

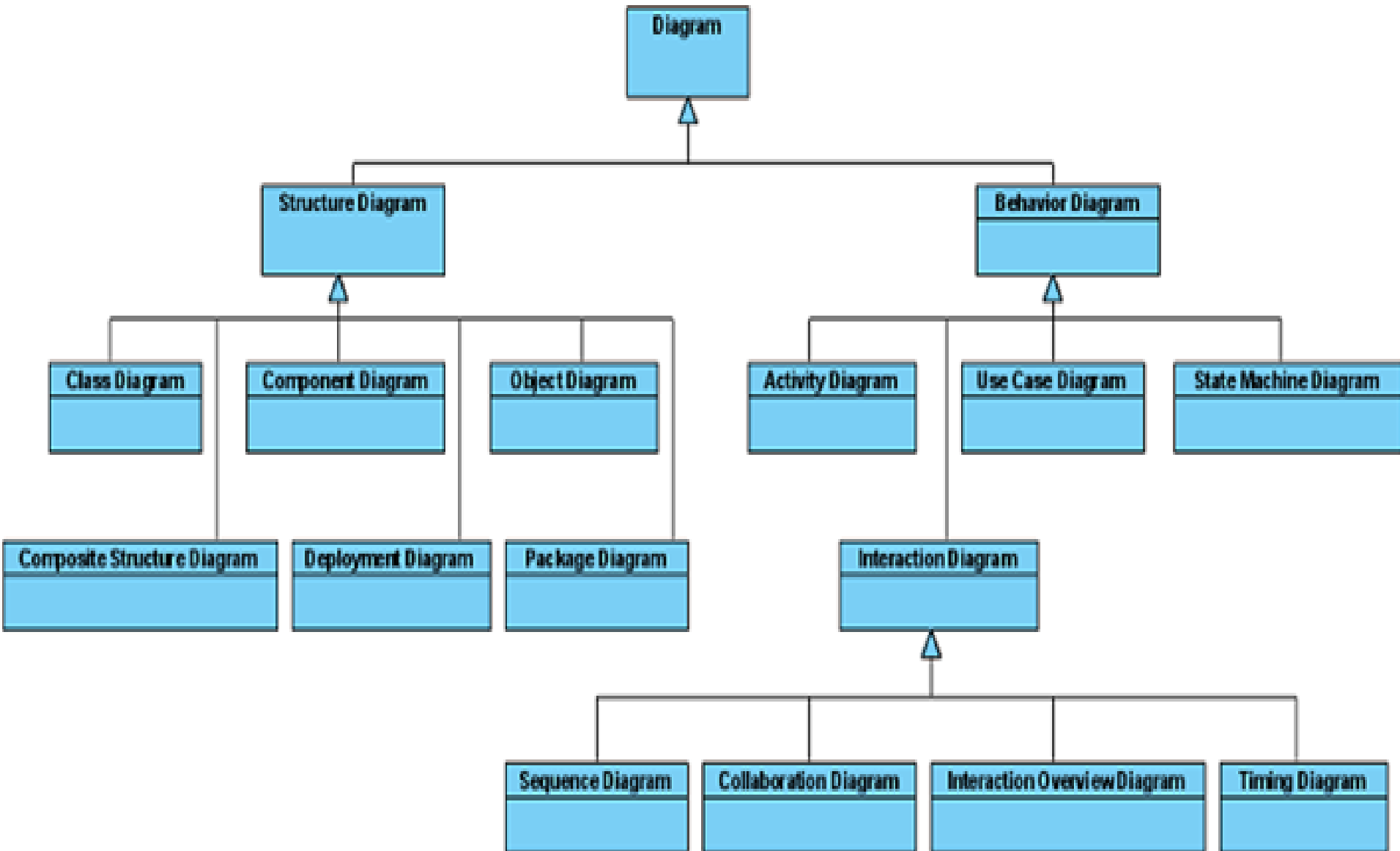
Yazılım Gereksinimlerinin Görsel
Çözümlenmeleri:

UML

(Unified Modeling Language)

Birleştirilmiş Modelleme Dili

UML Diyagramlarının Sınıflandırması

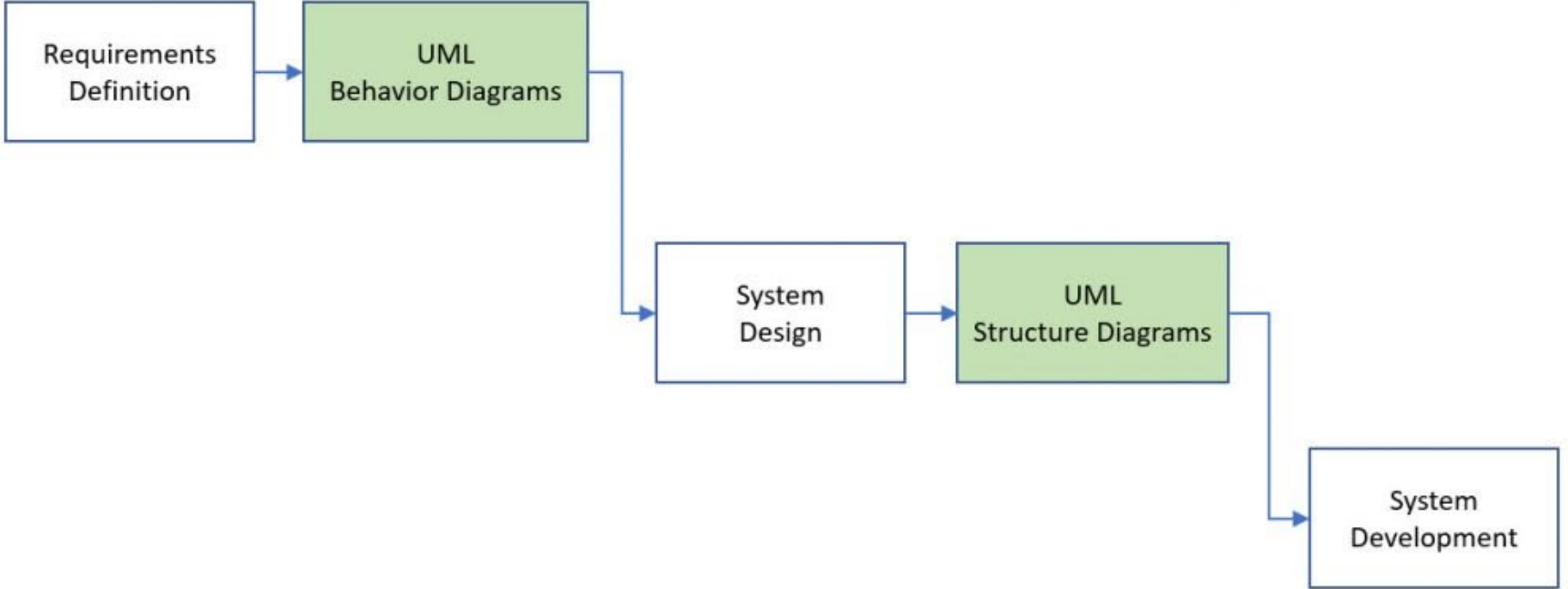


UML ile

Dinamik

Davranışsal (Behaviour) Modelleme

use case diyagramları



Use Case Diyagramı

□ **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.

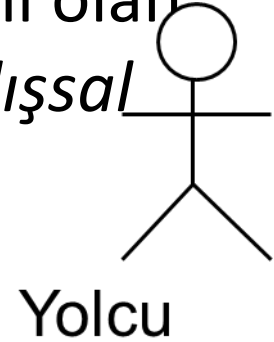
Aktör, dışsal kullanıcıların sistemle olan etkileşimlerini karakterize eder.

UML Reference Manual (1999, Rumbaugh)

Aktör ve use case

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

Örnek olarak: Yolcu, GPS uydusu



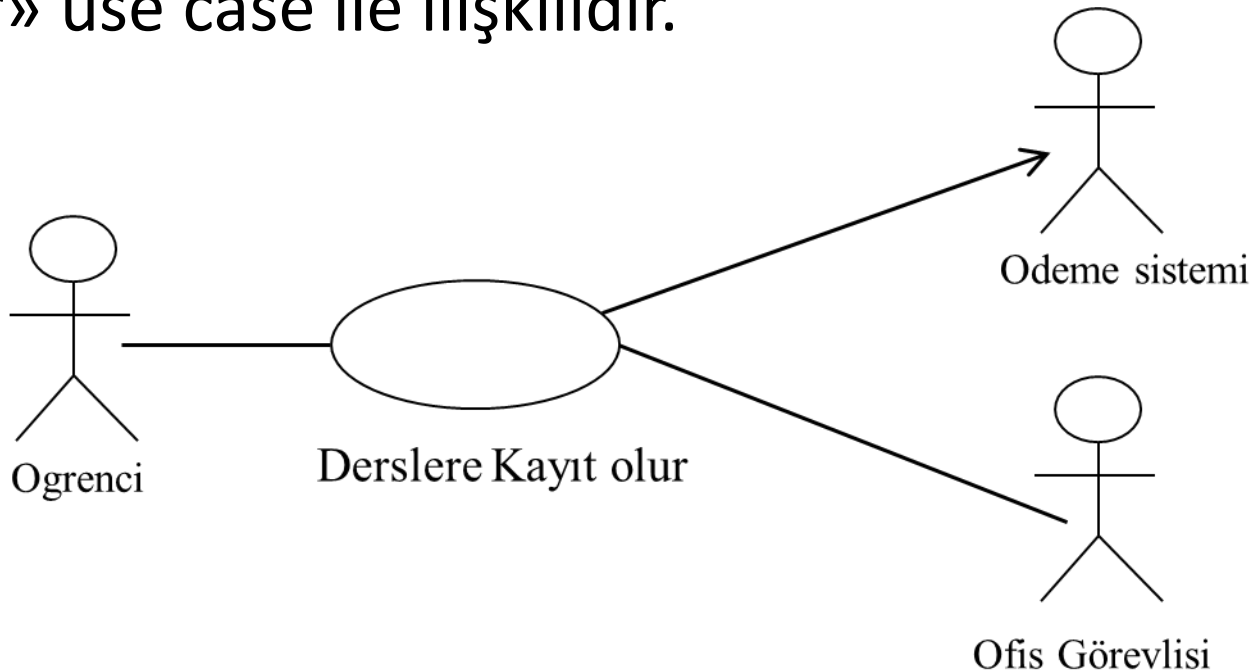
- ❑ Herbir «use case» tek bir isim ile tanımlanır, aktörlerle etkileşir, giriş koşulları ve çıkış koşulları içerir, olayların akışını ve özel gereksinimleri içerir.
- ❑ Herbir «use case», use case diyagramında elips ile gösterilir.
- ❑ Herbir «use case» mutlaka bir isim ile etiketlenmesi gerekir.

Bilet Satın Alır

aktör & use case =use case diyagramı

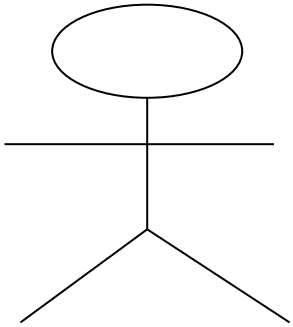
Öğrenci isimli aktör , «Derslere Kayıt olur» use case ile ilişkilidir.

Ödeme sistemi ve Ofis Görevlisi isimli aktörler «Derslere Kayıt olur» use case ile ilişkilidir.

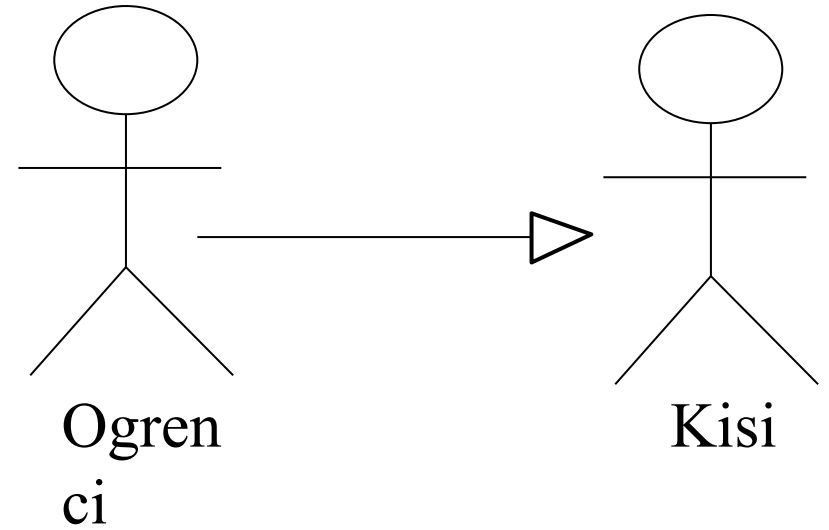


Aktör

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

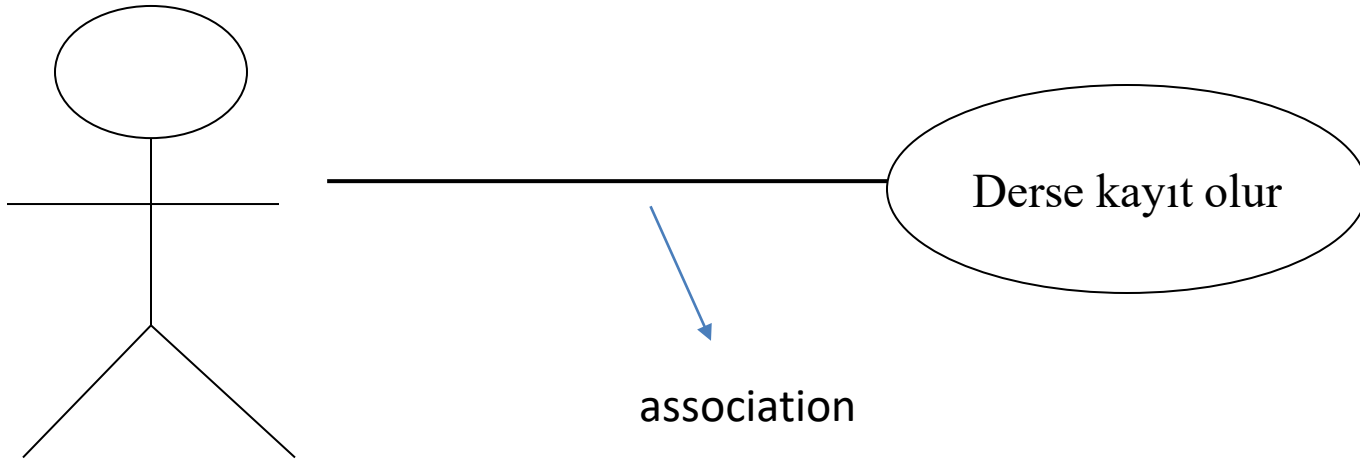


Bir aktörün , başka bir aktör ile genelleştirme (generalization) ilişkisi içerebilir.

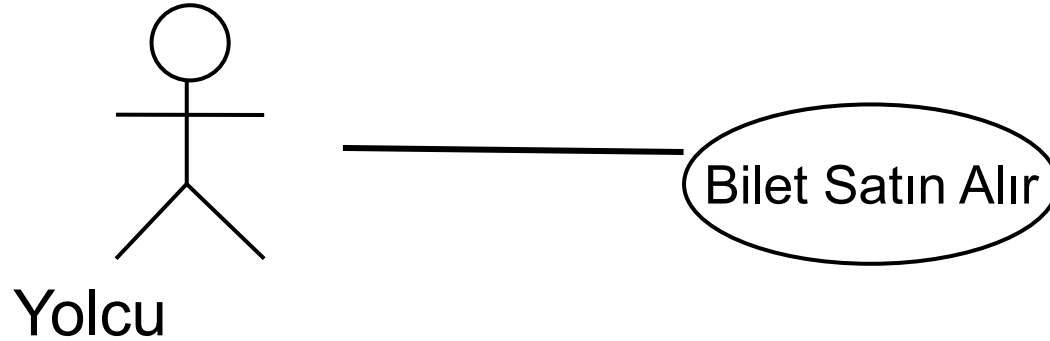


Aktör- use case ilişkisi

Aktörler ile use case bağlantısı «association» şeklinde gerçekleşir. Herhangi bir bildirim ve ilişki içeremez.



use case Diyagramları



- Gereksinimlerin anlaşılması-anlamlandırılması (requirements elicitation) sırasında harici (external) davranışı simgelemek üzere kullanılır.
- use case modeli tüm «use case» ler kümesinden oluşur.
- Use case diyagramı sistemin ve çevresinin fonksiyonelliğinin betimlemesidir.

Buna göre:

- Her bir «use case» belli bir fonksiyonelliğe ait bir dizi etkileşimi betimler.
- Aktörler sistemdeki bir tip kullanıcının rollerini betimler.

«use case» Tanımlamaları

□ Bir «use case» olası işlevlerini (fonksiyonelliğini) tanımlarken her bir aktörün 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?

use case Diyagramı Örneği

İsim: Bilet Satın Alınması

Aktör :Yolcu

Giriş Koşulu (Entry condition):

Yolcu bilet bankosunda bekler

Yolcunun bilet satın alacak parası vardır

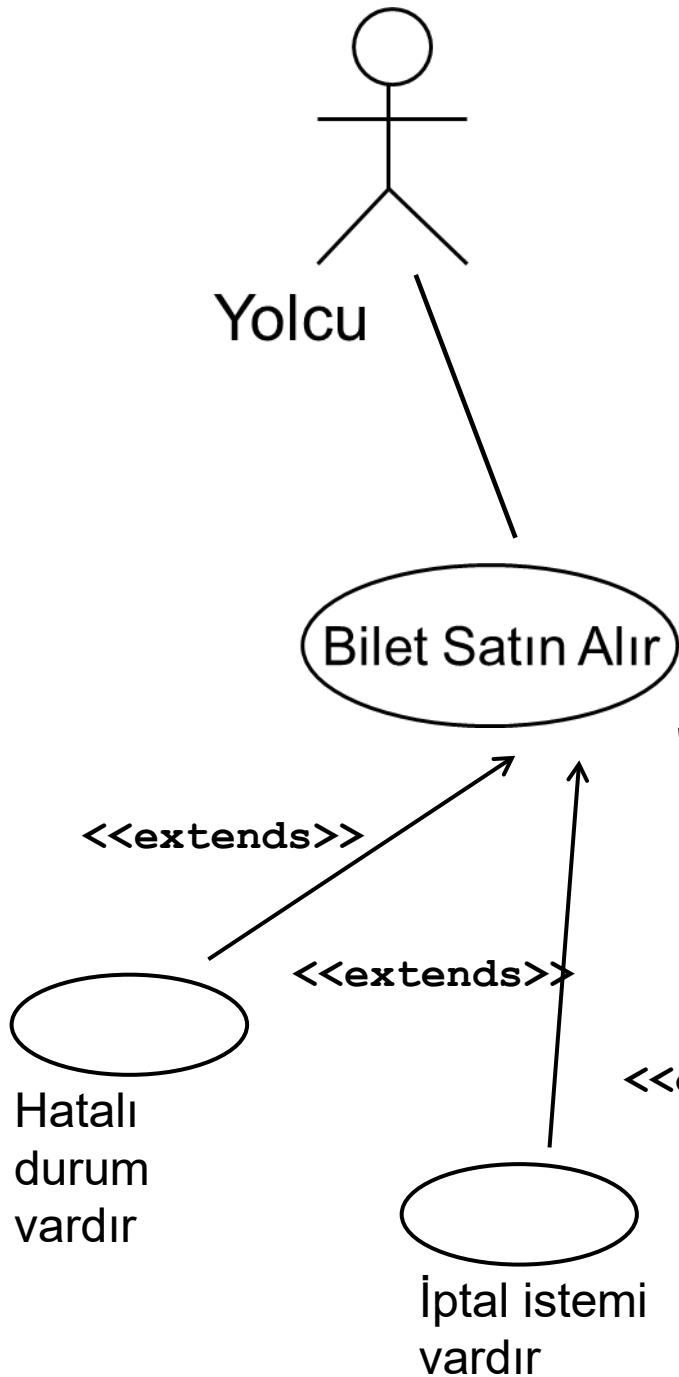
Çıkış Koşulu(Exit condition):

Yolcunun bileti vardır

Olay Akılı (Event flow):

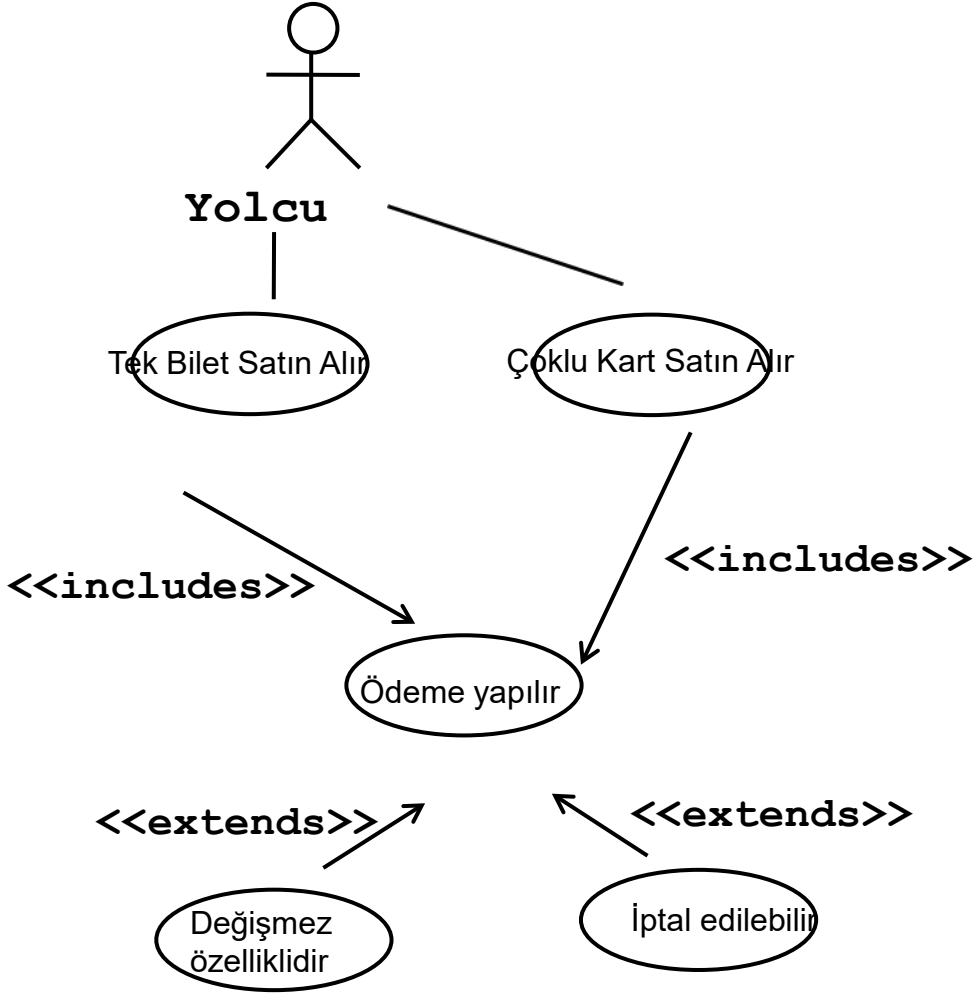
1. Yolcu gideceği bölgelerin sayısını belirler
2. Gişe memuru toplam ücreti belirler
3. Yolcu ödemeyi yapar
4. Gişe memuru üstünü verir
5. Gişe memuru bileti basar

<<extends>> ilişkisi



- ❑ Kural dışı (exceptional) veya az karşılaşılan *use case* ilişkilerini gösterir.
- ❑ Kural dışı olay akışları temel use case akışının açıklanmasında kullanılır.
- ❑ Olası akışları betimleyen «*use case*» ler birden fazla olası «*use case*» e genişleyebilir.
- ❑ <<extends>> okunun yönü genişletilen (base/temel) «*use case*» yönüne doğrudur.

«includes» ilişkisi



- ❑ «includes» ilişkisi, bir «use case»in davranışını (eylemlerini) diğer «use case» in içerdiğini simgeleyen ilişkidir.
- ❑ İki «use case» arasında gerçekleşir
- ❑ «includes» davranışı yeniden kullanım (reuse) için tanımlanır,
- ❑ «includes» ilişkisinin yönü, kullanılan (içerilen) **use case** yönündedir.
 - ❖ «extends» ilişkisinin yönünün tersi yönde işaretlenir.

use case örneği

Haberleri yayınlayan bir web sitesindeki ilişkilerin görsel tasarımı

4 aktör ve 2 use case tanımlayarak haberleri yayınlayan bir web sitesinin gereksinimlerinin görselleştirilmiş olarak kavramsal tanımlaması

Aktörler

Okuyucu (web kullanıcısı)

Tasarımcı

Editör

Ajans -----Sistemle iletişimde bulunan özel bir aktördür. Fakat diğer aktörler gibi canlı bir varlık (insan) değildir.

Use cases

web arayüzü

içeriğin tutulduğu veritabanı

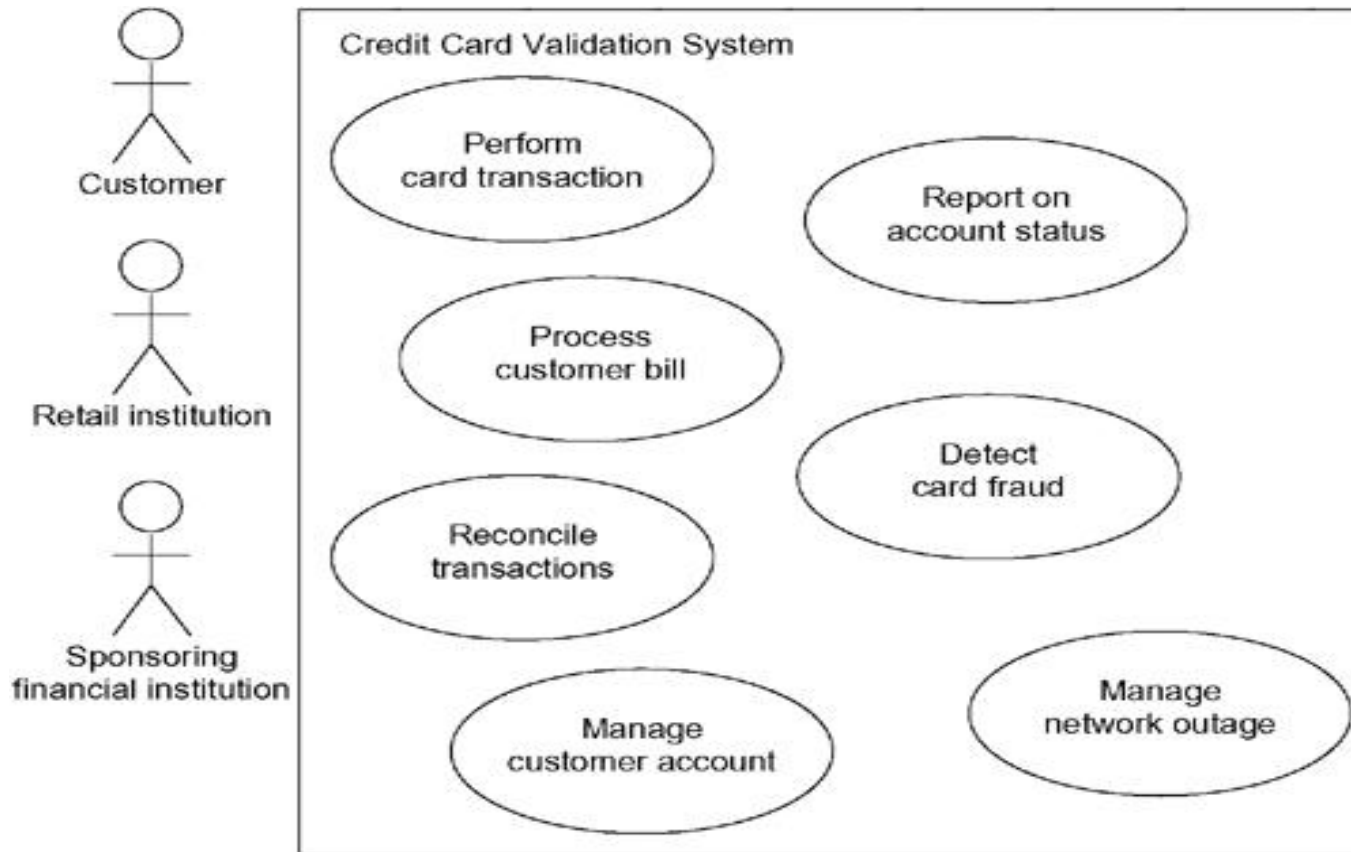
aktörler ve «use cases» arasındaki iletişimler

- ❑ Bir aktör (bir varlık/entity) pek çok «*use case*» ile iletişimde olabilir.
- ❑ Bir «*use case*» işlevini gerçekleştirirken bir veya daha fazla aktör ile iletişimde olabilir.
- ❑ İki «*use case*» aynı aktörü (varlığı) betimleyerek birbirini ile iletişimde bulunamaz.
 - ❖ Çünkü «*use case*» lerin her biri aktörü (varlığı) tekil olarak (kendi *aktör - use case* ilişkisi ile) kullanır.
 - ✓ İki *use case* birbirleri ile iletişimde ise , *include* ya da *extend* ilişkisindeki *use case* aktörle bağıntılı olamaz.
 - İlişkideki (*include* ya da *extend*) *use case* aktörle ilişkisini diğer *use case* üzerinden alır.

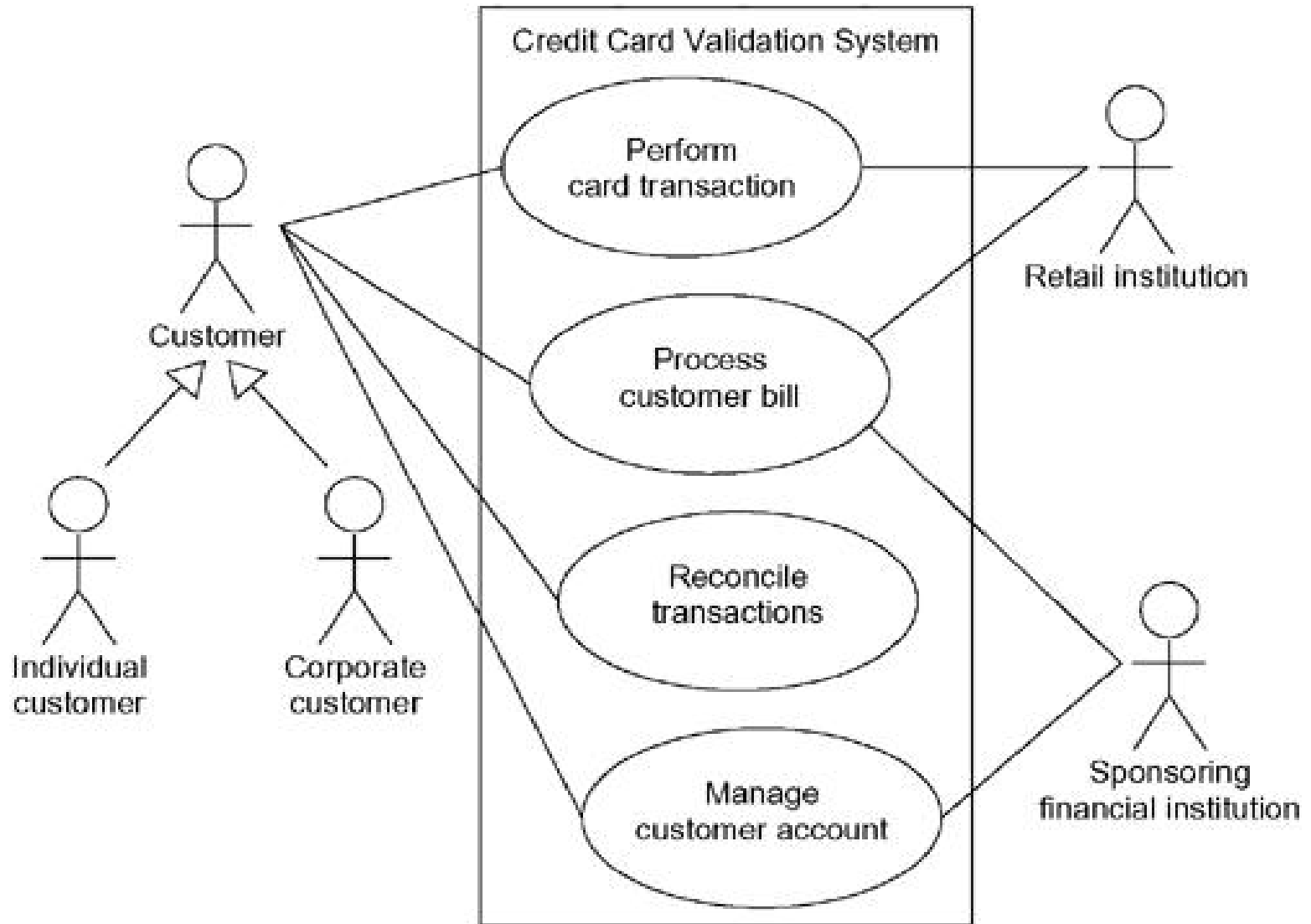
«includes» ve «extends» ilişkileri arasındaki temel farklar

	include use case	Extend use case
Yandaki «use case» seçmeli midir (opsiyonel midir?)	No	Yes
«Base (Temel) <<use case», yandaki «use case» olmadan işlevini tamamlar mı?	No	Yes
Yandaki «use case» in çalışması koşula bağlı mıdır?	No	Yes
Yandaki «use case», «base (temel) use case» in davranışını değiştirir mi?	No	Yes

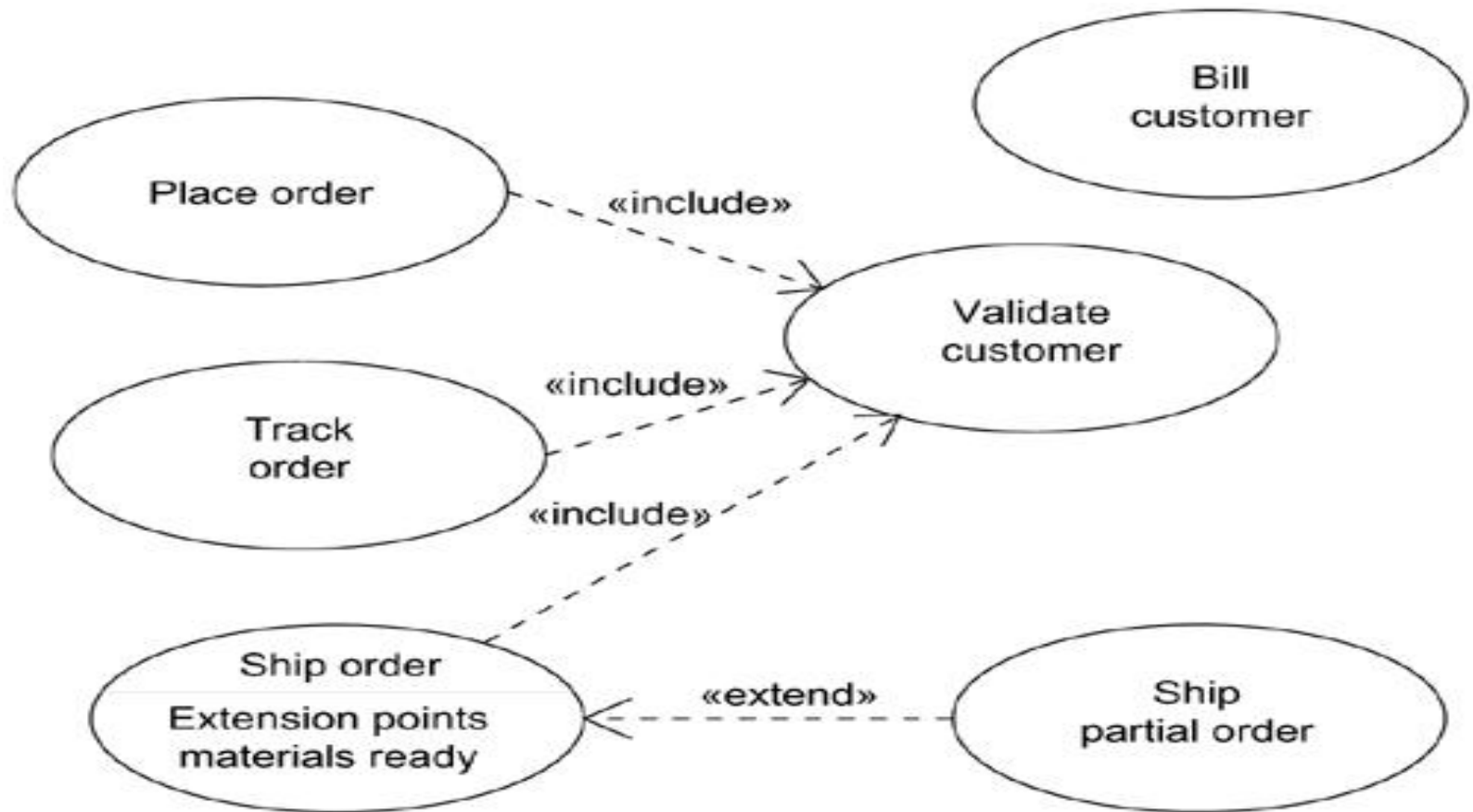
Yazılım Sisteminin Gereksinimlerinin Modellenmesi



Sistem Bileşenlerinin Modellenmesi



Herhangi bir Elemanın Davranışının Modellenmesi



Davranışsal Diyagramlar II

(Behavioural Diagrams)

«Sequence» Diyagram

Sequence Diyagramı

- ❑ «Sequence» diyagramı bir nesneden (object) diğer nesneye aktarılan mesajları gösterir.
- ❑ Bu akışlar fonksiyon çağrılarını şeklinde gerçekleştirir.
- ❑ Böylece «sequence diyagramı», belli bir işlevi (fonksiyonelliği gerçekleştirmek üzere) sistem tarafından gerçekleştirilen bir dizi çağrıdan meydana gelir.

Nesneler, ömür çizgisi, mesajlar

(Objects, Lifelines, Messages)

- ❑ Nesneler ve sınıflar en üstte birlikte bulunurlar.
- ❑ Nesneleri sınıflardan ayırmanın bir yolu nesnelerin altında çizgi olmasıdır.
- ❑ Çöp adam (aktör) solda isimsiz bir nesneyi simgeler.
 - ❖ Bu çöp adam kaynaktır (source) ve sisteme giren ve çıkan tüm mesajları alır (biriktirir)
 - ❖ Tüm «sequence» diyagramlarında isimsiz aktör olmak zorunda değildir.

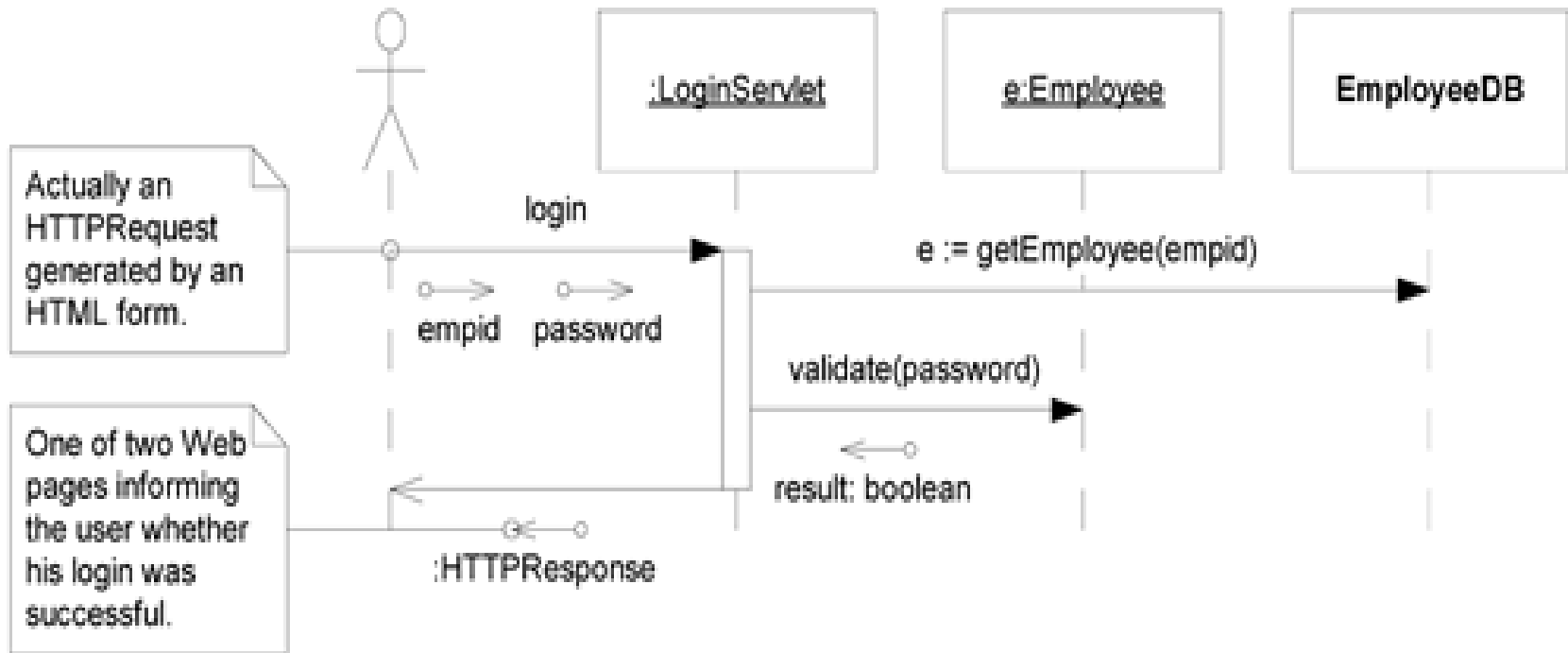
Nesneler, ömür çizgisi, mesajlar

(Objects, Lifelines, Messages)

- ❑ Nesnelere aşağıya kesikli çizgiler (dashed lines) aktörlerin ömrü olarak adlandırılır (lifelines).
- ❑ Bir nesneden diğerine gönderilen mesaj ok ile gösterilir (iki ömür çizgisi (lifeline) arasındaki oktur)
- ❑ Her mesajın bir ismi vardır (isim ile etiketlenmiştir)
- ❑ Argümanlar parantez içerisinde görüntülenirler.
- ❑ Zaman (Time) dikey boyuttur; böylece daha düşük bir mesaj gönderildikten sonra görünür.

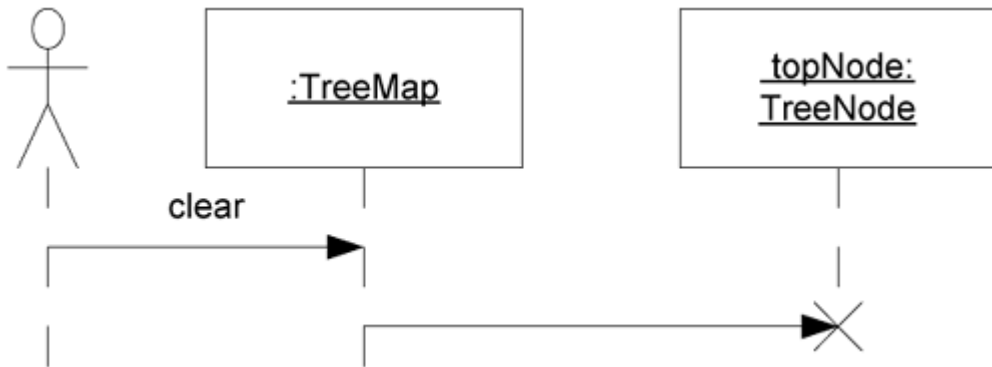
Örnek 1

Sequence Diyagramı



Örnek 2

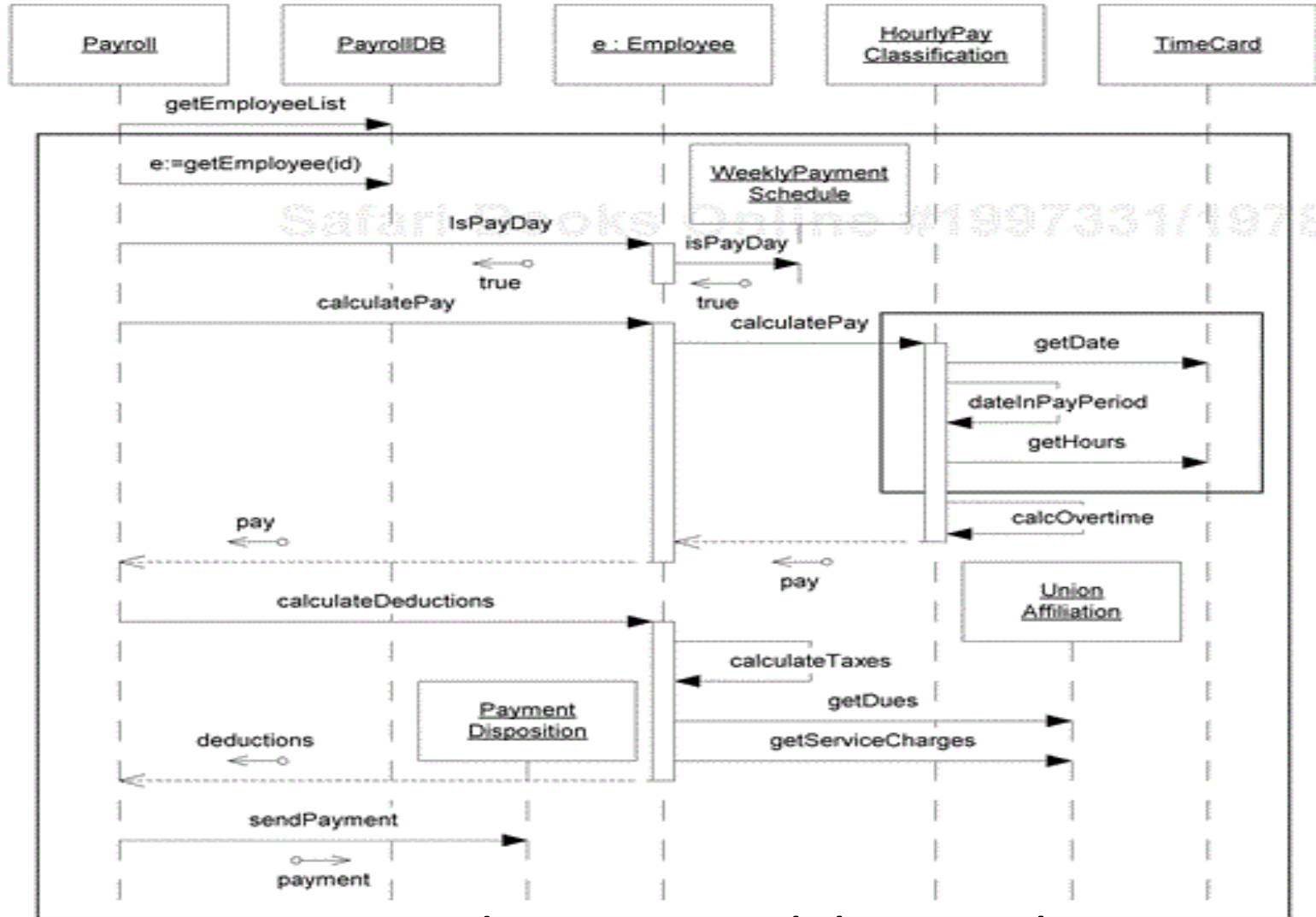
Sequence Diyagramı



Bu diyagram bir nesneyi çöp kutusuna gönderir.
(Releasing an object to the garbage collector)

Örnek 3

Sequence Diyagramı



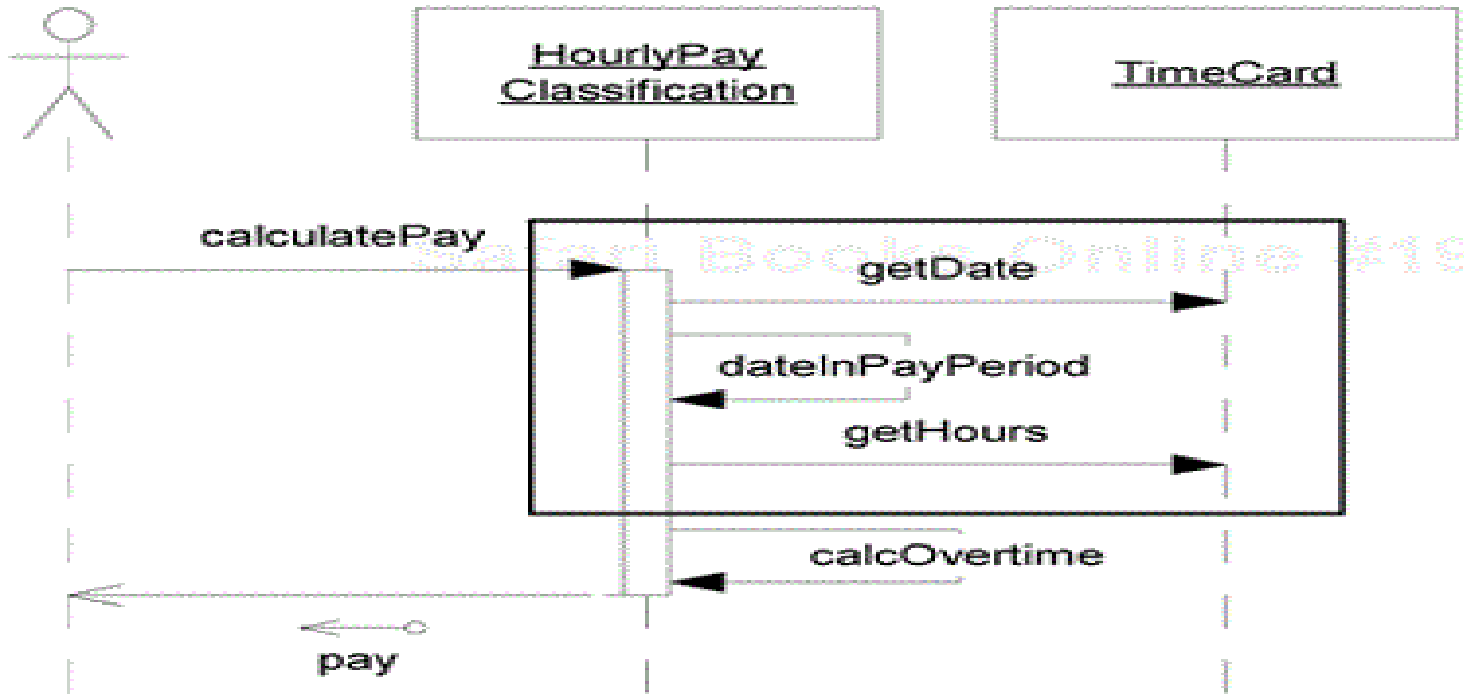
Bu sequence diyagramı çok karmaşıktır.

«Sequence» Diyagramı

Tasarlama Kuralları

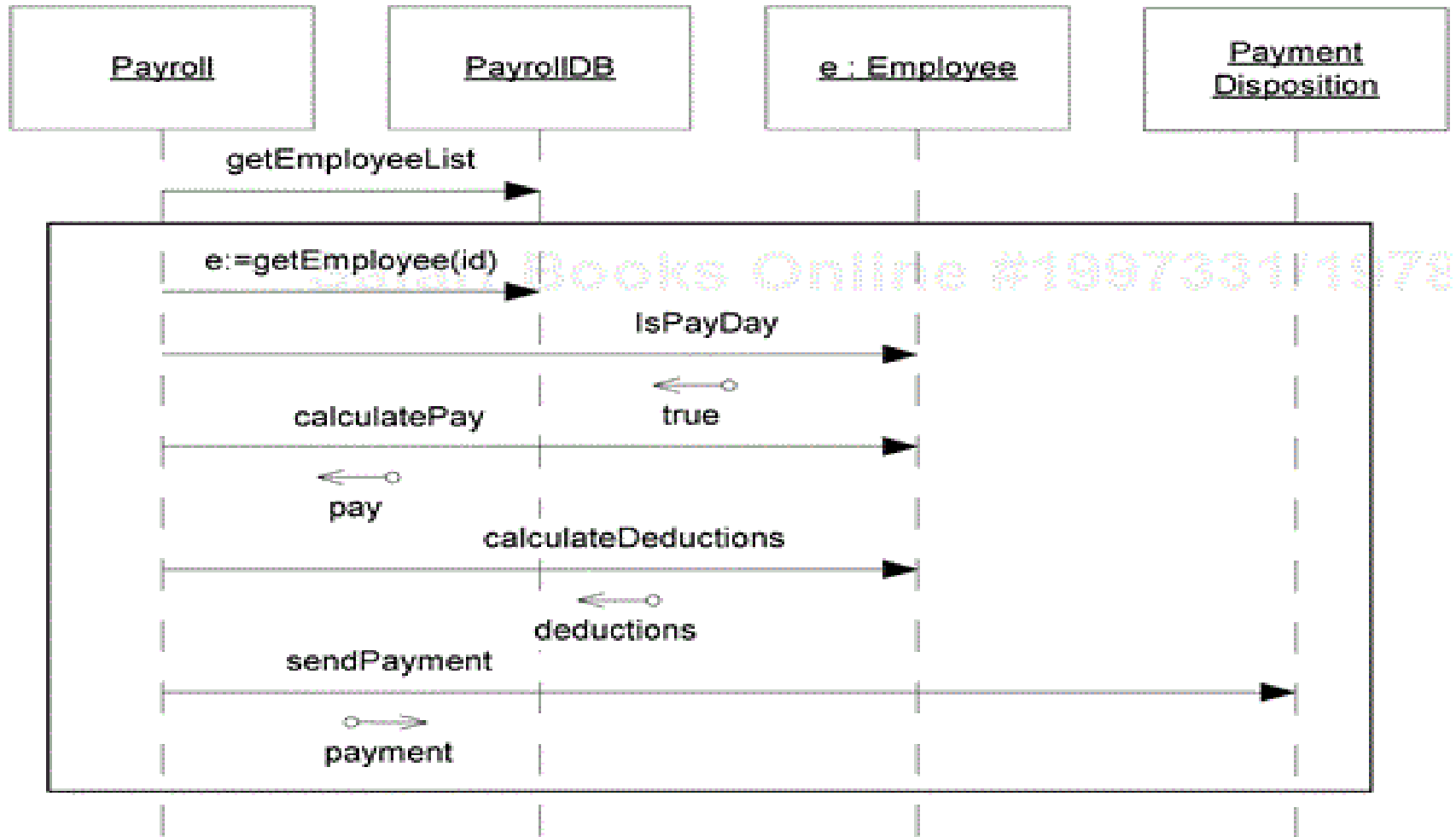
- ❑ <<sequence>> diyagramları kesinlikle karmaşık tasarlanmamalıdır
- ❑ Karmaşık bir diyagramı kimse okuyamaz .
- ❑ Karmaşık bir diyagramı okuyabilecek kişi olsa bile karmaşıklığından dolayı okumayacaktır.
- ❑ Bu da gereksiz bir zaman kaybıdır.
- ❑ O nedenle daha küçük «sequence» diyagramının nasıl tasarlanacağı öğrenilmelidir.
- ❑ Her «sequence» diyagramı tek bir sayfada tasarlanmış olmalıdır

Örnek 3 ile ilgili Farklı Durum

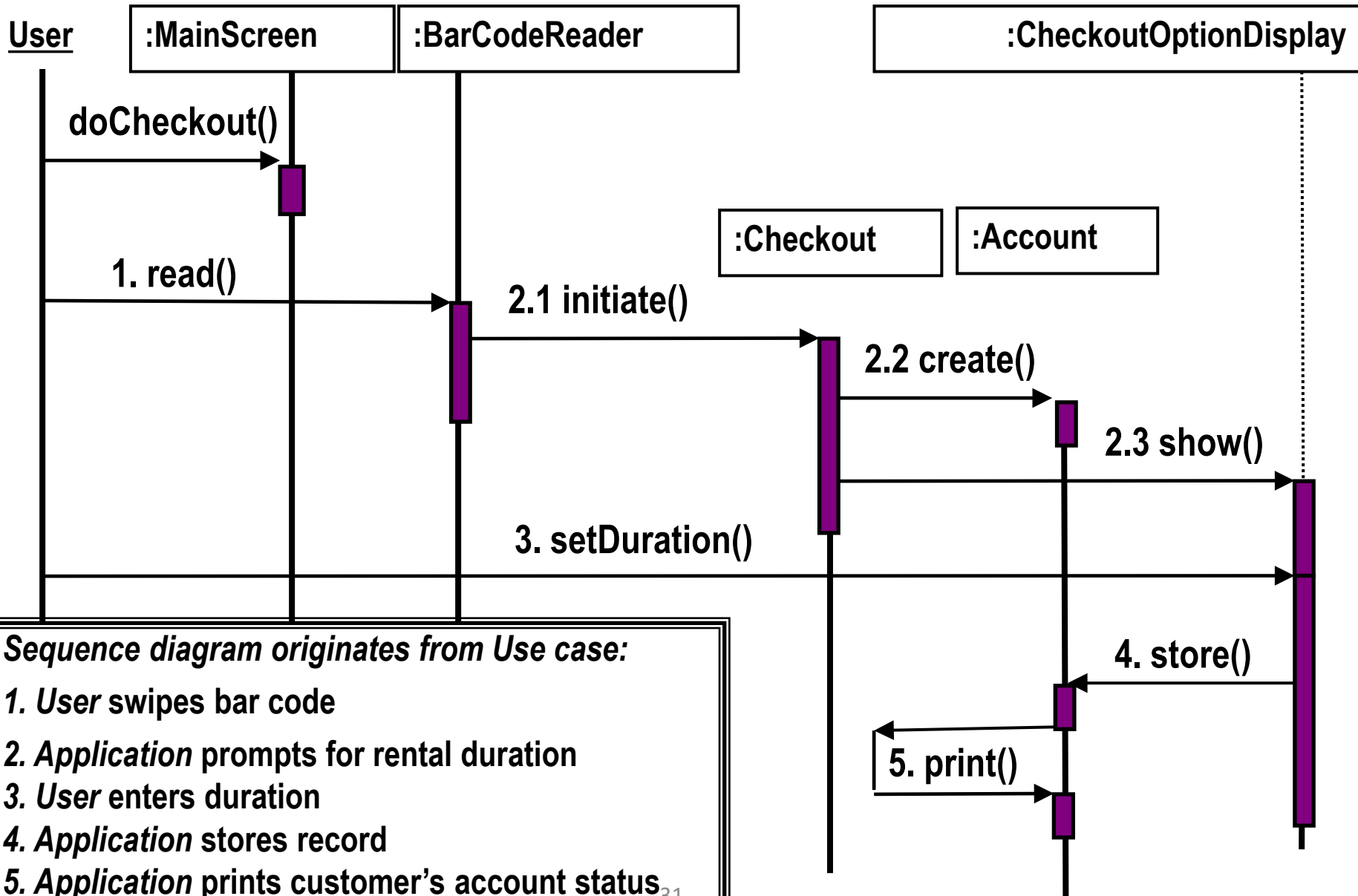


Örnek 3 Sequence Diyagramına ilişkin farklı senaryo

Örnek 3 diyagramı ile ilgili Farklı Durum ve Senaryo



Örnek 4 Sequence Diyagram (devam)



Sequence diagram originates from Use case:

1. User swipes bar code
2. Application prompts for rental duration
3. User enters duration
4. Application stores record
5. Application prints customer's account status

Visual Paradigm Tool (free)

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

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