

저작물 불법 유통 신고 시스템

소프트웨어 요구사항 명세서  
(Software Requirement Specification)

# 목 차

1. 서 론(Introduction)
  - 1.1. 목적(Purpose)
  - 1.2. 범위(Scope)
  - 1.3. 용어 및 약어 정의(Definitions, acronyms, and abbreviations)
  - 1.4. 참고 문헌(References)
  - 1.5. 개요(Overview)
2. 종합 기술(Overall Description)
  - 2.1. 프로젝트 결과물 전망(Product perspective)
  - 2.2. 제품 기능(Product functions)
  - 2.3. 사용자 특성(User characteristics)
  - 2.4. 제약조건(Constraints)
  - 2.5. 가정과 의존성(Assumptions and dependencies)
3. 상세 요구사항(Requirement Specification)
  - 3.1. 인터페이스 요구사항
  - 3.2. 기능적 요구사항
  - 3.3. 성능 요구사항
  - 3.4. 논리적 데이터베이스 요구사항
4. 추가 정보(Supporting Information)
  - 4.1 문서 이력

## 1. 서 론(Introduction)

### 1.1. 목 적(Purpose)

본 문서는 저작물의 불법 유통에 대한 신고를 받고 신고의 정보를 바탕으로 하이퍼링크를 탐색하여 다수의 불법 유통을 찾는 시스템의 구현을 위한 요구사항 명세서이다. 한양대학교 컴퓨터소프트웨어학부의 서왕규 프로젝트를 설계 및 구현하기 위해 요구사항을 정리하여 시스템을 설계 및 구현한다.

### 1.2. 범 위(Scope)

본 프로젝트는 다음의 범위의 개발을 진행한다.

- 신고 접수 웹 어플리케이션
- 보상재화 ERC20
- 합의 프로토콜과 그것을 수행하는 스마트 컨트랙트
- 웹 크롤링 스케줄링
- 웹 크롤러

### 1.3. 용어 및 약어 정의

저작권자	창작물에 대한 저작권을 소유한 자. 본 문서에서는 해당 저작물로 인해 수익을 창출하는 저작인접권자를 포함함.
신고자	일반 사용자의 역할 중 한 가지로 웹 어플리케이션을 통해 신고를 하는 사용자.
검증자	일반 사용자의 역할 중 한 가지로 웹 어플리케이션을 통해 신고의 유효성을 판단하는 사용자.
노드서버	네트워크에서 장치나 데이터의 지점을 의미. 본 문서에서는 각 저작권자의 웹 크롤러 서버 관리 프로그램이 설치된 서버를 뜻함
저작물	저작권에 의해 보호되는 창작물
중앙서버	본 문서에서는 해당 웹 어플리케이션과 중앙 데이터베이스를 연결하기 위한 API 서버를 뜻함
ERC20	이더리움 네트워크의 표준(EIP-20)을 준수하는 재화
웹 크롤러	조직적, 자동화된 방법으로 월드 와이드 웹을 탐색하는 소프트웨어

### 1.4. 참고 문헌 (References)

- [1] Najork, Marc. "Web Crawler Architecture." (2009): 3462-3465.
- [2] 이기룡, & 이희조. (2016). HTML 태그 순서를 이용한 불법 사이트 탐지 자동화 기술. 정보과학회논문지, 43(10), 1173-1178.

## 1.5. 개요 (Overview)

본 문서의 구성은 다음과 같다. 2장에서 저작물 불법 유통 신고 시스템의 전반적인 개요를 설명한다. 3장에서는 상세한 인터페이스 요구사항, 기능적 요구사항, 데이터 요구사항을 설명한다. 4장에서는 문서의 작성이력을 서술한다.

## 2. 종합 기술 (Overall Description)

### 2.1. 프로젝트 결과물 전망 (Product perspective)

본 프로젝트로 생성되는 결과물은 현재의 특정 인력의 고용과 각 인터넷 사업자들이 자체적으로 필터링하는 구조의 불법 유통 적발 방식이 아닌, 불특정 다수의 인력의 고용과 자체적인 웹 크롤러를 이용하여 불법 유통을 적발하는 방식의 시스템의 개발을 목적으로 한다. 이를 위해, 제작되는 시스템은 불법 유통 신고를 받는 웹 어플리케이션과 신고내역을 통해 탐색하는 웹 크롤러로 구성된다.

이 프로젝트의 결과물은 다음의 절차를 수행한다.

- ① 불법 유통의 신고를 받고, 검증, 수락, 보상한다.
- ② 각 노드는 신고내역을 통해 탐색하는 웹 크롤러를 통해 모니터링 시스템을 구축한다.

해당 절차를 수행하는 시스템의 개발을 위해, 다음과 같은 컴포넌트를 구축한다.

- (1) 신고자가 불법 유통을 신고하기 위한 웹 어플리케이션
- (2) 웹 어플리케이션과 통신하는 중앙서버
- (3) 신고의 내역과 사용자 정보의 저장을 위한 데이터베이스
- (4) 신고의 내역을 바탕으로 불법 유통을 탐색하는 웹 크롤러
- (5) 웹 크롤러의 스케줄링과 중앙서버와의 통신을 담당하는 노드의 서버 관리 프로그램
- (6) 웹 크롤러가 탐색한 정보를 저장하는 노드의 로컬 데이터베이스
- (7) 보상 재화인 ERC20 토큰
- (8) 보상의 합의 프로토콜을 수행하는 스마트 컨트랙트

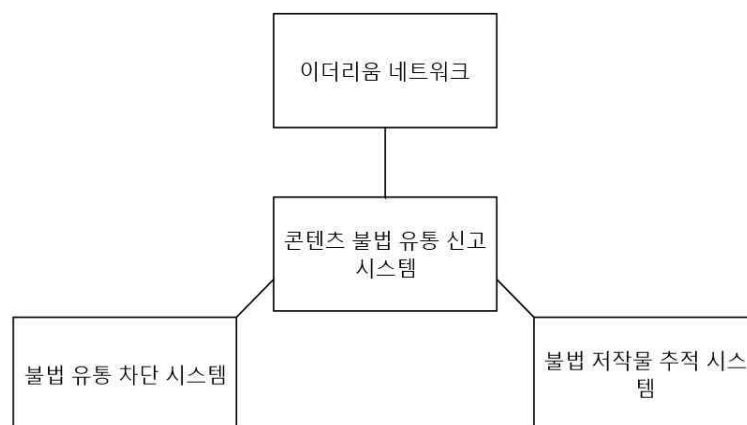


그림 1 Context diagram

개발된 시스템은 외부의 시스템들과 아래와 같이 연동되어 활용된다.

- 이더리움 네트워크에 ERC20을 배포하여 해당 ERC20을 신고자에게 주는 것으

로 보상한다.

- 발견한 불법 유통물을 불법 유통 차단 시스템인 DNS 차단, SNI 차단 또는 인터넷 사업자의 자체 시스템과 통신할 수 있는 인터페이스를 갖는다.
- 노드서버의 웹 크롤러는 디지털 워터마크, 디지털 포렌식과 같은 불법 저작물 추적 시스템과 연동할 수 있도록 인터페이스를 갖는다.
- 이더리움 네트워크에 스마트 계약을 배포하여 탈중앙화된 합의 모델을 수립한다.

## 2.2. 제품 기능 (Product functions)

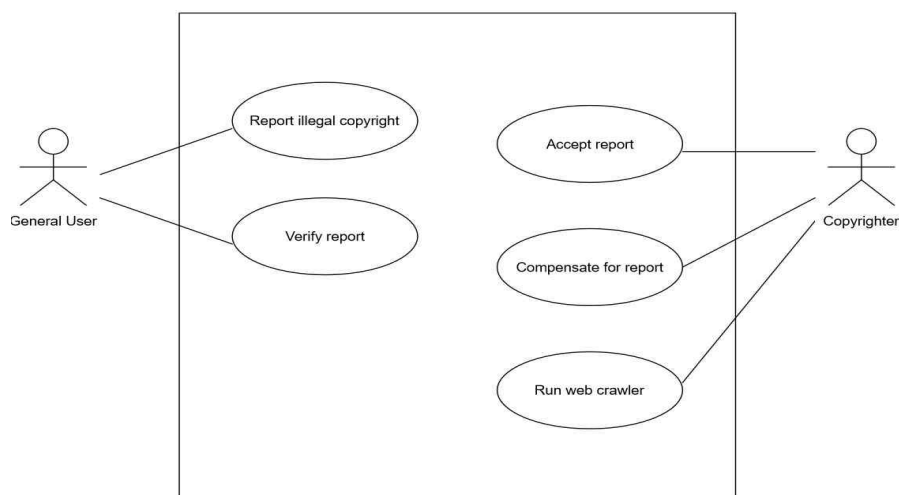


그림 2 Use-Case Model 명세

시스템은 관리자를 제외한 일반 사용자와 저작권자라는 두 종류의 사용자를 갖는다.

- 일반 사용자는 불법 저작물을 신고하거나, 신고를 검증한다.
- 저작권자는 신고를 수락하고, 보상을 부여하며, 웹 크롤러를 사용한다.

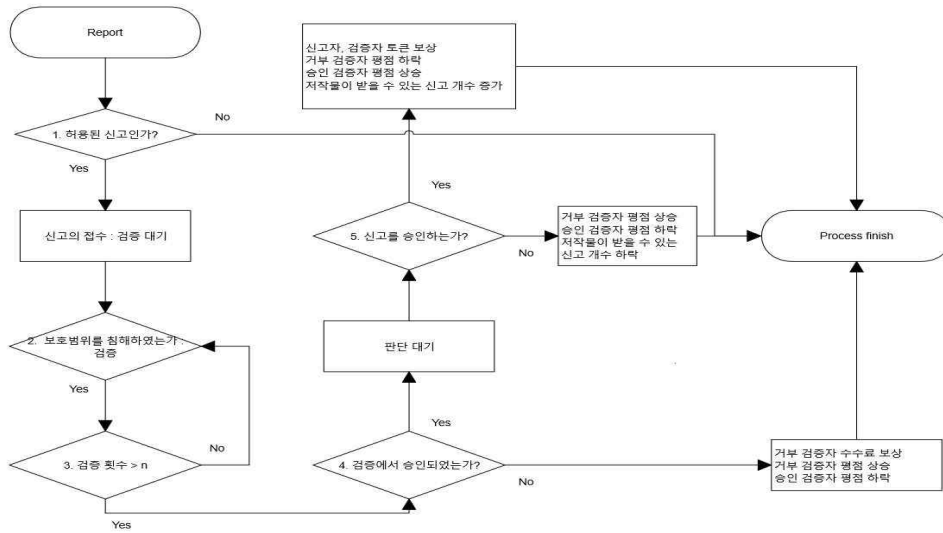


그림 3 신고 처리 순서도

신고는 그림3의 순서도와 같이 처리된다. 신고의 단계는 다음과 같이 구분된다.

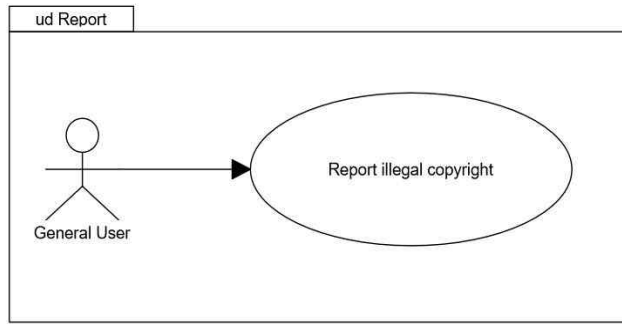
단계	이전 단계	내용
신고 접수	-	웹 어플리케이션을 통해 중앙서버로 전송된 직후의 신고로 사용자에게 접수의 결과가 전송되기 전
신고 접수 거부	신고 접수	중앙 서버에서 신고가 중복되거나, 유의미하지 않을 경우 거부
검증 대기	신고 접수	중앙 서버에서 신고가 유의미한지 판단한 이후의 신고로 일반 사용자로부터 검증하는 단계
검증 거부	검증 대기	일반 사용자들로부터 투표로 신고가 거부
판단 대기	검증 대기	일반 사용자들로부터 투표로 신고가 승인되어 저작권자의 판단을 대기하는 단계
판단 수락	판단 대기	저작권자에게 신고가 유효하다 판단되어 보상을 기다리는 단계
내역 전송 대기	판단 수락	이더리움 네트워크에서 보상이 부여된 것을 확인한 뒤, 노드에게 신고 내역을 전송하기 전 단계
판단 거부	판단 대기	저작권자에게 신고가 유효하지 않다고 판단되어 거부

적절한 처리 절차를 통해, 신고자와 저작권자 사이의 익명성이 보장되면서 신뢰도 높은 신고와 보상이 가능하도록 구현한다.

## 2.2.1 시나리오 : 일반 사용자

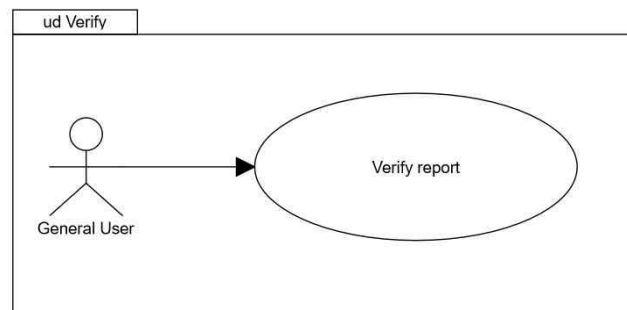
일반 사용자란 불법 유통에 대해 신고하고 검증하는 사용자를 지칭한다.

### 2.2.1.1 시나리오 : 신고



- (1) 일반 사용자는 온라인에서 저작물 불법 유통을 발견한다.
- (2) 웹 사이트에 접속해 로그인 한 뒤, 해당 저작물을 조회한다.
- (3) 존재한다면, Report버튼을 누르고 url과 키워드를 입력해 신고한다.

### 2.2.1.2 시나리오 : 검증



- (1) 일반 사용자는 웹 사이트에 로그인하여 접속한다.
- (2) 검증 탭을 누르고, 다른 사용자가 제출한 신규 신고를 확인한다.
- (3) 해당 url이 저작물의 보호 범위와 일치하는 지 확인한다.
- (4) Yes/No를 눌러 신고의 유효성을 판단한다.

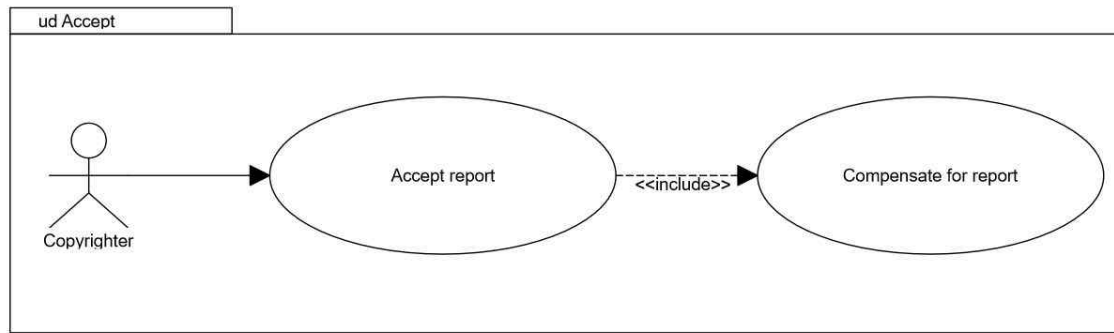
### 2.2.2 시나리오 : 저작권자

저작권자는 저작물의 보호를 위해, 불법 유통에 대한 신고를 받고, 신고에 보상을 부여하는 대상이자, 웹 크롤러를 운영하는 노드 서버의 관리자이다. 저작권자는 시스템을 사용하기 위해 2가지의 설치가 필요하다.

1. 이더리움 지갑 어플리케이션(Metamask)
2. 오픈소스로 제공되는 웹 크롤러

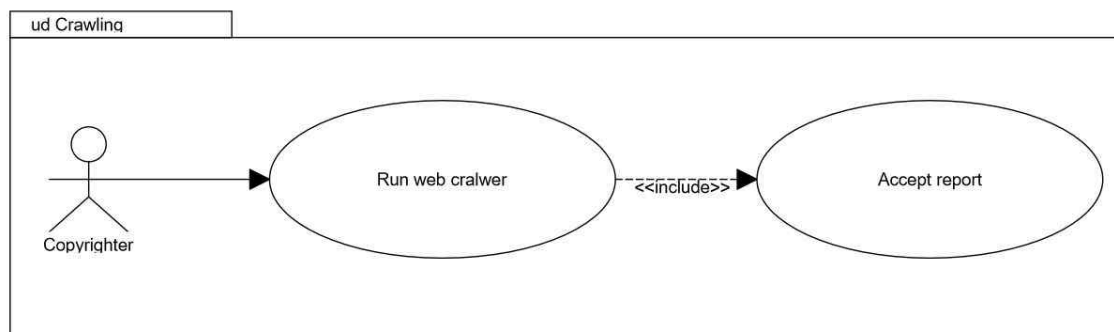
#### 2.2.2.1 시나리오 : 신고 승인 및 보상





- (1) 저작권자는 웹 사이트를 통해 검증이 완료된 신고를 받는다.
- (2) 신고의 링크에 접속해 해당 url이 원하는 정보인지 판단한다.
- (3) Yes/No를 눌러 해당 신고를 처리한다. 이 때, 신고를 수락할 경우 Metamask 어플리케이션을 통해, 신고자와 검증자에게 보상을 전송한다.

### 2.2.2.3 시나리오 : 웹 크롤러 운영



- (1) 저작권자는 오픈소스로 제공된 웹 크롤러를 실행한다.
- (2) 웹 크롤러는 중앙서버로 저작권자가 수락한 신고에 대한 내역을 받는다.
- (3) 웹 크롤러는 이 정보를 바탕으로 신고 내역으로부터 찾을 수 있는 다른 저작물 불법 유통을 찾는다.

## 2.3. 사용자 특성 (User characteristics)

시스템에는 일반 사용자인 신고자(검증자를 포함), 전체 시스템 관리자와 저작권자에 해당하는 세 종류의 사용자가 존재한다.

- 신고자는 Url의 정의를 알고, 웹 어플리케이션에 등록된 보호 요청에 대한 조건문을 원활하게 읽을 수 있는 능력을 갖추고 있으며, 일반 PC에서 인터넷을 사용하고 복사 붙여넣기 등 기본적인 컴퓨터의 기능을 사용할 수 있다고 가정한다. 또한, 퍼블릭 블록체인에 대한 이해가 없더라도, 복잡하지 않은 절차라면 이해할 수 있고, 자신의 비밀키를 보관할 수 있다고 가정한다.
- 저작권자는 저작물을 보호 요청한 대상으로 신고 내역을 받아 웹 크롤링을 수행할 노드서버의 관리자이고, 이더리움 네트워크를 통해 보상을 지급할 주체가

다. 제공되는 오픈소스 웹 크롤러와 서버 관리 프로그램을 절차에 따라 설치할 컴퓨터 지식이 있다고 가정한다. 전체 시스템 전반에 대한 이해가 없더라도, 보상을 준 신고내역에 관하여 지속적인 탐색을 수행하는 노드입장에서의 시스템을 이해하고, 활용할 수 있다고 가정한다. 또한, 때때로 관리자들 중 일부는 기존에 사용하던 시스템이나 디지털 워터마크, 디지털 포렌식 등 다양한 기술을 접합시켜 활용하기를 원한다고 가정한다.

- 전체 시스템 관리자는 충분히 시스템에 대한 정보를 습득하고, 시스템 전반의 과정과 이해를 갖춘 사람으로 한정한다. 컴퓨터 공학을 전공하는 사람으로 한정하고, 본 프로젝트를 설계하고 개발한 사람으로 한정한다.

## 2.4. 제약 조건 (Constraints)

본 프로젝트에서는 다음과 같은 항목을 준수하여 개발해야 한다.

- 통신에서 브라우저와 서버간의 퍼블릭 블록체인의 개인키에 대한 내용은 포함해선 안 된다.
- 지갑 어플리케이션을 사용하는 경우, Metamask로 한정한다.
- 서버의 데이터베이스에 사용자의 개인키를 저장하지 않는다.
- 노드가 사용하는 소프트웨어의 개발에서 추후의 확장성을 고려하여 설계하고, 커스터마이징이 용이하도록 설계한다.
- 웹 크롤러는 각 인터넷 사업자의 robot.txt로 표현되는 웹 크롤링 윤리를 준수하여 탐색한다.

## 2.5. 가정과 의존성 (Assumptions and dependencies)

본 문서에서 제시하는 시스템은 상용의 목적이 아닌 연습용 프로젝트로, 사용자의 수수료 비용 지불, 비즈니스 모델의 변경으로 인한 재구축을 고려하지 않고 설계된다.

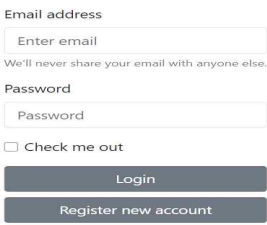
보상 및 스마트 컨트랙트의 수행을 위해 사용자가 소비하는 비용이 이더리움 네트워크의 수수료에 종속적이고 문서의 작성일 기준으로 이 수수료가 비싸기 때문에 실제 운용에서는 다른 네트워크를 고려할 수 있다. 졸업 프로젝트를 위한 소프트웨어의 테스트와 운영은 이더리움 테스트 네트워크에서 진행한다.

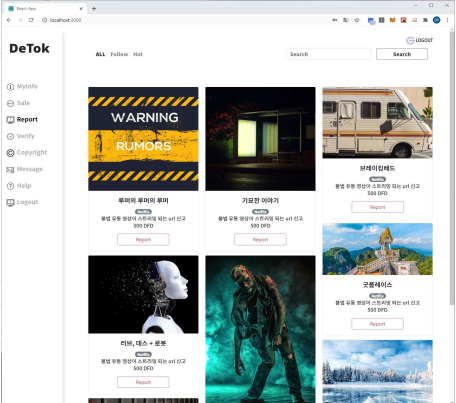
### 3. 상세 요구사항 (Requirement Specification)

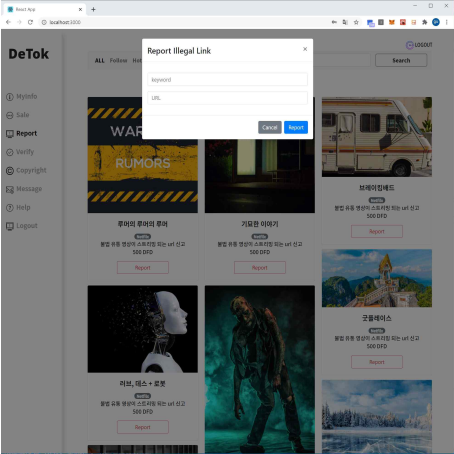
#### 3.1. 인터페이스 요구사항 (Interface Requirements)

##### 3.1.1 사용자 인터페이스 (User Interfaces)

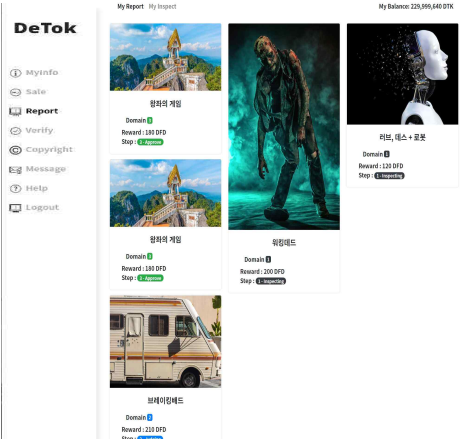
사용자는 웹 기반의 어플리케이션을 제공 받는다.

요구사항 번호		UIR-001
요구사항 명칭		사용자 로그인 인터페이스
상세 설명	정의	웹 어플리케이션 로그인 인터페이스
	세부 내용	<p>◎ 아이디와 패스워드를 입력하여 로그인 할 수 있도록 화면을 구성</p> <ul style="list-style-type: none"> <li>- 사용자가 손쉽게 이해할 수 있도록, 간결하게 디자인</li> <li>- 이메일 형식의 아이디와 패스워드를 입력받는 입력창 두 개와 로그인, 회원가입 두 버튼으로 구성</li> </ul> <p>◎ 저작권자일 경우 로그인 절차 생략</p> <ul style="list-style-type: none"> <li>- 저작권자일 경우, 가상화페 지갑(메타마스크)를 통해 주입된 정보를 통해 바로 로그인</li> </ul>
화면 설계서		

요구사항 번호		UIR-002
요구사항 명칭		신고 저작물 탐색 인터페이스
상세 설명	정의	등록저작물 탐색 인터페이스
	세부 내용	<p>◎ 신고할 저작물을 탐색하는 화면 구성</p> <ul style="list-style-type: none"> <li>- 왼쪽에는 다른 메뉴로 갈 수 있는 탭 화면이 있도록 구성</li> <li>- 각 저작물의 사진과 신고에 대한 세부 정보를 포함하여, 한 개의 박스로 표시</li> <li>- 검색창은 하나로 통합하여, 한 입력창에서 저작권자의 명칭이나 저작물의 이름으로 탐색할 수 있도록 표시</li> </ul>
화면 설계서		

요구사항 번호		UIR-003
요구사항 명칭		신고 인터페이스
상세 설명	정의	불법 유통 신고 인터페이스
	세부 내용	<p>◎ 선택한 저작물에 대한 신고 화면 구성</p> <ul style="list-style-type: none"> <li>- 키워드와 url을 입력하여 신고할 수 있도록 표시</li> <li>- 신고 창을 제외한 화면은 음영 처리로 표시</li> <li>- 신고 창의 위치는 상단 중앙 또는 전체 중앙에 배치</li> </ul>
화면 설계서		

요구사항 번호		UIR-004
요구사항 명칭		신규 신고 검증
상세 설명	정의	웹 어플리케이션 로그인
	세부 내용	<p>◎ 다른 사용자의 신규 신고에 대한 검증 화면 구성</p> <ul style="list-style-type: none"> <li>- 새로운 신고 발생 시 검증을 위해, 신고에 대한 내용이 출력</li> <li>- Yes/Pass/No 세 가지 결정이 가능하도록 구성</li> <li>- 화면을 간결하게 구성하여 사용자가 쉽게 접근할 수 있도록 표시</li> </ul> <p>◎ 상단에 3가지 종류의 탭을 구성</p> <ul style="list-style-type: none"> <li>- Random, Follow, Select로 구성하여, 검증할 신고의 저작물을 한정</li> </ul>
화면 설계서		

요구사항 번호		UIR-005
요구사항 명칭		사용자 로그인 인터페이스
상세 설명	정의	웹 어플리케이션 로그인
	세부 내용	◎ 선택한 저작물에 대한 신고 기능 구현 ① 신고 대상 저작물 선택 ② 신고할 URL과 검색한 키워드 입력 ③ 신고 접수 및 결과 출력
화면 설계서		

### 3.1.2 통신 인터페이스 (Communications Interfaces)

요구사항 번호	CIR-001
요구사항 명칭	클라이언트와 중앙서버 통신 인터페이스
내용	각 클라이언트에서 중앙서버에 웹 브라우저를 통해 원하는 자원의 정보를 생성, 수정, 삭제, 조회 요청 중앙서버에서 각 클라이언트의 요청을 수행하고 자원을 제공
데이터 형식	Json 형식
명령 형식	HTTP 프로토콜을 통한 request/response
입력/출력	클라이언트/중앙서버
시간/속도	최소 10Mbps 이상

요구사항 번호	CIR-002
요구사항 명칭	노드서버와 중앙서버 통신 인터페이스
내용	각 노드서버는 중앙서버에 자신이 수락한 신고에 대한 내역을 요청 중앙서버는 각 노드서버에게 내역을 전송
데이터 형식	Json 형식
명령 형식	HTTP 프로토콜을 통한 request/response
입력/출력	노드서버/중앙서버
시간/속도	최소 10Mbps 이상

### 3.2. 기능적 요구사항 (Functional Requirements)

요구사항 번호		FR-001
요구사항 명칭		회원 가입 기능
상세 설명	정의	일반 사용자 회원 가입
	세부 내용	◎ 홈페이지 접속 시 회원 가입 메뉴 구축 - 사용할 ID/PW를 입력받아 회원가입 기능 구현 ◎ 계정 규칙 준수 - ID는 email양식만 허용하고, 패스워드는 10자 이상만 허용
입력		ID, Password
출력		계정 생성 승인, 거부
사전 조건		None
사후 조건		None

요구사항 번호		FR-002
요구사항 명칭		로그인 기능
상세 설명	정의	일반 사용자 로그인
	세부 내용	◎ 홈페이지 접속 시 로그인 기능 구축 - ID/Password를 입력 받아 등록된 ID/Password와 일치하는 여부를 판단 - 로그인 성공 시, 브라우저의 종료 버튼을 누르거나, 로그아웃 버튼을 누르기 전까지 로그인 상태를 유지
입력		ID, Password
출력		로그인 성공 여부
사전 조건		None
사후 조건		None

요구사항 번호		FR-003
요구사항 명칭		저작물 조회 및 검색 기능
상세 설명	정의	신고의 대상(시스템에 등록된 저작물) 조회 및 검색
	세부 내용	◎ 저작물의 명칭 및 저작권자의 명칭으로 조회 구현 - 검색 대상: 시스템에 등록된 저작물 - 검색 규칙 : 검색어가 포함된 명칭이나 검색어가 포함된 저작권자의 명칭으로 데이터베이스에 등록된 저작물의 정보를 출력한다.
입력		검색어
출력		저작물의 명칭, 보호 요청 상세 내역, 보상 액수, 저작권자의 명칭
사전 조건		로그인 완료
사후 조건		브라우저에 로그인 정보 저장

요구사항 번호		FR-004
요구사항 명칭		불법 유통 신고 기능
상세 설명	정의	등록된 저작물에 대한 불법 유통 URL 신고
	세부 내용	◎ 선택한 저작물에 대한 신고 기능 구현 ① 신고 대상 저작물 선택 ② 신고할 URL과 검색한 키워드 입력 ③ 신고 접수 및 결과 출력
입력		Url, 키워드
출력		신고 접수 결과
사전 조건		로그인 완료
사후 조건		신고 거부 규칙 생성 및 신고 접수 단계

요구사항 번호		FR-005
요구사항 명칭		임시 신고 거부 규칙 생성 기능
상세 설명	정의	임시적인 불법 유통의 신고 URL 규칙 생성
	세부 내용	◎ 신고가 접수된 시점에서 중복된 신고 또는 접수된 신고를 통해 웹 크롤러가 탐색할 가능성이 있는 URL에 대한 신고의 접수를 임시적으로 차단 ① 신고 접수 이후 해당 URL의 host name과 동일한 host name의 신고를 거부하는 규칙을 생성 ② 신고가 거부되거나, 웹 크롤러가 해당 신고내역을 통한 탐색을 완료한 뒤 규칙 제거
입력		URL
출력		신고 거부 규칙
사전 조건		신고 접수
사후 조건		생성 규칙 제거

요구사항 번호		FR-006
요구사항 명칭		신고 규칙 생성 기능
상세 설명	정의	신고 규칙 생성 기능
	세부 내용	<p>◎ 보상한 신고의 정보를 통해 웹 크롤러가 탐색을 완료한 뒤 더 이상 신고를 받을 필요 없는 URL에 대하여 신고를 거부하는 규칙을 생성</p> <ul style="list-style-type: none"> <li>- 이미 발견한 웹 페이지, 발견된 불법 유통을 목적으로 하는 인터넷 사업자에 대한 신고를 거부</li> </ul> <p>◎ 저작물이 보호받고 싶은 범위를 지정하기 위한 규칙을 생성</p> <ul style="list-style-type: none"> <li>- 국가, 인터넷 사업자와 같은 정해진 범위의 신고만을 받기 위한 규칙 생성</li> </ul>
입력		Host 명칭 또는 URL 또는 IP
출력		신규 규칙
사전 조건		규칙이 적용될 저작물이 등록되어 있음
사후 조건		해당 저작물은 규칙에 반하는 신고를 신고 접수 거부 단계로 이동

요구사항 번호		FR-007
요구사항 명칭		신고 접수 필터링 기능
상세 설명	정의	신고와 신고 규칙 비교 및 접수 가능 판단
	세부 내용	<p>◎ 저작물에 설정된 규칙에 의해 신고를 필터링하여 접수할 수 있도록 규칙과 비교하는 기능을 구현</p> <ul style="list-style-type: none"> <li>- 규칙에 맞추어 신고접수 단계에서 필터링</li> </ul> <p>◎ 신고 접수가 가능한 저작물인지 판단</p> <ul style="list-style-type: none"> <li>- 신고 접수 개수가 초과된 저작물인 경우 거부</li> </ul>
입력		URL
출력		승인, 거부
사전 조건		신고 상태 : 신고접수 단계
사후 조건		신고의 상태를 검증대기 상태로 이동 또는 신고접수거부 단계로 이동

요구사항 번호		FR-008
요구사항 명칭		투표를 통한 불법 유통 검증 기능
상세 설명	정의	일반 사용자의 불법 유통 검증에 대한 투표
	세부 내용	<p>◎ 검증 대기 상태의 신고에 대하여, 유효성을 불특정 다수가 판단하는 기능을 구현</p> <ul style="list-style-type: none"> <li>- 다수결 투표로 결정</li> <li>- 사용자에게 보호 대상의 보호 범위를 명확하게 전달 받을 수 있는 사용자 인터페이스를 구현</li> </ul>
입력		찬성, 반대
출력		투표 결과
사전 조건		신고의 검증 투표 완료
사후 조건		신고 상태를 판단대기/검증거부 단계로 이동



요구사항 번호		FR-008
요구사항 명칭		검증(일반 사용자) 평점 기능
상세 설명	정의	검증의 결과에 따른 평점 관리
	세부 내용	◎ 일반 사용자에게 평점을 부여 - 올바른 검증에 평점 상승, 잘못된 검증에 평점 하락 - 검증의 행위에 평가치를 부여하여, 일반 사용자가 올바른 행동을 할 수 있도록 유도할 수 있는 절차를 구현
입력		검증 결과
출력		변경된 평점
사전 조건		저작권자의 판단대기 단계의 신고에 대한 수락/거부 또는 검증단계의 거부
사후 조건		평점 변동

요구사항 번호		FR-009
요구사항 명칭		신고 판단
상세 설명	정의	신고 판단
	세부 내용	◎ 신고의 유효성 여부를 저작권자가 판단 - 저작물을 등록할 때 신고 받기를 원하는 범주에 들어있는지 판단하여, 수락 또는 거부
입력		판단 대기 신고와 판단 내용
출력		
사전 조건		판단 대기 상태의 신고
사후 조건		신고를 판단수락/ 판단거부 상태로 이동

요구사항 번호		FR-010
요구사항 명칭		내 신고 상태 조회
상세 설명	정의	로그인된 사용자의 신고 상태 조회
	세부 내용	◎ 사용자의 제출한 신고와 검증한 신고의 상태 조회 - 현재 신고의 진행 단계 표시 - 신고에 대한 세부 정보 표시
입력		ID
출력		신고 상태
사전 조건		로그인, 신고 접수 또는 검증
사후 조건		None

요구사항 번호		FR-011
요구사항 명칭		저작물 등록 기능
상세 설명	정의	저작물의 등록 기능
	세부 내용	◎ 신고 받을 저작물을 등록 - 저작권자는 보상, 보호 범위, 명칭, 이미지를 입력
입력		보상, 보호범위, 명칭, 이미지, 정보
출력		승인/거부
사전 조건		저작권자로 등록
사후 조건		저작물 등록

요구사항 번호		FR-012
요구사항 명칭		저작권자 등록 기능
상세 설명	정의	저작권자 등록 기능
	세부 내용	◎ 지갑 어플리케이션(Metamask)를 통해 저작권자 등록 기능 구현 - 지갑 어플리케이션을 통해 저작권자를 인증할 정보를 받고, 저작권자를 나타낼 정보를 입력받아 저작권자로 등록 - 복잡한 절차 없이 지갑 어플리케이션의 설치만으로 회원가입 절차 간소화 - 저작권자의 이더리움 주소를 중앙서버에 저장
입력		명칭, 이더리움 주소
출력		승인/거부
사전 조건		메타마스크 설치
사후 조건		메타마스크에 로그인 되어 있을 시, 로그인 단계를 건너뛰고 저작권자로 바로 로그인

요구사항 번호		FR-013
요구사항 명칭		저작권자 로그인 인증 기능
상세 설명	정의	저작권자 로그인 인증 기능
	세부 내용	◎ 지갑 어플리케이션(Metamask)를 통해 웹 어플리케이션에 저작권자의 로그인 인증 구현 - 저작권자로 등록되어 있는 경우 웹 어플리케이션에서 로그인 절차 없이 바로 접속
입력		메타마스크에서 주입되는 주소 정보
출력		로그인 인증 정보
사전 조건		메타마스크 설치 및 저작권자 등록
사후 조건		저작권자 탭 활성화

요구사항 번호		FR-014
요구사항 명칭		보상내역 확인 기능
상세 설명	정의	보상내역 확인 기능
	세부 내용	<p>◎ 이더리움 네트워크의 블록을 읽어, 수락한 신고의 보상내역을 확인</p> <ul style="list-style-type: none"> <li>- 저작권자가 적절하게 보상을 주었는지 확인하고, 저작권자의 신규 신고 내역에서 삭제</li> <li>- 저작권자의 노드서버가 요청하면 보내줄 신고 내역에 추가</li> </ul>
입력		신고 수락
출력		보상 내역
사전 조건		저작권자가 판단 대기 상태인 신고를 수락할 경우
사후 조건		신고를 내역 전송 대기 단계로 이동

요구사항 번호		FR-015
요구사항 명칭		신고 보상 기능
상세 설명	정의	일반 사용자 로그인
	세부 내용	<p>◎ 이더리움 네트워크에서 ERC20을 통해 신고자와 검증자에게 보상</p> <ul style="list-style-type: none"> <li>- 신고자에게 설정한 보상액의 절반을 보상하고, 검증자들에게 평점에 비례하여 나머지를 나눔</li> <li>- 판단에서 수락버튼을 누를 경우 바로 보상 기능을 실행하여, 저작권자에게 이해하기 쉬운 인터페이스를 제공</li> </ul>
입력		수락 신고, 신고자 주소, 검증자 주소, 검증자 평점
출력		보상 결과
사전 조건		판단수락 단계 신고
사후 조건		None

요구사항 번호		FR-016
요구사항 명칭		접수가능 신고 개수 관리 기능
상세 설명	정의	저작물의 접수가능 신고 개수 설정
	세부 내용	<p>◎ 저작권자가 신고를 처리하지 않고 정보만을 취득하는 것을 막기 위해 판단대기와 상태의 신고의 최대치를 결정</p> <ul style="list-style-type: none"> <li>- 판단대기 신고를 거부할 경우 최대치 감소</li> <li>- 판단대기 신고를 수락할 경우 최대치 증가</li> <li>- 판단대기 상태의 신고 최대치와 일치할 경우, 신고 접수와 검증을 일시적 차단</li> </ul>
입력		현재 접수가능 신고 개수, 신규 판단 내역
출력		변경된 접수가능 신고 개수
사전 조건		저작물 등록
사후 조건		변경된 개수 적용

**데이터 요구사항**

서버와 클라이언트, 노드는 HTTP프로토콜을 이용하여 통신한다. 데이터는 JSON 형식으로 전송된다.

**<업데이트 예정>**

**데이터 구조 명세**

데이터 분류	명칭	변수
회원 인증	req_login	
회원 가입	req_register	
저작물 조회	req_search_copyright	
신고	req_report	
검증	req_verification	
내 신고 상태 조회	req_status	
검증 대상 조회	req_search_report_1	
판단대기 신고 조회	req_search_report_2	
From	클라이언트	
To	중앙서버	

데이터 분류	명칭	변수
회원 인증	res_login	
회원 가입	res_register	
저작물 조회	res_search_copyright	
신고	res_report	
검증	res_verification	
내 신고 상태 조회	res_status	
검증 대상 조회	res_search_report_1	
판단대기 신고 조회	res_search_report_2	
From	중앙서버	
To	클라이언트	

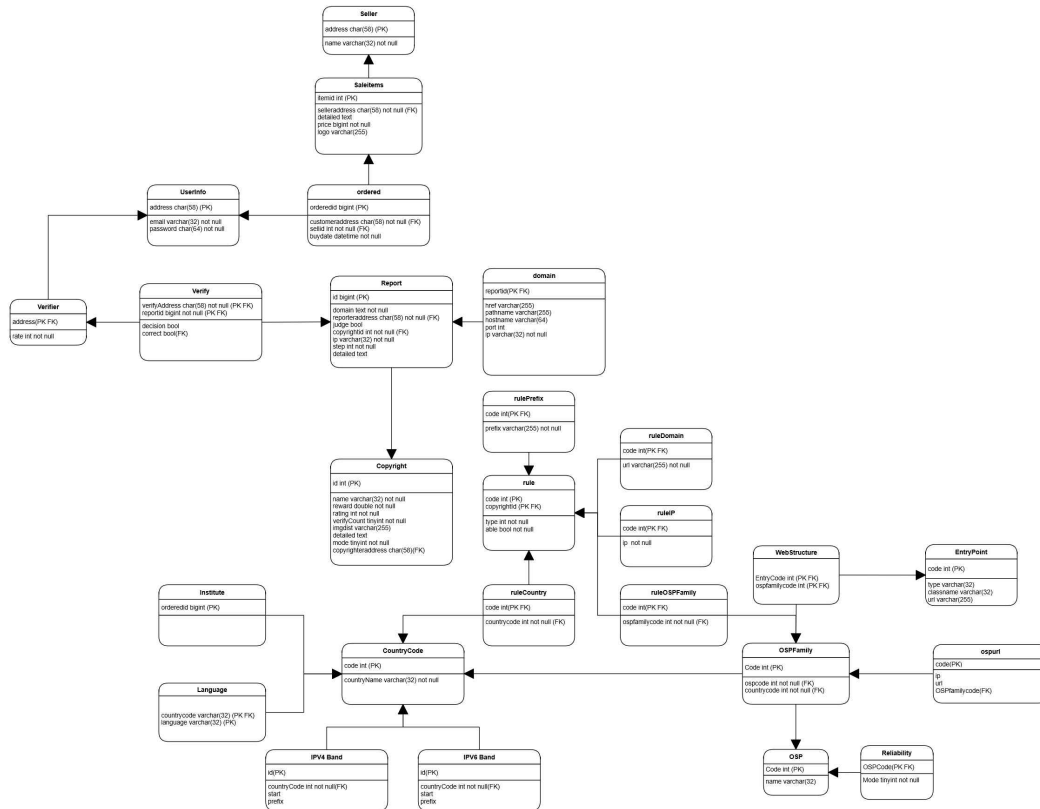
데이터 분류	명칭	변수
신규 신고내역 조회		
규칙 생성		
From	노드서버	
To	중앙서버	

데이터 분류	명칭	변수
신규 신고내역 조회		
규칙 생성		
From	중앙서버	
To	노드서버	

데이터 분류	명칭	변수
보상내역 조회		
From	이더리움 네트워크	
To	중앙서버	

데이터 분류	명칭	변수
보상		
From	클라이언트	
To	이더리움 네트워크	

## 논리적 데이터베이스



## 4. 추가 정보(Supporting Information)

### 4.1 문서이력

날짜	버전	설명	작성자
2020.09.25	0.1	스타일 작업 및 개요	서왕규
2020.10.05	0.2	종합 기술 작성	서왕규
2020.10.09	0.2.1	종합 기술 시나리오 추가	서왕규
2020.10.14	0.3	상세 요구 사항 작성	서왕규
2020.10.21	0.4	상세 요구 사항 사용자 인터페이스 추가	서왕규