
OOPT Stage 2040 Analysis

202112348 조영탁
202211282 김종혁
202312373 정석현

목차

2041 Design Real Use Cases

2042 Define Reports, UI, and Storyboards

2043 Refine System Architecture

2044 Define Interaction Diagrams

2045 Define Design Class Diagrams

2046 Define Traceability Analysis

2041 UseCase #01

Use Case	메뉴 보여주기
Actors	User
Purpose	유저에게 음료의 선택지 및 재고를 보여줌
Overview	사용자가 DVM에 시작을 입력하면 DVM은 현재 재고 정보를 확인하여 화면에 이를 출력한다
Type	Primary
Cross Reference	Functions: R1.1 Use Cases:
Pre-Requisites	시스템에 음료의 재고 정보가 존재
Typical Course of Events	(A): Actor, (S) System 1.(A) 화면에 시작을 입력함 2. (S) Stock에 현재 자판기의 재고 정보를 요청한다 3. (S) Stock에서 반환받은 재고정보를 바탕으로 20가지의 음료와 각 수량을 화면에 출력한다
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

2041 UseCase #02

Use Case	음료 선택
Actors	User
Purpose	유저가 구매하고자 하는 음료를 선택한다
Overview	사용자는 화면에 있는 음료와 수량을 확인하고 화면에 이를 입력한다. 현재 자판기 내에 존재하는 음료와 수량이라면 결제 단계로 넘어간다 현재 자판기에 존재하지 않은 음료라면 선결제 단계로 넘어간다
Type	Primary
Cross Reference	Functions: R1.2 Use Cases:
Pre-Requisites	N/A
Typical Course of Events	(A): Actor, (S): System 1. (A) 구매하고자 하는 음료의 코드와 갯수를 입력한다 2. (S) 입력 받은 값과 Stock에 저장되어 있는 재고를 비교한다. 3.(S) 현재 자판기에서 입력받은 값 만큼 음료를 구매할 수 있다면 결제 단계를 진행한다. 4.(S) 현재 자판기에서 구매할 수 없는 수량이라면 선결제 요청 메시지를 다른 자판기들에게 전송한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1. 제공하지 않는 음료 입력 시 재입력을 요구한다 E2. 재고 이상의 수량을 선택 시 재입력을 요구한다 E3. 음수값 입력 시 처리하지 않고 에러를 반환한다.

2041 UseCase #03

Use Case	결제
Actors	User, Bank
Purpose	음료의 종류와 갯수에 해당하는 금액을 결제한다
Overview	유저가 카드 번호를 입력하면 이전에 입력했던 음료 종류와 갯수를 토대로 가격을 산정해 이를 Bank에 전달해 유효성 검사를 진행한다. 결제가 성공할 시에 선결제 과정이 아니었다면 음료를 반환하고, 선결제 과정이었다면 선결제 과정의 나머지 작업을 진행한다.
Type	Primary
Cross Reference	Functions: R2.1, R2.2, R2.3 Use Cases: "음료 선택", "음료 제공", "음료 선결제"
Pre-Requisites	음료 선택이 완료되어야 한다
Typical Course of Events	(A): Actor, (S): System, (B): Bank 1. (A) 카드 번호를 입력한다. 2. (S) Bank에 카드번호 및 결제 금액을 전달한다 3.(B) 카드 번호 및 금액을 확인 후 결제 유효성을 반환한다. 4. (B) 결제가 성공할 시 카드의 잔고를 차감한다. 5. (S) 선결제 과정이 아니었다면 음료를 제공한다. 6.(S) 선결제 과정 중이었다면 나머지 선결제 과정을 진행한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1. 카드 번호가 올바르지 않을 경우 결제 실패를 반환한다 E2. 카드 잔액이 부족한 경우 결제 실패를 반환한다,

2041 UseCase #04

Use Case	음료 제공
Actors	User
Purpose	현재 자판기에서 결제한 음료를 제공한다
Overview	이용자가 현재 자판기에 존재하는 음료를 선택하고 결제가 성공 했을 시 자판기로부터 음료를 제공받는다
Type	Primary
Cross Reference	Functions: R2.5, R5.4 Use Cases: "결제"(UC03)
Pre-Requisites	음료 결제 과정이 정상적으로 완료되어야 한다
Typical Course of Events	(A): Actor, (S): System 1. (A) 유저가 정상적으로 결제과정을 완료했다. 2.(S) 결제한 만큼의 음료 재고를 차감한다. 3. (S) 결제가 완료된 음료를 Actor에게 제공한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

2041 UseCase #05

Use Case	음료 선결제
Actors	User, Other DVM
Purpose	타 자판기에 존재하는 음료를 현재 자판기에서 선결제
Overview	음료 선택 후 현재 자판기에 재고가 존재하지 않으면 시작된다. 음료 별 가까운 자판기와 재고를 출력하여 유저가 이를 선택하면 결제(UC03)을 진행한다. 결제가 성공하면 인증번호를 유저에게 출력해준다
Type	Primary
Cross Reference	Functions: R 3.1, R3.2, R3.3, R3.4, R4.1, R5.1, R5.2 Use Cases: "음료 선택"(UC02), "결제"(UC03)
Pre-Requisites	음료를 선택했을 때 현재 자판기에 재고가 존재하지 않아야 한다.
Typical Course of Events	(A): Actor, (O): Other DVM, (S): System 1. (A) 현재 자판기에 존재하지 않는 음료를 선택하거나, 현재 자판기에 존재하는 수량을 초과하는 수치를 입력한다. 2. (S) 다른 자판기에게 입력한 음료의 종류와 수량을 만족하는지 메시지를 브로드캐스트로 전송한다 3. (S) 선결제 구매가능 메시지를 받은 자판기 중 가장 가까운 자판기 위치를 계산한다 4. (A) 사용자가 위치를 확인한 후 입력 창에 Y를 누른다 5. (S) 결제를 진행한다 6.(S) 인증코드를 반환한다
Alternative Courses of Events	A1. 사용자가 위치를 확인한 후 선결제를 취소하면 홈 화면으로 돌아간다.
Exceptional Courses of Events	E2. 결제(UC03)에서 예외가 발생할 시 해당 케이스에서 예외에 대한 이벤트를 진행한다.

2041 UseCase #06

Use Case	선결제 음료 제공
Actors	User
Purpose	선결제 여부가 확인된 음료를 제공
Overview	선결제 코드가 유효한 경우 사용자에게 음료를 반환한다
Type	Primary
Cross Reference	Functions: R5.4, R6.1 Use Cases:
Pre-Requisites	선결제 된 음료에 대한 인증코드가 존재해야한다.
Typical Course of Events	(A): Actor, (S): System 1.(A): 사용자가 홈 화면에서 인증코드를 입력한다. 2. (S): 해당 인증코드와 매핑되어 있는 음료가 존재하는지 확인한다 3. (S): 인증코드가 유효하면 음료를 제공한다.
Alternative Courses of Events	A1. 유효하지 않은 번호를 입력한 경우 재입력을 요청한다
Exceptional Courses of Events	N/A

2041 UseCase #07

Use Case	재고확인
Actors	User
Purpose	자판기에 존재하는 음료의 종류와 재고를 알 수 있다
Overview	고객이 음료의 종류와 개수를 선택하면 현재 자판기와 다른 자판기의 재고를 확인한다
Type	Primary
Cross Reference	Functions: R2.2, R3.1 Use Cases: "음료 선택"(UC02)
Pre-Requisites	고객의 음료 종류, 개수 선택 (UC02)
Typical Course of Events	(O): Other DVM, (S): System 1. 다른 자판기에서 음료 선택(UC02) 과정이 진행된다. 2. (O) 다른 자판기가 재고 상황 현황을 요구한다. 3.(S) 자판기에서 재고 상황을 파악 후 메시지를 전송한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1. 시스템 점검(Admin의 작업)시 재고확인 요청을 차단

2041 UseCase #08

Use Case	재고 수정
Actors	Admin
Purpose	음료를 채워 재고를 확보한다
Overview	관리자가 DVM 시스템을 정지하고 음료를 채워놓은 후 시스템을 재실행한다
Type	Primary
Cross Reference	Functions: R5.4 Use Cases:
Pre-Requisites	재고 수정을 하고자 하는 자판기가 현재 작업을 처리하고 있지 않아야 한다.
Typical Course of Events	(A): Admin, (S): System 1.(A) 관리자 모드로 접속한다 2. (S) 현재 자판기 재고에 대한 구매 이벤트를 차단한다. 3. (A) 관리자가 선택한 음료수를 입력한 개수만큼 재고를 업데이트 한다. 4. (A) 관리자 모드를 해제한다 5.(S) 자판기의 구매 이벤트를 활성화한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1. 이벤트를 처리 중인 자판기에 대해 관리자가 접근을 시도 시 접속

2041 UseCase #09

Use Case	인증번호 생성
Actors	User, Other DVM
Purpose	선결제한 음료에 대한 인증 번호 생성 후 해당 자판기에 메시지를 전달한다
Overview	타 자판기로부터 선결제 완료 메시지가 올 경우 인증번호를 생성한다, 현재 생성된 인증번호 중 중복되는 번호가 있는지 확인 후 문제가 없으면 전송한다. 인증번호를 내부 저장소에 저장한다.
Type	Secondary
Cross Reference	Functions: R5.2 Use Cases: "음료 선결제"(UC05)
Pre-Requisites	타자판기에서 선결제가 완료되어야 한다
Typical Course of Events	(O): Other DVM, (S): System 1. (O) 타 자판기에서 선결제 완료 후 인증번호 요구 메시지를 전송한다 2. (S) 인증번호를 생성한다 3. (S) 인증번호를 Other DVM에 전송한다 4.(S) 동시에 재고를 업데이트 한다
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1. 인증번호가 중복될 경우 재생성한다

2041 UseCase #10

Use Case	타자판기와의 통신
Actors	Other DVM
Purpose	현재 재고 상황과 위치를 다른 자판기로 전송하거나 다른 자판기로부터 받아온다
Overview	타자판기로부터 재고 상황을 요청받은 경우 현재 재고 상태 확인 후 위치와 함께 메시지를 전송한다. 또는 재고가 없는 경우 타 자판기로부터 현황 메시지를 요구한다.
Type	Secondary
Cross Reference	Functions: R3.1, R5.1 Use Cases: "재고 확인"(UC07)
Pre-Requisites	N/A
Typical Course of Events	(O): Other DVM, (S): System 1. (O) 재고확인을 요청한다. 2. (S) 재고 확인 후 재고와 위치를 담은 메시지를 전송한다.(UC07) 3. (S) 재고가 존재하지 않을 시 broadcast 전송한다. 4. (O) 메시지를 수신하고 응답 메시지를 송신한다.
Alternative Courses of Events	A1: 메시지를 송신하지 않는 자판기가 존재 시 요청 메시지를 재전송 한다.
Exceptional Courses of Events	E1: 요구 시간을 초과하여도 특정 자판기로부터 메시지를 받지 못할 경우 해당 자판기를 제외한다.

1. 홈 화면

— — — — — — — — —

어서오세요.

시작하려면 시작을 입력하세요...

(선결제 인증코드 입력을 원하시면 인증코드를 입력하세요...)

— — — — — — — — —

>_

2. 음료 창

— — — — — — — — —

콜라 : 1 사이다 : 0 녹차 : 1 홍차 : 4 밀크티 : 1 탄산수 : 1 보리차 : 0 캔커피 : 3 물 : 0 유자차 : 1

에너지 드링크 : 3 식혜 : 0 아이스티 : 0 딸기주스 : 1 오렌지 주스 : 1 포도 주스 : 0 이온 음료 : 0 아메리카노 : 0 핫초코 : 0 카페라떼 : 0

음료 이름과 갯수를 입력해주세요. (e.g. 콜라 2)

— — — — — — — — —

>_

3. 로딩

— — — — — — — — —

#####

가장 가까운 자판기 찾는 중...

— — — — — — — — —

>_

4. 타자판기 위치 출력

— — — — — — — — —
-- . -- . -- . -- . -- . -- . -- .
-- . -- . -- . -- . -- . -- . -- .
-- . -- . -- . -- . -- . -- . -- .
-- . -- . -- . -- . -- . -- . -- .
-- . -- . -- . -- . -- . -- . -- .
-- . -- . -- . -- . -- . -- . -- .

• 현재 위치 • 목표 위치

선결제를 진행하시려면 Y / 홈으로 돌아가시려면 N를 입력해주세요.

— — — — — — — — —
>_

5. 결제

카드 번호를 입력해주세요. (e.g. xxxx xxxx xxxx xxxx)

>_

6. 결제 성공

— — — — — — — — —

(결제가 완료되었습니다.) (인증 코드는 **ax9cb**)
감사합니다.
(음료를 챙겨가세요.)

— — — — — — — — —

>_

7. 인증코드 입력

인증코드를 입력하세요. [a-zA-Z0-9]*

>_

8. 결제, 인증번호 오류

— — — — —

유효하지 않은 (카드 번호/인증 번호)입니다. 다시 한번 카드 번호를 입력해 주세요.

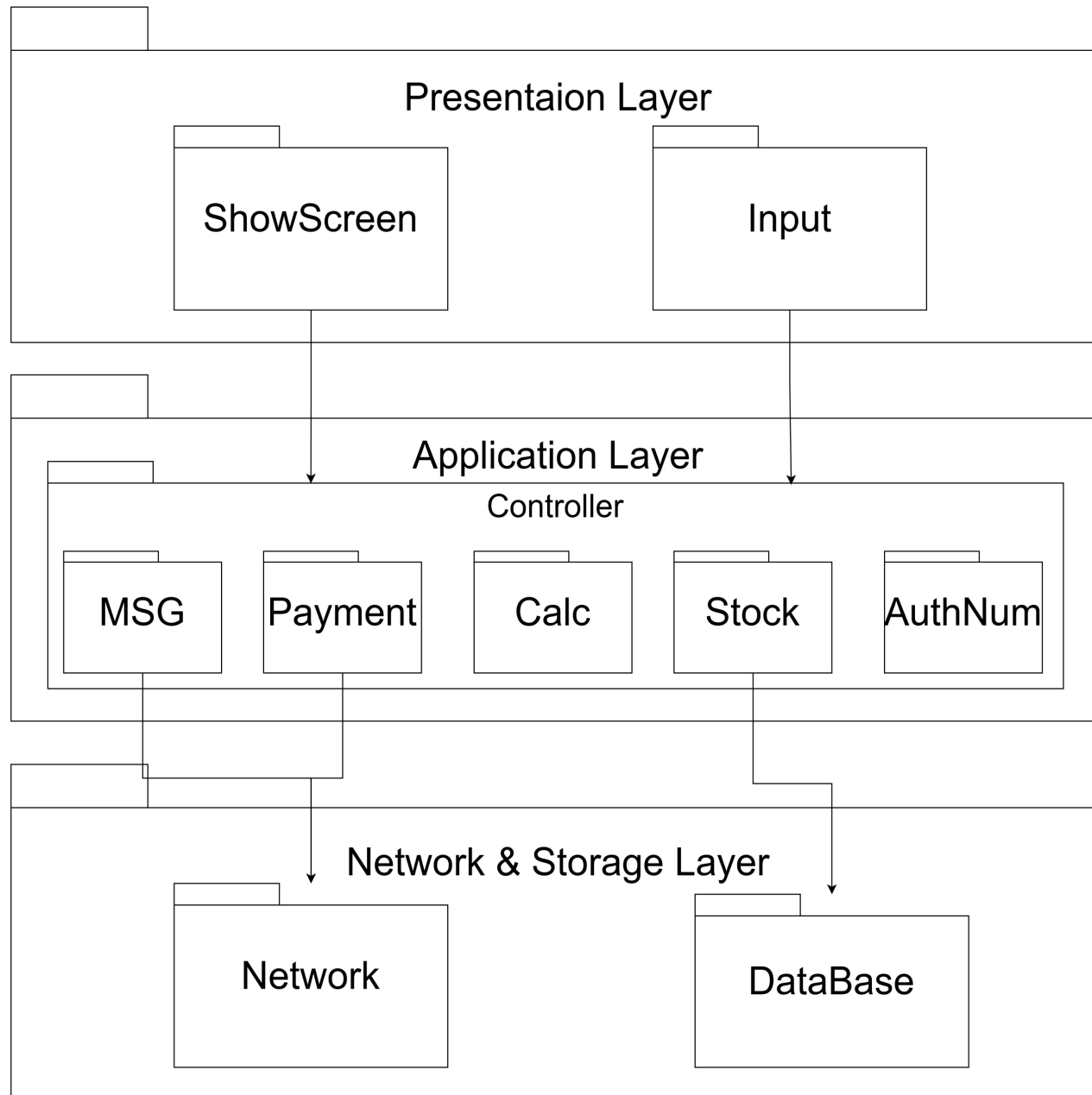
(e.g. xxxx xxxx xxxx xxxx)

or

(e.g. ab0c9)

— — — — —

>_

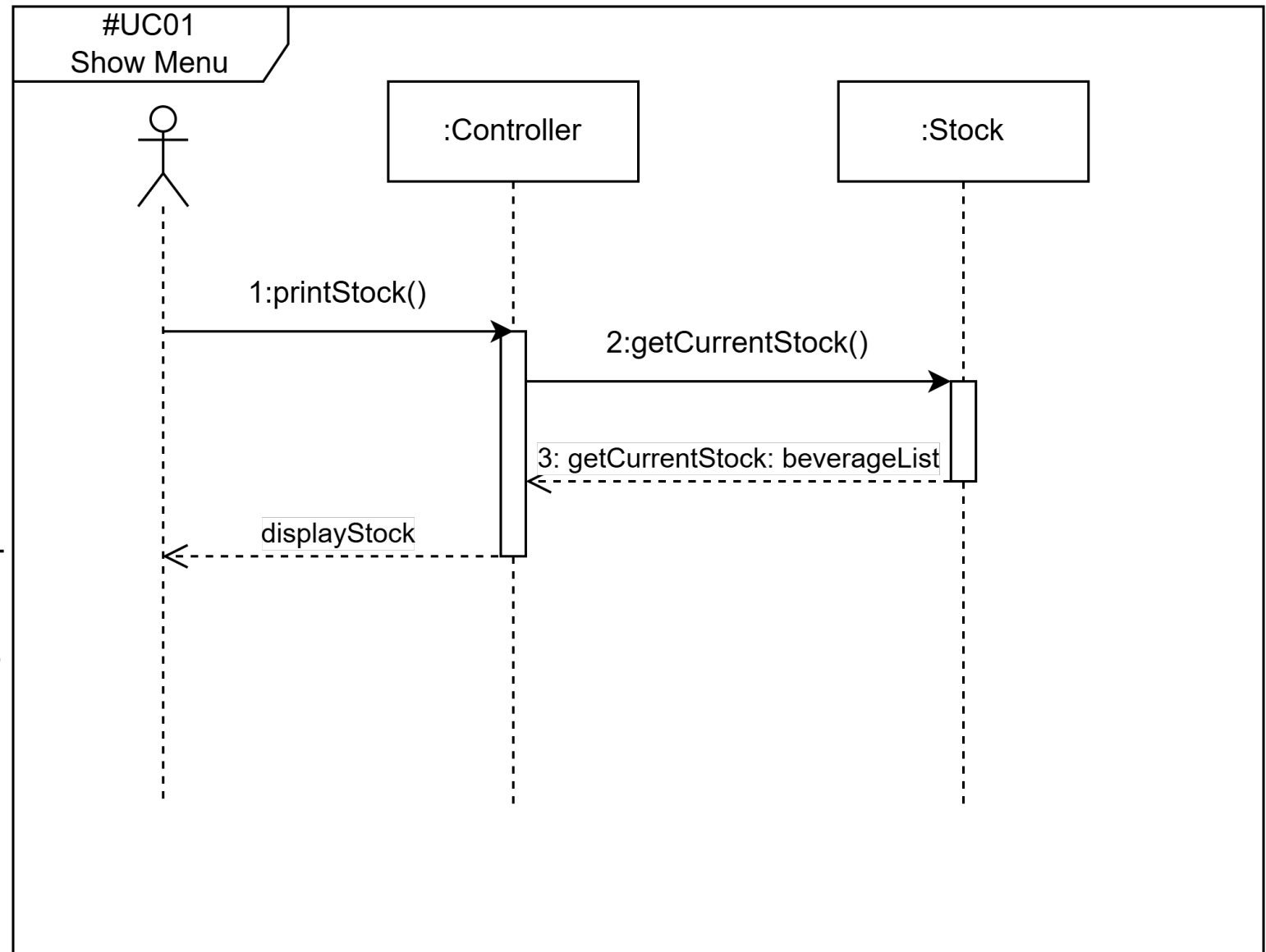


1. 메뉴 보여주기

1.(A) 화면에 시작을 입력함

2. (S) Stock에 현재
자판기의 재고 정보를 요청

3. (S) Stock에서 반환받은
재고정보를 바탕으로
20가지의 음료와 각 수량을
화면에 출력한다



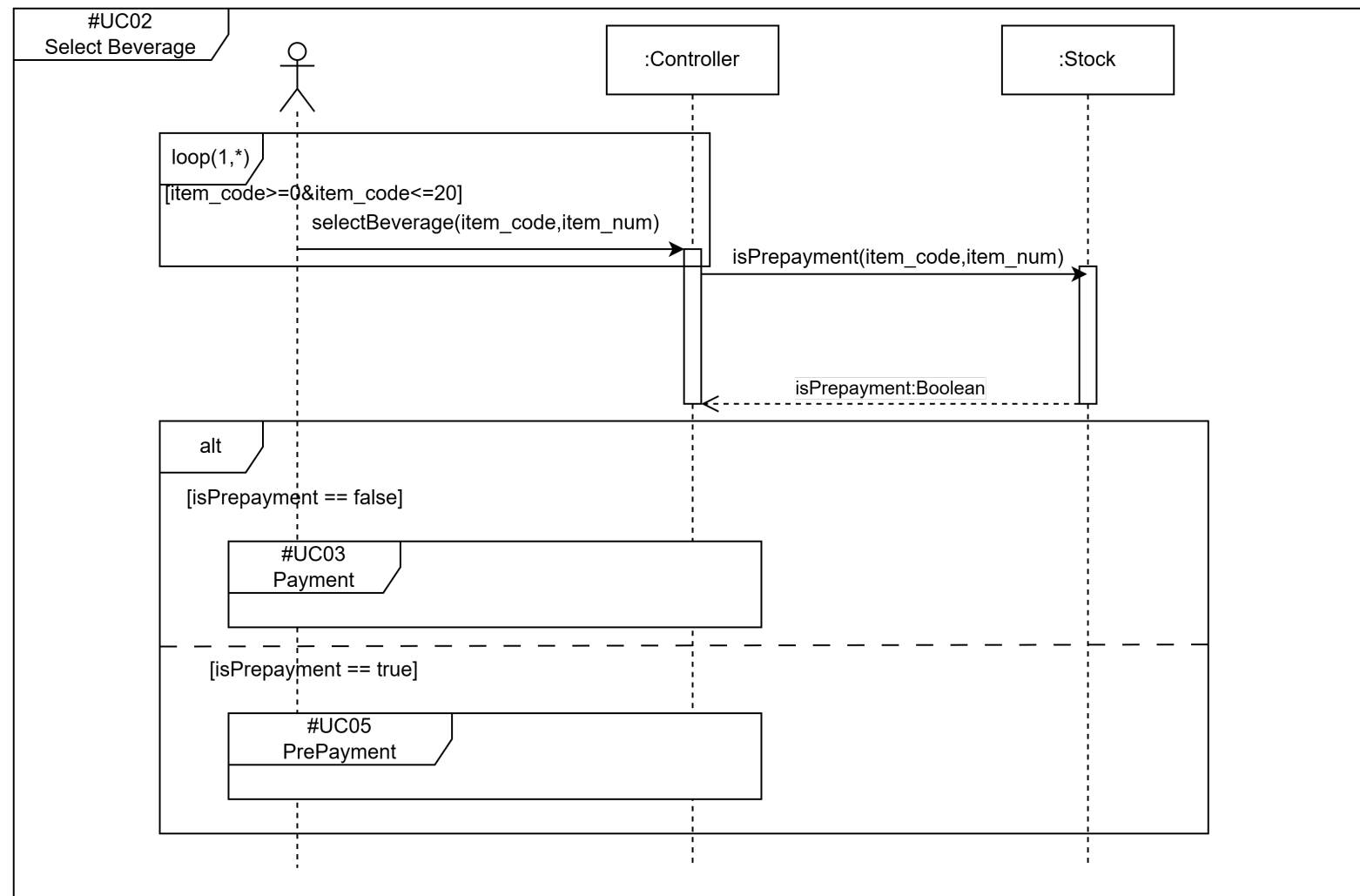
2. 음료 선택

1. (A) 구매하고자 하는 음료의 코드와 갯수를 입력한다

2. (S) 입력 받은 값과 Stock에 저장되어 있는 재고를 비교한다.

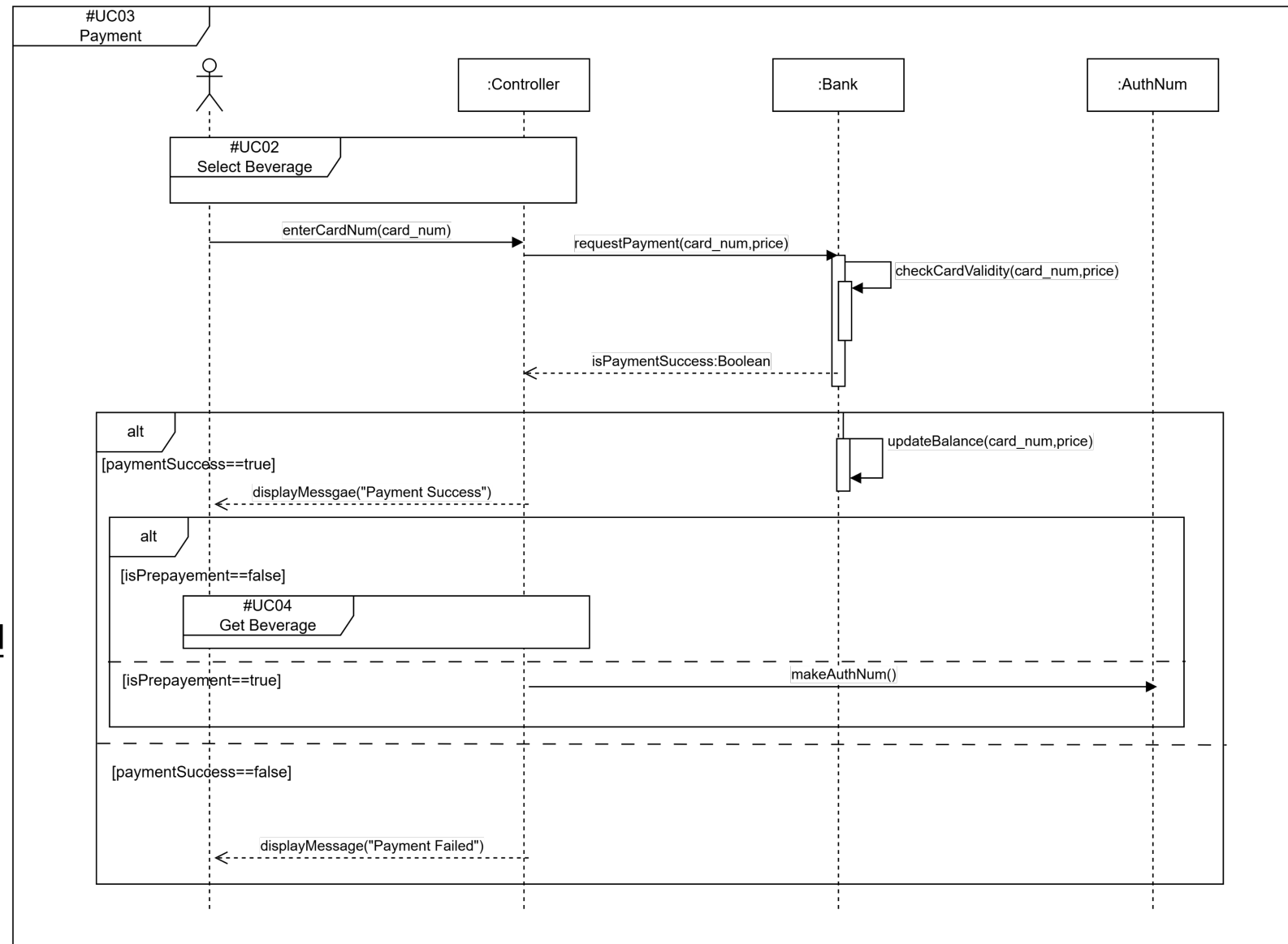
3.(S) 현재 자판기에서 입력받은 값 만큼 음료를 구매할 수 있다면 결제 단계를 진행한다.

4.(S) 현재 자판기에서 구매할 수 없는 수량이라면 선결제 요청 메시지를 다른 자판기들에게 전송한다.



3. 결제

1. (A) 카드 번호를 입력한다.
2. (S) Bank에 카드번호 및 결제 금액을 전달한다
- 3.(B) 카드 번호 및 금액을 확인 후 결제 유효성을 반환한다
4. (B) 결제가 성공할 시 카드의 잔고를 차감한다.
5. (S) 선결제 과정이 아니었다면 음료를 제공한다.
- 6.(S) 선결제 과정 중이었다면 나머지 선결제 과정을 진행한다.

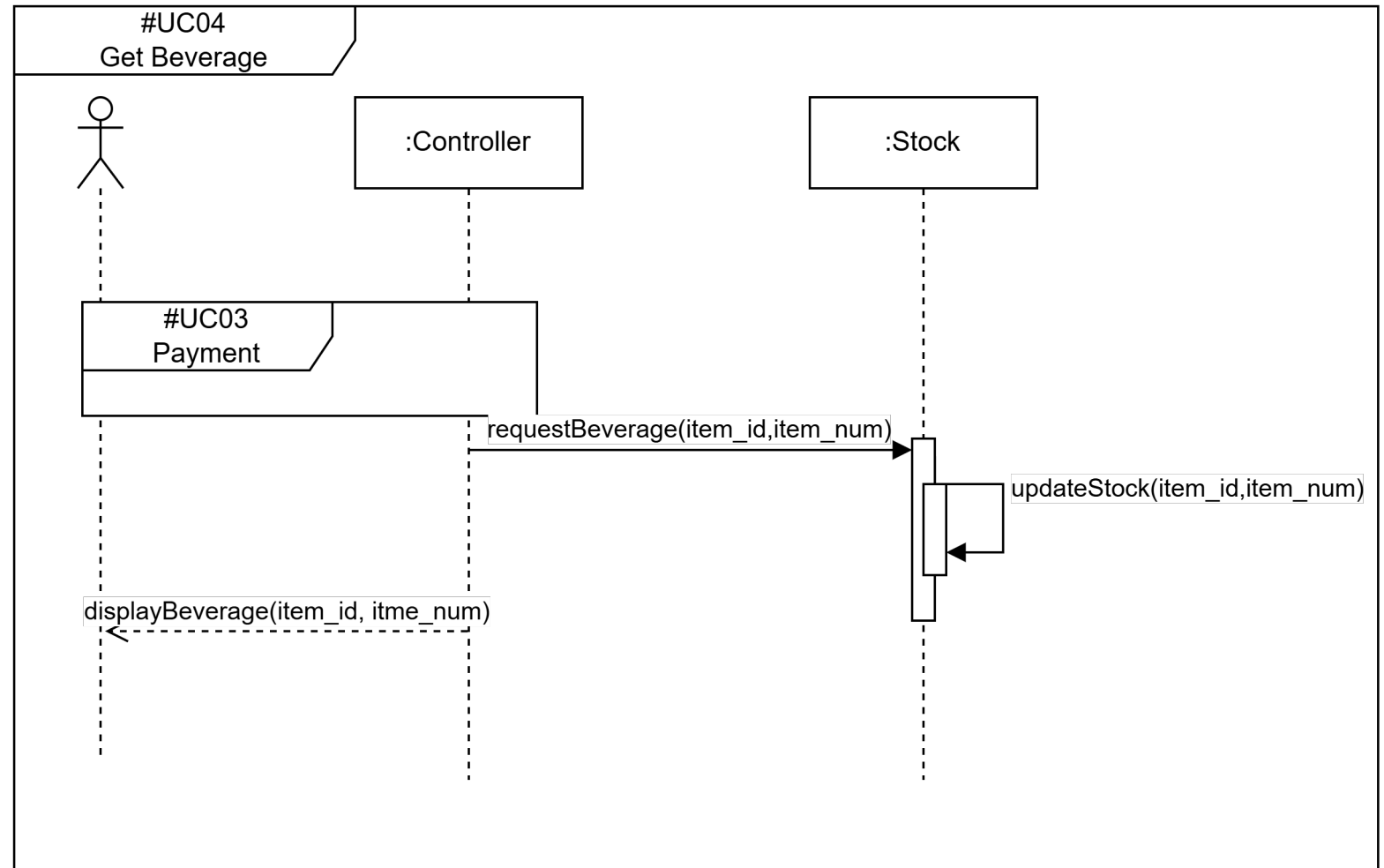


4. 음료 제공

1. (A) 유저가 정상적으로 결제과정을 완료했다.

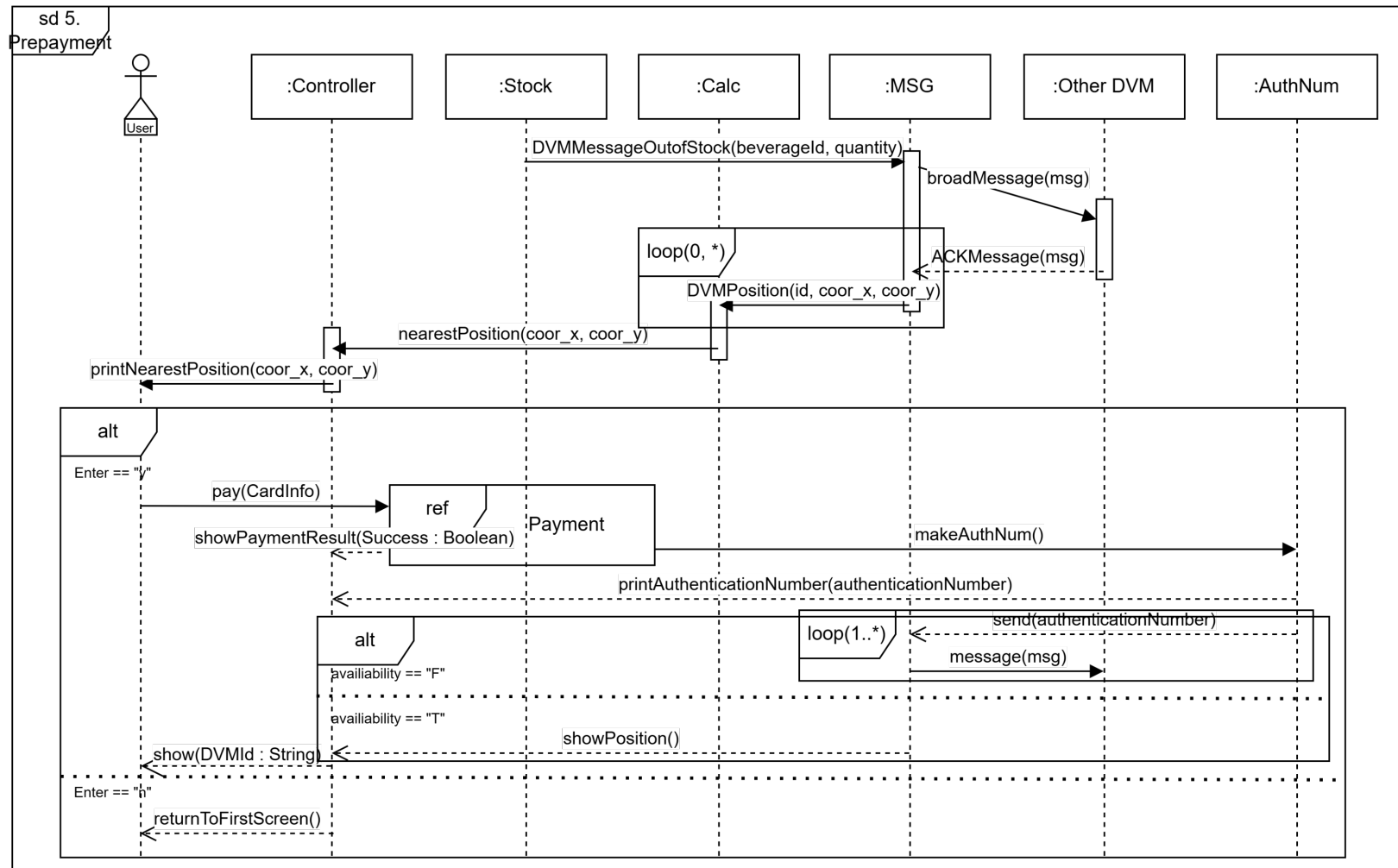
2.(S) 결제한 만큼의 음료 재고를 차감한다.

3. (S) 결제가 완료된 음료를 Actor에게 제공한다.



5. 음료 선결제

1. (A) 현재 자판기에 존재하지 않는 음료를 선택하거나, 현재 자판기에 존재하는 수량을 초과하는 수치를 입력한다.
2. (S) 다른 자판기에게 입력한 음료의 종류와 수량을 만족하는지 메시지를 브로드캐스트로 전송한다
3. (S) 선결제 구매가능 메시지를 받은 자판기 중 가장 가까운 자판기 위치를 계산한다
4. (A) 사용자가 위치를 확인한 후 입력 창에 Y를 누른다
5. (S) 결제를 진행한다
- 6.(S) 인증코드를 반환한다

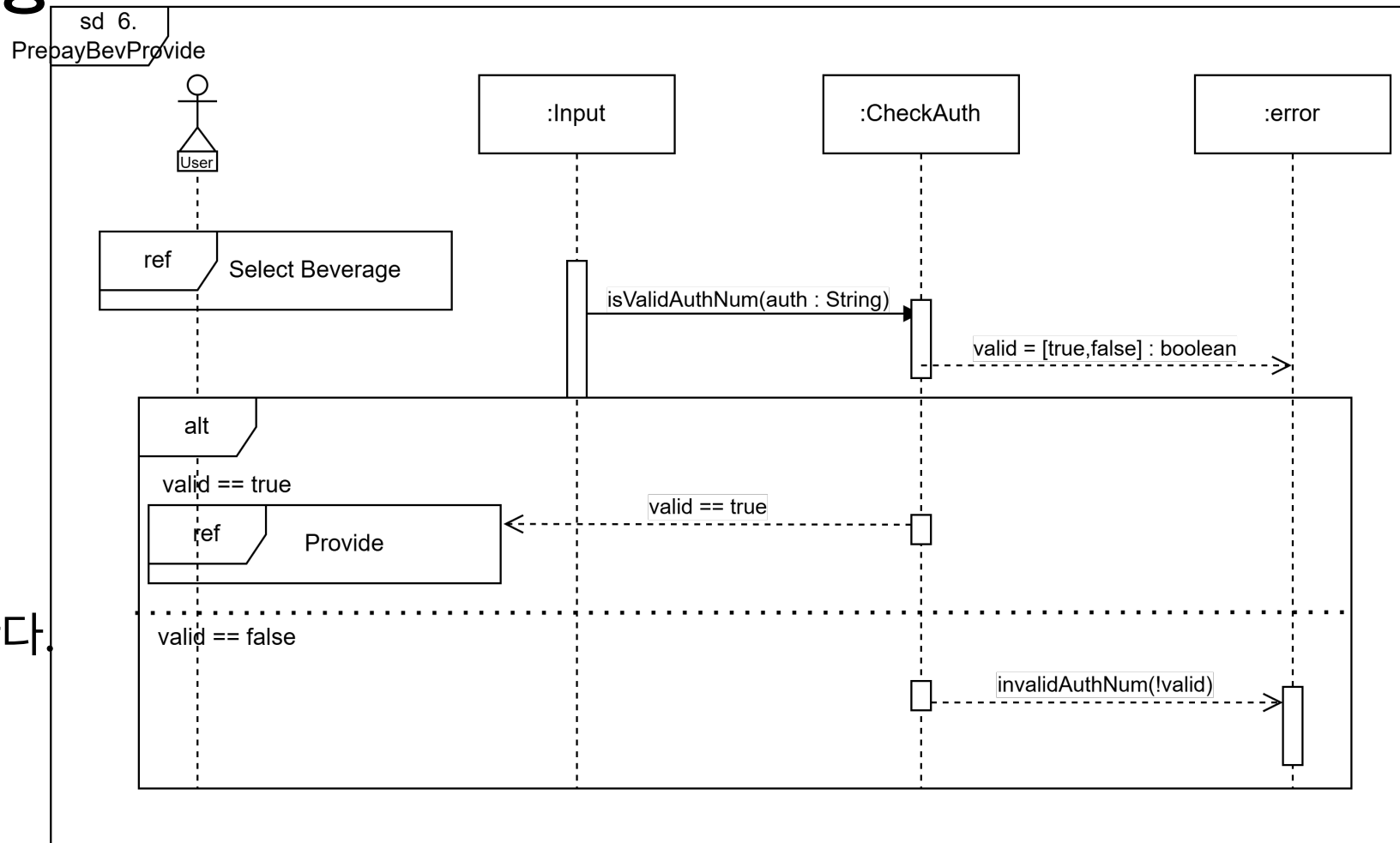


6. 선결제 음료 제공

1.(A): 사용자가 홈 화면에서 인증코드를 입력한다.

2. (S): 해당 인증코드와 매핑되어 있는 음료가 존재하는지 확인한다

3. (S): 인증코드가 유효하면 음료를 제공한다.



7. 재고 확인

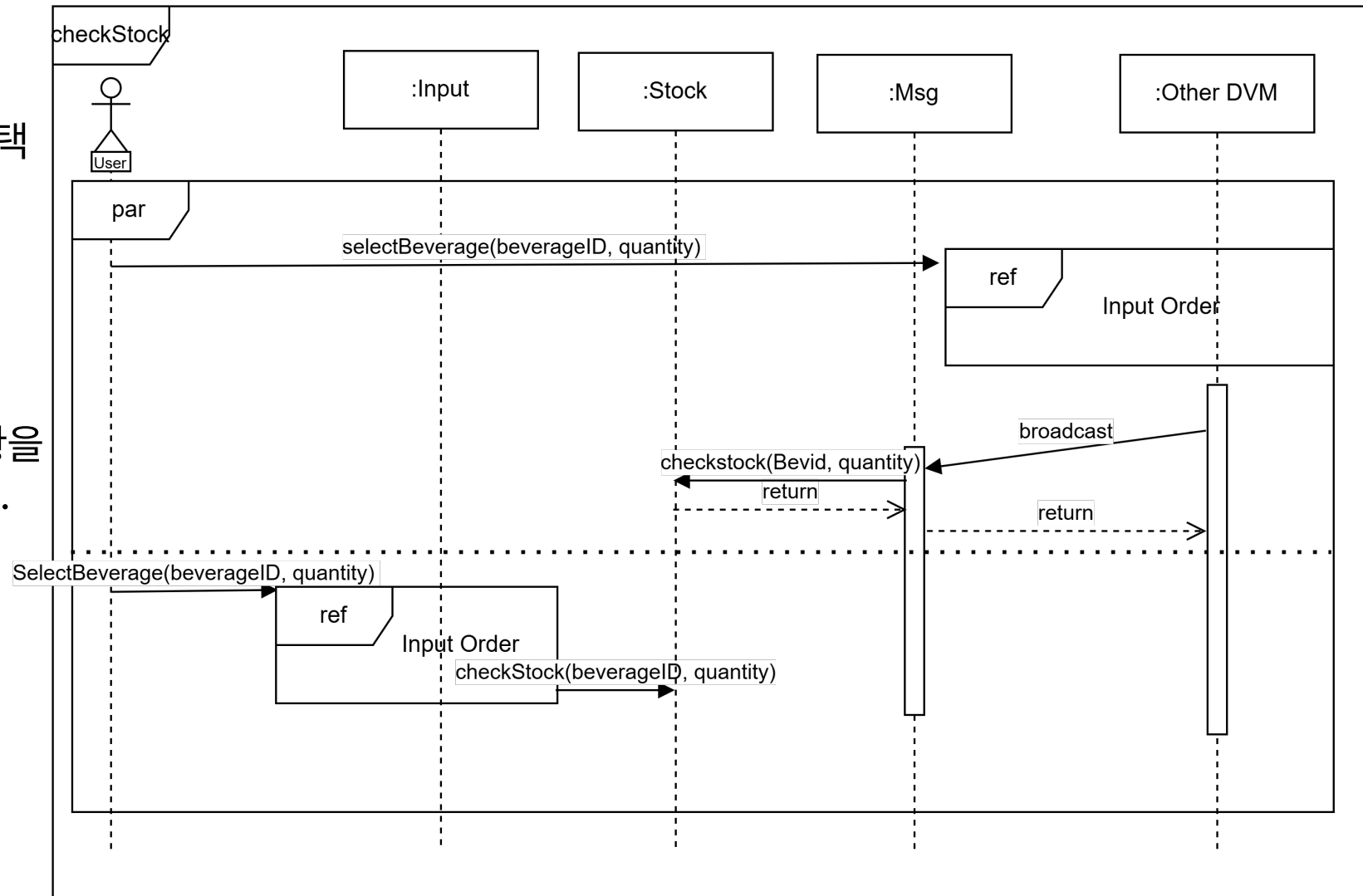
1. 다른 자판기에서 음료 선택 (UC02) 과정이 진행된다.

2. (O) 다른 자판기가 재고 상황 현황을 요구한다.

3.(S) 자판기에서 재고 상황을 파악 후 메시지를 전송한다.

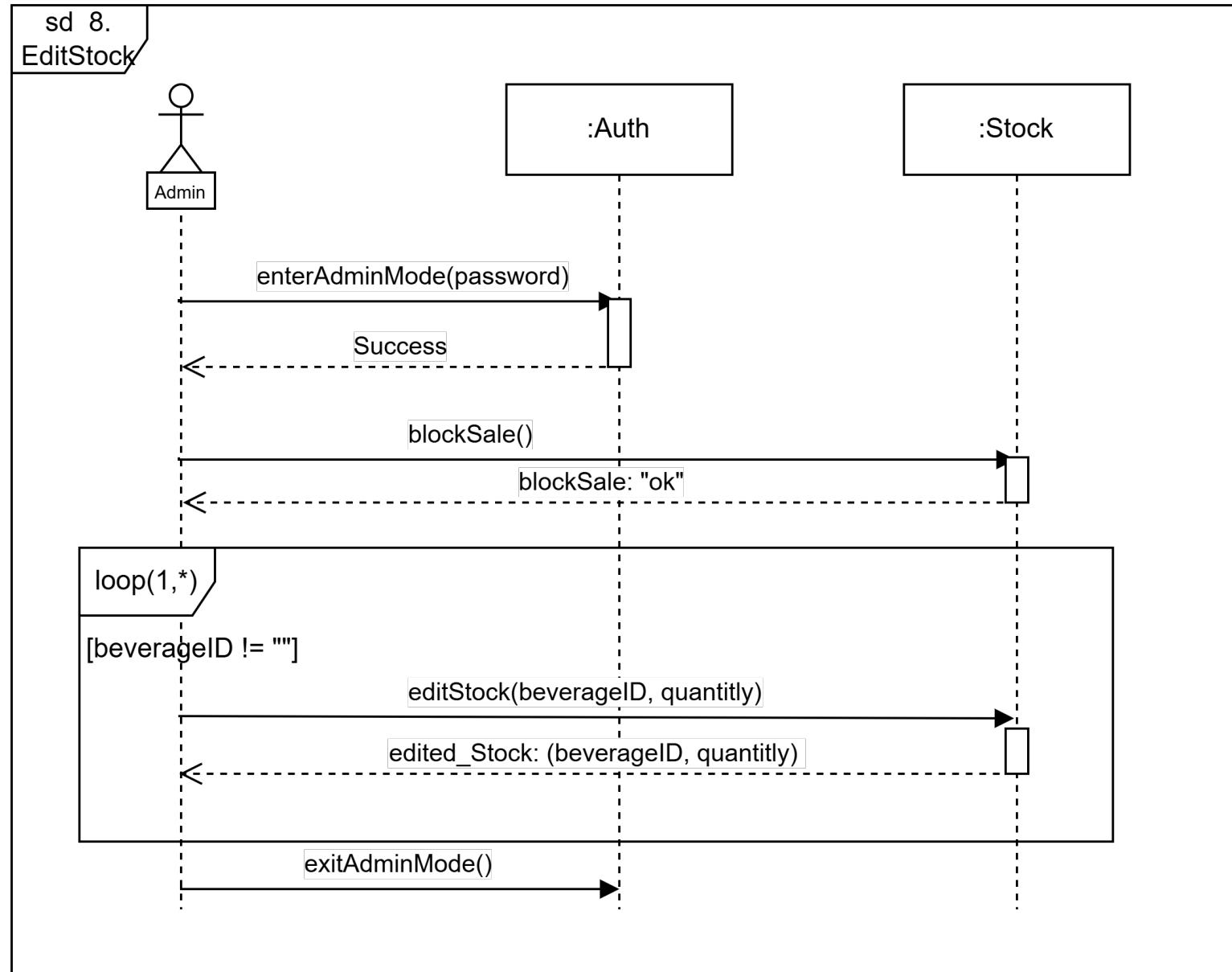
1. (U) 음료 개수와 종류를 요청한다.

2. (D) 자판기에 현재 재고 상황을 파악



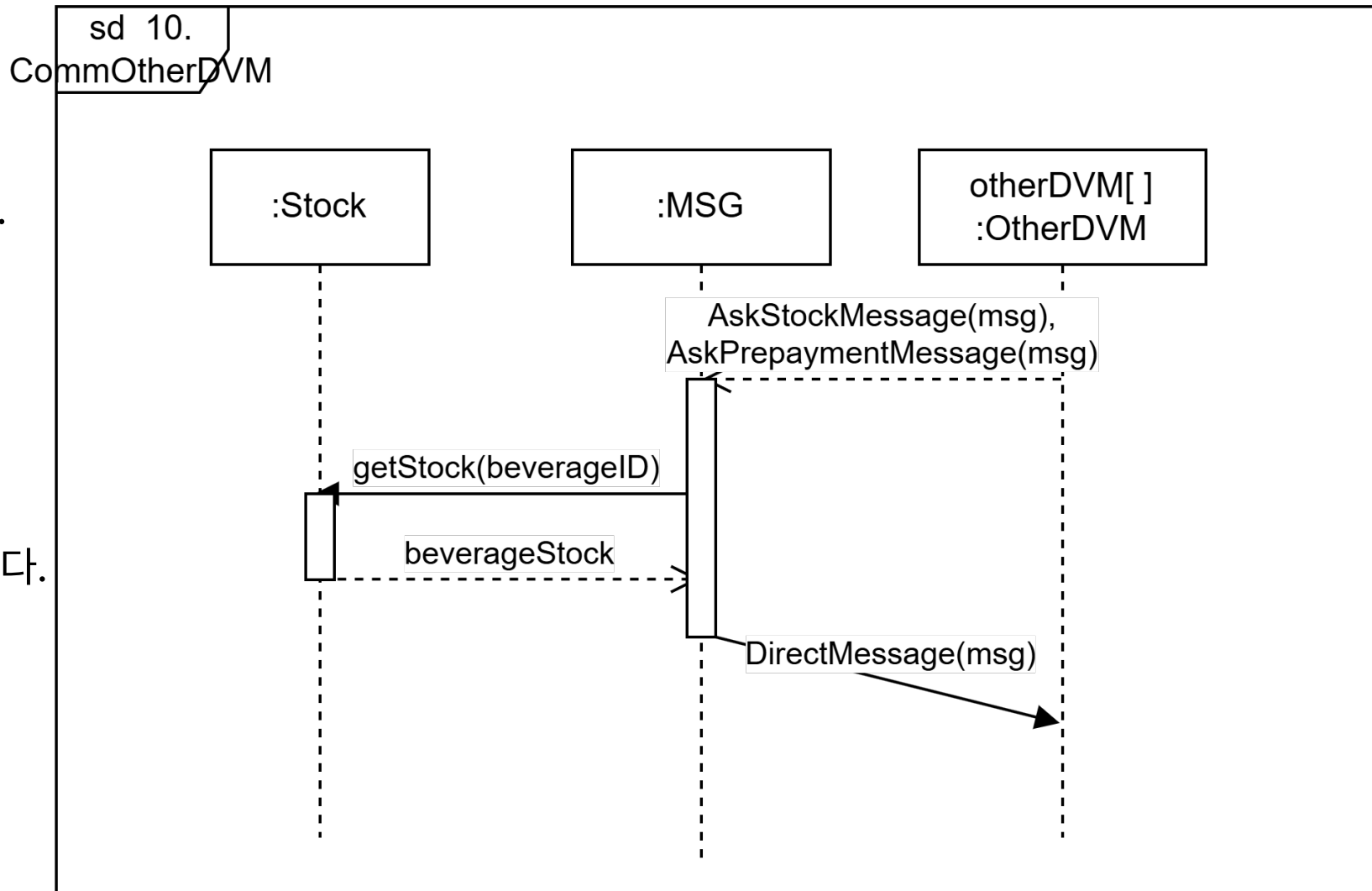
8. 재고 수정

- 1.(A) 관리자 모드로 접속한다
2. (S) 현재 자판기 재고에 대한 구매 이벤트를 차단한다.
3. (A) 관리자가 선택한 음료수를 입력한 개수만큼 재고를 업데이트 한다.
4. (A) 관리자 모드를 해제한다
- 5.(S) 자판기의 구매 이벤트를 활성화한다.



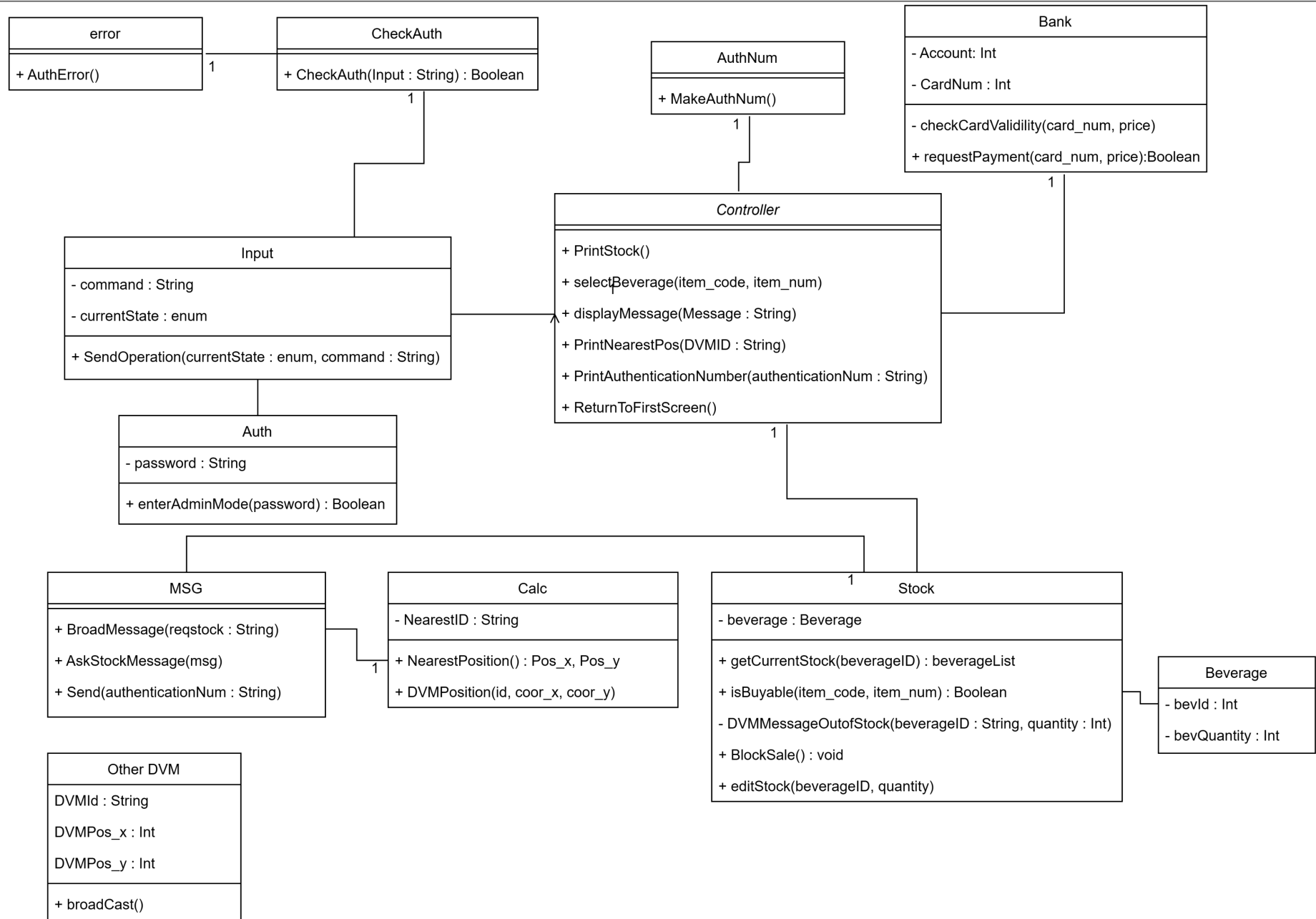
10. 타자판기와의 통신

1. (O) 재고확인을 요청한다.
2. (S) 재고 확인 후 재고와 위치를 담은 메시지를 전송한다.(UC07)
3. (S) 재고가 존재하지 않을 시 broadcast 전송한다.
4. (O) 메시지를 수신하고 응답 메시지를 송신한다.



2045

Define Design Class Diagrams



System Function	Use Case	System Operation	Method	Class
Ref 1.1 메뉴 보기	1. 메뉴 보여주기	pressStart()	authError()	Error
Ref 1.2 음료 선택	2. 음료 선택	showStock(beverageID, quantity)	checkAuth(Input):Boolean	CheckAuth
Ref 2.1 결제요청	3. 결제	SelectBeverage(beverageID, quantity)	makeAuthNum()	AuthNum
Ref 2.2 재고확인	4. 음료제공	currentBeverageAvailable()	checkCardValidity(card_num, price)	Bank
Ref 2.3 카드 유효성 검사	5. 음료 선결제하기	payment(CardInfo)	requestPayment(card_num, price):Boolean	Input
Ref 2.4 카드 잔액 검사	6. 선결제 음료 제공	sendPaymentInfo(CardInfo, Price)	sendOperation(currentState : enum, command : String)	Controller
Ref 2.5 음료 제공	7. 재고 확인	provideBeverage(beverageID, quantity))	printStock()	Controller
Ref 3.1 타 자판기 정보 가져오기	8. 재고 수정	pay(CardInfo)	selectBeverage(item_code, item_num)	Controller
Ref 3.2 재고 + 거리로 자판기 추리기	9. 인증번호 생성	printAuthenticationNumber(authenticationNumber)	displayMessage(Message : String)	Controller
Ref 3.3 자판기 위치 띄우기	10. 타 자판기와의 통신	inputAuthenticationNumber(authenticationNumber)	printNearestPos(DVMID : String)	Controller
Ref 3.4 선결제 여부		Message(position, beverageID, quantity)	printAuthenticationNumber(authenticationNum : String)	Controller
Ref 4.1 선결제 프로세스		enterAdminMode(password)	returnToFirstScreen()	Auth
Ref 5.1 인증 번호 생성		addStock(beverageID, quantity)	enterAdminMode(password) : Boolean	MSG
Ref 5.2 인증 번호 넘기기		exitAdminMode()	broadMessage(reqstock : String)	MSG
Ref 5.3 인증 번호 출력		makeAuthenticationNumberWithStore()	askStockMessage(msg)	MSG
Ref 5.4 자판기 위치 정보 출력		Message(VendingID, beverageID, Quantity)	send(authenticationNum : String)	MSG
Ref 6.1 선결제 타 자판기 요청			nearestPosition() : Pos_x, Pos_y	Calc
			dVMPosition(id, coor_x, coor_y)	Calc
			getCurrentStock(beverageID) : beverageList	Stock
			isBuyable(item_code, item_num) : Boolean	Stock
			dVMMessageOutOfStock(beverageID : String, quantity : Int)	Stock
			blockSale() : void	Stock
			editStock(beverageID, quantity)	Stock