|  |
| --- |
| **Web Based Data Crawling Engine**  **전공종합설계 – 최종 보고서** |

|  |  |  |  |
| --- | --- | --- | --- |
| **제출일** | 2017.5.21 | **소속학부** | 컴퓨터학부 |
| **과목명** | 전공종합설계1 | **조명** | Mr.ROBOT |
| **담당교수** | 이상준 | **조원** | 20112339 김성진  20112315 강준호  20112423 유상현 |

목 차

1. Project 개요

1.1. Project 제안 배경

1.2. Project 목적

1.3. Project 기대 효과

2. System 분석

2.1. System 구성도

2.1.1. Client Side

2.1.2. Server Side

2.2. 기술 분석

2.1.1. 요소 선택기 (Element Selector)

2.1.2. RTCS (Real-Time Communication System)

2.1.3. Script Injection

2.3. 유사 서비스 분석

2.3.1. FEEDLY

2.3.2. EverNote-WebClipper

2.3.3 Feed43

2.3.3 Konan사 Crawling Engine

2.4. 기술적 차별성

2.3.1. Convenience

2.3.1. Universality

2.3.1. Efficiency

2.3.1. Quality Insurance

목차

3. Design

3.1. 예상시나리오

3.2. Sequence Diagram

3.2.1 Change Detectioin

3.2.2 Rule Save

3.2.3 Data Export

3.2.4 Rule Edit/Delete

3.3. 프로그램 순서도

3.4. 클래스 다이어그램

4. Project 수행 계획

4.1. Project 개발 자원

4.2. Opensource Resource

4.3. 협업 Tools

4.3. Project 일정표

4.4. Project 개발환경

5. Project 수행 결과

5.1. 실행 절차

5.2. 실행 결과

5.3. 한계 및 발전전략

# 개요

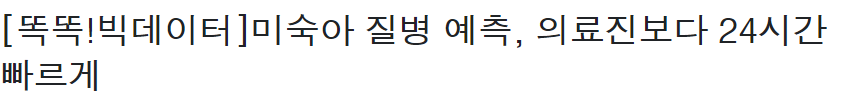
## 제안배경

현재 시점을 기준으로 4차 산업혁명이 대두되면서 최근 대선에서도 4차 산업혁명이라는 단어가 많이 사용되었다.

빅데이터 분석을 바탕으로 인공지능을 활용한 다양한 연구는 여전히 진행형이며, IT분야가 아닌 다른 분야에서도 데이터분석을 통한 효율적인 서비스를 개발하고자 연구중이다.



그림입니다. 원본 그림의 이름: 스포츠데이터분석.PNG 원본 그림의 크기: 가로 438pixel, 세로 51pixel



위의 사진은 포털사이트에서 ‘데이터 분석’이라는 키워드로 간단히 검색한 뉴스기사들 중

IT이외의 분야에서 데이터분석을 통한 연구를 하고있다는 기사의 제목을 발췌한 것이다.

‘데이터 분석’이라는 간단한 키워드 만으로도 타 분야에서의 데이터분석에 대한 관심도가 높다는 것을 알 수 있다.

데이터 분석의 수요가 증가함에 따라 효과적인 데이터 수집엔진의 수요 또한 같이 증가할 것으로 예상된다.

우리의 프로젝트는 IT분야가 아닌 타분야에서도 쉽게 사용할 수 있는 웹기반 데이터수집엔진을 개발하고자 한다.

RSS(Rich Site Summary)란?

뉴스나 블로그 사이트에서 주로 사용하는 콘텐츠 표현 방식으로 웹 사이트 관리자는 약속된 프로토콜을 따라 RSS Feed를 제공하면 사용자들은 RSS리더기를 통해서 웹 사이트 관리자가 의도하는 콘텐츠에 한하여 직접 사이트에 접근 하지 않고도 정보를 습득할 수 있게 해주는 방식이다.

현재 존재하는 RSS서비스들은 구독하고자 하는 웹 페이지의 관리자가 만들어준 피드를 사용하는 것이 대부분이다. 또한 RSS피드를 제공하지 않는 웹 페이지가 대다 수이다. 사용자 입장에서는 RSS서비스를 누리고 싶은데 누리지 못하는 경우가 태반 이며, 본인이 지정하지 않은 데이터도 불필요하게 받아야 하는 불합리함이 있다. 따라서 RSS피드를 제공하지 않는 웹 페이지의 피드를 제공하고, 사용자가 원하는 섹션의 정보만을 지정하여 받아볼 수 있게 하는 시스템이 필요하다.

더 나아가 사이트 관리자에 의해서 RSS가 제공되는 경우에 있어서도 사용자의 사용용도와 적합하지 않는 정보들도 의도치 않게 정보를 받아들여야하는 상황이 생길 수 있다.

예를 들어 ‘블로터’ “www.bloter.net”에서 제공하는 RSS를 chrome extension Feedly 리더기를 통하여 보면 다음과 같은 결과를 얻을 수 있다.



위의 그림은 블로터에서 제공하는 모든 카테고리의 구분없이 새롭게 추가되는 기사모두를 시간순으로 제공하고 있다. 블로터에서 제공하는 모든 기사를 구독하고 싶은 사용자도 있는 반면 블로터에서 제공하는 ‘IT열쇳말’ 카테고리만을 구독하고 싶은 사용자도 있을 수 있다.

앞서 예기한 데이터수집엔진의 수요도 증가와 RSS라는 기술에 착안하여 서버단에서는 데이터 수집을 클라이언트 단에서는 기존의 서비스되고 있는 RSS리더기와 호환이가능한 RSS피드를 제공하여 수집된 데이터를 RSS리더기를 통해 클라이언트에게 제공하고자 한다.

## 목적

우리 프로젝트는 웹 환경에서 RSS피드를 제공하지 않는 웹 페이지로부터 rss피드를 제작하여, 일반적인 사용자에게 RSS서비스를 제공하는 것에서 시작한다. 기존의 RSS서비스와 달리 사용자가 원하는 페이지의 원하는 섹션을 지정하면, 그 부분의 RSS피드를 우리가 제작하여 보다 사용자 욕구에 가까운 RSS서비스를 누릴 수 있게 하는 것이다. 그리고 궁극적으로는 우리 프로젝트가 만들어낸 RSS피드를 바탕으로 크롤러 엔진을 제작하여, 크롤러 제작을 용이하게 하는 것을 목표로 한다. 우리가 살고 있는 시대는 정보의 바다를 넘어 정보의 우주라고 할 정도로 데이터가 과다한 시대에 살고 있다. 일반적인 사용자는 사용해본 적도 없는 제타바이트(ZB) 단위의 데이터가 웹 상에서 돌아다니고 있다. 몇 년째, 언론에서는 빅데이터를 외치 며 데이터의 중요성을 강조하고 있다. 뿐만 아니라 지난 해 알파고가 초래한 인공 지능이라는 화두는 더욱 더 데이터의 중요성을 우리에게 상기시킨다. 이러한 현실 속에서 일반 사용자들에게는 RSS서비스를 응용하여 사용자 맞춤형 데이터 구독 서 비스를 제공함과 동시에 기존의 데이터 수집엔진이 가지고 있는 한계를 극복한 수집엔진을 개발하고자 한다. 우리의 프로젝트는 편리성, 범용성, 품질 보증, 효율성 에 초점을 맞추고 기존의 데이터 수집엔진과 차별화를 하고자 한다.

## 기대효과

- RSS 피드를 제공하지 않는 사이트에서도 RSS리더기를 통해서 마치 사이트에서 RSS피드를 제공하는 것처럼 사용할 수 있다.

– 사용자의 입맛에 맞게 제작된 각사이트들의 피드들의 사용량을 바탕으로 실제 사이트운영에 RSS피드 제공의 필요성을 제고할 수있다.

* IT분야가 아닌 다른 분야에서도 사용하고자 하는 데이터를 쉽게 선택하여 수집할 수 있다.

# SYSTEM 분석

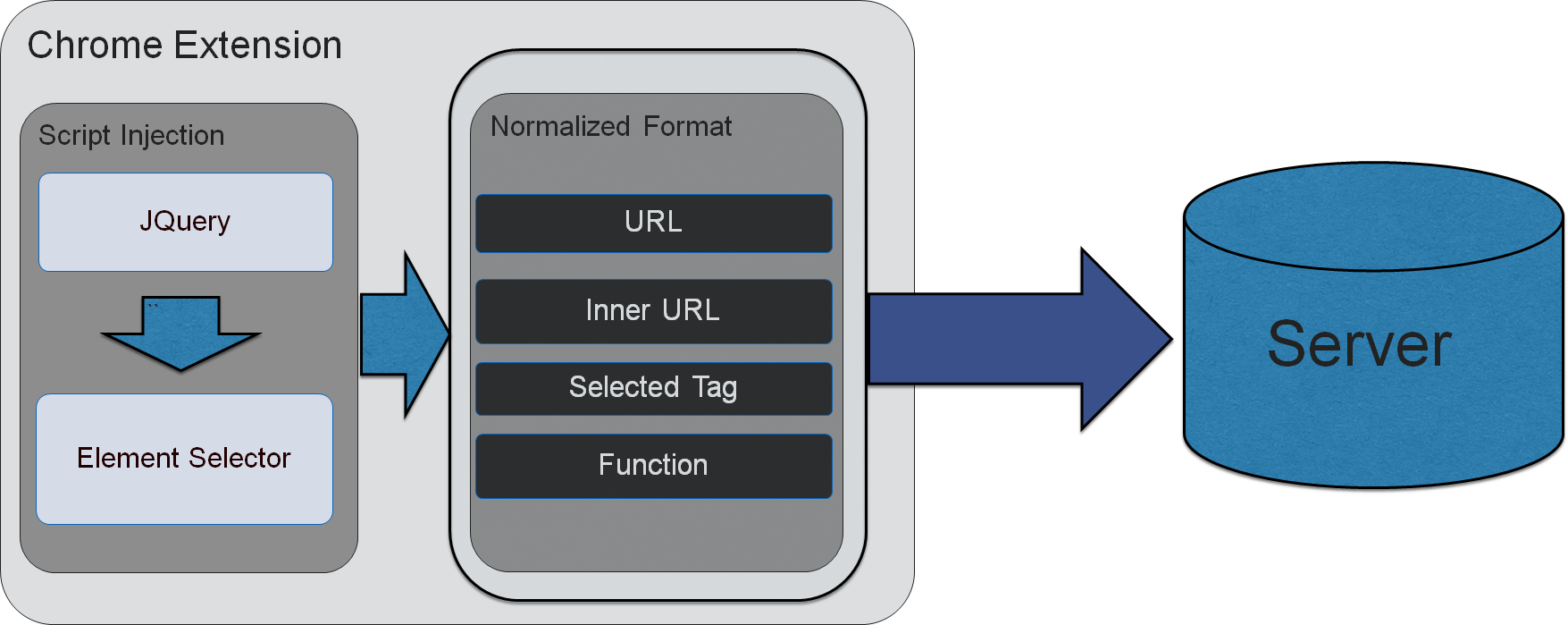
2.1. System 구성도 - 10 -

2.1.1. Client Side - 10 -

프로젝트의 사용용도로 판단한 Client는 2분류로 나눌 수 있다.

첫 번째는 웹 페이지 상에서 수집을 원하는 Client이고 두 번째는 서버가 수집한 데이터를 API형태로 제공받고자 하는 Client이다. 우리의 프로젝트는 먼저 첫 번째 Client에 초점을 맞추어 데이터를 축적하는 것을 기본 목표로 설정하였다.

Client Side 시스템 구성도는 다음과 같다.



Client는 Chrome Extension을 사용하여 우리의 서비스를 이용할 수 있다.

Chrome Web Store를 통해 브라우저에 우리의 프로젝트를 설치한 Client는 수집을 하고자하는 웹사이트에 접근한 후 Extension을 실행 시킨다.

최초 실행 시 기존의 RSS리더기에서 제공하는 몇가지 표출 Format을 제시하여 주고 그 중 Client가 원하는 Format을 선택할 수 있다.

예를 들어, 사진과 제목, 내용을 포함하는 Format, 제목만 포함하는 format등의 여러 Format을 제공하고 그 중 한가지를 고를 수 있도록 한다.

Client가 Format을 선택하면 Extension Library를 이용하여 JQuery를 기반으로 제작한 Element Selector script코드를 실제 웹페이지 상에 Injection 시킨다.

Injection된 코드는 마치 원래 웹페이지상에 있는 코드처럼 동작하여 Client에게 보여진다.

Client는 이기능을 이용하여 초기에 선택한 Format에 맞추어 원하는 Tag를 선택한다.

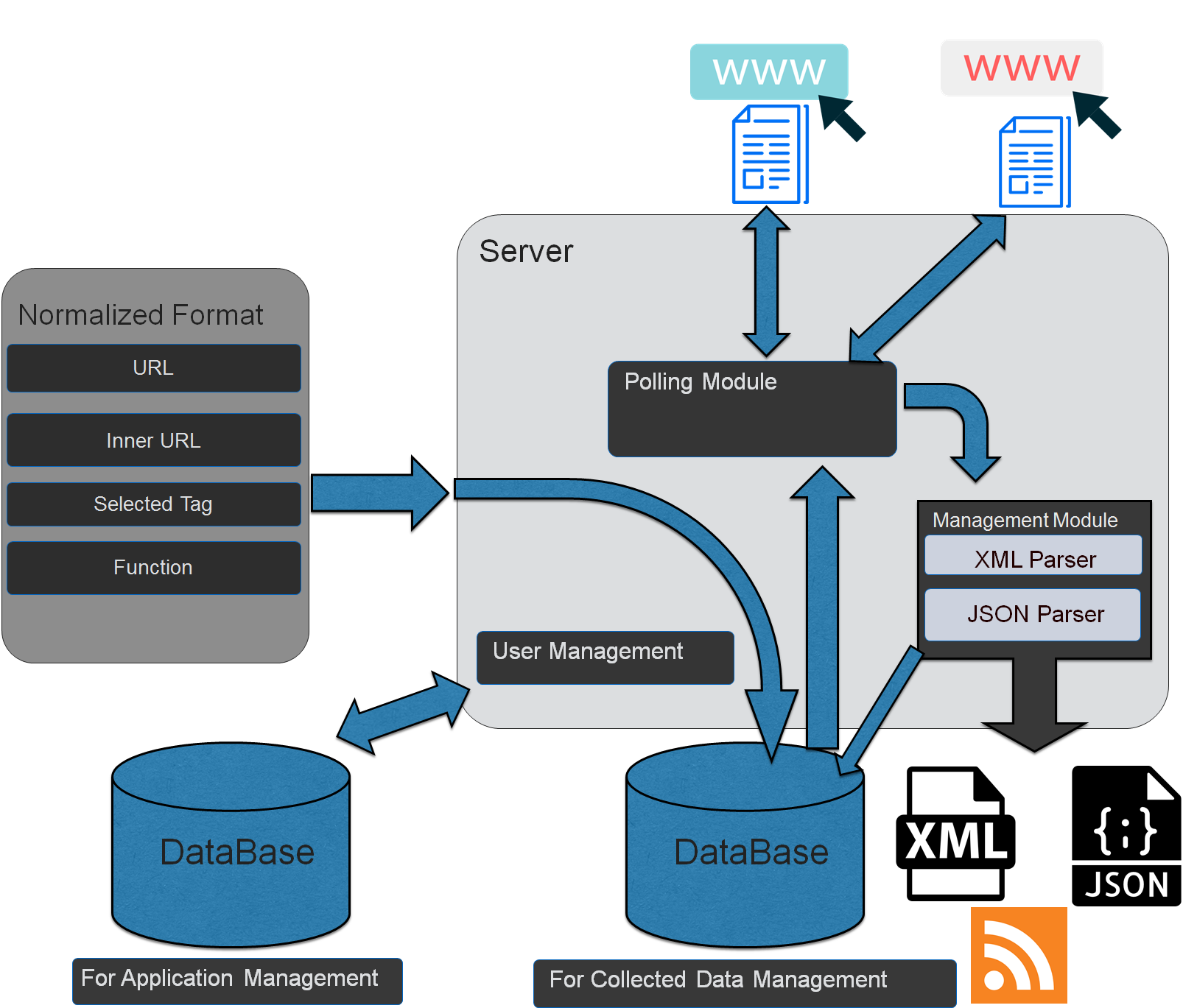
Element Selector는 선택된 Tag를 확인하고 데이터를 수집함에 있어서 반복되는 부분의 여부를 확인한다. 반복되는 부분이 있다고 판단하는 경우 Client가 선택한 Tag와 반복적으로 표출되는 모든 Tag를 한꺼번에 묶어서 사용자에게 보여준다.

이러한 방법으로 원하는 Format에 맞게 선택된 정보들은 서버에서 쉽게 인식하고 처리하기 위하여 표준화를 시킨후 서버로 Format을 전송한다.

2.1.2. Server Side - 10 -

서버는 Client로부터 전송받은 Format을 바탕으로 모든 작업을 수행한다.

서버의 구성도는 다음과 같다.



서버는 Application 관리에 사용되는 Data와 Contents를 포함하는 Data를 따로 관리하기위해 2개의 Database를 사용한다. 사용자 관리를 비롯한 Application 내부에 필요한 Data는 RDBS를 이용하여 별도로 관리를 하고 많은 양의 Data를 수집하고 관리를 해야하는 Contents의 경우에는 NoSql database를 사용한다.

서버는 클라이언트로부터 받은 표준화된 포맷을 Database에 저장을 한다.

서버 내부의 Polling Module은 사용자가 수집을 원하는 실제 웹페이지와 통신을 하기 위한 모듈로 데이터베이스로부터 사용자가 요청하였던 페이지에 대한 포맷을 바탕으로 실제 페이지와 Polling을 통해 변경된 정보, 추가된 정보가 있는지를 지속적으로 감시를 한다.

변경된 요소가 감지된 경우 변경사항을 다시 데이터베이스에 저장을 하고 관리 모듈을 통해서 RSS피드를 갱신시킨다.

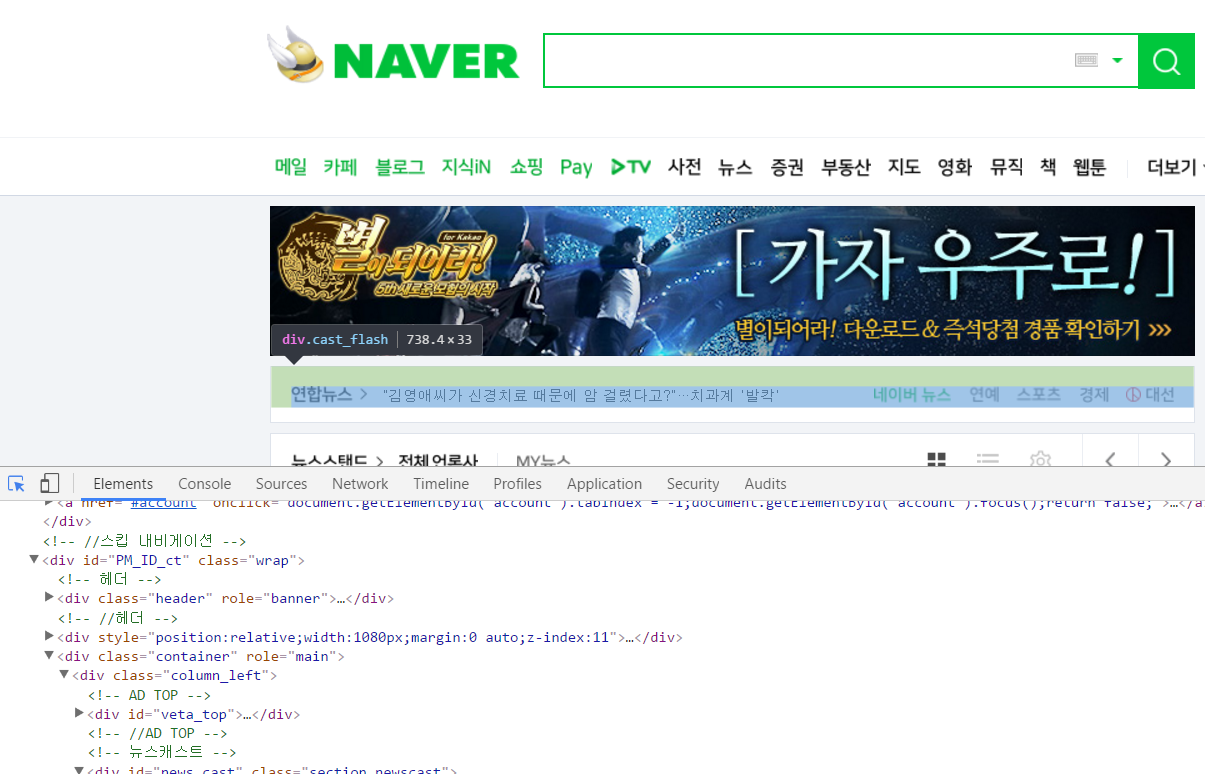
관리모듈은 XML Parser를 통해서 RSS피드를 생성시킬 수 있고 JSON Parser를 통해 수집된 데이터를 JSON 형식으로도 Client에게 제공할 수 있도록함으로써 다른 Application에서 우리의 아이템을 API 형태로 호출하여 사용할 수 있도록 한다.

2.2. 기술 분석

2.1.1. 요소 선택기 (Element Selector)

**- Element Selector**

기존에 사이트에서 제공하는 Format 외에는 원하는 데이터가 있어도 RSS를 통해서 정보를 받아볼 수 가 없다. SMS는 사용자로부터 원하는 부분을 HTML 태그 단위로 쪼개어 선택할 수 있는 기능을 제공해야 한다.



Chrome 요소검사의 Element Selector

Chrome 브라우저에서 제공하는 요소검사 기능 중 HTML태그를 선택하는 기능에 착안하여 Element seletor를 구현하고자 한다. 수시로 업데이트 되는 부분을 선택해야 하기 때문에 그림에서 보이는 것처럼 마우스로 클릭한 부분만을 선택하는 것이 아닌 반복되는 Tag를 통해 출력 되고 있는 모든 요소들을 한꺼번에 선택 할 수 있도록 구현할 예정이다.

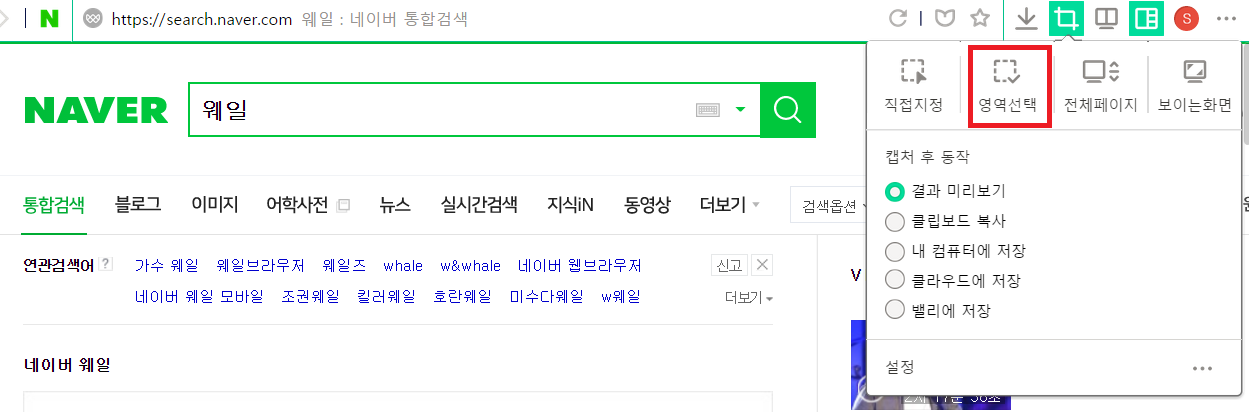


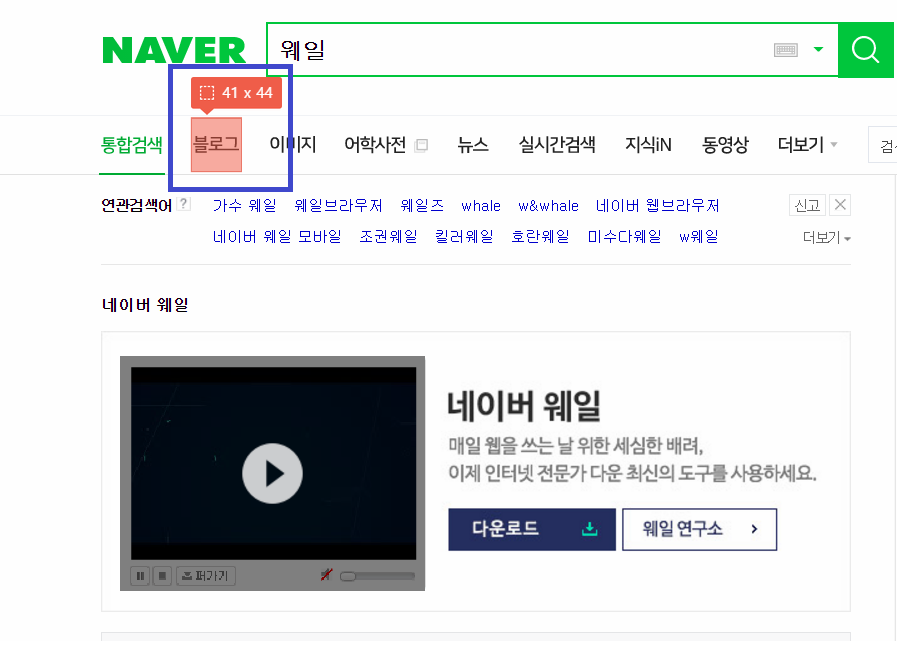
태그 선택의 예시화면

예시화면에서 보이는 것처럼 “한국 경제”부분을 클릭하면 같은 구조로 이루어져 있는 모든 태그를 선택할 수 있도록 하는 기능을 구현할 예정이다.

네이버에서 최근 공개한 Whale 브라우저의 경우를 살펴보면 Element Selector의 기능을 잘 보여주고 있다.

웨일 브라우저에서는 페이지를 캡처하는 기능을 제공하는데 여기서 영역을 선택하여 캡처하는 부분은 Element Selector 부분을 잘활용하고 있다.





위의 그림에서처럼 HTML태그별로 영역을 선택할 수 있다. 우리의 프로젝트도 HTML 태그별로 공통된 부분을 추출하는 기능을 핵심기술로 이용해야하기 때문에 이기능을 적극 활용할 예정이다.

2.1.2. RTCS (Real-Time Communication System)

**-RealTime Communication System(RTCS)**

사용자에 의해서 선택된 부분에 대해서 데이터를 지속적으로 갱신해주는 것이 무엇보다도 우선시 되어야 할 기술이다. 기존에 사이트에서 RSS를 제공해주었다면 사이트에서 RSS 피드를 갱신을 해주어야만 업데이트된 정보를 받아볼 수가 있다. 하지만 SMS은 RSS 피드를 독점적으로 생산, 관리를 하기 때문에 실제 사이트에서 갱신된 정보를 받아오는 과정과 피드를 갱신시켜주는 과정을 구현해야 한다.

웹은 일반적으로 비동기방식 통신을 기반으로 하기 때문에 정보를 RealTime으로 보이기 위해서는 동기방식 기술구현이 필요하다. 현재 웹에서 사용되는 실시간 통신 방식은 몇가지가 있다.

가장기본적인 통신 방식은 Polling 방식인데 서버에 주기적으로 요청하는 방식이기 때문에 서버에 오버헤드가 발생할 수 있어 SMS에서는 WebSocket을 기반으로하는

클라이언트의 한번의 요청만으로 지속적으로 통신이 가능한 방법으로 구현하고자 한다.

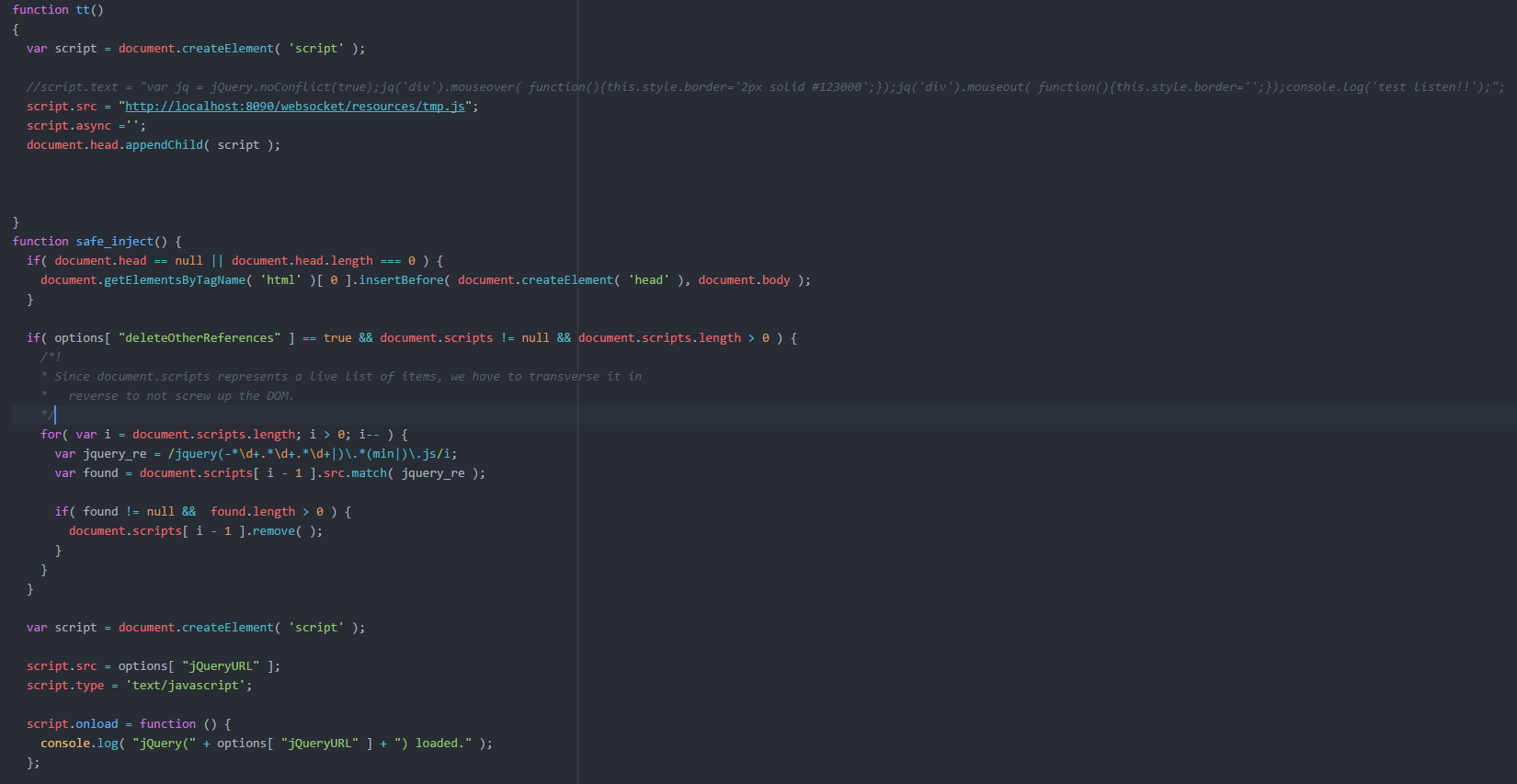
2.1.3. Script Injection

프로젝트의 최초 시작은 Chrome Extension에서 동작하는 형태로 시작할 예정이기 때문에

Chrome Extension에서 제공하는 Script Injection을 통해서 상용화된 페이지에서 원하는 Script를 실행할 수 있도록 하여 Element Selector를 구현한다.

위의 그림은 Extension을 통해 네이버 뉴스 페이지에 태그를 선택하는 Script를 Injection한 예시이다.

브라우저에 보여지는 페이지를 DOM객체로 접근하여 기능을 구현할 것이므로 jQuery를 먼저 페이지에 삽입 후 JQuery를 이용하여 Injection 할 기능을 개발한다.

다음은 Jquery를 삽입 후 Sample Source를 삽입한 예제 그림이다.

2.3. 유사 서비스 분석

우리가 진행하는 프로젝트는 현재 서비스 하고 있는 RSS리더기와의 호환성을 토대로 우리의 서비스에 대한 수요군을 폭넓게 확장하고자 한다. 즉 사용자들이 RSS피드를 한눈에 보기 쉽도록 UI를 구성하는 작업에 큰 비중을 두지 않는다. 하지만 서비스 되고 있는 리더기들과의 호환을 위해 그들이 추구하는 UI의 형태로 표출할 수 있도록 피드를 제작하고자 한다.

현재 서비스 하고있는 대표적인 리더기들은 다음과 같다.

2.3.1. FEEDLY

기능, 디자인, UI 면에서 세련된 점이 특징이다.

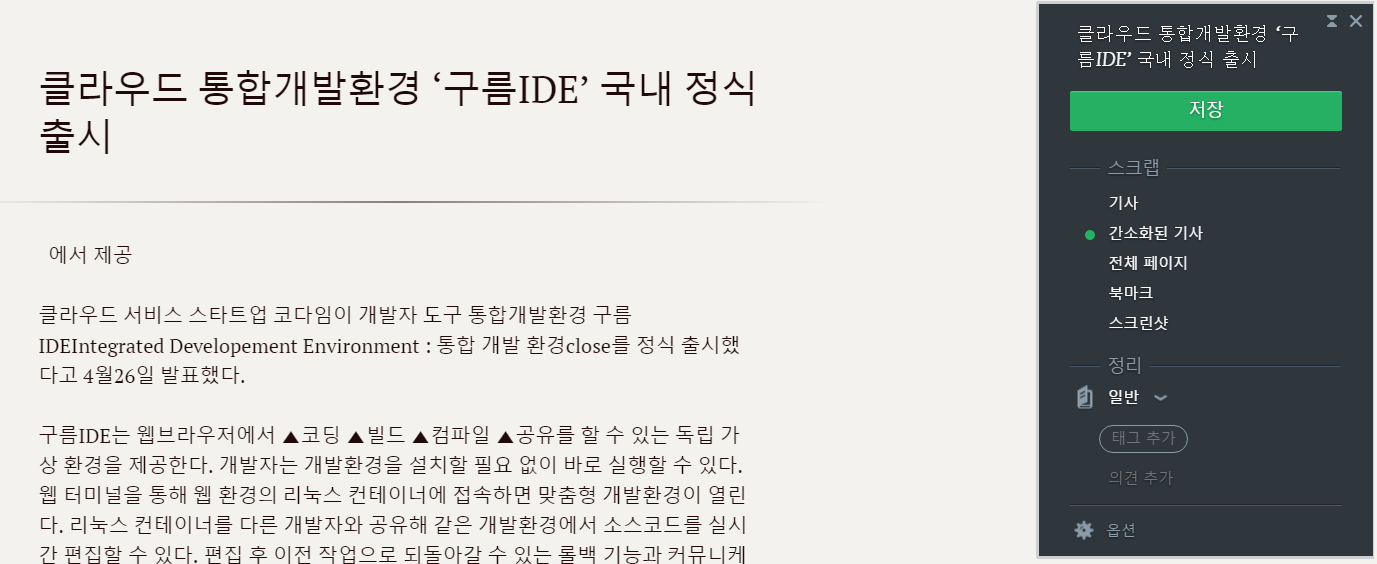


블로터 사이트에서 제공하는 RSS피드를 바탕으로 FEEDLY에서 구독하는 경우 위와 같은 UI로 제공받을 수 있다. UI를 구성하는데 사용되는 RSS피드 태그는 다음과 같다.



2.3.2. EverNote-WebClipper

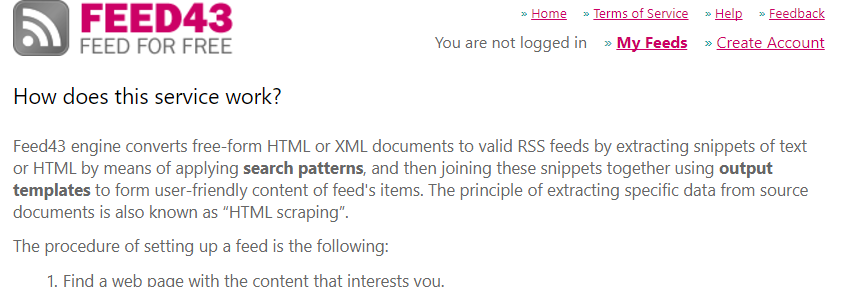
Web-Clipper란 텍스트, 이미지, 링크 등을 포함해 웹 페이지를 스크랩할 수 있는 기능이다. EverNote에서 지원하는 웹클리핑 기능은 현재 서비스되고 있는 웹클리퍼중에 가장 뛰어난 성능을 보여준다.



위의 그림처럼 EverNote의 웹클리핑 기능은 뉴스기사 와 같은 페이지를 자동으로 복사하여 저장할 수 있는 기능을 제공한다.

우리의 프로젝트는 웹 클리핑 기능에서 더 나아가 웹클리핑을 하기전 사전에 반복되는 구조를 서비스에 인식을 시켜둠으로써 새롭게 추가되는 정보를 자동으로 클리핑(HTML태그를 그대로 복사한다면 웹클리핑이 되고 RSS format이나 ATOM format으로 제공한다면 RSS FEED가 될것이다)할 수 있다는 점에서 차이가 있다.

2.3.3 Feed43



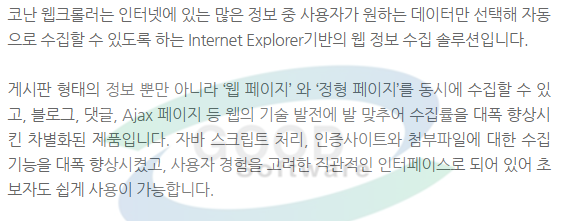
이것은 Feed43의 서비스 소개 말이다.

Feed43은 검색패턴을 적용하여 텍스트 또는 HTML의 발췌 문장을 추출한 다음 출력 템플릿을 사용하여, 발취 문장을 결합하고 사용자가 쉽게 사용할 수 있는 피드 항목을 구성함으로써 자유 형식의 HTML 또는 XML 문서를 유효한 RSS 피드로 전환한다.



다음 그림에서 알 수 있듯이 IT 지식이나 비 개발자의 경우는 이런 서비스를 이용하는 것이 쉽지 않다. 실제 HTML 코드를 보고 정규식을 이용해서 RSS피드를 추출해내야 하는데, Web based Data Crawling Engine은 그런 작업들을 간단한 UI 조작만으로 할 수 있게 한다는 점에서 차별점이 있다.

2.3.4 Konan사 Crawling Engine

코난 테크놀로지에서 제공하는 웹 크롤러는 사용자가 원하는 데이터를 선택해 자동으로 수집을 할 수 있도록 한다는 점에서는 우리의 프로젝트와 동일하다. 하지만 우리는 코난 웹 크롤러가 가진 문제를 파악하고 그문제를 보완하여 더 나은 수집엔진을 개발하고자한다.

1.코난 웹크롤러는 Internet Explorer 기반의 웹 정보 수집 솔루션이다. 그렇다면 Internet Explorer를 지원하지 않는 맥 OS같은 경우 이런 서비스를 이용할 수 없다.

Web based Data Crawling Engine의 경우에도 Chrome을 기반으로 동작하다는 점에서는 유사하다고 생각될 수 있으나 우리의 프로젝트는 서버단 과 클라이언트단을 독립적으로 분할을 하여 설계를 하기 때문에 초기 개발은 Chrome Extension을 기반으로 개발을하지만 추후 서버단 개발이 완료되면 Client는 모바일, 이나 다른 브라우저를 통해서도 접근을 할 수 있도록하는 과정이 용이하다.

2. 코난 웹크롤러의 경우 정형화된 페이지만을 수집을 할 수있다.

Web based Data Crawling Engine은 사용자가 원하는 방법을 설정할 수 있도록함으로써 정형화된 페이지가 아닌경우엥도 사용자의 간단한 규칙설정만으로도 정형화를 시켜 수집을 할 수 있다.

2.4. 기술적 차별성

2.4.1. Convenience

FEED43의 경우 실제사이트에서 RSS피드를 제공하지 않는 경우 사용자가 직접 RSS피드를 제작할 수 있게 해준다는 점에서 우리의 프로젝트와 유사성을 지닌다. 하지만 우리는 사용자가 아이템을 사용하는데 있어서 편리하게 사용할 수 있도록 하는데 중점을 두어 간단한

요소 선택만으로 쉽게 수집 규칙을 설정할 수 있도록 할 예정이다.

2.4.2. Universality

상용중인 수집엔진의 중 Quality있는 수집엔진의 경우 범용적인 사이트에 대한 수집엔진이 아닌 트위터와 같은 특정 사이트에 국한되어 제작이 되는 경우가 많다.

특정 페이지에 국한된 수집엔진의 문제에 착안하여 우리는 Quality있는 수집을 제공함과 동시에 범용적으로 사용될 수 있도록 구현하고자 한다.

2.4.3. Efficiency

위에서 언급한 범용성의 문제를 극복한 수집엔진도 존재한다. 하지만 범용성의 문제를 해결하고자 하는 수집엔진의 경우 사용자가 실제로 원하는 데이터에 대한 부분에 모호해지는 경향이 있어 굳이 필요로 하지 않는 Noise Data가 포함된 Data를 수집하는 경우가 많다.

우리는 사용자로부터 필요로 하는 부분만을 선택하게 함으로써 필요하지 않는 부분은 제외하고 필요로 하는 부분만을 수집할 수 있도록 하여 효율성을 높이고자 한다.

2.4.4. Quality Insurance

실제 사이트와 사용자간에 Bridge역할을 담당하는 우리의 수집엔진은 사용자에게 제공하는데이터가 실제 사이트로부터 갱신되었다는 것을 보증해주어야 할 필요가 있다.

사용자에게 제공되는 데이터가 실제 사이트로부터 수집이 되었다는 것을 보장하기 위해

서버가 실제사이트로부터 수집한 시각을 LOG로 관리를 하여 사용자가 요청할 경우 서버의 수집기록을 제공함으로써 사용자가 제공받는 품질에대한 보증을 할 수 있다.

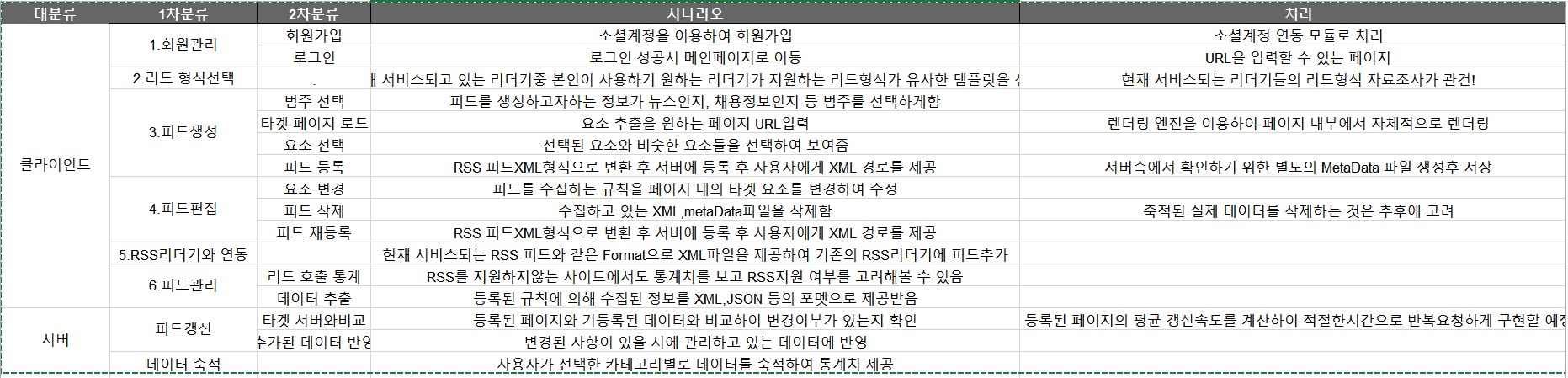
# DESIGN

3.1. 예상시나리오

예상 시나리오와 오픈소스 리소스의 경우 팀원들간 협력을 위해 구글 스프레드시트를 사용하고 있다.

MR.ROBOT 기능명세서 링크

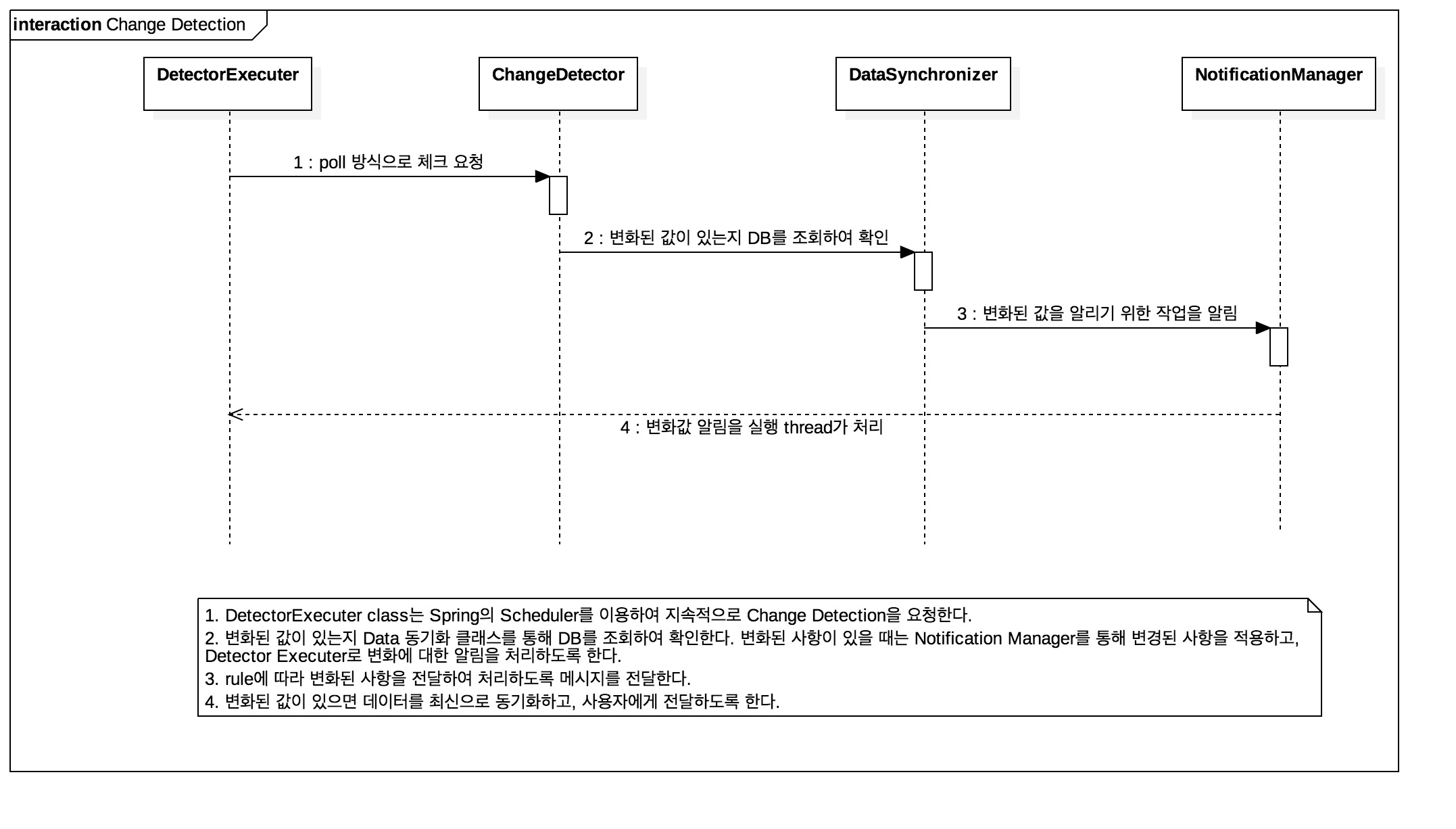
[https://docs.google.com/spreadsheets/d/1v-TmOXf6Pf4ruMLK1TrmkFX24jVkVs-eBNfDzxkKR50/edit#gid=0](https://docs.google.com/spreadsheets/d/1v-TmOXf6Pf4ruMLK1TrmkFX24jVkVs-eBNfDzxkKR50/edit)



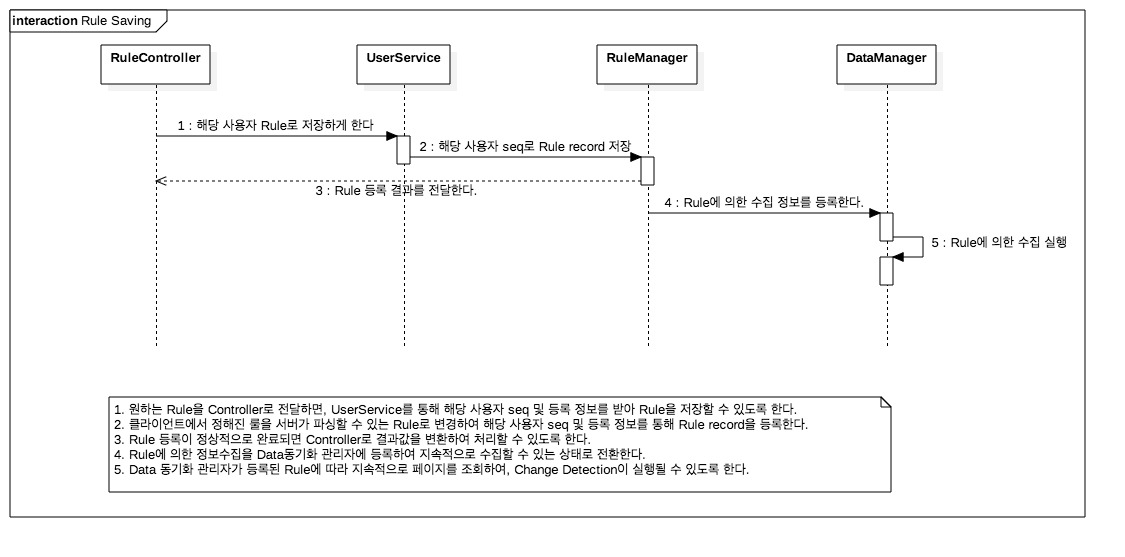
예상 시나리오 예시

3.2. Sequence Diagram

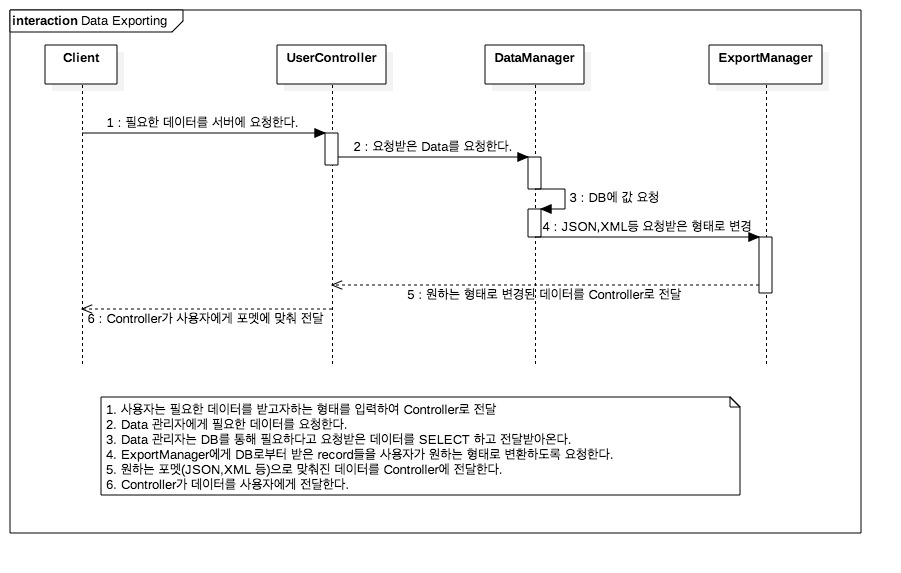
3.2.1 Change Detectioin Process



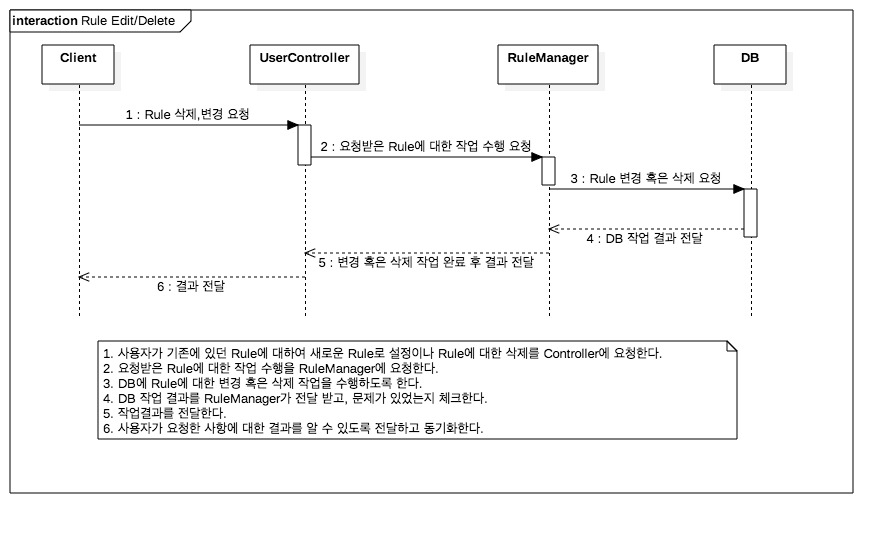
3.2.2 Rule Saving Process



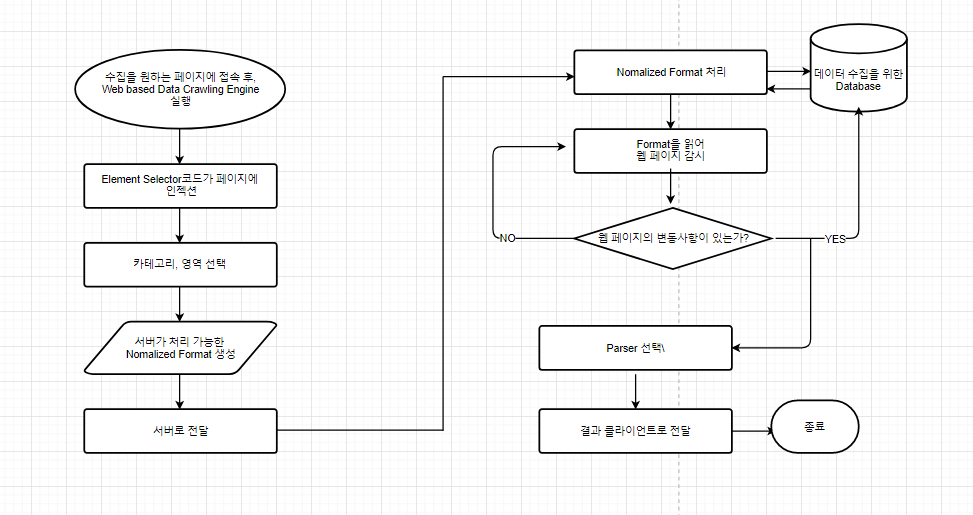
3.2.3 Data Export



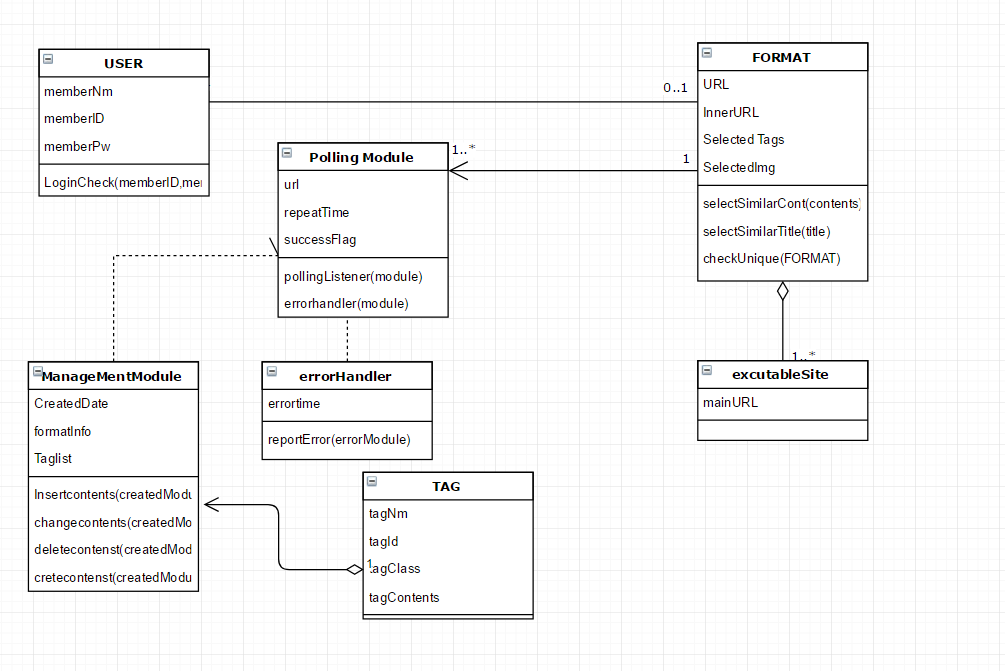
3.2.4 Rule Edit/Delete



3.3. 프로그램 순서도



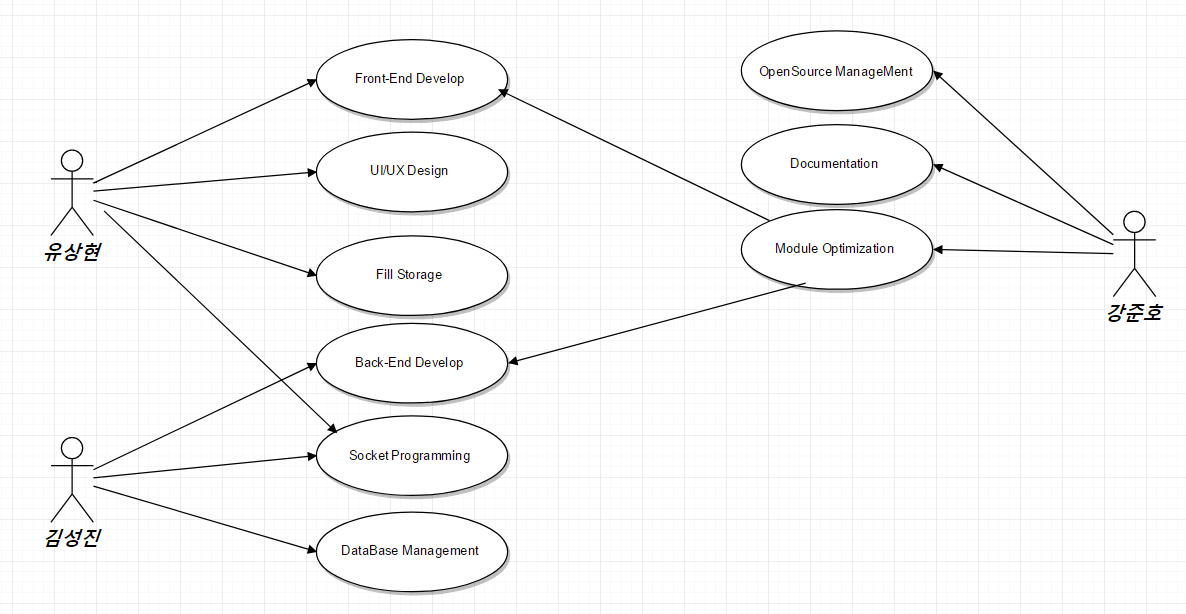
3.4. 클래스 다이어그램



# PROJECT 수행 계획

4.1. Project 개발 자원

|  |  |
| --- | --- |
| **책임 및 역할** | **책임자** |
| 사용하는 오픈소스 최적화 | 강준호 |
| Server 개발 | 김성진 |
| Front 개발 | 유상현 |



4.2. Opensource Resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **분류** | **오픈소스** | **설명** | **비고** | **추가날짜** |
| 화면개발 | Bootstrap | 자바스크립트 및 CSS 오픈소스로 웹페이지를 반응형으로 쉽게 개발 할 수 있도록 함 |  | 2017-04-18 |
| jquery | 자바스크립트 오픈소스로 웹 요소(엘리먼트)를 효율적으로 찾고 수정할 수 있음 |  | 2017-04-18 |
| datatable | 수집된 정보를 웹화면서에 쉽게 개발하기 위해 사용 |  | 2017-04-18 |
| 서버측 개발 | sockjs | 실제 페이지에서 수정이 일어난 다음 빠른 시간안에 피드를 수정해주어야 하기때문에 동기방식을 이용한 통신을 하기위해 사용 | <https://github.com/sockjs/sockjs-client> | 2017-04-18 |
| gson | 서버측 구현을 JAVA를 통해 개발할 계획이므로 자바 객체를 JSON화 시키기위해사용 | <https://github.com/google/gson> | 2017-04-18 |
| mongoDB | 크로스 플랫폼 도큐먼트 지향 데이터베이스 시스템으로 각 웹사이트별로 상이한 데이터들을 정형화시키기에는 무리가있음으로NOSQL기반 몽고디비를 사용한다. | <https://www.mongodb.com/> | 2017-04-18 |

4.3. 협업 Tools

소스코드와 문서 작업은 GitHub을 이용해 형상관리를 한다.

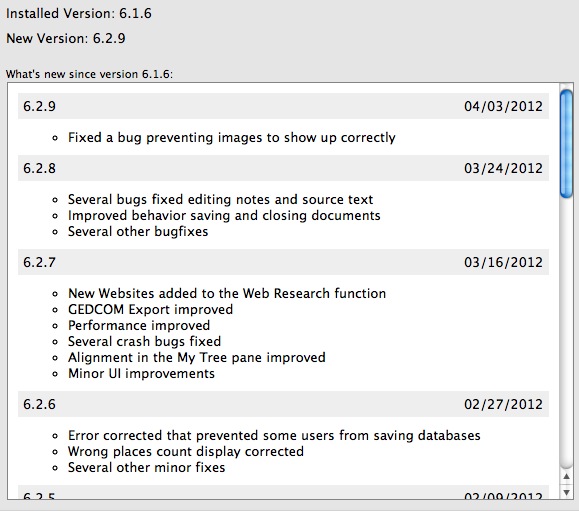
README 파일을 구성하여, 사용 메뉴얼 및 소프트웨어 정보와 라이선스를 제공한다.

<https://github.com/sjkim2322/Mr_ROBOT>

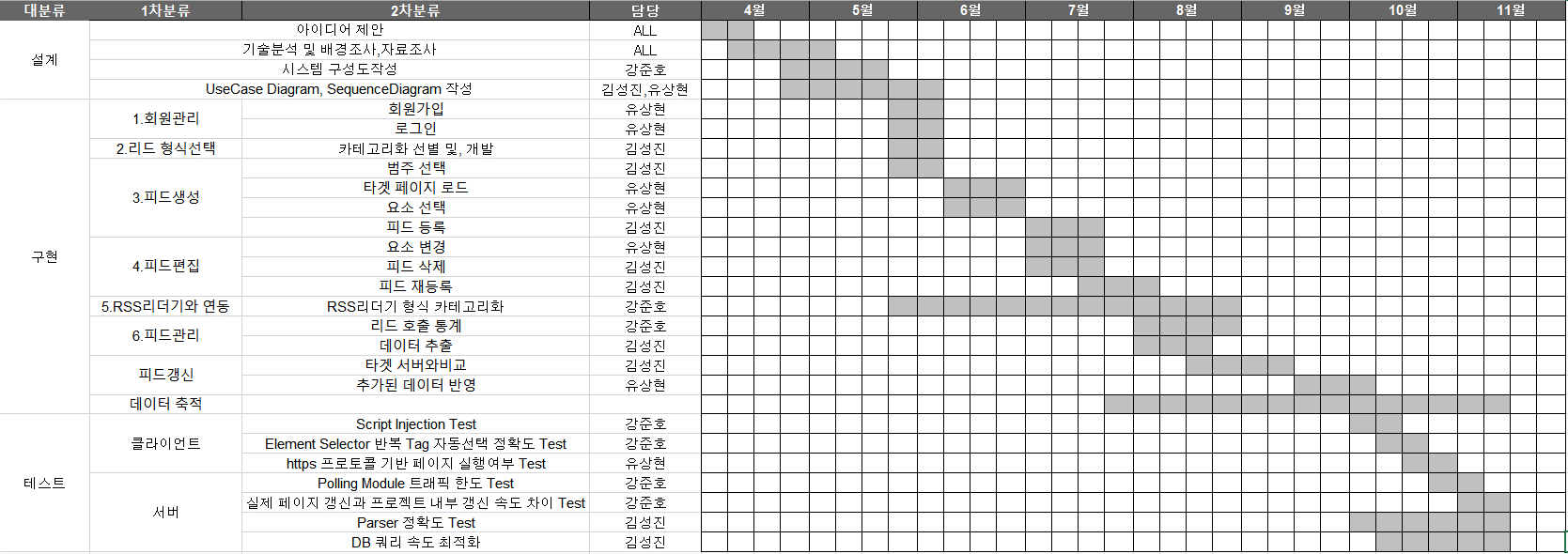
\*별도로 효율적인 형상관리 및 개발 일정 관리를 위해 GitHub 내의 ChangeLog 파일에 개발 일지를 상세하게 기술한다.

개발일지는 commit comment와 별도로 관리되며, 개발 기간 단위별로 추가 사항이나 개발 사항을 상세하게 기술하여, 아래 첨부한 예시와 같이 타임라인 형식을 유지하여, 직관적으로 변경 사항에 대한 이해를 돕는다.

예시)



4.3. Project 일정표



4.4. Project 개발환경

- 개발도구 : STS– SpringFramewor Developer Tools

: ATOM Editor

* 개발환경 : Chrome Browser

MongoDB v3.0

MySQL Server v5.0

Apache Tomcat v7.0

H2DB (for development)

5. Project 수행 결과

5.1. 실행 절차

5.2. 실행 결과

5.3. 한계 및 발전전략

1. Ajax로 구현된 사이트에 대한 데이터 수집의 문제

- Chrome 브라우저에서 태그선택을 통해 서버로 넘겨주는 과정은 모든 사이트에 대해서 공통적으로 적용되는 부분이다.

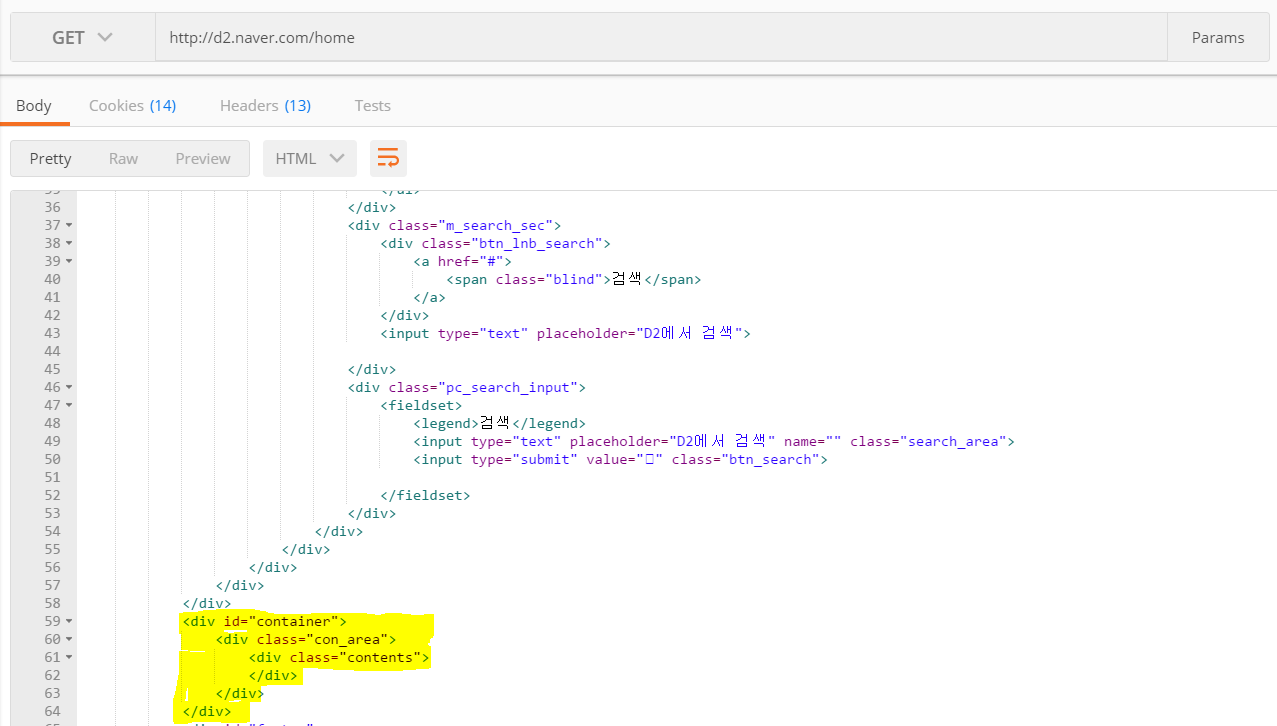
하지만 페이지 내부에 Data를 보여주는 과정에서 특정사이트들은 Ajax를 통해서 Data를 따로 가져오는 경우가 있다.

Naver D2사이트가 그 예를 단적으로 보여준다.



D2 사이트

브라우저를 통해서 D2 사이트에 접근할 경우 위의 그림과 같이 Data가 담겨 있는 것을 확인 할 수 있다.



포스트맨으로 요청시

하지만 서버내부에서 외부사이트로 접근을 할경우 브라우저를 통하지 않기 때문에 포스트맨에서 받아볼 수 있는 결과와 같은 결과를 받을 수 있다.

위의 그림처럼 포스트맨에서 같은 URI에 대해 요청을 할경우 Data를 담는 태그 부분이 비어있는 것을 확인 할 수 있다.

위와 같은 결과가 나오는 이유를 분선한 결과 사이트 자체적으로 Data를 가져오는 요청이 Ajax를 통해서 따로 구현이 되어있다는 것을 알게 되었다.

메인 URI만 가지고 Data를 수집하는 우리의 프로젝트의 특성상 Ajax를 통해서 Data를 가지고 오는 부분은 Data를 수집할 수 없다.

발전전략

* 서버 내부에 브라우저를 거처서 데이터를 가지고올 수 있도록 하는 방법을 모색중에 있다.
* 초기 Html 코드에 삽입된 js파일을 로드할 수 있도록한다면 위의 문제를 해결할 수 있다.