

# **OSP Stage 1000**

**- Document v\_120422**

**T 6**

**200811425** 김평석

**200811435** 신성호

**200811449** 이찬희

**200811454** 전인서

**200811462** 최현빈

# **Contents**

**1001. DefineDraft Plan**

**1002. Create Preliminary Investigation Report**

**1003. DefineRequirements**

**1004. RecordTerms in Glossary**

**1005. ImplementPrototype**

**1006. Define Business Use Case**

**1007. Define Business Concept Model**

**1008. Define DraftSystem Architecture**

**1009. Refine Plan**

## **1001. Define Draft Plan**

### **1. Motivation**

- Desktop과 Laptop의 보급률이 꾸준히 증가하고 기술의 발전으로 인하여 컴퓨터를 활용하는 인구가 증가하고, 컴퓨터의 2차 저장소의 크기가 꾸준히 증가하고 있다.
- 따라서 기존의 폴더 시스템의 단점을 보완하면서, 많은 파일 효율적으로 관리할 수 있는 관리 시스템을 필요로 하게 되었다.
- 파일/폴더에 Tag를 부여함으로써 파일관리를 효율적으로 수행할 수 있는 Tagged Database Management System(TDMS)로 사용자의 기호에 맞게 파일 시스템을 자유롭게 사용할 수 있다.

### **2. Project Objectives**

- 파일 및 폴더에 Tag를 부여하여 파일들을 효율적으로 관리할 수 있으며, 구조화하여 보여준다.
- 프로그램을 처음 쓰는 사람이라도 쉽게 모든 기능을 사용할 수 있도록 구현한다.

### **3. Functional Requirements**

- 파일 실행
- 파일 삭제
- 의존성 검사
- 파일 수정
- 파일 검색
- 파일 정보 보기
- Tag 생성

- Tag 삭제
- Tag 수정
- Tag 검색
- Tag 지정
- Tag 해제
- Tag 의존성 부여
- Tag 의존성 삭제
- Tag 구조 보기
- 최근 사용한 Tag 목록 저장
- 최근 사용한 Tag 목록 보기
- Service Directory
- Name Tag 생성
- Name Tag 삭제
- Name Tag 수정
- Name Tag 목록 보기
- Log File 생성
- Log File 보기

#### **4. Non – Functional Requirements**

- User Interface를 직관적으로 구성한다.
- 컴퓨터에서 기본적인 기능을 수행하기 때문에 dependability, reliability를 고려하여 만든다.

- 단, TDMS상에서 파일이 수정된 경우가 아니라면 dependability를 보장할 수 없다.

## **5. Resource Estimation**

- Human Efforts (Man-Month) : 3명
- Human Resource : 5명
- Project Duration : 12주
- Cost : 480,000 ₩

## **1002. Create Preliminary Investigation Report**

### **1. Alternative Solutions**

- 이미 개발된 파일 관리 시스템을 참고한다.
- 기존의 자료 관리를 위한 다양한 알고리즘을 알아본다.

### **2. Project Justification**

- cost :

- Duration :

외주를 쓰는 것보다 단축 시킬 수 있다.

- Risk :

외주를 하는 경우 유지보수를 하기 어렵다.

다른 제품을 구입하여 사용하는 경우 우리가 원하는 기능을 사용할 수 없다.

### 3. Risk management

Risk	Probability	Significance	Weight
자바에 대한 지식이 부족하다.	2	4	8
OSP를 이용한 첫 번째 프로젝트이다.	3	4	12
기존에 파일 관리 시스템을 만든 경험이 없다..	3	4	12
UML에 대한 경험이 부족하다.	4	3	12
객체지향적 설계에 대한 경험이 부족하다.	2	3	12
다른 전공과목으로 시간이 부족할 수 있다.	4	4	16
팀원들 간에 소통이 원활하지 않을 수 있다.	1	5	5

### 4. Risk Reduction Plan

#### 자바에 대한 지식이 부족하다(8)

- 팀원들 각자 java를 공부하고 서로의 지식을 공유하여 프로젝트 하는데 부족함이 없도록 한다.

#### OSP를 이용한 첫 번째 프로젝트이다.(12)

- 이전에 진행되었던 프로젝트를 참고하고, 교수님께 자문을 구한다.

#### 기존에 파일 관리 시스템을 만든 경험이 없다.(12)

- 이미 나와있는 파일 관리 시스템과, 자료 검색 알고리즘을 조사해본다.

#### UML에 대한 경험이 부족하다.(12)

- 우리 팀과 다른 팀들의 UML보고서를 참고하고 예제를 검색하여 본다.

#### 객체지향적 설계에 대한 경험이 부족하다.(12)

- 디자인 패턴 및 객체지향적인 설계에 대한 도서를 이용한다.

#### 다른 전공과목으로 시간이 부족할 수 있다.(16)

- 목요일 금요일 그리고 주말을 최대한 이용하여 한번에 끝낸다.

#### 팀원들 간에 소통이 원활하지 않을 수 있다.(5)

- 최대한 서로를 존중하며, 모임 시에 단것을 항상 주위에 둔다.

## **5. Market Analysis**

- 그 전에 만들어졌던 파일 관리 시스템을 조사해보고, TDMS의 기능 및 UI를 User에게 최적화 한다.

## **6. Other Managerial Issues**

- 이 프로젝트는 2012년 5월 4일까지 완성되어야 한다.

### **1003. Define Requirements**

#### **1. Functional Requirements**

- 파일 실행
  - TDMS에서 파일을 실행할 수 있다.
- 파일 삭제
  - TDMS상에서 존재하는 파일을 삭제할 수 있다.
  - 파일 삭제 시 의존성 검사를 시행한다.
- 의존성 검사
  - 파일 삭제 시 자동으로 실행되며, 파일의 의존성을 확인한다.
  - 의존성이 존재하는 경우는 파일을 삭제 할 수 없게 한다.
- 파일 수정
  - TDMS에서 파일의 이름을 수정할 수 있다.
  - 중복되는 이름으로 수정할 수 없다.
- 파일 검색
  - TDMS상에서 파일을 이름으로 검색할 수 있다.

- 파일 정보 보기
  - TDMS상에서 파일이 가지고 있는 정보(이름, 수정된 날짜, 지정된 Tag, Tag가 지정된 날짜, 등등)를 확인 할 수 있다.
- Tag 생성
  - Tag이름과 Tag에 대한 설명 등을 입력 받아 Tag를 생성한다.
  - 입력 받은 이름의 중복확인을 한다.
- Tag 삭제
  - 생성된 Tag를 삭제한다.
- Tag 수정
  - 생성된 Tag의 정보 수정한다.
  - 중복되는 이름으로 수정할 수 없다.
- Tag 검색
  - 해당 Tag를 가지고 있는 파일을 검색한다.
- Tag 지정
  - 파일에 생성된 Tag를 지정할 수 있다.
- Tag 해제
  - 파일에 지정된 Tag를 해제할 수 있다.
- Tag 의존성 부여
  - TDMS상의 Tag끼리의 등을 설정하여 준다.
- Tag 의존성 삭제
  - 생성된 종속, 연관관계를 삭제하여준다. 삭제 시 확인 창이 생성된다.



- Tag 구조 보기
  - Tag 연관성 부여를 통하여 설정된 Tag의 구조를 보여준다.
- 최근 사용한 Tag 목록 저장
  - TDMS상에서 최근 사용된 Tag 목록을 자동으로 저장한다.
- 최근 사용한 Tag 목록 보기
  - TDMS상에서 최근 사용된 Tag 목록을 확인할 수 있다.
- Service Directory
  - TDMS상에서 특정한 Directory에서 주기적으로 해당 Name Tag의 명령을 검사하여 자동으로 폴더 이동을 시행할 수 있다.
- Name Tag 생성
  - Service Directory상에서 실행될 Name Tag 명령(폴더위치 지정)을 생성할 수 있다.
- Name Tag 삭제
  - Service Directory상에서 실행될 Name Tag명령(폴더위치 지정)을 삭제할 수 있다.
- Name Tag 수정
  - Service Directory상에서 실행될 Name Tag명령(폴더위치 지정)을 수정할 수 있다.
- Name Tag 목록 보기
  - Service Directory에서 생성한 Name Tag의 목록을 확인할 수 있다.
- Log File 생성
  - TDMS상에서 Tag와 파일에 대한 변경(생성, 삭제, 이동, 등)을 시간에 따라 기록한다.
- Log File 보기
  - 생성된 Log File의 내용을 확인할 수 있다.

**2. Functional Requirements (Table)**

Ref. #	Function	Category
R 1.1	파일 실행	Evidence
R 1.2.1	파일 삭제	Evidence
R 1.2.2	의존성 검사	Hidden
R 1.3	파일 수정	Evidence
R 1.4	파일 검색	Evidence
R 1.5	파일 정보 보기	Evidence
R 2.1	Tag 생성	Evidence
R 2.2	Tag 삭제	Evidence
R 2.3	Tag 수정	Evidence
R 2.4	Tag 검색	Evidence
R 2.5.1	Tag 지정	Evidence
R 2.5.2	Tag 해제	Evidence
R 2.6.1	Tag 의존성 부여	Evidence
R 2.6.2	Tag 의존성 삭제	Evidence
R 2.7	Tag 구조 보기	Evidence
R 3.1	최근 사용한 Tag 목록 저장	Hidden
R 3.2	최근 사용한 Tag 목록 보기	Evidence
R 4.1	Service Directory	Evidence
R 4.2.1	Name Tag 생성	Evidence
R 4.2.2	Name Tag 삭제	Evidence
R 4.2.3	Name Tag 수정	Evidence
R 4.2.4	Name Tag 목록 보기	Evidence
R 5.1	Log File 생성	Hidden
R 5.2	Log File 보기	Evidence

**3. Performance Requirements**

- Tag를 이용한 검색은 일반 파일 또는 폴더 검색보다 느리지 않게 한다.

#### 4. Operating Environment

- A. OS: Microsoft Window XP – 7
- B. CPU : Dual core 이상
- C. Memory : 2G 이상.
- D. HDD 100GB 이상
- E. Browser : Explorer 8.0

#### 5. Interface Requirements

- 기본적으로 GUI 형식을 그대로 사용하며, 사용자가 지정된 행동(클릭, 우클릭, 더블클릭 등)을 하며 수행된다.

#### 1004. Record Term in Glossary

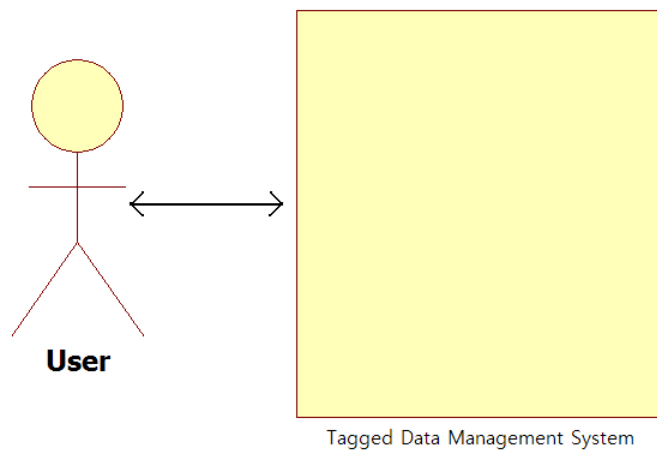
Term	Description
Attach(지정)	File에 Tag를 지정하여 특징을 표현하는 것
Detach(해제)	File에 Attach된 Tag를 해제하는 것.
Log File	TDMS상에서 실행되는 일련의 과정을 어떤 File에 저장하는 것
Tag	file이나 folder에 붙어서 파일의 특성이나 성질을 표현하는 것.
Relationship	하나 이상의 Tag들이 구성하는 Tag들의 관계를 표현하는 것.
Dependency	Tag끼리의 의존성을 나타냄. 의존을 받고 있는 Tag가 부여된 file 이나 folder는 Tag의 의존성을 삭제하거나, Tag를 제거 하기 전에 삭제 할 수 없다.
Hierarchy	Tag끼리의 계층을 나타내어 보여줌. Tag에 대한 relation과 dependency를 볼 수 있다.
Duplication	두 개 이상의 Tag를 가지고 있는 file이나 폴더를 의미한다.
Service	프로그램에서 background로 실행되어지는 기능을 의미한다.
Name Tag	이름으로 식별되어지는 Tag으로 이것을 기반으로 service가 진행되어진다.

## 1005. Implement Prototype

- **Skip**

## 1006. Define Business Use Case

### 1. Define System Boundary



### 2. Identify and Describe Actors

- **User**
  - **TDMS**의 사용자이며, **TDMS**상의 시스템 기반으로 자신이 원하는 정보를 추가, 편집, 삭제를 하는 관리자이다

### 3. Identify Use-Case

#### Use-Cases by actor-based

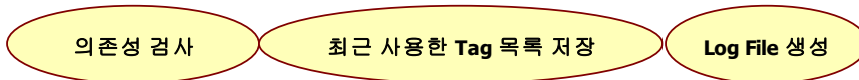
##### User



##### TDMS



#### Use-cases by event-based



### 4. Allocate System Function into Related Use-Case

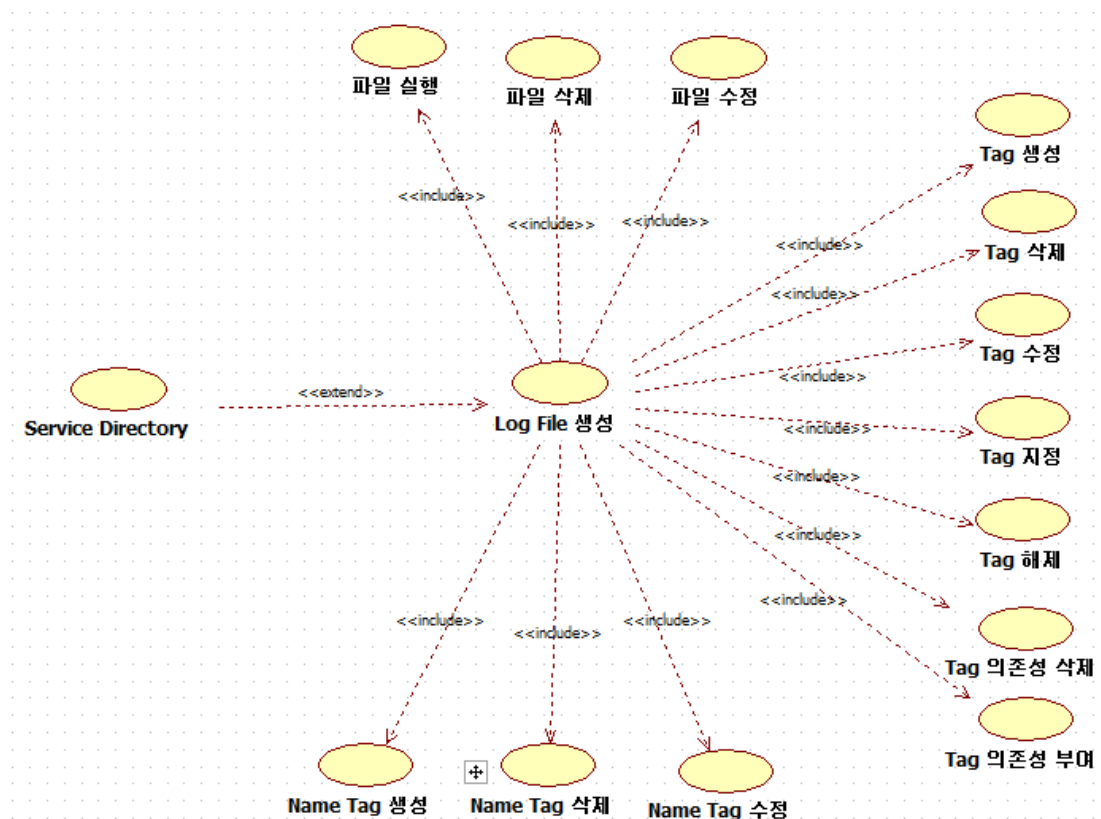
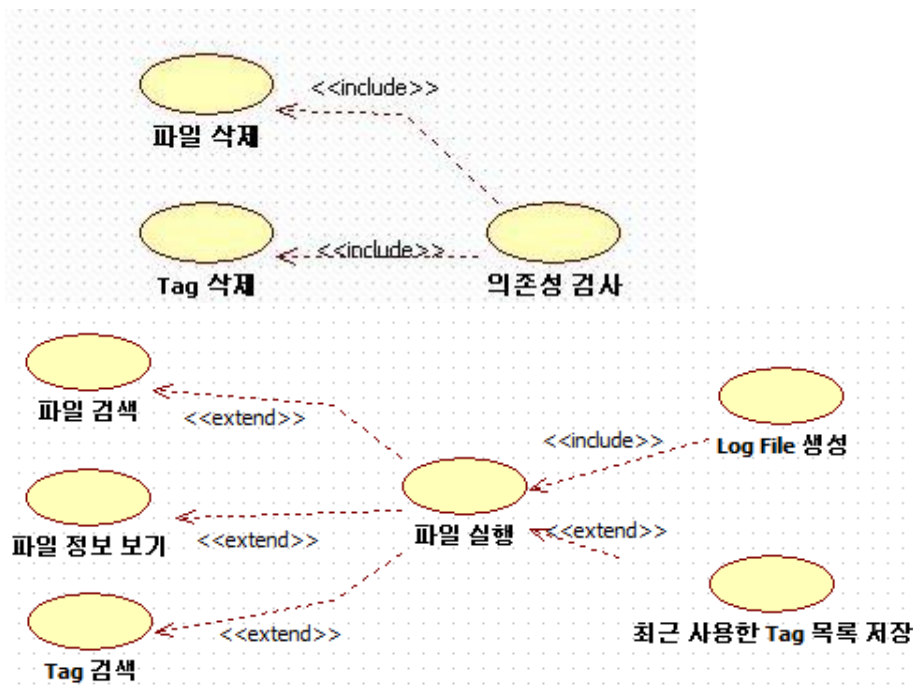
Ref. #	Function	Remarks
R 1.1	1. 파일 실행	
R 1.2.1	2. 파일 삭제	
R 1.2.2	3. 의존성 검사	
R 1.3	4. 파일 수정	
R 1.4	5. 파일 검색	
R 1.5	6. 파일 정보 보기	
R 2.1	7. Tag 생성	
R 2.2	8. Tag 삭제	

R 2.3	9. Tag 수정	
R 2.4	10. Tag 검색	
R 2.5.1	11. Tag 지정	
R 2.5.2	12. Tag 해제	
R 2.6.1	13. Tag 의존성 부여	
R 2.6.2	14. Tag 의존성 삭제	
R 2.7	15. Tag 구조 보기	
R 3.1	16. 최근 사용한 Tag 목록 저장	
R 3.2	17. 최근 사용한 Tag 목록 보기	
R 4.1	18. Service Directory	
R 4.2.1	19. Name Tag 생성	
R 4.2.2	20. Name Tag 삭제	
R 4.2.3	21. Name Tag 수정	
R 4.2.4	22. Name Tag 목록 보기	
R 5.1	23. Log File 생성	
R 5.2	24. Log File 보기	

**5. Categorize Use-Cases**

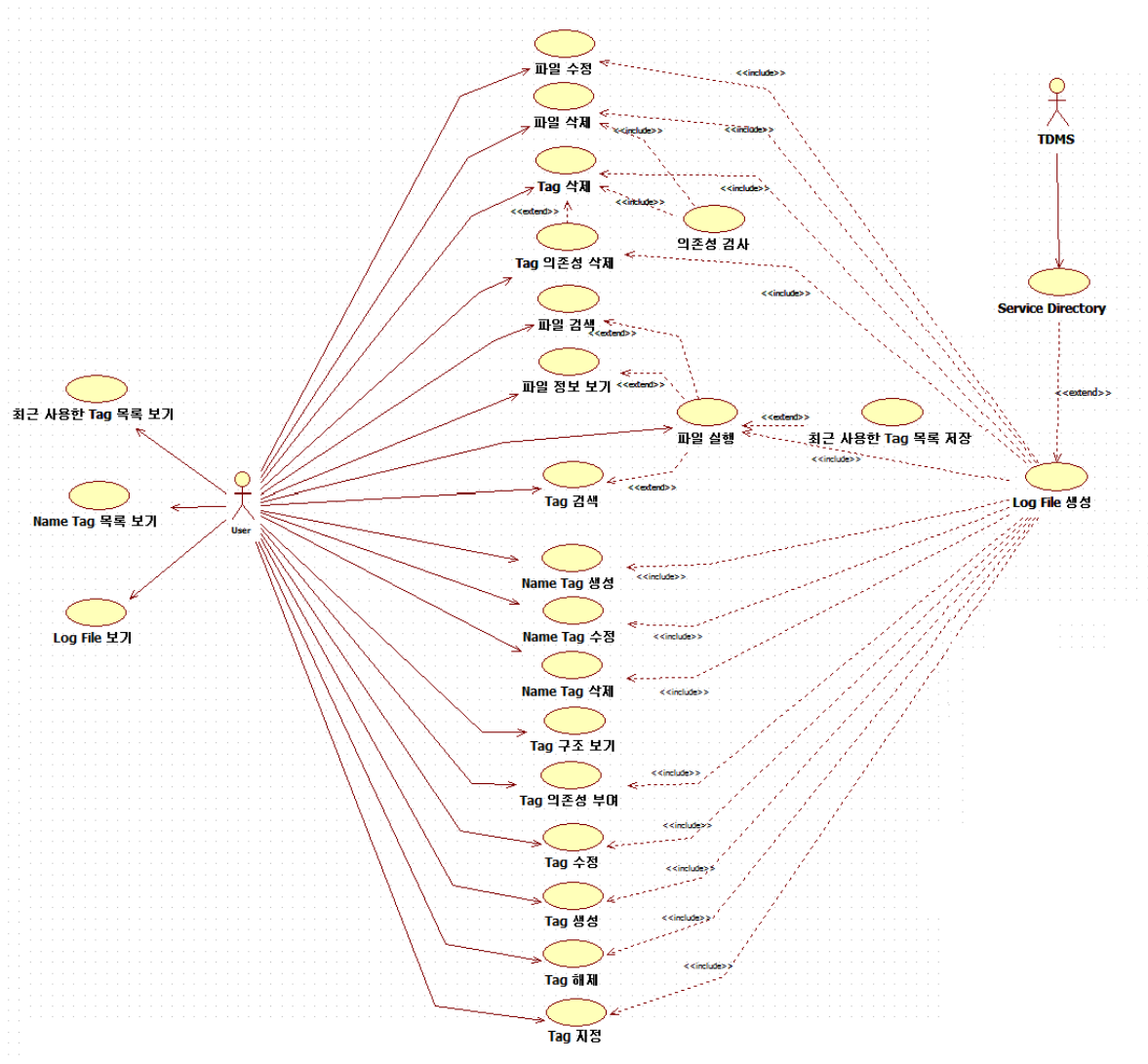
Use-Case Number & Name	Category
1. 파일 실행	Primary
2. 파일 삭제	Primary
3. 의존성 검사	Primary
4. 파일 수정	Primary
5. 파일 검색	Primary
6. 파일 정보 보기	Primary
7. Tag 생성	Primary
8. Tag 삭제	Primary
9. Tag 수정	Primary
10. Tag 검색	Primary
11. Tag 지정	Primary
12. Tag 해제	Primary
13. Tag 의존성 부여	Primary
14. Tag 의존성 삭제	Primary
15. Tag 구조 보기	Primary
16. 최근 사용한 Tag 목록 저장	Primary
17. 최근 사용한 Tag 목록 보기	Primary
18. Service Directory	Primary
19. Name Tag 생성	Primary
20. Name Tag 삭제	Primary
21. Name Tag 수정	Primary
22. Name Tag 목록 보기	Primary
23. Log File 생성	Primary
24. Log File 보기	Primary

**6. Identify the Relationships between Use-Case**





### 7. Draw a Use-Case diagram



## 8. Describe Use-Case

- Name : 1. 파일 실행  
Actor : User  
Description :
  - 이 Use case는 파일을 실행한다.
  - Log File에 기록을 남긴다
  
- Name : 2. 파일 삭제  
Actor : User  
Description :
  - 이 Use case는 파일을 삭제하는 것이다.
  - 삭제할 파일의 의존성 검사를 시행하여, 의존성 여부를 사용자에게 알리고 삭제 여부를 확인한다.
  - Log File에 기록을 남긴다
  
- Name : 3. 의존성 검사  
Actor : None  
Description :
  - 이 Use case는 삭제 시에 자동으로 실행된다.
  - 해당 파일 또는 폴더에 지정된 Tag 의 의존성을 검사하여, 그 결과를 알리고 정말 삭제할 것인지 확인 창을 띄워준다.
  
- Name : 4. 파일 수정  
Actor : User  
Description :
  - 이 Use case는 파일의 이름을 수정한다.
  - Log File에 기록을 남긴다
  
- Name : 5. 파일 검색  
Actor : User  
Description :
  - 이 Use case는 파일의 이름을 기반으로 찾는다.
  - 검색 시행 시 여러 가지 옵션을 추가할 수 있다.
  
- Name : 6. 파일 정보 보기  
Actor : User  
Description :
  - 이 Use case는 파일의 기본적인 정보를 확인 할 수 있다.

- 이름, 수정된 날짜, 지정된 Tag, Tag가 지정된 날짜 등의 정보를 확인할 수 있다.
  
- Name : 7. Tag 생성  
Actor : User  
Description :
  - 이 Use case는 사용자가 특정파일에 Tag를 지정할 수 있도록 한다.
  - 사용자가 위치를 지정하여 생성하며, 폴더이름을 입력 받는다.
  - Log File에 기록을 남긴다
  
- Name : 8. Tag 삭제  
Actor : User  
Description :
  - 이 Use case는 Tag를 삭제하는 역할을 한다.
  - 시행 시 사용자에게 삭제여부를 확인한다.(Tag를 등록한 파일들을 보여준다)
  - 삭제를 시행 하면 해당 Tag를 등록하고 있는 파일들을 찾아 해제하고, 파일들의 리스트와 체크박스를 띄워 파일도 삭제할지 여부를 결정한다.
  - Log File에 기록을 남긴다
  
- Name : 9. Tag 수정  
Actor : User  
Description :
  - 이 Use case는 Tag의 여러 가지 정보들을 항목별로 수정 가능하도록 한다.
  - 수정 시 사용자로부터 실행 여부를 확인한다.
  - 수정이 완료되면 성공메시지를 보여준다.
  - Log File에 기록을 남긴다
  
- Name : 10. Tag 검색  
Actor : User  
Description :
  - 이 Use case는 사용자에게 검색 옵션을 선택하도록 한다.
  - 사용자 옵션에 따라 Tag를 검색하여 옵션에 맞게 사용자에게 보여준다.
  
- Name : 11. Tag 지정  
Actor : User  
Description :
  - 이 Use case는 사용자가 특정파일에 Tag를 지정할 수 있도록 한다.

- 미리 등록된 Tag를 부여하거나 사용자가 새로운 Tag를 등록할 수 있도록 한다.
- Tag정보를 이용하여 Hierarchy를 형성한다.
- Log File에 기록을 남긴다.

Name : 12. Tag 해제

Actor : User

Description :

- 이 Use case는 특정파일에 사용자가 지정한 Tag를 해제시킨다.
- 파일을 삭제할 것인지 물어본다.
- 해제가 완료되면 성공메시지를 보여준다.

- Name : 13. Tag 의존성 부여

Actor : User

Description :

- 이 Use case는 사용자가 Tag가 부여된 파일들의 관계를 지정하여 Hierarchy를 형성할 수 있도록 한다.
- 관계부여시 실행 현재 형성된 Hierarchy를 보여 주어 선택하도록 한다.

- Name : 14. Tag 의존성 삭제

Actor : User

Description :

- 이 Use case는 사용자가 관계 삭제를 실행하면 Hierarchy를 보여주고 삭제할건지 여부를 확인한다.
- 관계를 삭제하고 성공메시지를 출력한다.
- Log File을 기록을 남긴다.

- Name : 15. Tag 구조 보기

Actor : User

Description :

- Tag목록 들의 Hierarchy를 보여준다.
- 사용자가 Hierarchy상에서 수정이 가능하도록 한다..

- Name : 16. 최근 사용한 Tag 목록 저장

Actor : None

Description :

- 이 Use case는 Tag 삭제, 이동, 수정에 대한 동작이 일어나면 로그를 남기고 최근 목록에 순차적으로 저장한다.
  
- Name : 17. 최근 사용한 Tag 목록 보기  
Actor : User  
Description :
  - 이 Use case는 사용자가 실행하면 최근 사용한 Tag목록을 10개 이내로 보여준다.
  
- Name : 18. Service Directory  
Actor : TDMS  
Description :
  - 이 Use case는 TDMS상에서 지정한 Directory에서 주기적으로 해당 Name Tag 의 명령을 검사하여 자동으로 폴더 이동을 시행한다.
  - Log File에 기록을 남긴다.
  
- Name : 19. Name Tag 생성  
Actor : User  
Description :
  - 이 Use case는 Service Directory상에서 실행되는 Name Tag를 생성한다.
  - Name Tag의 이름과 해당 폴더를 지정하여 준다.
  
- Name : 20. Name Tag 삭제  
Actor : User  
Description :
  - 이 Use case는 Service Directory상에서 실행되는 Name Tag를 삭제한다.
  - 삭제 시 확인 창을 띄워준다.
  
- Name : 21. Name Tag 수정  
Actor : User  
Description :
  - 이 Use case는 Service Directory상에서 실행되는 Name Tag를 수정한다.
  - 각각의 이름 또는 지정 폴더위치를 수정할 수 있다.
  
- Name : 22. Name Tag 목록 보기  
Actor : User  
Description :

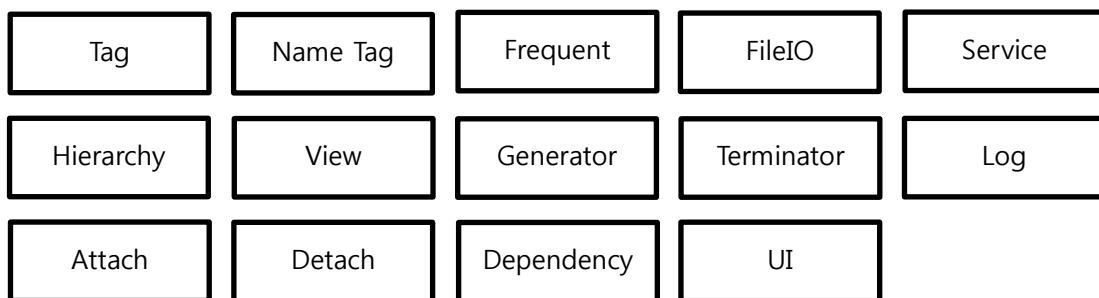
- 이 Use case는 Service Directory상에서 실행되는 Name Tag의 모든 목록을 보여준다.
- 이름과 해당되는 폴더위치를 보여준다.
- Name : 23. Log File 생성  
 Actor : None  
 Description :
  - 이 Use case는 파일 생성, 삭제, 수정, Tag 생성, 삭제, 수정, 지정, 해제, 연관, 연관삭제 등의 대부분의 수행이 Log File에 기록된다.
- Name : 24. Log File 보기  
 Actor : User  
 Description :
  - 이 Use case는 기록된 Log File의 내용을 확인할 수 있다.

**9. Rank Use-Case**

Rank	Use-Case Number & Name
Medium	1. 파일 실행
Medium	2. 파일 삭제
Medium	3. 의존성 검사
Medium	4. 파일 수정
Medium	5. 파일 검색
Medium	6. 파일 정보 보기
High	7. Tag 생성
High	8. Tag 삭제
High	9. Tag 수정
High	10. Tag 검색
High	11. Tag 지정
High	12. Tag 해제
High	13. Tag 의존성 부여
High	14. Tag 의존성 삭제
High	15. Tag 구조 보기
Medium	16. 최근 사용한 Tag 목록 저장
Medium	17. 최근 사용한 Tag 목록 보기

High	18. Service Directory
High	19. Name Tag 생성
High	20. Name Tag 삭제
High	21. Name Tag 수정
High	22. Name Tag 목록 보기
Medium	23. Log File 생성
Medium	24. Log File 보기

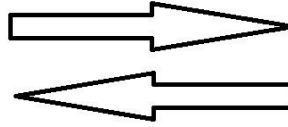
**1007. Define Business Concept Model**



## 1008. Define Draft System Architecture



User



## 1009. Refine Plan

### 1. project scope

- TDMS (Tagged Data Manage System)은 기존의 폴더 기반의 파일 관리 시스템을 보완하여 기존의 파일 관리뿐만 아니라 Tag를 이용하여 빠르고 간편하게 파일 및 폴더를 관리 할 수 있으며, Tag된 파일들의 관계 및 의존성을 볼 수 있다.

### 2. Project Object

- TDMS을 개발하여 보다 편리하게 파일에 접근 할 수 있으며, Tag된 파일을 통하여 자동으로 서비스를 제공하여 사용자에게 편의성을 제공한다. 또한 Tag된 파일의 관계 및 의존성을 설정 하여 파일 관리에 도움을 주게 한다.

### 3. functional requirement

- 파일 실행
- 파일 삭제
- 의존성 검사
- 파일 수정



## Software Modeling & Analysis

OSP Stage 1000 v\_120422

- 파일 검색
- 파일 정보 보기
- Tag 생성
- Tag 삭제
- Tag 수정
- Tag 검색
- Tag 지정
- Tag 해제
- Tag 의존성 부여
- Tag 의존성 삭제
- Tag 구조 보기
- 최근 사용한 Tag 목록 저장
- 최근 사용한 Tag 목록 보기
- Service Directory
- Name Tag 생성
- Name Tag 삭제
- Name Tag 수정
- Name Tag 목록 보기
- Log File 생성
- Log File 보기

#### **4. Performance Requirements**

- **Tag**를 이용한 검색은 일반 파일 또는 폴더 검색보다 느리지 않게 한다.

#### **5. Operation Environment**

- OS: Microsoft Window XP – 7
- CPU : Dual core 이상
- Memory : 2G 이상.
- HDD 100GB 이상
- Browser : Explorer 8.0

#### **6. User Interface Requirements**

- Window-based를 기본으로 하여 User가 익숙하게 사용할 수 있도록 한다.
- 각 메뉴에 따라서 여러 가지 기능들이 동작한다..
- Name Tag는 CUI와 비슷한 형식을 갖기 때문에 도움말을 제공하여 사용자가 쉽게 이용할 수 있도록 한다.
- 

#### **7. Other Requirements**

- 프로그램 외에서 프로그램에서 저장하고 있는 파일을 변경 시에 reliability와 dependency를 보장 할 수 없다.
- 프로그램의 정보는 file로 저장이 되며, file은 외부에서 수정 없이 보호되어야 한다.

## **8. Resource**

- Man Month : 3 People
  - i. Team Leader / Engineer / Document Manager 는 서로 겸한다.
  - ii. Period : 12 Weeks (include three cycles.)
- Hardware
  - i. OS : Microsoft Windows XP – 7
  - ii. CPU : Dual Core

**9. Scheduling**

STage	Phase(00x0)/ Activity(000x)	Schedule(Week)											
		1	2	3	4	5	6	7	8	9	10	11	12
<b>1000 Plan&amp; Elaborate</b>	1001. Define Draft Plan	■											
	1002. Create Preliminary Investigation Report	■											
	1003. Define Requirements	■											
	1004. Record Terms in Glossary	■											
	1006. Define Use Cases		■										
	1007 Define Draft Conceptual Model			■									
	1008. Define Draft System Architecture			■									
	1009. Refine Plan			■									
	<b>2000 build</b>	<b>2n10. Revise Plan</b>											
<b>2n20. Synchronize Artifacts</b>													
<b>2n30. Analyze</b>													
2n31. Define Essential Use Case			■							■			
2n32. Refine Use Case Diagrams			■							■			
2n33. Refine Conceptual Model			■							■			
2n34. Refine Glossary			■							■			
2n35. Define System Sequence Diagrams				■						■			
2n36. Define Operation Contracts				■						■			
2n37. Define State Diagrams				■						■			
<b>2n40. Design</b>													
2n41. Define Real Use Cases				■						■			
2n42. Define Reports, UI and Storyboards				■	■					■			
2n43. Refine System Architecture				■						■			
2n44. Define Interaction Diagrams				■						■			
<b>2n50. Construct</b>													
2n51. Implement Class & Interface Definition							■				■		
2n52. Implement Methods.								■	■		■		
2n53. Implement Windows									■	■	■		
2n54. Implement Reports										■	■		