

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**HAFTALIK DERS PROGRAMI OLUŞTURMA**

**Hi̇lal AYDIN - 220501030**

**Şevval ÖZEREN - 220501028**

**Esmanur AYDIN - 220501005**

**Dr. Öğr. Üyesi Elif Pınar HACIBEYOĞLU**

**22.03.2025**

1. **GİRİŞ**
   * **Projenin amacı**

* Projenin amacı, mühendislik fakültesine ait haftalık ders programını otomatik olarak oluşturan bir sistem geliştirmektir.
* Projede gerçekleştirilmesi beklenenler şunlardır:
  + Öğrenci, öğretim üyesi ve yönetici ekleme ve çıkarma
  + Yeni bölüm ekleme ve mevcut bölümleri düzenleme
  + Bölümlere ait ders planlarını sisteme ekleme
  + Ders ekleme, düzenleme ve silme işlemleri
  + Dersin öğretim üyesini belirleme
  + Haftalık ders saatlerini belirleme
  + Ortak derslerin eş zamanlı atanmasını sağlama
  + Derslik ekleme ve silme
  + Kapasiteye uygun derslik atama
  + Pazartesi-Cuma günleri için saat bazlı ders ataması.
  + Online derslerin 17:00-21:00 saatlerine yerleştirilmesi
  + Aynı sınıf veya öğretim üyesine çakışan derslerin önlenmesi
  + Ortak öğrencisi olan dersler için uygun zaman aralığı belirleme
  + Aynı dersin birden fazla sınıfa aynı anda atanmasını önleme
  + Öğretim üyelerinin uygunluk durumlarına göre ders planlama
  + Derslik kapasitesine uygun şekilde öğrenci yerleşimi yapma

1. **GEREKSİNİM ANALİZİ**
   * **Arayüz gereksinimleri**

* Kullanıcı, işlemleri bir menü sistemi üzerinden seçer.
* Kullanıcının işlemler arasında rahatça geçiş yapabilmesi için geri dönme ve çıkış seçenekleri bulunur.
* Kullanıcı girişindeki yönlendirmeler net ve anlaşılır.
* Kullanıcı geçersiz bir giriş yaptığında hata mesajı verir.
* Kullanıcılar oluşturdukları ders programlarını Excel (.xlsx) formatında dışa aktarır.
  + **Fonksiyonel gereksinimler**

**Kullanıcı Yönetimi:**

* Kullanıcılar (öğrenci, öğretim üyesi, yönetici) sisteme eklenebilmeli, silinebilmeli ve güncellenebilmelidir.
* Kullanıcılar terminal üzerinden CLI (Command Line Interface) aracılığıyla sistemle etkileşime girebilmelidir.
* Kullanıcı bilgileri veritabanında saklanmalıdır.
* Öğretim üyeleri bir veya birden fazla bölüme atanabilmelidir.

**Bölüm Yönetimi:**

* Yeni bölümler sistem üzerinden eklenmeli ve var olan bölümler silinebilmelidir.
* Bölümlerin ders listesi oluşturulmalı ve her bölümde okutulacak dersler belirlenmelidir.

**Ders Yönetimi:**

* Yeni ders eklenmeli, mevcut dersler silinebilmelidir.
* Dersler belirli bir döneme ve bölüme atanmalıdır.
* Derslerin öğretim üyeleri ile ilişkisi kurulmalı ve her dersin bir öğretim üyesi olmalıdır.
* Aynı ders farklı bölümlerde ortak olarak okutulabiliyor olmalıdır.
* Dersler haftalık programda uygun saatlere atanmalıdır.

**Derslik Yönetimi:**

* Derslikler sisteme eklenebilmeli ve kapasite bilgileri belirlenmelidir.
* Derslikler normal sınıf veya laboratuvar olarak sınıflandırılmalıdır.
* Her ders uygun kapasitedeki dersliklere atanmalıdır.

**Haftalık Ders Programı Oluşturma:**

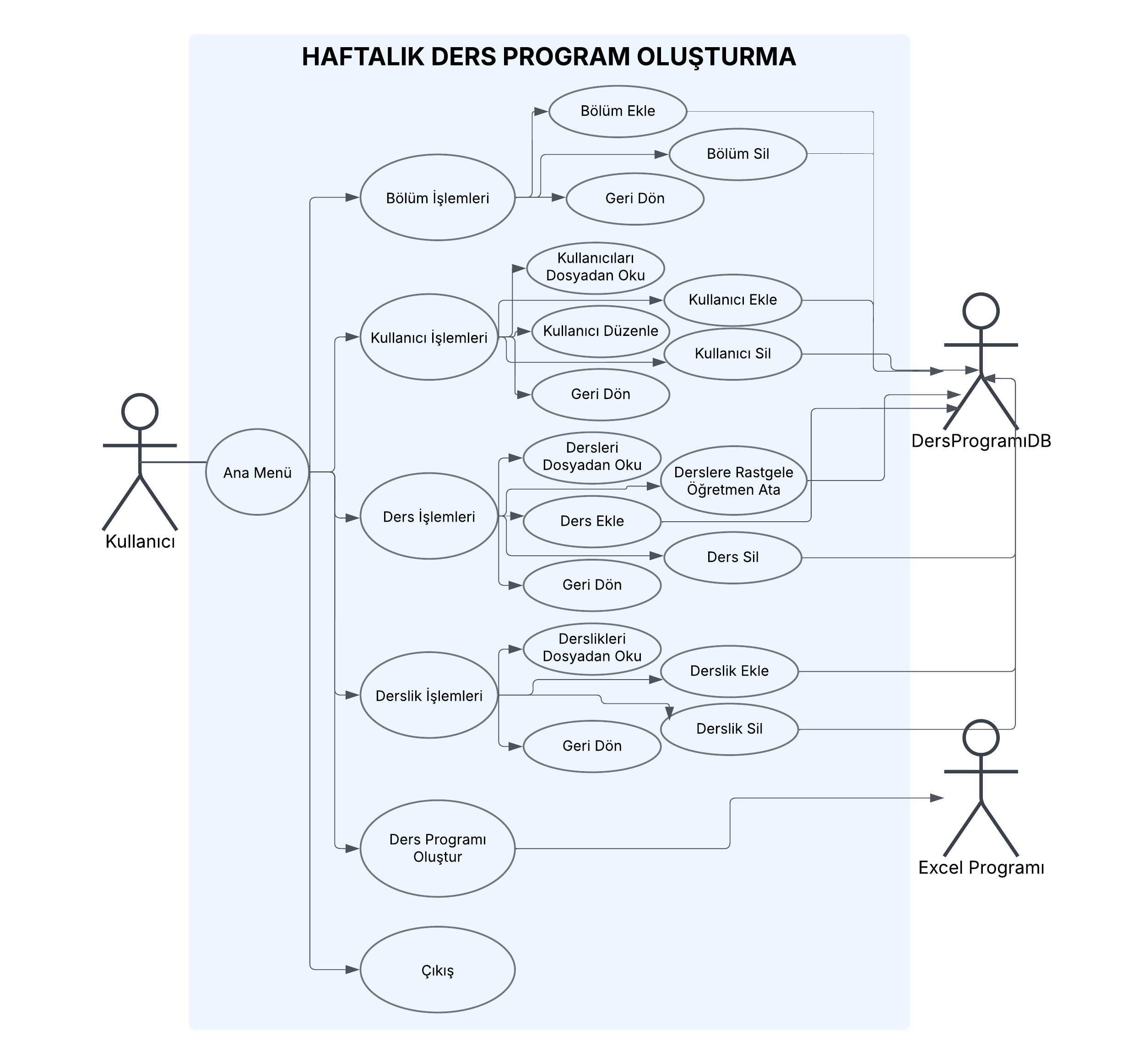
* Dersler haftanın günlerine ve saat aralıklarına uygun olarak yerleştirilmelidir.
* Öğretim üyelerinin dersleri çakışmayacak şekilde planlanmalıdır.
* Online dersler saat 17:00 – 21:00 arasında planlanmalıdır.
* Ders programı, belirlenen dersliklere uygun şekilde oluşturulmalıdır.

**Veritabanı Yönetimi:**

* SQLAlchemy kullanılarak MSSQL tabanlı bir veritabanı oluşturulmalıdır.
* Bilgiler veritabanında saklanmalı ve yönetilmelidir.
* Veritabanındaki tablolar otomatik olarak oluşturulmalı ve eksik tablolar tamamlanmalıdır.

**Excel Çıktısı Alma:**

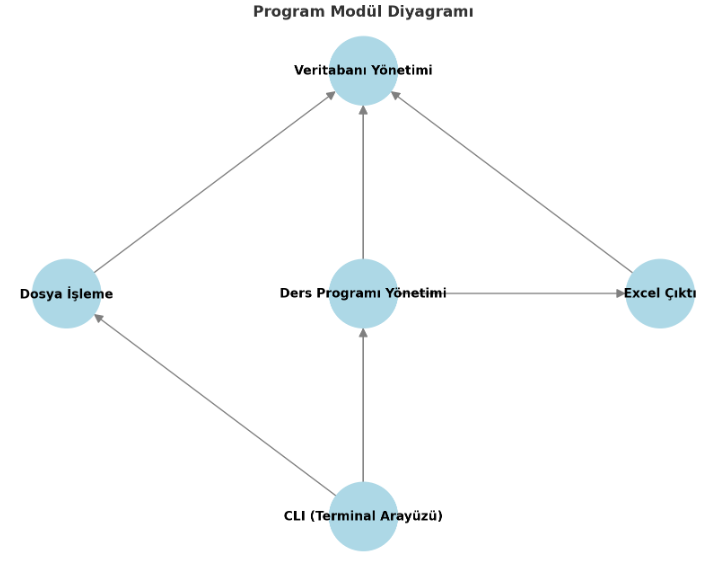
* Oluşturulan haftalık ders programı Excel (.xlsx) formatında kaydedilebilmelidir.
* Program, belirlenen formatta Excel dosyasına yazdırılmalıdır.
* Farklı bölümler için ayrı Excel sayfaları oluşturulmalıdır.
  + **Use-Case diyagramı**



1. **TASARIM**
   * **Mimari tasarım**

**Katmanlı Mimari Yapı:**

* **Veritabanı** **Katmanı**: SQLAlchemy ile MSSQL veritabanı yönetimi sağlanmıştır. Kullanıcı, ders, derslik ve bölüm bilgileri burada saklanır.
* **İş Mantığı Katmanı**: Ders programı oluşturma, öğretim üyesi atama, çakışma kontrolleri ve derslik atamaları burada gerçekleştirilir.
* **Arayüz Katmanı**: Kullanıcıların terminal üzerinden etkileşim sağlamasına olanak tanır.



* + **Kullanılacak teknolojiler**
* Bu program Python dili kullanılarak oluşturulmuştur.
* SQLAlchemy kütüphanesi kullanılmıştır. Bu kütüphane veritabanı yönetimi ve etkileşimi için kullanılır.
* Collections modülü kullanılmıştır**.** Bu modül, gelişmiş veri yapıları sağlar ve standart veri tiplerine ek özellikler kazandırır. Defaultdict ile varsayılan bir değer atayarak sözlük oluşturur.
* Openpyxl kütüphanesi kullanılmıştır. Bu kütüphane Excel (.xlsx) dosyalarını oluşturmak, okumak ve düzenlemek için kullanılır.
* Random kütüphanesi kullanılmıştır. Bu kütüphane rastgele sayı üretmek ve rastgele işlemler yapmak için kullanılır.
  + **Veri tabanı tasarımı**
* Ders Programı verilerini kaydetmek amacıyla MSSQL üzerinden “DersProgramiDB” veri tabanı ve veriler için ilgili tablolar oluşturuldu.
  1. **Kullanıcılar Tablosu (kullanicilar):** Öğretim üyeleri, öğrenciler ve yöneticileri tutar.

**Anahtarlar**:

* + - **id:** Birincil anahtar (PK).
    - **mevki**: Kullanıcının rolü (OGRETIM\_UYESI, OGRENCI, YONETICI).
    - **ad**: Kullanıcının adı.

**İlişkiler**:

* + - Öğretim üyeleri derslere atanabilir (Ders tablosu ile ogretim\_uyesi\_id üzerinden ilişki).
    - Öğretim üyeleri bölümlere bağlı olabilir (Bolum tablosu ile Many-to-Many ilişkisi).
  1. **Bölüm Tablosu (bolum):** Üniversitedeki bölümleri yönetir.

**Anahtarlar**:

* + - **id:** Birincil anahtar.
    - **kod:** Bölümün kodu (unique).
    - **ad:** Bölüm adı.

**İlişkiler**:

* + - Bölümün dersleri olabilir (Ders tablosu ile One-to-Many ilişki).
    - Bölüm öğretim üyelerine sahip olabilir (Kullanicilar tablosu ile Many-to-Many ilişki).
  1. **Ders Tablosu (ders):** Bölümlere ait dersleri tanımlar.

**Anahtarlar**:

* + - **id:** Birincil anahtar.
    - **bolum\_kod:** Bölüm ile ilişkilidir (Bolum.kod).
    - **donem:** Hangi dönemde verildiğini belirtir.
    - **kod:** Dersin kodu.
    - **ad:** Dersin adı.
    - **ders\_tipi**: Zorunlu veya seçmeli gibi türleri belirler.
    - **teorik\_saat, uyg\_saat**: Ders saat bilgileri.
    - **ogretim\_uyesi\_id:** Dersi veren öğretim üyesi (Kullanicilar.id ile ilişkilidir).

**İlişkiler**:

* + - Dersler bölümlere bağlıdır (Bolum ile Many-to-One).
    - Derslerin öğretim üyeleri olabilir (Kullanicilar ile Many-to-One).
  1. **Derslik Tablosu (derslik):** Üniversitedeki dersliklerin kapasite ve durumlarını yönetir.

**Anahtarlar**:

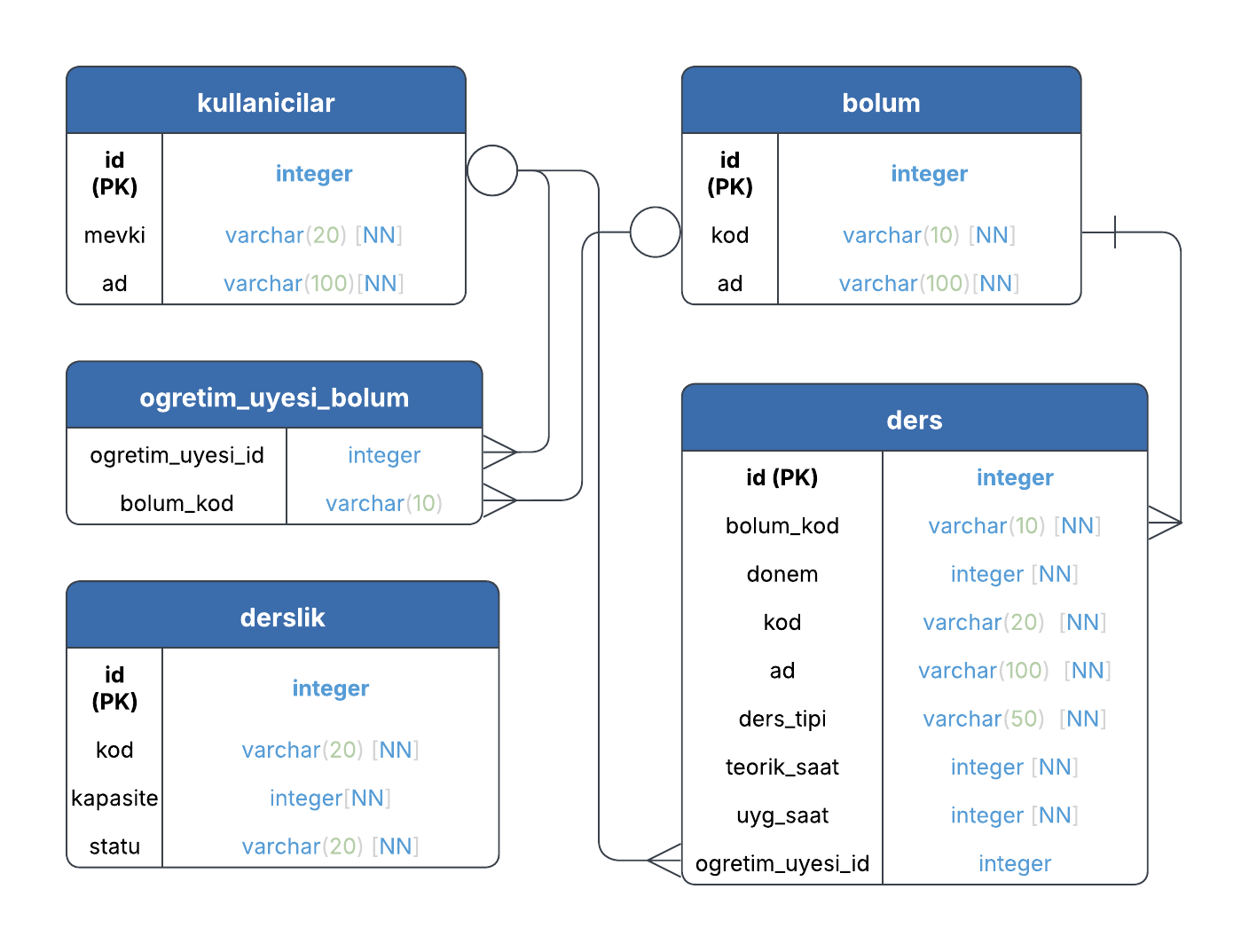
* + - **id**: Birincil anahtar.
    - **kod**: Derslik kodu (örneğin D101, LAB201 gibi).
    - **kapasite:** Derslik kapasitesi.
    - **statu:** Normal veya laboratuvar olup olmadığını belirler.
  1. **Öğretim Üyesi - Bölüm İlişki Tablosu (ogretim\_uyesi\_bolum):** Öğretim üyelerinin bir veya birden fazla bölüme bağlı olmasını sağlar.

**Anahtar Alanlar**:

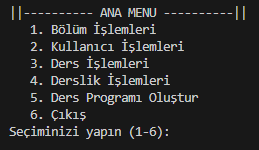
* + - **ogretim\_uyesi\_id:** Kullanicilar.id ile ilişkili.
    - **bolum\_kod**: Bolum.kod ile ilişkili.

**İlişkiler**:

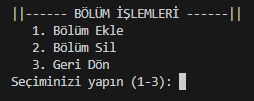
* + - Many-to-Many ilişkiyi temsil eder.
* **DersProgramiDB ER Diyagramı**



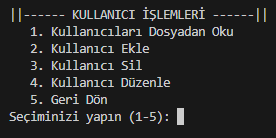
* + **Kullanıcı arayüzü tasarımı**
* Komut tasarımı arayüzünde bir menü oluşturuldu. Bu menüde Bölüm işlemleri, kullanıcı işlemleri, ders işlemleri, derslik işlemleri, ders programı oluşturma ve çıkış bulunur.



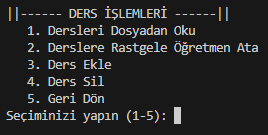
* Bölüm işlemleri seçildiğinde bölüm ekleme, bölüm silme ve geri dönme seçenekleri bulunur.
  + Bölüm eklemek için bölümün adı ve kodu (BM, YM) girilir.
  + Bölüm silmek için silinecek bölümün kodu girilir.



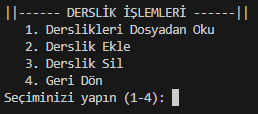
* Kullanıcı işlemleri seçildiğinde kullanıcıları txt dosyasından okuma, kullanıcı ekleme, kullanıcı silme, kullanıcı düzenleme ve geri dönme seçenekleri bulunur.
  + Kullanıcıları dosyadan okumak için dosyanın adı girilir.
  + Kullanıcı eklemek için kullanıcının adı ve mevkisi (öğretim\_üyesi, öğrenci, yönetici) girilir.
  + Kullanıcı silmek için kullanıcının adı girilir.
  + Kullanıcı düzenlemek için kullanıcının adı ve düzenlenecek bilgisi (mevki, bölüm) girilir.



* Ders işlemleri seçildiğinde dersleri txt dosyasından okuma, derslere rastgele öğretmen atama, ders ekleme, ders silme ve geri dönme seçenekleri bulunur.
  + Dersleri dosyadan okumak için dosyanın adı girilir.
  + Derslere rastgele öğretmen atanır.
  + Ders eklemek için bölüm kodu, dersin dönemi, dersin kodu, dersin adı, ders tipi, dersin teorik saati, dersin uygulama saati ve dersin öğretim görevlisinin id’si girilir.
  + Ders silmek için dersin kodu girilir.



* Derslik işlemleri seçildiğinde derslikleri txt dosyasından okuma, derslik ekleme, derslik silme ve geri dönme seçenekleri bulunur.
  + Derslikleri dosyadan okumak için dosyanın adı girilir.
  + Derslik eklemek için dersliğin kodu, dersliğin kapasitesi ve dersliğin statüsü girilir.
  + Derslik silmek için silinecek dersliğin kodu girilir.



* Ders programı oluşturma seçildiğinde hangi dönem olduğu seçilir.

1. **UYGULAMA**
   * **Kodlanan bileşenlerin açıklamaları**

* **database\_exists**(): Veritabanının mevcut olup olmadığını kontrol eder.
* **create\_database()**: Veritabanı yoksa otomatik olarak oluşturur.
* **tables\_exist()**: Gerekli tabloların var olup olmadığını denetler.
* **Kullanicilar**: Kullanıcıları (öğrenci, öğretim üyesi, yönetici) saklar.
* **Bolum**: Bölümleri ve bölüme ait dersleri tutar.
* **Ders**: Ders bilgilerini ve öğretim üyesi atamalarını saklar.
* **Derslik**: Derslik kodu, kapasite ve statüsünü tutar.
* **ogretim\_uyesi\_bolum:** Öğretim üyelerinin bölümlerle olan ilişkisini yöneten Many-to-Many ilişki tablosudur.
* **load\_kullanicilar\_from\_file()**: kullanicilar.txt dosyasından kullanıcıları veritabanına ekler.
* **load\_ogretim\_uyesi\_bolum\_from\_file()**: Öğretim üyelerinin bölümlerle ilişkisini dosyadan okur ve veritabanına kaydeder.
* **load\_dersler\_from\_file()**: dersler.txt dosyasından dersleri okur ve veritabanına ekler.
* **load\_derslikler\_from\_file()**: derslik.txt dosyasından derslik bilgilerini okur ve kaydeder.
* **assign\_random\_courses()**: Öğretim üyelerine rastgele ders ataması yapar.
* **dersleri\_oku()**: Belirli bir döneme ait dersleri veritabanından çeker.
* **ders\_programi\_olustur()**: Bilgisayar mühendisliği ve yazılım mühendisliği bölümleri için haftalık ders programını oluşturur.
* **dersleri\_ayir()**: Ortak dersler ile bölümlere özel dersleri birbirinden ayırır.
* **ortak\_dersleri\_yerlestir()**: Ortak dersleri her iki bölüm için aynı saatlere yerleştirir.
* **derslikleri\_atama()**: Derslere uygun derslikler atar.
* **derslikleri\_oku()**: Mevcut derslikleri veritabanından çeker.
* **menu()**: Kullanıcının işlem seçimi yapacağı ana menüyü oluşturur.
* **kullanici\_islemleri()**: Kullanıcı ile ilgili işlemlerin seçileceği menüyü oluşturur.
* **kullanici\_ekle()**, **kullanici\_sil()**, **kullanici\_duzenle()**: Kullanıcı ekleme, silme ve güncelleme işlemlerini gerçekleştirir.
* **bolum\_islemleri()**: Bölüm ile ilgili işlemlerin seçileceği menüyü oluşturur.
* **bolum\_ekle()**, **bolum\_sil()**: Yeni bölüm ekleme ve var olan bölümü silme işlemlerini yapar.
* **ders\_islemleri()**: Ders ile ilgili işlemlerin seçileceği menüyü oluşturur.
* **ders\_ekle()**, **ders\_sil()**: Ders ekleme ve silme işlemlerini gerçekleştirir.
* **derslik\_islemleri()**: Derslik ile ilgili işlemlerin seçileceği menüyü oluşturur.
* **derslik\_ekle()**, **derslik\_sil()**: Derslik ekleme ve silme işlemlerini gerçekleştirir.
* **excele\_yazdir()**: Ders programını Excel formatında kaydeder.
* **yeni\_sayfa\_olustur()**: Excel dosyasında yeni bir sayfa oluşturur.
* **program\_excele\_yaz()**: Dersleri belirtilen formata uygun şekilde Excel'e yazar.
  + **Görev dağılımı**
* Bileşenlerinin tasarım ve geliştirilmesinde herkes belirli kısımlarda rol almıştır.
* Rapor ortak hazırlanmıştır.
  + **Karşılaşılan zorluklar ve çözüm yöntemleri**
* Geliştirme sürecinde ders programı oluşturma kısımlarında derslerin çakışma kontrolü kısımlarında problemler yaşanmış, ilgili sınıfın fonksiyonları üzerinde yapılan değişiklikler, eklemeler vs. yöntemler ile problemin çözülmesi için çalışılmıştır.
  + **Proje isterlerine göre eksik yönler**
* Ders programı oluşturulurken, derslere derslik ataması belirli kriterlere göre yapılıyor ancak aynı saatte olan farklı dersler için derslik ataması yapılırken çakışma kontrolü yetersiz kalıyor.

1. **GİTHUB LİNKLERİ**

* [**https://github.com/sevvalozrn**](https://github.com/sevvalozrn)
* [**https://github.com/HilallAydinn**](https://github.com/HilallAydinn)
* [**https://github.com/esm4ydn**](https://github.com/esm4ydn)