# Bilgisayar Mühendisliği Bölümü

# Yazılım Mühendisliği Dersi

# Proje Raporu

**Proje Konusu:** Kitapların kaydedilmesi, aranması, ödünç alınıp, iade edilebilmesi ve silinmesi gibi işlemleri gerçekleştirebilen Kütüphane Otomasyon Programı

Öğretim Üyesi: Levent Karacan

Öğrenci Adı-Soyadı: Şule Yücelbaş

Öğrenci No: 222523222

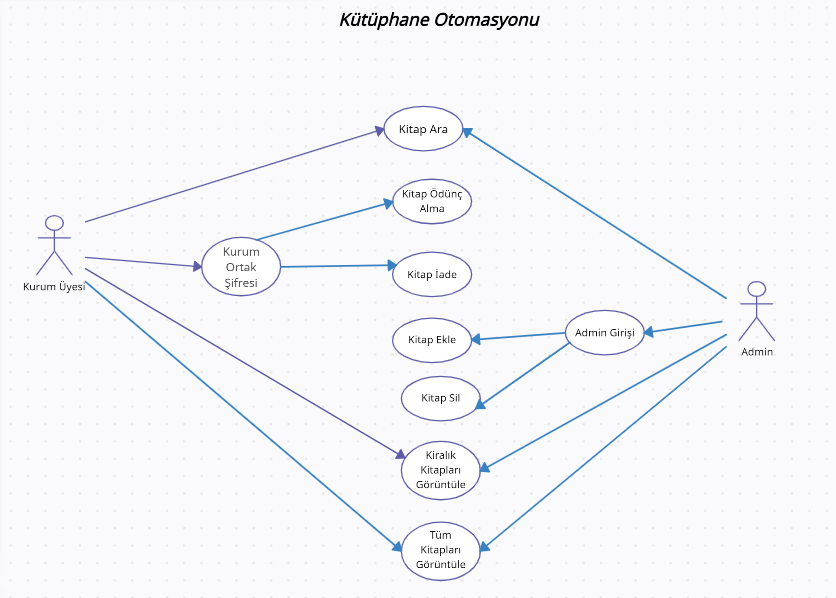
Öğrenci e-mail: syucelbas.mdbf22@iste.edu.tr

# Proje İçeriği ve Çözümlemeler

Bu projede, bir kütüphane otomasyonu uygulamasının basit bir grafik ara yüzü oluşturulmaya çalışılmıştır. Uygulama kitap eklemek, kitap silmek, kitap aramak, kitap ödünç almak, kitap iade etmek gibi işlevleri yerine getirebilmektedir. Bu proje, nesneye yönelik programlama (OOP) prensiplerine uygun bir şekilde Python dilinde hazırlanmıştır. Projeye ait kod, iki sınıf olan LibrarySystem ve LibraryApplication sınıflarını içermektedir.

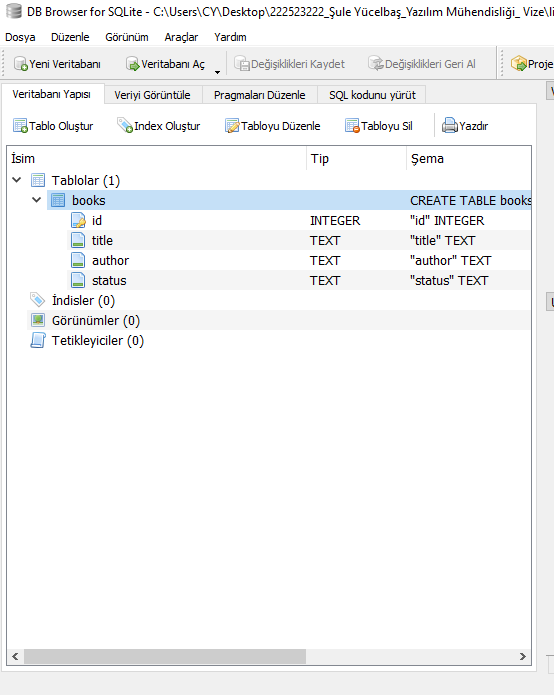
LibrarySystem sınıfı, veri tabanı bağlantısı ve veri tabanı işlemlerini gerçekleştiren metotları içerir. LibraryApplication sınıfı, Tkinter kullanarak bir kullanıcı ara yüzü oluşturur ve bu ara yüzdeki butonlara bağlı işlevleri gerçekleştirir. Aynı zamanda LibrarySystem sınıfını kullanarak kitap ekleme, kitap silme, kitap arama gibi işlemleri gerçekleştirir. Böylece yazılan kod, nesne yönelimli programlama prensiplerini takip eder ve sınıflar arasında ilişkiyi sağlar. Her sınıf, özel metotlar ve veri alanları içerir ve ilgili işlemleri gerçekleştirir. Ayrıca, sınıflar arasında mesajlaşma ve işbirliği de vardır.

Admin olarak giriş yapıldığında, kitap ekleme, kitap silme, tüm kitapları görüntüleme gibi ek işlevler kullanılabilir hale gelir. Kullanıcı olarak giriş yapıldığında, kitap arama, kiralanan kitapları görüntüleme, kitap ödünç alma ve kitap iade etme gibi işlevler kullanılabilir hale gelir. Kodun çalışması için Tkinter ve sqlite3 modüllerinin kurulu olması gerekmektedir. Bu kodda kullanıcı rolü, admin şifresi (admin123) veya kurum ortak şifresi (555) ile doğrulanır. Projeye ait use case diyagramı Şekil 1’de görülmektedir.

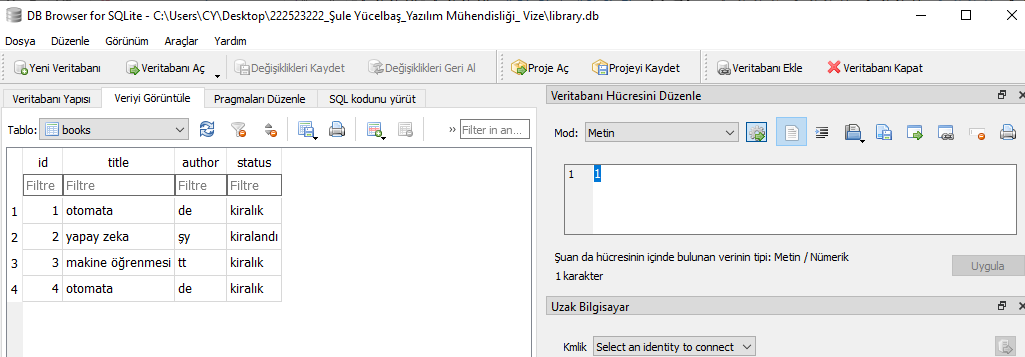


**Şekil 1.** Kütüphane Otomasyonu use case diyagramı

LibrarySystem sınıfı, kitapları veri tabanında yönetmek için gerekli olan işlevleri içerir. sqlite3 modülünü kullanarak library.db adında bir veri tabanı oluşturur ve kitapları bu veri tabanında saklar. LibraryApplication sınıfı, grafik ara yüzünü oluşturmak ve kullanıcının kitapları yönetmesine olanak sağlamak için tkinter modülünü kullanır. Kodu çalıştırdıktan sonra yapılan eklemeler ve çıkarmalarla SQLite3 veri tabanı programında oluşan books tablosu Şekil 2’de ve 3’te görülmektedir.



**Şekil 2.** Kütüphane Otomasyonu Database browser

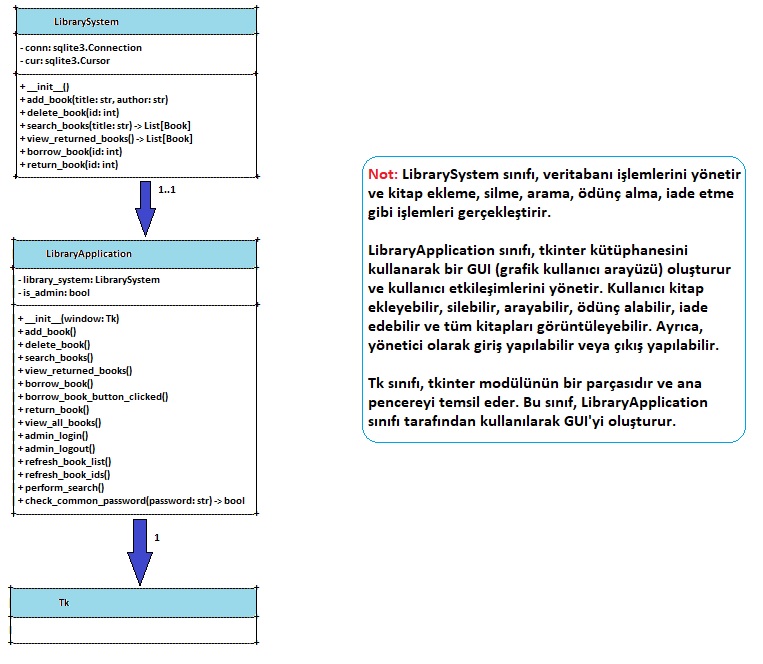


**Şekil 3.** Kütüphane Otomasyonu Database browserda açılmış books tablosu

Ara yüz, kitap eklemek, silmek, aramak, ödünç almak, iade etmek gibi işlemleri gerçekleştirebilen düğmeler ve metin giriş alanları içerir. Projeye ait kodda yer alan metotların açıklamaları şu şekildedir:

1. \_\_init\_\_(self): LibrarySystem sınıfının yapıcı metodudur. SQLite veritabanı ile bağlantı kurar ve books adında bir tablo oluşturur.
2. add\_book(self, title, author): Bir kitabı veritabanına ekler. title ve author parametreleriyle kitap adı ve yazarını alır.
3. delete\_book(self, id): Verilen bir kitabın kaydını veritabanından siler. id parametresiyle kitap kimliğini alır.
4. search\_books(self, title): Verilen bir kitap adına göre kitapları arar ve bulunan kitapların listesini döndürür. Eğer title parametresi boş ise tüm kitapları döndürür.
5. view\_returned\_books(self): Kiralık olmayan kitapların listesini döndürür.
6. borrow\_book(self, id): Bir kitabı ödünç alındı durumuna getirir. id parametresiyle kitap kimliğini alır.
7. return\_book(self, id): Bir kitabın ödünç alındı durumunu kiralık durumuna getirir. id parametresiyle kitap kimliğini alır.
8. \_\_init\_\_(self, window): LibraryApplication sınıfının yapıcı metodudur. Arayüzün oluşturulduğu Tkinter penceresini ayarlar.
9. add\_book(self): Kitap ekleme işlemini gerçekleştirir. Giriş alanlarından kitap adı ve yazarını alır, library\_system nesnesi üzerinden add\_book metoduyla kitabı veritabanına ekler.
10. delete\_book(self): Seçili kitabı siler. Listbox'ta seçili kitabın kimliğini ve durumunu kontrol eder. Eğer kitap kiralandıysa silme işlemi gerçekleştirmez ve bir hata mesajı gösterir.
11. search\_books(self): Kitapları aramak için kullanılır. Giriş alanından kitap adını alır, library\_system nesnesi üzerinden search\_books metoduyla kitapları arar ve sonuçları Listbox'ta gösterir.
12. view\_returned\_books(self): Kiralık olmayan kitapları görüntüler. library\_system nesnesi üzerinden view\_returned\_books metoduyla kiralık olmayan kitapları alır ve sonuçları Listbox'ta gösterir.
13. borrow\_book(self): Kitap ödünç alma işlemini gerçekleştirir. Listbox'ta seçili kitabı alır, library\_system nesnesi üzerinden borrow\_book metoduyla kitabın durumunu değiştirir.
14. borrow\_book\_button\_clicked(self): Bu metot, "Kitap Ödünç Al" düğmesine tıklandığında çağrılır. Öncelikle, is\_admin değişkenine bakarak kullanıcının yönetici mi yoksa normal kullanıcı mı olduğunu kontrol eder. Eğer kullanıcı yönetici ise, borrow\_book metodu çağrılır. Yönetici değilse, kullanıcıdan bir parola istenir ve bu parola check\_common\_password metoduna gönderilir. Eğer parola doğru ise, borrow\_book metodu çağrılır. Aksi takdirde kullanıcıya "Geçersiz Kurum Ortak Şifresi!" şeklinde bir hata mesajı gösterilir.
15. return\_book(self): Bu metot, "Kitap İade" düğmesine tıklandığında çağrılır. Öncelikle, kullanıcının yönetici olup olmadığını kontrol eder. Eğer yönetici ise, return\_book metodu doğrudan çağrılır ve kitap iade edilir. Yönetici değilse, kullanıcıdan bir parola istenir ve bu parola check\_common\_password metoduyla kontrol edilir. Eğer parola doğru ise, return\_book metodu çağrılır ve kitap iade edilir. Aksi takdirde kullanıcıya "Geçersiz Kurum Ortak Şifresi!" şeklinde bir hata mesajı gösterilir.
16. view\_all\_books(self): Bu metot, "Tüm kitapları gör" düğmesine tıklandığında çağrılır. search\_books metodu çağrılarak tüm kitaplar getirilir ve Listbox'a eklenir.
17. admin\_login(self): Bu metot, "Giriş Yap" düğmesine tıklandığında çağrılır. Kullanıcının girdiği parola, "admin123" ile karşılaştırılır. Eğer parola doğru ise, is\_admin değişkeni True olarak ayarlanır ve yönetici girişi gerçekleşir. Düğme metni "Çıkış Yap" olarak değiştirilir ve admin\_logout metodu çağrıldığında yönetici çıkışı yapılır.
18. admin\_logout: Bu metot, "Çıkış Yap" düğmesine tıklandığında çağrılır. is\_admin değişkeni False olarak ayarlanır ve yönetici çıkışı gerçekleşir. Düğme metni "Giriş Yap" olarak değiştirilir ve admin\_login metodu çağrıldığında yönetici girişi yapılır.
19. refresh\_book\_list: Bu metot, kitap listesini güncellemek için kullanılır. Öncelikle Listbox temizlenir ve search\_books metodu çağrılarak tüm kitaplar getirilir ve Listbox'a eklenir.
20. refresh\_book\_ids(self) metodu, kitapların ID'lerini güncellemek için kullanılır. Bu metot, tüm kitapları sorgular ve her bir kitabın ID'sini index değeriyle günceller. Başlangıç değeri 1'den başlayarak kitapları sıralar ve her kitabın ID'sini günceller. Bu işlem, kitaplar silindiğinde veya yeni kitaplar eklediğimizde ID'lerin doğru bir şekilde sıralanmasını sağlar.
21. perform\_search(self): Bu metot, arama yapmak için kullanılır. title değişkenine girilen kitap adını alır ve search\_books metodunu çağırarak ilgili kitapları bulur ve listeler.
22. check\_common\_password(self, password): Bu metot, kurum ortak şifresini kontrol etmek için kullanılır. password parametresi olarak girilen şifreyi kontrol eder ve eşleşip eşleşmediğini döndürür. Şu anda kabul edilen şifre "555" olarak belirlenmiştir.

Kütüphane Otomasyonu projesine ait UML class diyagramı Şekil 4’de görülmektedir.

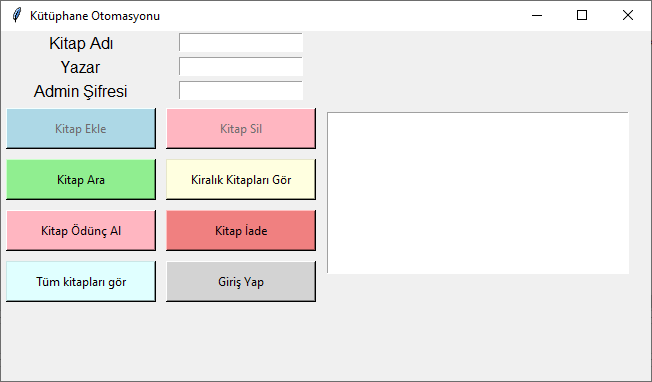


**Şekil 4.** Kütüphane Otomasyonu UML class diyagramı

Hazırlamış olduğum kütüphane otomasyon programının bazı güçlü yanları aşağıda belirtilmiştir:

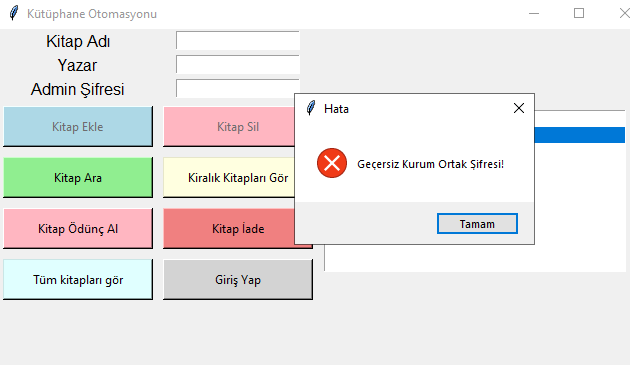
* Sınıf tabanlı bir tasarıma sahiptir: Kod, sınıf tabanlı bir yaklaşımla düzenlenmiştir. LibrarySystem ve LibraryApplication sınıfları, kütüphane işlemlerini ve GUI arayüzünü yönetir. Bu, kodun daha düzenli ve sürdürülebilir olmasını sağlar.
* SQLite veritabanı kullanımı: LibrarySystem sınıfı, SQLite veritabanını kullanarak kitapların depolanmasını ve işlenmesini sağlar. Bu, veri kalıcılığını sağlar ve verilerin kaybolmasını önler.
* Tkinter kullanımı: LibraryApplication sınıfı, Tkinter kütüphanesini kullanarak basit bir GUI (grafik kullanıcı arayüzü) oluşturur. Bu, kullanıcıların kitap eklemesini, silmesini, aramasını ve ödünç almasını/iade etmesini sağlar.
* Kullanıcı yetkilendirmesi: admin\_login ve admin\_logout yöntemleri, admin kullanıcısının giriş yapmasını ve çıkış yapmasını sağlar. Bu, sadece yetkilendirilmiş kullanıcıların kitapları silmesine ve eklemesine izin verir.
* Veri girişi doğrulaması: borrow\_book\_button\_clicked ve return\_book yöntemleri, kurum ortak şifresinin doğruluğunu kontrol eder. Bu, sadece doğru şifreyi giren kullanıcıların kitapları ödünç almasına ve iade etmesine izin verir.
* Kitap listesinin güncellenmesi: refresh\_book\_list ve refresh\_book\_ids yöntemleri, kitap listesini günceller ve kitapların ID'lerini sıralar. Bu, kullanıcıların herhangi bir değişiklik olduğunda güncel bir kitap listesi görmesini sağlar.
* Hata mesajları: messagebox modülü kullanılarak hata mesajları gösterilir. Bu, kullanıcılara hatalı girişler veya işlemler hakkında geri bildirim sağlar.

Bu iyi yanlar, projeye ait kodun okunabilir, yönetilebilir ve kullanıcı dostu olmasını sağlamaktadır. Ayrıca, veri tabanı kullanımı ve doğrulama işlemleri gibi güvenlik önlemleri de mevcuttur. Projeye ait grafik ara yüzü Şekil 5’de görülmektedir.



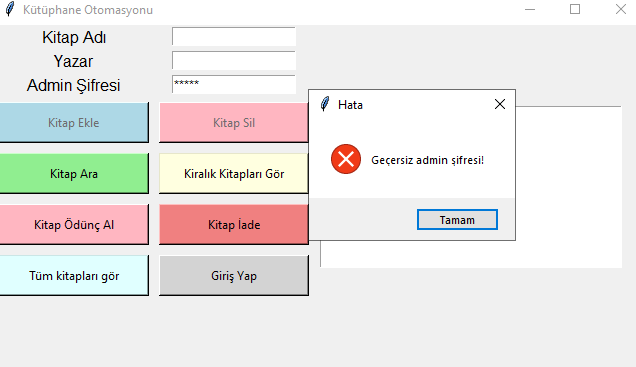
**Şekil 5.** Kütüphane Otomasyonu grafik ara yüzü

Programda, messagebox modülü kullanılarak hata mesajları ve girişler gösterilir. Bu, kullanıcılara hatalı girişler veya işlemler hakkında geri bildirim sağlamaktadır. Aşağıda, hatalı girişler sonucunda alınan hata mesajları gösterilmektedir. Şekil 6, hatalı kurum ortak şifresi girildiğinde karşılaşılan hata mesajıdır.



**Şekil 6.** Yanlış kurum ortak şifresi girişi sonucunda alınan hata mesajı

Şekil 7 hatalı admin şifresi girildiğinde karşılaşılan hata mesajıdır.



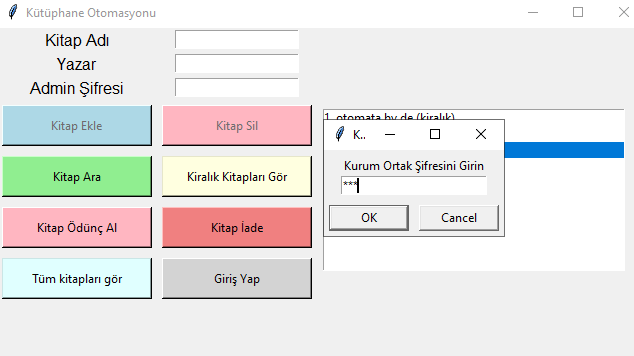
**Şekil 7.** Yanlış admin şifresi girişi sonucunda alınan hata mesajı

Şekil 8, kiralanmış bir kitap admin tarafından silinmeye çalışıldığında karşılaşılan hata mesajıdır.



**Şekil 8.** Kiralanmış bir kitap admin tarafından silinmeye çalışıldığında karşılaşılan hata mesajı

Son olarak Şekil 9, ortak kullanıcı bir kitabı kiralamak veya iade etmek istediğinde karşılaştığı giriş mesajını göstermektedir.



**Şekil 9.** Ortak kullanıcı bir kitabı kiralamak veya iade etmek istediğinde karşılaştığı giriş mesajı