

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**GRAF TEORİSİNE DAYALI YERLEŐTİRME
UYGULAMALARI**
(Yüksek Lisans Tezi)

Hazırlayan:
Kenan KILIÇASLAN

İSTANBUL, 2007

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYRA MÜHENDİSLİĞİ ANABİLİM DALI

**GRAF TEORİSİNE DAYALI YERLEŐTİRME
UYGULAMALARI**
(Yüksek Lisans Tezi)

Hazırlayan:
Kenan KILIÇASLAN
Öğrenci No:
050820006

Danışman:
Dr. Rifat ÇÖLKESEN

İSTANBUL, 2007

GRAF TEORİSİNE DAYALI YERLEŐTİRME UYGULAMALARI

Tezi Hazırlayan: Kenan KILIÇASLAN

Özet

Graf bir ilişkiler kümesidir. Hemen her alanda kullanılır. Graf teorisi ile ders programı, sınav programı yerleőtirme gibi yazılımlar gerçekleştirilebilir. Bu çalışmada graf renklendirme algoritması ile sınav programı yerleőtirmesi gerçekleştirilmiştir.

Sınav programı ve algoritması geliştirilirken Beykent Üniversitesi Meslek Yüksek Okulu ihtiyacı dikkate alınmıştır.

Graf renklendirme algoritması ile ders çakışmaları önlenabilir ve en az sayıda sınav oturumu tespit edilebilir. Bu bize akademik takvimde verilen sınav tarihlerinde yerleőtirmenin başarımının ölçütünü de verir.

Graf renklendirme algoritması, özel kısıt ve ihtiyaçları karşılamakta yetersizdir. Bu nedenle başka algoritmalar ile birlikte kullanılmalıdır. Bu kısıtlardan bazıları, deęişik bölümlerde okutulan ancak ders içerięi aynı olan derslerin aynı gün ve saatte olması istenir. Bazı derslerin sınavı uygulamalıdır, yerleőtirme yapılırken birden fazla ders aynı gün ve aynı saatte aynı uygulama yerini istememelidir.

Graf renklendirme yapılırken aynı renge sahip dersler isteęi tam olarak yerine getirmeyebilir. Örneęin, üniversitemizde final sınavları 10 iş gününde yapılır, bu 40 sınav oturumuna denktir, renklendirme algoritması ise 16 renk bulur. Bu 16 rengi 40 oturuma dengeli ve optimum şekilde dağıtmak gerekir.

Bu çalışmanın hedefi öğrenciyi, sınav gözetmenini en az yoran, derslik kısıtlarını dikkate alan bir sınav programı çıkarmaktır. Bu çalışma bir İnternet uygulamasıdır.

Anahtar Kelimeler: Graf, Sınav, Yerleőtirme, Algoritma, Yazılım.

ORGANIZATION APPLICATIONS BY USING GRAPH THEORY

Presented by: Kenan KILIÇASLAN

Abstract

Graph is a set of relations. Graphs are used in different areas. By using Graph Theory softwares which organize exam and course schedules can be developed. In this study by using Graph Coloring Algorithm we developed an application that manages exam schedules.

During developing this software and algorithm we took into consideration the needs of Beykent University School of Vocational Studies.

By using Graph Coloring Algorithm coincidences can be prevented and the least number of exam sessions can be determined. We can find out Success criteria of allocating exams between dates stated in Academic Calendar.

Graph Coloring Algorithm is incapable of responding special restrictions and requirements. Therefore it must be used with other algorithms. One of these restrictions is: The course exams that are thought in different departments with some contents are must be hold at the same date and time. Some of the exams are hold in labs. Thus, at a given time more than one exam must not be allocated at the same lab.

While doing graph coloring, the courses in same color may not answer the purpose. For example final exams at our university last 10 working days. It means 40 exam sessions. However coloring algorithm finds 16 colors. These 16 colors must be applied to 40 sessions in an optimum way.

The aim of this study developing an online exam scheduling system that can organize exam schedules which tire both students and exam supervisors minimum by taking into consideration restrictions

Key Words: Graph, Exam, Organize, Algorithm, Software

İÇİNDEKİLER

Yemin Metni	
Jüri Sayfası	
Türkçe Özet ve Anahtar Kelimeler	
İngilizce Özet ve Anahtar Kelimeler (Abstract)	
Tablolar Listesi	III
Şekiller Listesi	IV
1. GENEL İHTİYAÇLAR	1
1.1. İHTİYAÇ TESPİTİ	1
1.1.1. Genel	1
1.1.2. Veri Yapısı	1
1.1.2.1. Veritabanında Kullanılan Tablolar ve Depolanan Bilgiler	1
1.1.2.1.1. Ana Tablo ve View'ler	1
1.1.2.1.2. Yardımcı Tablolar ve View'ler	6
2. ALGORİTMA	11
2.1. Graf Renklendirme Algoritması	
11	
2.2. Tarih Kısıdına Uygun Yerleştirmenin Yapılması	
13	
2.2.1. Tarih Kısıtı Koşulunun Sağlanması	14
2.2.2. Aynı Anda Olma Koşulunun Sağlanması	17
2.2.3. Derslerin Çakışmasını Önlemek	18
2.2.4. Bir Sonraki Gün Olması Gereken Sınavlar	20
2.2.5. Matrislerin Oluşturulması	21
2.2.5.1. Aynı Anda Olmayacak Derslerin Matrisi	21
2.2.5.1.1. Öğrencilerin Derslerinin Çakışmasını Önlemek için	21
2.2.5.1.2. Dersliklerdeki sınavın çakışmasını önlemek için	22
2.2.5.1.3. Her Ders İçin Aynı Anda Olmayacak Diğer Dersler Dizisi	24
2.2.5.2. Bir Sonraki Gün Olacak Dersler Matrisinin Oluşturulması	24
2.2.5.2.1. Her Ders İçin Aynı Anda Olmayacak Diğer Dersler Dizisi	26
2.2.5.3. Bir Sonraki Oturumda Olmayacak Dersler Matrisi Oluşturma	26
2.2.5.3.1. Her Ders İçin Bir Sonraki Oturumda Olmayacak Diğer Dersler Dizisi	28
2.2.6. Renklendirme	28

2.2.6.1.	Düğümün derecesinin hesabı	28
2.2.6.2.	Renk Numaralarının Verilmesi	29
2.2.7.	Yerleştirme	31
2.2.7.1.	AAO Kontrol Fonksiyonu	31
2.2.7.2.	BSOO Kontrol fonksiyonu	32
2.2.7.3.	BSGO Kontrol fonksiyonu	33
2.2.7.4.	Oturum Numarası Atama Fonksiyonu	34
3.	YAZILIM VE ÖRNEK KULLANIM	35
3.1.	Ana Menü	
	35	
3.1.1.	Tüm menü elemanlarının ortak kodları	35
3.1.1.1.	connect.php dosyası	36
3.1.1.2.	listeleme-sorgu-sinavsalonukayit.php	36
3.1.1.3.	islemler.php	37
3.1.2.	Örnek Kullanım	43
3.1.2.1.	Ders Bölüm	43
3.1.2.2.	Bölümler menüsü	45
	Bölümler liste görünümü	45
4.	SONUÇ VE DEĞERLENDİRME	49
	Kaynakça	50
	ÖZGEÇMİŞ	51

TABLULAR

<u>Tablo No.</u>	<u>Sayfa</u>
1. Graf Renklendirme	11
2. Komşuluk matrisi	12
3. Ana Menü	35
4. Bölümlerin ders listesi görünümü	43
5. Kayıt değiştirme ekran görünümü	44
6. Kayıt değiştir bölümün görünüşü	44
7. Bölümlerin liste görümü	45
8. Bölümler değiştir ekran görünümü	45
9. Ders listesi görünümü	46
10. Öğretim elemanları liste görünümü	46
11. Ders grupları liste görünümü	47
12. Sınav salonları liste görünümü	47
13. Sonuç ders programının bir kısmının ekran görünümü	48

ŞEKİLLER

<u>Şekil No.</u>	<u>Sayfa</u>
Şekil 1. Graf Renklendirme	11
Şekil 2. Komşuluk matrisi	12
Şekil 3. Ana Menü	35
Şekil 4. Bölümlerin ders listesi görünümü	43
Şekil 5. Kayıt değiştirme ekran görünümü	44
Şekil 6. Kayıt değiştir bölümün görünüşü	44
Şekil 7. Bölümlerin liste görümü	45
Şekil 8. Bolumler değiştir ekran görünümü	45
Şekil 9. Ders listesi görünümü	46
Şekil 10. Öğretim elemanları liste görünümü	46
Şekil 11. Ders grupları liste görünümü	47
Şekil 12. Sınav salonları liste görünümü	47
Şekil 13. Sonuç ders programının bir kısmının ekran görnümü	48

1. GENEL İHTİYAÇLAR

1.1. İHTİYAÇ TESPİTİ

1.1.1. Genel

Beykent Üniversitesi Meslek Yüksek Okulu'nda bulunan öğrenciler 2 yıl okumaktadır. Yönetmelik gereği sadece kendi bölümlerinden ders alabilmektedir. Öğrencilerin aldığı dersler veritabanında tutulmaktadır.

Sınav programında herhangi bir öğrencinin sınavı çakışmamalıdır.

Aynı ders içeriğine sahip, farklı Bölümler de okutulan ve aynı öğretim elemanı tarafından verilen derslerin sınavları aynı gün ve aynı saatte olmalıdır.

Bir öğrenci bir sınava girdikten sonra en az bir oturum boşluk olmalıdır.

Akademik Takvimde gün sayısı yeterli ise sadece kendi sınıfında sınava giren yani alttan aldığı ders bulunmaya öğrenci aynı gün sadece bir sınava girmelidir.

Uygulamalı derslerin, uygulama sınavları aynı gün ve saatte aynı sınıf istememelidir. Örneğin, 16 Programda BMYOD172 Bilgisayar 2 dersi vardır. Bu dersin sınavları PC-LAB 1 ve PC-LAB 3 de yapılmaktadır. Bu 16 adet BMYOD172 Bilgisayar 2 dersi aynı renkte olmamalıdır.

Her ne kadar dersin öğretim elemanı bulunmadan sınav yapılabilse de bazı durumlarda bazı sınavlar için tarih kısıdı da tanımlanır. Özellikle uygulamalı dersin öğretim elemanı yarı zamanlı ise bu dersin sınavı öğretim elemanının geldiği günlerden birine konmalıdır. Yazılım bunu da başarmalıdır.

1.1.2. Veri Yapısı

Program için öğrencinin aldığı dersleri, öğretim elemanlarını, derslikleri, fakültede okutulan tüm dersler ve bir bölümde okutulan dersler öğretim elemanları ilişkisi, derslik ve tarih kısıtları gibi bilgilerin kayıtlı olduğu bir veritabanı gereklidir. Bu amaçla bir metin dosyası da kullanılabilse de bu çalışmada Microsoft SQL Server 2005 kullanılmıştır.

Kullandığımız veritabanına “sınav” adı verdik.

1.1.2.1. Veritabanında Kullanılan Tablolar ve Depolanan Bilgiler

1.1.2.1.1. Ana Tablo ve View'ler

Ana tablolar ve view'ler da, öğrencilerin aldığı dersler, derslik sınav kapasiteleri, derslik kısıtları gibi doğrudan algoritmanın kullandığı bilgiler vardır.

Karakter alanlarda, varchar datatipi seçilmiştir. Varchar data tipi değişken uzunluklu karakter katarlarını saklamakta kullanılır. Örneği varchar(50) datatipinde olan bir alanda “KENAN” değerini sakladığımızda veritabanında 50 karakter uzunlukta değil gerçek verinin uzunluğu artı bir karakterlik yer kaplar yani 6 karakter uzunluktadır. Bu tip datatipileri uzunluğu önceden kestirilemeyen durumlarda kullanılır.

- **tblBolumler tablosu** : Fakülte'de bulunan bölümlerin kodları ve isimleri kayıtlı.

Tablo 1. tblBolumler tablosu alanları

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>	<i>Açıklama</i>
BolumNo	varchar(10)	NOT NULL	Bölüme verilen kod saklanır.
BolumAdi	varchar(50)	NOT NULL	Bölümün adı saklanır.
SiraNo	int	IDENTITY(1,1) NOT NULL	Her bölüme birden başlayıp artan numara verir. Diğer tablolarla ilişkilerde bu alan kullanılmıştır.
İndexler	BolumNo PrimaryKey		Her kayıda birden başlayıp artan numara verir.

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE dbo.tblBolumler (BolumNo varchar(10) NOT NULL, BolumAdi
varchar(50) NOT NULL, SiraNo int IDENTITY(1,1) NOT NULL,
CONSTRAINT PK_tblBolumler PRIMARY KEY CLUSTERED ( BolumNo ASC ) )
```

- **tblDersler tablosu** : Fakülte'de okutulan derslerin kodları ve isimleri kayıtlı

Tablo 2. tblDersler tablosu alanları

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>	<i>Açıklama</i>
DersKodu	varchar(12)	NOT NULL	Fakültede okutulan dersin kodu yer alır.
DersAdi	varchar(50)	NOT NULL	Dersin adı yer alır.
SiraNo	int	IDENTITY(1,1) NOT NULL	Her kayıda birden başlayıp artan numara verir.

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE dbo.tblDersler(DersKodu varchar(12) NOT NULL, DersAdi
varchar(50) NOT NULL, SiraNo int IDENTITY(1,1) NOT NULL, CONSTRAINT
PK_tblDersler PRIMARY KEY CLUSTERED ( DersKodu ASC ) )
```

• **tblOgretimEleman tablosu** : Ders veren öğretim elemanlarının sicil numaraları, isimleri ve ünvanları vardır.

Tablo 3. tblOgretimEleman tablosu alanları

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>	<i>Açıklama</i>
SicilNo	int	NOT NULL	Öğretim elemanının sicil no saklanır.
Adi	varchar(50)	NOT NULL	Öğretim elemanının adı saklanır.
Soyadi	varchar(50)	NOT NULL	Öğretim elemanının soyadı saklanır.
Unvani	varchar(50)	NULL	Öğretim elemanının ünvanı saklanır.
GozetmenlikVar	tinyint	DEFAULT (1)	Sınavda gözetmenlik verilip verilmeyeceği bilgisi yer alır. Varsayılan 1 değeri gözetmenlik yapacaktır.
Kadrolu	tinyint	DEFAULT (1)	Öğretim elemanının kadrolumu ders saat ücretli mi olduğu bilgisi yer alır. Varsayılan 1 değeri Kadroludur.
SiraNo	int	IDENTITY(1,1) NOT NULL	Her öğretim elemanına birden başlayıp artan numara verir. Diğer tablolarla ilişkilerde bu alan kullanılmıştır.

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE dbo.tblOgretimEleman (SicilNo int NOT NULL,Adi [varchar(50)
NOT NULL, Soyadi varchar(50) NOT NULL, Unvani varchar(50) NULL,
GozetmenlikVar tinyint NULL CONSTRAINT DF_tblOgretimEleman_GozetmenlikVar
DEFAULT (1),
Kadrolu tinyint NULL CONSTRAINT DF_tblOgretimEleman_Kadrolu DEFAULT (1),
SiraNo int IDENTITY(1,1) NOT NULL, CONSTRAINT PK_tblOgretimEleman PRIMARY
KEY CLUSTERED ( SicilNo ASC ))
```

• **tblDersBolum tablosu** : Bu tablo, tblDersler tablosunda kayıtlı dersler, tblBolumler de kayıtlı blüm isimleri ve tblOgretimEleman tablosunda kayıtlı öğretim elemanları arasında

çok-çok ilişkisini sağlar. Herhangi bir bölümde okutulan dersin hangi öğretim elemanı tarafından verildiği bilgisi, derslik kısıtı, tarih kısıtının olup olmaması bilgisi ve başka bir dersle aynı anda olma durumu bilgisi de vardır.

Tablo 4. tblDersBolum tablosu alanları

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>	<i>Açıklama</i>
DersKodu	varchar(12)	NOT NULL	Bolumde okutulan dersin kodu bulunur.
BolumNo	varchar(10)	NOT NULL	Bölüm numarası bulunur.
YariYil	tinyint	NOT NULL	Dersin okutulduğu yarı yıl bilgisi yer alır.
SicilNo	int	NOT NULL	Bu dersi veren öğretim elemanının sicilno bulunur.
OgrenciSayisi	smallint		Bu dersi alan öğrenci sayısı yer alır.
OturumSayisi	tinyint	DEFAULT (1)	Bu sınavın süreceği oturum sayısı yer alır.
DerslikKisitiVar	tinyint	DEFAULT (0)	Bu ders için sınav salonu olarak belirtilmiş bir istek varmı. Örneğin sınav laboratuvarda mı yapılacak. Bu alana sadece 0 ve 1 rakamı girilir. 0 ise kısıt yok 1 ise kısıt var.
OturumKisitiVar	tinyint	DEFAULT (0)	Bu ders için bir tarih kısıtı varmı. 0 ise kısıt yok 1 ise kısıt var.
AlttanAlanYok	tinyint	DEFAULT (0)	Bu dersi alttan alan varmı bilgisi kaydedilir. 0 ise alttan alan yok 1 ise alttan alan var.
SiraNo	İnt	IDENTITY(1,1) NOT NULL	Her kayıda birden başlayıp artan numara verir. Diğer tablolarla ilişkilerde bu alan kullanılmıştır.

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE dbo.tblDersBolum(DersKodu varchar(12) NOT NULL, BolumNo
varchar(10) NOT NULL, YariYil tinyint NOT NULL, SicilNo int NOT NULL,
OgrenciSayisi smallint, OturumSayisi tinyint CONSTRAINT
DF_tblDersBolum_OturumSayisi DEFAULT (1), [DerslikKisitiVar tinyint
```

```
CONSTRAINT DF_tblDersBolum_DerslikKisitiVar DEFAULT (0), OturumKisitiVar
tinyint CONSTRAINT DF_tblDersBolum_OturumKisitiVar DEFAULT (0),
AlttanAlanYok tinyint CONSTRAINT DF_tblDersBolum_AlttanAlanYok
DEFAULT (0), SiraNo int IDENTITY(1,1) NOT NULL)
```

• **tblGrup tablosu** : Farklı bölümlerde okutulan ve aynı içeriğe sahip dersler, bir grupta toplanır. Bu tabloda grup kodu ve grup isimleri bulunmaktadır.

Tablo 5. tblGrup tablosunun alanları ve özellikleri

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>	<i>Açıklama</i>
GrupNo	varchar(50)	NOT NULL	Aynı anda olma koşulu için birleştirilen derslere verilen isimler yer alır.
GrupAdi	varchar(50)	NOT NULL	Gruba verilen isim bulunur.
SiraNo	int	IDENTITY(1,1) NOT NULL	Her kayıda birden başlayıp artan numara verir.

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE dbo.tblGrup(GrupNo varchar(50) NOT NULL, GrupAdi varchar(50)
NOT NULL, SiraNo int IDENTITY(1,1) NOT NULL, CONSTRAINT PK_tblGrup PRIMARY
KEY CLUSTERED ( GrupNo ASC))
```

• **tblGrupDers tablosu**: Bu tabloda tblDersBolum tablosundaki herhangi bir kaydın bağlı olduğu grubun kodu ve bu kaydın ID numarası yer alır. Böylece herhangi bir ders için aynı anda olma durumunu sağlayan gruplandırma yapılır.

Tablo 6. tblGrupDers tablosunun alanları ve özellikleri

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>
GrupNo	varchar(50)	NOT NULL
SiraNo	int	NOT NULL

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE dbo.tblGrupDers([GrupNo varchar(50), SiraNo int NOT NULL])
```

• **tblDerslikKisiti tablosu** : tblDersBolum tablosundaki her hangi bir kaydın eğer varsa hangi dersliklerde olacağı bilgisi vardır. Bu tabloda salon no ve kaydın ID numarası yer alır.

Tablo 7. tblDerslikKisiti tablosunun alanları ve özellikleri

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>
SiraNo	int	NOT NULL
SalonNo	varchar(10)	NOT NULL

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE dbo.tblDerslikKisiti (SiraNo int NOT NULL, SalonNo varchar(10) NOT NULL)
```

• **tblTarihKisit tablosu** : tblDersBolum tablosundaki herhangi bir kayıt için varsa hangi oturumlarda sınavının yapılabileceği bilgisi vardır. Bu tabloda oturum kısıt değeri girilir. Renklendirme de öncelikli olarak dikkate alınır.

Tablo 8. tblTarihKisit tablosunun alanları ve özellikleri

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>
BolumDersSiraNo	int	NOT NULL
OturumSiraNo	int	NOT NULL

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE dbo.tblTarihKisit(BolumDersSiraNo int NOT NULL, OturumSiraNo int NOT NULL)
```

vwOgrenciDersler viewi : Meslek Yüksek Okulunda her bölüm bir öğrenci olarak dikkate alınabilir. Bu çalışma, doğrudan öğrenciyi dikkate alabilmesi için 1. sınıf derslerinin tüm birinci sınıf öğrencileri ve tüm 2. sınıf öğrencilerinin aldığı, 2. sınıf derslerini ise sadece 2. sınıf öğrencilerini aldığı varsayımı ile program denenmiştir. Bu view, bir tablo olarak oluşturulduğunda öğrenciler ve aldığı dersler kaydedilebilir ve program buna göre sınav programını oluşturur.

Viewi oluşturan SQL kodu:

```
create view [dbo].[vwOgrenciDersler] as  
select *, ltrim(rtrim(str(yy)))+'.'+ltrim(rtrim(str(BolumNo))) AS BolumKodu  
from vwBolumDersler
```

1.1.2.1.2. Yardımcı Tablolar ve View'ler

Yardımcı tablolar ve view'ler da, algoritmanın doğrudan kullanmadığı, verinin bütünlüğünü sağlayan bilgiler vardır.

• **tblSinavSalonlari tablosu:** Sınavların yapılacağı salonların numaraları, salonda sınava girecek öğrenci sayısı gibi bilgiler vardır.

Tablo 9. tblSinavSalonlari tablosunun alanları ve özellikleri

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Özelliği</i>
SalonNo	varchar(10)	NOT NULL
SinavKapasitesi	smallint	NOT NULL
AtamaOnceligi	tinyint	DEFAULT (1)
SiraNo	int	IDENTITY(1,1)
GozetmenSayisi	tinyint	DEFAULT (1)

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE tblSinavSalonlari( SalonNo varchar(10) NOT NULL,  
SinavKapasitesi smallint NOT NULL, AtamaOnceligi tinyint CONSTRAINT  
DF_tblSinavSalonlari_AtamaOnceligi DEFAULT (1),  
SiraNo int IDENTITY(1,1) NOT NULL, GozetmenSayisi tinyint  
CONSTRAINT DF_tblSinavSalonlari_GozetmenSayisi DEFAULT (1),  
CONSTRAINT PK_tblSinavSalonlari PRIMARY KEY CLUSTERED ( SalonNo ASC))
```

• **tblSinavTarih tablosu :** Sınavların yapılacağı tarihler ve oturum bilgileri girilir.

Tablo 10. tblSinavTarih tablosunun alanları ve özellikleri

<i>Alan Adı</i>	<i>Data Tipi</i>	<i>Açıklama</i>
SinavNo	int	Vize ve finale verilen bir numara
Tarih	datetime	Sınavın yapılacağı tarih
OturumNo	tinyint	Sınavın ilk günündeki ilk sınava 1 değeri, diğerlerine sırası ile artan değerler.

Tabloyu oluşturan SQL kodu:

```
CREATE TABLE tblSinavTarih( SinavNo int NOT NULL,Tarih datetime,  
OturumNo tinyint)
```

• **vwBolumDersler viewi** : Bu çalışma öncelikle öğrencilerin aldığı dersleri dersleri dikkate alarak sınav yerleştirmesi yapmaktadır. Beykent Üniversitesi Meslek Yüksek Okulunda öğrenim süresi 2 yıldır (4 yarıyıl). Birinci sınıfta okuyan tüm öğrenciler birinci sınıfın tüm derslerini almaktadır. İkinci sınıf öğrencileri için ise hem tüm ikinci sınıf, hemde tüm birinci sınıf derslerini almakta olduğu kabulü yapılmıştır.

Her bölümün birinci sınıfı bir öğrenci, ikinci sınıfı ise başka bir öğrenci olduğu düşünülmüştür.

Her bölümde birinci ve ikinci sınıflar ayrı öğrenci olduğu için yukarıdaki kabulü sağlamak için her B.Ü. MYO nun her bölümünün her sınıfı için ayrı view oluşmuştur. Artık her view bir öğrencidir.

Yaptığımız uygulamada; örneğin Bankacılı ve Sigortacılık bölümünün 2. yarıyıldaki öğrencinin aldığı dersleri veren view aşağıdadır.

```
CREATE VIEW [dbo].[vw2701] as
SELECT 2 as yy,* FROM vwDersBolumGrup WHERE BolumNo='701' and YariYil=2
```

Bankacılı ve Sigortacılık bölümünün 4. yarıyıldaki öğrencinin aldığı dersleri veren view aşağıdadır. 4. yarıyıl öğrencileri 2. yarıyıldan okutulan BMOD102 ve BMOD104 kodlu derslerden başarısız olan yoktur.

```
CREATE VIEW [dbo].[vw4701] as
SELECT 4 as yy,* FROM vwDersBolumGrup WHERE BolumNo='701' and
(DersKodu+ltrim(YariYil) not in ('BMOD102','BMOD104'))
```

örneğin Bilgisayar Teknolojileri ve Programlama bölümünün 2. yarıyıldaki öğrencinin aldığı dersleri veren view aşağıdadır.

```
CREATE VIEW [dbo].[vw2702] as
SELECT 2 as yy,* FROM vwDersBolumGrup WHERE BolumNo='702' and YariYil=2
```

Bilgisayar Teknolojileri ve Programlama bölümünün 4. yarıyıldaki öğrencinin aldığı dersleri veren view aşağıdadır. 4. yarıyıl öğrencileri 2. yarıyıldan okutulan BMOD102 ve BMOD104 kodlu derslerden başarısız olan yoktur.

```
CREATE VIEW [dbo].[vw4702] as
SELECT 4 as yy,* FROM vwDersBolumGrup WHERE BolumNo='702'
```

Bunu tüm bölümler için ayrı ayrı yapıp bütün viewleri aşağıdaki gibi birleştirerek kabulümüzü gerçekleştirebiliriz.

Yukarıda oluşturulan tüm viewler aşağıdaki kod ile birleştirilerek vwBolumDersler viewi oluşturulur. Artık bu viewde kabulümüze uygun olarak tüm öğrencileri aldığı dersler bulunuz.

```
CREATE VIEW [dbo].[vwBolumDersler]
AS
SELECT      yy, DersKodu, BolumNo, YariYil, GrupNo
FROM        dbo.vw2701
UNION
SELECT      yy, DersKodu, BolumNo, YariYil, GrupNo
FROM        dbo.vw2702
UNION
SELECT      yy, DersKodu, BolumNo, YariYil, GrupNo
FROM        dbo.vw4701
UNION
SELECT      yy, DersKodu, BolumNo, YariYil, GrupNo
FROM        dbo.vw4702
```

vwTarihKisit viewi : Bu view'da sadece tarih kısıtı bulunan dersler ve istenilen oturum numaraları bulunur.

```
CREATE VIEW vwTarihKisit
AS
SELECT      tblDersBolum.DersKodu, tblDersBolum.BolumNo,
tblSinavTarih.SiraNo AS OturumSiraNo, LTRIM(STR(tblDersBolum.YariYil))
+ '.' + LTRIM(RTRIM(tblDersBolum.BolumNo)) + '-'
+ LTRIM(RTRIM(tblDersBolum.DersKodu)) AS BolumDers
FROM        tblSinavTarih INNER JOIN tblTarihKisit ON tblSinavTarih.SiraNo =
tblTarihKisit.OturumSiraNo INNER JOIN tblDersBolum ON
tblTarihKisit.BolumDersSiraNo = tblDersBolum.SiraNo
```


2. ALGORİTMA

2.1. Graf Renklendirme Algoritması

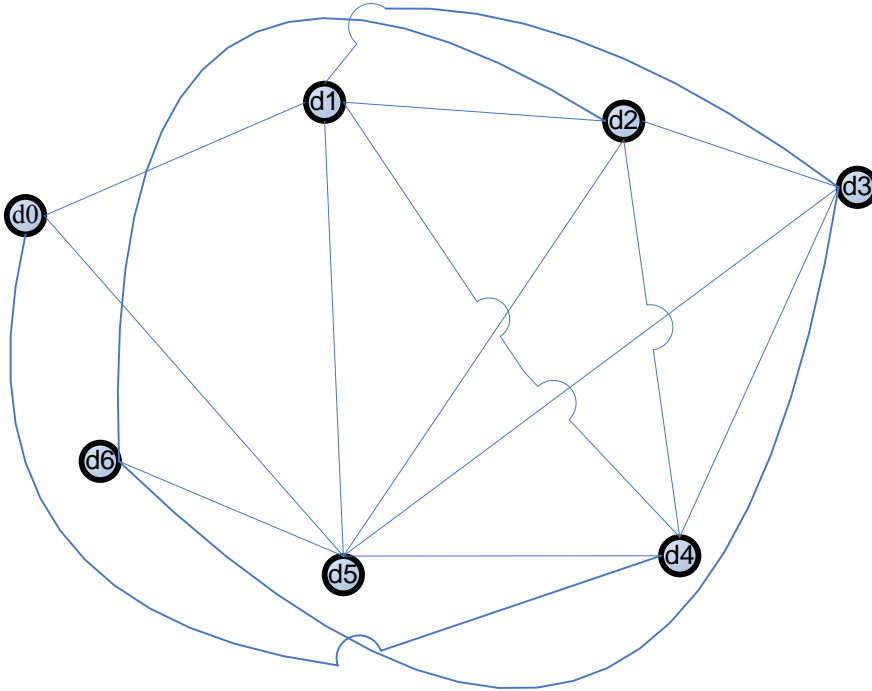
Çalışmamızda Welch ve Powel algoritması kullanılmıştır. Bu algoritmanın davranışı aşağıdaki gibidir.¹

Adım 1: D ğ mler derecelerine g re b y kten k   ge dođru sıralanır.

Adım 2: İlk renk numarası birinci sıradaki d ğ me atanır, daha sonra aynı renk numarası komşuluk matrisinde komşu olmayan diđer d ğ me verilir.

Adım 3: Renk numarası bir artırılır, bu numara daha  nce atama yapılmamıř d ğ mlerden derecesi en b y k olana verilir ve adım 2 diđer d ğ mler i in tekrarlanır.

řekil 1'deki grafa uygun komşuluk matrisi řekil 2'dedir. Buna yukarıdaki algoritmayı uygulayalım.



řekil 1. Graf Renklendirme

¹  lkesen, R. "Veri Yapıları ve Algoritmalar, Papatya Yayıncılık, İstanbul, řubat 2004.

	d_0	d_1	d_2	d_3	d_4	d_5	d_6	Derece
d_0		1	1		1	1		4
d_1	1		1	1	1	1		5
d_2	1	1		1	1	1	1	6
d_3		1	1		1	1	1	5
d_4	1	1	1	1		1	1	6
d_5	1	1	1	1	1		1	6
d_6			1	1	1	1		4

Şekil 2. Komşuluk matrisi

Uygulama adımları:

1. Adım: Dğümleri derecelerine göre aşağı doğru dizelim.

Düğüm	Derece
d2	6
d4	6
d5	6
d1	5
d3	5
d0	4
d6	4

Sırası ile renk numarası ataması yapalım. d2 dersine 1 numarayı atayalım. Komşuluk matrisinde d2 ile komşu olamayan ders yok, bu nedenle sıradaki diğer derse geçiyorum. d4 dersine sıradaki numara yani 2'yi atıyorum. Komşu olmayan düğüm yok diğer düğüme geçiyorum. d5'e 3 numarayı atıyorum. d5 ile de komşu olmayan yok. Sıradaki düğüm olan d1'e 4 numarayı atıyorum ve matrise baktığımızda d1'in d6 ile komşu olmadığı görülmektedir. Bu nedenle d6 yada 4 numarayı atıyorum. Hem d1 ve hemde d6 ile komşu olmayan var mı diye matrise bakalım. İkisine de aynı anda komşu olmayan yok sıradaki düğüme d3 e 5 numarayı atıyorum. d3 e komşu olmayan ve renk atanmamış var mı? d0 var ve bunada 5 numarayı atıyoruz. Renk atanmayan ders yok bu nedenle atama işini bitiriyoruz.

2.2. Tarih Kısıdına Uygun Yerleřtirmenin Yapılması

Tarih kısıt bilgileri tblTarihKisit tablosunda bulunmaktadır. Yazılım aynı bölüme aynı tarihi kısıt olarak girmeye izin vermemektedir. İlk girilen tarihi kabul etmektedir. Bu algorithmada SQL sorguları kullanılmaktadır. Her oturuma seri olarak bir numara verilmektedir. Örneğin sınav toplam 40 oturumda gerçekleşecek ise her oturuma sırası ile 1 - 40 arasında numara verilmektedir.

Bir ders için birden fazla oturum tercihi yapılabilmektedir. Örneğin BMYOD102 Türk Dili II dersi için 10,11,12,22,24 nolu oturumlar kısıt olarak girilmektedir. Sınav bu oturumlardan birinde yapılması gerekir. Bir başka ders 22,24,25 nolu oturumu tercih edebilir.

Tablo 11. Tarih kısıt tablosundaki örnek veriler

DersSıraNo	OturumNo
1	10
1	11
1	12
2	11
3	11
3	12
3	13
4	16
4	17
4	18
4	19
5	16
5	17
6	17

Tablo 12. Tarih kısıt tablosunda her bir oturumu istek sayısı

OturumNo	Sayı
10	1
11	3
12	2
13	1
16	2
17	3
18	1
19	1

Algoritmanın çalışması

1- Tablo 11 de bulunan tarih kısıt tablosunda oturum numarasına göre grupla ve herbir oturumun sayısını bul. SQL sorgusu sonucu olarak Tablo 12'daki veriler oluşur.

2- Tablo 11 daki verilerden maksimum olan oturum numarası tespit edilir. Örnek verimizde en çok tercih edilen oturumlar 11 ve 17 dir. Küçük numaralı oturum seçilir. Örneğimizde 11 nolu oturumdur.

3- Tablo 11 da 11 numaralı oturum tercih edilmiş olan derslere bu oturum atanır. 1,2,3 nolu derslere 11 nolu oturum atanır.

4- Oturum atanmış dersler hariçte tutularak, 1, 2, 3 nolu işlemler tekrarlanır.

5- Oturum atanmayan dersler kalmayıncaya kadar döngü devam eder.

6- Her atanan gruba TarihGrup0, TarihGrup1, gibi artan isim ver

7- Yerleştirme yaparken dersi kodu yerine TarihGrup0 gibi grup isimlerini kullan.

2.2.1. Tarih Kısıtı Koşulunun Sağlanması

2. bölümündeki algoritma aşağıdaki sorgu ile oluşturulmuş view üzerinde çalıştırıldı.

Tablo 13. vwTarihKisit view'ini oluşturan sorgu

```
CREATE VIEW vwTarihKisit AS
SELECT db.DersKodu, db.BolumNo, st.SiraNo AS OturumSiraNo,
LTRIM(STR(db.YariYil)) + "." + LTRIM(RTRIM(db.BolumNo)) + "-" +
LTRIM(RTRIM(db.DersKodu)) AS BolumDers FROM tblSinavTarih st INNER JOIN
dbo.tblTarihKisit tk ON st.SiraNo = tk.OturumSiraNo INNER JOIN dbo.tblDersBolum db
ON tk.BolumDersSiraNo = db.SiraNo
```

Tablo 14. vwTarihKisit view'indeki alanlar ve açıklamaları

<i>Alan Adı</i>	<i>Açıklama</i>
DersKodu	tblDersBolum tablosundaki DersKodu alanı
BolumNo	tblDersBolum tablosundaki BolumNo alanı
BolumDers	YY.BN-DK düzeninde dersin okutulduğu yarıyıl, bölüm no ve ders kodu alanlarının birleşmesinden oluşur. Burada YY: yarıyıl, BN: Bölüm no, DK:Ders kodu. Örnek : 2.702-BMYOD102 ; Burada 2. yarıyıl da 702 kodlu bölümde okutulan BMYOD102 kodlu ders.

2 deki algoritmaya uygun olarak PHP de aşağıdaki kod yazılmıştır.

a) İçi boş \$TarihGrup dizisi oluştur

```
1 $TarihGrup=array()
```

b)

```
2 $sorgu="SELECT TOP 1 OturumSiraNo, COUNT(*) AS Sayi FROM dbo.vwTarihKisit
GROUP BY OturumSiraNo ORDER BY Sayi DESC, OturumSiraNo ASC";
```

Bu sorgu ile *vwTarihKisit* view'i OturumSiraNo alanına göre gruplandırılıyor, OturumSiraNo sayısına göre büyükten küçüğe sıralanıyor, sıralanan bu veri kendi içinde OturumSiraNo alanına göre sıralanıyor. Elde edilen veri tablosunun sadece ilk satırını okumaktadır. Bu sorgu çalıştırıldığında en çok tercih edilen oturum numarası bulunur.

c) Aşağıdaki kod ile \$sorgu değişkeni içindeki sorgu çalışırılır.

```
3 $sonuc=mssql_query($sorgu, $link);
```

d) Aşağıdaki kod ile en çok tercih edilen oturum numarasını tercih eden dersler tespit edilir. Kullanılan değişkenler ve anlamları;

\$i	: TarihGrup sayacı
\$dersler	: TarihGrup ataması yapılan DersBolum listesi
\$satis	: Veritabanı kursorünün bulunduğu satırdaki değer dizisi
\$OturumSiraNo	: Seçili oturum no
\$sorgu	: TarihGrup numarası vwTarihKisit view'i sorgulanarak verilmektedir. 9. satırdaki sorgu TarihGrup ataması yapılmadığı durumda çalışır. 11. satırdaki sorgu da vwTarihKisit view'ini sorgular ancak daha önce TarihGrup numarası verilenleri dikkate almaz.
\$AtanmisOturumNumarasi	: Bir tarih grubunda bulunan derslere öncelikte oturum numarası atanır. Bu değişken içinde herhangi bir tarih grubunun bulunması gereken oturum numarası vardır.
\$TarihGrup	: Bu bir iki boyutlu dizidir. Dizinin birinci boyutunun indisi TarihGrup0, TarihGrup1 gibi grup adı, ikinboyutun indisi ise aynı grup içinde yer alan dersin sıra numarası vardır. Değeri DersBolumKodu'dur. Örnek: \$TarihGrup["TarihGrup0"][0] = "2.702-BMYOD102" dir.

```
4 $i=0;
5 $dersler="";
6 while ($satis=mssql_fetch_array($sonuc)) {
7     $OturumSiraNo=$satis['OturumSiraNo'];
8     if ($dersler=="") {
9         $sorgu="SELECT BolumDers FROM vwTarihKisit WHERE
10                OturumSiraNo=".$satis['OturumSiraNo'];
11     } else {
12         $sorgu="SELECT BolumDers FROM vwTarihKisit WHERE
13                OturumSiraNo=".$satis['OturumSiraNo']." AND
14                BolumDers NOT IN (".substr($dersler,1).")" ;
15     }
16 }
17 $snc=mssql_query($sorgu,$link);
18 $j=0;
19 while ($str=mssql_fetch_array($snc)) {
20     $TarihGrup["TarihGrup".$i][$j]=$str['BolumDers'];
21     $j++;
22 }
```

```

19     $AtanmisOturumNumarasi[$str['BolumDers']]=$OturumSiraNo;
20 }

21 $AtanmisOturumNumarasi["TarihGrup".$i]=$OturumSiraNo;

22 foreach($TarihGrup as $TarihGrup)
23     foreach($TarihGrup as $TarihDers) {
24         $tdersler=$tdersler."",".$TarihDers.""";
25     }

26 $sonuc=mssql_query($sorgu,$link);
27 $i++;
28 }

```

e) Bazı dersler aynı anda sınav olmalıdır. Örneğin BMYOD102 Türk Dili II dersinin içeriği tüm bölümlerde aynıdır. Bu dersin sınavı tüm bölümlerde aynı gün ve aynı saatte olmalıdır. Bu ders için herhangi bir bölümde tarih kısıtı belirtilmiş ise, diğer bölümler de aynı kısıtın kapsama alanındadır. Aşağıdaki kod, aynı anda olma koşullu bir derse tarih kısıtı verilmiş ise bu kısıtın diğer derslere de uygulanmasını sağlamak amacıyla yazılmıştır.

4 – 28 arasındaki kodlar Tablo 15 deki \$TarihGrup dizisini oluşturmaktadır.

Tablo 15. \$TarihGrup dizisinin örnek değeri

```

Array ( [TarihGrup0] => Array ( [0] => 2.706-BMYIY126 [1] => 2.709-BMYIY126
[2] => 4.707-BMYOD104 [3] => 2.712-BMYOD104 [4] => 2.713-BMYOD104
[5] => 2.714-BMYOD104 [6] => 2.715-BMYOD104 [7] => 2.716-BMYOD104
[8] => 2.717-BMYOD104 )
[TarihGrup1] => Array ( [0] => 2.708-BMYHA104 [1] => 2.708-BMYHA104
[2] => 2.712-BMYOD161 [3] => 2.711-BMYOD161 [4] => 2.711-BMYOD161 )
[TarihGrup2] => Array ( [0] => 4.706-BMYGI262 [1] => 4.717-BMYUT262
[2] => 4.709-BMYIY208 [3] => 4.711-BMYMU262 )
[TarihGrup3] => Array ( [0] => 4.702-BMYBT266 [1] => 4.704-BMYCG262
[2] => 4.715-BMYTR218 [3] => 4.708-BMYHA264 ))

```

```

29 foreach ($TarihGrup as $TarihGrupAdi=>$DersDizisi) {
30     $i=count($DersDizisi);
31     $ii=$i;

```



```

32  foreach ($DersDizisi as $DersAdi) {
33      $sorgu="SELECT * FROM vwDersBolumGrup WHERE
          LTRIM(STR(YariYil)) + '-' +ltrim(rtrim(BolumNo))+'-'+DersKodu like '$DersAdi'";
34      $sonuc=mssql_query($sorgu,$link);
35      $grupNo=mssql_result($sonuc,0,"GrupNo");
36      if (!empty($grupNo)) {
37          $sorgu="SELECT * FROM vwDersBolumGrup WHERE GrupNo like '$grupNo'";
38          $sonuc=mssql_query($sorgu,$link);
39          while ($satir=mssql_fetch_array($sonuc)) {
40              $TarihGrup[$TarihGrupAdi][i-$ii] =
                  $satir["YariYil"].".".$satir["BolumNo"].".".$satir["DersKodu"];
41              $i++;
42          }
43      }
44  }
45  }

```

29-45 arası kodun kaba kod olarak açıklaması

- foreach döngüsü ile TarihGrup dizisi içinde dolaş, dizinin birinci boyutun indisi olan Tarih grubunun adını \$TarihGrupAdi değişkeninde tut. (29. satır)
- TarihGrup dizisinin birinci boyutunun değerini \$DersDizisi değişkeninde sakla.
- foreach döngüsü ile \$DersDizisi içinde dolaş, dizinin değerini \$DersAdi değişkeninde tut.(32. satır)
- \$DersAdi değişkeni içindeki dersin, başka bir ders ile aynı anda olma koşulunun var olup olmadığını tespit eden sorguyu yaz. (33-35. satırlar)
- \$DersAdi değişkeni içindeki dersin , başka bir ders ile aynı anda olma koşulunun varlığı tespit edilmiş ise (36. satır), bu derslerin hepsine aynı TarihGrup numarasını ver. (37-43 arası satırlar)

2.2.2. Aynı Anda Olma Koşulunun Sağlanması

Aynı öğretim elemanı tarafında verilen, aynı kodlu yani müfredatı aynı olan derslerin sınavı aynı gün ve saatte olması gerekir. Aynı anda olma koşulunu sağlamak için, aynı anda olan dersler bir grup içinde toplanır. Yerleştirme ve renklendirme yapılırken dersin kodu yerine GrupNo kullanılır. Her grup bir dersmiş gibi yerleştirme yapılır. Yerleştirme yapıldıktan sonra GrupNo yerine DersKodu yazılarak işlem yapılır.

İki farklı gruplandırma söz konusudur. Birincisi önceki bölümde ki TarihGrup, ikincisi aynı anda olma grubudur.

2.2.3. Derslerin Çakışmasını Önlemek

Derslerin renk tespiti ve yerleştirme için, aşağıdaki işlemler takip edilir.

Adım 1. Sıralama yapmak ve bir dersin bulunduğu grubun tespiti için Transact-SQL komutları kullanılmıştır. SQL komutu sınav programı için girilen verilere bağlı olarak yazılım tarafından oluşturulmaktadır.

Adım 2. Ders bir tarih grubu içinde olup olmadığı tespit edilir. Tarih grubu içinde ise yerleştirmede DersKodu yerine TarihGrubu kullanılır.

Adım 3. Ders bir tarih grubu içinde değil ise aynı anda olma grubu içinde olup olmadığı tespit edilir. Bu grup içinde ise yerleştirmede aynı anda olma grubu numarası DersKodu yerine kullanılır.

Adım 4. Hem tarih grubu içinde değil, hemde aynı anda olma grubu içinde değil ise DersKodu yerleştirmede kullanılır.

Yazılımımızda bu işlem 46 – 62 satırlar arasında oluşturulan SQL komutu ile gerçekleştirilir.

- 46 – 54 arası satırlar \$TarihGrup dizisi boş değilse çalışır.
- 55 – 57 arası satırla \$TarihGrup dizisi boş ise çalışır.
- 63 – 64. satırlarda daha sonra kullanmak için bir view oluşturulmaktadır.

```
46 $s="";
47 foreach ($TarihGrup as $TarihGrupAdi=>$DersDizisi) {
48     $s=$s."WHEN LTRIM(STR(YariYil)) + ' ' +ltrim(rtrim(BolumNo))+'-'+DersKodu in (";
49     $s1="";
50     foreach ($DersDizisi as $DersAdi) {
51         $s1=$s1.", ".$DersAdi."";
52     }
53     $s=$s.substr($s1,1.") THEN '$TarihGrupAdi' ";
54 }

55 if (trim($s)!="") {
56     $s=" CASE ".$s." WHEN GrupNo IS NOT NULL THEN GrupNo ELSE
                    ltrim(YariYil)+' ' +ltrim(rtrim(BolumNo))+'-'+DersKodu END ";
57 }
```

```

58 if ($s=="") {
59     $sorgu="SELECT DISTINCT BolumKodu, ltrim(rtrim(BolumNo)) as BolumNo,
           isnull(GrupNo,ltrim(YariYil)+'.'+ltrim(rtrim(BolumNo))+'-'+DersKodu) as DersKodu
                                           FROM vwOgrenciDersler";
60 } else {
61     $sorgu="SELECT BolumKodu, ltrim(rtrim(BolumNo)) as BolumNo, $s as DersKodu
                                           FROM vwOgrenciDersler";
62 }
63 $sorgu="CREATE VIEW vw".$PHPSESSID." AS ".$sorgu;
64 $sonuc=mssql_query($sorgu, $link);

```

46-62 satırlar arasındaki kodlar SQL sorgusu oluşturmaktadır. SELECT deyimi içinde çoğu programlama dillerinde bulunan CASE WHEN.... THEN END yapısı kullanılmıştır. Ms SqlServer’da CASE deyiminin genel yazım şekli aşağıdadır.

```

CASE
  WHEN Boolean_expression THEN result_expression
  [ ...n ]
  [
  ELSE else_result_expression
  ]
END

```

Bu sorgu sayesinde bir ders TarihGrup içinde ise DersKodu olarak o tarih grubu atanır. Eğer ders tarih grubu içinde değil ancak aynı anda olacak ders grubunda ise DersKodu olarak GrupNo atanır. Her iki durum sözkonusu olmaz ise DersKodu kullanılır.

59. ve 61. satırlardaki SQL sorgusu TarihGrubunun ve içindeki ders sayısına göre dinamik olarak değişmektedir. Örneğin, 61. satırdan aşağıdaki SQL sorgusu çıkar.

```

SELECT BolumKodu, ltrim(rtrim(BolumNo)) as BolumNo,
CASE
  WHEN LTRIM(STR(YariYil)) + '.' +ltrim(rtrim(BolumNo))+'-'+DersKodu IN
                                           ('4.701-BMYBS262') THEN 'TarihGrup0'
  WHEN LTRIM(STR(YariYil)) + '.' +ltrim(rtrim(BolumNo))+'-'+DersKodu IN
                                           ('4.701-BMYBS266','4.717-BMYBS260') THEN 'TarihGrup1'
  WHEN GrupNo IS NOT NULL THEN GrupNo
  ELSE ltrim(YariYil)+'.'+ltrim(rtrim(BolumNo))+'-'+DersKodu END AS DersKodu
FROM vwOgrenciDersler

```

Yukarıda vwOgrenciDersler tablosu sorgulanmakta ve kayıtlarda DersKodu'nun değeri; LTRIM(STR(YariYil)) + '.' +ltrim(rtrim(BolumNo))+'-'+DersKodu değeri ('4.701-BMYBS262') içinde ise 'TarihGrup0' değeri, ('4.701-BMYBS266', '4.717-BMYBS260') içinde ise 'TarihGrup1' değerini almaktadır. Birinci ve ikinci WHEN deyimi içindeki mantıksal işlem yanlış ise üçüncü WHEN deyimindeki mantıksal işlem yapılmaktadır. Bu kayıttaki GrupNo değeri boş değil ise DersKodu'nun değeri, bu kayıttaki GrupNo dur. Buda yanlış değer ise ELSE deyimi içindeki değer DersKodu olmaktadır.

CASE deyimi içindeki WHEN ... THEN ... sayısı TarihGrup sayısına bağlı olarak değişmektedir.

2.2.4. Bir Sonraki Gün Olması Gereken Sınavlar

Sınav takvimi içinde alttan dersi olmayan, veya birinci sınıf öğrencileri, sınav takvimi uygun ise aynı gün sadece bir sınava girmelidir. Veritabanındaki kayıtlar bu amacı sağlayacak düzende olmalıdır.

Bir sonraki gün olacak derslerle ilgili bilgiler aşağıda oluşturulan view ile sağlanır.

```
65 if ($s=="") {
66     $sorgu="SELECT DISTINCT YariYil, ltrim(rtrim(BolumNo)) as BolumNo,
           isnull(GrupNo,ltrim(YariYil)+'.'+ltrim(rtrim(BolumNo))+'-'+DersKodu) as DersKodu
           FROM vwDersBolumGrup";
67 } else {
68     $sorgu="SELECT YariYil, ltrim(rtrim(BolumNo)) as BolumNo, $s as DersKodu
           FROM vwDersBolumGrup";
69 }

70 $sorgu="CREATE VIEW vwBSGO"'.$PHPSESSID.'" AS " '.$sorgu;
71 $sonuc=mssql_query($sorgu, $link);
```

65-71 arası satırlarda “bir sonraki gün olacak” dersler ile ilgili bilginin bulunduğu view oluşturmaktadır. Aynı anda birden fazla kişi bu modülü çalıştırdığında çakışma olmaması için view adı içinde PHP nin session numarası kullanılmıştır.

2.2.5. Matrislerin Oluşturulması

2.2.5.1. Aynı Anda Olmayacak Derslerin Matrisi

Bir öğrencinin sınavının aynı anda olması önlenmelidir. Ayrıca farklı iki sınav aynı anda, aynı dersliği talep etmemelidir. Örneğin Bilgisayar dersi sınavları laboratuarda yapılmaktadır. İki tane bilgisayar dersi sınavı aynı gün aynı saatte olmamalıdır.

2.2.5.1.1. Öğrencilerin Derslerinin Çakışmasını Önlemek için

Aynı anda olmayacak dersler için bilgi, Bölüm 2.3’de oluşturulan view’den alınacaktır. Bu view’daki tüm kayıtlar BolumKodu alanına göre sıralı olacaktır. Matrisin oluşma işlem sırası aşağıdadır.

- Bolum 2.3 deki 63 – 64. satırlarda oluşturulan view’i BolumKodu’na göre sırala
- İlk kayıttaki BölümNo değerini oku
- While kayıtsonuna gelindimi
 - BolumNo ve DersKodu’nu oku
 - if şuanki bölümNo ile önceki kayıttaki bölümNo farkımı
 - Evet farklı
 - foreach (\$aao as \$bd)
 - foreach (\$aao as \$Bd)
 - if \$bd != \$Bd ise AAO[\$Bd][\$bd]=1
 - \$aao dizisini boşalt
 - Önceki BolumNo değerine bu kayıttaki bölüm noyu ata
 - \$i sayacını sıfırla
 - \$aao[\$i] dizisine DersKodu’nu ata
 - foreach (\$aao as \$bd)
 - foreach (\$aao as \$Bd)
 - if \$bd != \$Bd ise AAO[\$Bd][\$bd]=1

Bu kaba kodun PHP Kod Karşılığı Tablo 2-6 da dır.

Tablo 16. PHP dilinde aynı anda olmayacak ders dizisinin oluşumu

```
72 $i=0;
73 $aao=array();
74 while ($satir=mssql_fetch_array($sonuc)) {
75     $bolum=$satir["BolumKodu"];
```

```

76     $derskodu=$satir["DersKodu"];
77     if($bolum != $Bolum) {
78         foreach ($sao as $bd) {
79             foreach ($sao as $Bd) {
80                 if($bd != $Bd) $AAO[$bd][$Bd]=1;
81             }
82         }
83         unset($sao);
84         $Bolum=$bolum;
85         $i=0;
86     }
87     $sao[$i] = $derskodu;
88     $i++;
89 }

90 foreach ($sao as $bd) {
91     foreach ($sao as $Bd) {
92         if($bd != $Bd) $AAO[$bd][$Bd]=1;
93     }
94 }

```

2.2.5.1.2. Dersliklerdeki sınavın çakışmasını önlemek için

Bölüm 2.5.1.1' deki algoritma kullanılır. Laboratuvar gibi sınav salonları da bir öğrenci gibi düşünülerek aynı anda olmayacak dersler dizisi oluşturulur.

Tablo 17. Sınav Salonlarındaki Derslerin Çakışmasını Önleyen Kod

```

95     $sorgu="SELECT SalonNo,
           ISNULL((SELECT GrupNo FROM tblGrupDers gd WHERE gd.SiraNo=dk.SiraNo),dk.YBDersKodu)
           as YBDersKodu FROM vwDerslikKisiti dk ORDER BY SalonNo";

96     $sonuc=mssql_query($sorgu,$link);
97     $SalonNo=mssql_result($sonuc,0,"SalonNo");
98     $i=0;
99
100    $sao = array();
101    while ($satir = mssql_fetch_array($sonuc)) {
102        $salonNo=$satir['SalonNo'];
103        $derskodu=$satir['YBDersKodu'];
104        if($SalonNo != $salonNo) {

```

```

105     foreach ($sao as $bd) {
106         foreach ($sao as $bd) {
107             if($bd != $Bd) $AAO[$bd][$Bd]=1;
108         }
109     }
110     unset($sao);
111     $SalonNo=$salonNo;
112     $i=0;
113 }
114 $sao[$i]=$derskodu;
115 $i++;
116 }

117 foreach ($sao as $bd) {
118     foreach ($sao as $Bd) {
119         if($bd != $Bd) $AAO[$bd][$Bd]=1;
120     }
121 }

```

vwDerslikKisiti view'inde sadece sınav salon kısıtı olan derslerin bilgileri vardır. vwDerslikKisiti view'ndaki kayıtlara 2.5.1.1'deki işlemler yapılır. 95 – 121 arasındaki satırlarda sınav salonu kısıtı olan dersler için aynı anda olmayacak matrisine değer atamaktadır.

\$AAO aynı anda olmayacak derslerin matrisidir. Değeri null ise aynı anda olabilir, değeri 1 ise aynı anda olamaz.

Tablo 18. vwDerslikKisiti view'inin yapısı

```

CREATE VIEW vwDerslikKisiti AS
SELECT db.DersKodu, db.BolumNo, db.YariYil, db.SicilNo, db.OgrenciSayisi,
db.OturumSayisi, db.DerslikKisitiVar, db.OturumKisitiVar, db.SiraNo,
dk.SalonNo,
LTRIM(db.YariYil) + '.' + LTRIM(RTRIM(db.BolumNo)) + '-' + db.DersKodu
AS YBDersKodu
FROM tblDersBolum db INNER JOIN tblDerslikKisiti dk ON db.SiraNo =
dk.SiraNo WHERE (db.DerslikKisitiVar = 1) ORDER BY dk.SalonNo

```

2.2.5.1.3. Her Ders İçin Aynı Anda Olmayacak Diğer Dersler Dizisi

\$AAO dizisi 2 boyutlu ve değeri 1 veya NULL'dur. Her ders için aynı anda olmayacak diğer derslerin kodlarını \$AAODersler dizisinde tutacağız. Bu dizi oluştuktan sonra \$AAO dizisinin işi bitmektedir. Bu nedenle \$AAO dizisini yok edeceğiz.

Tablo 19. AAODersler dizisinin oluşumu

```
122 foreach ($AAO as $a=> $d) {
123     foreach ($d as $dd=>$D) {
124         if (!empty($AAO[$a][$dd])) $AAODersler[$a][$dd]=$Dersler[$dd];
125     }
126     arsort($AAODersler[$a]);
127     echo "</font>";
128 }
129 unset ($AAO);
```

2.2.5.2. Bir Sonraki Gün Olacak Dersler Matrisinin Oluşturulması

Eğer sınav takvimi uygun ise, alttan dersi olmayan öğrenci aynı gün sadece bir sınava girmelidir. Yani aynı yarıyıl dersinin hergün bir tane olmalıdır. Bu nedenle bir sonraki gün olacak dizisi (\$BSGO) oluşturuldu. BSGO iki boyutlu dizidir. İlişki var ise değeri 1, yok ise null' dur.

Kaba kodu

- Bolum 2.3 deki 70 – 71. satırlarda oluşturulan view'i BolumNo ve Yarıyıl'a göre sırala
- İlk kayıttaki BölümNo değerini oku
- While kayıtsonuna gelindimi
 - BolumNo ve DersKodu'nu oku
 - if şuanki bölümNo ile önceki kayıttaki bölümNo farkımı
 - Evet farklı
 - foreach (\$sao as \$bd)
 - foreach (\$sao as \$Bd)
 - if \$bd != \$Bd ise AAO[\$Bd][\$bd]=1
 - \$sao dizisini boşalt
 - Önceki BolumNo değerine bu kayıttaki bölüm noyu ata
 - \$i sayacını sıfırla

- \$aao[\$i] dizisine DersKodu'nu ata
- foreach (\$aao as \$bd)
 - foreach (\$aao as \$Bd)
 - if \$bd != \$Bd ise AAO[\$Bd][\$bd]=1

Tablo 20. Bir sonraki gün olacak dersler'in kodu

```

130 $sorgu="SELECT * FROM vwBSGO.".$PHPSESSID." ORDER BY BolumNo, YariYil";

131 $sonuc=mssql_query($sorgu,$link);
132 $Bolum=mssql_result($sonuc,0,"YariYil")."."mssql_result($sonuc,0,"BolumNo");

133 $aao=array();
134 while ($satir=mssql_fetch_array($sonuc)) {
135     $bolum=$satir["YariYil"].".".$satir["BolumNo"];
136     $derskodu=$satir["DersKodu"];
137     if($bolum != $Bolum) {
138         foreach ($aao as $bd) {
139             foreach ($aao as $Bd) {
140                 if($bd != $Bd) $BSGO[$bd][$Bd]=1;
141             }
142         }

143         unset($aao);
144         $Bolum=$bolum;
145     }
146     $aao[ ] = $derskodu;
147 }

148 foreach ($aao as $bd) {
149     foreach ($aao as $Bd) {
150         if($bd != $Bd) $BSGO[$bd][$Bd]=1;
151     }
152 }

```

2.2.5.2.1. Her Ders İçin Aynı Anda Olmayacak Diğer Dersler Dizisi

\$BSGO dizisi 2 boyutlu ve değeri 1 veya null'dur. Her ders için bir sonraki gün olacak diğer derslerin kodlarını \$BSGODersler dizisinde tutacağız. Bu dizi oluştuktan sonra \$BSGO dizisinin işi bitmektedir. Bu nedenle \$BSGO dizisini yok edeceğiz.

Tablo 21. Bir sonraki gün olacak derslerin dizi oluşumu

```
153 $BsgoDersler=array();
154 foreach ($BSGO as $b=>$deger) {
155     $stop=0;
156     foreach ($deger as $a=>$Deger) {
157         $stop+=$Deger;
158     }
159     $BsgoDersler[$b]=$stop;
160 }

161 foreach ($BSGO as $a=> $d) {
162     foreach ($d as $dd=>$D) {
163         if (!empty($BSGO[$a][$dd])) $BSGODersler[$a][$dd]=$Dersler[$dd];
164     }
165     arsort($BSGODersler[$a]);
166 }
167 unset ($BSGO);
```

2.2.5.3. Bir Sonraki Oturumda Olmayacak Dersler Matrisi Oluşturma

Bir öğrenci aynı gün ard arda sınava girmemeli arada en az bir oturum boşluk olmalıdır. Bunu sağlamak için bir sonraki oturumda olmayacak (BSGO) dizisi oluşturulmaktadır.

Kaba kodu

- Bolum 2.3 deki 63 – 64. satırlarda oluşturulan view'i BolumKodu'na göre sırala
- İlk kayıttaki BölümNo değerini oku
- While kayıtsonuna gelindimi
 - BolumNo ve DersKodu'nu oku
 - if şuanki bölümNo ile önceki kayıttaki bölümNo farkımı
 - Evet farklı
 - foreach (\$aao as \$bd)
 - foreach (\$aao as \$Bd)

- if \$bd != \$Bd ise AAO[\$Bd][\$bd]=1
 - \$sao dizisini boşalt
 - Önceki BolumNo değerine bu kayıttaki bölüm noyu ata
 - \$i sayacını sıfırla
- \$sao[\$i] dizisine DersKodu'nu ata
- foreach (\$sao as \$bd)
 - foreach (\$sao as \$Bd)
 - if \$bd != \$Bd ise AAO[\$Bd][\$bd]=1

Bu kaba kodun PHP karşılığı aşağıdadır.

```

168 if ($s=="") {
169     $sorgu="SELECT DISTINCT BolumKodu, ltrim(rtrim(BolumNo)) as BolumNo,
            isnull(GrupNo,ltrim(YariYil)+'.'+ltrim(rtrim(BolumNo))+'-'+DersKodu) as DersKodu FROM
            vwOgrenciDersler Order By BolumKodu";
170 } else {
            $sorgu="SELECT BolumKodu, ltrim(rtrim(BolumNo)) as BolumNo, $s as DersKodu FROM
            vwOgrenciDersler ORDER BY BolumKodu";
171 }

172 $sonuc=mssql_query($sorgu,$link);
173 $Bolum=mssql_result($sonuc,0,"BolumKodu");
174 $i=0;

175 $sao=array();
176 while ($satir=mssql_fetch_array($sonuc)) {
177     $bolum=$satir["BolumKodu"];
178     $derskodu=$satir["DersKodu"];
179     if($bolum != $Bolum) {
180         foreach ($sao as $bd) {
181             foreach ($sao as $Bd) {
182                 if($bd != $Bd) $BSOO[$bd][$Bd]=1;
183             }
184         }

185     unset($sao);
186     $Bolum=$bolum;
187     $i=0;

```

```

188     }
189     $aao[$i] = $derskodu;
190     $i++;
191 }
192 foreach ($aao as $bd) {
193     foreach ($aao as $Bd) {
194         if($bd != $Bd) $BSSOO[$bd][$Bd]=1;
195     }
196 }

```

2.2.5.3.1. Her Ders İçin Bir Sonraki Oturumda Olmayacak Diğer Dersler Dizisi

\$BSSOO dizisi 2 boyutlu ve değeri 1 veya null'dur. Her ders için bir sonraki oturumda olmayacak diğer derslerin kodlarını \$BSSOODersler dizisinde tutacağız. Bu dizi oluşturduktan sonra \$BSSOO dizisinin işi bitmektedir. Bu nedenle \$BSSOO dizisini yok edeceğiz.

Tablo 22. Bir sonraki oturumda olmayacak derslerin dizi oluşumu

```

197 foreach ($BSSOO as $a=> $d) {
198     foreach ($d as $dd=>$D) {
199         if (!empty($BSSOO[$a][$dd])) $BSSOODersler[$a][$dd]=$Dersler[$dd];
200     }
201     arsort($BSSOODersler[$a]);
202 }
203 unset ($BSSOO);

```

2.2.6. Renklendirme

Renklendirmede Aynı Anda Olmayacak Dersler dizisi \$AAO 'ya göre yapılır. \$AAO daki her düğümün derecesi aşağıdaki gibi hesaplanır.

2.2.6.1. Düğümün derecesinin hesabı

Her ders düğümünün derecesi \$AAO matrisi üzerinde, her ders için yataydaki 1'lerin toplamıdır. PHP 'de dizi indisi sayı olmak zorunda değildir. Bu nedenle \$Dersler dizisinin indisi DersKodu'dur. 9. satırdaki arsort fonksiyonu içinde yer alan diziyi değerine göre tersten sıralamaktadır. Yani \$Dersler dizisinde derecesi en büyük olan 1. sırada, en küçük olan son sırada yer almaktadır.

10. satırda yer alan \$keys dizisinde \$Dersler dizisinin anahtar değerleri yani derece sırasına göre ders kodları bulunmaktadır.

```
1 $Dersler=array();
2 foreach ($AAO as $b=>$deger) {
3     $stop=0;
4     foreach ($deger as $a=>$Deger) {
5         $stop+=$Deger;
6     }
7     $Dersler[$b]=$stop;
8 }
9 arsort($Dersler);
10 $keys=array_keys($Dersler);
```

2.2.6.2. Renk Numaralarının Verilmesi

Renk numarası, \$keys dizisinin her bir elemanına numara atanarak verilmektedir. Renk atamasında işlem sırası aşağıdadır.

- İçi boş bir \$renkler dizisi oluştur.
- \$renkno =0
- While \$keys dizi eleman sayısı 0 oluncaya kadar dön.
 - \$renkno değerini bir arttır.
 - \$renkler[\$keys] değerine \$renkno'yu ata
 - \$B değişkenine \$keys dizisinin ilk elemanını ata
 - \$k=0
 - Geçici \$renk dizisini boş olarak oluştur
 - \$renk[\$k]=\$B atamasını yap
 - For (\$i=1; \$i< count(\$keys);\$i++)
 - \$m=1
 - for (\$j=0; \$j<count(\$renk); \$j++)
 - if (!empty(\$BAO[\$keys[\$i]][\$renk[\$j]])) {\$m=0; break;}
 - if (\$m= =1) {
 - \$renkler[\$keys[\$i]]=\$renkno
 - \$k++

- \$renk[\$k] = \$keys[\$i];
- \$i=0
- foreach (\$renk as \$renks) {
 - \$Renkler["renk".\$renkno][\$i]=\$renks;
 - \$i++;
- \$keys1=array_diff(\$keys,\$renk);
- \$keys=array_values(\$keys1);

Bu kaba kodun PHP’de yazılmış hali

```

1  $renkler=array();
2  $renkno=0;

3  while (count($keys)>0) {
4      $renkno++;
5      $renkler[$keys[0]] = $renkno;
6      $B=$keys[0];

7      $k=0;
8      unset($renk);
9      $renk=array();
10     $renk[$k] = $B;
11     for ($i=1; $i<count($keys);$i++) {
12         $m = 1;
13         for ($j=0; $j<count($renk); $j++) {
14             if (!empty($BAO[$keys[$i]][$renk[$j]])) {$m=0; break;}
15         }
16         if ($m==1) {
17             $renkler[$keys[$i]]=$renkno;
18             $k++;
19             $renk[$k] = $keys[$i];
20         }
21     }

22     $i=0;
23     foreach ($renk as $renks) {
24         $Renkler["renk".$renkno][$i]=$renks;
25         $i++;

```

```
26     }
27     $keys1=array_diff($keys,$renk);
28     $keys=array_values($keys1);
29 }
```

2.2.7. Yerleştirme

Renk sayısı toplam oturum sayısından az olmaktadır. Bulunan renk numaralarını oturum numarası olarak atamak yanlıştır. Bölüm 2.5 de elde edilen dizilere uygun olarak yerleştirme yapılmalıdır. Bu diziler;

- Aynı anda olmayacak dersler Dizisi (AAO)
- Bir sonraki oturumda olmayacak dersler dizisi(BSOO)
- Bir sonraki gün olacak dersler dizisi (BSGO)
- Tarih kısıtı olanlarada oturumunun atanması gereklidir.

Yerleştirmenin yukarıdaki dizilere uyumlu olup olmadığı ile ilgili fonksiyonlar yazılmıştır. Bu fonksiyonlarda sonuç TRUE (DOĞRU) ise yerleştirme uygun, FALSE (YANLIŞ) ise yerleştirme uygundeğildir.

Oturum no kullanımı: Sınavın başladığı ilk oturuma bir değeri verilmiş ve sırası ile her oturuma bir artan numara verilmiştir. Akademik takvimde sınav 10 gün olarak düşünülmüş ve günde 4 tane sınav yapıldığı düşünülürse oturum numarası 1 den 40'a kadar sayıdır ve ilk oturumdan itibaren sırası ile her oturuma artan numara verilir.

Sırası ile bu dizileri kontrol eden fonksiyonların kaba kodu ve PHP kodunu yazalım.

2.2.7.1. AAO Kontrol Fonksiyonu

Bu fonksiyon yerleştirilmek istenen dersin başka bir dersle çakışıp çakışmadığını kontrol eder.

- Fonksiyonun parametreleri : \$AAODersler, \$DersKodu, \$OturumNo,\$Oturumlar
 - \$AAODersler : Aynı anda olmayacak dersler dizisi
 - \$DersKodu : Kontrol edilecek dersin kodu
 - \$OturumNo : kontrol edilecek oturumun numarası
 - \$Oturumlar : Daha önce atanmış oturum numaraları
- \$sonuc = TRUE

- Eğer \$Oturumlar["renk".\$OturumNo] bir dizi mi? Cevap evet ise
 - \$k=array_intersect(\$Oturumlar["renk".\$OturumNo], array_keys(\$AAODersler[\$DersKodu]));
 - if (count(\$k)>0) \$sonuc=FALSE;
- return \$sonuc

PHP Kodu ;

```

1 function AAOKontrol($AAODersler, $DersKodu, $OturumNo,$Oturumlar) {
2     $sonuc = TRUE;
3     if (is_array($Renkler["renk".$OturumNo])) {
4         $k=array_intersect($Oturumlar["renk".$OturumNo],
5                             array_keys($AAODersler[$DersKodu]));
6         if (count($k)>0) $sonuc=FALSE;
7     }
8     return $sonuc;
9 }

```

3. satırdaki is_array fonksiyonu, içindeki değişkenin bir dizi olup olmadığını kontrol etmektedir. 4.satırdaki array_intersect fonksiyonu iki diziyi karşılaştırıp aynı elemanları döndürür. array_keys fonksiyonu ise dizinin anahtarlarını döndürür.

4. satırda \$k dizisinin elemanı varsa sonuç uygun değildir.

2.2.7.2. BSOO Kontrol fonksiyonu

Bu fonksiyon bir dersin başka atanmış başka derslerle BSOO kontrolü yapar.

```

1 function BSOOKontrol($BSOODersler, $DersKodu, $OturumNo,$Oturumlar,$DersOturumSayisi) {
2     $sonuc = TRUE;
3     $gun=floor( $OturumNo/4)+1;
4     if (($OturumNo%4)==0) $gun--;
5
6     if (($OturumNo % 4) != 1 && ($BSOODersler[$DersKodu])) {
7         $OturumSiraNo=$OturumNo-1;
8         if (is_array($Oturumlar["renk".$OturumSiraNo])) {
9             foreach($Renkler["renk".$OturumSiraNo] as $d) {

```



```

9             if ($DersOturumSayisi[$d]==3) $sonuc=FALSE;
10            }
11            $k=array_intersect ($Oturumlar["renk".$OturumSiraNo],
                                array_keys($BSOODersler[$DersKodu]));
12            if (count($k)>0) { $sonuc=FALSE; }
13        }
14    }

15    if (($OturumNo % 4) != 0) { // Gün içinde 4. oturum ise
16        $OturumSiraNo=$OturumNo+1;
17        if (is_array($Oturumlar ["renk".$OturumSiraNo]) &&
                                is_array($BSOODersler[$DersKodu])) {
18            $k=array_intersect ($Renkler["renk".$OturumSiraNo],
                                array_keys($BSOODersler[$DersKodu]));
19            if (count($k)>0) { $sonuc=FALSE; }
20        }
21    }
22    return $sonuc;
23 }

```

2.2.7.3. BSGO Kontrol fonksiyonu

Bu fonksiyon bir dersin başka atanmış başka derslerle BSGO kontrolü yapar.

```

1 function BSGOKontrol ($BSGODersler, $DersKodu, $OturumNo,$Oturumlar) {
2     $sonuc = TRUE;
3     $gun=floor( $OturumNo/4)+1;
4     if (($OturumNo%4)==0) $gun--;
5     for ($i=1; $i<=4; $i++) {
6         $OturumSiraNo=( $gun - 1) * 4 + $i;
7         if (is_array($Oturumlar["renk".$OturumSiraNo])) {
8             if (is_array($Renkler["renk".$OturumSiraNo])) {
9
10                $k=array_intersect($Oturumlar["renk".$OturumSiraNo],
                                    array_keys($BSGODersler[$DersKodu]));
11                if (count($k)>0) { $sonuc=FALSE; break;}
12            }
13        }
14    return $sonuc;

```

15 }

2.2.7.4. Oturum Numarası Atama Fonksiyonu

Bu fonksiyon bir derse bir oturum numarası atar.

```
1 function BosRenkBul($AAODersler, $BSGODersler, $BSOODersler, $DersKodu, $Renkler,
   $ToplamOturumSayisi, $BirOturumdaSinifSayisi,$DersSinifSayisi,$DersOturumSayisi){
   global $RenkSinifSayisi;
2   $RenkNo=0;
3   for ($i=1; $i<=$ToplamOturumSayisi; $i++) {
4       if ($DersOturumSayisi[$DersKodu]==1 ||
           ($DersOturumSayisi[$DersKodu]==2 && ($i%4)>0) ||
           ($DersOturumSayisi[$DersKodu]==3 && (($i%4)==1 ||($i%4)==2)) )
5           if (AAOKontrol($AAODersler, $DersKodu, $i,$Renkler) &&
               BSGOKontrol($BSGODersler, $DersKodu, $i,$Renkler) &&
               BSOOKontrol($BSOODersler, $DersKodu, $i,$Renkler,$DersOturumSayisi) )
6               {
7                   if (empty($RenkSinifSayisi[$i])) {
8                       $RenkNo = $i;
9                       $RenkSinifSayisi[$i]=$DersSinifSayisi[$DersKodu];
10                      break;
11                      } else {
12                          if($RenkSinifSayisi[$i]+$DersSinifSayisi[$DersKodu]<=$BirOturumdaSinifSayisi) {
13                              $RenkNo=$i;
14                              $RenkSinifSayisi[$i]+=$DersSinifSayisi[$DersKodu];
15                              break;
16                              } //iç if
17                          } //if
18                      } // for
19
20   return $RenkNo;
21 }
```

3. YAZILIM VE ÖRNEK KULLANIM

3.1. Ana Menü

Yapılan çalışma bir internet yazılımıdır. Bu yazılımda yapılacak kayıtlar ve listeler şekil 3 deki menü ile kontrol edilmektedir. Her menü elemanı seçildiğinde adı geçen işle ilgili listeleme, yeni kayıt ekleme, değiştirme ve silme işlemleri gerçekleştirilir.

Menüden kontrol edilen işler için yeniden kod yazılmadı, bir parametre kodu oluşturuldu. Her bir menü elemanı için parametre tanımlandı.

- Ders Bölüm
- Bölümler
- Ders Kayıt
- Öğretim
- Elemanları
- Ders Grupları
- Sınav Salonu
- Kayıt
- Sınav Programı

Şekil 3. Ana Menü

3.1.1. Tüm menü elemanlarının ortak kodları

Tüm menü elemanlarında aşağıdaki kod çalıştırılır. Her menü elemanı için sadece 2. satırda çağrılan PHP dosyası değişmektedir. Bu PHP dosyasında parametre bilgileri bulunmaktadır.

```
<?
1      include("connect.php");
2      include("listeleme-sorgu-sinavsalonukayit.php");
3      include("islemler.php"); ?>
4 <h1>Sýnav Salonları</h1>
<?
5      include("listeleme.php");
6      include("close.php");
?>
```

include () fonksiyonu ile çağrılan dosyaları açıklayalım.

3.1.1.1. connect.php dosyası

Bu dosyadaki kodlar ile SQL_Server veri tabanına bağlantı sağlanır.

```
<?
1      $link=mssql_connect("SINAV","sinav","123456");
2      mssql_select_db("sinav",$link);
?>
```

3.1.1.2. listeleme-sorgu-sinavsalonukayit.php

Aşağıdaki kodlarda listeleme, yeni kayıt formu ile ilgili parametre bilgileri vardır.

Açıklamalar kod satırının üzerindedir.

```
<?
1      $SayfadakiKayitSayisi=50; //bir sayfada gösterilecek kayıt sayısı
// Liste alınacak sql sorgusu
2      $listesorgu ="SELECT * FROM tblSinavSalonlari ORDER BY SalonNo";
//
// listeleme kısmı
//
//listede yer alacak tablonun kolon isimleri
3      $alanlar=array("SalonNo","SinavKapasitesi","AtamaOnceligi","GozetmenSayisi");
// Değiştir durumunda ekrana gelecek textbox uzunluğu
4      $uzunluk=array(10,10,10,5);
//Kaç kolon listelenecek
5      $alansayisi=4;
//listelenen alan sayısal ise değeri 1, değilse 0
6      $sayisalmi=array(0,1,1,1);
//listelenen kolonun hücre içindeki pozisyonu
7      $align=array("center","center","center","center");

// Ekleme ve değiştirme durumunda bazı kolonlar değeri başka bir tablodan alacak mı alacaksa çalışacak
// sorgu. Var ise değeri 1 değilse 0
8      $acilanKutuvar=array(0,0,1,0);
//Açılan kutunun veri kaynağı olan sorgu
9      $acilanKutuSorgu=array("", "", "SELECT * FROM tblOncelik", "");
//Açılan kutudada yer alan alanın adı
10     $acilanKutuAlan=array("", "", "OncelikNo", "");
11     $acilanKutuDeger=array("", "", "OncelikNo", "");
```

```

//
// listedeki başka tablodan değer
//
12     $BaskaTablodanDegerVar=array(0,0,1,0);
13     $BaskaTablodanDegerSorgu=array("", "", "SELECT * FROM tblOncelik WHERE
OncelikNo=", "");
14     $BaskaTablodanDegerAlan=array("", "", "OncelikNo", "");
// veri girildiğinde tümü büyük, ilk harf büyük gibi işlemler yapmak için
15     $buyukKucukHarf=array(2,0,0,0);
/*
        0: Değişirme
        1: İlk Karakter Büyük Diğerleri küçük
        2: Tümü büyük
        3: tümü küçük
        4: Kelimelerin ilk karakteri büyük, diğerleri küçük
*/
// Yeni ekleme durumunda yer alan kolonlar veri giriş tabloları, başlıkları
16     $tablokolonbasligi=array("Salon No", "Sınav Kapasitesi", "Atama Önceliği", "Gözetmen Sayısı");
17     $tablokolonsayisi=4;
18     $tablogenisligi=475;
19     $tablokolongenisligi=array(140,80,50,50);
20     $degistirtablokolongenisligi=array(140,80,50,50);
//
//Yeni kayıt Tablosu
//
21     $yenikayitTabloGenisligi=476;//1.kolon 30 2.kolon :124
22     $yenikayitTabloBasligi="YENİ SINAV SALON KAYDI";
23     $yenikayitTabloKolonGenisligi=array(120,202);

//tablo adı
24     $tabloAdi="tblSinavSalonlari";
?>

```

3.1.1.3. islemler.php

Bu dosya ile silme değiştirme ve ekleme işlemleri gerçekleştirilmektedir. PHP kodu ve açıklaması aşağıdadır.

İşlem No	Görevi
1	Veritabanına yeni kayıt ekleme
2	Kayıt değiştirme için form oluştur
3	Veritabanındaki mevcut kaydı değiştir
4	Kayıt silme onay isteği
5	Veritabanındaki kaydı silme

<?

// 1-20 arasındaki kodlar text kutularına girilen metinleri büyük küçük harf dönüşümü yapmaktadır

```

1  if ($islem==1) {
2      if ($modul!=200) {
3          for ($k=0;$k<$alansayisi;$k++) {
4              switch ($buyukKucukHarf[$k]) {
5                  case 0:
6                      $Deger[$k]=$_POST[$alanlar[$k]];
7                      break;
8                  case 1:
9                      $Deger[$k]=ucfirst(strtolower(trim($_POST[$alanlar[$k]])));
10                     break;
11                     case 2:
12                         $Deger[$k]=strtoupper(trim($_POST[$alanlar[$k]]));
13                         break;
14                         case 3:
15                             $Deger[$k]=strtolower(trim($_POST[$alanlar[$k]]));
16                             break;
17                             case 4:
18                                 $Deger[$k]=ucwords(strtolower(trim($_POST[$alanlar[$k]])));
19                                 }
20                     }

```

// 21-38 arası kod veri tabanına ekleme sorgusu oluşturmaktadır. Bölüm 3.1.1.2 deki parametrelere göre

// işlem yapmaktadır.

```

21     $sorgu="INSERT ".$tabloAdi." (";
22     for ($k=0;$k<$alansayisi;$k++) {
23         if ($k!=$alansayisi-1) {
24             $sorgu=$sorgu.$alanlar[$k].",";
25         } else {

```

```

26             $sorgu=$sorgu.$alanlar[$k].") VALUES (";
27             }
28         }

29         for ($k=0;$k<$alansayisi;$k++) {
30             if ($k!=$alansayisi-1) {
31                 if (empty($sayisalmi[$k])) $sorgu=$sorgu."".$Deger[$k].""." ";
32                 else $sorgu=$sorgu.$Deger[$k].""." ";
33             } else {
34                 if (empty($sayisalmi[$k])) $sorgu=$sorgu."".$Deger[$k].""." ";
35                 else $sorgu=$sorgu.$Deger[$k].""." ";
36             }
37         }
38     }

```

DersBolum modülünün yapısı diğerlerinden farklı satır 21-38 arası bunun için çalıştırılmaz.

```

39     if($modul==200) {
40         $sorgu="INSERT tblDersBolum (BolumNo,DersKodu,YariYil,SicilNo) VALUES
41             ('$BolumNo', '$DersKodu', $YariYil, $SicilNo)";

```

// 21-42 arasında oluşturulan ekleme sorgunun çalıştırılması

```

42     $sonuc=@mssql_query($sorgu,$link);

43     if ($sonuc) {
44         echo "<br>Kayıt Başarılı Olarak Yapıldı<br>\n";
45     } else { echo "<br>\nAynı Ders kodundan veritabanında var<br>\n";}
46 } /* işlem 1 sonu */
/* Kayıt değiştirme işlemi için sorgu oluşturma */

47 if ($islem==3) {
/* modül =200 (DersBolum modülü ise) */

48     if ( $modul==200 ) {
// Derslik kısıtı daha önce tanımlı ise bu derse bağlı bütün kısıtları sil
// Yeni kısıtı ekle
// Derslik kısıtı ekleme

49         if ($_POST["derlikkısıtı"]=="on") {
50             $sorgu="DELETE tblDerslikKısıtı WHERE SiraNo=$sirano";
51             $sonuc=@mssql_query($sorgu,$link);
52             $sorgu="UPDATE tblDersBolum SET DerslikKısıtıVar=1 WHERE
53                 SiraNo=".$sirano;

```

```

54         foreach ($salonno as $salonNo) {
55             $sorgu="INSERT tblDerslikKisiti (SiraNo, SalonNo) VALUES
                                                    ($sirano,$salonNo)";
56             $sonuc=@mssql_query($sorgu,$link);
57         }
// Derslik kisiti yok ise daha önceki kısıtları sil
58     } else {
59         $sorgu="UPDATE tblDersBolum SET DerslikKisitiVar=0 WHERE
                                                    SiraNo=$sirano";
60         $sonuc=@mssql_query($sorgu,$link);
61         $sorgu="DELETE tblDerslikKisiti WHERE SiraNo=$sirano";
62         $sonuc=@mssql_query($sorgu,$link);
63     }
// Tarih kisiti var ise önceki kisiti sil bunun yerine yeni kisit ata
// Aynı bölüm için aynı oturumno daha önce atanmış ise bunları ekleme
64     if ($_POST["tarikhisiti"]=="on") {
65         $sorgu="DELETE tblTarihKisit WHERE BolumDersSiraNo=$sirano";
66         $sonuc=@mssql_query($sorgu,$link);
67         $sorgu="UPDATE tblDersBolum SET OturumKisitiVar=1 WHERE
                                                    SiraNo=".$sirano;
68         $sonuc=@mssql_query($sorgu,$link);
69         foreach ($oturumlar as $oturumNo) {
70             $snc=mssql_query("SELECT COUNT(*) AS SAYI FROM
71                                 vwTarihKisit where BolumNo=".$BolumNo." and
                                                    OturumSiraNo=$oturumNo", $link);
72             $snc1=mssql_fetch_array($snc);
73             if ($snc1['SAYI']==0) {
74                 $sorgu="INSERT tblTarihKisit (BolumDersSiraNo,
75                                 OturumSiraNo) VALUES ($sirano,$oturumNo)";
76                 $sonuc=@mssql_query($sorgu,$link);
77             }
78             $snc=mssql_query("SELECT COUNT(*) AS SAYI FROM vwTarihKisit
79                                 where BolumNo=".$BolumNo." and
                                                    DersKodu=".$DersKodu."", $link);
80             $snc1=mssql_fetch_array($snc);
81             if ($snc1['SAYI']==0) {
82                 $sorgu="UPDATE tblDersBolum SET OturumKisitiVar=0 WHERE
                                                    SiraNo=".$sirano;
83                 $sonuc=@mssql_query($sorgu,$link);

```



```

82         }
// Tarih kısıtı yok ise var olan önceki kısıtları sil
83     } else {
84         $sorgu="UPDATE tblDersBolum SET OturumKisitiVar=0 WHERE
                                                SiraNo=$sirano";
85         $sonuc=@mssql_query($sorgu,$link);
86         $sorgu="DELETE tblTarihKisit WHERE BolumDersSiraNo=$sirano";
87         $sonuc=@mssql_query($sorgu,$link);
88     }
//Aynı anda olma grubu ataması
89     if (empty($grupno)){
        $sonuc=@mssql_query("DELETE tblGrupDers WHERE
                                                SiraNo=$sirano",$link);
90     } else {
91         $sonuc=@mssql_query("INSERT tblGrupDers (GrupNo, SiraNo) VALUES
                                                (".$grupno.",$sirano)",$link);
92     }
// Kaydı değiştirme sorgusu
93     $sorgu="UPDATE tblDersBolum SET YariYil=".$YariYil.",BolumNo=".$BolumNo.",
94     DersKodu=".$DersKodu.", SicilNo=".$SicilNo.",
        OturumSayisi=".$OturumSayisi.", OgrenciSayisi=".$OgrenciSayisi." WHERE
        SiraNo=".$sirano;
95     $sonuc=@mssql_query($sorgu,$link);
/* Diğer tüm modüller
96     } else {
// Textboxlardan gelen bilgileri büyük küçük harf dönüşümünü yap
97     for ($k=0;$k<$alansayisi;$k++) {
98         switch ($buyukKucukHarf[$k]) {
99         case 0:
100            $Deger[$k]=$_POST[$alanlar[$k]];
101            break;
102         case 1:
103            $Deger[$k]=ucfirst(strtolower($_POST[$alanlar[$k]]));
104            break;
105         case 2:
106            $Deger[$k]=strtoupper($_POST[$alanlar[$k]]);
107            break;
108         case 3:
109            $Deger[$k]=strtolower($_POST[$alanlar[$k]]);

```

```

110             break;
111         case 4:
112             $Deger[$k]=ucwords(strtolower($_POST[$alanlar[$k]]));
113         }
114     }
// Değişirme sogusunun oluşumu ve çalıştırılması
115     $sorgu="UPDATE ".$tabloAdi." SET ";
116     for ($k=0;$k<$alansayisi;$k++) {
117         if ($k!=$alansayisi-1) {
118             if (empty($sayisalmi[$k]))
119                 $sorgu=$sorgu.$alanlar[$k]."=".$Deger[$k].",";
120             else $sorgu=$sorgu.$alanlar[$k]."=".$Deger[$k].",";
121         } else {
122             if (empty($sayisalmi[$k]))
123                 $sorgu=$sorgu.$alanlar[$k]."=".$Deger[$k]." WHERE
124                                     SiraNo=$sirano";
125             else $sorgu=$sorgu.$alanlar[$k]."=".$Deger[$k]." WHERE
126                                     SiraNo=$sirano";
127         }
128     }
129     $sonuc=@mysql_query($sorgu,$link);
130     if ($sonuc) {
131         echo "<br>Kayit Başarılı Olarak Değiştirildi<br>\n";
132     } else { echo "<br>\nKayit Degistirilemedi<br>\n";}
133 }
// Kayıt silme sogusunun oluşumu
134 if ($islem==5) {
135     $sorgu="DELETE ".$tabloAdi." WHERE SiraNo=$sirano";
136     $sonuc=@mysql_query($sorgu,$link);
137     if ($sonuc) {
138         echo "<br>Kayit Başarılı Silindi<br>\n";
139     } else { echo "<br>\nKayit Silinemedi<br>\n";}
140 }
?>

```

3.1.2. Örnek Kullanım

3.1.2.1. Ders Bölüm

Programa 17 bölüm, 158 ders, 88 öğretim elemanı, 37 aynı anda olma ders grubu 46 sınav salonu kaydedilmiştir. Bölümlere toplam 263 ders, öğretim elemanı bağlantısı yapılmıştır.

| Sıra No | İşlem | Bölüm Adı | Ders Kodu - Adı | Derslik Kıstı | Tarih Kıstı | Grup No | Öğrenci Sayısı | Oturum Sayısı |
|---------|--------------|----------------------------|----------------------------------|---------------|-------------|-------------|----------------|---------------|
| 1 | Değiştir Sil | Bankacılık ve Sigortacılık | BMYBS214 - Sigorta İşlemleri 2 | | | 36 | 1 | |
| 4 | | Öğr.Gör. Berna AKSOY | | | | MYDIL-1 | | |
| 2 | Değiştir Sil | Bankacılık ve Sigortacılık | BMYBS252 - Mesleki Yabancı Dil 2 | | | 40 | 1 | |
| 4 | | Öğr.Gör. Açelya DOĞRULU | | | | | | |
| 3 | Değiştir Sil | Bankacılık ve Sigortacılık | BMYBS262 - Banka İşlemleri 2 | | | 41 | 1 | |
| 4 | | Öğr.Gör. Alptekin GÜNEY | | | | | | |
| 4 | Değiştir Sil | Bankacılık ve Sigortacılık | BMYBS264 - Kredi İşlemleri | | | 35 | 1 | |
| 4 | | Öğr.Gör. Ümit HACIOĞLU | | | | BMYBS266-AG | | |
| 5 | Değiştir Sil | Bankacılık ve Sigortacılık | BMYBS266 - Kambiyo Bilgisi | | | 31 | 1 | |
| 4 | | Öğr.Gör. Alptekin GÜNEY | | | | | | |
| 6 | Değiştir Sil | Bankacılık ve Sigortacılık | BMYBS268 - Sermaye Piyasaları | | | 35 | 1 | |
| 4 | | Öğr.Gör. Bülent HOCA | | | | | | |

Şekil 4. Bölümlerin ders listesi görünümü

Şekil-4'deki listedeki bölüm adına tıklandığında sadece o bölümde okutulan dersler listelenmektedir. Ders adına tıklandığında aynı liste sadece bu derslerin olduğu liste gelir. Öğretin görevlisine tıklandığında sadece o öğretim görevlisinin verdiği dersler listelenir.

“Bölüm Adı” başlığına tıklandığında ise mevcut liste bölüm adına göre, “Ders Kodu – Adı” başlığına tıklandığında ders konuna göre, “Öğretim Elemanı” başlığına tıklandığında mevcut liste öğretim elemanın adına göre listelenir.

Sinav Yazılımı - Windows Internet Explorer

http://www.kilicaslan.com.tr/it/index.html?modul=2008islem=2&sisrno=30&sayfa=1&siralama=suz=&whDeger=#30

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Sinav Yazılımı

Bölümlerin Ders Listesi

Tümünü Listele
 Yeni Kayıt Ekle
 Tümünü Değiştirme

| Sıra No | İşlem | Bölüm Adı | | Derslik Kısıtı | Tarih Kısıtı | Grup No | |
|---------|----------------|----------------------------|----------------------------------|--------------------------------|--------------------------------|----------------|---------------|
| | | Ders Kodu - Adı | Öğretim Elemanı | | | Oğrenci Sayısı | Oturum Sayısı |
| 1 | Değiştir X Sil | Bankacılık ve Sigortacılık | BMYBS214 - Sigorta İşlemleri 2 | | | 36 | 1 |
| | | | Öğr.Gör. Berna AKSOY | | | | |
| 2 | Değiştir X Sil | Bankacılık ve Sigortacılık | BMYBS252 - Mesleki Yabancı Dil 2 | | | 40 | MYDİL-1 |
| | | | Öğr.Gör. Açelya DOĞRULU | | | | 1 |
| 3 | Değiştir | Bankacılık ve Sigortacılık | BMYBS262 - Banka İşlemleri 2 | <input type="checkbox"/> Kısıt | <input type="checkbox"/> Kısıt | 41 | |
| | | | Öğr.Gör. Alptekin GÜNEY | | | 1 | |
| 4 | Değiştir X Sil | Bankacılık ve Sigortacılık | BMYBS264 - Kredi İşlemleri | | | 35 | 1 |
| | | | Öğr.Gör. Ümit HACIOĞLU | | | | |
| 5 | Değiştir X Sil | Bankacılık ve Sigortacılık | BMYBS266 - Kambiyo Bilgisi | | | 31 | BMYBS266-AG |
| | | | Öğr.Gör. Alptekin GÜNEY | | | | 1 |
| 6 | Değiştir X Sil | Bankacılık ve Sigortacılık | BMYBS268 - Sermaye Piyasaları | | | 35 | 1 |
| | | | Öğr.Gör. Bülent HOCA | | | | |
| 7 | Değiştir X Sil | Bankacılık ve Sigortacılık | BMYIY122 - İktisat 2 | | | 39 | BMYIY122-BK |
| | | | Öğr.Gör. Bilal KARGI | | | | 1 |
| | Değiştir X Sil | Bankacılık ve Sigortacılık | | | | | BMYIY126-FG |

Şekil 5. Kayıt değiştirme ekran görünümü

Değiştir linki tıklandığında Şekil-5'deki resim oluşur. Burada bazı alanlar açılan kutu, bazıları metin kutusudur. Bölüm adı açılan kutusu veri kaynağı bölüm tablosudur. Benzer şekilde, ders kodu, öğretim elemanı, grup no gibi alanlarda da veri kaynağı kendi tablolarıdır.

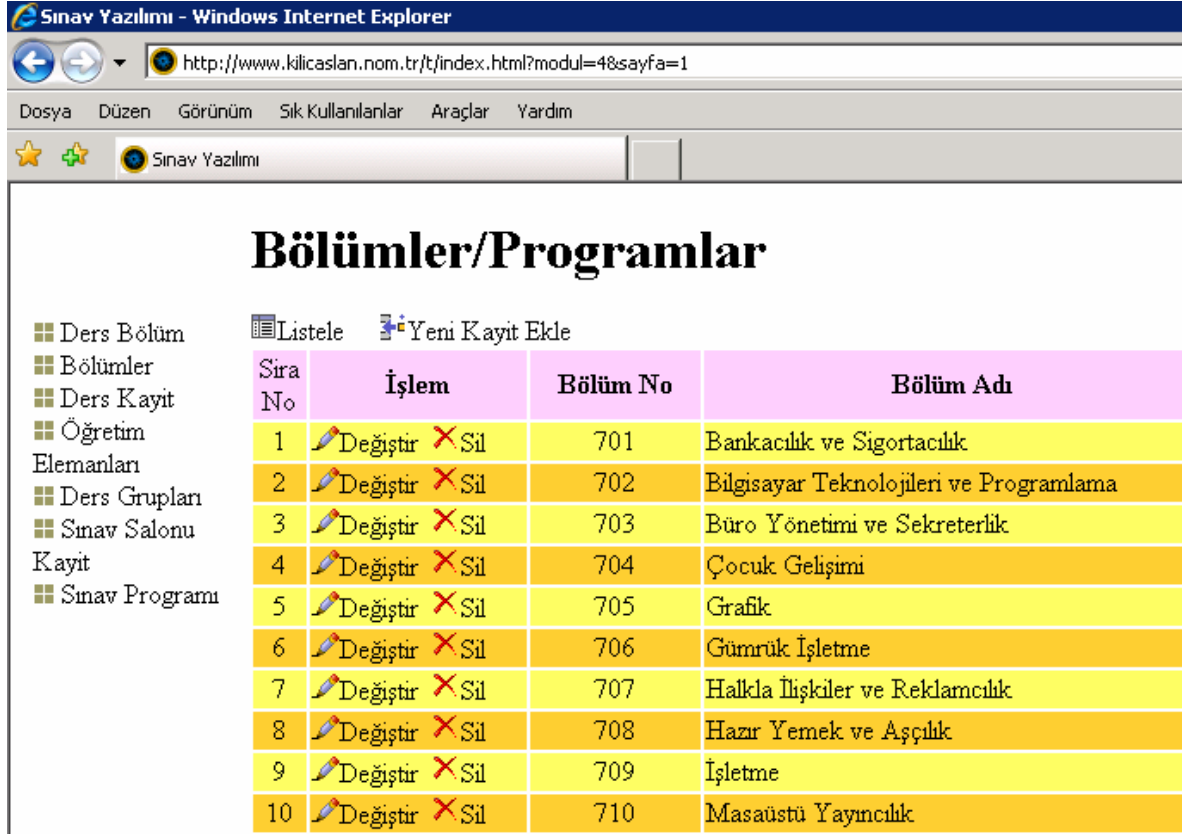
Tarih checkbox'ı ve derslik checkbox'ı seçili olduğunda ekrana bir seçim kutusu gelmektedir. Şekil 6'da bu ders için PC-LAB1 ve PC-LAB2 seçili, 4.6.2007 de 1 ve 2. oturumlar seçilidir.

| | | | | |
|---|-------------------------|------------------------------|---|---|
| 4 | Öğr.Gör. Açelya DOĞRULU | | | 1 |
| 3 | Değiştir | Bankacılık ve Sigortacılık | <input checked="" type="checkbox"/> Kısıt | <input checked="" type="checkbox"/> Kısıt |
| | | BMYBS262 - Banka İşlemleri 2 | MAC-LAB2 | 1.6.2007/4 |
| | | Öğr.Gör. Alptekin GÜNEY | MUTFAK | 4.6.2007/1 |
| | | | PC-LAB1 | 4.6.2007/2 |
| | | | PC-LAB2 | 4.6.2007/3 |

Şekil 6. Kayıt değiştir bölümün görünüşü

3.1.2.2. Bölümler menüsü

Bölümler liste görünümü



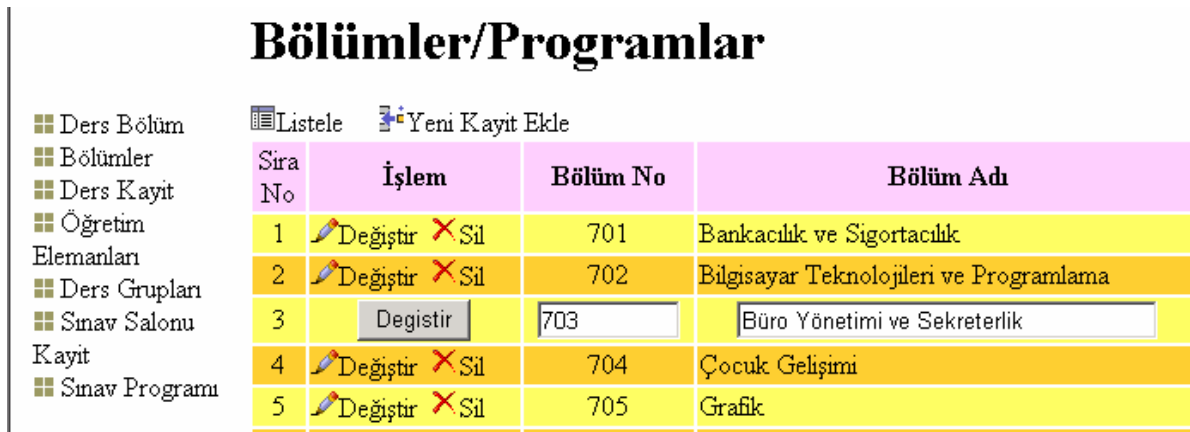
Bölümler/Programlar

■ Ders Bölüm
■ Bölümler
■ Ders Kayıt
■ Öğretim
Elemanları
■ Ders Grupları
■ Sınav Salonu
Kayıt
■ Sınav Programı

Listele Yeni Kayıt Ekle

| Sıra No | İşlem | Bölüm No | Bölüm Adı |
|---------|--------------|----------|---|
| 1 | Değiştir Sil | 701 | Bankacılık ve Sigortacılık |
| 2 | Değiştir Sil | 702 | Bilgisayar Teknolojileri ve Programlama |
| 3 | Değiştir Sil | 703 | Büro Yönetimi ve Sekreterlik |
| 4 | Değiştir Sil | 704 | Çocuk Gelişimi |
| 5 | Değiştir Sil | 705 | Grafik |
| 6 | Değiştir Sil | 706 | Gümrük İşletme |
| 7 | Değiştir Sil | 707 | Halkla İlişkiler ve Reklamcılık |
| 8 | Değiştir Sil | 708 | Hazır Yemek ve Aşçılık |
| 9 | Değiştir Sil | 709 | İşletme |
| 10 | Değiştir Sil | 710 | Masaüstü Yayıncılık |

Şekil 7. Bölümlerin liste görünümü



Bölümler/Programlar

■ Ders Bölüm
■ Bölümler
■ Ders Kayıt
■ Öğretim
Elemanları
■ Ders Grupları
■ Sınav Salonu
Kayıt
■ Sınav Programı

Listele Yeni Kayıt Ekle

| Sıra No | İşlem | Bölüm No | Bölüm Adı |
|---------|--------------|----------|---|
| 1 | Değiştir Sil | 701 | Bankacılık ve Sigortacılık |
| 2 | Değiştir Sil | 702 | Bilgisayar Teknolojileri ve Programlama |
| 3 | Değiştir | 703 | Büro Yönetimi ve Sekreterlik |
| 4 | Değiştir Sil | 704 | Çocuk Gelişimi |
| 5 | Değiştir Sil | 705 | Grafik |

Şekil 8. Bölümler değiştir ekran görünümü

Ders Listesi

Listele Yeni Kayıt Ekle

| Sıra No | İşlem | Ders Kodu | Ders Adı |
|---------|----------------|-----------|-----------------------|
| 1 | Değiştir X Sil | BMYS214 | Sigorta İşlemleri 2 |
| 2 | Değiştir X Sil | BMYS252 | Mesleki Yabancı Dil 2 |
| 3 | Değiştir X Sil | BMYS260 | Kambiyo Bilgisi |
| 4 | Değiştir X Sil | BMYS262 | Banka İşlemleri 2 |
| 5 | Değiştir X Sil | BMYS264 | Kredi İşlemleri |
| 6 | Değiştir X Sil | BMYS266 | Kambiyo Bilgisi |
| 7 | Değiştir X Sil | BMYS268 | Sermaye Piyasaları |

Şekil 9. Ders listesi görünümü

Öğretim Elemanları

























Listele Yeni Kayıt Ekle

| Sıra No | İşlem | Sicil No | Adı | Soyadı | Ünvanı |
|---------|----------------|----------|-------------|--------------|-------------|
| 51 | Değiştir X Sil | 74 | Orhan | KOÇAK | Yrd.Doç.Dr. |
| 52 | Değiştir X Sil | 47 | Zeynep | KOPAN | Öğr. Gör. |
| 53 | Değiştir X Sil | 27 | Muharrem | KOZ | Öğr. Gör. |
| 54 | Değiştir X Sil | 19 | Cevdet | KUÇUK | Prof.Dr. |
| 55 | Değiştir X Sil | 39 | Nazmiye Ece | MACİT | Öğr. Gör. |
| 56 | Değiştir X Sil | 26 | Kamil | MALKOÇOĞLU | Yrd.Doç.Dr. |
| 57 | Değiştir X Sil | 58 | Ayşe Çiğdem | MERCANLIOĞLU | Yrd.Doç.Dr. |
| 58 | Değiştir X Sil | 31 | Hülya | NALÇAKAN | Öğr. Gör. |
| 59 | Değiştir X Sil | 40 | İbrahim | OK | Yrd.Doç.Dr. |
| 60 | Değiştir X Sil | 82 | Nesrin | ONUR | Öğr. Gör. |

Şekil 10. Öğretim elemanları liste görünümü



Ders Grupları

  Yeni Kayıt Ekle



| Sıra No | İşlem | Grup No | Grup Adı |
|---------|---|-------------|-----------------------|
| 1 |   | BMYBS266-AG | ALPTEKİN |
| 2 |   | BMYHA104-G | HY-Mutfak Teknikleri |
| 3 |   | BMYHR162-AS | aybike |
| 4 |   | BMYHR262-AS | ALİ SAYGILI |
| 5 |   | BMYIY122-AU | ADEM UZUN |
| 6 |   | BMYIY122-BH | İKTİSAT2 |
| 7 |   | BMYIY122-BK | Bilal Kargı iktisat 2 |
| 8 |   | BMYIY126-EB | Ticaret Hukuku |
| 9 |   | BMYIY126-FG | Ticaret Hukuku |
| 10 |   | BMYIY208-MB | MEHMET BİRİNCİ |
| 11 |   | BMYIY232-ÇM | ÇİĞDEM MERCANLIOĞLU |
| 12 |   | BMYIY235-MK | MÜJDE KOCAMAN |





















Şekil 11. Ders grupları liste görünümü

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

  Sinav Yazılımı

Sınav Salonları

  Yeni Kayıt Ekle

| Sıra No | İşlem | Salon No | Sınav Kapasitesi | Atama Önceliği | Gözetmen Sayısı |
|---------|---|----------|------------------|----------------|-----------------|
| 1 |   | 111 | 20 | 1 | 1 |
| 2 |   | 112 | 20 | 1 | 1 |
| 3 |   | 113 | 20 | 1 | 1 |
| 4 |   | 114 | 20 | 1 | 1 |
| 5 |   | 115 | 20 | 1 | 1 |
| 6 |   | 116 | 20 | 1 | 1 |
| 7 |   | 117 | 20 | 1 | 1 |
| 8 |   | 118 | 20 | 1 | 1 |
| 9 |   | 61 | 20 | 1 | 1 |
| 10 |   | 62 | 20 | 1 | 1 |

- Ders Bölüm
- Bölümler
- Ders Kayıt
- Öğretim Elemanları
- Ders Grupları
- Sınav Salonu Kayıt
- Sınav Programı

Şekil 12. Sınav salonları liste görünümü

| Internet Explorer | | | | | | | |
|--|--|--|--|--|--|---|--|
| w.kilicaslan.nom.tr/t/index.html?modul=100 | | | | | | | |
| Sık Kullanılanlar Araçlar Yardım | | | | | | | |
| Uluslararası Lojistik | | Atatürk İlkeleri
İnkılap Tarihi 2
64-65-66 | | | | BMYPD102 - Park Dm
2
63-64 | |
| Uluslararası Ticaret | | BMYPD104 -
Atatürk İlkeleri
İnkılap Tarihi 2
67-72-73 | | | | BMYPD260 -
Uluslararası Ticarete
Kullanılan Belgeler
65-66 | |

2006-2007 BAHAR DÖNEMİ FINAL SINAVI PROGRAMI

| 29/ 5/2007
SALI | 1.Sınıflar | | | | 2.Sınıflar | | | |
|---|---|---|--|--|--|-------------|--|--|
| | 09:30-10:30 | 11:30-12:30 | 14:00-15:00 | 16:00-17:00 | 09:30-10:30 | 11:30-12:30 | 14:00-15:00 | 16:00-17:00 |
| Bankacılık ve Sigortacılık | | BMYPD172 -
Bilgisayar 2
PC-LAB1, PC-LAB3 | | | | | | BMYPD240 - Mal
Tablolar Analizi
111-112-113 |
| Bilgisayar Teknolojileri ve Programlama | | BMYPD118 - Veri
Tabanı Yönetim
Sistemleri 2
B1-71 | | | | | | |
| Büro Yönetimi ve Sekreterlik | | BMYPD104 -
Dosyalama Ve
Arşivleme Teknikleri
111-112-113 | | | | | | |
| Çocuk Gelişimi | | | BMYPD172 -
Bilgisayar 2
PC-LAB1, PC-LAB2,
PC-LAB3 | | BMYPD266 - Özel
Eğitim 2
111-112 | | | |
| Grafik | | BMYPD164 - Desen
2
B1-29 | | | | | | BMYPD262 -
Grafik Tasarım
Proje 2
B1-36 |
| Gümrük İşletme | BMYPD134 -
İstatistik
111-112-113-114 | | | | | | | BMYPD206 -
Serbest Bölgeler Ve
Gümrük
114-115-116 |
| Halkla İlişkiler ve Reklamcılık | | | | | | | BMYPD232 -
Girişimcilik Ve Küçük
İşletme Yönetimi
111-112-113-114 | |
| Hazır Yemek ve Aşçılık | | | BMYPD232 -
Girişimcilik Ve Küçük
İşletme Yönetimi
111-112-113-114 | | BMYPD210 -
Yiyecek-İçecek
Mahiyet Kontrol
113 | | | |
| İşletme | BMYPD134 -
İstatistik
115-116-117-118 | | | | | | BMYPD232 -
Girişimcilik Ve Küçük
İşletme Yönetimi
115-116-117 | |
| Masaüstü Yayıncılık | | | | BMYPD172 -
Bilgisayar 2
PC-LAB1, PC-
LAB2 | BMYPD202 -
Tipografi 2
114-115-116 | | | |

Şekil 13. Sonuç ders programının bir kısmının ekran görünümü

4. SONUÇ VE DEĞERLENDİRME

Yaptığımız çalışmada PHP script dili kullanılmıştır. Çalışmamızda aşağıdaki konularda sonuca ulaşılmıştır.

- Bir sınav programında olması kesinlikle gerekli olan çakışmama koşulu sağlanmaktadır.
- Belirtilen tarih koşuluna uygun olarak birinci öncelikle yerleştirme yapmaktadır.
- Bir sonraki gün olacak ders koşulunu sağlamaktadır.
- Bir sonraki oturumda olmayacak dersler koşulunu sağlamaktadır.
- Her oturumda yaklaşık olarak eşit sayıda sınav olması gerekir. Bu koşulu sağlamaktadır.
- Aynı anda olması gereken sınav koşulunu yerine getirmektedir.

Ancak aşağıdaki kısımların geliştirilme ihtiyacı vardır.

- Bir ders için aynı anda hem derslik ve hemde tarih kısıtı çalışmamaktadır.
- Bir ders için birden fazla tarih kısıtı belirtildiğinde bu derse hangi oturumun atanacağını belirlemede en çok tercih edilen oturuma yapılmaktadır. Tarih kısıtı olarak sınav yerleştirilmeyecek gün de belirtilmelidir.
- Bir yerleşimde birden çok koşul istenmektedir. Bu kısıtlar nedeniyle yerleştirme mümkün olmayabilir. Bu durumda bazı dersler için bu koşulların bir kısmından vaz geçilebilir.
- Bu yazılım vaz geçme durumunda aşağıdaki sıraya uygun olarak vaz geçer.
 - Her oturumda yaklaşık olarak eşit sayıda sınav olması gerekir. Yerleştirme yapılamaması durumunda bundan vazgeçilebilir.
 - Yinede yerleştirme yapılamaz ise bir sonraki oturumda olmayacak ders kısıtı dikkate alınmaz.
- Her şeye rağmen yerleştirme yapılamaz ise bazı tarih kısıtlarını iptal etmek gerekir.

Kaynakça

Çölkesen, R. “Veri Yapıları ve Algoritmalar, Papatya Yayıncılık, İstanbul, 2004.

West, D.B., Introduction To Graph Theory, Second Ed. Prentice Hall Inc. 2001.

Bradley, P, Brassard G., Fundamentals of Algoritm, Prentice Hall Inc., 1996.

<http://oneweb.utc.edu/~Christopher-Mawata/petersen/index.htm> (05.03.2007)

<http://www.maths.mq.edu.au/~wchen/lndmfolder/lndm.html> (05.03.2007)

<http://theory.csail.mit.edu/classes/6.042/fall05/ln5.pdf> (05.03.2007)

ÖZGEÇMİŞ

16 Mayıs 1967 tarihi, Giresun İli Tirebolu ilçesinde doğdu. İlk, Orta ve Liseyi yine aynı ilçede tamamladıktan sonra, Yıldız Üniversitesi, Mühendislik Fakültesi, Makine Mühendisliği Bölümüne kaydoldu. Bu bölümden 1989 yılında mezun olduktan sonra, 1992 yılında aynı Üniversitenin Fen Bilimleri Enstitüsü Makine Mühendisliği Enerji Anabilim Dalında Yüksek Lisans eğitimi bitirdi. 1991 -1995 yıllarında Adem Çelik İnşaat Firmasında Tesisat Mühendisi olarak çalıştı. 1995-1997 yılları arasında Uçar Plastik, 1997 - 2004 yılları arasında Dizayn Teknik A.Ş. Alt Yapı konusunda çalıştıktan sonra halen Beykent Üniversitesi Meslek Yüksek Okulun'da Öğretim Görevlisi olarak çalışmaktadır. 2006 yılında Beykent Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladı.

Yabancı dili İngilizce olup, evli ve bir çocuk babasıdır.