

CS 3331 Use Case Template

USE CASE # 001	Make Deposit
<ul style="list-style-type: none">• Main objective	Allows the user to deposit money into their account using the ATM
Scope and level	Main because this achieves a specific user goal that is depositing money into an atm which goes into your account.
Preconditions	<ul style="list-style-type: none">- User must have an bank account- ATM must be able to interact with the bank- User must be logged in somehow via card or pin
Condition for a success final	<ul style="list-style-type: none">- Whatever money the user inputs, it gets added into the user's account after they confirm it.
Actors	User in front of an ATM
Trigger condition (event)	User selecting the Deposit menu on an ATM

For the scenarios use the following template

Scenario name:

Ok or UC name with specific characteristic

Note: one action/response x line

	Actor action	System Response
1	User inserts ATM card.	////////
		System verifies ATM card is a valid card.
2	/////	System prompts for a pin number associated with account.
3	User enters the pin.	
4		System verifies the pin.
5		System displays main menu.
6	User selects Deposit.	
7		System prompts for deposit type (cash/check).
8	User selects deposit type.	
9		System prompts for the user to input the deposit.
10	User inserts the deposit.	
11		System verifies the deposit
12		System displays the amount deposited
13		System asks for confirmation on whether or not to deposit this
14	User confirms deposit amount	
15		System updates user's bank balance
16		System displays confirmation on deposit
17		System returns to user's main menu

* Do not write actions and responses in the same row

* use one row per system response

Sub-flows address how to handle failed verifications

- you need a sub-flow for each of the verifications in the scenario
- There are two possible options: 1. being able to handle errors or
2. Send a message and terminate UC
- A sub-flow follows the same rules as a scenario.
- Start your scenario with the same number of the associated verification activity and write the same verification activity
- Sub-flows may also include actors inputs if needed
- An example of a sub-flow that deals with errors and comebacks to step in the main

flow

Sub-flow name:

Failed deposit

	Actor action	System Response
#11		System verifies the deposit
#11.1		System fails to verify the deposit
#11.2		System displays invalid deposit type
#11.3		goto 7

Sub-flow name:		Verification atm card read
	Actor action	System Response
#2		System verifies ATM card is a valid card.
#2.1		System detects ATM card is not valid card
#2.2		System spits out card
#2.3		System displays Invalid card
#2.4		Goto 1

Sub-flow name:		Verification pin fail
	Actor action	System Response
#4		System verifies the pin.
#4.1		System sees that user entered pin != actual pin associated with account
#4.2		System displays Invalid pin
#4.3		Goto 2

Sub-flow name:		Manu menu verification
	Actor action	System Response
#4	User exits the main menu	System displays main menu.
#5		
#5.1		System displays that it logs out of the account
#4.3		Terminate UC

Sub-flow name:		Deposit verificaiton
	Actor action	System Response
#13	User inputs No	System asks for confirmation on whether or not to deposit this
#14		
#14.1		System ejects deposit type
#13.3		Goto 5

Alternate Scenarios: other ways to provide the same service (use case)

Alternate scenario name:	Deposit Cash
Alternate scenario name:	Quick deposit

Use the following checklist

Category	Criteria	Pass Y/N	Comment
Steps	are numbered	yes	Steps are numbered correctly
	sequential order	yes	Steps follow typical logical order
	Sub-flow # starts with the verification activity number	yes	Matches main flow steps
Flow	Start with an actor or system + verb	yes	
	Start with an actor	yes	
	After actor action must follow system actions	yes	
Sub-flows	must include system verifications	yes	verification steps at 2,4,11,13
	Sub-flow name is the failed verification activity	yes	
	start number at verification action number	yes	
	must terminate the scenario or go back to the scenario	yes	labeled with goto/terminate