**Kütüphane Uygulaması**

Sinem Kulaç

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

sinemkulac1998[@gmail.com](mailto:yigitkemalagac@gmail.com)

**1.Problem Tanımı**

Proje Web üzerinde çalışan kütüphane uygulamasıdır. Sistemde kullanıcı ve yönetici olmak üzere iki kullanıcı vardır. Yöneticinin kitap ekleme, zaman atlama ve kullanıcı listeleme yetkileri vardır. Kitap eklerken sisteme kitap resmi yüklenir, görüntü işleme ile kitabın ISBN numarası bulunur, yönetici sisteme kitabın adını girer böylelikle sisteme kitap adı ve ISBN numarası kaydedilir. Zaman atlama işleminde güncel zaman 20 gün sonrasına çekillir. Kullanıcı listelendiğinde kullanıcılar, aldığı kitaplarla görüntülenir.

Sisteme kayıtlı kullanıcıllar kitap arama, kitap alma ve kitap verme özelliklerini kullanabilirler. Arayüzde bulunan kitap arama özelliği sayesinde kitaba ve sistemdeki durumuna ulaşabilirler. Kitap alma özelliğini kullanarak kitap alabilirler. Kitap geri verirken sisteme kitabın resmi yüklenir, görüntü işleme ile kitabın ISBN numarası çekilir ve kullanıcı kitabı sisteme geri verir.

**2.Yapılan Araştımalar**

Projede web için back-end için Python, web için Flask kullanılmıştır. Flask kullanımı için araştırma yapılmıştır.

Veri tabanı olarak MySQL kullanılmıştır. Veri tabanı tasarımında arştırma yapılmıştır. Görüntü işleme için OpenCV ve Tesseract kütüphanelerinden yararlanılmıştır.

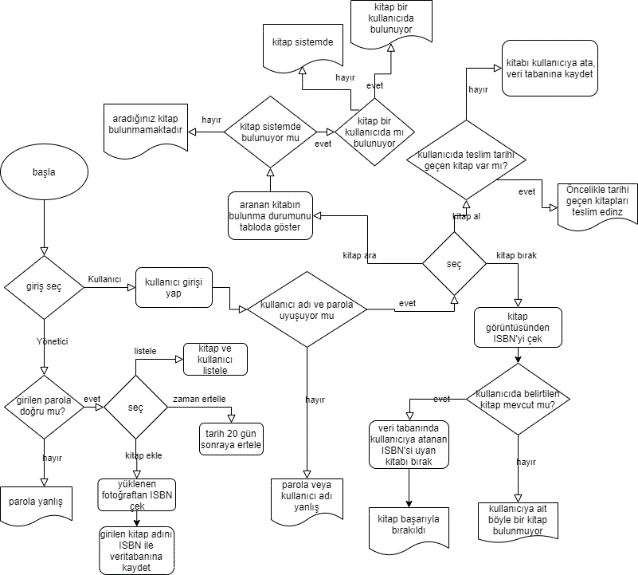
**2.1 Karşılaşılan Zorluklar ve Çözüm Yolları**

Tesseract kütüphanesi bazı resimler de kötü sonuçlar verir ve resimden ISBN numaraasını çıkaramayız bu durumun önüne geçmek için OpenCV ile resim üzerinde görüntü işleme teknikleri uygulanır ve ardından ortaya çıkan resim Tesseracta gönderilir.

**3.Tasarım**

Proje temelinde Yönetici ve kullanıcıya kolaylık sağlamak üzere web arayüzü, kullanıcı ve kitap bilgilerini saklamak için MySQL veritabanı, back-end kısmında Python programlama dili kullanılmıştır.

**3.1Akış Şeması**



**3.2Yazılım Mimarisi**

Porje web kısmı , veri tabanı ve python ile yazılmış back-end kısmından oluşur.

Web kısmında kullanıcıyla etkileşim için html ve css kullanılmıştır. Web kısmı ile back-end bağlantısında Flask mikro web çerçevesi kullanılmıştır.

Back-end kısmında nesneye yönelik Python programlama dili kullanılmıştır.

Program Flask ile kısımlara ayrılmıştır.

**3.2.1 Giriş Ekranı**

**@app.route(/Index) s**iteye giren kullanıcıyı karşılayan kısımdır, **index.html** yönlendirir. Kullanıcı için yönetici veya kullanıcı girşi sağlar

**3.2.2. Yönetici kısmı**

**@app.route(/admin)** yönetici girişi yapılınca yönlendirilen kısımdır. Yönetici işlemleri için geliştirilmiştir. Yöneticinin işlemine bağlı olarak kitap eklemeye veya kullanıcı listelemeye yönlendirebilir.

Kitap yüklemek isterse arayüzü kullanarak kitabın resmini sisteme yükler, bu işlem **@app.route(upload\_image)** sayesinde UPLOAD\_FOLDER dizinine resim

yüklenmesiyle gerçekleşir. Yönetici arayüzde bulunan ilgili bölüme kitap adı girer ve **@app.route(/admin)** altındaki kodlar çalışır, **resim\_oku** fonksiyonu resmi okur ve resimden ISBN çıkartıp String olarak geri döndürür ardından veritabanına kitabın adıyla beraber ISBN numarası kaydolur.

**@app.route(admin/liste)**  bu kısım arayüzde bulunan “Kitap ve Kullanıcı Listele” butonuna tıklandığında kullanıcıları ve aldığı kitapları veritabanından çekip arayüze gönderir ve kullanıcıyı **liste.html** e yönlendirir.

**@app.route(admin/ertele)** yönetici arayüzünde bulunan “Zaman Ertele butonuna tıklandığında bu kısımda bulunan kodlar sayesinde anlık zaman 20 gün sonrasına alınır.

**@app.route(/upload\_imag)**

**3.2.3 Kullanıcı kısmı**

**@app.route(/user)** kullanıcı için giriş ekranında kontrol yapıldığı kısımdır. Kullanıcı adı ve şifre girilir bir veri tabanında kontrol yapılır eğer doğruysa **@app.route(/usr)** ksımına yönlendirilir.

**@app.route(/usr)** kullanıcın giriş yaptıktan sonra yönlendirildiği kısımdır, kullanıcıya işlemlerini gerçekleşitirmesi için **usr1.html** e yönlendirir.

**@app.route(/usrkitapara) usr1.html** de bulunan “kitap ara” butonuna tıklandığında buraya yönlendirilr. Kullanıcının girdiği kitap adı veya ISBN numarası alınır, veritabanınnda sorgu yapılır kitap varsa sistemdeki durumuyla beraber kullanıcıya gösterilir kitap yoksa “aradığınız kitap bulunmamaktadır” mesajı döndürülür.

**@app.route(/usrkitapal) usr1.html** de bulunan kitap al butonuna tıklandığında bu kısma yönlendirilir. Girilen ISBN numarası veya kitap adı bilgisi alınır. Kitabın sistemdeki varlığı kontrol edilir kitap yoksa “aradığınız kitap bulunmamaktadır” mesajı gönderilir. Kitap bulunuyorsa veritabanındaki durumuna bakılır kitap kütüphanede değilse “bir kullanıcıda” mesajı döndürülür. Kitap sistemde ise veritabanında kullanıcı ile ilgili kontroller yapılır. Kullanıcının teslim tarihi geçmiş kitabı yoksa veya aldığı kitapların sayısı 3 taneden az ise kitap kullanıcıya verilir. Bunun için veritabanında kitabın sistemdeki durum bilgisi düzenlenir. Ardından kayıt tablosuna; kitabın ISBN numarası, kullanıcı adı ve kitabın alındığı güncel tarih bilgisi girilir. Son olarak kullanıcı tablosunda kullanıcın sahip olduğu kitap sayısı bilgisi güncellenir ve kitap kullanıcıya verilmiş olur.

**@app.route(/usrkitapbırak)** **usr1.html**de bulunan kitap bırak butonuna basıldığında bu kısma yönlendirme yapılır. Kitap bırakmak için **usr1.html**de bulunan dosya seç butonuna basılıp sisteme kitabın resmi yüklenir. Kitabın resmi **resim\_oku** fonksiyonuna gönderilir. Fonksiyonda resimden ISBN çıkartılır ve string olarak geri döndürülür. Sistemde kullanıcının kitabı geri vermesi için veritabanı işlemleri başlatılır. Veritabanında kitabın kullanıcıda olup olmadığı kontrol edilir. Kitap kullanıcıdaysa; kitabın sistemdeki durumu gücellenir, kitap\_kayit tablosundan kayıt silinir ve kullanıcın aldığı kitap sayısı bilgisi güncellenir. Kitap kullanıcıda değilse **@app.root(/usr)** kısmına yönledirme yapılır.

**3.2.3 Fonksiyonlar**

**resim\_oku** resmi okunacak kitabın adını string olarak alır. Aldığını string değerindeki resmi UPLOAD\_FOLDER dizininde arar, okur ve numpy dizisi olarak tutar. Projede resimden yazı okumak için Tesseract kullanılmıştır, Tesseract her resmi düzgün okuyamadığı için resim Tesseracta gönderilmeden önce resme Görüntü İşleme uygulanır. Sırasıyla gri ton a çevrilir, treshold uygulanır, gürültü azaltma metodu uygulanır ve son olarak harf kalınlaştırma yöntemi uygulanır. Sonuçta ortaya çıkan resim Tesseracta gönderilir. Tesseract bir string değeri döndürür. Alınan stringte regex ile ISBN bilgisi alınır ve fonksiyonda geri döndürülür.

**log\_required –** Seçilen html sayfasına girmek için giriş yapılmış olmasını gerektirir.

**allowed\_file –** Seçilen dosyatürünün istenen dosya türlerinden biri olmasını sağlar.

**3.2.4 Sınıflar**

**Zaman** zaman bilgisi tutulur

**Record** kullanıcı adı ve ihtiyaç duyulduğunda seçilen kitap adı tutulur

**Addbook-** StringField’dan aldığımız kitap ismi ve isbn değişkenlere atılır.

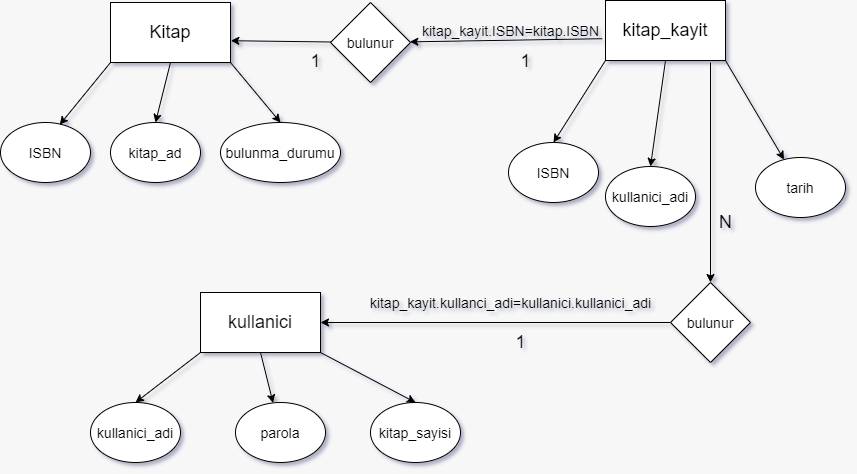
**Login –** StringField ve PasswordField’dan aldığımız kullanıcı adı ve parolayı saklar.

**3.2.5 Veritabanı tabloları**

**Kitap** kitap adı, ISBN numarası ve sistemdeki bulunma durumu tutulur; kitap sistemdeyse 1, kullanıcıdaysa 0 değerini alır.

**Kullanici\_ad** kullanıcı adı, parolası ve kullanıcının aldığı kitap adedi bilgisi tutulur.

**Kitap\_kayit** kullanıcı adı, aldığı kitapların ISBN numaraları ve aldığı tariler tutulur. Tarih bilgisi Date tipinde tutulur.

**3.3 Veritabanı diyagramı**

**4.Genel Yapı**

Proje web kısmı için html, css ve Flasktan; Back-end kısmında Python programlama diliyle yazılmış olup, Veri tabanı olarak MySQL kullanılmıştır.

Kullanıcıyla rahat bir kullanım arayüzü sağlamak için html ve css kullanılmıştır. Flask ile web yapısı kurulmuştur. Projede kullanıcı girişi ya da yönetici girişi yapılır. Yönetici girişi yapıldığında sisteme kitap yükleme kullanıcı bilgisi görüntüleme ve zaman atlama yetkileri bulunur. Sisteme kitap yüklenirken kitap resmi yüklenir ve kitap adı girilir Tesseract ile resimden ISBN çıkarıldığından kitap veritabanına kayıt edilir. Dilendiği zaman kullanıcılar aldıkları kitaplarla beraber görüntülenir veya sistemdeki güncel zaman bilgisi 20 gün sonrasına alınır.

Veritabanı kullanıcı bilgilerini, kitap bilgilerini ve kullanıcıların aldığı kitapların kaydını tutmak üzere 3 tablodan oluşur.

**5.Referanslar**

<https://www.youtube.com/watch?v=pNqtDpcPJm0&list=PLIHume2cwmHfDUQ2t6f-Q7VCBvNSEAMAS>

<https://tesseract-ocr.github.io/tessdoc/ImproveQuality>

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

<https://www.sinanerdinc.com/python-re-modulu>

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

<https://docs.opencv.org/4.2.0/d5/d69/tutorial_py_non_local_means.html>

<https://www.geeksforgeeks.org/python-denoising-of-colored-images-using-opencv/>

<https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html>

**Kazanımlar**

Python ile Flask kullanarak web projesi geliştirme konusunda tecrübe edinildi.

Veritabanı için araştırmalar yapıldı veritabanı tasarımı konusunda bilgi edinildi.

Tesseract kullanımı ve Python ile OpenCV kullanarak görüntü işleme teknikleri hakkında tecrübe kazanıldı.