

# *Distributed vending machine*

학번	이름
201511238	허준호
201911147	강속영
201711320	오준엽
201511218	이지영
201710397	이지훈

## Design Real Use Cases

2041.	Use case	1. Start Order
	Actor	User
	Purpose	자판기 사용을 시작한다.
2042.	Overview	자판기의 상품을 고르기 위해 카드를 투입한다.
	Type	evident
	Cross Reference	System Functions : 1.1, 1.2 Use Case : "Check Card Input"
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다. 전원이 켜져 있어야 한다.
2044.	Typical Courses of Events	(AU) : Actor User, (S) : System 1.(AU) 카드리더기에 카드를 투입한다. 2.(S) Window-2(상품 선택 화면)을 출력한다. 3.(S) "Input choice"로 넘어간다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	all line-a. (AU) 카드리더기에서 카드를 제거한다. all line-b. (S) 모든 작업을 중단한다 all line-c. (S) Window-24(카드 제거 안내 화면)을 출력한다 all line-d. (S) Window-1(대기 화면)을 출력한다.
2047.		

# Design Real Use Cases

Use case	2.Input Choice
2041. Actor	User
Purpose	사용자가 상품을 선택한다.
2042. Overview	사용자가 선택한 상품의 재고가 현재 자판기에 존재하는지 확인한다.
Type	evident
Cross Reference	System Functions : 2.1, 2.2, 3.2 Use Case : "Check chosen item Stock", "Payment"
2043. Pre-Requisites	자판기의 전원이 켜져 있어야 한다. 결제 가능한 카드가 투입되어 있어야 한다.
2044. Typical Courses of Events	(AU) : Actor User, (S) : System 1.(AU) 사용자가 Window-2(상품 선택 화면)에서 원하는 상품의 버튼을 선택한다. 2.(S) 현재 자판기에서 재고를 확인한다. 3.(S) 재고가 존재한다면 "Payment"로 넘어간다.
Alternative Courses of Events	N/A
2045. Exceptional Courses of Events	3-a. (S) 현재 자판기에 상품이 존재하지 않는다. 3-b. (S) "Check chosen item Stock"으로 넘어간다.  all line-a. (AU) 카드리더기에서 카드를 제거한다. all line-b. (S) 모든 작업을 중단한다. all line-c. (S) Window-24(카드 제거 안내 화면)을 출력한다. all line-d. (S) Window-1(대기화면)을 출력한다.
2047.	

## Design Real Use Cases

2041.	Use case	3. Check Chosen Item Stock
	Actor	System
	Purpose	사용자가 선택한 상품의 재고가 다른 자판기에 존재하는지 확인한다.
2042.	Overview	다른 모든 자판기에 사용자가 선택한 상품의 재고가 있는지 확인을 요청하고 정보를 받는다.
	Type	evident
	Cross Reference	System Functions : 2.2, 2.3, 2.4 Use Case : "Answer Stock Info", "Determine location"
2043.	Pre-Requisites	현재 자판기에 상품 재고가 존재하지 않는다.
	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs (S) : System 1.(S) AD의 location, 상품의 quantity를 요청한다. 2. (AD) "Answer Stock Info"을 통해 위치, 상품 재고 정보를 받는다.
2044.	Alternative Courses of Events	N/A
	Exceptional Courses of Events	2-a. 재고가 없다 2-b. Window-23(모든 재고 소진 화면)을 출력한다.  all line-a. (AU) 카드리더기에서 카드를 제거한다. all line-b. (S) 모든 작업을 중단한다. all line-c. (S) Window-24(카드 제거 안내 화면)을 출력한다. all line-d. (S) Window-1(대기화면)을 출력한다.
2047.		

## Design Real Use Cases

2041.	Use case	4. Answer Chosen Item Stock
	Actor	System
	Purpose	자판기의 재고와 위치 정보를 알려준다.
2042.	Overview	전달받은 상품의 재고를 확인 후 해당 정보를 자판기 위치와 함께 요청한 자판기에 보내준다.
	Type	hidden
	Cross Reference	System Functions : //2.2,// 2.3 //Use Case : "Check chosen item Stock"//
2043.	Pre-Requisites	(AD)으로 부터 재고 확인 요청이 있어야한다.
2044.	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVM, (S) : System 1.(S) "Check chosen item Stock "을 통해 요청된 상품의 재고를 확인한다. 2.(S) 확인한 재고 정보를 자신의 위치정보와 함께 (AD)로 전달한다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	N/A
2047.		

## Design Real Use Cases

2041.

Use case	5. Determine location
Actor	System
Purpose	자판기들 사이의 거리를 계산한다.
Overview	재고가 존재하는 자판기와 현재 자판기 사이의 거리를 계산한다.
Type	hidden
Cross Reference	System Functions : 2.4, 3.1 Use Case : "Determine Prepayment"
Pre-Requisites	재고가 있는 자판기가 존재한다.
Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs (S) : System 1.(S) 현재 자판기에서 재고가 있는 자판기들의 거리를 계산한다. 2.(S) 안내할 자판기를 결정한다. 3.(S) "Determine Prepayment"로 넘어간다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

2042.

2043.

2044.

2045.

2047.

# Design Real Use Cases

2041.	Use case	6. Determine Prepayment
	Actor	User
	Purpose	선결제를 한다.
2042.	Overview	사용자가 선결제 여부를 선택하고, 선택에 따라 결제를 요청한다.
	Type	evident
	Cross Reference	System Functions : 2.4, 3.1, 3.2 Use Case : "Determine location", "Payment"
2043.	Pre-Requisites	재고가 있는 가장 가까운 자판기의 위치 정보와 재고 정보가 현재 자판기에 존재해야 한다.
2044.	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs 1.(S) "Determine location"에서 온 재고정보와 위치정보를 Window-4에 출력한다. 2.(AU) 선결제 버튼을 누른다. 3.(S) "Payment"로 넘어간다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	2-a. (AU) 취소 버튼을 누른다. 2-b. (S) Window-2(상품 선택 화면)을 출력한다.  all line-a. (AU) 카드리더기에서 카드를 제거한다. all line-b. (S) 모든 작업을 중단한다. all line-c. (S) Window-24(카드 제거 안내 화면)을 출력한다. all line-d. (S) Window-1(대기화면)을 출력한다.
2047.		

# Design Real Use Cases

	7. Payment
2041. Actor	System
Purpose	결제를 진행한다.
Overview	결제를 진행하고 결제 종류에 따라 다음 행동을 지시한다.
2042. Type	hidden
Cross Reference	System Functions : 3.2, 3.3, 3.5 Use Case : "Create Precode", "Serve Item"
Pre-Requisites	카드가 삽입되어 있어야 한다.
2043. Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs 1.(S) Payment 클래스에서 item클래스의 price를 받아온다. 2.(S) Payment클래스의 prepayment값을 판단해 선결제인지 아닌지 판단한다. 3.(S) prepayment가 false이다. 4.(S) "Serve Item"을 진행한다.
2044. Alternative Courses of Events	N/A
2045. Exceptional Courses of Events	1-a. (S) 결제를 실패한다. 1-b. (S) Window-26(결제 실패 화면)을 알린다.  3-a. (S) prepayment가 true이다. 3-b. (S) "Create Precode"을 진행한다.  1-a. (AU) 카드리더기에서 카드를 제거한다. 1-b. (S) 모든 작업을 중단한다. 1-c. (S) Window-24(카드 제거 안내 화면)을 출력한다. 1-d. (S) Window-1(대기화면)을 출력한다.
2047.	



## Design Real Use Cases

2041.	Use case	8. Create Precode
	Actor	System
	Purpose	선결제 코드를 생성하여 저장한다.
2042.	Overview	선결제 코드를 생성하여 저장한다.
	Type	hidden
	Cross Reference	System Functions : 3.3, 3.4 Use Case : "Complete prepayment"
2043.	Pre-Requisites	카드가 투입되어 있어야 한다. "Payment"가 선행되어야 한다.
2044.	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs (S) : System 1.(S) Precode instance를 생성한다. 2.(S) 생성된 선결제 코드를 저장한다. 3.(S) "Complete prepayment"로 넘어간다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	N/A
2047.		

## Design Real Use Cases

2041.	Use case	9. Complete prepayment
	Actor	System
	Purpose	선결제가 완료되었음을 안내한다.
2042.	Overview	선결제 코드와 선결제 코드가 사용 가능한 자판기의 위치를 사용자에게 안내해 선결제를 완료한다.
	Type	hidden
	Cross Reference	System Functions : 3.3, 3.4 Use Case : "Create Precode"
2043.	Pre-Requisites	재고가 있는 가장 가까운 자판기의 위치 정보와 재고 정보가 현재 자판기에 존재해야 한다.
2044.	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs 1.(S) "Create Precode"에서 생성된 선결제 코드와 사용가능한 자판기 위치를 Window-6(선결제 완료 화면)에 출력한다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	N/A
2047.		

## Design Real Use Cases

2041.	Use case	10. Serve Item
	Actor	System
	Purpose	사용자에게 상품을 배출한다.
2042.	Overview	결제된 상품을 취출구로 배출한다.
	Type	hidden
	Cross Reference	System Functions : 3.5
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다. 상품에 대한 결제가 되어있어야한다. 취출구에 상품을 배출할 공간이 있어야한다.
2044.	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs (S) : System 1.(S) 결제된 상품을 배출한다. 2.(S) Window-8(결제 완료 화면)을 출력한다. 3.(S) Window-2(상품 선택화면)을 출력한다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	N/A
2047.		

## Design Real Use Cases

2041.	Use case	11. Check Precode
	Actor	System
	Purpose	입력된 선결제 코드와 생성된 선결제 코드의 일치 여부를 확인한다.
2042.	Overview	입력된 선결제 코드를 다른 모든 DVM에게 보낸 뒤 일치한다는 답변이 오면 상품을 배출한다.
	Type	evident
	Cross Reference	System Functions : 3.5, 3.6, 3.7, 3.8 Use Case : “Serve Item”, “Answer Precode Info”, “Input Precode”
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs (S) : System 1.//(S) “Input Precode”에서 입력 받은 선결제 코드를 (AD)에게 보낸다. 2.(AD) “Answer Precode Info”를 통해 선결제 코드 일치 여부를 전달한다. 3.(S) 선결제 코드가 일치한다.// 4.(S) 코드 생성시 안내된 DVM이 현재 DVM이다. 5.(S) “Serve Item”을 진행한다.
2044.	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	3-a. (S) 선결제 코드가 불일치한다. 3-b. (S) Window-19(잘못된 입력 화면)을 출력한다.  4-a. (S) 현재 DVM과 선결제시 안내된 DVM이 아니다. 4-b. (S) Window-22(잘못된 DVM 화면)을 출력한다.
2047.		

## Design Real Use Cases

2041.	Use case	12. Answer Precode Info
	Actor	System
	Purpose	선결제 코드의 일치 여부를 확인한다.
2042.	Overview	다른 DVM이 확인을 요청받은 선결제 코드가 발급된 적이 있는 선결제 코드인지 응답한다.
	Type	hidden
	Cross Reference	System Functions : 3.6, 3.7 Use Case : "Check Precode"
2043.	Pre-Requisites	"Check Precode"을 통해 요청받아야 한다.
	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs (S) : System 1.(S) "Check Precode"을 통해 요청된 선결제 코드를 확인한다. 2.요청된 코드와 발급된 코드가 일치한다. 3.(S) 선결제 코드가 일치함을 알린다.
2044.	Alternative Courses of Events	N/A
	Exceptional Courses of Events	2.a (S) 일치하지 않으면, 요청된 선결제 코드가 발급된 적이 없다는 것을 알린다.
2045.		
2047.		

## Design Real Use Cases

2041.	Use case	13. Input Precode
	Actor	User
	Purpose	선결제 코드를 입력한다.
2042.	Overview	선결제 코드 입력을 선택 후 코드를 입력한다.
	Type	evident
	Cross Reference	System Functions : 3.8
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
2044.	Typical Courses of Events	(AU) : Actor User, (AD) : Actor other DVMs (S) : System 1.(AU) Window-1(대기 화면)에서 선결제 입력 버튼을 누른다 2.(S) Window-3(선결제 입력 화면)을 출력한다. 3.(AU) 선결제 코드를 입력한다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	3-a. (AU) Window-3(선결제 입력 화면)의 취소버튼을 누른다. 3-b. (S) Window-1(대기 화면)을 출력한다.  3-a. (S) 10초 내에 응답이 없을 경우 Window-1(대기 화면)을 출력한다.
2047.		

## Design Real Use Cases

2041.	Use case	14. Enter Administrator Mode
	Actor	Manager
	Purpose	관리자모드로 진입한다.
2042.	Overview	관리자 인증 수단을 통해서 관리자모드로 진입한다.
	Type	evident
	Cross Reference	System Functions : 4.1
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
	Typical Courses of Events	(AM) : Actor Manager, (S) : System 1.(AM) 인증 수단을 사용하여 (S)에게 인증을 요청한다. 2.(S) 인증에 성공한다. 3.(S) Window-12(관리자모드)를 출력한다.
2044.	Alternative Courses of Events	N/A
	Exceptional Courses of Events	2-a. 인증에 실패한다. 2-b. Window-19(잘못된 입력 화면)을 출력한다.
2045.		
2047.		

## Design Real Use Cases

2041.	Use case	15. Add Itemlist
	Actor	Manager
	Purpose	아이템 목록을 추가한다.
2042.	Overview	아이템 목록을 추가한 후에 갱신한다.
	Type	evident
	Cross Reference	System Functions : 4.2
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
	Typical Courses of Events	(AM) : Actor Manager, (S) : System 1.(AM) Window-12(관리자 모드 화면)의 추가버튼을 누른다. 2.(S) Window-13(상품 목록 추가 화면)을 출력한다. 3.(AM) Window-13에서 Item의 name을 변경한다. 4.(S) Item 목록을 갱신한다.
2044.	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	1-a. 상품 목록 변경의 범위가 지정된 범위(20 종류)를 벗어난다 1-b. 추가 버튼이 비활성화 된다.
2047.		



## Design Real Use Cases

2041.	Use case	16. Remove Itemlist
	Actor	Manager
	Purpose	아이템 목록을 삭제한다.
2042.	Overview	아이템 목록을 삭제한 후에 갱신한다.
	Type	evident
	Cross Reference	System Functions : 4.2
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
2044.	Typical Courses of Events	(AM) : Actor Manager, (S) : System 1.(AM) Window-12(관리자 모드 화면)의 삭제버튼을 누른다. 2.(S) Window-14(상품 목록 삭제 화면)을 출력한다. 3.(AM) 삭제할 상품을 선택한다. 4.(S) Window-15(삭제 확인 화면)을 출력한다. 5.(AM) 확인 버튼을 누른다. 6.(S) Item 목록을 갱신한다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	1-a. 상품 목록 변경의 범위가 지정된 범위(20 종류)를 벗어난다 1-b. 추가 버튼이 비활성화 된다.
2047.		

## Design Real Use Cases

2041.	Use case	17. Change Stock
	Actor	Manager
	Purpose	상품의 재고를 변경한다.
2042.	Overview	상품 재고를 변경한 것을 시스템에 반영시킨다.
	Type	evident
	Cross Reference	System Functions : 4.3
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
2044.	Typical Courses of Events	(AM) : Actor Manager, (S) : System 1.(AM) Window-12(관리자 모드)에서 상품버튼을 누른다. 2.(S) 해당 상품의 Window-16(상품 가격, 재고변경)을 출력한다. 3.(AM) Window-16(상품 가격, 재고변경)에서 Item의 Payment Lineitem의 quantity를 변경한다. 4.(AM) Window-16(상품 가격, 재고변경)에서 확인 버튼을 누른다. 5.(S) 재고 변경을 감지한다. 6.(S) 반영한다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	4-a. (AM) Window-16(상품 가격, 재고변경)에서 취소 버튼을 누른다. 4-b. (S) Window-12(관리자 모드)를 출력한다.
2047.		

## Design Real Use Cases

2041.	Use case	18. Change Item price
	Actor	Manager
	Purpose	상품의 가격을 변경한다.
2042.	Overview	상품의 가격을 변경한 것을 시스템에 반영시킨다.
	Type	evident
	Cross Reference	System Functions : 4.5
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
2044.	Typical Courses of Events	(AM) : Actor Manager, (S) : System 1.(AM) Window-12(관리자 모드)에서 상품버튼을 누른다. 2.(S) 해당 상품의 Window-16(상품 가격, 재고변경)을 출력한다. 3.(AM) Window-16(상품 가격, 재고변경)에서 Item 의 price를 변경한다. 4.(AM) Window-16(상품 가격, 재고변경)에서 확인버튼을 누른다. 5.(S) 변경이 설정된 범위 안에 있다. 6.(S) price를 갱신한다.
	Alternative Courses of Events	N/A
2045.	Exceptional Courses of Events	5-a. 설정한 가격이 지정된 범위를 벗어난다. 5-b. Window-19(잘못된 입력)을 3초간 출력한다.  4-a. (AM) Window-16(상품 가격, 재고변경)에서 취소 버튼을 누른다. 4-b. (S) Window-12(관리자 모드)를 출력한다.
2047.		

## Design Real Use Cases

2041.	Use case	19. Check Sales History
	Actor	Manager
	Purpose	판매내역을 확인한다.
2042.	Overview	상품의 판매 수량, 총 판매 금액을 출력한다.
	Type	evident
	Cross Reference	System Functions : 4.6
2043.	Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
	Typical Courses of Events	(AM) : Actor Manager, (S) : System 1.(AM) Window-12(관리자 모드 화면)에서 판매 내역 확인 버튼을 누른다. 2.(S) Window-17(판매 내역 확인) 화면을 출력한다.
2044.	Alternative Courses of Events	N/A
	Exceptional Courses of Events	N/A
2045.		
2047.		

## Design Real Use Cases

Use case	20. Turn off System
2041. Actor	Manager
Purpose	자판기의 전원을 끈다.
2042. Overview	관리자가 자판기를 비활성화 시킨다.
Type	evident
Cross Reference	System Functions : 4.6
2043. Pre-Requisites	자판기의 전원이 켜져 있어야 한다.
2044. Typical Courses of Events	(AM) : Actor Manager, (S) : System 1.(AM) Window-12(관리자 모드 화면)에서 전원 끄기 버튼을 누른다. 2.(S) 현재 자판기의 정보를 저장한다. 3.(S) 시스템을 종료한다.
Alternative Courses of Events	N/A
2045. Exceptional Courses of Events	N/A
2047.	

## Design Real Use Cases

Use case	21. Turn on System
Actor	Manager
Purpose	자판기의 전원을 켜다.
Overview	관리자가 자판기를 활성화 시킨다.
Type	evident
Cross Reference	System Functions : 4.7
Pre-Requisites	자판기의 전원이 꺼져 있어야 한다.
Typical Courses of Events	(AM) : Actor Manager, (S) : System 1.(AM) 기기의 전원 버튼을 누른다. 2.(S) 저장되어 있는 자판기의 정보를 불러온다. 3.(S) Window-1(대기 화면)을 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

2041.

2042. ◀

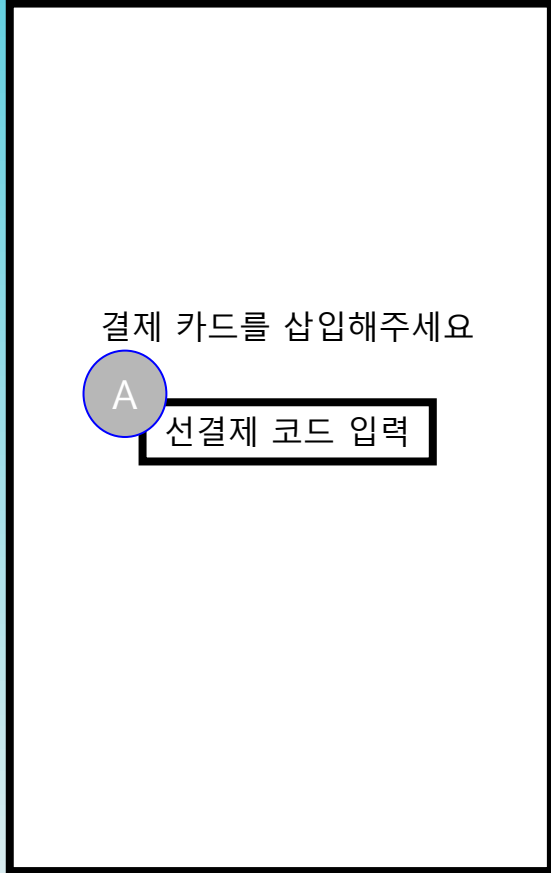
2043.

2044.

2045.

2047.

대기 화면 : Window-1



카드를 입력하면 상품 선택화면(Window-2)으로 넘어간다.  
(에뮬레이터-> 키보드 입력 'c')

A. 누르면 선결제 입력 화면(Window-3)으로 넘어간다.

2041.

2042. ◀

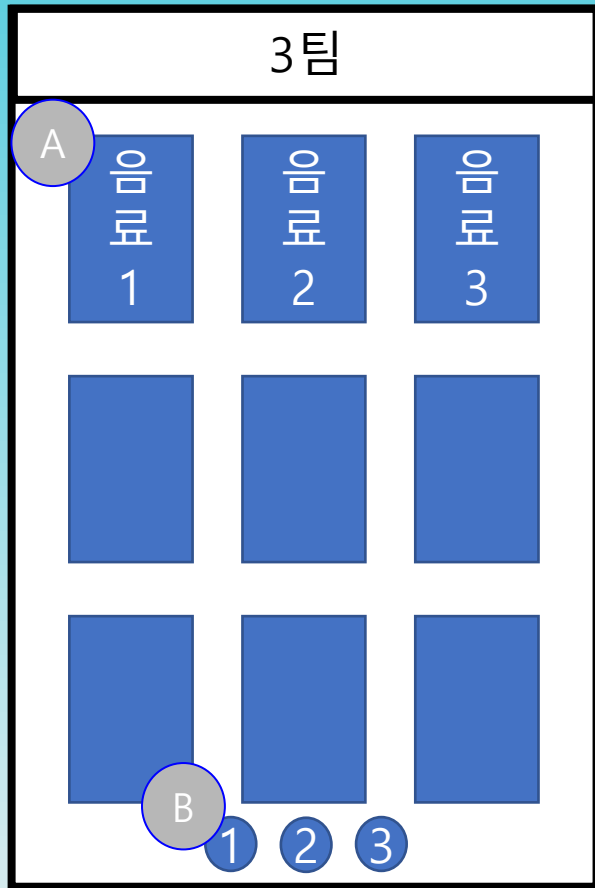
2043.

2044.

2045.

2047.

상품 선택 화면 : Window-2



모든 음료들을 화면에 출력한다.

A. 현 기기에 재고가 있을 시 결제 완료 화면(Window-8)으로 넘어간다.

현 기기에 재고가 없을 시 선결제 선택 화면(Window-4)으로 넘어간다.

모든 기기에 재고가 없을 시 재고가 없음을 알리는 화면(Window-23)으로 넘어간다.

B. 다음 음료 리스트를 보여준다.



2041.

2042. ◀

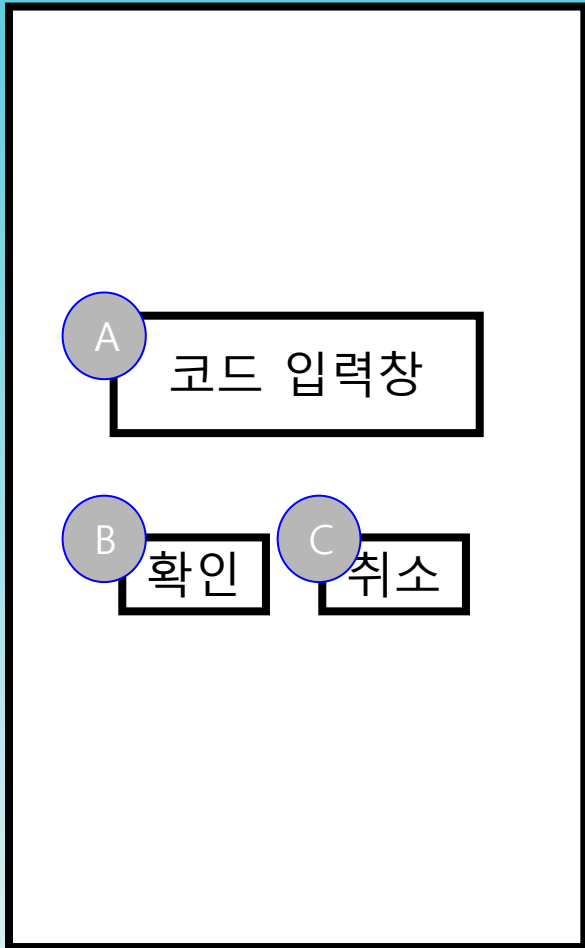
2043.

2044.

2045.

2047.

선결제 입력 화면 : Window-3



- A. 결제한 선결제 코드를 입력하는 창이다. 누르면 숫자패드 (Window-18)가 팝업으로 출력된다.
- B. 코드를 입력 완료 시 결제 완료 화면(Window-8)으로 넘어간다.
- C. 취소하면 대기화면(Window-1)으로 돌아간다.

2041.

2042. ◀

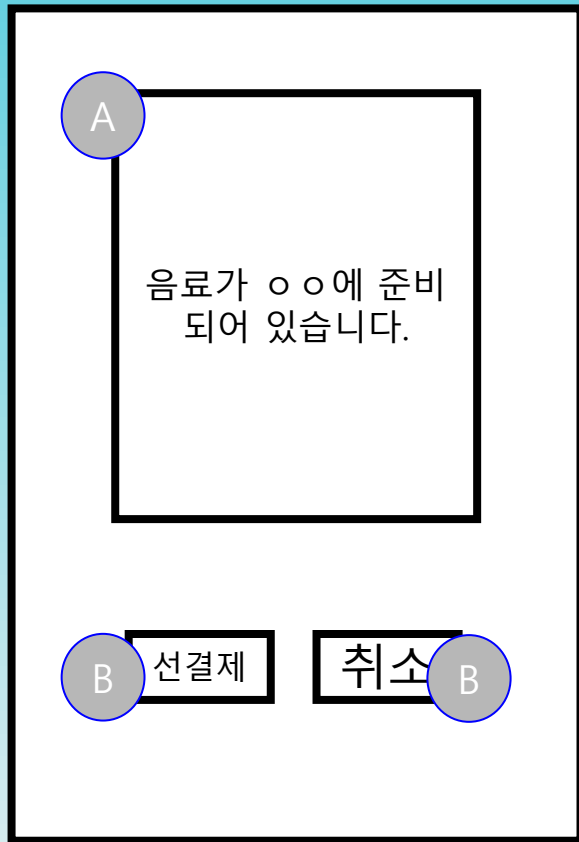
2043.

2044.

2045.

2047.

선결제 선택 화면 : Window-4



선결제를 진행한다.

- A. 재고가 있는 자판기를 안내하는 문구를 출력한다.
- B. 누르면 결제가 진행된다. 결제가 완료되면 해당 화면(Window-6)을 출력한다.
- C. 취소 시 상품 선택 화면(Window-2)이 출력된다.

2041.

2042. ◀

2043.

2044.

2045.

2047.

선결제 완료 화면 : Window-6

자판기 위치

선결제 코드

선결제한 음료가 있는 자판기의 위치와 함께 선결제 코드를 화면에 출력한다.

2041.

2042. ◀

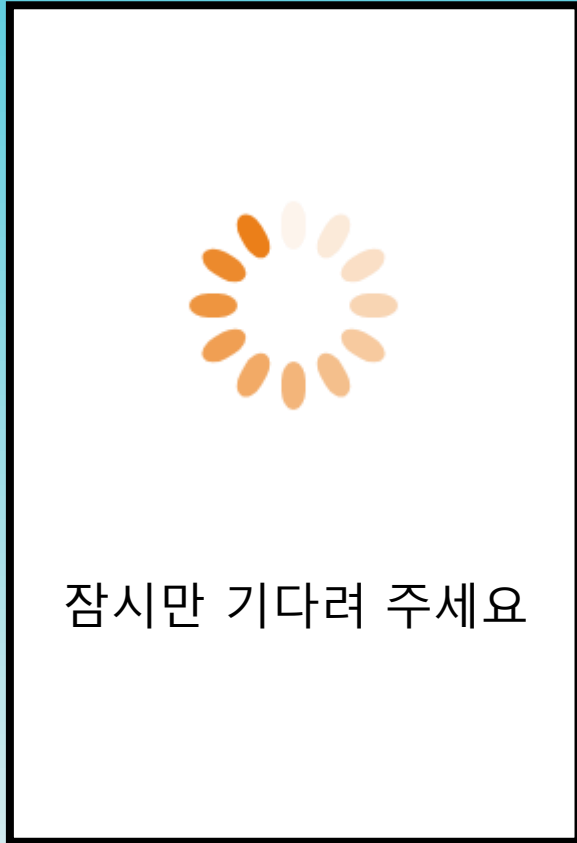
2043.

2044.

2045.

2047.

로딩 화면 : Window-7



결제가 진행중이거나 음료를 찾고 있는 동안 출력되는 화면이다.

2041.

2042. ◀

2043.

2044.

2045.

2047.

결제 완료 화면 : Window-8

선택하신 000 나왔습니다

이용해주셔서 감사합니다

결제가 완료되었을 시 출력되는 화면이다.

2041.

변경 완료 화면 : Window-9, 10, 11

2042. ◀

변경 되었습니다

추가 완료 되었습니다

삭제 완료 되었습니다.

2043.

2044.

2045.

2047.

2041.

2042. ◀

2043.

2044.

2045.

2047.

**관리자 모드 : Window-12**

The diagram shows a window titled '관리자 모드 : Window-12'. At the top is a menu bar with four buttons: 'A' (추가), 'B' (삭제), 'C' (판매 내역 확인), and 'F' (X). Below the menu bar is a grid of product items. The first row contains three items, each with a blue button labeled 'D' and '음료' followed by a number (1, 2, 3). The second and third rows each contain three empty blue squares. At the bottom of the grid, there are three small blue buttons labeled 'E', '1', '2', and '3'.

- A. 상품 추가 화면(Window-13)으로 넘어간다. 실패 시 상품 추가 불가 화면(Window-21)을 출력한다.
- B. 상품 삭제 화면(Window-14)으로 넘어간다.
- C. 판매 내역 확인 화면(Window-17)으로 넘어간다.
- D. 상품 가격/재고 변경 화면(Window-16)으로 넘어간다.
- E. 다음 페이지의 상품 이미지를 출력한다.
- F. 자판기를 종료한다.

2041.

2042. ◀

2043.

2044.

2045.

2047.

관리자 모드 - 상품 목록 추가 : Window-13

The diagram shows a form for adding a product. It contains the following elements:

- A:** A blue button labeled "이미지 첨부" (Image Upload).
- B:** A text input field for "상품명" (Product Name).
- C:** A numeric keypad input field for "상품 가격" (Product Price).
- D:** A button labeled "확인" (Confirm).
- E:** A button labeled "취소" (Cancel).

A: 추가할 상품 이미지를 입력 받는다.

B: 상품명을 입력 받는다.

C: 상품 가격을 넘버패드(Window-18)로 입력 받는다.

D: 관리자 모드 화면(Window-12)으로 이동한다.

E: 관리자 모드 화면(Window-12)으로 이동한다.



2041.

2042.

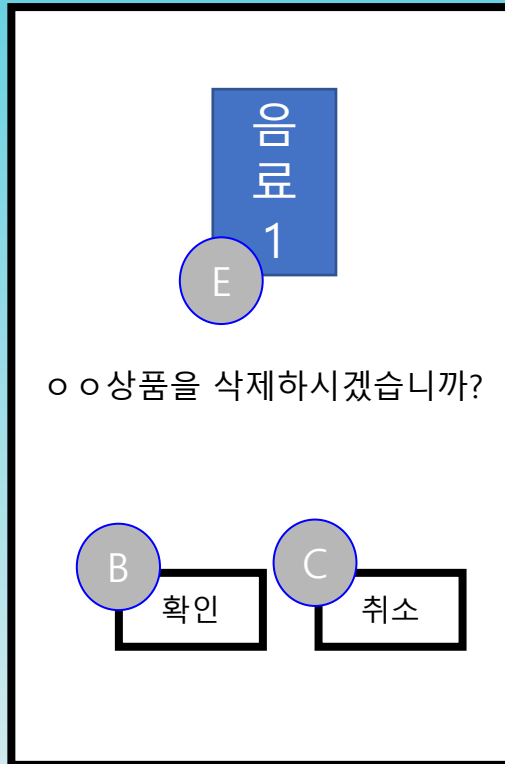
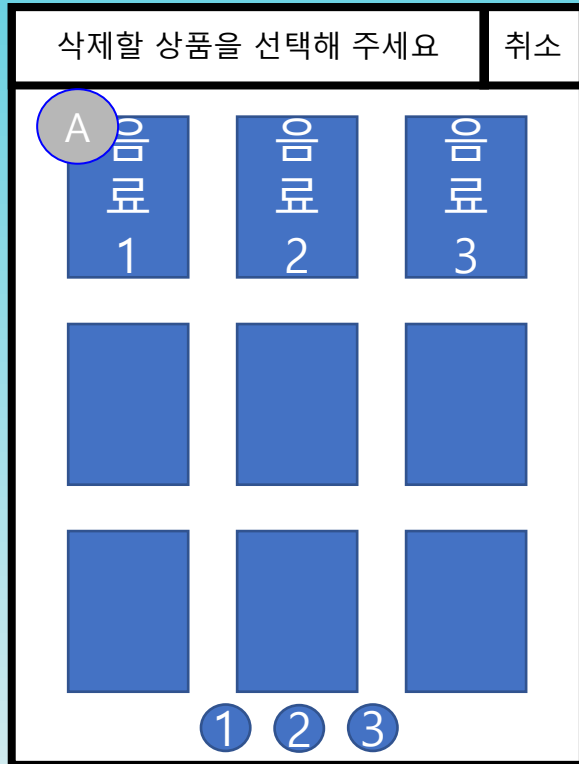
2043.

2044.

2045.

2047.

관리자 모드 - 상품 목록 삭제 : Window-14, 15



- A. 삭제 확인 화면(Window-15)으로 넘어간다.
- B. 상품을 삭제하고 상품 삭제 화면(Window-14)으로 넘어간다.
- C. 상품 삭제 화면(Window-14)으로 넘어간다.
- D. 관리자 모드 화면(Window-12)으로 넘어간다.
- E. 상품의 사진/이름을 출력한다.

2041.

2042. ◀

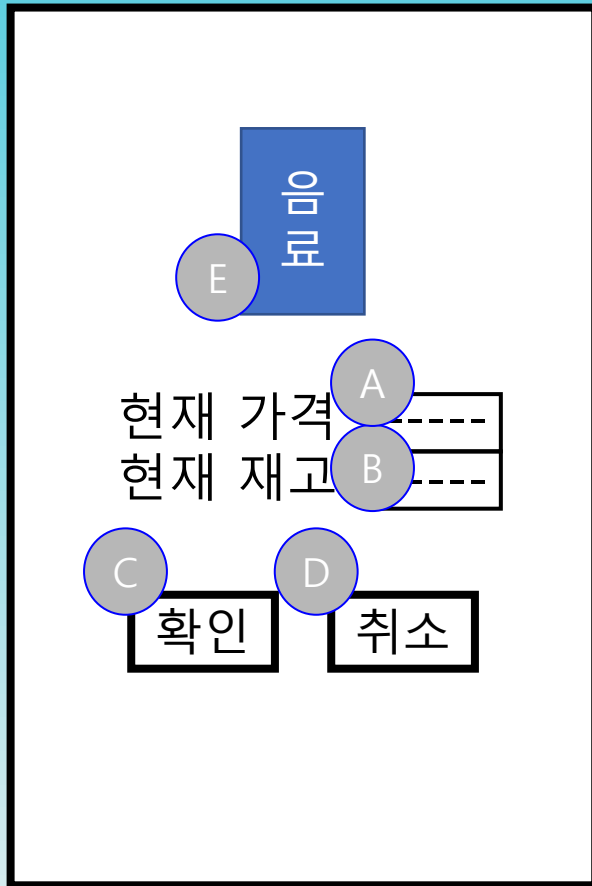
2043.

2044.

2045.

2047.

관리자 모드 - 상품 가격, 재고 변경 : Window-16



- A. 상품의 가격을 출력한다. 수정할 가격을 넘버패드(Window-18)로 입력 받는다.
- B. 상품의 재고를 출력한다. 수정할 재고를 넘버패드(Window-18)로 입력 받는다.
- C. 관리자 모드 화면(Window-12)으로 이동한다.
- D. 관리자 모드 화면(Window-12)으로 이동한다.
- E. 상품의 사진/이름이 출력된다.



2041.

2042. ◀

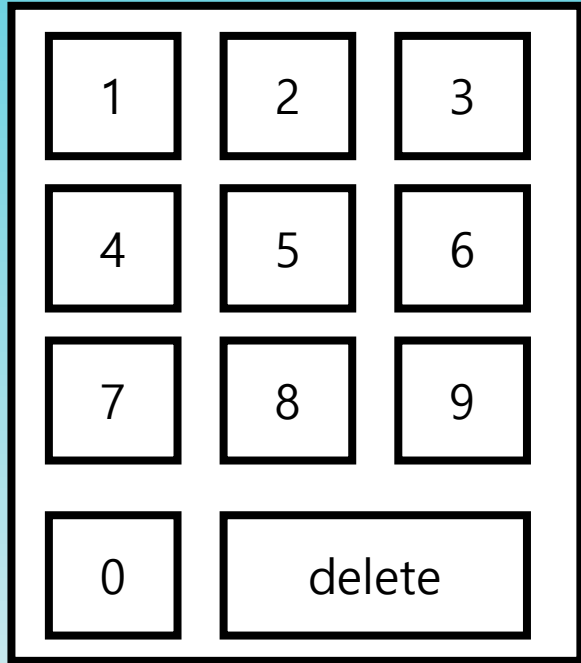
2043.

2044.

2045.

2047.

숫자패드 : Window-18



가격 변경, 재고 변경 시 화면에 출력한다.

2041.

2042. ◀

2043.

2044.

2045.

2047.

오류 화면 : Window-19, 21, 22

잘못된 입력입니다.

더이상 추가할 수 없습니다.

이 기기에서는  
해당 선결제코드를 사용할 수  
없습니다.

2041.

기타 화면 : Window-23, 24, 26

2042. ◀

죄송합니다

모든 자판기에 재고가  
존재하지 않습니다.

빠른 시일내에 재고를 확보하  
도록 하겠습니다

카드가 제거 되었습니다.

결제를 실패하였습니다.

2043.

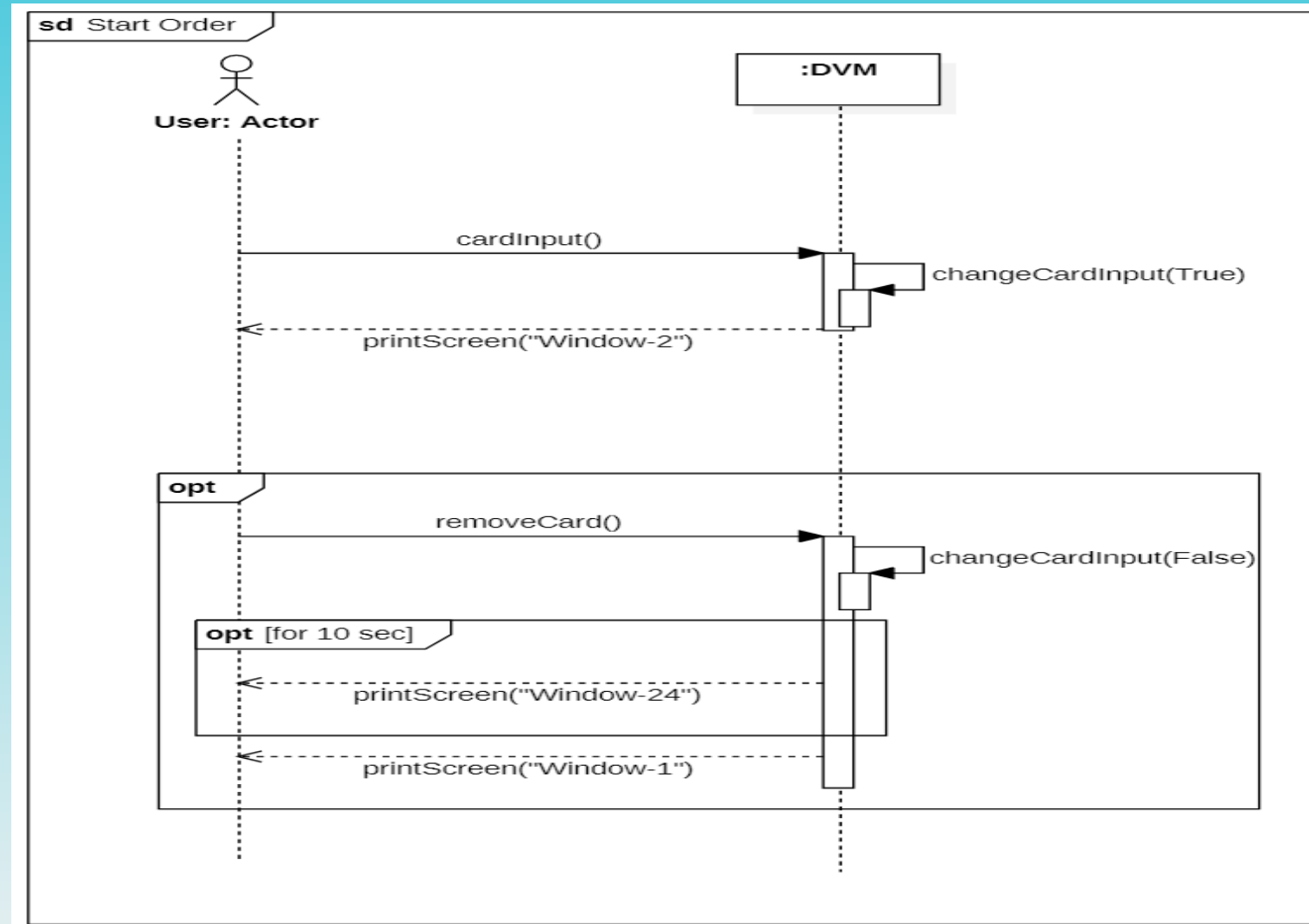
2044.

2045.

2047.

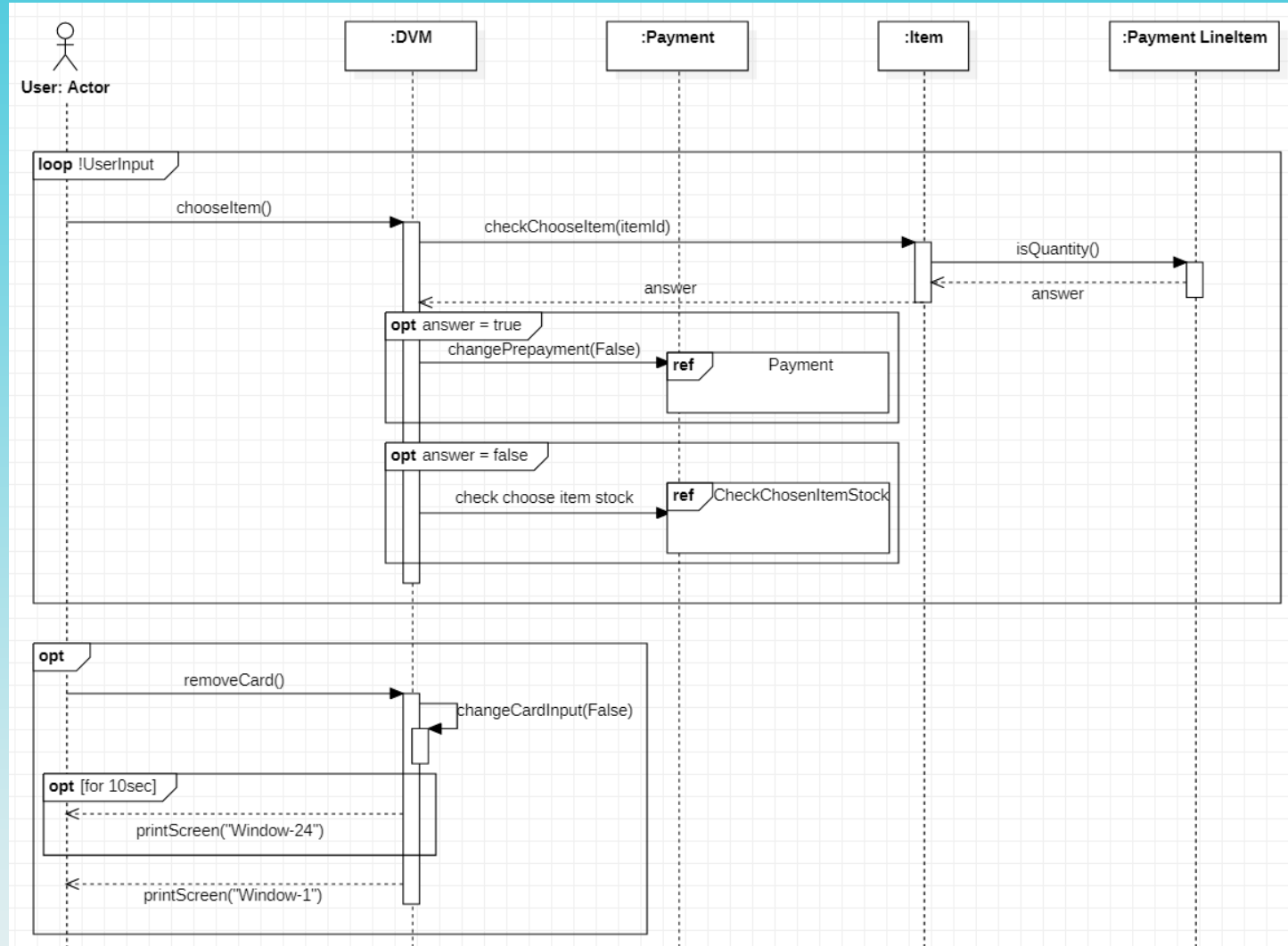
# Define Interaction Diagrams

## 1. Start Order



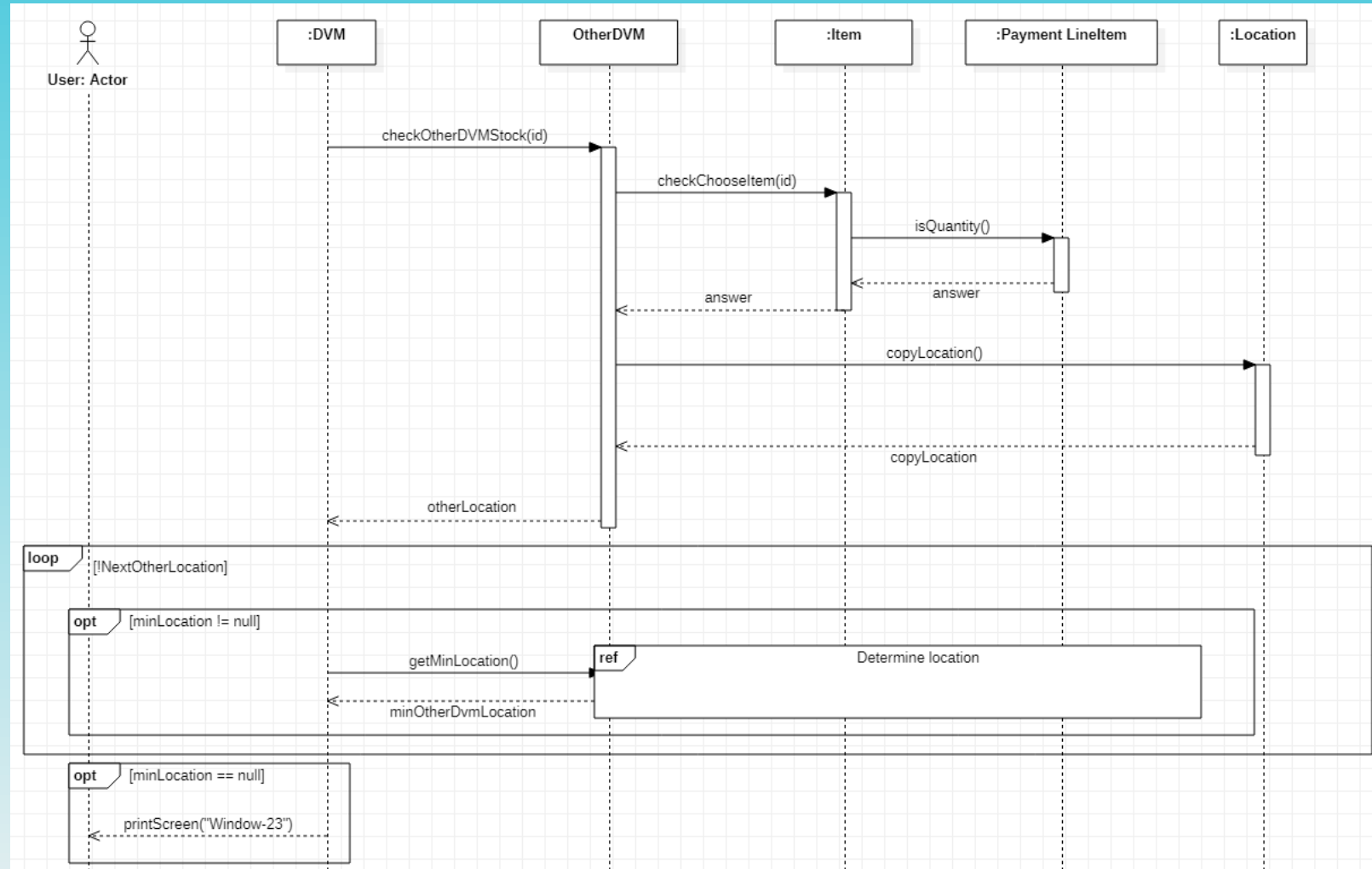
# Define Interaction Diagrams

## 2. Input Choice





## 3. CheckChosenItemStock



# Define Interaction Diagrams

2041.

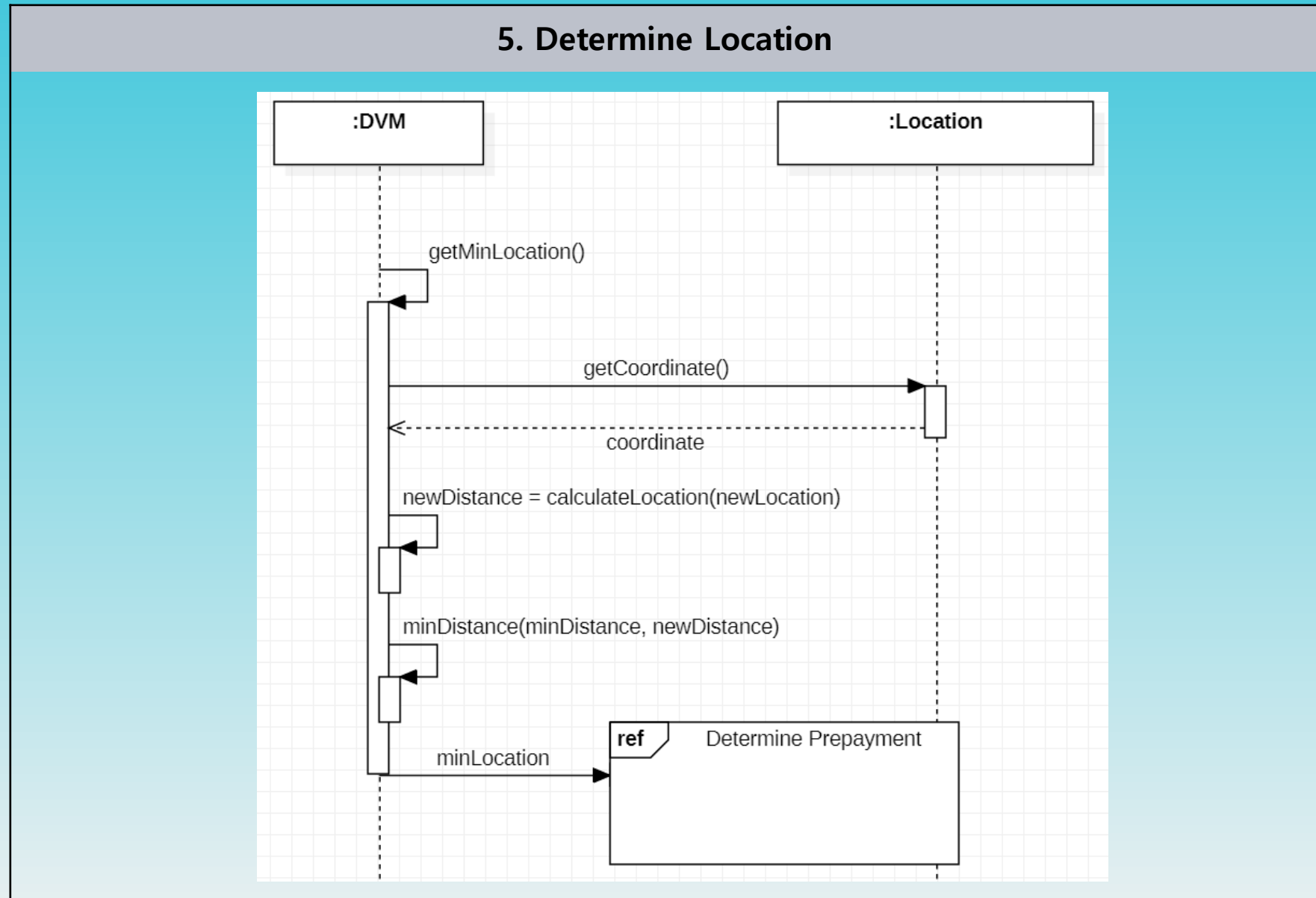
2042.

2043. ◀

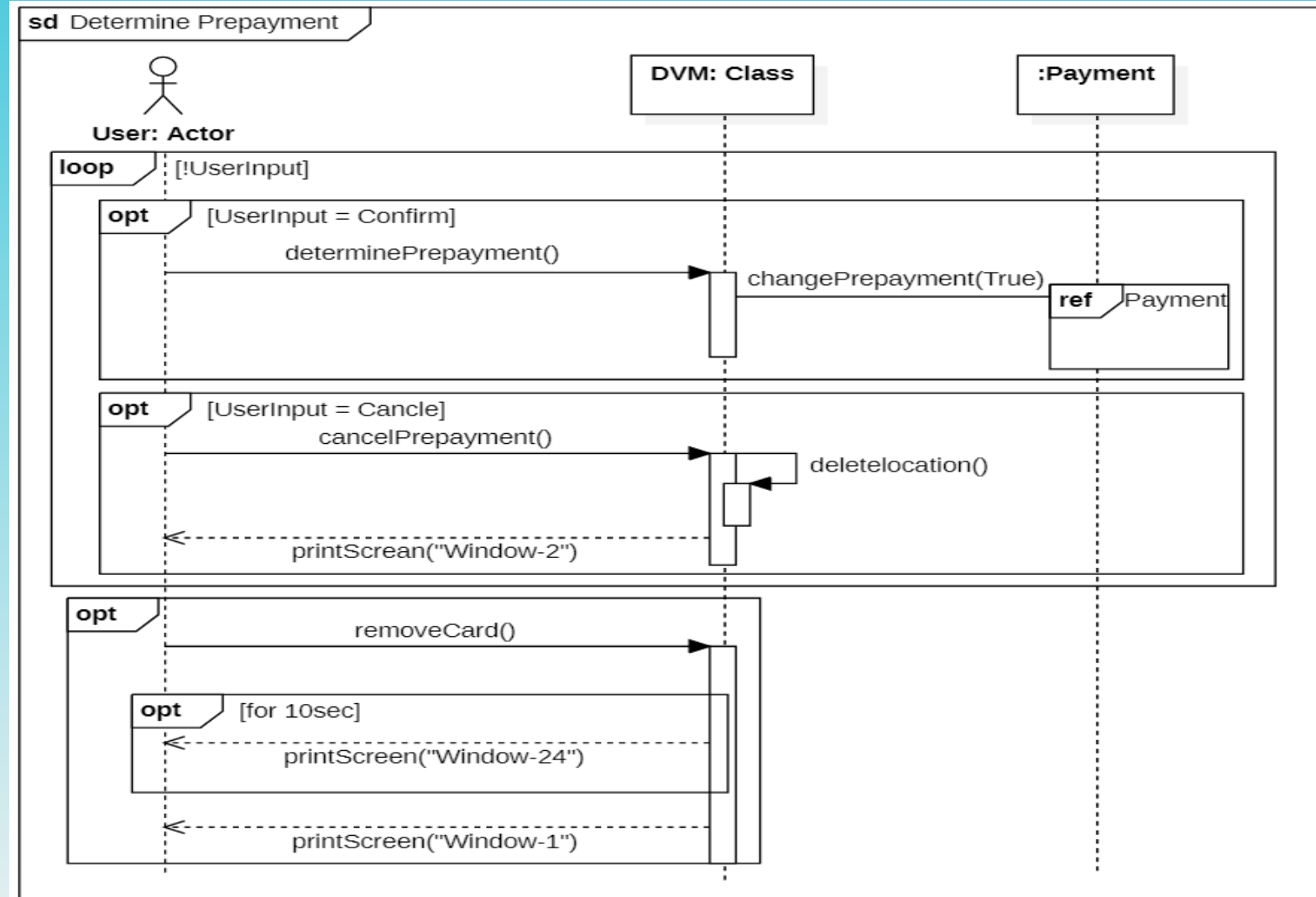
2044.

2045.

2047.



## 6. Determine Prepayment



# Define Interaction Diagrams

2041.

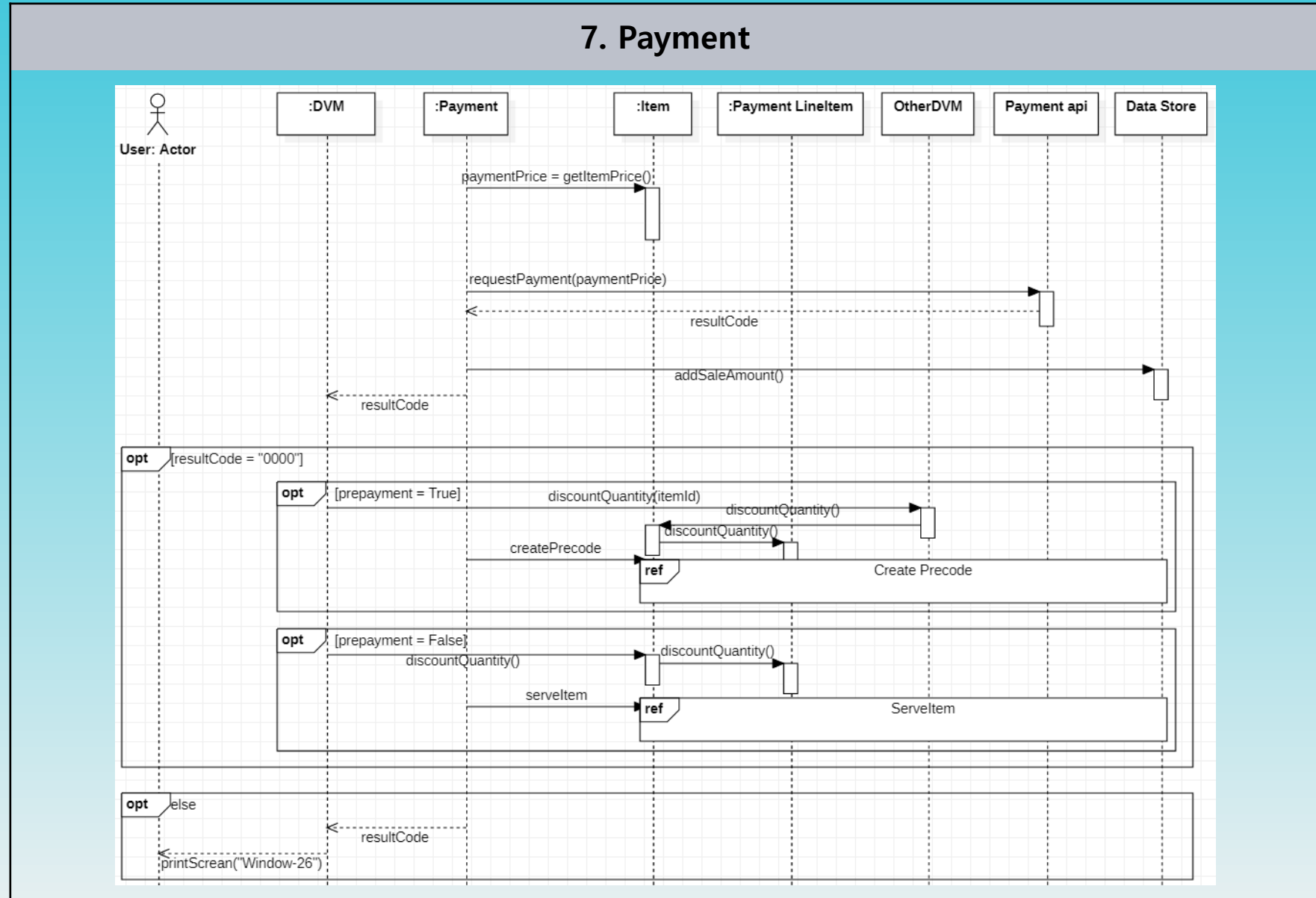
2042.

2043. ◀

2044.

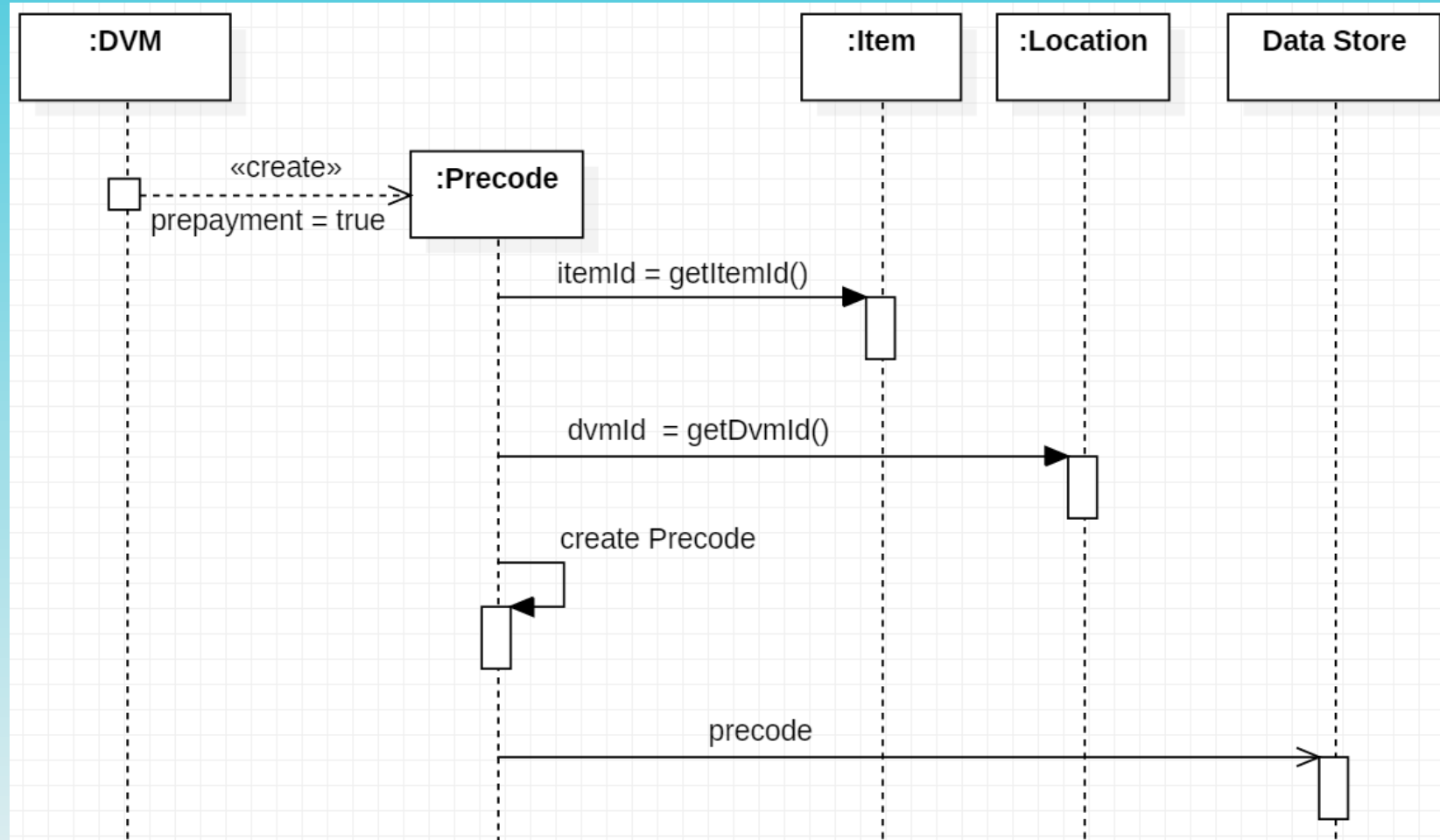
2045.

2047.

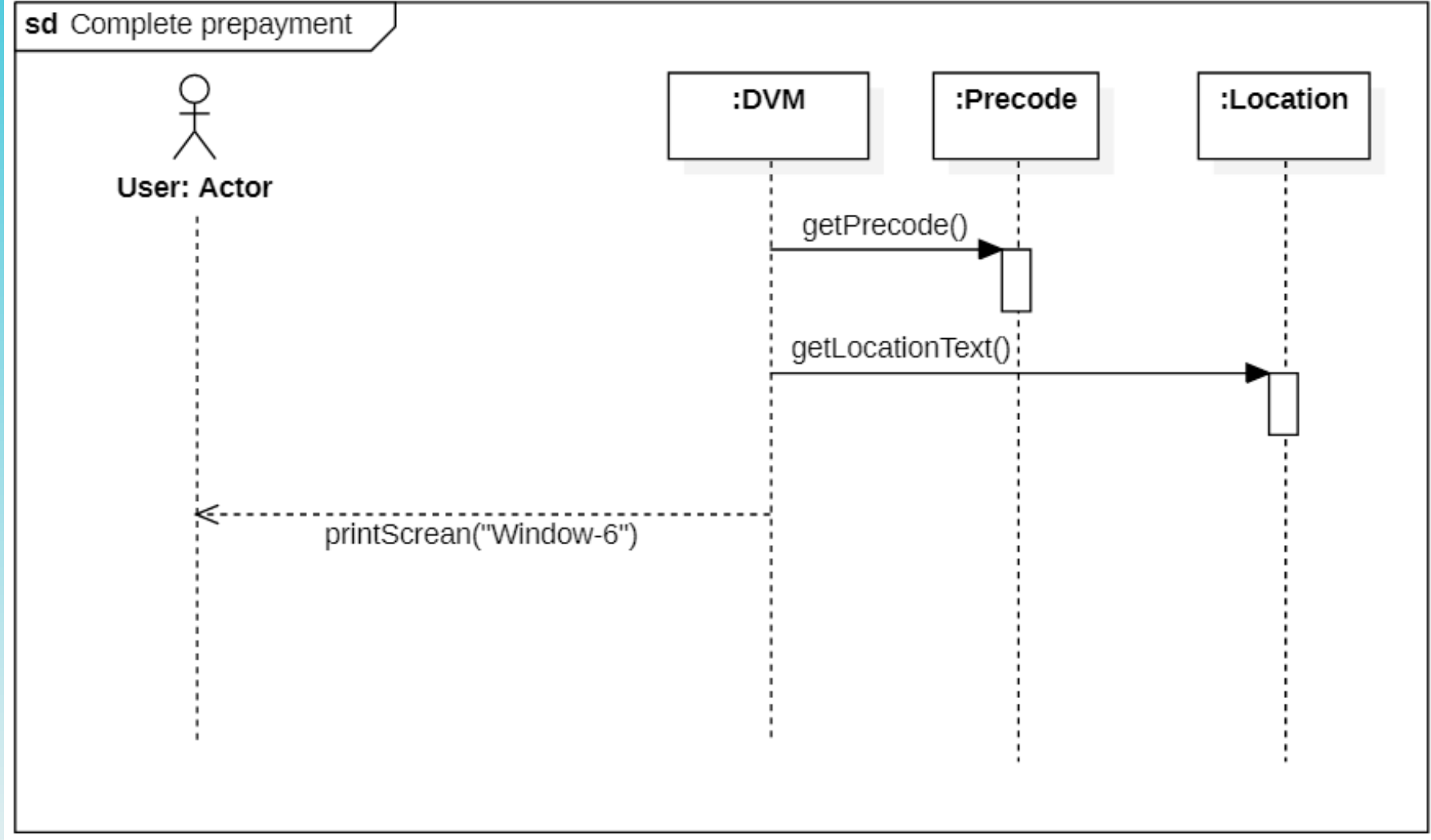


# Define Interaction Diagrams

## 8. Create Precode



### 9. Complete Prepayment



2041.

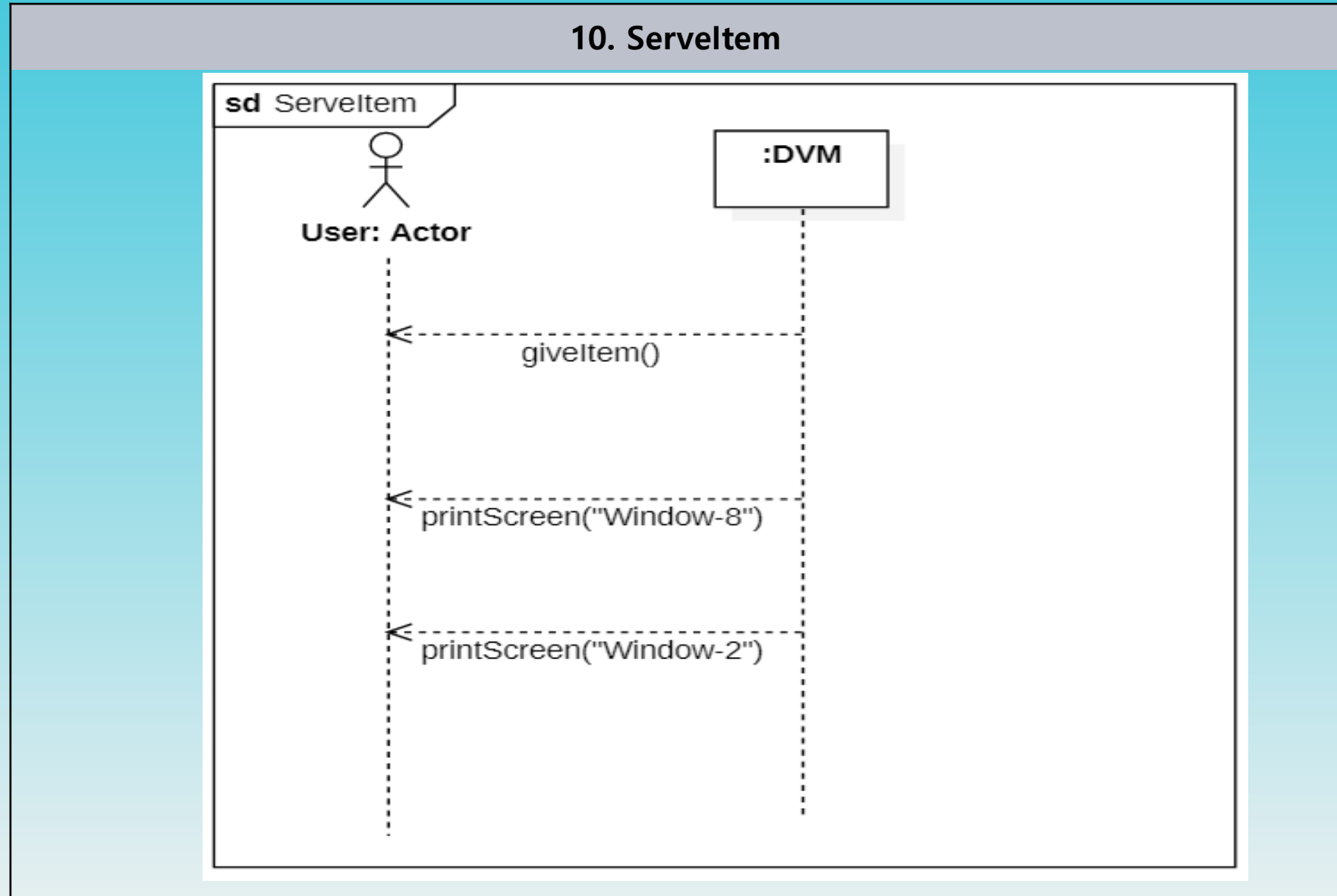
2042.

2043. ◀

2044.

2045.

2047.



# Define Interaction Diagrams

2041.

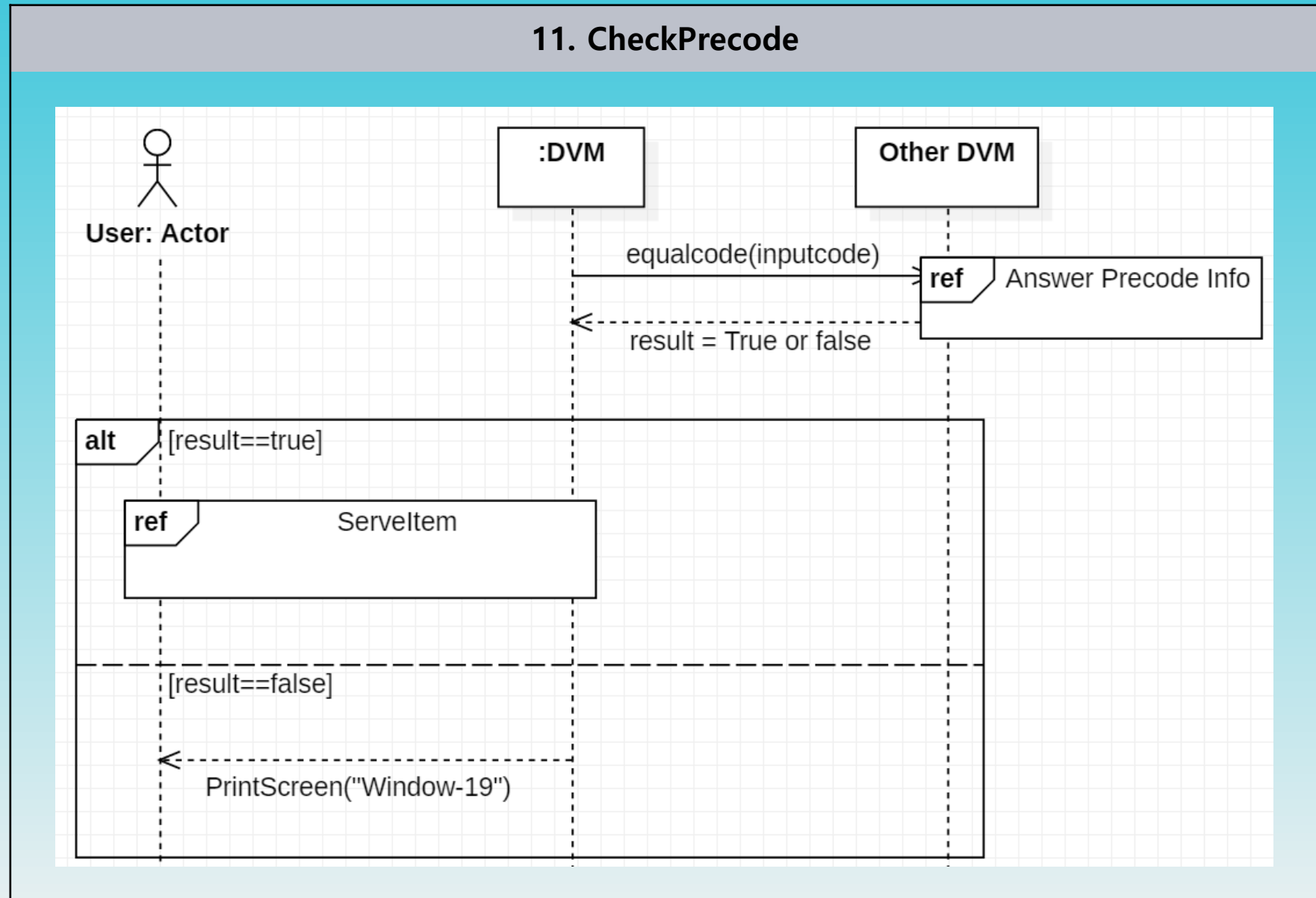
2042.

2043. ◀

2044.

2045.

2047.





2041.

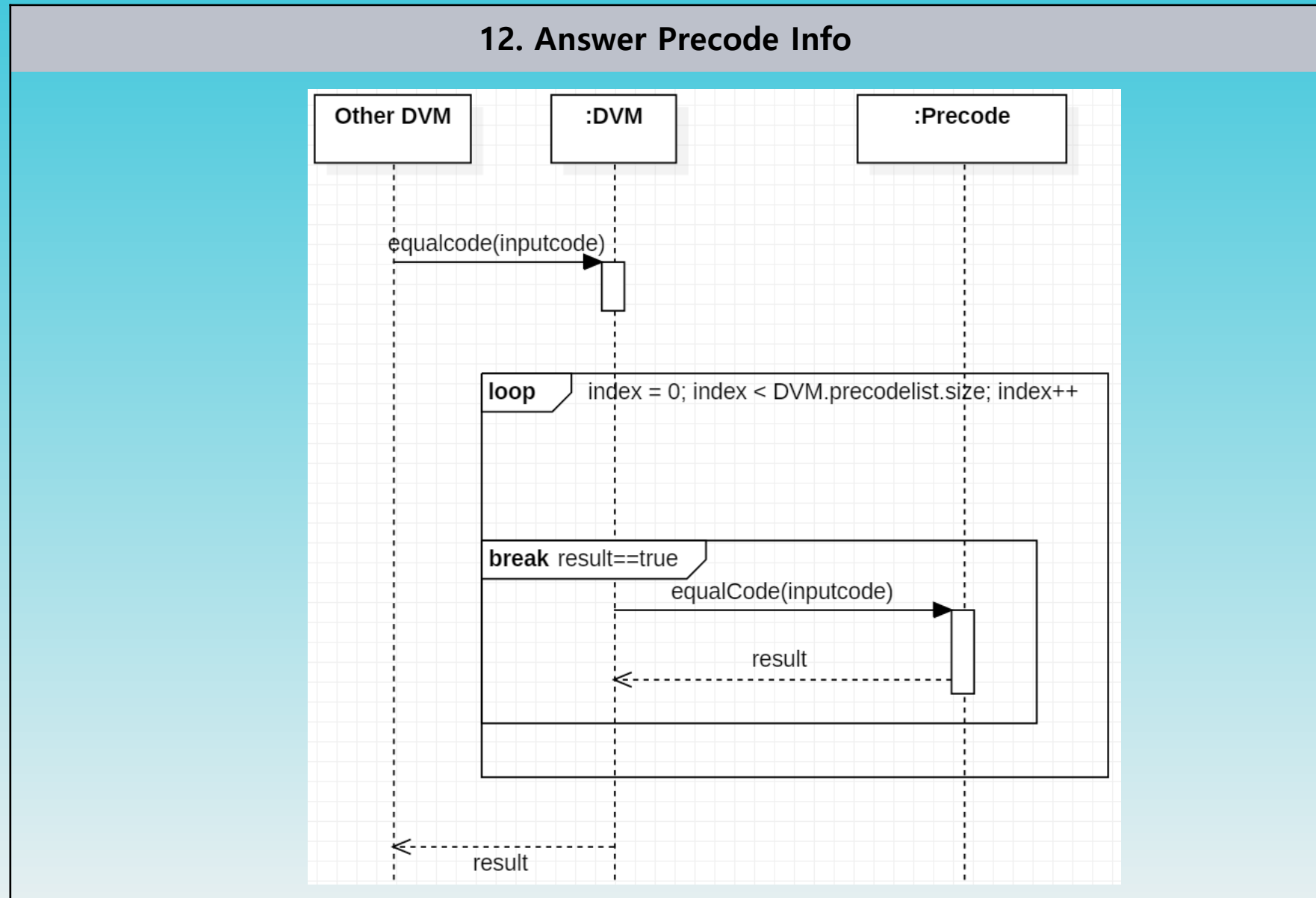
2042.

2043. ◀

2044.

2045.

2047.



# Define Interaction Diagrams

2041.

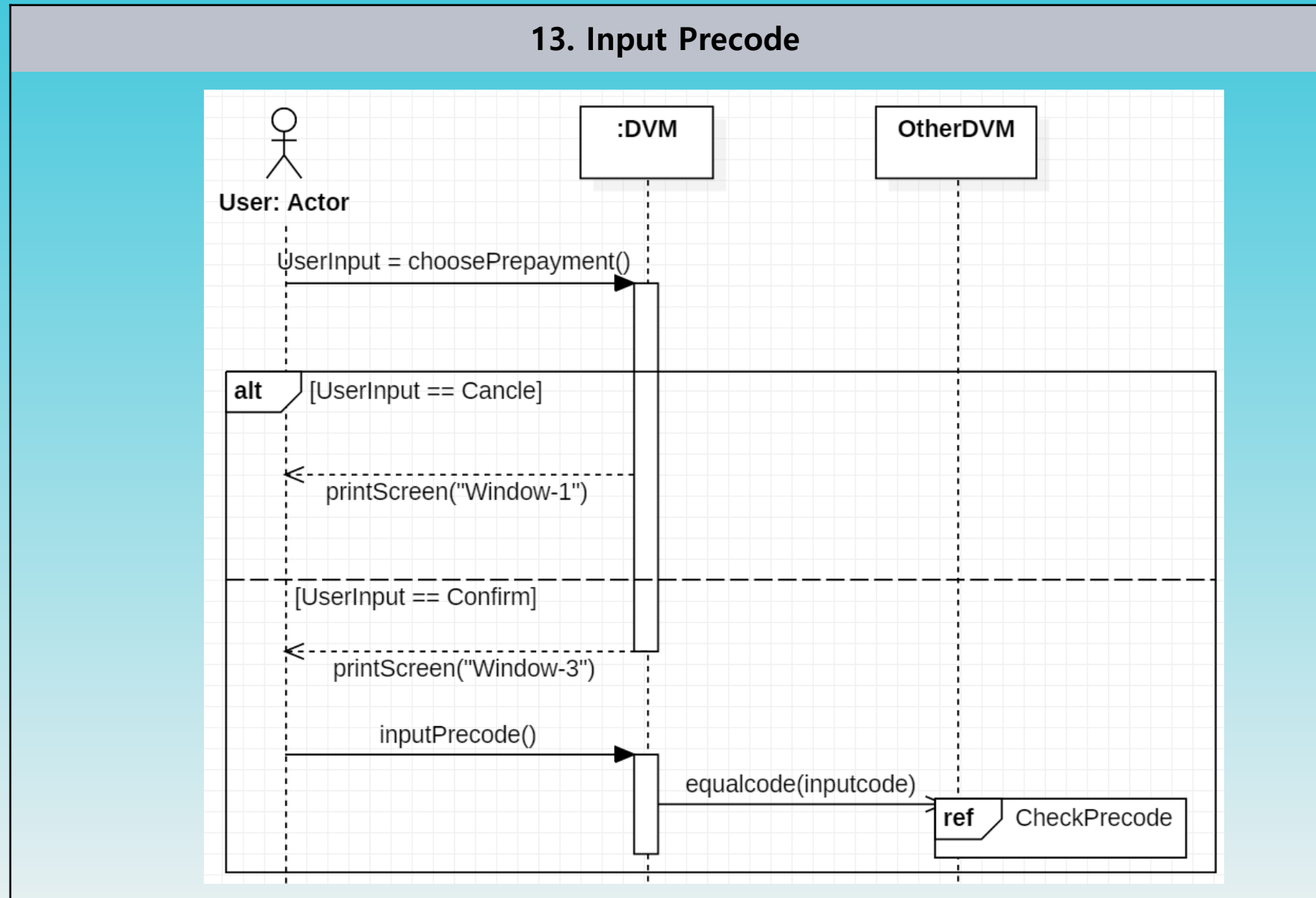
2042.

2043. ◀

2044.

2045.

2047.



# Define Interaction Diagrams

2041.

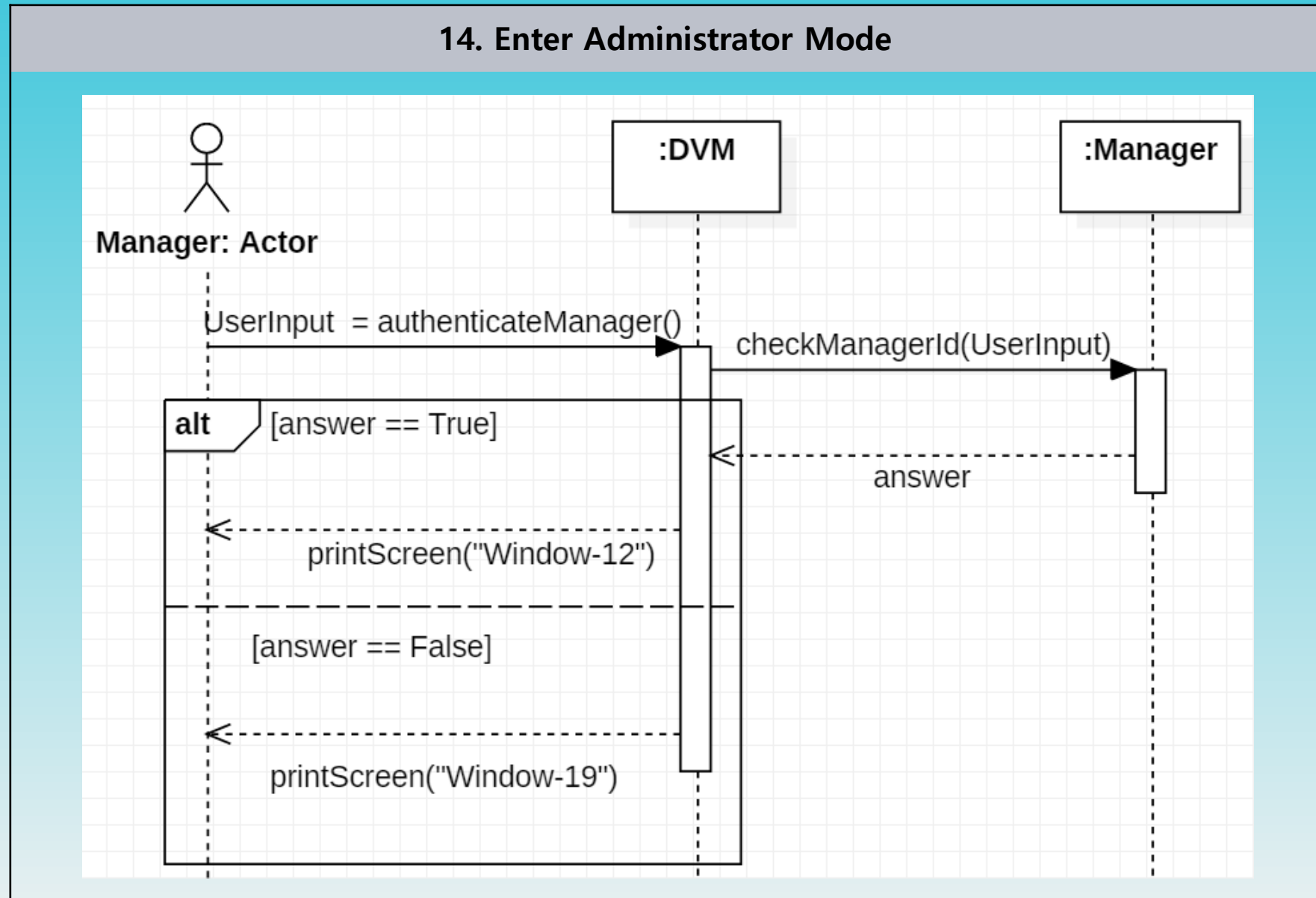
2042.

2043. ◀

2044.

2045.

2047.



2041.

2042.

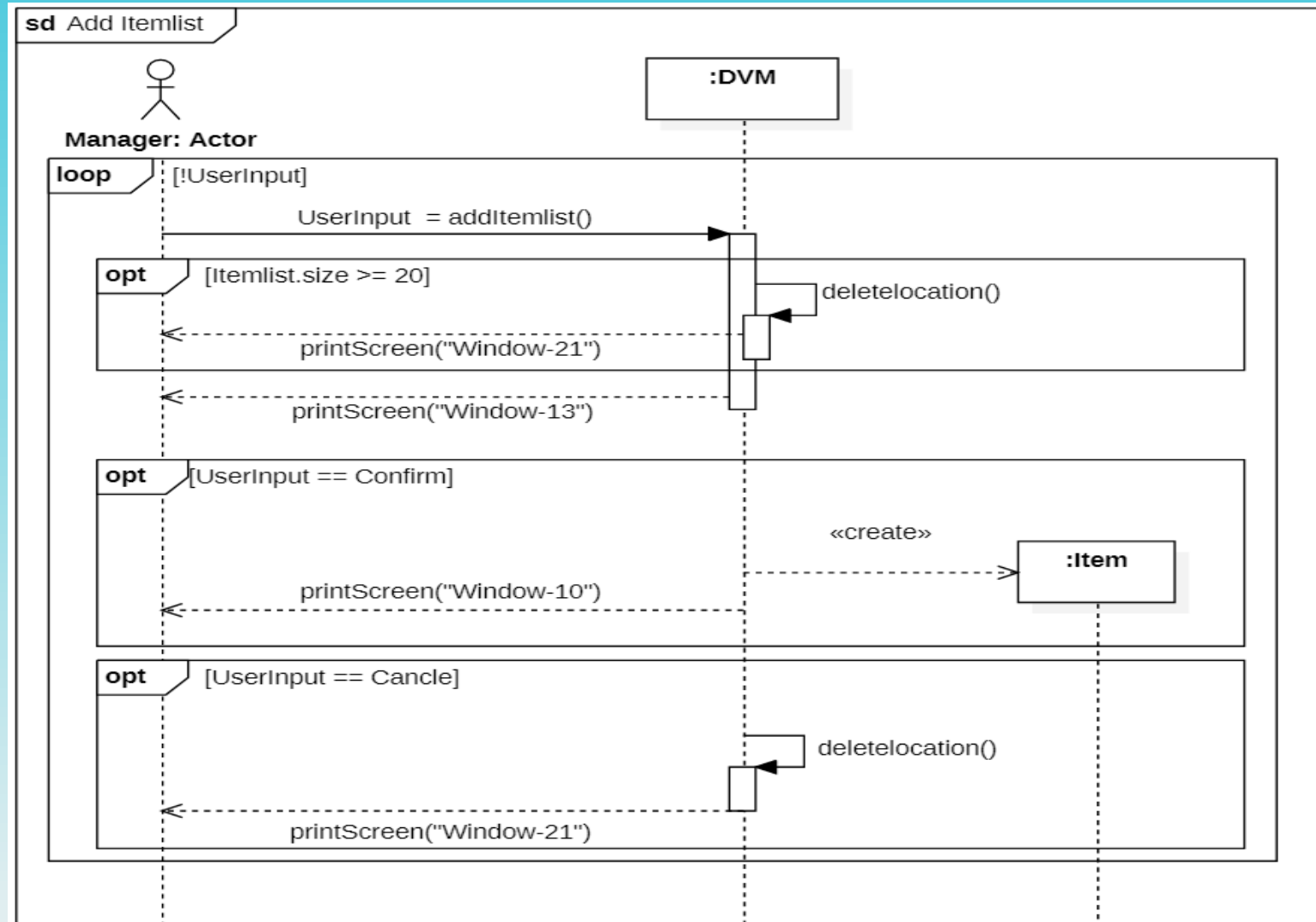
2043. ◀

2044.

2045.

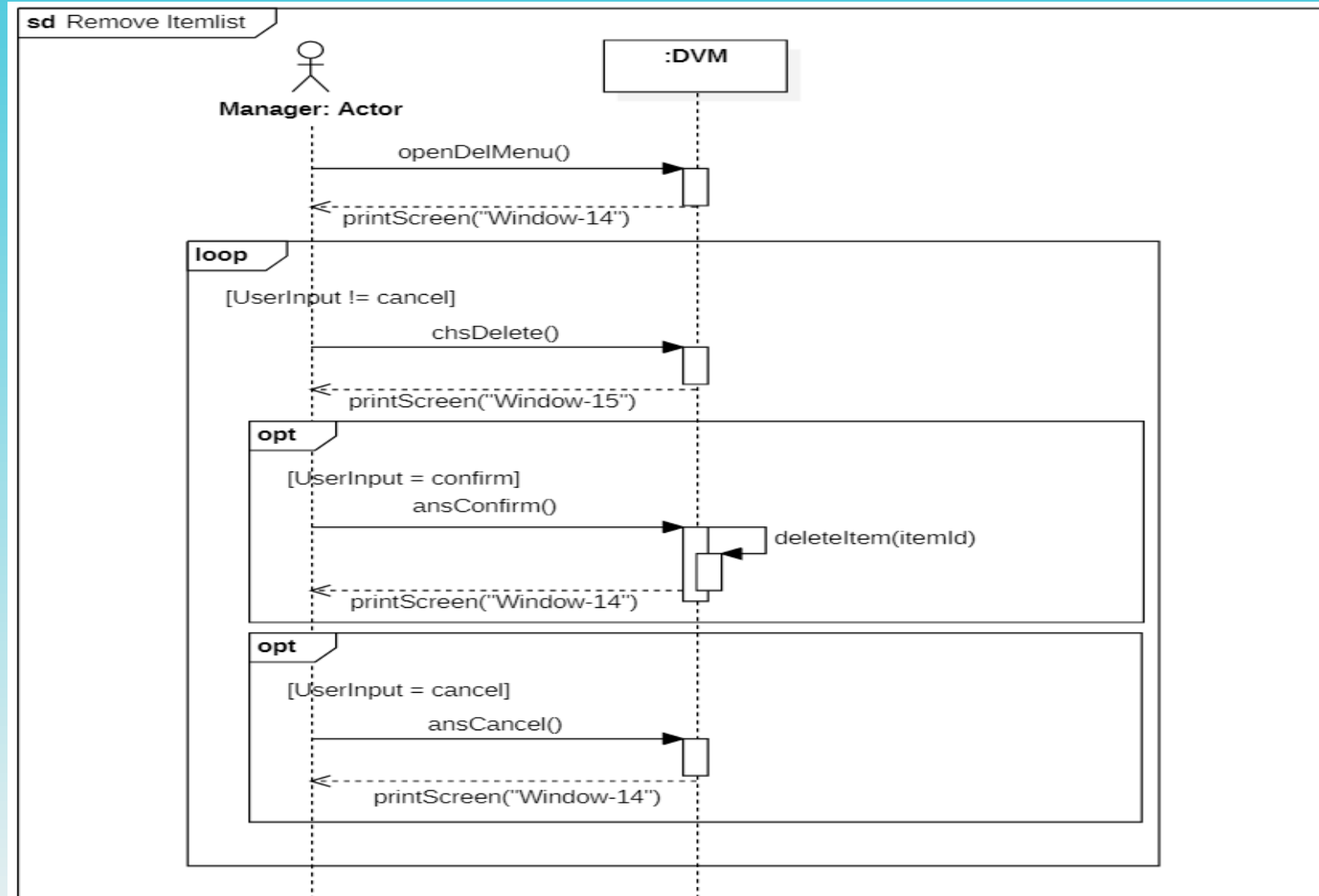
2047.

### 15. Add Itemlist



# Define Interaction Diagrams

## 16. Remove Itemlist



2041.

2042.

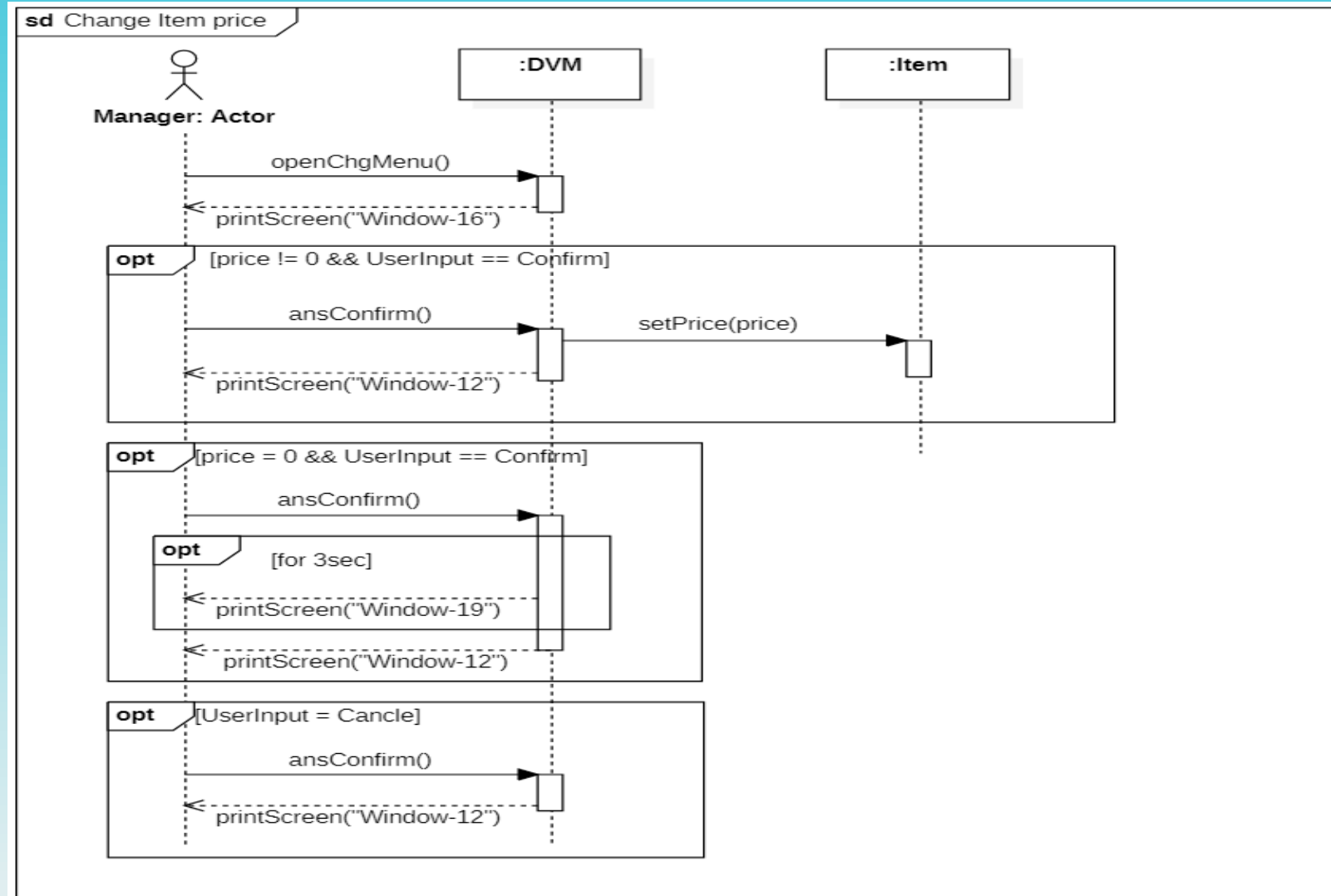
2043. ◀

2044.

2045.

2047.

### 17. Change Item Price



# Define Interaction Diagrams

2041.

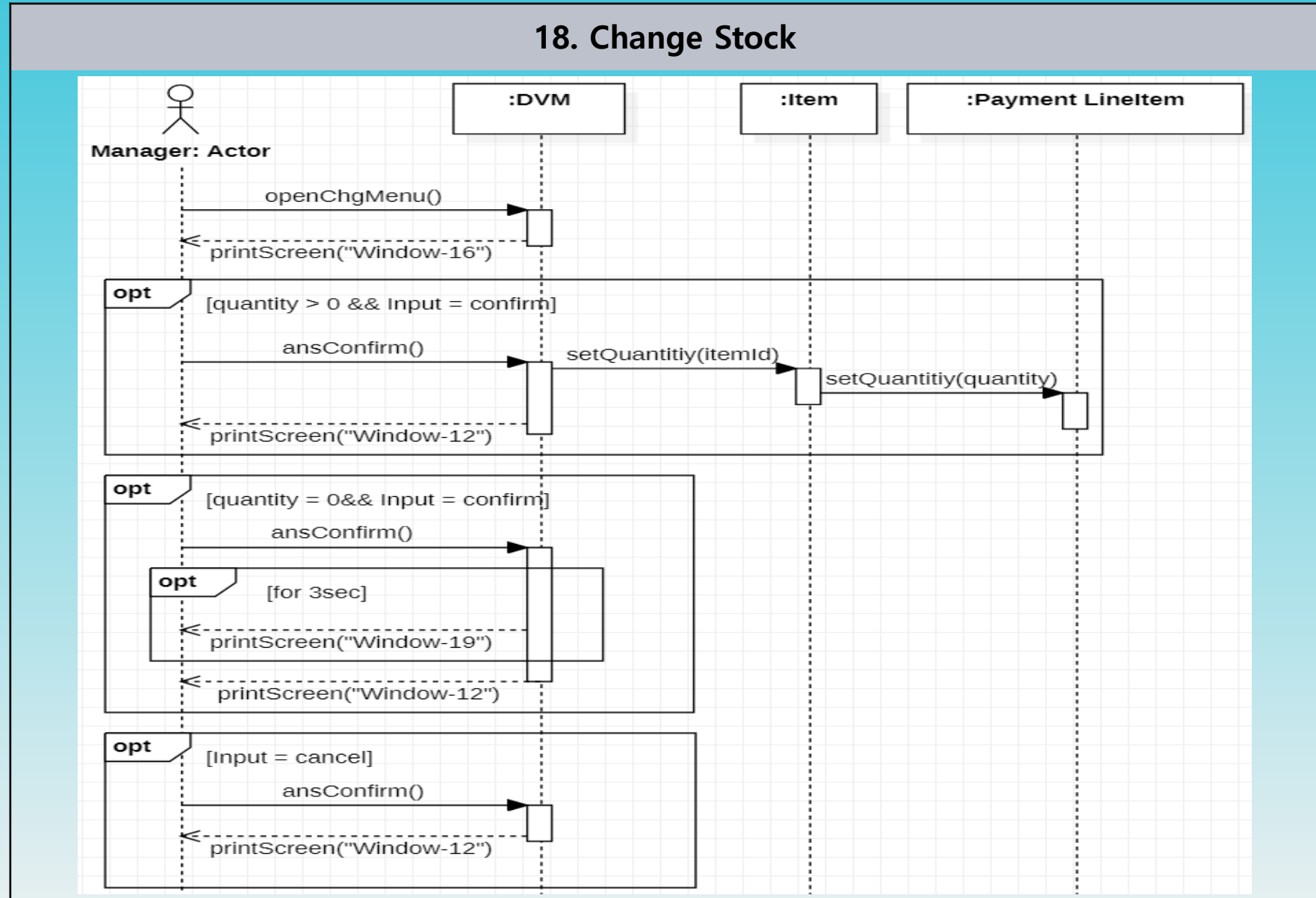
2042.

2043. ◀

2044.

2045.

2047.



# Define Interaction Diagrams

2041.

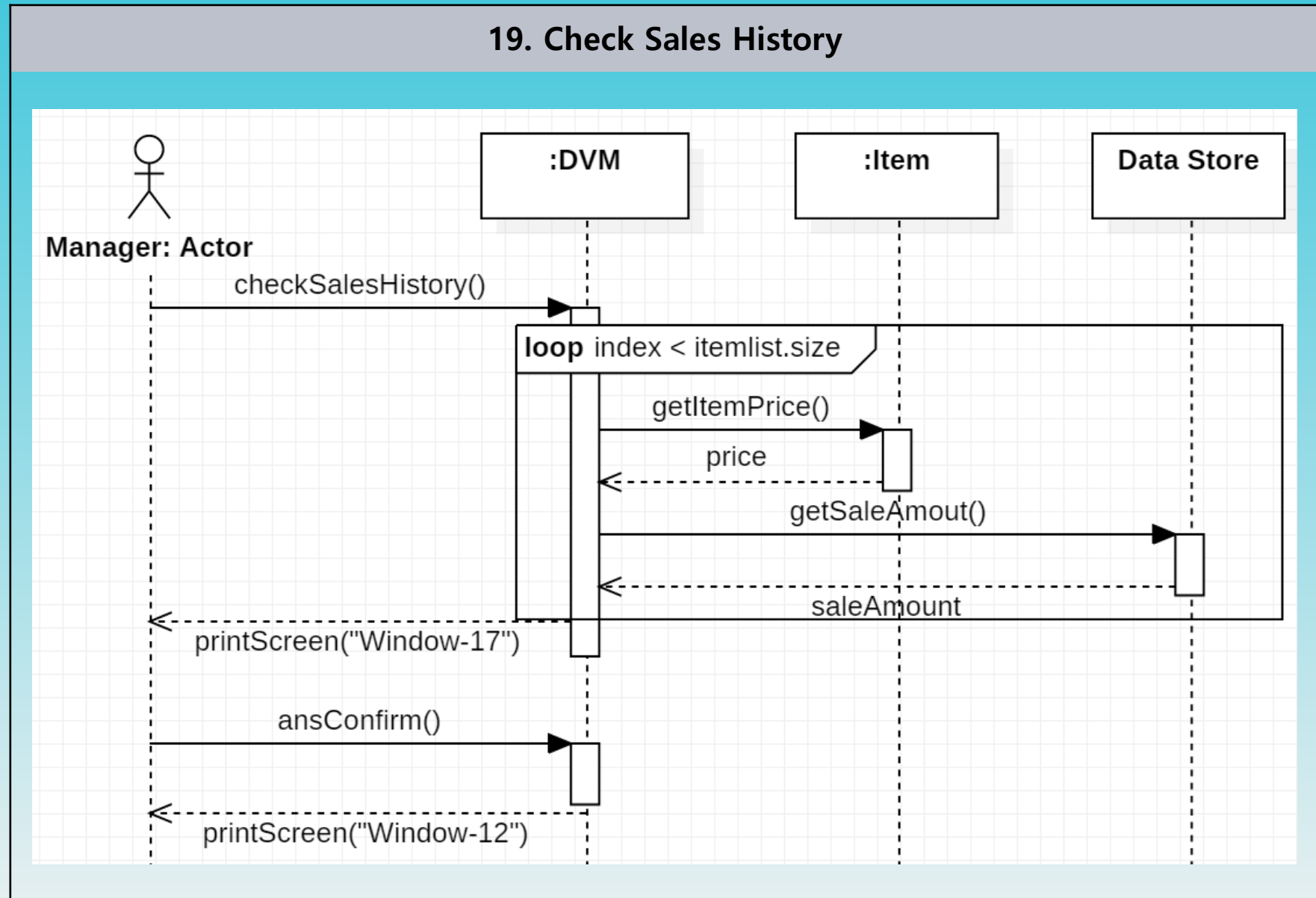
2042.

2043. ◀

2044.

2045.

2047.





2041.

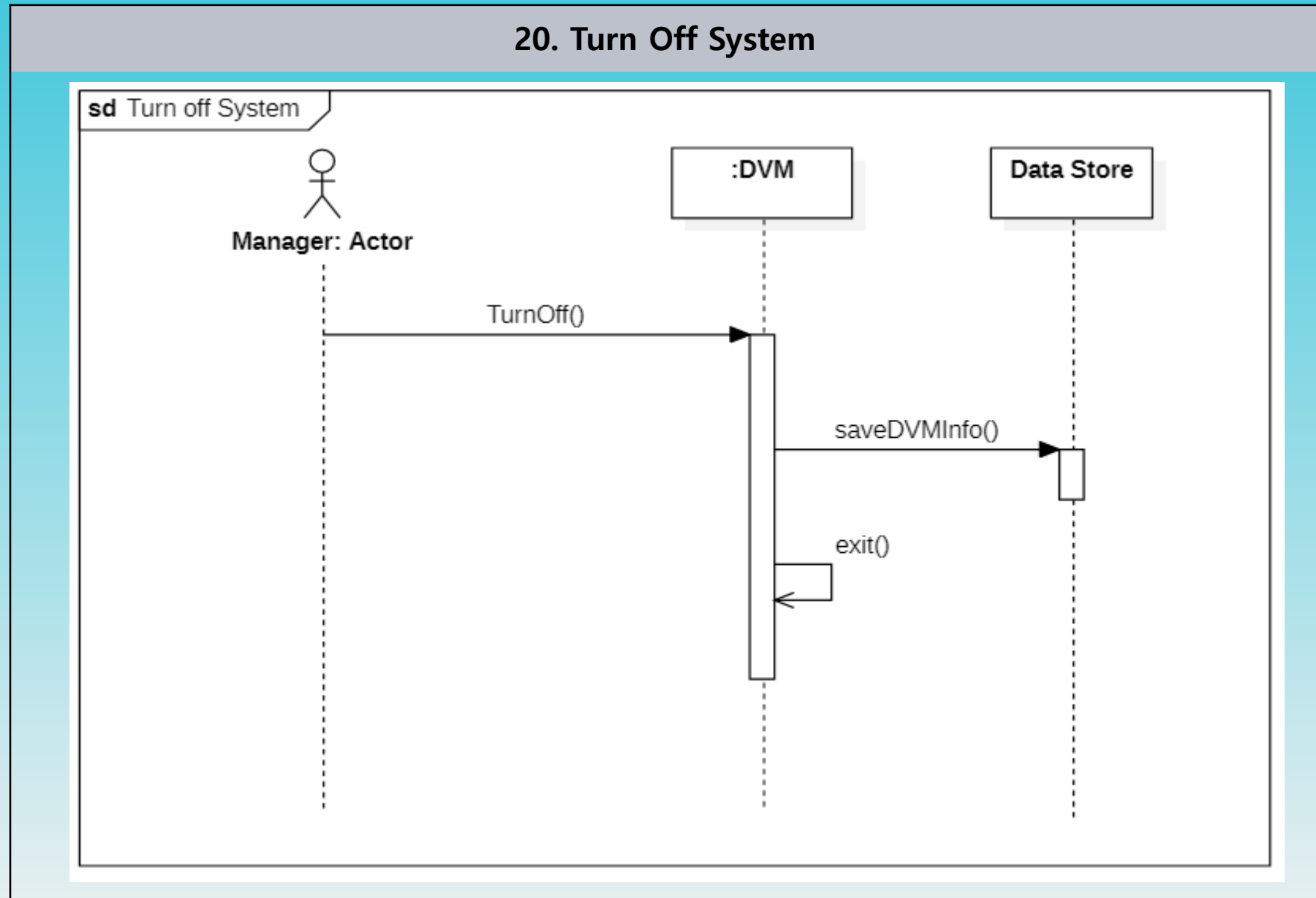
2042.

2043. ◀

2044.

2045.

2047.



2041.

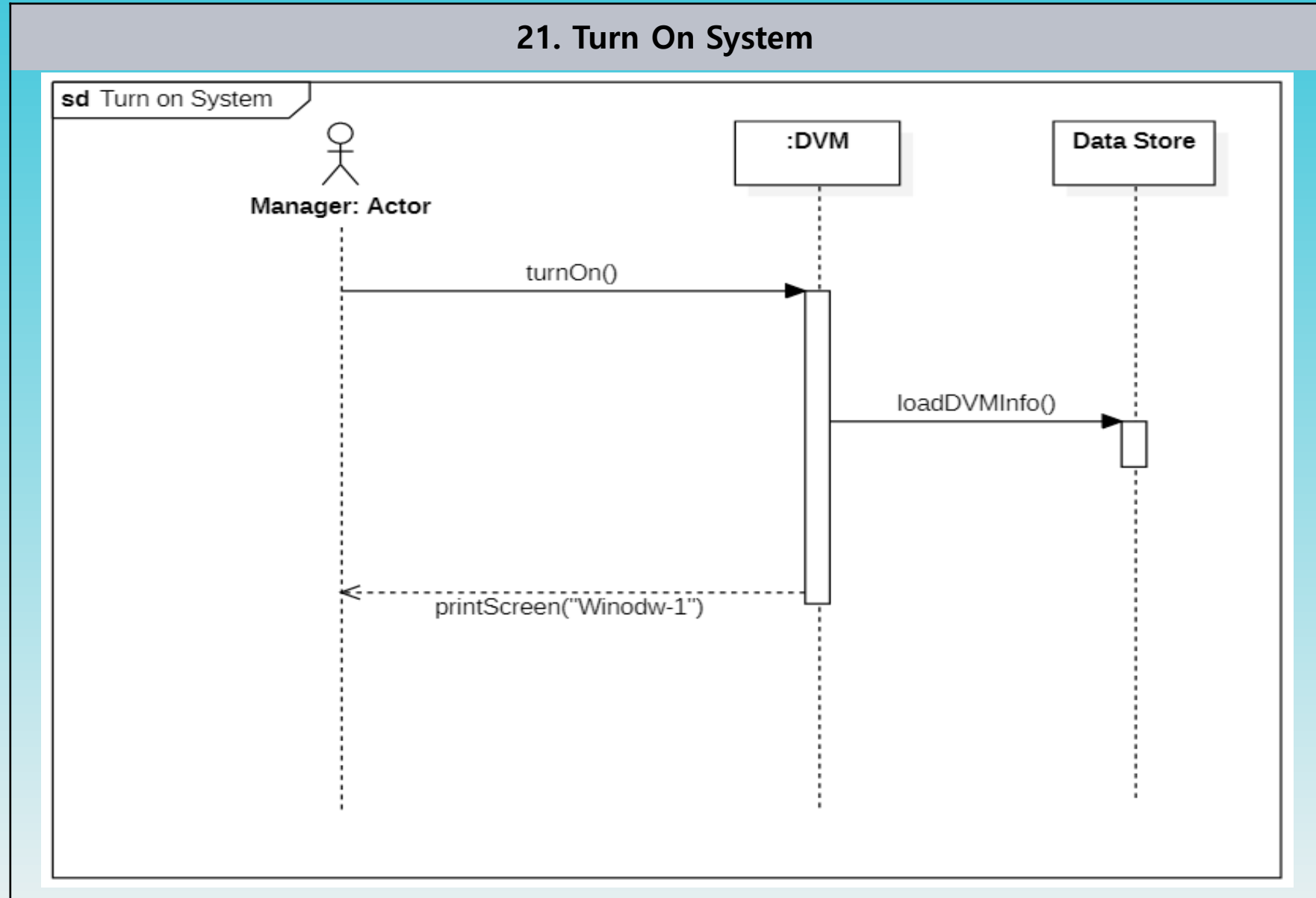
2042.

2043. ◀

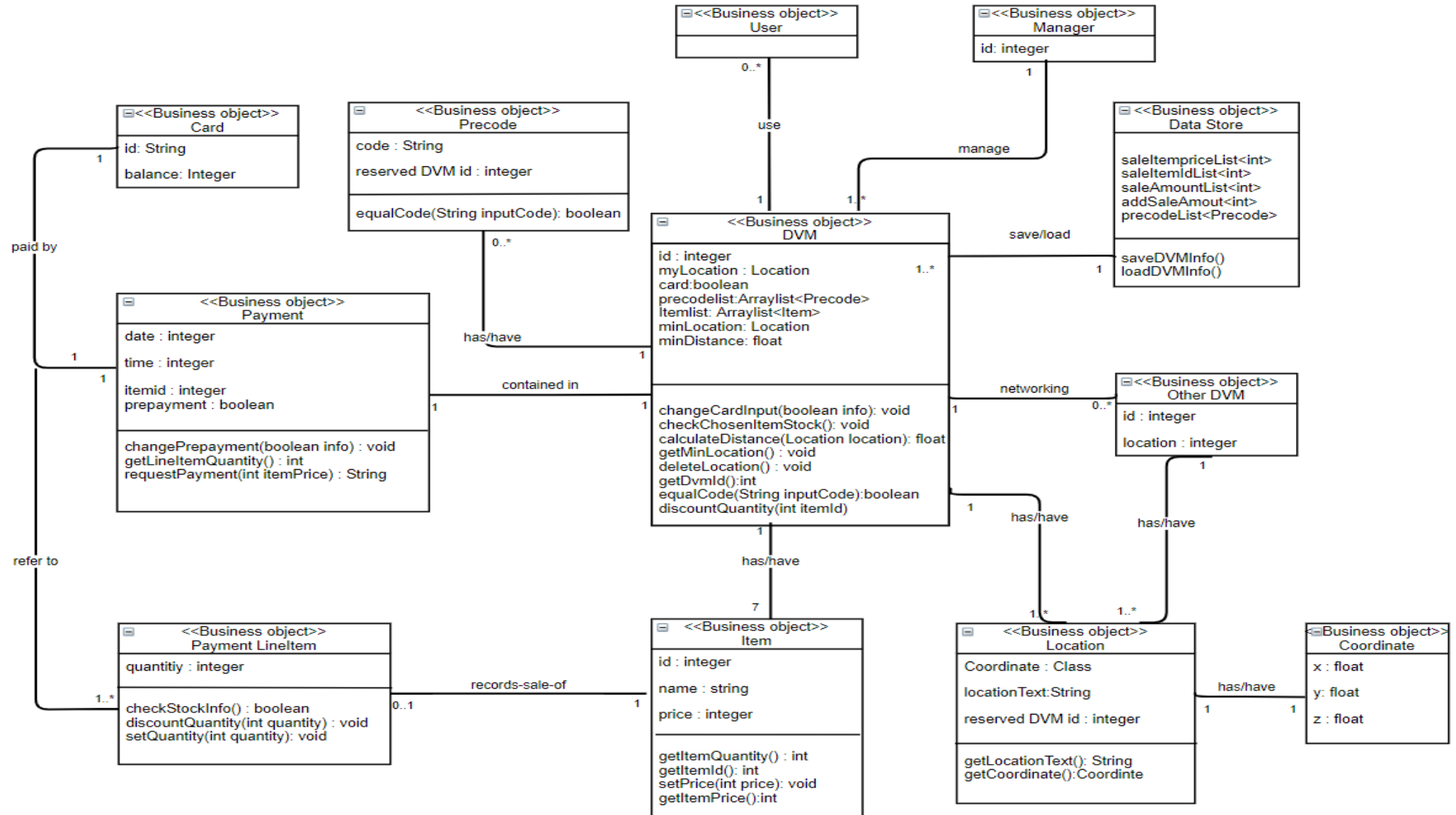
2044.

2045.

2047.



# Define Design Class Diagram



# System Architecture Description

2041.

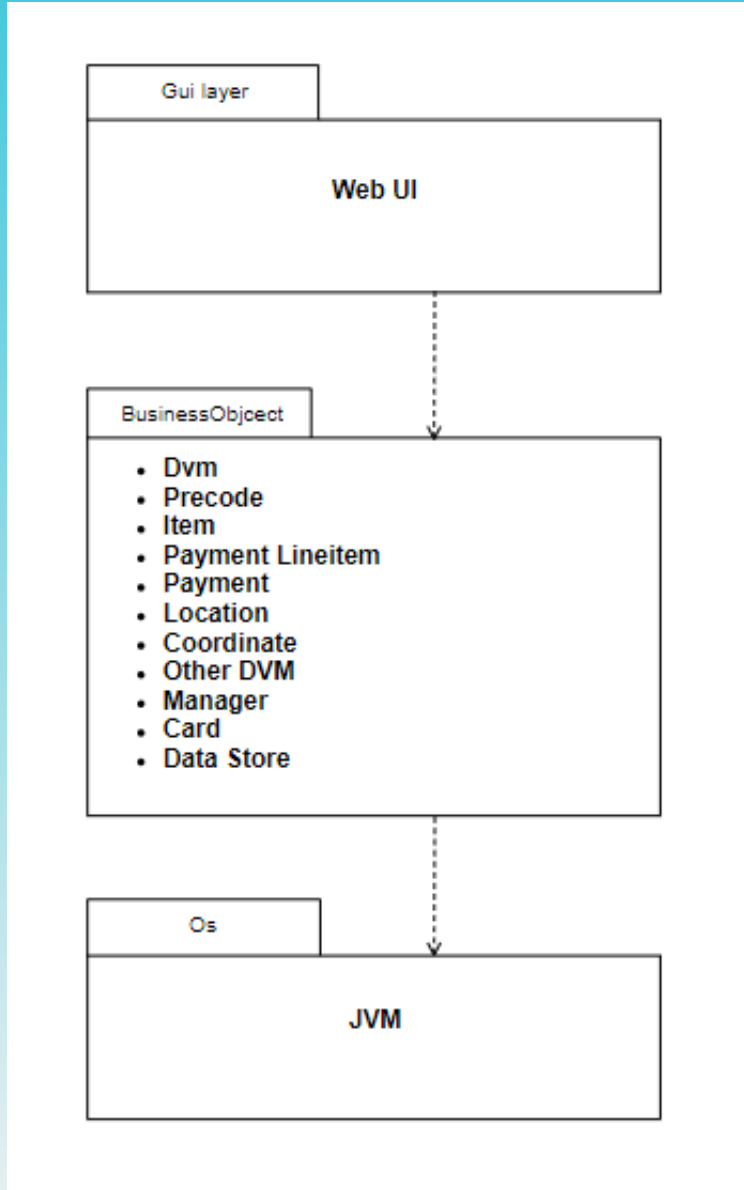
2042.

2043.

2044

2045. ◀

2047.



# Perform 2040 Traceability Analysis

System Function	Essential Use Case	System Operation	Method	Class
start order	Start Order	cardInput()	changeCardInput(boolean info): void	DVM
input choice	Input Choice	chooseItem()	checkChosenItem(): void	
check chosen item stock	Check Chosen Item Stock	answerStockInfo()	calculateDistance(Location location): float	
answer chosen item stock	Answer Chosen Item Stock I	answerLocationInfo()	getMinLocation() : void	
determine location	Determine Location	determinePrepayment()	deleteLocation() : void	
determine prepayment	Determine Prepayment	answerPrecodeInfo()	equalCode(String inputCode):boolean	
payment	Payment	choosePrepayment()	discountQuantity(int itemId)	
create precode	Create Precode	writePrecode()	checkOtherDVMStock(int): location	
complete prepayment	Complete Prepayment	authenticateManager()	cardInput():void	
serve item	Serve Item	changeItemList()	determinePrepayment():void	
check precode	Check Precode	changeItemStock()	cancelPrepayment():void	
answer precode info	Answer Precode Info	checkAllStock()	choosePrepayment():String	
input precode	Input Precode	changeItemPrice()	inputPrecode():void	
enter administrator mode	Enter Administrator mode	checkSalesHistory()	addItemList():void	
add itemlist	Add Itemlist	turnOffSystem()	openDelMenu():void	
remove itemlist	Remove Itemlist	turnOnSystem()	chDelete():void	
change stock	Change Stock		ansConfirm():void	
check stock	Check Stock		ansCancel():void	
change item price	Change Item Price		openChgMenu():void	
check sales History	Check Sales History		checkSalesHistory():void	
turn off system	Turn Off System		getItemQuantity() : int	Item
turn on system	Turn On System		getItemId() : int	
			setPrice(int price): void	
			getItemPrice():int	
			checkChooseItem(int) :boolean	Payment
			discountQuantity(int):void	
			setQuantity(int):void	
			changePrepayment(boolean info) : void	
			getLineItemQuantity() : int	Payment Lineitem
			requestPayment(int itemPrice) : String	
			checkStockInfo() : boolean	
			discountQuantity() : void	
			setQuantity(int quantity): void	Location
			isQuantity():boolean	
			getLocationText(): String	
			getCoordinate():Coordinate	
			copyLocation():Location	Precode
			getDvmId():int	
			equalCode(String inputCode): boolean	Data Store
			saveDVMInfo()	
			loadDVMInfo()	

감사합니다