SOFTWARE SPECIFICATIONS

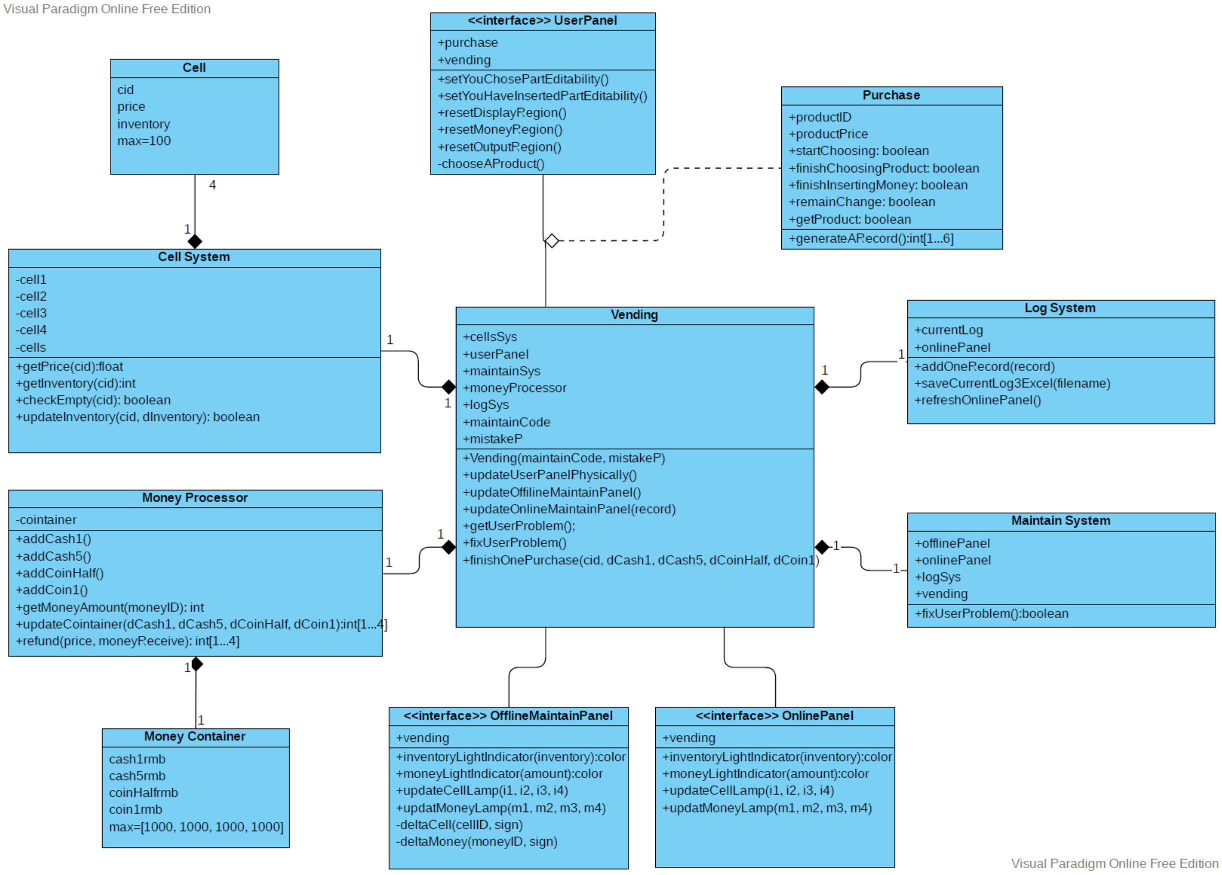
Vending Machine

Group 1
Author: Ziqi Gao

目录

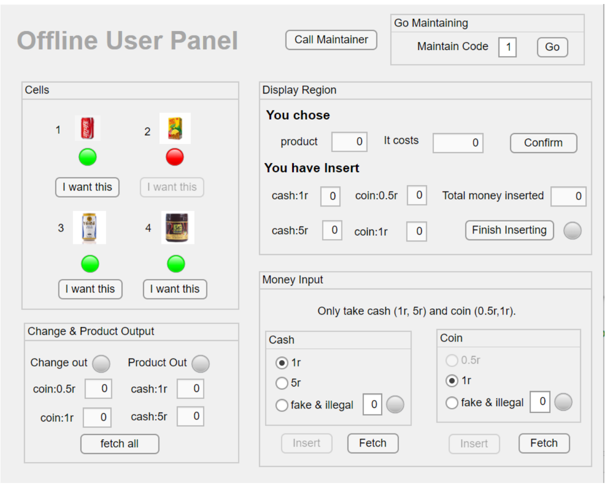
System Architecture

The system architecture is shown below:


Cell
cid
price
inventory
max=100
4
1
Cell System
-cell1
-cell2
-cell3
-cell4
-cells
+getPrice(cid):float
+getInventory(cid):int
+checkEmpty(cid): boolean
+updateInventory(cid, dInventory): boolean
<<interface>> UserPanel
+purchase
+vending
+setYouChosePartEditability()
+setYouHaveInsertedPartEditability()
+resetDisplayRegion()
+resetMoneyRegion()
+resetOutputRegion()
-chooseAProduct()
Purchase
+productID
+productPrice
+startChoosing: boolean
+finishChoosingProduct: boolean
+finishInsertingMoney: boolean
+remainChange: boolean
+getProduct: boolean
+generateARecord():int[1...6]
Vending
+cellsSys
+userPanel
+maintainSys
+moneyProcessor
+logSys
+maintainCode
+mistakeP
+Vending(maintainCode, mistakeP)
+updateUserPanelPhysically()
+updateOffilineMaintainPanel()
+updateOnlineMaintainPanel(record)
+getUserProblem();
+fixUserProblem()
+finishOnePurchase(cid, dCash1, dCash5, dCoinHalf, dCoin1)
1
1
Log System
+currentLog
+onlinePanel
+addOneRecord(record)
+saveCurrentLog3Excel(filename)
+refreshOnlinePanel()
1
1
Money Processor
-cointainer
+addCash1()
+addCash5()
+addCoinHalf()
+addCoin1()
+getMoneyAmount(moneyID): int
+updateCointainer(dCash1, dCash5, dCoinHalf, dCoin1):int[1...4]
+refund(price, moneyReceive): int[1...4]
1
1
Maintain System
+offlinePanel
+onlinePanel
+logSys
+vending
+fixUserProblem():boolean
1
1
1
1
Money Container
cash1rmb
cash5rmb
coinHalfrmb
coin1rmb
max=[1000, 1000, 1000, 1000]
<<interface>> OfflineMaintainPanel
+vending
+inventoryLightIndicator(inventory):color
+moneyLightIndicator(amount):color
+updateCellLamp(i1, i2, i3, i4)
+updatMoneyLamp(m1, m2, m3, m4)
-deltaCell(cellID, sign)
-deltaMoney(moneyID, sign)
<<interface>> OnlinePanel
+vending
+inventoryLightIndicator(inventory):color
+moneyLightIndicator(amount):color
+updateCellLamp(i1, i2, i3, i4)
+updatMoneyLamp(m1, m2, m3, m4)

Software Specifications

(S0:) Overview and Mode Switch



Maintain Code 1

You chose

product 0 It costs 0

You have Insert

cash:1r 0 coin:0.5r 0 Total money inserted 0

cash:5r 0 coin:1r 0

Only take cash (1r, 5r) and coin (0.5r,1r).

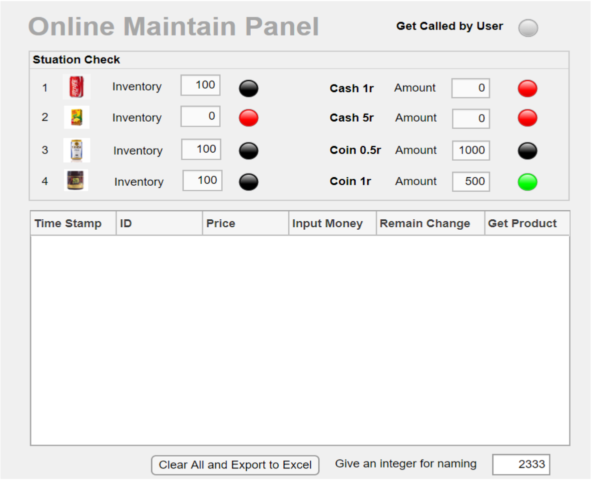
Cash: ◉ 1r ○ 5r ○ fake & illegal 0

Coin: ○ 0.5r ◉ 1r ○ fake & illegal 0

Change out Product Out

coin:0.5r 0 cash:1r 0

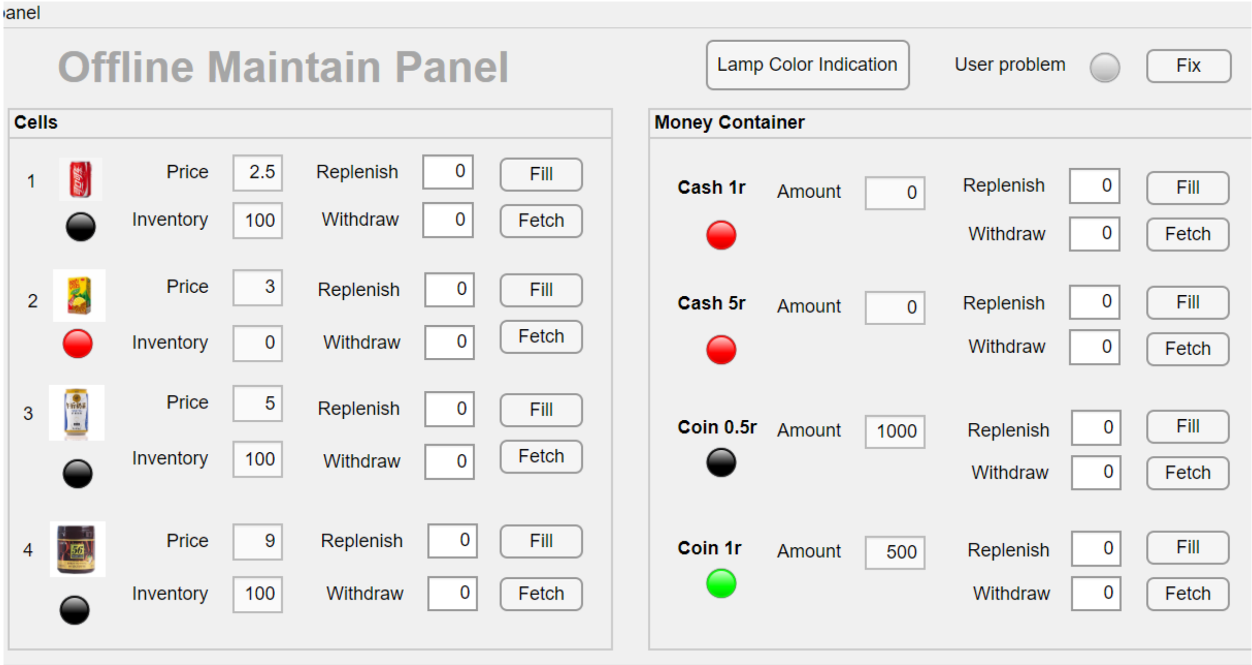
coin:1r 0 cash:5r 0



1	Inventory	100	Cash 1r	Amount	0
2	Inventory	0	Cash 5r	Amount	0
3	Inventory	100	Coin 0.5r	Amount	1000
4	Inventory	100	Coin 1r	Amount	500

Time Stamp	ID	Price	Input Money	Remain Change	Get Product

Give an integer for naming 2333



1	Price	2.5	Replenish	0
	Inventory	100	Withdraw	0
2	Price	3	Replenish	0
	Inventory	0	Withdraw	0
3	Price	5	Replenish	0
	Inventory	100	Withdraw	0
4	Price	9	Replenish	0
	Inventory	100	Withdraw	0

Cash 1r	Amount	0	Replenish	0
			Withdraw	0
Cash 5r	Amount	0	Replenish	0
			Withdraw	0
Coin 0.5r	Amount	1000	Replenish	0
			Withdraw	0
Coin 1r	Amount	500	Replenish	0
			Withdraw	0

The software has 3 panels in total, which are offline user panel, offline maintain panel and online maintain panel respectively and shown in one device. Since only one device is used to display, the difference of different kind of users is not expressed as clear as a real vending machine; however, we've tried our best to express the differences by 2 means:

✓ Command line Input: The controller (vending) takes 2 input parameters, the first of which is maintainCode.

1. If maintainCode == a preset number (1 in the implementation)
   a. Offline User Panel, Offline Maintain Panel, and Online Maintain Panel are editable.
2. Else

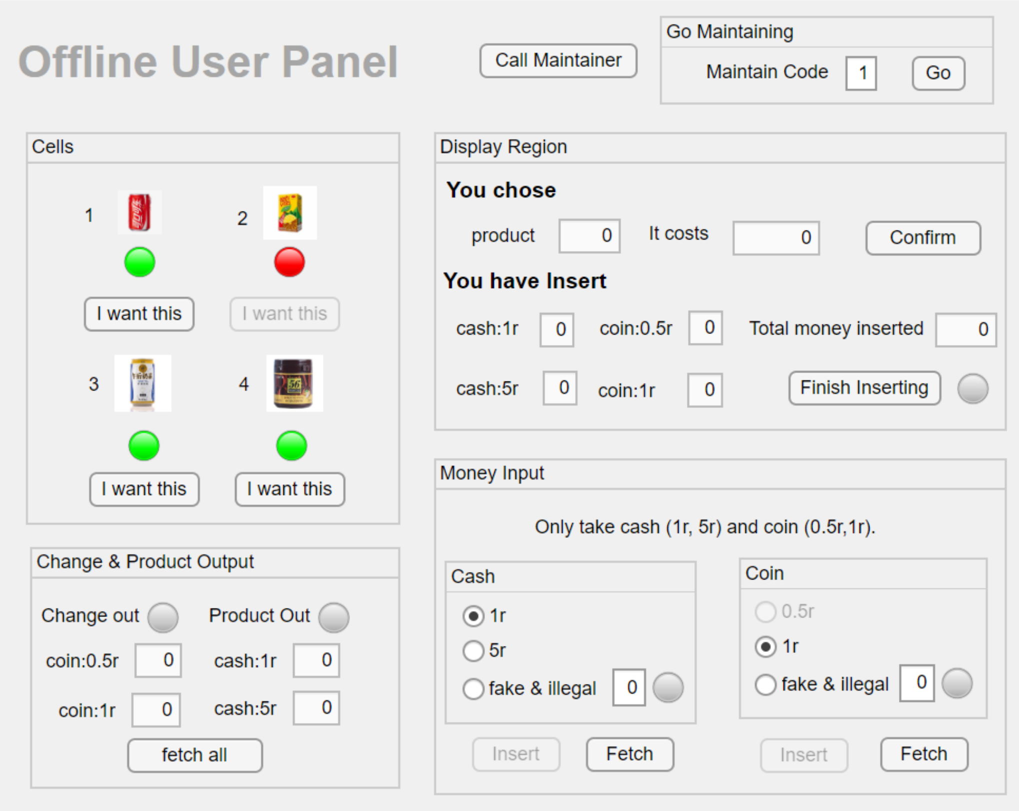
a. Offline User Panel and Online Maintain Panel are editable; Offline Maintain Panel is not editable.

✓ User Interface Input: The user interface has a panel named “go maintaining”, and can take a maintainCode as input.

1. If maintainCode == a preset number (1 in the implementation)
   a. The Offline Maintain Panel becomes editable.
2. Else
   a. The Offline Maintain Panel keeps its original editable state.

The Online Maintain Panel and User Panel is set to be always editable since buying and online checking should be done all the time.

S1: User Mode Implementation



Maintain Code 1

product 0 It costs 0

cash:1r 0 coin:0.5r 0 Total money inserted 0

cash:5r 0 coin:1r 0

Only take cash (1r, 5r) and coin (0.5r,1r).

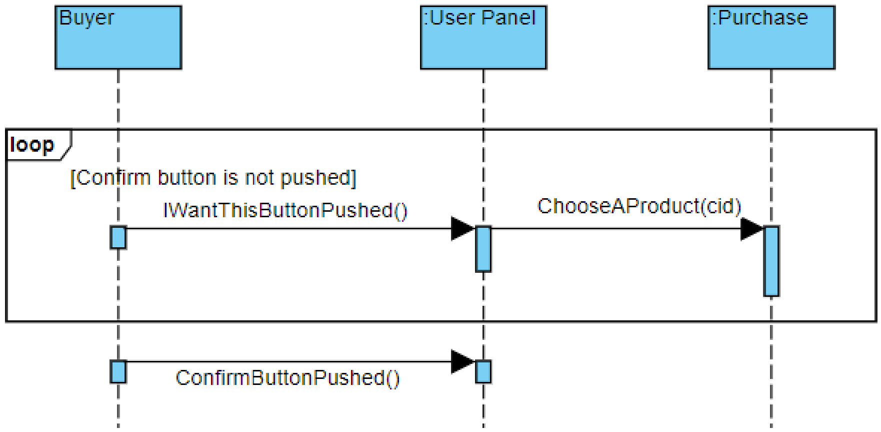
Cash: ● 1r ○ 5r ○ fake & illegal 0

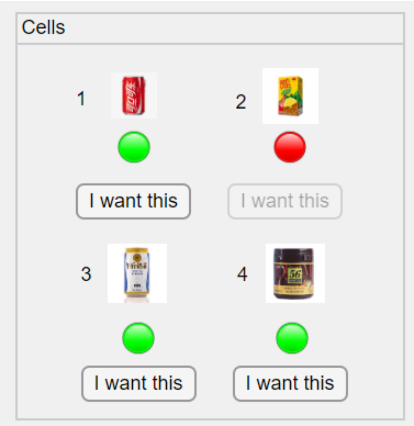
Coin: ○ 0.5r ● 1r ○ fake & illegal 0

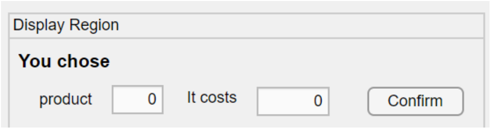
coin:0.5r 0 cash:1r 0

coin:1r 0 cash:5r 0

S1.1: Select Product



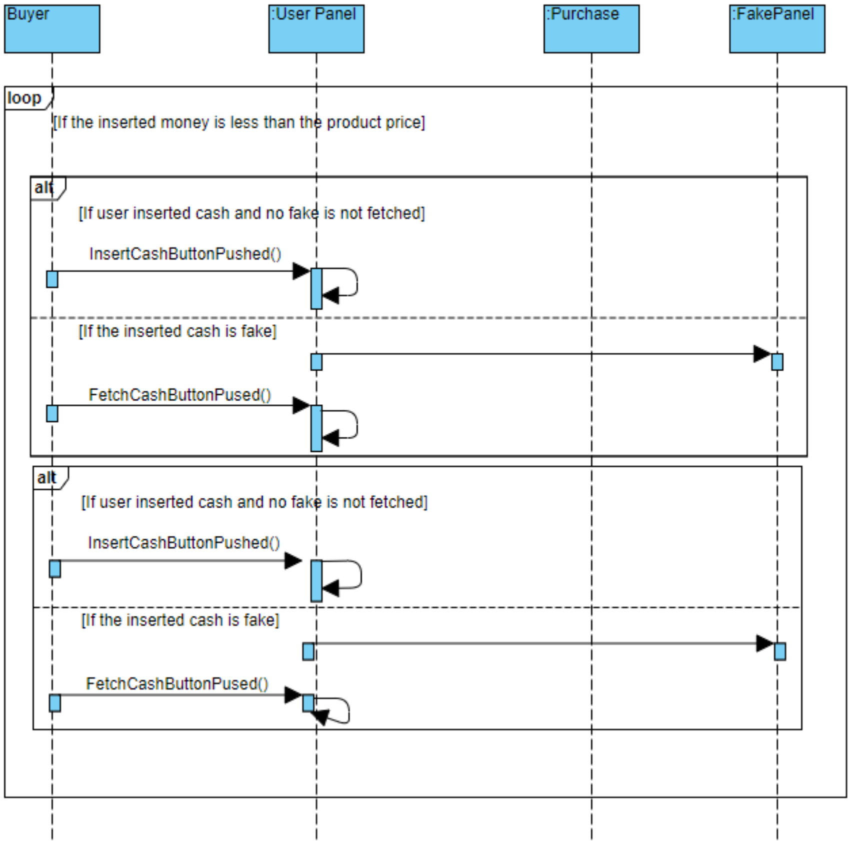


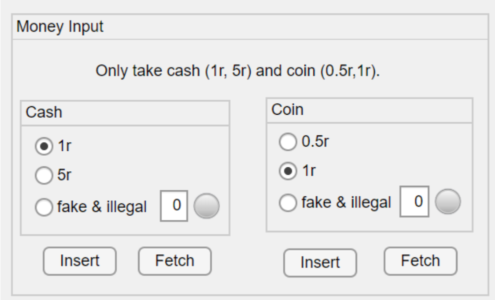


product 0 It costs 0

- S1.1.1: Select Product and view the product information
  1. There are 4 “I want this” buttons, and the corresponding IwantthisButtonPushed() function called by the corresponding button
  2. app.ChooseAProduct(cid) is called from the call back function of “I want this button” with corresponding cell id as input
     a. If the current purchase has not finished
        i. Reset product information in the purchase
        ii. Reset purchase status in the purchase
        iii. Reset display information in the user panel
- S1.1.2: Press “Confirm” Button
  1. ConfirmButtonPushed() function get called
     a. If the current purchase has been started and does not finish choosing product or insert money before
        i. Don’t allows for changing product anymore
        ii. Set the status of the current purchase to “finishChoosingProduct”
        iii. Allows money insertion in User Panel

S1.2 Input Money


Buyer
:User Panel
:Purchase
:FakePanel
loop
[If the inserted money is less than the product price]
alt
[If user inserted cash and no fake is not fetched]
InsertCashButtonPushed()
[If the inserted cash is fake]
FetchCashButtonPused()
alt
[If user inserted cash and no fake is not fetched]
InsertCashButtonPushed()
[If the inserted cash is fake]
FetchCashButtonPused()

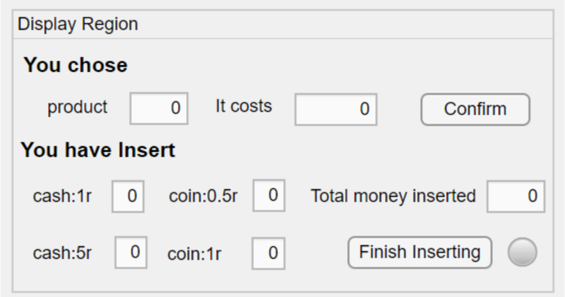


Money Input

Only take cash (1r, 5r) and coin (0.5r,1r).

Cash
● 1r
○ 5r
○ fake & illegal 0

Coin
○ 0.5r
● 1r
○ fake & illegal 0



Display Region

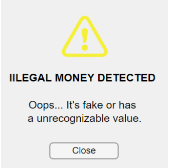
You chose

product 0 It costs 0

You have Insert

cash:1r 0 coin:0.5r 0 Total money inserted 0

cash:5r 0 coin:1r 0



IILEGAL MONEY DETECTED

Oops... It's fake or has a unrecognizable value.

- S1.2.1: Insert Money by selecting in “Money Input” Panel
  1. While the inserted money is less than the product price
     a. Insert Cash
        i. If No fake or illegal money is left, InsertCashButtonPushed() is called and the “You have insert” Part is updated in real time
           1. If the inserted cash is fake or illegal
              a. FakePanel() is called and the display text and lamp in the right of the “fake&illegal” becomes 1 and red respectively.
              b. User is not allowed to insert any cash before fetch the fake cash.

ii. Else, user cannot input cash before pressing the fetch button, after which InsertCashButtonPushed() is called.

b. Insert Coin

  i. If No fake or illegal money is left, InsertCoinButtonPushed() is called and the “You have insert” Part is updated in real time

   1. If the inserted coin is fake or illegal

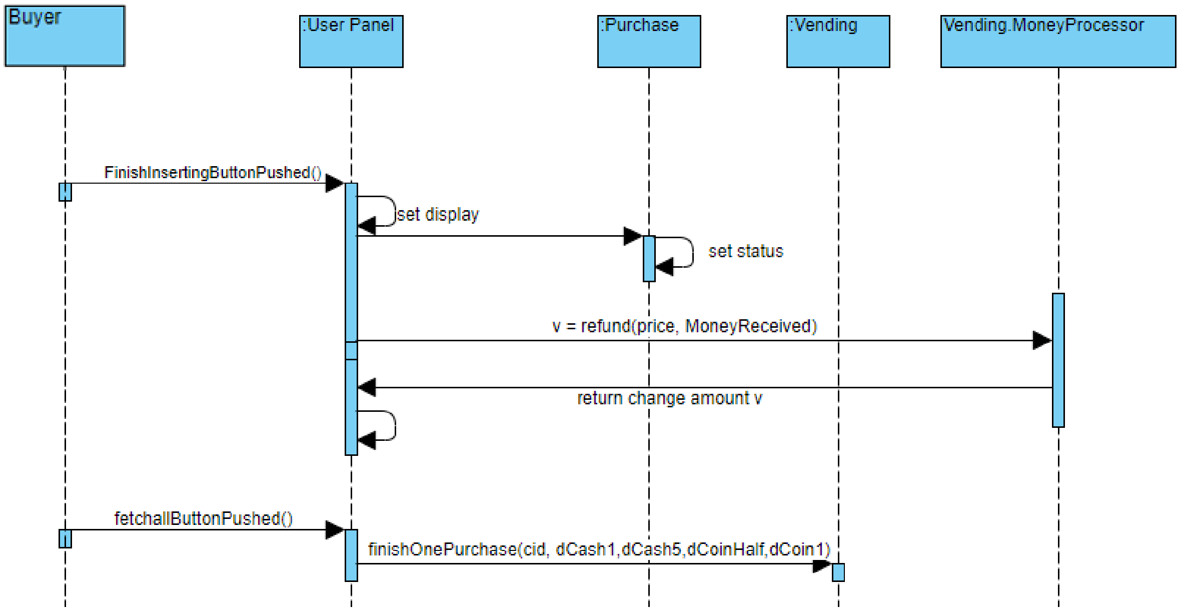
    a. FakePanel() is called and the display text and lamp in the right of the “fake&illegal” becomes 1 and red respectively.

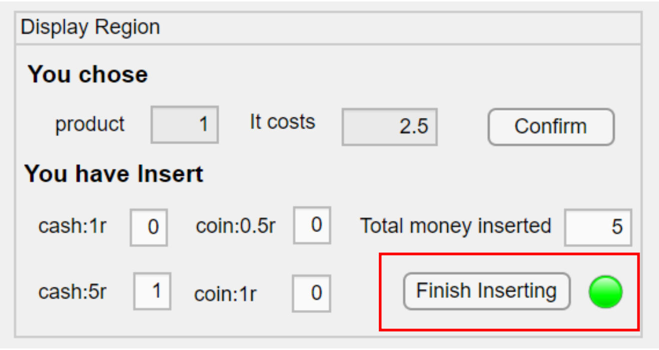
    b. User is not allowed to insert any coin before fetch the fake coin.

  ii. Else, user cannot input coin before pressing the fetch button, after which InsertCashButtonPushed() is called.

2. If the inserted money is equal to or surpass the selected product’s price, insertion is not allowed and the “finish inserting button” is set to be editable and its corresponding lamp is green for indication.

S1.3 Get Change and product




Display Region

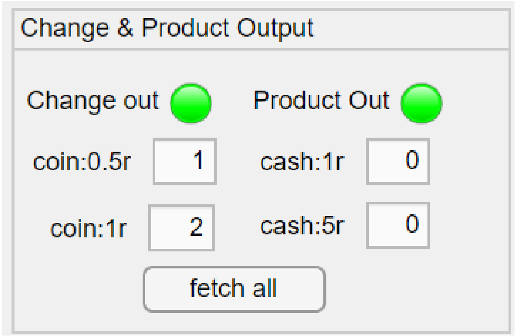
You chose

product 1 It costs 2.5 Confirm

You have Insert

cash:1r 0 coin:0.5r 0 Total money inserted 5

cash:5r 1 coin:1r 0 Finish Inserting ●


Change & Product Output

Change out ● Product Out ●

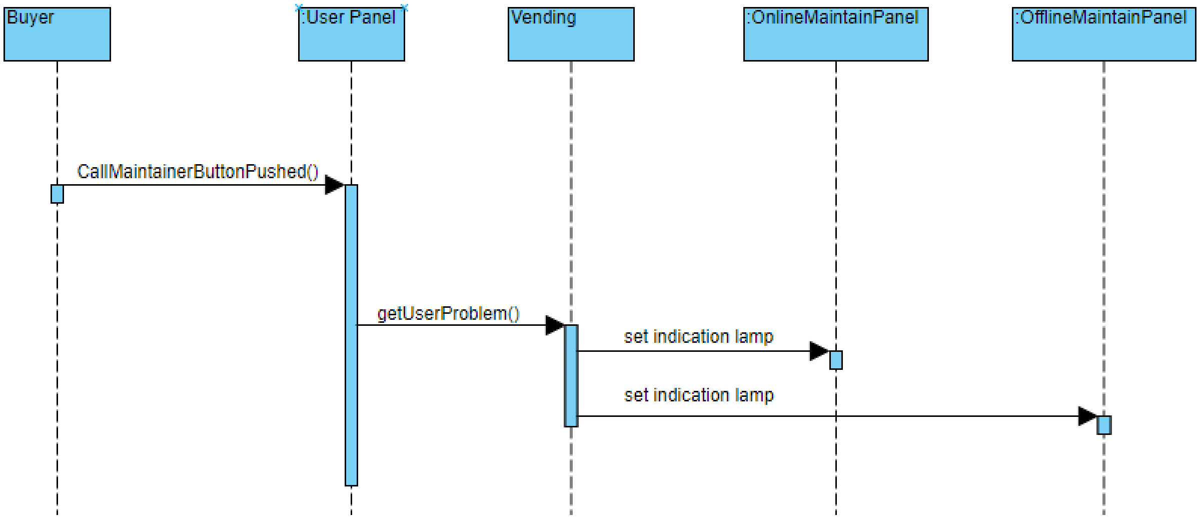
coin:0.5r 1 cash:1r 0

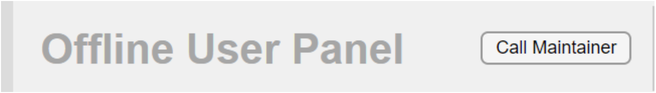
coin:1r 2 cash:5r 0

fetch all

- S1.3.1: "Finish Inserting" button is pushed
  1. Set User Panel's display text and indication lamp in "Change & Product Output", indicating whether change and product are out respectively
  2. Compute correct refund by calling app.vending.refund(price, MoneyReceived)
     a. If money in the vending is enough for change, return v = [numOfCoinHalf, numOfCoin1]
     b. Else, return v= [-1, remainChange, coinHalf, coin1]
  3. Update the lamps and texts of "Change & Product Output" Region according to v.
  4. Set the status of the current purchase
- S1.3.2: "fetch all" button is pushed
  1. Reset the lamps and texts of "Change & Product Output" Region
  2. If there is enough money for change, Call Vending.finishOnePurchase(cid, dCash1, dCoinHalf, dCoin1), the system behaviour will be explained in S4
  3. Else, call vending.finishOnePurchase(-1,0,0,0,0), the system behaviour will be explained in S4

S1.4 Call Maintainer

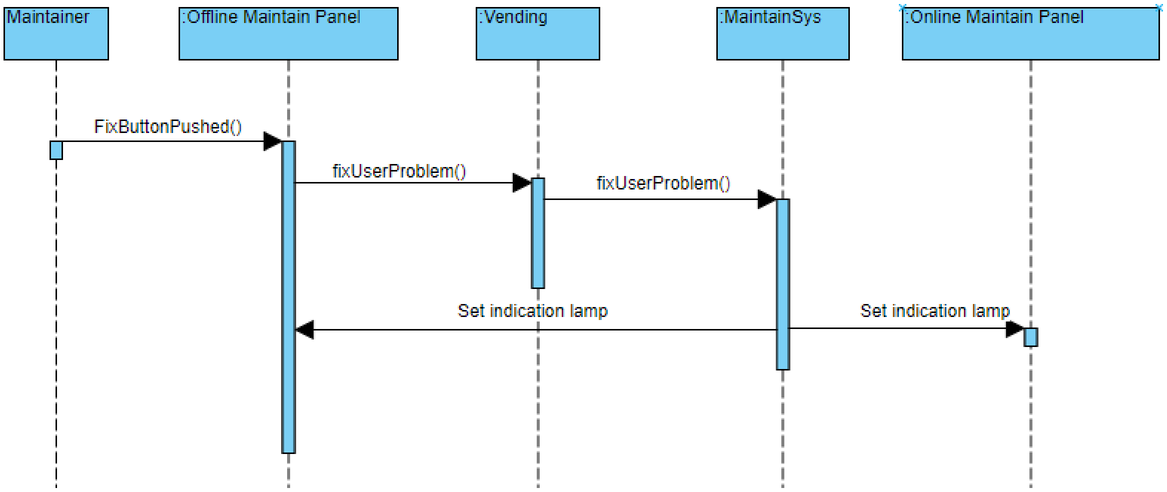


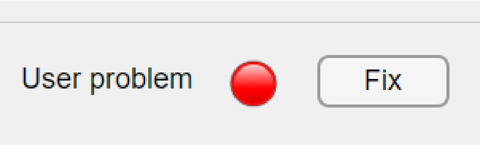


- S1.4.1 Press “Call Maintainer” button
  1. CallMaintainerButtonPushed() gets called
     a) Vending.getUserProblem gets called and set indication lamps on online maintain panel and offline maintain panel

S2: Offline Maintain Mode Implementation

S2.1 Answer User’s Call



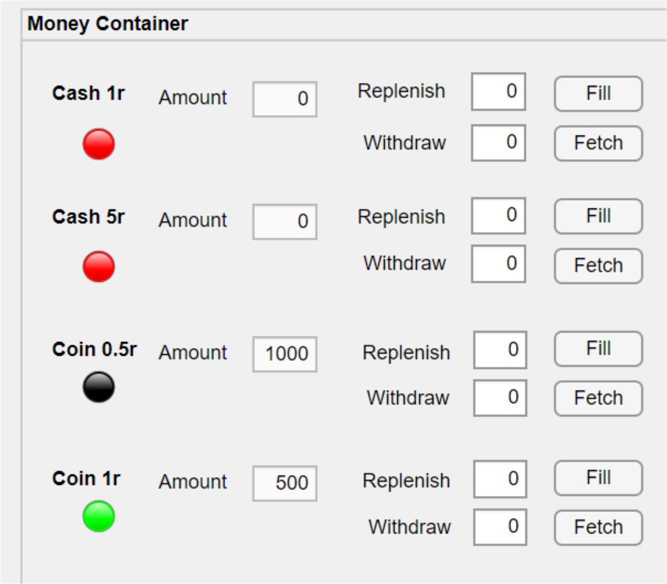


User problem 🔴 Fix

- S2.1.1 Press “Fix” button

1. Call vending.maintainSys.fixUserProblem()
   a. Solve the problem for user
   b. Set user problem indication lamps into grey on Online Maintain Panel and Offline Panel

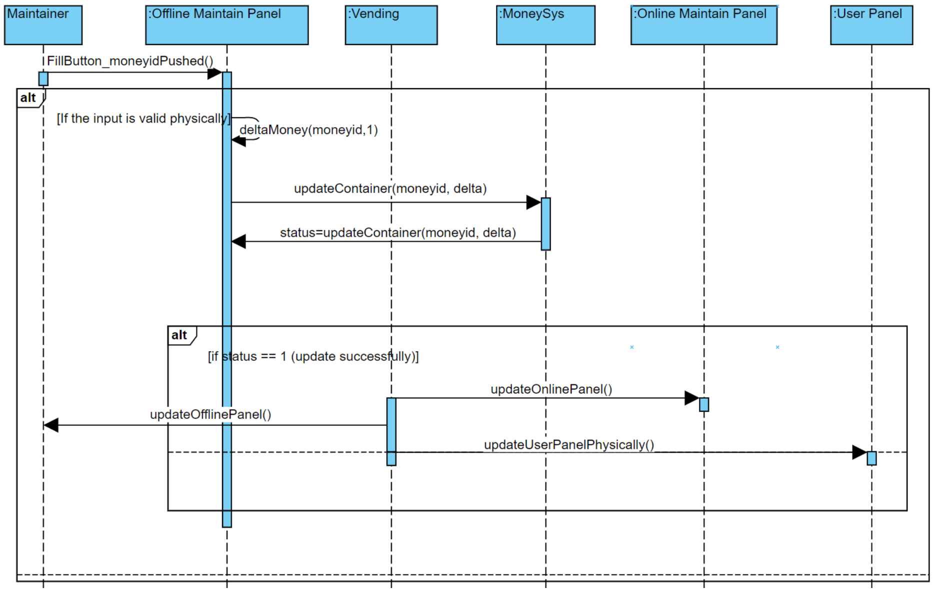
S2.2 Replenish and Withdraw Money



Money Container

Cash 1r	Amount	0	Replenish	0	Fill
🔴			Withdraw	0	Fetch
Cash 5r	Amount	0	Replenish	0	Fill
🔴			Withdraw	0	Fetch
Coin 0.5r	Amount	1000	Replenish	0	Fill
⚫			Withdraw	0	Fetch
Coin 1r	Amount	500	Replenish	0	Fill
🟢			Withdraw	0	Fetch

- S2.2.1 Replenish Money



Maintainer
:Offline Maintain Panel
:Vending
:MoneySys
:Online Maintain Panel
:User Panel
FillButton_moneyidPushed()
alt
[If the input is valid physically]
deltaMoney(moneyid,1)
updateContainer(moneyid, delta)
status=updateContainer(moneyid, delta)
alt
[if status == 1 (update successfully)]
updateOnlinePanel()
updateOfflinePanel()
updateUserPanelPhysically()

1. Select a money amount to replenish and input in the corresponding text and press the corresponding “fetch” button. The corresponding callback will be called.

a) If the input is not an integer in [0,inf], ignore it (done by setting in mlapp directly)

b) Else if the input is valid physically

i. Call app.deltaCell(cid,1)

ii. Call app.vending.updateInventory(cid) and get updating status from the vending. (updateInventory() will be discussed in S4)

iii. If the update in inventory is done successfully, update the online panel, offline panel and user panel respectively.

- S2.2.2 Withdraw Money

1. Select a money amount to withdraw and input in the corresponding text and press the corresponding “fetch” button. The corresponding callback will be called.

a) If the input is not an integer in [0,inf], ignore it (done by setting in mlapp directly)

b) Else if the input is valid physically

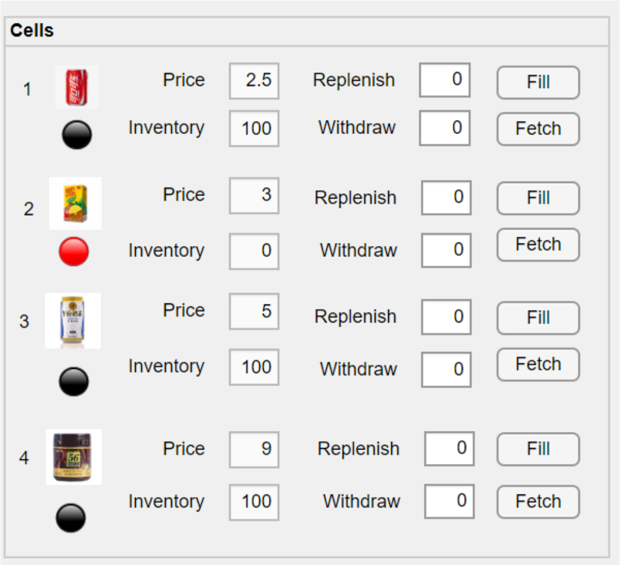
iv. Call app.deltaCell(cid,-1)

v. Call app.vending.updateInventory(cid) and get updating status from the vending. (updateInventory() will be discussed in S4)

vi. If the update in inventory is done successfully, update the online panel, offline panel and user panel respectively.

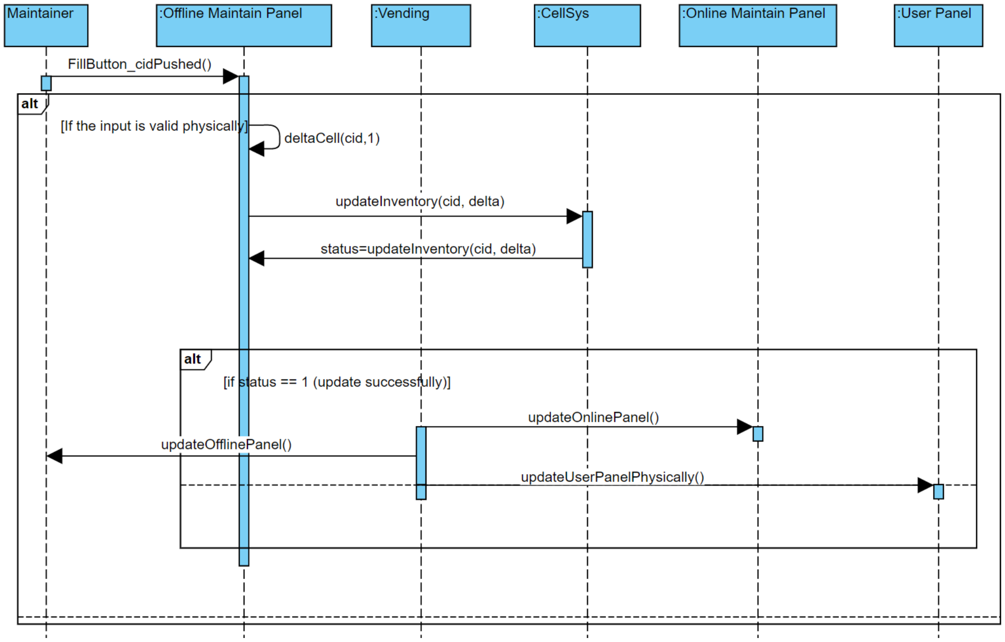
S2.3 Replenish and Withdraw Product

- S2.3.1 Replenish Product



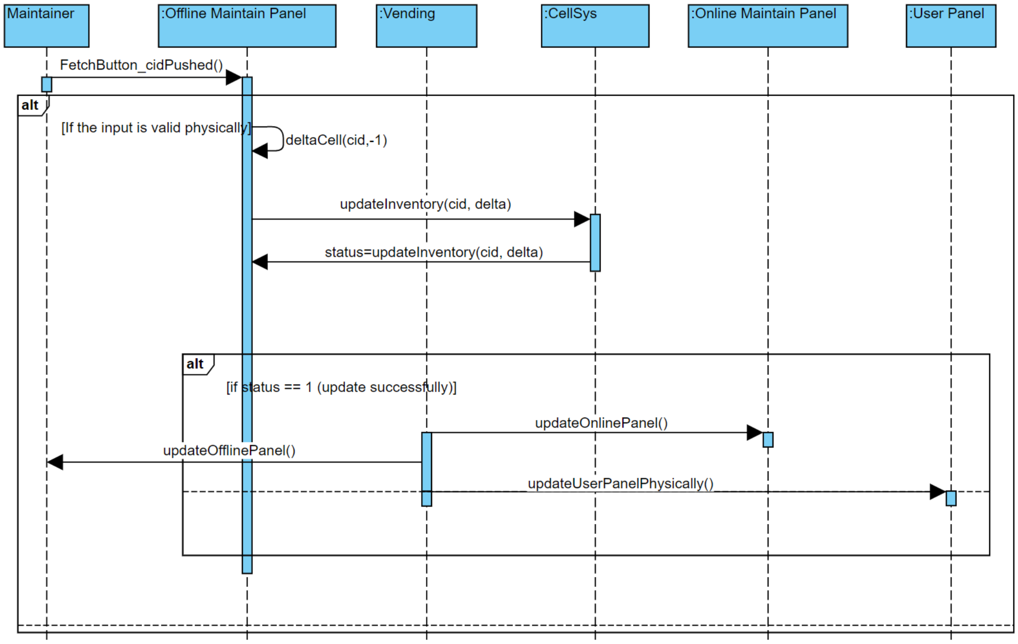
Cells

1		Price	2.5	Replenish	0	Fill
	●	Inventory	100	Withdraw	0	Fetch
2		Price	3	Replenish	0	Fill
	●	Inventory	0	Withdraw	0	Fetch
3		Price	5	Replenish	0	Fill
	●	Inventory	100	Withdraw	0	Fetch
4		Price	9	Replenish	0	Fill
	●	Inventory	100	Withdraw	0	Fetch



- Select a product to replenish and input in the corresponding text and press the corresponding “fetch” button. The corresponding callback will be called.
  - If the input is not an integer in [0,inf], ignore it (done by setting in mlapp directly)
  - Else if the input is valid physically
    - Call app.deltaMoney(cid,1)
    - Call app.vending.updateContainer(cid) and get updating status from the vending. (updateContainer () will be discussed in S4)
    - If the update in container is done successfully, update the online panel, offline panel and user panel respectively.

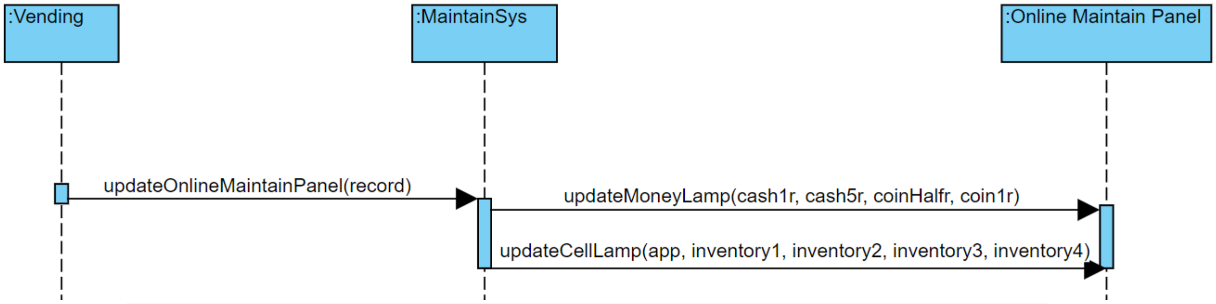
- S2.3.2 Withdraw Product

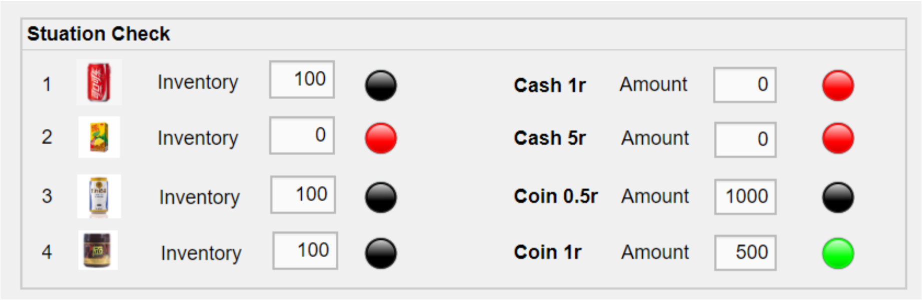


1. Select a product to withdraw and input in the corresponding text and press the corresponding "fetch" button. The corresponding callback will be called.
   a) If the input is not an integer in [0,inf], ignore it (done by setting in mlapp directly)
   b) Else if the input is valid physically
      i. Call app.deltaMoney (cid,-1)
      ii. Call app.vending.updateContainer(cid) and get updating status from the vending. (updateContainer () will be discussed in S4)
      iii. If the update in money container is done successfully, update the online panel, offline panel and user panel respectively.

S3: Online Maintain Mode and Log System Implementation

S3.1 Check Current Situation of the Vending Machine


:Vending
:MaintainSys
:Online Maintain Panel
updateOnlineMaintainPanel(record)
updateMoneyLamp(cash1r, cash5r, coinHalfr, coin1r)
updateCellLamp(app, inventory1, inventory2, inventory3, inventory4)

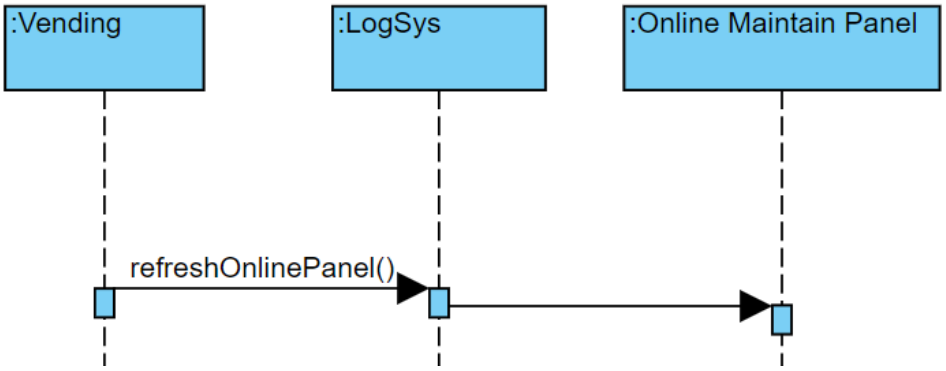


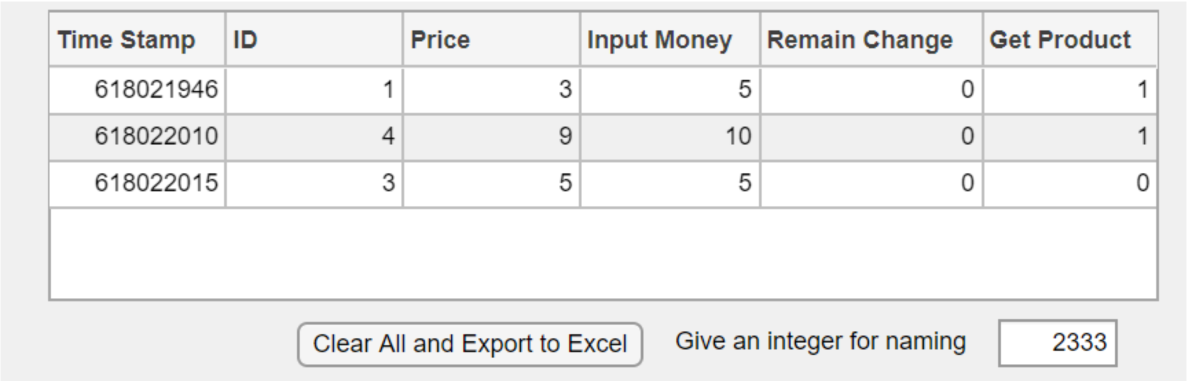
Stuation Check

1	Inventory	100	●	**Cash 1r**	Amount	0	🔴
2	Inventory	0	🔴	**Cash 5r**	Amount	0	🔴
3	Inventory	100	●	**Coin 0.5r**	Amount	1000	●
4	Inventory	100	●	**Coin 1r**	Amount	500	🟢

- S3.1.1 Display the inventory of products
  1. Vending calls updateOnlineMaintainPanel(record), usually with a record
  2. UpdateCellLamp(i1,i2,i3,i4) is called immediately after, which updates the value in the texts too.
- S3.1.2 Display the amount of money
  1. Vending calls updateOnlineMaintainPanel(record), usually with a record
  2. UpdateMoneyLamp(i1,i2,i3,i4) is called immediately after, which updates the value in the texts too.

S3.2 View Log



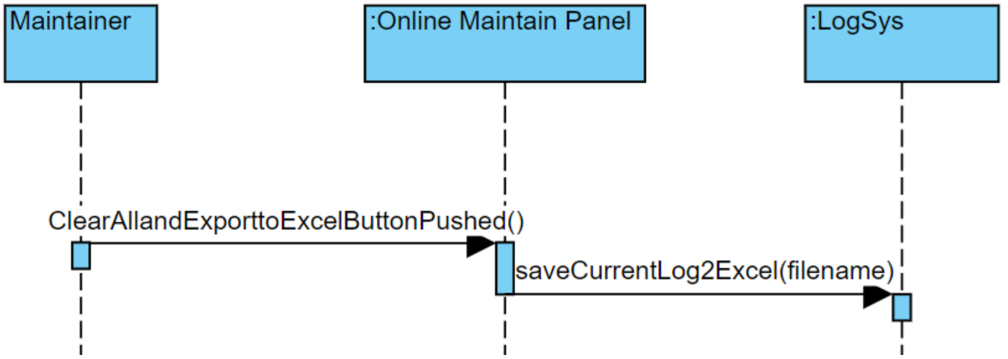


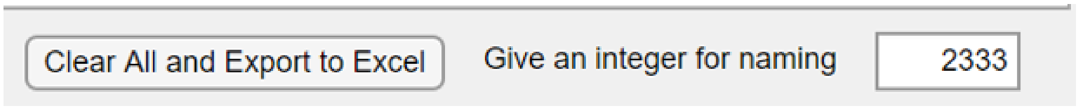
Time Stamp	ID	Price	Input Money	Remain Change	Get Product
618021946	1	3	5	0	1
618022010	4	9	10	0	1
618022015	3	5	5	0	0

Give an integer for naming 2333

- S3.2.1 When the controller calls refreshOnlinePanel(), the log is refreshed. A record of the log system include a time stamp, which is composed of month_date_hour_minute_second, a product ID, a product price, a total input money, a remain change, and a flag indicating whether the user get the product, namely “Get Product”.

S3.3 Export Log


Maintainer
:Online Maintain Panel
:LogSys
ClearAllandExporttoExcelButtonPushed()
saveCurrentLog2Excel(filename)

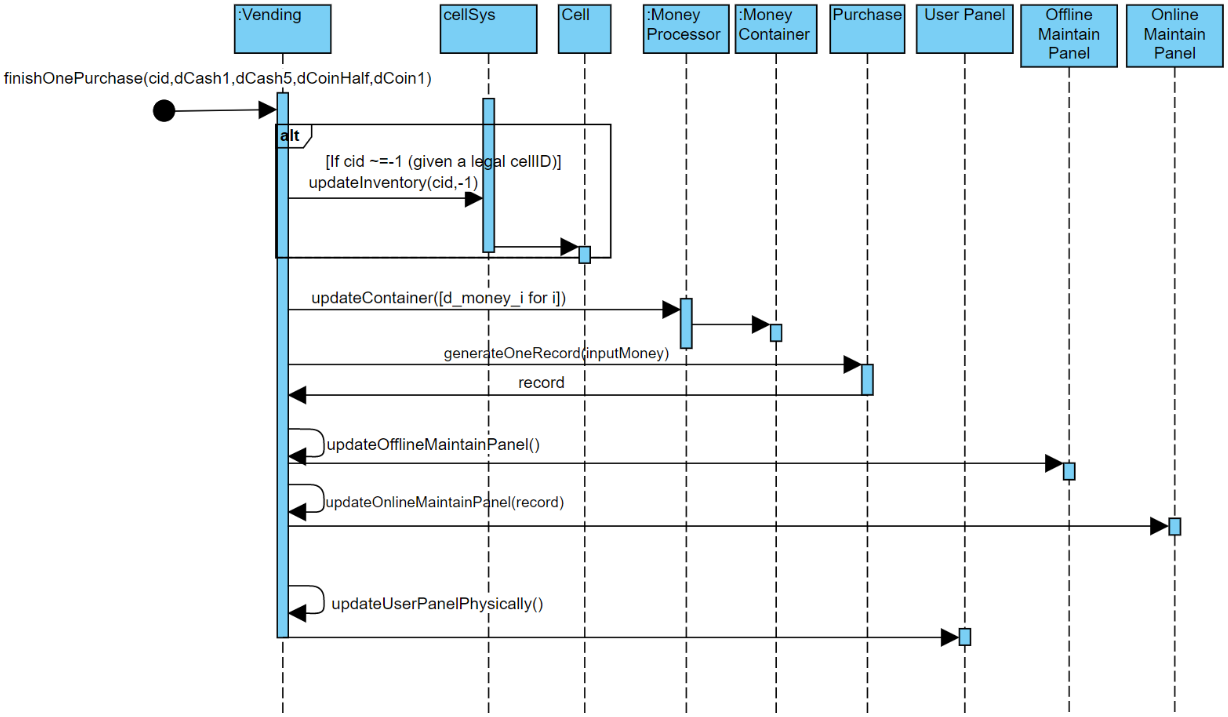


Give an integer for naming 2333

- S3.3.1 When the maintainer presses “Clear All and Export to Excel”, the log is clear in online maintain system and saved as an excel “xxx.xlsx” file.
  1. If the user does not specify the name, xxx==current time stamp
  2. Else if the user specifies an invalid name, xxx=2333
  3. Else xxx=the input in the text box.

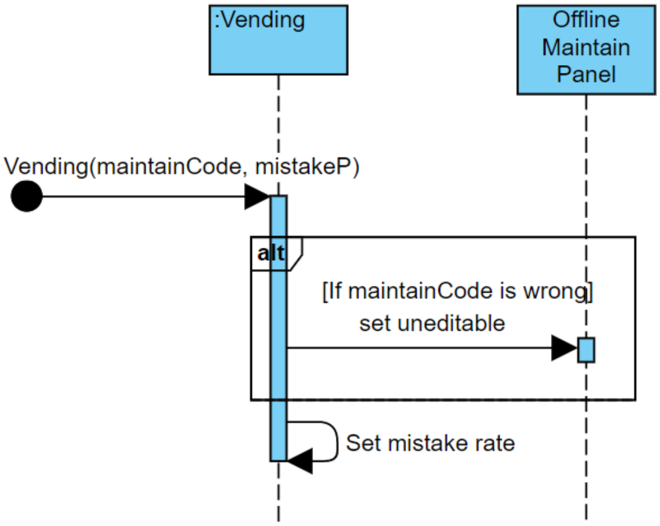
S4: Vending and Systems Implementation (Supplementary to S1, S2, S3)

S4.1 Update Systems After Finishing One Purchase


:Vending
cellSys
Cell
:Money Processor
:Money Container
Purchase
User Panel
Offline Maintain Panel
Online Maintain Panel
finishOnePurchase(cid,dCash1,dCash5,dCoinHalf,dCoin1)
alt
[If cid ~=-1 (given a legal cellID)]
updateInventory(cid,-1)
updateContainer([d_money_i for i])
generateOneRecord(inputMoney)
record
updateOfflineMaintainPanel()
updateOnlineMaintainPanel(record)
updateUserPanelPhysically()

- S4.1.1 If not given an illegal CellD, update inventory of cell CellID with delta = -1. (One product inventory decreases in one cell.)
- S4.1.2 Update Money Container with dCash1, dCash5, dCoinHalf, dCoin1 using moneyProcessor.updateContainer(dCash1, dCash5, dCoinHalf, dCoin1). This updates the money container.
- S4.1.3 Generate One Record of purchase using Purchase.generateOneRecord() and get the record.
- S4.1.4 Update Offline Maintain Panel updateOfflineMaintainPanel()
- S4.1.5 Update Online Maintain Panel with that record with updateOnlineMaintainPanel()
- S4.1.6 Update User Panel by updateUserPanelPhysically()

S4.2 Vending Initialization: checking maintain code and setting machine error rate



- S4.2.1 Initialize the whole system with vending(maintainCode, mistakeP)
    1. If maintainCode != a reset string (1 in the implementation)
        a) The offline panel will be set uneditable
    2. Set the mistake rate of failing outputting the selected product of the vending machine to mistake.