

OOPT 2050 Implementation

201612020 조석현 201811300 하승래
202010375 김만재 201811220 정유빈

Contents

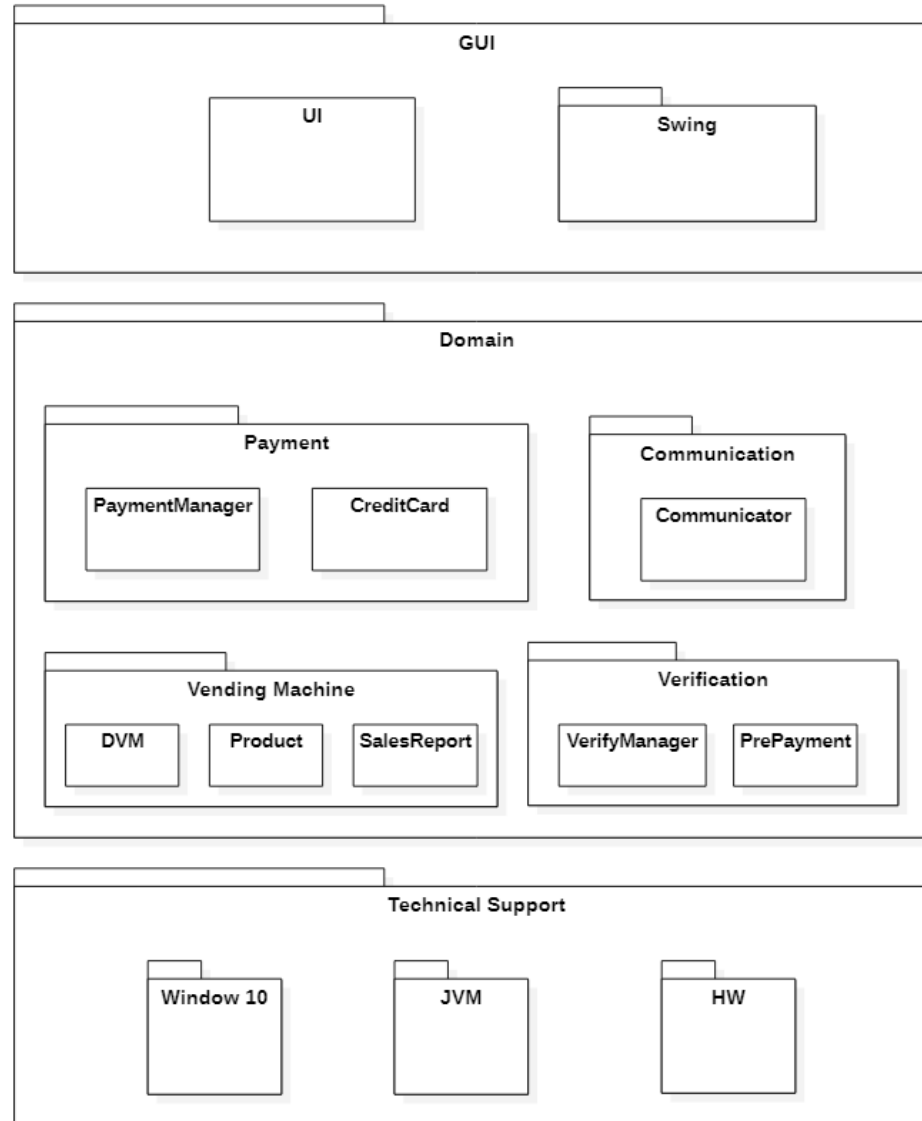
● 1. System Architecture

● 2. Design Class Diagram

● 3. System Test Case

● 4. Traceability

1. System Architecture



2. Design Class Diagrams



3. System Test Case

Use Case	Test Case Description	Expected Result	Data Input	Pass / Fail
1. 구입할 상품 선택	사용자가 선택한 상품 품목이 정상적으로 전달된 경우	재고확인을 진행한다.	콜라 / 10개 재고가 있는 상품에 대해 콜라 / 2개 입력	PASS
	사용자가 선택한 상품 품목이 정상적으로 전달되지 않은 경우	오류를 출력하고 상품 품목의 재선택을 요청한다.	GUI에서 예외 차단	PASS
	사용자가 입력한 상품과 상품의 개수에 대해 재고확인을 정상적으로 한 경우	결제 화면으로 이동한다	콜라 / 10개 재고가 있는 상품에 대해 콜라 / 2개 입력	PASS
	사용자가 입력한 상품과 상품의 개수에 대해 재고확인을 정상적으로 하지 못한 경우	오류를 출력하고 상품구입을 취소한다.	GUI에서 예외 차단	PASS
2. 타 자판기 재고 및 위치 안내	다른 자판기로부터 사용자가 선택한 상품의 재고가 있다는 응답을 받은 경우	사용자에게 재고가 존재하는 자판기의 위치를 제공한 후, 선결제를 요청한다.	메시지 내에 비어 있는 항목이나 형식이 틀린 항목이 없는 메시지 입력	PASS
	다른 자판기로부터 사용자가 선택한 상품의 재고가 없다는 응답을 받은 경우	사용자에게 재고가 존재하지 않는다는 메시지를 출력하고 상품선택 화면으로 이동한다.	메시지 내에 비어 있는 항목이 있는 메시지 입력	PASS
			메시지 내에 형식이 올바르지 않는 항목이 있는 메시지 입력	PASS
	다른 자판기로부터 응답이 오지 않을 경우	사용자에게 재고가 존재하지 않는다는 메시지를 출력하고 상품선택 화면으로 이동한다.	메시지를 입력하지 않음	PASS
	사용자가 선결제 여부를 선택한 경우	선결제 화면으로 이동한다.	선결제 다이얼로그에서 확인 선택	PASS
사용자가 선결제 여부를 선택하지 않은 경우	상품선택 화면으로 이동한	선결제 다이얼로그에서 캔슬 선택	PASS	
		선결제 다이얼로그를 강제 종료	PASS	

3. System Test Case

Use Case	Test Case Description	Expected Result	Data Input	Pass / Fail
3. 결제	사용자의 카드가 유효한 카드일 경우	사용자의 카드 잔액을 확인한다.	자판기 내에 저장되어 있는 16자리 숫자 카드 번호 입력	PASS
	사용자의 카드가 유효하지 않은 카드일 경우	오류를 출력하고 결제를 종료한다.	자판기 내에 저장되어 있지 않은 카드 번호 입력	PASS
	사용자의 카드의 잔액이 충분한 경우	선택한 상품에 대한 결제를 진행한다.	잔액을 초과하지 않는 결제 금액 입력	PASS
	사용자의 카드의 잔액이 충분하지 않은 경우.	오류를 출력하고 결제를 종료한다.	잔액을 초과하는 결제금액 입력	PASS
4. 선결제	사용자의 카드가 유효한 카드일 경우	사용자의 카드 잔액을 확인한다.	자판기 내에 저장되어 있는 16자리 숫자 카드 번호 입력	PASS
	사용자의 카드가 유효하지 않은 카드일 경우	오류를 출력하고 상품 선택화면으로 이동한다.	자판기 내에 저장되어 있지 않은 카드 번호 입력	PASS
	사용자의 카드의 잔액이 충분한 경우	선택한 상품에 대한 결제를 진행한다.	잔액을 초과하지 않는 결제 금액 입력	PASS
	사용자의 카드의 잔액이 충분하지 않은 경우.	오류를 출력하고 상품 선택화면으로 이동한다.	잔액을 초과하는 결제금액 입력	PASS

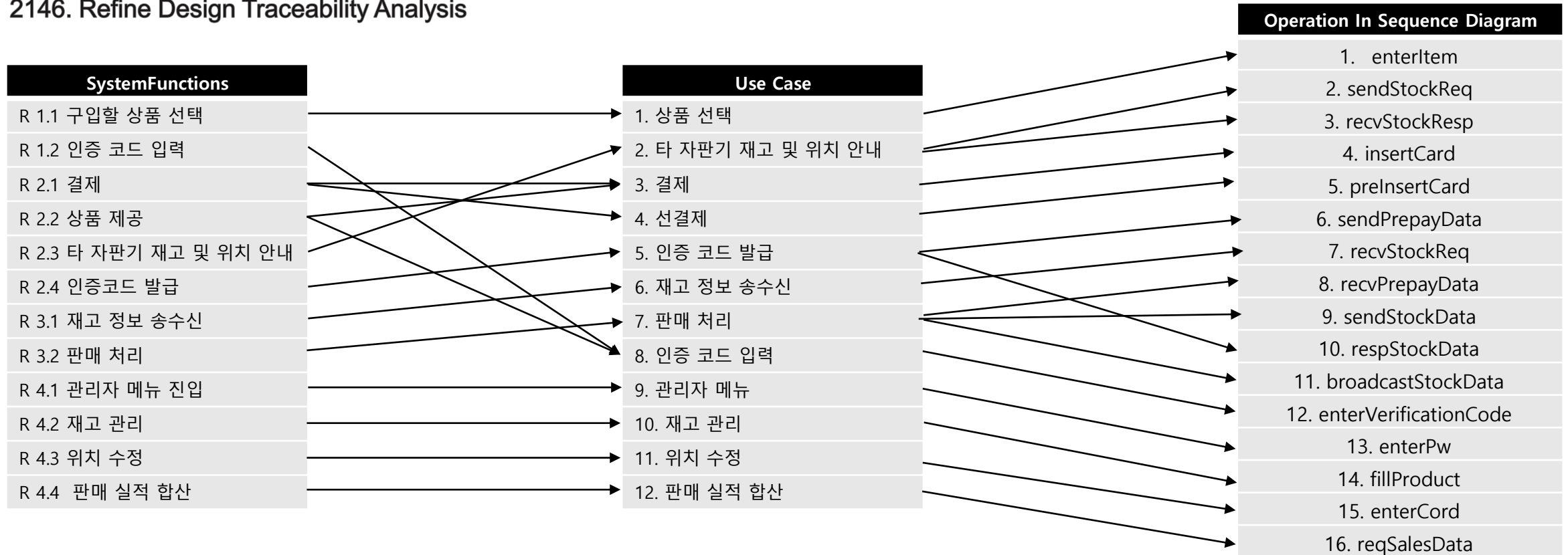
3. System Test Case

Use Case	Test Case Description	Expected Result	Data Input	Pass / Fail
5.인증코드 발급	다른 DVM에서 받은 프로토콜 (음료 판매 확인)이 유효한 경우	사용자에게 인증코드를 전달한다.	메시지 내에 비어 있는 항목이나 형식이 틀린 항목이 없는 메시지 입력	PASS
	다른 DVM에서 받은 프로토콜 (음료 판매 확인)이 유효하지 않은 경우	결제 금액을 환불한 후 상품 선택화면으로 이동한다.	메시지 내에 비어 있는 항목이 있는 메시지 입력	PASS
			메시지 내에 형식이 올바르지 않는 항목이 있는 메시지 입력	PASS
	다른 DVM에서 5초 이상 음료 판매 확인이 오지 않는 경우	결제 금액을 환불한 후 상품 선택화면으로 이동한다.	메시지를 입력하지 않음	PASS
6. 재고 정보 송수신	다른 DVM에서 받은 프로토콜 (재고 확인 요청)을 잘 분석한 경우	알맞은 응답(재고 확인 응답)을 송신한다.	메시지 내에 비어 있는 항목이나 형식이 틀린 항목이 없는 메시지 입력	PASS
	다른 DVM에서 받은 프로토콜 (재고 확인 요청)을 잘 분석하지 못한 경우	조치를 취하지 않는다.	메시지 내에 비어 있는 항목이 있는 메시지 입력	PASS
			메시지 내에 형식이 올바르지 않는 항목이 있는 메시지 입력	PASS
7.판매처리	다른 DVM에서 받은 프로토콜(선결제 확인)을 잘 분석한 경우	받은 상품의 정보와 인증코드를 내부에 저장한다.	메시지 내에 비어 있는 항목이나 형식이 틀린 항목이 없는 메시지 입력	PASS
	다른 DVM에서 받은 프로토콜(선결제 확인)을 잘 분석하지 못한 경우	조치를 취하지 않는다.	메시지 내에 비어 있는 항목이 있는 메시지 입력	PASS
			메시지 내에 형식이 올바르지 않는 항목이 있는 메시지 입력	PASS

3. System Test Case

Use Case	Test Case Description	Expected Result	Data Input	Pass / Fail
8.인증코드 입력	사용자가 입력한 인증코드가 내부에 저장된 인증코드들 중 하나와 동일한 경우	상품 수령을 진행한다.	자판기 내부에 저장이 되어있는 인증코드를 입력	PASS
	사용자가 입력한 인증코드가 내부에 저장된 인증코드들과 동일하지 않은 경우	오류 메시지를 출력 후 상품 선택 화면으로 이동한다.	자판기 내부에 저장이 되어있지 않은 인증코드를 입력	PASS
10. 관리자 메뉴	비밀번호가 유효할 경우	관리자 메뉴로 이동한다.	자판기 내부에 저장이 되어있는 비밀번호를 입력	PASS
	비밀번호가 유효하지 않을 경우	비밀번호 오류 메시지를 출력 후 관리자 메뉴로 이동한다.	자판기 내부에 저장이 되어있는 비밀번호와 다른 문자 또는 숫자를 입력	PASS
10. 재고관리	관리자가 범위 내의 재고 값을 입력한 경우	입력된 값에 기존 재고를 더하 재고를 수정한다.	0 ~ 999의 숫자 중 기존에 있던 재고와 더하여 999를 초과하지 않는 숫자 입력	PASS
	관리자가 범위 밖의 재고 값을 입력한 경우	오류를 출력하고 관리자 메뉴로 이동한다.	0 ~ 999의 숫자 중 기존에 있던 재고와 더하여 999를 초과하는 숫자 입력	PASS
11. 위치수정	관리자가 범위 내의 좌표값을 입력한 경우	입력된 값으로 좌표값을 수정한다.	X, Y 좌표 각각 0 ~ 99 범위 내의 숫자 입력	PASS
	관리자가 범위 밖의 좌표값을 입력한 경우	오류를 출력하고 재입력을 받는다.	GUI에서 예외 차단	PASS
12. 판매실적 합산	관리자가 판매실적 합산 기능을 요청하였을 경우	월 단위로 판매 금액을 누적하여 출력한다.	판매 실적 버튼을 누를 경우	PASS

2146. Refine Design Traceability Analysis



2146. Refine Design Traceability Analysis

Operation In Sequence Diagram
1. enterItem
2. sendStockReq
3. rcvStockReq
4. insertCard
5. preInsertCard
6. sendPrepayData
7. rcvStockReq
8. rcvPrepayData
9. sendStockData
10. respStockData
11. broadcastStockData
12. enterVerificationCode
13. enterPw
14. fillProduct
15. enterCord
16. reqSalesData

Operation in interaction diagram
enterItem(product_code, product_num)
getProduct(product_code)
sendStockReq(product_code, product_num)
sendMsg(oppo_id, msg_type, product_code, product_num, dst_cord_x, dst_cord_y)
rcvMsg(oppo_id, msg_type, product_code, product_num, dst_cord_x, dst_cord_y)
rcvStockReq(product_code, product_num, dst_id, dst_cord_x, dst_cord_y)
insertCard(card_num)
checkValid(card_num)
checkBalance(card_num, total_price)
getBalance()
pay(card_num, total_price)
setBalance(balance)
broadStockData(product_code, product_num)
updateSales(product_id, product_num, total_price)
preInsertCard(card_num, product_code, num)
refund(card_num, total_price)
createVerificationCode(product_code, dst_id)
sendPrepayData(product_code, product_num, verification_code)
setNum(product_num)
getProductInfo(product_code)
getInfo()
getCord_x()
getCord_y()
sell_forPrepay(product_code, num)
savePrePayment(verify_code, product, product_num)
enterVerificationCode(verify_code)
checkPrePayment(verify_code)
checkAdminPw(password)
setProductNum(num, code)
setCoordinate(x, y)
getSalesReportInfo()
getTotalSalesReport()

Method	Class
enterItem(String product_code, String product_num)	DistributedVendingMachine
insertCard(String card_num, String product_code, int num)	
preinsertCard(String card_num, String product_code, int num)	
selectPaymode(boolean isPrepay, String total_price)	
sendStockReq(String product_code, String product_num)	
rcvStockReq(String product_code, String product_num, String dst_id, dst_cord_x, dst_cord_y)	
sendPrepayData(String product_code, int product_num, String verification_code)	
enterVerificationCode(String verify_code)	
checkAdminPW(String password)	
setProductNum(int num, String code)	
setCoordinate(int x, int y)	
getSalesReport()	
getProductInfo(String productCode)	
sell_forPrepay(String product_code, int num)	
createVerificationCode(String product_code, String dst_id)	
refund(String card_num, int total_price)	
getCord_x()	Product
getCord_y()	
getProduct(String product_code)	
setNum(int product_num)	
getPrice()	PaymentManager
getInfo()	
checkValid(String card_num)	
checkBalance(String card_num, int total_price)	CreditCard
pay(String card_num, int total_price)	
getBalance()	Prepayment
setBalance(int balance)	
getProduct()	VerifyManager
getProductNum()	
checkPrePayment(String verify_code)	Communicator
savePrePayment(String verify_code, String product_code, int product_num)	
sendMsg(String oppo_id, int msg_type, String product_code, int product_num, int cord_x, int cord_y, String verify_code)	SalesReport
rcvMsg(String oppo_id, int msg_type, String product_code, int product_num, int cord_x, int cord_y, String verify_code)	
broadStockData(String product_code, int product_num)	
updateSales(String product_id, int product_num, int total_price)	
getTotalSalesReport()	

감사합니다