

# SEQUENCE DIAGRAMS

# SIRALAMA DİYAGRAMLARI



*Arş. Gör. Simge YILDIRIM*



# Sequence Diagrams

"Sequence" kelime anlamı olarak "birbirini takip eden, ardışıl olan, peşi sıra" anlamlarına gelmektedir. Sequence Diagram (Sıralama Diyagramı) daha çok nesnelerin birbirleriyle olan iletişimlerini sırasıyla göstermek için kullanılır.

Bir "sequence" diyagramı temel olarak nesnelere(objects), mesajlardan (messages) ve zaman eksenlerinden(timeline) oluşmaktadır



# Nesneler (Objects)

Belirli görevleri gerçekleştirmek için tanımlanan yapı bloklarıdır. Nesneler, tasarım ve modellemenin en küçük yapı taşı olduğu için hemen hemen her tür diyagramda nesnelere görmekteyiz. "sequence" diyagramında nesnelere aşağıdaki şekilde görüleceği üzere dikdörtgen ve bu dikdörtgen içinde nesne ismi olacak şekilde temsil edilir. Önemli noktalardan birisi ise nesne isminin :Nesnelismi formatında olmasıdır. "sequence" diyagramlarında nesnelere genellikle diyagramın fiziksel olarak en üstünden ve soldan sağa olacak biçimde sıralanır. Yani nesnelere "sequence" diyagramlarındaki orijini sol üst köşedir. Ve yayılım biçimi soldan sağadır. Bu bir kural olmamakla beraber kullanılan en yaygın biçimi bu şekildedir



# Sequence Diagrams

Sequence diyagrami iki boyutludur:

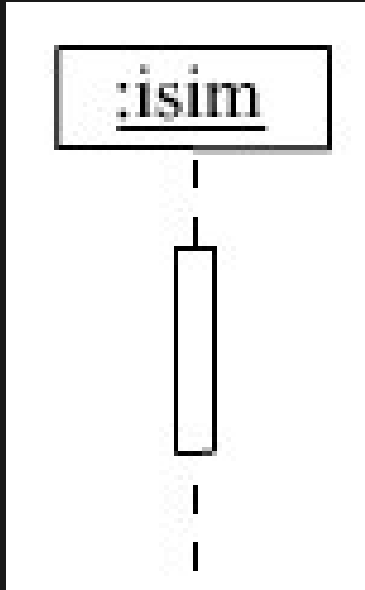
**Dikey boyut:** Mesajların/olayların sırasını oluşma zamanı sıralarına göre gösterir.

**Yatay boyut:** Mesajın gönderildiği nesne örneklerini (object instances) gösterir.

Sequence diyagramlarının akışı soldan sağa doğru olmalıdır.



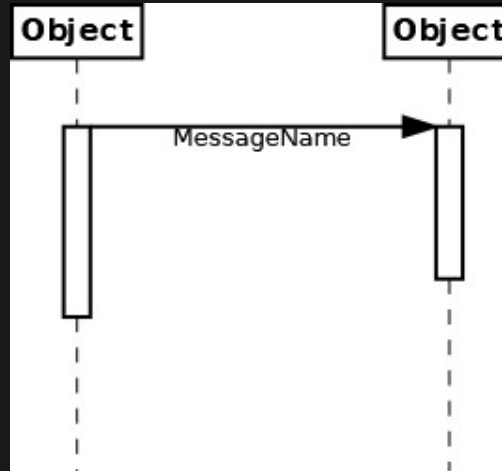
# Nesne ve Lifeline'ın Gösterimi



Nesneler (Objects) diyagramda kutucuklar ile gösterilirken (Class Diyagramları'ndaki sınıflar gibi), nesnelerin Lifeline (Yaşam Çizgisi) adı verilen ömürleri ise bu kutucukların altına çizilen dikey bir kesikli çizgi ile belirtilir. Kesik çizginin en altı teorik olarak sonraki zamanı göstermektedir.

# Activation Bar ve Mesajların Gösterimi

Diyagramda Lifeline'lar üzerinde bir de nesnenin hangi anlarda ne kadar süreyle aktif olduğunu gösteren ve **Activation Bar (Aktivasyon Çubuğu)** olarak adlandırılan dikdörtgensel çubuklar yer alır. Bu çubuklar ile mesajlaşma anlarında nesnelerin ne zaman ve ne kadar süreyle aktif kaldığı görülür. Ayrıca iki nesne arasındaki etkileşime bu diyagramda basitçe **mesaj (message)** adı verilir ve o da oklarla temsil edilir. Okların yönü mesajı gönderenden alana doğrudur. Okun altında (veya üstünde) mesajın ismi yazılmaktadır.



# Mesajlar (Messages)

Mesajlar nesnelar arasında iletiřimi belirtmek için kullanılan çizgilerden oluşmaktadır. Burada nesnelar arasında geçiş yaparken ileilmek istenen mesajda belirtilmelidir. 3 tip mesaj çizgisi kullanılmaktadır. Bunlar;

## *-Basit Mesaj*

Basit mesajlar nesnelar arasındaki akış kontrolünün iletimini göstermek için kullanılır. Nesneların methodlarını doğrudan çağırılmazlar. Sık kullanılan mesaj tipi değildir.

## *-Senkron Mesaj*

Nesne mesajı alıcı nesneye gönderir ve onun işlemini bitirmesini bekler, bu durumda senkron mesaj tipi kullanılır. Nesne tabanlı programlamada çağırılan birçok method senkron çalıştığından en çok kullanılan mesaj tipidir.

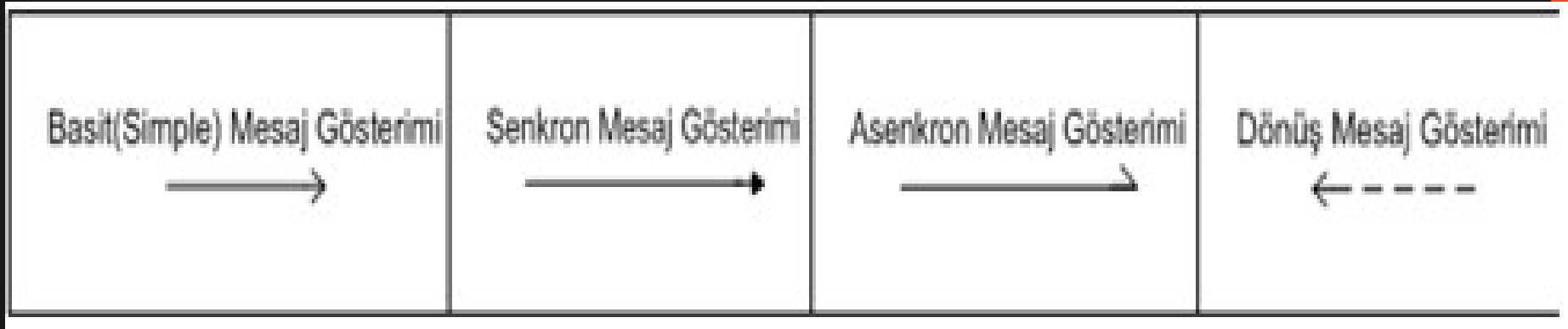
## *-Asenkron Mesaj*

Senkron mesajların tersine, asenkron mesajlar nesneye mesaj gönderdikten sonra cevap beklemeden işleme devam etmesinin gösteriminde kullanılır. Genellikle komut zincirlerinde kullanılır.

## *-Return Mesaj*

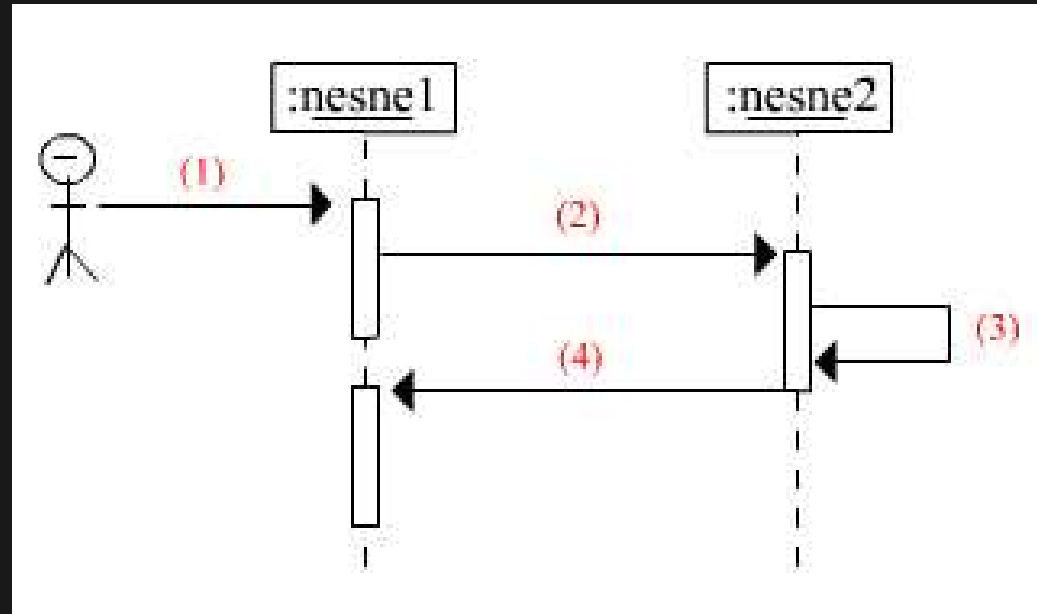
Senkron mesajlarda alıcı nesnenin işleminin bitimini, gönderen nesneye bildirmesinde kullanılır.

# Mesaj Gösterimleri



# Basit bir "sequence" Diyagramı

Nesneye en yakın olan aktiviteler daha erken gerçekleşir. Diyagramda senkron mesajlar 1-2-3-4 sırasında gerçekleşmektedir. Her mesaj gönderiminden sonra dikkat ederseniz aktivite kutuları çizilmiştir. Ayrıca 3. mesajda ise öz yinelemeli (recursion - kendi kendine mesaj) mesajlara örnek verilmiştir.



# "sequence" Diyagramı

Genellikle iki biçimde incelenmektedir : **Instance (örnek) ve generic (genel)** "sequence" diyagramları.

Instance olarak adlandırılan diyagramlarda genellikle basit bir şema çizilir, ayrıntılar gözardı edilir ve "**best case**" denilen en iyi olasılıklar ele alınır.

Generic diyagramlarda ise model daha karışık ve her yönüyle ele alınır. Öyleki döngü ve koşul yapıları dahi "sequence" diyagramında gösterilir.



## Instance "Sequence" Diyagramına Örnek :

---



Üç Nesneli Bir ATM Makinasının Çalışmasını Modelleyen "Sequence" Diyagramını Örneği;

**Tuş Takımı ve Para Alma Modülü:** Kullanıcının ATM makinası ile haberleştiği arayüzler (Nesne 1)

**Hesap Kontrol Modülü :** Kullanıcı doğrulama ve bakiye kontrol gibi mantıkların işletildiği birim (Nesne 2)

**Para İletme Bölümü :** Kullanıcının yani talebinin arayüz yani Nesne1 yardımıyla kullanıcıya sunulması. (Nesne 3)

---





## Instance "Sequence" Diyagramına Örnek :

---



iş kuralları:

1 - Kullanıcı şifresini yazar.

2 - Ardından çekmek istediği tutarı nesne 1 yardımıyla yazar. (tuş takımı)

3 - Çekmek istenilen tutar nesne 2 tarafından kontrol edilir.

4 - Eğer bakiye uygun ise Nesne 3'e mesaj gönderilerek bu modüle para aktarımı sağlanır.

Not : Bu örnekte bakiyenin uygun olacağı varsayılacaktır. (best case durumu)

5 - Nesne 3, Nesne 1 i yani arayüzü uyararak paranın alınması sağlanır.

---





## Instance "Sequence" Diyagramına Örnek :

---



iş kuralları:

1 - Kullanıcı şifresini yazar.

2 - Ardından çekmek istediği tutarı nesne 1 yardımıyla yazar. (tuş takımı)

3 - Çekmek istenilen tutar nesne 2 tarafından kontrol edilir.

4 - Eğer bakiye uygun ise Nesne 3'e mesaj gönderilerek bu modüle para aktarımı sağlanır.

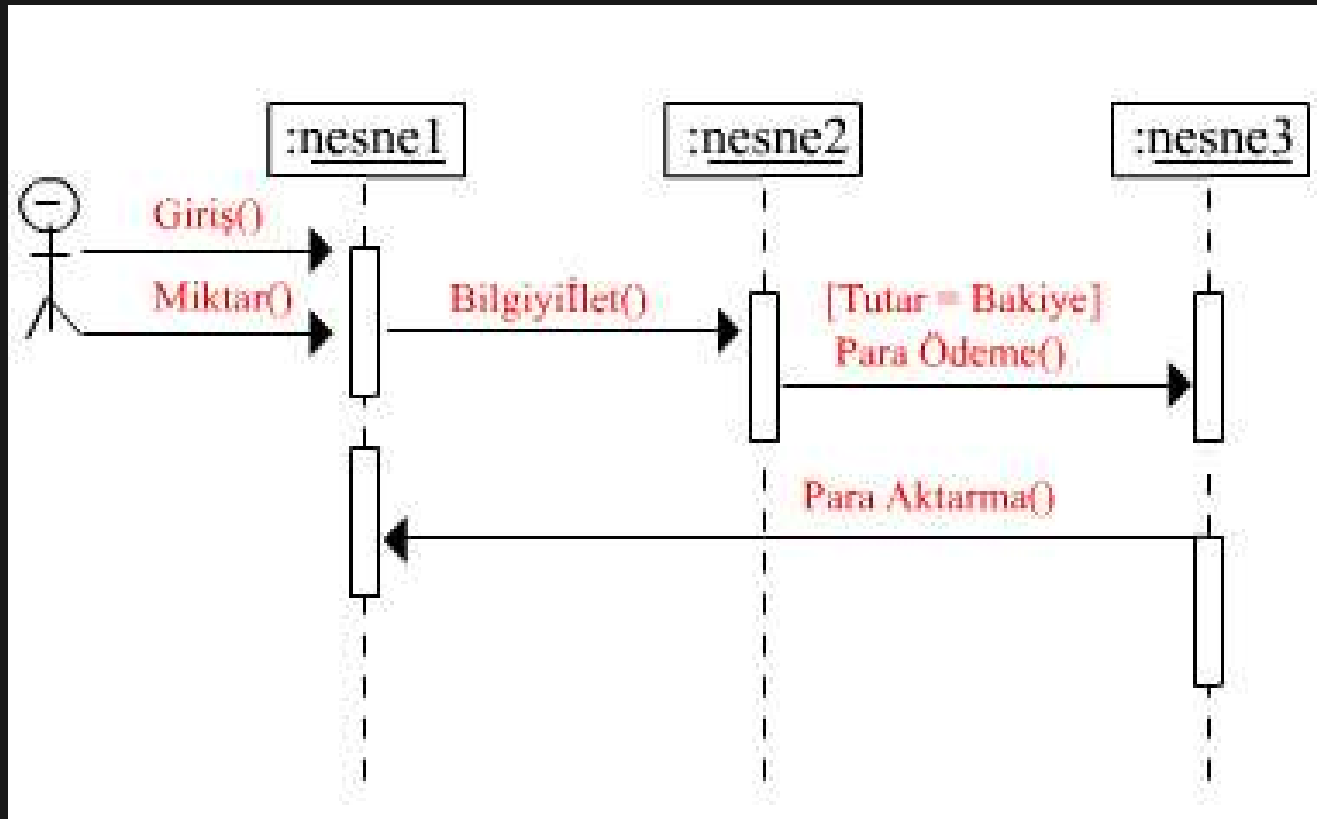
Not : Bu örnekte bakiyenin uygun olacağı varsayılacaktır. (best case durumu)

5 - Nesne 3, Nesne 1 i yani arayüzü uyararak paranın alınması sağlanır.

---

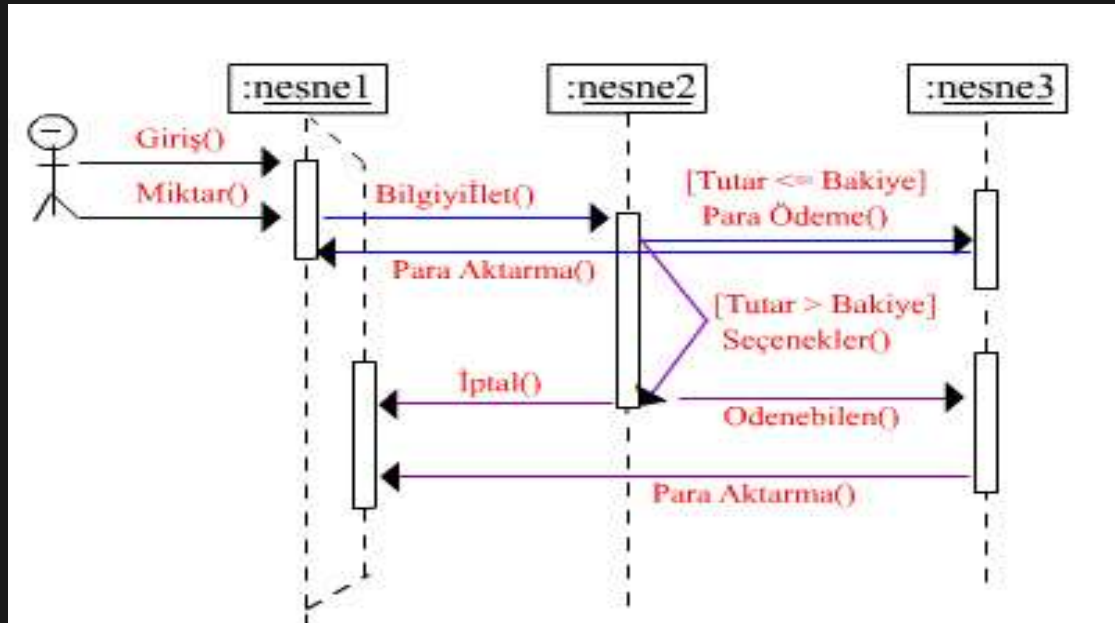


## Instance "Sequence" Diyagramına Örnek :

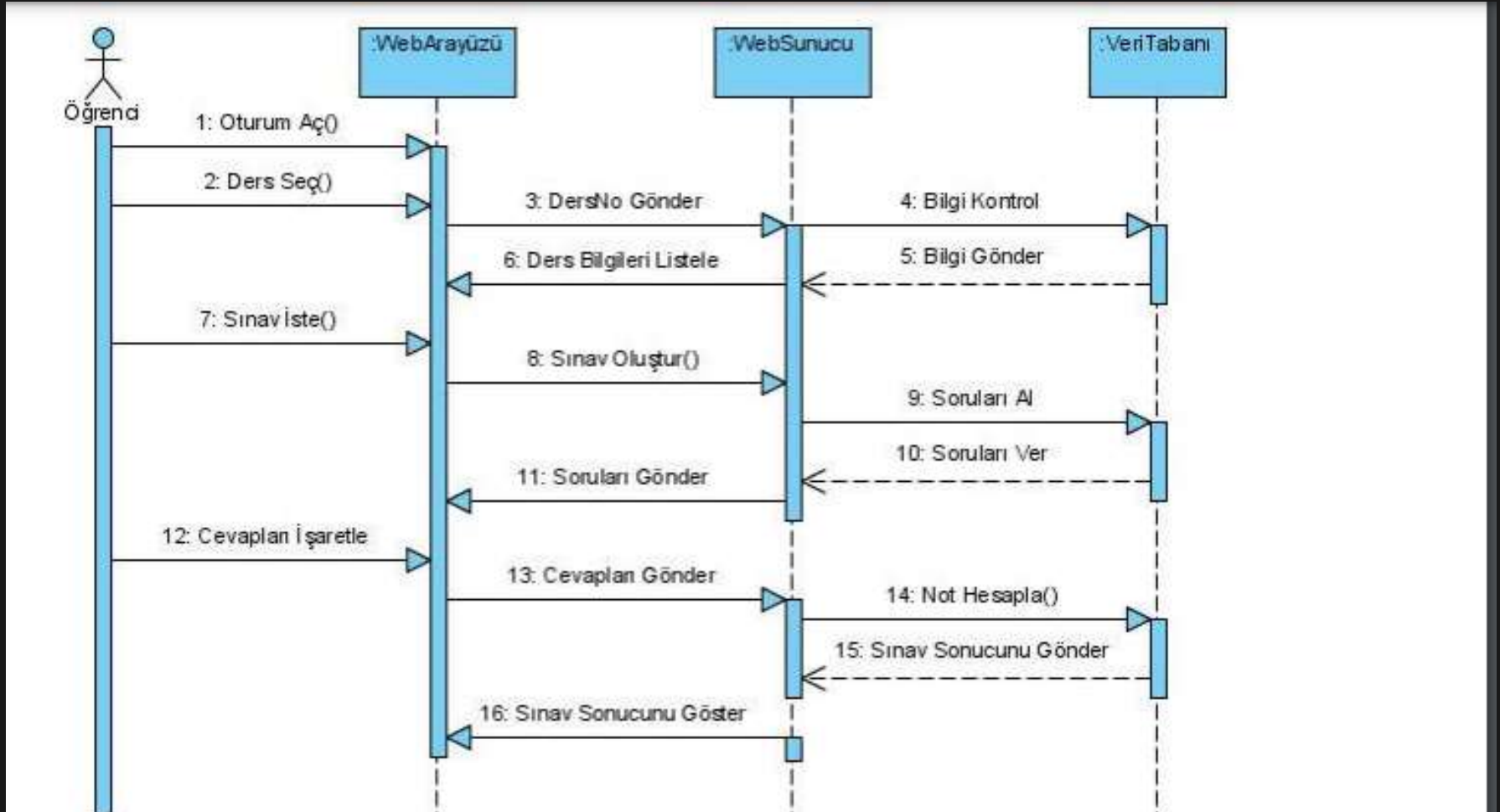


## Generic "Sequence" Diyagramına Örnek :

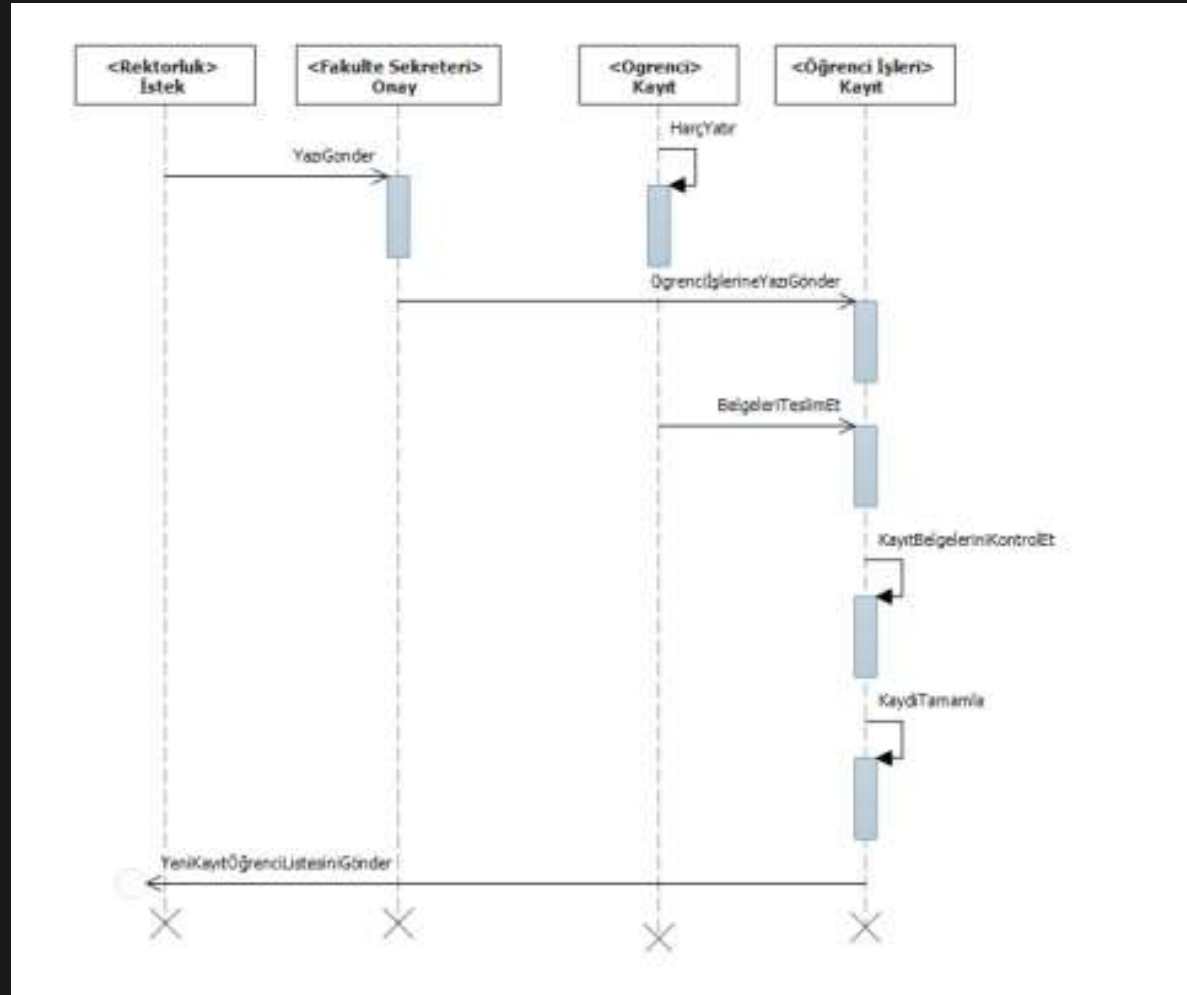
Generic "sequence" diyagramında birden fazla senaryo aynı anda ele alınmaktadır. Aşağıdaki örnekte kullanıcının girdiği tutar bakiyeden büyük ise iki senaryo daha eklenmektedir. Bu durumda ya ödenebilen maksimum miktar ödenmekte yada işlem iptal edilmektedir. Bunun gibi bir çok senaryo diyagrama eklenebilir. Tabii eklenen her senaryo diyagramı iyice karıştıracaktır. Bu yüzden diyagramları oluştururken bütün senaryoları içerebilecek durumları çıkarmaya çalışma ile birlikte maksimum okunabilirliği de gözönünde bulundurmak gerekir.



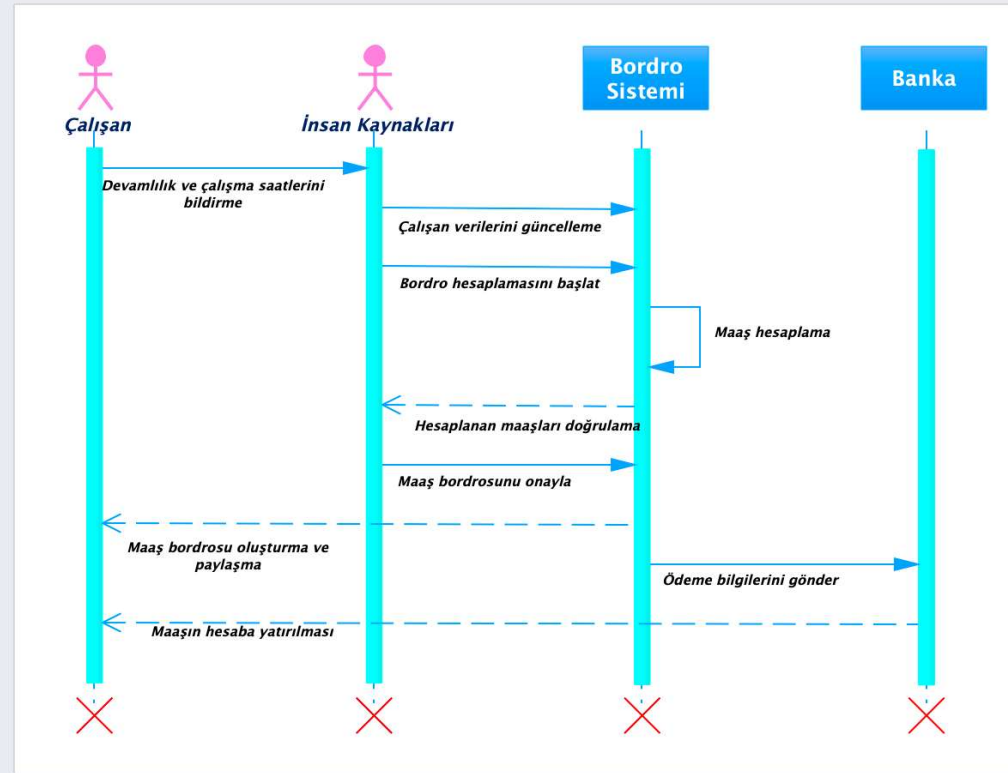
# Sınav Sequence Diyagramı



# Öğrenci Mezuniyet Sıralama Diyagramı

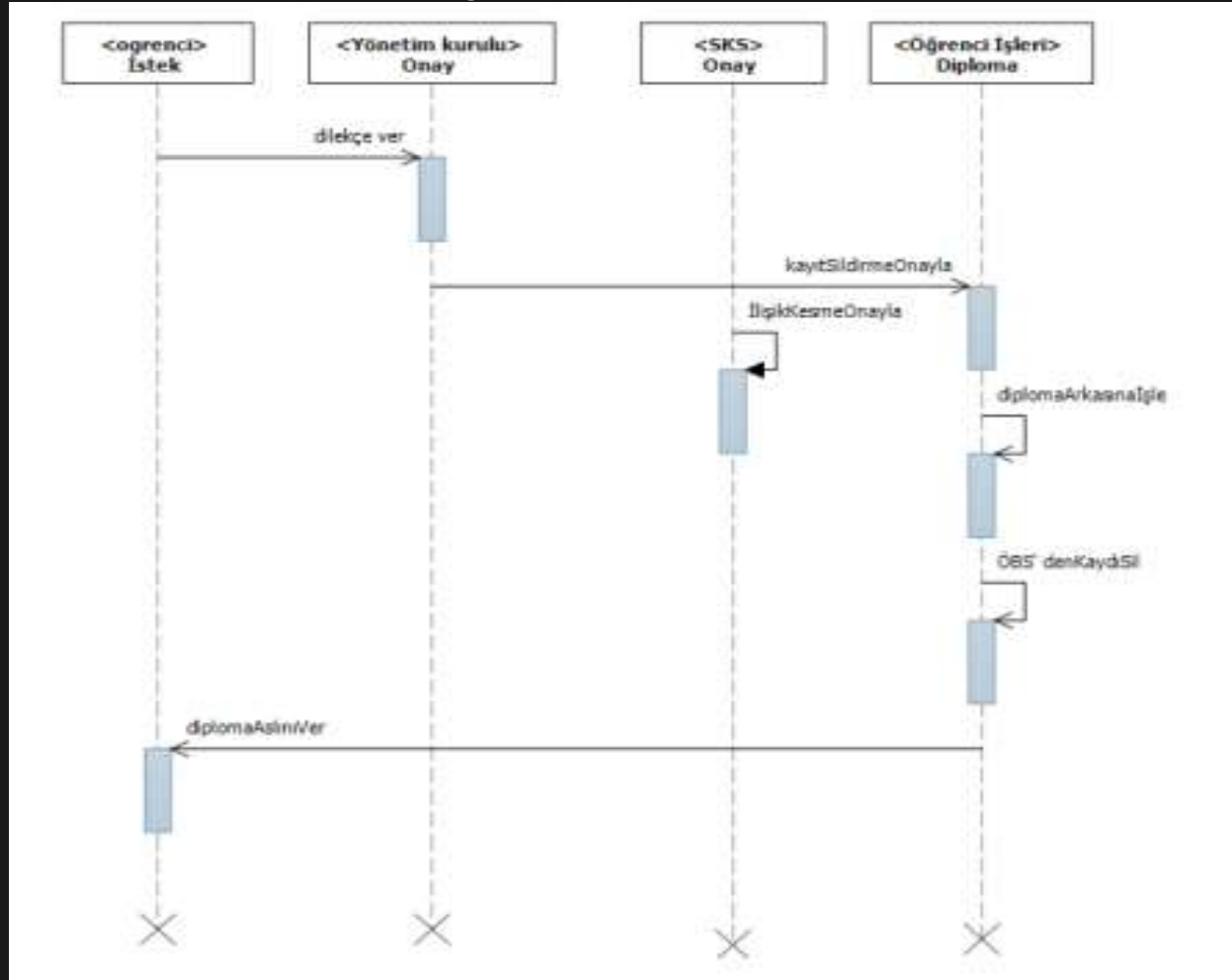


# Sıralama Diyagramı

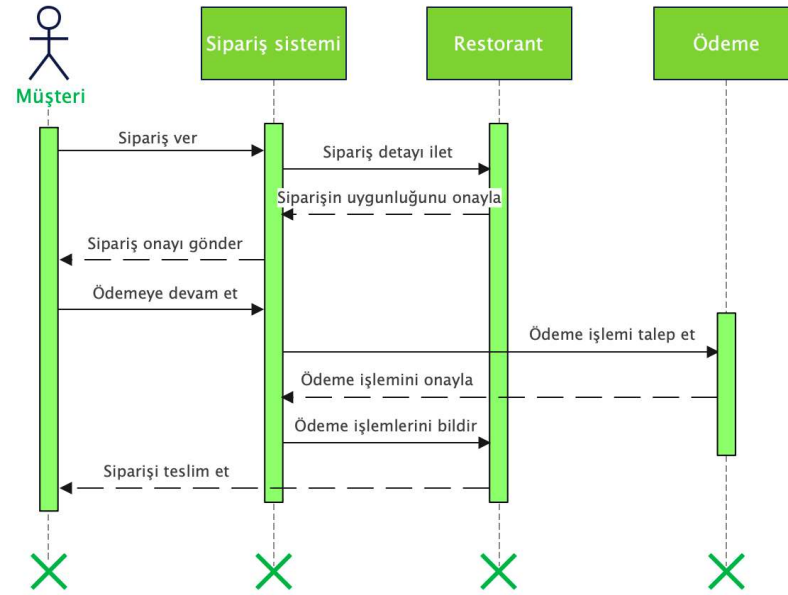




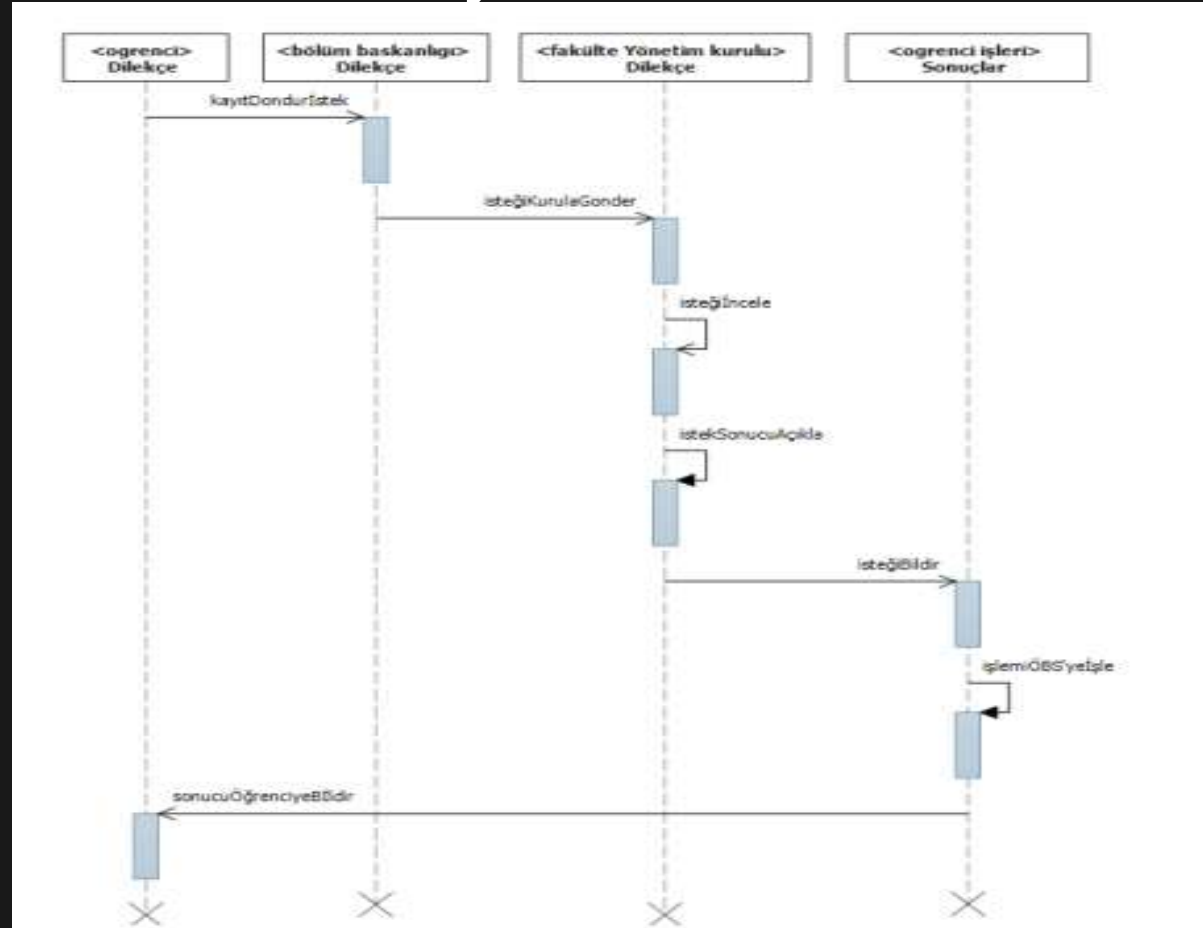
## Kayıt Sildirme İşlemleri Sıralama Diyagramı



# Sipariş Sıralama Diyagramı

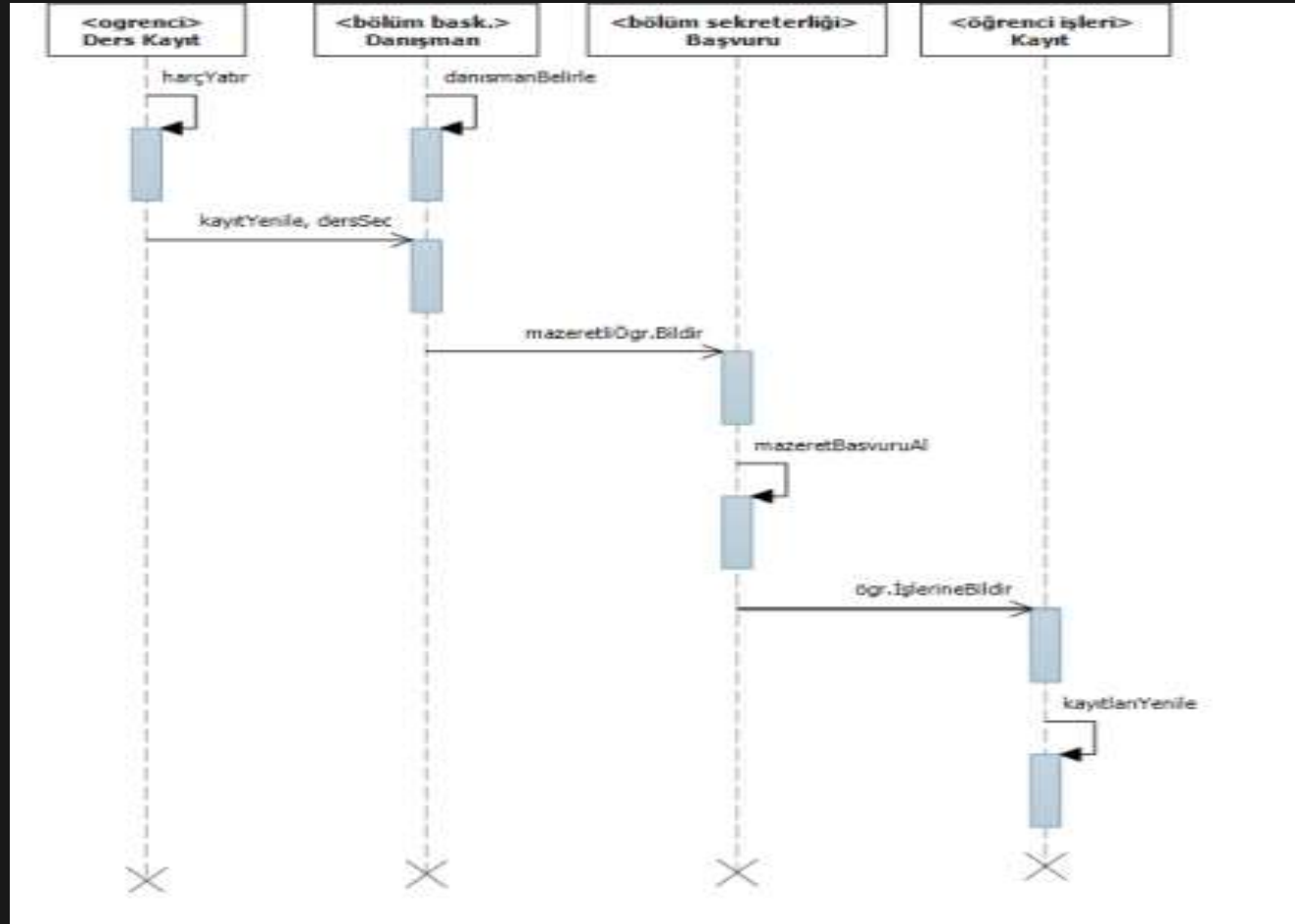


# Kayıt Dondurma İşlemleri Sıralama Diyagramı

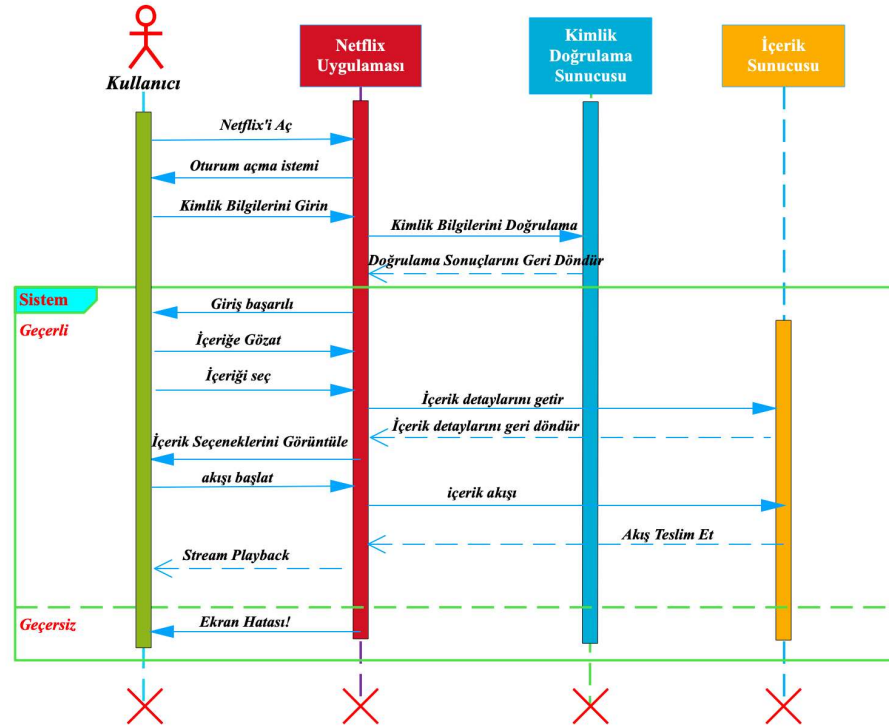




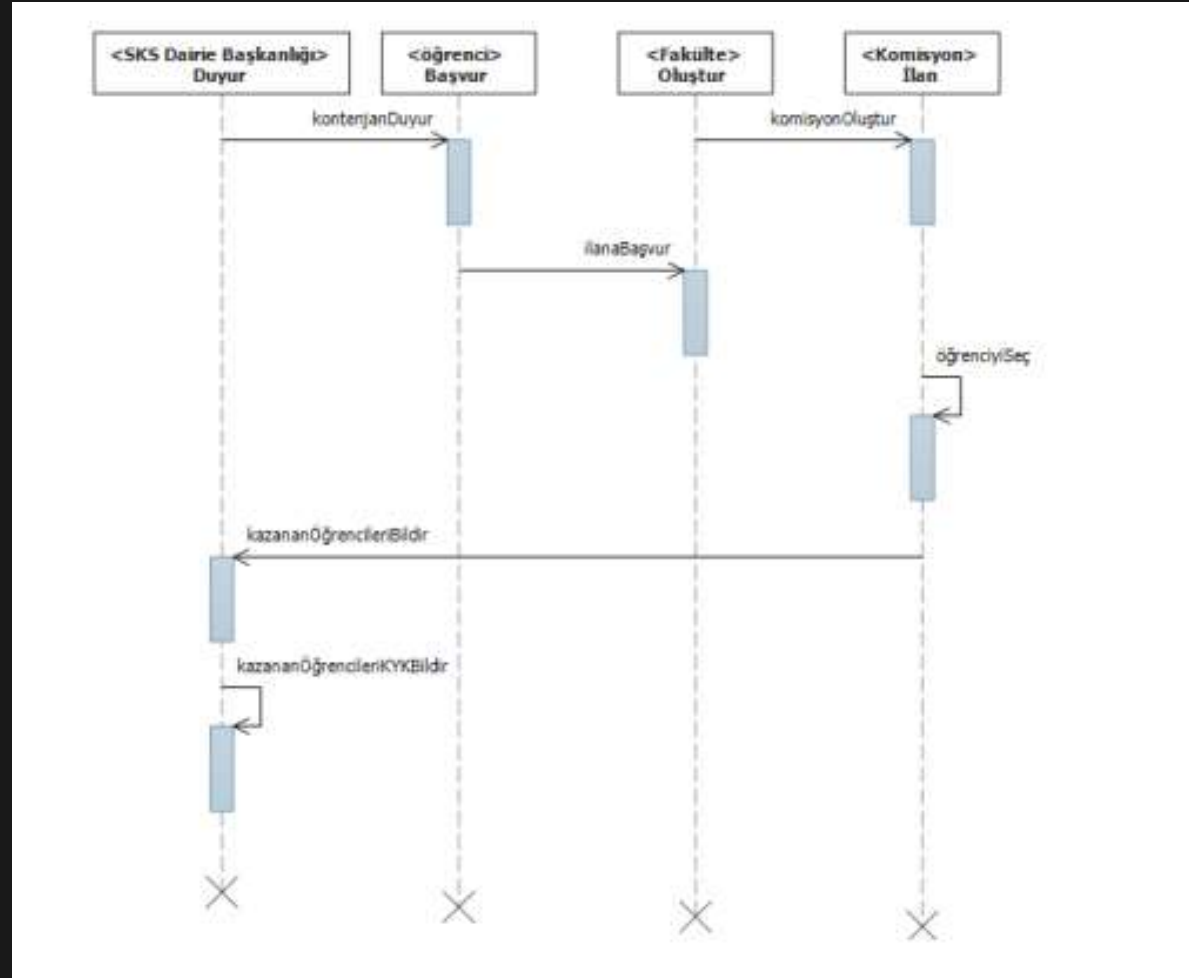
## Kayıt Yenileme İşlemleri Sıralama Diyagramı



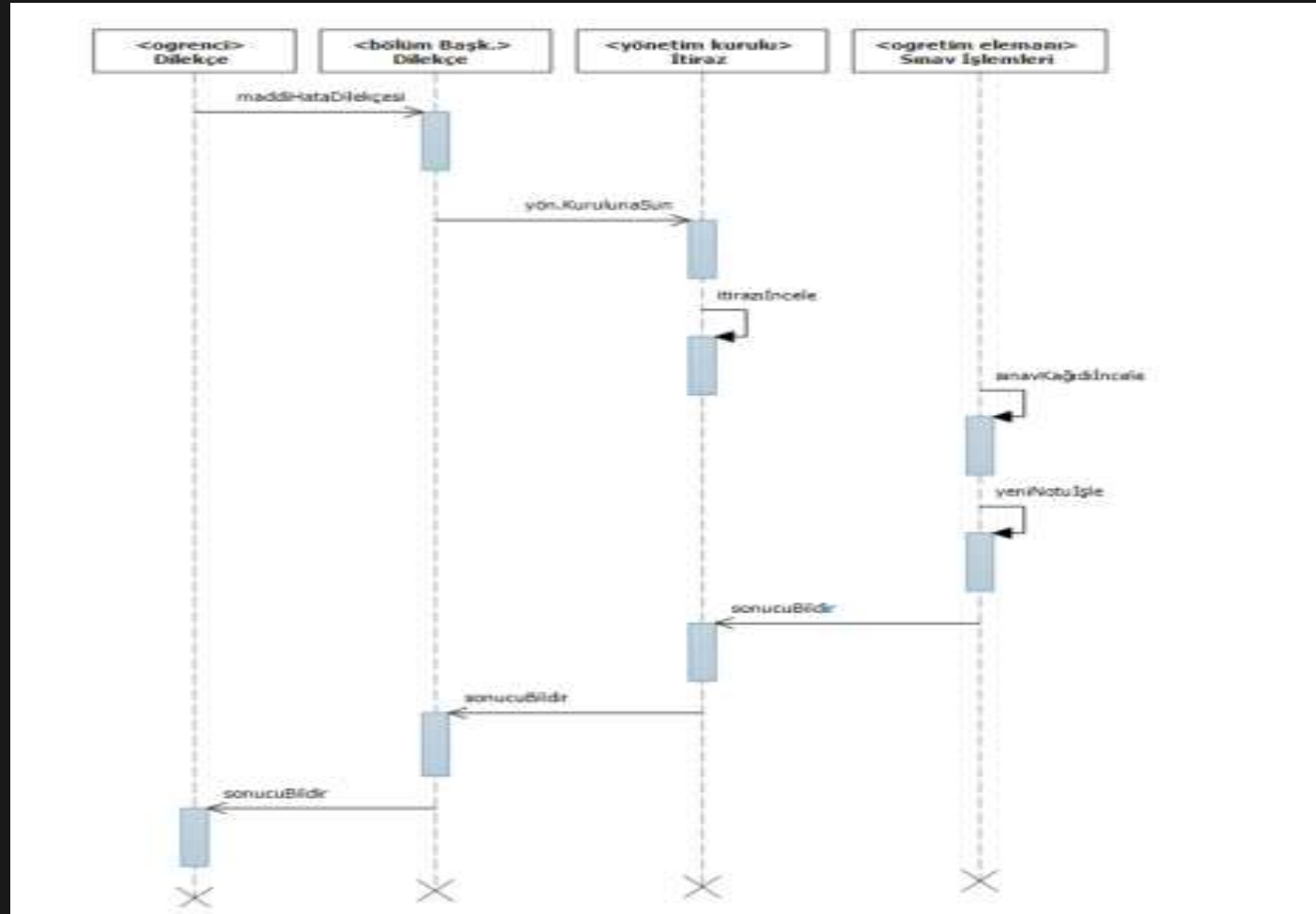
# Netflix için Sıralama Diyagramı



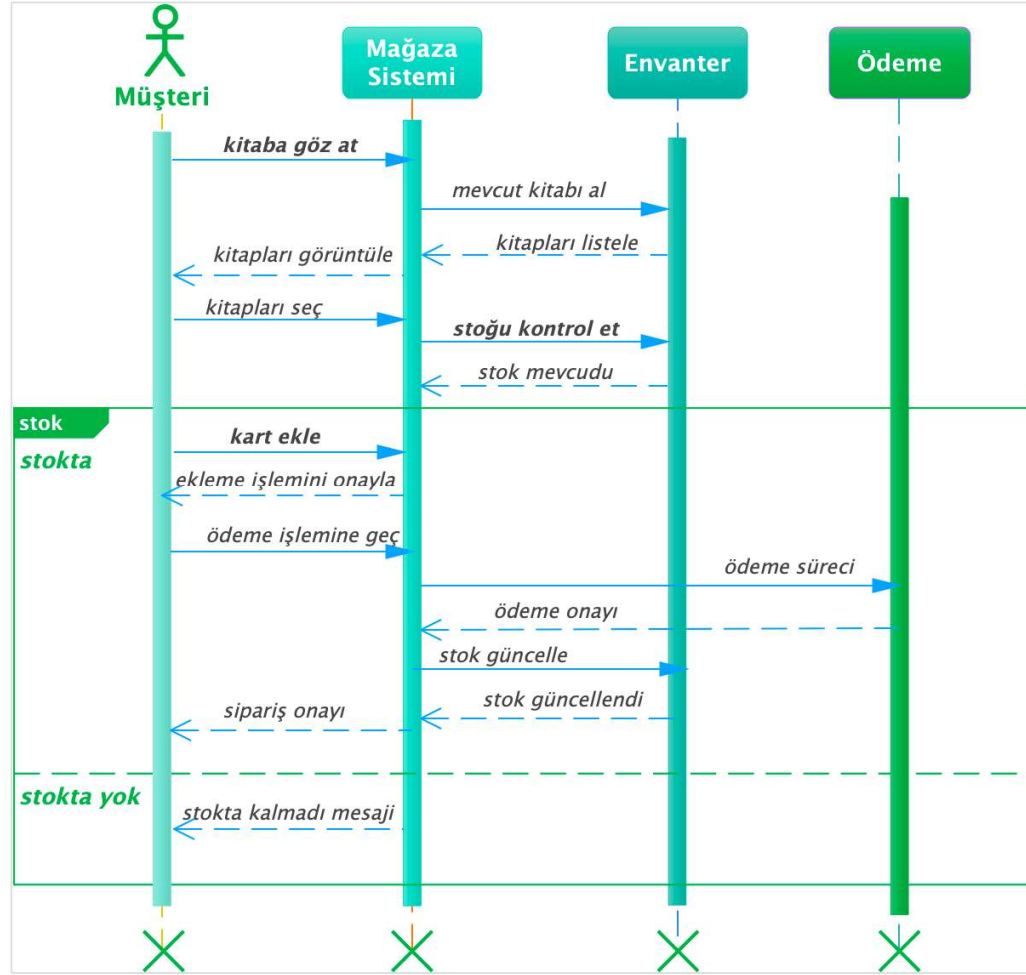
# Öğrencilerin Burs İşlemlerin Sıralama Diyagramı



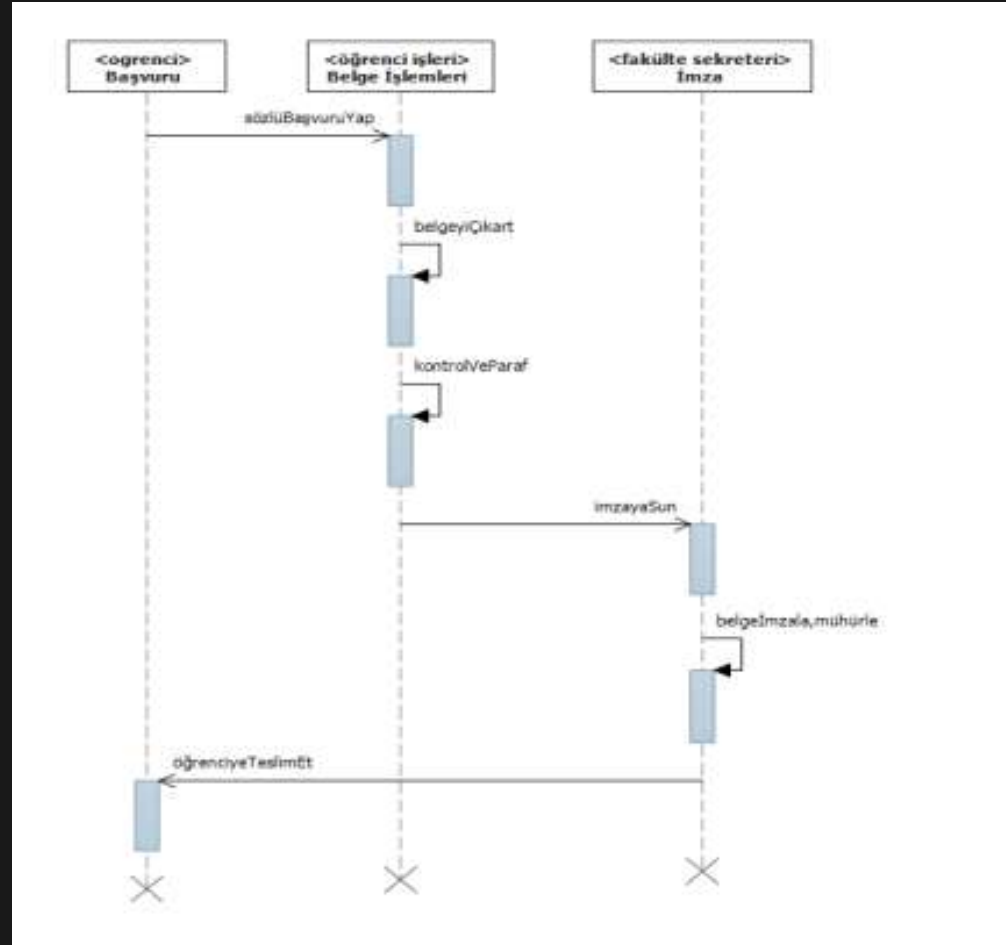
# Sınavlara İtiraz İşlemlerinin Sıralama Diyagramı



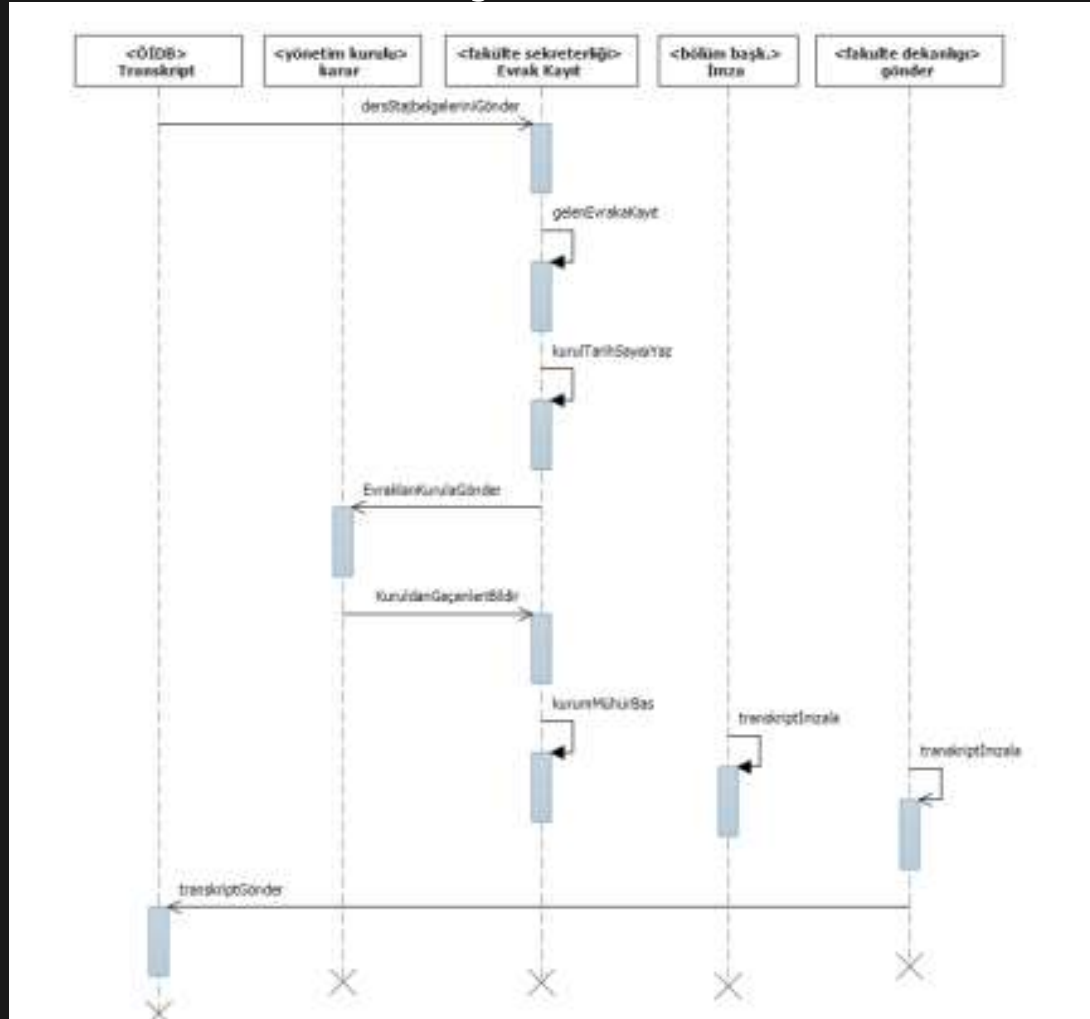
# Sıralama Diyagramı



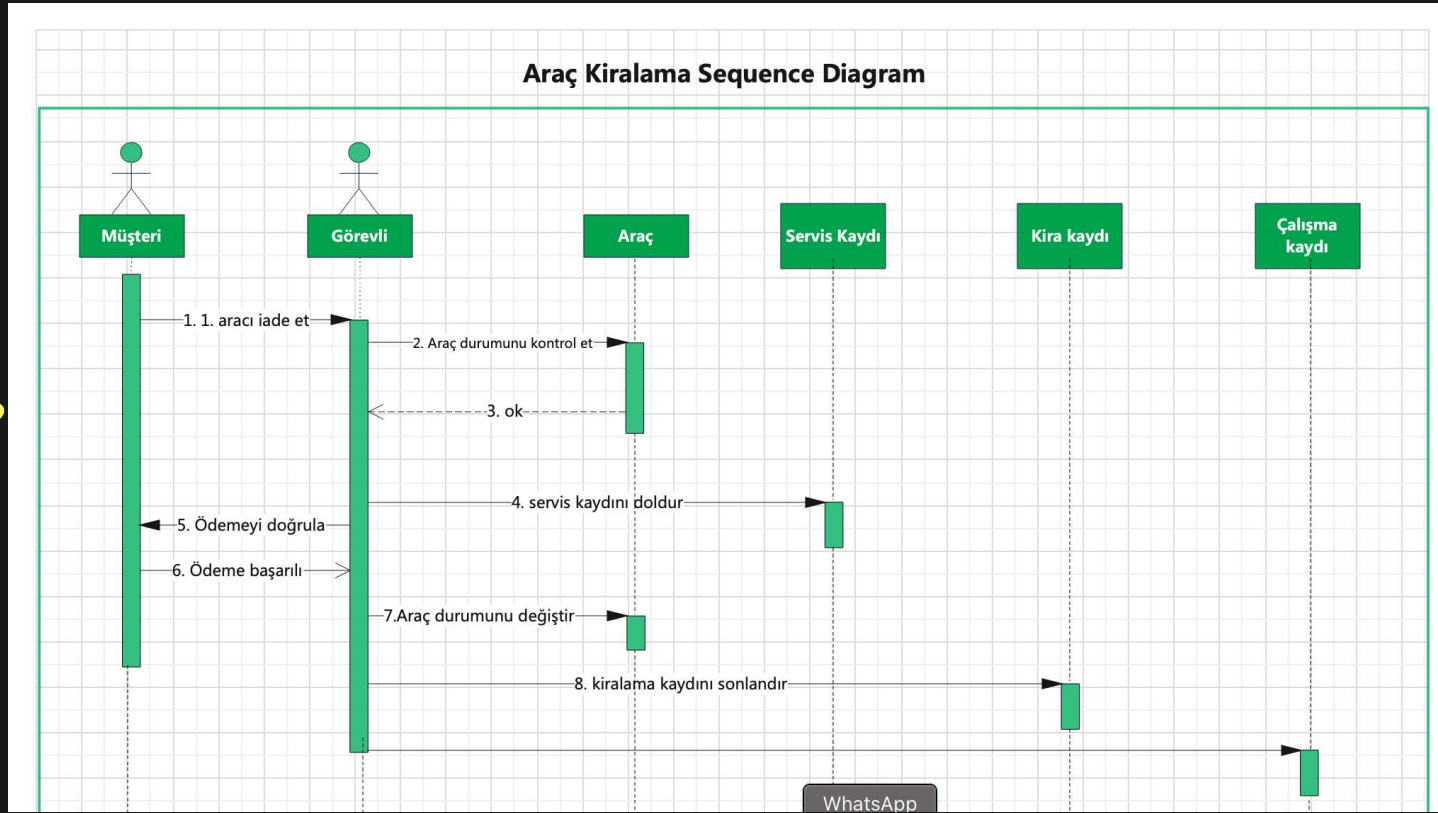
# Öğrenci Belgesi Verme İşlemlerinin Sıralama Diyagramı



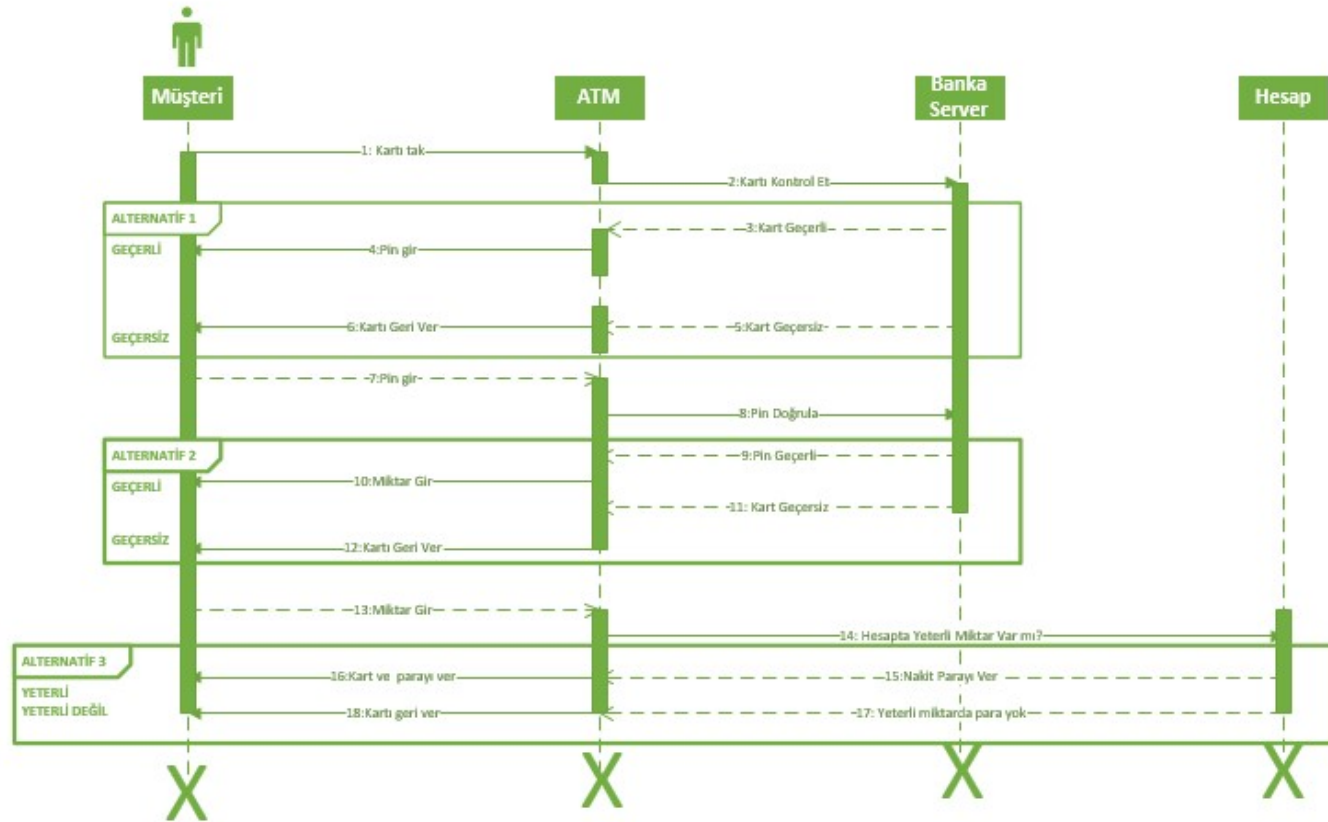
# Öğrenci Mezuniyet İşlemlerinin Sıralama Diyagramı



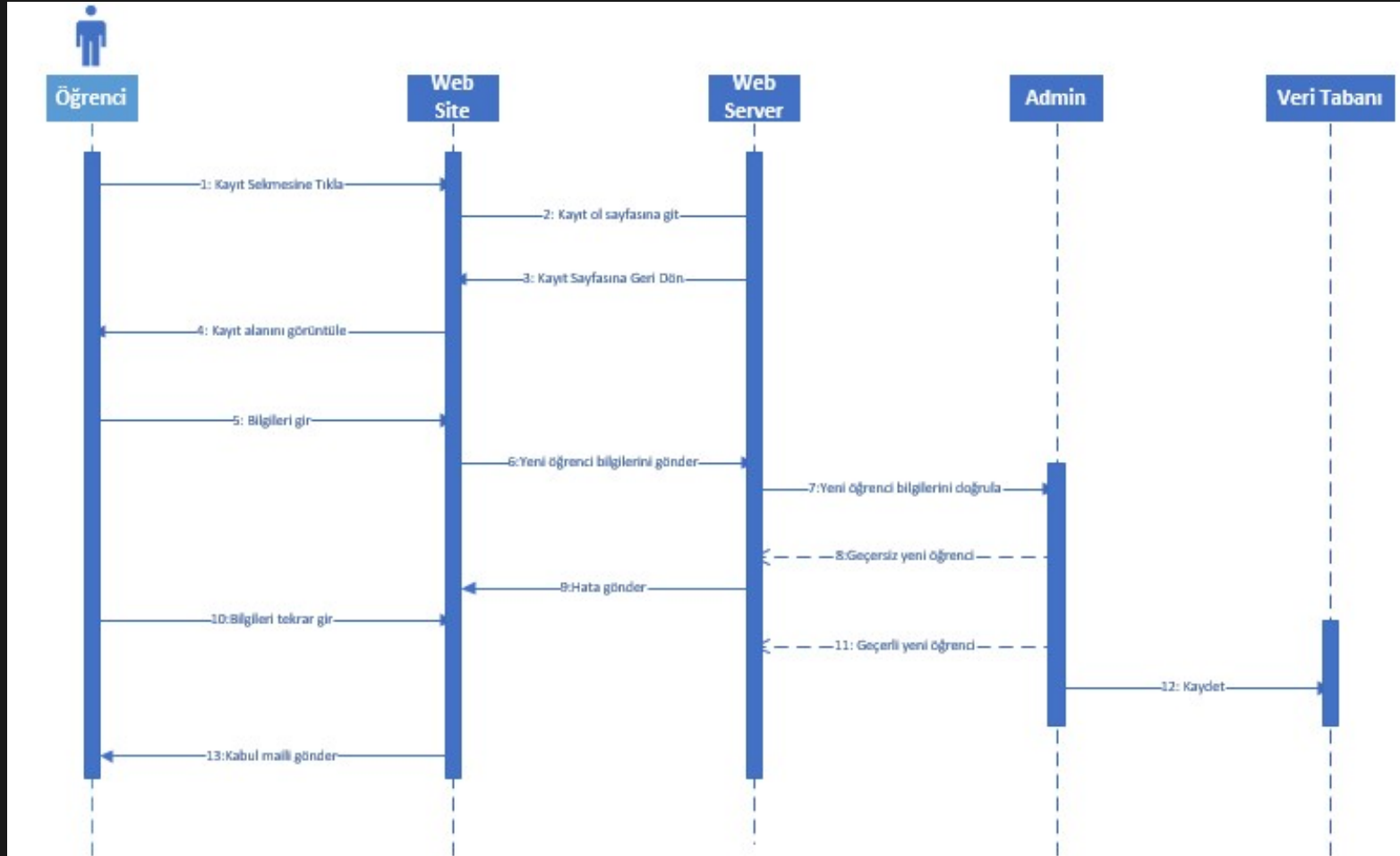
# Araç Kiralama Sıralama Diyagramı



# ATM Sıralama Diyagramı



# Öğrenci Kayıt Sıralama Diyagramı



# ATM Para Çekme Sıralama Diyagramı

