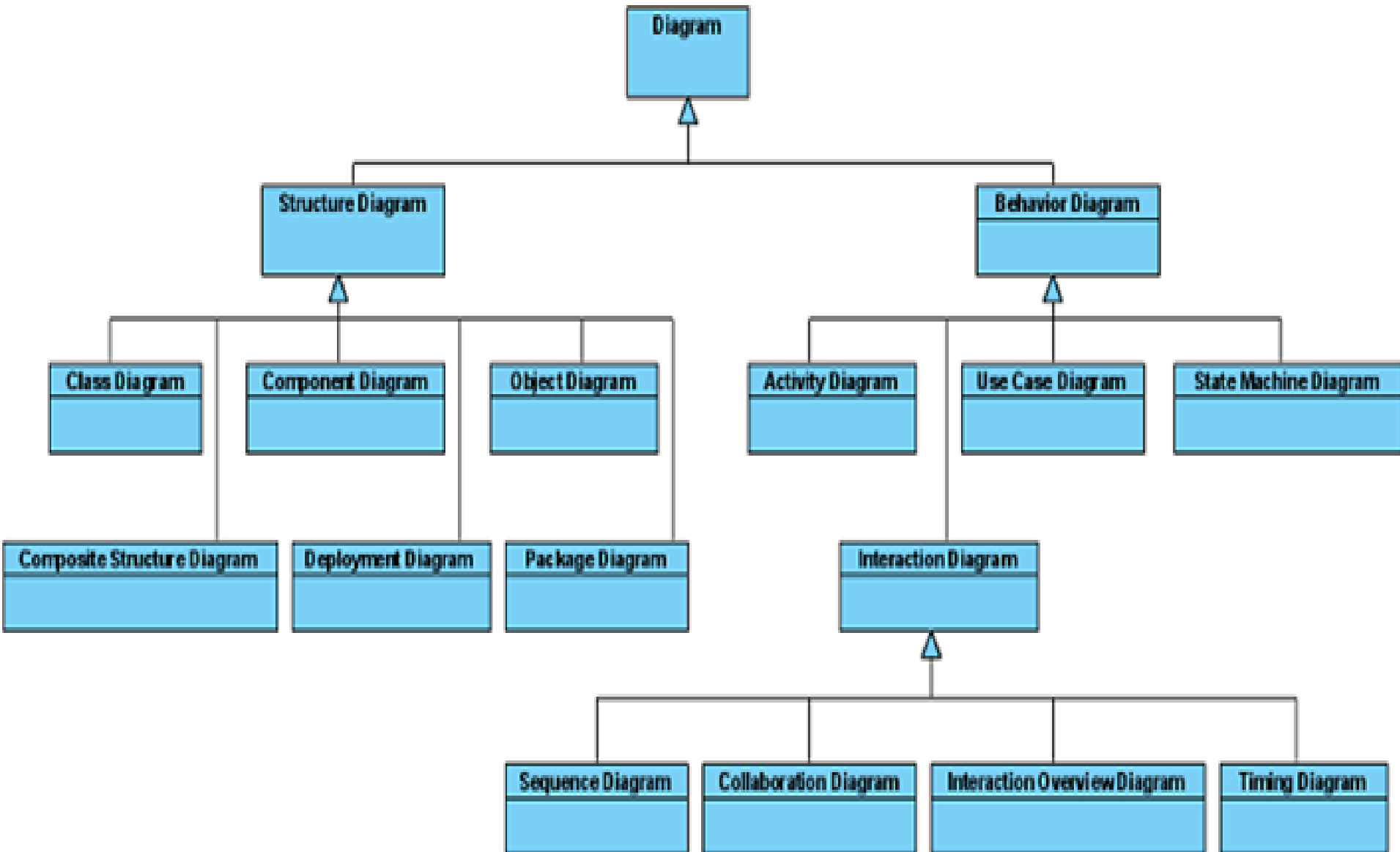


Yazılım Gereksinimlerinin Görsel  
Çözümlenmeleri:

UML

(Unified Modeling Language)  
Birleştirilmiş Modelleme Dili

# UML Diyagramlarının Sınıflandırması



# UML ile Dinamik Davranışsal (Behaviour) Modelleme use case diyagramları

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Visual Paradigm Community Edition ücretsizdir ve öğrenci epostaları ile kayıt olmak koşulu ile indirilebilmektedir.

# Use case

- **use case**, bir sistemin ya da sistemin bir parçasının davranışını betimler. Bu bağlamda değişimleri içeren bir dizi eylemin (action) tanımlaması yapılır. Böylece sistem herhangi bir aktöre (actor) ait olan gözlemlenebilir bir sonucun değerini gösterir.

The UML User Guide (Booch, 1999)

- **Aktör** , bir sistem, alt sistem veya sınıfla etkileşime giren dışsal (external) bir kişi, bir süreç veya şeydir. Bir aktör , dışsal kullanıcıların sistemle olan etkileşimlerini karakterize eder.

UML Reference Manual (1999, Rumbaugh)

# use case

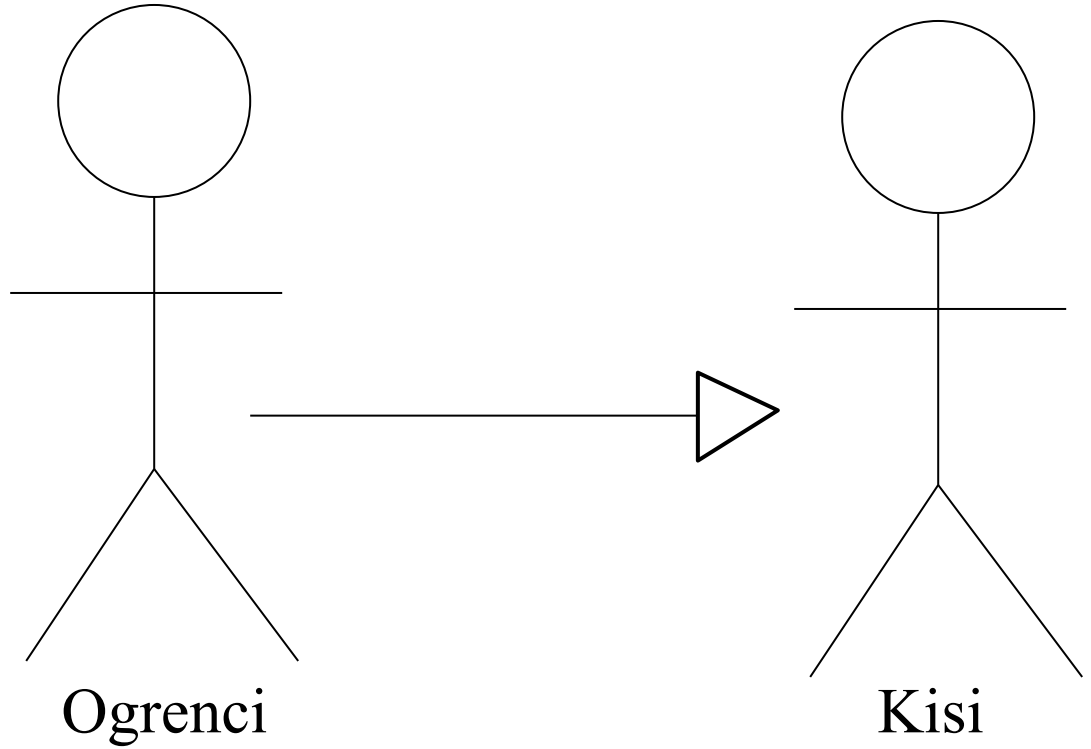
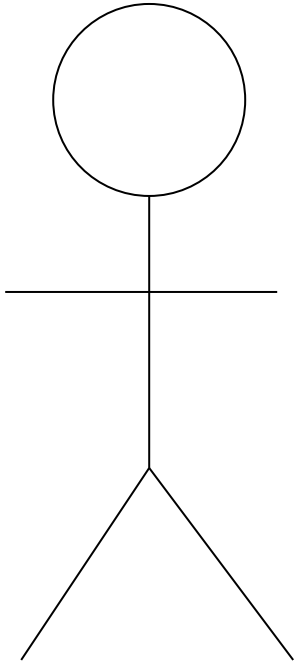
- ❑ Bir «use case», use case diyagramında elips ile gösterilir.
- ❑ Bir «use case» her zaman ismi ile etiketlenir.



# Aktör

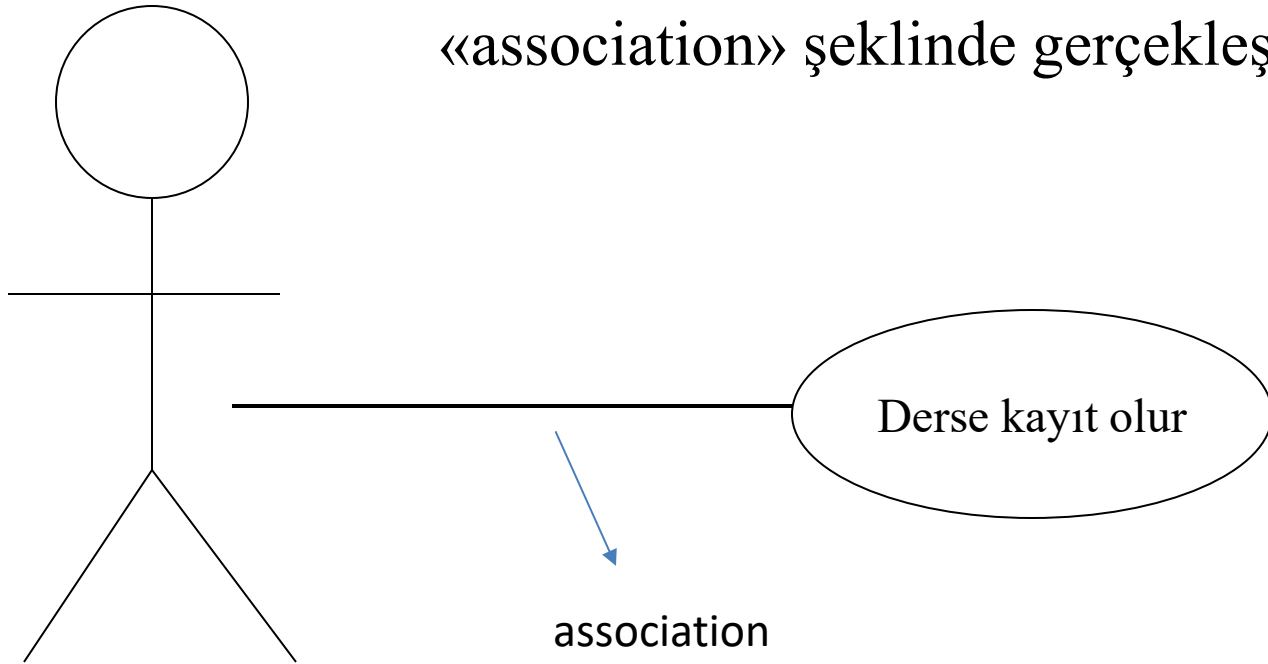
Her aktör bir veya daha fazla «use case» ile ilişkili olabilir.

Bir aktörün , başka bir aktör ile genelleştirme (generalization ) ilişkisi olabilir.



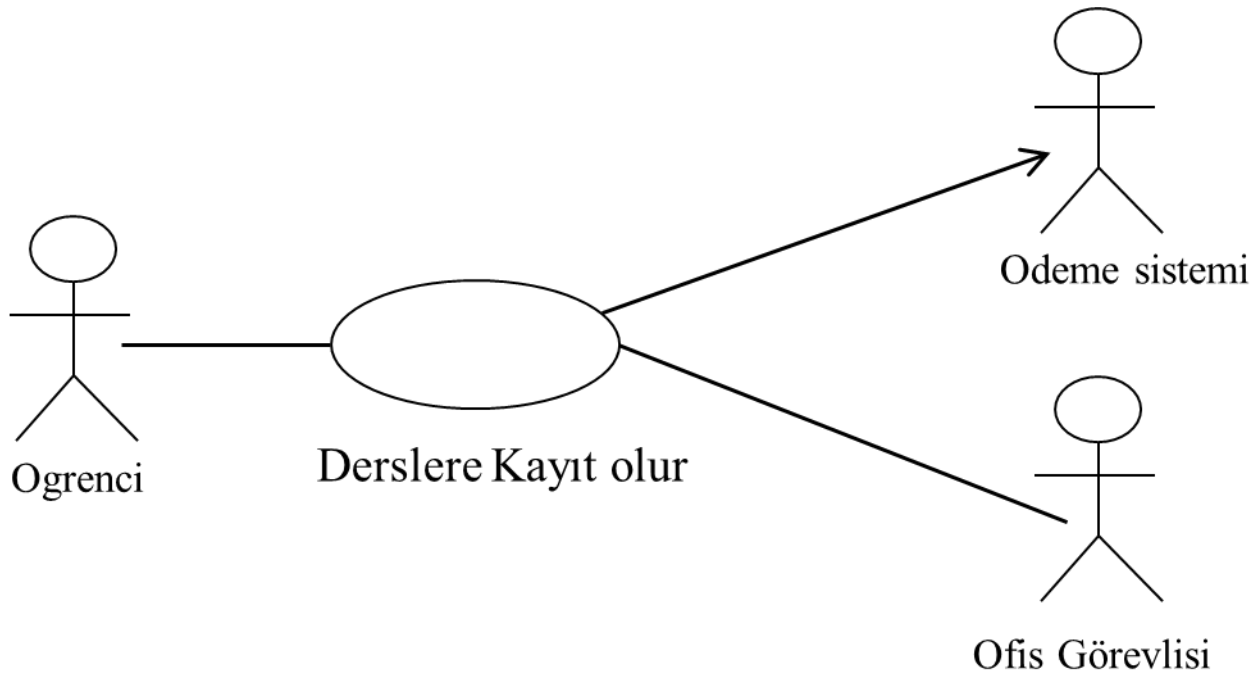
# Aktör

Aktörler ile use case bağlantısı  
«association» şeklinde gerçekleşir.

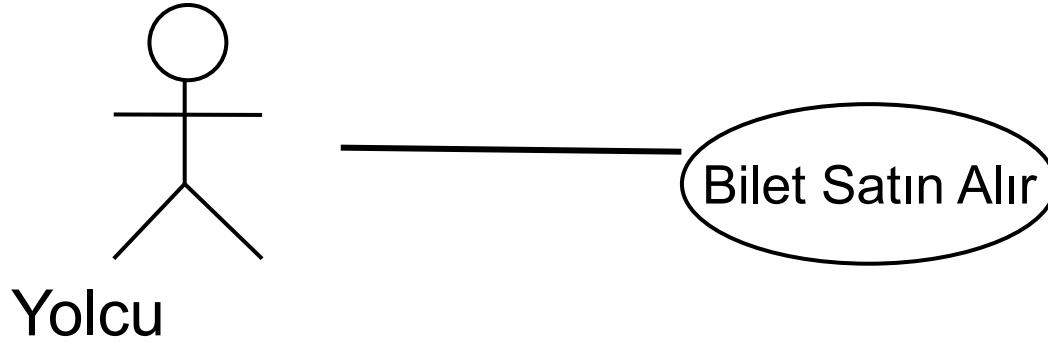


# aktör & use case =use case diyagramı

Öğrenci, «Derslere Kayıt Olunması» use case aracılığı ile Ofis Görevlisi ve Ödeme Sistemi ile etkileşimdedir.



# use case Diyagramları



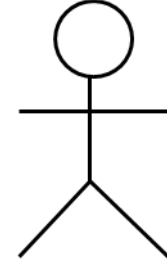
- Gereksinimlerin anlaşılması-anlamlandırılması (requirements elicitation) sırasında harici (external) davranışı simgelemek üzere kullanılır.
  - Use case modeli tüm «use case» ler kümesidir.
  - Sistemin ve çevresinin fonksiyonelliğinin betimlemesidir.
- Buna göre:
- Her bir «use case» belli bir fonksiyonelliğe ait bir dizi etkileşimi betimler ve
  - Aktörler sistemdeki bir tip kullanıcının rollerini betimler

# Aktör ve use Case

- ❑ Aktör olarak tanımlanan sistemle etkileşimli olan dışsal varlık (external entity) kullanıcı, dışsal sistem ya da fiziksel ortam olabilir.

Örnek olarak: Yolcu, GPS uydusu

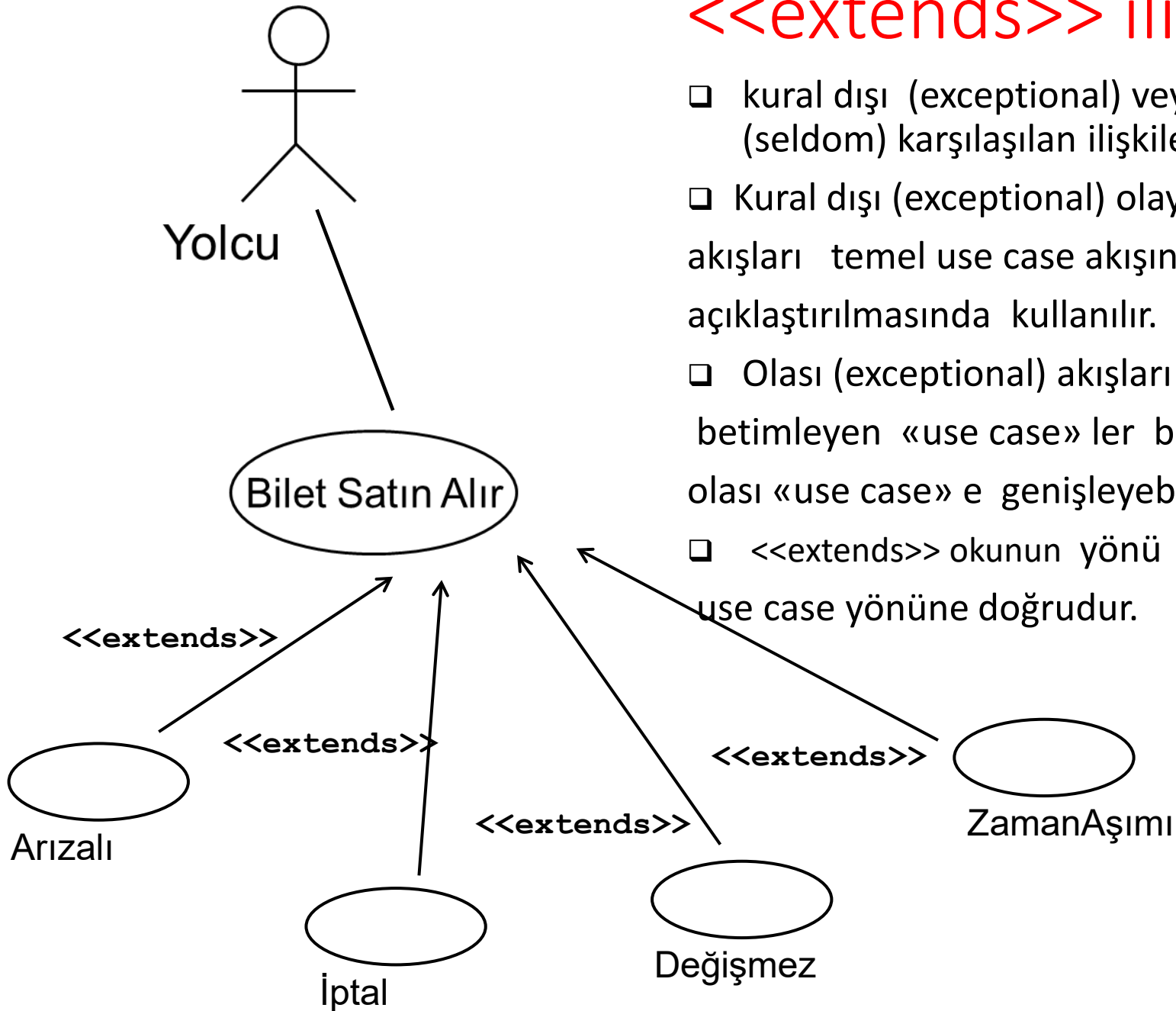
- ❑ Her «use case» tek bir isim ile tanımlanır, aktörlerle etkileşir, giriş koşulları ve çıkış koşulları içerir, olayların akışını ve özel gereksinimler



Yolcu

Bilet Satın Alır

# <<extends>> ilişkisi



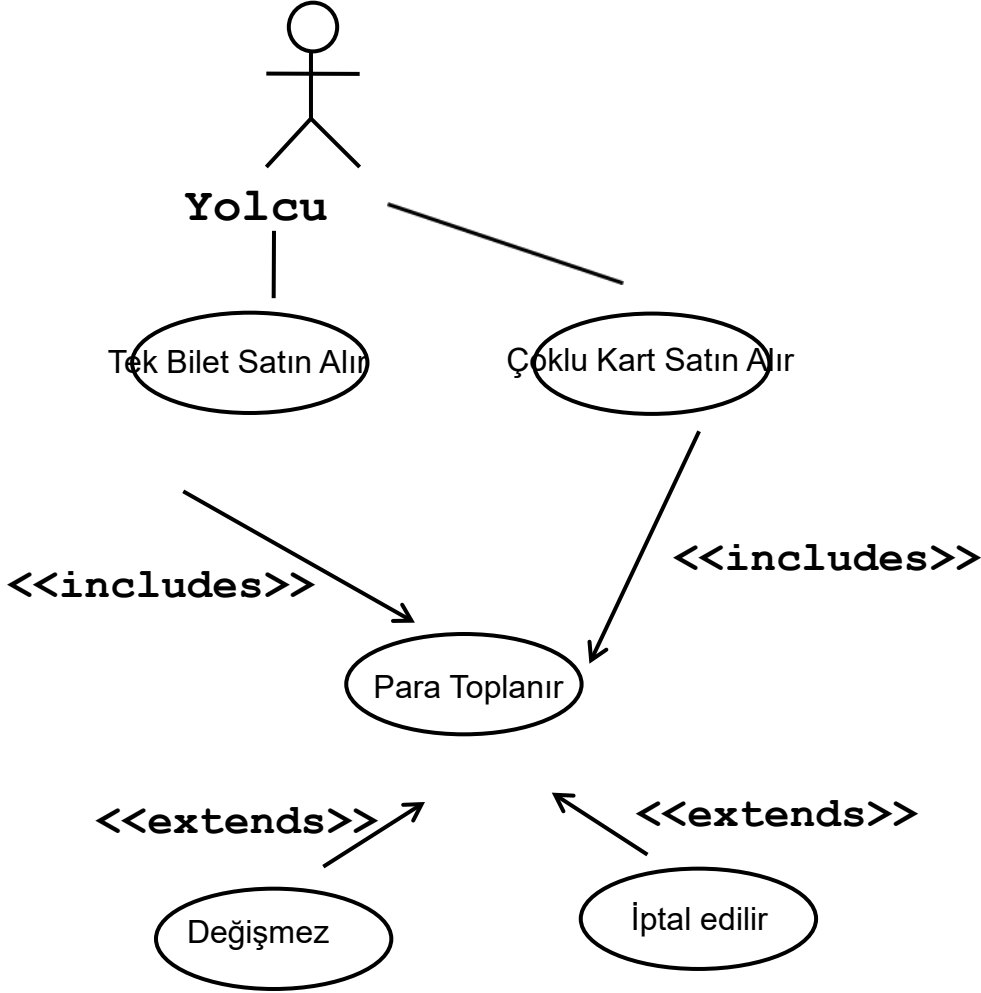
❑ kural dışı (exceptional) veya nadir (seldom) karşılaşılan ilişkileri gösterir.

❑ Kural dışı (exceptional) olay akışları temel use case akışının açıklanmasında kullanılır.

❑ Olası (exceptional) akışları betimleyen «use case» ler birden fazla olası «use case» e genişleyebilir.

❑ <<extends>> okunun yönü genişletilen use case yönüne doğrudur.

# «includes» ilişkisi



- ❑ «includes» ilişkisi «use case» in davranışını simgeleyen ilişkidir.
- ❑ «includes» davranışı yeniden kullanım (reuse) için tanımlanır, exception değildir.
- ❑ «includes» ilişkisinin yönü, kullanılan use case yönündedir.
- ❑ «extends» ilişkisinin yönünün tersi yönde işaretlenir.

# UML Diyagramlarının Ortak Sözdizimi

- **Aktörler(Actors):** Bir UML aktörü sistem ile bir arayüzü (etkileşim noktasını) işaret eder..
  - Aktörlerin kullanılmasının amacı sistem etkileşimlerinin gruplandırılmasıdır.
  - Aktörler insan /canlı olabildiği gibi, diğer sistemler de olabilir. Bir aktör modellenen sistemin bir parçası olamaz
  - Aktör sistemin dışında (external) olan sistem ile ilgili bir şeydir.
- **Kutular (Boxes):** ayrık elemanları, gruplamaları ve içermeleri göstermek üzere kullanılır.

# Aktör ve «Use Case» arasındaki etkileşim

- ❑ Bir aktör herhangi bir varlıkla ait pek çok «use case» vasıtası ile iletişimde olabilir.
- ❑ Bir «use case» işlevini gerçekleştirirken bir veya daha fazla aktör ile iletişimde olabilir.
- ❑ İki «use case» aynı varlığı betimleyerek birbirini ile iletişimde bulunamaz.
  - ❖ Çünkü «use case» lerin herbiri varlığı tek başına kullanır.
    - ✓ İki use case birbirleri ile iletişimde ise, *include* ya da *extend* ilişkisindeki *use case* aktörle bağıntılı olamaz.
      - Aktörle ilişki diğer use case den gelir.

# «Use Case» Tanımlamaları

□ Bir «use case» olası işlevlerini tanımlarken aktörünün aşağıdaki soruları cevaplaması beklenir.

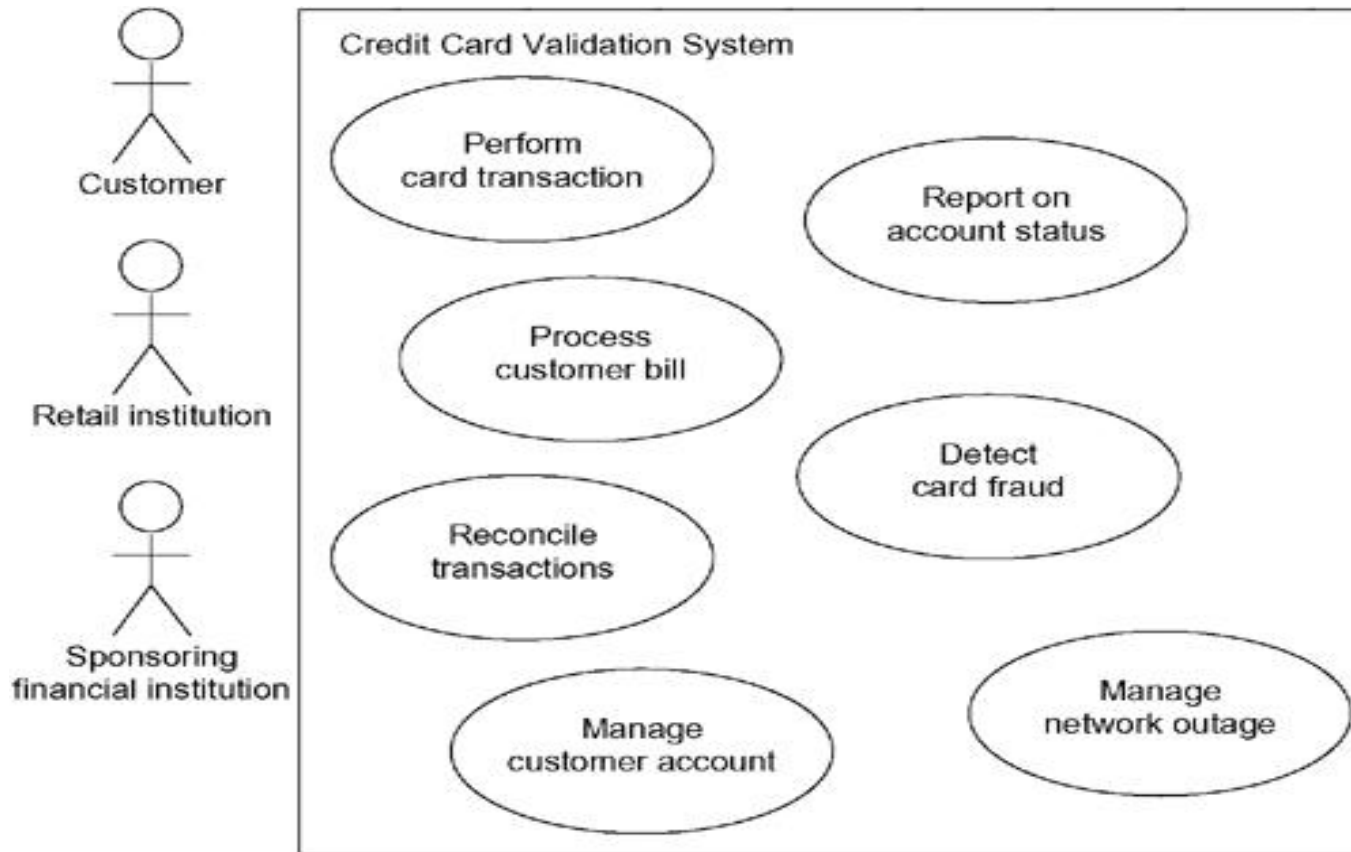
❖ Bunlar ne (what) sorularıdır.

- ✓ Aktör **ne** gerçekleştirmek istemektedir?
- ✓ Aktörün **neler** yapabilme kapasitesi vardır?
- ✓ Aktörün temel görevi (task) **nedir?**
- ✓ Aktörün sistemden alması gereken bilgiler **nedir?**
- ✓ Aktör sisteme **ne** bilgiler sağlar?

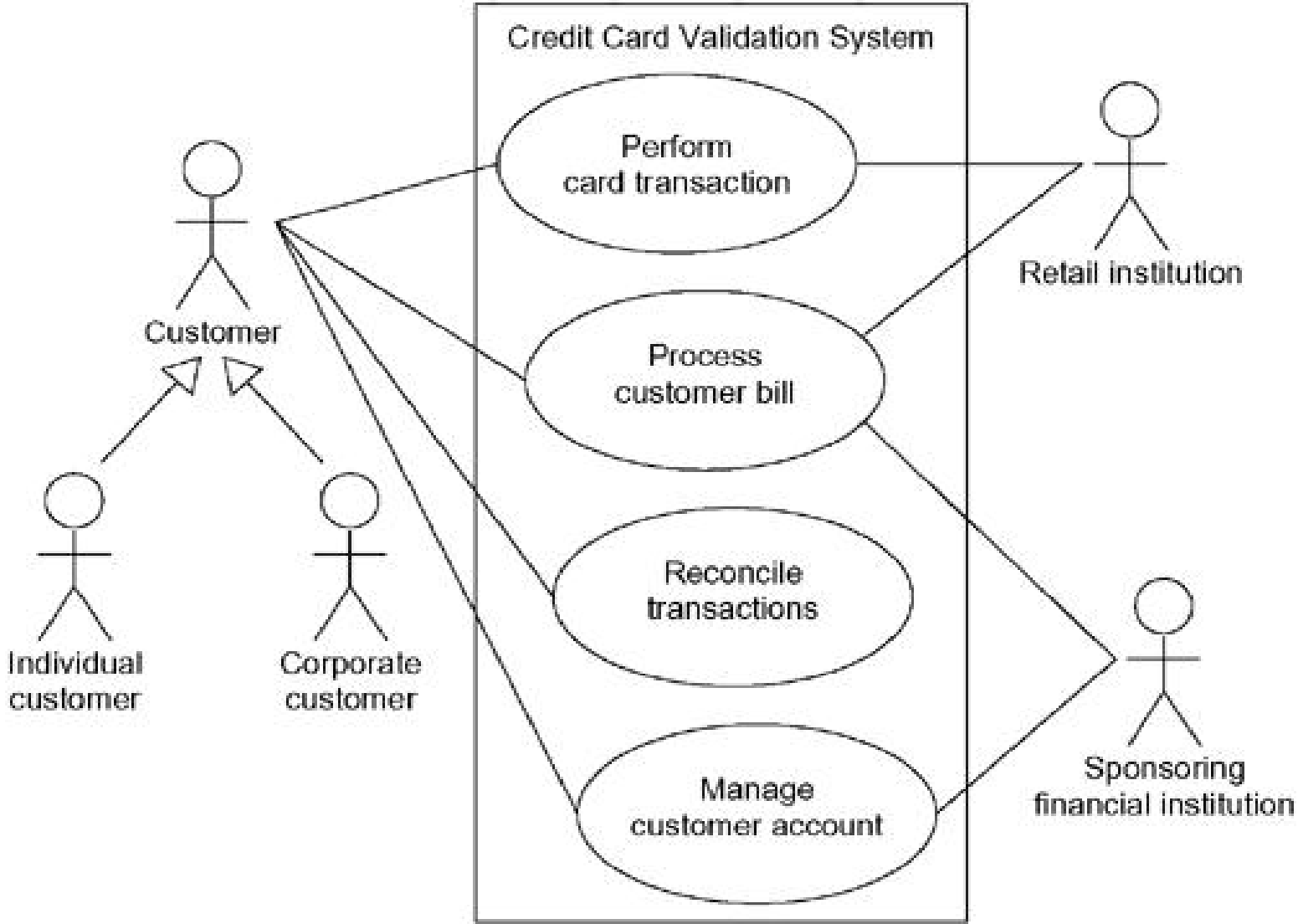
# «include» ve «extend» İlişkilerinin Karşılaştırması

	include use case	Extend use case
Bu «use case» seçmeli midir?	No	Yes
«Base use case» bu «use case» olmadan işlevini tamamlar mı?	No	Yes
Bu «use case» in çalışması koşula bağlı mıdır?	No	Yes
Bu «use case» «base use case» in davranışını değiştirir mi?	No	Yes

# Bir Yazılım Sistemine ait Gereksinimlerin Modellenmesi- I



# Bu Sistemin Bileşenlerinin Modellenmesi -II



# Sistemin Herhangi bir Elemanın Davranışının Modellenmesi -III

