

3조

챗봇을 활용한
메뉴 추천 앱

강승수 김태원
윤수름 이준호
전영준 정태현

차례

- 1 ■ 프로젝트 개요
- 2 ■ 진행 프로세스
 - 2-1 개발 일정
 - 2-2 개발 환경
- 3 ■ 프로세스 상세
 - 3-1 구조도
 - 3-2 ERD
 - 3-3 기술 설명
- 4 ■ 프로세스 시연
- 5 ■ 개선 방안



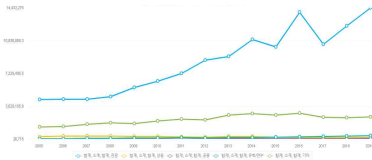
1. 프로젝트 개요

1. 프로젝트 개요

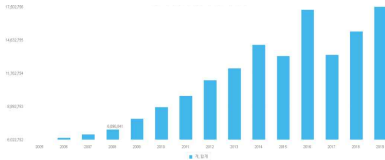
외국인 유입의 증가

한류 문화의 확산으로 관광을 목적으로 한국을 방문하는 **외국인의 수가 증가**하고 있으며 이에 따라 한국 문화 중 하나인 **한국음식에 대한 관심이 증가**할 것이라 예측됨.

입국목적별 외국인 방문객 현황_05_19



외국인 방문객 현황_05_19



1. 프로젝트 개요

■ AI 기술 서비스의 발전

AI 기술의 발달로 AI가 다양한 분야에 사용되고 있고 많은 결정에 대해 쾌적한 응답을 하는 대화형 AI가 발전함에 따라 챗봇을 통한 음식 및 음식점 추천 서비스를 개발하고자 함.



2. 진행 프로세스

2. 진행 프로세스

2-1. 개발 일정

세부내용	주차				
	1주차 (~9/23)	2주차 (~9/30)	3주차 (~10/7)	4주차 (~10/14)	5주차 (~10/21)
프론트	설계	UI개발	챗봇 대화 방식 구현 & 구글로그인		구글 지도
AI모델	설계	의도분류	추천모델 설계	추천모델 학습	추천모델 완성
백엔드	설계	DB생성	음식점 찾기 API 개발	AWS 환경설정	구글 소셜로그인(OAuth2.0개발)
		서버 - DB 연동	메뉴 사전서비스 개발	클라이언트-서버 연동	Filter 개발
6주차 (~10/28)		7주차 (~11/3)		8주차 (~11/9)	
구글로그인 토큰 및 리뷰, 찜목록		구글 로그아웃 및 리뷰, 챗봇 연동, 디버깅		발표 준비	
추천모델 개선		서버 연동 및 개선			
개발된 서버 AWS에 업로드		통합 및 디버깅			
리뷰, 찜목록 서비스 개발					

2. 진행 프로세스

2-2. 개발 환경

협업툴



EC2



RDS

데이터
전송타입



Json

API



Google
Cloud



GoogleMap



DialogFlow



RestApi

사용언어



JAVA



Python

개발툴



PyCharm



STS



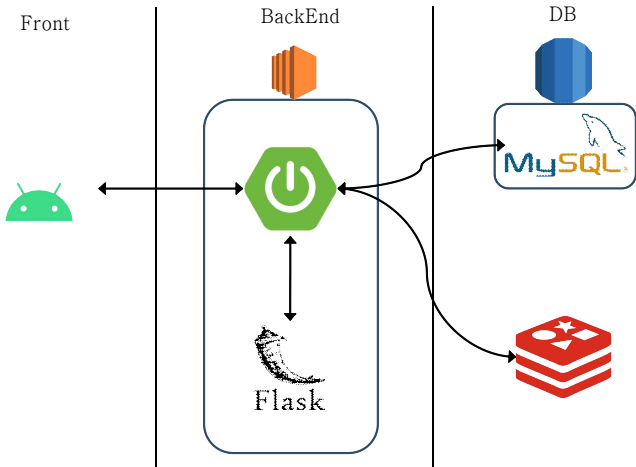
Android



3. 프로젝트 상세

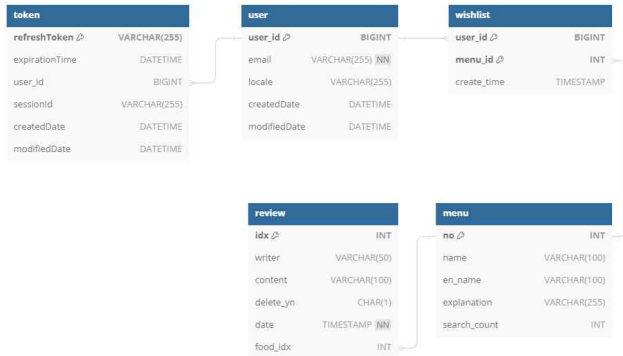
3. 프로젝트 상세

3-1. 구조도



3. 프로젝트 상세

3 - 2. ERD

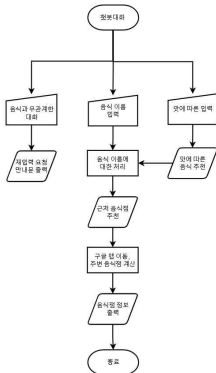
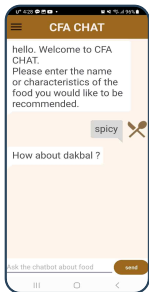
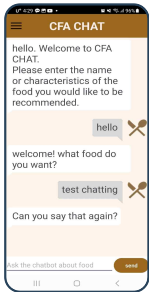


3. 프로젝트 상세

3 - 3. 기술 설명

챗봇

- 챗봇 시작 시 안내말 표시
- 음식 이름을 영어로 입력 시 그에 해당하는 음식 추천
- 서버에서 처리 후 사용자 주변의 조건에 맞는 음식점 최대 5개 추천
- 답변 버튼을 통해 Google 지도로 이동

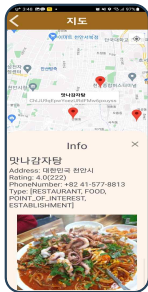
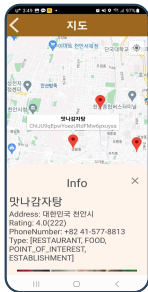
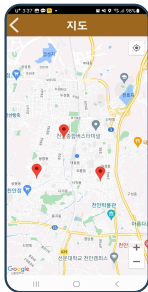


3. 프로젝트 상세

3-3. 기술 설명

지도

- 챗봇에게 추천 받은 **음식점의 위치정보**를 지도에 **마커**로 표시
- 마커 클릭 시 음식점에 대한 **상세정보**를 출력
- 구글 맵 툴바를 통해 구글 지도로 이동

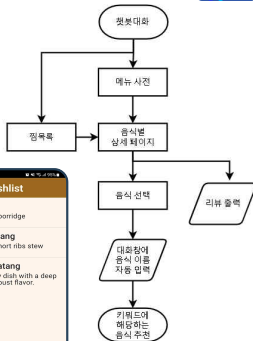
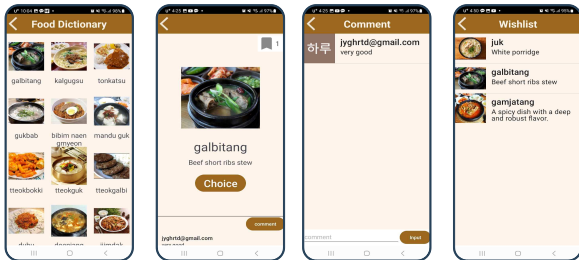


3. 프로젝트 상세

3-3. 기술 설명

음식 사전 & 찜목록 & 리뷰

- 어플에서 보유중인 음식 정보를 **이름과 이미지 형태로 출력**
- 항목 선택 시 **상세 페이지**로 이동
- Choice 버튼을 통해 챗봇 화면으로 이동하여 대화를 진행 가능
- **최근 5개의 리뷰** 출력 및 **찜 항목** 추가
- 찜목록 이동시 찜한 목록에 한하여 **음식정보 출력**

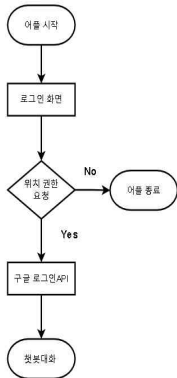
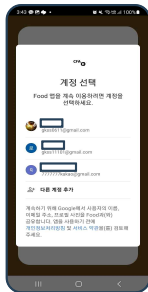
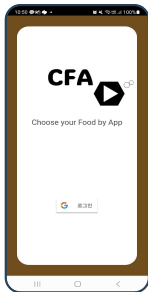
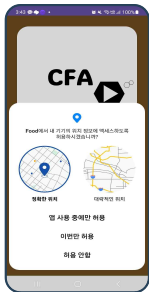


3. 프로젝트 상세

3-3. 기술 설명

로그인

- 외국인이 사용함에 따라 구글 로그인만 구현
- OAuth2.0 방식의 로그인

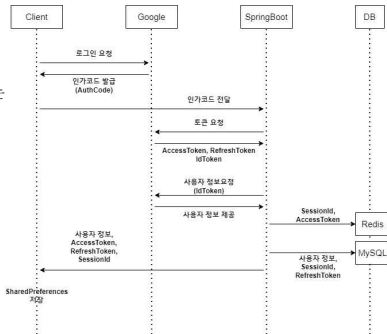


3. 프로젝트 상세

3 - 3. 기술 설명

OAuth2.0을 적용한 로그인

- OAuth2.0를 이용한 보안, 액세스 토큰 또는 리프레시 토큰을 이용한 사용자 검증

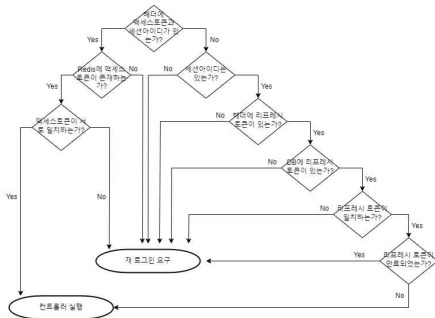


3. 프로젝트 상세

3-3. 기술 설명

액세스 토큰과 리프레시 토큰을 확인하는 필터

1. 구글의 액세스 토큰은 Bearer 타입이지만 jwt토큰이 아니다.
2. 세션 불일치로 인해 SecurityContextHolder와 Session의 기능을 사용 불가능
3. 인터셉터의 한계
 - 액세스 토큰 또는 리프레시 토큰이 유효할 경우에만 컨트롤러를 실행 시킨다.

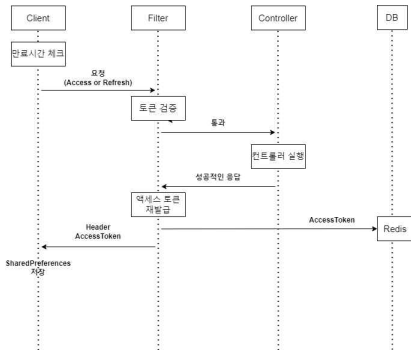


3. 프로젝트 상세

3-3. 기술 설명

갱신되는 액세스 토큰

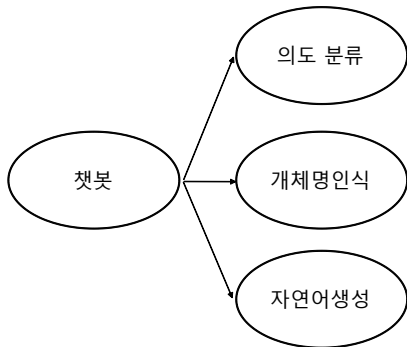
- 프론트에서 만료시간을 검증한뒤 액세스 토큰 또는 리프레시 토큰으로 요청
- 매 요청마다 액세스 토큰을 재발급



3. 프로젝트 상세

3-3. 기술 설명

챗봇의 핵심 기능



- 의도분류와 개체명인식은 **dialogflow** 사용
- 자연어생성은 **Rule**방식으로 직접 입력

3. 프로젝트 상세

3 - 3. 기술 설명

의도 분류

- 인텐트라고 불리는 의도 라벨링

	intent	label
0	add another song to the cita rom ntica playlist	AddToPlaylist
1	add clem burke in my playlist pre party r b jams	AddToPlaylist
2	add live from aragon ballroom to trapeo	AddToPlaylist
3	add unite and win to my night out	AddToPlaylist
4	add track to my digster future hits	AddToPlaylist
...
11779	can a i get the movie schedule for sympathy fo...	SearchScreeningEvent
11780	find movie schedules for animated movies aroun...	SearchScreeningEvent
11781	what time is bordertown trail showing	SearchScreeningEvent
11782	in the neighbourhood find movies with movie times	SearchScreeningEvent
11783	what cinema has the closest movies	SearchScreeningEvent

11784 rows x 2 columns

Default Fallback Intent

Default Welcome Intent

input_food

input_property ⓘ ▾

95 **savory** food is my best

95 if you want **spicy** food i can recommend one

95 i want some **illy**

95 i want **tart** one

95 can you recommended **mild** flavor food

95 i want to some **zesty** and made with **pork**

3. 프로젝트 상세

3 - 3. 기술 설명

개체명 분류

cold	cold, chilly, cool, frigid, freezing, icy, frosty, nippy, gelid, wintry, brisk
spicy	spicy, fiery, pungent, zesty, tangy, picante, burning, sharp, peppery, scorching
salty	salty, savory, briny, saline, salted

“ **savory** food is my best

“ if you want **spicy** food i can recommend one

“ i want some **oily**

“ i want **tart** one

“ can you recommended **mild** flavor food

“ i want to some **zesty** and made with **pork**

PARAMETER NAME	ENTITY	VALUE
user_foodname	@user_foodName	\$user_foodname
user_taste	@user_taste	\$user_taste
user_ingredient	@user_ingredient	\$user_ingredient
Enter name	Enter entity	Enter value

“

- 직접 개체명 라벨링 및 자동 라벨링

3. 프로젝트 상세

3-3. 기술 설명

메뉴 기반 주변 음식점 찾기



JSON 데이터 형식

Request		Response	
key	values	key	values
foodName	{음식이름(ex.떡볶이)}	name	{음식점 이름}
latitude	{위도}	rating	{평점}
longitude	{경도}	reviewCount	{리뷰 수}
		distanceInMeters	{거리}
		placeId	{장소의 고유 값}

API 테스트 결과

```
POST http://localhost:8080/restaurants
{
  0: {
    0: "foodName",
    1: "떡볶이",
    2: "latitude",
    3: "37.5095",
    4: "longitude",
    5: "126.899"
  }
}
```

```
{
  0: {
    1: {
      0: "name",
      1: "교촌치킨 연국점",
      2: "rating",
      3: "4.48",
      4: "reviewCount",
      5: 82,
      6: "distanceInMeters",
      7: 1370.9,
      8: "placeId",
      9: "ChIJ3DV0F_E0jFOURq85Jey6Vzcw"
    }
  }
}
```

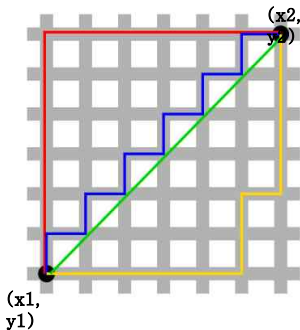
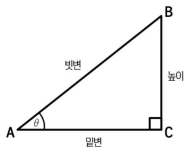
3. 프로젝트 상세

3-3. 기술 설명

두 좌표 간의 거리 계산

- 맨해튼 거리 : 19세기의 수학자 헤르만 민코프스키가 고안한 알고리즘

$$|x1 - x2| + |y1 - y2|$$



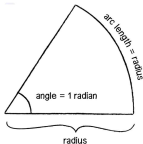
3. 프로젝트 상세

3-3. 기술 설명

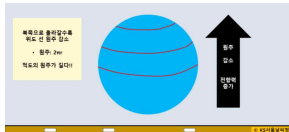
하지만, 지구는 둥글다!

- 지구는 3차원이기에 2차원 좌표 거리를 계산하는 맨해튼 거리는 오류가 있다.

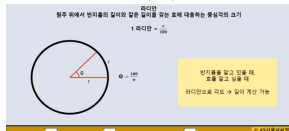
- 라디안(radian)** : 원의 반지름에 대한 호의 길이의 비



위도 경도 1° = 몇 km?
KS서울남씨정 / 서울기상센터
위도 선



위도 경도 1° = 몇 km?
KS서울남씨정 / 서울기상센터
라디안 (호도법)

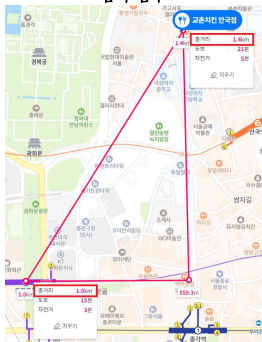


```
private double calManhattanDistanceInMeters(double lat1, double lon1, double lat2, double lon2) {  
  
    final int R = 6371; // 지구 반지름 (km)  
    double dLat = Math.abs(Math.toRadians(lat2 - lat1));  
    double dLon = Math.abs(Math.toRadians(lon2 - lon1));  
  
    double distance = dLat + dLon;  
    double distanceInKm = R * distance;  
    return distanceInKm * 1000;  
}
```

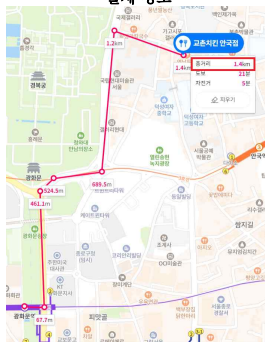

3. 프로젝트 상세

3-3. 기술 설명

삼각 함수



실제 경로



3. 프로젝트 상세

3 - 3. 기술 설명

추천 모델 로직



korean_nairomanized_name	taste	made_with	english_na	menu_num	
갈비	galbi	greasy,sweet	beef	Grilled bee	0
갈비찜	galbi-jjim	greasy,sweet	beef	Braised be	1
게장	gejang	salty,greasy	seafood,rice	Marinated	2
생선구이	saengseon-gui	greasy,salty	fish	Grilled fish	3
곱창전골	gopchang-jeong	spicy,hot	beef,pork	Spicy beef	4
김치볶음밥	kimchi-bokkeumb	spicy,greasy	kimchi,rice	Kimchi frie	5
김치찌개	kimchi-jjigae	spicy,hot	kimchi,pork	Kimchi ste	6
물냉면	mul-naengmyeon	cold,sour	noodles	Korean col	7

- **taste**와 **made_with**의 값으로 **romanized_name**을 특정하여 출력

3. 프로젝트 상세

3-3. 기술 설명

AI 모델 선정

- 코사인유사도

=> **재료**와 **맛**에 각 점수를 주고
메뉴에 점수를 주는 방법 사용
재료와 맛이 달라도 점수가 같으면
유사하다 판단해 추천 모델에
적합하지 않음

- 원핫인코딩

=> 단어 간의 **유사성**을 알 수
없기에 재료와 맛으로 음식을
추천해주는 추천 모델에 적합하지
않음

- 딥러닝

장기적으로 봤을 때, 좋은 표본이나
단기적인 프로젝트 특성상 **적은**
데이터셋으로 인해, **후에 사용** 할
의향있음.

중심 단어 주변 단어

The **fat** cat sat on the mat

The **fat** cat sat on the mat

The fat **cat** sat on the mat

The fat cat **sat** on the mat

The fat cat sat **on** the mat

The fat cat sat on **the** mat

The fat cat sat on the **mat**

중심 단어	주변 단어
[1, 0, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 1, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 1, 0, 0]	[0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 1, 0]	[0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0, 1]	[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 0, 1]

3. 프로젝트 상세

3-3. 기술 설명

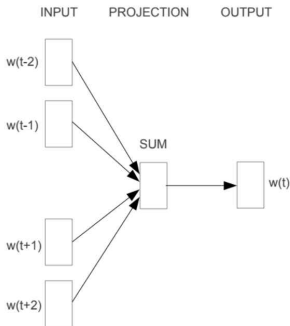
Word2Vec

- Word2Vec

=> 주변단어로 중심단어인
특정단어를 특정하기 용이함

- Word2Vec 중 CBOW를 사용

• 주변 단어들의 원핫벡터가
input 레이어아웃으로 들어가고
output층의 레이어아웃에
예측하고자 하는 중간단어의
원핫벡터가 나오게 함.





4. 프로젝트 시연



5. 개선 방안

5. 개선 방안



다양한 언어 지원



비슷한 단어 감지



API 응답 속도 개선



리뷰 기능 개선



감사합니다!