“Priority” Category Partitioning Testing Tool

OOPT stage 1000

Plan & Elaboration

Project Team

T4

201311265 김상원

201210194 김정환

201311269 김제헌

201311297 이상명

Date

2017-03-29

Index

Activity 1001. Define Draft Plan

Activity 1002. Create Preliminary Investigation Report

Activity 1003. Define Requirements

Activity 1004. Record Terms in Glossary

Activity 1005. Implement Prototype

Activity 1006. Define Draft System Architecture

Activity 1007. Define Business Use Case

Activity 1008. Define Business Concept Model

Activity 1009. Define System Test Case

Activity 1010. Refine Plan

Activity 1001. Define Draft Plan

1. Motivation

소프트웨어 개발 과정 중, 테스트 단계는 매우 중요한 단계이다. 결과물이 요구사항에 부합하는지 확인하기 위해 많은 테스트 케이스를 작성하여 테스트를 진행한다. 테스트 케이스를 도출하기 위한 여러 방법론 중에 Category-Partition Method가 있다. Category-Partition 방법론은 테스터가 카테고리를 나누고 constraint를 정하는 등 테스트 명세를 만들고 그 명세서에 따라 테스트 케이스들을 조합하여 도출해내는 방식이다. 앞부분은 테스터가 직접 해야하지만, 테스트 케이스를 생성해내는 부분은 자동화 시킬 수 있다. 이번 프로젝트에서는 테스트 단계의 소모 시간을 줄일 수 있도록 테스트 명세를 입력 받아 자동으로 테스트 케이스를 생성해주는 도구를 만들고자 한다.

1. Project Objectives

테스트 명세를 입력 받아, 자동으로 테스트 케이스를 생성하고 그 결과를 파일 형태로 저장해주는 Category Partitioning Test Case Generating Tool

* category, choice, constraint, priority를 입력 받아 테스트 케이스를 생성하고 이를 보여준다.
* 테스트 케이스들을 우선순위대로 정렬하여 보여준다.
* 생성된 테스트 케이스들을 엑셀 파일 형태로 export한다.

1. Project Scope

“Priority”는 테스트 단계에서, 모든 테스트 케이스가 동일한 중요도를 갖지 않는다는 사실에 착안하였다. 더 중요한 테스트 케이스를 덜 중요한 케이스 보다 시간상 먼저 테스팅 함으로써 테스터는 중요한 시스템 오류를 짧은 시간 안에 파악하여 대처할 수 있다. 이는 소프트웨어의 규모가 클수록 더욱 필요한 기능일 것이다. 따라서 단순히 테스트 케이스를 생성시키는 것뿐만 아니라 케이스들 간에 우선순위를 매기는 것을 목표로 한다.

1. Functional Requirements

- Make new specification file

- Load specification file

- Open specification file by double clicking

- Shut down program

- Set specification file name

- Add category

- Delete category

- Add input value

- Delete input value

- Add property

- Delete property

- Add if property

- Delete if property

- Add single and error constraints

- Delete single and error constraints

- Set priority

- Show all

- Generate test cases

- Save contemporary specification file

- Save and go back to first page

- Go previous

- Go next

- Export test case to excel file

1. Non-Functional Requirements

* 테스트 케이스들 간의 우선순위를 순서대로 보여준다.

1. Resource Estimation
   1. Human efforts (M/M)

4Man/3Month

* 1. Human resources

4명

* 1. Duration

12주

* 1. Budget

200,000원

1. Other Information

N/A

Activity 1002. Create Preliminary Investigation Report

1. Alternative Solutions

* 도구를 사용하지 않고 손으로 직접 테스트 케이스를 작성한다.
* 시중에서 상용화된 Test Case Generating Tool을 사용한다.
* 외주로 개발한다.

1. Project Justification (Business Demands)
   1. Cost

시중의 제품보다 저렴하다.

* 1. Duration

외주보다 기간을 단축시킬 수 있다.

* 1. Risk

직접 손으로 케이스를 조합하면 실수가 발생하여 케이스를 누락할 수도 있다.

* 1. Effect

원하는 기능(priority)을 추가하여 개발할 수 있다.

1. Risk Management

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Probability | Significance | Weight |
| OOPT 경험 부족 | 5 | 5 | 25 |
| JAVA 사용 경험 부족 | 3 | 4 | 12 |
| Category Partition Method에 대한 이해 부족 | 4 | 1 | 4 |
| 작업 공간 부족 | 3 | 3 | 9 |

1. Risk Reduction Plan

|  |  |
| --- | --- |
| Risk | Reduction Plan |
| OOPT 경험 부족 | 교수님, 조교님께 자문을 구한다 |
| JAVA 사용 경험 부족 | JAVA 및 GUI 공부 |
| Category Partition Method에 대한 이해 부족 | Category Partition Method 공부 |
| 작업 공간 부족 | 실습실 적극 이용 |

1. Market Analysis

* 소규모 프로젝트의 경우 MS엑셀 프로그램 활용만으로도 테스트케이스를 쉽게 조합, 도출할 수 있다.
* 테스트 케이스 생성기가 시중에 상용화되어 있다.

1. Managerial Issue

* 정해진 개발 일정에 맞추어 단계별 개발을 진행한다. (<http://dslab.konkuk.ac.kr/Class/2017/17SMA/Team_Organization.htm> 참조)

Activity 1003. Define Requirements

1. Functional Requirements

|  |  |
| --- | --- |
| **Function** | **Description** |
| NewSpecification | Make new specification file. |
| LoadSpecification | Load specification file. |
| RecentSpecification | Open specification file by double clicking. |
| Quit | Shut down program. |
| SetFileName | Set specification file name. |
| AddCategory | Add category. |
| DeleteCategory | Delete category. |
| AddInputValue | Add input value. |
| DeleteInputValue | Delete input value. |
| AddProperty | Add property rank. |
| DeleteProperty | Delete property. |
| AddIfProperty | Add if property. |
| DeleteIfProperty | Delete if property. |
| AddErrorSingle | Add single and error constraints. |
| DeleteErrorSingle | Delete single and error constraints. |
| SetPriority | Set priority. |
| ShowAllSpecification | Show all specification information. |
| GoPrevStep | Go previous step page. |
| GoNextStep | Go next step page. |
| GoFinish | Go to complete page. |
| ExtractTestCase | Extract a Category Partitioning Test Case from a given Specification. |
| SaveTestCase | Save the extracted Category Partitioning Test Case. |
| MakeTestCaseFile | Generate the extracted Category Partitioning Test Case as an Excel file. |
| GoInitialPage | Go back to first page. |
| SaveOngoing | Save ongoing specification file |
| FileIO | Load/Save specification file. |
| CheckSameName | Check existance of the same name at each step |

1. System Functions

|  |  |  |
| --- | --- | --- |
| **Reference No.** | **Function** | **Category** |
| R.1.1 | NewSpecification | Evident |
| R.1.2 | LoadSpecification | Evident |
| R.1.3 | RecentSpecification | Evident |
| R.1.4 | Quit | Evident |
| R.2.1 | SetFileName | Evident |
| R.3.1 | AddCategory | Evident |
| R.3.2 | DeleteCategory | Evident |
| R.4.1 | AddInputValue | Evident |
| R.4.2 | DeleteInputValue | Evident |
| R.5.1 | AddProperty | Evident |
| R.5.2 | DeleteProperty | Evident |
| R 6.1 | AddIfProperty | Evident |
| R 6.2 | DeleteIfProperty | Evident |
| R.7.1 | AddErrorSingle | Evident |
| R.7.2 | DeleteErrorSingle | Evident |
| R.8 | SetPriority | Evident |
| R.9 | ShowAllSpecification | Evident |
| R.10.1 | GoPrevStep | Evident |
| R.10.2 | GoNextStep | Evident |
| R.10.3 | GoFinish | Evident |
| R.11.1 | ExtractTestCase | Hidden |
| R.11.2 | SaveTestCase | Evident |
| R.12.1 | MakeTestCaseFile | Hidden |
| R.12.2 | GoInitialPage | Evident |
| R.13 | SaveOngoing | Evident |
| R.14 | FileIO | Hidden |
| R.15 | CheckSameName | Hidden |

1. Performance Requirements

* Category Partitioning Test Case 생성시간은 5초 이내로 수행되어야 한다.

1. Operation Environment

* OS : 7 Microsoft Windows 7 이상

1. Development Environment

A. 운영체제 : Windows 7, 10

B. CPU : Intel

C. IDE : Eclipse, NetBeans

D. 개발 언어 : JAVA

E. UML 툴 : StarUML

1. Interface Requirements

* Tool을 시작했을 때, 최근 Specification 파일들이 보여야 한다.
* Category와 Value를 입력 받을 때, 하나 이상의 Category와 Value 값을 입력받아야만, 다음 Step으로 넘어갈 수 있어야 한다.
* Property 값을 설정한 이후 Step에서는 Category와 Value값을 수정할 수 없고, 각 Step과 관련된 값들만 수정할 수 있어야 한다.
* ShowAllSpecification에서는 모든 값들을 수정할 수 없고 볼 수만 있어야 한다.
* MakeTestCaseFile를 통해 생성된 엑셀파일은 Test Case들이 Priority에 따라 보기 좋게 표현되어야 한다.
* FileLoad를 통해 불러 읽은 경우, 도중에 만들던 Step에서 시작해야 한다.

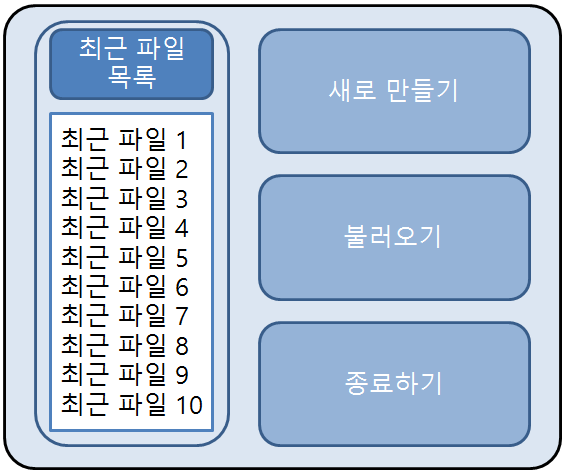
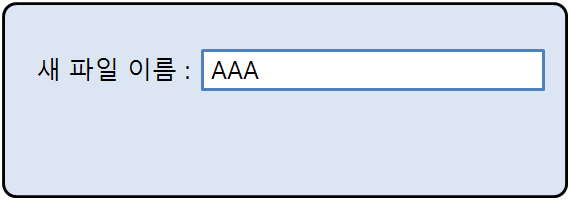
1. Other Requirements

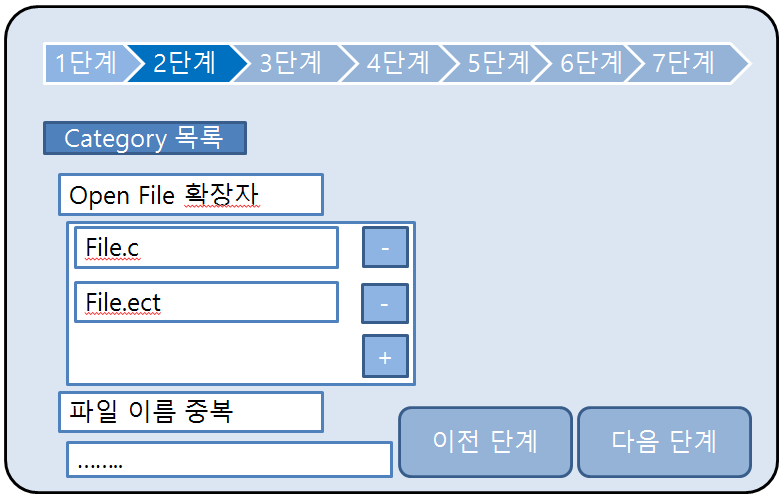
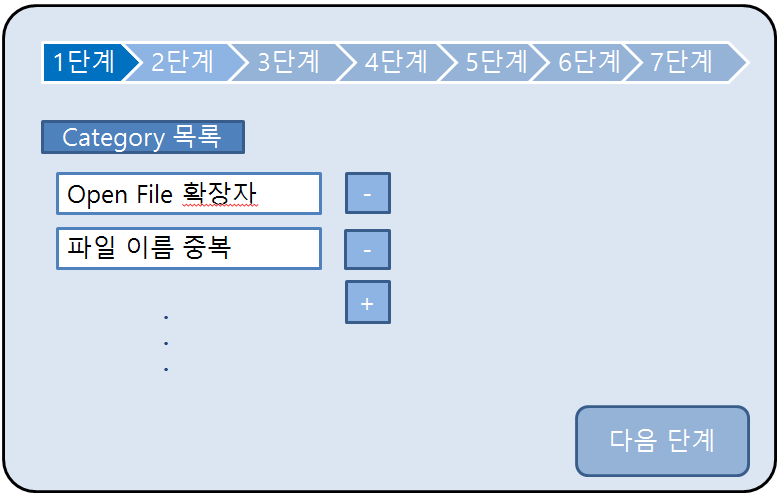
* N/A

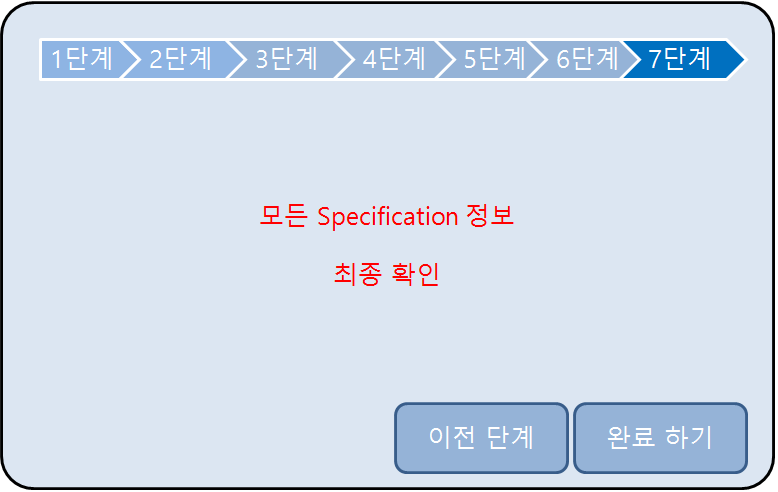
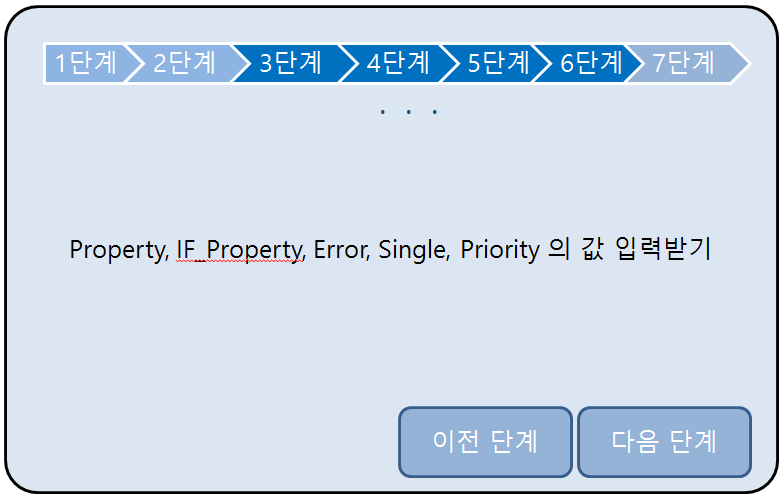
Activity 1004. Record Terms in Glossary

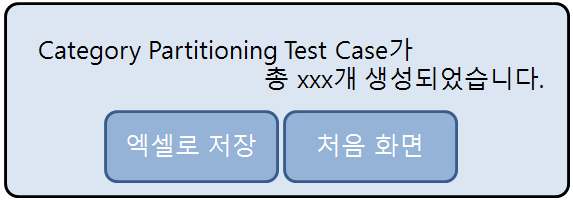
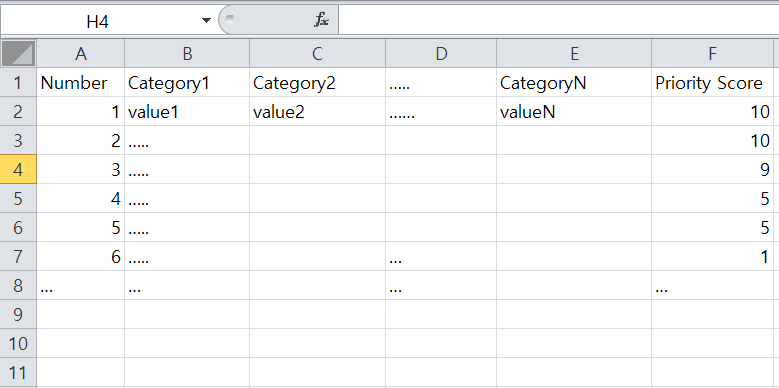
|  |  |  |
| --- | --- | --- |
| **Term** | **Description** | **Remarks** |
| Specification | Category Partitioning Test Case에 관련된 Category Value, Property, Error, Single 등의 정보들을 담고 있는 파일 |  |
| Priority | 사용자가 판단한 Test Case 각각의 우선순위[중요도] |  |
| Step | Category Partitioning Test Case에 필요한 정보들을 입력받는 순차적인 단계 |  |

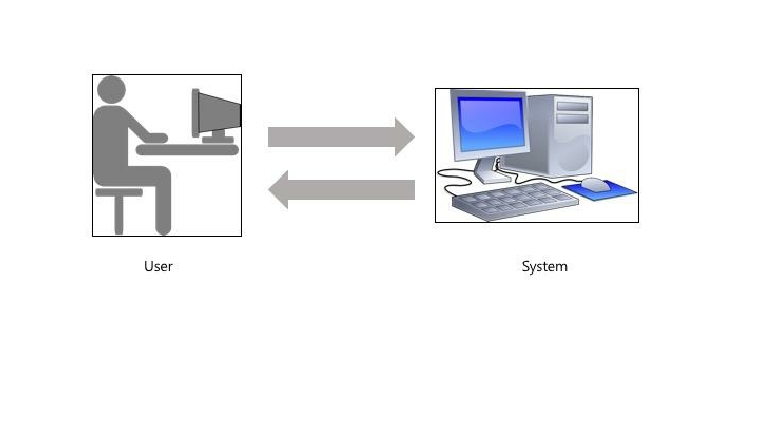
Activity 1005. Implement Prototype



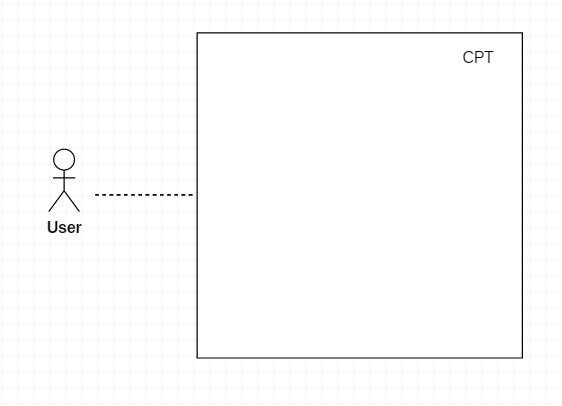


Activity 1006. Define Draft System Architecture

Activity 1007. Define Business Use Case

1. Define System Boundary

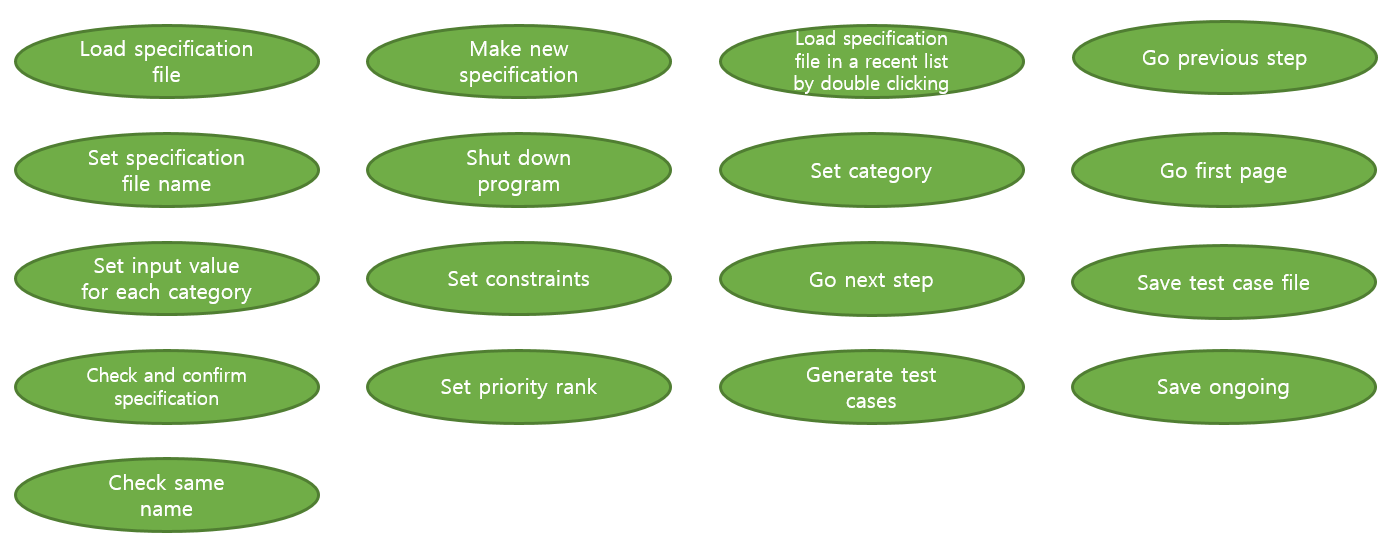


1. Identify and Describe Actors

* User: CPT test case를 만들기 위해 시스템과 상호작용하는 actor

1. Identify Use Cases

3.1 Actor-Based

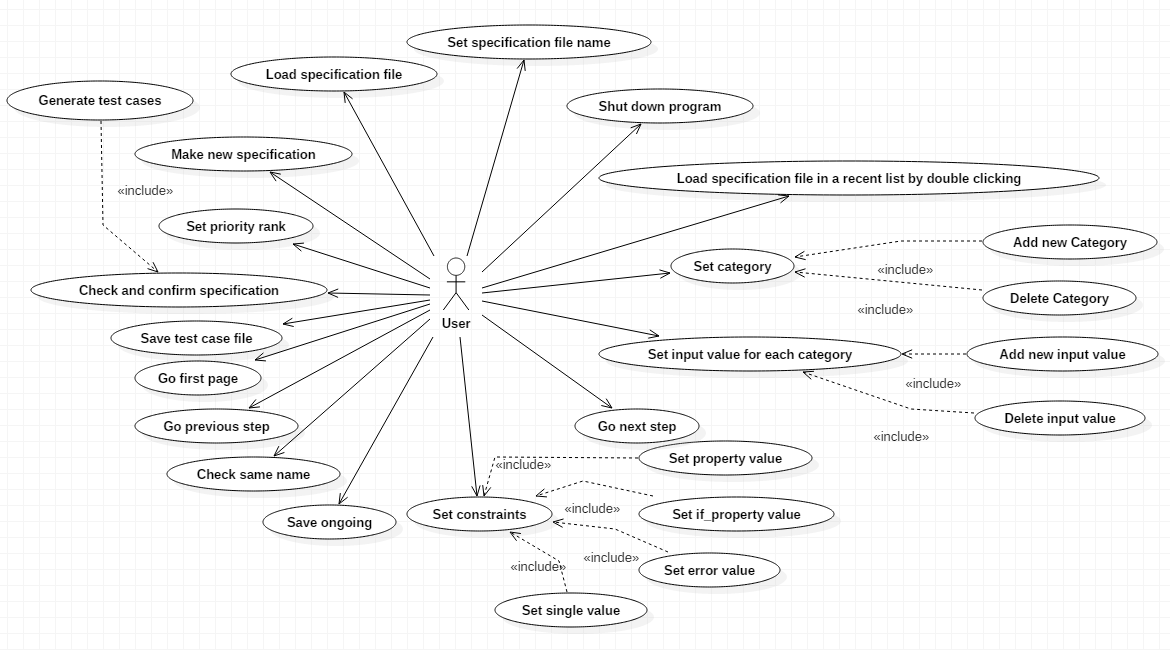


3.2 Event-Based

1. Allocate System Functions into related Use Cases and Categorize Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference No.** | **Function** | **Use case Number** | **Category** |
| R.1.1 | NewSpecification | 2 | Primary |
| R.1.2 | LoadSpecification | 1 | Primary |
| R.1.3 | RecentSpecification | 3 | Primary |
| R.1.4 | Quit | 6 | Primary |
| R.2.1 | SetFileName | 5 | Primary |
| R.3.1 | AddCategory | 7 | Primary |
| R.3.2 | DeleteCategory | 7 | Primary |
| R.4.1 | AddInputValue | 9 | Primary |
| R.4.2 | DeleteInputValue | 9 | Primary |
| R.5.1 | AddProperty | 10 | Primary |
| R.5.2 | DeleteProperty | 10 | Primary |
| R 6.1 | AddIfProperty | 10 | Primary |
| R 6.2 | DeleteIfProperty | 10 | Primary |
| R.7.1 | AddErrorSingle | 10 | Primary |
| R.7.2 | DeleteErrorSingle | 10 | Primary |
| R.8 | SetPriority | 14 | Primary |
| R.9 | ShowAllSpecification | 13 | Primary |
| R.10.1 | GoPrevStep | 4 | Primary |
| R.10.2 | GoNextStep | 11 | Primary |
| R.10.3 | GoFinish | 13 | Primary |
| R.11.1 | ExtractTestCase | 15 | Primary |
| R.11.2 | SaveTestCase | 12 | Primary |
| R.12.1 | MakeTestCaseFile | 13 | Primary |
| R.12.2 | GoInitialPage | 8 | Primary |
| R.13 | SaveOngoing | 16 | Primary |
| R.14 | FileIO | 15 | Primary |
| R.15 | CheckSameName | 17 | Primary |

1. Draw a Use Case Diagram



1. Describe Use Cases

Use case Number - 1

|  |  |
| --- | --- |
| Use case | Load specification file |
| Actors | Users |
| Description | * 이미 작성했던 specification중 저장한 파일을 불러온다. * 초기화면에서 버튼을 클릭하는 use case이다. * 버튼 클릭 후 파일을 찾기 위한 창으로 넘어간다. |

Use case Number - 2

|  |  |
| --- | --- |
| Use case | Make new specification |
| Actors | Users |
| Description | * Test case generate을 위해 새로운 specification을 작성한다. * 초기화면에서 버튼을 클릭하는 use case이다. * 버튼 클릭 후 저장할 파일 이름을 입력하는 화면으로 넘어간다. |

Use case Number – 3

|  |  |
| --- | --- |
| Use case | Load specification in recent list by double clicking |
| Actors | Users |
| Description | * 초기화면에 있는 recent file list에서 한 항목을 더블 클릭한다. * 더블 클릭 후 파일을 찾기 위한 창으로 넘어간다. |

Use case Number - 4

|  |  |
| --- | --- |
| Use case | Go previous step |
| Actors | Users |
| Description | * Specification을 입력 받는 각각 단계에서 그 전 단계로 넘어간다. |

Use case Number - 5

|  |  |
| --- | --- |
| Use case | Set specification file name |
| Actors | Users |
| Description | * Specification을 입력하기 전 서로 다른 specification들을 식별하기 위한 이름을 지정한다. * 지정된 이름은 저장할 파일과 recent file list에 사용된다. |

Use case Number - 6

|  |  |
| --- | --- |
| Use case | Shut down program |
| Actors | Users |
| Description | * 프로그램을 종료한다. * 초기화면에서 버튼을 클릭하는 use case이다. |

Use case Number - 7

|  |  |
| --- | --- |
| Use case | Set category |
| Actors | Users |
| Description | * Specification의 category들을 입력한다. * 각 category의 이름을 정할 수 있다. * 이미 입력된 category를 지울 수 있다. |

Use case Number - 8

|  |  |
| --- | --- |
| Use case | Go first page |
| Actors | Users |
| Description | * Test case generate이 끝난 후 버튼을 누르는 use case이다, * 버튼을 클릭하면 초기화면으로 넘어간다. |

Use case Number - 9

|  |  |
| --- | --- |
| Use case | Set input value for each category |
| Actors | Users |
| Description | * 작성한 각 category마다 input value값을 넣어준다. * 이미 작성된 input value를 지울 수 있다. |

Use case Number - 10

|  |  |
| --- | --- |
| Use case | Set constraints |
| Actors | Users |
| Description | * 입력한 각 input value마다 constraints를 줄 수 있다. * Property, if\_property, single, error를 줄 수 있다. * 각각의 constraints는 단계별로 부여하며 default값을 가진다. |

Use case Number - 11

|  |  |
| --- | --- |
| Use case | Go next step |
| Actors | Users |
| Description | * Specification을 입력하는 각 단계에서 버튼을 누르는 use case이다. * 버튼을 누르면 다음 단계로 넘어가며 입력한 정보를 저장한다. |

Use case Number - 12

|  |  |
| --- | --- |
| Use case | Save test case file |
| Actors | Users |
| Description | * Test case generate이 끝난 후 생성된 test cases를 저장한다. * 저장은 지정된 경로에 엑셀파일의 형태로 이루어진다. |

Use case Number - 13

|  |  |
| --- | --- |
| Use case | Check and confirm specification |
| Actors | Users |
| Description | * 입력한 전체 specification을 볼 수 있다. * 보여지는 specification을 checking하고 문제가 있으면 입력단계로 돌아갈 수 있다. * Confirm을 하는 경우 test case generate이 수행된다. |

Use case Number - 14

|  |  |
| --- | --- |
| Use case | Set priority rank |
| Actors | Users |
| Description | * 입력된 input value들에게 사용자가 priority(중요도)를 부여한다. * 입력할 수 있는 priority(중요도)는 총 5단계로 정해진다. |

Use case Number - 15

|  |  |
| --- | --- |
| Use case | Generate test case |
| Actors | Users |
| Description | * Specification이 confirm되면 test case generate이 시작된다. * 입력된 specification을 통해 test case를 만들고 그 총 개수를 화면을 통해 사용자에게 알려준다. * 사용자는 생성된 test cases의 수를 보고 generate이 잘 되었는지 판단할 수 있다. |

Use case Number - 16

|  |  |
| --- | --- |
| Use case | Check same name |
| Actors | Users |
| Description | * 저장할 specification의 파일이름이나 입력할 category, input value들의 이름들이 중복되지 않는지 검사한다. |

Use case Number – 17

|  |  |
| --- | --- |
| Use case | Save ongoing |
| Actors | Users |
| Description | * specification입력 중 작업을 중단하고 다시 시작할 수 있도록 현재까지 진행한 입력 사항을 저장할 수 있다. * 저장한 후 나중에 초기화면의 recent list에서 불러올 수 있다. |

1. Rank Use Cases

|  |  |  |
| --- | --- | --- |
| **Rank** | **Use case Number & Names** | **Category** |
| High | 2. Make new specification | Primary |
| High | 1. Load specification file | Primary |
| High | 3. Load specification in recent list by double clicking | Primary |
| High | 6. Shut down program | Primary |
| High | 5. Set specification file name | Primary |
| High | 7. Set category | Primary |
| High | 7. Set category | Primary |
| High | 9. Set input value for each category | Primary |
| High | 9. Set input value for each category | Primary |
| High | 10. Set constraints | Primary |
| High | 10. Set constraints | Primary |
| High | 10. Set constraints | Primary |
| High | 10. Set constraints | Primary |
| High | 10. Set constraints | Primary |
| High | 10. Set constraints | Primary |
| High | 14. Set priority rank | Primary |
| High | 13. Check and confirm specification | Primary |
| High | 4. Go previous step | Primary |
| High | 11. Go next step | Primary |
| High | 13. Check and confirm specification | Primary |
| High | 15. Generate test case | Primary |
| High | 12. Save test case file | Primary |
| High | 13. Check and confirm specification | Primary |
| High | 8. Go first page | Primary |
| High | 15. Generate test case | Primary |

Activity 1008. Define Business Concept Model

최근 사용 명세 보존/재사용

기존 명세 불러오기

새로운 명세 입력

중요도 설정

테스트 케이스 외부 파일 출력

테스트 케이스 생성

Category, Value, Constraints 입력

Activity 1009. Define System Test Case

|  |  |  |
| --- | --- | --- |
| **Ref No.** | **Function** | **Test Case** |
| R.1.1 | NewSpecification | - New 버튼을 눌렀을 때, 생성될 파일의 경로를 입력하는 대화상자가 제대로 출력되는지 확인한다. |
| R.1.2 | LoadSpecification | - Load 버튼을 눌렀을 때, 파일 탐색기가 제대로 출력되는지 확인한다.  - 명세 파일 선택 시 제대로 명세 파일이 프로그램으로 로드 되는지 확인한다.  - 명세 파일이 아닌 파일을 선택 시 경고 창과 함께 에러 핸들링을 했는지 확인한다. |
| R.1.3 | RecentSpecification | - 최근 사용했던 명세 파일 목록이 리스트에 잘 표시되는지 확인한다.  - 리스트에서 항목 하나를 더블클릭 했을 때, 작업했던 환경을 그대로 불러올 수 있는지 확인한다. |
| R.1.4 | Quit | - quit 버튼을 눌렀을 때 정상적으로 종료되는지 확인한다. |
| R.2.1 | SetFileName | - 파일명을 지정해 줬을 때 해당 경로로 파일이 제대로 생성되는지 확인한다. |
| R.3.1 | AddCategory | - 버튼을 눌렀을 때, 카테고리 입력폼이 새로 생성되는지 확인한다. |
| R.3.2 | DeleteCategory | - 버튼을 눌렀을 때, 해당 카테고리가 삭제되는지 확인한다.  - value, constraints를 다 설정해준 뒤 눌렀을 경우, 하위 항목(value, constraints)들까지 제대로 삭제되는지 확인한다. |
| R.4.1 | AddInputValue | - 버튼을 눌렀을 때, Input Value를 입력할 수 있는 폼이 생성되는지 확인한다. |
| R.4.2 | DeleteInputValue | - 버튼을 눌렀을 때, 해당 입력 값이 삭제되는지 확인한다.  - constraints를 설정해준 뒤 눌렀을 경우, 하위 항목(constraints)이 제대로 삭제되는지 확인한다. |
| R.5.1 | AddProperty | - 버튼을 눌렀을 때 입력폼이 생성되는지 확인한다. |
| R.5.2 | DeleteProperty | - 버튼을 눌렀을 때, property가 제대로 삭제되는지 확인한다.  - 삭제할 때, 연관된 if constraints를 삭제해 주는지 확인한다. |
| R 6.1 | AddIfProperty | - 버튼을 누르면, 이전 단계에서 생성해낸 property들이 콤보박스에 제대로 출력되는지 확인한다. |
| R 6.2 | DeleteIfProperty | - 버튼을 눌렀을 때 if constraints가 정상적으로 삭제되는지 확인한다. |
| R.7.1 | AddErrorSingle | - 각 value 별 체크박스가 정상작동하는지 확인한다. |
| R.7.2 | DeleteErrorSingle | - 각 value 별 체크박스가 정상작동하는지 확인한다. |
| R.8 | SetPriority | - 각 value 별 중요도를 제대로 부여할 수 있는지 확인한다.(라디오 버튼이 잘 작동하는지 확인) |
| R.9 | ShowAllSpecification | - 이전 단계에서 적용한 모든 명세가 제대로 모두 표시되는지 확인한다. |
| R.10.1 | GoPrevStep | - 이전 단계로 이동하는 버튼이 정상 작동하는지 확인한다.  - 전 단계로 이동할 때, 명세 보존이 잘 되는지 확인한다. |
| R.10.2 | GoNextStep | - 다음 단계로 이동하는 버튼이 정상 작동하는지 확인한다.  - 단계에서 필요한 사항을 입력하지 않았을 때 넘어가지 못하게 제어가 잘 되는지 확인한다. |
| R.10.3 | GoFinish | - 종료용 대화상자가 잘 출력되는지 확인한다. |
| R.11.1 | ExtractTestCase | - Category Partitioning 알고리즘에 의해 정상적으로 테스트 케이스가 구해지는지 확인한다. |
| R.11.2 | SaveTestCase | - 저장 버튼이 정상적으로 구동하는지 확인한다. |
| R.12.1 | MakeTestCaseFile | - 테스트 케이스를 엑셀 파일로 정상적으로 저장하는지, 명세 파일도 저장해내는지 확인한다. |
| R.12.2 | GoInitialPage | - 버튼을 눌렀을 때, 내용을 초기화 시키고 첫 화면으로 이동 되는지 확인한다. |
| R.13 | FileIO | - 전 과정에서 파일 입출력 및 파일 스트림 관리가 잘 이루어지는지 확인한다. |

Activity 1010. Refine Plan

1. Project Scope

Category-Partition Method에 따라 작성된 테스트 명세를 입력 받아 테스트 케이스를 생성해주는 도구를 개발한다.

1. Project Objectives

여러 테스트 케이스들을 생성하고 케이스들 사이의 우선순위를 매겨 테스터에게 제공한다.

1. Functional Requirements

|  |  |
| --- | --- |
| **Function** | **Description** |
| NewSpecification | Make new specification file. |
| LoadSpecification | Load specification file. |
| RecentSpecification | Open specification file by double clicking. |
| Quit | Shut down program. |
| SetFileName | Set specification file name. |
| AddCategory | Add category. |
| DeleteCategory | Delete category. |
| AddInputValue | Add input value. |
| DeleteInputValue | Delete input value. |
| AddProperty | Add property. |
| DeleteProperty | Delete property. |
| AddIfProperty | Add if property. |
| DeleteIfProperty | Delete if property. |
| AddErrorSingle | Add single and error constraints. |
| DeleteErrorSingle | Delete single and error constraints. |
| SetPriority | Set priority. |
| ShowAllSpecification | Show all specification information. |
| GoPrevStep | Go previous step page. |
| GoNextStep | Go next step page. |
| GoFinish | Go to complete page. |
| ExtractTestCase | Extract a Category Partitioning Test Case from a given Specification. |
| SaveTestCase | Save the extracted Category Partitioning Test Case. |
| MakeTestCaseFile | Generate the extracted Category Partitioning Test Case as an Excel file. |
| GoInitialPage | Go back to first page. |
| FileIO | Load/Save specification file. |

1. Performance Requirements

* Category Partitioning Test Case 생성시간은 5초 이내로 수행되어야 한다.

1. Operating Environment

* OS : 7 Microsoft Windows 7 이상

1. User Interface Requirements

* Tool을 시작했을 때, 최근 Specification 파일들이 최대 10개까지 보여야 한다.
* Category와 Value를 입력 받을 때, 하나 이상의 Category와 Value 값을 입력받아야만 다음 Step으로 넘어갈 수 있어야 한다.
* Property 값을 설정한 이후 Step에서는 Category와 Value값을 수정할 수 없고, 각 Step과 관련된 값들만 수정할 수 있어야 한다.
* ShowAllSpecification에서는 모든 값들을 수정할 수 없고 볼 수만 있어야 한다.
* MakeTestCaseFile를 통해 생성된 엑셀파일은 Test Case들이 Priority에 따라 보기 좋게 표현되어야 한다.
* FileLoad를 통해 불러 읽은 경우, 도중에 만들던 Step에서 시작해야 한다.

1. Other Requirements

* N/A

1. Resources
   1. Human Efforts (M/M)

4M/3M

* 1. Human Resources

프로그래머 4명

* 1. Duration

3개월

* 1. Budget

200,000원

1. Scheduling

