

# YAZILIM LABORATUVARI II PROJE 1

Abdulcelil Kurt  
170202001  
celilkurt82@gmail.com

## PROBLEM TANIMI

Yönteici ve kullanıcı şeklinde iki farklı rolün olduğu bir kütüphane uygulamasının sunucu ve istemci kısımlarını yapmamız bekleniyor. Yöneticinin kitap ekleme-silme ve sistemdeki kitap veya kullanıcı kayıtlarını görme yetkisi var. Kullanıcının ise belli kurallara bağlı olarak kitap ödünç alma, geri verme ve kitap arama yetkisi var. Bunların yanında yöneticinin ödünç alma kurallarının uygulanıp uygulanmadığını görmek için zaman atlama seçeneği var. Yönetici bu özelliklerle, bir kullanıcının elinde zamanı geçmiş kitap olduğu halde kitap alabilip alamadığını görebilecek.

## Özet

**Yönetici ve kullanıcı kısımlarından oluşan kütüphane uygulamasıdır. Uygulamanın veritabanı ile bağlantısını Spring Boot ile oluşturulan RESTful servis sağlıyor. İstemci kısmında ise Angular framework kullanılmıştır.**

## GİRİŞ

Uygulamaya tarayıcıdan localhost:4200/ adresinden erişilir. Herhangi bir işlem yapabilmek için giriş yapmak gerekir. Giriş yaptıktan sonra, başlıkta kullanıcının rolüne göre butonlar ortaya çıkar. Kullanıcı yönetici ise kitap listesini, ödünç alınmış kitapların listesini veya kullanıcıların listesini görebilir, kitap ekleyebilir ve sistemin tarihini değiştirebilir eğer değilse kitap listesini veya ödünç aldığı kitapların listesini görebilir.

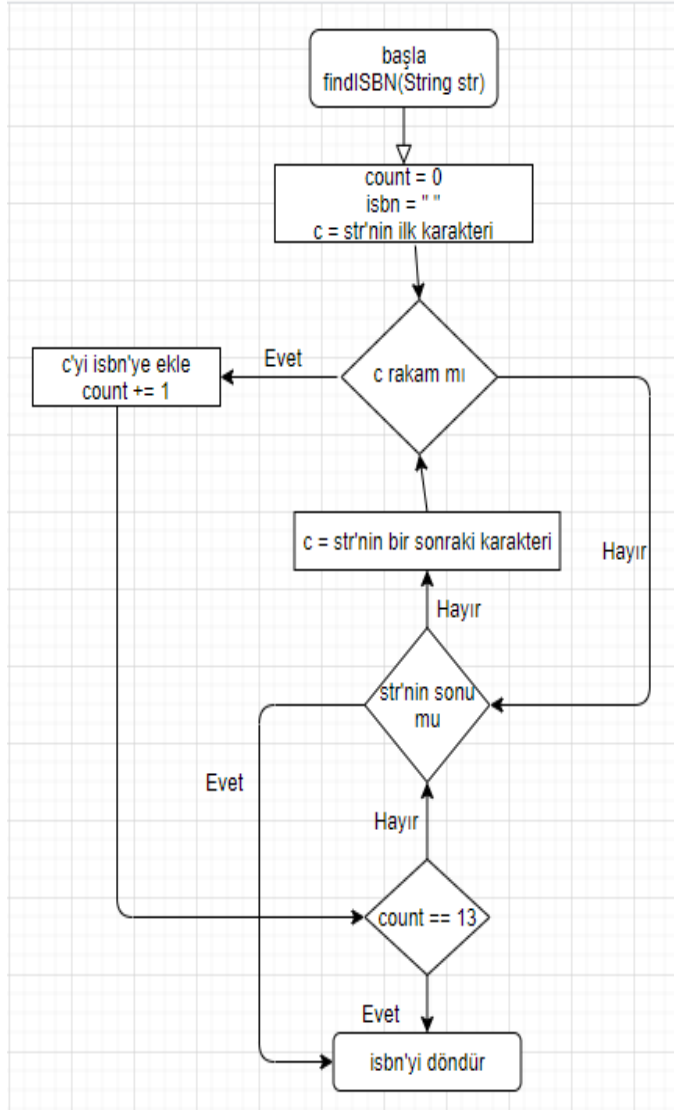
## YAPILAN ARAŞTIRMALAR

Projenin başlarındayken REST servisi Postman'den kontrol ediyordum. İstemci kısmına geçtiğimde Postman'de karşılık bulan isteklerin karşılık bulmadığını gördüm, sunucu kısmını uzun bir süre kurcaladıktan sonra istemci kısmındaki hatalar için tarayıcının 'console' kısmına bakmam gerektiğini öğrendim. Console'da belirtilen hatayı araştırdığımda bunun Angular framework'e özgü bir hata olduğunu ve çözümü için sunucuya yapılan isteklerin proxy aracılığı ile yapılması

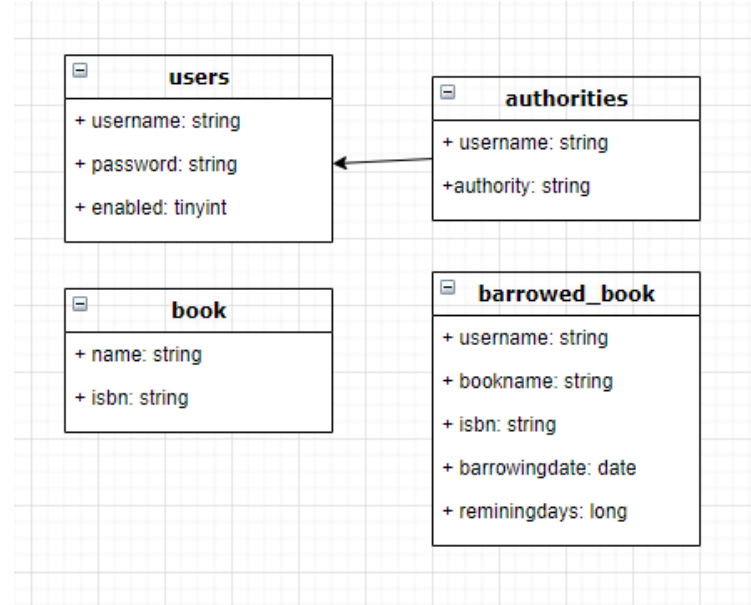
gerektiğini öğrendim. Sunucuya proxy ile bağlanmak için package.json dosyasında birkaç değişiklik yapmam yetti ve bundan sonra sunucuyu aktifleştirmek için 'npm start' komutunu kullanmaya başladım ve sorun çözülmüş oldu.

## AKIŞ ŞEMASI

findISBN(str) metodu için.



## VERİ TABANI DİYAGRAMI



'users' tablosu kullanıcı kayıtlarının tutulduğu tablodur. 'username' primary key dir. 'authorities' tablosunda da 'username' foreign key dir ve 'users' tablosunu referans alır.

'book' tablosunda kütüphanedeki kitapların kaydı tutulur. 'barrowed\_book' tablosunda ödünç alınan bütün kitapların kaydı tutulur.

'book' ve 'barrowed\_book' listesinde 'isbn' primary key dir.

## GENEL YAPI

Veri tabanında tutulan kayıtlara istemci kısmında yapılan http request'ler aracılığı ile erişilir. Yapılacak herhangi bir istekle (user kaydı oluşturma hariç) kullanıcı bilgilerinin gönderilmesi zorunludur. Aksi takdirde erişim yetkisi yoktur cevabı dönecektir. Sunucu tarafında yapılan istekler controllerlar aracılığıyla yakalanır ve cevap döndürülür.

```
@GetMapping(path="/books" )
public List<Book> getBooks() {
    return (List<Book>) bookRepository.findAll();
}
```

Yukarıdaki metod 'localhost:8080/api/books' adresine yapılan http get request'e karşılık 'book' tablosunda tutulan bütün kayıtları döndürür.

İstemci kayıtların listelenmesini, silinmesi, güncellenmesini veya yeni kayıt eklenmesini sunucuya yaptığı http requestlerle yapar.

```
httpClient.get<User>('http://localhost:4200/api/login/'+  
username,{headers})
```

Yukarıdaki kod belirtilen adrese yaptığı get request'ten karşılık olarak sadece rolü'nün tanımlı olduğu bir 'User' objesi döndürür çünkü diğer kullanıcı bilgilerini(username ve password) göndermeden yapılan bir request hata döndürür.

## Belli başlı işlemler

Kullanıcı '/login' ve '/register' sayfalarına dışındaki herhangi bir sayfaya erişebilmek için giriş yapması gerekir. Sayfalara erişimi 'authGuardService' adındaki ts dosyasıyla denetlenir.

### Giriş - 'localhost:4200/login'

Giriş için kullanıcı adı ve şifre alınır ve '/api/login' adresine bu bilgilerle istekte bulunulur. Giriş başarılı ise daha sonra kullanmak üzere username, password ve role 'sessionStorage' da tutulur. Header'da gösterilecek butonlar 'sessionStorage'de tutulan role bilgisine göre listelenir.

### Kitap listeleme, Ödünç alma veya Kitap silme - 'localhost:4200/books'

'api/books' adresine yapılan get request'e karşılık kitapların listesi alınır ve bu kitaplar istemci kısmında bir tablo aracılığıyla kullanıcıya listelenir.

Giriş yapan yönetici ise her bir kitabın karşısında

'delete' butuno çizdirilir bu buton aracılığı ile '/api/books/{isbn}' adresine kitabın isbn'si parametre olacak şekilde bir istekte bulunulabilir. Bu işlemin sonucunda kitap kaydı silinir.

Giriş yapan kullanıcı ise her bir kitabın karşısında ödünç al butonu çizdirilir. Butona tıklandığında önce '/api/borrowedbooks' adresine istemcide username, bookname ve isbn ile geçici olarak oluşturulan bir borrowedbook objesi parametre olacak şekilde bir POST request'te bulunulur. Sunucu kısmında 'borrowCheck' metodu ile kullanıcının elindeki kitap sayısı ve iade tarihi geçen kitap olup olmadığı sorgulanır. Eğer ödünç kriterleri sağlıyorsa gönderilen kitabın ödünç alınma tarihi ve kalan zamanı atanır ve borrowed\_book tablosunda yeni kayıt oluşturulur. Sonra 'api/books/{isbn}' adresine ödünç alınan kitabın isbn'si parametre olacak şekilde http DELETE isteğinde bulunulur. İstekle birlikte kitap 'book' tablosundan silinir.

### Ödünç alınan kitapları listeleme - 'localhost:4200/borrowedbooks'

Giriş yapan yönetici ise sunucuya 'api/borrowedbooks' get request yapılır. Bu isteğin sonucunda ödünç alınan bütün kitaplar güncel kalan zamanları hesaplanarak döndürülür ve bir tablo aracılığı ile kullanıcıya listelenir. Giriş yapan kullanıcı ise sunucuya istekte bulunan kullanıcının adı parametre olacak şekilde '/api/borrowedbooks/{username}' get request yapılır. İstek sonucunda istekte bulunan kullanıcının ödünç aldığı kitaplar güncel kalan günler bilgisi ile döndürülür ve tablo aracılığı ile listelenir.

Giriş yapan kullanıcı ise 'Return' butonu çizdirilir. Bu buton aracılığı ile ödünç alınan kitap resim okutularak veya form'a isbn adresi girilerek 'api/borrowedbooks/{isbn}' adresine iade edilecek kitabın isbn'si parametre olacak şekilde DELETE request'te bulunulur. Bu işlem sonucunda kitap 'borrowed\_book' tablosundan silinerek 'book' listesine eklenir. Sonuçta kitap iade edilmiş olur.

## Kitap ekleme - 'localhost:4200/addbook'

Resim seçildikten sonra çizdirilen 'Scan' butonu aracılığı ile 'api/book/readphoto' adresine seçilen resim parametre olacak şekilde get request'te bulunulur ve sunucu kısmında gelen dosya geçici olarak yazdırıldıktan sonra tesseract ocr ile okunur ve bulunan text parse edilerek bulunan isbn isteğe karşılık döndürülür. isbn illa resim okutularak bulunması gerekmez form'a da girilebilir. Kitap ismi ve isbn girildikten sonra 'Submit' butonu ile '/api/books' adresine girilen bilgilerden oluşturulan book objesi parametre olacak şekilde get request'te bulunulur. Bu işlemle gönderilen kitap sunucu kısmında 'book' tablosuna eklenir. 'Create Books' butonu ile örnek kitaplar oluşturulabilir.

## Tarih değiştirme - 'localhost:4200/changedate'

Kullanıcıdan alınan sayıyı parametre olarak göndererek sunucuya bir istekte bulunulur karşılığında güncel tarihin string hali alınır ve bu tarih sayfanın sağ üst köşesinde gösterilir.

## Kaynaklar

Mysql ile giriş için

<https://www.websparrow.org/spring/spring-boot-spring-security-with-jpa-authentication-and-mysql>

Tablolara DataSource objesi ile erişmek için

<https://www.journaldev.com/2509/java-datasource-jdbc-datasource-example>

MultiPart File dosyası işlemleri için

<https://medium.com/@ashan.lakmal/tesseract-simple-java-optical-character-recognition-aab9f23a8f94>

Angular framework'te fotoğraf yüklemek ve göndermek için

<https://medium.com/@rameez.s.shaikh/upload-and-retrieve-images-using-spring-boot-angular-8-mysql-18c166f7bc98>

<https://w3path.com/new-angular-8-file-upload-or-image-upload/>

Spring Boot Basic Auth için

<https://medium.com/@rameez.s.shaikh/angular-7-spring-boot-basic-authentication-example-98455b73d033>

Angular Auth için

<https://www.javainuse.com/spring/ang7-login>

Tarih işlemleri için

<https://stackoverflow.com/questions/51299944/get-current-date-with-yyyy-mm-dd-format-in-angular-4>

Backend'e proxy ile erişebilmek için

<https://medium.com/better-programming/setup-a-proxy-for-api-calls-for-your-angular-cli-app-6566c02a8c4d>