



Unified Modeling Language (UML) Giriş

Dr. Öğr. Üyesi Fatih ÖZYURT

Fırat Üniversitesi Yazılım Mühendisliği Bölümü

Bölüm-5

Amaçlar



- Nesneye yönelik modelin ifade edilmesi için bir takım görsel diller kullanılmaktadır. Bunlardan en bilineni ve artık bir standart haline gelmiş olan dil, Unified Modelling Language (UML) adı ile bilinir.
- Tasarım aşaması genelde hedef sistemin çeşitli açılardan görünüşlerini ifade eden UML belgeleri ile son bulur.

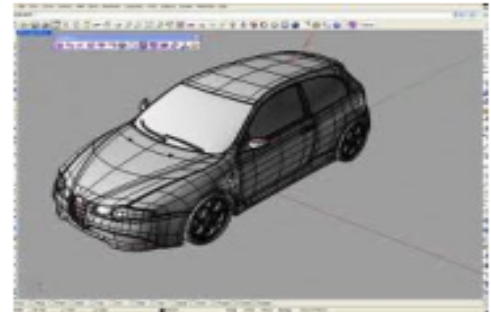
MODEL NEDİR?



- Model, gerçek dünyadaki bir olayın veya sistemin soyutlanması, basitleştirilmesi ve kavramlaştırılmasıdır.
- Model, olayı veya sistemi tanımlamaya başka bir deyişle bir örnek üretmeye yardımcı olur.
- Modeller gerçek dünyadaki örneklerinin yerini alamazlar, ancak gerçek olay veya sistemin karmaşık yapısının anlaşılabilir parçalara indirgenmesinde yararlı olurlar.



MODELLEME NEDİR?



- Modelleme bir sistemi incelemek üzere o sistemin basit bir örneği yapılması anlamına gelir. Bu örnek gerçek sistemin yardımcısı ve basitleştirilmiş bir şeklidir.
- Modelleme sistemlerin karmaşıklığını çözümlemede kullanılan en eski ve en etkin yöntemdir.

Yazılımda sistemin modellenmesi

- Yazılım projelerinde yer alan proje yöneticileri, müşteriler, analistler, tasarımcılar, programcılar, testçiler ve teknik yazarlardan her birinin eğitim düzeyleri ve alt yapıları farklıdır.
- Eğer bir sistem, tüm proje ekibinin anlayabileceği ortak bir dille **modellenirse**, çok karmaşık anlatımlar basitleşebilir ve aralarındaki iletişim çeşitli diyagramlarla maksimum düzeyde tutulabilir.

Yazılımda sistemin modellenmesi

- Yazılım sektöründe modelleme için geliştirilmiş çeşitli diller mevcuttur.
- Nesneye yönelik sistemlerin analiz ve tasarımında standart olarak kullanılan modelleme dili **UML**'dir.

UML NEDİR?



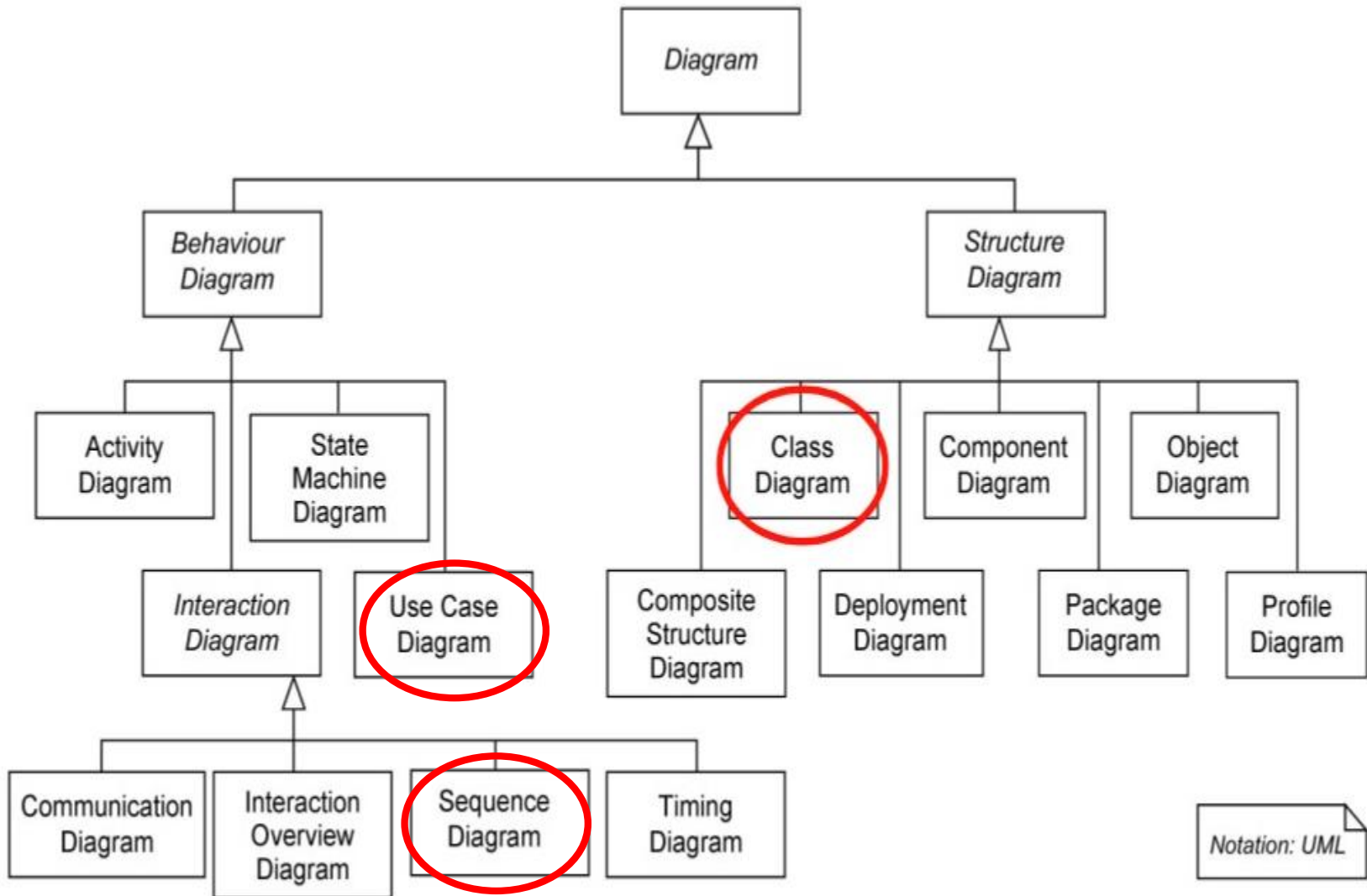
- UML, yazılımın modellenmesi ve planlanması için kullanılan standart bir dildir.
- UML yazılım mühendisliğinde nesneye yönelik sistemleri modellemede kullanılan açık standart olmuş bir görsel modelleme dilidir.
- Bir program ya da yazılım geliştirme dili değildir.
- Yazılım geliştirmenin analizden bakıma kadar tüm aşamalarında ekipler ve bireyler arasındaki iletişimin düzgün yürütülmesi için kullanılmaktadır.
- Yazılımın yaşam döngüsü içinde farklı görev gruplarının projeye ve sisteme farklı bakış açıları vardır. Bundan dolayı UML çeşitli bakış açılarını ifade eden diyagramlar içermektedir.
- Çok zengin bir dil olmasından dolayı, Yazılım Mühendisliği'nin bir çok yönden ihtiyaçlarını karşılamaktadır.

UML'NİN GELİŞİM SÜRECİ

- 1994 yılında Grady Booch (Booch) ve Jim Rumbaugh (OMT) Rational firmasının çatısı altında sahip oldukları iki yöntemi birleştirecek bir yöntem yaratmak için çalışmaya başladılar.
- Firmaya 1995 yılında Ivar Jacobson'ın (OOSE) da katılmasıyla, 3 Amigolar olarak bilinen grup, kendi yöntemlerinin güçlü yönlerini birleştirip bir sistem modelleme dili olarak UML'i geliştirdiler.
- 1997'de OMG (Object Management Group), UML'yi sahiplendi ve açık standart olarak geliştirmeye başladı.

UML'YE NEDEN GEREK VAR?

- Hataların kolaylıkla fark edilip en düşük seviyeye indirgenmesi.(Risk, zaman, maliyet)
- Yazılım üretiminde başarı oranının düşük olması.
- Yazılımda paylaşım önemlidir. Tüm ekibin aynı dili konuşabilmesi gerekmektedir.
- Sistemin tamamını basit bir dille ve görsellikle görebilmek ve tasarlayabilmek gerekli.
- Modellenmiş ve dokümante edilmiş bir yazılımın tanıtımının kolay olması.
- Yazılım kalitesini artırma.



USE CASE DİYAGRAMLARI

- Sistemin çok basit bir şekilde modellenmesini ve işlerin detayının (senaryonun) metin olarak anlatılmasını içerir.
- Aktörden gelen bazı isteklere karşı sistemin yaptığı aktiviteleri gösterir.
- Amaç
 - ✓ Sistemin içeriğini belirtmek.
 - ✓ Sistemin gereksinimlerini elde etmek.
 - ✓ Sistemin mimarisini geçerli kılmak.
- Analistler ve uzmanlar tarafından geliştirilir.

Use-Case Diagram

- Use case'ler vasıtasıyla bir sistemin fonksiyonel gereksinimleri kolayca tanımlanıp, yönetilebilir ve bu şekilde belirlenen kapsam içerisinde kalmak kolaylaşır.
- Fikirler; müşterilere, geliştirme ekibine, yöneticilere ve proje içerisindeki diğer paydaşlara hızlı, kolay ve anlaşılabilir bir şekilde aktarılabilirler.
- Tüm bu avantajlar, use case analizinin, bir sistemin fonksiyonel gereksinimlerinin tanımlanması noktasında en sık kullanılan teknik olmasını sağlamıştır.

Use-Case Diagram

- Use case diyagramı, bir uygulamanın yada sistemin kimler tarafından kullanıldığını ve söz konusu kullanıcıların bu uygulama yada sistemle neler yapabildiklerini özetlemek amacıyla kullanılan bir UML diyagramıdır.
- Başka bir deyişle, bu diyagramın amacı, sistemin hangi fonksiyonlarının hangi aktörler tarafından gerçekleştirileceğini ve use case'ler arasındaki ilişkileri göstermektir.

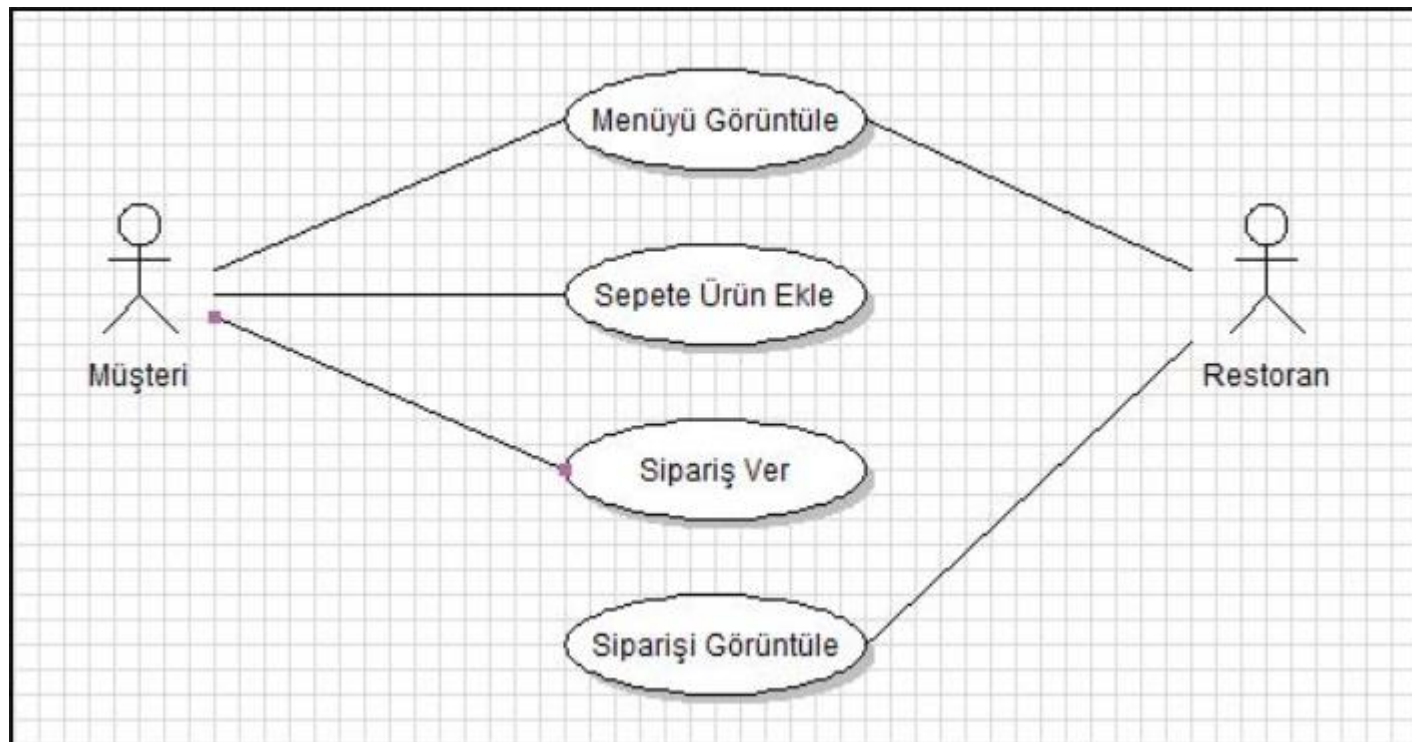
Dolayısıyla bir use case diyagramının dört temel bileşeni vardır;

1. Use case olarak adlandırdığımız sistemsel fonksiyonlar
2. Aktörler
3. Use case'ler ve aktörler arasındaki ilişkiler
4. Sistem

Use-Case Diagram

- Sistemsel gereksinimler analiz edildikten sonra elde edilen fonksiyonlar use case'ler şeklinde ifade edilir.
- Aslında, use case'ler **fonksiyonel gereksinimlerin** belirli bir formata bağlı kalınarak yazılmasından başka birşey değildir.
- Use case'ler sistemin fonksiyonel olmayan gereksinimleriyle ilgili bilgi içermezler.

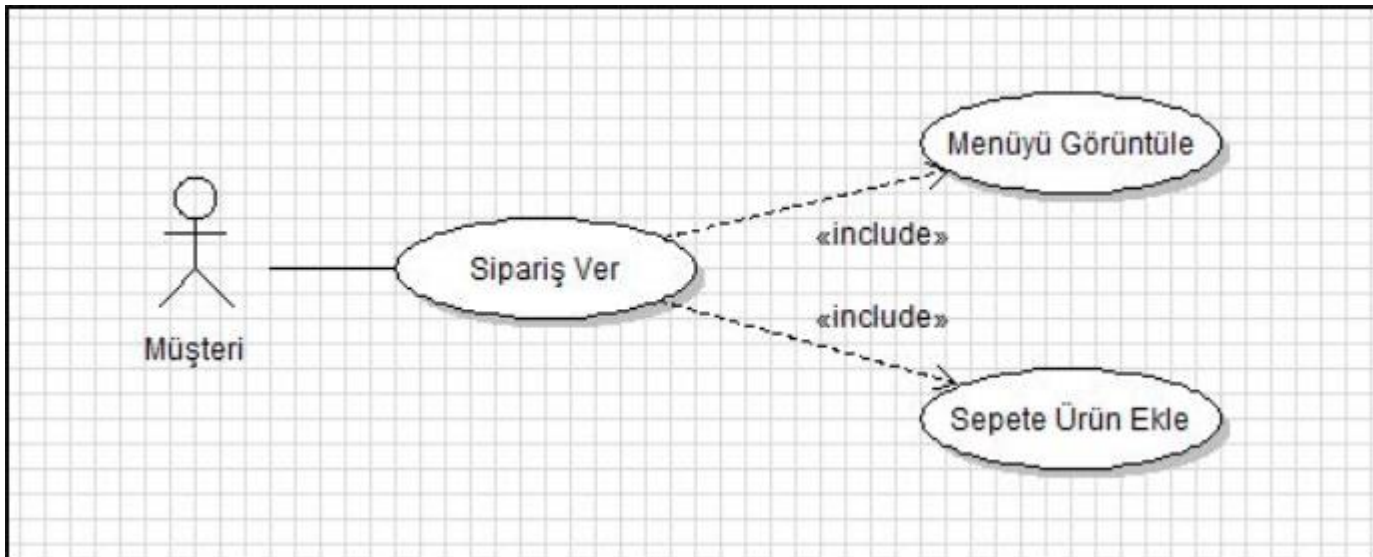
Use-Case Diagram



Use-Case Diagram

Include İlişkisi

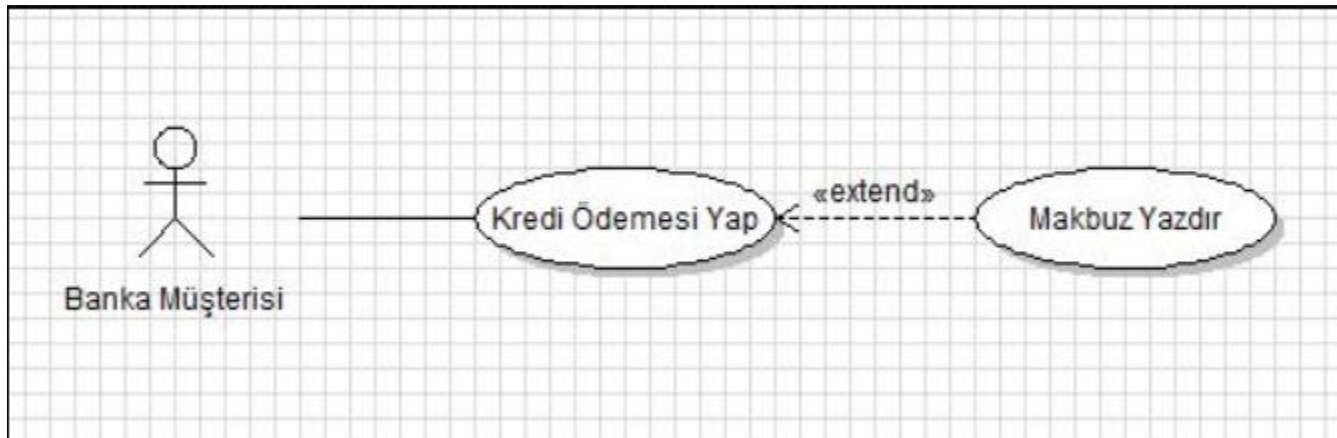
Bazı durumlarda bir use case'in tamamlanabilmesi için, başka use case'lerin işin içerisine katılması zorunludur. Yani söz konusu use case, bir takım başka use case'ler olmaksızın tamamlanamaz.



Use-Case Diagram

Extend İlişkisi

İki use case arasında opsiyonel olarak varolabilecek bir durumu ifade etmek için kullanılır.



Use Case ID:	UC-1.1.1		
Use Case Adı:	Sepete Ürün Ekle		
Oluşturan:	Emre Alagöz	Son Güncelleyen:	
Oluşturulduğu Tarih:	25.07.2013	Son Güncellenme Tarihi:	
Aktör:	Müşteri		
Tanım:	Bu use case, Dominos Pizza'ya ait "Orta Boy Pizza + Litrelik İçecek" başlıklı ürünün, alışveriş sepetine eklenmesine ilişkin senaryoları ve diğer bir takım ayrıntıları içermektedir.		
Ön Koşullar:	<ol style="list-style-type: none"> 1. Müşteri, sisteme giriş yapmış olmalıdır. 2. Müşteri, Dominos Pizza'nın ürünlerinin yer aldığı menü sayfasını görüntülemiş olmalıdır. 		
Son Koşullar:	Söz konusu ürün müşterinin sepetine eklenmiş ve müşteri ilgili menü sayfasına yönlendirilmiştir.		
Ana Başarı Senaryosu:	<ol style="list-style-type: none"> 1. Bu use case, müşterinin "Orta Boy Pizza + Litrelik İçecek" adlı ürünün linkine tıklamasıyla başlar. 2. Sistem "Sepete Ekle" adlı diyalog kutusunu görüntüler. 3. Sistem, söz konusu ürünün fiyatını veritabanından çeker ve ilgili diyalog kutusu üzerinde görüntüler. 4. Kullanıcı, "Adet" adlı drop down menüden sepete eklemek istediği ürün sayısını seçer. 5. Kullanıcı, "Pizza Seçimi" adlı drop down menüye tıklar. 6. Sistem, "Orta Boy Pizza + Litrelik İçecek" adlı ürünün kapsamı içerisinde yer alan pizzaları veritabanından çeker ve listeler. 7. Kullanıcı, listedeki pizzalardan birini seçer. 8. Sistem, seçilen pizzaya bağlı olarak, pizza içerisindeki malezemeleri görüntüler. 9. Kullanıcı, "Hamur Tipi" adlı drop down menüye tıklar. 10. Sistem, Dominos Pizza tarafından daha önceden sisteme kaydedilmiş hamur tiplerini listeler. 11. Kullanıcı, listedeki hamur tiplerinden birini seçer. 12. Sistem, ürünün fiyatına seçilen hamur tipinin fiyatını ekler ve sonucu hesaplayarak ilgili diyalog kutusunda görüntüler. 13. Kullanıcı, "İçecek" adlı drop down menüye tıklar. 14. Sistem, Dominos Pizza tarafından, seçilen ürüne bağlı olarak, daha önceden sisteme kaydedilmiş içecekleri listeler. 15. Kullanıcı, listedeki içeceklerden birini seçer. 16. Sistem, ürünün fiyatına seçilen içeceğin fiyatını ekler ve sonucu hesaplayarak ilgili diyalog kutusunda görüntüler. 17. Kullanıcı, "Sepete Ekle" butonuna basar. 18. Sistem, yapılan seçimleri veri tabanına kaydederek ürünün kullanıcının sepetine ekler. 		
Alternatif Senaryolar:	<ol style="list-style-type: none"> 7a. Kullanıcı, herhangi bir pizza seçimi yapmamıştır. 1. Sistem, "Pizza Seçimi alanı için seçim yapmalısınız" şeklinde bir uyarı mesajı görüntüler. 11a. Kullanıcı, herhangi bir hamur tipi seçimi yapmamıştır. 1. Sistem, "Hamur Tipi alanı için seçim yapmalısınız" şeklinde bir uyarı mesajı görüntüler. 15a. Kullanıcı, herhangi bir hamur tipi seçimi yapmamıştır. 1. Sistem, "İçecek alanı için seçim yapmalısınız" şeklinde bir uyarı mesajı görüntüler. 17a. Kullanıcı, "Menüye Dön" butonuna yada diyalog kutusunun sağ üst köşesindeki "x" ikonun basar. 1. Sistem, "Sepete Ekle" adlı diyalog kutusunu kapatır. 		

Use-Case Diagram

Use Case Senaryolarının Avantajları

- Sistemin erişimini, sınırlarını belirler. Böylelikle geliştirilecek sistemin boyutunu ve karmaşıklığını kafamızda daha rahat canlandırabiliriz.
- Kullanım senaryoları isteklerin çözümlenmesine çok benzemektedir: daha nettir ve tamdır.
- Basit oluşu müşteri ile geliştirme ekibi arasında iletişime olanak tanır. Geliştirme aşaması için temel oluşturur.
- Sistem testi için temel oluşturur.
- Kullanıcı klavuzu hazırlamaya yardımcı olur.

Use-Case Diagram

Use Case Diyagram Oluşturulurken Nelere Dikkat Edilmeli?

- Aktörler tespit edilirken sistemde çok sık adı geçen özneler kullanılmalı
Özneler çıkarılırken rollere göre gruplandırılmalıdır.
- Aktör sayısının en aza indirilmesine özen gösterilmelidir.
- Use Case diyagramları oluştururken cümleler arasında geçen kilit fiiller önemlidir.
- Fiillerin çok sık geçmesi ve bunların projenin ana hedeflerine uygun, sisteme değer katan iş süreçlerini içermesine dikkat edilmelidir.

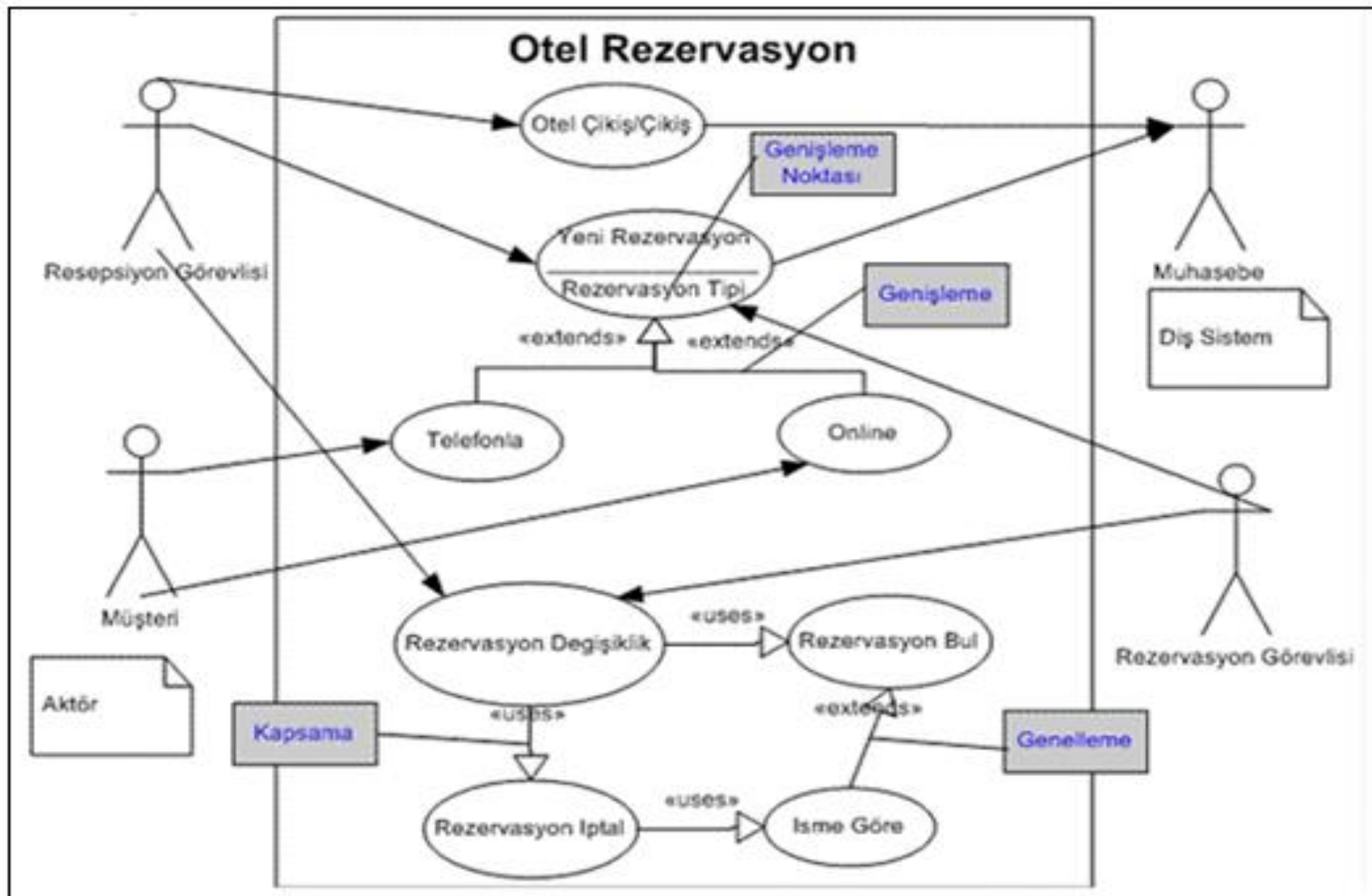
Use-Case Diagram

Use Case Diyagram Oluşturulurken Nelere Dikkat Edilmeli?

Zorunlu olmadıkça diğer ilişkileri kullanmaktan kaçınılmalıdır.

Use Case diyagramlarının kullanım amacını unutmamak gerekir.

Çok fazla Use Case diyagramı çizmekten ziyade kolaylıkla anlaşılabilir, kullanıcılar için değer taşıyabilen çekirdek iş süreçlerini anlatan diyagramlar olmalıdır.



Referanslar

1. <http://is-sanalisti.blogspot.com/2013/07/use-case-analizi-1-use-case-diyagramlar.html>

2. Özlem AYDIN ders notları

Sorularınız

