Kütüphane Otomasyon Sistemi

Orazjemal Meredova  
Kocaeli Üniversitesi  
E-posta: pamukmeredova@gmail.com

# Özet

Bu proje, kütüphanelerdeki işlemleri dijital ortamda daha hızlı ve düzenli bir şekilde yönetmeyi amaçlayan kapsamlı bir kütüphane otomasyon sistemidir. Uygulama içerisinde yazar, kitap ve kategori ekleme, silme ve güncelleme gibi temel yönetim işlemleri yer almaktadır. Ayrıca kullanıcılara, sistem üzerinden kitap ödünç verme ve iade etme işlemlerini gerçekleştirilebilmektedir.  
  
Sistemin önemli özelliklerinden biri, geç iade edilen kitaplara karşı ceza puanı uygulamasıdır. Kullanıcılar kitapları zamanında teslim etmediklerinde sistem yöneticisi tarafından ceza puanı verilir. Bu puanlar birikerek belirli bir eşiğe ulaştığında, kullanıcı 'cezalı' rolüne geçirilir ve sistem yöneticisi bazı işlemleri yapmalarını kısıtlar. Bu sayede kitapların zamanında teslim edilmesi teşvik edilerek kaynakların verimli kullanımı sağlanır.

## Anahtar Kelimeler

Kütüphane otomasyonu, ceza sistemi, PHP, MySQL, kitap yönetimi, kullanıcı takibi

## 1. Giriş

Geleneksel kütüphane sistemlerinde işlemler genellikle manuel yöntemlerle yürütülmekte, bu durum zamanla verimliliğin düşmesine, hata oranlarının artmasına ve kullanıcı memnuniyetinin azalmasına yol açmaktadır. Özellikle kitap ödünç alma, iade etme, stok takibi ve kullanıcı kayıt işlemleri gibi temel işlevlerin elle yapılması hem zaman kaybına neden olmakta hem de bilgi karışıklıklarına zemin hazırlamaktadır. Bu sebeple, günümüzde dijital teknolojilerin sağladığı olanaklardan faydalanılarak kütüphane işlemlerinin otomasyon sistemleri aracılığıyla yürütülmesi büyük bir ihtiyaç haline gelmiştir.

Bu proje kapsamında geliştirilen **Kütüphane Otomasyon Sistemi**, kütüphane yetkililerinin, kütüphaneye kayıtlı üyelerin talep ettiği kitapları dijital ortamda hızlı ve etkin bir şekilde bulabilmesine ve bu kitapları uygunluk durumuna göre kullanıcıya ödünç verebilmesine imkân tanımaktadır. Sistem, kitap veritabanında arama yaparak, talep edilen kitabın mevcut olup olmadığını kısa sürede tespit edebilmekte; varsa, ilgili kullanıcıya ödünç verilme işlemi başlatılabilmektedir.

Ayrıca sistem, kitapların iade süreçlerini de yönetebilmekte; ödünç alınan kitabın teslim süresi aşıldığında, kullanıcıya belirlenen oranlarda **ceza puanı** uygulaması gerçekleştirilmektedir. Ceza puanları, sistemde otomatik olarak veya yönetici müdahalesiyle güncellenmekte, böylece kullanıcı davranışları izlenebilmektedir. Kullanıcının ceza puanı belirli bir eşiği —örneğin 100 puan— aştığında sistem, bu kullanıcının yeni kitap taleplerini geçici süreyle kısıtlamaktadır. Bu özellik, kitapların zamanında iadesini teşvik etmekte ve kütüphane kaynaklarının adil dağıtımını sağlamaktadır.

Sonuç olarak, bu otomasyon sistemi; işlemlerin dijital ortama taşınması sayesinde zaman ve emek tasarrufu sağlamanın yanı sıra, kaynak yönetimini daha şeffaf ve izlenebilir hale getirerek hem yöneticiler hem de kullanıcılar açısından önemli avantajlar sunmaktadır. Sistem, kütüphane hizmetlerinin kalitesini artırmakta, kullanıcı davranışlarını veri temelli olarak izlemeyi mümkün kılmakta ve bilgiye erişimi kolaylaştırarak öğrenme süreçlerini desteklemektedir.

## 2. Kullanılan Teknolojiler

Proje geliştirme sürecinde aşağıdaki teknolojiler tercih edilmiştir:

* **PHP**: Sunucu taraflı programlama dili olarak kullanılmıştır.
* **MySQL**: Veritabanı sistemi olarak tercih edilmiştir.
* **HTML, CSS ve Bootstrap**: Arayüz tasarımı için kullanılmıştır.

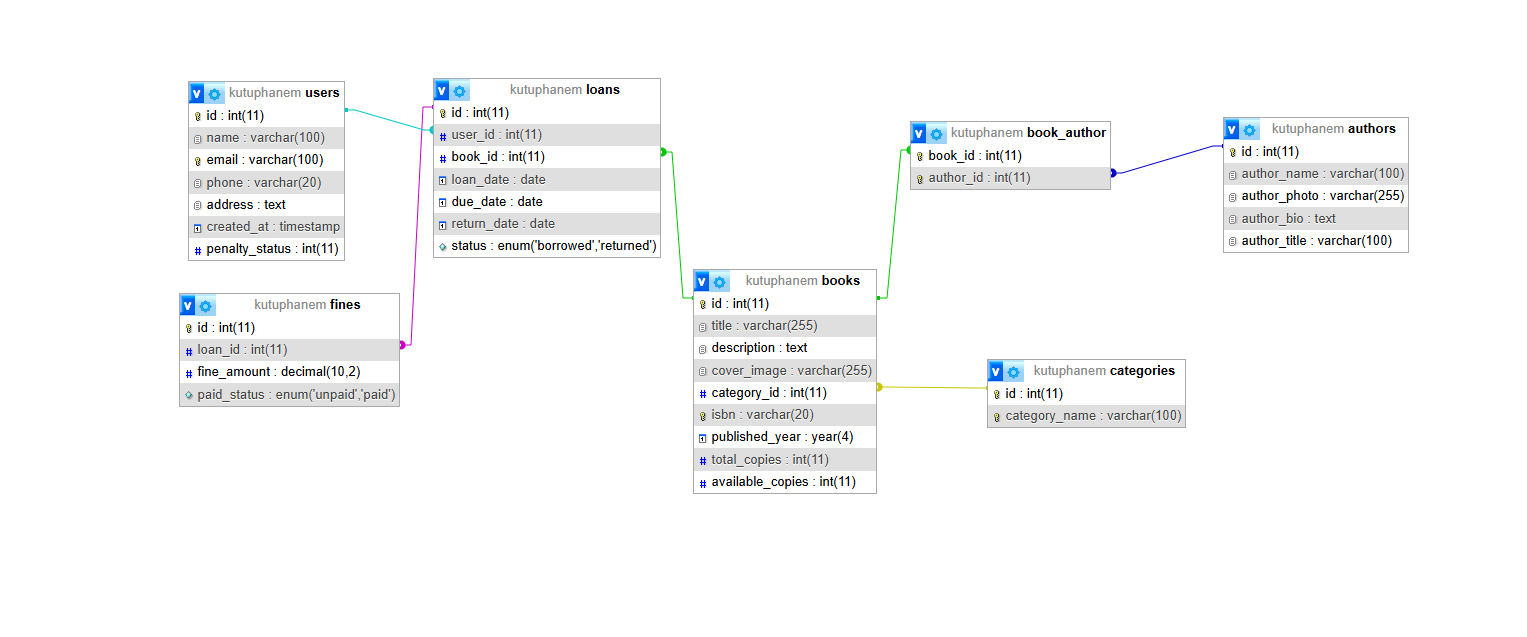
## 3. Sistem Özellikleri

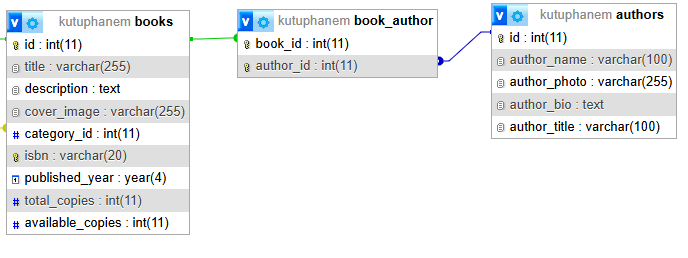
Sistem, kütüphane yöneticilerinin aşağıdaki işlemleri gerçekleştirmesine olanak tanımaktadır:

* Yazar Yönetimi
* Kitap Yönetimi
* Kategori Yönetimi
* Kitap Ödünç Alma ve İade Etme
* Ceza Puanı Sistemi

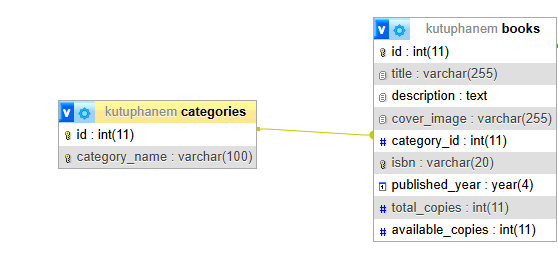
**4.** **Veritabanı Tasarımı ve İlişkisel Model**

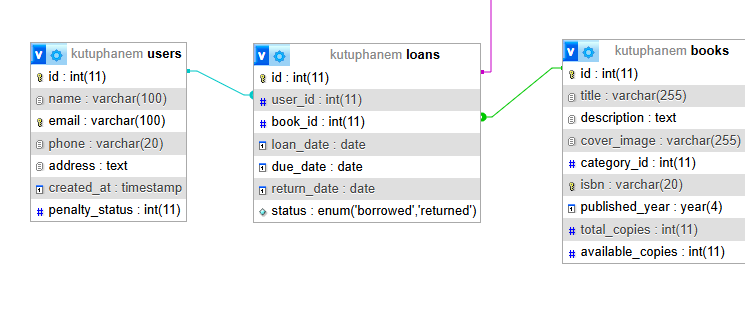
Uygulama, ilişkisel bir veritabanı modeli kullanmakta olup, sistemdeki bileşenler arasındaki bağlantılar kullanıcı hareketlerini ve kütüphane kaynaklarının takibini kolaylaştırmaktadır.

Şekil 1. Veritabanı Tablo İlişkileri  
  
**4.1. Yazar-Kitap İlişkisi**Her kitap bir veya birden fazla yazara ait olabilir. Bu çoktan çoğa (n:n) ilişki, **book\_authors** adlı ara tablo aracılığıyla gerçekleştirilmiştir.

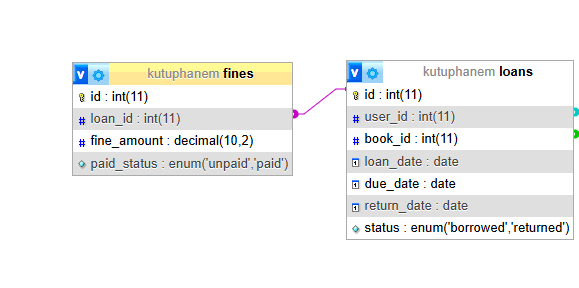
  
Şekil 2. Yazar ve Kitap İlişkisi

**4.2. Kategori-Kitap İlişkisi**Her kitap bir kategoriye aittir. Bu ilişki sayesinde kitaplar sistem içerisinde türlerine göre sınıflandırılabilir.

  
Şekil 3. Kategori ve Kitap İlişkisi   
  
**4.3. Kullanıcı-Ödünç Alma İlişkisi**Kullanıcı bilgileri users tablosunda saklanır. Bir kullanıcı birden fazla kitap ödünç alabilir. Bu ilişki loans tablosu ile sağlanır.

  
Şekil 4. Kullanıcı ve Ödünç Alma İlişkisi

**4.4. Ceza Sistemi**Eğer bir kitap geç iade edilirse sistemde ceza uygulanır. Bu cezalar fines tablosunda tutulur.

  
Şekil 5. Ceza Sistemi İlişkisi

**5. Ceza Puanı ve Cezalı Kullanıcı Yönetimi**

Ceza puanı sistemi, kitapların zamanında iadesini teşvik etmek amacıyla tasarlanmıştır. Sistem şu şekilde işlemektedir:

* Kitap geç iade edilirse kullanıcıya belirli bir ceza puanı eklenir.
* Ceza puanı eşiği aşıldığında kullanıcı “cezalı” rolüne geçirilir.
* Cezalı kullanıcılar yeni kitap ödünç alamaz.
* Ceza ödemesi gerçekleştikten sonra (kütüphaneye bir kitap bağışlama) kullanıcı normal rolüne geri döner.

Bu yapı sayesinde kullanıcılar ödünç aldıkları kitapları zamanında iade etmeye yönlendirilmekte ve sistemin sürdürülebilirliği sağlanmaktadır.

## 6. Sistemin Sağladığı Faydalar

* Yazar, kitap, kategori ve kullanıcı yönetimi merkezi bir sistem üzerinden kolaylıkla sağlanır.
* Ceza puanı uygulaması ile geç iade sorunlarının önüne geçilir.
* Kitap stoğu ve kullanıcı hareketleri anlık olarak takip edilebilir.
* Otomasyon ile işlem süresi kısalır ve hata oranı düşer.

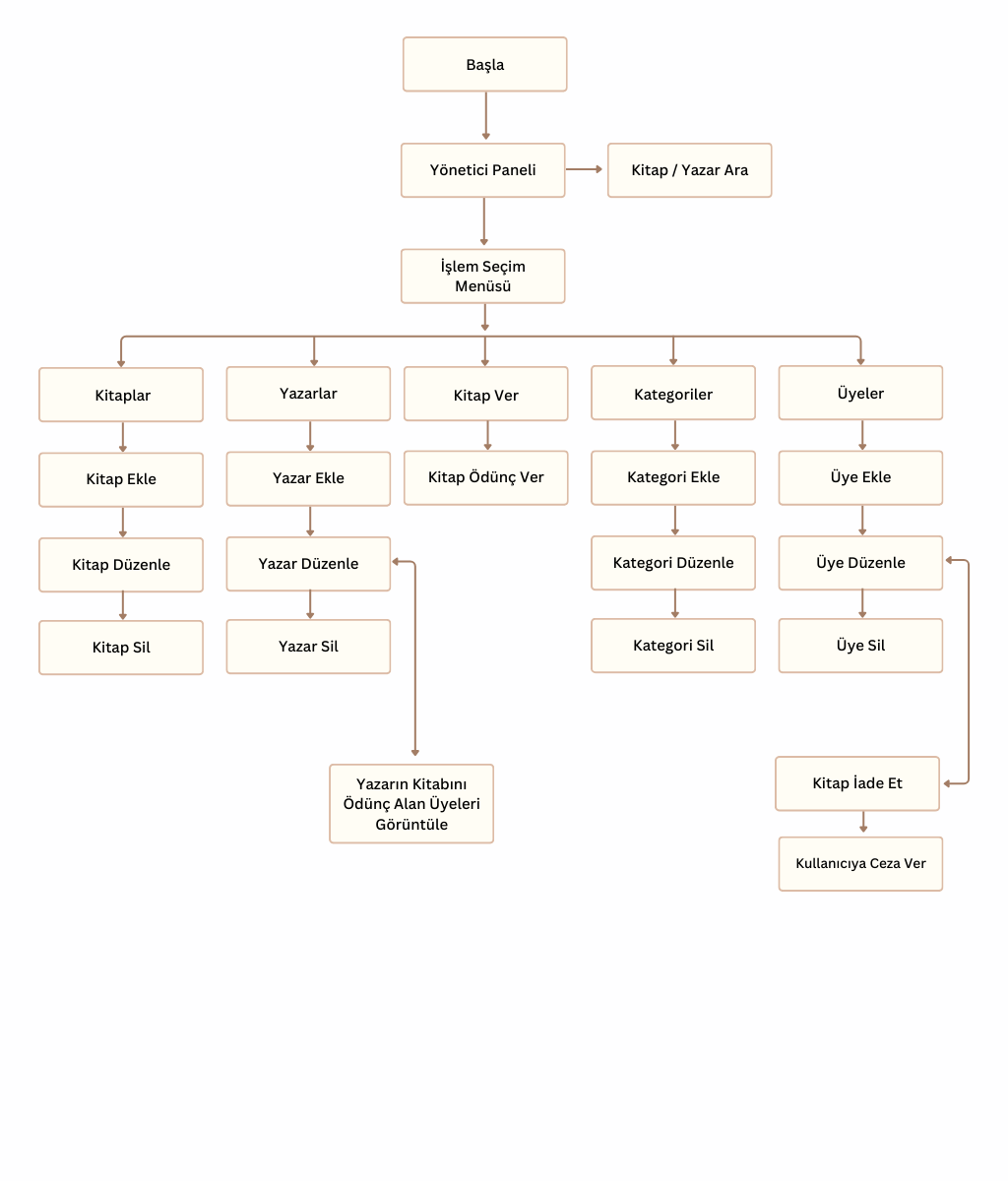
## 7.ER Diyagramı

Bu ER (İlişkisel Varlık) diyagramı, **Kütüphane Otomasyon Sistemi**nin veritabanı tasarımını özetler; her tablo bir varlığı, çizgiler / semboller ise aralarındaki ilişkileri gösterir.

Şekil 6. ER Dİyagramı

## 8. Akış Şeması

Bu akış şemasını hazırlarken, sistemde hem yöneticinin hem de kitap arayan bir kullanıcının izleyeceği adımları tek bir hiyerarşi altında toplamak istedim. Şemanın tepesi “Başla” düğümüyle açılıyor; yani program çalıştırıldığı anda doğrudan **Yönetici Paneli** ekranına düşüyoruz. buradan ya hızlı **Kitap/Yazar Arama** yapılabiliyor ya da **İşlem Menüsü**ne geçiliyor. Menü, beş temel modüle ayrılıyor: Kitaplar, Yazarlar, Kategoriler, Üyeler ve Kitap Ver/İade. Her modülde sırayla **Ekle → Düzenle → Sil** adımları var. Ödünç sürecinde “Kitap İade Et”ten sonra geç teslim kontrolüyle “Kullanıcıya Ceza Ver” kutusu devreye giriyor. Orta kısımda eklediğim “Yazarın kitabını ödünç alan üyeleri görüntüle” dalı da raporlama ihtiyacını karşılıyor. Böylece yöneticinin tüm CRUD ve ödünç işlemleri tek bakışta anlaşılır hale geliyor.

Şekil 7. Akış Şeması

## 8. Yapılan Araştırmalar

1. **Veritabanı Tasarımı**  
   Kitaplarla yazarlar arasındaki çok‑tan‑çoğa ilişkiyi nasıl kuracağımı anlamak için MySQL’in resmi dokümantasyonunda yer alan ilişki anlatımlarını inceledim.1 Bu kaynaklar sayesinde “ara tablo” yönteminin en doğru yol olduğunu gördüm ve book\_authors adında, yalnızca book\_id ile author\_id tutan bir köprü tablo oluşturdum.
2. **Aynı İşlemlerin Tekrarı**  
   CRUD kodlarında tekrarları azaltmak amacıyla php.net’in Nesne Yönelimli Programlama kılavuzunu okuyup Laracasts’teki “Refactor to Classes”2 mini serisini izledim. Bu içeriklerden öğrendiklerimle ortak veritabanı işlemlerini tek bir BaseModel sınıfına taşıdım; Book, User, Category gibi sınıflar bu temelden kalıtım alarak kod tekrarını büyük ölçüde ortadan kaldırdı.
3. **Kullanıcı Arayüzü**  
   Arayüzün hem sade hem de mobil uyumlu olması için Bootstrap’i tercih ettim. Karar vermeden önce Bootstrap’in resmi belgelerini ve Traversy Media’nın “Bootstrap Crash Course”3 videosunu inceledim. Bunlar, grid sistemi ve hazır bileşenlerle kısa sürede temiz bir görünüm oluşturabileceğimi gösterdi; ek JavaScript yazmaya gerek kalmadan tutarlı bir tasarım yakaladım.
4. **Kitap ve Yazar Arama**  
   Esnek arama fonksiyonu için Learn Microsoft’daki4 “Full-Text Search” makalesinden yardım aldım. Anlamadığım noktalarda Youtube videolarından da aldığım destekle hem kitap adına hem yazar adına göre hızlı arama yapabildim.
5. **Kod Düzeni**  
   Proje büyüdükçe klasör karmaşasını engellemek adına “PHP The Right Way”5 rehberinde önerilen dizin yapısını uyguladım. Kodları class, css, include klasörlerine bölerek her dosyaya tek bir sorumluluk verdim; böylece hem bakım kolaylaştı hem de yeni bir geliştiricinin projeyi anlaması hızlandı.

## Kaynaklar

1. PHP - [https://www.php.net](https://www.php.net/)
2. MySQL Dökümasyonü - [https://dev.mysql.com](https://dev.mysql.com/)
3. Bootstrap Kütüphanesi - [https://getbootstrap.com](https://getbootstrap.com/)
4. Learn Microsoft - <https://learn.microsoft.com/>
5. PHP The Right Way - <https://phptherightway.com/>