

OSP Stage 2140

- 1st Cycle v_120506

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, R4.2.4, R 5.1 Use Case : 파일 검색, 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 선택한다. 2. (S) : window-b의 버튼을 활성화 시킨다. 3. (A) : window-b의 파일 Excute버튼을 클릭한다. 4. (S) : 선택한 <i>file</i> 을 실행한다. 5. (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 사용자 에게 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 선택한다. 2. (S) : window-b의 버튼을 활성화 시킨다. 3. (A) : window-b의 Delete 버튼을 클릭한다. 4. (S) : <i>file</i> 의 <i>dependency</i> 를 체크한다.

	<p>5. (S) : 삭제 여부를 확인하는 팝업 창을 보여준다.</p> <p>6. (A) : 팝업 창의 삭제여부를 결정한다.</p> <p>7. (S) : <i>Logfile</i>을 생성한다.</p> <p>8. (S) : 선택한 <i>file</i>을 삭제한다.</p>
Alternative Courses of Events	NONE
Exceptional Courses of Events	<p>4. 파일에 의존성이 있으면 메시지를 보여주고 삭제를 취소한다.</p> <p>6. 삭제 요청을 취소할 시, 이후 실행 없이 원상태로 돌아간다.</p>

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

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

	<ol style="list-style-type: none"> 1. (A) : window-c에서 수정하고자 하는 file을 선택한다. 2. (S) : window-b의 버튼을 활성화 시킨다. 3. (A) : window-b의 Modify 버튼을 클릭한다. 4. (S) : 파일 수정을 위한 팝업 창을 보여준다. 5 (A) : 팝업 창에 새로운 file의 name을 입력한다 6. (S) : 입력 받은 name이 중복되는지 확인한다. 7. (S) : 선택한 file의 name을 바꿔준다. 8. (S) : Logfile을 생성한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	6. 만약 중복된 이름이면 메시지를 출력하고 재입력을 요청한다.

Use Case	파일 검색
Actor	User
Purpose	파일 및 폴더를 이름을 기반으로 찾는다.
Overview	사용자가 파일 및 폴더를 검색하고, 검색 시행 시 여러 가지 옵션을 추가할 수 있다.
Type	Primary and Essential
Cross Reference	R 1.1, R 1.4 Use Case : 파일 실행
Pre-Requisites	window-a에서 파일 메뉴가 선택된 상태여야 한다.
Typical Courses of Events	(A) : Actor, (S) : System <ol style="list-style-type: none"> 1. (A) : window-e에서 파일 검색 찾고자 하는 파일 이름인 name과 태그된 파일 중 찾을 것인지에 대한 여부 bTagged 선택하고, 검색 위치가 되는 location을 window-b에서 선택한다. 2. (A) : 검색 버튼을 누른다.. 3. (S) : 태그 된 파일을 찾는다면 Tagged File List에서 name을 찾고, 태그 된 파일이 아니면 입력된 Location의 sub directory에서 fileName을 포함하고 있는 파일을 찾는다. 4. (S) : 검색 결과를 window-c에 출력해준다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

Use Case	파일 정보 보기
Actor	User
Purpose	파일 기본적인 정보를 확인할 수 있다.
Overview	이름, 수정된 날짜, 지정된 Tag, Tag가 지정된 날짜 등의 정보를 확인할 수 있다.
Type	Primary and Essential
Cross Reference	R 1.5
Pre-Requisites	window-a에서 파일 메뉴가 선택된 상태여야 한다.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : window-c에서 정보를 보고자 하는 파일을 선택한다.. 2. (S) : 파일 정보를 window-d에 출력한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

Use Case	Tag 생성
Actor	User
Purpose	새로운 Tag를 만든다.
Overview	파일 및 폴더에 지정할 Tag를 만든다.
Type	Primary and Essential
Cross Reference	R 2.1, R 5.1 use case : Log File 생성
Pre-Requisites	window-a에서 Tag 메뉴가 선택된 상태여야 한다.
Typical Courses of Events	(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.
Alternative Courses of Events	NONE
Exceptional Courses of Events	4. 중복된 이름이 존재할 시, 재입력을 요청한다.

Use Case	Tag 삭제
Actor	User
Purpose	기존의 Tag를 삭제한다.
Overview	지정한 Tag를 삭제한다. 삭제할 Tag는 의존성이 없어야 한다.
Type	Primary and Essential
Cross Reference	R 1.2.2, R 2.2, R 2.5.2., R 5.1 use case : 의존성 검사, Tag 의존성 삭제, Log File 생성
Pre-Requisites	window-a에서 Tag 메뉴가 선택된 상태여야 한다. 삭제하고자 하는 Tag가 존재해야 한다.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : window-b의 태그 목록 중에서 삭제할 tag를 선택한다. 2. (S) : window-b의 버튼을 활성화한다. 3. (A) : window-b의 Delete버튼을 클릭한다. 4. (S) : Tag의 의존성 검사를 한다. 5. (S) : Tag 삭제 여부를 다시 한번 팝업으로 확인한다. 6. (A) : 삭제 여부를 선택한다. 7. (S) : 해당 Tag를 삭제 하겠다고 선택하였으면 tag를 삭제한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	4. Tag에 의존성이 있을 시, 팝업 창을 출력하고 원상태로 복귀한다. 6. 팝업 창에서 취소하면 원상태로 복귀한다.

Use Case	Tag 수정
Actor	User
Purpose	Tag의 정보를 수정한다.
Overview	Tag를 지정하여 정보(이름, 의존성, Tag 설명 등)를 수정한다.
Type	Primary and Essential
Cross Reference	R 2.3 , R 5.1 use case : Log File 생성
Pre-Requisites	window-a에서 Tag 메뉴가 선택된 상태여야 한다. 수정하고자 하는 Tag가 이미 존재하여야 한다.
Typical Courses of Events	(A): Actor, (S) : System 1. (A) : window-b의 태그 목록 중에서 수정할 tag를 선택한다. 2. (S) : window-b의 버튼을 활성화한다. 3. (A) : window-b의 Modify버튼을 클릭한다. 4. (S) : Tag 수정을 위한 팝업 창을 보여준다.

	<p>5. (A) : window-c에 태그의 이름 <i>name</i>, 종류 <i>type</i>, 색깔 <i>color</i>, 의 존성 <i>dependency</i> 정보를 수정한다.</p> <p>6. (S) : 입력된 Tag의 정보 중 <i>name</i>이 중복되었는지 체크한다.</p> <p>7. (S) : <i>Tag</i>의 정보를 수정한다.</p> <p>8. (S) : <i>LogFile</i>을 생성한다.</p>
Alternative Courses of Events	NONE
Exceptional Courses of Events	6. 입력된 Tag의 이름이 존재할 시, 메시지를 출력하고 재입력을 요청한다.

Use Case	Tag 지정
Actor	User
Purpose	파일, 폴더에 Tag를 지정한다.
Overview	존재하는 Tag를 파일, 폴더에 지정합니다.
Type	Primary and Essential
Cross Reference	R 2.4.1 , 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 <ol style="list-style-type: none"> 1. (A) : 사용자가 window-C에서 <i>file</i>을 선택한다. 2. (S) : window-B에 버튼을 활성화 시킨다. 3. (A) : window-B의 Attach Tag 버튼을 클릭한다. 4. (S) : 팝업 창에 선택 가능한 <i>Tag</i>들을 출력한다. 5. (A) : 팝업 창에서 <i>Tag</i>를 선택한다. 6. (S) : 해당 <i>file</i>에 선택한 <i>Tag</i>를 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 2.4.2 , R 5.1

	use case : Log File 생성
Pre-Requisites	<p>window-a에서 Tag 메뉴가 선택된 상태여야 한다.</p> <p>파일에 Tag를 해제하기 위해서는, 그 파일에 Tag가 이미 지정되어 있어야 한다.</p>
Typical Courses of Events	<p>(A) : Actor (S) : System</p> <ol style="list-style-type: none"> 1. (A) : 사용자가 window-C에서 <i>file</i>을 선택한다. 2. (S) : window-b의 버튼을 활성화 시킨다. 3. (A) : window-b의 Detach Tag 버튼을 클릭한다. 4. (S) : 선택된 <i>file</i>에 지정된 <i>Tag</i>가 있는지 확인한다. 5. (S) : 팝업 창을 띄워 Tag해제 여부를 확인한다. 6. (A) : Tag해제 여부를 선택한다. 7. (S) : 해당 <i>file</i>의 Tag를 해제한다.
Alternative Courses of Events	<ol style="list-style-type: none"> 7. Tag가 지정된 파일을 삭제한다면, 그 파일에 대한 Tag를 해제해준다.
Exceptional Courses of Events	<ol style="list-style-type: none"> 4. 선택한 파일에 Tag가 지정되어 있지 않다면, 팝업 창을 출력하고 원상태로 복귀한다.. 6. 취소를 선택하면 원상태로 돌아간다.

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 2.5.1, R 5.1 use case : Log File 생성
Pre-Requisites	<p>window-a에서 Tag 메뉴가 선택된 상태여야 한다.</p> <p>Tag에 의존성을 부여하기 위해서는 Tag가 미리 생성되어 있어야 한다.</p>
Typical Courses of Events	<p>(A) : Actor (S) : System</p> <ol style="list-style-type: none"> 1. (A) : 사용자가 window-C에서 <i>Tag</i>를 선택한다. 2. (A) : window-B의 Tag 의존성 부여 버튼을 클릭한다. 3. (S) : 팝업 창에 의존성 부여가 가능한 <i>Tag</i>들을 출력한다. 4. (A) : 팝업 창에서 의존성을 부여할 <i>Tag</i>를 선택한다. 5. (S) : 해당 <i>Tag</i>에 의존성을 부여해 준다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

Use Case	Tag 의존성 삭제
Actor	User
Purpose	Tag간의 의존성을 삭제한다.
Overview	Tag의 삭제 및 구조를 변경하기 위해서 Tag의 의존성을 삭제한다.
Type	Primary and Essential
Cross Reference	R 2.2, R 2.5.2 , R 5.1 Use Case : Tag 삭제, Log File 생성
Pre-Requisites	window-a에서 Tag 메뉴가 선택된 상태여야 한다. Tag에 의존성을 삭제하기 위해서는 Tag가 미리 생성되어 있어야 하며, 그 Tag에 의존성이 존재하여야 한다.
Typical Courses of Events	(A) : Actor (S) : System 1. (A) : 사용자가 window-b에서 Tag를 선택한다. 2. (A) : window-B에서 Tag 의존성 삭제 기능을 선택한다. 3. (S) : 팝업 창에 의존성의 목록을 출력한다. 4. (A) : 팝업 창에서 삭제하고자 하는 의존성을 클릭한다. 5. (S) : 선택된 <i>Dependency</i> 를 삭제한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

Use Case	Tag 구조 보기
Actor	User
Purpose	파일들의 구조를 보여준다
Overview	Tag목록들의 Hierarchy를 보여준다. 파일간의 연관성을 토대로 트리 형태의 구조를 보여준다. 사용자가 Hierarchy상에서 수정이 가능하도록 한다.
Type	Primary and Essential
Cross Reference	R 2.6
Pre-Requisites	NONE
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : window-A에서 Tag 메뉴를 선택한다. 2. (S) : window-C에 Tag의 Dependency를 Tree 구조로 출력한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

Use Case	ServiceDirectory
Actor	User
Purpose	Nametag를 이용하여 지정된 directory에서 파일 자동 분류 기능을 수행한다.
Overview	자동으로 파일을 분류한다.
Type	Primary and Essential
Cross Reference	R 4.1
Pre-Requisites	Nametag가 생성되어있어야 한다.
Typical Courses of Events	(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 <i>true</i> 값을 반환한다. . 3. (S) : <i>true</i> 가 반환되면 <i>Logfile</i> 에 동작을 기록한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

Use Case	Name Tag 생성
Actor	User
Purpose	Name Tag를 생성한다.
Overview	Service Directory상에서 실행되는 파일 및 폴더 이동을 위한 Name Tag를 생성한다. Name Tag의 이름과 해당 폴더위치를 입력하여 시행한다
Type	Primary and Essential
Cross Reference	R 4.2.1, R 5.1 use case : Log File 생성
Pre-Requisites	window-a에서 Name Tag 메뉴가 선택된 상태여야 한다.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : window-B에서 Name Tag 생성 기능을 선택한다. 2. (S) : 팝업 창으로 Name Tag의 생성 창을 출력한다. 3. (A) : <i>Nametag</i> 에 대한 이름 <i>name</i> , 검사주소 <i>serviceDirectory</i> , 이동주소 <i>serviceDirectory</i> , 설명 <i>explain</i> 을 입력하고 확인버튼을 누른다. 4. (S) : 입력 받은 조건의 <i>Nametag</i> 를 생성한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

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 선택한다. 2. (A) : window-b에서 Delete 버튼을 클릭한다. 3. (S) : Name Tag 삭제 여부를 팝업으로 확인한다. 4. (A) : 팝업 창에서 삭제 여부를 선택한다. 5. (S) : Name Tag를 삭제하고, 확인 버튼을 팝업 창으로 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 선택한다. 2. (A) : window-b에서 Modify 버튼을 클릭한다. 3. (S) : 팝업 창으로 Name Tag의 수정 창을 출력한다. 4. (A) : 팝업 창에서 수정할 이름 <i>name</i> , 검사주소 <i>serviceDirectory</i> , 이동주소 <i>serviceDirectory</i> , 설명 <i>explain</i> 을 입력하고 확인 버튼을 누른다.

	5. (S) : <i>NameTag</i> 의 정보를 수정한다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	4. 취소 버튼을 누르거나 창을 끄면 정보가 수정되지 않는다.

Use Case	Name Tag 목록보기
Actor	User
Purpose	Name Tag의 목록을 보여준다.
Overview	- Service Directory상에서 실행되는 모든 Name Tag의 목록을 보여준다. - Name Tag의 이름과 해당하는 폴더 위치를 보여준다.
Type	Primary and Essential
Cross Reference	R 1.1, R 4.2.4 Use Case : 파일실행
Pre-Requisites	
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : window-a에서 Name Tag를 선택한다. 2. (S) : window-b에 Name Tag 목록을 보여준다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

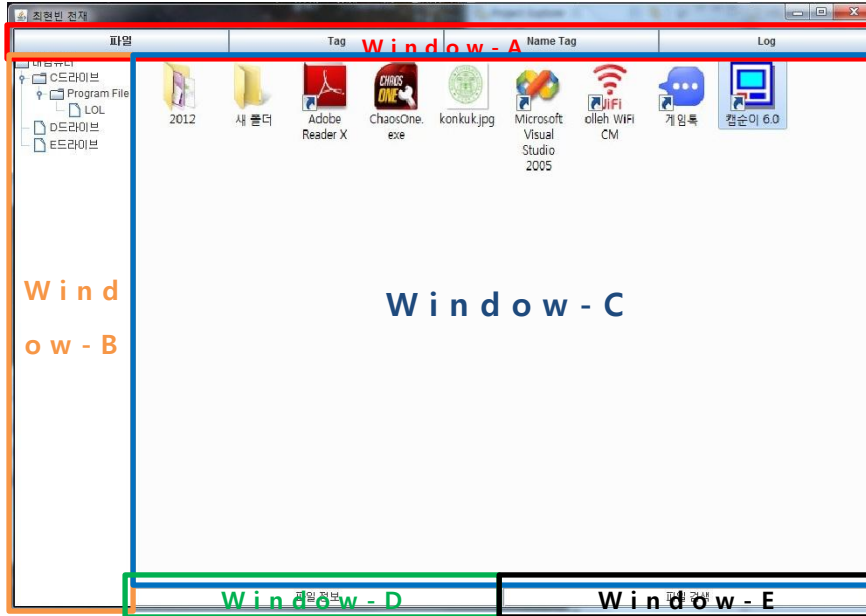
Use Case	Log File 생성
Actor	User
Purpose	TDMS상의 Tag의 관련 동작을 Log File에 기록한다
Overview	파일 삭제, 파일 수정, Tag 생성, Tag 삭제, Tag 수정, Tag 지정, Tag 해제, Tag 의존성 부여, Tag 의존성 삭제, Name Tag 생성, Name Tag 삭제, Name Tag 수정의 동작을 Log File에 기록한다.
Type	Primary and Essential
Cross Reference	R 1.2.1, R 1.3, R 2.1, R 2.2, R 2.3, R 2.4.1, R 2.4.2, R 2.5.1, R 2.5.2, R 4.2.1, R 4.2.2, R 4.2.3, R5.1 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
Pre-Requisites	TDMS상에서 사용자가 조회 또는 보기 등을 제외한 다른 동작을 수행하여야 한다.
Typical Courses of Events	(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.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

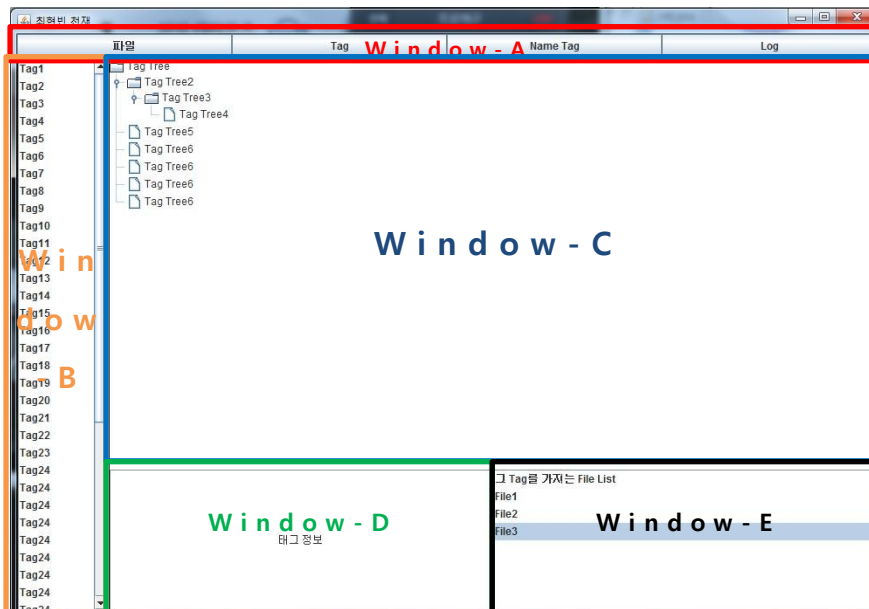
Use Case	Log File 보기
Actor	User
Purpose	생성된 Log File을 확인한다.
Overview	- 생성된 Log File을 확인할 수 있다.
Type	Primary and Essential
Cross Reference	R 5.2
Pre-Requisites	window-a에서 Log 메뉴가 선택된 상태여야 한다. TDMS상에서 사용자가 검색 또는 보기 등을 제외한 다른 동작을 수행하여야 한다.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : window-b에서 Log File을 선택한다. 2. (S) : window-c에 선택된 Log File의 내용을 보여준다.
Alternative Courses of Events	NONE
Exceptional Courses of Events	NONE

Stage 2142. Define Reports, UI, and Storyboards

- User Interface (File)



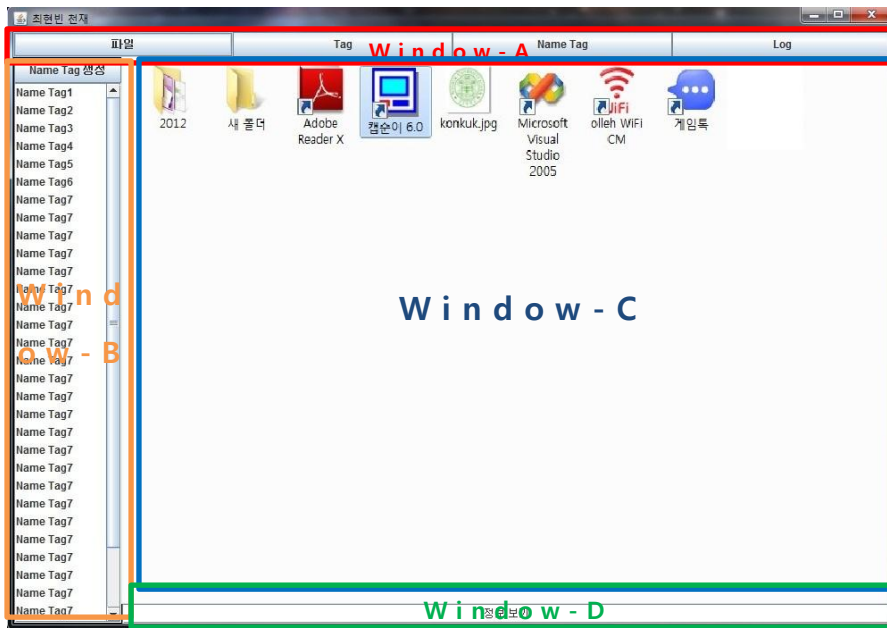
- User Interface (Tag)



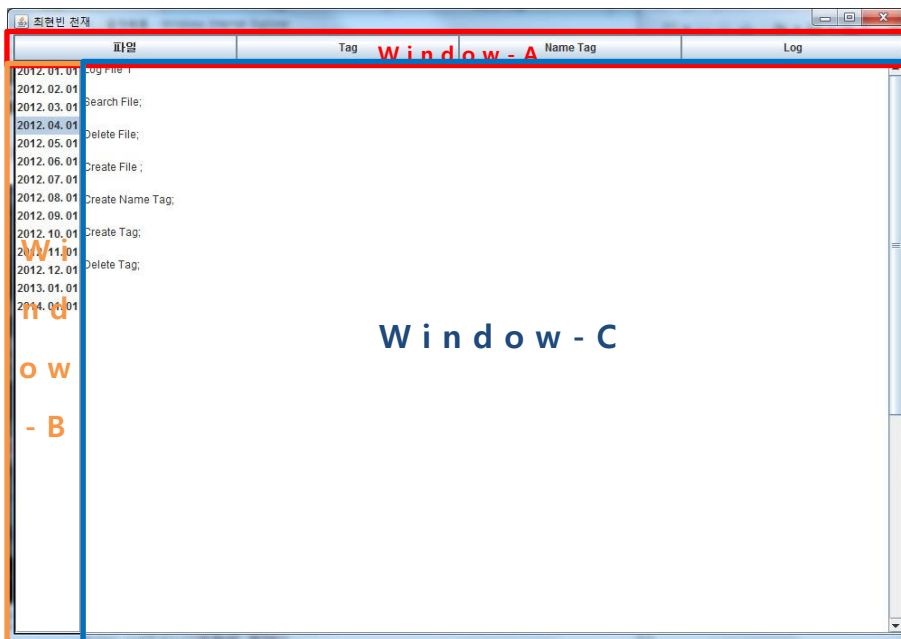
Software Modeling & Analysis

OSP Stage 2140 v_120506

- User Interface (Name Tag)



- User Interface (Log)



Stage 2143. Refine System Architecture

TDM5

Interface

```

-!og: string
-!oad: enum
-!ob: enum

+removeFile(id: int): void
+showDependency(): void
+throw(input: boolean)
+excuteFile(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

```

-!ag: 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

```

-!ame: string
-!explain: string
-!d: 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
-!d: int
    
```

Nametag

```

-!name: string
-!explain: string
-!d: int
-!serviceDirectory
    
```

FileMgr

```

-!file: list<File>

+removeFile(id: int): void
+excuteFile(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

```

-!d: int
-!information: string
-!binder
-!tagId: int
-!isTagged: boolean
    
```

Service

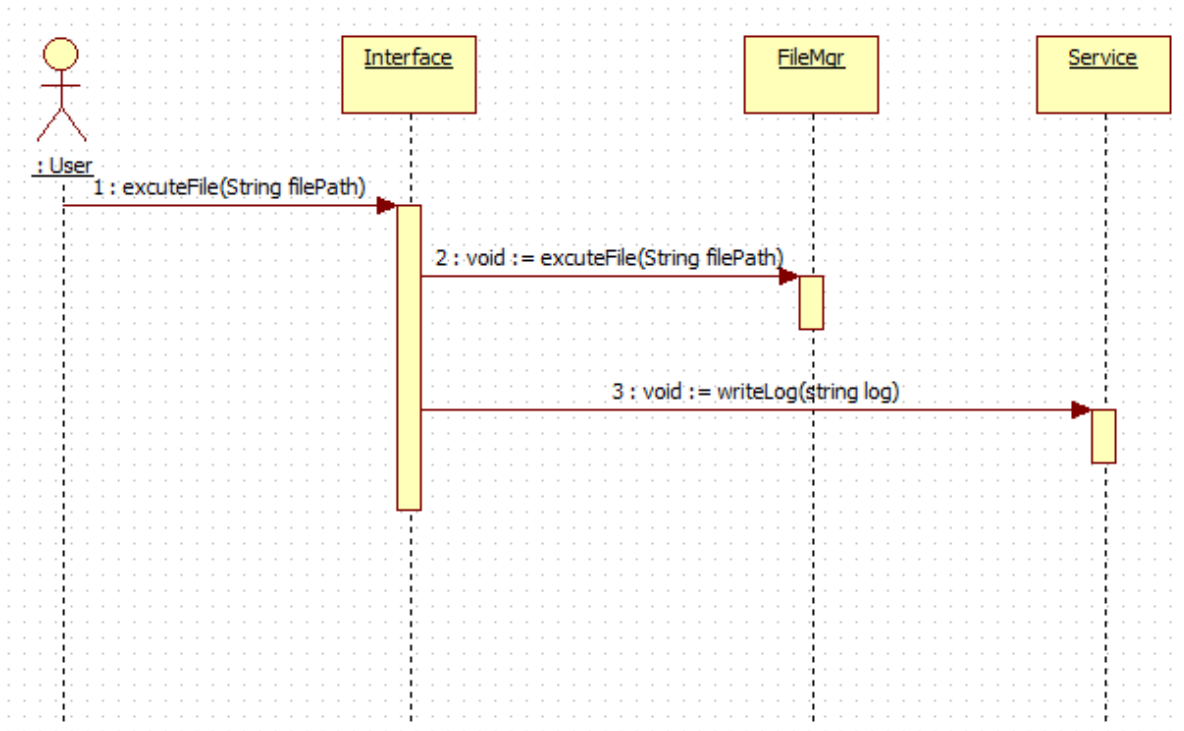
```

-!log: list< string >

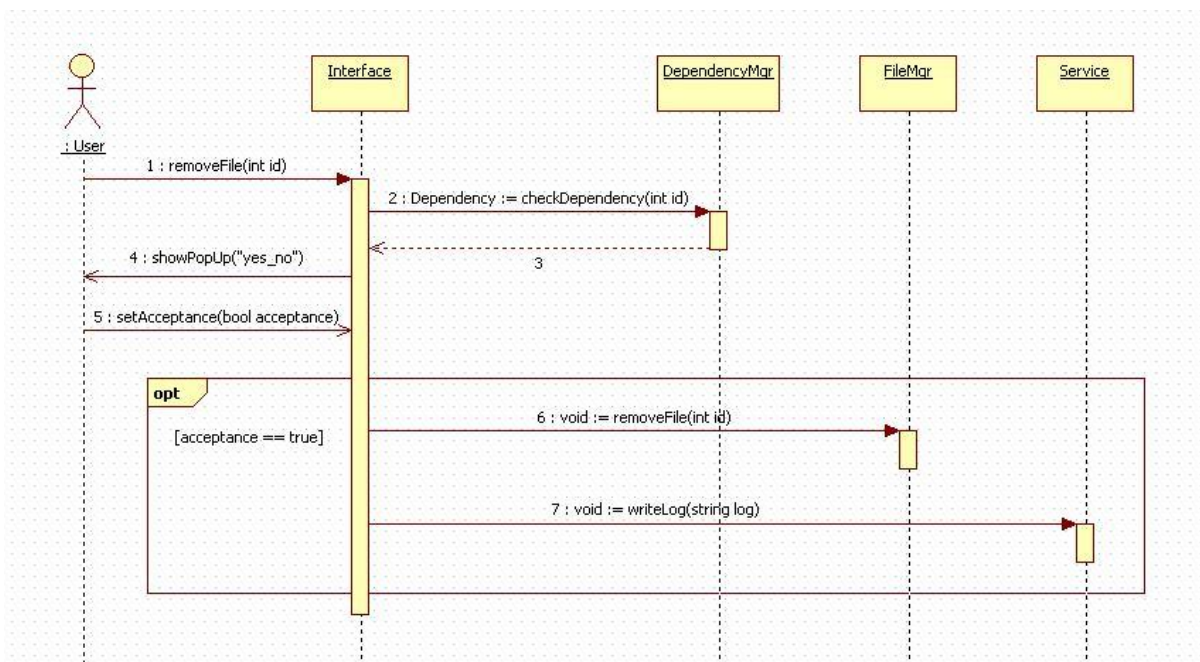
+writeLog(log: string): void
+getLogFile(): string
+serviceDirectory(): void
    
```

Stage 2144. Define Interaction Diagrams

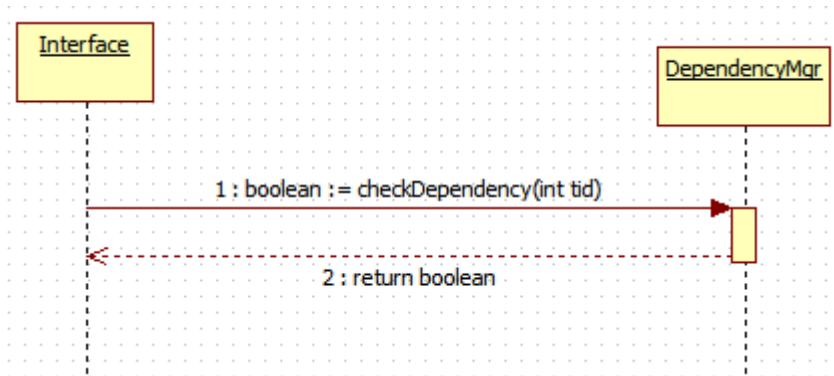
- excuteFile()



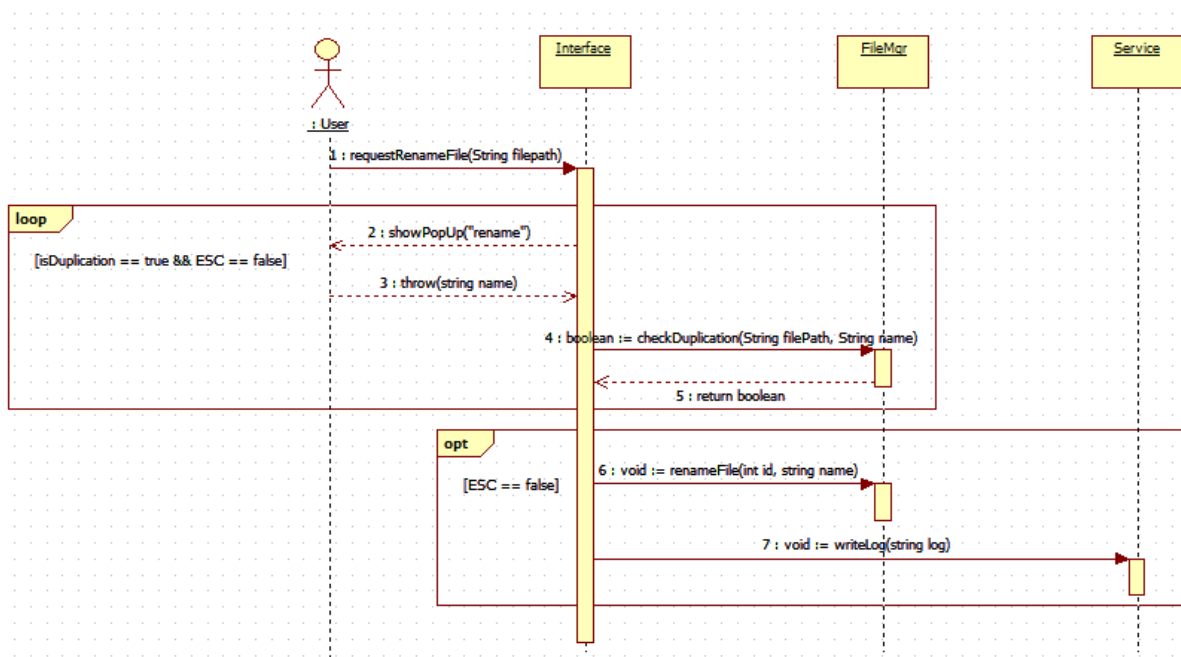
- removeFile()



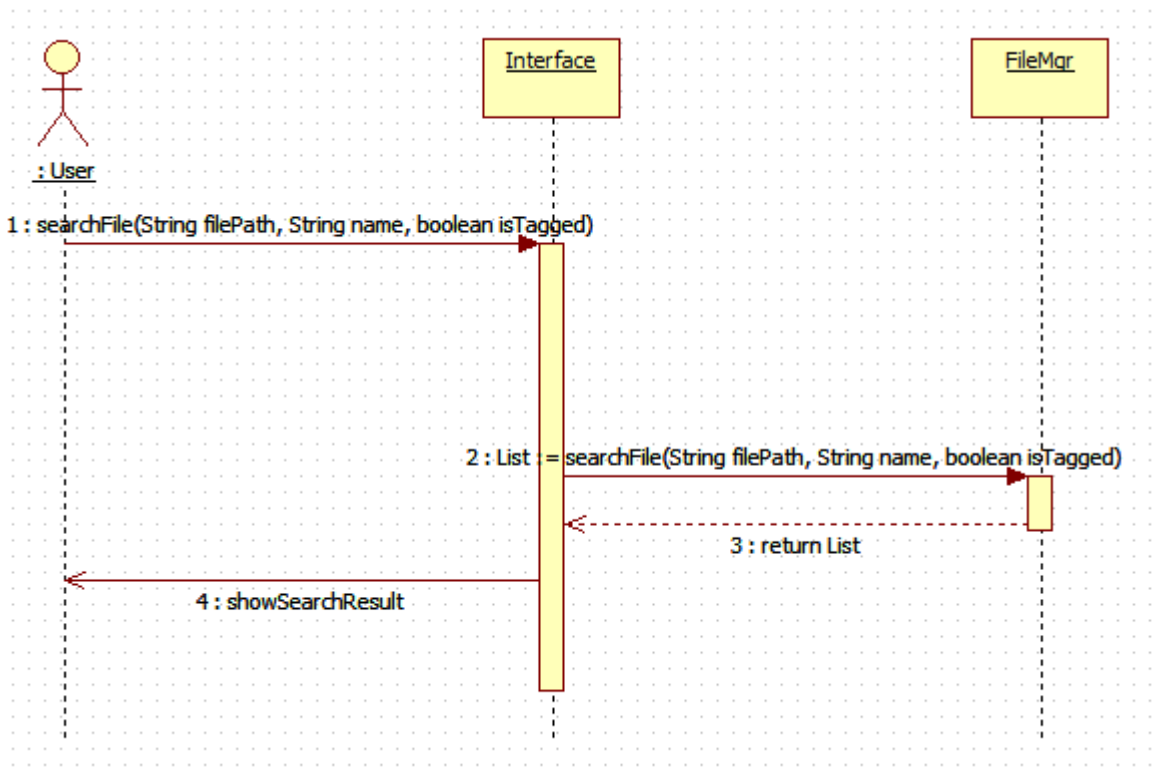
- checkDependency()



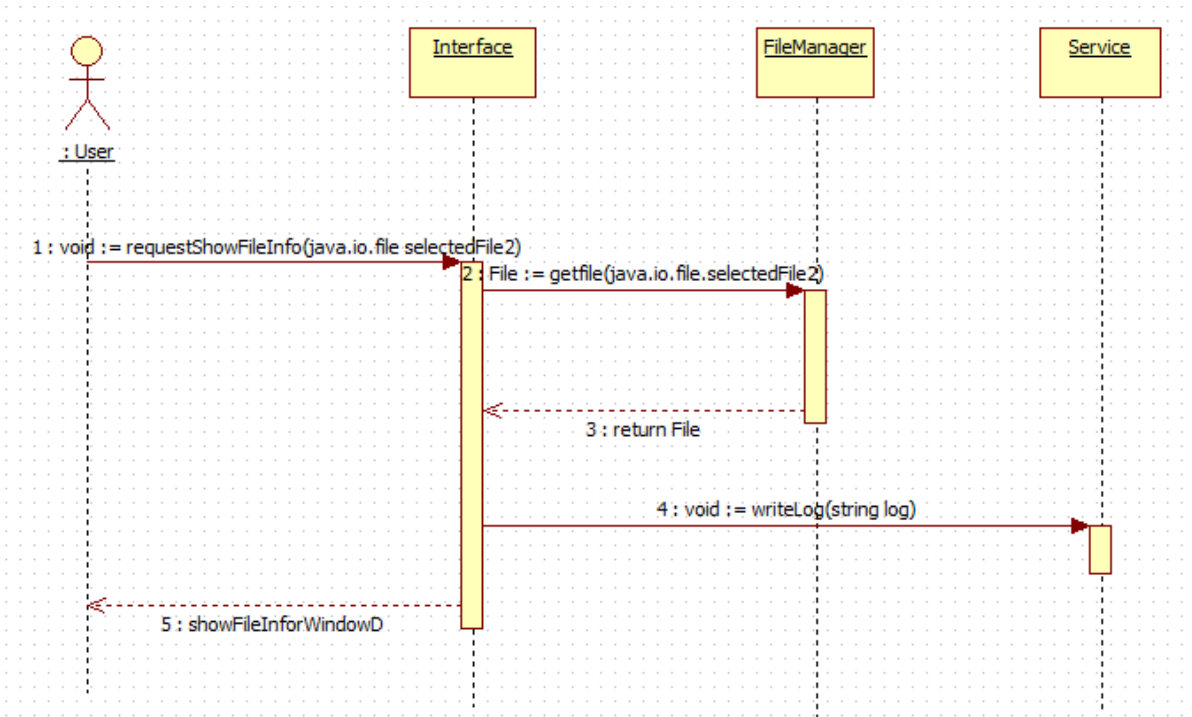
- requestRenameFile()



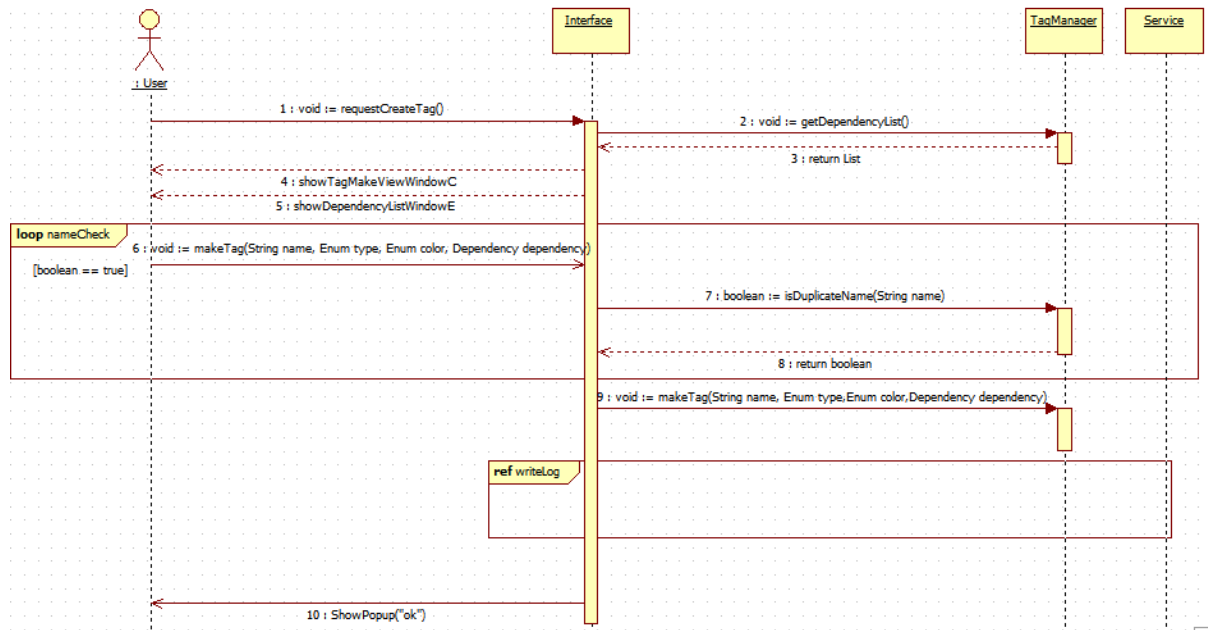
- searchFile()



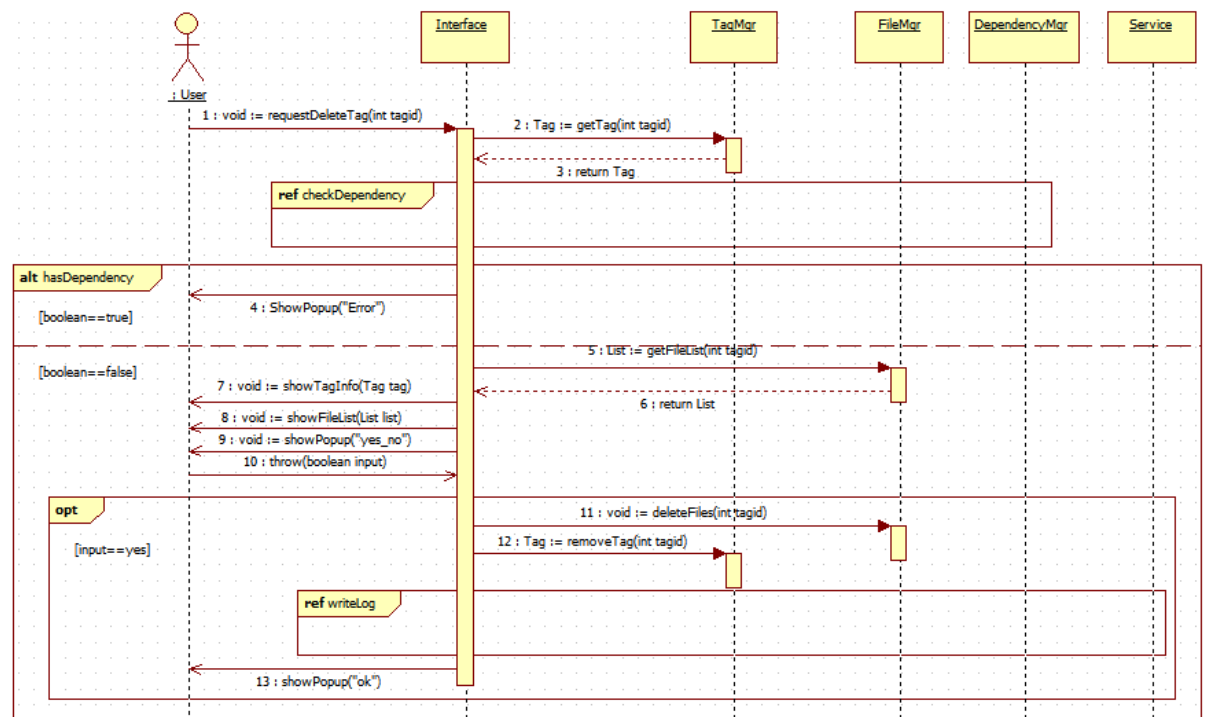
- requestShowFileInfo()



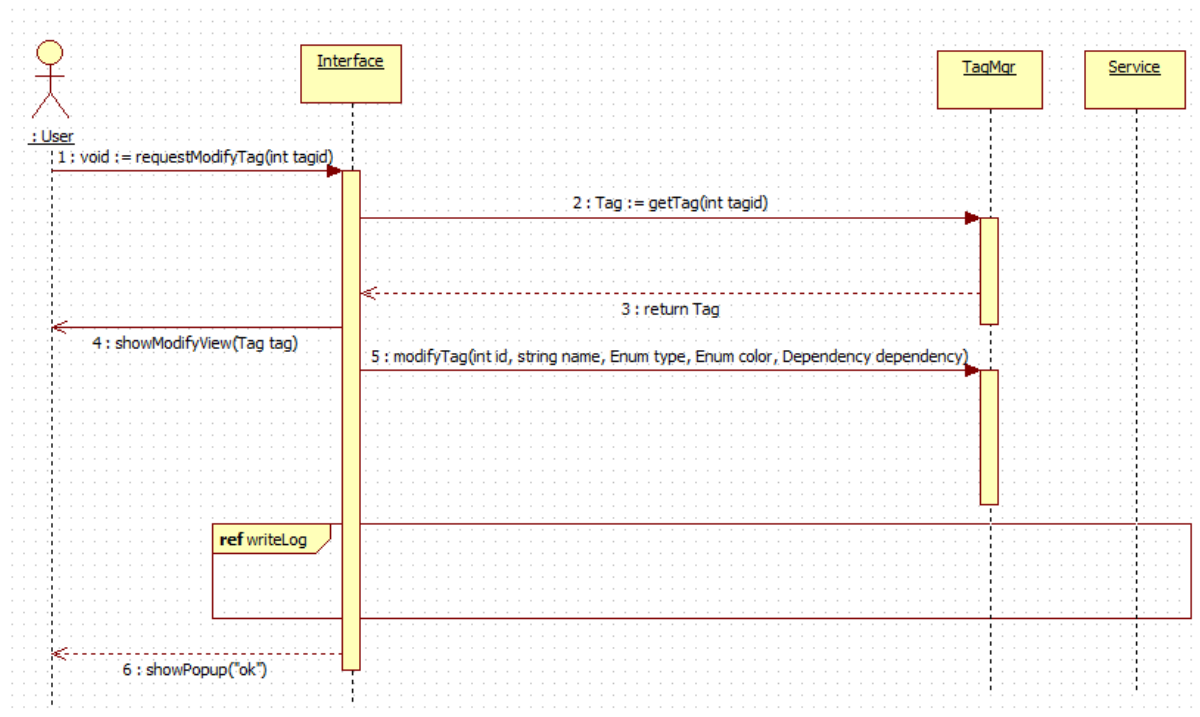
- requestCreateTag()



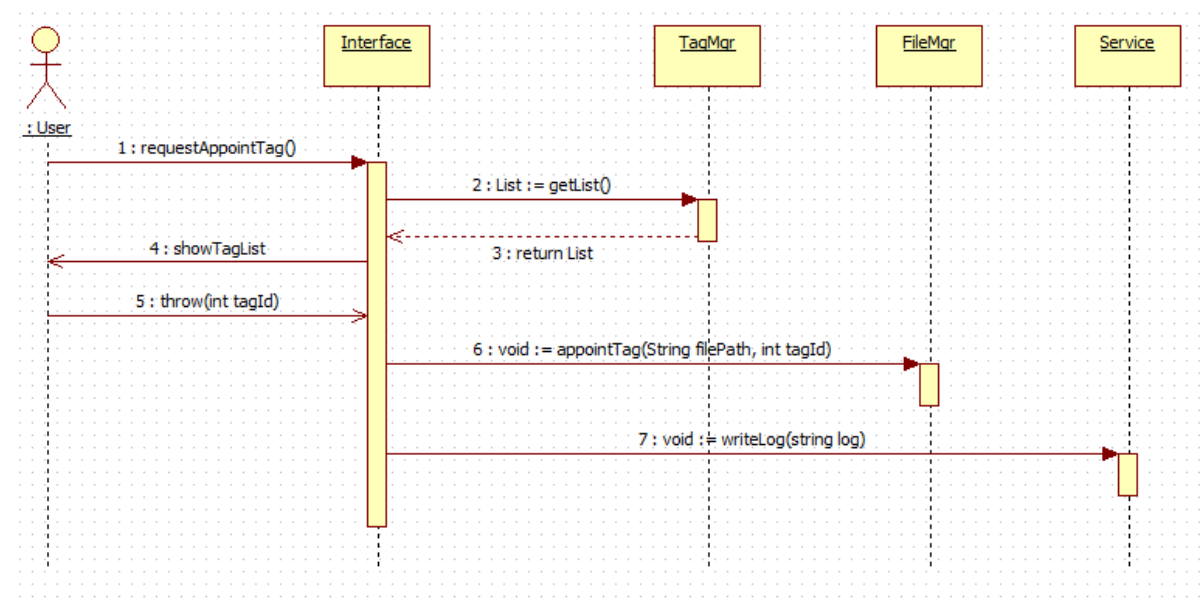
- requestDeleteTag()



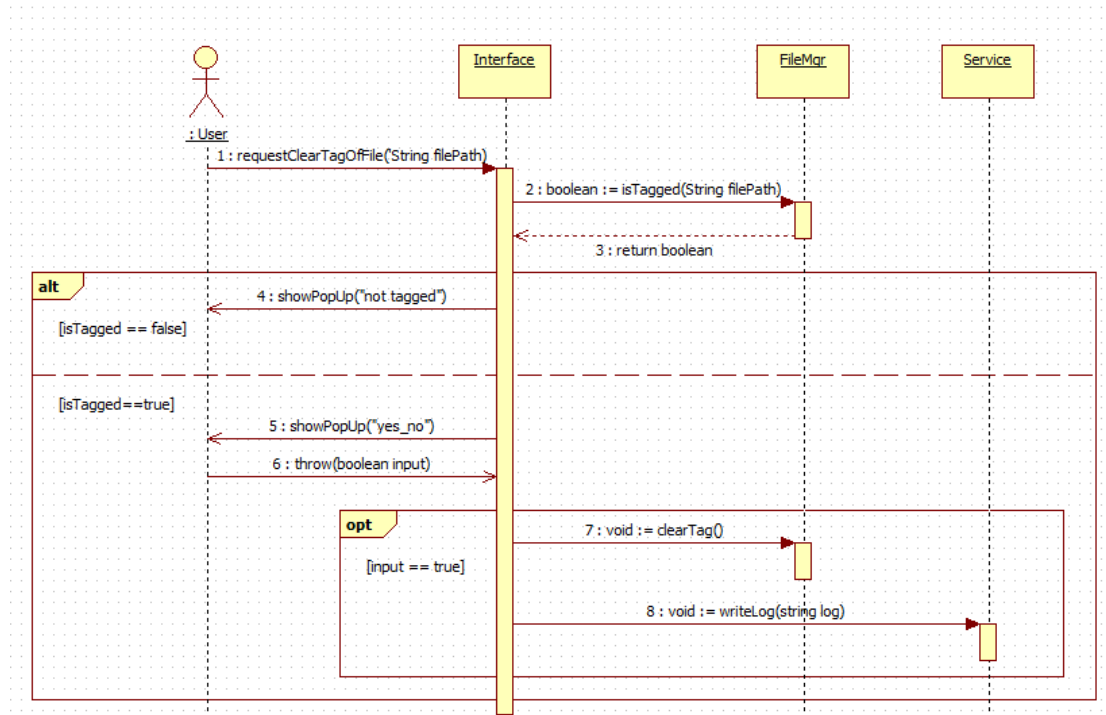
- requestModifyTag()



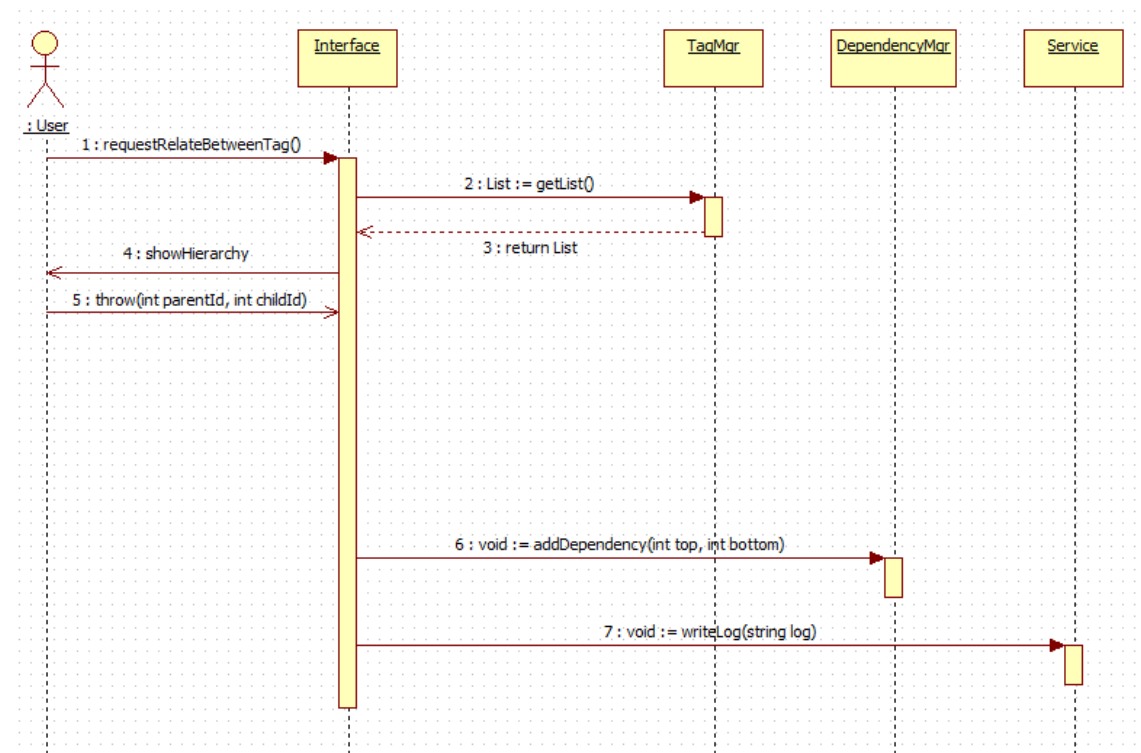
- requestAppointTag()



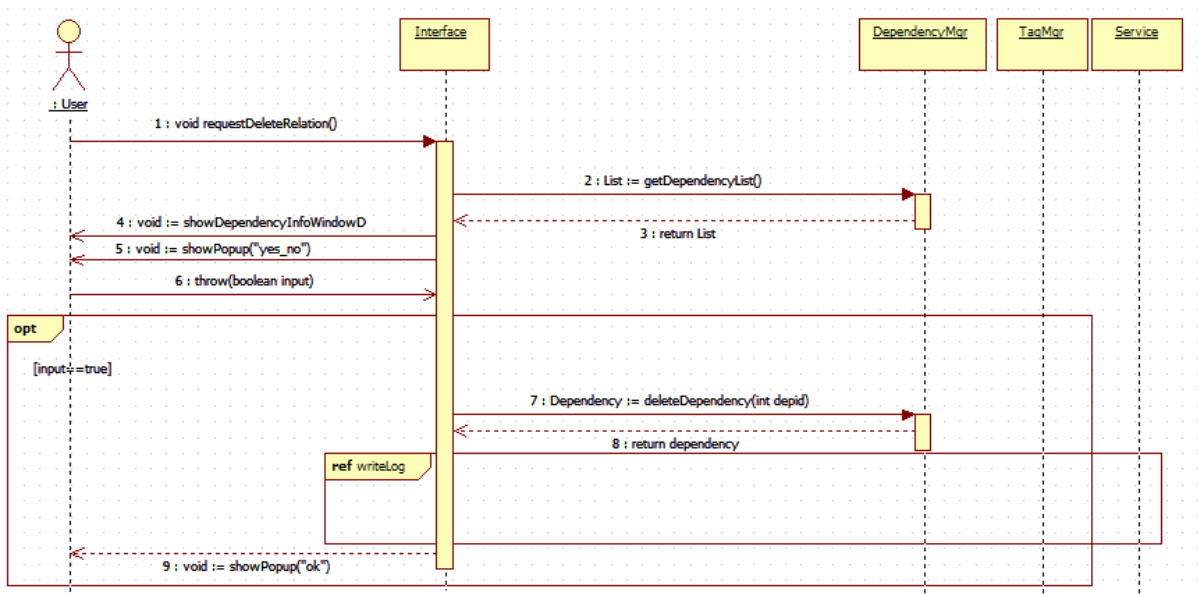
- requestClearTagOfFile()



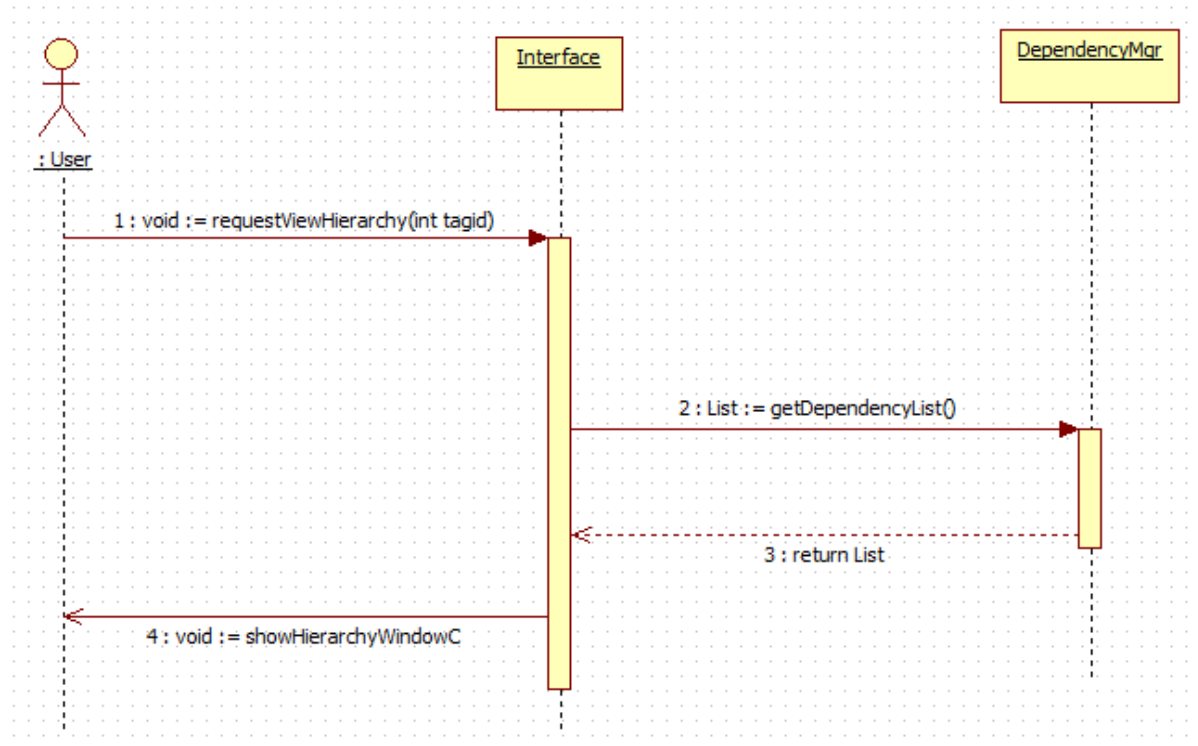
- requestRelateBetweenTag()



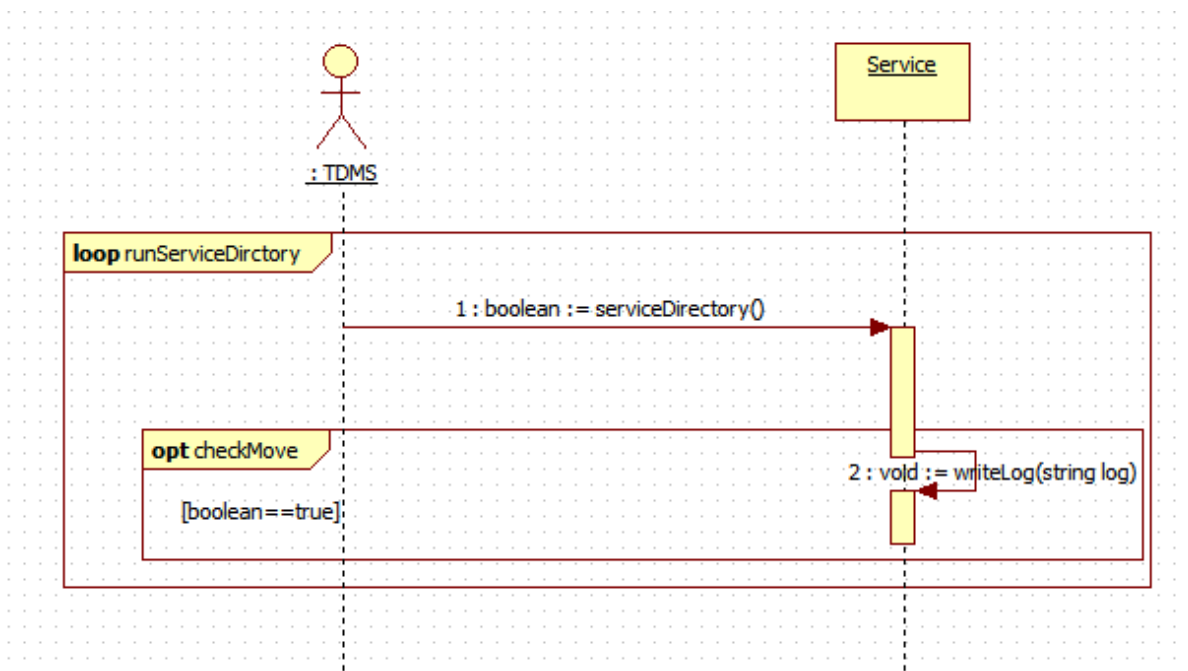
- requestDeleteRelation()



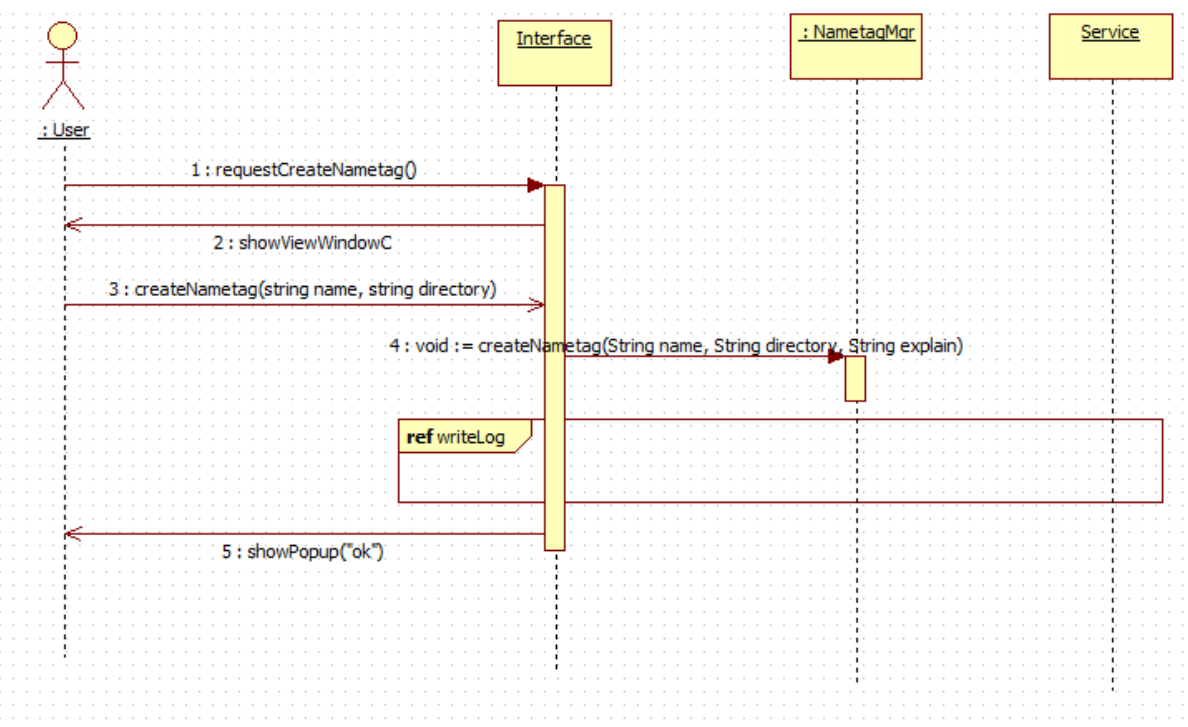
- requestViewHierarchy()



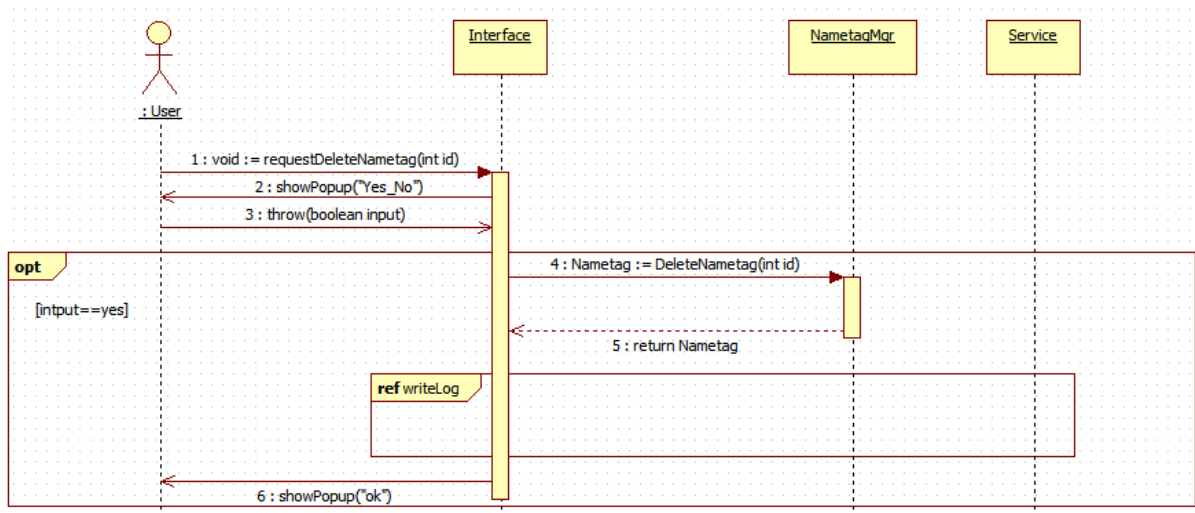
- serviceDirectory()



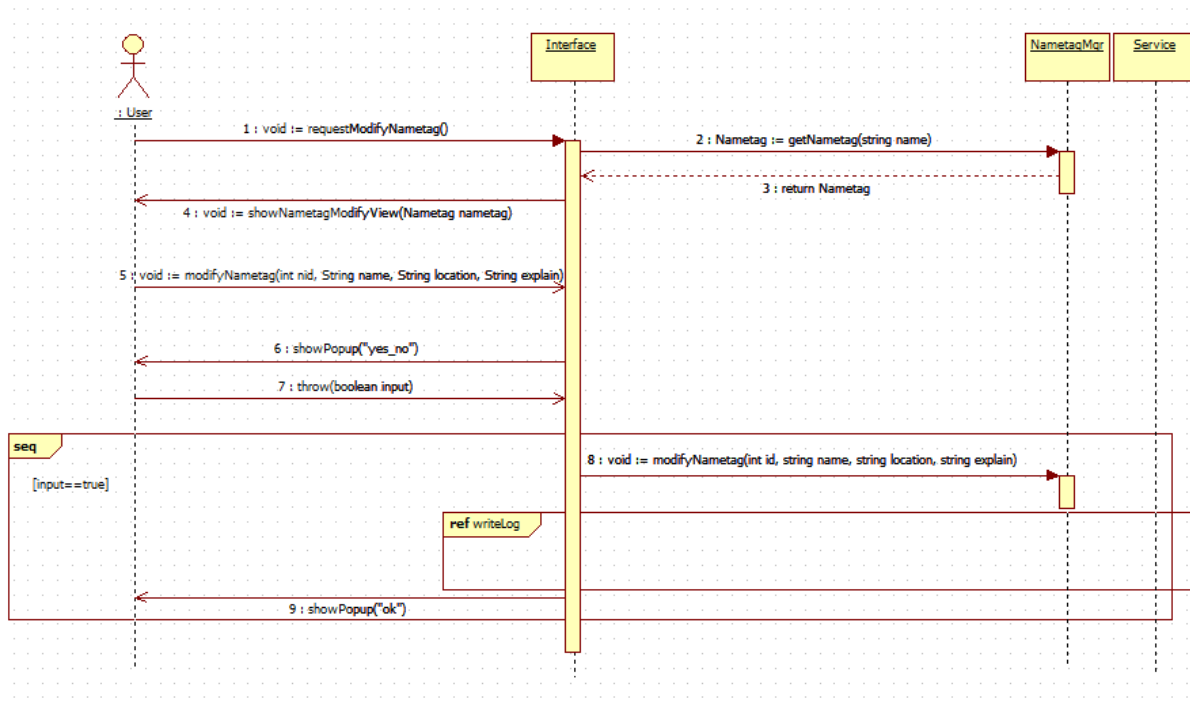
- requestCreateNameTag()



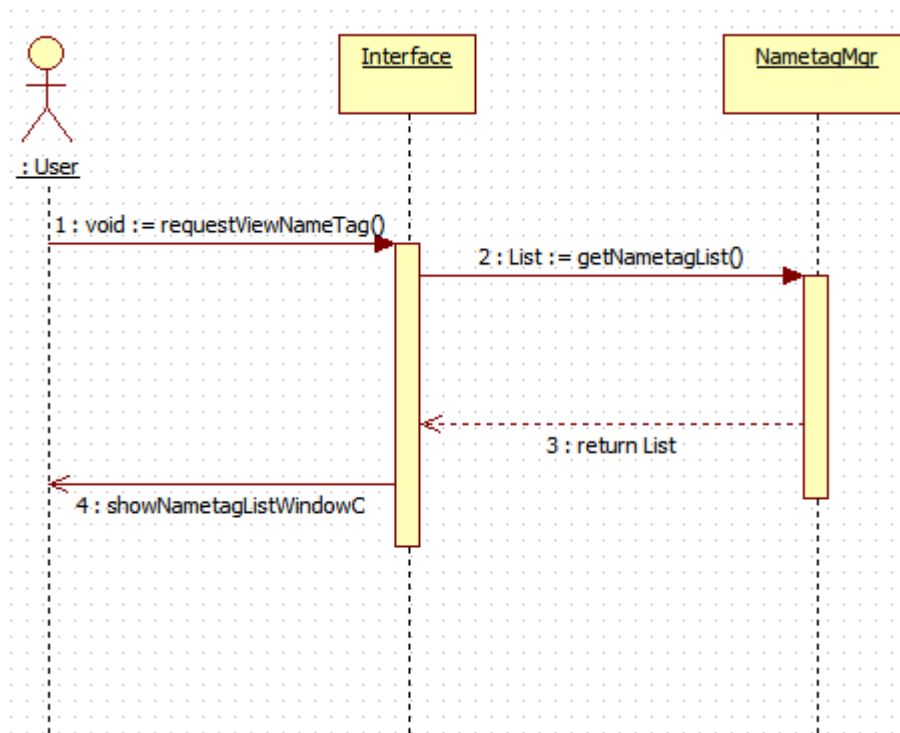
- requestDeleteNameTag()



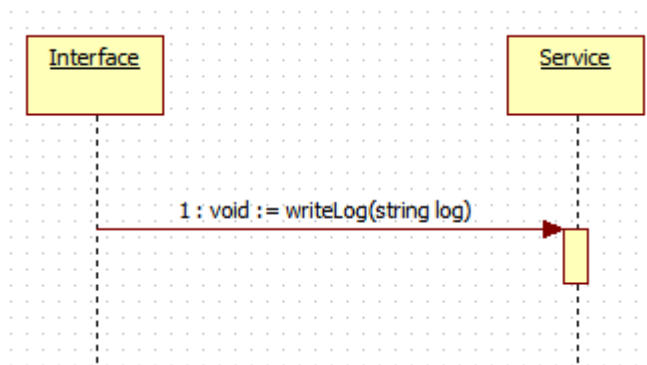
- requestModifyNameTag()



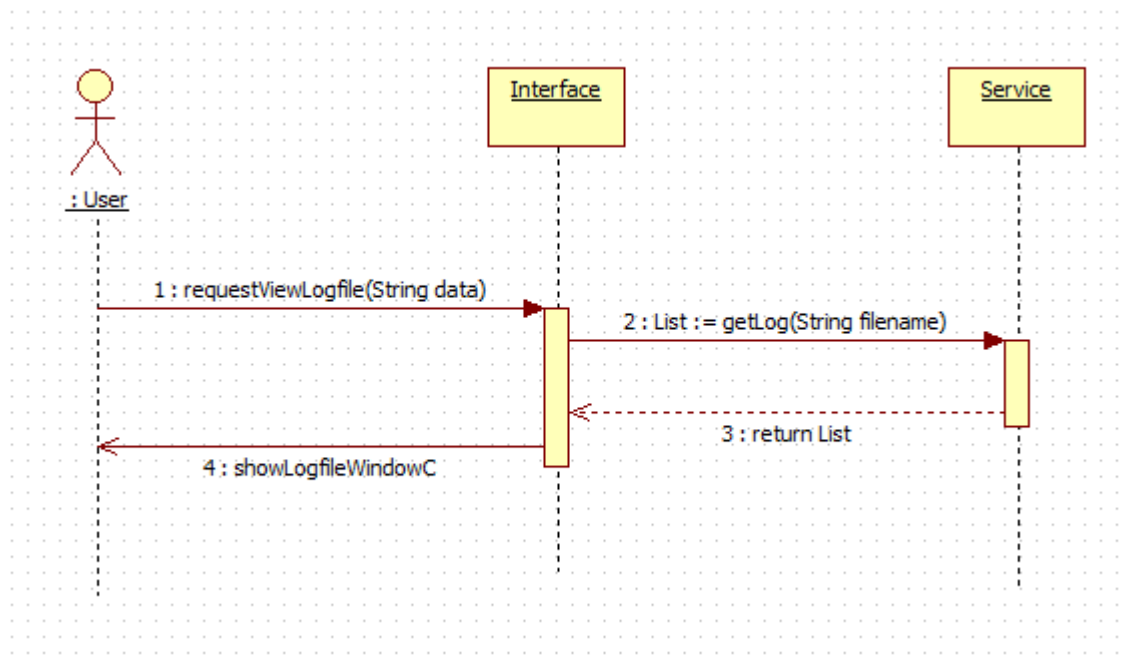
- requestViewNameTag()



- writeLog()



- requestViewLogFile()



Stage 2145. Define Design Class Diagrams

