OOPT Stage 2040

객체지향개발방법론

Team[6]

201711374 권규형
201814119 문지영
201911167 김현정
202011370 조석래

CONTENTS

2140 Design

2141 Design Real Use Case

2142 Define Reports, UI, and Storyboards

2143 Refine System Architecture

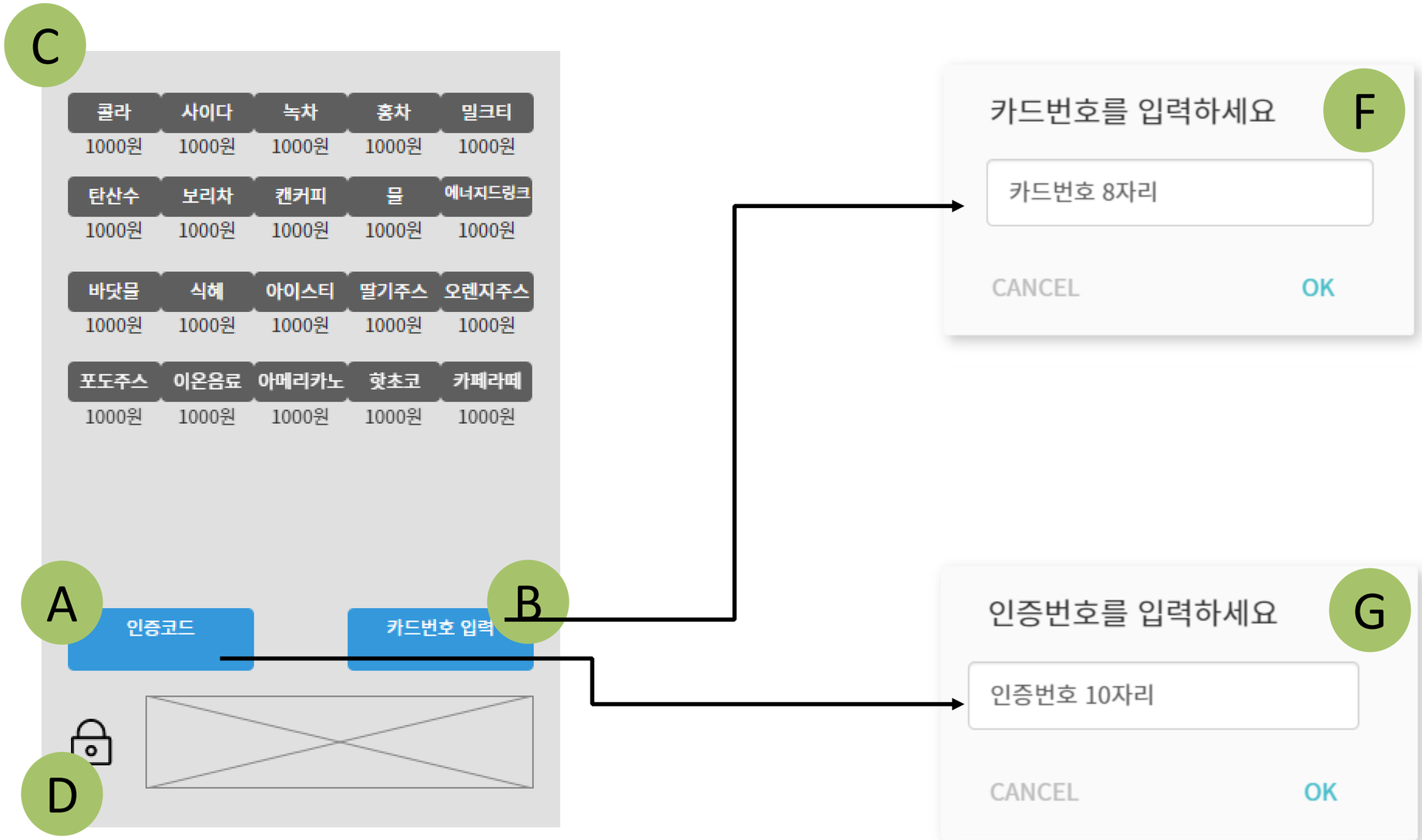
2144 Define Interaction Diagrams

2145 Define Design Class Diagrams

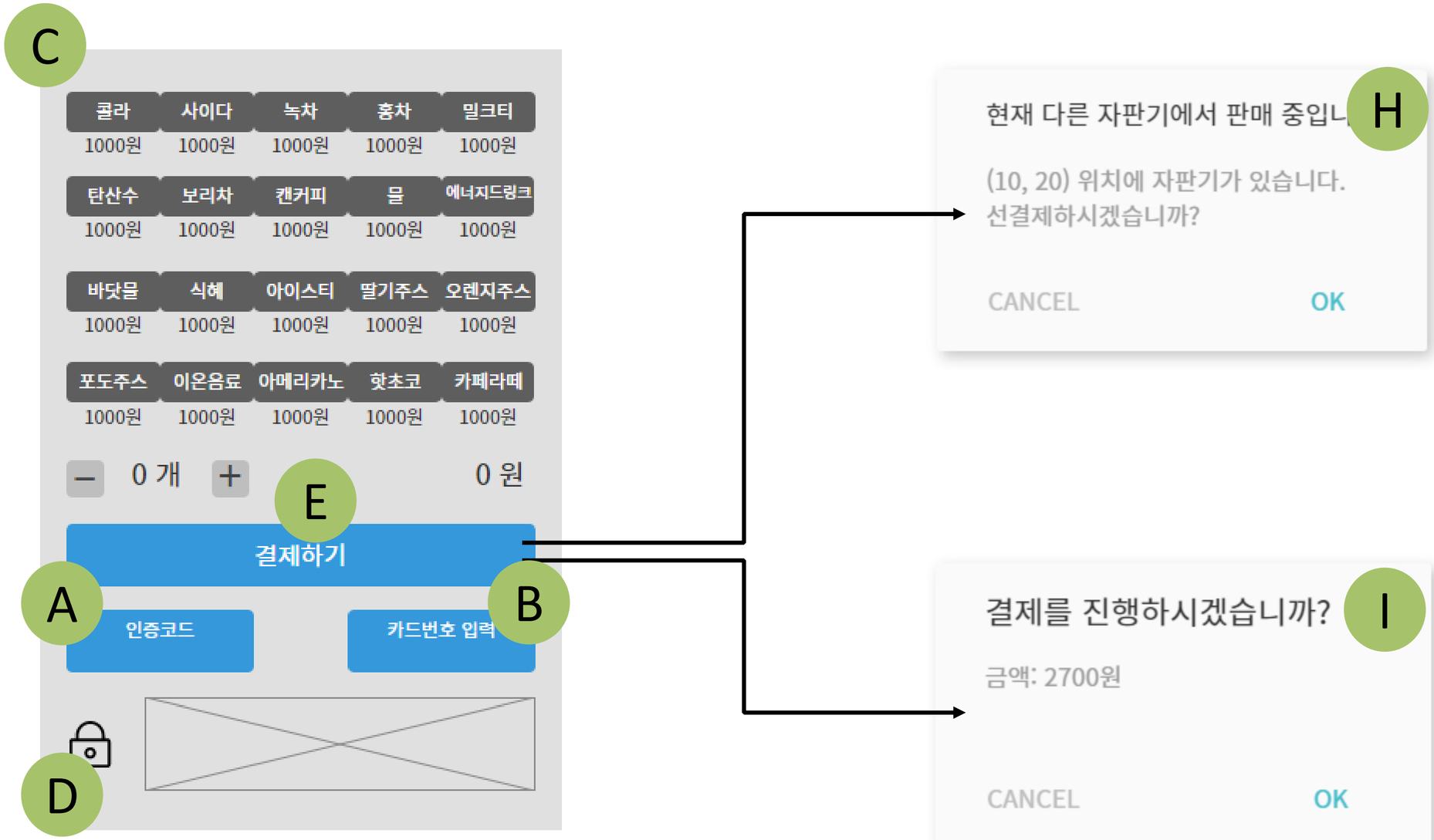
2146 Design Traceability Analysis

2147 Define Database Schema

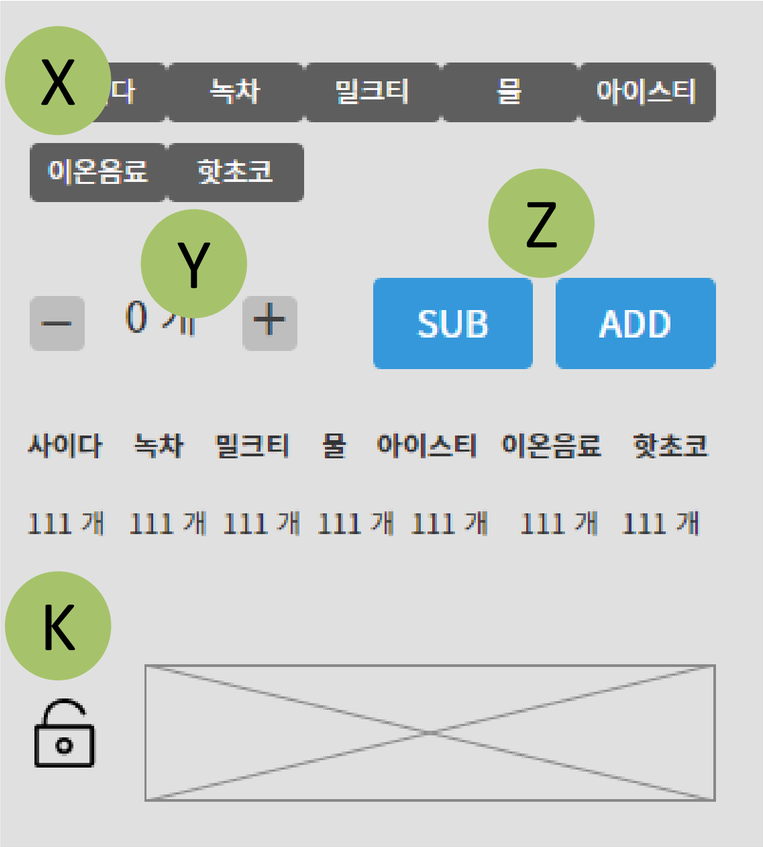
Define Reports, UI, and Storyboards



Define Reports, UI, and Storyboards

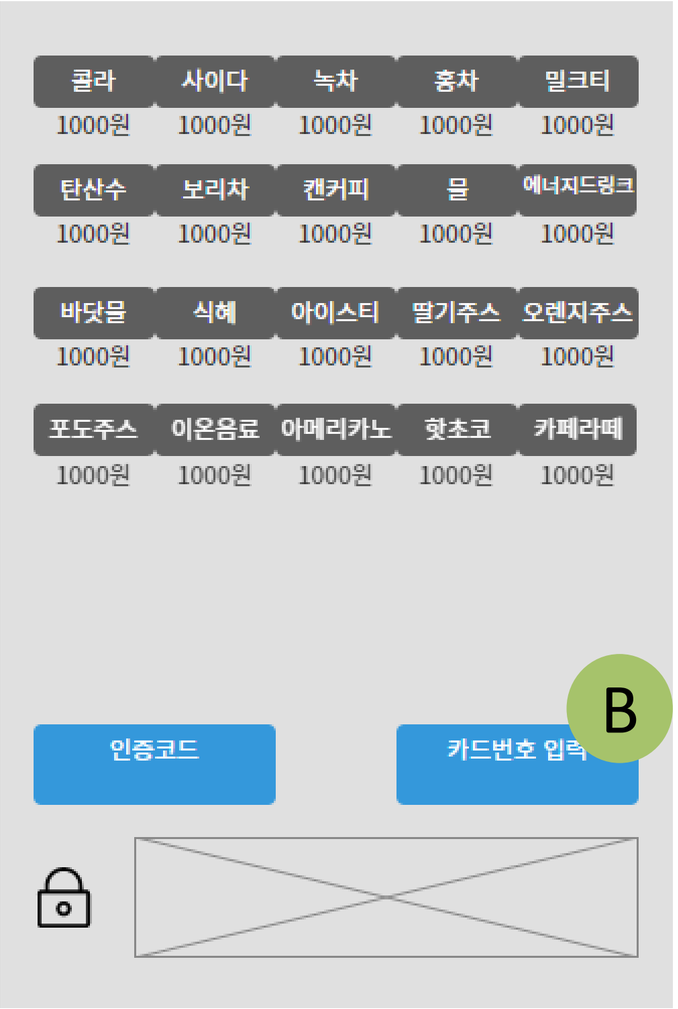


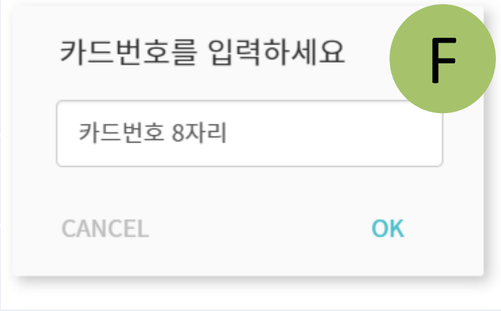
Define Reports, UI, and Storyboards



Design Real Use Cases

Use Case	1. 카드번호 입력
Actor	고객
Purpose	자판기를 이용하기 위해 카드번호를 입력한다.
Overview	사용자가 카드를 자판기 안에 넣으며 자판기는 이 카드가 유효한 지 체크한다. 유효하다면 기본 상태에서 자판기를 활성화한다.
Type	Evident
Cross Reference	Functions : R1.1
	Use Cases :
Pre-Requisites	N/A
Typical	(A) : Actor, (S) : System 1. (A) 고객이 자판기에 8자리 숫자의 카드번호를 입력한다. 2. (S) 카드의 정보를 읽어 카드사를 통해 카드가 유효한 지 체크한다. (E1) 3. (S) 자판기를 활성화 한다.
Alter	
Except	E1: 고객이 넣은 카드가 유효하지 않은 카드이면 유효하지 않음을 알리고 기본 상태를 유지한다.



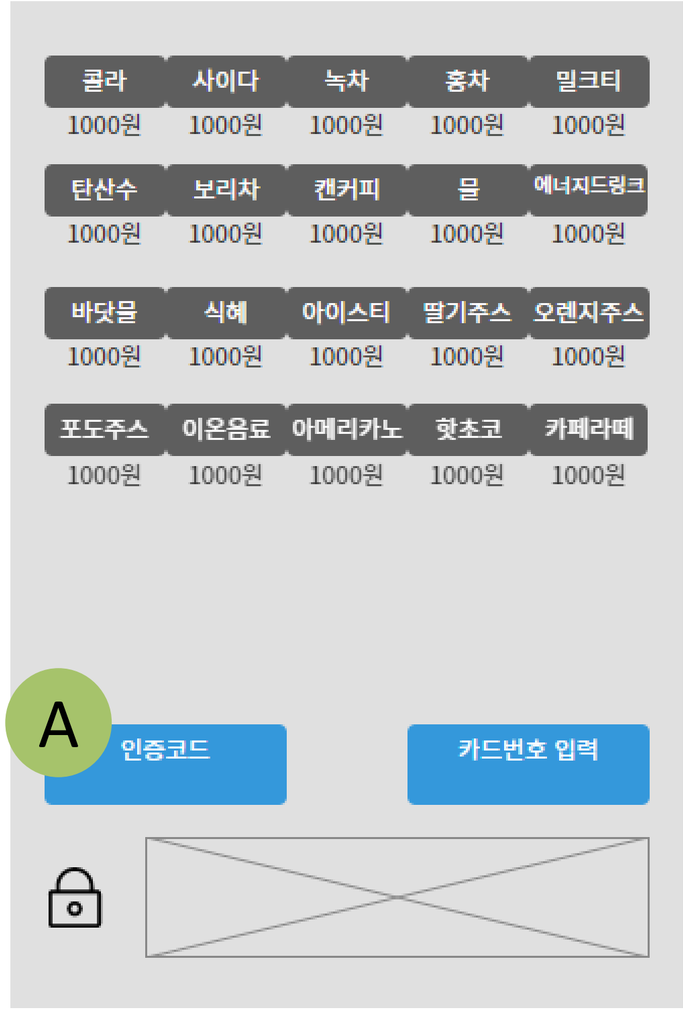


2141

Design Real Use Cases

Use Case	2. 인증코드 입력하기
Actor	고객
Purpose	외부 자판기에서 선결제한 음료를 찾기 위해 인증코드를 입력한다.
Overview	사용자는 인증코드를 입력하고 자판기는 인증코드가 유효한지 검사한다.
Type	Evident
Cross Reference	Functions: R1.2, R5.3
	Use Case : 9.선결제 정보 전송하기
Pre-Requisites	N/A
Typical	(A): Actor, (S): System 1. (A) 고객이 발급받은 10자리의 인증코드를 입력한다. 2. (S) 시스템은 입력 받은 인증코드가 유효한지 검사한다.(E1) (E2) 3. (S) 선결제한 음료가 제공된다.
Alter	
Except	E1: 존재하지 않은 인증코드인 경우 존재하지 않음을 알린다. E2: 잘못된 선결제 인증코드인 경우 잘못된 선결제 인증코드임을 알린다.

A G

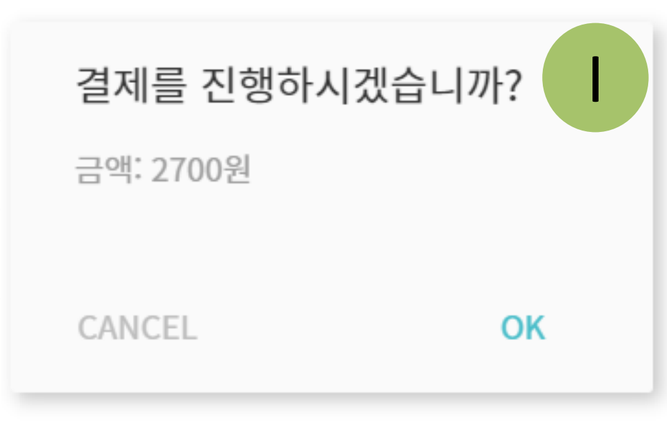


Design Real Use Cases

Use Case	3. 음료 선택하기
Actor	고객
Purpose	구매할 음료를 선택한다.
Overview	고객이 구매할 음료와 개수를 입력한다.
Type	Evident
Cross Reference	Functions: R1.1, R2.1
	Use Case: 1. 카드번호 입력하기
Pre-Requisites	유효한 카드번호를 입력한다.
Typical	(A): Actor, (S): System 1. (A) 고객이 구매할 음료와 수량을 입력한다. (E1) 2. (S) 현 자판기의 재고와 고객이 구매할 음료의 수량을 비교한다. (E2) 3. (A) 고객이 결제하기 버튼을 누른다
Alter	
Except	(E1): 잘못된 음료를 선택한 경우 잘못 되었음을 알리고 기본 상태로 돌아간다. (E2): 재고가 충분하지 않다면 대상 자판기를 탐색한다.

Design Real Use Cases

Use Case	4. 결제하기
Actor	고객
Purpose	음료를 받기 위해 값을 지불한다
Overview	선택한 음료의 수량이 자판기의 재고를 넘지 않을 때 사용자의 카드 정보를 읽어 결제를 요청한다. 결제 가능 시 결제한다.
Type	Evident
Cross Reference	Functions :R1.1, R2.1, R3.1
	Use Case: 1. 카드 넣기, 3: 음료 선택하기
Pre-Requisites	선택한 음료의 개수가 현 자판기의 재고를 넘지 않는다.
Typical	(A) : Actor, (S) : System 1. (A) 고객이 결제를 요청한다. 2. (S) 선택한 음료의 재고가 충분한지 한 번 더 확인한다 (E1) 3. (S) 카드 정보를 읽고 카드사에 결제를 요청한다. (E2) 4. (S) 현 자판기의 상품 정보를 업데이트 한다. 5. (S) 상품을 배출한다. 6. (S) 기본 상태로 돌아간다.
Alter	
Except	(E1) : 재고가 부족한 경우 결제 실패 메세지를 띄우고 기본 상태로 돌아간다. (E2) : 카드의 잔액이 없을 경우 결제 실패했음을 띄우고 기본 상태로 돌아간다.

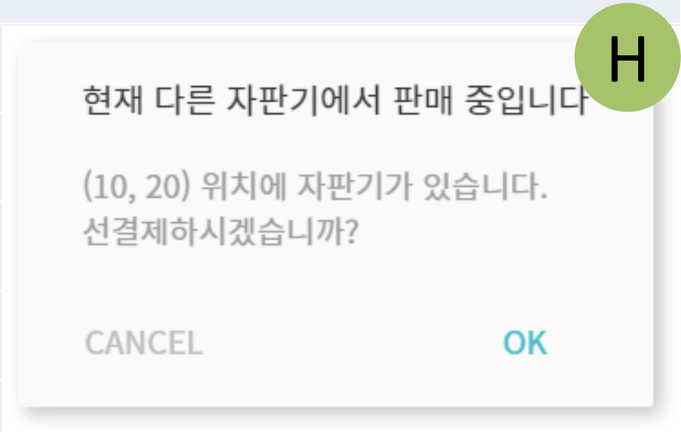


Design Real Use Cases

Use Case	5. 대상 자판기 확인하기
Actor	-
Purpose	대상자판기 위치 확인
Overview	선택한 음료의 재고가 충분하지 않거나 품목이 현 자판기에 없을 때, 대상 자판기의 위치를 보여주고, 선 결제 여부를 묻는다
Type	Hidden
Cross Reference	Functions: R1.1, R2.1, R4.1
	Use Case: 1. 카드번호 입력하기, 3. 음료 선택하기
Pre-Requisites	선택한 음료의 개수가 현 자판기의 재고를 넘는다
Typical	(A): Actor, (S): System 1. (S) 외부자판기로 음료 판매 확인 프로토콜을 전송한다. 2. (S) 외부자판기 중 해당 음료를 판매하는 자판기에서 음료 판매 응답 프로토콜을 응답을 받는다 (E1) 3. (S) 대상 자판기를 선정한다 4. (S) 고객에게 대상 자판기 정보와, 그 위치를 보여준다. 5. (S) 고객에게 선결제 여부를 묻는다.
Alter	
Except	E1 : 모든 자판기에서 프로토콜을 보내지 않으면 사용자에게 재고 부족을 알리고 기본 상태로 돌아간다.

Design Real Use Cases

Use Case	6. 선결제 하기
Actor	고객
Purpose	선결제
Overview	고객이 선결제를 요청했을 때, 사용자의 카드 정보를 읽어 결제를 요청한다. 결제 가능 시 결제한다.
Type	Evident
Cross Reference	Functions: R1.1, R2.1, R4.1, R4.2
	Use Case: 1. 카드번호 입력하기, 3. 음료 선택하기, 5. 대상 자판기 확인하기
Pre-Requisites	대상 자판기가 선정 되어있다
Typical	(A): Actor, (S): System 1. (A) 고객이 선결제를 요청한다. 2. (S) 선결제를 진행할 외부 자판기에 재고 확인을 한다. (E1) (E2) 3. (S) 카드사에 결제를 요청한다. (E3) 4. (S) 선결제 인증코드를 생성한다. 5. (S) 선결제 대상 외부 자판기에 구매한 음료와 개수, 인증코드를 전송한다. 6. (S) 선결제 인증코드를 보여준 후 기본 상태로 돌아간다.
Alter	
Except	E1: 외부자판기에서 재고 응답이 오지 않으면 결제를 진행할 수 없다. E2: 외부자판기에서 응답이 왔지만 재고가 부족하면 결제를 진행할 수 없다. E3: 카드의 잔액이 부족한 경우 결제가 실패했음을 알리고 기본 상태로 돌아간다.



Design Real Use Cases

Use Case	7. 음료 판매 정보 확인하기
Actor	외부 자판기
Purpose	음료 판매 여부 파악
Overview	외부 자판기에서 현 자판기의 음료 판매 정보를 확인한다.
Type	Evident
Cross Reference	Functions: R5.1
	Use Cases:
Pre-Requisites	
Typical	(A): Actor, (S): System 1. (A) 외부 자판기에서 음료 판매 확인을 요청한다. 2. (S) 음료 품목이 개수만큼 존재하는지 확인한다. (E1) 3. (S) 프로토콜에 맞게 응답한다
Alter	
Except	(E1): 요청 받은 개수만큼 존재하지 않으면 응답하지 않는다.

Design Real Use Cases

Use Case	8. 재고 정보 확인하기
Actor	외부 자판기
Purpose	재고 파악
Overview	외부 자판기에서 현 자판기의 음료 재고 정보를 확인한다.
Type	Evident
Cross Reference	Functions: R5.2
	Use Cases:
Pre-Requisites	N/A
Typical	(A): Actor, (S): System 1. (A) 외부 자판기에서 재고 확인 요청을 한다. 2. (S) 현 자판기의 재고를 확인한다 3. (S) 프로토콜에 맞게 응답한다.
Alter	
Except	

Design Real Use Cases

Use Case	9. 선결제 정보 전송하기
Actor	외부 자판기
Purpose	선결제 정보 전송
Overview	선결제가 진행된 외부 자판기가 어떤 음료가 몇 개 선결제 되었는 지 인증코드와 함께 전송한다.
Type	Evident
Cross Reference	Functions: R5.3
	Use Cases:
Pre-Requisites	
Typical	(A): Actor, (S): System 1. (A) 선결제가 진행된 외부 자판기에서 음료 코드, 개수, 인증 코드를 전송한다. 2. (S) 선결제 정보를 저장한다. 3. (S) 현 자판기의 상품 정보를 업데이트한다. (E1)
Alter	
Except	(E1) : 선결제 한 음료의 재고가 현 자판기의 재고보다 많다면, 재고를 빼지 않고, 유효하지 않은 선결제임을 저장한다. 상품 정보 업데이트는 되지 않는다.

2141

Design Real Use Cases

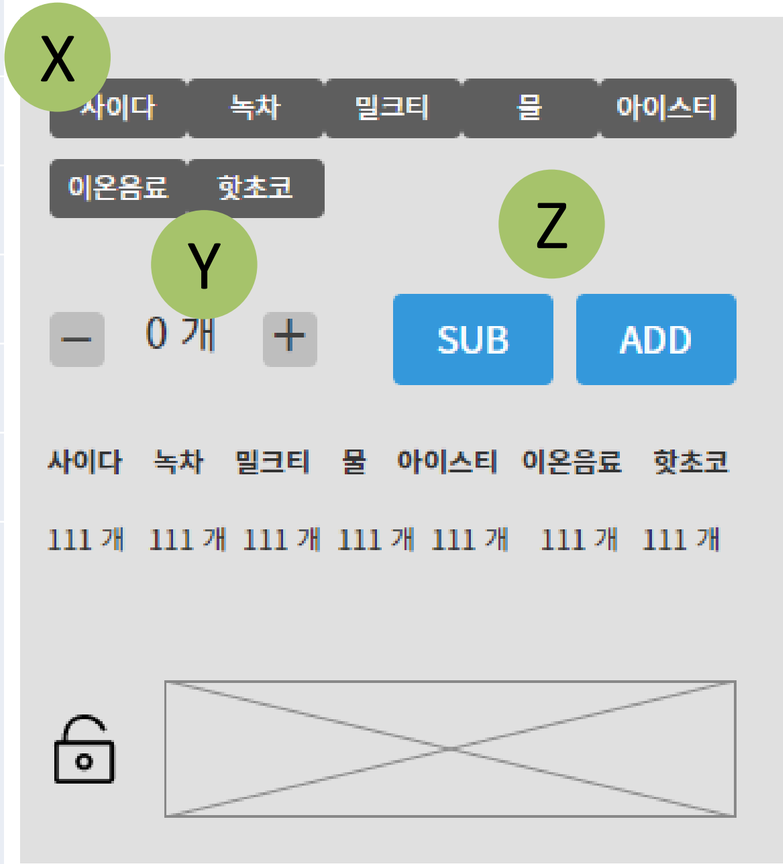
Use Case	10. 관리자 모드 진입하기
Actor	관리자
Purpose	재고 관리
Overview	관리자가 관리자 모드에 진입한다
Type	Evident
Cross Reference	Functions: R6.1, R6.2, R6.3
	Use Cases: 11. 재고 변경하기, 12.관리자 모드 종료하기
Pre-Requisites	N/A
Typical	(A): Actor, (S): System 1. (A) 관리자 모드로 진입한다 (E1) 2. (S) 관리자 화면을 보여준다.
Alter	
Except	E1: 관리자가 아닐 시 진입하지 못한다

D

D

Design Real Use Cases

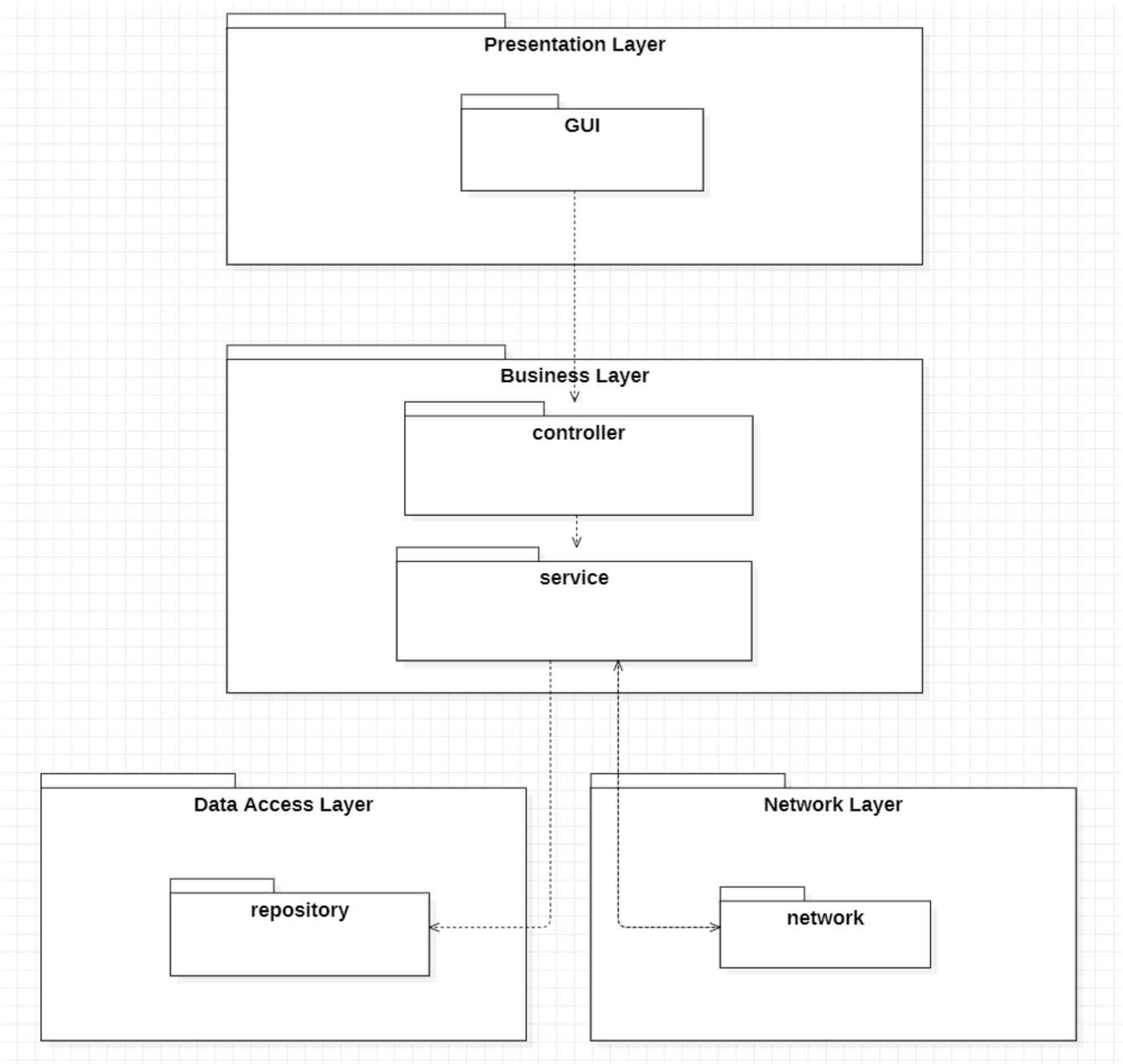
	Use Case	11. 재고 변경하기
2142	Actor	관리자
	Purpose	재고 변경
2143	Overview	관리자 모드 변경을 원하는 음료를 선택하여 재고를 변경한다.
	Type	Evident
2144	Cross Reference	Functions: R6.1, R6.2
		Use Cases: 10. 관리자 모드 진입하기
2145	Pre-Requisites	관지라 모드에 진입한 상태
2146	Typical	(A): Actor, (S): System 1. (A) 관리자가 재고를 변경할 음료들과 개수를 입력한다. (E1) 2. (S) 시스템이 해당 음료의 재고를 변경한다.
2147	Alter	
2140	Except	E1: 관리자가 재고를 0~999가 아닌 숫자로 만들려고 할 때 에러임을 알리고 이 행위를 무시한다.



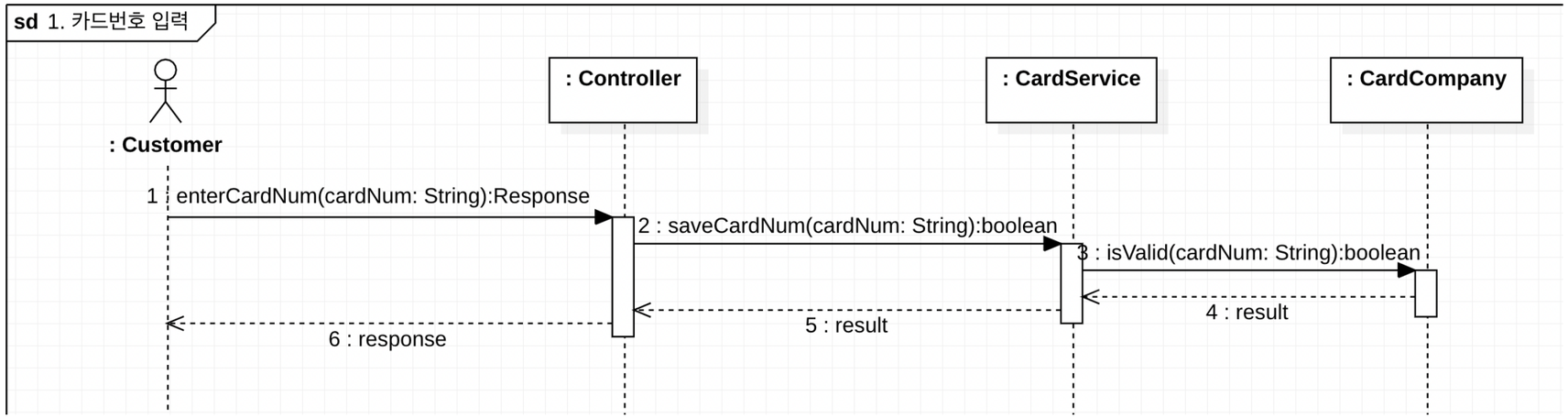
Design Real Use Cases

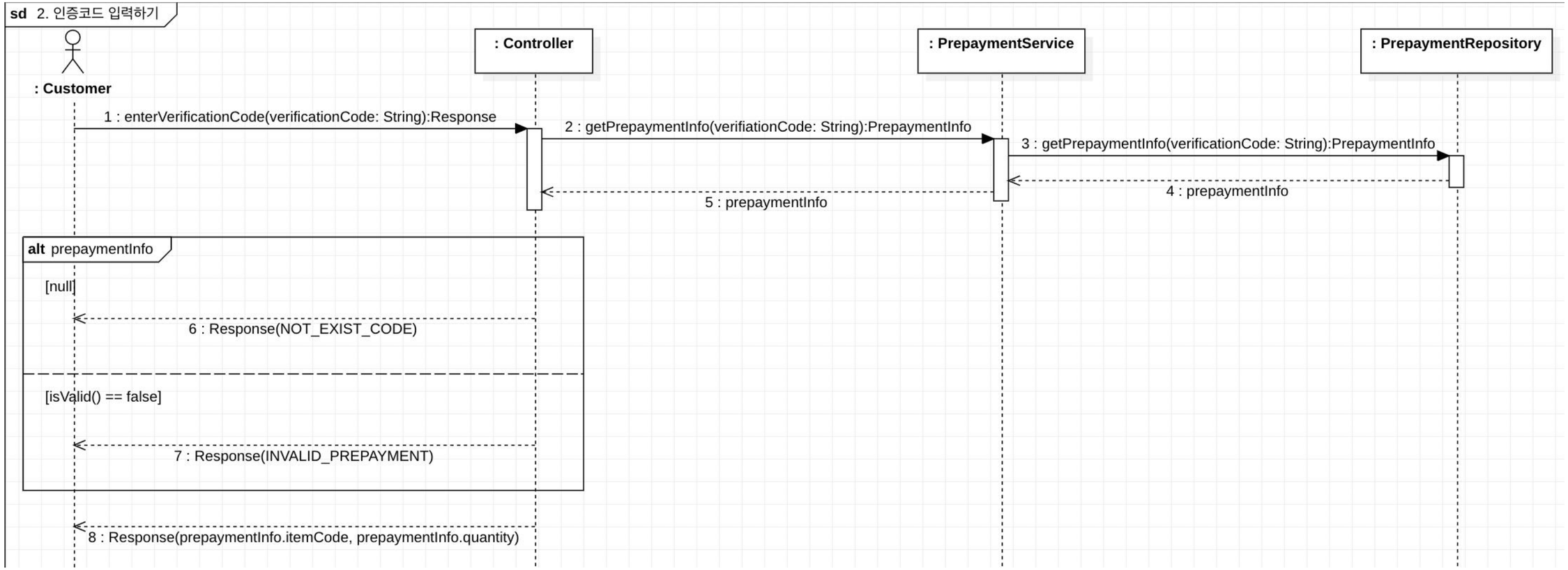
Use Case	12. 관리자 모드 종료하기
Actor	관리자
Purpose	관리자 모드 종료
Overview	관리자 모드를 종료한다.
Type	Evident
Cross Reference	Functions: R6.1, R6.3
	Use Cases: 10. 관리자 모드 진입하기
Pre-Requisites	관리자 모드에 진입한 상태
Typical	(A): Actor, (S): System 1. (A) 관리자 모드를 종료한다. 2. (S) 기본 상태로 돌아간다.
Alter	
Except	

Refine System Architecture

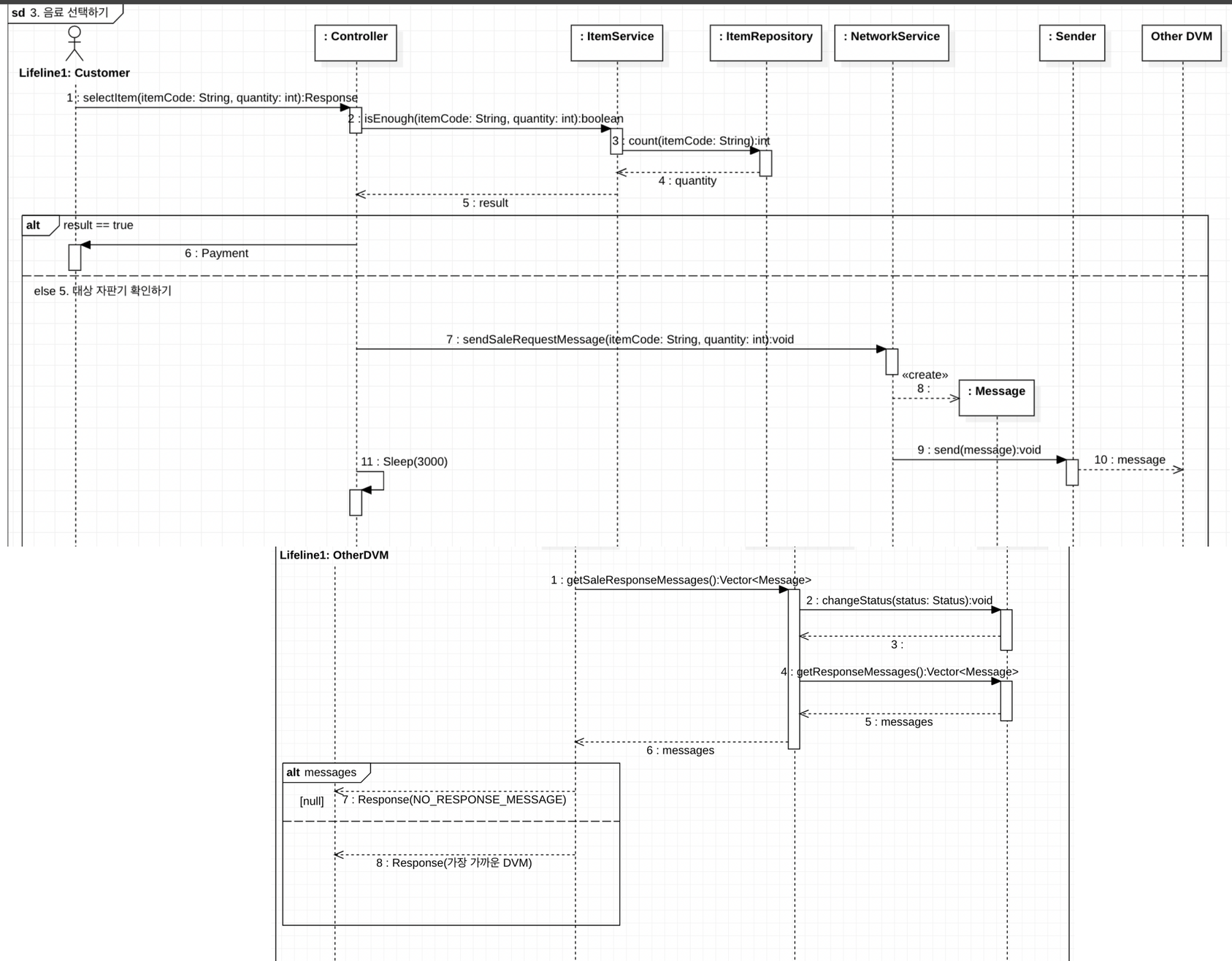


Define Interaction Diagrams

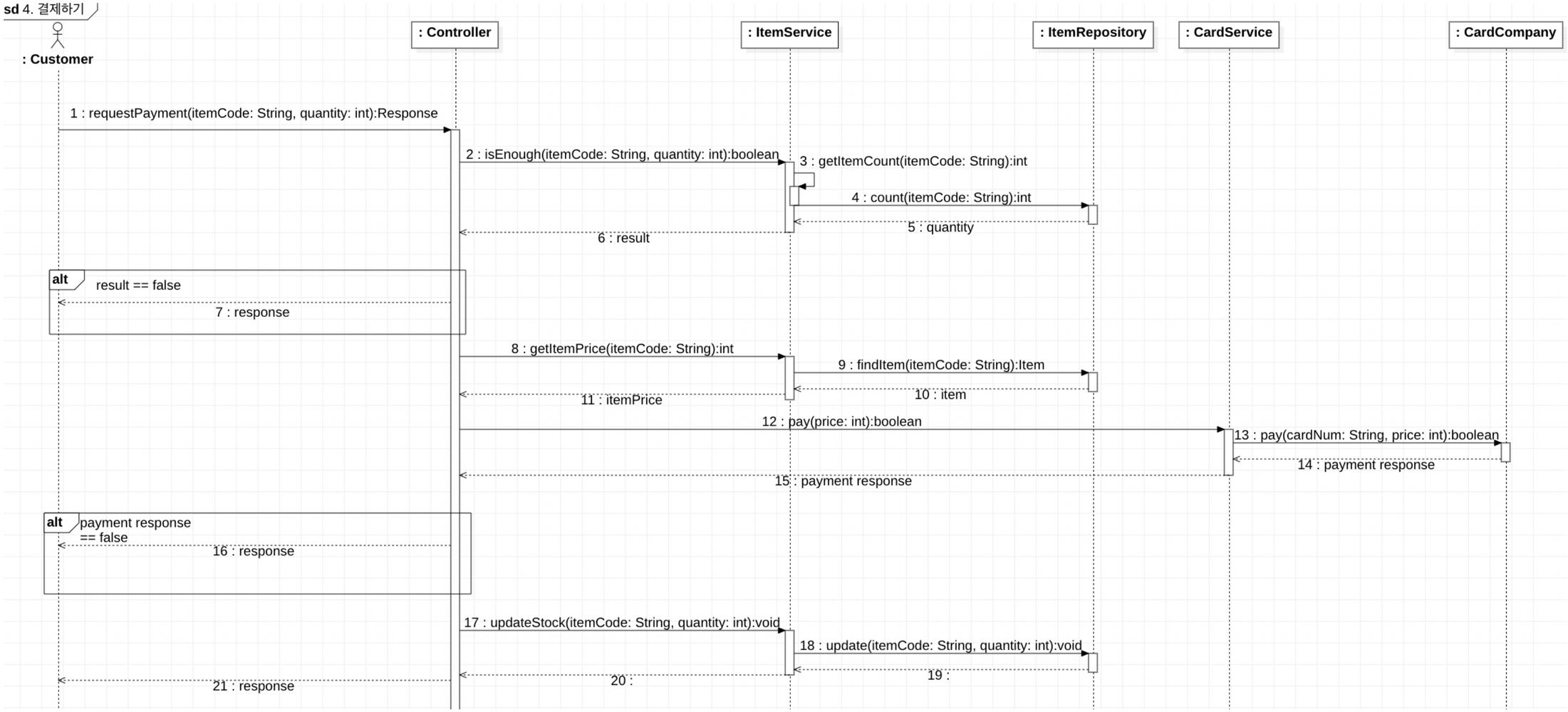




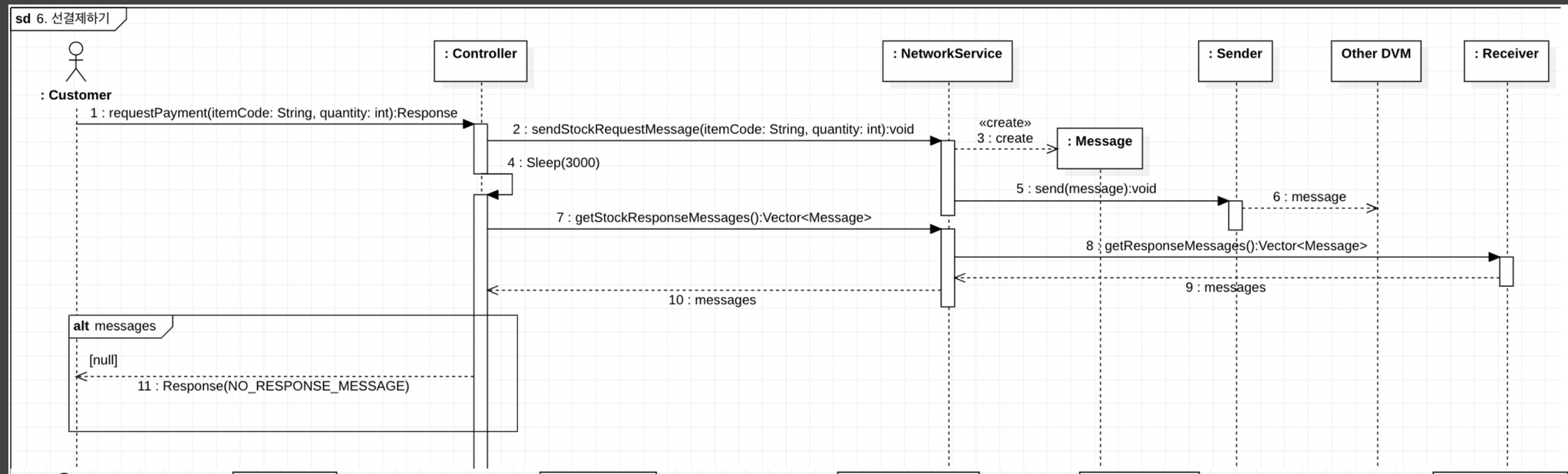
Define Interaction Diagrams

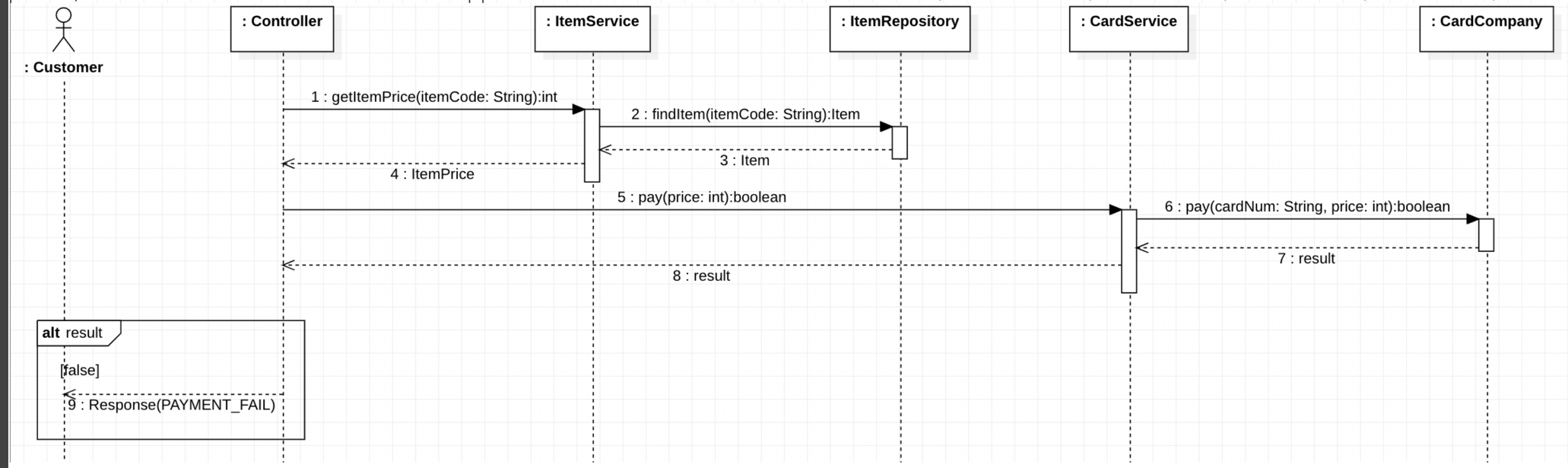


Define Interaction Diagrams

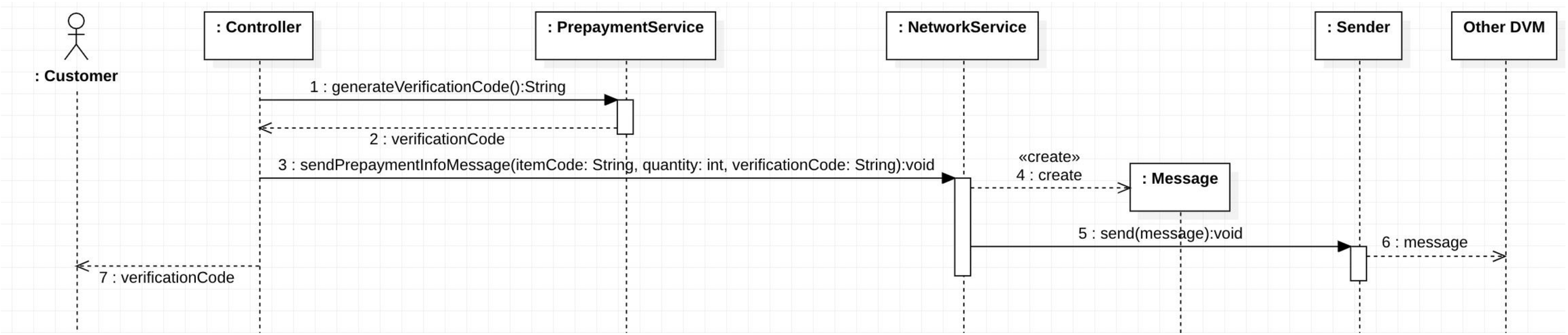


Define Interaction Diagrams

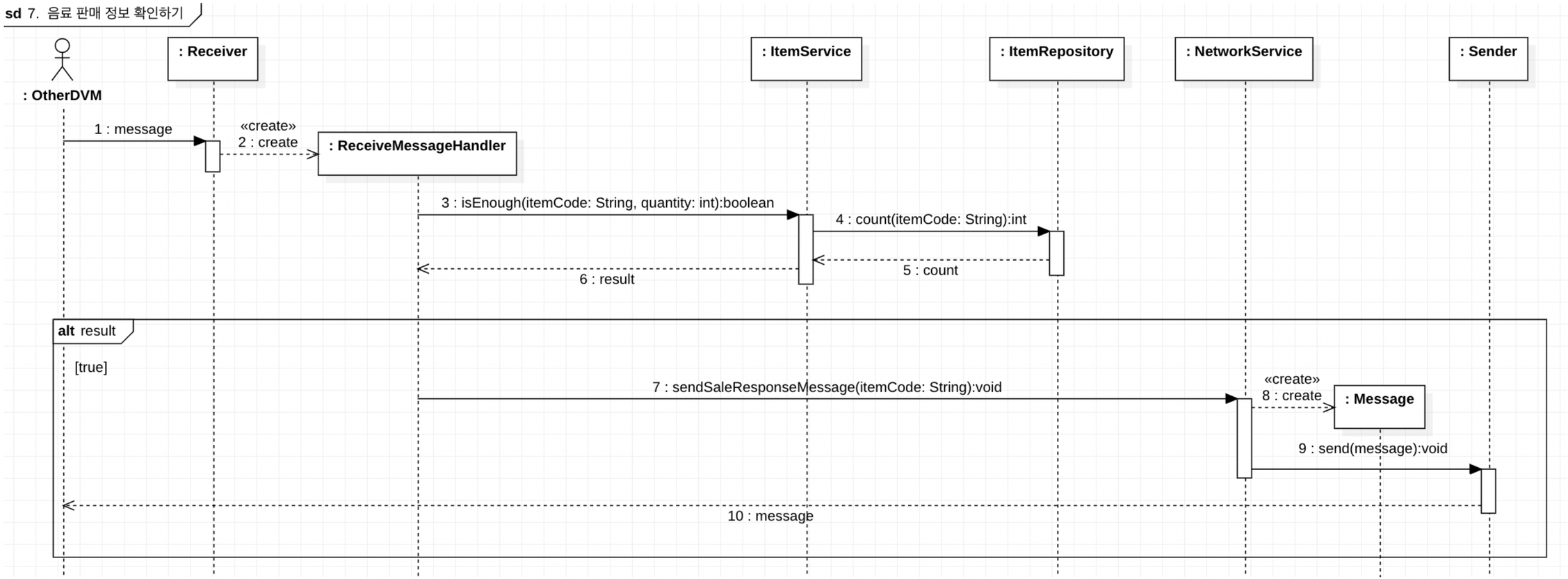




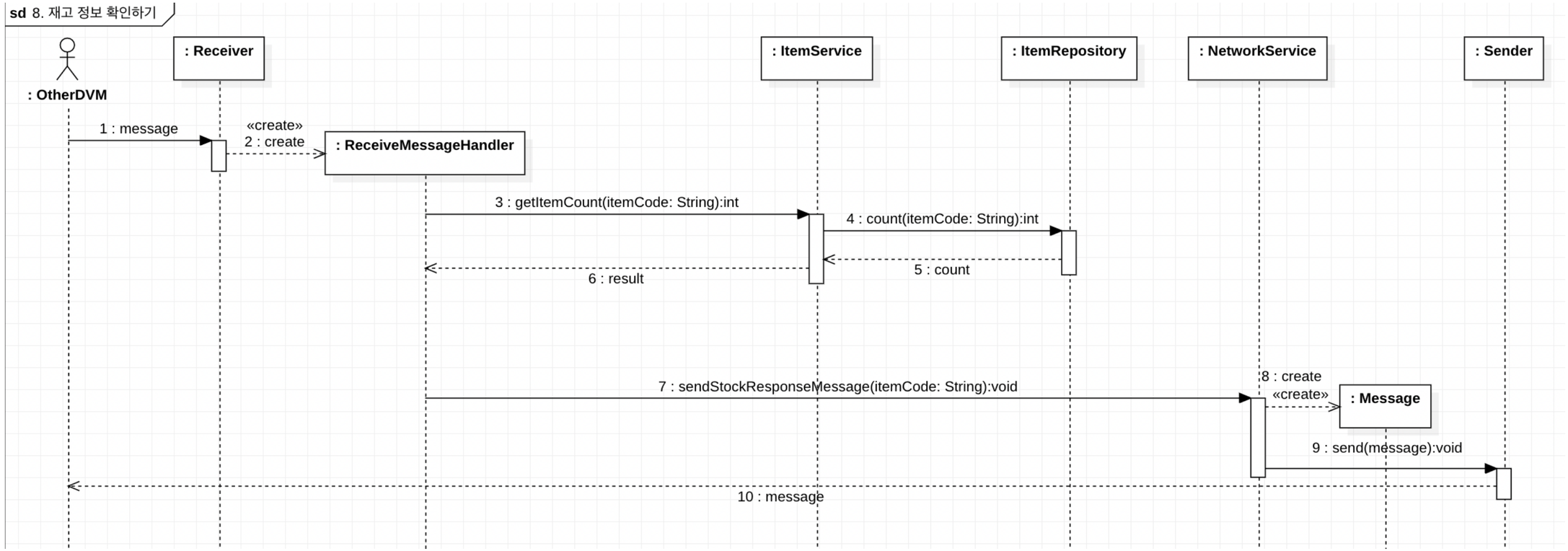
Define Interaction Diagrams



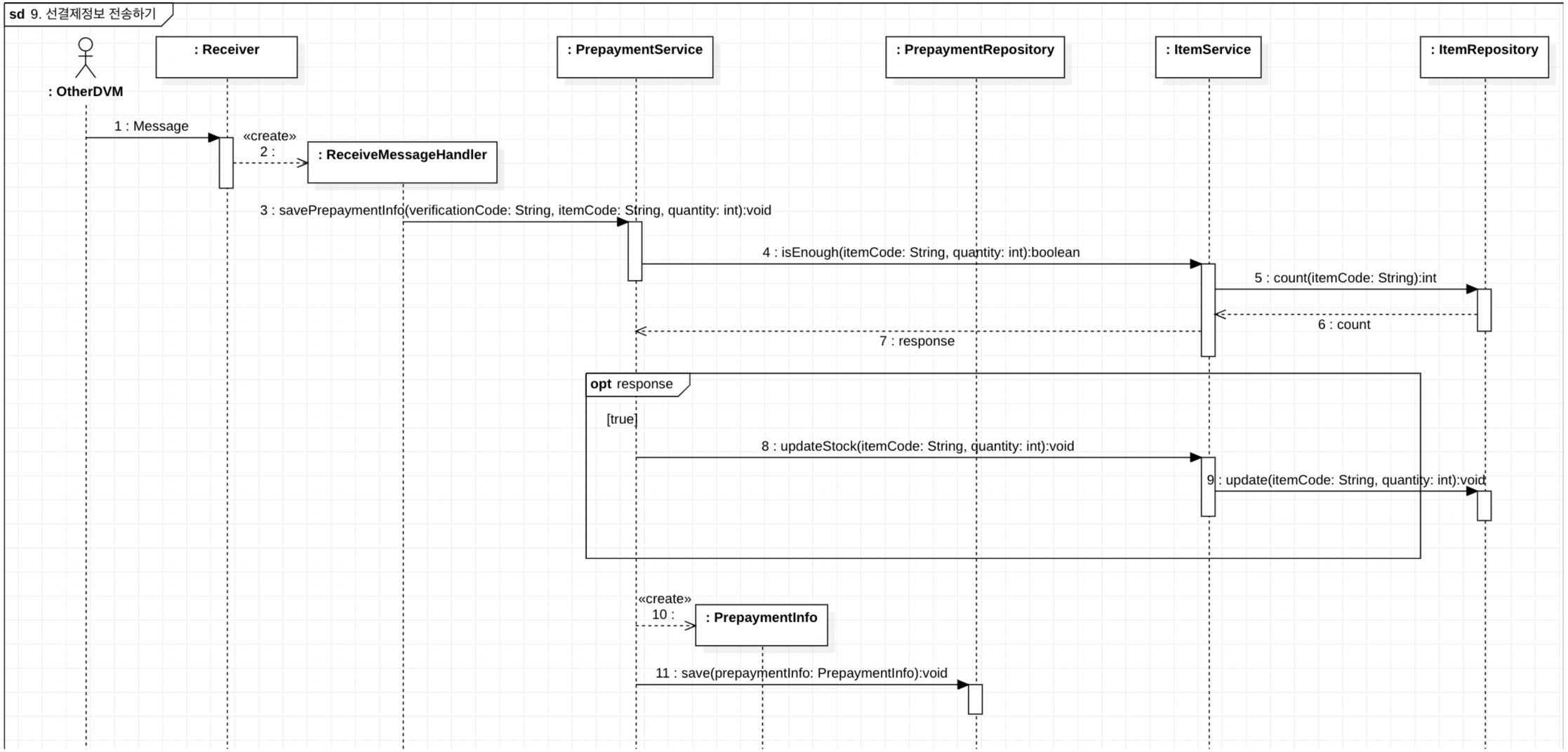
Define Interaction Diagrams



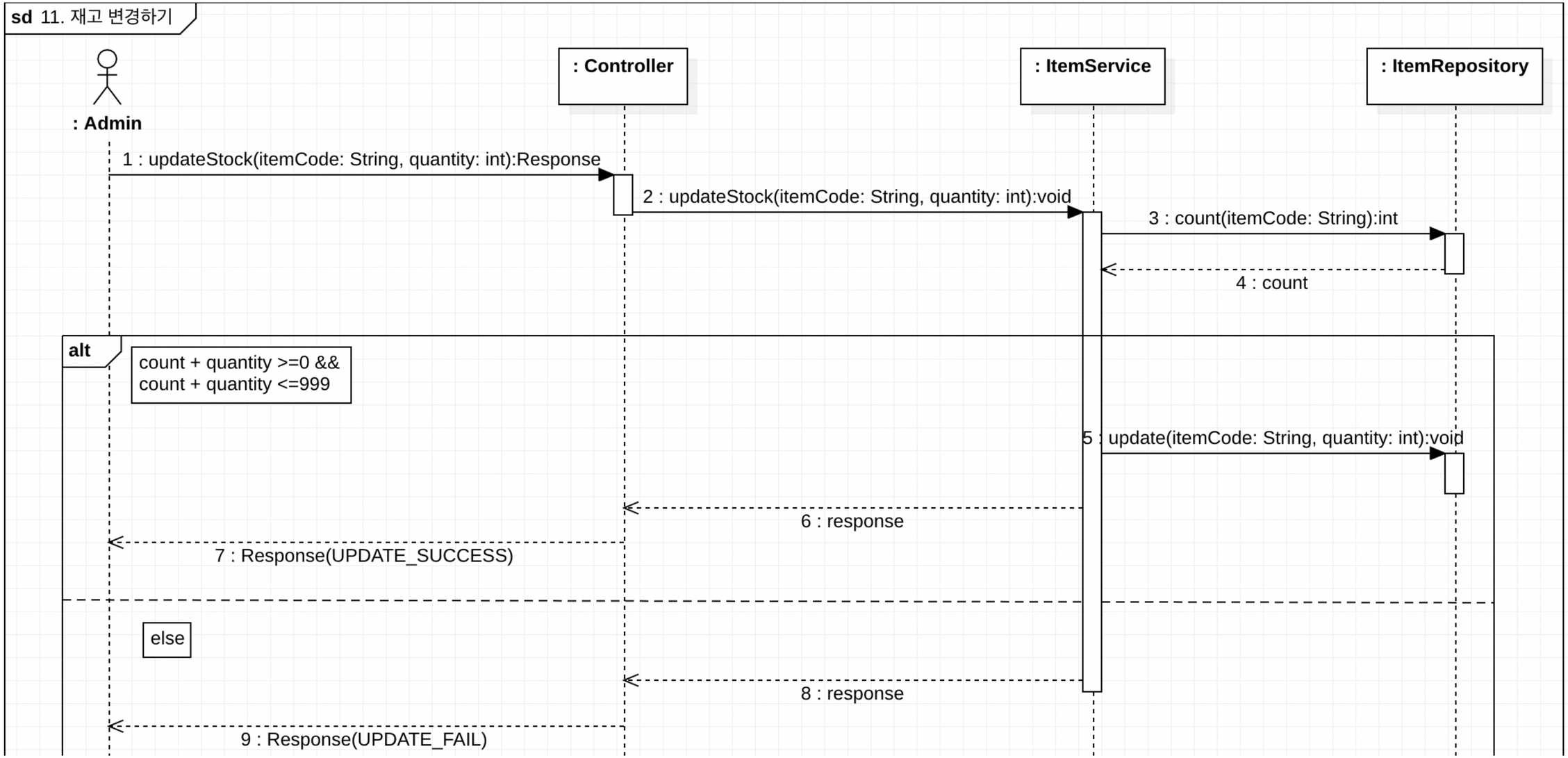
Define Interaction Diagrams



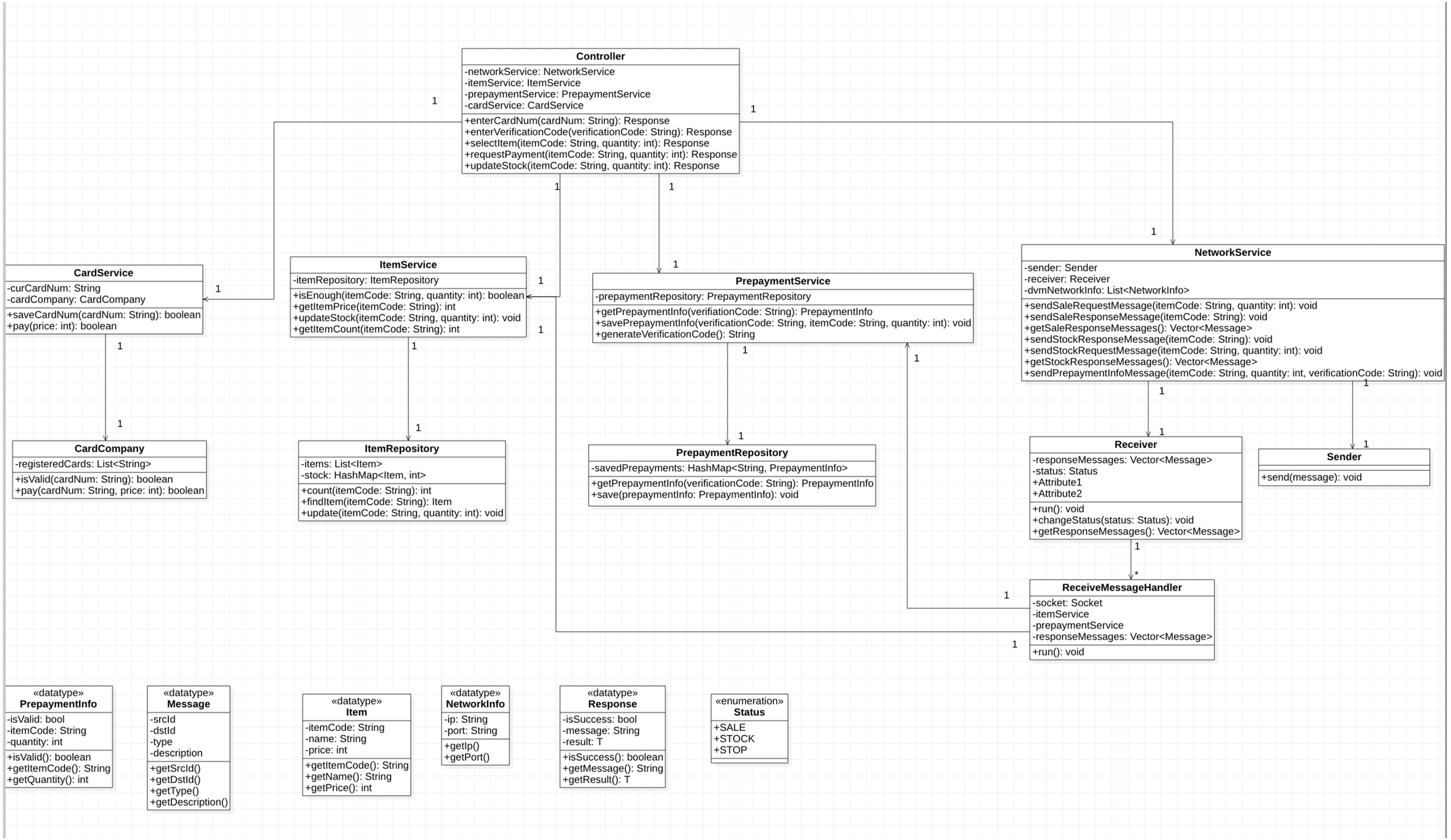
Define Interaction Diagrams



Define Interaction Diagrams



Define Design Class Diagrams



Design Traceability Analysis

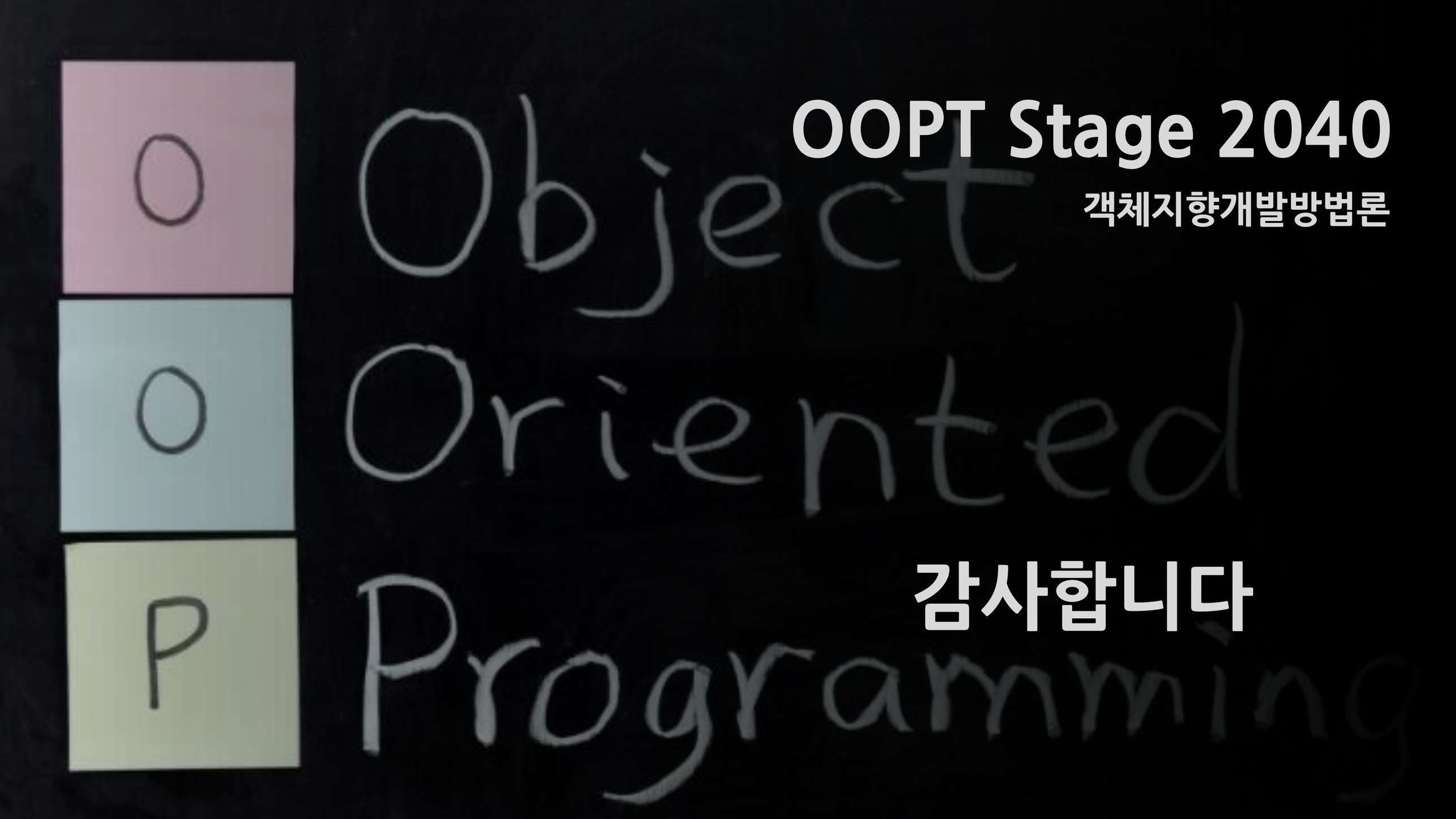
Functional Requirements	
1.1	카드 정보 읽기
1.2	인증코드 확인하기
2.1	음료 선택하기
3.1	결제 진행하기
4.1	대상 자판기 선정하기
4.2	선결제 진행하기
5.1	음료판매 정보 응답하기
5.2	재고 정보 응답하기
5.3	선결제 정보 저장하기
6.1	관리자 모드 진입하기
6.2	재고 변경하기
6.3	관리자 모드 종료하기

Use Case	
1	카드번호 입력하기
2	인증코드 입력하기
3	음료 선택하기
4	결제하기
5	대상 자판기 확인하기
6	선결제 하기
7	음료판매 정보 확인하기
8	재고 정보 확인하기
9	선결제 정보 전송하기
10	관리자 모드 진입하기
11	재고 변경하기
12	관리자 모드 종료하기

System Operation
enterCardNum
enterVerificationCode
enterItem
reqPayment
reqPrepayment
checkOtherDvmItem
checkRepository
enterAdmin
updateStock
exitAdmin
sendPrepaymentInfo

Method
enterCardNum(cardNum: String): Response
enterVerificationCode(verificationCode: String): Response
selectItem(itemCode: String, quantity: int): Response
requestPayment(itemCode: String, quantity: int): Response
updateStock(itemCode: String, quantity: int): Response
saveCardNum(cardNum: String): Boolean
pay(price: int): Boolean
isVaild(cardNum: String): Boolean
pay(cardNum: String, price: int): boolean
isEnough(itemCode: String, quantity: int): Boolean
getItemPrice(itemCode: String, quantity: int): boolean
updateStock(itemCode: String, quantity: int): void
getItemCount(itemCode: String): int
count(itemCode:String): int
findItem(itemCode:String): Item
update(itemCode:String, quantity:int): void
sendSaleRequestMessage(itemCode:String, quantity:int): void
sendSaleResponseMessage(itemCode:String): void
getSaleResponseMessages(): Vector
sendStockResponseMessage(itemCode:String): void
sendStockRequestMessage(itemCode:String, quantity:int): void
getStockResponseMessages(): Vector
sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): void
send(message): void
run(): void
changeStatus(status): void
getResponseMessages(): Vector
run(): void
isSuccess(): boolean
getMessage(): String
getResult(): T
getPrepaymentInfo(String verifiationCode): PrepaymentInfo
savePrepaymentInfo(String verificationCode, String itemCode, int quantity): void
generateVerificationCode(): String
getPrepaymentInfo(String verificationCode): PrepaymentInfo
save(PrepaymentInfo prepaymentInfo): void
isValid(): boolean
getItemCode(): String
getQuantity(): int

Class
Controller
CardService
CardCompany
ItemService
ItemRepository
NetworkService
Sender
Receiver
ReceiveMessageHandler
Response
PrepaymentService
PrepaymentRepository
PrepaymentInfo



OOPT Stage 2040

객체지향개발방법론

감사합니다