**KÜTÜPHANE UYGULAMASI**

*Mert BİLGİÇ – Nurdan VAYNİ*

[bilgic.mert44@gmail.com](mailto:bilgic.mert44@gmail.com) - nurdannvaynii0@gmail.com

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

**Özet**

Sunucu-istemci mimarisi ve görüntü işleme algoritmaları kullanılarak kütüphane sisteminin uygulanması amaçlanmaktadır.

**1. Giriş**

Sisteme giriş için kullanıcının üye olması gerekmektedir. Üyelik işlemleri sırasında kullanıcının seçebileceği iki rol bulunmaktadır. Bu roller kullanıcı ve yöneticidir. Üyelik işlemi tamamlandıktan sonra kullanıcı giriş ekranına yönlendirilir. Bir rolün diğer rolün alanlarına erişim yetkisi yoktur. Navbar role göre dizayn edilmiştir.

Sisteme giriş yapan kullanıcılar kitap arayabilir alabilir iade edebilir. Kitap alma işlemi için iki koşul vardır. En fazla üç kitap alabilir ve teslim tarihi geçmiş kitabı iade etmende yeni kitap alamaz. Kitap iade işlemi için kullanıcının kitaba ait ISBN numarasını yüklenmesi gerekmektedir. Yüklemi işlemi tamamladıktan sonra ISBN numarası input alanına yüklenir ve kullanıcı kitabın ismini girip kitabı iade edebilir. Kullanıcı ISBN numarası veya kitap ismi ile arama yapabilir. Aranan anahtar kelimeye ait kitaplar listelenir.

Sisteme giriş yapan yöneticiler kitap ekleyebilir kullanıcı bilgilerini listeleyebilir ve sistem tarihini ileriye alabilir. Kitap ISBN numarası ile kitap ekleyebilir. Bunun için sisteme ISBN numarası içeren resim yüklemesi gerekmektedir. Sisteme ait zamanı girdiği gün sayısı ile ileriye alabilir. Kullanıcı listeleme sekmesinde kullanıcının üstündeki kitapları ve kullanıcı bilgilerini görüntüleyebilir.

**2. Temel Bilgiler**

Projenin gelişiminde;

Programla dili olarak “Python” kullanılmıştır.

Tümleşik geliştirme ortamı olarak “Visual Studio Code” kullanılmıştır.

Database olarak “MongoDB” kullanılmıştır.

Projenin bağımlıkları arasında “Pip”,”Virtulenv”,”Npm” bulunmaktadır.

Görüntü işleme için “OpenCV” ve “Tesseract” kullanılmıştır.

**3. Tasarım**

Proje aşağıdaki başlıklar altında gerçekleştirilmiştir.

**3.1 Yazılım Tasarımı**

Yazılım tasarımı aşağıdaki başlıklar altında geliştirilmiştir.

**3.1.1 App Routing**

**@app.route( / )** uygulama ayağa kaldırıldığında kullanıcıyı ilk karşılayan sayfayı yönlendilir. Kullanıcı girişi başarılı olursa kullanıcıyı anasayfa yönledirilir

**@app.route( /index )** kullanıcıya ait özelliklere erişebilmesini sağlar.

**@app.route( /signup )** kullanıcı kayıt sayfanına yönledirir.Oluşturulan formlar için WTForms kullanılmıştır. GET ve POST requestler alır.

**@app.route( /logout )** giriş yapan kullanıcıya ait session temizlenir.

**@app.route( /addbook )** yöneticiye ait alandır. GET ve POST metoda sahiptir. Gelen request ile ISBN numarasını alır ve db ye yazar.

**@app.route( /uploadfile )** yönetici ve kullanıcının için ortak olarak kullanılmaktadır. addbook veya deliverbook sayfalarından gelen yönlendirmeye göre yüklenen resme ait ISBN numarasını döner.

**@app.route( /listuser )** yöneticiye aittir.Kullanıcı datalarını listeler.

**@app.route( /changelocaldate )** yöneticiye saati ileri alacağı sayfaya yönledirir.

**@app.route( /searchbook )** kullanıcının kitap ismi veya ISBN numarası ile arama fulltext search yapmasını sağlar. GET ve POST metoda sahiptir

**@app.route( /rentbook )** kullanıcının kitap kiralamasını sağlayan sayfaya yönlendirir. Database üzerinden kiralanabilecek kitaplar listelenir. GET ve POST metoda sahiptir

**@app.route( /deliverbooks )** kullanıcının ISBN numrasını içeren resim ile kitap iade edebilmesini sağlar.

**3.1.2 Fonksiyonlar**

Uygulamada çalışan bölümler aşağıda listelenmiştir

**3.1.2.1 auth\_helpers.py**

**login\_required(role=None)** rollerin erişim alanlarını session daki data ile kontrol eder. Flash mesajlar ile kullanıcıyı bilgilendirir

**login\_required()** kullanıcının giriş yaptıktan sonra singup ve login sayfasını erişimini engellemek için geliştirilmiştir.

**role\_required()** kullanıcılar için ortak alan sayfalar için kullanılmak için geliştirilmiştir.

**session\_openned(\_id,username,role)** kullanıcı girişi başarılı olursa bilgileri sessiona yazar.

**session\_closed()** kullanıcı çıkış yaptığında sessionı temizler.

**get\_sha256\_password(password)** kullanıcının şifresini sha256 çevirir.

**3.1.2.2 config\_helpers.py**

**get\_value\_from\_name(name)** develop.conf dosyasından dataları almak için kullanılır.

**3.1.2.3 date\_helpers.py**

**get\_date\_time()** sistem saatini verir.

**set\_next\_day(day)** yöneticinin sistem saatini ileriye alabilmesini sağlar

**3.1.2.4 db\_crud\_helpers.py**

CRUD işlemleri bulunmaktadır.

**book\_transfer(remov\_col,insert\_col,query,data)**

Tablolar arasında kitap transferi gerekli fonksiyonlar için çağırır.

**insert\_rent\_col(removcol,insert\_col,query,user)**

Kiralanan kiralanan kitabı ve kiralayan kişinin bilgilerini tabloya ekler

**deliver\_book(remove\_col,insert\_col,query)**

Kitap iade işlemini gerçekleştirken kitabın kiralayan kişinin bilgilerini temizler.

**3.1.2.5 image\_proc\_helpers.py**

**imread\_image(self)** belirtilen dosyadan resim yükler

**imwrite\_image(self)** belirtilen resmi depolama aygıtına kayıt eder.

**proc\_image(self)** resim üzerinde görüntü işleme yapar

**read\_image(self)** tesseract ile resmin üzerindeki texti okur

**get\_isbn(self)** tessracttan gelen texti parse ederek ISBN numarasını döner.

**get\_result(self)** resmin işlemesi ve okunması için gerekli fonksiyonların çağrıldığı fonksiyondur.

**3.1.2.6 json\_helpers.py**

**json\_encoder(self,o)** json encode işlemini yapar

**3.1.2.7 mongo\_helpers.py**

**\_\_init\_\_(self)** constructor metodudur. MongoClient için singleton işlemi yapılır.

**getInstance()** pymongo.MongoClient objesini döner

**3.2 Modeller**

Uygulamada kullanılan modeller aşağıda listelenmiştir

**3.2 form.py**

Formlar için WTForms kullanılmıştır. Formalar için validation işlemleri yapılmıştır. Gerekli cssler formlar içinde render\_kw özelliği ile verilmiştir. Benzer yapılar olduğu için tüm modelleri içermemektedir.

**3.2.1 RegistrationForm**

**username = StringField()** kullanıcının username’ni almak için input alanı oluşturur.

**password = PasswordField()** passworda uygun input alanı oluşturur.

**role = SelectField()** choices özelliğine verilen değişkene rolleri kullanarak kayıt olan kullanıcının rolü seçeceği alanı oluştur.

**button = SubmitField()** formun submit tipi ile göndermesini sağlar

**3.2.2 BookForm**

**ISBN = StringField()** ISBN numarasını için input alanı oluştur

**book\_name = StringField()** kitabın adını almak için input alanını oluşturur

**button = SubmitField()** formun submit tipi ile göndermesini sağlar

**3.3 Templates**

Templateler oluştururken tekrarını azaltmak için jinja template kullanıldı.HTML dosyasılarında kullanılan ortak alanlar için base.html authbase.html gibi yapılar oluşturuldu. Bu sayede tekrar eden alanlar azaltıldı. İncludes klasörü altında footer header gibi html dosyaları toplandı bunlar ilgili sayfalara include edildi. Tekrar eden taglari azaltmak için for döngüsü kullanılmıştır. Formlar WTF forms ile oluşturulmuştur.

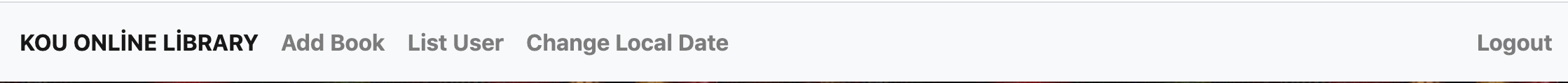
**4. Static**

Statik dosyalar static klasörünün altında toplanmıştır. İmage ve css ler burada toplanmıştır.

**5. Arayüz**

Kitap uygulamasının arayüz bileşenleri aşağıda listelenmiştir.

Uygulamanın header’ı kullanıcı tipine göre değişmektedir.



**(Şekil 1: Yönetici Navbar)**

Yöneticiye ait bileşenler;

**AddBook** ISBN içeren resim ile yönetici kitap ekleyebilir.

A screenshot of a cell phone

Description automatically generated

**(Şekil 3: Yönetici Navbar)**

**ListUser** yönetici sisteme kayıtlı olan kullanıcıları ait bilgileri listeler.

A screenshot of a cell phone

Description automatically generated

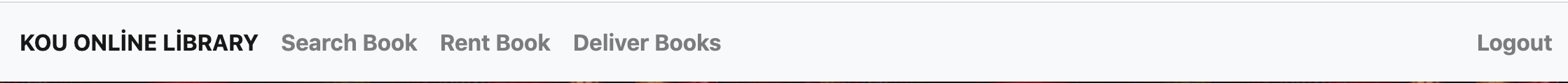
**(Şekil 3: ListUser)**

**ChangeLocalDate** yöneticinin sistemin saatini ileriye alabilmesini sağlayacak arayüzü sağlar.

A screenshot of a cell phone

Description automatically generated

**(Şekil 4: ChangeLocalDate)**



**(Şekil 4: Kullanıcı Navbar)**

Kullanıcıya ait bileşenler;

**SearchBook** ISBN numarası ve kitap adı ile kullanıcının aradığı kitabı listeler.

A picture containing book, shelf, sitting, bookshelf

Description automatically generated

**(Şekil 5: SearchBook)**

**RentBook** kullanıcının koşulları sağladığı takdirde kitap alabilmesi sağlayacak arayüzü kullanıcıya sunar.

A screenshot of a cell phone

Description automatically generated

**(Şekil 5: RentBook)**

**DeliverBook** ISBN numarası ile kullanıcın kitap iade etmesini sağlar.

A screenshot of a cell phone

Description automatically generated

**(Şekil 5: DeliverBook)**

Ortak Alan;

**Login** Kullanıcının sisteme giriş yapmasını sağlar.

A screenshot of a cell phone

Description automatically generated

**(Şekil 6: Login)**

**6.Akış Şeması**

**A close up of a piece of paper

Description automatically generated**

**(Şekil 7: Akış Şeması)**

**7. Kazanımlar**

**7.1 Web uygulaması Tecrübesi**

Python ve teknolojilerini hakkında tecrübemizi arttırdığımız bir projeydi. Projedeki öncelikli amacımız yeni teknolojiler kullanıp bilgilerimizi artırmaktı. Bunun için MongoDB,WTForms gibi teknolojileri kullandık.

**7.2 MongoDB Tecrübesi**

Daha önceki projelerde MySQL kullandığımız için NoSQL bir teknoloji kullandık. Bu noktada collectionların birleştirilmesi gibi konularda araştırma yapamamız gerekti. Aynı zamanda ne zaman sql ne zaman nosql bir database kullanmamız gerektiği konusunda tecrübe edindik.

**7.3 Git ve Github Tecrübesi**

Okulların kapalı olmasından dolayı farklı şehirlerdeydik. Bu konuda git konusundaki tecrübemizi artırdık.

**7.4 Readme Tecrübesi**

Projemizi açık kaynak kodlu olarak geliştirdiğimiz için isteyen herkesin kendi bilgisayarında kurup kullanama bilmesi için gerekli adımları anlatan bir readme hazırladık.

**7.5 VirtualEnv Tecrübesi**

Projedeki gerekli paketleri sistemden bağımsız bir şekilde kullana bilmek için virtualEnv kullandık. Bu sayede geliştirme ortamımızdaoluşabilecek problemleri ortadan kaldırmış olduk.

**8. Kaynakça**

<https://flask.palletsprojects.com/en/1.1.x/quickstart/>

<https://flask.palletsprojects.com/en/1.1.x/patterns/fileuploads/>

<https://www.w3schools.com/python/python_mongodb_getstarted.asp>

<https://www.w3schools.com/python/default.asp>

<https://docs.opencv.org/master/d6/d00/tutorial_py_root.html>

<https://pythonprogramming.net/loading-images-python-opencv-tutorial/>

<https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/>

<https://stackoverflow.com/questions/38133529/aggregate-query-in-mongo-works-does-not-in-pymongo>

<https://mongodb.tecladocode.com/mongodb_with_python/database.py.html#code-for-database-py>

<https://nanonets.com/blog/ocr-with-tesseract/>

<https://stackoverflow.com/questions/16586180/typeerror-objectid-is-not-json-serializable>

<https://www.tutorialspoint.com/python_design_patterns/python_design_patterns_singleton.htm>

<https://github.com/incebellipipo/apitizer>

<https://getbootstrap.com/docs/4.0/components/forms/>

<https://flask.palletsprojects.com/en/1.1.x/templating/>

<https://pythonhosted.org/Flask-Session/>

<https://github.com/mertbilgic/PythonFlaskCnn>

<https://flask.palletsprojects.com/en/1.1.x/patterns/wtforms/>

<https://medium.com/gokhanyavas/virtualenv-nedir-nas%C4%B1l-kurulur-205f1fd57f05>

<https://stackoverflow.com/questions/2885190/using-configparser-to-read-a-file-without-section-name>