



(<https://creately.com/blog/>)

Creately Blog (<https://creately.com/blog/>) > diagrams (<https://creately.com/blog/category/diagrams/>)

Sequence Diagram Tutorial: Complete Guide with Examples

Updated on: 19 August 2020

This sequence diagram tutorial is to help you understand sequence diagrams better; to explain everything you need to know, from how to draw a sequence diagram (<https://creately.com/diagram-type/sequence-diagram>) to the common mistakes you should avoid when drawing one.

There are 3 types of Interaction diagrams; Sequence diagrams, communication diagrams, and timing diagrams. These diagrams are used to illustrate interactions between parts within a system. Among the three, sequence diagrams are preferred by both developers and readers alike for their simplicity.

In this sequence diagram tutorial you will learn about;

- What is a Sequence Diagram
- Sequence Diagram Notations
- Sequence Diagram Best Practices
- How to Draw a Sequence Diagram
- Sequence Diagram Common Mistakes
- Sequence Diagram Templates and Examples
- Sequence Diagram SlideShare Presentation
- Feedback on the Sequence Diagram Guide

What is a Sequence Diagram?

Sequence diagrams, commonly used by developers, model the interactions between objects in a single use case. They illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur

when a particular use case is executed.

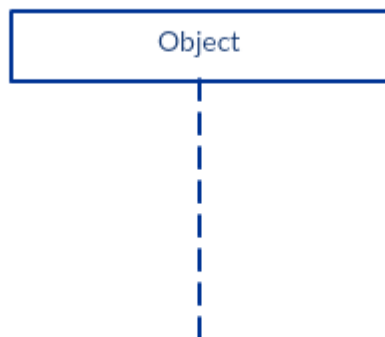
In simpler words, a sequence diagram shows different parts of a system work in a 'sequence' to get something done.

Sequence Diagram Notations

A sequence diagram is structured in such a way that it represents a timeline which begins at the top and descends gradually to mark the sequence of interactions. Each object has a column and the messages exchanged between them are represented by arrows.

A Quick Overview of the Various Parts of a Sequence Diagram

Lifeline Notation



A sequence diagram is made up of several of these lifeline notations that should be arranged horizontally across the top of the diagram. No two lifeline notations should overlap each other. They represent the different objects or parts that interact with each other in the system during the sequence.

A lifeline notation with an actor element symbol is used when the particular sequence diagram is owned by a use case.



A lifeline with an entity element represents system data. For example, in a customer service application, the Customer entity would manage all data related to a customer.



A lifeline with a boundary element indicates a system boundary/ software element in a system; for example, user interface screens, database gateways or menus that users interact with, are boundaries.



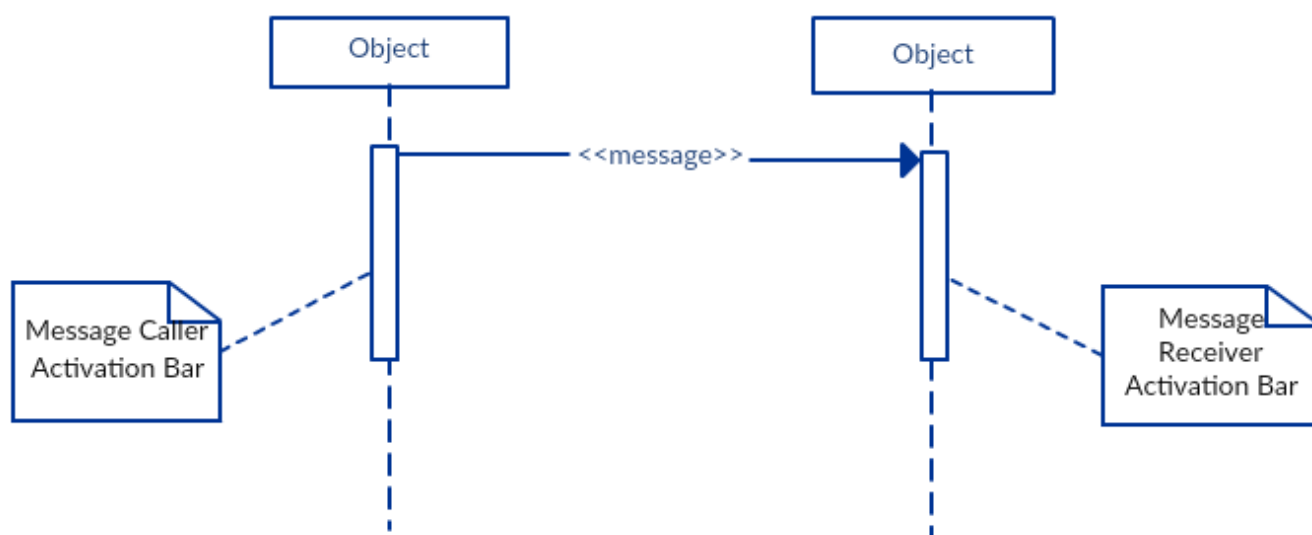
And a lifeline with a control element indicates a controlling entity or manager. It organizes and schedules the interactions between the boundaries and entities and serves as the mediator between them.



Activation Bars

Activation bar is the box placed on the lifeline. It is used to indicate that an object is active (or instantiated) during an interaction between two objects. The length of the rectangle indicates the duration of the objects staying active.

In a sequence diagram, an interaction between two objects occurs when one object sends a message to another. The use of the activation bar on the lifelines of the Message Caller (the object that sends the message) and the Message Receiver (the object that receives the message) indicates that both are active/is instantiated during the exchange of the message.



Message Arrows

An arrow from the Message Caller to the Message Receiver specifies a message in a sequence diagram. A message can flow in any direction; from left to right, right to left or back to the Message Caller itself. While you can describe the message being sent from one object to the other on the arrow, with different arrowheads you can indicate the type of message being sent or received.

The message arrow comes with a description, which is known as a message signature, on it. The format for this message signature is below. All parts except the message_name are optional.

attribute = message_name (arguments): return_type

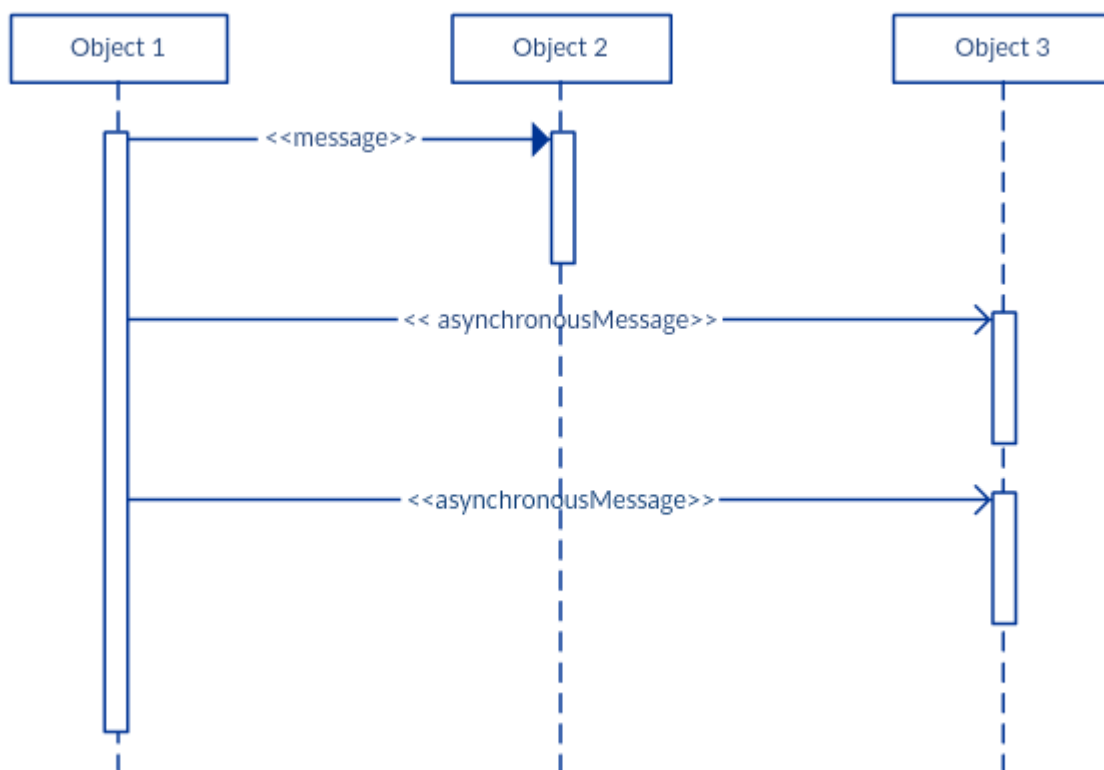
- *Synchronous message*

As shown in the activation bars example, a synchronous message is used when the sender waits for the receiver to process the message and return before carrying on with another message. The arrowhead used to indicate this type of message is a solid one, like the one below.



- *Asynchronous message*

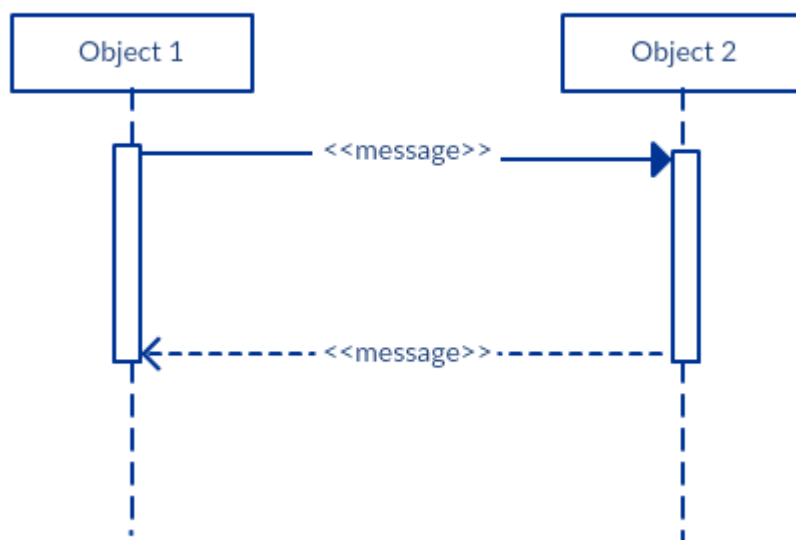
An asynchronous message is used when the message caller does not wait for the receiver to process the message and return before sending other messages to other objects within the system. The arrowhead used to show this type of message is a line arrow like shown in the example below.



- *Return message*

A return message is used to indicate that the message receiver is done processing the message and is returning control over to the message caller. Return messages are optional notation pieces, for an activation bar that is triggered by a synchronous message always implies a return message.

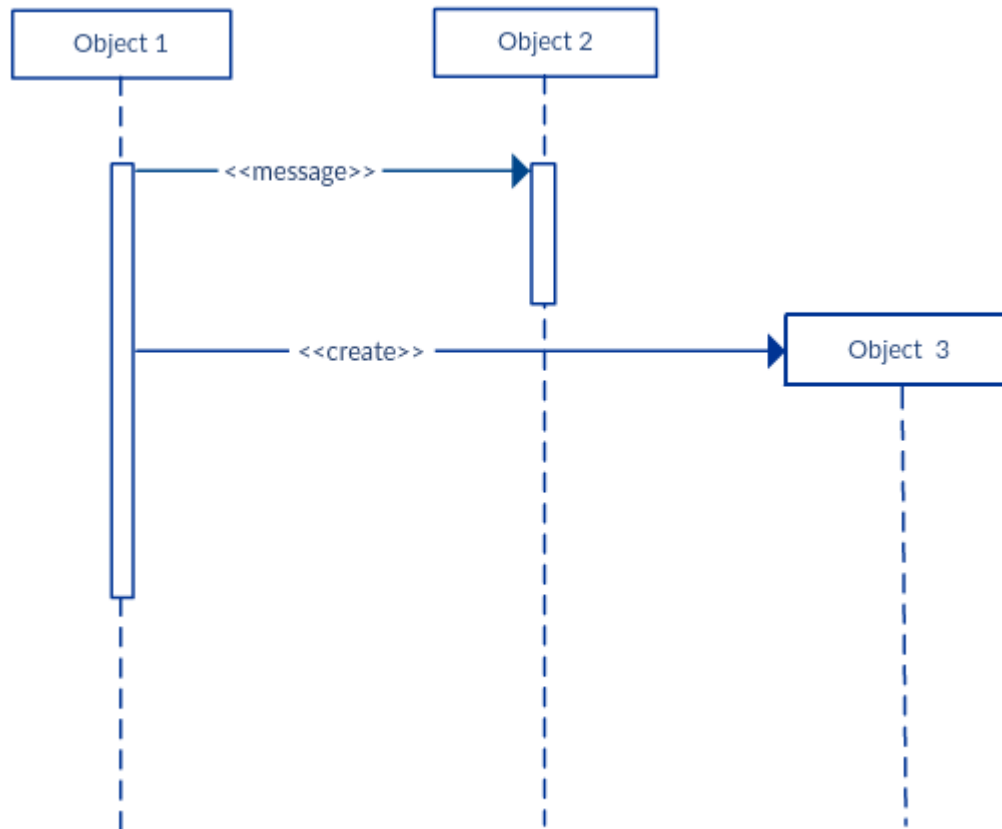
Tip: You can avoid cluttering up your diagrams by minimizing the use of return messages since the return value can be specified in the initial message arrow itself.



- *Participant creation message*

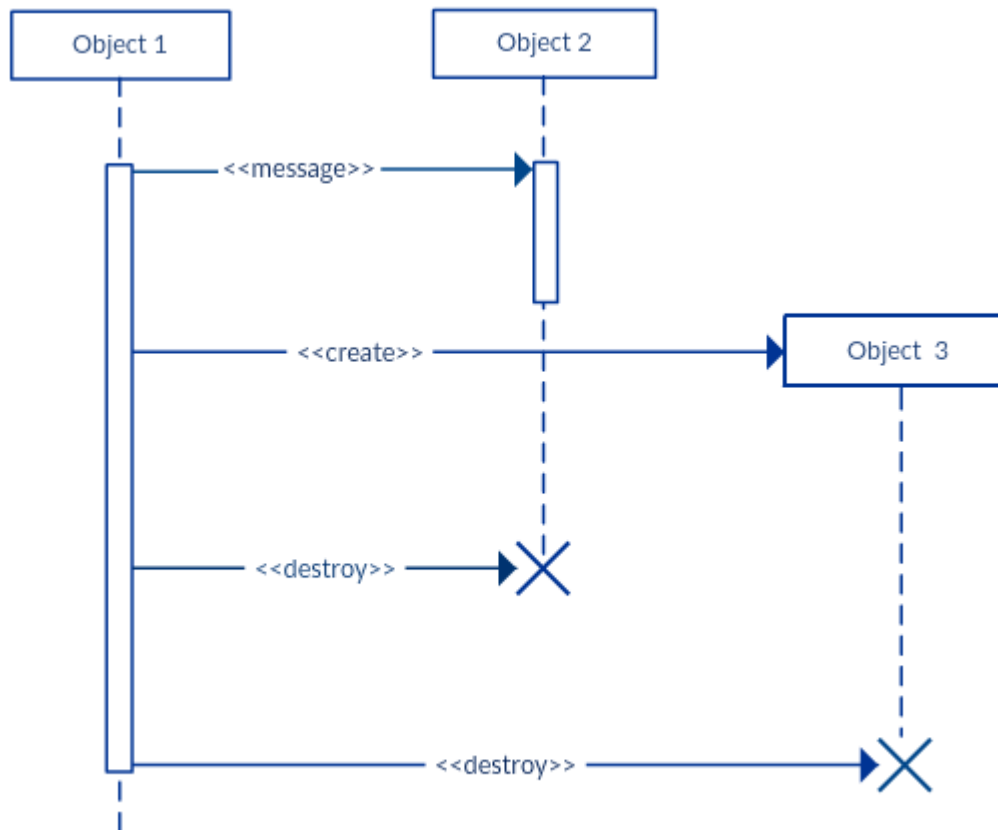
Objects do not necessarily live for the entire duration of the sequence of events. Objects or participants can be created according to the message that is being sent.

The dropped participant box notation can be used when you need to show that the particular participant did not exist until the create call was sent. If the created participant does something immediately after its creation, you should add an activation box right below the participant box.



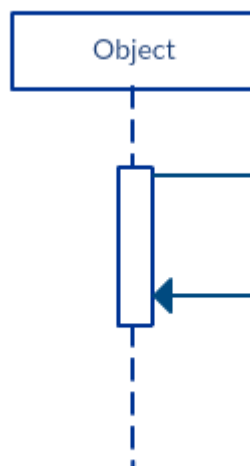
- *Participant destruction message*

Likewise, participants when no longer needed can also be deleted from a sequence diagram. This is done by adding an 'X' at the end of the lifeline of the said participant.



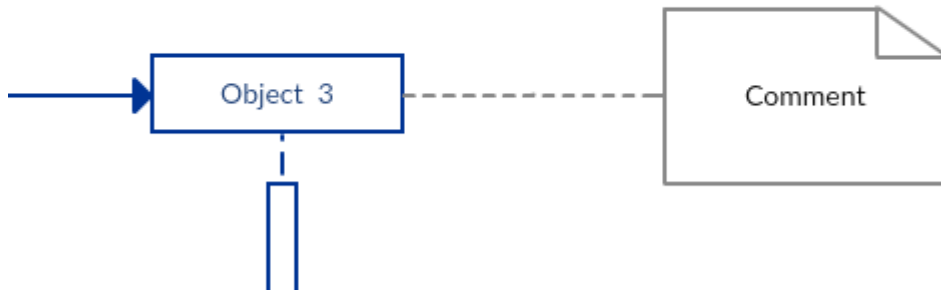
- *Reflexive message*

When an object sends a message to itself, it is called a reflexive message. It is indicated with a message arrow that starts and ends at the same lifeline as shown in the example below.



Comment

UML diagrams (<https://creately.com/lp/uml-diagram-tool/>) generally permit the annotation of comments in all UML diagram types (<https://creately.com/blog/diagrams/uml-diagram-types-examples/>). The comment object is a rectangle with a folded-over corner as shown below. The comment can be linked to the related object with a dashed line.



Note: View Sequence Diagram Best Practices to learn about sequence fragments.

Sequence Diagram Best Practices

- **Manage complex interactions with sequence fragments**

A sequence fragment is represented as a box that frames a section of interactions between objects (as shown in the examples below) in a sequence diagram.

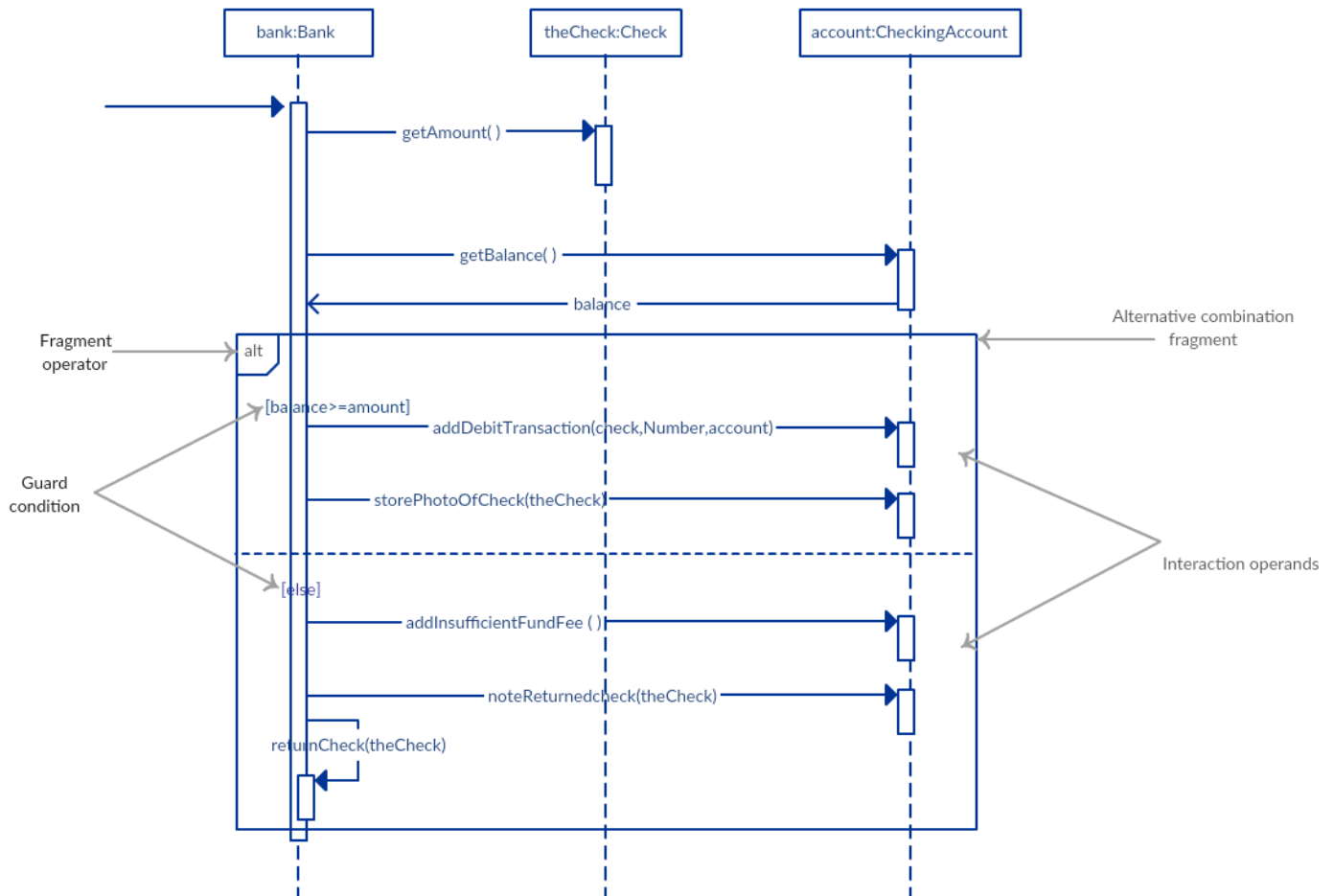
It is used to show complex interactions such as alternative flows and loops in a more structured way. On the top left corner of the fragment sits an operator. This – the fragment operator – specifies what sort of a fragment it is.

Alternatives

The alternative combination fragment is used when a choice needs to be made between two or more message sequences. It models the “if then else” logic.

The alternative fragment is represented by a large rectangle or a frame; it is specified by mentioning ‘alt’ inside the frame’s name box (a.k.a. fragment operator).

To show two or more alternatives, the larger rectangle is then divided into what is called interaction operands using a dashed line, as shown in the sequence diagram example above. Each operand has a guard to test against and it is placed at the top left corner of the operand.



Options

The option combination fragment is used to indicate a sequence that will only occur under a certain condition, otherwise, the sequence won't occur. It models the "if then" statement.

Similar to the alternative fragment, the option fragment is also represented with a rectangular frame where 'opt' is placed inside the name box.

Unlike the alternative fragment, an option fragment is not divided into two or more operands. Option's guard is placed at the top left corner.

(Find an example sequence diagram with an option fragment in the Sequence Diagram Templates and Examples section).

Loops

Loop fragment is used to represent a repetitive sequence. Place the words 'loop' in the name box and the guard condition near the top left corner of the frame.

In addition to the Boolean test, the guard in a loop fragment can have two other special conditions tested against. These are minimum iterations (written as *minint* = [the number]) and maximum iterations (written as *maxint* = [the number]).

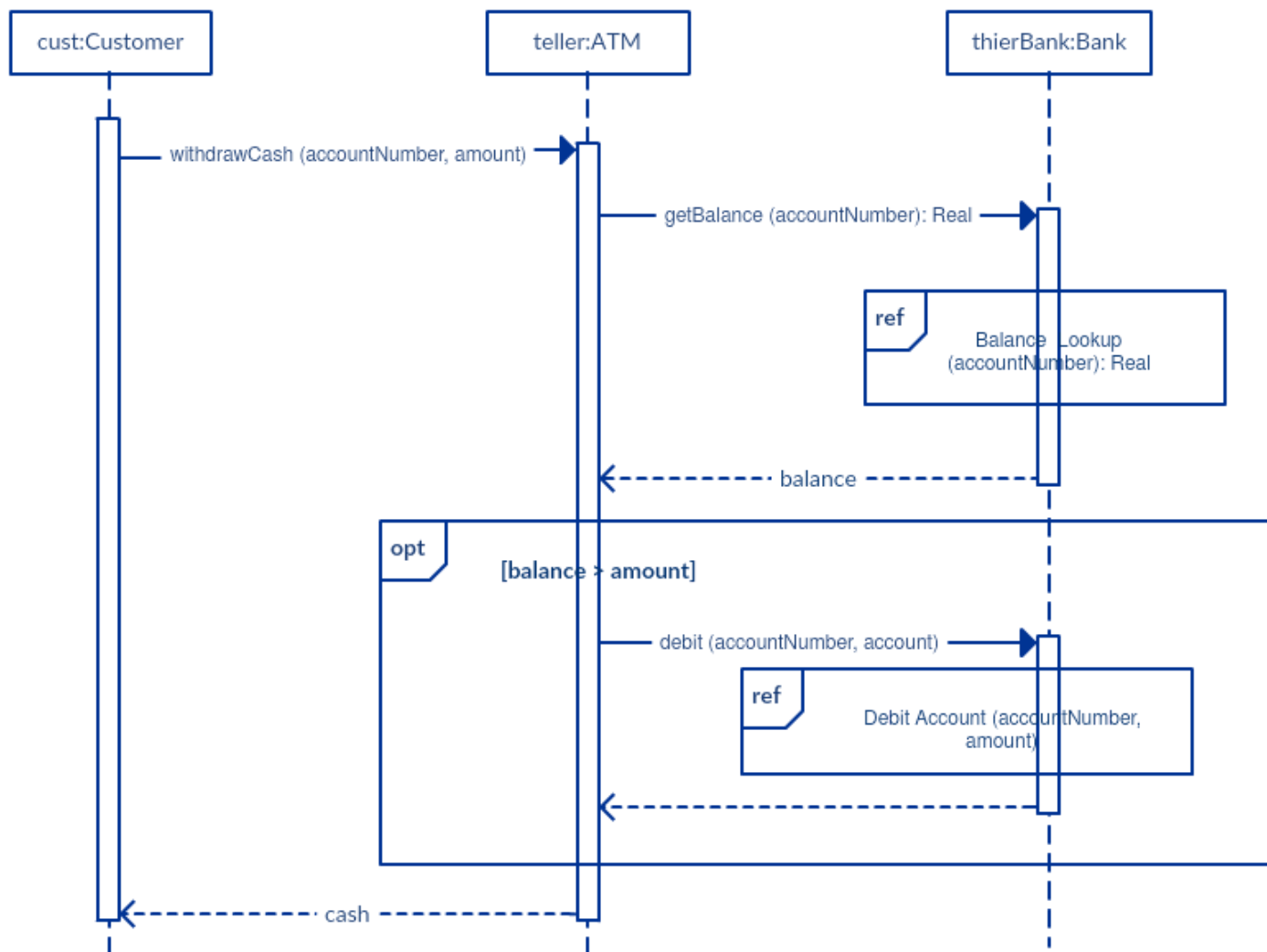
If it is a minimum iterations guard, the loop must execute not less than the number mentioned, and if it is a maximum iterations guard, the loop mustn't execute more than the number indicated.

(Find an example of a loop fragment below in the sequence diagram templates and example section)

Reference Fragment

You can use the ref fragment to manage the size of large sequence diagrams. It allows you to reuse part of one sequence diagram in another, or in other words, you can reference part of a diagram in another diagram using the ref fragment.

To specify the reference fragment, you have to mention 'ref' in the name box of the frame and the name of the sequence diagram that is being referred to inside the frame.



For more sequence fragments refer to *Beyond the Basics of Sequence Diagrams: Part 1* (<https://creately.com/blog/diagrams/beyond-the-basics-of-sequence-diagrams-part-1/>), *Part 2* (<https://creately.com/blog/diagrams/beyond-the-basics-of-sequence-diagrams-part-2/>) and *Part 3* (<https://creately.com/blog/diagrams/beyond-the-basics-of-sequence-diagrams-part-3/>).

- **Draw smaller sequence diagrams that capture the essence of the use case**

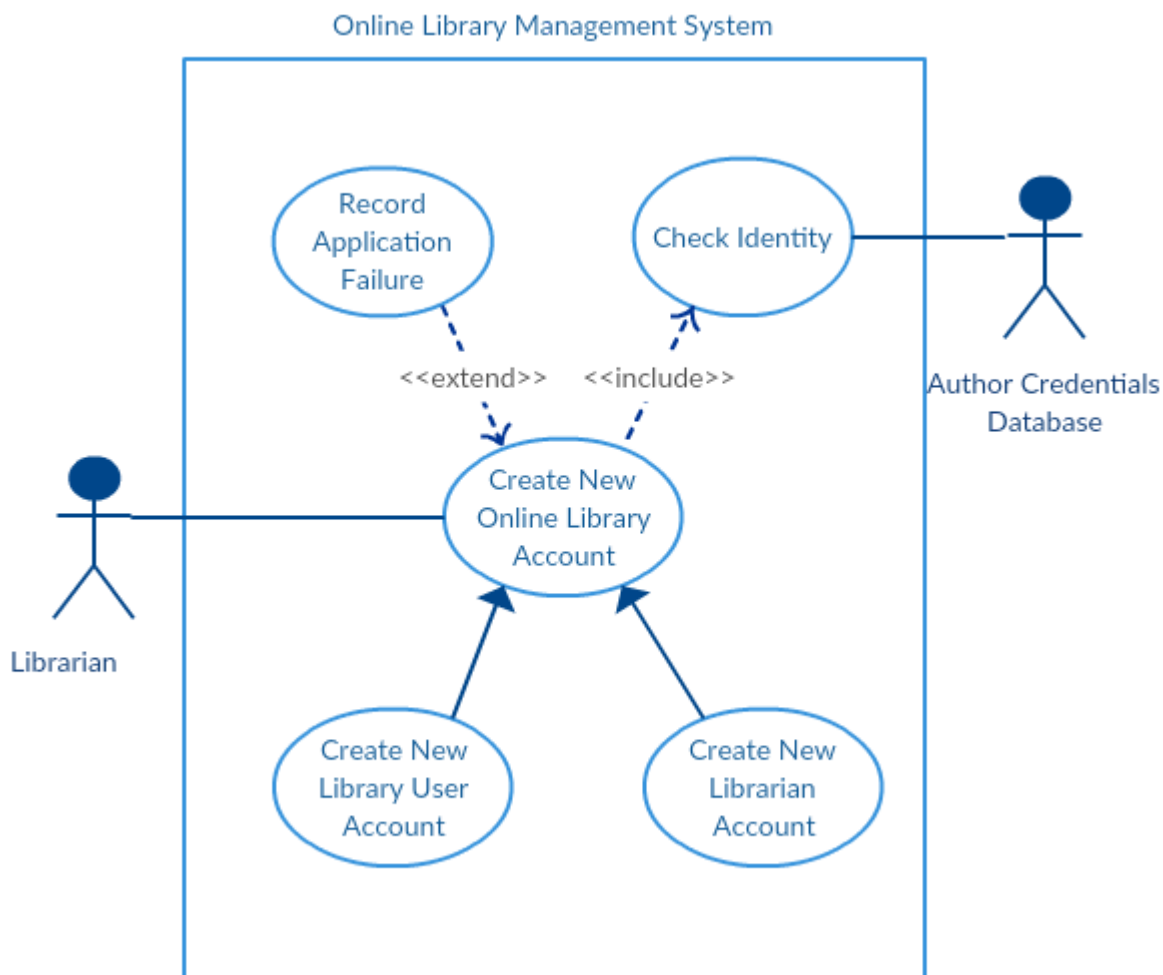
Instead of cluttering your sequence diagram with several objects and groups of messages that will confuse the reader, draw a few smaller sequence diagrams that aptly explain what your system does. Make sure that the diagram fits on a single page and leaves space for explanatory notes too.

Also instead of drawing dozens of sequence diagrams, find out what is common among the scenarios and focus on that. And if the code is expressive and can stand on its own, there's no need to draw a sequence diagram in the first place.

How to Draw a Sequence Diagram

A sequence diagram represents the scenario or flow of events in one single use case. The message flow of the sequence diagram is based on the narrative of the particular use case.

Then, before you start drawing the sequence diagram or decide what interactions should be included in it, you need to draw the use case diagram (<https://creately.com/diagram-type/use-case>) and ready a comprehensive description of what the particular use case does.



From the above use case diagram example of 'Create New Online Library Account', we will focus on the use case named 'Create New User Account' to draw our sequence diagram example.

Before drawing the sequence diagram, it's necessary to identify the objects or actors that would be involved in creating a new user account. These would be;

- Librarian
- Online Library Management system
- User credentials database
- Email system

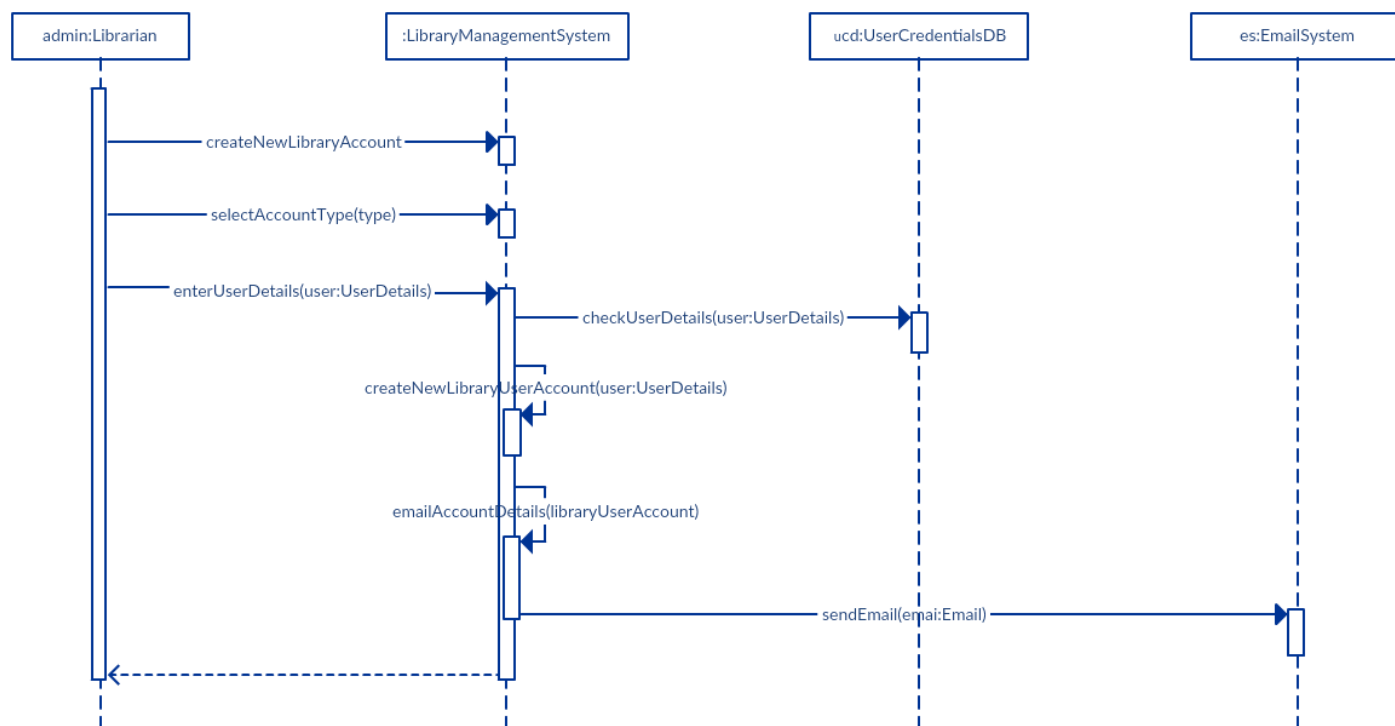
Once you identify the objects, it is then important to write a detailed description on what the use case does. From this description, you can easily figure out the interactions (that should go in the sequence diagram) that would occur between the objects above, once the use case is executed.

Here are the steps that occur in the use case named 'Create New Library User Account'.

- The librarian request the system to create a new online library account
- The librarian then selects the library user account type
- The librarian enters the user's details
- The user's details are checked using the user Credentials Database
- The new library user account is created
- A summary of the of the new account's details are then emailed to the user

From each of these steps, you can easily specify what messages should be exchanged between the objects in the sequence diagram. Once it's clear, you can go ahead and start drawing the sequence diagram.

The sequence diagram below shows how the objects in the online library management system interact with each other to perform the function 'Create New Library User Account'.



Sequence Diagram Common Mistakes

When drawing sequence diagrams, designers tend to make these common mistakes. By avoiding these mistakes you can ensure the quality of your diagram.

- Adding too much detail. This clutters up the diagram and makes it difficult to read.
- Obsolete and out of date sequence diagrams that are irrelevant when compared to the interfaces, actual architectures etc. of the system. Don't forget to replace them or modify them.
- Leaving no blank space between the use case text and the message arrow; this makes it difficult for anyone to read the diagram.
- Not considering the origins of message arrows carefully.

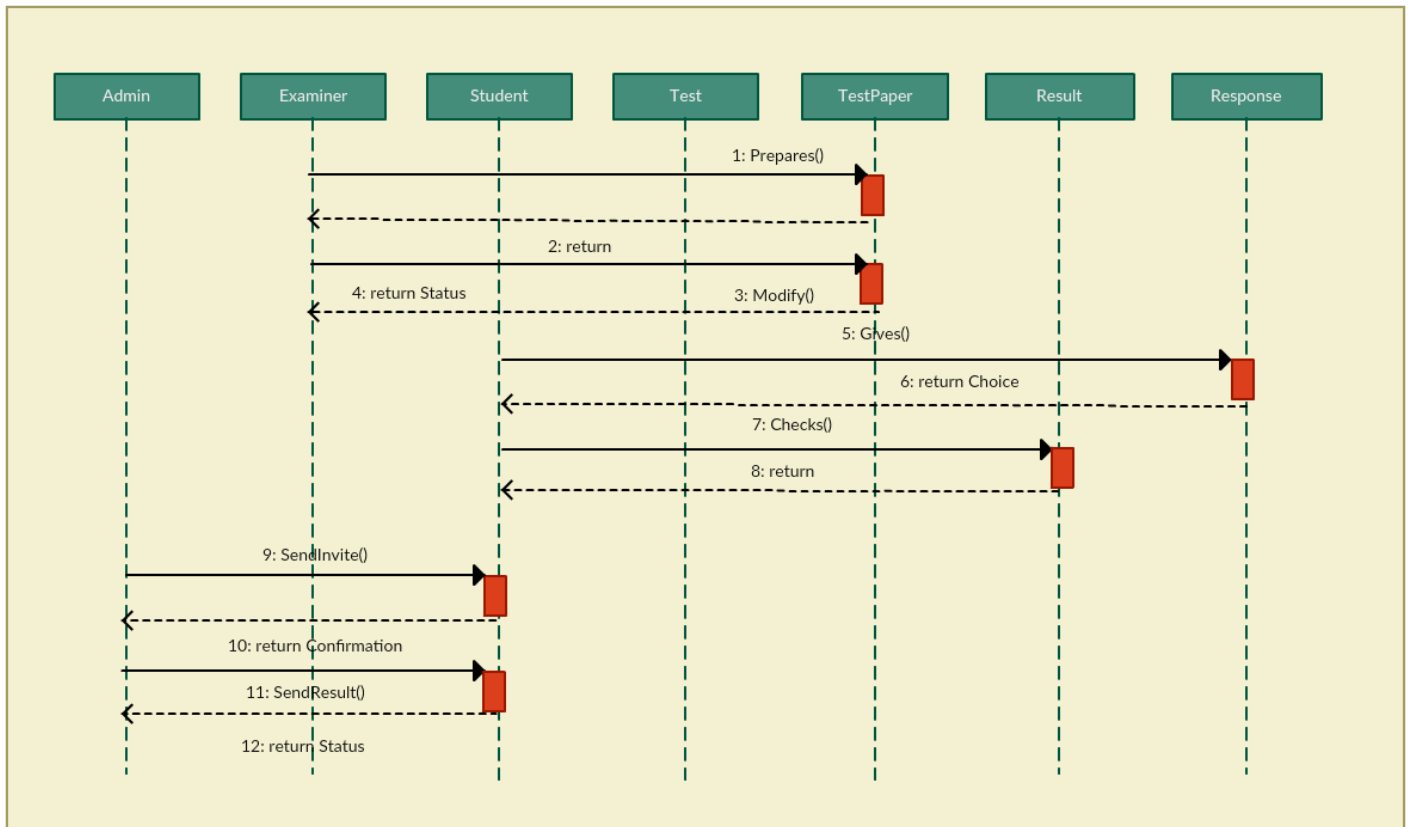
See these common mistakes explained in detail in [Sequence Diagram Guide: Common Mistakes to Avoid When Drawing Sequence Diagrams](https://creately.com/blog/diagrams/10-common-mistakes-to-avoid-in-sequence-diagrams/)

(<https://creately.com/blog/diagrams/10-common-mistakes-to-avoid-in-sequence-diagrams/>).

Sequence Diagram Examples and Templates

Following are a few sequence diagram examples (<https://creately.com/diagram-community/examples/t/sequence-diagram>) and templates that are drawn using Creately. Create sequence diagrams online (<https://creately.com/diagram-type/uml-sequence-diagrams>) using Creately's online tool. Click on the template to open it in the editor.

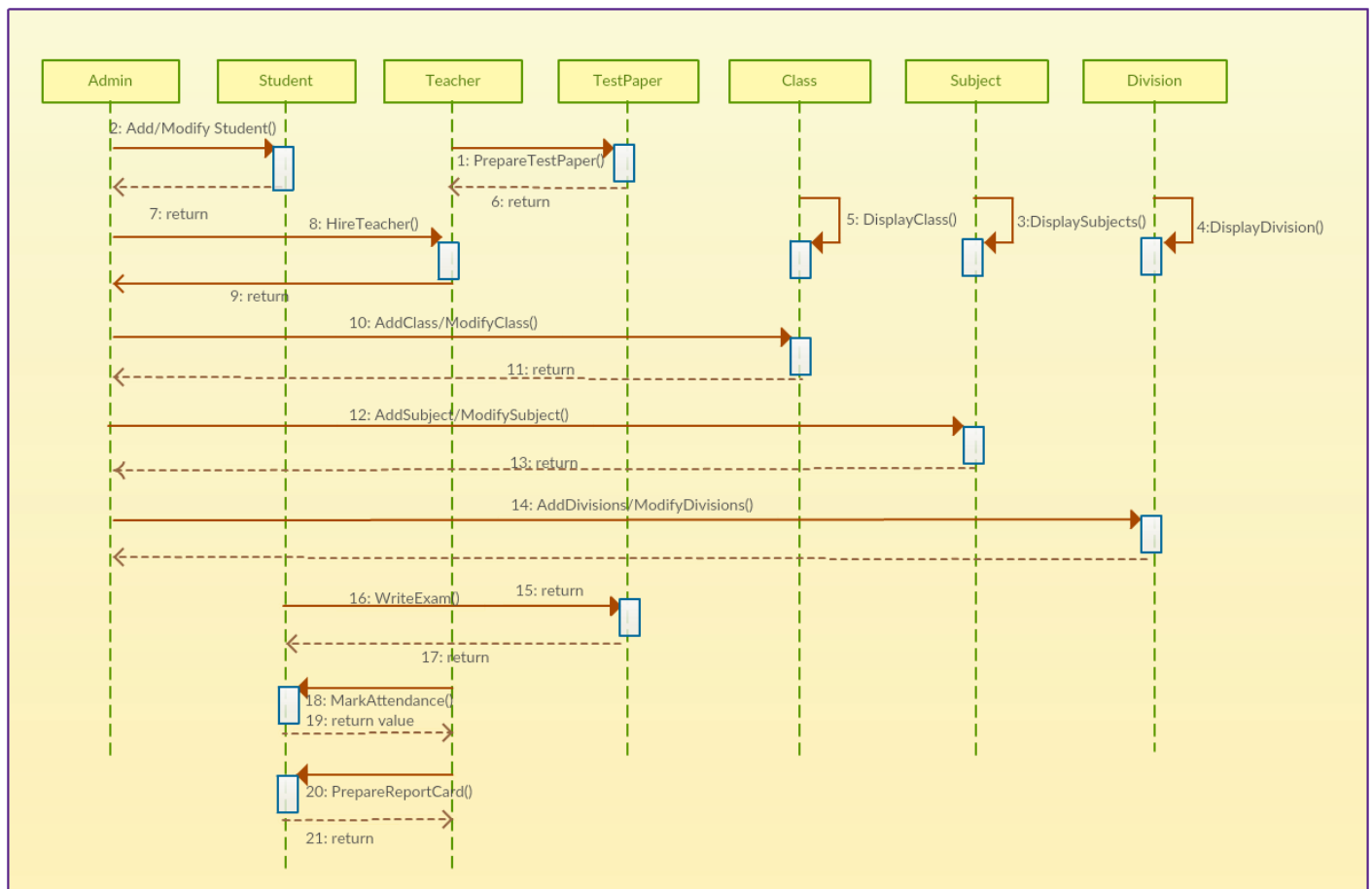
Sequence Diagram of an Online Exam System



(<https://creately.com/creately-start?templID=imd2m8tw1>)

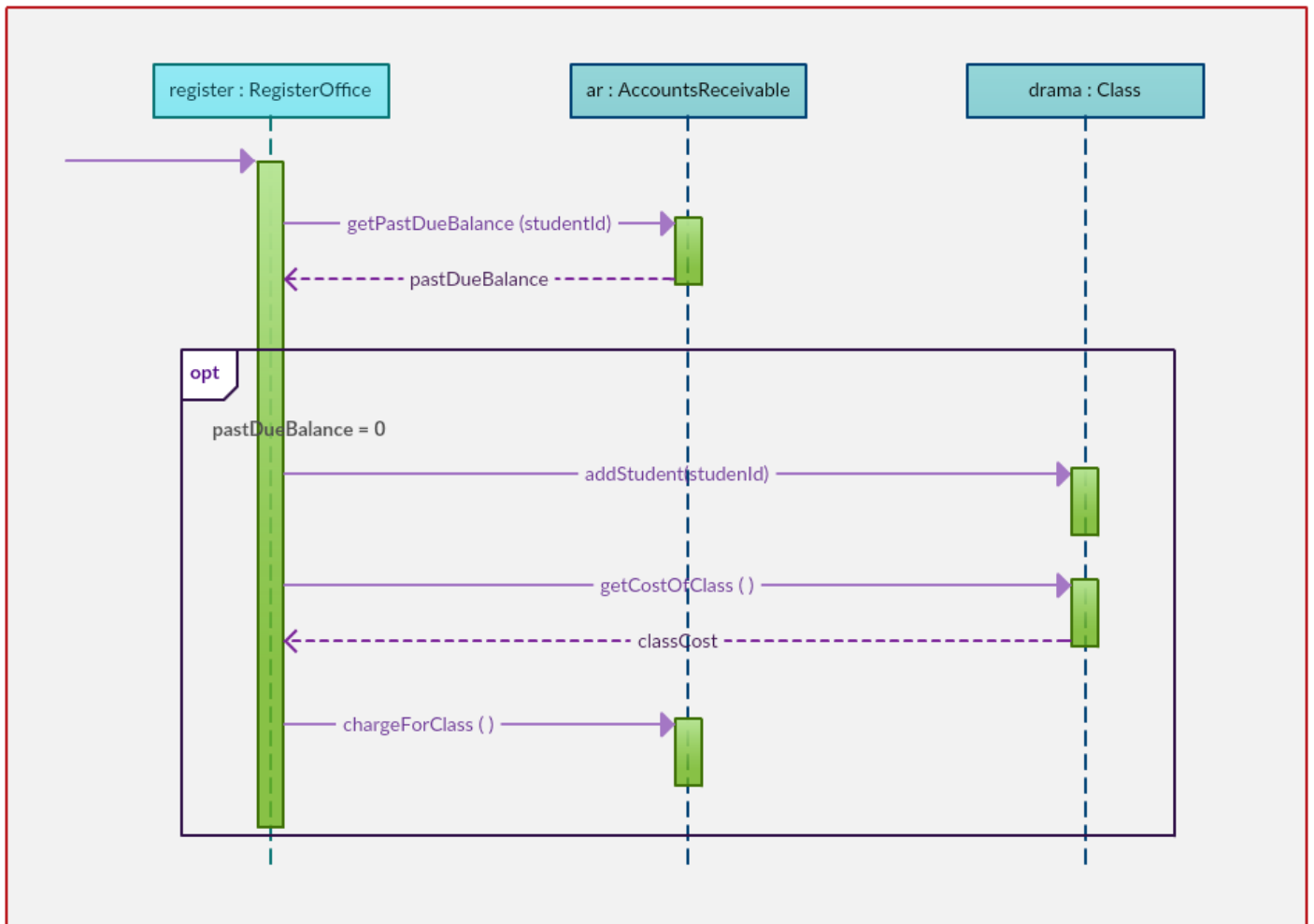
Click the image to edit it online

Sequence Diagram Example of a School Management System



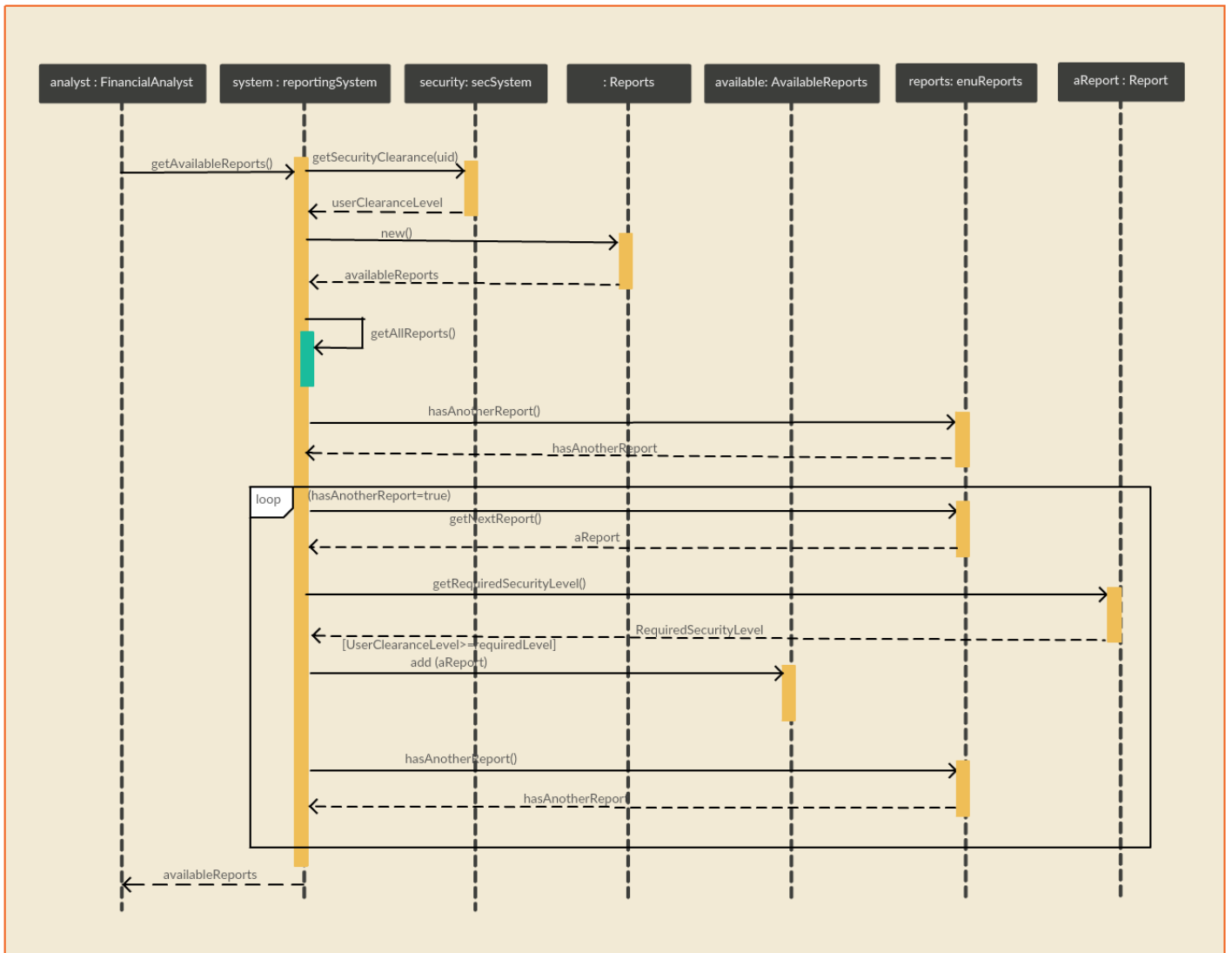
(<https://creately.com/creately-start?tempID=imcvynir1>)

Example of an Option Combination Fragment



(<https://creately.com/creately-start?templID=iy020p1n1>)

Example of a Loop Sequence



(<https://creately.com/creately-start?templID=io1a1c3h1>)

Here are some more sequence diagram templates and examples

(<https://creately.com/diagram-community/popular/t/sequence-diagram>) that you can edit right away.

Sequence Diagram Tutorial – SlideShare Presentation

- **Use Case Diagram Tutorial (Guide with Examples)**
(<https://creately.com/blog/diagrams/use-case-diagram-tutorial/>)

sequence diagram examples (<https://creately.com/blog/tag/sequence-diagram-examples/>)

sequence diagram guide (<https://creately.com/blog/tag/sequence-diagram-guide/>)

sequence diagram tutorial (<https://creately.com/blog/tag/sequence-diagram-tutorial/>)

sequence diagrams (<https://creately.com/blog/tag/sequence-diagrams/>)

< (<https://creately.com/blog/creately/a-new-beginning/>)

>

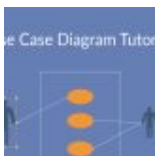
(<https://creately.com/blog/announcements/latest-aws-icons-and-libraries/>)

More related articles



Sequence Diagram Templates to Instantly View Object Interactions

(<https://creately.com/blog/examples/sequence-diagram-templates/>)



Use Case Diagram Tutorial (Guide with Examples)

(<https://creately.com/blog/diagrams/use-case-diagram-tutorial/>)



The Easy Guide to UML Class Diagrams | Class Diagram Tutorial

(<https://creately.com/blog/diagrams/class-diagram-tutorial/>)

Author

Amanda Athuraliya



THE ULTIMATE SEQUENCE DIAGRAM TUTORIAL

<http://creately.com> | [@creately](#)

1 of 37

Sequence diagram guide Power Point (PPT) (<http://www.slideshare.net/creately/the-ultimate-sequence-diagram-guide>) from Creately (<http://www.slideshare.net/creately>)

Feedback on the Sequence Diagram Tutorial

This sequence diagram tutorial covers everything you need to know on sequence diagrams and drawing them. If you have any suggestions or questions regarding the sequence diagram tutorial, feel free to leave a comment.

More Diagram Tutorials

- Business Process Modeling Tutorial (BPM Guide Explaining Features) (<https://creately.com/blog/diagrams/business-process-modeling-tutorial/>)
- Ultimate Flowchart Guide (Complete Flowchart Tutorial with Examples) (<https://creately.com/blog/diagrams/flowchart-guide-flowchart-tutorial/>)

Amanda Athuraliya is the communication specialist/content writer at Creately, online diagramming and collaboration tool. She is an avid reader, a budding writer and a passionate researcher who loves to write about all kinds of topics.

View all posts by Amanda Athuraliya → (<https://creately.com/blog/author/amanda/>)

Comments

Fazal Abbas

The easiest and one of the best tutorial for learning about UML Sequence diagram

30.09.2019 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-67682>) - reply

Kathy

An example of how to read a diagram would be nice.

18.03.2019 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-64389>) - reply

kerreme

how to find those objects is not clear, always same missing thing in tutorials, you should explain entity, boundry and conrol objects those needed to be in seq diagram, one should not be able to put login as a object or other actions on object place, u dont know even how to draw it, shame on u

26.11.2017 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-53072>) - reply

Nishadha (<http://rumblinglankan.com>)

Hi Kerreme,

Thanks for your comment and I will try to answer your questions to the best of my ability.

How to find the objects involved in the sequence diagrams is answered in the "How to draw a sequence diagram" section. I will talk to the editor to see if we can make it clearer or maybe have its own subsection.

Agree with the point that you need to explain entity, boundary and control objects. Will update the article with an explanation.

Not sure about the login as an object one. Can you explain it a bit more?

13.12.2017 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-53217>) - reply

Sanjay

Very well explained. It is very good for anyone going to work on sequence diagrams.

14.11.2017 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-53006>) - reply

Tuges

thank you for putting the effort into this tutorial. You are helping so many such as me. God bless you and your work..

16.08.2017 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-52287>) - reply

Mehwish Mehak

This is a complete and easy to understand article, thanks for such an awesome information and cleaver delivery. <3 <3

25.06.2017 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-51889>) - reply

Akila Gardiye

This is a very clear tutorial with simple examples . I started viewing this blog not having a clue about sequence diagrams and now i'm leaving with the knowledge of what it actually is and on how to draw one. Thank you for such a good tutorial !!

13.04.2017 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-51222>) - reply

kasun

Admire the work..a great job

13.04.2017 (<https://creately.com/blog/diagrams/sequence-diagram-tutorial/#comment-51214>) - reply

Mier

Clear tutorial. Thanks!

Leave a comment

Your message *

Your name *

Your email *

Please enter an answer in digits:

17 - 14 =

Submit

Home (/)

Plans (/plans)

Features (/tour)

Examples (/diagram-examples)

About (/about-us)

© Cinergix Pvt. Ltd 2020 | All rights reserved.