

**KOCAELİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**2018**

**YAZILIM LABORATUVARI I (BLM307)**

**PROJE 1 RAPORU**

**KİTAP OKUMA VE ÖNERİ SİSTEMİ**

MEHMET OĞUZ AKTAŞ

150201170

oguzaktass@gmail.com

EBUBAKİR ŞİT

150201245

[sitebubekir@gmail.com](mailto:sitebubekir@gmail.com)

**İçindekiler**

Problem Analizi ve Çözüm Algoritmasının Oluşturulması 2

Yazılım Mimarisi 6

Akış Şeması 7

Yazılımın Çalıştırılması ve Özellikler 7

Kaynakça 7

1 - Problem Analizi ve Çözüm Algoritmasının Oluşturulması

Bu projede, Book-Crossing datasetini kullanarak oluşturulan veritabanının, MySQL veritabanı yönetim sistemi ve Java kullanılarak collaborative filtering yöntemiyle öneri sisteminin yapılması, bir PDF okuma kütüphanesi kullanılarak program üzerinden sayfa sayfa PDF okuması yapılması, login ve istenen şartları sağlayan register bölümlerinin yazılması, login yaptıktan sonra açılan üyelik sayfasında istenen özelliklere göre veritabanı sorgularının çalıştırılarak sonuçların getirilmesi ve admin paneli tasarlanarak kayıt ekleme, silme, güncelleme, filtreleme gibi temel işlemlerin yapılması istenmektedir. Problemin çözümünü daha kolay yapabilmek için problemi şu şekilde alt parçalara ayırdık;

1 - İstenen özelliklerde Book-Crossing datasetindeki tabloları ve BX-Admin tablosunu içeren veritabanının oluşturulması

2 - MySQL ile JDBC kullanımı bilgilerinin tekrar edilmesi

3 - Login ve Register modülleri için GUI tasarımının yapılması ve SQL sorgularının denenmesi

4 - Veritabanındaki tablolar ile program arasındaki bağlantıyı sağlayabilmek için gerekli classların oluşturulması

5 - 10 kitap oylamadan üyeliğin tamamlanmaması şartının sağlanması

6 - Üye panelinin tasarlanması ve istenen sonuçlar için gerekli SQL sorgularının bulunup metotlar içinde kullanılması

7 - En iyi 10 kitap ve en çok oylanan 10 kitap sonuçları için gerekli SQL sorgularının yazılması ve İstatistikler bölümünde kullanılması

8 - Admin panelinin tasarlanması

9 - PDF’den kitap okuma işlemlerinin yapılması

10 - Collaborative filtering Kosinüs benzerliği (Cosine similarity) ile Öklid uzaklığı (Euclidian distance) yöntemlerinin araştırılması ve öneri sisteminin yazılması

11 - Veritabanı hatalarının düzeltilmesi

1 - İstenen özelliklerde Book-Crossing datasetindeki tabloları ve BX-Admin tablosunu içeren veritabanının oluşturulması

Veritabanını oluşturmak ve yönetmek için MySQL Workbench ve Navicat Premium Essentials for MySQL programlarını kullandık. İlk olarak MySQL Workbench üzerinden yeni schema oluşturup Book-Crossing datasetinde bulunan 3 tabloyu (BX-Users, BX-Books, BX-Book-Ratings) ve BX-Admin tablosunu gerekli sütunları ile birlikte oluşturduk. BX-Book-Ratings tablosunda bulunan User-ID ve ISBN sütunlarını BX-Users’taki User-ID ve BX-Books’taki ISBN sütunlarına foreign key olarak bağlamayı denerken “Error code 1215 cannot add foreign key constraint” hatasını aldık. Bu hatanın BX-Book-Ratings tablosunda oluşturduğumuz User-ID ve ISBN sütunlarını diğer tablolardaki sütunlarla aynı veri tipinde tanımlamadığımız için oluştuğunu çözdükten sonra User-ID’yi sorunsuz bir şekilde MySQL Workbench’te oluşturduğumuz EER diyagramı üzerinden bağladık. CSV dosyalarını kullanarak tablo içeriklerini veritabanına import etmek için MySQL Workbench’te “load data infile” komutunu kullandık fakat “secure\_file\_priv” hatası çıktığı için CSV dosyalarını /var/lib/mysql-files dizinine atıp tekrar denedik. Daha sonra BX-Book-Ratings tablosunda ISBN sütunu için “Error code 1452 Cannot add or update a child row foreign key constraint fail” hatasını aldık, bu hatanın bize verilen BX-Book-Ratings.csv dosyasındaki bazı ISBN numaralarının BX-Books.csv dosyasında bulunmadığı için ortaya çıktığını anladık. Sorun kullanmamız için verilen datasette olduğu için bu sorunu çözümsüz bırakmak zorunda kaldık. Import etme işleminde “Error code 2013 lost connection” hatasını da aldık, bunu MySQL Workbench ayarlarından MySQL session sürelerini 2 katına çıkararak çözdük. Son olarak “Error code 1064 character set” hatasını aldık, bu hatayı da veritabanı ve tabloları oluştururken default character seti latin1 olarak ayarlayarak çözdük.

2 - MySQL ile JDBC kullanımı bilgilerinin tekrar edilmesi

Geçen dönem yaptığımız MySQL Java JDBC Araç Yönetim Sistemi projesinde veritabanı işlemlerini nasıl yaptığımızı inceleyerek bu konudaki bilgilerimizi tazeledik. O projede MySQL Connector J 5.1.23’ü kullanmıştık, bu projede MySQL Connector Java 8.0.13’ü kullandık. <https://docs.oracle.com/javase/tutorial/jdbc/basics/connecting.html> adresindeki örneğe göre java.sql API’I içerisinde bulunan DriverManager classını, DriverManager.getConnection() metodunu ve yine java.sql API’i içerisindeki Connection interface’ini kullanarak veritabanı bağlantısının gerekli olduğu classlarda kullanmak için getConnection() metodunu yazdık. Statement ve PreparedStatement interface’lerini SQL ifadelerini çalıştırabilmek için, ResultSet nesnesini SELECT sorgularını JFrame classlarında JTable tablolarında gösterebilmek için kullandık.

3 - Login ve Register modülleri için GUI tasarımının yapılması ve SQL sorgularının denenmesi

Login ve Register formlarını NetBeans IDE’de Design kısmından gerekli Swing Componentlerini kullanarak oluşturduk. Login için Kullanıcı ID TextField alanına java.awt.event.KeyEvent classını kullanarak klavyeden sadece rakamların input olarak alınmasını sağladık. Register formunda da benzer şekilde Kullanıcı ID ve Yaş alanları için sadece Integer tipinde input girilmesini sağladık. Ayrıca 2 modülde de doldurulması gereken son TextField alanında klavyeden Enter’a basılınca yine KeyEvent classını kullanarak Giriş Yap ve Kayıt Ol butonlarının çalışmasını sağladık. Her 2 modülde de kullanılan bir ID numarası girildiğinde ve herhangi bir alan boş bırakıldığında JOptionPane mesajı görüntülenmesini sağladık. Register formunda showMinID() metodunu yazarak veritabanı sorgusuyla kullanılabilir durumda olan en küçük ID numarasını gösterip sürekli kullanılan ID’lerin seçilmemesi için yol gösterdik. Login formunda Giriş Yap butonuna basıldığında “*SELECT \* FROM `BX-Users` WHERE `User-ID` = ? AND Password = ?;*” veritabanı sorgusunu çalıştırarak (? kısımlarını girilen inputlardan alıp) eğer üye veritabanında bulunursa üye panelinin açılmasını, bulunamazsa “Kullanici ID veya sifreniz hatalidir.” şeklinde JOptionPane mesajı görüntülenmesini sağladık.

4 - Veritabanındaki tablolar ile program arasındaki bağlantıyı sağlayabilmek için gerekli classların oluşturulması

Veritabanı sorgularıyla alacağımız sonuçları istediğimiz veri tipinde gösterebilmek için BX-Users tablosunun özelliklerini içeren Users classını, BX-Books tablosunun özelliklerini içeren Books classını (bu class içinde Image-URL sütunlarından sadece Image-URL-M’yi kullanacağımız için diğer 2 image sütununu içermeyen ayrı constuctor da oluşturduk), BX-Book-Ratings tablosunun özelliklerini içeren BookRatings classını ve BX-Admin tablosunun özelliklerini içeren Admins classını Encapsulation’ı sağlayacak şekilde oluşturduk.

5 - 10 kitap oylamadan üyeliğin tamamlanmaması şartının sağlanması

10 kitap oylamadan üyeliğin tamamlanmaması şartını sağlamak için UserConfirmWindow isminde bir GUI classı oluşturduk. Kayıt Ol butonuna basıldığında inputların boş bırakılıp bırakılmadığı, girilen ID numarasının kullanılıp kullanılmadığı, girilen şifrelerin birbiriyle eşleşip eşleşmediği kontrolleri yapıldıktan sonra girilen inputları constructor parametresi olarak alan yeni bir UserConfirmWindow() nesnesi oluşturulmasını sağladık. UserConfirmWindow formunda veritabanındaki tüm kitaplar ArrayList<Books> tipindeki getBookList() ve tablodaGoster() metotları kullanılarak bir JTable aracılığıyla listelenir, çok fazla kitap bulunduğundan Ara butonunu yazarak ArrayList<Books> tipindeki searchBooks() metoduyla filtreleme yapılmasını sağladık (JTable üzerinden sayfalama algoritma karmaşıklığını pek azaltmadığı için onun yerine filtreleme kullandık). Kitabın üstüne tıklandığında ISBN, Kitap, Yazar, Yayın yılı, Yayınevi, Image alanlarının tabloda tıklanan kitabın değerleriyle dolmasını sağlamak için showItem1() ve showItem2() metotlarını yazdık. Image alanında, veritabanından BX-Books tablosundan Image-URL-M sütunundan aldığımız URL’yi kullanarak JLabel componentinin setIcon() metodu yardımıyla kitap resminin gösterilmesini sağladık. Seçili olan kitabı ComboBox’tan 1-10 arasında değer seçip Oy Ver butonuna basıldığında kayıtlı oy değerini ve kitabın ISBN değerini oluşturduğunuz isbnlist ve ratinglist ArrayList’lerinde sakladık. Eğer 10 kitap oylamadan Üyeliğimi Tamamla butonuna basılırsa üyeliğin tamamlanması oylanması gereken kaç kitap olduğu JOptinPane mesajıyla gösterilir, 10 kitap olduktan sonra butona basılırsa Register formunda kontrol edilerek alınan ve UserConfirm formuna parametre olarak gönderilen Kullanıcı ID, Şifre, Konum ve Yaş değerleri kullanılarak “*INSERT INTO `BX-Users`(`User-ID`, Location, Age, Password) VALUES(?, ?, ?, ?);”* sorgusuyla veritabanında yeni üyeliğin oluşmasını sağladık. Daha sonra isbnlist ve ratinglist ArrayList’lerinde tuttuğumuz değerleri “*INSERT INTO `BX-Book-Ratings`(`User-ID`, ISBN, `Book-Rating`) VALUES(?, ?, ?);”* SQL sorgusunu kullanarak veritabanına ekledik.

6 - Üye panelinin tasarlanması ve istenen sonuçlar için gerekli SQL sorgularının bulunup metotlar içinde kullanılması

Üye panelinde UserConfirm panelindeki gibi veritabanındaki tüm kitapların Ara butonuyla birlikte gösterildiği bir JTable ve tabloda kitabın üstüne tıklandığında değerlerin gösterilmesi için ISBN, Kitap, Yazar, Yayın yılı, Yayınevi, Image labellerini oluşturduk. Yeni kitaplar bölümünde kayıt eklendiğinde BX-Books tablosundaki Timestamp (Default değeri Current\_Timestamp olduğu için kayıt eklendiği andaki tarihi ve zamanı kaydeder.) tipindeki Insertion-Date sütununu kullanılarak veritabanına son eklenen 5 kitabın ayrı bir JTable’da gösterilmesini sağladık. Bunun için ArrayList<Books> tipindeki getNewBookList() metodunu yazarak “*SELECT \* FROM `BX-Books` ORDER BY `Insertion-Date` DESC LIMIT 5;*” SQL sorgusunu kullandık. Yine UserConfirm panelinde olduğu gibi Oy Ver butonuna basıldığında seçili kitaba oy verilmesini sağladık, verilen oyu veritabanına kaydetmeden önce kitaba daha önce oy verilip verilmediğini kontrol etmek için boolean tipinde checkRating() metodunu yazarak “*SELECT \* FROM `BX-Book-Ratings` WHERE `User-ID` = '" + id + "' AND ISBN = '" + isbn + "';*” sorgusuyla kontrol işlemini yaptık.

7 - En iyi 10 kitap ve en çok oylanan 10 kitap sonuçları için gerekli SQL sorgularının yazılması ve İstatistikler bölümünde kullanılması

Üye paneline İstatistikler butonu koyarak tıklandığında StatisticsWindow formunun açılmasını sağladık. Bu formda yine üye panelinde olduğu gibi 2 tane JTable bulunuyor. İlk tabloda ortalama oyu en yüksek olan 10 kitabı (en iyi kitaplar), diğer tabloda da en fazla kullanıcı tarafından oylanan 10 kitabı (popüler kitaplar) gösterdik. Ortalama oyu en yüksek olan 10 kitabı bulmak için ArrayList<String> tipinde getBestISBN() ve ArrayList<Books> tipinde getBestBookList() metotlarını yazdık. getBestISBN() metodunda “*SELECT ISBN FROM `BX-Book-Ratings` GROUP BY ISBN ORDER BY AVG(`Book-Rating`) DESC, COUNT(ISBN) DESC LIMIT 10;*” SQL sorgusunu kullanarak ortalama oyu en yüksek olan 10 kitabın ISBN değerlerini bulup getBestBookList() metodunda da bu ISBN değerlerini kullanarak kitapların diğer özelliklerinin veritabanından alınmasını sağladık. Aynı mantıkla yazdığımız getPopularISBN() ve getPopularBookList() metotlarıyla aynı işlemleri en fazla kullanıcı tarafından oylanan 10 kitabı bulmak için yaptık, getPopularISBN() metodunda “*SELECT ISBN FROM `BX-Book-Ratings` GROUP BY ISBN ORDER BY COUNT(ISBN) DESC LIMIT 10;*” SQL sorgusunu kullandık. Ayrıca ilk JTable'a Rating sütunu ekleyerek kitabın ortalama ratinginin gösterilmesini sağladık, ortalama ratingi bulmak için de ArrayList<Double> tipinde getRating() metodunu yazdık. Popüler kitapları gösterdiğimiz diğer JTable'a da Total Votes sütunu ekleyerek kitaba verilen toplam oy sayısının gösterilmesini sağladık, bunu bulmak için de ArrayList<Integer> tipinde getTotalVotes() metodunu yazdık.

8 - Admin panelinin tasarlanması

Admin Login formunu üye girişindekine benzer şekilde yazdıktan sonra admin paneli için oluşturduğumuz AdminWindow classında 4 farklı JFrame tanımladık. Ana JFrame’de Üyeleri Düzenle, Kitapları Düzenle, Yöneticileri Düzenle ve Çıkış Yap seçenekleri bulunuyor. Üyeleri Düzenle, Kitapları Düzenle ve Yöneticileri Düzenle panellerinde ortak olarak ilgili tablolardan veritabanı kayıtlarını çekip JTable’da gösterdik ve Insert, Update, Delete butonlarıyla veritabanına kayıt eklenebilmesini, güncellenebilmesini ve silinebilmesini sağladık. Insert butonu için kitap ekleniyorsa ISBN numarasının, üye ekleniyorsa ID numarasının kullanılıp kullanılmadığını kontrol ettikten sonra “*INSERT INTO `BX-Users`(`User-ID`, Location, Age, Password) VALUES(?, ?, ?, ?);”* şeklindeki örnek SQL sorgusuyla kayıt eklenmesini sağladık, kitap ve yönetici ekleme işlemleri için de benzer sorgular kullanıldı. Update işlemi için “*UPDATE `BX-Users` SET Location = ?, Age = ?, Password = ? WHERE `User-ID` = ?;”* örnek SQL sorgusunu kullandık. Delete işlemi için de “*DELETE FROM `BX-Users` WHERE `User-ID` = ?;*” örnek sorgusunu kullandık, aynı SQL sorgularını kitap ve yönetici tablolarındaki sütunlara göre değiştirerek kullandık.

9 - PDF’den kitap okuma işlemlerinin yapılması

PDF’den sayfa sayfa kitap okuma işlemi için önce Apache PDFBox kütüphanesini kullanmayı denedik fakat sadece IDE Output ekranından text çıktısı alabildiğimiz için pek kullanışlı gelmedi. Görsel çıktı oluşturmayla ilgili bir dökümantasyon bulamadık. Daha sonra StackOverflow’de denk geldiğimiz ICEpdf 6.3.1 kütüphanesini kullanmaya karar verdik. Icesoft.org’daki kullanım kılavuzundan yararlanarak kütüphaneyi projeye import ettikten sonra SwingController ve SwingViewBuilder araçları ile ayrı bir JFrame’de PDF’nin görüntülenmesini sağladık. planetebook.com adresinden indirdiğimiz telif hakkı bulunmayan 13 tane klasik edebiyat kitabını yazdığımız readBook() metodunda random seçerek kullandık. Üye panelinde bulunan Kitabı Oku butonuna basıldığında readBook() metodunun çalışmasını sağladık.

10 - Collaborative filtering Kosinüs benzerliği (Cosine similarity) ile Öklid uzaklığı (Euclidian distance) yöntemlerinin araştırılması ve öneri sisteminin yazılması

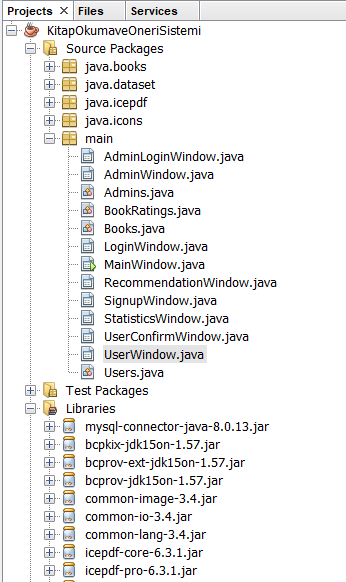
Collaborative filtering (işbirlikçi filtreleme) yöntemlerini araştırıp öneri sistemlerinde user-based (kullanıcı tabanlı) olarak en çok Cosine similarity ve Euclidian distance yöntemlerin kullanıldığını keşfettik. Bu yöntemleri araştırarak temel mantığını kavradık fakat Java’da nasıl kullandığıyla ilgili kod örnekleri pek bulamadık. Bu nedenle RecommendationWindow classında birçok deneme yapıp en sonunda Euclidian distance yöntemini kullanarak üyenin verdiği ratinglere göre benzer üyeleri bulan findSimilarity() ve benzerlik oranına göre bulunan üyenin diğer yüksek rating verdiği kitapları seçen ArrayList<Books> tipinde recommendBook() metotlarını yazdık. Ayrıca aynı yaş aralığından da öneri almak için getBooksInAgeGroupISBN() metodunu ve aynı lokasyondan öneri almak için getBooksLocationISBN() metotlarını yazdık. Gelen sonuçları üye panelindeki gibi bir JTable’da listeledik.

11 - Veritabanı hatalarının düzeltilmesi

Admin panelinde üyelerin listelenmesinde bazı satırlar atlanarak listeleniyordu. JTable’ın oluşturulmasında ve tablodaGoster() methodunda herhangi bir hata bulamadığımız için bu sorunun neyden kaynaklandığını uzun süre çözemedik. Daha sonra veritabanında BX-Users tablosunda bazı kayıtların yaş sütununun NULL olduğunu fark ettik. NULL değerleri olan kayıtları programda Users classı constructorına parametre olarak gönderemediğimiz için Age sütunu NULL olan değerleri “*UPDATE `BX-Users` SET Age =* *FLOOR(20 + RAND() \* 10) WHERE Age IS NULL;*” komutuyla 20-30 arasında random değerler atayarak veritabanını güncelledik, bu şekilde sorunu çözmüş olduk. Programda denemeler yaparken bazı durumlarda BX-Users tablosundan silme işlemi yaptığımızda BX-Book-Ratings tablosundaki User-ID sütunu ile bağlı kullanıcının silinmediğini fark ettik. Bunun nedeninin bir tablonun başka bir tabloya foreign key kısıta ile bağlı olmasını ve veritabanı oluştururken “creates-db-tables.sql” dosyasında foreign keyleri tanımlarken “ON DELETE NO ACTION” seçeneği ile tanımlamamızdan dolayı olduğunu anladık. Bu kısmı “ON DELETE CASCADE” seçeneği ile değiştirerek sorunu çözdük.

2 - Yazılım Mimarisi

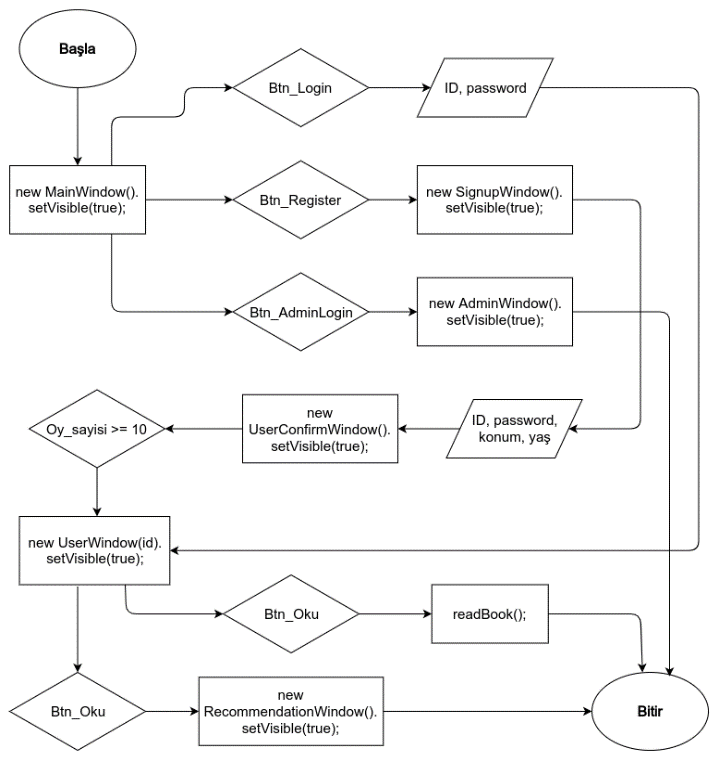
Programı yazmak için JDK 8u191 sürümüyle NetBeans 8.2 IDE’yi ve JDK 11.0.1 sürümüyle JetBrains IntelliJ IDEA’yı, veritabanı bağlantısı için Java Database Connectivity kütüphanesi içinde bulunan MySQL Connector Java 8.0.13’ü, GUI için Swing ve AWT API’lerini kullandık.



*Şekil 1: Programda bulunan paketler, sınıflar ve kütüphaneler*

Uygulamada MainWindow, LoginWindow, SignupWindow, UserConfirmWindow, UserWindow, StatisticsWindow, RecommendationWindow, AdminLoginWindow ve AdminWindow isimlerinde 9 tane JFrame formu şeklinde GUI classı kullandık. Sorgu sonuçları tabloda gösterebilmek için ve veritabanı ile program arasındaki bağlantıyı sağlamak amacıyla 4 tablo için Users, Books, BookRatings ve Admins isimlerinde normal class oluşturduk. Bu classları veritabanındaki kayıtları GUI’de oluşturduğumuz JTable’lara atabilmek için kullandık. Tüm classlar main kütüphanesinde bulunuyor. java.icons kütüphanesinde butonlar için gerekli .png biçiminde ikonlar bulunuyor. java.icepdf kütüphanesinde PDF okuma işlemi için gerekli kütüphaneler bulunuyor. java.books içerisinde telif hakkı bulunmayan 13 adet klasik edebiyat kitabı PDF’i bulunuyor.

3 - Akış Şeması



*Şekil 2: Akış şeması*

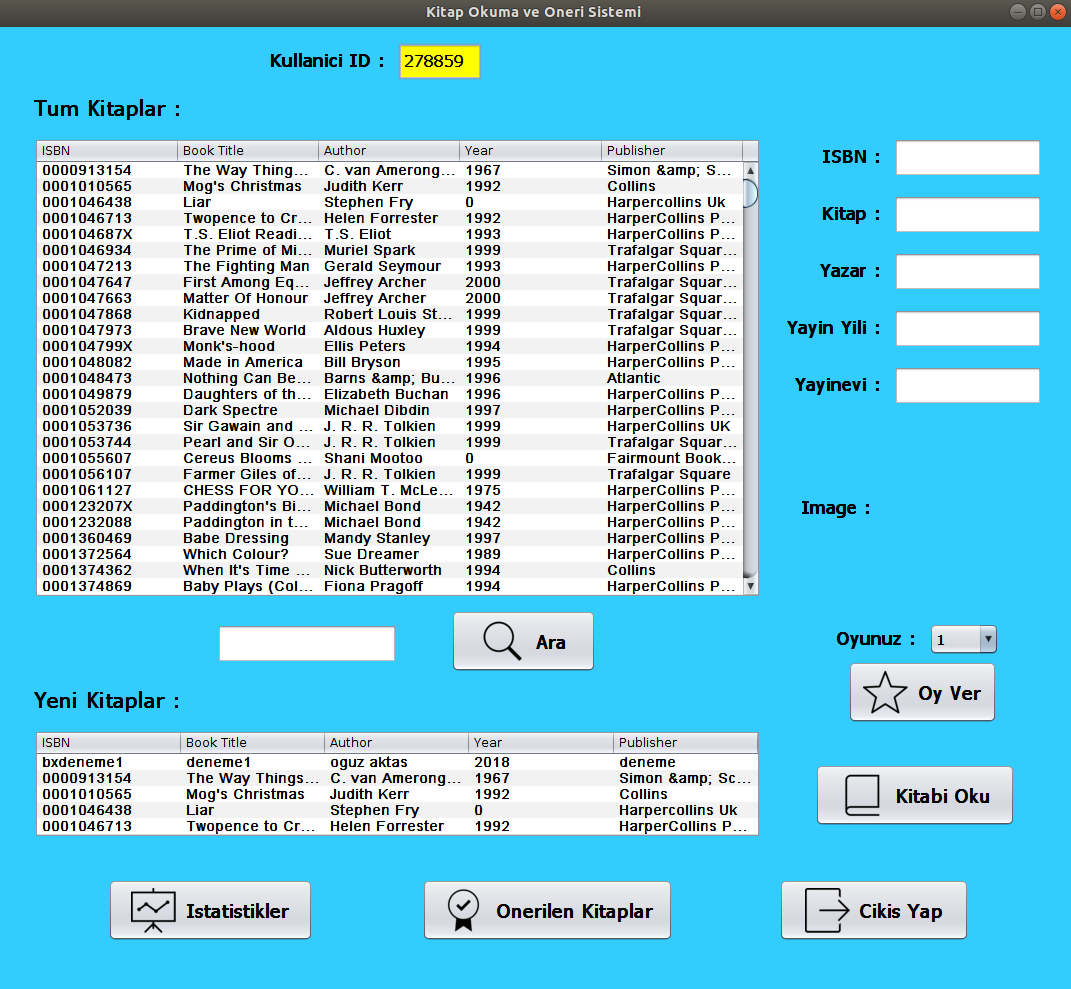
4 - Yazılımın Çalıştırılması ve Özellikleri

Programı çalıştırınca ilk olarak ana menü açılıyor.



*Şekil 3: Programın ilk çalıştırılması, ana menü*

Buradan üye girişi, üye kayıt ve yönetici girişi butonlarından istenen ekrana gidilebilir. Her GUI’de Ana Ekrana Git butonu ile ana ekrana dönülebilir. Giriş yaptıktan sonra admin ve üye panellerinde Çıkış Yap seçenekleri ile Login ekranlarına geri dönülebilir.



*Şekil 4: Üye paneli*

5 - Kaynakça

[1] <https://stackoverflow.com/>

[2] <http://www2.informatik.uni-freiburg.de/~cziegler/BX/> (Erişim tarihi: 12.10.2018)

[3] <https://dev.mysql.com/doc/refman/8.0/en/load-data.html> (Erişim tarihi: 12.10.2018)

[4] <http://www.mysqltutorial.org/import-csv-file-mysql-table/> (Erişim tarihi: 13.10.2018)

[5] <https://www.youtube.com/watch?v=i5UG6ACtnEg> (Erişim tarihi: 13.10.2018)

[6] <https://icons8.com/> (Erişim tarihi: 13.10.2018)

[7] <http://www.icesoft.org/wiki/display/PDF/Using+the+PDF+Viewer+Component> (Erişim tarihi: 21.10.2018)

[8] <https://www.planetebook.com/ebooks/> (Erişim tarihi: 21.10.2018)

[9] <https://www.baeldung.com/java-collaborative-filtering-recommendations> (Erişim tarihi: 26.10.2018)

[10] <https://www.youtube.com/watch?v=6mGMBipt7kU> (Erişim tarihi: 27.10.2018)

[11] <https://www.researchgate.net/publication/309277132_Collaborative_Filtering_using_Euclidean_Distance_in_Recommendation_Engine> (Erişim tarihi: 27.10.2018)

[12] <https://www.codeproject.com/Articles/1232150/Building-a-Recommendation-Engine-in-Csharp> (Erişim tarihi: 27.10.2018)