

Spec Review 통합본

OOPT 1000: Planning 문서

- 1001. Define Draft Plan

- Non Functional Requirements

- Non Functional Requirements

- 직관적인 UI를 제공하고 사용하기 편해야 한다.
 - Java 사용
 - 네트워크 통신 중 데이터 누락이 없어야 한다.
 - 단독망 사용
 - 네트워크 통신 중 Latency 가 없어야 한다.

- Resource Estimation

👉 "직관적인 UI를 제공하고 사용하기 편해야 한다"라는 말은 너무 정성적인 평가입니다. 구체적으로 어떻게 직관적인 UI와 편한 UX를 제공했는지를 정량적으로 파악할 방법이 있는지 작성해야 합니다.

⇒ 직관성은 곧 사용성이므로 이미 존재하고 많이 이용되는 시스템을 벤치마킹한다고 말하면 이는 증명되는 것이므로 해당 서술로 수정하는 것도 좋습니다.

👉 "네트워크 통신 중 데이터 누락과 Latency가 없어야 한다."라는 항목은 물리적으로 딜레이가 아예 없을 수는 없어 실현하기 어렵습니다. 따라서 '투명성'과 '무결성'이 제공된다고 수정하면 좋을 것이라 생각합니다.

💡 Java를 사용할 때 JDK 버전도 보고서에 작성하셔야 합니다.

- 1003. Define Requirements

- Performance Requirements

- 자판기 간 통신 응답 시간이 1초 이내로 수행되어야 한다.
데이터를 확인하는 시간이 1초 이내로 수행되어야 한다.

3.3 Performance Requirements

- Network Message들의 전송속도는 0.1s안에 이루어 지고 전송간 오류가 없어야 한다.
- 사용자가 screen에 입력 후 해당 기능이 실행되는데 걸리는 시간은 **0.1s** 이내여야 한다.

SRS 3.3 Performance Requirements 항목, 20p

☞ Q. 1초인가요? 0.1초인가요? SRS 문서에선 0.1초라고 되어있어 서로 말이 상충됩니다. 하나로 통일 부탁드립니다.
A. 1초로 통일하겠습니다.

• 1004.Record Term in Glossary(p.7)

Verification Code	단어에 대한 한글
Stock	재고
Count	Item의 수량
...	...



Stock과 Count이 문서상 같은 표현으로 사용되고 있습니다. 같은 의미를 혼용하고 있는데, **Stock을 '품목(Item의 종류)'으로 고치거나 혼동을 줄이기 위해 한 가지 표현으로 통일하는 것이 좋습니다.**

또한 Stock을 '재고의 수량'을 표현한 것이라면 "**재고**" 대신에 "**재고량**"으로 수정하셔야 합니다. '재고'는 물건을 나타내는 단어지 수량을 표현하는 단어가 아닙니다.

• 1006. Define Business Use Case (p.9)

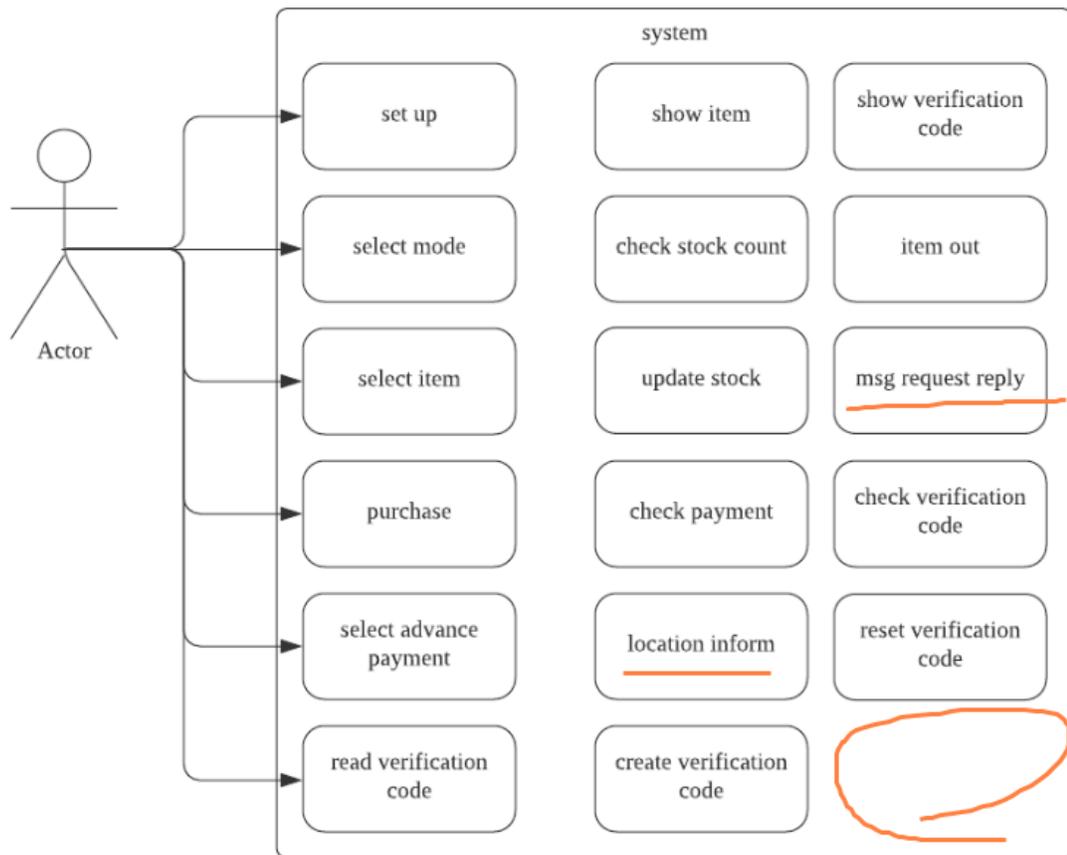
• Identify use cases

- **Quiet vending machine** : 정류장 자동판매기에 관련된 문제
- Identify use cases
 - Use-cases by actor-based
 - Set Up, Select Mode, Select Item, Purchase, Select Verification Code
 - Use-case by evident-based
 - Show Item, Check Stock Count, Update stock, Check Create code, Show Code, Item Out, Check Verification Message Request, Message Response
- Allocate system functions into related use cases

💡 'Use-cases by actor-based' 에 Use-cases by evident-based인 'Set UP'이 있습니다. Set up은 최초 프로그램 실행시 동작되는 UseCase 이므로 Hidden이 맞습니다.

💡 Use-case by evident-based 에서 Use-case's' by evident-based로 오타 수정 부탁드립니다.

- use case diagram



👉 'set up'은 hidden use case로 actor가 없어야 하는데 diagram 상에서는 있는 것으로 그려져 있습니다.

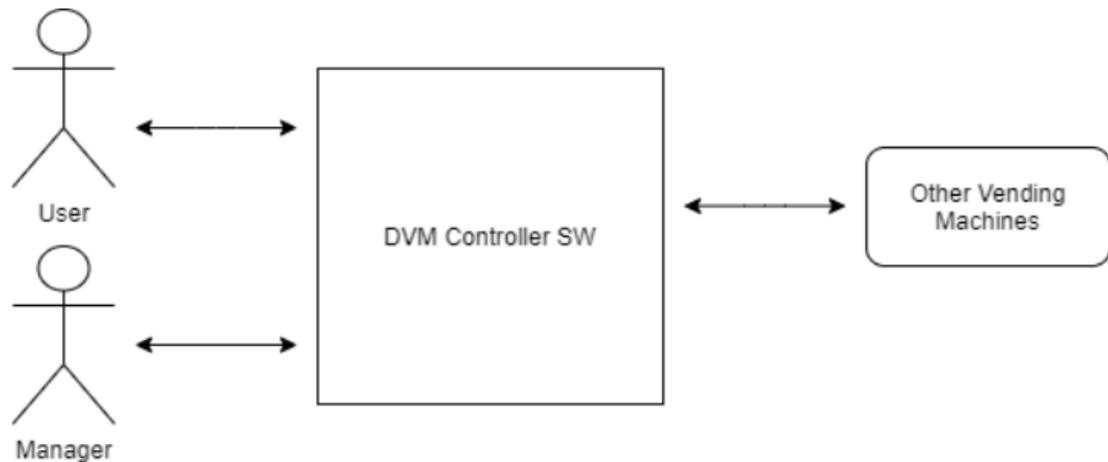


UseCase의 명칭 오류

(없는 항목) → manage stock, advance purchase

msg request reply → message request, message response

location inform → inform location



위 그림(Define System boundary)에는 Manager Actor가 있는데 v2 버전에선 UseCase Diagram에 Actor만 표시되어있습니다. User와 Manager를 구분하여 각각 UseCase를 연결해 UseCase Diagram을 나타내는 것이 좋습니다.



위 내용들은 2030 문서부터 Refine 항목에 잘 고쳐져 있어 동기화를 부탁드립니다.

- Describe use cases
 - 4. Show Item

Use Case	4. Show Item
Actor	None
Description	- 사용자가 현재 자판기의 재고 유무에 상관없이 구매 가능한 음료를 보여준다.

☞ "사용자가 현재 자판기의 재고 유무에 상관없이 구매 가능한 **음료**를 보여준다." → "사용자가 현재 자판기의 재고 유무에 상관없이 구매 가능한 **모든 음료**를 보여준다." 로 구체화하여 수정 부탁드립니다.

- 7. Update Stock

☞ Q. '현장구매로 음료 구매시(=FR2.5=UC8)' 또는 '선구매코드로 음료 구매시(=FR2.8=UC11)' 또는 '관리자가 재고 변경시(=FR1.3=UC3)' 이 3가지 경우에 해당 VM의 재고가 변경되었음을 broadcast를 통해 곧 바로 알려서 모든 VM들의 재고 정보를 매 순간 동기화 시키는 것인지
vs
아니면 이 3가지 경우마다 broadcast를 하는 것이 아니라 다른 VM이 재고 정보를 요청할 때 응답만 하는 것인지 궁금합니다.
(OOPT 2030와 OOPT 2040 문서에서도 마찬가지로)
A. local 재고 증감 후, remote에 재고 감소 요청을 진행합니다.

💡 이 부분은 v2에서 Flow chart가 누락되면서 시퀀스를 알 수 없어 나온 질문입니다. 누락된 플로우 차트를 재작성 부탁드립니다.

- 12. Inform Location

Use Case	12. Inform Location
Actor	None
Description	- 다른 자판기로부터 응답 msg가 도착한 후 실행되는 use case. - 사용자에게 선택한 음료를 판매하고 있는 자판기의 위치를 알려준다.

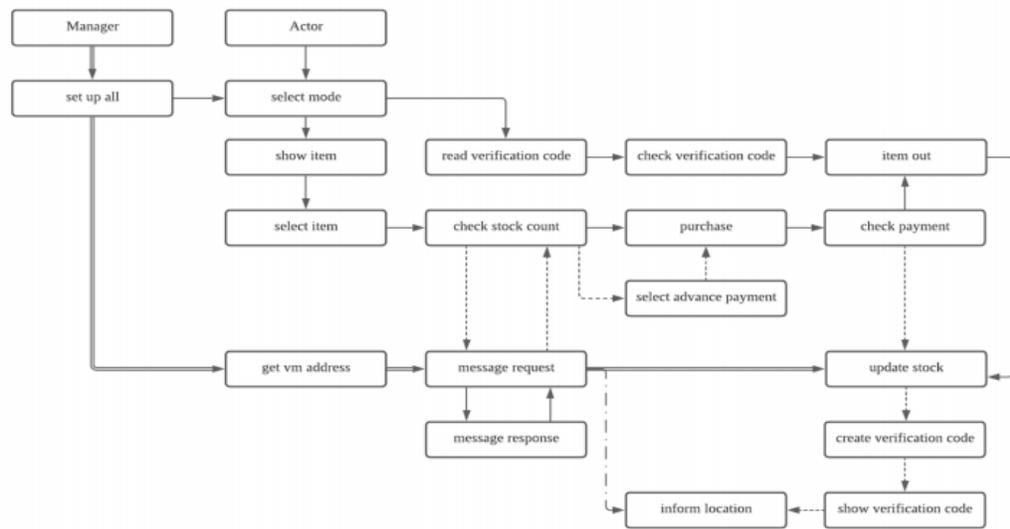
☞ "- 사용자에게 선택한 음료를 판매하고 있는 **자판기**의 위치를 알려준다" → "- 사용자에게, 선택한 음료를 판매하고 있는 **모든 자판기**의 위치를 알려준다" 로 수정 부탁드립니다.

- 15. Item Out

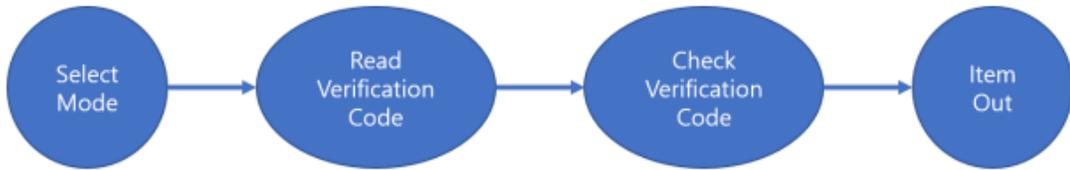
☞ "결제 완료된 음료를 제공하는 use case" → "결제 완료된 음료를 제공하는 use case"
 ⇒ 사소한 표현 수정이므로 넘어가셔도 됩니다.

- 플로우차트 (p.11)

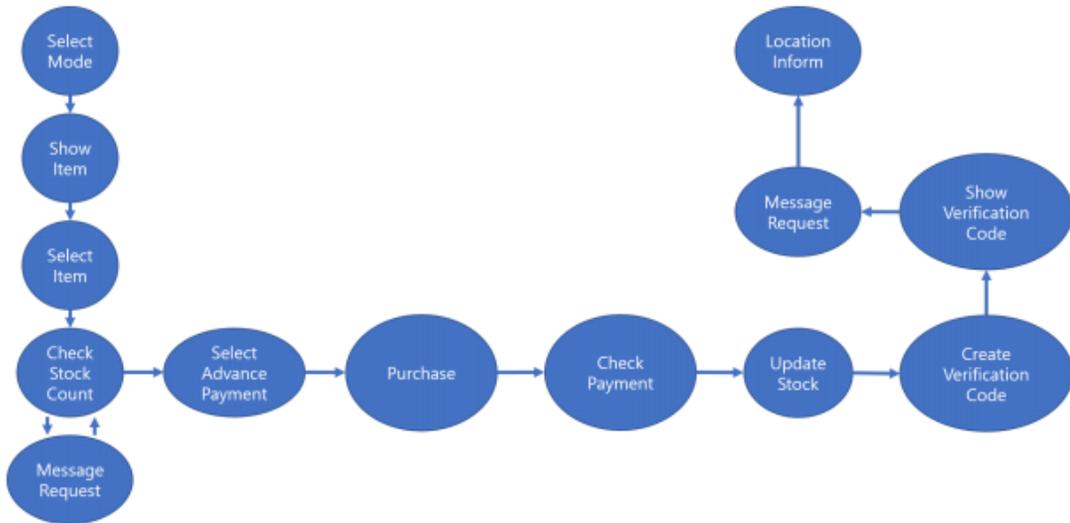
FR flow chart



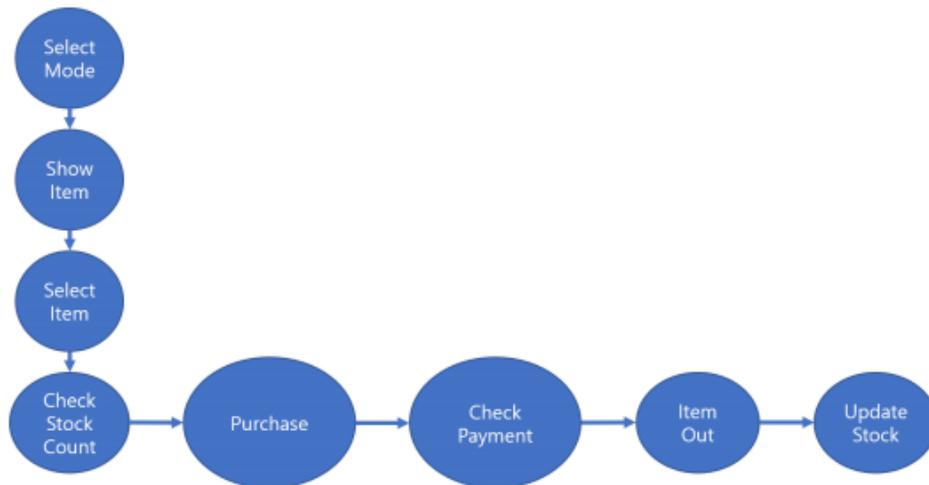
a. 선결제된 코드를 사용하여 음료를 구매하는 경우



b. 다른 자판기에 있는 음료를 현재 자판기에서 구매하여 선결제 코드를 발급받는 경우

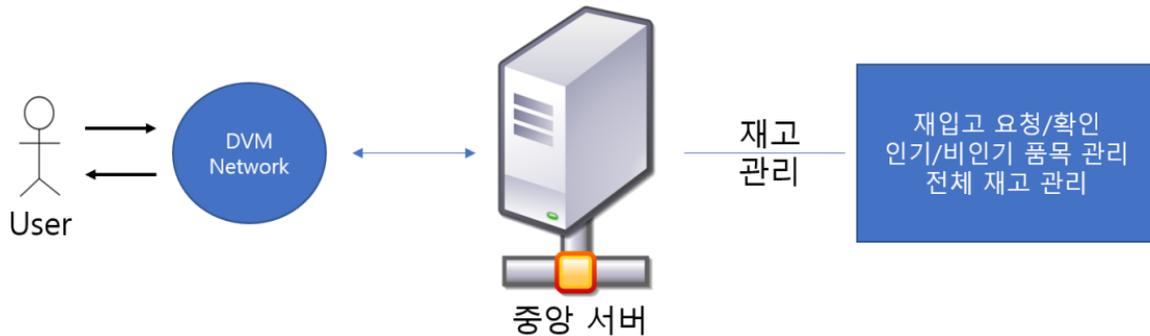


c. 내 자판기에 있는 음료를 구매하여 먹는 경우



☞ 각 UseCase 간의 연결을 나타내는 플로우차트가 v2에 누락된 것을 확인했습니다. 해당 내용이 v2에 없어 이후 페이지의 TestCase에 언급된 '다음 단계'가 무엇을 의미하는지 알기 어렵습니다. 해당부분 다시 추가 부탁드립니다.

- 1008. Define Draft System Architecture



☞ Q. 1001. Define Draft Plan의 Motivation에서는 중앙 서버를 두지 않는다고 했는데 1008. Define Draft System Architecture에는 중앙 서버 그림이 그려져 있습니다. 어떤 것이 맞는 건가요?
 A. next version이라고 적혀있듯 위와 같은 구조로 확장할 계획이고, 현 개발단계 사진은 아닙니다.

- 1009. Refine Plan (v2 p.19)

- Functional Requirement Test Case

💡 "Set up"는 Usecase 명입니다. Function명은 Set up all이므로 p.9에 정의된 Function 이름으로 수정하셔야합니다.

- No.6 Check Stock Count
 - Negative Test Case
 - 재고가 있는데 없다고 파악하는 경우, 다른 자판기의 재고 수를 요청한다

6	Check Stock Count	1. 기본: 실제 재고와 일치하는 값으로 저장해 다음단계를 진행한다.	1. 재고가 있는데 없다고 파악하는 경우: 다른 자판기의 재고 수를 요청한다.
---	-------------------	--	---



Q. 이게 무슨 뜻인가요?

A. local에 재고가 없고 remote에 재고가 있는 경우를 나타냅니다.



"자판기에 재고가 없지만, 다른 자판기에 재고가 있는 경우"로 수정 부탁드립니다.

- No.12 Location Inform

- Positive Test Case



"다른 자판기의 위치를 출력한다" → "결제할 상품이 판매되는 모든 자판기의 위치를 출력한다."로 더 자세하게 수정 부탁드립니다.

- No.18 Reset Verification Code

- Positive Test Case

- 사용된 코드를 폐기하고, 모든 자판기에 방송한다.



'방송한다'를 "코드 폐기 내용을 알린다"로 수정하는게 좋습니다.

-
- JAVA 언어를 사용할 때는 JDK 버전도 같이 써야합니다.

- Resources

- Man Month : 5 Persons
 - A Team Leader
 - A Document Manager
 - 2-3 Engineers
 - Period : 6 Days(Around 20Hours)
 - Hardware : Intel PC
 - Software
 - OS : Windows 10
 - Programming Language : JAVA

- User Interface requirements

- User Interface requirements
 - UI가 직관적이어야 한다.
 - 처음 사용하는 사람도 큰 문제없이 사용할 수 있어야 한다.

 "UI가 직관적이어야 한다"를 구체적으로 나타낼 수 있는 방안과 함께 정량적으로 서술하거나, 직관성은 곧 사용성이므로 어떤 유명한 시스템을 벤치마킹한다는 내용이 추가되어야 합니다.

OOPT 2030 : Analysis 문서

- 2131. Define Essential Use Cases

 Alternative Courses of Events와 Exceptional Courses of Events에서 Line number의 mapping이 안 맞는 경우들이 있습니다. (OOPT 2040 문서도 마찬가지입니다.)

- 1. Set Up (p.3)
 - Cross Reference

Use Case	1. Set Up
Actor	None
Purpose	(business use case에 포함)
Overview	(business use case에 포함)
Type	Hidden
Cross Reference	System Functions: R1.1, R1.2, R4.1, R2.4, <u>R4.1</u> Use Case: "Message Request", "Update Stock", "Select Mode"
Pre-Requisites	N/A

☞ Q. R4.2는 왜 없나요? (OOPT 2040 문서도 마찬가지)
A. 수정하면서 삭제되었습니다.

💡 "R4.1"이 중복되었으므로 삭제 부탁드립니다.

- Typical Course of Events

💡 2. (S): "다른 DVM의 재고를 받아온다" → "다른 DVM의 재고 정보를 받아온다."로 수정 부탁드립니다.

💡 3. (S): "자판기의 재고를 업데이트 한다." → "모든 자판기의 재고 정보를 업데이트 한다."로 수정 부탁드립니다.

- Alternative Courses of Events

Alternative Courses of Events	Line 1. (A3)의 응답이 3번 이상 오지 않을 경우, 해당 (A3)은 재고가 없다고 설정한다.
Exceptional Courses of Events	N/A

💡 적힌 내용은 다른 성공 시나리오의 사례가 아니라 예외처리에 대한 문장이므로 Exceptional Course of Events로 옮기는 것이 적절해 보입니다.

- 2. Select Mode (p.4)

- Alternative Courses of Events

Alternative Courses of Events	Line 1. 세 가지 모드를 동시에 선택할 경우 현장구매 모드로 간주한다.
Exceptional Courses of Events	N/A



현재 구현된 프로그램으로는 마우스 버튼 입력으로 진행되기 때문에 세 가지 모드를 동시에 선택하는 경우가 생길 수 없습니다. 따라서 해당 내용은 삭제처리하는 것이 좋습니다.

- 3. Manage Stock (p.4)

Cross Reference	System Functions: R1.3, R1.2, R2.4 Use Case: "Select Mode", "Update Stock"
Pre-Requisites	N/A
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (A2): 추가된 음료의 재고를 입력한다. 2. (S): 음료 재고를 업데이트 한다. 3. (S): "Select Mode"를 실행한다.
Alternative Courses of Events	Line 1. 기존에 해당 자판기에 없던 음료를 추가할 경우, 새로운 항목을 추가하여 업데이트 한다. Line 2. 음수의 재고를 업데이트 할 경우, 감소시킨다.
Exceptional Courses of Events	Line 1. 음료 종류가 7개를 초과하는 경우 에러를 발생시킨다. Line 2. 재고가 0 미만이면 "잘못 된 값" 메시지를 출력한다.

- Cross Reference



Select Mode는 선행 조건이므로 삭제하고, Pre-Requisites에 추가하는 것이 좋습니다.

- Pre-Requisites



Cross Reference의 수정에 따라 "재고 관리를 누른다"로 수정하는 것이 좋습니다.

- Typical Courses of Events

- 
 1. (A2) : 추가된 음료의 재고를 입력한다.
 → "추가된 음료 수량을 입력한다"로 수정 부탁드립니다.
- 2. (S): 음료 재고를 업데이트 한다.
 → 업데이트는 확실하지 않은 표현이므로 "입력한 값만큼 음료의 재고량을 증가시킨다"로 수정합니다. '재고'는 수량을 나타내는 단어가 아닙니다.

- Alternative Course of Events

- 
 Line 2. "음수의 재고를 업데이트 할 경우, 감소시킨다." → "입력된 값이 음수일 경우, 재고량을 입력된 값만큼 감소 시킨다." 로 구체화하여 작성 부탁드립니다.

- Exceptional Course of Events

- 
 Line 2. "재고가 0 미만이면 '잘못된 값' 메시지를 출력한다." → "입력 결과 예상 재고량이 0 미만이면 최신화를 하지 않고, '잘못된 값' 알림 메시지를 출력한다."로 수정 부탁드립니다.

- 4. Show Item (p.5)

Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S): 음료 항목을 모두 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1. 취소 버튼을 누르면 "Select Mode"를 실행한다.

- Typical Courses of Events

- 
 1. (S) : "음료 항목을 모두 보여준다" → 20가지 음료 선택 버튼이 출력되는 상황이므로 "20가지 음료 선택 버튼을 모두 보여준다."로 수정합니다.

- 5. Select Item (p.5)

- Typical Courses of Events

Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (A1): 사용자가 구매하고자 하는 음료를 선택한다. 2. (S): 선택한 음료에 대해 “Check Stock Count”를 실행한다. 3-1. (S): 재고가 현재자판기에 있으면 “Purchase”를 실행한다. 3-2.1. (S): 재고가 다른 자판기에 있으면 “Select Advance Payment”를 실행한다.
---------------------------	---

 3-1 부분을 3으로 두고, 3-2 내용은 또 다른 성공 시나리오이므로 Alternative Course of Events로 이동시키는 것이 좋습니다.

- 7. Update Stock (p.7)

Cross Reference	System Functions: R2.4, R1.1, R1.3 Use Case: “Set Up”, “Manage Stock”
Pre-Requisites	최신화 해야되는 음료 정보와 최신화 값이 입력됨
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S) : 음료에 대해 재고를 최신화 한다.

- Cross Reference

 다른 자판기에서 선결제 하여 음료 재고 정보가 변했을 때도 갱신이 일어나야 하므로 Use Case 20. Message Response(R4.2)도 추가되어야 합니다.

- Pre-Requisites

 '최신화'와 '업데이트' 두 가지로 용어가 혼용되고 있는데, 용어는 통일되는 편이 좋습니다.

- Typical Course of Events



1. (S) : "음료에 대해 재고를 최신화 한다." → "수량 변동 내역이 있는 음료 재고 정보를 업데이트 한다." 로 수정하는 것이 좋습니다.

- 10. Check Payment (p.8)
 - Exceptional Courses of Events

Courses of Events	
Exceptional Courses of Events	Line 1. 정상적인 결제 카드가 아니라면, "잘못된 카드"라는 에러 메시지를 출력한다.



2050 문서를 보았을 때 해당 예외처리 사항은 존재하지 않았습니다. 따라서 해당 문구는 삭제하는 것이 좋습니다.

- 11. Select Advance Payment (p.8)

Cross Reference	System Functions: R2.8, R2.3, R2.6, R4.1, R1.2 Use Case: "Check Stock Count", "Advance Purchase", "Message Request", "Select Mode"
Pre-Requisites	음료가 선택됨
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (A1): 선택한 음료에 대해 선구매를 선택한다. 2. (S): 선택된 음료에 대해 "Message Request"로 재고를 확인한다. 3. (S): 여전히 재고가 남아있다면 "Message Request"로 다른 자판기에 있는 음료의 재고를 짝는다. 4. (S): 선구매 진입을 반환한다.
Alternative Courses of Events	1. 확인과 취소를 둘다 누를 경우 사용자로부터 재입력 받는다.
Exceptional Courses of Events	Line 1. 취소를 누를 경우 "Select Mode"로 돌아간다. Line 1. 위치보기를 누를 경우 "Inform Location"을 실행한다. Line 3. 재고가 없다면 품질 메시지를 띄우고, "Select Mode"로 돌아간다.

- Cross Reference

 Select Mode는 세 가지 모드를 선택하는 Use Case이므로 Cross Reference에 적절하지 않습니다. 따라서 R1.2(Select Mode)는 삭제하는 것이 좋습니다. Pre-Requisites에서 이미 'Select Item'(R2.2) Usecase가 진행된 상태이므로 R2.2는 굳이 추가하지 않아도 됩니다. 대신 Exceptional Courses of Event에서 "Infrom Location"이 실행되므로 R2.9를 추가하는 것이 맞습니다.

- Typical Courses of Events

 1. (A1) : "선택한 음료에 대해 선구매를 선택한다" → "선택한 음료에 대해 선구매를 진행한다." 로 수정 부탁드립니다.

- Alternative Courses of Events

 해당 내용은 예외 처리 사항이므로 Exceptional Courses of Events에 있어야 합니다. 하지만 2050에서 구현된 프로그램은 **버튼 입력으로 진행되어 동시에 입력되는 경우가 없으므로 해당 문구를 삭제하는 것이 좋습니다.**

- Exceptional Courses of Events

 해당 내용들은 또 다른 성공 시나리오 이므로 Alternative Courses of Events로 옮기는 것이 좋습니다.

- 12. Inform Location (p.9)

Type	Content
Cross Reference	System Functions: R2.9, R2.8, R4.1 Use Case: "Select Advance Payment", "Message Request"
Pre-Requisites	음료가 선택됨

- Pre-Requisites

 위치보기 버튼을 누를 경우 실행되므로 "음료가 선택됨"이 아니라 "위치보기를 누름"으로 바뀌어야 합니다.

- 13. Create Verification Code (p.10)

Cross Reference	System Functions: R2.10, R2.6 Use Case: "Advance Purchase"
Pre-Requisites	선택된 음료에 대한 결제가 완료됨
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S): 새로운 선구매 코드를 생성한다.

- Pre-Requisites

 보다 정확하게 "선택된 음료에 대한 선결제가 완료됨."으로 수정 부탁드립니다.

- Typical Courses of Events

 1. (S): "새로운 선구매 코드를 생성한다." → "6자리 난수의 선구매 코드를 생성한다." 로 수정해야 합니다. OOPT 1000 문서에서 이미 언급된 내용이기 때문입니다.

- 15. Item Out (p.11)

- Typical Courses of Events

Cross Reference	System Functions: R2.12, R2.5, R3.1 Use Case: "Purchase", "Read Verification Code"
Pre-Requisites	선택된 음료에 대한 결제가 완료됨
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S): 선택된 음료를 제공한다.
Alternative	N/A

 1. (S) : "선택된 음료를 제공한다." → "판매된 음료를 제공한다."로 수정 부탁드립니다.

- 16. Read Verification Code (p.12)

Cross Reference	System Functions: R3.1, R1.2, R3.2, R3.3, R2.12 Use Case: "Select Mode", "Check Verification Code", "Reset Verification Code", "Item Out"
Pre-Requisites	N/A
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (A1): 선구매 코드를 입력한다.
Alternative Courses of Events	Line 1. 동시에 여러가지 숫자를 누를 경우 아무것도 입력받지 않는다.
Exceptional Courses of Events	Line 1. 취소를 누를경우 "Select Mode"로 돌아간다.

- Cross Reference



해당 UseCase는 이미 'Select Mode'가 선행된 상태이므로 R1.2는 삭제해도 됩니다.

- Pre-Requisites



"Select Mode에서 선구매 버튼을 누름."이라고 수정합니다.

- Typical Courses of Events



"선구매 코드 6자리를 입력한다"로 구체적으로 수정합니다.

- Alternative Courses of Events



Exceptional Courses of Events로 내용을 옮깁니다.

- Exceptional Courses of Events



Alternative Course of Events로 내용을 옮깁니다.

- 17. Check Verification Code (p.12)

Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S): 입력된 코드를 가지고 있는지 검사한다.
---------------------------	--

- Typical Courses of Events



1. (S) : "입력된 코드를 가지고 있는지 검사한다." → "입력된 코드가 유효한지 검사한다"로 수정합니다.

- 18. Reset Verification Code (p.13)

Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S): 사용한 선구매 코드의 정보를 삭제한다. 2. (S): "Message Request"를 통해 모든 자판기에게 해당 코드가 사용되었음을 Broadcast 한다.
---------------------------	--

- Typical Courses of Events



1. (S) : "사용한 선구매 코드의 정보를 삭제한다" → "사용한 선구매 코드 정보를 삭제한다."
⇒ '~의'는 일본식 표현이라 재량껏 수정하시면 됩니다.

- 19. Message Request (p.14)

- Typical Courses of Events

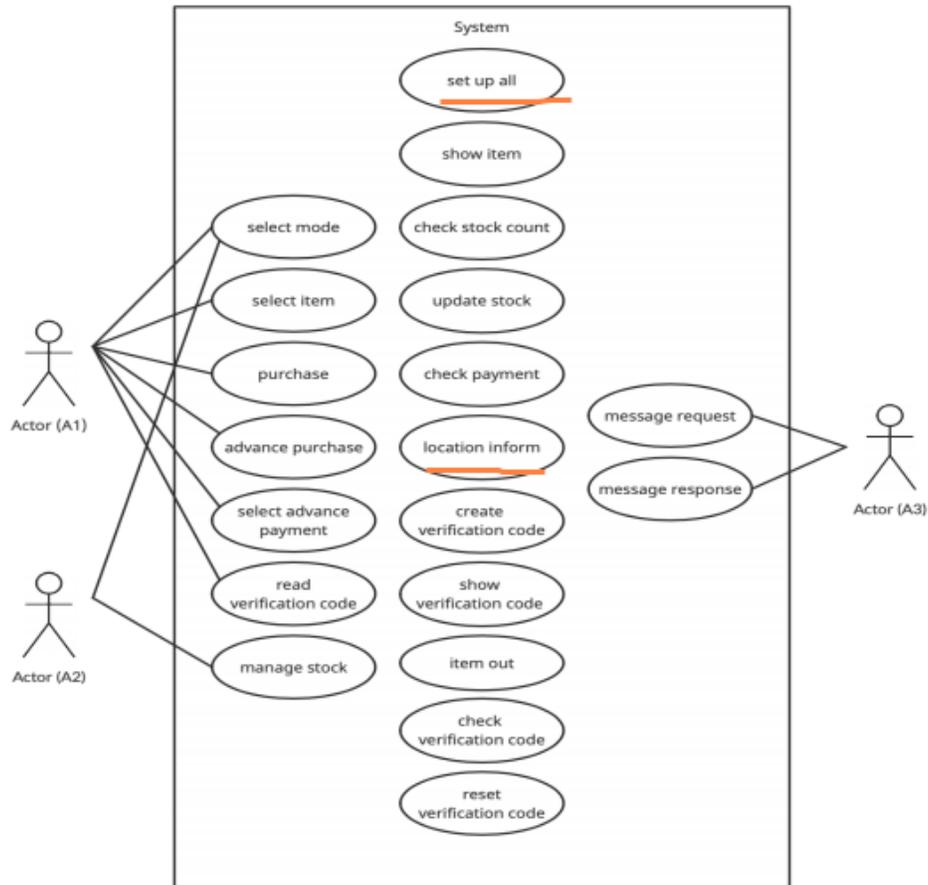
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1.1. (S): Broadcast의 경우, 네트워크에 등록된 모든 (A3)에게 보내고자 하는 메시지를 전달한다. 1.2. (S): 돌려받은 응답을 반환한다. 2.1. (S): 특정 (A3)에게 메시지를 보낼 경우, 해당 (A3)에게 메시지를 전달한다. 2.2 (S): 돌려받은 응답을 반환한다.
---------------------------	--



개행이 안 되어 있습니다.

• 2132. Refine Use Case Diagrams (p.15)

2132. Refine Use Case Diagrams



* (A1): User, (A2): Manager, (A3): Other DVMs



(1) 'set up all'은 function 명입니다. 해당 function에 대응되는 UseCase 명칭은 "set up"이므로 수정 부탁드립니다.

(2) 'location inform'은 function 명입니다. 해당 function에 대응되는 UseCase 명칭은 "inform location"이므로 수정 부탁드립니다.

- **2133. Define System Sequence Diagrams (p.16)**



2131. Define Essential Use Cases 항목의 수정사항에 맞게 노트 내용 수정 부탁드립니다.

- **2134. Refine Glossary (p.22)**

- Stock : Item의 재고
- Count : Item의 수량



Q. Stock은 DVM 전체에 걸쳐 있는 Item의 총 개수이고 Count는 특정 VM 내에 있는 Item의 개수인가요? (OOPT 2040 문서의 Class Diagram에 stock이라는 변수가 없는 걸 보고 그렇게 추측했습니다.)

A. 둘 다 같은 의미로 사용되고 있습니다. Stock은 삭제하겠습니다.

OOPT : Software Requirements Specification 문서

1. Introduction

- 1.3 Definitions, Acronyms and Abbreviations

1.3 Definitions, Acronyms, and Abbreviations

No.	Term	Description
1	SW	Software의 약자
2	DVM	Distributed Vending Machine의 약자
3	자판기	DVM의 또 다른 지칭
4	Mode	자판기에서 사용자에게 제공하는 사용 유형 0: showMode, 1: 현장구매모드, 2: 선구매코드사용 모드, 3: 재고 관리 모드
5	Item	자판기에서 판매되는 음료
6	Purchase	Item에 대한 결제
7	Payment	Item에 대한 결제 방법으로 선구매와 현장구매로 나뉨
8	Verification Code	선구매 인증번호
9	<u>Stock</u>	Item의 재고
10	Advanced	다른 자판기에 있는 음료에 대한 선구매 기능으로 Advance가 붙어있을 경우 선구매 결제임을 의미
11	ID	자판기가 가지고 있는 고유한 아이디로 DVM 네트워크에서 자판기를 구별하는 아이디
12	Message	DVM간 네트워크에서 통신할때 전달하는 메시지
13	Msg	message의 축약어
14	HW	Hardware의 약자
15		
16		
17		



앞선 문서에서는 Count와 Stock이 혼용되었고, Stock이란 단어를 삭제하기로 하였으나, 해당 문서에서는 Count가 삭제되고, Stock이 남아있습니다. 통일하는 것이 좋습니다.



15~17번이 공란입니다. 행 삭제 부탁드립니다.

- **2.4 Constraints**

- 동시에 두 개 이상의 메시지를 받고 응답할 수 없다



해당 조건은 2050 문서에서 구현된 버튼 입력 방식으로는 나올 수 없는 경우이므로 삭제하는 것이 좋습니다.

• 3.2 Functional Requirements

- 3.2.9 Item Out (p.17)

3.2.9 Item Out	
구분	내용
요구사항ID	Req-9
요구사항명	<u>Show Verification Code</u>
개요	음료를 제공한다.
사전조건	1. 음료가 선택되어 있다. 2. 선택된 음료에 대해 결제가 완료되었거나, 필요한 선구매 코드가 입력되었다.
시나리오	1. 선택된 음료를 제공한다.



요구사항명이 잘못되어 있습니다. 수정 부탁드립니다.

• 3.3 Performance Requirements

3.3 Performance Requirements

- Network Message들의 전송속도는 0.1s안에 이루어 지고 전송간 오류가 없어야 한다.
- 사용자가 screen에 입력 후 해당 기능이 실행되는데 걸리는 시간은 0.1s 이내여야 한다.



앞선 문서에서 전송 시간은 1초 이내에 이루어지는 것으로 되어있으나 해당 문서에선 0.1초로 되어있습니다. 1초로 수정 부탁드립니다.

OOPT 2040 : Design 문서

- 2141. Define Essential Use Cases

- Traceability Table : Funtional Requirement ↔ Use Case ↔ System Operation(개념) ↔ Method(코드) ↔ Unit Test Case

☞ UC ↔ UC Use Case끼리의 전후관계에 대한 명시가 있었으면 좋겠습니다.

- 4. Show Item(pg 4)

Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S): 모든 <i>Item</i> 을 모두 보여준다.
---------------------------	--

☞ Q. 모든 아이টে을 모두 보여준다는게 수량과 종류를 모두 보여준다는 의미인가요?
A. 음료의 종류만을 보여준다는 의미입니다.

- 5. Select Item(pg 5)

Use Case	5. Select Item
Actor	User
Purpose	(business use case에 포함)
Overview	(business use case에 포함)
Type	Evident
Cross Reference	System Functions: R2.2, R1.2, R2.1, R2.3 Use Case: "Select Mode", "Check Stock Count", "Show Item"
Pre-Requisites	"showItem"이 실행됨

- Pre-Requisites

☞ Q. select item은 현장구매모드에서 사용자가 구매하려는 음료를 선택하는 것이기 때문에 "mode값이 1이다.(현장구매모드이다)"라는 내용을 추가하면 더 명확해져서 좋을 것 같아요

- 6. Check Stock Count(pg 6)

Use Case	6. Check Stock Count
Actor	None
Purpose	(business use case에 포함)
Overview	(business use case에 포함)
Type	Hidden
Cross Reference	System Functions R2.3, R2.2, R4.1 Use Case: "Select Item", "Message Request"
Pre-Requisites	<i>Item</i> 이 선택되었다.

- Pre-Requisites

☞ Q. 음료를 선택하고 재고를 확인하는 부분이니 선택한 자판기의 "mode값이 1이다.(현장구매모드이다)"라는 내용을 추가하면 더 명확해져서 좋을 것 같아요

- 7. Update Stock(pg 6)

Use Case	7. Update Stock
Actor	None
Purpose	(business use case에 포함)
Overview	(business use case에 포함)
Type	Hidden
Cross Reference	System Functions: R2.4, R1.1, R1.3 Use Case: "Set Up", "Manage Stock"
Pre-Requisites	최신화 해야되는 <i>Item</i> 정보와 최신화 값이 입력됨
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S) : <i>Item</i> 에 대해 <i>count</i> 를 최신화 한다.

 Q. 음료의 재고가 어느 시점에서 Update되나요?

- 8. Purchase와 9. Advance Purchase(pg 7)

Use Case	8. Purchase
Actor	User
Purpose	(business use case에 포함)
Overview	(business use case에 포함)
Type	Evident
Cross Reference	System Functions: R2.5, R2.7 Use Case: "Check Payment"
Pre-Requisites	1. <i>Item</i> 이 선택되었고 <i>mode</i> 값이 1 이다. 2. <i>count</i> > 0 이다.
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (A1): 카드를 삽입한다. 2. (S): " <i>getPaymentResult</i> "를 실행하여 결과를 받아 반환한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 2. 결제 실패시 [결제 실패]이라는 내용의 <i>prine</i> 를 실행한다.

Use Case	9.Advance Purchase
Actor	User
Purpose	(business use case에 포함)
Overview	(business use case에 포함)
Type	Evident
Cross Reference	System Functions: R2.6, R2.3, R2.7 Use Case: "Select Advance Payment", "Check Payment"
Pre-Requisites	1. <i>Item</i> 이 선택되었고 <i>isAdvance</i> 가 <i>true</i> 이다.

 Q. Use Case 8번은 Mode=1이면서 isAdvance=false이고 Use Case 9번은 Mode=1이면서 isAdvance=true라는 차이가 있는 것인가요? 그렇다면 그 차이를 명확하게 알 수 있도록 Pre-Requisite에 명시했으면 좋겠네요!
 Q. isAdvance가 뭔가요?
 A. 큰 쓰임새는 없고 간단하게 쓰임. isAdvance가 true인 경우 현장구매 모드에서 선구매를 실시했을 경우 입니다.

- 11. Select Advance Payment(pg 9)

Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (A1): 선택한 <i>Item</i> 에 대해 선구매를 선택한다. 2 (S): 선택된 <i>Item</i> 에 대해 "sendMessage"로 재고를 확인한다. 3 (S): 여전히 재고가 남아있다면 "sendMessage"로 다른 자판기에 있는 <i>Item</i> 의 <i>count</i> 를 깎는다. 4 (S): "advancePurchase"를 실행한다.
---------------------------	--

- Typical Courses of Events
 - 3. (S): 여전히 재고가 남아있다면 "sendMessage"로 다른 자판기에 있는 *Item*의 *count*를 깎는다

- Q. 어떤 VM의 count를 깎아야 하는지 어떻게 알 수 있나요?
 A. 모든 VM의 count를 깎고, 선결제 코드 입력시 구매한 VM을 제외한 나머지 VM에서 해당 음료의 count를 증가시킵니다.

Q. 선결제 코드 생성시 미리 다른 자판기의 재고(count)를 깎는다는 건가요?
 A. 네. 미리 count를 감소시킨 후 선구매를 한 VM에서만 count를 유지하고 그외 VM에서는 해당 음료의 count를 다시 증가시킵니다.

- Alternative Courses of Events

Alternative Courses of Events	1. 확인과 취소를 둘 다 누를 경우 사용자로 부터 재입력 받는다.
Exceptional	Line 1 취소를 누른 경우 "showMode"로 돌아간다

- 1. 확인과 취소를 둘 다 누를 경우 사용자로 부터 재입력 받는다

- Q. 확인과 취소 둘 다 버튼으로 입력을 받기때문에 동시에 입력받는 경우는 없을 것 같습니다.

- 8. Purchase/ Advance Purchase/ 10. Check Payment/ 11. Select Advance Payment /17. Check Verification Code (pg 8-9)

Exceptional Courses of Events	Line 2. 결제 실패시 [결제 실패]이라는 내용의 <i>prine</i> 를 실행한다.
-------------------------------	--

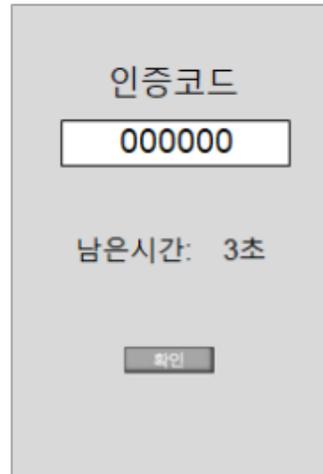
Exceptional Courses of Events	Line 1. 정상적인 결제 카드가 아니라면, [잘못된 카드]이라는 내용의 <i>info</i> 를 실행한다.
-------------------------------	---

- 오류가 발생하는 상황을 명시 후 순서순환에 대한 방식에 대한 내용이 필요할 것 같습니다.

2142. Define Reports, UI, and Storyboards

- 7) Show Verification Code(pg 18)

7) Show Verification Code



인증코드

000000

남은시간: 3초

확인

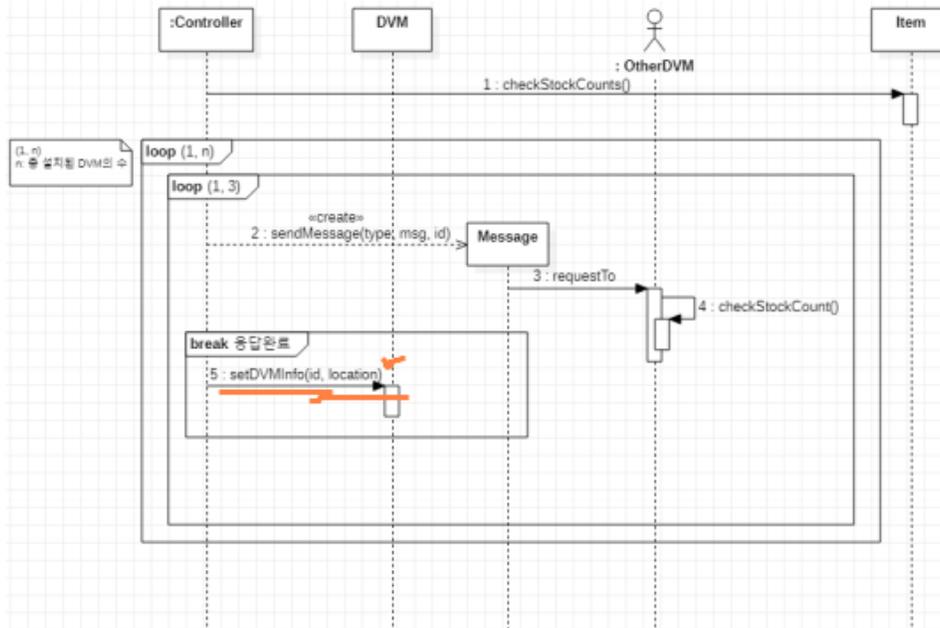
☞ Q. 남은 시간에 관한 정의는 이전 문서에는 없었는데 UI에서는 표현이 되어 있었으나 추후 문서에서는 수정되었습니다.

2143. Define Interaction Diagrams

- (1) Set up(pg 19)

2143. Define Interaction Diagrams

(1) Set Up

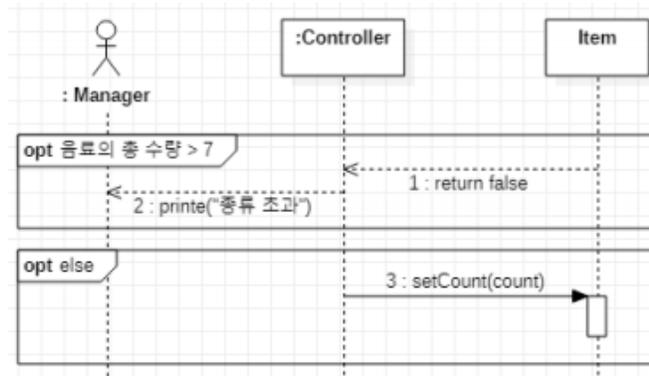


"setDVMInfo" 오퍼레이션은 DVM의 id, location 그리고 item의 count 항목을 업데이트하는 것이므로 매개변수로 item도 같이 넣어야 합니다.

⇒ "setDVMInfo(id, location)" → setDVMInfo(id, location, item)

- (7) Update Stock (pg 22)

(7) Update Stock



☞ Q. 음료의 총 수량이 7초과일경우 종류 초과 라는 에러 메시지를 생성한다고 되어있는데 "음료의 총 수량"이라는 의미는 음료의 종류가 7가지를 넘어간 경우인가요/음료의 재고의 수가 7개를 넘어간 경우인가요?
 A. 음료의 종류의 가짓수가 7개를 초과한 경우입니다.

💡 음료의 종류의 가짓수가 7개 초과한 경우라고 의미가 분명하게 나타나게 수정바랍니다.

Use Case	7. Update Stock
Actor	None
Purpose	(business use case에 포함)
Overview	(business use case에 포함)
Type	Hidden
Cross Reference	System Functions: R2.4, R1.1, R1.3 Use Case: "Set Up", "Manage Stock"
Pre-Requisites	최신화 해야되는 <i>Item</i> 정보와 최신화 값이 입력됨
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S) : <i>Item</i> 에 대해 <i>count</i> 를 최신화 한다.

6

Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A



에러 메시지를 생성하는 경우가 발생한다면 2141. Define Essential Use Cases
7.Update Stock 의 Exceptional Courses of Events부분에 에러메시지가 발생함을 명시해 주셨으면 합니다.

기타 의문



Q. 선결제 코드는 어떻게 관리되나요?

A. 특정 시점에 한가지 음료에 대해서 선결제 코드가 여러개가 있는 경우도 존재하며 선결제 코드는 배열에 들어가 있습니다.



Q. User는 1명으로 가정하나요, 여러 명으로 가정하나요?

A. User는 여러명이며 식별할 필요는 없습니다.



Q. PFR에는 VM당 음료의 종류의 개수가 7개로 고정인 것 같은데 B1팀은 유동적으로 음료의 종류가 변할 수 있도록 할 것인가요?

A. 네 음료의 종류는 변합니다. 판매되는 음료의 종류의 가짓수가 7개 이하인 경우 새로운 음료가 추가될 수 있습니다.



Q. VM ID 관리는 어떻게 하시나요?

A. 현재 로컬 자판기의 ID는 0으로 고정되었고 VM ID가 IP주소를 대체합니다.



Q. 네트워크 통신 라이브러리를 사용해서 실제로 분산 시스템을 만들 것인가요, 아니면 네트워크 통신 없이 분산 시스템처럼 보이는 단일 프로그램을 만들 것인가요?

A. 네트워크 통신을 사용하지 않습니다.



Q. 자판기는 총 몇 대인가요?

A. 총 5대입니다.



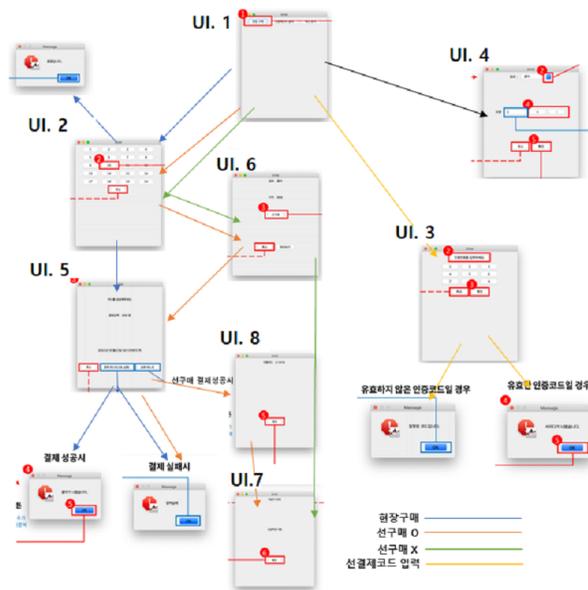
Q. OOPT 1000 FR Test Case에서 어떤경우 positive/negative 가 되나요?

A. use case의 목적에 맞으면 positive, 아니면 negative test case 가 됩니다

Test Stub

가상 Stub(매크로) 제작 방법

- 다음의 2가지 input event를 자동화 시키는 프로그램 작성
 1. 마우스로 버튼 클릭
 2. 콤보박스에서 마우스로 특정 항목 선택
- 결과적으로 다음 5가지 flow가 자동화 됨
 1. 파랑색 화살표에 해당하는 flow (현장구매)
 2. 빨간색 화살표에 해당하는 flow (선구매)
 3. 노란색 화살표에 해당하는 flow (선구매코드입력)
 4. 아래 그림에는 없는 flow (품질)
 5. 아래 그림에는 없는 flow (재고 관리)



실행 방법

1. 현장 구매 버튼
 - 자동화 대상 : 현장 구매 / 선구매 / 품질
 - 과정
 - 'Item 번호' input 필드에 1~20 사이의 값 입력

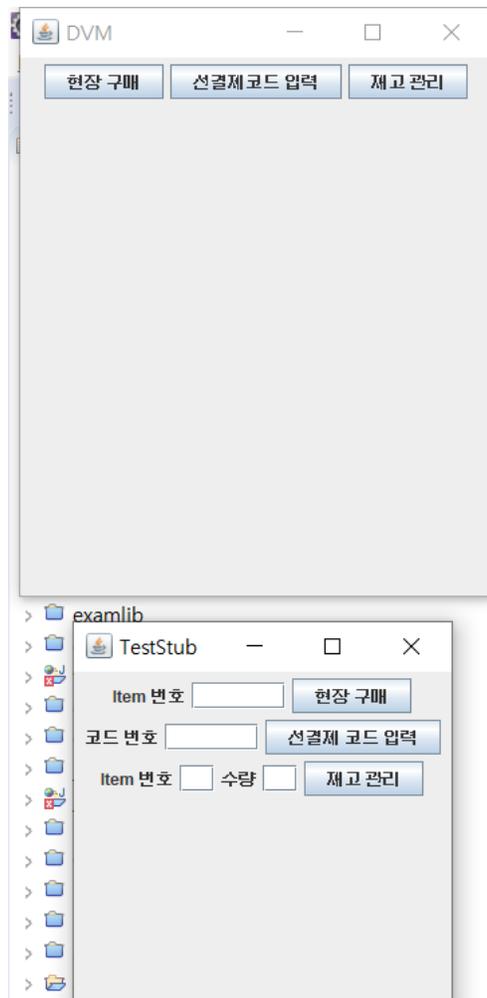
- '현장 구매 버튼' 클릭

2. 선결제 코드 입력

- 자동화 대상 : 선결제 코드 입력
- 과정
 - '코드 번호' input 필드에 선결제 코드 입력
 - '선결제 코드 입력' 버튼 클릭

3. 재고 관리

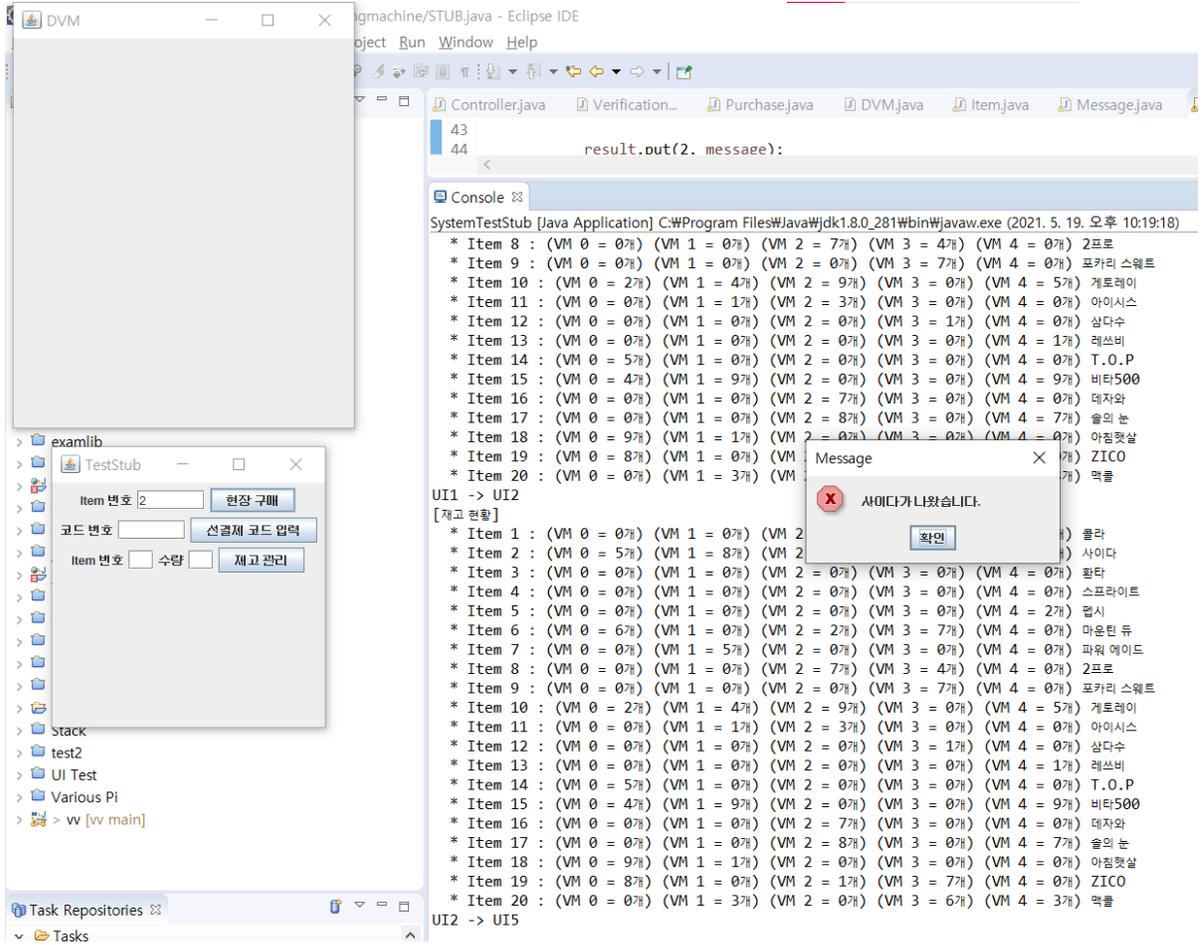
- 자동화 대상 : 재고 관리
- 과정
 - 'Item 번호' input 필드에 1~20 사이의 값 입력
 - '수량' input 필드에 정수 입력
 - '재고 관리' 버튼 클릭



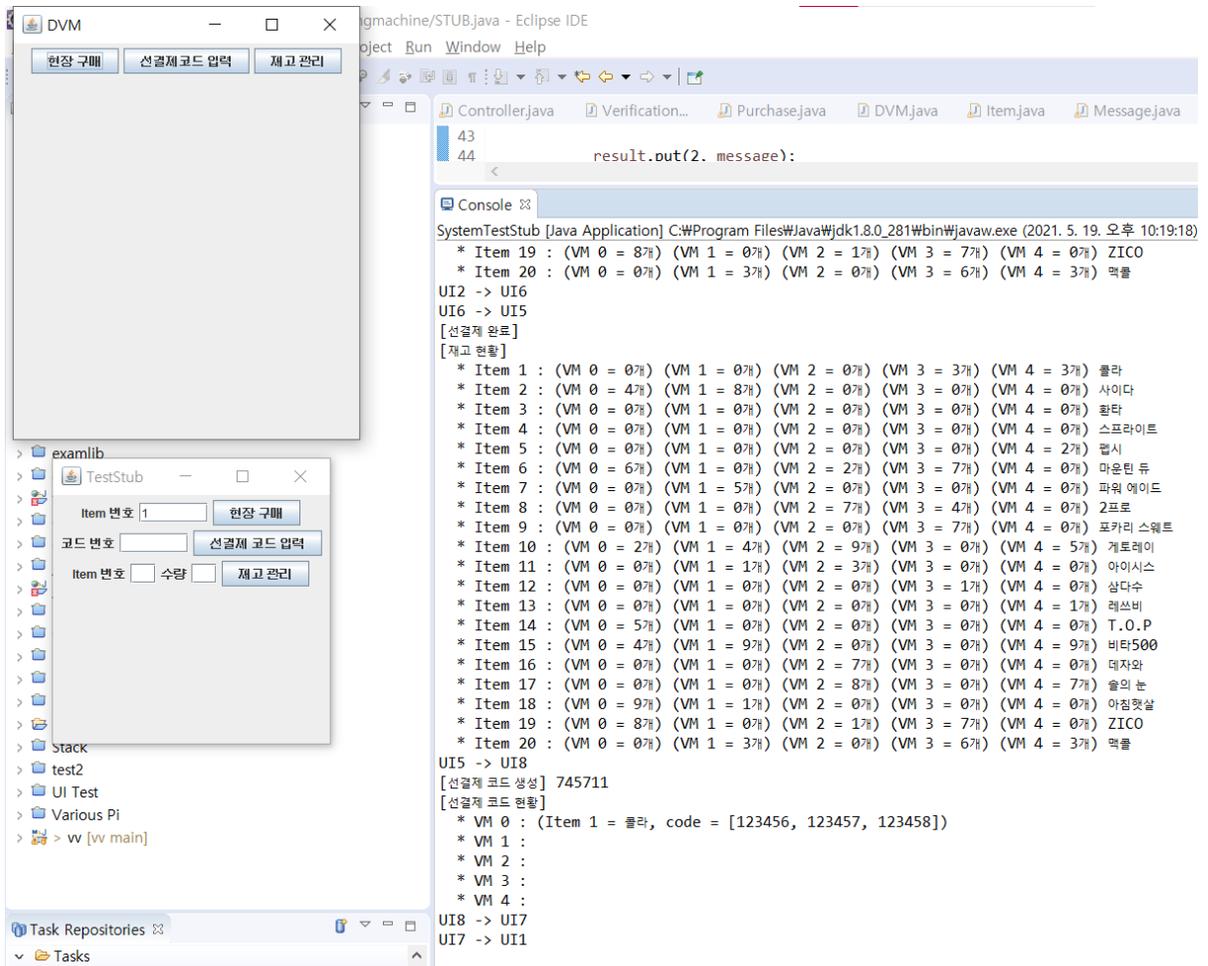
실제 실행 모습

1. 현장 구매 버튼

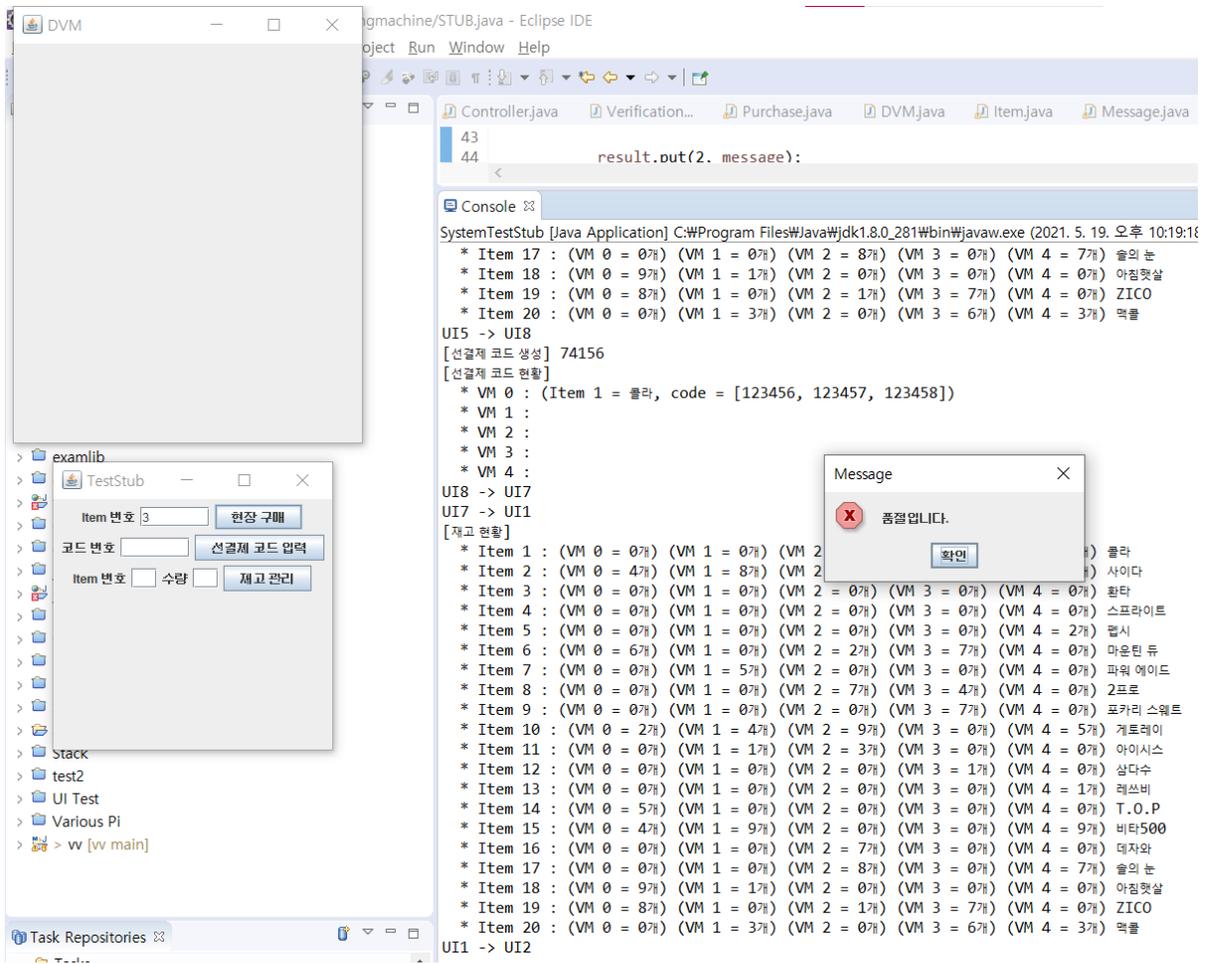
1) 현장 구매



2) 선구매

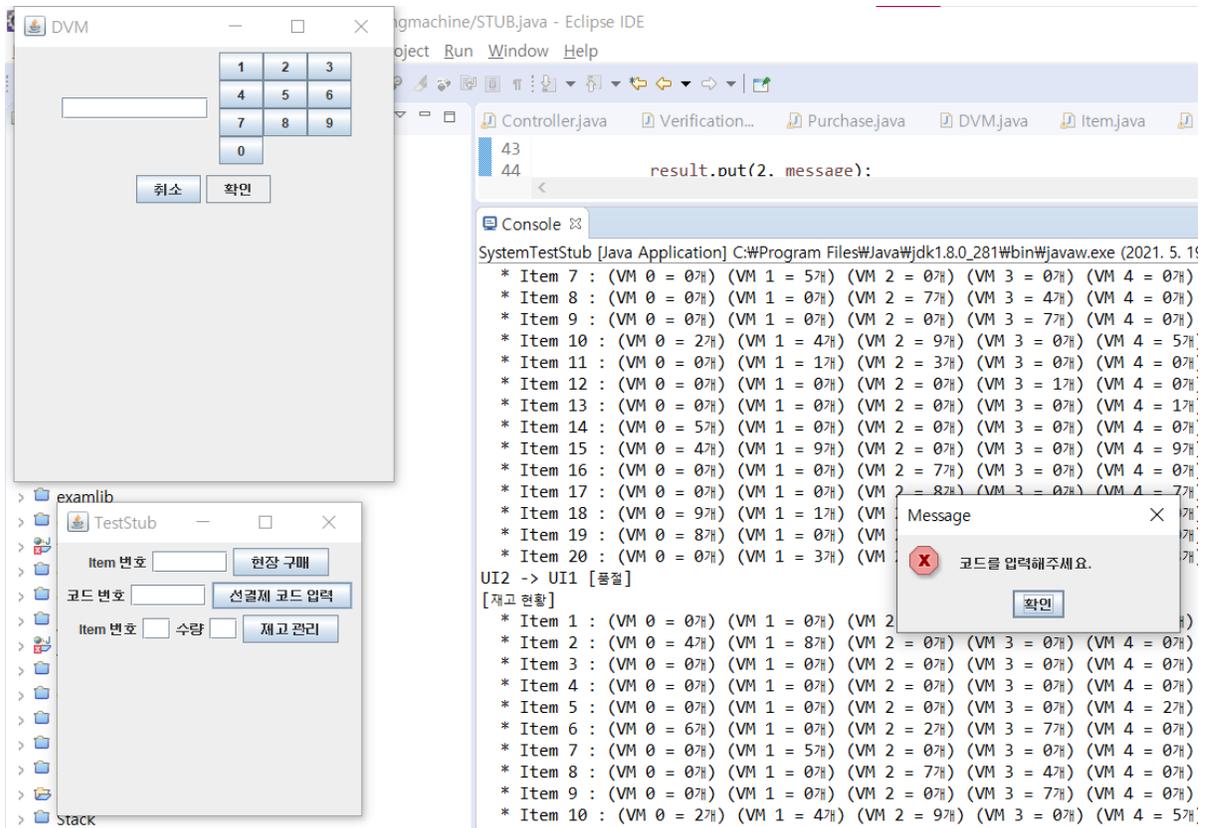


3) 품질

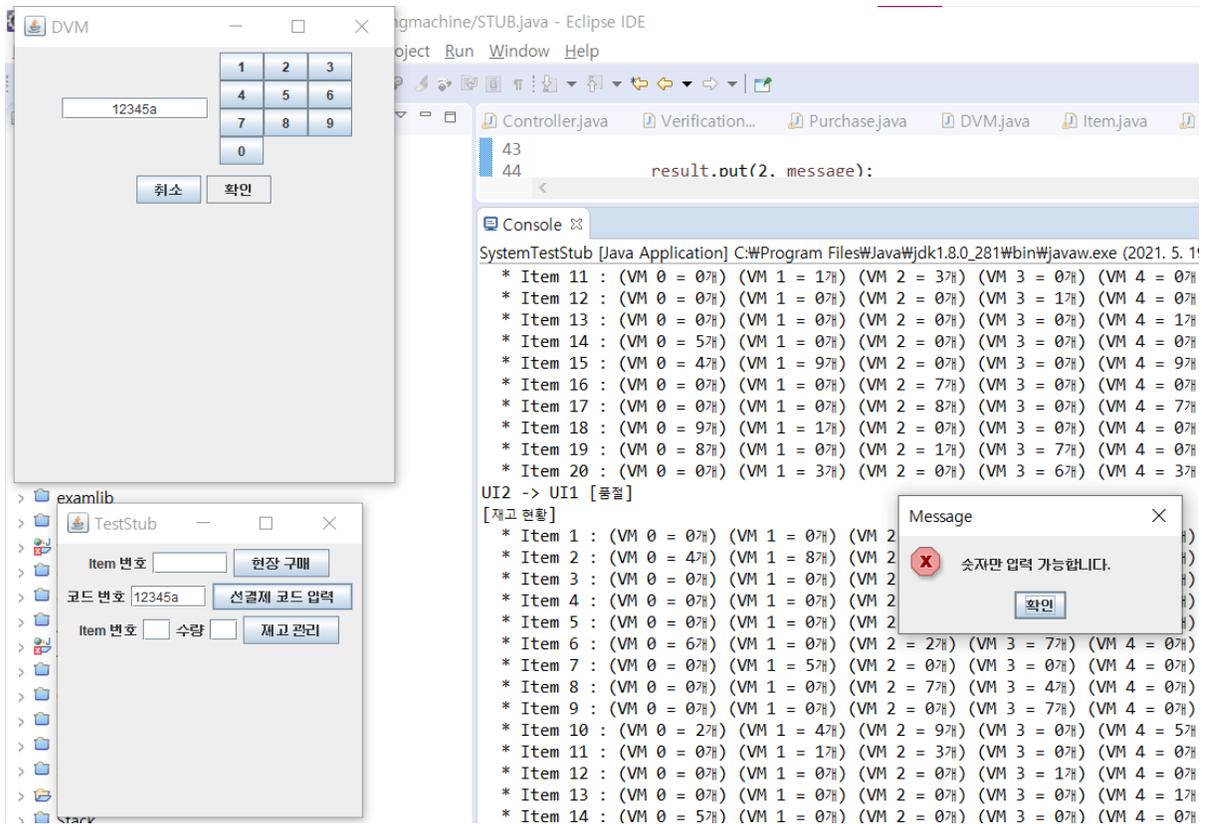


2. 선택제 코드 입력

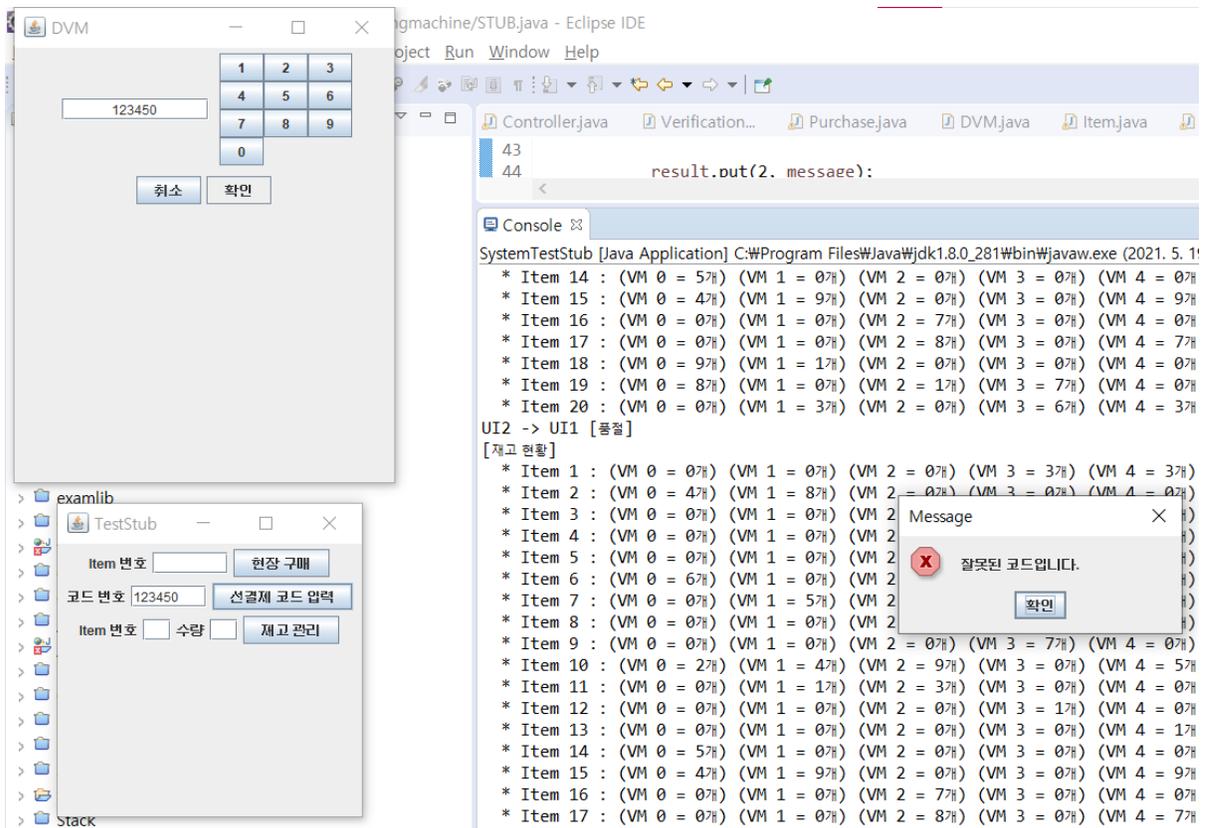
1) 빈칸 입력



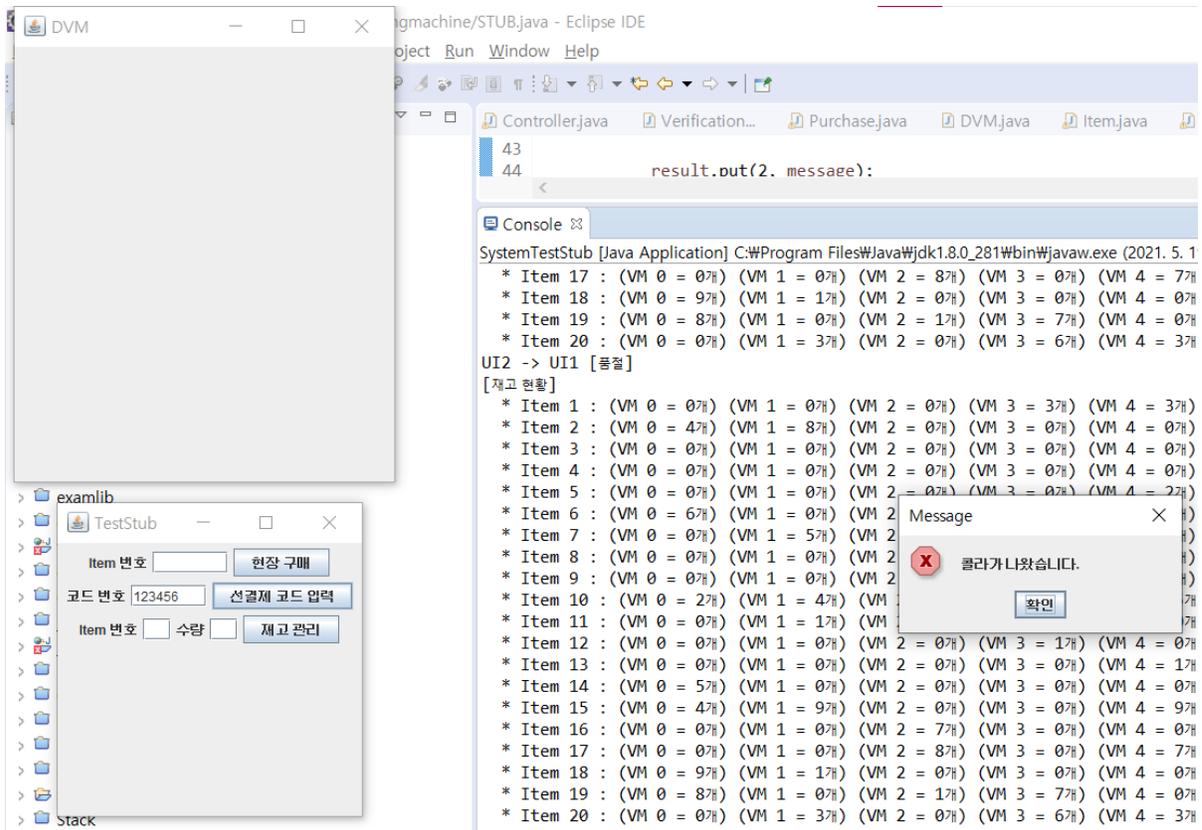
2) 숫자 외의 문자 입력



3) 유효하지 않은 코드 입력

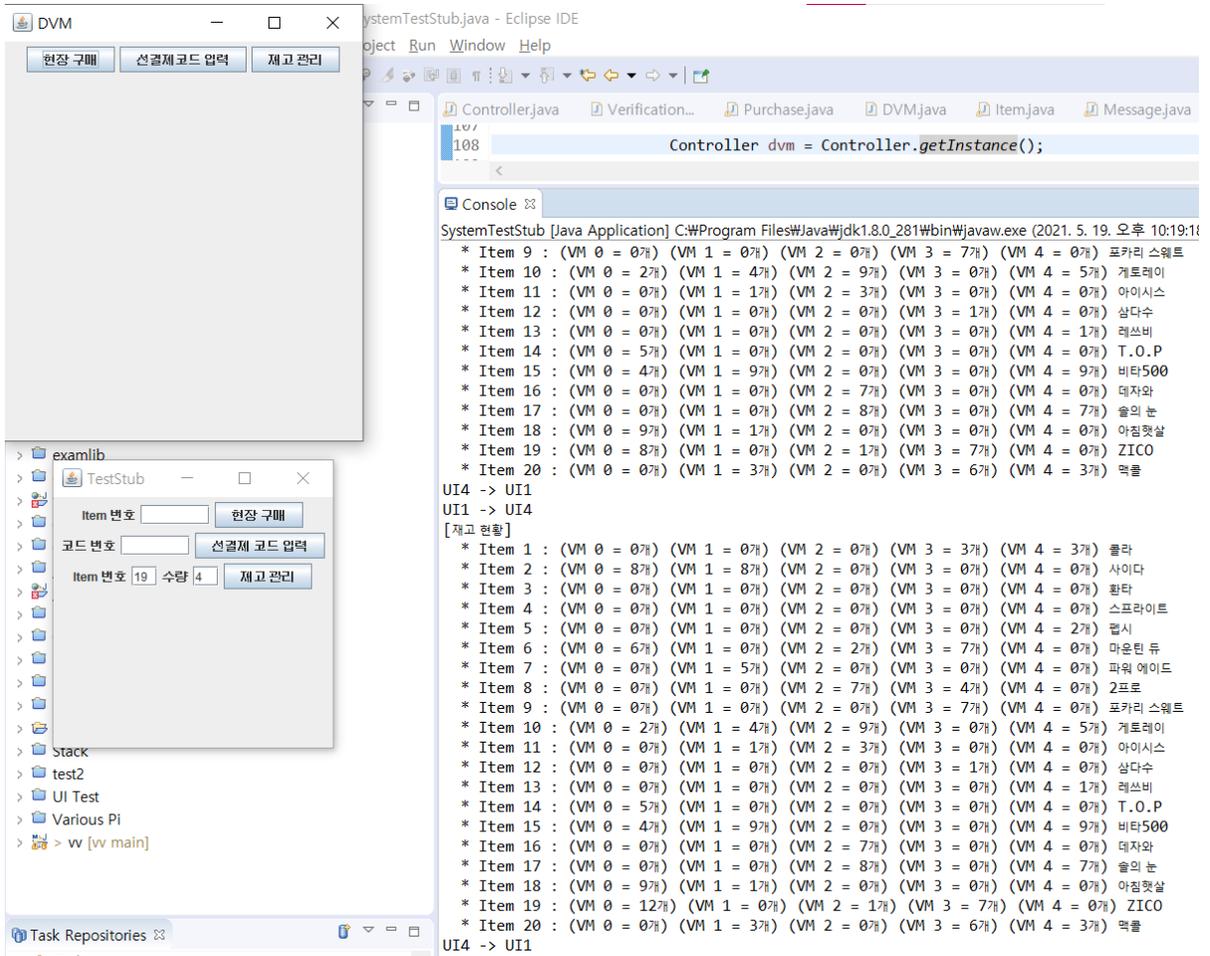


4) 유효한 코드 입력

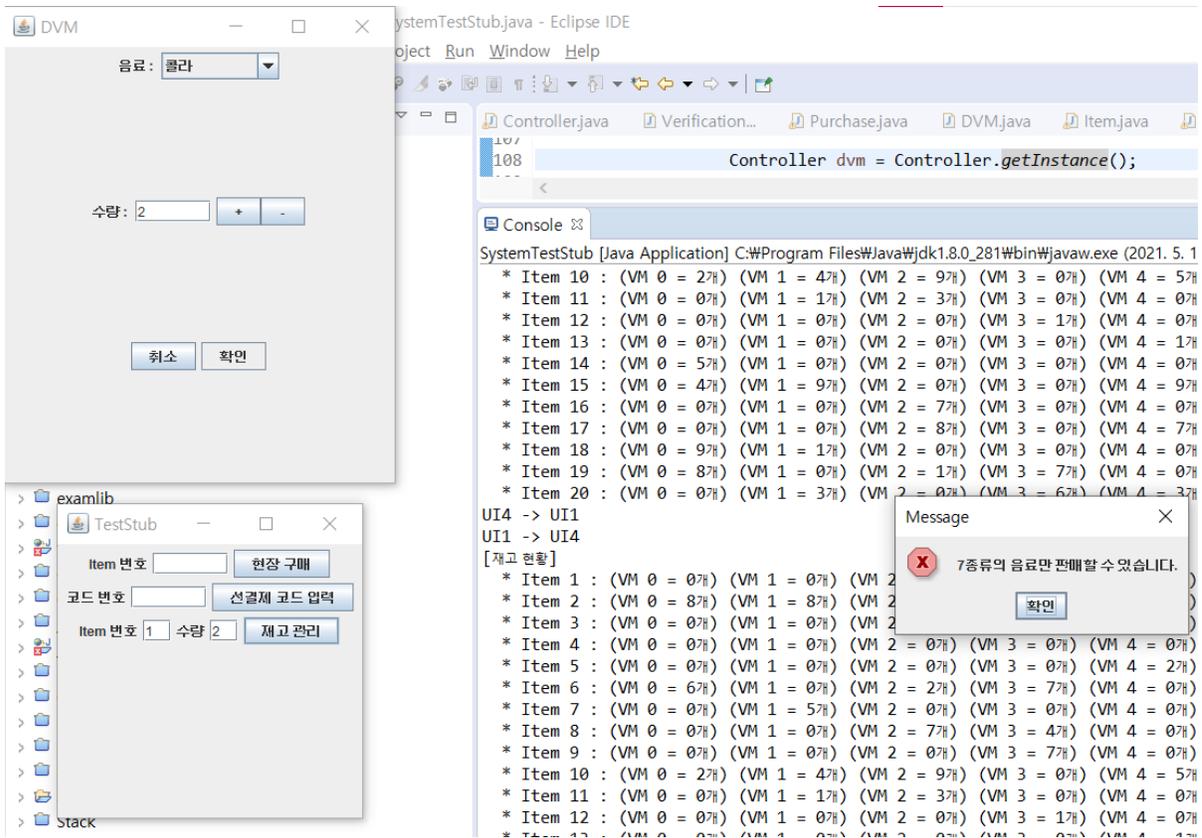


3. 재고 관리

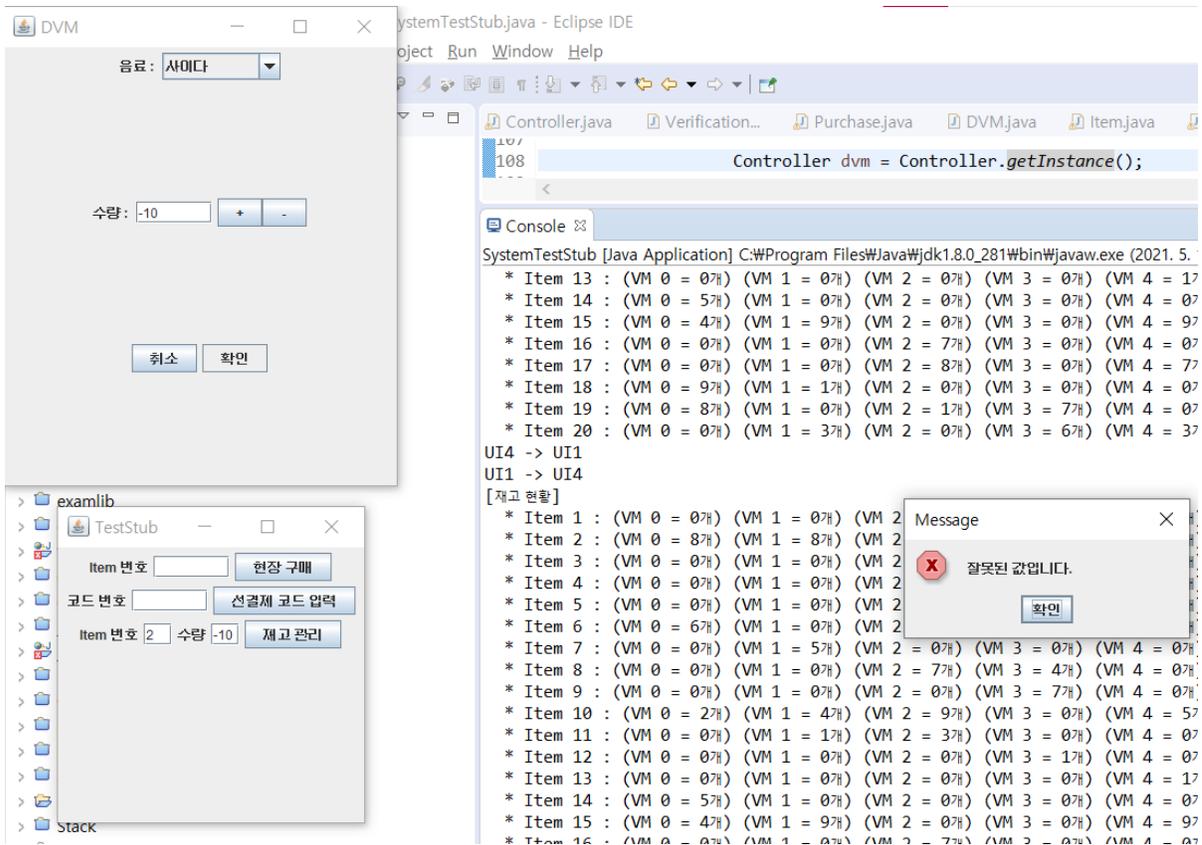
1) 정상 처리



2) 예상 음료 종류가 7가지를 초과하는 경우



3) 예상 재고가 음수가 되는 경우



매크로의 매크로로 테스트하기 어려운 이유

- 매크로의 매크로란?
 - 현재 만든 매크로는 버튼 3개를 직접 마우스로 클릭해야 함
 - 매크로의 매크로는 그 버튼 3개의 클릭마저도 자동화 함
 - 아래 1번과 2번 문제는 시간과 노력을 투자하면 해결 가능하나 3번과 4번 문제는 그렇지 않음
1. 프로그램 시작시마다 재고가 랜덤으로 생김
 - 최초의 재고를 백업해놓고 test routine이 한 번 끝날 때마다 재고를 초기화하여 재수행하는 코드를 작성해야 함
 - 그런 코드를 작성하더라도, 중간에 에러가 발생하여 프로그램이 종료되면 처음부터 다시 테스트해야 함
 2. 매크로를 통해서 자동으로 리포트 생성하는 코드 작성 필요
 - fail만 log (input / actual output / expected output / environment)
 3. 현장구매 하나만 가지고도 item out 순서에 따른 조합이 매우 많음
 - 예를 들어 특정 샘플에 대해 조합의 개수를 계산해보면

- 15 x 4 x 8 x 9 x 4 x 7 x 19 x 5 x 5 x 21 x 8 x 9 x 11 x 18 x 12 x 12 x 9 x 7
- 즉, 10,403,133,021,388,800 개의 조합을 테스트해야 함
- 버튼을 클릭하여 연산이 수행되고 페이지가 넘어가는 delay를 무시한다고 해도 시간적으로 테스트 불가능
- 샘플을 단순화 시켜서 item당 재고가 최대 2개만 가능하다고 해도 1,048,576개의 조합 생성

4. 현장구매/선결제코드입력/재고관리의 수행 순서가 섞이는 경우

- 그런데 그 1,048,576개의 조합이 '선결제코드입력'과 '재고관리'와 또 다시 조합을 이루게 되면, 역시 테스트 불가능한 개수가 나옴

TestStub 소스 코드

```
package ku.ooad.b1.core;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Set;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

import ku.ooad.b1.vendingmachine.Controller;
import ku.ooad.b1.vendingmachine.Item;
import ku.ooad.b1.vendingmachine.STUB;
import ku.ooad.b1.vendingmachine.VerificationCode;

public class SystemTestStub {
    public static void main(String[] args) {
        Controller dvm = Controller.getInstance();
        dvm.initialize();
        TestFrame testFrame = new TestFrame();
    }

    static class TestFrame extends JFrame{
        void stockLog() {
            Controller dvm = Controller.getInstance();
            System.out.println("[재고 현황]");

            List<Item> itemList = dvm.getItems();
            for(int i=0; i<itemList.size(); i++) {
                System.out.print(" * Item "+ (i+1) +" : ");
                Item item = itemList.get(i);
                HashMap<Integer, Integer> count = item.count;
                Set<Integer> keys = count.keySet();
                Iterator<Integer> iter = keys.iterator();
                while(iter.hasNext()) {
```

```

        Integer key = iter.next();
        System.out.print("VM "+ key + " = " +count.get(key)+"개) ");
    }
    System.out.print(Item.names[i]);
    System.out.println();
}
}

void codeLog() {
    // VM0가 갖고 있는 codeList
    System.out.println("[선결제 코드 현황]");
    System.out.print(" * VM 0 : ");
    HashMap<Integer, ArrayList<Integer>> codeList = VerificationCode.getInstance().codeList;
    Set<Integer> keys = codeList.keySet();
    Iterator<Integer> iter = keys.iterator();
    while(iter.hasNext()) {
        Integer key = iter.next();
        System.out.print("Item "+ (key+1) +" = "+Item.names[key]+", code = "+codeList.get(key)+" ");
    }
    System.out.println();

    // VM1 ~ Vm4가 갖고 있는 codeList
    Controller dvm = Controller.getInstance();
    List<STUB> stubList = dvm.getStubs();
    for(int i=0; i<stubList.size(); i++) {
        System.out.print(" * VM "+ (i+1) +" : ");
        STUB stub = stubList.get(i);
        VerificationCode codeManager = stub.getCodeManager();
        HashMap<Integer, ArrayList<Integer>> STUBcodeList = codeManager.codeList;
        Set<Integer> KEYS = STUBcodeList.keySet();
        Iterator<Integer> keyIter = KEYS.iterator();
        while(iter.hasNext()) {
            Integer key = keyIter.next();
            System.out.print("Item "+ (key+1) +" = "+Item.names[key]+", code = "+STUBcodeList.get(key)+" ");
        }
        System.out.println();
    }
}

TestFrame() {
    this.setTitle("TestStub");
    this.setBounds(40,460,300,300);
    this.setVisible(true);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JPanel panel = new JPanel();
    this.add(panel);

    JLabel label1 = new JLabel("Item 번호");
    panel.add(label1);

    JTextField itemInputField = new JTextField("", 6);
    panel.add(itemInputField);

    JButton macroBtn1 = new JButton("");
    macroBtn1.setVisible(true);
    macroBtn1.setText("현장 구매");
    macroBtn1.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e) {
            stockLog();

            String inputText1 = itemInputField.getText();

```

```

if(inputText1.length()==0) return; // field가 빈 칸이면 수행 안 함

Controller dvm = Controller.getInstance();

JPanel initialPagePanel = (JPanel) dvm.curPanel;
JButton localPurchaseButton = (JButton) initialPagePanel.getComponent(0);
localPurchaseButton.doClick(); // UI1 -> UI2
System.out.println("UI1 -> UI2");

int itemNumber = Integer.parseInt(inputText1);
JPanel localPurchasePanel = (JPanel) dvm.curPanel.getComponent(0);
JButton item1 = (JButton) localPurchasePanel.getComponent(itemNumber - 1);
item1.doClick(); // UI2 분기
stockLog();

if (dvm.cur_stockOK == 0) { // UI2 -> UI5
    System.out.println("UI2 -> UI5");

    JPanel purchasePanel = (JPanel) dvm.curPanel.getComponent(3);
    JButton purchaseButton = (JButton) purchasePanel.getComponent(1);
    purchaseButton.doClick(); // UI5 -> UI1
    System.out.println("[현장 구매 완료]");
    stockLog();
    System.out.println("UI5 -> UI1");

    System.out.println("UI2 -> UI5");
} else if (dvm.cur_stockOK == 1) { // UI2 -> UI6
    System.out.println("UI2 -> UI6");
    JPanel advancePurchasePanel = (JPanel) dvm.curPanel.getComponent(1);
    JButton advancePurchaseButton = (JButton) advancePurchasePanel.getComponent(0);
    advancePurchaseButton.doClick(); // UI6 -> UI5
    System.out.println("UI6 -> UI5");

    JPanel purchasePanel = (JPanel) dvm.curPanel.getComponent(3);
    JButton purchaseButton = (JButton) purchasePanel.getComponent(1);
    purchaseButton.doClick(); // UI5 -> UI8
    System.out.println("[선결제 완료]");
    stockLog();
    System.out.println("UI5 -> UI8");

    // 선결제 코드 log
    JTextField codeField = (JTextField) ((JPanel) dvm.curPanel.getComponent(0)).getComponent(1);
    System.out.println("[선결제 코드 생성] "+codeField.getText()); // 방금 생성된 코드
    codeLog(); // 전체 자판기 코드 목록

    JPanel codePanel = (JPanel) dvm.curPanel.getComponent(2);
    JButton codeConfirmButton = (JButton) codePanel.getComponent(0);
    codeConfirmButton.doClick(); // UI8 -> UI7
    System.out.println("UI8 -> UI7");

    JPanel locationPanel = (JPanel) dvm.curPanel.getComponent(2);
    JButton locationButton = (JButton) locationPanel.getComponent(0);
    locationButton.doClick(); // UI7 -> UI1
    System.out.println("UI7 -> UI1");
}
else {
    System.out.println("UI2 -> UI1 [품질]");
    stockLog();
}
}
});
panel.add(macroBtn1);

```

```

JLabel label2 = new JLabel("코드 번호");
panel.add(label2);

JTextField codeInput = new JTextField("", 6);
panel.add(codeInput);

JButton macroBtn2 = new JButton("");
macroBtn2.setVisible(true);
macroBtn2.setText("선결제 코드 입력");
macroBtn2.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e) {
        String inputText = codeInput.getText();

        Controller dvm = Controller.getInstance();

        JPanel initialPagePanel = (JPanel) dvm.curPanel;
        JButton codeInputButton = (JButton) initialPagePanel.getComponent(1);
        codeInputButton.doClick(); // UI1 -> UI3
        System.out.println("UI1 -> UI3");

        // 코드 입력
        JPanel codeInputPanel = (JPanel) dvm.curPanel.getComponent(0);
        JTextField codeInputField = (JTextField) codeInputPanel.getComponent(0);
        codeInputField.setText(inputText);

        // 확인 버튼 클릭
        JPanel codeConfirmPanel = (JPanel) dvm.curPanel.getComponent(2);
        JButton confirmButton = (JButton) codeConfirmPanel.getComponent(1);
        confirmButton.doClick();

        if(inputText.equals(""))
            || dvm.curCodeItem <= -1
            || dvm.curError) { // 실패 시 UI1으로 자동 이동 (테스트 편의를 위해 만듦)
            // 취소 버튼 클릭
            JButton cancelButton = (JButton) codeConfirmPanel.getComponent(0);
            cancelButton.doClick();
        }
        if(inputText.equals("")) System.out.println("[미입력]");
        else if(dvm.curCodeItem <= -1) System.out.println("[존재하지 않는 코드]");
        else if(dvm.curError) System.out.println("[숫자가 아닌 문자 입력]");
        else {
            stockLog();
            codeLog();
        }
        System.out.println("UI3 -> UI1");
    }
});
panel.add(macroBtn2);

JLabel label3 = new JLabel("Item 번호");
panel.add(label3);

JTextField sortInput = new JTextField("", 2);
panel.add(sortInput);

JLabel label4 = new JLabel("수량");
panel.add(label4);

JTextField countInput = new JTextField("", 2);
panel.add(countInput);

```

```

JButton macroBtn3 = new JButton("");
macroBtn3.setVisible(true);
macroBtn3.setText("재고 관리");
macroBtn3.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e) {
        String inputText1 = sortInput.getText();
        String inputText2 = countInput.getText();
        if(inputText1.length()==0 || inputText2.length()==0) return; // field가 빈 칸이면 수행 안 함

        Controller dvm = Controller.getInstance();

        JPanel initialPagePanel = (JPanel) dvm.curPanel;
        JButton stockManagementButton = (JButton) initialPagePanel.getComponent(2);
        stockManagementButton.doClick(); // UI1 -> UI4
        System.out.println("UI1 -> UI4");

        // 종류, 수량 입력
        JPanel sortPanel = (JPanel) dvm.curPanel.getComponent(0);
        JComboBox sortCombobox = (JComboBox) sortPanel.getComponent(1);
        sortCombobox.setSelectedIndex(Integer.valueOf(inputText1)-1);
        JPanel countPanel = (JPanel) dvm.curPanel.getComponent(1);
        JTextField countField = (JTextField) countPanel.getComponent(1);
        countField.setText(inputText2);

        // 확인 버튼 클릭
        JPanel confirmPanel = (JPanel) dvm.curPanel.getComponent(2);
        JButton confirmButton = (JButton) confirmPanel.getComponent(1);
        confirmButton.doClick();

        if(dvm.curCount < 0 || dvm.curCnt > 7) { // 실패 시 UI1으로 자동 이동 (테스트 편의를 위해 만들)
            // 취소 버튼 클릭
            JButton cancelButton = (JButton) confirmPanel.getComponent(0);
            cancelButton.doClick();
        }
        if(dvm.curCount < 0) System.out.println("[수량 음수]");
        else if(dvm.curCnt > 7) System.out.println("[종류 초과]");
        else stockLog();
        System.out.println("UI4 -> UI1");
    }
});
panel.add(macroBtn3);

panel.setVisible(true);
panel.revalidate();
}
}
}

```

대표적인 시나리오들에 대한 log



[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 5개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스위트

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 5개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스위트

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈

- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI1 → UI2

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 5개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스위트

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI2 → UI5

[현장 구매 완료]

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스위트

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수

- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI5 → UI1

UI2 → UI5

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI1 → UI2

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로

* Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨
트

* Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이

* Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스

* Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수

* Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비

* Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P

* Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500

* Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와

* Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈

* Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살

* Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO

* Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI2 → UI6

UI6 → UI5

[선결제 완료]

[재고 현황]

* Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라

* Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다

* Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타

* Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트

* Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시

* Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀

* Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드

* Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로

* Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨

트

* Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이

* Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스

* Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수

* Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비

* Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P

* Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500

* Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와

* Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈

* Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살

* Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO

* Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI5 → UI8

[선결제 코드 생성] 745711

[선결제 코드 현황]

* VM 0 : (Item 1 = 콜라, code = [123456, 123457, 123458])

* VM 1 :

* VM 2 :

* VM 3 :

* VM 4 :

UI8 → UI7

UI7 → UI1

[재고 현황]

* Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라

* Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다

* Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타

* Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트

* Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펍시

* Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀

* Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드

* Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로

* Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨

트

* Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이

* Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스

* Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수

* Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비

* Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P

* Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500

* Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와

* Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈

* Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살

* Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO

* Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI1 → UI2

[재고 현황]

* Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라

* Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다

* Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타

* Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트

* Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펍시

* Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀

* Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드

* Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로

* Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨

트

* Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이

* Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스

- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI2 → UI6

UI6 → UI5

[선결제 완료]

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펩시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 케토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI5 → UI8

[선결제 코드 생성] 74156

[선결제 코드 현황]

- * VM 0 : (Item 1 = 콜라, code = [123456, 123457, 123458])
- * VM 1 :
- * VM 2 :
- * VM 3 :
- * VM 4 :

UI8 → UI7

UI7 → UI1

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스위트

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI1 → UI2

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스위트

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500

- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI2 → UI1 [품질]

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스위트

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI1 → UI3

[미입력]

UI3 → UI1

UI1 → UI3

[숫자가 아닌 문자 입력]

UI3 → UI1

UI1 → UI3

[존재하지 않는 코드]

UI3 → UI1

UI1 → UI3

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 4개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타

- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펍시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

[선결제 코드 현황]

- * VM 0 : (Item 1 = 콜라, code = [123457, 123458])
- * VM 1 :
- * VM 2 :
- * VM 3 :
- * VM 4 :

UI3 → UI1

UI1 → UI4

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 8개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펍시
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P

- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 8개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI4 → UI1

UI1 → UI4

[재고 현황]

- * Item 1 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 3개) (VM 4 = 3개) 콜라
- * Item 2 : (VM 0 = 8개) (VM 1 = 8개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 사이다
- * Item 3 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 환타
- * Item 4 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 스프라이트
- * Item 5 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 2개) 펄스
- * Item 6 : (VM 0 = 6개) (VM 1 = 0개) (VM 2 = 2개) (VM 3 = 7개) (VM 4 = 0개) 마운틴 듀
- * Item 7 : (VM 0 = 0개) (VM 1 = 5개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 파워 에이드
- * Item 8 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 4개) (VM 4 = 0개) 2프로
- * Item 9 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 7개) (VM 4 = 0개) 포카리 스웨

트

- * Item 10 : (VM 0 = 2개) (VM 1 = 4개) (VM 2 = 9개) (VM 3 = 0개) (VM 4 = 5개) 게토레이
- * Item 11 : (VM 0 = 0개) (VM 1 = 1개) (VM 2 = 3개) (VM 3 = 0개) (VM 4 = 0개) 아이시스
- * Item 12 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 1개) (VM 4 = 0개) 삼다수
- * Item 13 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 1개) 레쓰비
- * Item 14 : (VM 0 = 5개) (VM 1 = 0개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) T.O.P
- * Item 15 : (VM 0 = 4개) (VM 1 = 9개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 9개) 비타500
- * Item 16 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 7개) (VM 3 = 0개) (VM 4 = 0개) 데자와
- * Item 17 : (VM 0 = 0개) (VM 1 = 0개) (VM 2 = 8개) (VM 3 = 0개) (VM 4 = 7개) 솔의 눈
- * Item 18 : (VM 0 = 9개) (VM 1 = 1개) (VM 2 = 0개) (VM 3 = 0개) (VM 4 = 0개) 아침햇살
- * Item 19 : (VM 0 = 12개) (VM 1 = 0개) (VM 2 = 1개) (VM 3 = 7개) (VM 4 = 0개) ZICO
- * Item 20 : (VM 0 = 0개) (VM 1 = 3개) (VM 2 = 0개) (VM 3 = 6개) (VM 4 = 3개) 맥콜

UI4 → UI1

UI1 → UI4

[종류 초과]

UI4 → UI1

UI1 → UI4

[수량 음수]

UI4 → UI1

CPT/BFT 통합본

System Testing

1. Category-Partition Testing(CPT)

1.1 Categorize & Property & Constraints

 Group	 Category	 Value	 Constraint [Property prop]
<u>Environments</u>	Current UI	initial page.	[property ui1]
<u>Environments</u>	Current UI	item select.	[property ui2]
<u>Environments</u>	Current UI	input verification code	[property ui3]
<u>Environments</u>	Current UI	modify stock	[property ui4]
<u>Environments</u>	Current UI	payment	[property ui5]
<u>Environments</u>	Current UI	inform prepurchase	[property ui6]
<u>Environments</u>	Current UI	inform location	[property ui7]
<u>Environments</u>	Current UI	inform verification code	[property ui8]
<u>Environments</u>	Mode	purchase	[property mode1] [if ui5]
<u>Environments</u>	Mode	prepurchase	[property mode2] [if ui5]
<u>Environments</u>	is Advance	false	[if ui5]
<u>Environments</u>	is Advance	true	[if ui5]
<u>Environments</u>	selected item stock	local stock zero and remote stock zero	[if ui2]
<u>Environments</u>	selected item stock	local stock zero and valid remote stock	[if ui2]
<u>Environments</u>	selected item stock	valid local stock	[if ui2]
<u>Environments</u>	expected changed local stock	less than zero	[if ui4]

<u>Aa</u> Group	 Category	 Value	 Constraint [Property prop]
<u>Environments</u>	expected changed local stock	more than zero	[if ui4]
<u>Environments</u>	expected changed local item type	less than seven	[if ui4]
<u>Environments</u>	expected changed local item type	more than seven	[if ui4]
<u>Parameters</u>	Page move button	initial page-purchase button.	[if ui1]
<u>Parameters</u>	Page move button	initial page-prepurcahse button.	[if ui1]
<u>Parameters</u>	Page move button	initial page-stock management button	[if ui1]
<u>Parameters</u>	Page move button	select item button	[if ui2]
<u>Parameters</u>	Page move button	cancel button	[if ui2 ui3 ui4 ui5 ui6]
<u>Parameters</u>	Page move button	confirm button	[if ui3 ui4 ui7 ui8]
<u>Parameters</u>	Page move button	location button	[if ui6]
<u>Parameters</u>	Page move button	1% payment fail	[if ui5]
<u>Parameters</u>	Page move button	100% payment fail	[if ui5]
<u>Parameters</u>	Manage Stock	select item zero	[if ui4]
<u>Parameters</u>	Manage Stock	select item less than zero	[if ui4]
<u>Parameters</u>	Manage Stock	select item more than zero	[if ui4]
<u>Parameters</u>	Verification Code	valid verification code	[if ui3]
<u>Parameters</u>	Verification Code	invalud verification code	[if ui3]

1.2 Description

<u>Aa</u> Values	 Description
<u>initial_page</u>	현장구매/선결제코드입력/재고관리 버튼이 있는 첫화면(UI1)인 경우

<u>Aa</u> Values	Description
<u>item select.</u>	구매할 음료를 선택하는 음료선택 화면인 경우(UI2)
<u>input verification code</u>	선구매 진행 후 발급받은 인증번호를 입력하는 화면인 경우(UI3)
<u>modify stock</u>	음료를 선택하고 해당 음료의 재고를 변경하는 화면인 경우(UI4)
<u>payment</u>	카드를 입력하라는 안내글과 함께 결제를 진행하는 화면인 경우(UI5)
<u>inform prepurchase</u>	음료 선택후 해당 자판기에 재고가 없어 선구매를 진행하는 화면인 경우 (UI6)
<u>inform location</u>	선구매를 진행하고 결제에 성공후 음료를 판매하는 자판기의 위치를 안내하는 화면인 경우(UI7)
<u>inform verification code</u>	선구매를 진행하고 결제에 성공후 인증번호를 안내받는 화면인 경우(UI8)
<u>purchase</u>	로컬 자판기에 재고가 있어 결제화면(UI5)에서 결제를 진행한다
<u>prepurchase</u>	로컬 자판기에 재고가 없고, 외부 자판기에는 재고가 있어 결제화면(UI5)에서 선결제를 진행한다
<u>false</u>	선구매를 진행하지 않은 경우
<u>true</u>	선구매를 진행한 경우
<u>local stock zero and remote stock zero</u>	현재구매를 진행하고 있는 로컬자판기에 선택한 음료의 재고가 0이고 다른 자판기에도 해당 음료의 재고가 0인경우
<u>local stock zero and valid remote stock</u>	현재구매를 진행하고 있는 로컬자판기에 선택한 음료의 재고가 0이지만 다른 자판기에는 해당 음료의 재고가 1이상인 경우
<u>valid local stock</u>	현재구매를 진행하고 있는 로컬자판기에 선택한 음료의 재고가 1이상인 경우
<u>less than zero</u>	재고관리 화면(UI4)에서 재고 변경 후 해당 음료의 재고가 0미만이 되는 경우
<u>more than zero</u>	재고관리 화면(UI4)에서 재고 변경 후 해당 음료의 재고가 0이상인 되는 경우
<u>less than seven</u>	재고관리 화면(UI4)에서 재고 변경 후 음료의 총 가짓수가 7개 이하인경우
<u>more than seven</u>	재고관리 화면(UI4)에서 재고 변경 후 음료의 총 가짓수가 7개를 초과하는 경우
<u>initial page-purchase button.</u>	초기화면[UI1]에서 현장구매 버튼을 누른다
<u>initial page-prepurcahse button.</u>	초기화면[UI1]에서 선결제코드입력 버튼을 누른다

 Values	 Description
<u>initial page-stock management button</u>	초기화면[UI1]에서 재고관리 버튼을 누른다
<u>select item button</u>	Item select[UI2]에서 음료를 선택한다
<u>cancel button</u>	취소버튼을 누른다
<u>confirm button</u>	확인버튼을 누른다
<u>location button</u>	
<u>1% payment fail</u>	Payment화면[UI5]에서 1% payment fail버튼을 누른다
<u>100% payment fail</u>	Payment화면[UI5]에서 100% payment fail버튼을 누른다
<u>select item zero</u>	재고관리 화면[UI4]에서 재고가 0인 음료를 선택한다
<u>select item less than zero</u>	재고관리 화면[UI4]에서 재고가 음수인 즉, 판매중이지 않은 음료를 선택한다
<u>select item more than zero</u>	재고관리 화면[UI4]에서 재고가 양수인 즉, 현재 판매중인 음료를 선택한다
<u>valid verification code</u>	인증번호를 입력하는 화면[UI3]에서 유효한 인증번호를 입력한다
<u>invalid verification code</u>	인증번호를 입력하는 화면[UI3]에서 유효하지 않은 인증번호를 입력한다

1.3 TextFile for TSL-Generator



```
#  
# Test specification for DVM  
#
```

Environments:

Current UI:

```
initial page.  
    [property ui1]  
item select.    [property ui2]  
input verification code. [property ui3]  
modify stock.    [property ui4]  
payment.        [property ui5]  
inform prepurchase. [property ui6]  
inform location.    [property ui7]  
inform verification code. [property ui8]
```

Mode:

```
purchase.        [property mode1] [if ui5]  
prepurchase.    [property mode2] [if ui5]
```

is Advance:

```
false.          [if ui5]  
true.           [if ui5]
```

selected item stock:

```
local stock zero and remote stock zero.  
    [if ui2]  
local stock zero and valid remote stock.  
    [if ui2]  
valid local stock.    [if ui2]
```

expected changed local stock:

```
less than zero.    [if ui4]  
more than zero.   [if ui4]
```

expected changed local item type:

less than seven. [if ui4]

more than seven. [if ui4]

Parameters:

Page move button:

initial page-purchase button.

[if ui1]

initial page-prepurchase button.

[if ui1]

initial page-stock management button.

[if ui1]

select item button. [if ui2]

cancel button. [if ui2 || ui3 || ui4 || ui5 || ui6]

confirm button. [if ui3 || ui4 || ui7 || ui8]

1% payment fail. [if ui5]

100% payment fail. [if ui5]

location button. [if ui6]

Manage Stock:

select item zero. [if ui4]

select item less than zero. [if ui4]

select item more than zero. [if ui4]

Verification Code:

valid verification code. [if ui3]

invalid verification code. [if ui3]

1.4 Example of Test Data

- **1.4.1 Test Case Specification**



Test Case 1 <single>

Pre-Condition

Current UI : initial page

Input

<n/a>

Expected Output : 프로그램이 실행되어 UI1이 출력됨.



Test Case 2 (Key = 2.0.0.1.0.0.4.0.0.)

Pre-Condition

Current UI : item select

Mode : <n/a>

is Advance : <n/a>

selected item stock : local stock zero and remote stock zero

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : select item button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 남아있는 전체 재고가 없으므로 "품절입니다"라는 안내창이 생성됨.



Test Case 3 (Key = 2.0.0.1.0.0.5.0.0.)

Pre-Condition

Current UI : item select

Mode : <n/a>

is Advance : <n/a>

selected item stock : local stock zero and remote stock zero

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : cancel button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 4 (Key = 2.0.0.2.0.0.4.0.0.)

Pre-Condition

Current UI : item select

Mode : <n/a>

is Advance : <n/a>

selected item stock : local stock zero and valid remote stock

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : select item button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 본체에 재고가 없어 외부 VM에서 구매해야하므로 선결제화면 (UI6)으로 넘어가게 된다.



Test Case 5 (Key = 2.0.0.2.0.0.5.0.0.)

Pre-Condition

Current UI : item select

Mode : <n/a>

is Advance : <n/a>

selected item stock : local stock zero and valid remote stock

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : cancel button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 6 (Key = 2.0.0.3.0.0.4.0.0.)

Pre-Condition

Current UI : item select

Mode : <n/a>

is Advance : <n/a>

selected item stock : valid local stock

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : select item button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 해당 상품의 결제 금액 및 현재 시각과 함께 현장구매 화면(UI5)이 출력된다.



Test Case 7 (Key = 2.0.0.3.0.0.5.0.0.)

Pre-Condition

Current UI : item select
Mode : <n/a>
is Advance : <n/a>
selected item stock : valid local stock
expected changed local stock : <n/a>
expected changed local item type : <n/a>

Input

Page move button : cancel button
Manage Stock : <n/a>
Verification Code : <n/a>

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 8 (Key = 3.2.0.0.0.0.5.0.1.)

Pre-Condition

Current UI : input verification code
Mode : input verification code
is Advance : <n/a>
selected item stock : <n/a>
expected changed local stock : <n/a>
expected changed local item type : <n/a>

Input

Page move button : cancel button
Manage Stock : <n/a>
Verification Code : valid verification code

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 9 (Key = 3.2.0.0.0.0.5.0.2.)

Pre-Condition

Current UI : input verification code

Mode : input verification code

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : cancel button

Manage Stock : <n/a>

Verification Code : invalud verification code

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 10 (Key = 3.2.0.0.0.0.6.0.1.)

Pre-Condition

Current UI : input verification code

Mode : input verification code

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : confirm button

Manage Stock : <n/a>

Verification Code : valid verification code

Expected Output : 유효한 인증코드를 입력했을 경우, 해당 인증코드와 대응되는 "\$상품명 가 나왔습니다."라는 안내창이 출력된다.



Test Case 11 (Key = 3.2.0.0.0.6.0.2.)

Pre-Condition

Current UI : input verification code

Mode : input verificatin code

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : confirm button

Manage Stock : <n/a>

Verification Code : invalud verification code

Expected Output : 잘못된 인증코드를 입력했을 경우, "잘못된 코드입니다."라는 안내창이 출력된다.



Test Case 12 (Key = 4.0.0.0.1.1.5.1.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : less than seven

Input

Page move button : cancel button

Manage Stock : select item zero

Verification Code : <n/a>

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 13 (Key = 4.0.0.0.1.1.5.2.0.)

Pre-Condition

Current UI : modify sotck

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : less than seven

Input

Page move button : cancel button

Manage Stock : select item less than zero

Verification Code : <n/a>

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 14 (Key = 4.0.0.0.1.1.5.3.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : less than seven

Input

Page move button : cancel button

Manage Stock : select item more than zero

Verification Code : <n/a>

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 15 (Key = 4.0.0.0.1.1.6.1.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : less than seven

Input

Page move button : confirm button

Manage Stock : select item zero

Verification Code : <n/a>

Expected Output : 재고가 음수가 될 수는 없으므로 "잘못된 값입니다."라고 안내창이 출력된다.



Test Case 16 (Key = 4.0.0.0.1.1.6.2.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : less than seven

Input

Page move button : confirm button

Manage Stock : select item less than zero

Verification Code : <n/a>

Expected Output : 재고가 음수가 될 수는 없으므로 "잘못된 값입니다."라고 안내창이 출력된다.



Test Case 17 (Key = 4.0.0.0.1.1.6.3.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : less than seven

Input

Page move button : confirm button

Manage Stock : select item more than zero

Verification Code : <n/a>

Expected Output : 재고가 음수가 될 수는 없으므로 "잘못된 값입니다."라고 안내창이 출력된다.



Test Case 18 (Key = 4.0.0.0.1.2.5.1.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : more than seven

Input

Page move button : cancel button

Manage Stock : select item zero

Verification Code : <n/a>

Expected Output : 다시 초기 화면(UI1)으로 돌아간다.



Test Case 19 (Key = 4.0.0.0.1.2.5.2.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : more than seven

Input

Page move button : cancel button

Manage Stock : select item less than zero

Verification Code : <n/a>

Expected Output : 일어날 수 없는 경우이다.



Test Case 20 (Key = 4.0.0.0.1.2.5.3.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : more than seven

Input

Page move button : cancel button

Manage Stock : select item more than zero

Verification Code : <n/a>

Expected Output : 일어날 수 없는 경우이다.



Test Case 21 (Key = 4.0.0.0.1.2.6.1.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : more than seven

Input

Page move button : confirm button

Manage Stock : select item zero

Verification Code : <n/a>

Expected Output : 판매 품목은 7종류를 넘길 수 없으므로 '7종류의 음료만 판매 할 수 있습니다.'라는 안내창이 출력된다.



Test Case 22 (Key = 4.0.0.0.1.2.6.2.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : more than seven

Input

Page move button : confirm button

Manage Stock : select item less than zero

Verification Code : <n/a>

Expected Output : 일어날 수 없는 경우이다.



Test Case 23 (Key = 4.0.0.0.1.2.6.3.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : less than zero

expected changed local item type : more than seven

Input

Page move button : confirm button

Manage Stock : select item more than zero

Verification Code : <n/a>

Expected Output : 일어날 수 없는 경우이다.



Test Case 24 (Key = 4.0.0.0.2.1.5.1.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : less than seven

Input

Page move button : cancel button

Manage Stock : select item zero

Verification Code : <n/a>

Expected Output : 초기화면(UI1)으로 돌아갑니다.



Test Case 25 (Key = 4.0.0.0.2.1.5.2.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : less than seven

Input

Page move button : cancel button

Manage Stock : select item less than zero

Verification Code : <n/a>

Expected Output : 초기화면(UI1)으로 돌아갑니다.



Test Case 26 (Key = 4.0.0.0.2.1.5.3.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : less than seven

Input

Page move button : cancel button

Manage Stock : select item more than zero

Verification Code : <n/a>

Expected Output : 초기화면(UI1)으로 돌아갑니다.



Test Case 27 (Key = 4.0.0.0.2.1.6.1.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : less than seven

Input

Page move button : confirm button

Manage Stock : select item zero

Verification Code : <n/a>

Expected Output : 선택된 음료의 재고가 증가되고, 다시 초기 화면(UI1)으로 돌아갑니다.



Test Case 28 (Key = 4.0.0.0.2.1.6.2.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : less than seven

Input

Page move button : confirm button

Manage Stock : select item less than zero

Verification Code : <n/a>

Expected Output : 음료의 종류가 1개 증가하고, 다시 초기화면(UI1)으로 돌아갑니다.



Test Case 29 (Key = 4.0.0.0.2.1.6.3.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : less than seven

Input

Page move button : confirm button

Manage Stock : select item more than zero

Verification Code : <n/a>

Expected Output : 선택된 음료의 수량이 변경되고, 다시 초기화면(UI)으로 돌아갑니다.



Test Case 30 (Key = 4.0.0.0.2.2.5.1.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : more than seven

Input

Page move button : cancel button

Manage Stock : select item zero

Verification Code : <n/a>

Expected Output : 초기화면(UI)으로 돌아갑니다.



Test Case 31 (Key = 4.0.0.0.2.2.5.2.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : more than seven

Input

Page move button : cancel button

Manage Stock : select item less than zero

Verification Code : <n/a>

Expected Output : 초기화면(UI)으로 돌아갑니다.



Test Case 32 (Key = 4.0.0.0.2.2.5.3.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : more than seven

Input

Page move button : cancel button

Manage Stock : select item more than zero

Verification Code : <n/a>

Expected Output : 초기화면(UI)으로 돌아갑니다.



Test Case 33 (Key = 4.0.0.0.2.2.6.1.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : more than seven

Input

Page move button : confirm button

Manage Stock : select item zero

Verification Code : <n/a>

Expected Output : "7종류의 음료만 판매할 수 있습니다"라는 경고창이 생성됩니다.



Test Case 34 (Key = 4.0.0.0.2.2.6.2.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : more than seven

Input

Page move button : confirm button

Manage Stock : select item less than zero

Verification Code : <n/a>

Expected Output : "7종류의 음료만 판매할 수 있습니다"라는 경고창이 생성됩니다.



Test Case 35 (Key = 4.0.0.0.2.2.6.3.0.)

Pre-Condition

Current UI : modify stock

Mode : <n/a>

is Advance : <n/a>

selected item stock : <n/a>

expected changed local stock : more than zero

expected changed local item type : more than seven

Input

Page move button : confirm button

Manage Stock : select item more than zero

Verification Code : <n/a>

Expected Output : Expected Output : "7종류의 음료만 판매할 수 있습니다"라는 경고창이 생성됩니다.



Test Case 36 (Key = 5.1.1.0.0.0.5.0.0.)

Pre-Condition

Current UI : payment

Mode : purchase

is Advance : false

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : cancel button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 초기화면(UI1)으로 돌아갑니다.



Test Case 37 (Key = 5.1.1.0.0.0.7.0.0.)

Pre-Condition

Current UI : payment

Mode : purchase

is Advance : false

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : 1% payment fail

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : "[음료]가 나왔습니다" 라는 메시지가 생성됩니다.



Test Case 38 (Key = 5.1.1.0.0.0.8.0.0.)

Pre-Condition

Current UI : payment

Mode : purchase

is Advance : false

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : 100% payment fail

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : "결제 실패-잔액 부족.다른 카드를 넣어주세요."라는 경고창이 생성됩니다.



Test Case 39 (Key = 5.1.2.0.0.0.5.0.0.)

Pre-Condition

Current UI : payment

Mode : purchase

is Advance : true

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : cancel button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 초기화면(UI1)으로 돌아갑니다.



Test Case 40 (Key = 5.1.2.0.0.0.7.0.0.)

Pre-Condition

Current UI : payment

Mode : purchase

is Advance : true

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : 1% payment fail

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 인증코드를 안내하는 화면(UI7)으로 전환됩니다.



Test Case 41 (Key = 5.1.2.0.0.0.8.0.0.)

Pre-Condition

Current UI : payment
Mode : purchase
is Advance : true
selected item stock : <n/a>
expected changed local stock : <n/a>
expected changed local item type : <n/a>

Input

Page move button : 100% payment fail
Manage Stock : <n/a>
Verification Code : <n/a>

Expected Output : "결제 실패-잔액 부족.다른 카드를 넣어주세요."라는 경고창이 생성됩니다.



Test Case 42 (Key = 6.0.2.0.0.0.5.0.0.)

Pre-Condition

Current UI : inform prepurchase
Mode : <n/a>
is Advance : true
selected item stock : <n/a>
expected changed local stock : <n/a>
expected changed local item type : <n/a>

Input

Page move button : cancel button
Manage Stock : <n/a>
Verification Code : <n/a>

Expected Output : 초기화면(UI1)으로 돌아갑니다.



Test Case 43 (Key = 7.0.2.0.0.0.6.0.0.)

Pre-Condition

Current UI : inform location

Mode : <n/a>

is Advance : true

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : confirm button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 초기화면(UI1)으로 돌아갑니다.



Test Case 44 (Key = 8.0.2.0.0.0.6.0.0.)

Pre-Condition

Current UI : inform verification code

Mode : <n/a>

is Advance : true

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : confirm button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 위치안내화면(UI7)로 전환됩니다.



Test Case 44 (Key = 8.0.2.0.0.0.6.0.0.)

Pre-Condition

Current UI : inform verification code

Mode : <n/a>

is Advance : true

selected item stock : <n/a>

expected changed local stock : <n/a>

expected changed local item type : <n/a>

Input

Page move button : confirm button

Manage Stock : <n/a>

Verification Code : <n/a>

Expected Output : 위치안내화면(UI7)로 전환됩니다.

• 1.4.2 Test Data Specification



기본적인 입력은 버튼을 통해 이루어지므로 버튼 입력으로 이루어지는 Test Data의 결과는 위 내용과 일치한다.

따라서 버튼을 제외한 특별한 입력값이 주어지는 경우만 추가로 작성한다.



Test Data for Test Case 15

Pre-Condition

1. Current UI is modify stock

2. expected changed local stock < 0

3. expected changed local stock \leq 7

Input

1. 'a' (a < 0)

2. press 'confirm button'

Expected Output

1. 재고수는 음수가 될 수 없으므로 selected item의 count가 변동되지 않고 "잘못 된 값입니다."라고 안내창이 출력됨.



Test Data for Test Case 16

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock < 0
3. expected changed local stock ≤ 7

Input

1. select item 'C' which count less than zero
2. 'a' ($a < 0$)
3. press 'confirm button'

Expected Output

1. 재고수는 음수가 될 수 없으므로 selected item(C)의 count가 변동되지 않고 "잘못된 값입니다."라고 안내창이 출력됨.



Test Data for Test Case 17

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock < 0
3. expected changed local stock ≤ 7

Input

1. select item 'C' which count 'b' ($|b| < |a|$)
2. 'a' ($a < 0$)
3. press 'confirm button'

Expected Output

1. 재고수는 음수가 될 수 없으므로 selected item(C)의 count가 변동되지 않고 "잘못된 값입니다."라고 안내창이 출력됨.



Test Data for Test Case 21

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock < 0
3. expected changed local stock > 7

Input

1. select item 'B' which count 0
2. 'a' ($a > 0$)
3. press 'confirm button'

Expected Output

1. 판매 품목은 7종류를 넘길 수 없으므로 '7종류의 음료만 판매할 수 있습니다.'라는 안내창이 출력된다.



Test Data for Test Case 27

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock ≥ 0
3. expected changed local stock ≤ 7

Input

1. select item 'D' which count 0
2. 'a' ($a > 0$)
3. press 'confirm button'

Expected Output

1. 선택한 품목 D의 재고 개수가 a만큼 증가되고, 다시 초기 화면(UI1)으로 돌아간다.



Test Data for Test Case 28

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock ≥ 0
3. expected changed local stock ≤ 7

Input

1. select item 'E' which count '-1' (because of setting or error)
2. 'a' ($a > 2$)
3. press 'confirm button'

Expected Output

1. 선택한 품목 E의 재고 개수가 a만큼 증가되어 a-1개가 됨. 그리고 다시 초기 화면 (UI1)으로 돌아간다.



Test Data for Test Case 28

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock ≥ 0
3. expected changed local stock ≤ 7

Input

1. select item 'E' which count 'b' ($b > 0$)
2. 'a' ($a > 0$)
3. press 'confirm button'

Expected Output

1. 선택한 품목 E의 재고 개수가 a만큼 증가되어 a+b개가 됨. 그리고 다시 초기 화면 (UI1)으로 돌아간다.



Test Data for Test Case 33

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock ≥ 0
3. expected changed local stock > 7

Input

1. select item 'F' which count '0'
2. 'a' ($a > 0$)
3. press 'confirm button'

Expected Output

1. "7종류의 음료만 판매할 수 있습니다"라는 경고창이 생성됩니다.



Test Data for Test Case 35

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock ≥ 0
3. expected changed local stock > 7

Input

1. select item 'F' which count '-1' (because of setting or error)
2. 'a' ($a > 1$)
3. press 'confirm button'

Expected Output

1. "7종류의 음료만 판매할 수 있습니다"라는 경고창이 생성됩니다.



Test Data for Test Case 36

Pre-Condition

1. Current UI is modify stock
2. expected changed local stock ≥ 0
3. expected changed local stock > 7

Input

1. select item 'F' which count 'b' ($b > 0$)
2. 'a' ($a > 0$)
3. press 'confirm button'

Expected Output

1. "7종류의 음료만 판매할 수 있습니다"라는 경고창이 생성됩니다.

1.5 CPT Result

<u>Aa</u> TestCase	<u>≡</u> Result	<u>≡</u> Description
<u>1</u>	Pass	
<u>2</u>	Pass	
<u>3</u>	Pass	
<u>4</u>	Pass	
<u>5</u>	Pass	
<u>6</u>	Pass	
<u>7</u>	Pass	
<u>8</u>	Pass	
<u>9</u>	Pass	
<u>10</u>	Pass	
<u>11</u>	Pass	
<u>12</u>	Pass	
<u>13</u>	Pass	
<u>14</u>	Pass	
<u>15</u>	Pass	

 TestCase	 Result	 Description
<u>16</u>	Pass	
<u>17</u>	Pass	
<u>18</u>	Pass	
<u>19</u>	-	
<u>20</u>	-	
<u>21</u>	Pass	
<u>22</u>	-	
<u>23</u>	-	
<u>24</u>	Pass	
<u>25</u>	Pass	
<u>26</u>	Pass	
<u>27</u>	Pass	
<u>28</u>	Pass	
<u>29</u>	Pass	
<u>30</u>	Pass	
<u>31</u>	Pass	
<u>32</u>	Pass	
<u>33</u>	Pass	
<u>34</u>	Pass	
<u>35</u>	Pass	
<u>36</u>	Pass	
<u>37</u>	Pass	
<u>38</u>	Pass	
<u>39</u>	Pass	
<u>40</u>	Pass	
<u>41</u>	Pass	
<u>42</u>	Pass	
<u>43</u>	Pass	
<u>44</u>	Pass	

⇒ (40 / 40) = 100% Pass

2. Brute Force Testing(BFT) Report ⇒ (35 / 50) = 70% Pass

Use case	Number	Test case	Test Result
Set up	1-1	시스템이 시작되었을때 다른 DVM의 ID를 받아왔는지 확인	PASS
	1-2	시스템이 시작되었을대 다른 DVM의 재고를 받아왔는지 확인	PASS
	1-3	시스템이 시작되었을 때 자판기의 재고가 업데이트 되었는지 확인	PASS
Select mode	2-1	USER가 선택한 모드에 따라 정해진 메뉴가 실행되는지 확인	PASS
	2-2	USER가 여러 개의 모드를 선택했을때 현장 구매모드가 실행되는지 확인	FAIL ->버튼 입력을 통한 input으로 한번에 여러 개의 모드를 선택하는 경우는 불가능
Manage Stock	3-1	Manager가 갱신한 재고 정보가 적용되는지 확인	PASS
	3-2	Manager가 총 8가지 이상의 음료를 갱신할 경우 에러가 나오는지 확인	PASS
Show item	4-1	모든 음료가 나오는지 확인	PASS
	4-2	User가 선택한 음료가 선택되는지 확인	FAIL ->선구매 코드를 이용한 음료 구매시 선택한 음료와 다른 음료가 Item out되는 경우 발생
Check stock count	6-1	User가 선택한 음료에 대해 현재 자판기에서 판매 가능한지 확인	PASS
	6-2	현재 자판기에서 판매 가능하다면 Purchase를 정상적으로 실행하는지 확인	PASS
	6-3	통신을 통해 다른 자판기로부터 재고를 받아오는지 확인	PASS
	6-4	다른 자판기로부터 재고가 있다는 응답을 받으면, "Select Advance Payment"를 정상적으로 실행하는지 확인	PASS
Update stock	7-1	음료 재고정보를 주어진 값에 맞게 최신화 하는지 확인	FAIL ->선구매 코드를 이용하여 item out 실행 후 다른 자판기에서 재고가 update되지

			않는 상황 발생
Purchase	8-1	결제 성공했을 때 음료 재고를 맞게 줄이는지 확인	PASS
	8-2	결제 성공했을 때 해당 음료를 배출하는지 확인	PASS
	8-3	결제 실패했을 때 결제 실패 메시지를 보여주는지 확인	PASS
Advance purchase	9-1	선구매결제에 성공했을 때, 선구매코드를 받아오는지 확인	PASS
	9-2	선구매 결제에 성공했을 때, 받은 선구매코드를 "Message Request" 통해 "Broadcast" 하는지 확인	FAIL -> 선구매 코드는 현재 자판기에만 저장되고 다른 자판기에는 저장되지 않음
	9-3	선구매 결제에 성공했을 때, 해당 음료가 있는 자판기에 대해 "Inform Location"을 실행하는지 확인	PASS
	9-4	선구매 결제에 성공했을 때, 생성된 선구매코드를 보여주는지 확인	PASS
	9-5	선구매 결제에 실패했을 때, 결제 실패 메시지를 보여주는지 확인	PASS
	9-6	선구매 결제에 실패했을 때, 감소시켰던 재고 수량을 다시 증가시키는지 확인	FAIL -> 스펙과 다르게 코드를 구현 케이스인것 같음
	9-7	선구매 진행 후 선구매 코드를 현재 자판기에 저장하는지 확인	PASS
	9-8	선구매 코드 입력 결제 시, 해당 item의 재고가 줄어드는지 확인	FAIL -> 줄어들지 않음
Check payment	10-1	결제 결과를 받아오는지 확인	PASS
	10-2	정상적인 결제 카드가 아니라면 "잘못된 카드"라는 예외 메시지를 보여주는지 확인	PASS
Select advance payment	11-1	선구매를 선택한 음료에 대해 "Message Request"로 재고를 받아오는지 확인	PASS
	11-2	선구매결정한 음료의 재고가 남아있다면 "Message Request"로 재고를 감소시키는지 확인	FAIL -> 줄어들지 않음
	11-3	재고를 감소시킨 후 "Advance Purchase"를 실행하는지 확인	FAIL -> 재고가 감소되지

			않아 테스트 불가
	11-4	선구매 결정된 음료의 재고가 없다면 품질메시지를 보여주고, "Select Mode"로 돌아가는지 확인	PASS
	11-5	위치확인 모드를 선택한 음료에 대해 "Message Request"로 재고를 받아오는지 확인	PASS
	11-6	위치확인 모드를 선택한 음료의 재고가 남아 있다면 해당 음료에 대해 "Inform Location"을 실행하는지 확인	PASS
	11-7	위치확인 모드를 선택한 음료의 재고가 남아 있지 않다면 품질메시지를 보여주고, "Select Mode"로 돌아가는지 확인	PASS
	11-8	모드 선택과 취소를 틀다 누를 경우 사용자로부터 재입력 받는지 확인	FAIL ->버튼을 통한 Input을 받으므로 버튼을 중복하여 누를수가 없는 상황 발생
	11-9	선구매 진행을 위한 결제시 다른 자판기들의 재고가 바로 줄어드는지 확인	FAIL ->현재 자판기에서 Item out을 하기 전까지는 다른 자판기에는 재고가 줄어들지 않아 선결제가 무한정으로 가능함
	11-10	선결제 후 선구매 코드 입력시 해당 item의 재고가 줄어드는지 확인	FAIL -> 줄어들지 않음
Inform location	12-1	음료를 갖고있는 자판기에 대해 위치를 보여 주는지 확인	PASS
Create verification code	13-1	음료가 선택되었을 때, 새로운 선구매 코드를 생성하는지 확인	PASS
	13-2	"Create Verification Code"를 통해 생성된 선구매 코드를 보여주는지 확인	PASS
	13-3	선결제 코드가 여러 종류의 음료에 중복되어 생성되지 않는지 확인	FAIL ->서로 다른 종류의 음료에 같은 선구매 코드가 할당되어 선택한 음료외의 다른 음료가 item out되는 상

			황 발생
Item out	15-1	음료가 선택되어 있고, 결제가 완료되었거나 유효한 선구매 코드를 입력했을 때만 음료를 제공하는지 확인	FAIL -> 현재 자판기에 재고가 없는데도, 선구매 코드를 입력하면 item out이 되고 있음
	15-2	현재 자판기에 재고가 없을 때는, 선구매 코드를 입력해도 item out이 되지 않아야 함	FAIL -> item out 됨
Read verification code	16-1	User가 입력한 선구매 코드를 "Check Verification Code"에 전달하고 그 결과를 받는지 확인	PASS
	16-2	기록된 코드가 유효하다면 "Item Out"을 실행하는지 확인	PASS
	16-3	"Item Out"을 실행한 후에 "Reset Verification Code"를 실행하는지 확인	PASS
Check verification code	17-1	시스템에 입력된 코드가 존재하는지 확인	PASS
Reset verification code	18-1	시스템에서 입력된 코드를 정상적으로 삭제하는지 확인	PASS
Message request	19-1	보내고자하는 메시지 종류를 수신 대상에게 잘보내는지 확인	FAIL -> "Broadcast"메시지 실패로 다른 DVM에게 선구매 코드를 전송하지 못함
	19-2	수신된 메시지에 따라 명시된 응답을 전송한 DVM에 정상적으로 보내는지 확인	PASS
	19-3	알 수 없는 종류에 메시지를 수신한 경우 알 수 없음을 응답하는지 확인	PASS
전체 Test Case : 50 개			Pass : 35 개

(35 / 50) = 70% Pass

