

ANKARA ÜNİVERSİTESİ

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü



KUTUPHANE YONETİM SİSTEMİ

BLM4531 - Ağ Tabanlı Teknolojiler Ve Uygulamaları

Oğrenci: Ahmet Akgun
Oğrenci No: 22290602
Danışman: Enver Bağcı

Ocak 2026

Contents

1	Ozet	2
2	Giris ve Amac	2
2.1	Projenin Amaci	2
2.2	Kapsam	2
3	Sistem Mimarisi	2
3.1	Katmanli Mimari	2
3.2	Proje Dosya Yapisi	3
4	Veritabani Tasarimi	3
4.1	Entity-Relationship Diyagrami	3
4.2	Tablo Yapilari	4
4.2.1	Users Tablosu	4
4.2.2	Categories Tablosu	4
4.2.3	Books Tablosu	4
4.2.4	Loans Tablosu	4
5	API Endpoint Dokumantasyonu	5
5.1	Kimlik Dogrulama (Auth)	5
5.2	Kitaplar (Books)	5
5.3	Kategoriler (Categories)	5
5.4	Odunc Islemleri (Loans)	5
5.5	Admin Odunc Yonetimi	5
5.6	Istatistikler (Statistics)	6
6	Frontend Gelistirme	6
6.1	Web Uygulamasi	6
6.2	Kullanilan Teknolojiler	6
6.3	Sayfa Yapisi	6
7	Sonuc ve Degerlendirme	7
7.1	Tamamlanan Ozellikler	7
8	Kaynaklar	7

1 Ozet

Bu proje, ASP.NET Core Web API kullanılarak geliştirilen full-stack bir kutuphane yönetim sistemidir. Sistem, kitap yönetimi, kullanıcı yönetimi, ödünç alma/iade işlemleri ve istatistik raporlama gibi temel kutuphane işlevlerini içermektedir.

Kullanılan Teknolojiler:

- **Backend:** ASP.NET Core 8.0 Web API, Entity Framework Core, PostgreSQL
- **Web Frontend:** HTML5, CSS3, Bootstrap 5, JavaScript, Chart.js
- **Kimlik Doğrulama:** JWT (JSON Web Token)

2 Giriş ve Amac

2.1 Projenin Amacı

Bu projenin temel amacı, modern yazılım geliştirme pratiklerini kullanarak kapsamlı bir kutuphane yönetim sistemi geliştirmektir. Proje şu hedefleri kapsamaktadır:

1. Kitap kataloglama ve arama
2. Kullanıcı kayıt ve kimlik doğrulama
3. Ödünç alma taleplerinin yönetimi
4. Admin paneli ile tam yönetim
5. İstatistik ve raporlama

2.2 Kapsam

Sistem iki kullanıcı rolünü desteklemektedir:

- **Uye (Member):** Kitap görüntüleme, ödünç talep etme, kendi ödünçlerini takip etme
- **Admin:** Tüm işlemler + kitap/kategori yönetimi, talep onaylama/reddetme, istatistik görüntüleme

3 Sistem Mimarisi

3.1 Katmanlı Mimari

Proje, Clean Architecture prensiplerine uygun olarak katmanlı bir yapıda tasarlanmıştır:

Controllers (API Layer) AuthController, BooksController, CategoriesController, LoansController, AdminLoansController, StatisticsController
Services (Business Logic Layer) AuthService, BookService, CategoryService, LoanService, StatisticsService, JwtService
Repositories (Data Access Layer) GenericRepository, UnitOfWork Pattern
Data (Database Layer) LibraryDbContext, Entity Configurations
PostgreSQL Database

Figure 1: Katmanli Mimari Yapisi

3.2 Proje Dosya Yapisi

```

Library.Net2/
|-- Controllers/           # API Endpointleri
|-- Services/             # Is mantigi
|-- Repositories/        # Veri erisim
|-- Data/                 # DbContext, Seeder
|-- Models/
|   |-- Domain/           # Entityler
|   |-- DTOs/             # Data Transfer Objects
|   |-- Enums/            # Enum tipleri
|-- wwwroot/              # Frontend dosyalari
|   |-- pages/            # HTML sayfalari
|   |-- js/               # JavaScript
|   |-- css/              # Stiller
|   |-- images/           # Gorseller
|-- Program.cs            # Uygulama giris noktasi

```

4 Veritabani Tasarimi

4.1 Entity-Relationship Diyagrami

Sistem 4 ana entity icermektedir:

Tablo	Aciklama	Iliskiler
Users	Kullanici	1:N Loans
Categories	Kitap kategorileri	1:N Books
Books	Kitaplar	N:1 Category, 1:N Loans
Loans	Odunc kayitlari	N:1 User, N:1 Book

Table 1: Veritabani Tablolari

4.2 Tablo Yapilari

4.2.1 Users Tablosu

```
CREATE TABLE "Users" (  
    "Id" SERIAL PRIMARY KEY,  
    "FullName" VARCHAR(100) NOT NULL,  
    "Email" VARCHAR(100) UNIQUE NOT NULL,  
    "PasswordHash" TEXT NOT NULL,  
    "Role" INTEGER NOT NULL -- 0: Member, 1: Admin  
);
```

4.2.2 Categories Tablosu

```
CREATE TABLE "Categories" (  
    "Id" SERIAL PRIMARY KEY,  
    "Name" VARCHAR(100) NOT NULL,  
    "IsActive" BOOLEAN DEFAULT TRUE  
);
```

4.2.3 Books Tablosu

```
CREATE TABLE "Books" (  
    "Id" SERIAL PRIMARY KEY,  
    "Title" VARCHAR(200) NOT NULL,  
    "Author" VARCHAR(100) NOT NULL,  
    "CategoryId" INTEGER REFERENCES "Categories"("Id"),  
    "ISBN" VARCHAR(13),  
    "PublishYear" INTEGER,  
    "ImageUrl" TEXT,  
    "IsAvailable" BOOLEAN DEFAULT TRUE  
);
```

4.2.4 Loans Tablosu

```
CREATE TABLE "Loans" (  
    "Id" SERIAL PRIMARY KEY,  
    "UserId" INTEGER REFERENCES "Users"("Id"),  
    "BookId" INTEGER REFERENCES "Books"("Id"),  
    "LoanDate" TIMESTAMP NOT NULL,  
    "DueDate" TIMESTAMP,  
    "ReturnDate" TIMESTAMP,  
    "Status" INTEGER NOT NULL, -- Pending, Borrowed, Returned,  
    Late  
    "AdminNote" TEXT  
);
```

5 API Endpoint Dokumantasyonu

5.1 Kimlik Dogrulama (Auth)

Metod	Endpoint	Aciklama
POST	/api/Auth/register	Yeni kullanıcı kaydı
POST	/api/Auth/login	Giris, JWT token doner
GET	/api/Auth/me	Mevcut kullanıcı bilgisi

Table 2: Auth API Endpointleri

5.2 Kitaplar (Books)

Metod	Endpoint	Aciklama
GET	/api/Books	Tum kitapları listele
GET	/api/Books/{id}	Kitap detayı
GET	/api/Books/search?query=x	Kitap ara
POST	/api/Books	Yeni kitap ekle (Admin)
PUT	/api/Books/{id}	Kitap guncelle (Admin)
DELETE	/api/Books/{id}	Kitap sil (Admin)

Table 3: Books API Endpointleri

5.3 Kategoriler (Categories)

Metod	Endpoint	Aciklama
GET	/api/Categories	Tum kategoriler
GET	/api/Categories/active	Aktif kategoriler
POST	/api/Categories	Kategori ekle (Admin)
PUT	/api/Categories/{id}	Guncelle (Admin)
DELETE	/api/Categories/{id}	Sil (Admin)

Table 4: Categories API Endpointleri

5.4 Odunc Islemleri (Loans)

Metod	Endpoint	Aciklama
GET	/api/Loans/my	Kullanıcının oduncleri
POST	/api/Loans	Odunc talebi olustur
POST	/api/Loans/{id}/return	Iade et

Table 5: Loans API Endpointleri

5.5 Admin Odunc Yonetimi

Metod	Endpoint	Aciklama
GET	/api/Admin/Loans	Tum oduncler
GET	/api/Admin/Loans/pending	Bekleyen talepler
GET	/api/Admin/Loans/late	Gecikmis oduncler
POST	/api/Admin/Loans/{id}/approve	Talebi onayla
POST	/api/Admin/Loans/{id}/reject	Talebi reddet

Table 6: Admin Loans API Endpointleri

5.6 İstatistikler (Statistics)

Metod	Endpoint	Aciklama
GET	/api/Statistics	Genel istatistikler
GET	/api/Statistics/categories	Kategori bazli
GET	/api/Statistics/monthly-loans	Aylik trend
GET	/api/Statistics/top-books	Populer kitaplar

Table 7: Statistics API Endpointleri

6 Frontend Gelistirme

6.1 Web Uygulaması

Web frontend, modern ve responsive bir tasarıma sahiptir:

- **Ana Sayfa:** Kitap listesi, arama, kategori filtreleme
- **Kitap Detay:** Kitap bilgileri, odunc alma butonu
- **Kullanıcı Paneli:** Odunclerim, profil bilgileri
- **Admin Paneli:** Dashboard, kitap/kategori/odunc yonetimi, istatistikler

6.2 Kullanılan Teknolojiler

- Bootstrap 5 - Responsive tasarım
- Chart.js - İstatistik grafikleri
- Vanilla JavaScript - API iletisimi

6.3 Sayfa Yapısı

```
wwwroot/
|-- pages/
|   |-- index.html           # Ana sayfa
|   |-- login.html          # Giris
|   |-- register.html       # Kayit
|   |-- book-detail.html     # Kitap detay
```

```
|  |-- my-loans.html      # Odunclerim
|  |-- admin-dashboard.html # Admin paneli
|  |-- statistics.html    # Istatistikler
|-- js/
|  |-- api.js             # API client
|  |-- admin.js           # Admin islemleri
|  |-- auth.js            # Kimlik dogrulama
|-- css/
|  |-- style.css          # Genel stiller
```

7 Sonuc ve Degerlendirme

7.1 Tamamlanan Ozellikler

- JWT tabanlı kimlik dogrulama sistemi
- Kitap ve kategori CRUD islemleri
- Odunc talep, onaylama, reddetme, iade sistemi
- Gecikmis odunc takibi
- Kapsamli istatistik ve raporlama (grafikler dahil)
- Responsive web arayuzu

8 Kaynaklar

1. Microsoft ASP.NET Core Documentation - <https://docs.microsoft.com/aspnet/core>
2. Entity Framework Core - <https://docs.microsoft.com/ef/core>
3. Chart.js - <https://www.chartjs.org>
4. Bootstrap 5 - <https://getbootstrap.com>
5. JWT.io - <https://jwt.io>