

종합설계 프로젝트 수행 보고서

프로젝트명	위치기반의 푸드트럭 애플리케이션
팀번호	S1-3
문서제목	수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 4차발표 중간보고서() 최종결과보고서(O)

2020.01.22

팀원 : 황성우 (팀장)
배지예
정혜서

지도교수 : 최종필 교수
지도교수 : 교수

문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2019.12.18	황성우(팀장)	1.0	수행계획서	최초작성

문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (5월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I II III
	II. 본론 (1~3)	II. 본론 (1~4)	II. 본론 (1~5)	II. 본론 (1~7)	
	참고자료	참고자료	참고자료	참고자료	

이 문서는 한국산업기술대학교 컴퓨터공학부의
 “종합설계”교과목에서 프로젝트
 “위치기반의 푸드트럭 애플리케이션”을 수행하는
 (예시: S1-3, 황성우, 배지예, 정혜선)들이 작성한 것으로
 사용하기 위해서는 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성
2. 기존 연구/기술동향 분석
3. 개발 목표
4. 팀 역할 분담
5. 개발 일정
6. 개발 환경

II. 본론

1. 개발 내용
2. 문제 및 해결방안
3. 시험시나리오
4. 상세 설계
5. Prototype 구현
6. 시험/ 테스트 결과
7. Coding & DEMO

III. 결론

1. 연구 결과
2. 작품제작 소요재료 목록

참고자료

I. 서론

1. 작품선정 배경 및 필요성

최근 들어 붕어빵을 파는 푸드트럭이 눈에 잘 띄지 않는다. 그리하여 SNS에서 ‘붕세권’이라는 용어까지 생겼다. ‘역세권’이라는 단어를 패러디한 것으로 붕어빵을 파는 곳과 가까운 곳을 뜻한다. 어떤 사람들은 붕어빵을 팔고 있는 위치를 알기위해 SNS 페이지에 어느 지역에 붕어빵 파는 곳을 알려달라며 게시글을 올려주도록 요청하기도 한다. 붕어빵 말고도 푸드트럭에는 여러가지가 있다. 본 작품은 우리 주변에 있는 푸드트럭의 위치를 알기 쉽게 하기 위함과 우리나라 사람들의 성격이 급하다는 특징을 고려하여 미리 주문을 하여 푸드트럭 앞에서 음식을 기다리지 않고 바로 가져갈 수 있는 시스템을 만들면 이점이 많다고 생각된다.

2. 기존 연구/기술동향 분석

최근 ‘배달의 민족’, ‘요기요’, 다이닝 코드’ 등 다양한 음식 관련한 어플이 각광받고 있다. 이 어플들은 위치 기반으로 맛집을 추천, 또는 음식 배달을 도와주는 어플이다. 배달의 민족을 예로 들자면 사용자의 휴대폰의 gps기능을 사용해서 주변에 있는 배달 가능한 음식점을 보여준 후 주문을 도와주는 어플이다. 편리함 덕분에 꾸준히 사랑을 받고 있다. 이렇게 gps 기능과 음식 관련 서비스를 제공하는 결과물들이 꾸준히 나오고 있다.

3. 연구 목표

사람들이 쉽게 접근할 수 있는 애플리케이션을 만들어 사용자 주변 푸드트럭들의 위치, 그 곳에서 판매하고 있는 음식의 종류도 사용자가 알게 함과 미리 주문할 수 있는 시스템으로 사용자의 시간절약까지 되는 등 사용자에게 초점을 맞춘 실용적인 푸드트럭 시스템 개발하는 것이다.

4. 팀 역할 분담

	황성우	배지예	정혜선
자료수집	어플 관련 사례 조사		
설계	서비스 관련 세부 기능들 설계 (포인트, 후기 등등)		
구현	서버 구축 및 네트워크 기능 웹 세부 기능 구현 (관리자)	GPS 기능 구현 어플 세부 기능 구현 (사용자)	DB 기능 구현 어플 세부 기능 구현 (점주)
테스트	사용자, 관리자, 점주 별 어플 개별 작동 테스트 어플들 간의 통신 테스트		

5. 개발 일정

추진사항	12월	1월	2월	3월	4월	5월	6월	7-9월
사전조사 및 제안서 발표	■							
자료수집 및 스터디	■	■						
설계 발표		■						
시스템 구현			■	■	■	■		
프로토 타입 완성					■	■		
통합 테스트 및 보완					■	■	■	
최종 보고서 발표								■

6. 개발환경

- 개발언어 : JAVA, 안드로이드
- 사용 프레임워크 : PhoneGap
- 개발방법론 : 통합 프로세스 모델
- 주요 라이브러리 ; JDK
- 서버 프로세서 : Tomcat server 9.0
- DB : SQLite
- 디자인 : 구글 Material 디자인

II. 본론

1. 개발내용 :

< SW >

- 푸드트럭 위치 알려주기 : GPS 기능 사용
- 푸드트럭 정보, 사용자 정보 저장 : SQLite DB 사용
- 실시간 양방향 정보 주고 받기 : Socket 통신
- Big Data 수집
- 관리자용 웹 : Eclipse 사용

< HW >

- 안드로이드 스마트폰

2. 문제 및 해결방안 :

- 데이터 저장을 위한 서버 구축
- 사용자, 점주, 관리자 등 상호 통신을 위한 소켓 프로그래밍

- 서비스의 체계화를 위해 어플과 웹으로 분리
- 위치를 파악하기 위하여 사용자용, 점주용 어플에 GPS 기능을 추가
- 점주용 어플은 사용자의 요청에 따라 위치 기능을 제어 가능 및 서버에 등록
- 주문 예약 시 시간에 맞춘 알람 설정

3. 시험 시나리오

3.1 점주 시나리오

점주 Ver.																
초기 화면	회원 가입	최초 로그인	위치 등록	가게 소개 입력												
<p>어플명</p> <p>아이디</p> <p>비밀번호</p> <p>회원가입 아이디 찾기 비밀번호 찾기</p>	<p>회원가입</p> <p>이름</p> <p>휴대폰 번호</p> <p>아이디</p> <p>중복확인</p> <p>이메일</p> <p>비밀번호</p> <p>비밀번호 확인</p> <p>사업자 등록증 첨부</p> <p><input type="radio"/> gps 동의 <input type="radio"/> 개인정보 수집 동의</p> <p>회원 신청</p>	<p>위치등록</p> <p>가게 소개 입력</p> <p>메뉴 정보 입력</p> <p>리뷰 보기</p> <p>들어온 주문 보기</p> <p>회원정보 수정</p> <p>영업 on/ off</p>	<p>Gps 기반 지도</p> <p>상세주소</p>	<p>사진등록</p> <p>가게 소개 내용 오픈 날짜 시간 상세 위치 메뉴 소개</p>												
점주 Ver.																
메뉴 정보 입력	리뷰 보기	들어온 주문 보기	회원 정보 수정													
<p>추가하기</p> <table border="1"> <thead> <tr> <th>메뉴이름</th> <th>가격</th> </tr> </thead> <tbody> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </tbody> </table>	메뉴이름	가격											<p>리뷰</p> <p>댓글달기</p> <p>리뷰</p> <p>댓글달기</p> <p>리뷰</p> <p>댓글달기</p> <p>리뷰</p> <p>댓글달기</p> <p>리뷰</p> <p>댓글달기</p>	<p>주문자 정보 메뉴</p> <p>주문자 정보 메뉴</p> <p>주문자 정보 메뉴</p> <p>주문자 정보 메뉴</p> <p>주문자 정보 메뉴</p>	<p>이름</p> <p>휴대폰 번호</p> <p>아이디</p> <p>이메일</p> <p>기존 비밀번호</p> <p>새로운 비밀번호</p> <p>새로운 비밀번호 확인</p> <p>사업자 등록증 첨부</p> <p>정보 수정</p>	
메뉴이름	가격															

- 어플 실행 시 초기 화면에 아이디와 비밀번호 입력창, 고객 회원가입, 점주회원 가입 버튼이 보인다.
- 푸드트럭 등록을 위하여 점주 회원가입 버튼을 클릭하여 회원가입 화면으로 이동한다.
- 이름, 휴대폰 번호, 아이디(중복확인), 이메일, 비밀번호, 비밀번호 확인란을 모두 작성하고, 사업자 등록증을 관리자에게 보낸다. gps 동의와 개인정보 수집 동의를 완료 후 회원 신청을 한다. 관리자가 회원 신청 승인을 하여야 영업이 가능하다.
- 승인 후, 아이디 비밀번호 입력을 하여 로그인을 한다.
- 로그인을 하면 위치등록, 가게 소개 입력, 메뉴 정보 입력, 리뷰 보기, 들어온 주문 보기, 회원정보 수정, 영업 on/off 버튼이 화면에 나타난다.
- 위치등록 클릭 시, GPS 기반 지도가 나타나며 자신의 정확한 위치를

다시 표시할 수 있으며 상세 주소를 입력한다.

- 가게 소개 입력 클릭 시, 푸드트럭의 사진을 등록할 수 있고, 가게 소개 할 내용, 오픈 날짜와 시간, 상세 위치, 메뉴 소개 등을 입력할 수 있다.
- 메뉴 정보 입력 클릭 시, 메뉴의 이름과 각각의 가격을 기입할 수 있다.
- 리뷰보기 클릭 시, 자신의 푸드트럭에 사용자가 리뷰한 내용들이 나열되어있다.
- 들어온 주문 보기 클릭 시, 주문자의 정보와 메뉴가 나열되어 있다.
- 회원 정보 수정 클릭 시, 휴대폰 번호, 이메일, 비밀번호만 변경 가능하다.
- 영업 on/off 버튼은 on으로 되어있을 경우 사용자의 어플에서 해당 푸드트럭이 존재하고, off일 경우 해당 푸드트럭이 사용자의 어플에서 존재하지 않는다.

3.2 사용자 시나리오

사용자 Ver.			
초기화면	회원가입	비밀번호 찾기	메인 화면
<p>어플명</p> <p>아이디</p> <p>비밀번호</p> <p>회원가입 아이디 찾기 비밀번호 찾기</p>	<p>회원가입</p> <p>이름</p> <p>휴대폰 번호</p> <p>아이디</p> <p>중복확인</p> <p>이메일</p> <p>비밀번호</p> <p>비밀번호 확인</p> <p><input type="radio"/> gps 동의</p> <p><input type="radio"/> 개인정보 수집 동의</p> <p>회원가입</p>	<p>비밀번호 찾기</p> <p>가입한 아이디</p> <p>이름</p> <p>휴대폰번호</p> <p>확인</p> <p>새 비밀번호</p> <p>새 비밀번호 확인</p> <p>변경완료</p>	<p>어플명</p> <p>한식</p> <p>중식</p> <p>양식</p> <p>일식</p> <p>분식</p> <p>Etc...</p> <p>주문내역</p> <p>나의 정보</p>
사용자 Ver.			
카테고리 선택	푸드트럭 선택	후기	리뷰 쓰기
<p>Gps 지도 보여주기</p> <p>푸드트럭 정보</p> <p>푸드트럭 정보</p> <p>푸드트럭 정보</p> <p>푸드트럭 정보</p> <p>푸드트럭 정보</p>	<p>푸드트럭 이름</p> <p>사진 오픈 날짜 시간 위치</p> <p>메뉴 & 가격</p> <p>후기</p> <p>주문하기</p>	<p>별점</p> <p>리뷰쓰기</p> <p>아이디 내용</p> <p>아이디 내용</p> <p>아이디 내용</p> <p>아이디 내용</p> <p>아이디 내용</p>	<p>별점</p> <p>사진첨부</p> <p>내용</p> <p>등록</p>
사용자 Ver.			
주문하기	주문내역	나의 정보	회원 정보 수정
<p><input type="radio"/> 메뉴</p> <p><input type="radio"/> 메뉴</p> <p><input type="radio"/> 메뉴</p> <p><input type="radio"/> 메뉴</p> <p><input type="radio"/> 메뉴</p> <p><input type="radio"/> 메뉴</p> <p>담기</p> <p>주문 내용 & 최종 가격</p> <p>주문 접수</p>	<p>푸드트럭 이름</p> <p>일시 주문 메뉴</p> <p>푸드트럭 이름</p> <p>일시 주문 메뉴</p> <p>푸드트럭 이름</p> <p>일시 주문 메뉴</p>	<p>회원정보 수정</p> <p>포인트 : **점</p> <p>환경설정</p>	<p>회원정보 수정</p> <p>가입한 아이디</p> <p>이름</p> <p>휴대폰번호</p> <p>기존 비밀번호</p> <p>새 비밀번호</p> <p>새 비밀번호 확인</p> <p>변경완료</p>

- 어플 실행 시 초기 화면에 아이디와 비밀번호 입력창, 점주 회원가입, 고객 회원가입 버튼이 뜬다.
- 푸드트럭 등록을 위하여 고객 회원가입 버튼을 클릭하여 회원가입 화면으로 이동한다.
- 이름, 휴대폰 번호, 아이디(중복확인), 이메일, 비밀번호, 비밀번호 확인란을 모두 작성하고, gps동의를 개인정보 수집 동의를 완료 후 회원가입 버튼으로 완료한다.
- 아이디 비밀번호 입력을 하여 로그인을 한다.
- 아이디를 까먹었을 시, 아이디 찾기 버튼을 통하여 이름, 휴대폰 번호, 이메일로

찾을 수 있다.

- 비밀번호를 까먹었을 시, 비밀번호 찾기 버튼을 통하여 아이디, 이름, 휴대폰 번호, 이메일로 찾을 수 있다.
- 로그인 후, 지도, 주문내역과 나의 정보를 볼 수 있는 하단 네비게이션 메뉴가 있다.
- 로그인을 하면 자동으로 지도가 뜬다. GPS 지도가 보여지면서 나의 위치 기반으로 푸드트럭이 아이콘으로 표시되어 있다.
- 푸드트럭 선택 시, 푸드트럭 이름이 맨 위에 쓰여 있으며, 푸드트럭 사진, 오픈 날짜와 시간, 위치, 후기, 메뉴들의 가격이 나열되어있으며 메뉴 선택 후 주문하기 버튼을 통하여 주문을 할 수 있다.
- 후기를 클릭 시, 해당 푸드트럭 주문자라면 리뷰쓰기 버튼을 통하여 쓸 수 있으며 다른 사람이 쓴 리뷰들도 나열되어있다.
- 리뷰 쓰기 클릭 시, 후기 내용을 쓰고 리뷰등록을 할 수 있으며 등록이 완료되면 토스트 메시지로 '등록 되었습니다'라고 알려준다.
- 주문하기에 들어가면 메뉴들이 나열되어 있고 담기 버튼을 통하여 장바구니에 담을 수 있고, 선택한 음식에 대한 주문내용과 최종 가격이 표시되며 주문 접수 버튼을 통하여 주문을 완료한다.
- 메인 화면에서 주문내역 클릭 시, 자신이 지금까지 주문한 푸드트럭들의 정보를 나열하여 보여준다.
- 나의 정보 클릭 시, 회원정보 수정, 포인트, 환경설정 버튼이 있다.
- 회원정보 수정 클릭 시 자신이 회원가입 시 입력한 정보들이 뜨는데 그 중 휴대폰 번호, 비밀번호만 변경 가능하다.

3.3 관리자 시나리오

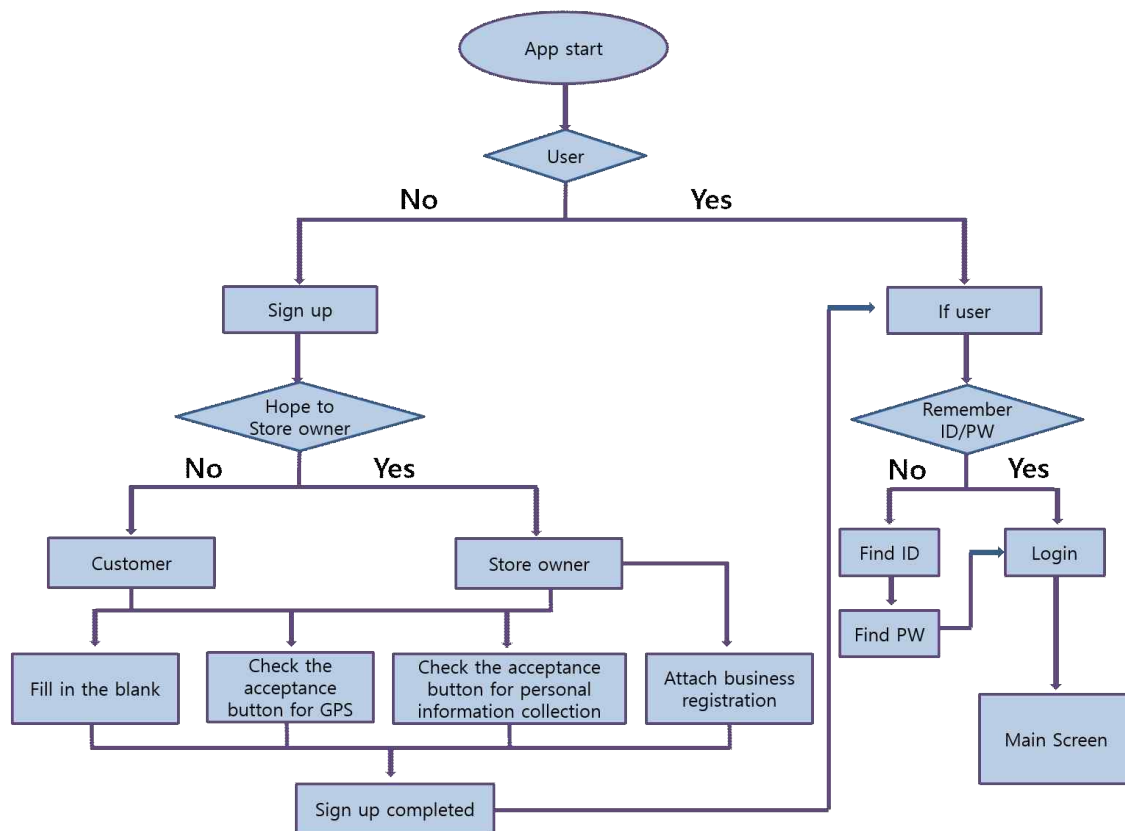
관리자 Ver.		
로그인 화면	메인 화면	푸드트럭 신청 확인
<p><u>여름명</u>(관리자 로그인)</p> <div>아이디</div> <div>비밀번호</div> <p>아이디 찾기 비밀번호 찾기</p>	<div>푸드트럭 신청 확인</div> <div>점주 정보 삭제</div> <div>사용자 정보 삭제</div> <div>리뷰 삭제</div>	<div>푸드트럭 신청 정보</div> <div>승인</div> <div>푸드트럭 신청 정보</div> <div>승인</div> <div>푸드트럭 신청 정보</div> <div>승인</div>

- 점주가 첨부한 푸드트럭에 대한 사업자 증빙자료를 잘 검토하여 해당 푸드트럭 등록을 승인해준다.
- 웹을 통하여 점주를 삭제할 수 있다.
- 사용자를 삭제할 수 있다.
- 리뷰를 삭제할 수 있다.

4. 상세 설계

4.1 점주 상세 설계

4.1.1 점주 회원가입, 로그인



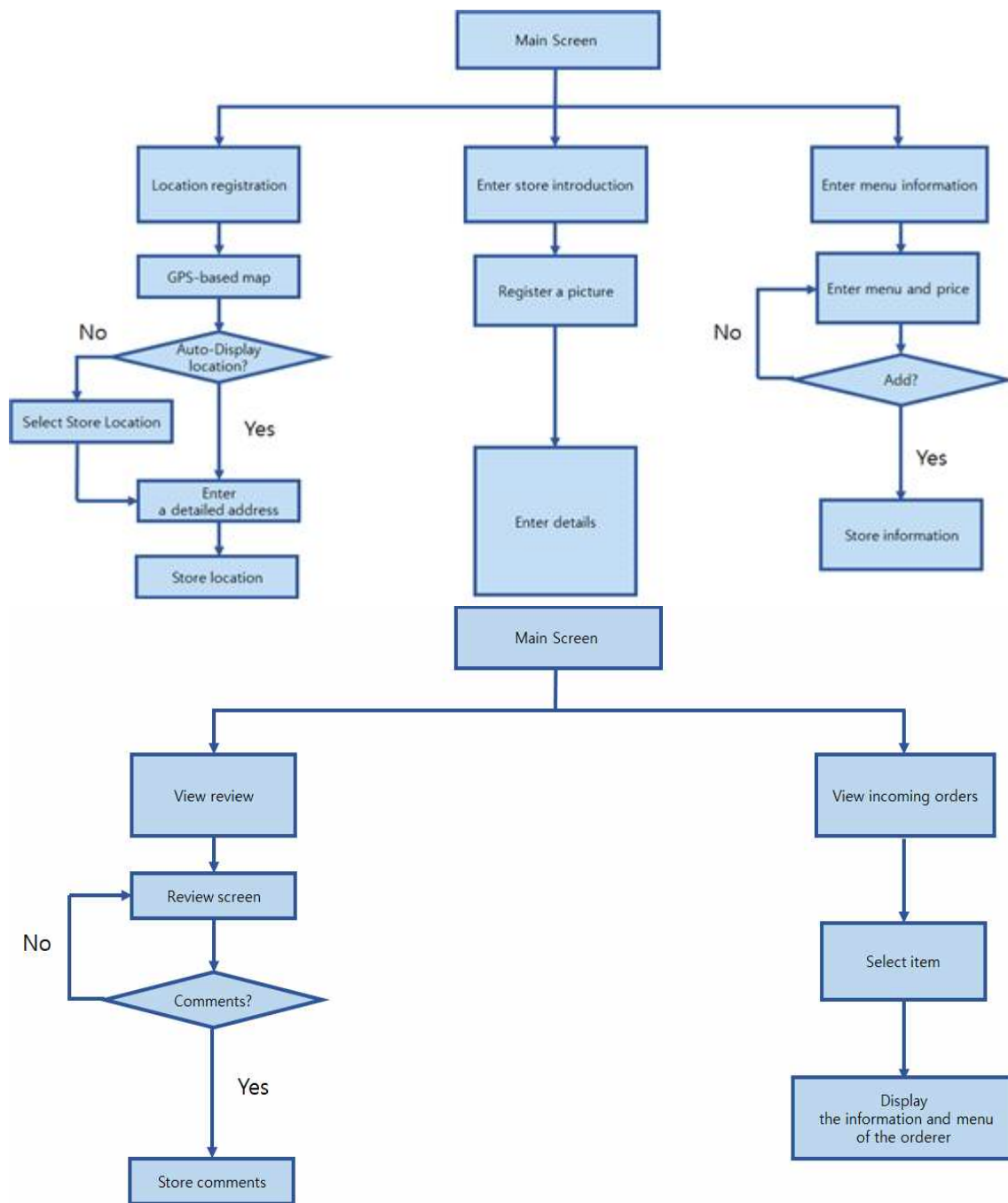
4.1.2 점주 가게 위치 등록 및 가게 정보 입력

4.1.3 리뷰 보기, 들어온 주문 보기

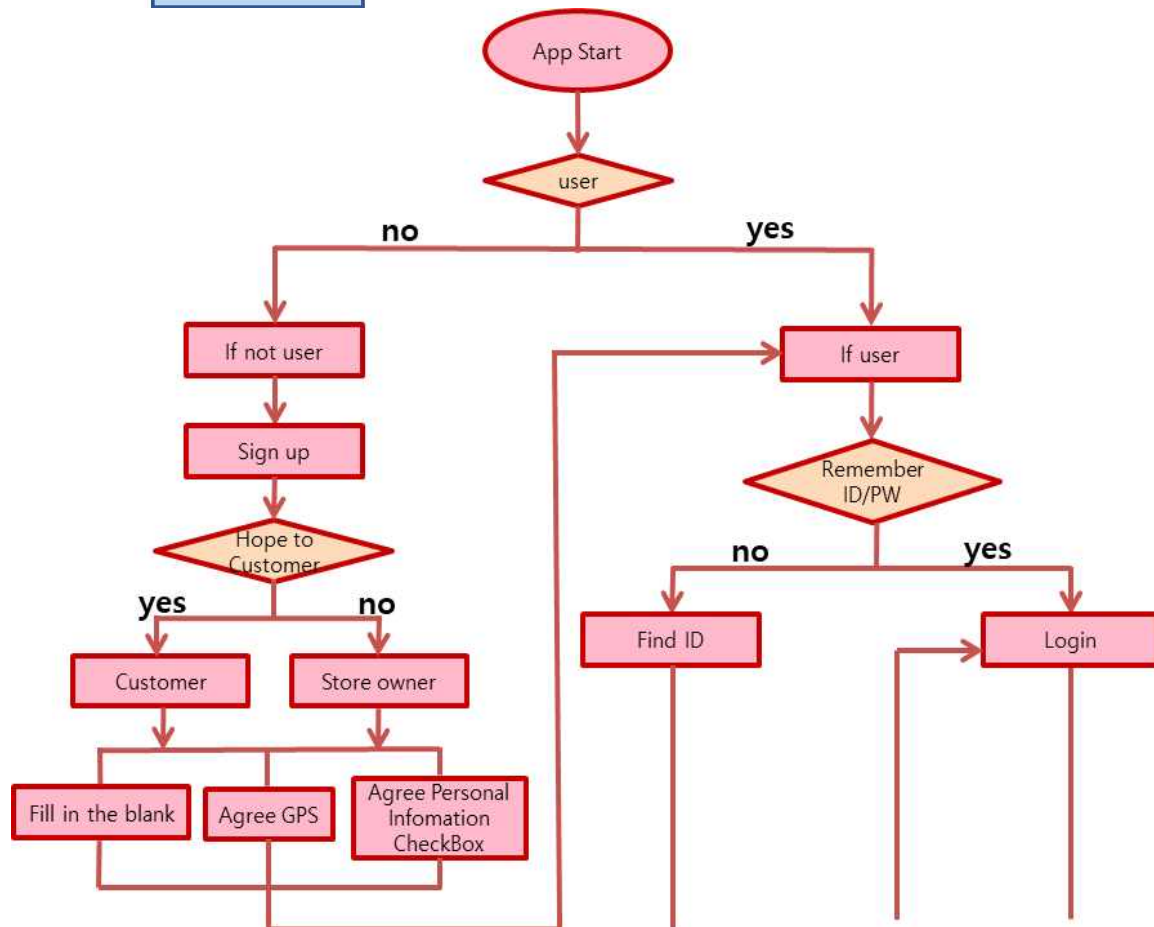
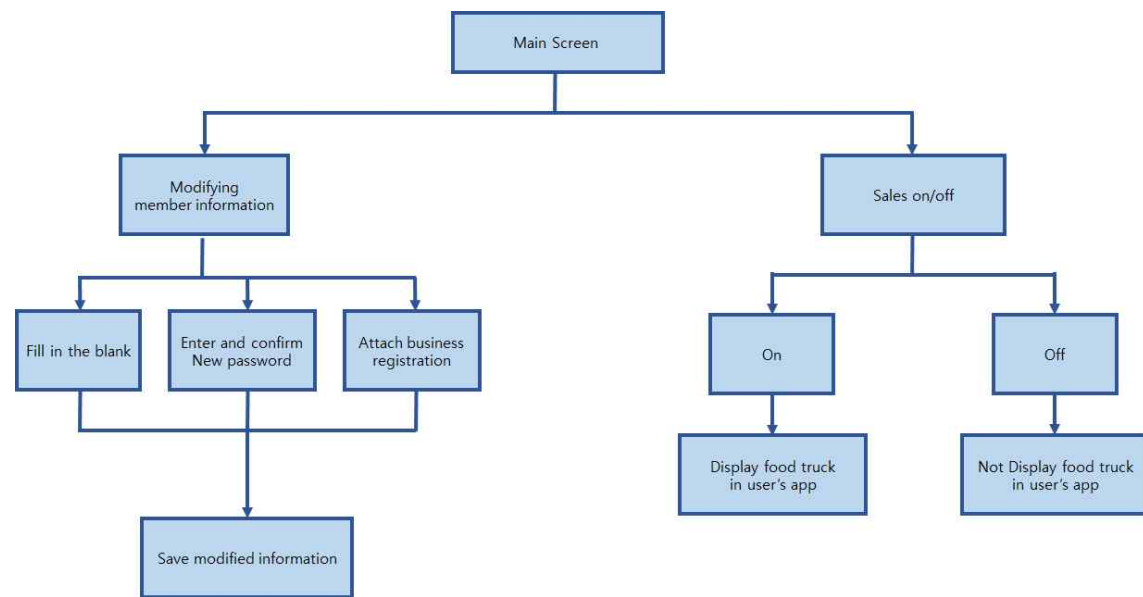
4.1.4 회원정보 수정, 영업 on/off

4.2 사용자 상세 설계

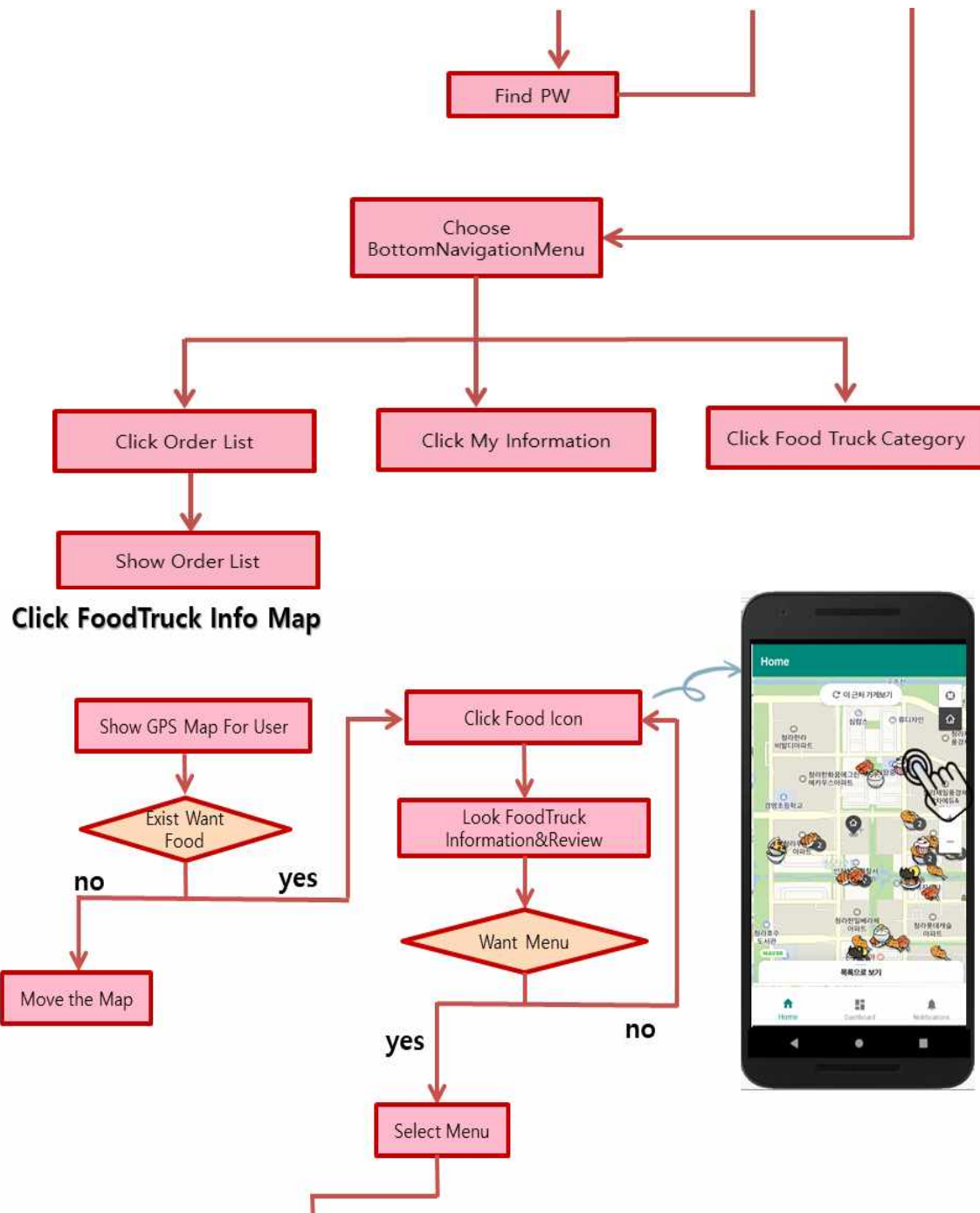
4.2.1 사용자 회원가입, 로그인, 로그인 후 기능 수행

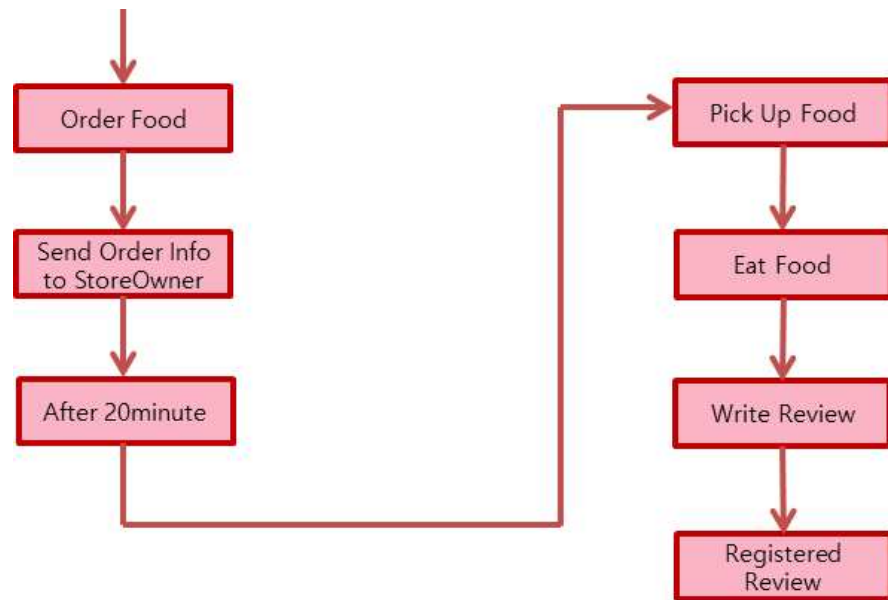


4.2.2 푸드트럭 지도보기, 음식 주문하기 기능

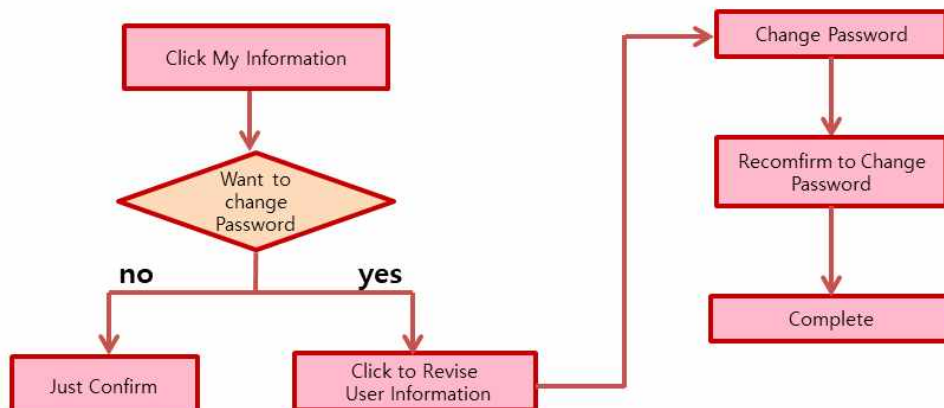


4.2.3 나의 정보 수정하기





Click My Information



5. Prototype 구현

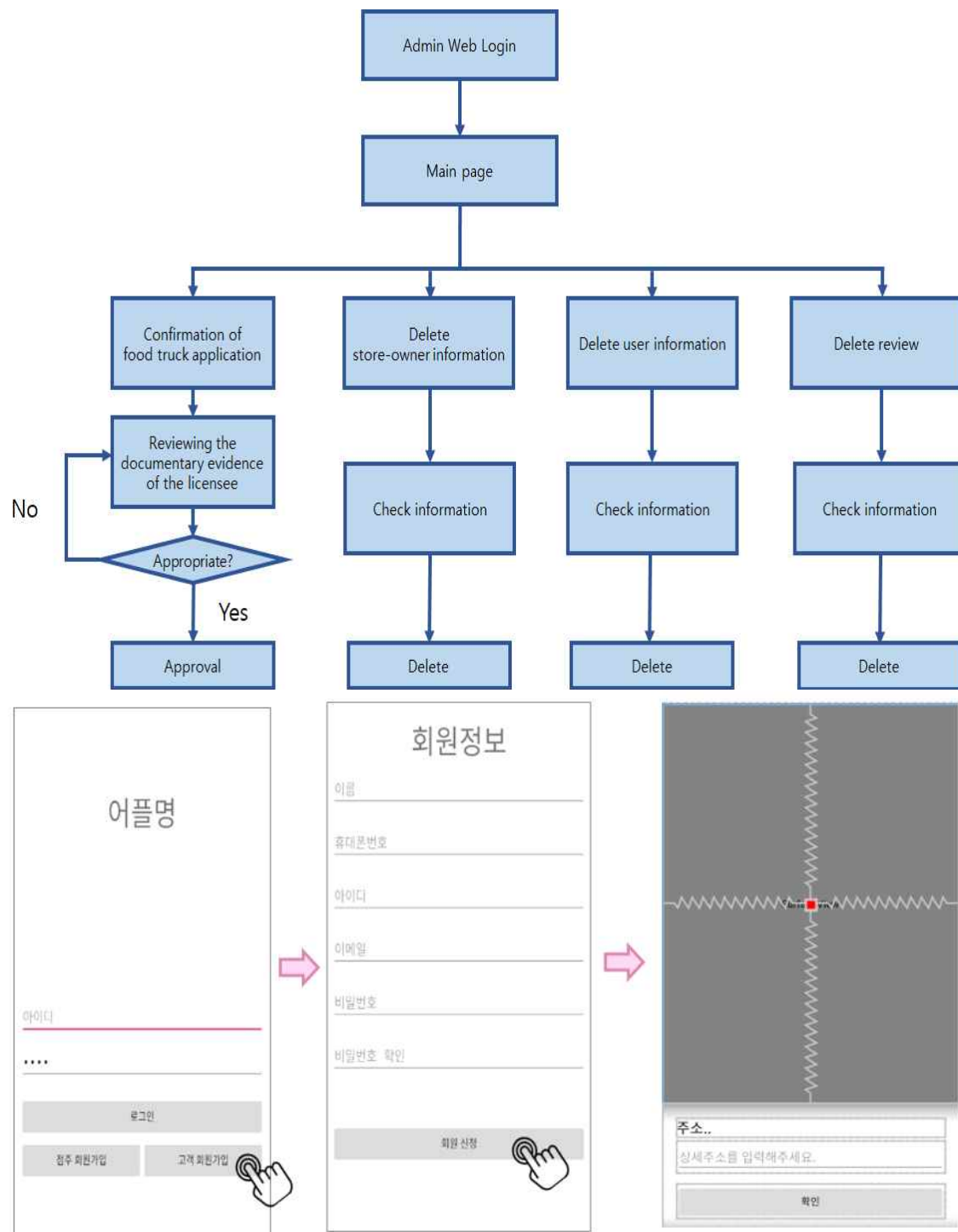
5.1 고객 Prototype

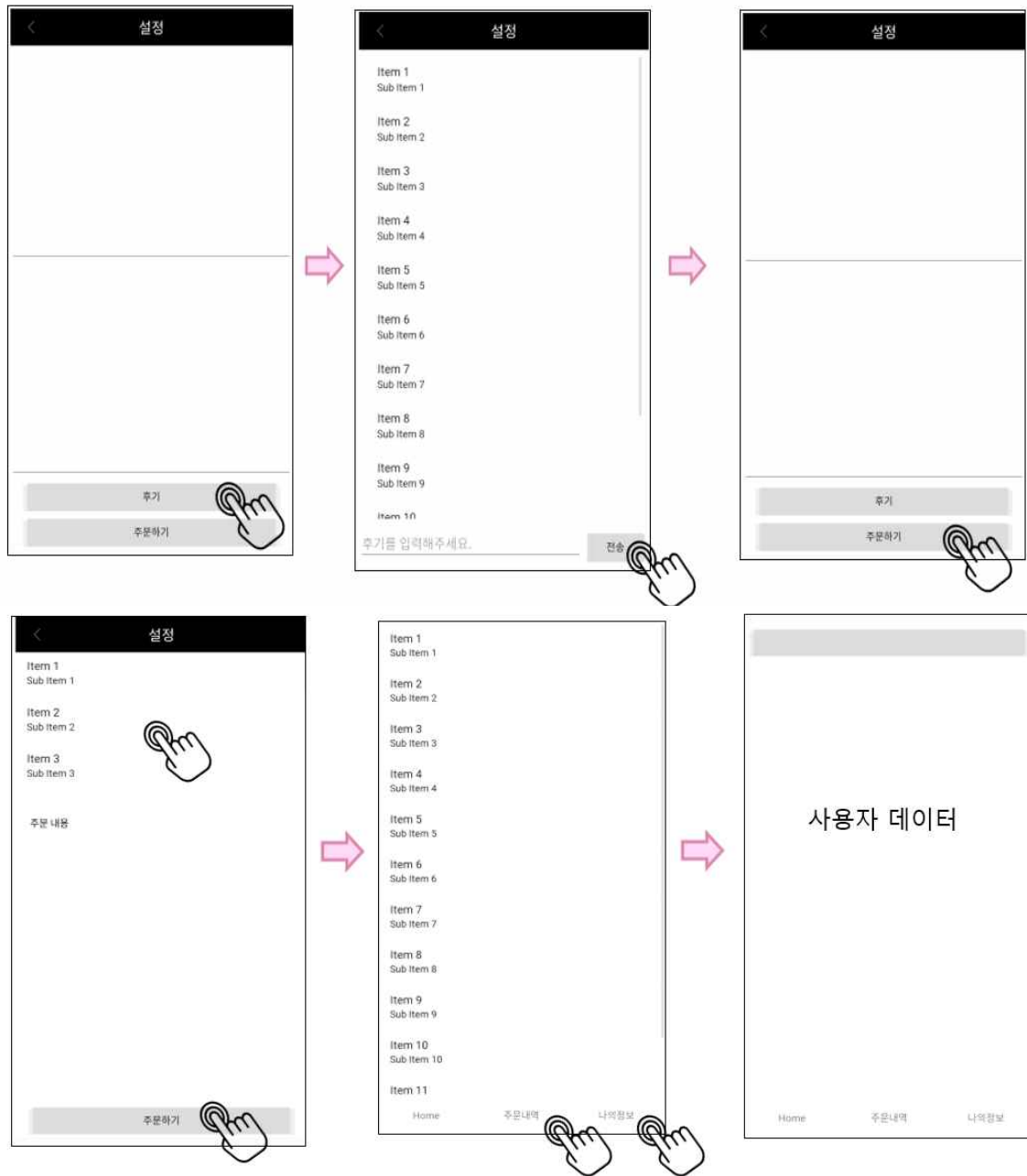
5.1.1 회원가입 / 위치 설정

5.1.2 후기 작성 / 주문하기 / 나의 정보 수정하기

5.2 점주 Prototype

5.2.1 로그인 / 회원가입



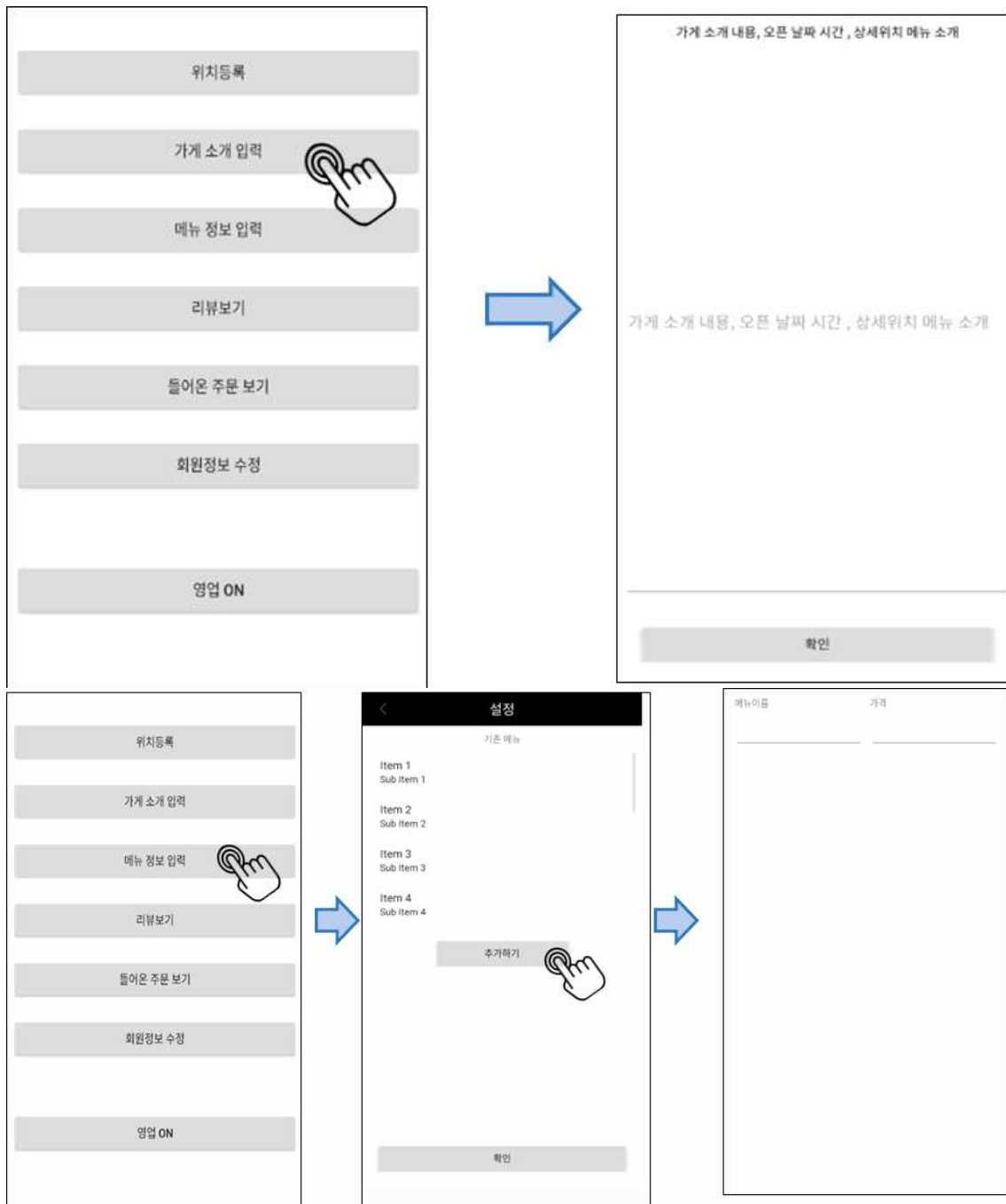


5.2.2 가게 위치 등록 & 가게 소개 입력

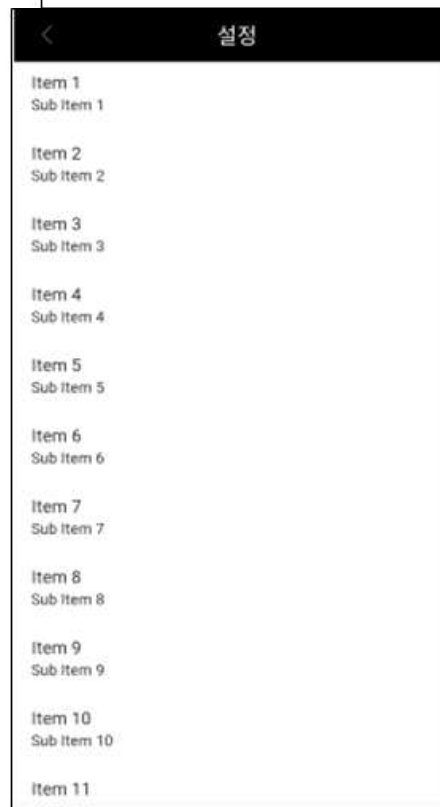
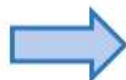
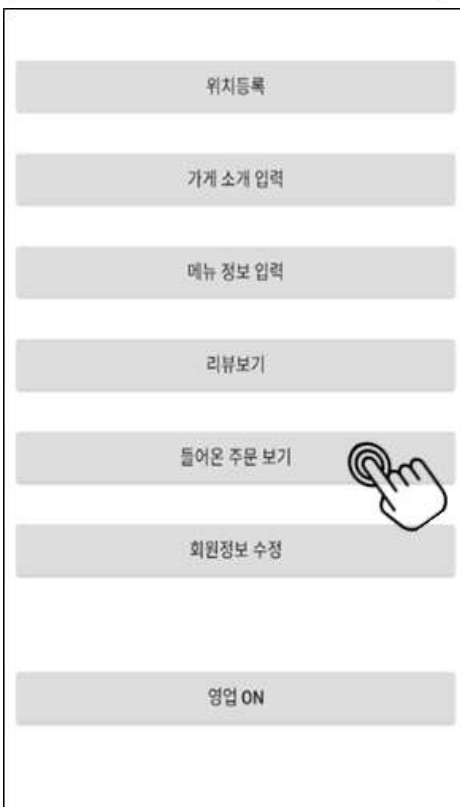
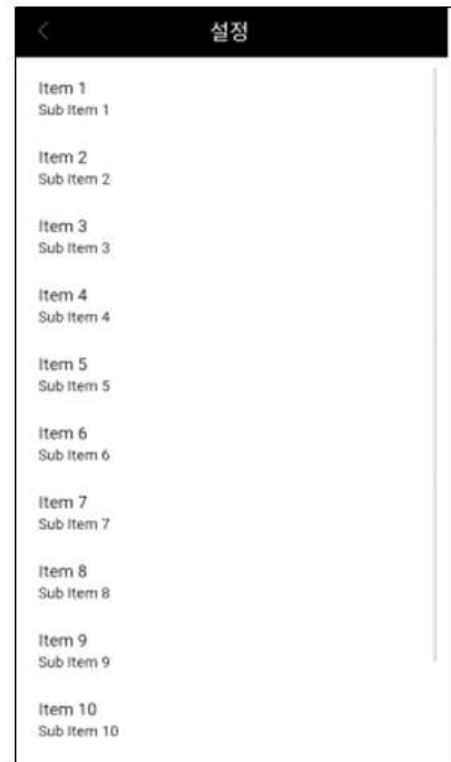
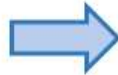


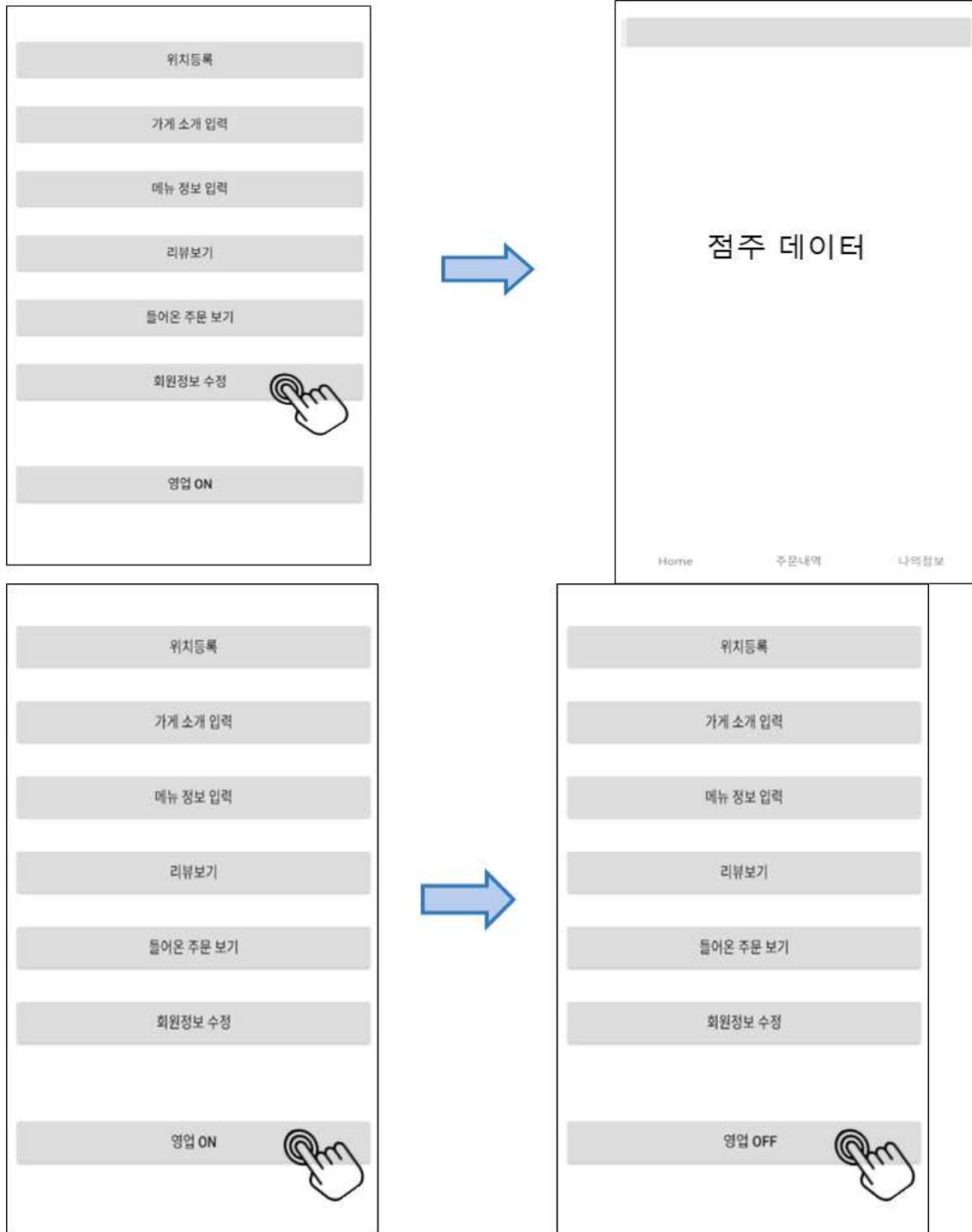
5.2.3 메뉴 정보 입력

5.2.4 리뷰보기 & 들어온 주문 보기



5.2.5 회원정보 수정 & 영업 on/off





5.3 관리자 Prototype

5.3.1 관리자 웹 로그인

5.3.2 점주 관리(승인, 사용중지, 탈퇴)

관리자

관리시스템

LOGIN



관리자
고객관리
리뷰
Hi 관리자님 [Log Out](#)

[HOME](#)
[고객관리](#)
[점주](#)

고객관리

점주

리뷰

점주

승인
사용중지
탈퇴

NO	아이디	이름	전화번호	이메일	주소	승인여부
1	srac40	홍길동	01027005914	srac0425@gmail.com	대한민국 서울특별시 중구 소공동 110-1 태스트	Y
2	srac50	테스트	01026163174	srac50@gmail.com		Y
3	hysun	hysun	01012345678	hysun@naver.com		Y

«
1
page 1 of 0
»



5.3.3 고객관리(고객 정보 및 리뷰 삭제)

관리자

고객관리

목록

로그아웃

고객관리

HOME

고객관리

일반

경주

일반

일반

목록

NO	아이디	이름	폰번호	이메일
0	oc1	고객	01066666242	oc1
-1	wor13120	hyesun	01075819596	wor13120naver.com
-2	hsw9503	sunwoo	01041890637	hsw9503naver.com
-3	8	sunwoo	01041890637	hsw9503naver.com
-4	2017152019	지혜	01068476311	z67111naver.com
-5	wor1234	hyesun	01012345678	wor1234

1

page 1 of 0

6. 시험/테스트 결과

6.1 점주용 (APP)

6.1.1 로그인



어플명

snap40

로그인

점주 회원가입

고객 회원가입

|||

○

<



위치등록

가게 소개 입력

메뉴 정보 입력

리뷰보기

들어온 주문 보기

회원정보 수정

영업 ON

|||

○

<

6.1.2 위치 등록

KT 4:39 84%

위치등록

가게 소개 입력

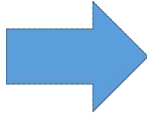
메뉴 정보 입력

리뷰보기

들어온 주문 보기

회원정보 수정

영업 ON



주소..

상세주소를 입력해주세요.

확인

6.1.3 가게 소개 입력

KT 4:39 84%

위치등록

가게 소개 입력

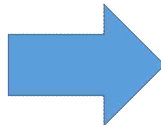
메뉴 정보 입력

리뷰보기

들어온 주문 보기

회원정보 수정

영업 ON



KT 4:39 84%

가게 소개 내용, 오픈 날짜 시간, 상세위치 메뉴 소개

testshop 푸드트럭입니다.

OPEN 10 : 00

CLOSE 21 : 00

본 푸드트럭은 시청역 앞에 위치해있습니다.

확인

6.1.4 메뉴 정보 입력

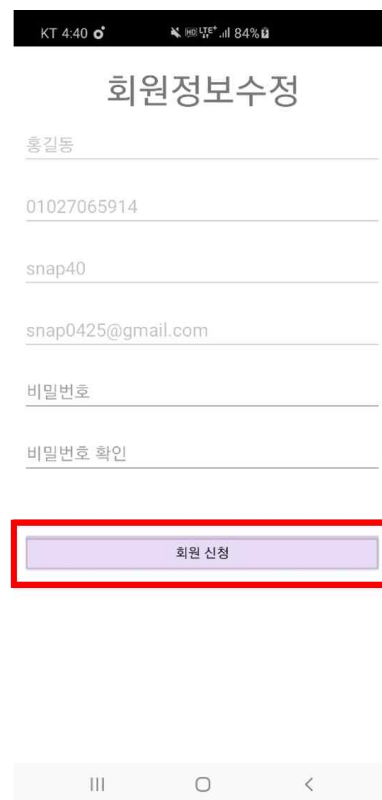
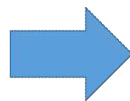


6.1.5 리뷰 확인



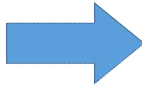
6.1.6 들어온 주문 보기

6.1.7 회원정보 수정



6.2 고객용 (App)

6.2.1 로그인



6.2.2 푸드 트럭 선택



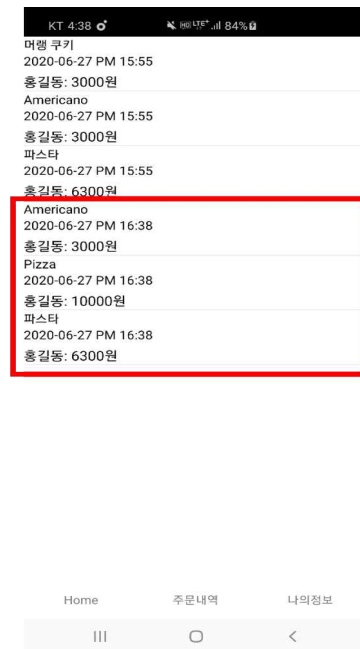
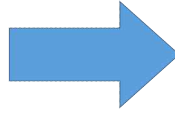
6.2.3 주문



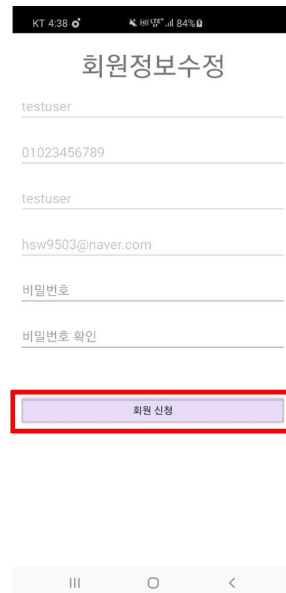
6.2.4 리뷰 작성



6.2.5 주문내역 확인



6.2.6 회원정보 수정



6.3 WEB

본론의 5.3 관리자 Prototype (p.19~p.21) 참조

7. Coding & DEMO

7.1 로그인

7.1.1 LoginActivity.java (App 로그인)

```
edit_id = (EditText)findViewById(R.id.edit_id);
edit_pw = (EditText)findViewById(R.id.edit_pw);
//(생략)

Handler mAfterAccum = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        if (msg.arg1 == 0) {
            customProgressClose();
            String res = (String) msg.obj;
            // res = res.replace(" ", "").replace("\n",
            , "");
            Log.e("SKY", "" + res);
            try {
                JSONObject jsonObject = new
                JSONObject(res); //SUCCESS

                //{"rc": "0000", "rc_txt": "ok", "rc_data": 3397}
                if
                (jsonObject.getString("rc").equals("000")) {
                    JSONObject pushobj =
                    jsonObject.getJSONObject("rc_data");
```

```
Check_Preferences.setAppPreferences(getApplicationCo
ntext(), "KEYINDEX",
pushobj.getString("KEYINDEX"));
//(생략)
if(pushobj.getString("SHOP_KEEPER").equals("CUSTOMER
")){
            startActivity(new
            Intent(getApplicationContext(),
            CustomerMainActivity.class));
        }else{
            startActivity(new
            Intent(getApplicationContext(),
            MainActivity.class));
        }
        return;
    }
}

else{
    Toast.makeText(getApplicationContext(), "" +
    jsonObject.getString("rc_txt"),
    Toast.LENGTH_SHORT).show();
}

} catch (JSONException e) {
    e.printStackTrace();
    customProgressClose();
}
}
}
```

7.2 메인화면

7.2.1 MainActivity.java (점주용)

```
//버튼 리스너 구현 부분
View.OnClickListener btnListener = new
View.OnClickListener() {
    public void onClick(View v) {
        switch (v.getId()) {

            case R.id.btn1: //위치등록
                startActivity(new
                Intent(getApplicationContext(),
                MapActivity.class));
                break;
            case R.id.btn2: //가게소개입력
                startActivity(new
                Intent(getApplicationContext(),
                IntroduceActivity.class));
            //(생략)
```

```
mThread = new
AccumThread(getApplicationContext(), mAfterAccum,
map, 0, 0, null);
mThread.start(); //스레드 시작!!
break;
}
}

Handler mAfterAccum = new Handler() {
    try {
        JSONObject jsonObject = new JSONObject(res);
        //(생략)
    }
```

7.2.2 MainActivity.java (사용자용)

```
address_txt =
(TextView)findViewById(R.id.address_txt);
detail_address_edit =
(EditText)findViewById(R.id.detail_address_edit);

bodyview =
(RelativeLayout)findViewById(R.id.bodyview);
mapview = (MapView)findViewById(R.id.mapview);
mapview.setMapViewEventListener(this);
//생략

View.OnClickListener btnListener = new
View.OnClickListener() {
//생략
```

```
map.clear();
map.put("url", DEFINE.SERVER_URL +
DEFINE.MEMBER_MAP_UPDATE);
map.put("key_index", "" +
Check_Preferences.getAppPreferences(getApplicationCo
ntext() , "KEYINDEX"));
map.put("wi", "" + wi);
map.put("gi", "" + gi);
map.put("address", "" + addressName +
" " + detailAddressName);
//스레드 생성
mThread = new
AccumThread(getApplicationContext(), mAfterAccum,
map, 0, 0, null);
mThread.start(); //스레드 시작!!
}
```

7.3 메뉴

7.3.1 MenuListAdapter.java

```
public MenuListAdapter(Activity a,
ArrayList<MenuObj> m_board) {
    activity = a;
    inflater = (LayoutInflater)
activity.getSystemService(Context.LAYOUT_INFLATER_SE
RVICE);
    items = m_board;
}
//생략
public View getView(final int position, View
convertView, ViewGroup parent) {
    final MenuObj board = items.get(position);
    ViewHolder vh = new ViewHolder();
```

```
if (convertView == null) {
    convertView =
inflater.inflate(R.layout.menulist_item, null);
    vh.item_id = (TextView)
convertView.findViewById(R.id.item_id);
    vh.item_price = (TextView)
convertView.findViewById(R.id.item_price);
    convertView.setTag(vh);
} else {
    vh = (ViewHolder) convertView.getTag();
}

vh.item_id.setText(board.getNAME());
vh.item_price.setText(board.getPRICE() + "원");
return convertView;
}
```

7.3.2 MenuObj.java

```
public static Creator<MenuObj> getCreator() {
    return CREATOR;
}
String KEYINDEX;
String M_KEYINDEX;
String NAME;
String PRICE;
}
public MenuObj(String KEYINDEX, String m_KEYINDEX,
String NAME, String PRICE) {
    this.KEYINDEX = KEYINDEX;
    M_KEYINDEX = m_KEYINDEX;
    this.NAME = NAME;
    this.PRICE = PRICE;
}
//생략
```

```
private void readFromParcel(Parcel in){
    KEYINDEX= in.readString();
    M_KEYINDEX= in.readString();
    NAME= in.readString();
    PRICE= in.readString();
}
@SuppressWarnings("rawtypes")
public static final Creator<MenuObj> CREATOR = new
Creator() {
    public Object createFromParcel(Parcel in) {
        return new MenuObj(in);
    }
    public Object[] newArray(int size) {
        return new MenuObj[size];
    }
};
```

7.3.3 MenuInfoInputActivity.java

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menuinfoinput);
    init();
    //메뉴 가져오기 api
    selmenu();
}
//(생략)
private void selmenu(){
    customProgressDialog();
    map.clear();
    map.put("url", DEFINE.SERVER_URL +
    DEFINE.CUSTOMER_SEL_MENU);
    map.put("m_keyindex",
    Check_Preferences.getAppPreferences(getApplicationContext()
    , "KEYINDEX"));
}
```

```
    mThread = new
    AccumThread(getApplicationContext(), mAfterAccum,
    map, 0, 1, null);
    mThread.start();
}
View.OnClickListener btnListener = new
View.OnClickListener() {
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.plus_btn:
                //(생략)
            case R.id.savebtn:
                //(생략)
        }
    }
}
```

7.4 주문

7.4.1 OrderListAdapter.java

```
public class OrderListAdapter extends BaseAdapter {

    private Activity activity;
    private static LayoutInflater inflater = null;
    ArrayList<OrderListObj> items;
    CommonUtil dataSet = CommonUtil.getInstance();

    public OrderListAdapter(Activity a,
    ArrayList<OrderListObj> m_board) {
        activity = a;
        inflater = (LayoutInflater)
        activity.getSystemService(Context.LAYOUT_INFLATER_SE
        RVICE);
        items = m_board;
    }
    //(생략)
    public View getView(final int position, View
    convertView, ViewGroup parent) {
        final OrderListObj board =
        items.get(position);
        ViewHolder vh = new ViewHolder();
    }
```

```
        if (convertView == null) {
            convertView =
            inflater.inflate(R.layout.orderlist_item, null);
            vh.item_food_nm = (TextView)
            convertView.findViewById(R.id.item_food_nm);
            vh.item_date = (TextView)
            convertView.findViewById(R.id.item_date);
            vh.item_body = (TextView)
            convertView.findViewById(R.id.item_body);
            convertView.setTag(vh);
        } else {
            vh = (ViewHolder) convertView.getTag();
        }
        vh.item_food_nm.setText(board.getNAME());
        vh.item_date.setText(board.getDATE());
        vh.item_body.setText(board.getM_NAME() + ":
        "+ board.getPRICE() + "원");

        return convertView;
    }
}
```


7.4.2 OrderActivity.java

```
private void selmenu(){
    customProgressPop();
    map.clear();
    map.put("url", DEFINE.SERVER_URL +
    DEFINE.CUSTOMER_SEL_MENU);
    map.put("m_keyindex", obj.getKeyINDEX());

    //스레드 생성
    mThread = new
    AccumThread(getApplicationContext(), mAfterAccum,
    map, 0, 0, null);
    mThread.start();
}

private void init(){
    common_title_tv =
    (TextView)findViewById(R.id.common_title_tv);
    edit_body =
    (TextView)findViewById(R.id.edit_body);
    menuList = (ListView)findViewById(R.id.menuList);
    edit_body.setEnabled(false);

    edit_body.setText("주문해주세요.");
    common_title_tv.setText("주문");
}
```

```
findViewById(R.id.btn_order).setOnClickListener(btnL
istener);

findViewById(R.id.common_left_btn).setOnClickListener(btnListener);
}

Handler mAfterAccum = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        else if (msg.arg1 == 1) {
            customProgressClose();
            String res = (String) msg.obj;
            // res = res.replace(" ", "").replace("\n"
            , "");
            Log.e("SKY", "" + res);
            Toast.makeText(getApplicationContext() ,
            "주문이 완료 되었습니다." ,
            Toast.LENGTH_SHORT).show();
            edit_body.setText("주문해주세요.");
        }
    }
};
```

7.4.3 OrderListObj.java

```
public class OrderListObj implements Parcelable {

    public static Creator<OrderListObj> getCreator()
    {
        return CREATOR;
    }

    String NAME;
    String DATE;
    String M_NAME;
    String PRICE;

    public OrderListObj(String NAME, String DATE,
    String m_NAME, String PRICE) {
        this.NAME = NAME;
        this.DATE = m_NAME;
        this.PRICE = PRICE;
    }
    //(생략)
```

```
public void writeToParcel(Parcel dest, int flags) {

    dest.writeString(NAME);
    dest.writeString(DATE);
    dest.writeString(M_NAME);
    dest.writeString(PRICE);

}

private void readFromParcel(Parcel in){

    NAME= in.readString();
    DATE= in.readString();
    M_NAME= in.readString();
    PRICE= in.readString();
}

@SuppressWarnings("rawtypes")
public static final Creator<OrderListObj> CREATOR =
new Creator() {
    public Object createFromParcel(Parcel in) {
        return new OrderListObj(in);
    }
    public Object[] newArray(int size) {
        return new OrderListObj[size];
    }
};
}
```

7.4.4 OrderListActivity.java

```
private void init(){
    orderList =
    (ListView)findViewById(R.id.orderList);

    SelectOrderApi();

}
private void SelectOrderApi(){

    customProgressPop();
    map.clear();
    map.put("url", DEFINE.SERVER_URL +
    DEFINE.MEMBER_ORDER_LIST);
    map.put("f_keyindex",
    Check_Preferences.getAppPreferences(getApplicationContext()
    ntext(), "KEYINDEX"));

    //스레드 생성
    mThread = new
    AccumThread(getApplicationContext(), mAfterAccum,
    map, 0, 0, null);
    mThread.start();
}
```

7.5 리뷰

7.5.1 HoogiListAdapter.java

```
private Activity activity;
private static LayoutInflater inflater = null;
ArrayList<HoogiObj> items;
CommonUtil dataSet = CommonUtil.getInstance();

public HoogiListAdapter(Activity a,
ArrayList<HoogiObj> m_board) {
    activity = a;
    inflater = (LayoutInflater)
    activity.getSystemService(Context.LAYOUT_INFLATER_SE
    RVICE);
    items = m_board;
}
//(생략)
public View getView(final int position, View
convertView, ViewGroup parent) {
    final HoogiObj board = items.get(position);
    ViewHolder vh = new ViewHolder();
```

```
    if (convertView == null) {
        convertView =
        inflater.inflate(R.layout.hoogilist_item, null);
        vh.item_id = (TextView)
        convertView.findViewById(R.id.item_id);
        vh.item_date = (TextView)
        convertView.findViewById(R.id.item_date);
        vh.item_body = (TextView)
        convertView.findViewById(R.id.item_body);
        convertView.setTag(vh);
    } else {
        vh = (ViewHolder) convertView.getTag();
    }

    //
    vh.item_worker_no.setText(board.getFACE_ID());
    vh.item_id.setText(board.getID());
    vh.item_date.setText(board.getDATE());
    vh.item_body.setText(board.getREVIEW());

    return convertView;
}
}
```

7.5.2 HoogiActivity

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_hoogi);
    obj = getIntent().getParcelableExtra("obj");

    init();
    // 메뉴 가져오기 api
}

private void init(){
    common_title_tv =
    (TextView)findViewById(R.id.common_title_tv);
    body_txt = (EditText)findViewById(R.id.body_txt);
    reviewList = (ListView)
    findViewById(R.id.reviewList);

    common_title_tv.setText("후기");

    findViewById(R.id.btn_send).setOnClickListener(btnLi
stener);
    findViewById(R.id.common_left_btn).setOnClickListener
r(btnListener);
    selectList();
}
```

```
private void selectList(){
    customProgressPop();
    map.clear();
    map.put("url", DEFINE.SERVER_URL +
    DEFINE.CUSTOMER_HOOGI_SELECT);
    map.put("k_keyindex", obj.getKeyINDEX());

    // 스레드 생성
    mThread = new
    AccumThread(getApplicationContext(), mAfterAccum,
    map, 0, 1, null);
    mThread.start();
}

View.OnClickListener btnListener = new
View.OnClickListener() {
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btn_send:
                //(생략)
            case R.id.common_left_btn:
                //(생략)
        }
    }
};
```

7.5.3 HoogiObj.java

```
public class HoogiObj implements Parcelable {

    public static Creator<HoogiObj> getCreator() {
        return CREATOR;
    }

    String KEYINDEX;
    String M_KEYINDEX;
    String K_KEYINDEX;
    String REVIEW;
    String DATE;
    String ID;
    String PW;
    String NAME;
    String PHONE;
    String SHOP_KEEPER;
    String IMG;
    String EMAIL;
    String KEEPER_YN;
    String ADDRESS;
    String WI;
    String GI;
    String INTRODUCE;
    String POINT;
```

```
public HoogiObj(String KEYINDEX, String m_KEYINDEX,
String k_KEYINDEX, String REVIEW, String DATE, String
ID, String PW, String NAME, String PHONE, String
SHOP_KEEPER, String IMG, String EMAIL, String
KEEPER_YN, String ADDRESS, String WI, String GI,
String INTRODUCE, String POINT) {
    this.KEYINDEX = KEYINDEX;
    M_KEYINDEX = m_KEYINDEX;
    K_KEYINDEX = k_KEYINDEX;
    this.REVIEW = REVIEW;
    this.DATE = DATE;
    this.ID = ID;
    this.PW = PW;
} //(생략)
@SuppressWarnings("rawtypes")
public static final Creator<HoogiObj> CREATOR = new
Creator() {
    public Object createFromParcel(Parcel in) {
        return new HoogiObj(in);
    }

    public Object[] newArray(int size) {
        return new HoogiObj[size];
    }
};
```

7.5.4 ReviewActivity.java

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_review);
    init();
}

private void init(){
    reviewList =
    (ListView)findViewById(R.id.reviewList);
    selectList();
}

private void selectList(){
    customProgressPop();
    map.clear();
    map.put("url", DEFINE.SERVER_URL +
    DEFINE.MEMBER_HOOGI_SELECT);
    map.put("k_keyindex",
    Check_Preferences.getAppPreferences(getApplicationContext()
    , "KEYINDEX"));

    //스레드 생성
    mThread = new
    AccumThread(getApplicationContext(), mAfterAccum,
    map, 0, 1, null);
    mThread.start();
}
```

```
else if(msg.arg1 == 1){
    customProgressClose();
    String res = (String) msg.obj;
    // res = res.replace(" ", "").replace("\n", "");
    Log.e("SKY", "" + res);
    arr.clear();
    try {
        JSONObject jsonObject = new JSONObject(res);
        //SUCCESS
        //{ "rc": "0000", "rc_txt": "ok", "rc_data": 3397 }
        if (jsonObject.getString("rc").equals("0000"))
        {

            JSONArray dataobj =
            jsonObject.getJSONArray("rc_data");
            //채팅 db insert

            for (int i = 0; i < dataobj.length(); i++)
            {
                JSONObject resultListObject =
                dataobj.getJSONObject(i);
                HoogiObj item = new HoogiObj(

                resultListObject.getString("KEYINDEX"),
                resultListObject.getString("M_KEYINDEX"),
                resultListObject.getString("K_KEYINDEX"),

                //(생략)
            }
        }
    }
};
```

7.6 그 외

7.6.1 IntroduceActivity.java (점주용 소개페이지)

```
private void init(){
    img = (ImageView)findViewById(R.id.img);
    introduce_edit =
    (EditText)findViewById(R.id.introduce_edit);

    introduce_edit.setText(Check_Preferences.getAppPrefe
    rences(this, "INTRODUCE"));

    findViewById(R.id.img_btn).setOnClickListener(btnLis
    tener);

    findViewById(R.id.savebtn).setOnClickListener(btnLis
    tener);
}
```

```
//버튼 리스너 구현 부분
View.OnClickListener btnListener = new
View.OnClickListener() {
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.img_btn:
                //(생략)
            case R.id.savebtn:
                //(생략)
        }
    }
}
```

7.6.2 CommonUtil.java (ip, 주소, 이미지)

```

public static String getIPAddress(boolean useIPv4) {
    try {
        List<NetworkInterface> interfaces =
Collections.list(NetworkInterface.getNetworkInterfaces());
        for (NetworkInterface intf : interfaces) {
            List<InetAddress> addrs =
Collections.list(intf.getInetAddresses());
            for (InetAddress addr : addrs) {
                if (!addr.isLoopbackAddress()) {
                    String sAddr = addr.getHostAddress();
                    //boolean isIPv4 =
InetAddressUtils.isIPv4Address(sAddr);
                    boolean isIPv4 = sAddr.indexOf('.')<0;

                    if (useIPv4) {
                        if (isIPv4)
                            return sAddr;
                    } else {
                        if (!isIPv4) {
                            int delim = sAddr.indexOf('%');
// drop ip6 zone suffix
                            return delim<0 ?
sAddr.toUpperCase() : sAddr.substring(0,
delim).toUpperCase();
                        }
                    }
                }
            }
        }
    } catch (Exception ex) { } // for now eat
exceptions
    return "";
}

public static String getLocalIpAddress() {
    try {
        for (Enumeration<NetworkInterface> en =
NetworkInterface.getNetworkInterfaces();
en.hasMoreElements();) {
            NetworkInterface intf = en.nextElement();
            for (Enumeration<InetAddress> enumIpAddr =
intf.getInetAddresses();
enumIpAddr.hasMoreElements();) {
                InetAddress inetAddress =
enumIpAddr.nextElement();
                if (!inetAddress.isLoopbackAddress() &&
inetAddress instanceof Inet4Address) {
                    return inetAddress.getHostAddress();
                }
            }
        }
    } catch (SocketException ex) {
        ex.printStackTrace();
    }
    return null;
}

```

```

public static void saveBitmaptoJpeg(Bitmap bitmap,
String folder, String name){
    String ex_storage =
Environment.getExternalStorageDirectory().getAbsolute
Path();
    // Get Absolute Path in External Sdcard
    String foler_name = "/" + folder + "/";
    String file_name = name + ".jpg";
    String string_path = ex_storage + foler_name;
    Log.e("SKY", "string_path :: " + string_path);

    File file_path;
    try{
        file_path = new File(string_path);
        if(!file_path.isDirectory()){
            file_path.mkdirs();
        }
        FileOutputStream out = new
FileOutputStream(string_path + file_name);

        bitmap.compress(Bitmap.CompressFormat.JPEG,
100, out);
        out.close();

    }catch(FileNotFoundException exception){
        Log.e("FileNotFoundException",
exception.getMessage());
    }catch(IOException exception){
        Log.e("IOException", exception.getMessage());
    }
}

```

7.7 WEB

7.7.1 main.html

```
app.$mount("#app");
var activeIndex = 1;
var offsetArr = [];
$(function(){
    $(".swiper-slide").each(function(i){
        var offsetTop =
$(this).offset().top;

$(this).css("top",offsetTop+"px");
        offsetArr.push(offsetTop);
    });

$(".swiper-slide").eq(0).find(".s03").hide();
    $(".home").hammer().on("drag",
function(ev){
        if(ev.gesture.direction == "up"
){
            if(activeIndex > 0){
                var _top =
offsetArr[activeIndex]+ev.gesture.deltaY;

$(".swiper-slide").eq(activeIndex).css({"top":
_top+'px'},0);
$(".swiper-slide").eq(activeIndex).find(".s03").hide
();
            }
        }else if(ev.gesture.direction
== "down" ){
            if(activeIndex > 1){
$(".swiper-slide").eq(activeIndex-1).css({"top":
ev.gesture.deltaY+'px'},0);
            }
        }
    });
    $(".home").hammer().on("dragend",
function(ev){
        if(activeIndex > 0){

console.log(ev.gesture);
        console.log(ev.gesture.deltaY);
        if(ev.gesture.direction == "up" ){
            if(ev.gesture.deltaY < - 200){
$(".swiper-slide").eq(activeIndex).stop().animate({"
top": "0px"},400);
                if(activeIndex < offsetArr.length){
                    $(".swiper-slide").removeClass("on");
                    //$(".swiper-slide .s02").hide();
$(".swiper-slide").eq(activeIndex).addClass("on");
                    //$($(".swiper-slide").eq(activeIndex).find(".s02").fa
deIn(500);

                    activeIndex++;
                }
            }
        }
    });
}
```

```
        }else{
$(".swiper-slide").eq(activeIndex).stop().animate({"
top": offsetArr[1]+"px"},400);
        }
    }else
    if(ev.gesture.direction == "down" ){
        if(ev.gesture.deltaY < - 200){
$(".swiper-slide").eq(activeIndex-1).stop().animate(
{"top": "0px"},400);
        }else{
//
$(".swiper-slide").eq(activeIndex-1).stop().animate(
{"top": offsetArr[activeIndex-1]+"px"},400);
        if(activeIndex > 1){
$(".swiper-slide").removeClass("on");

$(".swiper-slide").eq(activeIndex-2).addClass("on");

$(".swiper-slide").eq(activeIndex-1).find(".s03").sh
ow();

        activeIndex--;
    }
    }
    }
});
```

위 코드들은 디자인 등 코드를 제외하였
고. 기능에 관련 된 코드도 전체가 아닌
일부 기능들에 관하여 일부 코드만을 기
입하였습니다.

Ⅲ. 결론

1. 연구 결과

이 시스템을 사용함으로써 요즘 찾기 힘든 푸드트럭을 자신의 위치기반으로 탐색해 볼 수 있으며 성격이 대체로 급한 한국인들에게 푸드트럭 앞에서 음식을 기다리는 시간을 절약한다는 장점을 제공해준다. 또한, 길을 가다가 우연히 발견한 푸드트럭을 이용하는 것이 아닌 이 시스템을 통하여 자신이 먹고 싶은 메뉴를 판매하는 푸드트럭을 미리 검색하여 주문하고 찾아갈 수 있다.

고객이 GPS기능을 통하여 자신 주변의 푸드트럭 목록을 제공받을 수 있으며, 푸드트럭의 메뉴 주문 후에는 음식이 나오는 소요시간이 얼마나 걸리는지 알림을 받을 수 있다. 이 알림을 통하여 사용자가 푸드트럭을 알맞은 시간에 찾아가서 음식을 바로 찾아갈 수 있다.

2. 작품제작 소요재료 목록

2.1 소프트웨어 목록

- 푸드트럭 위치 알려주기: Kakao Map, GPS 사용
- 푸드트럭 정보, 사용자 정보 저장: SQLite DB사용
- 실시간 양방향 정보주고받기: socket통신

2.2 하드웨어 목록

- 안드로이드 스마트폰