



Düzce Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Veritabanı Yönetim Sistemleri
Öğrenci Bilgi Sistemi Projesi Raporu
N.Ö. Grup 7

Teslim Eden:

Numarası	Adı Soyadı
191001009	Aleyna GENÇ
191001057	Gülçin APAYDIN
201001038	Ahsen Nisa AKOVA
211001301	Meliha DEMİRCİ

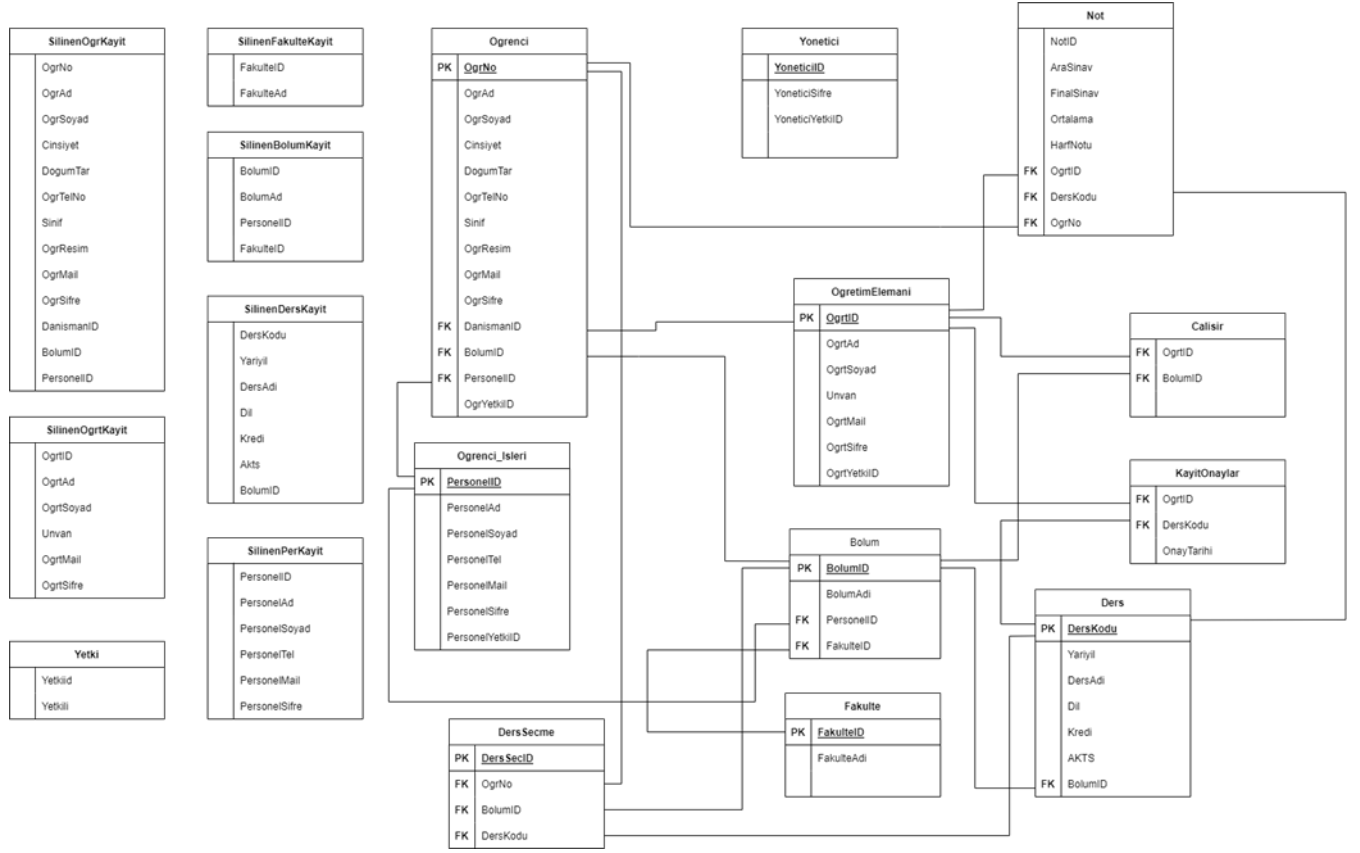
İÇİNDELİKER

1-ER DİYAGRAMLARININ OLUŞTURULMASI	3
2-VERİTABANI TABLOLARININ OLUŞTURULMASI.....	4
3-TABLOLAR	5
4-TRIGGERLAR	7
5-RAPORLAR	12
6-XLS UZANTILI DOSYA ŞEKLİNDE KAYIT ALINMASI	16
7-UYGULAMA GİRİŞ EKRANI.....	17
8-KULLANICI TANIMLAMA VE YETKİLENDİRME	18
8.1-VERİ OKUMA-SİLME-GÜNCELLEME-EKLEME İŞLEMLERİ	20
9-IMPORT/EXPORT	21
9.1 – IMPORT	21
9.2 - EXPORT	22
10-YEDEKLE/YEDEKTEN DÖN	23
11- UYGULAMA ARAYÜZÜ.....	24

Şekil 1 – ER diyagramları	3
Şekil 2 – Veritabanı tabloları	4
Şekil 3 – Not_ Tablosu.....	5
Şekil 4 – OgretimElemani Tablosu	5
Şekil 5 – Ders Tablosu	6
Şekil 6 – Öğrenci İşleri Tablosu.....	6
Şekil 7 – ogrsil Trigger.....	7
Şekil 8 – ogrtsil Trigger	7
Şekil 9 – derssil Trigger	7
Şekil 10 – bolumsil Trigger.....	8
Şekil 11 – persil Trigger.....	8
Şekil 12 – fakultesil Trigger.....	8
Şekil 13 – YetkiKoru Trigger.....	9
Şekil 14 – ogrEklendi Trigger.....	9
Şekil 15 – OgrSay Trigger	9
Şekil 16 – ogrtEklendi Trigger.....	10
Şekil 17 – OgrtSay Trigger	10
Şekil 18 - personelEklendi	10

Şekil 19 – personelSay Trigger	11
Şekil 20 – personelguncelle Trigger.....	11
Şekil 21 – secilmisdersler Procedure	12
Şekil 21.1 – Ders Seçme İşlemleri.....	12
Şekil 22 – OgrBilgi Procedure	13
Şekil 22.1 – Öğrenci Görüntüleme işlemleri	13
Şekil 23 – tbNotlar Procedure	14
Şekil 23.1 – Not Girişi İşlemleri.....	14
Şekil 24 – akademisyenbilgi Procedure	15
Şekil 24.1 – Öğretim Elemanı Görüntüleme İşlemleri	15
Şekil 25 – Excel Dosyası.....	16
Şekil 25.1 – Excel Dosyası Kodları.....	16
Şekil 26 – Uygulama Giriş Ekranı	17
Şekil 26.1 – Öğrenci Giriş Ekranı	17
Şekil 26.2 – Öğrenci Şifre Güncelleme Ekranı	18
Şekil 27 – Yönetici Giriş Ekranı	18
Şekil 27.1 – Yönetici Ana Menü Ekranı.....	19
Şekil 27.2 – Öğretim Elemanı Ana Menü Ekranı.....	19
Şekil 28 – Öğretim Elemanı Formu	20
Şekil 28.1 – Ders İşlemleri Formu.....	20
Şekil 29 – Import İşlemi.....	21
Şekil 29.1 – Import İşlemi	21
Şekil 29.2 – Import İşlemi	22
Şekil 30 – Export İşlemi.....	22
Şekil 31 – Yedekle ve Yedekten Dön İşlemi	23
Şekil 32 – Öğrenci Bilgileri	24

2-VERİTABANI TABLOLARININ OLUŞTURULMASI



Şekil 2 – Veritabanı tabloları

ER diyagramlarının veritabanı tablolarına dönüştürülmüş hali Şekil 2’de görülmektedir. Öğrenci tablosunda OgrNo primary key olarak belirlenmiş olup DanismanID, BolumID ve PersonelID ilgili tablolardan foreign key olarak getirilmiştir. Öğrenci işleri tablosunda PersonelID primary key olarak belirlenmiş olup herhangi bir foreign key’e sahip değildir. Fakülte tablosu yalnızca FakulteID primary key’ine sahiptir. DersSecme tablosu DersSecID adında bir primary key’e sahip olup OgrNo, BolumID ve DersKodu niteliklerini ilgili tablolardan foreign key olarak almıştır. Bölüm tablosu primary key olarak BolumID’ye sahip olup PersonelID ve FakulteID ilgili tablolardan foreign key olarak gelmiştir. Öğretim elemanı tablosu OgrtID adında primary key’e sahiptir. foreign key içermemektedir. Not tablosu NotID adında primary key içermektedir, OgrtID, DersKodu ve OgrNo foreign key olarak ilgili tablolardan gelmiştir. Ders tablosu DersKodu adında primary key içermektedir, BolumID foreign key olarak tabloda bulunmaktadır.

3-TABLolar

```
create table Not_  
NotID int primary key identity,  
vize int,  
final int,  
ortalama float,  
harfnotu char(2)  
    constraint ck_harfnotu  
    check(harfnotu like '[A-F][A-F]'),  
OgrtID char(10)  
    constraint fk_ogrtI  
    foreign key references OgretimElemani(OgrtID) on delete set null,  
DersKodu char(5)  
    constraint fk_derskoduu  
    foreign key references Ders(DersKodu) on delete cascade,  
OgrNo char(9)  
    constraint fk_ogrno  
    foreign key references Ogrenci(OgrNo) on delete cascade  
)
```

Şekil 3 – Not_ Tablosu

Şekil 3’de görüldüğü üzere Not_ tablosunu oluşturmak için gerekli Sql kodları yer almaktadır. Harf notu AA-FF arasında olacak şekilde constraint ile kural koyulmuş ve check ile kontrol edilmiştir. En uygun veri türü seçilmiş, ilgili foreign keyler uygun veri silme biçimi ile eklenmiştir.

```
CREATE TABLE OgretimElemani(  
OgrtID char(10) PRIMARY KEY,  
    constraint ck_OgrtID  
    check(OgrtID like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
OgrtAd varchar(50) not null,  
OgrtSoyad varchar(50) not null,  
Unvan varchar(50),  
OgrtMail varchar(60)  
    constraint ck_OgrtMail  
    check(OgrtMail like '%@duzce.edu.tr'),  
OgrtSifre varchar(15)  
)  
  
alter table OgretimElemani add OgrtYetkiID char(1) constraint ck_oyetki check(OgrtYetkiID in ('2'))  
constraint df_ogrtyetki default('2')
```

Şekil 4 – OgretimElemani Tablosu

Şekil 4’te Öğretim Elemanı tablosunu Sql’e eklemek için gerekli kod dizisi görülmektedir. Uygun veri türleri seçilmiştir. Kod dizisi içerisinde OgrtID’nin 10 karakterli ve karakterlerin hepsi yalnızca 0-9 arasında değer alabilecek şekilde check - constraint oluşturulmuştur. OgrtMail ise sonunun mutlaka “@duzce.edu.tr” ile bitmesi için yine check - constraint eklenmiştir.

```

create table Ders(
DersKodu char(5) primary key
  constraint ck_derskodu
  check(derskodu like '[A-Z][A-Z][0-9][0-9][0-9]'),
Yariyil char(1)
  constraint ck_dersyariyil
  check(yariyil like '[0-9]'),
DersAdi varchar(100),
Dil varchar(15)
  constraint ck_dersdil
  check(dil in ('Türkçe','İngilizce')),
Kredi float,
Akts float,
BolumID int
  constraint fk_bolum
  foreign key references Bolum(BolumID) on delete cascade
)

```

Şekil 5 – Ders Tablosu

Şekil 5’te Ders tablosunu Sql’e eklemek için gerekli kod dizisi görülmektedir. Uygun veri türleri seçilmiştir. Kod dizisinin içinde DersKodu’nun 5 karakterli ve ilk iki karakterin A-Z arasında bir harf olması, geri kalan üç karakterin 0-9 arasında olması şeklinde constraint ile kural tanımlanmış check ile kontrol edilmiştir. Dil kısmında Türkçe ve İngilizce olması halinde sadece iki seçeneğe izin verilecek şekilde constraint – check işlemi yapılmıştır. İlgili foreign keyler uygun veri silme biçimi ile eklenmiştir.

```

CREATE TABLE OgrenciIsleri(
PersonelID int identity(1,1) Primary Key,
PersonelAd varchar (30) not null,
PersonelSoyad varchar (50) not null,
PersonelTel char(10)
  constraint ck_personelTEL
  check(PersonelTel like '[5][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
PersonelMail varchar(60)
  constraint ck_permail
  check (PersonelMail like '_%@duzce.edu.tr'),
PersonelSifre varchar(15),
)
alter table OgrenciIsleri add PersonelYetkiID char(1) constraint ck_pyetki check(PersonelYetkiID in ('3'))
constraint df_personelyetki default('3')

```

Şekil 6 – Öğrenci İşleri Tablosu

Şekil 6’da Öğrenci işleri tablosunu Sql’e eklemek için gerekli kod dizisi görülmektedir. Uygun veri türleri seçilmiştir. Kod dizisinin içinde PersonelTel’in 10 karakterli olması ve ilk karakterin ‘5’ sonraki her bir karakterin 0-9 arasında olması şeklinde bir constraint – check işlemi yapılmıştır. PersonelMail ise sonunun mutlaka “@duzce.edu.tr” ile bitmesi için yine check - constraint eklenmiştir. PersonelYetkiID default girildiği takdirde 3 yazılması sağlanmıştır.

4-TRIGGERLAR

```
create trigger ogrsil
on Ogrenci
after delete
as
insert into dbo.SilinenOgrKayit(OgrNo, OgrAd, OgrSoyad, cinsiyet, DogumTar, OgrTelNo, Sinif,
OgrResim, OgrMail, OgrSifre, DanismanID, BolumID, PersonelID)
select d.OgrNo, d.OgrAd, d.OgrSoyad, d.Cinsiyet, d.DogumTar, d.OgrTelNo, d.Sinif,
d.OgrResim, d.OgrMail, d.OgrSifre, d.DanismanID, d.BolumID, d.PersonelID
from deleted d
```

Şekil 7 – ogrsil Trigger

Şekil 7’de bulunan ogrsil isimli trigger, Ogrenci tablosunda bulunan bir öğrencinin kaydının silinmesi durumunda belirtilen ve deleted(silinen) tablosunda tutulan öğrenci verilerini SilinenOgrKayit adlı tabloya eklemektedir. Her öğrenci silme eyleminde otomatik olarak bu işlemi gerçekleştirmektedir.

```
create trigger ogrtsil
on OgretimElemani
after delete
as
insert into dbo.SilinenOgrtKayit(OgrtID, OgrtAd, OgrtSoyad, Unvan, OgrtMail, Ogrtsifre)
select d.OgrtID, d.OgrtAd, d.OgrtSoyad, d.Unvan,d.OgrtMail, d.Ogrtsifre
from deleted d
```

Şekil 8 – ogrtsil Trigger

Şekil 8’de bulunan ogrtsil isimli trigger, OgretimElemani tablosunda bulunan her bir öğretim elemanının kaydının silinmesi durumunda belirtilen ve deleted tablosunda tutulan öğretim elemanı verilerini SilinenOgrtKayit adlı tabloya otomatik olarak eklemektedir.

```
create trigger derssil
on Ders
after delete
as
insert into dbo.SilinenDersKayit(DersKodu, Yariyil, DersAdi, Dil, Kredi, Akts, BolumID)
select d.DersKodu, d.Yariyil, d.DersAdi, d.Dil,d.Kredi, d.Akts,d.BolumID
from deleted d
```

Şekil 9 – derssil Trigger

Şekil 9’da bulunan derssil isimli trigger, Ders tablosunda bulunan her bir dersin kaydının silinmesi durumunda belirtilen ve deleted tablosunda tutulan ders verilerini SilinenDersKayit adlı tabloya otomatik olarak eklemektedir.


```
create trigger bolumsil
on Bolum
after delete
as
insert into dbo.SilinenBolumKayit(BolumID,BolumAd,PersonelID,FakulteID)
select d.BolumID, d.BolumAd, d.PersonelID, d.FakulteID
from deleted d
```

Şekil 10 – bolumsil Trigger

Şekil 10’da bulunan bolumsil isimli trigger, Bolum tablosunda bulunan her bir bölümün kaydının silinmesi durumunda belirtilen ve deleted tablosunda tutulan bölüm verilerini SilinenBolumKayit adlı tabloya otomatik olarak eklemektedir.

```
create trigger persil
on OgrenciIsleri
after delete
as
insert into dbo.SilinenPerKayit(PersonelID,PersonelAd,PersonelSoyad,
PersonelTel,PersonelMail,PersonelSifre)
select d.PersonelID,d.PersonelAd,d.PersonelSoyad,
d.PersonelTel,d.PersonelMail,d.PersonelSifre
from deleted d
```

Şekil 11 – persil Trigger

Şekil 11’de bulunan persil isimli trigger, OgrenciIsleri tablosunda bulunan her bir öğrenci işleri personelinin kaydının silinmesi durumunda belirtilen ve deleted tablosunda tutulan personel verilerini SilinenPerKayit adlı tabloya otomatik olarak eklemektedir.

```
create trigger fakultesil
on Fakulte
after delete
as
insert into dbo.SilinenFakulteKayit(FakulteID,FakulteAd)
select d.FakulteID,d.FakulteAd
from deleted d
```

Şekil 12 – fakultesil Trigger

Şekil 12’de bulunan fakultesil isimli trigger, Fakulte tablosunda bulunan her bir fakültenin kaydının silinmesi durumunda belirtilen ve deleted tablosunda tutulan fakülte verilerini SilinenFakulteKayit adlı tabloya otomatik olarak eklemektedir.

```

create trigger YetkiKoru
on Yetki
instead of delete
as
begin
print('Yetki Silinemez')
rollback
end

```

Şekil 13 – YetkiKoru Trigger

Şekil 13'te bulunan YetkiKoru isimli trigger, Yetki tablosunda bir silme işlemi olması durumunda “Yetki Silinemez” uyarısı vererek yapılan silme işlemi rollback komutu ile geri alır.

```

create trigger ogrEklendi
on Ogrenci
for insert
as
declare @OgrAd varchar(30)
declare @OgrSoyad varchar(50)
begin
select @OgrAd=(select OgrAd from inserted)
select @OgrSoyad=(select OgrSoyad from inserted)
print(@OgrAd + ' '+@OgrSoyad+' isimli öğrenci sisteme kaydedildi')
end

```

Şekil 14 – ogrEklendi Trigger

Şekil 14'te bulunan ogrEklendi isimli trigger, Ogrenci tablosuna yeni bir öğrenci eklenmesi durumunda, öğrencinin ismi ve soyadı ile birlikte “isimli öğrenci sisteme kaydedildi” mesajını çıktı olarak üretir.

```

create trigger OgrSay
on Ogrenci
for insert
as
declare @say varchar
begin
select @say= count(*) from Ogrenci
print('Sistemde '+@say+' Öğrenci kaydı var')
end

```

Şekil 15 – OgrSay Trigger

Şekil 15'te bulunan OgrSay isimli trigger, Ogrenci tablosuna yeni bir öğrenci eklenmesi durumunda, Ogrenci tablosundaki tüm öğrencileri count(*) ile sayarak elde edilen öğrenci kaydı sayısını çıktı olarak yazar.

```

create trigger ogrtEklendi
on OgretimElemani
for insert
as
declare @OgrtAd varchar(30),@OgrtSoyad varchar(50)
begin
select @OgrtAd=(select OgrtAd from inserted)
select @OgrtSoyad=(select OgrtSoyad from inserted)
print(@OgrtAd + ' '+@OgrtSoyad+' isimli Ogretim Elemanı sisteme kaydedildi')
end

```

Şekil 16 – ogrtEklendi Trigger

Şekil 16’da bulunan ogrtEklendi isimli trigger, OgretimElemani tablosuna yeni bir öğretim elemanı eklenmesi durumunda, öğretim elemanının ismi ve soyadı ile birlikte “isimli Öğretim Elemanı sisteme kaydedildi” mesajını çıktı olarak üretir.

```

create trigger OgrtSay
on OgretimElemani
for insert
as
declare @sayoe varchar
begin
select @sayoe= count(*) from OgretimElemani
print('Sistemde '+@sayoe+' Öğretim Elemanı kaydı var')
end

```

Şekil 17 – OgrtSay Trigger

Şekil 17’de bulunan OgrtSay isimli trigger, OgretimElemani tablosuna yeni bir öğretim elemanı eklenmesi durumunda, OgretimElemani tablosundaki tüm öğretim elemanlarını count(*) ile sayarak elde edilen öğrenci kaydı sayısını çıktı olarak yazar.

```

create trigger personelEklendi
on OgrenciIsleri
for insert
as
declare @PersonelAd varchar(30), @PersonelSoyad varchar(50)
begin
select @PersonelAd=(select PersonelAd from inserted)
select @PersonelSoyad=(select PersonelSoyad from inserted)
print(@PersonelAd + ' '+@PersonelSoyad+' isimli Personel sisteme kaydedildi')
end

```

Şekil 18 - personelEklendi

Şekil 18’de bulunan personelEklendi isimli trigger, OgreciIsleri tablosuna yeni bir personel eklenmesi durumunda, personelin ismi ve soyadı ile birlikte “isimli Personel sisteme kaydedildi” mesajını çıktı olarak üretir.

```
create trigger personelSay
on OgreciIsleri
for insert
as
declare @sayp varchar
begin
select @sayp= count(*) from OgreciIsleri
print('Sistemde '+@sayp+' Personel kaydı var')
end
```

Şekil 19 – personelSay Trigger

Şekil 19’da bulunan personelSay isimli trigger, OgreciIsleri tablosuna yeni bir personel eklenmesi durumunda, OgreciIsleri tablosundaki tüm personelleri count(*) ile sayarak elde edilen personel kaydı sayısını çıktı olarak yazar.

```
create trigger personelguncelle
on dbo.Bolum
for update
as
begin
if update (PersonelID)
update dbo.Ogreci
set PersonelID=inserted.PersonelID
from inserted
inner join dbo.Ogreci on inserted.BolumID=Ogreci.BolumID
where inserted.PersonelID <> Ogreci.PersonelID
end
```

Şekil 20 – personelguncelle Trigger

Şekil 20’de bulunan personelguncelle isimli trigger, Bolum tablosunun PersonelID kolonu üzerinde güncelleştirme olduğunda, güncellenen PersonelID’nin BolumID’si ile aynı BolumID’ye sahip olan Ogreci tablosunun PersonelID’si de güncellenir. Yani örneğin BolumID’si 1 olan bölümün Personeli değişirse bu değişim Ogreci tablosunda sadece BolumID’si 1 olan öğrencilerin PersonelID’sini etkileyecektir.

5-RAPORLAR

```
USE [ObsDb]
GO
/***** Object: StoredProcedure [dbo].[secilmisdersler]    Script Date: 11.01.2023 23:40:09 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[secilmisdersler]
as
select DersSecID,OgrAd,OgrSoyad,Yariyil,DersAdi,Akts,Kredi,BolumAd from DersSecme
inner join Ogrnci
on Ogrnci.OgrNo=DersSecme.OgrNo
inner join Ders
on DersSecme.DersKodu=Ders.DersKodu
inner join Bolum
on Bolum.BolumID=DersSecme.BolumID
```

Şekil 21 – secilmisdersler Procedure

DÜZCE ÜNİVERSİTESİ Ders Seçme İşlemleri

Bölüm Seçiniz: Bilgisayar Mühend. Ders Seçiniz: Algoritma

Öğrenci No: 191001057
Öğrenci Ad: Gülçin
Öğrenci Soyad: Apaydın
Bölüm ID: 1
Ders Kodu: BM108
Yarıyıl: 1
Web Programlama
Türkçe
AKTS: 5

DersKodu	Yarıyıl	DersAdi	Dil	Kredi	Akts	Bolum
BM104	1	Algoritma	Türkçe	3	5	1
BM105	1	Veritabanı Yönetim	Türkçe	3	5	1
BM108	1	Web Programlama	Türkçe	3	5	1
EE105	1	EDT	Türkçe	3	5	2

DersSecID	OgrAd	OgrSoyad	Yariyil	DersAdi	Akts	Kr
5	Aleyna	Genç	1	Algoritma	5	3
6	Aleyna	Genç	1	Veritabanı Yönetim	5	3
7	Meliha	Demirci	1	Algoritma	5	3
8	Meliha	Demirci	1	Veritabanı Yönetim	5	3
9	Gülçin	Apaydın	1	EDT	5	3
10	Gülçin	Apaydın	1	Web Programlama	5	3
11	Ahaen	Akova	1	Veritabanı Yönetim	5	3

DERS EKLE
DERS KALDIR
LİSTELE

Şekil 21.1 – Ders Seçme İşlemleri

Şekil 21’de seçilmiş dersler store prosedürü Ders seçme, Öğrenci, Ders, Bolum tablolarının verilerini birleştirilmesiyle oluşturulmuştur. Öğrenci seçtiği dersleri daha detaylı bir şekilde görebilmesi için ders seçme işlemlerinde kullanılmıştır. Aynı zamanda not girme işlemlerinde öğretim elamanı için hangi öğrencinin hangi dersleri olduğu bilgisi de verilmektedir.

Şekil 21.1’de görüldüğü üzere ders seçme işleminde öğrenci numarasını girdiğinde ara butonuna basarak kendi isim ve soy ismi görüntülenmesi sağlandı. Yukarıda bulunan dersler tablosundan listeleyerek istediği derse çift tıklayıp ekranda bulunan textboxlara gelen veriler ekle butonuna basarak eklendi, secilmisdersler store prosedür ile oluşturulmuş seçilmiş dersler tablosunda listele butonuna tıklayarak eklenen ders detaylı bir şekilde gösterildi, dilerse seçtiği derse silebilmesi sağlandı. Ders seçme işleminde bölüm adı seçerek veya ders seçerek filtreleme özelliği eklendi.

```

USE [ObsDb]
GO
/***** Object: StoredProcedure [dbo].[OgrBilgi]    Script Date: 12.01.2023 02:29:50 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[OgrBilgi]
AS
Select OgrNO,OgrAd,OgrSoyad, Cinsiyet,DogumTar,Sinif,
OgrMail,OgrTelNo,BolumAd,DanismanID,OgrResim from Ogresci
inner join Bolum
on Bolum.BolumID=Ogresci.BolumID
inner join OgresciIsleri
on OgresciIsleri.PersonelID=Ogresci.PersonelID
inner join OgretimElemani
on OgretimElemani.OgrtID=Ogresci.DanismanID

```

Şekil 22 – OgrBilgi Procedure

OgrNO	OgrAd	OgrSoyad	Cinsiyet	DogumTar	OgrTelNo	Sinif	OgrResim	OgrMail	OgrSifre	DanismanID
191001009	Aleyna	Genç	K	24.07.1999	5362841516	3		aleynagenc@ogr...	aleyna123	1234567890
191001057	Gülçin	Apaydın	K	8.01.2000	5362457484	4	C:\Users\LENOV...	gulcinapaydin@o...	123456	1234567890
201001038	Ahsen	Akova	K	8.01.2002	5341458574	3	C:\Users\LENOV...	ahsen@ogr.duzc...	ahsen123	1234567890
211001301	Melha	Demirci	K	11.01.2023	5362541458	3	C:\Users\LENOV...	melha@ogr.duzc...	melha123	1234567890

Şekil 22.1 – Öğrenci Görüntüleme işlemleri

Şekil 22’de akademisyenin listele butonuna basarak şekil 22.1’de görüldüğü üzere bütün öğrencileri görüntüleyebileceği ‘Ogrescibil_’ adında bir store prosedür işlemi kullanıldı. Ogrescibil_ store prosedürü Ogresci, Bolum, Personel ve OgretimElemani tablolarının verilerinin birleştirilmesiyle oluşturulmuştur. Bu işlemi tekrar tekrar yazıp görüntülemek yerine store prosedür kullanarak akademisyenin tüm öğrenci verilerine detaylı bir şekilde erişebilmesine olanak sağlanmıştır.

Şekil 22.1’de öğrenci görüntüleme kısmında Öğrenci no, öğrenci ad, öğrenci soyad veya hem öğrenci ad hem soyada göre; sınıf, cinsiyet, danışmanın adına ve soyadına göre arama, listeleme işlemleri yapılmıştır.

```

USE [ObsDb]
GO
/***** Object: StoredProcedure [dbo].[tbNotlar]    Script Date: 11.01.2023 23:41:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[tbNotlar]
as
select NotID,OgrAd,OgrSoyad,DersAdi,vize,final,ortalama,harfnotu from Not_
inner join DersSecme
on Not_.OgrNo=DersSecme.OgrNo
inner join Ogrenci
on Ogrenci.OgrNo=Not_.OgrNo
inner join Ders
on Ders.DersKodu=DersSecme.DersKodu

```

Şekil 23 – tbNotlar Procedure

Ders Seçiniz

DersSecID	OgrAd	OgrSoyad	Yarıyıl	DersAdi	Akts	Kır
5	Aleyna	Genç	1	Algoritma	5	3
6	Aleyna	Genç	1	Veritabanı Yönetim	5	3
7	Melha	Demirci	1	Algoritma	5	3
8	Melha	Demirci	1	Veritabanı Yönetim	5	3
9	Gülçin	Apaydın	1	EDT	5	3
10	Gülçin	Apaydın	1	Web Programlama	5	3
11	Ahsen	Akova	1	Veritabanı Yönetim	5	3
12	Ahsen	Akova	1	Algoritma	5	3

Öğrenci No

Öğrenci Ad

Öğrenci Soyad

Not ID

Ders Kodu

Vize

Final

Ortalama

Harf Notu

HESAPLA **EKLE**
GÜNCELLE **LİSTELE**
SİL

NotID	Vize	final	ortalama	harfnotu	OgrtID
1	80	90	86	BA	
2	90	85	87	BA	

Şekil 23.1 – Not Giriş İşlemleri

Şekil 23'te TbNotlar store prosedürü Not, Ders seçme, Öğrenci, Ders tablolarının birleştirilmesiyle oluşturulmuştur. Öğretim elemanı bu tablo sayesinde girmiş olduğu derslerin notlarını Şekil 23.1'de görüldüğü üzere görüntüleyebilir, değiştirebilir veya silebilir. Not girme işlemleri için kullanılmıştır.

Şekil 23.1'de öğretim elemanı seçilmiş dersler tablosundaki öğrencilerin seçmiş olduğu dersleri görür ve ders notu girer. Seçilmiş dersler tablosundaki girmek istediği öğrencinin verisine ulaşabilir, çift tıkladığında öğrencinin adı, soyadı ders kodu textboxlara gelmektedir. Öğretim elemanı vize ve final notlarını girer, hesapla butonuna tıklayarak ders ortalamasını ve harf notunu hesaplar. Ekle butonuna basarak not girme işlemini

tamamlamış olur. Öğretim elemanı dilerse seçtiği öğrencinin ders notunu değiştirebilir. Store prosedür ile oluşturulmuş tbNotlar tablosundaki veriye çift tıkladığında seçili olan ders notlarının ID'si ve ders kodu bulunmaktadır. Öğretim elemanı değiştireceği ders notunu değiştirir, ortalama ve harf notunu hesaplar, günceller. Tekrardan listeleme tıkladığında not değerinin değiştiğini görür. Öğretim elemanı girdiği ders notlarını isterse silme butonuna basarak silebilir.

```
USE [ObsDb]
GO
/***** Object: StoredProcedure [dbo].[akademisyenbilgi]    Script Date: 12.01.2023 02:58:48 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[akademisyenbilgi]
AS
Select OgrtID,OgrtAd,OgrtSoyad, Unvan, OgrtMail from OgretimElemani
```

Şekil 24 – akademisyenbilgi Procedure

OgrtID	OgrtAd	Unvan	OgrtMail	OgrtSoyad
1234567890	Nihat	profesör	nihatdogan@duz...	Doğan

Şekil 24.1 – Öğretim Elemanı Görüntüleme İşlemleri

Öğretim elemanı görüntüleme işleminde personelin listele butonuna basmasıyla Şekil 24.1'de görüldüğü üzere tüm öğretim elemanlarının listelenmesini Şekil 24'te bulunan akademisyenbilgi adındaki store prosedür kullanılarak yapıldı. Akademisyen bilgilerinin detaylı bir şekilde görüntülenmesi sağlandı.

Şekil 24'te akademisyenin ID'sine, adına, soyadına, adına ve soyadına göre arama, listeleme işlemleri yapıldı.

6-XLS UZANTILI DOSYA ŞEKLİNDE KAYIT ALINMASI

The screenshot shows the 'Öğrenci Görüntüle' (Student Search) page of Düzce University. The page has a blue header with the university logo and name. Below the header, there are search filters for 'Öğrenci No', 'Öğrenci Ad ve Soyad', 'Danışman Ad ve Soyad', 'Öğrenci Adı', 'Sınıf', 'Öğrenci Soyadı', and 'Cinsiyet'. A 'LİSTELE' (List) button is on the right. Below the filters is a table of student records with columns: OgrNO, OgrAd, OgrSoyad, Cinsiyet, DogumTar, OgrTelNo, Sınıf, OgrResim, OgrMail, OgrSifre, and DanismanID. The table contains four rows of data. Below the table, there is an 'EXCEL' button. The bottom part of the screenshot shows a preview of the Excel file being generated, with the same data as the table above.

OgrNO	OgrAd	OgrSoyad	Cinsiyet	DogumTar	OgrTelNo	Sınıf	OgrResim	OgrMail	OgrSifre	DanismanID
191001009	Aleyna	Genç	K	24.07.1999	5362841516	3		aleynagenc@ogr...	aleyna123	1234567890
191001057	Gülçin	Apaydın	K	8.01.2000	5362457484	4	C:\Users\LENOV...	gulcinapaydin@o...	gulcin123	1234567890
201001038	Ahsen	Akova	K	8.01.2002	5341458574	3	C:\Users\LENOV...	ahsen@ogr.duzc...	ahsen123	1234567890
211001301	Melha	Demirci	K	11.01.2023	5362541458	3	C:\Users\LENOV...	melha@ogr.duzc...	melha123	1234567890

Şekil 25 – Excel Dosyası

```
private void buton1_Click(object sender, EventArgs e)
{
    if(dataGridView1.Rows.Count>0)
    {
        excel.Application app = new excel.Application();
        app.Visible = true;
        workbook kitap = app.Workbooks.Add(System.Reflection.Missing.Value);
        Worksheet sayfa = (Worksheet)kitap.Sheets[1];
        for (int i = 0; i < dataGridView1.Columns.Count; i++)
        {
            Range alan = (Range)sayfa.Cells[1, 1];
            alan.Cells[1, i + 1] = dataGridView1.Columns[i].HeaderText;
        }
        for (int i = 0; i < dataGridView1.Columns.Count; i++)
        {
            for (int j = 0; j < dataGridView1.Rows.Count; j++)
            {
                Range alan2 = (Range)sayfa.Cells[j + 1, i + 1];
                alan2.Cells[2, 1] = dataGridView1[i, j].Value;
            }
        }
    }
}
```

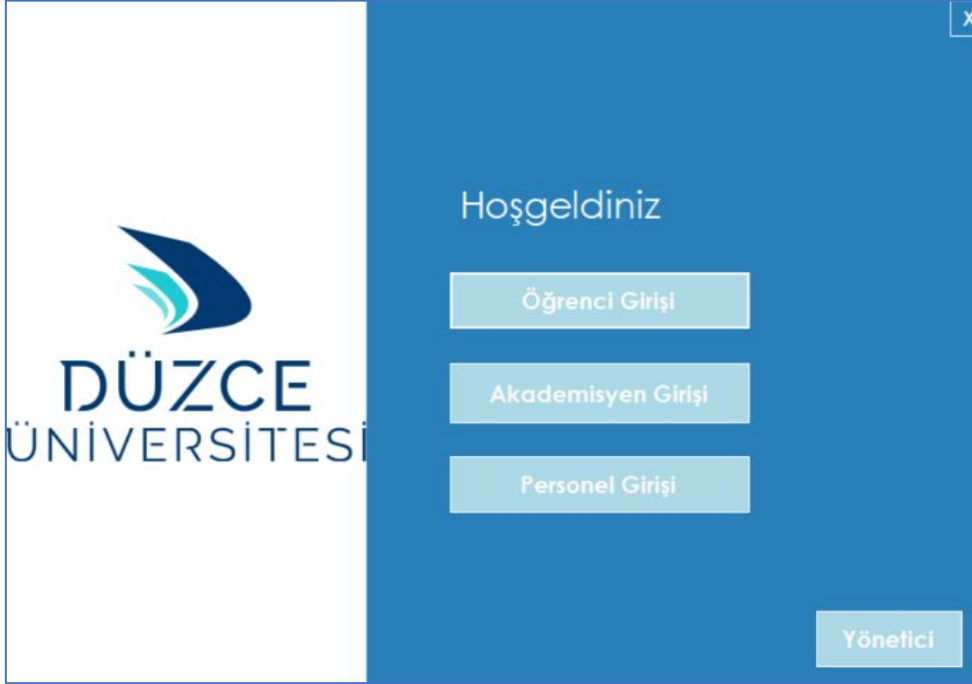
Şekil 25.1 – Excel Dosyası Kodları

Şekil 25'te görüldüğü üzere öğrenci görüntüle sayfasında bulunan 'EXCEL' butonu ile listelenen öğrenci tablosu raporu excel dosyası biçiminde kaydedilebilmektedir. Veritabanındaki istenen veri datagridview tablolarına aktarılır. Aktarılan veriler

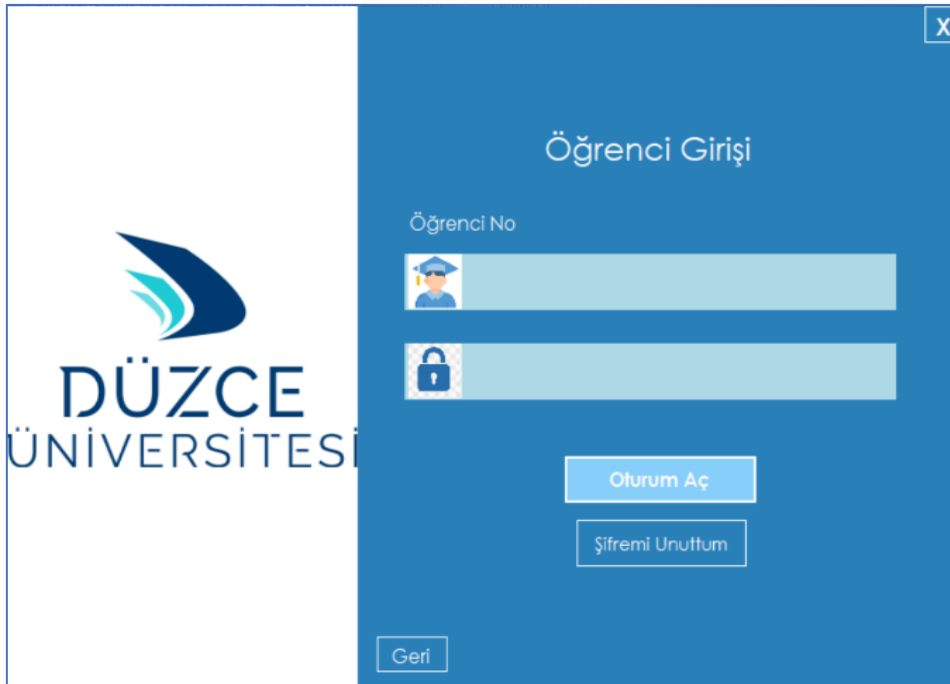
datagridview'in satırları ve sütunları boyunca excel bağlantısındaki sayfaya yazılır. Böylelikle verilerimiz excele kaydedilmiş olur.

Şekil 25.1'de bu işlemin kodları yer almaktadır.

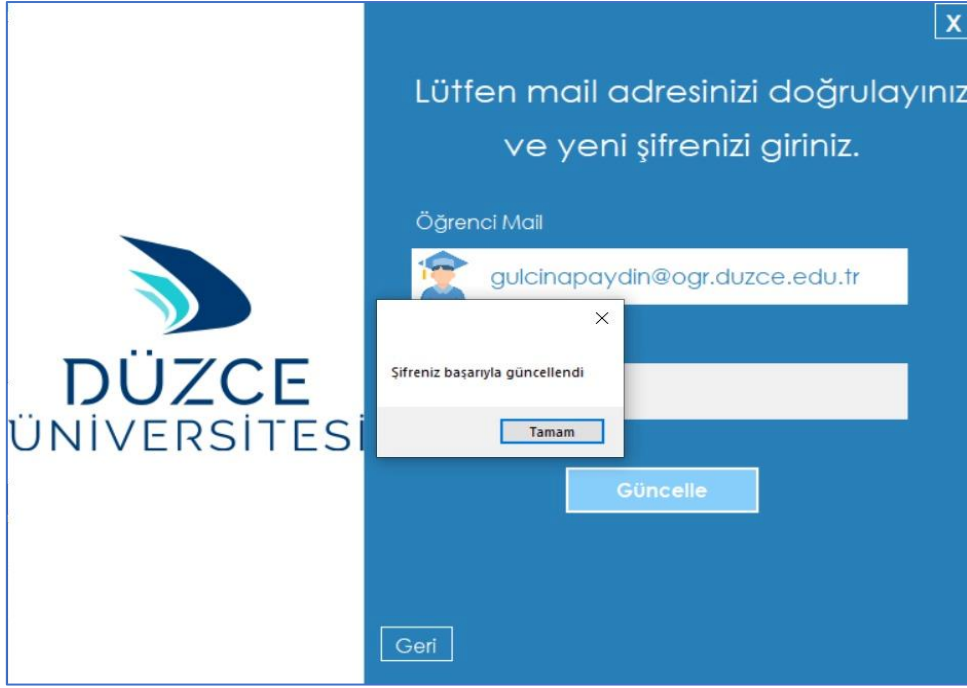
7-UYGULAMA GİRİŞ EKRANI



Şekil 26 – Uygulama Giriş Ekranı



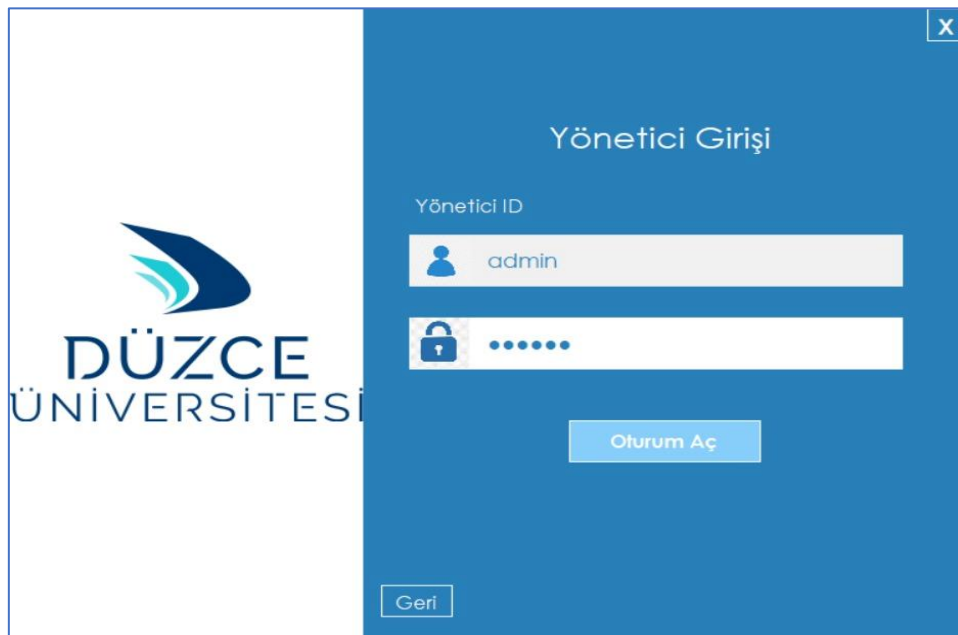
Şekil 26.1 – Öğrenci Giriş Ekranı



Şekil 26.2 – Öğrenci Şifre Güncelleme Ekranı

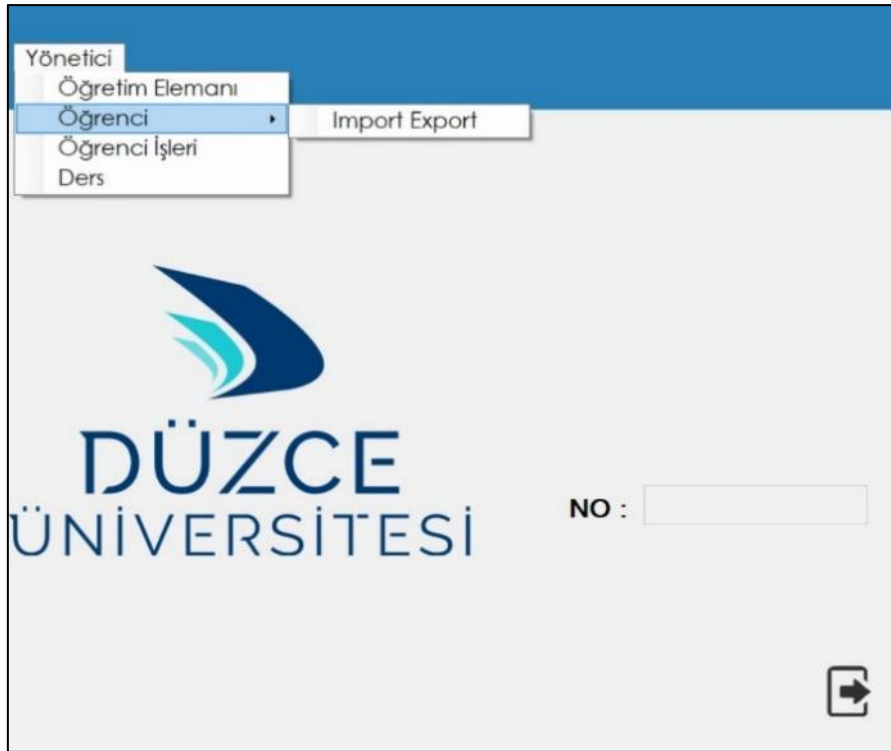
Öğrenci bilgi Sistemi projesinde Şekil 26'da görüldüğü üzere bir uygulama giriş ekranı tasarlandı. Şekil 26.1'de bulunan Öğrenci giriş ekranı başta olmak üzere, Akademisyen, Personel ve Yönetici adında üç farklı giriş ekranı daha tasarlanıp “kullanıcı adı ve şifre” sorusu ile güvenlik kontrolü yapıldı. Yönetici hariç her bir giriş ekranına şekil 26.2'de görüldüğü üzere ‘Şifremi Unuttum’ butonu ekleyerek olası bir şifre hatırlayamama durumlarında şifrelerini güncelleyebilmelerine olanak sağlandı.

8-KULLANICI TANIMLAMA VE YETKİLENDİRME

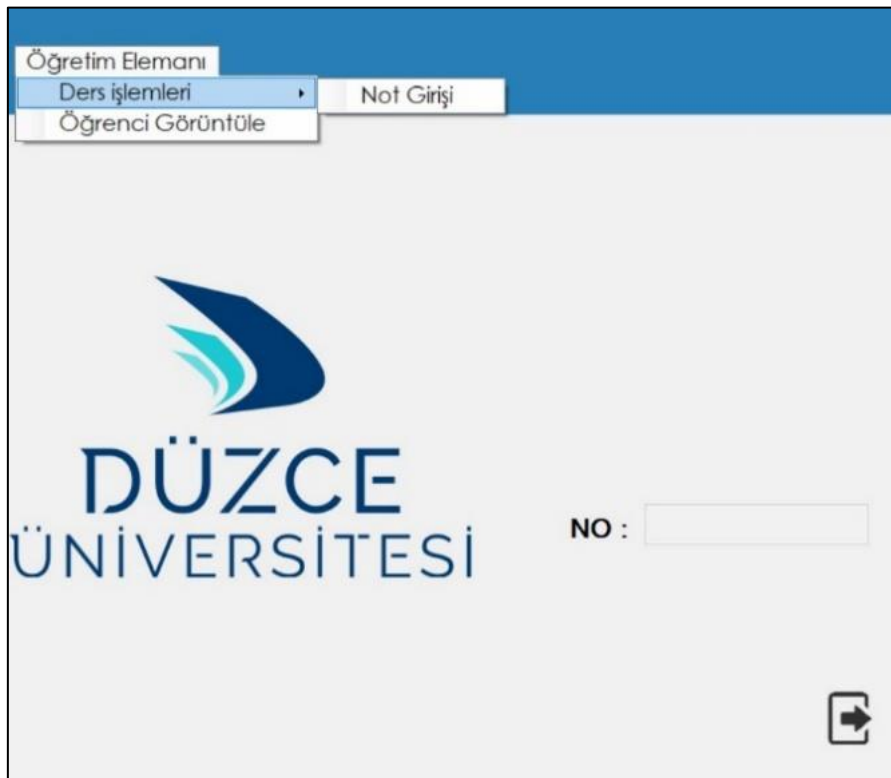


Şekil 27 – Yönetici Giriş Ekranı

Şekil 27’de görüldüğü üzere admin adında bir kullanıcı tanımlanmıştır.



Şekil 27.1 – Yönetici Ana Menü Ekranı



Şekil 27.2 – Öğretim Elemanı Ana Menü Ekranı

Sisteme giriş yapıldığında Öğrenci, Öğretim Elemanı, Personel ve Yöneticinin işlemlerini yapabilmesine olanak sağlayacak bir ana menü ekranı tasarlandı. Şekil 27.1 ve Şekil

27.2’de görüldüğü üzere tek bir menü ekranı üzerinden hangi giriş ekranından giriş yapıldıysa o kullanıcının yapabileceği işlemler görüntülenebildi, bu şekilde bir yetkilendirme işlemi yaparak öğretim elemanının yöneticinin işlemlerine erişebilmesinin önüne geçildi. Örneğin öğretim elemanı ekranından giriş yapıldıysa yetki ID’sine 2 vererek sadece öğretim elemanının yapabileceği işleri görüntülemesi sağlandı.

8.1-VERİ OKUMA-SİLME-GÜNCELLEME-EKLEME İŞLEMLERİ

Öğretim Elemanı Kayıt Formu

Akademisyen ID: 1234567892

Akademisyen Adı: Filiz

Akademisyen Soyadı: Akın

Unvan: profesör

Akademisyen Mail: filizakin@duzce.edu.

Akademisyen Şifre:

Akademisyen Yetki ID: *

EKLE SİL

GÜNCELLE GÖRÜNTÜLE

OgrtID	OgrtAd	OgrtSoyad	OgrtMail	OgrtSfite	Unvan
1234567890	Nihat	Doğan	nihatdogan@duz...	123456	profesör
1234567891	Nisa	Akdoğan	nisaakdogan@d...	nisa123	Araştırm

Akademisyen bilgileri sisteme başarılı bir şekilde kaydedildi.

Tamam

Şekil 28 – Öğretim Elemanı Formu

Ders Kayıt Formu

Bölüm Ad: Bilgisayar Mühendisli

Ders Kodu: BM202

Yarıyıl: 3

Ders Adı: İşaretler ve Sistemler

Dil: Türkçe

Kredi: 3

AKTS: 4

EKLE SİL

GÜNCELLE GÖRÜNTÜLE

DersKodu	Yarıyıl	DersAdı	Dil	Kredi
BM104	1	Algoritma	Türkçe	3
BM105	1	Veritabanı Yönetim	Türkçe	3
BM108	1	Web Programlama	Türkçe	3
BM201	3	İşletim Sistemleri	Türkçe	4
EE105	1	EDT	Türkçe	3

Ders bilgileri sisteme başarılı bir şekilde kaydedildi.

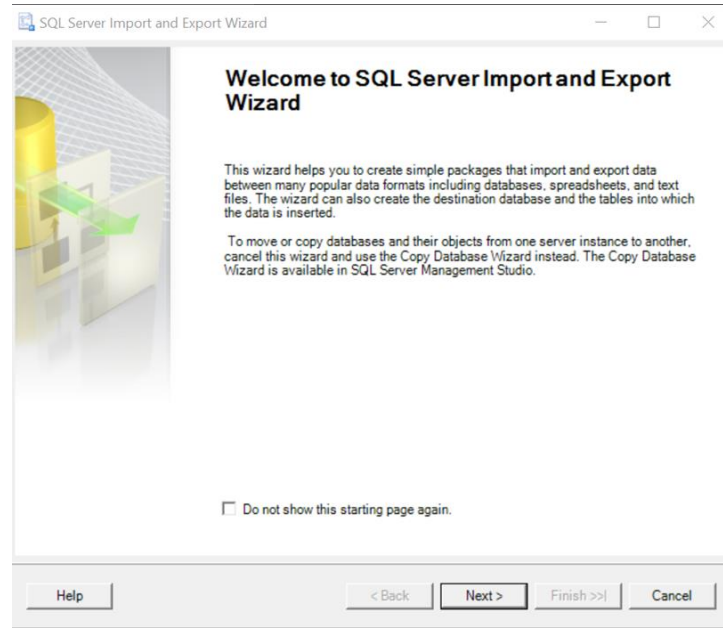
Tamam

Şekil 28.1 – Ders İşlemleri Formu

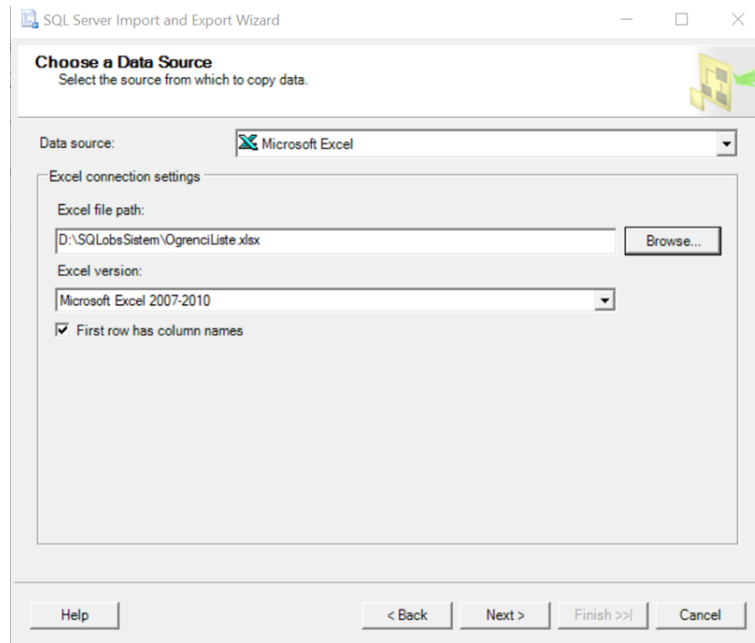
Yönetici bilgileri ile giriş yapıldıktan sonra yönetici ana menü kısmında Öğretim Elemanı, Öğrenci, Öğrenci İşleri ve Ders yer almaktadır. Yönetici, Şekil 28’de görüldüğü üzere öğretim elemanı kısmında öğrenci bilgi sistemine akademisyen ekleme işlemini, silme işlemini, güncelleme işlemini ve görüntüleme işlemini yapmaktadır. Şekil 28.1’de ise bu işlemleri ders için yapmaktadır. Aynı işlemleri öğrenci ve öğrenci işleri için de yapmaktadır.

9-IMPORT/EXPORT

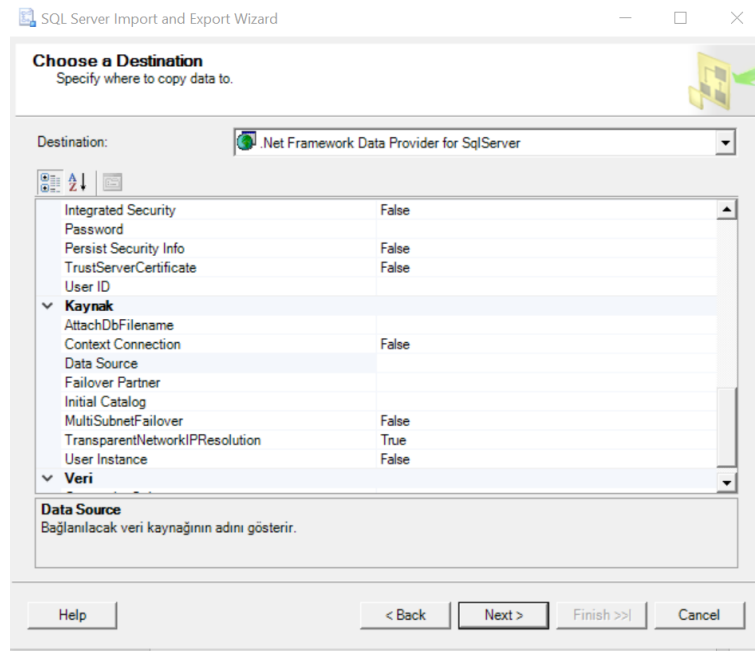
9.1 – IMPORT



Şekil 29 – Import İşlemi



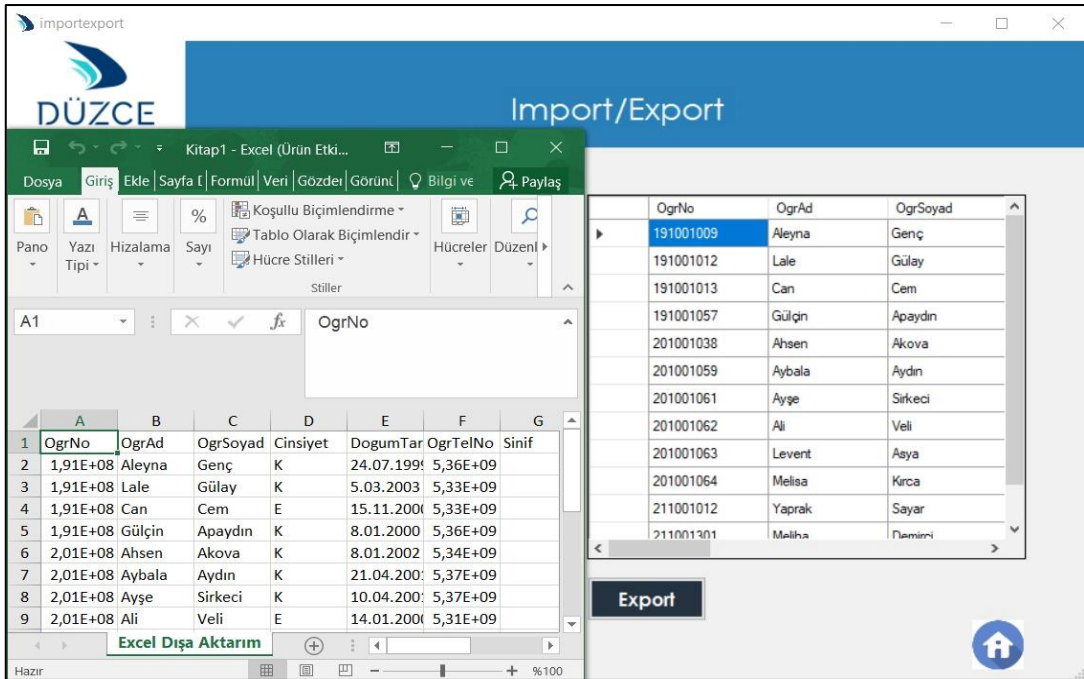
Şekil 29.1 – Import İşlemi



Şekil 29.2 – Import İşlemi

Dışarıda alınan verilerin veritabanında kullanılması için import edilmesi gerekir. Şekil 29, Şekil 29.1 ve Şekil 29.2’de görüldüğü üzere Import için öncelikle Microsoft Office import ve export data arayüzüne tıklanır. Veri kaynağı Microsoft Excel seçilir, dosya yolu için de bilgisayarda bulunan adres seçilir. Next’e tıklanıp tanımlama kısmından provider for SQL seçilir böylelikle işlemin tamamlanması sağlanır. Excel’deki veriler veritabanına aktarılmıştır.

9.2 – EXPORT



Şekil 30 – Export İşlemi

Öncelikle projeye excel'i eklemek için ilgili projenin başvurular kısmından excel seçilir, gerekli kütüphaneler eklenir. Daha sonra veritabanındaki veriler datagridview'e aktarılır ve ekranda görüntülenmesi sağlanır. Şekil 30'da bulunan 'Export' butonuna basılarak ekrandaki veriler excel dosyasına aktarılır. Bu işleme export adı verilir.

10-YEDEKLE/YEDEKTEN DÖN



Şekil 31 – Yedekle ve Yedekten Dön İşlemi

Şekil 31'de veritabanı yedekleme ve yedekten döndürme işleminin butonlar aracılığıyla yapıldığı bir arayüzün görüntüsü yer almaktadır. Bu arayüze, yönetici girişi ile ulaşılabilmektedir. “Veritabanını Yedekle” kutucuğunda bulunan “Seç” butonu ile veritabanı yedeğinin bilgisayarda hangi klasöre alınacağı seçilir. Seçimin ardından yedekle butonuna basılması ile belirtilen dosyaya veritabanının yedeği çıkartılır. Veritabanı yedeği dosyasının isminde yedekleme tarih ve saati bulunur, dosya “.bak” uzantılıdır. Veritabanı yedeği geri yüklenmek istendiğinde, “Veritabanını Yedekten Dön” kutucuğunda bulunan “Seç” butonu ile veritabanı yedeğinin bulunduğu dosya seçilir. Seçimin ardından “Yedekten Dön” butonuna basılması ile veritabanı yedeği geri döndürülür.

11- UYGULAMA ARAYÜZÜ

Öğrenci Bilgileri

Öğrenci Bilgi Formu

Cinsiyet: K

Öğrenci Mail: n@ogr.duzce.edu.tr

Öğrenci Şifre: 123456

Danışman Ad: Nihat

Danışman Soyad: Doğan

Bölüm: Bilgisayar Mühendisi

Tel No: 5362457484

Öğrenci No: 191001057

Öğrenci Adı: Gülçin

Öğrenci Soyadı: Apaydın

Sınıf: 4

GÜNCELLE

Şekil 32 – Öğrenci Bilgileri

Öğrenci girişi yapıldığı takdirde ana menüde ‘Ders İşlemlerim’ ve Bilgilerim kısmı yer alacak şekilde bir tasarım yapıldı. Şekil 32’de görüldüğü üzere bilgilerim kısmında öğrenci, bilgilerini görüntüleyebilecek, görüntülediği bilgilerin çoğunu değiştiremeyecek, eğer isterse mail adresini, şifresini ve telefon numarasını güncelleyebilecek şekilde ayarlandı. Bilgileri veritabanından eğer öğrencinin numarası OgrNo ile eşleşiyorsa o öğrenci numarasına ait verileri çekmesi ve öğrencinin bu bilgileri görüntülenmesi sağlandı. Öğrenci Resim yolu veritabanına eklenip görüntülenmesi sağlandı.

Not Görüntüleme İşlemleri

Öğrenci No: 191001009

	final	ortalama	harfnotu	OgrtID	DersKodu	OgrNo
	90	86	BA		BM104	191001009

Şekil 33 – Not Görüntüleme İşlemleri

Şekil 33’de bulunan Not Görüntüleme işlemleri kısmında öğrenci seçmiş olduğu dersin vizesini, finalini, ortalamasını ve harf notunu görüntüleyebilecek şekilde bir tasarım yapıldı. Öğrenci, ilk önce öğrenci numarasını ilgili textboxa girebilmesi ve daha sonra ara butonuna bastığında ise veritabanındaki Not_ tablosundan ilgili veriler çekilerek

öğrencinin ortalamasını, vize, final ve harf notunu görüntüleyebilmesi sağlandı. Veri çekme işlemi Entity Framework ile yapıldı.