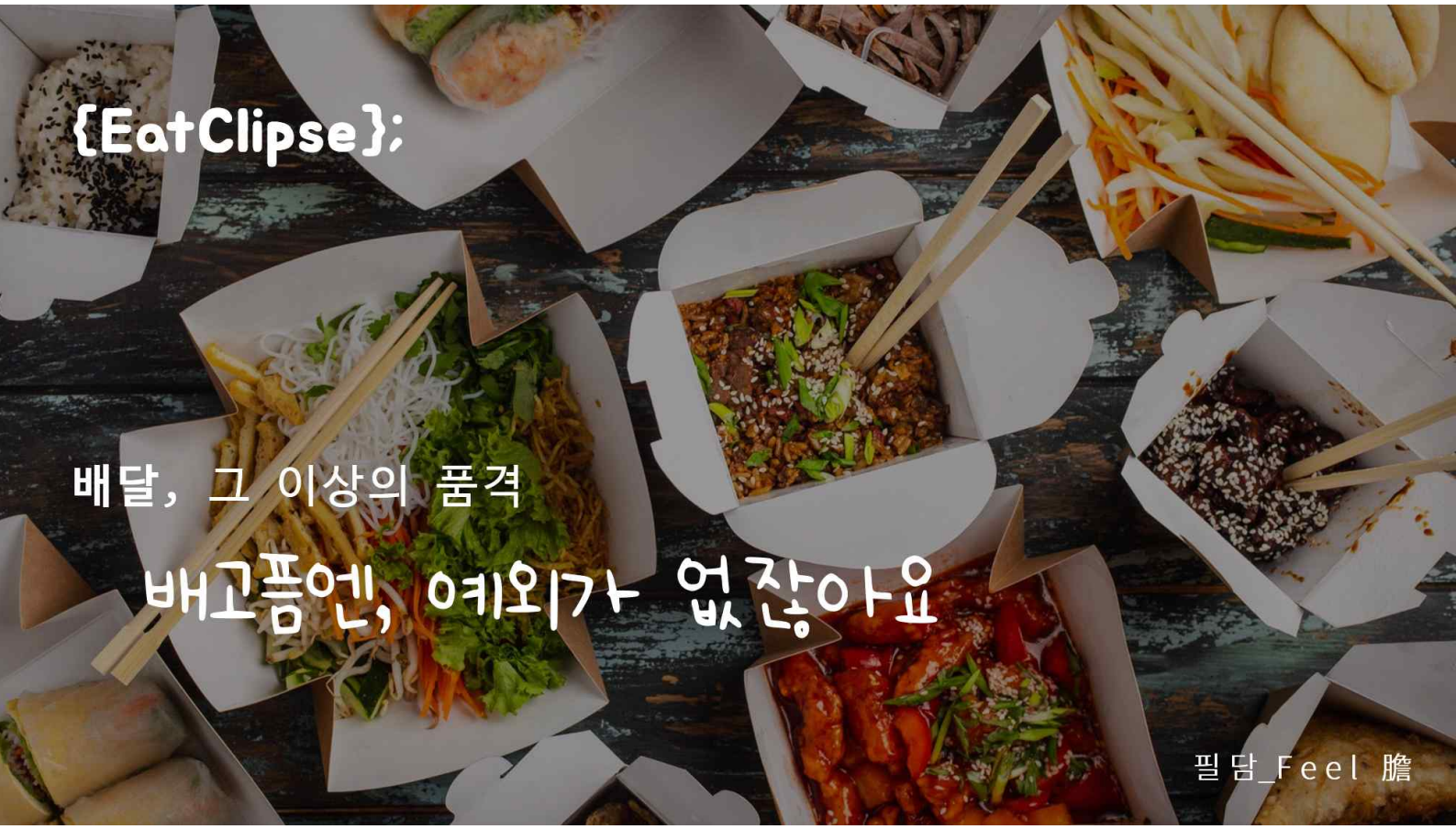
{EatClipse};

배달 웹 프로그램

[2022.01.03(월) ~ 01.14.(금)]



개 발 서

과정	프로젝트 기반 빅데이터 엔지니어 양성과정
담당 강사	이경희
팀	필담_ Feel 膽(2조)
구성원	왕찬현, 김소형, 원종현, 이여름

차 례

01 기획안 02

1. 프로젝트 개요 02
2. 개발 환경 03
3. 팀원 구성 및 역할 03
4. 개발 일정 03

02 요구사항 명세서 04

1. 메인 4
2. 회원가입 4
3. 고객 5
4. 식당 5
5. 배달 기사 6
6. 관리자 6

03 프로그램 설계 7

1. Site Map 7
2. DB Structure 8
3. Table List 8
4. Table Structure 9
5. SQL Query 12

04 UI 정의서 18

05 보완점 및 소감 25

1. 보완점 25
2. 소감 25

06 참고자료 27

1. 기획안

1. 프로젝트 개요

팀명	필담_ Feel 膽(2조)
팀원	왕찬현, 김소형, 원종현, 이여름
프로젝트명	EatClipse
개발 기간	2022.01.03 - 2022.01.14
개발 동기	▪ Covid-19의 영향으로 비대면 배달 음식에 대한 수요가 증가한 만큼, 실생활에서 유용하게 사용할 수 있는 배달 웹 프로그램 ▪ 회원 관리(3중 : 고객, 식당, 배달 기사), 상품 관리(메뉴), 게시판(리뷰) 등 교육 중 학습했던 모든 기능 포함 ▪ 자동 메뉴 추천, 메뉴 이상형 월드컵 등 도전적인 과제 확장 가능 ▪ 향후, 축적된 데이터를 활용한 추가 분석 가능 주제
프로젝트 목표	▪ 정규 과정 중 학습한 내용 체화, 응용 및 확장 ▪ 프로그래밍 역량 향상 ▪ 기획부터 개발, 디자인, 테스트, 디버깅, 개발서 작성 및 최종 발표까지 웹 프로그램 구축 프로젝트 사이클 경험 ▪ 이해관계가 얽혀있는 다양한 플레이어가 상호 이익을 극대화할 수 있는 선순환 플랫폼 구축 ▪ 사용자 친화적 프로그램 구축 ▪ 협업 툴(Git Hub) 적극적 활용
향후 확장 방향 및 기대 효과	▪ 데이터 분석을 통해 새로운 인사이트 도출 및 추가적인 서비스 방안 모색 ▪ 수집된 데이터 및 머신러닝 · 딥러닝 알고리즘을 활용하여 고객 맞춤형 메뉴 추천 기능 정교화 ▪ SNS 공유 기능을 추가하여 다양한 플레이어가 자유롭게 활동할 수 있는 플랫폼 프로그램으로서의 저변 확대

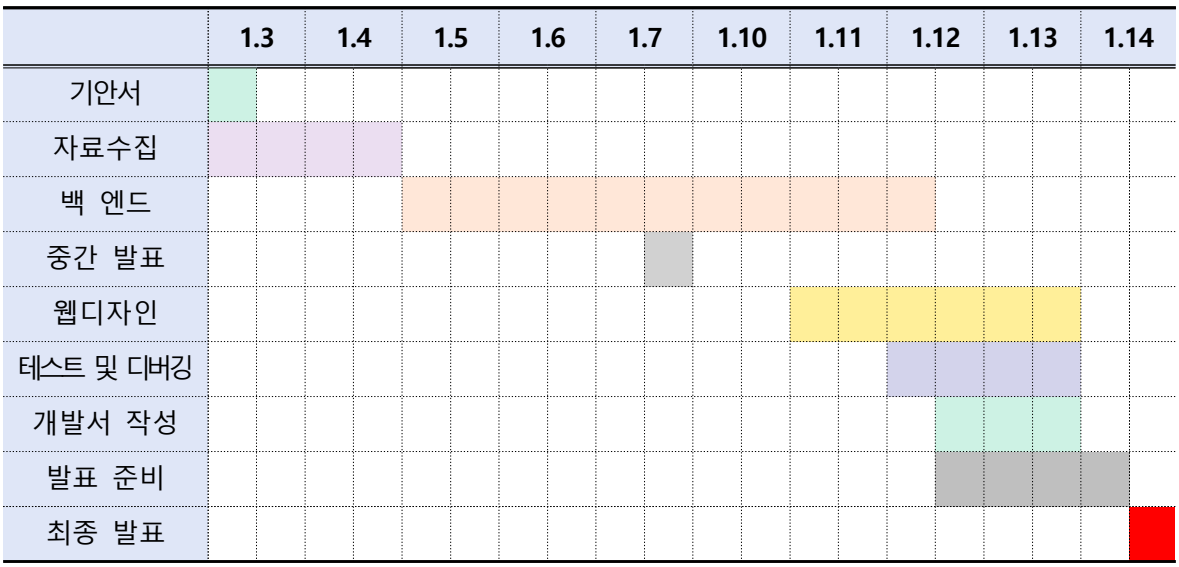
2. 개발 환경

OS	Windows10 64bit
IDE	SpringToolSuite 4.12.0
DB	MySQL 8.0CE
Language	JAVA : jdk-17.0.1 JSP, Servlet, HTML, CSS, JavaScript
Design Tool	Power Point 2016, Miri Canvas, Oven
Sever	Apach Tomcat 9.0

3. 팀원 구성 및 역할

왕 찬 현	프로젝트 총괄, 기안서 작성, 고객 기능, 개발서 작성
김 소 형	기획, 기안서 작성, 개발서 작성
원 종 현	프로그램 총괄, 관리자 기능, 배달 기사 기능, 메뉴 추천, 월드컵 기능
이 여 름	디자인, 기획, 식당 기능

4. 개발 일정



2. 요구사항 명세서

1. 메인

기능	요구사항	상세 요구사항
메인	권한에 적합한 각종 기능으로 이동	권한에 맞는 기능으로 접근 가능한 버튼을 제공, 기능 수행.
로그인	로그인을 위한 아이디와 비밀번호 입력	입력한 아이디와 비밀번호가 일치 여부 확인 로그인 성공/실패 메시지를 띄움 로그인한 사용자의 권한에 맞는 메인 페이지로 이동.
회원가입	회원가입을 위한 각종 정보입력 후 신규 회원등록	회원등록을 위한 각종 정보를 입력받고, 각 데이터의 양식에 맞는 예외 처리 실시.

2. 회원가입

기능	요구사항	상세 요구사항
고객 회원가입	회원가입을 위한 각종 정보입력 후 신규 회원등록	정보입력 후 회원가입 완료 로그인 화면으로 이동
배달 기사 회원가입	회원가입을 위한 각종 정보입력 후 신규 회원등록	정보입력 후 회원가입 완료 로그인 화면으로 이동
식당 회원가입	회원가입을 위한 각종 정보입력 후 신규 회원등록	정보입력 후 회원가입 완료 로그인 화면으로 이동

3. 고객

기능	요구사항	상세 요구사항
카테고리 선택	카테고리 : 한식, 중식, 일식, 양식, 디저트, 패스트푸드 분류	카테고리 선택 후 식당별 메뉴 출력
최근 주문한 식당	최근 주문한 식당	최근 주문한 식당을 순차적으로 나열, 선택 시 해당 식당 페이지로 이동
식당 선택	식당 상세 페이지로 접근	해당 식당에 등록된 메뉴 선택하여 장바구니 담기
마이 페이지	회원 정보	가입 시 입력한 정보 출력, 캐시 충전, 회원 정보 수정 또는 삭제 기능, 리뷰 작성
장바구니	장바구니 기능	선택한 메뉴 장바구니 담기, 삭제 충전된 캐시로 주문 및 결제

4. 식당

기능	요구사항	상세 요구사항
메뉴 등록	메뉴 등록	판매하고자 하는 메뉴, 가격, 사진 등록 등록된 메뉴 판매 중지 또는 삭제 기능
주문목록 확인	주문 확인	고객 주문 확인 후 수락 후 배달 요청 주문 완료 시간, 배달 완료 시간 출력
최근 리뷰 확인	최근 리뷰 확인	최근에 등록된 리뷰를 순차적으로 확인

5. 배달 기사

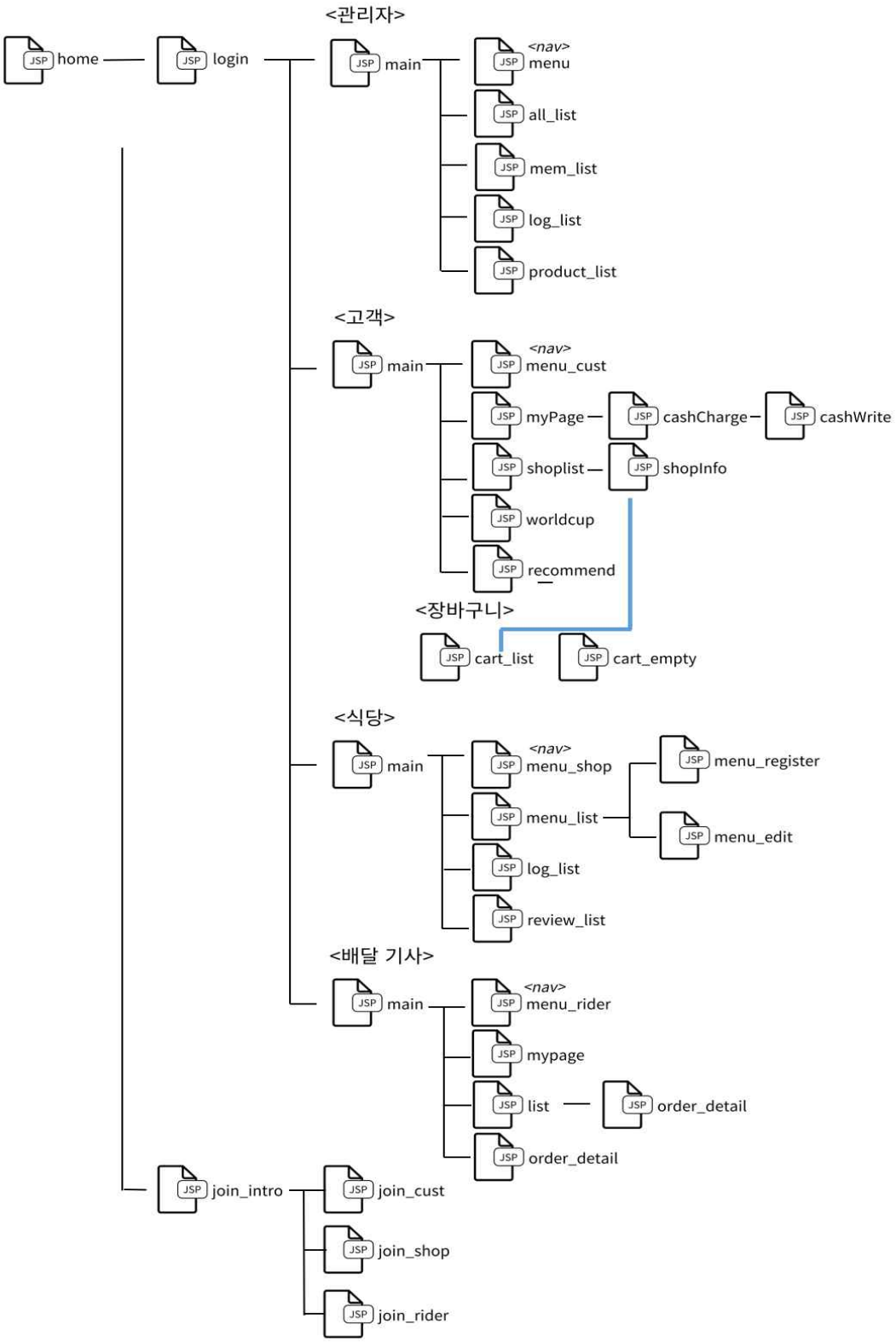
기능	요구사항	상세 요구사항
수락 가능한 배달목록	배달 요청 들어온 목록 확인	식당에서 요청한 배달목록 확인 후 선택
현재 배달 중인 목록	현재 배달목록 확인	요청 수락한 배달목록 확인 배달 완료 후 완료 선택
마이 페이지	회원 정보	회원 정보 수정, 삭제 배달 이력 및 수익금 확인 기능

6. 관리자

기능	요구사항	상세 요구사항
사용자 관리	모든 계정 확인	모든 계정 확인 및 관리
전체 로그 목록관리	로그 기록 확인	로그 기록 순차적으로 출력
메뉴 관리	메뉴 확인	모든 메뉴 확인 및 관리
추천 메뉴	메뉴 추천하기	고객의 취향에 맞춰 메뉴 추천 기능
현재 접속자 수	현재 접속자 수 확인	현재 로그인 되어있는 회원 수 확인

3. 프로그램 설계

1. Site Map



2. DB Structure

Member			
no	int	Not Null	Primary key
type	int	Not Null	
name	varchar(30)	Not Null	
userid	varchar(30)	Not Null	
passwd	varchar(50)	Not Null	
email	varchar(50)	Not Null	
money	int		
location	varchar(50)		
join_date	datetime		

Product			
no	int	Not Null	Primary key
shop_name	varchar(30)	Not Null	
prodcut_name	varchar(30)	Not Null	
price	int		
image	varchar(50)	Not Null	
type	int		

Cart			
product_no	int		
userid	varchar(30)	Not Null	
product_name	varchar(30)	Not Null	
amount	int	Not Null	
shop_name	varchar(30)	Not Null	
total_price	int		

Log			
no	int		
order_name	varchar(30)	Not Null	
shop_name	varchar(30)	Not Null	
product_name	varchar(30)	Not Null	
amount	int	Not Null	
rider_name	varchar(30)		
location	varchar(30)	Not Null	
status	int		
start_date	datetime		
end_date	datetime		
review	varchar(100)		
totalmoney	int		

3. Table List

테이블명	테이블 역할
Member	회원(관리자, 고객, 식당, 배달 기사) 정보를 저장하는 테이블
	Primary Key : id
Product	식당의 메뉴 정보를 저장하는 테이블
	Primary Key : no
Cart	고객의 장바구니 정보를 저장하는 테이블
Log	고객의 주문 및 리뷰 정보를 저장하는 테이블

4. Table Structure

4-1. Member Table

테이블명	Member			
테이블 역할	회원(관리자, 고객, 식당, 배달 기사) 정보를 저장하는 테이블			
필드 역할	필드명	Data type	Null 허용	비고
회원 고유 번호	no	int	Not Null	Primary Key auto_increment
회원 유형	type	int	Not Null	-1 : 관리자 0 : 고객 1 : 배달 기사 2 : 한식당 3 : 양식당 4 : 분식 5 : 중식 6 : 일식 7 : 디저트 가게 8 : 패스트푸드점
이름(상호명)	name	varchar(30)	Not Null	
아이디	userid	varchar(30)	Not Null	
비밀번호	passwd	varchar(50)	Not Null	
이메일	email	varchar(50)	Not Null	
충전된 금액	money	int		
주소	location	varchar(50)		
가입 일자	join_date	datetime		

4-2. Product Table

테이블명	Product			
테이블 역할	식당의 메뉴 정보를 저장하는 테이블			
필드 역할	필드명	Data type	Null 허용	비고
메뉴 고유 번호	no	int	Not Null	Primary Key
식당 상호	shop_name	varchar(30)	Not Null	Member Table의 name 필드와 동일한 값
메뉴명	product_name	varchar(30)	Not Null	
메뉴 가격	price	int		
메뉴 사진	image	varchar(50)	Not Null	
판매 여부 구분	type	int		0 : 판매 중 1 : 판매 중단

4-3. Cart Table

테이블명	Cart			
테이블 역할	고객의 장바구니 정보를 저장하는 테이블			
필드 역할	필드명	Data type	Null 허용	비고
메뉴 고유 번호	product_no	int		Product Table의 no 필드와 동일한 값
고객 아이디	userid	varchar(30)	Not Null	Member Table의 고객 userid 필드와 동일한 값
메뉴명	product_name	varchar(30)	Not Null	Product Table의 product_name 필드와 동일한 값
수량	amount	int	Not Null	
식당명	shop_name	varchar(30)	Not Null	
메뉴별 총 금액 (가격 * 수량)	total_price	int		

4-4. Log Table

테이블명	Log			
테이블 역할	고객의 주문 및 리뷰 정보를 저장하는 테이블			
필드 역할	필드명	Data type	Null 허용	비고
주문 번호	no	int		같은 주문은 동일한 값
주문자(고객)명	order_name	varchar(30)	Not Null	Member Table의 고객 name 필드와 동일한 값
식당명	shop_name	varchar(30)	Not Null	Member Table의 식당 name 필드와 동일한 값
메뉴명	product_name	varchar(30)	Not Null	Product Table 및 Cart Table의 product_name 필드와 동일한 값
수량	amount	int	Not Null	Cart Table의 amount필드와 동일한 값
배달 기사명	rider_name	varchar(30)		Member Table의 배달 기사 name 필드와 동일한 값
주소	location	varchar(30)	Not Null	Member Table의 고객 location 필드와 동일한 값
주문 상태	status	int		0 : 손님 주문 1 : 식당 주문 수락 2 : 배달 요청 3 : 배달 수락 4 : 배달 완료
주문 시간	start_date	datetime		
배달 완료 시간	end_date	datetime		
리뷰 내용	review	varchar(100)		
최종 결제 금액	totalmoney	int		

5. SQL Query

Class	AdminDAO,java	
Method	**SQL**	**Description**
alllist()[	select * from member order by no	회원 정보 출력
delete(int no)	delete from member where no = #{no}	회원 삭제
shoplist(int type)	select * from member where type>=#{type} order by no	식당 정보 출력
riderlist(int type)	select * from member where type=#{type} order by no	배달 기사 정보 출력
cuslist(int type)	select * from member where type=#{type} order by no	고객 정보 출력
loglist()	select * from log group by no order by no desc	주문 정보 출력
productlist()	select * from product order by shop_name, no	메뉴 정보 출력

Class	CommonsDAO,java	
Method	**SQL**	**Description**
login(CommonsDTO dto)	select no from member where userid = #{userid} and passwd = #{passwd}	로그인을 위한 아이디 및 비밀번호별 정보 출력
insert(CommonsDTO dto)	insert into member(type, name, userid, passwd, email, location) values(#{type}, #{name}, #{userid}, #{passwd}, #{email}, #{location})	회원 가입
search_id(CommonsDTO dto)	select userid from member where userid=#{userid}	중복된 아이디 검색을 위한 아이디별 정보 출력
view(int no)	select * from member where no=#{no}	회원 상세 정보 출력
search_name(Commons DTO dto)	select name from member where name = #{name}	중복 이름 검색을 위한 이름별 정보 출력

Class	CustDAO,java	
Method	**SQL**	**Description**
update(CommonsDTO dto)	update member set passwd=#{passwd}, name=#{name}, email=#{email} where userid=#{userid}	고객 회원 정보 수정
delete(String userid)	delete from member where userid=#{userid}	고객 탈퇴
view(String userid)	select * from member where userid=#{userid}	상세 정보 출력
cashCharge(CommonsDTO dto)	update member set money=#{money} where userid=#{userid}	캐시 충전
shopList(int type)	select * from member where type=#{type} order by no	카테고리별 식당 목록 확인
menuList(String shop_name)	select * from product where shop_name=#{shop_name} order by no	식당별 메뉴 출력
userlog(String userid)	select * from log where order_name = #{userid} group by no order by no desc	주문 목록 확인
detail(int no)	select * from Log where no=#{no}	해당 no의 상세 주문 출력
review_set(LogDTO dto)	update log set review=#{review} where no=#{no}	리뷰 작성
review_get(int no)	select review from Log where no=#{no} group by no	리뷰 조회
product_no()	select * from product where type=0	메뉴 확인
getrecent(LogDTO list)	select shop_name, image from product where shop_name=#{shop_name} and product_name=#{product_name}	최근 방문 식당 출력
getshopname(String shop_name)	select name from member where userid=#{shop_name}	식당명 출력

Class	CartDAO,java	
Method	**SQL**	**Description**
getproductname(int no)	select product_name from product where no=#{no}	메뉴명 출력
getproductprice	select price from product where no=#{no}	메뉴 가격 출력
cartlist(String userid)	insert into cart(product_no, userid, product_name, amount, shop_name, total_price) values(#{product_no}, #{userid}, #{product_name}, #{amount}, #{shop_name}, #{total_price})	장바구니 내 메뉴 목록 출력
sum_money(String userid)	select sum(total_price) as total_money from cart where userid = #{userid}	특정 메뉴 총금액 (수량 * 가격) 출력
Cartinsert(CartDTO dto)	insert into cart(product_no, userid, product_name, amount, shop_name, total_price) values(#{product_no}, #{userid}, #{product_name}, #{amount}, #{shop_name}, #{total_price})	장바구니 담기
anothershopcheck(Map <String,Object> map)	select shop_name from cart where userid = #{userid} and shop_name! = #{shop_name} LIMIT 1	동일 아이디 타 식당 검색
cart_total_price(String userid)	select sum(total_price) from cart where userid = #{userid}	장바구니 내 총 금액 계산
cartdeleteall(String userid)	delete from cart where userid = #{userid}	장바구니 전체 삭제
cartemptycheck(String userid)	select product_name from cart where userid = #{userid}LIMIT 1	장바구니 내 메뉴 삭제 확인
cartdelete(Map<String, Object> map)	delete from cart where userid = #{userid} and product_no =	장바구니 내 특정 메뉴 삭제

	#{product_no}	
getmaxno()	select max(no)+1 from log	주문 데이터 생성을 위한 로직
getshopid(String shop_name)	select userid from member where name = #{shop_name}	식당 아이디 출력
loginsert(LogDTO dto)	insert into log(no,order_name, shop_name, product_name, amount, location, status,start_date,totalmoney) values(#{no}, #{order_name}, #{shop_name}, #{product_name}, #{amount}, #{location}, #{status}, now(), #{totalmoney})	주문 데이터 생성

Class	ProductDAO,java	
Method	**SQL**	**Description**
menu_insert(productDTO dto)	insert into product(shop_name, product_name, price, image) values(#{shop_name}, #{product_name}, #{price}, #{image})	메뉴 등록
search_product_name(productDTO dto)	select product_name from product where product_name = #{product_name} and shop_name = #{shop_name}	중복 확인을 위한 메뉴명 검색
menu_list(String name)	select * from product where shop_name = #{name}	메뉴 리스트 출력
menu_no(int no)	select * from product where no = #{no}	특정 한 메뉴 출력
menu_update(productDTO dto)	update product set product_name = #{product_name}, price = #{price}, image = #{image} where no = #{no}	메뉴 수정

menu_delete(productDTO dto)	delete from product where no = #{no}	메뉴 삭제
typeto0(productDTO dto)	update product set type=0 where no = #{no}	상품 상태 변경 : 판매
typeto1(productDTO dto)	update product set type=1 where no = #{no}	상품 상태 변경 : 품절
log_list(String userid)	select * from product where shop_name = #{name}	주문 목록 출력
update_status(LogDTO dto)	select * from log group by no having shop_name = #{userid} order by start_date desc	주문 상태 변경
log_nameupdate(Map <String,Object> map)	update log set product_name = #{new_product_name} where shop_name = #{shopid} and product_name = #{old_product_name}	주문 데이터의 메뉴명 변경
review_list(String userid)	select * from log where status = 4 group by no having shop_name = #{userid} order by start_date desc	리뷰 목록 확인

Class	RiderDAO,java	
Method	**SQL**	**Description**
delivery_list()	select * from Log group by no having status = 2 order by no desc	배달 요청 목록 최신순 나열
accept_list(String name)	select * from log group by no having rider_name = #{name} and status = 3 order by no desc	배달 수락 목록 출력
accept(LogDTO dto)	update log set rider_name = #{rider_name}, status = (status+1) where no = #{no}	배달 수락

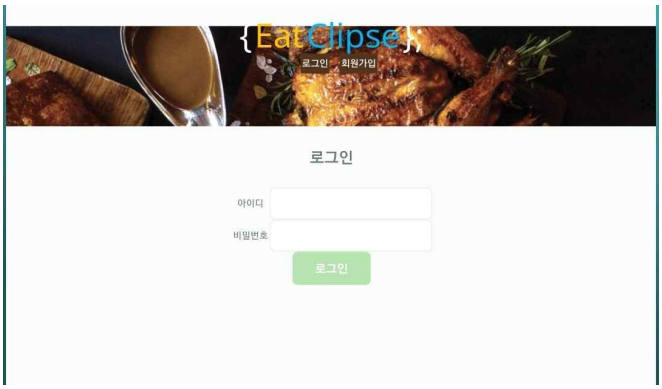
complete(int no)	update log set end_date = now(), status = (status+1) where no=#{no}	배달 완료
detail(int no)	select * from Log where no = #{no};	해당 no의 상세주문정보 출력
update(CommonsDTO dto)	update member set passwd = #{passwd}, name = #{name},email = #{email}, location = #{location} where userid = #{userid}	배달 기사 정보 수정
delete(String userid)	delete from member where userid = #{userid}	배달 기사 탈퇴
getlogdata(int no)	select price from product where shop_name = #{shop_name} and product_name = #{product_name}	주문 정보 출력
getmoney(Map<String, Object> map)	update member set money = money + #{money} where userid = #{userid}	배달 기사 배달비 입금
addmoney(Map<String, Object> map)	select userid from member where name = #{name}	배달 기사 배달비 입금을 위한 아이디 출력
getuserid(String name)	select name from member where userid = #{userid};	배달 기사 배달비 입금을 위한 이름 출력
getshopname(String userid)	select name from member where userid = #{userid};	식당명 출력
complete_list(String userid)	select * from log group by no having rider_name = #{userid} and status = 4 order by no desc	배달 기사별 배달 완료 목록 출력

4. UI 정의서

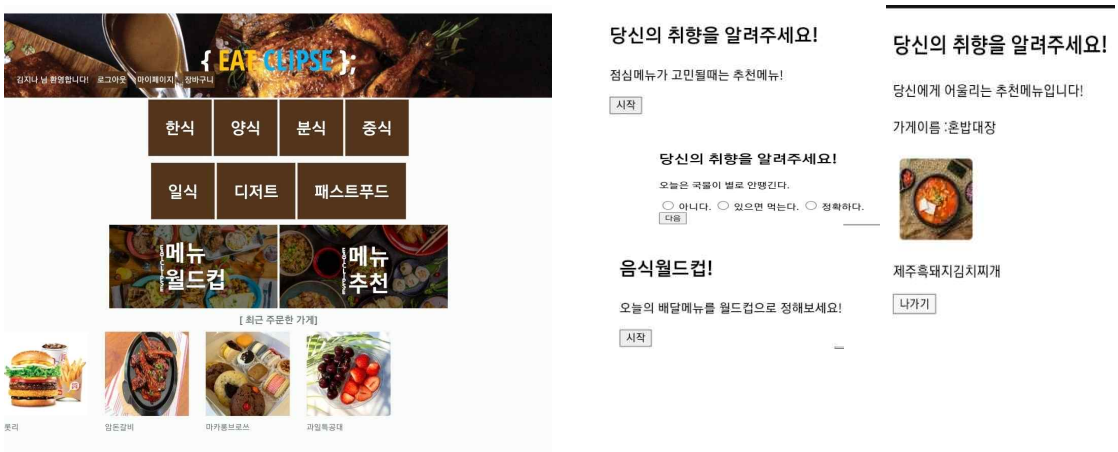
1. 회원가입 화면

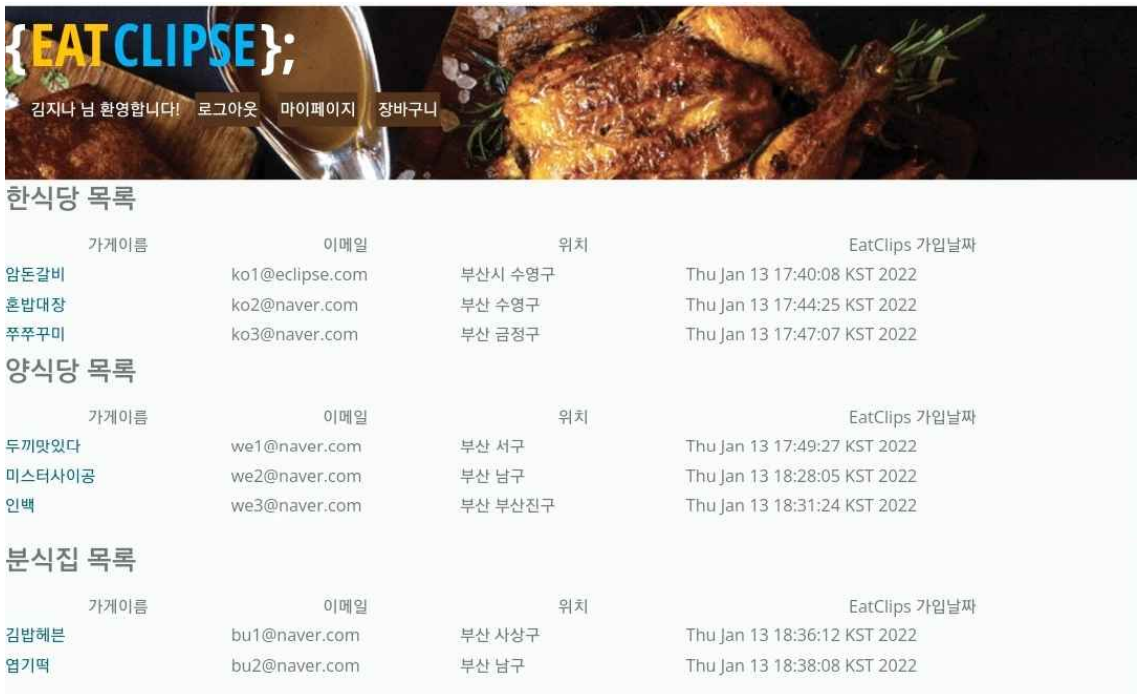
회원가입 화면	
기 능	회원가입
화 면	
기능설명	본인의 타입(손님, 식당, 배달 기사) 선택 후 ID/PW 등 정보입력 후 회원가입 완료

2. 로그인 화면

로그인 화면	
기 능	로그인
화 면	
기능설명	ID/PW 입력 후 로그인 ID/PW 불일치 할 경우 로그인 불가

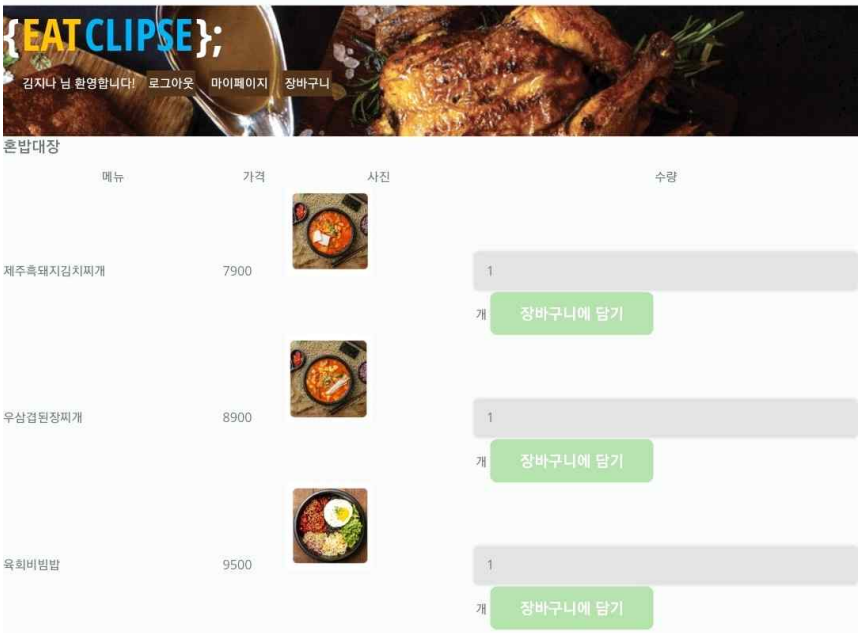
3. 고객 화면

고객 화면	
기 능	고객 메인
화 면	
기능설명	카테고리별 또는 최근 주문한 식당 선택, 메뉴 월드컵과 메뉴 추천 기능

고객 화면	
기 능	카테고리 별 식당목록 보기
화 면	

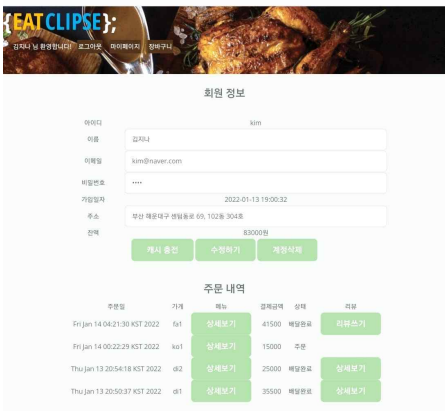
<table>
<tr><td></td><td>

</td></tr>
<tr><td>기능설명</td><td>선택한 카테고리의 식당 목록 보기</td></tr>
</table>

<table>
<tr><th colspan="2">고객 화면</th></tr>
<tr><td>기 능</td><td>식당별 메뉴보기</td></tr>
<tr><td>화 면</td><td>


</td></tr>
<tr><td>기능설명</td><td>선택한 식당의 메뉴목록 보기, 메뉴선택 후 장바구니 담기</td></tr>
</table>

고객 화면	
기 능	장바구니
화 면	 **주문이 완료되었습니다.**
기능설명	장바구니에 담겨있는 메뉴 수정, 삭제, 주문하기

고객 화면	
기 능	마이페이지
화 면	
기능설명	회원 정보 수정, 계정 삭제, 캐시 충전, 배달 완료 건 리뷰 작성

4. 식당 화면

식당 화면	
기 능	식당 관리
화 면	 [혼밥대장]관리 페이지 가게 메뉴메뉴 등록 이미지 / 메뉴 / 가격 / 상태 / 수정 제주흑돼지김치찌개 7900 판매중 판매중지 수정 우삼겹된장찌개 8900 판매중 판매중지 수정 육회비빔밥 9500 판매중 판매중지 수정 주문목록 주문번호 / 주문시간 / 메뉴이름 / 수량 / 결제금액 / 주문자명 / 주소 / 주문상태 10 22-01-14 09:47:07 육회비빔밥 1 9500 kim 부산 해운대구 센텀동로 69, 102동 304호 주문요청 주문수락 1 22-01-13 18:36:31 제주흑돼지김치찌개 3 51600 won 부산시 금정구 배달완료 • 라이더 : rider • 배달완료시간 : 22-01-13 18:40:03 최근 리뷰
기능설명	메뉴 등록, 삭제, 판매 중지 설정, 주문 수락, 배달 요청 기능, 리뷰 확인

6. 배달 기사

배달 기사 화면	
기 능	배달 기사 관리
화 면	

회원정보

아이디	rider
비밀번호	1234
이름	무면라이더입니다
이메일	rider@naver.com
잔액	14760원
지역	
가입일자	2022-01-13 18:26:34

수정

배달내역

주문자명	가게명	지역	상세보기	배달완료날짜	수익금
kim	ko2	부산 해운대구 센텀동로 69, 102동 304호	상세보기	Fri Jan 14 09:52:17 KST 2022	950.0
kim	fa1	부산 해운대구 센텀동로 69, 102동 304호	상세보기	Fri Jan 14 04:22:05 KST 2022	4150.0
kim	di2	부산 해운대구 센텀동로 69, 102동 304호	상세보기	Thu Jan 13 20:59:11 KST 2022	2500.0
lee	bbb	부산 금정구 부산대학로 9, 3층	상세보기	Thu Jan 13 19:08:46 KST 2022	2000.0
won	ko2	부산시 금정구	상세보기	Thu Jan 13 18:40:03 KST 2022	5160.0

기능설명	배달 가능한 주문목록 보기, 선택한 현재 배달목록 보기

5. 보완점 및 소감

1. 보완점

분 류	상 세 내 용
기 능	▪ 식당 검색 기능 추가 ▪ SNS 공유 기능 추가 ▪ 관리자 및 식당 회원의 통계 기능 구현 ▪ 주문 완료, 배달 시작 등 알림 기능 구현 ▪ 참조키, 외래키를 활용한 DB 및 Table 구조 개선
U I	▪ 반응형 웹 구현 ▪ 화면 구성 최적화 ▪ 기획 시, 디자인 시안 고려

2. 소감

왕 찬 현	본 프로젝트를 수행하며 얻은 개인적인 수확은 다음과 같습니다. 첫째, 개발 프로젝트의 전체 플로우를 이해하게 된 점. 특히 프로젝트는 코딩 이외에도 필수적인 기획, DB 설계, 디자인, 협업, 테스트, 디버깅, 개발서 작성 등 많은 요소를 복합적이고 유기적으로 포괄한다는 사실을 깨달았습니다. 둘째, 저의 실력을 냉정하게 파악했다는 점. 정규 과정의 수업 내용을 잘 따라가고 있다고 생각하고 있었습니다. 하지만 막상 코딩을 하며, 응용해야 할 때마다 큰 벽에 가로막혔습니다. 그만큼, 배웠던 과정을 온전히 소화하지 못하고 얄팍하게 알고 있었다는 증거였습니다. 크게 반성했습니다. 프로젝트를 통해, 내가 잘하지 못하는 부분, 잘 모르는 부분, 앞으로 더 집중해야 할 부분을 객관적으로 파악할 수 있었습니다.

	셋째, GitHub라는 협업 툴을 활용했다는 점. 개발자들이 현업 시 실제로 사용한다는 GitHub가 처음에는 무척이나 낯설고 어려웠습니다. 시간도 오래 걸리고, 심지어 하루 치 코딩을 날리기도 했습니다. 하지만, 곧 적응하여 팀원들 간에 자유롭게 협업을 진행하게 되었습니다. 개발의 협업 툴을 익혔다는 경험은 큰 자산이 되리라 생각합니다.
김 소 형	컴퓨터와 친해지는 시간이 남들보다 늦어서 수업에 뒤처진다는 느낌이 많았습니다. 하지만, 이번 프로젝트를 진행하면서 좋은 팀원들을 만나서 우리 팀원들의 많은 배려와 도움으로 또 한 번 복습할 수 있는 기회를 가질 수 있어 좋았습니다. 또한, 요즘 코로나 바이러스로 인해 사람 만나는 일이 드물고 바쁠 일이 없었는데 프로젝트를 마무리하면서 생각해 보니 다 같이 모여서 서로 배우고 의논하고 열정적으로 바쁘게 움직여서 뿌듯했던 2주였습니다. 다음 파이널 프로젝트까지 더 공부해서 제가 도움이 되는 팀원이 되겠습니다.
원 종 현	이번 프로젝트를 진행하면서 얕보고 있던 점들을 다시한번 바라보게 되었습니다. 프로젝트 시작시에는 되게 자신감이 가득차 db설계시의 관계점을 로직으로 대체가 충분히 가능할거라 생각했지만, 오만인걸 깨닫게 되었고 혼자 진행하는 프로젝트와 협업하는 프로젝트는 많이 다르다는 것을 깨닫게 되었습니다. 하지만, 이번 2주간의 프로젝트 기간동안 배운내용들을 이런 방식으로도 사용해보고 저런방식으로도 사용해보고 이렇게는 안될꺼냐는 팀원들의 질문에 답하기위해 여러 정보들을 찾아보면서 정말 많은공부가 되었습니다. 또한, 프로젝트 시작 전까지도 mvc패턴은 복습을 진행하지 않았는데 정말 알차게 복습이 되었습니다. 주로 복습을 제대로 못한내용이나, 머릿속에서 애매모호하게 남아있던 부분들을 사용하면서 내용이 정리가 되었기에 다음 프로젝트 진행시, 몇단계 더 높은 완성도를 가질 수 있다는 자신감이 생겼습니다. 그래도 자바스크립트와 css부분에서는 자신감이 생기지 않기에 좀 더 공부를 해야겠다는 생각을 가지게 되었습니다. 프론트 엔드는 관심을 적게 두고 있어서 자연스레 뒷전으로 밀려나게 되었지만, 백 엔드에서도 어느 정도 지식을 갖추고 있어야 된다는 점도 깨닫게 되었습니다.

이 여 름	코딩을 한 번도 해보지 않은 비전공자로서 두 달간 배운 지식으로 프로젝트를 하는 것이 쉽지는 않았지만, 컨트롤러를 중심으로 정보가 이동하는 방식을 제대로 익혔습니다. 웹 개발 프로젝트의 전체적인 흐름을 원활히 이어가고 이를 구현할 수 있게 되었고 웹 개발에 있어 폭넓고 확실한 이해를 통해 내가 원하는 방향에 맞는 코드를 쓰는 법을 배웠습니다. 무엇보다 좋은 팀원들을 만나 서로 소통하고 질문하는 과정에서 개념을 확실하게 짚고 넘어갈 수 있었으며, 서로의 코드 리뷰를 통해 다양한 방식으로 효율적인 코드를 작성할 수 있었습니다. 겁나기만 했던 오류 페이지와 콘솔을 보면서 어디서 오류가 났는지 함께 찾고 고치는 과정이 공부에 가장 큰 도움이 되었던 것 같습니다. 프로젝트를 하면서 공부와 복습을 함께 하게 되다 보니 속도가 더뎌, 부가 기능 구현에 많이 투자하지 못했던 점이 가장 아쉽습니다. db 구조적 한계로 다양한 구현을 하지 못했기 때문에 웹 개발에서 db 구성에 더욱 많은 투자를 해야 함을 배웠습니다. 앞으로의 과정에서 배우는 다양한 툴과 언어들을 통해 백 엔드에서 탄탄하게 구성을 잡고, CSS를 자유롭게 구현하여 더욱 완성도 높은 프로젝트를 진행해보고 싶습니다.

6. 참고자료

1. 배달의 민족 : https://www.baemin.com/
2. 요기요 : https://www.yogiyo.co.kr
3. 쿠팡 이츠 : https://www.coupangeats.com/
4. W3Schools : https://www.w3schools.com/

{EatClipse};
배고픔엔, 예외가 없으니까
필담_Feel 膽