

(01-4) 최종 결과 보고서

Flo:be

플라워 카페 상품 주문 및 클래스 예약 서비스

1

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

2

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 정보

상품에 대한 문의 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

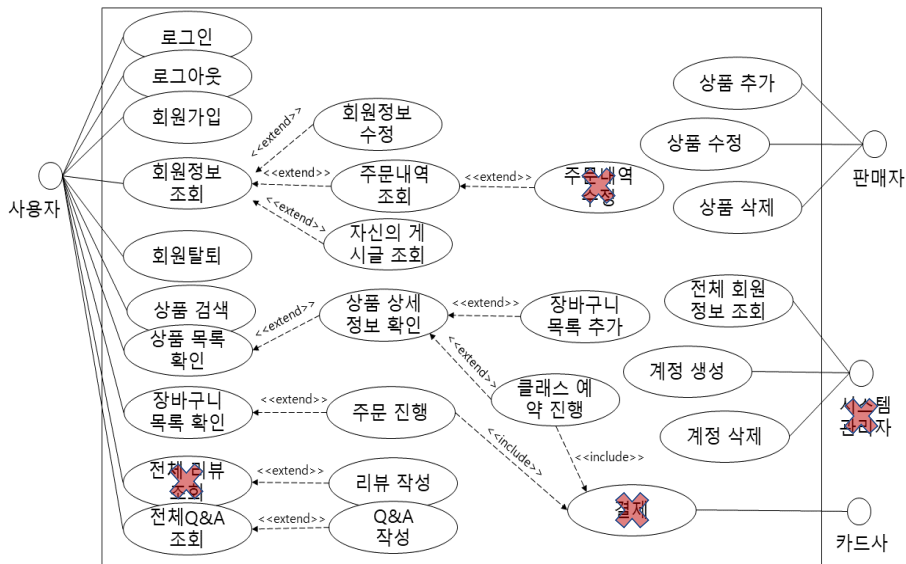
현재 열려 있는 클래스를 확인하고 예약할 수 있다.

상품 문의

게시판에 상품에 대한 문의사항을 작성할 수 있다.

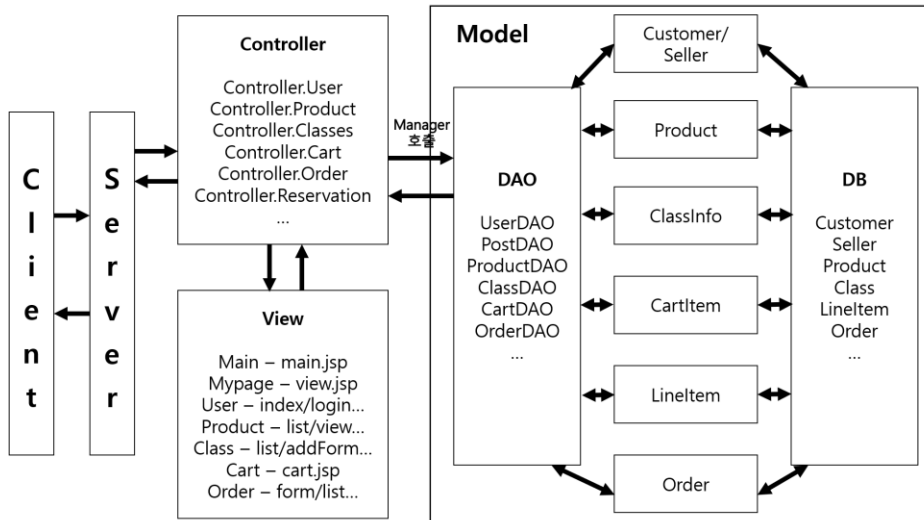
3

P03 요구사항 분석 결과 (Use Case)



4

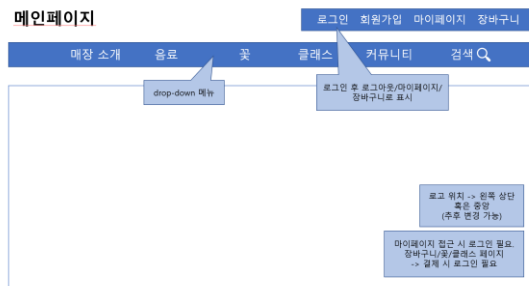
P04 시스템 구성도



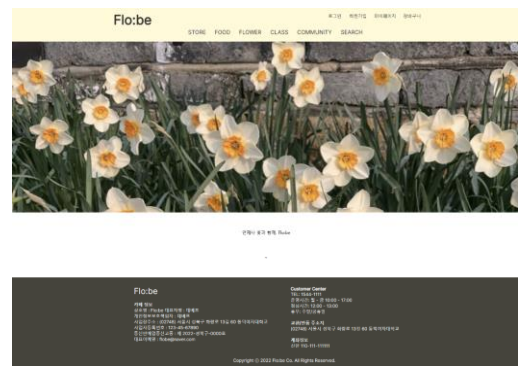
5

P04 UI 설계 (메인페이지)

메인페이지



UI 구상



구현 결과

6

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

7

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

가까이 API 사용

Select box 사용

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

회원가입

UI 구상

회원 가입

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

회원가입

구현 결과

8

P04 UI 설계 (마이페이지)

마이페이지 (1)



27

UI 구상



구현 결과

9

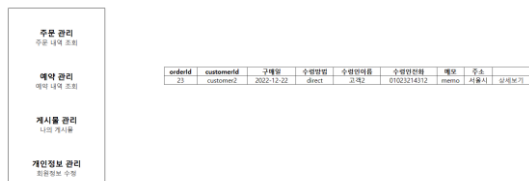
P04 UI 설계 (마이페이지)

마이페이지 (2)



28

UI 구상



구현 결과

10

P04 UI 설계 (상품등록)

상품 페이지 (판매자)

삭제 버튼 클릭 시 10의 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

사진

가격

주문수량

재고수량

편집

UI 구상

상품 목록

상품 이름을 클릭하면 상세 정보를 확인할 수 있습니다.

상품명	가격	옵션	옵션
test	10000	옵션	옵션
sunflower	9990	옵션	옵션
flower_1	5000	옵션	옵션
flower_3	5000	옵션	옵션
flower_5	5000	옵션	옵션
flower_5	5000	옵션	옵션
flower_5	5000	옵션	옵션
해바라기	10000	옵션	옵션
해바라기	10000	옵션	옵션
장미	100000	옵션	옵션

상품 추가

구현 결과

11

P04 UI 설계 (상품등록)

상품 추가(수정)

수정 시 원래 상품의 정보가 쓰여지 있게

사진 이름 카테고리 가격 재고

사진

이름

카테고리

가격

재고

사진 삽입

확인 취소

UI 구상

상품 추가

상품 이름 상품 이름

상품 가격 상품 가격

상품 설명 상품 설명

종류 선택 (종류)

카테고리 카테고리

추가

구현 결과

12

P04

UI 설계 (상품결제)

결제 페이지 (2)

주문자 정보

주문자 성명 *

휴대폰 번호 * 선택 v -

이메일 * 직렬입력 v

UI 구상

☐ 직접 수령
☐ 배송

주문 상품

상품 정보	수량	상품 금액	결제 금액
test	1	10000	10000
test	1	10000	10000

주문자 정보

주문자 이름 * asdf

휴대폰 번호 * 01012341234

이메일 * asdf@naver.com

구현 결과

15

P04

UI 설계 (상품결제)

결제 페이지 (4)

배송 정보

☐ 주문자 정보와 동일

네로박스 체크 시
주문자 이름, 번호가
자동으로 박스를 채움

받으시는 분 *

휴대폰 번호 * 선택 v -

배송지 주소 * 선택 v -

우편번호 우편번호검색

기본주소 나머지 주소

배송 메모

UI 구상

☐ 주문자 정보와 동일
☐ 배송 정보

받으시는 분 *

휴대폰 번호 *

배송지 주소 *

우편번호

기본주소

나머지 주소

배송 메모

☐ 주문자 정보와 동일
☐ 배송 정보

받으시는 분 *

휴대폰 번호 *

배송지 주소 *

우편번호

기본주소

나머지 주소

배송 메모

상품 결제 금액

20000

배송비

0

총 결제 금액

20000

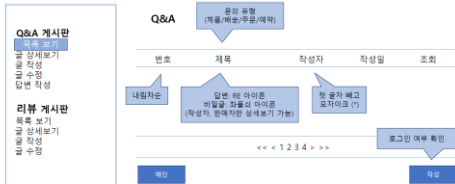
[결제하기]

구현 결과

16

P04 UI 설계 (QnA)

게시판 (1)



17

UI 구상

Q&A

No.	Title	Name	Date	View
4	문의 문의	customer11	2022-12-22 18:31:36	0
5	제품 문의	customer5	2022-12-22 18:32:56	0
3	제품 문의	abc1231	2022-12-22 18:31:01	0
7	제품 문의	customer3	2022-12-22 20:31:34	0
6	제품 문의	customer2	2022-12-22 20:28:36	0
1	제품 문의	customer2	2022-12-22 14:37:00	0
2	제품 문의	customer5	2022-12-22 17:58:42	0

1/10

1/10

구현 결과

17

P04 UI 설계 (QnA)

게시판 (2)



18

UI 구상

Q&A

Date	2022-12-22 18:31:36
Title	7
Content	문의 문의
제품, 제품 문의?	

LIST

DELETE

MODIFY

REPLY

구현 결과

18

P04 UI 설계 (QnA)

게시판 (3)

Q&A 게시판

- 목록 보기
- 글 상세보기
- 글 작성
- 답변 작성

리뷰 게시판

- 목록 보기
- 글 상세보기
- 글 작성
- 글 수정

Q&A

Select box
(제품/배송/주문/예약)

제목

비밀글 ☐ 공개글 ☒

작성 취소

19

UI 구상

Q&A

Title: 배송 문의

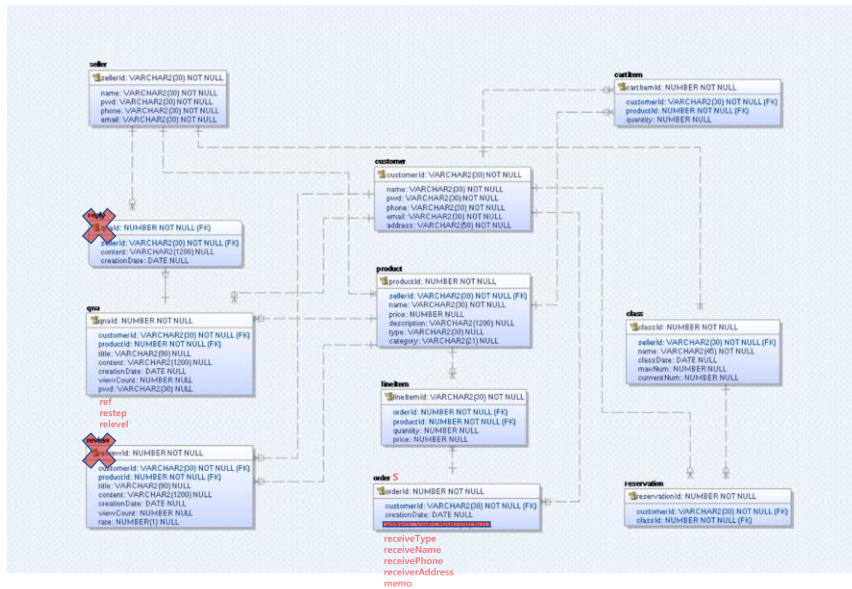
SECRET ☐ 공개글 ☒ 비밀글

WRITE CANCEL

구현 결과

19

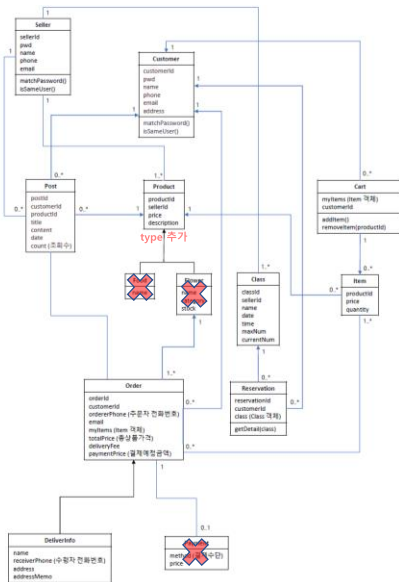
P04 데이터베이스 설계



20

P04

CLASS 설계 (Domain/Controller)

**UserController**

ViewUserController
LoginController
LogoutController
ListUserController
RegisterUserController
UpdateUserController
DeleteUserController
UserSessionUtils

ProductController

ViewProductController
ManageProductController
SearchProductController
ListProductController

ClassController

ListClassController
ManageClassController
ReservationController

CartController

ViewCartController
ManageCartController

OrderController

OrderController
ListOrderController

QnaController

ViewQnaController
ListQnaController
RegisterQnaController
UpdateQnaController
DeleteQnaController
UserSessionUtils

MyPageController

ManagerMypageController
ViewMypageController

21

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()

OrderManager

orderId, quantity,
productId

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

CartManager

cartItemId, customerId

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

ReservationManager

customerId, reservationId, classId

classReservation(Reservation)
findReservationList(customerId)

QnaManager

qnaId

createQna(Qna), updateQna(Qna)
removeQna(qnaId), findQna(qnaId), findQnaList()

22

P04

CLASS 설계 (DAO)

UserDAOcustomerId, pwd, name,
phone, email, addresscreateCustomer(Customer)
updateCustomer(Customer)
removeCustomer(customerId)
findCustomer(customerId)
findCustomerList()
existingCustomer(customerId)**ProductDAO**productId, sellerId, name,
price, description, type,
categoryadd(Product)
update(Product)
remove(productId)
findProduct(productId)
findProductList(type)
findProductByName(name)**ClassDAO**classId, sellerId,
name, date,
maxNum,
currentNumadd(ClassInfo)
update(ClassInfo)
delete(classId)
findClassList()
findClass(classId)**OrderDAO**orderId, customerId,
creationDate, receiveType,
receiverName, receiverPhone,
memo, receiverAddress, price,
lineItemId, productId, quantityaddOrder(Orders, quantity,
productIds)
findOrder(orderId)
findOrderList()
findOrderList(customerId)**CartDAO**cartItemId, quantity,
customerId, productIdaddItem(CartItem), removeItem(cartItemId),
getCartItem(cartItemId), getCartItemList(customerId)**ReservationDAO**

customerId, reservationId, classId

classReservation(Reservation)
findReservationList(customerId)**QnaDAO**qnaId, title, content, creationDate, viewCount,
pwd, customerId, productId, ref, restep, relevelcreateQna(Qna), updateQna(Qna),
removeQna(qnaId), findQna(qnaId), findQnaList()

23

P04

Request 처리 흐름

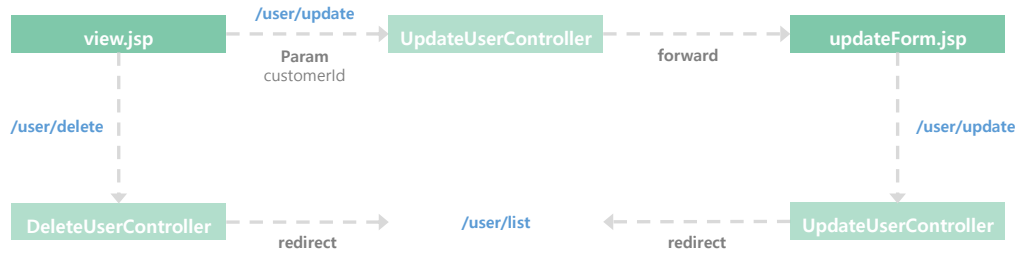
로그인, 회원가입



24

P04 Request 처리 흐름

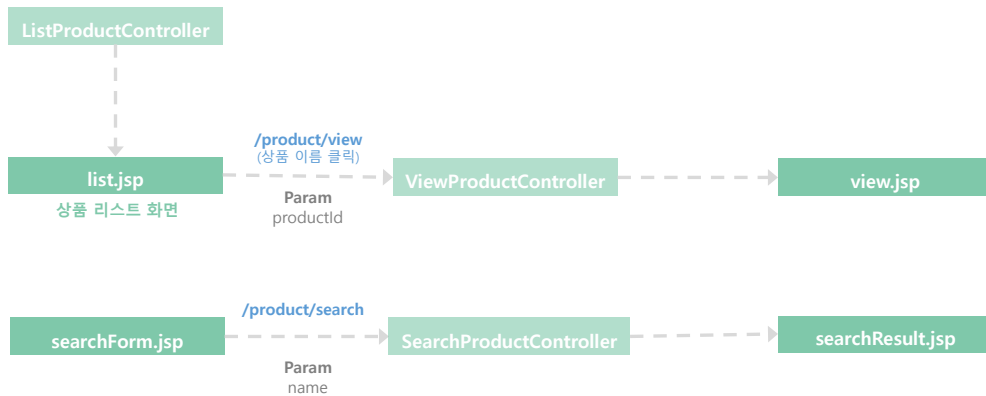
회원 정보 수정/탈퇴



25

P04 Request 처리 흐름 (수정)

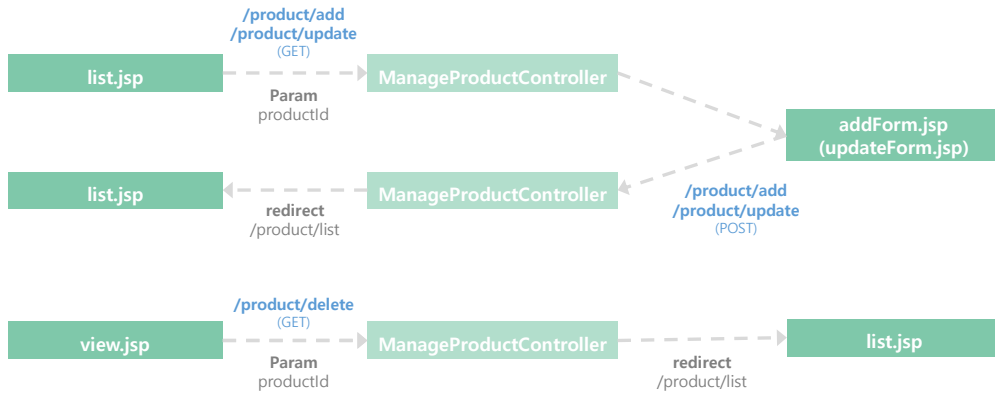
상품 상세조회/검색



26

P04 Request 처리 흐름 (수정)

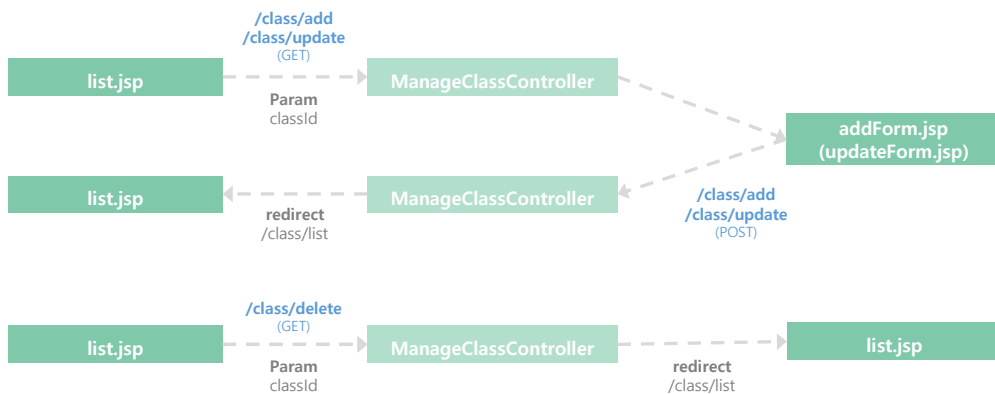
상품 추가/수정/삭제



27

P04 Request 처리 흐름 (수정)

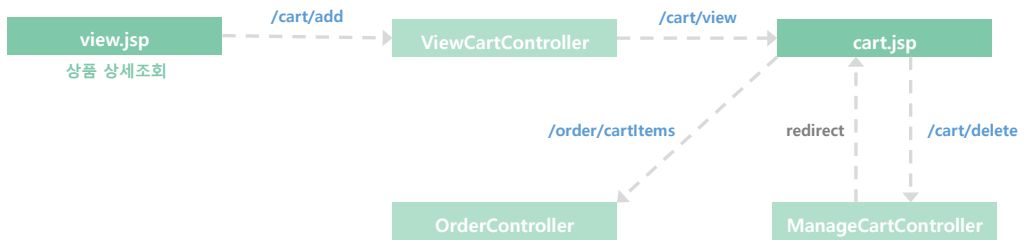
클래스 추가/수정/삭제



28

P04 Request 처리 흐름 (수정)

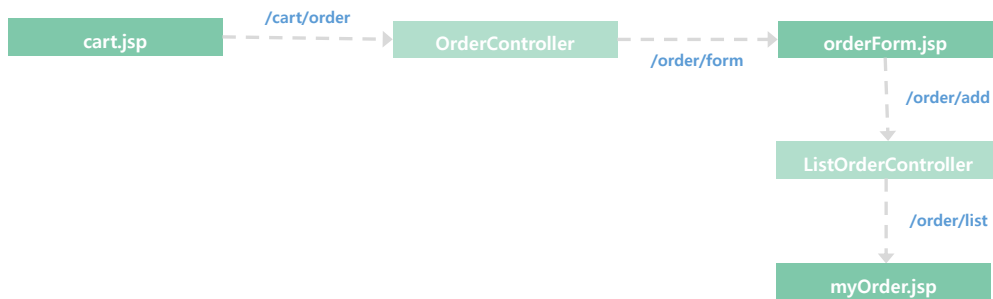
장바구니 상세조회, 추가/삭제



29

P04 Request 처리 흐름 (수정)

상품 주문, 결제



30

P04 Request 처리 흐름

문의 목록조회/작성



31

P05 Eclipse 프로젝트 구조



32

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, 클래스 예약

4

문의 사항 조회, 문의 사항 작성

5

각 회원의 상품구매 / 클래스예약 / 문의 내역조회

33

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 }

```

34

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     public 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.getPathInfo().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 (상품정보가 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.getPathInfo().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                 // 수정이 불가능할 경우 /product/list로 이동
69                 // IllegalStateException
70             }
71
72             // POST: 상품 수정 (상품정보가 parameter로 전송됨)
73             Product product = new Product(
74                 Integer.parseInt(request.getParameter("productId")),
75                 request.getParameter("name"),
76                 Integer.parseInt(request.getParameter("price")),
77                 request.getParameter("description"),
78                 request.getParameter("type"),
79                 request.getParameter("category"),
80                 request.getParameter("sellerId"));
81
82             manager.update(product);
83             log.debug("Update Product: " + product.toString());
84             return "redirect:/product/list?type=" + request.getParameter("type");
85         }
86
87         // 삭제
88         else {
89             int deleteId = Integer.parseInt(request.getParameter("productId"));
90
91             // 해당 상품의 판매자인지 확인하는 코드 추가
92             manager.delete(deleteId);
93
94             return "redirect:/product/list?type=flower";
95         }
96     }
97 }
98
99 }

```

35

P06

업무 분담

이채윤

20201008

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

김호민

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 툴 안내
발표 자료 제작User 정보
로그인, 회원가입
회원 수정/탈퇴
회원 목록/상세조회Qna 정보
문의사항 작성
문의사항 목록/상세조회

이승연

20201001

수행일지 작성
자료준비 및 발표

메인 페이지

마이 페이지
주문/예약/문의 내역
회원정보수정Cart 정보
장바구니 추가/삭제

36

P05

결론 및 소감

이채운

20201008

배운 지식들을 모두 활용해야 해서 조금 어려웠지만 그 지식들을 검색하고 더 깊게 공부하면서 많은 것을 알게 되었습니다.
혼자 했다면 해매고 좌절도 많았을 텐데 팀원분들이 도와 주신 덕분에 프로젝트에 열심히 참여할 수 있었습니다.

직접 웹사이트를 만들었다는 게 무엇보다 뿌듯했습니다. 처음 해보는 프로젝트여서 그런지 해매는 것도 많았고 일정 관리도 힘들었지만 어떤 식으로 프로젝트를 진행해야 하는지 알 수 있었고, 이 경험이 바탕이 되어서 더 좋은 프로젝트를 진행할 수 있을 것 같습니다..

김호민

20200962

이시윤

20200998

웹 페이지 설계부터 구현까지 여러 단계를 경험하면서 협업 능력을 기를 수 있었습니다. DB와 웹을 연동하는 방법, 페이지 간의 데이터 처리 과정 등을 배울 수 있는 유익한 시간이었습니다.

프로젝트 기획, 설계, 개발까지 모두 경험하면서 웹으로 요청이 들어오면 어떤 방식으로 연결되어 동작하는지 배울 수 있었습니다. 처음이라 힘들고 아쉬운 점이 많았지만 팀원들과 같이 의미있는 결과물을 만들어내서 뿌듯하고 유익한 시간이었습니다.

이승연

20201001

37

감사합니다

38