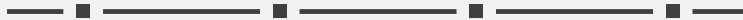




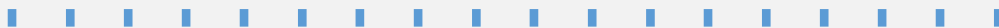
ACORN TEAM 1

FOOD CLOUD





CONTENTS



01. 프로젝트 개요



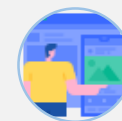
02. 팀원 소개



03. 프로젝트 설계



04. 서비스 구현



05. 시연



MANGO
PLATE



01

프로젝트 개요

FOOD + CLOUD

FOOD CLOUD 서비스는 음식을 의미하는 FOOD와 데이터가 저장되는 저장소를 의미하는 CLOUD의 합성어입니다.

FOOD CLOUD에서는 검색을 통한 다양한 음식점의 정보를 카테고리별로 제공하고, 각 음식점의 고유 페이지를 운영하여 해당 음식점에 대한 다양한 정보와 사용자가 남긴 리뷰를 확인할 수 있습니다.

또한 사용자의 서비스 이용 로그를 기반으로 수집한 데이터를 가공하여 차트 형태로 제공함으로써 다양한 볼거리를 제공합니다.

개발환경

 Windows10  spring  ORACLE  Google Cloud

BACKEND



Maven™



elastic



FRONTEND



협업도구



01

프로젝트 개요

Project 명

FOOD CLOUD

기간

2023/01/03 ~ 2023/03/31

팀원

팀장 : 조동환 | 팀원 : 박정현 서대규 송상언 김선현

개발 환경

OS : Windows10

개발 도구 : Spring Tool Suite3

DB : Oracle Database

서버 : Google Web Server

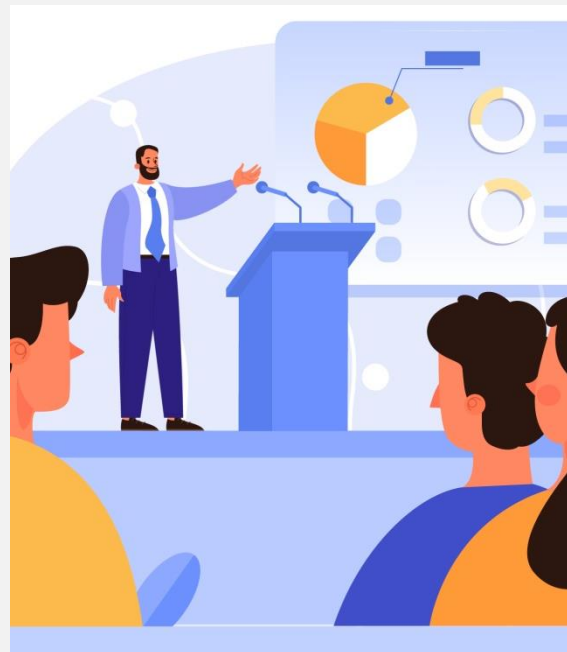
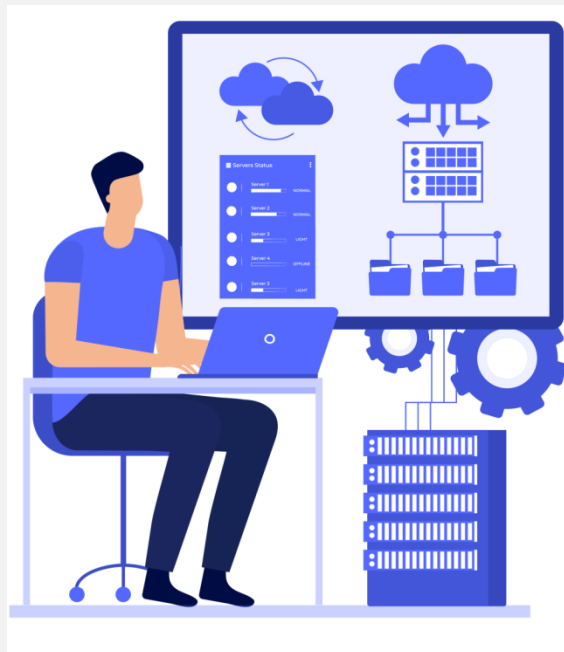
사용 기술

Back-end : Java11, Maven, MyBatis, Elasticsearch, Quantum DB

Front-end : HTML, CSS, JavaScript, JSP, AJAX, Jquery, Vue.js

Bootstrap, Chart.js

협업 도구 : GitHub, Slack, Trello



02

팀원 구성 및 역할

조동환(팀장)

- 프로젝트 진행, 유지 보수 총괄
- 문서 작성(기능 명세서, 화면 정의서)
- ES 데이터 적재, DB/JSP 연동, 음식점 상세보기 페이지 구현

김선현

- 문서 작성(ERD, ERD 가이드 문서)

박정현

- 문서 작성(기능 명세서, 결과 보고서)
- 전체 페이지 UI 구성/디자인/기능 구현
- Chart.js 활용 통계 페이지의 차트 콘텐츠 구현

서대규

- 문서 작성(기능 명세서, 화면 정의서)
- 사용자 기능(로그인, 회원가입, 사용자 관리) 구현
- ES 에러 적재 기능, 통합 검색 기능 구현

송상언

- ES 에러 적재 기능, 음식점 상세보기 페이지, DB 테이블, 채팅 구현
- GCP VM 서버, 웹 소켓 구축, Elasticsearch 서버 연동

단 계		일 정	1~4주차	5주차	6주차	7주차	8주차	9주차	10주차	11~12주차
분석 및 설계	조직 구성									
	주제 선정									
	기 획									
	ERD									
	기능 리스트									
	DB 설계									
	화면 설계									
	상세 설계									
개발	DB 구성									
	WAS									
	UI 구성									
	서비스 구축									
	사용자 서비스									
	DB 연동									
	Log 수집									
	실시간 서비스									
	데이터 시각화									
	테 슣									
	오류 수정									



프로젝트 설계

구현 설계 - 프로젝트 일정

메인 페이지	헤더	공통	헤더 바		화면 스크롤에 상관없이 상단에 고정된 메뉴		
			페이지네임/Logo		클릭 시 메인페이지로 이동한다		
			유저 메뉴 (모든 사용자) info.jsp	ID 버튼	기본 동작	1. ID 버튼 - 클릭 시 회원정보 화면으로 이동 한다.	
					P/W 변경	1. 비밀번호 변경페이지로 이동한다 2. 현재 비밀번호, 변경할 비밀번호, 변경할 비밀번호를 입력한다 3. CHANGE 버튼 클릭 시 비밀번호 변경되고 로그아웃 처리 후 로그인 페이지로 이동한다	
					이메일 변경	1. 회원 정보 페이지의 이메일 폼에 변경하고자 하는 주소를 입력한다 2. 하단의 EDIT 버튼 클릭 시 변경하고자 하는 이메일 주소가 회원 정보에 반영된다	
					회원 탈퇴	1. 회원 탈퇴 버튼 - 클릭 시 회원 탈퇴를 확인하는 창을 보여주고 사용자가 확인 시 회원 탈퇴 처리한다 - 회원 탈퇴 처리 이후 알림이 팝업되고 로그아웃 후, 로그인 페이지로 이동한다	
				로그아웃		1. 로그아웃 - 클릭 시 로그아웃처리 되고, 로그아웃 전에 보고있던 페이지는 유지된다 - 로그인 필요한 페이지라면 로그인 필요 안내 페이지로 이동한다	
				로그인 상태	음식점 등록	기본동작	1. 사용자에게 정보를 제공 할 음식점의 정보를 등록할 수 있다
						가게 이름	1. 사용자에게 정보를 제공 할 음식점의 이름을 입력한다
						카테고리	1. 사용자에게 정보를 제공 할 음식점의 분류를 입력한다 (ex. 한식, 중식, 양식 ...)
			전화번호			1. 사용자에게 정보를 제공 할 음식점의 전화번호를 입력한다	
			주소			1. 사용자에게 정보를 제공 할 음식점의 주소를 입력한다 2. 품 클릭 시 주소 검색을 할 수 있는 팝업창이 뜨고 주소 입력 후 클릭 시 기입이 완료된다	
			유저 메뉴 (관리자)		영업 시작	1. 사용자에게 정보를 제공 할 음식점의 개점시간을 입력한다 2. 오전/오후, 00시~24시, 00분~60분 으로 입력이 가능하다	
					영업 종료	1. 사용자에게 정보를 제공 할 음식점의 폐점시간을 입력한다 2. 오전/오후, 00시~24시, 00분~60분 으로 입력이 가능하다	
					가게 소개	1. 사용자에게 정보를 제공 할 음식점에 대한 간단한 설명을 입력한다	
					가게 사진	1. 사용자에게 정보를 제공 할 음식점에 대한 사진을 업로드한다	
					등록	1. 사용자에게 제공 할 음식점의 정보를 등록할 수 있다. 2. 제공되는 폼에 음식점 이름, 카테고리, 전화번호, 주소, 개점시간, 폐점시간, 설명, 사진을 입력한다. 3. 등록하기 버튼을 눌러 음식점 등록을 마친다. 4. 서비스에 가입한 사용자들의 목록을 확인/검색 할 수 있다. 5. 이용자의 ID, 이메일, 가입일, 차단 버튼, 차단 여부 를 테이블 형식으로 확인할 수 있다.	
			이용자 관리	3. 차단처리가 된 이용자의 경우 테이블에 차단 여부를 보여주는 STATUS 행에 BAN 이라고 표시된다.			
				이용자 차단	1. 서비스에 가입한 이용자에 대한 제재가 필요할 경우 이용자를 차단할 수 있다. 2. 차단 버튼을 누를 경우, "차단하시겠습니까?" 확인 창이 뜨고 확인을 누르면 이용자 차단이 완료된다. 3. 차단된 이용자와 동일한 아이디로는 가입이 불가능하다.		
			로그아웃 상태	회원 가입	1. SignUp 버튼 클릭 시 회원 가입 페이지로 이동한다. 2. 회원가입 시에는 ID, PW, PW확인, 이메일 네가지를 입력해야 한다. 각 항목에 대한 제약 조건은 아래와 같다. - ID : 영문자 소문자로 시작하고 5글자~10글자 이내 - PW : 특수문자와 숫자를 포함한 8글자 이상의 비밀번호 - 이메일 : foodcloud@acorn.com 과 같은 이메일 정규표현식 - 정규표현식을 통과하지 못하거나 공백인 입력란이 있을경우 회원가입 버튼을 누르지 못하도록 한다. 3. 입력 후 가입 버튼을 누르면 회원 가입이 완료되고, 로그인 페이지로 이동된다.		
					1. 회원 가입을 완료한 ID와 PW를 각각 입력하고 로그인 버튼을 누른다. 2. 올바른 ID와 PW를 입력했을 경우 로그인 처리되고, 로그인 버튼을 누르지 않면 이용중이던 페이지가 있다면 해당 페이지로 이동된다.		
				성공	1. 로그인 시도한 ID/PW에 해당하는 정보가 DB에 존재하지 않는 경우에 해당한다. 2. 로그인 실패 후 ID/PW를 확인해달라는 화면을 보여주는 페이지로 이동한다 3. 페이지 이동 후 하단의 로그인 페이지로 이동하기 버튼을 통해 로그인 페이지로 돌아올 수 있다.		
					로그인 실패 (ID/PW 불일치)	로그인 실패 (차단된 ID)	1. 로그인 시도한 ID/PW가 차단된 ID일 경우에 해당한다. 2. 로그인 실패 후 ID/PW를 확인해달라는 화면을 보여주는 페이지로 이동한다 3. 페이지 이동 후 하단의 로그인 페이지로 이동하기 버튼을 통해 로그인 페이지로 돌아올 수 있다.
							1. 로그인 시도한 ID/PW가 차단된 ID일 경우에 해당한다. 2. 로그인 실패 후 ID/PW를 확인해달라는 화면을 보여주는 페이지로 이동한다 3. 페이지 이동 후 하단의 로그인 페이지로 이동하기 버튼을 통해 로그인 페이지로 돌아올 수 있다.
				검색 바	검색어 입력		1. 검색어 입력 창에 검색어를 입력할 수 있다
					검색		1. 입력된 검색어를 포함하는 결과를 요청하여 통합 검색 결과 페이지로 이동한다 2. 이용자들이 실시간으로 사용할 수 있는 채팅룸 3. 좌측하단에 기본적으로 정혀있는 상태로 표시된다. 4. 로그아웃 상태라면 로그인이 필요하다는 문구를 보여주며 입력창에 텍스트 입력을 막는다. 5. 메시지 전송 시 타 유저의 메시지는 좌측, 본인의 메시지는 우측에 노란색 음영의 말풍선으로 출력된다.
		유저 채팅					

03

프로젝트 설계

구현 설계 - 기능 리스트

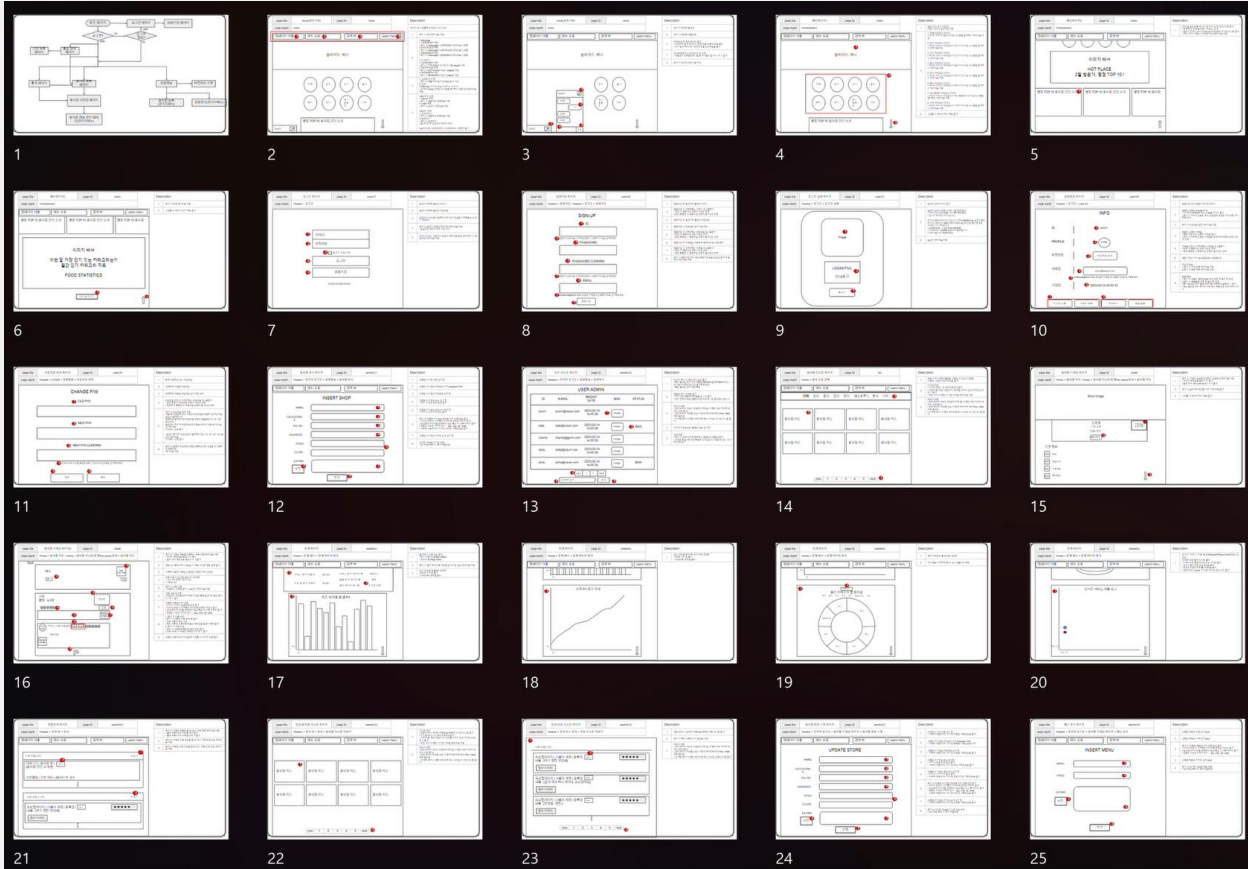
1. 구현할 기능에 대한 기능 리스트를 작성

2. 각 기능별 depth에 따라 상세 기능을 명세

자세한 내용은 아래 링크에서 확인해주세요

X

기능 리스트



프로젝트 설계

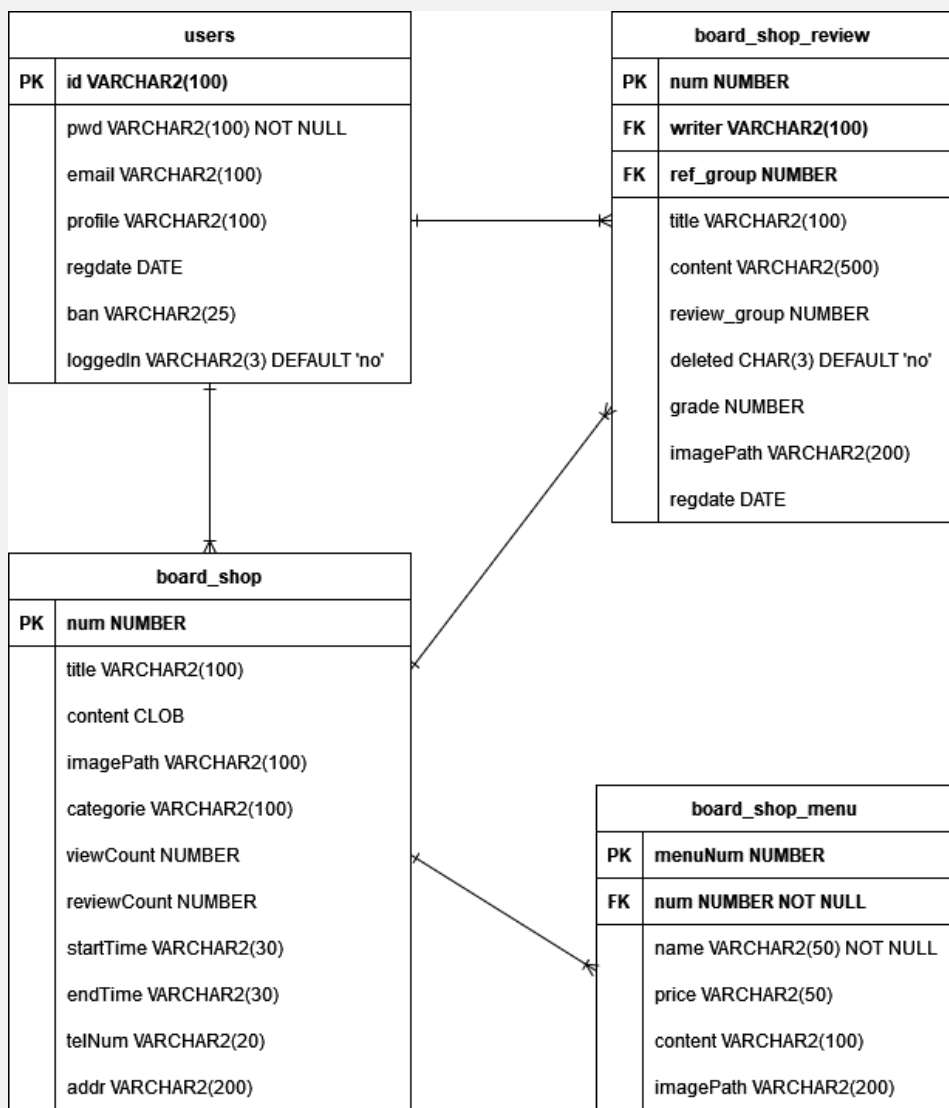
구현 설계 - 화면 정의서

1. 기능명세서를 기반으로 각 화면에 대한 화면 정의서를 작성
2. 각 화면들의 title, page id, page depth를 기재
3. 각 화면에서의 상세 동작을 우측의 Description에 기재

자세한 내용 확인은 아래링크에서 부탁드립니다.



화면 정의서



프로젝트 설계

구현 방향성 - ERD

04

서비스 구현 DATABASE

DataBase

- Quantum DB. SQL PLUS

TABLE

-USERS : 유저 Data 관리

-BOARD_SHOP : 가게 Data 관리

-BOARD_SHOP_REVIEW : 가게 리뷰 Data 관리

-BOARD_SHOP_MENU : 가게 메뉴 Data 관리





USERS TABLE

USERS					
칼럼명	데이터형	크기	NULL	PRIMARY KEY	COMMENT
ID	VARCHAR2	100	NOT NULL	YES	아이디
PWD	VARCHAR2	100	NOT NULL		비밀번호
EMAIL	VARCHAR2	100			이메일
PROFILE	VARCHAR2	100			프로필 이미지 경로
REGDATE	DATE				
LOGGEDIN	VARCHAR2	3			사용자 일일 로그인 여부
BAN	VARCHAR2	25			사용 차단 여부



BOARD_SHOP TABLE

BOARD_SHOP				
칼럼명	데이터형	크기	NULL	COMMENT
NUM	NUMBER		NOT NULL	가게 고유 번호
TITLE	VARCHAR2	100		가게 이름
CONTENT	CLOB			소개 내용
IMAGEPATH	VARCHAR2	100		대표 이미지
CATEGORIES	VARCHAR2	100		음식 분류
REVIEWCOUNT	NUMBER			리뷰 개수
STARTTIME	VARCHAR2	30		영업 시작 시간
ENDTIME	VARCHAR2	30		영업 종료 시간
TELNUM	VARCHAR2	20		전화번호
ADDR	VARCHAR2	200		주소
VIEWCOUNT	NUMBER			가게 페이지 뷰
BOARD_SHOP_SEQ				BOARD_SHOP > NUM값 관리용 시퀀스



BOARD_SHOP_REVIEW TABLE

BOARD_SHOP_REVIEW				
칼럼명	데이터형	크기	NULL	COMMENT
NUM	NUMBER		NOT NULL	리뷰의 고유 번호
WRITER	VARCHAR2	100		리뷰의 작성자 아이디
TITLE	VARCHAR2	100		리뷰 제목
CONTENT	VARCHAR2	500		리뷰 내용
REF_GROUP	NUMBER			가게 별 리뷰 구분자
REVIEW_GROUP	NUMBER			추후 기능 구현용 칼럼
DELETED	CHAR	3		리뷰 삭제 여부
GRADE	NUMBER			평점
IMAGEPATH	VARCHAR2	200		리뷰 이미지
REGDATE	DATE			리뷰 등록일
BOARD_SHOP_REVIEW_SEQ				BOARD_SHOP_REVIEW > NUM 관리용 시퀀스



BOARD_SHOP_MENU TABLE

BOARD_SHOP_MENU				
칼럼명	데이터형	크기	NULL	COMMENT
NUM	NUMBER		NOT NULL	어떤 상점의 메뉴인지를 표시
MENUNUM	NUMBER			고유번호
NAME	VARCHAR2	50		상품 이름
PRICE	VARCHAR2	50		메뉴 가격
CONTENT	VARCHAR2	100		한눈에 들어오는 짧은 글
IMAGEPATH	VARCHAR2	200		이미지 경로
BOARD_SHOP_MENU_SEQ				BOARD_SHOP_MENU > MENUNUM 관리용 시퀀스

SHOP CONTROL/SERVICE/DAO

	Shop_Control			Shop_Service		Shop_DAO	
	메소드 명	Request URL	기능	메소드 명	기능	메소드 명	기능
DB 연동	index detail reviewList	/Index /shop/detail.jsp /shop/ajax_review_list.jsp	DATA를 UI에 표시 List에 담긴 정보를 jsp파일로 전달	getList getDetail menuGetList moreReviewList	DB의 정보를 사용 가능한 형태로 가공 PAGING 처리, 검색 기능	getList getData	DB의 DATA 취득
DB 제어	insertform insert updateform updateContent reviewInsert reviewDelete reviewUpdate menuInsertform menuInsert	/shop/insertform.jsp /shop/insert.jsp /shop/updateform.jsp /shop/update.jsp /shop/detail.jsp /shop/detail.jsp /shop/detail.jsp /shop/menu_insertform.jsp	DB의 DATA를 사용 가능하도록 가공 Form을 통해 전달받은 값을 dto type으로 변환	saveContent updateContent deleteContent saveReview deleteReview updateReview saveMenu	입력된 데이터로 DB제어에 필요한 각 종 parameter를 Matching form을 통해 얻어온 정보를 정리하고, 누락된 정보를 보충하여 DB에 입력할 수 있는 형태로 완성한다.	insert delete update	mapper를 통해 Spring에서 가공 한 정보로 DB의 데이터를 제어한 다.
기타 메소드				getData countReview	DB, 사용자가 제공한 정보를 좀 더 특정 한 용도를 지닌 정보로 2차 가공	getCount getSequence getGrade	DB의 정보를 2차 가공하여 유용한 데이터로 변환하 는 역할
	imageUpload reviewImageUpload	이미지 파일 업로드 이미지 파일 저장에 위한 imagePath 산출		reviewImageUpload	업로드된 이미지의 경로를 DB에서 인 식할 수 있는 데이터로 가공		***** shop, shop_review, shop_menu의 dao는 각각 따로 존재

USER CONTROL/SERVICE/DAO

Users_Control				Users_Service		Users_DAO	
	메소드 명	Request URL	기능	메소드 명	기능	메소드 명	기능
DB 연동	info getList	/users/info /users/list	DB DATA를 UI에 표시 List에 담긴 정보를 jsp 파일로 전달	getInfo getData	DB의 정보를 사용 가능한 형태로 가공 PAGING 처리, 검색 기능	getList getData	DB의 DATA 취득
DB 제어	signup signupform update updateform pwdupdate pwdupdateform delete Profileupload login loginform logout	/users/signup /users/signinup_form /users/update /users/updateform /users/pwd_update /users/pwd_updateform /users/delete /users/profileupload /users/login /users/loginform /users/logout	DB의 DATA를 사용 가능하도록 가공 Form을 통해 전달받은 값을 dto type으로 변환	addUser updateUserPwd updateUser deleteUser saveProfileImage	입력된 데이터로 DB제어에 필요한 각 종 parameter를 Matching form을 통해 얻어온 정보를 정리하고, 누락된 정보를 보충하여 DB에 입력할 수 있는 형태로 완성한다.	insert updatePwd Update Delete	mapper를 통해 Spring에 서 가공한 정보로 DB의 데 이터를 제어한다.
기타 메소드						getCount	DB의 정보를 2차 가공하여 유용한 데이터로 변환하는 역할
	ban checkid	/users/ban /users/checkid	DB에 저장된 정보를 바 탕으로 jsp파일을 ajax 요청을 할 수있게 응답 해준다.	banUser loginProcess isExistId	DB에 저장된 데이터로 필요한 각종 파 라미터를 Matching Form을 통해 얻어온 정보를 정리하고 검증하여 DB에 입력할 수 있는 형태로 완성한다.	ban isExist	Mapper를 통해 Spring에 서 가공한 정보로 DB의 데 이터를 제어한다.

04

FOOD CLOUD 서비스 구현

0. 공통

(1) TOP 내비게이션 메뉴

- 로고 : FOOD CLOUD 서비스의 로고이자 클릭 시 메인 페이지로 이동이 가능하다
- 클릭 메뉴 : 클릭 시 페이지별로 설정된 기능(페이지 스크롤, 이동)을 실행시켜 주는 메뉴
- 검색 메뉴 : 음식점 명, 리뷰 내용 등을 검색할 수 있는 메뉴
- 회원가입/로그인 메뉴 : 서비스에 가입/로그인 할 수 있는 메뉴

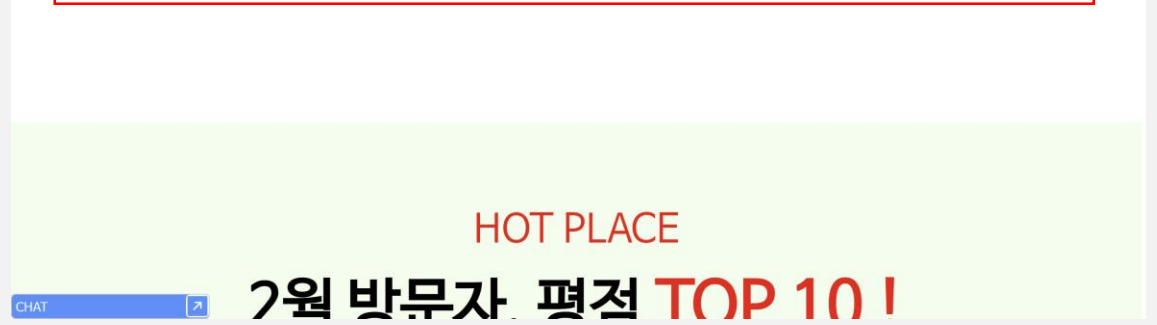
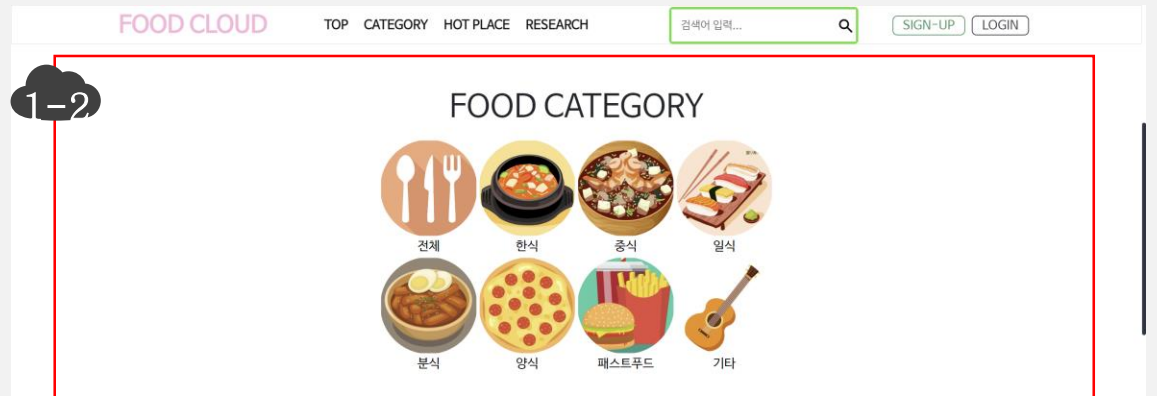
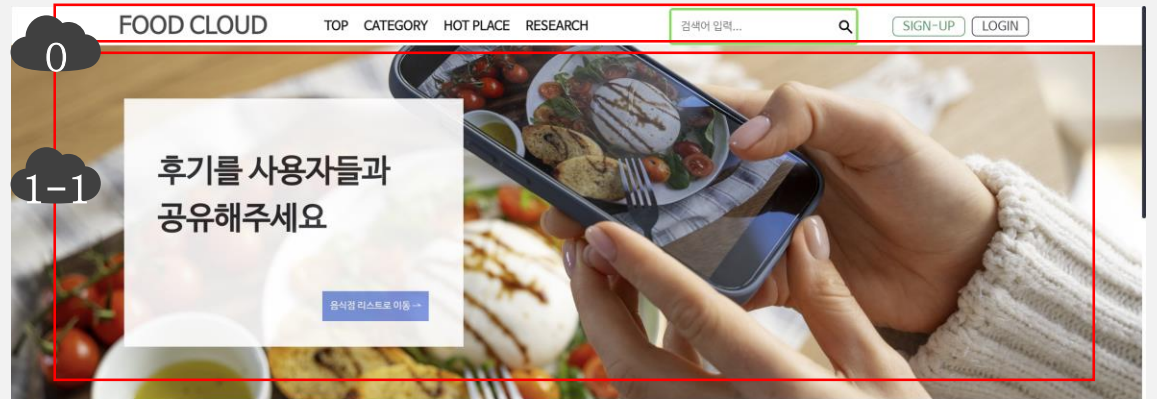
1. 메인 페이지

(1) 슬라이드 배너

- 서비스에서 제공하는 중요한 정보나 이벤트 등을 메인 화면의 슬라이드 배너를 통해 사용자에게 보여주는 메뉴

(2) 카테고리 메뉴

- 클릭 시 카테고리 별 음식점 리스트로 이동할 수 있는 메뉴



04

FOOD CLOUD 서비스 구현

1. 메인 페이지

(3) HOT PLACE 메뉴

- 서비스에 등록된 음식점 중 평점 상위 10개 음식점의 리스트를 보여주는 메뉴

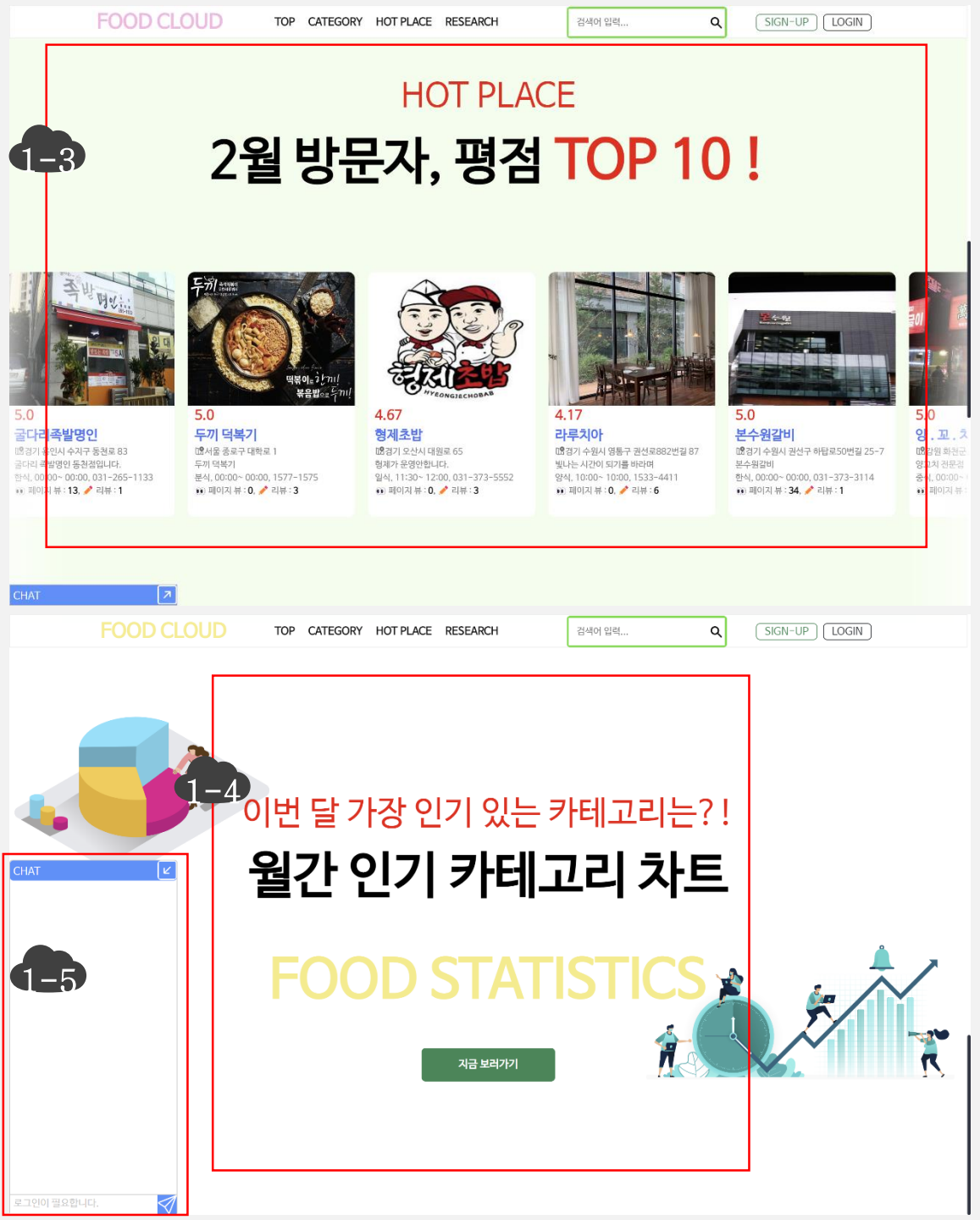
- 카드 메뉴 클릭 시 해당 음식점 페이지로 이동한다

(4) 차트 페이지 메뉴

- 서비스 이용 데이터를 가공하여 시각화 된 차트로 보여주는 페이지로 이동할 수 있는 메뉴

(5) 채팅 메뉴

- 좌측 하단의 CHAT 메뉴를 클릭 시 서비스를 이용하고 있는 사용자들과 채팅을 할 수 있는 메뉴
- 사용하지 않을 경우 접기 기능을 클릭하여 최소화
- 로그인 한 사용자만 이용 가능



04

FOOD CLOUD 서비스 구현

2. 카테고리 별 분류 페이지

(0) 기본

- 최초 화면은 전체 리스트의 페이지를 보여준다

(1) 카테고리 내비게이션 메뉴

- 특정 카테고리만의 리스트를 보여주는 메뉴

(2) 음식점 리스트

- 선택한 카테고리에 해당하는 음식점 리스트를 보여준다
- 음식점 리스트는 카드형식으로 페이지별로 8개 씩 보여준다
- 각 카드는 평점, 이름, 주소, 소개, 정보, 페이지 뷰, 리뷰로 구성
- 클릭 시 해당 음식점의 상세 페이지로 이동한다



04

FOOD CLOUD 서비스 구현

3. 가게 상세 정보 페이지

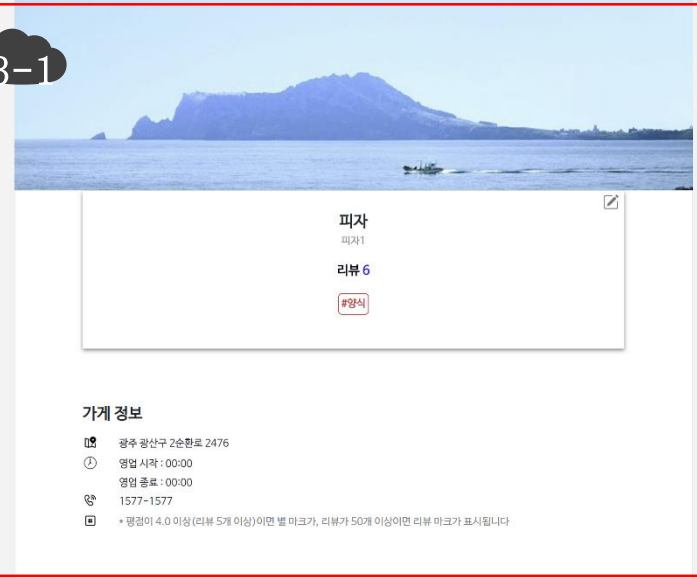
(1) 가게 정보

- 서비스에 등록된 음식점의 상세 정보를 보여주는 페이지
- 상단에는 음식점에 대한 이미지와 간략한 이름, 소개, 리뷰 수
음식점의 카테고리가 보여지는 타이틀 파트가 있다
- 다음으로는 음식점에 대한 정보(주소, 영업시간, 전화번호, 태그)를
보여주는 가게 정보 파트가 있다

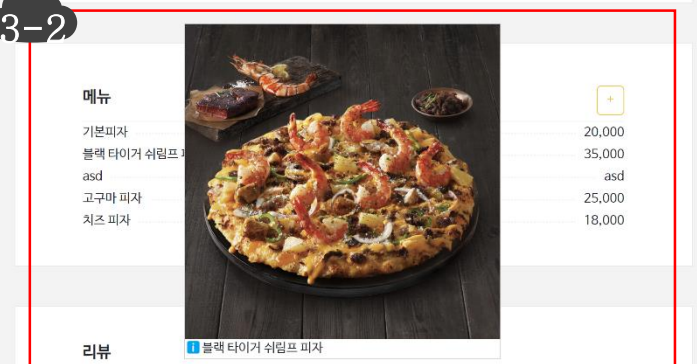
(2) 판매 메뉴

- 음식점의 판매 메뉴/가격을 보여주는 파트
- 판매 메뉴명에 마우스를 올릴 경우 대표 이미지를 보여준다
- 마우스를 바깥으로 옮기면 이미지가 사라진다

3-1



3-2



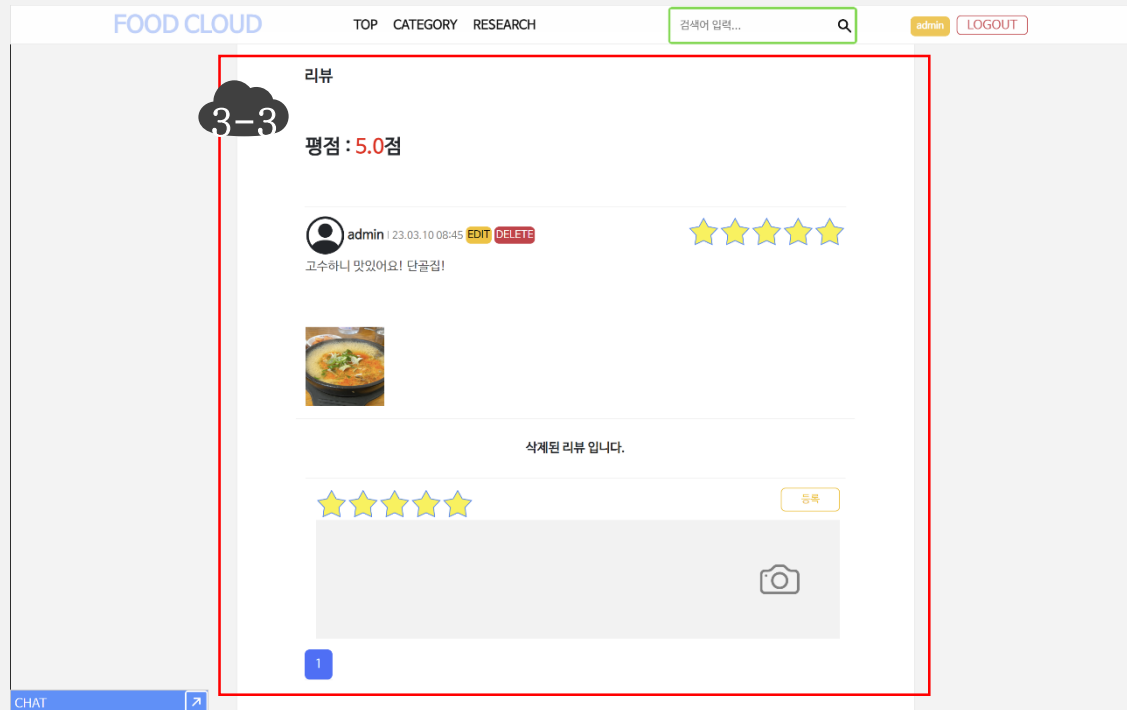
04

FOOD CLOUD 서비스 구현

3. 가게 상세 정보 페이지

(3) 리뷰

- 사용자가 음식점에 남긴 리뷰들을 볼 수 있는 파트
- 상단에는 전체 리뷰의 평균 점수를 표시해준다
- 각 리뷰들은 프로필, 아이디, 등록 시간, 별점, 내용, 사진으로 구성되어 있다
- 리뷰는 수정, 삭제를 할 수 있고, 삭제 시 '삭제된 리뷰입니다' 라고 표시된다
- 리뷰는 하단의 리뷰 등록 파트에서 등록이 가능하다
- 관리자는 사용자의 리뷰가 적합하지 않다고 판단되면 리뷰를 삭제할 수 있다





FOOD CLOUD 서비스 구현

4. 통계 페이지

(1) 서비스 이용 통계 프리뷰

- 서비스에서 수집한 각종 데이터 중 PV(page view)와 UV(Unique Visitor)를 기반으로 간략한 데이터를 보여주는 프리뷰 파트

- 월간 PV 1위 음식점 명을 클릭하면 해당 페이지로 이동한다

(2) 최근 12개월 간 월별 PV 통계

- 최근 12개월 간 누적된 월별 PV 합을 막대 차트 형태로 보여준다

(3) 누적 PV 증가 추세

- 각 월별 PV를 누적인 수치를 선형 그래프로 보여준다
- 서비스의 PV가 어느 정도 속도로 증가하고 있는지 볼 수 있다



04

FOOD CLOUD 서비스 구현

4. 통계 페이지

(4) 월간 카테고리 별 점유율

- 한달 간 카테고리 별 PV의 점유율을 최근 12개월까지 도넛 차트로 보여주는 파트

- 차트 상단의 드롭 박스를 통해 보고싶은 달을 선택하면 변경된다

(5) 사용자들의 서비스 이용 로그

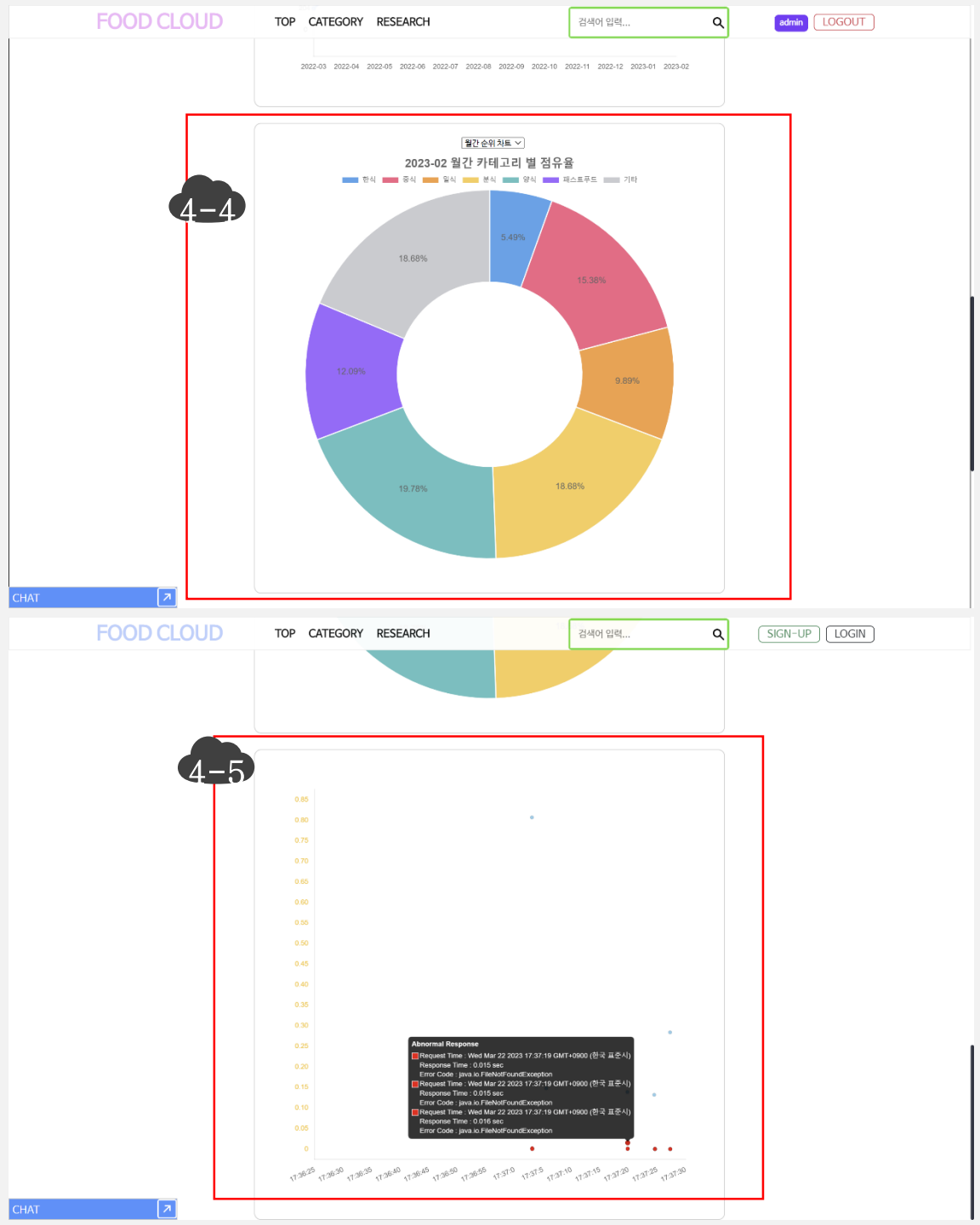
- 사용자들의 서비스 이용 로그를 차트로 보여주는 파트

- 로그는 Request/Response/Error를 보여준다

- RequestTime과 Request에 대한 ResponseTime을 확인할 수 있다

- ResponseType에 따라 Normal/Abnormal/Pending 세종류로 구분된 scatter type의 차트를 그려준다

- Normal은 파란색, Abnormal은 Error 종류와 함께 빨간색, Normal이지만 ResponseTime이 긴 Pending은 주황색으로 표시된다





05

서비스 시연

A1

팀장 조동*

부족한 부분도 많고 구현하지 못한 기능도 있어 좀 아쉬웠지만 조원들과 함께 프로젝트를 마무리 할 수 있어 뿌듯했습니다.

프로젝트 진행 당시에는 정신없이 이것 저것 알아보고 많은 코딩을 한 것 같은데 막상 프로젝트를 끝내고 기능들을 살펴보니 계획을 짰 당시에 너무 쉬운 방향으로만 설계한 게 아닌가 싶기도 했지만, 몰랐던 기능이나 ES와 같은 새로운 기술들을 스스로 공부할 기회가 있어서 좋은 경험이었다고 생각합니다.

김선*

프로젝트를 하면서 여러 것을 배웠습니다. 물론 원하던 기능이었던 특정 키워드로 유저를 찾는 방법과 같은 것 등을 만들지는 못하였지만 ERD 라는 새로운 문서를 작성하는 법을 배웠는데 확실히 써보면서 여러 질문이 생겨서 멘토님에게 많은 도움을 받았으며 결국에는 완성을 해보였습니다. 에러 코드와 관련된 코딩 체크, 채팅 앱과 관련된 코딩 등 여러 가지 코딩을 해보면서 많은 코딩과 문서 작업을 해보았습니다. 다음에도 프로젝트를 하게 된다면 더 많은 것을 포함한 목표를 만들어서 이루어 보고 싶습니다.

박정*

- 적은 인원과 시간의 압박감으로 인해 지금보다 더 다양한 기능이 갖춰진 서비스를 만들어보지 못해서 아쉽다

- 각 페이지의 script를 작성할 때 중구난방하게 여러가지의 프레임워크를 사용했는데, 다음에 진행하게 된다면 한가지로 통일하고 싶다.

- 멘토링을 통해 이끌어주는 시니어 개발자의 존재가 크다는 것을 체감했고, 지금보다 더 많은 노력을 해야겠다는 생각이 드는 프로젝트였다.

서대*

사용자 관리 기능에서 이메일 인증을 통한 ID찾기, PW 찾기 기능을 구현하지 못해서 좀 아쉬웠고 팀원들과 엘라스틱 서치와 차트js를 이용하여 그래프를 구현해본 것이 재밌었다

송상*

ElasticSearch와 DB에 대해 새로운 것을 경험해 봐서 좋았다. 이번 프로젝트에서의 시행착오 덕분에 다음에 다른 기술을 새로이 배울 때 해매는 일이 좀 더 적어지지 않을까 싶다. css를 비롯한 프론트엔드에 지나치게 기여를 안한것 같아서 프론트 엔드분들에게 미안하다. 프론트엔드 분들이 예상보다 훨씬 그럴듯한 결과물을 내주셔서 감사하다. 구현해야 할 목표도 잘 제시해주셨다. 처음 구상했던 목표치는 만족스럽게 달성했으나 더 도전적이고 진취적인 목표를 설정하는 부분에서 아쉬움이 남는다.



THANK YOU