**소프트웨어 설계 및 실험 보고서 (5조)**

**앱 명 : 혹시나도...?**

**201624476 박상운**

**201624440 김연재**

**201424545 지승원**

**제출 일자 : 2018/06/25**

**개요**

앱 명 : 혹시나도

분 류 : 보조형 SNS

요 약 : 마이피누(mypnu.net) 반짝이 게시판에 올라온 글의 요소를 분석하여 DB에 태그를 추가하여 저장하고, 사용자의 신상정보 및 추가 입력 정보와 일치하는 요소들이 있는 글을 검색하여 보여주는 웹 기반 어플리케이션

상 세 : 마이피누 반짝이 게시판에서 제목만 보고 본인에 대한 내용이 없다고 판단하는 경우가 많이 있습니다. 혹은 시간이 없어서 대충 훑어보거나 못보고 넘어가는 경우도 있습니다. 이런 경우를 대비하여 마이피누 반짝이 게시판의 제목과 내용을 분석하여 본인에 대한 언급의 유무를 판단하는 어플리케이션입니다. 쉽게 설명을 하면 마이피누 반짝이 게시판에서 여러 가지 조건으로 검색하는 어플리케이션이라고 할 수 있습니다.

크롤러를 활용하여 마이피누 반짝이 게시판에 게시된 글을 수집합니다. 수집한 글에서 찾는 대상의 특성들과 글을 올린 시간, 해당 글의 URL 등 여러 가지로 분류합니다. 그리고 분류된 정보를 DB에 저장합니다.

사용자가 어플리케이션을 통해 입력한 사용자의 이름, 학번, 성별 등 기본적인 신상정보와 학과, 수강하는 과목 등 기본 신상정보에 해당하지 않는 다양한 키워드를 추가로 입력을 합니다. 입력된 데이터를 서버에 제공하여 DB에 저장된 정보들에서 해당 내용과 한 개 이상 일치하는 정보들을 검색합니다. 한 개 이상 일치해서 검색되어진 정보들은 마이피누 반짝이 게시판에 올린 시간 순으로, 즉 최신 순으로 정렬을 하여 출력을 합니다. 이때 DB에 저장한 URL을 활용하여 출력된 게시물의 제목을 클릭하면 반짝이 게시판으로 넘어가서 댓글을 달 수 있습니다.

**프로그램 기능(임시)**

1. 수집 기능

크롤링을 통하여 인터넷의 글을 수집합니다.

2. 분류 기능

수집한 데이터들을 원하는 조건으로 분류를 합니다.

3. 저장 기능

분류되어진 정보들을 DB에 저장합니다.

4. 검색 기능

사용자가 입력한 정보들을 토대로 DB에서 검색하여 원하는 결과를 가져옵니다.

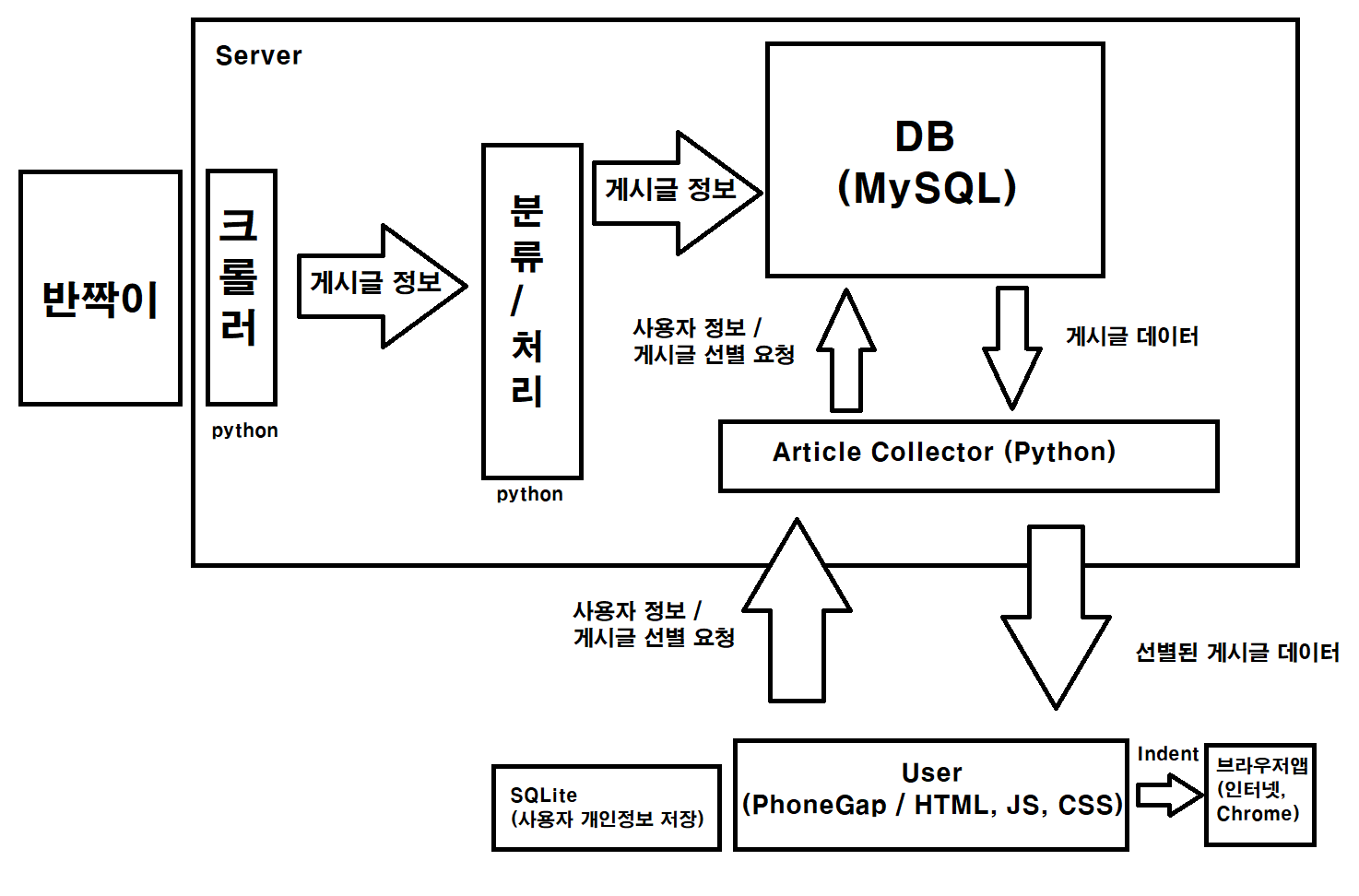
5. 표시 기능

검색한 결과들을 출력합니다.

6. 온라인 기능

이 어플리케이션은 인터넷과 연결되어 있고 서버가 켜져 있을 때면 어디서든지 접속하여 사용이 가능합니다.

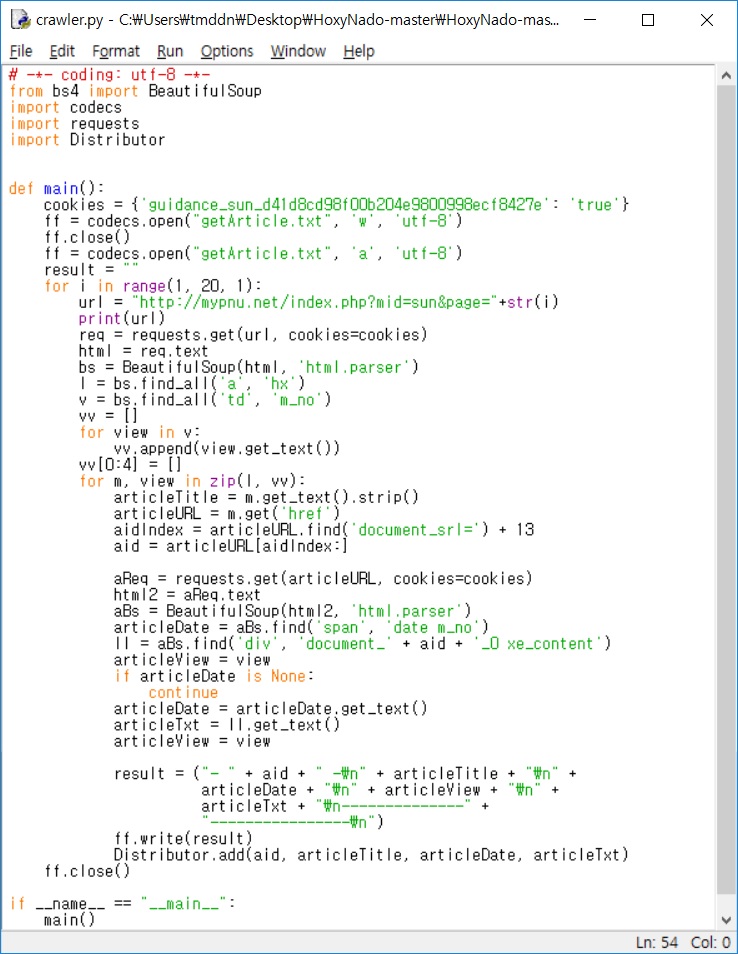
**파이프라인 변경 사항 (수정안했음)**

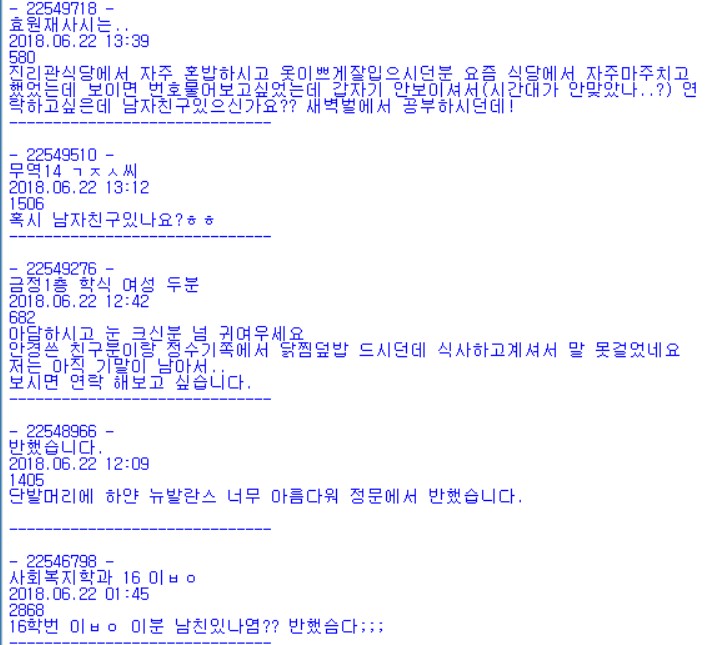


1. 백엔드 전부 Python (DB 접근 : PyMySQL, Article Collector로 Script Data 작성)
2. 불편하게 직접 만들지 말고 PhoneGap 쓰기
3. 게시글 수정에는 투자하지 말고 Indent 사용

**파이프라인에 따른 각 모듈 설명 – 크롤링 모듈**

1. **게시판 및 게시물 크롤링 // (제목 중 [파이프라인에 따른 각] 삭제 고민중**





크롤러 모듈의 경우 getArticle이라는 txt파일을 만들어서 크롤링 한 데이터를 저장합니다. 크롤링은 총 19회 즉 1~19페이지까지 할 예정으로 만들었으며 마이피누 반짝이 게시판의 경우 19페이지의 경우 평균적으로 1개월 이상의 분량입니다. 그래서 게시판과 어플리케이션의 특성 상 충분한 분량이기 때문에 19페이지까지 크롤링을 하게 됩니다. 크롤링의 경우 request를 통해 해당 URL의 HTML을 가져오며 beautifulsoup를 사용하여 해당하는 parser를 가진 html 코드를 전부 찾습니다.

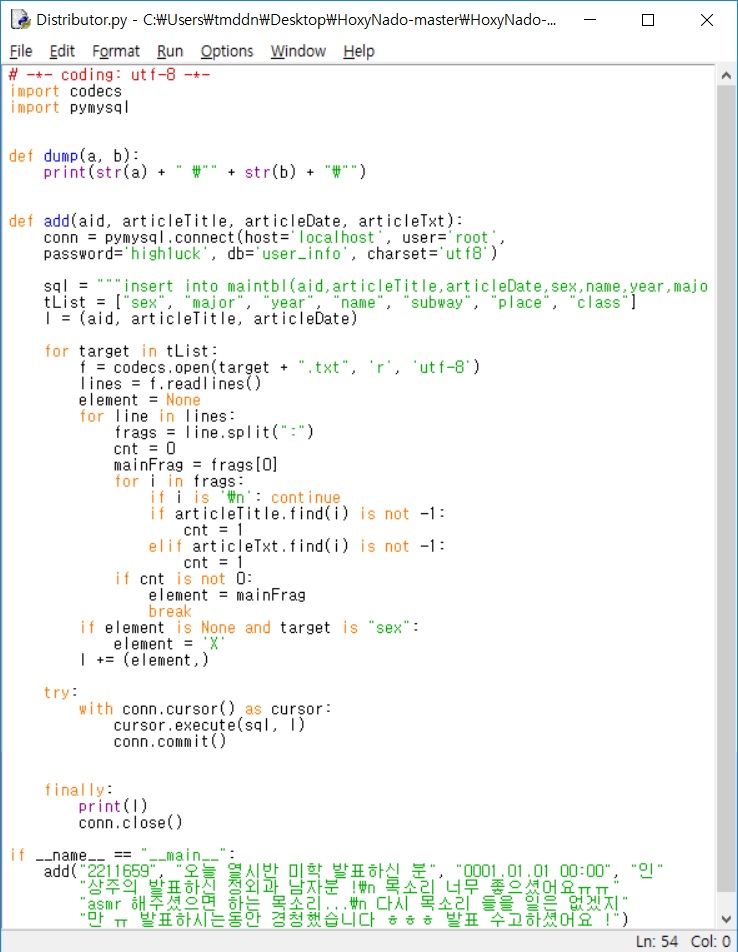
처음 게시판에서 크롤링을 할 때는 글의 제목이 있는 HTML 코드와 조회 수를 가져옵니다. 글의 제목을 가진 HTML 코드에서 반복적으로 게시물의 제목을 저장하고 해당 글의 URL또한 추출해서 게시물의 내용을 크롤링하도록 합니다. 그리고 게시물 URL에 포함된 게시물 ID를 추출하여 저장합니다.

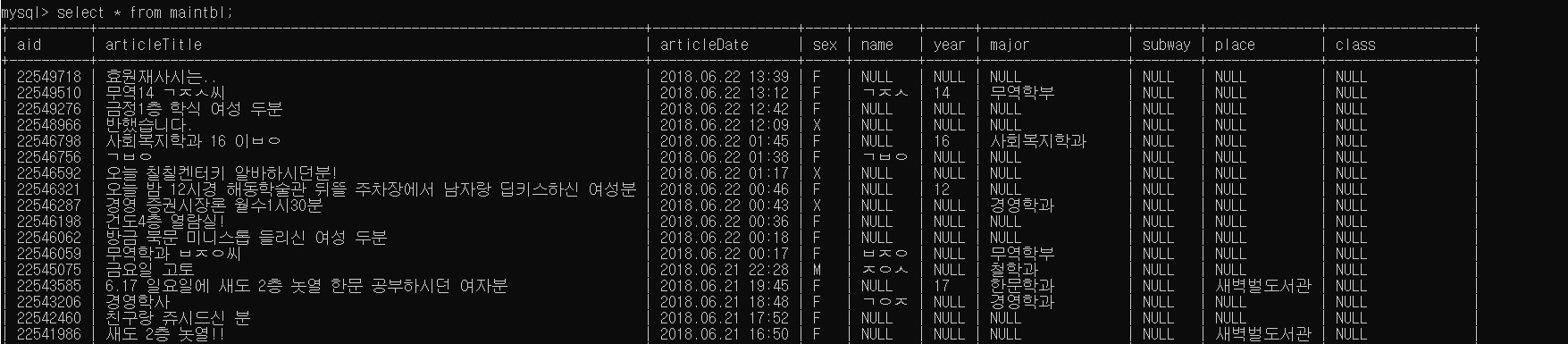
게시물의 내용에서는 게시물의 작성 날짜 및 시간과 게시물의 내용을 저장합니다. 이때 저장한 결과물들을 모두 getArticle.txt에 기록을 하면서 분류처리 모듈인 Distributor로 보내게 됩니다. 마지막으로 모든 과정이 끝나면 getArticle파일을 닫으면서 크롤러 모듈은 종료합니다.

위 화면은 크롤링한 데이터들의 결과물인 result를 print한 결과물 중 일부이며 글의 제목과 내용들이 분류처리 모듈에서 분류가 될 것입니다.

**파이프라인에 따른 각 모듈 설명 – 분류처리모듈**

1. **크롤링 된 데이터 분류 처리 // (제목 중 [파이프라인에 따른 각] 삭제 고민중**





우선 크롤링하여 나온 데이터들을 입력받습니다. 입력한 데이터를 분류 조건리스트에 저장된 내용들을 하나씩 반복하면서 분류를 진행합니다. 모든 분류조건들은 ‘ : ’ 으로 구분을 하고 해당 분류조건에서 줄이 바뀌기 전까지는 같은 소분류 조건으로 판단합니다. 만약 조건에 하나라도 부합한다면 같은 소분류조건의 처음에 나오는 조건을 메인으로 하여 element에 저장하면서 다음 분류조건으로 넘어갑니다. 성별의 경우 조건에 부합하지 못하여 element가 초기 값인 None가 그대로 유지 될 경우 ‘X’라는 값이 들어갑니다. 이렇게 분류되어 나온 값들은 처음에 분류되어 나올 값들을 저장할 튜플에 하나씩 저장이 되어 집니다. 이 튜플은 분류를 하지 않는 게시물ID(aid), 제목, 작성시간 및 날짜를 기본적으로 저장하며 분류되어서 나오는 값들을 하나씩 추가하여 최종적으로 10개의 값을 가지게 됩니다.

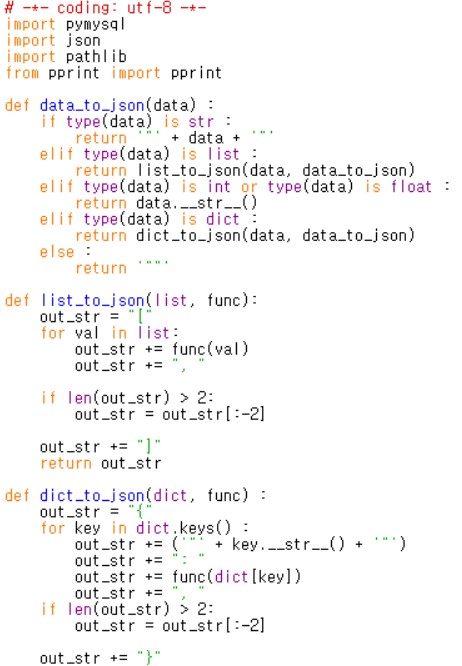
분류가 완료 되면 처음에 편의를 위해 만들어 놓은 conn을 사용하여 서버와 연결을 합니다. 그리고 마찬가지로 편의를 위해 만들어 놓은 sql을 사용하여 총 10개의 컬럼을 가진 maintbl에 문자열로 임시 저장용으로 만든 튜플의 값들을 저장합니다. 최종적으로 이러한 변경사항들을 commit하여 저장을 하고 서버와의 접속을 종료합니다.

만약 분류처리모듈을 모듈이 아닌 단독 파일로 실행 시 크롤링된 데이터가 없기 때문에 임의로 값을 추가하여야 합니다.

위 출력 화면은 Mysql의 서버에 있는 DB에서 분류된 값들이 저장되는 테이블인 maintbl을 출력하여 10개의 컬럼들의 값들이 출력이 된 화면입니다. 크롤링된 데이터는 위와 같이 분류 처리되어 최종적으로 DB에 저장이 되어 집니다.

**파이프라인에 따른 각 모듈 설명 – 검색모듈**

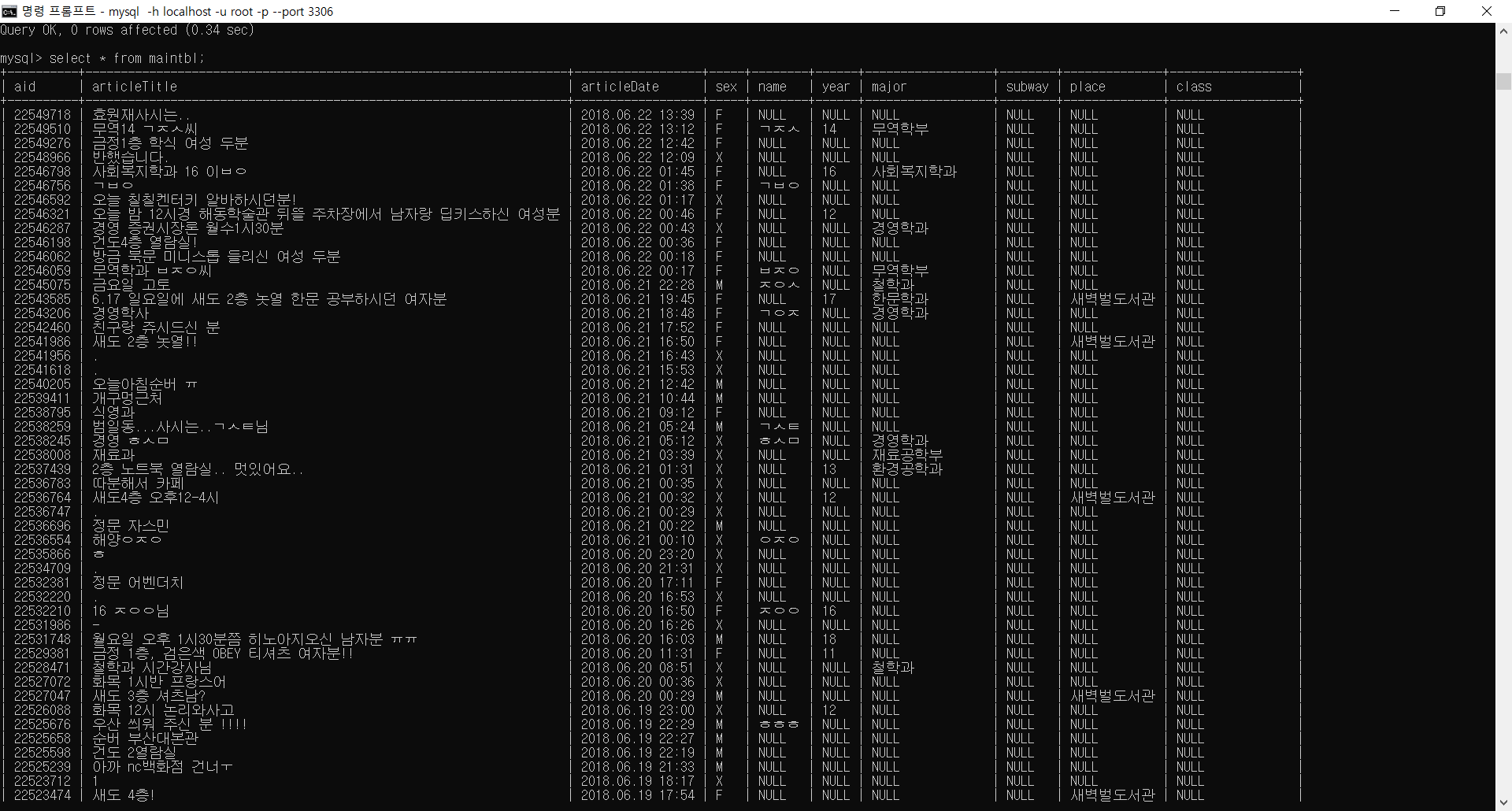
1. **DB에서 사용자로부터 받은 데이터를 검색**



...

**파이프라인에 따른 각 모듈 설명 – DB**

1. **크롤링한 데이터를 저장**

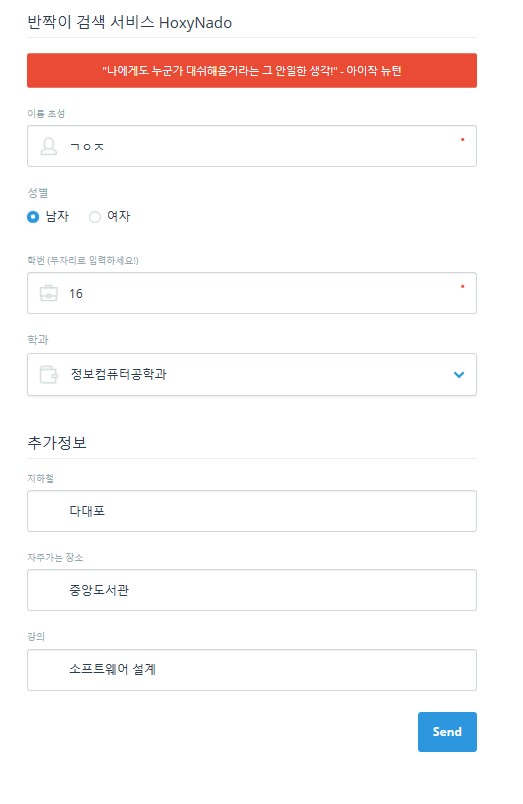
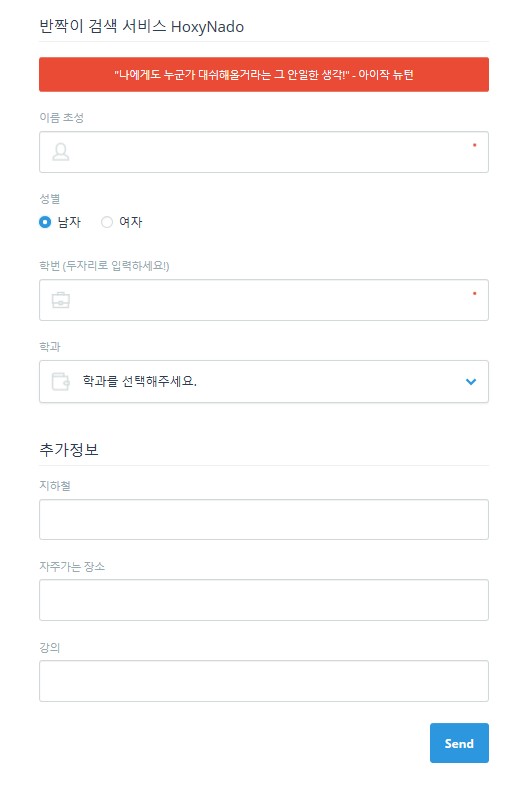


DB에 저장되는 내용들은 게시물에서 가져오는 aid(게시물 URL), 게시물 제목, 게시물 작성 날짜 및 시간뿐만 아니라 sex, name, year, major, subway, place, class 라는 총 7가지의 특성들을 저장합니다. 그리고 DB는 MYSQL을 사용하여 서버를 구축하고 DB를 구성합니다. Python을 사용하기 때문에 파일들은 pymysql을 통해 DB가 있는 서버로 접속을 합니다.

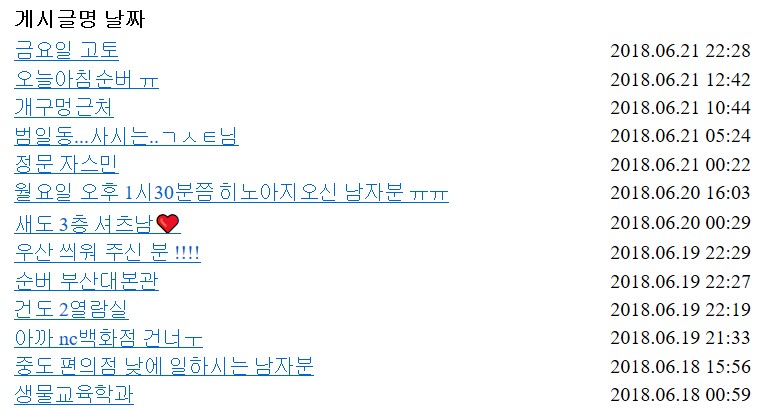
분류처리 모듈의 경우 pymysql로 서버-DB에 접속을 하여 분류 처리된 데이터들을 DB에 저장을 하고 검색 모듈의 경우 사용자가 어플리케이션에서 입력한 정보를 받아서 기존에 DB에 저장된 내용에서 검색하여 확인 후 일치하는 조건을 가진 게시물의 데이터를 DB에서 가져갑니다.

**실제 실행 예시**

실행 화면1 – 사용자가 정보 입력하는 화면



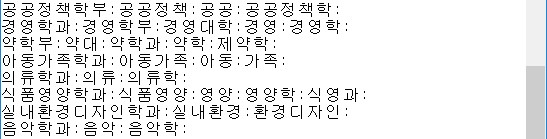
실행 화면2 – 사용자가 입력한 정보를 토대로 검색한 결과가 출력되는 화면



**개선해야하는 점**

1. 분류 조건 및 순서(?)

분류 조건 때문에 의도하지 않은 데이터가 분류되어 나올 수 있습니다. 현재 분류 조건들을 살펴보면 다음과 같습니다.



위 화면은 분류 조건 중 major의 일부분입니다. 학과 이름의 경우 줄임말로 사용하는 경우가 많아서 줄임말 또한 분류조건에 포함이 되어 있습니다. 이러한 줄임말들 중에서 실생활에서 학과이름 줄임말 이외에도 일상적으로 사용되는 언어들 또한 있습니다. 예를 들면 ‘공공시설’에서 공공이 분류되어 공공정책학부가 나올 수 있고 ‘음악을 들으면서 가는 여성분’ 이라는 글에서 음악에서 분류되어 음악학과로 분류되어 질 수 있습니다. 학과 이외에도 이름 초성에서 ㅋㅋㅋ라던가 ㅎㅎㅎ같이 웃음으로 주로 쓰는 초성들 또한 마찬가지로 이름으로 분류되어 질 수 있습니다. ㅋㅋㅋ같은 경우 사람이름으로 쓰이는 경우가 없다고 할 수 있지만 외국인 이름의 한글이 ㅋㅋㅋ가 될 수 있으며 ㅎㅎㅎ또한 사람이름에서 사용되는 초성이기 때문입니다. 이러한 가능성 때문에 예외적으로 해당 조건만 삭제 할 수 없다고 생각합니다. 이러한 부분은 더 많은 논의를 해서 해결을 해 나가야할 문제점 중 하나입니다.

이와 같은 문제와 연계되어 나타나는 또 다른 개선사항이 있습니다. 현 분류처리 모듈의 경우 조건에 부합하는 데이터가 있다면 그 자리에서 해당 조건의 분류를 종료합니다. 제일 처음 부합되는 데이터가 올바른 값이라면 전혀 문제가 없지만 글쓴이가 의도한 조건이 두 번째 혹은 그 이후에 나오는 값에 있다면 문제가 됩니다. 예를 들면 ‘어떤 남성분과 같이 가던 여성분’ 이라는 내용에서 글쓴이는 여자를 찾고 있으나 앞의 남성에서 이 게시물은 남자를 찾는 여자로 분류가 되어 해당 게시물은 어플리케이션을 사용하는 남성들에게 링크가 가서 남자와 남자가 연결되는 슬픈 일이 발생합니다. 이 문제는 대부분 분류조건에서 해결된다면 남녀 구별을 제외한 나머지의 경우는 해결이 가능할 것으로 예상되어집니다. 그러나 남녀 같은 경우에는 좀 더 자세한 분류과정이 필요할 것입니다.

**반성 및 느낀점 (내용 수정 안했음)**

