

OOPT Stage 2050 & 2060

<Construct & Testing>

Version 1.0

Project Team

T7 Team

Date

2017-05-24

Team Information

201414134 오세욱

201414136 임현유

201211375 임동현

201211387 하헌규

Table of Contents

1	Activity 2051. Implement Class & Methods Definitions.....	3
2	Activity 2052. Implement Windows.....	21
3	Activity 2055. Write Unit Test Code.....오류! 책갈피가 정의되어 있지 않습니다.	
4	Activity 2061. Unit Testing.....	33
5	Activity 2063. System Testing.....	41
6	Activity 2067. Testing Traceability Analysis.....오류! 책갈피가 정의되어 있지 않습니다.	

1. Activity 2051. Implement Class & Methods Definitions

Type	Class
Name	Input_Manager
Purpose	Actor 의 Input 을 수행하는 클래스
Overview	Category, RV 객체 관리, TestCase 계산, 초기화 등을 제어한다.
Cross Reference	Functions : ALL Use Cases : ALL
Exceptional Courses of Events	N/A

Type	Class
Name	Output_Manager
Purpose	Input 에 대한 결과 값을 출력하는 클래스
Overview	Test Case 결과를 File 로 출력, Selected_TC 를 화면에 출력한다.
Cross Reference	Functions : R2.1, R3.1 Use Cases : TestCase 계산, 초기화
Exceptional Courses of Events	N/A

Type	Class
Name	Calculate
Purpose	TestCase 계산 클래스

Overview	현재 Category, RV 의 상태에 따라 TestCase 를 계산한다.
Cross Reference	Functions : R2.1 Use Cases : TestCase 계산
Exceptional Courses of Events	N/A

Type	Class
Name	Category
Purpose	Category 클래스
Overview	Category 데이터 관리와 관련 기능을 한다.
Cross Reference	Functions : R1.2, R1.4 Use Cases : Category 수정, 대푯값 추가
Exceptional Courses of Events	N/A

Type	Class
Name	RV
Purpose	RV 클래스
Overview	RV 데이터 관리와 관련 기능을 한다.
Cross Reference	Functions : R1.5, R1.7, R1.8 Use Cases : 대푯값 수정, Constraint 추가, Costraint 삭제
Exceptional Courses of Events	N/A

Type	Class
-------------	-------

Name	TC
Purpose	TC 클래스
Overview	TC 데이터 관리와 관련 기능, TestCase 계산 관련 기능을 한다.
Cross Reference	Functions : R2.1 Use Cases : Test Case 계산
Exceptional Courses of Events	N/A

Type	Method
Name	Search_Category
Purpose	전체 Category 중 특정 Category 를 찾는다.
Cross Reference	Functions : R1.2, R1.4, R1.5, R1.6, R1.7, R1.8 Use Cases : Category 수정, 대푯값 추가, 대푯값 수정, 대푯값 삭제, Constraint 추가, Constraint 삭제
Input	int Category_Id
Output	Category Category
Abstract Operation	Category_Id 를 받아서, 일치하는 Category 를 반환한다.
Exceptional Courses of Events	N/A

Type	Method
Name	Add_Category
Purpose	Category객체를 추가한다.

Cross Reference	Functions : R1.1 Use Cases : Category 추가
Input	void
Output	void
Abstract Operation	Category_List에 Category객체를 추가한다.
Exceptional Courses of Events	N/A

Type	Method
Name	Change_Category
Purpose	Category 객체의 이름을 바꾼다.
Cross Reference	Functions : R1.2 Use Cases : Category 수정
Input	int Category_Id, String Category_Name
Output	void
Abstract Operation	1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 객체의 이름을 Category_Name으로 바꾼다.
Exceptional Courses of Events	Category_Name이 ""라면 이름을 변경하지 않는다.

Type	Method
Name	Delete_Category
Purpose	Category 객체를 삭제한다.

Cross Reference	Functions : R1.3 Use Cases : Category 삭제
Input	int Category_Id
Output	void
Abstract Operation	1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 Category 객체를 삭제한다.
Exceptional Courses of Events	Category에 Property가 존재하면 삭제하지 않는다.

Type	Method
Name	Add_RV
Purpose	RV 객체를 추가한다.
Cross Reference	Functions : R1.4 Use Cases : 대폿값 추가
Input	int Category_Id
Output	void
Abstract Operation	1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 Category 객체에 RV 객체를 추가한다.
Exceptional Courses of Events	N/A

Type	Method
Name	Change_RV

Purpose	RV 객체의 이름을 변경한다.
Cross Reference	Functions : R1.5 Use Cases : 대폿값 수정
Input	int Category_Id, Int RV_Id, String RV_Name
Output	void
Abstract Operation	1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 Category 객체에서 RV 객체를 찾는다. 3. RV 객체의 이름을 변경한다.
Exceptional Courses of Events	RV_Name이 ""라면 이름을 변경하지 않는다.

Type	Method
Name	Delete_RV
Purpose	RV 객체를 삭제한다.
Cross Reference	Functions : R1.6 Use Cases : 대폿값 삭제
Input	int Category_Id, int RV_Id
Output	void
Abstract Operation	1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 Category 객체에서 RV 객체를 찾는다. 3. 찾는 RV객체를 삭제한다.
Exceptional Courses of Events	N/A

Type	Method
Name	Add_Constraint
Purpose	Constraint를 추가한다.
Cross Reference	Functions : R1.7 Use Cases : Constraint 추가
Input	int Category_Id, int RV_Id, String Input_Constraint, String Constraint_Name
Output	void
Abstract Operation	<ol style="list-style-type: none"> 1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 Category 객체에서 RV 객체를 찾는다. 3. 찾은 RV객체에 Input_Constraint와 Constraint_Name에 따라 Constraint를 추가한다.
Exceptional Courses of Events	Property중 Constraint_Name이 존재하지 않는 If Property는 추가하지 않는다.

Type	Method
Name	Delete_Constraint
Purpose	Constraint를 삭제한다.
Cross Reference	Functions : R1.8 Use Cases : Constraint 삭제
Input	int Category_Id, int RV_Id, String Select_Constraint,

	String Constraint_Name
Output	void
Abstract Operation	<ol style="list-style-type: none"> 1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 Category 객체에서 RV 객체를 찾는다. 3. 찾는 RV객체에 Select_Constraint와 Constraint_Name에 따라 Constraint를 삭제한다.
Exceptional Courses of Events	N/A

Type	Method
Name	Choice_RV
Purpose	선택한 Constraint와 관련된 RV를 기억한다.
Cross Reference	<p>Functions : R1.7, R1.8</p> <p>Use Cases : Constraint 추가, Constraint 삭제</p>
Input	int RV_Id, int Category_Id
Output	void
Abstract Operation	<ol style="list-style-type: none"> 1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 Category 객체에서 RV 객체를 찾는다. 3. 찾는 RV객체를 기억한다.
Exceptional Courses of Events	N/A

Type	Method
-------------	--------

Name	Select_RV
Purpose	선택한 RV의 선택 여부를 변경한다.
Cross Reference	Functions : R1.9 Use Cases : 대폿값 선택
Input	int RV_Id, int Category_Id
Output	void
Abstract Operation	1. Category 객체중 Id 가 Category_Id와 일치하는 객체를 찾는다. 2. 찾은 Category 객체에서 RV 객체를 찾는다. 3. 찾는 RV객체의 선택 여부를 변경한다.
Exceptional Courses of Events	N/A

Type	Method
Name	Calculate
Purpose	Test Case를 계산한다.
Cross Reference	Functions : R2.1 Use Cases : Test Case 계산
Input	void
Output	void
Abstract Operation	Test Case를 계산한다.
Exceptional Courses of Events	RV가 존재하지 않는 Category가 있으면 계산하지 않는다.

Type	Method
Name	Clear
Purpose	초기화한다.
Cross Reference	Functions : R3.1 Use Cases : 초기화
Input	void
Output	void
Abstract Operation	1. Output_Manager를 Clear한다. 2. Input_Manager를 Clear한다.
Exceptional Courses of Events	N/A

Type	Method
Name	Delete_If
Purpose	If Constraint를 삭제한다.
Cross Reference	Functions : R1.6, R1.8 Use Cases : 대폿값 삭제, Constraint 삭제
Input	List<String> Property_List
Output	void
Abstract Operation	전체 Category 객체에서 Property_List와 관련된 If Constraint를 삭제한다.
Exceptional Courses of Events	Input이 String이라면 1개의 Property와 관련된 If_Constraint를 삭제한다.

Type	Method
Name	Search_Property_List
Purpose	Property_List에 Constraint_Name이 존재하는지 확인한다.
Cross Reference	Functions : R1.7 Use Cases : Constraint 추가
Input	String Constraint_Name
Output	Boolean
Abstract Operation	Property_list에서 Constraint_Name을 찾는다.
Exceptional Courses of Events	N/A

Type	Method
Name	Add_Property_List
Purpose	Property_List에 Constraigt_Name을 추가한다.
Cross Reference	Functions : R1.7 Use Cases : Constraint 추가
Input	String Constraint_Name
Output	void
Abstract Operation	Property_List에 Constraint_Name를 추가한다.
Exceptional Courses of Events	N/A

Type	Method
Name	Output_Screen

Purpose	화면에 선택된 TestCase를 출력한다.
Cross Reference	Functions : R2.1 Use Cases : Test Case 계산
Input	List<TC> Selected_TC
Output	void
Abstract Operation	Selected_TC를 받아 화면에 출력한다.
Exceptional Courses of Events	N/A

Name	Output_File
Purpose	결과를 File로 출력
Cross Reference	Functions : R2.1 Use Cases : Test Case 계산
Input	ArrayList All_TC
Output	Void
Abstract Operation	결과를 out.txt text file로 출력
Exceptional Courses of Events	N/A

Type	Method
Name	Output_Clear
Purpose	출력 부분 초기화
Cross Reference	Functions : R3.1 Use Cases : 초기화

Input	void
Output	Void
Abstract Operation	ArrayList Selected_TC 초기화
Exceptional Courses of Events	N/A

Type	Method
Name	setName
Purpose	Category의 이름 변경
Cross Reference	Functions : R1.2 Use Cases : Category 수정
Input	String Category_Name
Output	Void
Abstract Operation	Category의 이름을 Category_Name으로 변경
Exceptional Courses of Events	N/A

Type	Method
Name	Search_RV
Purpose	RV를 Search
Cross Reference	Functions : R1.4, R1.5, R1.6 Use Cases : 대폿값 추가, 대폿값 수정, 대폿값 삭제
Input	Int RV_id
Output	RV RV

Abstract Operation	RV_List 를 순회하면서 RV_id와 일치하는 RV를 반환
Exceptional Courses of Events	N/A

Type	Method
Name	hasProperty
Purpose	Category 내에 Property 값이 존재하는지 확인하기 위한 메소드
Cross Reference	Functions : R1.3 Use Cases : Category 삭제
Input	Int Category_Id
Output	Boolean
Abstract Operation	RV가 갖는 Property List size가 0이 아니면 True 반환
Exceptional Courses of Events	N/A

Type	Method
Name	Create_RV
Purpose	RV를 추가하기 위한 메소드
Cross Reference	Functions : R1.4 Use Cases : 대푯값 추가
Input	N/A
Output	Void
Abstract Operation	RV_List에 RV객체 생성 후 추가
Exceptional Courses of Events	N/A

Type	Method
Name	setName
Purpose	RV_Name 변경하기 위한 메소드
Cross Reference	Functions : R1.5 Use Cases : 대폿값 수정
Input	String RV_Name
Output	void
Abstract Operation	RV_Name을 변경
Exceptional Courses of Events	N/A

Type	Method
Name	getProperty_List
Purpose	현재 RV의 Property list를 반환하기 위한 메소드
Cross Reference	Functions : Use Cases :
Input	N/A
Output	List<String>
Abstract Operation	현재 RV의 Property list를 반환
Exceptional Courses of Events	N/A

Type	Method
-------------	--------

Name	setSingle_Error
Purpose	Input_Constraint에 따라 Single, Error를 정하는 메소드
Cross Reference	Functions : Use Cases :
Input	String Input_Constraint
Output	Void
Abstract Operation	Input_Constraint가 0이면 Single_Error = 0, “Single”이면 Single_Error = 1, “Error”면 Single_Error = 2
Exceptional Courses of Events	N/A

Type	Method
Name	getSingle_Error
Purpose	Single과 Error를 반환하기 위한 메소드
Cross Reference	Functions : Use Cases :
Input	N/A
Output	String
Abstract Operation	Single_Error가 1이면 “Single” 반환, Single_Error 가 2면 “Error” 반환
Exceptional Courses of Events	N/A

Type	Method
Name	Add_Property

Purpose	현재 RV의 property list에 Property 추가
Cross Reference	Functions : R1.7 Use Cases : Constraint 추가
Input	String Constraint_Name
Output	Void
Abstract Operation	RV_Property_List가 입력받은 Constraint_Name을 포함하지 않으면 리스트에 추가
Exceptional Courses of Events	N/A

Type	Method
Name	Add_If
Purpose	현재 RV의 If list에 If추가
Cross Reference	Functions : R1.7 Use Cases : Constraint 추가
Input	String Constraint_Name
Output	Void
Abstract Operation	If_List가 입력받은 Constraint_Name을 포함하지 않으면 리스트에 추가
Exceptional Courses of Events	N/A

Type	Method
Name	Delete_Property
Purpose	현재 RV의 property를 삭제

Cross Reference	Functions : R1.8 Use Cases : Constraint 삭제
Input	String Constraint_Name
Output	Void
Abstract Operation	RV_Property_List size만큼 순회하면서 Constraint_Name과 같은 Property를 삭제
Exceptional Courses of Events	N/A

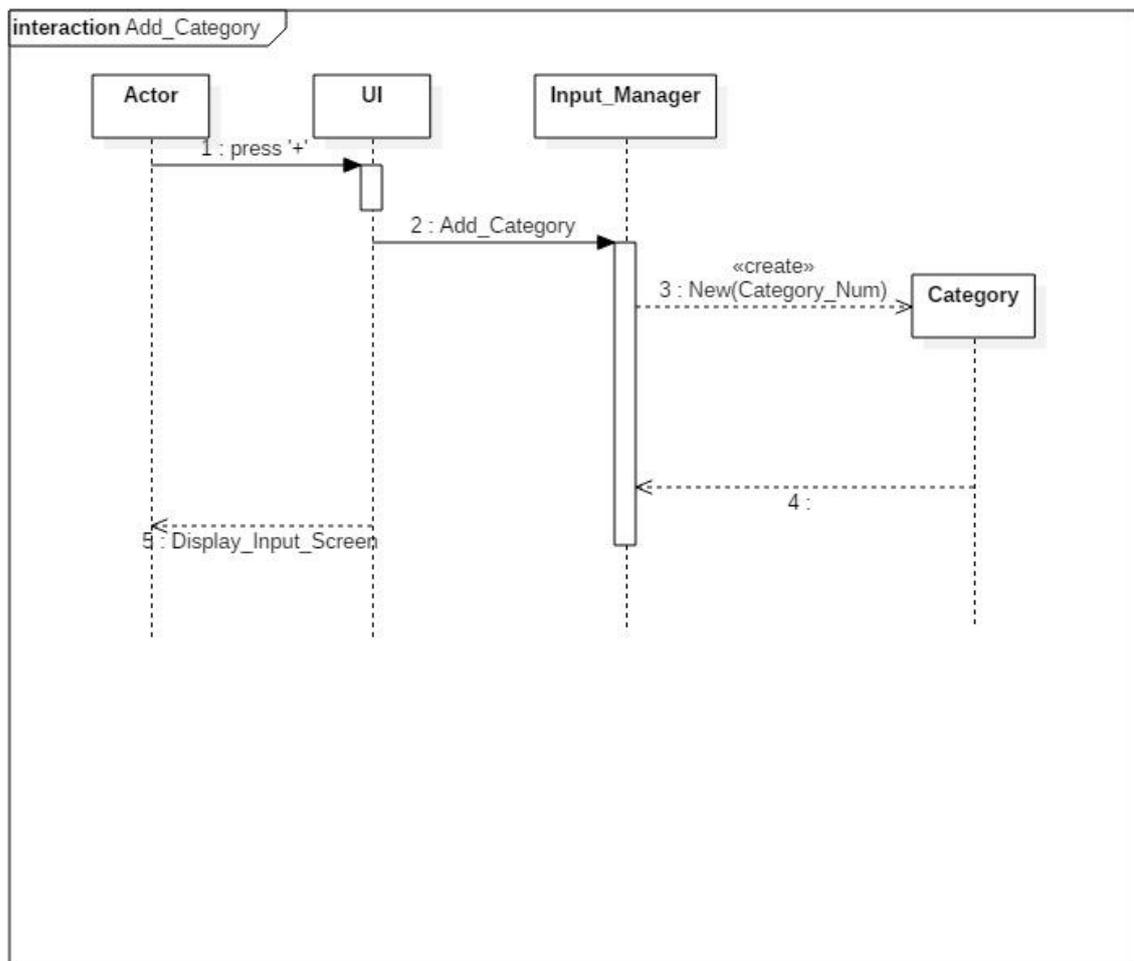
Type	Method
Name	Delete_If
Purpose	현재 RV의 If 를 삭제
Cross Reference	Functions : R1.8 Use Cases : Constraint 삭제
Input	String Constraint_Name
Output	Void
Abstract Operation	If_List size 만큼 순회하면서 Constraint_Name과 같은 If를 삭제
Exceptional Courses of Events	N/A

Type	Method
Name	If_Property_Check
Purpose	Property – If 관계를 확인한다.
Cross Reference	Functions : R1.7 R1.9 Use Cases : Constraint 추가, Test Case 계산

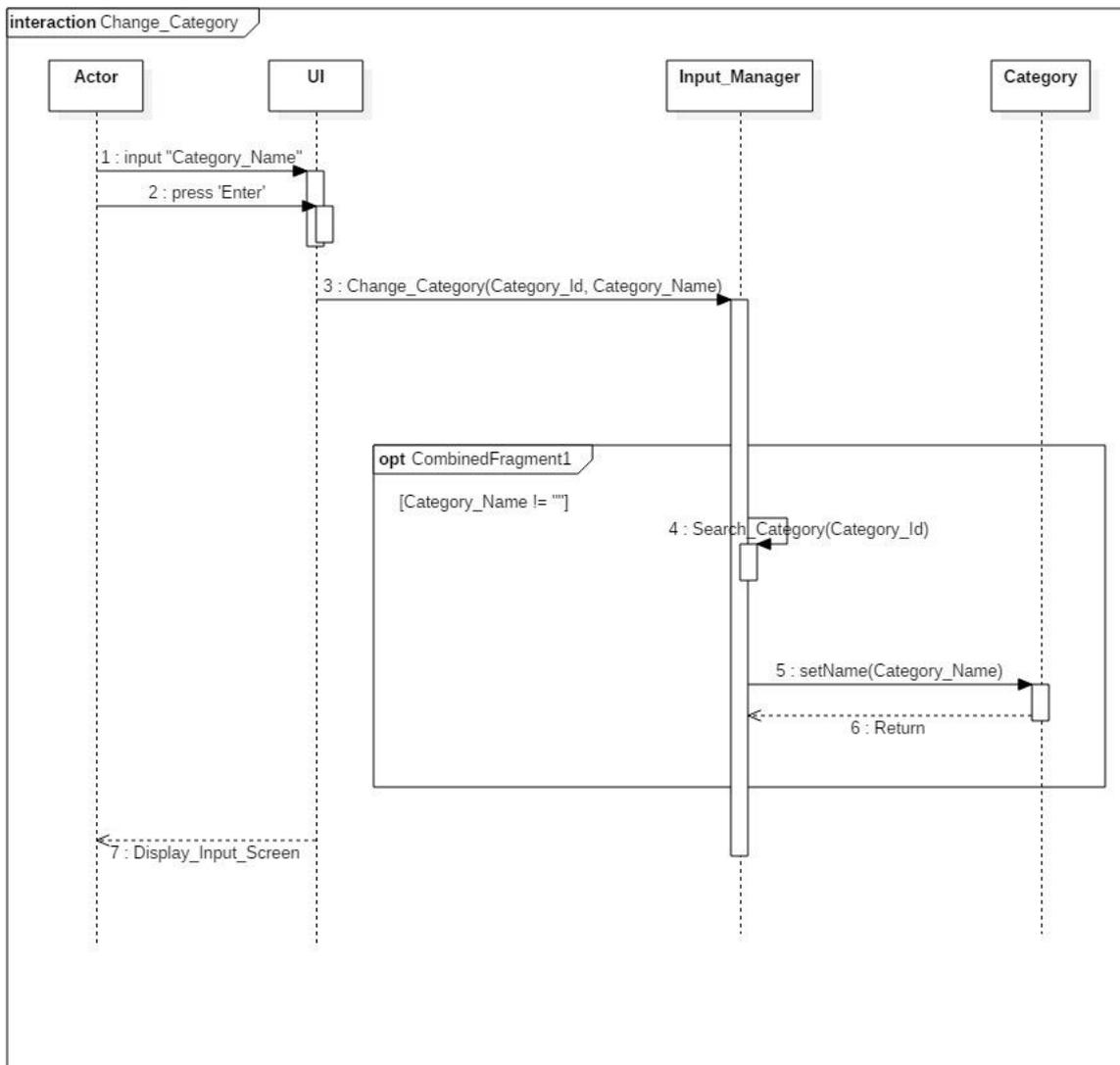
Input	N/A
Output	Boolean
Abstract Operation	If가 갖는 이름에 해당하는 Property 이름이 존재하면 True 반환
Exceptional Courses of Events	N/A

Type	Method
Name	TC_Calculate
Purpose	테스트 케이스를 계산한다.
Cross Reference	Functions : R2.1 Use Cases : Test Case 계산
Input	ArrayList Category_List, ArrayList Selected_RV_List
Output	void
Abstract Operation	Category_List와 Selected_RV_List로 Test Case를 계산한다.
Exceptional Courses of Events	N/A

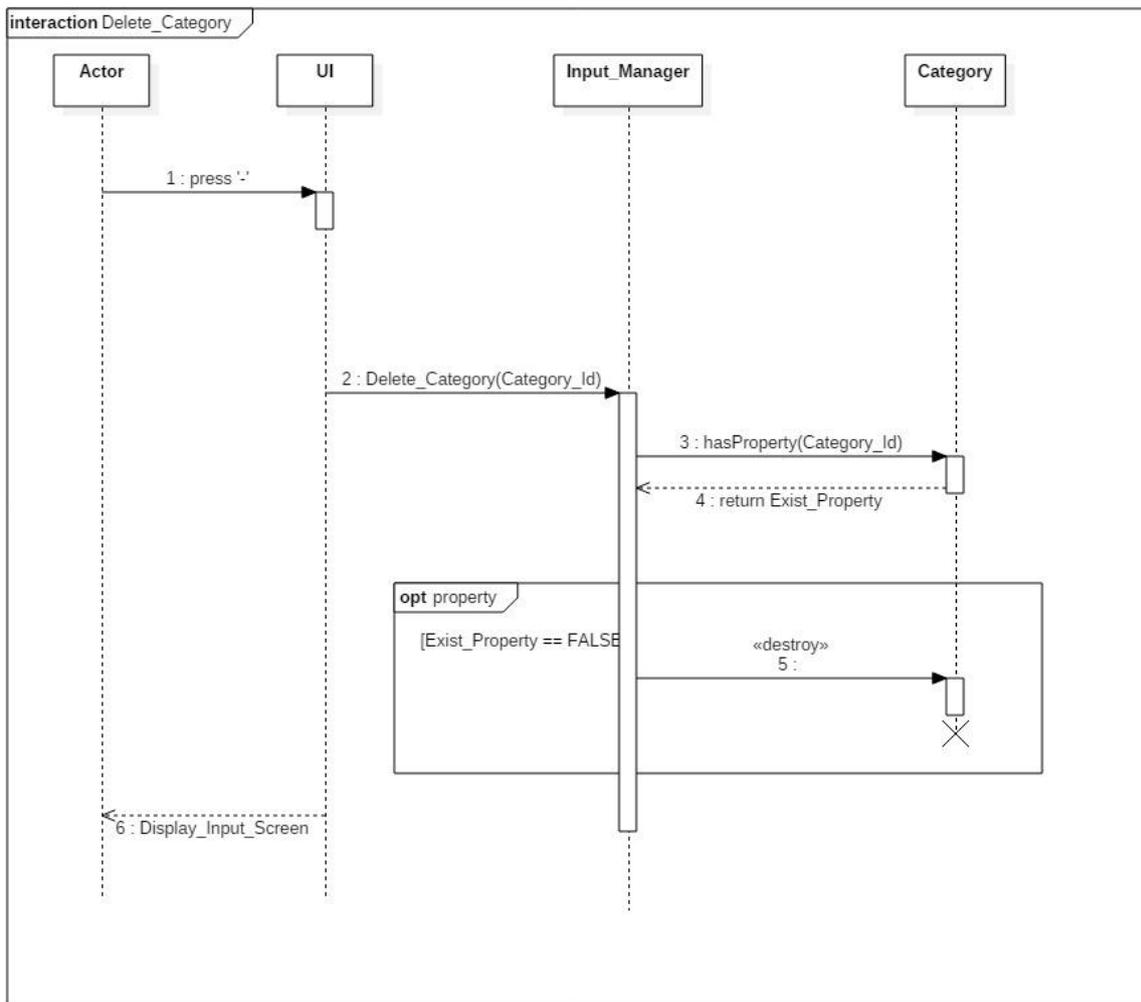
2. Activity 2052. Implement Windows



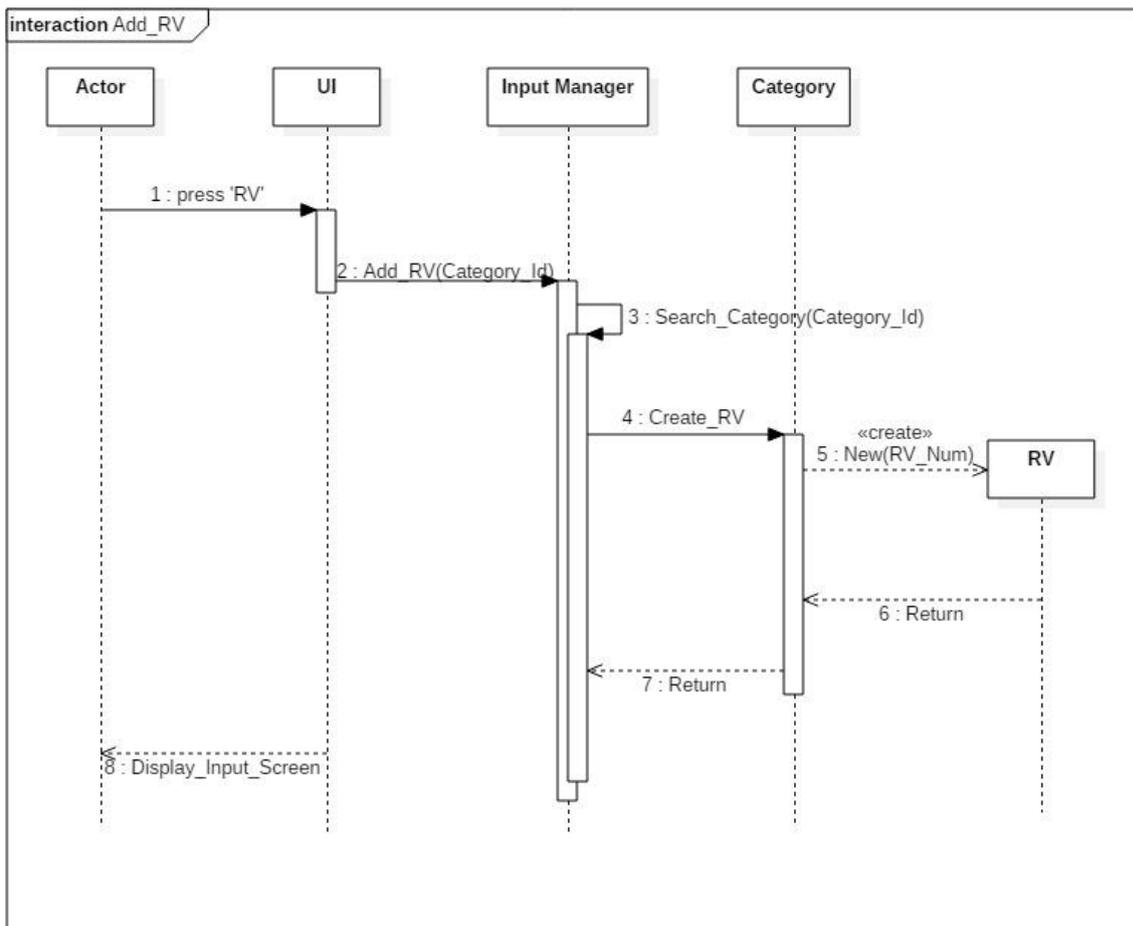
Name	Add_CategoryButton
Responsibilites	Category 추가 버튼을 클릭한다.
Type	GUI
Cross References	R1.1
Notes	Category를 추가해주는 Add_Category()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Add_Category()가 실행된다.



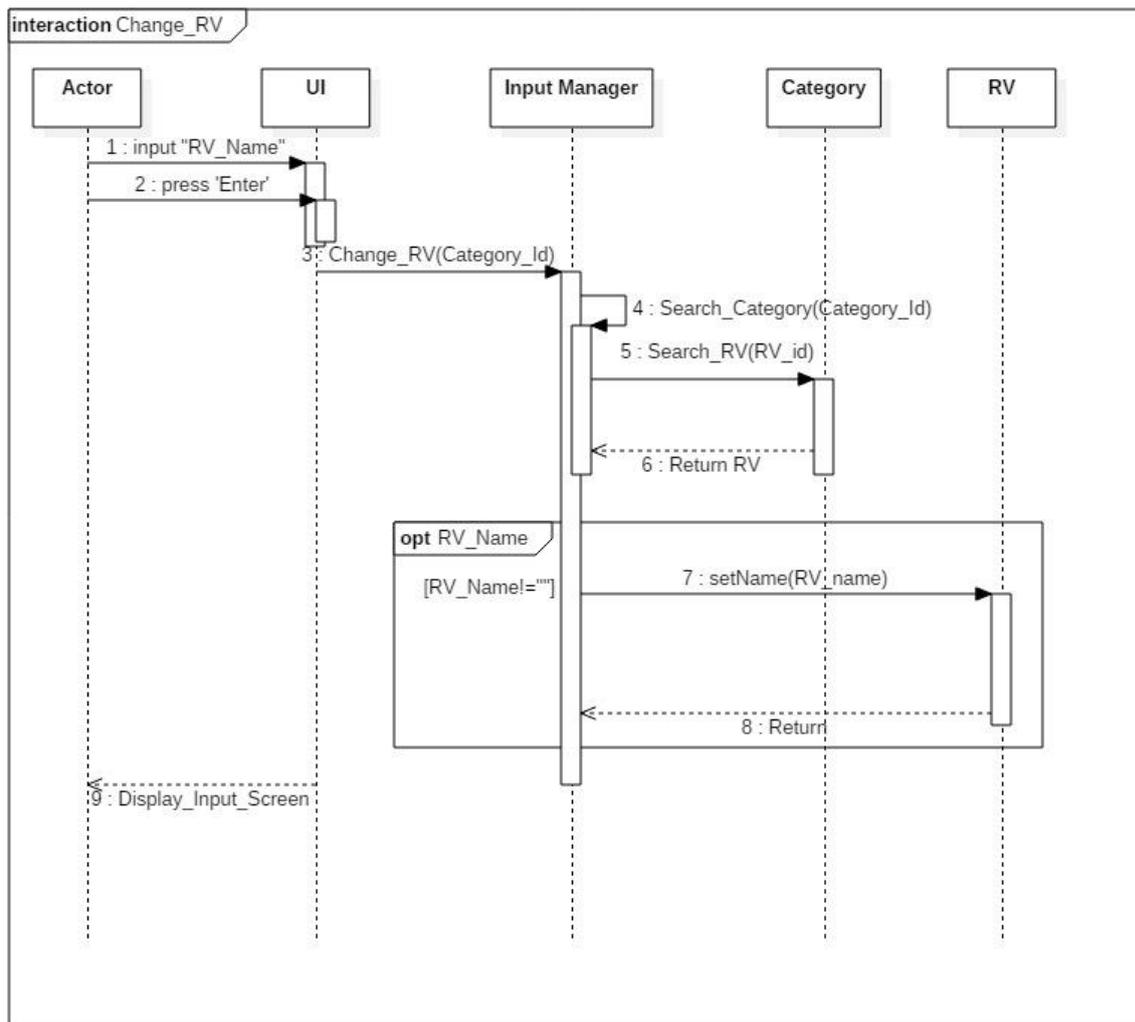
Name	CategorytextField
Responsibilities	Category 이름과 Enter 입력한다.
Type	GUI
Cross References	R1.2
Notes	Category이름을 변경해주는 Change_Category()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Change_Category()가 실행된다.



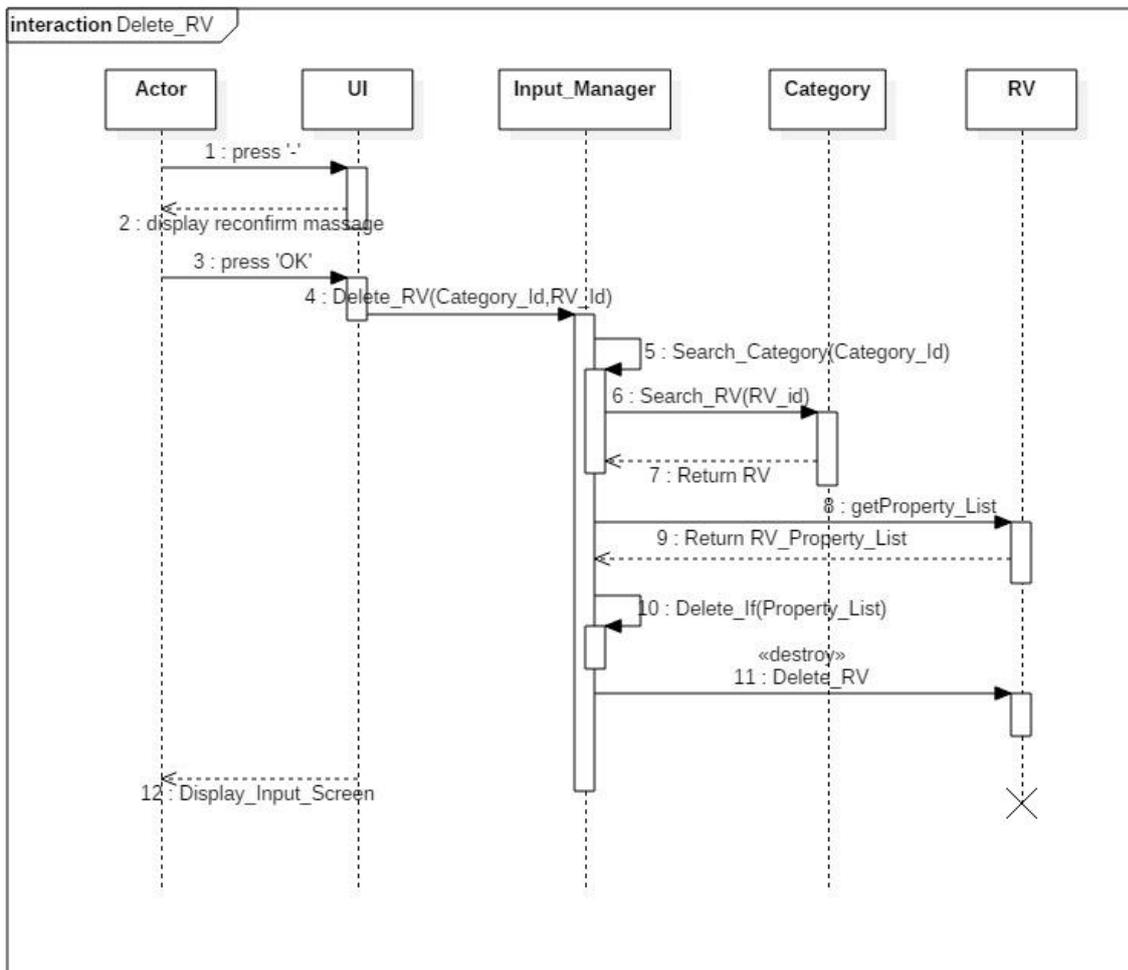
Name	Delete_CategoryButton
Responsibilites	Category 삭제 버튼을 클릭한다.
Type	GUI
Cross References	R1.3
Notes	Category를 삭제해주는 Delete_Category()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Delete_Category()가 실행된다.



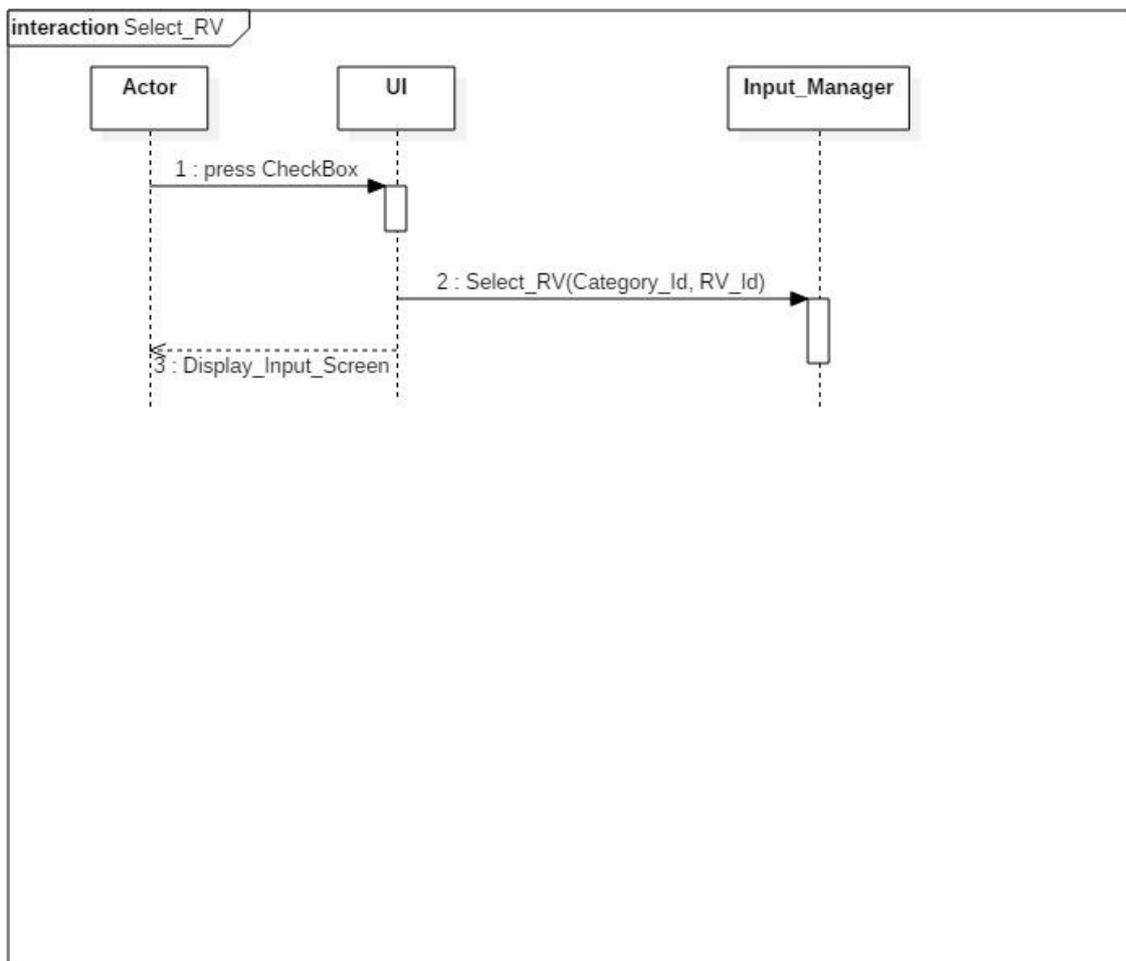
Name	Add_RVButton
Responsibilites	RV 추가 버튼을 클릭한다.
Type	GUI
Cross References	R1.4
Notes	RV를 추가해주는 Add_RV()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Add_RV()가 실행된다.



Name	RVtextField
Responsibilities	RV의 이름과 Enter를 입력한다.
Type	GUI
Cross References	R1.5
Notes	RV의 이름을 변경해주는 Change_RV()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Change_RV()가 실행된다.



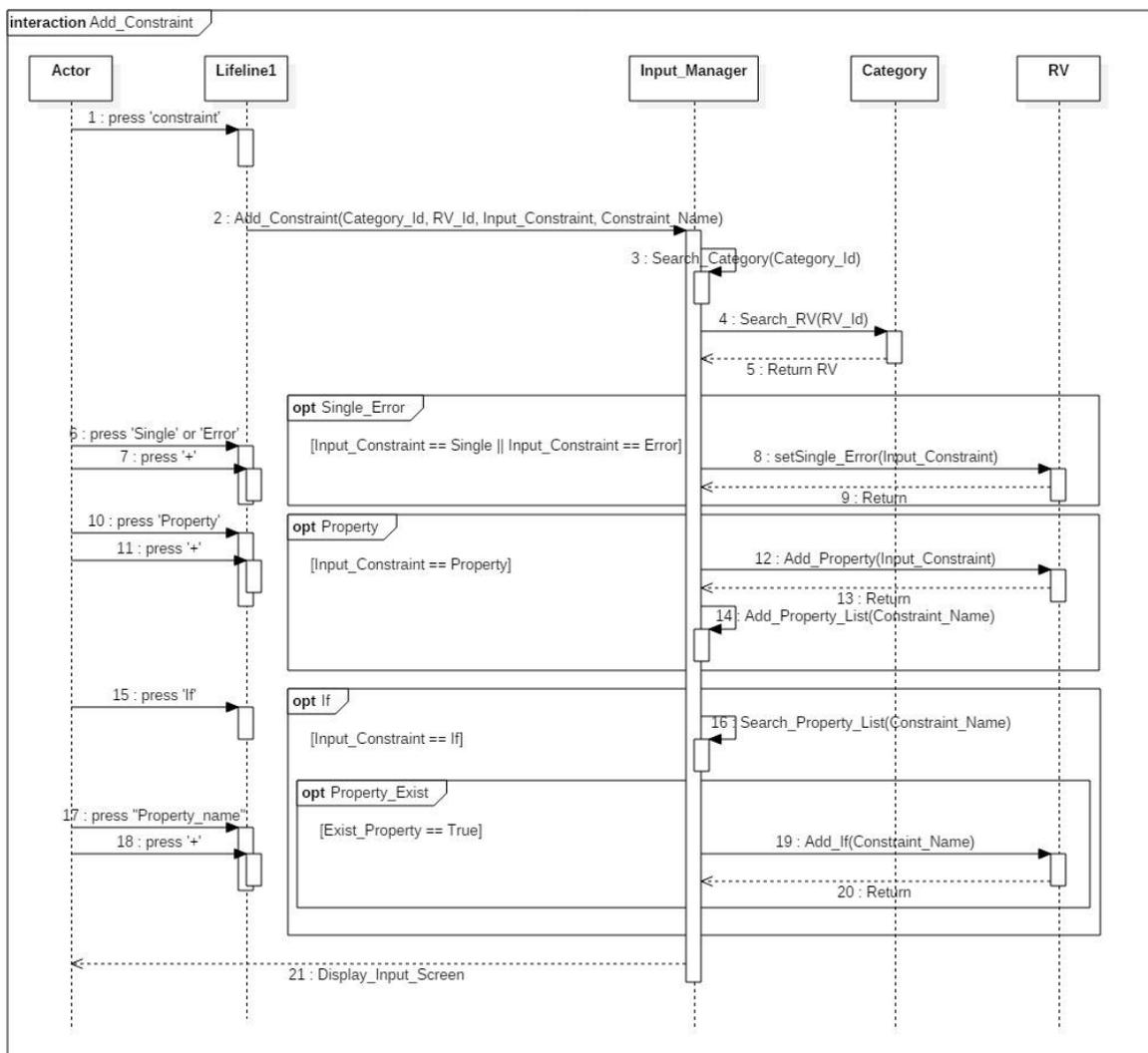
Name	Delete_RVButton
Responsibilities	RV 삭제 버튼을 클릭한다.
Type	GUI
Cross References	R1.6
Notes	RV를 삭제해주는 Delete_RV()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Delete_RV()가 실행된다.



Name	checkbox
Responsibilites	checkbox를 클릭한다.
Type	GUI
Cross References	R1.9
Notes	RV의 선택 여부를 변경해주는 Select_RV()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Select_RV()가 실행된다.

Name	chooseRVbutton
-------------	----------------

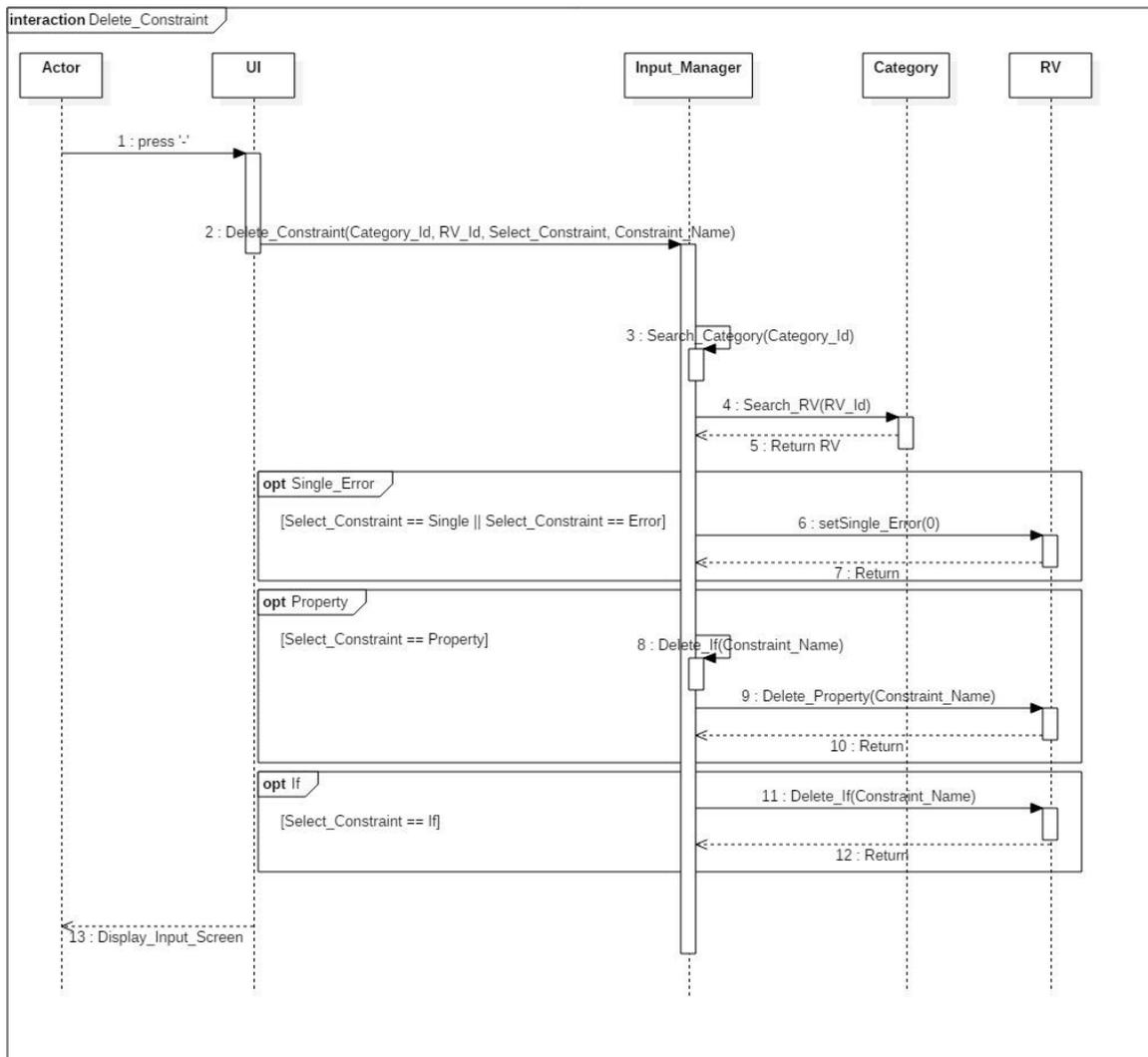
Responsibilities	chooseRV버튼을 클릭한다.
Type	GUI
Cross References	R1.7
Notes	Constraint를 추가할 RV를 선택한다.
Pre-Conditions	N/A
Post-Conditions	선택한 RV가 저장된다.



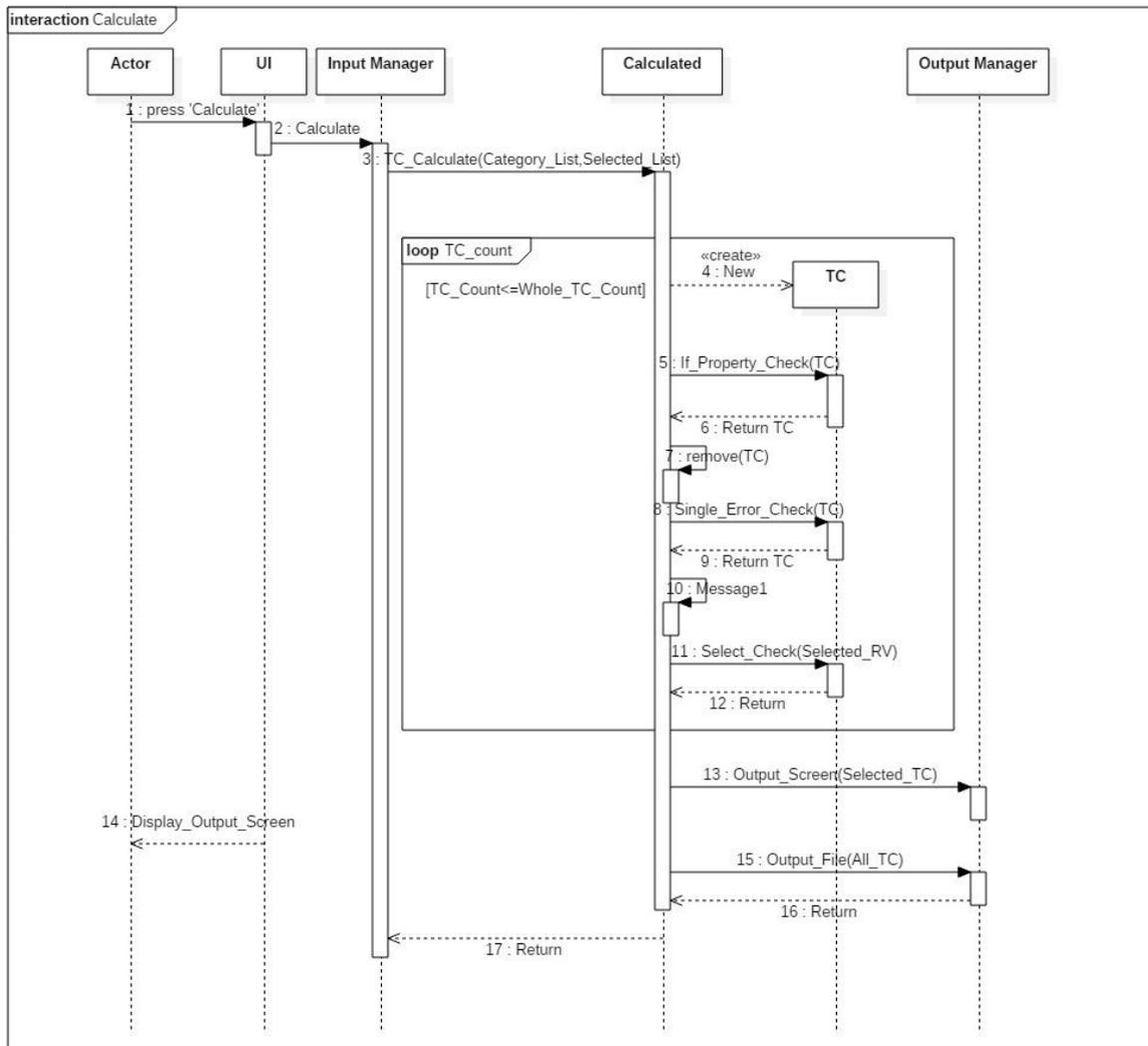
Name	addConstraintButton
-------------	---------------------

Responsibilites	Constraint 추가 버튼을 클릭한다.
Type	GUI
Cross References	R1.7
Notes	Constraint를 추가해주는 Add_Constraint()메서드를 호출한다.
Pre-Conditions	Constraint를 선택해야한다.
Post-Conditions	Add_Constraint()가 실행된다.

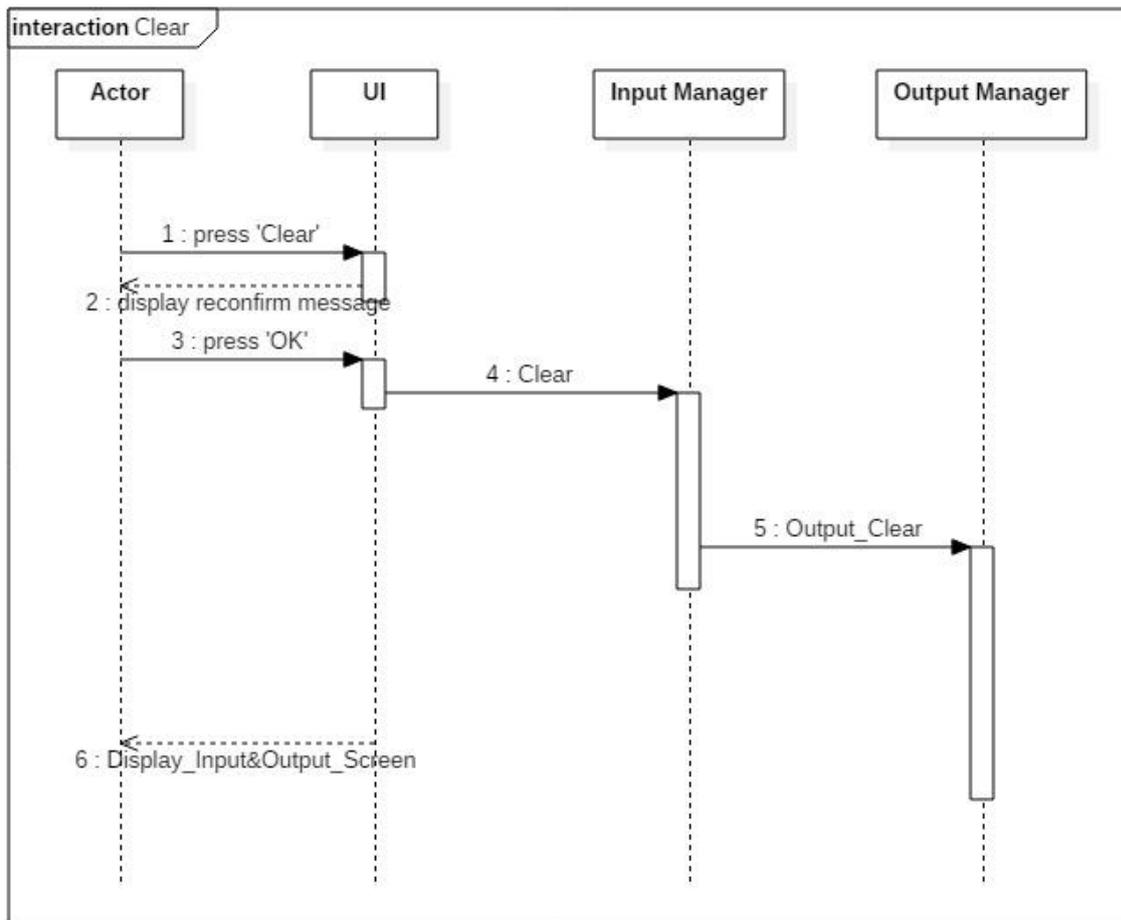
Name	selectConstraintButton
Responsibilites	selectConstraintButton을 클릭하고 Constraint를 선택한다.
Type	GUI
Cross References	R1.7
Notes	선택한 Constraint를 반환한다.
Pre-Conditions	chooseRVbutton을 클릭해야 한다.
Post-Conditions	선택한 Constraint를 저장한다.



Name	Delete_Constraintbutton
Responsibilities	Constraint 삭제 버튼을 클릭한다.
Type	GUI
Cross References	R1.8
Notes	Constraint를 삭제해주는 Delete_Constraint메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Delete_Constraint()가 실행된다.



Name	CalButton
Responsibilities	Calculate 버튼을 클릭한다.
Type	GUI
Cross References	R2.1
Notes	TestCase를 계산해주는 Calculate()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Calculate()가 실행된다.



Name	ClearButton
Responsibilities	Clear 버튼을 클릭한다.
Type	GUI
Cross References	R3.1
Notes	초기화를 해주는 Clear()메서드를 호출한다.
Pre-Conditions	N/A
Post-Conditions	Clear()가 실행된다.

3. Activity 2055. Write Unit Test Code

```
14 • @Test
15 public void testInput_Manager() {
16     Input_Manager im = new Input_Manager();
17
18     assertNotNull(im.Category_List);
19     assertNotNull(im.Selected_RV_List);
20     assertNotNull(im.Property_List);
21
22 }
23
24
25 • @Test
26 public void testAdd_Category() {
27     Input_Manager im = new Input_Manager();
28
29     im.Add_Category();
30     im.Add_Category();
31     im.Add_Category();
32
33     assertEquals(3, im.Category_List.size());
34     assertEquals(3, im.Category_Num);
35 }
36
37 • @Test
38 public void testChange_Category() {
39     Input_Manager im = new Input_Manager();
40
41     im.Add_Category();
42     im.Change_Category(0, "1");
43
44     assertEquals("1", im.Search_Category(0).Category_Name);
45 }
46
47 • @Test
48 public void testDelete_Category() {
49     Input_Manager im = new Input_Manager();
50
51     im.Add_Category();
52     im.Add_Category();
53     im.Delete_Category(0);
54
55     assertEquals(1, im.Category_List.size());
56     assertEquals(2, im.Category_Num);
57     assertEquals(null, im.Search_Category(0));
58 }
59
```

```
60 @Test
61 public void testAdd_RV() {
62     Input_Manager im = new Input_Manager();
63
64     im.Add_Category();
65     im.Add_RV(0);
66     im.Add_RV(0);
67     im.Add_RV(0);
68
69     assertEquals(3, im.Category_List.get(0).RV_List.size());
70     assertEquals(3, im.Category_List.get(0).RV_Num);
71
72 }
73
74 @Test
75 public void testChange_RV() {
76     Input_Manager im = new Input_Manager();
77
78     im.Add_Category();
79     im.Add_RV(0);
80     im.Change_RV(0,0, "1");
81
82     assertEquals("1", im.Search_Category(0).Search_RV(0).RV_Name);
83 }
84
85 @Test
86 public void testDelete_RV() {
87     Input_Manager im = new Input_Manager();
88
89     im.Add_Category();
90     im.Add_RV(0);
91     im.Add_RV(0);
92     im.Add_RV(0);
93     im.Delete_RV(0,0);
94
95     assertEquals(2, im.Search_Category(0).RV_List.size());
96     assertEquals(3, im.Category_List.get(0).RV_Num);
97
98     assertEquals(null, im.Search_Category(0).Search_RV(0));
99 }
100
```

```
101 • @Test
102 public void testAdd_Constraint() {
103     Input_Manager im = new Input_Manager();
104
105     im.Add_Category();
106     im.Add_RV(0);
107     im.Add_Constraint(0, 0, "Single", "");
108     assertEquals(1, im.Search_Category(0).Search_RV(0).Single_Error);
109
110     im.Add_Constraint(0, 0, "Error", "");
111     assertEquals(2, im.Search_Category(0).Search_RV(0).Single_Error);
112     im.Add_Constraint(0, 0, "Property", "1");
113     im.Add_Constraint(0, 0, "Property", "1");
114     assertEquals(1, im.Search_Category(0).Search_RV(0).RV_Property_List.size());
115     im.Add_Constraint(0, 0, "If", "1");
116     im.Add_Constraint(0, 0, "If", "1");
117     im.Add_Constraint(0, 0, "If", "2");
118     assertEquals(1, im.Search_Category(0).Search_RV(0).If_List.size());
119 }
120
121 • @Test
122 public void testDelete_Constraint() {
123     Input_Manager im = new Input_Manager();
124
125     im.Add_Category();
126     im.Add_RV(0);
127     im.Add_Constraint(0, 0, "Single", "");
128     im.Add_Constraint(0, 0, "Property", "1");
129     im.Add_Constraint(0, 0, "If", "1");
130     im.Delete_Constraint(0,0, "Single", "");
131     im.Delete_Constraint(0,0, "Property", "1");
132     im.Delete_Constraint(0,0, "If", "1");
133     assertEquals(0, im.Search_Category(0).Search_RV(0).Single_Error);
134     assertEquals(0, im.Search_Category(0).Search_RV(0).RV_Property_List.size());
135     assertEquals(0, im.Search_Category(0).Search_RV(0).If_List.size());
136 }
137
```

```

150 • @Test
151 public void testCalculate() {
152     Input_Manager input_manager= new Input_Manager();
153
154     input_manager.Add_Category();
155     input_manager.Add_RV(0);
156     input_manager.Add_Constraint(0,0,"Error","");
157     input_manager.Add_RV(0);
158     input_manager.Add_Constraint(0,1,"Error","");
159     input_manager.Add_RV(0);
160     input_manager.Select_RV(0,0);
161
162     input_manager.Change_Category(0, "Model number");
163     input_manager.Change_RV(0, 0,"Malformed");
164     input_manager.Change_RV(0, 1,"Not in database");
165     input_manager.Change_RV(0, 2,"Valid");
166
167     input_manager.Add_Category();
168     input_manager.Add_RV(1);
169     input_manager.Add_Constraint(1,0,"Single","");
170     input_manager.Add_RV(1);
171     input_manager.Add_Constraint(1,1,"Single","");
172     input_manager.Add_Constraint(1,1,"Property", "RSNE");
173     input_manager.Add_RV(1);
174     input_manager.Add_Constraint(1,2,"Property", "RSNE");
175     input_manager.Add_Constraint(1,2,"Property", "RSMANY");
176
177     input_manager.Change_Category(1, "#SMRS");
178     input_manager.Change_RV(1, 0,"0");
179     input_manager.Change_RV(1, 1,"1");
180     input_manager.Change_RV(1, 2,"MANY");
181
182     input_manager.Add_Category();
183     input_manager.Add_RV(2);
184     input_manager.Add_Constraint(2,0,"Single","");
185     input_manager.Add_RV(2);
186     input_manager.Add_Constraint(2,1,"Single","");
187     input_manager.Add_Constraint(2,1,"Property", "OSNE");
188     input_manager.Add_RV(2);
189     input_manager.Add_Constraint(2,2,"Property", "OSNE");
190     input_manager.Add_Constraint(2,2,"Property", "OSMANY");
191
192     input_manager.Change_Category(2, "#SMOS");
193     input_manager.Change_RV(2, 0,"0");
194     input_manager.Change_RV(2, 1,"1");
195     input_manager.Change_RV(2, 2,"MANY");

```

```
196
197     input_manager.Add_Category();
198     input_manager.Add_RV(3);
199     input_manager.Add_Constraint(3,0,"Error","");
200     input_manager.Add_RV(3);
201     input_manager.Add_Constraint(3,1,"Single","");
202     input_manager.Add_RV(3);
203
204     input_manager.Change_Category(3, "#DBM");
205     input_manager.Change_RV(3, 0,"0");
206     input_manager.Change_RV(3, 1,"1");
207     input_manager.Change_RV(3, 2,"MANY");
208
209     input_manager.Add_Category();
210     input_manager.Add_RV(4);
211     input_manager.Add_Constraint(4,0,"Error","");
212     input_manager.Add_RV(4);
213     input_manager.Add_Constraint(4,1,"Single","");
214     input_manager.Add_RV(4);
215
216     input_manager.Change_Category(4, "#DBC");
217     input_manager.Change_RV(4, 0,"0");
218     input_manager.Change_RV(4, 1,"1");
219     input_manager.Change_RV(4, 2,"MANY");
220
221     input_manager.Add_Category();
222     input_manager.Add_RV(5);
223     input_manager.Add_Constraint(5,0,"Error","");
224     input_manager.Add_RV(5);
225     input_manager.Add_Constraint(5,1,"Error","");
226     input_manager.Add_RV(5);
227     input_manager.Add_Constraint(5,2,"Error","");
228     input_manager.Add_RV(5);
229
230     input_manager.Change_Category(5, "Correspondence of selection with model slots");
231     input_manager.Change_RV(5, 0,"Omitted slots");
232     input_manager.Change_RV(5, 1,"Extra slots");
233     input_manager.Change_RV(5, 2,"Mismatched slots ");
234     input_manager.Change_RV(5, 3,"Complete correspondence ");
235
236     input_manager.Add_Category();
237     input_manager.Add_RV(6);
238     input_manager.Add_Constraint(6,0,"Error","");
239     input_manager.Add_Constraint(6,0,"If","RSNE");
240     input_manager.Add_RV(6);
241     input_manager.Add_Constraint(6,1,"Error","");
```

```

242     input_manager.Add_Constraint(6,1,"If","RSNE");
243     input_manager.Add_RV(6);
244     input_manager.Add_Constraint(6,2,"If","RSMANY");
245
246     input_manager.Change_Category(6, "of required components");
247     input_manager.Change_RV(6, 0,"0");
248     input_manager.Change_RV(6, 1,"<RS");
249     input_manager.Change_RV(6, 2,"=RS");
250
251     input_manager.Add_Category();
252     input_manager.Add_RV(7);
253     input_manager.Add_Constraint(7,0,"Single","");
254     input_manager.Add_RV(7);
255     input_manager.Add_RV(7);
256     input_manager.Add_RV(7);
257     input_manager.Add_RV(7);
258     input_manager.Add_RV(7);
259     input_manager.Add_Constraint(7,5,"Error","");
260
261     input_manager.Change_Category(7, "Required component selection");
262     input_manager.Change_RV(7, 0,"Some defaults");
263     input_manager.Change_RV(7, 1,"All valid");
264     input_manager.Change_RV(7, 2,">=1 incompatible with slots");
265     input_manager.Change_RV(7, 3,">=1 incompatible with another selection");
266     input_manager.Change_RV(7, 4,">=1 incompatible with model");
267     input_manager.Change_RV(7, 5,">=1 not in database ");
268
269     input_manager.Add_Category();
270     input_manager.Add_RV(8);
271     input_manager.Add_RV(8);
272     input_manager.Add_Constraint(8,1,"If","OSNE");
273     input_manager.Add_RV(8);
274     input_manager.Add_Constraint(8,2,"If","OSMANY");
275
276     input_manager.Change_Category(8, "# of optional components");
277     input_manager.Change_RV(8, 0,"< #SMOS");
278     input_manager.Change_RV(8, 1,"= #SMOS");
279
280     input_manager.Add_Category();
281     input_manager.Add_RV(9);
282     input_manager.Add_Constraint(9,0,"Single","");
283     input_manager.Add_RV(9);
284     input_manager.Add_RV(9);
285     input_manager.Add_RV(9);
286     input_manager.Add_RV(9);
287     input_manager.Add_RV(9);

```

```

287     input_manager.Add_RV(9);
288     input_manager.Add_Constraint(9,5,"Error","");
289
290     input_manager.Change_Category(9, "Optional component selection");
291     input_manager.Change_RV(7, 0,"Some defaults");
292     input_manager.Change_RV(7, 1,"All valid");
293     input_manager.Change_RV(7, 2,">=1 incompatible with slots");
294     input_manager.Change_RV(7, 3,">=1 incompatible with another selection");
295     input_manager.Change_RV(7, 4,">=1 incompatible with model");
296     input_manager.Change_RV(7, 5,">=1 not in database ");
297
298     input_manager.Calculate();
299     assertEquals(input_manager.cal.All_TC.size(),67);
300     assertEquals(input_manager.cal.Selected_TC.size(),1);
301 }
302
303 @Test
304 public void testClear() {
305     Input_Manager im = new Input_Manager();
306
307     im.Add_Category();
308     im.Add_RV(0);
309     im.Add_RV(0);
310     im.Add_RV(0);
311
312     im.Clear();
313
314     assertEquals(im.Category_List.size(),0);
315     assertEquals(im.cal.All_TC.size(),0);
316     assertEquals(im.cal.Selected_TC.size(),0);
317
318 }
319
320

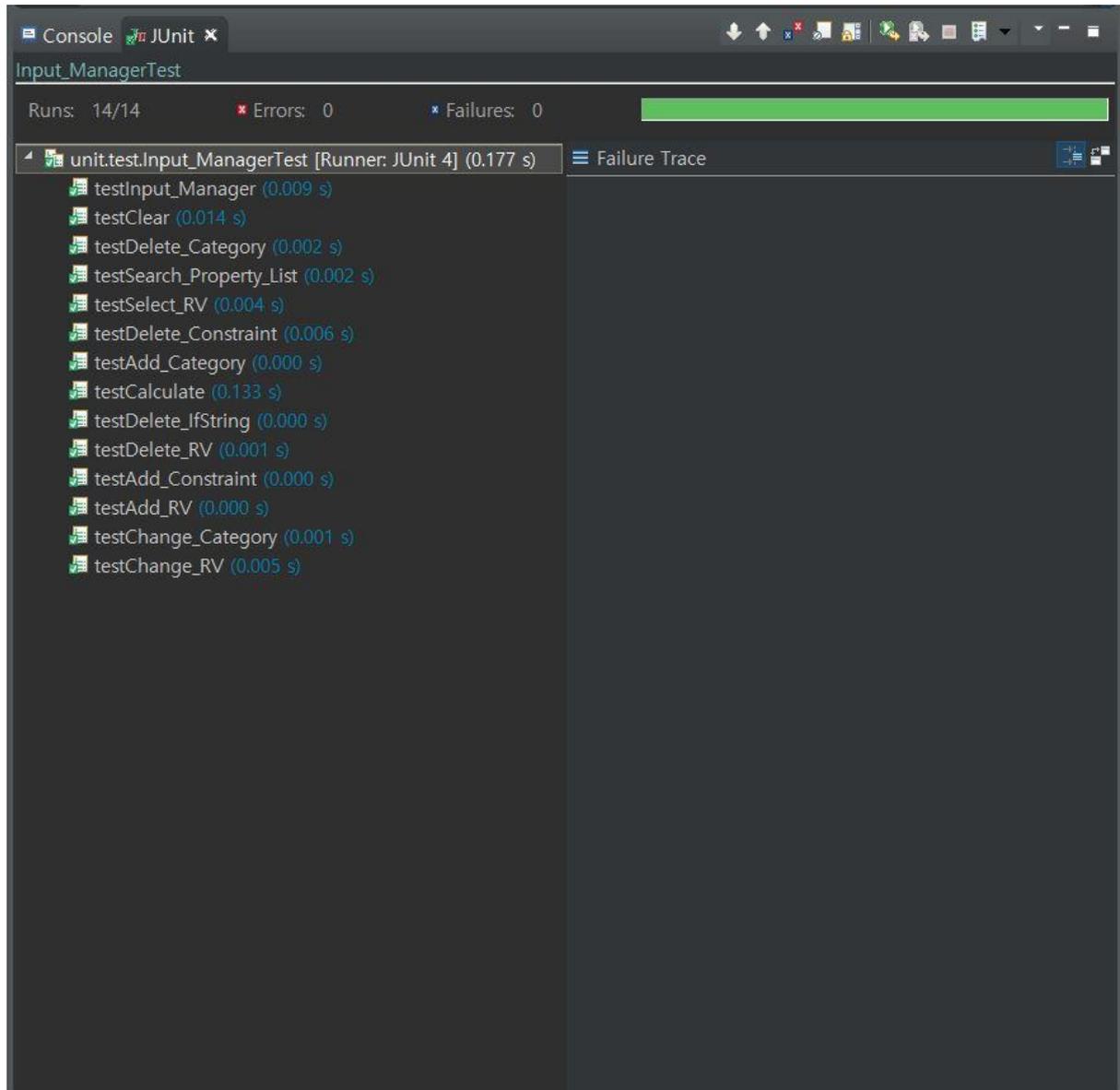
```

```

322 @Test
323 public void testDelete_IfString() {
324     Input_Manager im = new Input_Manager();
325     im.Add_Category();
326     im.Add_RV(0);
327     im.Add_RV(0);
328     im.Add_Constraint(0, 0, "Single", "");
329     im.Add_Constraint(0, 0, "Property", "1");
330     im.Add_Constraint(0, 0, "If", "1");
331     im.Add_Constraint(0, 1, "If", "1");
332     im.Delete_If("1");
333     assertEquals(im.Search_Category(0).Search_RV(0).If_List.size(),0);
334     assertEquals(im.Search_Category(0).Search_RV(1).If_List.size(),0);
335 }
336
337 @Test
338 public void testSearch_Property_List() {
339     Input_Manager im = new Input_Manager();
340     im.Add_Property_List("1");
341     assertEquals(true,im.Search_Property_List("1"));
342     assertEquals(false,im.Search_Property_List("2"));
343
344 }
345
346
347 }
348

```

4. Activity 2061. Unit Testing



5. Activity 2063. System Testing

Test Number	Test 항목	Description	Use Case	System Function	Pass/Fail
1-1	Category 추가 Test	Category를 추가 할 때 Category #number로 추가되는지 확인한다.	Category 추가	R1.1	P
2-1	Category 수정 Test	Category의 이름이 수정되는지 확인한다.	Category 수정	R1.2	P
2-2	Category 수정 Test	Category의 수정을 요청하고 이름을 입력하지 않았을 때 기존의 이름인지 확인한다.	Category 수정	R1.2	P

3-1	Category 삭제 Test	Category의 삭제를 요청하고 승인하면 Category가 삭제되는지 확인한다.	Category 삭제	R1.3	P
3-2	Category 삭제 Test	Category의 삭제를 요청하고 승인하지 않으면 Category가 유지되는지 확인한다.	Category 삭제	R1.3	P
3-3	Category 삭제 Test	Category내에 Property가 존재할때 삭제가 안되는지 확인한다.	Category 삭제	R1.3	P
4-1	대푯값 추가 Test	대푯값을 추가 할 때 RV #number로 추가되는지 확인한다.	대푯값 추가	R1.4	P
5-1	대푯값 수정 Test	대푯값의 이름이 수정되는지 확인한다.	대푯값 수정	R1.5	P
5-2	대푯값 수정 Test	대푯값의 수정을 요청하고 이름을 입력하지 않았을 때 기존의 이름인지 확인한다.	대푯값 수정	R1.5	P
6-1	대푯값 삭제 Test	대푯값의 삭제를 요청하고 승인하면 대푯값이 삭제되는지 확인한다.	대푯값 삭제	R1.6	P
6-2	대푯값 삭제 Test	대푯값의 삭제를 요청하고 승인하지 않으면 대푯값 유지되는지 확인한다.	대푯값 삭제	R1.6	P
6-3	대푯값 삭제 Test	대푯값을 삭제할때 Property가 존재하면 관련된 If를 삭제하는지 확인한다.	대푯값 삭제	R1.6	P
7-1	Constraint 추가 Test	Constraint의 이름과 속성을 입력하고 추가 할 때 입력된 이름과 속성대로 추가되는지 확인한다.	Constraint 추가	R1.7	P
7-2	Constraint 추가 Test	Constraint 추가를 위해 선택한 RV가 표시가 되는지 확인한다.	Constraint 추가	R1.7	P
7-3	Constraint 추가 Test	Property 속성을 갖는 Constraint가 존재하지 않을 때, If Property 속성을 갖는 Constraint 추가요청이 들어올 경우 Constraint를 추가하지 않는지 확인한다.	Constraint 추가	R1.7	P
8-1	Constraint 삭제 Test	Constraint의 삭제를 요청하고 승인하면 Constraint가 삭제되는지 확인한다.	Constraint 삭제	R1.8	P
8-2	Constraint 삭제 Test	Property를 삭제하면 관련된 If가 모두 삭제되는지 확인한다.	Constraint 삭제	R1.8	P
9-1	대푯값 선택 Test	대푯값을 선택하고 Test Case를 계산했을 때 화면에 선택한 대푯값의 Test Case가 출력되는지 확인한다.	대푯값 선택	R1.9	P
10-1	Test Case 계산 Test	Test Case를 예상 결과값과 비교한다.	Test Case 계산	R2.1	F
11-1	초기화 Test	초기화 요청 시 모든 입력과 출력이 삭제되는지 확인한다.	초기화	R3.1	P

6. Activity 2067. Testing Traceability Analysis

System Test Case	Operation in sequence diagram	Operation in interaction diagram	Method	Class	Unit Test
1-1 Category 추가 Test	1. Category 추가	→ Add_Category	→ Add_Category(void)	Input_Manager	→ testInput_Manager
2-1 Category 수정 Test	2. Category 수정	→ Change_Category	→ Change_Category(Category_IdInt, Category_NameString,void)		→ testAdd_Category
2-2 Category 삭제 Test	3. Category 삭제	→ Delete_Category	→ Delete_Category(Category_IdInt,void)		→ testChange_Category
3-1 Category 삭제 Test	4. 대륙값 추가	→ Add_RV	→ Add_RV(Category_IdInt,void)		→ testDelete_Category
3-2 Category 삭제 Test	5. 대륙값 수정	→ Change_RV	→ Change_RV(Category_IdInt,RV_IdInt,RV_NameString,void)		→ testAdd_RV
3-3 Category 삭제 Test	6. 대륙값 삭제	→ Delete_RV	→ Delete_RV(Category_IdInt,RV_IdInt,void)		→ testChange_RV
4-1 대륙값 추가 Test	7. Constraint 추가	→ Add_Constraint	→ Add_Constraint(Category_IdInt,RV_IdInt,Input_ConstraintString,Constraint_NameString,void)		→ testDelete_RV
5-1 대륙값 수정 Test	8. Constraint 삭제	→ Delete_Constraint	→ Delete_Constraint(Category_IdInt,RV_IdInt,Select_ConstraintString,Constraint_NameString,void)		→ testAdd_Constraint
5-2 대륙값 삭제 Test	9. 대륙값 선택	→ Select_RV	→ Select_RV(Category_IdInt,RV_IdInt,void)		→ testDelete_Constraint
6-1 대륙값 삭제 Test	10. Test Case 계산	→ Calculate	→ Calculate(void)		→ testSelect_RV
6-2 대륙값 삭제 Test	11. 초기화	→ Clear	→ Clear(void)		→ testCalculate
6-3 대륙값 삭제 Test			→ Remove_Category(Category_IdInt,void)	→ testClear	
7-1 Constraint 추가 Test			→ Search_Category(Category_IdInt,Category)	→ testDelete_HString	
7-2 Constraint 추가 Test			→ Delete_HProperty_List(List<String>)>void	→ testSearch_Property_List	
7-3 Constraint 추가 Test			→ Search_Property_List(Constraint_NameString,void)		
8-1 Constraint 삭제 Test			→ Add_Property_List(Constraint_NameString,void)		
8-2 Constraint 삭제 Test			→ Delete_HConstraint_NameString(void)		
9-1 대륙값 선택 Test			→ setName(Category_NameString,void)		
10-1 Test Case 계산 Test			→ Search_RV(RV_IdInt,RV)		
11-1 초기화 Test			→ hasProperty(Category_IdInt,Boolean)		
			→ Create_RV(void)	Category	
			→ Delete_HConstraint_NameString)	RV	
			→ setNameRV_nameString(void)		
			→ getPropertyValue(List<String>)		
			→ setSingle_ErrorInput_ConstraintString(void)		
			→ Add_Property(Constraint_NameString,void)		
			→ Add_HConstraint_NameString(void)		
			→ Delete_Property(Constraint_NameString,void)	Output_Manager	
			→ Output_Screen(Selected_TCList<TC>)>void	Calculate	
			→ Output_File(All_TCList<TC>)>void	TC	
			→ Output_Clear(void)		
			→ TC_Calculate(Category_List<Category>,Selected_ListList<Category>)>void		
			→ Select_Check(Selected_RVList<RV>)>Boolean		