

5.Scenario: CodeSharing Platform

Objects:

- a. Item:
DATA: Code, Quantity, Amount
- b. VendingMachine:
DATA: State
BEHAVIOR: obtainUserInput, obtainItemCode, sendRestockMessage, isAmountExact, popAmountToUser, pushChangetoUser, pushItemFromQueue
- c. UserEntry:
DATA: ItemCode,Amount
- d. Amount:
DATA: Value
- e. Inventory:
DATA: Item[] item
BEHAVIOR: ItemDetails

Instances:

VendingMachine vm
Item selecteditem
Inventory inventory
UserEntry userentry
Amount balance,change

Sequence:

```
vm.obtainUserInput -> userInput  
  
vm.State = "ENTER ITEM CODE"  
  
START:  
  
CASE OF vm.State  
  
CASE "ENTER ITEM CODE":  
  
    vm.obtainItemCode -> ItemCode : userentry.ItemCode  
  
    IF userentry.ItemCode IS EQUAL TO ERROR_USER_PRESSED_RESET  
        vm.State = "RESET"  
        break  
    ENDIF  
  
    inventory.ItemDetails -> userentry.ItemCode : selecteditem  
  
    IF selecteditem is NULL  
        Please enter Valid Code message  
  
    ELSE IF selecteditem.Quantity is EQUAL to 0  
        Item Unavailable, Please choose another item message  
        vm.sendRestockMessage -> selecteditem.Code, Quantity,admin.email  
    ELSE  
        vm.State = "ENTER AMOUNT"  
  
    ENDIF  
  
CASE "ENTER AMOUNT":  
    vm.obtainUserAmount -> Amount : userentry.Amount
```

```
IF userentry.Amount==ERROR_USER_PRESSED_RESET:
    vm.State = "RESET"
    break
ENDIF

vm.isAmountExact -> userentry.Amount, selecteditem.Amount : Balance
vm.ChangeToProvide = Balance
vm.State = "DISPATCH"

CASE "RESET":
    userentry.ItemCode = null
    IF userentry.amount IS NOT NULL
        vm.popAmountToUser -> userentry.amount
    ENDIF
    vm.State = "ENTER ITEM CODE"

CASE "DISPATCH":
    IF vm.ChangeToProvide is NOT NULL
        vm.pushChangetoUser -> vm.ChangeToProvide : Change
    ENDIF
    vm.pushItemFromQueue -> userentry.ItemCode : Item
ENDCASE

GOTO start
```