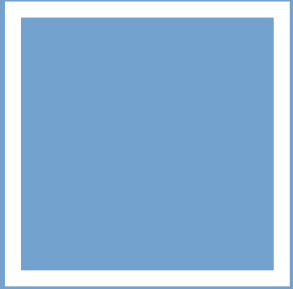


2주차 발표

6조

안윤진 진소희
진창용 이관훈 이소영



목차

A table of Contents

#1 개요

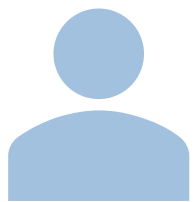
#2 요구사항 정의서

#3 업무 플로우

#4 사이트 개선사항

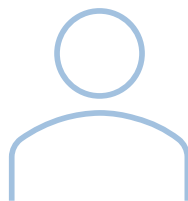
#5 기능 소스코드

1-1. 조원 소개 및 역할 분담



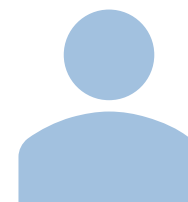
이소영

조장
상품 검색 기능
발표



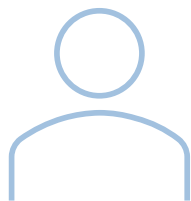
안윤진

로그인/회원가입 기능
PPT 제작



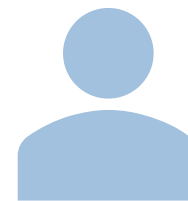
진소희

장바구니 기능



진창용

기획 상품 기능



이관훈

베스트 상품 기능
PPT 제작

1-2. 지난주 리뷰

역할 분담 시 과제 내용에
대해 조사만 하는 사람과,
ppt작성 혹은 발표만 부담하
며 개인 공부에 도움 되지
않을 수 있음

→ 각 기능마다 담당자를 정
해조원 모두가 과제 내용 전
체(분석~코드작성)에 대해
생각해볼 수 있도록 분담

개선 요구사항 작성시 현행
사이트 프로세스 개선이
아닌 추가가 필요한 기능적
요구사항들만 나열

→ 사이트의 프로세스 적인
개선과 업무 플로우의
변화를 제시

요구사항정의서 작성시
중복되는 부분이나
한 카테고리에 묶을 수 있는
부분들을 세부적으로
나누어 작성

→ 중복되거나 같은 카테고
리로 묶을 수 있었던 부분은
묶어서 읽기 편하고
알아보기 쉽도록 작성

요구사항 분석 시 현행
사이트의 요구사항과
프로세스 분석이 아닌
사이트 개선방안
서술에만 치중

→ 현행 사이트의 요구사항
분석과 사이트의 개선방안
모두 적절하게 분석

1-3. 일정 계획서

계 획 내 용	5월						
	22	23	24	25	26	27	28
- 지난주 리뷰 및 일정 계획 수립							
- 주제 사이트 선정, 역할분담							
- 샘플 작성							
- 샘플 피드백 및 수정							
- PPT 및 발표 초안 작성							
- 발표자료 수정 및 확정, PPT 제출							
- 최종 발표							

1-4. 주제 사이트

CJ THE MARKET

검색어를 입력하세요.

Q

바로가기 OFF | 로그인 | 고객센터

🚚

👤

🛒 0

☰ 전체 카테고리

마켓하달베스트기획전신제품선물하기가정간편식건강간편식즉킷산선식품건강식품

the프라임 | 기프트카드 | 쿠폰&혜택

가정간편식

건강간편식

즉킷

산선식품

반찬/김치

양념/오일

음료/간식

건강식품

대용량 식자재


장기배송

선물하기

대량구매권적

플레이북

브랜드




비비고

지금이 제철

비비고 김치 경품대전

선착순 사은품부터 김치냉장고까지



10 / 12 ||

추천!신제품

마켓하달



비비고의 신박한 면요리
비비고 소고기고추장 비빔유수면
비비고 소고기고추장비빔유수면636g
13% 6,480원 7,400원



감자에 치즈에 베이컨까지!
고메 클래식 포테이도 피자 400gX2개
23% 10,749원 13,960원
쿠폰할인 · 냉장



탱글한 새우로 꽉~채운 속!
비비고 수제새우만두325g*2x2개
25% 12,720원 16,960원
★★★★★ (18)



비비고쌔은배추김치900gx2개
30% 14,700원 21,000원
★★★★★ (1)
쿠폰할인 · 냉장

🛒 0

구매하기

최근본

TOP

CJ 더 마켓
(쇼핑몰)

2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
SCH-0001	상단 검색창	상단 검색창	키워드 입력	상	이소영	2021_5_24
SCH-0002		인기 검색어	인기 검색어 1~10위 표시	상	이소영	2021_5_24
SCH-0003		최근 검색어	이용자의 최근 검색어 기록 표시	상	이소영	2021_5_24
SCH-0004		자동저장 설정	최근 검색어 저장 기능 On/Off 설정	상	이소영	2021_5_24
SCH-0005		최근 검색어 삭제	기록 개별 삭제, 전체 삭제	상	이소영	2021_5_24
SCH-0006		자동완성 - 검색어	많이 검색되는 내용으로 검색어 자동 완성 제시	상	이소영	2021_5_24
SCH-0007		자동완성 - 기획전	입력한 내용을 포함하는 기획전 제시	상	이소영	2021_5_24
SCH-0008		추천 상품	자동완성 검색어 1순위의 상품 정보와 링크 제시.	중	이소영	2021_5_24
SCH-0009	상세 검색	연관검색어	연관검색어 제시	중	이소영	2021_5_24
SCH-0010		결과 내 재검색	검색어 입력	상	이소영	2021_5_24
SCH-0011		검색어 제외	검색어 입력	상	이소영	2021_5_24
SCH-0012		검색 조건 설정	카테고리, 혜택, 온도/배송유형, 품질여부, 브랜드, 상품구성, 별점, 가격	상	이소영	2021_5_24
SCH-0013		검색 조건 초기화	조건 설정 해제	상	이소영	2021_5_24
SCH-0014	검색 결과	탭 - 상품	검색된 상품을 추천 순(기본 정렬)로 출력	상	이소영	2021_5_24
SCH-0015		탭- 상품: 정렬	판매량순, 상품평점순, 높은 할인순, 신제품순, 낮은 가격순, 높은 가격순	상	이소영	2021_5_24
SCH-0016		탭- 상품: 표시 설정	한 페이지에 표시할 상품의 개수 설정 (기본80개)	상	이소영	2021_5_24
SCH-0017		탭- 상품: 간편 장바구니	상세 페이지로 이동하지 않고 장바구니 추가	상	이소영	2021_5_24
SCH-0018		탭- 기획전	키워드가 포함된 기획전 데이터 표시	상	이소영	2021_5_24
SCH-0019		탭- 동영상/레시피	키워드가 포함된 동영상/레시피 데이터 표시	중	이소영	2021_5_24

2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
LO-01	로그인	회원 로그인	아이디,비밀번호 입력하여 일치시 로그인	상	안윤진	2021_5_24
LO-02		아이디 저장	체크시 로그인시 입력한 아이디가 저장됨	하	안윤진	2021_5_24
LO-03		아이디 찾기	본인인증 또는 회원가입시 입력했던 회원정보를 입력 시 일치하는 아이디의 일부를 제공	상	안윤진	2021_5_24
LO-04		비밀번호 찾기	본인인증을 하여 새 비밀번호를 설정하거나 회원가입 시 입력했던 회원정보를 입력하여 일치 시 임시 비밀번호 발급	상	안윤진	2021_5_24
LO-05		SNS 로그인	카카오톡,네이버,페이스북 회원정보를 이용하여 로그인	하	안윤진	2021_5_24
LO-06		비회원 배송조회	로그인하지 않고 주문번호와 비밀번호를 입력하여 배송내역을 조회	상	안윤진	2021_5_24
JO-01	회원가입	가입여부 확인	이름,생년월일,휴대전화번호를 이용하여 기존에 가입되어 있는지 여부를 확인	중	안윤진	2021_5_24
JO-02		휴대전화 인증	입력한 휴대전화번호를 이용하여 본인 인증	중	안윤진	2021_5_24
JO-03		공인인증서 인증	공인인증서를 이용하여 본인 인증	중	안윤진	2021_5_24
JO-04		체크/신용카드 인증	체크/신용카드를 이용하여 본인 인증	중	안윤진	2021_5_24
JO-05		회원정보 입력	회원가입을 위한 이름,아이디,비밀번호,생년월일,휴대전화,이메일의 회원정보를 입력	상	안윤진	2021_5_24
JO-06		아이디 중복 확인	입력한 아이디가 기존에 사용되고 있는 아이디와 중복되지 않는지 확인	상	안윤진	2021_5_24
JO-07		비밀번호 조건 확인	입력한 비밀번호가 영문자,숫자,특수문자를 조합하여 8~12자리 아이디와 4자리 이상 동일, 반복 문자 숫자 불가능의 조건을 만족하는지 확인	상	안윤진	2021_5_24

2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
BS-01	베스트 > 판매 베스트	카테고리별 베스트 > 전체	전체 식품종류를 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등을 표시해준다. -제품 이미지 클릭 시 제품 구매페이지로 이동한다. 전체 식품종류 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스 필요	상	이관훈	2021_5_24
BS-02		카테고리별 베스트 > 가정간편식	가정간편식 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등을 표시해준다. -제품 이미지 클릭 시 제품 구매페이지로 이동한다. 가정간편식 카테고리 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스 필요	중	이관훈	2021_5_24
BS-03		카테고리별 베스트 > 건강간편식	건강간편식 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등을 표시해준다. -제품 이미지 클릭시 제품 구매페이지로 이동한다. 건강간편식 카테고리 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스 필요	중	이관훈	2021_5_24
BS-04		카테고리별 베스트 > 쿡킷	쿡킷 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등을 표시해준다. -제품 이미지 클릭시 제품 구매페이지로 이동한다. 쿡킷 카테고리 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스 필요	중	이관훈	2021_5_24

2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
BS-05	베스트 > 판매 베스트	카테고리별 베스트 > 신선식품	신선식품 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송 정보 등을 표시해준다. -제품 이미지 클릭 시 제품 구매페이지로 이동한다. 신선식품 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스필요	상	이관훈	2021_5_24
BS-06		카테고리별 베스트 > 반찬/김치	반찬/김치 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송 정보 등을 표시해준다. -제품 이미지 클릭 시 제품 구매페이지로 이동한다. 반찬/김치 카테고리 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-07		카테고리별 베스트 > 양념/오일	양념/오일 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송 정보 등을 표시해준다. -제품 이미지 클릭 시 제품 구매페이지로 이동한다. 양념/오일 카테고리 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-08		카테고리별 베스트 > 음료/간식	음료/간식 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송 정보 등을 표시해준다. -제품 이미지 클릭 시 제품 구매페이지로 이동한다. 음료/간식 카테고리 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스필요	중	이관훈	2021_5_24

2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
BS-09	베스트 > 판매 베스트	카테고리별 베스트 > 건강식품	건강식품 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송 정보 등을 표시해준다. -제품 이미지 클릭 시 제품 구매페이지로 이동한다. 건강식품 카테고리 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스필요	상	이관훈	2021_5_24
BS-10		카테고리별 베스트 > 대용량 식자재	대용량 식자재 카테고리의 식품을 베스트 순위별로 나열하여 보여준다 -상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송 정보 등을 표시해준다. -제품 이미지 클릭 시 제품 구매페이지로 이동한다. 대용량 식자재 카테고리 베스트 순위 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-11	베스트 > 생생 리뷰	생생 리뷰 > 신제품 체험단 리뷰	리뷰와 함께 업로드한 사진표시, 사진하단에 리뷰제목과 리뷰내용을 3줄 표시 이후는 ...으로 생략 처리 업로드 날짜 표시 하단에는 리뷰한 제품의 사진과 제품명, 별점, 가격 표시 -리뷰내용(리뷰글, 사진 등)을 클릭시 리뷰를 자세하게 볼 수 있는 팝업창생성 -리뷰 하단에 제품 정보를 클릭시 해당제품 구매페이지로 이동한다. 해당 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 가격, 별점, 신제품정보, 할인정보, 보관방법(냉동, 실온&상온), 배송정보 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-12		베스트 리뷰 > 전체	전체 식품종류를 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭시 제품 구매/상품평페이지로 이동한다. 전체 식품종류 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	중	이관훈	2021_5_24

2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
BS-13	베스트 > 생생 리뷰	베스트 리뷰 > 가정간편식	가정간편식 카테고리의 식품을 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 가정간편식 카테고리 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	상	이관훈	2021_5_24
BS-14		베스트 리뷰 > 건강간편식	건강간편식 카테고리의 식품을 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 건강간편식 카테고리 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-15		베스트 리뷰 > 쿡킷	쿡킷 카테고리의 식품을 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 쿡킷 카테고리 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-16		베스트 리뷰 > 신선식품	신선식품 카테고리의 식품을 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 신선식품 카테고리 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	중	이관훈	2021_5_24

2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
BS-17	베스트 > 생생 리뷰	베스트 리뷰 > 반찬/김치	반찬/김치 카테고리의 식품을 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 반찬/김치 카테고리 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	상	이관훈	2021_5_24
BS-18		베스트 리뷰 > 양념/오일	양념/오일 카테고리의 식품을 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 양념/오일 카테고리 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-19		베스트 리뷰 > 음료/간식	전체 식품종류를 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 전체 식품종류 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-20		베스트 리뷰 > 건강식품	건강간편식 카테고리의 식품을 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 건강간편식 카테고리 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	중	이관훈	2021_5_24
BS-21		베스트 리뷰 > 대용량 식자재	대용량 식자재 카테고리의 식품을 리뷰 베스트 순위별로 나열하여 보여준다 -상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수를 표시해준다. -별점 표시 하단에 날짜의 내림차수 순으로 3개의 리뷰를 보여준다 -제품 이미지 클릭 시 제품 구매/상품평페이지로 이동한다. 대용량 식자재 카테고리 리뷰 베스트 순위 데이터베이스 필요, 제품 리뷰 데이터베이스 필요, 상품상세정보(상품 이미지와 별점, 추천한 사람 수, 맛있어요를 남긴 사람 수 등) 데이터베이스필요	중	이관훈	2021_5_24

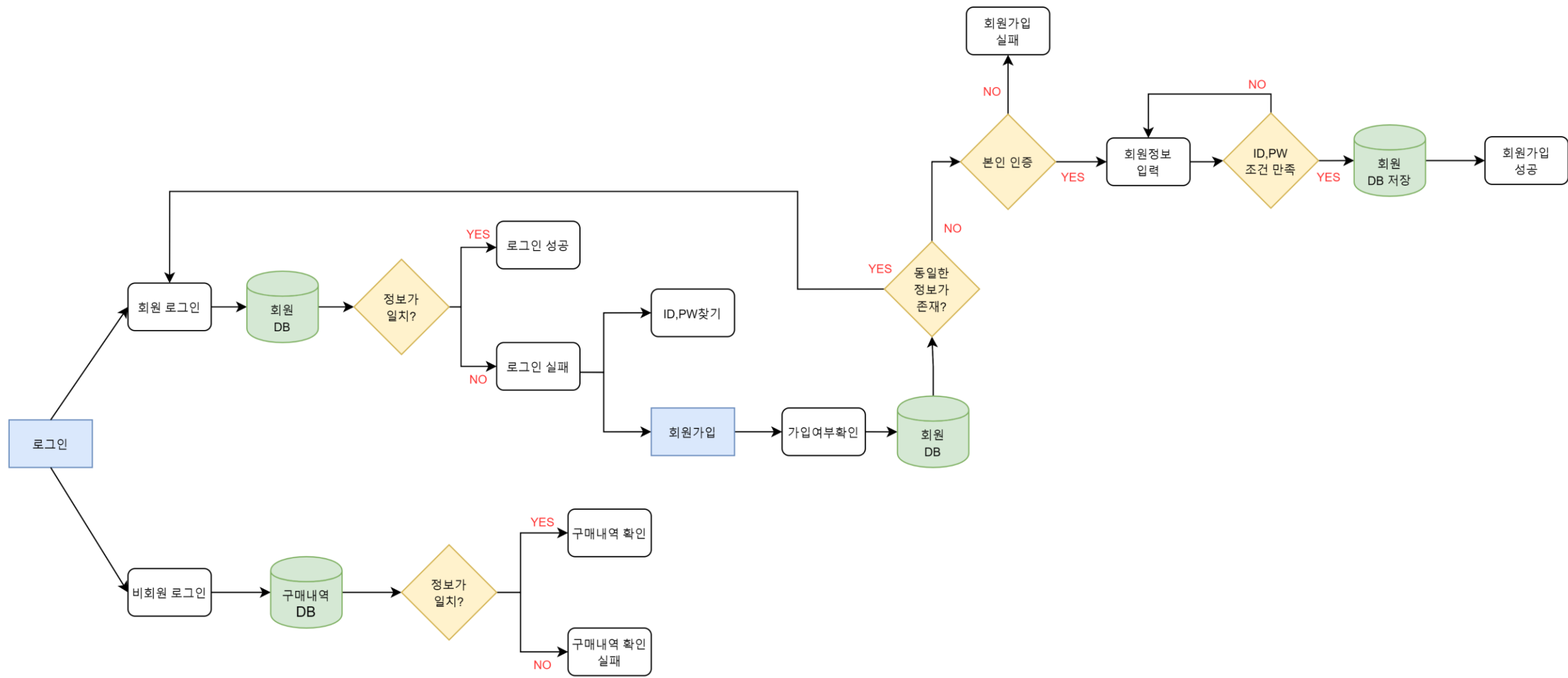
2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
MA-01	메인, 상품상세	상품 수량 결정	상품 설명 및 나열 화면으로부터 구매할 상품의 종류와 수량을 넘겨받음	상	진소희	2021_5_24
MA-02		바로구매	상품화면에서 바로구매 버튼 클릭시 장바구니 화면 경유 없이 바로 상품 구매 페이지로 이동	중	진소희	2021_5_24
JA-01	장바구니	상품 수량 조정	장바구니 내에서 구매할 상품의 수량을 조정	중	진소희	2021_5_24
JA-02		구매 대상 상품 선택	장바구니에 담긴 상품 가운데 구매 대상 상품을 선택	상	진소희	2021_5_24
JA-03		상품 삭제	장바구니에 넣은 상품을 선택하여 전체 및 일부를 삭제	중	진소희	2021_5_24
JA-04		상품 총계 표시	선택된 상품의 총계를 표시 (default: 전체선택)	상	진소희	2021_5_24
JA-05		구매불가상품 표시	재고가 없는 상품 표시 및 선택 가능 대상 제외	중	진소희	2021_5_24
JA-06		가격변동 상품 표시	가격변동이 일어난 상품은 기존 가격을 취소선으로 표시하고 그 위에 새로 변경된 가격을 진한 글씨로 표기	중	진소희	2021_5_24
JA-07		배송비 할인	선택된 상품 가격의 총계가 정책상의 지정금액 (현행: 3만원)을 초과할 경우 배송비를 0원으로 표기	중	진소희	2021_5_24
JA-08		포인트 혜택	적립되는 포인트의 총량 표기	중	진소희	2021_5_24
JA-09		배송방식 선택	일반배송/정기배송/선물세트 가운데 원하는 배송방식 선택	상	진소희	2021_5_24
JA-10		배송 정책 표시	장바구니 화면의 하단부에 배송 관련 정책을 표시	중	진소희	2021_5_24
OD-01	상품주문	배송정보 입력	상품을 배송할 주소 입력	상	진소희	2021_5_24
OD-02		회원 데이터 호출	기존 구매이력이 있는 회원의 경우 과거 배송주소/수령인명을 default값으로 지정	상	진소희	2021_5_24
OD-03		할인혜택 적용	할인혜택 선택	중	진소희	2021_5_24
OD-04		결제방식 선정	결제수단 선택 및 외부 결제 API 연결	상	진소희	2021_5_24

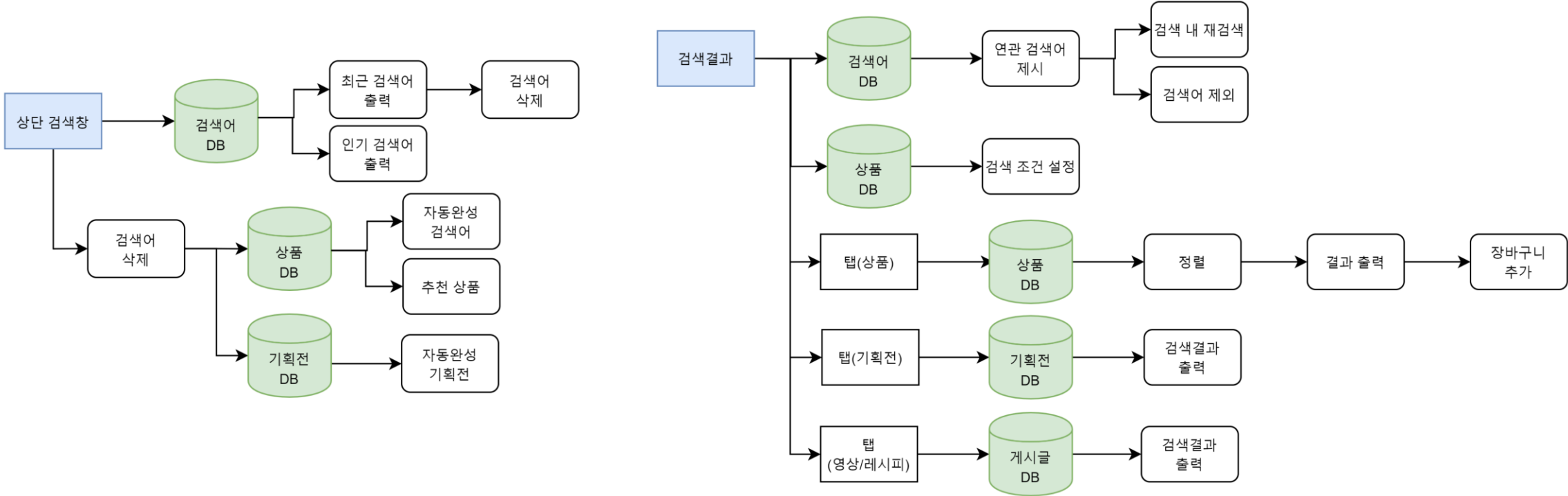
2. 요구사항 정의서

ID	화면명	요구사항명	요구사항 내용	중요도	부서/작성자	날짜
HD-01	마켓핫딜	타임카운트	상품 설명 및 나열 화면으로부터 구매할 상품의 종류와 수량을 넘겨받음	상	진창용	2021_5_24
HD-02		핫딜 상품추가	상품화면에서 바로구매 버튼 클릭시 장바구니 화면 경유 없이 바로 상품 구매 페이지로 이동	중	진창용	2021_5_24
GI-01	기획전	기획전 만들기	장바구니 내에서 구매할 상품의 수량을 조정	중	진창용	2021_5_24
GI-02		기획전 상품 등록	장바구니에 담긴 상품 가운데 구매 대상 상품을 선택	중	진창용	2021_5_24
GI-03		Sns 공유	장바구니에 넣은 상품을 선택하여 전체 및 일부를 삭제	하	진창용	2021_5_24

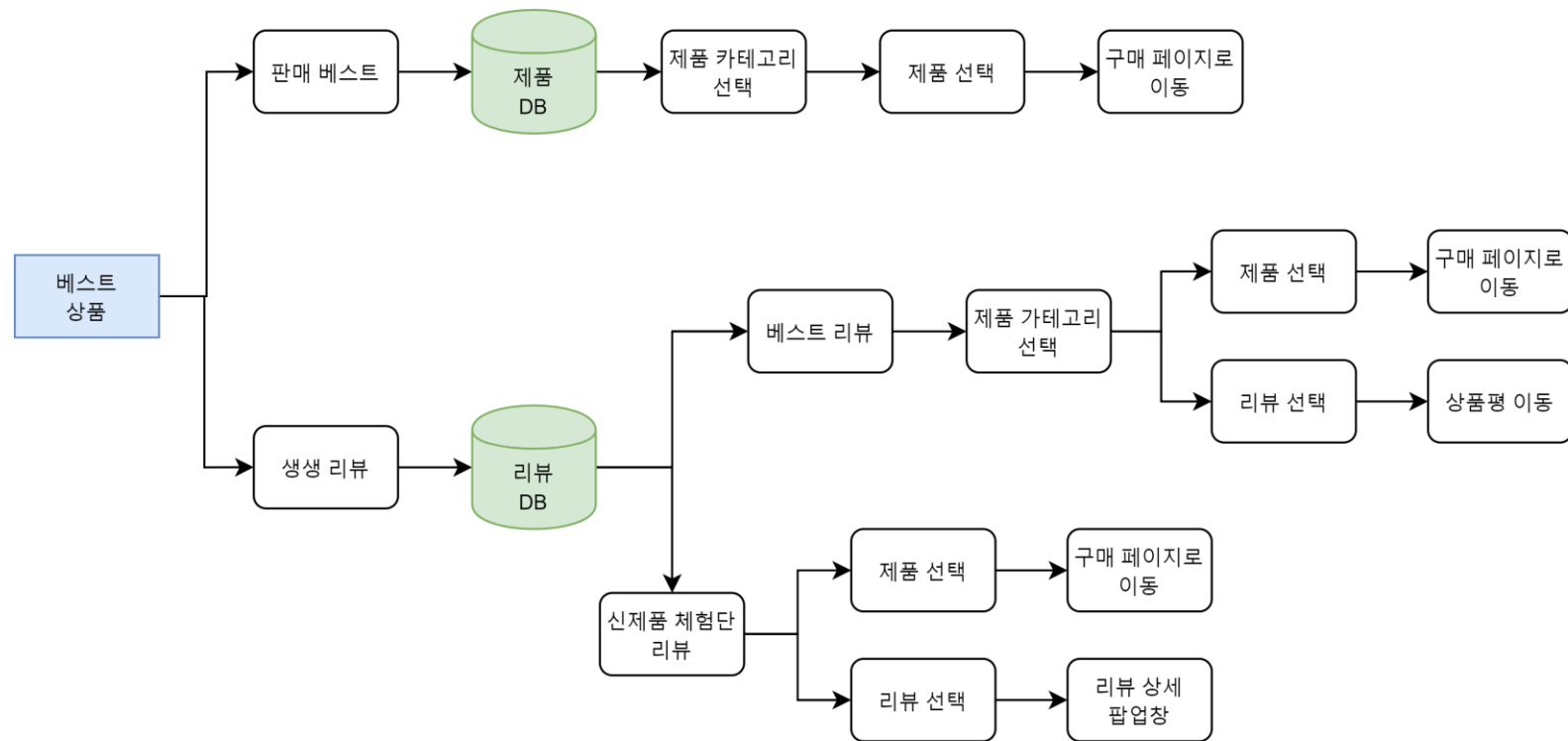
3. 업무 플로우 (로그인/회원가입)



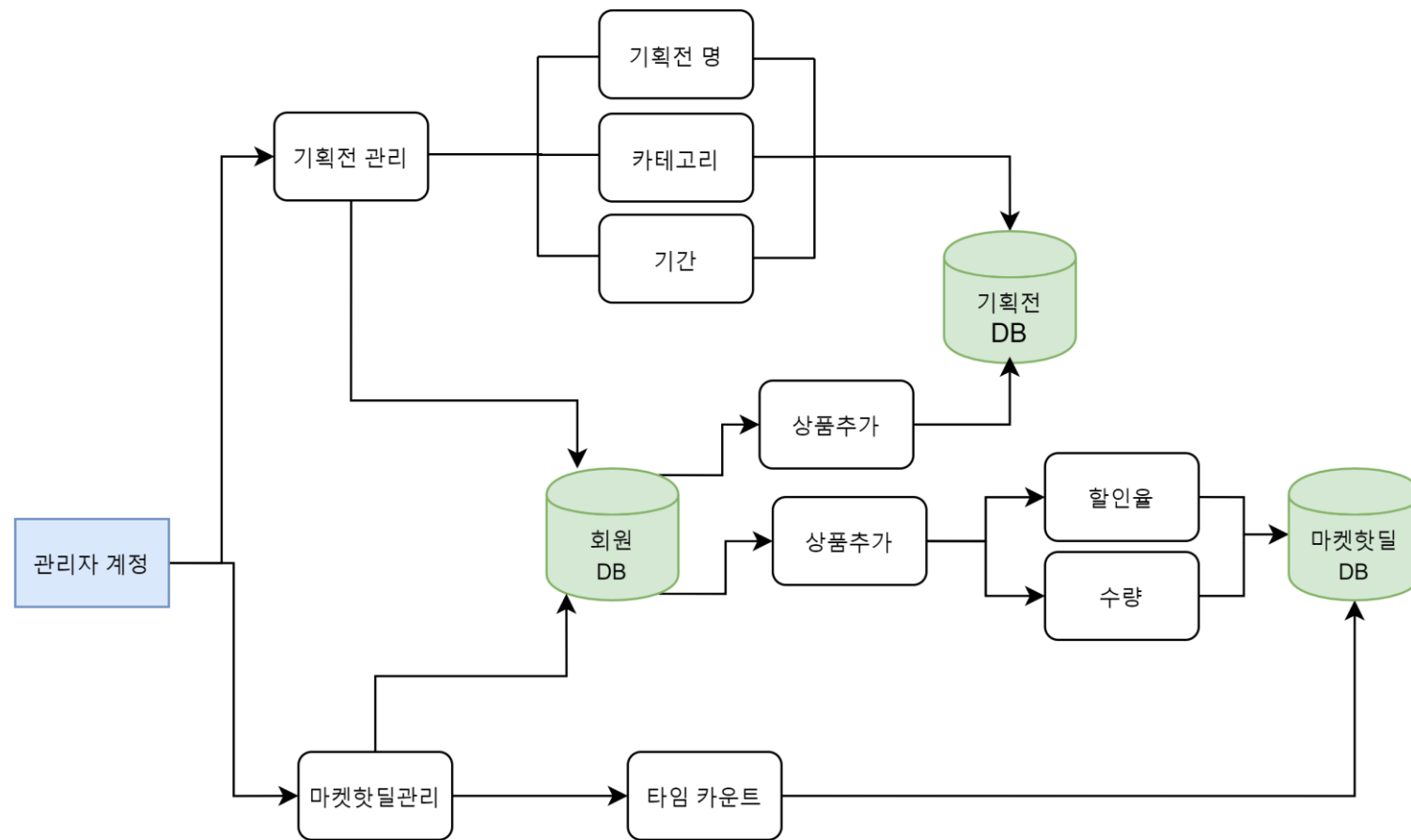
3. 업무 플로우 (상품 검색)



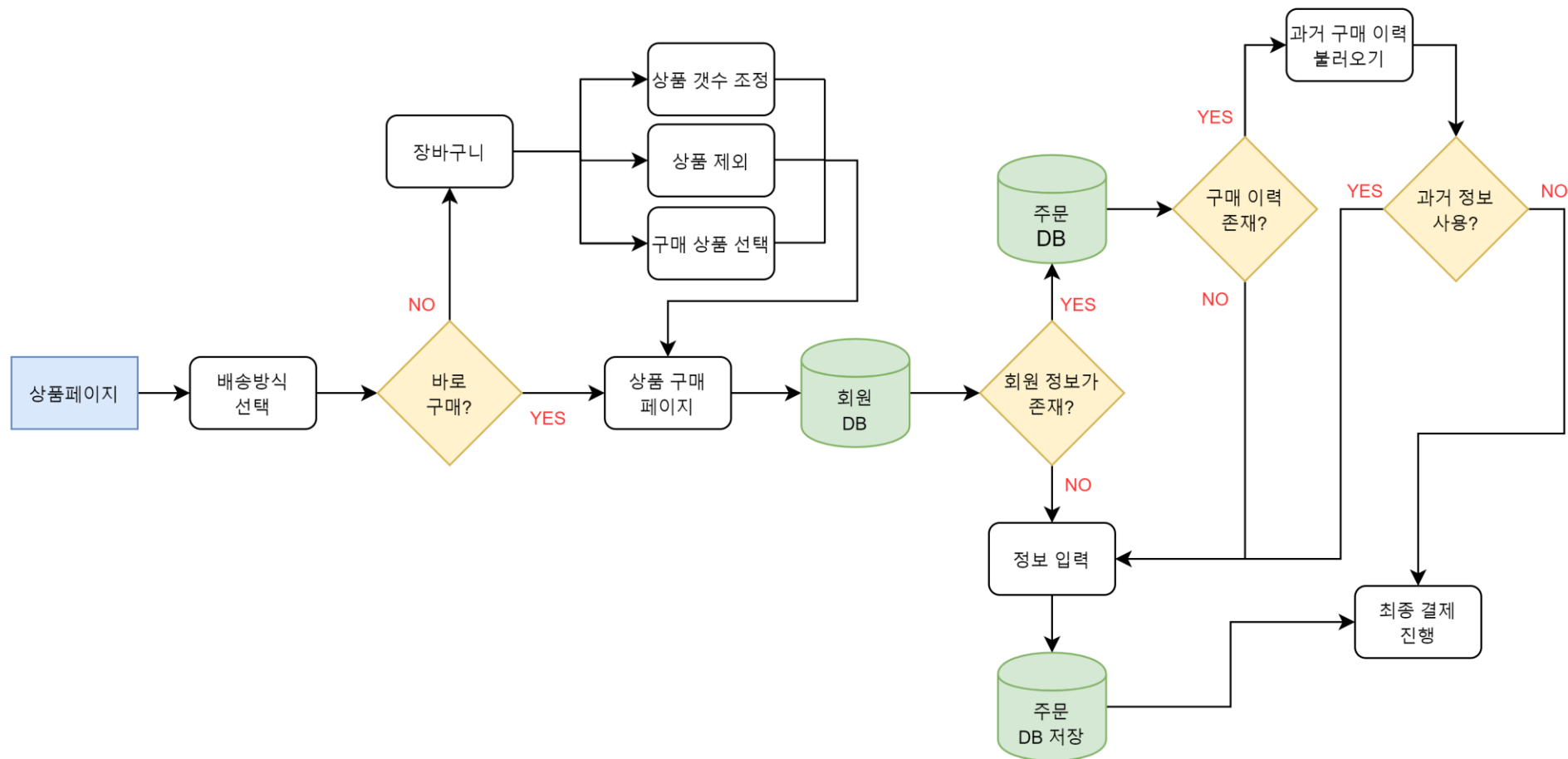
3. 업무 플로우 (베스트 상품)



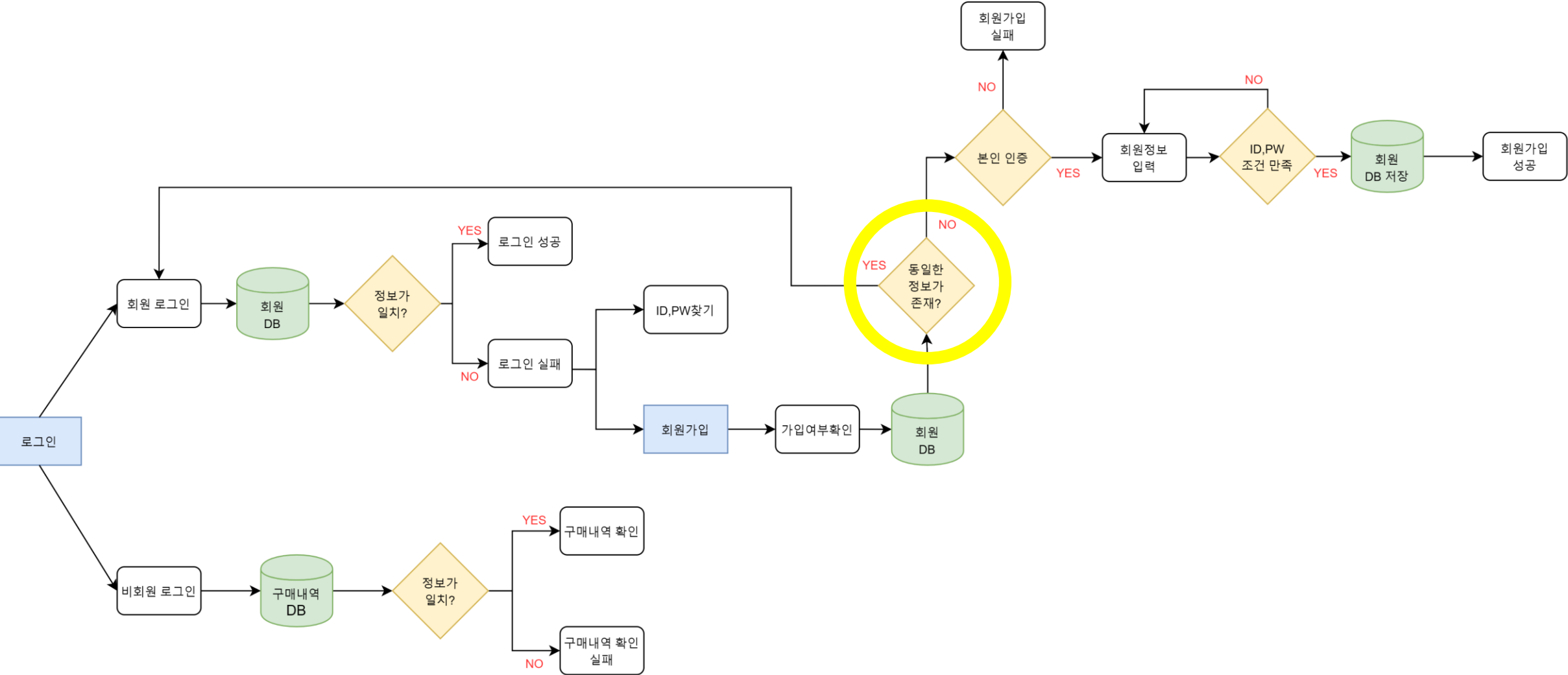
3. 업무 플로우 (기획전/마켓 핫딜 관리)



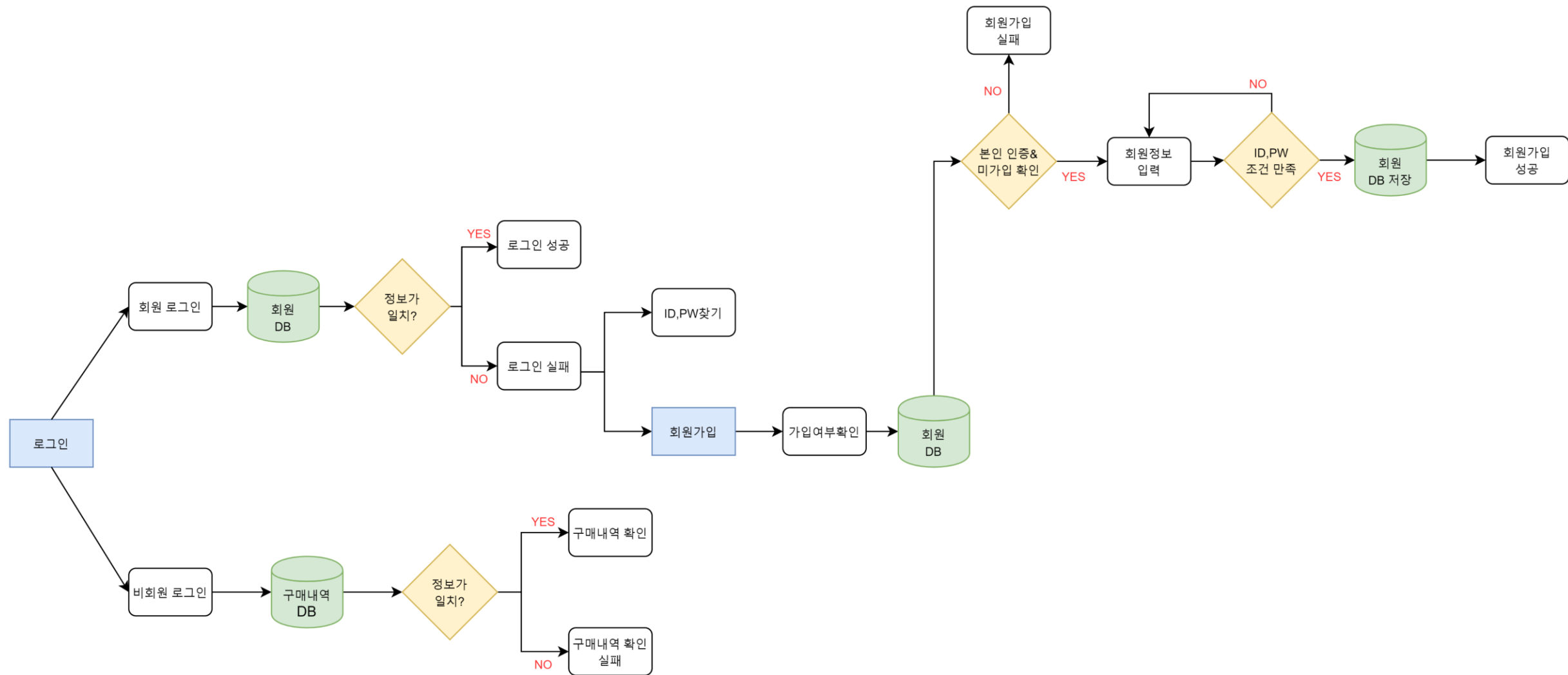
3. 업무 플로우 (장바구니)



4. 개선 사항 (기존 프로세스)

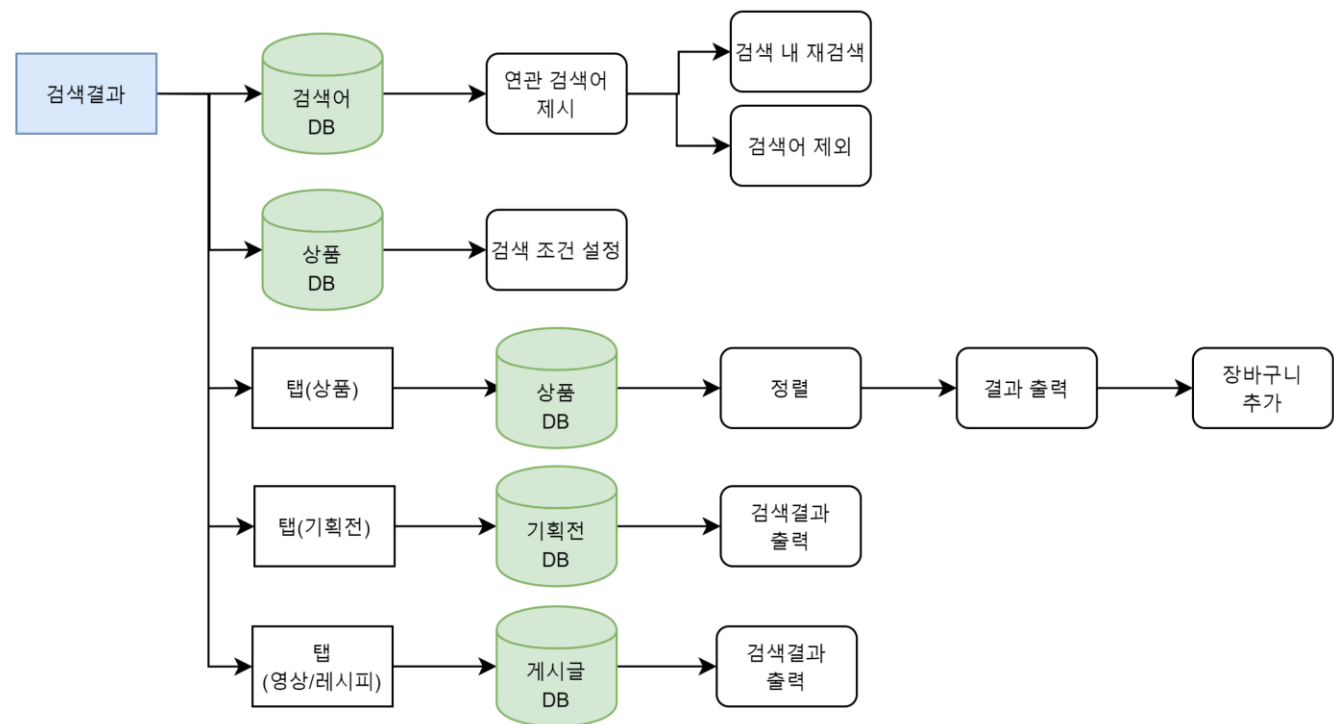
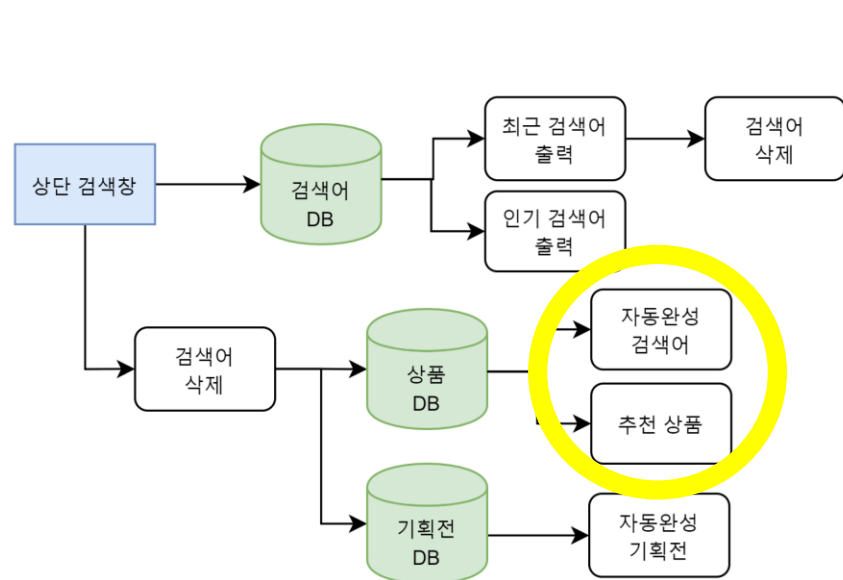


4. 개선 사항 (개선 프로세스)

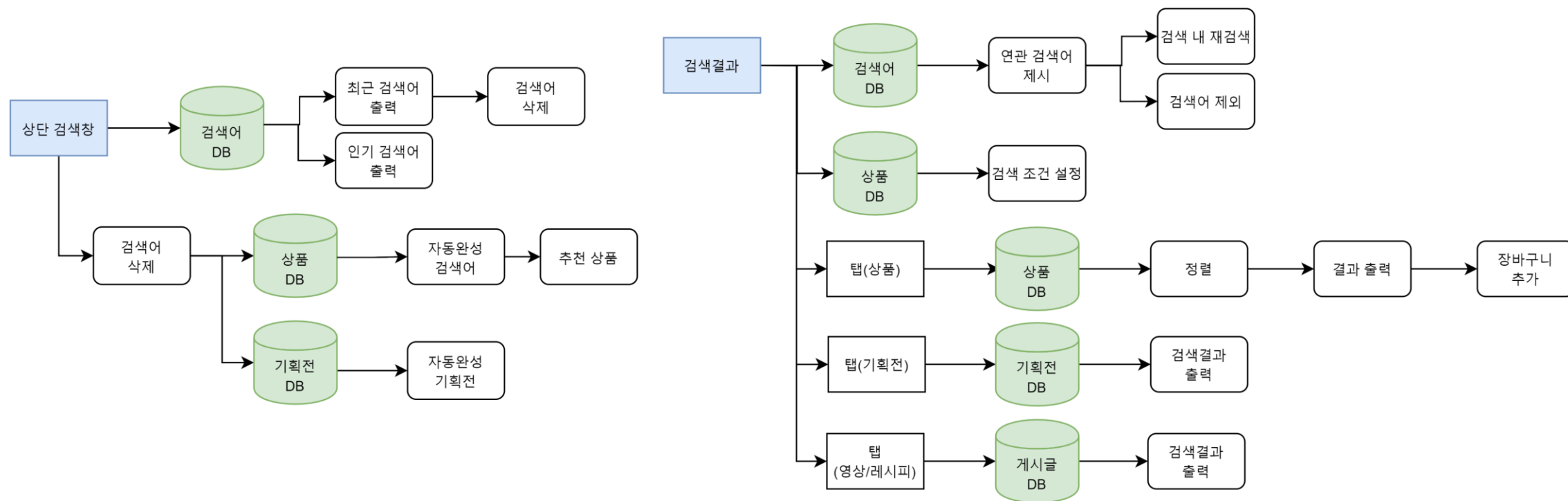


- 회원가입 시 가입 여부를 확인한 후 또 본인 인증을 하는 회원 가입의 복잡한 절차를 본인 인증으로 가입 여부까지 한번에 확인하는 프로세스로 개선

4. 개선 사항 (기존 프로세스)

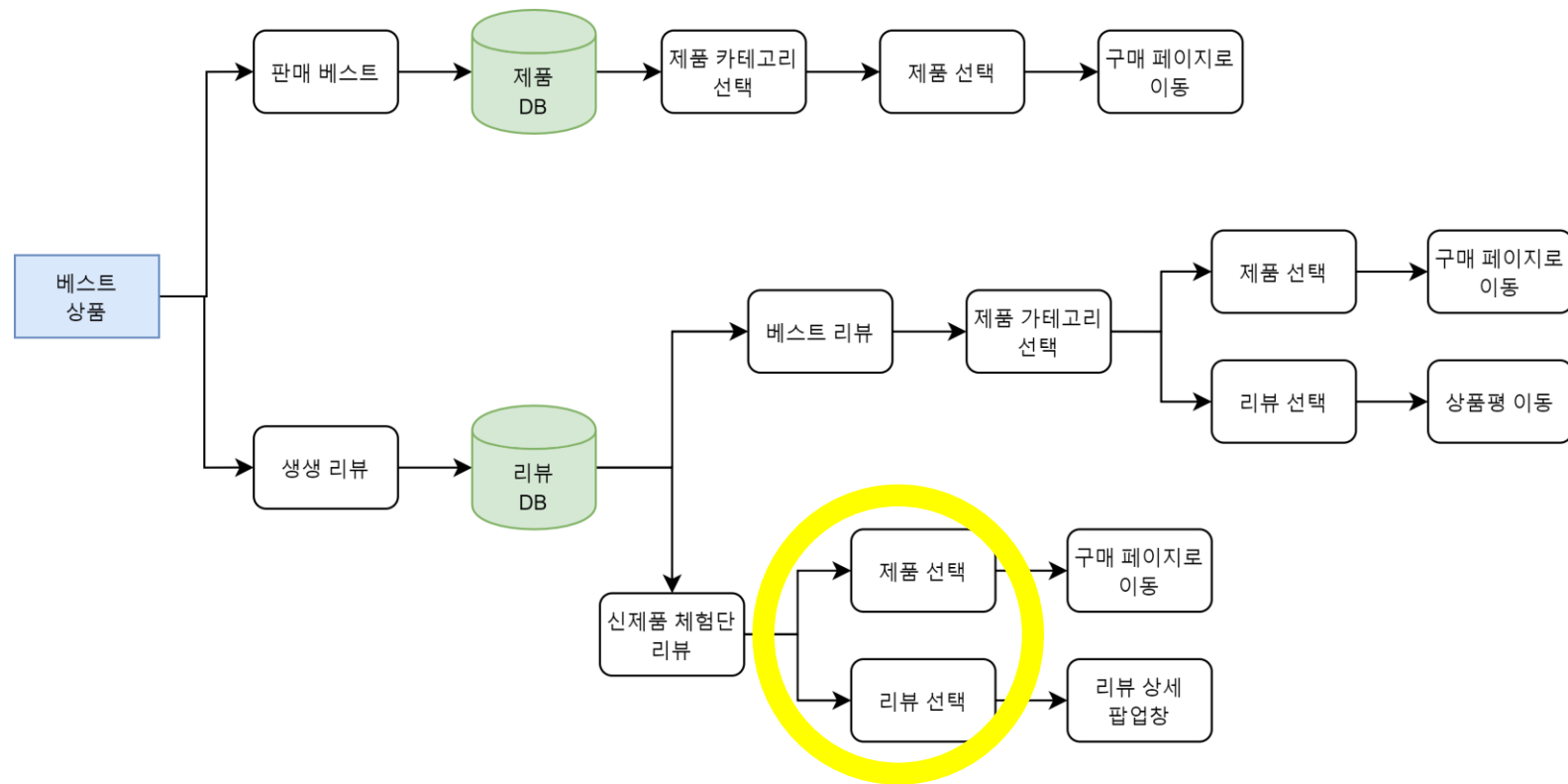


4. 개선 사항 (개선 프로세스)

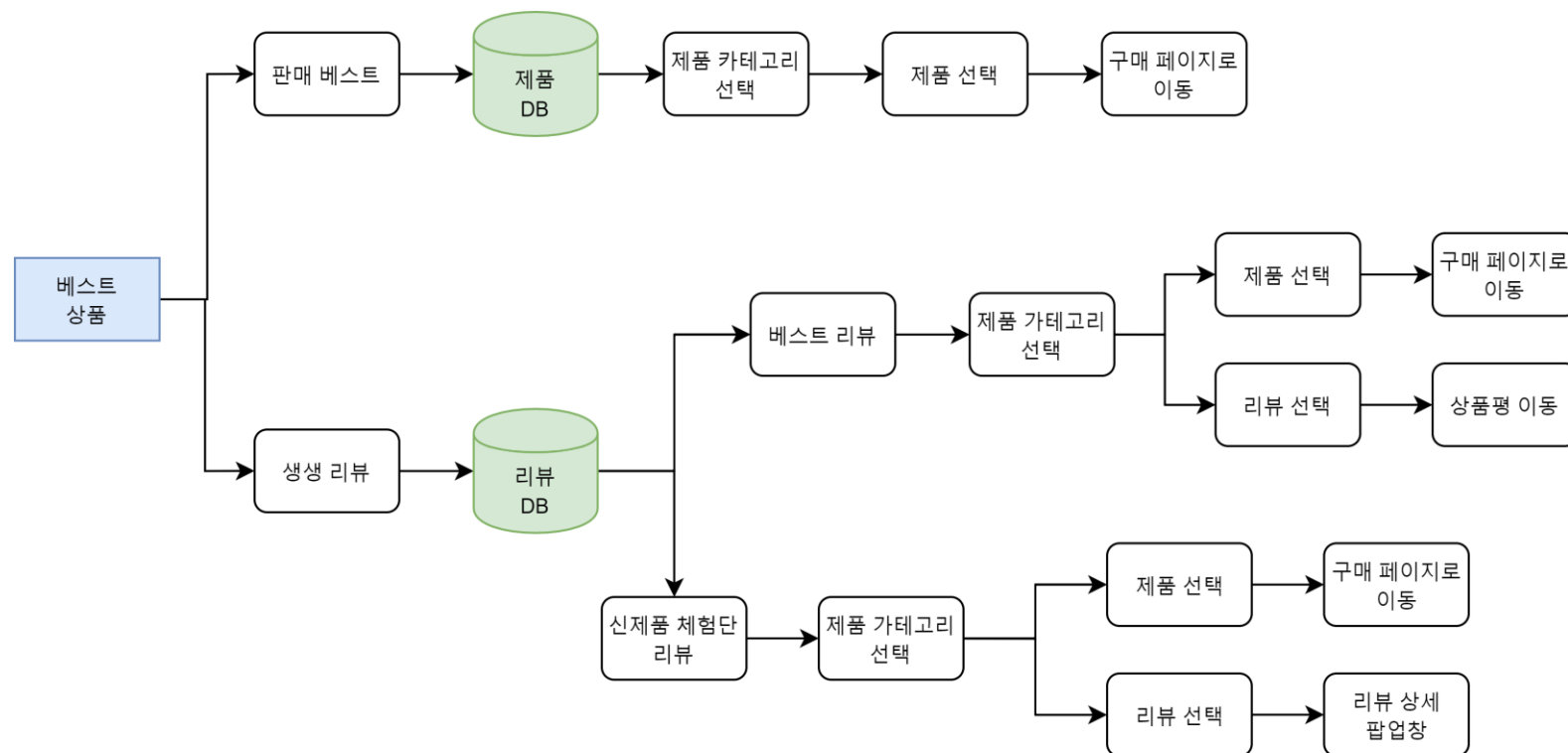


- 자동완성 검색어에서 제시하는 상품을 토대로 추천 상품을 출력 함.
- 자동완성 기능과 함께, 더 많이 검색되는 상품들에 고객이 쉽게 접근할 수 있게 함.

4. 개선 사항 (기존 프로세스)

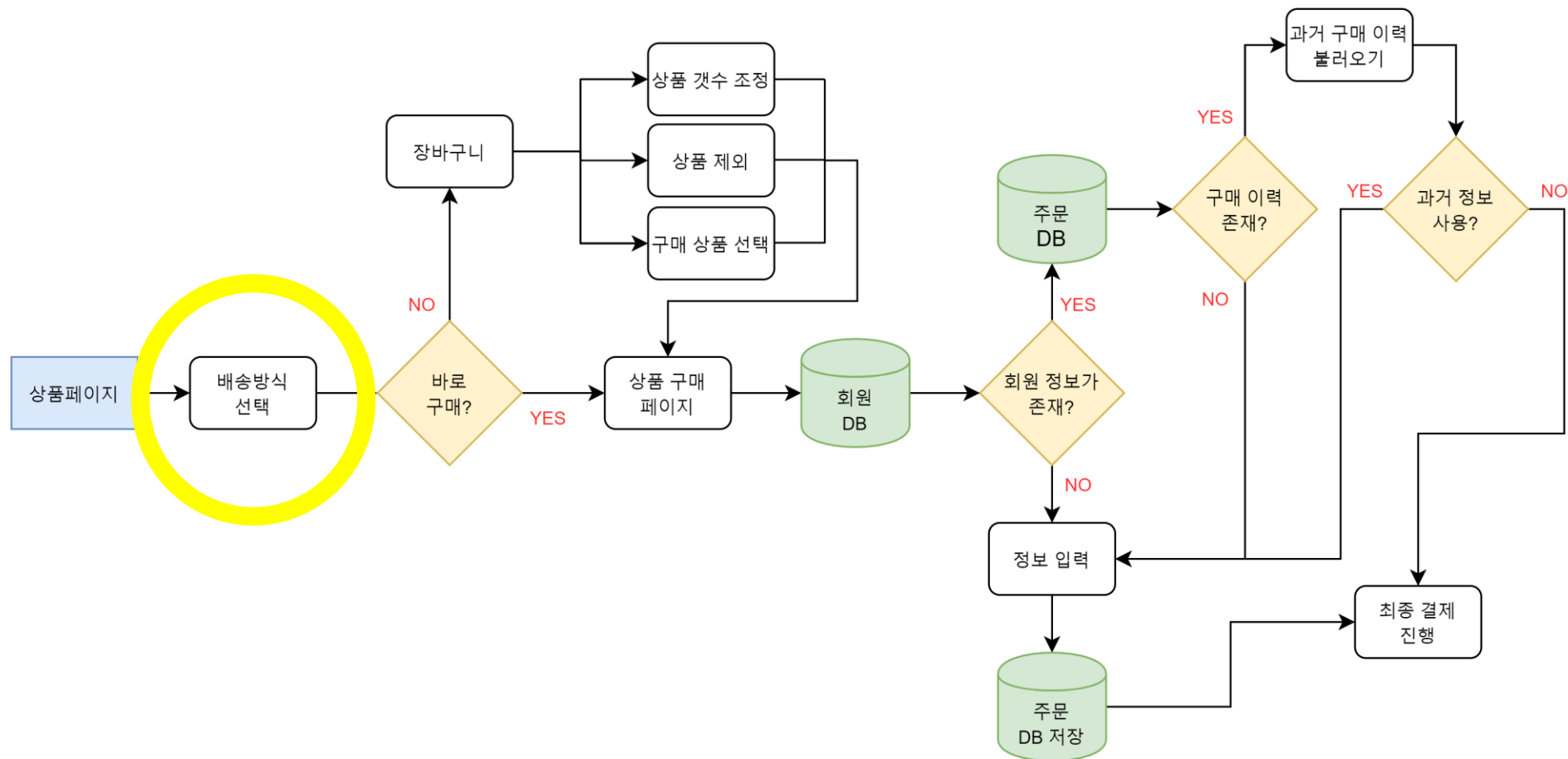


4. 개선 사항 (개선 프로세스)

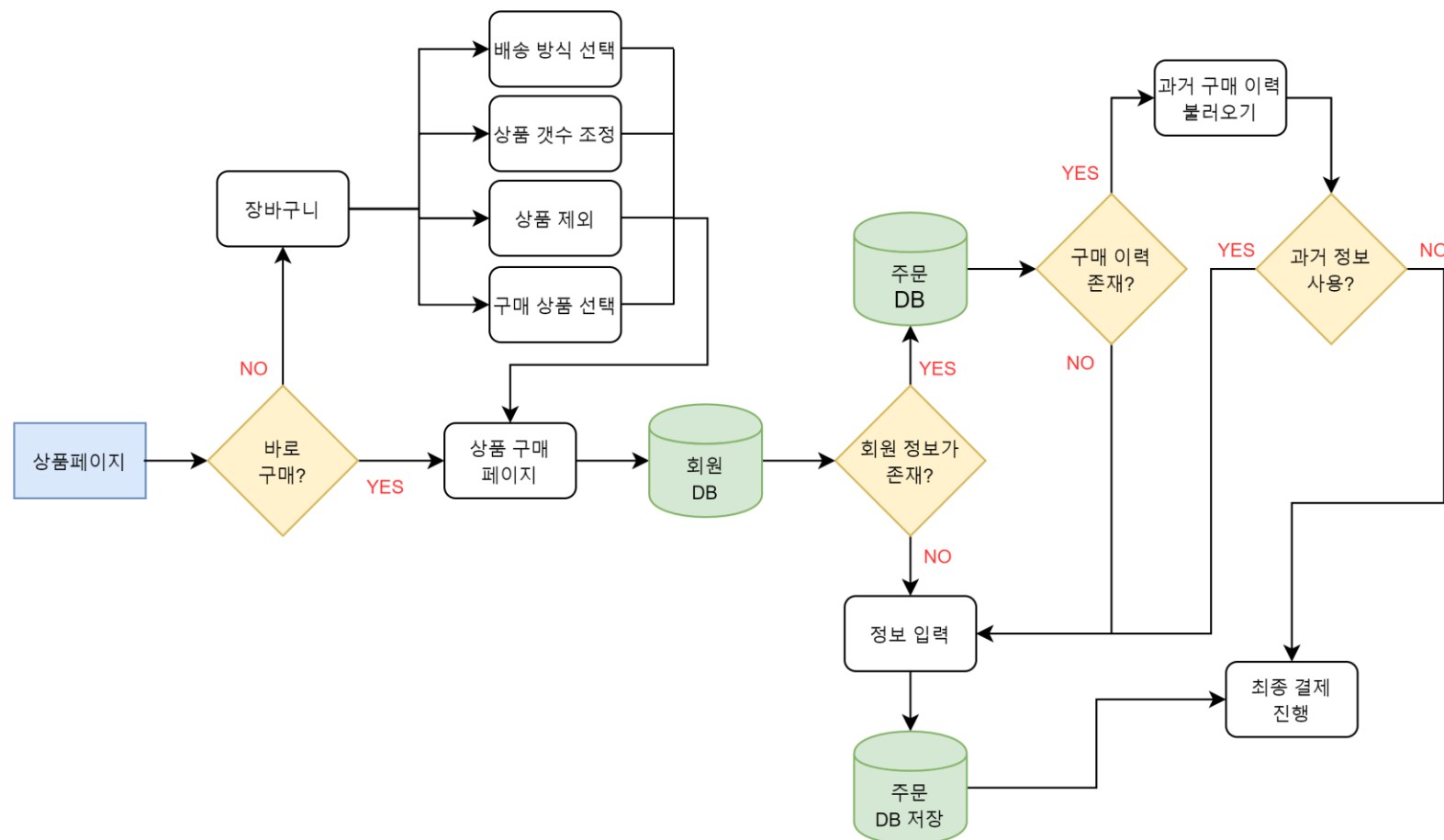


- 신제품 체험단 리뷰도 베스트리뷰와 같이 식품 카테고리로 나누어 사용자가 원하는 신제품 리뷰를 볼 수 있도록 개선한다.

4. 개선 사항 (기존 프로세스)



4. 개선 사항 (개선 프로세스)



- 배송 방식을 미리 지정해야 하는 프로세스를 배송 방식 선택을 장바구니 내에서 진행하는 기능들 중 하나로 개선하여 사용자의 편의성을 높인다

5. 기능 소스코드 (회원 가입)

이름	<input type="text"/>
아이디	<input type="text"/> <input type="button" value="중복확인"/> <input type="button" value="CJ ONE 통합 아이디 안내"/> <p>CGV, CJmail, Mnet 등 기존 CJ ONE 제품 브랜드 회원님의 경우는 기존에 사용하고 계신 아이디가 새롭게 생성하신 통합 아이디로 변경됩니다.</p>
비밀번호	<input type="text"/> <p>영문자, 숫자, 특수문자 조합하여 8~12자리, 아이디와 4자리이상 동일, 반복 문자숫자 불가 사용 가능 특수 문자 : !"#%&'()*+,-./:;<=>?@[]^_`{ }~</p>
비밀번호 확인	<input type="text"/>
생년월일	19 <input type="text"/> 년 <input type="text"/> 월 <input type="text"/> 일 <p>설정하신 생일을 기준으로 15일 전 생일쿠폰이 발행됩니다.</p>
휴대전화	010 - <input type="text"/> - <input type="text"/> <p>주은 및 배송, 쿠폰, 이벤트 정보 등을 제공 받으실 수 있습니다.</p>
이메일	<input type="text"/> @ <input type="text"/> <input type="text"/> <p>이메일 주소 입력 시 사용 가능 특수 문자 : - _ .</p>

```
public class Member {
    private String id;
    private String pass;
    private String name;
    private String birth;
    private String phoneNum;
    private String email;
}
```

<Dao>

```
// 회원가입시 처리할 메서드
public String insert(Member insert) {
    System.out.println("# 회원 DB 접근(회원 정보 등록) #");
    System.out.println(insert.getId());
    System.out.println(insert.getPass());
    System.out.println(insert.getName());
    System.out.println(insert.getBirth());
    System.out.println(insert.getPhoneNum());
    System.out.println(insert.getEmail());

    return "회원 등록 성공";
}
```

<Controller>

```
public String regMember(Member insert, Model d) {
    // 회원 가입 결과
    d.addAttribute("register_result", dao.insert(insert));

    return "호출 화면";
}
```

1. 회원 가입시 회원정보로 받는 목록들을 Member 클래스로 정리
2. insert() -> 입력한 회원 정보 목록들을 DB에 저장 완료하면 회원 가입 성공

5. 기능 소스코드 (가입 여부 확인)

가입여부 확인

<Dao>

```
// 가입여부 확인 메서드
public String exist(Member m) {
    System.out.println("# 회원 DB 접근(가입 여부 확인) #");
    System.out.println(m.getName());
    System.out.println(m.getBirth());
    System.out.println(m.getPhoneNum());

    // DB에서 받은 결과값을 가정
    return m.getName().equals("홍길동")&&
        m.getBirth().equals("770125")&&
        m.getPhoneNum().equals("56781234")?
            "이미 등록된 회원": "회원가입 가능";
}
```

<Controller>

```
public String exist(Member m, Model d) {
    // 가입 여부 확인
    d.addAttribute("check_exist", dao.exist(m));

    return "호출 화면";
}
```

1. exist() -> 회원 가입시 먼저 이름,생년월일,휴대전화번호로 입력 받아 가입여부를 확인, 등록된 회원이라 DB에 이미 동일한 정보가 존재한다면 회원 가입 불가능

5. 기능 소스코드 (로그인)

로그인

회원

비회원 (배송조회)

CJ ONE 통합 아이디 6~20자

비밀번호 영문, 특수문자, 숫자혼합 8~12자

☐ 아이디 저장

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

<Dao>

```
// 회원 로그인시 처리할 메서드
public String login(Member m) {
    System.out.println("# 회원 DB 접근(로그인) #");
    System.out.println(m.getId());
    System.out.println(m.getPass());

    // DB에서 받은 결과값을 가정
    return m.getId().equals("himan") &&
           m.getPass().equals("7777") ?
           "로그인 성공" : "로그인 실패";
}
```

<Controller>

```
public String login(Member m, Model d) {
    // 로그인 결과
    d.addAttribute("login_result", dao.login(m));

    return "호출 화면";
}
```

1. login() -> 아이디와 비밀번호를 입력 받아 DB에 저장되어 있는 회원의 아이디, 비밀번호와 일치하면 로그인 성공

5. 기능 소스코드 (상단 검색창 1)

🔍

최근 검색어		인기 검색어	21.05.24 기준
비비고	×	1 핫반	
아이시스	×	2 고메 피자	
		3 비비고	
		4 만두	
		5 김치	
		6 비비고 왕교자	
		7 big	
		8 스팸	
		9 주먹밥	
		10 냉면	

자동저장끄기
최근검색어 전체삭제
닫기

- 상단 검색창 클릭 시, 최근 검색어 이력과 인기검색어 출력.
- 검색 이력을 저장한 DB에서
 lately() -> 최근 검색했던 검색어들을 String[] lately 배열로 출력.
 ranked() -> 오늘 기준, 가장 많이 검색된 검색어들을 String[] ranked 배열로 출력

<Dao>

```
// 최근 검색어
public String[] lately(Keyword k) {
    String lately[] = {"검색어1", "검색어2", "검색어3"};

    //가장 최근에 추가된 키워드가 가장 위에 출력 됨 (index 역순 출력)
    System.out.println("# 최근 검색어 #");
    for(int i=lately.length-1; i>=0; i--) {
        System.out.println(lately[i]);
    }
    return lately;
}

// 인기 검색어
public String[] ranked(Keyword k) {
    String ranked[] = {"핫반", "고메 피자", "비비고", "만두", "김치",
        "비비고 왕교자", "big", "스팸", "주먹밥", "냉면"};

    System.out.println("# 인기 검색어 #");
    for(int i=0; i<ranked.length; i++) {
        System.out.println((i+1)+" "+ranked[i]);
    }
    return ranked;
}
```

<Controller>

```
//최근검색어
public String lately(Keyword k, Model d) {
    d.addAttribute("최근검색어", schDao.lately(k));
    return "호출화면";
}

//인기검색어
public String ranked(Keyword k, Model d) {
    d.addAttribute("인기검색어", schDao.ranked(k));
    return "호출화면";
}
```


5. 기능 소스코드 (상단 검색창 2)



1. 상단 검색창에 검색어 입력 시,
연관 검색어와 연관 기획전, 추천상품 출력.
2. autoSuggestProduct() -> 검색어와 일치하는
인기검색어, 추천상품 출력.
3. autoSuggestEvent() -> 검색어와 일치하는 기획전명 출력

<Dao>

```
//추천 상품+자동완성(상품명)
public String[] autoSuggestProduct(Keyword k) {
    String suggest[] = {"상품명1", "상품명2", "상품명3"};

    //추천 상품(자동완성 상품명 중 첫번째 순서 상품 출력)
    System.out.println("# 추천 상품 #");
    System.out.println(suggest[0]);

    //자동완성(상품명)
    System.out.println("# 자동완성 검색어 #");
    for(int i=0; i<suggest.length; i++) {
        System.out.println(suggest[i]);
    }
    return suggest;
}

//자동완성 기획전
public String[] autoSuggestEvent(Keyword k) {
    String suggest[] = {"기획전1"};

    System.out.println("# 자동완성 기획전 #");
    for(int i=0; i<suggest.length; i++) {
        System.out.println(suggest[i]);
    }
    return suggest;
}
```

<Controller>

```
//추천 상품+자동완성(상품명)
public String autoSuggestProduct(Keyword k, Model d) {
    d.addAttribute("자동완성(검색어)", schDao.autoSuggestProduct(k));
    return "호출화면";
}

//자동완성(기획전)
public String autoSuggestEvent(Keyword k, Model d) {
    d.addAttribute("자동완성(기획전)", schDao.autoSuggestEvent(k));
    return "호출화면";
}
```

5. 기능 소스코드 (상세 검색)

<Dao>

연관검색어 | [만두](#) [왕교자](#) [김치](#) [비비고죽](#) [미역국](#) [짜개](#) [육개장](#) [사골곰탕](#) [비비고밥](#) [볶음밥](#) [칼국수](#)

결과 내 재검색 ▾ Q

카테고리	<input type="checkbox"/> 가정간편식 (187)	<input type="checkbox"/> 반찬/김치 (85)	<input type="checkbox"/> 신선식품 (17)	<input type="checkbox"/> 음료/간식 (8)	<input type="checkbox"/> 대용량 식자재 (4)	<input type="checkbox"/> 건강간편식 (2)
혜택	<input type="checkbox"/> 무료배송	<input type="checkbox"/> 할인상품	<input type="checkbox"/> 덤 증정	<input type="checkbox"/> 사은품	<input type="checkbox"/> 포인트 적립	
온도·배송유형	<input type="checkbox"/> 상온·실온	<input type="checkbox"/> 냉장·냉동	<input type="checkbox"/> 업체직송	<input type="checkbox"/> 정기배송	<input type="checkbox"/> 새벽배송	
품질여부	<input type="checkbox"/> 품질제외					
브랜드	<input type="checkbox"/> 비비고(201)	<input type="checkbox"/> 기타(5)	<input type="checkbox"/> 백설(3)	<input type="checkbox"/> 하선정(2)	<input type="checkbox"/> 햇바삭김(1)	<input type="checkbox"/> 더비비고(1)
상품구성	<input type="checkbox"/> 날개상품	<input type="checkbox"/> 묶음상품	<input type="checkbox"/> 대용량상품			
별점	<input type="checkbox"/> ★★★★★	<input type="checkbox"/> ★★★★★☆	<input type="checkbox"/> ★★★★☆	<input type="checkbox"/> ★★★☆☆	<input type="checkbox"/> ★★☆☆☆	
가격	<input type="checkbox"/> 5천원이하	<input type="checkbox"/> 5천원~1만원	<input type="checkbox"/> 1만원~2만원	<input type="checkbox"/> 2만원~3만원	<input type="checkbox"/> 3만원이상	상세검색 닫기 —

초기화 ↺

1. related() -> 입력한 검색어를 기반으로 연관 검색어 출력
2. searchDetail() -> 상세검색 조건들을 담아 검색결과를 다시 요청한다.

<Controller>

```
//연관 검색어
public String[] related(Keyword k) {
    String related[] = {"연관검색어1", "연관검색어2", "연관검색어3"};

    System.out.println("# 연관 검색어 #");
    for(int i=0; i<related.length; i++) {
        System.out.print(related[i]+" ");
    }
    System.out.println();
    return related;
}

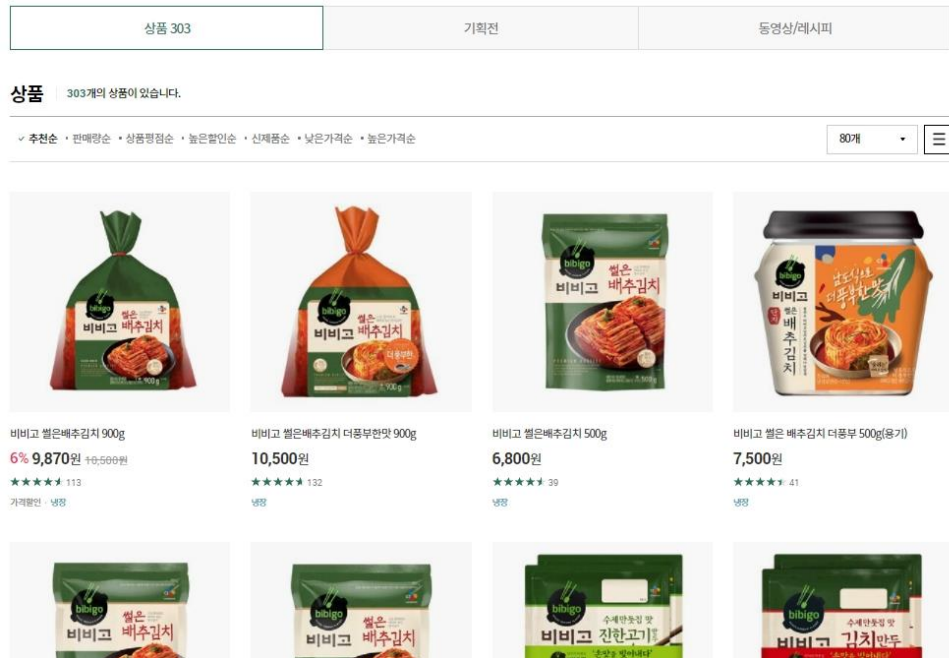
//상세 검색
public String[] searchInDetail(Keyword k) {
    //검색 조건들
    String indetail[] = {"카테고리: 가정간편식",
        "혜택: 무료배송",
        "온도/배송유형: 냉장냉동",
        "상품구성: 묶음상품",
        "별점: ★★★★★"};

    System.out.println("# 상세검색 #");
    for(int i=0; i<indetail.length; i++) {
        System.out.println(indetail[i]);
    }
    return indetail;
}
```

```
//연관검색어
public String related(Keyword k, Model d) {
    d.addAttribute("연관검색어", schDao.related(k));
    return "호출화면";
}

//상세 검색
public String searchInDetail(Keyword k, Model d) {
    d.addAttribute("상세검색", schDao.searchInDetail(k));
    return "호출화면";
}
```

5. 기능 소스코드 (검색 결과 - 상품)



1. SearchResult() → 검색어와 상세검색 조건에 의해 검색되고 정렬된 상품 배열을 출력한다.

<Dao>

```
//검색 결과 (상품)
public ArrayList<Product> SearchResult(Keyword k) {
    //상품 DB에 접근.....
    // plist = keyword에 따라 sort된 검색 결과
    ArrayList<Product> plist = new ArrayList<Product>();
    for(int i=0; i<10; i++) {
        plist.add(new Product("상품"+(i+1)));
    }

    System.out.println("# 검색결과(상품) #");
    // 검색된 상품들 (plist) 출력
    for(int i=0; i<plist.size() ; i++) {
        System.out.print(plist.get(i).getName()+" ");
        if((i+1) % 5 == 0 && i != 0 && (i + 1) != plist.size()) {
            System.out.println();
        }
    }
    System.out.println();
    return plist;
}
```

<Controller>

```
//상품검색결과
public String SearchResult(Keyword k, Model d) {
    System.out.println("===== 상품 탭 (''+k.getKeyword()+'' 검색결과) =====");
    d.addAttribute("검색결과(상품)", schDao.SearchResult(k));
    return "호출화면";
}
```

5. 기능 소스코드 (검색 결과 - 기획전)



1. SearchResultEvent() → 검색어와 상세검색 조건에 의해 검색되고 정렬된 기획전 배열을 출력한다.

<Dao>

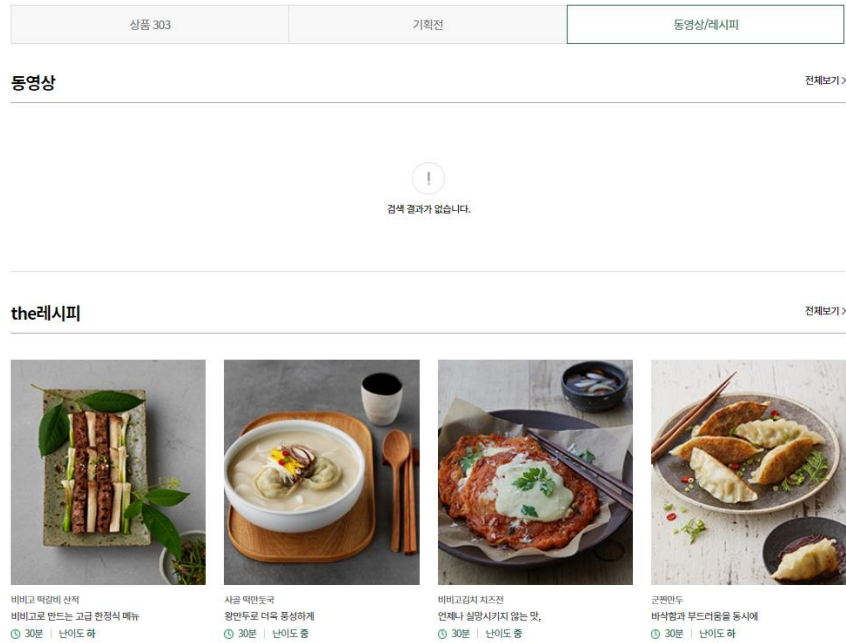
```
//검색 결과 (기획전)
public ArrayList<Content> SearchResultEvent(Keyword k) {
    // 기획전 DB에 접근.....
    // elist = keyword에 따라 sort된 검색 결과
    ArrayList<Content> elist = new ArrayList<Content>();
    for (int i = 0; i < 7; i++) {
        elist.add(new Event("기획전" + (i + 1)));
    }

    System.out.println("# 검색결과(기획전) #");
    // 검색된 기획전 (elist) 출력
    for (int i = 0; i < elist.size(); i++) {
        System.out.print(elist.get(i).getTitle() + " ");
        if ((i+1) % 5 == 0 && i != 0 && (i + 1) != elist.size()) {
            System.out.println();
        }
    }
    System.out.println();
    return elist;
}
```

<Controller>

```
//기획전 검색결과
public String SearchResultEvent(Keyword k, Model d) {
    System.out.println("===== 기획전 탭 ('"+k.getKeyword()+"' 검색결과) =====");
    d.addAttribute("검색결과(기획전)", schDao.SearchResultEvent(k));
    return "호출화면";
}
```


5. 기능 소스코드 (검색 결과 - 동영상/레시피)



<Dao>

```
// 검색 결과 (동영상)
public ArrayList<Content> SearchResultMovie(Keyword k) {
    // 동영상 DB에 접근.....
    // plist = keyword에 따라 sort된 검색 결과
    ArrayList<Content> mlist = new ArrayList<Content>();
    for (int i = 0; i < 3; i++) {
        mlist.add(new Movie("동영상" + (i + 1)));
    }

    System.out.println("# 검색결과(동영상) #");
    // 검색된 동영상 (mlist) 출력
    for (int i = 0; i < mlist.size(); i++) {
        System.out.print(mlist.get(i).getTitle() + " ");
        if ((i + 1) % 5 == 0 && i != 0 && (i + 1) != mlist.size())
            System.out.println();
    }
    System.out.println();
    return mlist;
}
```

```
// 검색 결과 (레시피)
public ArrayList<Content> SearchResultRecipe(Keyword k) {
    // 레시피 DB에 접근.....
    // rlist = keyword에 따라 sort된 검색 결과
    ArrayList<Content> rlist = new ArrayList<Content>();
    for (int i = 0; i < 18; i++) {
        rlist.add(new Recipe("레시피" + (i + 1)));
    }

    System.out.println("# 검색결과(레시피) #");
    // 검색된 레시피 (rlist) 출력
    for (int i = 0; i < rlist.size(); i++) {
        System.out.print(rlist.get(i).getTitle() + " ");
        if ((i + 1) % 5 == 0 && i != 0 && (i + 1) != rlist.size()) {
            System.out.println();
        }
    }
    System.out.println();
    return rlist;
}
```

<Controller>

```
//동영상 검색결과
public String SearchResultMovie(Keyword k, Model d) {
    System.out.println("동영상 탭 (" + k.getKeyword() + " 검색결과)");
    d.addAttribute("검색결과(동영상)", schDao.SearchResultMovie(k));
    return "호출화면";
}

//레시피 검색결과
public String SearchResultRecipe(Keyword k, Model d) {
    System.out.println("레시피 탭 (" + k.getKeyword() + " 검색결과)");
    d.addAttribute("검색결과(레시피)", schDao.SearchResultRecipe(k));
    return "호출화면";
}
```

1. SearchResultMovie()
2. SearchResultRecipe()
 - 검색어와 상세검색 조건에 의해 검색되고 정렬된 콘텐츠들을 출력한다.

5. 기능 소스코드 (장바구니 담기)



<Dao>

```
//DB에서 상품 가격 정보 가져오기
public int getprice() {
    return price;
}

//상품 총계 계산
public int Total(int price, int purchaseCnt) {
    //(1) 상품 총액이 정액금액(3만원)미만일 경우
    if (price*purchaseCnt<30000) {
        int total=price*purchaseCnt+3000;
    }

    //(2) 상품 총액이 정액금액(3만원)을 초과할경우
    else {
        int total=price*purchaseCnt;
    }

    return total;
}

//상품 주문 회원명
public String member() {
    return id;
}
```

<Controller>

```
public class Cart {
    int purchaseCnt=0;
    int price=0;
    String name;
    String id;
    int total;
    //장바구니 상품 입력
    public Cart(String name, int price, int purchaseCnt) {
        this.name=name;
        this.price=price;
        this.purchaseCnt=purchaseCnt;
    }

    //장바구니에 넣을 상품의 수량
    public int getCnt() {
        return purchaseCnt;
    }

    //장바구니에 넣을 상품의 종류
    public String name() {
        return name;
    }
}
```

5. 기능 소스코드 (장바구니)

장바구니

the프라임 주문할 때마다 7% 추가 할인
[가입하기](#)

일반 (2)	정기배송 (1)	선물세트 (0)
--------	----------	----------

☒ 전체선택 | ☐ 선택삭제 | ☐ 품질/마감삭제
 🕒 정기배송담기

☒ 냉장&냉동

☒

냉동
비비고 수제세우만두 325g*2x2개
0 P
05/27(목) 배송 예정 ?

- 2 +

33,920원 ?

바로구매

🗑

☒

냉동
고메 베이커리 크로와상 630g
159 P
05/27(목) 배송 예정 ?

- 1 +

15,980원 ?

바로구매

🗑

상품금액 49,900원

-

할인금액 0원

+

배송비 0원

=

49,900원

결제정보 (2)

총 상품금액 49,900원
총 할인금액 -0원
총 배송비 +0원

결제예정금액 **49,900원**
적립예정 포인트 ? 159P

구매하기

🎁 선물

<Dao>

```
public class Purchase {
    itemDao A=new itemDao();

    public String selectedItem(String a) {
        String select=A.getItem(); //장바구니 내 존재하는 물건 중 구매할 물건 선택
        return select;
    }

    public int Itemprice(int b) { //구매할 물건 총계
        int total=A.getPrice();
        return total;
    }
}
```

<Controller>

```
public class Cart1 {

    public static void main(String[] args) {

        List<String> purchaseList= new ArrayList<String>();
        List<Integer> purchaseTotal=new ArrayList<Integer>();
        Cart CartA=new Cart("-", 0, 0); //장바구니에 상품등록

        purchaseList.add("구입예정상품"); //장바구니에 물건 추가
        purchaseList.remove(0); //장바구니에 넣었던 상품 삭제

        System.out.println("총 가격은"+CartA.Total(0, 0)); //장바구니에 담긴 상품 총계

    }
}
```

5. 기능 소스코드 (상품 주문)

주문/결제

주문자 정보

주문자명

[운송장예시](#)

배송지 설정

여러 곳으로 보내기

1곳 ▾

배송지

☐ 기존 배송지
 ☒ 새주소 입력

받는 분 *

최대 10자

☐ 주문자와 동일

주소 *

주소검색

상세주소 입력

☐ 기본배송지로 등록

연락처1 *

010 ▾

-

-

(!) 배송 알림 받으실 휴대폰번호를 입력해주세요.

연락처2

010 ▾

-

-

일반배송 요청사항

일반배송 요청사항 선택 ▾

최종 결제정보

총 상품금액 49,900원

총 배송비 0원

총 할인금액 - 8,480원

포인트/기타 결제 - 0원

최종 결제금액 41,420원

L 적립예정 포인트 159P

☐ 주문 약관 모두 동의

[결제하기](#)

<Dao>

```
public class Purchase {
    public static void main(String[] args) {
        MemberDao A=new MemberDao();
        String memberName=A.Membername(); //DB에서 회원정보 (회원명, 주소) 호출
        String Address;
        //과거 구매이력 여부에 따라
        //(1)새로운 주소를 입력받거나
        if(A.pastpurchase()==null) {
            Scanner S=new Scanner(System.in);
            Address=S.nextLine();
        }
        //(2) DB에 기록된 주소를 불러옴
        else {
            Address=A.getAddress();
            String pastpur=A.pastpurchase();
        }
    }
}
```

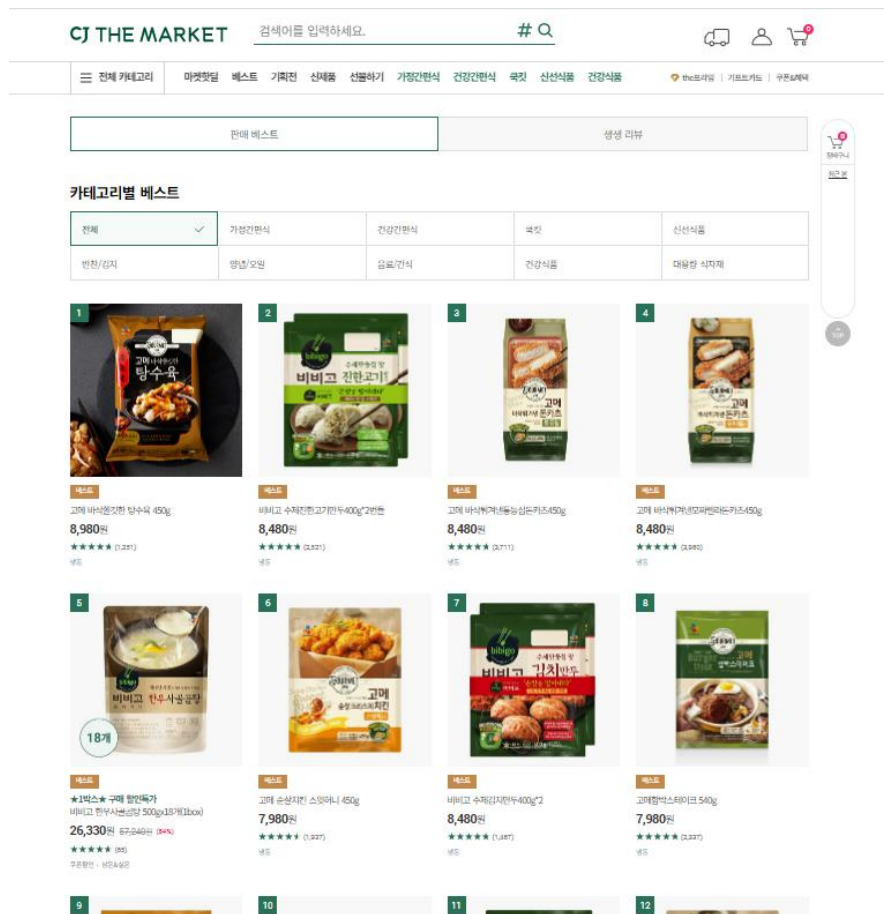
<Controller>

//DAO에서 전달받은 결과값 화면에 표시

```
System.out.println("구매자명"+memberName);
System.out.println("주소"+Address);
```

```
System.out.println("결제방식 및 혜택 선정");
```


5. 기능 소스코드 (판매 베스트)



1. Productlist() -> 제품DB에 저장된 정보를 기반으로 제품 정보(이름, 가격, 순위, 별점)를 리스트로 출력
2. regProduct() -> 제품DB에 제품 정보(이름, 가격, 순위, 별점)를 리스트로 등록

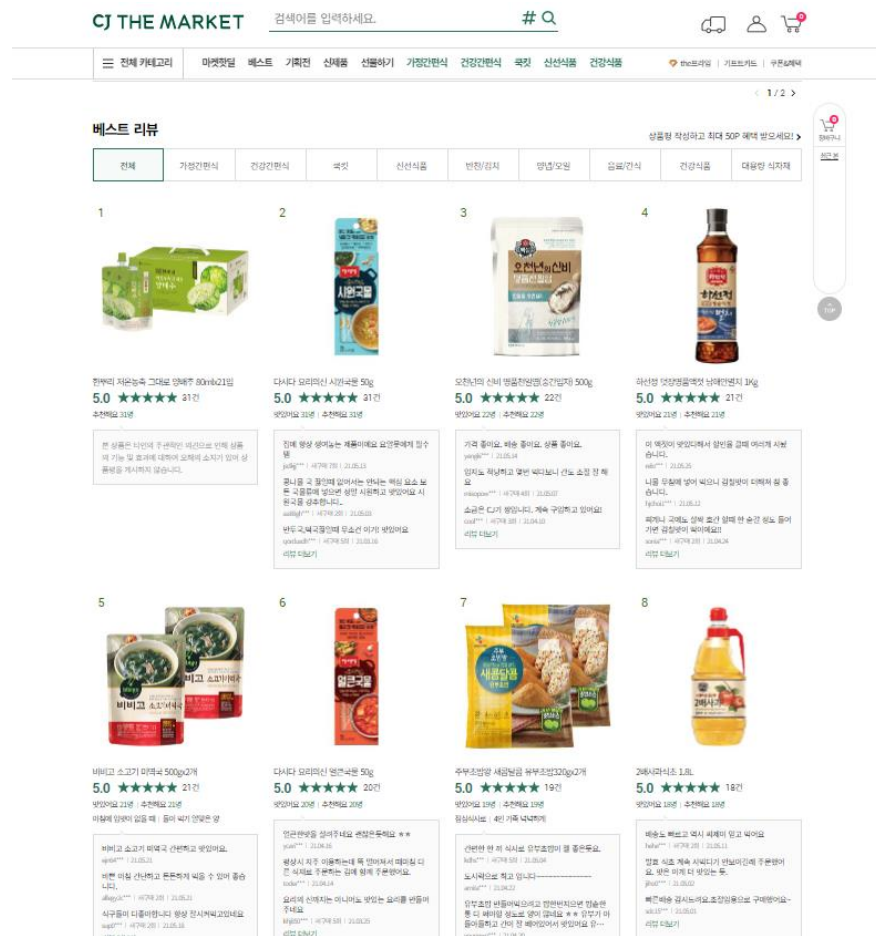
<Dao>

```
public class ProductDao {
    private ArrayList<FoodProduct> list;
    private ArrayList<ReviewList> list1;
    //제품 등록
    public String insert(FoodProduct insert) {
        list = new ArrayList<FoodProduct>();
        System.out.println("DB 처리(상품등록)");
        System.out.print(insert.getName()+"\t");
        System.out.print(insert.getPrice()+"\t");
        System.out.print(insert.getRank()+"\t");
        System.out.println(insert.getRating());
        list.add(insert);
        return "상품 등록 성공";
    }
    //제품 리스트
    public void foodlistout(){
        for(FoodProduct Food:list) {
            Food.info();
        }
    }
}
```

<Controller>

```
private ProductDao dao = new ProductDao();
//제품 등록 결과 호출
public String regProduct(FoodProduct insert) {
    System.out.println("최종결과 "+dao.insert(insert));
    return "호출할화면";
}
//제품 리스트 호출
public void Productlist() {
    dao.foodlistout();
}
//리뷰 등록 결과 호출
public String regReviewList(ReviewList insert) {
    System.out.println("최종결과 "+dao.insert(insert));
    return "호출할화면";
}
//리뷰 리스트 호출
public void ReviewList() {
    dao.reviewlistout();
}
```

5. 기능 소스코드 (생생 리뷰)



1. regReview() -> 제품DB에 저장된 정보를 기반으로 제품 정보(이름, 가격, 순위, 별점, 리뷰)를 리스트로 출력
2. ReviewList() -> 제품DB에 제품 정보(이름, 가격, 순위, 별점, 리뷰)를 리스트로 등록

<Dao>

```
//리뷰 등록
public String insert(ReviewList insert) {
    list1 = new ArrayList<ReviewList>();
    System.out.println("DB 처리(리뷰등록)");
    System.out.print(insert.getName()+"\t");
    System.out.print(insert.getPrice()+"\t");
    System.out.print(insert.getRank()+"\t");
    System.out.print(insert.getRating()+"\t");
    System.out.println(insert.getReivew());
    list1.add(insert);
    return "리뷰 등록 성공";
}

//리뷰 리스트
public void reviewlistout(){

    for(ReviewList reivew:list1) {
        reivew.info();
    }
}
```

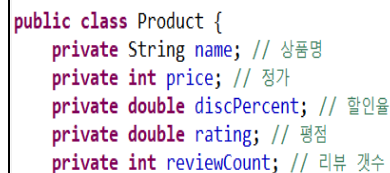
<Controller>

```
private ProductDao dao = new ProductDao();
//제품 등록 결과 호출
public String regProduct(FoodProduct insert) {
    System.out.println("최종결과 "+dao.insert(insert));
    return "호출할하면";
}

//제품 리스트 호출
public void Productlist() {
    dao.foodlistout();
}

//리뷰 등록 결과 호출
public String regReviewList(ReviewList insert) {
    System.out.println("최종결과 "+dao.insert(insert));
    return "호출할하면";
}

//리뷰 리스트 호출
public void ReviewList() {
    dao.reviewlistout();
}
```



```
// Product 모델의 데이터를 array에 넣어준다.
public void insert(Product product) {
    Product productModel = new Product(
        product.getName(),
        product.getPrice(),
        product.getDiscPercent(),
        product.getRating(),
        product.getReviewCount());
    productList.add(productModel);
}

// ProdcutDao 클래스의 array에 저장된 모델값을 출력
public void getProductList() {
    // productList에 담긴 모델들의 데이터를 1개씩 출력
    for (Product product : productList) {
        product.ProductData();
    }
}
```

```
public class Controller {
    private ProductDao productDao = new ProductDao();

    // Product 모델의 데이터를 ProductDao 클래스의 array에 넣어줍니다
    public void registProduct(Product product) {
        productDao.insert(product);
    }

    // ProductDao 클래스의 array에 저장된 모델값을 출력합니다
    public void productList() {
        productDao.getProductList();
    }
}
```

감사합니다