

Erciyes Üniversitesi
Bilgisayar Mühendisliği Bölümü

BZ 313 Yazılım Mühendisliği

7. Senaryolar ve Use-Case'ler

Senaryolar

Senaryo

Senaryo, önerilen bir sistemle bazı etkileşimleri gösteren bir **sahnedir**.

Senaryo, önerilen bir sistemin belirli bir kullanımını tanımlamak için gereksinim analizi sırasında kullanılan bir araçtır. Senaryolar, belirli örnekler kullanarak sistemi dışarıdan bakıldığında, örneğin bir kullanıcı tarafından nasıl görüneceğini ele alır.

Terminoloji hakkında not

Bazı yazarlar, bir kullanıcının sistemle toplam etkileşimine atıfta bulunmak için "senaryo" kelimesini kısıtlar.

Diğer yazarlar, etkileşimin bölümlerine atıfta bulunmak için "senaryo" kelimesini kullanır. Bu derste terim her iki anlamda da kullanılmaktadır.

Bir Senaryonun Tanımı

Bazı kuruluşların bir senaryoyu açıklamak için karmaşık belge standartları bulunmaktadır.

En azından, açıklamada şu bilgiler bulunmalıdır:

- Senaryonun **amacına** yönelik bir beyan
- Senaryo boyunca izlenen bireysel **kullanıcı** veya **işlem**
- **Ekipman** veya yazılımla ilgili varsayımlar
- Senaryonun **adımları**

Bir Müşteri ile Senaryo Geliştirme

Bir müşteriyle nasıl senaryo geliştirileceğine dair örnek

Üniversite öğrencilerinin web tarayıcısı kullanarak kendi odalarından online olarak sınava girmelerini sağlayacak bir sistem için gereksinimler geliştiriliyor.

Tipik bir öğrencinin sistemle nasıl etkileşime girdiğine dair bir senaryo oluşturun.

Sonraki birkaç slaytta, mavi renkli sorular, senaryoyu geliştirirken müşteriye sorulabilir örnek sorulardır.

Bir Müşteriyle Senaryo Geliştirme: Temsili Bir Öğrenci

Amaç: Temsili bir öğrenci tarafından çevrimiçi bir Sınav sisteminin kullanımını açıklayan senaryo

Birey: *[Tipik bir öğrenci kimdir?]* Öğrenci A, Erciyes Üniversitesinde Bilgisayar Mühendisliği bölümü son sınıf öğrencisi,. *[Öğrenci nerede bulunabilir? Diğer üniversiteler farklı mı?]*

Ekipman: Desteklenen bir tarayıcıya sahip herhangi bir bilgisayar. *[Desteklenen tarayıcıların bir listesi var mı? Herhangi bir ağ kısıtlaması var mı?]*

Senaryo:

1. Öğrenci A kimlik doğrulaması yapar. *[Bir Erciyes öğrencisi nasıl kimlik doğrulaması yapar?]*
2. A öğrencisi tarayıcıyı başlatır ve Sınav sisteminin URL'sini yazar. *[Öğrenci URL'yi nasıl biliyor?]*
3. Sınav sistemi seçenekler listesini görüntüler. *[Liste bireysel kullanıcıya göre uyarlanmış mı?]*

Bir Müşteri ile Senaryo Geliştirme (Devam)

4. A öğrencisi BZ 1234 Sınav 1'i seçer.
5. Her biri tamamlanıp tamamlanmadığını belirtmek için işaretlenmiş bir soru listesi görüntülenir. *[Sorular herhangi bir sırayla cevaplanabilir mi?]*
6. A öğrencisi bir soru seçer ve yeni bir yanıt göndermeyi veya önceki bir yanıtı düzenlemeyi seçebilir. *[Önceki bir cevabı düzenlemek her zaman mümkün müdür? Başka seçenekler var mı?]*
7. *[Ne tür sorular var: metin, çoktan seçmeli vb.?]* İlk soru için yazılı bir cevap verilmektedir. Öğrenci A yeni bir cevap gönderir. Öğrenci, çözümü tarayıcıya yazma veya ayrı bir dosya ekleme seçeneğine sahiptir. A öğrencisi bir dosya eklemeye karar verir. *[Ne tür dosyalar kabul edilir?]*

Bir Müşteri ile Senaryo Geliştirme (Devam)

8. İkinci soru için, öğrenci önceki bir yanıtı düzenlemeyi seçer. Öğrenci A, daha önce tarayıcıya yazılmış bir çözümü silmeyi ve ekli bir dosyayla değiştirmeyi seçer. *[Öğrenci önceki bir yanıtı düzenleyebilir mi, yoksa her zaman yeni bir yanıtla mı mevcut soruyu değiştirebilir?]*
9. Tüm sınavı tek bir oturumda tamamlamaya alternatif olarak, Öğrenci A tamamlanan soruları daha sonra devam etmek üzere kaydetmeye karar verir. *[Buna her zaman izin veriliyor mu?]*
10. A Öğrencisi oturumu kapatır.
11. Daha sonra Öğrenci A oturum açar, sınavı bitirir, cevapları gönderir ve oturumu kapatır. *[Bu süreç, bu sınavdaki ilk defa sayfayı açmaktan farklı mı?]*
12. A Öğrencisi artık sınavı tamamlamıştır. Öğrenci, sınavı not sistemine gönderen bir seçenek belirler. *[Öğrenci her soruyu çözmediyse ne olur? Not veren kişi bilgilendirildi mi?]*

Bir Müşteri ile Senaryo Geliştirme (Devam)

- Bir müşteriyle bir senaryo geliştirmek, bir sistem kurulmadan önce üzerinde anlaşmaya varılması gereken politikalar, prosedürler vb. gibi birçok **işlevsel gereksinimi** açıklığa kavuşturur.
- Senaryo genellikle **kullanıcı arayüzü gereksinimlerini** netleştirir, ancak kullanıcı arayüzü tasarımı senaryonun bir parçası olmamalıdır.

Bu senaryo oldukça basit olmasına rağmen, birçok ayrıntı atlanmıştır.

Karmaşık bir sistemde birçok senaryo bulunabilir.

Özel Gereksinimleri Analiz Etme Senaryoları

Senaryolar özellikle olağandışı özel gereksinimleri analiz etmek için çok kullanışlıdır.

Örnekler

- **Geri Dönüşler.** Bir finansal sistemde, bir işlem yanlış hesaba yatırılır. İşlemi tersine çevirmek için hangi adım dizisi kullanılır?
- **Hatalar.** Bir posta siparişi şirketi, envanter veritabanının birkaç kopyasına sahip olabilir. Bu veritabanları tutarsız hale gelirse ne olur?
- **Görevi kötüye kullanma.** Bir oylama sisteminde, bir seçmenin iki şehirde evi vardır. Her ikisinde de oy kullanmaya kalkarsa ne olur?

Hata kurtarma senaryoları

Murphy Yasası: "*Bir şeyler ters gidebilirse, gider*". Yanlış gidebilecek şeyleri sistemin nasıl ele alması bekleniyorsa bütün bu senaryoları oluşturun.

Use-Case Olarak Senaryo Modelleme

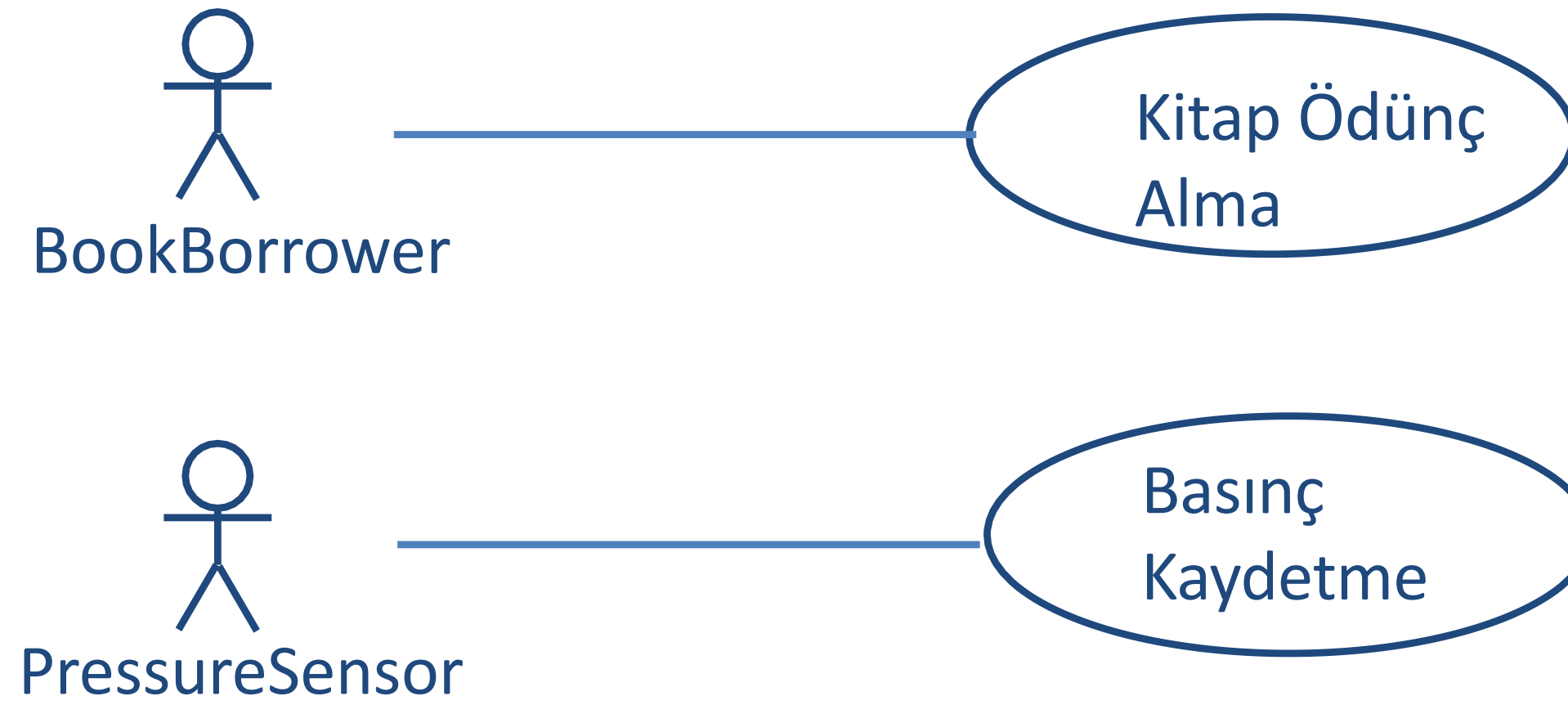
Modeller

Senaryolar, önerilen bir sistemi bir müşteriyle tartışmak için yararlıdır, ancak bir sistem tam olarak anlaşılmadan önce gereksinimlerin daha kesin hale getirilmesi gerekir.

Gereksinim **modellemesinin** amacı budur.

Use case'ler bu türde bir model oluşturulmasını sağlar.

İki Basit Use Case



Aktörler ve Use Case Diyagramları



BookBorrower



PressureSensor

Kitap Ödünç
Alma

Basınç
Kaydetme

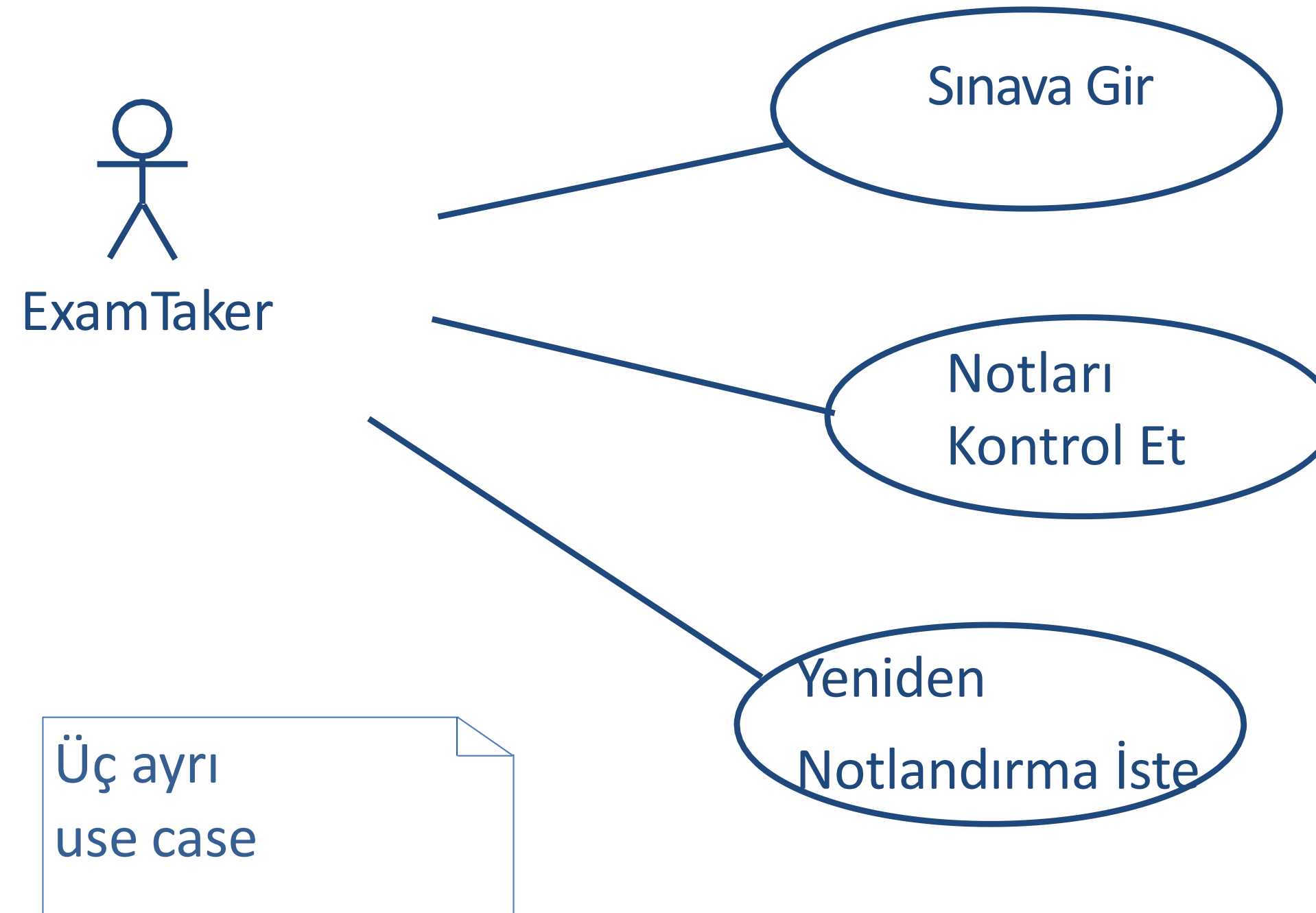
- **Aktör**, belirli bir roldeki bir sistemin **kullanıcısıdır**.
Bir aktör insan veya harici bir sistem olabilir.
- Bir **use case**, bir aktörün sistem yardımıyla gerçekleştirmesi gereken görevine karşılık gelir.

Use Case ve Aktörler

- **Aktör** bir birey değil, roldür (Örneğin, kütüphanecinin birçok rolü olabilir)
- Aktör, bir use-case'ten **yararlanmalıdır**.
(örneğin, ödünç alındığında kitabı işleyen kütüphaneci değil)

Aktörleri adlandırırken, "kullanıcı" veya "müşteri" gibi genel adları değil, rolü tanımlayan adları seçin.

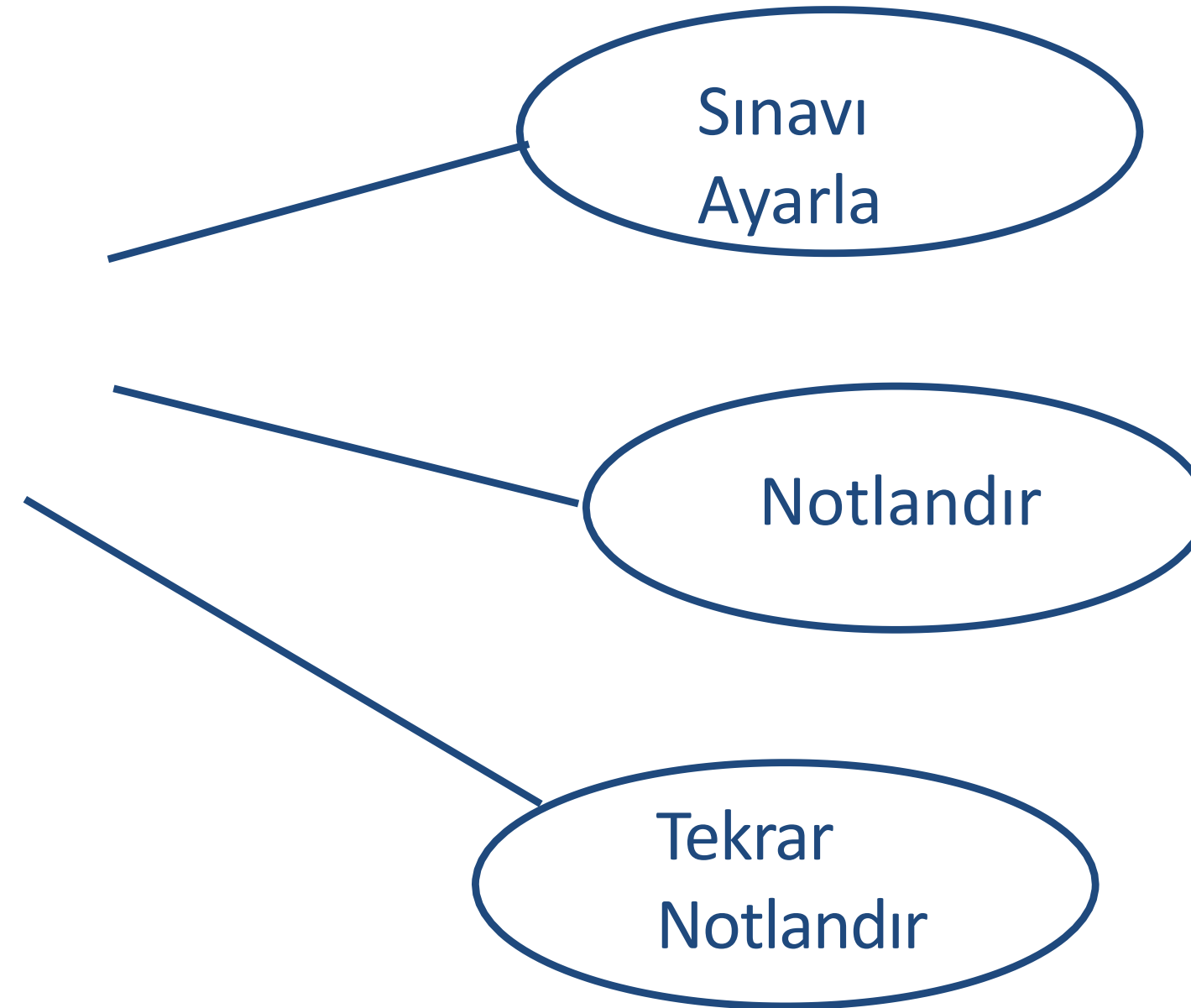
Örnek: Sınav Sistemi için Use-Case



Sınav Sistemi için Use Case (Devam)



Aktörün bir rol olduğunu unutmayın. Bir kişi bir vesileyle Sınav Katılımcısı ve farklı bir zamanda Öğitmen olabilir.



Use Case Açıklaması

Bazı kuruluşların karmaşık dokümantasyon standartları vardır.

Metadata

- Use case'in **adı**
- Use case'in **amacı**
- **Aktör** veya **aktörler**
- **Tetikleyici**
- Başlangıçta **giriş koşulları**
- En sondaki **son durum**

Olayların akışı

- Olayların **temel akışı**
- Alternatif **olay akışları**
- **Özel durumlar**

Sınava Girme Use Case'i: Metadata

Use-Case Adı: Sınava Gir

Hedef: Öğrencinin bir web tarayıcısı ile çevrimiçi olarak sınava girmesini sağlar.

Aktör(ler): ExamTaker

Tetikleyici: ExamTaker'a sınavın hazır olduğu bildirilir.

Giriş koşulları: ExamTaker'ın kursa kayıtlı olması gerekmektedir. ExamTaker'ın kimlik doğrulama kimlik bilgilerine sahip olması gerekir.

Son durum: Tamamlanan sınav notlandırılmaya hazır.

Sınava Girme Use Case'i: Temel Akış

Olayların temel akışı:

1. ExamTaker sunucuya bağlanır.
2. Sunucu, ExamTaker'ın kimliğinin önceden doğrulanıp doğrulanmadığını kontrol eder ve gerekirse kimlik doğrulama işlemini çalıştırır.
3. ExamTaker, seçenekler listesinden bir sınav seçer.
4. ExamTaker tekrar tekrar bir soru seçer ve bir çözüm yazar, çözüm içeren bir dosya ekler veya bir çözümü düzenler.
5. ExamTaker ya tamamlanmış sınavı gönderir ya da mevcut durumu kaydeder.
6. Tamamlanmış bir sınav gönderildiğinde, sunucu tüm soruların denenip denenmediğini kontrol eder ve ExamTaker'a onay gönderir.
7. ExamTaker oturumu kapatır.

Sınava Girme Use Case'i: Alternatif Akış

Alternatif akışlar ve özel durumlar, temel akış dışındaki kullanım örneği üzerinden yolları modeller.

Aşağıdaki listede, her akış temel akışın bir adımına bağlanmıştır.

Alternatif akışlar, kullanım örneğinin başarıyla tamamlanması için alternatif yollardır.

3. ExamTaker daha önce sınavın bir bölümüne girmiş, ancak göndermemiş.
4. Çözüm dosyası sistem tarafından kabul edilmedi.
6. Eksik gönderim.

Özel durumlar, use case'in başarısız olmasına yol açar.

2. Kimlik doğrulama hatası

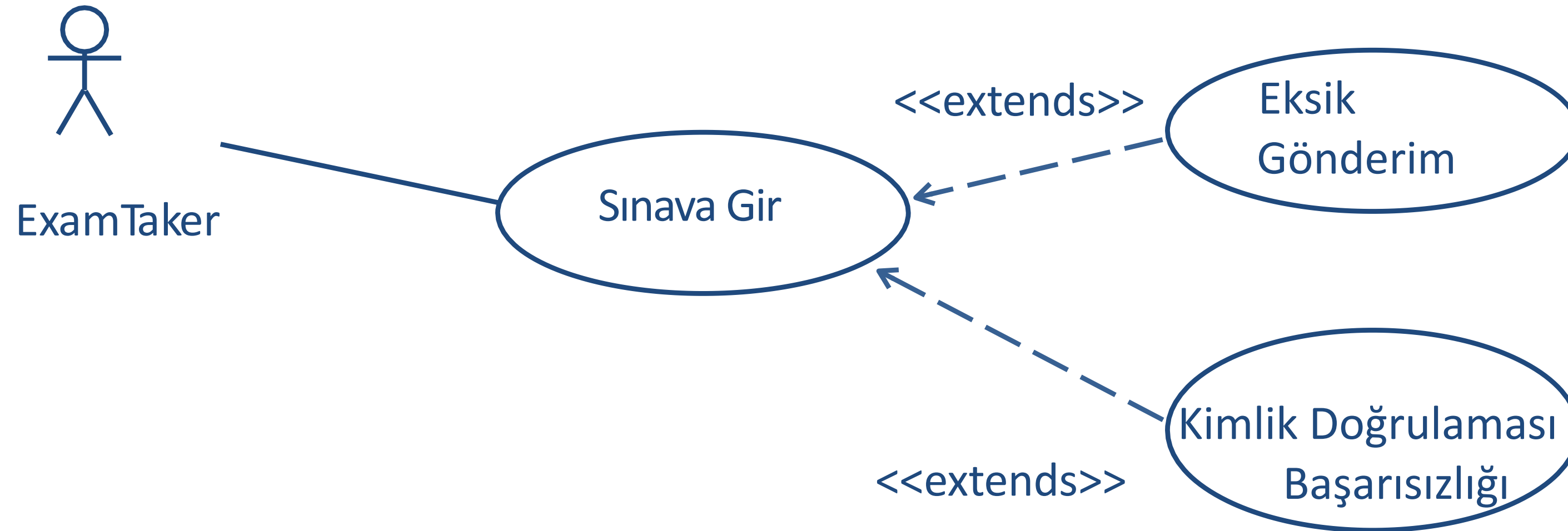
<<extends>> ilişkisi

«extends» ilişkisi

Use case'ler, diğer use case'lerden yararlanabilir

Alternatif bir akış veya özel durum daha fazla ayrıntıya ihtiyaç duyuyorsa, «extends» ilişkisi kullanılarak ayrı bir use-case olarak modellenebilir.

Use Case'ler Arasındaki İlişkiler: <<extends>>



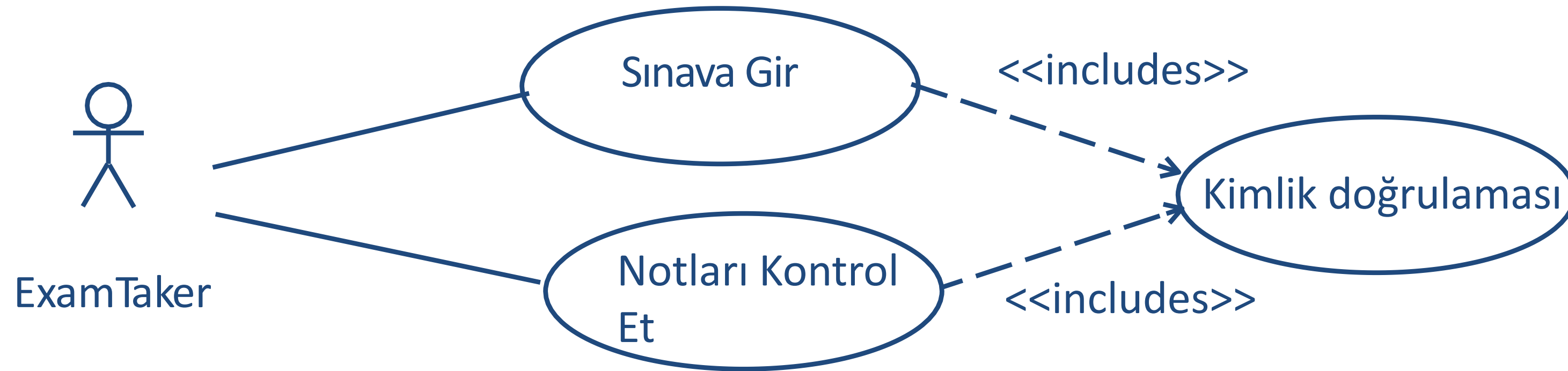
<<includes>> ilişkisi

Başka bir use-case'i kullanma

«includes» ilişkisi, bir use case'in başka bir use-case'deki adımları içermesine olanak tanır.

Bu, dahil edilen use-case başka bağlamlarda gerçekleştiğinde değerlidir. Genellikle bağımsız olarak geliştirilir.

Use-case'ler Arasındaki İlişkiler: <<includes>>



Kimlik doğrulama use-case'i başka bağlamlarda da kullanılabilir.

(ÖZET) Geliştirme döngüsündeki senaryolar ve use-case'ler

Senaryolar ve use-case'ler sezgiseldir -- Müşterilerle tartışmak kolaydır

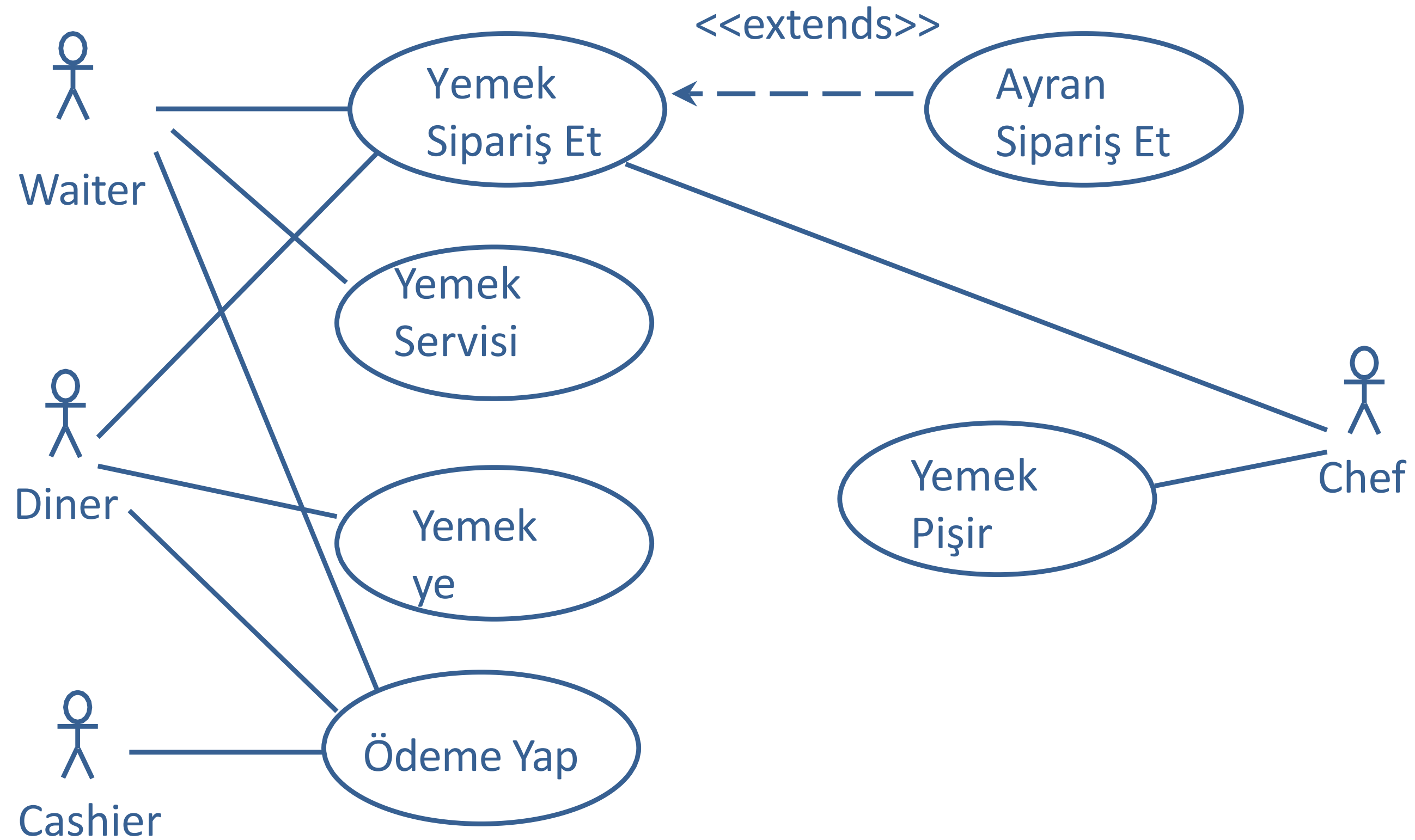
Senaryolar, gereksinim analizi için bir araçtır.

- Use-case'leri doğrulamak ve bir sistemin tasarımını kontrol etmek için kullanılırlar.
- Kabul testi için **test senaryoları** olarak kullanılabilirler.

Use-case'ler, modelleme gereksinimleri için bir araçtır.

- Bir dizi use-case, **gereksinim belirtimi** için bir çerçeve sağlayabilir.
- Use-case'ler, sistem ve program tasarımının temelini oluşturur, ancak genellikle sınıf modellerine çevrilmesi zordur.

Birkaç Aktörlü Use-Case



Bu restoran örneği, Wikipedia'daki bir use-case diyagramına dayanmaktadır.

Use Case Diyagramları

Use-case diyagramları

Use-case diyagramı, aktörler arasındaki ilişkileri ve bunların bir sistemle etkileşimlerini gösterir.

Bu etkileşimlerin **mantığını** göstermez.

Örnek Sınav Sorusu

Pizza Sipariş Sistemi

Pizza Sipariş Sistemi, bir web tarayıcısı kullanıcısının eve teslimat için pizza sipariş etmesine olanak tanır. Bir müşteri, sipariş vermek için satın alınacak ürünleri bulmak için arama yapar, ürünleri tek tek alışveriş sepetine ekler ve muhtemelen daha fazla ürün için tekrar arama yapar.

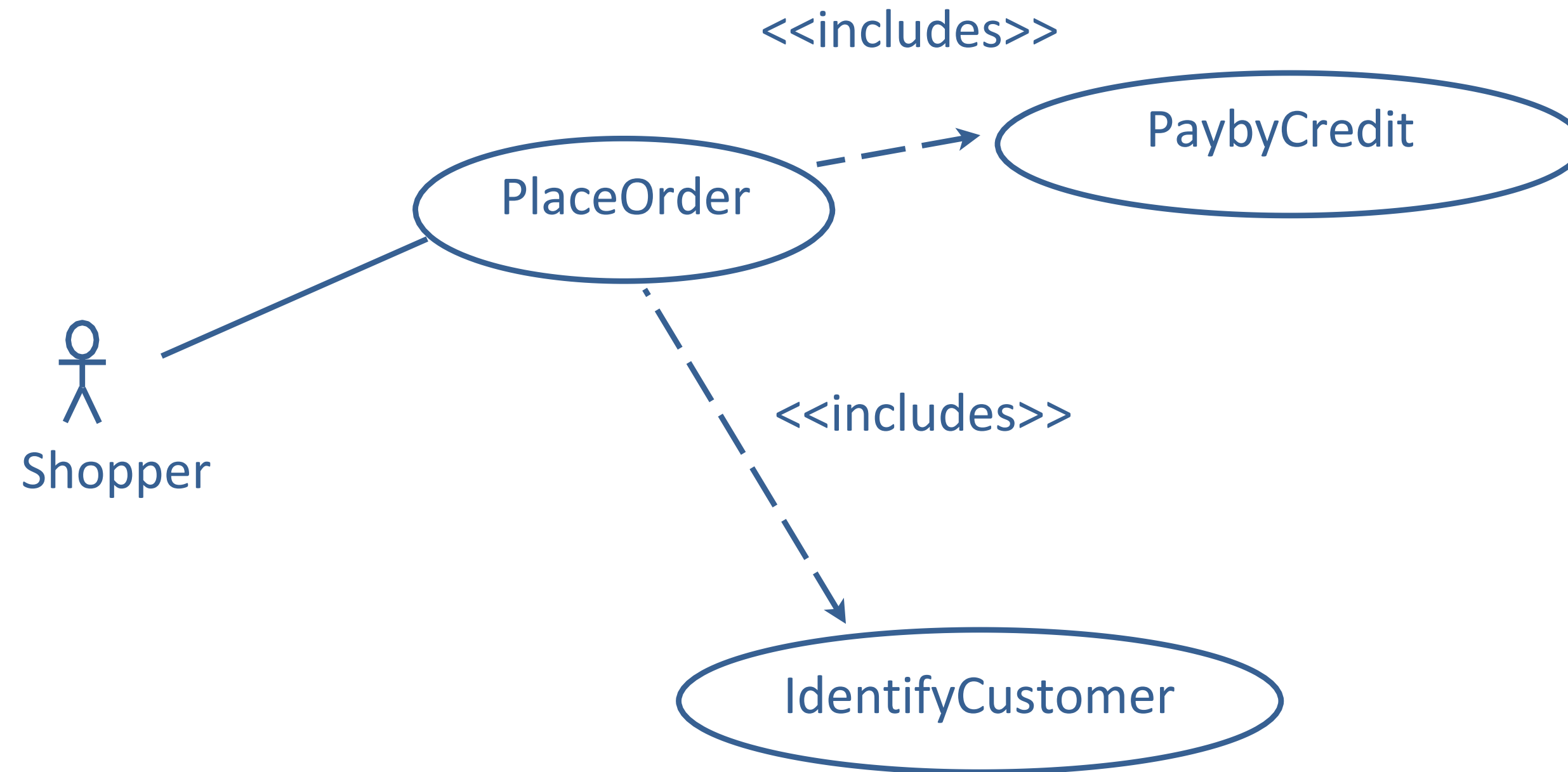
Tüm ürünler seçildiğinde, müşteri bir teslimat adresi sağlar. Nakit ödeme yapmıyorsa, müşteri kredi kartı bilgilerini de verir.

Sistem, alışveriş yapanların pizza dükkanına kaydolma seçeneğine sahiptir. Daha sonra adlarını ve adres bilgilerini kaydedebilirler, böylece her sipariş verdiklerinde bu bilgileri girmek zorunda kalmazlar.

Sipariş vermek için PlaceOrder use-case diyagramını geliştirin. Use-case diyagramı, kullanıcının kaydolmasına ve oturum açmasına olanak tanıyan IdentifyCustomer ve kredi kartı ödemelerini modelleyen PaybyCredit olmak üzere daha önce belirtilen iki use-case diyagramı ile bir ilişki göstermelidir.

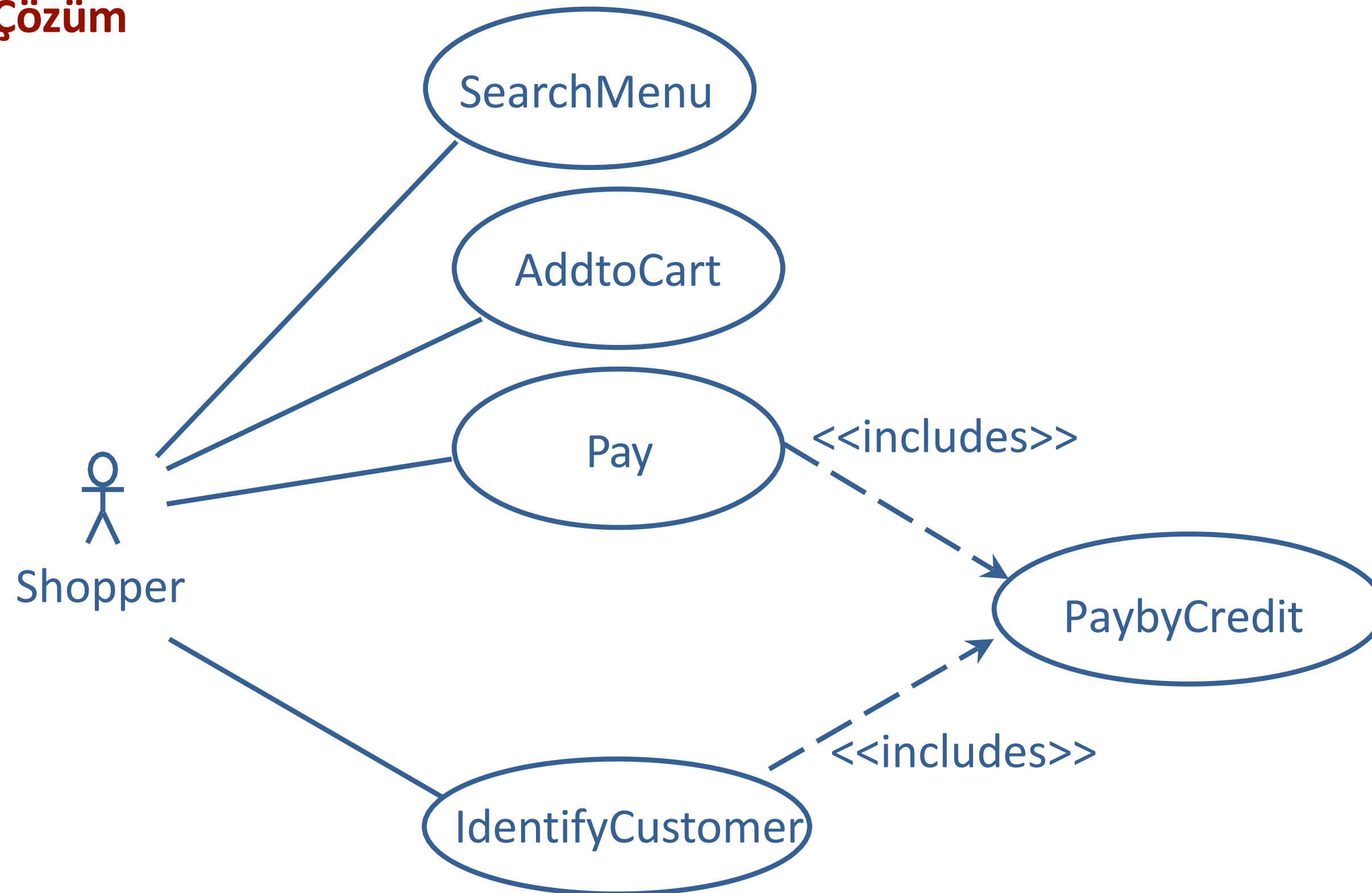
Örnek Sınav Sorusu

Doğru Çözüm



Örnek Sınav Sorusu

Yanlış Çözüm



Erciyes Üniversitesi

Bilgisayar Mühendisliği Bölümü

BZ 313 Yazılım Mühendisliği

7. Senaryolar ve Use-Caseler

Ders Sonu