

파워블로그 분석을 통한
유행 패션 및 연관 트렌드 정보제공 앱 개발

01반 16팀
20180369 박세훈 20180388 이재환

Aim & Background (2장)

- 목표

최신 유행 패션에 관한 정보들을 사용자가 어플을 통해 간단하게 제공받으며, 사용자가 수많은 패션 정보들을 모니터링하여 패션 트렌드를 일일이 검색하는 시간을 절약하게 하기 위함.

- 핵심내용

한국 젊은 세대들의 패션 트렌드는 인터넷 매체들의 영향을 많이 받는다. 이런 젊은 세대들에게 인터넷 매체에 있는 패션 정보들 중 유행 패션의 정보들을 선별하고, 유행하는 패션 뿐만 아니라 그와 연관된 패션 트렌드들까지 사용자에게 제공한다.

- 중요성

정보매체의 발달로 패션 정보가 점점 과도하게 넘쳐나고 있어서, 개개인이 시간을 내서 유행 패션 정보를 선별하여 꾸준히 접하기란 쉬운 일이 아니다. 이 어플은 개개인 대신 방대한 패션 정보들을 분석하고 패션 트렌드들을 뽑아서 이용자에게 제공하기 때문에, 개개인이 유행 패션 정보를 얻는데 투자해야 할 시간을 효과적으로 줄여준다.



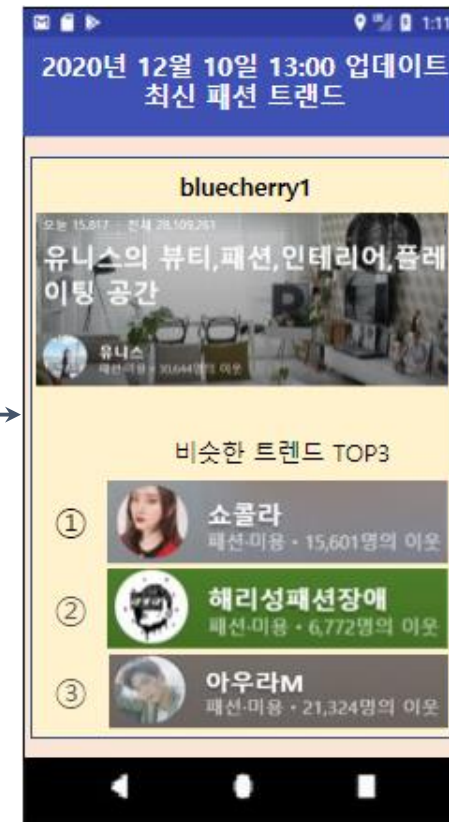
[1]

[2][3][4]



click

[5][6][7]



Aim & Background (2장)

- 배경설명, 사례분석

집단주의 문화권인 한국에서는, 한국인은 남들의 시선에 민감하므로, 패션 트렌드에도 민감하다.[8] 그리고 한국의 젊은 세대들은 어릴때부터 스마트 디바이스를 끼고 생활하였기 때문에, 그들의 패션 트렌드는 스마트 디바이스와 같이 발전한 인터넷 매체, 인플루언서들의 영향을 받는 현상을 보인다.[9][10] 그러나 현재 들어서 인터넷에서 얻을 수 있는 유행에 대한 정보들이 과도할 정도로 많아지게 되면서, 정보를 얻는데 투자하는 시간도 점점 늘어나게 되는 문제가 발생하였다.[11]

- 문제정의

case 1. 넘쳐나는 패션 정보들로 인해, 유행이 어떤 식으로 흘러가고 있는지를 알기 위해서는 상당히 많은 시간을 투자해야 한다.

case 2. 비록 시간을 투자해서 한 분야의 유행 패션 정보를 찾았다 하더라도, 이와 연관된 비슷한 성향의 패션 트렌드를 찾는 데에 또다시 시간을 투자해야 한다.

이를 통해 개인이 시간을 내서 패션 트렌드에 대한 정보를 꾸준히 얻는 것이 점점 힘들어지고 있다는 문제점을 찾아낼 수 있다.

- 극복방안

1. 패션 정보들 중 유행 패션 정보를 선별하여 사용자에게 제공하는 시스템 구축

2. 사용자에게 제공된 패션 트렌드와 성향이 유사한 '연관된 트렌드 정보'도 같이 제공하는 기능 구현

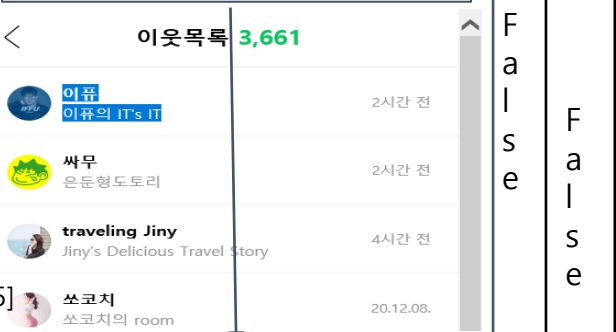
이러한 기능을 탑재한 어플을 개발하여, 개개인들이 수많은 패션 정보들을 모니터링하여 패션 트렌드를 일일이 검색하는 시간을 절약하게 한다.

Contents (3장)

- 전체적인 앱 구조 (SW 관점)

```
String[10] list;      int count = 0;      int number;
String[3] sublist;   int subcount = 0;
```

1) 저장된 list의 아이디를 위에서부터 차례로 하나 선정 및 그 아이디의 이웃 목록 스크랩.
2) 이웃 목록에서 이웃 아이디 블로그로 하이퍼링크 되어있는 것을 위에서부터 차례로 하나 선정 및 그 이웃 아이디 블로그 스크랩



패션 정보를 다루는지 판단

인플루언서 인지 판단

sublist[subcount]에 그 네이버 아이디 저장 및 subcount++;

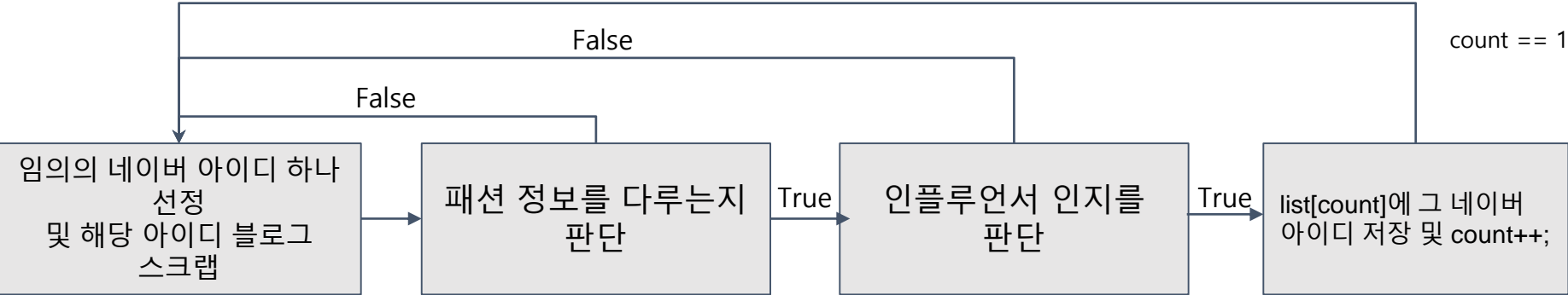
사용자 정보 제공 및 알고리즘 종료

count != 10

count == 10

False
False

subcount == 3



NAVER

아이디

bushcherry1

5~20자의 영문 소문자, 숫자와 특수기호(.),(-)만 사용 가능합니다.

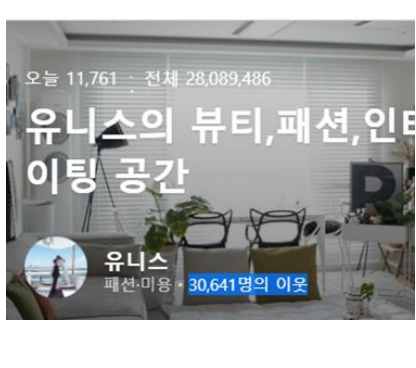
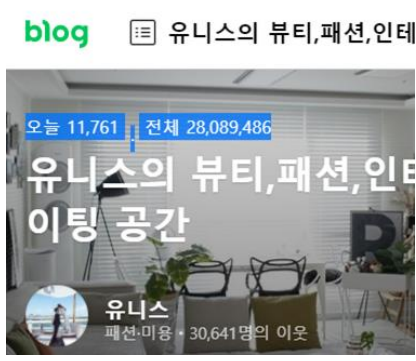
[12]

(아이디)의 경우의 수

$$= \sum_{n=5}^{20} (26+10+2)^n$$

해당 (아이디) 블로그

[https://m.blog.naver.com/PostList.nhn?blogId=\(아이디\)](https://m.blog.naver.com/PostList.nhn?blogId=(아이디))



count	list
0	bushcherry1
1	naverID1
2	naverID2
...	...
8	naverID8
9	naverID9

Contents (3장)

- 핵심 구성 요소 설명 및 구현 방법

임의의 네이버 아이디
하나 선정
및
해당 아이디 블로그
스크랩

- 설명 : 네이버 아이디가 a~z 0~9 -/_의 조합이라는 것을 이용해서 5자리~20 자리의 임의의 아이디를 불러오고, 그 임의의 아이디에 대한 블로그 화면을 스크랩한다.
- 구현 : random()과 boolean()을 이용해 랜덤으로 true, false 를 리턴. true일 시 임의의 소문자를, false 임의의 랜덤 한 숫자를 가져온다. URLConnection을 통해서 개발 앱과 웹을 연결하고, InputStream()을 통해서 전체 텍스트를 읽어온다. [16]

패션 정보를 다루는지
판단

- 설명 : 모바일 웹에서는 블로그 프로필에 카테고리가 명시되어 있으므로, 카테고리에 패션이 포함되는 지 확인
- 구현 : indexOf("패션.미용")을 통해서 패션이란 카테고리를 갖고 있는지 확인한다.

// indexOf() 는 특정
문자나 문자열이 앞에서부터 처음 발견되는 인덱스를 반환하며
만약 찾지 못했을 경우 "-1"을 반환한다.[18]

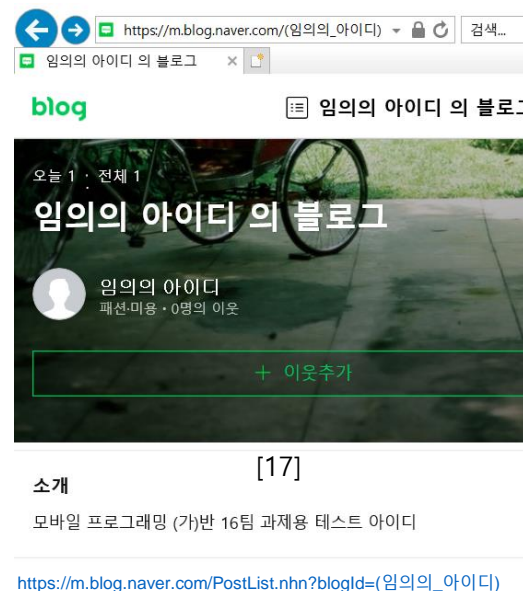
인플루언서 인지를
판단

- 설명 : 모바일 웹에서는 블로그 프로필에 오늘자 방문자수와 전체 방문자수가 모두 명시되어 있으므로 오늘자 방문자수와 전체 방문자 수가 일정 수 이상(오늘 100명, 전체 100000명 이상)이면 '인플루언서' 라고 정의한다.
- 구현 : scanner를 복합적으로 사용하여 오늘 방문자수와 전체 방문자 수를 int로 받고, '오늘자 방문자 수 >= 100명' && '전체 방문자 수 >= 100000명 이상'이면 인플루언서로 분류한다.

아이디

(임의의_아이디)

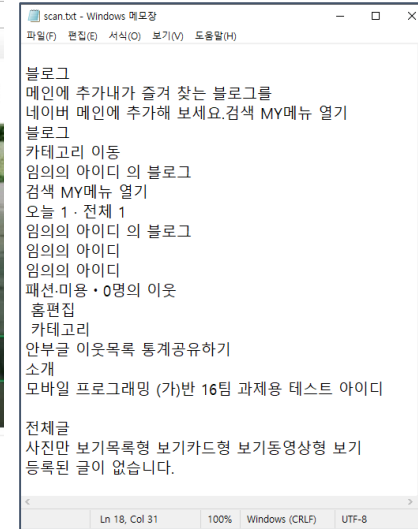
5~20자의 영문 소문자, 숫자와 특수기호(.,-)만 사용 가능합니다.



"패션.미용"이라는 문자열이 있으면, 블로그가 패션 정보를 다룬다고 판단하고, 없으면 다루지 않는 것으로 판단한다. 모바일 블로그의 경우, 텍스트의 형식이 똑같기 때문에, 모든 블로그 카테고리는 항상 스크랩한 scan.txt의 13번째 줄에 위치해 있다. 그렇기에 13번째 줄의 문자열에 "패션.미용"이 있는지 없는지 판단하면 된다



오늘자 방문수가 100명 이상이고, 전체 방문자 수가 100000명 이상이면, 인플루언서라고 정의한다. 조건 중 하나라도 만족하지 아니할시엔 인플루언서로 취급하지 않는다. 모바일 블로그의 경우, 텍스트의 형식이 똑같기 때문에, 모든 오늘 방문자 및 전체 방문자 수는 항상 스크랩한 scan.txt의 9번째 줄에 위치해 있다. 그런데 보면 9번째줄에는 방문자수 말고도 필요없는 문자도 있다. 9번째 line은 항상 "오늘"+" "(오늘 방문자수)+" "+"+" "+"+" "(전체 방문자수) 형식으로 나누어져 있다. JAVA의 Scanner클래스의 next()함수가 " "이전 까지의 문자열을 받는다는 성질을 이용하면, next()->nextInt()->next()->next()->nextInt()로 오늘 방문자수와 전체 방문자 수를 int로 받아들이고, 조건문을 적용하여 인플루언서 인지 아닌지를 판단할수 있다.[19]



블로그의 텍스트를 읽어온다

Contents (3장)

- 핵심 구성 요소 설명 및 구현 방법

list[count]에 그
네이버 아이디 저장
및 count++;

1) 저장된 list의 아이디를
위에서부터 차례로 하나 선정 및
그 아이디의 이웃 목록 스캔.
2) 이웃 목록에서 이웃 아이디
블로그로 하이퍼링크 되어있는
것을 위에서부터 차례로 하나
선정 및 그 이웃 아이디 블로그
스크랩

패션 인플루언서 인지
판단

sublist[subcount]에
그 네이버 아이디
저장 및 subcount++;

사용자 정보 제공 및
알고리즘 종료

count	1	2	3	4	5	6	7	8	9	10
list	ID 1	ID 2	ID 3	ID 4	ID 5	ID 6	ID 7	ID 8	ID 9	ID 10

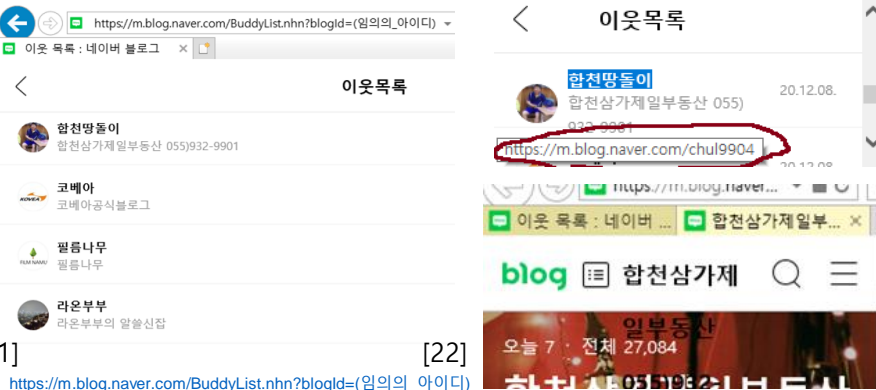
- 설명 : 기준에 맞는 아이디가 선정되면 count를 1 증가시키고(초기 count값은 0), 그 아이디를 list[count]에 저장한다. 이 과정을 count가 10이 될때까지 반복, 최종적으로는 10명의 인플루언서가 선정되게 된다. 또한 10명의 인플루언서를 '전체 방문자수 대비 오늘 방문자 수'가 가장 많은 순으로 재정렬한다.
- 구현 : 데이터베이스에 각각의 조건을 만족시키는 네이버 아이디를 저장하고, for문을 이용해 그럴 때마다 count를 1씩 증가시킨다. 그리고 10명의 인플루언서에 대해 sort 정렬 방식을 이용해 순위를 매긴다.
- 설명 : 위의 그림처럼 이웃 팔로워들의 아이디를 볼 수 있다. 또한, 이웃목록 내의 아이디들은 각각 아이디들의 블로그로 하이퍼링크 되어 있다. 아이디들에 하이퍼링크 되어 있는 블로그 페이지를 위에서 부터 하나씩 불러와 텍스트를 스캔해온다.
- 구현 : URLConnection을 통해서 개발 앱과 팔로워 블로그 프로필 웹을 연결하고[20], InputStream()을 통해서 전체 텍스트를 읽어온다.
- 설명 : 한 인플루언서들에 대해서 가져온 이웃목록 팔로워 블로그의 카테고리과 오늘자 방문자수와 전체 방문자 수를 가지고 팔로워가 '패션 인플루언서'인지 아닌지를 판단한다(기준은 이전과 같다).
- 구현 : 이전 '패션 정보를 다루는 지 판단', '인플루언서 인지 판단'하고 똑같은 구조로 설계한다.

- 설명 : 기준에 맞는 아이디가 선정되면 subcount를 1 증가시키고(초기 subcount값은 0), 그 아이디를 sublist[subcount]에 저장한다. 이 과정을 subcount가 3이 될때까지 반복, 최종적으로는 3명의 인플루언서가 선정되게 된다. 이것을 나머지 10명 인플루언서들의 이웃목록에 대해서도 반복한다.
- 구현 : 데이터베이스에 각각의 조건을 만족시키는 네이버 아이디를 저장하고, for문을 이용해 그럴 때마다 subcount를 1씩 증가시킨다. 3명에 인플루언서에 대해서도 sort정렬 방식을 이용하여 대해 '오늘 방문수 / 전체 방문수'가 큰 순서대로 내림차순으로 재정렬한다.
- 설명 : list에 저장된 인플루언서 총 10명에 대한 정보와 각 인플루언서 중에서 sublist에 들어간 팔로워들(인플루언서)들에 대한 ID에 대한 정보를 UI로 보여준다.
- 구현 : 각 인플루언서들의 ID 클릭시 3명의 팔로워들의 ID가 보이게 한다. OnClick리스너 및 URLConeection()를 이용하여, list에 저장된 인플루언서 ID 10개를 (button) 형식으로 background에 제시한다. background의 (button)들을 누를시에 해당 ID에 대한 foreground가 뜬다. foreground는 (Textview)로 해당 인플루언서 ID를 사용자에게 보여주고 해당 인플루언서 ID의 블로그로 직행하는 (Button) 1개와 해당 인플루언서의 팔로워 3명의 블로그로 직행하는 (button) 3개로 이루어져 있다.

재정렬

ID 1	ID 2	ID 3	ID 4	ID 5	ID 6	ID 7	ID 8	ID 9	ID 10
RE 1	RE 2	RE 3	RE 4	RE 5	RE 6	RE 7	RE 8	RE 9	RE 10

'오늘 방문수 / 전체 방문수'가 큰 순서대로 내림차순으로 재정렬한다. 그 이유는 전체 방문자 수는 누적이기 때문에 오래된 블로그의 경우 전체 방문자의 수는 많을 수 밖에 없지만, 그에 비해 오늘자 방문자 수가 적다는 것은 적은 인기, 즉 주류 트렌드에서 약간 벗어난 정보가 있을 가능성이 있기 때문이다.



[21] [https://m.blog.naver.com/BuddyList.nhn?blogId=\(임의의_아이디\)](https://m.blog.naver.com/BuddyList.nhn?blogId=(임의의_아이디)) [22]

count	1			2			...	10		
list	ID1			ID2			...	ID10		
sub count	(1) 1	(1) 2	(1) 3	(2) 1	(2) 2	(2) 3	...	(10) 1	(10) 2	(10) 3
sublist	Folq 1	Folq 2	Fol1 3	Fol2 1	Fol2 2	Fol2 3	...	Fol10 1	Fol10 2	Fol10 3

(button) 인플루언서 ID1

(button) 인플루언서 ID2

(button) 인플루언서 ID3

⋮

background.xml

(Textview) 인플루언서 ID

(Button) 인플루언서 블로그 URL

(Textview) "비슷한 트렌드 TOP3"

(button) 팔로워1 블로그 URL

(button) 팔로워2 블로그 URL

(button) 팔로워3 블로그 URL

foreground.xml

Reference (페이지 제한 없음)

- [1]<https://www.dreamstime.com/illustration/fashion-app.html>
- [2]<https://m.blog.naver.com/PostList.nhn?blogId=bushcherry1>
- [3]<https://m.blog.naver.com/PostList.nhn?blogId=coolps8>
- [4]<https://m.blog.naver.com/PostList.nhn?blogId=damsluv>
- [5]<https://m.blog.naver.com/PostList.nhn?blogId=iamchocolat>
- [6]<https://m.blog.naver.com/PostList.nhn?blogId=goflvotus>
- [7]https://m.blog.naver.com/PostList.nhn?blogId=aura_m
- [8]<https://www.asiae.co.kr/article/2020091814373571286>
- [9]<https://www.sisajournal.com/news/articleView.html?idxno=194599>

Reference (페이지 제한 없음)

- [10]<https://www.hankyung.com/economy/article/2020071948261>
- [11]<http://www.munhwa.com/news/view.html?no=2018011701032809000001>
- [12]https://nid.naver.com/user2/V2Join.nhn?token_sjoin=XBs4bjJiGeeGpHR1&chk_all=on&termsService=on&termsPrivacy=on&termsLocation=Y&termsEmail=Y
- [13]<https://m.blog.naver.com/PostList.nhn?blogId=bushcherry1>
- [14]<https://admin.blog.naver.com/testaccount>
- [15]<https://m.blog.naver.com/BuddyList.nhn?blogId=haejukdl>

Reference (페이지 제한 없음)

[16]<https://coding-factory.tistory.com/281>

[17]<https://blog.naver.com/testaccount>

[18]https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/String/indexOf

[19]<https://maktooob.tistory.com/17>

[20]<https://www.jejusarang.com/?1225201715&PHPSESSID=1452fb7ec7a5e5e95971c802f3f1eec1>

[21]<https://m.blog.naver.com/BuddyList.nhn?blogId=haejukdl>

[22]<https://m.blog.naver.com/chul9904>

Appendix (페이지 제한 없음)

- 깃허브 레파지토리 (작성 중)

<https://github.com/vendergood/TeamProject>

Peer review (타 분반에서 3명씩 평가)

- 신규성

- 새로운 부분이 있는지? (+1)
- 얼마나 창의적인지? (+1)
- 실제로 수요가 있을지? (+1)

- 진보성

- 비교/분석이 충분히 이루어졌는지? (+1)
- 해당 분야에서 얼마나 경쟁력 있는지? (+1)
- 다른 응용에도 파급력이 있는지? (+1)

- 완결성

- 얼마나 잘 구현하였는지? 제시된 방법이 실현 가능성이 있을지? (+1)
- 핵심 내용 및 세부 사항들을 이해하기 쉽게 잘 전달하였는지? (+1)
- 기술 난이도가 충분히 도전적이었는지? (+1)
- 창의/융합/공동체/의사소통/리더쉽/글로벌 역량 관련 기타 요소 (+1)