
졸업 작품 최종 보고서

- 어제 TV에 나온 그곳 -

강남대학교 컴퓨터 공학과	
학번	201301251
이름	김 지 운
제출일	2019.12.05

목 차

I. 개요

1. 개발 동기 및 목적
2. 프로젝트 디렉토리 구성
3. 기본 기능

II. 개발 환경

1. 개발 언어 / S·W
2. Server / DB

III. 프로젝트 구조

1. DB 구조
2. 웹 서버 구조
3. 애플리케이션 구조
4. 추천시스템 적용 구조

IV. 구현 결과

V. 제작 일정

VI. 보완사항 및 소감

1. 보완사항
2. 소감

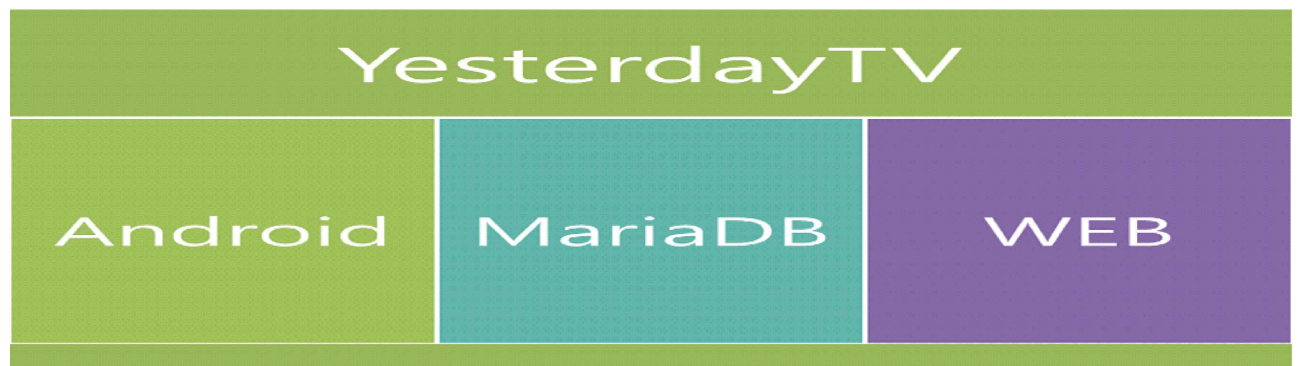
VII. 참고 사항

I. 개요

1. 개발 동기 및 목적

- 많은 현대인들이 TV 프로그램에 나오는 장소, 가게에 대한 관심이 많아서 인터넷에 검색하거나, 맛집 어플 등을 이용하여 원하는 정보를 얻거나 공유하는데, 대부분의 정보가 식당 즉 먹을 것에 관한 정보뿐이다.
- 따라서 TV 프로그램에 출연한 요식업을 제외한 다양한 업종의 사업장, 관광지 같은 촬영 장소 등의 정보를 사용자들에게 제공하는 어플리케이션을 만들고자 한다.

2. 프로젝트 디렉토리 구성



3. 기본 기능

- 정보 확인 : 해당 장소의 위치, 방영날짜, 전화번호, 설명 등의 정보를 보여준다.
- 목록 정렬 : 최근방영일, 사용자와의 거리, 인기순(즐거찾기 횟수)로 목록을 정렬해서 본다.
- 검색 : 이름, 주소, 키워드 3가지 방식으로 검색 가능
- 지도 검색 : 지도를 사용해 화면 내 장소들을 표시 및 정보를 본다.
- 추천 : 평점데이터를 활용하여 사용자들에게 장소를 추천한다.
- 즐겨찾기 : 즐겨찾기 등록을 통해 사용자들의 목록을 만들 수 있다.
- 리뷰 / 별점 : 해당 장소에 대해 리뷰와 평점을 남겨 사용자간 정보를 교환한다.
- 프로필설정 : 사용자가 이름, 생일, 성별, 프로필사진을 변경할 수 있게 한다.

II. 개발 환경

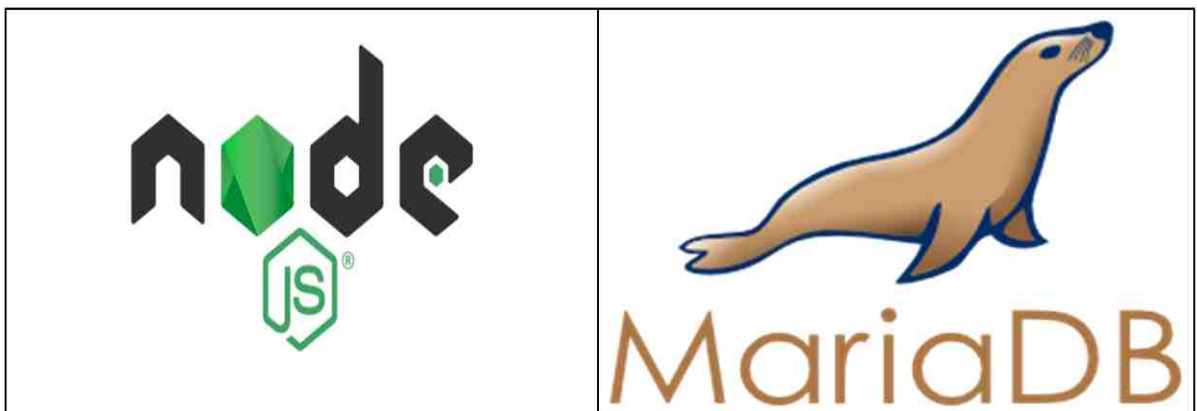
1. 개발 언어 / S·W

- Android Studio / JAVA / RScript(추천시스템)



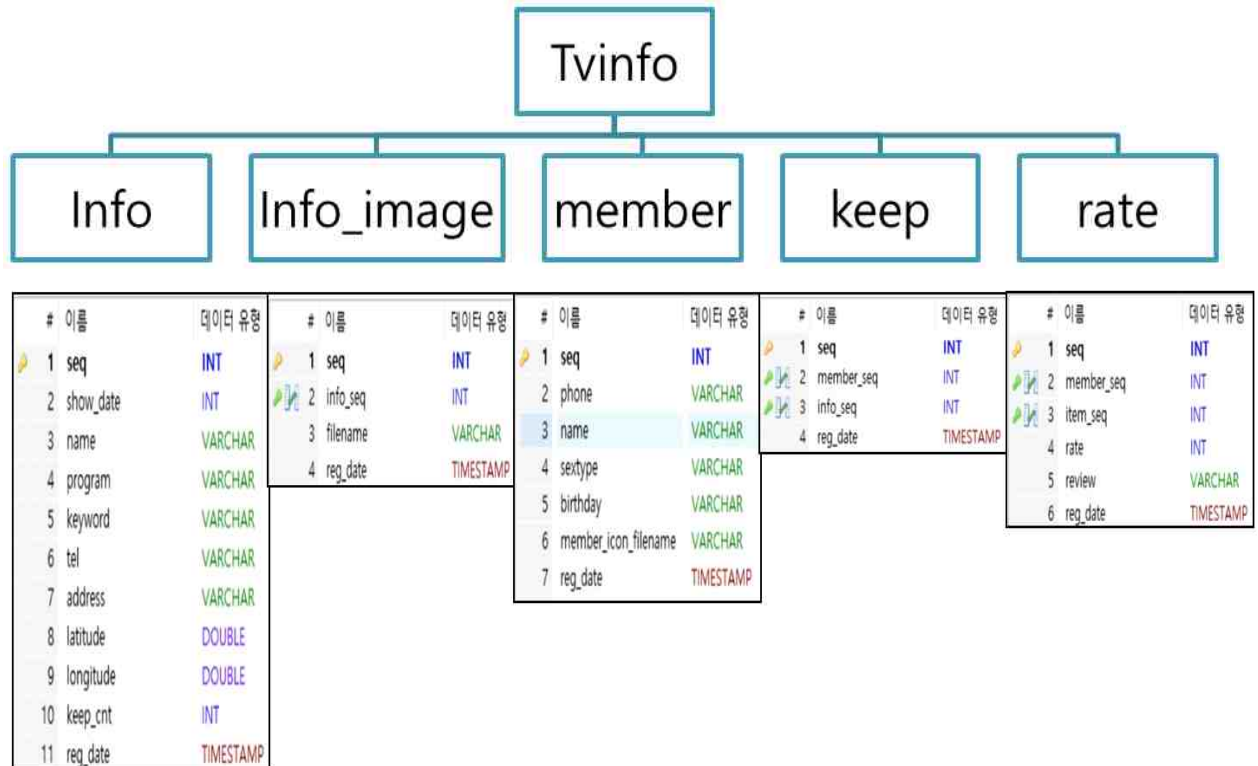
2. Server / DB

- nodejs(express framework) / MariaDB

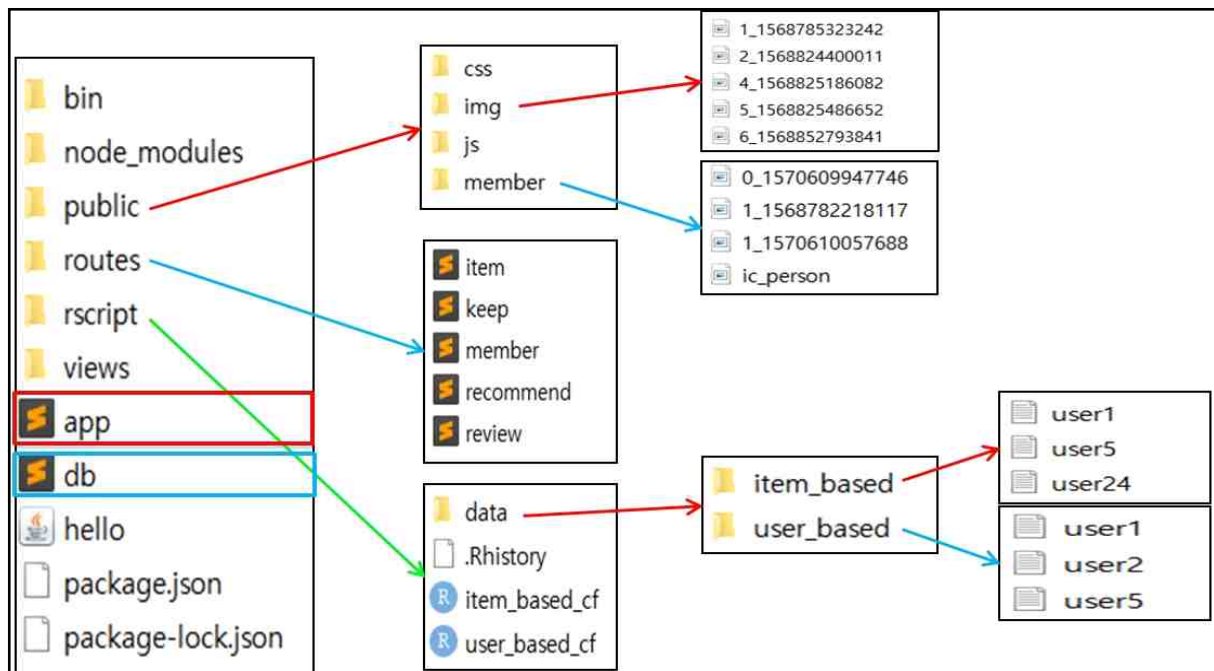


Ⅲ. 프로젝트 구조

1. DB 구조



2. 웹 서버 구조



* Express 프로젝트 생성 커맨드(cmd)

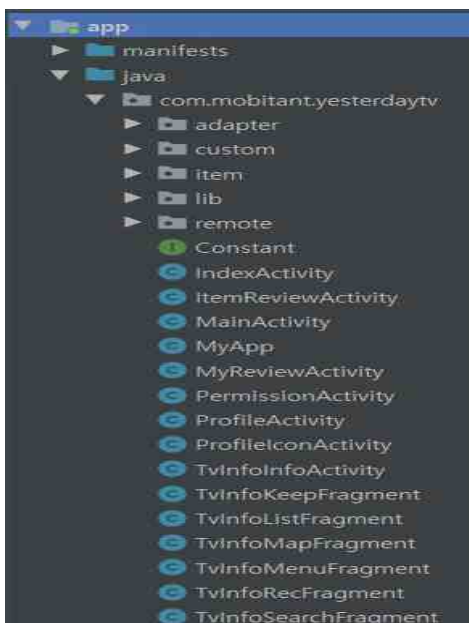
```
> npm install express-generator -g          /* express-generator 모듈 전역 설치 */
> express -e web                             /* 프로젝트 디렉토리 생성*/
> cd web
web> npm install                             /* 필요 패키지 자동 설치 */
web> npm install formidable mysql --save     /* 파일업로드, DB연결 모듈 설치 */
web> npm start                               /* 서버 실행 */
```

* 프로젝트와 연관된 디렉토리 및 파일 설명(자동으로 설치된 파일 제외)

- public 디렉토리 : 아이템(img)과 멤버(member)의 사진파일을 저장한다.
- routes 디렉토리 : 안드로이드에서 오는 경로에 따른 요청을 처리하는 파일들이 존재
- item.js : TV프로그램에 나온 장소에 대한 정보를 다룬다.
- member.js : 사용자 정보를 다룬다.
- keep.js : 즐겨찾기 동작에 대한 처리를 한다.
- recommend.js : Rscript를 연동하여 추천시스템을 다룬다.
- review.js : 리뷰와 별점 정보를 다룬다.
- rscript 디렉토리 : 추천시스템의 코드가 들어있으며, 각 사용자별로 생성된 추천아이템이 저장된 텍스트파일을 저장한다.
- item_based_cf.R : 아이템기반 협업필터링 추천시스템 코드
- user_based_cf.R : 사용자기반 협업필터링 추천시스템 코드
- app.js : 웹 애플리케이션을 위한 기본적인 설정을 가진 파일이다. 모듈을 로딩하고 템플릿 엔진을 설정하며, 라우트를 설정하는 코드를 가지고 있다.
- db.js : DB 커넥션 풀을 생성하고 DB를 사용하게 해준다.

3. 모바일 애플리케이션 구조

* 안드로이드 스튜디오 프로젝트 탭



* Activity

Permission / Index / Main / Profile / ProfileIcon
/ TvInfoInfo / ItemReview / MyReview

* Fragment

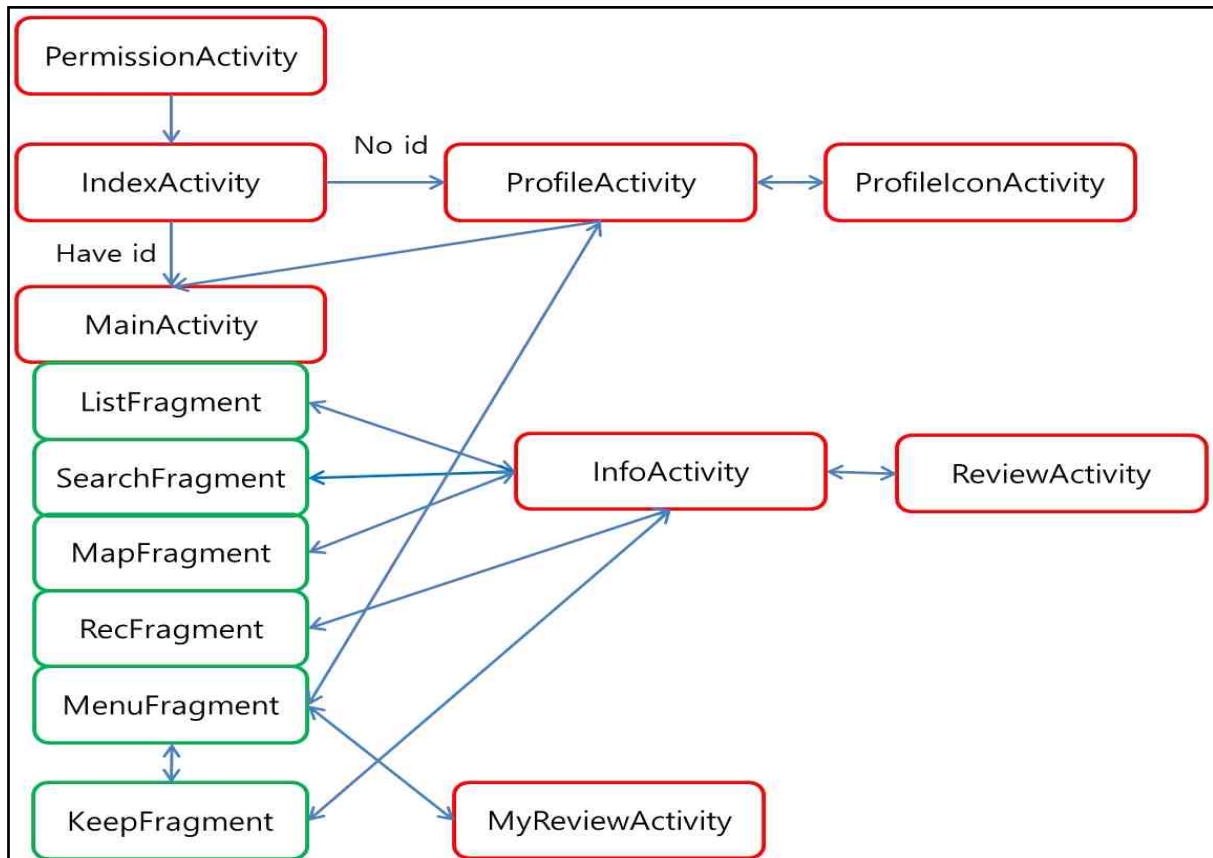
TvInfoList / TvInfoSearch / TvInfoMap
TvInfoRec / TvInfoMenu / TvInfoKeep

* Package

adapter : 리사이클러뷰를 사용하기 위한 어댑터
custom / lib : 자주쓰는 기능들을 정리
item : DB 테이블 또는 데이터를 표현
remote : 서버와 통신하기 위해 필요한 기능
-> HTTP 통신을 위해 Retrofit2 라이브러리 사용

* Constant / MyApp : 앱 전역에서 사용하는 상수, 클래스

* 서비스 흐름도



- node 서버와의 통신은 Retrofit2 라이브러리를 사용하여 통신한다.

EX) 프로필 설정에서 확인 버튼을 눌러 입력한 정보들을 서버로 보내는 방법

① RemoteService.java에 원격 서버로 호출할 경로와 메소드를 선언

```

@POST("/member/info")
Call<String> insertMemberInfo(@Body MemberInfoItem memberInfoItem);
    
```

② 각 액티비티, 프래그먼트에서 서버로부터 원하는 정보들을 삽입, 조회 할 수 있게 메소드 호출

```

Call<String> call = remoteService.insertMemberInfo(newItem);
MyLog.d(TAG, "text: here seq : " + newItem.seq);
call.enqueue(new Callback<String>() {
    
```

③ member.js의 '/info' 경로로 객체가 전달되어 코드가 실행된다.

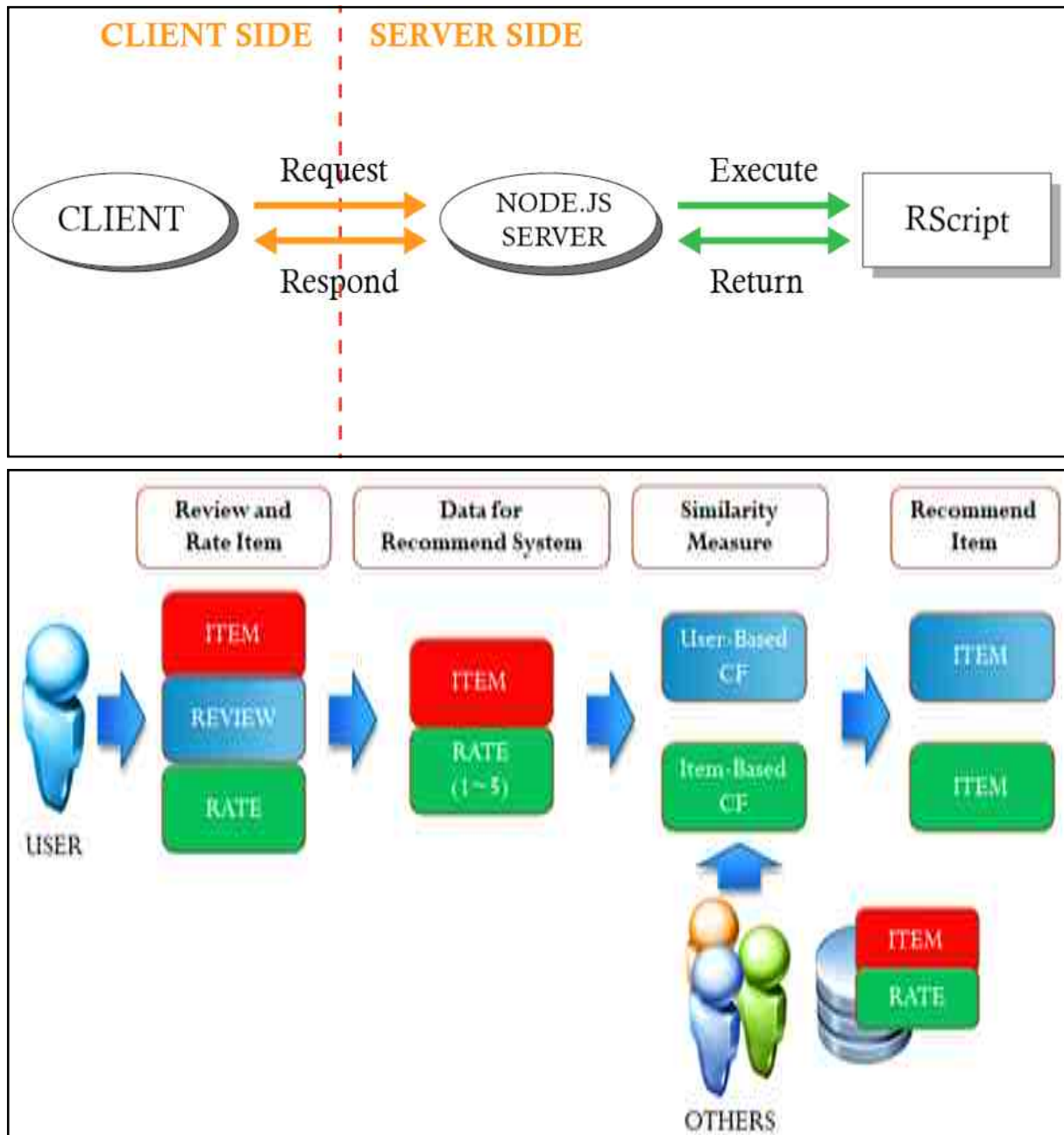
```

//member/info
router.post('/info', function(req, res) {
    var phone = req.body.phone;
    var name = req.body.name;
    var sextype = req.body.sectype;
    var birthday = req.body.birthday;

    console.log({name, sextype, birthday, phone});

    var sql_insert = "insert into tvinfo_member (phone, name, sextype, birthday) va
    var sql_update = "update tvinfo_member set name = ?, sextype = ?, birthday = ?
    var sql_select = "select seq from tvinfo_member where phone = ?; ";
    
```

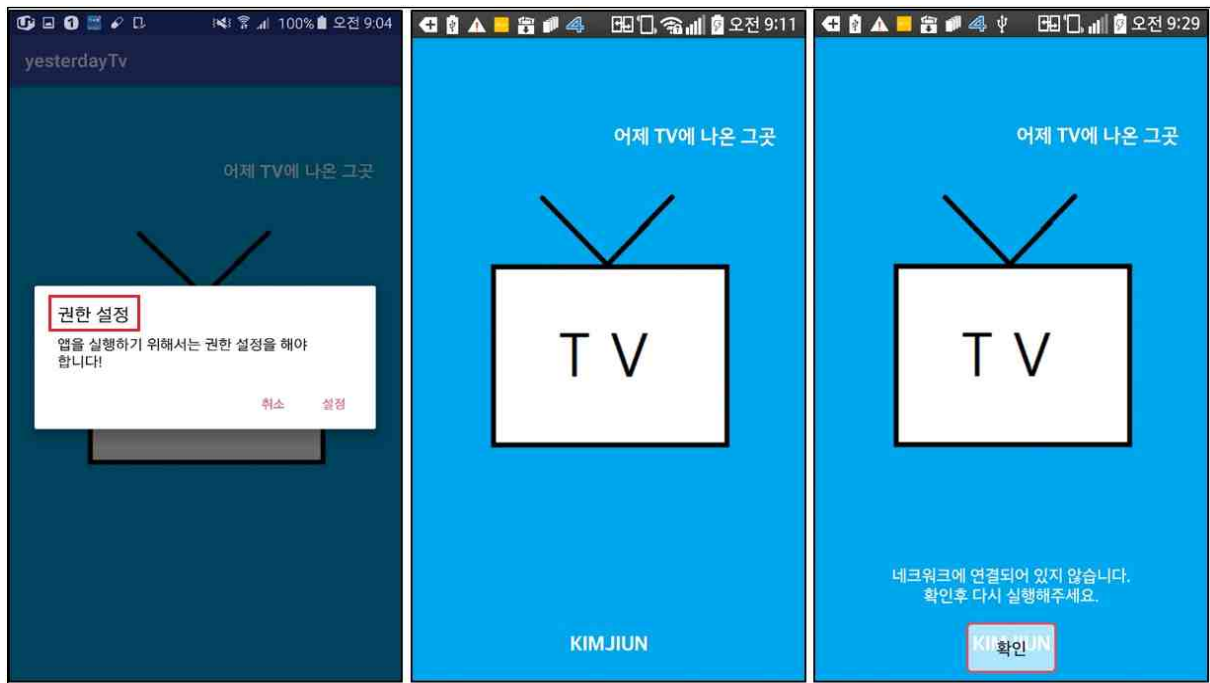

4. 추천시스템 적용 구조



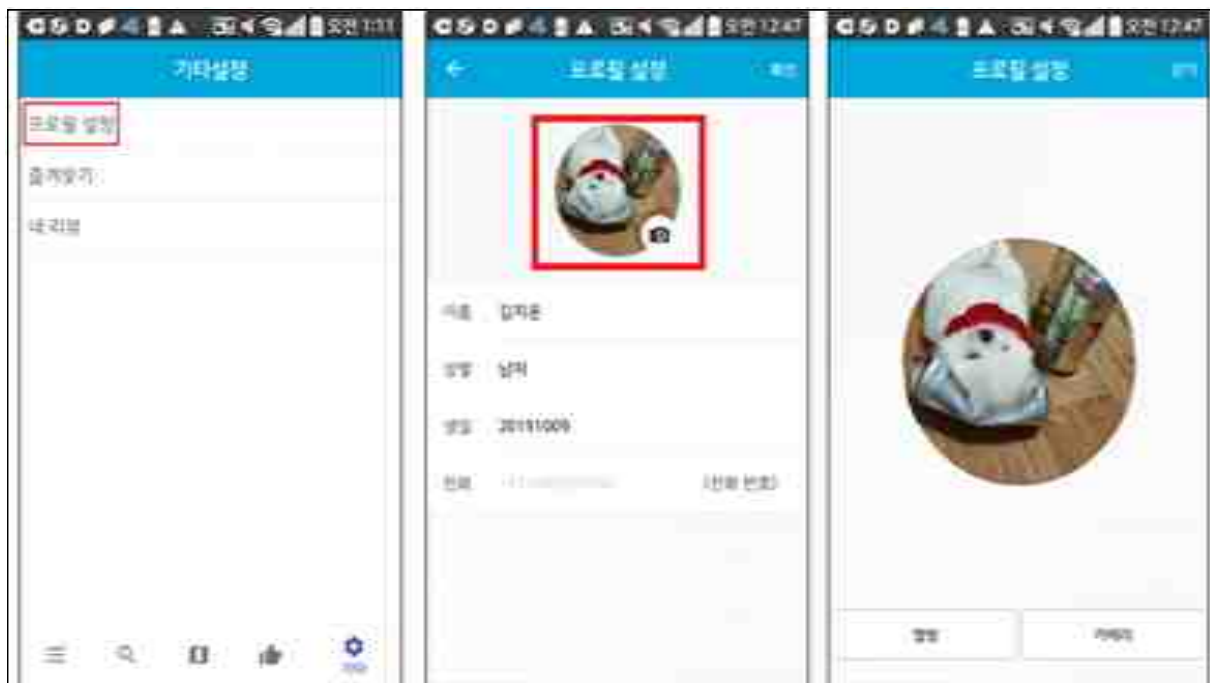
- 사용자가 서버로 추천아이템들을 요청을 하면 서버에서 사용자번호를 R-script로 전달한다. DB내의 평점데이터를 이용해서 추천모델을 생성한 다음 해당 사용자에게 추천모델을 적용한 유사도 상위 10개의 아이템을 추천한다.
- 아이템 기반 협업필터링 : 아이템간 유사도를 계산하여 사용자가 이전에 평가한 아이템들과 가장 유사한 아이템을 추천한다.
- 사용자 기반 협업필터링 : 사용자간 유사도를 계산하여 유사한 사용자가 가장 선호하는 아이템을 사용자에게 추천한다.

IV. 구현 결과

- 권한 설정 : 안드로이드 6.0부터 사용자로부터 권한을 승인 받아야한다.
- 시작 화면 : 서버에서 사용자 정보를 전화번호로 조회 한 다음 화면을 보여준다.
- > 사용자 정보가 존재하면 목록 탭이 열리고 / 없다면 프로필 설정 화면으로 이동한다.



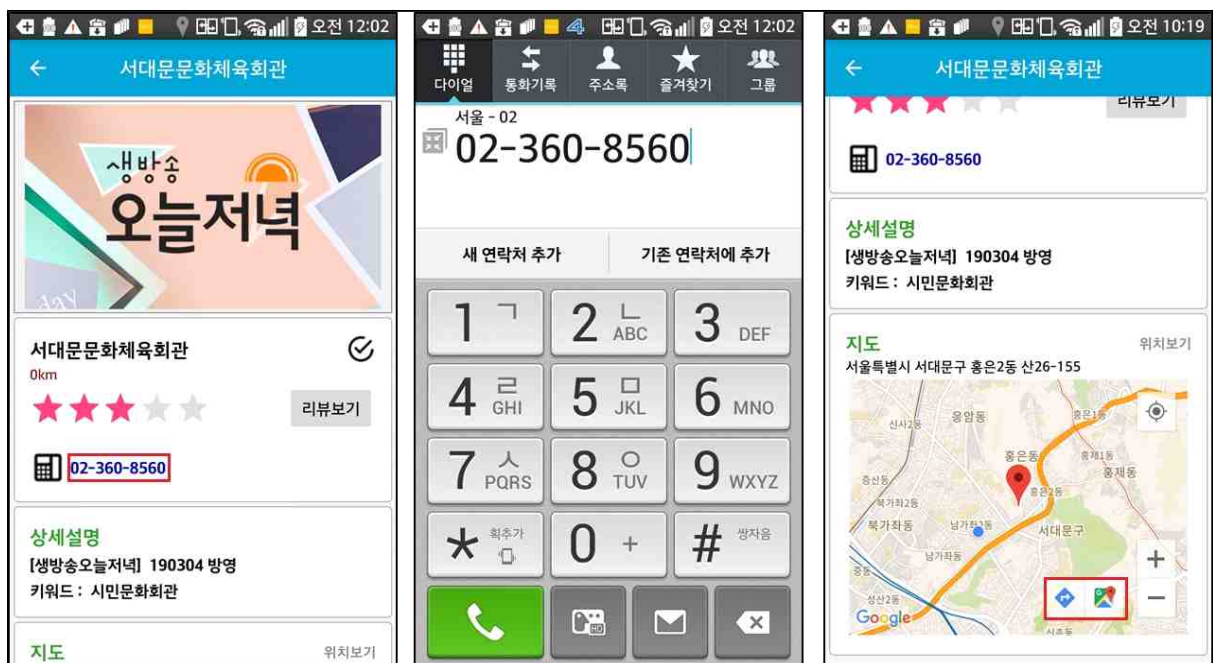
- 프로필 설정 : 사용자 정보를 입력 한다. 프로필 사진은 서버에 자동으로 업로드된다.



- 목록 : 거리순, 인기순, 최근순으로 정렬하여 목록을 볼 수 있고 아이টে를 누르면 해당 장소에 대한 상세한 정보가 나온다.



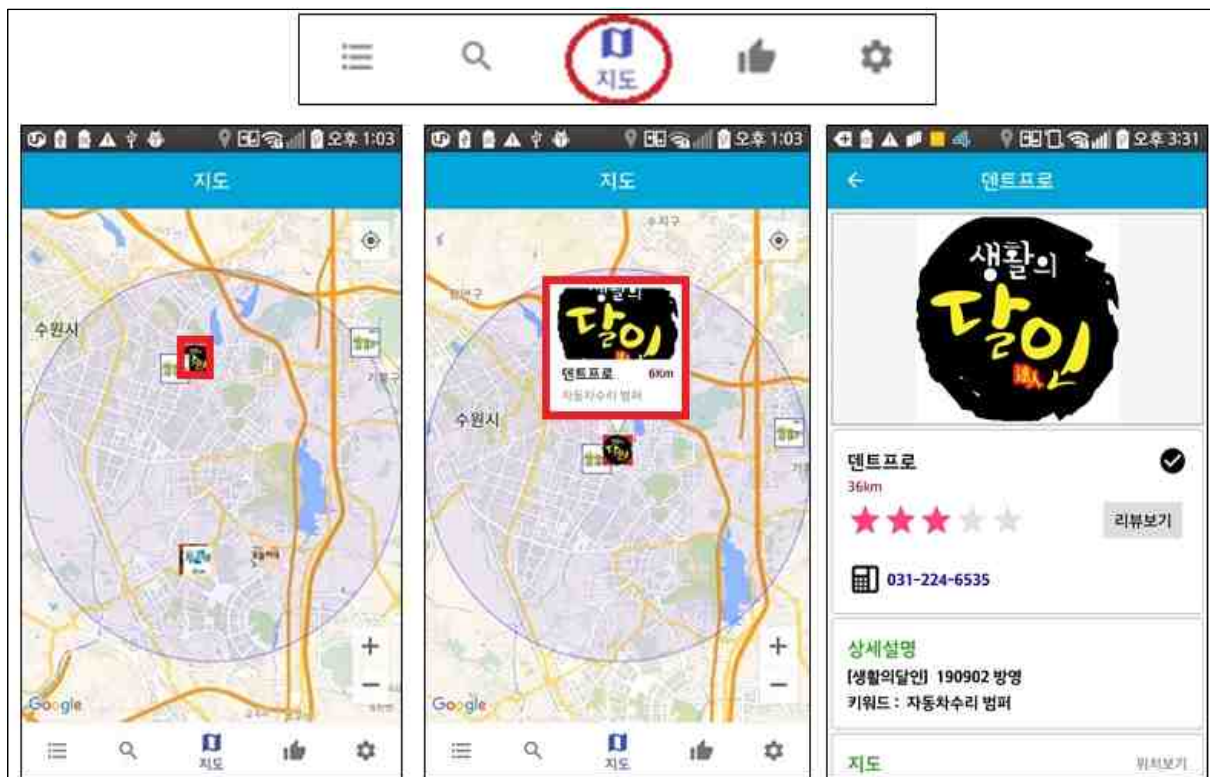
- 상세 정보창 : 해당 장소의 사진, 이름, 거리, 평점, 전화번호, 설명, 지도 등의 정보가 있다. 목록, 검색, 지도, 추천시스템, 즐겨찾기 아이템이 존재하는 모든 화면에서 누를 경우 해당 장소에 대한 상세 정보창이 열린다.



- 검색 : 이름, 주소, 키워드 3가지로 원하는 정보를 검색한다.



- 지도 : 지도상의 파란색 원 안에 TV프로그램에 나온 가게들이 표시된다. 마커를 누르면 간략한 정보가 나오며, 해당 창을 누르면 상세한 정보창이 나온다.



- The image displays a user interface for a recommendation system, titled '추천시스템' (Recommendation System). It features two main tabs: '아이템기반' (Item-based) and '사용자기반' (User-based). The '아이템기반' tab is active, showing a list of recommended items. The first item is '동기구 키제작' (Key Making) by '경기 용인시 처인구...' (Gyeonggi Province, Yongin City, Cheoin-gu...). The second item is '생방송 오늘저녁' (Live Broadcast Today Night) by '도원바이오테크' (Dowon Biotech). The '사용자기반' tab is also visible, showing a recommendation for '생방송 오늘저녁' (Live Broadcast Today Night) by '도원바이오테크' (Dowon Biotech). The interface includes a search bar at the top and a grid layout for the recommendations.

```
item_based Recommend for ID : 1
[ 81, 99, 10, 24, 7, 31, 26, 8, 112, 25 ]
sql : select a.*, (( 6371 * acos( cos( radians(?) ) * cos(
) ) * sin( radians( latitude ) ) ) ) * 1000) AS user_distance
.seq), 'true', 'false') as is_keep, (select filename from
re a.seq in (81,99,10,24,7,31,26,8,112,25) order by user_dis
params : 37.276364,127.1339639,37.276364,0,20
rows : [{"seq":31,"show_date":190218,"name": "농기쿠 키제작",
"원인구 양피면 수목리 152-8번지", "latitude":37.276364, "lon
```

```
user_based Recommend for ID : 1
[ 116, 12, 104, 36, 78, 97, 118, 49, 4, 56 ]
sql : select a.*, (( 6371 * acos( cos( radians(?) ) * cos( radians( la
)) * sin( radians( latitude ) ) ) ) * 1000) AS user_distance_meter, i
.seq), 'true', 'false') as is_keep, (select filename from twinfo_info
re a.seq in (116,12,104,36,78,97,118,49,4,56) order by user_distance_me
params : 37.276364,127.1339639,37.276364,,0,20
rows : [{"seq":4,"show_date":190920,"name": "구룡산 오토캠핑장 나이힐스",
"경기도 용인시 처인구 위성면 문성리 89-61", "latitude":37.117911,"longitude":127.1339639}]]
```

-

- 리뷰 / 나의 리뷰 : 장소에 대한 리뷰와 별점을 남길 수 있으며, 사용자들이 남긴 평점의 평균이 해당 장소의 평점으로 반영된다. 기타 설정 탭에서 내가 남긴 리뷰를 확인할 수 있다.



<평점 반영>

<나의 리뷰>

V. 제작 일정

일정	9월				10월					11월				12월	
	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2
DB 및 레이아웃 구성															
어플리케이션 제작															
추천시스템 연동															
졸업작품 발표															
작품 보완															
최종보고서 제출															

VI. 보완사항 및 소감

1. 보완사항

- * 데이터 수집 자동화 -> 파이썬의 BeautifulSoup를 이용하여 각 사이트를 크롤링 하였다.
- (SBS) 생활의달인, 생방송투데이 -> 모바일 페이지가 크롤링 되어 실패

① 텍스트가 들어간 div 태그의 class가 'aba inner' 인 것을 확인

[illegible]

② 해당 url의 html 문서를 불러온다.

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

dalin_html = urlopen("https://programs.sbs.co.kr/culture/lifemaster/board/55736/?cmd=view&page=1&board_no=5044")
print(dalin_html)

bsObject = BeautifulSoup(dalin_html, "html.parser")

print(bsObject)
```

③ pc 버전이 아니라 모바일 페이지가 크롤링 된다.

```
<!--<link rel="stylesheet" type="text/css" href="/css/channel-front-mobile-1.0.0.min.css" media="all" />-->
<!-- 게시판 모듈 -->
<link href="//board.sbs.co.kr/m/css/board-1.0.0.min.css" media="all" rel="stylesheet" type="text/css"/>
<script defer="" src="https://developers.kakao.com/sdk/js/kakao.min.js" type="text/javascript"></script> <!-- 카카오톡 관련 프로젝트 실행하기 위해 필수 -->
<script id="SBSBoardModule" src="//board.sbs.co.kr/m/js/sbs_board_module.js" type="text/javascript"></script>
<script src="https://www.gstatic.com/firebasejs/7.4.0/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.4.0/firebase-messaging.js"></script>
<script src="//programs.sbs.co.kr/front-mobile/js/config.js"></script>
<script src="//programs.sbs.co.kr/front-mobile/js/pushService.js"></script>
</link></head>
<body>
<div id="app-program-front-mobile"></div>
<script type="text/javascript">
  var googleTranslate = true;
  var googleTranslateLanguage = '';
  var googleTranslateUseful = true;
  var googleTranslateSelectedLanguage = '';
  var googleTranslateIncludeLanguageCodes = ['en','ja','zh-CN','th','vi','id','ru','es','pt','de','fr','ar'];
  //var googleTranslateIncludeLanguages = ['English','Japan','China','Thailand','Russia'];
  <div id="app-program-front-mobile"></div>
```

④ 해당 html 문서를 div 태그를 모두 불러온다. -> 내용 존재 X

```
In [85]: tagDiv = bsObject.findAll('div')
         print(tagDiv)

[<div id="app-program-front-mobile"></div>]
```


- (KBS) 2TV 생생정보 / (MBC) 생방송 오늘저녁 -> 자바스크립트로 내용이 들어가 있어서 크롤링 했을 때 html 문서에 내용이 보이지 않아서 실패.



* 추천시스템 정확도 향상

- 추천시스템을 적용하기 위해 200명의 사용자가 120개의 아이템중에 무작위의 아이템에 무작위의 평점을 부여한 약 15000개의 데이터를 만들었다.
- 하지만 무작위로 아이템을 선정해서 평점과 아이템에 상관성이 없어서 추천시스템을 동작 시켰을 때마다 다른 아이템이 추천되는 등의 문제가 발생하였고 평가 점수도 낮게 나왔다.
- 따라서 기존의 사람들이 추천시스템을 적용하기 위해 만든 영화 평점 데이터를 사용자 번호(1~200)와 아이템 번호(1~120)로 범위로 매핑하여 추천시스템을 동작 시켰는데 처음 과 같이 정확한 추천을 하지 못하였다.
- 다음으로 사용자 번호(1~200)와 아이템 번호(1~120)보다 큰 번호를 가진 데이터를 삭제 후, 추천시스템을 동작 시켰는데 정확도가 약간 향상 되었다.

2. 소감

졸업작품을 처음 시작할 때 단순한 맛집앱 비슷한 것을 만들려고 했었다. 하지만 교수님께서 조언을 해주셔서 추천시스템이라는 것이 어떤 것인지 알게 되었고, 내가 만든 앱에 추천 시스템을 적용할 때 가장 갑갑했던 점이 신빙성 있는 데이터를 수집하는 일이었다. 차라리 아이템을 영화로 바꿔서 진행했으면 하는 후회도 했지만, 처음 구상했던 앱을 만들고 싶다는 고집 때문에 정확한 추천시스템을 적용하는 것은 실패하고 말았다.

또 한 가지 아쉬운 점은 추천시스템을 자바 기반의 아파치 머하웃을 이용해서 구현하고 싶었는데 이미 서버를 nodejs로 구현해서 자바프로젝트와 데이터를 주고받는 것에 실패해서 R을 이용해서 만든 것이다.

하지만 1년 동안 졸업작품을 제작하면서 주변의 도움과 특히 교수님과의 커뮤니케이션을 통해 모르는 부분을 채워나가는 점이 좋았다.

VII. 참고 사항

1. 참고 서적

* 추천시스템 관련

- Suresh K.Gorakala, 추천 엔진을 구축하기 위한 기본서, 에이콘, 2017
- Sean Owen, 머하웃 완벽 가이드, 한빛미디어, 2012
- Suresh K.Gorakala, R로 만드는 추천시스템, 에이콘, 2017

* 안드로이드 관련

- 우재남, 박길식, Android Studio를 활용한 안드로이드 프로그래밍, 한빛아카데미, 2017
- 한동호, Node.js와 오픈소스를 활용한 안드로이드 서비스 개발, Jpub, 2017

2. 참고 웹사이트

* 추천시스템 관련

- <https://rstatistics.tistory.com/31>
- <https://kkokkilkon.tistory.com/102>
- <https://cinema4dr12.tistory.com/980>
- <https://stackoverflow.com/questions/50358346/run-r-script-from-nodejs-and-express>
- <https://github.com/joshkatz/r-script>

* 안드로이드 관련

- https://github.com/kairo96/and_node

* 웹 크롤링 관련

- <https://webnautes.tistory.com/779>
- https://www.fun-coding.org/crawl_basic2.html
- <https://yoonpunk.tistory.com/4>