

**GAZİ ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BMT-311 VERİ TABANI YÖNETİM SİSTEMLERİ**

**2021-2022 GÜZ DÖNEMİ**

**PROJE RAPORU**

**GRUP ÜYELERİ:**

**<191816754> – <HÜSEYİN YAZICI>**

**<181816073> – <RESUL ÜNSAL>**

İÇİNDEKİLER

[**1. GİRİŞ 2**](#_Toc92731202)

[**Github Linki 3**](#_Toc92731203)

[**2. PROJE İÇERİĞİ 3**](#_Toc92731204)

[**2.1 PROJE TANITIMI 3**](#_Toc92731205)

[**Projenin Amacı 3**](#_Toc92731206)

[**Projenin Konusu 4**](#_Toc92731207)

[**Projenin Yetenekleri 4**](#_Toc92731208)

[**Projenin Problem Çözümleri 4**](#_Toc92731209)

[**2.2 PROJE TASARIM VE GELİŞTİRME AŞAMALARI 4**](#_Toc92731210)

[**2.3 KULLANILAN TEKNOLOJİLER 9**](#_Toc92731211)

[**3. VERİ TABANI TASARIM AŞAMALARI 10**](#_Toc92731212)

[**3.1 KAVRAMSAL TASARIM 10**](#_Toc92731213)

[**3.2 MANTIKSAL TASARIM 12**](#_Toc92731214)

[**3.3 FİZİKSEL TASARIM 14**](#_Toc92731215)

[**4. GÜNCEL VERİTABANI VE UML DİYAGRAMI 16**](#_Toc92731216)

[**4.1 Veri Tabanı Tabloların Son Hali 16**](#_Toc92731217)

[**4.2 Uml Diyagramı 17**](#_Toc92731218)

[**5. SONUÇ 18**](#_Toc92731219)

[**6. KAYNAKÇA 19**](#_Toc92731220)

# **1. GİRİŞ**

Proje, kütüphane ödünç alma sistemi olarak tasarlanmıştır. Kütüphane ödünç alma projemizde, sisteme üye olanların kitap kısıtlaması olmadan istediği her kitabı kolay ve hızlı bir şekilde alabilme imkanı sunulmuştur. Sisteme üye olmuş kullanıcıların okumak istedikleri kitaplara daha hızlı ulaşarak ödünç almalarını amaçlamaktadır. Kullanıcılar ödünç aldıkları kitabı belli bir süre zarfında iade etmek zorundalardır. Aksi halde sistemden başka bir kitap alamazlar. Projenin hedefi olarak ise kütüphaneyi daha kolay yönetilebilir bir hale getirmek ve kitaplara erişmek isteyen kişilerin kitaplara daha hızlı ulaşılabilmesi gibi ihtiyaçlara yönelik bir çözüm olarak geliştirilmesi hedeflenmiştir. Projenin hedef kitlesi, kitap okumayı seven ve okumak istediği kitaplara daha hızlı ulaşmak isteyen kişilerdir. Bu kişiler projenin hedef kitlesini olarak belirlenmiştir.

## Github Linki

* https://github.com/huseyinyazicii/GaziLibrary

# **2. PROJE İÇERİĞİ**

## **2.1 PROJE TANITIMI**

### Projenin Amacı

Proje, kütüphane ödünç alma sistemi olarak tasarlanmıştır. Sisteme üye olmuş kullanıcıların okumak istedikleri kitaplara daha hızlı ulaşarak ödünç almalarını amaçlamaktadır. Kullanıcılar ödünç aldıkları kitabı belli bir süre zarfında iade etmek zorundalardır. Aksi halde sistemden başka bir kitap alamazlar.

### Projenin Konusu

Kütüphane ödünç alma projemizde, sisteme üye olanların kitap kısıtlaması olmadan istediği her kitabı kolay ve hızlı bir şekilde alabilme imkanı sunulmuştur.

### Projenin Yetenekleri

* Kullanıcılar sisteme üye olabilecekler.
* Üyeler istedikleri kitapları eğer başkası almadıysa ödünç alabileckeler.
* Üyeler daha önce eklenmiş olan kitaplar arasından seçim yapabilirler.
* Üyeler yöneticelere mesaj göndererek onlarla iletişim kurabilecekler.
* Sadece yöneticiler kitap ekleme yapabilir.

### Projenin Problem Çözümleri

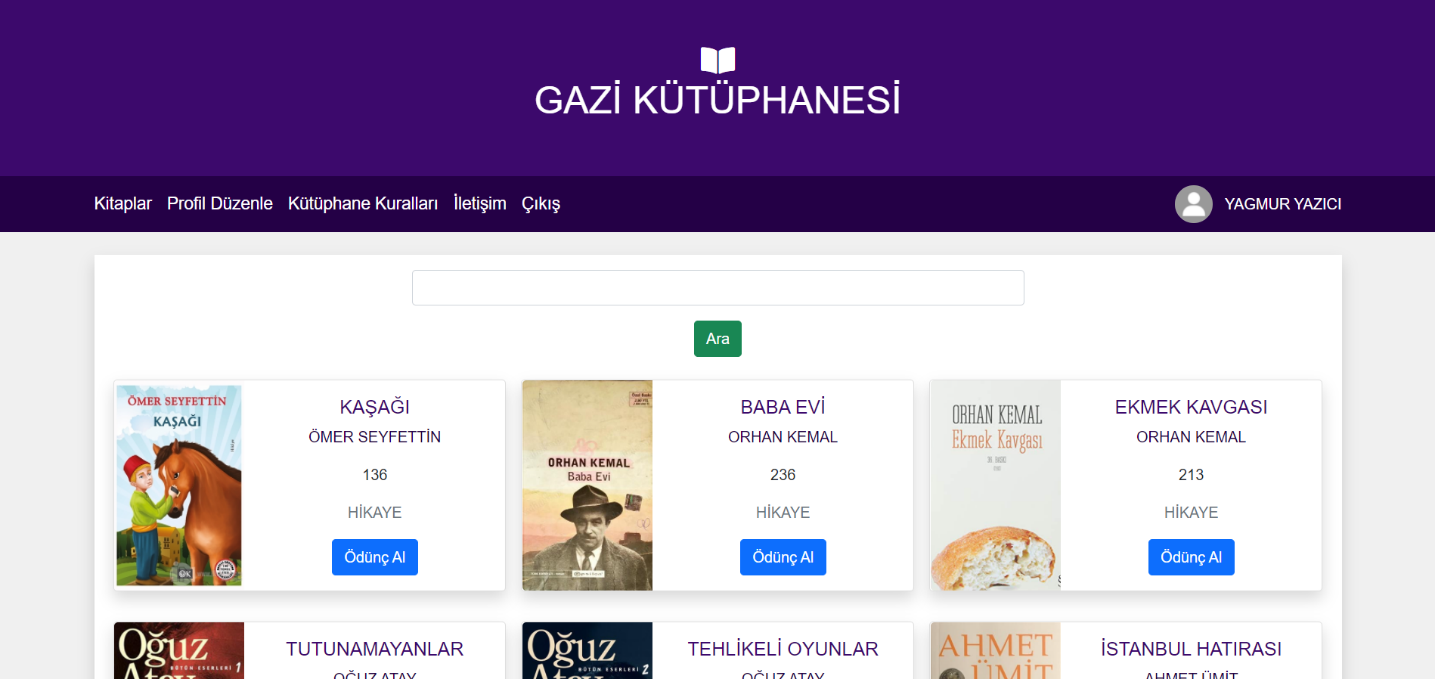
Kütüphaneyi daha kolay yönetilebilir bir hale getirmek ve kitaplara erişmek isteyen kişilerin kitaplara daha hızlı ulaşılabilmesi gibi ihtiyaçlara yönelik bir çözüm olarak geliştirilmiştir.

## **2.2 PROJE TASARIM VE GELİŞTİRME AŞAMALARI**

Projemizin tasarım aşamasında ilk olarak UX (User Experience) ve UI (User Interface) çalışmalarını gerçekleştirdik. Bunun için Figma tasarım aracını kullandık. Böylelikle web sitemizde bulunacak sayfaların taslak tasarımını bitirmiş olduk. Bu adım arayüzleri kodlarken daha hızlı olmamızı ve ortaya daha güzel arayüzler çıkarmamız açısından önemli bir adımdır. Günümüzde ki projelerde bu adıma verilen önem giderek artmaktadır. Daha sonra bu tasarımları web teknolojilerini (html, css, javascript) kullanarak kullanıcı arayüzlerimizi oluşturduk.

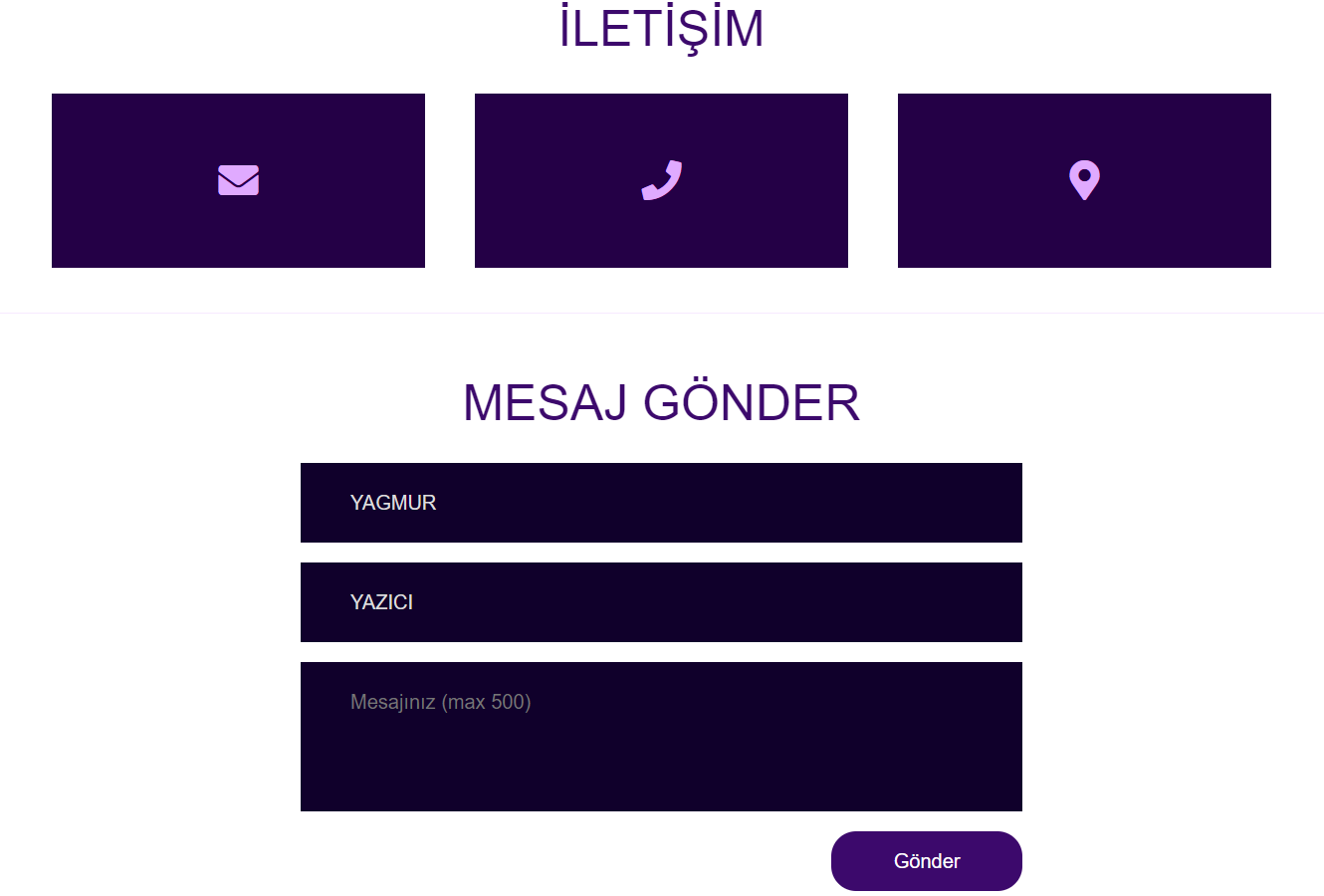
**Kitapların Listelenmesi**

Bu sayfada kütüphanede bulunan ve ödünç alınmamış tüm kitaplar listelenir. Kullanıcı arama kısmından istediği kitabın adına, yazarına ve türüne göre arama yapabilir. Eğer hali hazırda ödünç aldığı bir kitabı yoksa, istediği bir kitabı ödünç alabilir.



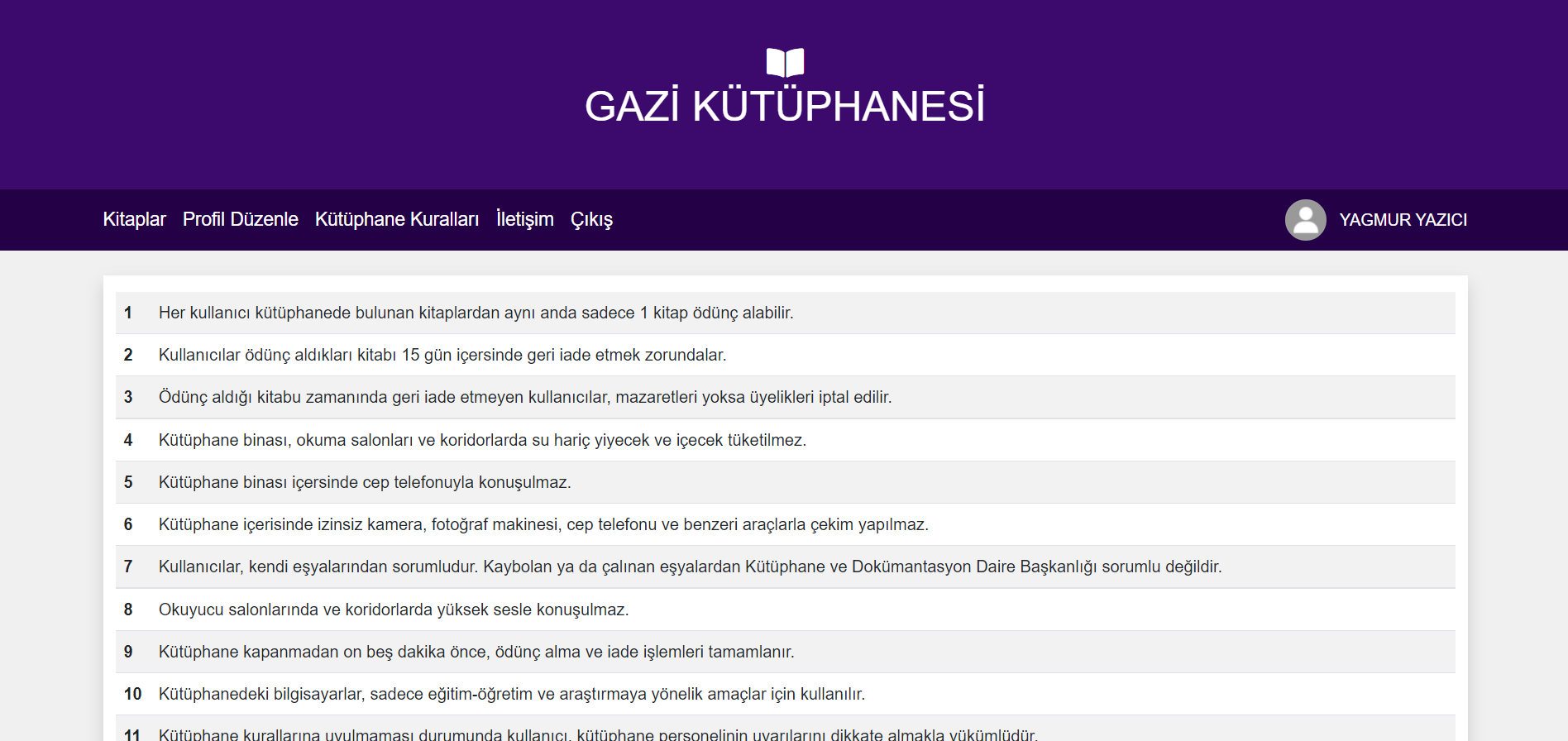
**İletişim**

Kullanıcı bu sayfada kütüphanenin iletişim bilgilerine erişebilir. Ayrıca personellere mesaj atarak iletişime geçebilir.



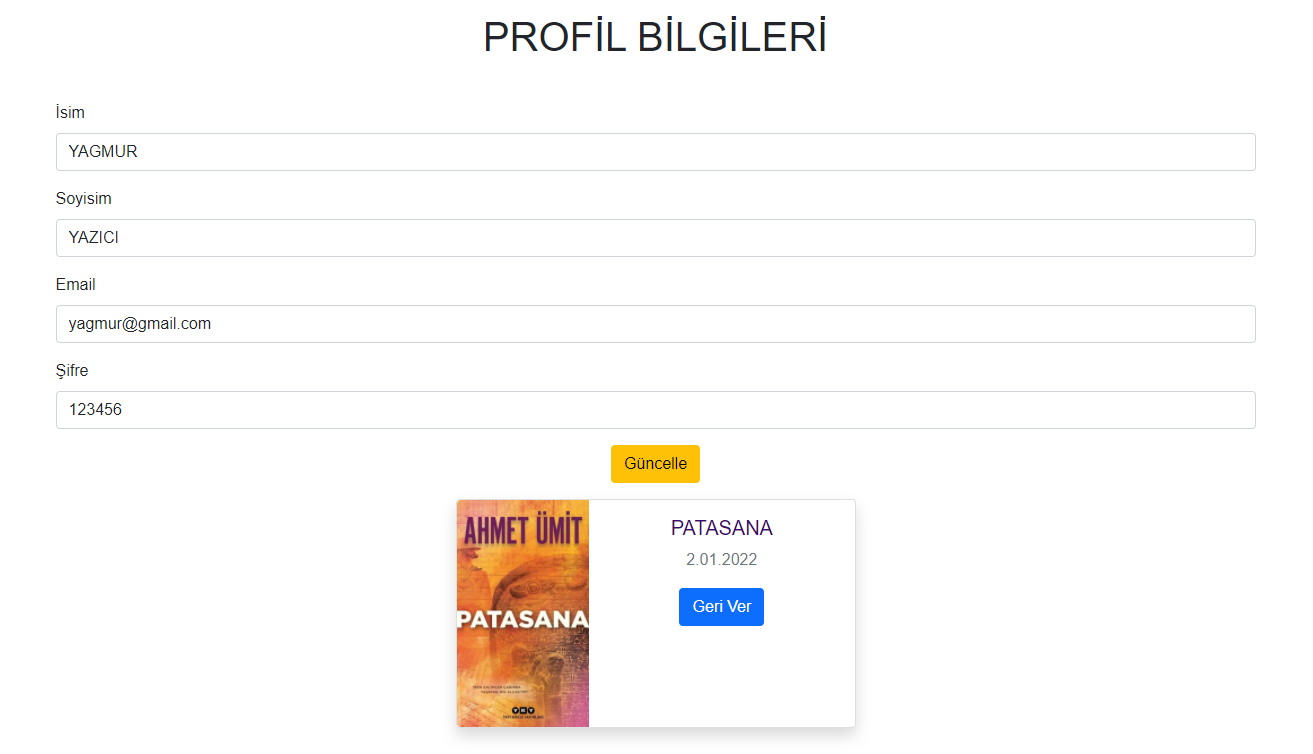
**Kütüphane Kuralları**

Bu sayfada kullanıcı kütüphane kurallarını öğrenebilir.



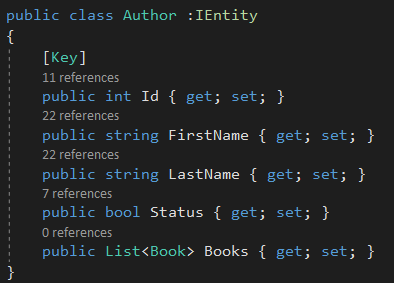
**Kullanıcı Profil Düzenleme**

Bu sayfada kullanıcı kendi bilgilerini ulaşabilir, değiştirebilir, ödünç aldığı kitabı görebilir ve ödünç aldığı kitabı iade edebilme işlemlerini gerçekleştirebilir.

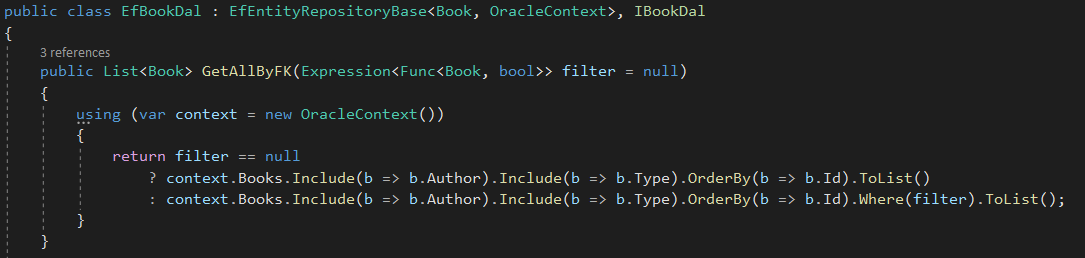


Görsel tasarım bittikten sonra veritabanı tasarımı adımları gerçekleştirilmiştir. Burada veritabanında bulunacak tablolar, bu tabloların sütunları, tablolar arasındaki ilişkiler ve gerekli normalizasyon işlemleri yapılmıştır. Bu adım 3.bölümde detaylı bir şekilde açıklanmıştır.

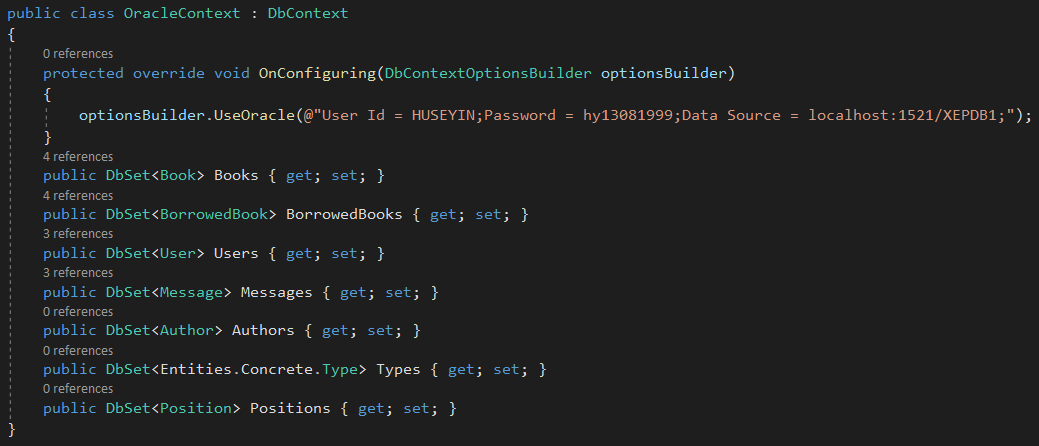
Sırada web sitemizi dinamik bir hale getirmek için gerekli MVC yapılandırılması kaldı. Bu adım projemizin en uzun sürecek ve çok fazla detayın bulunduğu kısımdır. Bu adımda neler yaptıklarımız özet olarak şu şekilde açıklayabiliriz. İlk olarak veritabanı işlemllerini gerçekleştirmek için veritabanı kodlarını yazmamız gerekiyor. Bu yüzden sırasıyla veritabanı tablolarına karşılık gelecek entity class’larımızı, veritabanına gerekli CRUD işlemlerini içeren her bir entity class için EntityDal class’larımızı (burada Repository Design Pattern kullanılmıştır) ve iş kurallarını yazmak için EntityManager class’larımız oluşturuyoruz. Veritabanı bağlantısı için gerekli Context class’ımızı yazıyoruz. Verilerimiz doğrulamak için Validation işlemlerini yapuyoruz. Son olarakta kullanıcının yapabileceği her türlü eyleme karşılık gelecek işlemleri Controller sınıflarında tanımlıyoruz.



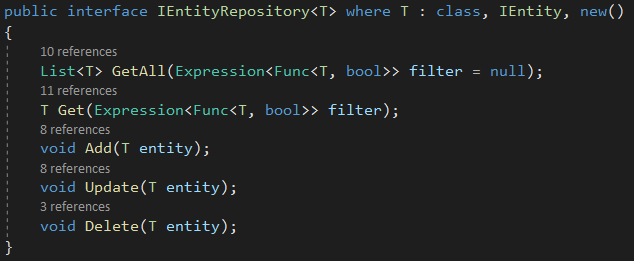
Author Entity Class



EfBookDal Class (Görselde tüm metodları bulunmamaktadır)



Context Class



Repository Pattern

## **2.3 KULLANILAN TEKNOLOJİLER**

Projemiz bir web uygulaması olduğundan dolayı bize web geliştirme ortamı sunan Microsoft tarafından geliştirilen **Asp.Net Core Mvc** ortamı kullanılarak projemiz geliştirilmiştir. Dolayısıyla programlama dili olarak **C#** kullanılmıştır.

Veritabanı olarak ise ileri seviyede bir ilişkisel veritabanı yönetim sistemi olması ile beraber tüm ilişkisel veritabanı sistemleri gibi yoğun miktarda verinin birçok kullanıcılı ortamlarda depolanmasını ve sağlıklı bir şekilde erişimini sağladığından dolayı **Oracle** kullanılmıştır.

Veritabanı ile uygulama arasında ilişki kurmak için yazılan kodların karmaşıklığını basite indiren ve daha okunaklı kod satırları yazabilmek için ORM (Object Relational Mapping) yapıları kullanılır. Biz de bu amaçlar doğrultusunda projemizde **Entity Framework** kullandık.

Uygulamamızda kullanıcıdan aldığımız verileri doğrulama kurallarından geçirmek için **Fluent Validation** kütüphanesini kullandık.

Projemizin kullanıcı arayüzlerini oluşturmak için her web projesinde kullanılan **Html**, **Css** ve **Javascript** teknolojilerini kullandık.

Daha hızlı bir şekilde tasarım yapabilmek için hazır bir css kütüphanesi olan **Bootstrap**’i kullandık.

Kullanıcının uygulamamız ile etikleşime geçtikten sonra kullanıcıya bilgilendirecek mesajlar için **Toast Notification** kütüphanesini kullandık.

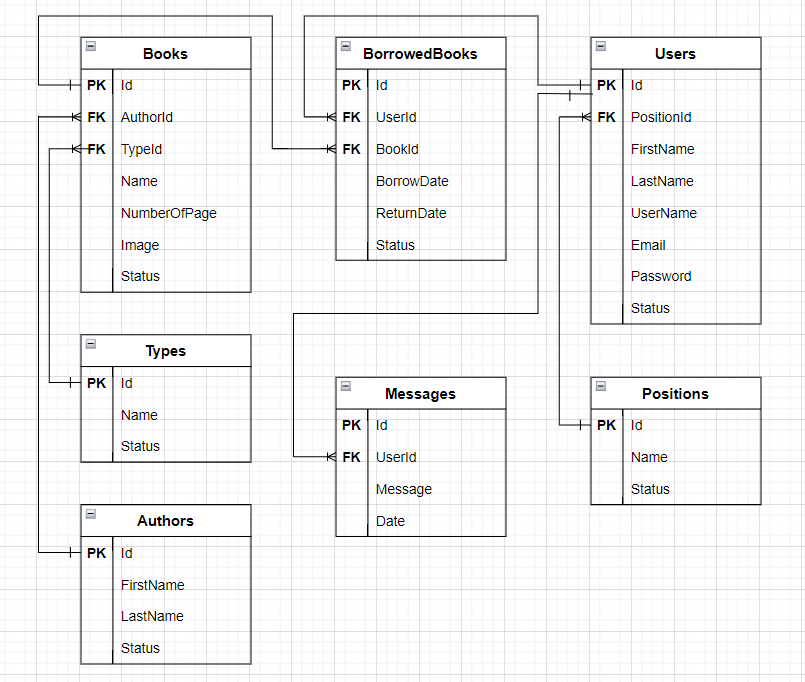
# 

# **3. VERİ TABANI TASARIM AŞAMALARI**

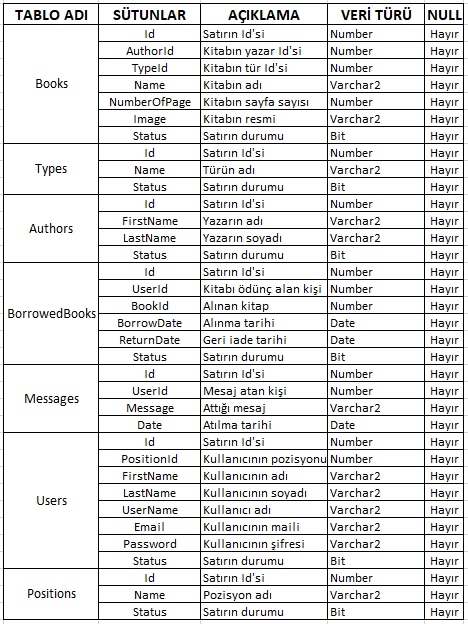
Veritabanı oluşturulurken gerekli veritabanı tasarım adımlarını şu sırayla gerçekleştirdik; oluşturulacak sistemin nelerden oluşması gerektiği ve hangi işlemlerin hangi aşamalarda yapıldığı belirlenmesi, oluşturulan bu metne göre varlık ilişki-modelinin oluşturulması, varlık ilişki modelinin tablolara dönüştürülerek tabloların oluşturulması, anahtar sütunların belirlenmesi, normalizasyon kurallarına uygun olmayan durumlarda tabloların bölünmesi ve ilişkilerin kurulmasıdır.

## **3.1 KAVRAMSAL TASARIM**

ER Diagram



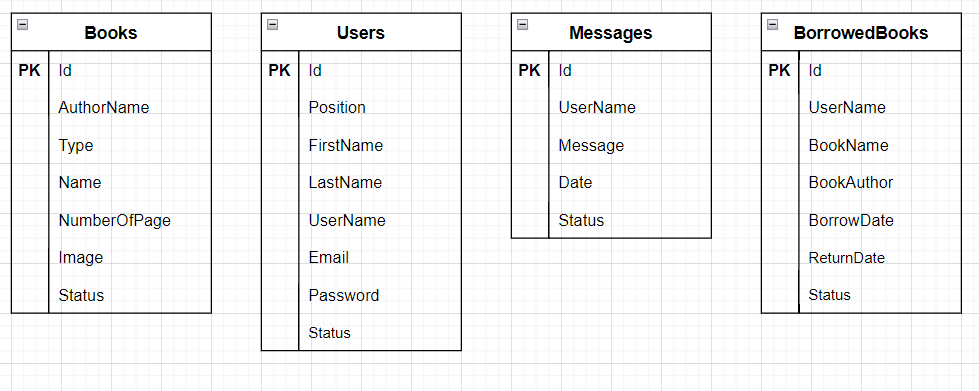
Veri Sözlüğü



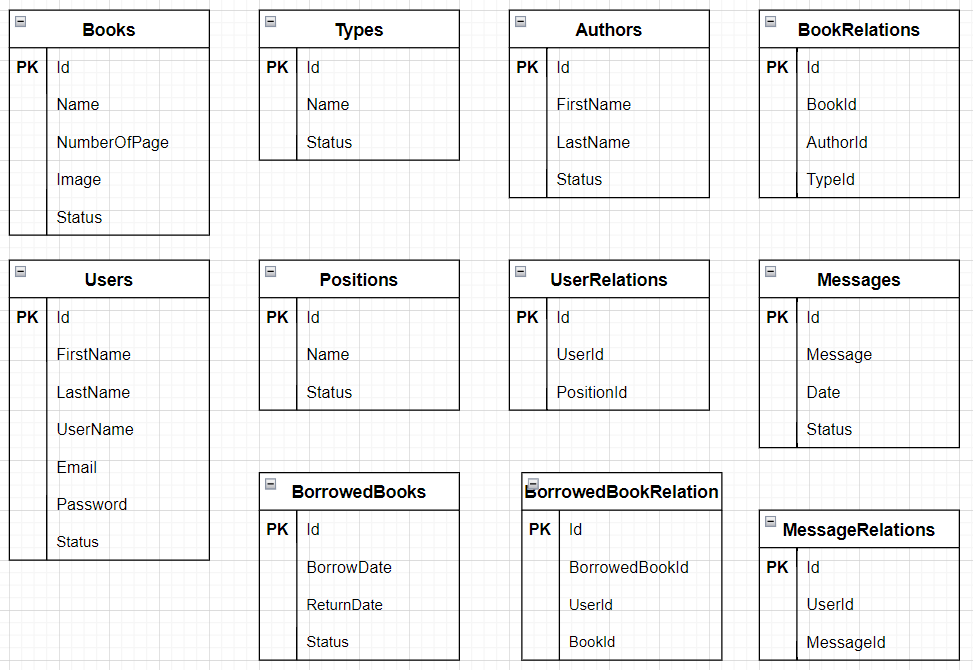
## **3.2 MANTIKSAL TASARIM**

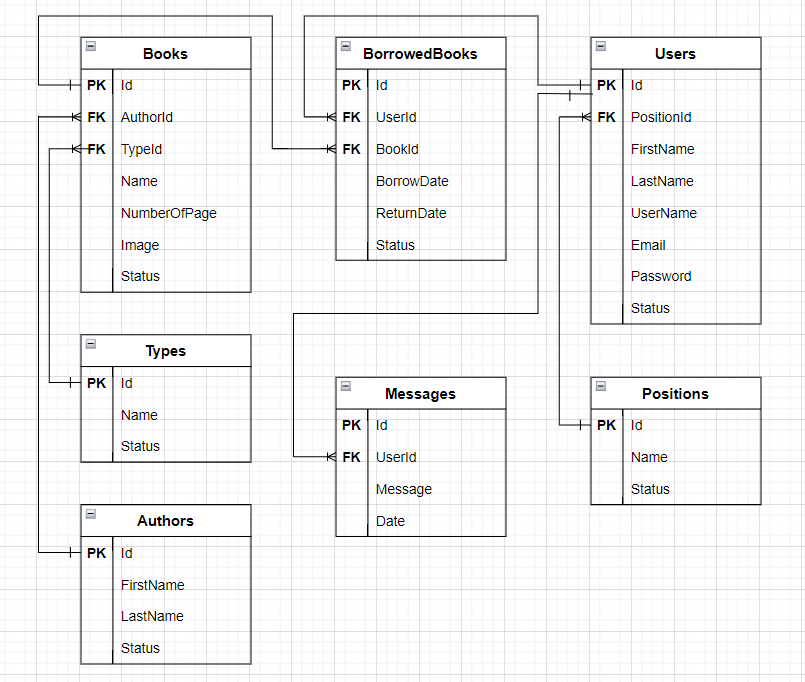
**Normalizasyon**

N1



**N2**



**N3**

## **3.3 FİZİKSEL TASARIM**

Tabloların ve Kısıtların Oluşturulması

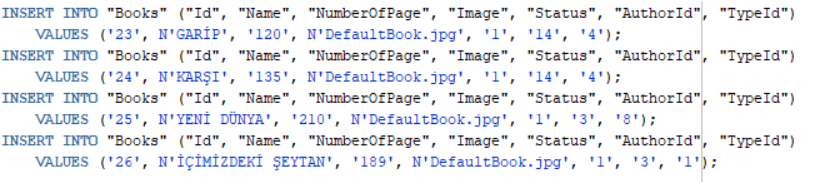
tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

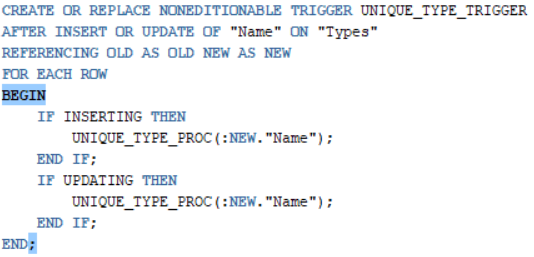
tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

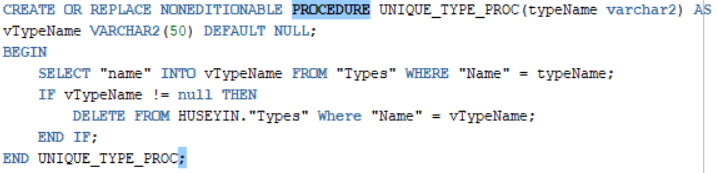
İlk Verilerin Girilmesi



Trigger

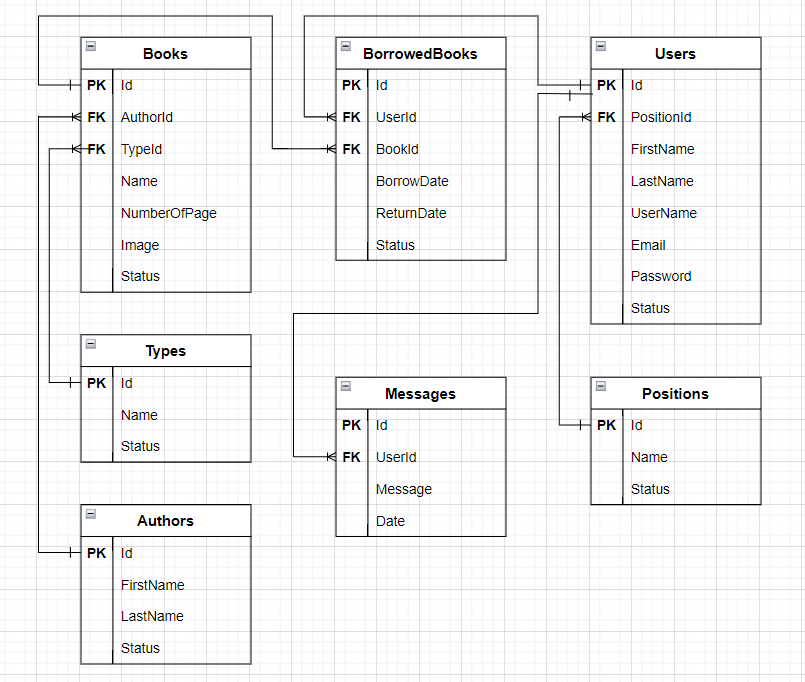


Procedure

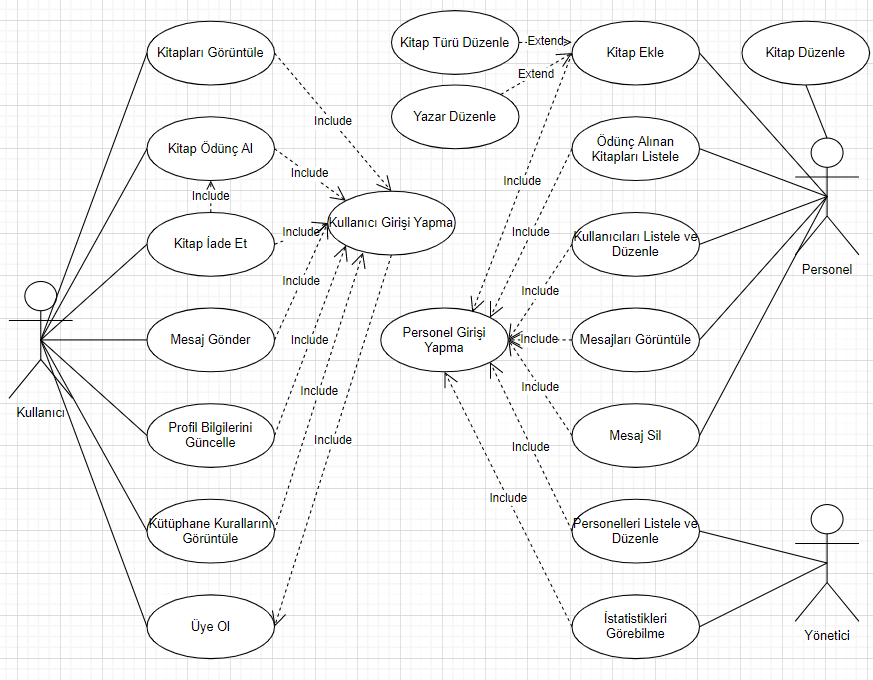


# **4. GÜNCEL VERİTABANI VE UML DİYAGRAMI**

## 4.1 Veri Tabanı Tabloların Son Hali



## 4.2 Uml Diyagramı



# **5. SONUÇ**

Kütüphane projemiz, projenin yapılış amacı olan kütüphaneyi daha yönetilebilir bir hale getirmek ve kitap okumak isteyen kişilerin aradıkları kitaplara daha hızlı ve kolay bir şekilde ulaşmalarını başarılı bir şekilde gerçekleştirmektedir. Hali hazırda bulunan bir çok kütüphane projesi bulunmaktadır. Bizim bu projelere göre artılarımız kullanıcı arayüzlerinin modern bir tasarıma sahip olması ve kullanıcıların kitap ödünç alabilme işlevine odaklı bir proje olmasıdır. Eksik yönü olarak ise personeller ve kullanıcılar arasındaki iletişimin zayıf olmasıdır.

Bu projeyi gerkçeklerştirirken edindiğimiz bir çok kazanım mevcuttur. Bunlardan ilki farklı bir veritabanı yönetim sistemi kullandığımızda ne gibi farklılıkların ortaya çıktığını öğrendik. Veritabanı tasarımı sırasında gerçeklerştirilen aşamaları daha detaylı bir şekilde öğrenmiş olduk. Ayrıca bu aşamaları gerçekleştirmenin önemini ve kod yazarken ki kolaylıklarını benimsemiş olduk.

# **6. KAYNAKÇA**

* [1]: Bootstrap - https://getbootstrap.com/docs/5.1/getting-started/download/
* [2]: Asp.Net Core Mvc - https://docs.microsoft.com/tr-tr/aspnet/core/mvc/overview?view=aspnetcore-5.0
* [3]: Fluent Validation - https://fluentvalidation.net/
* [4]: Figma - <https://www.figma.com/>
* [5]: Karşılaşılan Sorunlara Çözüm - <https://stackoverflow.com/>
* [6]: EntityFramework - https://docs.microsoft.com/tr-tr/ef/
* Not : Proje raporu yazılırken herhangi bir kaynak kullanılmamıştır.