

2043.Design Real Use Cases

Use Case	R0.1 메뉴 화면 선택
Actor	user, admin
Overview	관리자 페이지 혹은 구매 페이지로 이동한다.
Type	Evident
Cross Reference	System Functions : R1.1, R2.1, R1.5, R2.2 Use cases : 음료 종류 선택, 관리자 페이지 로그인, 관리자 페이지 로그아웃, 인증코드 입력
Pre-Requisites	N/A
Typical Courses of Events	(U) : 사용자, (S) 시스템, (A) 관리자 1. (S) 시스템이 메뉴 화면 페이지를 띄워준다. 2. (U) 사용자는 구매 페이지를 고른다. 3. (S) 시스템은 구매 페이지로 이동한다. invoke R1.1
Alternative Courses of Events	A 1. (A) 관리자 로그인 화면을 선택할 시 R2.1로 분기 A 2. (A) 인증 코드 페이지를 선택할 시 R1.5로 분기
Exceptional Courses of Events	N/A

Use Case	R1.1. 음료 종류 선택
Actor	user
Overview	사용자가 음료를 선택한다.
Type	Evident
Cross Reference	System Functions : R0.1, R1.2, R1.3, R1.4 Use Cases : “메뉴화면 선택”, “현재 자판기 음료 개수 선택”, “타자판기 음료 개수 선택”, “타 자판기 재고 확인 요청”
Pre-Requisites	N/A
Typical Courses of Events	(U) : 사용자, (S) 시스템 1. (S) 시스템은 음료수 목록을 띄워준다 2. (U) 사용자는 구매할 음료를 선택한다. 3. (S) 시스템은 재고가 있을 경우, R1.2로 분기
Alternative Courses of Events	재고가 없을 경우 시스템은 R1.4로 분기하게 된다.
Exceptional Courses of Events	N/A

2043.Design Real Use Cases

Use Case	R1.2 현재 자판기 음료 개수 선택
Actor	user
Overview	현재 자판기에서 구매할 음료수 개수를 선택한다.
Type	Evident
Cross Reference	System Functions : R1.1, R1.6 Use cases : 음료 종류 선택, 음료 결제 요청
Pre-Requisites	현재 자판기에 존재하는 음료수를 선택해야 한다
Typical Courses of Events	(U) : 사용자, (S) 시스템 1. (S) DB에서 음료수 재고를 받아와 페이지를 띄워준다. 2. (U) 구매할 음료수 개수를 선택한다. 3. (S) '음료 결제 요청'으로 이동한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	R1.3 타 자판기 음료 개수 선택
Actor	user
Overview	타 자판기에서 구매할 음료수 개수를 선택한다.
Type	Evident
Cross Reference	System Functions : R1.1, R1.4, R1.6 Use cases : 음료 종류 선택, 타 자판기 재고확인 요청, 음료 결제 요청
Pre-Requisites	현재 자판기에 존재하지 않는 음료수를 선택해야 한다. 타 자판기에게서 재고정보를 수신해야 한다.
Typical Courses of Events	(S) 시스템, (U) 사용자 1.(S) '타 자판기 재고확인 요청'을 통해 각 자판기로부터 거리 정보와 음료 개수 정보를 받는다. 2,(U) 음료 재고를 선택한다. 3.(S) '음료 결제 요청'으로 이동한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E 1. 다른 자판기에도 존재하지 않을 경우 구매 불가 페이지를 출력한다.

2043.Design Real Use Cases

Use Case	R1.4 타 자판기 재고확인 요청
Actor	System
Overview	타 자판기에 재고확인 요청 메시지를 보낸다
Type	Hidden
Cross Reference	System Functions : R1.1, R1.3 Use cases : 음료 종류 선택, 타 자판기 음료 개수 선택
Pre-Requisites	사용자가 선택한 음료수가 현재 자판기에 존재하지 않는 음료 수여야 한다.
Typical Courses of Events	(S) 시스템, (O) 타 자판기 1. (S) : 타 자판기에 재고 요청 broadcast를 한다. 2. (O) : 요청받은 음료의 재고를 보내준다. 3. (S) : 재고 정보를 받는다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E 1. 5초 동안 응답이 오지 않는다면 재전송한다.

Use Case	R1.5 인증코드 입력
Actor	user
Overview	타 자판기에서 결제하고 받은 인증코드를 입력해 음료를 수령한다.
Type	Evident
Cross Reference	System Functions : R0.1 Use cases : 메뉴 화면 선택
Pre-Requisites	N/A
Typical Courses of Events	(S) 시스템, (U) 사용자 1.(U) 인증코드를 입력한다. 2.(S) 사용자가 입력한 인증코드가 유효한지 확인한다. 3.(S) DB에 저장한 인증코드를 가져와 확인한다. 3.(S) 인증코드가 유효하다면 음료수를 내보낸다. 4.(S) 수령완료 페이지를 띄워준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E 1. 인증코드가 유효하지 않다면 유효하지 않다고 출력한다.

2043.Design Real Use Cases

Use Case	R1.6 음료 결제 요청
Actor	user
Overview	음료 결제를 처리한다.
Type	Evident
Cross Reference	System Functions : R1.2, R1.3 Use cases : 현재 자판기 음료 개수 선택, 타 자판기 음료 개수 선택, 결제 처리
Pre-Requisites	사용자가 음료 종류와 개수를 선택해야 한다.
Typical Courses of Events	(S) 시스템, (U) 유저, (C) 카드사 1.(U) 출력된 정보를 확인하고 카드 정보를 입력한다. 2.(S) 카드사에 카드 정보와 가격을 전달한다. 3.(C) 카드사에서 결제 처리한다. 4.(S) 결제가 성공적이면 재고를 주문한 만큼 줄인다 5.(S) 결제가 정상적으로 처리되었다는 메시지를 안내한다.
Alternative Courses of Events	A 1. 선결제 구매인 경우 '구매 자판기 선결제 요청'을 부른다.
Exceptional Courses of Events	E 1. 결제 실패된 경우 사용자에게 결제 실패를 안내한다.

Use Case	R1.7 구매 자판기 선결제 요청
Actor	System
Overview	사용자가 선택한 자판기에 선결제 요청을 보낸다.
Type	Hidden
Cross Reference	System Functions : R1.6 Use cases : 음료 결제 요청
Pre-Requisites	사용자가 타 자판기에서 구매할 음료수의 개수를 선택한다.
Typical Courses of Events	(S) 시스템, (O) 다른 자판기 1.(S) 인증코드를 생성한다. 2.(S) 생성한 인증코드와 음료 개수, 종류를 타 자판기에 전송한다. 3.(O) 음료 재고를 확인하여 결제가 가능한지 응답한다. 4.(S) 결제 가능 응답이 오면 인증코드와 자판기 위치를 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E 1. 결제 불가능 응답이 오면 그 다음으로 가까운 자판기에 선결제 요청을 보낸다. 모든 자판기에서 결제 불가능 응답이 오면 결제 실패 페이지를 안내한다.

2043.Design Real Use Cases

Use Case	R2.1 관리자 페이지 로그인
Actor	admin
Overview	관리자 페이지 접근을 위한 로그인
Type	Evident
Cross Reference	System Functions : R0.1, R2.3, R2.2 Use cases : 메뉴 화면 선택, 재고 관리, 관리자 페이지 로그아웃
Pre-Requisites	메뉴 화면에서 관리자 페이지를 선택해야 한다.
Typical Courses of Events	(A) : Admin, (S) : System 1. (A) : 관리자 페이지를 진입하기 위한 비밀번호를 입력한다. 2. (S) : 비밀번호가 유효한지 검사한다. 3. (S) : 모든 재고를 받아온다. 4. (S) : 관리자 페이지로 이동한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E 1. 인증코드가 유효하지 않다면 유효하지 않다는 메시지를 출력한다.

Use Case	R2.2 관리자 페이지 로그아웃
Actor	admin
Overview	메뉴 화면으로 돌아가기 위한 로그아웃
Type	Evident
Cross Reference	System Functions : R2.1, R0.1 Use cases : 관리자 페이지 로그인, 메뉴 화면 선택
Pre-Requisites	메뉴 화면에서 관리자 페이지 로그인 상태여야 한다.
Typical Courses of Events	(A) : Admin, (S) : System 1. (A) : 관리자 페이지 로그아웃 버튼을 누른다. 3. (S) : '메뉴 화면 선택'으로 이동한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

2043.Design Real Use Cases

Use Case	R2.3 재고 관리
Actor	admin
Overview	관리자 페이지 로그인 후 재고관리 기능
Type	Evident
Cross Reference	System Functions : R2.1 Use cases : 관리자 페이지 로그인
Pre-Requisites	관리자 페이지 로그인 성공
Typical Courses of Events	(A) : 관리자, (S) : System 1. (A) 관리자가 음료의 재고와 종류를 선택해서 수정함. 2. (S) 수정한 데이터를 시스템에 반영함.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	R3.1 재고 확인 요청
Actor	other DVM
Overview	타 자판기에서 재고 확인 요청 메시지 수신 시 재고 응답
Type	Hidden
Cross Reference	System Functions : Use cases : N/A
Pre-Requisites	N/A
Typical Courses of Events	(O) : 다른 자판기 (S) : 시스템 1. (O) : 재고 확인 요청 메시지 송신 2. (S) : 재고 확인 요청 메세지 수신 3. (S) : 시스템의 재고 확인 후 재고 내용 응답
Alternative Courses of Events	N/A
Exceptional Courses of Events	E 1. 수신 메시지의 dst_id가 올바르지 않을 경우 응답하지 않는다. E 2. 수신 메시지의 src_id가 올바르지 않을 경우 응답하지 않는다.

2043.Design Real Use Cases

Use Case	R3.2 선결제 요청 처리
Actor	other DVM
Overview	타 자판기에서 선결제 요청 메시지 수신 시 선결제 요청 처리
Type	Hidden
Cross Reference	System Functions : R3.3 Use cases : 인증코드 유효 시간 확인
Pre-Requisites	N/A
Typical Courses of Events	(O) : 타 자판기 (S) : 시스템 1. (O) : 선결제 요청 메시지 송신. 2. (S) : 시스템의 재고 확인 3. (S) : 해당 음료수 재고 차감 4. (S) : 받은 인증코드 저장 5. (S) : 선결제 가능 응답.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E 1.재고가 없는 경우 선결제 실패 메시지를 보낸다.

Use Case	R3.3 인증코드 유효 시간 확인
Actor	timer
Overview	타이머를 통해 인증코드 유효시간을 확인함.
Type	hidden
Cross Reference	System Functions : R3.2 Use cases : 선결제 요청 처리
Pre-Requisites	인증코드가 저장되어야 함.
Typical Courses of Events	(S) : 시스템, (T) 타이머 1. (T) : 일정 시간 마다 유효시간 확인 요청 2. (S) : 만료 메시지 수신 시 유효시간 확인 후 인증코드 파기 3. (S) : 차감된 음료 개수 복구
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

2042.Story Board



대충 자판기 로고나 타이틀



Buy Drink



Receive Drink

2042.Story Board

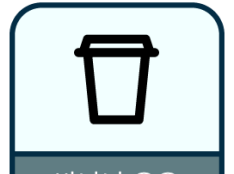
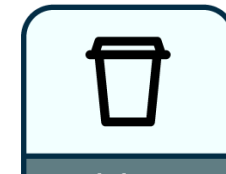
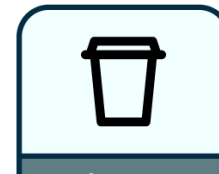


인증코드를 입력하세요

확인



대충 상단바

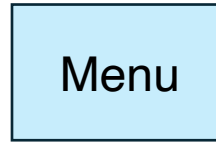


2042.Story Board

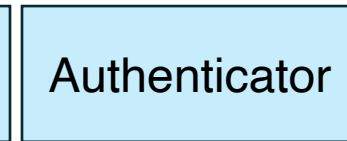
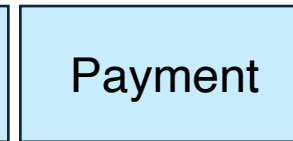
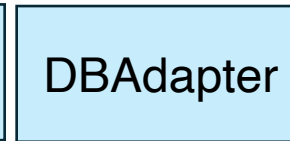
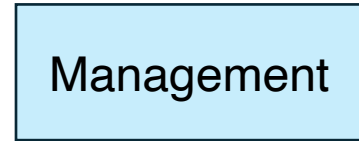
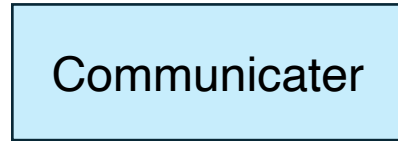
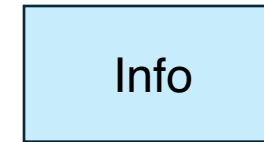
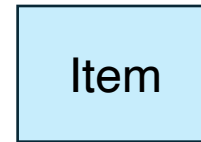
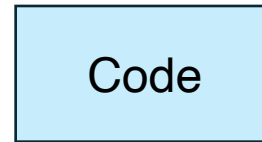
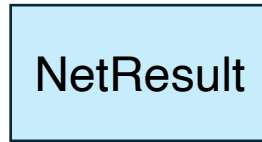
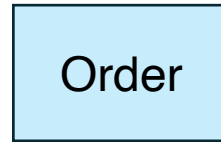


2043.System Architecture

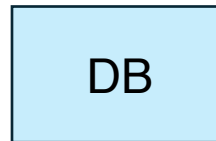
Presentation
Layer



Application
Layer



Storage



2044. Define Interaction Diagrams

sd 메뉴화면 선택

User

: Menu

1 : requestHTML(index.html)

2 : responseHTML(selectMenu.html)

3 : requestHTML(page: String)

alt 페이지 선택

[page == Buy.html]

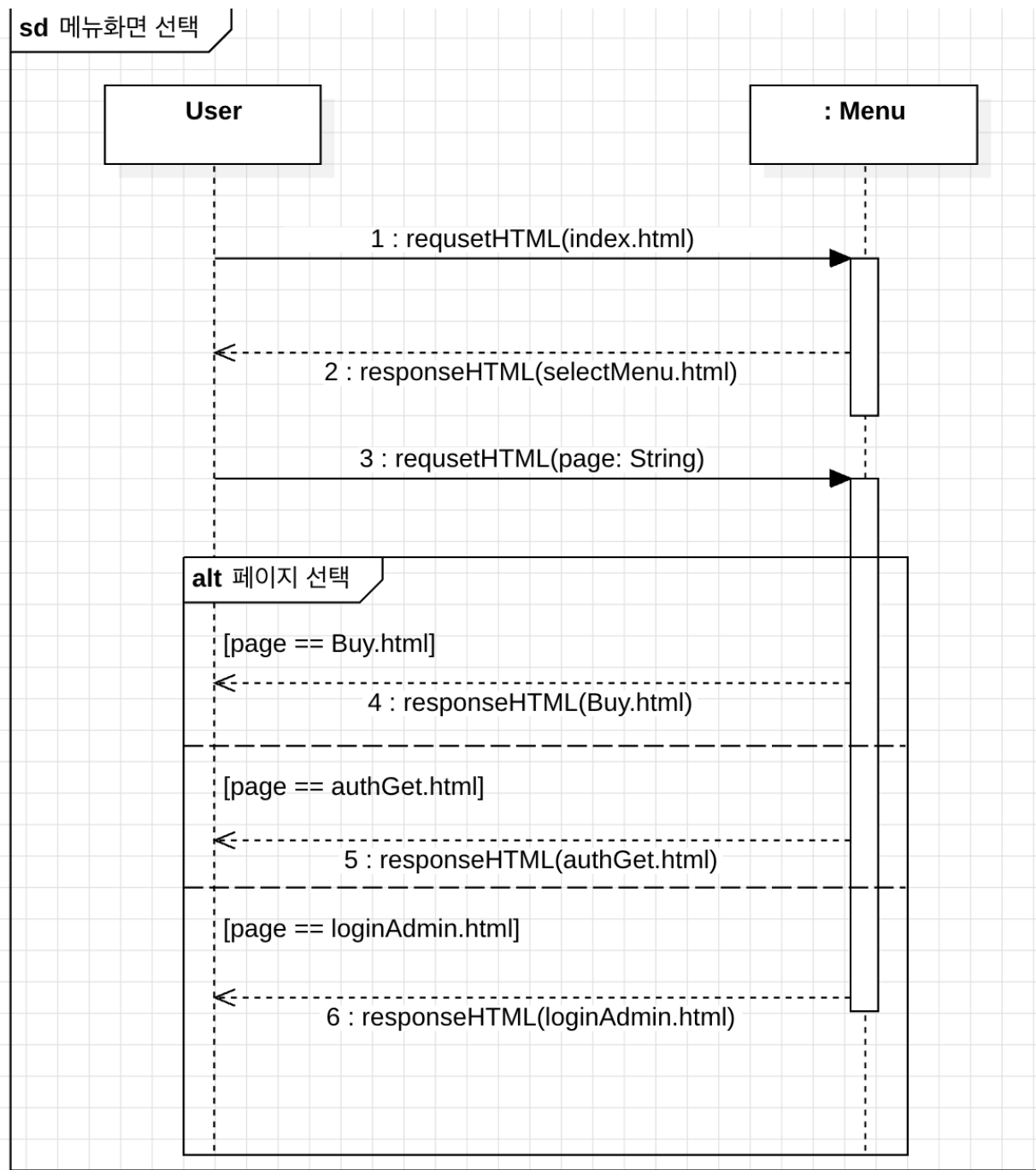
4 : responseHTML(Buy.html)

[page == authGet.html]

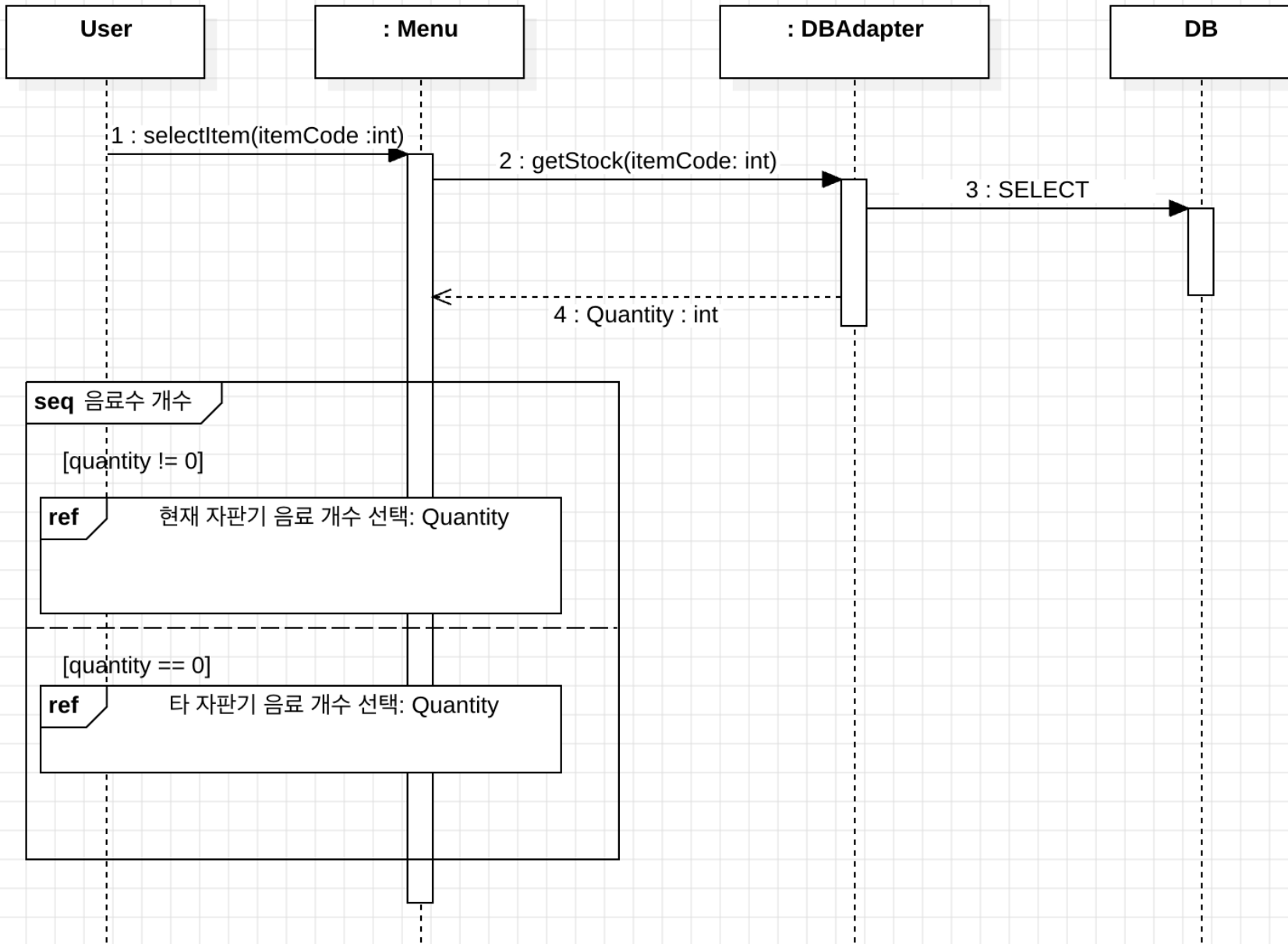
5 : responseHTML(authGet.html)

[page == loginAdmin.html]

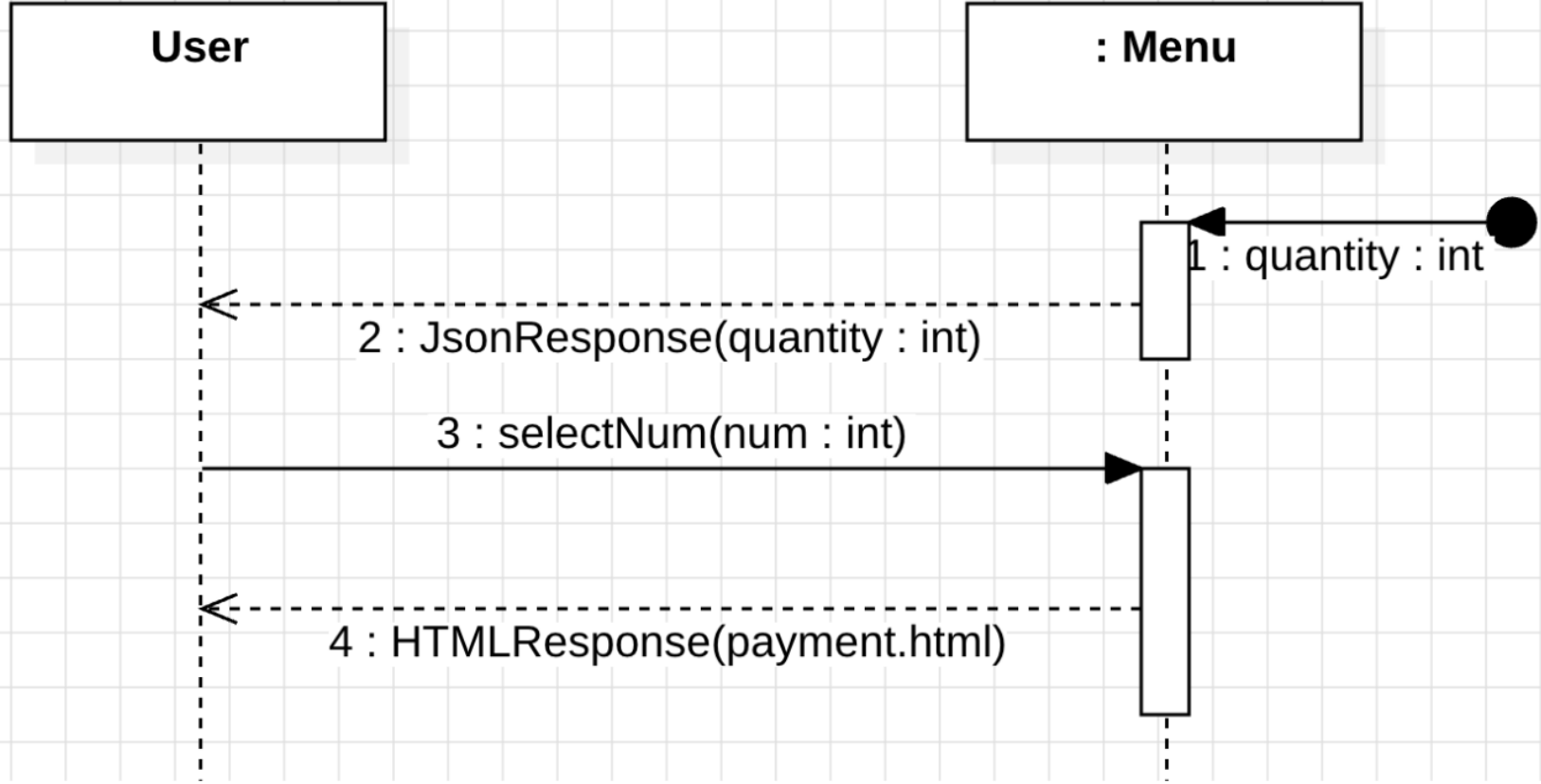
6 : responseHTML(loginAdmin.html)



sd 음료 종류 선택



sd 현재 자판기 음료 개수 선택



sd 타 자판기 음료 개수 선택

User

: Menu

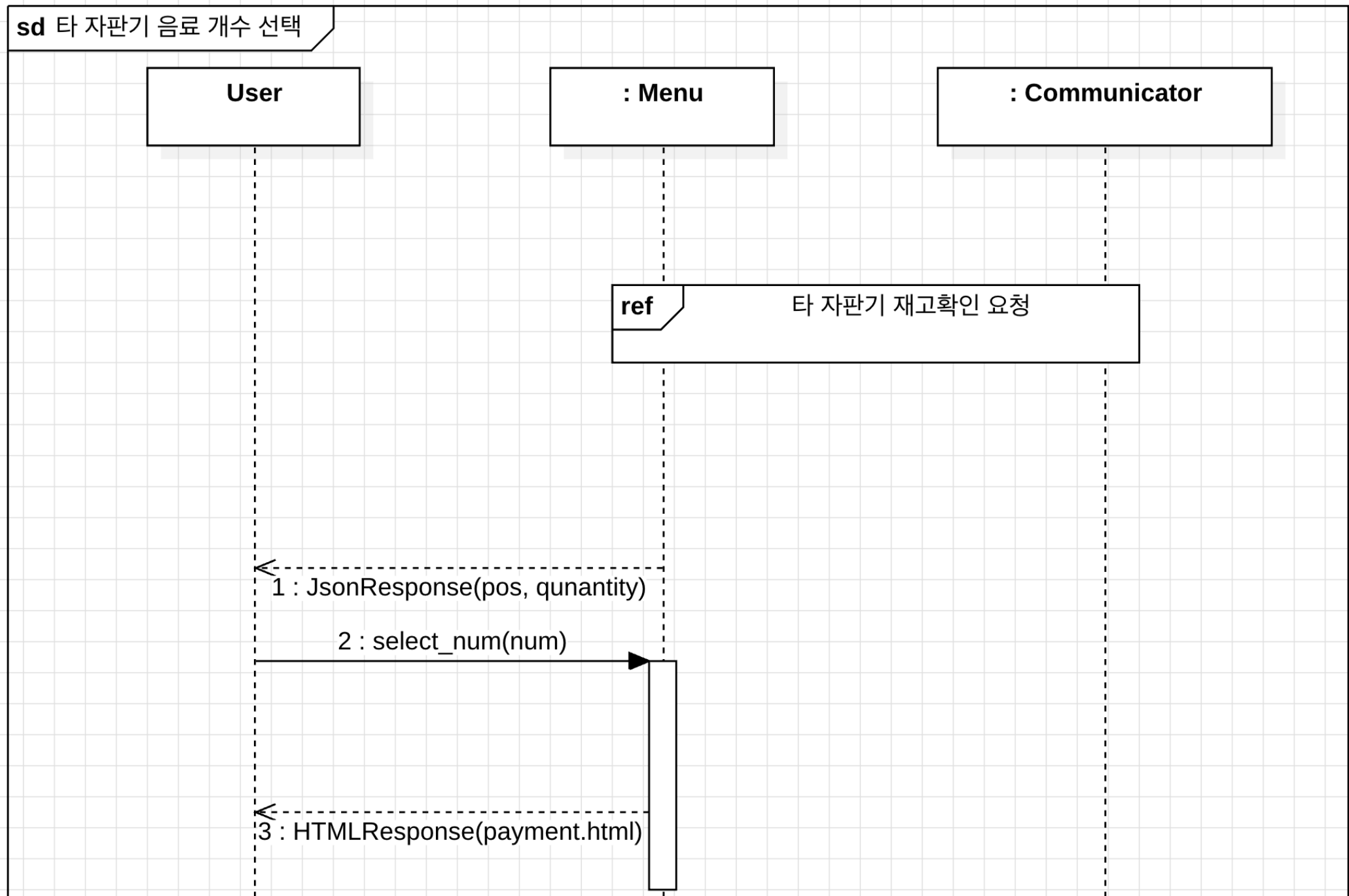
: Communicator

ref
타 자판기 재고확인 요청

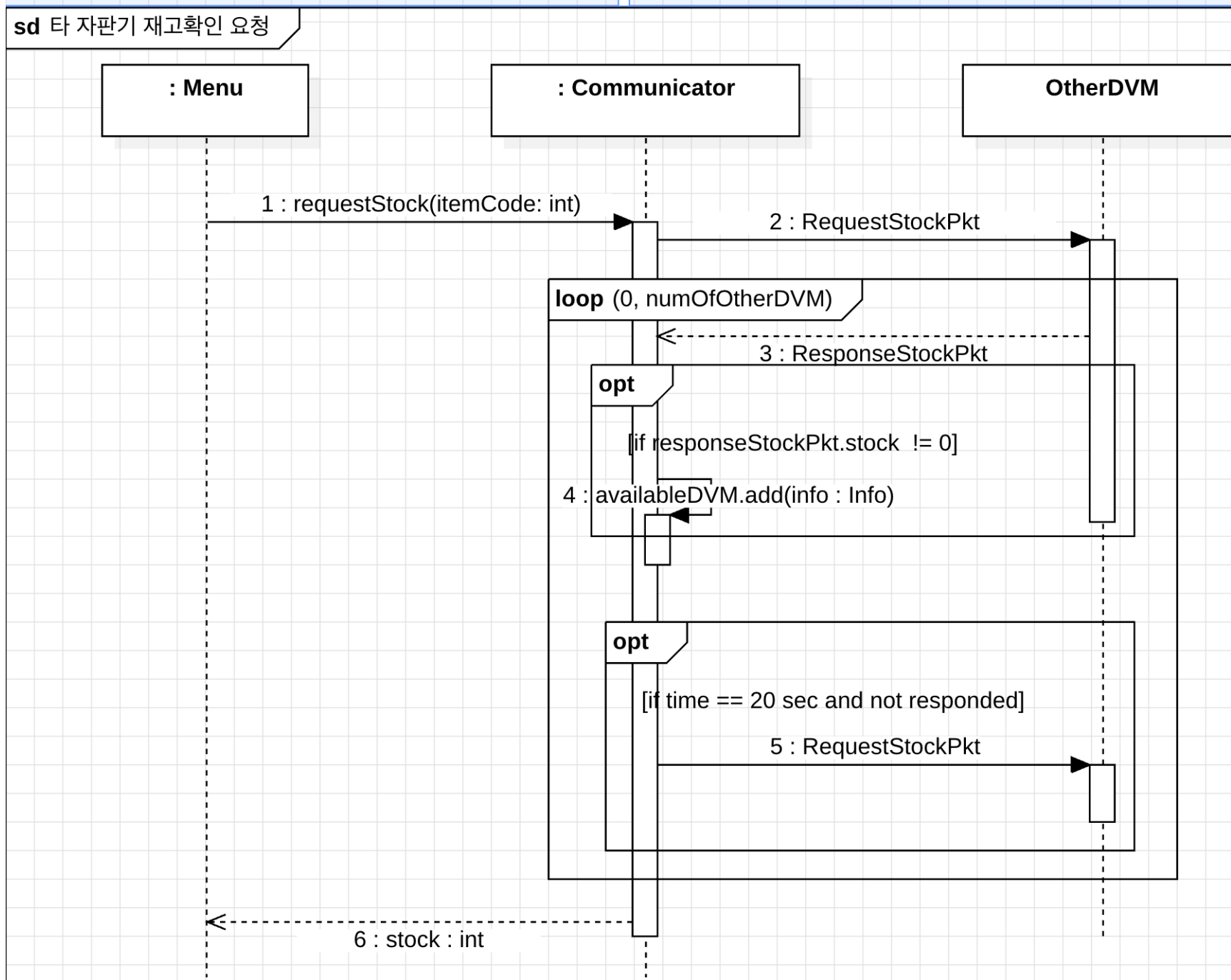
1 : JsonResponse(pos, qunantity)

2 : select_num(num)

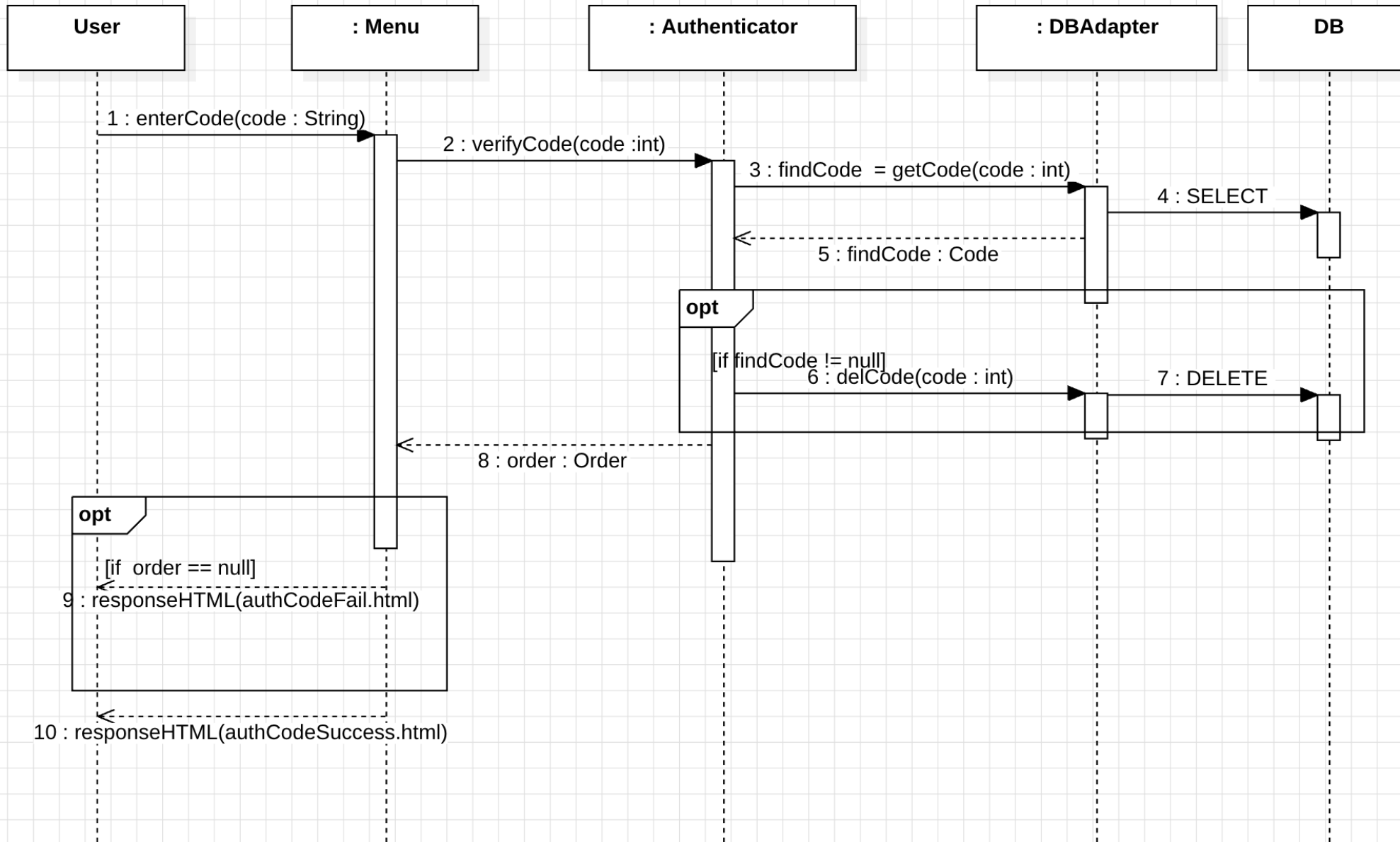
3 : HTMLResponse(payment.html)



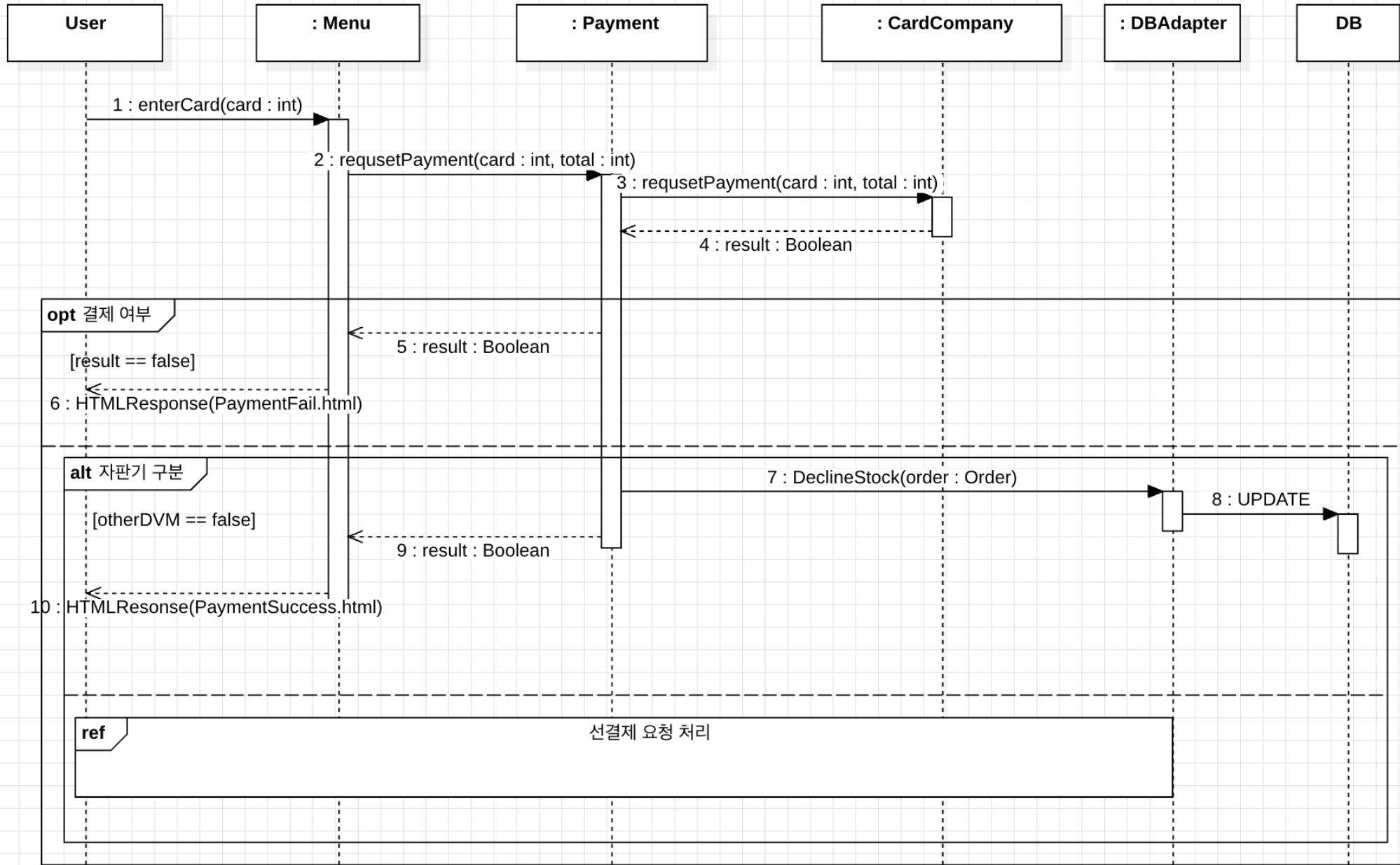
sd 타 자판기 재고확인 요청



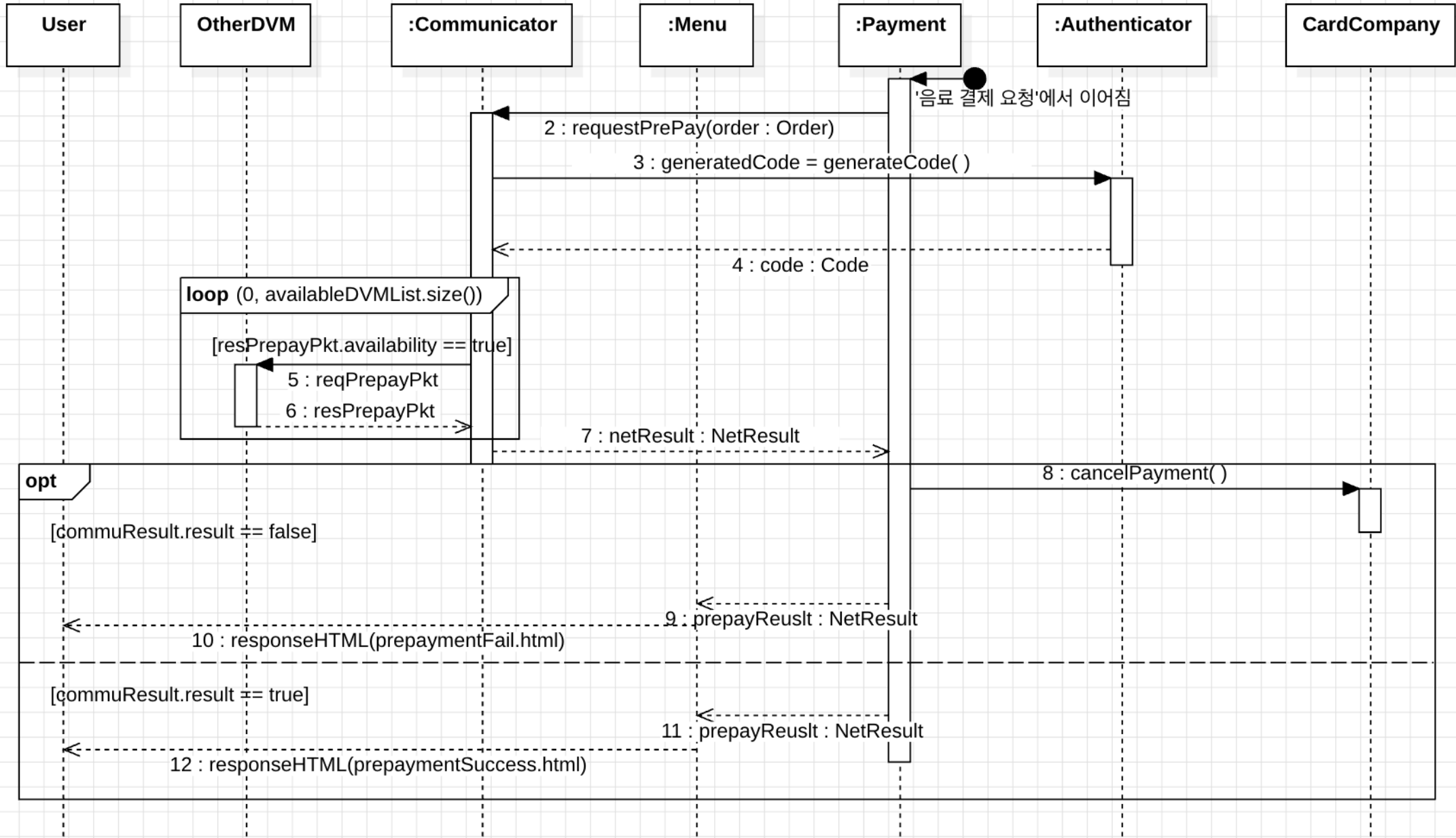
sd 인증코드 입력



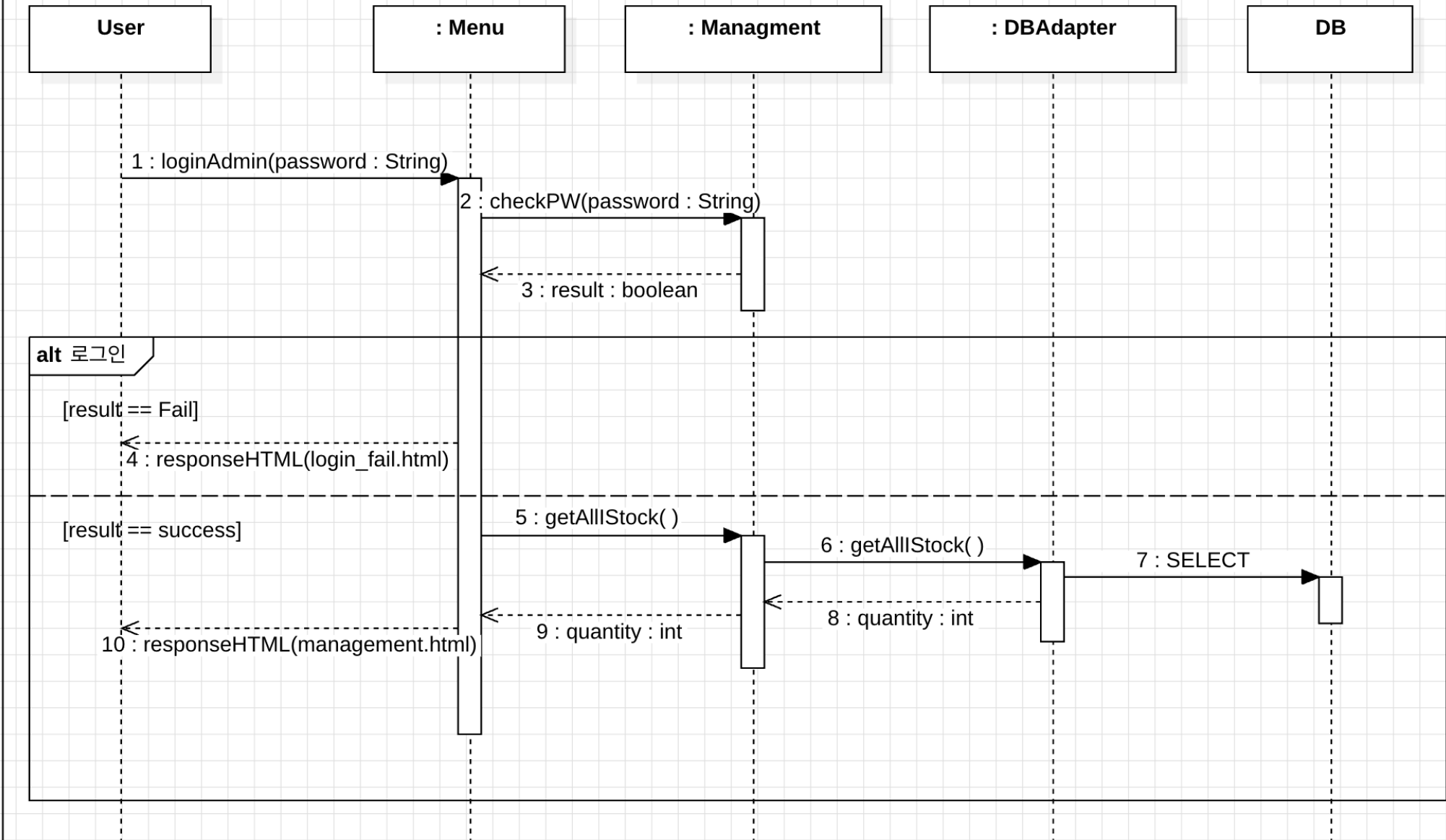
sd 음료 결제 요청

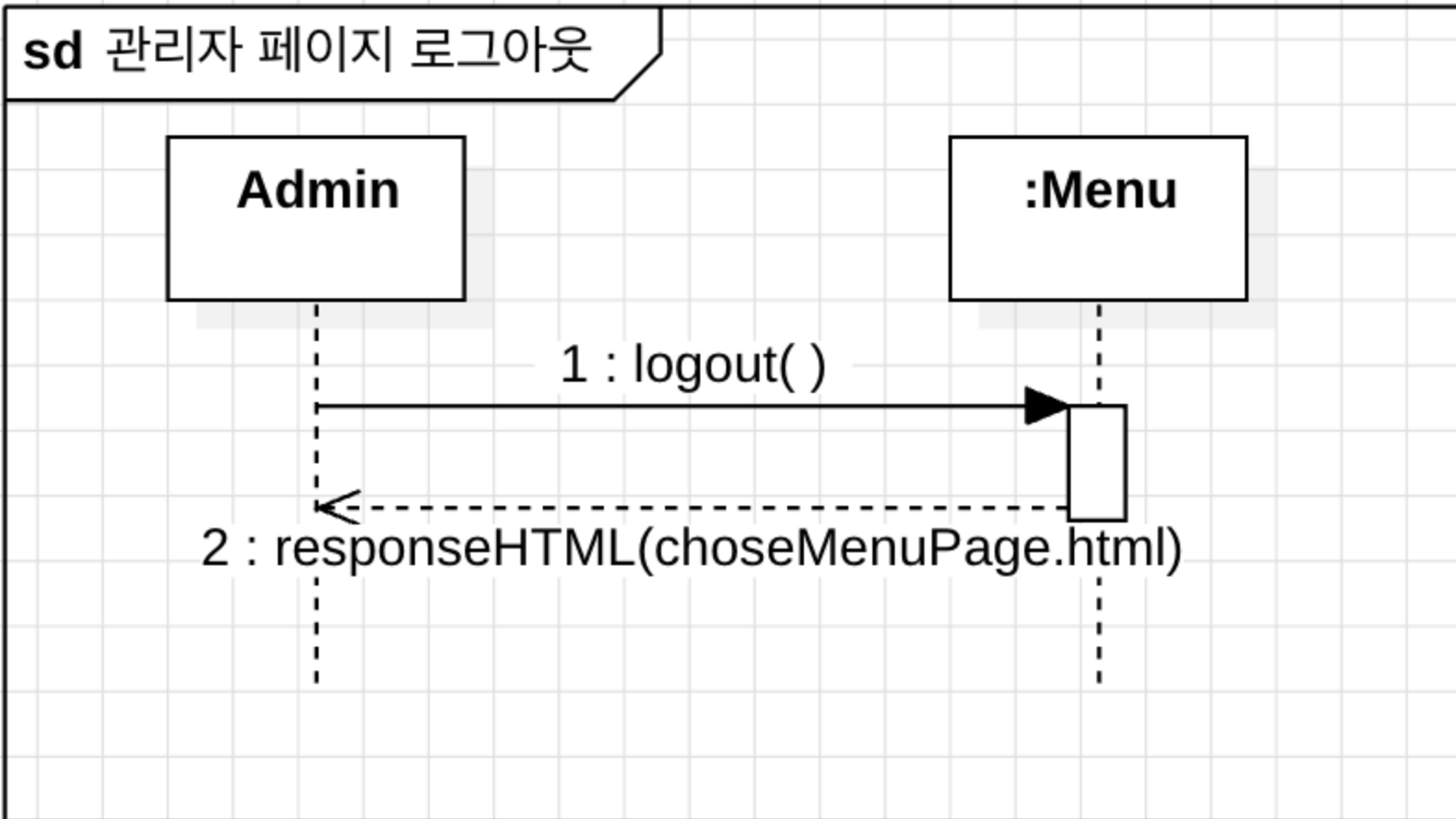


sd 구매 자판기 선결제 요청

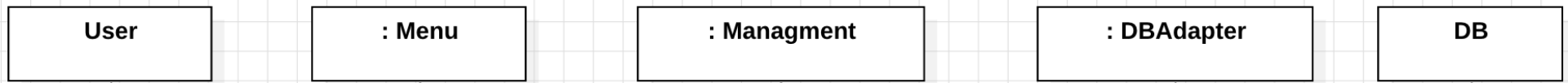


sd 관리자 페이지 로그인





sd 재고 관리



1 : enterStock(order : Order)

2 : changeAdminStock(order : Order)

opt 재고 값 확인

[quantity < 0]

3 : result : boolean

4 : result : boolean

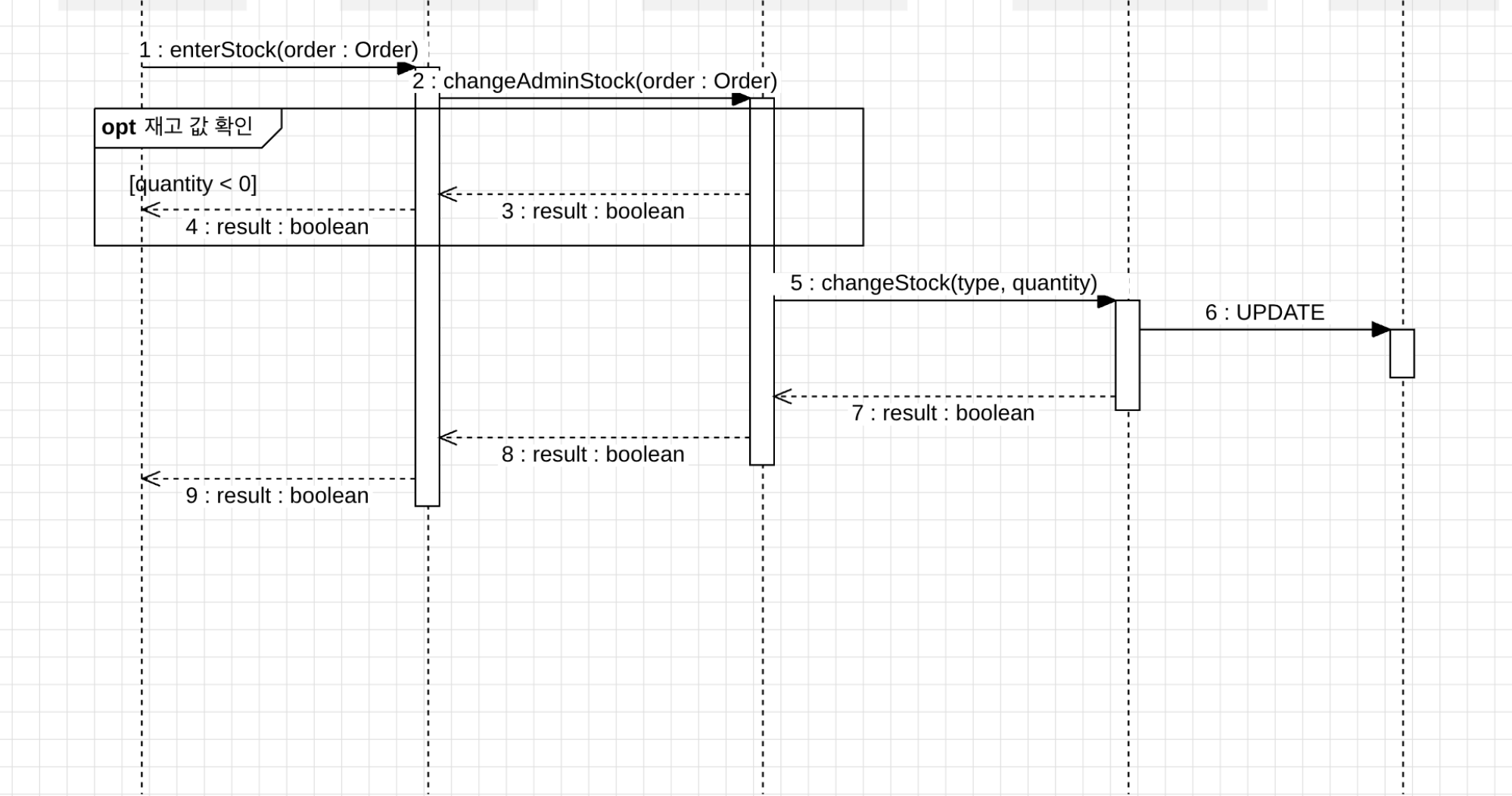
5 : changeStock(type, quantity)

6 : UPDATE

7 : result : boolean

8 : result : boolean

9 : result : boolean



sd 재고 확인 요청

OtherDVM

: Communicator

: DBAdapter

DB

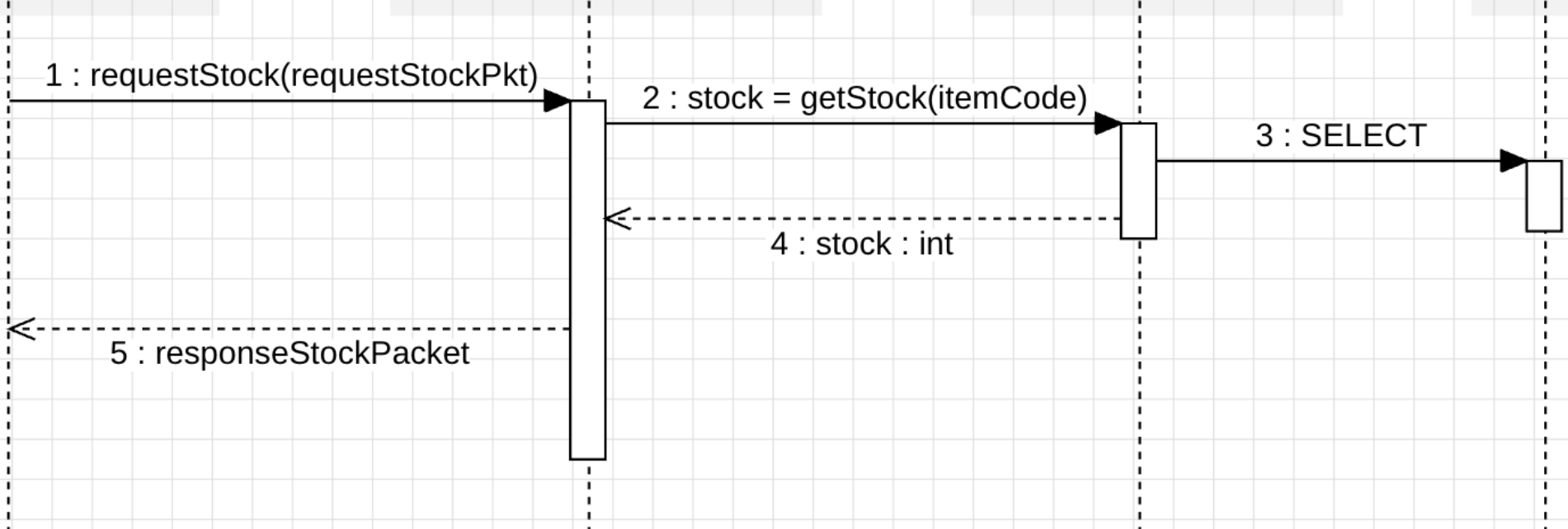
1 : requestStock(requestStockPkt)

2 : stock = getStock(itemCode)

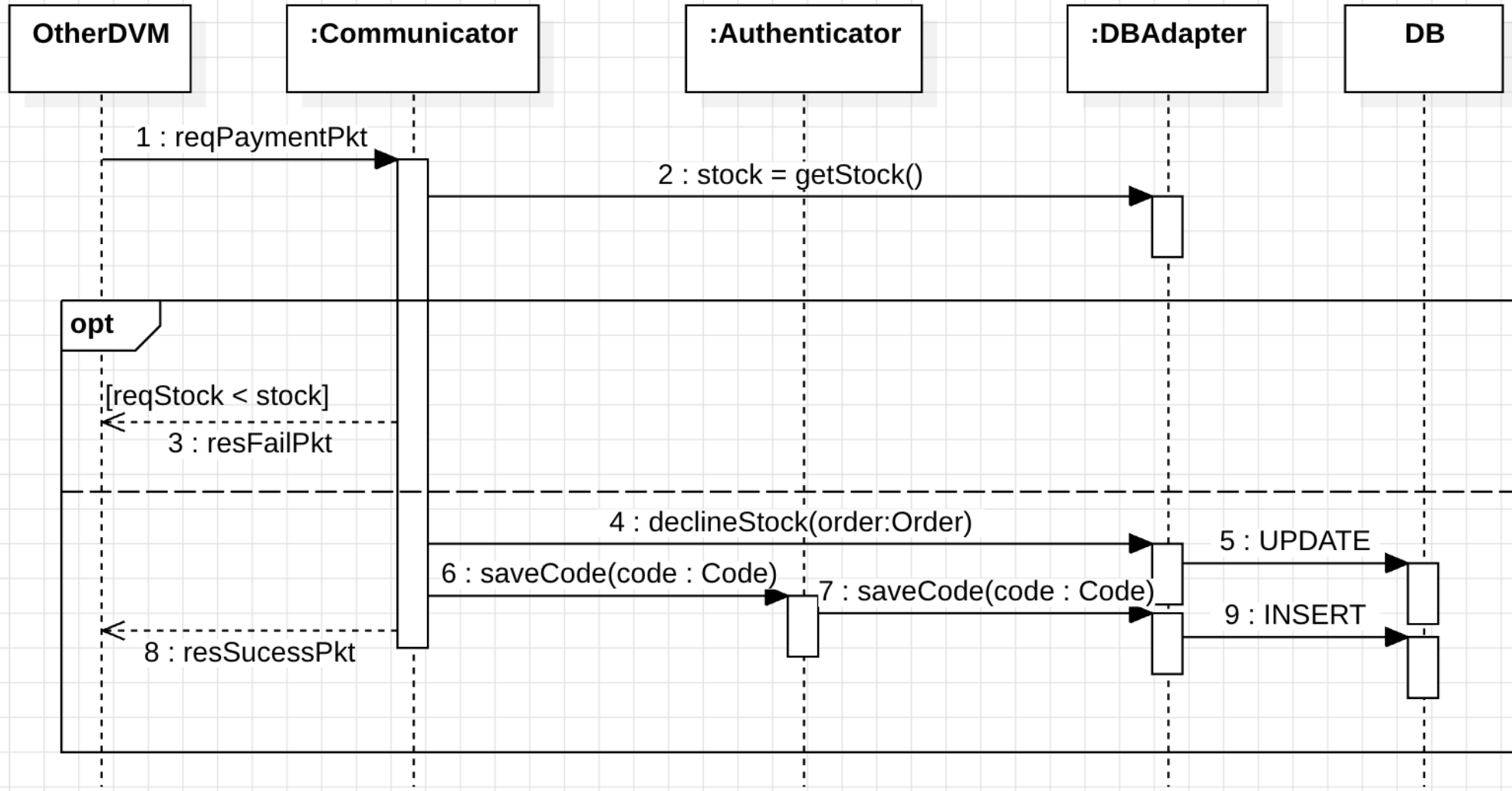
3 : SELECT

4 : stock : int

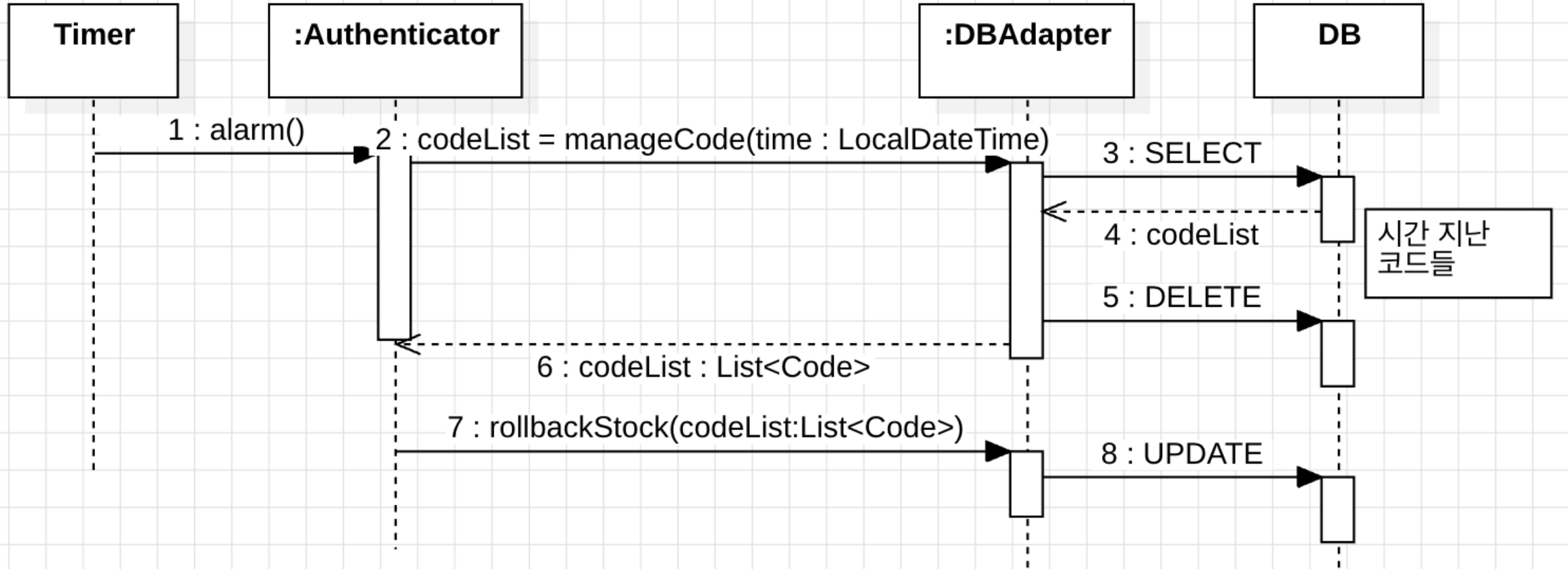
5 : responseStockPacket



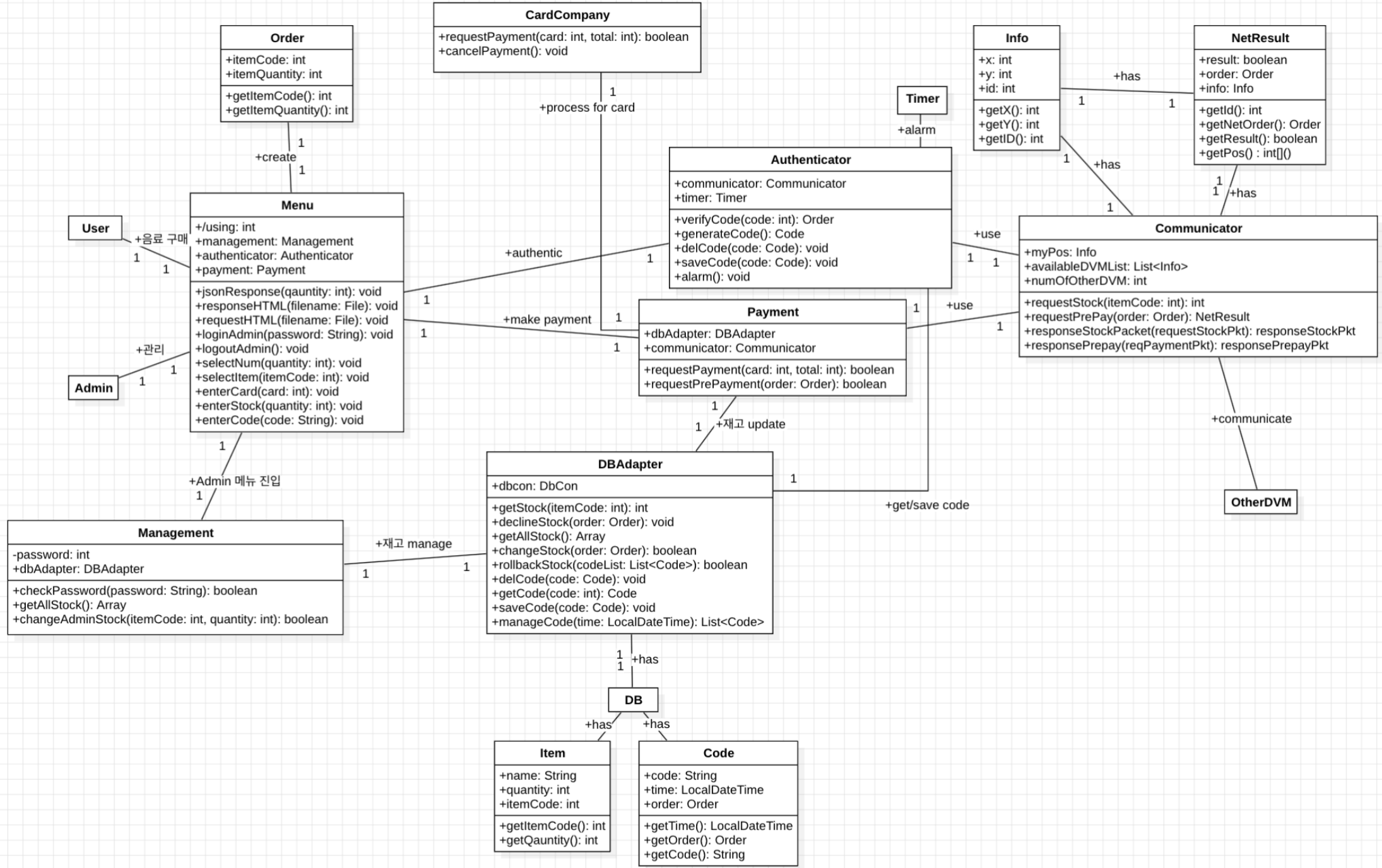
sd 선결제 요청 처리



sd 인증코드 유효 시간 확인

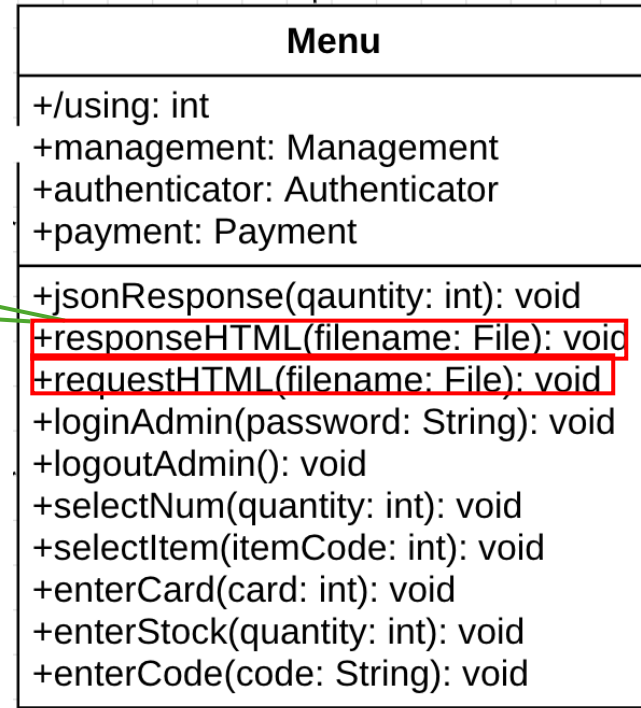
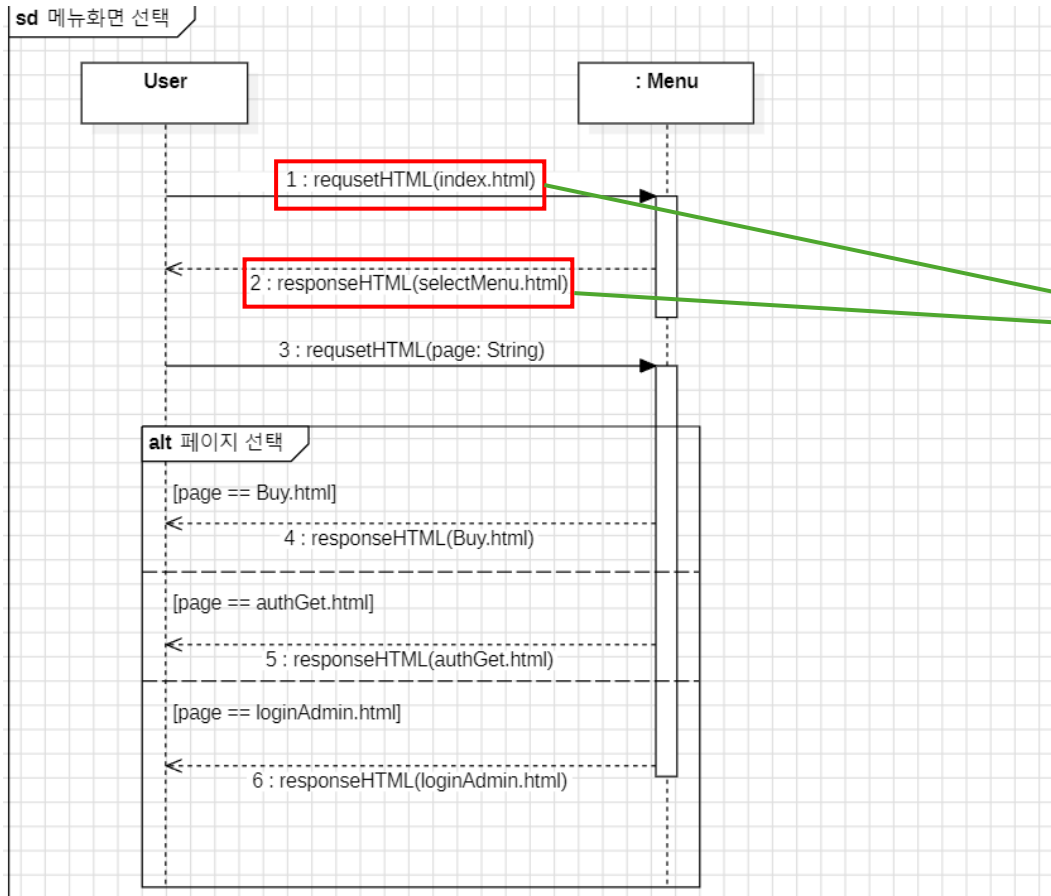
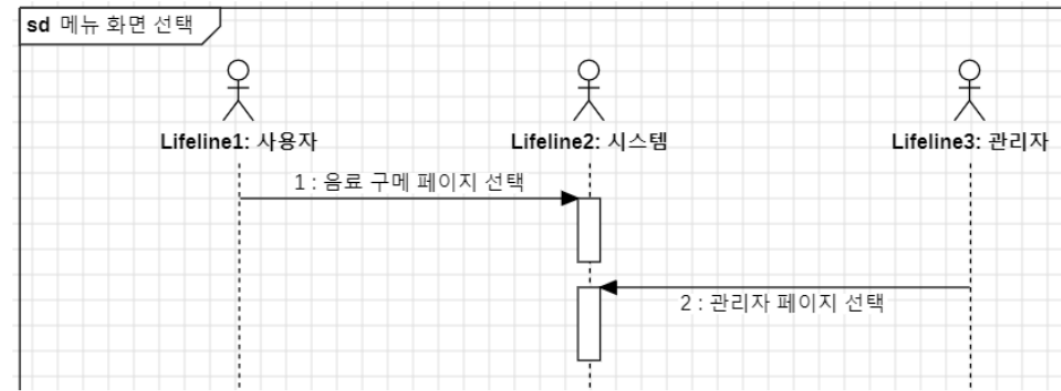


2045. Define Design Class Diagrams



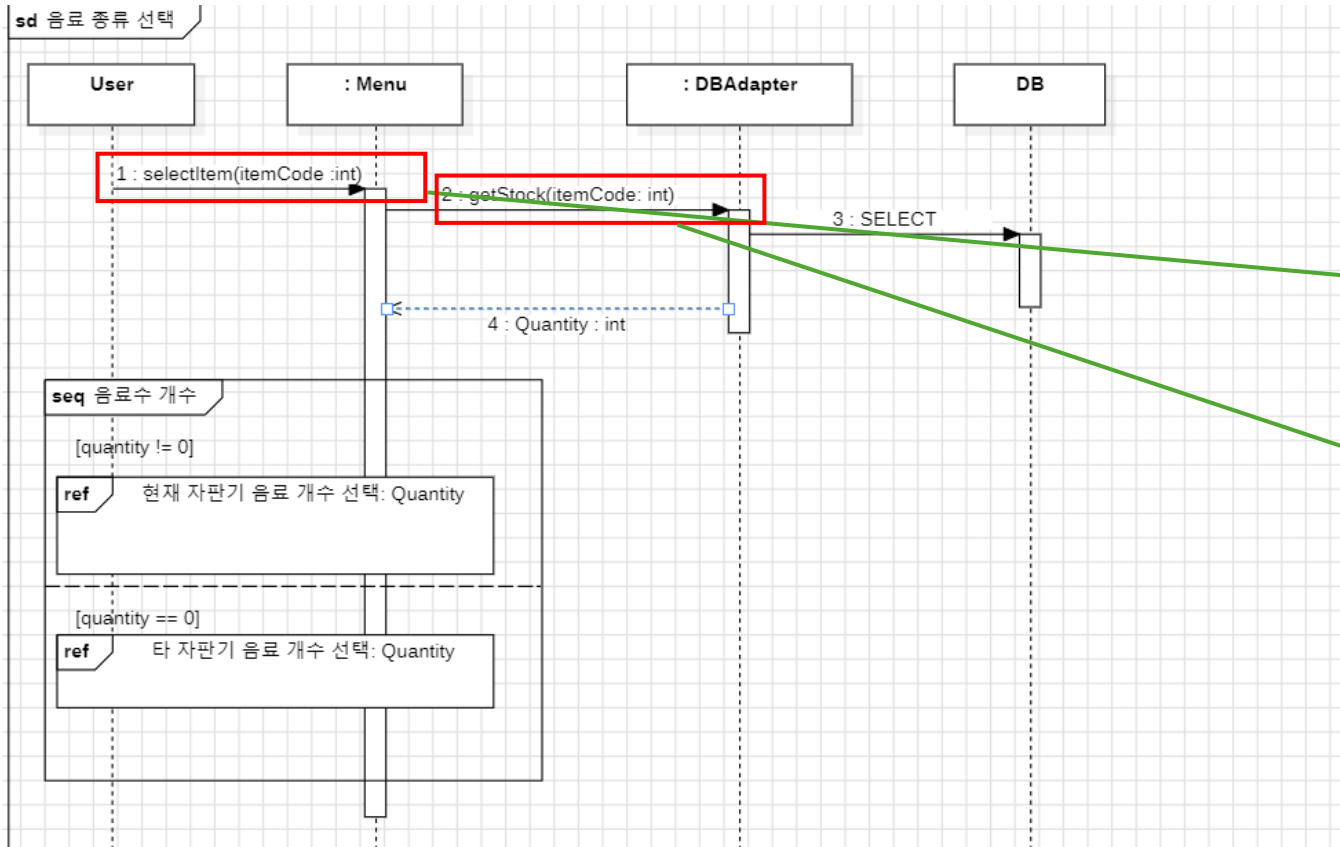
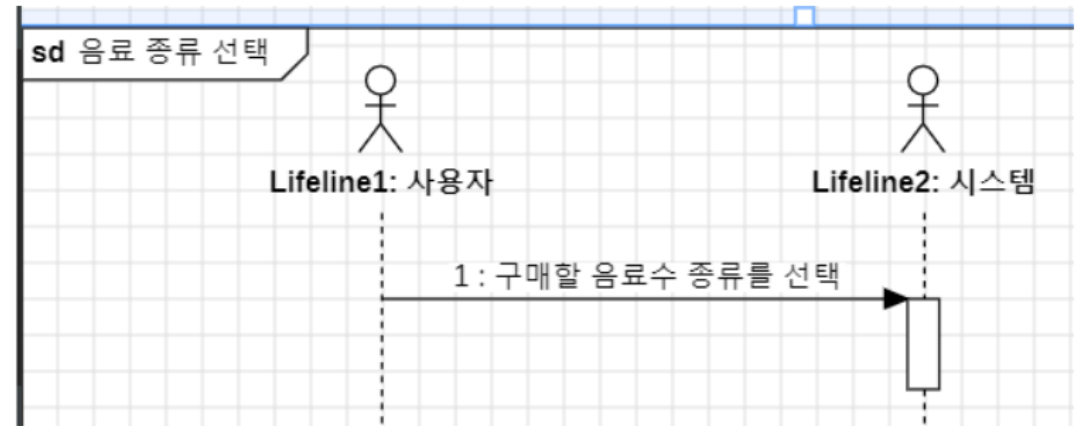
2046. Design Traceability Analysis

- (U) : 사용자, (S) 시스템, (A) 관리자
1. (S) 시스템이 관리자 화면으로 들어 갈 지 구매 화면으로 들어갈지 선택지를 띄워준다.
 2. (U) 사용자는 구매 화면을 고른다.
 3. (S) 시스템은 선택한 화면에 따라 인터페이스를 변경한다.



(U) : 사용자, (S) 시스템

1. (S) 시스템은 음료수 종류를 띄워준다
2. (U) 사용자는 구매할 음료수 종류를 선택한다.
3. (S) 시스템은 보유중인 재고에 따라 '현재 자판기 음료 개수 선택', '타 자판기 음료 개수 선택'을 부름.'

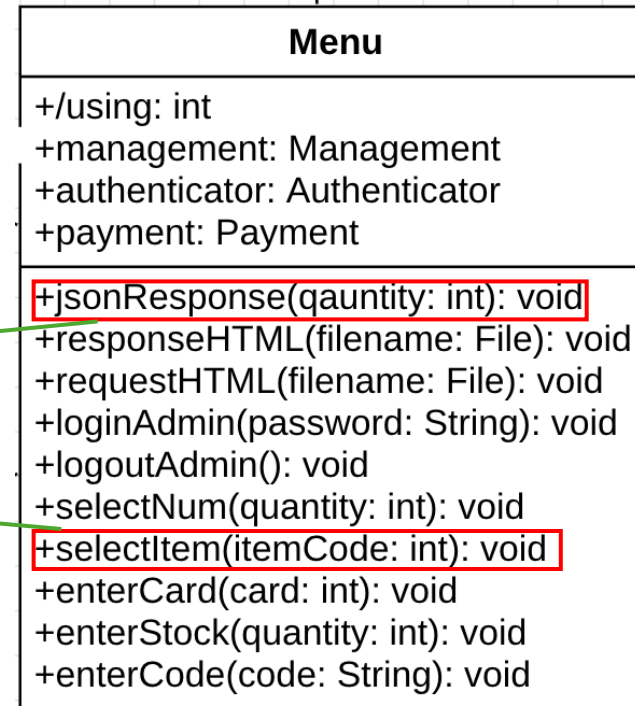
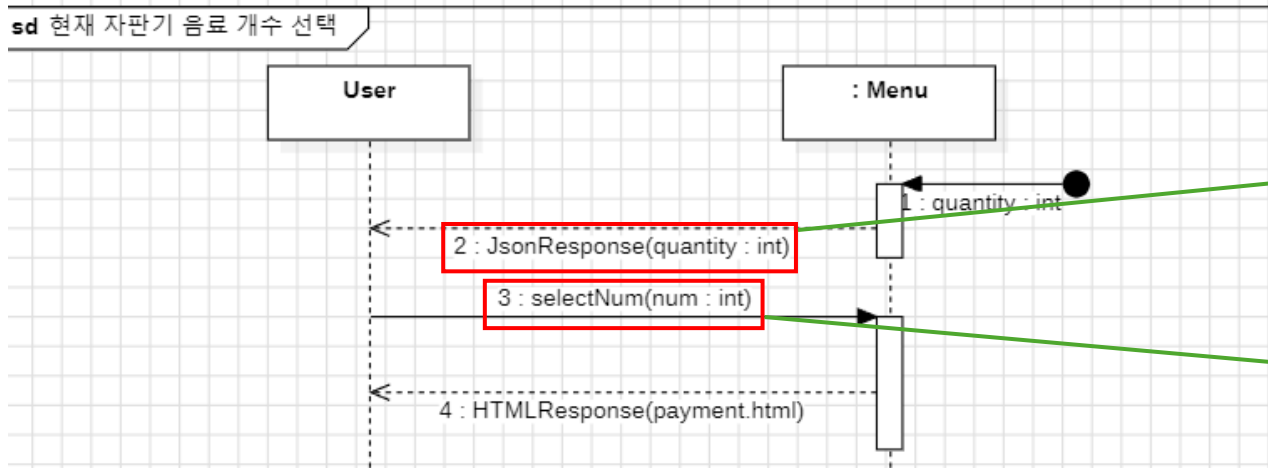
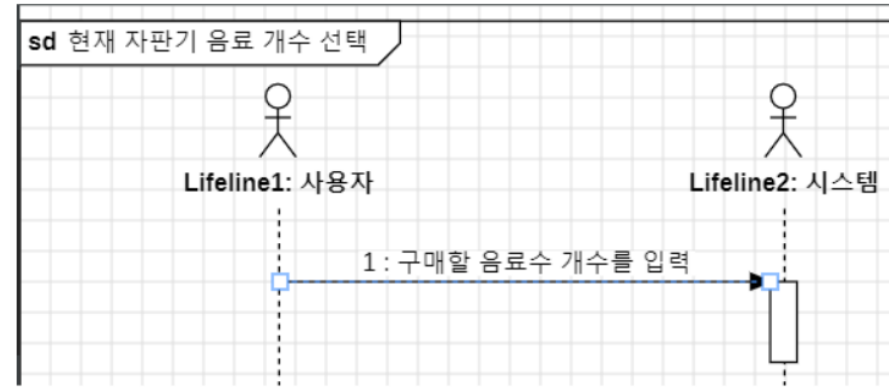


Menu
+/using: int
+management: Management
+authenticator: Authenticator
+payment: Payment
+jsonResponse(quantity: int): void
+responseHTML(filename: File): void
+requestHTML(filename: File): void
+loginAdmin(password: String): void
+logoutAdmin(): void
+selectNum(quantity: int): void
+selectItem(itemCode: int): void
+enterCard(card: int): void
+enterStock(quantity: int): void
+enterCode(code: String): void

DBAdapter
+dbcon: DbCon
+getStock(itemCode: int): int
+declineStock(order: Order): void
+getAllStock(): Array
+changeStock(order: Order): boolean
+rollbackStock(codeList: List<Code>): boolean
+delCode(code: Code): void
+getCode(code: String): Code
+saveCode(code: Code): void
+manageCode(time: LocalDateTime): List<Code>

(U) : 사용자, (S) 시스템

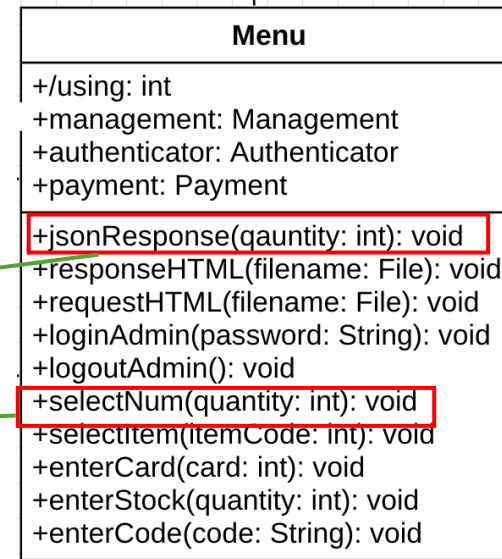
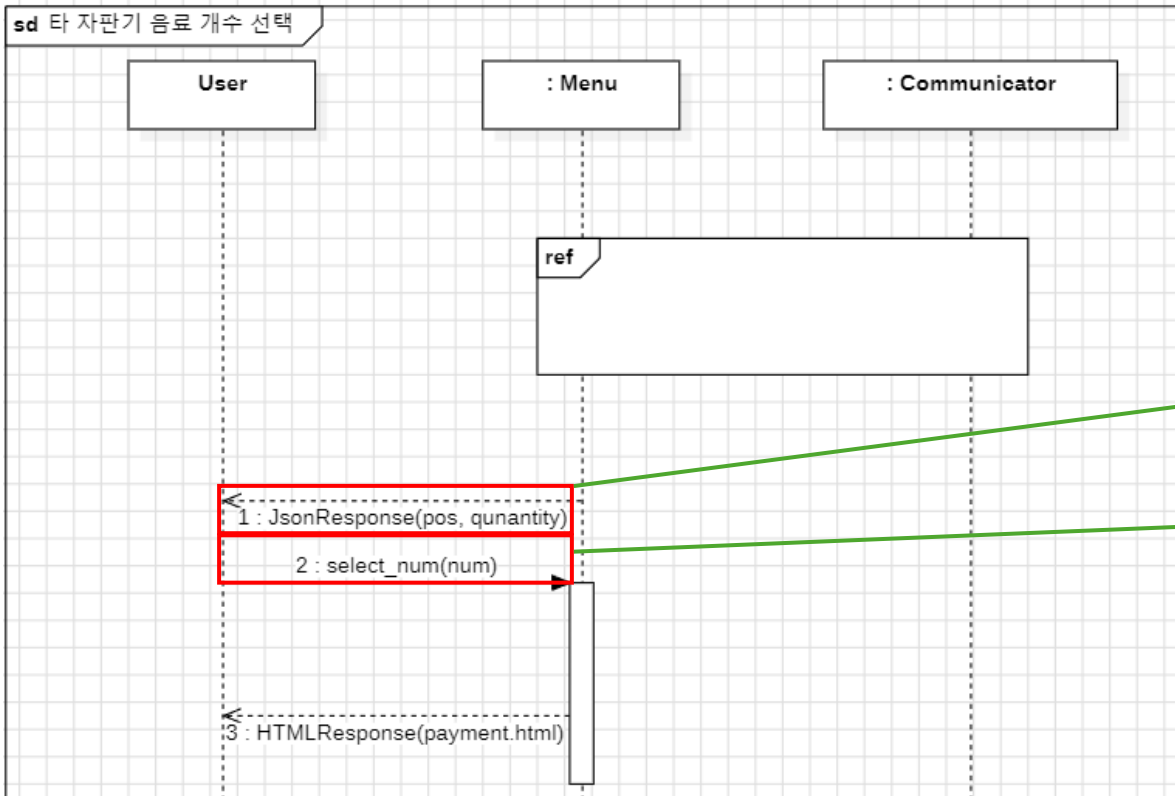
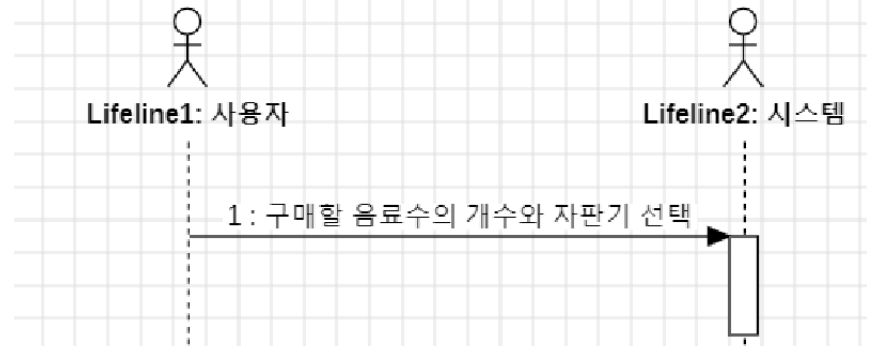
1. (S) 보유중인 음료수 개수를 띄워준다
2. (U) 구매할 음료수 개수를 선택한다.
3. (S) '음료 결제 요청'으로 이동한다.



(S) 시스템, (U) 사용자

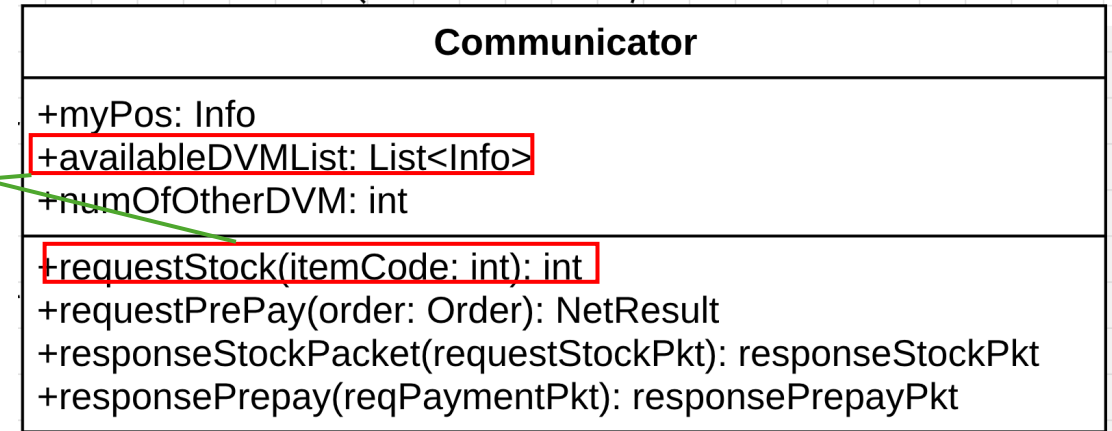
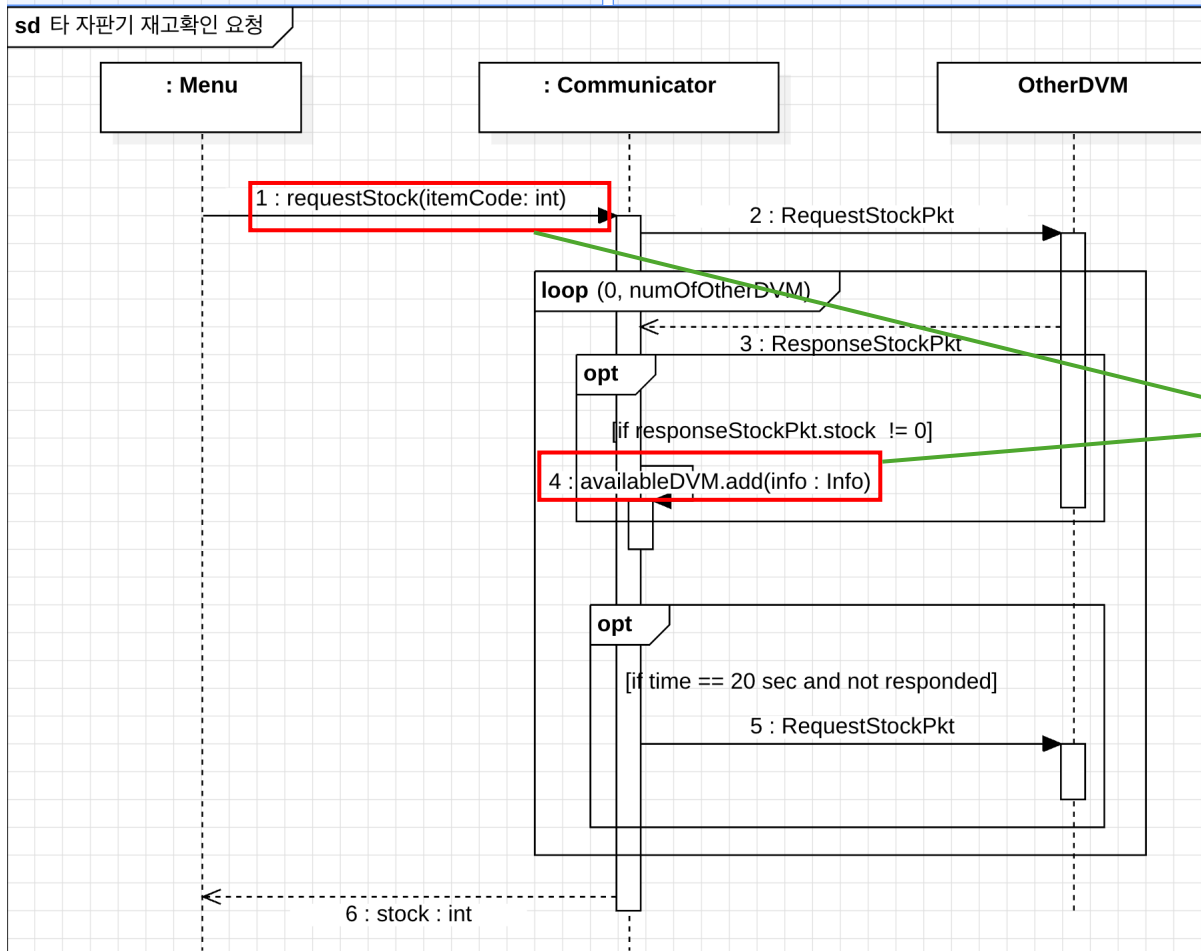
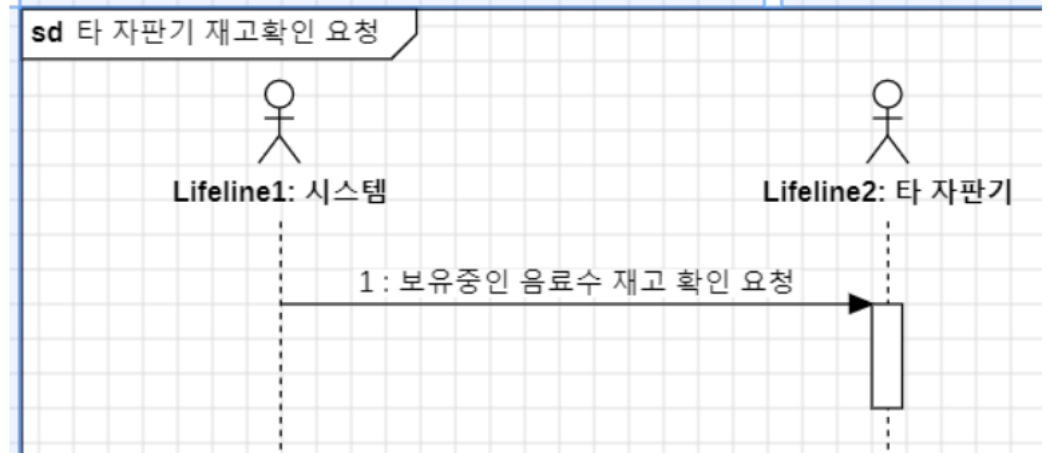
- 1.(S) '타 자판기 재고확인 요청'을 통해 각 자판기로부터 거리 정보와 음료 개수 정보를 받는다.
- 2.(S) 다른 자판기에서 받은 음료 재고를 거리 순으로 표시한다.
- 3.(U) 원하는 자판기를 골라 음료 재고를 선택한다.
- 4.(S) '음료 결제 요청'으로 이동한다.

sd 타 자판기 음료 개수 선택

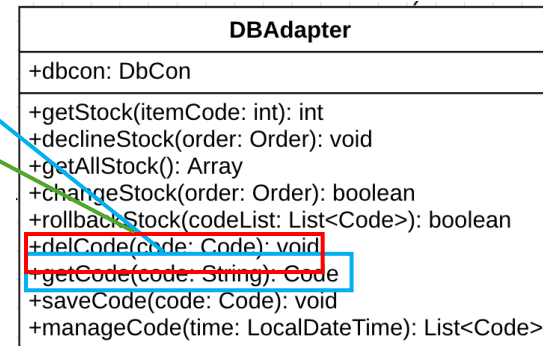
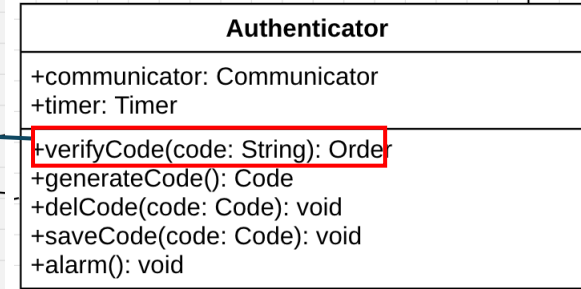
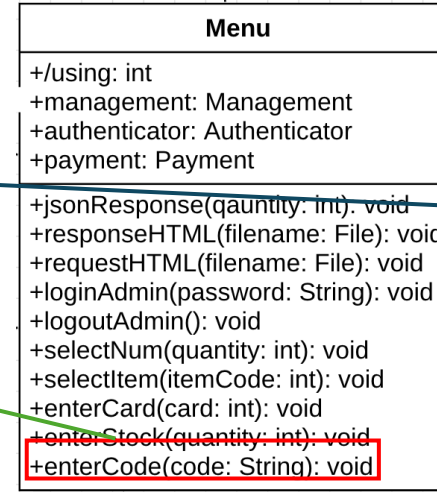
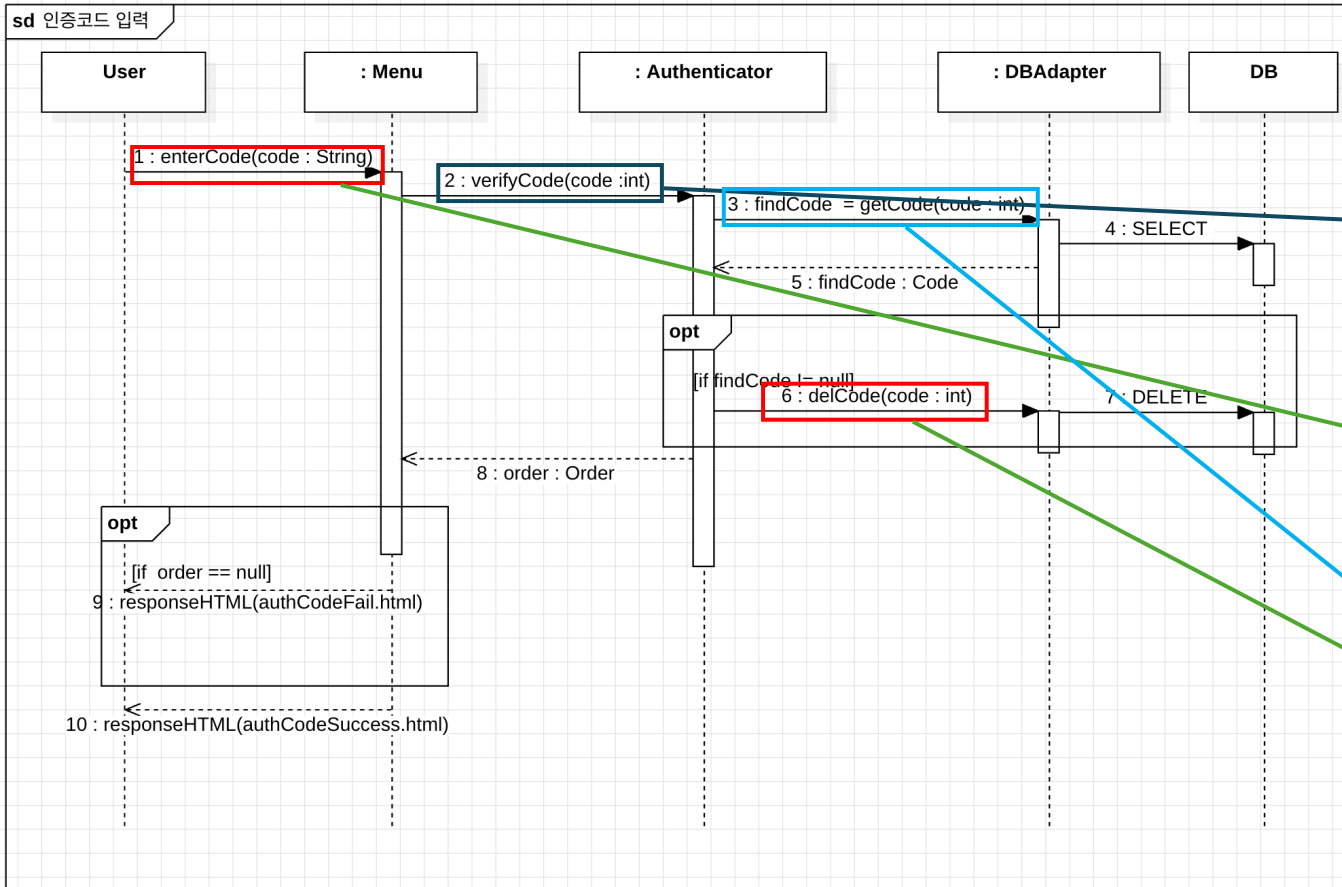
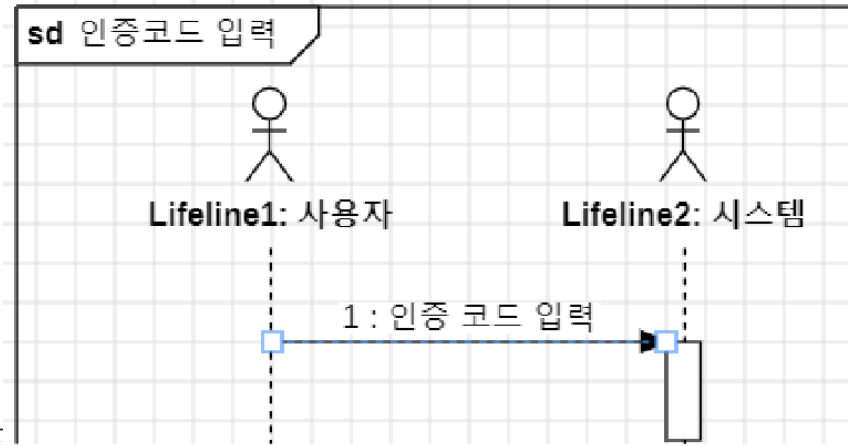


(S) 시스템, (O) 타 자판기

1. (S) : 타 자판기에 재고 요청 broadcast를 한다.
2. (O) : 요청받은 음료의 재고, 위치정보를 보내준다.
3. (S) : 재고, 위치정보를 받는다.

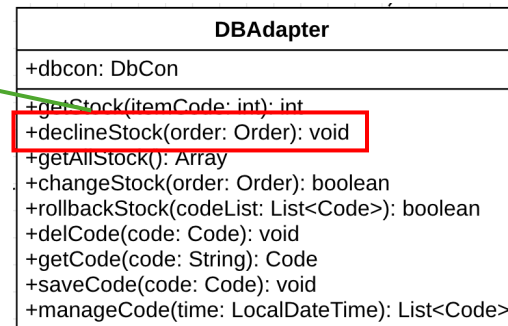
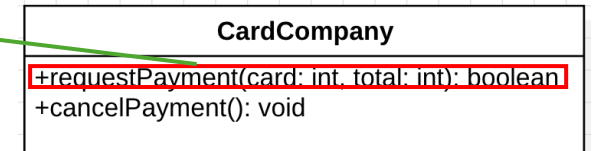
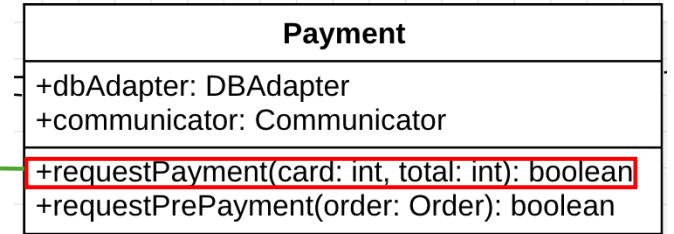
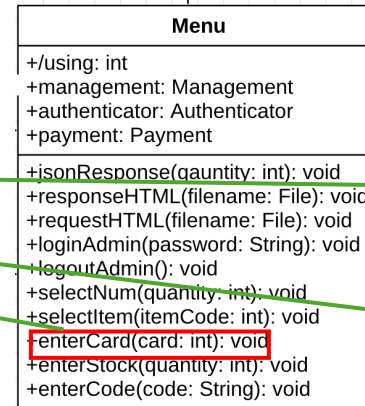
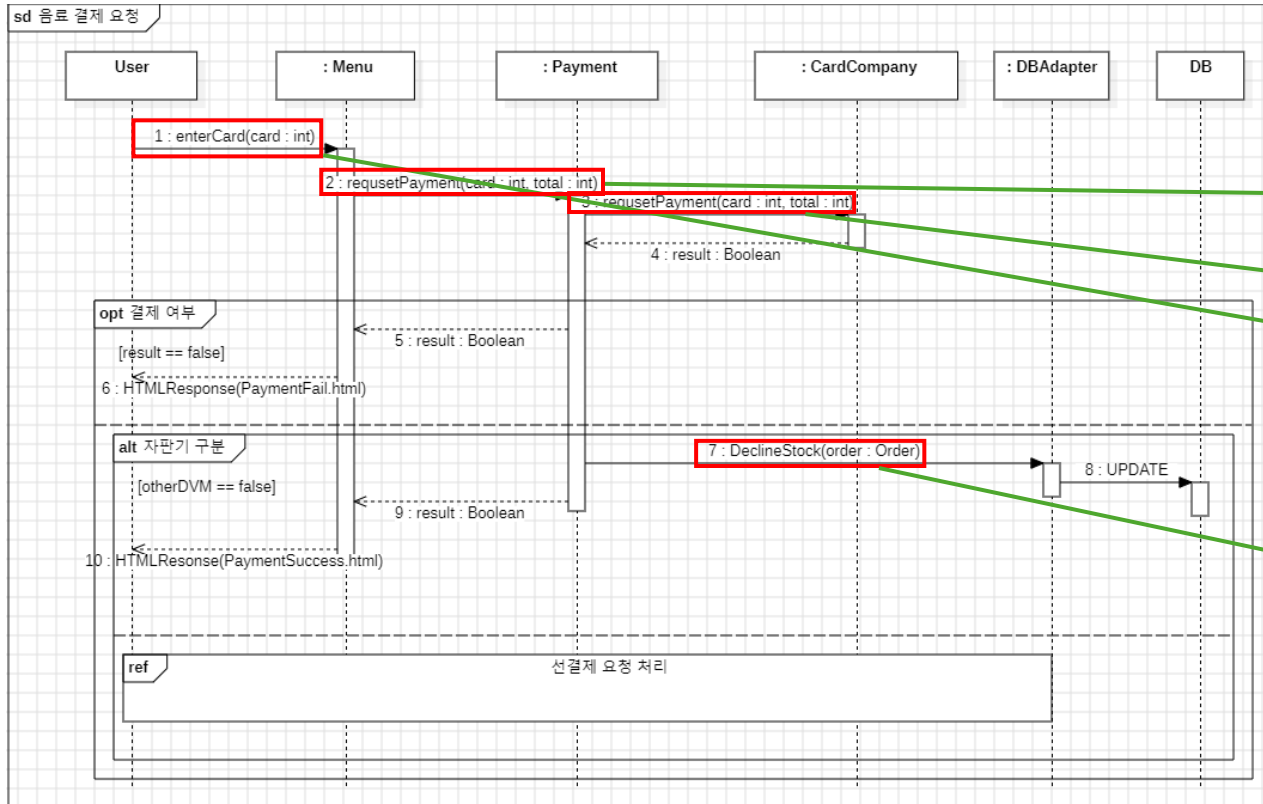
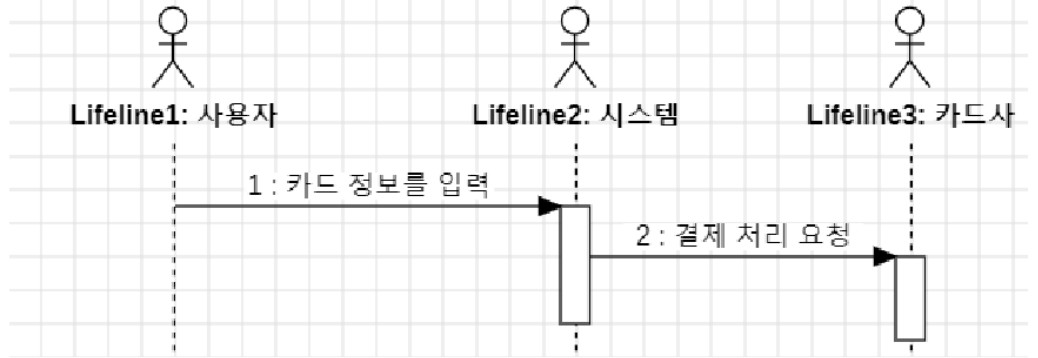


- (S) 시스템, (U) 사용자
 1.(U) 인증코드를 입력한다.
 2.(S) 사용자가 입력한 인증코드가 유효한지 확인한다.
 3.(S) 인증코드가 유효하다면 음료수를 내보낸다.
 4.(U) 나온 음료를 수령한다.



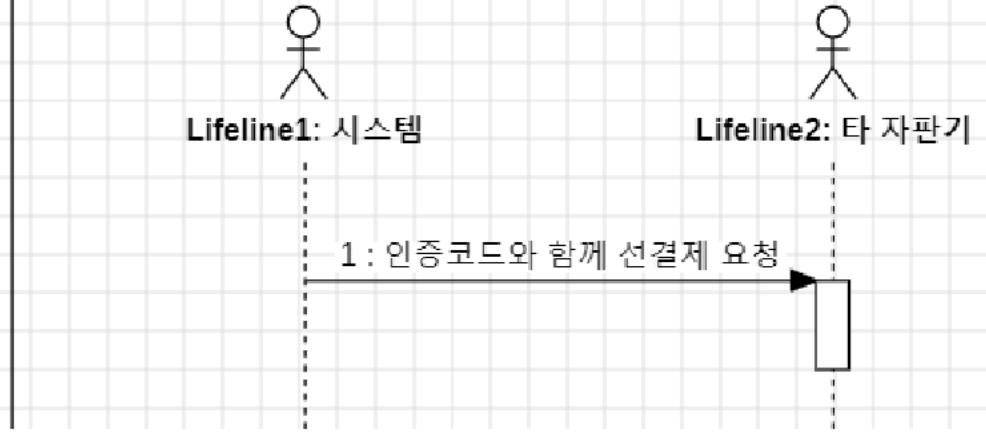
- (S) 시스템, (U) 유저, (C) 카드사
- 1.(S) 선택한 음료 종류와 개수, 총 가격을 출력한다.
 - 2.(U) 출력된 정보를 확인하고 카드 정보를 입력한다.
 - 3.(S) 카드사에 카드 정보와 가격을 전달한다.
 - 4.(S) 카드사에서 결제 처리한다. invoke "결제 처리"
 - 5.(S) 결제가 정상적으로 처리되었다는 메시지를 안내한다.
 - 6.(U) 나온 음료를 수령한다.

sd 음료 결제 요청

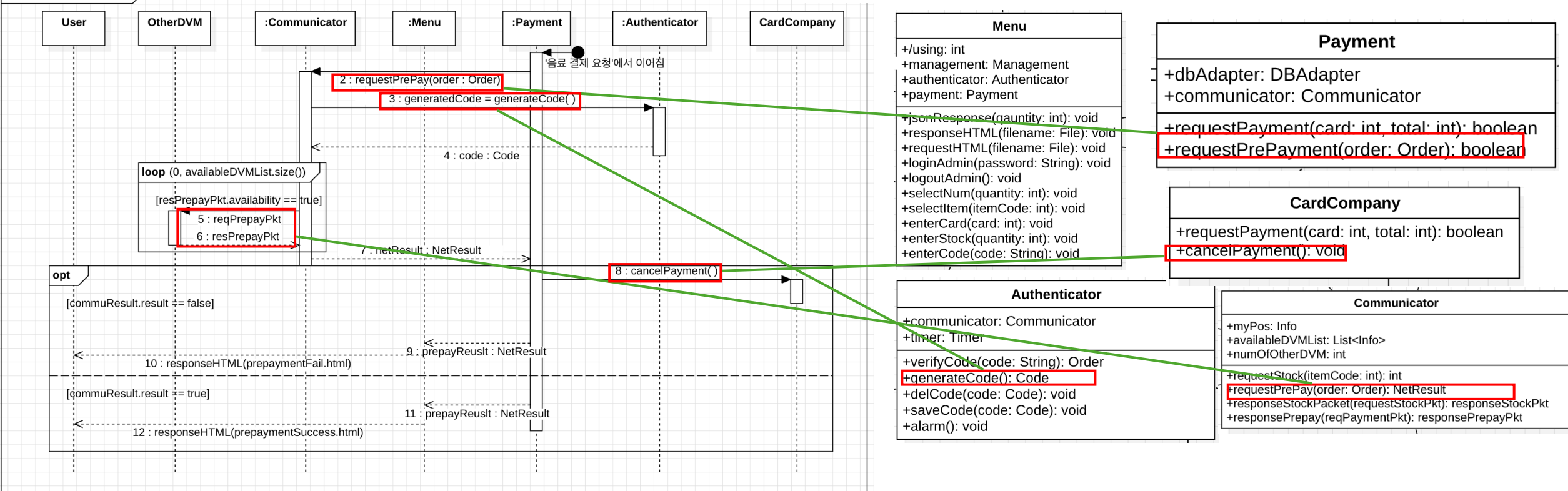


- (S) 시스템, (O) 다른 자판기
- 1.(S) 인증코드를 생성한다.
 - 2.(S) 생성한 인증코드와 음료 개수, 종류를 타 자판기에 전송한다.
 - 3.(O) 음료 재고를 확인하여 결제가 가능한지 응답한다.
 - 4.(S) 결제 가능 응답이 오면 '인증 코드 발급 및 위치 안내'를 부른다.

sd 구매 자판기 선결제 요청



sd 구매 자판기 선결제 요청



Menu

- +using: int
- +management: Management
- +authenticator: Authenticator
- +payment: Payment
- +jsonResponse(quantity: int): void
- +responseHTML(filename: File): void
- +requestHTML(filename: File): void
- +loginAdmin(password: String): void
- +logoutAdmin(): void
- +selectNum(quantity: int): void
- +selectItem(itemCode: int): void
- +enterCard(card: int): void
- +enterStock(quantity: int): void
- +enterCode(code: String): void

Payment

- +dbAdapter: DBAdapter
- +communicator: Communicator
- +requestPayment(card: int, total: int): boolean
- +requestPrePayment(order: Order): boolean

CardCompany

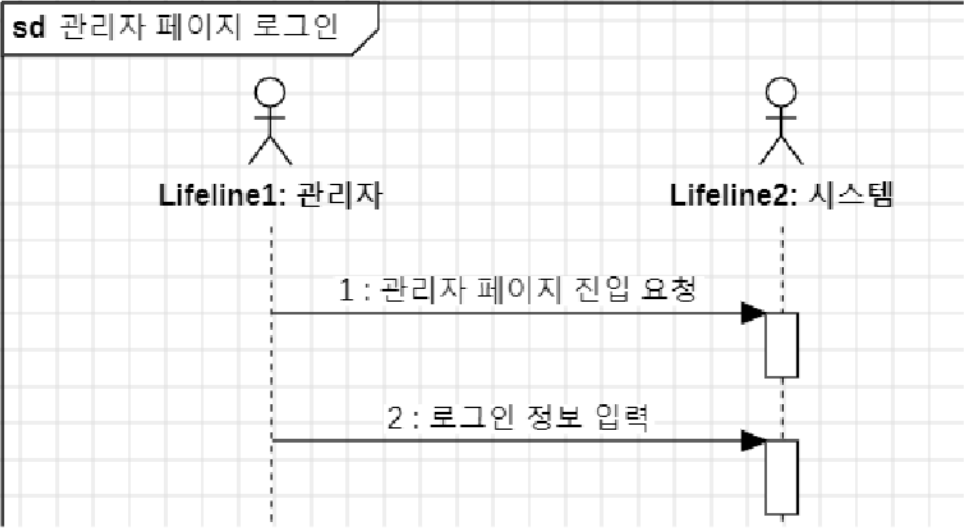
- +requestPayment(card: int, total: int): boolean
- +cancelPayment(): void

Authenticator

- +communicator: Communicator
- +timer: Timer
- +verifyCode(code: String): Order
- +generateCode(): Code
- +delCode(code: Code): void
- +saveCode(code: Code): void
- +alarm(): void

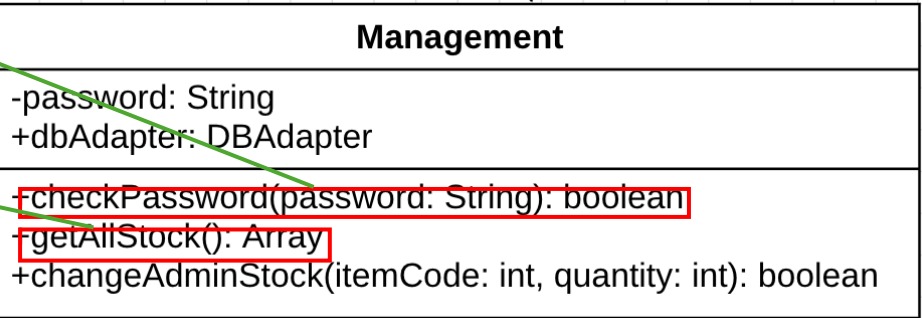
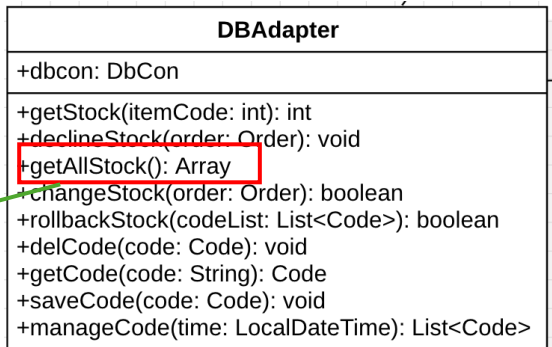
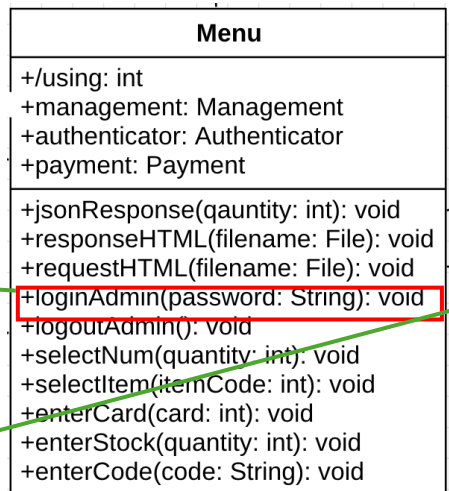
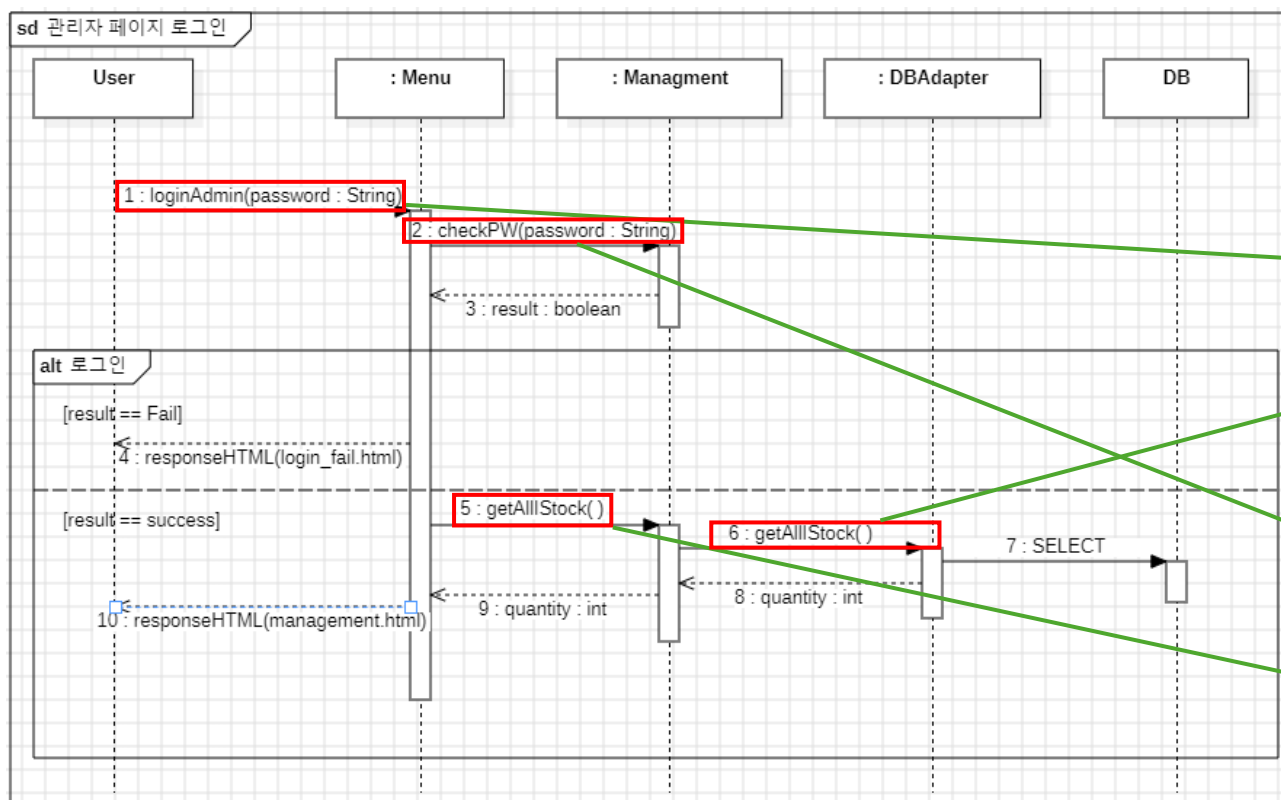
Communicator

- +myPos: Info
- +availableDVMList: List<Info>
- +numOfOtherDVM: int
- +requestStock(itemCode: int): int
- +requestPrePay(order: Order): NetResult
- +responseStockPacket(requestStockPkt): responseStockPkt
- +responsePrepay(reqPaymentPkt): responsePrepayPkt



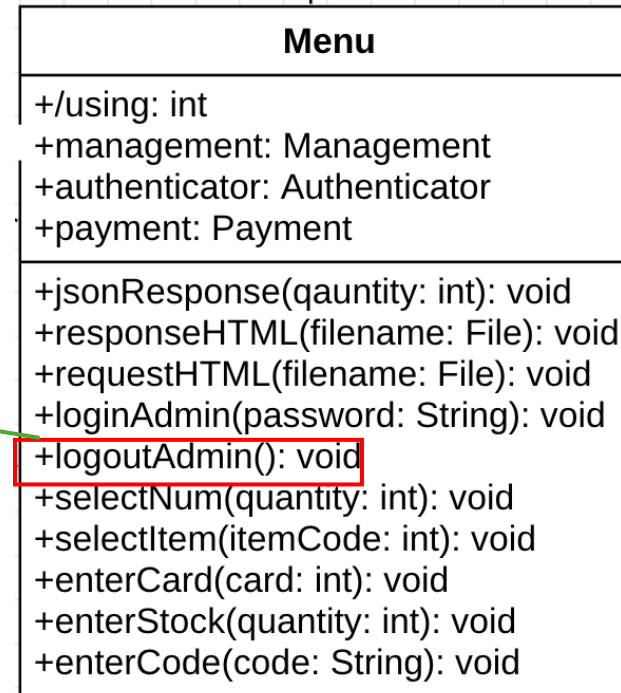
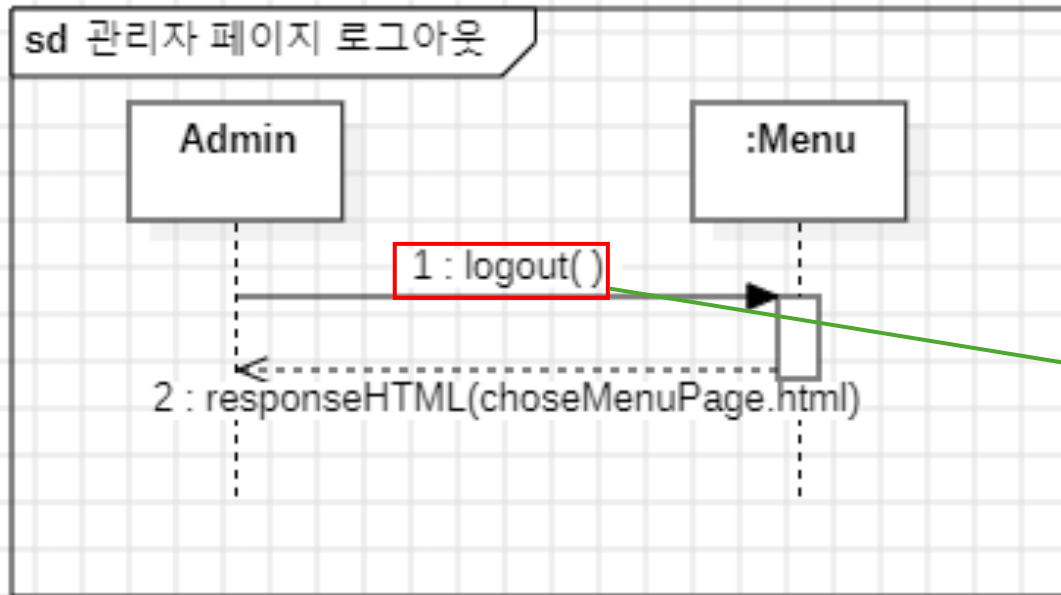
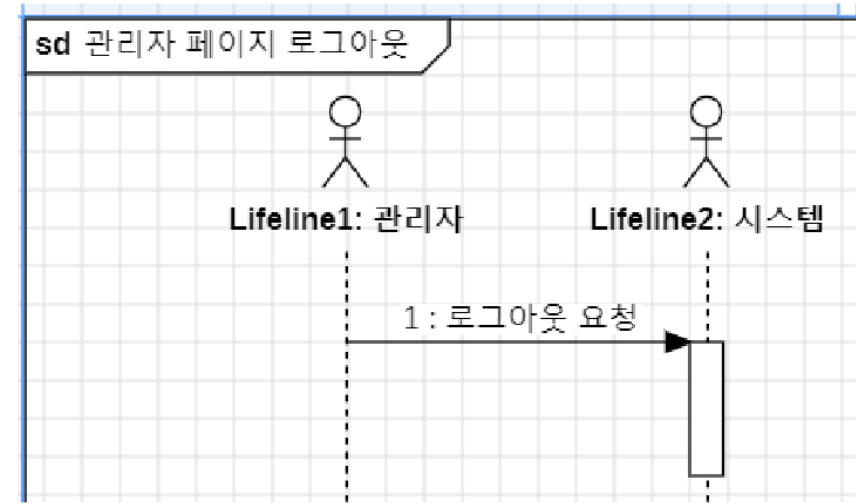
(A) : Admin, (S) : System

1. (A) : 관리자 페이지 진입 버튼을 누른다.
2. (A) : 관리자 페이지를 진입하기 위한 인증 번호를 입력한다.
3. (S) : 관리자 페이지로 이동한다.



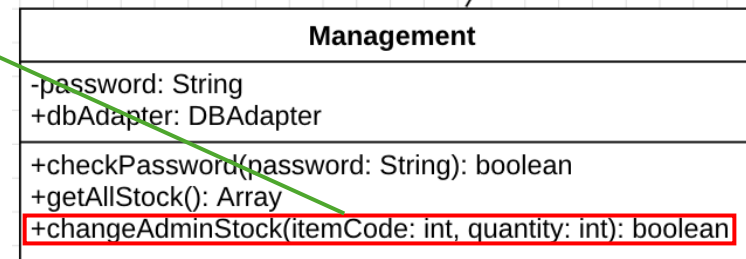
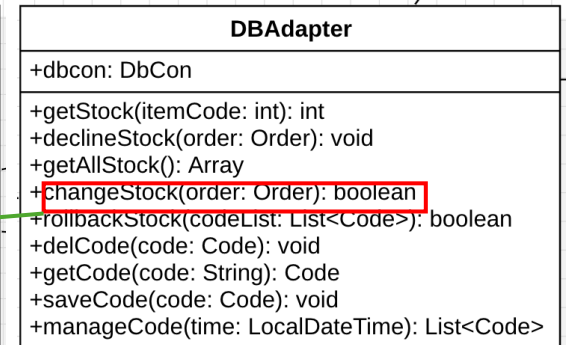
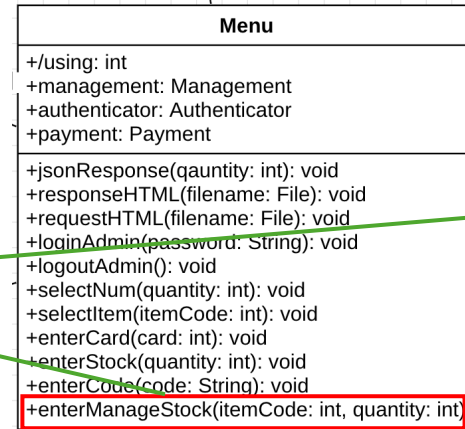
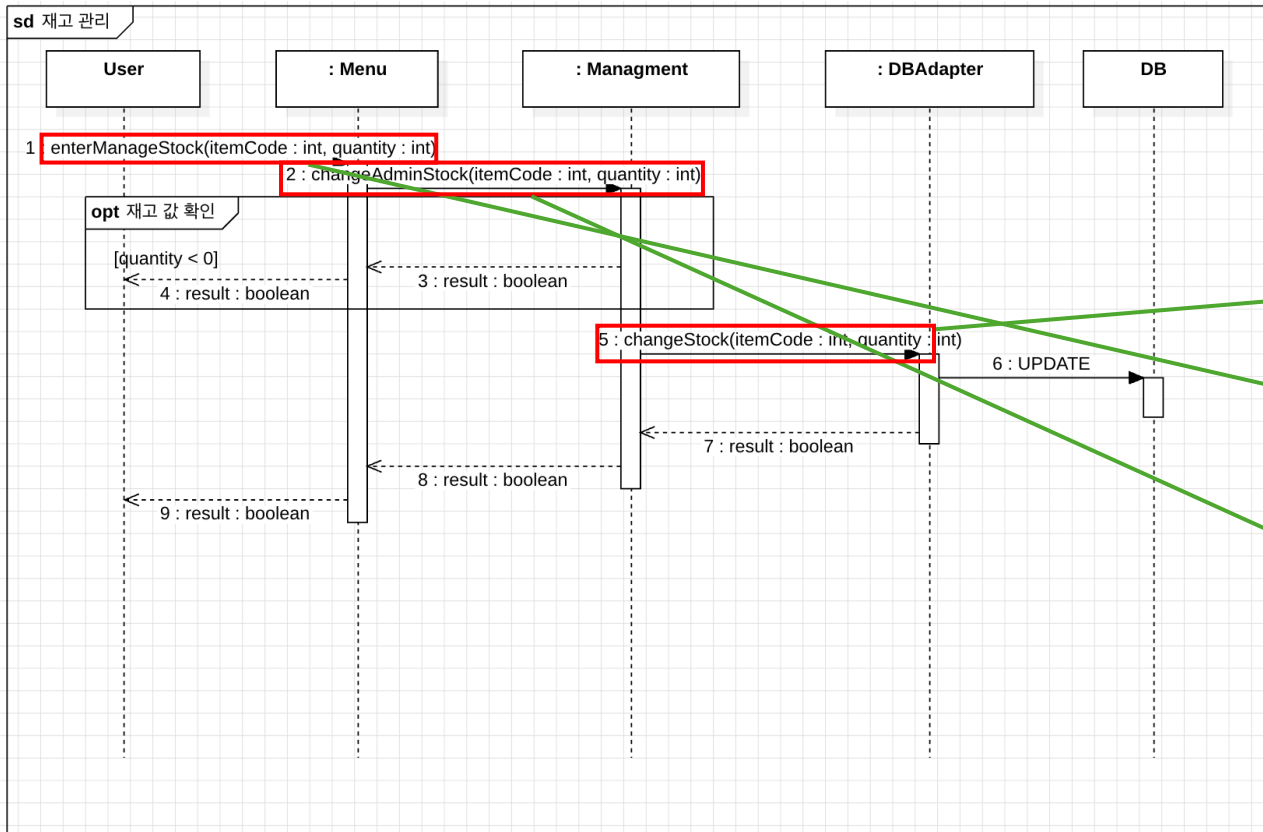
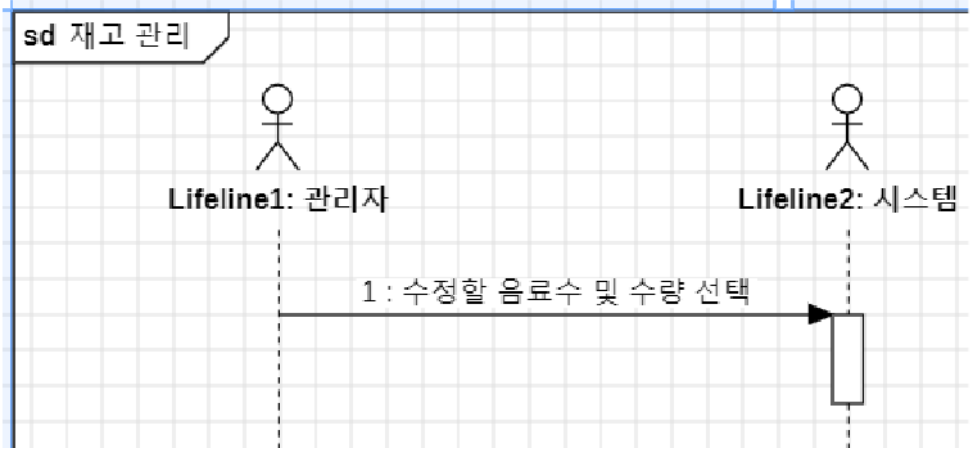
(A) : Admin, (S) : System

1. (A) : 관리자 페이지 로그아웃 버튼을 누른다.
3. (S) : '메뉴 화면 선택'으로 이동한다.



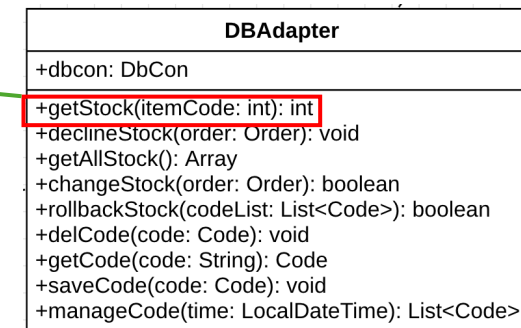
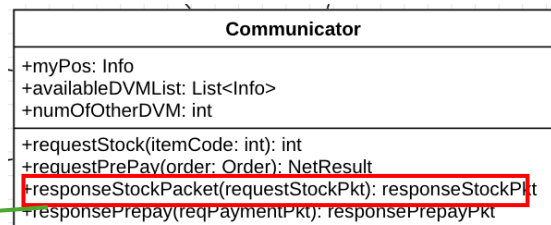
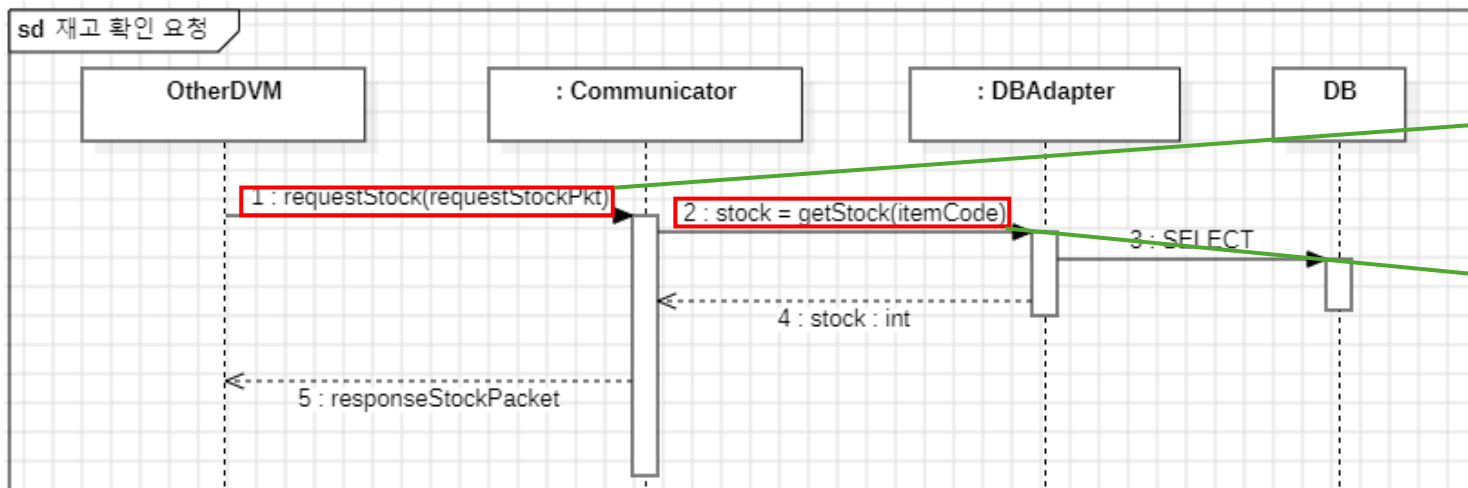
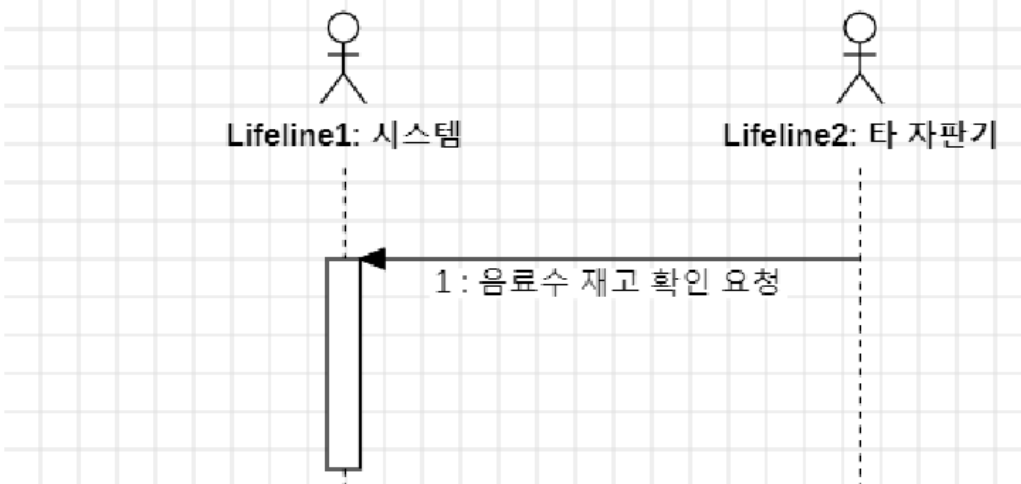
(A) : 관리자, (S) : System

1. (A) 관리자가 음료의 재고와 종류를 선택해서 수정함.
2. (S) 수정한 데이터를 시스템에 반영함.



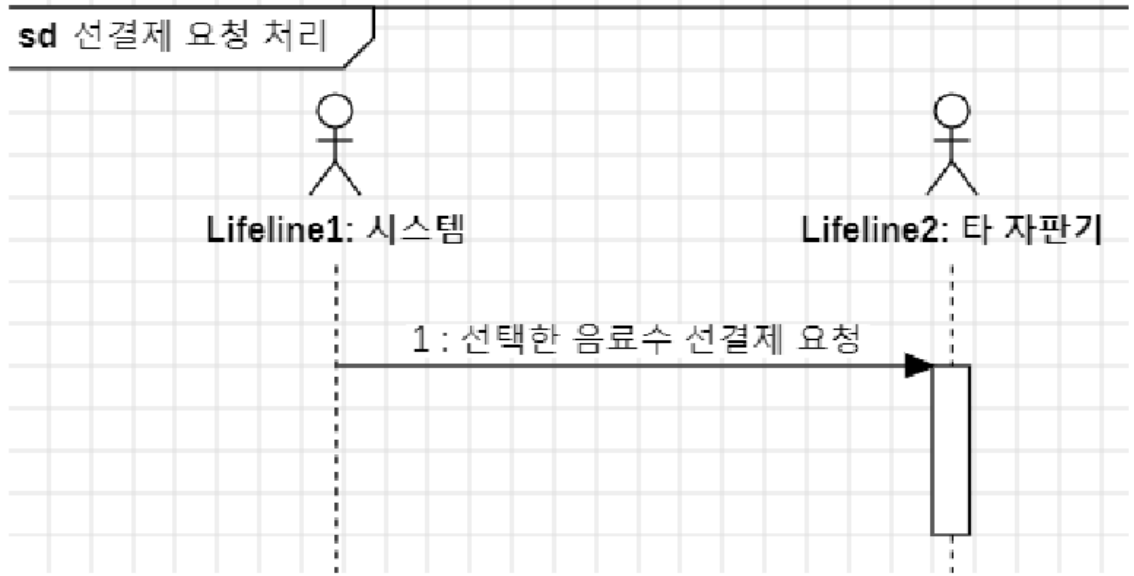
sd 재고 확인 요청

- (O) : 다른 자판기 (S) : 시스템
1. (O) : 재고 확인 요청 메시지 송신
 2. (S) : 재고 확인 요청 메세지 수신
 3. (S) : 시스템의 재고 확인 후 재고 내용 응답

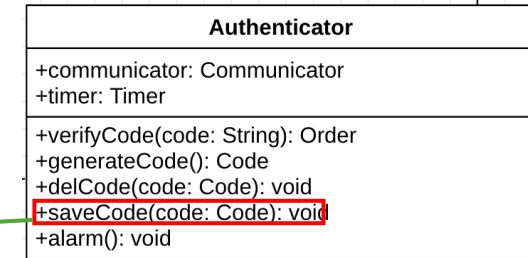
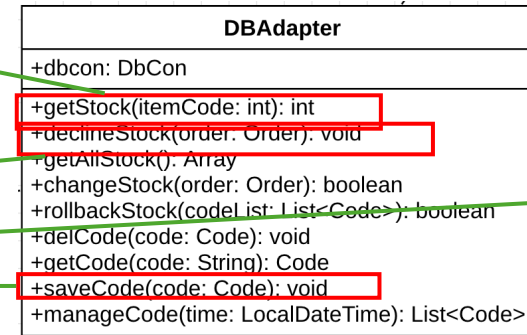
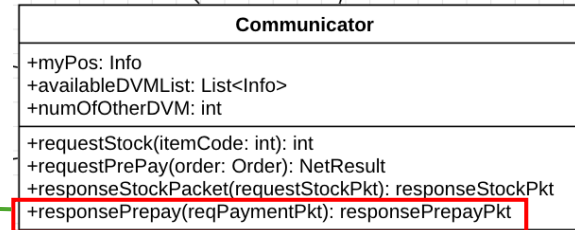
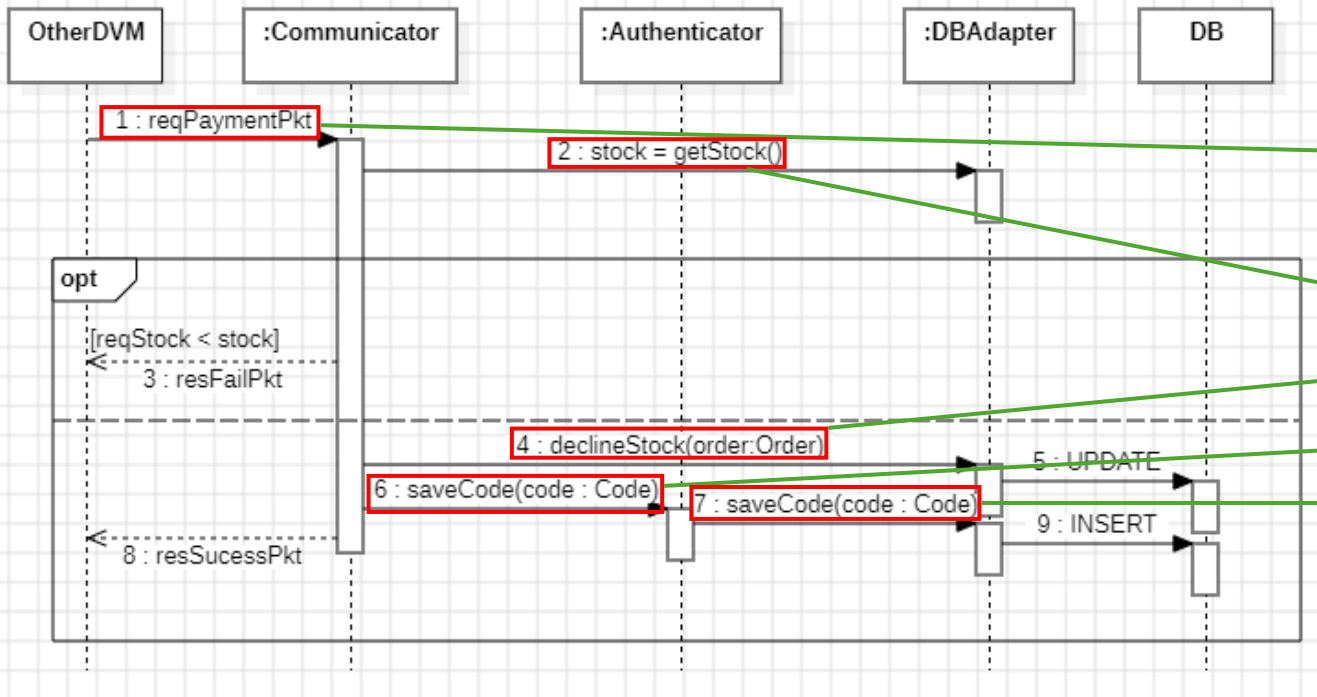


sd 선결제 요청 처리

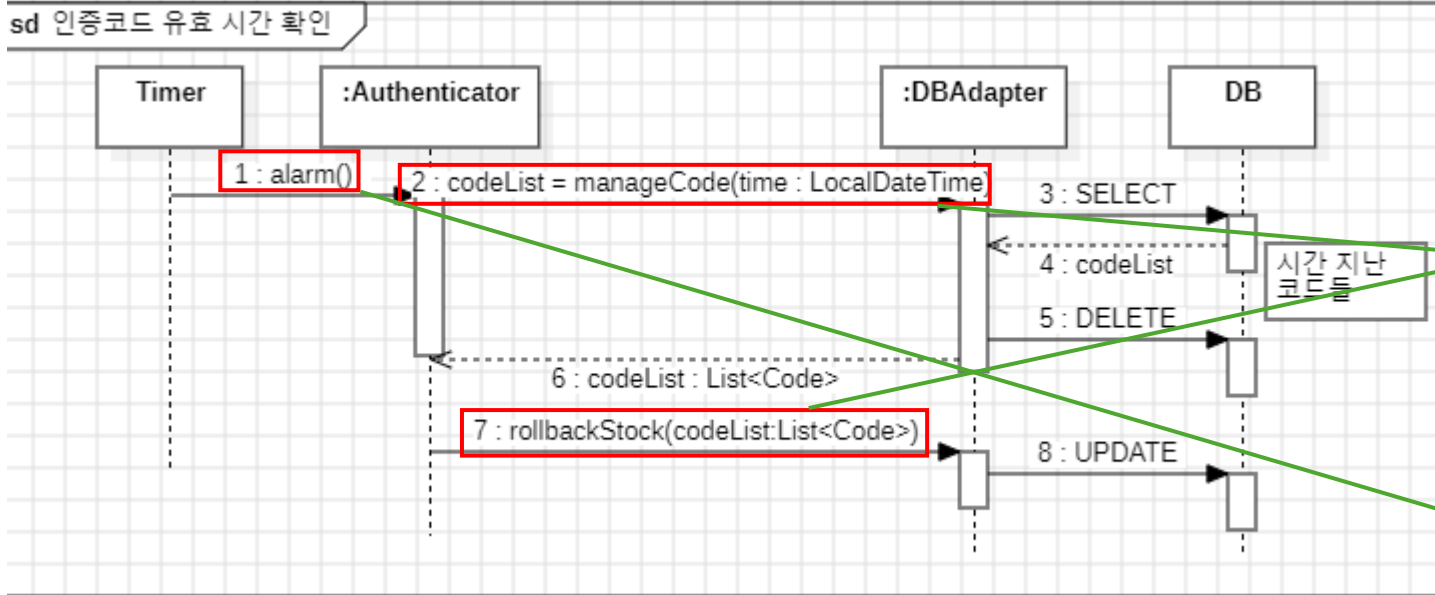
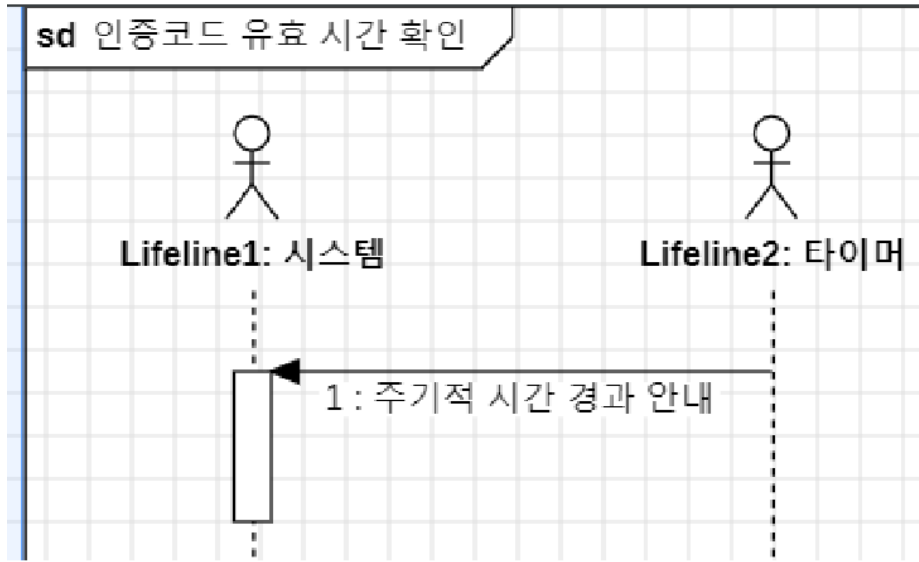
- (O) : 타 자판기 (S) : 시스템
1. (O) : 선결제 요청 메시지 송신.
 2. (S) : 시스템의 재고 확인
 3. (S) : 해당 음료수 재고 차감
 4. (S) : 선결제 가능 여부 응답.
 5. (S) : 선결제 가능 시 인증코드 저장



sd 선결제 요청 처리



- (S) : 시스템, (T) 타이머
1. (T) : 일정 시간마다 유효시간 확인
 2. (S) : 만료 메시지 수신 시 인증코드 파기
 3. (S) : 차감된 음료 개수 복구



DBAdapter
+dbcon: DbCon
+getStock(itemCode: int): int
+declineStock(order: Order): void
+getAllStock(): Array
+changeStock(order: Order): boolean
+rollbackStock(codeList: List<Code>): boolean
+delCode(code: Code): void
+getCode(code: String): Code
+saveCode(code: Code): void
+manageCode(time: LocalDateTime): List<Code>

Authenticator
+communicator: Communicator
+timer: Timer
+verifyCode(code: String): Order
+generateCode(): Code
+delCode(code: Code): void
+saveCode(code: Code): void
+alarm(): void

UseCase	Operation in sequence diagram	Operation in interaction diagram	Method	Class
R0.1 메뉴 화면 선택	1. 음료 구매 페이지 선택	requestHTML(index.html)	jsonResponse(in quantity:int): void	Menu
R1.1 음료 종류 선택	2. 관리자 페이지 선택	requsetHTML(page: String)	responseHTML(in filename:File): void	
R1.2 현재 자판기 음료 개수 선택	3. 구매할 음료수 종류를 선택	selectItem(itemCode :int)	requestHTML(in filename:File): void	
R1.3 타 자판기 음료 개수 선택	4. 구매할 음료수 개수를 입력	getStock(itemCode: int)	loginAdmin(in password:String): void	
R1.4 타 자판기 재고 확인 요청	5. 타 자판기 음료 개수 선택	JsonResponse(quantity : int)	logoutAdmin(): void	
R1.5 인증코드 입력	6. 타 자판기 재고확인 요청	selectNum(num : int)	selectNum(in quantity:int): void	
R1.6 음료 결제 요청	7. 인증코드 입력	selectNum(num)	selectItem(in itemCode:int): void	
R1.7 구매 자판기 선결제 요청	8. 카드 정보를 입력	requestBroadcast(itemCode: int)	enterCard(in card:int): void	
R2.1 관리자 페이지 로그인	9. 결제 처리 요청	RequestStockPkt	enterStock(in quantity:int): void	
R2.2 관리자 페이지 로그아웃	10. 인증코드와 함께 선결제 요청	enterCode(code : String)	enterCode(in code:String): void	
R2.3 재고관리	11. 관리자 페이지 진입 요청	verifyCode(code :int)	checkPassword(in password:String): boolean	
R3.1 재고 확인 요청	12. 로그인 정보 입력	getCode(code : int)	getAllStock(): Array	
R3.2 선결제 요청 처리	13. 로그아웃 요청	enterCard(card : int)	changeStock(in order:Order): boolean	
R3.3 인증코드 유효 시간 확인	14. 수정할 음료수 및 수량 선택	requsetPayment(card : int, total : int)	changeAdminStock(in itemCode:int, in quantity:int): boolean	
	15. 음료수 재고 확인 요청	requsetPayment(card : int, total : int)	getStock(in itemCode:int): int	
	16. 선결제 요청 처리	DeclineStock(order : Order)	getCode(in code:int): Code	
	17. 주기적 시간 경과 안내	requestPrePay(order : Order)	declineStock(in order:Order): void	
		generateCode(order : Order)	saveCode(in code:Code): void	DBAdapter
		reqPrepayPkt	delCode(in code:Code): void	
		cancelPayment()	getAllStock(): Array	
		delCode(code : Code)	changeStock(in order:Order): boolean	
		loginAdmin(password : String)	manageCode(in time:LocalDateTime): List<Code>	
		checkPW(password : String)	rollbackStock(in codeList:List<Code>): boolean	
		getAlIStock()	verifyCode(in code:int): Order	
		getAlIStock()	generateCode(in order:Order): Code	
		logout()	delCode(in code:Code): void	
		enterStock(order : Order)	saveCode(in code:Code): void	
		changeAdminStock(type, quantity)	requestPayment(in card:int, in total:int): boolean	Payment
		changeStock(type, quantity)	requestPrePayment(in order:Order): boolean	
		requestStock(requestStockPkt)	requestStock(in itemCode:int): int	Communicater
		getStock(itemCode)	requestPrePay(in order:Order): CommuResult	
		reqPaymentPkt	responseStockPacket(in requestStockPkt): responseStockPkt	
		getStock()	responsePrepay(in reqPaymentPkt): responsePrepayPkt	Card Company
		declineStock(order:Order)	requestPayment(in card:int, in total:int): boolean	
		saveCode(code : Code)	cancelPayment(): void	
		saveCode(code : Code)		
		manageCode(time : LocalDateTime)		
		rollbackStock(codeList:List<Code>)		