



국민대학교
소프트웨어융합대학


캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	FOODAY
팀 명	주먹
문서 제목	7조_FOODAY_중간보고서

Version	1.5
Date	2021-04-04

팀원	정 지연 (조장)
	박 종민
	차 유미
	최 은솔
지도교수	황 선대 교수

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05


CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어융합학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 “FOODAY”를 수행하는 팀 “주먹”의 팀원들의 자산입니다. 국민대학교 소프트웨어융합학부 및 팀 “주먹”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	중간보고서-FOODAY.doc
원안작성자	차유미
수정작업자	박종민,최은솔,정지연,차유미

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2021-03-31	차유미	1.0	최초 전체 개요 작성	프로젝트 목표, 개발목표, 수행내용, 수정사항, 고충 및 건의사항
2021-04-01	박종민	1.1	수행 내용 및 중간결과	크롤링, 추천 시스템(날씨)
2021-04-02	최은솔	1.2	수행 내용 및 중간결과	추천 알고리즘
2021-04-03	정지연	1.3	수행 내용 및 중간결과	데이터 수집, 태그, 카테고리, 추천 시스템(시간대), aws personalize
2021-04-03	차유미	1.4	내용 흐름 검토 및 수정	전체적인 보고서 흐름 정리
2021-04-04	전체	1.5	전체 내용 검토	오타 검토 및 내용 첨삭

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

목 차

1. 개요	5
1.1 프로젝트 개요	5
1.2 추진배경 및 필요성	6
2. 프로젝트 목표	9
3. 수행 내용 및 중간결과	10
3.1 수행목표	10
3.2 수행내용(개발목표에 따른 프로젝트 수행 내용 상세하게 기술하고 목표했던 진도와 현재 진도상황을 비교분석)	11
3.2.1 user data collection	11
3.2.2 데이터 크롤링을 통한 음식점 데이터 수집	11
3.2.3 배달음식 카테고리 분석 및 생성	12
3.2.4 배달음식 카테고리 태그 분석 및 생성	13
3.2.5 DB 구축	13
3.2.6 날씨 기반 추천 기능 구현	15
3.2.7 시간대 기반 추천 구현	15
3.2.8 AWS personalize 구축	16
3.2.9 추천 알고리즘 구현	16
3.2.10 Application(using React Native)	17
4. 수정된 연구내용 및 추진 방향	25
수정사항	25
5. 향후 추진계획	28
5.1 향후 계획의 세부 내용	28
6. 고충 및 건의사항	29

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

1 개요

1.1 프로젝트 개요

‘오늘 뭐 먹지?’ 라는 고민은 혼자 밥을 먹을 때도, 친구들과 만나서 먹을 때도, 회사 점심시간에도 누구나 한번쯤 해보았을 법한 고민이다. 이 고민은 최근들어 코로나로 인해 외출을 삼가하게 되면서 ‘오늘 뭐 시켜먹지?’로 변화하였다.

실제로 마케팅·빅데이터 분석업체인 NICE 디엔알에서 스마트폰 이용자의 앱 로그 데이터를 분석한 결과, 코로나 첫 확진자 발생 이후로 주요 식음료 배달 앱의 이용자 수가 눈에 띄게 지속적으로 증가했다고 하며, 코로나 이후 배달앱 카테고리의 이용이 **51.8%**로 가장 크게 증가했다는 결과를 얻었다. 배달 수요가 급증함과 동시에 배달음식의 종류는 많아지고, 소비자가 선택해야하는 선택지 역시 다양해져 배달음식을 주문하기까지의 시간은 점점 더 지체되어 갈 수 밖에 없다.


우리는 실제 사용자들이 배달음식을 주문할 때 얼마나 고민하는지를 알아보기 위해 배달어플 사용자 **480명**을 대상으로 혼자일 때, 여러명일 때 두 상황으로 나누어 조사를 실시하였다. 그 결과, 두 상황에서 모두 고민한다고 대답한 사람이 **90%** 이상이었다. 이를 통해 많은 사람들이 실제로 배달음식을 주문할 때 결정을 못해 시간을 지체하고 있다는 사실을 인지하게 되었다.

혼자 배달음식을 시킬 때, 메뉴를 결정하기까지 얼마나 고민하는 편인가요?

응답 480개



<배달음식 고민정도에 대한 설문조사 - 혼자일 때>

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

여러 명의 사람들과 함께 배달음식을 시킬 때, 메뉴를 결정하기까지 얼마나 고민하는 편인가요?

응답 480개



<배달음식 고민정도에 대한 설문조사 - 여럿일 때>


해당 프로젝트에서는 이러한 고민을 데이터 기반 추천 시스템을 통해 사용자가 신뢰할 수 있는 추천 목록을 제공하여 결정을 내릴 수 있도록 도움을 주어 해결하고자 한다. 더하여 서비스를 사용하는 사용자들이 혼자서 추천시스템을 제공받을 뿐만 아니라, 여러명의 사람들과 함께 공통의 추천목록을 제공받을 수 있도록 다수를 위한 추천시스템도 개발하고자 한다.

1.2 추진배경 및 필요성

1.2.1 기 개발된 시스템 현황

1) 고잇(Go Eat)

해당 어플은 상황에 따른 개인 맞춤형 메뉴추천을 제공한다. 초기 랜덤음식에 대한 선호도 조사를 실시하여 유저의 선호도를 파악하고, 누구와 먹는지, 기분 등의 상황을 입력받아 그에 맞는 추천 목록을 제공한다. 추천 목록 제공 후에는 항목에 대한 피드백을 받을 수 있고, 해당 메뉴에 맞는 음식점을 보여준다. 음식점을 클릭하면 실제로 판매하는 메뉴들을 볼 수 있고, 네이버 주문과 연동하여 음식점 예약을 진행할 수 있다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05



<고잇의 앱 소개 이미지>

2) 땡땡

해당 어플은 실제 배달 앱이며, 나의 주문 패턴 분석 및 다른 사용자의 최근 주문수, 재주문율, 즐겨찾기 수, 배달비, 평균주문금액 등 맛집 선정에 도움되는 요소들의 조합으로 최적의 맛집 추천해주는 기능을 갖추고 있다.

 땡땡 강남센터

역삼동 689-11

전다정님을 위한
오늘의 추천 맛집이에요.




바라다김선생 논현아크로힐스점
#김밥 #달밥 #만두



써브웨이 역삼점
#샌드위치 #내맘대로 골라서 #한 입

<땡땡의 앱 소개 이미지>

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

1.2.2 기 개발된 시스템 문제점

1) 고잇(Go Eat)

해당 어플은 모든 외식/배달음식메뉴와 음식점을 대상으로 하고 있어 배달음식을 주문하고자 하는 사용자에게 적절한 음식메뉴와 음식점을 추천할 수 없다. 또한 배달음식을 주문할 때, 다수가 함께 주문하는 경우에는 해당 추천 서비스는 사용자 한사람을 위한 추천목록이기 때문에 큰 도움을 줄 수 없다. 추천 알고리즘은 자세한 내용을 알 수 없기 때문에 그 부분에 대해서는 평가하지 않도록 하겠다.

2) 땡땡


해당 어플은 .배달음식 메뉴가 아닌 음식점 추천으로, 사용해보니 음식점은 자체적으로 주문 많은 순, 리뷰 많은 순, 거리 순 등의 필터를 사용해서 카테고리 내의 음식점을 선정하는 과정을 거치곤 하는데, 제공해주는 추천목록은 어떤 기준으로 추천됐는지 알 수 없어 신뢰도가 떨어진다. 또한 위의 어플과 마찬가지로 한사람을 위한 추천 목록이기 때문에 다수가 함께 주문하는 경우 큰 도움을 줄 수 없다.

1.2.3 개발할 시스템의 필요성

많은 비즈니스 영역에서 추천시스템은 단순히 필요한 수준을 넘어 핵심적인 경쟁력의 구성요소가 됐다. 넷플릭스 역시 대여되는 영화의 2/3이 고객에게 추천되는 영화라고 한다. 코로나로 인해 점점 커지고 있는 배달앱 시장 내에서 경쟁력있는 배달서비스를 제공하기 위해서도 추천시스템은 필수적이라고 할 수 있다.

기존의 추천서비스를 보았을 때, 공통적으로 서비스 이용자의 가구 수를 고려하지 않는다는 점을 확인하였다. 소비자 데이터 전문 기업 오픈서베이에 따르면 서비스 이용자가 가구수 별로 1~2인 가구가 63.6%, 3인 이상이 57.8%라고 한다. 고민의 시간을 줄이고 결정을 도와주는 추천시스템이 아무리 많아도 63%의 사용자에게만 적절한 기능이었던 것이다.

위 같은 기 개발 서비스 분석을 통해, 배달음식 사용자를 위한 1인 혹은 다수를 함께 만족시키는 추천서비스의 필요성을 느꼈다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

2 프로젝트 목표

오늘날 코로나의 영향으로 배달 수요가 급증하는 추세이다. 더불어 소비자가 선택해야 하는 메뉴의 카테고리별 세부항목 또한 다양해지면서 사용자가 주문을 하기 위해 모든 종류의 음식을 접하기 어려워졌고, 음식 종류의 다양성으로 인해 음식을 주문하기까지의 시간이 지체되는 현상이 발생하고 있다.

사용자 맞춤형 배달음식 추천 서비스인 “FOODAY”는 이러한 사람들의 메뉴에 대한 고민을 감소시키고 서비스 이용 대상의 범위를 1인이 아닌 다수로 확장하는 것을 목표로 한다. 프로젝트 목표는 다음과 같다.

1. 다양한 시나리오의 개인 맞춤형 배달음식 추천 시스템을 제공한다.

1-1 수집된 유저 데이터를 분석하여 선호도를 기반으로 음식 추천 정보를 제공한다.


1-2 사용자의 주문내역을 분석해 가져온 날씨 데이터를 기반으로 날씨 맞춤형 음식 추천 정보를 제공한다

1-3 사용자의 주문내역을 분석해 가져온 주문 시간 데이터를 기반으로 시간대별 맞춤형 음식 추천 정보를 제공한다.

2. 다수의 사용자가 사용 시, 각 사용자의 추천리스트를 조합하여 다수를 위한 공통의 배달음식 추천리스트를 제공한다.

3. 추천시스템에 데이터를 쌓고, 추천시스템을 시연할 수 있도록 주문기능을 약식으로 개발하여 제공한다.

본 프로젝트를 통해 “FOODAY”는 위의 명시된 사항과 같이 기존의 배달 앱들과는 차별화된 서비스를 제공하고자 한다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05


3 수행 내용 및 중간결과

3.1 수행목표

위에서 언급한 프로젝트 목표를 수행하기 위한 세부적인 수행 목표는 다음과 같다.

1. user data collection
2. 데이터 크롤링을 통한 음식점 데이터 수집
3. 배달음식 카테고리 분석 및 생성
4. 배달음식 카테고리 태그 분석 및 생성
5. DB 구축
6. 날씨 기반 추천 기능 구현
7. 시간대 기반 추천 기능 구현
8. AWS personalize 구축
9. 추천 알고리즘 구현
10. Application (by react native)


다음 페이지에서는 이러한 수행 목표를 진행하기 위해 계획한 일정과 현재까지 완료된 사항에 대한 결과, 해야될 일에 대한 계획과 내용을 설명한다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

3.2 수행내용(개발목표에 따른 프로젝트 수행 내용 상세하게 기술하고 목표했던 진도와 현재 진도상황을 비교분석)

프로젝트에서 계획한 진도는 다음과 같다.

항목	세부내용	3월	4월	5월
분석	주제결정 및 시나리오 작성			
	배달음식 카테고리 분석 및 생성			
	배달음식 카테고리 태그 생성			
	데이터 크롤링을 통한 음식점 데이터 수집			
	유저 데이터 수집			
설계	개발 환경 구축			
	DB 설계 및 구축			
	추천 알고리즘 설계			
구현	Application 구현			
	날씨 API 연결			
	AWS personalize 구축			
	API 구현			
	추천 알고리즘 구현			
	배포			
테스트	추천 알고리즘과 서비스 연결			
	추천 알고리즘 테스트 및 성능 개선			
	추천 알고리즘과 AWS personalize 성능 비교			
	전체 프로세스 및 시스템 테스트			

 <div> <p>국민대학교</p> <p>소프트웨어융합대학</p> <p>캡스톤 디자인 I</p> </div>	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

3월 내에 완료하고자 했던 계획 중 데이터 크롤링을 통한 음식점 데이터 수집이 현재 진행 중인 사항을 제외하고는 모두 마쳤다. 세부적인 수행내용에서 그 내용과 결과를 공유할 것이고 4월, 5월에 진행할 수행내용은 설계와 계획을 작성하여 진행상황을 공유한다.

3.2.1 user data collection

추천 알고리즘 학습데이터에 필요한 유저 데이터를 얻기 위해 구글 폼 설문조사를 진행하였다. 기존 계획했던 데이터 수집 방법은 배달어플의 주문기록을 요청하여 받아오는 것이었지만, 보안상의 문제로 불가능하게 되어 구글 폼 설문조사를 통해 데이터 수집을 진행하는 것으로 수정하였다.

설문조사에서 수집하는 유저 데이터는 성별, 연령대, 배달음식 주문 시 음식 양의 중요도(1~5), 배달음식 주문 시 음식 맛의 중요도(1~5), 배달음식 주문 시 음식 가격의 중요도(1~5), 최근 주문기록과 주문시간(0~24), 선호하는 음식카테고리에 대한 점수(1~5)이다.

배달음식 수요조사(응답) ☆										
파일 수정 보기 삽입 서식 데이터 도구 설문지 추가기능 도움말 몇 초 전에 마지막으로 수정했습니다.										
100% 123 기본값(Arri... 10 B I A										
M24 fx '01080052778										
	E	F	G	H	I	J	K	L		
1	배달음식을 주문할 때 맛·배달음식을 주문할 때 양·배달음식을 주문할 때 가·배달음식을 시킬 때, 주로 혼자 배달음식을 시킬 때, 여러 명의 사람들과 함께 아래 배달음식 보기 중에서 좋아하는 음식이나 싫어하는 최근 주문기록 중 디저트									
2		4	4	4 2-4명	에너지도 고민한다	조금 고민한다	달걀비 5/ 맥북이 5/ 삼겹살 4/ 초밥 4/ 지킨 4	지킨 18		
3		5	4	3 2-4명	오래 고민한다	에너지도 고민한다	치킨5/아귀찜1/꽃미미5/맥북이5/마라탕5	맥북이 13/지킨19/맥북이		
4		5	5	5 2-4명	조금 고민한다	조금 고민한다	글쎄 5/ 지킨4/ 맥북이 5/			
5		5	3	5 혼자	에너지도 고민한다	에너지도 고민한다	치킨5/달걀정5/생선구이5/피자5/떡볶이5/마라탕5/마라탕5/제육18/돈까스14/지킨20			
6		5	5	3 2-4명	에너지도 고민한다	조금 고민한다	치킨 3/맥북이 5/피자 5/불국수 4/ 버거3	죽보 23/파스타 19/불국수		
7		5	5	5 혼자	바로 시킨다	바로 시킨다	치킨5/맥1/맥북이4			
8		5	4	4 2-4명	조금 고민한다	조금 고민한다	치킨5/달걀비4/만들기5/오로노미야끼2/파스타4/마라탕5	지킨 22		
9		5	4	4 2-4명	조금 고민한다	조금 고민한다	치킨5/맥북이4/피자4/마라탕5/돈까스3 5	얼얼 12		
10		4	4	5 혼자	에너지도 고민한다	바로 시킨다	마라탕5/치킨5/감자탕5/글쎄5/회1(배달되는 거의 맛이 5	제육볶음10/감자탕20/참		
11		5	4	5 2-4명	조금 고민한다	조금 고민한다	달걀비5/탕수육5/보리도5/맥북이5/달걀5/오물백5	지킨 18		
12		5	5	5 혼자	에너지도 고민한다	오래 고민한다	치킨 5 / 찜닭 5 / 마라탕 2 / 초밥 3 / 샌드위치 1 / 죽보	햄버거 19/ 돈까스 20		
13		5	3	5 2-4명	에너지도 고민한다	에너지도 고민한다	치킨 5 닭강정 5 보쌈 5 도시락 5 맥북이 4 피자 5	지킨 60		
14		4	3	3 2-4명	오래 고민한다	오래 고민한다	치킨 5 아귀찜 3 피자 4 햄버거 4 맥북이 5 스테이크 5	보쌈 18/ 삼겹살 19/ 지킨		
15		5	4	5 혼자	에너지도 고민한다	바로 시킨다	치킨3 5/맥북이5/초밥4 5/죽보4 5/달걀4	짜장면 11/샐러드16/말보		


<설문조사 응답 자료>

1000명의 유저데이터를 목표로 설문조사를 진행하였으며, 현재 360명의 데이터를 수집하였다. 캡스톤을 진행하는 기간동안 목표치까지 계속해서 수집할 계획이다.

3.2.2 데이터 크롤링을 통한 음식점 데이터 수집

추천알고리즘 학습에 사용될 데이터를 수집하기 위하여 주문시스템을 직접 구현하기로 했고, 주문시스템 구현에 필요한 가게 데이터를 요기요를 크롤링해서 얻어내기로 했다.

수집할 데이터는 가게 이름, 가게 위치, 가게 소개, 가게 이미지, 가게가 가지고 있는 음식리스트, 음식 이미지, 음식에대한 가격, 음식 이름 이다. 이중 가게 이미지와 음식 이미지는

 <div> 국민대학교 소프트웨어융합대학 캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

robot.txt에서 허용하지 않음을 명시해서 제외하고 크롤링을 하기로 했다.

자동으로 모든 카테고리 별로 음식점 리스트를 뽑아내고 그 음식점 리스트마다의 메뉴를 크롤링해서 DB에 가게 테이블과 메뉴 테이블의 데이터를 채워넣는 것을 목표로 하고 있다. 현재는 크롤링을 하기위한 셀레니움의 webdriver 패키지 문서를 참고하며 코드 작성중이다.

```


backend > yogiyo_crawling > crawl.py > go_to_category
1 # 기본 패키지
2 import re
3 import time
4 import pickle
5 import pandas as pd
6 from tqdm import tqdm
7 from tqdm import trange
8 import warnings
9 warnings.filterwarnings('ignore')
10
11 # 크롤링 패키지
12 import requests
13 from selenium import webdriver
14 import myslack
15
16 # 0. 페이지 한번 맨아래로 내리기
17 def scroll_bottom():
18     driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")
19
20 # 1. 입력한 위치로 설정하기
21 def set_location(location):
22     print(location+'으로 위치 설정 하는중...')
23     driver.find_element_by_css_selector('#search > div > form > input').click()
24     driver.find_element_by_css_selector('#button_search_address > button.btn-search-location-cancel.btn-search-location.btn.btn-default > span').click()
25     driver.find_element_by_css_selector('#search > div > form > input').send_keys(location)
26     driver.find_element_by_css_selector('#button_search_address > button.btn.btn-default.ico-pick').click()
27     time.sleep(2)
28     print(location+'으로 위치 설정 완료!')
29
30 # 2-1. 카테고리 페이지 Element Number Dictionary 정의
31 food_dict = { '프랜차이즈':3, '치킨':4, '피자&양식':5, '중국집':6,
32             '한식':7, '일식&돈까스':8, '족발&보쌈':9,
33             '아식':10, '분식':11, '카페&디저트':12 }
34
35 # 2-2. 카테고리 페이지로 넘어가기
36 def go_to_category(category):
37     print(category+' 카테고리 페이지 로드중...')
38     driver.find_element_by_xpath('//*[@id="category"]/ul/li[{}]/span'.format(food_dict.get(category))).click()
39     time.sleep(4)
40     print(category+' 카테고리 페이지 로드 완료!')
41
42 # 3. 카테고리의(음식점 리스트) 페이지 모두 열차기
43 def stretch_list_page():
44     restaurant_count = int(driver.find_element_by_css_selector('#restaurant_count').text)
45     scroll_count = int((restaurant_count/20))

```

3.2.3 배달음식 카테고리 분석 및 생성

배달음식 카테고리는 추천시스템에서 제공할 추천 항목으로, 배달음식 관련 논문 『사용자 선호기반 개인화 음식메뉴 추천 기법 연구』와 구글 검색을 통해 나오는 메뉴추천표에서 선정한 음식 메뉴들과 배달앱 메뉴들을 비교하여 총 81개의 카테고리를 생성하였다. 생성한 카테고리 81개는 다음과 같다.

치킨, 닭강정, 버거, 찜닭, 닭갈비, 감자탕, 삼계탕, 김치찌개, 된장찌개, 부대찌개, 순두부찌개, 육개장, 청국장,

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

설렁탕, 김치찌개, 수육, 동파육, 아귀찜, 갈비찜, 뼈해장국, 떡볶이, 김밥, 순대, 만두, 라면, 오뎅, 쫄면, 라볶이, 국수, 튀김, 돈까스, 초밥, 커리, 덮밥, 우동, 메밀소바, 회, 오키노미야끼, 라멘, 하이라이스, 나베, 짜장면, 짬뽕, 마라탕, 마라상귀, 짜장밥, 마파두부, 덩성, 탕수육, 간pong, 유산슬, 고추잡채, 유린기, 양장피, 냉면, 비빔밥, 볶음밥, 국밥, 닭볶음탕, 불고기, 갈비, 삼겹살, 제육, 떡갈비, 생선구이, 전, 찜밥, 두루치기, 계장, 족, 스파게티, 파스타, 리조또, 오믈렛, 피자, 스테이크, 바베큐, 필라프, 샐러드, 샌드위치, 보쌈, 족발, 곰창, 도시락, 닭발, 오돌뼈, 막창, 쌀국수, 브리또, 타코, 팟타이

<추천 카테고리 81개>

3.2.4 배달음식 카테고리 태그 분석 및 생성

태그는 배달음식 카테고리의 특성을 나타내는 단어로, 추천시스템의 콘텐츠 기반 추천 알고리즘에 사용될 데이터이다. 실제로 넷플릭스의 콘텐츠 추천이 태그기반 중심으로 이루어지고 있다고 하며, 영화에 관심도가 높은 사람들로 구성된 ‘태거’들이 직접 콘텐츠를 분석한 후 태그를 생성하여 부여한다. 이 프로젝트에서도 우리가 직접 카테고리에 태그를 부여하는 방식으로 진행하였다. 하지만 넷플릭스의 태거와 달리 음식에 대한 도메인 지식이 깊지 않은 관계로, 음식 카테고리에 사용되는 주재료와 대분류 카테고리만을 기준으로 태그를 생성하여 총 24개의 태그를 카테고리의 특성에 맞게 부여하였다. 생성한 태그 24개는 다음과 같다. 태그는 계속해서 수정할 계획이다.

닭고기, 돼지고기, 소고기, 김치, 된장, 햄, 두부, 생선, 떡, 밥, 면, 튀김, 감자류, 채소, 한식, 찜, 탕, 분식, 일식, 양식, 중국, 아시안, 패스트푸드, 야식

<배달음식 카테고리의 태그 24개>

3.2.5 DB 구축

주문시스템을 위한 정보들과 추천알고리즘에 사용될 정보를 포함한 DB를 설계하고 DB를 구축했다.

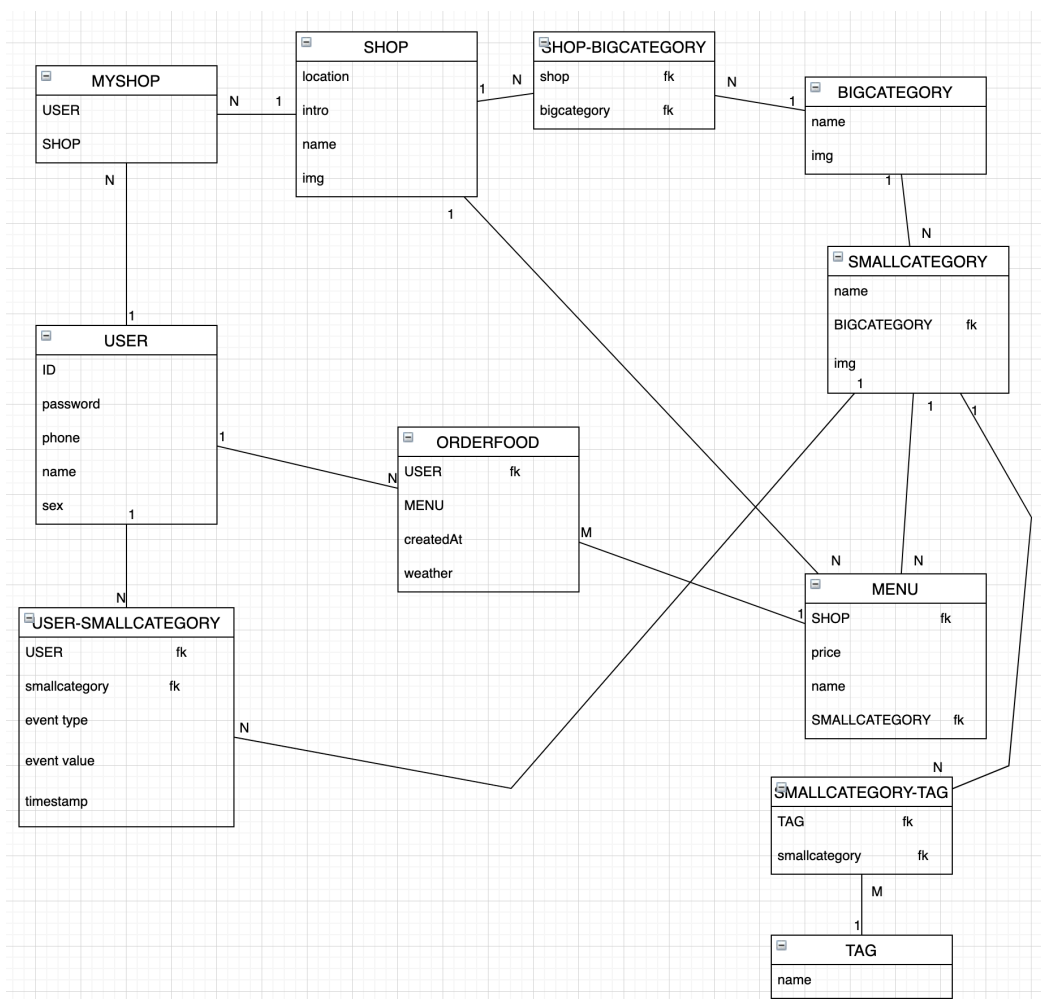
유저의 카테고리 별 선호도를 알기위한 테이블 **USER-SMALLCATEGORY** 테이블과 메뉴간의 유사도를 계산하기 위한 **SMALLCATEGORY-TAG**외에는 단순 주문시스템을 위한 DB설계이다.

첫번째 프로토타입 버전의 시스템을 위한 DB설계는 끝났으나 앞으로 추천알고리즘이 수정되거나 또


다른 추천 알고리즘을 사용하게 될 시에는 DB를 재설계 하게 될 수도있다.

DB는 MYSQL로 AWS에서 인스턴스를 할당받아서 구축했다.

아래는 특정 디비 다이어그램을 그리는 규칙에 따라 그린것이 아닌 장고의 모델 다이어그램을 직접 그린 것이다.



<Django 모델 다이어그램>

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

3.2.6 날씨 기반 추천 기능 구현

날씨 기반 추천 기능은 여러가지 추천 시나리오 중 하나로 채택하게 된 기능이다. 추천 버튼을 눌렀을 때 사용자 맞춤 추천리스트 밑에 따로 추천을 받는 시각의 날씨에 적합한 메뉴 카테고리 리스트를 제공하게 된다.

주문 데이터 수집(날씨)

구현 방법은 먼저 사용자가 주문시스템을 이용하면서 생기는 주문 기록을 전부 DB에 저장할 함과 동시에 주문하는 시각의 날씨를 같이 저장을 할것이다. 날씨는 사용자가 주문을 할때 위치 정보를 프론트에서 받아오고 서버의 시각으로 날씨 api를 이용하여 주문자 위치의 날씨를 받아올것이다.

추천 시 주문 데이터 이용(날씨)


그다음 사용자들이 추천 버튼을 누를 때 추천을 받는 시간대와 사용자의 위치를 받아와서 날씨 api를 이용하고 추천받는 사용자 주변의 날씨를 파악한다. 그 다음 해당 날씨로 주문데이터를 필터링하여 해당 날씨에 가장 많이 시킨 메뉴리스트를 가져오고 사용자에게 제공한다.

날씨 군집화

날씨 api는 openweather api를 사용할 것이다. 군집화는 기본적으로 openweather api에서 제공하는 날씨 description(ex. mist, clear sky, etc..)에 온도를 결합해서 군집화 할 예정이다.

3.2.7 시간대 기반 추천 구현

시간대 기반 추천 기능은 여러가지 추천 시나리오 중 하나로 채택하게 된 기능이다. 추천 버튼을 눌렀을 때 사용자 맞춤 추천리스트 밑에 따로 추천을 받는 시각에 적합한 메뉴 카테고리 리스트를 제공하게 된다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

자체적으로 시간대를 몇가지 군집으로 군집화했다.

오전 6시~ 11시 → 아침식사 시간대

11시 오후 3시 → 점심식사 시간대

3시 ~ 6시 → 점심 겸 저녁을 먹는 시간대

6시 ~9시 → 저녁식사 시간대

9시 ~ 오전6시 → 야식 시간대

그 다음 주문시스템 이용자들이 주문하는 시각을 주문 기록에 같이 저장을 한다음 추천을 받는 사용자들이 추천을 받는 시간대로 DB를 필터링 하여 해당 시간대에 많이 시키는 메뉴리스트를 제공하는 식으로 구현할 예정이다.


3.2.8 AWS personalize 구축

개인화 맞춤형 추천 시스템 개발에서, 추천 시스템을 직접 구현하는 목표 외에 AWS personalize 서비스를 이용하여 추천시스템을 구현하고자 하는 목표를 함께 세웠다. 이 서비스는 개별화된 추천을 손쉽게 생성할 수 있도록 도와주는 AWS 기계 학습 서비스로, 구축과정이 간단하고 고품질의 추천을 실시간으로 제공해주기 때문에 롯데마트, Pomelo 등의 기업에서도 사용하며 좋은 평가를 남겼다. 이 프로젝트에서 클라우드 기반 추천시스템을 구축한 경험을 쌓는 것과 동시에 직접 구현한 추천 알고리즘과의 성능평가를 진행하여 더욱 의미있는 프로젝트가 될 수 있도록 하고자 한다.

현재 AWS personalize 구축에 필요한 데이터를 수집하고 있으며, 서비스 사용에 대한 지식을 학습하고 있다.

3.2.9 추천 알고리즘 구현

현재까지 설계한 추천 알고리즘은 추천 시스템에서 활발히 이용되고 있는 협업필터링에 기반하고 있다. 협업 필터링 속에서도 user based와 item based, latent factor based를 활용하고 있다. 우선 데이터로 두가지 프로파일을 만들어 사용한다. 첫번째 프로파일은 사용자들이 음식 카테고리 81개에 대하여 1점(매우 싫음)에서 5점(매우 좋음)까지 선호도 평가를 한 음식

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

카테고리 평점 정보이다. 두번째 프로파일은 인구통계학적 변수 '성별, 연령'과 사용자들이 음식을 선택할 때 속성(맛, 가격, 양)에 어느 정도 중요도를 두는 지에 대한 데이터를 결합하여 생성한 프로파일이다.


이제 평점 예측 모델에 사용할 **feature**를 만들어야 한다. 첫번째 프로파일을 활용하여 사용자들간 유사도를 계산하고, 두번째 프로파일을 활용하여 사용자들간 유사도를 계산한다. 그리고 두 유사도를 합산하고 평균내어 최종 유사도를 산출한다. 이렇게 유사한 사용자 **top 5**가 특정 음식 카테고리에 남긴 평점 **feature**를 생성한다. 그리고 첫번째 프로파일을 활용하여 유사한 카테고리 **top 5**의 평균 평점을 나타내는 **feature**를 생성한다. 마지막으로 **global average**를 나타내는 3가지 **feature**를 만든다. 이렇게 13가지 **feature**를 이용하여 **ensemble boosting model**인 **XGBoost**를 학습시키고 평점을 예측할 것이다.

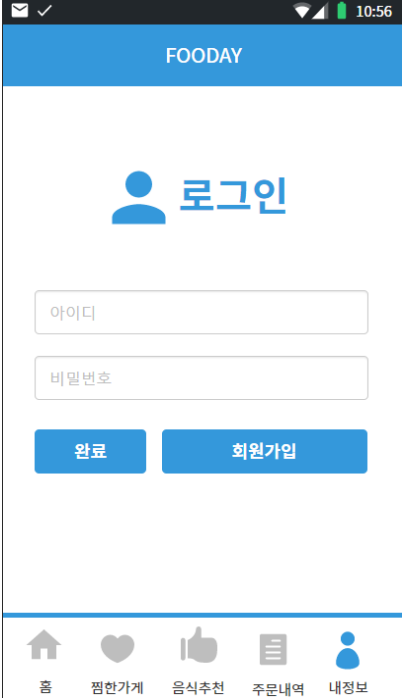
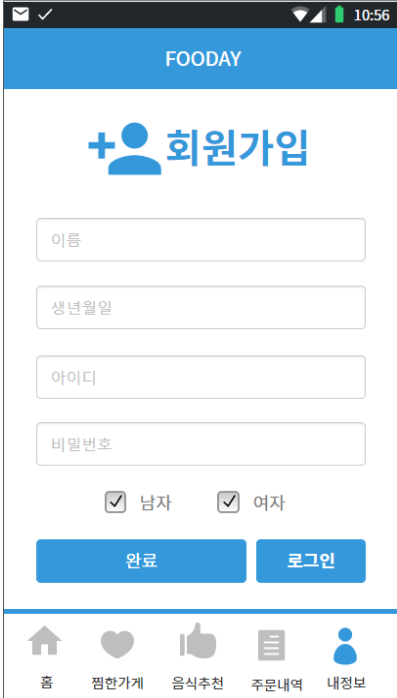
여기까지의 알고리즘은 사용자의 과거 선호만을 반영할 수 있다. 선호는 변화할 수 있기 때문에 주문횟수 데이터를 사용하여 변화하는 선호도 고려해 보려고 한다. 주문횟수 데이터를 **Matrix Factorization**로 학습시킨 결과와 위 **XGBoost** 모델의 결과를 앙상블하여 음식 추천을 해보려고 한다. 앙상블 방법은 아직 고려 중에 있다.

그리고 또 현재 고려하고 있는 점은 음식 카테고리 태그를 활용한 **content based** 알고리즘을 추가하여 하이브리드 추천 시스템으로 만드는 것과 추천 결과 피드백의 적용이다. 이것들을 이해하고 적용시키기 위해 관련 논문을 찾아보고 있는 중이다.

3.2.10 Application(using React Native)

- 와이어프레임 제작
총 4차례에 걸쳐 수정했으며, 각각의 수정사항은 **scrum board**에 링크와 함께 공유해 팀원들이 편리하게 확인할 수 있게 했다.


 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

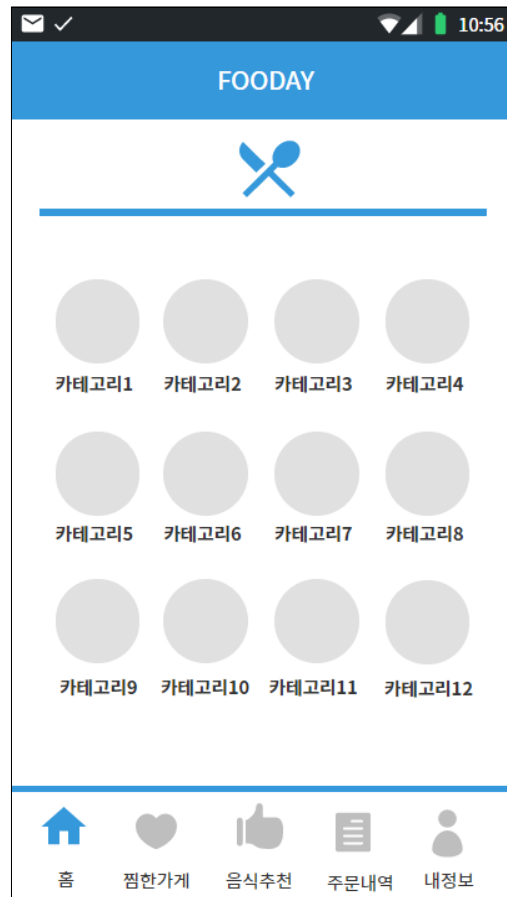



<로그인 화면>

<회원가입 화면>


로그인과 회원가입 기능은 맞춤형 추천을 이용하기 위한 필수적인 요소이다. 네비게이션 바에서 내정보 화면으로 이동하려면 로그인이 되어있어야 하기 때문에 로그인이 되었지 않다면 로그인 화면으로 넘어가고 계정이 없는 경우를 대비해 회원가입 화면으로 넘어갈 수 있는 버튼을 같이 제공해준다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05



<메인 홈 화면>

메인 홈 화면은 추천 서비스를 이용하지 않고도 카테고리별 음식과 음식점 리스트를 볼 수 있는 화면이다. 직관적인 디자인으로 한 눈에 파악하기 쉽다.

 <div> 국민대학교 소프트웨어융합대학 캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05




<설문1 화면>



<설문2 화면>


맨 처음 음식추천을 이용하려고 할 때 초기 데이터가 없는 사용자의 취향을 파악할 수 있도록 설문을 진행한다. 설문1 화면에서 10개 이상의 음식을 선택해 점수를 입력하고 완료를 누르면 설문2 화면으로 넘어가게 되고 완료를 누르면 음식추천을 받을 수 있게 된다.

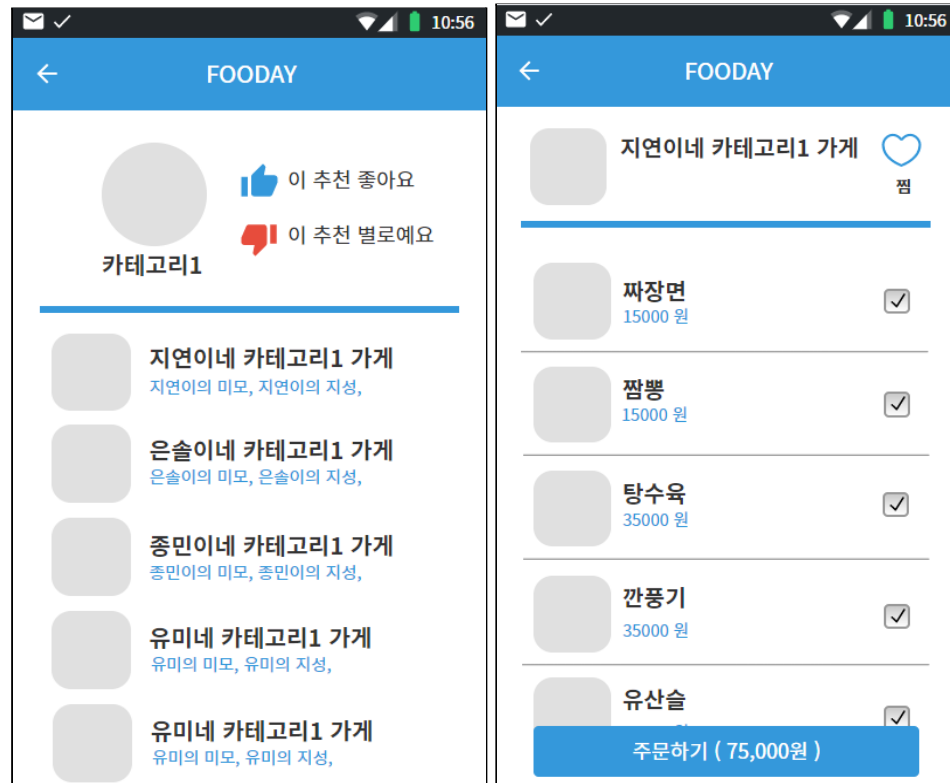
 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05



<음식 추천 화면>

위 화면은 본 프로젝트에서 가장 핵심으로 구현하고자 하는 음식 추천 화면이다. 총 3가지의 시나리오(나한테 딱 맞는, 시간대별 음식, 지금 날씨에 어울리는 음식)로 음식 카테고리를 추천해주는 방식이며 각각의 추천된 카테고리를 클릭하면 관련 음식점 리스트를 볼 수 있다.


 <div> 국민대학교 소프트웨어융합대학 캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05



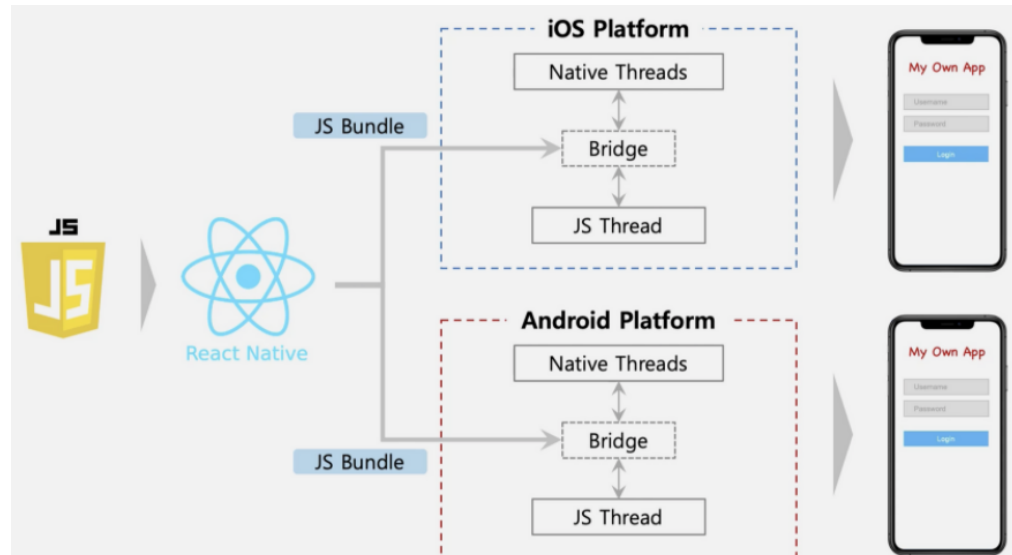
<음식점 리스트 화면>

<음식점 내 메뉴 리스트 화면>

카테고리를 클릭했을 때 나오는 음식점 리스트 화면에는 더욱 정확한 추천을 위해 실시간으로 피드백을 할 수 있는 버튼을 만들었고, 카테고리화 관련된 음식점들을 확인할 수 있게 리스트로 보여준다. 리스트에 있는 음식점들 중 하나를 클릭할 경우 음식점의 메뉴들을 보여주고 선택한 후 바로 주문기능과 이어지게 제작했다. 또한 가게 찜하기 기능이 있어서 가게를 찜해놓으면 네비게이션 바에 있는 찜한 가게를 통해 원하는 가게만 볼 수 있다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

- 개발환경 setting



< 개발환경 기본원리 >

APP의 전체 로직을 가지고 있는 **JS Bundle**을 만들어서 각 플랫폼에 심어주는 것이 **React Native**의 기본원리이다.

JS Bundle은 **JS Thread**에 의해서 실행되는데 이때 각 플랫폼의 앱을 실행하기 위한 **Native Thread**는 **JS Thread**와 직접 커뮤니케이션을 할 수 없고, 이 둘의 상호작용을 가능하게 하는 것이 바로 **React Native**에서 제공하는 **Bridge**이다.


JS Bundle을 만드는 두가지 방법 중 하나인 **Expo CLI**를 사용하기로 했는데 실제 개발이 쉽고 편하면서 개발환경 구축이 용이하다는 장점이 있다.

- 개발 컨벤션 정하기

디자인 및 프론트의 수행내용을 2명의 인원이 각각 역할을 분담해 진행하였다. 따라서 통일화된 코드와 개발을 위해 개발 컨벤션을 정한 뒤 작업을 진행하기로 하였고 효율적인 **state**관리를 중점적으로 한 상의 결과는 다음과 같다.

1. arrow function 사용하기
2. hook api 사용하기
3. context api 사용하여 상태관리하기

- 네비게이션바 라우팅 구현.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

```

import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

import { NavigationContainer } from '@react-navigation/native';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';

import Main from './components/main';
import Recommend from './components/recommend';
import MyOrder from './components/myorder';
import MyProfile from './components/myprofile';
import MyStore from './components/mystore';

const Tab = createBottomTabNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Tab.Navigator
        initialRouteName = "main"
      >
        <Tab.Screen name= "main" component = {Main}/>
        <Tab.Screen name= "mystore" component = {MyStore}/>
        <Tab.Screen name= "recommend" component = {Recommend}/>
        <Tab.Screen name= "myorder" component = {MyOrder}/>
        <Tab.Screen name= "myprofile" component = {MyProfile}/>
      </Tab.Navigator>
    </NavigationContainer>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
  },
});

```

```

components
  JS main.js
  JS myorder.js
  JS myprofile.js
  JS mystore.js
  JS recommend.js

```

<네비게이션 바 구현 코드>


<components>

가장 먼저 화면 전환 및 이동이 가능하도록 하는 것이 우선이라 생각하였고, 이를 구현하는데 있어 **react navigation** 모듈을 사용하였다. 네비게이션바를 구성하는 각각의 화면들은 **component**로 나눈 뒤 라우팅하였다

- github 협업

두명에서 역할 분담 후 작업이기 때문에 각각의 진행상황을 공유하고 협업하는 도구로 **github**를 사용하였다. 두 팀원의 작업 방식은 다음과 같다

1. 캡스톤 **repository** 안에 **front** 브랜치를 생성한다
2. 개발환경 **setting**이 끝난 프로젝트를 **front** 브랜치에 올린다
3. 각자 작업할 브랜치 생성한다
4. 프로젝트를 **clone**받은 후 각자의 브랜치에 **push**한다
5. 작업후 **add, commit, push** 반복한다

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

6. push 이후 새로운 변경사항이 있을 수 있으니 pull 먼저 받고 시작한다
7. front로 pull request와 merge 요청한다
8. conflict 등 확인 후 merge한다

작업시에 추가적인 사항으로 root는 변경하지 않기로 했다. 또한 프로젝트 생성시 기본적인 Main으로 제공되는 App.js는 conflict방지 등의 이유로 최대한 변경하지 않는 방향으로 진행하였다.

4 수정된 연구내용 및 추진 방향

4.1 수정사항

3.1.1 아이디어 변경


본 프로젝트를 진행하기 이전에 수어 관련 프로젝트를 먼저 기획했었는데 기술적인 부분이 아닌 현실적인 문제로 인해 3월 중순쯤 급하게 아이디어를 변경하게 됐다. 시간이 많이 남지 않은 상황인만큼 가장 효율적으로 프로젝트를 진행할 수 있게 스프린트를 정해 계획을 짜고 수행하는 방식을 사용했으며 매일 회의를 통해 꾸준히 피드백을 주고받았다.

진행중 6	피드백 전 3	완료 13
추천페이지 틀 구현 유미 차 스프린트1 2021년 3월 25일 오후 11:19 프론트	추천 두종류 유즈케이스, 시퀀스 다이어그램 그리기 정지연, 종민 박 스프린트1 2021년 3월 22일 오전 9:57 백엔드	스프린트 1 - 와이어프레임 작성 유미 차, 종민 박 스프린트1 2021년 3월 22일 오후 11:14 프론트
추천 알고리즘 설계 은솔 최, 정지연 스프린트1 2021년 3월 31일 오후 9:00 백엔드	와이어프레임4차 수정 유미 차 스프린트1 2021년 4월 2일 오후 11:04 프론트	카테고리 생성 정지연 스프린트1 2021년 3월 23일 오후 7:59 백엔드

<scrum board>

3.1.2 user data collection 방식 수정

처음 계획했던 유저 데이터 수집 방식은 기존 어플리케이션에 요청해 주문기록에서 필요한

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

데이터만 받아오는 것이었다. 그러나 개인정보 보안상의 문제로 인해 데이터를 받아올 수 없는 관계로 설문 폼을 직접 구현해 설문조사를 하기로 했다.

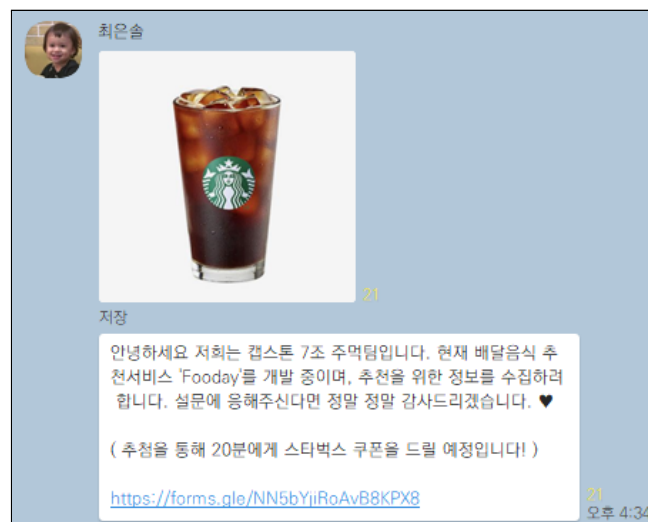
배달음식 수요조사

안녕하세요! 국민대학교 캡스톤 7조 주먹팀입니다.
저희는 현재 배달음식 추천서비스 'Fooday'를 개발중이며, 추천을 위한 다양한 정보를 수집하려 합니다.

설문조사가 끝난 후 맨 아래 번호를 적어주시면 추첨을 통해 20분에게 스타벅스 아메리카노 쿠폰을 드릴 예정이니 많은 참여 부탁드립니다!

*** 필수항목**


<설문 폼을 이용한 데이터 수집>




<SNS를 이용한 설문조사 참여 유도>


3.1.3 추천 알고리즘 구현방식 수정

초기에는 AWS에서 제공하는 AWS personalize 서비스를 사용해 추천 시스템을 구현하려고

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

계획했었다. 그러나 본 프로젝트인 ‘배달음식 추천 시스템’의 핵심은 추천 알고리즘 구현으로 **aws**에서 제공되는 알고리즘이 아닌 직접 구현한 알고리즘을 사용하여 서비스를 만드는 것이 의미가 있을 것 같다는 의견이 나왔으며 또 한편으로는 이미 제공되어 있는 **aws personalize** 서비스 역시 사용해 보는 것이 좋을 것 같다는 의견도 있었다. 또한 두 가지 방법 모두를 사용자에게 제공하기에는 의미적으로 크게 다르지 않다는 생각이 들어 이와 같은 방향성과 모두 구현하게 됐을 경우에 사용자에게 다르게 보여줄 수 있는 방법에 대해 조언을 얻고자 교수님께 문의를 드렸다.


황선태 교수 5일 전
 이런 경우는 두가지 방법을 비교하면 좋을 것 같은 생각이 드네요. 즉 서비스는 직접 구현한 것으로 구축하는데 백엔드에서는 두가지 추천 시스템이 동시에 동작하도록 하여 비교하는 것입니다. 이 비교 결과는 보고서에서 다루면 되겠지요.



허대영 교수 5일 전
 프로젝트의 남은 시간의 문제이기도 합니다. 짧은 시간에 어떤 구현 전략을 쓸 것 인지 결정이 필요한 사항이네요.

1. **aws personalize** 를 선택하는 경우는, 전반적인 시스템의 완성도를 먼저 구축하고자 하는 경우라고 생각해볼 수 있습니다. 이 후, 다른 여건이 해결 된 후, **aws personalize** 부분을 자체 개발한 것으로 변경해나가는 과정이 있을 수 있습니다.
2. 사용자 추천 시스템을 선택하는 경우는, 말 그대로 시스템의 완성도보다는 사용자 추천 시스템에 대한 연구를 우선적으로 해야 할 때, 고려해볼 수 있습니다. 이 추천 시스템이 여러분만 가지고 있는 고유한 것이라면, 이 방법도 좋은 선택입니다.
3. 개발 인력이 충분하다면 1,2를 동시에 취할 수도 있습니다. 방법의 1의 전단계와 방법 2를 동시에 진행하고, 방법1의 전단계가 완료 된 후, 방법2의 내용을 차근차근 적용해가는 전략입니다.

여러분이 생각하고 있는 5월 평가에서 평가자들에게 보여줄 소프트웨어의 모습이 어떤 것인가에 따라, 개발 전략을 선택해야 할 것으로 보입니다. 1,3번 전략을 취한 경우, 개발 속도가 충분하다면, @황선태 교수 의 말씀처럼 두 추천 시스템을 비교해볼 수 있고, 이것도 매우 유익한 경험이 될 것입니다. (편집됨)

<slack을 통한 질의응답>

결과적으로는 둘 다 사용하여 서비스를 구현해보기로 했다. 사용자들에게 보여지는 것은 크게 다르지 않겠지만 두 추천 서비스를 개발하는 과정에 있어서 두 기능을 서로 비교하고 분석하며 연구하는 것이 팀원들에게도 큰 도움이 되고, 본 프로젝트를 더욱 가치있게 할 수 있는 방향이라고 생각해서 결정을 내리고 각자 맡을 역할을 분담했다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

5 향후 추진계획


5.1 향후 계획의 세부 내용

5.1.1 다수를 위한 공통 추천리스트 계산 방법 개선

현재, 추천알고리즘의 결과로 나오게 되는 유저의 카테고리별 **rating** 점수를 이용하여 함께 추천받고 싶은 유저들의 카테고리별 **rating** 합의 평균으로 추천리스트를 제공하기로 계획했다.. 이 부분을 더 나은 방법을 통해 정확한 다수의 추천시스템을 구현할 계획이다.

5.1.2 Application 구현

와이어프레임으로 제작했던 화면들을 실제 **APP**으로 구현해낼 예정이다. 기본 틀은 **React Native + Expo CLI** 로 구현할 것이고 비동기 통신 라이브러리인 **Axios**를 사용하여 서버와 통신하며 데이터를 주고받을 예정이다. 이는 **json** 파싱 또는 **stringfy**의 번거로움 없이 손쉽게 셋팅하는 부분을 도와준다.

 국민대학교 소프트웨어융합대학 캡스톤 디자인 I	중간보고서		
	프로젝트 명	FOODAY	
	팀 명	주먹	
	Confidential Restricted	Version 1.5	2021-04-05

6 고충 및 건의사항

1. 추천 알고리즘에 적용될 데이터

1-1. 추천 알고리즘을 구현하는 과정에서 필요한 데이터를 수집하기 위해 기존 배달 어플에 주문기록 데이터를 요청했지만 개인정보 보안상의 이유로 제공받지 못했다.

1-2. 1-1과 같은 이유로 직접 설문 폼을 작성해 데이터를 수집해야 했고, 많은 양의 데이터가 필요한 상황에서 인원수를 충족하는데 어려움이 있었다.

2. 음식 카테고리 분류

기존에 서비스에서 제공되던 카테고리화 카테고리별 음식의 종류가 다양하고 그 수가 많았다. 본 프로젝트에서 더 정확한 결과를 도출해내기 위해 카테고리를 효율적으로 재분류하는 과정에서 많은 고민이 필요했다.