11. Hafta

YMÜ225
Yazılım Gereksinim Analizi

Dr. Öğr. Üyesi Feyza Altunbey Özbay

İçerik

- Sistem Modelleme
- UML
- Use Case Diyagramı
- Sıra Diyagramları

Model Nedir?

- Gerçek dünyadaki bir olayın veya sistemin soyutlanması, basitleştirilmesi ve kavramsallaştırılmasıdır.
- Model, olayı veya sistemi tanımlaya, diğer bir ifade ile bir örnek türetmeye yardımcı olur.
- Modeller gerçek dünyadaki örneklerinin yerini alamazlar. Fakat gerçek olay veya sistemin karmaşık yapısının anlaşılır parçalara indirgenmesine yardımcı olurlar.

Modelleme Nedir?

- Bir sistemi incelemek üzere o sistemin basit bir örneğin yapılmasıdır.
- Gerçek sistemin yardımcısı ve basitleştirilmiş halidir.
- **Sistemlerin karmaşıklığını çözümlemede kullanılan en eski ve en etkin yöntem modellemedir.**
- Sistem modellenirken farklı bakış açıları ile değerlendirilir.
- Modellemeyi yapan kişi sistem özelliklerinden o an ilgilendiklerini ön plana çıkarırken diğerlerini göz ardı edebilir.
- Sonuç olarak oluşturulan soyut yapı sistemin ilgilenilen özelliklerini içeren bir model olur.
- Hiçbir model gerçek sistemin özelliklerini tümüyle İÇERMEZ.

Yazılım Sistemin Modellenmesi

- Yazılım projelerinde yer alan proje yöneticileri, müşteriler, analistler, tasarımcılar, programcılar, testçiler ve teknik yazarlardan her birinin eğitim düzeyi ve at yapıları farklıdır.
- Eğer sistem, tüm proje ekibinin anlayabileceği ortak bir dile modellenirse, karmaşık anlatımlar basitleşir ve aralarındaki iletişim çeşitli diyagramlarla en üst düzeye getirilmiş olur.

Sistem Modelleme

- Bir sistemin soyut modellerinin geliştirilmesi sürecidir. Bir model sistemin soyutlanmış hali olarak görülmelidir.
- Sistem modelleme günümüzde bir sistemin Birleşik Modelleme Dili (UML) diyagram tiplerinin kullanılmasına dayanan bir çeşit grafiksel gösterim ile temsil edilmesidir.
- Modeller gereksinim mühendisliği sürecinde;
  - Sistemin detaylı gereksinimlerini saptamak amacıyla,
  - Tasarım sürecinde sistemi geliştirecek olan mühendislere, gerçekleştirmekte oldukları sistemi betimlemek amacıyla,
  - Gerçekleştirim sonrasında da sistemin yapısı ve işlevini belgelemek amacıyla kullanılır.

Sistem Modelleme

Sistemi değişik açılardan göstermek için farklı modeller oluşturulabilir. Örneğin;

- Sistemin bağlam ve çevresini modellediğimiz dış perspektif.
- Bir sistemin çevresi ile olan etkileşimi veya sistemin bileşenleri arasındaki etkileşimi modellediğimiz iç perspektif.
- Bir sistemin organizasyonu veya sistem tarafından işlenen verinin yapısını modellediğimiz bir yapısal perspektif.
- Sistemin dinamik davranışını ve olaylara tepkisini modellediğimiz bir davranışsal perspektif.

UML Nedir?

- UML, yazılımın modellenmesi ve planlanması için kullanılan standart bir dildir.
- Yazılım mühendisliğinde nesneye yönelik sistemleri modellemede kullanılan açık standart olmuş bir görsel modelleme dilidir.
- Yazılım geliştirmenin analizden bakıma kadar tüm aşamalarında ekipler ve bireyler arasındaki iletişimin düzgün yürütülmesi için kullanılmaktadır.
- Yazılım yaşam döngüsü içinde farklı görev gruplarının projeye ve sisteme farklı bakış açıları bulunmaktadır. Bundan dolayı UML çeşitli bakış açılarını ifade eden diyagramlar içermektedir.
- UML 1997 yılında yazılımın, diyagram şeklinde ifade edilmesi için bir standartlar komitesi tarafından oluşturuldu.

UML

UML kullanımının yazılım sürecine katkıları:

- Yazılımın geniş bir analizi ve tasarımı yapılmış olacağından kodlama işlemi daha kolay ve doğru olur.
- Hataların en aza inmesine yardımcı olur.
- Geliştirme ekibi arasındaki iletişimi kolaylaştırır.
- Tekrar kullanılabilir kod sayısını artırır.
- Tüm tasarım kararları kod yazmadan verilir.
- Yazılım geliştirme sürecinin tamamını kapsar.
- “resmin tamamını” görmeyi sağlar.

UML

UML sistemlerin incelenmesi ve tasarımım için aşağıdaki modellerin kullanılmasına olanak sağlar.

- Kullanıcı Senaryoları (Kullanıcı gereksinimlerinin modellenmesi için kullanılır.)
- Sınıf Modeli (Yazılımın durağan yapısını modellenmesi için kullanılır.)
- Etkileşim Modeli (Senaryoların ve ileti akışının modellenmesi için kullanılır.)
- Durum Modeli (Nesnelerin devingen davranışlarının modellenmesi için kullanılır.)
- Gerçekleştirim Modeli (Gerçekleştirime dönük modelleme için kullanılır.)
- Yaygınlaştırma Modeli (Bileşenlerin mimari yapı üzerinde nasıl dağıtıldığının modellenmesi için kullanılır.)

UML

Verilen modellerden her birinin kurulması için aşağıdaki UML diyagramlarından bir ya da birkaçının çizilmesi gerekir.

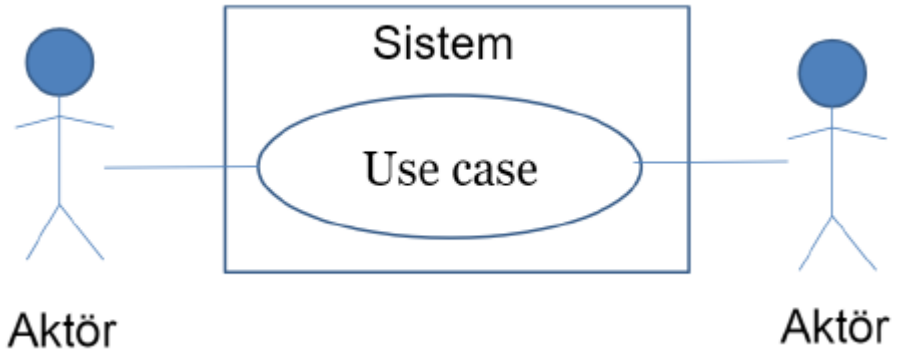
1. Kullanıcı Senaryosu Diyagramları (Use-case diagrams)
2. Sınıf Diyagramları (Class diagrams)
3. Nesne Diyagramları (Object diagrams)
4. Ardıl Etkileşim Diyagramları (Sequence diagrams)
5. İşbirliği Diyagramları (Collaboration diagrams)
6. Durum Diyagramları (State diagrams)
7. Etkinlik Diyagramları (Activity diagrams)
8. Bileşen Diyagramları (Component diagrams)
9. Yaygınlaştırma Diyagramları (Deployment diagrams)

Kullanım Durum (Use-Case) Diyagramı

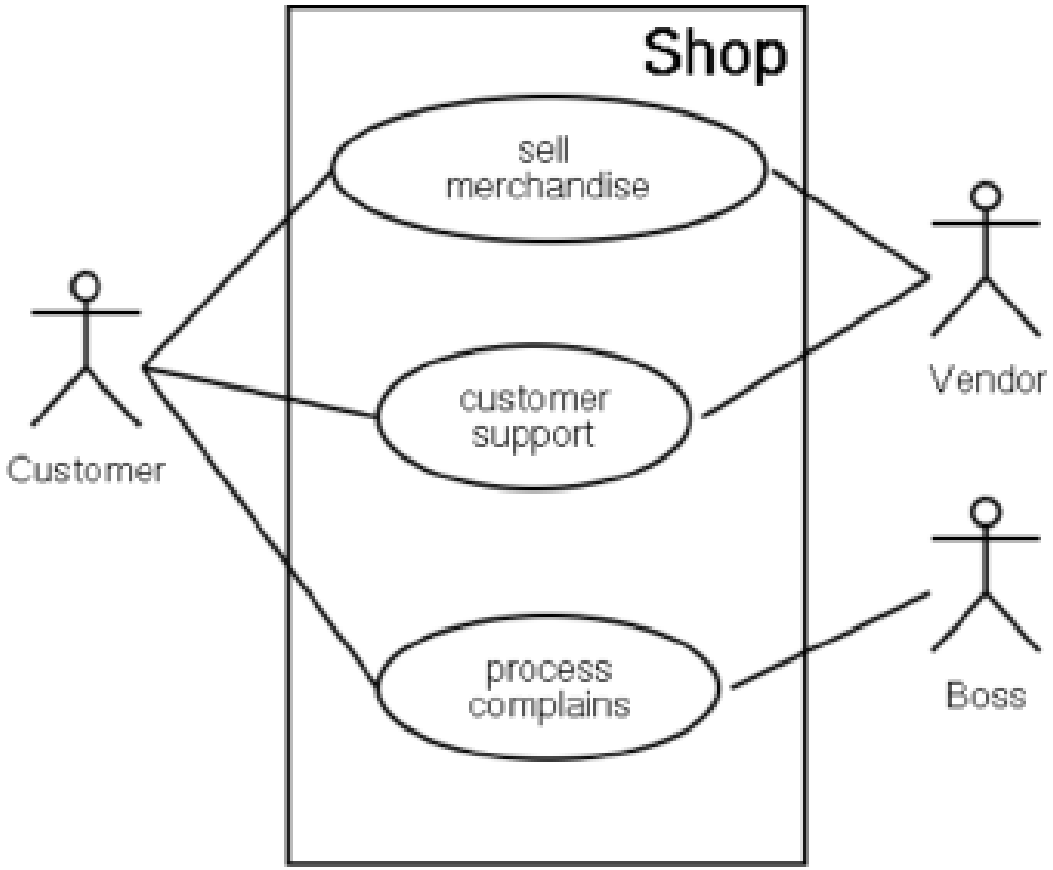
- Analiz aşaması projeler için hayati önem taşır. İyi bir analizden geçmemiş projelerin başarı şansı oldukça düşüktür.
- Bir sistemi modellemek, gereksinimleri ortaya koymak ve bunun sonucunda istenilen ürünü tam olarak gerçekleştirmek için en önemli nokta sistemde gerçekleştirilecek olan dinamik davranışları yakalamaktır.
- Sistemin tasarlanmasında ihtiyaçları ayrıntılı olarak açıklığa kavuşturmak için bahsedilen bu dinamik davranış, sistemin çalıştırıldığındaki davranışını ifade eder.
- Kullanım durumu (Use-Case -- UC) diyagramı UML’de dinamik davranışları modellemek için kullanılan diyagramlardan biridir.

Kullanım Durum (Use-Case) Diyagramı

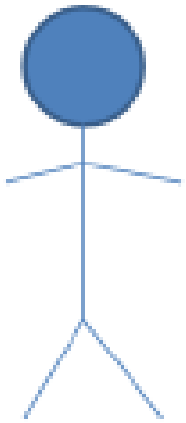
- Use case diyagramları, kullanıcı-sistem etkileşimini modellemek için kullanılan bir yöntemdir. Bunlar sistemin uç kullanıcı bakış açısından modellenmesini sağlar. Bu nedenle amaç odaklı olarak nitelendirilir.
- Use case diyagramlarında uç kullanıcı ve sistem olmak üzere iki katılımcı bulunur.
- Aktörden gelen isteklere karşı sistemin yaptığı aktiviteleri gösterir.



Basit bir Use case diyagramı örneği



Use Case Diyagramı Bileşenleri



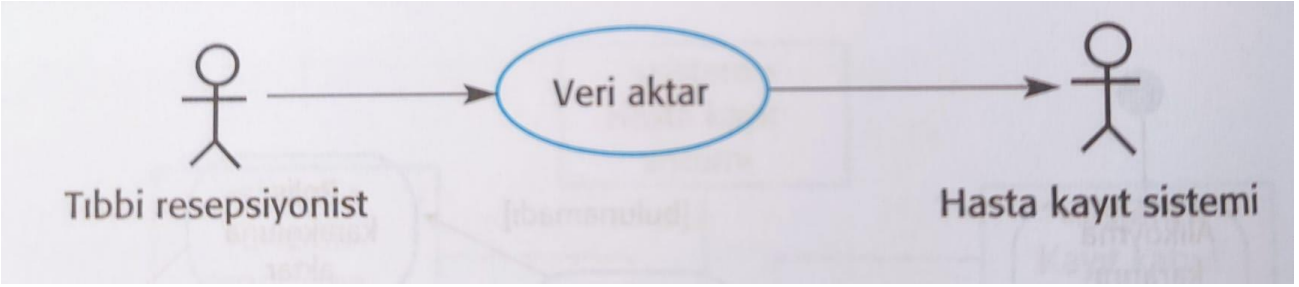
<u>Aktör:</u>

Sistemin kullanıcılarıdır.

Genellikle belirli bir rol ifade ederler.

Diğer aktörlerle bağlantılı olabilirler, bu bağlantı bir ok ile gösterilir.

Sistemin sınırları dışında gösterilir.



ZihinSaS sistemi: Veri aktar	
Aktörler	Tıbbi resepsiyonist, Hasta kayıt sistemi (HKS)
Betimleme	Bir resepsiyonist ZihinSaS sisteminden bir sağlık otoritesince yürütülen bir hasta kayıt sistemine veri aktarabilir. Aktarılan bilgi güncellenmiş kişisel bilgiler olabilir (adres,telefon numarası vs.) veya hastaya ilişkin teşhis ve tedavinin bir özeti olabilir.
Veri	Hastanın kişisel bilgileri, tedavi özeti
Uyaran	Tıbbi resepsiyonist tarafından verilen kullanıcı komutu
Tepki	HKS'nin güncellendiğinin teyidi
Yorumlar	Resepsiyonist hasta bilgisine ve HKS'ye erişmek için gerekli güvenlik izinlerine sahip olmalıdır.

Use Case Diyagramı Bileşenleri

<u>Use Case:</u>

Sistemin destekleyeceği işleri gösterir.

Sistem fonksiyonelliğinin büyük bir parçasını gösterir.

Farklı bir use case ile genişletilebilir.

Farklı bir use içerebilir.

Sistemin sınırları içerisinde gösterilir.

Use Case Diyagramı Bileşenleri

<u>Sistem sınırı:</u>

- İçerisinde sistemin adı yazılıdır.
- Sistemin kapsamını gösterir.

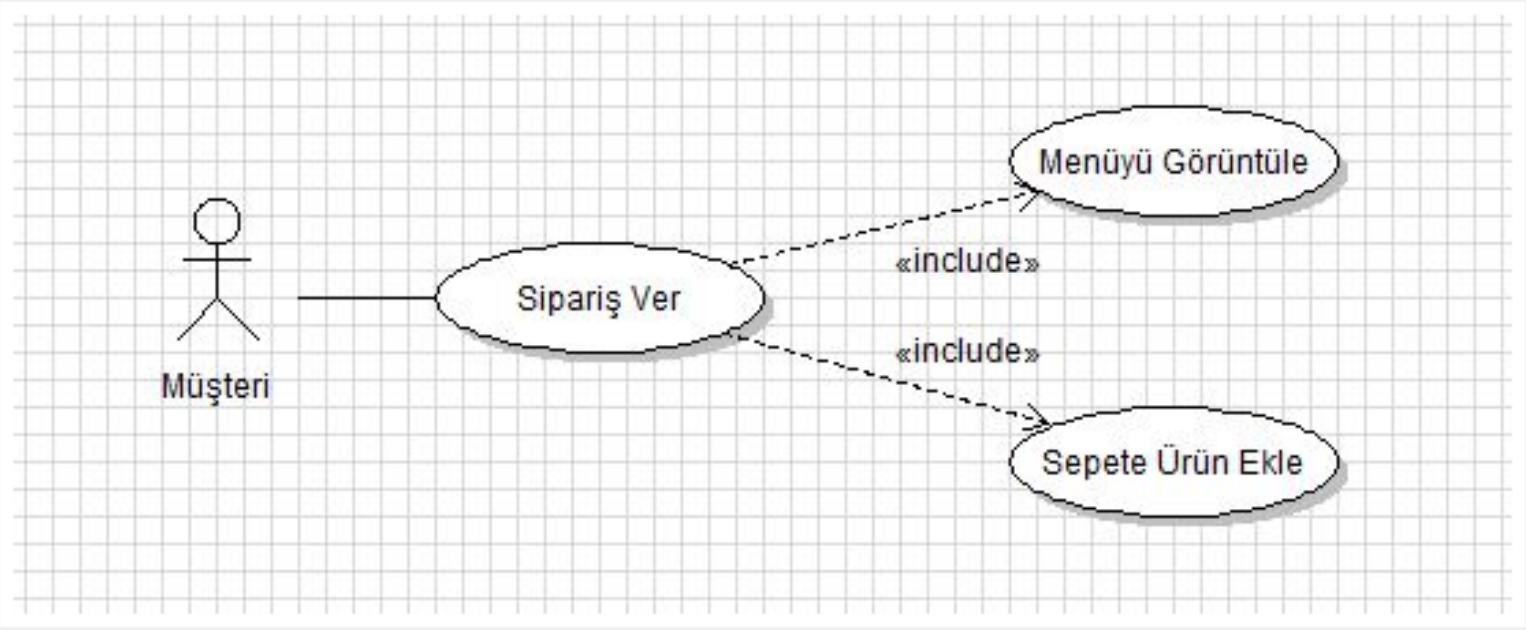
<u>Bağıntı İlişkisi:</u>

Aktör ve use case'ler arasındaki bağıntıyı gösterir.

Use Case Diyagramı Bileşenleri

<u>Include (içerme) ilişkisi:</u> <<include>>

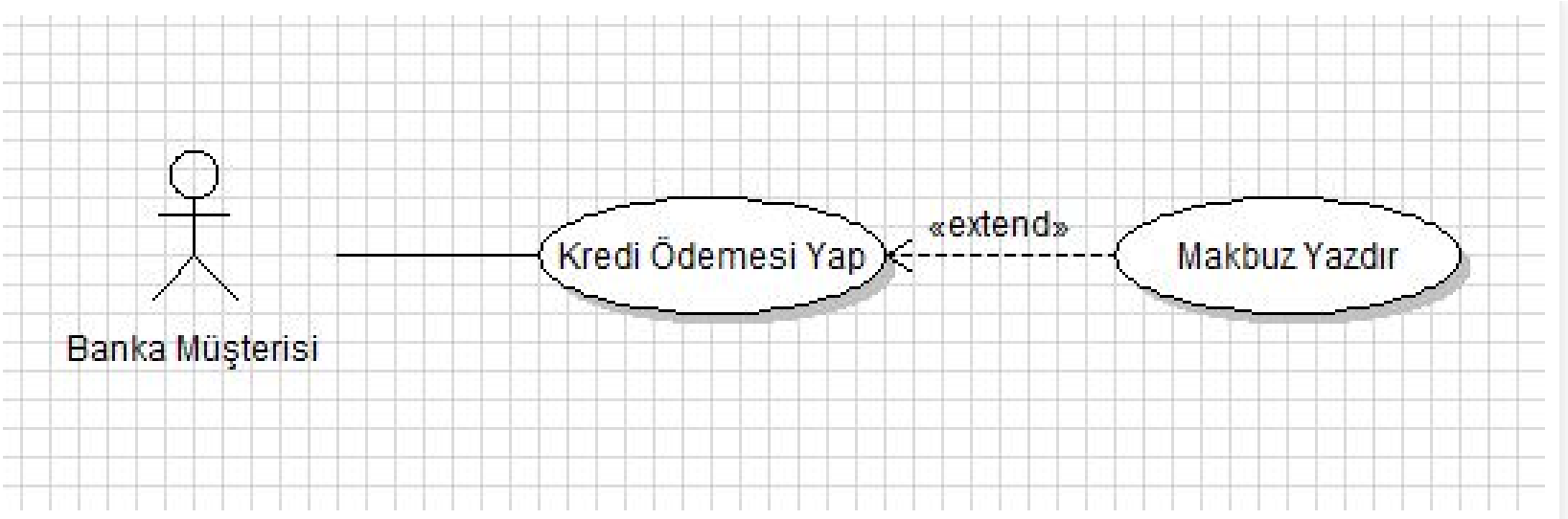
Bazı durumlarda bir use case'in tamamlanabilmesi için, başka use case'lerin işin içerisine katılması zorunludur. Yani söz konusu use case, bir takım başka use case'ler olmaksızın tamamlanamaz.



Use Case Diyagramı Bileşenleri

<u>Extend (eklenti) İlişkisi:</u> <<extend>> <----------

İki use case arasında opsiyonel olarak var olabilecek bir durumu ifade etmek için kullanılır. Bu iki use case, “base use case” ve “extending use case” olarak adlandırılır.



!!!

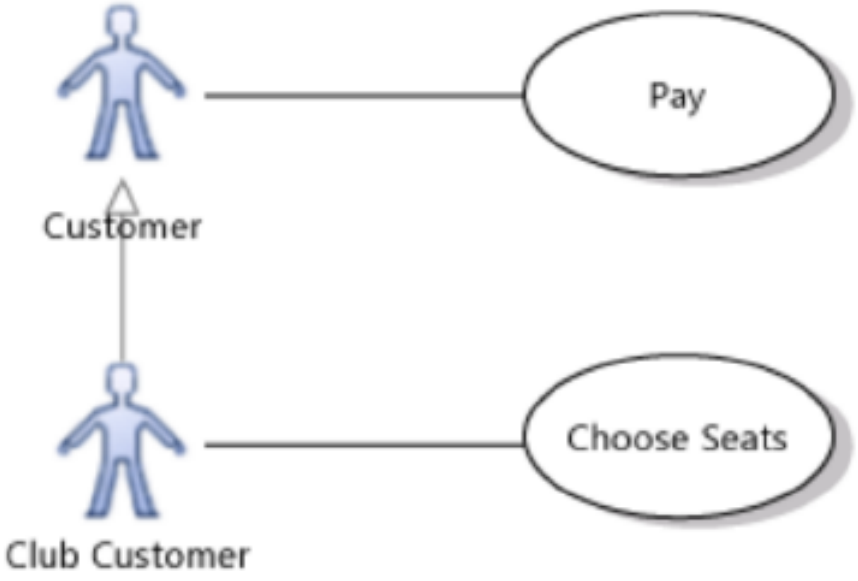
- Diyagramın okunabilirliğini azalttığı için, Extend ilişki türü, gerçekten ihtiyaç duyulmadıkça kullanılmamalıdır.
- Diyagram üzerindeki okun yönü bize use case’ler arasındaki bağımlılığın yönünü gösterir.
- Extending use case’in, base use case’e bağlı olduğu unutulmamalıdır. Yani bir extending use case, base use case olmadan tek başına bir anlam ifade etmez.

Use Case Diyagramı Bileşenleri

<u>Genelleme İlişkisi:</u>

İki use-case veya iki aktör arasındaki kalıtım ilişkisidir.

Yani özelleşmiş bir use case veya aktör ile daha geneli arasındaki ilişkidir.



ATM Uygulaması

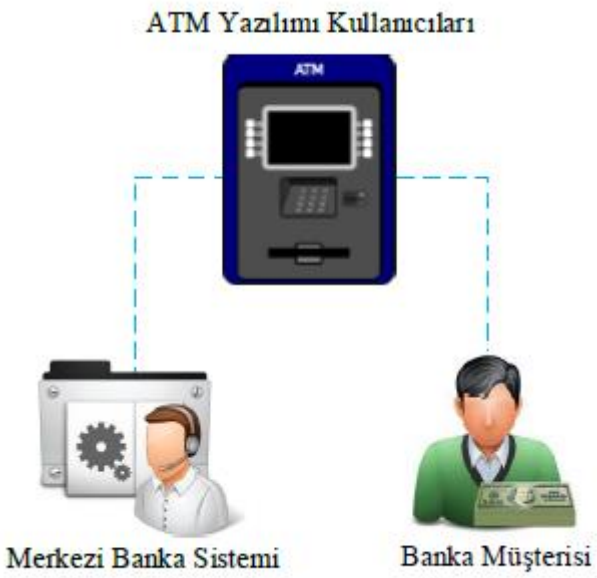
Bir bankanın ATM cihazı için yazılım geliştirilecektir. ATM, banka kartı olan müşterilerin hesaplarından para çekmelerine, hesaplarına para yatırmalarına ve hesapları arasında para transferi yapmalarına olanak sağlayacaktır. ATM, banka müşterisi ve hesapları ile ilgili bilgileri, gerektiğinde merkezi banka sisteminden alacaktır.

ATM uygulama yazılımının kullanıcıları:

Aktörler

- Banka müşterisi
- Merkezi Banka Sistemi

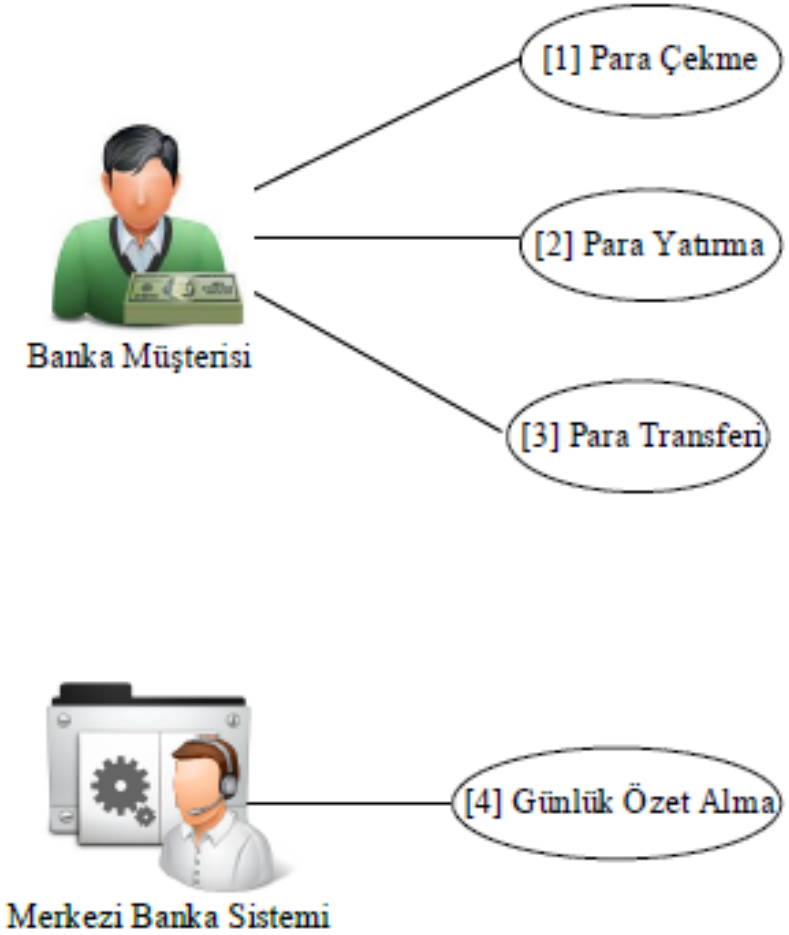
ATM Uygulaması

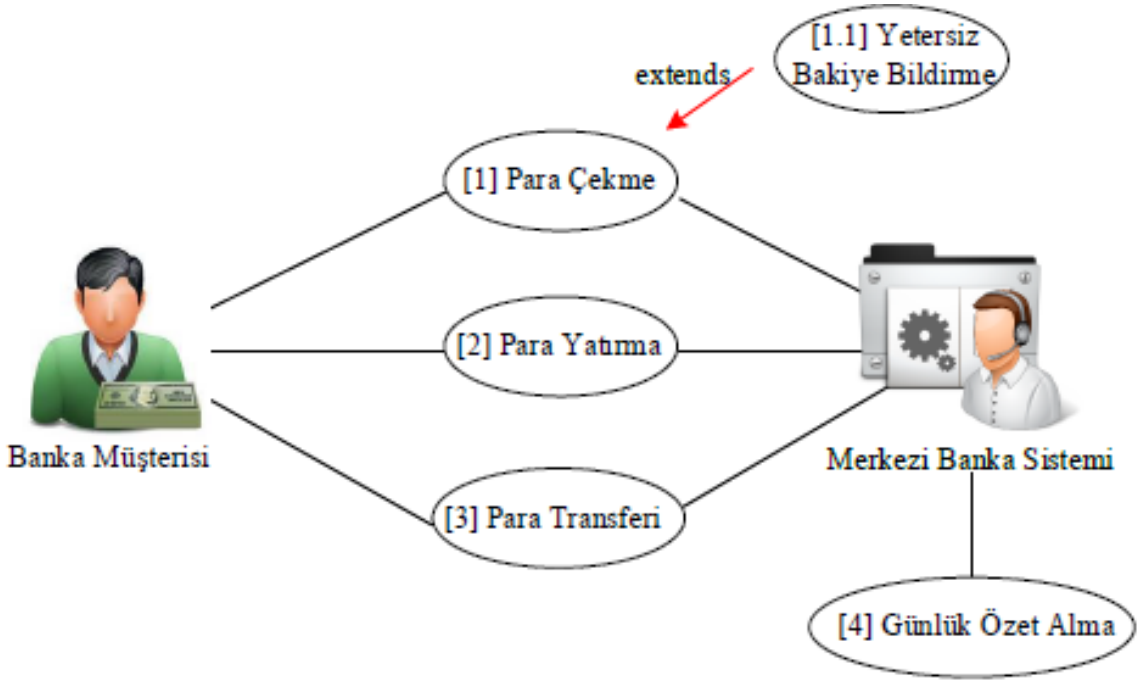


Belirlenen aktörler ATM'den ne istiyorlar?

- **Aktör: Banka müşterisi**
  - ✓ Para çekme
  - ✓ Para yatırma
  - ✓ Para transferi
- **Aktör: Merkezi Banka Sistemi**
  - ✓ Günlük özet alma

ATM Uygulaması



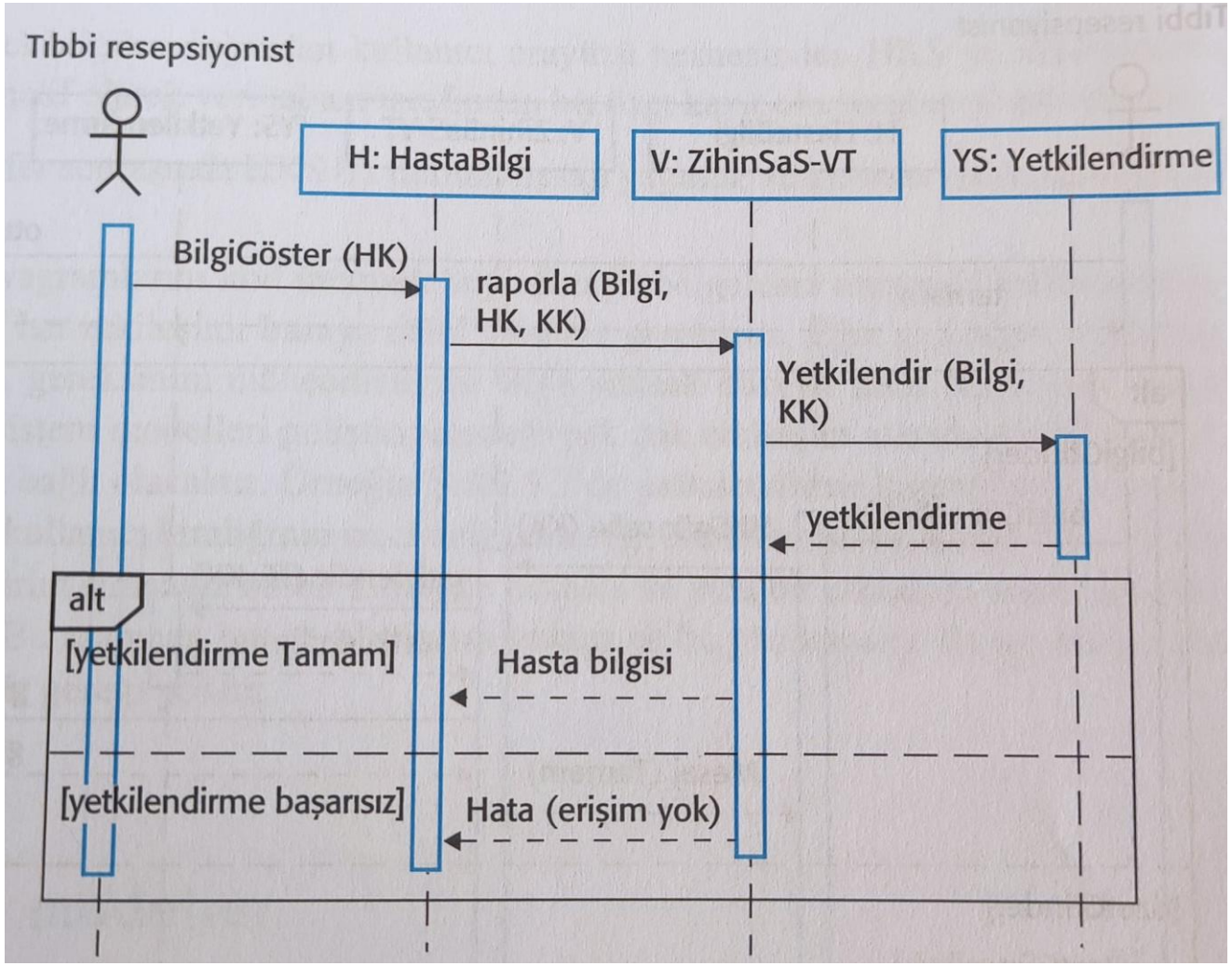


Sıra Diyagramları

Sıra diyagramları genellikle bir sistemin aktörleri ile nesneleri arasındaki ya da nesnelerin kendi aralarındaki etkileşimleri modellemek amacıyla kullanılır.

Tıbbi resepsiyonistin hasta bilgilerinin görmesini sağlayan «Hasta bilgilerini göster»kullanım durumunun etkileşimi gösterilmiştir. Etkileşime giren nesneler ve aktörler her birinden aşağı inen kesikli çizgiler olmak üzere diyagramın yukarı bölümünde listelenir.

Metinsel açıklama eklenmiş oklar nesneler arasındaki etkileşimi göstermektedir. Kesikli çizgiler üzerindeki dikdörtgenler ilgili nesnenin yaşam çizgisini belirtmektedir. Etkileşim sırasın yukarıdan aşağı okunur. Oklar üzerindeki açıklamalar nesnelere yapılan çağrıları, bu çağrıların parametrelerini ve döndürülen değerleri göstermektedir.



Tıbbi resepsiyonist HastaBilgi sınıfının H örneğindeki BilgiGöster metodunu hastanın kimliğini belirten HK parametresi ile tetikler. H bir kullanıcı arayüzü nesnesidir ve hasta bilgilerini içermek üzere tasarlanmış formu ekranda gösterir.

H örneği güvenlik kontörlü amacıyla resepsiyoniste kimliğini de vererek gereken bilgiyi sağlamak üzere veri tabanını çağırır.

Veri tabanı yetkilendirme sisteminden resepsiyonistin bu işlem için yetkili olup olmadığını kontrol eder.

Eğer yetkili ise hasta bilgisi döndürülür ve kullanıcı ekranında bir formda gösterilir. Eğer yetkilendirme başarısız olursa bir hata mesajı döndürülür. Sol üst köşesinde alt olarak işaretlenmiş kutu bir seçim kutusudur ve içerdiği mesajların sadece birinin çalıştırılacağının ifadesidir. Seçenekler arasında seçim yapılmasını sağlayan mantıksal şartlar köşeli parantez içinde verilmiştir.