

# MVC모델을 활용한 라면자판기

코알라조 == 코딩알려줘라  
발표자 : 김종현



## 조원소개



Model

황지민, 이준선



View

김수연, 이향준



Controller

김종현, 임환욱

# 목차

1. 개요
2. 개발환경
3. 스케줄 및 버전 관리
4. 요구사항 분석
5. ERD, UML
6. User Flow
7. Logic Process
8. 코드실행
9. 오류해결 및 개선사항
10. 소감



# 1. 개요 - 주제선정

## Ramen Machine



- ✓ 라면, 토핑, 매운 맛, 물의 양, 사이드 메뉴를 선택  
-> 본인의 취향에 맞는 라면을 만들 수 있는 라면 자판기
- ✓ 일반적인 자판기와는 다르게 여러 선택 기능을 구현  
하기 좋다고 판단하여 라면자판기로 선정!

## 설계목적

- ✓ 완성도가 높고 유지보수가 용이한 동시에  
사용자, 개발자 모두를 배려한 코드 작성 및 숙련도 향상

## 기대효과

- ✓ MVC의 정의 역할 별 코드 작성법 숙지
- ✓ 유지보수 중심적 코드 작성 숙지

## 기능요약

- ✓ 상품 구매, 상품 관리, 재고 관리

협업 도구



디스코드(Discord)



슬랙 (Slack)

사용 언어



이클립스(Eclipse)

자바(Java)  
JDK11



6/29

1.01 V

주제 선정  
팀명 선정  
기능 구상

7/2

1.03 V

사용자 모드  
관리자 모드 구분

7/4

2.04 V

가격 정보 기능 추가  
설계도 수정 및 작성 완료

7/7

2.06 V

UI 출력 부분 수정

7/10

3.07 V

총 주문서, 가격 출력  
기능 추가

7/1

1.02 V

역할 분담  
MVC 파트 별 설계 진행

7/3

1.04 V

관리자 모드  
재고 변경 기능 추가

7/6

2.05 V

변수명 수정

7/8

2.07 V

관리자 모드 각 메뉴에서  
재고 출력 기능 추가

7/11

3.08 V

각 품목 별 가격 출력  
기능 추가



라면, 토핑, 매운 맛 정도, 물의 양, 사이드 메뉴를 자신의 기호에 따라 주문

완성된 라면은 사용자 1명 당 1개 선택

## 사용자 기능

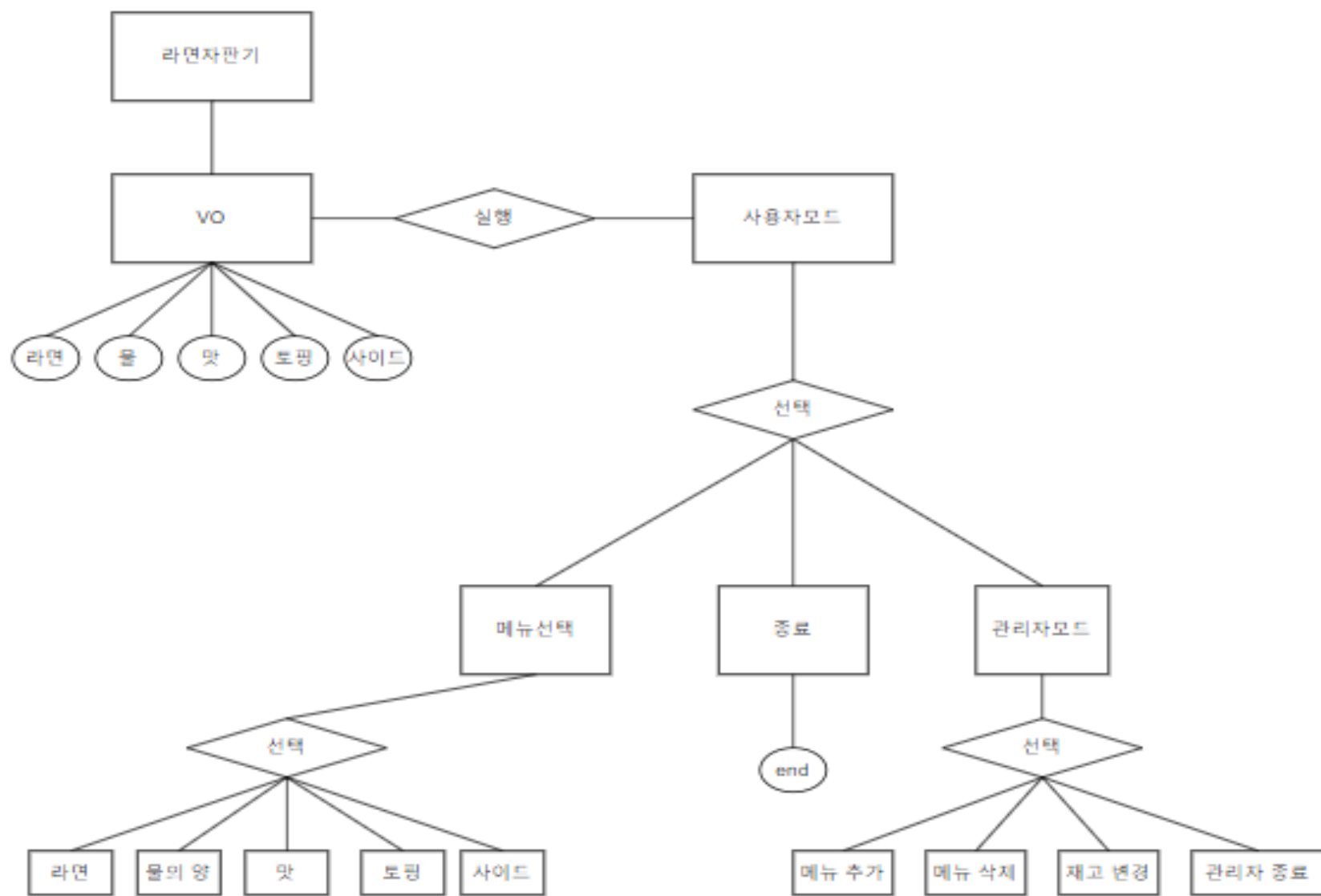
- ✓ 메뉴목록 출력
- ✓ 상품 선택
- ✓ 선택한상품 목록,가격출력
- ✓ 자판기종료문구출력

## 사용자 편의성

- ✓ 사용자기호에맞는주문가능
- ✓ 처음으로 돌아가기 문구 출력

## 관리자 기능

- ✓ 상품추가
- ✓ 상품삭제
- ✓ 재고량에 근거한 재고 변경
- ✓ 관리자모드종료



**ProductVO**

- num : int  
 - cnt : int  
 - price : int  
 - ramen : String  
 - topping : String  
 - side : String  
 - water : String  
 - spicy : String

+ getRamen() : String  
 + setRamen(String ramen) : void  
 + getTopping() : String  
 + setTopping(String topping) : void  
 + getSide() : String  
 + setSide(String side) : void  
 + getWater() : String  
 + setWater(String water) : void  
 + getSpicy() : String  
 + setSpicy(String spicy) : void  
 + getPrice() : String  
 + setPrice(int price) : void  
 + getNum() : int  
 + setNum(int num) : void  
 + getCnt() : int  
 + setCnt(int cnt) : void  
 + toString()

**ProductDAO**

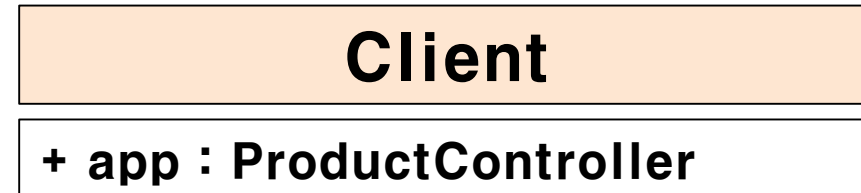
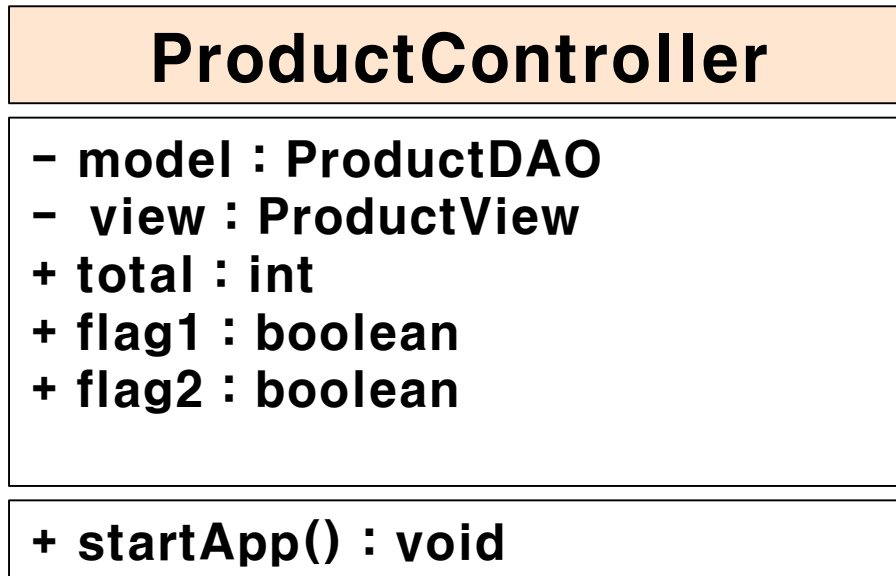
+ ramen : ArrayList<ProductVO>  
 + topping : ArrayList<ProductVO>  
 + side : ArrayList<ProductVO>  
 + water : ArrayList<ProductVO>  
 + spicy : ArrayList<ProductVO>  
 + price : ArrayList<ProductVO>  
 + result : ArrayList<ProductVO>  
 - ramenpk : int  
 - toppingpk : int  
 - sidepk : int  
 - waterpk : int  
 - spicypk : int

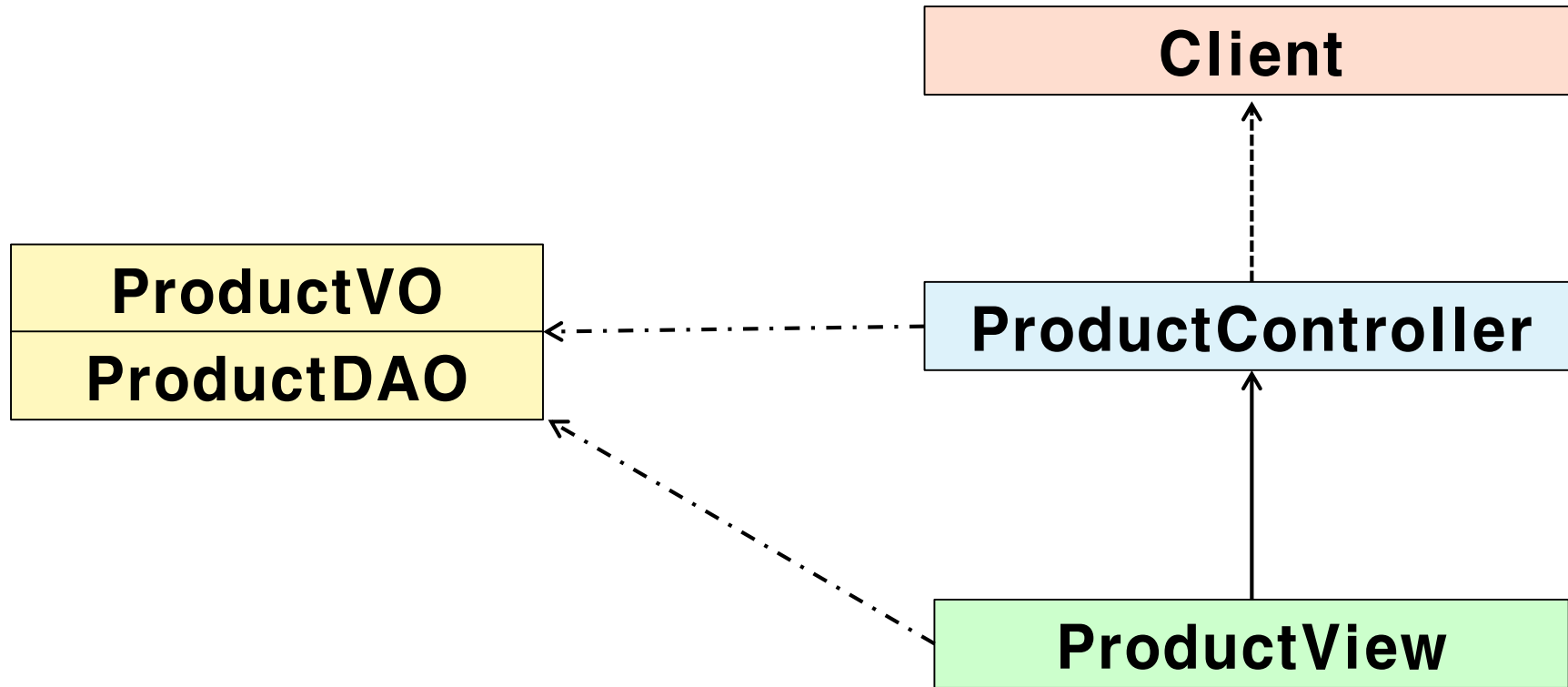
+ ramenInsert(ProductVO ramenvo) : boolean  
 + toppingInsert(ProductVO toppingvo) : boolean  
 + sideInsert(ProductVO sidevo) : boolean  
 + waterInsert(ProductVO watervo) : boolean  
 + spicyInsert(ProductVO spicyvo) : boolean  
 + ramenDelete(ProductVO ramenvo) : boolean  
 + toppingDelete(ProductVO toppingvo) : boolean  
 + sideDelete(ProductVO sidevo) : boolean  
 + waterDelete(ProductVO watervo) : boolean  
 + spicyDelete(ProductVO spicyvo) : boolean  
 + ramenUpdate(ProductVO ramenvo) : boolean  
 + sideUpdate(ProductVO sidevo) : boolean  
 + toppingUpdate(ProductVO toppingvo) : boolean  
 + ramenselectOne(ProductVO ramenvo) : ProductVO  
 + toppingselectOne(ProductVO toppingvo) : ProductVO  
 + sideselectOne(ProductVO sidevo) : ProductVO  
 + waterselectOne(ProductVO watervo) : ProductVO  
 + spicysselectOne(ProductVO spicyvo) : ProductVO  
 + ramenselectAll(ProductVO ramenvo) : ArrayList<ProductVO>  
 + toppingselectAll(ProductVO toppingvo) : ArrayList<ProductVO>  
 + sideselectAll(ProductVO sidevo) : ArrayList<ProductVO>  
 + waterselectAll(ProductVO watervo) : ArrayList<ProductVO>  
 + spicysselectAll(ProductVO spicyvo) : ArrayList<ProductVO>

**ProductView**

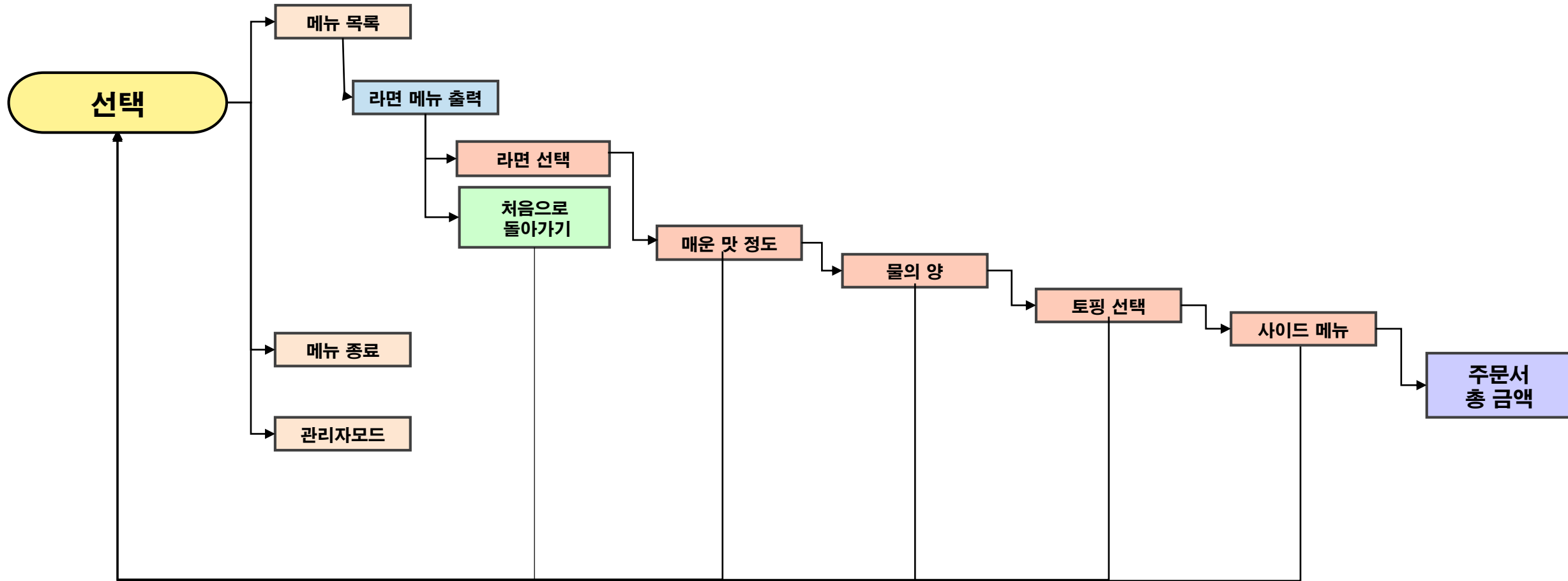
+ action : void

+ startView() : void  
 + inputString() : String  
 + inputInt() : int  
 + inputInt\_cnt() : int  
 + inputInt\_Price() : int  
 + input(ArrayList<ProductVO> vo) : int  
 + user\_RamenMenu(ArrayList<ProductVO> data) : void  
 + user\_ToppingMenu(ArrayList<ProductVO> data) : void  
 + user\_SiderMenu(ArrayList<ProductVO> data) : void  
 + user\_WaterMenu(ArrayList<ProductVO> data) : void  
 + user\_SpicyMenu(ArrayList<ProductVO> data) : void  
 + ramen() : void  
 + topping() : void  
 + side() : void  
 + water() : void  
 + spicy() : void  
 + end() : void  
 + back() : void  
 + printDatas(ProductVO vo) : void  
 + shoppingBasket() : void  
 + readyMenu() : void  
 + soldOut() : void  
 + adminView() : void  
 + admin\_addMenu() : void  
 + admin\_selectcheck(String check) : void  
 + admin\_deleteMenu() : void  
 + admin\_deletecheck(int check) : void  
 + admin\_updateCnt() : void  
 + admin\_RamenMenu(ArrayList<ProductVO> data) : void  
 + admin\_ToppingMenu(ArrayList<ProductVO> data) : void  
 + admin\_SideMenu(ArrayList<ProductVO> data) : void  
 + scoreQ() : String  
 + admin\_check() : String  
 + admin\_end() : void  
 + success() : void  
 + fail() : void

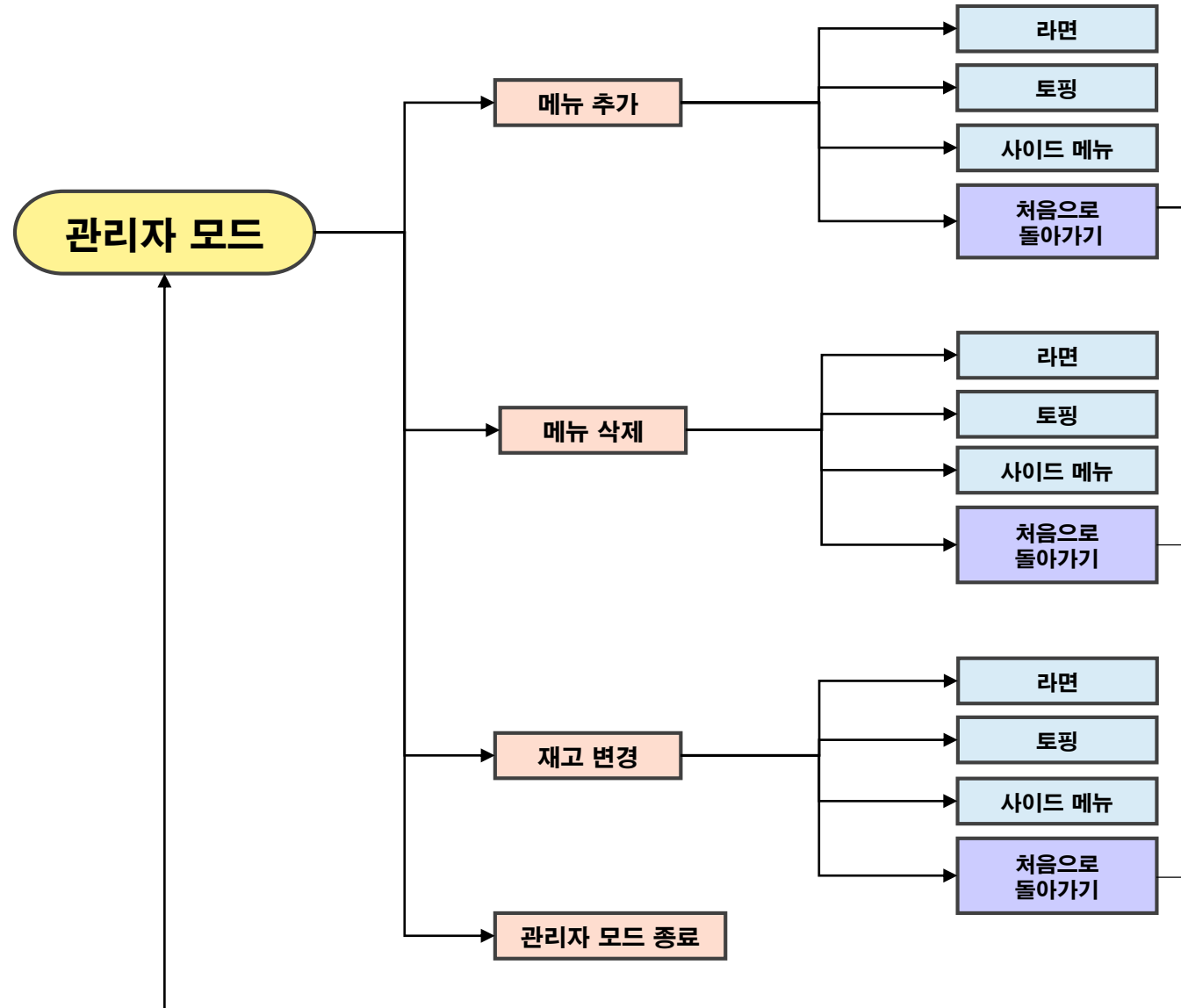




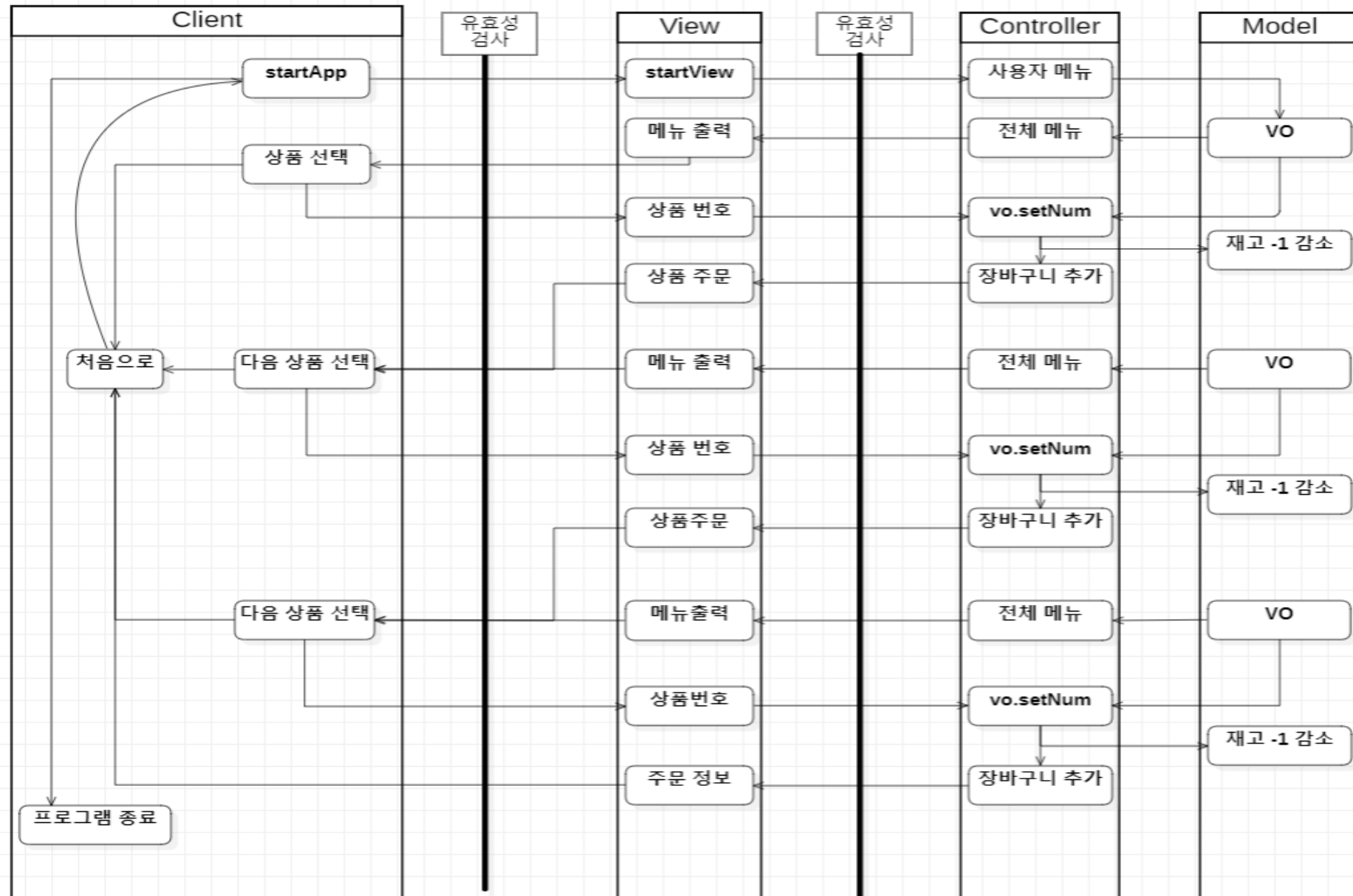
- 연관 관계
- - - - -> 실체화 관계
- . - . - .> 의존 관계



# USER FLOW (관리자 모드)

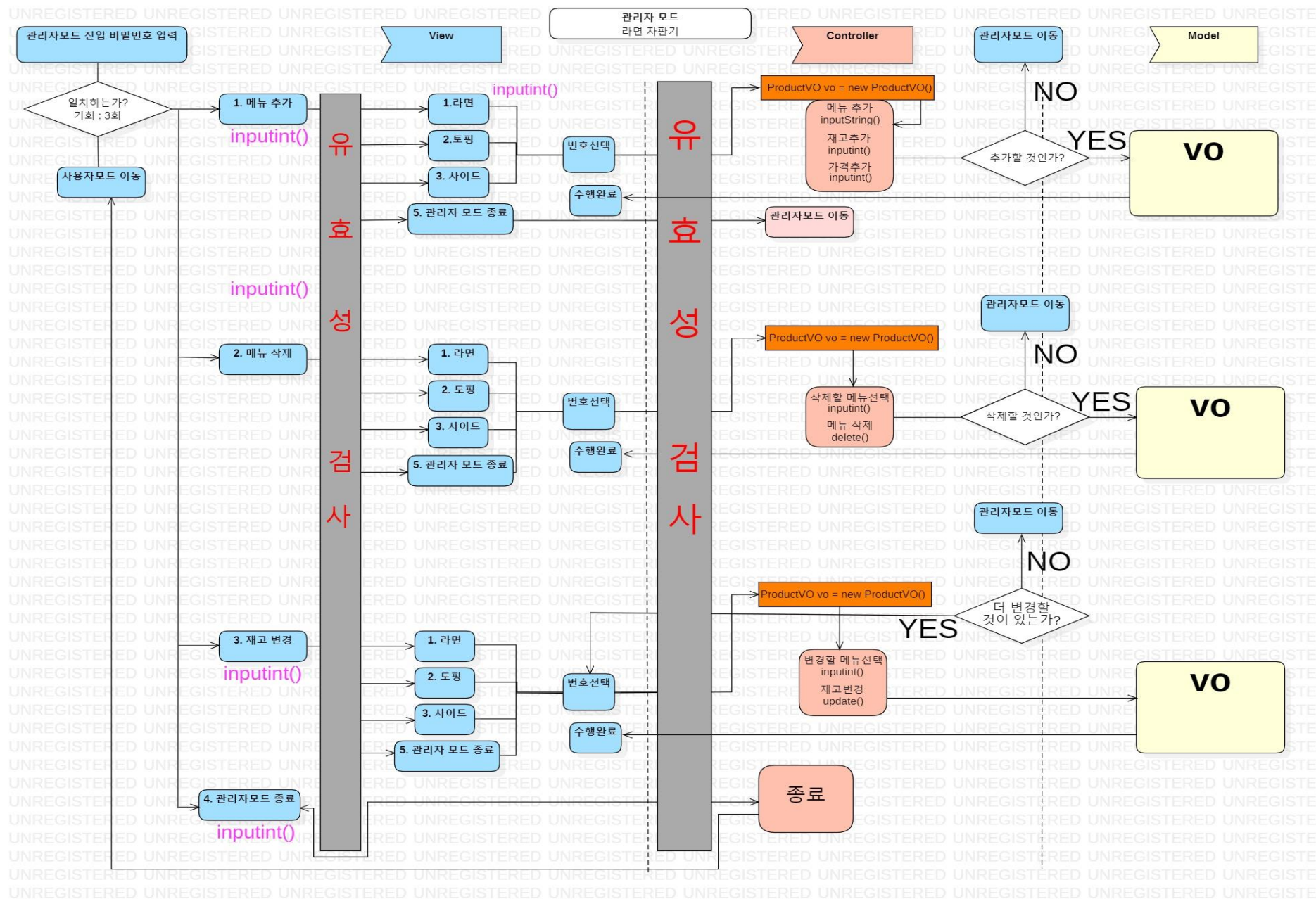


## 7. Logic Process (사용자모드)





**7.**



**코드 실행**

## 9. 오류해결

[M]

**문제점** : 메뉴 목록 첫 번째에 저장된 상품만 출력

**해결 방안** : 각각 다른 객체 명을 사용해 메뉴 출력 시 올바르게 출력

```
// 1
ProductVO ramenVo = new ProductVO();
ramenVo.setCnt(50);
ramenVo.setRamen("진라면");
ramenVo.setNum(ramenPk++);
ramenVo.setPrice(1500);
ramen.add(ramenVo);
```

```
// 2
ramenVo.setCnt(50);
ramenVo.setRamen("짜파게티");
ramenVo.setNum(ramenPk++);
ramenVo.setPrice(2000);
ramen.add(ramenVo);
```



```
// 1
ProductVO ramenVo1 = new ProductVO();
ramenVo1.setCnt(50);
ramenVo1.setRamen("진라면");
ramenVo1.setNum(ramenPk++);
ramenVo1.setPrice(1500);
ramen.add(ramenVo1);
```

```
// 2
ProductVO ramenVo2 = new ProductVO();
ramenVo2.setCnt(50);
ramenVo2.setRamen("짜파게티");
ramenVo2.setNum(ramenPk++);
ramenVo2.setPrice(2000);
ramen.add(ramenVo2);
```

## 9. 오류해결

[V]

**문제점** : inputInt 정수 입력 부분 입력값 오류 발생

**해결 방안** : 문자, 음수 입력 유효성 검사 추가

```
public int inputInt_Cnt() {    // 메뉴 추가할 때 필요한 재고입력창(음수가존재하면 안된다.)
    while(true) {
        System.out.print("재고입력 : ");
        try {
            int num = sc.nextInt();
            if(num > 0) {
                return num;
            }
            System.out.println("음수는 입력받을 수 없습니다.\n");
        }catch(Exception e) {
            sc.nextLine();
            System.out.println("다시 입력해주세요");
        }
    }
}
```

## 9. 오류해결

[C]

사용자 부분

PK번호 : 고유한 번호로 상품 추가/삭제에도 미 변경

**문제점** : 사용자에게 보이는 메뉴 번호가 PK번호이면 부자연스러움을 느낌

**해결 방안** : 메뉴 번호를 시스템 내부 PK로 전환하는 소스 코드 사용

```
// view.action : 사용자 입력 번호, allMenu : 전체 메뉴 목록  
vo.setNum(allMenu.get(view.action - 1).getNum());
```

- ✓ 메뉴 선택 시 바로 이전으로 돌아가기 기능 추가
- ✓ 자판기 금액 투입 후 계산 기능 추가
- ✓ 주문 가능 상품 개수 증가

**시행착오**

**책임감**

**협업의 중요성**

**MVC 패턴의 이해**

**유지보수 용이**





