanlis = 0;
    bos = 0;
    ogrAd = "";
    ogCevaplar = "";
    ogrNo = "";
    sinavTur = "";
    grup = "";
    puan = 0;
    //MessageBox.Show("Öğrenci Eklendi");
}
else
{
    for (int i = 0; i < bCevaplar.Length; i++)
    {
        //MessageBox.Show(i.ToString());
        if (ogCevaplar.Substring(i, 1) != " ")
        {
            if (ogCevaplar.Substring(i, 1) ==
bCevaplar.Substring(i, 1))

```



```

        {
            //MessageBox.Show(ogCevaplar.Substring(i, 1));

            dogru++;
            soruAnalizB[i] += 1;
        }
        else
        {
            yanlis++;
        }
    }
    else
    {
        bos++;
    }
}
puan = Convert.ToDouble(dogru) * soruPuanı;
notAraligi(puan);
dataGridView1.Rows.Add(ogrNo, ogrAd, sinavTur, grup,
dogru.ToString(), yanlis.ToString(), bos.ToString(), puan.ToString());
//richTextBox1.Text = richTextBox1.Text + ogCevaplar;
dogru = 0;
yanlis = 0;
bos = 0;
ogrAd = "";
ogCevaplar = "";
ogrNo = "";
sinavTur = "";
grup = "";
puan = 0;
//MessageBox.Show("Öğrenci Eklendi");
    }
}
}
//MessageBox.Show("A 1. soruyu doğru cevaplayan sayısı= " +
soruAnalizA[0].ToString());
//MessageBox.Show("B 1. soruyu doğru cevaplayan sayısı= " +
soruAnalizB[1].ToString());

int notlar = 0;
for (int i = 0; i < 20; i++)
{
    if (notlar == 95)
    {
        dtAralik.Rows.Add((notlar.ToString() + "-" + (notlar +
5).ToString()), aralikA[i]);
        notlar += 5;
    }
    else
    {
        dtAralik.Rows.Add((notlar.ToString() + "-" + (notlar +
4).ToString()), aralikA[i]);
        notlar += 5;
    }
}

}
if (grupSayisi == 1)
{

```

```

        for (int i = 0; i < soruSayisi; i++)
        {
            dtSoruAnalizA.Rows.Add((i + 1), soruAnalizA[i]);
        }
    }
    else
    {
        for (int i = 0; i < soruSayisi; i++)
        {
            dtSoruAnalizA.Rows.Add((i + 1), soruAnalizA[i]);
            dtSoruAnalizB.Rows.Add((i + 1), soruAnalizB[i]);
        }
    }
    /*for (int i = 0; i < dtSoruAnalizB.Rows.Count; i++)
    {
        MessageBox.Show(dtSoruAnalizB.Rows[i][1].ToString());
    }*/
    //MessageBox.Show(grupSayisi.ToString());
}

```

Excele aktar butonuna basıldığında aşağıdaki kodlar çalışır. Sayfalar oluşturulur ve veriler girilir. Sonra girilen verilerle ilgili grafikler oluşturulur.

```

private void btnExcelAktar_Click(object sender, EventArgs e)
{
    //MessageBox.Show(grupSayisi.ToString());
    if (dataGridView1.Rows.Count >= 2)
    {
        Excel.Application excelDosya = new Excel.Application();

        object Missing = Type.Missing;
        Workbook calismakitabi = excelDosya.Workbooks.Add(Missing);
        Worksheet sheet1 = (Worksheet)calismakitabi.Sheets[1];
        sheet1.Name = "Bireysel Notlar";

        Worksheet sheet2 = (Worksheet)calismakitabi.Sheets.Add(Missing,
Missing, 1, Missing) as Excel.Worksheet; ;
        sheet2.Name = "Soru Analizi";

        Worksheet sheet3 = (Worksheet)calismakitabi.Sheets.Add(Missing,
Missing, 1, Missing) as Excel.Worksheet; ;
        sheet3.Name = "Not Aralıkları";
        Excel.Range formatRange;

        formatRange = sheet1.get_Range("h:h");
        formatRange.NumberFormat = "@";

        Excel.Range formatRange2;

        formatRange2 = sheet3.get_Range("a:a");
        formatRange2.NumberFormat = "@";
        //sheet1.Cells[8, 2].NumberFormat = "General";
        int sutun = 1;
    }
}

```

```

int satir = 1;

for (int j = 0; j < dataGridView1.Columns.Count; j++)
{
    Range myrange = (Range)sheet1.Cells[satir, sutun + j];
    myrange.Value2 = dataGridView1.Columns[j].HeaderText;
}
satir++;
for (int i = 0; i < dataGridView1.Rows.Count; i++)
{
    for (int j = 0; j < dataGridView1.Columns.Count; j++)
    {
        Range myrange = (Range)sheet1.Cells[satir + i, sutun + j];
        myrange.Value2 = dataGridView1[j, i].ValueType == null ? ""
: dataGridView1[j, i].Value;
        //myrange.Select();
    }
}
//soru analiz

if (grupSayisi == 1)
{
    int analizSatir = 1, analizSutun = 1;

    for (int i = 0; i < dtSoruAnalizA.Rows.Count; i++)
    {
        for (int j = 0; j < dtSoruAnalizA.Columns.Count; j++)
        {
            Range myrange2 = (Range)sheet2.Cells[analizSatir + i,
analizSutun + j];
            myrange2.Value2 = dtSoruAnalizA.Rows[i][j] == null ? ""
: dtSoruAnalizA.Rows[i][j];
            //myrange2.Select();
        }
    }
}
if (grupSayisi == 2)
{
    //MessageBox.Show("Else çalışıyor");
    int analizSatirA = 1, analizSutunA = 1, analizSatirB = 1,
analizSutunB = 4;

    for (int i = 0; i < dtSoruAnalizA.Rows.Count; i++)
    {
        for (int j = 0; j < dtSoruAnalizA.Columns.Count; j++)
        {
            Range myrange2 = (Range)sheet2.Cells[analizSatirA + i,
analizSutunA + j];
            myrange2.Value2 = dtSoruAnalizA.Rows[i][j] == null ? ""
: dtSoruAnalizA.Rows[i][j];
            //myrange2.Select();
        }
    }

    for (int i = 0; i < dtSoruAnalizB.Rows.Count; i++)
    {

```

```

        for (int j = 0; j < dtSoruAnalizB.Columns.Count; j++)
        {
            Range myrange2 = (Range)sheet2.Cells[analizSatirB + i,
analizSutunB + j];
            myrange2.Value2 = dtSoruAnalizB.Rows[i][j] == null ? ""
: dtSoruAnalizB.Rows[i][j];
            //myrange2.Select();
        }
    }

    //Not aralıkları
    int aralikSatir = 1, araliksutun = 1;
    for (int i = 0; i < dtAralik.Rows.Count; i++)
    {
        for (int j = 0; j < dtAralik.Columns.Count; j++)
        {
            Range myrange3 = (Range)sheet3.Cells[aralikSatir + i,
araliksutun + j];
            myrange3.Value2 = dtAralik.Rows[i][j] == null ? "" :
dtAralik.Rows[i][j];
            //myrange2.Select();
        }
    }

    //Grafikler

    if (grupSayisi == 1)
    {
        Excel.Range chartRange;

        Excel.ChartObjects xlCharts =
(Excel.ChartObjects)sheet2.ChartObjects(Type.Missing);
        Excel.ChartObject myChart = (Excel.ChartObject)xlCharts.Add(50,
20, 800, 250);
        Excel.Chart chartPage = myChart.Chart;

        chartRange = sheet2.get_Range("B:A");
        chartPage.SetSourceData(chartRange, Missing);
        chartPage.ChartType = Excel.XlChartType.xlColumnClustered;

        excelDosya.Visible = true;
    }
    else
    {
        Excel.Range chartRange;

        Excel.ChartObjects xlCharts =
(Excel.ChartObjects)sheet2.ChartObjects(Type.Missing);
        Excel.ChartObject myChart = (Excel.ChartObject)xlCharts.Add(50,
20, 800, 250);
        Excel.Chart chartPage = myChart.Chart;
    }

```

```

        chartRange = sheet2.get_Range("B:A");
        chartPage.SetSourceData(chartRange, Missing);
        chartPage.ChartType = Excel.XlChartType.xlColumnClustered;

        Excel.Range chartRange2;

        Excel.ChartObjects xlCharts2 =
(Excel.ChartObjects)sheet2.ChartObjects(Type.Missing);
        Excel.ChartObject myChart2 =
(Excel.ChartObject)xlCharts2.Add(50, 240, 800, 250);
        Excel.Chart chartPage2 = myChart2.Chart;

        chartRange2 = sheet2.get_Range("D:E");
        chartPage2.SetSourceData(chartRange2, Missing);
        chartPage2.ChartType = Excel.XlChartType.xlColumnClustered;

        excelDosya.Visible = true;
    }

    //Not Aralıkları

    Excel.Range chartRange3;

    Excel.ChartObjects xlCharts3 =
(Excel.ChartObjects)sheet3.ChartObjects(Type.Missing);
    Excel.ChartObject myChart3 = (Excel.ChartObject)xlCharts3.Add(50,
80, 800, 250);
    Excel.Chart chartPage3 = myChart3.Chart;

    chartRange3 = sheet3.get_Range("A:B");
    chartPage3.SetSourceData(chartRange3, Missing);
    chartPage3.ChartType = Excel.XlChartType.xlColumnClustered;

    excelDosya.Visible = true;
}
else
{
    MessageBox.Show("Aktarmaya uygun veri bulunamadı.")

```

BÖLÜM V

SONUÇ ve ÖNERİLER

Projenin ilk aşamalarında sorunlar belirlenmiştir. Sınav oluşturma süreci incelenmiş, hata yapılmaya müsait alanlar tespit edilmiş ve bu alanlara yoğunlaşmıştır. Sorulara resim ekleme ve cevap şıklarına resim ekleme gibi istenebilecek tüm ihtimalleri barındırmaktadır.

Proje tüm amaçlarına ulaşmıştır. Aktif şekilde kullanıma hazır durumdadır. İstenilen sınav 5 şıklı test soruları halinde gerçekleştirilebilir ve sonuçları okunup değerlendirmeler yapılabilir.

Proje ilerde geliştirilmek istenirse bazı eklemeler yapılabilir. Örneğin sorular için düzenleme seçeneği eklenebilir. Sınav istenilirse dört şıklı, istenilirse beş şıklı olarak ayarlanabilir. Yeni kullanıcı ekleme seçeneği getirilebilir.

BÖLÜM VI

KAYNAKLAR

[1] Murat Yücedağ – c# Word işlemleri Ders 1

<https://www.youtube.com/watch?v=JhwEKOeciUc&t=214s>

[2] Murat Yücedağ – C# Word işlemleri Ders 2

<https://www.youtube.com/watch?v=QvOgflWlfGw&t=1147s>

[3] Murat Yücedağ – C# Excel işlemleri Ders 1

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

[4] Murat Yücedağ – C# Excel işlemleri Ders 2

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

[5] Excel ile Grafik Oluşturma

<http://www.ahmetcansever.com/programlama/c-chart-kullanimi-ve-grafik-olusturma-ornegi/>

BÖLÜM VII

ÖZGEÇMİŞ

Burak Boz

Ben Burak Boz. Marmara Üniversitesi Bilgisayar Programcılığı 2. Sınıf öğrencisiyim. Liseyi Ümraniye Anadolu Meslek Lisesinde okudum. Lisemdeki alanımda Veritabanı programcılığıydı. Küçüklüğümünden beri aktif şekilde bilgisayar kullanmaktayım ve yazılıma meraklıyım. Bilgisayar üzerinde sistemler kurmak, bazende robotikle ilgili projeler yapmak hoşuma gider. Daha önce Akıllı akvaryum projesi, mobil Word uygulaması ve bazı web siteleri projelerinde görev aldım Okulum bittikten sonra lisans tamamlama yapacağım ve bilgisayar mühendisi olmaya çalışacağım. Bu sektörde olduğum için mutluyum.