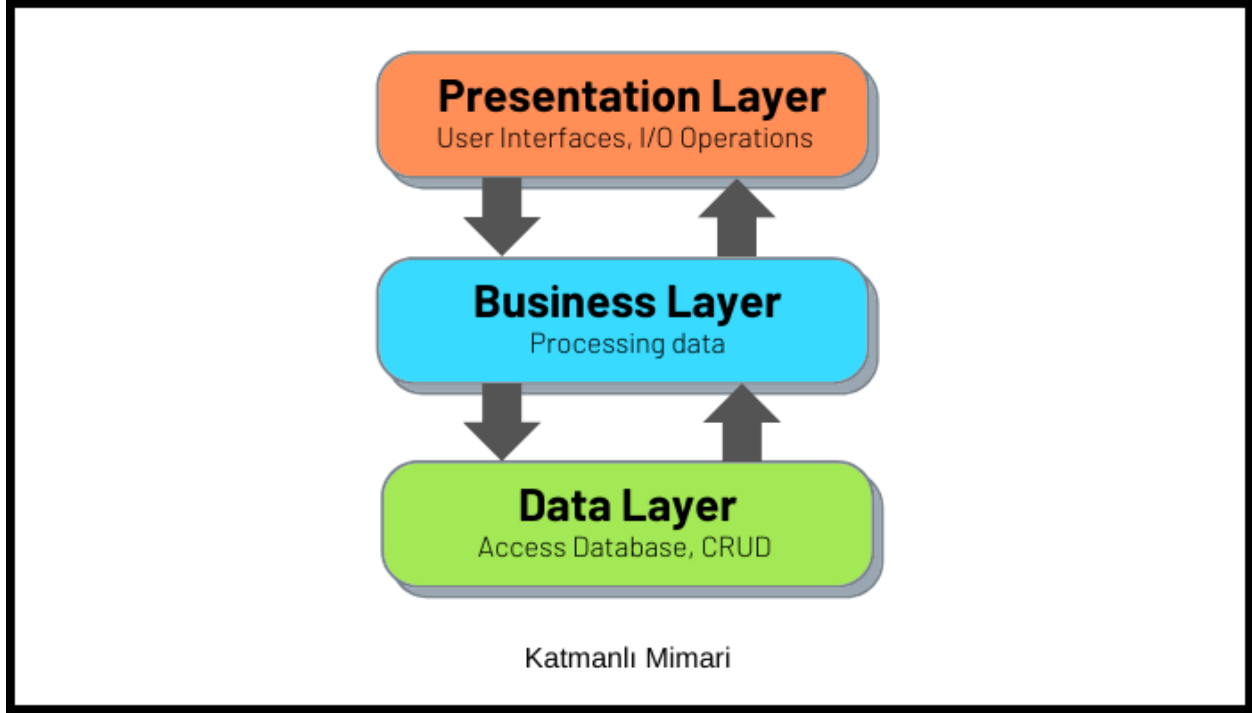


N-Tier Architecture(Çok(n) Katmanlı Mimari)

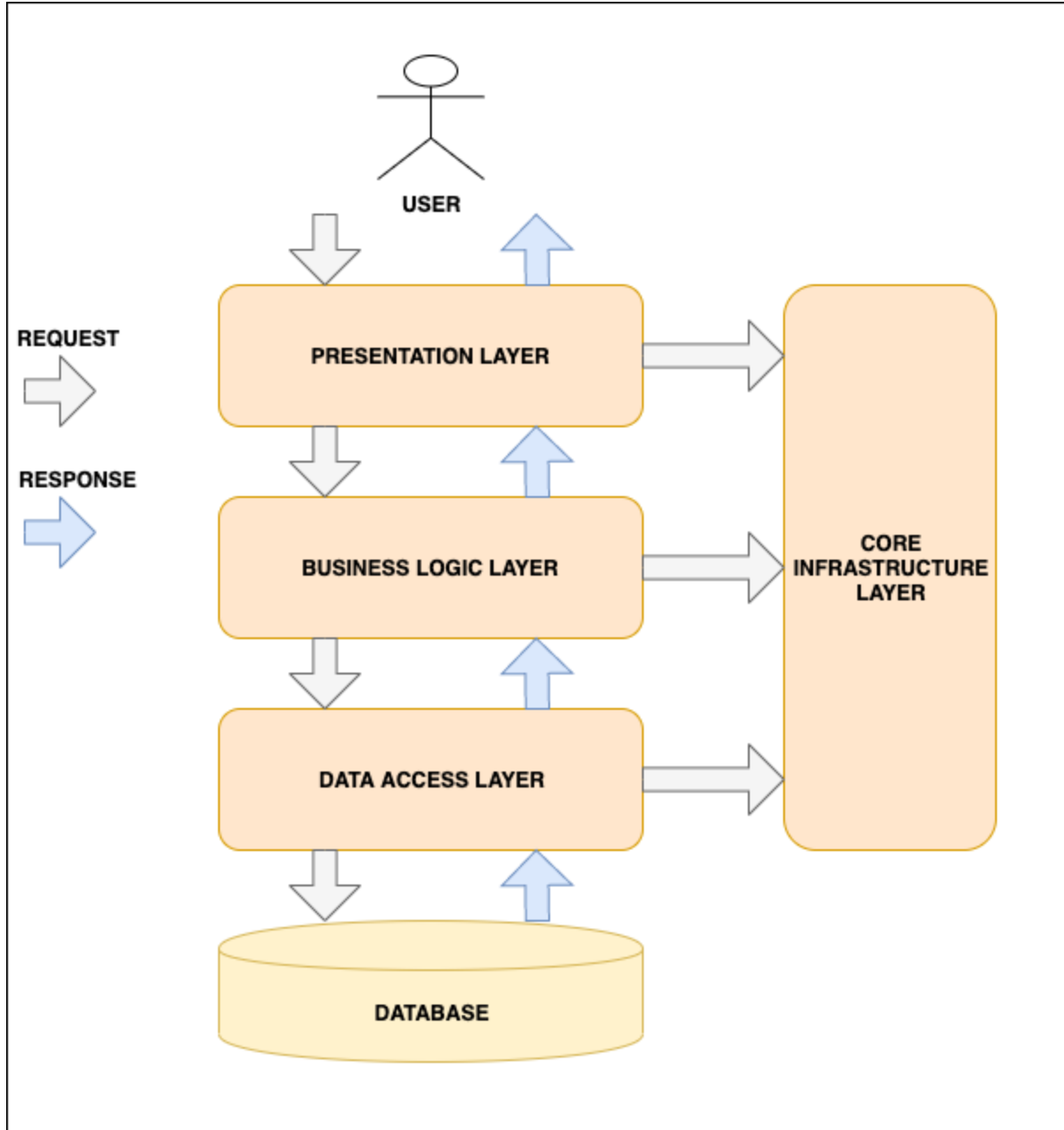


Yazılımda kod tekrarını önlemek amacıyla kodların katmanlara göre yazılması için katmanlar oluşturulur. Amacımız kendimizi tekrar etmememizdir.

Katmanlı mimarinin faydaları

- Kod tekrarını önler
- Projeye olan hakimiyet artar
- Clean code yapısı sağlar
- Hata yönetimi kolaylaşır
- Her katmanı ayrı ayrı yönetilebilir, diğer katmanları etkilemeden her katmana yenilikler ekleyebilir veya değiştirebiliriz.

Aslında 3 katman vardır ve diğerleri yardımcı katmanlardır.(Data Access Layer, Business Logic Layer, Presentation Layer)



N-Tier Architecture

* Data Access Layer

Genel olarak Veri Eriřim Katmanını (DAL) veritabanına yönelik bir ađ geçidi olarak algılayabiliriz . Bir uygulama, Sunum Katmanından (PL) girdi alır ve onu İş Mantığı Katmanında (BLL) işler. DAL'ın görevi anlambilimi ve yetkilendirmeyi kapsamlı bir şekilde doğrulamak ve verileri depolama alanından almaktır. Veri erişim ve CRUD operasyonlarının yapıldığı katmandır. Veri tabanı bağlantı kodları bulunur. Bu katmanda genellikle her veritabanına karşılık gelen veritabanı CRUD işlemlerini yapan kodlar vardır. Yani her

veritabanına karşılık gelen Repository classları mevcuttur. Dolayısıyla birden fazla repository bulunmasıyla beraber bir de repositoryleri tek elden yönetmeyi sağlayan UnitOfWork class'ı bulunur.

Crud operasyonları → Create, Read, Update, Delete

*** Business Logic Layer**

Validasyon işlemlerinin(Doğrulama) yapıldığı katmandır. İş Mantığı Katmanı (BLL) bir uygulamanın uygulama kısmını tutar . Ayrıca uygulamadaki işlemlerin işlenmesinden sorumludur. Ayrıca kullanıcı ile etkileşime yönelik sunum katmanı ile yazılımın ana gövdesi olarak görev yapar. Özellikle BLL, bir uygulamada yer alan gerekli hesaplamaları yapar. Ayrıca bu katman, kuralların uygulanmasından ve komutların veritabanına göre doğrulanmasından sorumludur. Ayrıca komutların doğrulanmasını sağlayarak güvenliği sağlar.

*** Presentation Layer**

PL, kullanıcı ile yazılım arasında bir arayüz görevi görür . Bu katman, çoğunlukla bir tür grafik kullanıcı arayüzü (GUI) aracılığıyla kullanıcıyla olan etkileşimlerden sorumludur. Bu katmanı kullanarak kullanıcıdan girdi alabilir ve sonucu görüntüleyebiliriz. Sunum yani UI işlemlerinin oluşturulduğu ve kullanıcıyla direkt olarak etkileşimde bulunan katmandır. Bu katman veri katmanının direkt olarak veri tabanı ile bağlantıda olması gibi direkt olarak kullanıcıyla bağlantıdadır.

*** Entity Layer(Entities)**

Varlık katmanı olarak geçmektedir. Proje boyunca kullanacağımız class(sınıf)'ları diğer bir deyişle somut nesnelerimizi oluşturduğumuz yerdir. Somut sınıfımız tekil isimli olmalıdır. Projede bulunan tablo ve sütunlar yer alır.

*** Core**

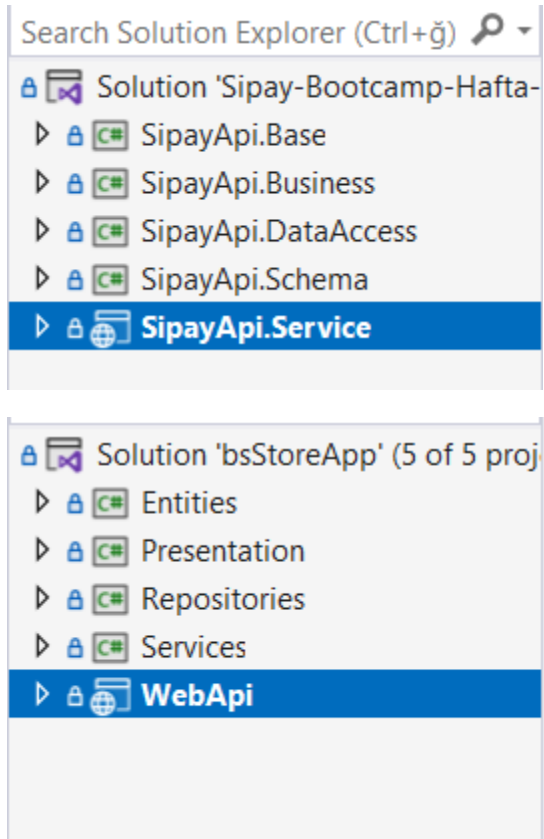
Core katmanın amacı ise proje boyunca sadece 1 kez yazacağımız ve her projede kullanabileceğimiz kodları barındırmaktır.

*** Service Layer(Servis Katmanı)**

Servis katmanı, uygulamalarımızı servis ettiğimiz katmandır. Bu katman üzerinden , business katmanı üzerinde yazdığımız fonksiyonellikleri web uygulaması, masaüstü uygulama, mobil uygulama gibi uygulamalara servis edebiliriz.

Katmanlı mimarideki katmanların yapısı ve oluşumu projenin ihtiyaçlarına göre belirlenir. İhtiyaçlar doğrultusunda proje katman sayısı artırılıp azaltılabilir.

Kendi projelerimden örnek N katmanlı mimari yapıları ve katmanlar



Projelerimi GitHubımdan inceleyebilirsiniz: <https://github.com/gulsunciftci>

Umarım faydalı bir kaynak olmuştur. Sonraki yazılarımda görüşmek üzere.

Mutlu kodlamalar 🚀