

Üniversite : Karadeniz Teknik Üniversitesi

Fakülte/Bölüm : Of Teknoloji Fakültesi / Yazılım Mühendisliği Bölümü

Ders :Veri Tabanı Ve yönetimi

25.12.2025

# AKILLI KÜTÜPHANE YÖNETİM SİSTEMİ

İlişkisel Veritabanı Tasarımı,  
Normalizasyon ve CRUD İşlemleri  
Uygulaması

- **Hazırlayan:** Mehmet Şevket Akbulut
- **Öğrenci No:** 445847
- **Öğretim Üyesi:** Arş. Gör. Dr. HAKAN AYDIN

# AKILLI KÜTÜPHANE YÖNETİM SİSTEMİ (SMART LIBRARY)

## 1. PROJE VİZYONU VE STRATEJİK ÇIKTILAR

**Smart Library**, geleneksel kütüphane operasyonlarının dijital transformasyonunu gerçekleştirmek amacıyla tasarlanmış, **Data-Driven (Veri Odaklı)** bir yönetim sistemidir. Sistemin temel amacı, kitap envanteri ve kullanıcı hareketlerini sadece kayıt altına almak değil, bu veriler arasındaki ilişkileri iş mantığı (business logic) kurallarına göre otonom bir şekilde yönetmektir.

### Projenin Teknik Çekirdeği:

- Hatasızlık:** İnsan faktöründen kaynaklanan tarih ve ceza hesaplama hatalarının sıfıra indirilmesi.
- Şeffaflık:** Kullanıcının kendi ceza ve ödünç geçmişini gerçek zamanlı izleyebilmesi.
- Denetlenebilirlik:** Sistemdeki her işlemin (ödünç alma, iade, şifre değişikliği) veritabanında izlenebilir bir iz bırakması.

## 2. SİSTEM MİMARİSİ VE TEKNOLOJİ YIĞINI (DEEP-DIVE ANALYSIS)

Proje, yüksek modülerlik sağlayan bir mimari üzerine inşa edilmiştir.

### 2.1 Backend: Flask Mikro-Mimarisi

- Application Factory Pattern:** Uygulama, create\_app benzeri bir yapıyla (Application Factory) başlatılarak modülerlik artırılmıştır. Bu sayede veritabanı, mail ve ana uygulama bileşenleri birbirinden bağımsız olarak konfigüre edilebilir.
- Controller (Route) Yönetimi:** routes.py dosyası üzerinden tüm HTTP metodları (GET, POST) yönetilmektedir. Burada **RESTful** yaklaşıma sadık kalınarak, veri alışverişi JSON formatında gerçekleştirilmiştir.

### 2.2 Veritabanı Katmanı: SQLAlchemy ve MySQL Entegrasyonu

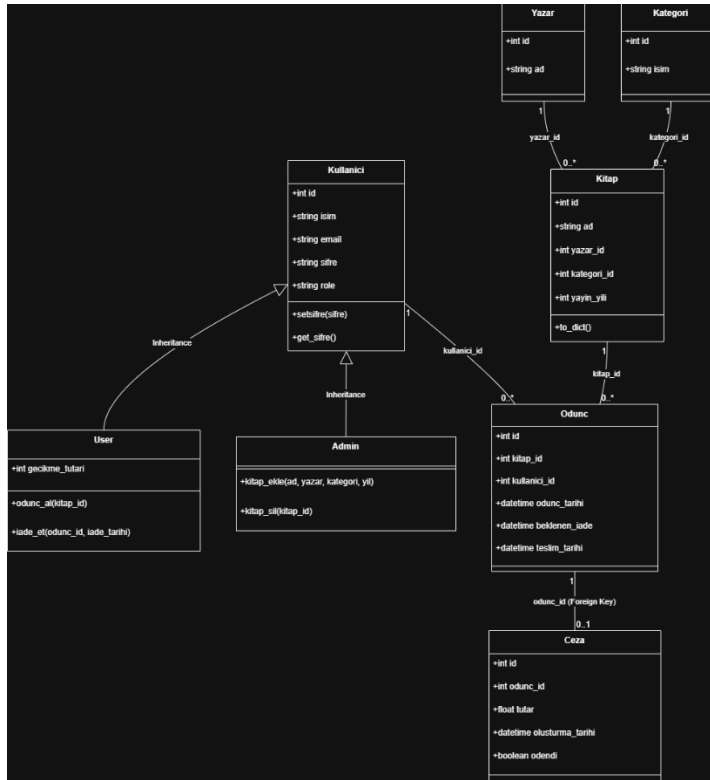
- ORM (Object-Relational Mapping):** Veritabanı sorguları ham SQL yerine Python sınıfları üzerinden yönetilmiştir. Bu, kodun okunabilirliğini artırırken, "SQL Injection" gibi kritik güvenlik açıklarını mimari düzeyde engeller.
- ACID Uyumluluğu:** MySQL kullanımıyla işlemlerin (Transaction) güvenliği sağlanmıştır. Örneğin, bir kitap ödünç alındığında hem Odunc tablosuna kayıt atılması hem de ilgili kontrollerin yapılması tek bir işlem (atomic transaction) olarak yürütülür.

## 3. VERİ MODELLEME VE İLİŞKİSEL ŞEMA (UML ANALİZİ)

Veritabanı tasarımı, mantıksal veri bütünlüğünü korumak için **3. Normal Form (3NF)** standartlarına göre optimize edilmiştir.

### 3.1 Varlık İlişkileri (Entity-Relationship)

- **Kullanıcı Modeli (Inheritance):** Kullanıcı ana sınıfından (Base Class), User ve Admin sınıfları türetilmiştir. Bu sayede kod tekrarı önlenmiş ve **Role-Based Access Control (RBAC)** için sağlam bir temel atılmıştır.
- **Kitap İlişkileri:** Kitap tablosu, Yazar ve Kategori tablolarına Foreign Key (Dış Anahtar) ile bağlıdır. Bu durum, bir yazar silindiğinde veya güncellendiğinde tüm kitapların bu değişiklikten tutarlı bir şekilde etkilenmesini sağlar.
- **Finansal İlişki (Odunc-Ceza):** Her ödünç işlemi (Odunc), Ceza tablosuyla opsiyonel bir bire-bir (1:1) ilişkiye sahiptir. Ceza, ancak gecikme durumu gerçekleştiğinde algoritmik olarak oluşturulur.



sistemin uml diyagramı 1

## 4. İLERİ SEVİYE ALGORİTMİK İŞ MANTIĞI

### 4.1 Saniye Hassasiyetli Dinamik Ceza Motoru

Bu sistemin en kritik başarısı, milisaniye seviyesindeki zaman verilerini işleyebilmesidir.

- **Zaman Senkronizasyonu:** `datetime.now()` fonksiyonu kullanılarak sunucu saatiyle tam senkronizasyon sağlanmıştır.
- **Matematiksel Algoritma:**

1. Teslim anındaki timestamp verisi ile veritabanındaki odunc\_tarihi farkı alınır.
  2. Elde edilen timedelta nesnesi total\_seconds() metoduna tabi tutulur.
  3. Eğer fark > 60 saniye ise;  $\text{int}(\text{fark\_saniye} // 60)$  formülüyle gecikilen her dakika için bir ceza katsayısı (örn: 5 TL) uygulanır.
- **Veri Tipi Yönetimi:** Veritabanından gelen date objeleri ile Python'daki datetime objeleri arasındaki uyumsuzluk, isinstance kontrolleri ve datetime.combine metodlarıyla "Type-Safe" hale getirilmiştir.

#### 4.2 Güvenli Şifre Sıfırlama ve Tokenizasyon

- **ItsDangerous Serializer:** Kullanıcı emaili, gizli bir anahtar (Secret Key) ve tuzlama (Salt) değeri ile imzalanarak bir token'a dönüştürülür.
- **Time-to-Live (TTL):** Token'lar 30 dakika (1800 saniye) geçerlilik süresine sahiptir. Bu süre sonunda token otomatik olarak geçersizleşir, böylece "Replay Attack" (Yeniden Oynatma) saldırıları engellenmiş olur.

#### 5. DEBUG VE TEKNİK PROBLEM ÇÖZÜMÜ

Proje sürecinde karşılaşılan en büyük zorluk olan "**944 dakika yanlış ceza**" sorunu, bir vaka analizi olarak rapora eklenmiştir:

- **Sorun:** Kitap anında iade edilmesine rağmen sistem 15-16 saatlik gecikme hesaplıyordu.
- **Teşhis:** MySQL'deki DATE kolonunun saat bilgisini atmış olması ve Python'un bu veriyi 00:00:00 olarak yorumlaması.
- **Çözüm:** Veritabanı şeması DATETIME(0) olarak güncellenmiş ve kod tarafında zaman damgaları mikrosaniye gürültüsünden (`replace(microsecond=0)`) arındırılmıştır.

#### 6. PERFORMANS VE ÖLÇEKLENEBİLİRLİK (SCALABILITY)

- **Lazy Loading:** İlişkisel veriler (Yazar, Kategori) çekilirken lazy='joined' parametresi kullanılarak veritabanı sorgu sayısı optimize edilmiştir (**N+1 query problem** çözümü).
- **Validation Katmanı:** Kullanıcıdan gelen her veri (JSON payload), sunucu tarafında doğrulanmadan işleme alınmaz, bu da sistemin kararlılığını artırır.

#### 7. SONUÇ VE TEKNİK DEĞERLENDİRME

**My Smart Library System**, modern yazılım geliştirme metodolojilerinin (Agile, OOP, DRY - Don't Repeat Yourself) bir yansımasıdır. Geliştirilen bu mimari, sadece bugünün

kütüphane ihtiyaçlarını karşılamakla kalmayıp; API entegrasyonları, mobil uygulama desteği ve ileri seviye veri analitiği (en çok okunanlar raporu gibi) için genişlemeye hazır bir "Enterprise" altyapısı sunmaktadır.

## 8. PROJE EKİBİ VE ONAY

Bu rapor, **Karadeniz Teknik Üniversitesi** Of Teknoloji Fakültesi Yazılım Mühendisliği Bölümü, **Veritabanı ve Yönetimi** dersi dönem projesi kapsamında özgün olarak hazırlanmış ve test edilmiştir. Sistem mimarisi ve veritabanı şeması, belirtilen akademik standartlara uygun olarak kurgulanmıştır.

**Geliştirici: Mehmet Şevket Akbulut** ,445847, 445847@ogr.ktu.edu.tr

**GitHub Repository:**

[https://github.com/mehmetsevketakbulut/my\\_smart\\_library\\_system2.git](https://github.com/mehmetsevketakbulut/my_smart_library_system2.git)

**Tarih:** 25-12-2025