

## SEQUENCE DIAGRAM

- در دسته‌ی interaction diagrams قرار می‌گیرد. (collaboration diagram هم در این دسته قرار دارد).
- از جمله نمودارهای رفتاری است که ترتیب تعاملات بین اجزا (participant ها) را در حین اجرای یک task (برای مثال یک سناریوی usecase) مشخص می‌نماید.
- برخلاف نمودار usecase که به what می‌پردازد بر روی how تمرکز دارد.
- علاوه بر این، با استفاده از می‌توان طراحی سیستم را نیز بهبود داد. (برای مثال، تشخیص روابط پیچیده بین object ها)

## Sequence diagram

### ■ اجزای تشکیل دهنده

■ **Participants**: نمودار توالی از تعدادی participant تشکیل شده که در یک ترتیب و توالی معین با یکدیگر به تعامل می پردازند.

■ هر participant به صورت عمودی توسط یک lifeline مشخص میگردد.

■ **Lifeline** بیان کننده ی این است که یک participant در این نقطه از نمودار توالی قابلیت ارتباط با دیگر اجزا را دارست. با استفاده از lifeline مشخص میشود که یک participant به چه میزان در حین انجام یک task یا سناریو شرکت دارد.

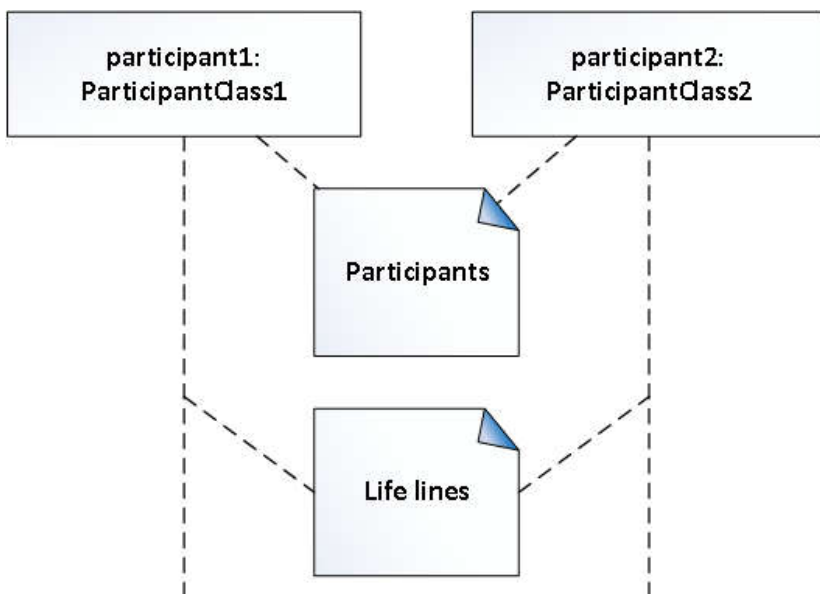
■ participant را می توان به صورت های زیر معرفی نمود:

■ participant1

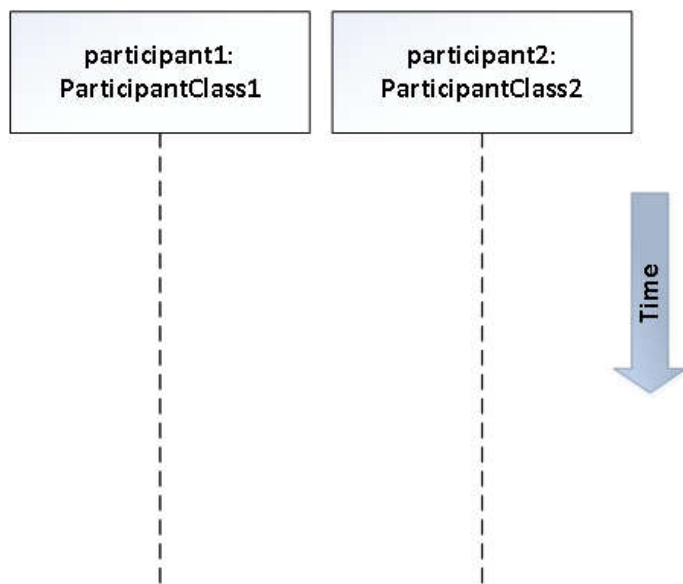
■ :ParticipantClass1

■ participant1:ParticipantClass1

➤ مشخص نمودن نام participant زمانی که ارجاع به آن لازم است، ضرورت دارد.



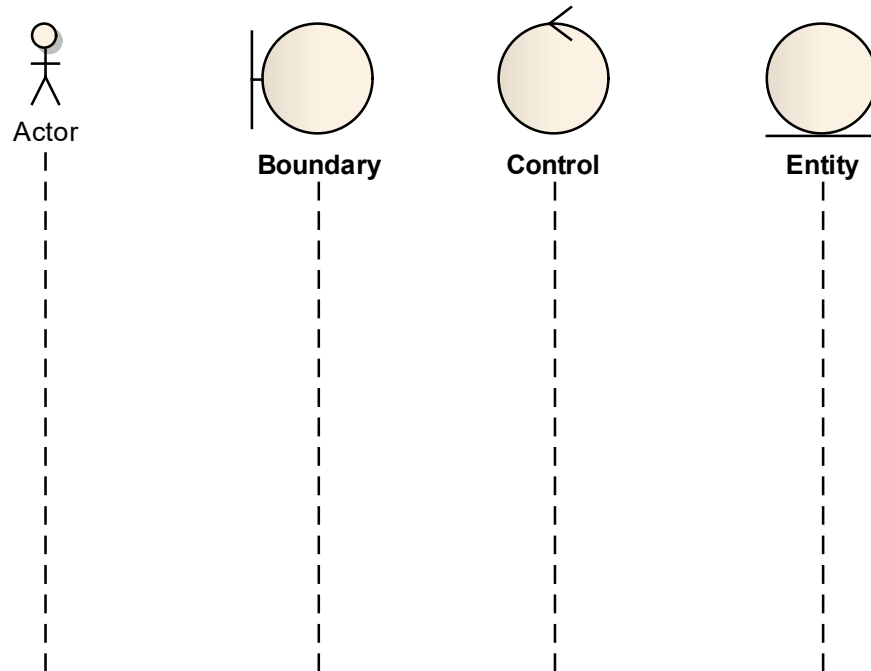
# SEQUENCE DIAGRAM



## ■ زمان

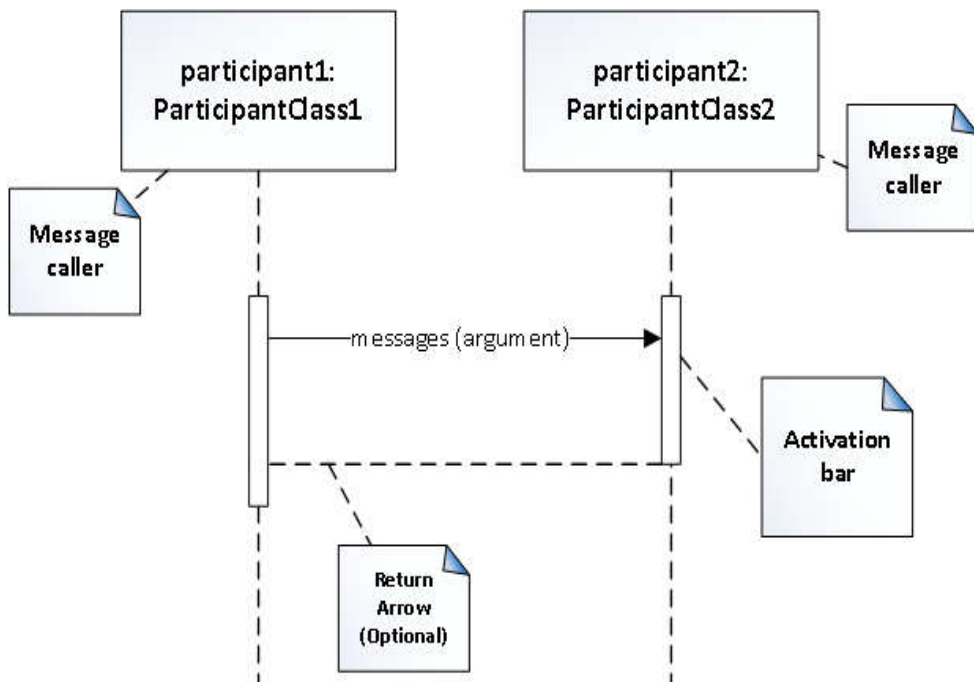
- با توجه به اینکه نمودار توالی ترتیب تعامل بین اجزا را نشان می‌دهد، زمان در این نمودار دارای اهمیت است.
- ترتیب بین پیغام‌ها به نحوی است که پیغامی که دیرتر فرستاده می‌شود در پایین تر قرار می‌گیرد.
- زمان در نمودار توالی تنها به لحاظ ترتیب و توالی دارای اهمیت نه از لحاظ duration.

# SEQUENCE DIAGRAM



■ انواع به خصوص participant

# SEQUENCE DIAGRAM



## Events and Messages and activation bars ■

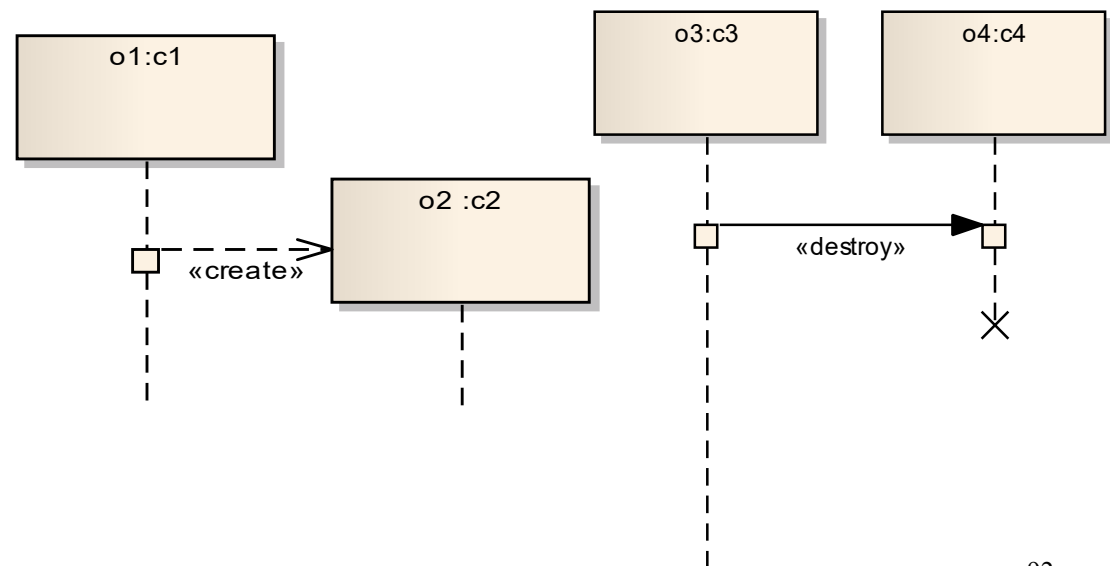
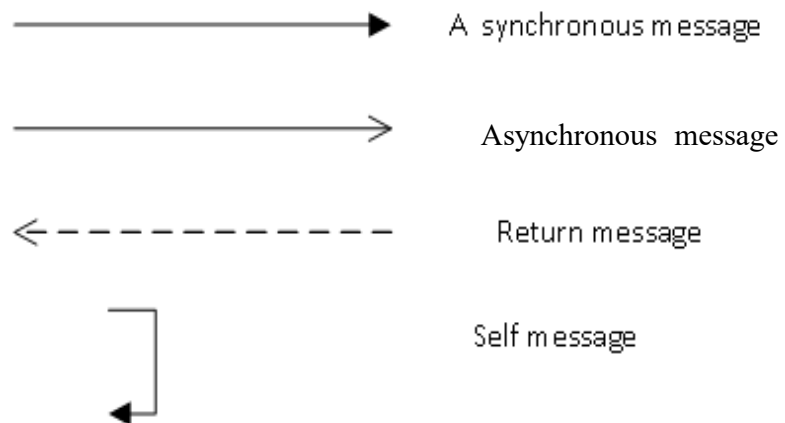
■ نمودار توالی مشخص میکند چگونه Event ها موجب تعامل بین participant ها می شود. (برای مثال انتقال پیغام ساده ترین نوع event است.)

■ زمانی که یک پیغام از یک participant به participant فرستاده می شود موجب trigger شدن یا فعال شدن آن برای انجام کاری به خصوصی میشود. این فعال شدن با activation bar نشان داده می شود.

■ قالب یک پیغام به صورت زیر است:

attribute = signal\_or\_operation\_name  
(arguments) : return\_value

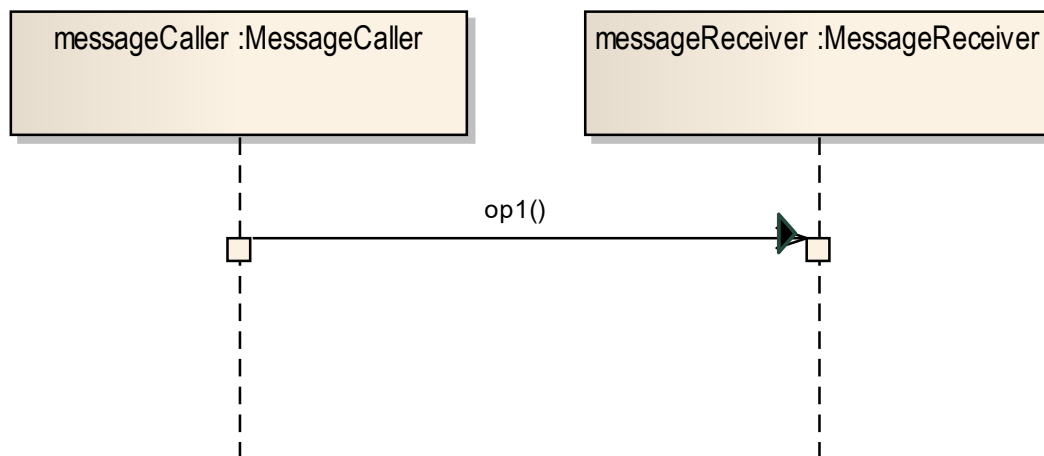
# SEQUENCE DIAGRAM



Messages ■

# SEQUENCE DIAGRAM

## ■ Synchronous messages



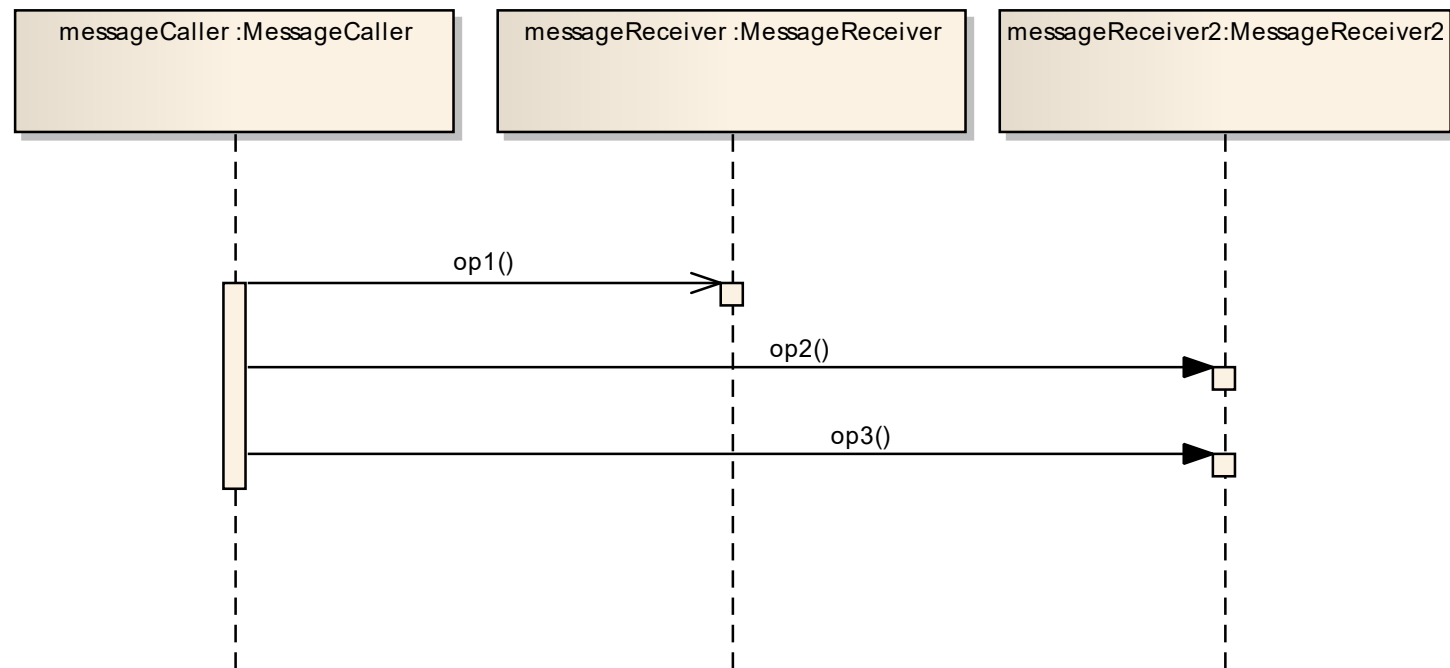
```
public class MessageReceiver
{
    public void op1( )
    {
        // Do something inside foo.
    }
}

public class MessageCaller
{
    private MessageReceiver messageReceiver;

    // Other Methods and Attributes of the class are declared here
    // The messageReceiver attribute is initialized elsewhere in
    // the class.
    public static void main (String[] args)
    {
        // The MessageCaller invokes the foo( ) method
        MessageCaller mc = new MessageCaller();
        mc.messageReceiver.op1( ); // then waits for the method to return
        // before carrying on here with the rest of its work
    }
}
```

# SEQUENCE DIAGRAM

- Asynchronous messages





# SEQUENCE DIAGRAM

- Asynchronous messages

```
public class MessageReceiver implements Runnable {
    public void operation1( ) {
        // Receive the message and trigger off the thread

        Thread op1Worker = new Thread(this);
        op1Worker.start(); // This call starts a new thread, calling the run( )
                           // method below
        // As soon as the thread has been started, the call to op1( ) returns.
    }
    public void run( ) {
        // This is where the work for the op1( ) message invocation will be executed.
    }
}
```

```
public class MessageCaller
{
    private MessageReceiver messageReceiver;
    private MessageReceiver messageReceiver2;
    // Other Methods and Attributes of the class are declared here

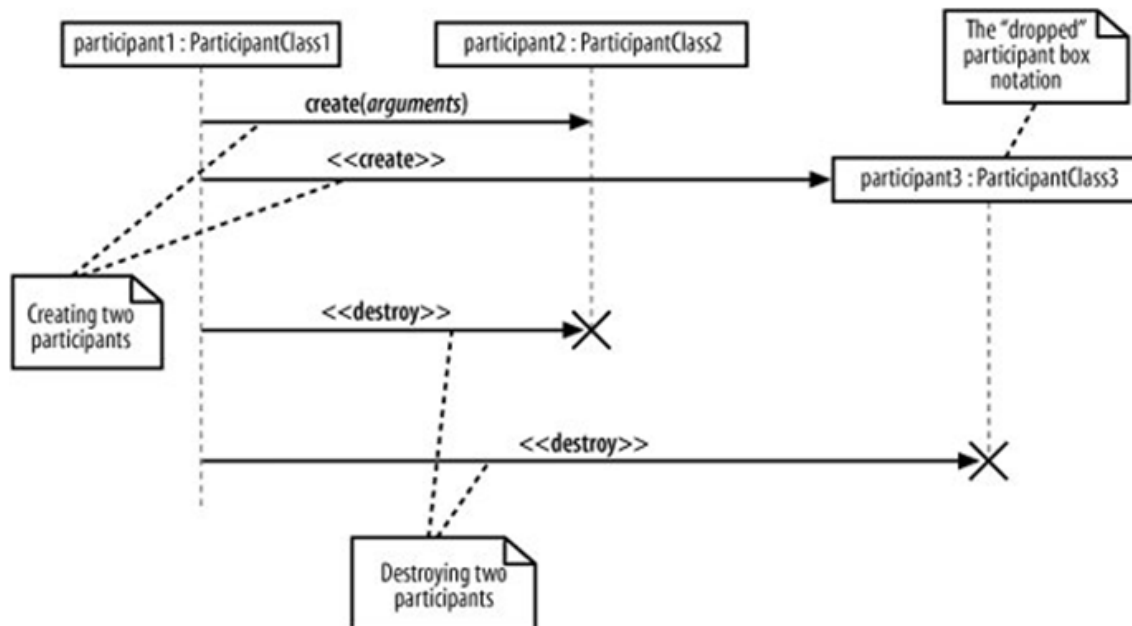
    // The messageReceiver attribute is initialized elsewhere in
    // the class.
    public static void main(String[] args) {
        // The MessageCaller invokes the operation1( ) operation

        mc.messageReceiver.operation1( );

        // then immediately carries on with the rest of its work
        mc.messageReceiver2.op2( );
    }
}
```

# SEQUENCE DIAGRAM

- Create and destroy message

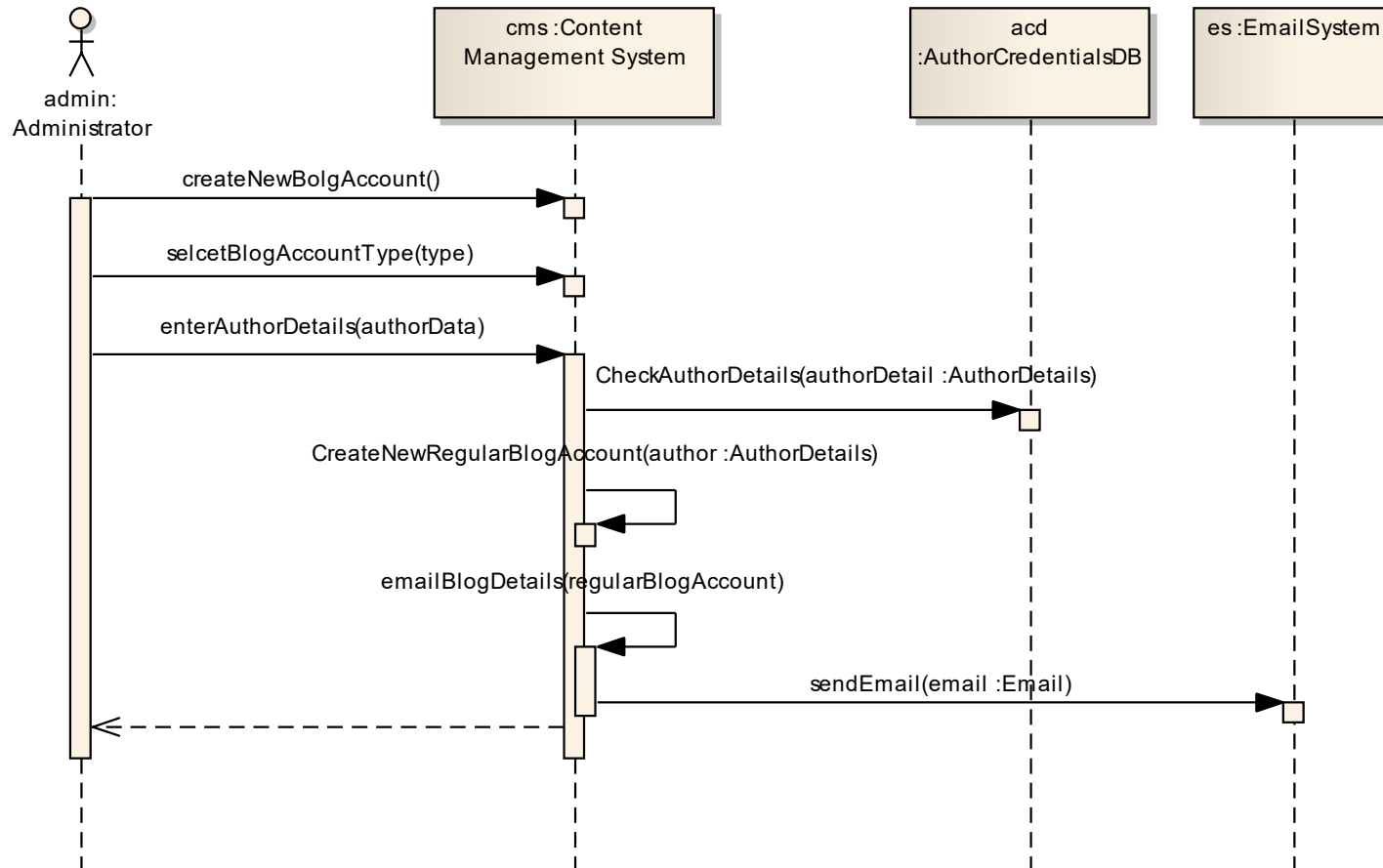


# USE CASE DIAGRAM AND SEQUENCE DIAGRAM

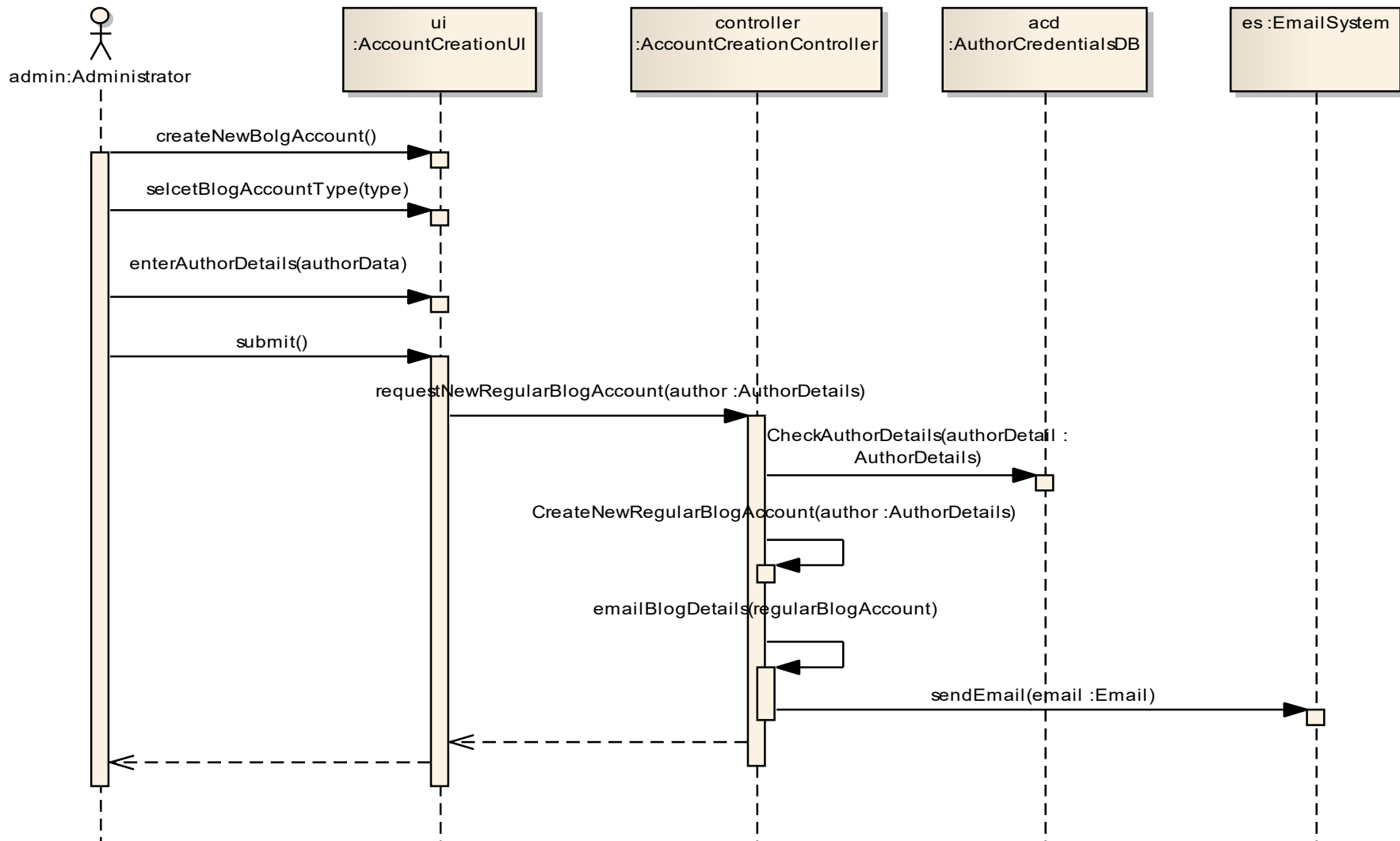
## ■ Use Case : Craete a New Regular Blog Account

Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects the regular blog account type.
	3	The Administrator enters the author's details.
	4	The author's details are checked using the Author Credentials Database.
	5	The new regular blog account is created.
	6	A summary of the new blog account's details are emailed to the author.

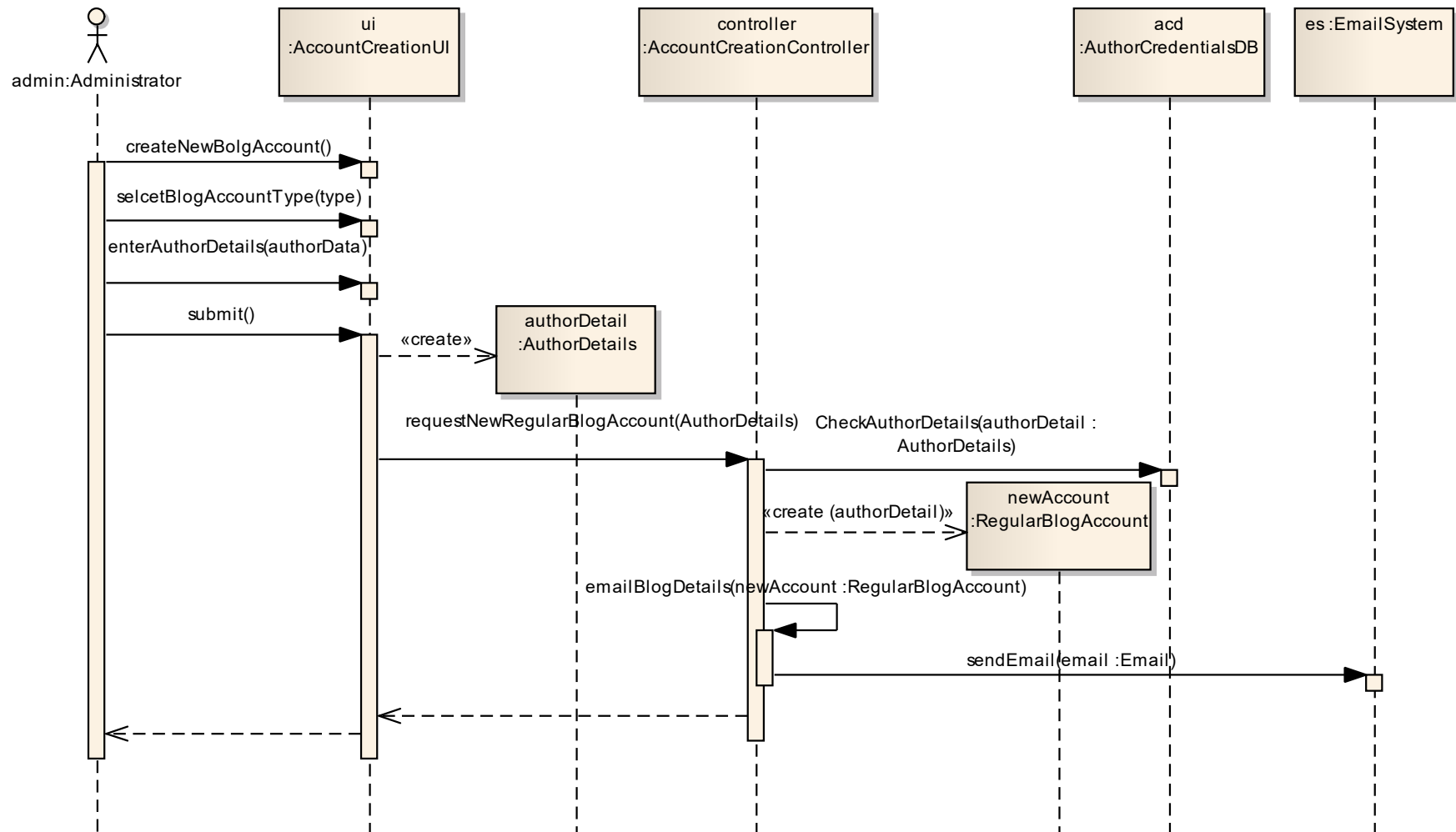
# Sequence diagram



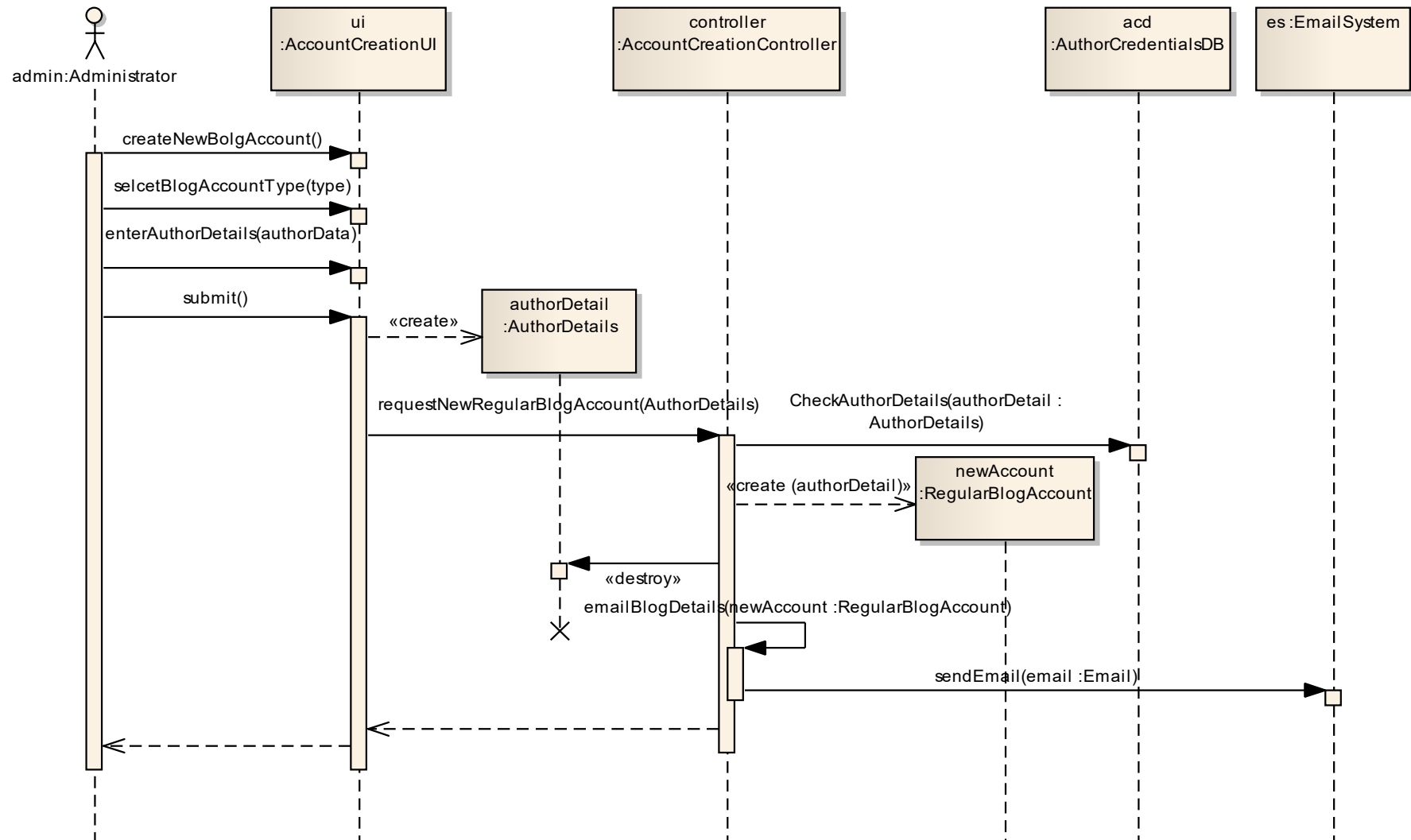
# Sequence diagram



## Sequence diagram-Creation message



## Sequence diagram- Deletion message



## Sequence diagram

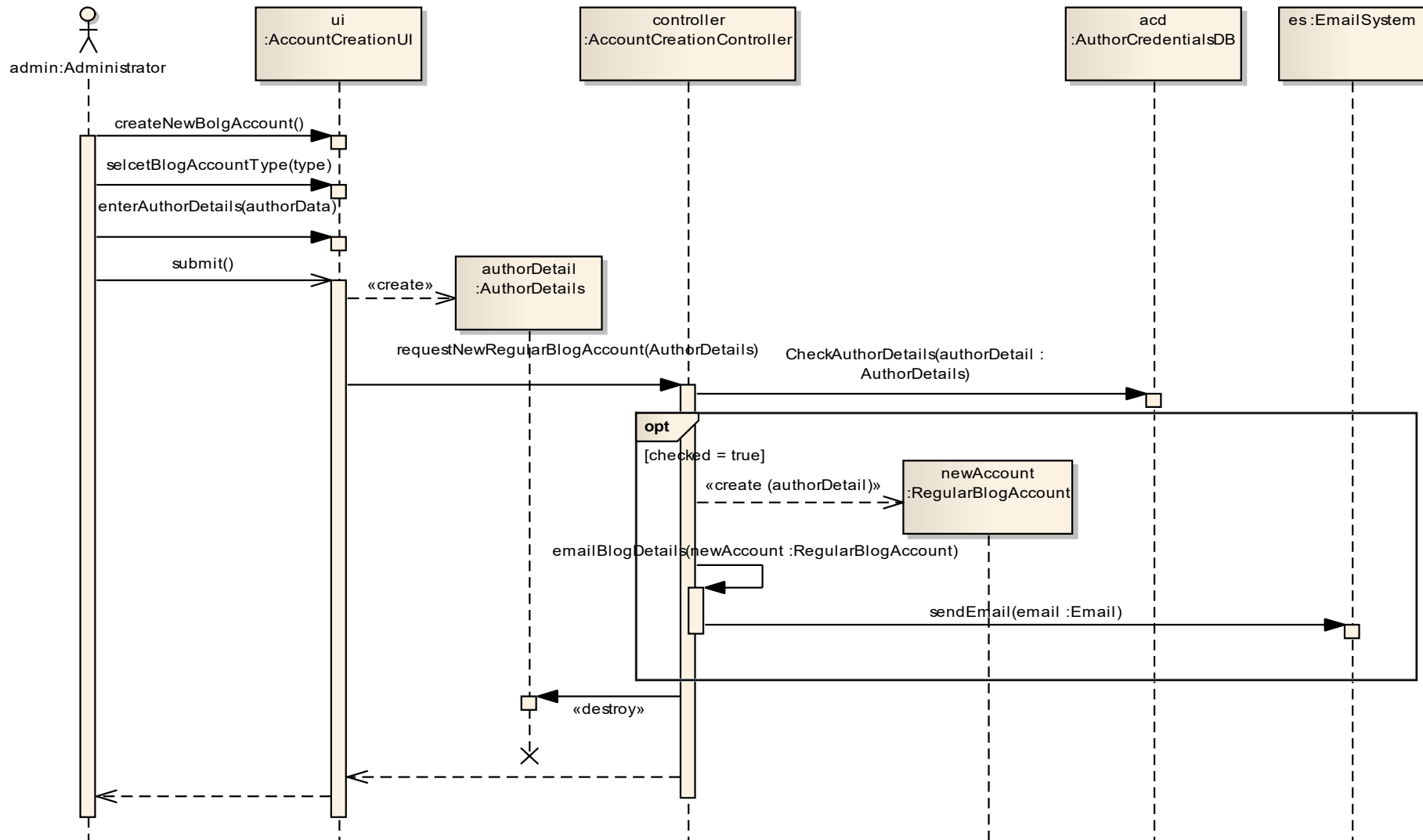
### Combined Fragments ■

- شامل یک یا چند دنباله پردازی است که در داخل یک frame جمع شده و از طریق آن تعاملات پیچیده را می توان به نحو ساده تری مدل نمود. هر combined fragment دارای یک interaction operand است که نحوه ی به خصوصی از تعاملات را فراهم می آورد. مهم ترین این operand ها عبارتند از:

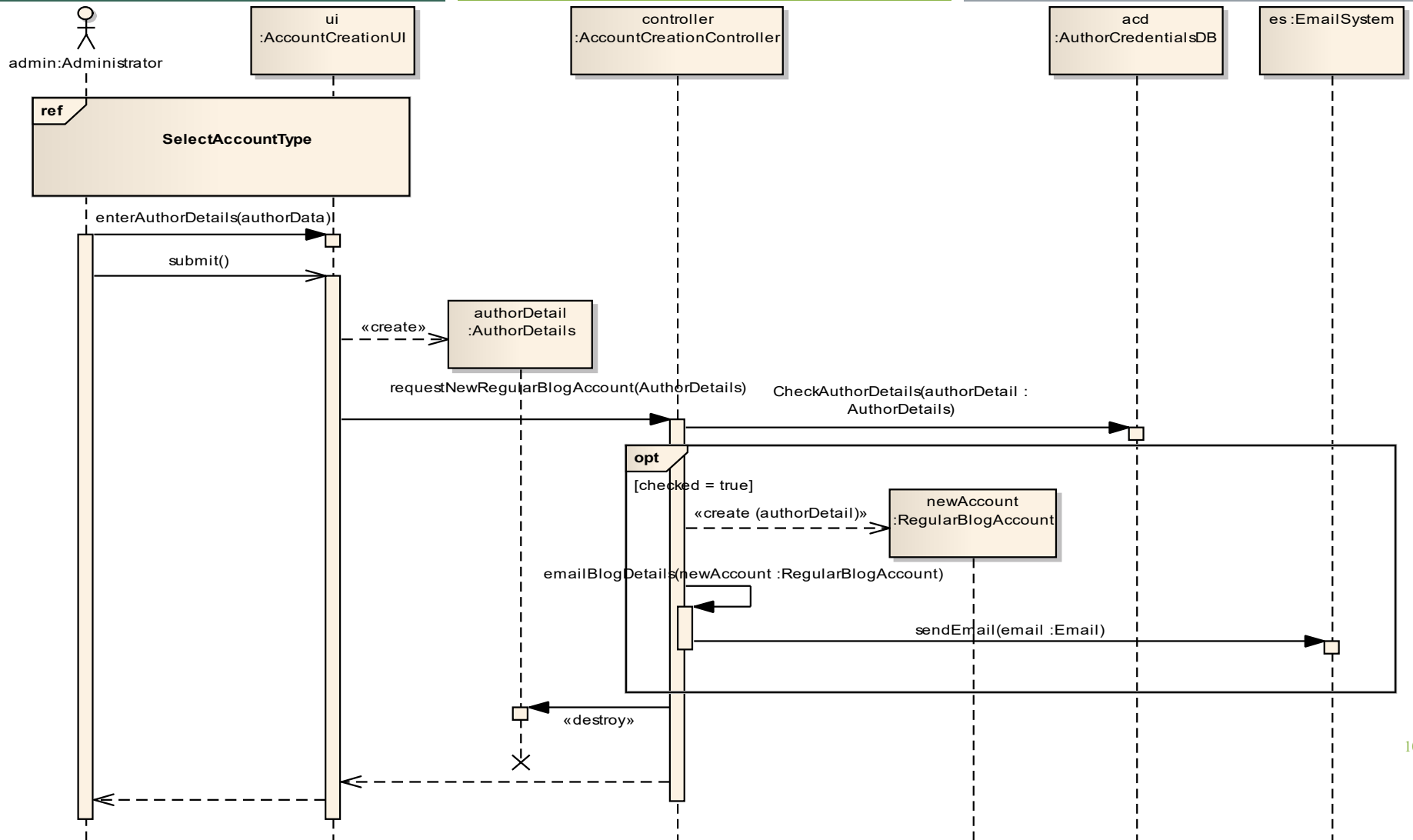
ref		بیان کننده ی یک مجموعه از تعاملات است که در نمودار دیگری تعریف شده است.
loop	min times, max times, [guard_condition]	با استفاده از آن می توان یک مجموعه از تعاملات را تا زمانی که شرط به خصوصی هنوز false نشده است چندین مرتبه تکرار نمود.
alt	[guard_condition1] ... [guard_condition2] ... [else]	زمانی که بسته به شروط مختلف تعاملات گوناگونی بین اجزا صورت می گیرد، استفاده میشود. ( مانند if then else )
opt	[guard_condition1] ...	تعاملاتی که در داخل این fragment قرار می گیرند فقط زمانی اجرا می شوند که شروط مورد نظر برقرار شود.



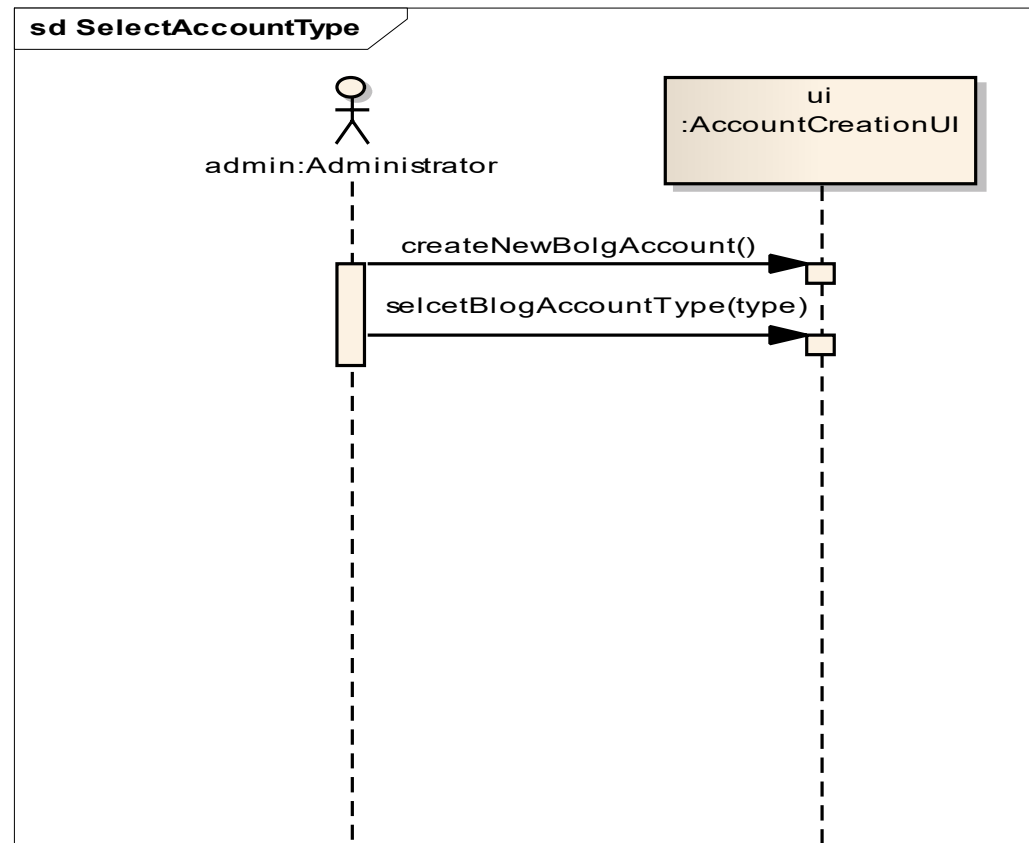
## Sequence diagram-opt fragment



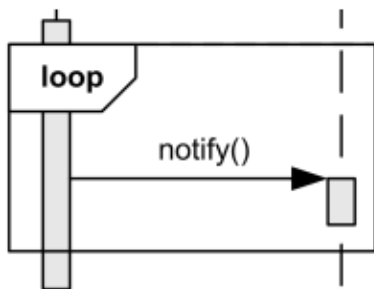
## Sequence diagram-ref fragment



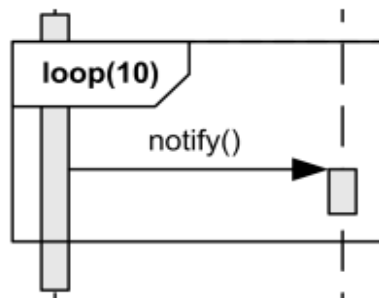
## Sequence diagram-ref fragment



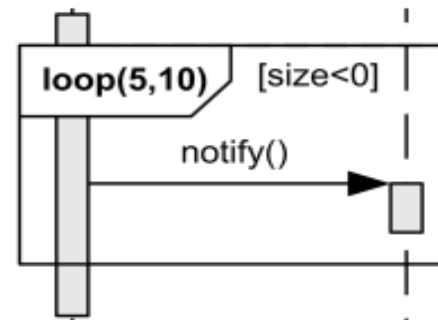
# SEQUENCE DIAGRAM



حلقه نامتناهی

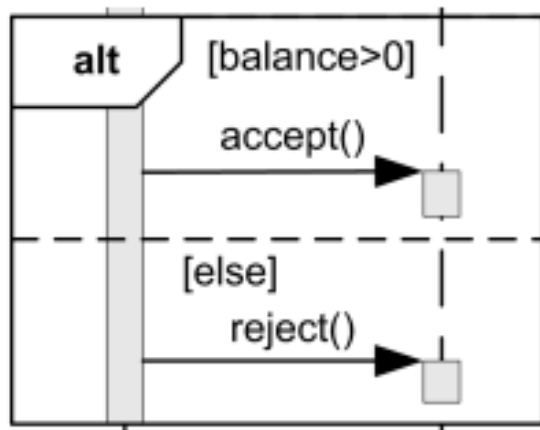


حلقه‌ای که دقیقاً 10 بار تکرار می‌شود.



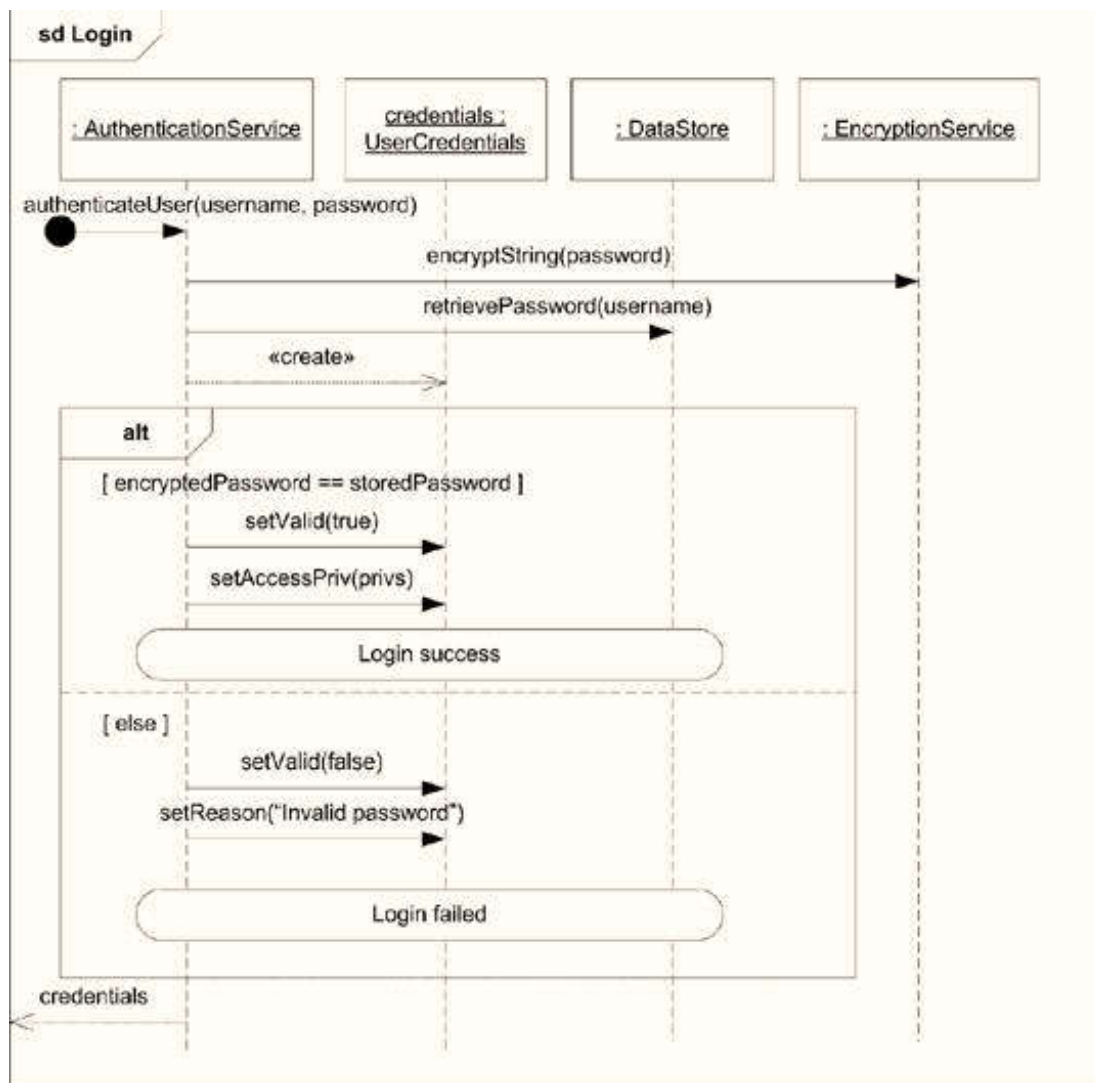
حلقه‌ای که به شرط برآورده سازی شرط حداقل 5 و حداکثر 10 بار اجرا می‌شود.

Loop fragment ■



Alternative ■

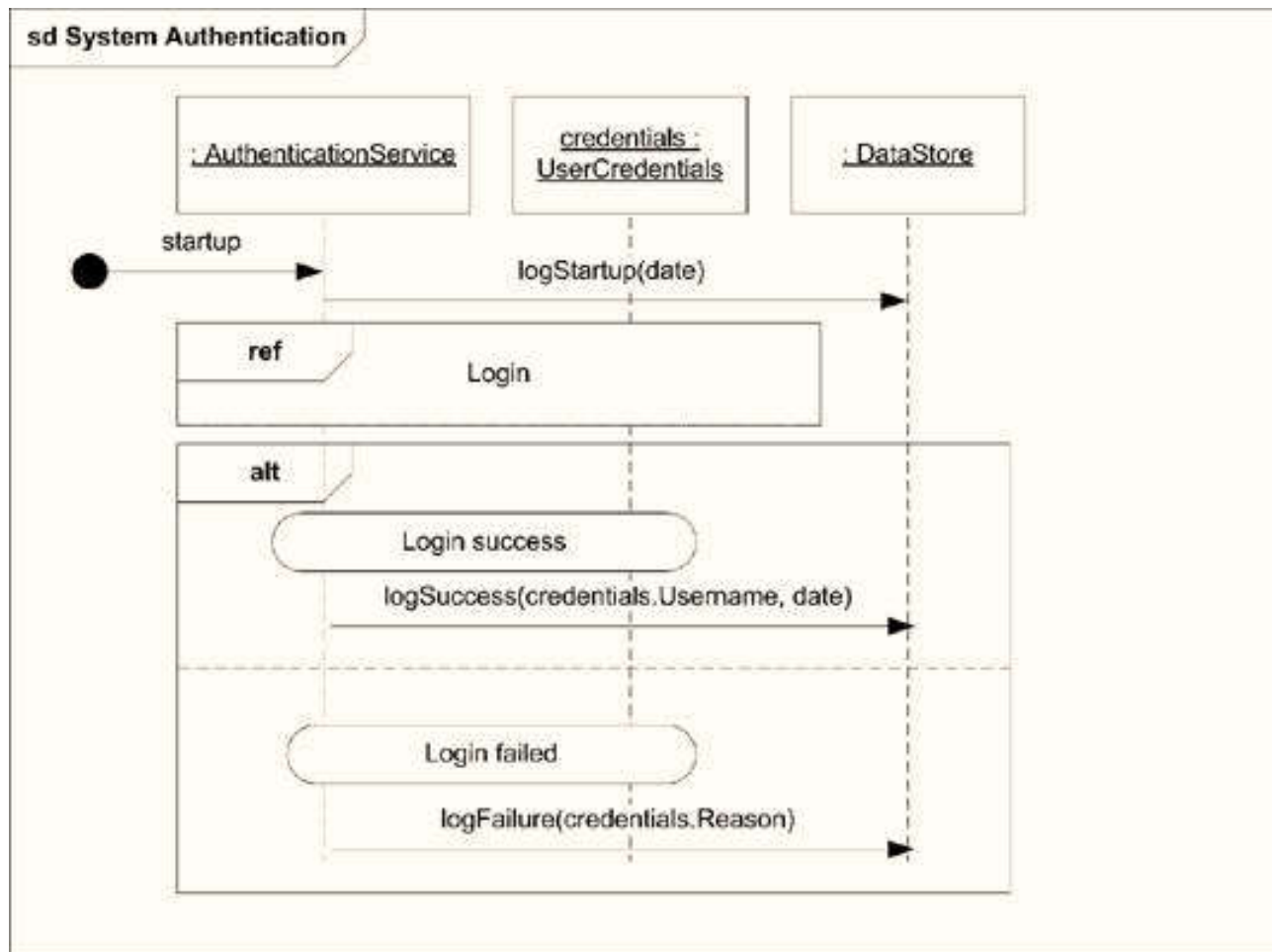
## Sequence diagram



### Continuations

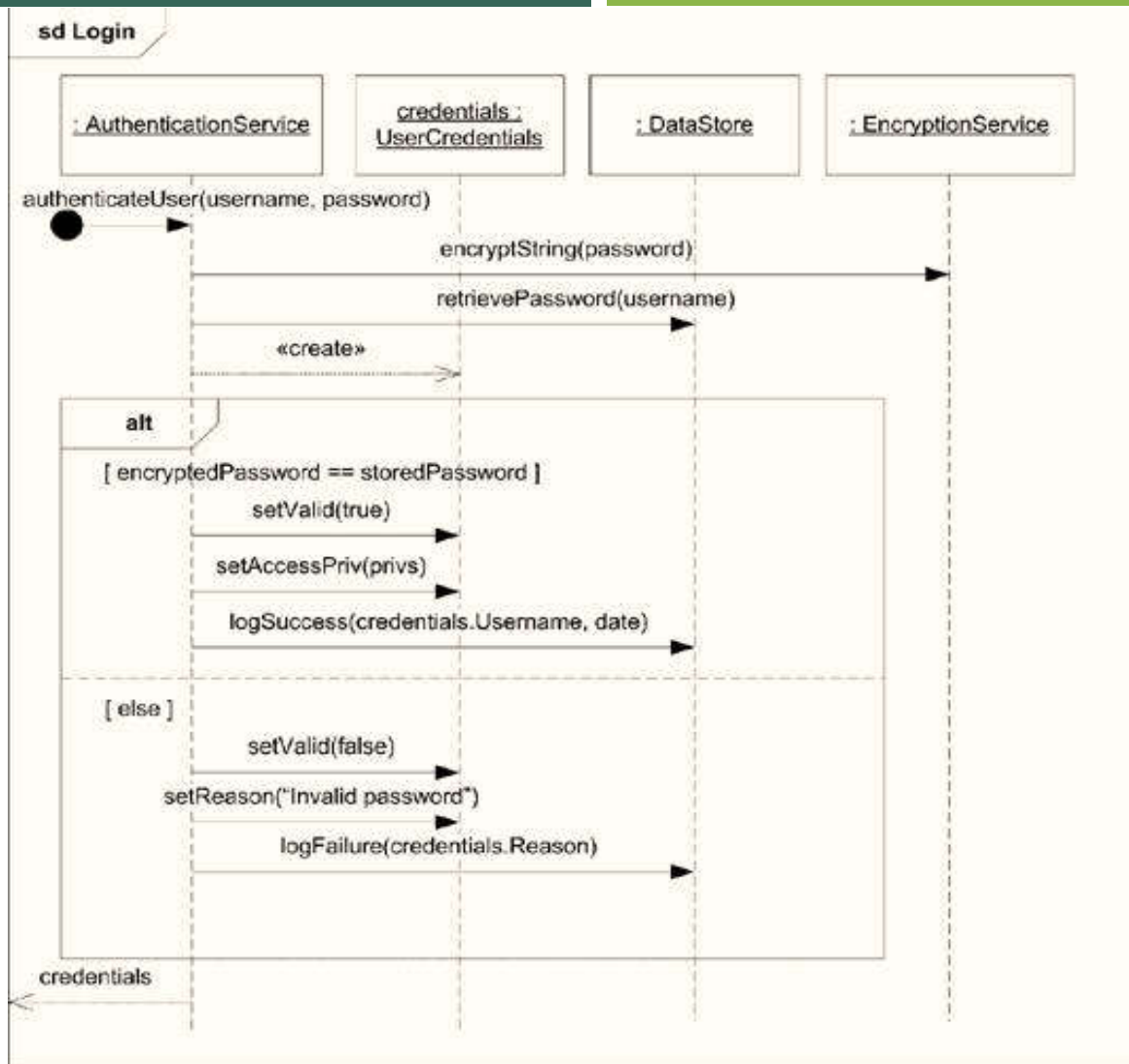
- با استفاده از آن شاخه‌های متعددی برای یک `alternative` در خارج از آن تعریف نمود.

## Sequence diagram



Continuations ■

## Sequence diagram



Continuations ■