DISTRIBUTED VENDING MACHINE

객체지향개발방법론 [T2]

- 201711391 류근범
 202282050 조문충
 - 202013557 하지라
- 201711436 홍윤표 ●







OOPT Stage 2000 : Build | Phase 2040 Cycle 1: Design

- 2141. Design Real Use Cases
- 2142. Define Reports, UI, and Storyboards
- 2143. Refine System Architecture
- 2144. Define Interaction Diagrams
- 2145. Define Design Class Diagrams
- 2146. Design Traceability Analysis



Use Case	1. Main Menu Display
Actor	System
Purpose	display main menu of the VM
Overview	System displays item types and other features upon starting the machine.
Туре	Evident
Cross Reference	Functions: R 1.1
Pre-requisites	Administrator turns the machine on before User uses System.
Typical Courses of Events	(S) System (S) displays Window1(system's main window). In window-1, button A(item type display), B(admin login), and C(verification code input) is prompted.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A



Use Case	2. Item Selection	
Actor	User	
Purpose	select item User want to buy	
Overview	User selects the type and quantity of the item. System checks stock of item user chosen. If stocks are available, System goes to use case to use case (4. Request Stock of Other VM).	
Туре	Evident	
Cross Reference	Function R 1.2	
Pre-requisites	Function R 1.1 needs to be executed before.	
Typical Courses of Events	 (U) User (S) System (U) presses A in Window1. (S) displays Window2. In Window2, button A(quantity select), B(go home), C(next step) is prompted. 	(U) selects quantity with A, and presses B.(S) checks the stock of selected item in System.(S) If stocks are available, goes to use case (3. Payment) [E1].
Alternative Courses of Events	 (U) selects quantity with A, and presses B. (S) checks the stock of selected item in System. (S) If stocks are unavailable, goes to use case (4. Request Stock of Other VM) . 	
Exceptional Courses of Events	[E1](S) If stocks are not available (after use case (4. Request Stock of Other VM), sends error message to User and System goes to Window1	



Use Case	3. Payment
Actor	User
Purpose	proceed payment on VM
Overview	System displays order sheet. User types card info, then System checks card's validity and if validation is successful, completes the payment.
Туре	Evident
Cross Reference	Function R 2.1
Pre-requisites	Function R 1.1 to 1.2 need to be executed before.
Typical Courses of Events	 (U) User (S) System (S) displays Window3-1. (U) presses A in Window3-1. (S) displays Window4. (U) types card info and presses A in Window4. (S) checks card's validity[E1] and completes the payment[E2] and displays Window5.
Alternative Courses of Events	N/A
Exceptional Courses of Events	 [E1] System shows message to user "Error in card's validity. Please try again". If validity failure happened more than three times, return to Window1. [E2] System shows message to user "Error in payment. Please try again". If payment failure happened more than three times, return to Windo1.



Use Case	4. Request Stock of Other VM
Actor	System
Purpose	request stock of other VM to check stock availability.
Overview	As System does not have the stock user chose, requests stock from the other VM.
Туре	Hidden
Cross Reference	Function R 3.1
Pre-requisites	Executed as Alternative Courses of Events of use case (2. Item Selection)
Typical Courses of Events	 (S) System (OVM) Other VM (S) sends stock message of selected item. (OVM) responds message of stock information. (S) receives message.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A



Use Case	5. Provide Stock Information to Other VM
Actor	Other VM
Purpose	provide stock information to Other VM
Overview	Other VM sends stock check message to System. System sends message if has stock of requested item.
Туре	Evident
Cross Reference	Function R 3.2
Pre-requisites	N/A
Typical Courses of Events	(S) System (OVM) Other VM(OVM) sends stock message to System.(S) If System has stock of requested items, sends message[E1].
Alternative Courses of Events	N/A
Exceptional Courses of Events	[E1] If System doesn't have stock of requested item, System doesn't send message.



Use Case	6. Provide Product Information to Other VM
Actor	Other VM
Purpose	provide product information to the Other VM.
Overview	Other VM sends product check message to System. System sends message if has product data of requested item.
Туре	Evident
Cross Reference	Function R 3.3
Pre-requisites	N/A
Typical Courses of Events	(S) System (OVM) Other VM(OVM) sends product check message to System.(S) If System has product data of requested items, sends message[E1].
Alternative Courses of Events	N/A
Exceptional Courses of Events	[E1] If System doesn't have any product data, System doesn't send message.



Use Case	7. Provide Prepayment Information to Other VM
Actor	System
Purpose	provide prepayment information to other VM
Overview	After prepayment happens and verification was made, send info of prepayment.
Туре	Hidden
Cross Reference	Function R 3.4
Pre-requisites	Function R 4.1 need to be executed before.
Typical Courses of Events	(S) System (S) sends verification and item info to Other VM.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A



Use Case	8. Prepayment
Actor	User
Purpose	User can prepay items they wanted to buy at other VM.
Overview	System gives options to user to prepay the stocks they chose which are available at other VM. User may proceed with prepayment if they agree to proceed with the transaction.
Туре	Evident
Cross Reference	Function R 4.1, R 4.3
Pre-requisites	Function R 3.1 need to be executed before.
Typical Courses of Events	 (U) User (S) System (S) displays Window3-2. (U) presses A in Window3-2. (S) displays Window4. (U) types card info and presses A in Window4[E1]. (S) checks card's validity[E1] and completes the payment[E2] and creates verification for this order and displays Window5
Alternative Courses of Events	N/A
Exceptional Courses of Events	 [E1] System shows message to user "Error in card's validity. Please try again". If validity failure happened more than three times, return to Window1. [E2] System shows message to user "Error in payment. Please try again". If payment failure happened more than three times, return to Window1.



Use Case	9. Synchronize Information of Prepayment
Actor	Other VM
Purpose	synchronize prepayment information.
Overview	If Other VM sends prepayment information, System synchronizes prepayment information.
Туре	Evident
Cross Reference	Function R 4.2
Pre-requisites	N/A
Typical Courses of Events	(OVM) Other VM (S) System (OVM) sends prepayment information. (S) synchronizes stock information[E1].
Alternative Courses of Events	N/A
Exceptional Courses of Events	[E1] If stock is unavailable, makes the prepayment information invalid.



Use Case	10. Go Main
Actor	User
Purpose	Return to the main page when User want to go to the main page.
Overview	After the User click the home button, System displays main page
Туре	Evident
Cross Reference	Function R 6.1
Pre-requisites	User must know the location of the back button on display screen.
Typical Courses of Events	 (U) User (S) System (U) Click back button. Back button is available on every windows. (S) go back to main page(Window1).
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A



Use Case	11. Confirm Verification
Actor	User
Purpose	give out items legitimately and successfully
Overview	User inputs the Verification and System checks it if it is correct or not.
Туре	Evident
Cross Reference	Function R 4.4
Pre-requisites	Function R 4.1 and 4.3 need to be executed first
Typical Courses of Events	 (U) User (S) System (U) presses C in Window1 and types the verification and presses C in Windows6. (S) verifies that it is the same as in the local storage [E1]. (S) If verification is successful, removes the verification code and item comes out[E2].
Alternative Courses of Events	N/A
Exceptional Courses of Events	 [E1] If the User enters an incorrect verification, System displays to the user "Your verification is incorrect, please enter again", and returns to Window6. [E2] If the User enters a correct verification but prepayment is impossible due to lack of stock, System cancels it and returns to Window1.



Use Case	12. Administrator Login
Actor	Administrator
Purpose	let administrator to login to the system.
Overview	Administrator may choose to login to the system by clicking a login button on the screen.
Туре	Evident
Cross Reference	Function R 5.1
Pre-requisites	 Administrator must know the location of the login button. Administrator must have login password to access the system.
Typical Courses of Events	 (A) Administrator (S) System (A) presses B in Window1 to login. (S) displays Windows7. (A) types password and presses C in Window7. [E1] (S) If the login password match with the administrator's login password in the local storage, Window8 will be displayed.
Alternative Courses of Events	N/A
Exceptional Courses of Events	[E1] If the login password does not match with the administrator's login password in the local storage, System sends an error message. The process is repeated three times. if login failed on the third time, System goes to Window1.

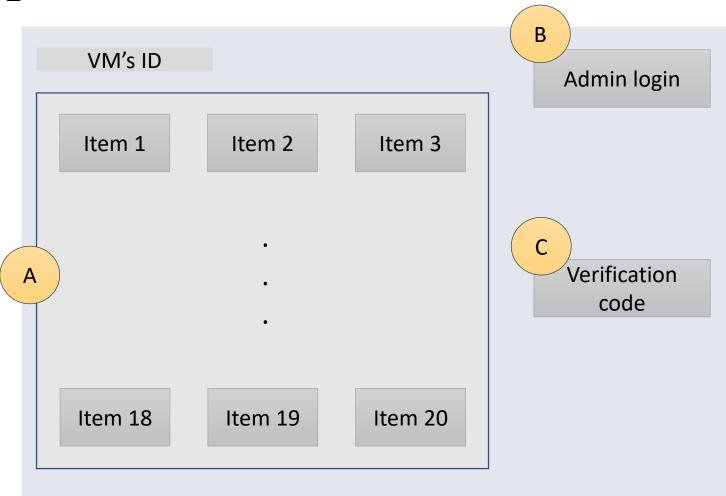


Use Case	13. Administrator Logout
Actor	Administrator
Purpose	let administrator to logout of the system.
Overview	Administrator choose to logout of the system by clicking a logout button on the screen.
Туре	Evident
Cross Reference	Function R 5.2 Use case 1. Main Menu display
Pre-requisites	 Function R 5.1 needs to be executed first. Administrator must know the location of the logout button.
Typical Courses of Events	 (A) Administrator (S) System (A) presses button C in Window8 to logout. (S) display Window1.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

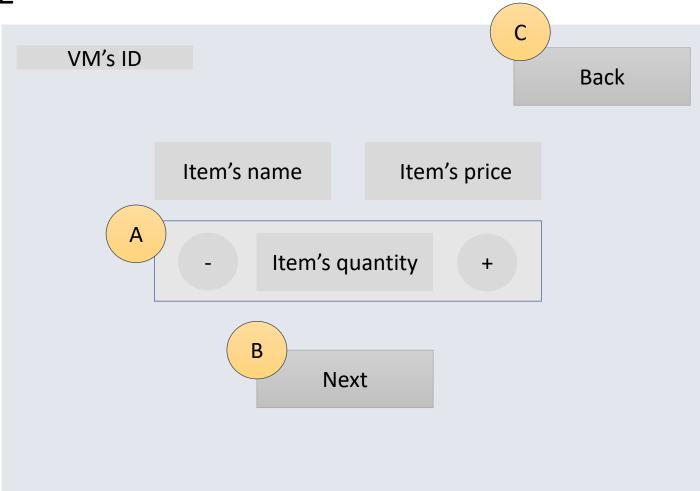


Use Case	14. Change Stock Quantity
Actor	Administrator
Purpose	change the stock quantity sold in the VM.
Overview	Administrator can add any available stock to the VM or remove any expired stock from the VM.
Туре	Evident
Cross Reference	Function R 5.3
Pre-requisites	 Function 5.1 needs to be executed first. Administrator must login successfully into System to execute this use case.
Typical Courses of Events	 (A) Administrator (S) System (S) displays Window8. (A) changes the stock quantity in the VM by updating item's quantity list on Window8-A. (A) presses button B to confirm changes. (S) updates the change to the local storage of System.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A









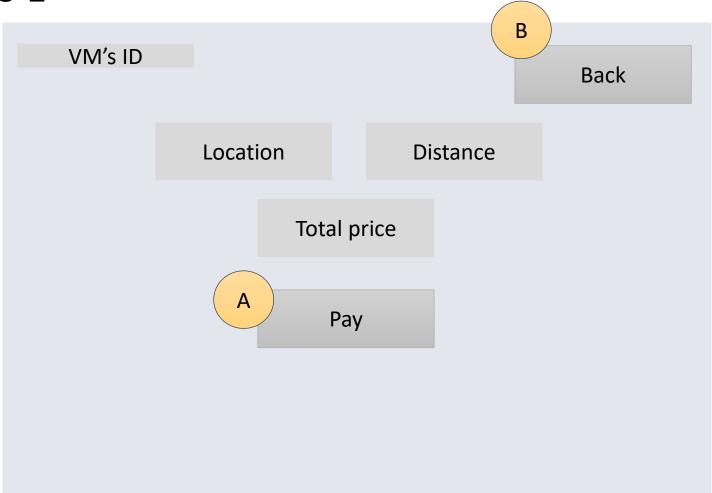


• Window-3-1

		В
VM's ID		Back
	Total price	
	A Pay	



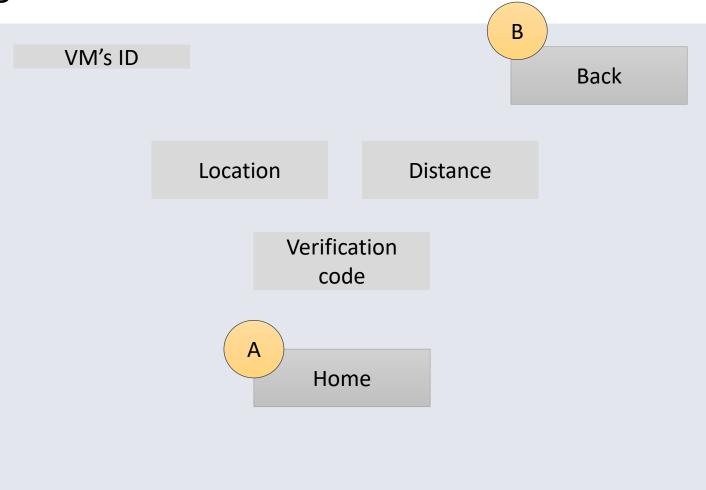
• Window-3-2



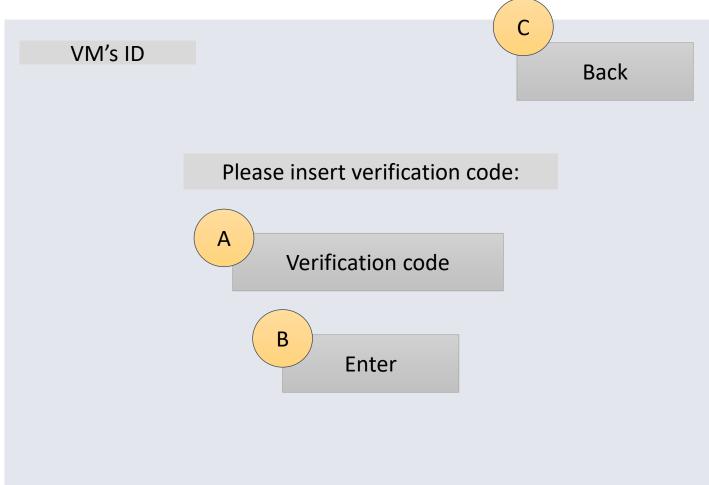


	В
	Back
Please insert card's info	
lime runout display (60 sec)	
A Enter	
	Please insert card's info Time runout display (60 sec)

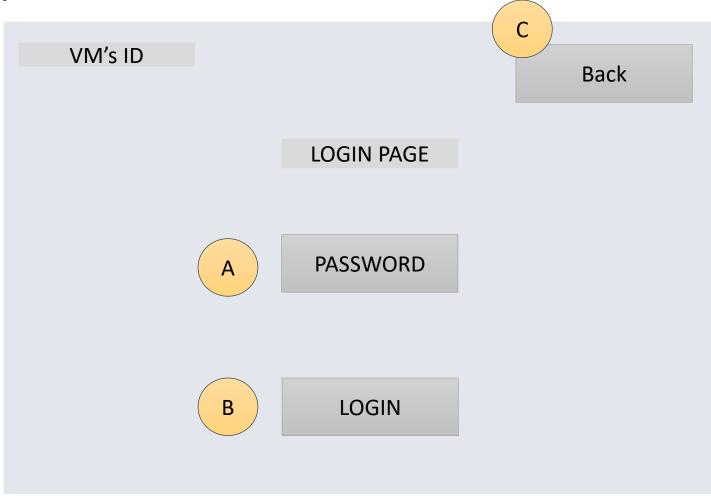




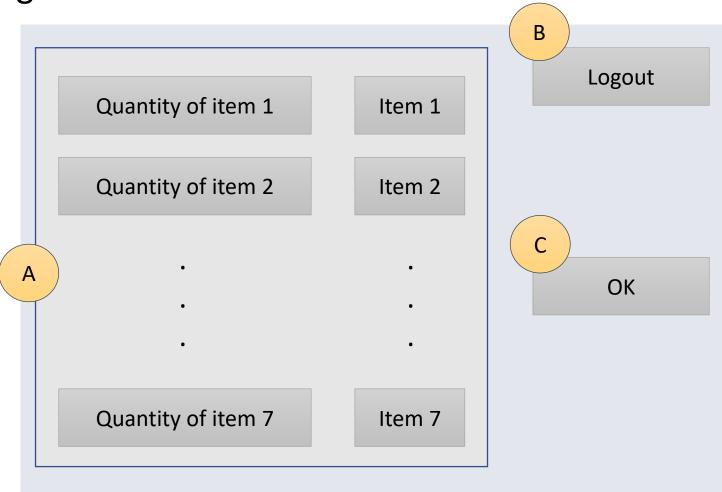




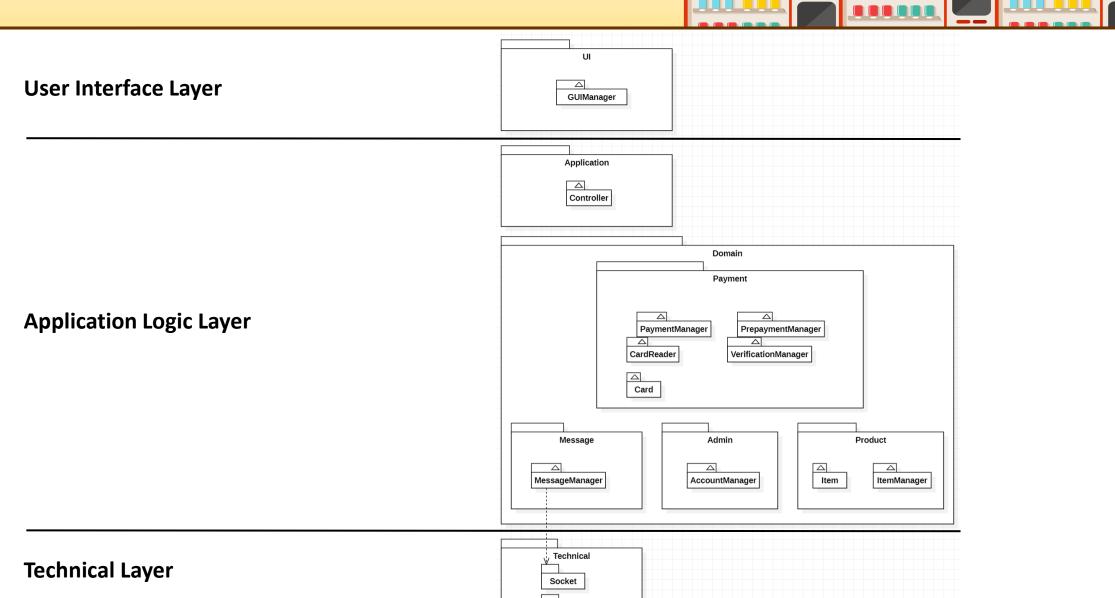






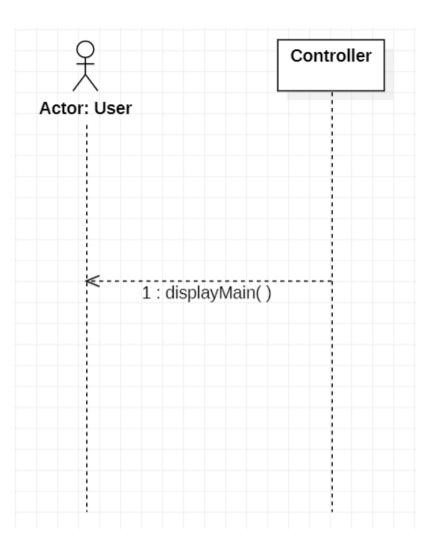


2143 REFINE SYSTEM ARCHITECTURE



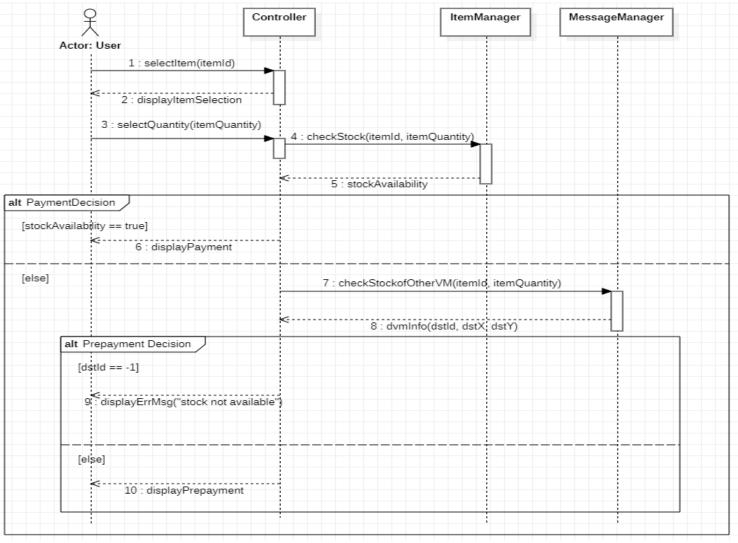
JVM

• Use case 1. Main Menu Display





• Use case 2. Item Selection



 \cap



• Use case 3. Payment

Ž.	Controller	CardReader	PaymentManager	ItemManager
Actor: User				
<				
p (1, 3)				
isValidCart false]				
2 : inputCardInfo(cardInfo)	3 : checkCard	IValidity(cardInfo)		
	<4 : ca	urdValidity		
cardValidity == false]				
5 : displayErrorMsg("Invalid Card")				
ese]				
	loop (1, 3)			
	[pamentSuccess == fa			
		6 : Payment(totalPrice, cardInfo)	► :	
	<	7 : paymentSuccess		
alt				
[paymentSuccess == false]				
8 : displayErrorMsg("Payment Error")				
[else]		9 : checkStock(iteml	d, itemQuantity)	
	<	10 : stockAva	ailability	
				Li li
alt				
[stockAvailability == false]	10	ment/tetelDrine_conducte)		
11 : displayErrorMsg("No item stock")	12 : canciePa	/ment(totalPrice, cardInfo)		
		13 : UpdateStockInfo(item	nld, itemQuantity)	
[else]	item Queentin d			
14 : displayPaymentMsg("Payment Complete", itemId,	itemQuantity)			
15 : getitem				
<pre></pre>				

CardReader

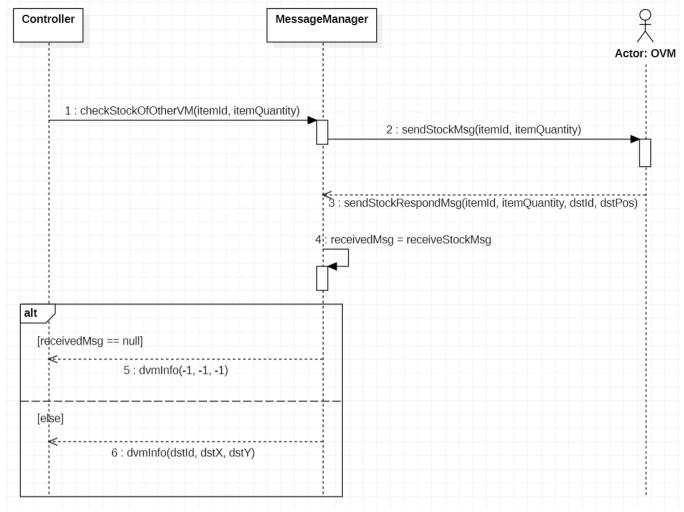
Controller

PaymentManager

ItemManager

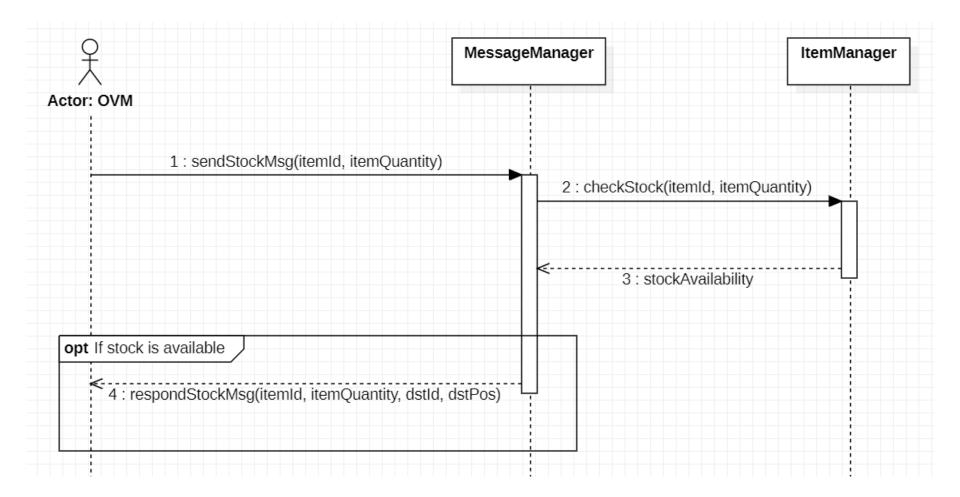


• Use case 4. Request Stock of Other VM



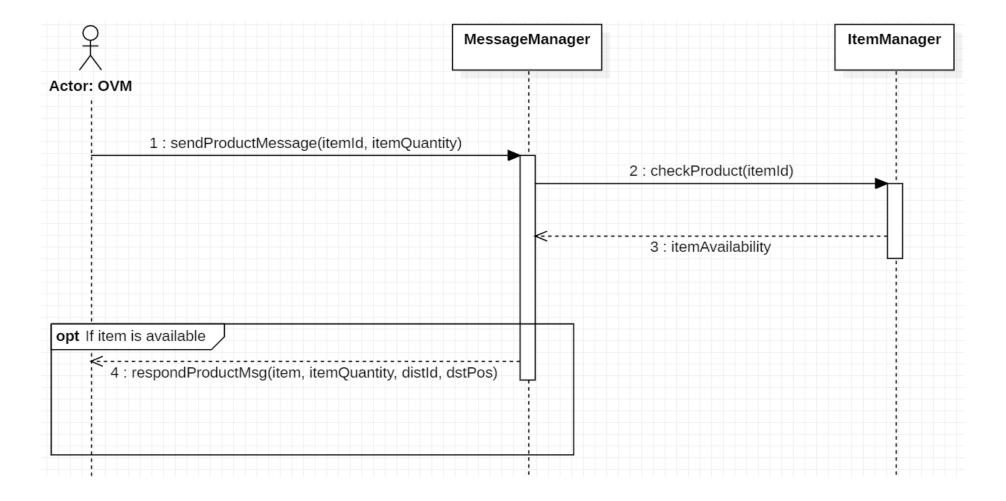


• Use case 5. Provide Stock Information of Other VM



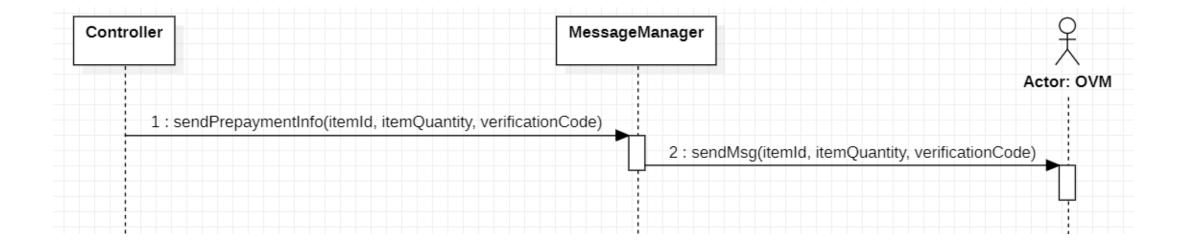


• Use case 6. Provide Product Information of Other VM





• Use case 7. Provide Prepayment Information to the Other VM





• Use case 8. Prepayment

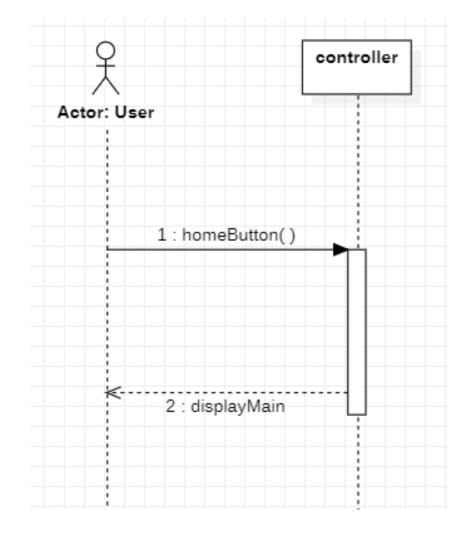
4		Controller	CardReader	PaymentManager	VerificationManager	MessageManag
\wedge						L
Actor: User	1 : displayPrepayment					
op (1, 3)						
[isValidCard == false]	2 : inputCardInfo(cardInfo)	3 : checkCard\	/alidity(cardInfo)			
		<	/alidCard			
		4:150				
: /						
_						
[isValidCard == false] ≪						
	5 : displayErrorMsg("Invalid Card")					
[else]		pop (1, 3)				
	<u> </u>					
			false] 6 : Payment(totalPrice, cardInfo)	⊳ , ¦		
		<	7 : paymentSuccess			
alt /						
[paymentSuccess == false]						
<	8 : displayErrorMsg("Payment Error")					
[else]			9 : RequestVerification	Code	10 : create	/erificationCbde
		<	11 : verificationCode		······································	
			12: sendPrepaymentInfo(i	emId, itemQuantity, verificationC	code)	•
13 : displayPrepaymentMsg("P	ayment Complete", itemId, itemQuantity, dstId, dstPos, veri	ficationCode)				Ļ
≪	14 : displayMain					



• Use case 9. Synchronize Information of Prepayment

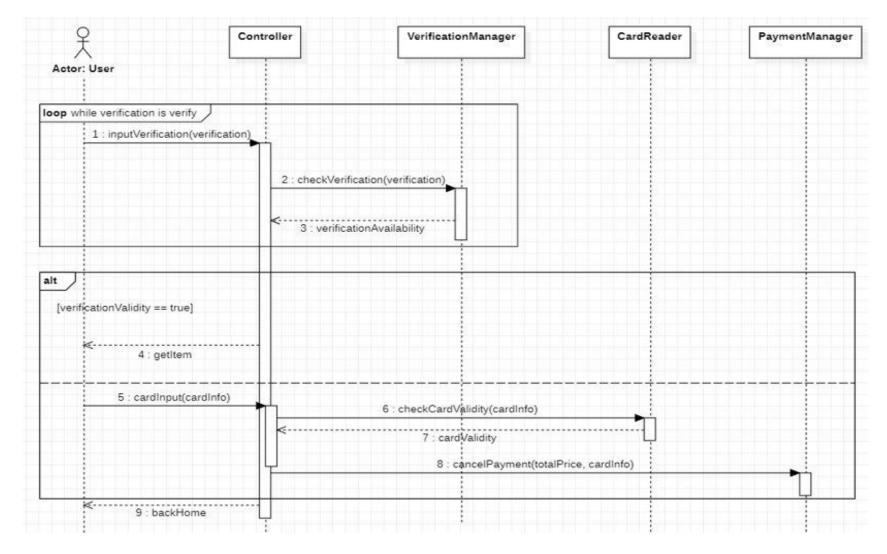
Ŷ	MessageManager	ItemManager	VerificationManager
Actor: OVM			
1 : sendPrepaymentMsg(itemId, ItemQuantity, verifica			
	2 : synchronize(iten	nld, itemQuantity)	
		3 : saveVerification(itemId, itemQu	uantity, verification, verificationvalidity)
		Ļ	
·····			

• Use case 10. Go Main

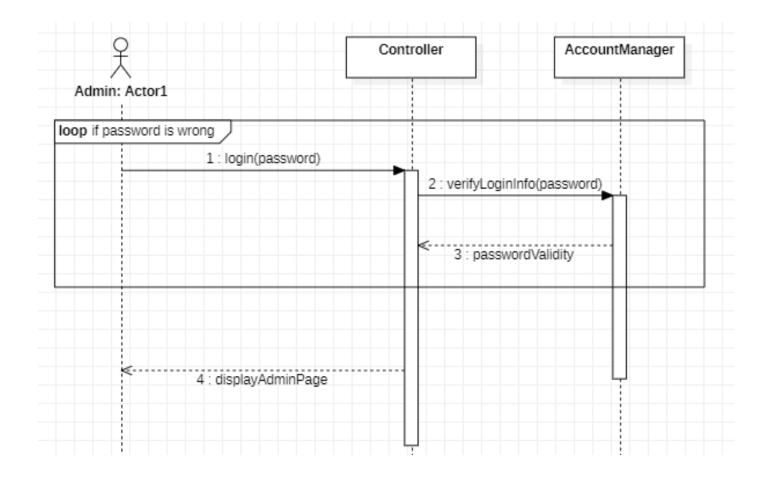




• Use case 11. Confirm Verification

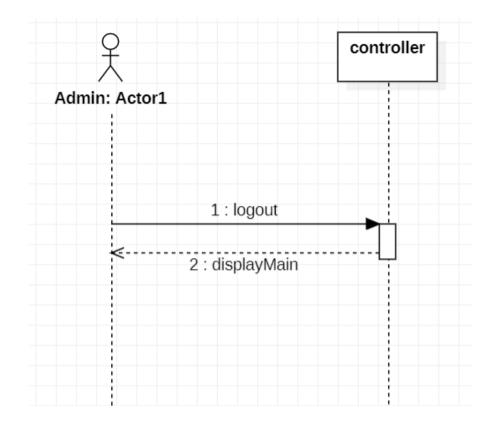


• Use case 12. Administrator Login



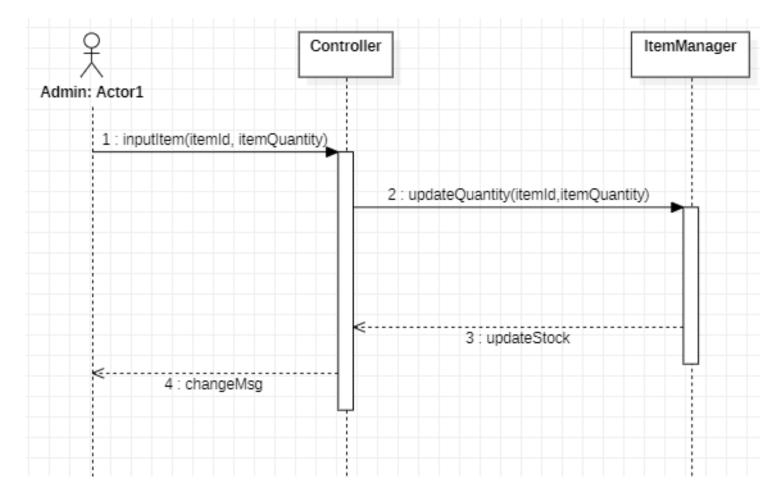


• Use case 13. Administrator Logout

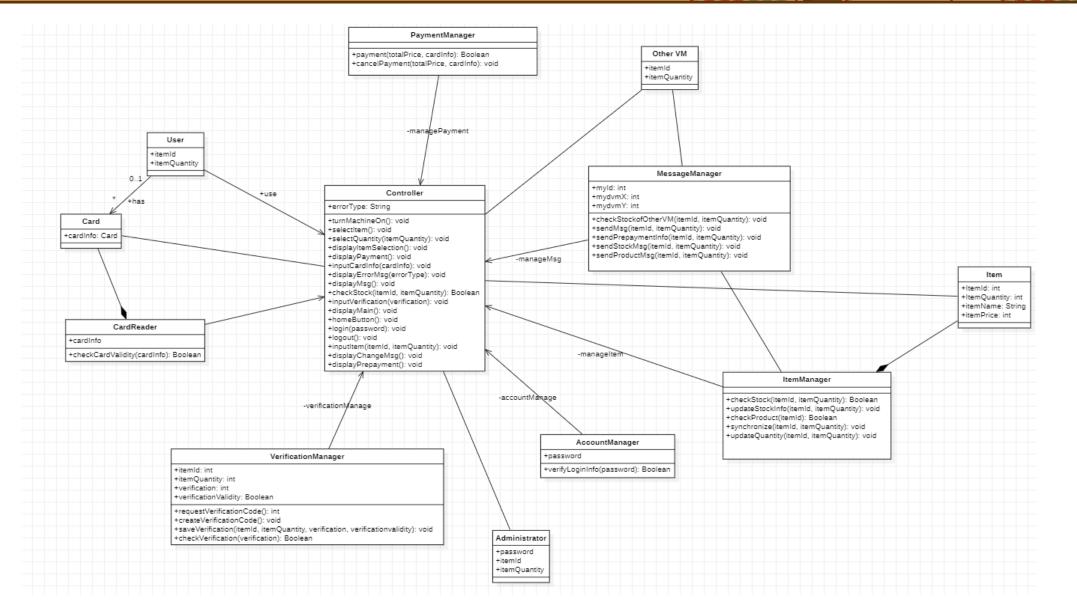




• Use case 14. Change Stock Quantity



2145 DEFINE DESIGN CLASS DIAGRAMS



			METHOD	CLASS
TRACEABILITY TABLE		turnMach	chineOn():void	
		selectiter	em(itemId:int, itemQuantity:int, dvmInfo:int[]):String	
		payment	t(itemId:int, itemQuantity:int, cardNum:String, cardPwd:int): String	
		prepayme	nent(itemId:int, itemQuantity:int, cardNum:String, cardPwd:int, dstId:int): String	
			untManager(): AccountManager	
			Manager(): MessageManager	Controller
			Manager():ItemManager	
			wanage ()endbaade	
			Reader():CardReader	
			nentManager():PaymentManager	
			icationManager():VerificationManager	
			mId:int, itemName:String, itemPrice:int, itemQuantity:int, onSale:boolean)	
		getitemid		
		setitemid	d(itemId:int):void	
		getitemQ	Quantity():int	
		setitemQ	Quantity(itemQuantity:int):void	10.00
			Name():String	item
0	peration		Name(itemName:String):void	
		getitemP		
1	. turnMachineOn()		Price(itemPrice:int):void	
			ile():boolean	
/ 🗯 2	. displayMain()			
	. selectItem(itemId)		ock(item/d:int, itemQuantity:int):boolean	
	. selection(itemito)		tockInfo(itemId:int.itemQuantity:int):void	
Test Case System Function Use Case 4	. displayPayment()		Quantity(itemId:int, itemQuantity:int):void	
			oduct(itemId:int):boolean	
1 Main Menu Display	. inputCardInfo(cardInfo)	// // ≰synchron	nize(itemId:int, itemQuantity:int, verificationCode:String):void	ItemManager
1. Main Menu Display		/// /// /// verificatio	ionValidity:boolean	
R 1.2 Item Selection	. displayErrorMsg()	getitemLi	List:Item[]	
	. sendMsg(itemId, itemQuantity)	l // //_loaditemi	nList():void	
3. Payment	. senumisg(iterniu, iternouantity)	showiten		
3. Payment R 3.1 Request Stock of	sendMsg(itemId, itemQuantity, dstld,	run():void		
Other VM 4. Request Stock of d	stPos)		eiver(itemManager:ItemManager)	
4. Message Request R 3.2 Provide Stock Other VM		msginfo:		
Information to Other VM 5. Provide Stock	. sendStockMsg(itemId, itemQuantity)		g(dstld:int, msg:Message):void	
	0 remendStackMag			
	0. respondStockMsg		paymentInfo(itemId:int, itemQuantity:int, dstid:int, verificationCode:String):void	MessageManager
	1. sendProductMsg(itemId, itemQuantity)		ckMsg(itemId:int, itemQuantity:int, dstId:int):void	
6. Prepayment			ductMsg(itemId:int, itemQuantity:int, dstId:int):void	
A vitovide repayment	2. respondProductMsg		dstld:String, itemId:int, itemQuantity:int, msgType:String, authCode:String):Message	
7. Synchronize Information R 4.1 Prepayment Information to Other VM	3. sendMsg(itemId, itemQuantity,		Des(msgDes:Message.MessageDescription, itemId:int, itemQuantity:int, authCode:String):void	
P 4 2 Synghroniza	erification)		Msg(msg:message):int[]	
8. Go Main	ermes dony		rdNum:String, cardPwd:int, money:int)	
9. Synchronize	4. displayPrepayment()		rdNum:String, cardPwd:int)	
9 Manage		→ getCardN		Card
R 4.3 Create Verification	5. sendPrepaymentMsg(itemId,	getCardP		
R 4.4 Confirm Verification	emQuantity, verificationCode)	getMone		
10 Administrator Login	6. homeButton()	setMone	ey(money:int):void	
R 5.1 Administrator Login	s. nemeborron()	checkCare	rdValidity(cardNum:String, cardPwd:int):boolean	
11. Administrator Logout R 5.2 Administrator	7. goHome	getCardin	Info(cardNum:String):Card	CardManager
12. Change Stock		encodeCa	CardNum(inputCardNum:String):String	Cardmanager
12. Change Stock Quantity R 5.3 Change Stock 12. Administrator Login	8. inputVerification(verification)	loadCard	dList():void	
	9. login()		t(cardReader:CardReader, totalPrice:int, cardNum:String):boolean	
	5.10811()		ayment(cardReader:CardReader, totalPrice:int, cardNum:String):void	PaymentManager
	0. displayAdminPage()	Verificatio	tion(itemId:int, itemQuantity:int, verificationCode:String, verificationValidity:boolean)	
Quantity			icationValidity:boolean	
	1. logout()		Id(): int	Verification
	2. inputItem(itemId, itemQuantity)		Quantity():int	
			icationCode:String	
2	3. changeMsg		tVerification():void	
			erificationCode():String	
			ification(itemId:int, itemQuantity:int, verificationCode:String, verificationValidity:boolean):void	
			rification(verificationCode:String):Verification	VerificationManager
			ode(code:String):boolean	
			Verification(verificationCode:String):void	
		verifyLog	ginInfo(password:String):boolean	AccountManager

CLASS

METHOD

QnA

