If code does not have any effect on the interaction, don’t show it.  
  
But if it does, show it. Eg Validation

**Every Use case**, needs an ISD

Lifeline: the local variables in a method, that dies at the end of that method.

Classes in your **ISD, must be** in your class diagram.

**Focus of control:** This shows which objects are currently in use (controlling the interaction)

**Messages:** Methods or functions (only those that effect ISD)  
Actor to Object  
Object to Actor:   
Object to Object  
Actor to Actor

**Message  
Synchronous**: Step by step. Step 2 waits for step 1 to be done. (Solid arrow)  
**Asynchronous**: Object does not until resulting action is done. (line arrow)  
**Creation:** (line arrow dotted line)  
**Reply:** (Solid arrow dotted line)

**The reply in the ISD diagram on blauw, where the class1 replies to the actor, is wrong. The device can not reply to the user directly?**

**As** the computer does not send you a message, but displays it on the computer (which is not you, so it can not physically tell/send u the message)

**Loop:** these are loops  
-condition

**Alt:** if statements   
-condition  
-else

**References:** When ISD become too big  
sd interaction 1… sd interaction 2  
-Then you have a Ref interaction 1, in your sd interaction 2.  
  
in interaction one, the arrows going in and out, and the classes used, is shown in interaction.