

Nesne Yönelimli Analiz ve Tasarım

Analiz Aşamasi

Gereksinimlerin Belirlenmesi

UML (Unified Modelling Language) Birleştirilmiş Modelleme Dili

Kullanım Durum (Use Case) Diyagramı

Sıralama Şeması

Etkinlik Şeması

Durum Makinesi Diyagramı

Nesne Yönelimli Analiz

- * Analiz aşamasının ilk adımı gereksinimlerin belirlenmesidir.
- * Gereksinimler, sistemin düzgün çalışması için yapması beklenenleri ifade eder
- * Gereksinimler ilk olarak kullanıcılarla(paydaşlarla) görüşülerek (gerekli ise standart, yasa, yönetmelik vb. dokümanlar takip edilerek) oluşturulur, ve neler istediği belirlenir.
- * Daha sonra kullanım durumları (use case) belirlenerek gereksinimler ayrıntılandırılır.
- * Sistemin tüm istenenlerle birlikte, beklenmeyen ya da istenmeyen durumları da göz önüne alınır.
- * Kullanım durumları analiz edilerek nesneler, nesne üyeleri ve nesneler arasındaki bağıntılar belirlenir.

The Unified Modeling Language
UML
Birleştirilmiş Modelleme Dili

Modelleme

Gerçegin soyutunu (basit gösterimini) oluşturma işlemidir.

Soyutlama sayesinde gereksiz ayrıntılar gizlenir ve ilgilenilecek detaylara odaklanılır. Böylece karmaşıklıktan kurtulunur.

Ayrıntıların gerekli ya da gereksiz olduğu modelin amacına bağlıdır.

Günümüz yazılımlarının kod karmaşıklığı her geçen gün artmaktadır.

Bu nedenle, yazılımların farklı yönlerini, farklı ayrıntılarla ifade etmek üzere model oluşturulur.

Modelleme

Model sayesinde;

- * fikirlerin tartışılması/paylaşılması, yazılım aktörlerinin (son kullanıcı, yazılım geliştirici, tasarımcı, iş analisti vb.) iletişimini kolaylaştır.
- * yazılıminin gerçeklenerek (kodlanması) ürün haline dönüştürülmesi daha kolay olur.

UML (Unified Modelling Language)

Birleştirilmiş Modelleme Dili

Yazılım sistemlerinin temel bileşenlerini ifade etmek ve farklı yönlerini, farklı ayrıntılarla göstermek için kullanılan grafiksel modelleme dilidir.

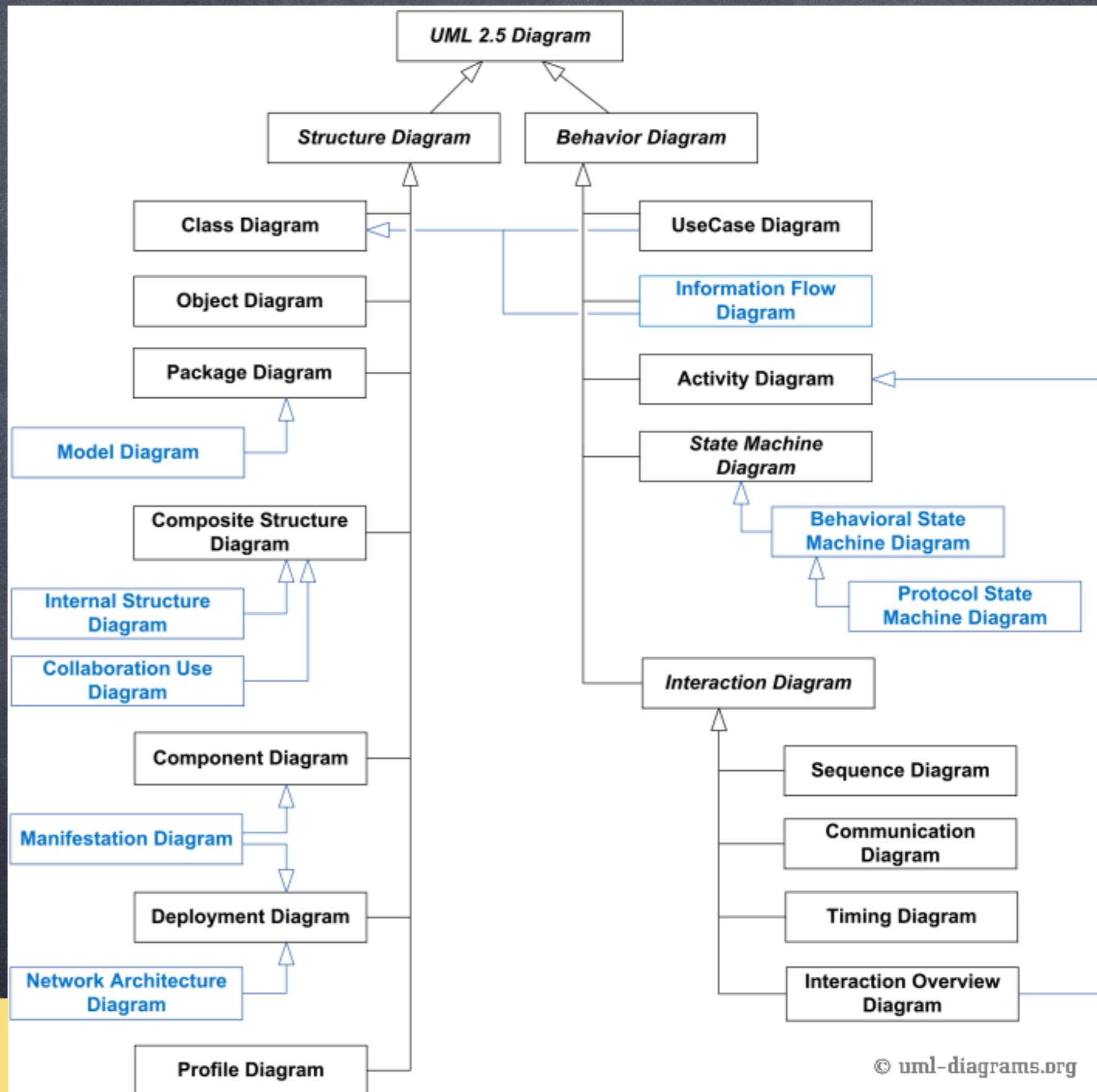
UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems. (<https://www.uml-diagrams.org/>)

UML

- * Yapı diyagramları (Structure Diagrams): sistemin(sistem bileşenlerinin) statik yapısını gösterir
- * Davranış Diyagramları (Behavior Diagrams): Sistemdeki nesnelerin davranışlarını (zamana göre değişimlerini) gösterir.

Not: mavi ile belirtilen gösterim şekilleri UML 2.5 standarı değildir.

<https://www.uml-diagrams.org/uml-25-diagrams.html>

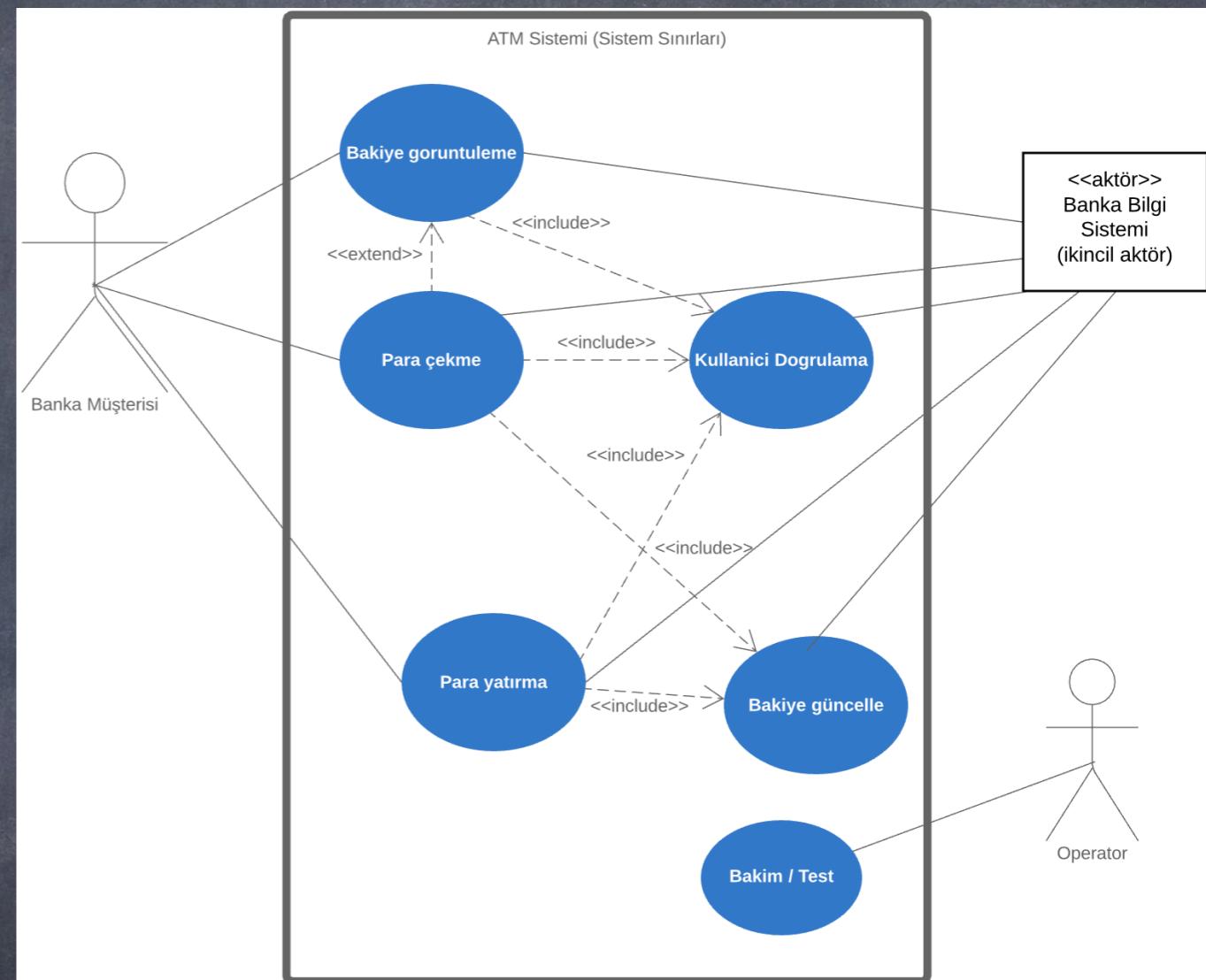


Örnek Uygulama: ATM İçin Yazılım Geliştirme Gereksinimlerin Belirlenmesi

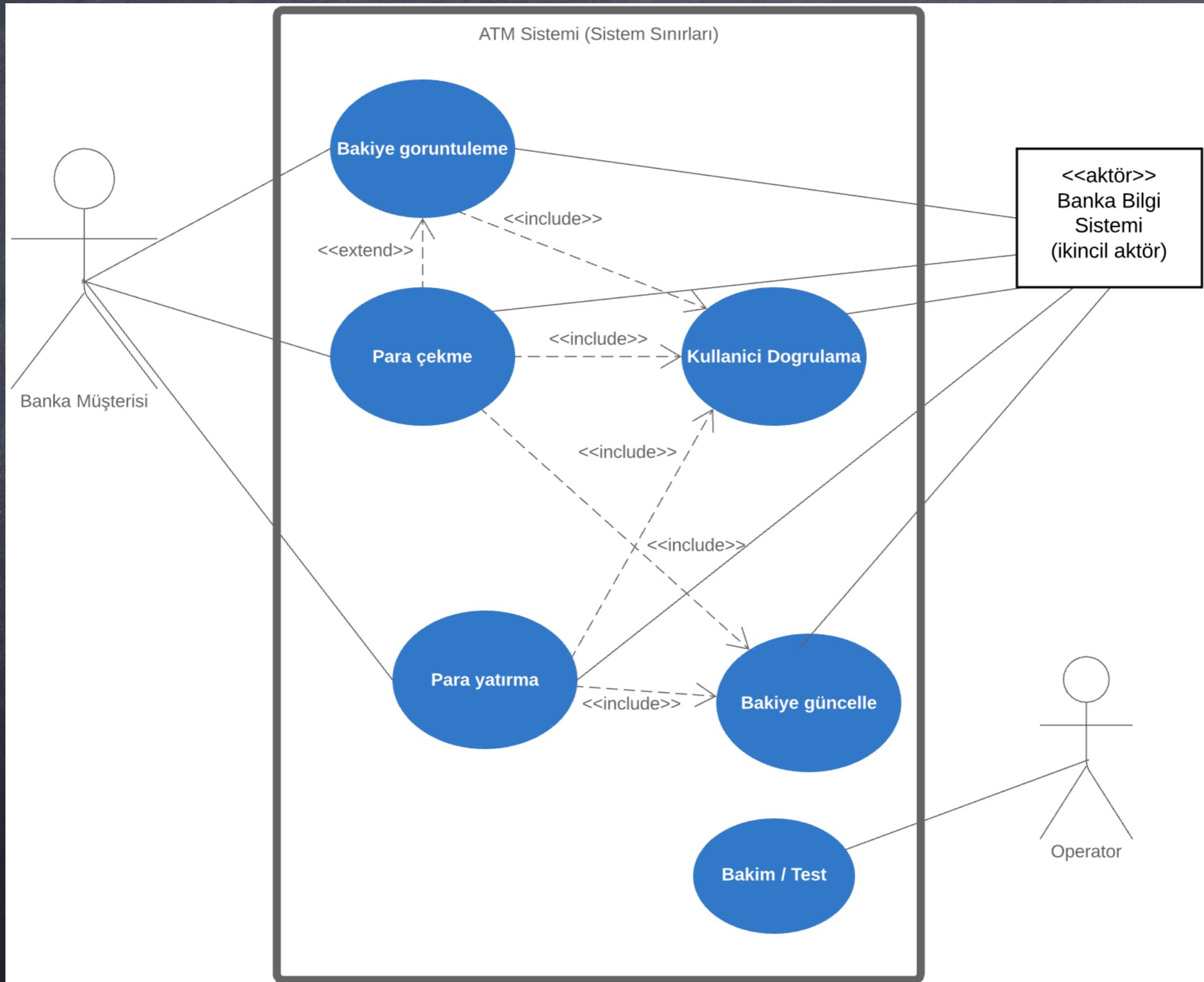
- * ATM aşağıdaki donanım bileşenlerinden oluşur:
 - * Tuş takımı; veri girişinden sorumludur.
 - * Ekran; mesajların görüntülenmesinden sorumludur.
 - * Para bölmesi; para alma ve para verme işleminden sorumludur.
 - * Kart bölmesi; banka kartını alıp doğrulanmasından sorumludur.
- * Kullanıcılar banka kartı ile para çekme, para yatırma, bakiye görüntüleme işlemlerini yapabilmelidirler.
- * Banka görevlileri ATM sisteminin bakımını (para doldurma, sıkışan/yutulan kartları alma v.b.) yapmalıdır.
- * Tüm işlemlerde güvenlik gereksinimleri göz önünde bulundurulmalıdır (yetki kontrolü).
- * Sistem, Banka Bilgi Sistemiyle (kredi kartı servis sağlayıcıları ile) gevrimiçi çalışmalıdır.

UML – Kullanım Durumu (Use Case) Diyagramları

- * Davranışsal gösterim şekillerinden biridir.
- * Kullanım durumu, sistem ile kullanıcıların (aktör/rol) etkileşimiini tanımlar.
- * Her kullanım durumunun tek bir hedefi olmalıdır. Bu hedefe giden farklı yollar olabilir.
- * Kullanım durumları bir araya gelerek (kullanım durumu senaryosu/modeli), aktör sistem ile etkileşime girdiğinde neler olduğunu anlatır.
- * Kullanım durumları "sistem aktör açısından ne yapar" ile ilgilenir. Nasıl yapar ile ilgilenmez.
- * Aktör belirli bir görev için olayları başlatan nesnedir. (Banka müsterisi, operator, banka v.s.)

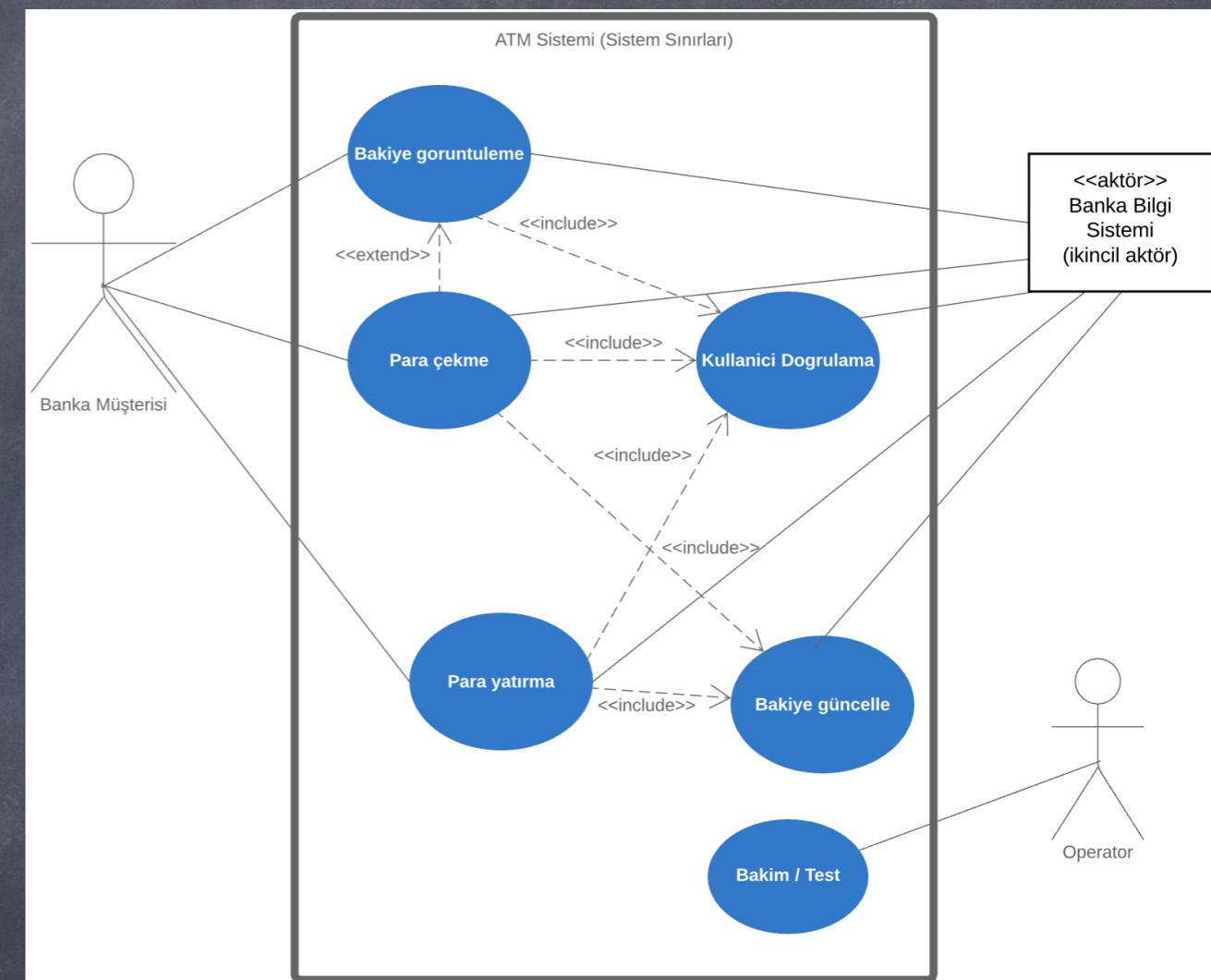


UML - Kullanım Durumu (Use Case) Diyagramları



Aktörler

- * Aktör belirli bir görev için olayları başlatan nesnedir. (yönetici, öğrenci, hasta, doktor, müşteri, operator, personel, banka, yolcu, diğer sistemler v.s.)
- * Sistem ile etkileşime giren harici varlıklardır
 - * Kullanıcılar
 - * Diğer sistemler / alt sistemler
 - * Fiziksel ortam (Hava, su vb.)
- * İkincil Aktör, Kullanım durumunun tamamlanması için etkileşime girilen aktördür.
- * Kullanım durumları birincil aktörler tarafından başlatılır. İhtiyaç duyulduğunda ikincil aktöre istek gönderilir.

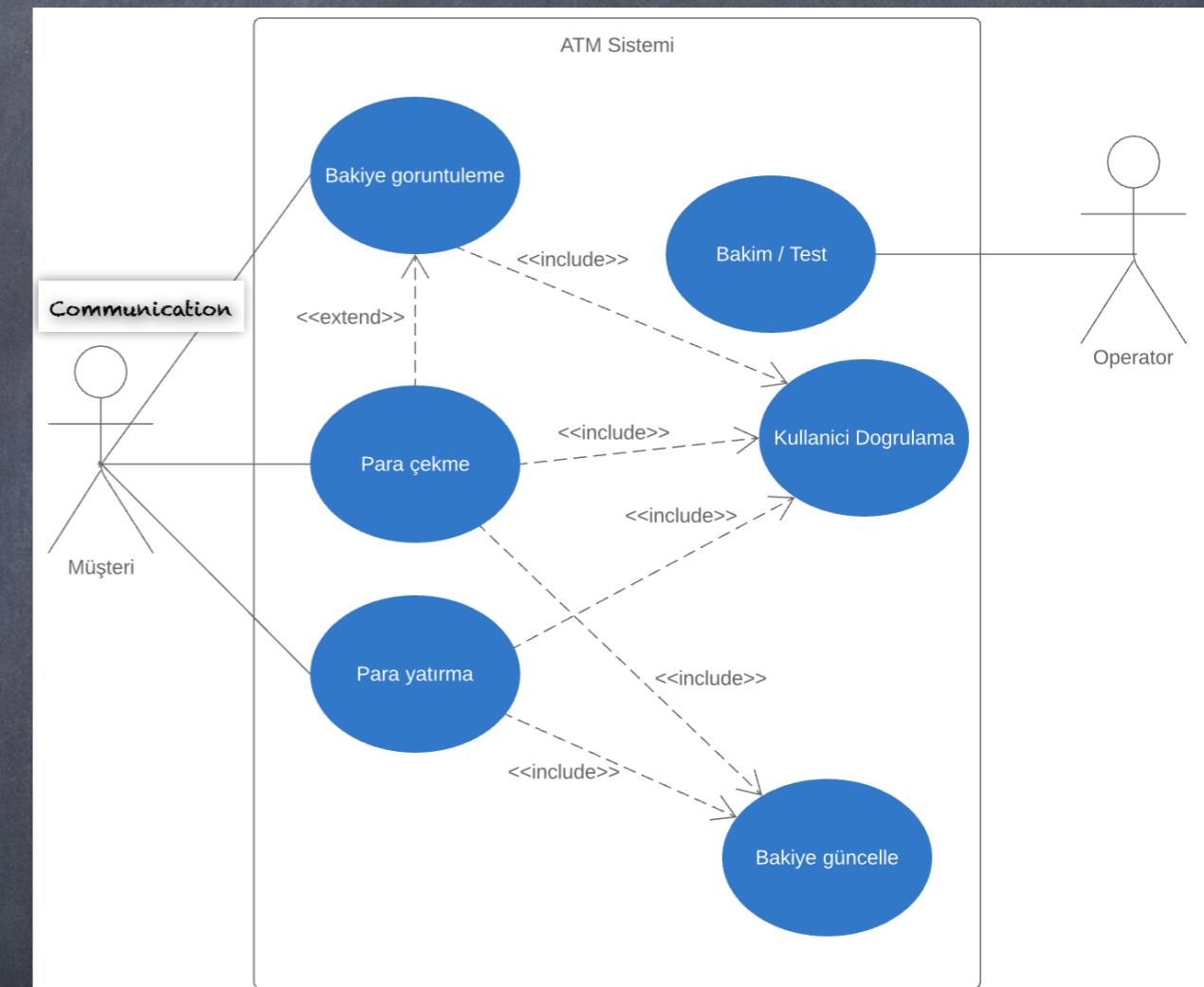


Kullanım Durumu (Use Case) Ayrıntıları

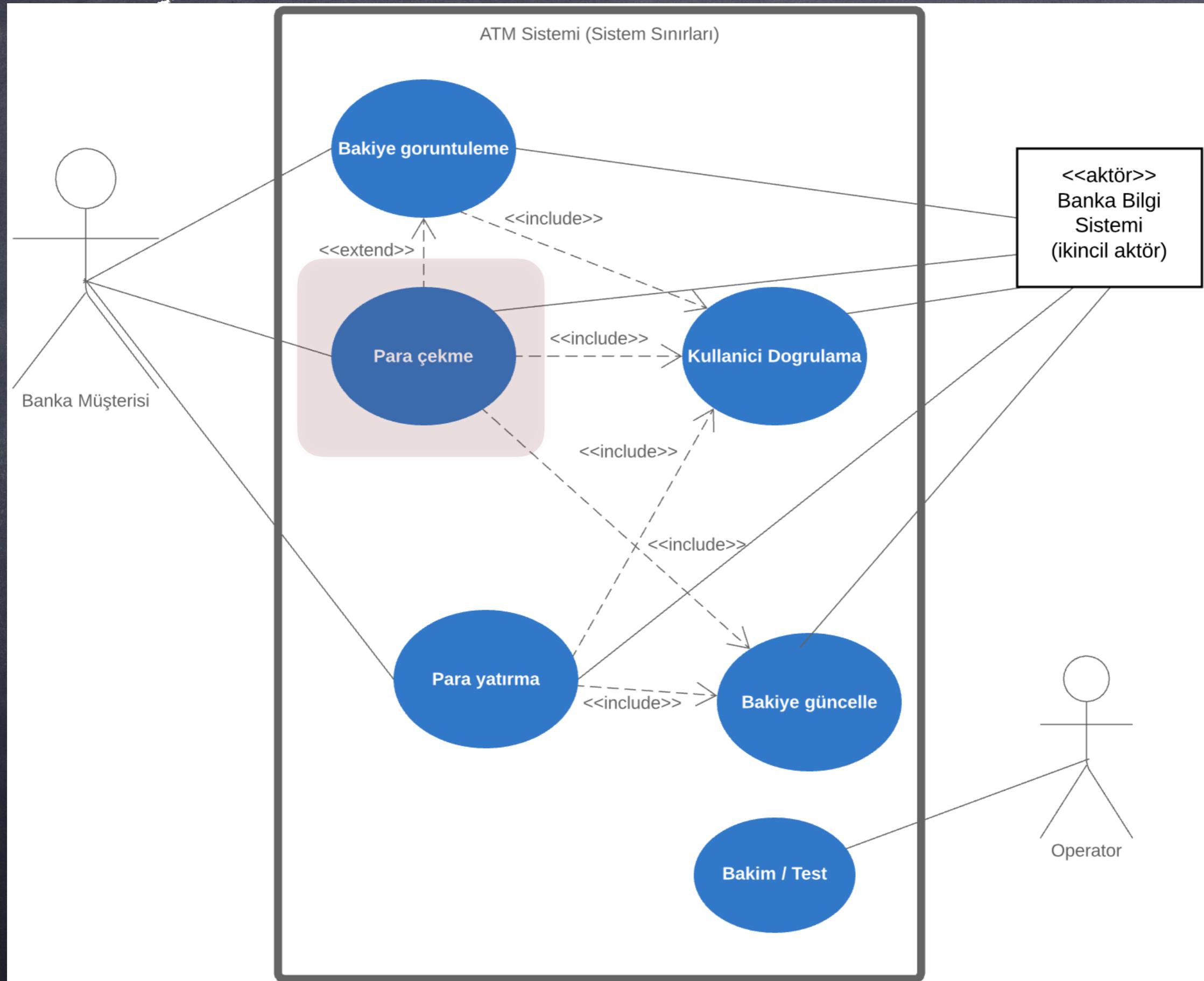
- * Her kullanım durumunun tek bir hedefi olmalıdır. Bu hedefe giden farklı yollar olabilir.
- * Kullanım durumunun ayrıntıları için, gereksinim listesi kullanılarak, metinsel ifadeler oluşturulur. Aktör ile sistem arasındaki olay akışı, istishnalar/ beklenmeyen durumlar da göz önüne alınarak anlatılır.
- * Kullanım durumu ayrıntılarında aşağıdakiler bulunur:
 - * Kullanım durumunun tanımı:
 - * Eşsiz bir ad, özet, oluşturma ve değiştirme tarihleri, sürüm numarası, sorumlu kişiler
 - * İlgili aktörler
 - * Giriş koşulu
 - * Çıkış koşulu
 - * Olay akışı
 - * Özel gereksinimler
 - * cevap süresi 5sn olmalı, yazılıan şifreler görülmemeli vb.

Bağıntılar

- * Communication: Aktör-sistem etkileşimi
- * Include: Birden fazla kullanım durumu için gerekli işlevler (code reuse)
- * Extends : istisna olarak ya da ara sıra gerçekleşen kullanım durumlarını ifade eder. Soyutlama adına (karmaşıklığı azaltmak) ayrı olarak verilebilir.
- * Kalıtım: ortak işlemler için temel kullanım durumu tanımlanır ve kalıtım kullanılır



Para Çekme Kullanım Durumu (Use Case)



Para Çekme Kullanım Durumu (Use Case) Ayrıntıları

- * Kullanım durumunun tanımı:
 - * Para çekme
 - * ATM den para çekme işlemini tanımlar
 - * 24.07.2009, 27.07.2009 V1.1.1 Kullaniciadi1
- * İlgili aktörler: Banka müsterisi, Banka BS (ikincil)
- * Giriş koşulu: Banka kartına sahip olan Müşteri ATM makinasına gelir.
- * Çıkış koşulu: Müşteri işlemini (transaction) tamamlar
- * Olay akışı:
- * Özel gereklilikler: UI gereklilikleri, İşlem gecikmesi en fazla 1 dk. olmalı, 24 saat çalışmalı vb.

"Para Çekme" Kullanım Durumu Olay Akışı (Ana senaryo-başarılı)

1. ATM sistemi Ekrana, müşteriden "kartını kart bölmeyecekini" isteyen mesaj yazdırır.
2. Kart sahibi kartını kart bölmeyecekini takar.
3. ATM sistemi kart doğrulamasını yapar.
4. Ekrana şifre girilmesini isteyen mesaj gönderilir.
5. Tuş takımını kullanarak girilen şifreyi alır.
6. Kullanıcı doğrulama ve yetki kontrolü için banka bilgi sistemine istek gönderilir.
7. Banka sistemi erişim isteğini kabul eder ve gerekli bilgileri gönderir.
8. ATM sistemi ekrana, çekilecek tutarları listeler.
9. Kart sahibi çekilecek tutarı girer.
10. ATM sistemi kartı çırktır.
11. Kart sahibi kartı alır.
12. ATM sistemi parayı ve makbuzu verir.
13. Kart sahibi parayı ve makbuzu alır.

Not: Kırmızı renkli işlem basamakları Kullanıcı Doğrulama (erişim denetimi) kullanım durumunu ifade eder.

"Para Çekme" Kullanım Durumu Olay Akışı (Alternatif akışlar)

A1. Kart doğrulanamadı (3)

4. Kart bölmesi kartı çökartır.

5. İşlem sonlandırılır.

A2. Yanlış şifre (6)

8. 3 den az kez yanlış ise yeniden gir.

8. 3 kez yanlış girilmiş ise kart yükülür.

9. İşlem sonlandırılır.

A3. Yeterli para yok (9)

10. ATM de yeterli bakiye yoksa yeni tutar girmesi istenir.

11. Hesapta yeterli bakiye yoksa yeni tutar girmesi istenir.

A4. Banka kartı alınmadı(10)

11. 30 sn. boyunca alınmadı ise kart yükülür

12. Gerekli bilgilendirme yapılır

13. İşlem sonlandırılır

A5. Para alınmadı (12)

13. 30 sn. boyunca alınmadı ise para yükülür

14. Gönderilen ve yükulan para karşılaştırılır.

15. Gerekli ise hesap güncellemesi yapılır

12. Gerekli bilgilendirme yapılır

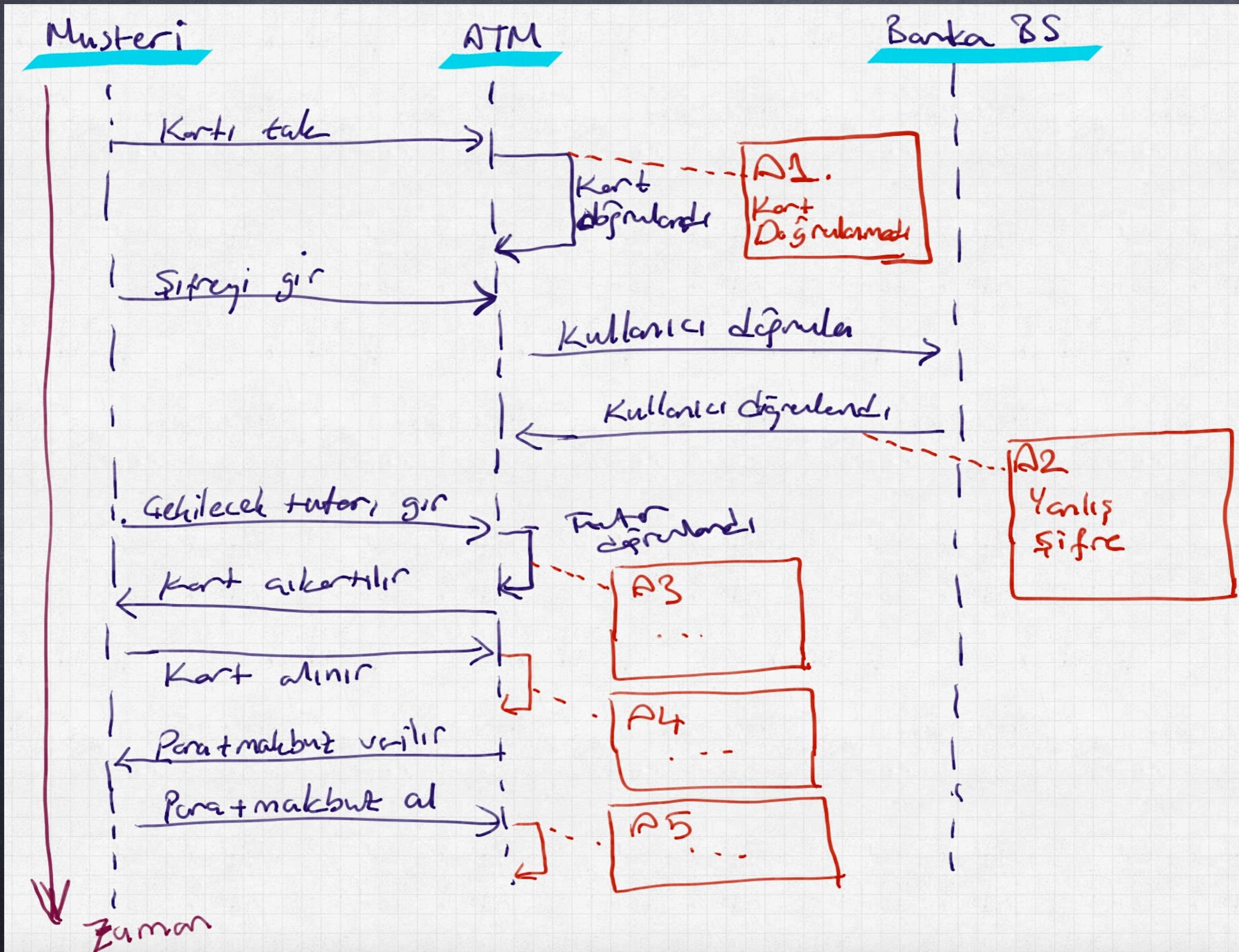
13. İşlem sonlandırılır

...

UML - Sıralama Şeması (Sequence Diagram)

- * Davranışsal (dinamik) gösterim şekillerinden biridir.
- * Kullanım durumunun grafiksel gösterimi için kullanılır
- * Nesneler arası işbirliğinin zamana göre anlatımı/ anlaşılması gereğinde oluşturulabilir
- * Ayrıntıları ihtiyaca göre değişiklik gösterebilir

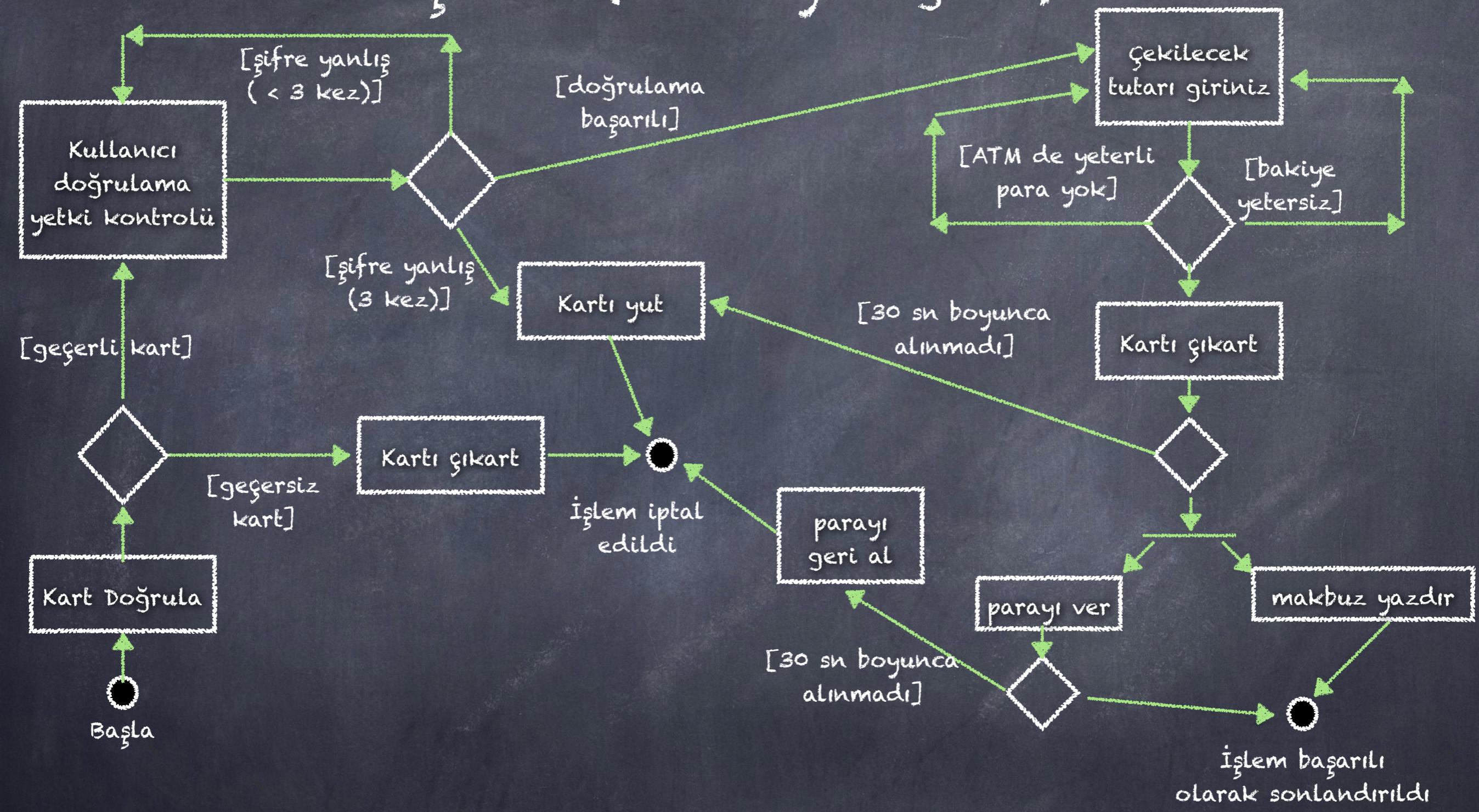
Para Çekme Kullanım Durumu İçin ATM Sıralama Şeması



UML - Etkinlik Şeması (Activity Diagram)

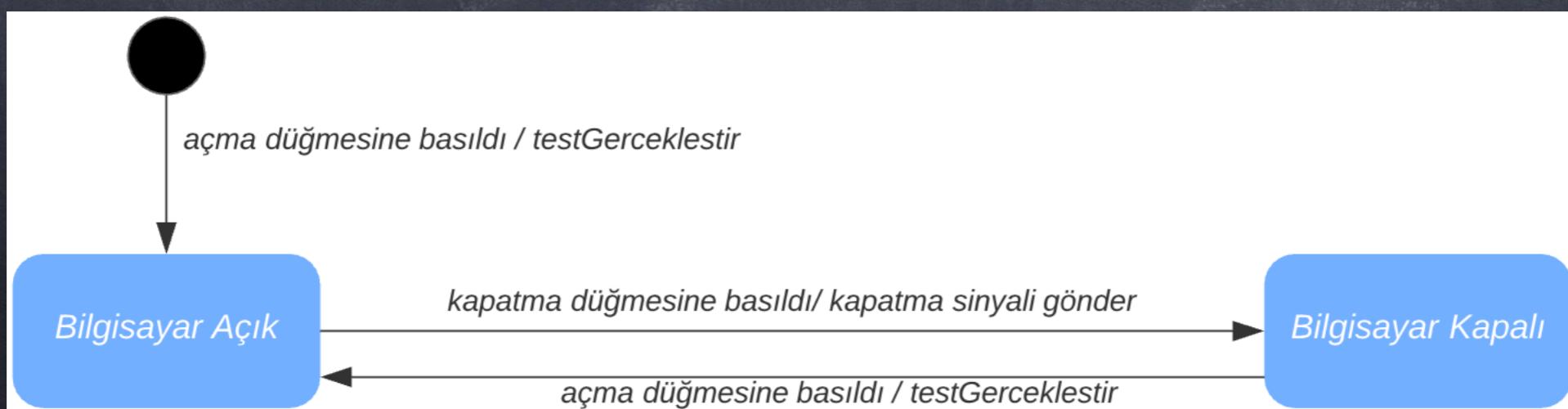
- * Davranışsal (dinamik) gösterim şekillerinden biridir.
- * Kullanım durumlarındaki akış denetiminin grafiksel gösterimi için kullanılır.
- * Ana akışla birlikte alternatif akışları da gösterir.
- * Akış şemalarına benzer.
- * Ayrıntıları ihtiyaca göre değişiklik gösterebilir.

Para Çekme Kullanım Durumu İçin ATM Etkinlik Şeması (Activity Diagram)



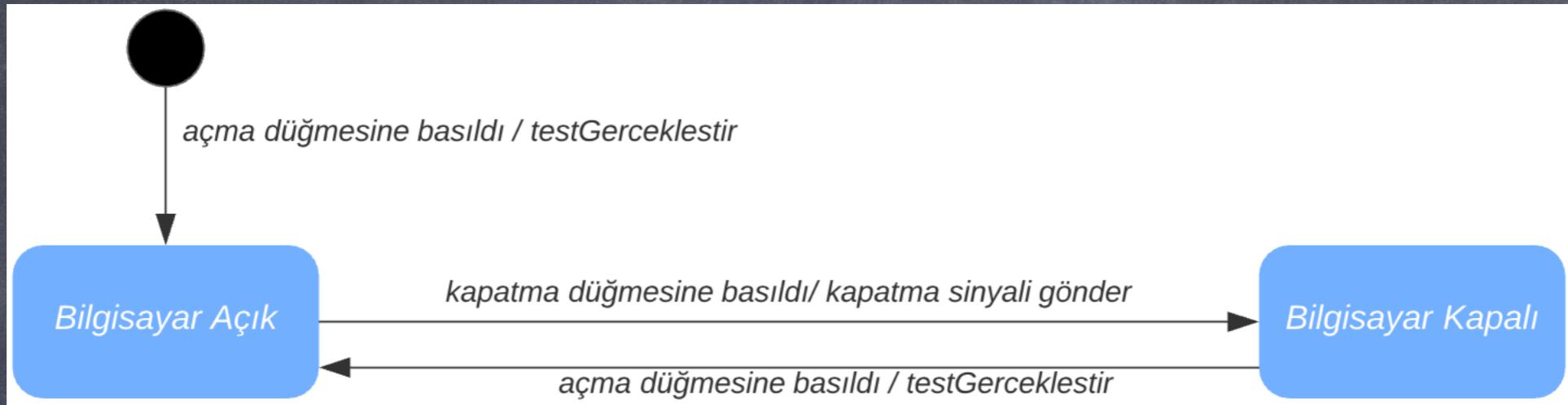
UML - Durum Makinası (State Machine) Diyagramı

- * Davranışsal (dynamik) gösterim şekillerinden biridir.
- * Sistemin içinde bulunduğu durumları ve durumlar arası geçişleri modellemek için kullanılır.
- * Haberleşme protokollerini, kullanıcı arayüzleri vb. olay güdümlü sistemlerin anlaşılabilirliğini sağlar



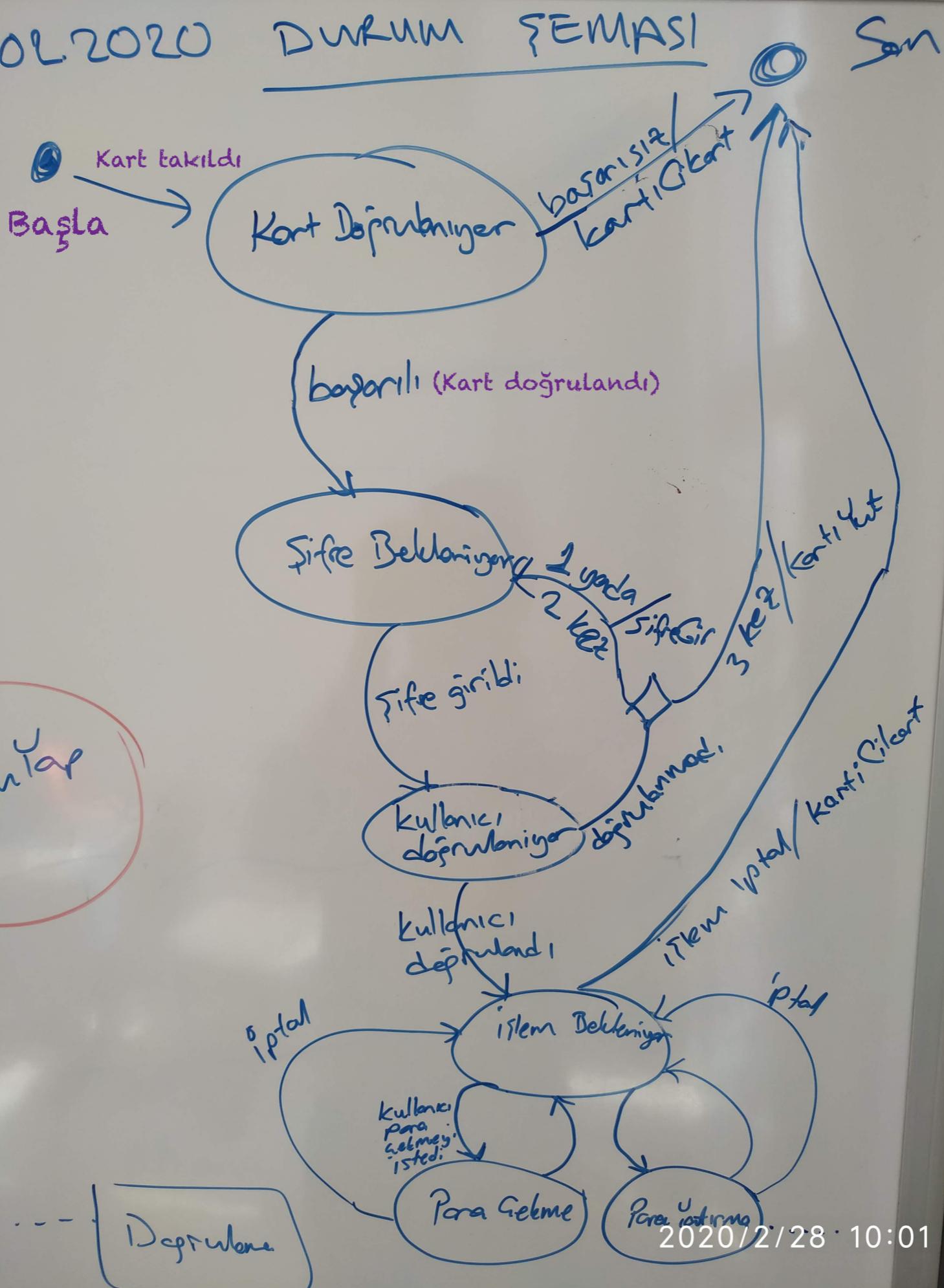
UML - Durum Makinası (State Machine) Diyagramı

* Durum diyagramları, genişletilmesi daha kolay olacağı için, metinsel olarak ifade edilebilir.

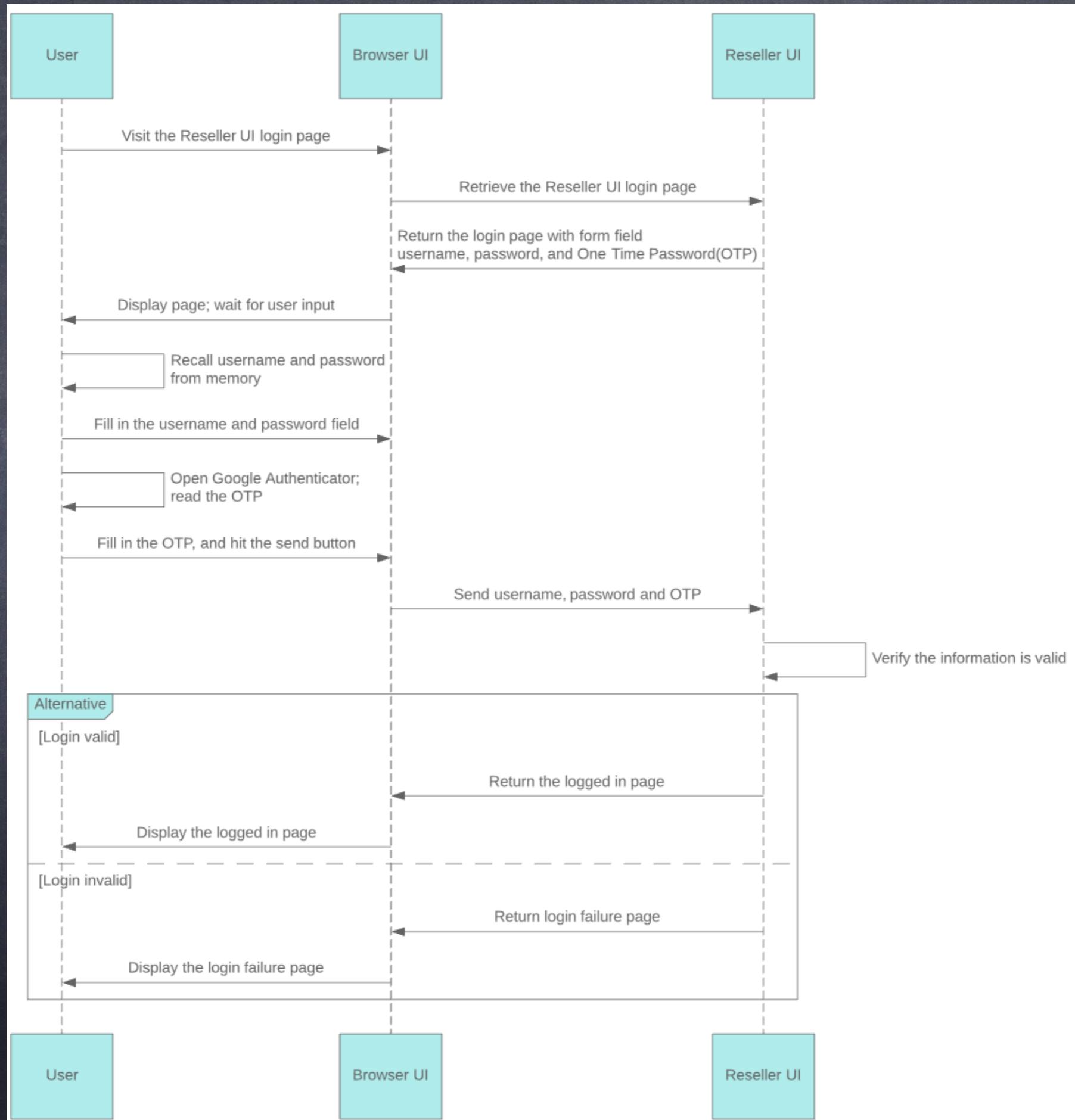


Mevcut Durum	Gercekleşen Olay	Yeni Durum	Hareket (action)
Başla	aç	Bilgisayar Açıktır	testGerceklestir
Bilgisayar Açıktır	kapat	Bilgisayar Kapalıdır	kapatmaIslemi
Bilgisayar Kapalıdır	aç	Bilgisayar Açıktır	testGerceklestir

ATM Durum Makinası Diyagramı



Örnek: Kimlik Doğrulama İşlemi Sıralama Şeması



Örnek: E-Ticaret Uygulaması

Belirtilen gereksinimlere (iş kurallarına) göre kullanım durum diyagramı oluşturunuz.

Senaryo

Elektronik ticaret yazılımının ihtiyacı olan verilerin yönetimi için bir veritabanı geliştirilmesi isteniyor. İş kuralları verilen bu veritabanının VB diyagramını ve ilişkisel şemasını oluşturunuz.

İş Kuralları

- Bu veritabanında her müşteriye bir numara verilerek müşterinin TC Kimlik numarası, adı, soyadı, yaşadığı ili saklanması düşünülüyor.
- İllerin plaka numarası ve adı saklanır.
- Her siparişe bakan bir satış temsilcisi mevcuttur. Satış temsilcilerinin TCKimlikNo, ad ve soyad bilgileri mevcuttur.
- Ürünlerin (kişisel bilgisayar, telefon vb.) kodu, adı, fiyatı ve stok miktarlarının saklanması gerekmektedir.
- Ürünlerin kategorileri (bilgisayar, ev elektroniği, kozmetik vb.) mevcuttur.
- Müşterilerin ürün siparişleri saklanarak her bir siparişe bir fatura kesilmesi sağlanmalıdır.
- Her siparişin eşsiz bir sipariş numarası ve sipariş tarihi mevcuttur.
- Sipariş edilen bir ürünün sipariş adedi ve birim fiyatı (kişiye özel indirim v.s. nedeniyle ürün tablosundaki fiyat farklılığı gösterebilir) da kaydedilmelidir.
- Faturaların fatura numarası, tarih ve fatura adresi bilgileri saklanmalıdır. (Toplam fiyat hesaplanabilir ya da saklanabilir).
- Siparişler bir kargo firması tarafından iletilir. Kargo firmasının kodu, adı, adresi bilgileri yer alır. Her kargo firmasında siparişlerden sorumlu bir yetkili yer alır.
- Bir ürünün yalnızca bir kategorisi mevcuttur. Bir kategori çok sayıda ürünün kategorisi olabilir.
- Bir siparişte en az bir ürün bulunur. Ancak çok sayıda ürün de bulunabilir. Bir ürün çok sayıda siparişte yer alabilir.
- Bir müşteri çok sayıda sipariş verebilir. Bir sipariş yalnızca bir müşteri tarafından verilebilir.
- Bir siparişin yalnızca bir faturası olabilir. Bir fatura yalnızca bir siparişin faturası olabilir.
- Bir sipariş ile yalnızca bir satış temsilcisi ilgilendir. Bir satış temsilcisi çok sayıda sipariş ile ilgilenebilir.
- Bir müşteri yalnızca bir ilde yaşayabilir. Bir ilde çok sayıda müşteri yaşayabilir.
- Bir sipariş yalnızca bir kargo firması tarafından iletilir. Bir kargo firması çok sayıda sipariş iletebilir.