

산학연계웹개발 프로젝트 최종보고서

팀 명	NeighborFood			
작 품 명	NeighborFood			
개발기간	2023 년 3 월 2 일 ~ 2023 년 6 월 5 일			
지도교수	컴퓨터공학과 윤 현 주			
참여기업	유라클			
구 분	학년	학 번	성 명	휴대전화
책임자(팀장)	졸업생	20190952	이혜원	01094345618
팀원1	휴학생	20181179	천예진	01056677728
팀원2	4	20181037	전민규	01085444313
팀원3	4	20200639	신승미	01072000776
팀원4	4	20200552	백지원	01025028130

본인은 산학연계웹개발프로젝트 최종보고서를 첨부와 같이 제출합니다.

2023년 06월 13일

팀 장 신승미 (서명)

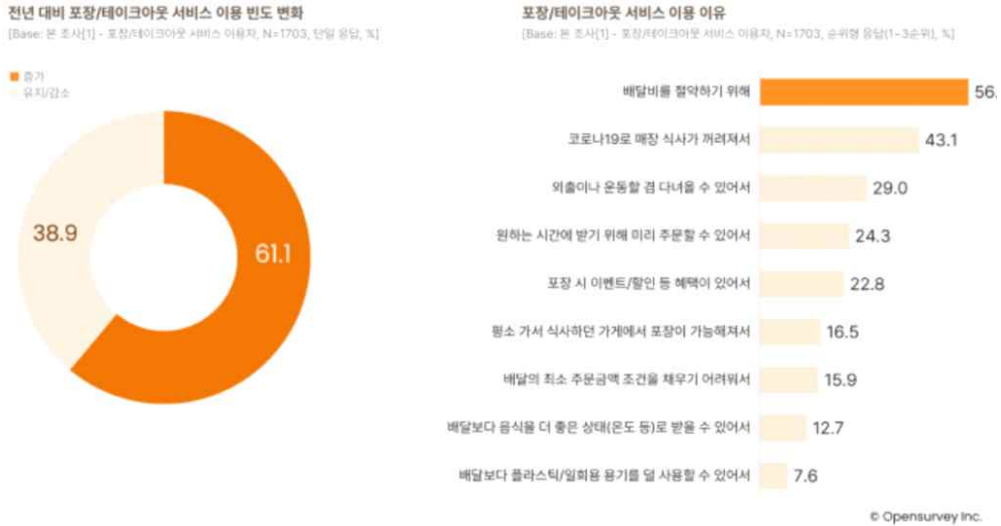
웹 개발 실무인재 양성 과정
금오공과대학교 AI·SW교육센터

프로젝트 최종보고서

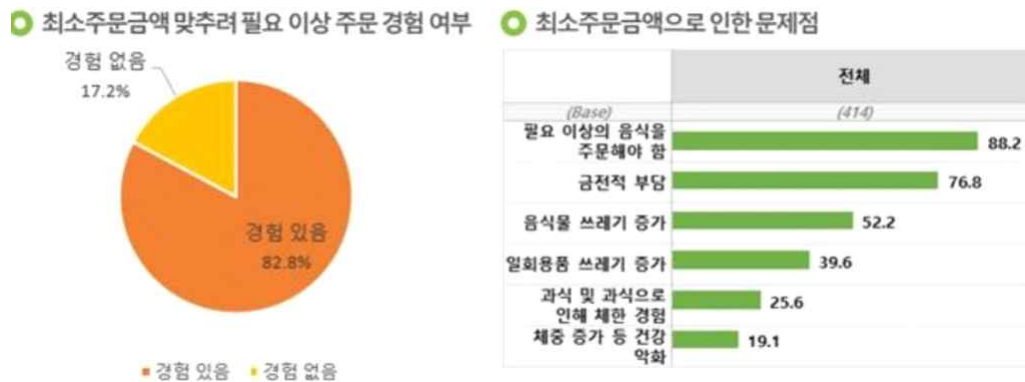
NeighborFood

1. 서론

1.1 배경 설명



[그림 1] 포장/테이크아웃 서비스 이용 빈도 변화와 이용 이유



[그림 2] 최소주문금액 맞추려 필요 이상 주문 경험 여부와 이로 인한 문제점



[그림 3] 배달료 부담 정도

최근 높아지는 배달료에 따라, 소비자들은 '배달의민족', '요기요' 같은 배달 앱에서의 배달 음식 소비가 예전보다 부담스러워지고 있다. 특히 혼자 배달 음식을 주문해 먹는 사람은 배달료를 혼자 부담해야 하기 때문에 더욱 그렇다. 게다가 최소 주문금액까지 있기 때문에 혼자서는 다 먹지 못할 양의 배달 음식을 주문하는 경우도 종종 있다.

커져가는 소비자들의 불만에 따라 우리는 금오공과대학교 학생들끼리 같이 음식을 시켜 배달받을 사람을 모집하여 최소 주문금액도 넘기고 배달료에 대한 부담을 줄 수 있는 앱을 만들고자 프로젝트를 기획하게 되었다.

1.2 개발 동기

코로나 이후로 많은 사람들이 배달 앱 사용하기 시작했다. 개인적으로 인원을 모아야 하는 시간을 단축할 수 있고 최소 배달 금액을 함께 채울 수 있는 인원을 모집하여 배달비에 대한 부담을 줄이는 애플리케이션이 필요할 것 같아 만들게 되었다.

1.3 개발 목표

1.3.1 배달 음식에 대한 부담을 줄이는 시스템

- 함께 배달 음식을 시키면서 최소 배달 금액 및 배달비에 대한 부담을 줄일 수 있다.

1.3.2 학생들이 안전하게 이용할 수 있는 시스템

- 금오공과대학교 이메일 한정으로 인증할 수 있도록 하여 학생들만 사용함으로써 외부인으로 인한 위험 상황을 방지할 수 있다.

1.3.3 간편하고 소통하기 편리한 시스템

- 댓글 기능을 이용한 방장과 참가자들의 소통창
- 모피어스 API(Location API)를 이용하여 사용자가 원하는 위치를 배달주소로 입력 가능

1.4 기술현황

1.4.1 배달의 민족의 '함께 주문'

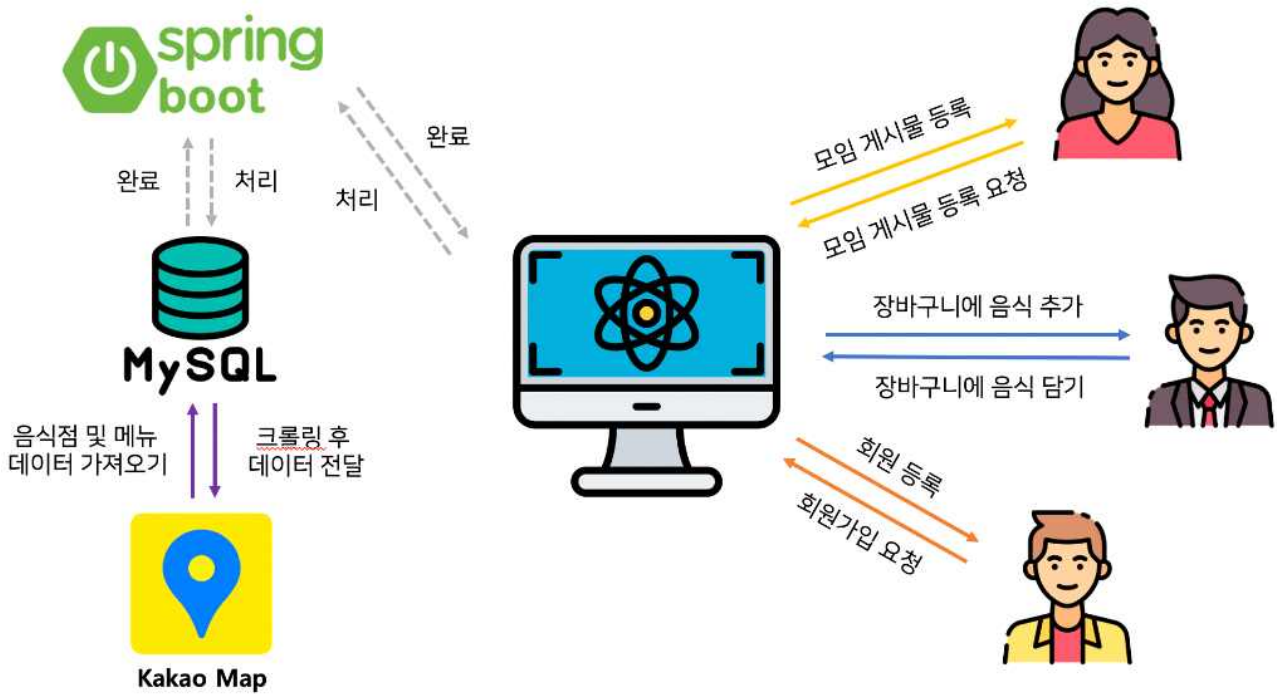
배달의민족은 배달비를 나눠 결제하는 공동 구매 시스템 '함께 주문' 서비스를 도입했다. '함께 주문'은 여러 사람과 장바구니를 공유해 각자 원하는 메뉴를 담은 다음 한 곳에서 수령하는 서비스다.

1.4.2 쿠팡이츠의 '친구 모아 함께 주문'

쿠팡이츠는 각자 주문하되 같은 장소에서 수령하는 '친구 모아 함께 주문' 서비스를 진행 중이다.

2. 작품 개요

2.1 전체 구성도



[그림 1] NeighborFood 시스템 구성도

<설계구성요소 및 설계제한 요소>

설계 구성요소						설계 제한요소						
목표 설정	합성	분석	구현/ 제작	시험/ 평가	결과 도출	성능	규격/ 표준	경제성	미학	신뢰성	안정성/ 내구성	환경
○		○	○	○	○	○	○			○	○	

2.2 설계구성요소

2.2.1 목표 설정

공동 배달 애플리케이션을 개발하여 배달시킬 음식점, 배달받을 위치 등을 공유하여 주변에 있는 사람들과 함께 배달시켜 배달비에 대한 부담을 줄이도록 한다.

2.2.2 분석

개발할 시스템의 기능은 다음과 같다.

- 회원관리 : 회원가입, 로그인, 로그아웃, 회원정보수정, 회원탈퇴
- 모임 게시물 관리 : 모임 게시물 등록/삭제/조회, 댓글 등록/삭제/조회
- 공동 장바구니 관리 : 장바구니 메뉴 등록/수정/조회, 장바구니 등록/삭제
- 음식점 관리 : 음식점 조회, 메뉴 조회

2.2.3 구현/제작

데이터베이스를 정규화 규칙에 기반하여 제작한다.

2.2.4 시험/평가

앱의 기능과 성능을 평가하기 위하여 평가자에게 시연동영상을 보여주어 평가를 받는다.

2.2.5 결과도출

최종 결과물은 웹과 앱 환경에서 모두 사용할 수 있는 하이브리드 앱이다.

2.3 설계제한요소

2.3.1 성능

작품의 성능 요소와 목표로 하는 성능은 다음과 같다.

- 동시접속자 수: 100명
- 메모리 사용량: 2GB
- 실행 속도: 불편함을 느끼지 않을 속도

2.3.2 규격/표준

사용하는 테이블은 다음과 같다.

- MEMBER : 학교 인증 이메일, 비밀번호, 이름, 푸쉬 이메일, 사용 은행, 계좌번호에 대한 정보를 저장한다.
- REPLY : 댓글 내용, 댓글을 등록한 날짜에 대한 정보를 저장한다.
- BOARD : 게시물의 제목, 내용, 카테고리, 경도, 위도, 주소, 상세주소, 주문 시간, 최대 인원, 등록 날짜, 수정 날짜에 대한 정보를 저장한다.
- BASKET : 게시물 번호, 이메일, 메뉴 번호에 대한 정보를 저장한다.
- RESTAURANT : 이름, 배달팁, 최소주문금액에 대한 정보를 저장한다.
- MENU : 메뉴 이름, 가격, 마지막 업데이트 날짜/시간에 대한 정보를 저장한다.

2.3.3 신뢰성

사용자가 입력한 내용과 서버의 관리 내용이 항상 일치해야 한다. 사용자의 입력이 잘못된 경우나 고의적인 오류 입력의 경우에도 적절한 오류 처리를 통해 데이터 무결성과 신뢰성이 보장되도록 한다.

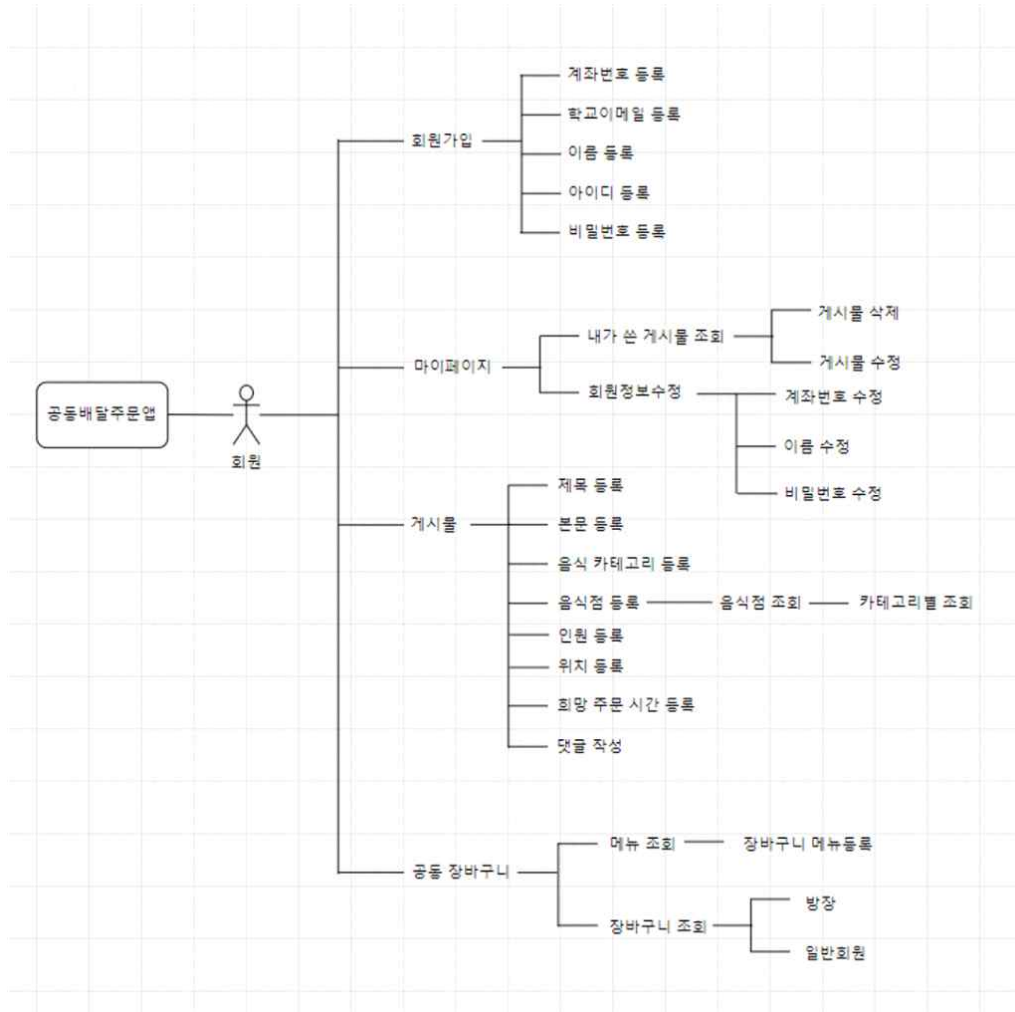
2.3.4 안전성/내구성

사용자의 입력이 잘못된 경우나 고의적인 오류 입력의 경우에도 적절한 오류 처리를 통해 앱 실행이 중단되지 않는다. 약 3시간 정도의 장시간 동작에도 이상 현상이 없도록 해야 한다.

3. 개발 내용

3.1 상세 설계 내역

3.1.1 마인드맵



[그림 2] 마인드맵

3.2 요구사항

3.2.1 회원관리

비즈니스 요구사항		기능 요구사항		행위자 (Actor)
식별 자	요구사항 명	식별자	요구사항명	
S1-B0 1	회원 관리	S1-R11	회원가입	비회원
		S1-R12	로그인	회원
		S1-R13	로그아웃	회원
		S1-R14	회원 정보 수정	회원
		S1-R15	회원 탈퇴	회원

[그림 3] 회원 리관 기능 요구사항

3.2.2 모임 게시물 관리

S1-B0 2	모임 게시물 관리	S1-R21	모임 게시물 등록	회원
		S1-R22	내가 쓴 모임 게시물 수정	회원
		S1-R23	내가 쓴 모임 게시물 삭제	회원
		S1-R24	모임 게시물 조회	회원
		S1-R25	내가 쓴 모임 게시물 조회	회원
		S1-R26	모임 게시물 댓글 등록	회원
		S1-R27	내가 쓴 모임 게시물 댓글 수정	회원
		S1-R28	내가 쓴 모임 게시물 댓글 삭제	회원
		S1-R29	모임 게시물 댓글 전체 삭제 (해당 모임 게시물 삭제 시)	시스템
		S1-R21 0	모임 게시물 댓글 조회	회원
		S1-R21 1	내가 쓴 모임 게시물 댓글 조회	회원

[그림 4] 모임 게시물 관리 기능 요구사항

3.2.3 공동 장바구니 관리

S1-B0 3	공동 장바구니 관리	S1-R31	장바구니 메뉴 등록	회원
		S1-R32	장바구니 메뉴 수정	회원
		S1-R33	장바구니 메뉴 조회	회원
		S1-R34	장바구니 등록 (해당 모임 게시물 등록 시)	시스템
		S1-R35	장바구니 삭제 (해당 모임 게시물 삭제 시)	시스템

[그림 5] 공동 장바구니 관리 기능 요구사항

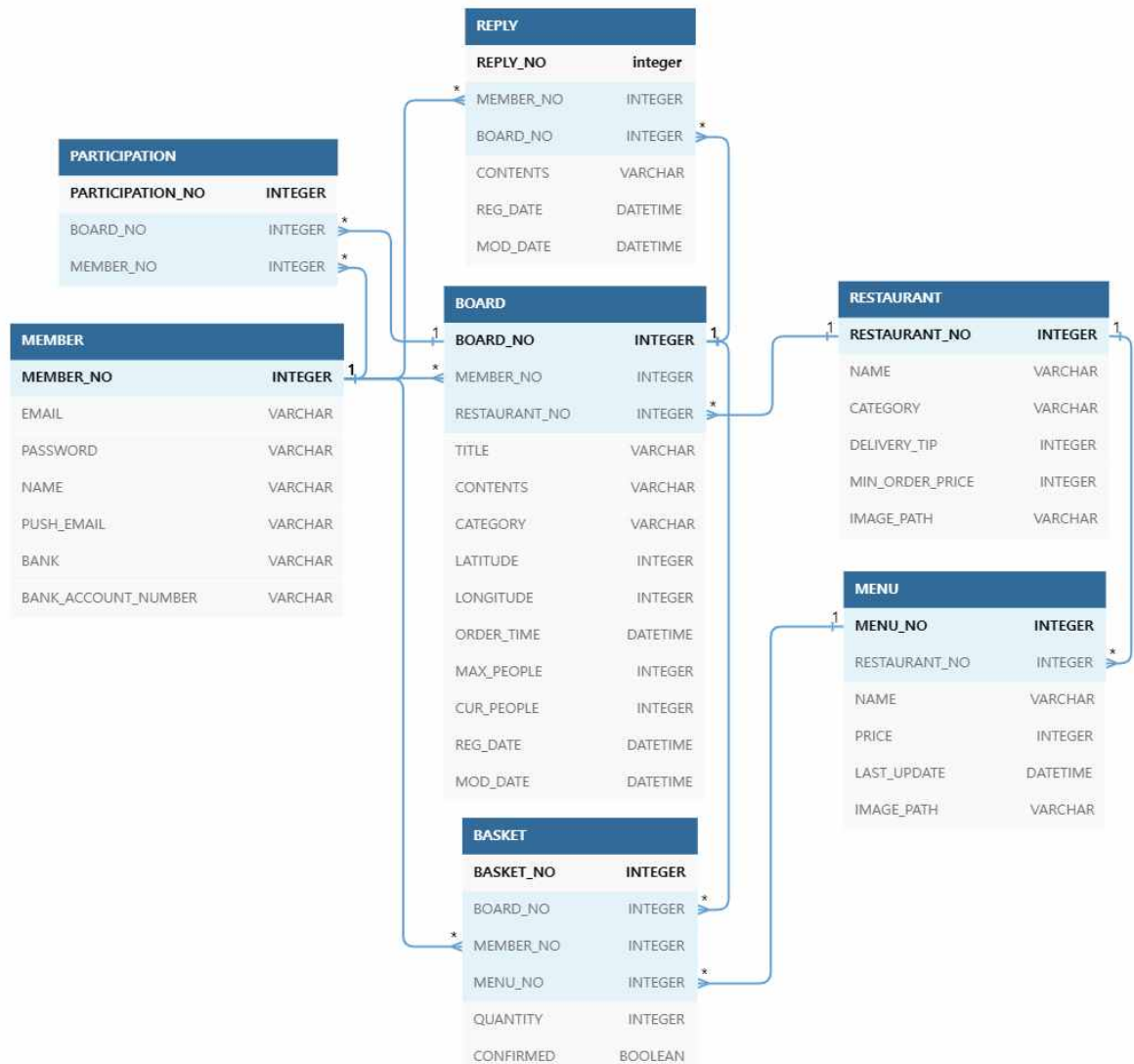
3.2.4 음식점 관리

S1-B0 4	음식점 관리	S1-R41	음식점 정보 조회	회원
		S1-R42	메뉴 조회	회원

[그림 6] 음식점 관리 기능 요구사항

3.3 논리 설계

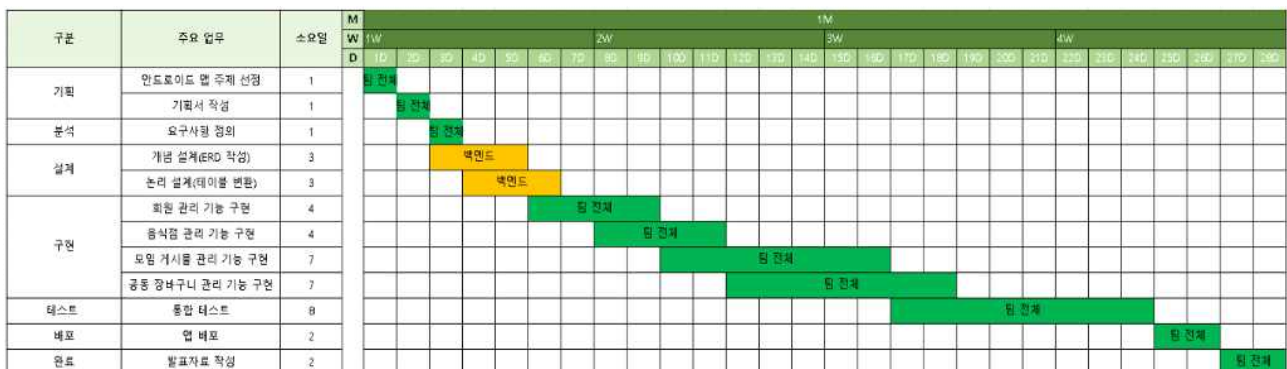
3.3.1 Entity Relationship Diagram (ERD)



dbdiagram.io

[그림 7] Entity Relationship Diagram

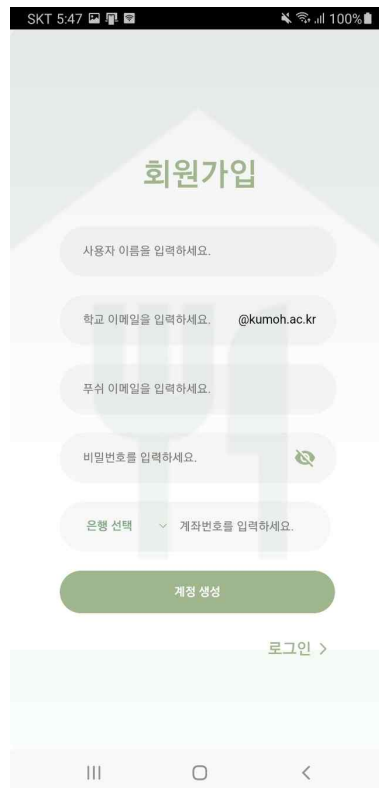
3.3.2 개발 일정(WBS)



[그림 8] 개발 일정

3.4 화면 설계

3.4.1 회원 관리 화면



이 화면은 회원가입을 위한 폼입니다. 상단에는 '회원가입'이라는 제목이 있습니다. 폼에는 사용자 이름, 학교 이메일, 알림 이메일, 비밀번호, 은행 선택, 계좌번호를 입력하는 필드가 있습니다. '로그인' 링크도 포함되어 있습니다.

SKT 5:47 100%

회원가입

사용자 이름을 입력하세요.

학교 이메일을 입력하세요. @kumoh.ac.kr

알림 이메일을 입력하세요.

비밀번호를 입력하세요.

은행 선택 계좌번호를 입력하세요.

계정 생성

로그인 >

[그림 9] 회원가입 화면

회원가입 화면에서 이름, 학교 이메일, 알림 이메일, 비밀번호, 은행, 계좌번호를 입력한다. 회원가입 시 학교 이메일로 인증 메일이 전송된다.



이 화면은 로그인 폼입니다. 상단에는 '로그인'이라는 제목이 있습니다. 폼에는 학교 이메일, 비밀번호를 입력하는 필드가 있습니다. '로그인' 버튼과 '회원가입' 링크도 포함되어 있습니다.

SKT 5:46 100%

로그인

학교 이메일을 입력하세요. @kumoh.ac.kr

비밀번호를 입력하세요.

로그인

회원가입 >

[그림 10] 로그인 화면

로그인 화면에서 사용자가 회원가입한 학교 이메일과 비밀번호를 바탕으로 로그인 한다.



[그림 11] 마이페이지 화면

마이페이지에서 학교 인증을 클릭하면 학교 이메일로 인증 메일이 전송된다. 그 외 마이페이지에서 내가 참여한 게시물, 회원정보수정, 로그아웃, 회원탈퇴의 기능을 수행할 수 있다.



[그림 12] 회원정보수정 화면

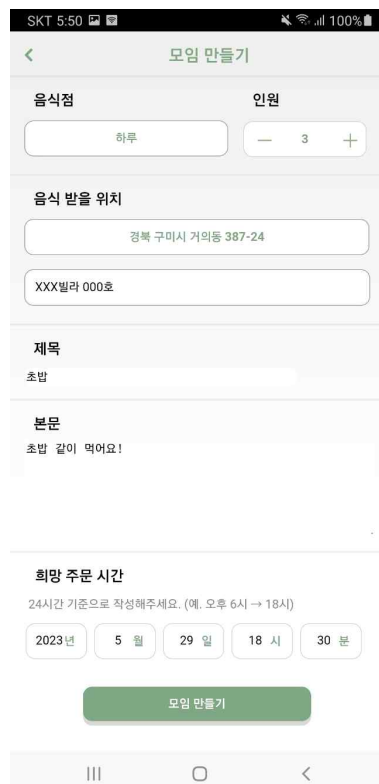
이름, 이메일, 비밀번호, 계좌번호를 수정할 수 있다.



[그림 13] 내가 참여한 게시물 화면

내가 참여한 게시물에 이동하거나, 모임에서 탈퇴할 수 있는 모임 참여 취소가 가능하다.

3.4.2 모임 게시물 관리 화면



[그림 14] 모임 만들기 화면

음식점, 인원 수, 음식 받을 위치, 제목, 본문 내용, 희망 주문 시각을 입력하여 모임을 만들 수 있다.



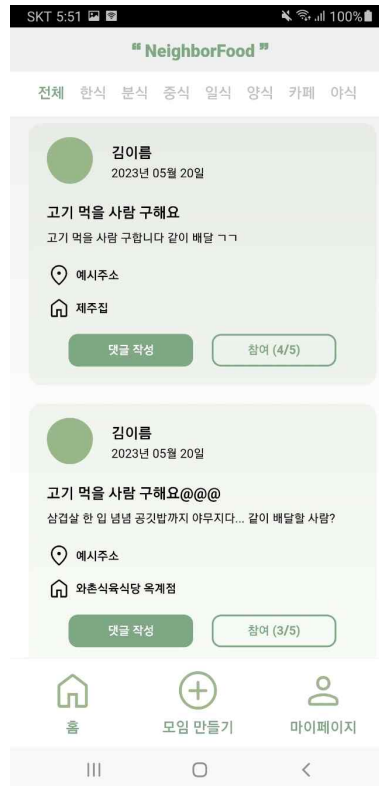
[그림 15] 모임 만들기 - 위치 선택하기 화면

지도의 마커 이동을 통해 위치를 선택할 수 있다. 마커를 이동하지 않을 시 현재 위치가 적용된다.



[그림 16] 모임 만들기 - 음식점 선택하기 화면

주문하고자 하는 음식점을 선택할 수 있다.



[그림 17] 홈 화면

홈 화면에서는 사람들이 올린 게시물을 카테고리 별로 조회할 수 있다. 참여 버튼을 누르면 모임에 참여할 수 있고, 댓글 작성 버튼을 누르면 해당 게시물에 들어갈 수 있다.



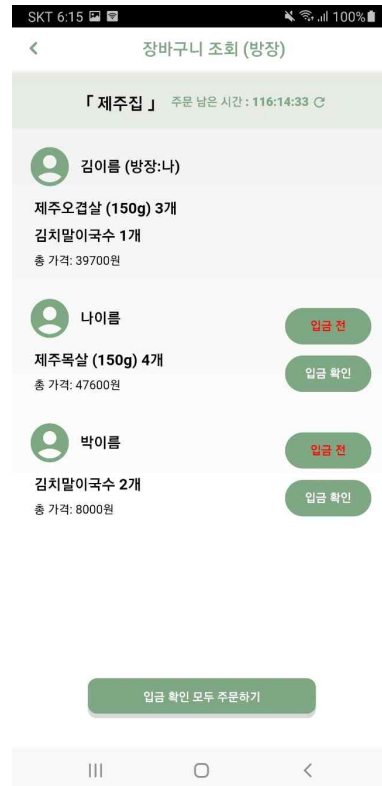
[그림 18] 모임 참가 및 댓글 작성 화면



[그림 19] 알림 이메일로 댓글 알림 푸시 화면

모임에 참여하면 댓글을 작성하여 모임에 참여한 사용자와 소통한다. 본인이 작성한 댓글 및 게시글은 삭제할 수 있다. 본인이 참여한 모임 게시물에 댓글이 달리면 모임에 참여한 모두에게 알림 이메일로 푸시 알림이 가게 된다.

3.4.3 장바구니 관리 화면



[그림 20] 방장 버전 장바구니 조회 화면

각 인원이 입금을 완료하면 방장이 입금 확인 버튼을 누를 수 있다. 입금 확인 버튼을 잘못 눌렀을 경우 확인 취소도 가능하다.



[그림 21] 참가자 버전 장바구니 조회 화면

방장에게 입금이 확인되면 '입금 후' 라고 뜨고, 입금하기 전이면 주문 취소가 가능하다.



[그림 22] 장바구니 메뉴 조회 화면

해당 음식점의 메뉴를 조회할 수 있다. 선택하면 메뉴 상세 조회가 가능하며 장바구니에 담을 수 있다.



[그림 23] 장바구니 메뉴 상세 조회 화면

메뉴의 상세 사진과 가격 정보를 확인할 수 있고, 수량을 선택하여 장바구니에 담을 수 있다.

4. 결 론

4.1 개발 내용 및 실험 결과 요약

금오공과대학교 학생들을 대상으로 한 공동 배달 주문 앱을 개발하였다. 사용자는 타 사용자와 함께 배달 시켜 먹고 싶을 경우 주문할 음식점, 인원 수, 음식 받을 위치, 제목, 본문, 희망 주문 시간을 입력하여 모임 게시물을 만들 수 있다. 이렇게 만든 게시물들은 타 사용자들에게 보여지게 되고 같이 주문 시키고 싶은 사용자들은 모임에 참가할 수 있게 된다. 모임에 참가한 사람들은 해당 게시물에 들어가 함께 주문시키는 사람이 누구인지, 몇 명인지 확인할 수 있게 되며 댓글을 통해 서로 대화할 수 있다. 또한 사용자들은 본인이 시키길 원하는 메뉴를 장바구니에 담고 다 담은 경우 방장의 계좌로 돈을 입금한다. 방장은 팀원들의 입금을 확인하고 주문 버튼을 입력함으로써 공동 주문을 완료할 수 있다.

4.2 향후 개선 과제

4.2.1 프로필 사진 업로드

현재 프로필 사진을 업로드할 수 없어 사용자 구분이 어렵다. 사용자마다 구분이 갈 수 있도록 프로필 사진을 업로드할 수 있도록 수정한다.

4.2.2 희망 주문 시간

직접 입력하는 것으로 설계하였는데, 직접 입력하는 것의 불편함을 느끼고 드롭 다운 방식으로 바꾼다.

4.3 기술적, 사회적, 경제적 파급 효과 및 기대 효과

최근에 배달료가 급증함으로써 소비자들의 부담이 증가하게 되었다. 대학생들은 학교 주변에서 서로 가까이 주거하고 있기 때문에 배달 음식을 같이 시킴으로써 배달비를 나누어 부담할 수 있다. 또한 금오공과대학교 이메일 한정으로 인증할 수 있도록 하여 외부인으로 인한 위험 상황을 방지할 수 있다. 즉, neighborfood 앱을 통해 배달비를 나누고 최소 배달 금액을 함께 채울 수 있는 인원을 짧은 시간에 모집하여 배달에 대한 부담을 줄이고 부가적 지출 감소로 인한 실질적으로 필요한 소비를 가능하게 한다.

5. 자체 평가서

5.1 자체 분석과 평가

5.1.1 이해원

장바구니 관리 프론트엔드 업무를 맡아 프로젝트를 진행하였다. github를 제대로 써본 적이 없었는데, 이번 프로젝트를 진행하며 많이 배울 수 있었다. 또한 웹앱을 처음 만들어보면서 모피어스 API 연동을 하게 되었는데 이 작업도 흥미로웠다.

프로젝트의 전체적인 결과에 대해서는 심사 때 질문이 들어온 관리자 페이지라던가, 사용자 신고 기능 등 아쉬운 점이 있지만 그래도 처음에 팀이 계획 했던 주요 목표는 다 달성했다고 생각하기에 만족스럽다. 수행 수준은 중~중상으로 평가한다.

5.1.2 천예진

회원관리, 모임 게시물 관리, 음식점 관리 백엔드 업무를 맡아 프로젝트를 진행했다. 수행 과정에서 가끔 어떻게 구현을 해야 할지 모르겠는 것도 있었으나, 열심히 인터넷 검색을 해보고 여러 예제 코드들을 살펴 보며, 예제 코드들을 우리 프로젝트에 어떻게 적용해야 할지 알게 되었다. 여러 예제 코드들을 살펴보는 게 중요하다고 생각한다. 수행 결과에 있어서는, 팀이 사전 계획했던 기능들을 다 구현했다고 생각한다. 수행 수준은 중상으로 평가한다.

5.1.3 전민규

데이터 준비, 댓글, 장바구니 관리를 맡아 프로젝트를 진행하였다. 특히 데이터 준비에서 어려움을 겪었는데, 마땅한 API를 찾을 수 없었기 때문에 크롤링을 진행하였다. 이때도 페이지마다 일정 패턴은 존재하였으나, 템플릿의 종류가 여러 가지였기 때문에 오랜 시간동안 크롤링 작업에 착수할 수 밖에 없었다. 이후 데이터들을 쿼리와 시키는 것을 무난하게 성공시켰고, 남은 기능들은 다른 팀원이 만든 작업이 기본 베이스로 활용할 수 있었기 때문에 빠르게 API개발에 재투입될 수 있었다. 수행 수준은 중상으로 평가한다.

5.1.4 백지원

회원 관리 프론트엔드 부분을 책임지고 프로젝트를 진행하였다. 초기에는 요구사항을 분석하고 기능 명세를 작성하는 단계에서 팀원들과 협업을 통해 목표를 명확히 설정하였으며, 필요한 컴포넌트를 구현하였다. 수행 과정 동안 사용자들이 편리하게 시스템을 이용할 수 있는 UI를 구현하는 것에 큰 만족감을 느꼈다. 또한 프로젝트 진행 중에 발생한 문제들을 빠르게 파악하고, 효과적인 해결책을 찾을 수 있게 되었다. 오류 수정과 최적화를 통해 문제 해결 능력을 기르고, 안정성과 성능이 보장된 웹 앱을 완성할 수 있었다.

5.1.5 신승미

수행 과정에서 계획한 일정보다 훨씬 더 느리게 진행되었다. 앱을 계획했던 것만큼, 그리고 성공적으로 마무리하고 싶었어서 팀원들을 많이 재촉하기도 했었다. 이에 팀원들이 잘 따라와준 덕분에 성공적으로 앱을 구현할 수 있었다. 개발이 끝난 후에 큰 산을 넘은 느낌이 있었는데 결과적으로 되돌아보니 과정에서 많이 성장한 것 같아 뿌듯함을 느꼈다.

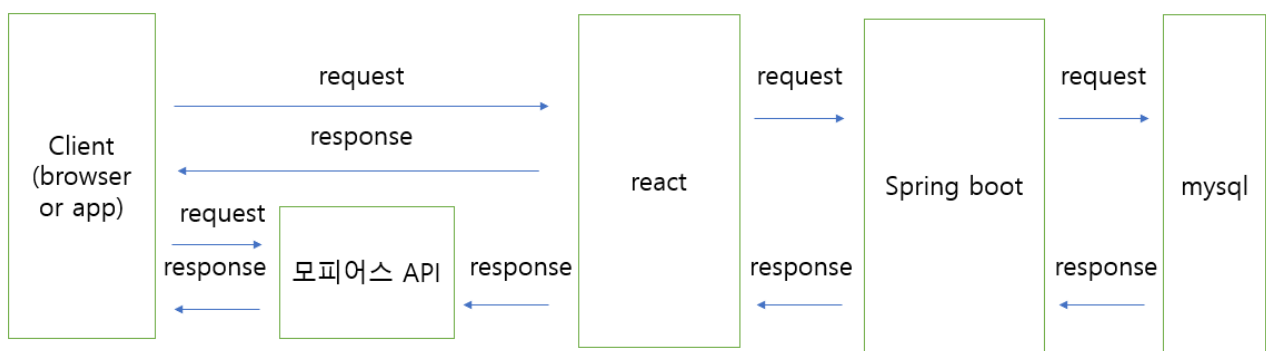
개발 과정에서 설계를 조금 미흡하게 한 탓에 중간중간에 수정을 많이 하게 되었는데 이 부분에서 시간을 많이 쓴 것 같아 좀 아쉬웠다. 설계를 꼼꼼하게 하는 게 얼마나 중요한 지 알게 되었고, 팀원들 사이에 소통을 많이 해서 설계를 미흡하게 했음에도 잘 해결해 나갔던 것 같다.

5.2 팀원 역할 및 기여도

이름	역할	기여도
이혜원	로그인, 회원가입, 장바구니 관리 화면, 메뉴 조회 화면 구현	5%
신승미	모임 게시물 관리 화면 구현, 음식점 조회 화면 구현	25%
백지원	마이페이지, 내가 참여한 게시물 화면 구현	25%
천예진	데이터베이스 설계, JWT 보안, API 작성 및 구현	20%
전민규	데이터 크롤링, AWS 배포, API 작성 및 구현	25%

6. 부록

6.1 시스템 설계도



6.2 핵심 소스코드

6.2.1 카카오 맵 API호출

```

const MakeMap = () => {
  const { kakao } = window;
  //position은 위도 경도
  const [position, setPosition] = useState({ lat: 33.450701, lng: 126.570667 });
  //address는 주소
  const [address, setAddress] = useState({});

  const getAddress = (lat, lng) => {
    const geocoder = new kakao.maps.services.Geocoder(); // 좌표 -> 주소로 변환해주는 객체
    const coord = new kakao.maps.LatLng(lat, lng); // 주소로 변환할 좌표 입력
    const callback = function (result, status) {
      if (status === kakao.maps.services.Status.OK) {
        setAddress(result[0].address);
      }
    };
    geocoder.coord2Address(coord.getLng(), coord.getLat(), callback);
  };
};

```

6.2.2 이메일 인증

```

public class EmailAuth {

  // 토큰 만료 시간
  private static final Long MAX_EXPIRE_TIME = 5L;

  @Id
  @GeneratedValue(generator = "uuid2")
  @GenericGenerator(name = "uuid2", strategy = "uuid2")
  @Column(length = 36)
  private String id;

  // private String code; // 인증코드

  private LocalDateTime expirationDate; // 만료날짜

  private boolean expired; // 만료여부

  // 사용자 email
  private String email;

  // 이메일 인증토큰 생성
  public static EmailAuth createEmailAuthToken(String email) {
    EmailAuth authToken = new EmailAuth();
    authToken.expirationDate = LocalDateTime.now().plusMinutes(MAX_EXPIRE_TIME);
    authToken.expired = false;
    authToken.email = email;
  }
}

```

참고문헌

- [1] 모피어스 위키 : <https://docs.morpheus.co.kr>
- [2] 김영한 자바 ORM 표준 JPA프로그래밍 에이콘 출판
- [3] 장철원 저 웹크롤링 & 데이터 분석 with 파이썬 인사이트 출판