



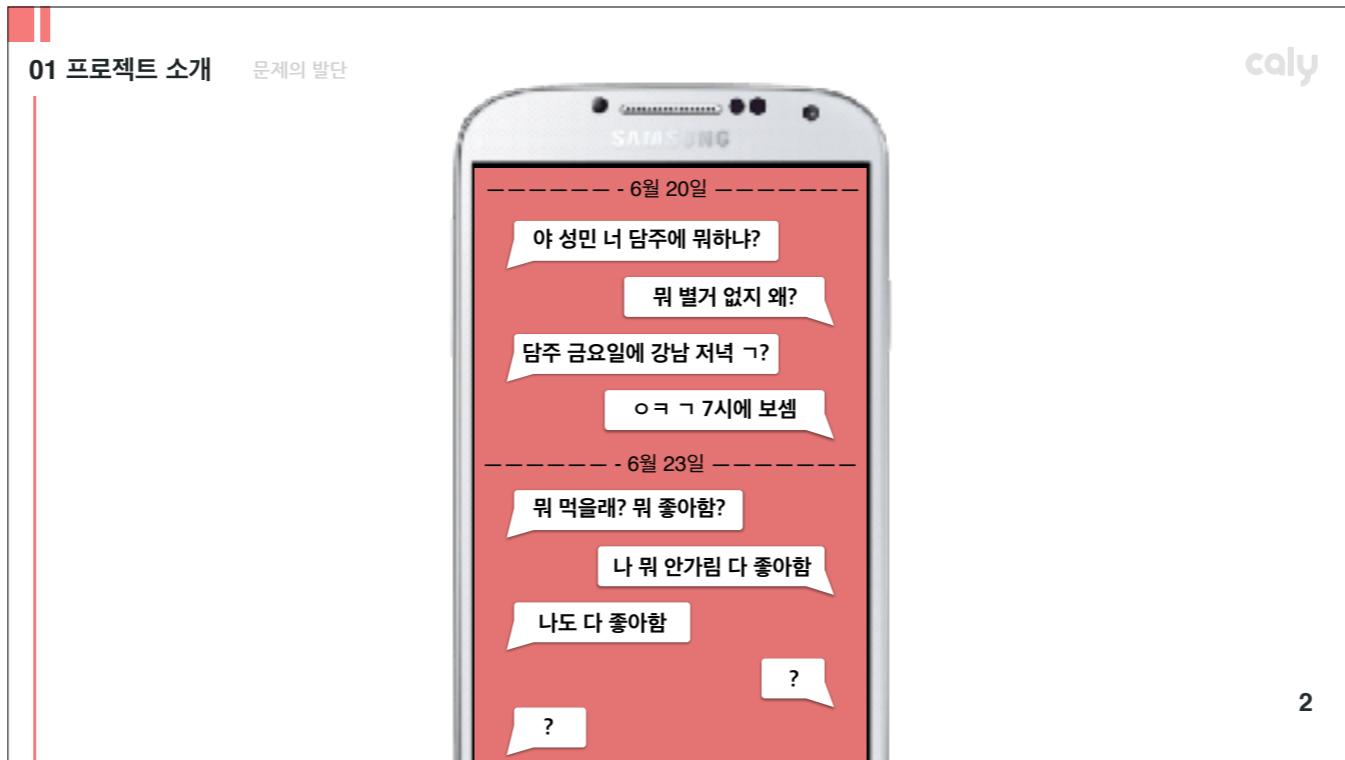
안녕하세요 일정 기반, 장소 추천 서비스인 캘리 발표를 시작하겠습니다.

먼저 오른쪽에 보이시는 화면은 저희 캘리의 실제로 워킹하는 프로덕트인데요, 누구보다 빠르게 4월 중순에 이미 런칭을 했고,
이 서비스를 지금까지 실운영하면서 겪은 문제와 이슈들을 어떻게 개선하고 발전시켜왔는지, 그 이야기를 들려드리겠습니다.

01 프로젝트 소개

문제의 발단

caly

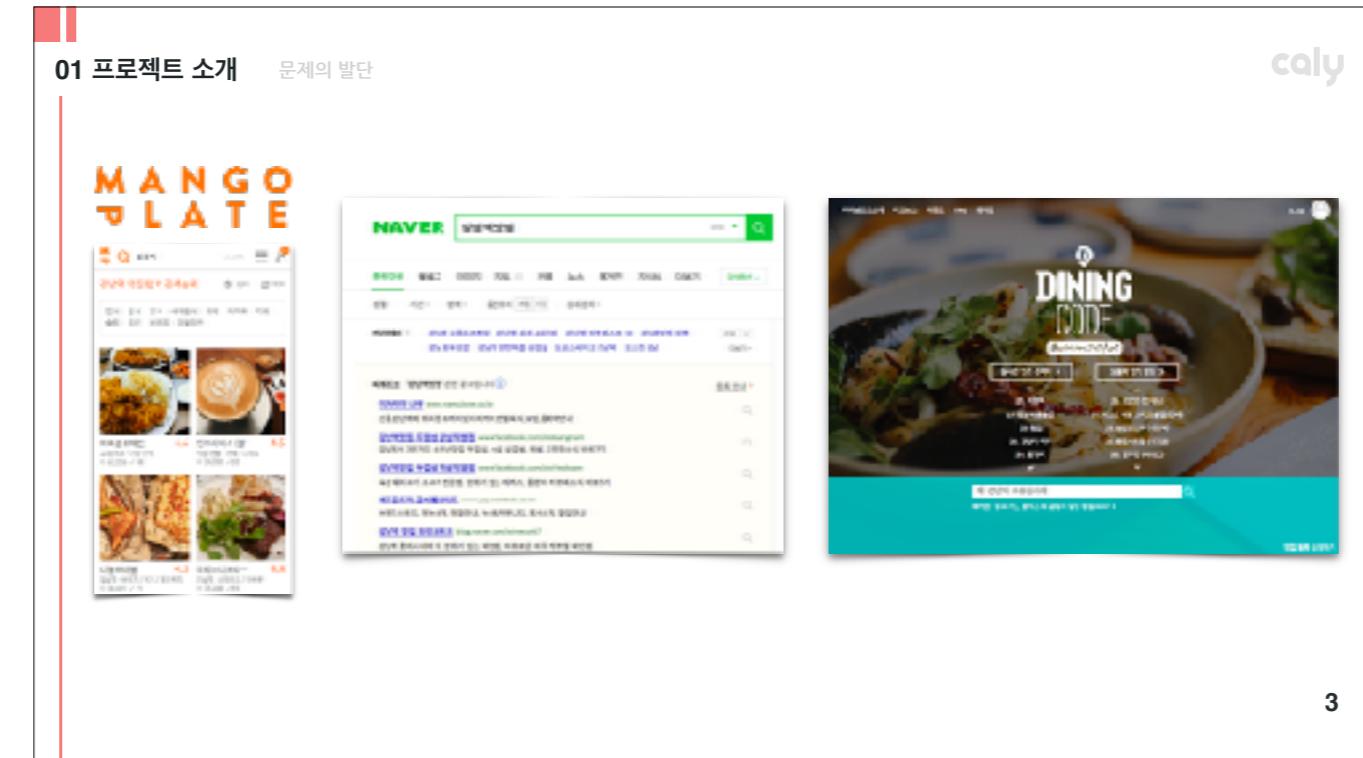


2

프로젝트 시작할때 제가 이 친구와 이렇게 약속을 잡았습니다. 당일날이 되어 허겁지겁 장소를 찾고, 그러다보니까 정작 좋은 곳에 못 갔죠. 이런 문제가 저만 그럴 것 같지 않는 않더라고요.

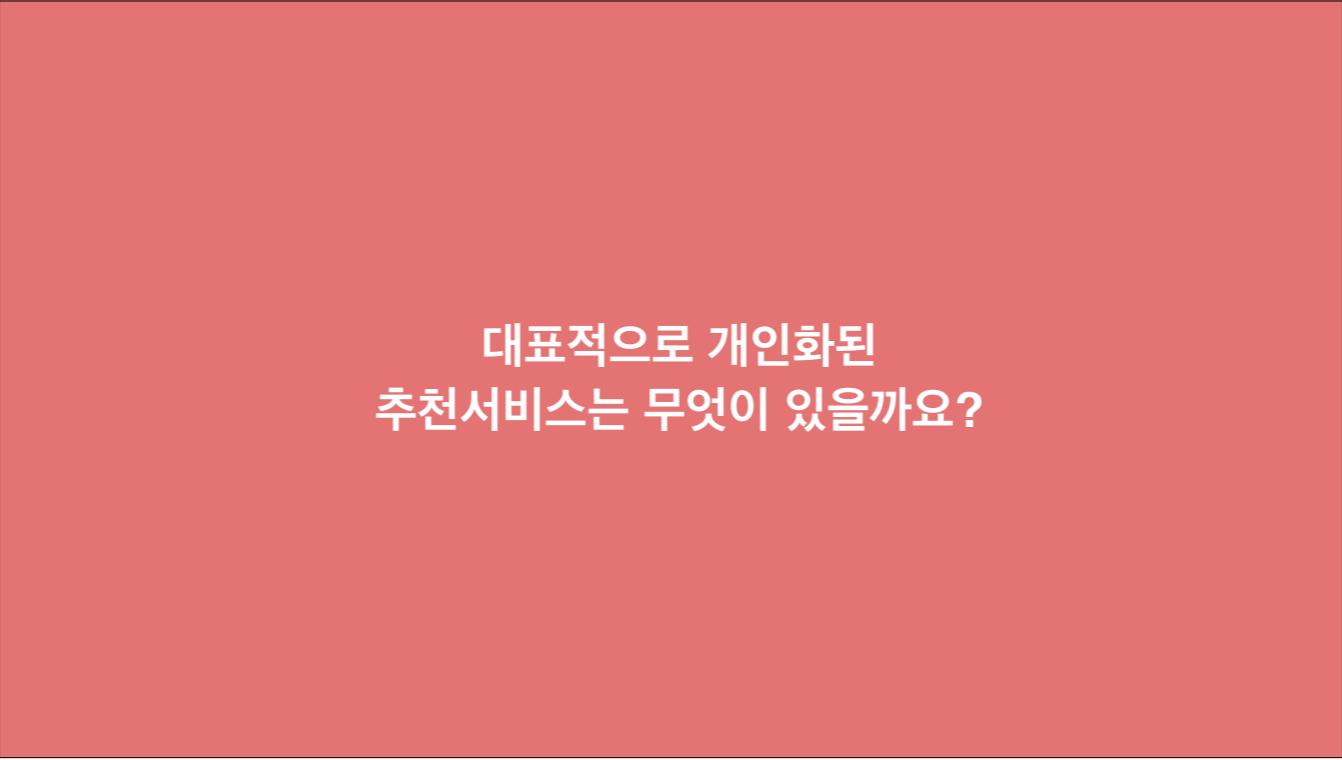
01 프로젝트 소개 문제의 발단

caly



3

왜그럴까 생각을 해봤더니 검색을 하면 매일 보던것만 나오고 광고성 블로그도 많아서 도움이 안되더라고요. 이와 같이 약속이 생길 때, 보통은 미리 고민을 안 하거나, 하더라도 피곤합니다. 왜냐하면 평소 바쁘기도 하고 별점과 순위만으로는 정확한 정보를 얻기도 힘들기 때문이죠. 그런데도 이렇게 검색을 해야된다는 것이 불편했습니다. 이러한 정보를 추천 받을 수는 없을까요?

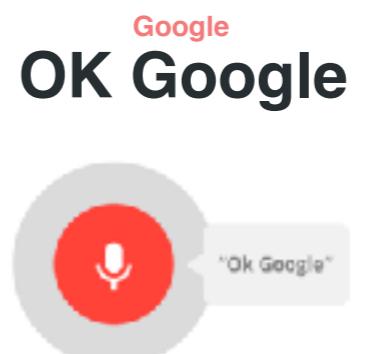
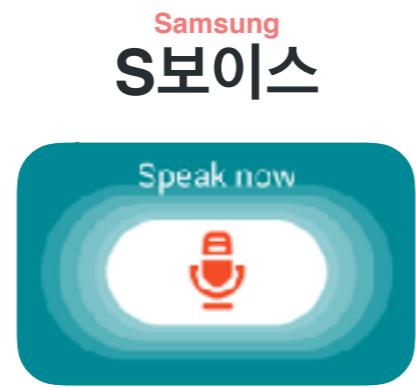


대표적으로 개인화된
추천서비스는 무엇이 있을까요?

있을거라 생각해서 그래서 좀 찾아봤습니다.

01 프로젝트 소개 대표적인 추천 서비스

caly



음성인식 서비스. 문자를 보내주거나 SNS 대신 업로드 가능.

5

바로 시리, S보이스, OK 구글 등 음성지원을 해주는 개인비서 서비스입니다. 이중에서 가장 대중적인 것은 바로 시리죠.

01 프로젝트 소개 시리의 활용성

caly

Nearly 85% of iOS 7 users surveyed have not called upon Siri

통신&모바일 iOS7 사용자 85% "시리 쓴 적 없다"



correctly called intelligent voice. The topic of the questioning was Siri. Those surveyed were asked if they used Siri on iOS 7, what they thought of the usefulness of Apple's voice-activated virtual assistant, and whether Apple had been sold the public when it came to Siri. Those who have not yet used Siri with the tenth build of iOS 7 were assumed as two groups and asked different questions.

Group one was asked if Apple had "oversold the voice recognition capabilities of Siri" and in their high 48% replied in the affirmative. Those in group two were asked to rate the difficulty. 91.8%, 12.1% (288,370, 30,300) "Extremely accurate", "Quite accurate", who

"AI 음성비서, '날씨 확인에 가장 많이 사용'...애플 '시리' 1위

기사입력 2017-04-01 11:21 | 최종편집 기자



-010-8 870-1111(한국전기통신공사 대체 번호) 0300000-0000

조사 결과에 따르면 AI 음성비서를 통해 가장 많이 이용하는 서비스는 날씨 확인(14%), 관심 주제에 대한 인터넷 검색(13%), 여행/장소 예약 검색(4.7%), 일정 확인(4.2%), 행사/자리 대한 푸지정보검색(3%) 등이 AI 음성비서의 가장 많이 이용하는 것으로 나타났다. 관심주제 4.2% 미만은 편차 범위 즉 "그동안 개인적인 서비스를 사용하는 것으로 나타나 사용하는 서비스를 이용하는 경우는 개인적인 것으로 조사된다. 이는 현재 AI 서비스로 기능의 서비스들의 기대수준에 미달하기 때문으로 해석된다. 결론적으로 집이나 교통수단 혼자 사용 Siri는 최종결론이다. 또 결과는 서비스에서 Siri를 이용하고 있는 것으로 조사된다.

자주 쓰이지 않는 이유?

“사용자와 암哑를 띄워한다”

6

가장 많이 알려진 시리조차도 대부분의 사람들이 사용하지 않고, 사용하더라도, 날씨 확인이나 음악 재생 정도의 한정된 기능만 사용한다고 합니다. 기능도 많은데 왜 자주 쓰이지 않는 걸까요? 저희는 그것에 대해 사용자가 그 기능을 알고서 써야한다는 것이 문제라고 생각했습니다.

조금 더 본질적으로 접근해보면, 왜 사용자가 알아야할까요? 시리가 먼저 추천해줄 수는 없을까요?
그 이유로 시리는 사용자가 언제 어느 정보를 원할지, 그 의도를 모르기 때문입니다.

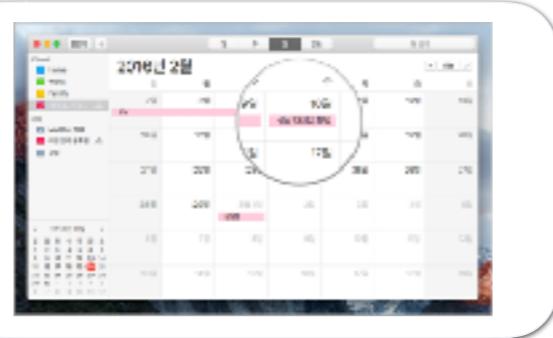
사용자의 정보와 일상을 쉽게 파악할 수 있는 방법이 없을까?

그렇다면, 그 정보를 쉽게 파악할 수 있는 방법이 없을까요?

01 프로젝트 소개

캘리 서비스의 기반

caly



사용자의 일상이 기록된
캘린더 앱

8

저희는 그것이 캘린더 앱이라고 생각했습니다.

사용자의 일상을 캘린더에 하나하나의 일정으로 기입하기 때문이죠

01 프로젝트 소개 캘리 서비스의 기반

+



사용자의 일상이 기록된
캘린더 앱

사용자의 선호도가
반영된 추천

9

이 사용자 정보와 개인화된 추천이 반영된다면 굉장히 시너지가 날 것으로 기대를 했구요, 그 결과

caly

개개인의 일정을 분석하여
스케줄에 어울리는 정보를
추천해주자 !

유저의 정보를 분석해서 어울리는 정보를 추천해주는 서비스를 개발하게 되었습니다.
하지만, 먼저 검증해야 할 것이 있습니다.

일반적으로 사람들이 캘린더 앱을 많이 사용하고 있을까?

1 얼마나 사용하고 있을까?

▶ 충분한 유저 Pool 검증

2 언제 어디서의 일정인지, 자세히 기록할까?

▶ 데이터 추출 정확도를 위한 충분한 리소스 검증

3 보통 어떤 캘린더를 어떤 디바이스로 사용할까?

▶ 최소한의 개발스코프 산정

바로 사람들이 캘린더를 충분히 사용하는가?에 대한 질문인데요,
그것을 알기 위해 다음 3가지 질문을 정의했습니다.

첫번째, 캘린더의 실사용률

두번째, 일정 하나하나에 대해 상세히 기록하는지

세번째, 어떤 캘린더를 주로 사용하는지, 무엇으로 캘린더를 기입하고 확인하는지였습니다.

이 세가지 질문에 대한 응답이 긍정적이라면, 다음과 같은 근거가 될 것입니다.

01 프로젝트 소개

일정관리 실태조사

caly



(1월 9일 ~ 1월 16일)

1,111명

구글 설문을 이용해 총 1,062명에게 설문조사를 받아,
(중복 설문응답자와 어뷰징 데이터 제외) 실데이터 1,052개 수집

454명

학생 집단

490명

회사원 집단

108명

기타 집단

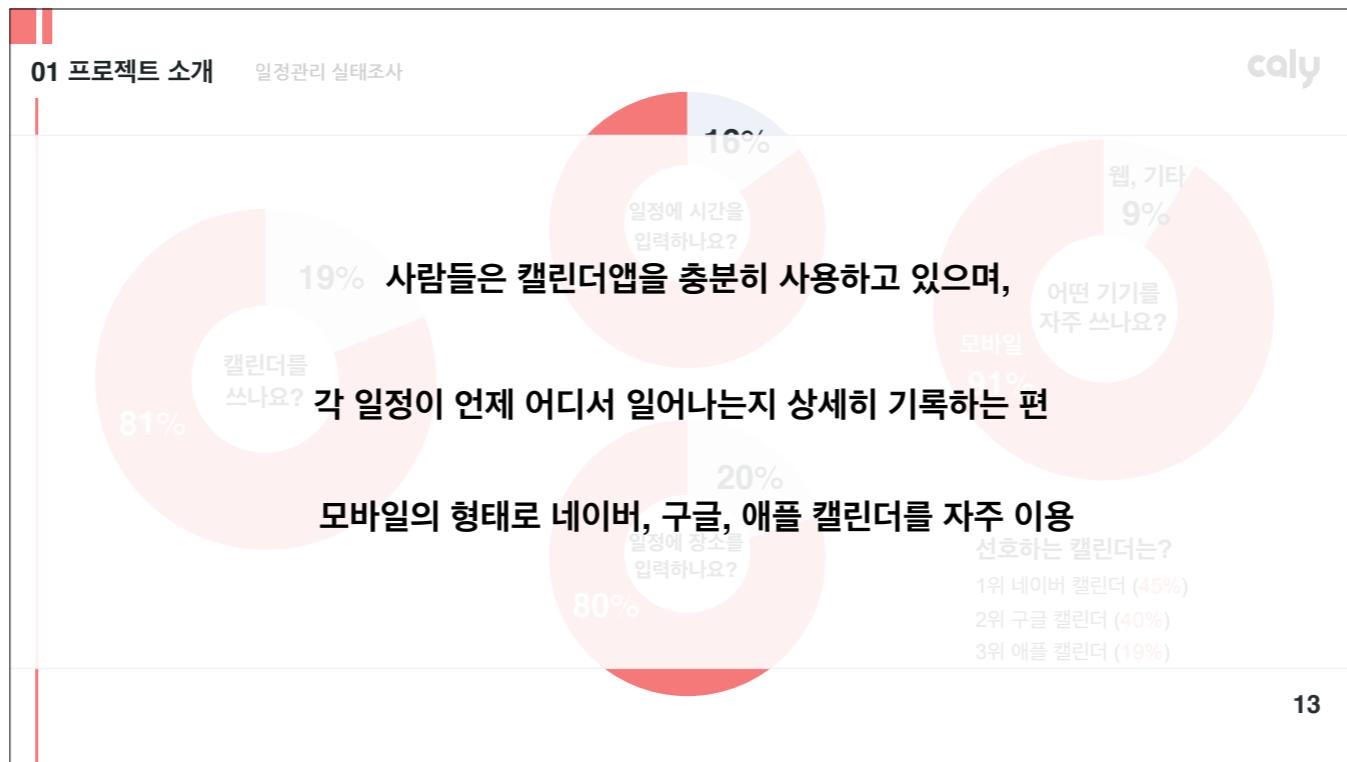
12

저희는 일정관리 실태조사를 일주일에 걸쳐서 진행하였고,
천명이 넘는 표본 시장이 여러 직업군에 골고루 분포되어 실시된 것을 확인하실 수 있습니다.

여기서 질문 하나 드리겠습니다.

보통 사람들이 얼마나 캘린더를 사용하고 있을까요?

저희가 이 조사가 끝나기 전에 주변에 물어봤을 때는 10명 중에 4명 정도가 사용하지 않는다고 하더라구요. 60%는 상당히 낮은 수치죠.



하지만 놀랍게도, 응답자의 81%가 사용하고 있다고 응답을 했습니다.

일정 기입도 목적 정도만 간단히 서술할 거라 생각했지만 두 경우 모두 80% 이상이 입력한다고 하더라구요. 이용 행태 모바일, 선후 캘린더는 다음과 같습니다.

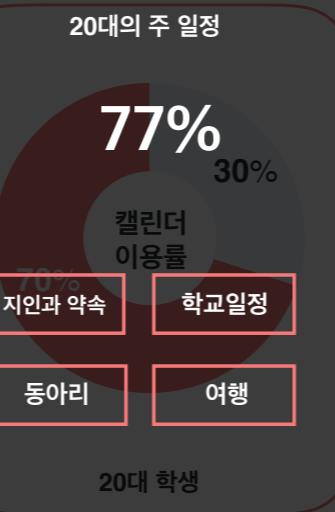
총 정리를 하면 다음과 같습니다.

여기서 마지막 네이버, 구글, 애플 캘린더는 저희가 최소한 지원해야할 캘린더 서버 범위가 됩니다.

01 프로젝트 소개

일정관리 실태조사 - 유저 세그먼테이션

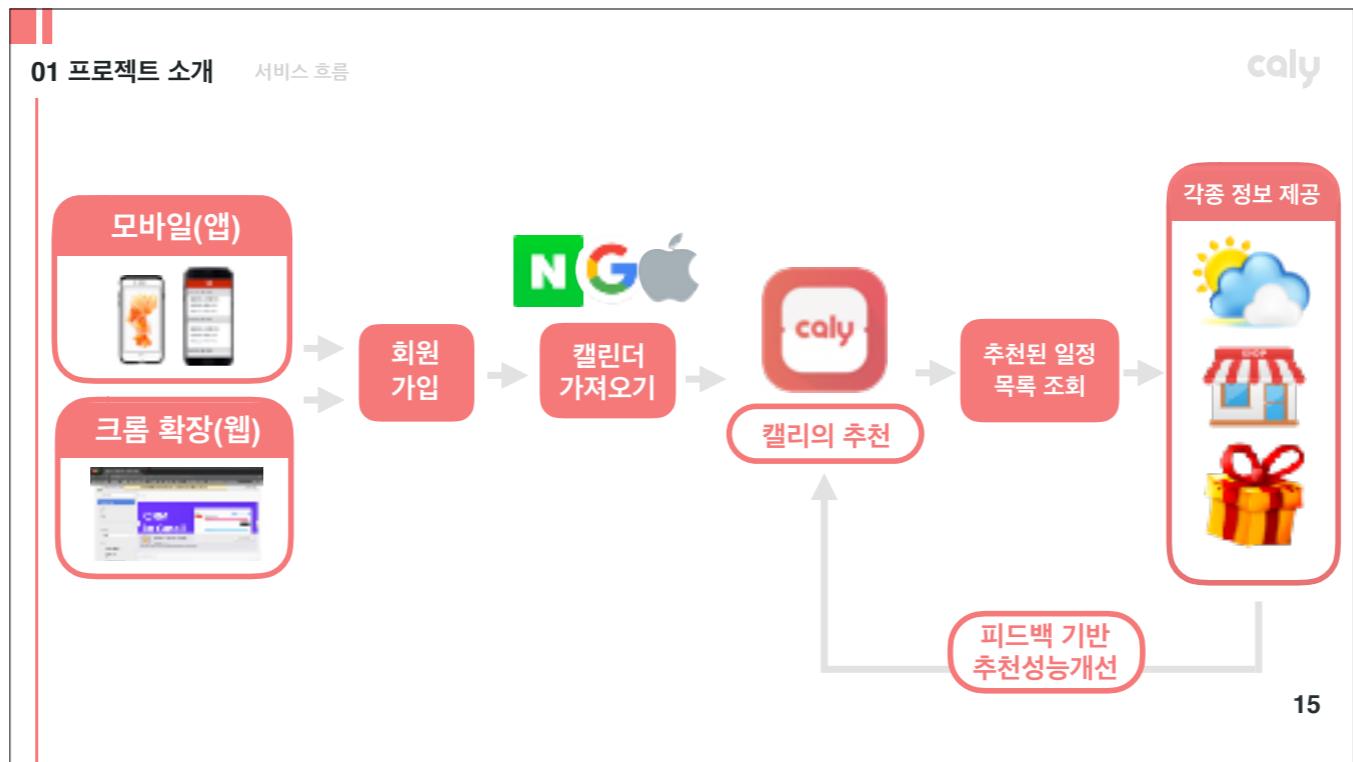
caly



14

유저 타겟팅도 선정을 했는데요, 캘린더 사용률은 다음과 같습니다.
이용 수치만 봤을 때는 당연히 30대를 선택해야할 것 같지만,
저희는 20대를 선택했습니다.

왜일까요? 20대는 대부분의 일정이 문화생활을 소비하는 반면,
30대는 회사 업무가 대부분이었기 때문입니다.
엔터테인먼트 정보를 제공하는 저희 서비스 특성상 20대가 더 적합하다고 판단했습니다.
저희가 접근하기도 좋고, 이들의 주행동 패턴을 알고있는 것도 장점이구요.



서비스 흐름을 간략히 보여드리죠.

유저는 다음과 같이 앱/웹의 형태로 회원가입을 진행하고 원하는 캘린더 서버에 동기화 합니다.

그 이후, 캘리가 추천을 하고나면 그 일정에 대해 날씨/가게/선물 추천정보를 선택하죠.

이 유저의 선택은 모두 행동로그로 수집하고 있기 때문에 피드백 기반으로 추천성능을 개선시키고 있습니다.

서비스 흐름을 현재 / 미래 둘 중 확실히 선택해서 설명?

초기 모델의 핵심 목표

1. 목적과 시간, 장소 정보 추출을 위해 사용자의 일정 확보에 집중
2. 사용자의 가입을 극대화하기 위해, 가입 절차 간소화

이 서비스 모델은 다음 두 가지 목표에 집중합니다.

먼저 사용자 일정 확보에 집중할 것이구요, 가입 극대화를 위해 가입 절차를 간소화하는 것입니다

02 MVP 모델 추천 모듈

caly

시간/장소/목적 분석

캘린더 서버 동기화 후
유저 일정의 목적, 시간,
장소 정보 추출 및 분석

추천정보 수집

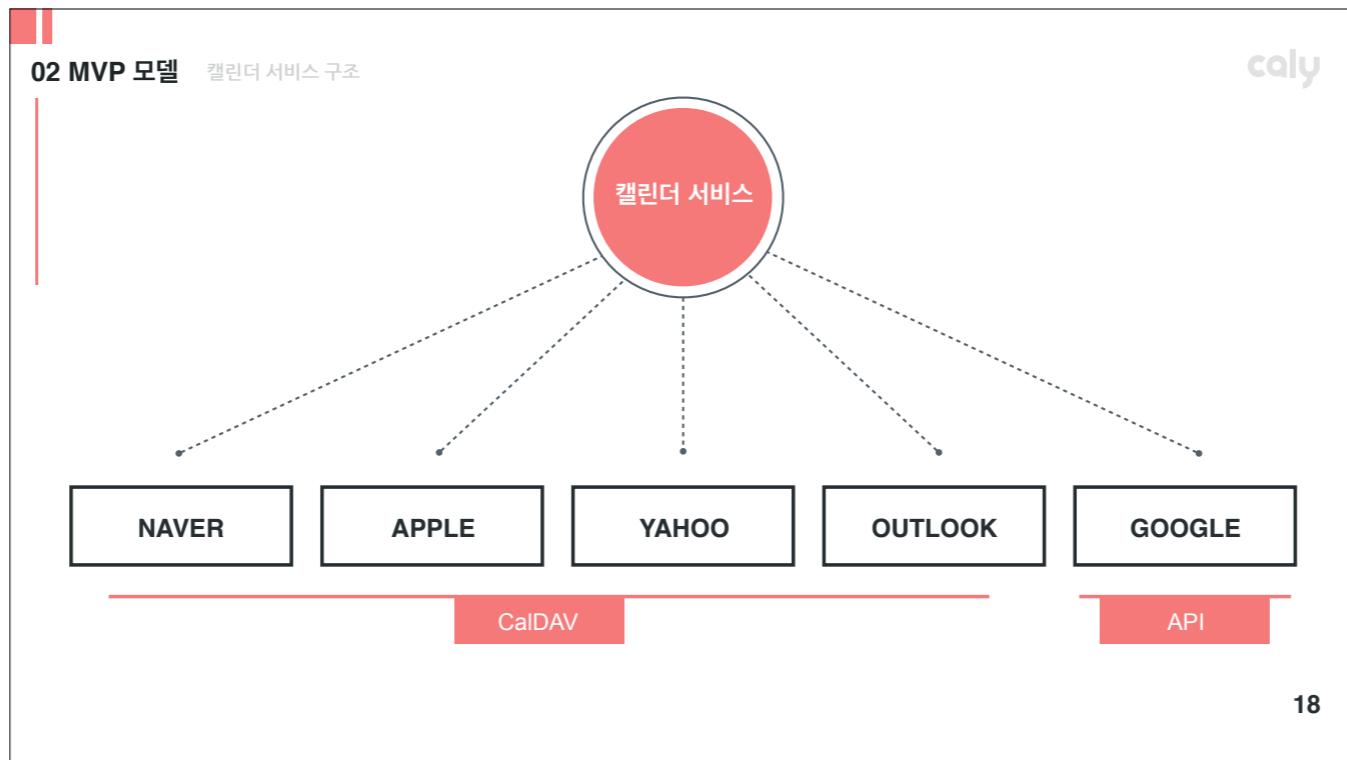
유저들에게
추천할 식당/카페/놀거리
가게 정보를 수집

추천 자동화

수집된 추천정보를
분석된 패턴을 통해서
랭킹/필터링을 통해 추천 제공

17

추천 모듈은 이 세 가지인데, 가장 먼저 되어야 하는 기능은 사용자의 일정 정보를 얻어와야하는 캘린더 서버 동기화입니다.



대부분의 캘린더 서비스는 CalDAV라는 표준 프로토콜 위에서 운영되고 있으며, 구글은 API로 제공하고 있습니다. NAVER나 APPLE, YAHOO 등은 표준 프로토콜 위에서 동작하긴 하지만, 각자 조금씩 다른 해석으로 통신하기 때문에 하나의 로직으로 통합하는데 어려움을 겪었습니다.

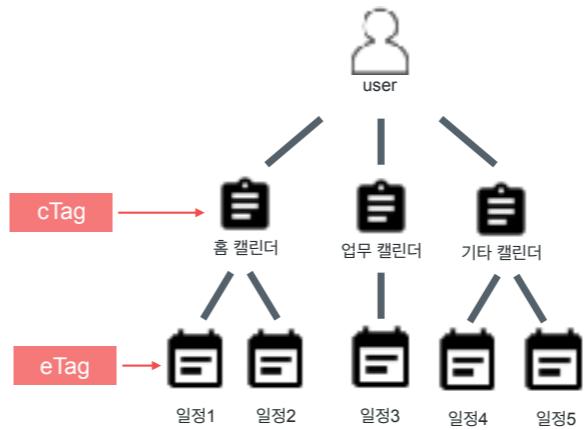
플랫폼 별로 변형되어 있어서 힘들다는걸 시각적으로 표현하자

02 MVP 모델 RFC-4791

caly



관련 프로토콜 표준문서
RFC-4791 CalDAV



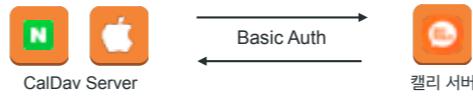
19

CalDAV는 국내 자료가 전무했기 때문에 공식 표준 문서인 RFC를 참고해서 직접 개발을 했는데요,
그 문서에 따르면 캘린더는 다음과 같은 구조를 가집니다.

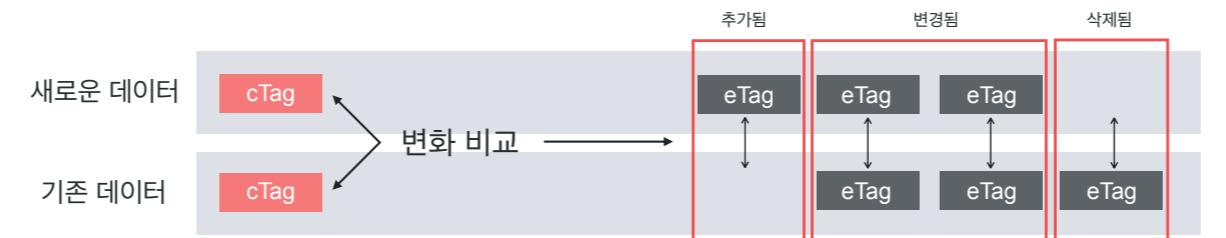
먼저 유저는 홈이나 업무, 기타 등 여러 개의 캘린더를 가지구요, 각 캘린더마다 여러 개의 일정을 가지고 있습니다. 각 캘린더와 각 일정은 고유한 태그인 cTag와 eTag를 가집니다. 변경된 eTag와 cTag만을 비교해
변경된 일정을 알 수 있구요

02 MVP 모델 캘린더 서버 동기화

caly

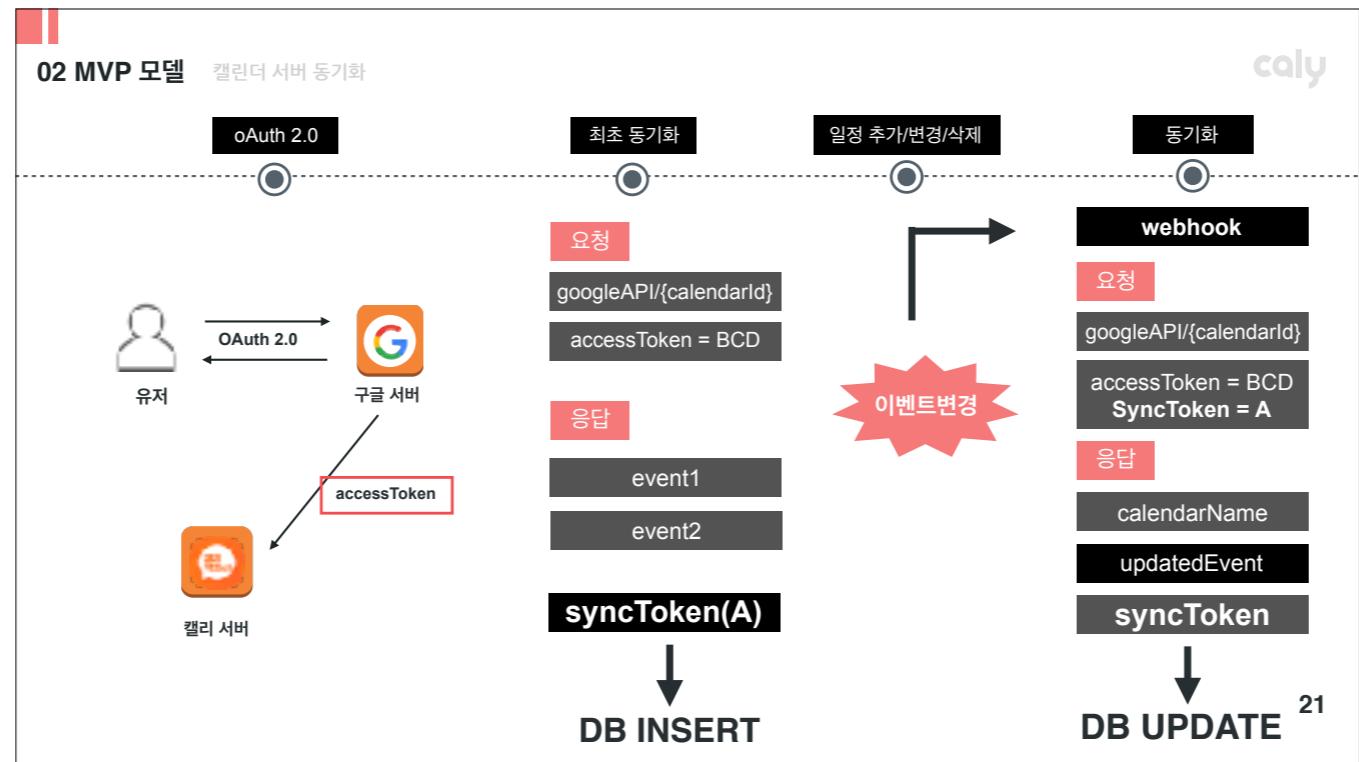


xml propfind http basic auth



20

CalDAV 캘린더 서버와 캘리는 Basic Auth 방식으로 통신을 하구요, 밑의 그림과 같이 cTag가 변경된 경우 추가/변경/삭제 작업을 진행합니다.



구글의 경우 API로 제공하며, OAuth 2.0을 통해 access token을 발급받아 접근 권한을 가져오게 되고, sync token을 기반으로 일정 정보를 갱신할 수 있습니다.

02 MVP 모델

추천 모듈

caly

시간/장소/목적 분석

캘린더 서버 동기화 후
유저 일정의 목적, 시간,
장소 정보 추출 및 분석

추천정보 수집

유저들에게
추천할 식당/카페/놀거리
가게 정보를 수집

추천 자동화

수집된 추천정보를
분석된 패턴을 통해서
랭킹/필터링을 통해 추천 제공

“ 어떤 데이터를 제공해야 할까? ”

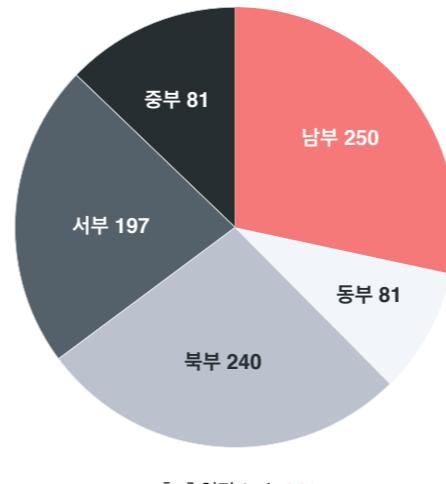
22

이젠 추천 정보 수집을 해야하는데요, 이 모듈에서의 관건은 어떤 데이터를 제공할까?입니다.

서울권 추천장소 DB구축

주타겟층 : 20대 학생층

- ① 대학교 지역과 핫플레이스(강남 등)를 중점으로 기본 데이터 수집
- ② 들어오는 일정의 장소가 없는 경우, 즉각적으로 수집 후 바로 추천



23

이에 대해서는 주타겟층인 20대 학생층의 주 활동지역을 기반으로 추천장소 DB구축을 하였는데,
먼저 대학교 지역 및 핫플레이스를 중점으로 수집하고, 들어오는 일정에 해당하는 장소가 없을 때 즉각적으로 수집 해 바로 추천했구요, 현재 총 추천장소 수는 900여개 정도 됩니다.

핵심기능이 추천서비스인데
추천을 할 수가 없다?

직접 추천 하자!

저희는 현재 유저가 언제 어떤 목적으로 이용하는지, 그 패턴 정보가 없습니다. 그러면 추천을 할 수 없을까요?
아뇨, 저희는 직접 추천하기로 했습니다. 다소 엉뚱해보이는 이 전략은 실제 유사 사례도 많습니다.

02 MVP 모델 유사 서비스

caly



명함관리서비스
리멤버



세무비서서비스
자비스

찍은 명함을
사람이 직접 입력

고객을
이해하는 것

찍은 영수증을
사람이 직접 입력

충분하게 데이터 쌓아두면
명함이미지 인식시스템 구축
일단 서비스를 시작해서
데이터 수집 후,
자체 인식 자동화 시스템 구축

25

명함관리 서비스인 리멤버는 카메라로 명함을 찍으면 그 정보를 자동으로 저장해주는 서비스입니다. 초기에는 사람이 직접 입력을 하고, 추후 데이터가 쌓이며 자동인식시스템을 도입합니다. 또 다른 예인 세무비서서비스 자비스도 마찬가지입니다.

이 두 서비스의 흐름이 그냥 빨리 런칭하는 것이 목표였을까요? 아닙니다. 일단 서비스를 시작해서 고객을 이해하고, 자체 인식 자동화 시스템을 구축하는 것입니다. 저희도 이와같이 최대한 빠르게 실고객에 접근하는 것이 목표였죠.



26

해당 시연 영상은 수동 추천을 통해 유저가 어떻게 추천을 받는지를 알 수 있는 영상입니다.

유저는 원하는 캘린더 서비스를 선택해 회원가입을 진행하는데요, 성별과 나이 정보와 함께 캘리의 이용 정책에 동의 후 회원가입을 진행합니다.

다음과 같이 동기화가 끝날때까지 화면이 멈춰있다가, 추천 분석 중인 상태가 되면 대기하라는 메시지를 받게 되죠. 이렇게 동기화가 끝나면 저희는 슬랙봇을 통해 즉각적으로 알 수 있게 되고 웹 관리자 추천 페이지를 통해서 유저를 확인할 수 있습니다. 해당 유저를 선택하면, 그 유저의 일정이 보이게 되고 그 일정에 어울리는 가게들을 선택해서 매핑하게 됩니다. 앞서 말씀드렸던 식당을 추천하고 있구요, 그 다음으로 카페, 마지막으로 놀거리를 추천합니다. 추가적인 매핑은 효율적인 데모를 위해 가속해서 보여드릴게요. 추천을 종료하게 되면, 남은 일정들은 추천하기 애매한 경우로 판단되어 비추천으로 분류됩니다.

왼쪽에서 유저는 추천 종료시 알람을 받게 되고 다음과 같이 여러 가게들을 추천받은 것을 보실 수 있습니다. 유저는 원하는 가게를 눌러서 그 가게를 실제로 이용한 블로거의 후기를 통해 어떤지 판단할 수 있구요, 그 위치를 다음과 같이 지도 뷰로 보실 수 있습니다.

실제 이 버전은 4월 중순에 런칭을 이미 했고, 그 사실은 플레이스토어에서 확인하실 수 있습니다.

02 MVP 모델 운영 경험 및 자동화의 필요성 캐치

버전	개선사항
1.0.0	베타 버전
1.0.1	통합 로그인
1.0.4	동기화 실패 재요청
1.0.5	앱 UI/UX 개선
1.0.6	지도 뷰 도입, 로그 분할 및 저장
1.0.7	유저 행동 로그 수집 개선
1.0.9	추천자동화 도입, 가격 항목 추가

릴리즈 노트



“30분만 기다려 달라고 했는데, 추천을 받는데 2시간이나 기다렸다.”



“메뉴들의 카테고리가 이전보단 다양해졌으나 여전히 추천되는 갯수가 부족한 것 같음. 총 갯수가 15 개가 아닌 레스토랑만이 10개정도 되면 좋겠음”

**빠른 수집자동화와
추천 자동화가 필요하다 !**

27

이 서비스를 운영하면서 다음과 같은 개선을 해왔구요,
다음과 같은 사용자 의견을 통해 빠른 자동화가 필요하다고 판단했습니다.

03 추천 자동화

추천 흐름

caly

시간/장소/목적 분석

캘린더 서버 동기화 후
유저 일정의 목적, 시간,
장소 정보 추출 및 분석

추천정보 수집

유저들에게
추천할 식당/카페/놀거리
가게 정보를 수집

추천 자동화

수집된 추천정보를
분석된 패턴을 통해서
랭킹/필터링을 통해 추천 제공

28

이제 남은 부분은 추천 자동화 파트인데요,
수동으로 추천하던 방식에서 어떻게 자동으로 추천을 만들어 나갔는지 설명드리겠습니다.

7시 성민이랑 데이트

목적	1 지인과 약속 2 데이트 3 각종 기념일 4 모임 뒷풀이 5 회의 및 데이트 6 문화생활
시간포맷	1 18시 2 7시 30분 3 7시 4 7시 반 5 오후 7시 6 저녁
장소단위	1 역이름 2 대학교명 3 대학교 명 출입문 4 대표 소재지 5 동 이름

```

1 + {
2   "event_hashkey": "----",
3   "event_types": [
4     {
5       "id": "CPI02"
6     }
7   ],
8   "time_set": [
9     "extract_start": "07:00",
10    "extract_end": "08:00",
11    "event_start": "10:00",
12    "event_end": "11:00"
13  ],
14  "locations": null
15 }
16

```

29

일정 정보에서 시간/장소/목적을 분석하는 부분입니다.

예를들어 '7시 성민이와 데이트'라는 정보가 들어온다면 '7시'라는 시간정보와 '데이트'라는 목적정보를 추출해내게 됩니다.

그렇게 추출하고나면 오른쪽과 같은 형식으로 데이터를 만듭니다.

보시는바와같이 이벤트 목적은 저희가 미리 매핑해놓은 데이트 id로 매핑되고, 시간은 추출한 시간 7시로 매핑되고, 장소는 추출할 수 없기에 null로 처리됩니다.

03 추천자동화 시간/장소/목적 분석

caly

장소	목적	방안
O	O	추천가능
O	X	추천가능
X	O	추천가능
X	X	추천불가

목적 상관없이 추천

주 활동지역으로 추천



▶ 주변역 추천

7시 성민이랑 데이트

- 1 강남역
- 2 신논현역(주변역)

30

장소와 목적을 중점으로 추천 여부를 결정하는 과정을 보여드리겠습니다.

장소와 목적이 없을 때 추천이 불가능해보이지만 다음과 같이 추천을 하기로 하였구요,

더 넓은 선택지를 위해 주변역 추천 기능도 추가했습니다.

앞의 예시에서는 장소 정보가 없으므로 세번째 경우인 과거기록을 기반으로 주활동지역을 지정하고, 그 주변지역인 신논현역이 추가됩니다.

- 01 장소 약속장소에서 가까운 장소가 추천되어야한다.
- 02 목적 데이트면 데이트 목적에 맞는, 스터디면 스터디 목적에 맞는 장소가 추천되어야한다.
- 03 유저 취향 유저의 반응을 기반으로 더 좋아하는것을 우선적으로 추천해야한다.
- 04 가격 / 거리 가격이 낮을수록, 역에서 거리가 가까울수록 우선적으로 추천되어야한다.

어느 장소들을 추천해줄지 알겠으니 그 장소들 안에서 어떤 순서로 보여줄지, 즉 랭킹이 관건인데요
저희들은 랭킹에 사용될 4가지 지표들을 뽑아 먼저 고려되어야 할 것부터 우선순위를 장소/목적/유저취향/가격-거리 순으로 정해서 계산했습니다.

03 추천자동화 추천 알고리즘

점수 우선순위

1 장소 2 목적 3 개인화 4 가격, 거리

$$Score = (10 - 장소순위) * 10^5 + \sum (\목적지수 - 1) * 10^4 + 개인화점수 * 10^3 + (a * 가격등급 + b * 거리등급) * 10^2$$

= 943,712

= 843,563

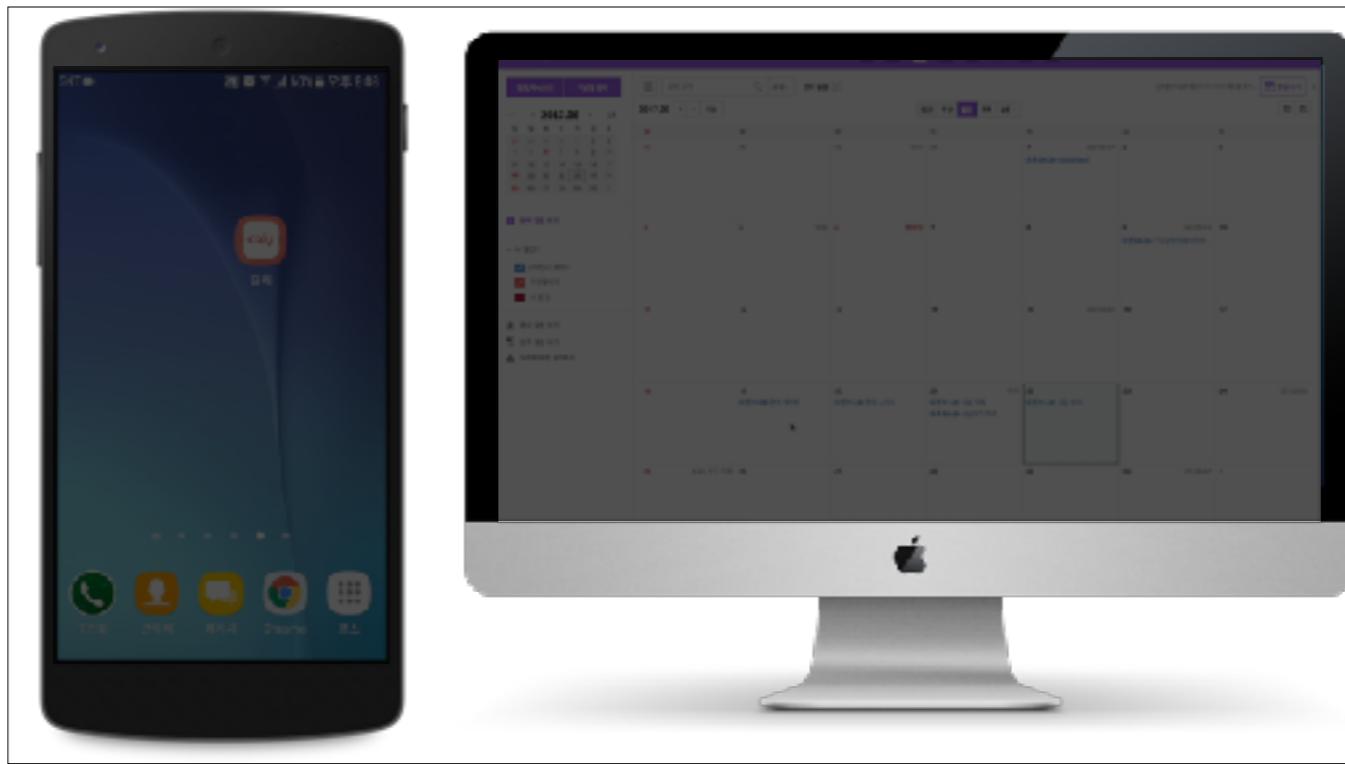
32

정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

앞서 보여드렸던 '7시 성민이랑 데이트'란 일정에 대해선

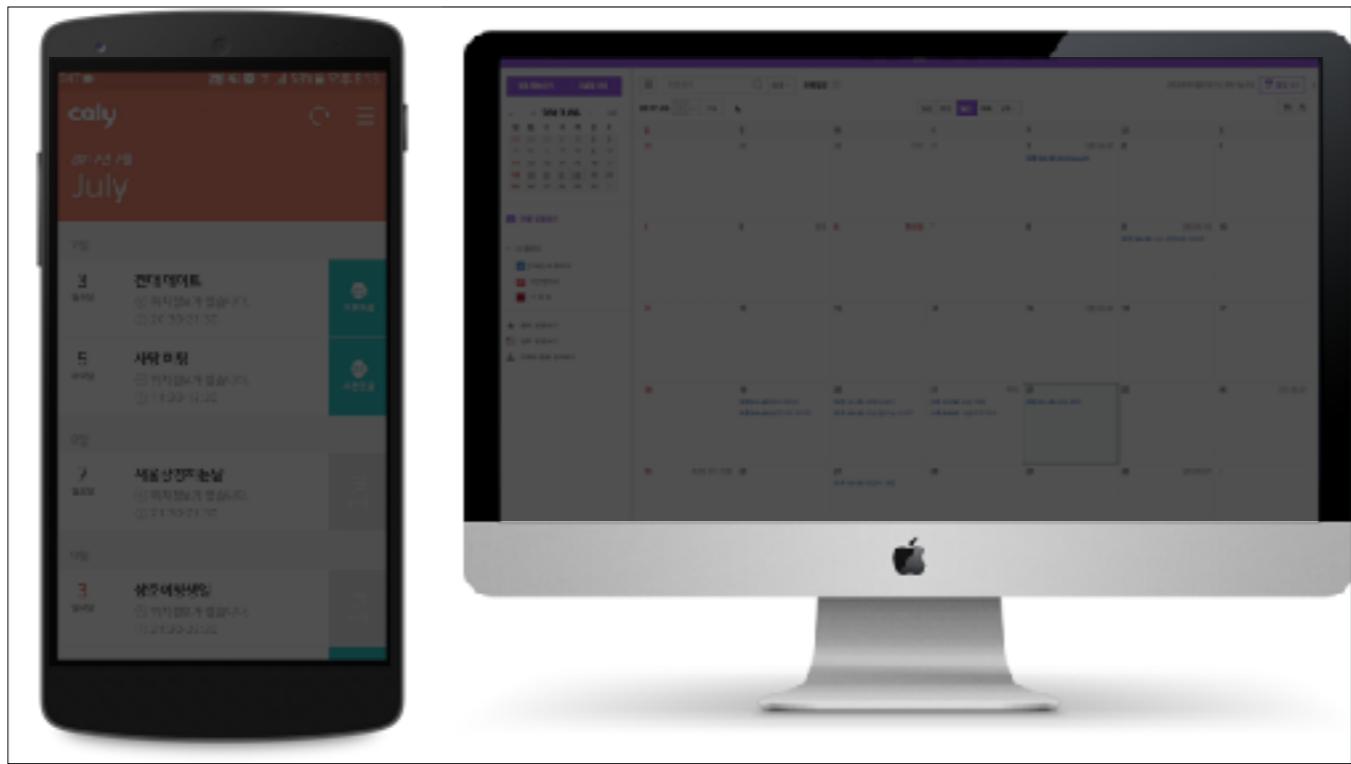
추출한 '강남역'에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.



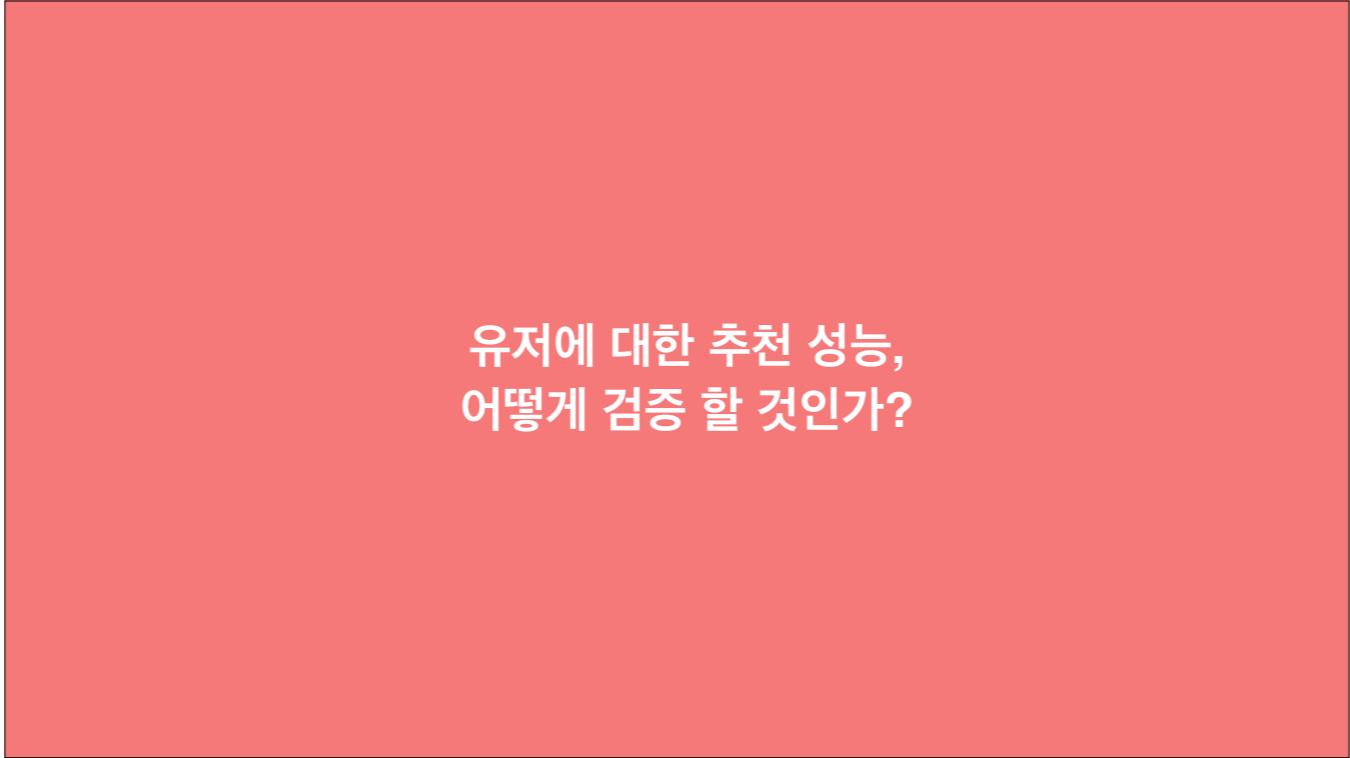
이 과정들이 실제로 어떻게 돌아가는지 보여드리겠습니다.

~~시연



이 과정들이 실제론 어떻게 돌아가는지 보여드리겠습니다.

~~시연



유저에 대한 추천 성능,
어떻게 검증 할 것인가?

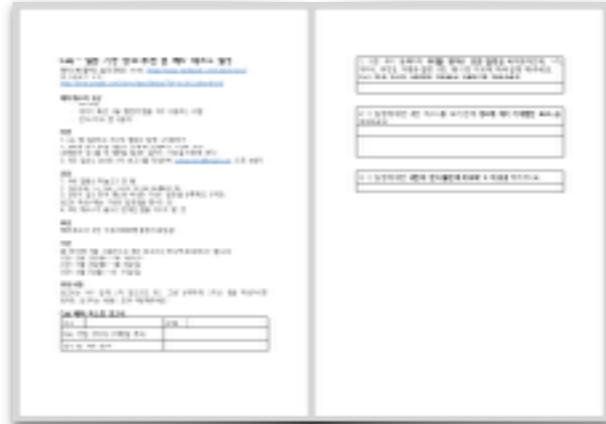
이렇게 잘 만든 추천에 대한 성능을
'어떻게 검증 할 것인가'가 매우 중요합니다.

03 추천자동화 베타테스터



번호	성별	나이	주소
1	여성	20	서울특별시 강남구 테헤란로 123
2	여성	30	경기도 용인시 기흥구 신갈동 456
3	여성	40	부산광역시 해운대구 해운대로 123
4	여성	50	제주특별자치도 제주시 한라로 123
5	여성	60	충청남도 천안시 동남구 청암로 123
6	여성	70	전라남도 목포시 목포로 123
7	여성	80	경상남도 창원시 마산합포구 경상로 123
8	여성	90	경상북도 경주시 경주로 123
9	여성	100	제주특별자치도 서귀포시 대정읍 대정로 123
10	여성	110	경상북도 경주시 경주로 123
11	여성	120	경상북도 경주시 경주로 123
12	여성	130	경상북도 경주시 경주로 123
13	여성	140	경상북도 경주시 경주로 123
14	여성	150	경상북도 경주시 경주로 123
15	여성	160	경상북도 경주시 경주로 123
16	여성	170	경상북도 경주시 경주로 123
17	여성	180	경상북도 경주시 경주로 123
18	여성	190	경상북도 경주시 경주로 123
19	여성	200	경상북도 경주시 경주로 123
20	여성	210	경상북도 경주시 경주로 123

20명의 베타테스터 고용



The left screenshot shows a survey page with a title "추천 기준 만족도를 통해 추천 했던 사람" and several questions about satisfaction levels (e.g., 1. 매우 만족, 2. 만족, 3. 보통, 4. 불만족, 5. 매우 불만족) for various categories like "친구", "가족", "동료", etc.

The right screenshot shows another survey page with a title "추천 기준 만족도를 통해 추천 했던 사람" and a section titled "추천 기준 만족도를 통해 추천 했던 사람" containing a table with columns for "선택한 친구", "선택한 가족", "선택한 동료", and "선택한 친구와 가족과 동료 모두".

매주 추천피드백 만족도 수집

36

이에 저희는,
저희의 타겟층과 일치하는 20명의 베타테스터를 고용해 어플리케이션을 사용하게 하고 매주 추천에 대한 피드백을 받고 있습니다.

(여기는 빠져도되고 넣어도될것같아요. 길게 설명할거 아니면 단점으로 보일 수 있음)

어플리케이션을 사용하는 실제 유저의 데이터보다는 부정확 하다고 볼 수 있겠지만,
현재 유저풀도 없고, 베타테스터에게 1:1로 만족도를 받고 있기에
유저들의 데이터를 분석하는것보다 정확도가 높다고 생각합니다.

꾸준히 운용하다보니, 추천 피드백이 수동추천과 자동추천 전후로 확연히 달라졌습니다.

03 추천자동화 베타테스터

caly

수동 추천

“(30분만 기다려 달라고 했는데, 추천을 받는데 2시간이나 기다렸다.”

“메뉴들의 카테고리가 이전보단 다양해졌으나 여전히 추천되는 갯수가 부족한 것 같음. 총 갯수가 15 개가 아닌 레스토랑만이 10개정도 되면 좋겠음”

“원래 종종 가던 아비꼬카레가 나와서 제법 괜찮은 신뢰도를 보여준다고 생각하게됨. 만족.”

자동 추천

“일정 등록하고 나서 추천되는게 예전보다 훨씬 빨라졌다. 추천이 온다.”

“예전엔 건대 악속에서 식당 6개를 추천 받았었는데, 업데이트 후에는 많아져서 선택의 폭이 넓어졌다.”

“따뜻한 국물이 있는 라멘집을 추천해줘서 만족스러웠다. 다만 차슈에서 조금 냄새가 났는데 그 또한 캘리와 연결된 후기에서도 그런 내용을 봐서 추천 내용에 대한 신뢰감이 들었다.”

37

(순서를 좀 변경했습니다.)

수동추천일때 가장 많았던 오래걸린다는 피드백은 빨라졌다고 바뀌었으며

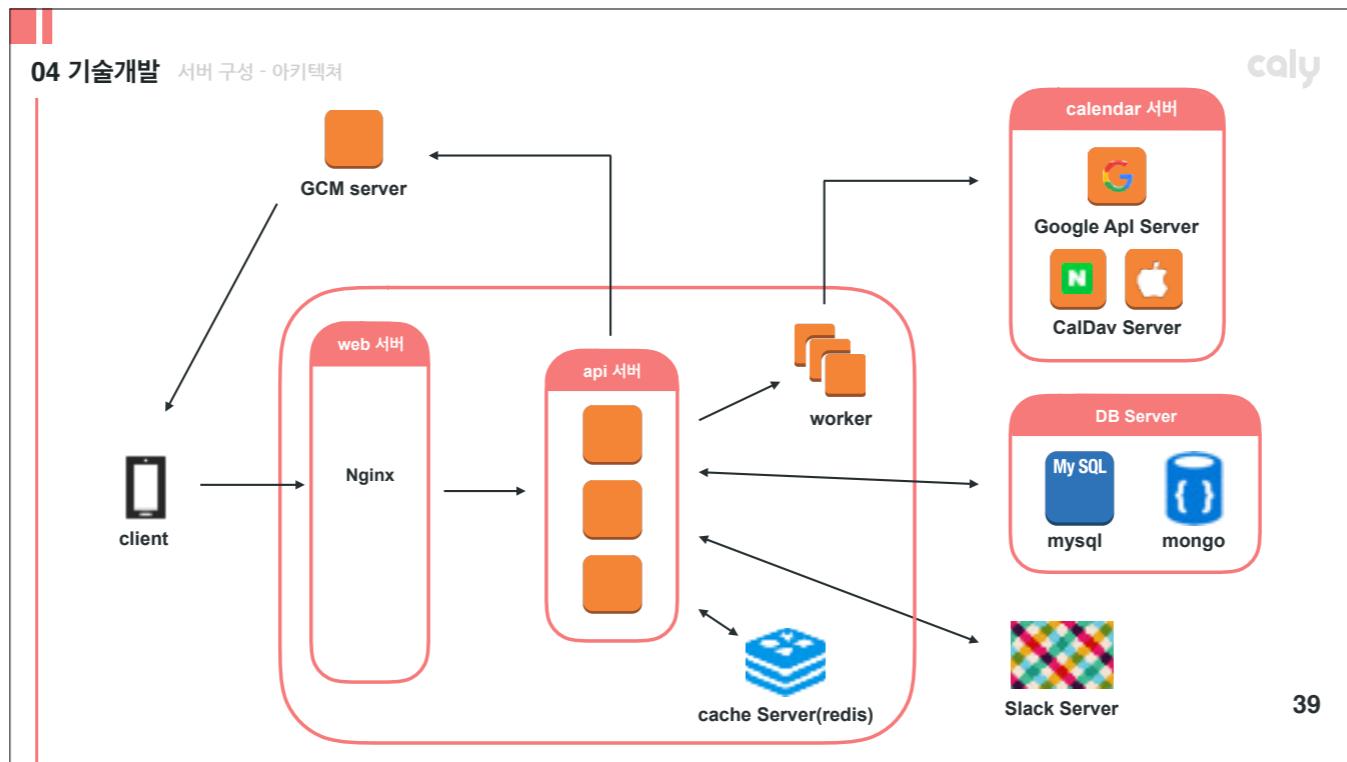
카테고리가 부족한 경우엔 선택의 폭이 넓어졌다고 바뀌었습니다.

이렇게 자동 추천으로 개선하면서도 신뢰도에 대한 피드백은 고스란히 유지되었습니다.

이는 유저가 어떤걸 좋아할지 깊게 고민한 덕분인것 같습니다.

기술개발

1. 서버구성
2. 동기화
3. CI/CD

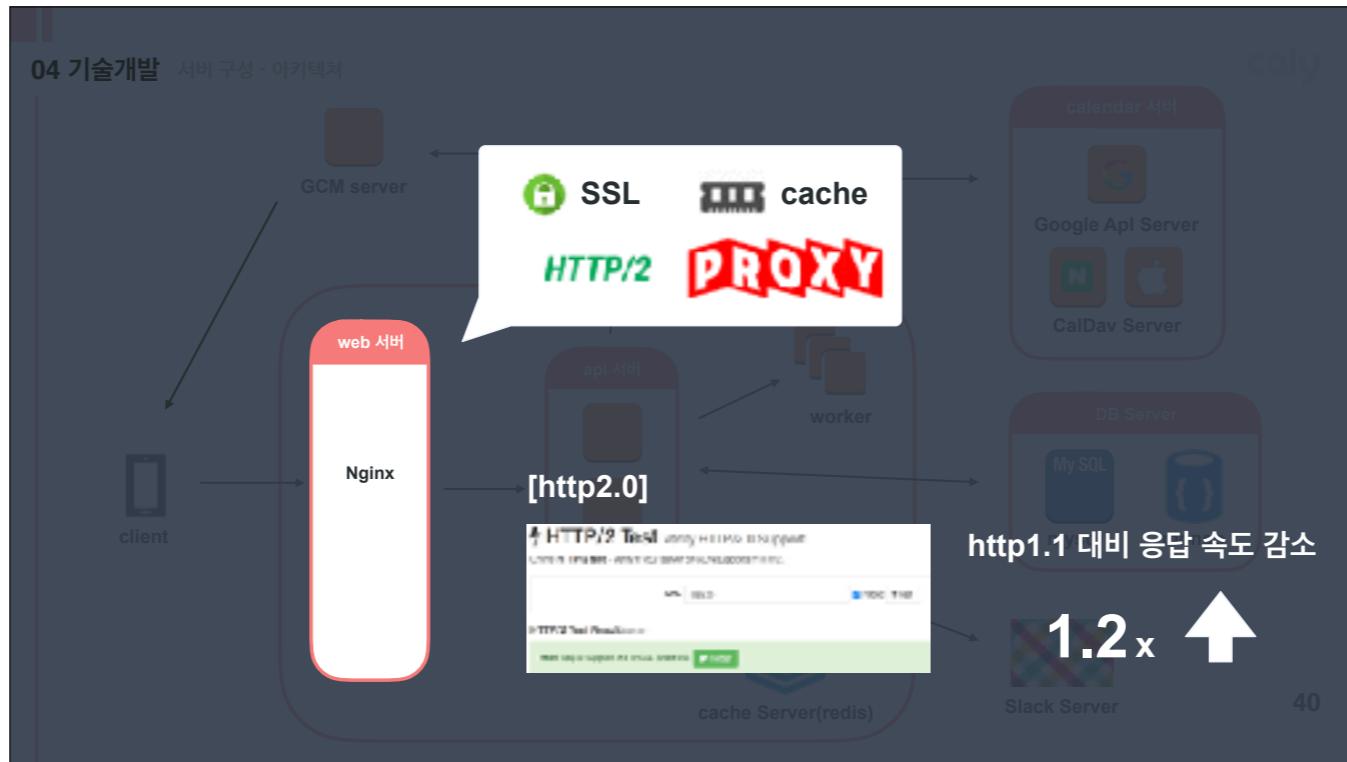


39

서버 아키텍쳐의 전반적인 모습입니다.

모든 요청을 앞단에서 받는 nginx서버, 실제 요청을 처리하는 api서버 그리고 캐리더 동기화 처리를 위한 워커들로 이루어져있습니다.

디비는 mysql을 사용하고 로그관리로 몽고디비를 사용하고있습니다.



nginx에서는 기본적인 보안을 위한 https 설정이 되어 있고,,

이미지 캐싱을 제공합니다. 실제 캐싱 적용후 이미지가 보다 빠르게 로드됨을 확인할 수 있었습니다.

http2가 붙어있는데요.

저희가 서비스를 실제 운영해보니, 저희가 의도하지 않은 네트워크 불안정 상태가 발생하는 경우가 있었습니다.

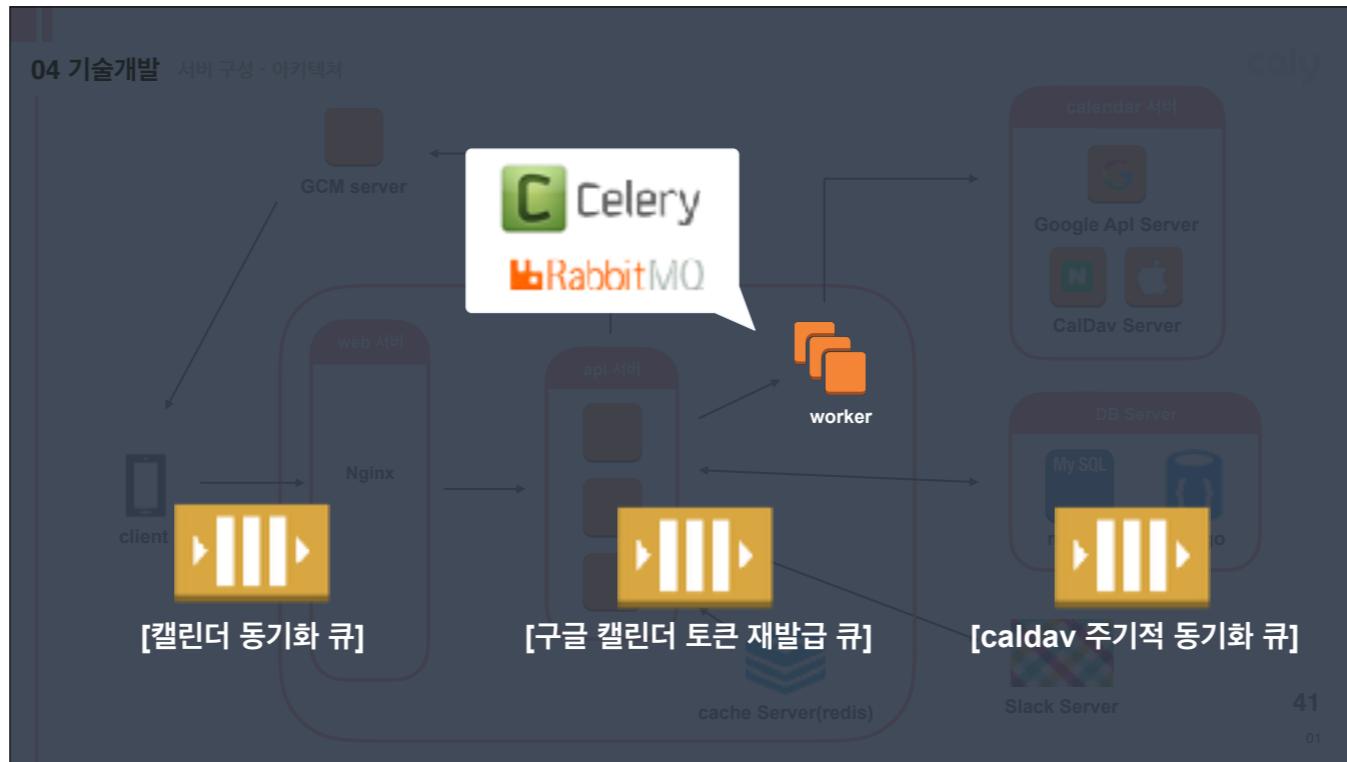
그때문에 latency가 발생하였고, 이를 절대적으로 줄이는 것은 불가능하므로 요청자체의 응답 속도를 낮춰야만 했습니다.

그래서 아래와 같이 [caly.io](#)에 http 2.0을 붙였고, 실제로 http1.1대비 평균 1.2배 반응속도가 좋아졌습니다.

마지막으로 reverse proxy입니다.

nginx에서 모든 요청을 받음으로써, loadbalancer 역할을 소프트웨어적으로 해결할 수 있습니다.

요청이 급작스럽게 불어나는 경우가 생기더라도 확장성있게 설계하였음으로, 문제없이 대응할 수 있을것입니다.



**컨텐츠 체크

다음은 큐-워커입니다.

최초 과거 일정 동기화 작업, 그리고 동기화 완료 후 5분마다 캘린더 동기화를 맞춰줘야하는 작업 등. 시간과 리소스가 많이 들어가는 작업들이 존재합니다.

시간이 많이드는 작업들을 하나의 요청이 잡고있어야할까요?

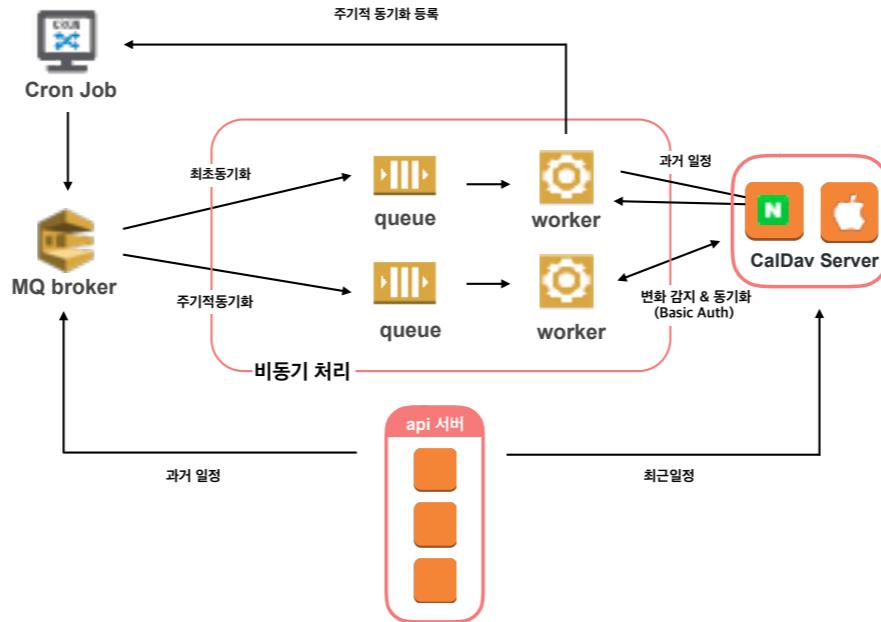
아닙니다. 말씀드린 작업들은 실제 5초 이상의 시간이 걸리는 작업들이고 요청이 많아지면 서버는 다 받아내지못할것입니다.

그렇기에 저희는 비동기로 해당요청을 처리하기로 하였고, rabbitmq와 celery로 큐 워커를 시스템 구성하였습니다.

아래 그림과 같이 각각의 비동기 업무에 따라 실제로 다양한 큐를 사용하고 있습니다.

04 기술개발 동기화 - CalDAV

caly

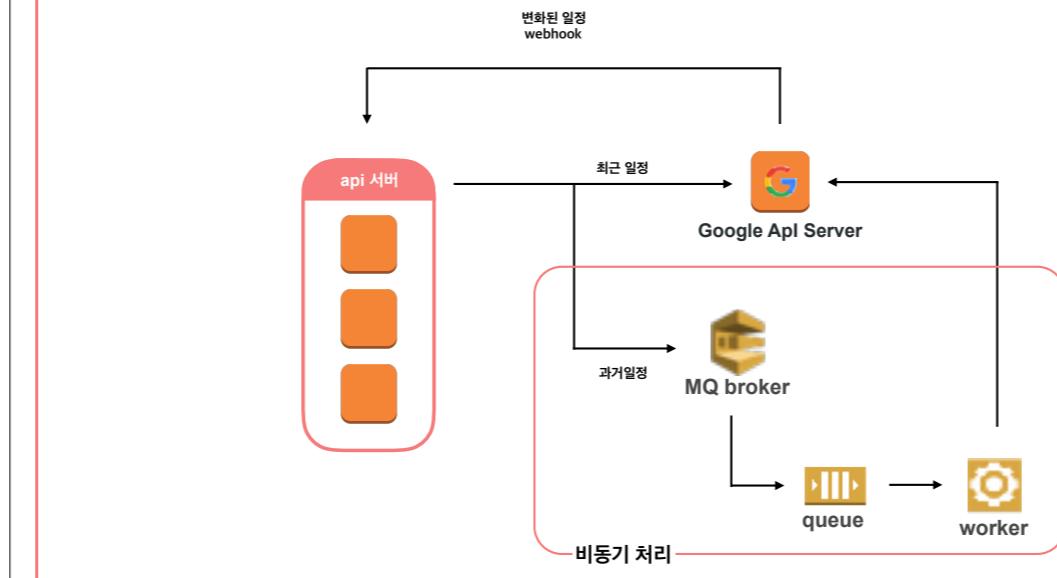


42

caldav동기화 상세 내용입니다.

위에서 말씀드린것과 같이 시간이 많이드는 최초 과거일정 동기화는 messageQueue브로커를 통해 큐에 각각 데이터를 할당하게됩니다.
큐에서 나온 데이터를 워커에서 받게되고 워커에서는 네이버,애플등 캘린더 서버와 통신을해 database에 값을 저장하게됩니다.

5분마다 추가 및 변경된 캘린더를 체크하는 과정도 동일합니다.

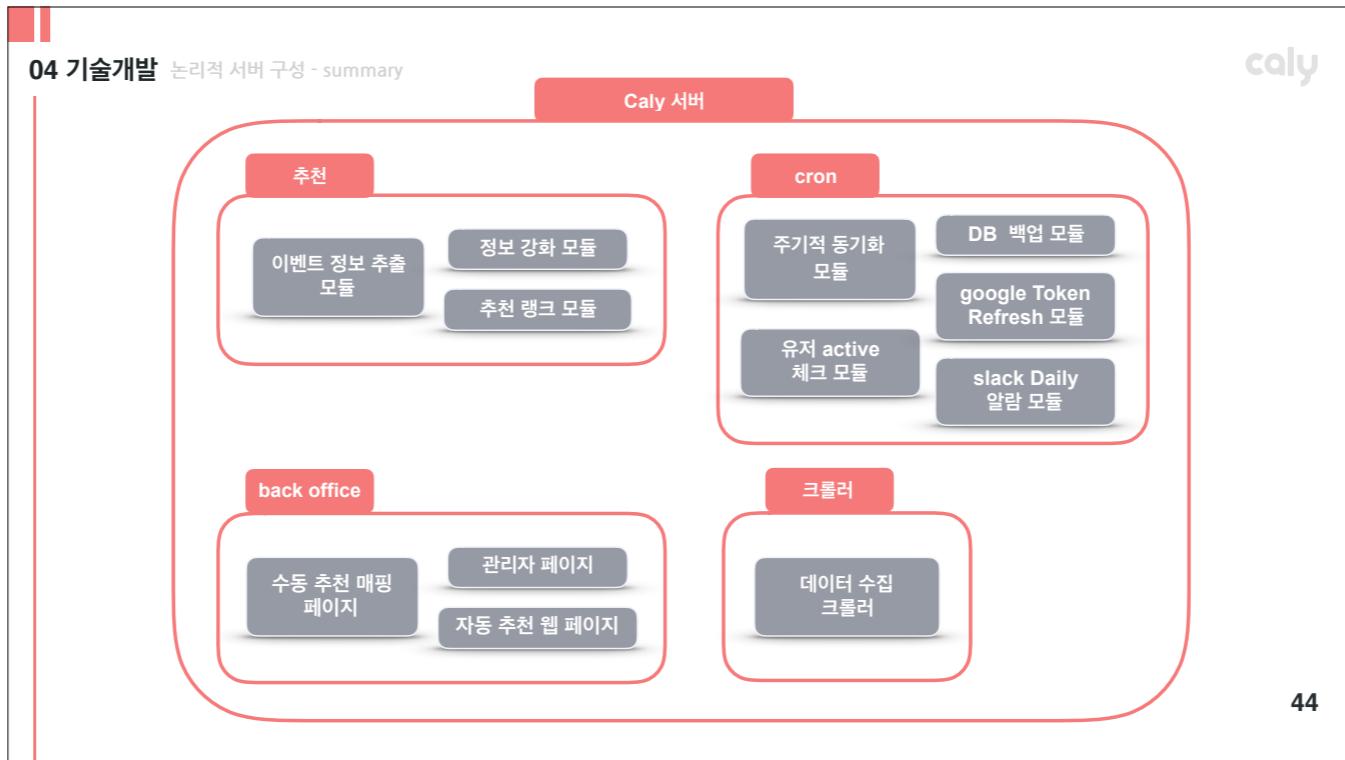


43

다음으로 googleApi동기화 상세 내용입니다.

구글 역시 과거일정을 가져오는데 시간이 많이 소요됩니다.

caldav와 마찬가지로 messagequeue와 worker를 통해 해당 작업을 비동기로 처리하게됩니다.



다음은 캘리서버의 논리구조입니다.

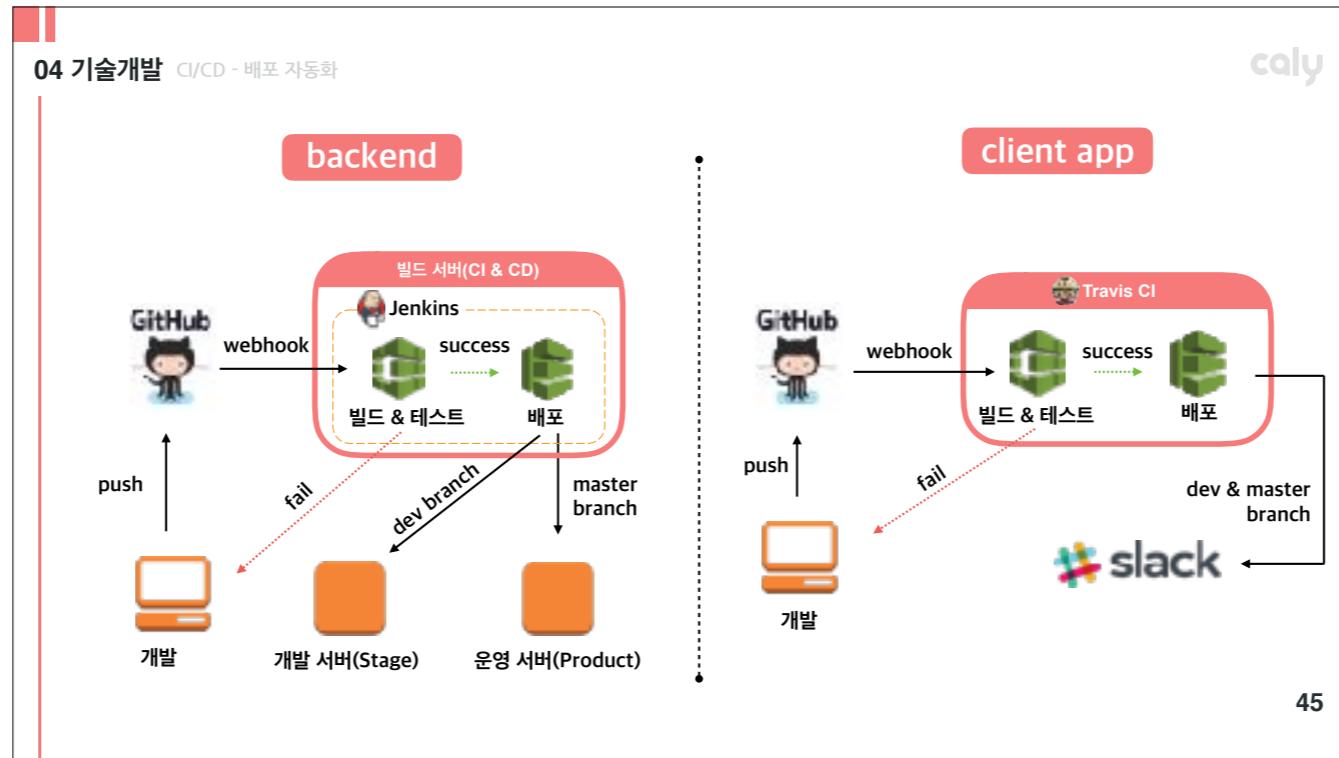
추천에는 이벤트 정보추출, 정보 보강, 추천랭크 모듈이 있습니다.

cron에는 5분마다 캘린더를 동기화하는 모듈, daily로 유저 active를 체크하는 모듈 등 주기적으로 작업해야할 모듈들이 있습니다.

backoffice에는 앞서 동영상에서 보셨던, 수동추천 매핑페이지, 관리자페이지 등이 있습니다.

마지막으로 추천 데이터를 반자동으로 수집할 수 있게 도와주는 데이터 수집 크롤러가 존재합니다.

caly는 이와같이 다양한 모듈들이 긴밀히 연결되어 실제로 아주 부드럽게 워킹하고 있습니다.



45

ci/cd 배포 자동화에 대한 이야기입니다.

먼저 백엔드 서버 배포에 대해 설명드리겠습니다.

백엔드 서버 배포할 것이 생겨 깃에 푸시를 하게되면, 웹훅을 통해 jenkins가 새로운 배포버전이 올라왔다는것을 알게됩니다.

이후 jenkins에서는 새로운 환경에서 빌드와 테스트를 하고 실제 빌드가 성공적이라면 실제 개발 혹은 운영서버에 배포하게됩니다.

실패하게되면 당연히 배포하지 않겠죠.

이와같이 배포 자동화를 거치면서 실제 일일이 손으로 받아 내부 빌드 테스트를 했것보다 10배정도 빠르게 빌드 & 테스트를 처리할 수 있게되었고, 자동화를 통한 생산성 증대에 대한 중요성을 깨닫게 되었습니다.

추가 기술성과

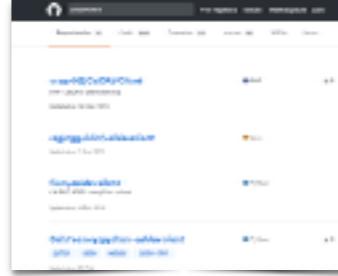
1. 기술블로그

05 추가 기술성과 CalDAV 오픈소스

caly



RFC-4791 CalDav
표준문서 준수



Python CalDav Client
오픈소스화

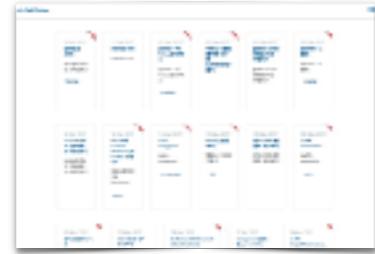


이슈 등록 및 특정 기능 개선 문의

47

저희는 최초에 caldavclient 오픈소스를 찾아보았습니다. 하지만 제대로된 caldavclient 소스는 없었으며, 네이버, 다음, 애플 등을 지원하는 라이브러역시 존재하지 않았습니다. 그래서 저희가 표준문서를 보고 caldavclient를 만들 수 밖에 없었습니다.

이렇게 caldavclient를 만들고보니 실제 필요로하는 사람으로부터 이슈등록과 특정 기능 개선 문의를 받을 수 있었습니다.



Calyfactory 블로그
23주차 58포스팅



구글 최상단 검색



네이버 API 포럼
캘린더 담당자 인용

또한 실제 개발을 하다보니, 저희와 같이 고민을 하는 수많은 개발자들을 위해, 캘린더 개발 생태계를 위해 개발 지식 노하우를 공유하는것 또한 매우 의미가 있을거나는 생각이들어 23주차 58개의 포스팅을 진행했습니다.

현재 구글 검색시 최상단에 나옵니다.

실제 네이버 api포럼에 api담당자가 저희 블로그 글을 인용하여 답변을 달았던것을 확인할 수 있었습니다.

네이버에서 인용할 만큼 저희 개발 내용이 안정적이고, 잘 만들어졌다는것을 검증받은 기분이들어 굉장히 뿌듯했습니다.

서비스 운영

서비스를 어떻게 운영했나?

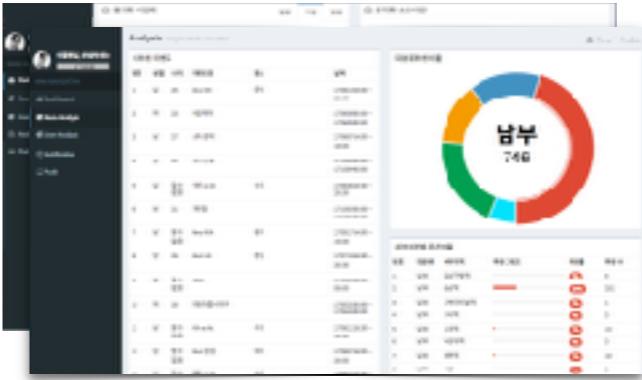
1. Back Office
2. ChatBot 알림
3. 마케팅
4. 성과

06 서비스 운영 Back Office

caly

The screenshot displays a complex dashboard titled "CalyFactory". It includes sections for "System Status", "Logs", "Logs (Detailed)", and "Logs (Summary)". The "Logs" section shows a list of log entries with columns for timestamp, level, and message. There are also sections for "Process Monitoring" and "Process Log". A sidebar on the left lists users with their roles and contact information.

관리자 매핑웹



모니터링 대시보드

50

출시만 하고 끝낸것이 아니라 꾸준히 모니터링을 했습니다.

앞서 보여드렸던 수동추천에서 사용된 관리자 매핑웹을 만들어 수동으로 매핑해줬고,

어떤 유저가있는지, 어떤 일정을 추가하는지, 어떤 행동을하는지, 계속 관찰하기 위해 모니터링 대시보드를 만들어 항상 개선을 위해 노력하고 있습니다.

06 서비스 운영 Chat Bot 알림



51

능동적으로 관리자페이지에서 보는것 뿐만 아니라 즉각적으로 반응해야하는것이나 알림을 받아야하는것들은
챗봇을 만들어 자동화하였는데요

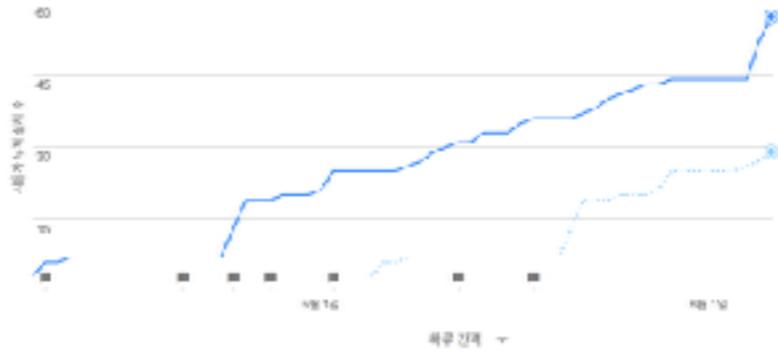
운영에 필요한 알림을 주는 알림봇, 배포의 상태를 알려주는 배포봇, 에러를 대응하게 도와주는 모니터링봇, 이슈변경사항을 알려주는 관리봇 등이 있습니다.

다양한 봇들을 만들어 자동화를 하였는데요, 이 과정에서 봇이 1인분의 역할을 대신 해 줬다고 생각합니다.
이러한 자동화 과정을 통해 세세한 것에 집중하지 않고 중요한 이슈에 집중 할 수 있었습니다.

06 서비스 운영 마케팅 성과

caly

사용자 누적 설치 수



112

가입 유저수



380

연결된 캘린더 수



15,734

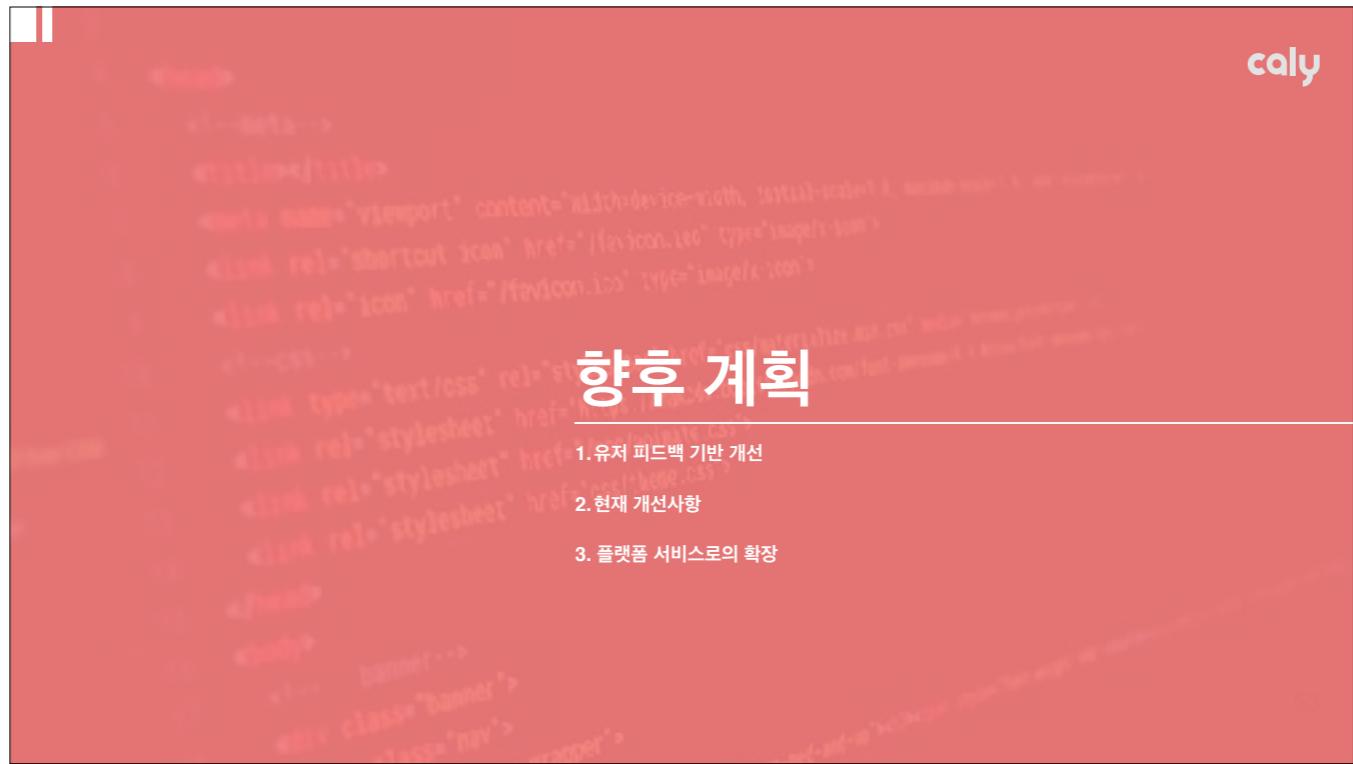
동기화된 일정 수

52

(수치는 최신 데이터로 바꿨습니다.)

(마케팅 페이지가 빠지니까 전페이지에서 넘어오는게 좀 어색하네요)

4월 런칭 후부터 꾸준히 운영을 하면서 현재는 112명의 유저, 380개의 캘린더, (강조하면서) 만오천개 이상의 일정데이터를 모을 수 있었습니다.
마케팅과 홍보를 통해 유저수가 꾸준히 증가하고 있고, 앞으로 기능 고도화를 통해 더욱더 성장한 모습을 볼 수 있을거라고 생각합니다.



타이틀은 조금 더 고민해볼것

07 향후 계획 유저피드백 기반 개선



실제 유저 피드백

① 기능적인 피드백

② 추천 성능에 대한 피드백

③ 디자인 UI/UX등의 피드백

54

실제 유저 피드백입니다. 이는 세 가지인 기능, 추천성능, 디자인으로 분류할 수 있습니다.

07 향후 계획

현재 개선사항

caly

1. 이미 진행한 아이템



“약속장소 주변 장소들을 쉽게 보고싶어”

지도 기능

진행완료

2. 개선 진행 중



“더 많은 데이터를 추천해주세요”

크롤러 반 자동화

70% 진행

3. 우선 개선 포인트



“앱에서 일정을 바로 추가하고 싶어요”

일정 추가 기능

개발 예정

4. 추가 예정 기능



“다양한 카테고리를 만들어주세요”

추천 카테고리 다양화

기획 예정

55

그 피드백들을 우선순위를 정해서 진행 중인데요, 지도 기능은 이미 개발했고 크롤러 반자동화는 70% 정도 진행되었습니다. 일정 추가는 개발 예정, 추천 카테고리 다양화는 기획 중입니다. 저희가 이렇게 사용자의 의견을 바로 반영하는 이유는, 단순히 앱 기능 개선이 목표가 아닙니다.



caly

새로운 플랫폼 서비스

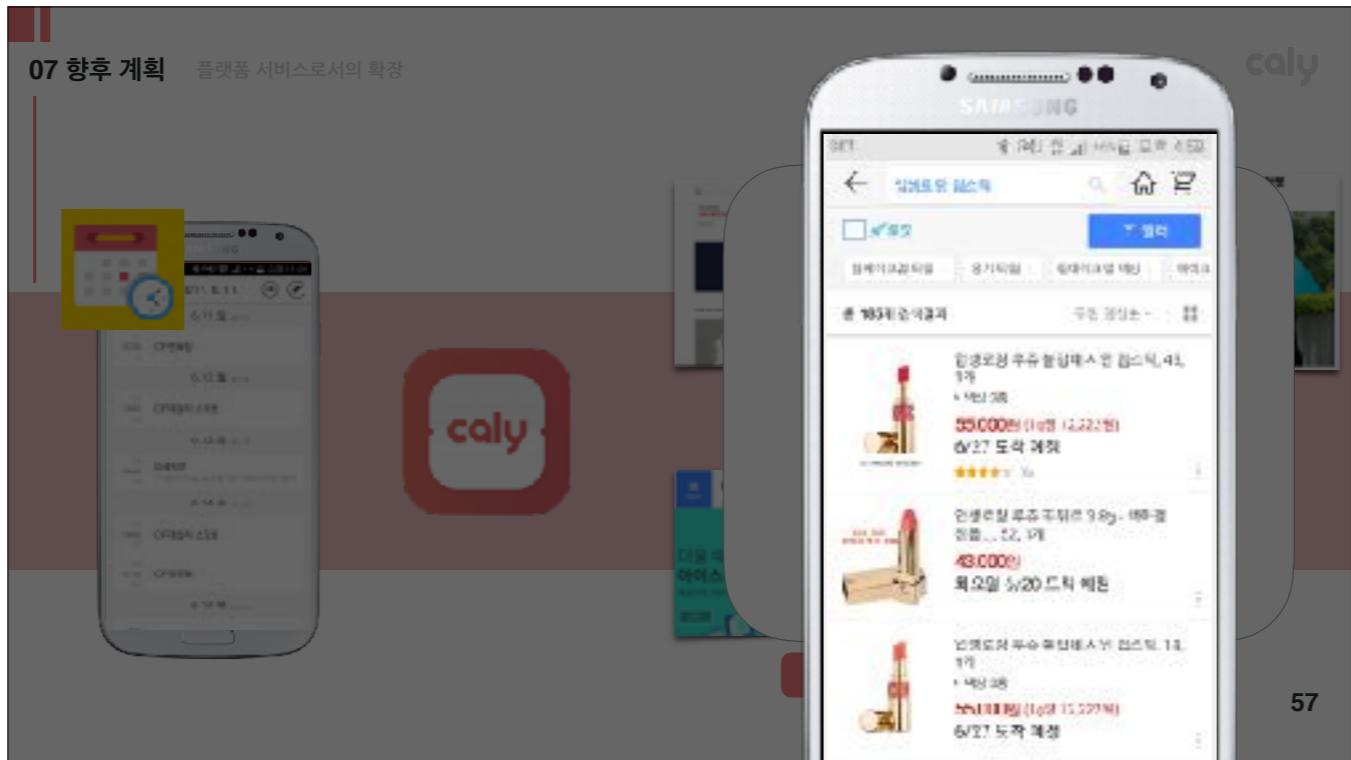


56

바로 새로운 플랫폼 서비스로서의 캘리를 위해서죠.

07 향후 계획

플랫폼 서비스로서의 확장



57

캘리는 사용자가 입력하는 일정과 목적을 연관지어 여러 서비스들을 연결하는 다리가 되는, 플랫폼 서비스로의 확장이 가능하다고 확신합니다.
그 예로 사람들은 기념일을 깜빡할 수도 있지만, 캘리가 추천과 함께 기념일을 행여줍니다. 그 추천을 클릭하게 되면 연령대에 어울리는 선물 구매까지 바로 연계됩니다.
이 기능과 같은 연계는 유저의 일정 기반으로 충분히 가능한 서비스이고, 이를 위해 또 열심히 개발 중에 있습니다.

프로젝트 관리

1. 예자일
2. 프로젝트 진행일정
3. 배운 점
4. 팀원소개



- ① 동기 부여
- ② 기술적 이슈 공유



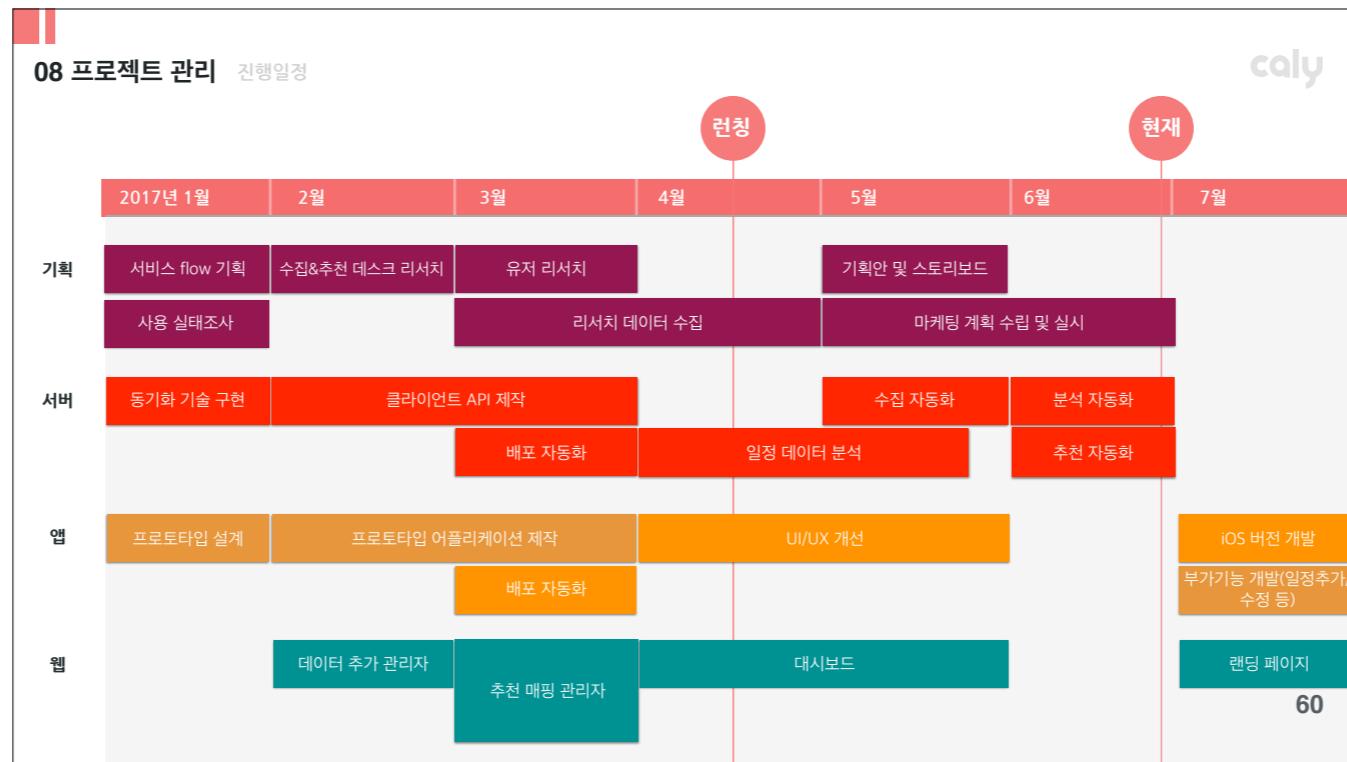
Agile

Daily Scrum

저희 프로젝트는 스프린트단위로 작업을 관리하는 에자일 방법론을 선택하고, 매일 아침 9 시 매일 전날의 이슈를 공유하는 데일리 스크럼을 진행했습니다. 데일리스크럼을 진행하면서, 투명한 작업내용으로 서로에게 동기부여가 되었고, 실제 개발 이슈들이 공유가 되어 기술적으로 디스커션하는 자리가 생겨 개발시 서비스에 알맞은 의사결정을하는데 크게 도움이 되었습니다.

08 프로젝트 관리 진행일정

caly



다음은 프로젝트 관리입니다.

저희는 이 그림처럼 매우 많은 업무를 해왔습니다. 굉장히 6개월이란 기간동안 이렇게 빽세게 진행해본것도 처음이고, 꾸준하게 진행할수 있을거란 생각도 하지 못었습니다. 하지만 저희는 실제로 이 모든것을 일정을 소화했고 보시는것처럼 4월에 런칭을 하여 꾸준한 업데이트를 통해 현재는 분석, 추천 자동화까지 진행되었습니다.

08 프로젝트 관리 앞으로의 진행일정

caly



61

하지만

저희의 일정은 여기서 끝나지 않습니다.

앞으로 실제 서비스를 더욱 디벨롭하기 창업 6개월 철린지 프로그램, 그리고 비즈니스모델을 위한 업체 광고수주, 추천 sdk개발 등 수 많은 일정을 다시 앞의 6개월처럼 완벽하게 달려나갈 예정입니다.

웹/서버



프로젝트 관리



CI/CD



앞서 일정을 소화하면서 배운것도 보시는 바와같이 이렇게나 많습니다.

실제 프로젝트하기전에는

웹 서버단에서는 테스트서버에 sql붙여 돌리기밖에 하지 못하였고, 프로젝트 관리나 CI/CD에 대해서는 거의 무지했습니다.

하지만

이번 프로젝트를 하면서 react,nginx,caldav등 수많은 웹/서버 지식을 쌓을 수 있었고,

프로젝트 관리를 위해 agile기반으로 jira,confluence툴을 써보니 일을 열심히하는게 중요한것이아니라 효율적으로 하는것이 더욱 중요하다는 것을 깨달았습니다.

CI/CD 역시 안정적인 백엔드 및 클라이언트 유지를 위해 기본적이며, 꼭 사용해야할 툴들이라는 것을 알았습니다.

비록 6개월동안 학생들끼리하는 소프트웨어마에스트로 과정이지만

저희가 얻은것은 학교에서 배우는 수준이 아닌 실무에서만 배울 수 있는 경험 그 이상이었다 라는 생각이 듭니다.



멤버소개



Mentee 최평강

Mentor

- 프로젝트 관리
- (현) GSShop
- (전) 데일리호텔
- (전) 요기요

Mentee 정성민

- Google OAuth2.0 이용한 GoogleCalendar API 등기화
- 모바일 API 서버 개발
- 수집자동화 웹크롤러 개발
- 일정보강 모듈 개발

Mentee 이종현

- CalDAV Calendar 등기화
- 추천데이터 등록 및 분석
- 안드로이드 앱
- NLP를 활용한 이벤트 분석
- 일정 추출 모듈 개발

Only I can change my Life
Only I can change my Life

안정적인 배포

효율적인 프로젝트 관리

도전 그리고 도전

caly

64

멤버소개

Mentee 정성민

Mentor

- 프로젝트
- 추천데이터 등록 웹페이지
- 안드로이드 앱
- 추천 모듈 개발

(현) GSShop
(전) 데일리호스피탈
(전) 요기요

Mentee 이종현

앞으로의 성장

- CalDAV Calendar 등기화
- 추천데이터 등록 웹페이지
- 안드로이드 앱
- 추천 모듈 개발

CalDAV Calendar 등기화
추천데이터 등록 웹페이지
NLP를 통한 이벤트 분석
일정 추출 모듈 개발

테스트의 중요성

프로세스 자동화

65
01

멤버소개



Mentee 이종현

Mentor

- 프로젝트
- 웹관리자 페이지
- NLP를 통한 이벤트 분석
- 일정 추출 모듈 개발

(현) GSShop
(전) 데일리
(전) 요기요

Only I can change my Life
No one can do it for me.

새로운 개발언어에 대한 도전

6개월 중 6개월

Only I can change my Life
No one can do it for me.

Mentee 정성민

인생최고 장기 프로젝트 매일갱신

- CalDAV Calendar 등기화
- 추천 알고리즘 구현
- 안드로이드 앱
- 추천 모듈 개발

Mentee 이종현

- CalDAV Calendar 등기화
- NLP를 통한 이벤트 분석
- 일정 추출 모듈 개발

66
01

caly

1	추천 자동화 과정	P.68~76
2	이미지 출처 표기	P.77
3	맛집 랭킹의 신뢰성	P.78
4	비즈니스 모델	P.79
5	테스트 주도 개발	P.80
6	마케팅	P.81
7	통합로그인	P.82

앞서 일정을 소화하면서 배운것도 보시는 바와같이 이렇게나 많습니다.

실제 프로젝트하기전에는

웹 서버단에서는 테스트서버에 sql붙여 돌리기밖에 하지 못하였고, 프로젝트 관리나 CI/CD에 대해서는 거의 무지했습니다.

하지만

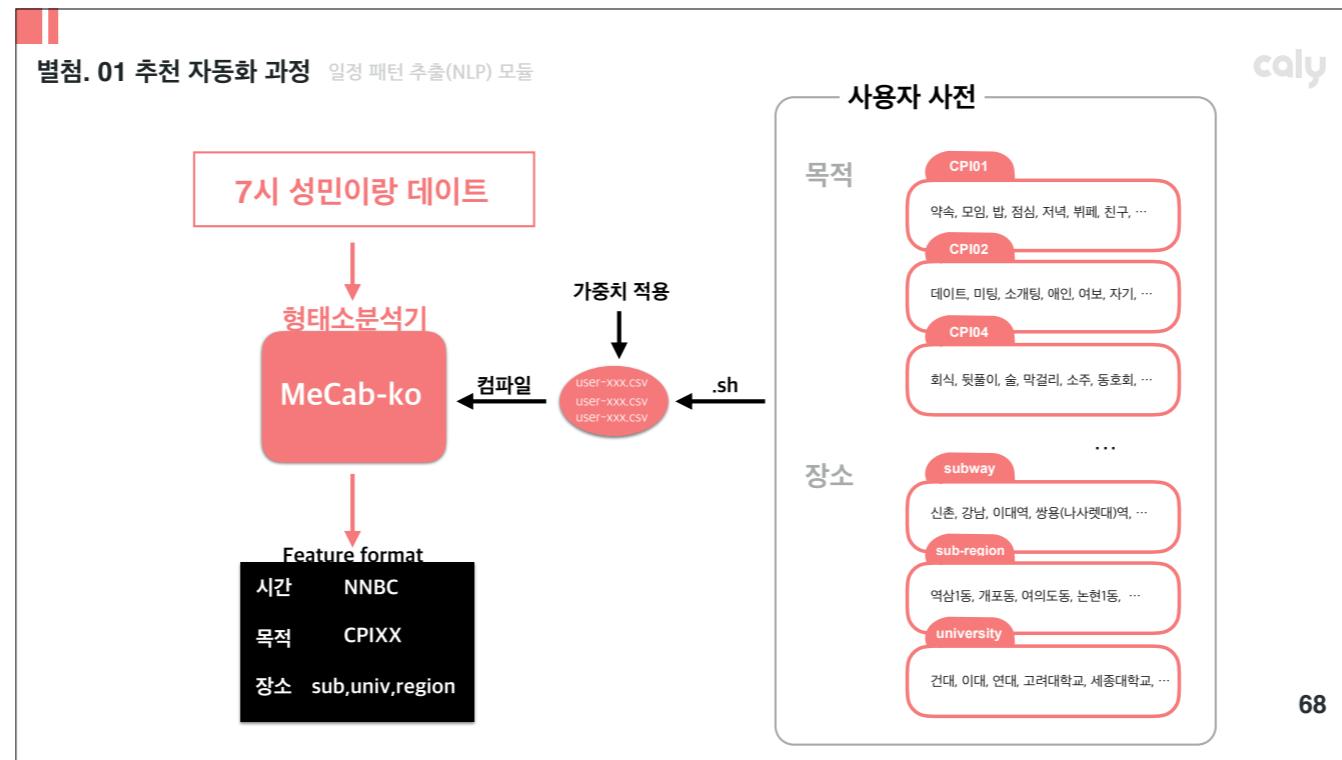
이번 프로젝트를 하면서 react,nginx,caldav등 수많은 웹/서버 지식을 쌓을 수 있었고,

프로젝트 관리를 위해 agile기반으로 jira,confluence툴을 써보니 일을 열심히하는게 중요한것이아니라 효율적으로 하는것이 더욱 중요하다는 것을 깨달았습니다.

CI/CD 역시 안정적인 백엔드 및 클라이언트 유지를 위해 기본적이며, 꼭 사용해야할 툴들이라는 것을 알았습니다.

비록 6개월동안 학생들끼리하는 소프트웨어마에스트로 과정이지만

저희가 얻은것은 학교에서 배우는 수준이 아닌 실무에서만 배울 수 있는 경험 그 이상이었다 라는 생각이 듭니다.



앞서 일정을 소화하면서 배운것도 보시는 바와같이 이렇게나 많습니다.

실제 프로젝트하기전에는

웹 서버단에서는 테스트서버에 sql붙여 돌리기밖에 하지 못하였고, 프로젝트 관리나 CI/CD에 대해서는 거의 무지했습니다.

하지만

이번 프로젝트를 하면서 react,nginx,caldav등 수많은 웹/서버 지식을 쌓을 수 있었고,

프로젝트 관리를 위해 agile기반으로 jira,confluence툴을 써보니 일을 열심히하는게 중요한것이아니라 효율적으로 하는것이 더욱 중요하다는 것을 깨달았습니다.

CI/CD 역시 안정적인 백엔드 및 클라이언트 유지를 위해 기본적이며, 꼭 사용해야할 툴들이라는 것을 알았습니다.

비록 6개월동안 학생들끼리하는 소프트웨어마에스트로 과정이지만

저희가 얻은것은 학교에서 배우는 수준이 아닌 실무에서만 배울 수 있는 경험 그 이상이었다 라는 생각이 듭니다.

별첨. 01 추첨자동화 랭크순위

점수 우선순위

1 장소 2 목적 3 개인화 4 가격, 거리

$$Score = (10 - 장소순위) * 10^5 + \sum_{i=1}^n (\text{목적지수} - 1) * 10^4 + 개인화점수 * 10^3 + (a * \text{가격등급} + b * \text{거리등급}) * 10^2$$

= 943,712

= 843,563

69

앞서 일정을 소화하면서 배운것도 보시는 바와같이 이렇게나 많습니다.

실제 프로젝트하기전에는

웹 서버단에서는 테스트서버에 sql붙여 돌리기밖에 하지 못하였고, 프로젝트 관리나 CI/CD에 대해서는 거의 무지했습니다.

하지만

이번 프로젝트를 하면서 react,nginx,caldav등 수많은 웹/서버 지식을 쌓을 수 있었고,

프로젝트 관리를 위해 agile기반으로 jira,confluence툴을 써보니 일을 열심히하는게 중요한것이아니라 효율적으로 하는것이 더욱 중요하다는 것을 깨달았습니다.

CI/CD 역시 안정적인 백엔드 및 클라이언트 유지를 위해 기본적이며, 꼭 사용해야할 툴들이라는 것을 알았습니다.

비록 6개월동안 학생들끼리하는 소프트웨어마에스트로 과정이지만

저희가 얻은것은 학교에서 배우는 수준이 아닌 실무에서만 배울 수 있는 경험 그 이상이었다 라는 생각이 듭니다.



앞서 일정을 소화하면서 배운것도 보시는 바와같이 이렇게나 많습니다.

실제 프로젝트하기전에는

웹 서버단에서는 테스트서버에 sql붙여 돌리기밖에 하지 못하였고, 프로젝트 관리나 CI/CD에 대해서는 거의 무지했습니다.

하지만

이번 프로젝트를 하면서 react,nginx,caldav등 수많은 웹/서버 지식을 쌓을 수 있었고,

프로젝트 관리를 위해 agile기반으로 jira,confluence툴을 써보니 일을 열심히하는게 중요한것이아니라 효율적으로 하는것이 더욱 중요하다는 것을 깨달았습니다.

CI/CD 역시 안정적인 백엔드 및 클라이언트 유지를 위해 기본적이며, 꼭 사용해야할 툴들이라는 것을 알았습니다.

비록 6개월동안 학생들끼리하는 소프트웨어마에스트로 과정이지만

저희가 얻은것은 학교에서 배우는 수준이 아닌 실무에서만 배울 수 있는 경험 그 이상이었다 라는 생각이 듭니다.

별첨. 01 추첨자동화 목적지수

7시 성민이랑 데이트

일정 예상 목적

1 데이트
2 NONE

→

‘호우초밥’ 목적지수

1 지인과의 약속 적합함
2 데이트 적합함
3 각종 기념일 적합함
4 모임 뒷풀이 적합함
5 회의 및 스터디 부적합함
6 문화생활 부적합함

적합함 부적합함

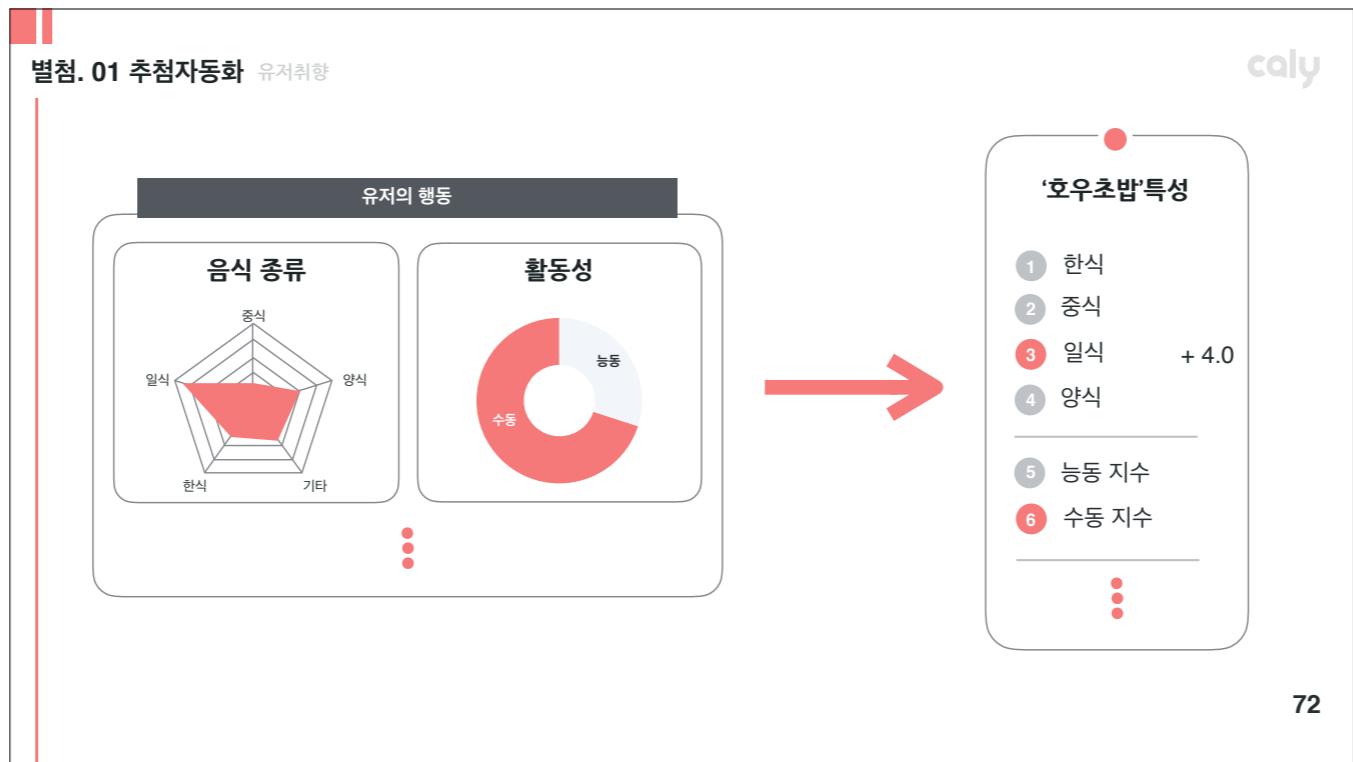
= ‘호우초밥’은 ‘데이트’에 적합함 => 점수부여

71

추천순위로 통일

부적합 적합이 잘 구분되게 색깔이나 아이콘 배경 등으로 수정해보기

포인트 : 극혐 화살표와 항목/특성 나열한 테이블이 이뻐지면 좋겠어요



유저의 행동을 어떻게 수집하는지 말로 설명

방사형 그래프로 한식/중식/일식 + 파이그래프로 능동/수동 + 로맨틱은 ...으로 표시 해서 수정

실제 등급 구간

등급	1	2	3	4	5	6	7	8	9
등급비율	4%	10%	22%	40%	59%	77%	89%	95%	100%
가격	1,000원	3,000원	4,000원	5,500원	9,000원	14,000원	21,000원	35,000원	140,000원
거리	1분	2분	3분	4분	6분	8분	11분	19분	30분

가격 등급 예시

500원
1,500원 → 1등급
6,000원
2등급
5등급

가격/거리 등급

별첨. 01 추천자동화 추천 알고리즘

점수 우선순위

1 장소 2 목적 3 개인화 4 가격, 거리

$$Score = (10 - 장소순위) * 10^5$$

$$= 943,712$$

$$+ \sum (\text{목적지수} - 1) * 10^4$$

$$+ 개인화점수 * 10^3$$

$$+ (a * \text{가격등급} + b * \text{거리등급}) * 10^2$$

$$= 843,563$$

74

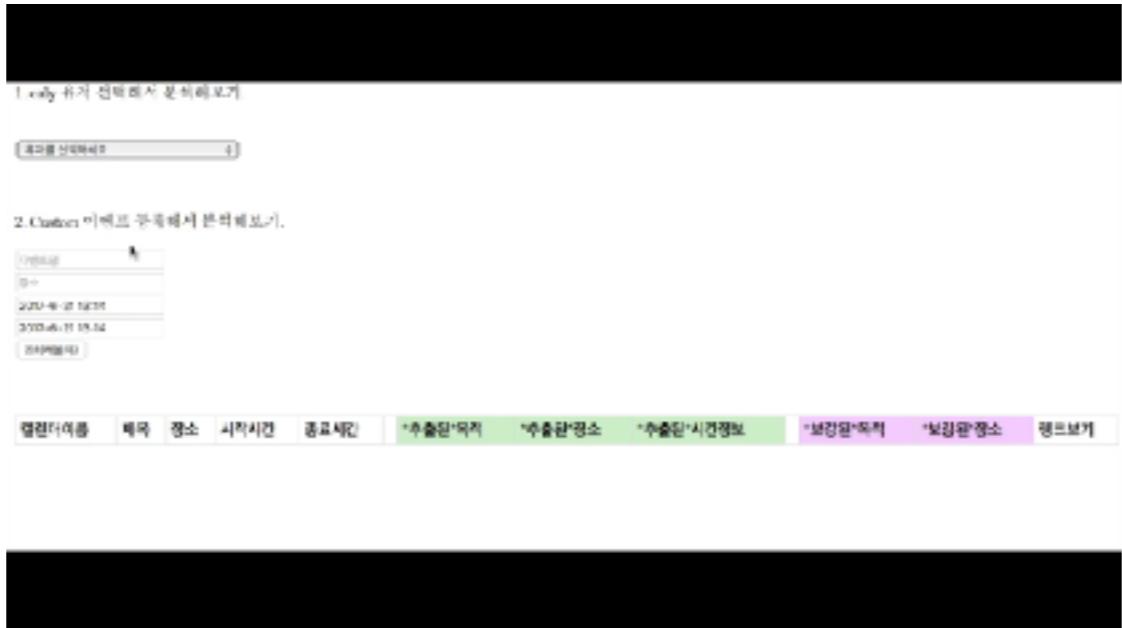
정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

앞서 보여드렸던 '7시 성민이랑 데이트'란 일정에 대해선

추출한 '강남역'에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.

별첨. 01 추천자동화 웹 자동추천

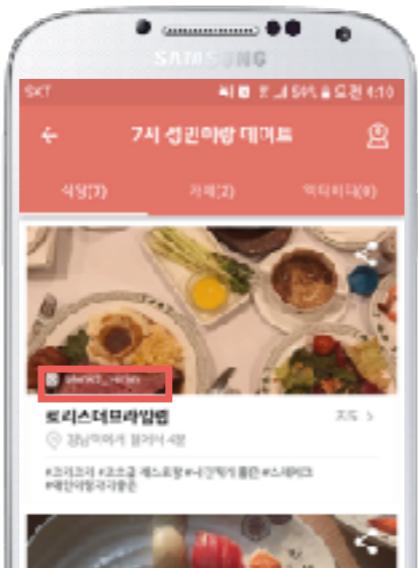


별첨. 01 추천자동화 크롤러

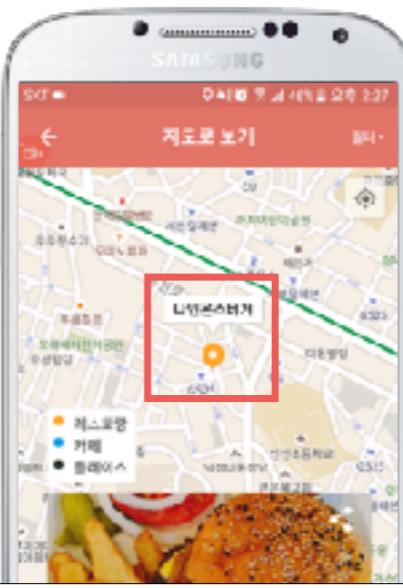
1 데이터 수집 크롤링

별첨. 01 추천자동화 크롤러

② 인스타그램 사진 주소에서 유저 ID를 뽑아내는 크롤러



③ 장소의 주소에서 위도 경도를 추출하는 크롤러



별첨. 02 이미지 출처표기 출처표기방식



▲ admin

- ① 내부적으로 이미지 출처를 저장

“인스타 사진 사용 여부 물었을때 80%가량이 당일에 YES라고 응답”



▲ 인스타그램 사진



▲ pixlee

78

정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

앞서 보여드렸던 ‘7시 성민이랑 데이트’란 일정에 대해선

추출한 ‘강남역’에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.

별첨. 03 맛집 랭킹의 신뢰성 맛집 랭킹

“맛집에 대해 광고성 없는 신뢰성있는 추천을 어떻게 할것인가?”

“티맵 맛집은 왜 사람들이 안갈까?”



▲ 블루리본

- ① 블루리본을 통한 객관적 지표(미슐랭과같은 레스토랑 평가서)
- ② 유저가 실제 선택한 데이터를 가지고 다시 반영
- ③ 망고 플레이트, 다이닝코드와 같은 맛집추천 지표 반영



▲ 맛집 추천 서비스

79

정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

앞서 보여드렸던 ‘7시 성민이랑 데이트’란 일정에 대해선

추출한 ‘강남역’에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.

별첨. 04 비즈니스모델 bm



80

정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

앞서 보여드렸던 '7시 성민이랑 데이트'란 일정에 대해선

추출한 '강남역'에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.

별첨. 05 테스트 주도 개발 tdd

Recommend-system TDD

목적/장소 분석 모듈

Name	Stmts	Miss	Cover
reinforce.py	127	4	97%
test.py	67	0	100%
TOTAL	194	4	98%

테스트케이스 문서화



추천 모듈

Name	Stmts	Miss	Cover
./reco.py	238	44	82%
./test_reco.py	238	3	99%
TOTAL	476	47	98%

81

정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

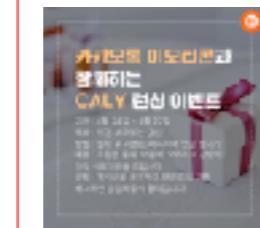
앞서 보여드렸던 '7시 성민이랑 데이트'란 일정에 대해선

추출한 '강남역'에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.

별첨. 06 마케팅 마케팅

caly



런칭이벤트 30명
₩ 40,000원 28일



베태스트 이벤트 20명
₩ 45,000원 28일



카드뉴스 이벤트 17명
₩ 20,000원 7일

제시글	게시일	작성자	작성 일자	행사
2017-08-28	2017-08-28	2017-08-28	2017-08-28	2017-08-28
2017-08-28	2017-08-28	2017-08-28	2017-08-28	2017-08-28
2017-08-28	2017-08-28	2017-08-28	2017-08-28	2017-08-28
2017-08-28	2017-08-28	2017-08-28	2017-08-28	2017-08-28
2017-08-28	2017-08-28	2017-08-28	2017-08-28	2017-08-28

페이스북 마케팅

82

정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

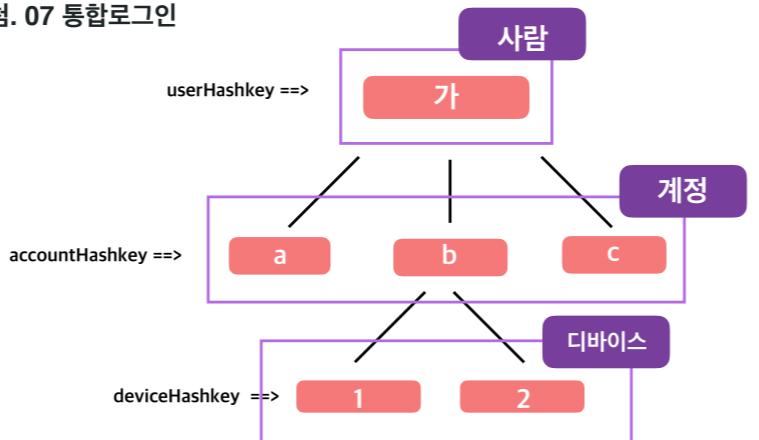
앞서 보여드렸던 '7시 성민이랑 데이트'란 일정에 대해선

추출한 '강남역'에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.

별첨. 07 통합로그인

caly



example

2. 로그인한 계정 'a' 인 유저가 계정 'b'를 추가함.
'가' 의 유저해시키로 계정 'b'를 회원가입시킴
-> 로그아웃(계정 a의 apikey상실) -> 유저가 'b'로 로그인시도.
유저 b의 apikey를 생성 유저 'a'의 apikey 삭제.
(하나의 디바이스에서 두개의 계정이 등록되어있을 필요가없어서)
1. 'b' 가 '디바이스1' 에서 로그인하여 회원가입. apikey '1'을 받음
'디바이스2' 에서 로그인 -> 앗 새로운 디바이스다 새로운 apikey를 발급
두가지요청 모두 같은 유저를 가지고, 디바이스 정보만 다름.

83

CASE 1

다른 디바이스에서 로그인할 경우

CASE 2

계정 추가후 추가한 계정으로 로그인할 경우

CASE 3

다른디바이스에 등록된 계정을 자신의 계정의 추가할 경우

CASE 4

모든계정을 삭제할 경우

CASE 5

회원 탈퇴를 할 경우

CASE 6

비밀번호가 바뀌었을 경우

정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

앞서 보여드렸던 '7시 성민이랑 데이트'란 일정에 대해선

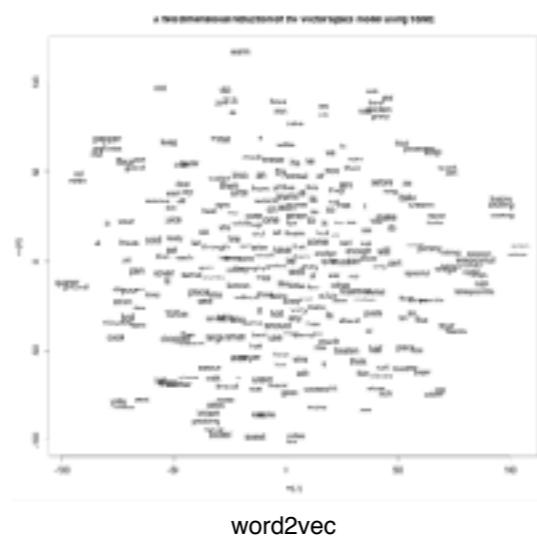
추출한 '강남역'에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.



타이틀은 조금 더 고민해볼것

별첨. a01 성능 고도화

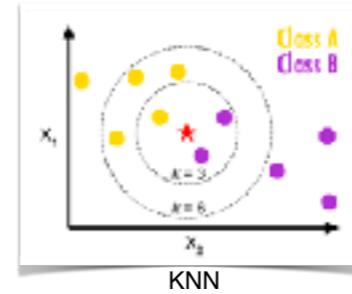


$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Likelihood Class Prior Probability
Posterior Probability Evidence Term Probability

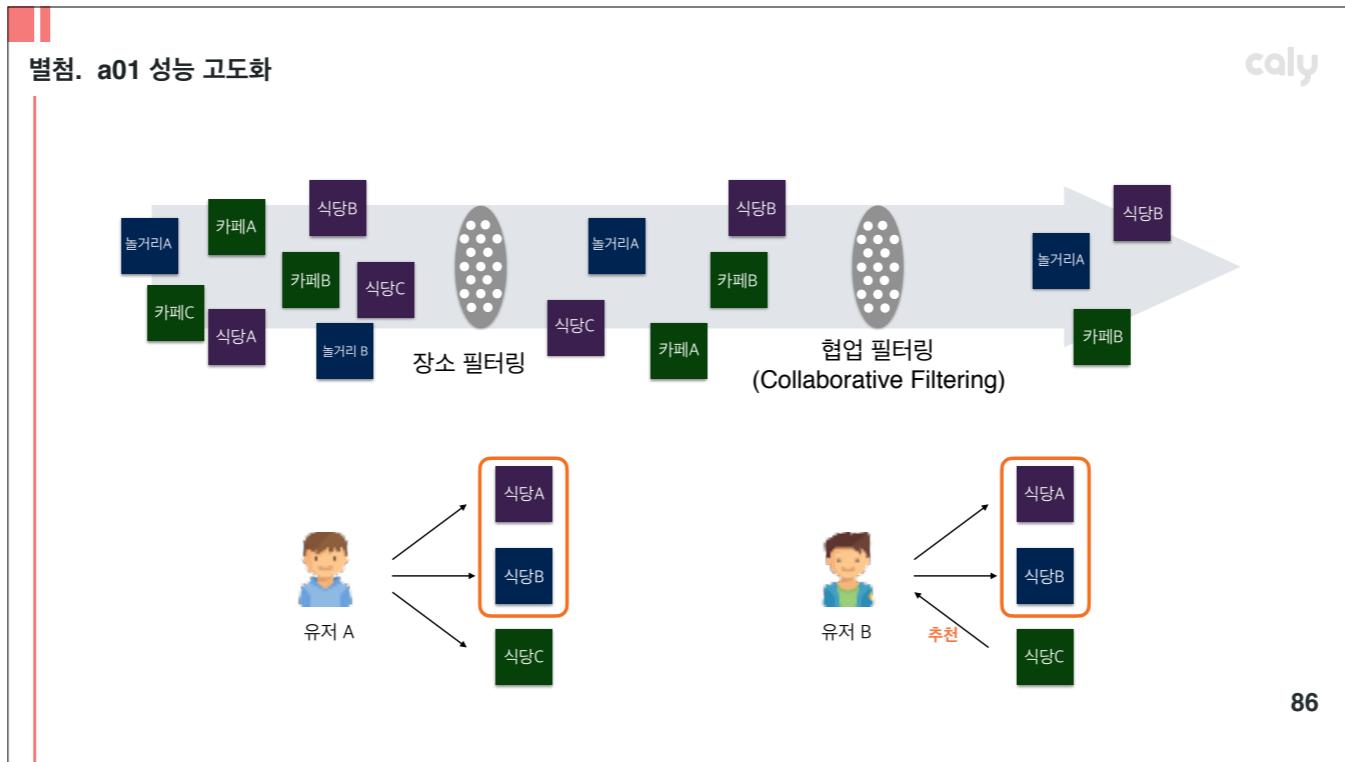
$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

나이브 베이즈 분류



85

현재 nlp와 커스텀 딕셔너리만으로 시간/장소/목적을 추출하고 있는데
 추후에는 word2vec 과 같은 머신러닝 모델을 이용해 단어들과의 관계를 유추해
 이것이 장소인지 목적인지 시간인지 분석하도록 개선하고
 나이브베이즈나 KNN과 같은 분류기 모델을 이용해 목적을 더 정확하게 분류하도록 개선해나갈것이다.
 실제로 나이브 베이즈 모델은 적용을 시도해봤으나, 데이터의 부족으로 성능을 내지 못하였다.
 좀 더 데이터가 모이고 난 후 꼭 적용해보고싶다.



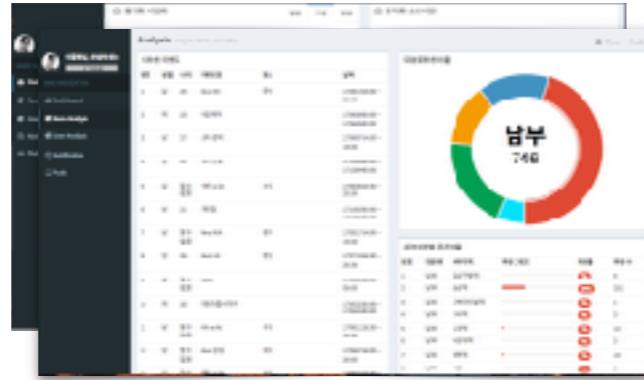
현재는 추천해 줄 수 있는 데이터가 많지 않기에, 최소한으로만 필터링을 하고 있고 그 기준은 '장소'이다.

추천해줄수 있는 데이터와 유저가 충분히 쌓인 후에는

유저가 싫어할만한것들을 걸러내도록 collaborative filtering을 도입할것이다.

cf설명

별첨. a02 유저나 일정 데이터가 부족



모니터링 대시보드



TimeBlock

87

유저수가 부족한건 중요하지 않다.

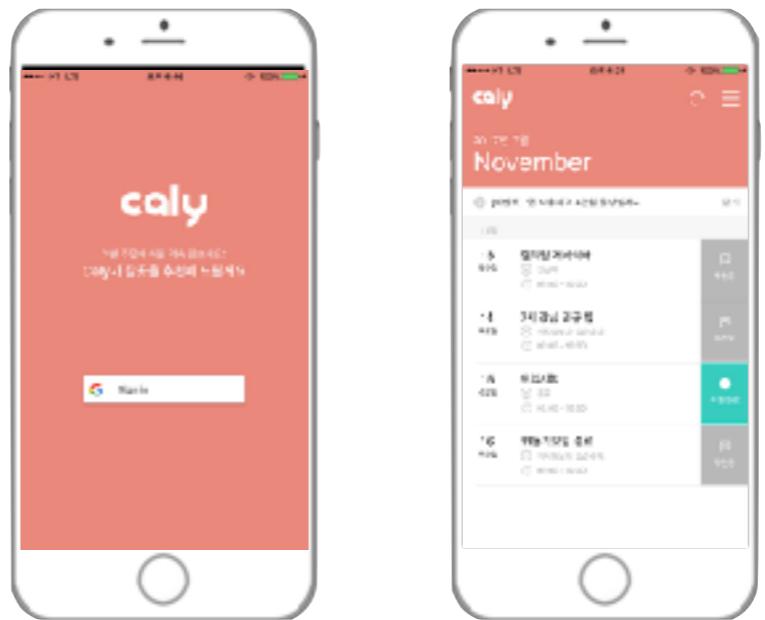
일정정보를 통해 데이터를 분석하는것이 훨씬 더 중요하다.

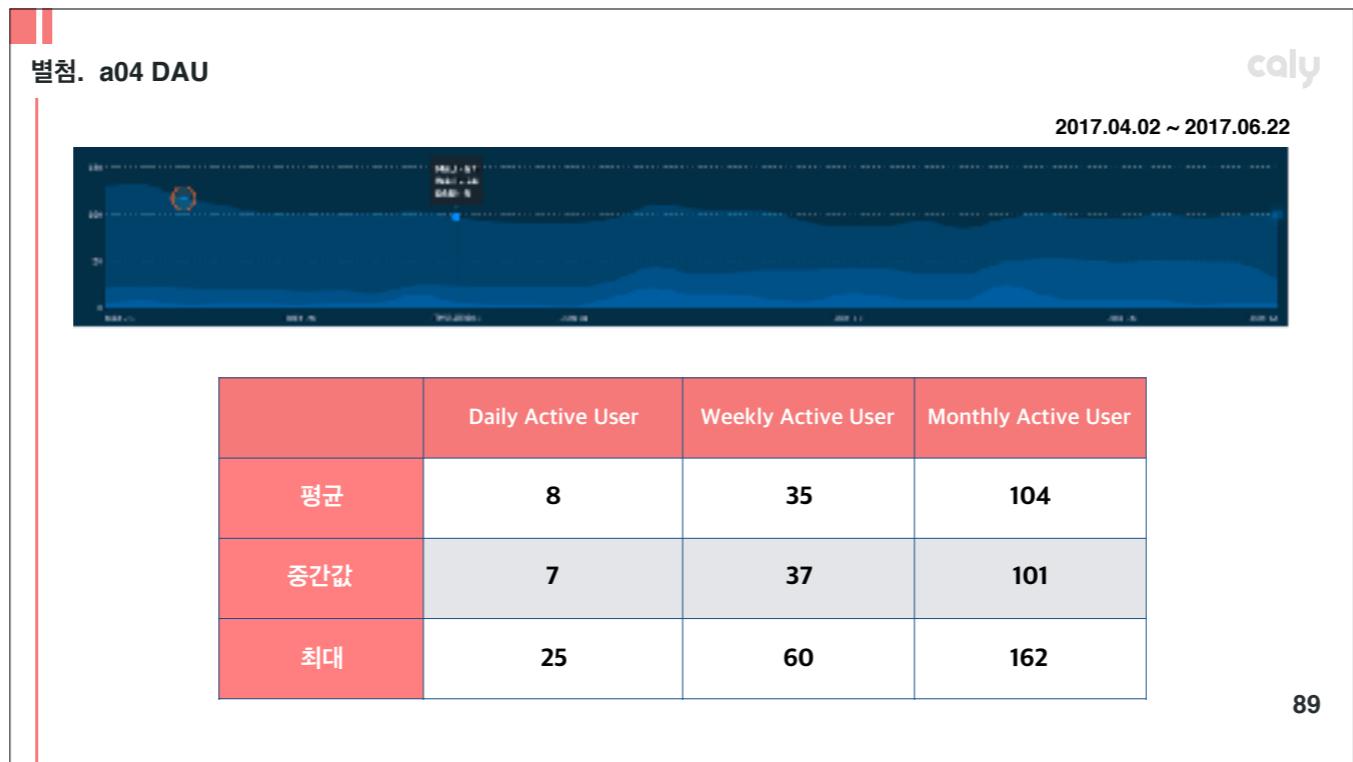
일정정보가 부족한 점에 대해서 우리는 time block 과 같은 캘린더 서비스와 연계해

일정정보를 받아오고, 추천서비스를 제공해주는 방안을 고려중이다.

별첨. a03 ios

comming soon..





유저수가 부족한건 중요하지 않다.

일정정보를 통해 데이터를 분석하는것이 훨씬 더 중요하다.

일정정보가 부족한 점에 대해서 우리는 time block 과 같은 캘린더 서비스와 연계해

일정정보를 받아오고, 추천서비스를 제공해주는 방안을 고려중이다.

별첨. a05 보안

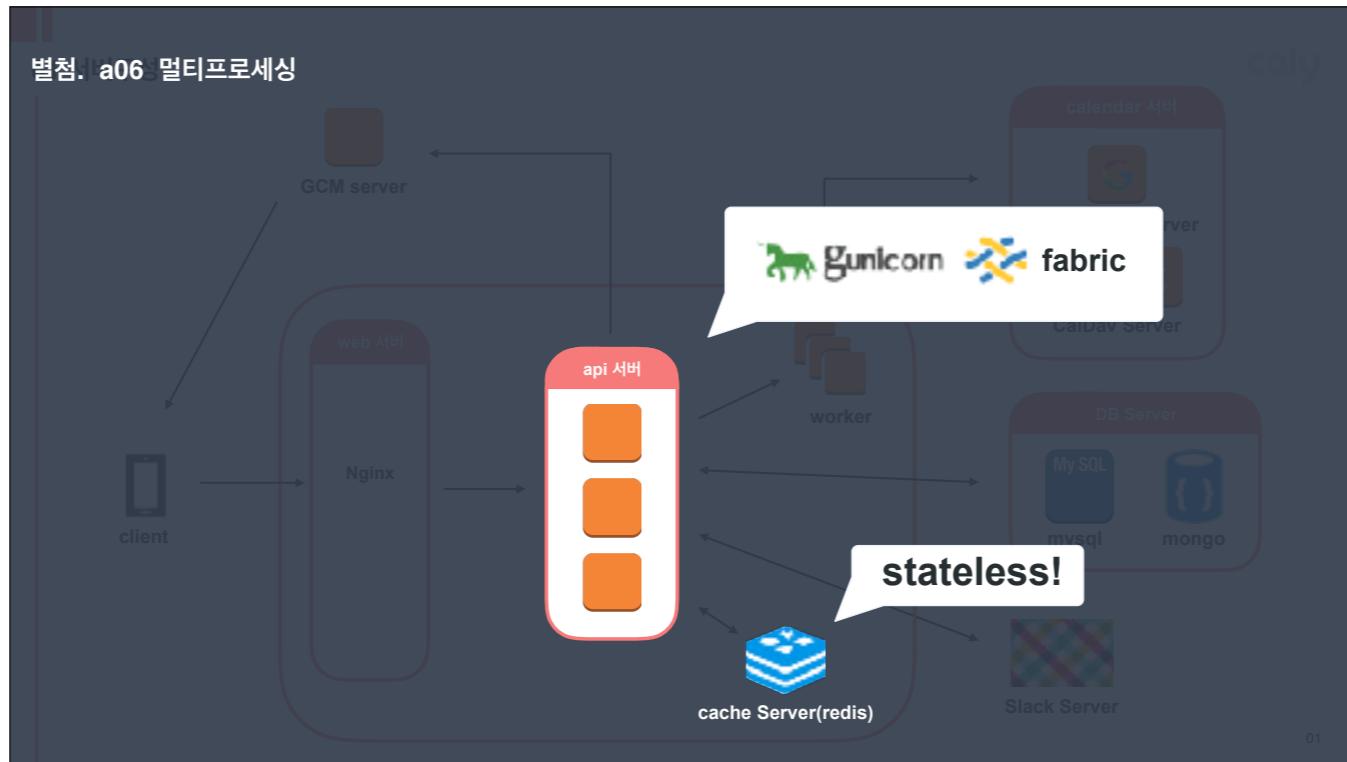


정한 우선순위대로 랭킹 스코어를 이렇게 계산했습니다. 수식만으로는 이해가 힘드실텐데요.

앞서 보여드렸던 ‘7시 성민이랑 데이트’란 일정에 대해선

추출한 ‘강남역’에 대해서

장소순위가 우선적으로 계산되어 신논현역보다 강남역에 있는 데이터가 더 높은 점수를 받은것을 보실 수 있습니다.



nginx => http2.0(사진), ssl, static cached(바로응답),reverser proxy => 확장성을 고려했다.
 api => fabric을 통한 스크립트 배포, 가상환경을 통한 추천모듈 및 외부library depdancy관리,gunicorn 멀티프로세싱
 확장 용이성을 위한 stateless server => apikey ,daily log,
 worker => celery를 사용해 queue worker 를 통해 비동기로 calendar server에 동기화 요청,
 reids => apikey 관리.

외부-> newrelic,jenkins,git

별첨. a07 세부 타겟층선정

	20대 학생	30대 직장인
캘린더 사용률	70%	88%
소비 컨텐츠의 질	정기적인 수입이 없어 저렴한 가격 대비 시간 위주의 소비	여가 생활에서 스트레스 해소 목적. 가격대성능비보다는 서비스의 퀄리티
피드백 반영을 위한 접근성	특정지역의 특성에 기반해 접근가능 (주변 지인이 주로 학생)	주활동 지역에 대한 조사부터 시작

선정한 주요타겟층



20대 대학생

1. 시간적-자원적 제한, 빠른 접근성 필요
2. 추후확장성 고려, 소비컨텐츠이용 결과기반 확장이 용이하다

별첨으로 빼기

서비스 특성상 최대한 빠른 유저피드백이 필요하고, 저희 인원이 적다는 자원적 제한이 있어 빠른 접근성이 필요합니다.

“사유가 명확하면 빨리 지나가도 어색하지 않을듯”

우리의 어쩔 수 없는 사정? 말이 이상하다. 모임, 카페 등에 적합한 사람같다. 30대 초 직장인에게는 크게 차이없이 적용가능할것
이부분이 잘 와닿지는 않는다.

저가의 컨텐츠에서 고가의 컨텐츠로 확장하는 것이 용이하다

1. 20대, 30대, 40대 이상 시장성 조사결과
2. 20대 학생층과 30대 직장인으로 구분해서 차별성을 어필
3. 접근성 및 인사이트 어필