**모바일 프로그래밍**

**[INSTEAGRAM]**



201611185 권혜지

201611193 김여리

201611244 한상규

201611245 홍혜진

**제출일: 2018년 6월 14일**

**Ⅰ. 서론**

1. **개발 배경**

대학가를 포함해 길거리를 걷다 보면 다양한 커피 전문점이 들어서 있는 것을 확인할 수 있다. 관세청이 발표한 자료에 따르면 성인 1명이 1년간 312잔의 커피를 마신다고 한다. 평균적으로 하루에 1잔의 커피를 마신다는 것이다. 이렇게 커피가 생활 깊숙이 자리매김 하면서, 소비자들은 다양한 카페를 원하게 되었다. 이런 소비자의 요구를 반영해, 가장 빠르게 트렌드를 반영하는 SNS인 인스타그램에서 내가 원하는 컨셉의 카페 정보를 제공하는 어플리케이션을 개발하게 되었다.

1. **개발 목표**

최종 목표: 인스타그램의 해시태그를 분석해 사용자의 취향에 맞는 게시글을 보여주고, 게시글에서 언급한 카페의 정보를 제공한다.

세부 목표:

1. 회원가입 및 페이스북 로그인

사용자는 아이디, 비밀번호를 설정해 회원가입을 하거나 페이스북 계정으로 로그인할 수 있다.

1. 관심 분야 설정

분위기, 이색 카페, 가격 등 사용자가 관심있는 키워드를 직접 선택해 관심 분야를 설정할 수 있다.

1. 선택 장소에 따른 게시글 노출

사용자가 지도에서 선택한 장소가 반영된 검색 결과를 보여준다.

1. 실시간 해시태그 분석 및 게시글 노출

인스타그램에 업로드 되는 해시태그들 중 특정 해시태그를 필터링해 사용자의 관심 분야와 높은 관련성을 갖는 게시글 들을 노출한다.

1. 카페 정보

사용자가 게시글을 클릭하면 게시글 정보에 포함된 카페의 위치정보를 보여준다.

1. 별점, 후기 작성

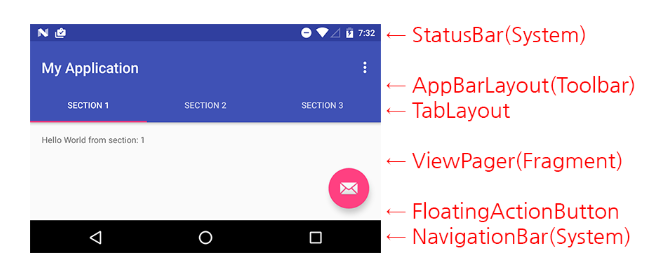
해당 카페에 별점과 후기를 남길 수 있다.

**Ⅱ. 개발 시스템 내용**

**◈Firebase Database 사용 현황**

|  |
| --- |
| User 클래스를 사용해 사용자 데이터 관리 |
| ReviewData 클래스를 사용해 카페 별로 리뷰 데이터 관리 |

- activity\_main2의 ViewPager내에 fragment\_main을 구성하여 프래그먼트에 메인 화면을 보여줬다. ViewPager에 SectionsPagerAdapter를 적용하여 탭의 수, 탭의 title등을 설정하였고, FragmentPagerAdapter의 onCreateView에서 상세 화면을 구성하였다.

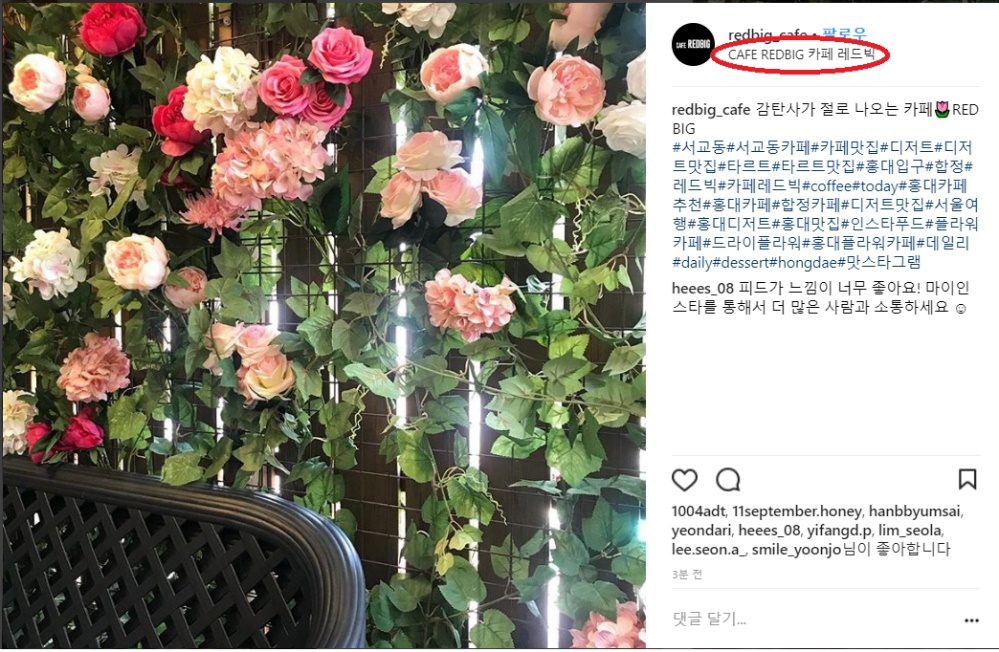


**◈ 화면 별 설명(UI)**

|  |  |
| --- | --- |
|  | <앱 첫 실행 화면>  일정시간 보여진 후 사라진다. |
|  | <시작 화면>  START버튼 클릭 시 로그인 화면으로 넘어간다. |
|  | <로그인 화면>  사용자는 앱 자체의 ID, PW로 로그인하거나 페이스북 계정으로 로그인할 수 있다.  페이스북으로 로그인 시 아이디가 데이터베이스에 저장되고, 사용자는 관심사를 따로 선택할 수 있도록 하였다. |
|  | <페이스북 로그인화면>  페이스북 로그인 버튼을 누르게 되면 다음과 같은 화면으로 이동하여 계속을 누르면 페이스북으로 로그인이 된다. |
|  | <회원가입 화면>  아이디, 비밀번호 입력 후 사용자별 관심사 카페를 1개에서 3개까지 선택할 수 있다. 중복확인 버튼 클릭 시 Iterator를 사용하여 DB에 이미 저장된 아이디인지 검사한다. 입력한 정보는 DB에 “id”, “preferences”, “pw”의 이름으로 저장된다.  후에 메인 화면에서 사용자가 선택한 카페에 따라 게시글을 분류하여 보여준다. |
|  | <메인 화면>  회원가입 시 선택한 관심사에 따라 탭을 생성하였다. 집 앞 카페는 항상 존재하는 탭으로, 사용자의 현재 위치를 가져와서 내 주변의 카페를 보여준다.  화면에는 실제 인스타그램 게시물과 게시물에 포함된 카페의 위치를 보여준다. |
|  | <관심사 변경>  메인화면FloatingActionButton 클릭 시 들어오는 화면이다. 관심사를 변경하면 변경된 결과를 메인 화면에 반영하여 탭을 동적으로 생성한다. |
|  | <관심사 변경 시 화면>  관심사 변경 시 화면이다. 탭이 관심사 수에 따라 동적으로 변화하였다. |
|  | <카페 목록 화면>  선택한 분야의 카페에 해당하는 정보를 인스타그램에서 가지고 온다. 카페 이름과 함께 사진을 출력하고, 클릭하면 카페 상세 정보 페이지로 넘어간다. |
|  | <카페 상세 페이지>  인스타그램에 게시된 사진과 카페 이름, 주소 정보, 메뉴 정보(제공되는 경우)를 출력한다. 지도에 해당 위치의 마커를 찍어 보여주고, 회원들이 작성한 리뷰와 평균 평점을 확인할 수 있다. 말풍선 버튼을 누르면 리뷰 작성 페이지로 넘어간다. |
|  | <리뷰 작성 화면>  RatingBar를 이용해 별점과 함께 리뷰를 남길 수 있다. 남긴 리뷰는 ReviewData 형태로 Firebase Database에 저장되며, 즉시 해당 카페의 리뷰 정보에 반영된다. |

**◈ 해시태그 속에서 장소 검출하기**

처음 인스타그램 홈페이지에서 파싱을 시도할 때,

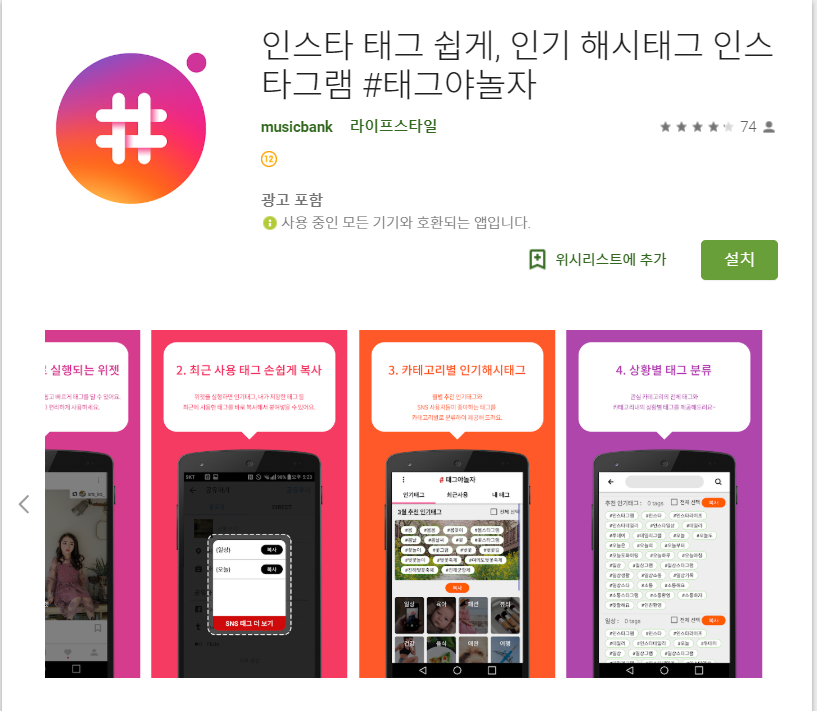


빨간 동그라미로 표시한 ‘장소’를 가져올 수 있을 것이라 생각했다. 그래서 좀 더 정확한정보들을 많이 가져올 수 있을 것이라 생각했다. 그러나, 홈페이지 소스 어디에서도 ‘장소’는 표시되지 않았다. 그리하여 해시태그에서 장소를 뽑아내는 것이 더욱더 중요해졌다. 개발을 위해 여러 해시태그를 살펴보았을 때, 주로 패턴이 있다는 것을 파악할 수 있었다.



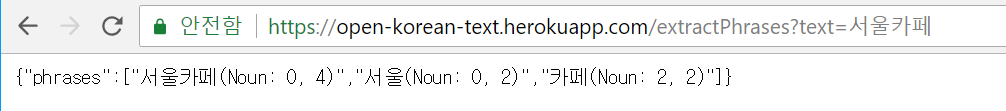
먼저, 위의 사진을 보면

#서교동#서교동카페#카페맛집#디저트#디저트맛집#타르트#타르트맛집#홍대입구#합정#레드빅#카페레드빅#coffee#today#홍대카페추천#홍대카페#합정카페#디저트맛집#서울여행#홍대디저트#홍대맛집#인스타푸드#플라워카페#드라이플라워#홍대플라워카페#데일리#daily#dessert#hongdae#맛스타그램

라는 해시태그가 쓰여져있다. 그러나 이런 해시태그를 보았을 때, 여행, 데일리, daily, dessert, 맛스타그램 등 일정 해시태그들이 다른 해시태그에서도 반복되고 있다는 것을 깨달았다. 그래서 이를 조사해보니 주제별로 자주 쓰고 인기있는 해시태그가 반복되고 있다는 것을 알게 되었다. 실제 인스타그램을 사용하는 많은 유저들은 해시태그 작성에 어려움을 겪고 있어 자주 쓰이는 해시태그를 복사 붙여넣기 하거나, 따로 해시태그를 작성해주는 앱이 있다는 것을 확인했다. 그리하여 미리 주로 사용되는 해시태그를 조사하여 1차 필터링 작업을 거쳤다. 

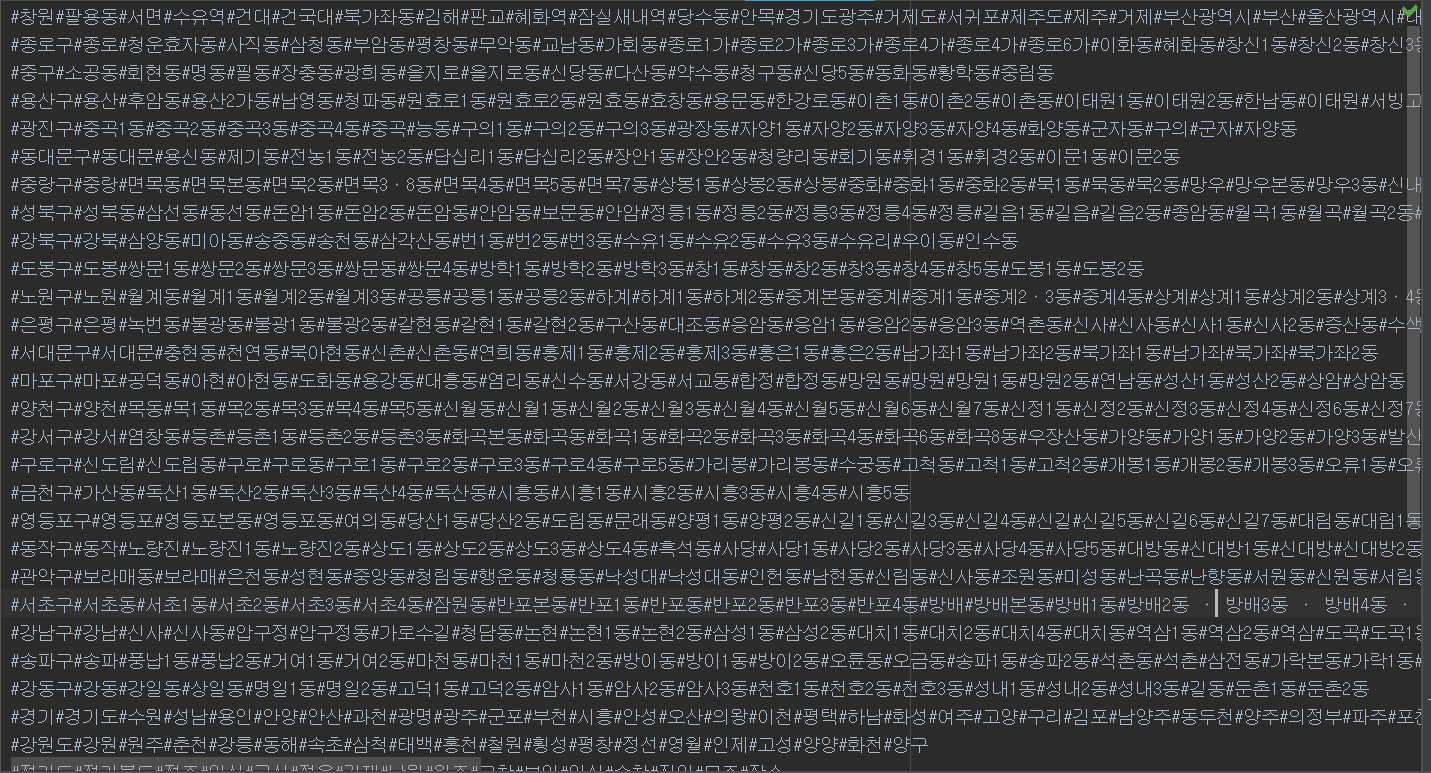


위의 정보들은 주로 사용하는 해시태그를 미리 입력한 것이다. 이를 통해서 불필요한 정보를 미리 거를 수 있다.

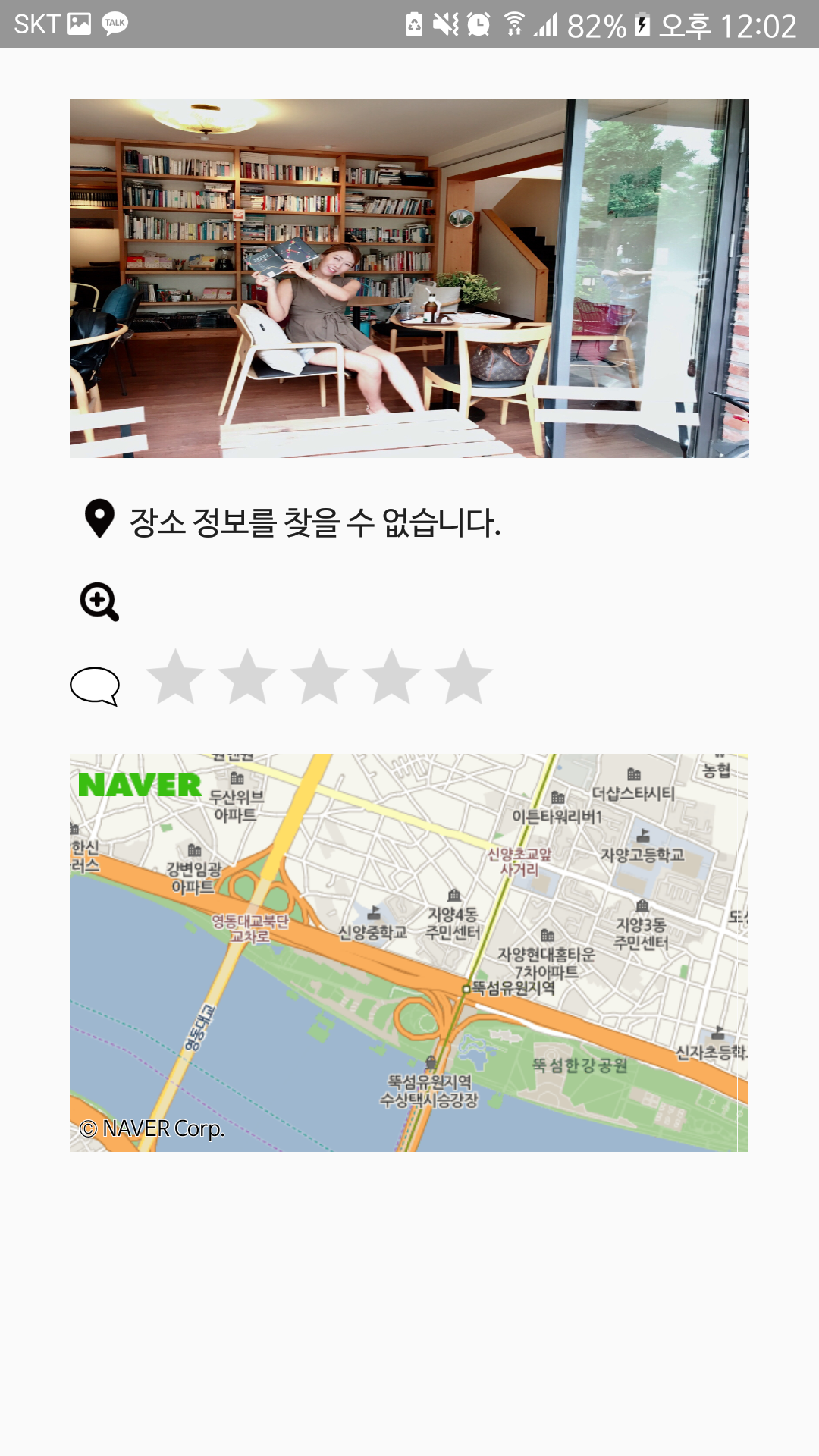
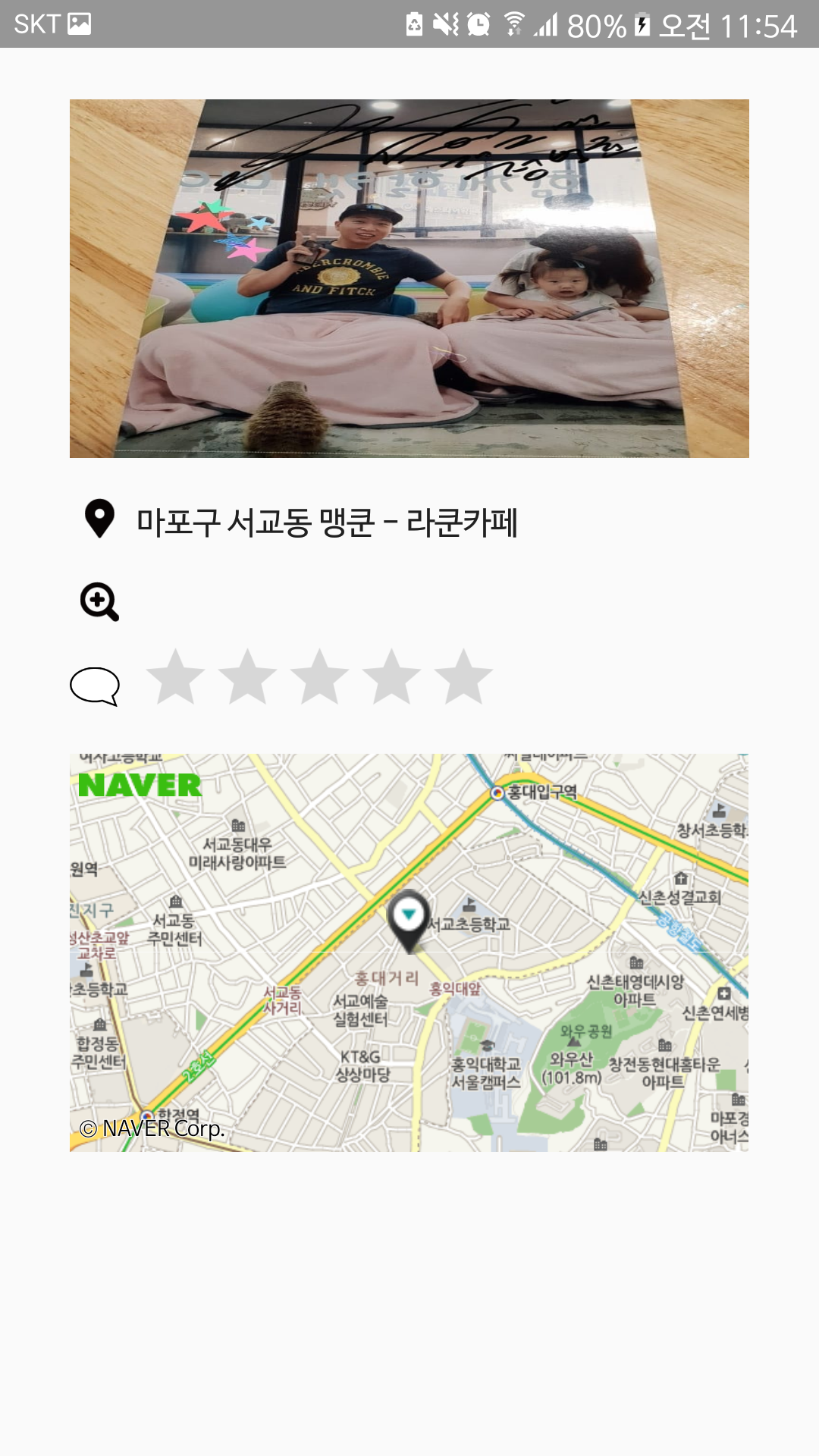
두 번째는, 어근화를 통한 장소추출이다. 해시태그에는 #서울카페 라는 것이 존재하는데 이때, 우리가 필요한 정보는 서울 이다. 이렇게 되면 서울과 카페를 분리할 필요성을 느끼는데 이는 open-korean-text라는 opensource를 사용하여 단어를 어근화 처리하였다.   


다음과 같이 서울카페, 서울, 카페로 분리되는 것을 볼 수 있다. 이는 Json형식이어서 JsonParshing을 통해 데이터를 가져왔다. 미리 입력해둔 장소 정보와 비교 후 장소라고 판단 될 시에는 장소 정보를 가지고 있는다. 그러나, 장소 정보가 없을 경우에는 태그만으로 검색 해야한다. 3차필터링이 필요한 것이다.

세 번째는 를 이용한 장소 자동완성 기능을 이용하였다. 본래 자동완성 기능은 구글에서 제공하는 뷰에서 띄우지만, 우리는 보여지는 뷰에서 사용자가 선택하는 것이 아니라 리스트 중에서 필요한 정보를 가져오는 것이기 때문에 ‘프로그래밍 방식으로 장소 예상 검색어 가져오기’를 이용했다. query에는 태그 하나하나를 넣어 for문을 돌고( 서교동 서교동카페 카페맛집 디저트 합정 레드빅 카페레드빅 ) 특정 영역으로 결과를 편중하는데 대한민국에서 주어진 카페를 가져와야 하기 때문에 LatLngBounds는 대한민국으로 제한하였다. 그리고 AutocompleteFilter에서 TYPE\_CAFÉ로 결과를 편중하여 CAFE값을 최우선으로 필터링하게 하였다. 그러나 문제가 되었던 부분은 ‘비동기’문제였다. 그래서 PendingResult에서 값을 바로 받아오는 것이 아니라 비동기로 값을 가져와서 await로 기다리게 되면 계속 앱이 종료되었다. 결국에는 핸들러로 처리를 했다.



위의 정보들은 지역정보들을 모두 담은 정보이다. 역정보(서울, 부산, 인천)와 지역정보, 동정보까지 모두 입력해서 더욱더 검색 결과를 높였다.



만약 장소 검색 결과가 있다면 왼쪽과 같이 GPS정보와 장소에 대한 상세정보가 나오고 장소검색 결과가 없다면 오른쪽 화면과 같이 뜬다. 해시태그에 장소에 대한 정보가 들어

이로써 인스타그램에서 해시태그를 3번의 필터링을 통해 장소와 유사하게 나오는 것을 볼 수 있다. 사용자가 좀 더 장소 태그를 제시해주면 조금 더 정확한 검색 결과를 볼 수 있을 것이라 기대해본다.

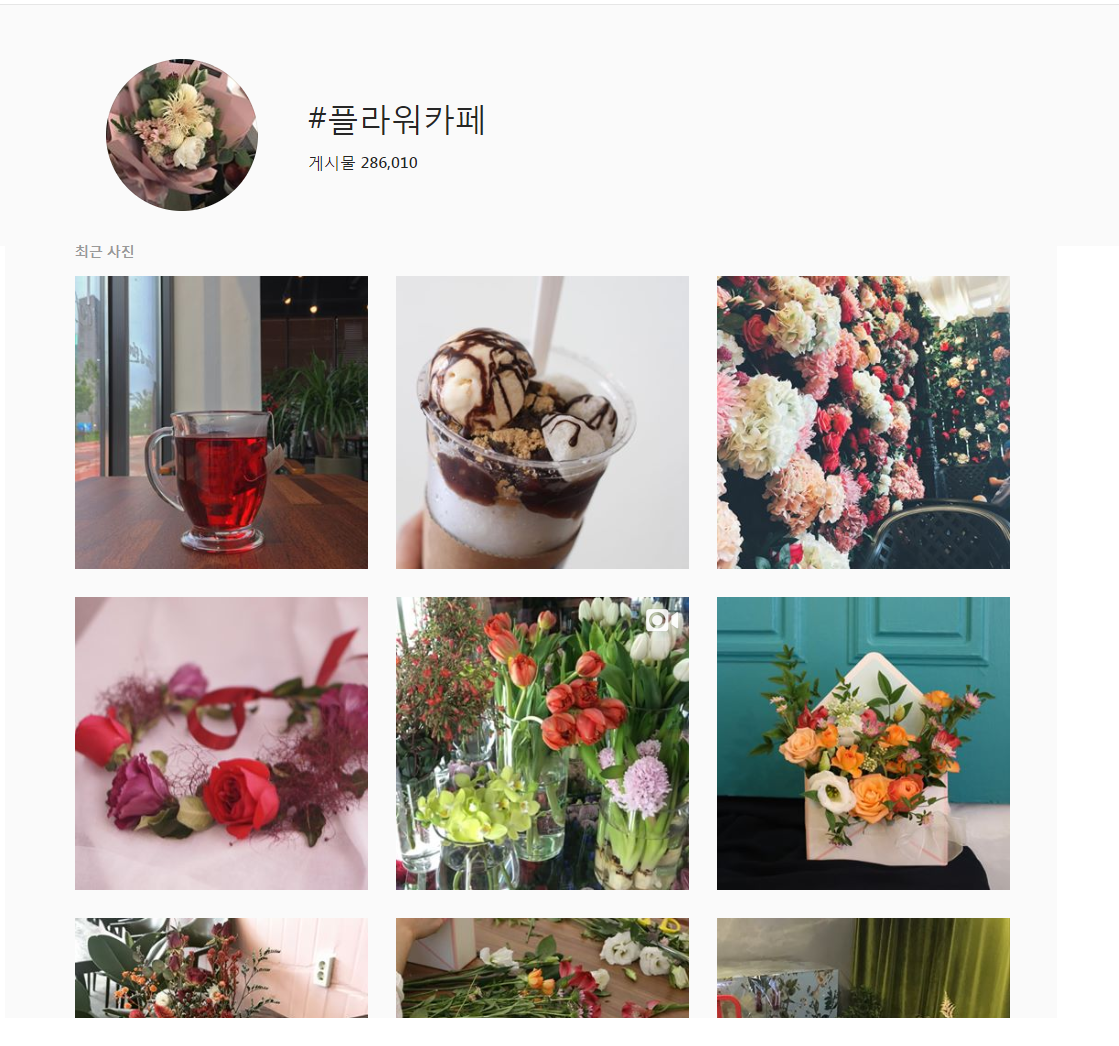
**◈ 태그를 기반으로 한 인스타그램 최신글 파싱(이미지, 게시글코드, 태그)**

인스타그램 API를 권한을 얻은 어플리케이션만 사용할 수 있다는 사실을 알았다. 그래서 웹파싱을 통해 직접 인스타 게시글의 이미지와 태그들을 가져오기로 하였다.

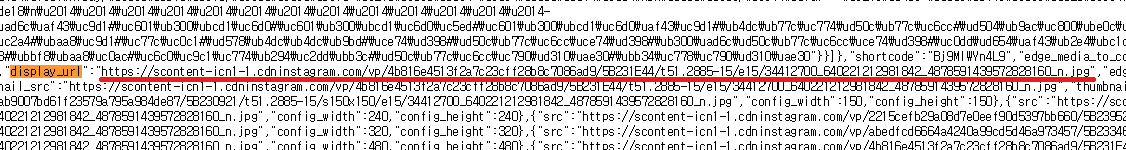
파싱을 하기 위해서는 URL이 필요하다.

<https://www.instagram.com/explore/tags/>**플라워카페**/?hl=ko

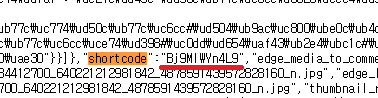
인스타그램은 위와 같은 형식의 URL을 사용한다. 빨간색 굵은 글씨로 적힌 부분에 내가 원하는 태그 문자열을 입력하면 해당 태그를 포함하고 있는 인스타 게시글을 볼 수 있다.



인스타 웹파싱을 위해 Ion 라이브러리를 사용했다.



**페이지 소스보기**를 통해 이미지 display\_url이 해당 게시글의 이미지라는 것을 알았다.



게시글의 태그를 가져오기 위해서는 게시글 url로 이동해야 한다. Shortcode는 게시글의 아이디이다. Shortcode를 이용해서 게시글 하나 하나에 접근할 수 있다.

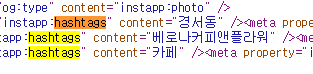
[https://www.instagram.com/p/**Bj9MlCpgB4E**/?hl=ko&tagged](https://www.instagram.com/p/Bj9MlCpgB4E/?hl=ko&tagged)=**플라워카페**

게시글의 URL은 위와 같다. Shortcode 부분을 바꿔주면서 게시글에 접근하면 된다.

Display\_url과 shortcode를 ArrayListAdapter에 넘겨주었다. 태그들은 getView에서 받는다.



게시글 URL로 게시들에 접근했다면 태그를 가져올 차례이다. getView를 생성할 때마다 Ion를 호출하여 해시태그들을 가져온다.



해시태그들은 hashtags라는 곳에 있었다. for문과 indexOf()를 이용하여

hashtags” content=” **~ ~ ~** “

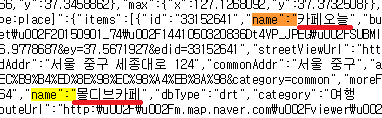
“와 “ 사이에 있는 값을 가져왔다.

**◈ 내 주변 카페 가져오기(카페 이미지, 메뉴, 영업시간, 주소)**

https://m.store.naver.com/places/listMap?display=40&level=top&nlu=%5Bobject%20Object%5D&query=카페&sortingOrder=distance&viewType=place&viewType=place&x=" **x**"&y=" **y**”

네이버 플레이스 거리순 정렬 URL이다.

x, y에 위도, 경도 정보를 넣어주면 근처에 있는 카페를 가져온다. 위도, 경도는 GPS를 이용했다.



Name”:”에 저장된 카페 이름을 가져온다.

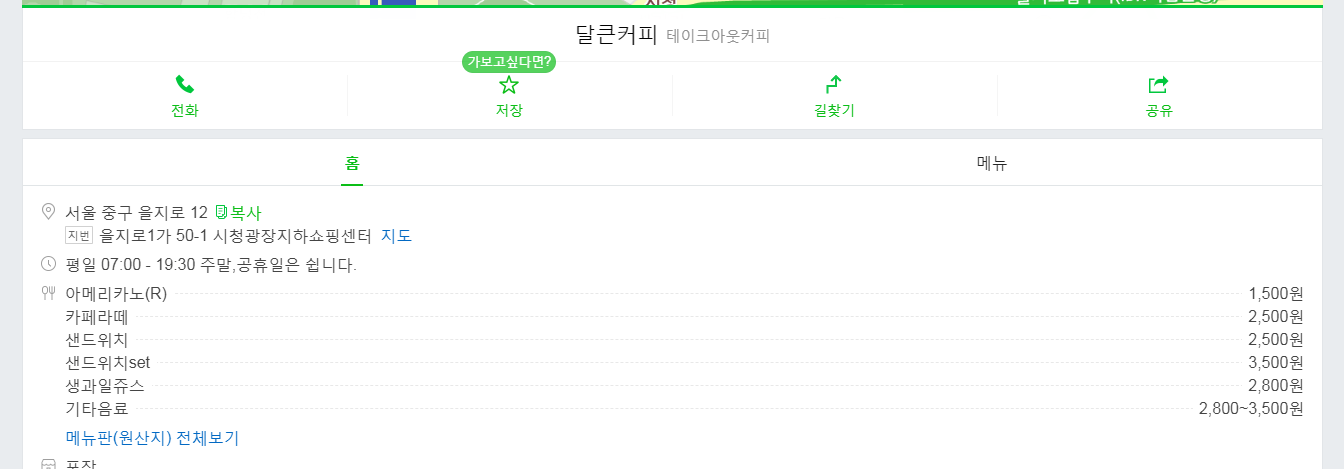


Id”;”에 저장된 값으을 이용하면 해당카페에 대한 상세페이지로 이동할 수 있다.

이 후 name에 있던 카페이름을 인스타 태그검색을 이용하여 해당 카페의 이미지를 가져온다.

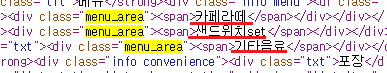


메뉴와 영업시간 주소 정보는 getView에서 처리해준다.

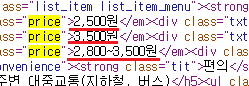




hourString에 영업시간이 포함돼있다



Menu\_area에 음식이름이 저장돼있다.



Price에 음식가격이 저장돼있다.

영업시간처럼 파싱해올 데이터가 한 개이면 for문을 한번 돌지만 메뉴정보처럼 파싱해올 데이터가 여러 개인 경우 이중 for문을 이용한다.

IndexOf를 사용하면 그 문자열이 시작되는 부분의 인덱스를 가져오기 때문에 파싱해서 값을 추출할 때마다 substring을 이용해서 indexOf가 -1값이 나올 때가지 반복한다.

**◈ 집 앞 카페를 가져오기 위한 사용자의 GPS 정보 Tracking**

: 집 앞 카페를 가져오기 위해서는 사용자의 현재 주소 정보를 Tracking 해야한다. 이를 위해서는 사용자에게 권한을 얻어야 한다. 그래서 GPSInfo 클래스를 이용하여서 사용자에게 권한 요청을 한 후에 사용자에게 권한을 얻을 시에 네트워크 정보로부터 위치값을 가져온 뒤 latitude, longtitude 값을 가져와서 검색하였다.

**◈ Naver지도 api를 주소 정보를 활용한 마커 찍기**

: 처음에는 다음 지도를 사용할 예정이었으나, 복잡한 절차와 불친절한 설명으로 Naver 지도를 사용하게 되었다. 마커 모양을 커스텀화하여서 사용하였다. 해시태그를 통한 장소 검색 결과가 있을 경우에는 GeoCode를 사용해 장소의 위도-경도 값을 가져와서 마커를 찍어주어 사용자의 편의성을 도왔다.

**III. 활용**

인스타그램의 일일 이용자 수는 4억명이 넘고, 셀 수 없이 많은 게시물들이 쏟아져 나온다. 그 게시글 가운데 내가 원하는 정보만을 찾기는 어려운 일이다. 이 어플리케이션을 사용하면 사용자가 미리 선택한 관심사를 포함하는 게시글을 보여주므로 관심사에 부합하는 게시글을 골라 볼 수 있다.

또한 게시글에 포함된 카페의 정보를 자동으로 인식해 위치 정보 및 가격 정보 등을 제공하기 때문에 다른 포털 사이트에 따로 카페 이름을 검색해 정보를 찾는 과정을 거치지 않아도 돼 편리하다.

회원가입 절차를 거치지 않아도 페이스북 계정이 있다면 페이스북 계정을 통해 서비스를 편리하게 이용할 수 있다.

**IV. 한계점**

1. 태그를 통한 장소검색의 한계

인스티그램은 인스타그램에 등록된 게시글에 기반한다. 따라서 게시글에 등록된 데이터를 가지고 검색을 진행하게 되는데 우리가 기지고 있는 데이터는 사용자가 입력한 태그뿐이다. 태그에 장소정보를 정확하게 입력하는 사람이 있는가 하면 대부분의 사람이 장소에 대한 정보를 입력조차 안한다. 우리는 조금이라도 언급된 지역정보를 수집하여 최대한 관련성 깊은 카페의 주소를 출력하는 알고리즘을 개발하였다. 하지만 태그에 장소에 대한 정보가 부족하고 너무 많은 상호명 때문에 검색 속도와 정확도가 떨어진다.

1. 검색결과 이미지의 무관련성

인스티그램은 모든 이미지는 인스타그램에서 가져온다. 하지만 인스타그램은 소셜 미디어로 신뢰할 수 있는 데이터가 많지 않다. 그래서 카페를 검색하더라고 엉뚱한 학교나 병원이 나올 수 있다. 이미지 처리를 통해 카페를 제외한 이미지는 걸러주면 되지만 다양한 유저들의 다른 인스타그램 특성 상 비전API로 학습시키는 것에 한계가 있었기 때문에 이 점은 처리를 하지 못했다. 때문에 관련성 없는 이미지가 표시될 수 있다.

1. 등록되지 않은 데이터에 의한 한계

인스타그램에는 희귀정보가 올라온다. 사용자가 많다보니 구글검색으로 나오지 않는 카페정보가 있다. 그런 경우 아무리 장소에 대한 정보가 많다 하더라도 검색결과가 정확할 수 없다.

1. 최신글 한계

인스티그램은 인스타그램에 등록된 게시글 중 최신글 72개를 표시힌다. 그래서 얼마전에 추천받은 카페는 다음번에는 추천받지 못할 수 있다. 이는 SNS의 실시간 업로드 특성 때문이다. 인스타그램을 포함한 SNS 서비스는 모두 이런 한계점을 가지고 있다.

1. 해시태그의 무관련성

인스타그램 해시태그는 검색시 필터 역할을 해준다. 사용자들은 최대한 자신의 게시글을 대중에게 노출시키기 위해 해시태그를 무작정 입력한다. 그러기 때문에 게시글과 전혀 관련성 없는 게시글이 태그 검색으로 나올 수 있다.