

Change Log

index UseCase

2141.	1	Pre-Requisites	삭제
	2	Pre-Requisites	'전원이 커져..' 삭제 3. -> 3. 4. 분리
	3	Type Typical Courses	Evident -> Hidden (AU) 삭제 "Answer Stock Info" -> "Answer Chosen Stock Info" 3번 추가
	4	Exceptional Courses	카드 제거 시 exceptional 삭제
		Typical Courses	(AU) 삭제 1번 "Check Chosen.."..통해' 삭제
	5	Exceptional Courses	추가
		Typical Courses	(AU), (AD) 삭제 1.에 자세한 계산식 추가 2. '안내할 자판기를 결정한다' -> '계산 결과가 가장 작은 자판기를 안내할 자판기로 결정한다.' 3. Use Case 6번의 1번에서 이동됨
	6	Typical Courses	(AD) -> (S) 1번 Use Case 5의 3번으로 이동
		Exceptional Courses	'2', 'all line'>'1'
	7	Typical Courses	(AU), (AD) 삭제 2번 추가
		Exceptional Courses	1-a. '(10% 확률)'추가 1-c. 추가 1... -> 2... 변경
		Pre-Requisites	'Payment에서 선결제 과정으로 진행되었다.' 추가 "Payment" -> "Determine Prepayment"
	8	Typical Courses	(AU), (AD) 삭제 '생성된 선결제 코드를 저장한다.' -> '결제된 상품의 선결제 코드를 생성하고 결제가 진행된 DVM에 선결제 코드를 저장한다.'
		Typical Courses	(AU), (AD) 삭제 (S) 추가
	9	Pre-Requisites	'자판기의 전원이 ...한다.' '취출구에 ..공간..한다.' 삭제
Typical Courses		(AU), (AD) 삭제 2. -> 3. 변경 3. -> 4. 변경 2. 추가 3. 앞에 '3초동안' 추가	
10	Type	Evident -> Hidden	
	Typical Courses	(AU) 삭제 전체 수정	
11	Exceptional Courses	전체 수정	
	Typical Courses	(AU), (AD) 삭제	
12	Exceptional Courses	10초간 입력이 없을시 삭제	
13	Purpose	수정	

- Overview 수정
- Pre-Requisites 삭제
- Typical Courses 3 “name, price”로 변경
- Exceptional Courses 3-ㄱ, 3-ㄴ, 3-A, 3-B,3-a,3-b 추가
- 16 Pre-Requisites “관리자모드이어야 한다.”로 변경
- Typical Courses 3. 4. 5. 수정
- Exceptional Courses 전체 변경
- 17 Pre-Requisites “관리자모드이어야 한다.”로 변경
- Typical Courses 3. 4. 5. 수정
- Exceptional Courses 전체 변경
- 18 Use Case 17로 통합
- 19 18로 number 변경
- Pre-Requisites “생성될 Precode는 중복되지 않는다.” 추가
- 2143. 11 제외 모두 변경
- 2144. User, Manager, Coordinate, Payment Lineltem, Card 삭제
Data Store -> DB
DVM, Precode, Payment, Item, Location, DB 내부 전체 수정
- 2145. User, Manager, Coordinate, Payment Lineltem, Card 삭제
Data Store -> DB
- 2147. 변경사항을 적용한 최신버전

10. Serve Item

Typical Courses of Events - (2) 결제된 상품의 재고를 1개 줄인다 추가
(2) -> (3), (3) -> (4) 이동

-
use case 5 typical courses of events 1.(S) 현재 자판기에서 재고가 있는 자판기들의 거리를 계산한다. -> 현재 자판기와 재고가 있는 자판기들의 거리를 $|위도(현재 자판기)-위도(재고 존재 자판기)| + |경도(현재 자판기) - 경도(재고 존재 자판기)|$ 식을 통해 계산한다.

-
use case 5 typical courses of events 2.(S) 안내할 자판기를 결정한다 -> 계산 결과가 가장 작은 자판기를 안내할 자판기로 결정한다.

-
use case 8 pre-requistic Payment에서 선결제 과정으로 진행되었다. 추가

-
use case 8 사용하지 않는 (AU) : Actor User, (AD) : Actor other DVMs 삭제

-
use case 8 (S) 생성된 선결제 코드를 저장한다. -> (S) 결제된 상품의 선결제 코드를 생성하고 결제가 진행된 DVM에 선결제 코드를 저장한다.

use case 16 Exceptional Courses of Events 1-b. 추가 버튼이 비활성화 된다 -> 원래 없었음

use case 9 (AU) : Actor User, (AD) : Actor other DVMs -> 삭제, (S): System -> 추가

use case 4 (AU) : Actor User -> 삭제

use case 5 (AU) : Actor User, (AD) : Actor other DVMs -> 삭제

use case 7 (AU) : Actor User, (AD) : Actor other DVMs -> 삭제

use case 6 (AD): Actor of other DVM -> (S): System

use case 9 (AU) : Actor User, (AD) : Actor other DVMs -> 삭제
use case 10 (AU) : Actor User, (AD) : Actor other DVMs -> 삭제
use case 11 (AU) : Actor User -> 삭제
use case 12 (AU) : Actor User, (AD) : Actor other DVMs -> 삭제
use case 12 (AD) : Actor other DVMs-> 삭제

-
use case 6 1번 -> use case 5 3번 이동
use case 3 (AD) “Answer Stock Info”을 통해 위치, 상품 재고 정보를 받는다. -> (AD)
“Answer Chosen Stock Info”을 통해 위치, 상품 재고 정보를 받는다. 수정
use case 10 3번 앞에 ‘3초동안’ 추가

use case 2 Pre-Requisites 전원이 켜져있어야한다. 삭제
use case 2 Typical Courses 3번 -> 3번 4번 분리
use case 3 Typical Courses (AU) 삭제
use case 3 Typical Courses 3번 추가
use case 3 Exceptional Courses 카드제거시의 exceptional 삭제
use case 4 Typical Courses 1번 “Check Chosen...”..통해’ 삭제
use case 6 Exceptional Courses ‘2’, ‘all line’>’1’
use case 7 Typical Courses 2번 추가
use case 7 Exceptional Courses 1-a. ‘(10% 확률)’추가
use case 7 Exceptional Courses 1-c. 추가
use case 8 Pre-Requisites “Payment” -> “Determine Prepayment”
use case 10 Pre-Requisites ‘자판기의 전원이 ...한다.’ ‘취출구에 ..공간..한다.’ 삭제
use case 11 Typical Courses 전체 수정
use case 11 Exceptional Courses 전체 수정
use case 15 Purpose
use case 15 Overview
use case 15 Pre-Requisites 삭제
use case 15 Typical Courses 3 “name, price”로 변경
use case 15 Exceptional 3-ㄱ, 3-ㄴ, 3-A, 3-B,3-a,3-b 추가
use case 13 Exceptional 10초간 입력이 없을시 삭제

use case 16 Pre-Requisites “관리자모드이어야 한다.”로변경
use case 16 Typical Courses 3,4,5변경
use case 16 Exceptional 전체 변경

use case 17 Pre-Requisites “관리자모드이어야 한다.”로변경
use case 17 Typical Courses 3,4,5변경
use cas 17 Exceptional 전체 변경

use case 18 Pre-Requisites “관리자모드이어야 한다.”로변경
use case 18 Typical Courses 3,4,5변경
use case 18 Exceptional 전체 변경

use case 3 type evident -> hidden
use case 11 type evident -> hidden
use case 11 Typical Courses 1,2 수정

use case 1 Pre-Requisites 삭제
 use case 10 Pre-Requisites '취출구에 ... 있어야한다' 삭제
 use case 10 Pre-Requisites '전원...' 삭제
 use case 19 Pre-Requisites '현재 .. 한다.' 추가
 use case 8 Pre-Requisites 생성될 Precode는 중복되지 않는다. 추가

use case 4 Exceptional 추가
 use case 11 Exceptional 3-A, 3-B 추가
 use case 7 exceptional 1-... -> 2-... 수정

system architecture paymentlineitem card 삭제

2141 Design Real Use Cases

Use case	1. Start Order
Actor	User
Purpose	자판기 사용을 시작한다.
Overview	자판기의 상품을 고르기 위해 카드를 투입한다.
Type	evident
Cross Reference	System Functions : 2.1 Use Case : "Input Choice"
Pre-Requisites	N/A
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
Exceptional Courses of Events	all line-a. (AU) 카드리더기에서 카드를 제거한다. all line-b. (S) 모든 작업을 중단한다 all line-c. (S) 카드가 제거되었다는 메시지를 출력한다 all line-d. (S) Window-1(대기 화면)을 출력한다.

Use case	2.Input Choice
Actor	User
Purpose	사용자가 상품을 선택한다.
Overview	사용자가 선택한 상품의 재고가 현재 자판기에 존재하는지 확인한다.
Type	evident
Cross Reference	System Functions : 2.2, 2.6 Use Case : “Check Chosen Item Stock”, “Payment”
Pre-Requisites	결제 가능한 카드가 투입되어 있어야 한다.
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 존재한다 4. (S) “Payment”로 넘어간다.
Alternative Courses of Events	N/A
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) 카드가 제거되었다는 메시지를 출력한다 all line-d. (S) Window-1(대기화면)을 출력한다.

Use case	3. Check Chosen Item Stock
Actor	System
Purpose	사용자가 선택한 상품의 재고가 다른 자판기에 존재하는지 확인한다.

Overview	다른 모든 자판기에 사용자가 선택한 상품의 재고가 있는지 확인을 요청하고 정보를 받는다.
Type	hidden
Cross Reference	System Functions : 2.3,2.4 Use Case : “Answer Chosen Stock Info”, “Determine Location”
Pre-Requisites	현재 자판기에 상품 재고가 존재하지 않는다.
Typical Courses of Events	(AD) : Actor other DVMs, (S) : System 1. (S) AD의 location, 상품의 quantity를 요청한다. 2. (AD) “Answer Chosen Stock Info”을 통해 위치, 상품 재고 정보를 받는다. 3. (S) “Determine Location”으로 넘어간다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	2-a. (S) 다른 모든 자판기에 재고가 존재하지 않는다. 2-b. (S) 모든 재고 소진 메시지를 출력한다.

Use case	4. Answer Chosen Stock Info
Actor	System
Purpose	자판기의 재고와 위치 정보를 알려준다.
Overview	전달받은 상품의 재고를 확인 후 해당 정보를 자판기 위치와 함께 요청한 자판기에 보내준다.
Type	hidden
Cross Reference	System Functions :2.2 Use Case : “Check Chosen Item Stock
Pre-Requisites	(AD)으로 부터 재고 확인 요청이 있어야한다.
Typical Courses of Events	(AD) : Actor other DVM, (S) : System 1. (S) 요청된 상품의 현재 자판기에서의 재고를 확인한다. 2. (S) 확인한 재고 정보를 자신의 위치정보와 함께 (AD)로 전달한다.

Alternative Courses of Events	N/A
Exceptional Courses of Events	2-a. (S) 재고가 없다. 2-b. (S) null을 반환한다.

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 : 2.5 Use Case : “Determine Prepayment”
Pre-Requisites	재고가 있는 자판기가 존재한다.
Typical Courses of Events	(S) : System 1. (S) 현재 자판기와 재고가 있는 자판기들의 거리를 $ 위도(현재 자판기)-위도(재고 존재 자판기) + 경도(현재 자판기) - 경도(재고 존재 자판기) $ 식을 통해 계산한다. 2. (S) 계산 결과가 가장 작은 자판기를 안내할 자판기로 결정한다. 3. (S) “Determine location”에서 온 재고정보와 위치정보를 Window-4에 출력한다. 4. (S) “Determine Prepayment”로 넘어간다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	6. Determine Prepayment
Actor	User
Purpose	선결제를 한다.
Overview	사용자가 선결제 여부를 선택하고, 선택에 따라 결제를 요청한다.
Type	evident
Cross Reference	System Functions : 2.6 Use Case : "Payment"
Pre-Requisites	재고가 있는 가장 가까운 자판기의 위치 정보와 재고 정보가 현재 자판기에 존재해야 한다.
Typical Courses of Events	(AU) : Actor User, (S): System 1. (AU) 선결제 버튼을 누른다. 2. (S) "Payment"로 넘어간다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1-a. (AU) 취소 버튼을 누른다. 1-b. (S) Window-2(상품 선택 화면)을 출력한다. 1-a. (AU) 카드리더기에서 카드를 제거한다. 1-b. (S) 모든 작업을 중단한다. 1-c. (S) 카드가 제거되었다는 메시지를 출력한다 1-d. (S) Window-1(대기화면)을 출력한다.

Use case	7. Payment
Actor	System
Purpose	결제를 진행한다.
Overview	결제를 진행하고 결제 종류에 따라 다음 행동을 지시한다.
Type	hidden
Cross Reference	System Functions : 2.7, 2.9

	Use Case : “Create Precode”, “Serve Item”
Pre-Requisites	카드가 삽입되어 있어야 한다.
Typical Courses of Events	(S): System 1. (S) Payment 클래스에서 item클래스의 price를 받아온다. 2. (S) 결제를 성공(90% 확률)한다. 3. (S) Payment클래스의 prepayment값을 판단해 선결제인지 아닌지 판단한다. 4. (S) prepayment가 false이다. 5. (S) “Serve Item”을 진행한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	2-a. (S) 결제를 실패(10% 확률)한다. 2-b. (S) 결제 실패 메시지를 출력한다. 2-c. (S) window-1(대기화면)을 출력한다. 3-a. (S) prepayment가 true이다. 3-b. (S) “Create Precode”을 진행한다.

Use case	8. Create Precode
Actor	System
Purpose	선결제 코드를 생성하여 저장한다.
Overview	선결제 코드를 생성하여 저장한다.
Type	hidden
Cross Reference	System Functions : 2.8 Use Case : “Complete Prepayment”
Pre-Requisites	카드가 투입되어 있어야 한다. “Determine Prepayment”가 선행되어야 한다. “Payment”에서 선결제 과정으로 진행되었다 생성될 Precode는 중복되지 않는다.
Typical Courses of Events	(S) : System 1. (S) Precode instance를 생성한다. 2. (S) 결제된 상품의 선결제 코드를

	<p>생성하고 결제가 진행된 DVM에 선결제 코드를 저장한다.</p> <p>3. (S) “Complete prepayment”로 넘어간다.</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	9. Complete Prepayment
Actor	System
Purpose	선결제가 완료되었음을 안내한다.
Overview	선결제 코드와 선결제 코드가 사용 가능한 자판기의 위치를 사용자에게 안내해 선결제를 완료한다.
Type	hidden
Cross Reference	System Functions : 2.7 Use Case : “Create Precode”
Pre-Requisites	재고가 있는 가장 가까운 자판기의 위치 정보와 재고 정보가 현재 자판기에 존재해야 한다. 결제가 되어있어야한다.
Typical Courses of Events	(S) : System 1. (S) “Create Precode”에서 생성된 선결제 코드와 사용가능한 자판기 위치를 Window-6(선결제 완료 화면)에 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	10. Serve Item
Actor	System

Purpose	사용자에게 상품을 배출한다.
Overview	결제된 상품을 제공한다.
Type	hidden
Cross Reference	System Functions : 2.6 Use Case : "Payment"
Pre-Requisites	상품에 대한 결제가 되어있어야한다.
Typical Courses of Events	(S) : System 1. (S) 결제된 상품을 배출한다. 2. 결제된 상품의 재고를 1개 줄인다 3. (S) 3초동안 Window-8(결제 완료 화면)을 출력한다. 4. (S) Window-2(상품 선택화면)을 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	11. Check Precode
Actor	System
Purpose	입력된 선결제 코드와 생성된 선결제 코드의 일치 여부를 확인한다.
Overview	입력된 선결제 코드를 다른 모든 DVM에게 보낸 뒤 일치한다는 답변이 오면 상품을 배출한다.
Type	hidden
Cross Reference	System Functions : 2.11, 2.12 Use Case : "Answer Precode Info", "Input Precode"
Pre-Requisites	N/A
Typical Courses of Events	(AD) : Actor other DVMs , (S) : System 1. (S) "Input Precode"에서 입력받은 선결제 코드를 (AD)에게 보낸다. 2. (S) "Answer Precode Info"를 통해 (AD)의 선결제 코드 발급 유무를 전달받는다. 3. (S) 선결제 코드가 올바른 경우 "Serve

	Item”을 진행한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>3-a. (S) 입력한 선결제 코드와 일치하는 선결제 코드가 존재하지 않는다.</p> <p>3-b. (S) 잘못된 선결제 코드가 입력되었다고 메시지를 출력한다.</p> <p>3-c. (S) Window-3(선결제 입력 화면)을 출력한다.</p> <p>3-ㄱ. (S) 선결제 코드 사용을 진행하는 DVM과 선결제시 안내된 DVM이 다르다.</p> <p>3-ㄴ. (S) 현재 DVM에서 선결제코드를 사용할 수 없다는 메시지를 출력한다.</p> <p>3-ㄷ. (S) Window-3(선결제 입력 화면)을 출력한다.</p> <p>3-A. (S) 선결제 코드가 올바르지만, 재고가 없다.</p> <p>3-B. (S) 재고가 없음 메시지를 출력한다.</p>

Use case	12. Answer Precode Info
Actor	System
Purpose	선결제 코드의 일치를 확인한다.
Overview	다른 DVM이 확인을 요청받은 선결제 코드가 발급된 적이 있는 선결제 코드인지 응답한다.
Type	hidden
Cross Reference	System Functions : 2.10 Use Case : “Check Precode”
Pre-Requisites	“Check Precode”을 통해 요청받아야 한다.
Typical Courses of Events	<p>(S) : System</p> <ol style="list-style-type: none"> 1. (S) “Check Precode”을 통해 요청된 선결제 코드를 확인한다. 2. 요청된 코드와 발급된 코드가 일치한다. 3. (S) 선결제 코드가 일치함을 알린다.
Alternative Courses of Events	N/A

Exceptional Courses of Events	2.a (S) 일치하지 않다 2.b (S) 발급된적이 없는 선결제 메시지를 출력한다.
-------------------------------	--

Use case	13. Input Precode
Actor	User
Purpose	선결제 코드를 입력한다.
Overview	선결제 코드 입력을 선택 후 코드를 입력한다.
Type	evident
Cross Reference	System Functions : 2.10, 2.11 Use Case : "Check Precode", "Answer Precode Info"
Pre-Requisites	N/A
Typical Courses of Events	(AU) : Actor User, (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
Exceptional Courses of Events	3-a. (AU) Window-3(선결제 입력 화면)의 취소버튼을 누른다. 3-b. (S) Window-1(대기 화면)을 출력한다.

Use case	14. Enter Administrator Mode
Actor	Manager

Purpose	관리자모드로 진입한다.
Overview	관리자 인증 수단을 통해서 관리자모드로 진입한다.
Type	evident
Cross Reference	N/A
Pre-Requisites	N/A
Typical Courses of Events	(AM) : Actor Manager, (S) : System 1. (AM) 비밀번호를 입력하여 (S)에게 인증을 요청한다. 2. (S) 인증에 성공한다. 3. (S) Window-12(관리자모드)를 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	2-a. (S) 인증에 실패한다. 2-b. (S) 잘못된 입력메세지를 출력한다.

Use case	15. Add Itemlist
Actor	Manager
Purpose	상품 목록에서 상품을 추가한다.
Overview	상품 목록에서 상품을 추가한 후에 갱신한다.
Type	evident
Cross Reference	System Functions : 3.1 Use Case : "Enter administrator mode"
Pre-Requisites	N/A
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 name, price를 입력한 후 확인버튼을 누른다. 4. (S) Item 목록을 갱신한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1-a. (S) Window-12(관리자 모드 화면)에서 상품 목록 변경의 범위가 지정된 범위(20 종류)를

	<p>벗어난다</p> <p>1-b. (S) 상품 추가 불가 메시지를 출력한다.</p> <p>3-(ㄱ) (AM)Window-13(상품 목록 추가 화면)에서 상품명 또는, 가격을 입력안한 채로 확인버튼을 누른다</p> <p>3-(ㄴ) (S)잘못된 입력 메시지를 출력한다.</p> <p>3-(A) (AM)Window-13(상품 목록 추가 화면)에서 가격에 음수 또는 5만원 초과인 가격을 입력한 채로 확인버튼을 누른다</p> <p>3-(B) (S)잘못된 입력 메시지를 출력한다</p> <p>3-a. (AM) Window-13(상품 목록 추가 화면)에서 취소버튼을 누른다.</p> <p>3-b. (S) 관리자 모드 화면을 출력한다.</p>
--	---

Use case	16. Remove Itemlist
Actor	Manager
Purpose	상품 목록에서 상품을 삭제한다.
Overview	상품 목록에서 상품을 삭제한 후에 갱신한다.
Type	evident
Cross Reference	System Functions : 3.1 Use Case : "Enter administrator mode"
Pre-Requisites	현재 관리자 모드이어야 한다.
Typical Courses of Events	<p>(AM) : Actor Manager, (S) : System</p> <ol style="list-style-type: none"> (AM) Window-12(관리자 모드 화면)의 삭제버튼을 누른다. (S) Window-14(상품 목록 삭제 화면)을 출력한다. (AM) Window-14의 삭제할 상품 버튼을 선택한다. (S) Window-15(삭제 확인 화면)을 출력한다. (AM) Window-15의 확인 버튼을 누른다. (S) Item 목록을 갱신한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	1-a. (S) 상품 목록 변경의 범위가 지정된 범위(1종류 이상)를 벗어난다.

	1-b. (S) 상품 삭제 불가 메시지를 출력한다.
--	------------------------------

Use case	17. Change Stock & Price
Actor	Manager
Purpose	상품의 재고를 변경한다.
Overview	상품 재고를 변경한 것을 시스템에 반영시킨다.
Type	evident
Cross Reference	System Functions : 3.1 Use Case : "Enter administrator mode"
Pre-Requisites	현재 관리자 모드이어야 한다.
Typical Courses of Events	(AM) : Actor Manager, (S) : System 1. (AM) 상품을 선택한다. 2. (S) 해당 상품의 가격&재고 변경 화면을 출력한다. 3. (AM) 상품의 재고와 가격을 변경한 후 확인버튼을 누른다. 4. (S) 변경된 재고와 가격이 지정된 범위 내(재고는 1이상 100 이하, 가격은 0 초과, 50000이하)에 존재한다. 5. (S) 상품의 재고와 가격을 갱신한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	3-a. (AM) 상품 가격&재고변경 화면에서 취소 버튼을 누른다. 3-b. (S) 관리자 모드를 출력한다. 4-a. (S) 변경된 재고와 가격이 지정된 범위(재고는 1 이상 100이하, 가격은 0 초과 50000 이하)를 벗어난다. 4-b. (S) 잘못된 입력 메시지를 출력한다.

Use case	18. Check Sales History
----------	-------------------------

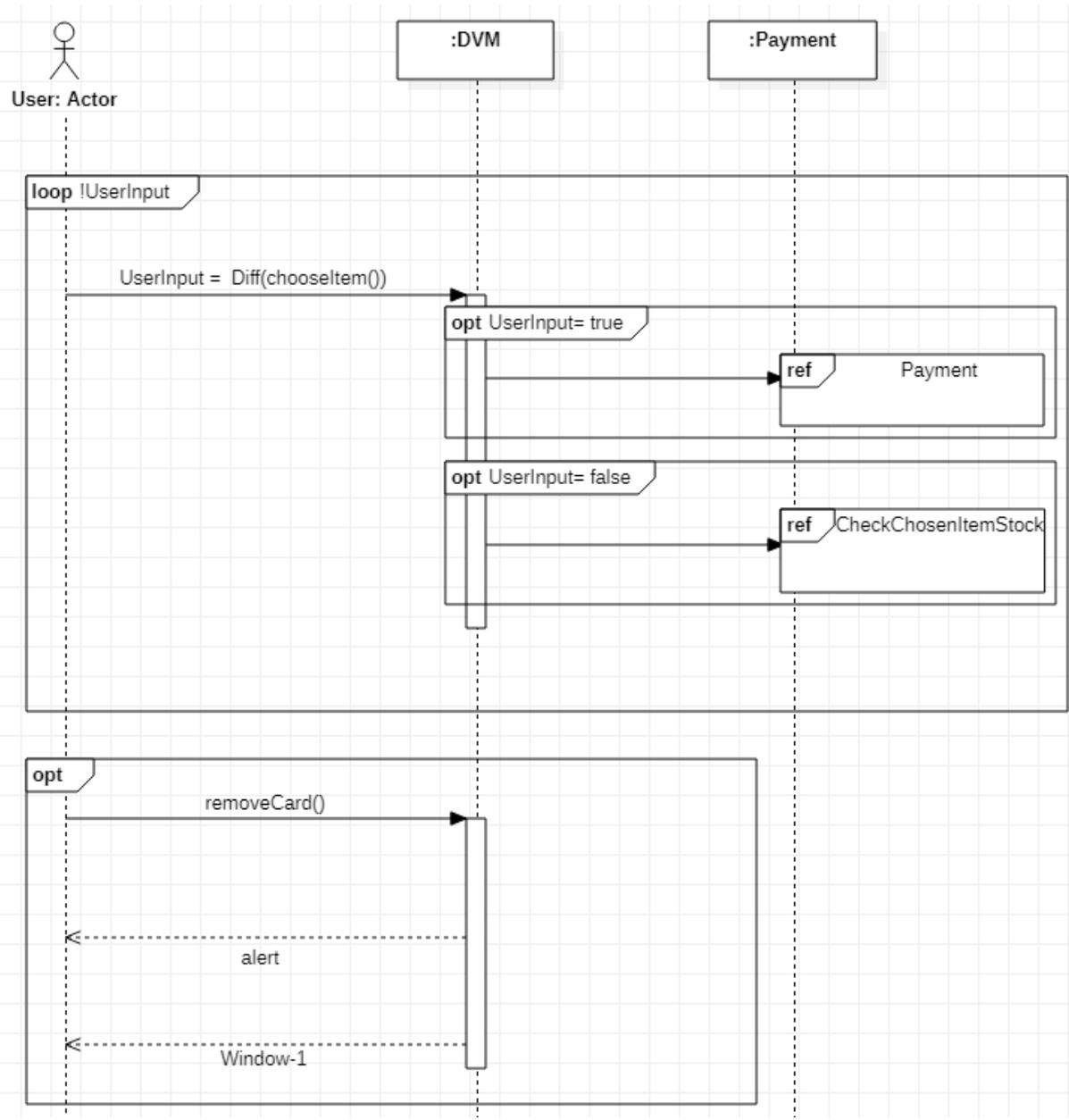
Actor	Manager
Purpose	판매내역을 확인한다.
Overview	상품의 상품명, 판매 수량, 총 판매 금액을 출력한다.
Type	evident
Cross Reference	System Functions : 3.1 Use Case : "Enter administrator mode"
Pre-Requisites	현재 관리자 모드이어야 한다.
Typical Courses of Events	(AM) : Actor Manager, (S) : System 1. (AM) Window-12(관리자 모드 화면)에서 판매 내역 확인 버튼을 누른다. 2. (S) Window-17(판매 내역 확인) 화면을 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

2142 Define Reports, UI, Storyboards

별도의 google slides에 작성

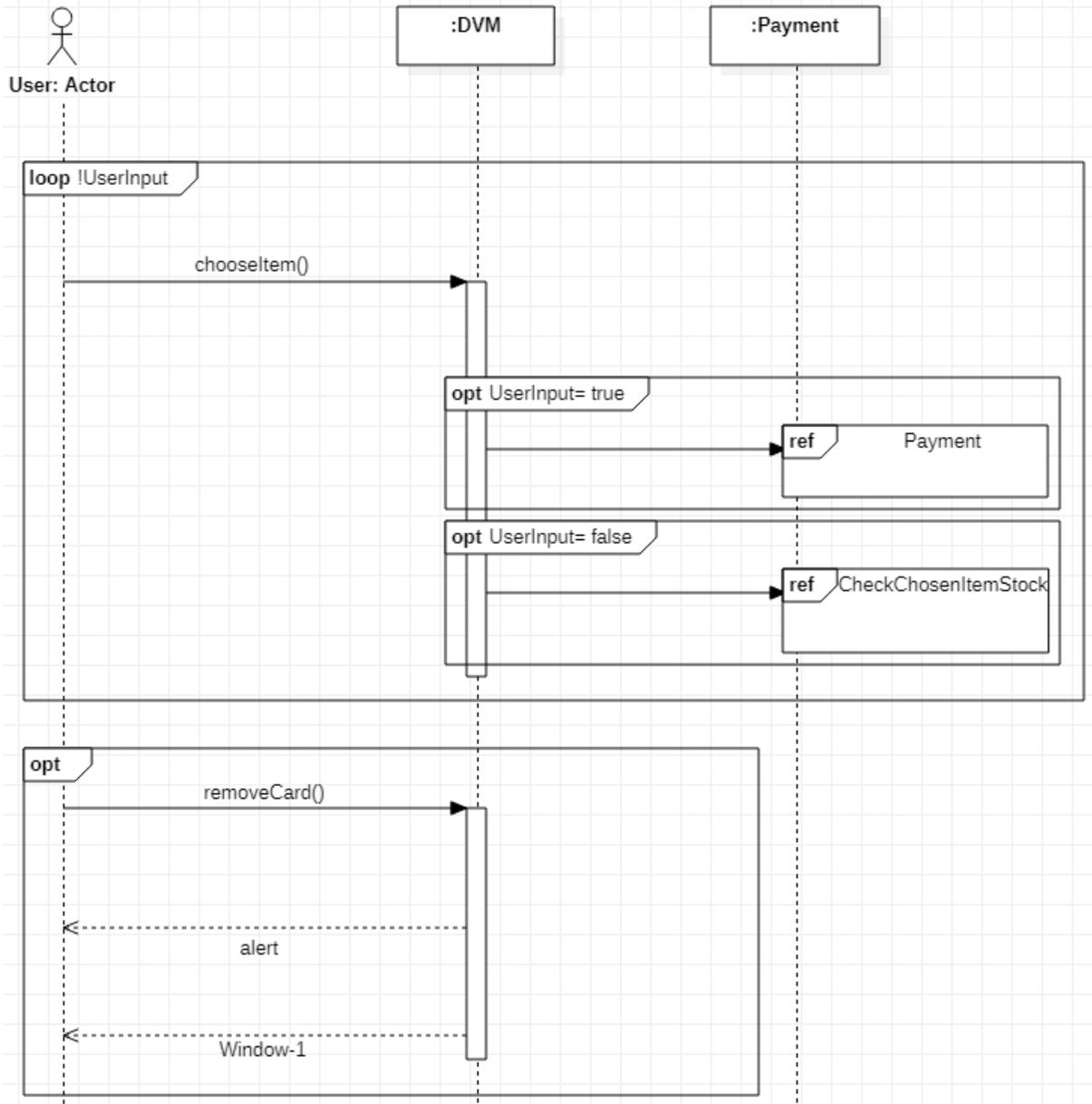
2143 Define Reports Interaction Diagrams

1. Start Order 변경

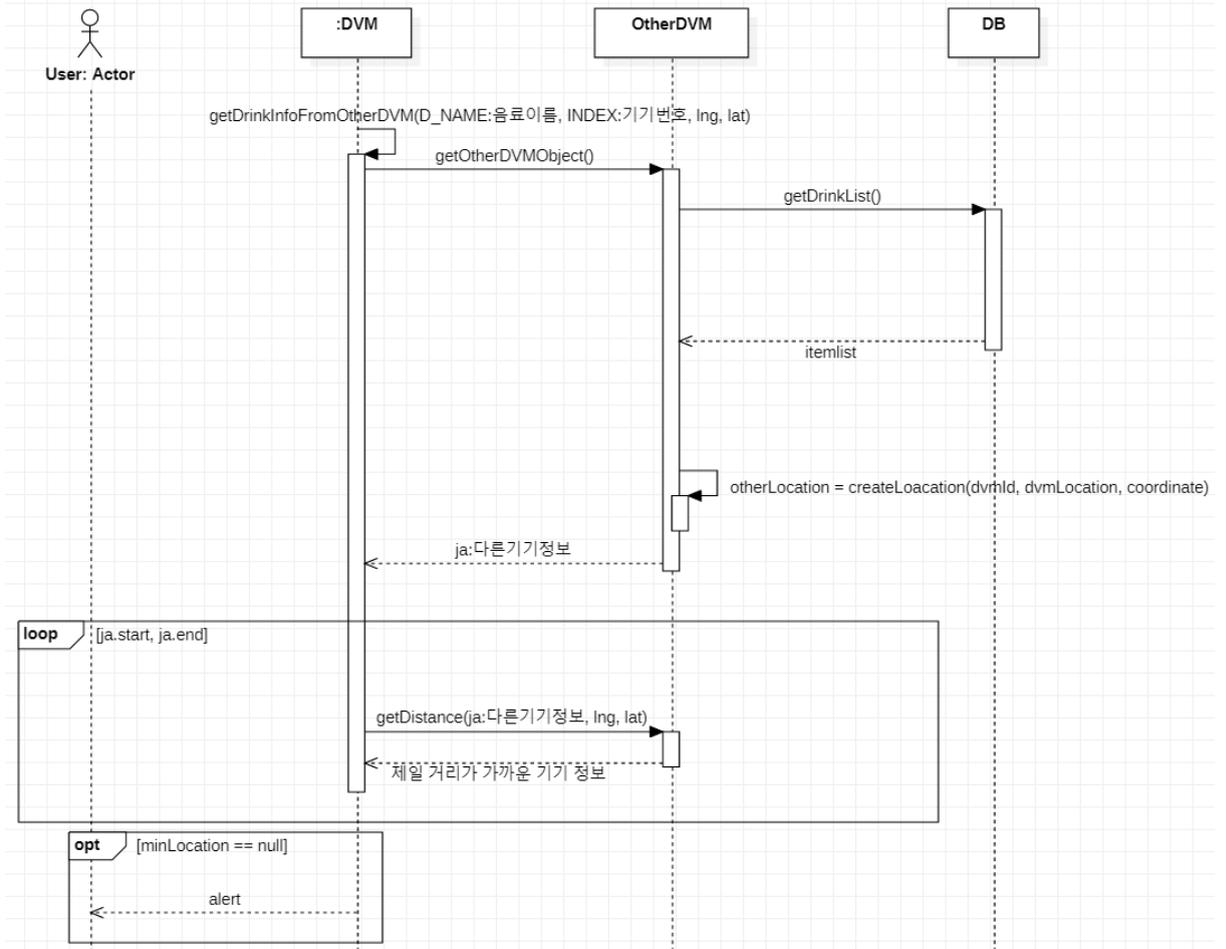


2. Input Choice

변경



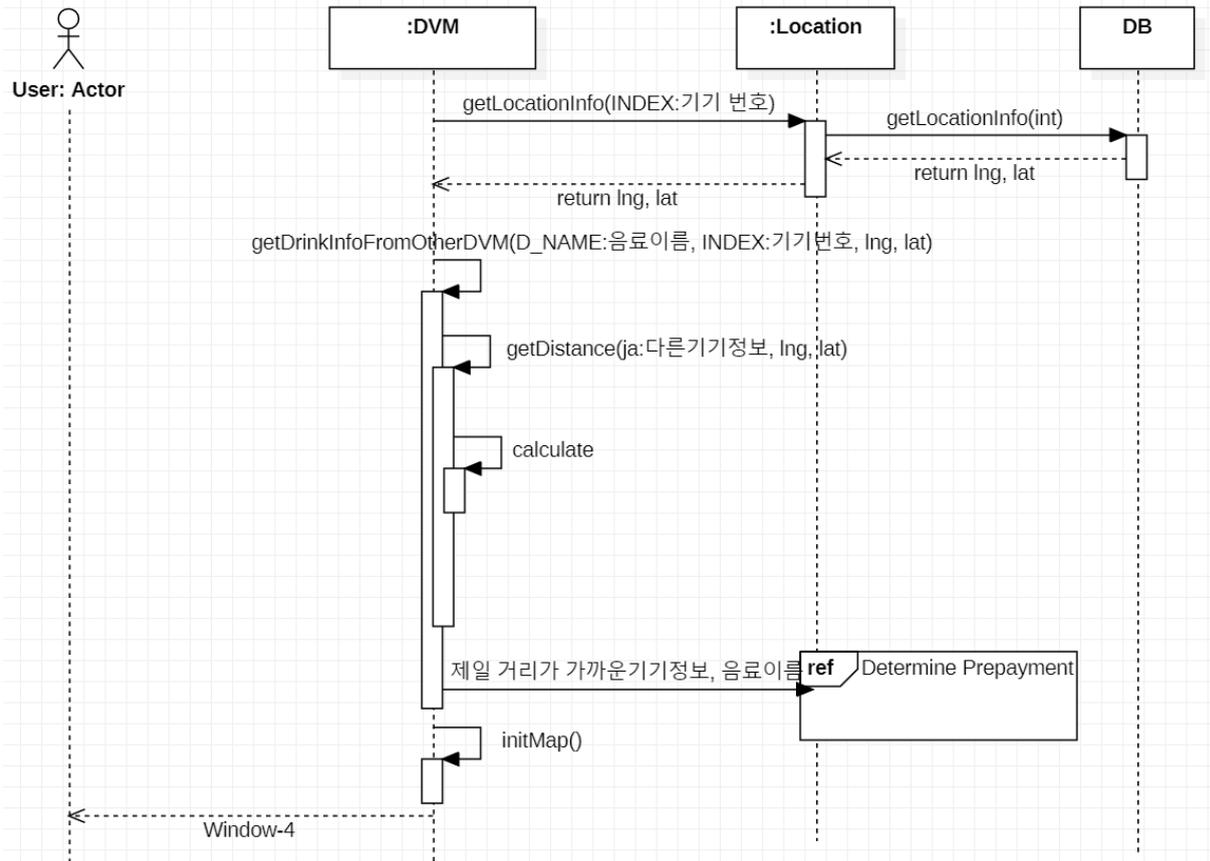
3. CheckChosenItemStock & 4. Answer Chosen Stock Info 변경



4. //

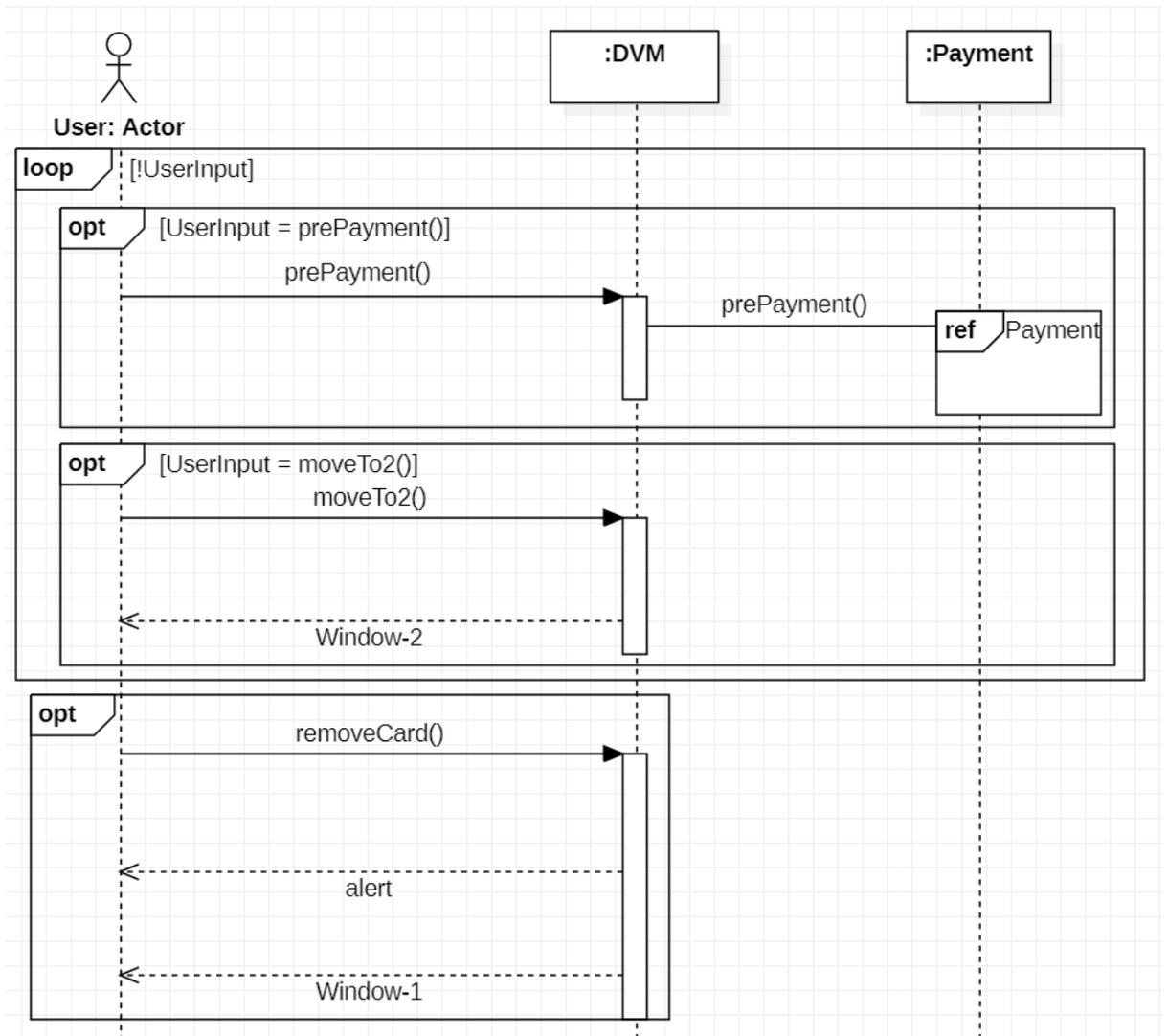
5. Determine Location

변경2



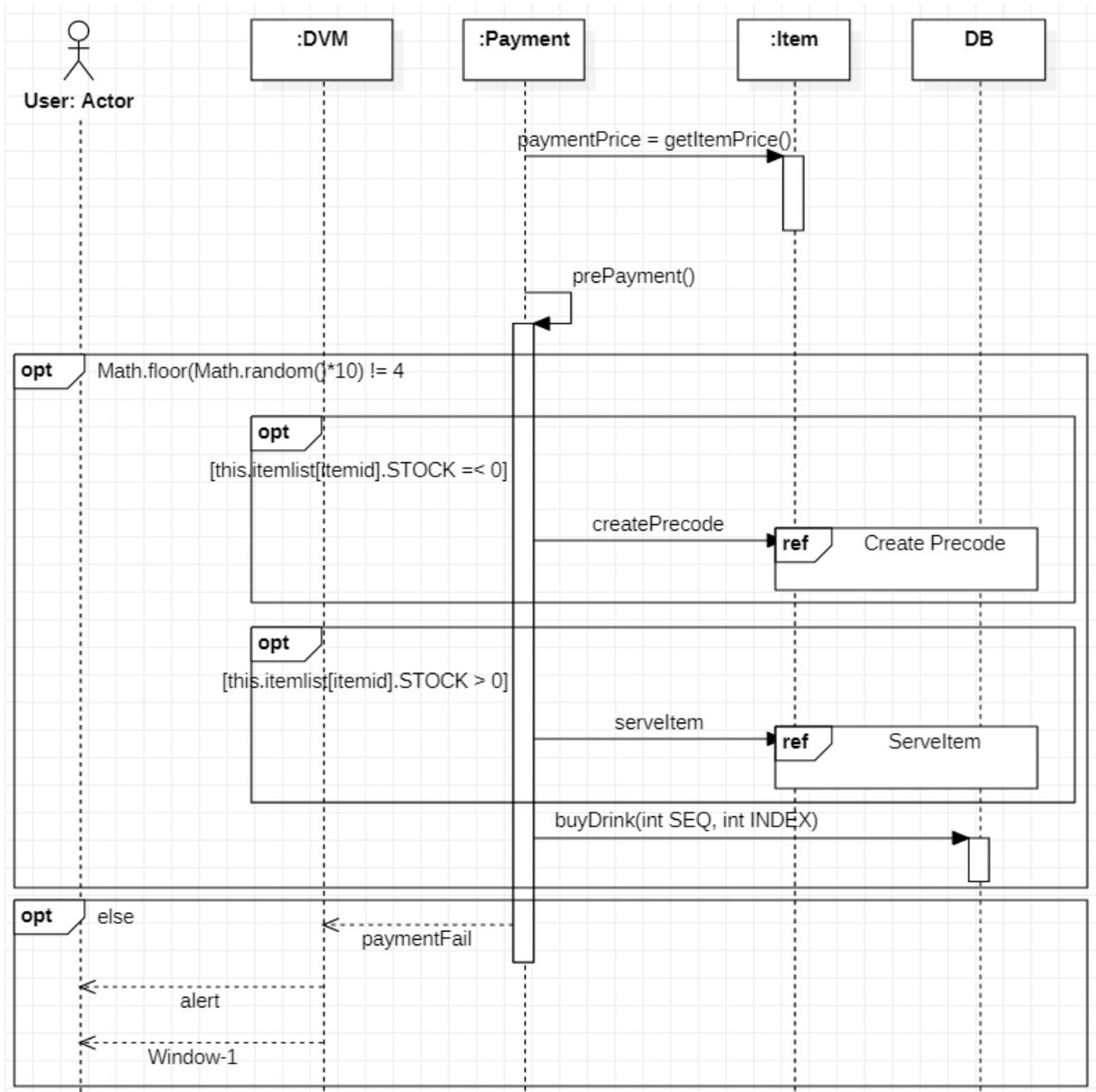
6. Determine Prepayment

변경2



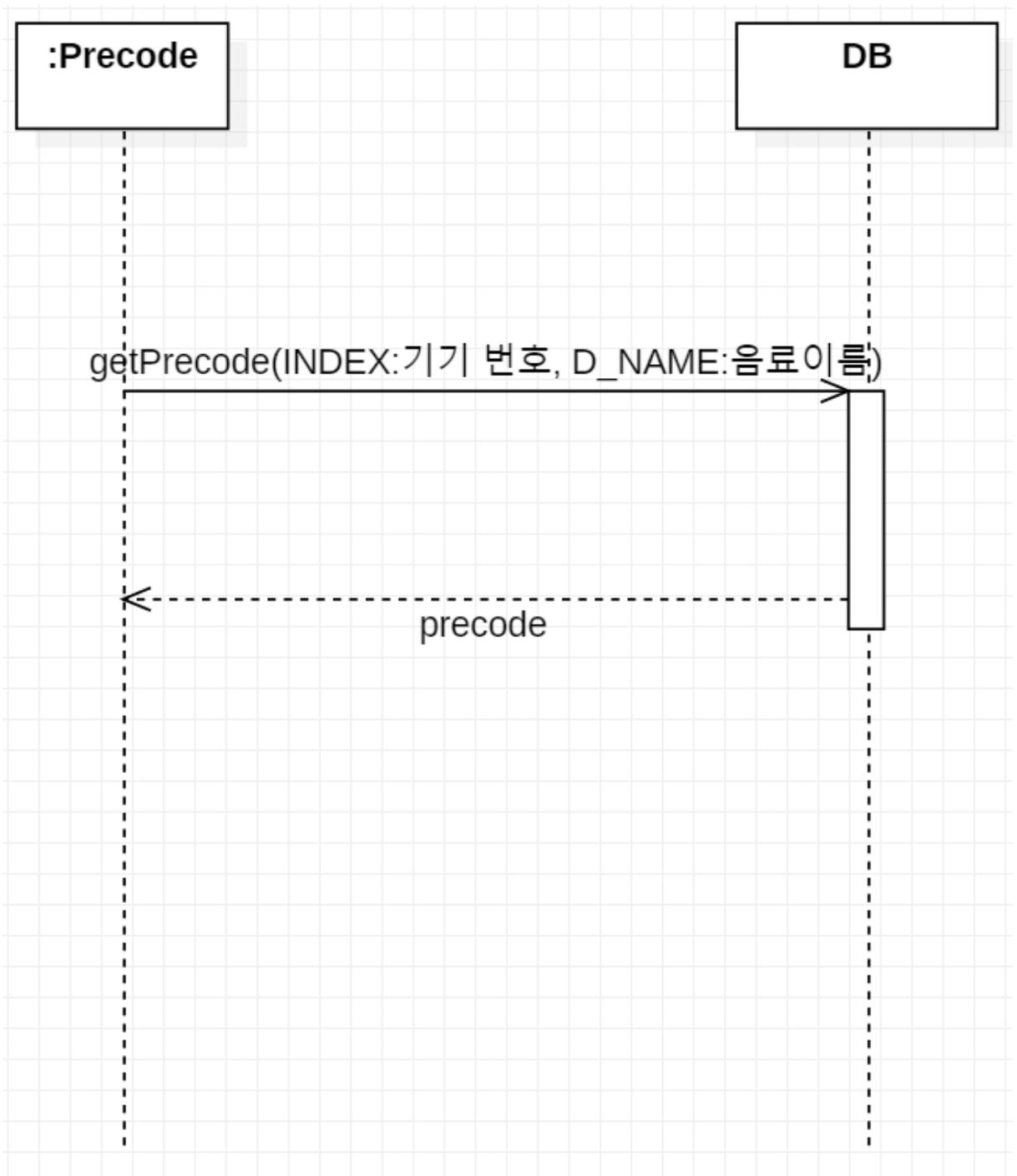
7. Payment

변경

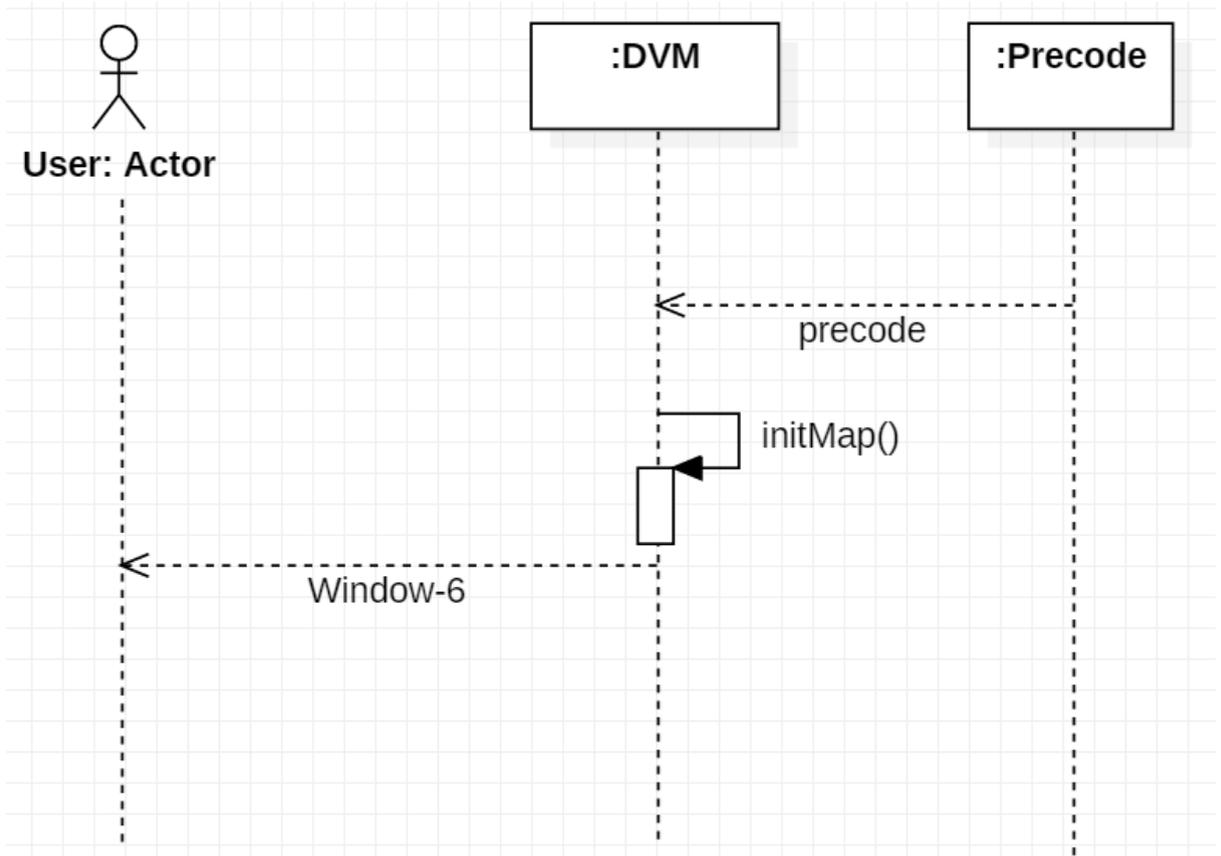


8. Create Precode

변경

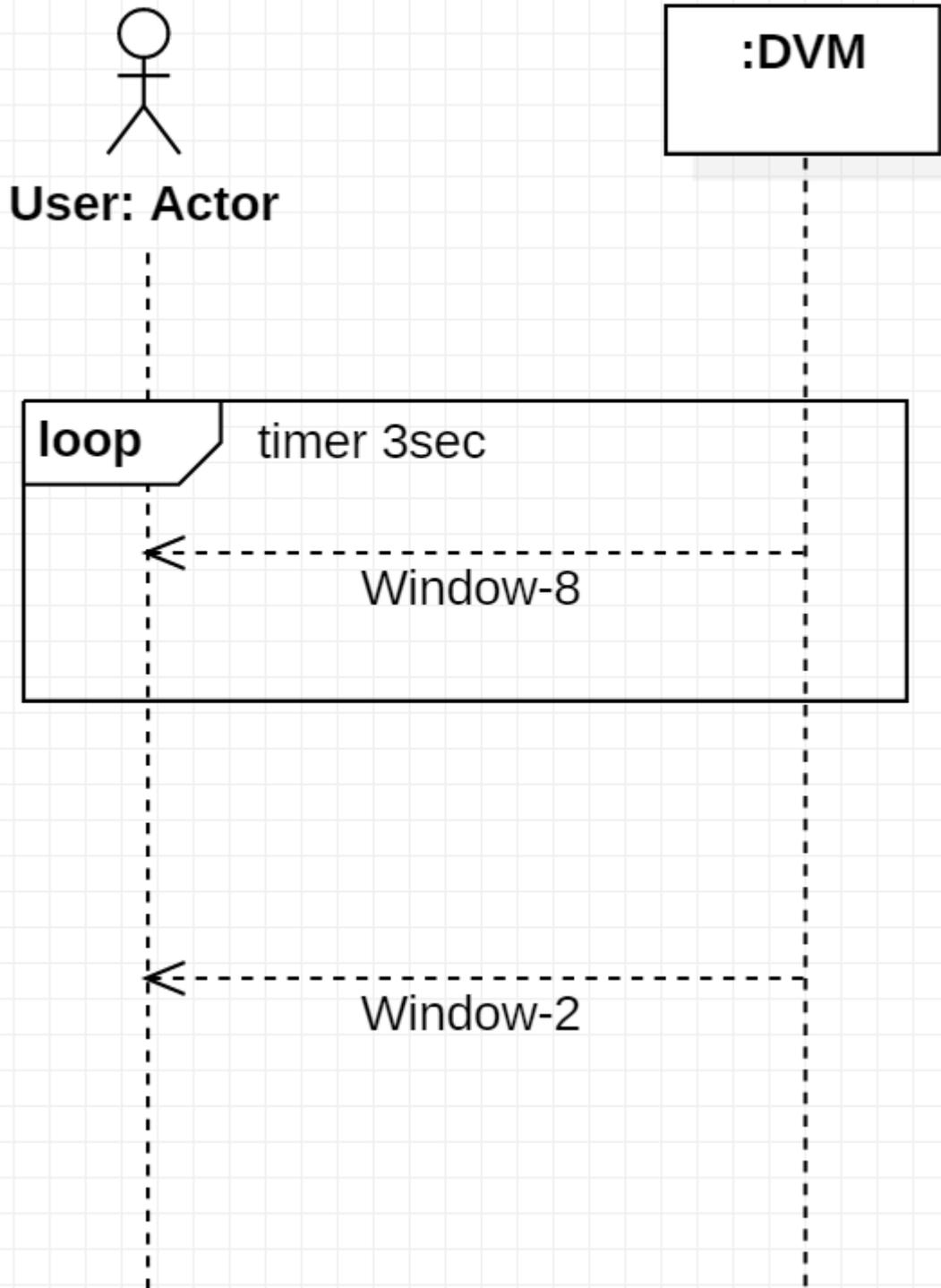


9. Complete Prepayment

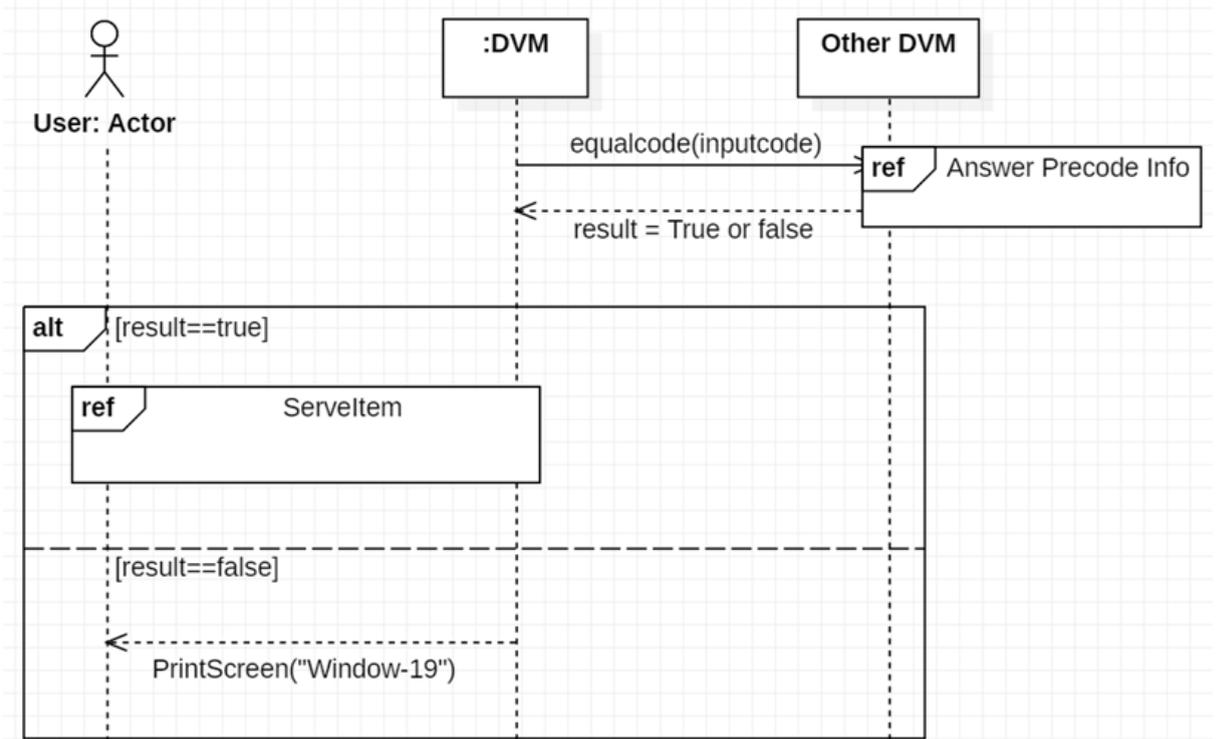


10. Servletem

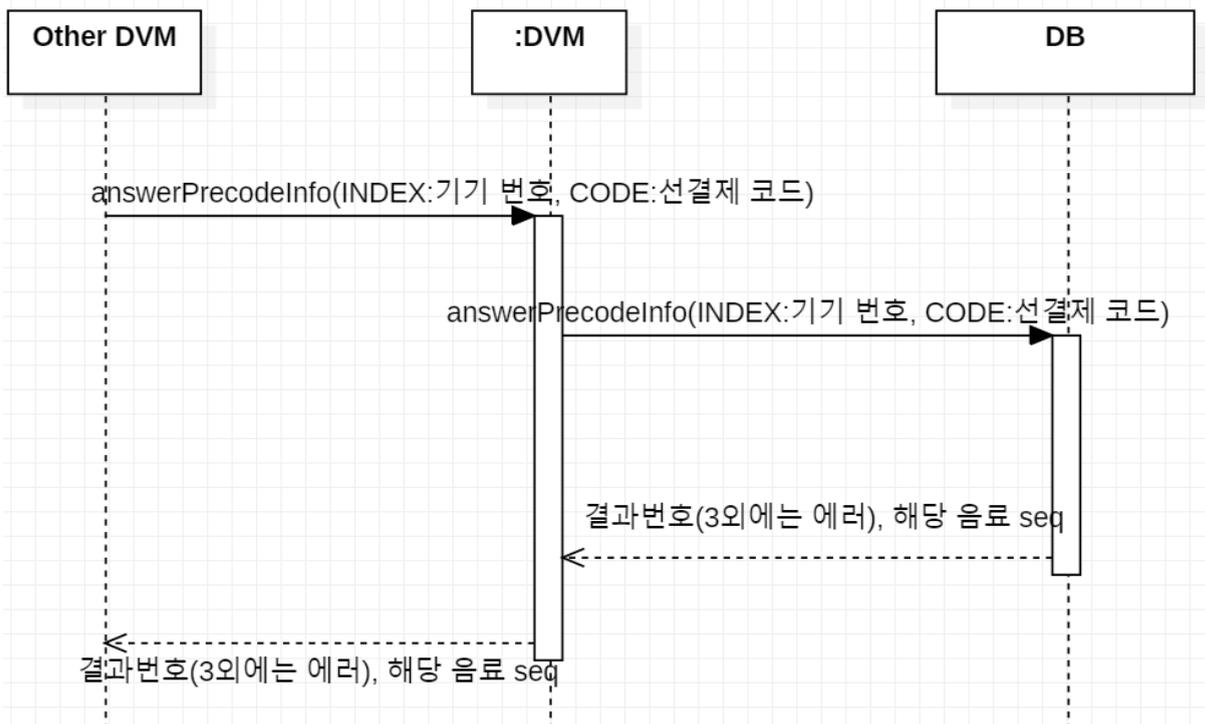
변경



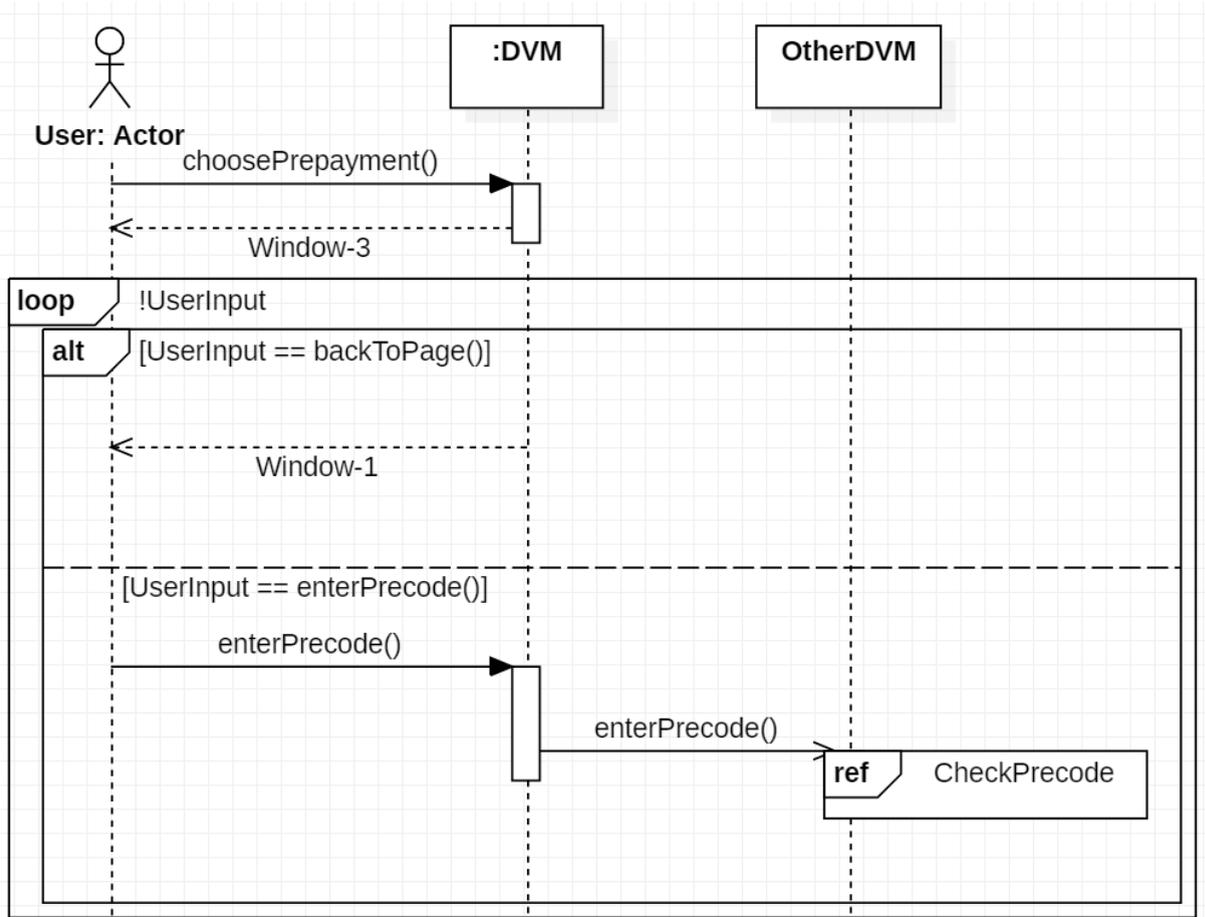
11. CheckPrecode



12. Answer Precode Info 변경

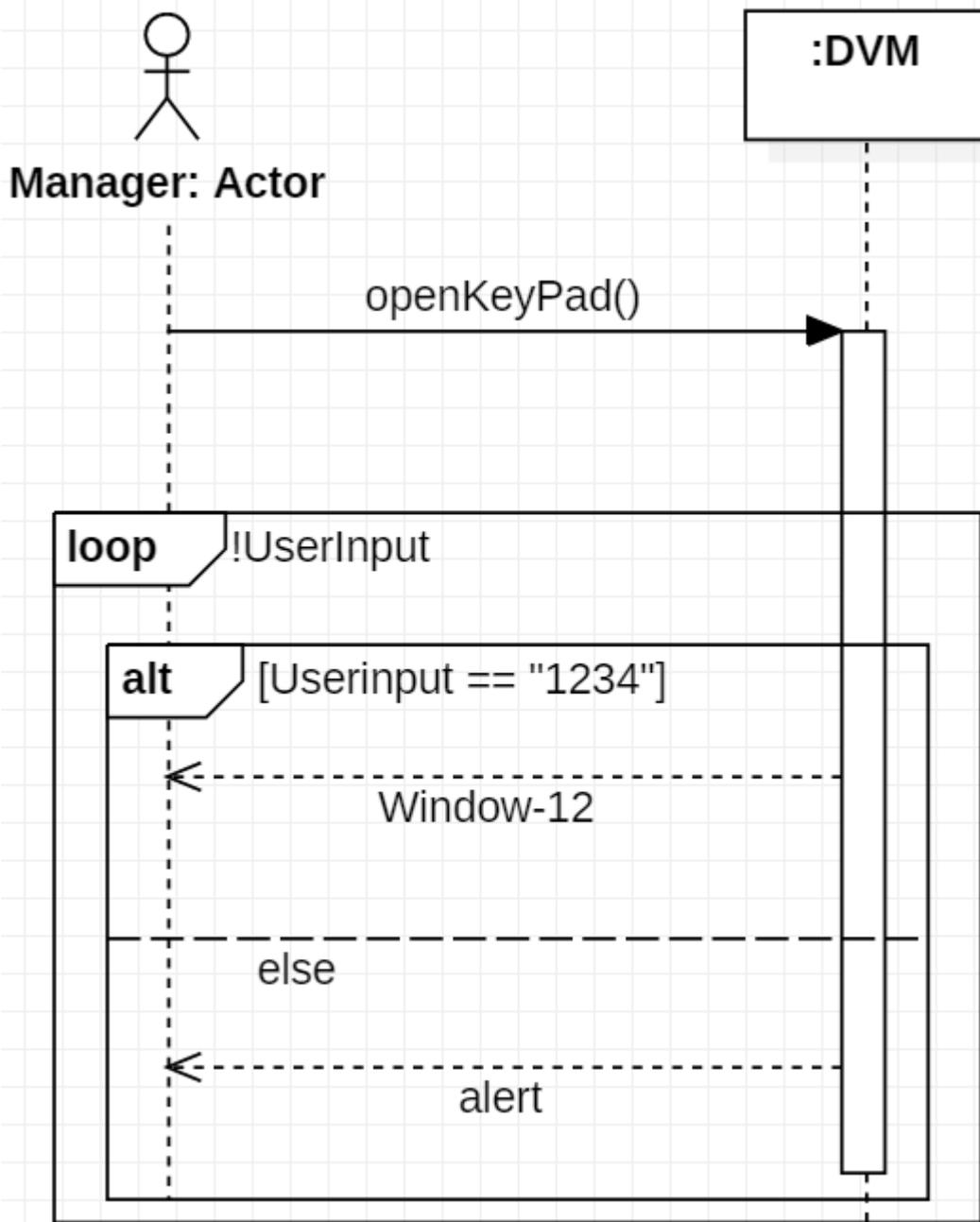


13. Input Precode 변경

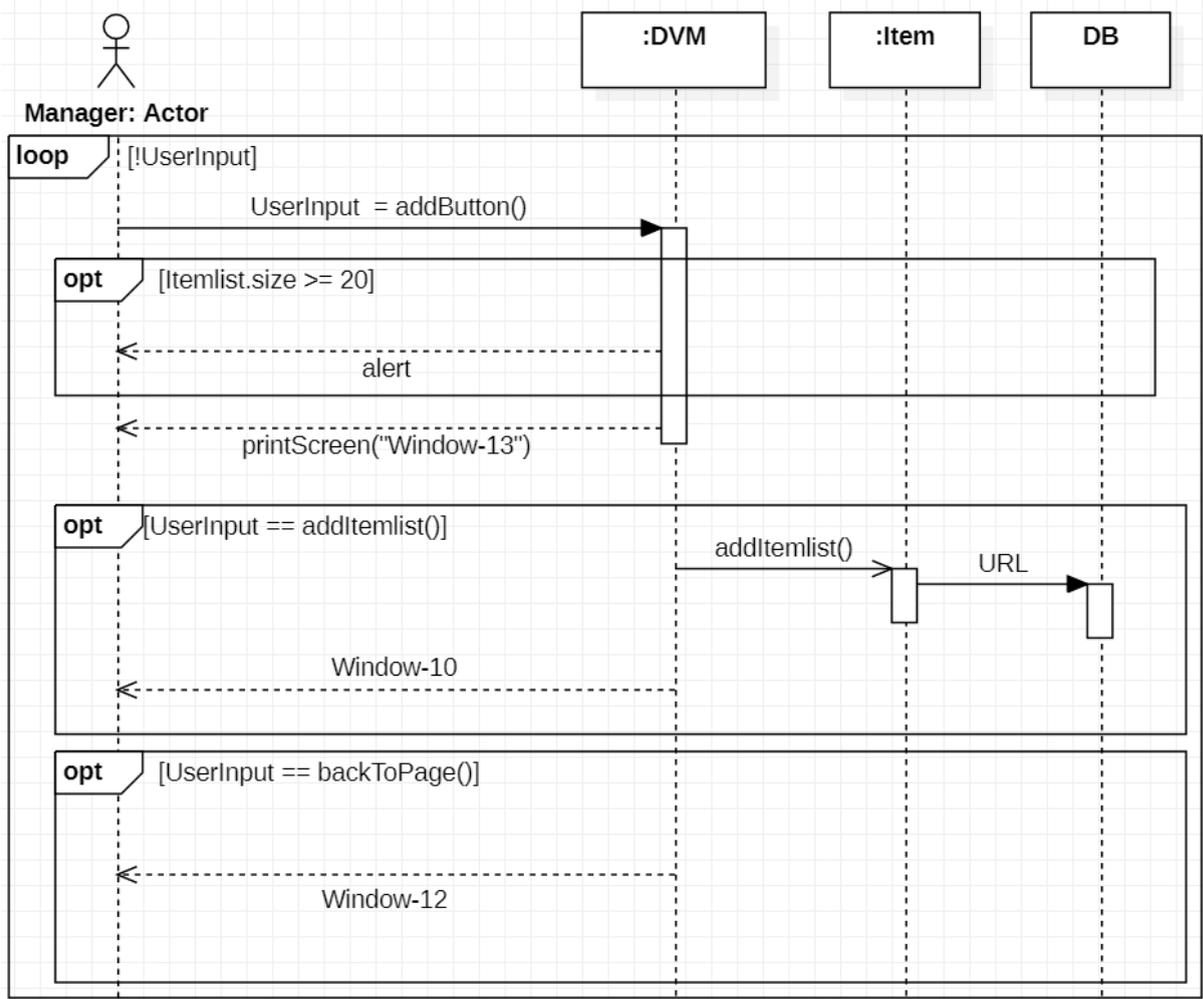


14. Enter Administrator Mode

변경

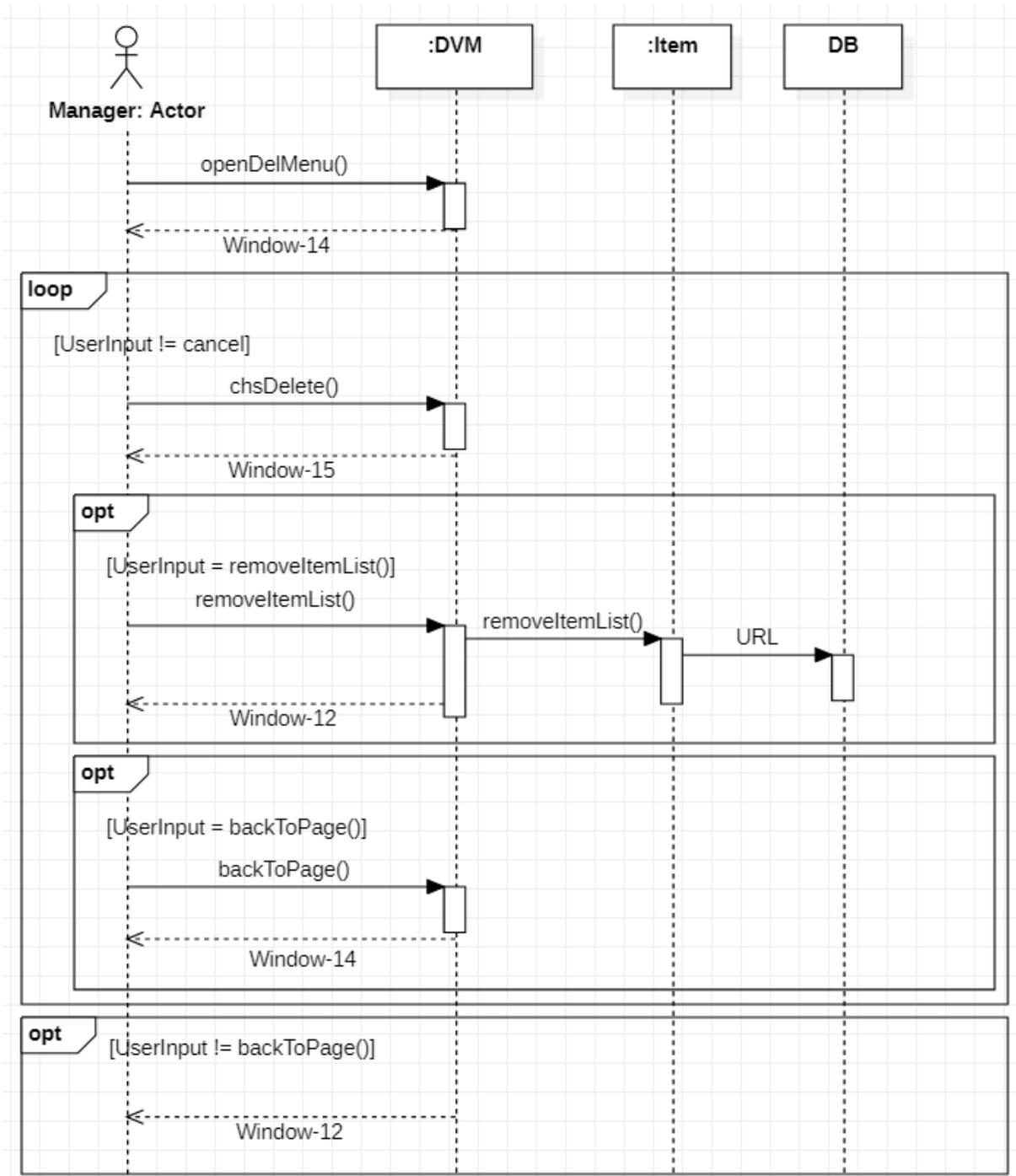


15. Add Itemlist 변경

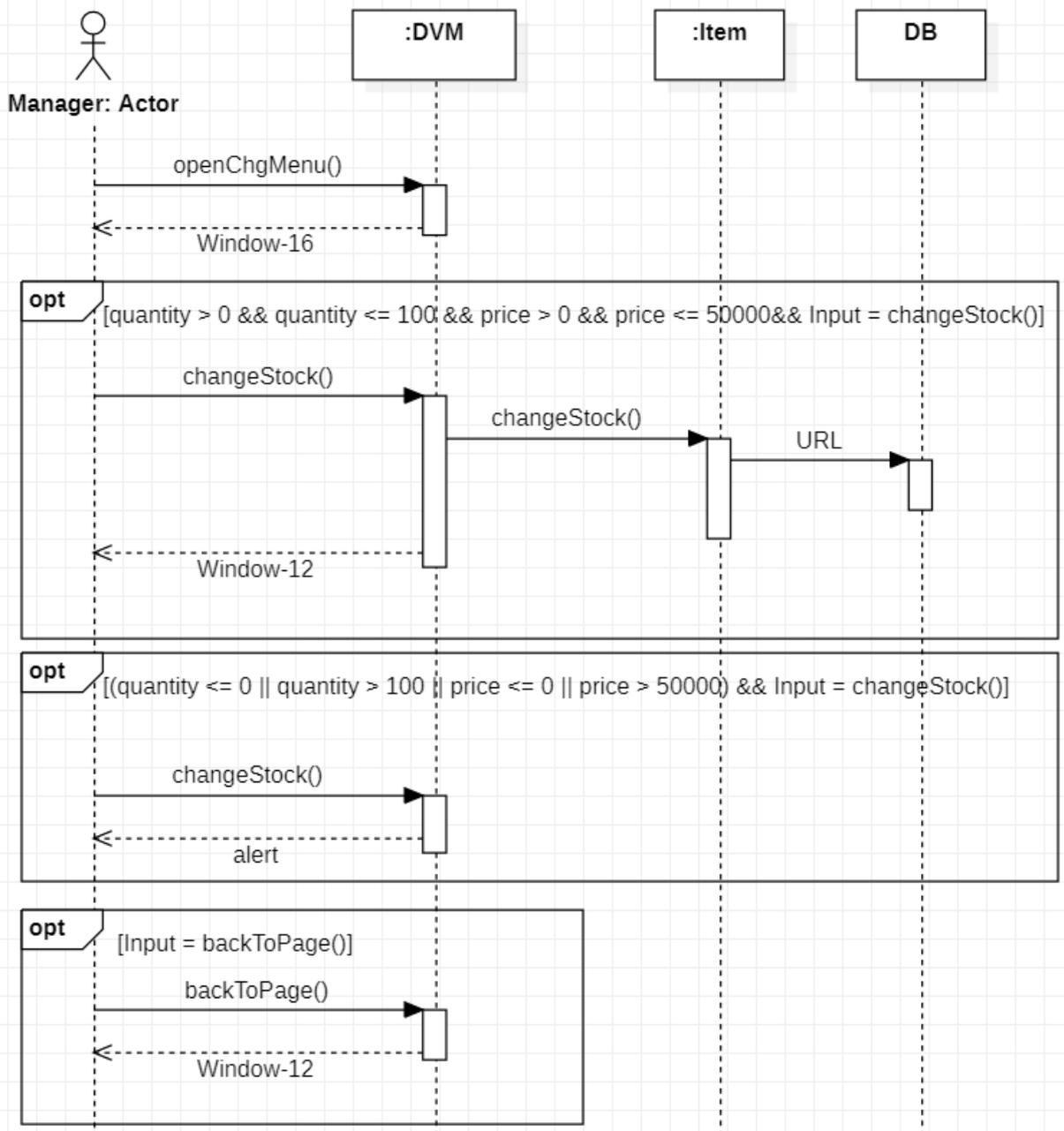


16. Remove Itemlist

변경

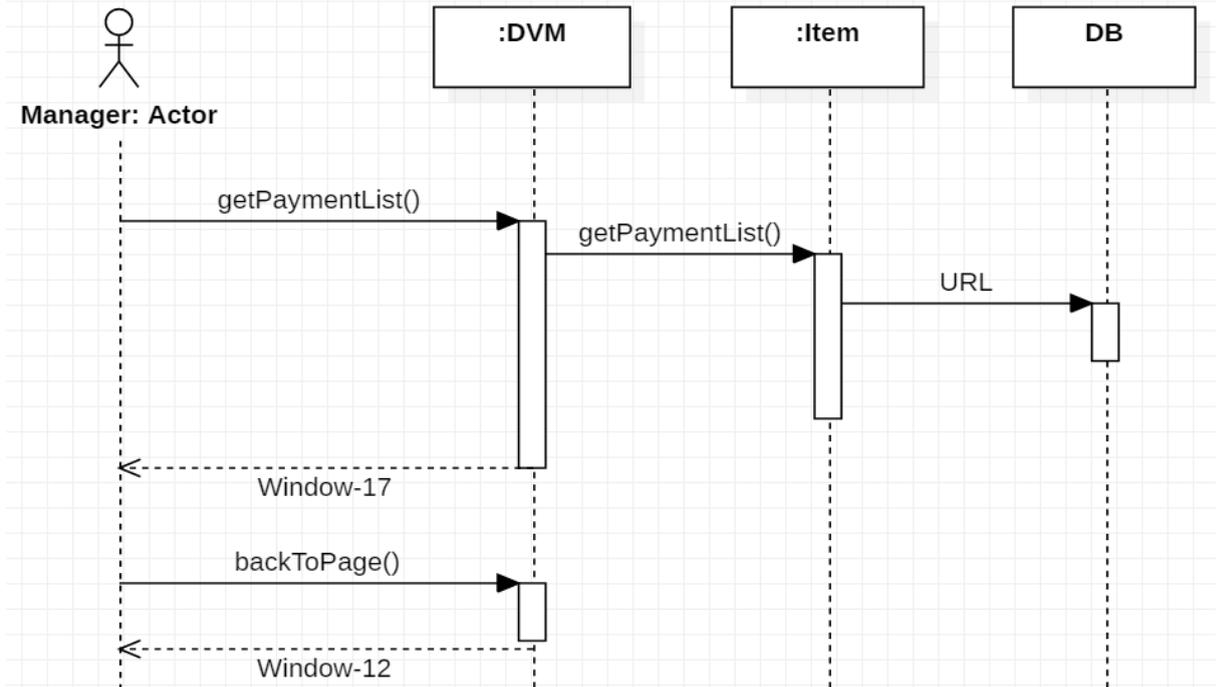


17. Change Stock & Price

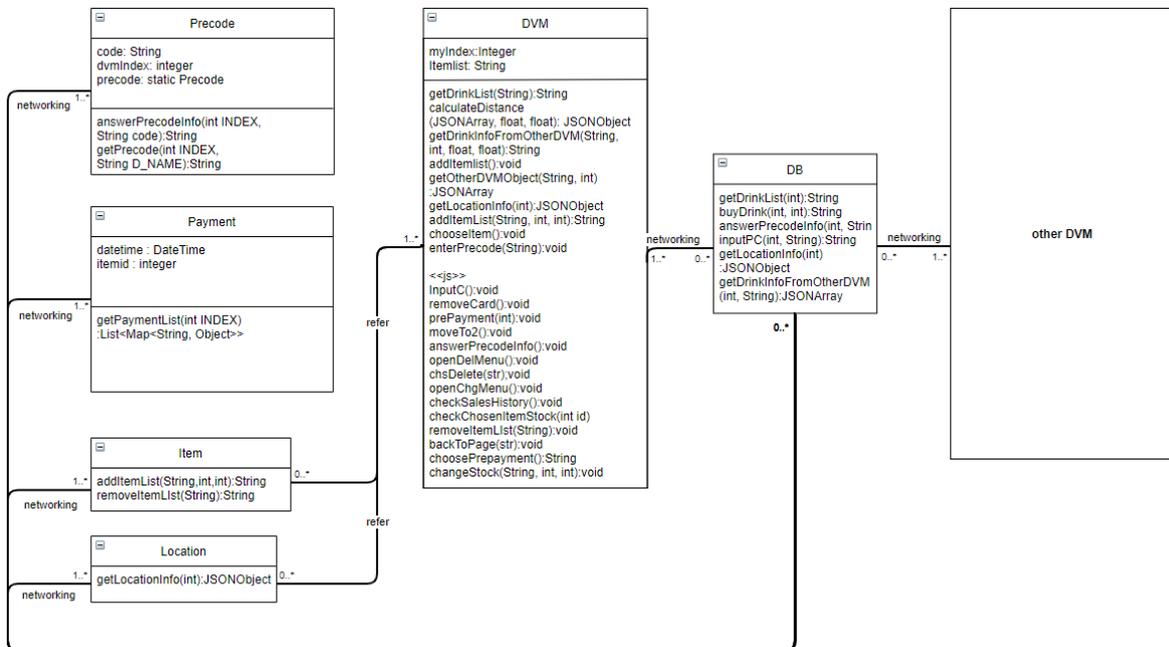


18. Check Sales History

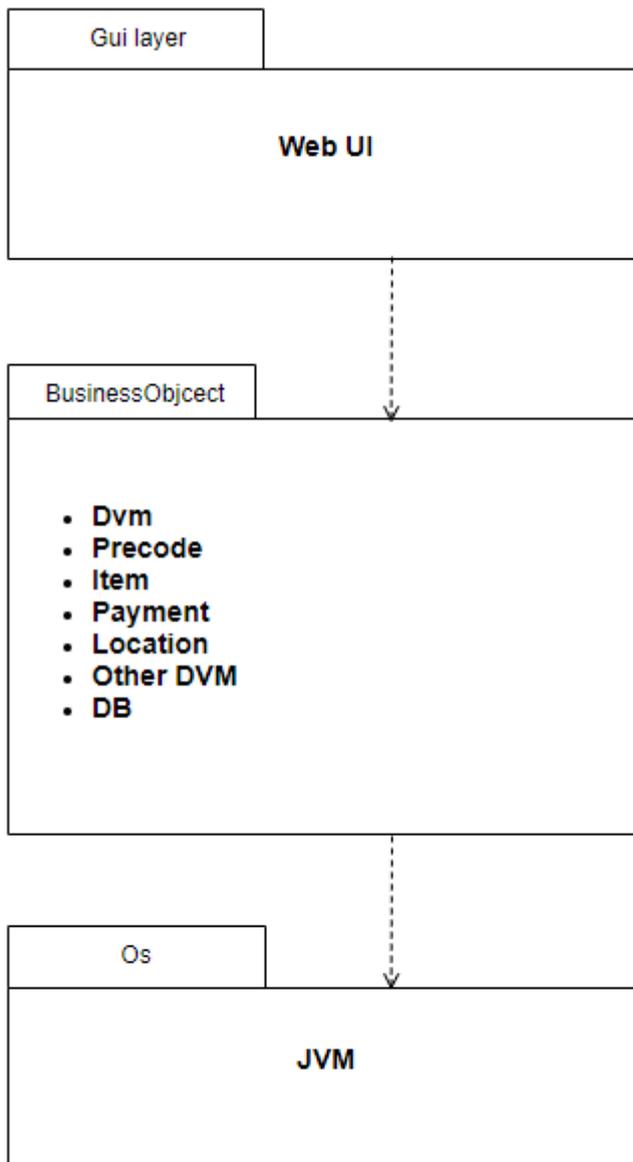
변경



2144 Define Design Class Diagrams



2145 Refine System Architecture



2146 Define Database Schema (x)

2147 Perform 2040 Traceability Analysis

System Function	Essential Use Case	System Operation	Method	class
start order	Start Order	inputC()	getDrinkList(String):String	DVM
input choice	Input Choice	chooseItem()	calculateDistance(JSONArray, float, float): JSONObject	
check chosen item stock	Check Chosen Item Stock	prePayment()	getDrinkInfoFromOtherDVM(String, int, float, float):String	
answer chosen item stock	Answer Chosen Stock Info	choosePrepayment()	addItemList():void	
determine location	Determine Location	enterPrecode()	getOtherDVMObject(String, int):JSONArray	
determine prepayment	Determine Prepayment	openKeyPad()	getLocationInfo(int):JSONObject	
payment	Payment	addButton()	addItemList(String, int, int):String	
create precode	Create Precode	addItemList()	inputC():void	
complete prepayment	Complete Prepayment	openDelMenu()	removeCard():void	
serve item	Serve Item	chsDelete()	prePayment(int):void	
check precode	Check Precode	removeItemList()	moveTo2():void	
answer precode info	Answer Precode Info	openChgMenu()	answerPrecodeInfo():void	
input precode	Input Precode	changeStock()	openDelMenu():void	
enter administrator mode	Enter Administrator mode	getPaymentlist()	chsDelete(str):void	
add itemlist	Add Itemlist		openChgMenu():void	
remove itemlist	Remove Itemlist		checkSalesHistory():void	
change stock & price	Change Stock&Price		checkChosenItemStock(int id)	
check sales History	Check Sales History		removeItemList(String):void	
			backToPage(str):void	
			choosePrepayment():String	
			changeStock(String, int, int):void	
			answerPrecodeInfo(int INDEX, String code):String	Precode
			getPrecode(int INDEX, String D_NAME):String	
			getPaymentList(int INDEX) (List<Map<String, Object>>	Payment
			addItemList(String,int,int):String	Item
			removeItemList(String):String	
			getLocationInfo(int):JSONObject	Location
			getDrinkList(int):String	DB
			buyDrink(int, int):String	
			answerPrecodeInfo(int, String):String	
			inputPC(int, String):String	
			getLocationInfo(int)	
			getDrinkInfoFromOtherDVM (int, String):JSONArray	