

(01-4) 중간 결과 보고서

Flo:be

플라워 카페 상품 주문 및 클래스 예약 서비스

P01 주제 및 개요

**개발 배경 및 목적**

플라워 카페의 홍보, 판매, 클래스 예약을 통합한 홈페이지
운영 및 관리의 불편함 해소, 소비자의 접근성 증가, 꽃 시장의 발전 도모

**특징 및 장점**

명확한 틀 제공: 운영시간, 가격 등의 정보가 명시되어 있음
편리한 검색: 꽃의 종류, 꽃의 이름, 꽃말, 가격 등을 쉽게 찾을 수 있음
직관적인 화면 구성: 한 화면에서 다양한 꽃을 비교해볼 수 있음
접근성이 좋음: 꽃을 쉽게 구매할 수 있고 검색했을 때 홈페이지에 접근하기 용이함

P02

주요 기능 및 특징

메인 페이지

로그인/회원가입

사용자는 상품 구매와 클래스 예약 등의 활동을 하려면 로그인 해야 한다. 회원가입 시 판매자인지 구매자인지 구분한다.

상품/클래스 정보

상품과 클래스에 대한 정보를 찾아볼 수 있다.

문의/리뷰 정보

상품에 대한 문의와 리뷰 목록을 확인할 수 있다.

판매자 기능

고객 관리

판매자는 고객 목록을 확인하고 고객의 정보를 수정하거나 삭제할 수 있다.

상품 관리

상품 목록에서 상품을 추가하거나 수정, 삭제할 수 있다.

클래스 관리

클래스 목록에서 클래스를 추가하거나 수정, 삭제할 수 있다.

구매자 기능

상품 주문

구매자는 장바구니를 통해 상품을 주문하고 주문 내역에서 주문 정보를 확인할 수 있다.

클래스 예약

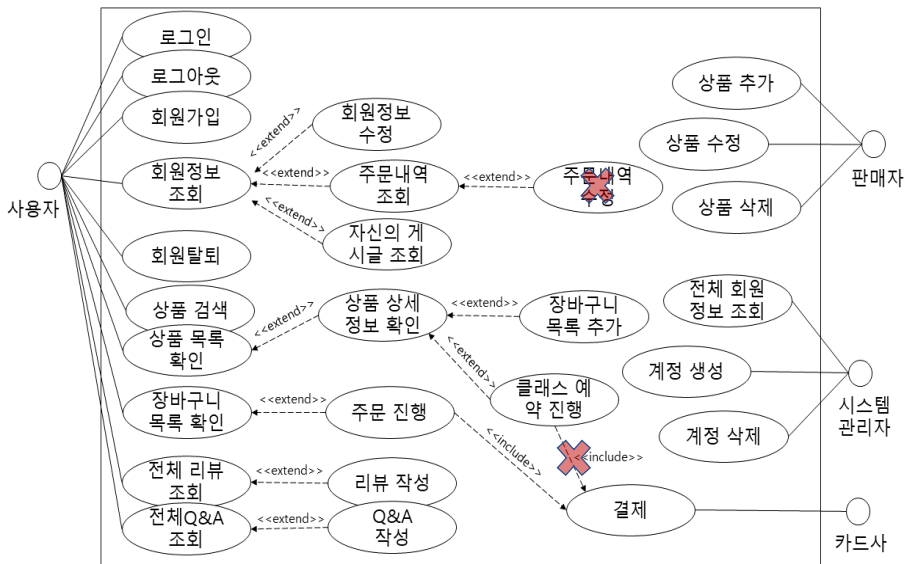
현재 열려 있는 클래스를 확인하고 예약할 수 있다.

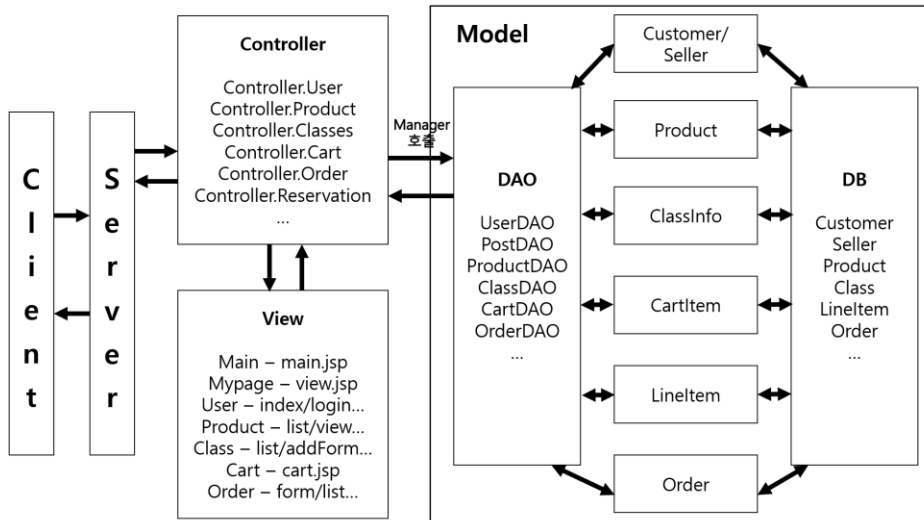
상품 문의 / 리뷰

게시판에 상품에 대한 문의사항과 리뷰를 작성할 수 있다.

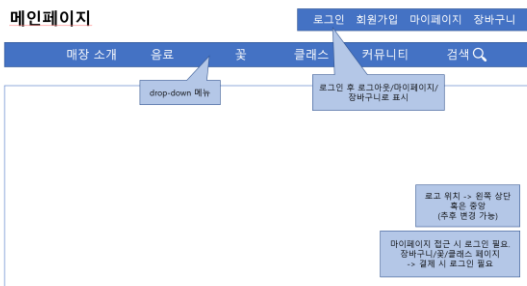
P03

요구사항 분석 결과 (Use Case)

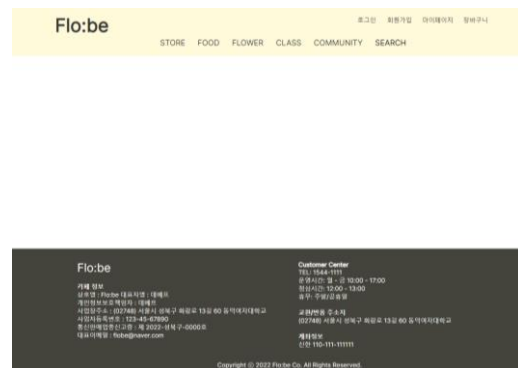




P04 UI 설계 (메인페이지)



UI 구상



구현 결과

P04 UI 설계 (로그인)

로그인

회원 로그인

아이디
비밀번호

로그인

아직 회원이 아니세요?
회원가입

로그인 성공시
메인화면으로 이동로그인 실패시
경고창 띄움

UI 구상

로그인

아이디
비밀번호

로그인

아직 회원이 아니신가요?

회원가입을 하시면 회원에게만 제공되는 다양한 혜택과
이벤트에 참여하실 수 있습니다.

회원가입

구현 결과

P04 UI 설계 (회원가입)

회원가입

회원 가입

아이디*
비밀번호*
비밀번호 확인
이름*
전화번호*
이메일*
주소
우편번호
도로명 주소
상세 주소

4~16자,
영문 소문자/숫자8~16자,
영문 대문자/숫자/특수문자

가가오 API 사용

Select box 사용

이메일 형식(@.com)에
적합인지 확인조건에 맞지 않을 경우,
경고창 띄우고 가사 초점

회원가입

UI 구상

회원 가입

아이디* (영문 소문자/숫자, 4~16자)
비밀번호* (영문 대문자/숫자/특수문자, 8~16자)
비밀번호 확인*
이름*
전화번호* 010- - - -
이메일*
주소
우편번호 주소
도로명 주소
상세 주소

회원가입

구현 결과

P04 UI 설계 (상품등록)

상품 페이지 (판매자)

삭제 버튼 클릭 시 UI의 편집 버튼>삭제 버튼으로 변경
각 상품에 대해 삭제 버튼 클릭 시 재확인 팝업 이후 삭제 처리

추가 삭제

사진

이름 카테고리

가격

주문수량 재고수량

편집

사진

가격

주문수량 재고수량

편집

UI 구상

추가

상품명	가격	
test	10000	수정 삭제
sunflower	9990	수정 삭제
flower_1	5000	수정 삭제
flower_3	5000	수정 삭제
flower_5	5000	수정 삭제
flower_5	5000	수정 삭제
해바라기	10000	수정 삭제
해바라기	10000	수정 삭제
2	2	수정 삭제

클래스 목록

추가

클래스 이름	날짜	현재인원	전체인원	
test2	2022-11-18 15:49:37	10		수정 삭제
test3	2022-11-18 15:52:26	0	10	수정 삭제
test4	2022-11-19 00:00:00	0	50	수정 삭제

구현 결과

P04 UI 설계 (상품등록)

상품 추가(수정)

수정 시 원래 상품의 정보가 쓰여지 있게

사진

이름

카테고리

가격

재고

사진 삽입

확인 취소

UI 구상

상품 검색

상품 이름을 입력하세요

장바구니에 담기

상품명 test

가격 10000

설명 aaaa

클래스 수정

클래스 이름 test2

날짜 2022-11-18

최대인원 10

현재인원 10

 [클래스 목록](#)

구현 결과

P04

UI 설계 (장바구니)

장바구니

장바구니 UI 구성

당긴 상품이 없는 경우 비어있다고 표시

주문에서 선택

이미지 상품정보 가격 수량 주문금액 배송비 합계

수량 변경 시 수량에 맞게 총 금액 변경

결제 시 실제 확인 할것함 띄우기

총 상품 금액 주문 금액 결제 예정 금액

선택 상품 주문 견제 상품 주문

UI 구상

CART

선택지	상품명	가격	수량	합계
<input type="checkbox"/>	커피	10000원	[1+]개	원
<input type="checkbox"/>	테스트	10000원	[3+]개	원
<input type="checkbox"/>	커피	10000원	[5+]개	원
<input type="checkbox"/>	sunflower	9990원	[1+]개	원
<input type="checkbox"/>	테스트	10000원	[2+]개	원
<input type="checkbox"/>	sunflower	9990원	[1+]개	원
<input type="checkbox"/>	제비꽃	10000원	[4+]개	원
<input type="checkbox"/>	제비꽃	10000원	[3+]개	원
<input type="checkbox"/>	flower_1	5000원	[8+]개	원
<input type="checkbox"/>	sunflower	9990원	[2+]개	원

총 상품 금액	결제 예정 금액
원	원
선택 상품 주문	견제 상품 주문

구현 결과

P04

UI 설계 (상품결제)

결제 페이지 (2)

주문자 정보

주문자 정보

주문자 성명 *

휴대폰 번호 * 선택 v- -

이메일 * 직접입력 v

UI 구상

직접 수형 <- 선택에 따라 품 비활성화
배송 추가

주문 상품

상품 정보	수량	상품 금액	결제 금액
-------	----	-------	-------

주문자 정보

주문자 이름 *

휴대폰 번호 *

이메일 *

구현 결과

P04

UI 설계 (상품결제)

결제 페이지 (4)

배송 정보

☐ 주문자 정보와 동일

 네오박스 체크 시
주문자 이름, 번호가
자동으로 박스를 채움

받으시는 분 *	<input type="text"/>
휴대폰 번호 *	선택 v <input type="text"/> - <input type="text"/> - <input type="text"/>
전 * 	선택 v <input type="text"/> - <input type="text"/> - <input type="text"/>
배송지 주소 *	우편번호 <input type="text"/> <input type="button" value="우편번호검색"/>
	기본주소 <input type="text"/> 나머지 주소 <input type="text"/>
배송 메모	<input type="text"/>

UI 구상

☐ 주문자 정보와 동일

배송 정보

추가

받으시는 분 *	<input type="text"/>
휴대폰 번호 *	<input type="text"/> <input type="button" value="우편번호"/> <input type="button" value="주소검색"/>
배송지 주소 *	기본주소 <input type="text"/> 나머지 주소 <input type="text"/>
메모	<input type="text"/>

상품 합계 금액:

0

배송비:

3000

총 결제 금액:

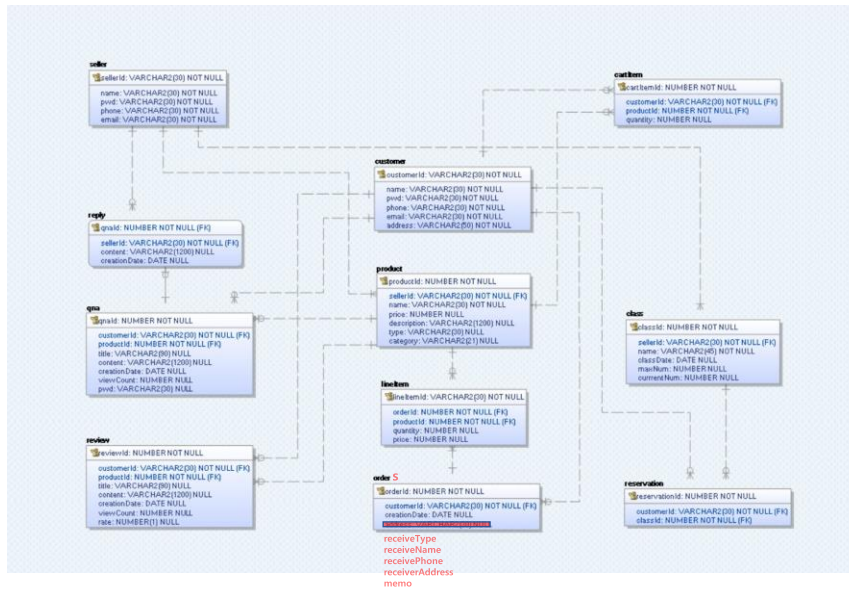
3000

결제하기

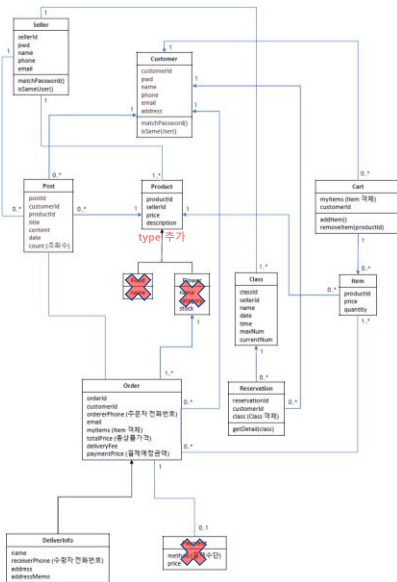
구현 결과

P04

데이터베이스 설계



P04 CLASS 설계 (Domain/Controller)



UserController

- ViewUserController
- LoginController
- LogoutController
- ListUserController
- RegisterUserController
- UpdateUserController
- DeleteUserController
- UserSessionUtils

ProductController

- ViewProductController
- ManageProductController
- SearchProductController
- ListProductController

ClassController

ListClassController
ManageClassController

CartController

ViewCartController
ManageCartController

OrderController

OrderController
ListOrderController

P04 CLASS 설계 (Manager)

UserManager

customerid, pwd

```
createCustomer(Customer)
updateCustomer(Customer)
removeCustomer(customerId)
findCustomer(customerId)
findCustomerList()
login(customerId, pwd)
```

ProductManager

productId, name, type

```
add(Product)
update(Product)
delete(productId)
findProduct(productId)
findProductList(type)
findProductByName(name)
```

ClassManager

classId

```
add(ClassInfo)
update(ClassInfo)
delete(classId)
findClassInfo(classId)
findClassList()
```

CartManager

cartItemId, customerId

```
addItem(CartItem)
removeItem(cartItemId)
getCartItem(cartItemId)
getCartItemList(customerId)
```

OrderManager

orderId, quantity,
productId

```
addOrder(Order,  
quantity, productId)  
findOrder(orderId)  
findOrderList()
```


P04

CLASS 설계 (DAO)

UserDAO

customerId, pwd, name,
phone, email, address

createCustomer(Customer)
updateCustomer(Customer)
removeCustomer(customerId)
findCustomer(customerId)
findCustomerList()
existingCustomer(customerId)

ProductDAO

productId, sellerId, name,
price, description, type,
category

add(Product)
update(Product)
remove(productId)
findProduct(productId)
findProductList(type)
findProductByName(name)

ClassDAO

classId, sellerId,
name, date,
maxNum,
currentNum

add(ClassInfo)
update(ClassInfo)
delete(classId)
findClassList()
findClass(classId)

CartDAO

cartItemId, quantity,
customerId, productId

addItem(CartItem)
removeItem(cartItemId)
getCartItem(cartItemId)
getCartItemList(customerId)

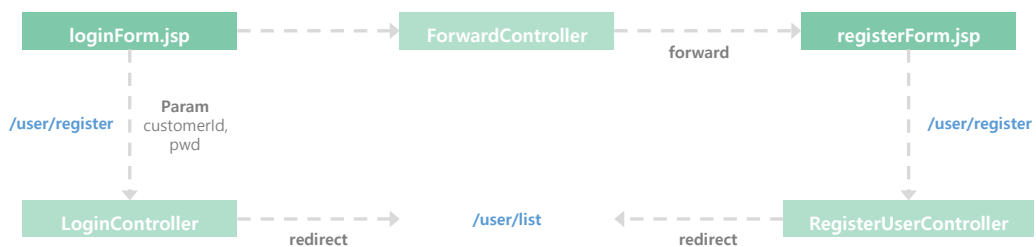
OrderDAO

orderId, customerId,
creationDate, receiveType,
receiverName, receiverPhone,
memo, receiverAddress, price,
lineItemId, productId, quantity

addOrder(Orders, quantity,
productIds)
findOrder(orderId)
findOrderList()
findOrderList(customerId)

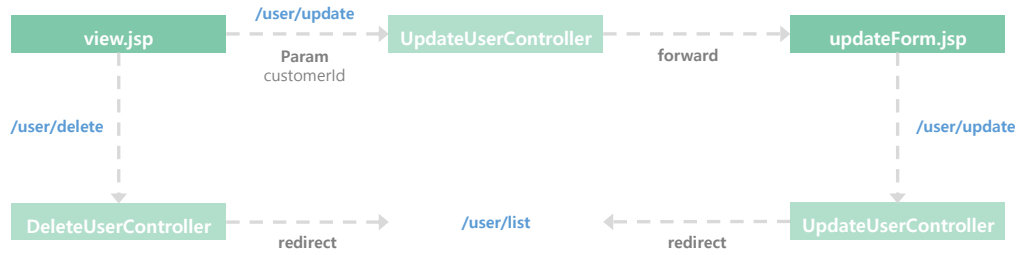
P04

Request 처리 흐름

로그인, 회원가입

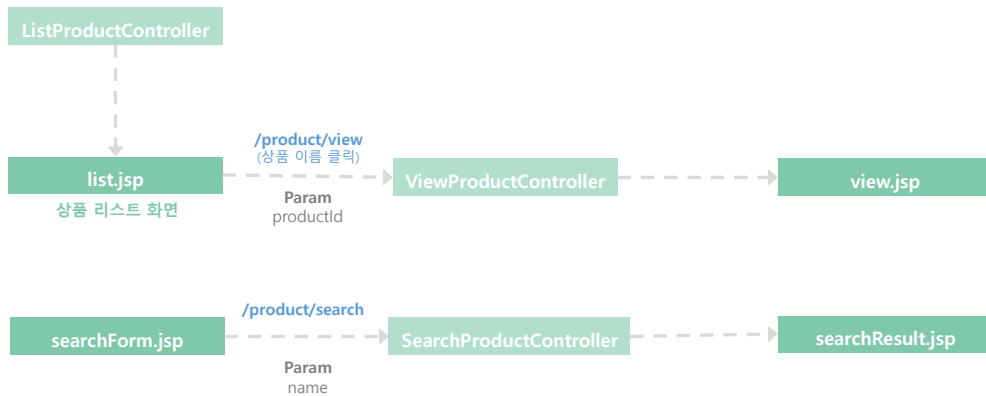
P04 Request 처리 흐름

회원 정보 수정/탈퇴



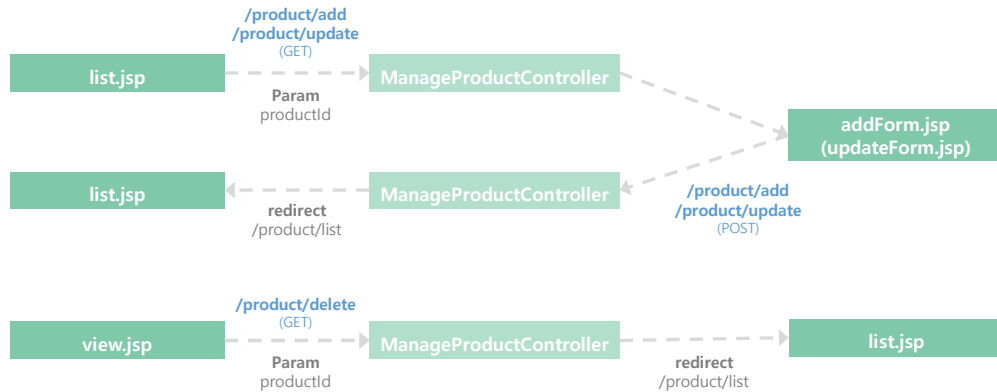
P04 Request 처리 흐름 (수정)

상품 상세조회/검색



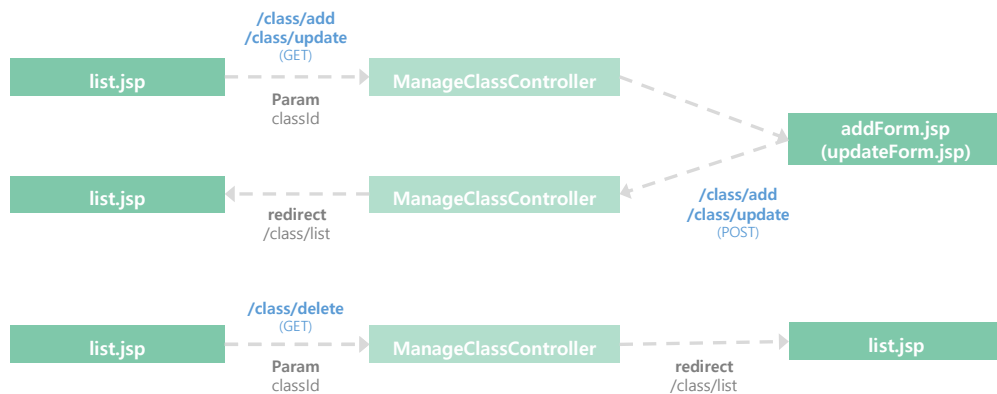
P04 Request 처리 흐름 (수정)

상품 추가/수정/삭제



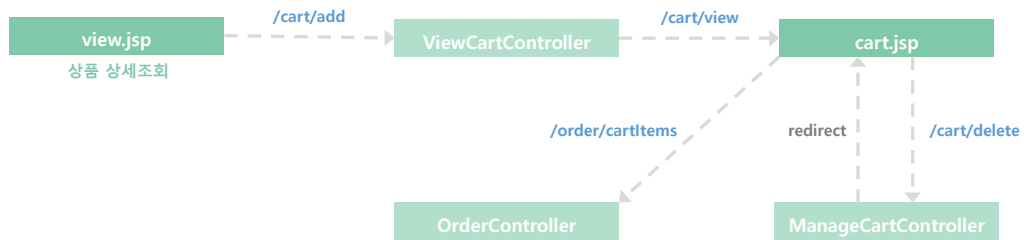
P04 Request 처리 흐름 (수정)

클래스 추가/수정/삭제



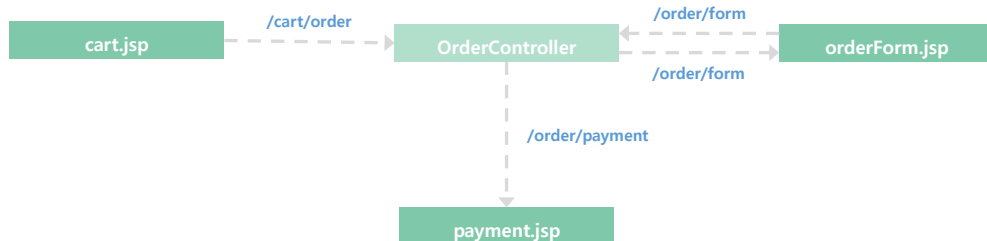
P04 Request 처리 흐름 (수정)

장바구니 상세조회, 추가/삭제



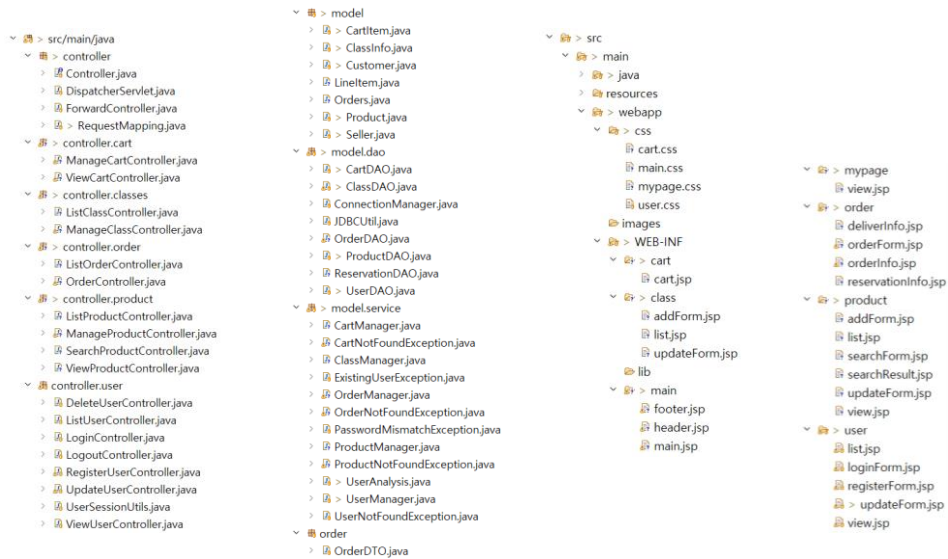
P04 Request 처리 흐름 (수정)

상품 주문, 결제



P05

Eclipse 프로젝트 구조



P05

구현된 세부 기능

1

회원가입 및 로그인/로그아웃, 회원 정보 수정 및 탈퇴

2

상품/클래스 검색 및 추가, 상품/클래스 정보 수정 및 삭제

3

장바구니 상품 추가 및 삭제, 상품 결제

P05 소스코드 예 (상품 검색)

SearchProductController

```

1 package controller.product;
2
3 import java.util.List;
4
15 public class SearchProductController implements Controller {
16     private static final Logger log = LoggerFactory.getLogger(ManageProductController.class);
17
18     @Override
19     public String execute(HttpServletRequest request, HttpServletResponse response) throws Exception {
20         ProductManager manager = ProductManager.getInstance();
21
22         // form 요청
23         if(request.getMethod().equals("GET")){
24             log.debug("SearchProductForm Request");
25             return "/product/searchForm.jsp";
26         }
27
28         // 검색 결과 요청
29         String name = request.getParameter("name");
30         List<Product> productList = manager.findProductByName(name);
31
32         request.setAttribute("productList", productList);
33         log.debug("productList: " + productList.toString());
34         return "/product/searchResult.jsp";
35     }
36
37 }

```

P05 소스코드 예 (상품 등록/수정/삭제)

ManageProductController

```

1 package controller.product;
2
3 import javax.servlet.http.HttpServletRequest;
4
15 public class ManageProductController implements Controller {
16     private static final Logger log = LoggerFactory.getLogger(ManageProductController.class);
17     private static final String USER_SESSION_KEY = "userId";
18
19     @Override
20     public String execute(HttpServletRequest request, HttpServletResponse response) throws Exception {
21         ProductManager manager = ProductManager.getInstance();
22
23         // 추가
24         if(request.getServletPath().equals("/product/add")) {
25
26             // GET: 상품 정보 추가 form 요청
27             if(request.getMethod().equals("GET")){
28                 log.debug("ProductAddForm Request");
29                 return "/product/addForm.jsp";
30             }
31
32             // POST: 상품 추가 (상품정보하기 parameter로 전송됨)
33             Product product = new Product(
34                 0, // productId
35                 request.getParameter("name"),
36                 Integer.parseInt(request.getParameter("price")),
37                 request.getParameter("description"),
38                 request.getParameter("type"),
39                 request.getParameter("category"));
40
41             HttpSession session = request.getSession();
42             product.setSellerId((String)session.getAttribute(USER_SESSION_KEY));
43             product.setSellerId("seller0"); // 테스트를 위해 임시로 판매자 id 지정
44
45             try {
46                 manager.add(product);
47                 log.debug("Create Product: " + product.toString());
48                 return "redirect:/product/list?type=" + request.getParameter("type");
49             } catch (Exception e) {
50                 e.printStackTrace();
51                 return "/product/addForm.jsp";
52             }
53         }
54
55         // 수정
56         else if(request.getServletPath().equals("/product/update")) {
57
58             // GET: 상품 정보 수정 form 요청
59             if(request.getMethod().equals("GET")) {
60                 int updateId = Integer.parseInt(request.getParameter("productId"));
61                 Product product = manager.findProduct(updateId);
62                 log.debug("ProductUpdateForm Request");
63
64                 // 여기에 해당 상품의 판매자인지 확인하는 코드 추가: if문으로 감싸기
65                 request.setAttribute("product", product);
66                 return "/product/updateForm.jsp";
67             }
68
69             // 수정이 불가능할 경우 /product/list로 이동
70             // IllegalStateException
71         }
72
73         // POST: 상품 수정 (상품정보하기 parameter로 전송됨)
74         Product product = new Product(
75             Integer.parseInt(request.getParameter("productId")),
76             request.getParameter("name"),
77             Integer.parseInt(request.getParameter("price")),
78             request.getParameter("description"),
79             request.getParameter("type"),
80             request.getParameter("category"),
81             request.getParameter("sellerId"));
82
83         manager.update(product);
84         log.debug("Update Product: " + product.toString());
85         return "redirect:/product/list?type=" + request.getParameter("type");
86     }
87
88     // 삭제
89     else {
90         int deleteId = Integer.parseInt(request.getParameter("productId"));
91
92         // 해당 상품의 판매자인지 확인하는 코드 추가
93         manager.delete(deleteId);
94
95         return "redirect:/product/list?type=flower";
96     }
97
98 }
99 }

```

P06

업무 분담

이채윤

20201008

팀장
프로젝트 진행Order 정보
주문 목록 조회
주문자/배송 정보
결제Class 정보
클래스 예약

김효민

20200962

일정 관리
자료준비 및 발표Product 정보
상품 검색
상품 추가/수정/삭제
상품 목록/상세조회Class 정보
클래스 추가/수정/삭제
클래스 목록/상세조회

이시윤

20200998

협업 톨 안내
발표 자료 제작User 정보
로그인, 회원가입
회원 수정/탈퇴
회원 목록/상세조회Post 정보
게시글 작성/수정/삭제
게시글 목록/상세조회

이승연

20201001

수행일지 작성
자료준비 및 발표

메인 페이지

마이 페이지
(주문내역/문의내역)Cart 정보
장바구니 추가/삭제

감사합니다