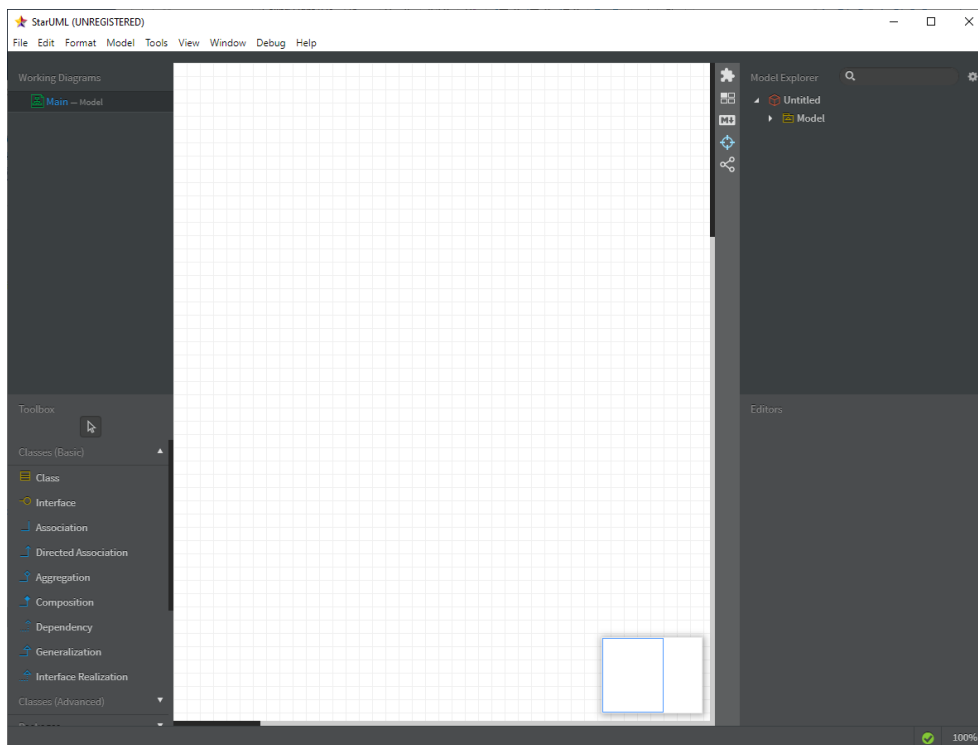


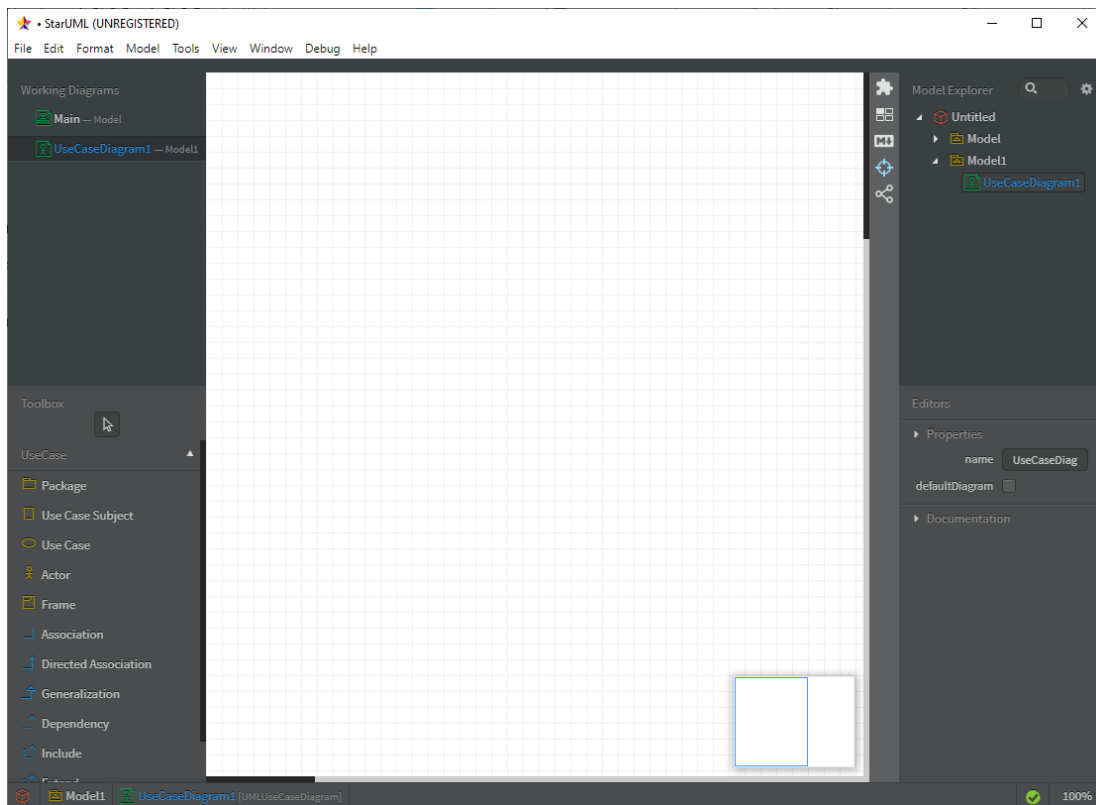
Open StarUML and run the evaluation version.

Use the menu **File/Save as** to save your model under the name BankATM.mdj.

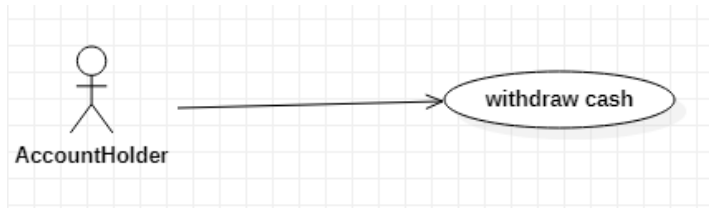
The default drawing diagram that appears is a **Class Diagram**. We want a usecase diagram.



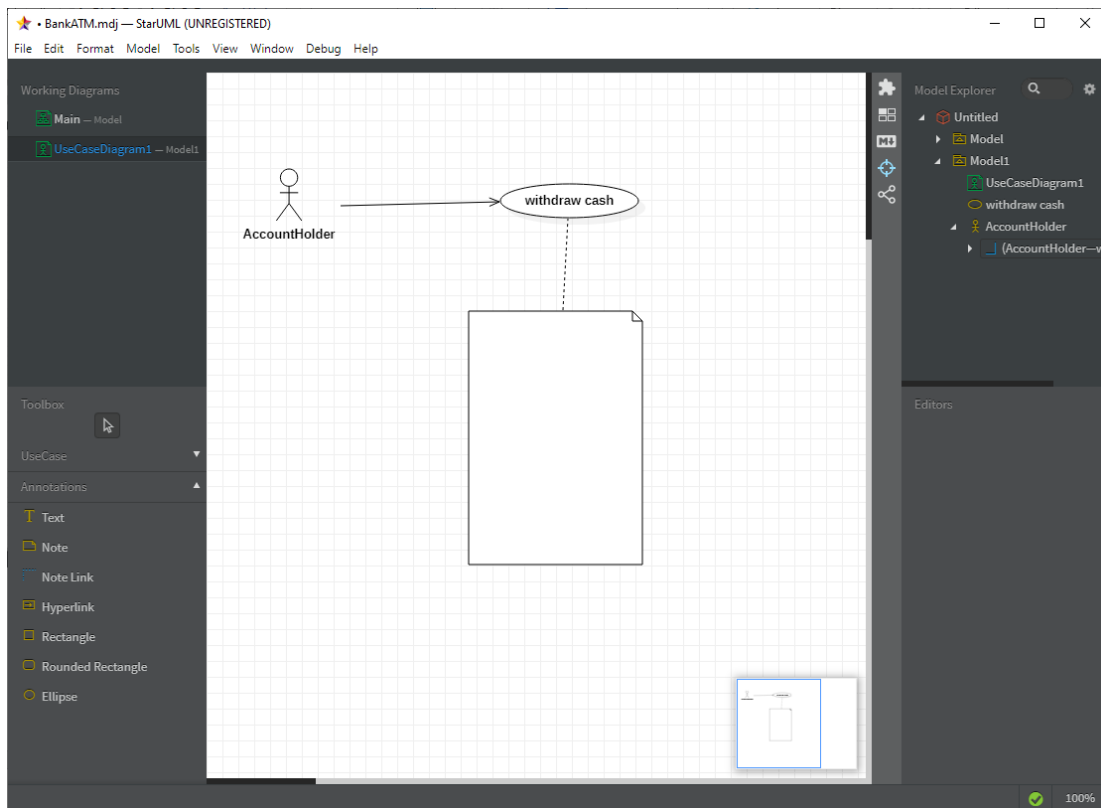
So use the menu **Model/Add Diagram /Use case Diagram** to get:



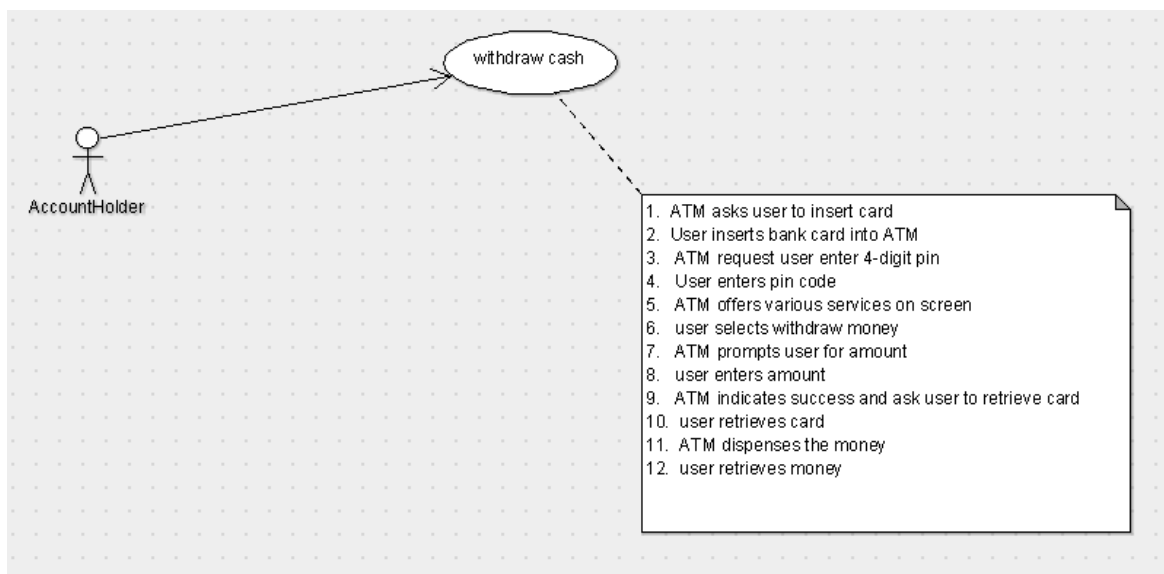
Then use the drawing tools provided to draw:



Expand the Annotations palette underneath the usecase one. Add a note and link it to the usecase diagram to get.

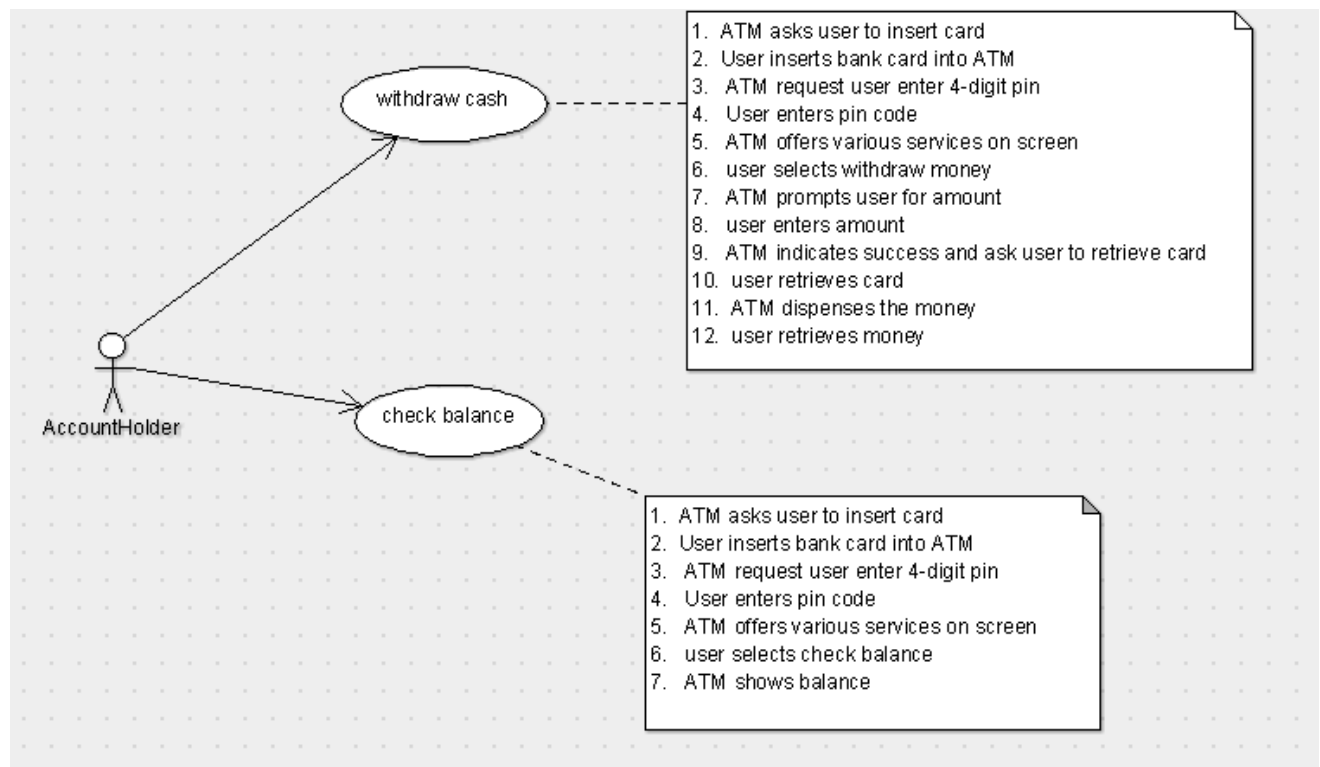


Then put a usecase scenario into the note to document the usecase “withdraw cash”. You will need to use **Shift Enter** to get from one line to the next.



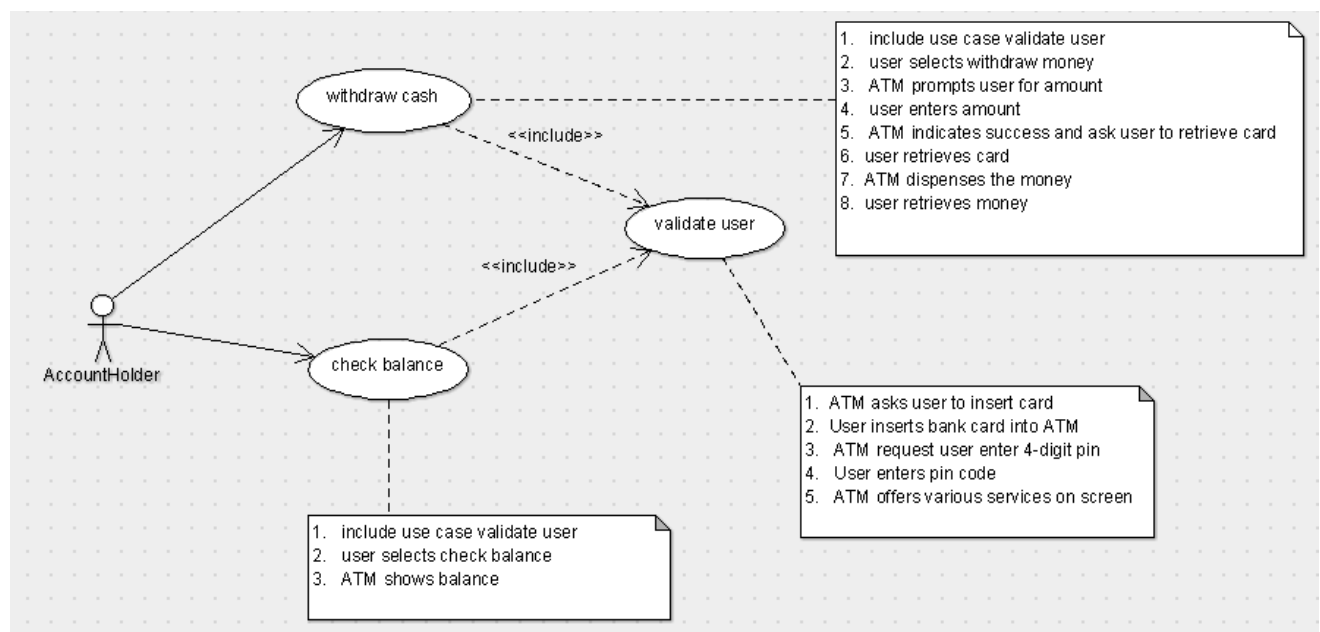
Use case overlap, restructuring usecases and <<include>>

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.

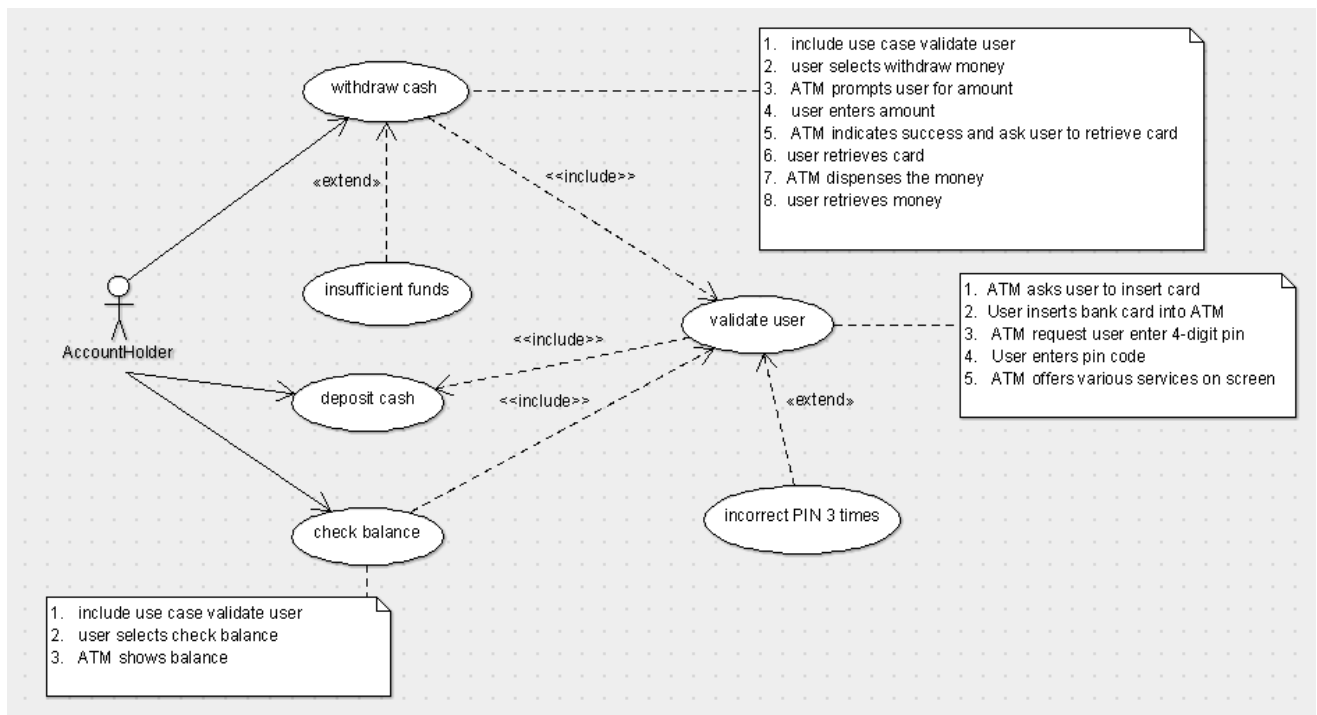
So let us separate these interactions into a separate use-case called *validate user* and include it in the other 2 usecases.



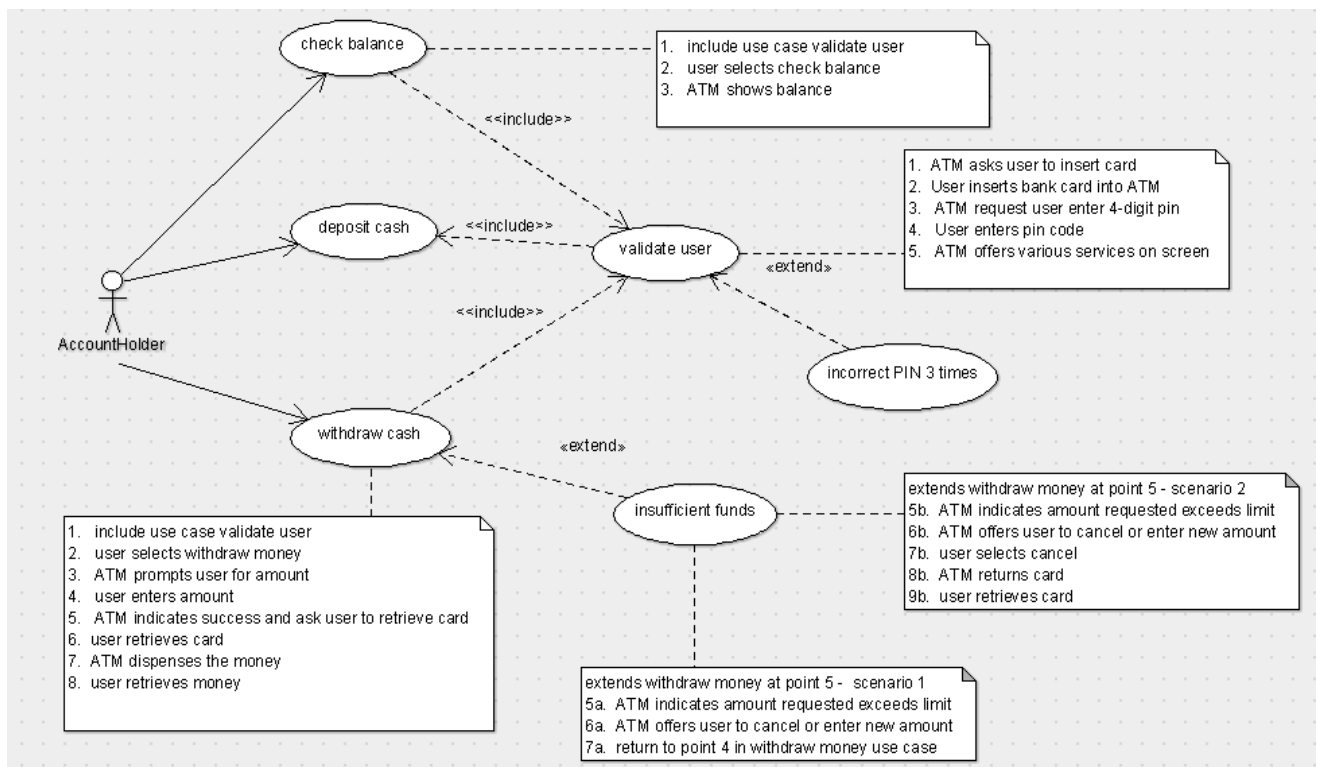
Writing more use case scenarios for ATM Model

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.

