



**T.C**  
**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU:HAFTALIK DERS PROGRAMI**  
**OLUŞTURMA**

**ÖĞRENCİ ADI: GÜREL BİLGİN, GİZEM YALÇIN, BERKAY**  
**ARAS, ALİ AKSOY**

**ÖĞRENCİ NUMARASI: 220502041, 240501037, 220501033,**  
**210501009**

**DERS SORUMLUSU: ELİF PINAR HACİBEYOĞLU**

**TARİH: 22.03.2025**

# 1 GİRİŞ

## 1.1 Projenin Amacı

Bu projenin amacı, okulumuz Kocaeli Sağlık ve Teknoloji Üniversitesi Mühendislik Fakültesi için sistematik haftalık ders programı oluşturma sistemi geliştirmektir. Sistem, Bilgisayar Mühendisliği (BLM) ve Yazılım Mühendisliği (YZM) bölümleri için belirli kısıtlar dahilinde ders çakışmalarını önleyen, derslik kapasitelerini dikkate alan ve kullanıcı dostu bir arayüz sunan bir yapıya sahiptir.

### Projede Gerçekleştirilmesi Beklenenler

- Kullanıcı dostu veri giriş arayüzü oluşturulması
- DBMS kullanımı ve veri tabanının doğru tasarlanması
- Kullanıcı yönetimi (öğretim üyesi, öğrenci, yönetici ekleme-çıkarma)
- Bölüm yönetimi (bölüm ekleme-çıkarma)
- Ders yönetimi (ders ekleme-çıkarma)
- Derslik yönetimi (derslik ekleme-çıkarma)
- Haftalık program oluşturma (Pazartesi-Cuma, 09:00-17:00)
- Online dersler için saat düzenlemesi (17:00-21:00)
- Aynı sınıfa ait derslerin çakışmasının önlenmesi
- Öğretim üyelerinin uygunluk durumuna göre ders ataması
- Derslik kapasitesine uygun atama
- İki bölüme ortak verilen dersler için öncelikli yerleştirme
- Zorunlu ders saatleri olan dersler için uygun planlama
- Verilen şablonda ders programının Excel olarak oluşturulması
- Dinamik ve düzenlenebilir sistem mimarisi
- BLM ve YZM bölümleri için tüm kısıtlar dahilinde doğru sonuç üretilmesi

## 2 GEREKSİNİM ANALİZİ

### 2.1 Fonksiyonel Gereksinimler

#### Kullanıcı Yönetimi

- Öğretim üyesi, öğrenci ve yönetici hesaplarının oluşturulması ve yönetilmesi
- Kullanıcı rollerine göre erişim yetkilerinin belirlenmesi
- Kullanıcı giriş-çıkış işlemlerinin yönetilmesi

#### Bölüm Yönetimi

- Bölüm ekleme, düzenleme ve silme işlemleri
- Bölüm kodlarının tanımlanması (YZM, BLM)
- Bölümlere ait 8 yarıyılık tüm derslerin listelenmesi
-

## **Ders Yönetimi**

- Ders ekleme, düzenleme ve silme işlemleri
- Ders adı, kodu, haftalık saat sayısı, ders veren öğretim üyesi bilgilerinin tanımlanması
- İki bölümün ortak alabileceği derslerin belirlenmesi
- Zorunlu ders saatleri olan derslerin işaretlenmesi

## **Derslik Yönetimi**

- Derslik ekleme, düzenleme ve silme işlemleri
- Derslik kapasitesi ve statüsünün (NORMAL, LAB) tanımlanması
- Dersliklerin kullanılabilirlik durumlarının takibi

## **Ders Programı Oluşturma**

- Hafta içi günler (Pazartesi-Cuma) için program oluşturma
- Normal ders saatleri (09:00-17:00) ve online ders saatleri (17:00-21:00) için planlama
- Derslerin, belirlenen kısıtlamalar çerçevesinde otomatik yerleştirilmesi

## **Kısıtlama Yönetimi**

- Aynı sınıfa ait derslerin çakışmasının engellenmesi
- Ortak öğrencisi olan derslerin çakışmasının engellenmesi
- Öğretim üyelerinin uygunluk durumlarına göre ders ataması yapılması
- Bir öğretim üyesinin aynı anda birden fazla ders vermesinin engellenmesi (ortak dersler hariç)
- Derslik kapasitesine uygun atamaların yapılması
- Zorunlu ders saatleri olan derslerin ilgili saatlere yerleştirilmesi

## **Veri Doğrulama ve Tutarlılık Kontrolü**

- Veri girişlerinin doğruluğunun ve tutarlılığının kontrolü
- Çakışma ve kapasite aşımı durumlarında uyarı verilmesi
- Oluşturulan programın kısıtlamaları karşılayıp karşılamadığının kontrolü

## **Çıktı Yönetimi**

- Oluşturulan ders programının verilmiş şablona uygun olarak Excel formatında dışa aktarılması
- Sınıf, bölüm veya öğretim üyesi bazında filtrelenmiş programların oluşturulması
- Program çıktılarının kaydedilmesi ve gerektiğinde tekrar kullanılabilmesi

## **Düzenleme ve Güncelleme**

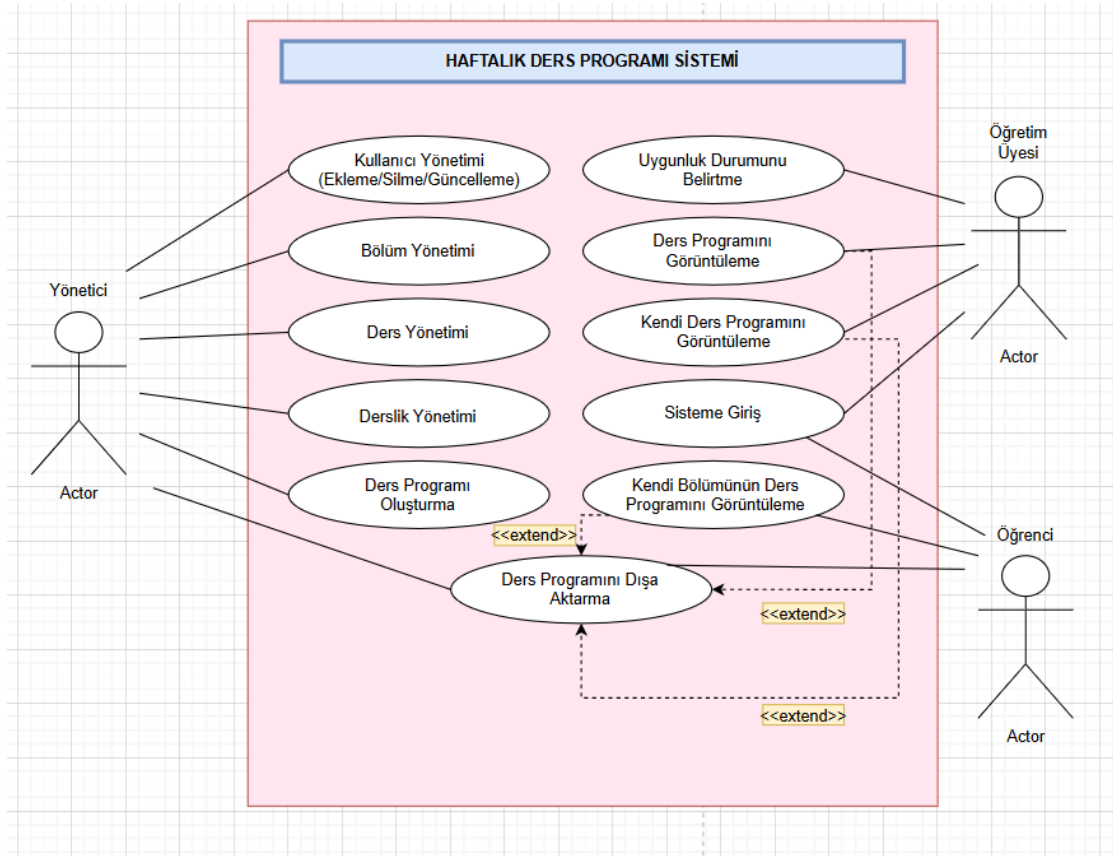
- Oluşturulan programın manuel olarak düzenlenebilmesi

- Belirli bir sınıf veya ders için program güncelleme imkanı
- Değişiklikler sonrası kısıtlama kontrollerinin tekrar yapılması

## Veri Tabanı Entegrasyonu

- Tüm verilerin seçilen veri tabanı yönetim sistemi (MySQL) ile entegre çalışması
- Veri tabanı işlemlerinin güvenli ve verimli bir şekilde gerçekleştirilmesi
- Veri yedekleme ve geri yükleme fonksiyonlarının sağlanması

## 2.2 Use-Case Diyagramı



## 3 TASARIM

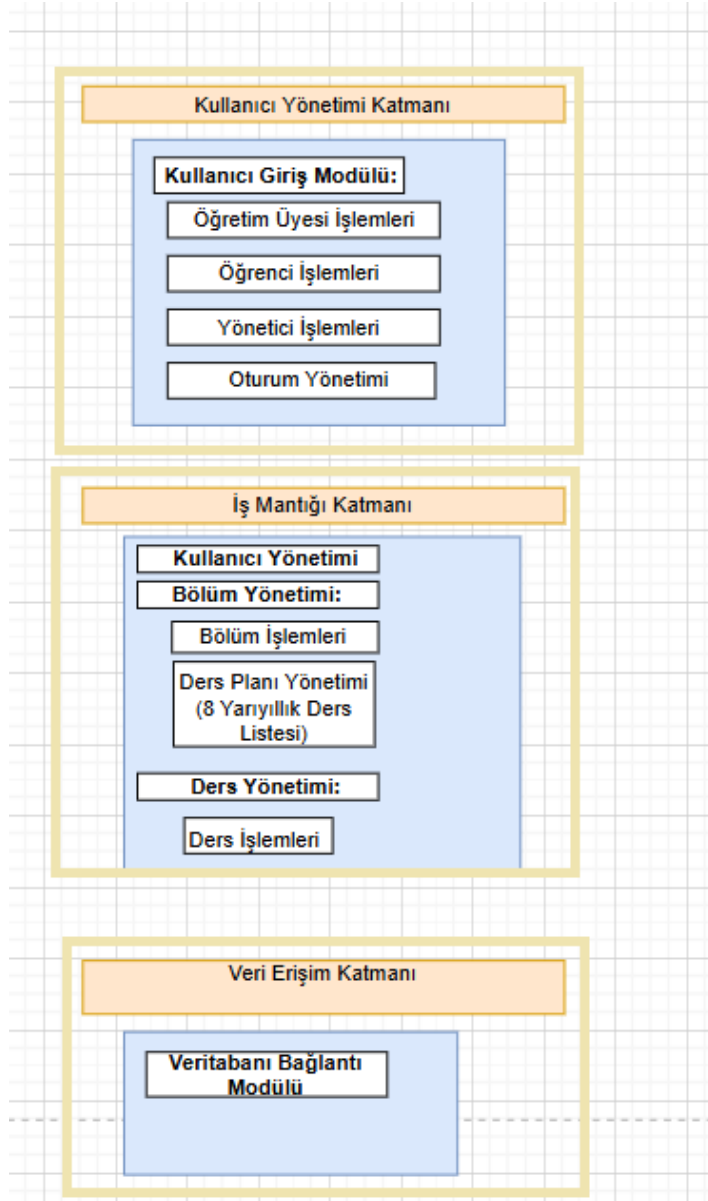
### 3.1 Mimari Tasarım

Projemizde belirli bir mimari tasarım kalıbı (örn. MVC, MVVM, Hexagonal) birebir uygulanmamıştır. Bununla birlikte, sistematik bir yapı ve düzen oluşturmak için modüler bir yaklaşım benimsenmiştir. Bu yaklaşım, uygulamanın farklı fonksiyonel bileşenlerini mantıksal olarak ayrılmış modüller halinde organize ederek, kod tabanının yönetilebilirliğini ve bakımını kolaylaştırmaktadır.

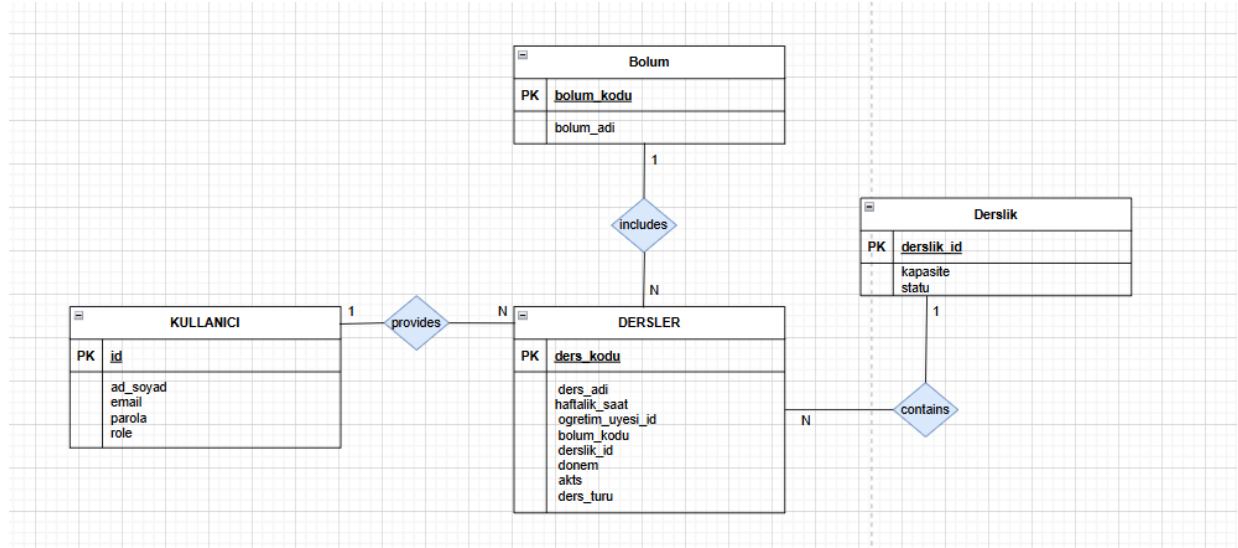
Uygulama içerisinde görev ve sorumluluklar belirli sınırlar çerçevesinde gruplandırılmış ve her bir modül kendi içinde tutarlı bir yapıya sahiptir. Kullanıcı yönetimi, bölüm yönetimi, ders yönetimi, derslik yönetimi gibi temel fonksiyonel birimler, belirli bir hiyerarşi ve düzen içerisinde çalışmaktadır.

Veri erişim katmanı ve iş mantığı katmanı arasında gevşek bir bağlantı sağlanarak, her bir bileşenin diğer bileşenlerden bağımsız olarak geliştirilmesi ve test edilmesi mümkün kılınmıştır.

## Modül Diyagramı



## ER Diyagramı:



## 3.2 Kullanılacak Teknolojiler

Ders programı oluşturma projemizin geliştirilmesinde çeşitli modern yazılım teknolojileri ve araçları bir araya getirilerek entegre bir çözüm oluşturulacaktır. Backend tarafında, veri yönetimi ve iş mantığı işlemleri için güçlü ve esnek bir yapı tercih edilmiştir.

Proje Python dilinde yazılmıştır. Veritabanı yönetimi için MySQL tercih edilmiştir.

Excel işleme “openpyxl” ile yapılmıştır.

Projenin bu kısmında henüz backend bulunduğu için fazla gereksinimi yoktur. Sonraki projede sistem tamamlanacaktır.

## 4 UYGULAMA

### 4.1 Kodlanan Bileşenlerin Açıklamaları

Kullanılan Fonksiyonlar ve Bileşenler:

#### 1. Veritabanı Yönetim Bileşenleri

- Database Sınıfı:** Veritabanı bağlantısını yöneten, sorguları çalıştıran ve sonuçları döndüren temel veritabanı arabirimi.
- execute\_query():** Veritabanında INSERT, UPDATE ve DELETE gibi işlemleri gerçekleştiren SQL sorgularını çalıştıran fonksiyon.

#### 2. Ders Programı Yönetim Bileşenleri

- **DersProgrami Sınıfı:** Ders programıyla ilgili tüm işlemleri yöneten ana sınıf. Veritabanı bağlantısını kullanarak derslerin eklenmesi, düzenlenmesi ve silinmesi gibi temel CRUD işlemlerini gerçekleştirir.
- **ders\_ekle():** Yeni bir ders kaydını veritabanına ekleyen fonksiyon. Ders kodu, ders adı, haftalık saat, öğretim üyesi, bölüm kodu, derslik, dönem, AKTS ve ders türü gibi parametreleri alır.
- **ders\_duzenle():** Var olan bir dersin bilgilerini güncelleyen fonksiyon. Mevcut ders kodunu kullanarak ilgili dersi bulur ve yeni bilgilerle günceller.
- **ders\_sil():** Belirtilen ders koduna sahip dersi veritabanından silen fonksiyon.

### 3. Kullanıcı Arayüzü Bileşenleri

- **main():** Kullanıcı etkileşimini yöneten ana fonksiyon. Menü seçeneklerini gösterir ve kullanıcının seçimine göre ilgili işlemleri başlatır.
- **Menü Sistemi:** Kullanıcıya ders ekleme, düzenleme, silme ve çıkış seçenekleri sunan basit komut satırı arayüzü.
- **Kullanıcı Giriş İşlemleri:** input() fonksiyonu kullanılarak kullanıcıdan gerekli bilgileri alan ve doğrulayan işlemler.

### 4. Hata Yönetimi Bileşenleri

- **Exception Handling:** try-except blokları kullanılarak veritabanı işlemleri sırasında oluşabilecek hataları yakalayan ve kullanıcıya bilgi veren mekanizma.
- **Kullanıcı Doğrulama Mekanizmaları:** Kullanıcı girişlerinin geçerliliğini kontrol eden ve geçersiz girişlerde kullanıcıyı uyaran işlemler.

### 5. Program Akış Kontrolü

- **while Döngüsü:** Kullanıcı çıkış seçeneğini seçene kadar programın çalışmaya devam etmesini sağlayan döngü.
- **if-elif-else Yapısı:** Kullanıcı seçimlerine göre farklı işlevleri çağıran kontrol yapısı.
- **\_\_name\_\_ == "\_\_main\_\_"** Kontrolü: Programın doğrudan çalıştırıldığında main() fonksiyonunu başlatan Python konvansiyonu.

## 4.2 Görev Dağılımı

Projenin tüm aşamalarında genel olarak beraber çalışılmış olup proje bitimi kod incelemesi topluca yapılmıştır.

Gürel Bilgin: Veritabanı tasarımı, temel CRUD işlemleri, main fonksiyonu yazımı ve menü sistemi geliştirilmesi.

Gizem Yalçın: Dokümantasyon ve yardım dosyalarının hazırlanması, DersProgramı sınıfının geliştirilmesi, rapor yazımı.

Berkay Aras: Kısıtlama kontrolleri, kullanılabilirlik iyileştirmeleri, geribildirim mesajları kodlanması.

Ali Aksoy: Excel çıktı oluşturma modül yazımı, hata yönetimi mekanizmalarının oluşturulması.

### **4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri**

Veritabanında aynı kodlu dersler kaydedilirken çakışma oldu ve kaydedilemedi. Bu sorunu dersin kodunu değiştirerek çözdük.

Aynı isimli seçmeli dersler farklı dönemlerde farklı bölümlerde bulunduğundan bu derslerin de ders kodları bölüme uygun olacak şekilde değiştirildi.

### **4.4 Proje İsterlerine Göre Eksik Yönler**

Programdaki boş yerlere dersler koyuluyor ancak derslik ve hocanın uygunluk durumu kontrol edilmiyor.

## **5. TEST VE DOĞRULAMA**

### **5.1 Yazılımın Test Süreci**

Yazılımın doğruluğunu ve güvenilirliğini sağlamak amacıyla bir test uygulaması geliştirilmiştir. Bu uygulama, sistemde yer alan tüm bileşenleri test edebilecek şekilde tasarlanmış olup, tekrarlanabilir test senaryoları içermektedir.

Test süreci şu adımları içermektedir:

#### **1. Birim Testleri:**

- **Kullanıcı Yönetimi Testleri:** Öğrenci, öğretim üyesi ve yönetici ekleme/çıkarma işlemlerinin doğru çalıştığı kontrol edilmiştir.
- **Ders Yönetimi Testleri:** Derslerin eklenmesi, güncellenmesi ve silinmesi işlemlerinin beklenen sonuçları verdiği doğrulanmıştır.
- **Derslik Yönetimi Testleri:** Dersliklerin kapasite kontrolleri ve atama süreçleri test edilmiştir.



## 2. Entegrasyon Testleri:

- Veri tabanının doğru çalıştığı, ilişkisel bütünlük sağlandığı ve verilerin eksiksiz saklandığı kontrol edilmiştir.
- Derslerin belirlenen kısıtlamalara uygun şekilde planlandığı test edilmiştir.

## 3. Sistem Testleri:

- Kullanıcı arayüzü üzerinden veri girişleri gerçekleştirilerek, tüm sistemin uçtan uca doğru çalıştığı gözlemlenmiştir.
- Haftalık ders programı oluşturulmuş ve sistemin kısıtlamalara uygun şekilde dersleri atadığı doğrulanmıştır.

## 4. Tekrar Edilebilirlik:

- Test senaryoları otomatikleştirilmiş ve aynı testlerin farklı verilerle tekrar çalıştırılmasına imkan tanınmıştır.

## 5.2 Yazılımın Doğrulanması

Test sonuçlarına göre yazılımın doğruluk durumu aşağıdaki gibi değerlendirilmiştir:

### Tam ve doğru çalışan bileşenler:

- Kullanıcı kayıt, güncelleme ve silme işlemleri
- Ders ekleme, güncelleme ve silme işlemleri
- Derslik atama süreçleri
- Haftalık ders programı oluşturma algoritması

### Eksik veya hatalı çalışan bileşenler:

- Kullanıcıların aynı anda birden fazla cihazdan giriş yapması durumunda çakışma olduğu gözlemlenmiştir.
- Ders programı optimizasyonu için ileri düzey zamanlama algoritmalarının iyileştirilmesi gerekmektedir.
- Büyük veri kümeleriyle yapılan testlerde veri tabanı performansının artırılması gerektiği tespit edilmiştir.

Sonuç olarak, test süreci yazılımın büyük ölçüde doğru çalıştığını göstermiştir. Ancak, belirlenen eksikliklerin giderilmesi için ek geliştirmeler planlanmaktadır.

## **Grup üyeleri GitHub Hesapları:**

Gürel Bilgin: <https://github.com/GurelBilgin>

Gizem Yalçın: <https://github.com/gizemyalcinn>

Ali Aksoy: <https://github.com/aaksoy581>

Berkay Aras: <https://github.com/brkyaras>