

Specification Review & System Test

for Distributed Vending Machine(DVM)

OOAD Class B – T2

Project Team

Team 6

Date

2021-05-20

Team Information

201511273 여찬종

201513205 송승현

201612368 이지우

201811301 한지희

Contents

1. Specification Review
 - 1.1. Documents Reference
 - 1.2. Specification Review List
 - 1.3. Issue upload to CTIP
2. Structural Coverage – Jacoco
 - 2.1. Jacoco Counter
 - 2.2. Unit Test Coverage
3. Category-Partition Testing
 - 3.1. Categorize
 - 3.2. Test Result
 - 3.3. Issue upload to CTIP
4. Brute Force Testing
 - 4.1. Test Result
 - 4.2. Issue upload to CTIP

1. Specification Review

1.1. Documents Reference

- 아래의 문서를 보고 Specification Review 를 진행하였습니다.

Stage	Document Name
	DVM_PFR
1000	OOPT Stage1000 Project Planning Ver.2
SRS	SRS Ver.2
2030	OOPT Stage2030 OOA Ver.3
SDS	SDS Ver.2
2040	OOPT Stage2040 OOD Ver.2

1.2. Specification Review List

- 아래 표의 Stage 항목은 1.1. Documents Reference 표의 Stage 항목입니다. 각 Stage 의 수정할 문서는 Document Name 에 기술하였습니다.

- 내용이 완전히 일치하는 항목을 갖는 경우, 각 문서에 대한 수정을 하나의 Issue 를 통해 요구하였습니다.

- Stage 2030 과 SRS, Stage 2040 과 SDS 가 동일한 내용을 갖는 경우, 하나의 Issue 에 두 문서에 대한 수정을 요구하였습니다.

No	Stage	Page	Name	Review
1	1000	3	Functional-Requirements add 1	DVM_PFR Page 2 의 1.2 Requirements 의 1 번째 항목에서 '총 음료의 개수는 20 종류이다.'에 대한 언급이 없으므로 추가 필요

2	1000	3	Functional-Requirements add 2	DVM_PFR Page 2 의 1.2 Requirements 의 2 번째 항목에서 '한 자판기는 7 종류의 음료를 판매한다.'에 대한 언급이 없으므로 추가 필요
3	1000	3	Functional-Requirements Refine 1	5 번째 항목에서 다른 위치의 자판기 정보 제공하는 조건에 다른 자판기에는 해당 상품에 대한 재고가 있다는 조건 추가 필요
4	1000	3	Non-Functional Requirements add 1	DVM_PFR Page 3 의 1.5 Assumptions and dependencies 의 1 번째 항목에서 '자판기의 개수는 최대 10 개로 가정한다.'에 대한 언급이 없으므로 추가 필요
5	1000	3	Non-Functional-Requirements Refine 1	'자판기의 조작이 쉬워야한다', '재고의 관리가 용이해야 한다.', '자판기가 튼튼해야 한다.'와 같은 모호한 표현 수정 필요
6	1000	5	Functional Requirement - Description Refine	Pay by Cash, Pay by Card 의 '가격합'이라는 표현과 Calculate Price(Cash), Calculate Price(Card)의 '상품들의'라는 표현을 보면 다중 선택이 가능하다는 의미가 될 수 있으므로 수정 필요
7	1000	5	Functional Requirement - Login	cf. switch 가 무슨 뜻인지 이해할 수 없으므로 삭제 또는 추가 설명 필요
8	1000	6	R1.1 name	Page 5 의 Function 1 번째 항목은 Show Item 인데, Show Menu 로 나와있으므로 수정 필요
9	1000	8	R1.5 Category	Page 6 의 R1.5 Category 는 Evident 인데, Hidden 으로 나와있으므로 수정 필요
10	1000	11	Use Case 15 Description	Page 14 의 Test Case 15 에는 '거리순서대로'라는 조건에 맞는 경우 Test Pass 라고 나와있는데, 해당 조건에 대한 설명이 없으므로 추가 필요
11	SRS		Page Number	Page Number 추가 필요

12	2030	13	Term Name - card	Stage1000 의 Page 7 의 Credit Description 과 동일하지만, Card 라고 나와있으므로 수정 필요
13	2030 SRS	3 5	Use Case 1 - Overview 1	세 가지 주요 기능이 무엇인지 정확히 명시하도록 수정 필요
14	2030 SRS	3 5	Use Case 1 - Overview 2	Stage1000 의 Page 9dml Description 에는 '시스템이 상시 DVM 에 있는 모든 제품의 이름과 재고 현황을 알려줍니다.'라고 나와있지만, '~ 초기 선택 메뉴를 보여줍니다.'로 불일치하는 설명 수정 필요
15	2030 SRS	3 5	Use Case 2	같은 의미의 용어로 보여지는 '제품', '상품'이 혼용되어 사용되고 있으므로 통일하는 방향으로 수정 필요
16	2030 SRS	9 11	Use Case 13 Type	Typical Courses of Events 에 System 의 Event 만 존재하므로 Hidden 으로 수정 필요
17	2030 SRS	9 11	Use Case 14 Type	Typical Courses of Events 에 System 의 Event(Another DVM)만 존재하므로 Hidden 으로 수정 필요
18	2030 SRS	11 12	Use Case Number	Stage1000 의 Use Case 는 19 개로 17. Manage Kind of Product 가 누락된 것이라면 수정 필요 (누락이 아닌 삭제의 경우, Stage1000 수정 필요)
19	2030	13	Refine Glossary	인증 코드라는 말을 주로 사용하지만 Glossary 에는 코드에 대해서만 정의되어 있으므로 수정 또는 통일 필요
20	2030	16	System Sequence Diagram	Use Case [12-14]로 하나의 Sequence Diagram 만 표현되어있는데, Use Case 별로 분리하여 작성 필요
21	2030	21	Traceability System Operation	9. GetNonItemLocation(), 10. GiveDVMInfo()에 대해서만 뒤에 '()'표시를 붙여 함수형로 표현하였는데, 다른 항목에 대해서도 동일하게 함수형으로 처리하거나 '()'표시 삭제 필요

22	SRS	14	Other Requirement	1 번째 항목의 '자판기 내부의' 다음에 이어지는 문장이 없으므로 삭제 또는 수정 필요
23	2030	18-20	System Test Case	거의 모든 항목에서 '정상적으로', '제대로'라는 표현을 사용하는데 어떤 상황이 정상적인지 파악 불가하므로 명확한 용어 사용 필요
24	2040	8	Use Case 10 - Cross Reference	Stage2030 Use Case 10 의 내용과 불일치하므로 삭제 또는 수정 필요
25	2040	9	Use Case 11 - Cross Reference	Stage2030 Use Case 11 의 내용과 불일치하므로 삭제 또는 수정 필요
26	2040 SDS	15-17 6-8	Interaction Diagram	Use Case 별로 Interaction Diagram 분리 작성 필요
27	2040	16	Use Case 16 - Event	Typical Courses of Events 에 1 번 Event 뒤에 아이디와 비밀번호 출력에 대한 System 의 Event 가 누락되어 있으므로 추가 필요
28	2040 SDS	16 7	Interaction Diagram - Use Code & Code Validation	CodeTable 의 경우 Class Diagram 에서는 객체가 아닌데, Interaction Diagram 에서는 객체인 것처럼 작성되어 있음
29	2040	21	Traceability line	연결하는 화살표가 많이 겹쳐서 내용 파악이 불가하므로 넘버링을 통해 표로 작성 요청
30	SDS		Page Number	Page Number 추가 필요
31	SDS	6	Interaction Diagram - 3.3.3	Calculate Price 에 '(Cash)'가 누락되어 추가 필요

1.3. Issue upload to CTIP

- Jira

프로젝트 / konkuk_VV_t_6_2021 / KVT62-29 / KVT62-30

Spec Review

첨부 하위 작업 생성 이슈 연결 ...

설명
 설명 추가하기...

링크된 이슈 +

blocks

<input checked="" type="checkbox"/>	KVT62-40 Functional-Requirements add 1	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-41 Functional-Requirements add 2	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-43 Non-Functional Requirements add 1	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-44 Non-Functional-Requirements Refine 1	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-45 Functional Requirement – Description Refine	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-46 Functional Requirement - Login	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-48 R1.5 Category	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-47 R1.1 name	↑	SPECREVIEW ▾

<input checked="" type="checkbox"/>	KVT62-49 Use Case 15 Description	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-50 Page Number	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-51 Term Name - card	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-53 Use Case 1 – Overview 2	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-52 Use Case 1 – Overview 1	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-42 Functional-Requirements Refine 1	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-54 Use Case 2	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-55 Use Case 13 Type	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-58 Refine Glossary	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-57 Use Case Number	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-59 System Sequence Diagram	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-56 Use Case 14 Type	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-61 Other Requirement	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-63 Use Case 10 – Cross Reference	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-62 System Test Case	↑	SPECREVIEW ▾

<input checked="" type="checkbox"/>	KVT62-60 Traceability System Operation	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-67 Interaction Diagram – Use Code & Code Validation	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-66 Use Case 16 - Event	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-65 Interaction Diagram	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-70 Interaction Diagram – 3.3.3	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-69 Page Number	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-64 Use Case 11 – Cross Reference	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/>	KVT62-68 Traceability line	↑	SPECREVIEW ▾

2. Structural Coverage – Jacoco

2.1. Jacoco Counter

- Jacoco 는 Instructions Counter 와 Branchs Counter 를 통해 Coverage 를 계산합니다.

- Instructions Counter : Jacoco 의 최소단위로 나누어진 코드의 실행여부로 계산
- Branchs Counter : 조건문으로 나누어지는 분기의 실행여부로 계산

2.2. Unit Test Coverage

- 전체 소스코드에 대한 Coverage 는 아래와 같습니다. (UI 를 구현 소스코드 포함)

- Coverage 수식은 (실행된 각 코드단위) / (전체 코드단위) X 100 입니다.

- Instructions Coverage : $2,902/3,586 \times 100 = 19\%$
- Branches Coverage : $48/124 \times 100 = 61\%$

OOADProject				
Element	Missed Instructions	Cov.	Missed Branches	Cov.
default		19%		61%
Total	2,902 of 3,586	19%	48 of 124	61%

- 개발팀에서 작성한 Class Diagram 을 통해 확인한 객체는 DVM, Payment, Item 입니다. 따라서 위 객체들에 대한 소스코드의 Coverage 는 각각 아래와 같습니다.

Class	Coverage	
DVM	Instructions	$473/525 \times 100 = 90\%$
	Branches	$50/56 \times 100 = 90\%$
Payment	Instructions	$90/90 \times 100 = 100\%$

	Branches	12/12 X 100 = 100%
Item	Instructions	58/58 X 100 = 100%
	Branches	4/4 X 100 = 100%










DVM

Element	Missed Instructions	Cov.	Missed Branches	Cov.
giveCode(Item)		55%		50%
getRegion()		0%		n/a
getLocation()		0%		n/a
DVM(String, int, int, int, String, String, ArrayList, double, Payment)		100%		87%
getNoneItemLocation(String)		100%		100%
getAnotherDVMInfo(String)		100%		100%
manageAmount(String, int)		100%		100%
getItemFromCode(String)		100%		100%
getItemFromName(String)		100%		100%
giveItem(Item)		100%		100%
inputCode(String)		100%		100%
login(String, String)		100%		100%
inputCash(int, Item)		100%		100%
removeCode(String)		100%		100%
manageAmountCash(int)		100%		100%
static {...}		100%		n/a
checkTotalCash(int)		100%		100%
insertCard(String, Item)		100%		n/a
sortDVM(ArrayList)		100%		n/a
codeValidation(String)		100%		n/a
setDist(double)		100%		n/a
getDist()		100%		n/a
getItemList()		100%		n/a
getCodeTable()		100%		n/a
Total	52 of 525	90%	6 of 66	90%

Payment

Element	Missed Instructions	Cov.	Missed Branches	Cov.
calculatePriceCard(String, Item)		100%		100%
calculatePriceCash(int, Item)		100%		100%
consumeCard(String, int)		100%		n/a
isSufficient(int, int)		100%		100%
isValidCard(String)		100%		n/a
static {...}		100%		n/a
calculateChange(int, int)		100%		n/a
Payment()		100%		n/a
Total	0 of 90	100%	0 of 12	100%

Item

Element	Missed Instructions	Cov.	Missed Branches	Cov.
● Item()		100%		n/a
● Item(String, int, int, int)		100%		n/a
● reduceAmount()		100%		100%
● changeAmount(int)		100%		100%
● getItemName()		100%		n/a
● getItemPrice()		100%		n/a
● getItemAmount()		100%		n/a
Total	0 of 58	100%	0 of 4	100%

3. Category-Partition Testing

3.1. Categorize

구현된 DVM 은 각 단계별로 사용자가 사용가능한 버튼이 다르기 때문에, 화면구성이 바뀌는 것을 기준으로 나누었습니다. 단, 결제 화면은 다른 단계(쿠폰결제, 일반결제) 로 접근하더라도 같은 UI 구성을 가지기 때문에 같은 Phase 로 취급하였습니다. 다른 화면으로 이동하는 버튼은 해당 화면이 오류 없이 출력되는 것을 Value-True 로 취급하였으며, 모든 입력 값은 해당 품에 적합한 입력 값 혹은 적합하지 않은 입력 값으로만 구분하였습니다. (동전 혹은 재고에 한계가 지정되지 않았기 때문. Ex) 수량은 숫자이며 정수이고 음수만 아니면 됨.)

No	Phase	Group	Category	Value	Constraint
1	Active Choices	Environment	Program Start	Success	[property SC]
		Input	Button	Buy Product	[if SC]
				Use Code	[if SC]
				Admin Login	[if SC]
		Value	Output	True	
				False	
2	Admin Choices	Environment	Program Start	Success	[property SC]
		Input	Button	Manage Stock	[if SC] [property MS]
				Manage Cash	[if SC] [property MC]
				Cancel	[if SC] [property C]
		Value	Output	Move to Manage Stock	[if MS]
				Move to Manage Cash	[if MC]
Move to Back	[if C]				
3	Admin Login	Environment	Program Start	Success	[property SC]
		Input	Button1	ID,PW Input	[if SC] [property IDPW]

				Cancel	[if SC] [property C]	
			Button2	Enter	[if SC] [if !C] [property Enter]	
			Admin Info	valid	[if IDPW] [property valid]	
				invalid	[if IDPW] [property invalid]	
			Value	Output	Move to admin Selection	[if valid]
					ID or Password incorrect	[if invalid]
Move to Back	[if C]					
4	Change_Cash	Environment	Program Start	Success	[property SC]	
		Input	Button1	Cash Input	[if SC] [property IDPW]	
				Cancel	[if SC] [property C]	
			Button2	Enter	[if SC] [if !C] [property Enter]	
			Number	Input is Integer	[if SNI] [property valid]	
		Input is not Integer		[if SNI] [property invalid]		
		Value	Output	Successful Change	[if SNI] [if valid]	
				input is not Integer	[if SNI && invalid]	
				Move to Back	[if C]	
		Cash	Same as Input	[if SC] [if SNI && valid] [if !C && !invalid]		
5	Choose_Item	Environment	Program Start	Success	[property SC]	
		Input	Item	In Stock Item	[if SC] [property ISI]	

	m			Not In Stock Item	[if SC] [property NISI]
				Cancel	[if SC] [property C]
		Value	Output	Move to Payment	[if ISI]
				Move to NISI select	[if NISI]
				Move to Back	[if C]
		6	Manage Amount	Environment	Program Start
Input	Button1			Stock Name Input	[if SC] [property SNI]
				Cancel	[if SC] [property C]
	Button2			Stock Amount Input	[if SC] [property SAI]
	Button3			Enter	[if SC] [if !C] [property Enter]
	Name			Correct Name	[if SC] [if SNI] [if !C] [property validName]
				Incorrect Name	[if SC] [if SNI] [if !C] [property invalidName]
	Amount			Same as Input	[if SC] [if SNI] [if !C] [if validName]
Value	Output			Successful Change	[if validName]
				Incorrect Name Input	[if invalidName]
				Move to Back	[if C]
7	Not_In_Stock_select			Environment	Program Start
		Input	Button	Search Another DVM	[if SC] [property SADVM]

	ct			Buy Code	[if SC] [property BC]		
				Admin Login	[if SC] [property C]		
				Pop up other DVM info	[if SADVM]		
		Value	Output	Move to Payment	[if BC]		
				Move to Back	[if C]		
8	Pay By Card	Environment	Program Start	Success	[property SC]		
		Input	Button1	Card Number Input	[if SC] [property CNI]		
				Cancel	[if SC] [property C]		
			Button2	Stock Amount Input	[if SC] [property SAI]		
			Button3	Enter	[if SC] [if !C] [property Enter]		
			Card Number Format	True	[if SC] [if CNI] [property ValidCard]		
				False	[if SC] [if CNI] [property InValidCard]		
		Value	Output	Successful Buy	[if ValidCard]		
				Fail to Buy	[if InValidCard]		
				Move to Back	[if C]		
		9	Pay By Cash	Environment	Program Start	Success	[property SC]
				Input	Button1	Cash Input	[if SC] [property CI]
Cancel	[if SC] [property C]						
Button2	Enter				[if SC] [if !C] [property Enter]		
Amount	Less Input				[if CI][property LeI]		
	Same Input				[if CI][property SI]		

				Larger Input	[if CI][property LI]
		Value	Output	Successful Buy	[if SI LI]
				Fail to Buy	[if LeI]
				Move to Back	[if C]
			Change	True	[if LI]
				False	[if !LI && !C]
10	Pay By Code	Environment	Program Start	Success	[property SC]
		Input	Button1	Code Number Input	[if SC] [property CNI]
				Cancel	[if SC] [property C]
			Button2	Enter	[if SC] [if !C] [property Enter]
			Code Number Format	True	[if SC] [if CNI] [property ValidCode]
				False	[if SC] [if CNI] [property InValidCode]
		Value	Output	Successful Buy	[if ValidCode]
				Fail to Buy	[if InValidCode]
				Move to Back	[if C]
		11	Payment Choices	Environment	Program Start
Input	Button			Pay By Cash	[if SC] [property PBCash]
				Pay By Card	[if SC] [property PBCard]
				Cancel	[if SC] [property C]
Value	Output			Move to Pay by Cash	[if PBCash]
				Move to Pay by Card	[if PBCard]

				Move to Back	[if C]
--	--	--	--	--------------	--------

3.2. Test Result

Phase1. Activate_Choices

No	Category1 (Program Start)	Category2 (Button)	Category3 (Output Success)	Test Result
1	Error			pass
2	Success	Buy Product	True	pass
3	Success	Buy Product	False	pass
4	Success	Use Code	True	pass
5	Success	Use Code	False	pass
6	Success	Admin Login	True	pass
7	Success	Admin Login	False	pass

Phase2. Admin_Choices

No	Category1 (Mode On)	Category2 (Button)	Category3 (Output Success)	Test Result
8	Success	Manage Stock	Move to Manage Stock	pass
9	Success	Manage Cash	Move to Manage Cash	pass
10	Success	Cancel	Move to Back	pass

Phase3. Admin_Login

No	Category1 (Mode On)	Category2 (Button)	Category3 (Button2)	Category4 (adminInfo)	Category5 (Output)	Test Result
----	------------------------	-----------------------	------------------------	--------------------------	-----------------------	-------------

11	Success	ID,PW Input	Enter	valid	Move to adminSelection	pass
12	Success	Cancel	<n/a>	<n/a>	Move to Back	pass

Phase4. Change_Cash

No	Category1 (Mode On)	Category2 (Button)	Category3 (Button2)	Category4 (Number)	Category5 (Output)	Category6 (Cash)	Test Result
13	Success	Cash Input	Enter	Input is Integer	Successful Change	Same as Input	pass
14	Success	Cash Input	Enter	Input is not Integer	Input is not Integer	<n/a>	pass
15	Success	Cancel	<n/a>	<n/a>	Move to Back	<n/a>	pass

Phase5. Choose_Item

No	Category1 (Mode On)	Category2 (Item)	Category3 (Value)	Test Result
16	Success	In Stock Item	Move to Payment	pass
17	Success	Not In Stock Item	Move to NISI select	pass
18	Success	Cancel	Move to Back	pass

Phase6. Manage_Amount

No	Category1 (Mode On)	Category2 (Button)	Category3 (Button2)	Category4 (Button3)	Category5 (Name)	Category6 (Amount)	Category7 (Output)	Test Result
19	Success	Stock Name Input	Stock Amount Input	Enter	Correct Name	Same as Input	Successful Change	pass

20	Success	Stock Name Input	Stock Amount Input	Enter	Incorrect Name	<n/a>	Incorrect Name Input	pass
21	Success	Cancel	Stock Amount Input	<n/a>	<n/a>	<n/a>	Move to Back	pass

Phase7. Not_In_Stock_select

No	Category1 (Mode On)	Category2 (Button)	Category3 (Output Success)	Test Result
22	Success	Search Another DVM	Pop up other DVM info	pass
23	Success	Buy Code	Move to Payment	pass
24	Success	Cancel	Move to Back	pass

Phase8. Pay By Card

No	Category1 (Mode On)	Category2 (Button)	Category3 (Button2)	Category4 (Card Number Format)	Category5 (Output)	Test Result
25	Success	Card Number Input	Enter	True	Successful Buy	pass
26	Success	Card Number Input	Enter	False	Fail to Buy	Pass
27	Success	Cancel	<n/a>	<n/a>	Move to Back	Pass

Phase9. Pay by Cash

No	Category1 (Mode On)	Category2 (Button)	Category3 (Button2)	Category4 (Amount)	Category5 (Output)	Category6 (Change)	Test Result
28	Success	Cash Input	Enter	Less Input	Fail to Buy	False	pass
29	Success	Cash Input	Enter	Same Input	Successful Buy	False	pass
30	Success	Cash Input	Enter	Larger Input	Successful Buy	True	pass
31	Success	Cancel	<n/a>	<n/a>	Move to Back	<n/a>	

Phase10. Pay By Code

No	Category1 (Mode On)	Category2 (Button)	Category3 (Button2)	Category4 (Cord Number Format)	Category5 (Output)	Test Result
32	Success	Code Number Input	Enter	True	Successful Buy	pass
33	Success	Code Number Input	Enter	False	Fail to Buy	Pass
34	Success	Cancel	<n/a>	<n/a>	Move to Back	Pass

Phase11. Payment Choices

No	Category1 (Mode On)	Category2 (Button)	Category3 (Output Success)	Test Result
35	Success	Pay By Cash	Move to Pay by Cash	pass
36	Success	Pay By Card	Move to Pay by Card	pass
37	Success	Cancel	Move to Back	pass

3.3. Issue upload to CTIP

- Jira

모든 Test 에 대해 Pass 의 결과를 보여 Jira 에 Issue 등록하지 않았습니다.

- Zephyr

KVT62-87	PASS	DVM-1.1	Category_Partition_Testing
KVT62-88	PASS	DVM-1.2	Category_Partition_Testing
KVT62-89	PASS	DVM-1.3	Category_Partition_Testing
KVT62-90	PASS	DVM-1.4	Category_Partition_Testing
KVT62-91	PASS	DVM-1.5	Category_Partition_Testing
KVT62-92	PASS	DVM-1.6	Category_Partition_Testing
KVT62-93	PASS	DVM-1.7	Category_Partition_Testing
KVT62-94	PASS	DVM-2.1	Category_Partition_Testing
KVT62-95	PASS	DVM-2.2	Category_Partition_Testing
KVT62-96	PASS	DVM-2.3	Category_Partition_Testing
KVT62-97	PASS	DVM-3.1	Category_Partition_Testing
KVT62-98	PASS	DVM-3.2	Category_Partition_Testing
KVT62-99	PASS	DVM-4.1	Category_Partition_Testing
KVT62-100	PASS	DVM-4.2	Category_Partition_Testing
KVT62-101	PASS	DVM-4.3	Category_Partition_Testing

KVT62-102	PASS	DVM-5.1	Category_Partition_Testing
KVT62-103	PASS	DVM-5.2	Category_Partition_Testing
KVT62-104	PASS	DVM-5.3	Category_Partition_Testing
KVT62-105	PASS	DVM-6.1	Category_Partition_Testing
KVT62-106	PASS	DVM-6.2	Category_Partition_Testing
KVT62-107	PASS	DVM-6.3	Category_Partition_Testing
KVT62-108	PASS	DVM-7.1	Category_Partition_Testing
KVT62-109	PASS	DVM-7.2	Category_Partition_Testing
KVT62-110	PASS	DVM-7.3	Category_Partition_Testing
KVT62-111	PASS	DVM-8.1	Category_Partition_Testing
KVT62-112	PASS	DVM-8.2	Category_Partition_Testing
KVT62-113	PASS	DVM-8.3	Category_Partition_Testing
KVT62-114	PASS	DVM-9.1	Category_Partition_Testing
KVT62-115	PASS	DVM-9.2	Category_Partition_Testing
KVT62-116	PASS	DVM-9.3	Category_Partition_Testing
KVT62-117	PASS	DVM-9.4	Category_Partition_Testing
KVT62-118	PASS	DVM-10.1	Category_Partition_Testing
KVT62-119	PASS	DVM-10.2	Category_Partition_Testing
KVT62-120	PASS	DVM-10.3	Category_Partition_Testing
KVT62-121	PASS	DVM-11.1	Category_Partition_Testing
KVT62-122	PASS	DVM-11.2	Category_Partition_Testing
KVT62-123	PASS	DVM-11.3	Category_Partition_Testing

4. Brute Force Testing

- 3. Category-Partition Testing 에서 DVM 자체의 기능에 대한 대부분의 Test 를 진행하였기 때문에 Brute Force Testing 에서는 여러 개의 DVM 이 존재하는 경우 발생할 수 있는 Case 와 기능 이외의 환경 조건에 대한 Test 를 중심으로 합니다.

4.1. Test Result

No	Test Case	Description	Test Result
1	11 DVMs	자판기를 11 개 만드는 게 불가능한지 확인한다.	Fail
2	Show 20 and Sell 7 item	자판기가 20 개의 음료를 가지며 재고가 있는 음료가 7 개 존재함을 확인한다.	Fail
3	Only One DVM Check non-item	자판기가 1 개일 때, 해당 자판기의 음료 A 에 대한 재고가 없는 경우를 설정한다. 이 후 다른 자판기로의 재고 요청 시 주변에 구입가능한 자판기가 없다는 결과를 확인한다.	Pass
4	Two DVM check non-item	자판기가 2 개일 때, 각 자판기의 음료 A 에 대한 재고가 없는 경우를 설정한다. 이 후 자판기 중 한 자판기에서 '다른 DVM 확인'을 입력하는 경우 주위에 가능한 자판기가 없다는 결과를 확인한다.	Pass
5	Left one item access - Cash	음료 A 에 대한 재고가 없는 자판기 1 과 음료 A 에 대한 재고가 1 개 남은 자판기 2 를 설정한다. 자판기 1 에서 음료 A 에 대한 구매를 누른 뒤, 자판기 2 에서 음료 A 를 현금으로 구매한다. 이 후 자판기 1 에서 '다른 DVM 확인'을 입력하는 경우 주위에 가능한 자판기가 없다는 결과를 확인한다.	Fail
6	Left one item access - Card	음료 A 에 대한 재고가 없는 자판기 1 과 음료 A 에 대한 재고가 1 개 남은 자판기 2 를 설정한다. 자판기 1 에서 음료 A 에 대한 구매를 누른 뒤, 자판기 2 에서 음료 A 를 카드로 구매한다. 이 후 자판기 1 에서 '다른 DVM 확인'을 입력하는 경우 주위에 가능한 자판기가 없다는 결과를 확인한다.	Fail

7	Left one item access - Code	음료 A 에 대한 재고가 없는 자판기 1 과 음료 A 에 대한 재고가 1 개 남은 자판기 2 를 설정한다. 자판기 1 에서 음료 A 에 대한 구매를 누른 뒤, 자판기 2 에서 음료 A 를 인증코드로 구매한다. 이후 자판기 1 에서 '다른 DVM 확인'을 입력하는 경우 주위에 가능한 자판기가 없다는 결과를 확인한다.	Fail
8	Card limit - one DVM	한도가 4000 인 카드로 자판기 1 에서 4000 을 결제하고 1000 인 음료 A 를 구매했을 때, 한도초과임을 확인한다.	Pass
9	Card limit - Two DVM	한도가 4000 인 카드로 자판기 1 에서 2000, 자판기 2 에서 2000 을 결제하고, 자판기 3 에서 결제 시 한도초과임을 확인한다.	Pass
10	Cash increase	관리자 메뉴에서 자판기의 현금보유량을 500 으로 설정한 뒤 가격이 1000 인 음료 A 를 구매했을 때, 현금보유량의 증가를 확인한다.	Fail
11	Code - One DVM	자판기 1 에서 재고가 없는 음료 A 에 대한 인증 코드를 구입한 뒤, 관리자 메뉴에서 자판기 1 의 음료 A 에 대한 재고를 추가한 뒤 인증 코드로 구매가 가능함을 확인한다.	Pass
12	Code - Two DVM	자판기 1 에서 재고가 없는 음료 A 에 대한 인증 코드를 구입한 뒤, 자판기 2 에서 음료 A 에 대해 인증 코드로 구매가 가능함을 확인한다.	Pass
13	One code two access	음료 A 에 대해 구매한 인증 코드를 자판기 1 과 자판기 2 에 동일하게 입력한 뒤 순차적으로 쿠폰번호 입력을 누르는 경우 자판기 1 에서만 구매가 가능함을 확인한다.	Pass
14	Item inventory check - Non-item	음료 A 에 대한 재고가 없는 자판기 1 과 음료 A 에 대한 재고가 1 개 남은 자판기 2 를 설정한다. 자판기 1 에서 음료 A 에 대한 구매를 누른 뒤, 자판기 2 의 관리자 메뉴에서 음료 A 에 대한 재고를 0 으로 수정한다. 이후 자판기 1 에서	Fail

		'다른 DVM 확인'을 입력하는 경우 주위에 가능한 자판기가 없다는 결과를 확인한다.	
15	Item inventory check – Item	음료 A 에 대한 재고가 없는 자판기 1, 자판기 2 와 음료 A 에 대한 재고가 2 개 남은 자판기 3 을 설정한다. 자판기 1 에서 음료 A 에 대한 구매를 누른 뒤, 자판기 2 의 관리자 메뉴에서 음료 A 에 대한 재고를 1 으로 수정한다. 이 후 자판기 1 에서 '다른 DVM 확인'을 입력하는 경우 자판기 2 에 대한 거리 결과를 확인한다.	Fail

4.2. Issue upload to CTIP

- Zephyr

*ID ^	*현황 v	*요약	*라벨(Label) v
KVT62-72	FAIL	11 DVMs	Brute_Force_Testing
KVT62-73	FAIL	Show 20 and Sell 7 item	Brute_Force_Testing
KVT62-74	PASS	Only One DVM Check non-item	Brute_Force_Testing
KVT62-75	PASS	Two DVM check non-item	Brute_Force_Testing
KVT62-76	FAIL	Left one item access - Cash	Brute_Force_Testing
KVT62-77	FAIL	Left one item access - Card	Brute_Force_Testing
KVT62-78	FAIL	Left one item access - Code	Brute_Force_Testing
KVT62-79	PASS	Card limit – one DVM	Brute_Force_Testing
KVT62-80	PASS	Card limit – Two DVM	Brute_Force_Testing
KVT62-81	FAIL	Cash increase	Brute_Force_Testing

KVT62-82	PASS	Code – One DVM	Brute_Force_Testing
KVT62-83	PASS	Code – Two DVM	Brute_Force_Testing
KVT62-84	PASS	One code two access	Brute_Force_Testing
KVT62-85	FAIL	Item inventory check – Non-item	Brute_Force_Testing
KVT62-86	FAIL	Item inventory check – Item	Brute_Force_Testing

- Jira

프로젝트 / konkuk_VV_t_6_2021 / KVT62-29 / KVT62-33

Brute Force

첨부 하위 작업 생성 이슈 연결

설명 • 저장되지 않은 변경 사항
System Testing 중 하나로 전체적인 System을 Test합니다.

링크된 이슈 +

is blocked by

<input checked="" type="checkbox"/> KVT62-34 BF1. 11 DVMs	↑	SYSTEMTESTING
<input checked="" type="checkbox"/> KVT62-35 BF2. Show 20 and Sell 7 item	↑	SYSTEMTESTING
<input checked="" type="checkbox"/> KVT62-36 BF5. Left one item access - Cash	↑	SYSTEMTESTING
<input checked="" type="checkbox"/> KVT62-37 BF6. Left one item access - Card	↑	SYSTEMTESTING
<input checked="" type="checkbox"/> KVT62-38 BF7. Left one item access - Code	↑	SYSTEMTESTING
<input checked="" type="checkbox"/> KVT62-39 BF10. Cash increase	↑	SYSTEMTESTING