

Static Analysis & 2nd System Test

for Distributed Vending Machine(DVM)

OOAD Class B – T2

Project Team

Team 6

Date

2021-06-03

Team Information

201511273 여찬종

201513205 송승현

201612368 이지우

201811301 한지희

Contents

1. Specification Review
 - 1.1. 1st Cycle Specification Review List & Revision Result
 - 1.2. 2nd Cycle Specification Review
 - 1.3. Issue Upload to Jira
2. Structural Coverage – Jacoco
 - 2.1. Unit Test Coverage
3. Category-Partition Testing
 - 3.1. Categorize
 - 3.2. Test Result
4. Brute Force Testing
 - 4.1. Test Result
5. Static Analysis
 - 5.1. Maintainability
 - 5.2. Complexity
 - 5.3. Issue List
 - 5.4. Issue Upload to Jira
6. Overall - 2nd Cycle
 - 6.1. Test Quality
 - 6.2. Feedback

1. Specification Review

1.1. Specification Review List & Revision Result

- Jira 에 등록했던 31 개의 Specification Review 에 대한 Issue 가 No.17 을 제외하고 모두 Revision 완료되어 있음을 확인하였습니다.

No	Stage	Page	Name	Review	Result
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	Pass
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	Pass
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	Pass
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	Pass
5	1000	3	Non-Functional-Requirements Refine 1	'자판기의 조작이 쉬워야한다', '재고의 관리가 용이해야 한다.', '자판기가 튼튼해야 한다.'와 같은 모호한 표현 수정 필요	Pass
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	Pass

7	1000	5	Functional Requirement - Login	cf. switch 가 무슨 뜻인지 이해할 수 없으므로 삭제 또는 추가 설명 필요	Pass
8	1000	6	R1.1 name	Page 5 의 Function 1 번째 항목은 Show Item 인데, Show Menu 로 나와있으므로 수정 필요	Pass
9	1000	8	R1.5 Category	Page 6 의 R1.5 Category 는 Evident 인데, Hidden 으로 나와있으므로 수정 필요	Pass
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	Pass
11	SRS		Page Number	Page Number 추가 필요	Pass
12	2030	13	Term Name - card	Stage1000 의 Page 7 의 Credit Description 과 동일하지만, Card 라고 나와있으므로 수정 필요	Pass
13	2030 SRS	3 5	Use Case 1 - Overview 1	세 가지 주요 기능이 무엇인지 정확히 명시하도록 수정 필요	Pass
14	2030 SRS	3 5	Use Case 1 - Overview 2	Stage1000 의 Page 9 dml Description 에는 '시스템이 상시 DVM 에 있는 모든 제품의 이름과 재고 현황을 알려줍니다.'라고 나와있지만, '~ 초기 선택 메뉴를 보여줍니다.'로 불일치하는 설명 수정 필요	Pass
15	2030 SRS	3 5	Use Case 2	같은 의미의 용어로 보여지는 '제품', '상품'이 혼용되어 사용되고 있으므로 통일하는 방향으로 수정 필요	Pass
16	2030 SRS	9 11	Use Case 13 Type	Typical Courses of Events 에 System 의 Event 만 존재하므로 Hidden 으로 수정 필요	Pass
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	Fail
18	2030 SRS	11 12	Use Case Number	Stage1000 의 Use Case 는 19 개로 17. Manage Kind of Product 가 누락된	Pass

				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	Pass
20	2030	16	System Sequence Diagram	Use Case [12-14]로 하나의 Sequence Diagram 만 표현되어있는데, Use Case 별로 분리하여 작성 필요	Pass
21	2030	21	Traceability System Operation	9. GetNoneltemLocation(), 10. GiveDVMInfo()에 대해서만 뒤에 '()'표시를 붙여 함수형로 표현하였는데, 다른 항목에 대해서도 동일하게 함수형으로 처리하거나 '()'표시 삭제 필요	Pass
22	SRS	14	Other Requirement	1 번째 항목의 '자판기 내부의' 다음에 이어지는 문장이 없으므로 삭제 또는 수정 필요	Pass
23	2030	18-20	System Test Case	거의 모든 항목에서 '정상적으로', '제대로'라는 표현을 사용하는데 어떤 상황이 정상적인지 파악 불가하므로 명확한 용어 사용 필요	Pass
24	2040	8	Use Case 10 - Cross Reference	Stage2030 Use Case 10 의 내용과 불일치하므로 삭제 또는 수정 필요	Pass
25	2040	9	Use Case 11 - Cross Reference	Stage2030 Use Case 11 의 내용과 불일치하므로 삭제 또는 수정 필요	Pass
26	2040 SDS	15-17 6-8	Interaction Diagram	Use Case 별로 Interaction Diagram 분리 작성 필요	Pass
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	Pass
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	Pass

29	2040	21	Traceability line	연결하는 화살표가 많이 겹쳐서 내용 파악이 불가하므로 넘버링을 통해 표로 작성 요청	Pass
30	SDS		Page Number	Page Number 추가 필요	Pass
31	SDS	6	Interaction Diagram – 3.3.3	Calculate Price 에 '(Cash)'가 누락되어 추가 필요	Pass

1.2. 2nd Cycle Specification Review

- 수정되지 않았던 No.17(Use Case14 의 Type 을 Evident->Hidden 으로 수정 요구)에 대해 개발팀과 논의한 결과, Use Case 14 에 대한 Type 은 Hidden 이라고 정의되었습니다. 따라서 Review 는 Use Case 14 의 Type 이 Hidden 이라는 전제에서 작성되었습니다.

No	Stage	Page	Name	Review
1	1000	7	Requirement – R2.3 Give DVM Info	Type 을 Hidden 으로 수정 필요
2	1000	10	Use Case – R2.3 Give DVM Info	Type 을 Hidden 으로 수정 필요
3	1000	10	Use Case Diagram	Type 이 Hidden 이므로 System 외부의 Another DVM 으로의 연결 불필요
4	2030	9	Use Case 14 Type	Type 을 Hidden 으로 수정 필요
5	2030	18	System Sequence Diagram – Use Case 14	System Sequence Diagram 불필요
6	2040	10	Use Case 14 Type	SRS 의 Use Case 14 의 Type 은 Hidden 임으로 통일 필요

1.3. Issue Upload to Jira

프로젝트 /  konkuk_VV_t_6_2021 / KVT62-126 /  KVT62-145

Specification Review

 첨부  하위 작업 생성  이슈 연결  

설명
Review는 Use Case 14의 Type이 Hidden이라는 전제에서 작성되었습니다

링크된 이슈 +

is blocked by

<input checked="" type="checkbox"/> KVT62-146	1. Requirement – R2.3 Give DVM Info	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/> KVT62-147	2. Use Case – R2.3 Give DVM Info	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/> KVT62-148	3. Use Case Diagram	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/> KVT62-149	4. Use Case 14 Type	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/> KVT62-150	5. System Sequence Diagram – Use Case 14	↑	SPECREVIEW ▾
<input checked="" type="checkbox"/> KVT62-151	6. Use Case 14 Type	↑	SPECREVIEW ▾

2. Structural Coverage – Jacoco

2.1. Unit Test Coverage

- 개발팀이 작성한 Unit Test 는 총 56 개이고, 100%의 성공률을 보였습니다.
- 전체 소스코드에 대한 Coverage 는 아래와 같습니다. (UI 를 구현 소스코드 포함)
- 1st Cycle 과 비교하여 표를 작성하였습니다.
- Coverage 수식은 (실행된 각 코드단위) / (전체 코드단위) X 100 입니다.

	1 st Cycle	2 nd Cycle
Instructions Coverage	$2,902/3,586 \times 100 = 19\%$	$740/3,947 \times 100 = 18\%$
Branches Coverage	$48/124 \times 100 = 61\%$	$52/138 \times 100 = 62\%$

- 개발팀에서 작성한 Class Diagram 을 통해 확인한 객체는 DVM, Payment, Item 입니다. 따라서 위 객체들에 대한 소스코드의 Coverage 는 각각 아래와 같습니다.
- Payment 와 Item 은 변화없이 동일하지만, DVM 의 소스코드 수정을 통해 새롭게 Unit Test 를 생성하였고, 출력되는 메시지(System.out.println())가 Coverage 에 해당되지 않음에 따라 Coverage 가 변경되었음을 확인하였습니다. 하지만 확인을 위해 출력하는 부분으로 따로 테스트의 필요성이 없다고 생각하여 Issue 등록은 하지 않았습니다.

Class	Coverage		
		1 st Cycle	2 nd Cycle
DVM	Instructions	$473/525 \times 100 = 90\%$	$529/560 \times 100 = 94\%$
	Branches	$50/56 \times 100 = 90\%$	$70/78 \times 100 = 89\%$
Payment	Instructions	$90/90 \times 100 = 100\%$	$90/90 \times 100 = 100\%$
	Branches	$12/12 \times 100 = 100\%$	$12/12 \times 100 = 100\%$
Item	Instructions	$58/58 \times 100 = 100\%$	$58/58 \times 100 = 100\%$
	Branches	$4/4 \times 100 = 100\%$	$4/4 \times 100 = 100\%$

3. Category-Partition Testing

3.1. Categorize

- 1st Cycle 과 비교하여 Phase12. Multiple_action 이 추가되었습니다.
- Phase 12. 의 경우 여러대의 DVM 이 존재할 경우 아이템재고와 보유 현금이 올바르게 변동되는지 DVM 간의 상호작용을 테스트합니다.
- Phase1-12. 총 40 개의 Test 가 진행되었고, 전체 Pass 되었습니다.

No	Phase	Group	Category	Value	Constraint
1	Activate_ 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_L ogin	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 Integer is	[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]
				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	Environment	Program Start	Success	[property SC]

	Amount	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	Success	[property SC]
		Input	Button	Search Another DVM	[if SC] [property SADVM]
				Buy Code	[if SC] [property BC]
				Admin Login	[if SC] [property C]
		Value	Output	Pop up other DVM info	[if SADVM]
				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	Success	[property SC]
		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]				
12	Multiple_action	Environment	DVM	Current DVM	[property CDVM]
				Target DVM	[property TDVM]
		Action	Stock Change	CDVM Item Sold	[if CDVM] [property CIS]
				TDVM Item Sold	[if TDVM] [property TIS]
			Return Change	Cash Over Input	[if CDVM] [if !CIS !TIS] [property COI]
		Value	Stock	item decrease	[if CIS TIS] [if !COI]
				Cash decrease.	[if COI]
			Cash	Cash increase	[if CIS TIS]

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	Category 1 (Mode On)	Category 2 (Button)	Category 3 (Button2)	Category 4 (Button3)	Category 5 (Name)	Category 6 (Amount)	Category 7 (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

Phase12. Multiple_action

No	Category1 (DVM)	Category2 (Stock Change)	Category3 (Return Change)	Category4 (Stock)	Category5 (Cash)	Test Result
38	Current DVM	CDVM Item Sold	N/A	item decrease	Cash increase	pass
39	Current DVM	N/A	Cash Over Input	N/A	Cash decrease.	pass
40	Target DVM	TDVM Item Sold	N/A	item decrease	Cash increase	pass

4. Brute Force Testing

- 1st Cycle 에서 모호한 Description 으로 인해 Test Case 에 대한 진행이 완만하지 못한 점을 참고하여 Description 을 전반적으로 자세하게 수정하였습니다.

- 총 12 개의 Test 가 진행되었고, 전체 Pass 되었습니다.

4.1. Test Result

No	Test Case	Description	Result
1	DVM 11 대 생성 시 생성 불가 메시지 출력	- DVM 을 11 대까지 생성하는 경우, '자판기는 최대 10 개까지만 생성 가능합니다'의 메시지가 출력된다. - '유효하지 않은 DVM'의 메시지 창과 함께 11 번째 DVM 은 생성되지 않고, 10 개의 DVM 만 생성된다.	Pass
2	DVM 생성 시 20 종류 미만의 음료 설정 시 생성 불가 메시지 출력	- DVM 생성 시 아이템리스트에 20 개 미만의 음료 종류를 포함하는 경우, '아이템리스트는 반드시 20 개의 아이템을 보유해야 합니다'의 메시지가 출력된다. - '유효하지 않은 DVM'의 메시지 창과 함께 DVM 이 생성되지 않는다.	Pass
3	DVM 생성 시 8 종류 이상의 음료 재고를 0 으로 설정 시 생성 불가 메시지 출력	DVM 생성 시 재고가 0 인 음료가 8 개 이상인 경우, '재고가 유효한 상품이 7 개가 아닌 자판기는 생성할 수 없습니다'의 메시지가 출력된다. - '유효하지 않은 DVM'의 메시지 창과 함께 DVM 이 생성되지 않는다.	Pass
4	DVM 이 1 개일 때, 재고가 없는 음료 선택 시 '재고 없음' 메시지 출력	DVM 에서 재고가 없는 음료를 선택하는 경우, '재고가 없는 상품을 선택하여 쿠폰 구입만 가능합니다'의 메시지 창이 출력된다.	Pass
5	DVM 이 2 개일 때, 모든 DVM 에서 재고가 없는 음료 선택 시 '재고 없음' 메시지 출력	- DVM 에서 재고가 없는 음료를 선택하는 경우, '재고가 없는 상품을 선택하여 쿠폰 구입만 가능합니다'의 메시지 창이 출력된다. - '다른 DVM 확인'을 클릭한 뒤 '근처에 구입가능한 자판기가 없습니다'의 메시지 창 출력을 확인한다.	Pass

6	DVM 이 2 개일 때, 재고가 0 인 DVM 에서 재고가 1->0 개로 변하는 DVM 에 접근 시 '재고 없음' 메시지 출력	<ul style="list-style-type: none"> - Main DVM 에서 재고가 0 개인 음료 A 에 대한 구매를 클릭하고, '재고가 없는 상품을 선택하여 쿠폰 구입만 가능합니다' 메시지 창에서 확인을 누른다. - '다른 DVM 확인'을 클릭한 뒤 구입 가능한 자판기에 Another DVM1 가 있음을 확인한다. - Another DVM1 에서 음료 A 를 구매한다. - Main DVM 에서 '다른 DVM 확인'을 클릭한 뒤 '근처에 구입가능한 자판기가 없습니다'의 메시지 창 출력을 확인한다. 	Pass
7	DVM 이 2 개일 때, 재고가 0 인 DVM 에서 재고가 0->1 개로 변하는 DVM 에 접근 시 거리 확인 가능	<ul style="list-style-type: none"> - Main DVM 에서 재고가 0 개인 음료 A 에 대한 구매를 클릭하고, '재고가 없는 상품을 선택하여 쿠폰 구입만 가능합니다' 메시지 창에서 확인을 누른다. - '다른 DVM 확인'을 클릭한 뒤 구입 가능한 자판기에 Another DVM1 가 없음을 확인한다. - Another DVM1 에서 음료 A 에 대한 재고를 변경한다. - Main DVM 에서 '다른 DVM 확인'을 클릭한 뒤 'Another DVM1'의 거리 정보에 대한 메시지 창 출력을 확인한다. 	Pass
8	DVM 이 1 개일 때, 카드에 대한 한도 초과 확인	<ul style="list-style-type: none"> - 한도가 4000 인 카드로 DVM 에서 4000 만큼의 음료를 구매한다. - 1000 인 음료를 구매했을 때, '카드 한도 초과입니다' 메시지 창을 확인한다. 	Pass
9	DVM 이 2 개일 때, 카드에 대한 한도 초과 확인	<ul style="list-style-type: none"> - 한도가 4000 인 카드로 Main DVM 에서 2000, Another DVM1 에서 2000 만큼의 음료를 구매한다. - Main DVM 에서 1000 인 음료를 구매했을 때, '카드 한도 초과입니다' 메시지 창을 확인한다. 	Pass
10	DVM 이 1 개일 때, 음료 구매 시 현금보유량 변화 확인	관리자 메뉴에서 자판기의 현금보유량을 500 으로 설정한 뒤 가격이 1000 인 음료를 구매했을 때, 현금보유량의 증가를 확인한다.	Pass
11	DVM 이 2 개일 때, 인증 코드를 통한 구매 가능여부 확인	<ul style="list-style-type: none"> - Main DVM 에서 재고가 없는 음료에 대한 인증 코드를 구입한 뒤 인증 코드를 메시지 창에서 확인한다. - 확인한 인증 코드를 Another DVM1 의 'Main menu - 쿠폰 사용'을 통해 입력한 뒤, 성공적으로 구매됨을 확인한다. 	Pass

12	DVM 에서 반복적으로 음료를 구매하는 경우, 오류가 발생하는지 확인	DVM 에서 특정 음료에 대한 구매를 재고 수량만큼 반복하는 과정에서 오류가 발생하는 지 확인한다.	Pass
----	--	---	------

5. Static Analysis

- GUI 쪽에서 발생한 오류는 고려하지 않았습니다.

5.1. Maintainability

- Code Smell 을 고칠 때, GUI 부분을 제외한 부분만을 살핀다면 DVM – 3h 2min, Item – 10min, Payment – 47min 으로 총 3h 59min 의 수정시간을 예상합니다.

Reliability
Security
Security Review
Maintainability **NEW CODE SM...**

Overview
On new code
Code Smells 28
Debt 3h 2min
Debt Ratio 2.8%
Rating A
Overall
Code Smells 28
Debt 3h 2min
Debt Ratio 2.8%
Rating A
Effort to Reach A 0

New Code Smells 28

teamv_testing
src/main/java/DVM.java

```

1 -- import java.util.*;
2 --
3 --
4 public class DVM {
5     String region;
6     Pair location;
7     int totalCash;
8     String adminID;
9     String adminPW;
10    ArrayList<Item> itemList;
11    double dist;
12    static HashMap<String,String> codeTable = new HashMap<>();
13    static ArrayList<DVM> DVMList = new ArrayList<>();
14    Payment myPayment;
15
16    public class Pair{
17        int x;
  
```

Reliability
Security
Security Review
Maintainability **NEW CODE SM...**

Overview
On new code
Code Smells 1
Debt 10min
Debt Ratio 0.9%
Rating A
Overall
Code Smells 1
Debt 10min
Debt Ratio 0.9%
Rating A
Effort to Reach A 0

New Code Smells 1

teamv_testing
src/main/java/Item.java

```

1 -- public class Item {
2     String itemName;
3     int itemID;
4     int itemPrice;
5     int itemAmount;
6
7     Item() {
8         this.itemName = "null";
9         this.itemID = -1;
10        this.itemPrice = 0;
11        this.itemAmount = 0;
12    }
13
14    public Item(String itemName, int itemID, int itemPrice, int itemAmount) {
15        this.itemName = itemName;
16        this.itemID = itemID;
  
```

Reliability
Security
Security Review
Maintainability **NEW CODE SM...**

Overview
On new code
Code Smells 4
Debt 47min
Debt Ratio 2.4%
Rating A
Overall
Code Smells 4
Debt 47min
Debt Ratio 2.5%
Rating A
Effort to Reach A 0

New Code Smells 4

teamv_testing
src/main/java/Payment.java

```

1 -- import java.util.*;
2 --
3 --
4 public class Payment {
5     static HashMap<String, Integer> validCardList = new HashMap<>() {{
6         put("1234", 999999);
7         put("5678", 200000);
8         put("4543", 999999);
9         put("3572", 1000);
10        put("1589", 999999);
11        put("3257", 200000);
12        put("1888", 999999);
13        put("0088", 4000);
14    }};
15
16    public Payment() {
  
```

5.2. Complexity

- Cyclomatic Complexity(순환 복잡도)가 51 이상인 경우, 테스트 불가능한 프로그램으로 위험성이 매우 높다고 할 수 있습니다. (1-10 단순 프로그램, 위험성 낮음/11-20 다소 복잡한 프로그램, 위험성 중간/21-50 복잡한 프로그램, 위험성 중간/51 이상 테스트 불가능한 프로그램, 위험성 매우 높음)
- Item.java 와 Payment.java 는 각각 9 와 13 으로 양호합니다. 하지만 DVM.java 의 경우 64 로 매우 높은 복잡도를 보입니다.

Cyclomatic Complexity 162	
AdminMenuGUI.java	5
CardGUI.java	8
CashGUI.java	9
DVM.java	64
Item.java	9
ItemListGUI.java	6
LoginGUI.java	5
Main.java	1
MainGUI.java	7
ManageAmountGUI.java	7
ManageCashGUI.java	7
NotEnoughAmountGUI.java	9
Payment.java	13
SelectPaymentGUI.java	5
UseCouponGUI.java	7

15 of 15 shown

- DVM.java 내부의 코드입니다. 조건문이 반복되서 사용되므로 수정을 통해 복잡도를 낮추어야하고, 이 Issue 에 대해서는 5.3. Issue List No.5 에서 처리 요구하였습니다.

```

public DVM(String region, int x, int y, int totalCash, String adminID, String adminPW, ArrayL
    if (DVMList.size() == 10) {
        System.out.println("자판기는 최대 10개까지만 생성 가능합니다.");
        System.out.println "[" + region + "]자판기는 생성되지 않았습니다.");
    } else if (itemList.size() != 20) {
        System.out.println("아이템리스트는 반드시 20개의 아이템을 보유해야 합니다.");
        System.out.println "[" + region + "]자판기는 생성되지 않았습니다.");
    } else {
        int availableItem = 0;
        for (int i=0; i<20; i++) {
            if (itemList.get(i).getItemAmount() > 0) {
                availableItem += 1;
            }
        }
    }
}

```

```

    }
}
if (availableItem != 7) {
    System.out.println("재고가 유효한 상품이 7개가 아닌 자판기는 생성할 수 없습니다.");
    System.out.println("[ " + region + "]자판기는 생성되지 않았습니다.");
} else {
    this.region = region;
    this.location = new Pair(x,y);
    this.totalCash = totalCash;
    this.adminID = adminID;
    this.adminPW = adminPW;
    this.itemList = itemList;
    this.dist = dist;
    this.myPayment = myPayment;
    boolean duplicated = false;
    for(int i=0;i<DVMList.size();i++){
        if(DVMList.get(i).location.x == x && DVMList.get(i).location.y ==y){
            duplicated=true;
        }
    }
}

```

5.3. Issue List

- 총 7 개의 Issue 를 선정하였습니다.

No	File	Line	Rule	Severity	Description
1	DVM.java	133	Bug	Minor	Math.pow 함수의 입력은 double형이어야 합니다.
2		26	Code Smell	Major	생성자의 파라미터가 너무 많습니다.
3		247	Bug	Major	문자열 비교에 ==을 사용하지 마세요.
4		29	Code Smell	Critical	String이 반복적으로 나타나고 있습니다.
5		26	Code Smell	Critical	조건문과 반복문으로 인해 코드가 복잡합니다.

6		228, 251	Code Smell	Minor	Boolean 대신 boolean 을 사용하세요. Boolean 사용 시 NPE 대비할 수 있도록 코드 작성해야 합니다.
7	Payment.java	13	Bug	Minor	Double Brace Intialization은 권장되지 않 습니다.

5.4. Issue Upload to Jira

- Jira 에 7 개의 Issue 를 등록하였고, Jira 와 sonarqube(Static Analysis tool)가 연동되어있어 Issue 클릭 시 해당되는 소스코드를 확인할 수 있습니다.

프로젝트 /  konkuk_VV_t_6_2021 / KVT62-126 /  KVT62-144

Static Analysis

 첨부  하위 작업 생성  이슈 연결 ▼ ⋮

설명
GUI를 제외한 DVM, Payment, Item.java에서의 Static Analysis Issue에 대해서만 Issue를 등록하였습니다.
해당 Issue 클릭 시 Sonarqube로 연결되고, 로그인 후 위치 확인 가능합니다!

링크된 이슈 +

blocks

<input checked="" type="checkbox"/>	KVT62-12 1. Math operands should be cast before assignment	↑		BACKLOG ▼
<input checked="" type="checkbox"/>	KVT62-131 5. Cognitive Complexity of methods should not be too high	↑		BACKLOG ▼
<input checked="" type="checkbox"/>	KVT62-25 2. Methods should not have too many parameters	↑		BACKLOG ▼
<input checked="" type="checkbox"/>	KVT62-127 3. Strings and Boxed types should be compared using "equals()"	↑		BACKLOG ▼
<input checked="" type="checkbox"/>	KVT62-129 4. String literals should not be duplicated	↑		BACKLOG ▼
<input checked="" type="checkbox"/>	KVT62-137 6. Boxed "Boolean" should be avoided in boolean expressions	↑		BACKLOG ▼
<input checked="" type="checkbox"/>	KVT62-13 7. Double Brace Initialization should not be used	↑		BACKLOG ▼

6. Overall – 2nd Cycle

6.1. Test Quality

- 1st Cycle 이 종료된 후의 전체적인 V&V Test 에 대한 Quality 는 아래와 같습니다.

	Specification Review	Category Partition Test	Brute Force Test	Static Analysis Issue	Total
Passed / All Test Case	31/37	40/40	12/12	0/7	83/96

- Jira 에 Issue 는 Static Analysis Issue 인 13 개가 등록되어있고, 개발팀에 수정요구하였습니다.

The screenshot shows a Jira Kanban board with the following details:

- Project:** konkuk_VV_t_6_2021 / KVT62 보드
- Board Name:** 칸반 보드
- Columns:**
 - 디버깅이슈 8
 - 스펙리뷰 7
 - 시스템테스팅 0
 - 진행 중 1
- Issues in Progress (5 items):**
 - Static Analysis (2nd Cycle) - KVT62-144
 - Specification Review (2nd Cycle) - KVT62-145
 - 2nd Cycle - KVT62-126
 - 1. Math operands should be cast before assignment - KVT62-12
 - 1. Requirement – R2.3 Give DVM Info (2nd Cycle) - KVT62-146
- Other Issues (132 items):**
 - 7. Double Brace Initialization should not be used
 - 2. Use Case – R2.3 Give DVM Info

6.2. Feedback

- DVM 에 대한 기능이 Test 가 가능한 범위 내에서 잘 구현되어있고, DVM 간 재고처리, 쿠폰 구입 등의 상호작용이 문제 없이 잘 구현되어있습니다.

- DVM.java 에서 나타나는 Static analysis Issue, 특히 조건문의 반복적인 사용에 대해서는 수정이 필요합니다.