

한이음 공모전 2017 개 발 보 고 서

2017. 10. 17

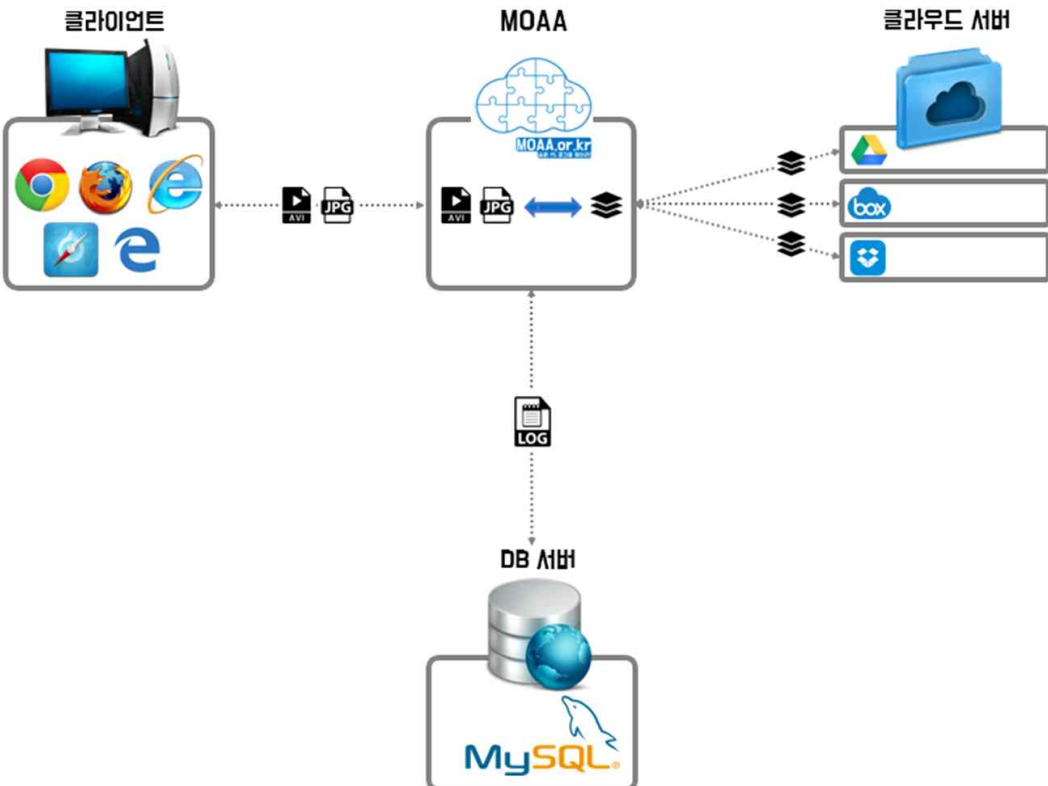
프로젝트명	국문	숨은 1% 공간을 찾아라!
	영문	Find hidden 1% space!
작 품 명	MOAA (Management of clouds, the One Above All)	
신 청 자	서울과학기술대학교 / 조익환	

요 약 본

팀 정보

팀 명	Mission 99			
팀 원	이 름	소 속	부서/학과	직위/학년
멘 토	정상미	안랩	SW QA	책임연구원
지도교수	이윤희	서울과학기술대학교	ITM 전공	주임교수
멘티 1(팀장)	조익환	서울과학기술대학교	ITM 전공	3학년
멘티 2	위선민	서울과학기술대학교	ITM 전공	3학년
멘티 3	이항복	서울과학기술대학교	ITM 전공	3학년
멘티 4	조민재	서울과학기술대학교	ITM 전공	3학년
멘티 5	황인원	서울과학기술대학교	ITM 전공	2학년



작품 정보		
프로젝트명	국문	숨은 1% 공간을 찾아라!
	영문	Find hidden 1% space!
작품명	MOAA (Management of clouds, the One Above All)	
작품 소개	<p>본 작품 MOAA는 클라우드 서비스와 Zip4J API를 이용하여 구현한 클라우드 분할관리시스템이다. 본 프로젝트의 작품은 기존의 클라우드 저장 공간을 병합하여 효율적으로 관리하고, 마치 하나의 저장 공간과 같은 개념으로 인식, 활용할 수 있도록 한다. 또한 파일 분산저장을 통해 보안의 새로운 방향을 제시한다.</p>	
작품 구성도		
작품의 개발배경 및 필요성	<p>기존 낭비되어왔던 독립 클라우드 저장 공간들의 효율적 활용과 체계적인 관리의 필요성이 있다고 생각했다. 또한 기존 서비스들의 보안 취약점이었던 중앙 집중형 저장 방식을 본 프로젝트의 '클라우드 통합연동과 분산저장'을 통하여 해결하고 싶어서 본 작품을 만들게 되었다.</p>	
작품의 특징점	<p>통합 관리 플랫폼을 제공함으로써 손쉬운 저장 공간 관리가 가능하고, 다양한 클라우드 서비스 연동을 통해 분할 업로드 시킴으로써 데이터 유지의 안정성을 보장한다. 또한 대시보드를 통해 사용자 편의성을 향상시키고, 분할 업로드/다운로드 과정 중 해쉬 함수와 로그파일 생성을 활용, 전체적인 시스템의 성능을 높인다.</p>	

<p>작품 기능</p>	<p>웹 홈페이지는 회원가입, 로그인 등의 기본적인 프로필 관리 기능을 지원한다. 또한 사용친화적인 대시보드와 클라우드 활성화/비활성화, 클라우드 추가/제외 기능을 통해 사용자의 손쉬운 클라우드 관리를 보장한다.</p> <p>SHA-1 해쉬함수와 로그파일 생성을 통해 사용자 파일 업로드/다운로드/삭제 측면에서 성능을 향상시켰고, 시스템 상에 연동된 클라우드들 뿐만 아니라 시스템 전체의 가용 저장 공간과 현 사용량을 단번에 파악할 수 있는 뷰어(Viewer)기능 역시 지원한다.</p>
<p>작품의 기대효과 및 활용분야</p>	<p>본 작품을 이용하여 서비스의 유연성과 확장성을 보장할 수 있고(분할 저장방식), 커스터마이징 환경 조성으로 다양한 고객의 니즈를 충족시킬 수 있다. 효율적인 저장 공간 활용과 손쉬운 관리를 할 수 있고, 고객 데이터의 높은 안정성과 보안성을 보장할 수 있다. 개인뿐만 아니라 기업, 국가차원의 중요 시스템 및 파일 백업에 사용가능하다.</p>

본 문

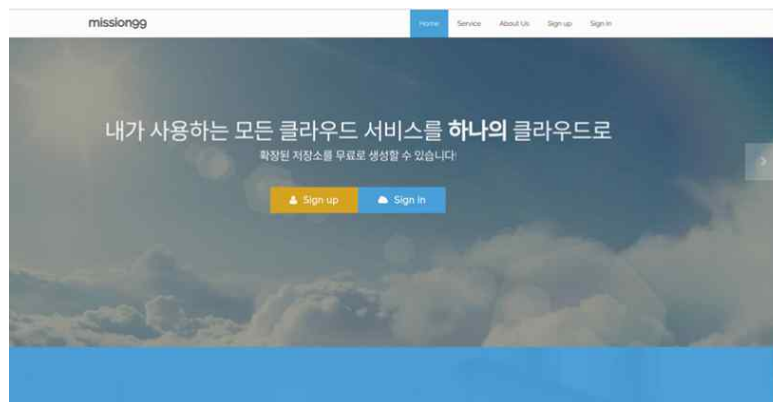
I. 작품 개요

※ 평가항목 : 기획력 (필요성, 차별성)

1. 작품 소개

○ Secure Cloud Management System

- 개인뿐만 아니라 기업, 국가차원에서 클라우드 시스템의 수요가 증가함에 따라 여러 클라우드 서비스들을 체계적이고 효율적으로 관리할 수 있는 시스템.
- 하나의 파일을 여러 개의 패킷으로 분할 저장함으로써, 사용자 데이터의 안정성을 보장함.
- 개별 클라우드에서 낭비되어왔던 가용공간들을 통합 시스템을 통해 관리함으로써, 사용자의 높은 이용효율을 보장함.



2. 작품의 개발 배경 및 필요성

○ 기존 클라우드 서비스의 취약점 보완

본 프로젝트의 파일분할/분산저장 시스템은 각각의 독립 클라우드 서비스들을 연동하여 분할 저장시킴으로써, 하나의 서버가 공격당하더라도 파일의 일부만 공격당하기 때문에 공격자는 파일 전체에 대한 정보를 획득할 수 없다는 장점이 있다.

○ 저장용량의 효율적인 통합관리

오픈 클라우드의 저장 공간들을 하나의 통합 시스템으로 병합하여 가용 저장 공간을 확장한다. 또한 통합 공간을 보다 안전하게 관리하고 사용자 편의성을 보장한다.

- 체계적인 클라우드 시스템관리의 필요

본 프로젝트는 클라우드 저장 공간의 병합 외에도 각 클라우드 서비스의 이용 현황을 손쉽게 관리할 수 있도록 한다. 아울러 대시보드의 뷰어(Viewer)기능을 통해 사용자는 클라우드 공간을 체계적으로 관리할 수 있다.

3. 작품의 특징 및 장점

- 클라우드 통합관리

- Open API를 이용하여 사용자가 등록한 기존 클라우드 서비스와 동기화가 가능토록 하여 서비스의 동시 제어를 보장한다.
- 대시보드와 같은 사용친화적인 웹 페이지를 통해 사용자의 높은 접근성을 보장한다.

- 데이터의 안정성 보장

- 패킷분할을 통해 사용자의 파일이 기존에 등록된 각 클라우드 서비스에 분산 저장된다.

- 대시보드를 통한 사용자 편의성 향상

- 액세스 토큰(Access Token)을 이용하여 사용자는 기존 사용하고 있던 클라우드 서비스에서 제공하는 드라이브를 대시보드 상에서 쉽게 등록하고 삭제할 수 있다.
- 사용자는 등록되어 있는 각각의 클라우드 서비스를 ‘내 드라이브’ 기능을 통해 통합관리(예: 클라우드 활성화/비활성화)가 가능하다.

- S/W 추가설치 불필요

- 서비스를 이용하기 위해 추가적인 소프트웨어 설치가 불필요하다. (사용자 요청에 따른 시스템의 모든 동작은 MOAA 서버 내의 JSP를 거쳐 실행)

- 해쉬함수를 이용한 성능 튜닝

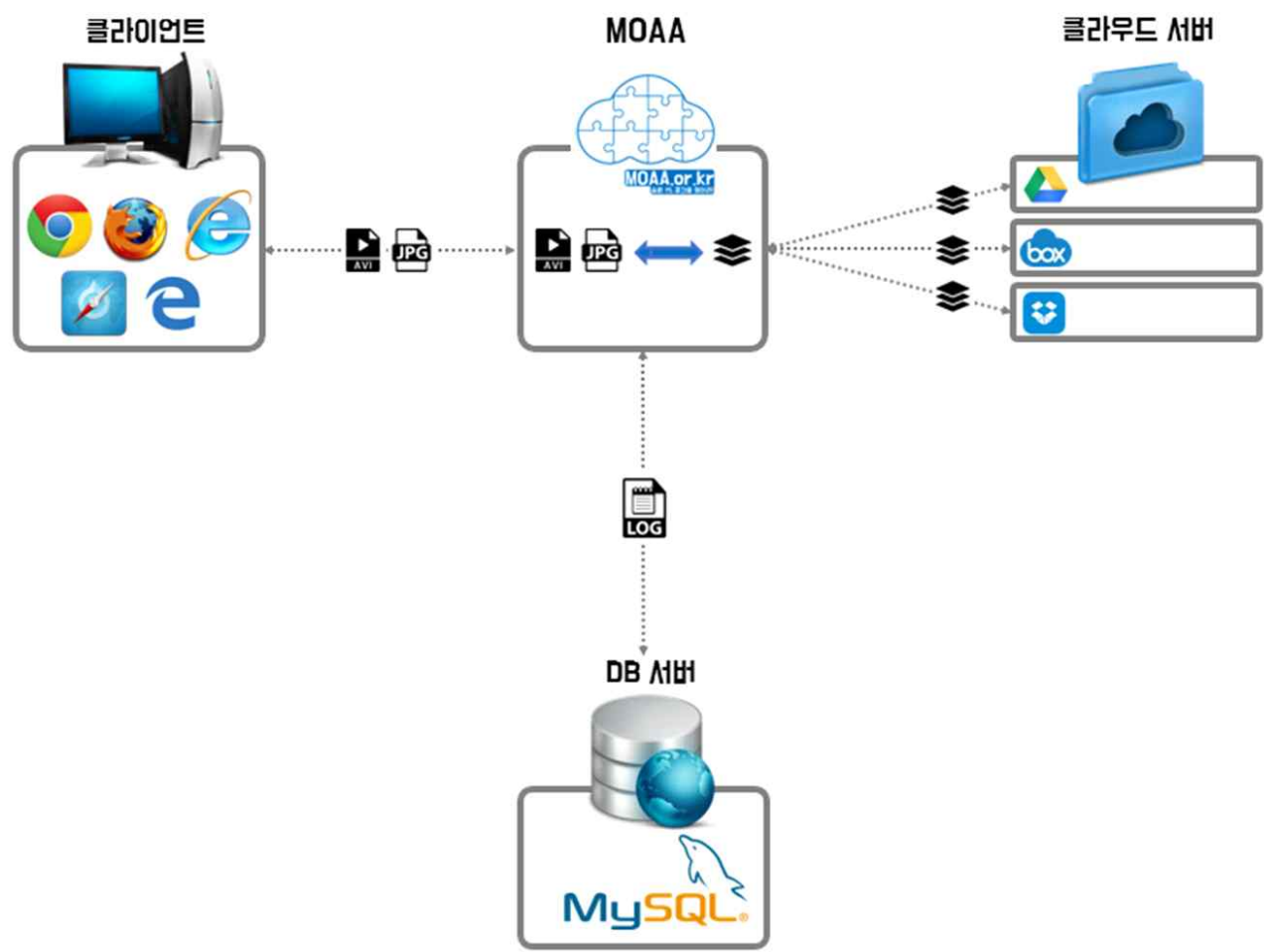
- 파일시스템 구축 시 해쉬함수를 이용함으로써 디렉토리 구조 및 사용자 도메인의 중복을 방지한다.
- 로그파일을 생성함으로써 업로드/ 다운로드/ 삭제시 필요한 데이터에 접근이 가능하고, 전체적인 시스템의 성능을 향상시킨다.

II. 작품 내용

※ 평가항목 : 기술력 (기능구체성, 난이도, 완성도)

1. 작품 구성도

○ 시스템 구성도

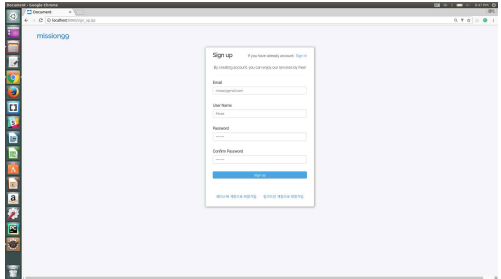
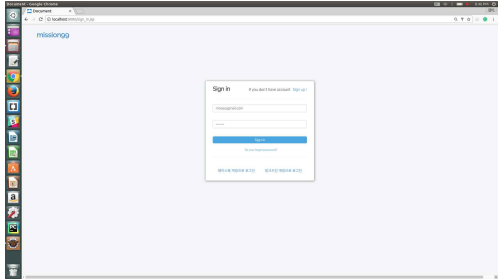
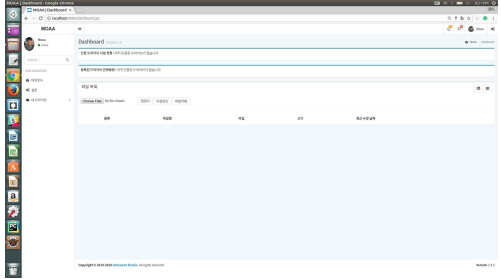
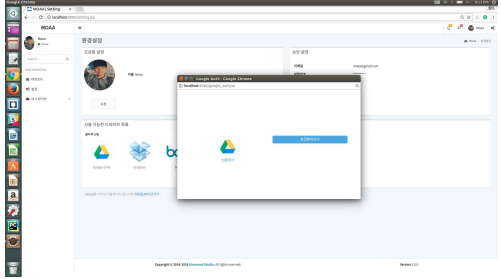
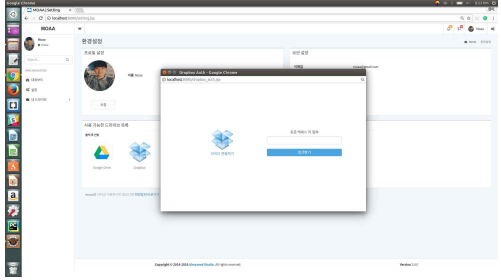
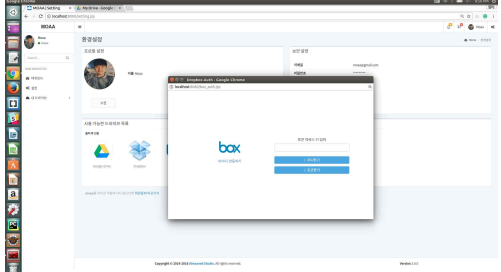


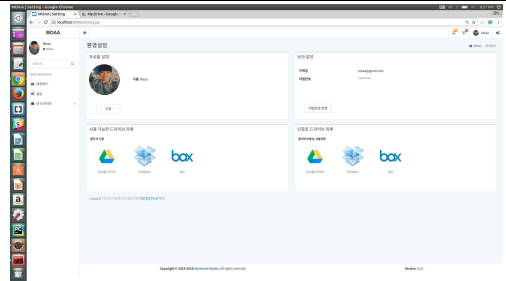
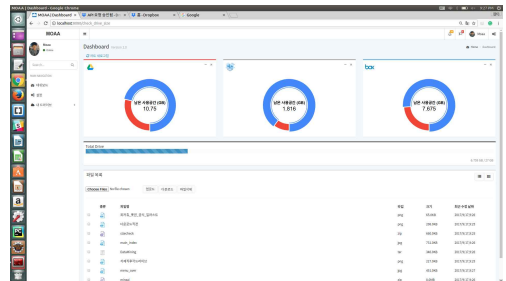
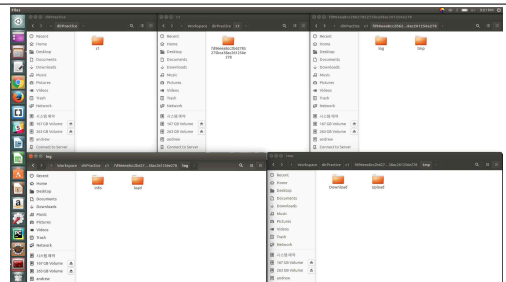
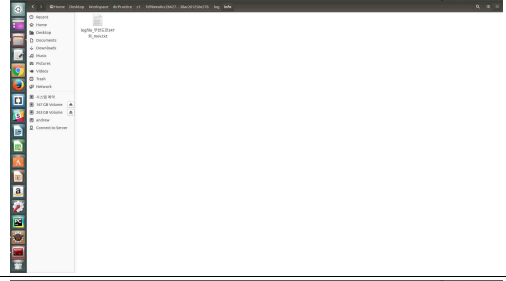
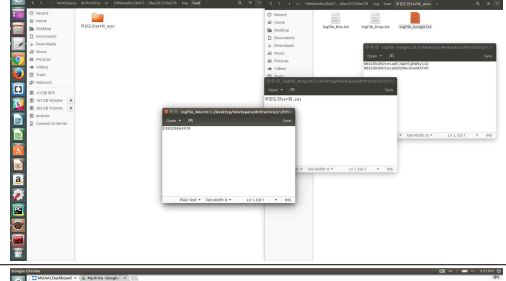
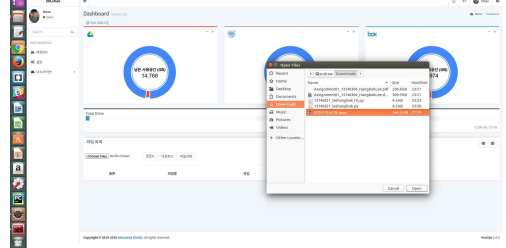

2. 작품 기능

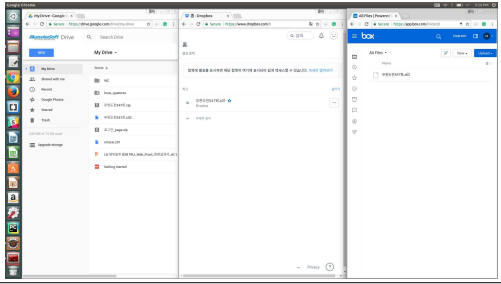
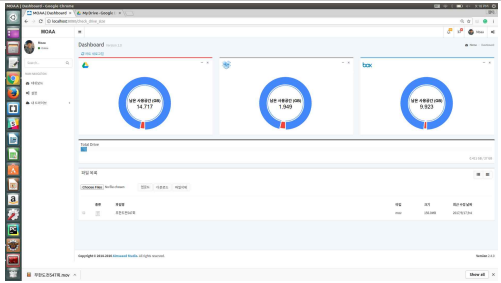
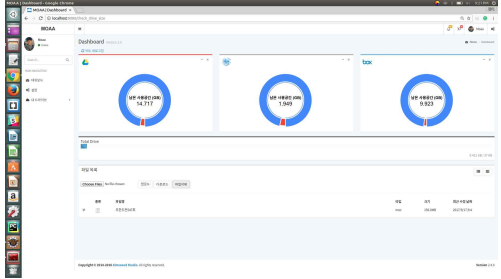
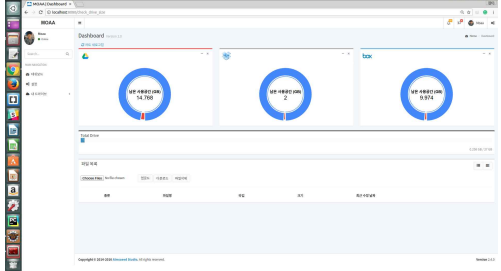
2-1. 전체 기능 목록

구분	기능	설명	현재진척도(%)
S/W	회원가입	MOAA에 회원가입할 수 있는 기능을 제공함	100%
	로그인	이메일과 비밀번호를 이용하여 로그인할 수 있는 기능을 제공함	100%
	프로필 관리	로그인 후 닉네임, 비밀번호 변경, 사진 등록 등 프로필 관리 기능을 제공함	100%
	회원탈퇴	회원탈퇴할 수 있는 기능을 제공함	100%
	대시보드	사용자 파일의 통합 관리기능을 제공함	100%
	클라우드 연동	클라우드 서비스를 시스템에 연동할 수 있는 기능을 제공함	90%
	클라우드 활성화	시스템에 등록된 클라우드 서비스를 활성화시킬 수 있는 기능을 제공함	100%
	클라우드 비활성화	시스템에 등록된 클라우드 서비스를 비활성화시킬 수 있는 기능을 제공함	100%
	Access Token Refresh	기간이 만료된 클라우드 서비스의 Access Token을 자동으로 Refresh시키는 기능을 제공함	70%
	독립 클라우드 사용/가용 공간 뷰어(Viewer)	개별 클라우드 서비스 사용/가용공간을 표시할 수 있는 기능을 제공함	100%
	시스템 총 사용/가용공간 뷰어(Viewer)	시스템(MOAA)의 총 사용/가용공간을 표시할 수 있는 기능을 제공함	100%
	Hashed User Domain	SHA-1 해쉬 함수를 활용하여 사용자별로 독립적인 디렉토리를 제공함	100%
	Log File 생성(원본파일)	원본파일의 정보(이름, 크기, 날짜 등)를 기록하는 로그파일 생성 기능을 제공함	100%
	Log File 생성(분할패킷)	분할된 패킷들의 ID와 저장경로를 기록하는 로그파일 생성 기능을 제공함	100%
	파일 분산저장	시스템에 연동된 클라우드 서비스로 분할 패킷들을 분산 저장하는 기능을 제공함	100%
	파일 업로드	HTML POST 폼을 이용하여 파일 업로드 기능을 제공함	100%
	파일 다운로드	JAVA Servlet과 JSP를 이용하여 파일 다운로드 기능을 제공함	100%
	파일 삭제	원본/분할파일 로그를 이용하여 대시보드와 시스템(MOAA), 클라우드 서비스에 저장된 파일들의 동시 삭제 기능을 제공함	100%

2-2. S/W 주요 기능

기능	설명	작품실물사진
회원가입	회원가입할 수 있는 기능을 제공함 (이메일, 이름, 비밀번호, 비밀번호 확인)	
로그인	회원가입시 기입한 이메일과 비밀번호를 이용하여 로그인할 수 있는 기능을 제공함	
대시보드	로그인 후 자동으로 User Domain으로 이동하여, 파일의 통합 관리기능을 제공함	
클라우드 연동 (구글 드라이브/ 드롭박스/ 박스)	클라우드 서비스를 Access Token을 이용하여 시스템에 연동할 수 있는 기능을 제공함	  

클라우드 활성화	시스템에 등록된 클라우드 서비스를 활성화시킬 수 있는 기능을 제공함	
독립 클라우드 사용/ 가용공간 뷰어(Viewer)	개별 클라우드 서비스 사용/가용공간을 표시할 수 있는 기능을 제공함	
시스템 총 사용/ 가용공간 뷰어(Viewer)	시스템(MOAA)의 총 사용/가용공간을 표시할 수 있는 기능을 제공함	
Hshed User Domain	SHA-1 해쉬 함수를 활용하여 사용자 별로 독립적인 디렉토리(Unique User Domain)를 제공함	
Log File 생성(원본파일)	원본파일의 정보(이름, 크기, 날짜 등)를 기록하는 로그파일 생성 기능을 제공함	
Log File 생성(분할파일)	분할된 패킷들의 ID와 저장경로를 기록하는 로그파일 생성 기능을 제공함	
파일 업로드	HTML POST 폼을 이용하여 파일 업로드 기능을 제공함	

파일 분산저장	시스템에 연동된 클라우드 서비스로 분할 패킷들을 분산 저장하는 기능을 제공함	
파일 다운로드	JAVA Servlet과 JSP를 이용하여 파일 다운로드 기능을 제공함	
파일 삭제	원본/분할파일 로그를 이용하여 대시보드와 시스템(MOAA), 클라우드 서비스에 저장된 파일들의 동시 삭제 기능을 제공함	<p data-bbox="1114 837 1241 869"><삭제 전></p>  <p data-bbox="1114 1173 1241 1205"><삭제 후></p> 

3. 주요 적용 기술

○ HTML

HyperText Markup Language. 웹 페이지 구성을 위하여 사용되는 가장 기본적인 프로그래밍 언어이다. HTML은 여러 태그들(예. <>/>)로 구성되어있으며, 각 태그들을 사용하여 원하는 형태의 문서를 만들 수 있다. 또한 여러 문서를 링크시켜 다양한 정보를 손쉽게 검색하여 볼 수 있게 만들어준다.

○ JQuery

엘리먼트(Element)를 다루는 자바스크립트 라이브러리이다. 또한 JQuery는 HTML의 클라이언트(프론트-엔드) 사이드 조작을 단순화 하도록 설계되었는데, 자유 소프트웨어(Free Software)와 오픈 소스의 성질을 둘 다 갖춘 자유-오픈 소프트웨어로써, 소스 코드의 사용/복사/연구/변경/개선권한을 모두 사용자에게 라이선스로 부여한다. JQuery의 문법은 코드 보기, 문서 객체 모델 찾기, 애니메이션 만들기, 이벤트 제어, Ajax 개발을 쉽게 할 수 있도록 디자인되었고, 개발자가 자체적으로 플러그인을 개발할 수 있는 기능 역시 제공한다.

○ Bootstrap

오픈소스기반(Apache 2.0라이선스) 의 HTML5, CSS3 UI 프레임워크이다. HTML5, CSS3, Javascript 로 구성되어 있고, UI를 빠르게 구현해낼 수 있는 컴포넌트들을 제공해 준다.

○ GitHub

깃허브(GitHub)는 분산 버전 관리 툴인 깃(Git)을 사용하는 프로젝트 지원 웹호스팅 서비스로써, 루비 온 레일스로 작성되었다. 깃(Git)이 텍스트 명령어 입력 방식인데 반해, 깃허브는 그래픽 유저 인터페이스(GUI)를 제공하고, Gist와 위키를 각 저장소마다 운영하고 있으며, 깃 저장소를 통해 수정이 가능하다. 본 프로젝트 역시 협업개발을 위하여 GitHub를 활용하였다.

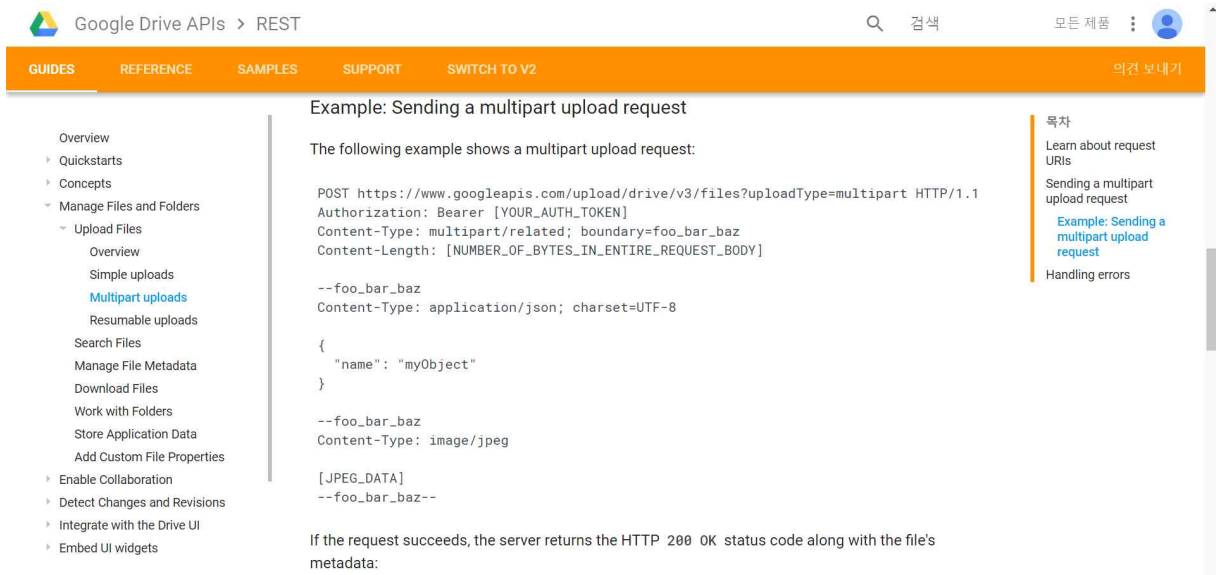
○ JSP

자바서버 페이지(JavaServer Pages)는 HTML내에 자바 코드를 삽입하고 웹 서버에서 동적으로 웹 페이지를 생성하여 웹 브라우저에 돌려주는 언어이다. JSP는 요청들을 자바 서블릿으로 변환한 후 실행하므로 서블릿과 거의 유사하다고 볼 수 있지만, HTML 표준에 따라 작성되므로 웹 디자인 측면에서는 자바 서블릿보다 편리하다고 할 수 있다. 본 프로젝트에서는

구글드라이브, 드롭박스, 박스 등 클라우드 서비스와 Zip4J API를 JSP상에서 구현하였다.

○ Open Cloud API

오픈소스 API형식으로 공개된 3개의 클라우드 서비스(구글의 Google Drive, 드롭박스의 Dropbox, 박스의 Box)를 JSP기반으로 구현하여, MOAA 시스템 상에서 연동될 수 있도록 하였다. 또한, 사용자는 대시보드에서 ‘클라우드 활성화/ 비활성화/ 연동’ 기능을 통하여 손쉬운 클라우드 서비스관리가 가능케 하였다.

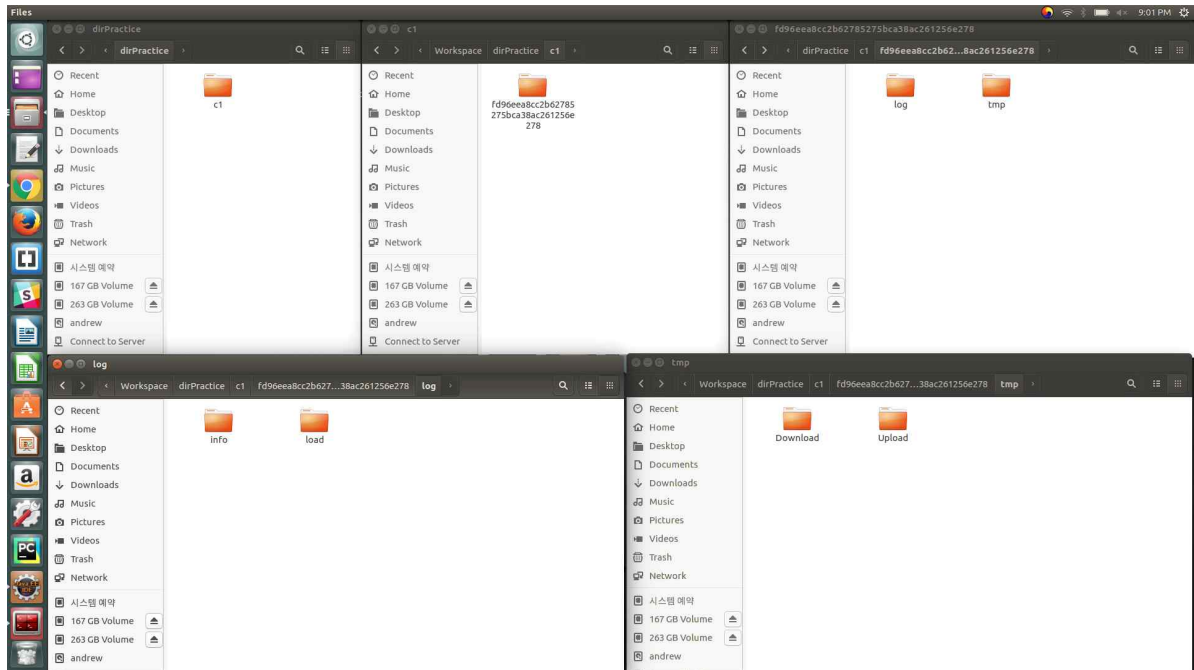


○ Zip4J API

Apache 라이선스가 적용된 API로써, 원하는 용량을 인자값(Parameter)으로 넣어주면 해당 byte만큼 분할 압축해주는 기능을 제공한다. 사용자 파일을 MOAA 시스템을 통해 업로드 시 웹 JSP상에 연동된 Zip4J API를 통하여 패킷형식으로 자동 분할 압축되어 클라우드 서버로 전송되도록 구현하였고 다운로드의 경우에도 마찬가지이다. 또한 API가 구동된 직후, 각기 다른 클라우드 시스템으로 분산 저장된 패킷들의 경로를 기록하는 로그파일을 생성하여 시스템성능을 향상시켰다.

○ SHA-1 해쉬 함수

DB상에 등록된 사용자의 ID(Primary Key)를 인자값(Parameter)으로 하여 산출된 값을 바탕으로 사용자마다 독립적인 디렉토리(User Domain)를 생성토록 하였다. 산출된 해쉬 값의 앞 두 글자는 부모 디렉토리로, 나머지 글자들은 자식 디렉토리로 설정하였다.



4. 작품 개발 환경

구분		상세내용
S/W 개발환경	OS	Windows 10, Linux(CentOS, Ubuntu), Mac OS
	개발환경(IDE)	Java Eclipse
	개발도구	putty, MySQL, Apache Tomcat 8.0
	개발언어	Java, MySQL, HTML, CSS, JavaScript, JQuery, JSP
	기타사항	
프로젝트 관리환경	형상관리	GitHub
	이슈관리	Trello
	의사소통관리	Google Docs, Kakao Talk
	기타사항	

III. 프로젝트 수행 내용

※ 평가항목 : 수행능력 (문제해결능력, 수행충실성)

1. 멘티(참여학생) 업무분장

번호	이름	대학	학과	학년	역할	담당업무
1	조익환	서울과학기술대학교	ITM 전공	3학년	팀장	- 프로젝트 일정관리 및 공유 - 기능 테스트 담당 - Git master branch 관리 - 멘토와 수행상황 공유 - 클라우드 서비스 API 담당 - 진행상황 문서화 및 공유
2	위선민	서울과학기술대학교	ITM 전공	3학년	팀원	- WEB 퍼블리싱 관할 및 공유 - 진행상황 문서화 및 공유 - DB 구축, 연결 및 동기화 - 파일 시스템 프로그래밍
3	이항복	서울과학기술대학교	ITM 전공	3학년	팀원	- WEB 퍼블리싱 담당 - 진행상황 문서화 및 공유 - DB 구축, 연결 및 동기화 - 파일 시스템 프로그래밍
4	조민재	서울과학기술대학교	ITM 전공	3학년	팀원	- 분할 업로드 및 병합 기능 담당 - 진행상황 문서화 및 공유 - 클라우드 서비스 API 담당 - 보고서 및 회의록 총괄
5	황인원	서울과학기술대학교	ITM 전공	2학년	팀원	- 서버 총괄 - DB 구축, 연결 및 동기화 - 진행상황 문서화 및 공유 - 파일 시스템 프로그래밍

2. 프로젝트 수행일정

프로젝트 기간 (한이음 사이트 기준)				2017.04.03. ~ 2017.11.30.									
구분	추진내용	프로젝트 기간											
					4월	5월	6월	7월	8월	9월	10월	11월	12월
계획	프로젝트 아이디어 선정 및 구체화												
분석	예상 클라이언트 니즈 분석												
	업무 요건 정의												
	유사 소프트웨어 제품 분석												
설계	예상 시나리오 설계												
	최 앞단 화면 및 기능 설계												
	DB 설계												
개발	최 앞단 페이지 퍼블리싱												
	분할 업로드 및 병합 기능 개발												
	파일시스템 개발												
	DB 구축 및 동기화												
테스트	기능 통합 및 서버 동기화												
	최종 테스트 및 디버깅												
종료	최종 문서화 작업 및 매뉴얼 작성												

3. 프로젝트 추진 과정에서의 문제점 및 해결방안

3-1. 프로젝트 관리 측면

- 동 시간, 동 장소에서의 협업을 위한 장소의 제약
 - 학교 그룹 스터디 실을 예약하여 회의 및 프로젝트 진행
 - 멘토를 통해 TOZ 스터디 센터 예약
 - 공통 업무는 2~3명씩 소규모로 만나 근처 카페에서 프로젝트 진행
 - 각자의 집에서 본인의 역할을 수행 후 병합
- 업무 및 역할의 중복
 - 업무요건정의서를 작성하여 구글을 통해 공유 - 본인의 역할을 확실히 명시
 - 프로젝트 수행 직후 본인의 산출물을 구글 문서에 추가 및 공유
- 팀원 간 생각 차이
 - 아이디어를 제시한 후 다수결로 채택.
 - 다수결로 채택이 안 될 경우 차후 오류발생의 확률이 적은 방향으로 채택.
- 팀원 간 정보의 차이
 - 메신저를 통한 하루 단위의 대화 및 진행상황 토론.
 - 필요한 공부를 위해 도움이 될 만 한 사이트를 끊임없이 공유.

A	B	C	D	E	F	G	H	I	J	K
한이름 업무요건 정의서										
전체적인 틀(초등)은 이 페이지에서 관리되고 각각 세부사항은 담당 업무자의 사이트에서 관리										
자신의 사이트는 각자 자신이 관리 (자신이 한 일, 피드백 필요한 부분 간단하게라도 정리 부탁)										
1	2									현재 관리자: 조익환, 이형복
댓글 및 자신의 사이트 관리: 전부터 해당										
9	1 시스템	1.4 데이터베이스	MySQL을 사용하여 데이터베이스를 구축한다.							조민재
10		1.5 API 모듈	가장 회원에 대한 정보를 관리하는 데이터베이스를 구축한다.							왕인원
11			파일 본산 및 병합에 대한 정보를 담은 데이터베이스를 구축한다.							
12			추후에 새로운 드라이브 API를 사용할 수 있게끔 시스템이 설계되어야 한다.							
13		2.1 인증	MySQL로부터의 인증 토큰을 사용하도록 한다.							조익환
14		2.2 계정	중복된 계정은 생성할 수 있도록 한다.							이형복
15	2 회원/계정	2.3 비밀번호	6 ~ 10문자를 요구조건으로 한다. (영어 + 숫자)							이형복
16		2.4 닉네임	한글, 영문, 숫자 외의 기호는 포함시킬 수 있도록 한다.							이형복
17			미션 99에 클라우드로에 대한 간단한 소개 및 이점을 홈페이지로 제작하도록 한다.							
18		3.1 Intro	첫화면(intro) sign in, sign up에 대한 링크를 포함시키도록 한다.							
19			계정 및 비밀번호 인증 창을 포함시킨다. - Sign in							
20		3.2 Sign in, Sign Up	회원가입 페이지는 이메일(인증), 이름, 비밀번호, 비밀번호 확인의 인풋 창만을 가진다.							
21			아이디 찾기 / 비밀번호 찾기 - 회원가입시 인증된 이메일(즉 User의 ID)로 아이디 및 비밀번호 전송							
22		3.3 대시보드	인트로 페이지를 제외한 모든 페이지의 링크가 가능하게끔 한다.							
23			내 계정에 대한 전체적인 정보를 보여줄 수 있도록 한다.							
24		3.4 내 계정	내 계정에 대한 정보를 수정할 수 있도록 한다.							
25			계정 탈퇴를 가능할 수 있도록 한다.							
26			드라이브를 추가, 삭제 할 수 있는 기능이 포함되어야 한다.							
27			본산 병합 기능 관리를 위한 디렉토리를 탐색할 수 있도록 한다.							
28		3.5 설정	내 계정 설정으로 링크가 가능하게끔 한다.							
29			사용 용량 및 가능 용량에 대한 정보를 표시한다.							
30		3.6 이용약관 (about us)	간단한 튜토리얼 제공							
31			파일에 대한 미리보기 기능을 지원한다.							
32		3.7 내 드라이브	디렉토리 관리가 가능하게끔 한다.							

3-2. 작품 개발 측면

- 역할 분담 및 데드라인 설정
 - 각 팀원의 주력 분야에 배치
 - Trello 및 구글 문서를 활용하여 각자의 업무를 명확하게 명시 및 관리
- 각자의 산출물 병합
 - 웹 어플리케이션 뿌리 프로젝트 생성 - 첫 번째 Commit
 - Git을 이용하여 버전관리
 - 원격저장소는 GitHub를 이용
 - Branch를 master, test, develop, 각자의 역할을 기준으로 생성하여 좀 더 효율적이고 안정적으로 버전을 관리
- 코어기능 구현 및 관련 API 사용의 어려움
 - 관련 서적 구입 후 공부 및 실습을 실행한 후 구현
 - 관련 인터넷 강의 및 관련 블로그 참조
 - 오픈소스를 적극 활용

4. 프로젝트를 통해 배우거나 느낀 점

무료로 제공되는 클라우드 서비스들을 통합한다는 아이디어를 실현시키기로 저희 팀은 결심했습니다. 무엇보다도 기존 제공 서비스들을 연동하여 분산 저장시킴(RAID-3 달성)으로써 파생되는 효과가 엄청날 것이라고 저희들 스스로 자부하며 힘차게 프로젝트를 시작했습니다. 하지만 학교에서 배우지 않은 많은 장벽들이 앞을 수시로 가로막았고 그럴 때마다 관련 서적과 인터넷 검색을 통해 난관을 헤쳐 나갔습니다.

프로젝트 외적으로 팀원 각자가 성장함을 느끼고 있습니다. 팀원 각자가 인턴, 자격증 시험, 학교에서 진행되는 다른 프로젝트 등으로 바쁜 일상에도 불구하고 서로의 역할을 다른 팀원에게 미루지 않고 책임을 다함으로써 협업이 무엇인지에 대해서도 알아가고 있습니다. 또한 협업을 위해 요구되는 기술적인 요건들 또한 각자 인턴을 통한 경험 등을 공유함으로써 실제 기술적인 역량을 향상시켰다고 생각합니다. 중간에 포기하고 싶었던 적도 많았지만 그럴 때마다 팀원 간에 서로 의지를 다짐으로써 개발이라는 프로그래머의 특수한 직무에 대해 좀 더 자부심을 갖게 되었습니다.

간단한 아이디어가 세상을 변화시킬 수 있다는 창조적이고 도전적인 정신을 갖는 것, 저희는 이 모토를 항상 가슴 속에 새기며 끝까지 프로젝트를 완주하겠습니다.

IV. 작품의 기대효과 및 활용분야

※ 평가항목 : 기획력 (활용가능성)

1. 작품의 기대효과

- 기존 단일 클라우드 시스템의 ‘중앙 집중형 서버 저장방식’이 아닌 통합 연동을 통해 ‘분산형 저장방식’을 구현함으로써 서비스의 유연성과 확장성을 보장한다. 또한 특정 서비스에 종속되지 않고 사용자들의 다양한 니즈(Needs)에 따라 커스터마이징 환경(예를 들어 사용자가 원하는 클라우드 서비스만을 선택하여 선택적 연동)을 제공하기 때문에 특정서비스에 대한 의존도를 낮춰줄 수 있다.
- 독립 클라우드 서비스들의 한정된 저장 공간을 통합연동을 통해 단일 확장 클라우드처럼 사용 가능토록 하고, 이를 통해 낭비되어왔던 저장공간들의 효율적인 활용이 가능해진다. 또한 본 프로젝트 시스템(MOAA)에 연동된 모든 클라우드 서비스들은 사용자 파일 업로드시 항상 동일한 크기의 분할 패킷들이 저장되기 때문에 관리비용을 최소화할 수 있다.
- 기존 클라우드 관리 플랫폼들과는 달리 Zip4j API와 자체 알고리즘을 활용한 파일 시스템을 활용함으로써 사용자 데이터에 높은 안정성과 보안성을 보장해준다. 이를 통해 하나의 클라우드 서비스가 노출되어도 나머지 클라우드에 저장된 패킷을 얻어내지 못하는 이상 공격자는 원본파일을 복구해 낼 수 없기 때문에(독자적인 사용이 불가능) 기존 클라우드 서비스들의 보안취약점을 보완할 수 있다.
- 원본/ 분할파일들의 저장경로와 ID가 기록되는 Log파일과 해쉬함수를 활용하여 만들어진 독자적인 디렉토리(User Domain)를 통해 서비스의 전체적인 성능을 향상시킬 수 있다. 또한 파일의 Read, Write, Delete시 단순 큐(Queue)형식의 병렬처리대신 Log파일을 읽어옴으로써 시스템 프로세스 관리를 최적화시킬 수 있다.

2. 작품의 활용분야

- 활용분야
 - 개인의 중요 파일 백업
 - 기업의 중요 시스템,DB 및 파일 백업

- 기존 클라우드 서비스들의 통합 관리
- 연동 클라우드 및 시스템 총 저장 공간 시각적으로 확인

V. 개발산출물

※ 평가항목 : 평가 전반에 참고

○ 소프트웨어 기능

시스템에 구현되는 소프트웨어 기능은 4가지(파일 관리, 클라우드 관리, 파일 분할 업로드/다운로드/삭제, 파일 검색 기능)로 구성된다.

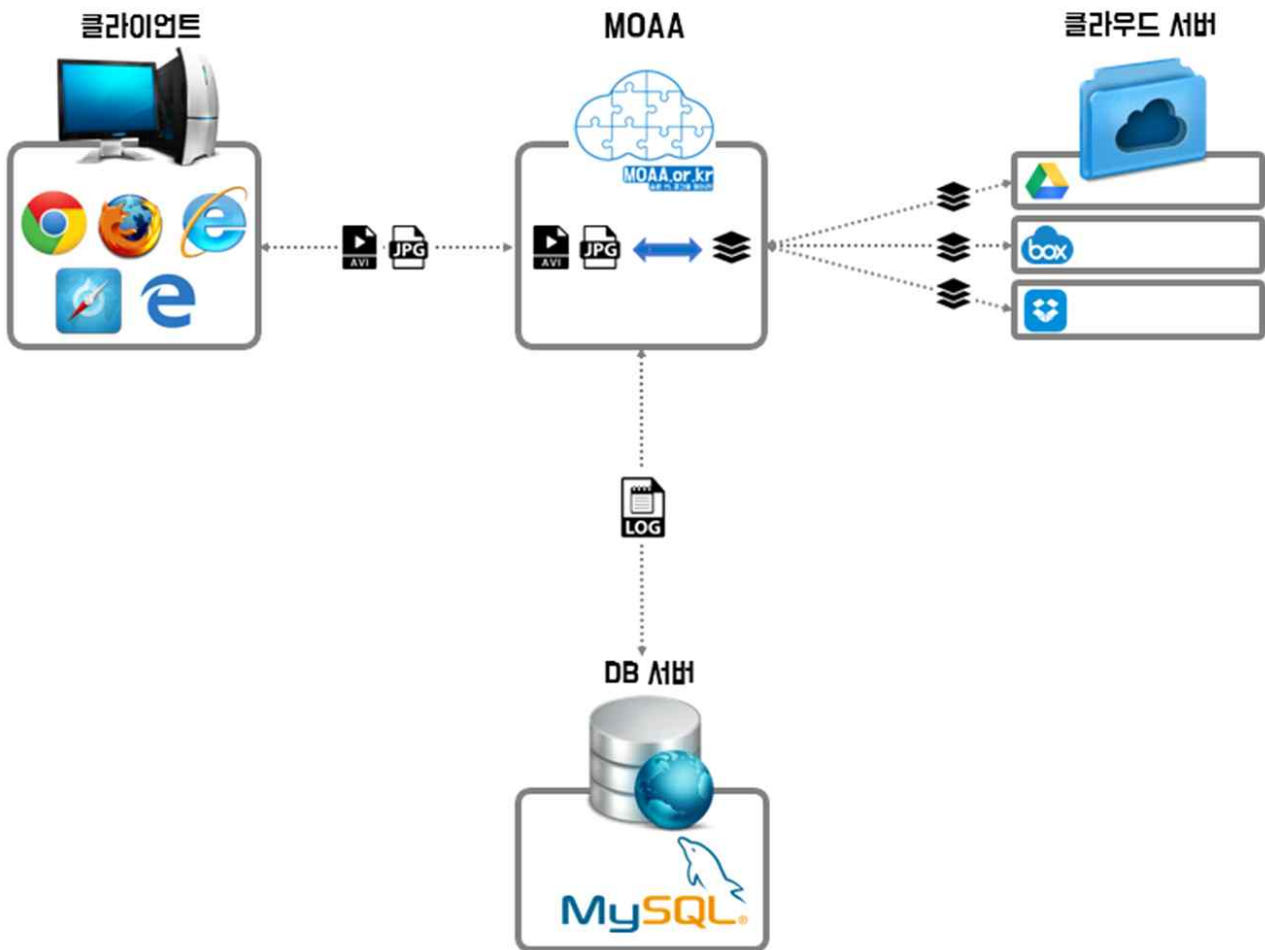
1) 파일관리 기능은 웹사이트 내에 분할 업로드 된 파일들의 목록 및 해당 파일들의 형식, 용량, 최근 수정된 날짜 등 파일 관리를 위한 기본적인 정보를 제공한다.

2) 클라우드 관리 기능은 사용자가 기존에 등록한 클라우드 서비스의 드라이브 현황을 ‘드라이브 설정’에서 표시한다. 또한 통합된 드라이브의 총 및 사용/가용공간을 그래프로 시각화 하여 최 앞단에 표시되며, 파일 업로드 시 드라이브 활성화 및 비활성화 여부를 지정할 수 있도록 한다.

3) 파일 분할 업로드 기능은 JSP내의 ZIP4j API를 이용하여 파일을 패킷으로 분할하고 활성화된 클라우드로 분산 저장시킨다. 업로드 되는 분산패킷에 대하여 경로와 ID를 기록하는 로그파일을 생성하고, 다운로드 기능은 이 저장된 로그파일의 경로를 인자값으로 하여 각각의 패킷들이 저장된 드라이브들로부터 다운로드, 병합한다.

4) 파일 검색 기능은 업로드된 파일 목록에서 사용자가 검색한 단어 및 문장과 일치하는 파일 목록을 제공한다.

○ 소프트웨어 구조



사용자 인터페이스는 HTML, CSS, JSP, JavaScript, Bootstrap을 기본으로 하고 생성된 해쉬함수 디렉토리 및 로그파일에 대해서는 MySQL을 사용하여 저장한다.

Linux(centOS)서버 내에 APACHE/TOMCAT 7.0 서비스를 사용하여 웹 서비스를 구동한다.