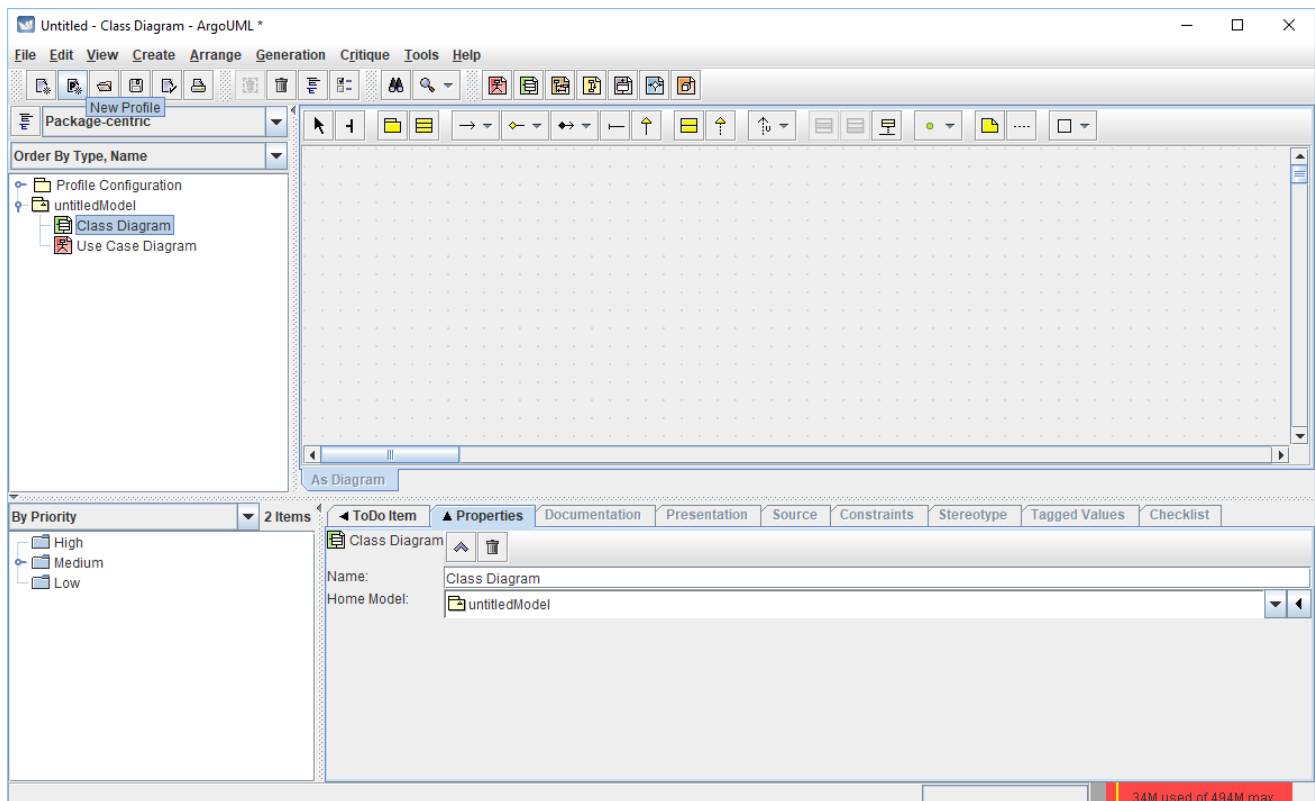


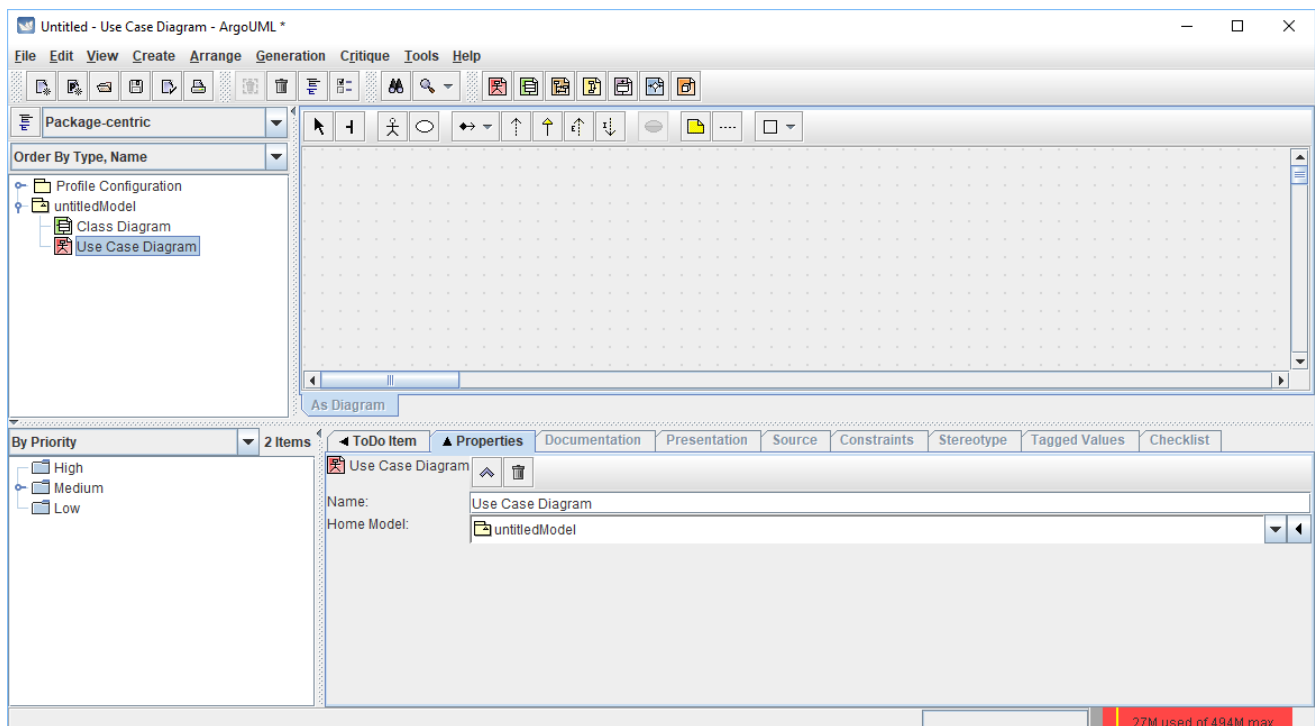
## Use cases in ArgoUML

Run ArgoUML from the lab PC or your laptop. The default drawing window is for class diagrams when ArgoUML is started as you can see from.



## Drawing a Use-case Diagram

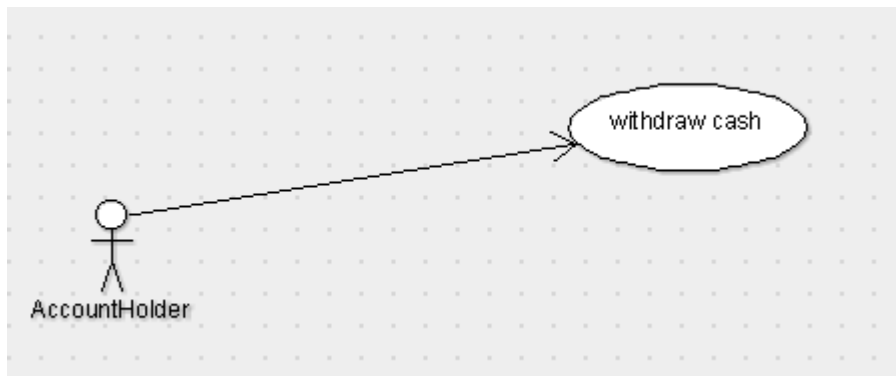
To change this, expand **untitledModel** folder in the ArgoUML browser window, then click on the Use case icon to get (notice how the drawing area has changed):



Note that the toolbar will change. Use the toolbar to begin creating a use-case diagram like:

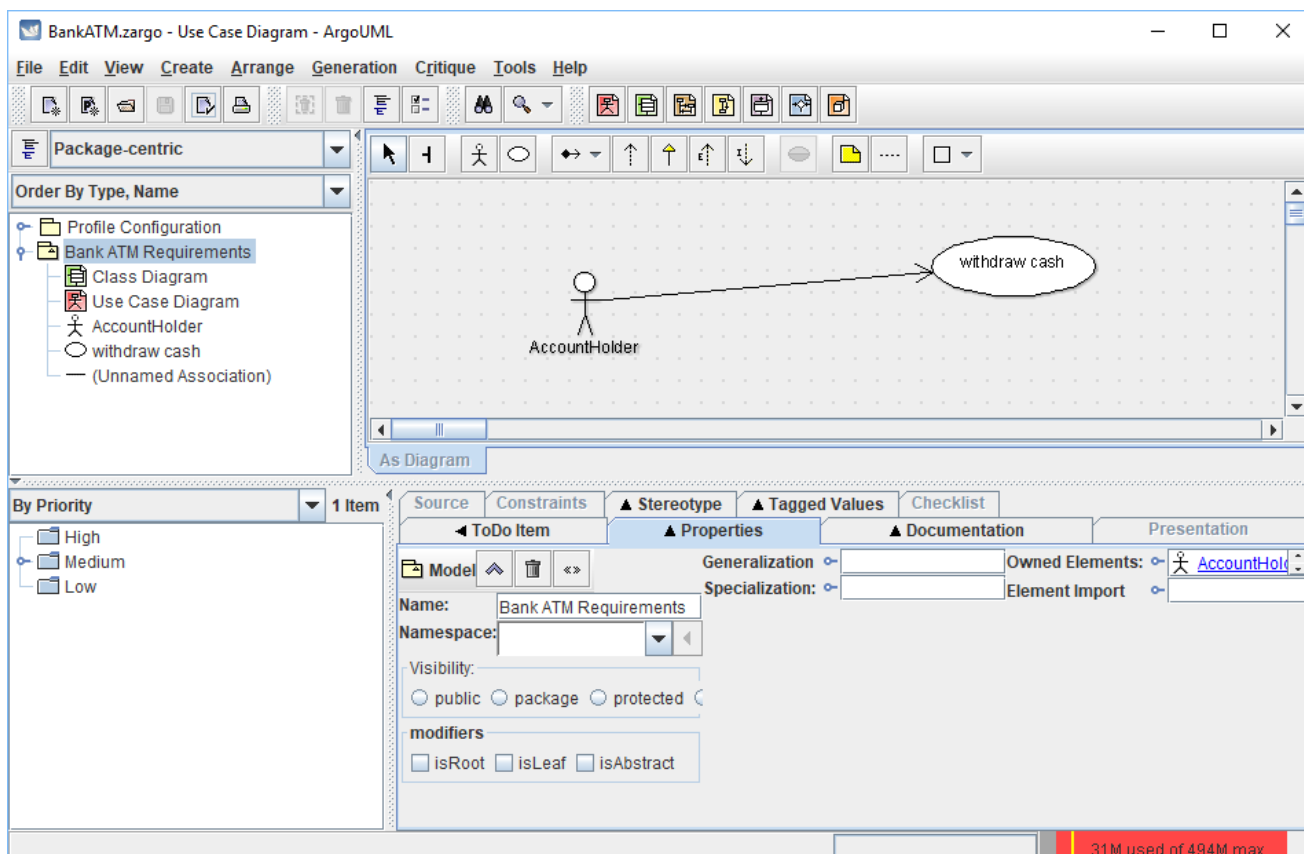


Then link the actor with the use-cases using the association arrow to get (make sure the connecting arrow is as shown):



Next click on the folder **untitledModel** in the browser window, its properties should appear in the bottom right hand window, use the properties window to rename to **Bank ATM Requirements**.

Next using menu **File | Save Project As**, save your ArgoUML model to a file named **BankATM.zargo**



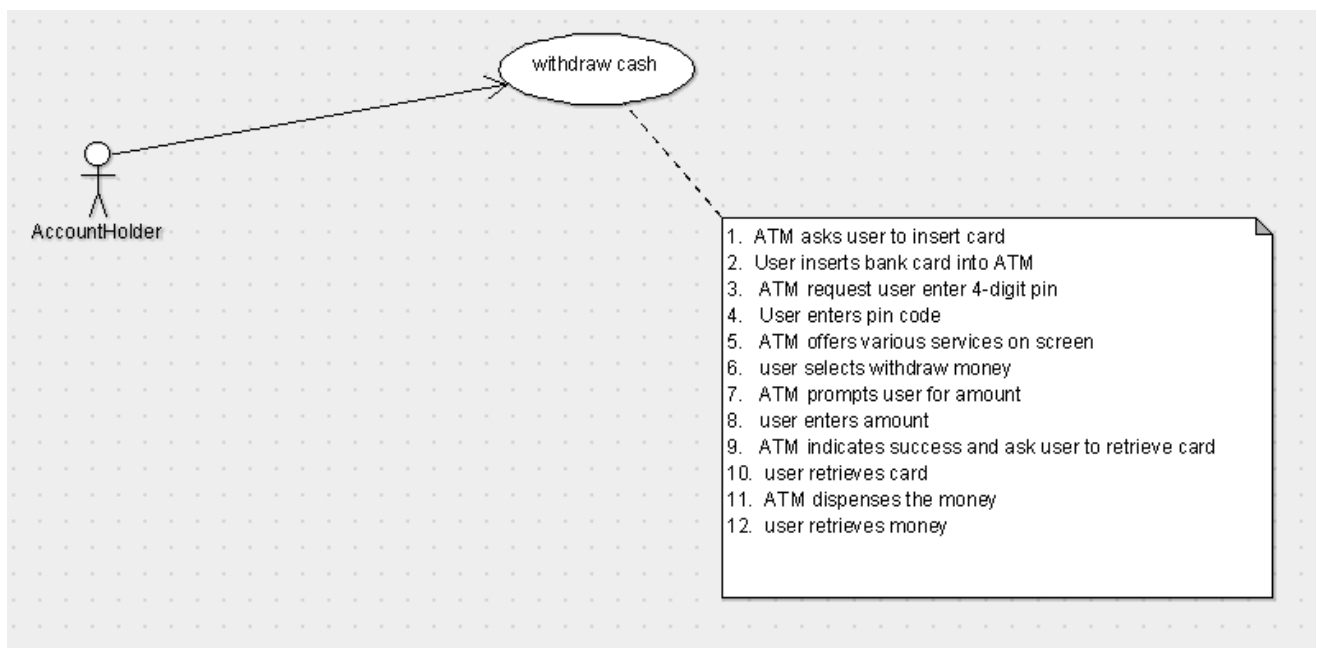
## Writing Use case scenarios

Of course, a use case diagram on its own is not enough to describe a software requirement. The actual interactions between user (actor) and software system have to be listed. This is normally done using a specialised CASE tool for requirements engineering.

But we will keep it simple and create UML comments containing a simple use case description in ArgoUML and link them to the appropriate use cases in the diagram. From the use case toolbar use the icons

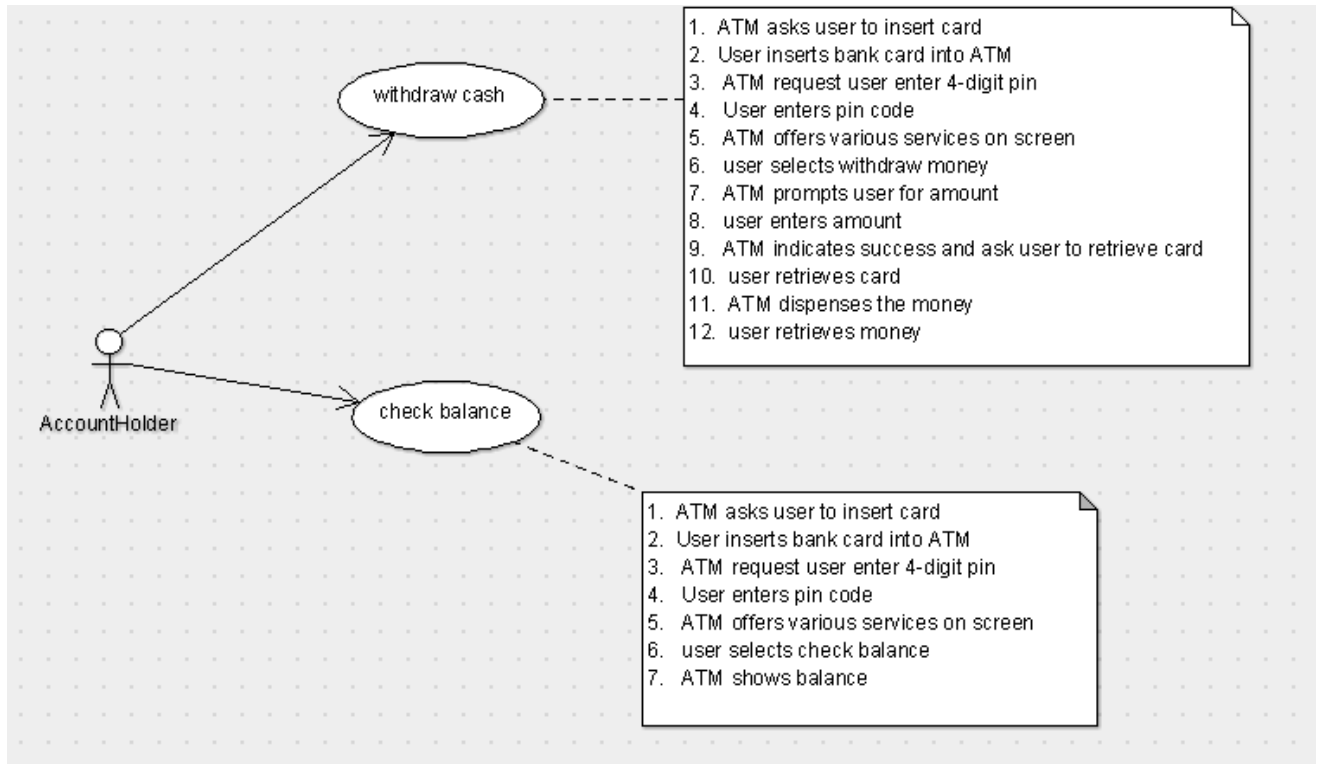


to select **New Comment** and **New Comment Link** to set up a comment as shown below and link it to the use case. You may need to rearrange use cases on the diagram for aesthetic reasons.




## Use case overlap & <<include>>

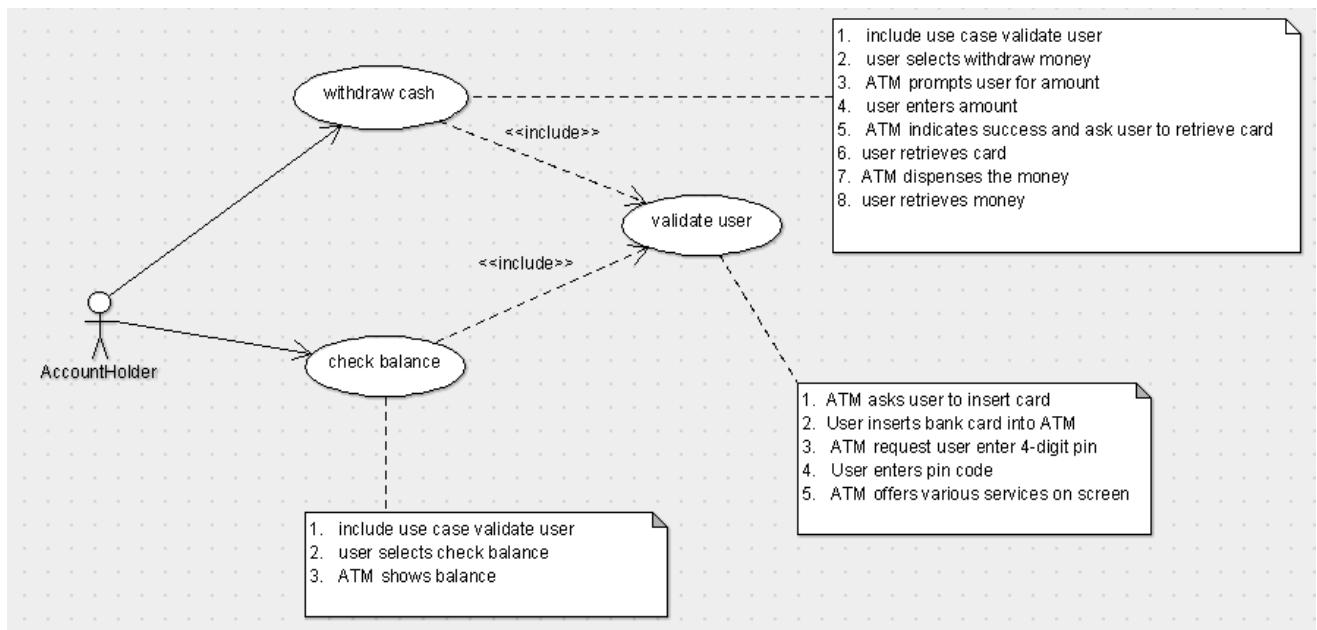
Next we add another use case: **check balance** as shown next.



Both of these use-cases have steps 1 to 5 in common, namely the account holder must first be validated by inserting his bankcard and entering a 4-digit pin code. The bank ATM will then check that the customer is legitimate. These actor-system interactions can be separated out of the two use-cases and put into a third which is in turn included in the original two. This is basically a way of reducing duplicate text in `describing requirements scenarios.

So let us separate these interactions into a separate use-case called *validate user* and include it in the other 2

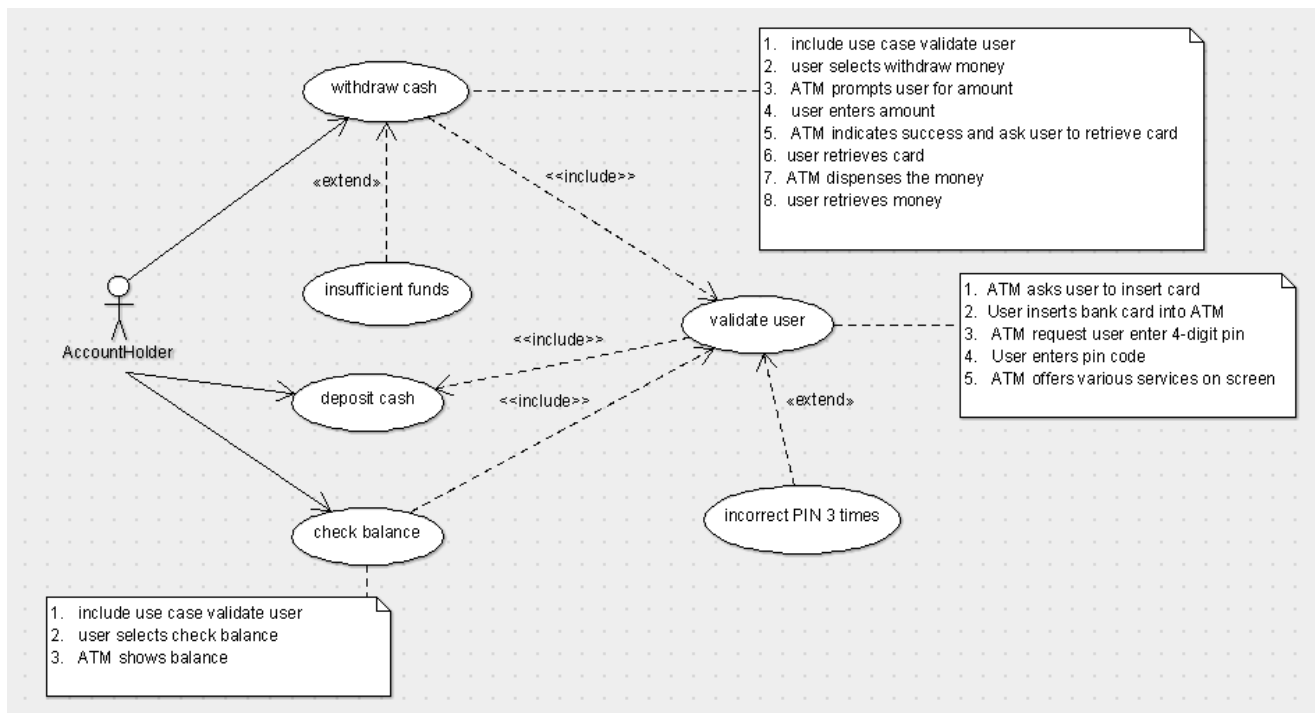
use-cases. Use the icon  for **New Include** to link them. Should result in something like:



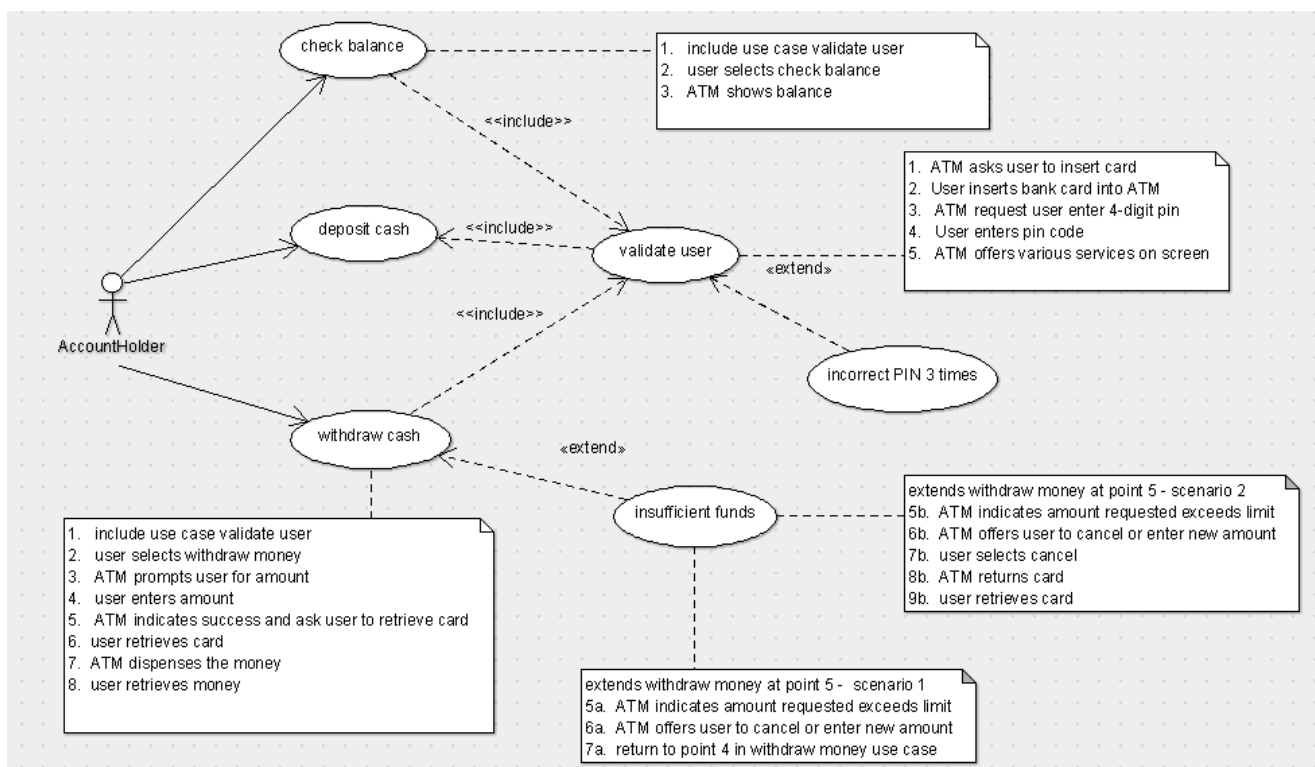
## Writing more use case scenarios for ATM Model

Sometimes it is better to place a significant alternative scenario for a complex use case into a separate one and say that the new one is an extension or specialisation of the original.

Modify your use case diagram along the following lines:



Next complete the new use cases along the lines of what is shown next.



### Exercise

Write a scenario for

1. “deposit cash” .
2. “incorrect PIN 3 times”

### CA comment

This work is will be examined as part of your CA marks.