KÜTAHYA SAĞLIK BİLİMLERİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

Veri Odaklı Sistem Tasarımı

Tasarım Dokümanı Hazırlama Kılavuzu

Halil Şimşek **2218111004**

14.03.2024

1 Giriş

Bu proje, bir kitap yönetim paneli geliştirmeyi amaçlayan bir uygulamadır.Kitap yönetim paneli, sadece kitap ekleme, silme, listeleme, güncelleme, arama ve filtreleme gibi temel işlevleri sağlamakla kalmayıp aynı zamanda kullanıcı dostu bir arayüzle donatılarak kitap koleksiyonlarını etkili bir şekilde yönetmeyi hedefliyorum.Projedeki veri seti, kitapların yanı sıra yazar bilgileri,fiyatlar ve kategoriler gibi çeşitli parametreleri içerecektir.Aynı zamanda bu proje 4 katmanlı bir mimari yapısana sahip bir çalışmadır.

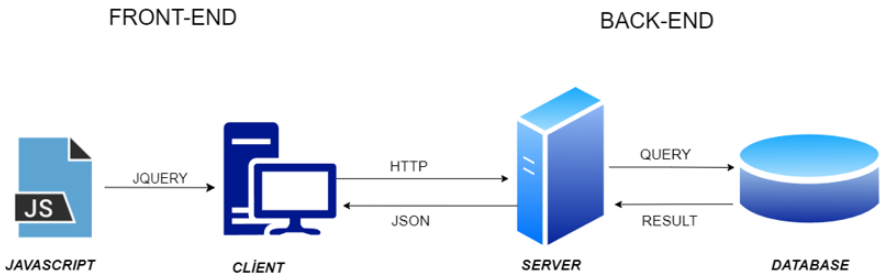
Projede client katmanında kullanıcı etkileşimini sağlamak amacıyla JavaScript, HTML, ve CSS dilleri kullanılacaktır. Server katmanında C Sharp programlama dili kullanırken aynı zamanda .NET CORE frameworkü ve ASP.NET MVC modelinden yararlanacağım.

Database katmanında ise verileri depolamak için PostgreSQL veritabanı yönetim sistemini kullanacağım.Bu teknoloji yığını, projenin ihtiyaçlarına uygun bir şekilde seçilmiştir ve kullanıcıya kitap yönetim panelini kolayca kullanma ve yönetme imkanı sağlayacak bir altyapı sunmayı amaçlamaktadır.Detaylı mimari ve metodolojik açıklamalar, "Metodoloji" bölümünde bulunacaktır.

2 Literatür Araştırması

Bu proje, Library-management-system adlı açık kaynak kodlu bir projeden alıntılar içerebilir. Library-management-system projesi, [`https://github.com/prabhakar267/library-management-system.git`] adresinden eişilebilir. Aynı zamanda Library-Management-System-JAVA adlı proje üzerinden de bilgiler toplanmıştır. Library-Management-System-JAVA projesine [`https://github.com/OSSpk/Library-Management-System-JAVA.git`] adresinden erişebilirsiniz

3 Metodoloji



Projenin 4 katmanlı bir yapıda olduğunu belirtmiştim. Katmanlarda sırasıyla bahsedecek olursam öncelikle Javascript katmanından başlamak isterim. Javascript katmanında jquery ile ajax ve isteği oluşturup bir olayın tetiklenmesini sağlıyorum. Bu olay tetiklemesi sonucunda client bir http isteği gönderiyor. Bu http isteği server katmanında tetiklenen olaya göre içerde bir sorguyu çalıştırıyor. Bu sorgu veri tabanına bağlanarak veri tabanında gerekli işlmeleri gerçekleşririp bir result dönüyor. Server katmanına gelen bu result json formatta client katmanına geliyor. Bu projenin 4 katmanlı olmasını sağlayan durum web sayfasının server tarafından hazır halde gelmeyip javascript katmnaında dinamik olarak hazırlanıyor olmasından kaynaklıdır.

Server tarafında, projenin iş mantığını ve veri tabanı etkileşimini yönetmek amacıyla .NET CORE üzerinde yazıp ASP.NET MVC modelini kullancağım. MVC mimarisi, uygulamayı modülerleştirir ve işlevselliği katmanlara bölerek daha yönetilebilir bir yapı oluşturur. Model katmanı, veri tabanına erişim ve veri tabanı ilişkileri gibi veri işleme süreçlerini içerir. View katmanı, kullanıcının etkileşimde bulunduğu arayüzü temsil ederken, Controller katmanı, kullanıcı komutlarını alarak bu komutları işleyen ara bir katman olarak görev yapar. Bu sayede, sistemin hem veri tabanı yönetimi hem de iş mantığı daha düzenli ve modüler bir yapı kazanmasını sağlayacaktır. Asp.Net MVC modeli hakkında araştırmalarda kullanılmıştır.**?**

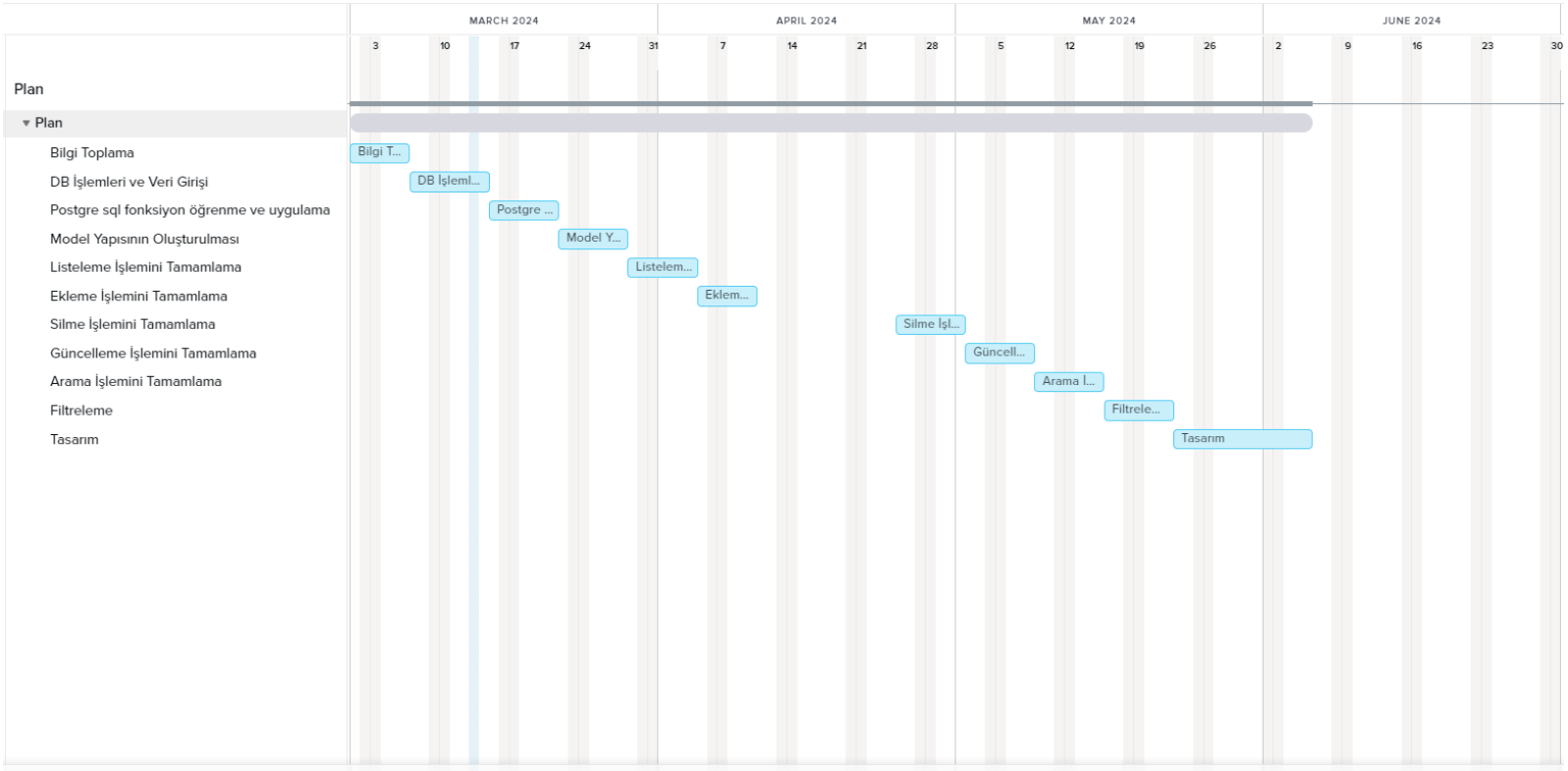
MVC mimarisini kullanırken aynı zamanda SOLID prensiblerine uygun sekilde kod yazmayı planlıyorum. SOLID prensiblerinden bahsedecek olursam beş temel ilkeye dayanıyor. SOLID prensipleri, yazılım geliştirme sürecinde kodun okunabilirliğini, bakımını, esnekliğini ve yeniden kullanılabilirliğini artırmak için önemlidir. Bu prensiplere uygun kodlar, daha az hata içerir, daha kolay anlaşılabilir ve daha sürdürülebilirdir.
Bu ilkeler ile ilgili araştırmamı ... adresinden yapmış bulunmaktayım.**?**

Aynı zamanda [https://chat.openai.com/c/d70b0aba-f98f-4fad-a808-5ec4815176d4 ile de yararlandım.]

Frontend tarafında ise JavaScript kullanarak sayfaları dinamikleştirme işlevselliği eklemeyi planlıyorum. HTML ve CSS, kullanıcı arayüzünü tasarlamak ve düzenlemek için kullanılacaktır. Bu sayede, kullanıcılar kitapları daha etkileşimli bir şekilde listeleyebilecek, arayabilecek ve filtreleyebileceklerdir.

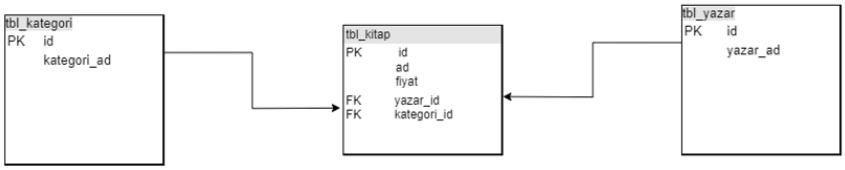
Bu proje, modern teknolojilerin entegrasyonunu kullanarak uygulama geliştirme sürecindeki önemli konulara odaklanmakta ve hem veritabanı yönetimi hem de web uygulaması geliştirme konularında öğrenmeye yönelik bir deneyim sunmaktadır. Bu seçilen teknolojiler, projenin amacına uygun olarak sistemin etkili bir şekilde işleyişini destekleyeceği fikrindeyim.

Şekil 1: GANTT CHART



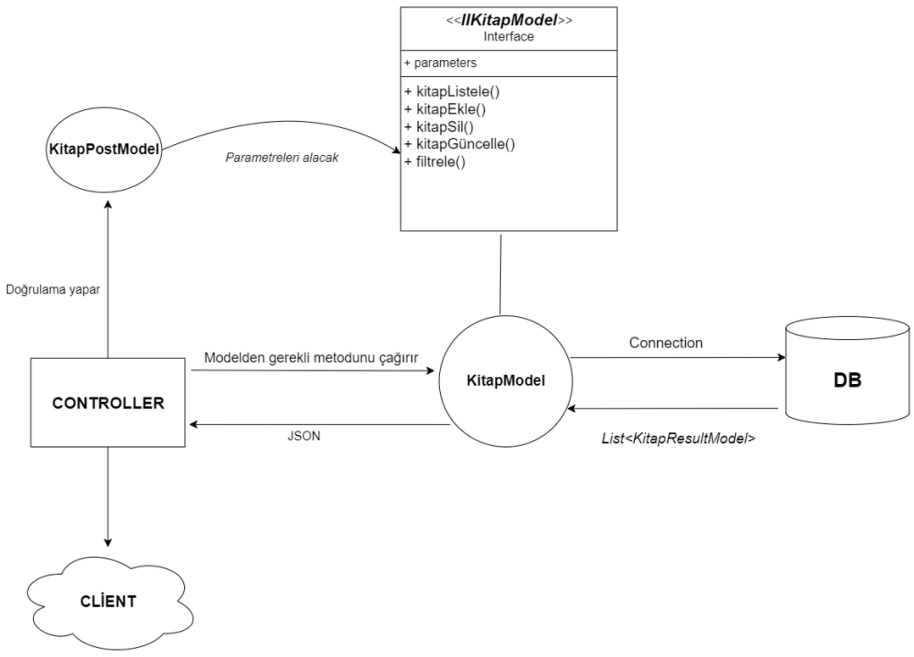
Şekil 1'de görebileceği üzere iş akış planı gösterilmektedir.

4 Veri Tabanı ve Veriler



Database katmanında PostgreSQL'i tercih etmemin sebebi PostgreSQL'in açık kaynaklı yapısı ve lisans ücreti gerektirmemesi gibi avantajlarının olmasıdır.Çekirdek başına ve sürümüne göre maliyet diğer dillerde oldukça fazladır. PostgreSQL ile projede temel verileri depolamak için kullanılacak olan "kitap," "kategori," ve "yazar" gibi tabloları içermektedir. Database işlemlerinde verinin kullanabilmesi için PostgreSQL fonksiyonlarından yararlanacağım.

PostgreSql fonksiyonları ile ilgili araştırmaları [`https://github.com/tubitak-bilgem-yte/pg-gelistirici.git`] adresinden gerçekleştirdim.



IKitapModel Arayüzü: Bu arayüz, kitaplarla ilgili işlemlerin sözleşmesini tanımlar. Yani, kitap listeleme ve kitap ekleme vb. işlevlerin prototiplerini içerir.Yapılacak işlemler için gerekli parametreleri KitapPostModel'den alır ve sonuç olarak bir KitapResultModel listesi döndürür.

KitapModel Sınıfı: IKitapModel arayüzünü uygular (implement eder) ve bu arayüzde tanımlanan metodların somut davranışlarını içerir. PostgreSQL veritabanına bağlanır, bir SQL sorgusu çalıştırır ve sonuçları bir KitapResultModel listesine dönüştürür. Veritabanı bağlantısı ve sorgu işlemleri için NpgsqlConnection ve NpgsqlCommand nesnelerini kullanır.

KitapPostModel Sınıfı: Kullanıcıdan alınan verileri taşımak için kullanılır. Bu model,örneğin kitap listeleme işlemi sırasında kullanıcı tarafından sağlanan verileri (örneğin, bir kitap adı) içerir.

KitapResultModel Sınıfı: Veritabanından dönen sonuçları temsil eder. Her bir KitapResultModel nesnesi, bir kitap kaydının id, ad, fiyat, yazarAd, kategoriAd gibi özelliklerini içerir.

HomeController Sınıfı: MVC'nin Controller bileşenidir ve kullanıcı isteklerini yönetir. Index metodu, uygulamanın ana sayfasını döndürür ve genellikle kullanıcıya HTML içeriği sunar. Örneğin kitapListe metodu, bir HTTP POST isteği ile çağrıldığında, KitapListePostModel tipindeki veriyi alır, modelin geçerliliğini kontrol eder ve KitapModel üzerinden kitap listesini çeker. Daha sonra bu listeyi JSON formatında istemciye (web tarayıcısına) geri döndürür.

- Kullanıcı, web arayüzünde bir eylem gerçekleştirir (örneğin, bir kitap listeleme isteği gönderir).
- İstemci tarafı kod (JavaScript), bu isteği HomeController'ın kitapListe metoduna POST olarak gönderir.
- HomeController, KitapListePostModel nesnesini alır ve doğrulama yapar.
- Doğrulama başarılıysa, KitapModel'in KitapListe metodunu çağırır ve veritabanından gerekli verileri çeker.

- Elde edilen veriler KitapResultModel listesi olarak dönüştürülür.
- Bu liste, JSON formatında istemciye geri gönderilir.
- İstemci tarafı JavaScript kodu, bu verileri alır ve kullanıcıya göstermek üzere işler.

Asp.Net MVC model sınıfı hakkında araştırmalarda kullanılmıştır. **?**
? ? ?

5 Beklenen Sonuçlar

Panelin, kullanıcıların kolayca erişebileceği ve etkileşimde bulunabileceği kullanıcı dostu arayüze sahip olması beklenmektedir . Bu, kullanıcıların kitap ekleme, düzenleme, silme gibi işlemleri hızlı ve verimli bir şekilde gerçekleştirebilmelerini sağlayacaktır.

Panelin, farklı kategorilerde kitapları yönetme yeteneği sağlaması beklenmektedir. Kullanıcıların kitapları başlık, yazar, gibi özelliklere göre arama yapabilmeleri ve filtreleyebilmeleri gerekmektedir.

Panelin, kullanıcıların kitap koleksiyonu içinde hızlıca arama yapabilmelerini sağlayacak gelişmiş arama ve filtreleme özelliklerine sahip olması beklenmektedir. Bu özellikler, kullanıcıların istedikleri kitapları kolayca bulmalarını sağlayacak şekilde tasarlanmalıdır.

Kaynakça

M. C. Sözeri, "Hayatımda hiç kitap yazmadım. bu yüzden önsöz böyle mi yazılır bilmiyorum. ama hayatımda yapmak istediklerimden bir tanesinin olduğunu söyleyebilirim kitap yazmak. kişisel blogumda (mcansozeri. wordpress. com) nisan 2010 tarihinden beri birçok yazı yayınladım ve 300bin sayfa görüntülenmeye doğru gidiyor. özellikle asp .net mvc konusunda çok fazla ingilizce kaynak okudum, blog takip ettim ve,"

O. TURAN and Ö. Ö. TANRIÖVER, "Solid ilkelerinin microsoft vs code metriğine etkisinin deneysel olarak değerlendirilmesi," *AJIT-e*, vol. 9, no. 34, p. 7, 2018.

Ö. B. Kurt, "Online stok yönetimi ve öneri sistemi," Master's thesis, Fen Bilimleri Enstitüsü.

T. Çelik, O. Bayat, A. D. Duru, and O. N. Uçan, "Öncül bilgi tabanlı kan paylaşım sistemi tasarım ve uygulaması," 2018.

A. T. Tüzemen, "Akademik bilgi sistemi tasarımı izmir örneği," Master's thesis, Fen Bilimleri Enstitüsü, 2014.

E. Keskin, *FACEBOOK: FACEBOOK PROGRAMLAMA-FACEBOOK PROGRAMMING*, vol. 126. KODLAB YAYIN DAĞITIM YAZILIM LTD. ŞTİ., 2014.