

# **OSP Stage 2140**

**- 1<sup>st</sup> Cycle**

**T 6**

**200811425** 김평석

**200811435** 신성호

**200811449** 이찬희

**200811454** 전인서

**200811462** 최현빈

# **Contents**

**2141. Design Real Use Cases**

**2142. Define Reports, UI, and Storyboards**

**2143. Refine System Architecture**

**2144. Define Interaction Diagrams**

**2145. Define Design Class Diagrams**

**Stage 2141. Design real Use Cases**

Use Case	파일 실행
Actor	User
Purpose	사용자가 파일 및 폴더를 실행한다.
Overview	지정한 파일 및 폴더를 실행한다. Log File에 기록을 남긴다.
Type	Primary and Essential
Cross Reference	R 1.1, R 1.4, R 1.5, R 2.4, R 3.1, R 5.1 Use Case : 파일 검색, 파일 정보 보기, Tag 검색, 최근 사용한 Tag 목록 저장, Log File 생성
Pre-Requisites	window-a에서 파일 메뉴가 선택된 상태여야 한다. 파일 및 폴더가 존재하여야 한다.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : window-c에서 실행하고자 하는 <i>file</i> 을 더블클릭 하여 실행한다.. 2. (S) : 선택한 <i>file</i> 을 실행한다. 3. (S) : <i>LogFile</i> 을 생성한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

Use Case	파일 삭제
Actor	User
Purpose	사용자가 파일을 삭제한다.
Overview	삭제할 파일의 의존성 검사를 시행하여, 의존성 여부를 사용자에게 알리고 삭제 여부를 확인한다. Log File에 기록을 남긴다.
Type	Primary and Essential
Cross Reference	R 1.2.1 R 1.2.2 R 5.1 Use Case : 의존성 검사, Log File 생성
Pre-Requisites	window-a에서 파일 메뉴가 선택된 상태여야 한다. 파일이 생성되어 있는 상태여야 함
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : window-c에서 삭제하고자 하는 <i>file</i> 을 선택하고, 우클릭 하여 파일삭제 기능을 선택한다. 2. (S) : <i>file</i> 의 <i>dependency</i> 를 체크한다. 3. (S) : 삭제 여부를 확인하는 팝업 창을 보여준다.

	4. (A) : 팝업 창의 삭제여부를 결정한다. 5. (S) : <i>Logfile</i> 을 생성한다. 6. (S) : 선택한 <i>file</i> 을 삭제한다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	2. 파일에 의존성이 있으면 메시지를 보여주고 삭제를 취소한다. 4. 삭제 요청을 취소할 시, 이후 실행 없이 원상태로 돌아간다.

<b>Use Case</b>	<b>의존성 검사</b>
<b>Actor</b>	System
<b>Purpose</b>	해당 파일 및 폴더의 의존성을 검사하여 사용자에게 출력한다.
<b>Overview</b>	삭제요청을 받을 시에 자동으로 실행된다. 해당 파일 또는 폴더에 지정된 Tag의 의존성을 검사하여, 그 결과를 알린다
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 1.2.1, R 1.2.2, R 2.2 Use Case : 파일 삭제, Tag 삭제
<b>Pre-Requisites</b>	User가 파일 삭제, Tag 삭제 중 하나를 수행해야 한다
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 해당 기능의 연산의 대상인 파일이나 Tag의 있는 <i>childTag</i> 와 <i>parantTag</i> 를 검사한다. 2. (S) : <i>childTag</i> 와 <i>parantTag</i> 가 지정되어 있다면 의존성 있는 것이므로 <i>showPopup("Error")</i> 를 사용하여 사용자에게 알려준다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	NONE

<b>Use Case</b>	<b>파일 수정</b>
<b>Actor</b>	User
<b>Purpose</b>	사용자가 파일 및 폴더의 이름을 수정한다.
<b>Overview</b>	파일 및 폴더의 이름을 수정하고 Log File에 기록을 남긴다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 1.3, R 5.1 Use Case : Log File 생성
<b>Pre-Requisites</b>	window-a에서 파일 메뉴가 선택된 상태여야 한다. 파일 및 폴더가 존재하여야 한다
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : window-c에서 수정하고자 하는 <i>file</i> 을 선택하고, 우클릭

	<p>하여 파일 수정 기능을 선택한다.</p> <p>2. (S) : window-e에 파일 수정 화면을 보여준다.</p> <p>3. (A) : window-e에 새로운 <i>file</i>의 <i>name</i>을 입력한다</p> <p>4. (S) : 입력 받은 <i>name</i>이 중복되는지 확인한다.</p> <p>5. (S) : 선택한 <i>file</i>의 <i>name</i>을 바꿔준다.</p> <p>6. (S) : <i>Logfile</i>을 생성한다.</p>
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	4. 만약 중복된 이름이면 메시지를 출력하고 재입력을 요청한다.

<b>Use Case</b>	<b>파일 검색</b>
<b>Actor</b>	User
<b>Purpose</b>	파일 및 폴더를 이름을 기반으로 찾는다.
<b>Overview</b>	사용자가 파일 및 폴더를 검색하고, 검색 시행 시 여러 가지 옵션을 추가할 수 있다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 1.1, R 1.4 Use Case : 파일 실행
<b>Pre-Requisites</b>	window-a에서 파일 메뉴가 선택된 상태여야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System <p>1. (A) : window-e에서 파일 검색 찾고자 하는 파일 이름인 <i>name</i>과 태그된 파일 중 찾을 것인지에 대한 여부 <i>bTagged</i> 선택하거나, 검색 위치가 되는 <i>location</i>입력한다.</p> <p>2. (A) : 검색 버튼을 누른다..</p> <p>3. (S) : 태그 된 파일을 찾는다면 <i>Tagged File List</i>에서 <i>name</i>을 찾고, 태그 된 파일이 아니면 입력된 <i>Location</i>의 <i>sub directory</i>에서 <i>fileName</i>을 <i>sub String</i>으로 가지고 있는</p> <p>4. (S) : 검색 결과를 window-c에 출력해준다.</p>
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	NONE

<b>Use Case</b>	<b>파일 정보 보기</b>
<b>Actor</b>	User
<b>Purpose</b>	파일 기본적인 정보를 확인할 수 있다.
<b>Overview</b>	이름, 수정된 날짜, 지정된 Tag, Tag가 지정된 날짜 등의 정보를 확인할 수 있다.
<b>Type</b>	Primary and Essential

<b>Cross Reference</b>	R 1.1, R 1.5 Use Case : 파일 실행
<b>Pre-Requisites</b>	window-a에서 파일 메뉴가 선택된 상태여야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : window-c에서 정보를 보고자 하는 파일을 선택한다.. 2. (S) : 파일 정보를 window-d에 출력한다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	NONE

<b>Use Case</b>	<b>Tag 생성</b>
<b>Actor</b>	User
<b>Purpose</b>	새로운 Tag를 만든다.
<b>Overview</b>	파일 및 폴더에 지정할 Tag를 만든다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 2.1, R 5.1 use case : Log File 생성
<b>Pre-Requisites</b>	window-a에서 Tag 메뉴가 선택된 상태여야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : window-b에서 Tag생성 버튼을 선택한다.. 2. (A) : window-c에 태그의 이름 <i>name</i> , 종류 <i>type</i> , 색깔 <i>color</i> , 의존성 <i>dependency</i> 정보를 입력한다. 3. (A) : 태그 생성 버튼을 누른다. 4. (S) : 입력된 <i>Tag</i> 이름이 중복되었는지 체크한다. 5. (S) : <i>Tag</i> 를 생성한다. 6. (S) : <i>LogFile</i> 을 생성한다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	4. 중복된 이름이 존재할 시, 재입력을 요청한다.

<b>Use Case</b>	<b>Tag 삭제</b>
<b>Actor</b>	User
<b>Purpose</b>	기존의 Tag를 삭제한다.
<b>Overview</b>	지정한 Tag를 삭제한다. 삭제할 Tag는 의존성이 없어야 한다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 1.2.2, R 2.2, R 2.6.2., R 5.1 use case : 의존성 검사, Tag 의존성 삭제, Log File 생성

<b>Pre-Requisites</b>	window-a에서 Tag 메뉴가 선택된 상태여야 한다. 삭제하고자 하는 Tag가 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : window-b의 태그 목록 중에서 삭제할 <i>tag</i> 를 선택, 우클릭 하여 tag삭제를 선택한다. 2. (S) : <i>Tag</i> 의 의존성 검사를 한다. 3. (S) : Tag 삭제 여부를 다시 한번 팝업으로 확인한다. 4. (A) : 팝업 여부를 선택한다. 5. (S) : 해당 Tag를 삭제 하겠다고 선택하였으면 <i>tag</i> 를 삭제한다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	2. Tag에 의존성이 있을 시, 메시지를 출력하고 원상태로 복귀한다. 4. 팝업 창에서 취소하면 원상태로 복귀한다.

<b>Use Case</b>	<b>Tag 수정</b>
<b>Actor</b>	User
<b>Purpose</b>	Tag의 정보를 수정한다.
<b>Overview</b>	Tag를 지정하여 정보(이름, 의존성, Tag 설명 등)를 수정한다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 2.3 , R 5.1 use case : Log File 생성
<b>Pre-Requisites</b>	window-a에서 Tag 메뉴가 선택된 상태여야 한다. 수정하고자 하는 Tag가 이미 존재하여야 한다.
<b>Typical Courses of Events</b>	(A): Actor, (S) : System 1. (A) : window-b의 태그 목록에서 수정할 <i>Tag</i> 를 선택, 우클릭 하여 Tag 수정을 선택한다. 2. (S) : window-c에 해당 <i>Tag</i> 의 수정화면을 보여준다. 3. (A) : window-c에 태그의 이름 <i>name</i> , 종류 <i>type</i> , 색깔 <i>color</i> , 의 존성 <i>dependency</i> 정보를 수정한다. 4. (S) : 입력된 Tag의 정보 중 <i>name</i> 이 중복되었는지 체크한다. 5. (S) : <i>Tag</i> 의 정보를 수정한다. 6. (S) : <i>LogFile</i> 을 생성한다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	4. 입력된 Tag의 이름이 존재할 시, 메시지를 출력하고 재입력을 요청한다.

Use Case	Tag 검색
Actor	User
Purpose	존재하는 Tag를 검색한다.
Overview	지정한 검색조건으로 존재하는 Tag들을 검색한다.
Type	Primary and Essential
Cross Reference	R 1.1, R 2.4 use case : 파일 실행
Pre-Requisites	window-a에서 Tag 메뉴가 선택된 상태여야 한다.
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 window-B에서 Tag검색 기능을 선택한다. 2. (S) : window-B에 Tag 검색화면을 출력한다. 3. (A) : window-B에 검색할 Tag의 name을 입력한다. 4. (S) : tag list에서 해당 name을 갖는 tag를 검색한다. 5. (S) : 검색결과를 window-C에 출력한다.
Alternative Courses of Events	NONE.
Exceptional Courses of Events	NONE.

Use Case	Tag 지정
Actor	User
Purpose	파일, 폴더에 Tag를 지정한다.
Overview	존재하는 Tag를 파일, 폴더에 지정합니다.
Type	Primary and Essential
Cross Reference	R 2.5.1 , R 5.1 Use Case : Log File 생성
Pre-Requisites	window-a에서 Tag 메뉴가 선택된 상태여야 한다. 사용자가 입력한 파일에 지정될 Tag가 이미 생성되어 있어야 한다.
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 window-C에서 file을 선택한다. 2. (S) : window-B에 Tag 지정 기능을 활성화 시킨다. 3. (A) : window-B의 Tag 지정 기능 또는 file을 우클릭하여 Tag 지정 기능을 선택한다. 4. (S) : window-C에 선택 가능한 Tag들을 출력한다. 5. (A) : window-C에서 Tag를 선택한다. 6. (S) : 해당 file에 선택한 Tag를 지정하여준다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE



Use Case	Tag 해제
Actor	User
Purpose	파일, 폴더의 Tag를 해제한다.
Overview	파일을 더 이상 Tag를 이용하여 관리할 필요가 없을 때, 파일에 지정된 Tag를 해제해 준다.
Type	Primary and Essential
Cross Reference	R 2.5.2 , R 5.1 use case : Log File 생성
Pre-Requisites	window-a에서 Tag 메뉴가 선택된 상태여야 한다. 파일에 Tag를 해제하기 위해서는, 그 파일에 Tag가 이미 지정되어 있어야 한다.
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 window-C에서 <i>file</i> 을 선택한다. 2. (S) : 선택된 <i>file</i> 에 지정된 <i>Tag</i> 가 있는지 확인한다. 3. (A) : <i>file</i> 을 우클릭하여 Tag 해제 기능을 선택한다. 4. (S) : Tag해제 여부를 팝업으로 확인한다. 5. (A) : Tag해제 여부를 선택한다. 6. (S) : 해당 <i>file</i> 의 Tag를 해제한다.
Alternative Courses of Events	7. Tag가 지정된 파일을 삭제한다면, 그 파일에 대한 Tag를 해제해준다.
Exceptional Courses of Events	2. 선택한 파일에 Tag가 지정되어 있지 않다면, 메시지를 출력하고 원상태로 복귀한다..

Use Case	Tag 의존성 부여
Actor	User
Purpose	Tag에 다른 Tag와의 의존성을 부여한다.
Overview	파일에 보안성을 부여하기 위해서, Tag에 의존성을 부여 할 수 있어야 한다. 그러기 위한 Tag에 의존성을 부여한다.
Type	Primary and Essential
Cross Reference	R 2.6.1, R 5.1 use case : Log File 생성
Pre-Requisites	window-a에서 Tag 메뉴가 선택된 상태여야 한다. Tag에 의존성을 부여하기 위해서는 Tag가 미리 생성되어 있어야 한다.
Typical Courses of Events	(A) : Actor (S) : System

	<ol style="list-style-type: none"> <li>1. (A) : 사용자가 window-C에서 <i>Tag</i>를 선택한다.</li> <li>2. (S) : window-B에 Tag 의존성 부여 기능을 활성화 시킨다.</li> <li>3. (A) : window-B의 Tag 의존성 부여를 선택하거나, <i>file</i>을 우클릭하여 Tag 의존성 부여 기능을 선택한다.</li> <li>4. (S) : window-C에 의존성 부여가 가능한 <i>Tag</i>들을 출력한다.</li> <li>5. (A) : window-C에서 의존성을 부여할 <i>Tag</i>를 선택한다.</li> <li>6. (S) : 해당 <i>Tag</i>에 의존성을 부여해 준다.</li> </ol>
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	NONE

<b>Use Case</b>	<b>Tag 의존성 삭제</b>
<b>Actor</b>	User
<b>Purpose</b>	Tag간의 의존성을 삭제한다.
<b>Overview</b>	Tag의 삭제 및 구조를 변경하기 위해서 Tag의 의존성을 삭제한다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 2.2, R 2.6.2 , R 5.1 Use Case : Tag 삭제, Log File 생성
<b>Pre-Requisites</b>	window-a에서 Tag 메뉴가 선택된 상태여야 한다. Tag에 의존성을 삭제하기 위해서는 Tag가 미리 생성되어 있어야 하며, 그 Tag에 의존성이 존재하여야 한다.
<b>Typical Courses of Events</b>	(A) : Actor (S) : System <ol style="list-style-type: none"> <li>1. (A) : 사용자가 window-C에서 <i>Dependency</i>를 선택한다.</li> <li>2. (A) : window-B에서 Tag 의존성 삭제 기능을 선택한다.</li> <li>3. (S) : 선택한 <i>Dependency</i>를 삭제할 수 있게 <i>Dependency</i> 구조를 재구성한다.</li> <li>4. (S) : 선택된 <i>Dependency</i>를 삭제한다.</li> </ol>
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	2. window-C에서 <i>Dependency</i> 를 선택하지 않았다면, 선택 후 요청하라는 메시지를 출력한다.

<b>Use Case</b>	<b>Tag 구조 보기</b>
<b>Actor</b>	User
<b>Purpose</b>	파일들의 구조를 보여준다
<b>Overview</b>	Tag목록들의 Hierarchy를 보여준다. 파일간의 연관성을 토대로 트리 형태의 구조를 보여준다.

	사용자가 Hierarchy상에서 수정이 가능하도록 한다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 2.7
<b>Pre-Requisites</b>	NONE
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : window-A에서 Tag 메뉴를 선택한다. 2. (S) : window-C에 Tag의 Dependency를 Tree 구조로 출력한다.
<b>Alternative Courses of Events</b>	1. Use Case 'Tag 의존성 부여', 'Tag 의존성 삭제'를 실행한다.
<b>Exceptional Courses of Events</b>	NONE

<b>Use Case</b>	<b>최근 사용한 Tag 목록 저장</b>
<b>Actor</b>	User
<b>Purpose</b>	최근 사용한 Tag의 목록을 저장한다
<b>Overview</b>	파일을 실행할 때 최근 목록에 순차적으로 저장한다. 최근 사용한 Tag목록은 10개까지 저장을 한다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 1.1, R 3.1 Use Case : 파일 실행
<b>Pre-Requisites</b>	파일을 실행시킨다
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 최근 실행한 tag의 name를 저장한다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	NONE

<b>Use Case</b>	<b>최근 사용한 Tag 목록 보기</b>
<b>Actor</b>	User
<b>Purpose</b>	최근 사용한 Tag의 목록을 보여준다
<b>Overview</b>	최근 사용한 Tag의 목록을 10개 이내로 보여준다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 3.2
<b>Pre-Requisites</b>	window-a에서 Tag 메뉴가 선택된 상태여야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : window-B에서 최근 사용한 Tag 목록 보기 기능을 선택한다. 2. (S) : window-C에 최근 사용한 Tag의 목록을 출력한다.
<b>Alternative Courses of Events</b>	NONE

<b>Exceptional Courses of Events</b>	NONE
--------------------------------------	------

<b>Use Case</b>	<b>ServiceDirectory</b>
<b>Actor</b>	User
<b>Purpose</b>	Nametag를 이용하여 지정된 directory에서 파일 자동 분류 기능을 수행한다.
<b>Overview</b>	자동으로 파일을 분류한다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 4.1
<b>Pre-Requisites</b>	Nametag가 생성되어있어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : <i>Nametag</i> 의 지정된 <i>directory</i> 에 접근하여 주기적으로 파일들의 이름을 검색한다. 2. (S) : <i>Nametag</i> 에 맞는 파일 이름이 있다면 파일을 지정된 <i>directory</i> 로 이동하고 true값을 반환한다. . 3. (S) : true가 반환되면 logfile에 동작을 기록한다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	1. Log File이 생성되지 않은 경우 에러 창을 보여준다.

<b>Use Case</b>	<b>Name Tag 생성</b>
<b>Actor</b>	User
<b>Purpose</b>	Name Tag를 생성한다.
<b>Overview</b>	Service Directory상에서 실행되는 파일 및 폴더 이동을 위한 Name Tag를 생성한다. Name Tag의 이름과 해당 폴더위치를 입력하여 시행한다
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 4.2.1, R 5.1 use case : Log File 생성
<b>Pre-Requisites</b>	window-a에서 Name Tag 메뉴가 선택된 상태여야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : window-B에서 Name Tag 생성 기능을 선택한다. 2. (S) : window-C에 Name Tag의 생성 창을 출력한다. 3. (A) : Nametag에 대한 이름 <i>name</i> , 설명 <i>explain</i> , 디렉토리 정보 <i>serviceDirectory</i> 를 입력하고 확인버튼을 누른다. 4. (S) : Nametag의 이름이 기존의 정보들과 중복되는지 확인한다 5. (S) : 입력 받은 조건의 Nametag를 생성한다.

**Alternative Courses of Events**

**Exceptional Courses of Events** 4. (S) : 이름이 중복되었다면, 재입력을 요청한다.

**Use Case****Name Tag 삭제****Actor**

User

**Purpose**

Name Tag를 삭제한다.

**Overview**

- Service Directory상에서 실행되는 Name Tag를 생성한다.
- 삭제 시 확인 창을 보여준다..

**Type**

Primary and Essential

**Cross Reference**

R 4.2.2, R 5.1

use case : Log File 생성

**Pre-Requisites**

window-a에서 Name Tag 메뉴가 선택된 상태여야 한다.  
1개 이상의 Name Tag가 생성되어 있어야 한다.

**Typical Courses of Events**

(A) : Actor, (S) : System

1. (A) : window-b에서 삭제할 Name Tag를 선택, 우클릭해서 Name Tag 삭제 버튼을 누른다.
2. (S) : Name Tag 삭제 여부를 다시 한번 팝업으로 확인한다.
3. (A) : 삭제 할 것인지 말 것인지를 선택한다.
4. (S) : 확인 버튼을 누르면 Name Tag를 삭제하고, 그 결과를 window-b에 다시 출력한다.

**Alternative Courses of Events**

NONE

**Exceptional Courses of Events**

2. 아니오 버튼을 누르거나 확인 창을 끄면 삭제하지 않는다.

**Use Case****Name Tag 수정****Actor**

User

**Purpose**

Name Tag를 수정한다.

**Overview**

- Service Directory상에서 실행되는 Name Tag를 수정한다.
- Name Tag의 이름이나 지정된 폴더위치를 수정할 수 있다.

**Type**

Primary and Essential

**Cross Reference**

R 4.2.3, R 5.1

use case : Log File 생성

**Pre-Requisites**

window-a에서 Name Tag 메뉴가 선택된 상태여야 한다.  
1개 이상의 Name Tag가 생성되어 있어야 한다.

**Typical Courses of Events**

(A) : Actor, (S) : System

1. (A) : window-b에서 수정할 Name Tag를 선택, 우클릭해서 Name Tag 수정 버튼을 누른다.

	<p>2. (A) : window-c에서 수정할 <i>Name Tag</i>의 이름 <i>name</i>, 설명 <i>explain</i>, 디렉토리정보 <i>serviceDirectory</i>를 입력하고 확인 버튼을 누른다..</p> <p>3. (S) : <i>NameTag</i>의 정보를 수정한 후 저장한다.</p>
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	2. 취소 버튼을 누르거나 창을 끄면 정보가 수정되지 않는다.

<b>Use Case</b>	<b>Name Tag 목록보기</b>
<b>Actor</b>	User
<b>Purpose</b>	Name Tag의 목록을 보여준다.
<b>Overview</b>	<p>- Service Directory상에서 실행되는 모든 Name Tag의 목록을 보여준다.</p> <p>- Name Tag의 이름과 해당하는 폴더 위치를 보여준다.</p>
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 4.2.4
<b>Pre-Requisites</b>	window-a에서 Name Tag 메뉴가 선택된 상태여야 한다.
<b>Typical Courses of Events</b>	<p>(A) : Actor, (S) : System</p> <p>1. (A) : window-b에서 Name Tag 목록보기를 선택한다.</p> <p>2. (S) : window-b에 Name Tag 목록을 보여준다.</p>
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	1. Name Tag가 없을 경우 Name Tag가 없다는 내용을 보여준다.

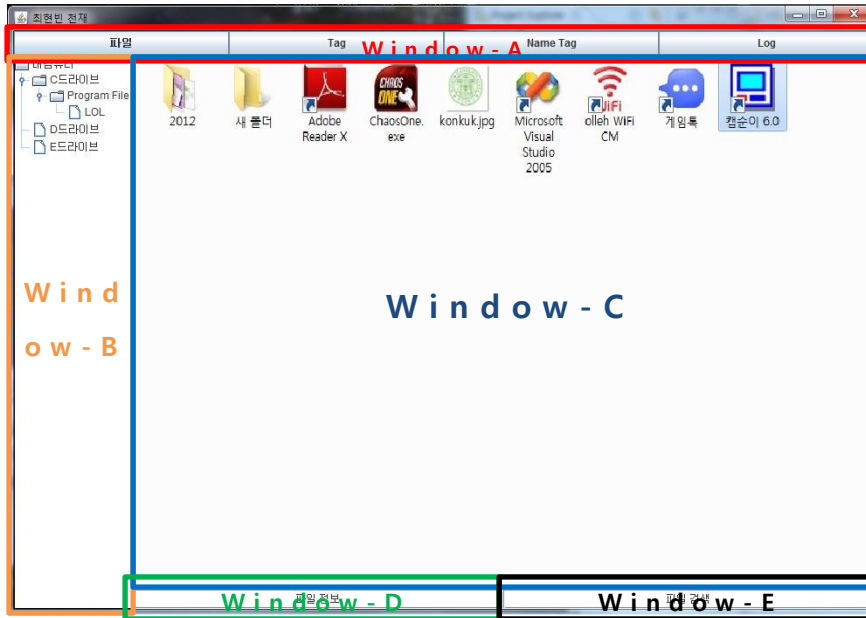
<b>Use Case</b>	<b>Log File 생성</b>
<b>Actor</b>	User
<b>Purpose</b>	TDMS상의 Tag의 관련 동작을 Log File에 기록한다
<b>Overview</b>	파일 삭제, 파일 수정, Tag 생성, Tag 삭제, Tag 수정, Tag 지정, Tag 해제, Tag 의존성 부여, Tag 의존성 삭제, Name Tag 생성, Name Tag 삭제, Name Tag 수정의 동작을 Log File에 기록한다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	<p>R 1.2.1, R 1.3, R 2.1, R 2.2, R 2.3, R 2.5.1, R 2.5.2, R 2.6.1, R 2.6.2, R 4.2.1, R 4.2.2, R 4.2.3, R5.1</p> <p>Use Case : 파일 삭제, 파일 수정, Tag 생성, Tag 삭제, Tag 수정, Tag 지정, Tag 해제, Tag 의존성 부여, Tag 의존성 삭제, Name Tag 생성, Name Tag 삭제, Name Tag 수정</p>
<b>Pre-Requisites</b>	TDMS상에서 사용자가 조회 또는 보기 등을 제외한 다른 동작을 수행하여야 한다.

<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S) : 지정된 기능에 대해서 Log File의 생성을 요구한다 2. (S) : 해당 기능에 대한 동작에 대한 내용 <i>string log</i> 을 Log File에 추가한다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	NONE

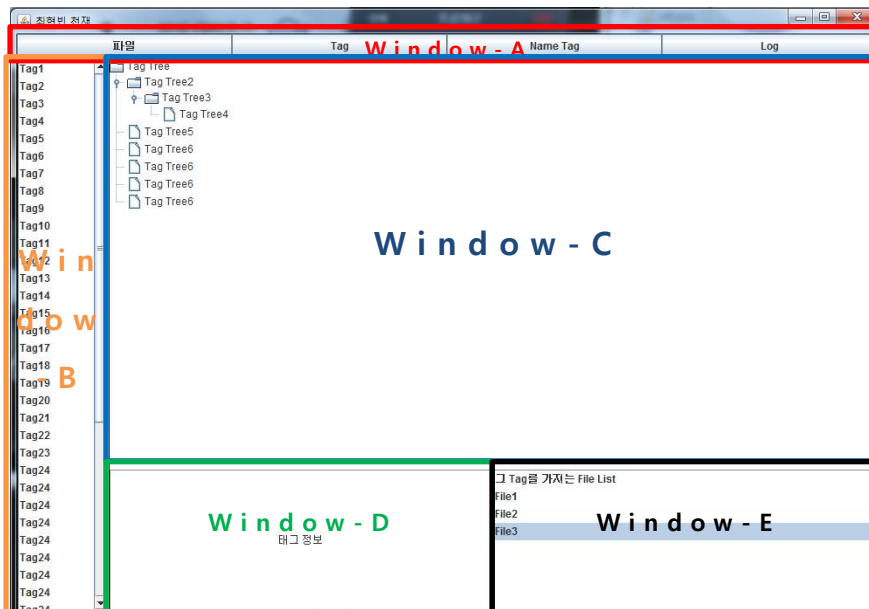
Use Case	Log File 보기
<b>Actor</b>	User
<b>Purpose</b>	생성된 Log File을 확인한다.
<b>Overview</b>	- 생성된 Log File을 확인할 수 있다.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	R 5.2
<b>Pre-Requisites</b>	window-a에서 Log 메뉴가 선택된 상태여야 한다. TDMS상에서 사용자가 검색 또는 보기 등을 제외한 다른 동작을 수행하여야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : window-b에서 Log File 보기 메뉴를 선택한다. 2. (S) : 생성된 Log File들을 window-c 창에 보여준다.
<b>Alternative Courses of Events</b>	NONE
<b>Exceptional Courses of Events</b>	1. Log File이 생성되지 않은 경우 에러 창을 보여준다.

## Stage 2142. Define Reports, UI, and Storyboards

### - User Interface (File)

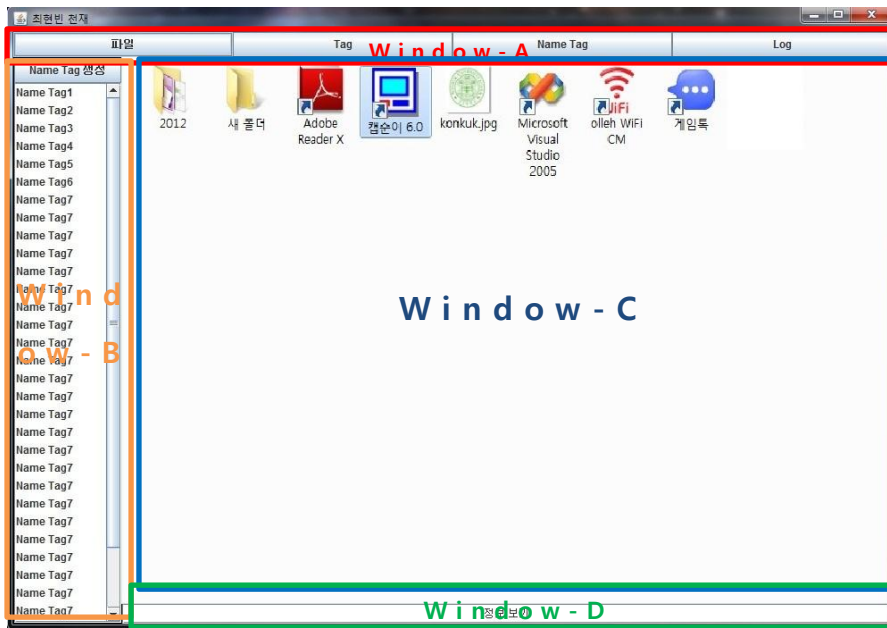


### - User Interface (Tag)

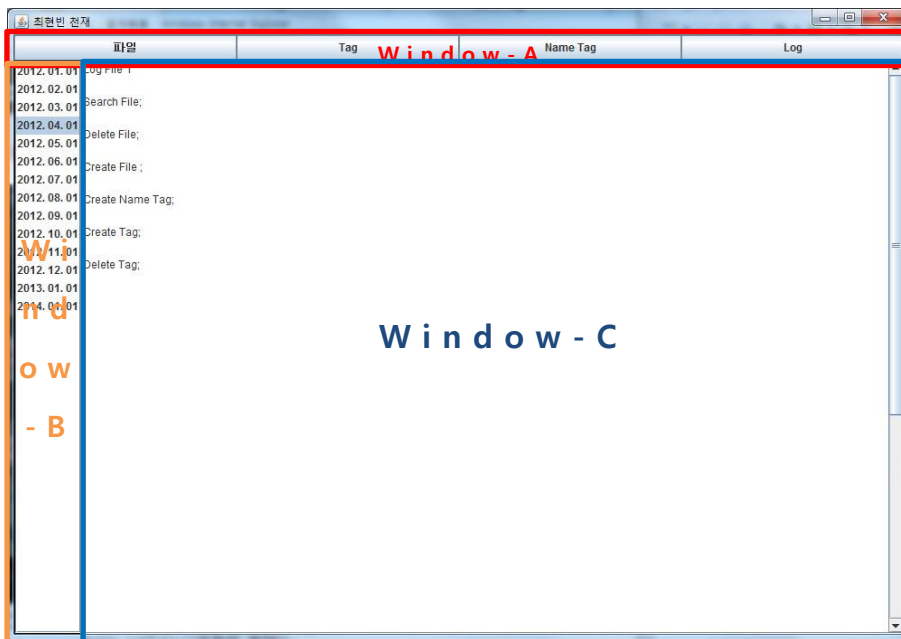




- User Interface (Name Tag)



- User Interface (Log)



### Stage 2143. Refine System Architecture

TDM5

#### Interface

```

- log: string
- load: enum
- job: enum

+ removeFile(id: int): void
+ showDependency(): void
+ throw(input: boolean)
+ executeFile(id: int): void
+ renameFile(id: int): void
+ showPopUp(message: string)
+ throw(name: string)
+ searchFile(): void
+ showSearchWindow(message: string): void
+ showSearchResult(): void
+ searchTag(): void
+ requestAppointTag(fileId: int, : void)
+ showTagList()
+ throw(tagId: int)
+ requestClearTagOfFile(fileId: int)
+ requestRelateBetweenTag()
+ showHierarchy()
+ showFileInfo(fileName: string): void
+ throw(parentId: int, childId: int)
+ deleteTag(tagId: int)
+ showTagInfo(tag: Tag): void
+ showFileList(list: List): void
+ showFileInforWindow()
+ requestTagMakeView()
+ showTagMakeViewWindowC()
+ showDependencyListWindowE()
+ makeTag(name: string, type: Enum, color: Enum, dependency: Dependency): void
+ modifyTag(name: string): void
+ showModifyView(tag: Tag)
+ requestDeleteNametag(name: string): void
+ requestViewNameTag(): void
+ showNametagListWindowC()
+ requestModifyNametag(name: string): void
+ showNametagModifyView(nametag: Nametag): void
+ modifyNametag(name: string, location: string, explain: string): void
+ requestViewLogFile(): void
+ showLogFileWindowC()
+ requestDeleteRelation(depId: int): void
+ showDependencyInfoWindowD(): void
+ requestViewHierarchy(tagname: string)
+ showHierarchyWindowC(list: List): void
+ requestViewRecentUsedTag()
+ showRecentUsedTag()
+ requestCreateNametag()
+ showViewWindowC()
+ createNametag(name: string, directory: string)
    
```

#### TagMgr

```

- tag: list<Tag>

+ searchTag(name: string): List
+ getList(): List
+ removeTag(tagId: int): Tag
+ setParent(tagId: int, parentId: int, : void)
+ setChild(tagId: int, childId: int, : void)
+ getTag(tagId: int): Tag
+ getDependencyList(): void
+ isDuplicateName(name: string): boolean
+ makeTag(name: string, type: Enum, color: Enum, dependency: Dependency): void
+ modifyTag(id: int, name: string, type: Enum, color: Enum, dependency: Dependency)
+ deleteTagDependency(depId: int): boolean
+ getRecentUsedTags(): List
+ writeRecentTag(): void
    
```

#### Tag

```

- name: string
- explain: string
- id: int
- type: enum
- color: enum
- usingCount: int
- parentId: int
- childId: int
    
```

#### NametagMgr

```

- nametag: list<Nametag>

+ deleteNametag(name: string): Nametag
+ getNametagList(): List
+ getNametag(name: string): Nametag
+ isDuplicate(name: string): boolean
+ modifyNametag(id: int, name: string, location: string, explain: string): void
+ createNametag(name: string, directory: string): void
+ getList(): list <NameTag>
    
```

#### DependencyMgr

```

- dependency: list<Dependency>

+ checkDependency(id: int): Dependency
+ addDependency(top: int, bottom: int, void)
+ getDependency(depId): Dependency
+ deleteDependency(depId: int): Dependency
+ getDependencyList(tagId): List
+ requestCheckDependency(tagId: int): boolean
    
```

#### Dependency

```

- topId: int
- bottomId: int
- id: int
    
```

#### Nametag

```

- name: string
- explain: string
- id: int
- serviceDirectory
    
```

#### FileMgr

```

- file: list<File>

+ removeFile(id: int): void
+ executeFile(id: int): void
+ checkDuplication(id: int, name: string): void
+ searchFile(name: string): List
+ appointTag(tagId: int, : void)
+ isTagged(id: int): boolean
+ clearTag(: void)
+ getFileList(tagId: int): List
+ deleteFiles(tagId: int): void
+ getFile(fileName: string): File
    
```

#### File

```

- id: int
- information: string
- binder
- tagId: int
- isTagged: boolean
    
```

#### Service

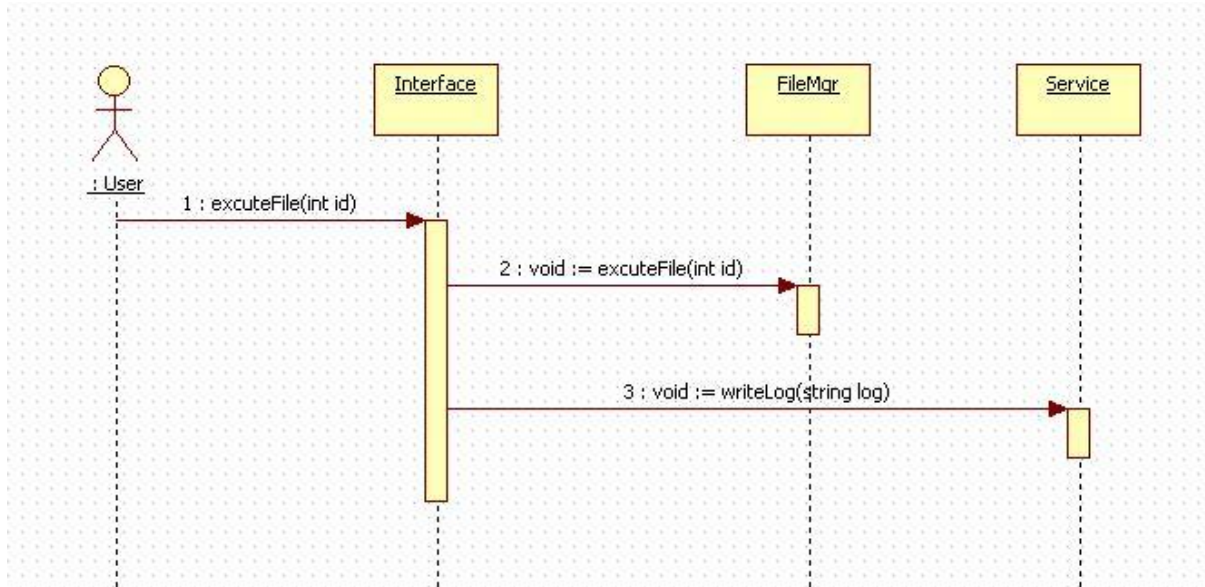
```

- log: list< string >

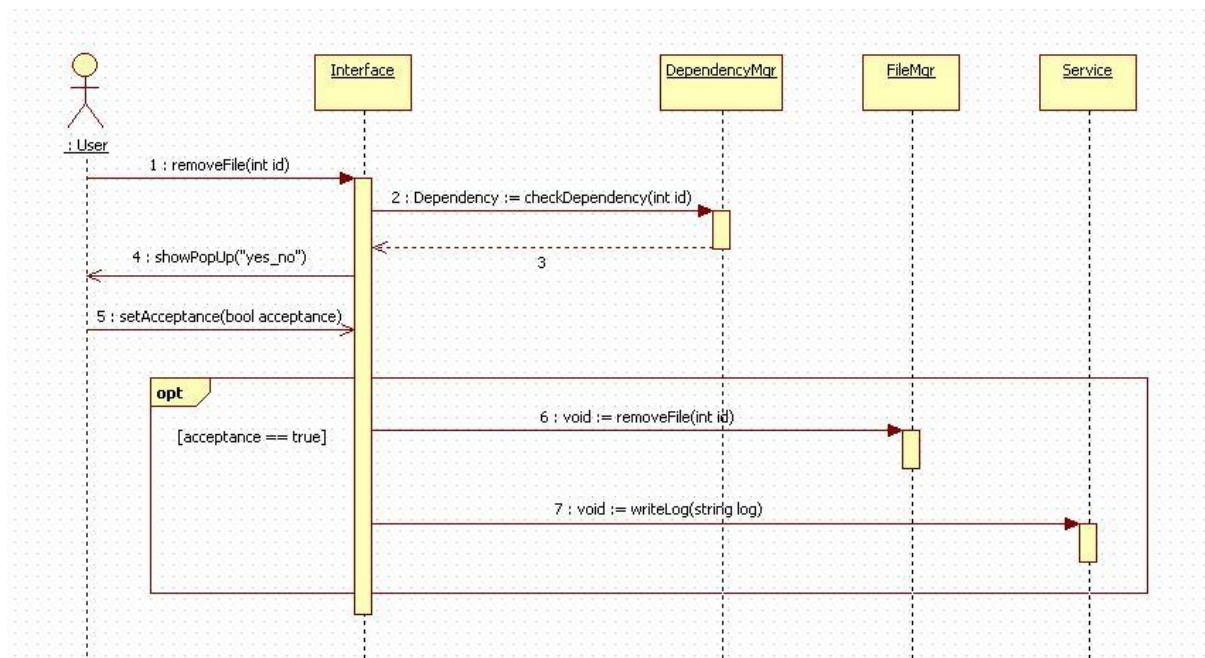
+ writeLog(log: string): void
+ getLogFile(): string
+ serviceDirectory(): void
    
```

## Stage 2144. Define Interaction Diagrams

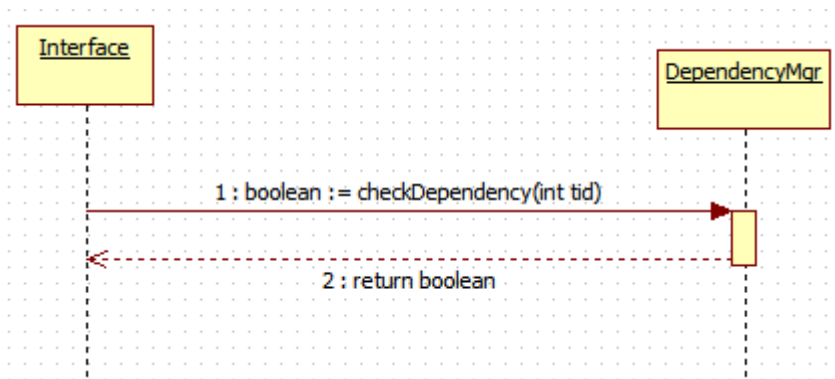
### - executeFile()



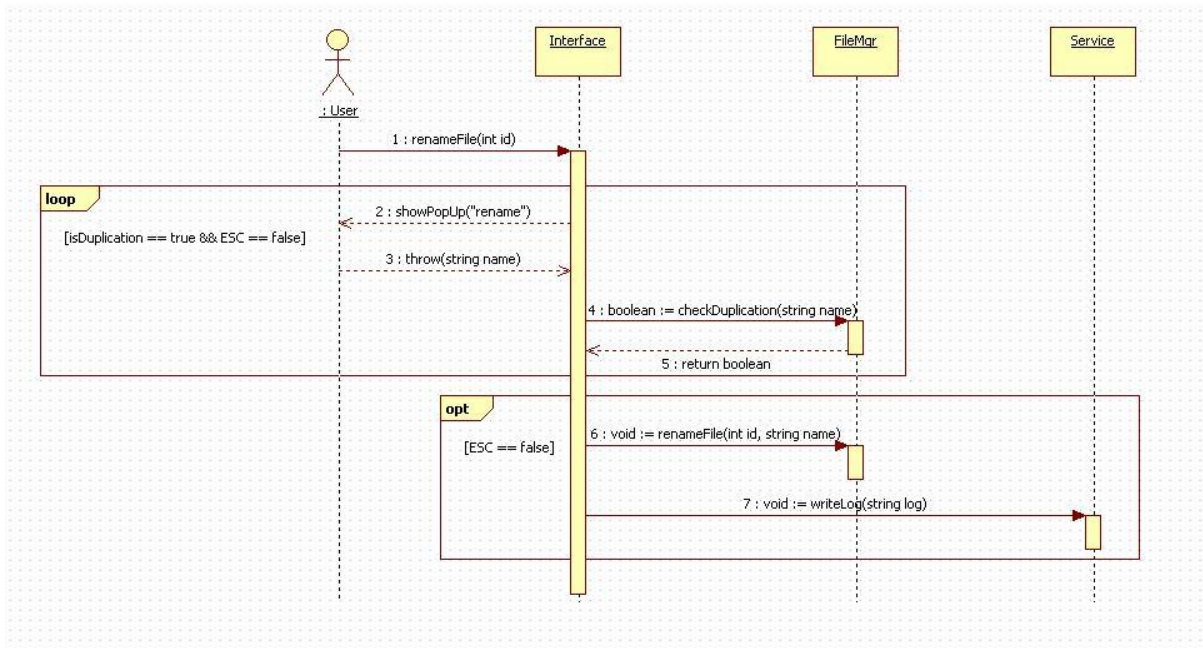
### - removeFile()



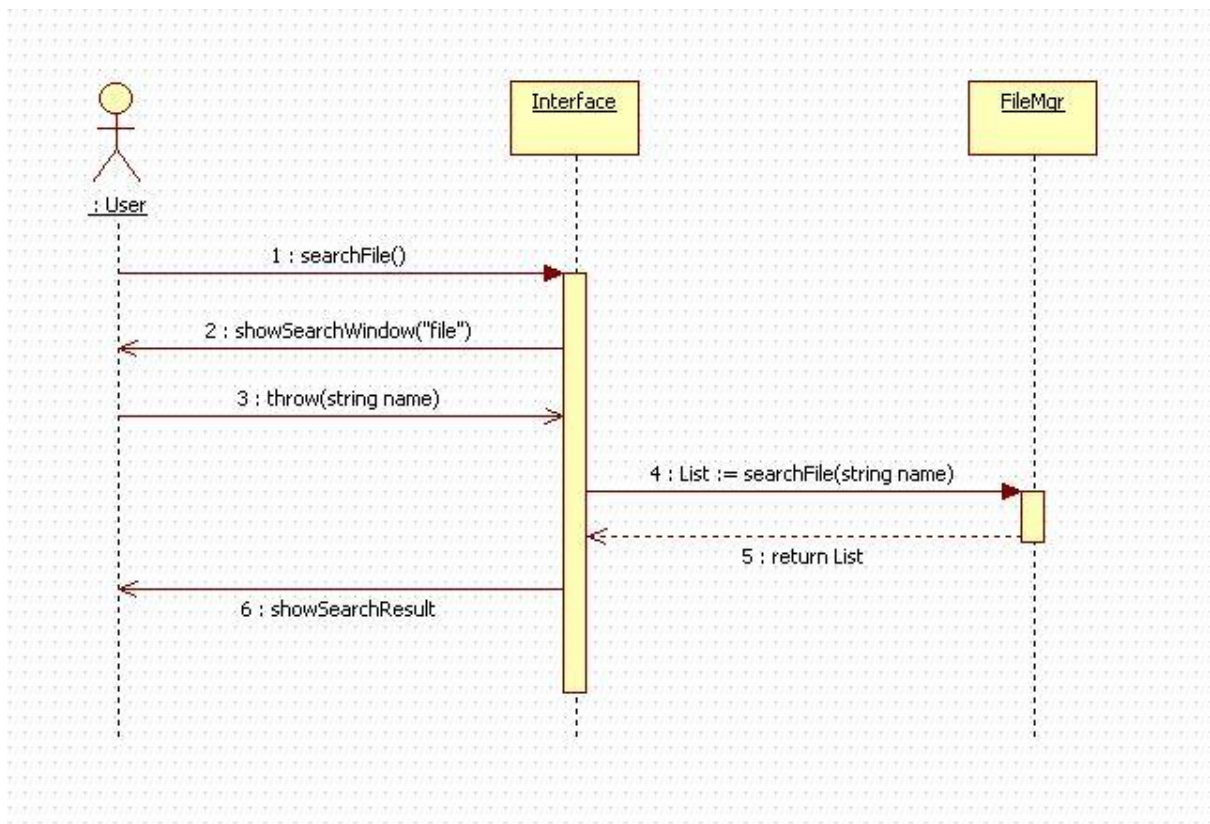
**- checkDependency()**



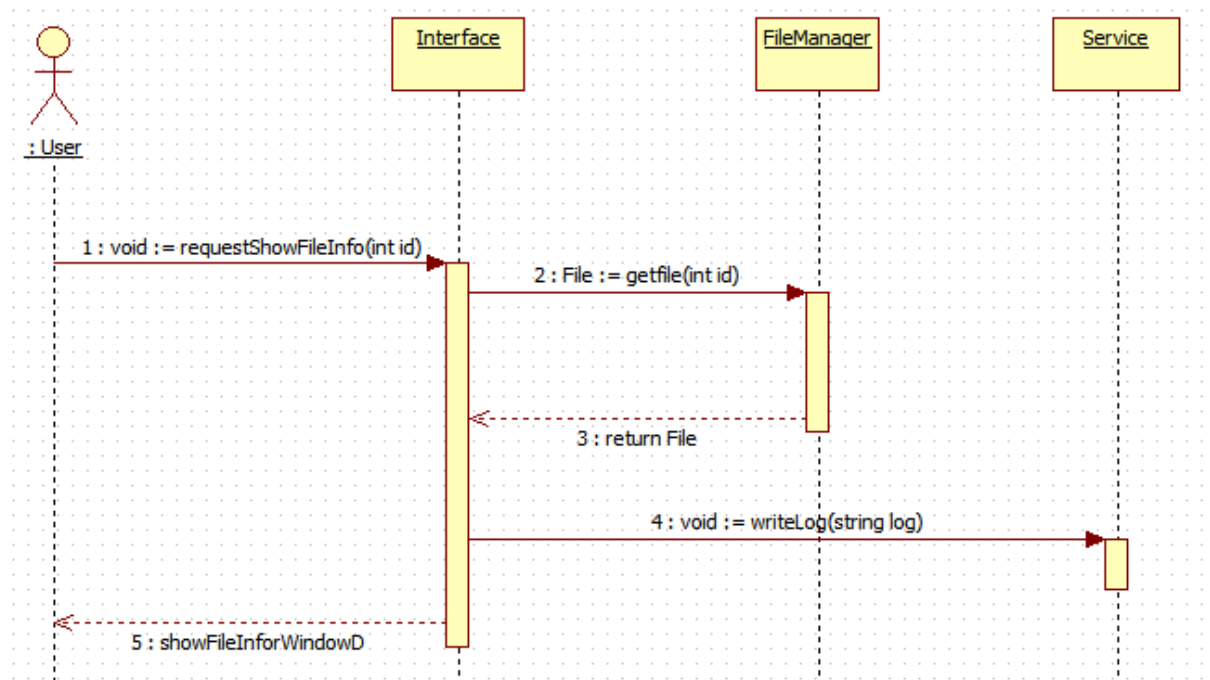
**- renameFile()**



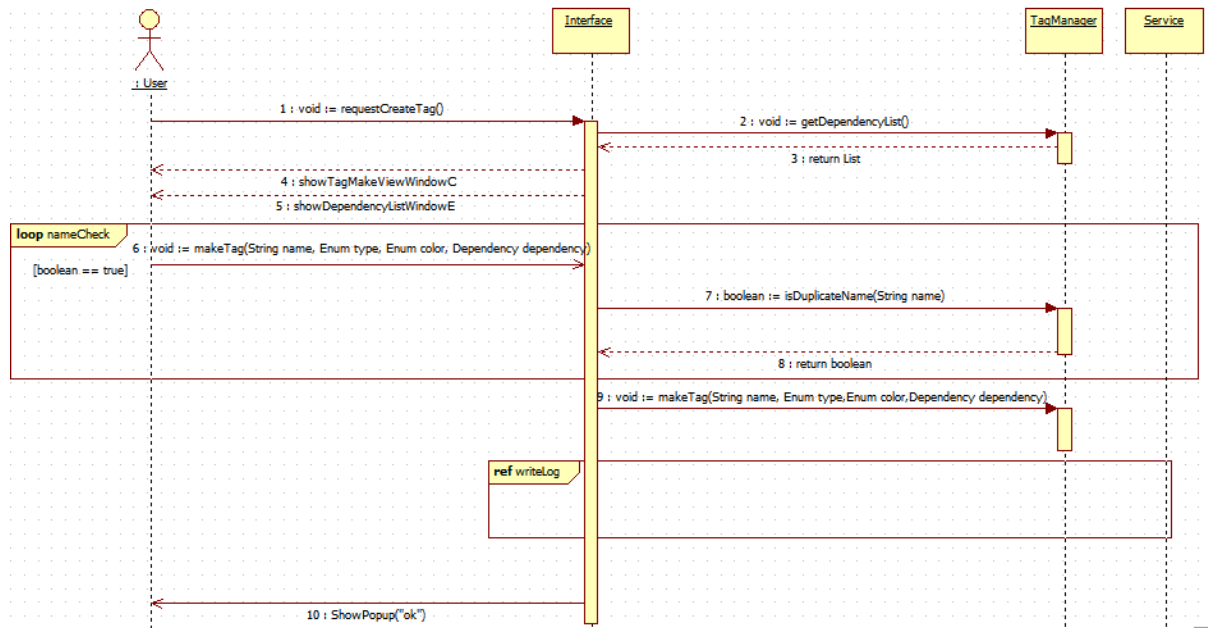
**- searchFile()**



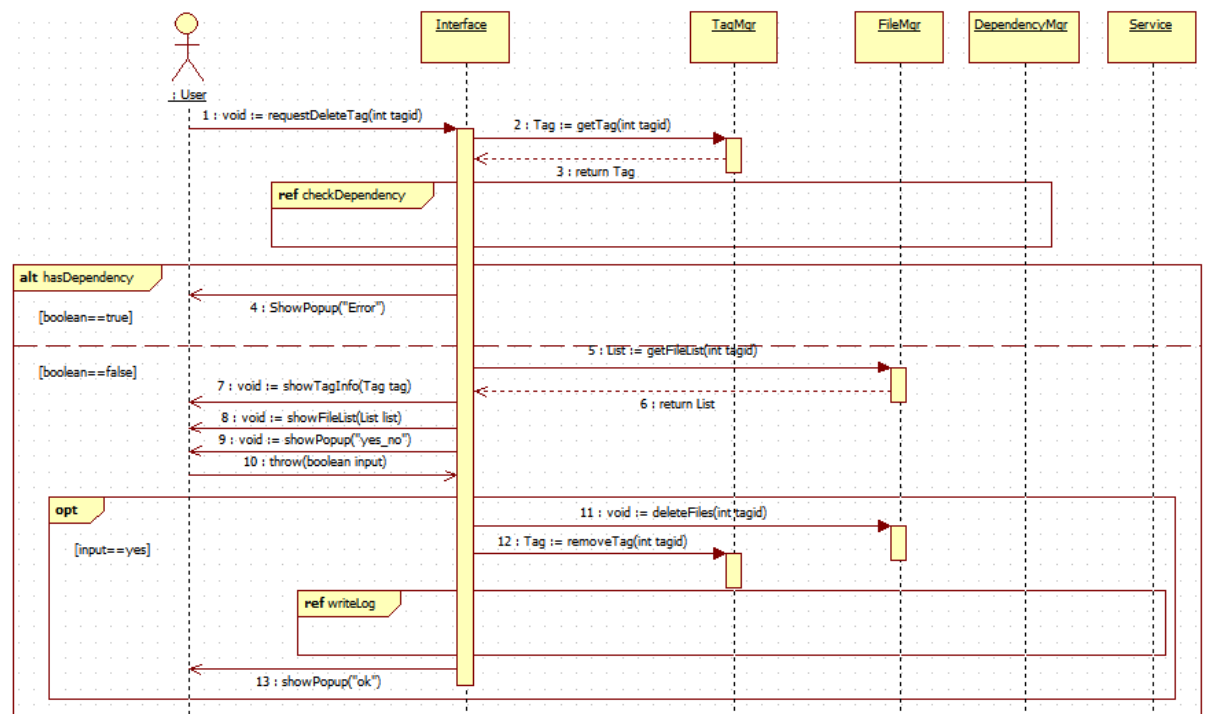
**- requestShowFileInfo()**



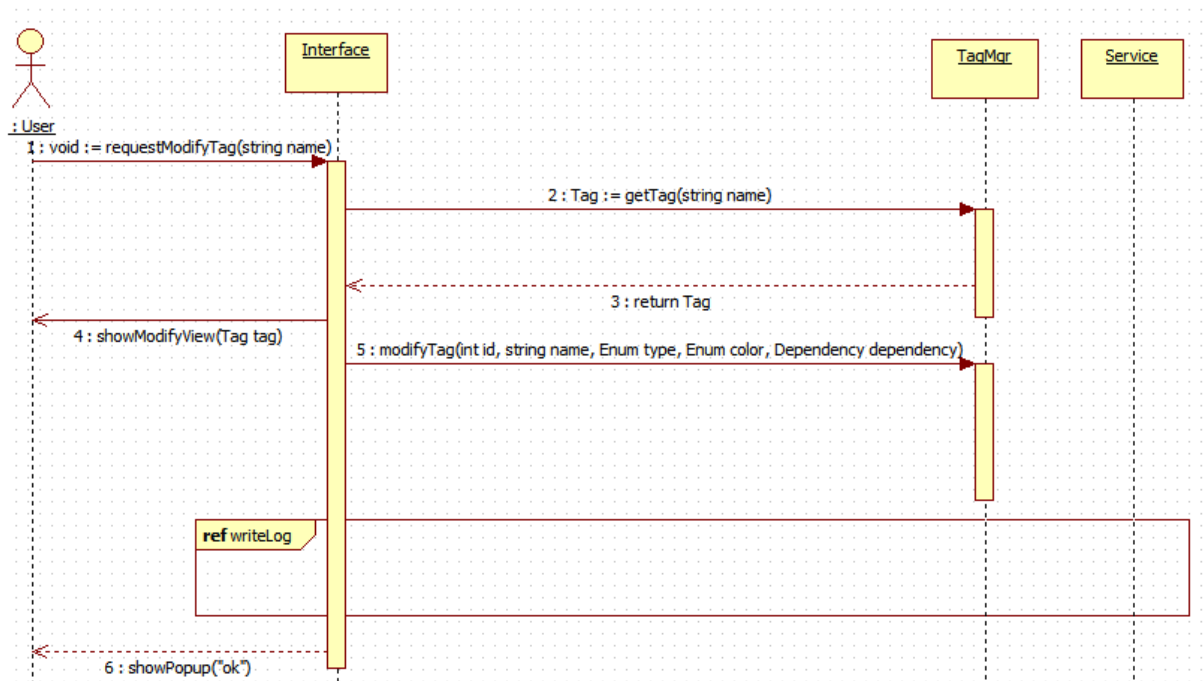
**- requestCreateTag()**



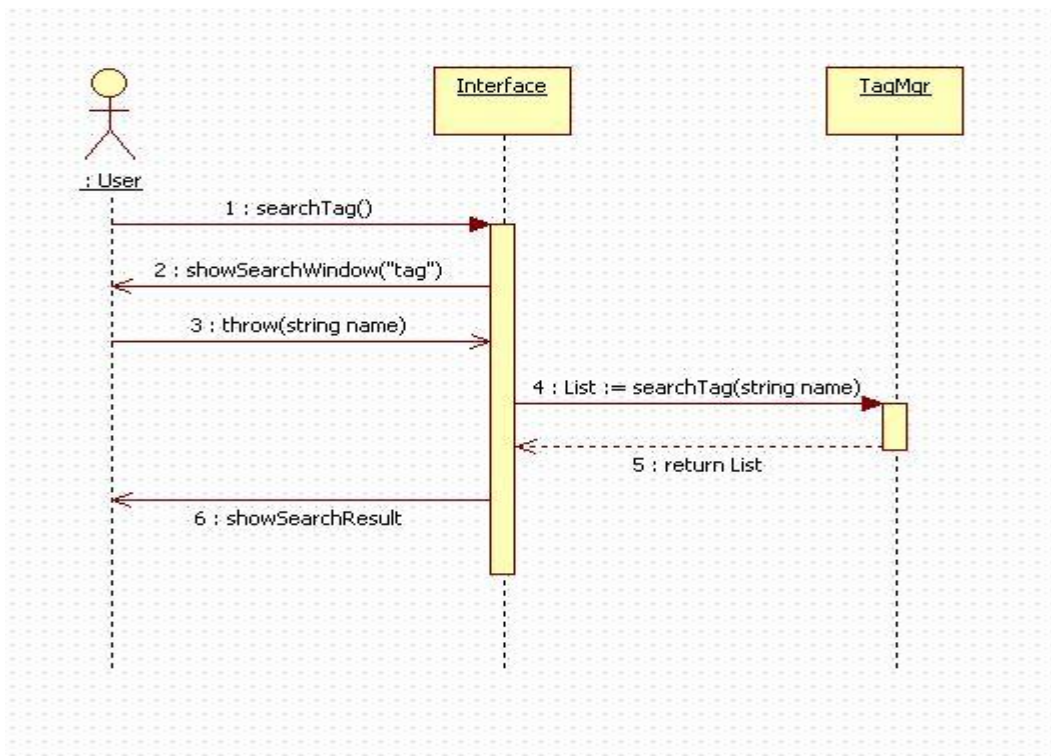
**- requestDeleteTag()**



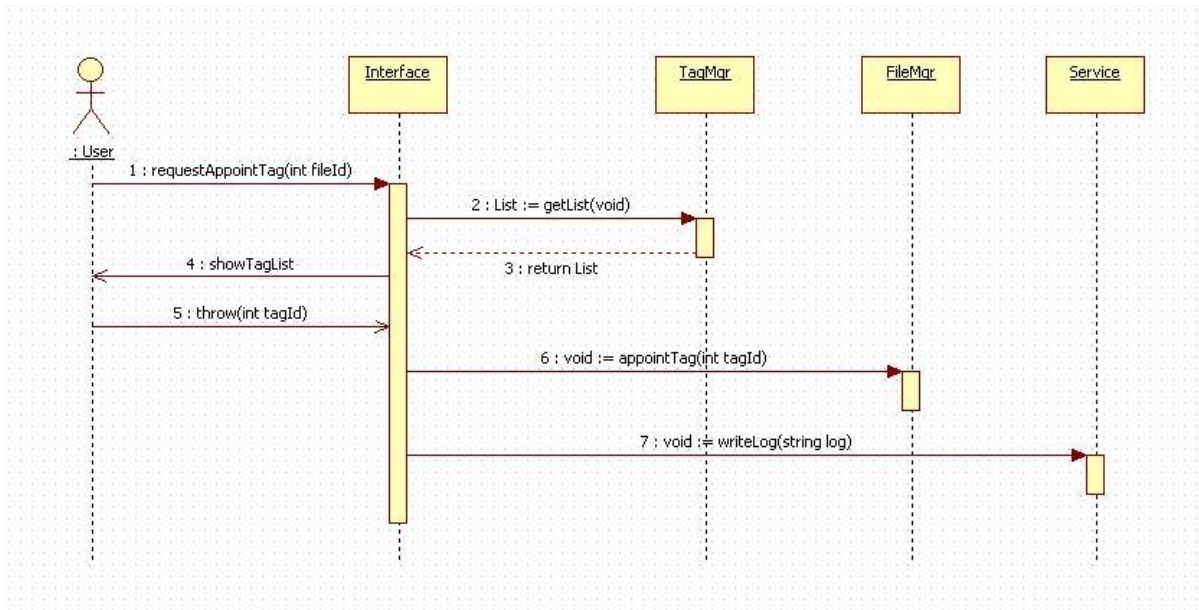
**- requestModifyTag()**



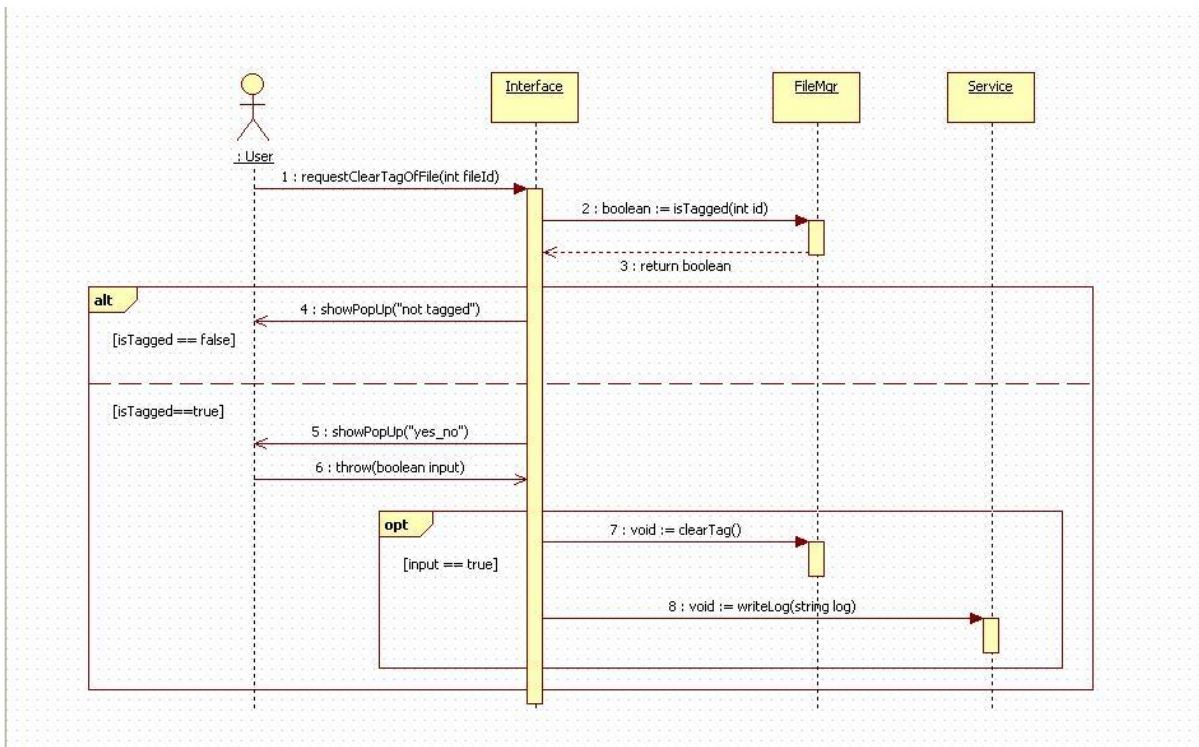
**- searchTag()**



**- requestAppointTag()**

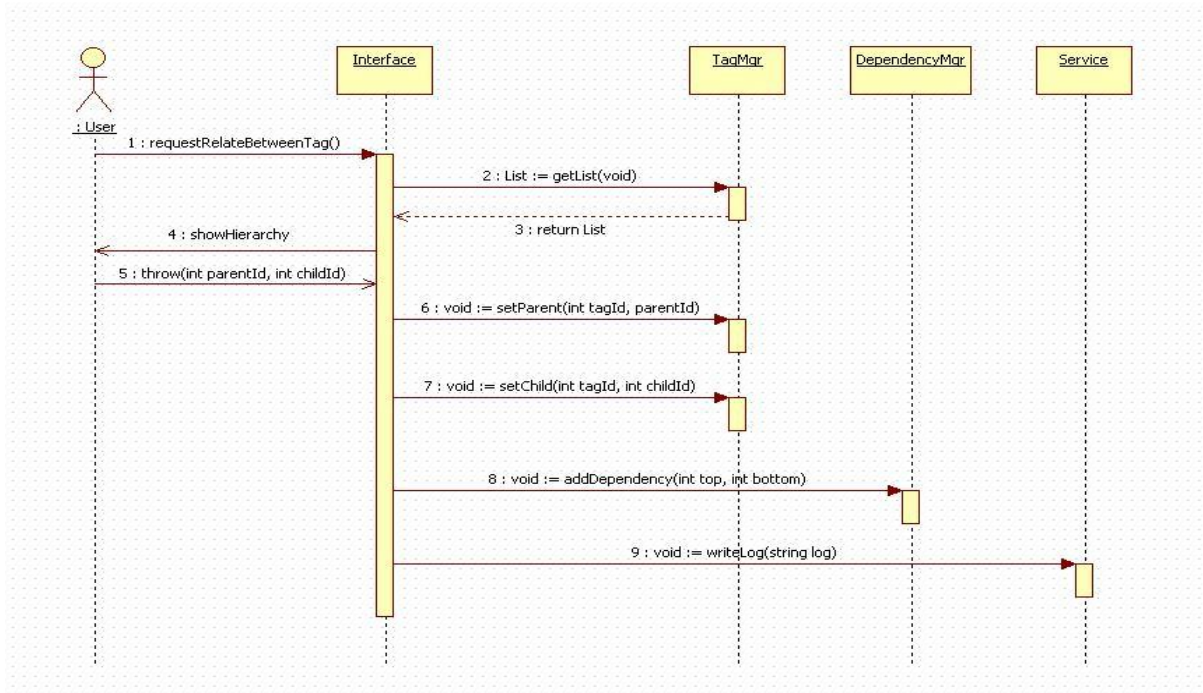


**- requestClearTagOfFile()**

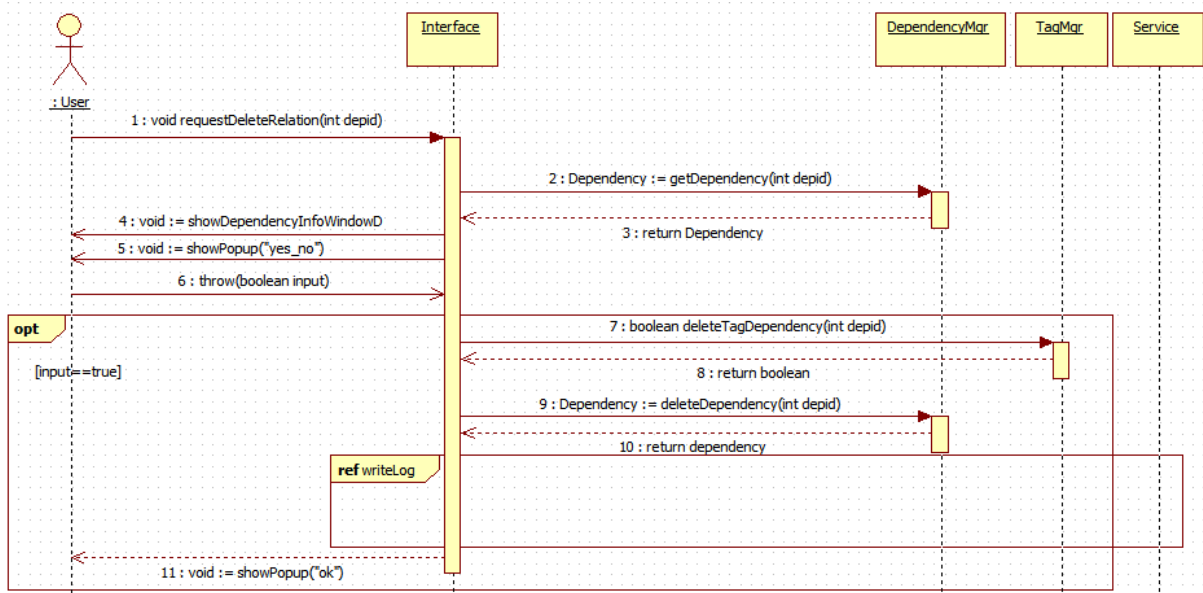




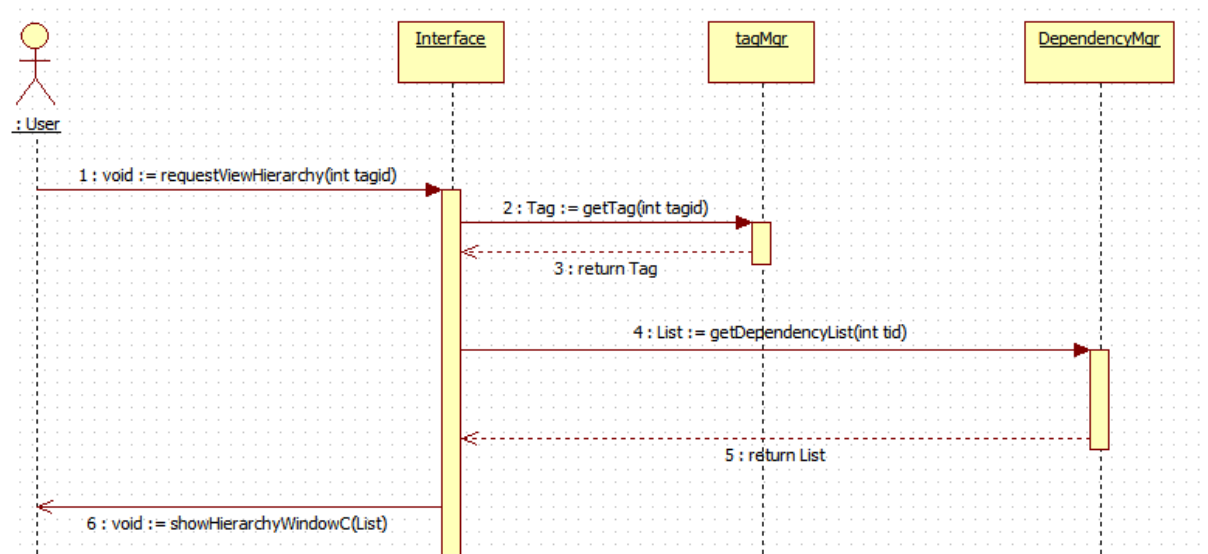
**- requestRelateBetweenTag()**



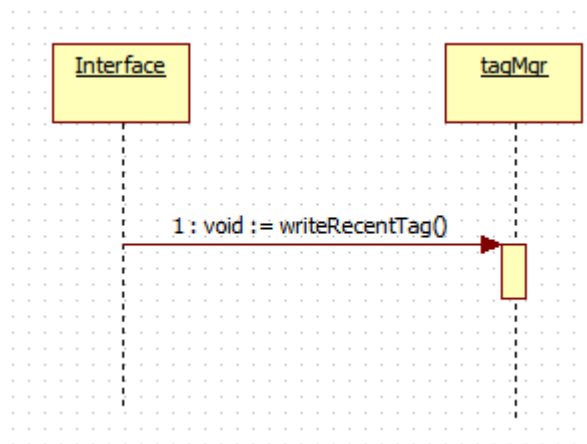
**- requestDeleteRelation()**



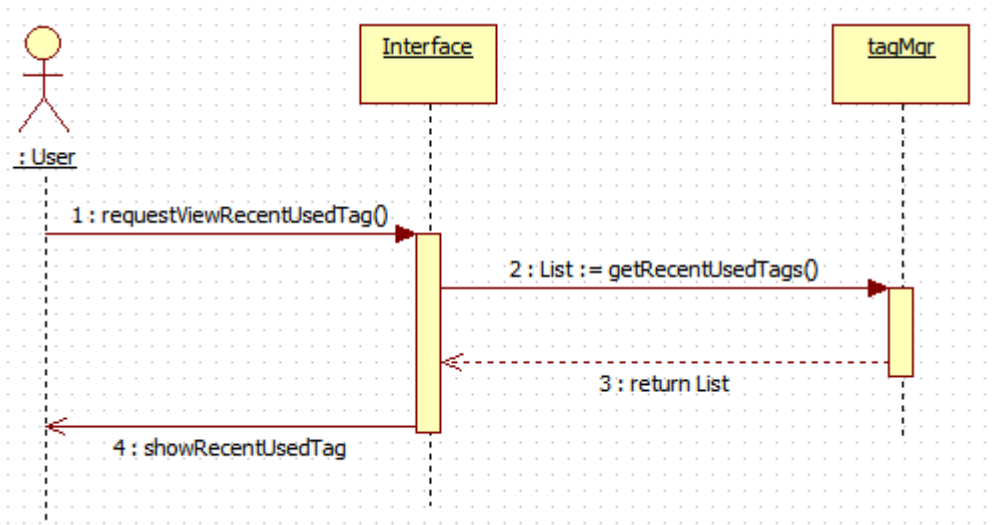
**- requestViewHierarchy()**



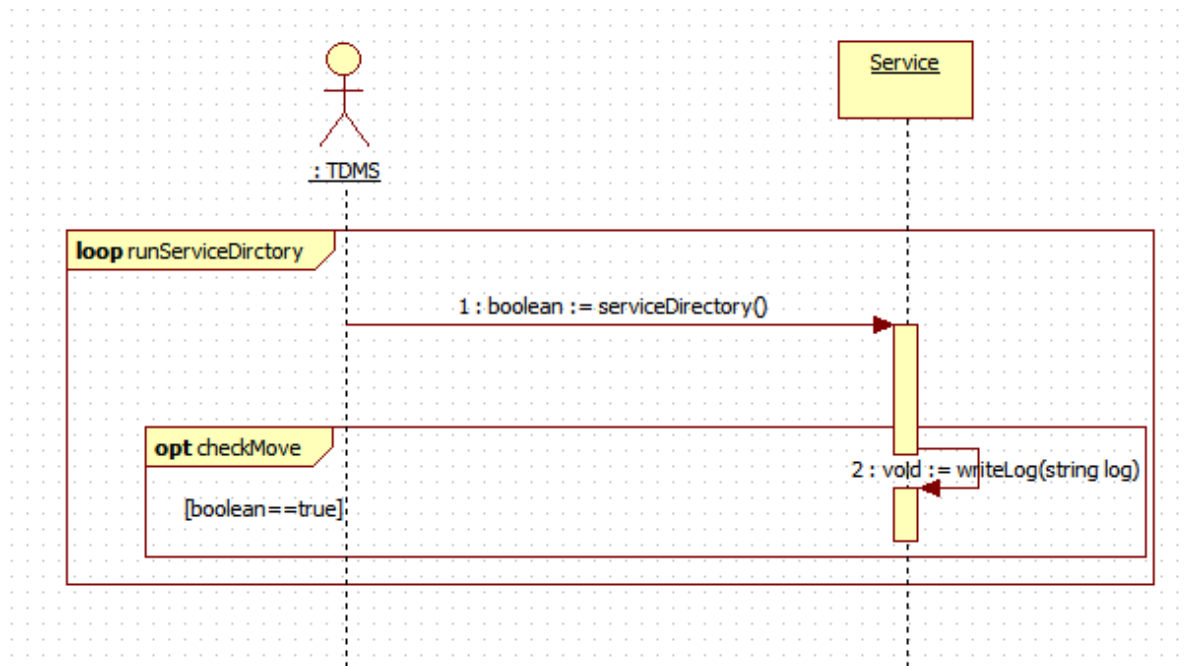
**- writeRecentTag()**



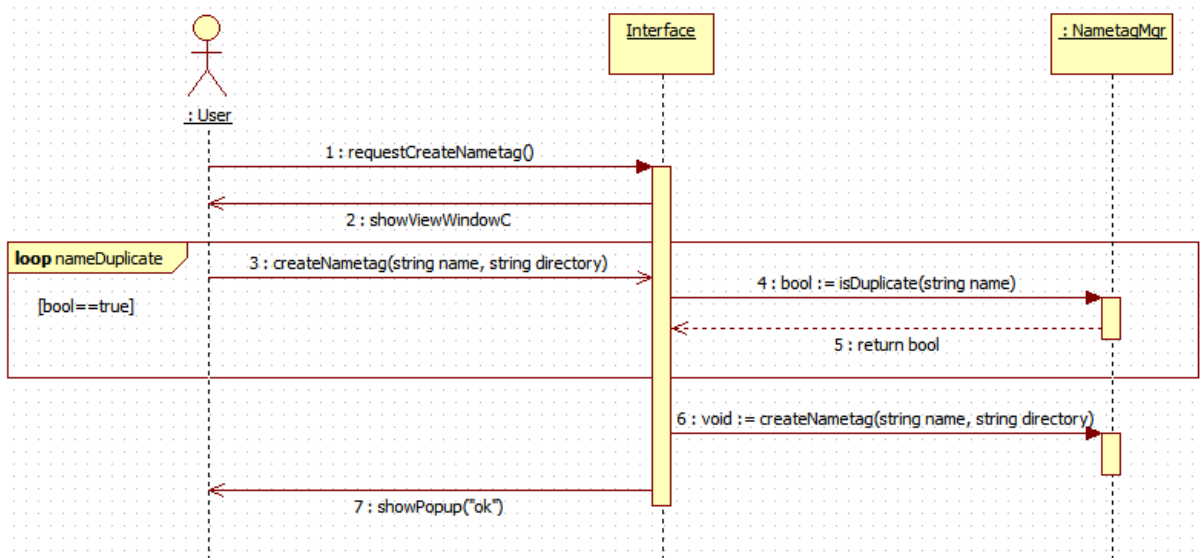
**- requestViewRecentUsedTag()**



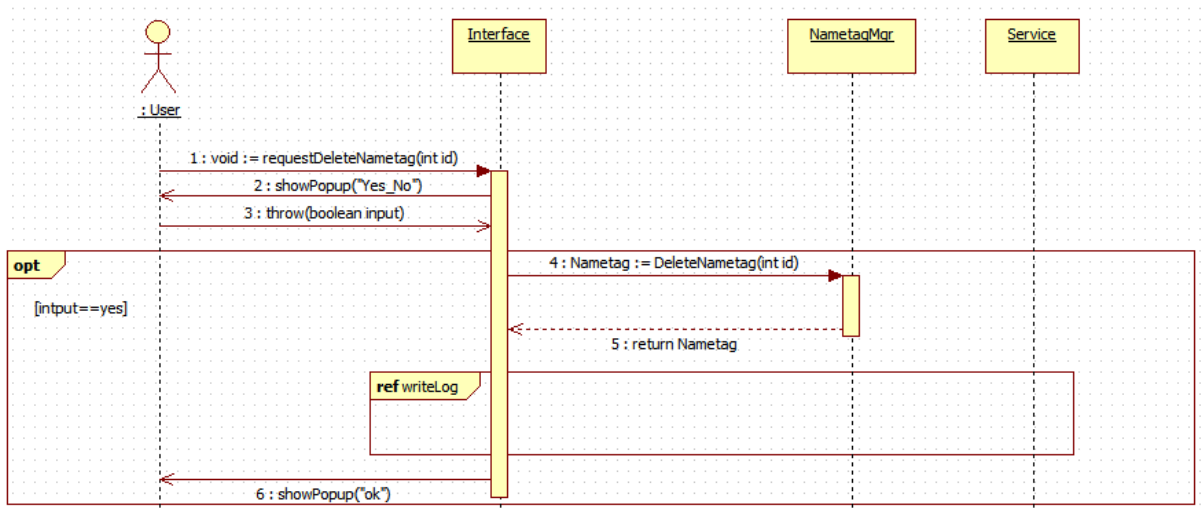
**- serviceDirectory()**



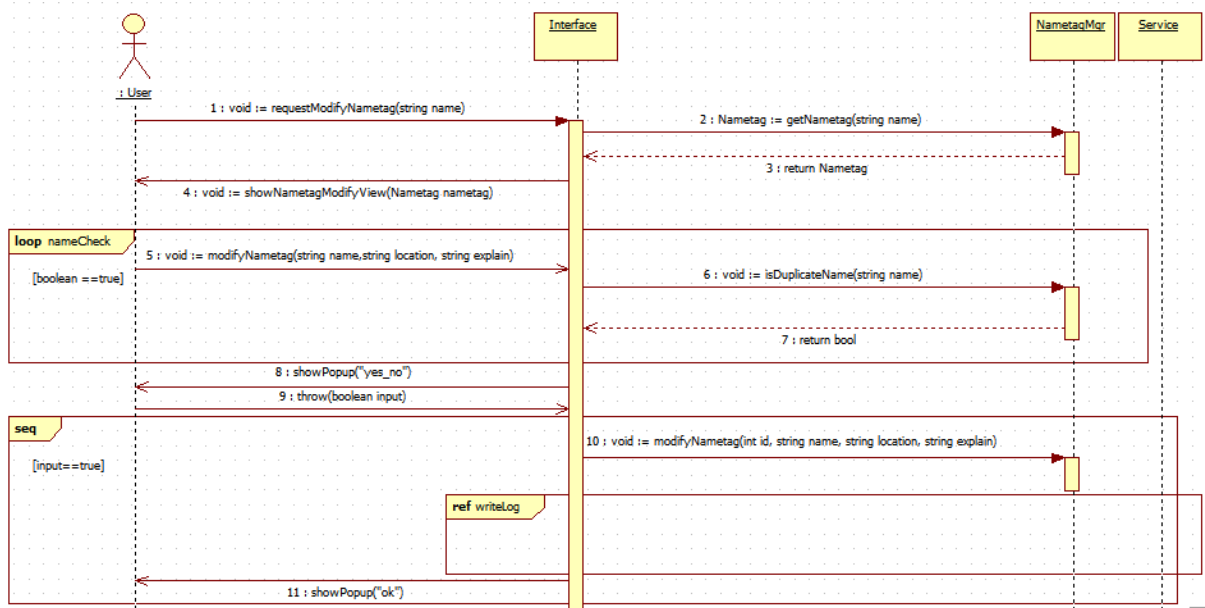
**- requestCreateNameTag()**



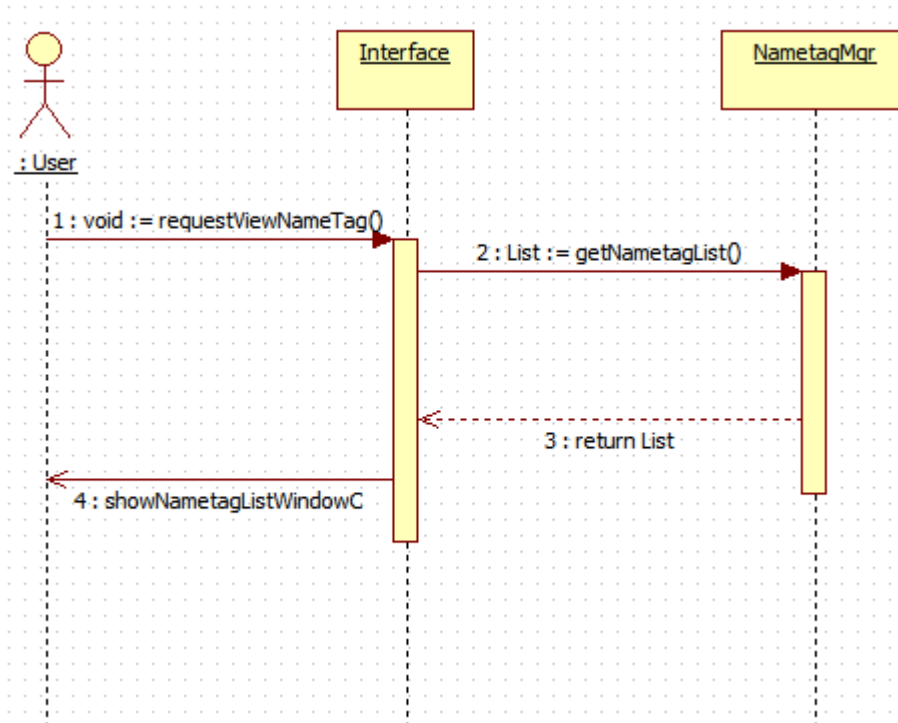
**- requestDeleteNameTag()**



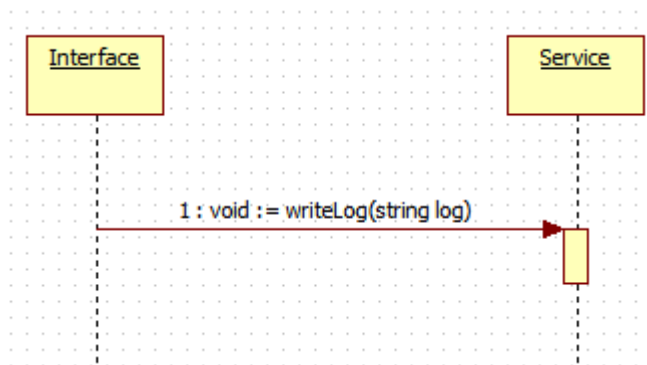
**- requestModifyNameTag()**



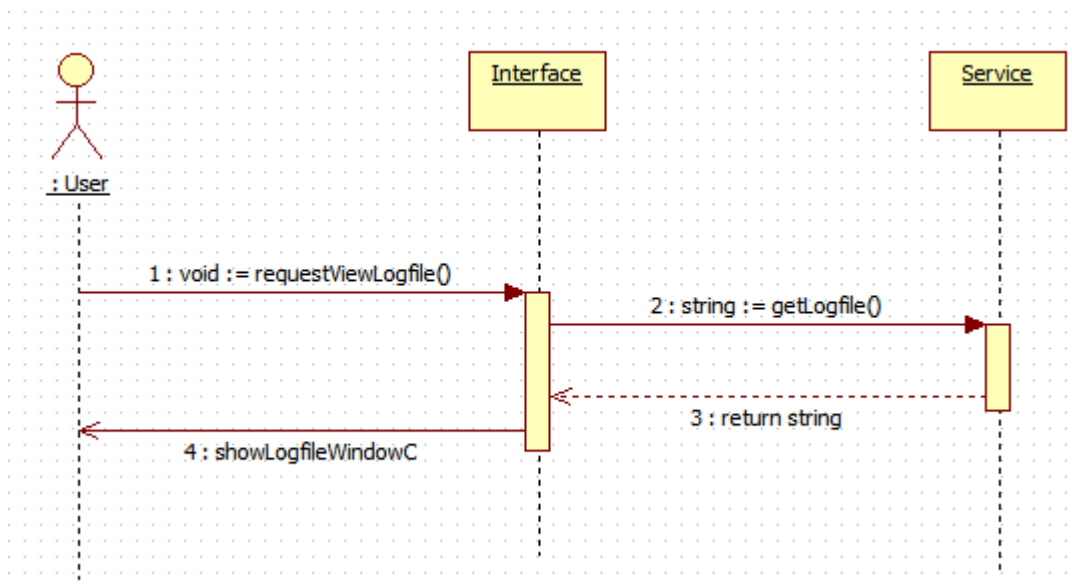
**- requestViewNameTag()**



**- writeLog()**



**- requestViewLogFile()**



**Stage 2145. Define Design Class Diagrams**

