

# Masaüstü Kitap Okuma ve Öneri Uygulaması

Ata GÜLALAN - Oğuzhan TÜRKER

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

xavaneo@gmail.com oguzturker8@gmail.com

## Özet

*Masaüstü kitap okuma ve öneri uygulaması, kullanıcıların uygulamaya kayıt olup giriş yapabildiği, kitap seçip, seçtiği kitaplara göre öneri kitaplar alabildiği bir masaüstü uygulamadır.*

## 1. Giriş

Masaüstü kitap okuma ve öneri uygulamasına giriş yapabilmek için öncelikle uygulamada bir hesabınızın olması gerekmektedir. Eğer kayıtlı değilseniz ana ekrandan Kayıt Ol seçeneğine tıklayarak kayıt olabilirsiniz. Uygulamaya zaten kayıtlı iseniz giriş bilgilerinizi yazarak giriş yapabilirsiniz.

Eğer uygulamaya kayıt olduktan sonra herhangi bir zamanda on kitap oylamadıysanız uygulamanın size doğru bir şekilde kitap önerisinde bulunması için en az on kitaba oy vermeniz gerekmektedir.

On veya daha fazla oy kullandıysanız uygulamanın ana arayüzüne ulaşabilirsiniz. Burada seçtiğiniz kitaplara göre uygulama size kitap önerilerinde bulunur.

Ayrıca önerilerin yanında en çok oy almış, en yüksek oy almış ve en yeni eklemiş kitapları da bulabilirsiniz.

Beğendiğiniz kitabın üzerine tıklayıp bilgilerini görebilir, isterseniz kitabı okuyabilirsiniz.

## 2. Temel Bilgiler

Proje gelişiminde;

Programlama dili olarak “C#”, tümleşik geliştirme ortamı olarak “Visual Studio 2017” ve “Visual Studio 2015” kullanılmıştır.

Veri tabanı yönetim sistemi için “MySQL” ve uygulama içeriği olarak “Windows Presentation Foundation (WPF)” kullanılmıştır.

## 3. Tasarım

Proje aşağıdaki başlıklar altında geliştirilmiştir.

### 3.1 Yazılım Tasarımı

Projenin yazılım aşaması bu başlık altında bulunan konular tarafınca geliştirilmiştir.

#### 3.1.1 Uygulama yapısı

Uygulamada kullanılan bileşenler aşağıda maddeler halinde listelenmiştir.

**Register :** Kullanıcıların uygulamaya kayıt olabildiği sayfa.

Kayıt olma işleminde kullanıcılardan kullanıcı adı, konum, yaş ve şifre bilgileri kullanıcıdan alınıp veritabanına yazılır. Bu esnada kullanıcının girdiği bilgilerin hatalı olup olmadığını veya var olan kayıt bilgilerini kullanıp kullanmadığını kontrol eder.

**Login :** Kaydınızın bulunduğu takdirde uygulamaya giriş yapabilmeniz için kullanılan giriş sayfası.

Kullanıcıdan giriş yapabilmesi için kullanıcı adı ve şifre bilgilerini alır. Veritabanından kontrol eder. Kullanıcı adı ve şifre doğru eşleşirse daha önceden kaç kitap oyladığını veritabanından kontrol eder. Ondan ve daha fazla kitap oylamış ise Read sayfasına, daha az ise **Books** sayfasına yönlendirir.

**MainWindow :** Giriş ve kaydın bulunduğu, her şeyin başlangıcı olan ana sayfa.

**Books :** Ondan daha az kitap oynunuz bulunduğu Read sayfasına erişebilmeniz için kitapları oyladığınız ön giriş sayfası.

Oy vermek için tüm kitaplar ekrana sayfanır. Ekrandaki kitaplar için **BookCover** sınıfı kullanılmıştır. Kitapların sayfanması için **Pagination** sınıfı kullanılmıştır. Kullanıcının kaç oy verdiğini dinamik olarak veritabanından gösteren ve ondan daha fazla ise **Read** sayfasına yönlendiren iler butonu, kitap oylarken arama yapabilmek için arama butonu ve son olarak giriş

yapılan kullanıcı adını ekranda gösteren hoşgeldin etiketi bulunur.

**Read:** On kitaptan daha fazla kitabı oyladığınızda karşınıza çıkan, uygulamanın ana ekranı.

Tasarımı Books ile benzer ve aynı sınıflar kullanılıyor. Books'tan farklı olarak "En Popüler 10 Kitap", "En İyi 10 Kitap", "En Yeni 10 Kitap" ve "Sizin İçin Önerilenler" kısmı bulunuyor.

Birinci bölüm; Kullanıcının oyladığı kitaplara göre **kosinüs benzerliği** yardımıyla on kitap öneren öneri bölümü.

**İkinci bölüm;** En çok oy alan kitapların bulunup gösterildiği en popüler kitaplar bölümü.

**Üçüncü bölüm;** En yüksek oyları alan kitapların bulunup gösterildiği en iyi kitaplar bölümü.

**Üçüncü bölüm;** Veritabanına en son eklenen kitapların gösterildiği en yeni kitaplar bölümü.

Boyut tasarrufu için ScrollView kullanılmıştır.

Kitaplar BookCover sınıfı ile gösterilmiştir.

Tüm kitaplar arasında gezinim için Pagination sınıfı kullanılmıştır.

**PDFRead :** PDF dosyalarını okuyabilmek için kullanılan ara sınıf.

PDF'leri kaynakta da belirtilen\* yöntemle okunduğu sınıf.

**Pagination :** Kitapları ekrana sayfalamak için kullanılan ara sınıf.

Her sayfada 7x3'lük şekilde gösterilen kitapların (**BookCover**) sayfalanmasını sağlayan sınıftır. Kitapların hareketleri bulunan 25 buton ile sağlanır.

**BookInner :** Sayfalanmış kitapları okumak için kullanılan, kitap bilgileri, resmi ve pdf dosyasını okumak için bir buton bulunan kitap içerik sayfası.

Kitap resmi sol üstte bulunan gridde gösterilir. Kitap adı üst orta kısımda gösterilir.

Alt gridde ise kitapları oylamak için bulunan 10 buton vardır. Hepsi kitaba üzerinde bulunan oy değerini verir. Ayrıca sağ tarafında kitapların bilgilerinin gösterildiği özellikler kısmı ve kitapları okumak için **PDFRead** sınıfının çalıştırıldığı kitap okuma butonu bulunmaktadır.

**BookCover :** Sayfalararak ekranda gösterdiğiniz kitapların sınıfı.

Books ve Read sayfalarında ekrana sayfalanarak gösterilen kitapların sınıfıdır. Arka planında kitabın resmi bulunur. Kitap özellikleri (adı, aldığı oy sayısı, ort. oy sayısı vs.) hover özelliği şeklinde üzerinde gösterilir. Ortalama oy gösteriminde star.png kullanılmıştır.

### 3.1.2 Fonksiyonlar

Uygulamada kullanılan fonksiyonlar sınıfların başlıkları altında listelenmiştir.

#### Register;

**hataVer :** Kayıt işleminde girilen bilgilerin kontrolünde kullanıcıya hatayı göstermek için kullanılır.

**hataSil :** Kayıt işleminde giriş bilgileri doğru ise hata metinlerini yok saymak için kullanılır.

**kayıtIslemi :** Kullanıcıların girdiği kayıt bilgilerini veritabanında işleyen fonksiyondur. Üstteki fonksiyonlar bu fonksiyon içinde kullanılır.

**geriGel :** Kayıt penceresini kapamak için kullanılan buton fonksiyonudur.

Window\_Closing : Uygulamadan çıkar.

#### Login;

**hataVer :** Giriş işleminde girilen bilgilerin kontrolünde kullanıcıya hatayı göstermek için kullanılır.

**login :** Kullanıcıların girdiği giriş bilgilerini veritabanında işler. Kullanıcı adı ve şifre birbirleri ile uyuşursa kullanıcının kaç oy verdiğini bulur. Oy sayısına göre **Books** veya **Read** sayfasını açar.

**girisIslemi :** login fonksiyonunu kontrol eden buton fonksiyonu.

**geriGel2 :** Giriş sayfasını kapatan fonksiyon.

**Window\_Closing :** Reelmain yapıcısını çalıştırır.

#### Books:

**paginationChanged :** BackgroundWorker elemanını kontrol eder.

OpenConnection : Sql bağlantısı açar

**getPageCount :** Sayfalanacak kitapların sayfa sayısını bulur.

**getBooks :** Kitap bilgilerini sql'den çeker.:

**disableClick :** İşlemler arası hidden olan yükleniyor border'ını görünür kılar.

**enableClick :** İşlemler arası hidden olan yükleniyor border'ını görünmez kılar.

**araButonu\_Click :** Girilen isme göre kitap araması yapan buton fonksiyonu.

oyArttir : Kullanıcının **Read** sayfasına ilerleyebilmesi için gerekli oy sayısını dinamik olarak hesaplar.

**book\_MouseDown :** BookInner açmak için kullanılan fonksiyon. BookCover'daki verileri BookInner'ın yapıcısına yollayarak yeni bir sayfa açar.

**ileriButonu\_Click :** Yeterli oy sayısı var ise **Read** sayfasına yönlendiren buton fonksiyonu.

**Window\_Closing :** Pencere kapanırsa tüm uygulama kapanır.

**kitapAraInput\_KeyDown :** Arama parametresini koyar ve Pagination'u yeniler.

**Read;**

**OpenConnection:** Veritabanı bağlantısını (açık değilse) açar.

**getPageCount:** Arama ve tüm kitaplar içerisinde kaç adet kitap var ise 21'e bölerek kaç sayfaya böleceğini hesaplar.

**getPopi:** Popüler kitapları veritabanından çeker, işler ve listeler.

**getIyi:** İyi kitapları veritabanından çeker, işler ve listeler.

**getYeni:** Yeni kitapları veritabanından çeker, işler ve listeler.

**getBooks:** (Arama parametresine göre değişebilir) Tüm kitapları veritabanından çeker, işler ve listeler.

**disableClick:** Bir arama yapılırken yükleniyor ekranı gösterir ve tüm formu tıklanamaz hale getirir.

**enableClick:** disableClick fonksiyonundaki yapılan değişiklikleri geri çevirir.

**araButonu\_Click:** Arama parametresini ekler ve tüm kitapları çeker.

**book\_MouseDown:** BookInner açmak için kullanılan fonksiyon. BookCover'daki verileri BookInner'ın yapısına yollayarak yeni bir sayfa açar.

**Window\_Closing:** Pencere kapanırsa tüm uygulama kapanır.

**AYRAC\_MouseDown:** Anasayfaya gider ve arama parametresini temizler.

**kitapAraInput\_KeyDown:** Enter'a basılıp basılmadığını anlayarak arama işlemindeki kullanıcı deneyimini artırır.

**cosine\_similarity:** Kosinüs benzerliği algoritması ile iki vektörün benzerliğini bulan fonksiyon. (Mevcut duruma göre işlem kolaylığı adına basitleştirmeler yapılmış ve fonksiyonda açıklanmıştır.)

**find\_my\_books:** Kullanıcının okuduğu kitapları bulan fonksiyon.

**find\_matches:** Kullanıcının okuduğu kitapları okuyan diğer kullanıcıları bulan fonksiyon.

**find\_matches\_value:** Parametreyle gönderilen kullanıcı adının, uygulamayı kullanan kullanıcı adıyla kaç kitabının eşleştiği hesaplanır.

**find\_rates:** oneri fonksiyonundan dönen değerlerin ilk 10 tanesini döndüren fonksiyon.

**oneri:** Eşleşen kullanıcı idleri, her kullanıcının okuduğu diğer kitaplar, o kitaplara verdiği oylar ve her kullanıcının **kosinüs benzerliği** ile bulunan benzerlik oranları burada toplanır. Kullanıcılar sahip oldukları benzerlik oranlarıyla diğer kitaplara verdikleri oyları

çarpılır. Daha sonra diğer kullanıcıların birbirleri arasında aynı kitaplara verdiği oylar toplanıp ortalaması alınır. Kitapların sahip oldukları son oylar büyüktür küçüğe sıralanır.

**PDFRead;** Okuma işlemini yapıcısında (constructor) yapmaktadır.

**Pagination;**

**removeClasses :** Tüm tıklanabilirlikleri sıfırlar.

**removeContents :** Tüm içerikleri siler. Bunu yapmasının nedeni, bul fonksiyonu çağırıldığında noktaların nereye koyulacağını bilmektir.

**sonlar :** En sağda bulunan üç noktadan sonraki elemanları bulur.

**normalSiral :** Baştaki elemanları üç noktasız sıralar.

**onceNoktali :** Baştaki elemanları üç noktalı sıralar.

**bul :** Belirlenen algoritmaya göre butonların sırasını belirler ve gerekli fonksiyonları çağırır.

**UserControl\_Loaded :** Pagination ilk yüklendiğinde yapılacak fonksiyonları içerir.

**kClick :** Herhangi bir butona tıklandığında çalışan fonksiyon. Butonun içerdiği veriye göre istenen fonksiyonları çalıştırır.

**MainWindow;** Projenin ana ekranıdır.

**sql :** SQL'i çalıştırır.

**dispatcherTimer\_Tick :** Uygulama açılırken daha önce giriş yapıp yapılmadığını bularak direkt olarak istenen pencereyi açar.

**OnClosed :** Kapatılırken çalışır.

**openLoginForm :** Giriş ekranını açar.

**openRegisterForm :** Kayıt ekranını açar.

**BookInner;**

**getImage :** Resmi çeker.

**disableButtons :** Butonları tıklanamaz hale getirir.

**isle :** BookInner açıldığında çalışan kod.

**oyla :** Kitabı oylamak için veritabanına istek atar.

**kitabıOku\_Click :** Kitabın okunabilmesi için PDFRead'i açar.

**BookCover;** Kitabın bilgilerini tutan, tıklanabilir interaktif buton.

**Publisher :** publisher değişkeni için get/set tanımlar.

**Date :** publisher değişkeni için get/set tanımlar.

**YearOfPublication :** yearofpublication değişkeni için get/set tanımlar.

**Isbn :** isbn değişkeni için get/set tanımlar.

**Author :** author değişkeni için get/set tanımlar.

**BookName :** bookname değişkeni için get/set tanımlar.

**Stars :** stars değişkeni için get/set tanımlar.

**VoteCount :** votecount değişkeni için get/set tanımlar.

**Image** :image değişkeni için get/set tanımlar.

**getImage** :URL'den byte array olarak veriyi çeker ve gösterir.

**cek** : Yıldızla göre ayarlamaları yapar, ismi girer, isbn'i kontrol eder.

**UserControl\_Loaded** : cek'i çağırır.

Tasarımı “Adobe Illustrator” programında çizdik ve XAML yardımı ile animasyonlaştırdık. Uygulamanın her bir kısmının birbiriyle uyumlu olmasına ayrıca dikkat ettik.

## 6. Kaynakça

Kosinüs benzerliği :  
<http://bilgisayarkavramlari.sadievrenseker.com/2012/11/08/kosinus-benzerligi-cosine-similarity-2/>

### 3.1.3 Karşılaşılan Sorunlar ve Çözüm Yöntemleri

## 4. Kazanımlar

### 4.1 C# Programlama Dili

Projeyi daha önce bilmediğimiz bir dil olan C# ile yaptık. Sonucunda C#'da programlama yeteneği kazandık.

### 4.2 Veriseti Yönetimi

Daha önce çalışmış olduğumuz tüm veritabanlarından daha büyük olan bu veritabanı bizde işlemlerin optimizasyonu hakkında bilgiler kazandırdı.

### 4.3 WPF Yapısı

Projemizi C#'ın bir yapısı olan WPF ile yaptık. Sonucunda bu yapı hakkında büyük tecrübelerimiz oldu. Ayrıca genel olarak görünüm panel yönetimini öğrenmiş olduk.

## 5. Tasarım Aşaması

