



# DIYAGRAM VE MİMARİ

MERYEM NUR CEYLAN  
OZAN ÇAKIR  
BERAT KUZU  
MUHAMMED FURKAN ÖLÇER

# İÇİNDEKİLER

- 1 Önsöz
- 1 Giriş
- 2 Sözlük
- 2 Kullanım Durum Diyagramı
- 5 Sınıf Diyagramı
- 7 Uygulama-Sistem Mimarisi

# Önsöz

---

Oluşturmuş olduğumuz diyagram ve mimari projemizin olmasını planladığımız halidir. Bu diyagram ve mimari projemizin kod kısmının uygulanışını daha basite indirgeyecektir.

# Giriş

---

Diyagramlar, bir fikrin veya sürecin görsel temsili için kullanılan grafiklerdir. Adım adım talimatlar sağlayarak ve görsel rehberlik sunarak, işin daha kolay ve anlaşılır bir şekilde yapılmasına yardımcı olurlar. Mimarinin tanımını her ne kadar diyagramdan farklı da olsa mimari tasarım aşamasında diyagramlar kullanılabilir.

# Sözlük

---

Diyagram: Diyagram, görsel olarak bir sistemin veya bir problemin açıklamasını yapmak için kullanılan bir grafik türüdür.

Extend ve Include: Extend ve include, UML (Unified Modeling Language) diyagramlarında kullanılan iki farklı davranış modelleridir. Include, bir davranış modelinin diğer bir davranış modelini zorunlu olarak içerdiği durumlarda kullanılır. Extend, bir davranış modelinin (use case) diğer bir davranış modelini opsiyonel olarak genişlettiği durumlarda kullanılır.

## Kullanım Durum Diyagramı

---

Kullanım Durum Diyagramı (Use Case Diagram), bir yazılım sisteminin davranışını modellemek ve kullanım senaryolarını göstermek için kullanılan bir UML (Unified Modeling Language - Birleştirilmiş Modelleme Dili) diyagram türüdür.

Kullanım Durum Diyagramları, bir yazılım sisteminin kullanıcılarının, sistemle nasıl etkileşimde bulunacaklarını gösteren senaryolara dayanarak oluşturulur.

# Kullanım Durum Diyagramı

Bu diyagramlar, sistemin işlevselliğini anlamak, gereksinimleri belirlemek ve analiz etmek için kullanılır.

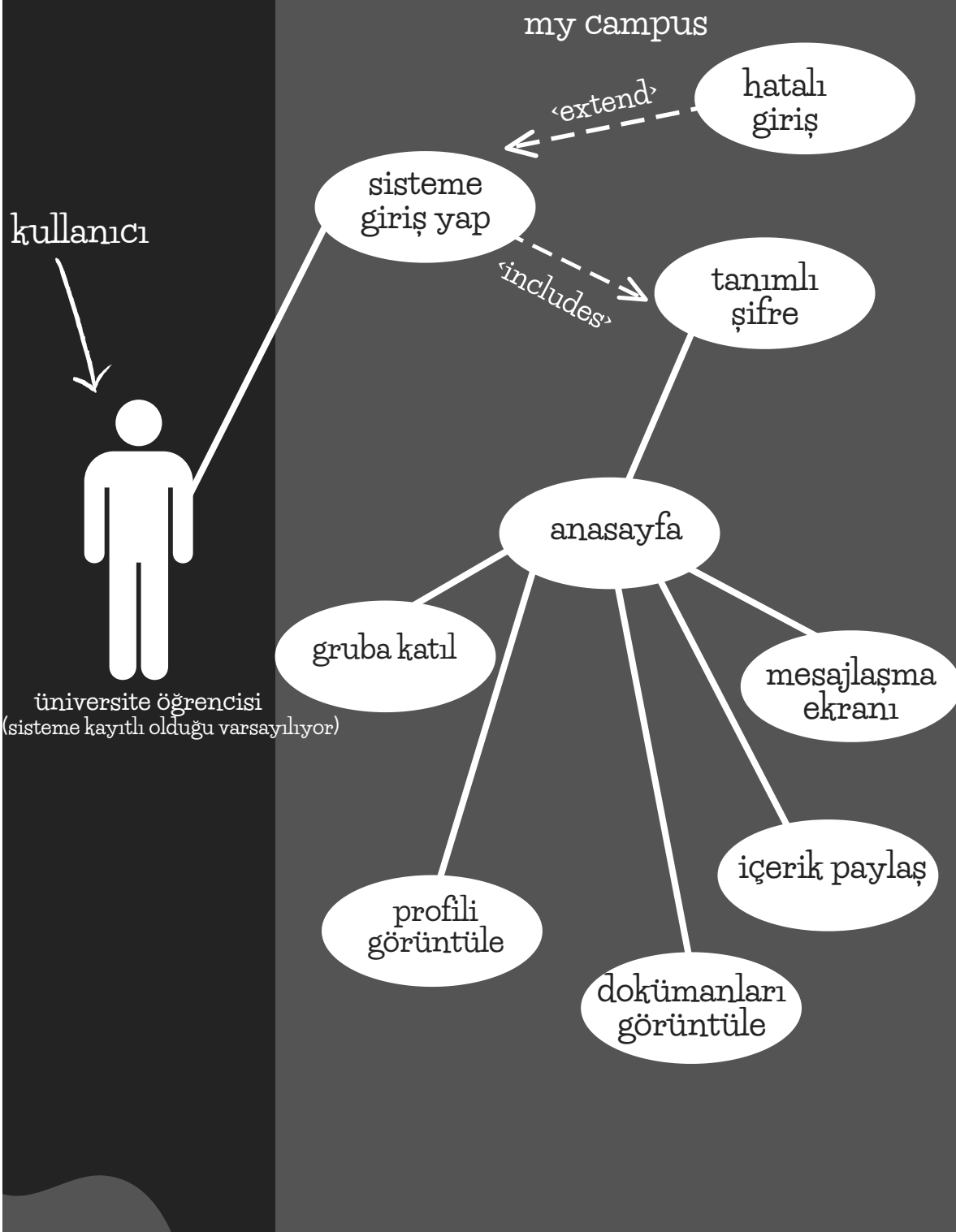
Kullanım durumları, kullanıcıların ihtiyaçlarına göre belirlenir ve her kullanım durumu, kullanıcının bir hedefine ulaşmak için sisteme hangi adımları takip edeceğini gösterir.

Kullanım Durum Diyagramları, bir sistemdeki kullanıcıların, işlevleri ve ilişkileri gösterir. Bu diyagramlar, sistem gereksinimlerini belirlemede, kullanıcıların ihtiyaçlarını analiz etmede, tasarımın iletilmesinde ve test senaryolarının oluşturulmasında yardımcı olabilir.

Kullanım Durum Diyagramları, yazılım geliştirme sürecinde kullanılan en yaygın ve basit UML diyagramlarından biridir.

Peki tanımını yaptığımız kullanım durum diyagramı projemizde nasıl olur bunu ifade edelim.

# Kullanım Durum Diyagramı



# Sınıf Diyagramı

---

Sınıf diyagramı, bir yazılım sisteminin sınıflarını ve bunların birbirleriyle olan ilişkilerini görsel olarak tanımlayan bir UML (Unified Modeling Language) diyagramıdır.

Sınıf diyagramları, bir yazılım sisteminin tasarımını göstermek için kullanılan en yaygın UML diyagramlarından biridir.

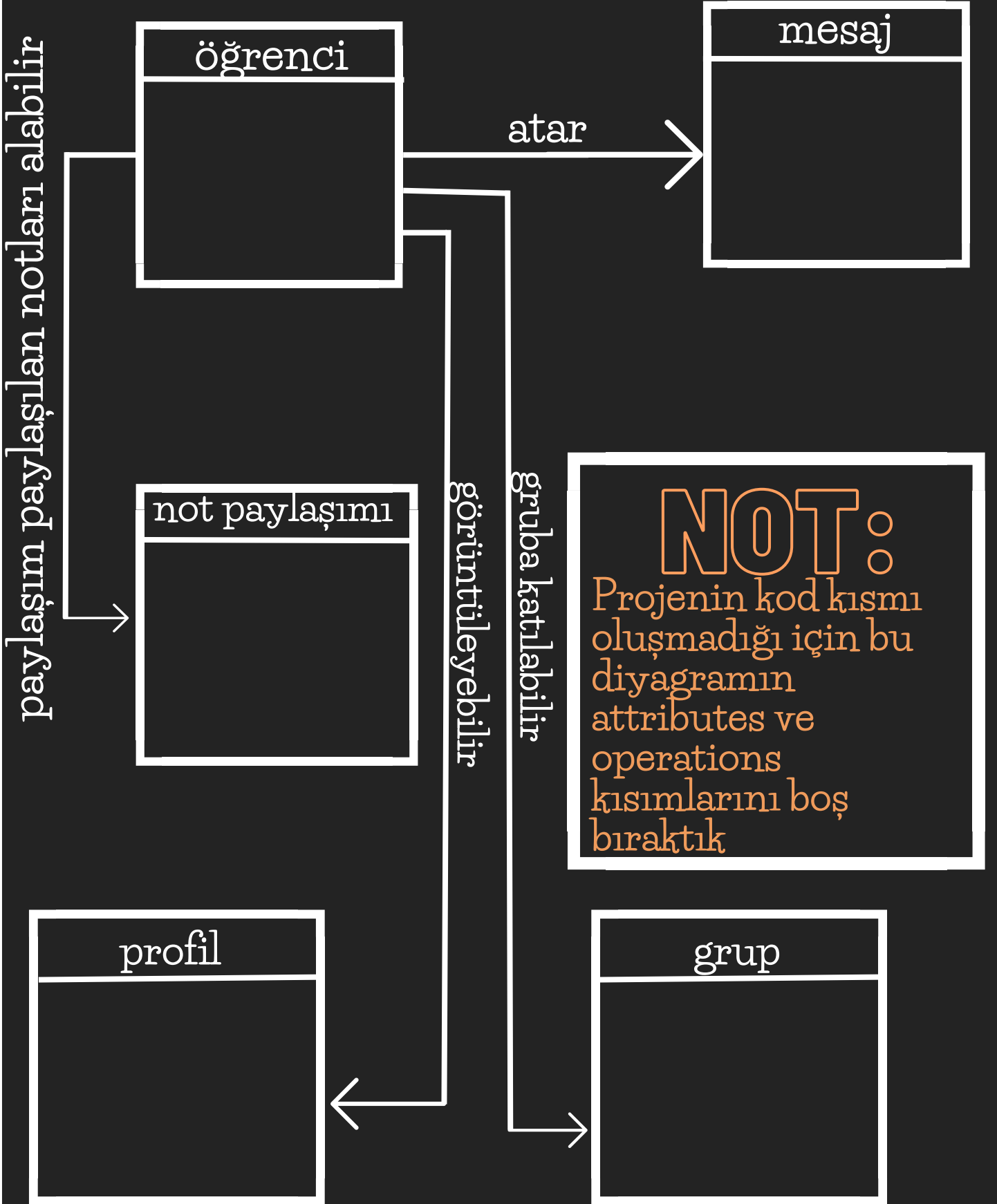
Bir sınıf diyagramı, genellikle dikdörtgenlerle gösterilen sınıflardan oluşur. Bu sınıflar, sistemde yer alan nesnelerin özelliklerini (veri elemanları) ve davranışlarını (fonksiyonlar ve metotlar) içerirler.

Sınıflar arasındaki ilişkiler ise oklarla gösterilir ve bu ilişkiler, sınıflar arasındaki bağımlılıkları, miras ilişkilerini, arayüzleri ve çoklu ilişkileri ifade eder.

Sınıf diyagramları, bir yazılım sistemi tasarımını anlamak ve iletişim kurmak için önemli bir araçtır.

Yazılım geliştiricileri, sınıf diyagramları kullanarak bir sistemdeki nesnelerin davranışlarını ve aralarındaki ilişkileri kolayca anlayabilirler.

# Sınıf Diyagramı





# Sistem Mimarisi

Uygulama mimarisi, bir yazılım sisteminin yapısını, bileşenlerini ve bunların birbirleriyle olan ilişkilerini tanımlayan bir terimdir.

Bir yazılım sistemi tasarlanırken, uygulama mimarisi, sistemin tüm parçalarını ve bu parçalar arasındaki ilişkileri açıkça tanımlamak için kullanılır.

Uygulama mimarisi, bir yazılım sistemi için bir plan veya çerçeve olarak düşünülebilir. Bu plan, uygulamanın temel bileşenlerinin (modüller, sınıflar, arayüzler vb.) ve bu bileşenlerin nasıl bir araya getirileceğinin belirlenmesiyle oluşturulur.

Uygulama mimarisi, bir yazılım sisteminin ölçeklenebilirliğini, performansını, bakım kolaylığını ve genişletilebilirliğini etkiler.

Uygulama mimarisi, genellikle bir UML (Unified Modeling Language) diyagramı olarak ifade edilir. Bu diyagram, sistemi oluşturan bileşenlerin, bu bileşenlerin arasındaki ilişkilerin ve veri akışlarının görsel olarak tanımlanmasına olanak tanır. Ayrıca, uygulama mimarisi, bir yazılım sistemi tasarlanırken gereksinimlerin, kullanılacak teknolojilerin ve donanım altyapısının da dikkate alınmasını sağlar.

# Sistem Mimarisi

Uygulamanın hangi kategoriye dahil olacağını belirlemek için uygulamada var olan mesajlaşma kısmını esas alalım.

Mesajlaşma uygulaması, bir yazılım sistemi olduğu için genellikle uygulama yazılımı (application software) kategorisine bağlıdır.

Ancak mesajlaşma uygulaması için gerekli olan altyapılar (sunucular, veritabanları, ağ bileşenleri vb.) dikkate alındığında, sistem mimarisi açısından bakıldığında birden fazla kategoriye dahil edilebilir. Örneğin, mesajlaşma uygulaması için gereken sunucular ve veritabanları, sunucu yazılımı (server software) kategorisine dahil edilebilir. Ayrıca, mesajlaşma uygulaması, mobil cihazlar veya bilgisayarlar gibi farklı platformlar için geliştirildiğinde, bu platformlara özgü uygulama yazılımı veya donanım bileşenleri de kullanılabilir. Bununla birlikte, mesajlaşma uygulaması, genellikle bir istemci-sunucu (client-server) mimarisi kullanır. Bu mimaride, mesajlaşma uygulamasının istemci kısmı, kullanıcının cihazında çalışır ve sunucu kısmı, uygulamanın arkasında çalışır. İstemci ve sunucu arasındaki etkileşimler, önceden belirlenmiş protokoller ve API'ler aracılığıyla gerçekleştirilir. Bu nedenle, mesajlaşma uygulaması, istemci-sunucu mimarisi kategorisine de dahil edilebilir.

İstemci-sunucu mimarisi, genellikle iki katlı bir mimaridir.



# Sistem Mimarisi

