

지리정보기반 취업추천사이트의 설계 및 개발
Design and development of geographic information based job
recommendation system

서대우 (경영정보과), 이재호 (경영정보과)
Dae-Woo Seo (Dept. of Management Information System),
Jae-Ho Lee (Dept. of Management Information System)

Key Words : 잡매칭, 취업추천, 지리정보, LBS

ABSTRACT : When the youth unemployment rate is approaching its peak, we provide job matching services at various job search sites and the need for appropriate referrals is increasing in universities. In a survey of students in our department, commuting time and distance were found to be important factors to support the recruiting company. However, research on job matching, including commuting time and distance, is still a mere step, and many workloads are required for actual job applicants to select job applicants with large address books. Therefore, in this study, the address book of the applicant or the student is displayed on the map, and the distance is displayed according to the job offerer or the central position, thereby reducing the work burden and designing and developing a service capable of various types of analysis.

요약 : 청년실업률이 최고치에 육박하는 이 때, 다양한 구인구직 사이트에서 잡매칭을 위한 서비스를 제공하고 있으며 대학에서도 업체의 구인 의뢰에 대한 적절한 추천의 필요성이 점점 높아지고 있다. 우리 학과의 학생들을 대상으로 한 설문조사에서 구인 업체의 지원 요인으로 통근 시간이나 거리가 중요한 변수로 나타났다. 그러나 통근 시간이나 거리를 포함하는 잡매칭에 대한 연구는 아직 미미한 단계이며, 실제 취업 담당자가 방대한 주소록으로 취업 추천자를 선별하기 위해서는 많은 업무량이 따른다. 따라서 이 연구에서는 구인자 또는 학생의 주소록을 지도 위에 표시하고, 구인 업체나 중심 위치에 따라 거리를 표시함으로써 이러한 업무 부담을 덜 수 있고 나아가 다양한 형태의 분석이 가능한 사이트를 설계 및 개발하였다.

1. 서론

근래 연애, 결혼, 출산을 포기한다는 3포 세대, 내집 마련과 인간관계까지 추가로 포기한다는 5포 세대, 여기에 꿈과 희망을 포기한다는 7포 세대(국민일보, 2015.08.28) 등의 용어는 한국 청년들이 직면한 어려운 현실을 대변하는 말이다. 이러한 청년들의 각종 포기 상태는 개인적인 문제이기도 하지만 사회적 측면에서 우리나라의 안정성을 해칠 수 있는 위험 요소라 할 수 있다. 국가통계포털 자료에 따르면 2017년 4월을 기준으로 15 ~ 29세 청년층의 실업률인 청년실업률은 동월 기준으로 1998년 4월의 외환위기 이후 역대 최고치인 11.2%를 기록하고 있다(국가통계포털(KOSIS), 검색일: 2017.06.10).

남진열(2010)은 청년실업 문제의 원인으로 노동 공급측면과 수요측면의 원인, 산업수요와 대학교육의 불부합 및 일자리 불부합 등을 지적하였다(1). 이러한 청년실업 문제에도 불구하고 워크넷의 잡매칭 성공률은 극히 낮은 수준이다(2). 박성익(2012)에 따르면 취업 알선의 성공률이 낮은 것은 적합한 구직자나 기업을 우선 선별하지 않고 정보의 홍수 속에서 너무 많은 정

보가 유통됨에 따라 구직자는 적합한 기업을 선별하기 어렵고 기업은 구직자 정보를 스팸메일 정도로 치부하는 상황에 기인한다고 판단했다. 따라서 성공적인 잡매칭을 위해서는 무조건 많은 양의 정보가 필요한 것이 아니라 정보탐색비용을 최소화하면서 기업과 주민에게 적합한 맞춤형 정보를 제공하는 것이 필요하다.

한편, 잡매칭에 관한 선행 연구들에서는 희망직업, 학력, 희망임금, 기업규모, 근로시간, 출근시간 등의 조건을 잡매칭의 변수로 활용하고 있다(3),(4). 그러나 용인송담대학교 경영정보과 학생들을 대상으로 한 취업 시 고려사항에 대한 설문조사에서는 급여수준, 통근시간(거리), 발전가능성, 근무시간, 복리후생 등의 순서로 응답하였다(복수응답). 이러한 연구의 예로서 김석주(2015) 등이 지리정보체계와 다층분석을 활용한 연구를 진행한 바 있다(5). 그러나 잡매칭 또는 취업 추천에 활용할 데이터로서 거주지와 근무지 간의 거리에 대한 변수로의 활용은 미미한 실정이다.

따라서 이 연구에서는 잡매칭 또는 취업 추천 시 지리 정보를 시각적으로 활용할 수 있는 도구를 설계 및 개발한다. 이 연구의 결과는 우리 대학 졸업예정 학생들의 취업 추천 뿐만 아니라 지원자, 신입생 등에 필요한 지리 정보에 기반하여 분석할 수 있는 도구로서 활용할 수 있다.

2. 기존 연구 및 자료 수집

조인호(2009)에 따르면 워크넷의 구인·구직시스템이 하드매칭방법을 이용하여 검색조건이 복잡해지면 해당 구인정보가 안 나오는 문제가 존재한다는 지적을 하고 있다(6). 아울러 새로운 직업이 나타나거나 정확한 직업 명칭을 모르는 경우에는 해당 구인정보를 제공하지 못할 수도 있다는 문제를 지적하고 있다. 그리고 오성욱(2009)은 스웨덴의 인터넷 매칭 취업서비스를 소개하고 있다. 그의 연구에서는 고용서비스 기능 강화를 한 신기술(소프트매칭) 도입 및 운영 부분을 설명하고 있다(7).

최근 들어 이와 같이 일자리 매칭에 있어서 소프트매칭을 활용해야 할 필요성이 강조있는데, 소프트매칭은 사용자의 심리가 반영된 중요도가 높은 조건과 항목별 상관관계가 높은 정보에 가치를 부여함으로써, 사용자의 의도에 가장 근한 결과를 찾아주는 방법이다. 소프트매칭의 장점으로서는 항목에 대한 중요성과 조건(유사직종, 학력, 거리 등)에 대한 허용(tolerance) 정도에 관한 상담원의 경험과 노하우(Know-how)를 시스템에 반영할 수 있다는 점이다. 즉 소프트매칭에서는 구직자가 반경 2km 내의 일자리를 찾겠다고 하여도 임금이 높다면 4km 거리의 일자리도 구직자에 따라 수용이 가능함을 의미한다. 이와 같이 많은 연구가 소프트매칭의 요성을 강조하고 있지만, 중요도 높은 조건에 대하여 어떻게 가치를 부여해야 하는지에 대해서는 아직 연구가 이루어진 것이 없다. 예외적으로 박성익 외(2009)가 잡매칭함수를 개발하여 그 적용 가능성을 연구한 바가 있다. 그러나 그들의 잡매칭함수는 성, 연령과 같은 모든 특성이 동일한 정도로 중요한 것이 아님에도 불구하고 전부 동일한 가치를 부여하고 있다는 문제점이 있다.

한편 박성익 외(2012)에서는 잡매칭 함수를 개발하는데 있어 성, 연령, 학력, 자격증 등의 변수와 함께 희망근무지나 거리에 따른 함수를 개발한 바 있다. 이 연구에 따르면 거리가 멀수록 취업확률 값은 작아지는 경향이 뚜렷이 나타났다.

위와 같은 기존 연구에서 논의된 각종 변수들에 대하여 용인송담대학교 경영정보과 재학생을 대상으로 구직 시 필요한 조건에 대하여 설문조사를 실시하였다. 설문조사는 용인송담대학교 경영정보과의 1학년과 2학년 전체 재학생을 대상으로 온라인으로 실시하였으며, 조사 대상 인원 45명 중 41명이 응답하였다.

설문조사의 주요 내용으로는 성별, 학년, 취업의사, 취업 시 고려할 점, 급여 수준, 가능한 통근 시간, 현재 준비 중인 자기개발 항목 등이다. 이 중 취업 시 고려할 점은 3개 문항 이내에서 복수 응답으로 실시하였다. 응답 인원 중 남학생은 43.9%, 여학생은 56.1%의 비율이었으

며, 세 번째 문항의 “취업을 하였거나 할 예정이라면 무엇을 가장 중요하게 생각하십니까? 상위 수준 3개를 선택해 주기 바랍니다”에 대한 결과는 Table 1과 같으며, 차트로 표현하면 Fig. 1과 같다.

Table 1. Response to work considerations

문항 번호	질문	응답수	응답인원 대비 비율
1	급여수준 - 회사에서 지급하는 급여 수준이 적정해야 한다.	26	63.41%
2	통근 시간이나 거리 - 가능한 가까운 근무지이어야 한다.	18	43.90%
3	수행하는 직무 - 내가 하고 싶은 일이어야 한다.	7	17.07%
4	기업 안정성 - 평생 근무할 수 있는 안정된 기업이어야 한다.	8	19.51%
5	기업 인지도 - 누구나 알만한 회사이어야 한다.	1	2.44%
6	발전가능성 - 회사 또는 직무가 향후 지속적인 발전을 해야 한다.	14	34.15%
7	자아성취 - 회사 근무를 통해 자아 성취를 할 수 있어야 한다.	1	2.44%
8	직무 적성 적합성 - 내 적성과 직무가 부합해야 한다.	8	19.51%
9	능력 적합성 - 나의 능력에 알맞은 직무를 수행해야 한다.	8	19.51%
10	근무환경 - 내 마음에 드는 근무 환경이어야 한다.	7	17.07%
11	근무지 주위 환경 - 점심시간 또는 퇴근 후 시간 활용 시 주위의 환경이 중요하다.	2	4.88%
12	근무시간 - 근무시간을 지키고, 주말은 쉬어야 한다.	13	31.71%
13	복리후생 - 다양한 복리후생 제도를 운영해야 한다.	10	24.39%



Fig. 1 Graph of responses to job considerations

또한 “도보, 대중교통, 자가용 이용 등을 통틀어 출퇴근 시간은 어느 정도까지 가능하십니까?”에 대한 응답은 Table 2와 같다.

Table 2. Preferred commute time

항목	응답인원	비율
30분 이내	2	4.9%
30분 ~ 1시간	31	75.6%
1시간 ~ 1시간 30분	5	12.2%
2시간 이내	0	0.0%
기숙사 희망	0	0.0%
관계 없음	3	7.3%
계	41	100%

Table 2에서 보는 바와 같이 전체 응답 인원 41명 중 80.5%인 33명이 1시간 이내의 출퇴근 시간을 희망하는 것으로 나타났으며 이는 구직 시 거리가 중요한 요소임을 반영하는 것이다.

위와 같은 선행 연구 및 설문조사 결과에 따라 취업 추천 시 구인 업체와 취업희망자 거주지 사이의 거리를 주소 목록이 아닌 시각적인 도구로 쉽게 파악하고, 거리에 따른 제약 조건을 보다 빠르게 파악할 수 있는 사이트를 개발하고자 한다.

3. 지리정보기반 취업추천사이트의 설계 및 구현

앞에서 본 바와 같이 설문조사 결과에 의하면 통근시간이나 거리가 중요한 변수임을 알 수 있다. 구직자 즉 학생을 기업에 추천하기 위한 기업의 조건은 구인 의뢰 시 이미 결정되어 있다. 직무, 연봉수준, 우대조건 등은 대부분 구직자가 변경할 수 있는 사항이 아니다. 그러나 지리적인 거리는 구직자가 여러 개의 구인 업체들 중 하나를 선택할 수 있다. 따라서 이러한 중요한 조건을 주소록만으로 파악한다면 검색 시 상당한 시간이 소요될 것이다. 그러므로 구인 업체를 중심으로 주소록이 아닌 지도 위에 학생 또는 구직자의 위치를 표시하고 이를 거리에 따라 검색한다면 취업 추천 시 업무를 보다 빠르게 추진할 수 있을 것이다.

한편 많은 IT 업체에서는 지도를 위한 API를 누구나 사용할 수 있도록 개방하여 활용할 수 있다. 네이버 개발자 페이지에서는 지도, 파파고, 로그인 서비스 등을 제공하고 있으며, 구글 개발자 페이지에서도 지도를 위한 API 이외에 방대한 양의 API를 이용할 수 있다(8)(9). 이외에도 다양한 업체에서 지도와 관련된 API를 개방하여 사용할 수 있다. 이러한 API를 이용하여 취업 추천 시 또는 대학의 의사 결정을 위한 다양한 용도로 활용될 수 있는 지역기반 취업 추천 사이트를 설계 및 구현한다.

시스템이 제공해야 할 기능을 먼저 살펴보면, 필수적으로 제공해야 하는 기능은 구직자 또는 학생의 주소를 포함한 기본 정보 입력, 구인업체 또는 중심이 되는 곳의 주소를 포함한 기본 정보 입력, 입력된 정보를 지도 위에 거리별로 표시 등이며, 부가적으로 자료의 일괄 입력, 기 입력 자료를 활용한 지도 다시 보기, 구인업체 또는 중심이 되는 곳의 명칭과 주소 변경하기 등의 기능을 제공해야 한다. 이와 같은 기능을 구현하기 위한 사이트의 개발 환경은 다음과 같다.

- OS 및 웹 서버 : Synology NAS (Linux based)
- 개발 언어 및 데이터베이스 : PHP5, SQLite v3, Javascript, HTML, CSS
- 지도 API : 네이버(NAVER) 지도 API v3

이 기능을 구현하기 위해서 필요한 최소한의 데이터베이스 테이블 설계는 Table 3 및

Table 4와 같다.

Table 3. Table structure of centerLoc table

번호	필드명	자료형	크기	비고
1	cserial	int	primary key	일련번호
2	userid	char	25	time+ip
3	name	varchar	60	업체명
4	address	varchar	128	업체주소
5	phone	char	14	업체전화
6	lat	float		위도 East
7	lng	float		경도 North
8	isradius	int	0 1	동심원출력여부
9	radius	int		동심원간격(m)
10	note	text		메모
11	isline	int	0 1	라인출력여부
12	date	timestamp		입력일시
13	title	text		제목

Table 4. Table structure of Data table

번호	필드명	자료형	크기	비고
1	dserial	int	primary key	일련번호
2	userid	char	25	time+ip
3	name	varchar	60	업체명
4	address	varchar	128	업체주소
5	phone	char	14	업체전화
6	lat	float		위도 East
7	lng	float		경도 North
8	centerLoc	int	foreign key	중심위치
9	distance	float		중심까지거리
10	type	char(20)		객체타입
11	note	text		메모
12	gender	text		성별
13	date	timestamp		입력일시

사이트 네비게이션 설계는 Fig. 2와 같으며, 각 페이지의 역할은 Table 5와 같다.

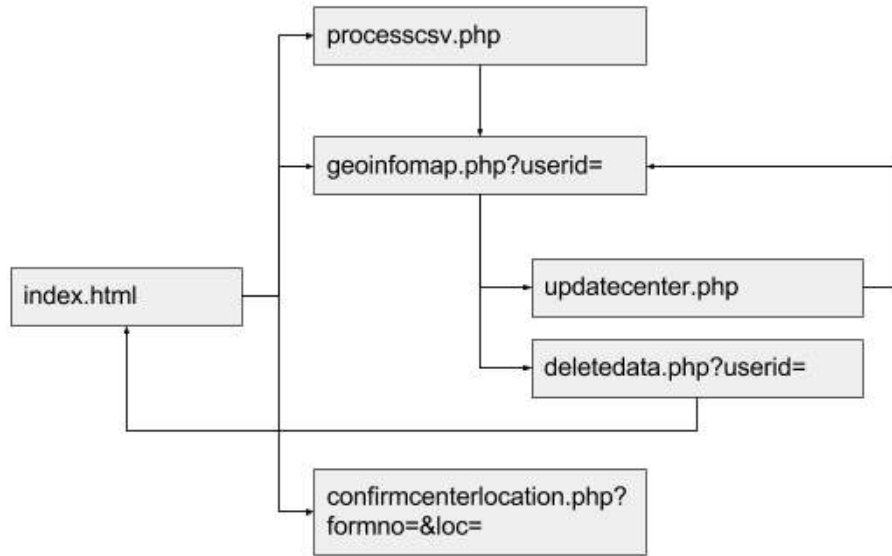


Fig. 2 Site navigation design

Table 5. Role of each page

파일명	역할	설명
index.html	초기화면	<ul style="list-style-type: none"> 이전에 입력한 자료를 기반으로 한 지도로 이동 새로운 데이터를 CSV 파일 형식으로 업로드
processscsv.php	CSV 데이터 처리	<ul style="list-style-type: none"> 사용자가 업로드한 데이터에서 주소를 경위도 좌표로 저장 데이터를 데이터베이스 테이블로 저장 Unique ID 생성
geoinfomap.php	지도 출력	<ul style="list-style-type: none"> 사용자의 Unique ID에 따른 데이터를 지도 위에 출력 중심 좌표 위치를 이용한 반경으로 검색 중심좌표 변경 데이터 입력
updatecenter.php	중심좌표 저장	<ul style="list-style-type: none"> geoinfomap.php로 부터 전달받은 중심 좌표를 이용하여 새로운 경위도 검색 및 데이터베이스 테이블 갱신
deletedata.php	자료 삭제	<ul style="list-style-type: none"> 중심좌표 및 그에 따른 전체 데이터를 삭제
confirmcenterlocation.php	중심좌표 확인	<ul style="list-style-type: none"> 초기화면에서 입력한 중심의 주소를 확인하여 출력

사용자가 입력한 데이터를 데이터베이스 테이블에 저장하기 위한 순서도는 Fig. 3과 같다. 이 순서도에서 알 수 있듯이, 두 지점의 위경도 좌표를 이용한 거리 계산을 위해서는 먼저 주소를 위경도로 변환하는 방법과, 거리 계산을 위한 수식이 필요하다. 주소를 위경도 좌표로 변환하기 위해서는 네이버의 주소 → 좌표변환 API를 사용한다(10).

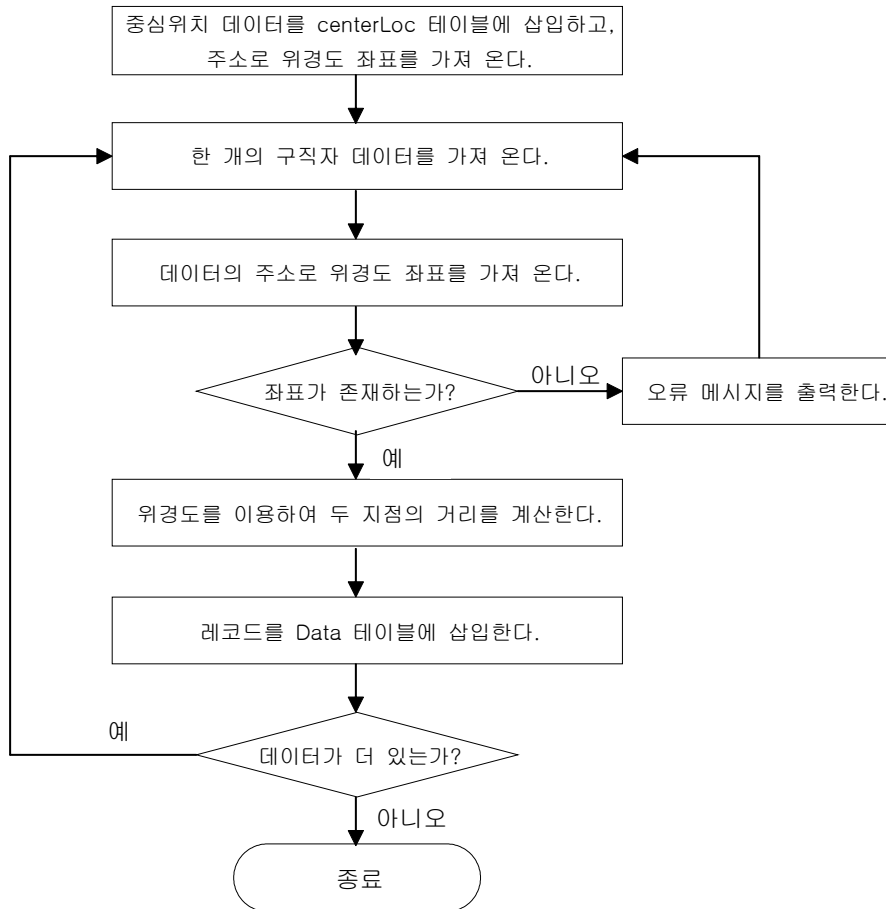


Fig. 3 Flowchart for storing input data

이 API를 이용하여 주소를 좌표로 변환하여 JSON 데이터로 받아서 처리하는 PHP 코드의 예는 다음과 같다.

```

function getcode($addr)
{
    $ch = curl_init();
    $address = urlencode($addr);
    $encoding = "utf-8";
    $coord = "latlng";
    $output = "json";
    $qry_str =
"?encoding=".$encoding."&coord=".$coord."&output=".$output."&query=".$address;
    $headers = array(
        "X-Naver-Client-Id: NAVER_CLIENT_ID",
        "X-Naver-Client-Secret: NAVER_CLIENT_SECRET");
    $url = "https://openapi.naver.com/v1/map/geocode";
    curl_setopt($ch, CURLOPT_URL, $url.$qry_str);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $res = curl_exec($ch);
    curl_close($ch);
    return $res;
}
  
```

용인송담대학교의 JSON 데이터의 반환 예는 다음과 같다.

```
{
  "result": {
    "total": 1,
    "userquery": "용인시 처인구 마평동 571-1",
    "items": [ {
      "address": "경기도 용인시 처인구 마평동 571-2",
      "addrdetail": {
        "country": "대한민국",
        "sido": "경기도",
        "sigugun": "용인시 처인구",
        "dongmyun": "마평동",
        "rest": "571-2"
      },
      "isRoadAddress": false,
      "point": {
        "x": 127.2181818,
        "y": 37.2291954
      }
    } ]
  }
}
```

위 함수를 이용하여 JSON 데이터에서 위경도를 가져 오려면 PHP에서 다음과 같이 호출한다.

```
$res = json_decode(getcode($address),true);
$lat1 = $res['result']['items'][0]['point']['y']; // 위도
$lng1 = $res['result']['items'][0]['point']['x']; // 경도
```

위도와 경도로 나타낸 두 지점사이의 거리를 계산하기 위해서는 'Haversine' 공식을 사용한다. 이 공식은 지구의 표면에서 가장 짧은 거리를 계산하며, 공식은 수식 (1) 및 (2)와 같다 (11).

$$\text{Haversine 공식 : } a = \sin^2(\Delta\phi/2) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2(\Delta\lambda/2) \quad (1)$$

$$\begin{aligned} \text{거리 계산을 위한 방정식 : } \quad c &= 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \\ d &= R \cdot c \end{aligned} \quad (2)$$

여기에서 ϕ 는 위도(latitude), λ 는 경도(longitude), R 은 지구의 반지름(6,378,137m)이다. 이 공식을 자바스크립트로 구현하면 다음과 같다.

```
function GetDistance(cy, cx, y, x)
{
```



```

var radius = 6378137;
var dLat = Math.abs((y-cy)*Math.PI/180.0);
var dLon = Math.abs((x-cx)*Math.PI/180.0);
var y1 = y*Math.PI/180.0;
var cy1 = cy*Math.PI/180.0;
var a = Math.pow(Math.sin(dLat/2),2) + Math.pow(Math.sin(dLon/2),2)
        * Math.cos(y1) * Math.cos(cy1);
var c = 2 * Math.asin(Math.sqrt(a));
var dDistance = radius * c;
return dDistance;
}

```

이 사이트를 구현하기 위하여 사용한 네이버 지도 API의 클래스는 Table 6과 같다.

Table 6. Naver Maps API class

기능	사용 클래스
지도 출력	naver.maps.Map
학생의 위치에 마커 출력	naver.maps.Marker
거리를 나타내는 원 출력	naver.maps.Circle
중심점과 학생들의 위치 사이 직선 출력	naver.maps.Polyline

4. 실험 및 고찰

개발한 사이트의 URL은 “http://211.195.54.208/geoinfo/index.html”이며, 접속한 초기화면의 예는 Fig. 4와 같다.



Fig. 4 Initial Screen

이 화면에서 졸업예정자의 데이터를 UTF-8 인코딩의 CVS 파일 형식으로 데이터를 업로드하여 주소 좌표 변환이 완료되면 Fig. 5와 같이 전체 학생의 주소를 지도 위에 출력한다. 단, 이 논문에서는 학생들의 개인정보 보호를 위하여 이름을 “* * *”으로 출력하도록 설정하였다.

CSV 파일의 데이터는 이름, 주소, 전화번호, 이메일, 메모, 성별 순으로 작성해야 한다. 이 데이터에 의해 생성된 지도는 마우스 스크롤 또는 확대 및 축소 버튼을 사용하여 확대와 축소가 가능하며, 그 예는 Fig. 6과 같다. 학생의 이름 및 거리를 출력하는 마커 위에 마우스를 올리면 툴팁의 형태로 학생의 이름, 주소, 전화번호가 보여진다.

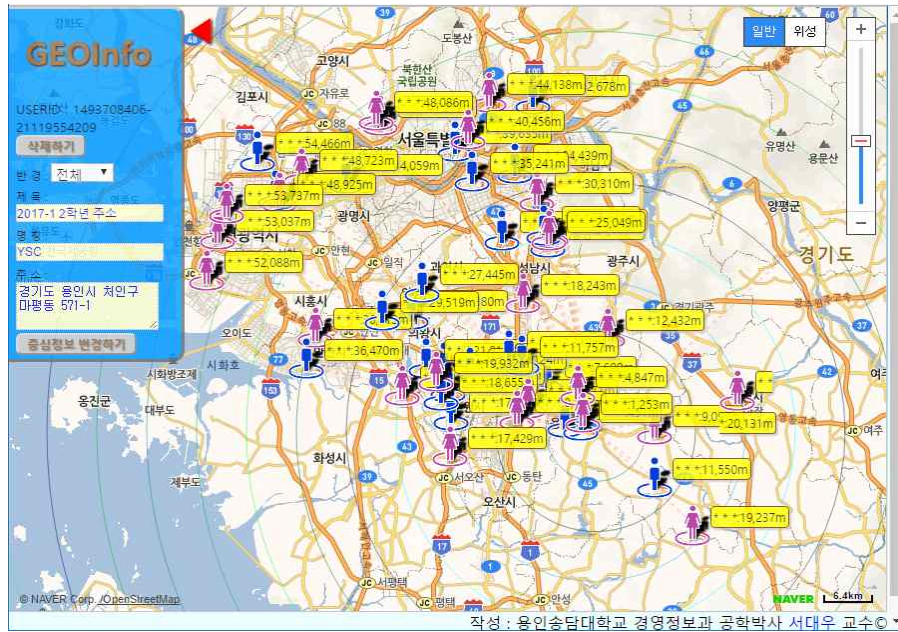


Fig. 5 Screen for all students on the map

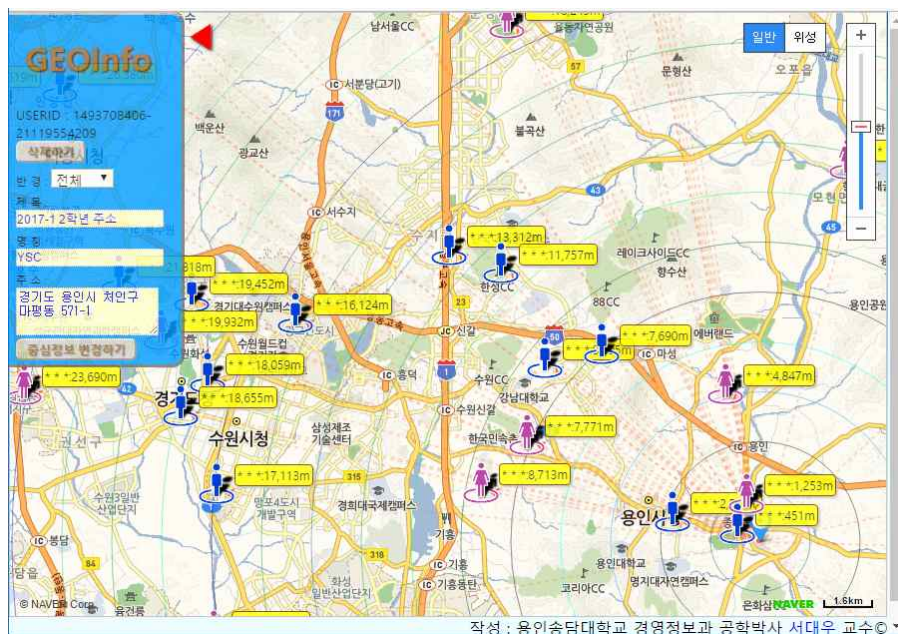


Fig. 6 Zoom-in screen

여기에서 특정 반경 내의 학생들만 검색하여 보려면 좌측 상단의 반경을 선택하면 Fig. 7과 같이 선택된 반경 내의 학생만 지도에 출력하게 된다.

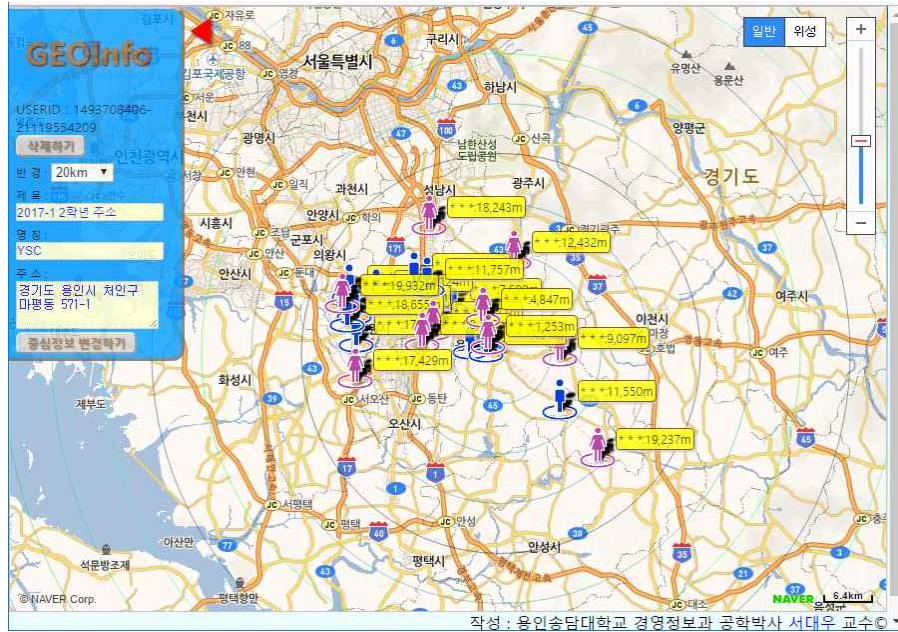


Fig. 7 Show only students within a radius of 20 km

어느 화면에서나 학생의 이름과 거리를 표시하는 영역을 클릭하면 Fig. 8과 같이 중심점과 학생 위치 간의 대중교통 경로를 출력할 수 있다. 이를 이용하면 구직자의 경우 구인업체까지 대중교통을 이용한 대략적인 통근 시간을 파악할 수 있다.

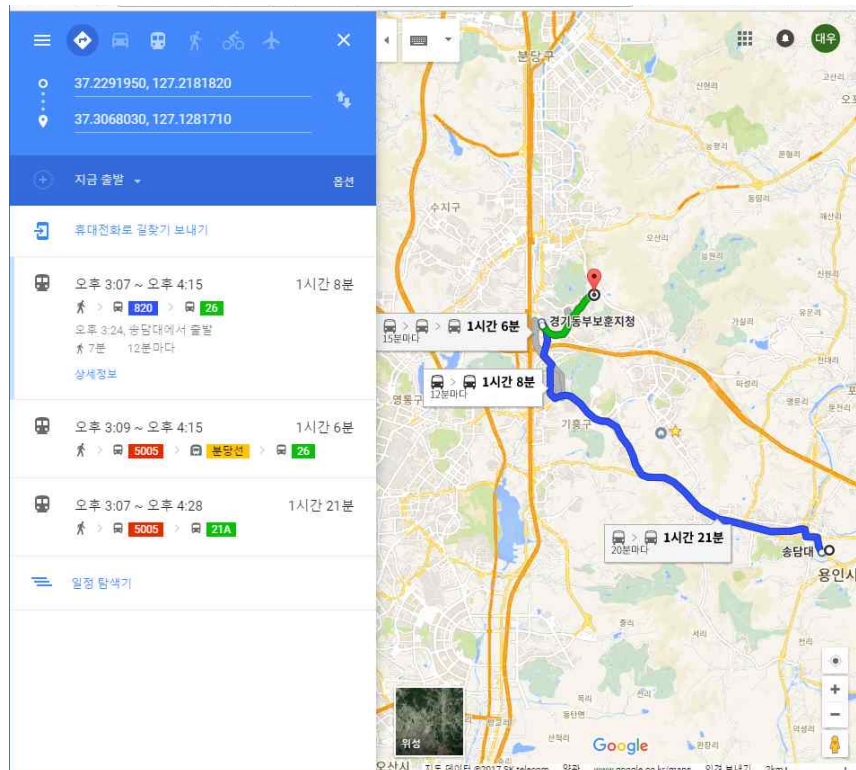


Fig. 8 Bus route map for a specific student

버스 노선의 출력은 구글지도를 이용하였으며, 두 지점의 경위도를 이용하여 간단한 링크로 대중교통을 이용한 두 지점간의 다양한 노선을 확인할 수 있다. 링크의 사용법은 아래와 같다.

```
http://maps.google.com/maps?saddr=출발지위도,출발지경도&daddr=도착지위도,도착지경도
```

Fig. 9는 취업 추천뿐만 아니라 다른 용도로도 활용하는 예로, 2017학년도 경영정보과 정시 1차 지원자의 분포를 나타낸 것이다. 이를 유선을 통한 입시 상담에도 활용할 수 있다. 또한 재학생의 멘토링 면담 시에도 활용하는 등 다양한 분야에서 활용할 수 있다.

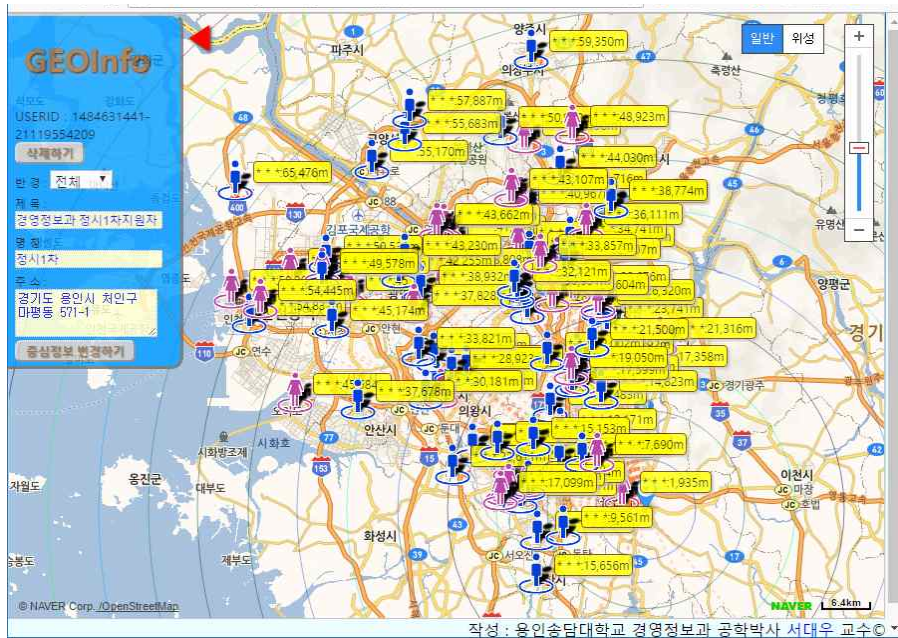


Fig. 9 Distribution of the first applicants in the management information department of Yong-in Songdam College

5. 결론 및 향후 연구 방향

우리 대학 경영정보과 학생을 대상으로 한 설문조사 결과, 직장과의 거리가 구인 업체를 선정하는데 중요한 변수가 됨을 확인할 수 있었다. 그러나 일반적인 잡매칭 사이트에서는 거리를 주소로 확인할 수 밖에 없는 불편함과 함께, 구인 의뢰를 처리할 때에도 주소록에 의존하여 큰 거리의 취업 추천자를 가리는데 많은 시간을 소요하고 있는 실정이다. 따라서 구직자가 취업을 할 때는 일반적인 잡매칭의 여러 변수에 더하여 통근 거리를 시각적으로 확인할 수 있는 도구를 활용하여 잡매칭의 효율을 극대화할 수 있다.

이 논문에서는 구직자 또는 학생들의 주소에 기반한 지리정보기반의 취업추천 사이트를 설계하고, 이의 프로토타입을 교내의 서버 상에서 동작하도록 구현하였다. 졸업예정자의 데이터를 기반으로 주소만 정확하다면 지도 위에 해당 학생의 위치를 표시하고, 거리까지 잘 표시함을 확인하였다. 또한, 일정 반경 이내의 학생들을 간단히 검색할 수 있음을 확인하였다. 또한 두 지점 사이의 경위도 좌표를 이용하여 대중교통편과 시간을 확인할 수 있었다. 이를 활용하면 구인 업체에 대한 취업 추천 뿐만 아니라 재학생이나 입시지원자의 지리적 분포를 한 눈에 파악할 수 있어 다양한 방향으로 활용이 가능하다.

향후 연구 방향으로는 로그인, 또는 자료 출력 시 암호를 입력해야 하는 등의 시스템의 보

안 측면 강화, 전체 졸업예정자를 대상으로 하거나, 입시분석 등 대량의 자료에 사용할 수 있도록 빅데이터의 처리 및 동심원 그래프 등 다양한 형태의 그래프로 출력, 우리 대학 학사 시스템에 연동하여 데이터를 자동으로 부여하도록 구현하는 등의 추가 설계 및 개발이 필요할 것이다. 또한 위 사이트를 아이폰이나 안드로이드폰 등의 스마트폰에서 원활히 연동되도록 확장하고, 터치 시 전화를 걸어주거나 문자를 보낼 수 있는 기능을 추가하여 사용자 편의성을 확대해야 할 것이다.

참고문헌

- (1) 남진열, 2010, 대학생의 개인적 직업 관련요인들과 취업준비상태의 관계연구, 청소년복지연구 제 12권 제4호 pp353~375
- (2) 박성익 외, 2012, 구인·구직조건에의 유연성을 고려한 잡매칭함수의 개발, 노동정책연구, 제12권 제3호 pp25~51
- (3) 조장식 외, 2011, 사회연결망분석을 이용한 잡매칭함수 분석, 한국통계학회논문집, 제18권 제6호, pp675~685
- (4) 박성익 외, 2010, 잡매칭을 위한 취약계층 주민 실태조사-부산 고용촉진지구 사례를 중심으로, 산업관계연구, 제20권 제3호, pp139~163
- (5) 김석주, 2015, 지리정보체계와 다층분석을 활용한 지역사회복지연구, 한국지역사회복지학회, 제54권, pp291-312
- (6) 조인호, 2009, 취업정보 알선/매칭 효과성 제고를 위한 소프트매칭 및 직업 유사성 연구, 미출판, 한국 고용정보원
- (7) 오성욱, 2009, 스웨덴의 강화된 인터넷 매칭 취업서비스 및 정책시사점, 직업과 고용서비스연구, 제4권 제1호, pp39~48
- (8) 네이버(NAVER) 개발자 페이지, <https://developers.naver.com>
- (9) 구글(GOOGLE) 개발자 페이지, <https://developers.google.com>
- (10) 네이버 지도 API 명세 페이지, <https://developers.naver.com/docs/map/overview/>
- (11) Calculate distance, bearing and more between Latitude/Longitude points, <http://www.movable-type.co.uk/scripts/latlong.html>