

N-Katmanlı Mimari

N- Katmanlı Mimari Nedir?

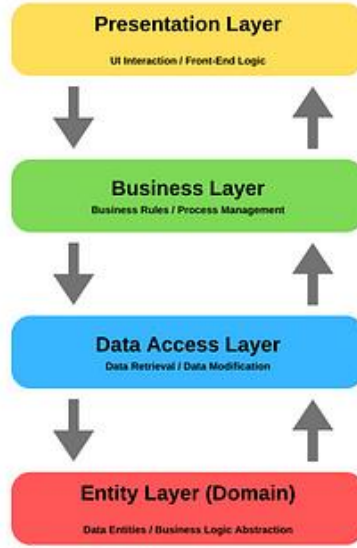
N-katmanlı mimari, bir yazılım uygulamasını farklı işlevsel katmanlara veya bileşenlere ayırarak geliştirmek için kullanılan bir yazılım tasarımı yaklaşımıdır. Bu bileşenler, asıl uygulamanın farklı işlevlerini ve sorumluluklarını taşırlar. Katmanlı mimari genellikle aşağıdaki katmanlardan oluşur.

Varlık Katmanı (Entity Layer) = Bu katman, veri tabanı tablolarına karşılık gelen sınıfları tutar ve bu özellikleri diğer katmanlara referans yoluyla iletir. Bu katmanda genellikle veri transfer objeleri (data transfer object), özelleştirilmiş hatalar için hata modelleri, log modelleri ve diğer gerekli modeller de bulunur.

Veri Erişim Katmanı (Data Access Layer) = Bu katman, uygulamada kullanılacak veri tabanına veya başka veri kaynaklarına erişim sağlayan katmandır. Bu katman veri tabanından veri okunması, silinmesi güncellenmesi veya oluşturulması (temel CRUD) için gerekli kodları içeren sınıfları bulundurur. Ayrıca veri tabanı bağlantısı için gerekli context sınıfı da bu katmanda bulunur.

İş Mantığı Katmanı (Business Logic Layer) = Uygulamanın asıl iş mantığının gerekli sınıflar ve metotlar ile kodlandığı katmandır. Burada veri tabanından verilerin hangi şartlar ile getirileceği, ekleme ve güncelleme için veri doğruluğu (validasyon), gerekli 3.parti yazılımlar için import işlemleri yapılır. Ve uygulamanın ana hatları çıkarılır.

Sunum Katmanı (Presentation Layer) = Kullanıcı arayüzünün bulunduğu katmandır. Kullanıcılar uygulamaya çeşitli protokollerle isteklerde bulunur. Ve uygulamanın yeteneklerine ve sınırlarına göre geri dönüş alır. Kısacası bu katman bir web sayfası, bir mobil uygulama gibi uygulamaların arayüzü olarak düşünülebilir.



N-katmanlı mimaride bir isteğin yaşam döngüsü şu şekildedir:

Kullanıcı, bir web tarayıcısı, masaüstü uygulama veya mobil uygulama gibi bir istemci tarafından sunum katmanında bir istek oluşturur. Bu istek, kullanıcı tarafından tetiklenir. Sunum katmanı, kullanıcının isteğini alır ve bu isteği işlemeye başlar. İşte bu katmanda kullanıcı arayüzü oluşturulur ve kullanıcıdan gelen veriler alınır. Sunum katmanı, isteği belirli bir iş mantığı işlemi veya veri erişim işlemi için iş mantığı katmanına (service layer) iletmek üzere hazırlar. İş mantığı katmanı, gelen isteği işler. Bu katmanda iş süreçleri, veri doğrulama, iş kuralları ve veri manipülasyonu gibi işlemler gerçekleştirilir. İş mantığı katmanı, gerekirse veri erişim katmanına (entity katmanı) veri sorguları gönderir veya veri erişim işlemlerini yürütür ve veri tabanı veya diğer veri kaynaklarına erişim gerektiren istekleri veri erişim katmanına iletir. Bu istekler, genellikle veri okuma, yazma, güncelleme veya silme işlemlerini içerir. Veri erişim katmanı, gelen veri erişim isteğini işler. Bu katman, veri kaynaklarına erişir, veri tabanı sorgularını yürütür ve verileri istenen istek doğrultusunda geri döndürür. Veri erişim katmanı veya iş mantığı katmanı, sonuçları sunum katmanına ileterek isteğin sonucunu oluşturur. Sunum katmanı, bu sonuçları kullanıcıya sunmak için kullanıcı arayüzünü günceller. Bu sonuç gelen isteğin uygunluğuna göre başarılı veya hatalı olabilir. Temelde sunum katmanı isteği başlatır, iş mantığı katmanı işler ve veri erişim katmanı veri işlemlerini yönetir.

N katmanlı mimari, uygulamanın farklı katmanları arasında açık ve tanımlanmış sınırların olması nedeniyle aşağıdaki avantajları sağlar:

Modülerlik = Her katman, belirli bir işlevi yerine getirir, bu da uygulamanın daha kolay yönetilmesini ve bakımını sağlar. Ayrıca, farklı katmanlar bağımsız olarak geliştirilebilir, test edilebilir ve güncellenebilir.

Ölçeklenebilirlik = Her katman, ihtiyaç durumunda ayrı ayrı ölçeklendirilebilir. Bu, yüksek trafikli uygulamalarda performans sorunlarını azaltmaya yardımcı olur.

Güvenlik = Farklı katmanların ayrılması, güvenlik politikalarının uygulanmasını kolaylaştırır. Örneğin, veri erişim katmanı, verilere erişimi sınırlayabilir ve iş mantığı katmanı, iş kurallarını uygulayabilir.

Taşınabilirlik = Farklı kullanıcı arayüzleri, aynı iş mantığı ve veri erişim katmanlarını kullanabilir. Bu, uygulamanın farklı platformlarda çalıştırılmasını kolaylaştırır.

Bakım Kolaylığı = Her katmanın belirli bir sorumluluğu olduğu için, değişikliklerin diğer katmanları olumsuz etkileme olasılığı düşüktür. Bu da yazılım bakımını kolaylaştırır.

N-katmanlı mimari, büyük ölçekli ve karmaşık yazılım uygulamalarının geliştirilmesi için sıklıkla tercih edilen bir yaklaşımdır. Ancak uygulamanın gereksinimlerine ve boyutuna göre katman sayısı ve karmaşıklık değişebilir.