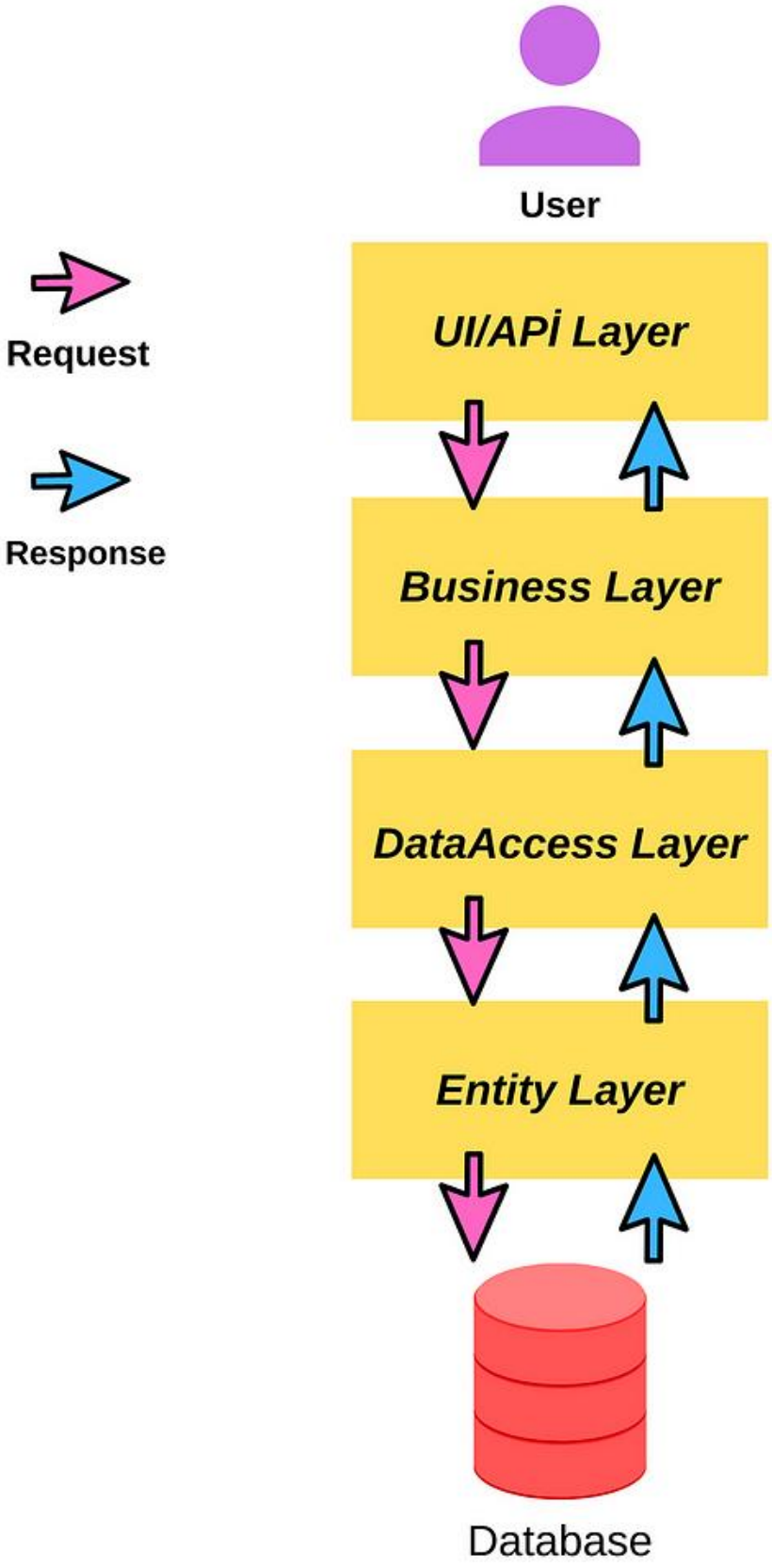
Backend Katmanlı Mimari: Modüler ve Ölçeklenebilir Uygulamalar İçin Tasarım

Backend **katmanlı mimari**, bir yazılım uygulamasını bir dizi mantıksal katmana bölen bir tasarım desenidir. Her katman, belirli bir işlevi yerine getiren modüllerden oluşur ve diğer katmanlarla iletişim kurar. Bu tasarım, uygulamanın daha organize edilmiş, yönetilebilir ve bakımı daha kolay hale getirmeyi amaçlar. Ayrıca, yeni özelliklerin eklenmesini ve mevcut işlevlerin değiştirilmesini daha az etkileyecek şekilde uygulamanın geliştirilmesine olanak tanır.


User
Request
Response
UI/APİ Layer
Business Layer
DataAccess Layer
Entity Layer
Database

Backend Katmanlı Mimari Katmanları

Genellikle üç ana katmanlı mimari kullanılır, ancak bazı durumlarda daha fazla katman da eklemek mümkündür:

1. **Sunucu Katmanı (Presentation Layer):** Bu katman, kullanıcı arayüzünü (UI) ve istemci tarafını temsil eder. Kullanıcıların uygulamayla etkileşime geçtiği kısım burasıdır. Web uygulamaları için, bu katman kullanıcıların tarayıcılarından erişebileceği ve görsel olarak etkileşime geçebileceği katmandır. Kullanıcı girişleri, sunucu katmanına iletilir ve gerekli işlemleri başlatmak için diğer katmanlara yönlendirilir.
2. **İş Mantığı Katmanı (Business Logic Layer):** Bu katman, uygulamanın temel iş mantığından sorumludur. Burada, işlemlerin gerçekleştirildiği ve verilerin işlendiği kodlar bulunur. Veritabanı işlemleri, hesaplamalar, doğrulamalar, yetkilendirmeler gibi işlemler bu katmanda gerçekleştirilir. İş mantığı katmanı, sunucu katmanından gelen istekleri alır, gerekli işlemleri yapar ve sonuçları tekrar sunucu katmanına gönderir.
3. **Veri Erişim Katmanı (Data Access Layer):** Bu katman, veritabanıyla doğrudan etkileşimden sorumludur. Veri erişim katmanı, iş mantığı katmanından gelen talepleri alır, veritabanına erişim sağlar ve veritabanı işlemlerini gerçekleştirir. Bu katman veri tabanı bağlantıları, sorgular ve veri işleme işlevleri içerir. Bu sayede iş mantığı katmanı, verilerle nasıl iletişim kuracağını bilmeden verilere erişebilir.

Backend Katmanlı Mimari Avantajları

- **Modülerlik:** Her katman birbirinden bağımsız çalışır, bu sayede herhangi bir katmanın değiştirilmesi, güncellenmesi veya yeniden yapılandırılması, diğer katmanları etkilemeden gerçekleştirilebilir.
- **Ölçeklenebilirlik:** Uygulama, katmanların tek tek ölçeklendirilmesi sayesinde ihtiyaca göre daha kolay bir şekilde genişletilebilir ve yüksek trafikle başa çıkabilir.
- **Bakım Kolaylığı:** Her katmanın belirli bir sorumluluğu olduğu için, hataların izlenmesi, bakımı ve sorun giderme işlemleri daha kolay ve hızlı gerçekleştirilebilir.
- **Kodun Tekrar Kullanılabilirliği:** Modüler yapı, kodun yeniden kullanılabilirliğini artırır ve yeni özelliklerin eklenmesini daha hızlı hale getirir.
- **Paralel Geliştirme:** Farklı katmanlardaki ekipler aynı anda çalışabilir ve ilgili katmanlarda değişiklik yapmak, diğer katmanları etkilemez.

Backend Katmanlı Mimari Örneği: E-Ticaret Uygulaması

Bir örnek üzerinde açıklayalım.***E-Ticaret Uygulaması***, Bu uygulama, kullanıcılara ürünleri görüntüleme, sepete ekleme ve sipariş oluşturma gibi temel işlevleri sunacaktır.

1. Sunucu Katmanı:

- Sunucu katmanı, istemci tarafından gelen **HTTP** isteklerini karşılayacak ve iş mantığı katmanına yönlendirecek.
- Kullanıcıların web tarayıcılarından uygulamaya erişebileceği **API endpoint**'leri sağlayacak.
- Bu katmanda gelen isteklerin doğruluğu kontrol edilecek ve gerekli kimlik doğrulama işlemleri gerçekleştirilecektir.

Örnek API endpoint'ler:

1. **/products:** Tüm ürünleri listelemek için GET isteğini işleyecek.
2. **/cart:** Kullanıcının sepetini yönetmek için GET, POST ve DELETE isteklerini işleyecek.
3. **/orders:** Kullanıcının siparişlerini yönetmek için GET ve POST isteklerini işleyecek.

2. İş Mantığı Katmanı:

- İş mantığı katmanı, gelen API isteklerini alacak, gerekli işlemleri yapacak ve veri erişim katmanına yönlendirecek.
- Bu katmanda, ürünlerin veritabanından çekilmesi, sepet işlemleri ve sipariş oluşturma gibi temel işlemler gerçekleştirilecektir.

Örnek işlevler:

1. **getProducts():** Tüm ürünleri veritabanından çekmek için kullanılır.
2. **addToCart(userId, productId):** Kullanıcının sepetine ürün eklemek için kullanılır.
3. **removeFromCart(userId, productId):** Kullanıcının sepetinden ürün çıkarmak için kullanılır.
4. **createOrder(userId, cartItems):** Kullanıcının sepetindeki ürünlerle yeni bir sipariş oluşturur.

Tabii, işte basit bir e-ticaret uygulaması üzerinden detaylı bir backend katmanlı mimari örneği:

Backend Katmanlı Mimari Örneği: E-Ticaret Uygulaması

Bu örnek, üç katmanlı bir mimari kullanılarak tasarlanmış basit bir e-ticaret uygulamasını açıklamaktadır. Bu uygulama, kullanıcılara ürünleri görüntüleme, sepete ekleme ve sipariş oluşturma gibi temel işlevleri sunacaktır.

1. Sunucu Katmanı:

- Sunucu katmanı, istemci tarafından gelen HTTP isteklerini karşılayacak ve iş mantığı katmanına yönlendirecek.
- Kullanıcıların web tarayıcılarından uygulamaya erişebileceği API endpoint'leri sağlayacak.
- Bu katmanda gelen isteklerin doğruluğu kontrol edilecek ve gerekli kimlik doğrulama işlemleri gerçekleştirilecektir.

Örnek API endpoint'ler:

1. /products: Tüm ürünleri listelemek için GET isteğini işleyecek.
2. /cart: Kullanıcının sepetini yönetmek için GET, POST ve DELETE isteklerini işleyecek.
3. /orders: Kullanıcının siparişlerini yönetmek için GET ve POST isteklerini işleyecek.

2. İş Mantığı Katmanı:

- İş mantığı katmanı, gelen API isteklerini alacak, gerekli işlemleri yapacak ve veri erişim katmanına yönlendirecek.
- Bu katmanda, ürünlerin veritabanından çekilmesi, sepet işlemleri ve sipariş oluşturma gibi temel işlemler gerçekleştirilecektir.

Örnek işlevler:

1. getProducts(): Tüm ürünleri veritabanından çekmek için kullanılır.
2. addToCart(userId, productId): Kullanıcının sepetine ürün eklemek için kullanılır.
3. removeFromCart(userId, productId): Kullanıcının sepetinden ürün çıkarmak için kullanılır.
4. createOrder(userId, cartItems): Kullanıcının sepetindeki ürünlerle yeni bir sipariş oluşturur.

3. Veri Erişim Katmanı:

- Veri erişim katmanı, iş mantığı katmanının taleplerini alacak ve veritabanıyla doğrudan iletişim kuracaktır.

- Bu katmanda, ürünlerin, sepetin ve siparişlerin veritabanında depolanması ve yönetilmesi sağlanacaktır.

Örnek işlevler:

1. **fetchProductsFromDatabase():** Tüm ürünleri veritabanından çekmek için kullanılır.
2. **saveCartItemToDatabase(userId, productId):** Kullanıcının sepetine ürün eklemek için kullanılır.
3. **deleteCartItemFromDatabase(userId, productId):** Kullanıcının sepetinden ürün çıkarmak için kullanılır.
4. **saveOrderToDatabase(userId, orderDetails):** Kullanıcının siparişini veritabanına kaydetmek için kullanılır.

Bu katmanlı mimari, uygulamanın farklı katmanlarda modüler bir şekilde tasarlanmasına ve geliştirilmesine olanak tanır. Örneğin, veritabanı değiştirilmek istendiğinde sadece ***veri erişim katmanı değiştirilirken, API endpoint'leri ve iş mantığı katmanı etkilenmez. Aynı şekilde, yeni özellikler eklemek istendiğinde de sadece ilgili katman üzerinde çalışmak mümkündür. Bu da kodun yeniden kullanılabilirliğini artırır ve daha iyi bakım yapılmasını sağlar.***