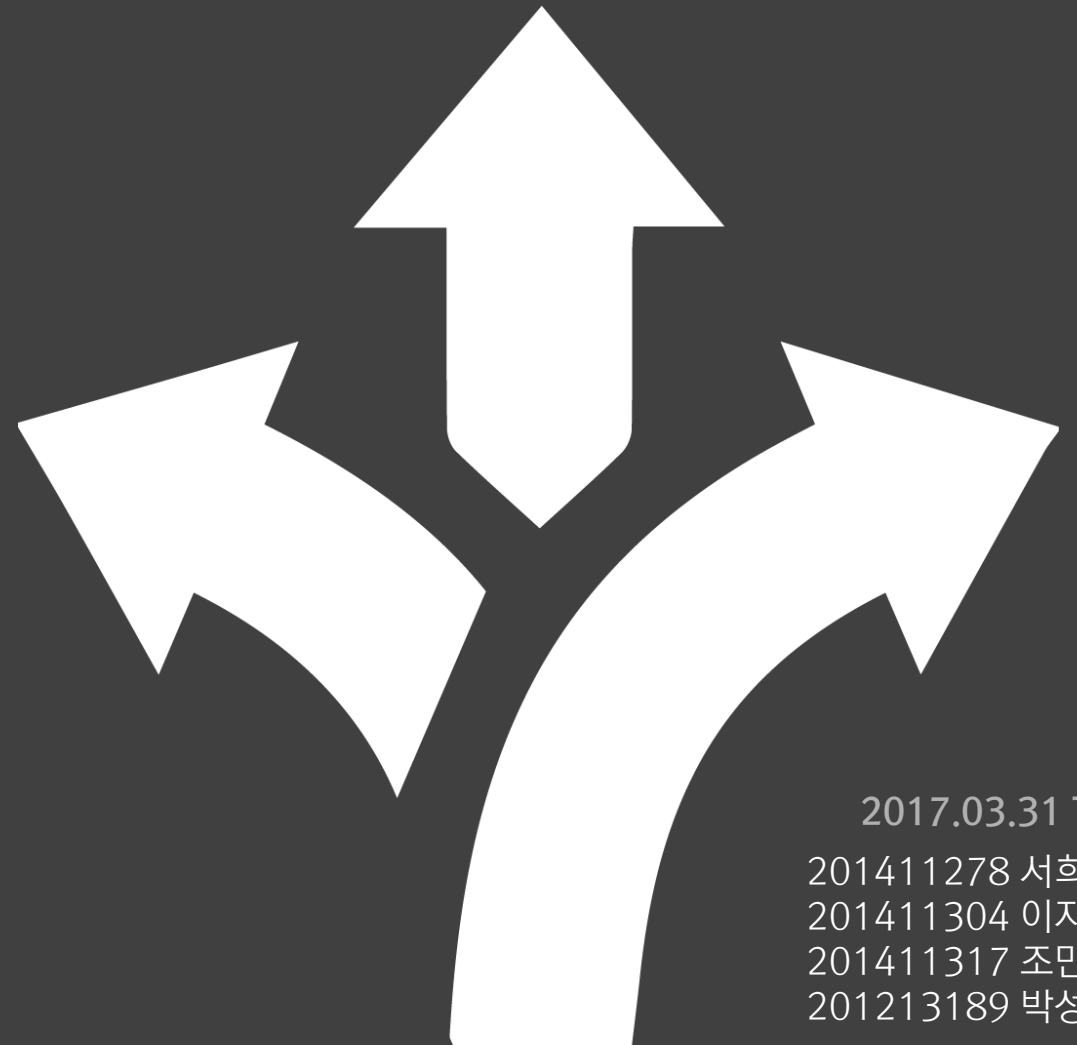


너랑나

Category Partitioning Testing Tool

Stage 1000



2017.03.31 T2

201411278 서희진
201411304 이지수
201411317 조민규
201213189 박성규

Contents

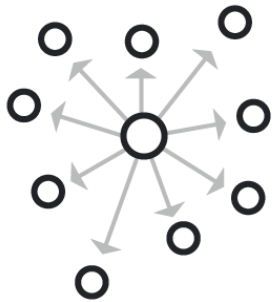
01.
Project Introduction

02.
Requirements

03.
Architecture

04.
Detailed Plan

Motivation



수많은
Test Case

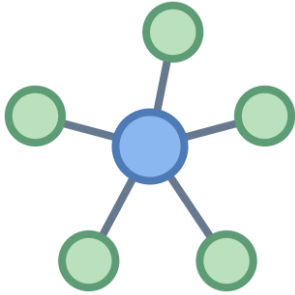


가독성이 좋지 못한
Test Case



잡은 문서 간의 공유

Project Objectives



Category를 기반으로
자동으로 Test Scripts 제작



조합된 Test Case를
한눈에 볼 수 있는 깔끔한 UI



구글 드라이브를 연동한
빠른 문서 공유

Contents

01.
Project Introduction

02.
Requirements

03.
Architecture

04.
Detailed Plan

Functional Requirements

- LOAD

Category Partitioning Testing Tool

Software Modeling & Analysis Team.T2



[메인 화면]

02. Requirements

Functional Requirements

- Analyze

Category Partitioning Testing Tool
Software Modeling & Analysis Team.T2

30 Categories

Category				
이지수	미인규	SungKyu	Software	더 이상
조민규	서영규	Minkyu	Modeling	쓸
서희진	희이진	Jisoo	Analysis	말이
박성규	고양이	일식	Team	없어
지이수	Heejin	양식	2%	웃!

조합하기

[카테고리 출력 화면]

Functional Requirements

- Analyze

Category Partitioning Testing Tool
Software Modeling & Analysis Team.T2

14 Choices

Category : 고양이

애옹애옹	Cute	1
왜옹왜옹	Noisy	2
야옹야옹	Happy	3
그릉그릉	Annoying	4

[If 민규바보] [Single]

돌아가기

[Value 출력 화면]

02. Requirements

Functional Requirements

- Make Case

Category Partitioning Testing Tool
Software Modeling & Analysis Team,T2

270 Test Cases

Test Script

1	6	11			
2	7	12		21	22
3	8	13		18	23
4	9	14		19	24
5	10	15		20	25

돌아가기

저장하기

이지수 : 룰루
 조민규 : 랄라
 서희진 : 야호
 박성규 : 까
 지이수 : 지슈슈
 미인규 : 멍키
 고양이 : 애옹애옹
 Software : 소모
 Modeling : 파이팅
 Analysis : π

[Test Case 출력 화면]


Functional Requirements

- Save

Category Partitioning Testing Tool
Software Modeling & Analysis Team,T2


오늘_저녁은_족발.txt

저장



파일로 저장

PC에 문서를 저장합니다.



클라우드에 저장

구글 드라이브에 문서를 저장합니다.

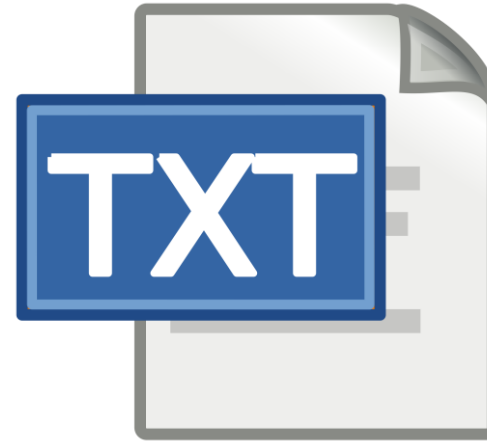
돌아가기

[파일 저장 화면]

Non-Functional Requirements



직관적인
Interface



텍스트 파일로 작성된
Test specification만 분석한다.

Contents

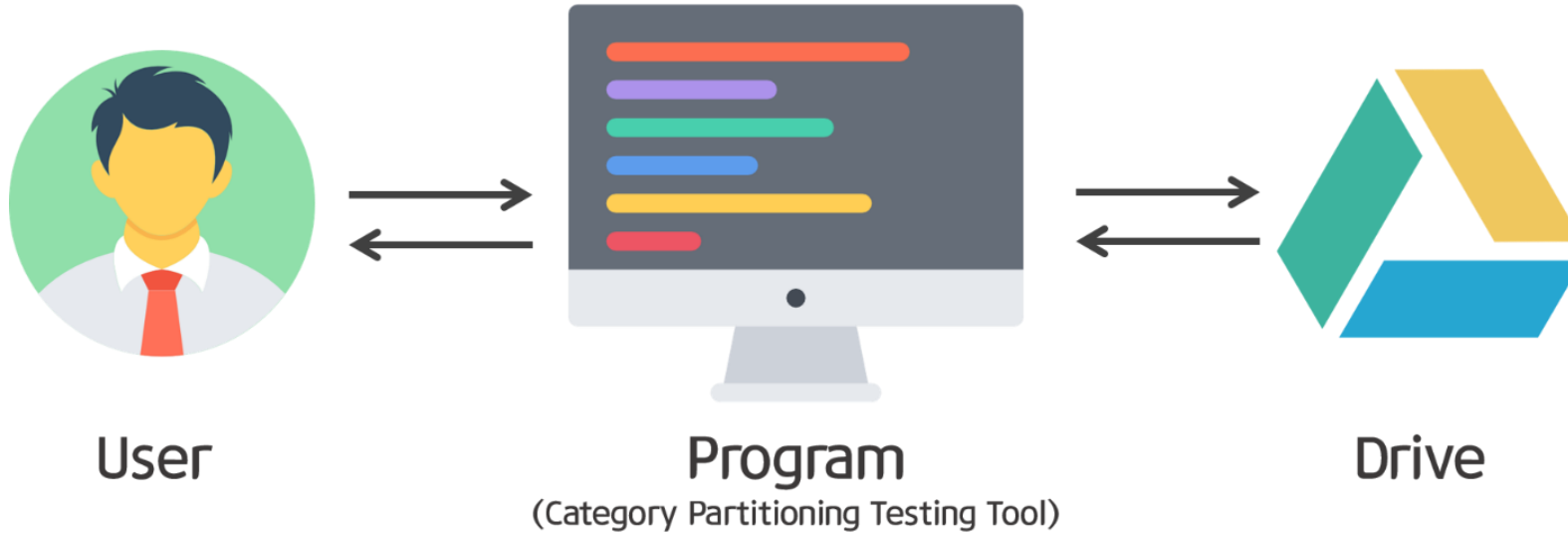
01.
Project Introduction

02.
Requirements

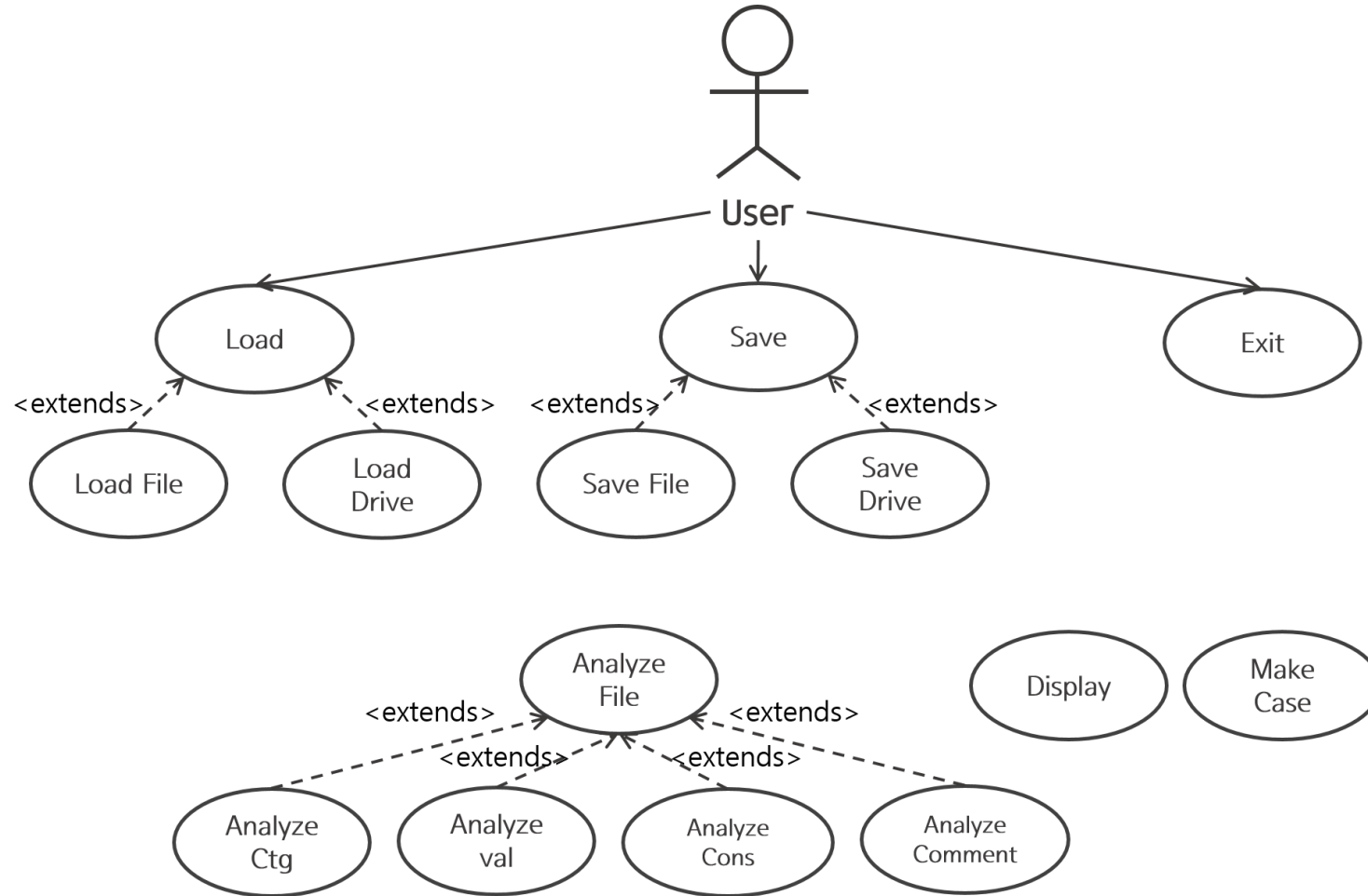
03.
Architecture

04.
Detailed Plan

System Architecture



Use Case



Contents

01.
Project Introduction

02.
Requirements

03.
Architecture

04.
Detailed Plan

04. Detailed Plan

Resource



4명의 팀원



3개월의 시간



120만원의 자원

Risk & Management

- | | | |
|---|-----------|---|
| 1 | 개발능력의부족 | 주변 동료의 소스코드를 활용, 적극적인 수업 참여와 관련 도서, 인터넷을 통해 학습한다. |
| 2 | 팀원의이탈 | 잘 다독이고 맛있는 걸 사 먹여서 기분을 좋게 만든다. |
| 3 | 디자인능력부족 | 많은 참고도서를 참고하여 디자인 한다. |
| 4 | 고양이의지나친애교 | 고양이의 까까를 물려주고 나오거나, 팀원의 집에서 프로젝트를 진행한다. |
| 5 | 지나친굶주림 | 매 만남마다 점심이나 저녁을 꼬박꼬박 챙겨 먹는다. |
| 6 | 과제의중첩 | 주변 동료의 과제를 활용하여 효율적으로 과제를 마무리한다. |
| 7 | 과도한음주및숙취 | 음주 전 여명이나 컨디션을 음용한다. |

04. Detailed Plan

Schedule

Stage	Phase/Activity	Schedule (Week)																
		1	2	3	4	5	6	7	8	9	10	11	12					
1000 Plan & Elaborate	1001. Define Draft Plan	█																
	1002. Create Preliminary Investigation Report	█																
	1003. Define Requirements	█																
	1004. Record Terms in Glossary		█															
	1005. Implement Prototype		█															
	1006. Define Draft System Architecture		█															
	1007. Define Business Use Case		█															
	1008. Define Business Concept Model		█	█														
	1009. Define System Test Case		█	█														
	1010. Refine Plan		█	█														
2000 Build	2010. Revise Plan			█	█													
	2020. Synchronize Artifacts			█	█													
	2030 Analyze	2031. Define Essential Use Case			█	█												
		2032. Refine Use Case Diagrams			█	█												
		2033. Refine Conceptual Model				█	█											
		2034. Refine Glossary				█	█											
		2035. Define System Sequence Diagrams				█	█											
		2036. Define Operation Contracts				█	█											
	2040 Design	2037. Define State Diagrams				█	█											
		2041. Define Real Use Cases					█	█										
		2042. Define Reports, UI and Storyboards					█	█										
		2043. Refine System Architecture					█	█										
		2044. Define interaction Class Diagrams					█	█										
		2045. Define Design Class Diagrams					█	█										
	2050 Construct	2046. Define Database Schema					█	█										
		2051. Implement Class & interface Definition							█	█								
		2052. Implement Methods							█	█								
		2053. Implement Windows								█	█							
		2054. Implement Reports									█	█						
		2055. Implement DB schema										█	█					
2060 Test	2056. Write Test Code										█	█						
	2061. Unit Testing											█	█					
	2062. Integration Testing												█	█				
	2063. System Testing													█	█			
	2064. Performance Testing														█	█		
	2065. Acceptance Testing															█	█	
2066. Documentation Testing																█		

Thank you