

“바람이 불지 않으면 노를 저어라”

**유 지 훈**



## profile

- 이름 : 유지훈 (Ryu Ji Hun)
- 학적사항 : 동국대학교 정보통신공학과
- 생년월일 : 1996.03.11
- 전화번호 : 010-5766-6393
- Email : nangam123@naver.com
- Github: <https://github.com/wlgns61>

# 목차

1	경유지를 포함한 택시 요금 계산 및 경로 탐색 어플리케이션
2	YOLO v3를 사용한 비전기반 스마트 보행 시스템
3	Spring Framework5.0을 사용한 블로그 만들기
4	Ruby On Rails로 인터넷 뉴스 포털 사이트 구현
5	기타 사항

# 01

경유지를 포함한 택시 요금 계산 및 경로 탐색  
어플리케이션

# 1-1

## 경유지를 포함한 택시 요금 계산 및 경로 탐색 어플리케이션 프로젝트 개요

출발지: 대전광역시 유성구 지족동 218-...

경유지: 반석역 대전1호선 외 3개

목적지: 경유지 중 택1

\*원하는 항목을 선택하세요.

**최소 비용 탐색**  
택시 요금: 22760원  
시간: 0시간 56분    거리: 21.0km

**최소 거리 탐색**  
택시 요금: 22760원  
시간: 0시간 56분    거리: 21.0km

**최소 시간 탐색**  
택시 요금: 22760원  
시간: 0시간 56분    거리: 21.0km

**입력 순서 탐색**  
택시 요금: 40710원  
시간: 1시간 19분    거리: 35.0km

뒤로 가기

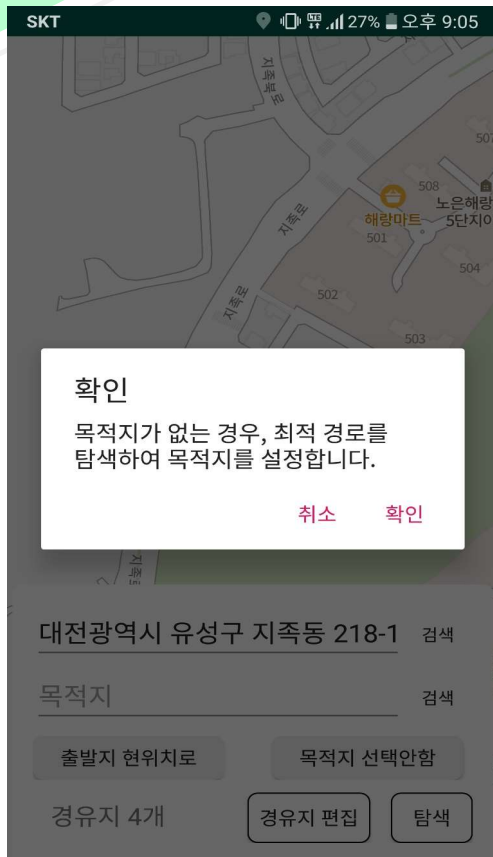
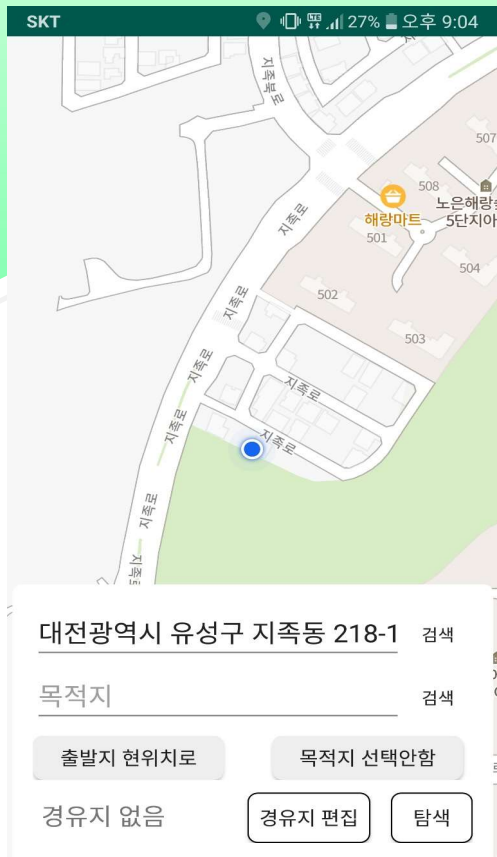
**최소 비용 탐색**  
택시 요금: 22760원  
시간: 0시간 56분  
거리: 21.0km

상세 경로 보기

- 경유지의 순서를 랜덤하게 입력하였을 때 출발지로부터 모든 경유지를 방문하는 순서를 결정하는 어플리케이션
- Back Tracking 알고리즘을 사용하여 최소 비용, 최소 거리, 최소 시간 기준으로 탐색
- Naver Map API v3, Place Search, Geocoding, Reverse Geocoding, Driving 5의 Open REST API사용
- Driving API 호출 시 Multi-Thread를 사용하여 로딩 시간을 단축시킴
- 기능1: 목적지 없이 경유지만 입력 된 경우 탐색 결과에 따라 경유지 중 한 지점을 목적지로 결정
- 기능2: 목적지와 경유지가 모두 입력되어 있는 경우 모든 경유지를 방문하고 최종 목적지까지의 탐색 결과를 제공(경유지가 없어도 무방)
- 동영상: <https://youtu.be/H2z7Wu-ZvM4>
- 소스코드: <https://github.com/wlgns61/AndroidOptimalPathSearch>

# 1-2

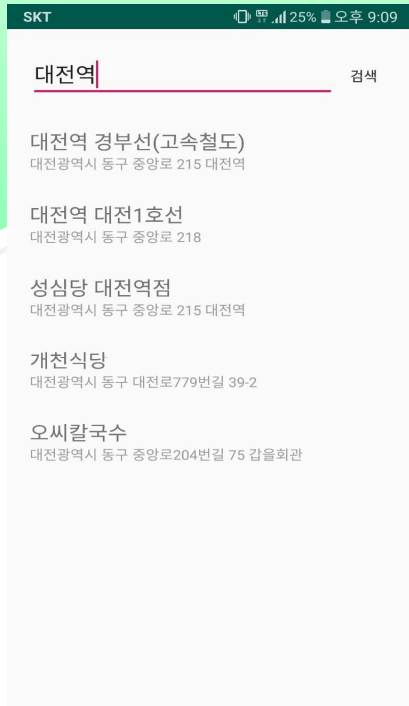
## 경유지를 포함한 택시 요금 계산 및 경로 탐색 어플리케이션 Main Activity



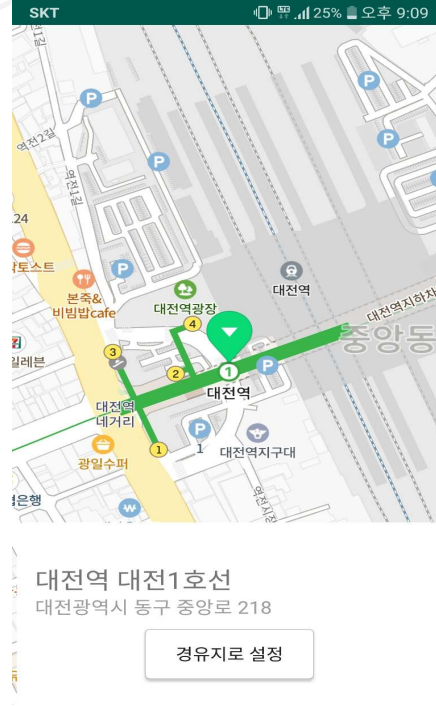
- GPS로 알아낸 좌표를 통해 Reverse Geocoding API를 사용하여 자신의 현재 위치를 출발지로 초기화
- 출발지, 목적지 모두 검색을 이용하여 설정 가능
- 경유지 편집 버튼을 사용하여 경유지를 추가 및 삭제 가능
- 탐색 버튼을 클릭 시 입력 된 출발지, 목적지, 경유지의 정보로 탐색
- 만약 최종 목적지를 선택하지 않을 시 우측 이미지와 같이 Confilm Dialog를 통해 확인

# 1-3

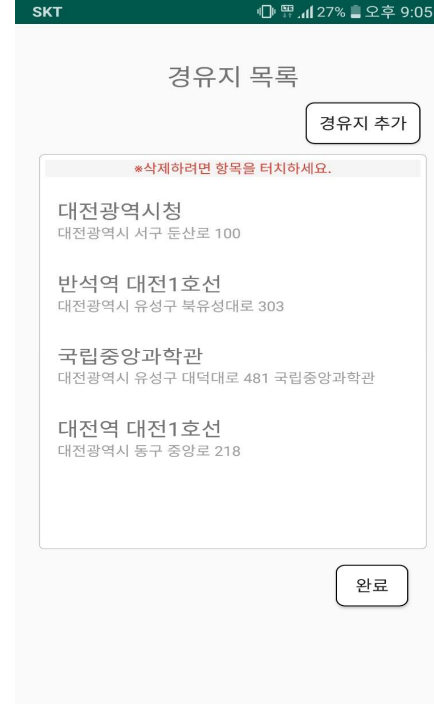
## 경유지를 포함한 택시 요금 계산 및 경로 탐색 어플리케이션 검색 및 경유지 설정



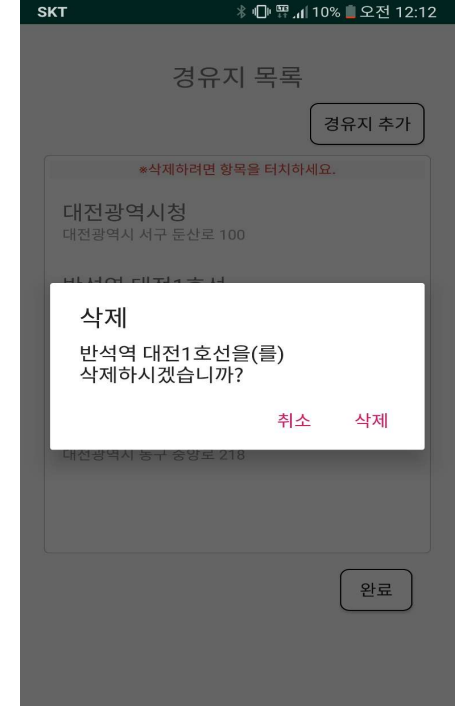
키워드 입력 후 검색 버튼 클릭 시  
Place Search API, Geocoding API를  
사용하여 검색 결과를 사용자에게 제공



List의 Item을 클릭 시 선택  
한 지점을 지도로 시각적인  
정보 제공



경유지는 최대 4개 까지 설정  
가능 하며 Item을 터치 할 시에  
항목 삭제 가능



항목을 터치하였을 때  
Confirm Dialog를 통해  
재확인

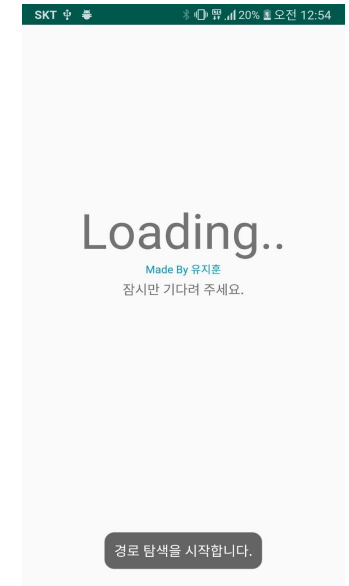
# 1-4

## 경유지를 포함한 택시 요금 계산 및 경로 탐색 어플리케이션 탐색 알고리즘과 Multi-Thread

```
private void backTracking(ArrayList<Integer> arr, Boolean[] visited, int cnt, int N){  
  
    if(cnt == N){  
        ArrayList<Integer> tmp = new ArrayList<>(arr);  
        this.permutation.add(tmp);  
        return;  
    }  
  
    for(int i=0;i<N;i++){  
        if(visited[i] == false){  
            visited[i] = true;  
            arr.add(i+1); //출발지가 0이므로 1이상의 숫자만 저장  
            backTracking(arr, visited, cnt + 1, N);  
            visited[i] = false;  
            arr.remove(index: arr.size() - 1);  
        }  
    }  
}
```

Back Tracking Algorithm을 사용하여 1~N번 경유지에 대한 가능한 순열을 구하는 알고리즘

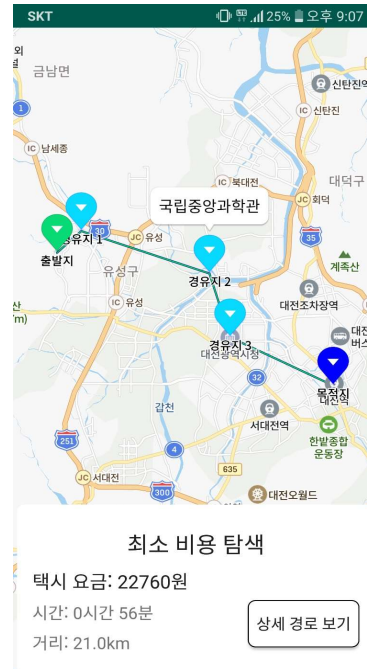
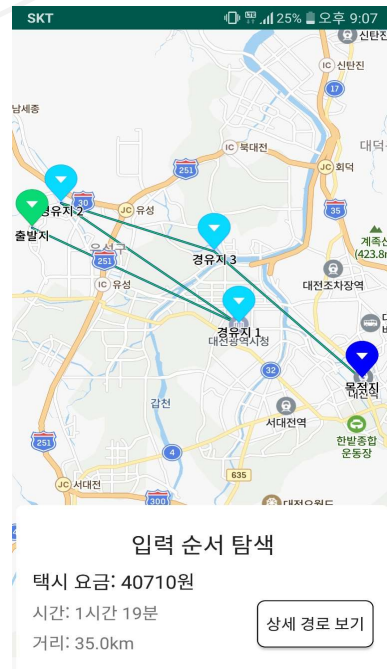
```
ArrayList<Thread> threads = new ArrayList<>();  
  
for(int i=0;i<permutation.size();i++){  
    threads.add(run() -> {  
        fromToInfos.add(makeInfo(paths.get(++idx)));  
    }));  
}  
  
for(int i=0;i<threads.size();i++){  
    Thread thread = threads.get(i);  
    thread.start();  
}  
  
for(int i=0;i<threads.size();i++){  
    try{ threads.get(i).join(); }  
    catch (Exception e){ e.printStackTrace(); }  
}
```



- 알고리즘의 결과를 통해 Driving API로 요청 할 수 있는 형태로 변환
- thread들을 동시에 실행시켜 API호출을 하고 모든 thread가 동작을 완료할 때 까지 대기
- 알고리즘 실행과 Multi-Thread가 작업을 완료하는 동안 로딩 화면을 사용자에게 보여줌
- Single-Thread: 평균 15초 소요 -> Multi-Thread: 평균 5초 소요

# 1-5

## 경유지를 포함한 택시 요금 계산 및 경로 탐색 어플리케이션 탐색 결과 Activity



탐색의 결과로 최소 비용, 최소 거리, 최소 시간, 사용자가 입력한 순서 4가지를 알 수 있으며 Item을 클릭 시 자세한 정보 제공

- 위 2개의 이미지는 사용자가 입력한 순서에 따른 결과와 알고리즘을 적용하여 계산한 최소 비용 탐색의 결과
- 보다 직관적인 결과를 제공하기 위해 직선으로 각 지점을 연결

상세 경로 보기를 터치하여 경로의 세부사항을 확인 할 수 있으며 지도에 표시된 마커를 터치하여 지점의 이름을 확인 가능



# 02

YOLO v3를 사용한 비전기반 스마트 보행 시스템

# 2-1

## YOLO v3를 사용한 비전기반 스마트 보행 시스템 프로젝트 개요



- YOLO v3를 활용하여 카메라로 사람을 인지하여 횡단보도의 신호등을 제어 하는 프로젝트.
- 기존의 보행 시스템은 교통량이 적은 지역의 교통 흐름에 방해가 될 수 있음.
- 사전 훈련되어 제공되는 YOLO weights를 사용하지 않고 오직 '사람' 만을 인식 하도록 재훈련.

### 동영상

<https://www.youtube.com/watch?v=ECp3WX20-AU>

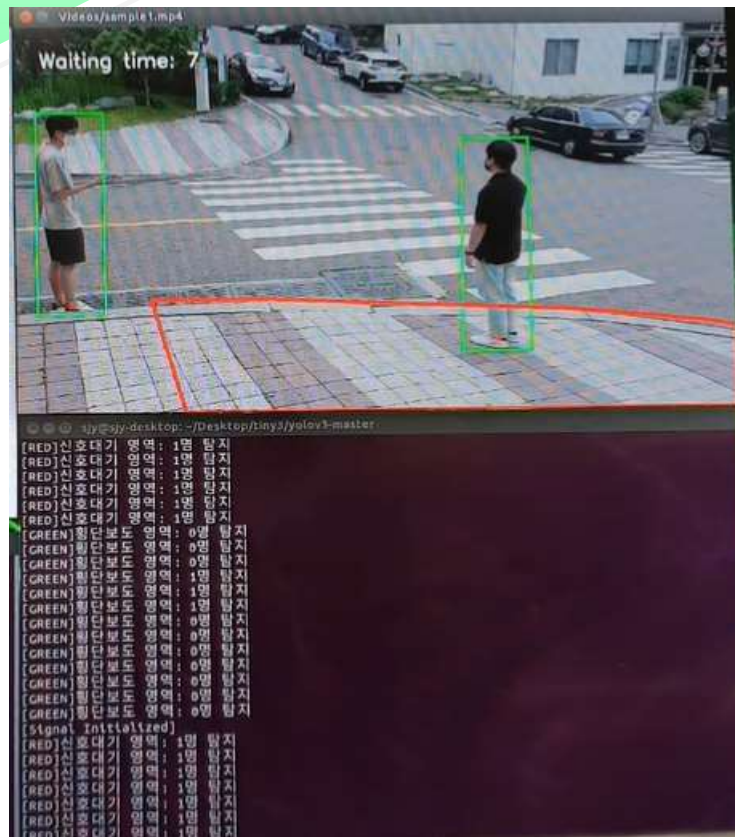
### 소스코드

<https://github.com/wlgns61/SmartCrossboardSystem>

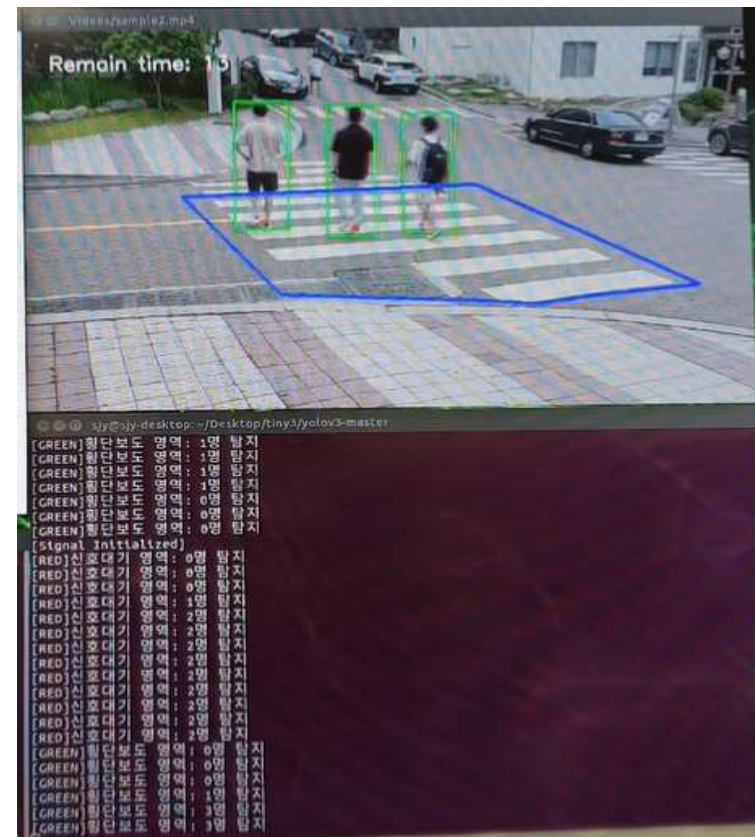
# 2-2

## YOLO v3를 사용한 비전기반 스마트 보행 시스템 신호에 따른 탐지 영역 변화

보행자 신호가 빨간 불인 경우



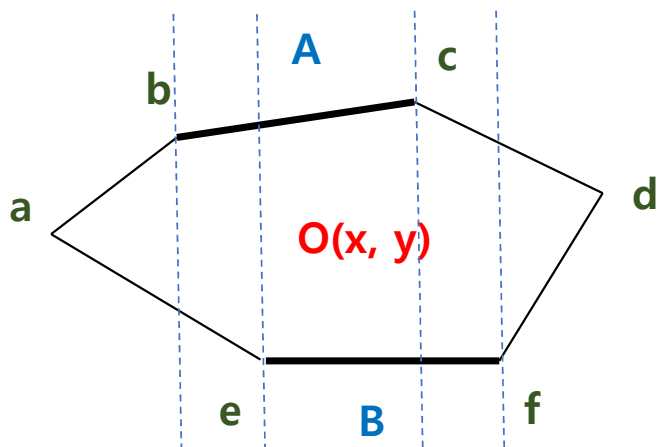
보행자 신호가 파란 불인 경우



# 2-3

## YOLO v3를 사용한 비전기반 스마트 보행 시스템

### 영역 내 객체 탐지 알고리즘 구현



- 영역을 여러 개의 꼭지점을 가진 다각형으로 생각함.
- 영역의 윗부분을 나타내는  $up\_boundary[] = \{a, b, c, d\}$ , 밑부분을 나타내는  $low\_boundary[] = \{a, e, f, d\}$ 의 배열로 표현.
- 탐지된 객체  $O$ 의  $x$ 좌표가  $up\_boundary$ 의 어떤 두 점  $[b, c]$  사이에 있는지 탐색 및  $down\_boundary$ 의 어떤 두 점  $[e, f]$  사이에 있는지 탐색.
- $up\_boundary$ 에서 탐색된 두 점  $[b, c]$ 를 이은 선분을  $A$ 라 하고,  $down\_boundary$ 에서 탐색된 두 점  $[e, f]$ 를 이은 선분을  $B$ 라 하자.
- 탐지된 객체  $O$ 의  $y$ 좌표가 만약 선분  $[선분 B] \leq y \leq [선분 A]$  를 만족한다면 객체는 영역안에 존재, 조건을 만족하지 않는다면 영역 밖에 존재.

# 03

**Spring Framework5.0을 사용한 블로그 만들기**

# 3-1

## Spring Framework5.0을 사용한 블로그 만들기 프로젝트 개요

JIHOON STORY My Info Category Admin Logout

admin님 환영합니다.

all

	글제목	작성자	조회수	작성일
자유게시판	<a href="#">테스트2</a>	abcd1234	0	2020-09-01
자유게시판	<a href="#">테스트</a>	abcd123	5	2020-09-01
자유게시판	<a href="#">ㅎㅎㅋㅋ</a>	admin	0	2020-09-01
공지사항	<a href="#">ㅎㅎㅎㅎㅎ</a>	admin	1	2020-09-01
공지사항	<a href="#">zzz</a>	tony6516	8	2020-08-30
공지사항	<a href="#">ㅎㅎ</a>	admin	3	2020-08-30
공지사항	<a href="#">ㅎㅎ</a>	admin	2	2020-08-30
공지사항	<a href="#">ㅎㅎ</a>	admin	1	2020-08-30
공지사항	<a href="#">ㅎㅎ</a>	admin	2	2020-08-30
공지사항	<a href="#">ㅎㅎ</a>	admin	1	2020-08-30

금쓰기

1 2 3

제목  검색

- Spring Framework 5.0를 개인적으로 공부하며 만든 블로그
- Gradle 빌드 툴과, Java configuration을 기반으로 시작.
- MySQL, MyBatis를 사용하여 데이터베이스 설정 및 연결.
- 회원가입, 로그인, 카테고리, 페이지네이션, 검색과 기본적인 CRUD를 갖추고 있는 게시물, 댓글 기능 구현.
- 스프링 시큐리티, REST, 비동기 통신, 트랜잭션을 적절히 사용
- 동영상: [https://www.youtube.com/watch?v=d-f\\_6qGeWSM](https://www.youtube.com/watch?v=d-f_6qGeWSM)
- 소스코드: <https://github.com/wlgns61/MyBlogSpring5>



# 3-2

## Spring Framework5.0을 사용한 블로그 만들기

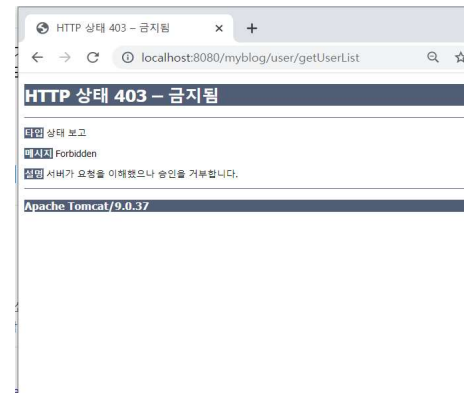
### Spring Security

/myblog/user/getUserList 요청 예시

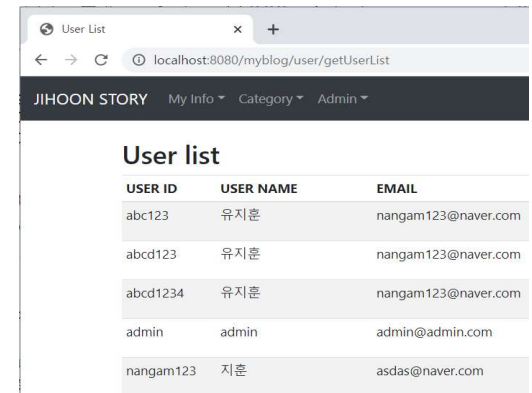
```
@Override
protected void configure(HttpSecurity http) throws Exception {

    CharacterEncodingFilter filter = new CharacterEncodingFilter();
    filter.setEncoding("UTF-8");
    filter.setForceEncoding(true);

    http
        .authorizeRequests()
            .antMatchers("/board/**").authenticated()
            .antMatchers("/user/getUserList*").hasRole("ADMIN")
            .antMatchers("/menu/**").hasRole("ADMIN")
            .and()
        .formLogin()
            .loginPage("/login/loginForm")
            .loginProcessingUrl("/login/doLogin")
            .defaultSuccessUrl("/board/getBoardList")
            .usernameParameter("uid")
            .passwordParameter("pwd")
            .failureUrl("/login/loginForm?error=true")
            .permitAll()
            .and()
        .logout()
            .logoutUrl("/logout")
            .invalidateHttpSession(true)
            .logoutSuccessUrl("/login/loginForm")
            .and()
        .csrf()
            .ignoringAntMatchers("/logout")
            .and()
        .addFilterBefore(filter, CsrfFilter.class);
}
```



일반 회원의 권한으로 접근한 경우



관리자의 권한으로 접근한 경우

- 관리자, 회원, 비회원으로 역할을 구분 지어 페이지의 주소에 접근 할 수 있는 요청을 제한함.
- 사용자의 개인정보 등이 담겨 있는 데이터를 컨트롤러로 전송시에는 URL에 정보가 담기는 것을 방지하기 위해, GET대신 POST방식을 사용하며 CSRF토큰을 같이 전송하도록 하였음.

# 3-3

## Spring Framework5.0을 사용한 블로그 만들기

### REST와 비동기 통신

회원가입

아이디

abc123

사용 불가능한 ID 입니다.

중복검사

이름

이름을 입력해 주세요

비밀번호

비밀번호를 입력해 주세요

비밀번호 확인

비밀번호를 입력해 주세요

이메일

@

이메일을 입력해 주세요

회원가입

취소

회원 가입 화면

댓글을 입력해 주세요

저장

Reply list

abcd123 수정 삭제

테스트 댓글

wigns61

안녕

댓글 작성 화면

- RestController와 Ajax를 사용하여 Json의 형태로 비동기 처리를 하였음
- 회원 가입 시 ID의 중복 검사는 페이지의 갱신 없이 부분적으로 데이터를 주고 받기 위하여 비동기 통신 처리
- 댓글의 CRUD, 카테고리(메뉴)의 CRUD 과정 또한 비동기 통신으로 처리

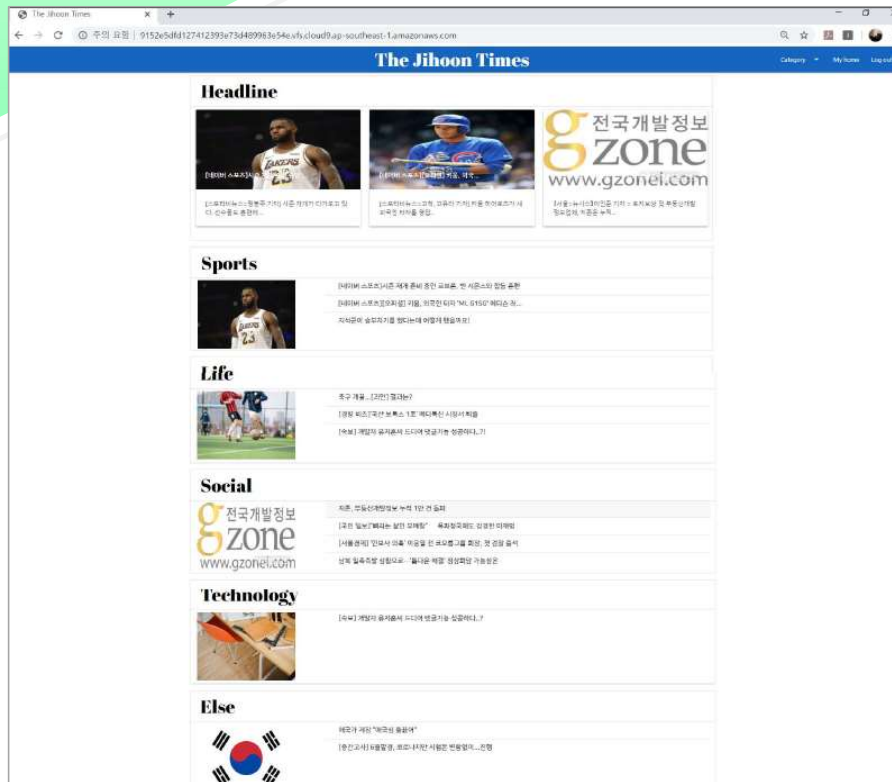


# 04

**Ruby On Rails로 인터넷 뉴스 포털 사이트 구현**

# 4-1

## Ruby On Rails로 인터넷 뉴스 포털 사이트 구현 프로젝트 개요



타임 라인

- 전공 과목인 소프트웨어 공학설계 개인 프로젝트
  - AWS에서 Linux기반의 C9 서버를 사용하여 구현.
  - 계정 생성 및 탈퇴, 로그인, 로그아웃 기능 구현.
  - 기사를 작성, 수정, 삭제 할 수 있는 기능 구현.
  - 로그인 여부와 상관없이 모든 사용자가 볼 수 있도록 메인 타임라인 및 카테고리 구현.
  - 댓글 및 좋아요 기능 구현.
  - 자신이 작성한 기사 리스트를 볼 수 있고, 계정의 정보 변경을 관리하는 페이지 구현.
- 
- 동영상 : <https://www.youtube.com/watch?v=J6ORj7hfuNU>
  - 소스코드 : <https://github.com/wlgns61/RubyOnRails>

# 4-2

## Ruby On Rails로 인터넷 뉴스 포털 사이트 구현 프로젝트 화면 캡처

The Jihoon Times

기사 작성하기

제목 입력

선택된 파일 없음

☐ Sports  
☐ Life  
☐ Social  
☐ Technology  
☒ Else

본문 입력

기사 작성

The Jihoon Times

My Articles

  
유지훈  
이메일 : nangam123@naver.com  
올린 글 개수 : 5

 동해물과 백두산이 마르고 닳도록 하나님이 보우하사 우리나라...  
2020-08-17 07:05:47 UTC

 동해물과 백두산이 마르고 닳도록 하나님이 보우하사 우리나라만세 우...  
2020-08-16 18:48:08 UTC

 지식이야 승부차기를 이겼다는데 어떻게 뒷물까지 나오도 ...  
2020-08-16 15:29:49 UTC

 어떨게 뒷물까요?  
2020-08-16 11:21:54 UTC

 [속보] 개발자 유지훈씨 드디어 댓글가능 성공하다..?  
2020-08-16 11:17:11 UTC

 개발자 유지훈씨가 돌아오와 댓글기능을 성공했다고합니다 자세한 내...  
2020-08-16 11:17:11 UTC

 남북 일촉즉발 상황으로... "틀려온 해결" 정상회담 가능성은...  
2020-08-16 11:17:11 UTC

 남북 일촉즉발 상황으로... "틀려온 해결" 정상회담 가능성은...  
2020-08-16 11:17:11 UTC

 [중간고사] 6월말경, 코로나지만 시험은 변함없이...전...  
2020-08-16 11:17:11 UTC

 [중간고사] 6월말경, 코로나지만 시험은 변함없이...전...  
2020-08-16 11:17:11 UTC

My home

The Jihoon Times

Sign up

Name

Email

Password (6 characters minimum)

Password confirmation

Avatar

선택된 파일 없음

[Log in](#)

회원 가입

- Devise gem을 사용하여 회원의 CRUD 및 로그인, 로그아웃, Session관리를 구현
- 처음 구현한 웹 프로그래밍으로 MVC모델의 model, view, controller 각각의 역할과 흐름을 이해

# 05

기타 사항

# 5-1

## 동아리 및 활동 이력

### AI Farm



- 2019.01 ~2019.12 딥러닝 기술을 다루는 동아리 활동을 하며 매주 <https://www.tensorflow.org/> 홈페이지의 tensorflow2.0을 사용하여 만들어진 가이드를 공부하고 발표하며 딥러닝의 기초를 다짐

### 과목 조교 활동

> 과목명	인공지능입문	> 구 분	조교
> 과목코드	SCS4033 -01	> 학기명	2019년 1학기
> 개설학과		> 이수구분	전선
> 학 점	3.0학점	> 수강생수	17명
> 진도율	0.0%	> 강의실 바로가기	<a href="#">이동</a>

> 과목명	머신러닝및딥러닝	> 구 분	조교
> 과목코드	SCS4037 -01	> 학기명	2019년 2학기
> 개설학과		> 이수구분	전선
> 학 점	3.0학점	> 수강생수	10명
> 진도율	0.0%	> 강의실 바로가기	<a href="#">이동</a>

- 2019년 1학기 인공지능 입문 수업보조 조교
- 2019년 2학기 머신러닝 및 딥러닝 실습 조교
- <Keras 창시자에게 배우는 딥러닝>[길벗, 2018]의 주피터 노트북으로 실습 진행

# 5-2

## 특허, 어학, 자격사항

### 특허



- 포트폴리오의 2번 목차의 YOLO v3를 사용한 비전기반 스마트 보행 시스템 프로젝트 특허 준비
- 2020년 12월 24일 출원
- 출원번호: 10-2020-0174002

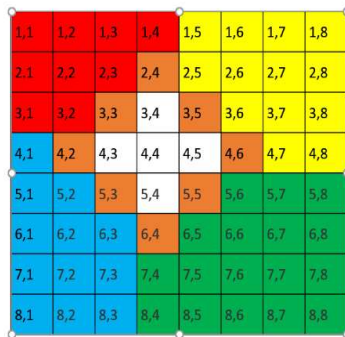
### 자격 및 어학 사항

- TOEIC : 715점
- 빅데이터 전문가 한국교육검정원 자격증 보유

# 5-3

## 알고리즘 문제 풀이 블로그 운영

구역 3: 파란색  
구역 4: 초록색  
구역 5: 갈색 테두리를 포함한 흰색  
 $n=8, x=2, y=4, d1=2, d2=2$  일때의 예시



(1) 구역 1

구역 1에 대한 범위는 문제에 주어졌듯이  $1 \leq i < x+d1, 1 \leq j \leq y$  이다. 하지만 이 범위에서 그림에서 빨간 부분만 표현하기 위해서는 갈색부분을 포함한 흰색 부분을 제외해야 한다. 이를 포함하지 않을 조건으로는 빨간색과 인접한 갈색부분을 보자. 그 갈색 부분의 합은  $x+y=6$ 으로 일정하다. 그러므로 모든 빨간부분은

$i+j < x+y=6$ 이다.

```
int area1(int x, int y, int d1, int d2) {  
    int sum = 0;  
    for (int i = 1; i < x + d1; i++)  
        for (int j = 1; j <= y; j++) {  
            if (i + j < x + y) {  
                visited[i][j] = 1;  
                sum += map[i][j];  
            }  
        }  
}
```

- 다양한 알고리즘 문제 풀이 및 코드 작성
- C++ 언어 사용
- 블로그 링크 : <https://countrysides.tistory.com/>