

# CONTENTS



2140 Design2141 Design Real Use Case

2142 Define Reports, UI, and Storyboards

**2143** Refine System Architecture

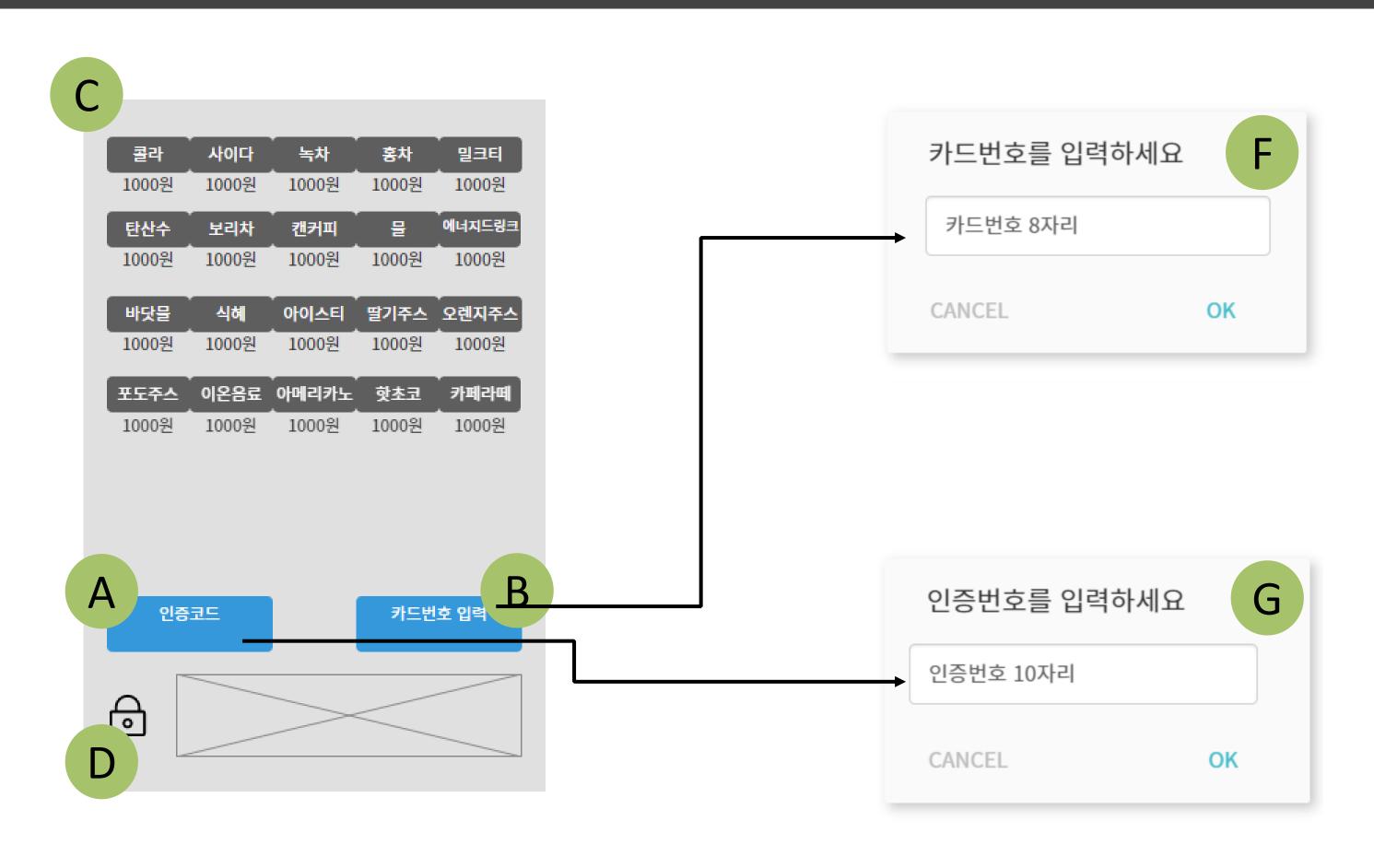
2144 Define Interaction Diagrams

2145 Define Design Class Diagrams

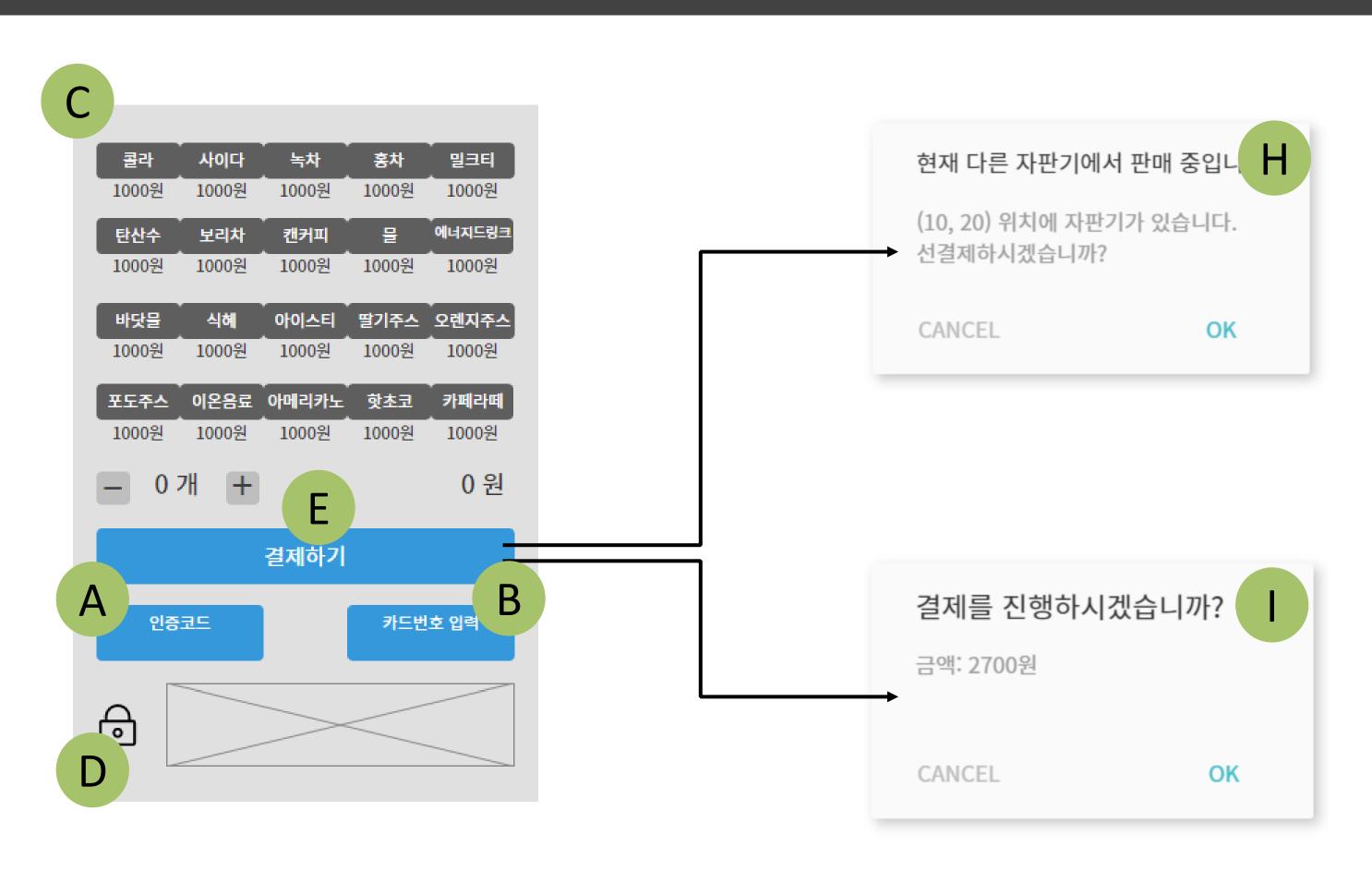
2146 Design Traceability Analysis

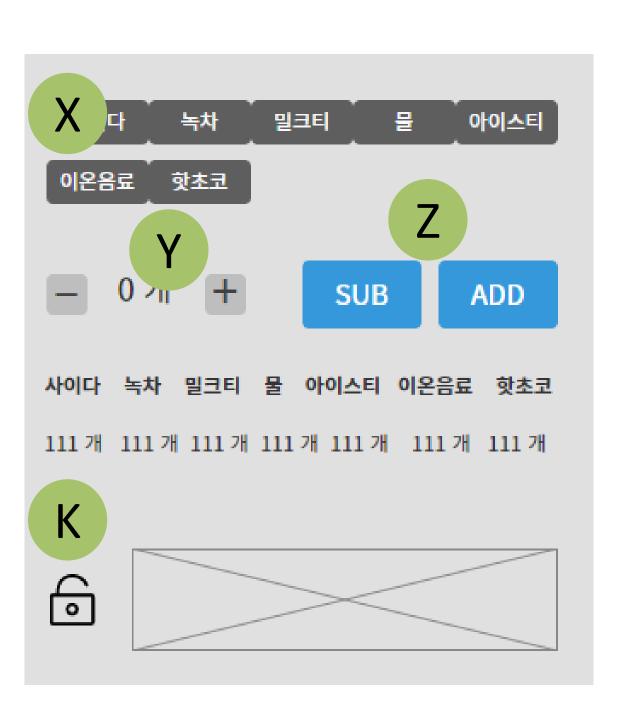
2147 Define Database Schema

## Define Reports, UI, and Storyboards



## Define Reports, UI, and Storyboards





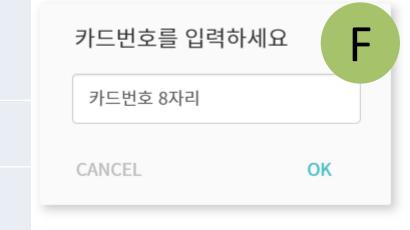
Except

#### Design Real Use Cases

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

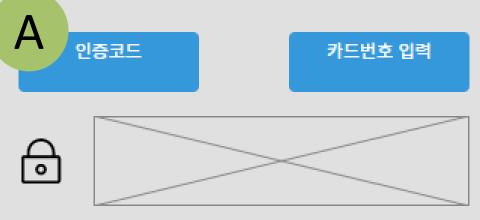
E1: 고객이 넣은 카드가 유효하지 않은 카드이면 유효하지 않음을 알리고 기본 상태를 유지한다.

콜라 사이다 녹차 홍차 밀크티 1000원 1000원 1000원 1000원 1000원 에너지드링크 탄산수 보리차 캔커피 믈 1000원 1000원 1000원 1000원 1000원 바닷물 아이스티 딸기주스 오렌지주스 1000원 1000원 1000원 1000원 1000원 카페라떼 이온음료 아메리카노 핫초코 1000원 1000원 1000원 1000원 1000원 카드번호 입력 인증코드 e



Use Case	2. 인증코드 입력하기	
Actor	고객	
Purpose	외부 자판기에서 선결제한 음료를 찾기 위해 인증코드를 입력한다.	
Overview	사용자는 인증코드를 입력하고 자판기는 인증코드가 유효한지 검사한다.	
Туре	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.	







#### 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. (A): Actor, (S): System 1. (A) 고객이 구매할 음료와 수량을 입력한다. (E1) **Typical** 2. (S) 현 자판기의 재고와 고객이 구매할 음료의 수량을 비교한다. (E2) 3. (A) 고객이 결제하기 버튼을 누른다 E

사이다 녹차 홍차 밀크티 1000원 1000원 1000원 1000원 1000원 탄산수 에너지드링크 보리차 캔커피 1000원 1000원 1000원 1000원 1000원 딸기주스 오렌지주스 바닷물 아이스티 1000원 1000원 1000원 1000원 1000원 이온음료 아메리카노 카페라떼 호초호 1000원 1000원 1000원 1000원 1000원 0 원 결제하기 카드번호 입력 인증코드 6

2147

Alter

Except

(E1): 잘못된 음료를 선택한 경우 잘못 되었음을 알리고 기본 상태로 돌아간다.

(E2): 재고가 충분하지 않다면 대상 자판기를 탐색한다.

Use Case	4. 결제하기					
2142 Actor 고객						
Purpose						
2143 Overview	선택한 음료의 수량이 자판기의 재고를 넘지 않을 때 사용자의 카드 정보를 읽어 결제를 요청한다. 결제 가능	시 결제한다.				
Туре	Evident					
2144 Cross Reference	Functions :R1.1, R2.1, R3.1	결제를 진행하시겠습	나까?			
	Use Case: 1. 카드 넣기, 3: 음료 선택하기	금액: 2700원				
2145 Pre-Requisites	선택한 음료의 개수가 현 자판기의 재고를 넘지 않는다.					
2146 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.	CANCEL	OK			
2147 Alter						
Aitei						
2140 Except	(E1) : 재고가 부족한 경우 결제 실패 메세지를 띄우고 기본 상태로 돌아간다. (E2) : 카드의 잔액이 없을 경우 결제 실패했음을 띄우고 기본 상태로 돌아간다.					

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
Туре	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.

	Use Case	6. 선결제 하기		
2142	Actor	고객		
	Purpose	선결제		
2143	Overview	고객이 선결제를 요청했을 때, 사용자의 카드 정보를 읽어 결제를 요청한다. 결제 가능 시 결제한다.		
	Туре	Evident	현재 다른 자판기에서 판매	중입니다 H
2144	Cross Reference	Functions: R1.1, R2.1, R4.1, R4.2	(10, 20) 위치에 자판기가 있습	습니다.
		Use Case: 1. 카드번호 입력하기, 3. 음료 선택하기, 5. 대상 자판기 확인하기	선결제하시겠습니까?	
2145	Pre-Requisites	대상 자판기가 선정 되어있다	CANCEL	ОК
<ul><li>2146</li><li>2147</li></ul>	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			
2140	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.		

Use Case	7. 음료 판매 정보 확인하기
Actor	외부 자판기
Purpose	음료 판매 여부 파악
Overview	외부 자판기에서 현 자판기의 음료 판매 정보를 확인한다.
Туре	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.

Use Case	8. 재고 정보 확인하기
Actor	외부 자판기
Purpose	재고 파악
Overview	외부 자판기에서 현 자판기의 음료 재고 정보를 확인한다.
Туре	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.
A.1.	
Alter	
Except	

	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.
	Туре	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	
2140	Except	(E1) : 선결제 한 음료의 재고가 현 자판기의 재고보다 많다면, 재고를 빼지 않고, 유효하지 않은 선결제임을 저장한다. 상품 정보 업데이트는 되지 않는다.

## 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 (A): Actor, (S): System Typical 1. (A) 관리자 모드로 진입한다 (E1) 2. (S) 관리자 화면을 보여준다.



2147

Alter

Except

pt E1: 관리자가 아닐 시 진입하지 못한다

#### Design Real Use Cases

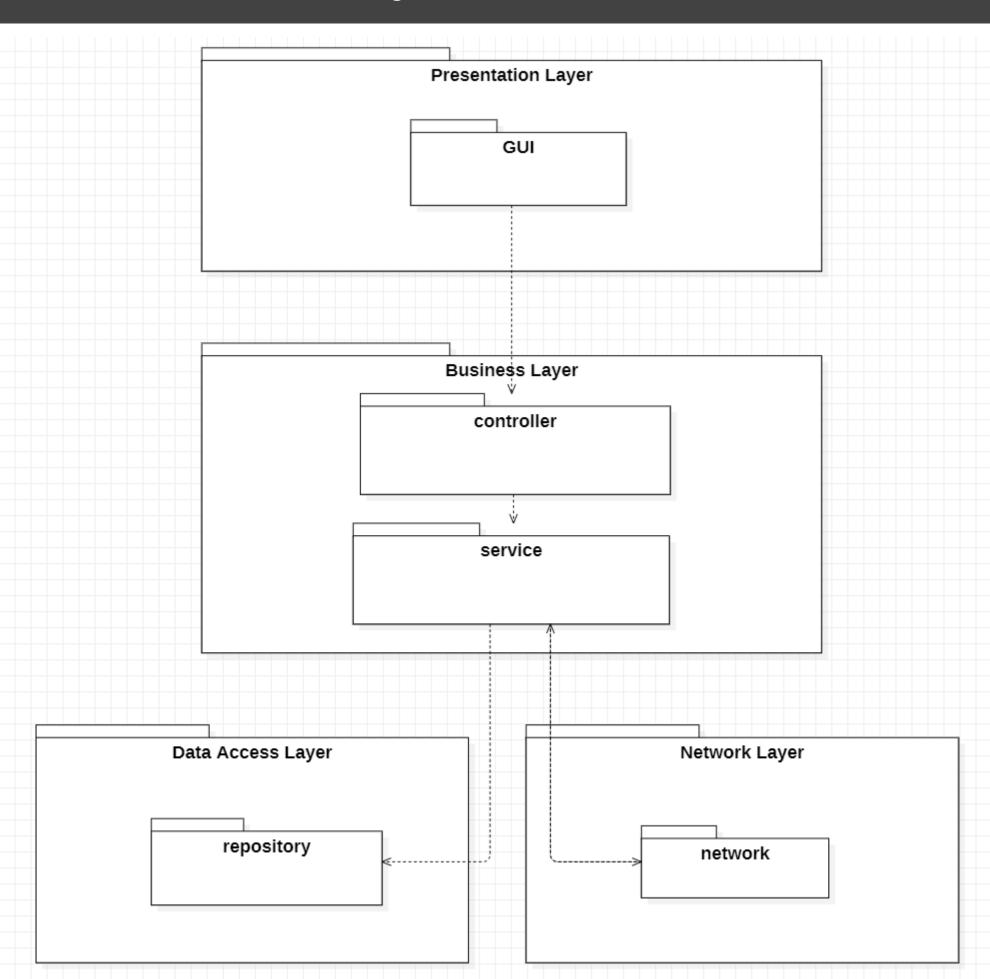
**Use Case** 11. 재고 변경하기 Actor 관리자 Purpose 재고 변경 아이스티 사이다 녹차 밀크티 Overview 관리자 모드 변경을 원하는 음료를 선택하여 재고를 변경한다. 이온음료 핫초코 Type **Evident Cross Reference** Functions: R6.1, R6.2 **SUB ADD** Use Cases: 10. 관리자 모드 진입하기 아이스티 이온음료 핫초코 **Pre-Requisites** 관지라 모드에 진입한 상태 111 개 111 개 111 개 111 개 111 개 111 개 (A): Actor, (S): System **Typical** 1. (A) 관리자가 재고를 변경할 음료들과 개수를 입력한다. (E1) ම 2. (S) 시스템이 해당 음료의 재고를 변경한다. Alter Except E1: 관리자가 재고를 0~999가 아닌 숫자로 만들려고 할 때 에러임을 알리고 이 행위를 무시한다.

Except

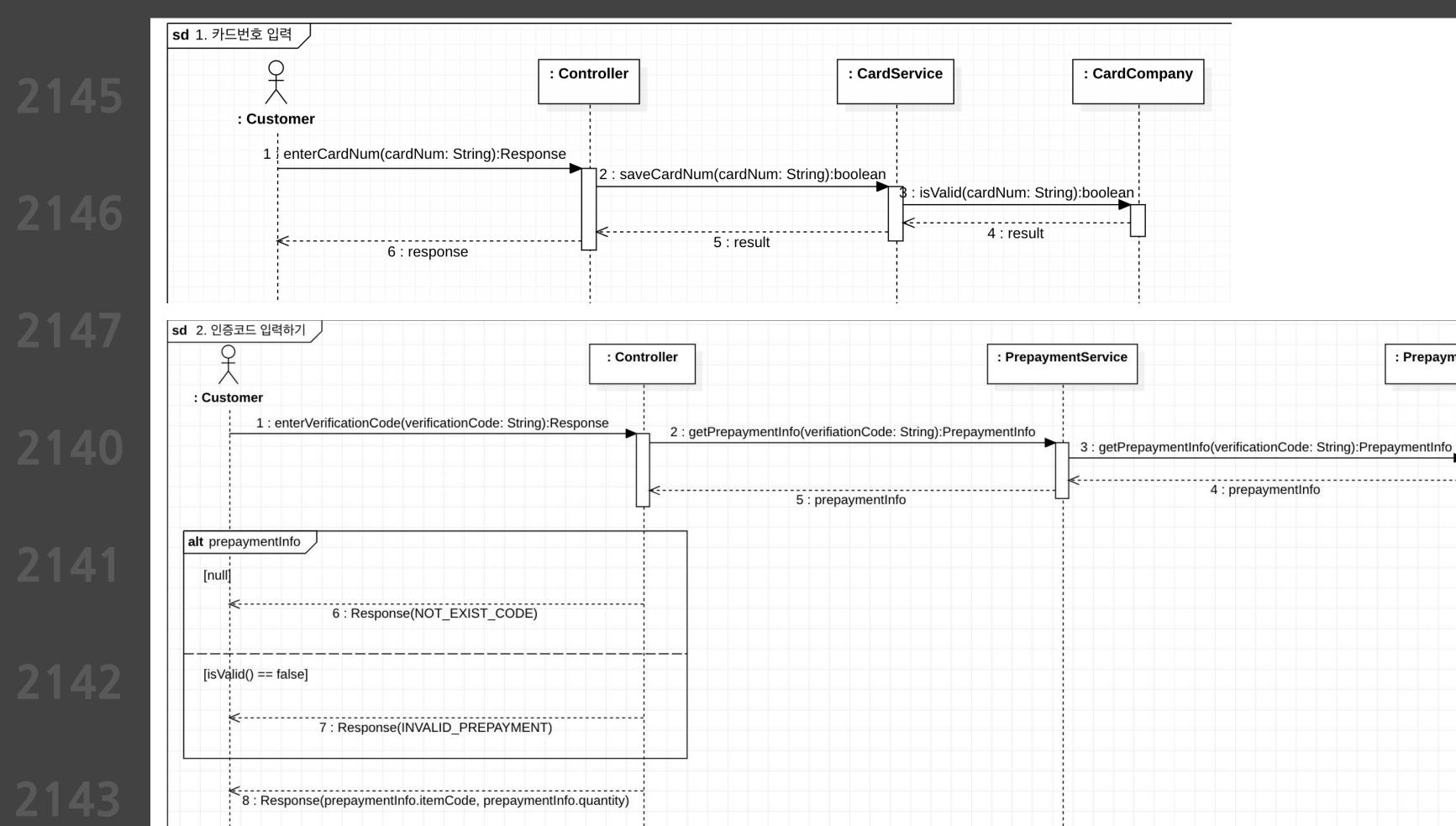
#### Design Real Use Cases

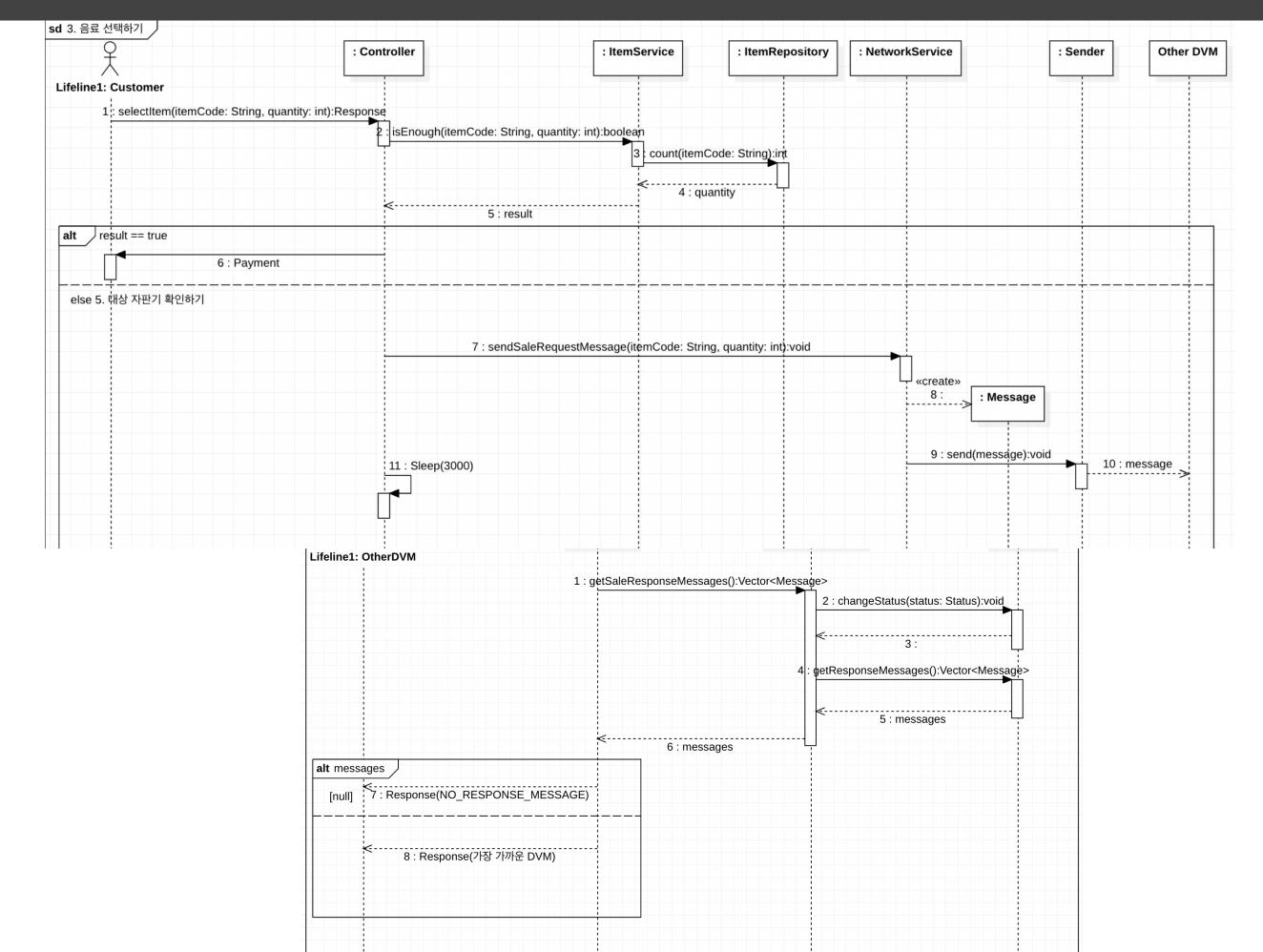
**Use Case** 12. 관리자 모드 종료하기 Actor 관리자 Purpose 관리자 모드 종료 아이스티 사이다 녹차 밀크티 Overview 관리자 모드를 종료한다. 이온음료 핫초코 Type **Evident Cross Reference** Functions: R6.1, R6.3 0 개 **SUB ADD** Use Cases: 10. 관리자 모드 진입하기 물 아이스티 이온음료 핫초코 밀크티 **Pre-Requisites** 관리자 모드에 진입한 상태 111 개 111 개 111 개 111 개 111 개 111 개 K K (A): Actor, (S): System Typical 1. (A) 관리자 모드를 종료한다. ම 2. (S) 기본 상태로 돌아간다. Alter

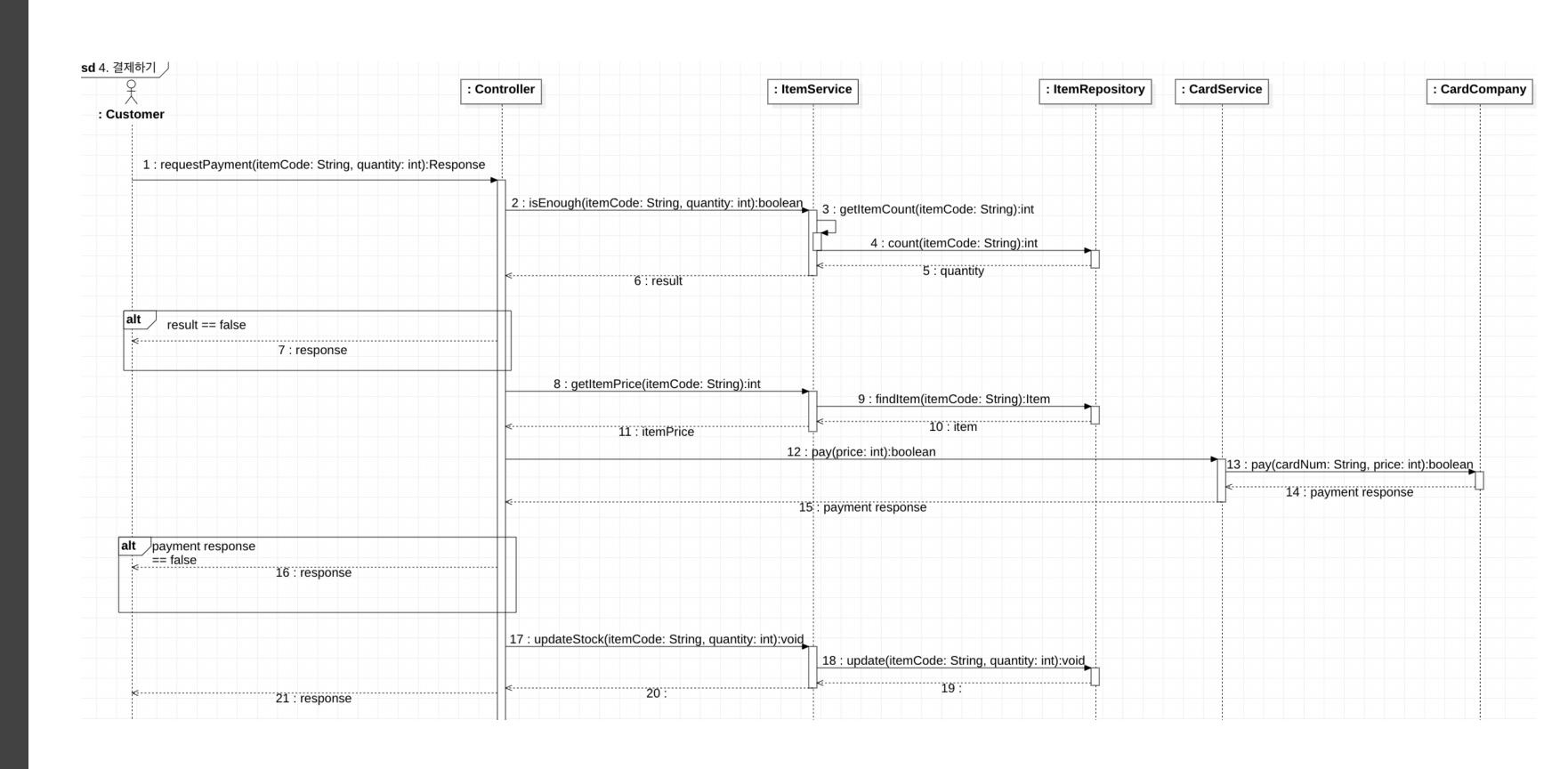
# Refine System Architecture

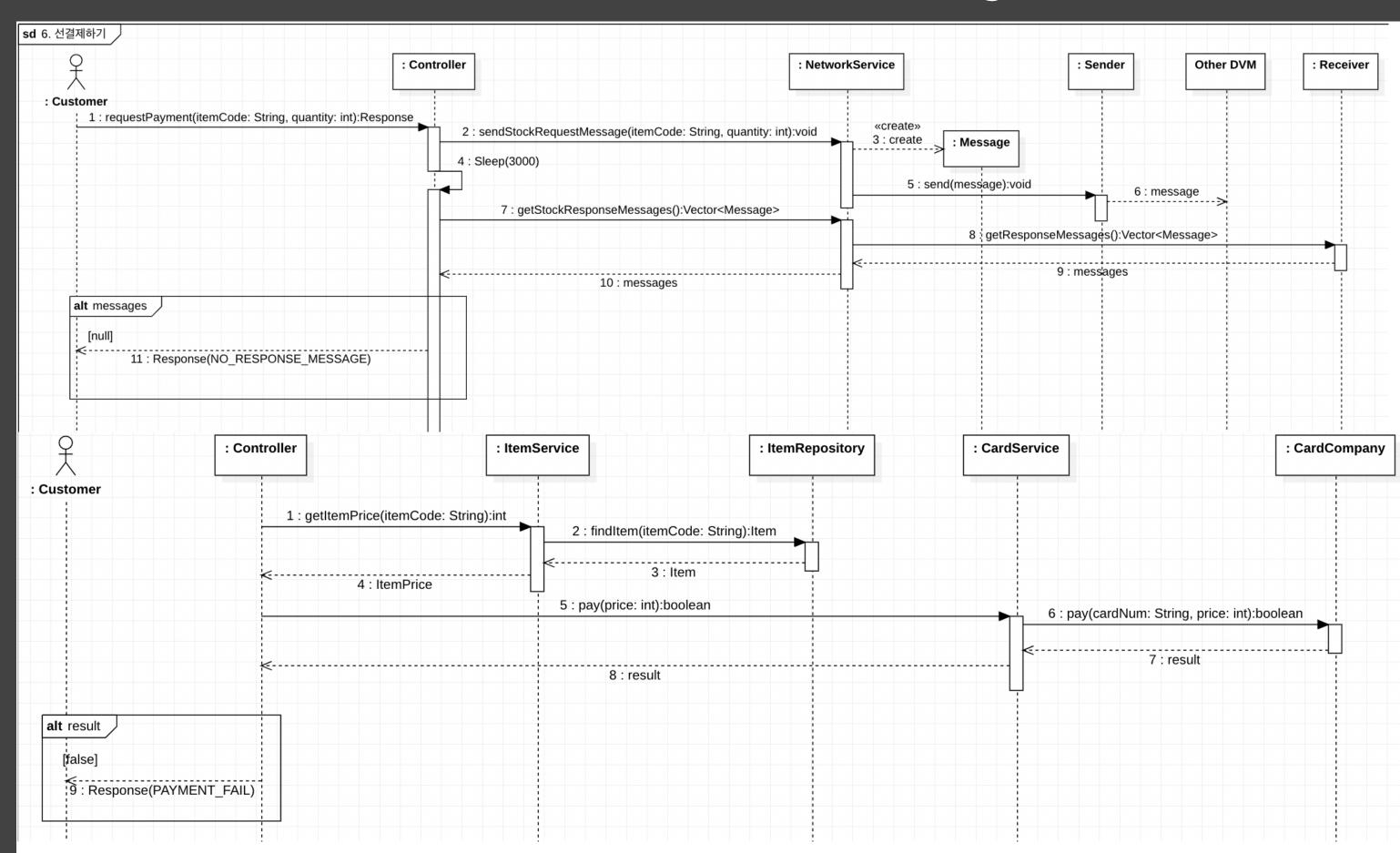


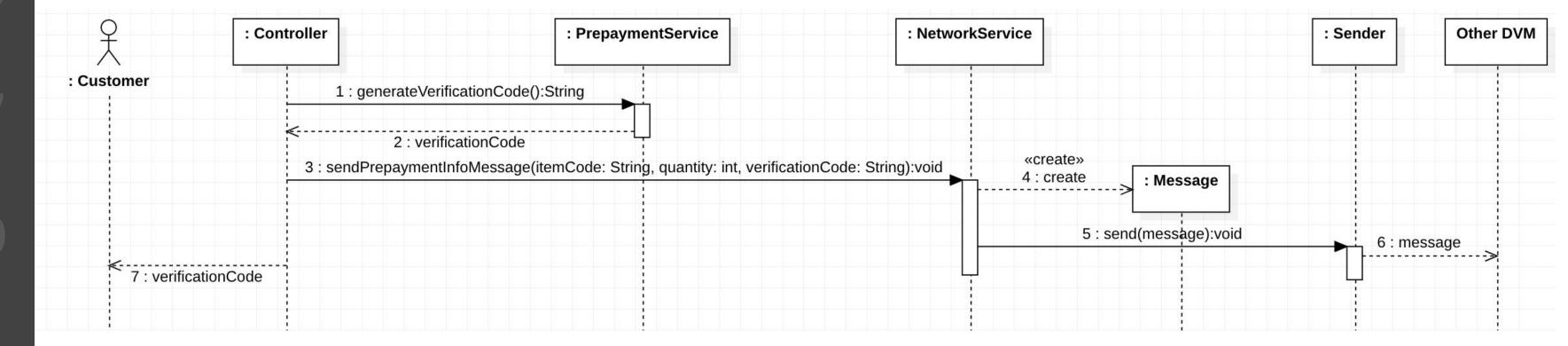
: PrepaymentRepository

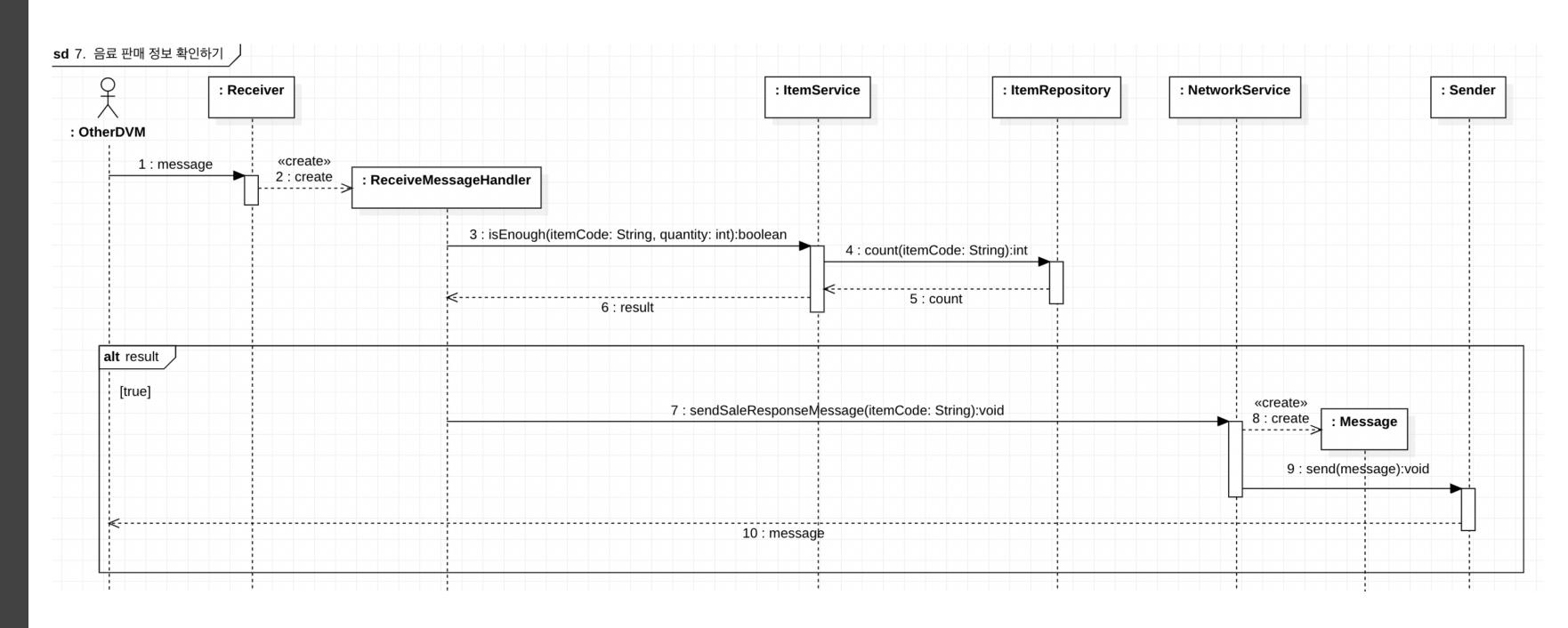


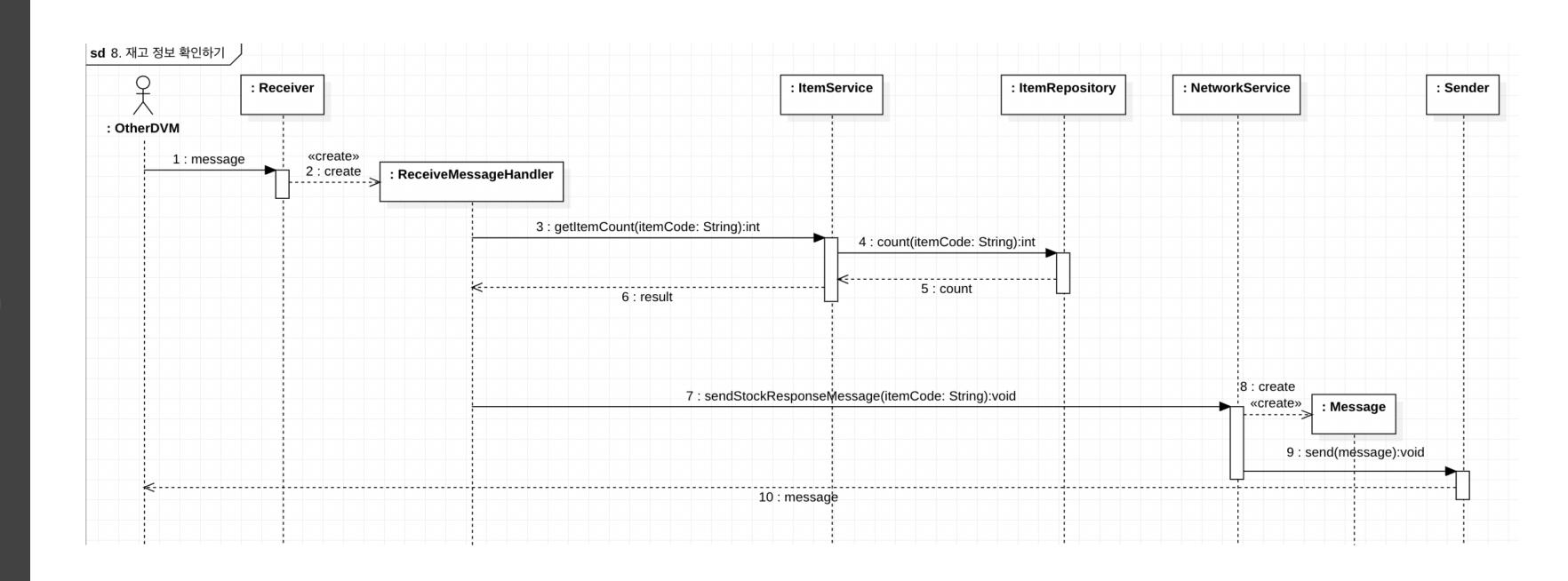


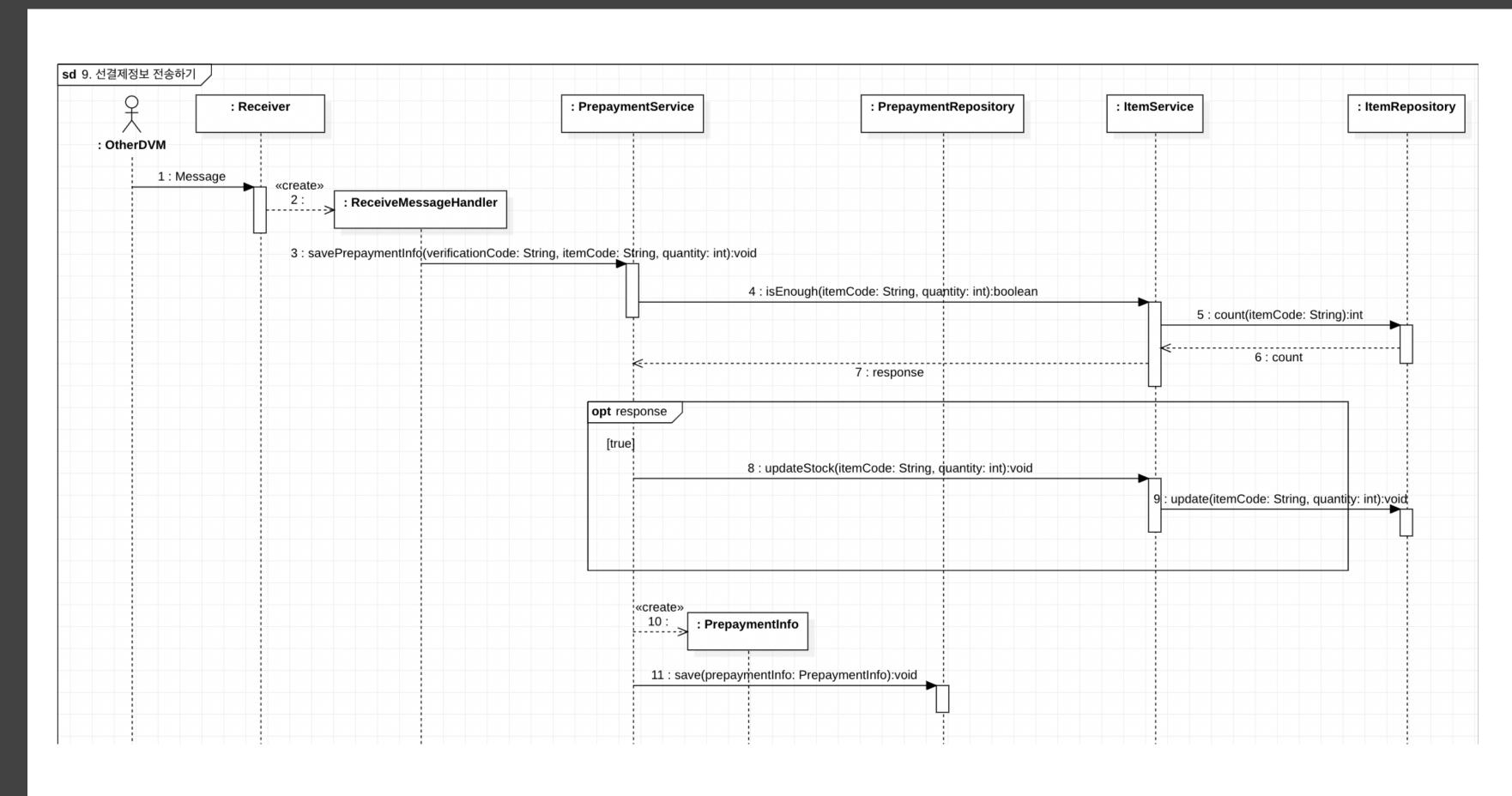


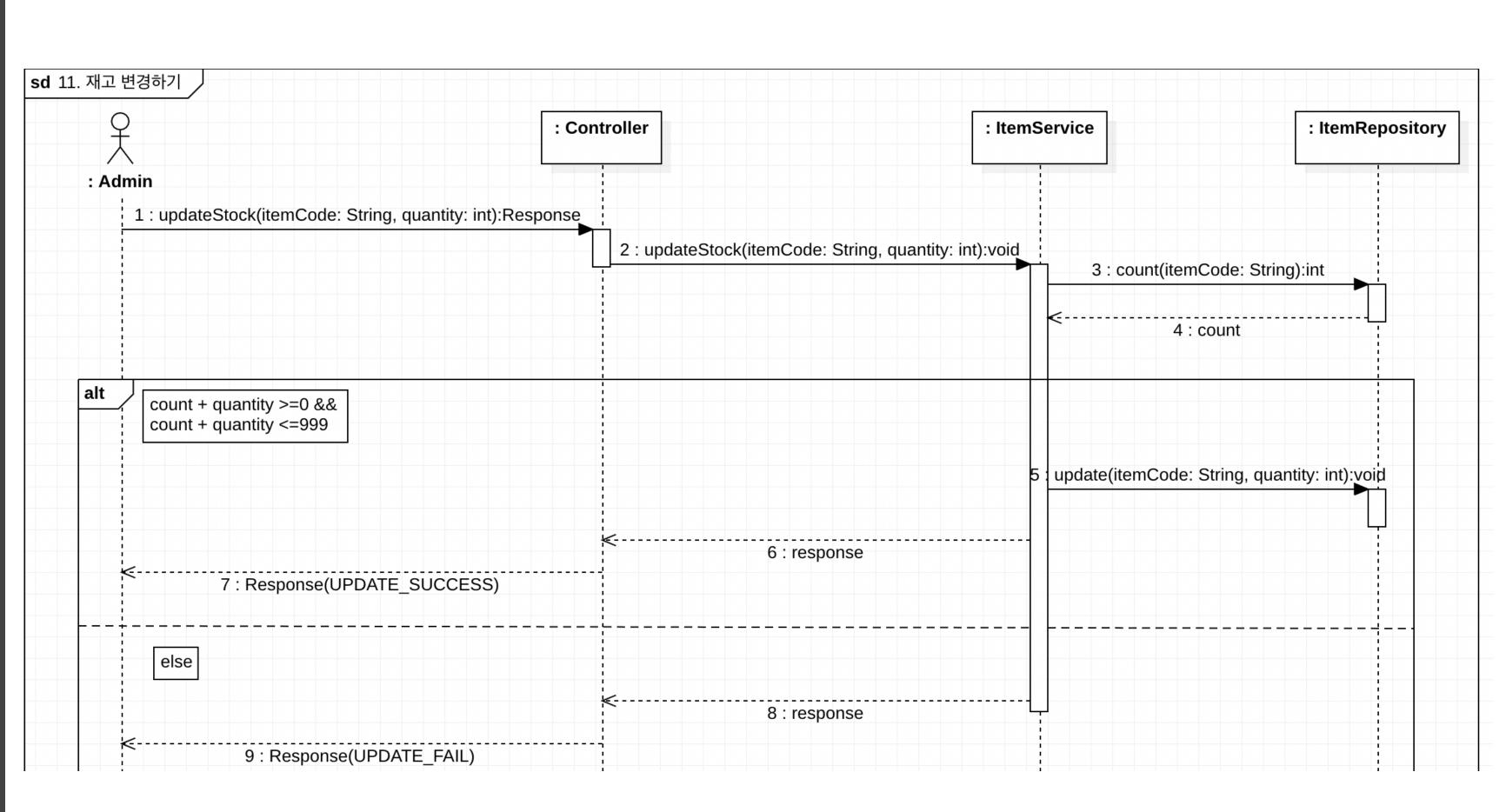












#### Define Design Class Diagrams

NetworkService

Sender

+send(message): void

-networkService: NetworkService -itemService: ItemService -prepaymentService: PrepaymentService -cardService: CardService +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 **ItemService** -sender: Sender -itemRepository: ItemRepository **PrepaymentService** -curCardNum: String -dvmNetworkInfo: List<NetworkInfo> +isEnough(itemCode: String, quantity: int): boolean -cardCompany: CardCompany -prepaymentRepository: PrepaymentRepository +sendSaleRequestMessage(itemCode: String, quantity: int): void +getItemPrice(itemCode: String): int getPrepaymentInfo(verifiationCode: String): PrepaymentInfo +saveCardNum(cardNum: String): boolean +sendSaleResponseMessage(itemCode: String): void +updateStock(itemCode: String, quantity: int): void +savePrepaymentInfo(verificationCode: String, itemCode: String, quantity: int): void +pay(price: int): boolean -getSaleResponseMessages(): Vector<Message> +getItemCount(itemCode: String): int +generateVerificationCode(): String sendStockResponseMessage(itemCode: String): void +sendStockRequestMessage(itemCode: String, quantity: int): void +getStockResponseMessages(): Vector<Message> +sendPrepaymentInfoMessage(itemCode: String, quantity: int, verificationCode: String): void Receiver CardCompany **ItemRepository** PrepaymentRepository -responseMessages: Vector<Message> -registeredCards: List<String> -items: List<Item> -savedPrepayments: HashMap<String, PrepaymentInfo> -status: Status -stock: HashMap<Item, int> +isValid(cardNum: String): boolean +Attribute1 +getPrepaymentInfo(verificationCode: String): PrepaymentInfo +count(itemCode: String): int +pay(cardNum: String, price: int): boolean +Attribute2 +save(prepaymentInfo: PrepaymentInfo): void +findItem(itemCode: String): Item +run(): void +update(itemCode: String, quantity: int): void +changeStatus(status: Status): void +getResponseMessages(): Vector<Message> ReceiveMessageHandler -socket: Socket -responseMessages: Vector<Message> «datatype» «datatype» «datatype» «datatype» PrepaymentInfo NetworkInfo Response Item Status -isValid: bool -isSuccess: bool -srcId -ip: String -itemCode: String +SALE -itemCode: String -dstId port: String -message: String -name: String +STOCK -result: T -quantity: int -type +getlp() +STOP -description +isValid(): boolean +getPort() +isSuccess(): boolean +getItemCode(): String +getItemCode(): String +getSrcId() +getMessage(): String +getName(): String +getQuantity(): int +getDstId() +getResult(): T +getPrice(): int +getType() +getDescription()

# Design Traceability Analysis

21	<b>4</b> 5

2 인증코드 확인하기 2 인증코드 입력하기 enterVerificationCode selectitem(itemCode: String, quantity: int): Response requestPayment(itemCode: String, quantity: int): Response updateStock(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 pay(pr										
기 전 경보 위기	Funct	ional Requirements			Use Case		System Operation			Method
2 인클코드 확인하기	1,1	카드 정보 읽기	<b></b>	1	카드번호 입력하기		enterCardNum		→[	
1 등로 선택하기						1				
1 일제 진행하기 4 일제하기 7 1 대상 자판기 선정하기 5 대상 자판기 확인하기 7 2 세계 진행하기 7 3 음표판매 정보 확인하기 7 1 음표판매 정보 확인하기 7 3 음표판매 정보 확인하기 9 1 선결제 정보 자장하기 1 1 대상 자판기 선정하기 1 1 1 제고 정보 확인하기 7 2 세계 전행하기 1 1 1 제고 정보 확인하기 1 1 1 제고 정보 확인하기 1 1 1 제고 보험하기 1 1 제고 변경하기 1 1 제고 변경하기 1 1 1 1 제고 변경하기 1 1 1 제고 변경하기 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1,2	인증코드 확인하기	<b></b>	2	인증코드 입력하기		enterVerificationCode			
1 전체 진행하기	2,1	으로 서태하기		2	으로 서태하기	<b>└</b>	antaritam			
1 대상 자판기 선정하기 1 대상 자판기 설정하기 2 선절세 전병하기 1 등료판매 정보 확단하기 2 세고 정보 확단하기 3 선절세 정보 자장하기 1 관리자 모드 권임하기 1 전계 정보 자장하기 1 가입 한 지수는 지원 기계 전 기계	2,1	금표 선택에게			급표 연락에게	<u> </u>	ententen		X	
1 내성 사반기 전성하기	3,1	결제 진행하기	<b></b>	4	결제하기		reqPayment			
2         신결제 진행하기         6         선결제 진행하기         theckOtherDvmittem         the EnoughtlemcCeds Sting, quantify, int): Booleam quantify; booleam quantify, int): booleam qua	4.1	대상 자파기 선정하기		5	대상 자파기 화인하기	/,	regPrenayment			isVaild(cardNum: String): Boolean
### Seturn 정보 용답하기 ### Seturn 정보 용답하기 ### Seturn 정보 확인하기 ### Ara 정보 확인하기 ### Of Meak Position ### Of Meak Positio	*• '	-110 11 12 11 12 13 11 11			-110 1 1 1 1 1 1 1 1 1 1 1	/	Тефі тераушені		$\Rightarrow$	pay(cardNum: String, price: int): boolean
1 음로판매 정보 응답하기	4.2	선결제 진행하기	<b></b>	6	선결제 하기	/ ,	checkOtherDvmltem			
8 세고 정보 확인하기 3 선결제 정보 저장하기 1 관리자 모드 진입하기 3 한리자 모드 중료하기 11 세고 변경하기 3 한리자 모드 종료하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 13 한리자 모드 종료하기 14 개고 변경하기 15 관리자 모드 종료하기 16 관리자 모드 종료하기 17 관리자 모드 종료하기 18 관리자 모드 종료하기 19 선결제 정보 전송하기 11 제고 변경하기 11 제고 변경하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 13 한리자 모드 종료하기 14 관리자 모드 종료하기 15 관리자 모드 종료하기 16 관리자 모드 종료하기 17 관리자 모드 종료하기 18 관리자 모드 종료하기 19 선결제 정보 전송하기 19 선결제 정보 전송하기 10 관리자 모드 종료하기 10 관리자 모드 종료하기 11 제고 변경하기 11 제고 변경하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 13 관리자 모드 종료하기 14 관리자 모드 종료하기 15 관리자 모드 종료하기 16 관리자 모드 종료하기 17 관리자 모드 종료하기 17 관리자 모드 종료하기 18 관리자 모드 종료하기 19 관리자 모드 종료하기 19 관리자 모드 종료하기 10 관리자 모드 종료하기 10 관리자 모드 종료하기 11 제고 변경하기 11 제고 변경하기 12 관리자 모드 종료하기 11 제고 변경하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 15 관리자 모드 종료하기 16 관리자 모드 종료하기 17 관리자 모드 종료하기 17 관리자 모든 종료하기 18 관리자 모드 종료하기 19 관리자 모드 종료하기 19 관리자 모드 종료하기 10 관리자 모드 종료하기 11 제고 변경하기 11 제고 변경하기 12 관리자 모드 종료하기 10 관리자 모드 종료하기 11 제고 변경하기 12 관리자 모드 종료하기 11 제고 변경하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 15 관리자 모드 종료하기 16 관리자 모든 종료하기 17 교리자 모든 종료하기 17 교리자 모든 종료하기 18 관리자 모든 종료하기 19 관리자 모든 종료하기 19 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 11 제고 변경하기 12 관리자 모든 종료하기 11 제고 변경하기 12 관리자 모든 종료하기 15 관리자 모든 종료하기 16 관리자 모든 종료하기 17 교리자 모든 종료하기 17 교리자 모든 종료하기 18 교리자 모든 종료하기 19 프리자 모든 종료하기 19 프로마 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 11 제고 변경하기 12 관리자 모든 종료하기 10 관리자 모든 종료하기				<b>-</b>		1/	1 10 "	MARKET /		
8 제고 정보 확인하기 3 선결제 정보 저장하기 1 관리자 모드 진입하기 3 관리자 모드 전입하기 11 개고 변경하기 3 관리자 모드 종료하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 11 개고 변경하기 3 관리자 모드 종료하기 12 관리자 모드 종료하기 11 광리자 모드 종료하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 13 관리자 모드 종료하기 14 관리자 모드 종료하기 15 관리자 모드 종료하기 16 관리자 모드 종료하기 17 관리자 모드 종료하기 18 제고 변경하기 19 변경하기 19 변경하기 19 변경하기 10 관리자 모드 종료하기 11 개고 변경하기 11 개고 변경하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 13 관리자 모드 종료하기 14 관리자 모드 종료하기 15 관리자 모드 종료하기 16 관리자 모드 종료하기 17 관리자 모드 종료하기 18 제고 변경하기 19 관리자 모드 종료하기 19 관리자 모드 종료하기 10 관리자 모드 종료하기 11 개고 변경하기 11 개고 변경하기 12 관리자 모드 종료하기 11 개고 변경하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 13 관리자 모드 종료하기 14 관리자 모드 종료하기 15 관리자 모드 종료하기 16 관리자 모드 종료하기 17 관리자 모드 종료하기 18 관리자 모드 종료하기 19 관리자 모드 종료하기 19 관리자 모드 종료하기 10 관리자 모드 종료하기 10 관리자 모드 종료하기 10 관리자 모드 종료하기 11 개고 변경하기 11 개고 변경하기 12 관리자 모드 종료하기 15 관리자 모든 종료하기 16 관리자 모든 종료하기 17 관리자 모든 종료하기 18 프리자 모든 종료하기 19 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 11 개고 변경하이 12 관리자 모든 종료하기 11 대고 변경하이 12 관리자 모든 종료하기 15 프리자 모든 종료하기 16 프리자 모든 종료하기 17 프로디자 모든 종료하기 18 프리자 모든 종료하기 19 프리자 모든 종료하기 10 관리자 모든	5.1	금됴판매 성모 응납하기	<b>&gt;</b>			<b>/</b> /	checkKepository			
9 선결제 정보 전장하기 10 관리자 모드 진입하기 21 재고 변경하기 32 관리자 모드 중로하기 12 관리자 모드 중로하기 13 관리자 모드 중로하기 14 관리자 모드 중로하기 15 관리자 모드 중로하기 16 관리자 모든 중로하기 17 대 전 변경하기 18 관리자 모든 중로하기 18 관리자 모든 중로하기 19 관리자 모든 중로하기 19 관리자 모든 중로하기 10 관리자 모든 중로하기 10 관리자 모든 중로하기 11 대고 변경하기 12 관리자 모든 중로하기 12 관리자 모든 중로하기 12 관리자 모든 중로하기 13 관리자 모든 중로하기 14 관리자 모든 중로하기 15 관리자 모든 중로하기 16 관리자 모든 중로하기 17 관리자 모든 중로하기 18 관리자 모든 중로하기 18 관리자 모든 중로하기 19 관리자 모든 중로하기 19 관리자 모든 중로하기 10 관리자 모든 중로하기 11 대고 변경하기 12 관리자 모든 중로하기 12 관리자 모든 중로하기 12 관리자 모든 중로하기 12 관리자 모든 중로하기 19 관리자 모든 중로하기 10 관리자 모든 중로하기 11 대고 변경하기 12 관리자 모든 중로하기 12 관리자 모든 중로하기 12 관리자 모든 중로하기 12 관리자 모든 중로하기 10 관리자 모든 전용하기 11 대고 변경하기 12 관리자 모든 전용하기 12 관리자 모든 전용하기 12 관리자 모든 전용하기 12 관리자 모든 전용하기 12 관리자 모든 중로하기 10 관리자 모든 전용하기 12 관리자 모든 전용하기 12 관리자 모든 전용하기 12 관리자 모든 전용하기 13 대로 변환하는 전승 전용	5,2	재고 정보 응단하기	<b>&gt;</b>	8	재고 정보 화이하기	ν.	enterAdmin	MINITAL	7	
### Propagation of P	-,~	- 1 - 0 - 0 - 1 / 1		<u> </u>	1 0 - 7 L M 1	/		A LANDA LAND		
1 관리자 모드 진입하기 2 재고 변경하기 3 관리자 모드 중료하기 10 관리자 모드 중료하기 11 재고 변경하기 12 관리자 모드 중료하기 12 관리자 모드 중료하기 12 관리자 모드 중료하기 12 관리자 모드 중료하기 13 관리자 모든 중료하기 14 개고 변경하기 15 관리자 모든 중료하기 15 관리자 모든 중료하기 16 관리자 모든 중료하기 17 개고 변경하기 18 관리자 모든 중료하기 18 관리자 모든 중료하기 19 관리자 모든 중료하기 19 관리자 모든 중료하기 10 관리자 모든 중료하기 10 관리자 모든 중료하기 11 개고 변경하기 12 관리자 모든 중료하기 12 관리자 모든 중료하기 12 관리자 모든 중료하기 12 관리자 모든 중료하기 13 관리자 모든 중료하기 14 개고 변경하는 전략 문제 대한	5.3	선결제 정보 저장하기	<b></b>	9	선결제 정보 전송하기	/_	updateStock		1/2	
11 개고 변경하기 12 관리자 모드 종료하기 11 개고 변경하기 12 관리자 모드 종료하기 11 개고 변경하기 12 관리자 모드 종료하기 12 관리자 모드 종료하기 13 항 관리자 모든 종료하기 14 관리자 모든 종료하기 15 관리자 모든 종료하기 16 관리자 모든 종료하기 17 관리자 모든 종료하기 18 관리자 모든 종료하기 18 관리자 모든 종료하기 19 관리자 모든 종료하기 19 관리자 모든 종료하기 10 관리자 모든 종료하기 10 관리자 모든 종료하기 11 개고 변경하기 12 관리자 모든 종료하기 11 개고 변경하기 12 관리자 모든 종료하기 12 관리자 모든 종료하기 12 관리자 모든 종료하기 13 항 관리자 모든 종료하기 14 관리자 모든 종료하기 15 관리자 모든 종료하기 16 관리자 모든 종료하기 17 교리자 모든 종료하기 18 관리자 모든 종료하기 18 관리자 모든 종료하기 19 교리자 모든 종료하기 19 교리자 모든 종료하기 10 교리자 모든 종료하기 10 교리자 모든 종료하기 11 개고 변경하기 12 관리자 모든 종료하기 15 교리자 모든 종료하기 16 교리자 모든 종료하기 17 교리자 모든 종료하기 18 교리자 모든 종료하기 18 교리자 모든 종료하기 18 교리자 모든 종료하기 18 교리자 모든 종료하기 19 교리자 모든 종료하기 19 교리자 모든 종료하기 10 교리자 모든 종료하기 10 교리자 모든 종료하기 10 교리자 모든 종료하기 11 개고 변경하기 20 교리자 모든 종료하기 11 개고 변경하기 20 교리자 모든 종료하기 12 교리자 모든 종료하기 12 교리자 모든 종료하기 15 교리자 모든 종료하기 16 교리자 모든 종료하기 16 교리자 모든 종료하기 17 교리자 모든 종료하기 18 교리자 모든 종료하기 18 교리자 모든 종료하기 19 교리자 모든 종료하기 10 교리자 모든 종료하기 10 교리자 모든 종료하기 10 교리자 모든 종료하기 11 개고 변경하기 20 교리자 모든 종료하기 11 개고 변경하기 20 교리자 모든 종료하기 12 교리자 모든 종료하기 15 교리자 모든 종료하기 16 교리자 모든 종료하기 16 교리자 모든 종료하기 17 교리자 모든 종료하기 18 교리자 모든 종료하기 18 교리자 모든 종료하기 19 교리자 모든 종료하기 10 교리자 모든 종료하기 10 교리자 모든 종료하기 10 교리자 모든 종료하기 10 교리자 모든 종료하기 11 기자 모든 종료하기 12 교리자 모든 종료하기 12 교리자 모든 종료하기 12 교리자 모든 종료하기 15 교리자 모든 종료하기 16 교리자 모든 종료하기 16 교리자 모든 종료하기 17 교리자 모든 종료하기 18 교리자 모든 종료하기 18 교리자 모든 종료하기 18 교리자 모든 종료하기 19 교리자 모든 종료하기 19 교리자 모든 종료하기 19 교리자 모든 종료하기 10 교리자 모든 종료하기 12 교리자 모든 종료하기 12 교리자 모든 종료하기 12 교리자 모든 종료하기 16 교리자 모든 종료하기 16 교리자 모든 종료하기 16 교리자 모든 종료하기 17 교리자 모든 종료하기 18 교리자 모든 종료하기 19 교리자 모든 종료하기 10 교리자 모든 종료하기 12 교리자 모든 종료하기 12 교리자 모든 종료하기 12 교리자 모든 종료하기 16 교리자 모든 종료하기 16 교리자 모든 종료하기 16 교리자 모든 종료하기 17 교리자 모든 종료						<b>∀</b> Χ/`			1	
2 재고 변경하기 11 재고 변경하기 12 관리자 모드 종료하기	.1	관리자 모드 진입하기	<b></b>	10	관리자 모드 진입하기	$^{\prime}$ $^{\prime}$	exitAdmin			sendSaleRequestMessage(itemCode:String, quantity:int): void
3 관리자 모드 종료하기  12 관리자 모드 종료하기  12 관리자 모드 종료하기  13 관리자 모드 종료하기  14 관리자 모드 종료하기  15 관리자 모드 종료하기  16 환리자 모드 종료하기  17 관리자 모드 종료하기  18 ***sendStockRegnonseMessage(itemCode:String, quantity:int): void getStockRegnorseMessage(itemCode:String, quantity:int, verificationCode:String): void sendStockRegnorseMessage(itemCode:String, quantity:int, verificationCode:String): void end(message): void  18 **sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): void end(message): void  19 **sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): void  20 **sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): void  21 **sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): void  22 **sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): void  23 **sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): void  24 **sendPrepaymentInfoMessage(itemCode:String): void  25 **sendPrepaymentInfoMessage(itemCode:String): void  25 **sendPrepaymentInfoMessage(itemCode:String): void  26 **sendPrepaymentInfoMessage(itemCode:String): void  27 **sendPrepaymentInfoMessage(itemCode:String): void  28 **sendPrepaymentInfoMessag		개그 변경하기		11	개그 변경하기	7/	condPronaymontInfo			
### PECIAL PECIAL PROPRENTIES AND PROPRENTIES	),Z	세포 단경에게		11	세포 단경에게	] /	Sendriepaymenumo	_// //////////////////////////////////		
sendStockRequestMessage(itemCode:String, quantity:int): void getStockRequestMessage(itemCode:String, quantity:int, verificationCode:String): voi sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): voi cangeStatus(status): void getResponseMessages(): Vector run(): void isSuccess(): boolean getMessage(): String getResult(): T getPrepaymentInfo(String verificationCode): PrepaymentInfo savePrepaymentInfo(String verificationCode, String itemCode, int quantity): void generateVerificationCode(): String getPrepaymentInfo(String verificationCode): PrepaymentInfo savePrepaymentInfo(String verificationCode): PrepaymentInfo	5.3	관리자 모드 종료하기	<b></b>	12	관리자 모드 종료하기			<i>                 </i>	.//\1	
send/repaymentinfoMessage(itemCode:String, quantity:int, verificationCode:String): void send(message): void run0: void changeStatus(status): void getResponseMessages(): Vector run0: void isSuccess(): boolean getMessage(): String getRessult(): T getPrepaymentinfo(String verificationCode): Prepaymentinfo savePrepaymentinfo(String verificationCode, String itemCode, int quantity): void gentPrepaymentinfo(String verificationCode): Prepaymentinfo save(Prepaymentinfo(String verificationCode): Prepaymentinfo(String verificationCode): Prepaymentinfo(String verificationCode): Prepaymentinfo(String verificationCode): Prepaymentinfo(String verificationCode): Prepaymentinfo(String verificationCode): Prepayme										
send(message): void run(): void changeStatus(status): void getResponseMessages(): Vector run(): void isSuccess(): boolean getMessage(): String getRespult(): T getPrepaymentInfo(String verificationCode): PrepaymentInfo sawePrepaymentInfo(String verificationCode, int quantity): void generateVerificationCode(): String getPrepaymentInfo(String verificationCode): PrepaymentInfo save(PrepaymentInfo(String verificationCode): PrepaymentInfo save(PrepaymentInfo(String verificationCode): VerpaymentInfo save(PrepaymentInfo(String verificationCode): String getPrepaymentInfo(String verificationCode): String getPrepaymentInfo(String verificationCode): String								// <b>/</b> /////////////////////////////////		
run(): void  changeStatus(status): void  getResponseMessages(): Vector  run(): void  isuccess(): boolean  getMessage(): String  getRespult(): T  getPrepaymentInfo(String verifiationCode): PrepaymentInfo  savePrepaymentInfo(String verificationCode, String itemCode, int quantity): void  generateVerificationCode(): String  getPrepaymentInfo(String verificationCode): PrepaymentInfo  save(PrepaymentInfo(String verificationCode): PrepaymentInfo  save(PrepaymentInfo(String verificationCode): PrepaymentInfo  save(PrepaymentInfo): void  isValid(): boolean  gettemCode(): String								\\ <b>\</b> \\\\		sendPrepaymentInfoMessage(itemCode:String, quantity:int, verificationCode:String): 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(String verificationCode): PrepaymentInfo save(PrepaymentInfo(String verificationCode): PrepaymentInfo save(PrepaymentInfo(String verificationCode): PrepaymentInfo save(PrepaymentInfo(String verificationCode): String getHemCode(): String								// /// //	<b>\</b> \\ ¶	send(message): void
getResponseMessages(): Vector run(): void isSuccess(): boolean getMessage(): String getResult(): T getPrepaymentInfo(String verificationCode): PrepaymentInfo savePrepaymentInfo(String verificationCode, String itemCode, int quantity): void generateVerificationCode(): String getPrepaymentInfo(String verificationCode): PrepaymentInfo save(PrepaymentInfo(String verificationCode): String getPrepaymentInfo(String verificationCode): String getPrepaymentInfo(String verificationCode): String getPrepaymentInfo(String verificationCode): String									1	
run(): void  isSuccess(): boolean  getMessage(): String  getResult(): T  getPrepaymentInfo(String verificationCode): PrepaymentInfo  savePrepaymentInfo(String verificationCode, String itemCode, int quantity): void  generateVerificationCode(): String  getPrepaymentInfo(String verificationCode): PrepaymentInfo  save(PrepaymentInfo(String verificationCode): Void  isValid(): boolean  getItemCode(): String								\\ <b>\</b> \\	1	changeStatus(status): 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								\\ \\	\ ₹	getResponseMessages(): Vector
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								\\ \\	\ [	run(): void
getResult(): T  getPrepaymentInfo(String verifiationCode): PrepaymentInfo  savePrepaymentInfo(String verificationCode, int quantity): void  generateVerificationCode(): String  getPrepaymentInfo(String verificationCode): PrepaymentInfo  save(PrepaymentInfo prepaymentInfo): void  isValid(): boolean  getItemCode(): String								\ <b>\                                  </b>	<u> </u>	isSuccess(): boolean
getPrepaymentInfo(String verificationCode): PrepaymentInfo savePrepaymentInfo(String verificationCode, String itemCode, int quantity): void generateVerificationCode(): String getPrepaymentInfo(String verificationCode): PrepaymentInfo save(PrepaymentInfo prepaymentInfo): void isValid(): boolean getItemCode(): String								\ \\	<b>\\\</b> [	getMessage(): String
savePrepaymentInfo(String verificationCode, String itemCode, int quantity): void  generateVerificationCode(): String  getPrepaymentInfo(String verificationCode): PrepaymentInfo  save(PrepaymentInfo prepaymentInfo): void  isValid(): boolean  getItemCode(): String								/ /	<b>\</b> \\\[	getResult(): T
generateVerificationCode(): String  getPrepaymentInfo(String verificationCode): PrepaymentInfo  save(PrepaymentInfo prepaymentInfo): void  isValid(): boolean  getItemCode(): String								\	<b>///</b>	getPrepaymentInfo(String verifiationCode): PrepaymentInfo
getPrepaymentInfo(String verificationCode): PrepaymentInfo save(PrepaymentInfo prepaymentInfo): void isValid(): boolean getItemCode(): String								\		savePrepaymentInfo(String verificationCode, String itemCode, int quantity): void
save(PrepaymentInfo): void isValid(): boolean getItemCode(): String								`	/ /	generateVerificationCode(): String
isValid(): boolean getItemCode(): String									/\	getPrepaymentInfo(String verificationCode): PrepaymentInfo
getItemCode(): String									¥	save(PrepaymentInfo prepaymentInfo): void
										isValid(): boolean
getQuantity(): int									Ī	getItemCode(): String
									Ī	getQuantity(): int

Class
Controller
CardService
CardCompany
ItemService
ItemRepository
NetworkService
Sender
Receiver
Receive Message Handler
Response
PrepaymentService
PrepaymentRepository
PrepaymentInfo

