

Software Modeling & Analysis

Team 3S

OOPT stage 2040

Design

Team No	Team5
과목	Software Modeling and Analysis
담당교수	JUNBEOM YOO Associate Professor / Ph.D
팀 구성원	201111383 전훈
	201311259 권오승
	201311292 유효상
	201511266 배윤희
제출일자	2017-05-10

Index

Activity2041. **Design Real Use Cases**

Activity2042. **Define Reports, UI, and Storyboards**

Activity2044. **Define Interaction Diagrams**

Activity2045. **Define Design Class Diagrams**

Activity2046. **Design Traceability Analysis**

Use case	Add Category
Actor	User
Purpose	Category 를 추가한다.
Overview	User 가 Category 를 추가한다.
Type	Primary and Essential
Cross Reference	System Function : R1.1, R1.4 Use-Case : “Display Category List”
Pre-Requisites	N/A
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 “+” 버튼을 누른다. (GUI#1-1). 2. (S) Category name 을 적을 공간을 보여준다. 3. (A) User 가 Category name 을 입력하고 enter 를 누른다. 4. (S) Category name 을 출력한다. (GUI#1-A)
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. (A) User 가 Category name 을 20 글자 초과 입력 시 20 글자만 출력한다.

Use case	Delete Category
Actor	User
Purpose	Category 를 삭제한다.
Overview	User 가 Category 를 삭제한다.
Type	Primary and Essential
Cross Reference	System Function : R1.2, R1.4 Use-Case : “Display Category List”, “Display Result”
Pre-Requisites	Category List 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 Delete 할 Category 를 클릭한다. (GUI#1-3) 2. (A) User 가 “-” 버튼을 누른다. (GUI#1-2) 3. (S) 선택한 Category 를 삭제한다. 4. (S) 반영된 Category List 를 출력한다. (GUI#1-A) 5. (S) 반영된 Test Case 갯수를 출력한다. (GUI#1-C)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	Edit Category
Actor	User
Purpose	Category name 을 수정한다.
Overview	User 가 Category name 을 수정한다.
Type	Primary and Essential
Cross Reference	System Function : R1.3, R1.4 Use-Case : “Display Category List”
Pre-Requisites	Category List 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 edit 할 Category 를 double click 한다. (GUI#1-3) 2. (S) Category name 이 Text field 로 바뀐다. 3. (A) User 가 원하는 name 을 입력 후 enter 를 누른다. 4. (S) 반영된 Category List 를 출력한다. (GUI#1-A)
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. (A) User 가 Category name 을 20 글자 초과 입력 시 20 글자만 출력한다.

Use case	Display Category List
Actor	Event-Based
Purpose	Add Category 를 통해 추가된 Category 를 Category List 에 출력한다.
Overview	Add Category 를 통해 추가된 Category 를 Category List 에 출력한다.
Type	Primary
Cross Reference	System Function : R1.4 Use-Case : -
Pre-Requisites	Category 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (S) 추가된 Category 를 받는다. 2. (S) Category List 에 출력한다. (GUI#1-A)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	Add Value
Actor	User
Purpose	Value 를 추가한다.
Overview	User 가 Value 를 추가한다.
Type	Primary and Essential
Cross Reference	System Function : R2.1, R2.4 Use-Case : “Display Value List”, “Display Result”
Pre-Requisites	Category List 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 “+” 버튼을 누른다. (GUI#1-5) 2. (S) Add Value name 을 적을 공간을 보여준다. (GUI#1-4) 3. (A) User 가 Value name 을 입력하고 enter 를 누른다. 4. (S) Value name 을 출력한다. (GUI#1-B) 5. (S) 반영된 Test Case 개수를 출력한다. (GUI#1-C)
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. (A) User 가 Value name 을 20 글자 초과 입력 시 20 글자만 출력한다.

Use case	Delete Value
Actor	User
Purpose	Value 를 삭제한다.
Overview	User 가 Value 를 삭제한다.
Type	Primary and Essential
Cross Reference	System Function : R2.2, R2.4 Use-Case : “Display Value List”, “Display Result”
Pre-Requisites	Value List 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 Delete 할 Value 를 클릭한다. (GUI#1-4) 2. (A) User 가 “+” 버튼을 누른다. (GUI#1-5) 3. (S) 선택한 Value 를 삭제한다. 4. (S) 반영된 Value List 를 출력한다. (GUI#1-B) 5. (S) 반영된 Test Case 개수를 출력한다. (GUI#1-C)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	Edit Value
Actor	User
Purpose	Value name 을 수정한다.
Overview	User 가 Value name 을 수정한다.
Type	Primary and Essential
Cross Reference	System Function : R2.3, R2.4 Use-Case : “Display Value List”
Pre-Requisites	Category List 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 edit 할 Value 를 double click 한다. (GUI#1-4) 2. (S) Value name 이 Text field 로 바뀐다. 3. (A) User 가 원하는 name 을 입력 후 enter 를 누른다. 4. (S) 반영된 Value List 를 출력한다. (GUI#1-B)
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. (A) User 가 Value name 을 20 글자 초과 입력 시 20 글자만 출력한다.

Use case	Display Value List
Actor	Event-Based
Purpose	Add Value 를 통해 추가된 Value 를 Value List 에 출력한다.
Overview	Add Value 를 통해 추가된 Value 를 Value List 에 출력한다.
Type	Primary
Cross Reference	System Function : R2.4, R4.2 Use-Case : “Display Result”
Pre-Requisites	Value 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (S) 추가된 Value 를 받는다. 2. (S) Value List 에 출력한다. (GUI#1-B)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	Check Single
Actor	User
Purpose	Single check box 를 선택한다.
Overview	User 가 Single check box 를 선택한다.
Type	Primary and Essential
Cross Reference	System Function : R2.5, R3.1, R4.2 Use-Case : “Display Value State”, “Display Result”
Pre-Requisites	Value List 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 Single check box 를 클릭한다. (GUI#1-7) 2. (S) Single check box 를 check 표시로 출력한다. (GUI#1-B) 3. (S) 반영된 Test Case 개수를 출력한다. (GUI#1-C)
Alternative Courses of Events	Check Error 를 선택한다.
Exceptional Courses of Events	N/A

Use case	Check Error
Actor	User
Purpose	Error check box 를 선택한다.
Overview	User 가 Error check box 를 선택한다.
Type	Primary and Essential
Cross Reference	System Function : R2.5, R3.2, R4.2 Use-Case : “Display Value State”, “Display Result”
Pre-Requisites	Value List 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 Error check box 를 클릭한다. (GUI#1-8) 2. (S) Error check box 를 check 표시로 출력한다. (GUI#1-B) 3. (S) 반영된 Test Case 개수를 출력한다. (GUI#1-C)
Alternative Courses of Events	Check Single 을 선택한다.
Exceptional Courses of Events	N/A

Use case	Add Property
Actor	User
Purpose	Value 에 Property 를 추가한다.
Overview	User 가 Value 에 Property 를 추가한다.
Type	Primary and Essential
Cross Reference	System Function : R2.5, R3.3, R4.2 Use-Case : “Display Value State”, “Display Result”
Pre-Requisites	1. Value List 가 하나 이상 있어야 한다. 2. 추가할 Property 의 Value 가 선택되어 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 Property 에 “+” 버튼을 누른다. (GUI#1-9) 2. (A) User 가 Pop-Up 된 창에 Property name 을 입력하고 Enter 를 누른다. 3. (S) Property name 을 출력한다. (GUI#1-B)
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. (A) User 가 Property name 을 20 글자 초과 입력 시 20 글자만 출력한다.

Use case	Add if-Property
Actor	User
Purpose	Value 에 if-Property 를 추가한다.
Overview	User 가 Value 에 Property 를 추가한다.
Type	Primary and Essential
Cross Reference	System Function : R2.5, R3.4, R4.2 Use-Case : “Display Value State”, “Display Result”
Pre-Requisites	Value List 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (A) User 가 IF Property 에 “+” 버튼을 누른다. (GUI#1-9) 2. (A) User 가 Pop-Up 된 창에 IF Property name 을 입력하고 Enter 를 누른다. 3. (S) If Property name 을 출력한다. (GUI#1-B)
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. (A) User 가 if-Property name 을 20 글자 초과 입력 시 20 글자만 출력한다.

Use case	Display Value State
Actor	Event-Based
Purpose	Check Single, Check Error, Add Property, Add If-Property 를 통해 추가된 State 를 Value State 에 출력한다.
Overview	Check Single, Check Error, Add Property, Add If-Property 를 통해 추가된 State 를 Value State 에 출력한다.
Type	Primary
Cross Reference	System Function : R3.4 Use-Case : -
Pre-Requisites	State 가 하나 이상 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (S) 추가된 State 를 받는다. 2. (S) Value State 에 출력한다. (GUI#1-B)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use case	Calculate
Actor	Event-Based
Purpose	Category 에 대한 Test cases 의 개수를 출력한다.
Overview	System 이 Category 에 대한 Test cases 의 개수를 계산한다.
Type	Primary and Essential
Cross Reference	System Function : R4.2 Use-Case : -
Pre-Requisites	최소한 하나 이상의 Category List 와 그에 대한 Value List 에 입력된 값이 존재해야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (S) Calculate 을 한다. 2. (S) Calculate 결과를 출력한다. (GUI#1-C)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

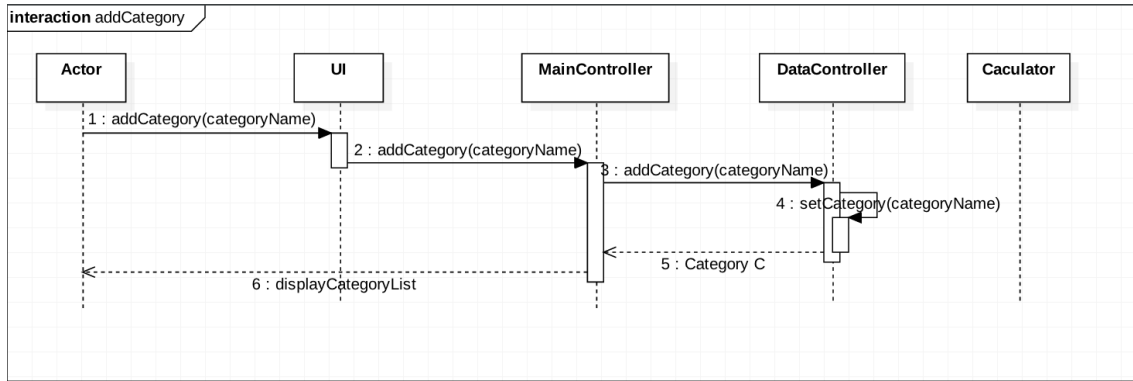
Use case	Display Result
Actor	Event-Based
Purpose	Calculate 을 통해 계산된 Test Case 의 개수를 Result 창에 출력한다.
Overview	Calculate 을 통해 계산된 Test Case 의 개수를 Result 창에 출력한다.
Type	Primary
Cross Reference	N/A
Pre-Requisites	Calculate 할 Category 가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (S) 추가된 Category 의 Test Case 를 계산한다. 2. (S) Result 창에 Test Case 개수를 출력한다. (GUI#1-C)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Activity 2042. Define Reports, UI, and Storyboards

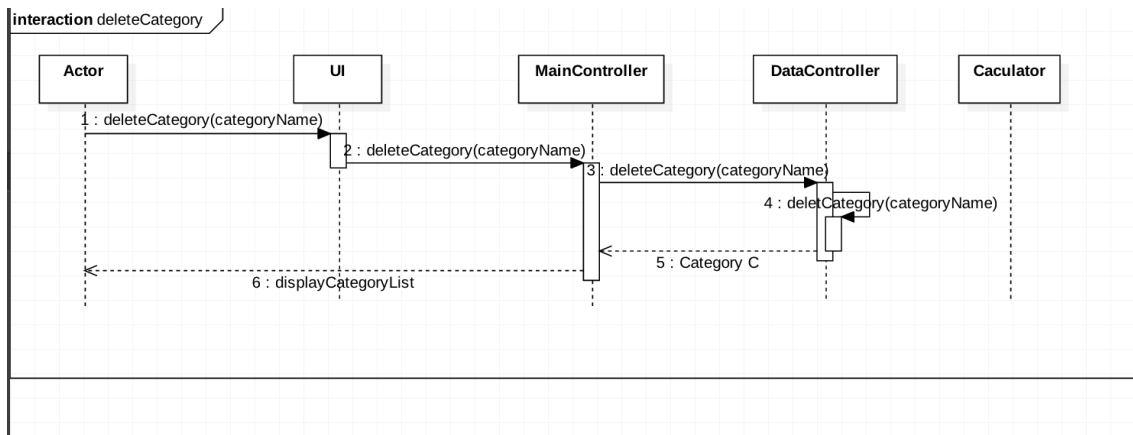
The image shows a UI design wireframe for a test case management application. It is divided into three main sections:

- Left Sidebar (A):** A dark blue sidebar containing a list of categories from "CATEGORY 1" to "CATEGORY 13". At the bottom of the sidebar is a red circle labeled "A" and a "+ -" toggle button with red circles "1" and "2" below it.
- Main Content Area (B):** A white area with a light green border. It contains:
 - A "Value 1" input field with a red circle "4" above it.
 - Three "Insert your value..." input fields.
 - A table with columns: "Single", "Error", "Property", and "if Property".
 - The "Single" column has three dropdown menus with red circles "7" and "8" next to them.
 - The "Error" column has three checkboxes.
 - The "Property" column has a dropdown menu with "property" selected and three "list" items below it, each with a "+" button. A red circle "9" is next to the first dropdown.
 - The "if Property" column has a dropdown menu with "property" selected and three input fields with "+" and "-" buttons. A red circle "10" is next to the first dropdown.
 - A red circle "B" is centered in the main content area.
 - At the bottom right of the main content area are two buttons: a "+" button with a red circle "5" and a "-" button with a red circle "6".
- Bottom Summary Bar (C):** A white bar with a light green border. It contains a red circle "C" and the text "24 TEST CASES".

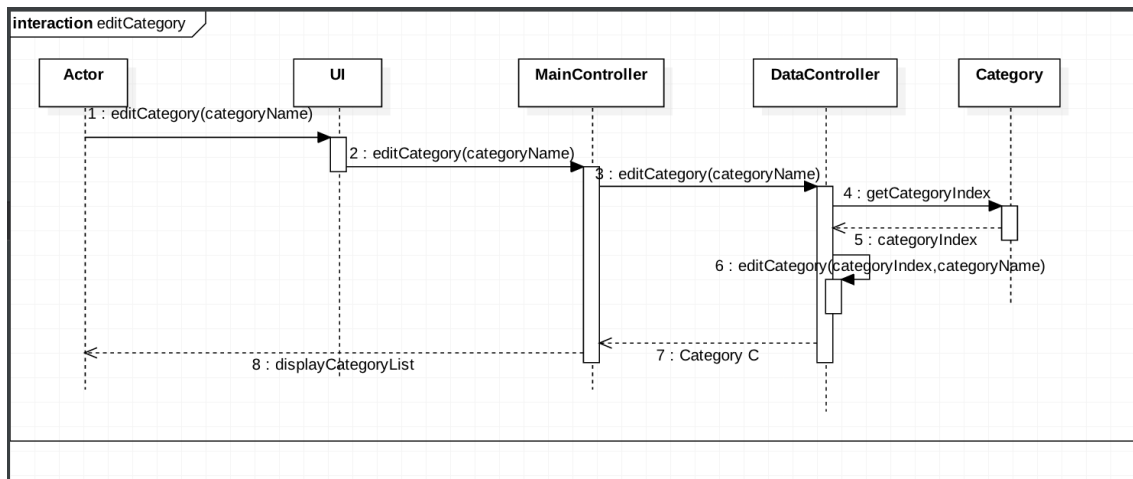
Activity 2044. Define Interaction Diagrams



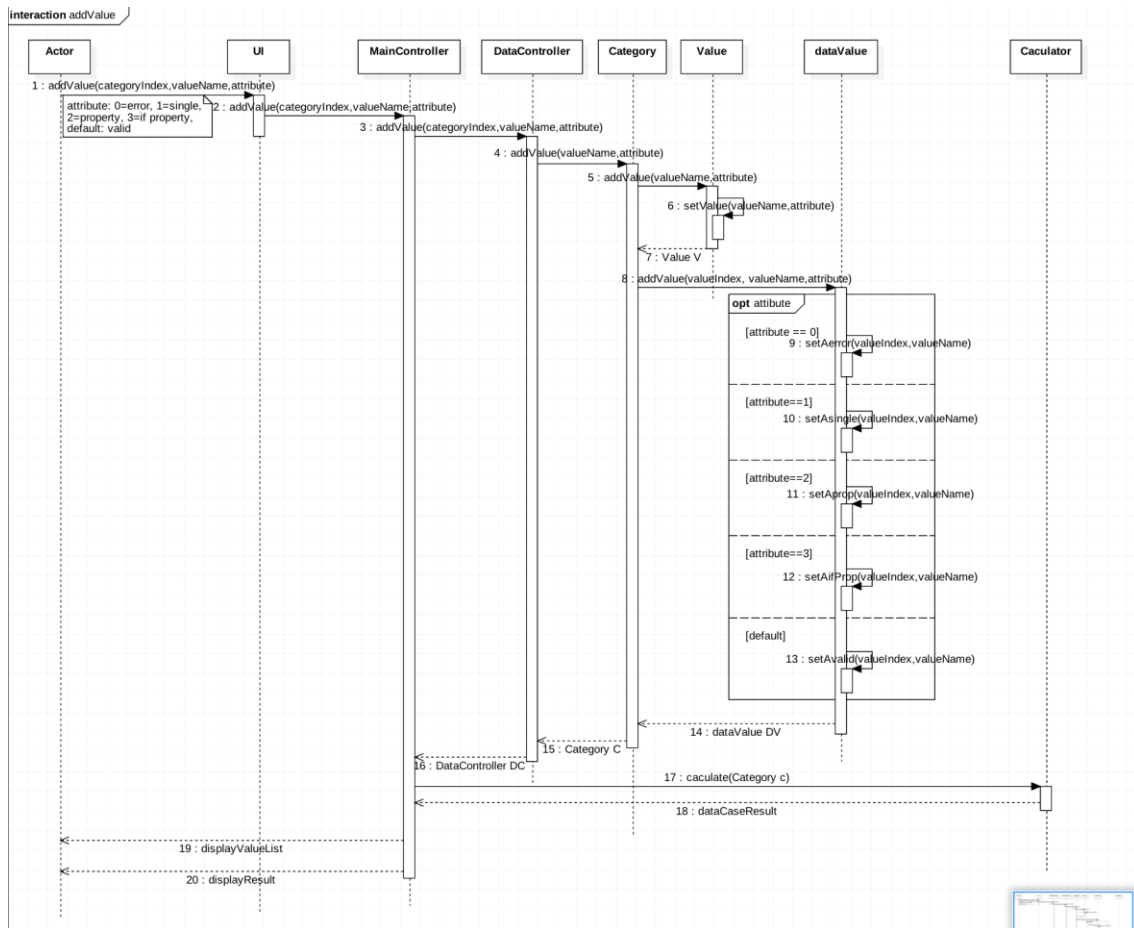
-add category



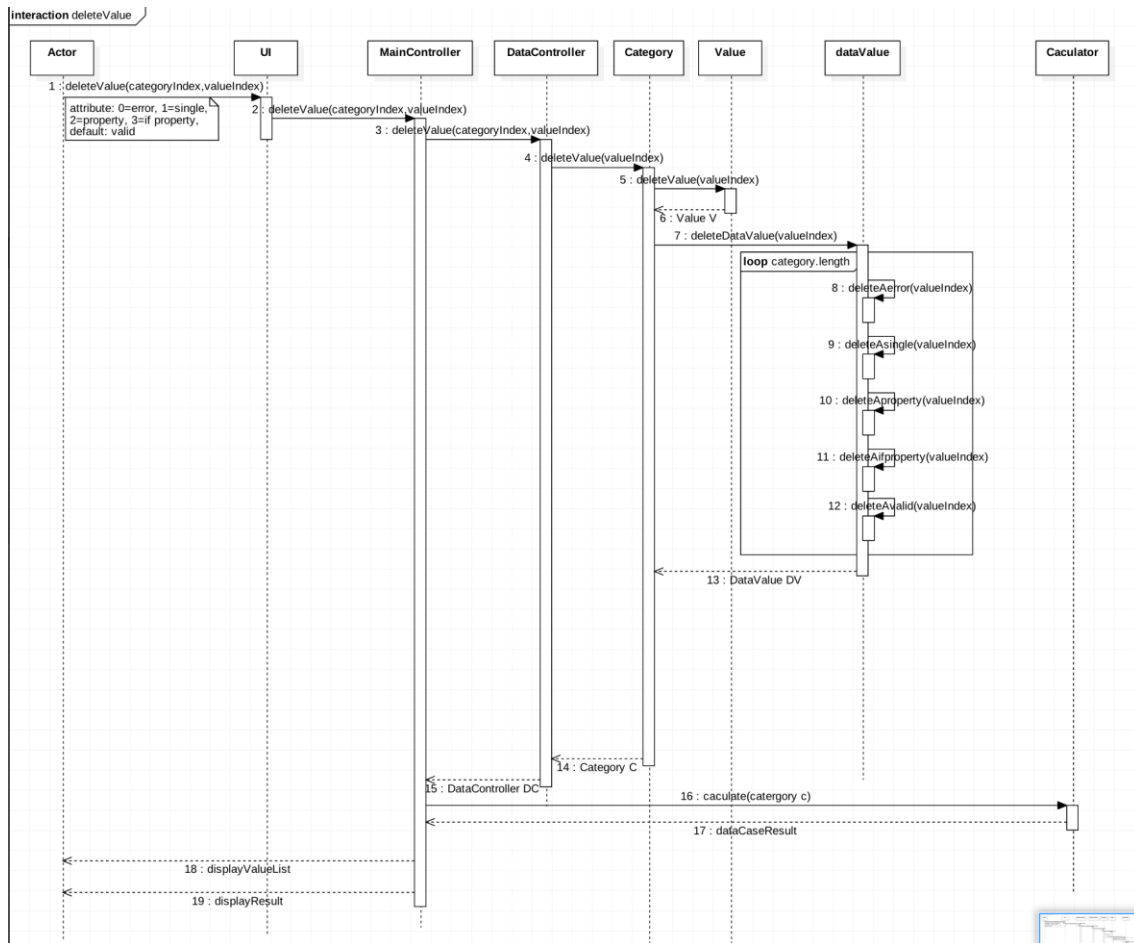
-delete category



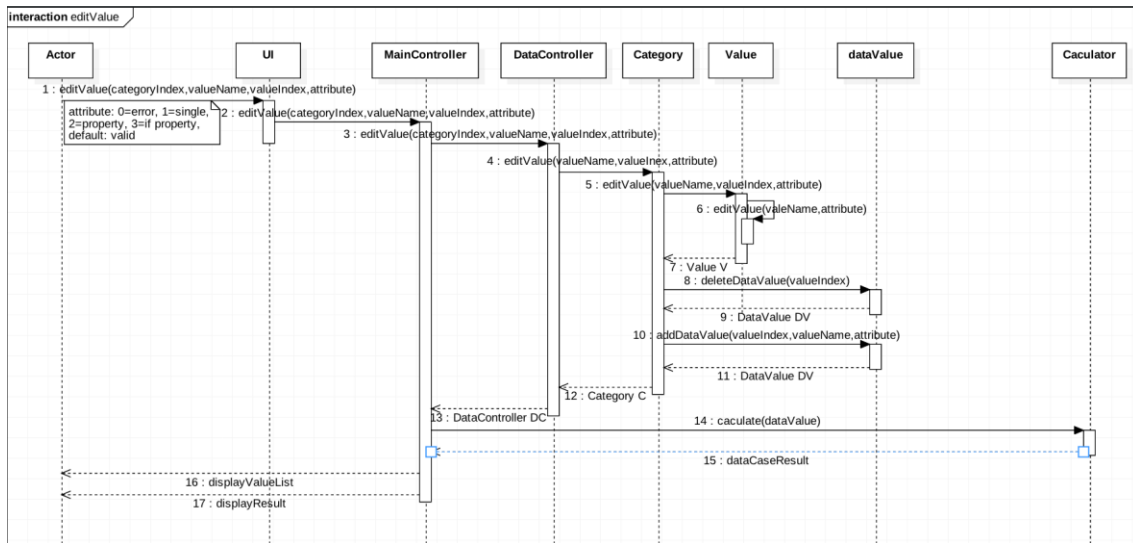
-edit category



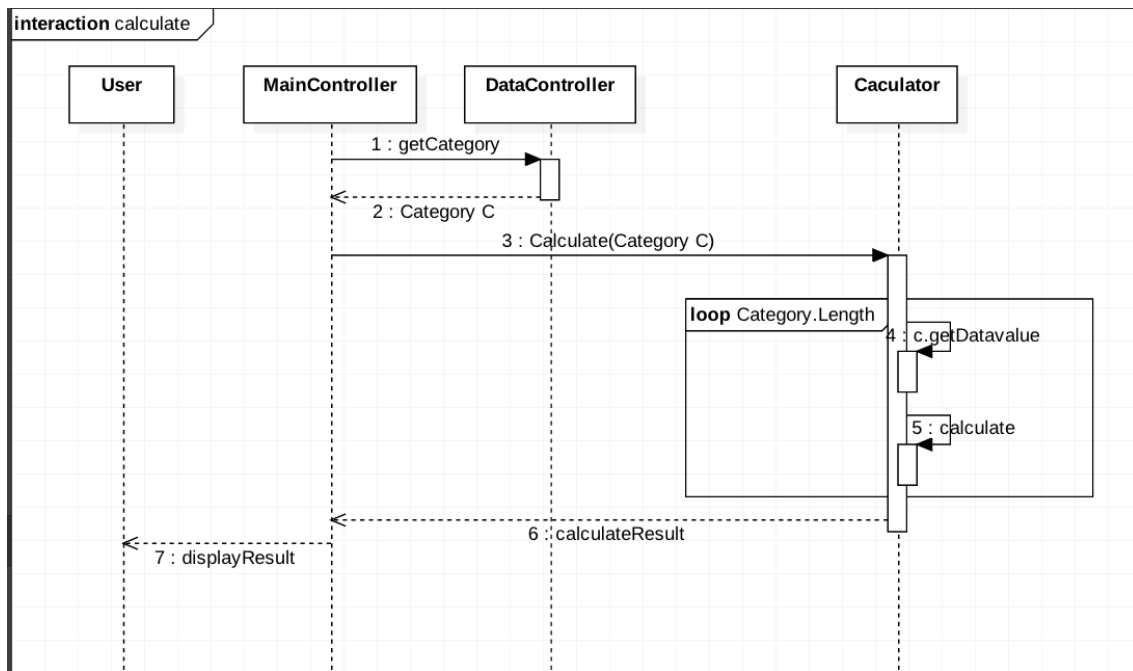
-add value



-delete value

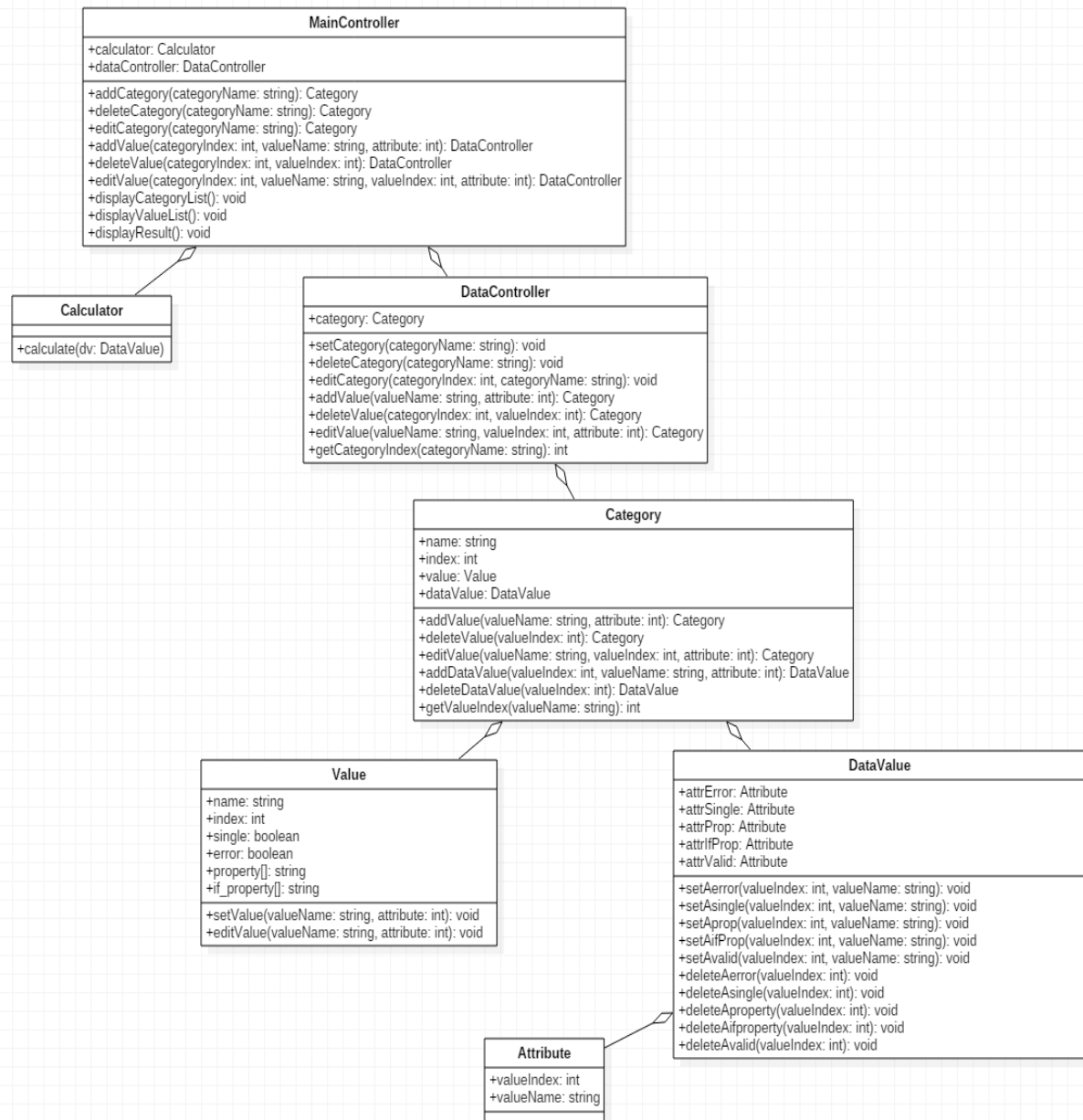


-edit value



-calculate

Activity 2045. Define Design Class Diagrams



Activity 2046. Design Traceability Analysis

Operation in sequence diagram	Operation in interaction diagram	Method	class
1. addCategory	addCategory(categoryName)	addCategory(categoryName: String): Category	MainController
2. deleteCategory	addCategory(categoryName)	deleteCategory(categoryName: String): Category	
3. editCategory	addCategory(categoryName)	editCategory(categoryName: String): Category	
4. addValue	setCategory(categoryName)	addValue(categoryIndex: int, valueName: String, attribute: int): DataController	
5. deleteValue	displayCategoryList()	deleteValue(categoryIndex: int, valueIndex: int): DataController	
6. editValue	deleteCategory(categoryName)	editValue(categoryIndex: int, valueName: String, valueIndex: int, attribute: int): DataController	
7. calculate	deleteCategory(categoryName)	displayCategoryList(): void	
	deleteCategory(categoryName)	displayValueList(): void	DataController
	deleteCategory(categoryName)	displayResult(): void	
	editCategory(categoryName)	setCategory(categoryName: String): void	
	editCategory(categoryName)	deleteCategory(categoryName: String): void	
	editCategory(categoryName)	editCategory(categoryIndex: int, categoryName: string): void	
	editCategory(categoryName)	addValue(valueName: String, attribute: int): Category	
	getCategoryIndex()	deleteValue(categoryIndex: int, valueIndex: int): Category	
	addValue(categoryIndex, valueName, attribute)	editValue(valueName: string, valueIndex: int, attribute: int): Category	
	addValue(categoryIndex, valueName, attribute)	getCategoryIndex(categoryName: String): int	
	addValue(valueName, attribute)	addValue(valueName: String, attribute: int): Category	
	addValue(valueName, attribute)	deleteValue(categoryIndex: int, valueIndex: int): Category	Category
	setValue(valueName, attribute)	editValue(valueName: String, valueIndex: int, attribute: int): Category	
	addValue(valueName, attribute)	addDataValue(valueIndex: int, valueName: String, attribute: int): DataValue	
	displayValueList()	deleteDataValue(valueIndex: int): DataValue	
	displayResult()	getValueIndex(valueName: String): int	
	deleteValue(categoryIndex, valueIndex)	setAerror(valueIndex: int, valueName: String): void	
	deleteValue(categoryIndex, valueIndex)	setAsingle(valueIndex: int, valueName: String): void	
	deleteValue(categoryIndex, valueIndex)	setAproperty(valueIndex: int, valueName: String): void	
	deleteValue(valueIndex)	setAifProperty(valueIndex: int, valueName: String): void	
	deleteValue(valueIndex)	getAvalid(valueIndex: int, valueName: String): void	
	deleteDataValue(valueIndex)	deleteAerror(valueIndex: int): void	DataValue
	dataCaseResult()	deleteAsingle(valueIndex: int): void	
	editValue(categoryIndex, valueName, valueIndex, attribute)	deleteAproperty(valueIndex: int): void	
	editValue(categoryIndex, valueName, valueIndex, attribute)	deleteAifProperty(valueIndex: int): void	
	editValue(categoryIndex, valueName, valueIndex, attribute)	deleteAvalid(valueIndex: int): void	
	editValue(valueName, valueIndex, attribute)	setValue(valueName: String, attribute: int): void	
	editValue(valueName, valueIndex, attribute)	editValue(valueName: String, attribute:int): void	
	editValue(valueName, attribute)	calculate(dv: DataValue): void	
	calculate(dataValue)		
	getCategory()		
	calculateResult()		Calculator