

OOPT 2140 Construction

201612020 조석현 201811300 하승래
202010375 김만재 201811220 정유빈

Contents

- 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

2141. Design Real Use Case

Use Case	1. 상품 선택		
Actor	User	Typical courses of event	1. (User) 사용자가 A of Window-0를 클릭하여 구매하려는 상품을 선택 2. (User) 상품의 개수를 A of Window-1_1을 이용하여 선택하고 B of Window-1_1를 클릭 3. (System) 상품 재고가 있는 것을 확인 4. (System) 결제 금액을 출력한 후 결제 기능 수행
Purpose	총 20가지 상품 중 구매하고자 하는 상품을 입력(선택)한다.		
Type	Evident	Alternative courses of event	2-a. (System) 상품 재고가 없는 것을 확인 2-b. (System) 타 자판기 재고 및 위치 안내 기능 수행
Cross Reference	Functions: R 1.1		
Pre-Requisites	N/A	Exceptional courses of event	N/A
UI Widgets	Window-0, Window-1_1		

2141. Design Real Use Case

Use Case	2. 타 자판기 재고 및 위치 안내	
Actor	System	Typical courses of event 1. (System) 타 자판기에 재고 확인 요청 전송 2. (Other Dvm) 재고 확인 응답을 전송 3. (System) 재고 확인 응답 수신 4. (System) 재고가 있는 가장 가까운 자판기의 위치를 Window-1_4를 통해 전달 5. (User) 사용자가 선결제 여부를 A, B of Window-1_3을 통해 선택 6. (System) 선결제를 선택할 경우, 결제 금액을 메시지로 출력한 후, 선결제 기능 수행
Purpose	사용자가 입력한 상품의 재고 요청 메시지를 타 자판기에게 보내고, 사용자에게 가장 가까운 자판기의 위치를 안내한다.	
Type	Evident	
Cross Reference	Functions: R 2.3, R 3.1	Alternative courses of event 5-a.(System) 선결제를 선택하지 않을 경우, Window-1_3을 닫고 Window-0 활성화
Pre-Requisites	선택 과정에서 사용자가 재고가 없는 상품을 선택하였을 경우	Exceptional courses of event N/A
UI Widgets	Window-0, Window-1_3, Window-1_4	

2141. Design Real Use Case

Use Case	3. 결제		
Actor	User	Typical courses of event	<ol style="list-style-type: none"> 1. (User) A of Window-1_2를 통해 사용자가 카드 번호를 입력 2. (System) 카드의 유효함을 확인 3. (System) 카드의 잔액이 충분함을 확인 4. (System) 카드의 잔액에서 결제 금액 차감 5. (System) 재고 차감 6. (System) 판매 실적 업데이트 7. (System) 배출될 상품을 메시지로 알림
Purpose	사용자로부터 카드를 입력 받아 유효성과 잔액을 확인한 후 구매할 상품에 대해 결제를 진행한다.	Alternative courses of event	N/A
Type	Evident	Exceptional courses of event	<ol style="list-style-type: none"> 1-a. (System) 5초 동안 카드 번호가 입력되지 않을 경우 Window-0를 활성화 2-a. (System) 카드의 유효하지 않음을 확인 2-b. (System) 사용자에게 결제 오류를 메시지로 알림 2-c. (System) Window-0를 활성화 3-a. (System) 카드의 잔액이 충분하지 않음을 확인 3-b. (System) 사용자에게 결제 오류를 메시지로 알림 3-c. (System) Window-0를 활성화 5-a. (System) 재고 차감이 불가능함을 확인 5-b. (System) 결제 금액 환불 5-c. (System) 결제가 취소되었음을 메시지로 알림 5-d. (System) Window-0를 활성화
Cross Reference	Functions: R 2.1		
Pre-Requisites	사용자가 현재 재고가 있는 상품을 선택함		
UI Widgets	Window-0, Window-1_2, Window-1_4		

2141. Design Real Use Case

Use Case	4. 선결제		
Actor	User	Typical courses of event	1. (User) A of Window-1_2를 통해 사용자가 카드 번호를 입력 2. (System) 카드의 유효함을 확인 3. (System) 카드의 잔액이 충분함을 확인 4. (System) 카드의 잔액에서 결제 금액 차감 5. (System) 인증 코드 발급 기능 수행
Purpose	사용자로부터 카드를 입력 받아 유효성과 잔액을 확인한 후 구매할 상품에 대해 선결제를 진행한다.		
Type	Evident	Alternative courses of event	N/A
Cross Reference	Functions: R 2.1, R 2.4, R 3.1	Exceptional courses of event	1-a. (System) 5초 동안 카드 번호가 입력되지 않을 경우 Window-1_2를 닫고 Window-0를 활성화
Pre-Requisites	사용자가 현재 자판기에는 재고가 없지만 다른 자판기에 재고가 있는 상품을 선택함		2-a. (System) 카드의 유효하지 않음을 확인 2-b. (System) 사용자에게 결제 오류를 메시지로 알림 2-c. (System) Window-0를 활성화
UI Widgets	Window-1_2		3-a. (System) 카드의 잔액이 충분하지 않음을 확인 3-b. (System) 사용자에게 결제 오류를 메시지로 알림 3-c. (System) Window-0를 활성화

2141. Design Real Use Case

Use Case	5. 인증 코드 발급		
Actor	System	Typical courses of event	1. (System) 인증코드를 생성 2. (System) 대상 자판기에 선결제 확인 메시지를 전송 3. (Other DVM) 음료 판매 응답을 전송 4. (System) 음료 판매 응답을 수신 5. (System) 사용자에게 인증 코드를 메시지로 전달
Purpose	인증 코드를 발급하고 다른 자판기에게 선결제된 상품에 대한 정보를 전송한 후 사용자에게 대상 자판기의 위치를 안내한다.		
Type	Hidden	Alternative courses of event	N/A
Cross Reference	Functions: R 2.4	Exceptional courses of event	4-a. (System) 수신한 음료 판매 응답 메시지가 유효하지 않음 4-b. (System) 결제 금액 환불 4-c. (System) 결제가 취소되었음을 메시지로 알림 4-d. (System) Window-0 활성화 4-a. (System) 5초 동안 메시지가 오지 않을 경우 결제 금액 환불 4-b. (System) 결제가 취소되었음을 메시지로 알림 4-c. (System) Window-0 활성화
Pre-Requisites	사용자가 선결제를 진행하였다		
UI Widgets	Window-0		

2141. Design Real Use Case

Use Case	6. 재고 정보 송수신		
Actor	other DVM	Typical courses of event	1. (Other DVM) 상품에 대한 재고 확인 요청 전송 2. (System) 재고 확인 요청 수신 3. (System) 프로토콜을 해석하여 재고 확인 응답을 전송
Purpose	재고 데이터를 요청한 DVM에게 현재 재고 데이터를 송신한다.		
Type	Evident	Alternative courses of event	N/A
Cross Reference	Functions: R 2.3, R 3.1	Exceptional courses of event	3-a. (System) 프로토콜을 해석하지 못하여 요청을 무시
Pre-Requisites	Other DVM으로부터 "재고 확인 요청" 메시지를 받아야 한다.		
UI Widgets			

2141. Design Real Use Case

Use Case	7. 판매 처리		
Actor	other DVM	Typical courses of event	1. (Other DVM) 다른 자판기에서 선결제 확인 메시지를 전송 2. (System) 재고가 있을 경우, 재고의 수량을 낮추고 대상 자판기에게 음료 판매 응답 메시지를 전송 3. (System) 다른 모든 자판기에게 음료 판매 확인 메시지 전송 4. (System) 받은 상품의 정보와 인증코드를 내부에 저장
Purpose	다른 DVM에서 선결제한 상품에 대해 판매 처리를 한다.		
Type	Evident	Alternative courses of event	N/A
Cross Reference	Functions: R 3.2	Exceptional courses of event	2-a. (System) 재고가 없을 경우, 음료 판매 확인 메시지에 음료 개수를 0으로 하여 메시지 전송
Pre-Requisites	Other DVM으로부터 "선결제 확인" 메시지를 받아야 한다.		
UI Widgets			

2141. Design Real Use Case

Use Case	8. 인증 코드 입력		
Actor	User	Typical courses of event	1. (User) 사용자가 인증코드를 A of Window-2_1을 통해 입력 2. (System) 내부에 저장된 인증코드와 사용자가 입력한 인증코드를 비교 3. (System) 인증코드가 유효할 경우, 판매 실적 업데이트 4. (System) 상품을 제공하고 상품 선택화면인 Window-0 활성화
Purpose	사용자로부터 인증코드를 입력 받고, 인증코드가 유효할 경우 상품을 배출한다.		
Type	Evident	Alternative courses of event	N/A
Cross Reference	Functions: R 1.3	Exceptional courses of event	3-a. (System) 인증코드가 유효하지 않을 경우, 오류 메시지를 출력 3-b. (System) Window-0 활성화
Pre-Requisites	N/A		
UI Widgets	Window-2_1, Window-0		

2141. Design Real Use Case

Use Case	9. 관리자 메뉴		
Actor	Admin	Typical courses of event	1. (Admin) A of Window-3_0 을 통해 비밀번호를 입력 2. (System) 비밀번호가 유효할 경우, Window-3 표시
Purpose	비밀번호 입력을 통해 관리자 메뉴에 접근할 수 있다.		
Type	Evident	Alternative courses of event	N/A
Cross Reference	Functions: R 4.1, R 4.2, R 4.3, R.4.4	Exceptional courses of event	2-a. (System) 비밀번호가 유효하지 않을 경우, Window-0 활성화
Pre-Requisites	N/A		
UI Widgets	Window-0, Window-3_0, Window-3		

2141. Design Real Use Case

Use Case	10. 재고 관리		
Actor	Admin	Typical courses of event	1. (Admin) 보충할 상품과 보충 재고수를 A, B of Window-3_1을 통해 선택/입력 2. (System) 관리자가 선택한 상품을 입력한 개수만큼 추가 3. (System) Window-3_1을 닫고 Window_3을 활성화
Purpose	관리자가 DVM에서 판매중인 상품의 재고를 관리할 수 있다.		
Type	Evident	Alternative courses of event	N/A
Cross Reference	Functions: R 4.1, R.4.2	Exceptional courses of event	2-a. (System) 보충 개수가 범위를 초과한 경우, 오류를 메시지로 알림 2-b. (System) Window-3_1을 닫고 Window_3을 활성화
Pre-Requisites	관리자 메뉴에 진입한 상태이어야 한다.		
UI Widgets	Window-3, Window-3_1		

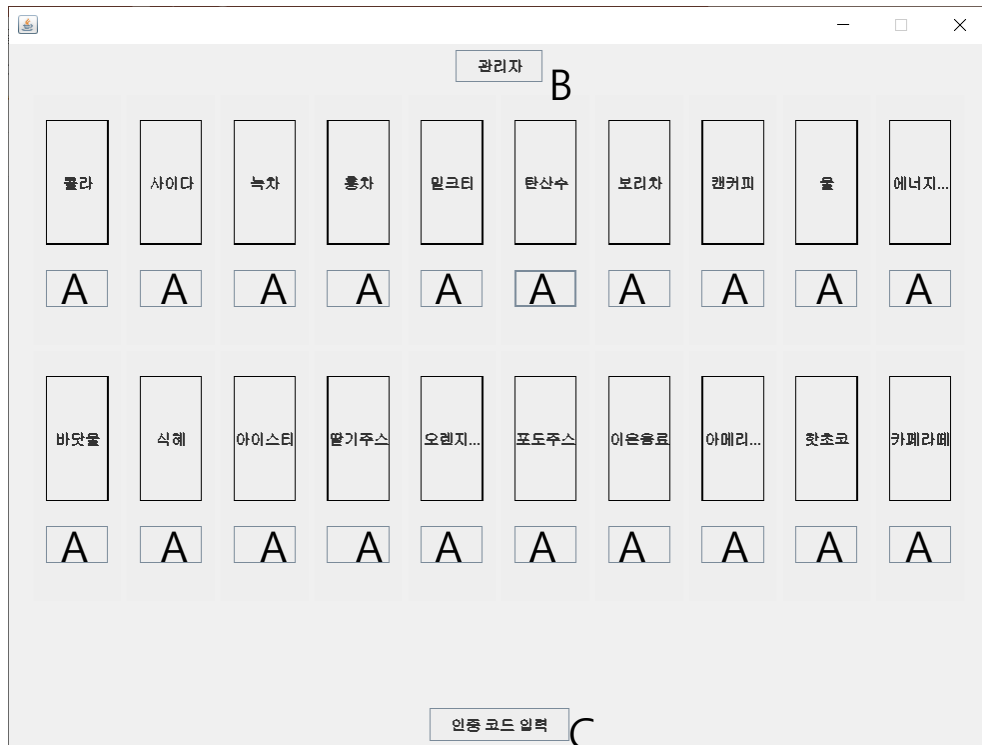
2141. Design Real Use Case

Use Case	11. 위치 수정		
Actor	Admin	Typical courses of event	1. (Admin) A, B of Window-3_2를 이용해 자판기의 x, y 좌표를 입력 2. (System) 관리자가 유효한 좌표를 입력한 경우, 입력한 좌표로 좌표를 수정 3. (System) 사용자에게 완료 알림 후 Window-3_2를 닫고 Window-3 활성화
Purpose	관리자가 DVM의 좌표를 수정할 수 있다.		
Type	Evident	Alternative courses of event	N/A
Cross Reference	Functions: R 4.1, R.4.3	Exceptional courses of event	2-a. (System) 관리자가 유효하지 않은 좌표를 입력한 경우, 오류를 출력 2-b. Window-3_2를 닫고 Window-3 활성화
Pre-Requisites	관리자 메뉴에 진입한 상태이어야 한다.		
UI Widgets	Window-3, Window-3_2		

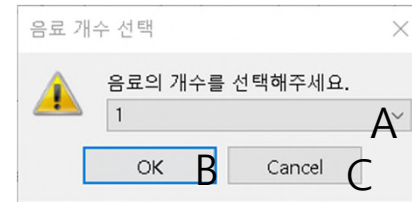
2141. Design Real Use Case

Use Case	12. 판매 실적 합산		
Actor	Admin	Typical courses of event	1. (Admin) 판매 실적을 요청 2. (System) 판매 실적을 Window-3_3에 출력하여 전달
Purpose	판매 실적을 관리자에게 보여준다.		
Type	Evident	Alternative courses of event	N/A
Cross Reference	Functions: R 4.1, R.4.4		
Pre-Requisites	관리자 메뉴에 진입한 상태이어야 한다.	Exceptional courses of event	N/A
UI Widgets	Window-3, Window-3_3		

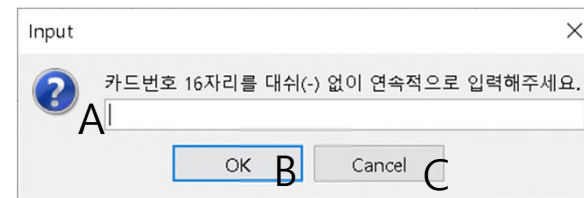
2142. Define Reports, UI and Storyboards



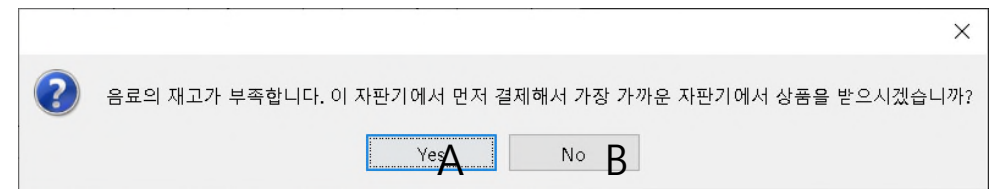
<<Window-0>>



<<Window-1_1>>

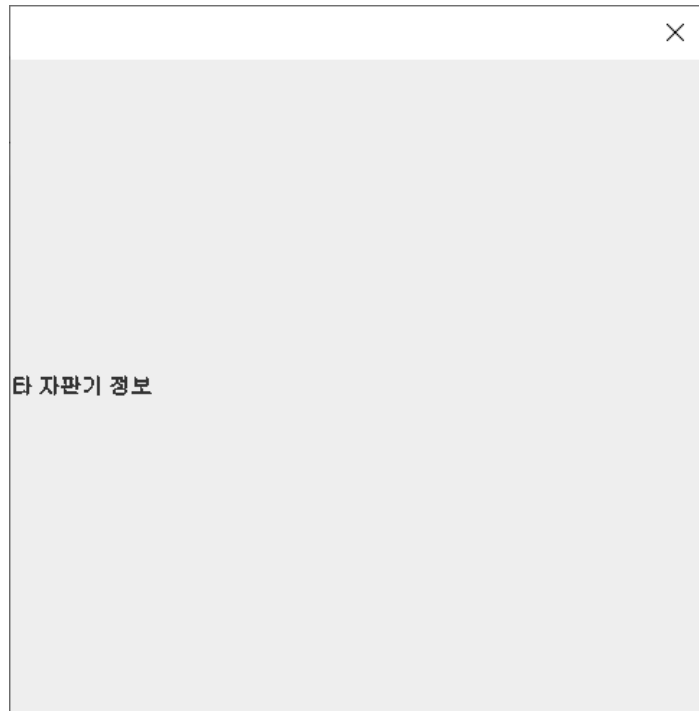


<<Window-1_2>>

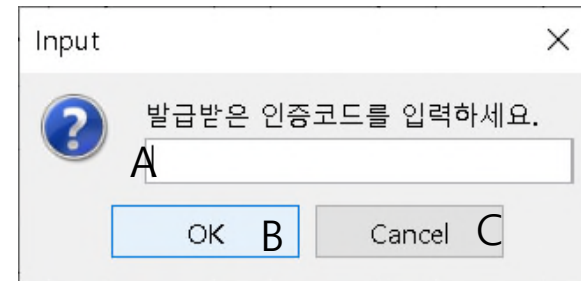


<<Window-1_3>>

2142. Define Reports, UI and Storyboards : Dialog-1

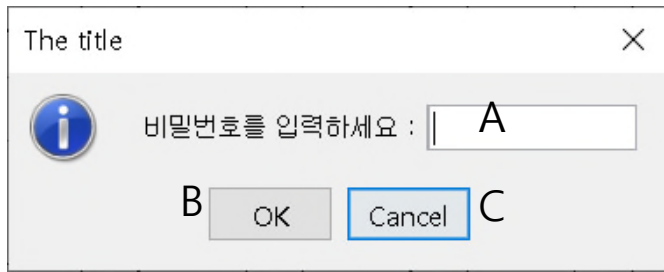


<<Window-1_4>>

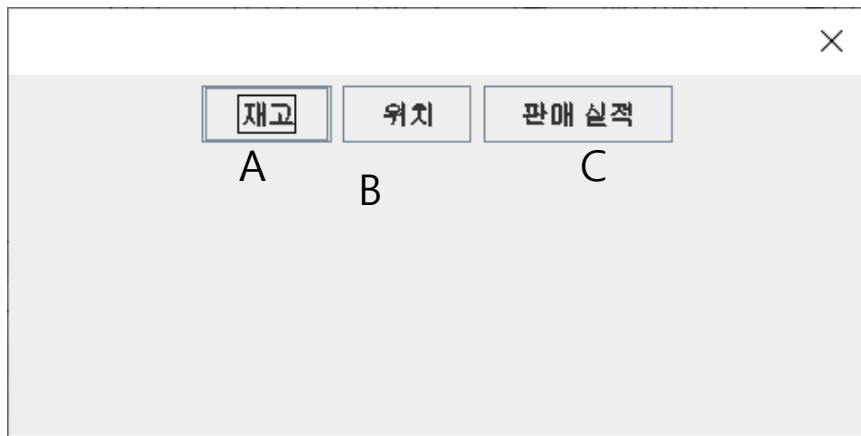


<<Window-2_1>>

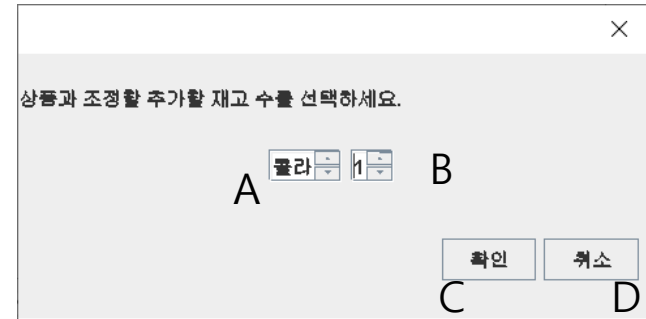
2142. Define Reports, UI and Storyboards : Dialog-2



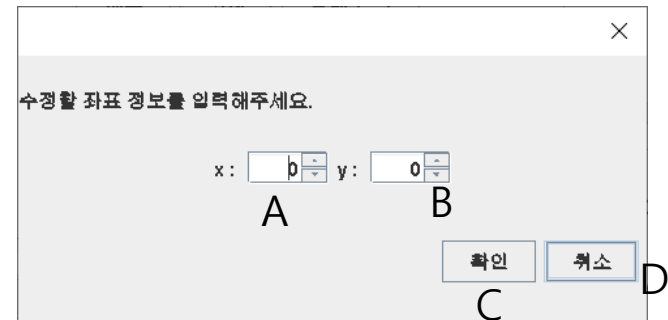
<<Window-3_0>>



<<Window-3>>

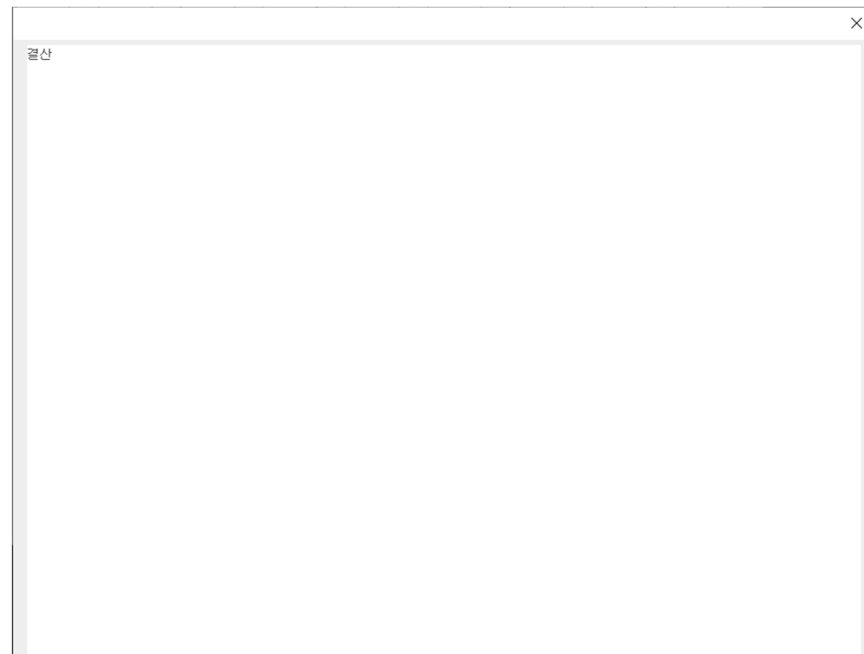


<<Window-3_1>>



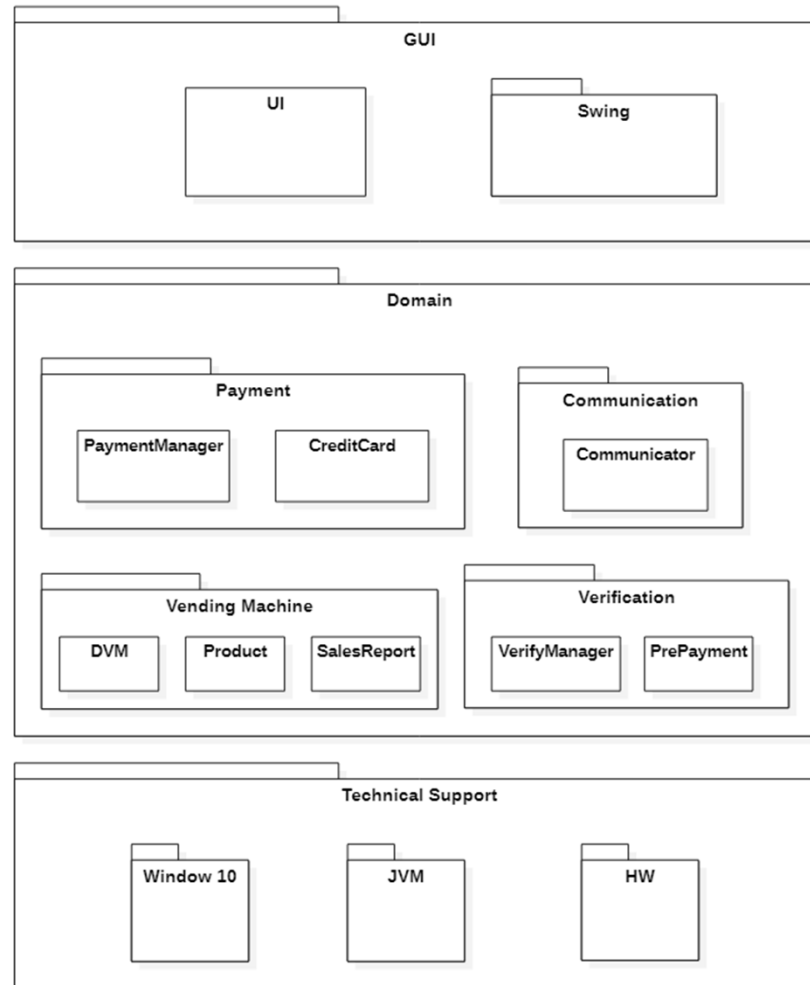
<<Window-3_2>>

2142. Define Reports, UI and Storyboards



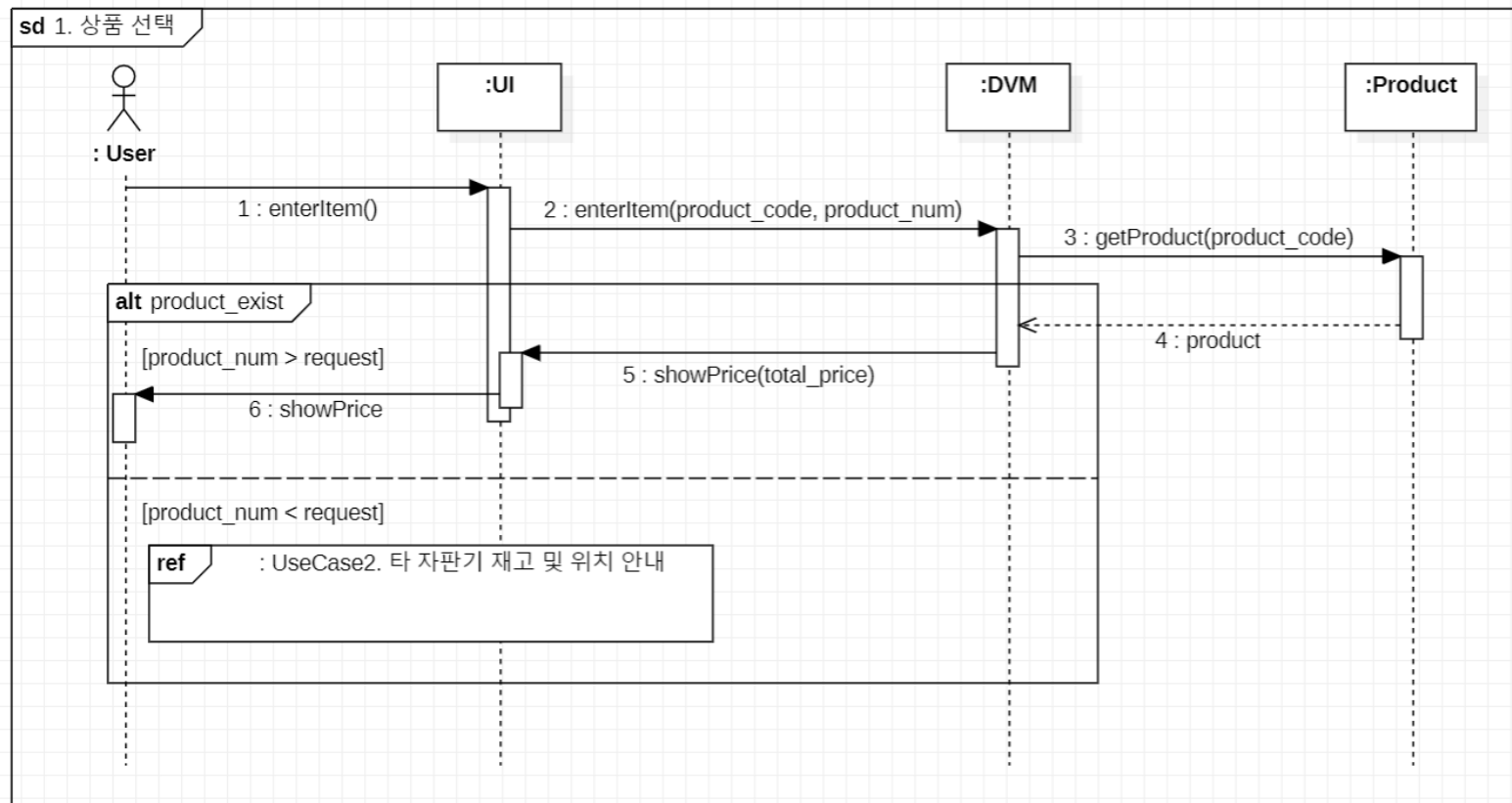
<<Window-3_3>>

2143. Refine System Architecture



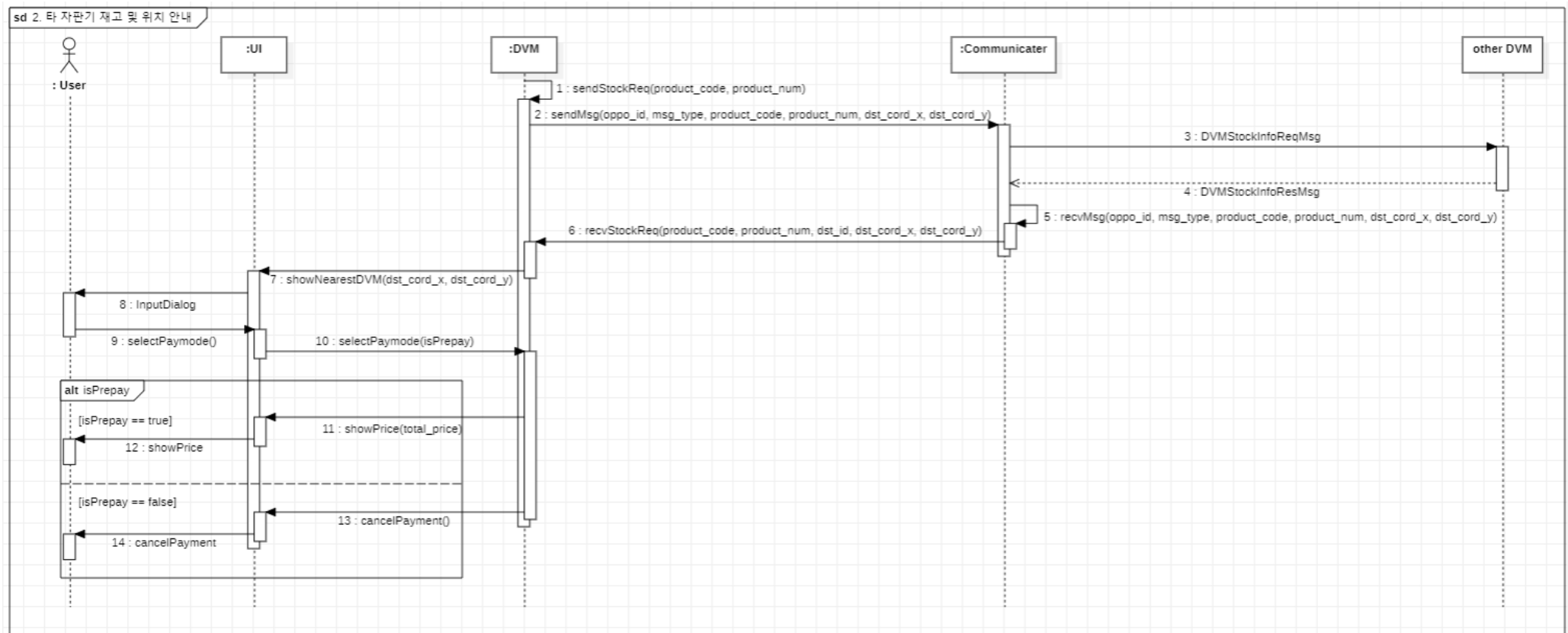
2144. Define Interaction Diagrams

UseCase1. 상품 선택



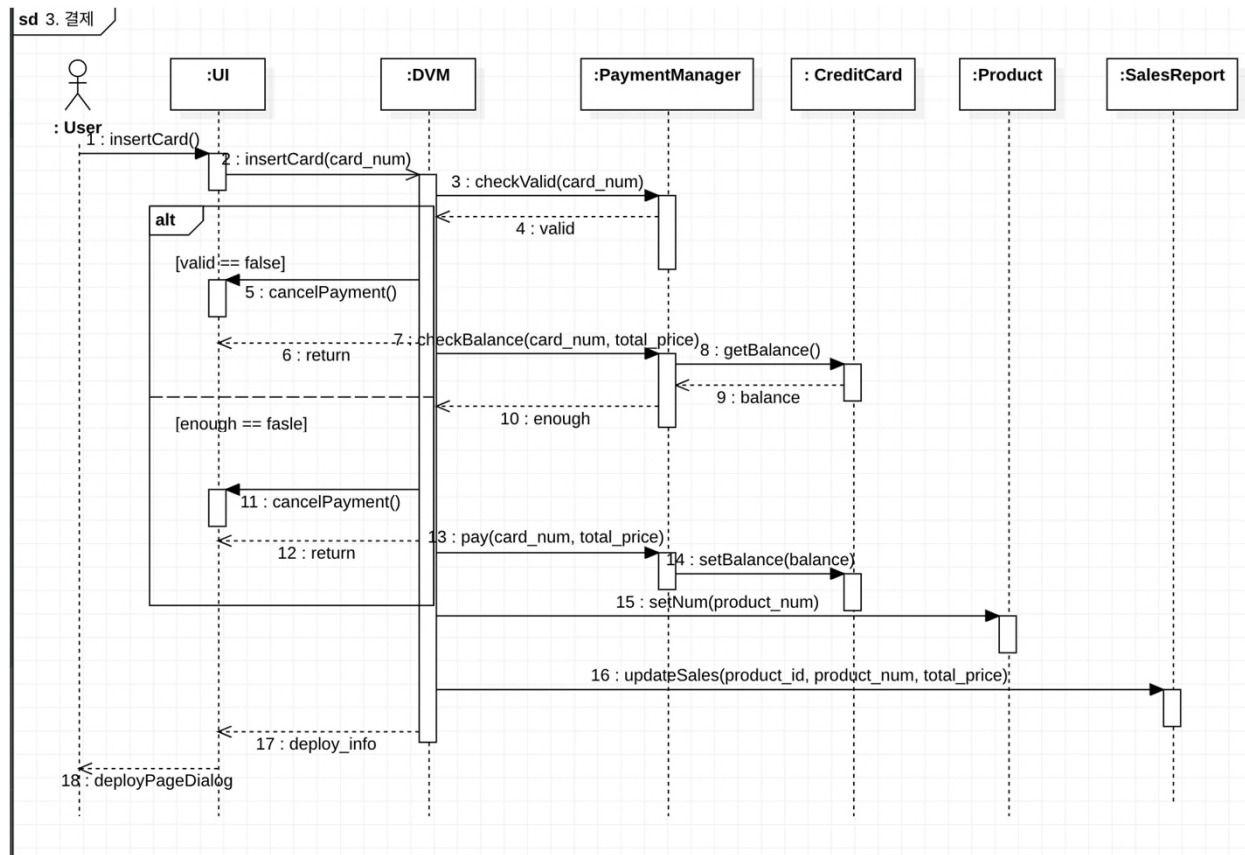
2144. Define Interaction Diagrams

UseCase2. 타 자판기 재고 및 위치 안내



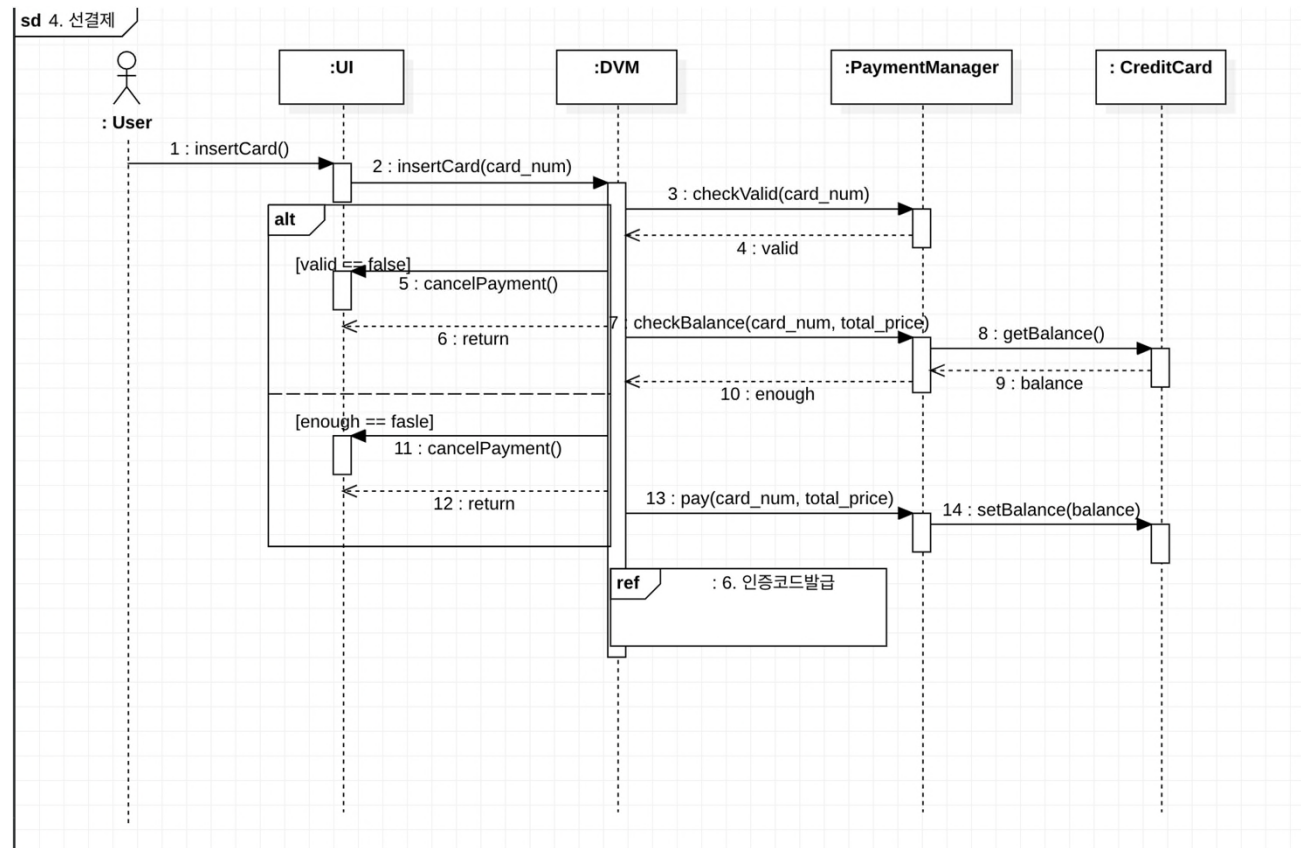
2144. Define Interaction Diagrams

UseCase3. 결제



2144. Define Interaction Diagrams

UseCase4. 선결제



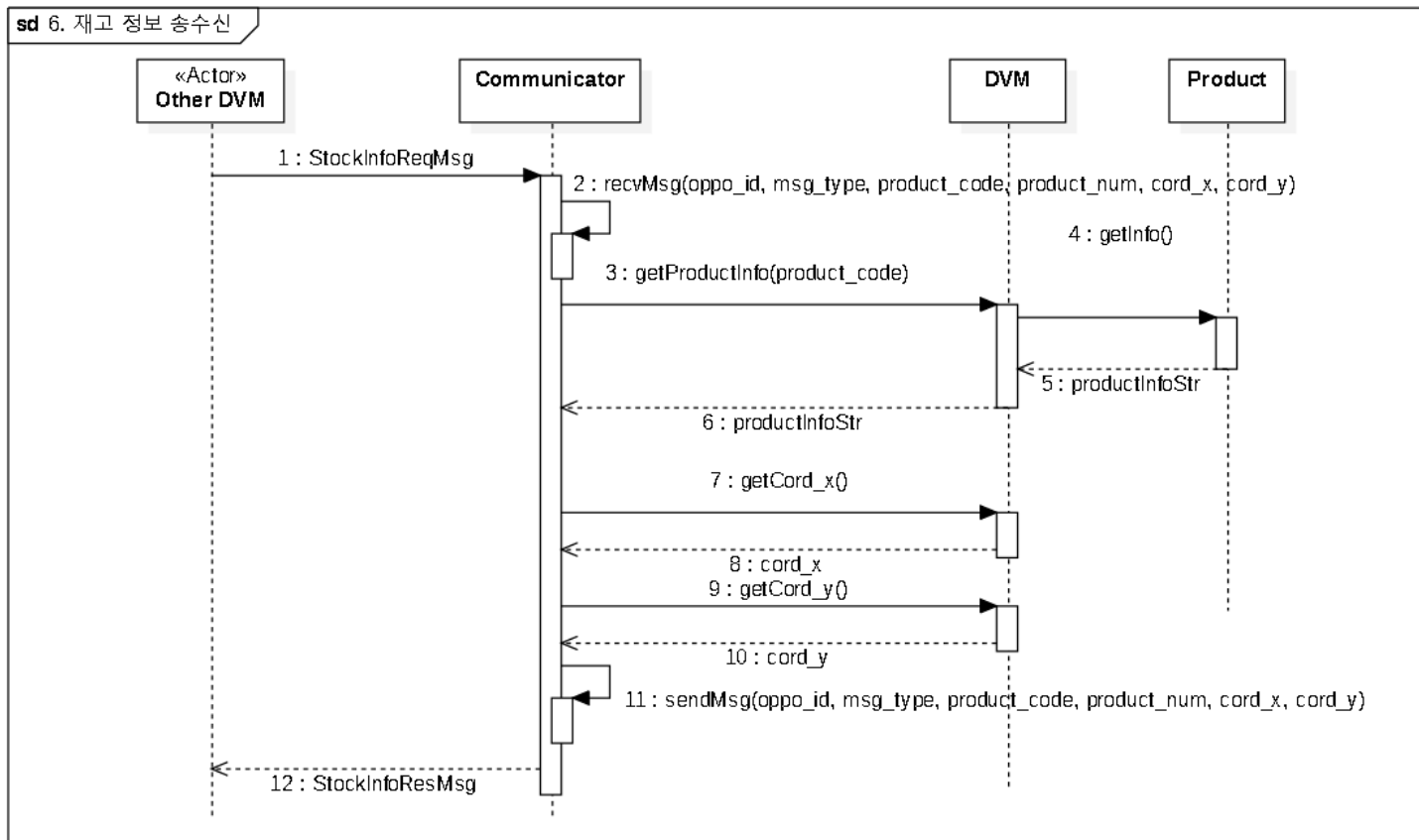
2144. Define Interaction Diagrams

UseCase5. 인증 코드 발급



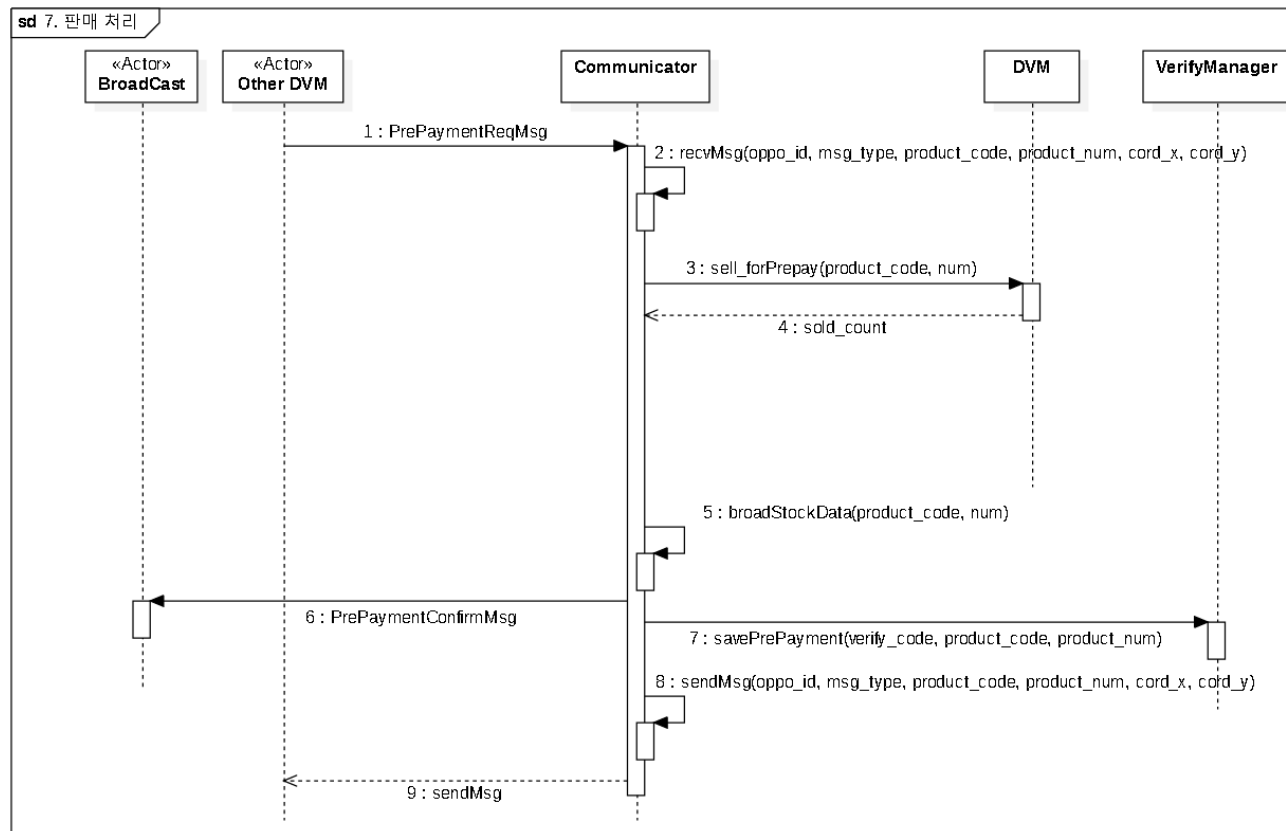
2144. Define Interaction Diagrams

UseCase6. 재고 정보 송수신



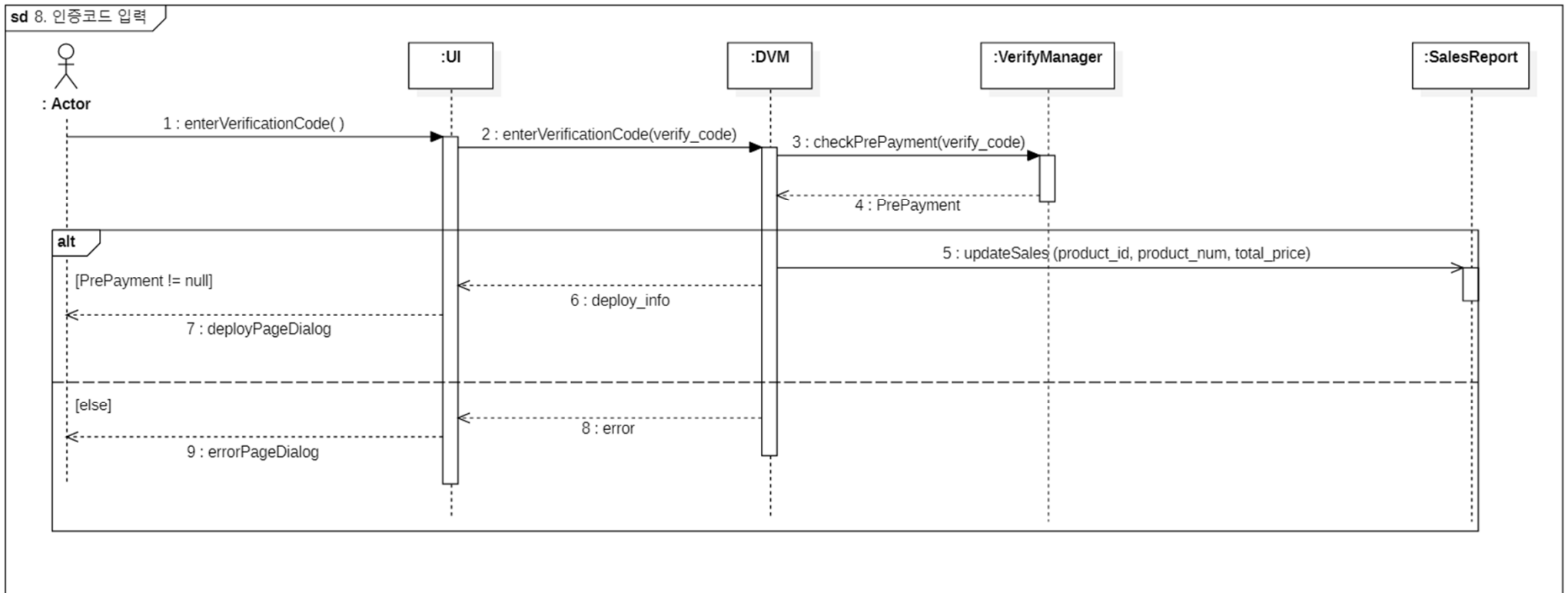
2144. Define Interaction Diagrams

UseCase7. 판매 처리



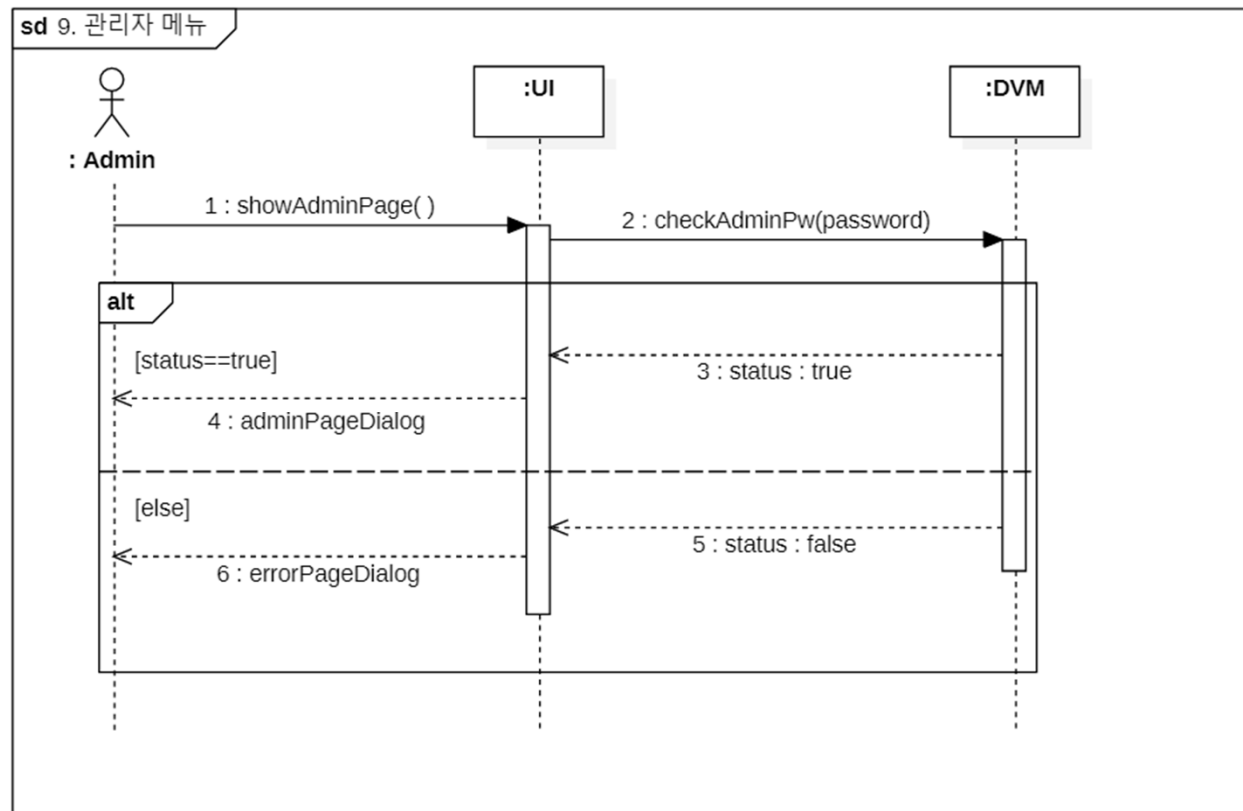
2144. Define Interaction Diagrams

UseCase8. 인증 코드 입력



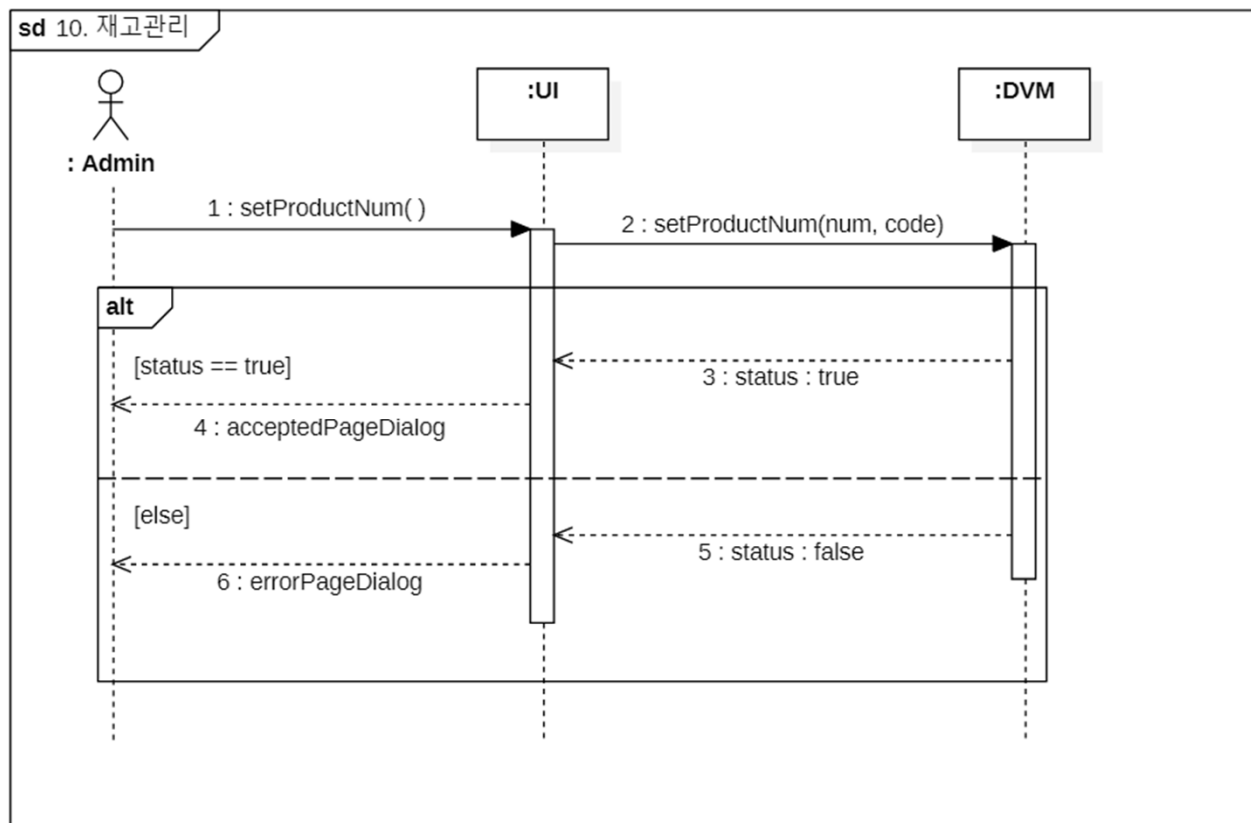
2144. Define Interaction Diagrams

UseCase9. 관리자 메뉴



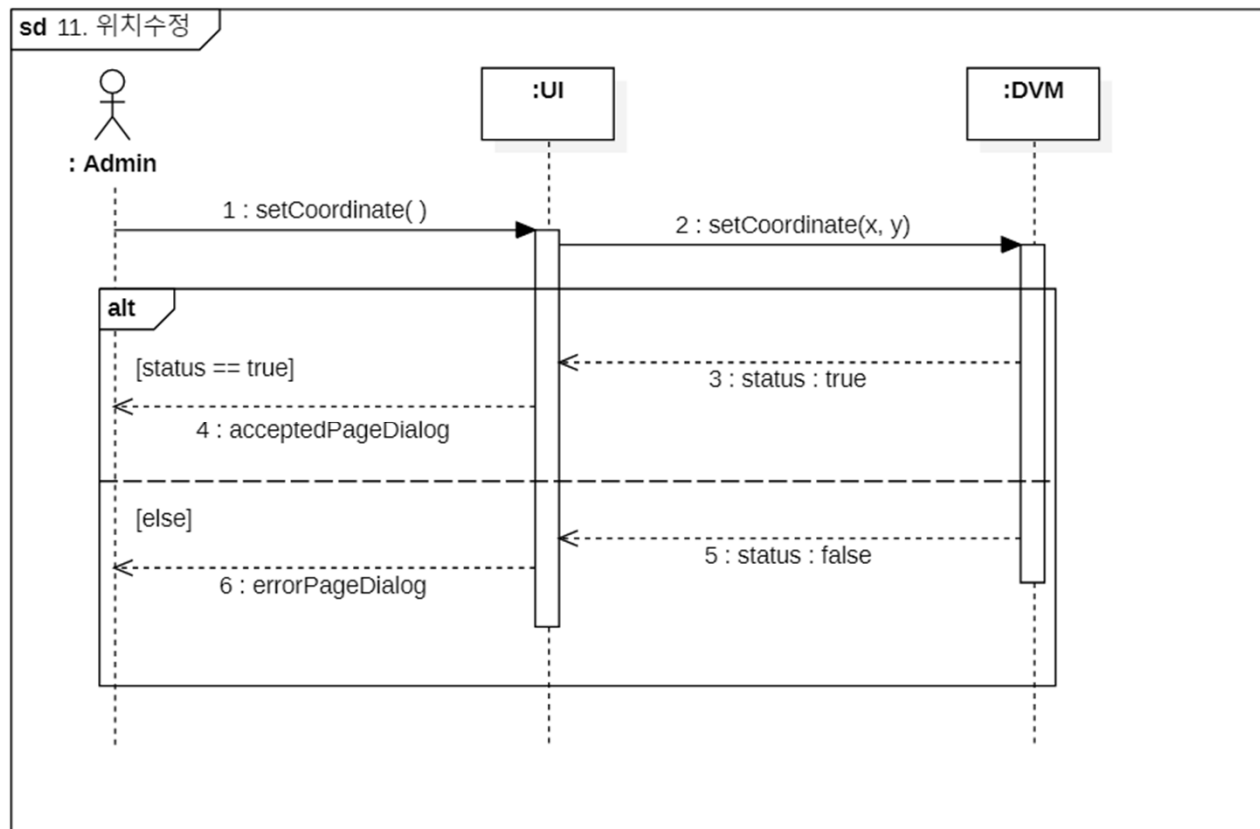
2144. Define Interaction Diagrams

UseCase10. 재고 관리



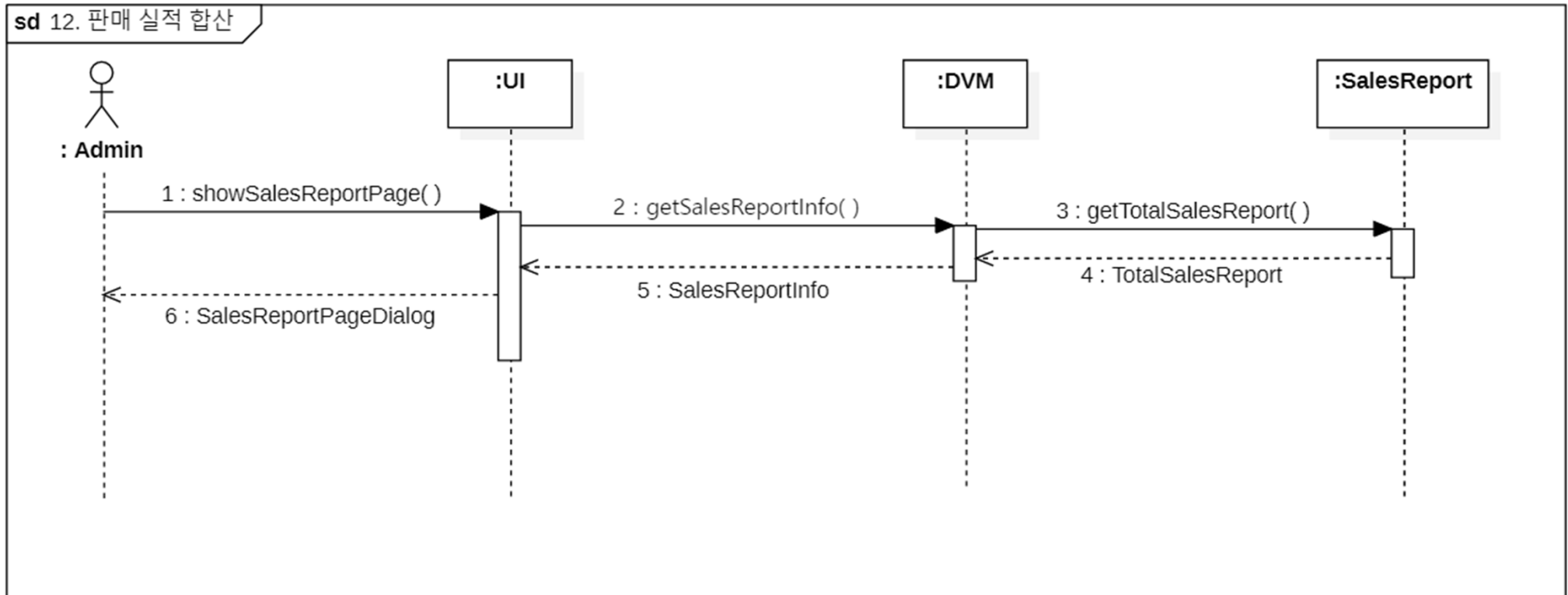
2144. Define Interaction Diagrams

UseCase11. 위치 수정

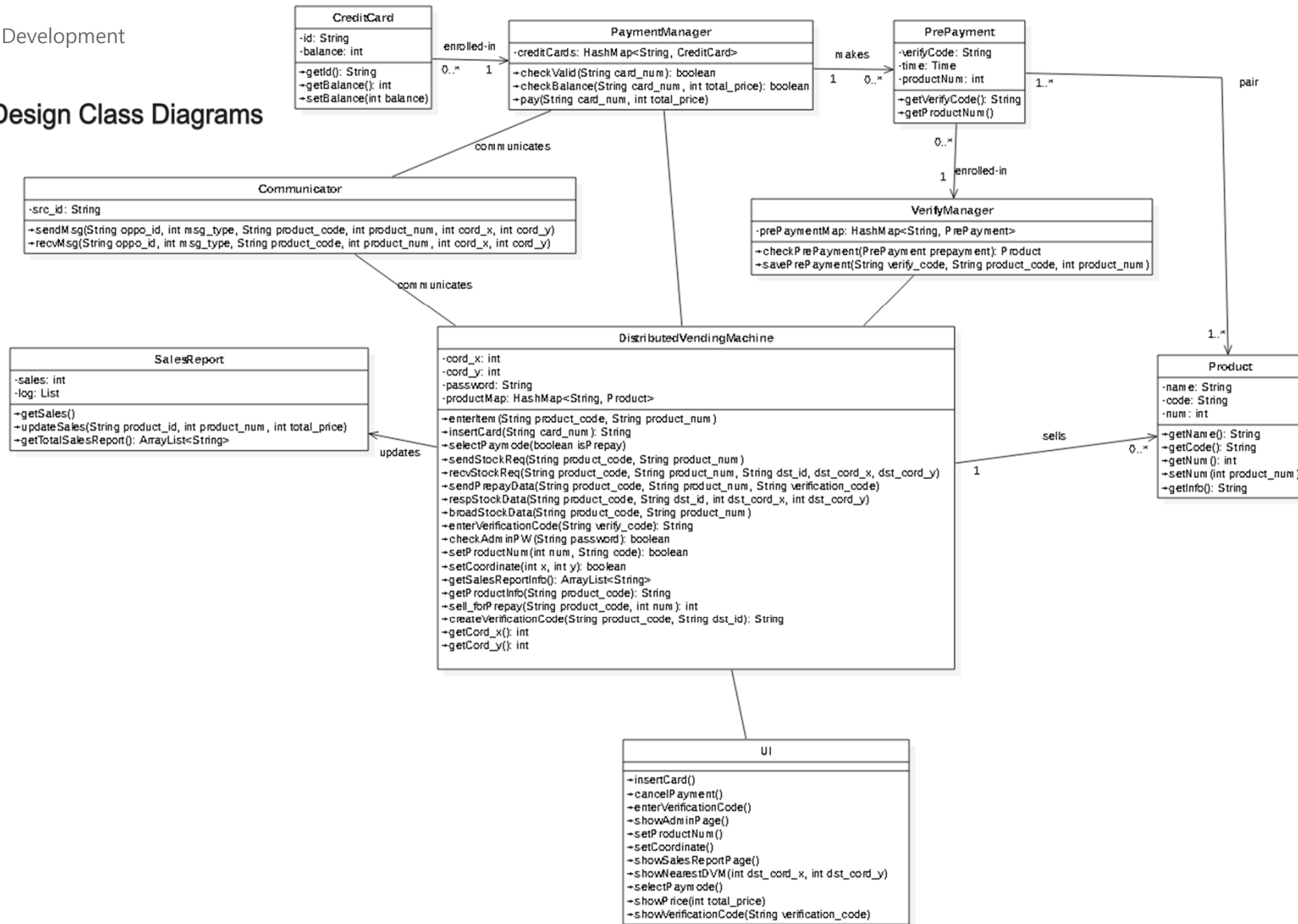


2144. Define Interaction Diagrams

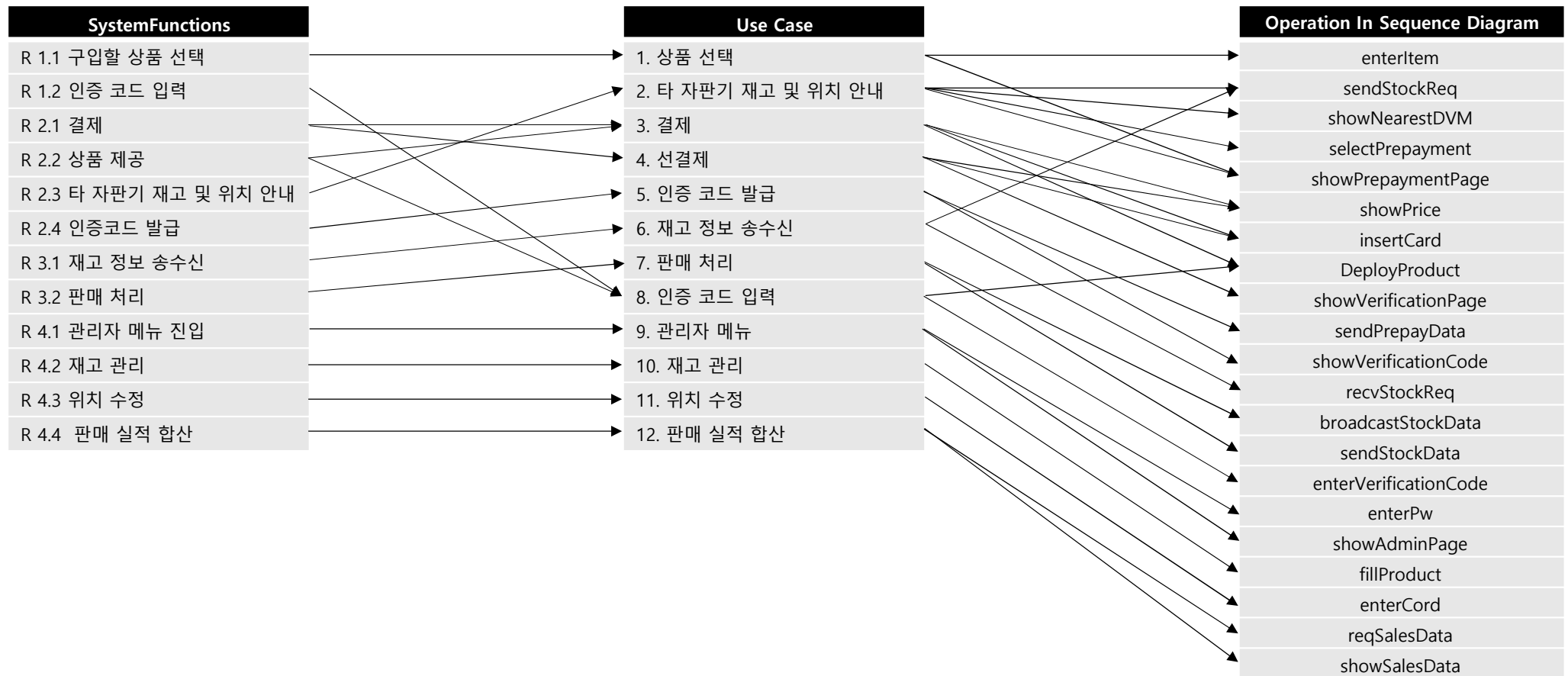
UseCase12. 판매 실적 합산



2145. Define Design Class Diagrams



2146. Design Traceability Analysis



2146. Design Traceability Analysis

Operation In Sequence Diagram

enterItem
sendStockReq
showNearestDVM
selectPrepayment
showPrepaymentPage
showPrice
insertCard
DeployProduct
showVerificationPage
sendPrepayData
showVerificationCode
recvStockReq
broadcastStockData
sendStockData
enterVerificationCode
enterPw
showAdminPage
fillProduct
enterCord
reqSalesData
showSalesData

Operation in interaction diagram
enterItem()
enterItem(product_code, product_num)
getProduct(product_code)
showPrice()
sendStockReq(product_code, product_num)
sendMsg(oppo_id, msg_type, product_code, product_num, dst_cord_x, dst_cord_y)
recvMsg(oppo_id, msg_type, product_code, product_num, dst_cord_x, dst_cord_y)
recvStockReq(product_code, product_num, dst_id, dst_cord_x, dst_cord_y)
showNearestDVM(dst_id)
selectPaymode()
selectPaymode(sPrepay)
showPrice(total_price)
insertCard()
insertCard(card_num)
checkValid(card_num)
cancelPayment()
checkBalance(card_num, total_price)
getBalance()
pay(card_num, total_price)
setBalance(balance)
setNum(product_num)
updateSales(product_id, product_num, total_price)
createVerificationCode(product_code, dst_id)
sendPrepayData(product_code, product_num, verification_code)
respStockData(product_code, dst_id, dst_cord_x, dst_cord_y)
showVerificationCode(verification_code)
broadStockData(product_code, product_num)
getProductInfo(product_code)
getCoordinate()
sell_forPrepay(product_code, num)
savePrePayment(verify_code, product, product_num)
enterVerificationCode()
enterVerificationCode(verify_code)
checkPrePayment(verify_code)
showAdminPage()
checkAdminPw(password)
setProductNum()
setProductNum(num, code)
setCoordinate()
setCoordinate(x, y)
showSalesReportPage()
getSalesReportInfo()
getTotalSalesReport()

Method	Class
enterItem(String product_code, String product_num)	DistributedVendingMachine
showPrice()	
insertCard(String card_num)	
selectPrepayment()	
sendStockReq(String product_code, String product_num)	
recvStockReq(String product_code, String product_num, String dst_id, dst_cord_x, dst_cord_y)	
sendPrepayData(String product_code, String product_num, String verification_code)	
respStockData(product_code, dst_id, dst_cord_x, dst_cord_y)	
broadStockData(product_code, iproduct_num)	
enterVerificationCode(Prepayment)	
checkAdminPW(password)	
setProduct(num, code)	
setCoordinate(x, y)	
getSalesReport()	
getProduct(productCode)	
sell(product, num)	
createVerificationCode(product_code, dst_id)	
getName()	
getCode()	
getNum()	PaymentManager
setNum(product_num)	
checkValid(card_num)	
getBalance(card_num)	
pay(card_num, total_price)	
getId()	CreditCard
getBalance()	
setBalance(balance)	
getVerifyCode()	Prepayment
getProductNum()	
checkPrePayment(in PrePayment): Product	VerifyManager
savePrePayment(verifyCode, product, productNum)	
sendMsg(oppo_id, msg_type, msg_des, msg_dir)	Communicator
recvMsg(oppo_id, msg_type, msg_des, msg_dir)	
getSales()	SalesReport
updateSales(product_id, product_num, total_price)	
getTotalSalesReport()	

감사합니다