

## 데이터베이스 프로그래밍 최종보고서

01분반 7팀  
20160947 김효경 (팀장)  
20160952 박나현  
20160965 성연주  
20160986 이유주

# 00 목 차

1

시스템 개요 및 특징

6

데이터베이스 설계

2

핵심 기능 및 세부 기능

7

클래스 설계

3

Use case

8

Request URI

4

시스템 구성도

9

구현 방법

5

UI 설계

10

결론, 소감 및 업무분장

# 01 시스템 개요 및 특징

저희 팀은 어른들이 많이 가지고 노는 장난감인 슬라임을 파는 쇼핑몰을 구현하였습니다. 쇼핑몰의 기본적인 기능인 로그인, 회원가입, 주문, 장바구니 등의 기능을 구현하였으며 그 외에 사용자가 질문을 하고 관리자가 답변을 할 수 있는 게시판 기능도 구현하였습니다.

특히, DIY 메뉴를 구현하여 사용자가 원하는 대로 슬라임을 조합하여 주문할 수 있도록 하였습니다.

# 02 핵심기능 및 세부기능



암암크런치

판매가 9000

수량 2

배송방법 택배

배송비 3000원(40,000원 이상 구매 시 무료)

총금액 18000

상세정보 엘머스폴 베이스에 노란 폼볼이 들어간 바나나향이 폼크런치 슬라임

장바구니

취소

맨 처음 홈페이지에서 완제품 이름을 클릭하면 해당 상품 페이지로 이동합니다.  
수량을 누른 후 수량을 입력하면 총 금액이 표시되고, 장바구니 버튼을 누르면  
장바구니 창으로 이동하며 해당 상품이 장바구니에 추가됩니다.

# 02 핵심기능 및 세부기능

## DIY SLIME

[total\\_slime](#) [diy\\_slime](#) [board](#) [basket](#) [logout\( admin \)](#) [myPage](#)

Slime Color

brown ▼

Slime Deco

foam ▼

Slime Scent

banana ▼

Slime Glue

school glue ▼

이름

test

판매가

8500

수량

1

배송방법

택배

배송비

3000원(40,000원 이상 구매 시 무료)

총금액

8500

장바구니

취소

diy\_slime 버튼을 누르면 diy메뉴로 이동합니다.  
색, 데코, 향, 풀의 종류를 직접 선택할 수 있으며  
선택한 옵션에 따라 가격이 달라집니다.

완제품과 마찬가지로 수량을 정할 수 있으며,  
사용자가 원하는 이름을 적어 장바구니 버튼을  
누르면 장바구니에 저장됩니다.

# 02 핵심기능 및 세부기능

## 완제품 목록



암암크런치

수량 : 2

9000

삭제

합계 : 18000

## DIY 목록

test

Scent : banana

Deco : foam

Color : brown

Glue : school

glue

수량 : 1

8500

합계 : 8500

DIY SLIME

삭제

주문하기

취소

장바구니에 들어가면 완제품과 DIY 의 목록을 따로 보여줍니다. 장바구니는 세션에 저장되며, DIY의 경우 자신이 선택한 옵션들을 볼 수 있도록 했으며 삭제를 누를 경우 장바구니에서 삭제됩니다.

주문하기 버튼을 누르면 주문하기 창으로 넘어 갑니다.

# 02 핵심기능 및 세부기능

## DIY SLIME

[total\\_slime](#) [diy\\_slime](#) [board](#) [basket](#) [logout\( peanut \)](#) [myPage](#)

주문자 정보

성명

주소

결제 정보

총금액 50600원

배송비 0원

최종 결제 금액 50600원

결제 수단 무통장 입금

입금 계좌 슬라임 요정 국민 12345-45678989

주문하기 창에 들어오면 금액 정보를 띄워줍니다.  
주소는 기본적으로 회원가입시 입력했던 주소를 띄워주고, 사용자가 원한다면 수정할 수 있습니다.  
(이 경우 사용자의 정보가 바뀌는 것이 아닌 주문 주소지만 바뀌는 것 입니다.)

배송비는 40000원 이상일 경우만 제외되며 결제하기 버튼을 누르면 주문이 완료되었다는 메시지를 띄워줍니다. 여기서 주문 번호를 클릭하면 해당 주문 내역을 상세히 볼 수 있습니다.



감사합니다. 주문이 정상적으로 완료되었습니다.

고객님의 주문번호는 28번입니다.

# 02 핵심기능 및 세부기능

## DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

### 사용자 정보

이름 admin

ID admin

Address admin

### 주문 내역

주문번호	배송지 정보	총금액
[2018/12/27] 28	admin	29500
[2018/12/26] 21	동덕여대	27600
[2018/12/24] 1	동덕여대	112700

MyPage에 들어가면 사용자 정보와 지금까지의 주문 내역을 보여줍니다.  
주문번호를 클릭하면 해당 주문의 상세정보를 볼 수 있습니다.



### 주문 내역

완제품 주문 내역



피크런지

수량 : 2

8500

합계 : 17000

### DIY 주문 내역

DIY SLIME

Scent : mango

Deco : glitter

Color : brown

Glue : amos

수량 : 2

7700

합계 : 15400



## 02 핵심기능 및 세부기능

## DIY SLIME

[total\\_slime](#)
[diy\\_slime](#)
[board](#)
[basket](#)
[logout\( peanut \)](#)
[myPage](#)

No	제목	ID	날짜
5	드디어 다 끝나가요...	admin	2018/12/24
4	문의 드립니다.	peanut	2018/12/24
3	오늘이 무슨날인가요?	hi	2018/12/24
1	피그크런치 문의	peanut	2018/12/23

[질문등록](#)

copyright© 2018 slime fairies all rights reserved

상단의 board를 누르면 게시판 목록으로 이동합니다. 질문 등록을 누르면 질문을 등록할 수 있고, 게시글의 제목을 클릭하면 해당 게시글 내용을 볼 수 있는 창으로 이동합니다.

# 02 핵심기능 및 세부기능

## DIY SLIME

total\_slime diy\_slime board basket logout( peanut ) myPage

제목

내용

질문등록 창에 들어오면 제목과 질문을 입력할 수 있습니다. 답변은 관리자가 입력해야 하기 때문에 일반 사용자가 입력할 수 없습니다. 따라서 사용자가 질문을 등록하면 관리자가 답변을 하기 전까지 목록에서 아래와 같이 보여집니다.



답변

판매자가 아직 답변을 등록하지 않았습니다.

질문등록

목록

# 02 핵심기능 및 세부기능

## DIY SLIME

[total\\_slime](#) [diy\\_slime](#) [board](#) [basket](#) [logout\( peanut \)](#) [myPage](#)

제목 피그크런치 문의

등록 날짜 2018/12/23

질문 피그크런치의 돼지바탕은 어떤 향인가요?

답변 딸기향과 브라우니향을 믹스한 향입니다.

[수정](#) [삭제](#) [답변](#) [목록](#)

제목을 클릭해 해당 게시글 내용으로 들어오면 본인의 게시글인 경우 수정과 삭제가 가능하며, 관리자의 경우 삭제와 답변이 가능합니다. 목록을 클릭하면 이전 페이지의 게시글 목록으로 돌아갑니다.

# 02 핵심기능 및 세부기능

## DIY SLIME

total\_slime diy\_slime board basket logout( peanut ) myPage

제목  
등록 날짜

피그크런치 문의

2018/12/23

내용

피그크런치의 돼지바탕은 어떤 향인가요?

수정

목록

copyright© 2018

게시글 수정창에 들어오면 기존 내용이 담겨있고, 이것을 수정할 수 있습니다. 수정버튼을 클릭하면 게시글 목록으로 이동하며 내용이 수정됩니다.

등록날짜는 DB에서 SYSDATE로 자동으로 생성되기 때문에 수정할 수 없습니다.

만약 본인이 작성한 글이 아니라면 수정창으로 들어오지 못하고, 앞의 창에서 아래와 같은 에러 메시지를 띄웁니다.



본인이 쓰지 않은 글은 수정할 수 없습니다.

## 02 핵심 기능 및 세부 기능

localhost:8081 내용:

정말 삭제하시겠습니까?

확인

취소

삭제 버튼을 누르면 다음과 같은 알림 창을 띄우고,  
확인을 누르면 게시글이 삭제됩니다.

만약 본인이 작성한 게시글이 아니라면 아래와 같  
은 에러 메시지를 띄웁니다.



다른 사용자의 글은 삭제할 수 없습니다.

# 02 핵심기능 및 세부기능

## DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

내용

답변 등록

목록

관리자가 답변 버튼을 클릭하면 해당 질문에 답변을 할 수 있습니다. 답변등록을 누르면 답변이 등록됩니다.

만약 이전 페이지에서 관리자가 아닌 사용자가 답변 버튼을 클릭할 시 아래와 같은 에러 메시지를 출력합니다.



관리자만 답변할 수 있습니다.

# 02 핵심기능 및 세부기능

## 로그인

사용자 ID

비밀번호

로그인

회원가입

## 회원가입

사용자 ID

비밀번호

비밀번호 확인

이름

주소

회원 가입

목록

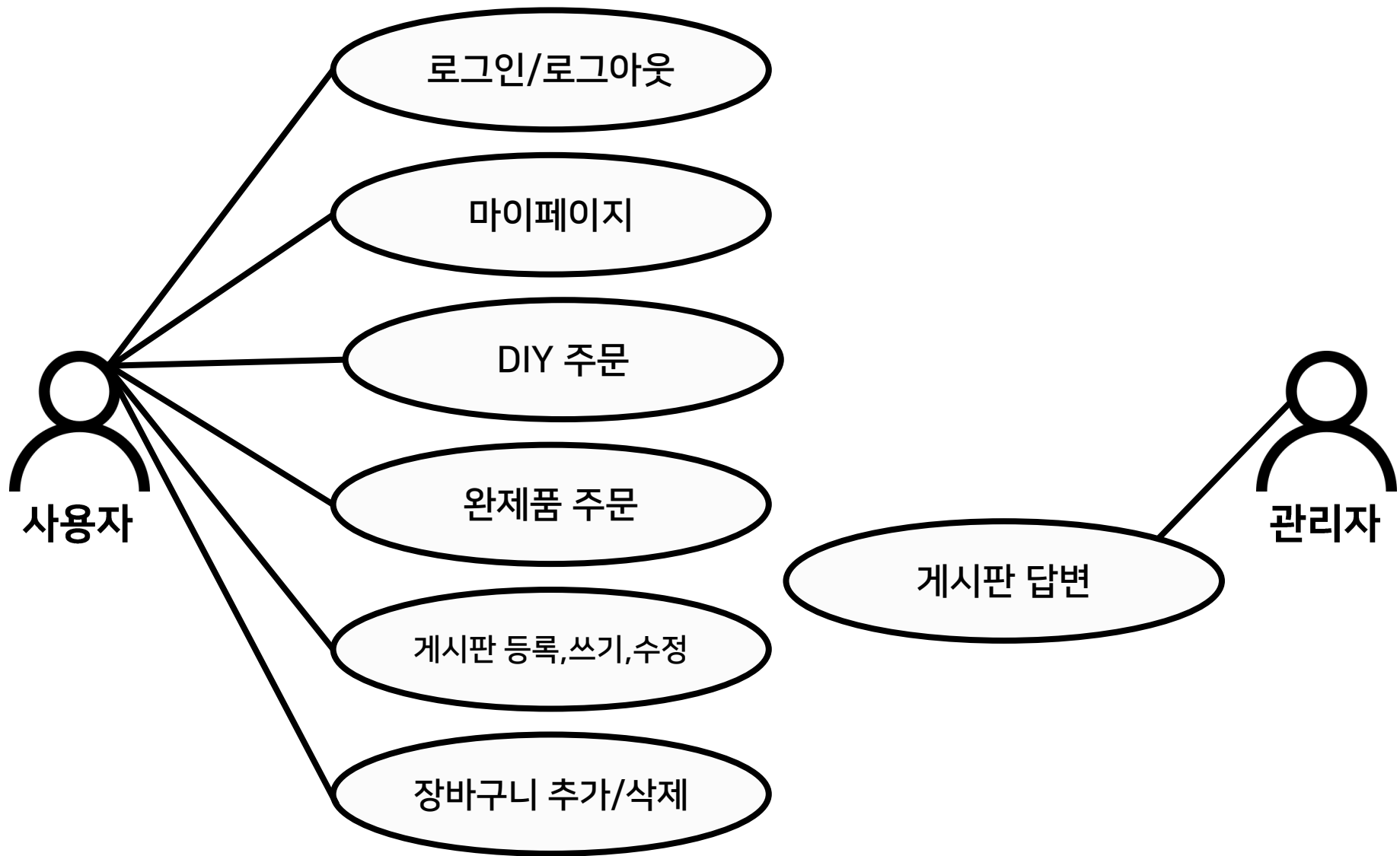
# DIY SLIME

total\_slime diy\_slime board basket **logout( admin )** myPage



헤더부분을 통해서 해당 페이지로 이동할 수 있으며, logout 상태에서는 login메뉴로 보여지며 login 상태에서는 logout ( userId ) 형태로 보여지게 됩니다.  
Login은 session에 정보를 저장하고, logout은 session에서 정보를 제거하는 형식입니다.

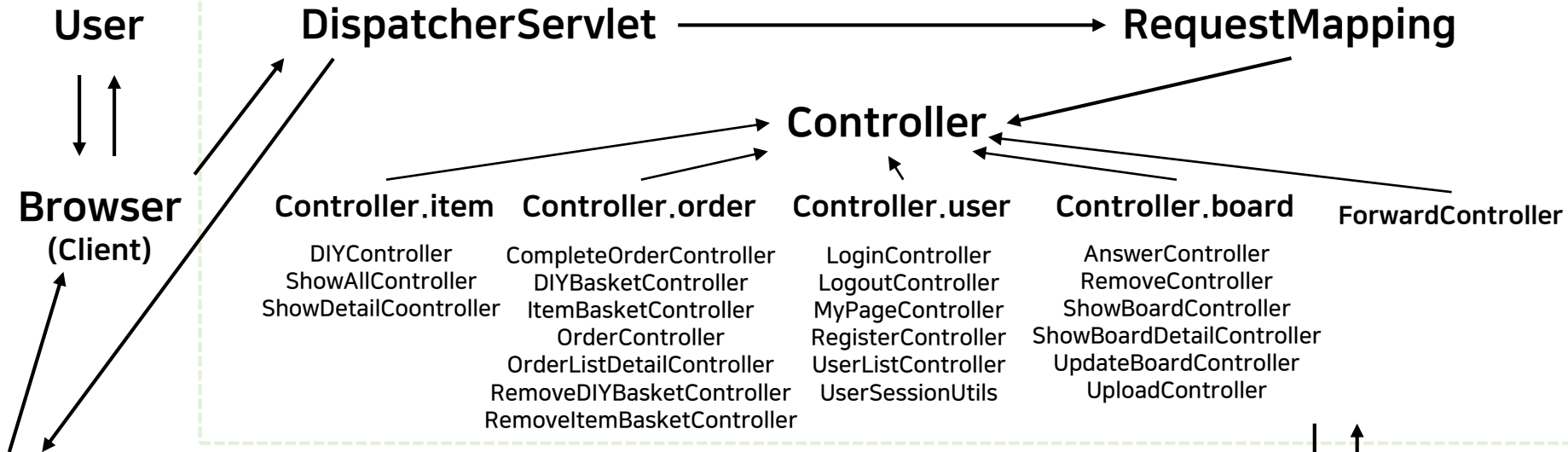
# 03 Use Case





# 04 시스템 구성도

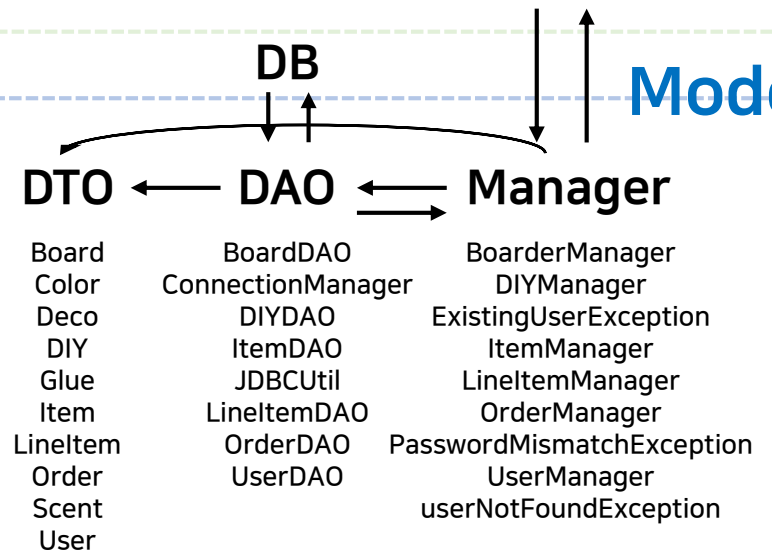
## Controller



## View

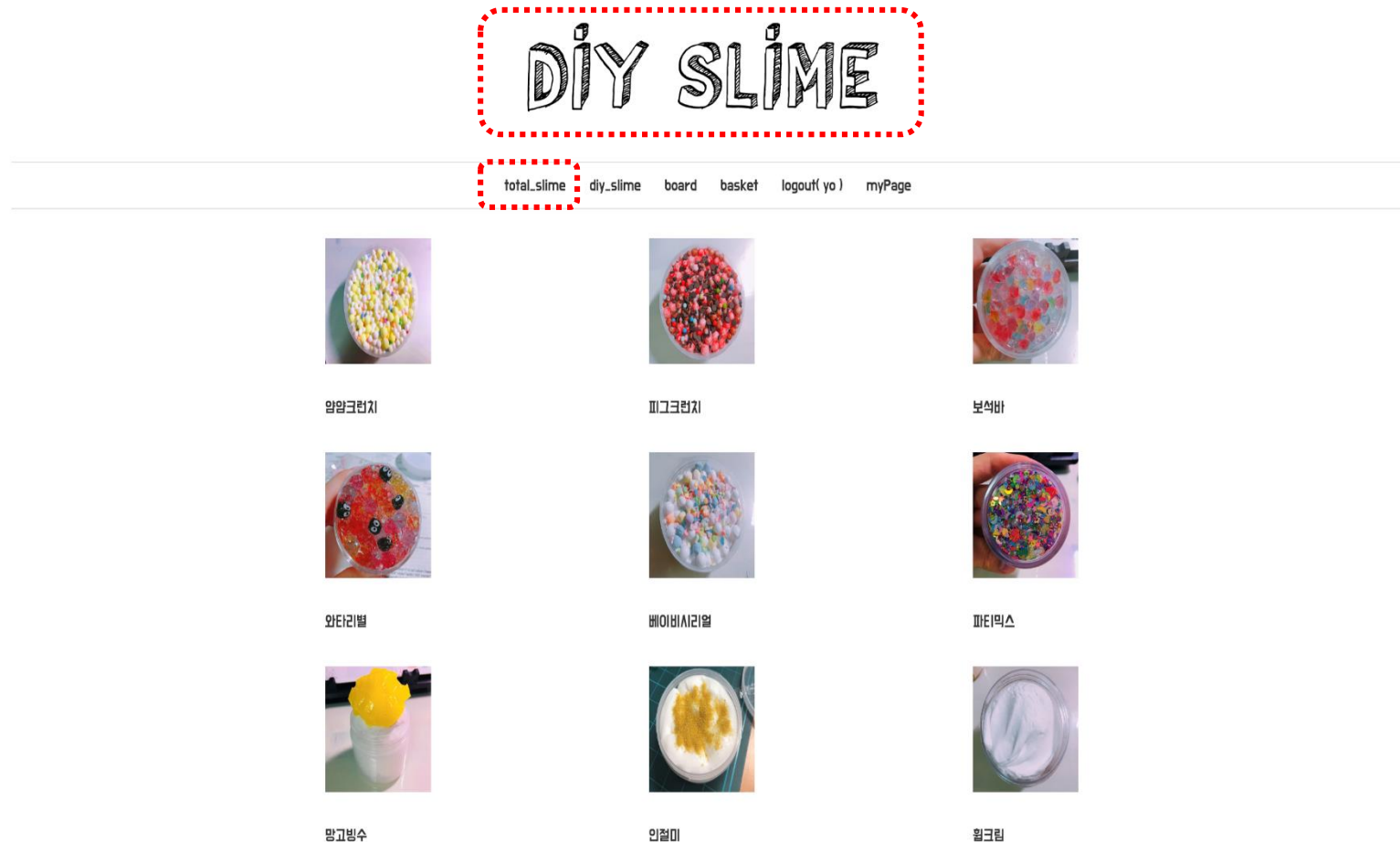
board	item	order	user
answerForm	Basket	Complete	List
boardDetail	Diy	orderDetail	loginForm
showBoard	itemDetail	orderForm	myPage
updateForm	showAll		registerForm
uploadForm			
	footer	header	

## Model



# 05 UI 설계

메인 화면



copyright© 2018 slime fairies all rights reserved

Category 의 Total\_slime과 메인 로고인 DIY SLIME클릭시 화면  
과 쇼핑몰 초기 화면인 main UI화면입니다

# 05 UI 설계

슬라임 상세 정보

## DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage



왕알크런치



피그크런치



보석바



와타리별



베어비시리얼



파티믹스



왕고빙수



인절미



왕크림



## DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage



피그크런치

판매가 8500

수량

배송방법 택배

배송비 3000원(40,000원 이상 구매 시 무료)

총금액 8500

상세정보 엘머스폴 베이스에 팔강,검정 폼볼이 들어간 돼지바향 폼크런치 슬라임

장바구니 취소

copy

Slime을 클릭시 slime 세부사항을 볼 수 있는 item\_detail 홈페이지로 넘어갑니다

# 05 UI 설계

## DIY SLIME

total\_slime **diy\_slime** board basket logout( yo ) myPage



알람크런치



피그크런치



보석바



와타리별



베이비시리얼



파티믹스



망고빙수



인절미



월크림

## DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage

Slime Color   
Slime Deco   
Slime Scent   
Slime Glue

이름   
판매가   
수량

배송방법

배송비 3000원(40,000원 이상 구매 시 무료)

총금액

카테고리의 diy\_slime 클릭시 사용자의 취향에 맞춰 슬라임을 제작할 수 있는  
DIY\_SLIME 홈페이지로 넘어갑니다.

# 05 UI 설계

## DIY SLIME

total\_slime diy\_slime board **basket** logout( yo ) myPage



피그크런치

판매가 8500  
 수량 1  
 배송방법 택배  
 배송비 3000원(40,000원 이상 구매 시 무료)  
 총금액 8500  
 상세정보 엘라스틱 베이스에 팹강,검정 풀볼이 들어간 돼지바탕 풍크런치 슬러임

장바구니 취소

copy



## DIY SLIME

total\_slime diy\_slime board **basket** logout( yo ) myPage

Slime Color black  
 Slime Deco biz  
 Slime Scent banana  
 Slime Glue amos  
 이름 diy slime  
 판매가 7800  
 수량 2  
 배송방법 택배  
 배송비 3000원(40,000원 이상 구매 시 무료)  
 총금액 15600

장바구니 취소

## DIY SLIME

total\_slime diy\_slime board **basket** logout( yo ) myPage

완제품 목록



삭제

피그크런치  
 수량 : 1  
 8500

합계 : 8500

DIY 목록

diy slime  
 Scent : banana  
 Deco : biz  
 Color : black  
 Glue : amos  
 수량 : 2  
 7800

합계 : 15600

주문하기

취소

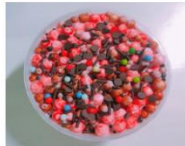
장바구니 버튼 또는 카테고리의 basket 클릭시 장바구니 홈페이지로 넘어갑니다.

## 05 UI 설계

DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage

완제품 목록



삭제

피그크런치

수량 : 1

8500

합계 : 8500

DIY 목록

diy slime

Scent : banana

Deco : biz

Color : black

Glue : amos

수량 : 2

7800

합계 : 15600

삭제

주문하기

취소

DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage

주문자 정보

성명

성연주

주소

나나나나

결제 정보

총금액

24100

원

배송비

3000

원

최종 결제 금액

27100

원

결제 수단

무통장 입금

입금 계좌

슬라임 요정 국민 12345-45678989

결제하기

취소

Basket 홈페이지의 주문하기 버튼 클릭시 주문을 작성할 수 있는 홈페이지로 넘어갑니다

# 05 UI 설계

주문 완료

## DIY SLIME

[total\\_slime](#) [diy\\_slime](#) [board](#) [basket](#) [logout\( peanut \)](#) [myPage](#)

### 주문자 정보

성명   
주소

### 결제 정보

총금액 50600원  
배송비 0원  
최종 결제 금액 50600원  
결제 수단 무통장 입금  
입금 계좌 슬라임 요정 국민 12345-45678989

결제하기

취소

## DIY SLIME

[total\\_slime](#) [diy\\_slime](#) [board](#) [basket](#) [logout\( yo \)](#) [myPage](#)

감사합니다. 주문이 정상적으로 완료되었습니다.

고객님의 주문번호는 22번입니다.

결제하기 버튼을 누르면 주문이 완료되어 주문 완료 홈페이지로 넘어가며 자신의 주문목록에 추가됩니다. 여기서 주문번호를 클릭하면 상세 내역을 볼 수 있습니다.

# 05 UI 설계

## DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage



망왕크렌치



파그크렌치



보석바



와타리별



베이비시리얼



파티믹스



망고빙수



인절미



퀵크림



## DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage

사용자 정보

이름 성연주

ID yo

Address LHLHLI

주문 내역

주문번호

[2018/12/26] 22

배송지 정보

LHLHLI

총금액

27100

카테고리의 mypage를 클릭시 자신의 개인정보와 주문내역을 확인할 수 있는 홈페이지로 넘어갑니다.



## 05 UI 설계

DIY SLIME

DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage

total\_slime diy\_slime board basket logout( yo ) myPage

감사합니다. 주문이 정상적으로 완료되었습니다.

고객님의 주문번호는 22번입니다.

DIY SLIME

total\_slime diy\_slime board basket logout( yo ) myPage

사용자 정보

이름 성연주

ID yo

Address LILILILIL

주문 내역

배송지 정보

LILILILIL

총금액

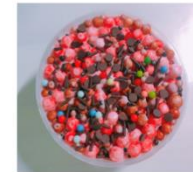
27100

주문번호

[2018/12/26] 22

## 주문 내역

완제품 주문 내역



피그크런치

수량 : 1

8500

합계 : 8500

## DIY 주문 내역

Scent : banana

Deco : biz

Color : black

Glue : amos

수량 : 2

7800

DIY SLIME

합계 : 15600

자신의 주문번호를 클릭시 주문한 슬라임의 목록을 볼 수 있는 홈페이지로 넘어갑니다.

# 05 UI 설계

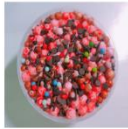
로그인

## DIY SLIME

total\_slime diy\_slime board basket login myPage



왕왕크런치



피그크런치



보석바



와타리젤



베이비시리얼



파티믹스



망고빙수



인절미



월크림



## DIY SLIME

total\_slime diy\_slime board basket login myPage

사용자 ID

비밀번호

로그인

회원가입

카테고리의 로그인을 클릭시 로그인을 할 수 있는 로그인 페이지로 넘어갑니다.

# 05 UI 설계

로그인

## DIY SLIME

total\_slime diy\_slime board basket login myPage

사용자 ID

비밀번호

로그인

회원가입



## DIY SLIME

total\_slime diy\_slime board basket **logout( admin )** myPage

사용자 ID	이름	주소
abv123	abv123	abv123
adf	hi	aaaaa
admin	admin	admin
hi	hi	hi
peanut	효경	서울시
yo	성연주	대전시

로그인 버튼을 누를 시 로그인이 되며 성공적으로 로그인 되었다면 사용자 목록을 보여주며 옆 화면과 같이 카테고리의 로그인 대신 로그아웃으로 표기된다.

# 05 UI 설계

회원가입

DIY SLIME

total\_slime diy\_slime board basket login myPage

사용자 ID

비밀번호

로그인

회원가입



DIY SLIME

total\_slime diy\_slime board basket login myPage

사용자 ID

비밀번호

비밀번호 확인

이름

주소

회원 가입

목록

로그인 버튼을 누를 시 로그인이 되며 성공적으로 로그인이 되었다면  
사용자 목록을 보여주며 옆 화면과 같이 카테고리의 로그인 대신 로그아웃으로 표기된다.

# 05 UI 설계

## DIY SLIME

total\_slime diy\_slime board basket login myPage

사용자 ID

비밀번호

비밀번호 확인

이름

주소

회원 가입 목록

## DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

사용자 ID	이름	주소
abv123	abv123	abv123
adf	hi	aaaaa
admin	admin	admin
hi	hi	hi
peanut	효경	서울시
yo	성연주	나나나나

회원가입 버튼을 누르면 회원가입이 성공적으로 됐을 시 사용자 목록을 보여주며, 목록 버튼을 누를 시 여태까지 가입한 회원들의 사용자 목록을 보여주며 옆 화면과 같이 카테고리의 로그인 대신 로그아웃으로 표기된다.

## 05 UI 설계

## DIY SLIME

total\_slime diy\_slime **board** basket logout( yo ) myPage

알왁런치



피그크런치



## DIY SLIME

total\_slime diy\_slime **board** basket logout( admin ) myPage

No	제목	ID	날짜
5	드디어 다 끝나가요...	admin	2018/12/24
4	문의 드립니다.	peanut	2018/12/24
3	오늘이 무슨날인가요?	hi	2018/12/24
1	피그크런치 문의	peanut	2018/12/23

질문등록

카테고리의 board 클릭시 게시판의 목록을 보여주는 홈페이지로 넘어간다.

## 05 UI 설계

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

No	제목	ID	날짜
5	드디어 다 끝나가요...	admin	2018/12/24
4	문의 드립니다.	peanut	2018/12/24
3	오늘이 무슨날인가요?	hi	2018/12/24
1	피그크림치 문의	peanut	2018/12/23

질문등록

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

제목

배송문의) 언제쯤 배송되나요?

내용

저번 주 수요일에 주문했는데 아직 주문 완료라고만 떠서요 언제쯤 배송되나요?

질문등록

목록

질문등록 버튼 클릭시 질문을 작성하고 등록할 수 있는 홈페이지로 넘어갑니다.

## 05 UI 설계

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

No	제목	ID	날짜
23	배송문의) 언제쯤 배송되나요?	admin	2018/12/26
5	드디어 다 끝나가요...	admin	2018/12/24
4	문의 드립니다.	peanut	2018/12/24
3	오늘이 무슨날인가요?	hi	2018/12/24
1	피그크런치 문의	peanut	2018/12/23

질문등록

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

제목	배송문의) 언제쯤 배송되나요?
등록 날짜	2018/12/26
질문	저번 주 목요일에 주문했는데 아직 주문 완료라고만 떠서요 언제쯤 배송되 나요?

답변 판매자가 아직 답변을 등록하지 않았습니다.

수정 삭제 답변 목록

등록한 질문을 클릭시 등록 질문의 상세정보를 볼 수 있는 홈페이지로 넘어갑니다.



## 05 UI 설계

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

제목 배송문의) 언제쯤 배송되나요?

등록 날짜 2018/12/26

질문 저번 주 목요일에 주문했는데 아직 주문 완료라고만 떠서요 언제쯤 배송되 나요?

답변 판매자가 아직 답변을 등록하지 않았습니다.

수정 삭제 답변 목록

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

제목

배송문의) 언제쯤 배송되나요?

등록 날짜

2018/12/26

내용

저번 주 목요일에 주문했는데 아직 주문 완료라고만 떠서요 언제쯤 배송되 나요?

수정

목록

수정 버튼 클릭시 질문을 수정할 수 있는 홈페이지로 넘어갑니다.

## 05 UI 설계

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

제목 배송문의) 언제쯤 배송되나요?

등록 날짜 2018/12/26

질문 저번 주 목요일에 주문했는데 아직 주문 완료라고만 떠서요 언제쯤 배송되 나요?

답변 판매자가 아직 답변을 등록하지 않았습니다.

수정 삭제 **답변** 목록

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

고객님 죄송합니다... 다음 주 월요일에 재 입고 예정이라서 주문하신 순서대로 차근차근 배송해드리겠습니다!!

내용

답변 등록

목록

답변 버튼 클릭시 답변을 등록할 수 있는 홈페이지로 넘어갑니다.

## 05 UI 설계

DIY SLIME

total\_slime diy\_slime board basket logout( admin ) myPage

제목	배송문의) 언제쯤 배송되나요?
등록 날짜	2018/12/26
질문	저번 주 목요일에 주문했는데 아직 주문 완료라고만 떠서요 언제쯤 배송되 나요?
답변	고객님 죄송합니다... 다음 주 월요일에 재입고 예정이라서 주문하신 순서대로 차근차근 배송해드리겠습니다!

수정 삭제 답변 목록

DIY SLIME

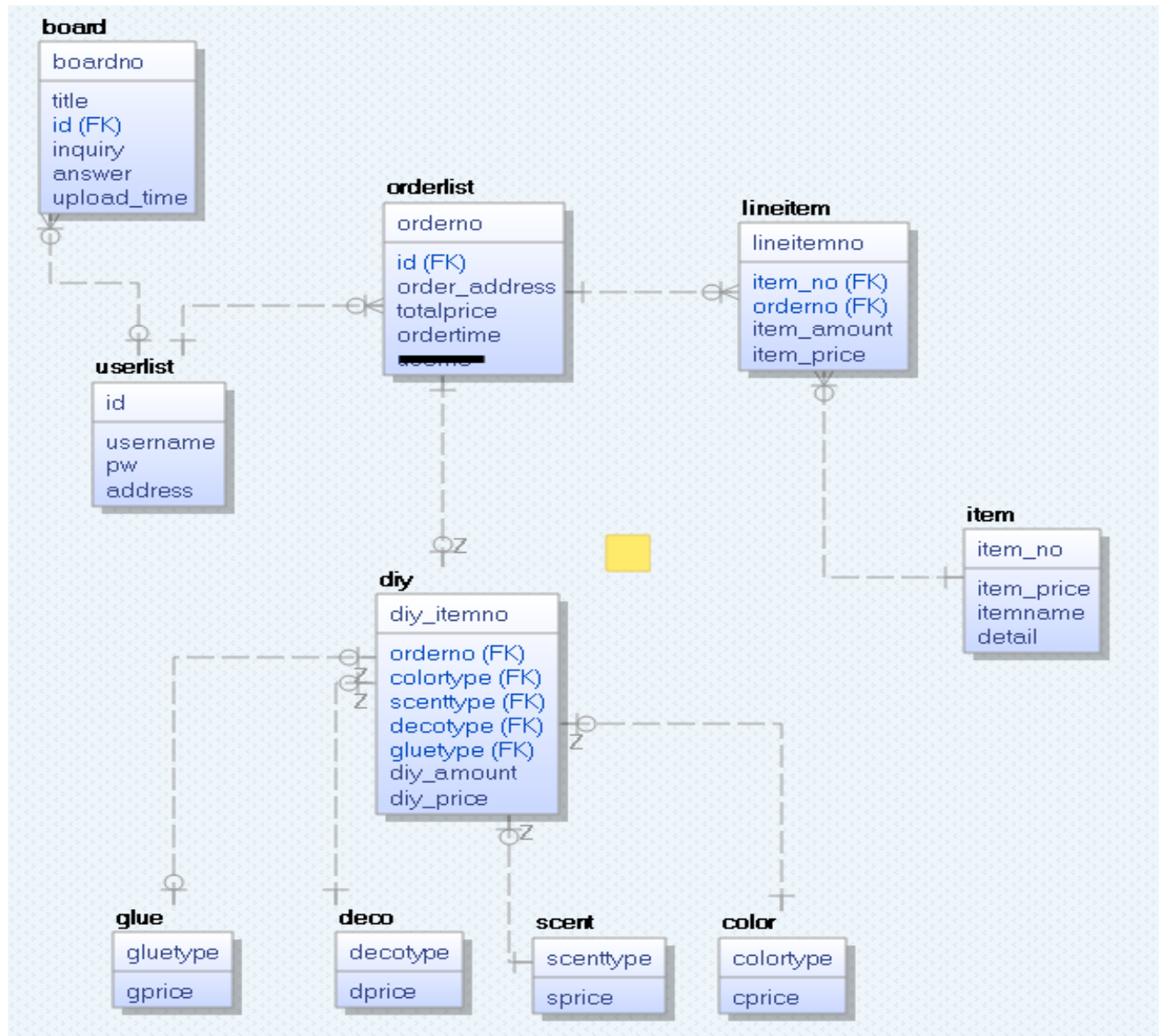
total\_slime diy\_slime board basket logout( admin ) myPage

No	제목	ID	날짜
5	드디어 다 끝나가요...	admin	2018/12/24
4	문의 드립니다.	peanut	2018/12/24
3	오늘이 무슨날인가요?	hi	2018/12/24
1	피그크런치 문의	peanut	2018/12/23

질문등록

삭제 버튼 클릭시 해당하는 게시판이 삭제됩니다

# 06 데이터베이스 설계



# 06 데이터베이스 설계

## 1) BOARD - N : 1 - USERLIST

-> 한 게시판에는 한 사용자만 등록할 수 있고 한 사용자는 여러 개의 게시판을 가질 수 있고, 게시판을 가지고 있지 않을 수도 있다.

## 2) USERLIST - 1 : N - ORDERLIST

-> 한 사용자는 여러 개의 주문 내역을 가질 수 있고 또는 주문 내역을 가지고 있지 않을 수도 있고, 한 주문 내역은 한 사용자를 가지고 있어야 한다.

## 3) ORDERLIST - 1 : N - DIY

-> 한 주문내역에는 여러 개의 diy를 가질 수 있거나 아예 가지고 있지 않을 수 있으며 한 diy는 하나의 주문내역을 가진다.

## 4) ORDERLIST - 1 : N - LINEITEM

-> 한 주문내역에는 여러 개의 lineitem을 가질 수 있거나 아예 가지고 있지 않을 수도 있으며, 한 lineitem은 하나의 주문내역을 가진다.

## 5) LINEITEM - N : 1 - ITEM

-> 한 lineitem은 한 item을 가지고, 한 item은 여러 개의 lineitem에 포함될 수 있거나 아예 포함되어 있지 않을 수 있다.

# 06 데이터 베이스 설계

## 6) DIY - N : 1 - GLUE

-> 한 DIY는 한개의 풀을 선택하거나 선택하지 않을 수 있고, GLUE는 여러 개의 DIY에 선택될 수 있거나 하나도 포함되지 않을 수 있다.

## 7) DIY - N : 1 - DECO

-> 한 DIY는 한개의 장식을 선택하거나 선택하지 않을 수 있고, DECO는 여러 개의 DIY에 선택될 수 있거나 하나도 포함되지 않을 수 있다.

## 8) DIY - N : 1 - SCENT

-> 한 DIY는 한개의 향을 선택하거나 선택하지 않을 수 있고, SCENT는 여러 개의 DIY에 선택될 수 있거나 하나도 포함되지 않을 수 있다.

## 9) DIY - N : 1 - COLOR

-> 한 DIY는 한개의 색깔을 선택하거나 선택하지 않을 수 있고, COLOR는 여러 개의 DIY에 선택될 수 있거나 하나도 포함되지 않을 수 있다.

# 06 데이터베이스 설계

board

boardno(PK)
title
id(FK)
inquiry
answer
upload_time

BOARD TABLE : 게시판 DB TABLE

- 1) boardno(integer) : 게시판 번호로, primary key이다.
- 2) title(string) : 게시판 제목이다.
- 3) id(string): user id로 어떠한 사용자가 게시판을 작성했는지 알아야 함으로 userlist table과 연결되어 있는 FK이다.
- 4) inquiry : 게시판 내용이다.
- 5) answer : 관리자가 답변을 작성할 때 필요한 필드다.
- 6) upload\_time : 게시판이 작성된 시간이다.

# 06 데이터베이스 설계

orderlist

orderno(PK)
id(FK)
order_address
totalprice
ordertime

ORDERLIST TABLE : 주문내역 DB TABLE

- 1) orderno(PK) : 주문번호 필드로 이 테이블에서 고유한 값을 가짐으로 PK이다.
- 2) id(FK) : 어떠한 사용자가 주문했는지 알아야 함으로 userlist와 연결되어 있는 FK이다.
- 3) order\_address : 상품이 발송될 주소이다.
- 4) totalprice : 한 주문내역 당 주문한 상품의 총가격이다.
- 5) ordertime : 주문한 시각이다.



# 06 데이터베이스 설계

lineitem

Lineitemno(PK)
Item_no(FK)
orderno(FK)
item_amount
item_price

LINEITEM TABLE : 주문 내역의 상세 품목 DB TABLE

- 1) orderno(PK) : 상세 품목 번호 필드로 이 테이블에서 고유한 값을 가짐으로 PK이다.
- 2) item\_no(FK) : 어떠한 상품 목록인지 알아야 함으로 item과 연결되어 있는 FK이다.
- 3) orderno : 어떠한 주문번호의 상품목록인지 알아야 함으로 orderlist와 연결되어 있는 FK이다.
- 4) item\_amount : 상품이 주문된 개수이다.
- 5) item\_price : 상품 목록의 금액이다.

# 06 데이터베이스 설계

diy

diy_itemno(PK)
orderno(FK)
colortype(FK)
scenttype(FK)
decotype(FK)
gluetype(FK)
diy_amount
diy_price

DIY TABLE : diy\_slime DB TABLE

- 1) diy\_itemno(PK) : diy\_slimed의 번호로 이 테이블에서 고유한 값을 가짐으로 PK이다.
- 2) orderno(FK) : 어떠한 주문내역과 관련되었는지 알아야 함으로 orderlist와 연결되어 있는 FK이다.
- 3) colortype (FK) : 어떠한 색상을 가진 diy\_slime 인지 알아야함으로 color와 연결되어 있는 FK이다.
- 4) scenttype (FK) : 어떠한 향을 가진 diy\_slime 인지 알아야함으로 scent와 연결되어 있는 FK이다.
- 5) decotype (FK) : 어떠한 장식을 가진 diy\_slime 인지 알아야함으로 deco와 연결되어 있는 FK이다.
- 6) gluetype (FK) : 어떠한 풀을 가진 diy\_slime 인지 알아야함으로 deco와 연결되어 있는 FK이다.
- 7) diy\_amount : diy\_slime을 주문한 총 개수이다.
- 8) diy\_price : diy\_slime의 가격이다.

# 06 데이터베이스 설계

item

item_no(PK)
item_price
itemname
detail

ITEM TABLE : 완제품 슬라임 DB TABLE

- 1) item\_no(PK) : 완제품 슬라임의 번호로 이 테이블에서 고유한 값을 가짐으로 PK이다.
- 2) item\_price : 완제품 슬라임의 가격이다.
- 3) itemname : 완제품 슬라임의 상품명이다.
- 4) detail: 완제품 슬라임의 상세 정보이다.

# 06 데이터베이스 설계

userlist

id(PK)
username
pw
address

USERLIST TABLE : 사용자 목록 DB TABLE

- 1) id(PK) : 사용자가 가입한 사용자 id 정보이다. 이 필드는 중복된 값을 가질 수 없고, 고유한 값을 가짐으로 PK이다.
- 2) username : 사용자 성명이다.
- 3) pw : 사용자가 가입한 비밀번호 정보이다.
- 4) adress: 사용자가 입력한 주소이다.

# 06 데이터 베이스 설계

glue

gluetype(PK)
gprice

GLUE TABLE : 슬라임 풀 종류 DB TABLE

- 1) gluetype(PK) : 슬라임 풀 재료의 종류이다. 이 테이블에서 고유한 값을 가짐으로 PK 이다.
- 2) gprice: 슬라임 재료 풀의 가격이다.

deco

decotype(PK)
dprcie

DECO TABLE : 슬라임 장식 종류 DB TABLE

- 1) decotype(PK) : 슬라임 장식 재료의 종류이다. 이 테이블에서 고유한 값을 가짐으로 PK 이다.
- 2) dprice: 슬라임 장식 재료의 가격이다.

# 06 데이터베이스 설계

color

colortype(PK)
cprice

COLOR TABLE : 슬라임 색상 종류 DB TABLE

- 1) colortype(PK) : 슬라임 색상 종류이다. 이 테이블에서 고유한 값을 가짐으로 PK 이다.
- 2) cprice: 슬라임 색상 종류의 가격이다.

scent

scenttype(PK)
sprice

SCENT TABLE : 슬라임 향 종류 DB TABLE

- 1) scenttype(PK) : 슬라임 향 종류이다. 이 테이블에서 고유한 값을 가짐으로 PK 이다.
- 2) sprice: 슬라임 향 재료의 가격이다.

# 07 클래스 설계

DTO

**Board**

Board : 게시판 DB

**Item**

Item : 완제품 DB

**DIY**

Color : DIY 제조 선택 사항 中 색 DB  
Deco : DIY 제조 선택 사항 中 데코 DB  
DIY : DIY DB  
Glue : DIY 제조 선택 사항 中 풀 DB  
Scent : DIY 제조 선택 사항 中 향 DB

**order**

Order : DIY&완제품 통합 최종 주문 DB  
Lineitem : 완제품 주문 DB

**user**

User : 사용자 DB

# 07 클래스 설계

**Board**

BoardManager : 게시판 관련 메소드 정의

**Item**

ItemManager : 완제품 관련 메소드 정의

**DIY**

DIYManager : DIY 관련 메소드 정의

**order**

LineItemManager : 완제품 주문 관련 메소드 정의  
OrderManager : 주문서 내용 관련 메소드 정의

**user**

UserManager : 사용자 관련 메소드 정의

**etc**

ExistingUserException : 사용자 존재 확인(회원가입 시)  
PasswordMismatchException : 패스워드 일치 확인  
UserNotFoundException : 사용자 유무 확인(로그인 시)



# 07 클래스 설계

## Board

boardList(): DB에 저장된 board 목록들을 불러옴  
showDetail(boardNo): boardno로 해당 게시글의 상세 내용을 불러옴  
upload(Board, userId): 게시글 등록  
update(Board, userId): 게시글 수정  
answer(board, boardno): 관리자 게시글 답변  
remove(boardno): 게시글 삭제

## Item

itemList(): DB에 저장된 item 목록들을 불러옴  
findSlimeName(item\_no): item\_no로 해당하는 슬라임 이름을 불러옴  
showDetail(itemName): itemName으로 해당하는 슬라임 정보를 불러옴

## DIY

create(diy): 받아온 diy정보를 DB에 저장함  
findScent(): DB에 저장된 DIY 향 정보를 가져옴  
findColor(): DB에 저장된 DIY 색상 정보를 가져옴  
findDeco(): DB에 저장된 DIY 데코 정보를 가져옴  
findGlue(): DB에 저장된 DIY 글루 정보를 가져옴  
findDIYList(orderno): 해당 번호에 해당하는 DIY 주문 정보를 가져옴

# 07 클래스 설계

## order

create(order): 받아온 주문 정보를 생성  
update(orderno): 해당 주문번호에 해당하는 주소 정보를 수정  
findOrder(orderno): 해당 주문번호에 해당하는 주문을 불러옴  
findOrderList(id): 해당하는 사용자가 주문한 list를 불러옴

## user

create(user): 받아온 user 정보를 DB에 등록함 (회원가입)  
findUser(userId): 해당하는 사용자의 사용자정보를 불러옴  
findUserList(): DB에 저장된 user들을 불러옴  
isExisting(userId): 해당하는 id를 가진 사용자가 존재하는지 검색

## lineItem

create(lineItem): 받아온 lineitem 정보를 DB에 등록(주문)  
findLineItemList(orderno): 주문번호에 해당하는 DB에 저장된 주문 목록을 불러옴

# 07 클래스 설계

## Board

boardList() : DB에 저장된 board 목록들을 불러옴  
showDetail(boardNo) : boardno로 해당 게시글의 상세 내용을 불러옴

## Item

ItemDAO : Item DB 접근

## DIY

DIYDAO : DIY DB 접근

## order

OrderDAO : Order DB 접근  
LineItemDAO : LineItem DB 접근

## user

UserDAO : User DB 접근

## etc

ConnectionManager  
JDBCUtil

# 07 클래스 설계

## Board

AnswerController : 게시글을 답변할 수 있도록 연결  
RemoveController : 게시글을 삭제할 수 있도록 연결  
ShowBoardController : 게시판 목록으로 연결  
ShowBoardDetailController : 게시판 자세히 보기로 연결  
UploadController : 게시글을 수정할 수 있도록 연결

## Item

ShowAllController : 완제품 전체보기로 연결  
ShowDetailController : 완제품 상세보기 및 주문으로 연결

## DIY

DIYController : DIY 주문으로 연결

# 07 클래스 설계

## Order

CompleteOrderController : 결제 완료로 연결  
DIYBasketController : DIY 장바구니 목록으로 연결  
ItemBasketController : 완제품 장바구니 목록으로 연결  
OrderController : 주문자 정보 및 결제 페이지로 연결  
OrderListDetailController : 특정 주문서 상세보기로 연결  
RemoveDIYBasketController : 장바구니 목록에서 DIY 품목 삭제 할 수 있도록 연결  
RemoveItemBasketController : 장바구니 목록에서 완제품 품목 삭제 할 수 있도록 연결

## user

LoginController : 로그인 할 수 있도록 연결  
LogoutController : 로그아웃 할 수 있도록 연결  
MyPageController : 마이페이지로 연결  
RegisterUserController : 회원가입으로 연결  
UserListController : 사용자 목록으로 연결  
UserSessionUtils : 세션에 있는 유저 목록 확인 연결

# 07 클래스 설계

## Board

answerForm : 답변 등록  
boardDetail : 질문 자세히 보기  
showBoard : 전체 질문 목록  
updateForm : 글 수정  
uploadForm : 질문 등록

## Item

Basket : 장바구니  
itemDetail : 완제품 자세히 보기 및 완제품 주문  
showAll : 완제품 전체 보기

## Item

Diy : color/scent/glue/deco를 선택해 DIY 슬라이م 주문

## order

Complete : 상품 주문 완료  
orderDetail : 특정 주문 내역 자세히 확인  
orderForm : 주문 정보 확인 및 결제

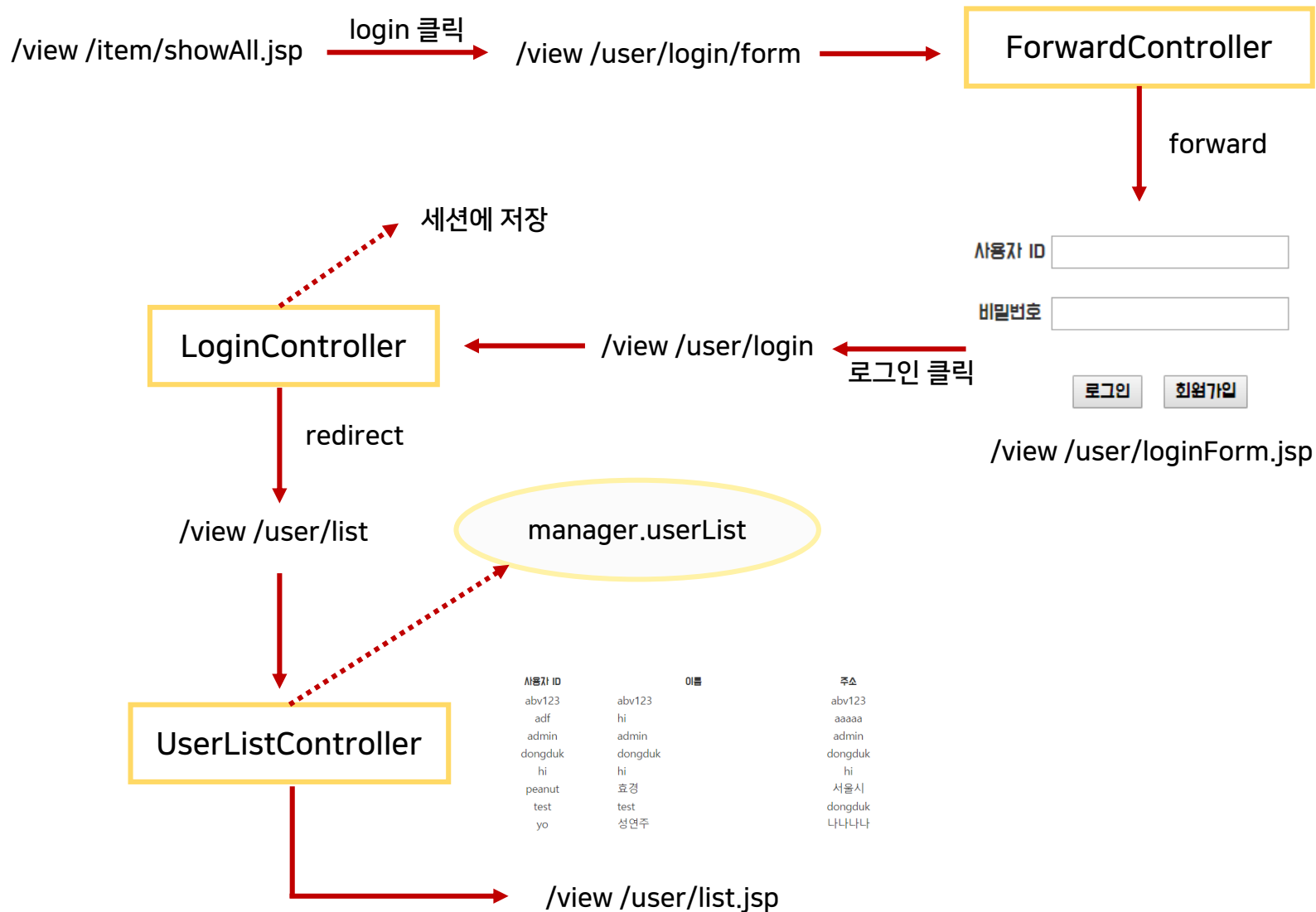
## user

List : 사용자 목록  
loginForm : 로그인  
myPage : 사용자 정보 및 주문 내역 확인  
registerForm : 회원 가입

# 08 Request URI

```
public void initMapping() {
    mappings.put("/", new ForwardController("index.jsp"));
    mappings.put("/view/item/list", new ShowAllController());
    mappings.put("/view/item/detail", new ShowDetailController());
    mappings.put("/view/user/login", new LoginController());
    mappings.put("/view/user/login/form", new ForwardController("/view/user/loginForm.jsp"));
    mappings.put("/user/logout", new LogoutController());
    mappings.put("/view/user/list", new UserListController());
    mappings.put("/view/user/register", new RegisterUserController());
    mappings.put("/view/user/register/form", new ForwardController("/view/user/registerForm.jsp"));
    mappings.put("/view/user/myPage", new MyPageController());
    mappings.put("/view/item/showAll", new ShowAllController());
    mappings.put("/view/item/basket", new ForwardController("/view/item/basket.jsp"));
    mappings.put("/view/item/itemBasket", new ItemBasketController());
    mappings.put("/view/item/diyBasket", new DIYBasketController());
    mappings.put("/view/item/removeItemBasket", new RemoveItemBasketController());
    mappings.put("/view/item/removeDiyBasket", new RemoveDIYBasketController());
    mappings.put("/view/item/diy", new DIYController());
    mappings.put("/view/order/form", new OrderController());
    mappings.put("/view/order/complete", new CompleteOrderController());
    mappings.put("/view/order/detail", new OrderListDetailController());
    mappings.put("/view/board/list", new ShowBoardController());
    mappings.put("/view/board/upload/form", new ForwardController("/view/board/uploadForm.jsp"));
    mappings.put("/view/board/upload", new UploadController());
    mappings.put("/view/board/detail", new ShowBoardDetailController());
    mappings.put("/view/board/remove", new RemoveController());
    mappings.put("/view/board/update/form", new UpdateBoardController());
    mappings.put("/view/board/update", new UpdateBoardController());
    mappings.put("/view/board/answer/form", new AnswerController());
    mappings.put("/view/board/answer", new AnswerController());
}
```

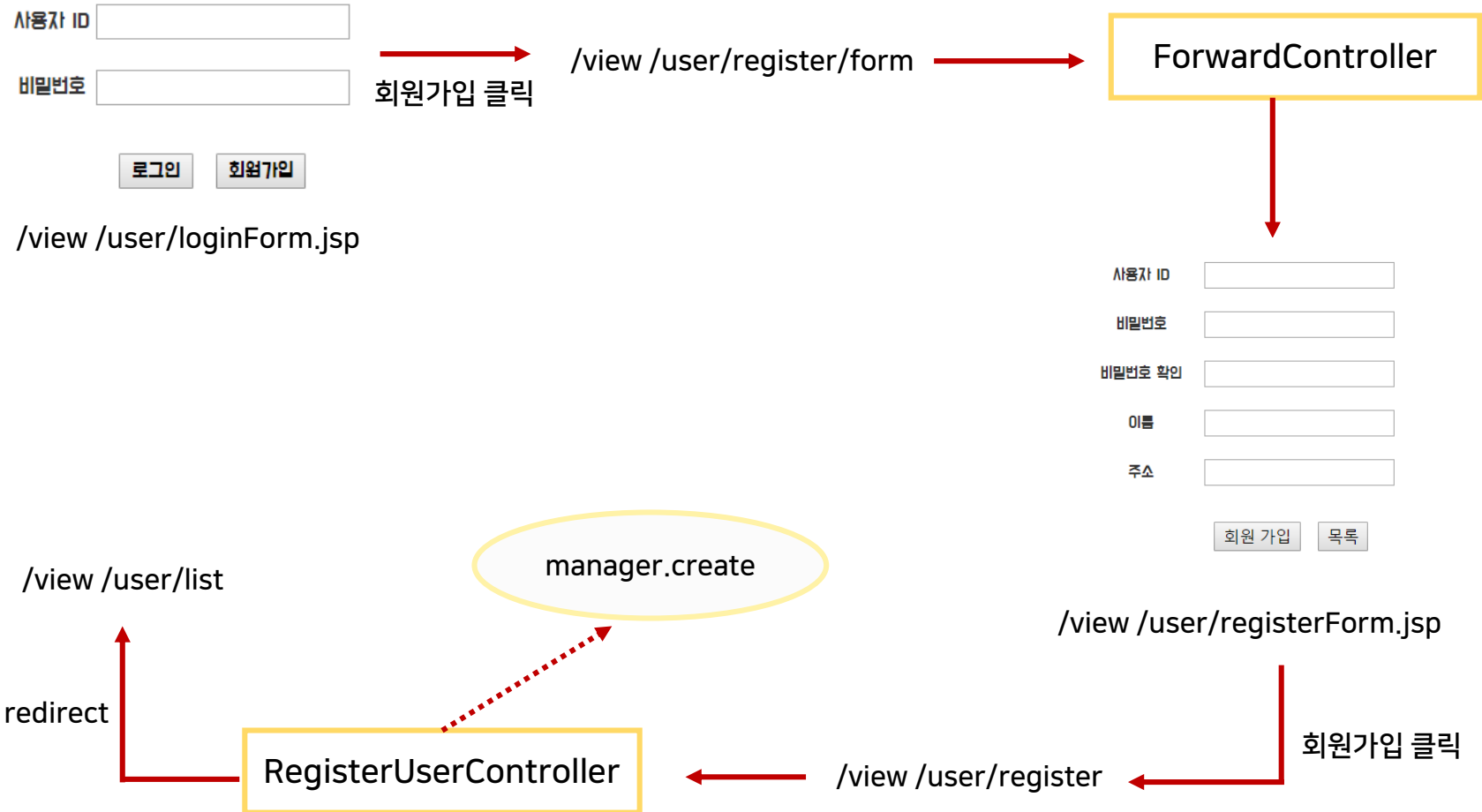
# 08 Request URI



\* 편의상 시작지점을 `/view/item/ShowAll.jsp`로 하지만 헤더에 메뉴가 있기 때문에 어느 페이지에서나 접근 가능합니다.

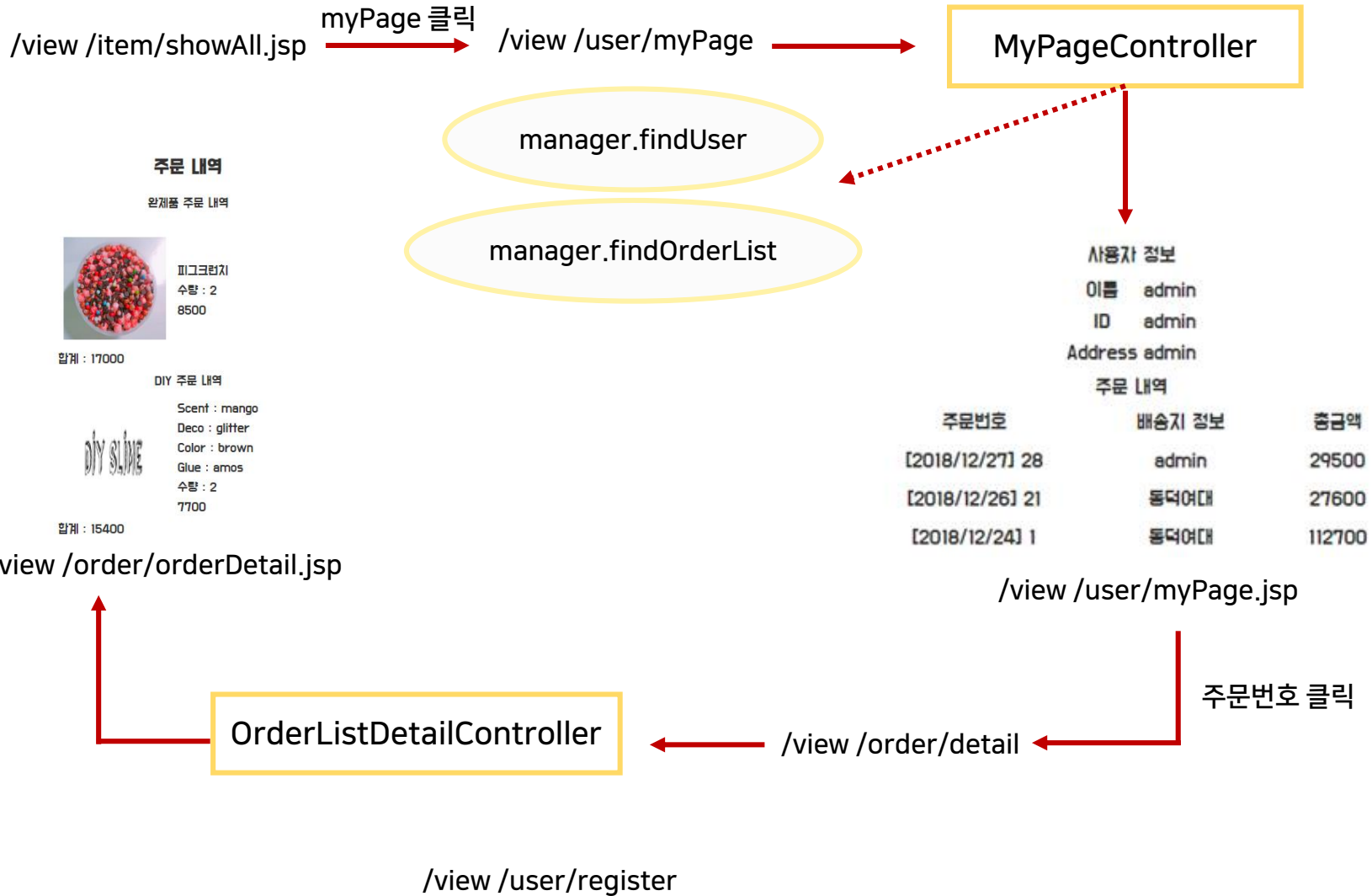


# 08 Request URI



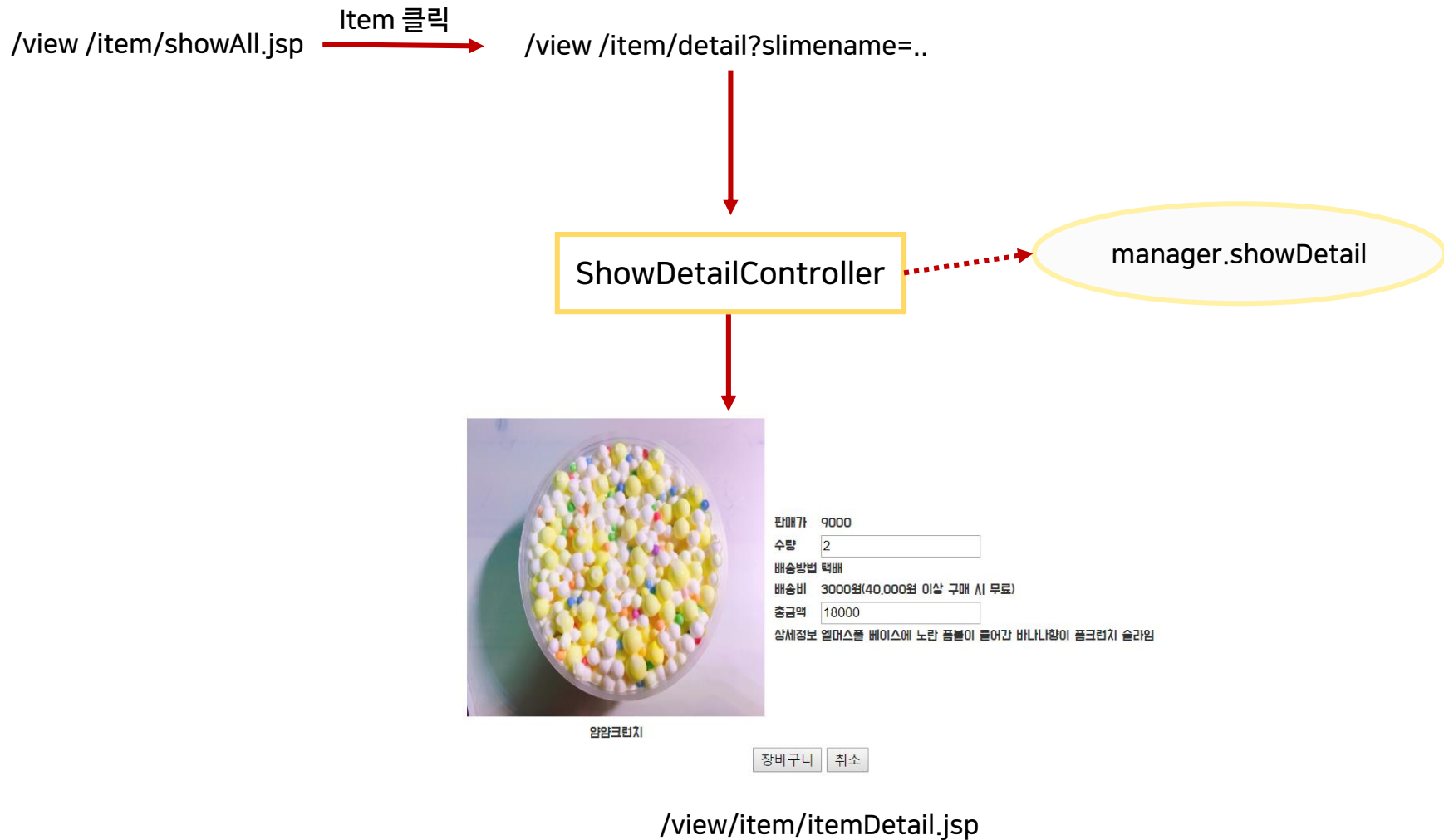
\* 편의상 시작지점을 `/view/item/ShowAll.jsp`로 하지만 헤더에 메뉴가 있기 때문에 어느 페이지에서나 접근 가능합니다.

# 08 Request URI

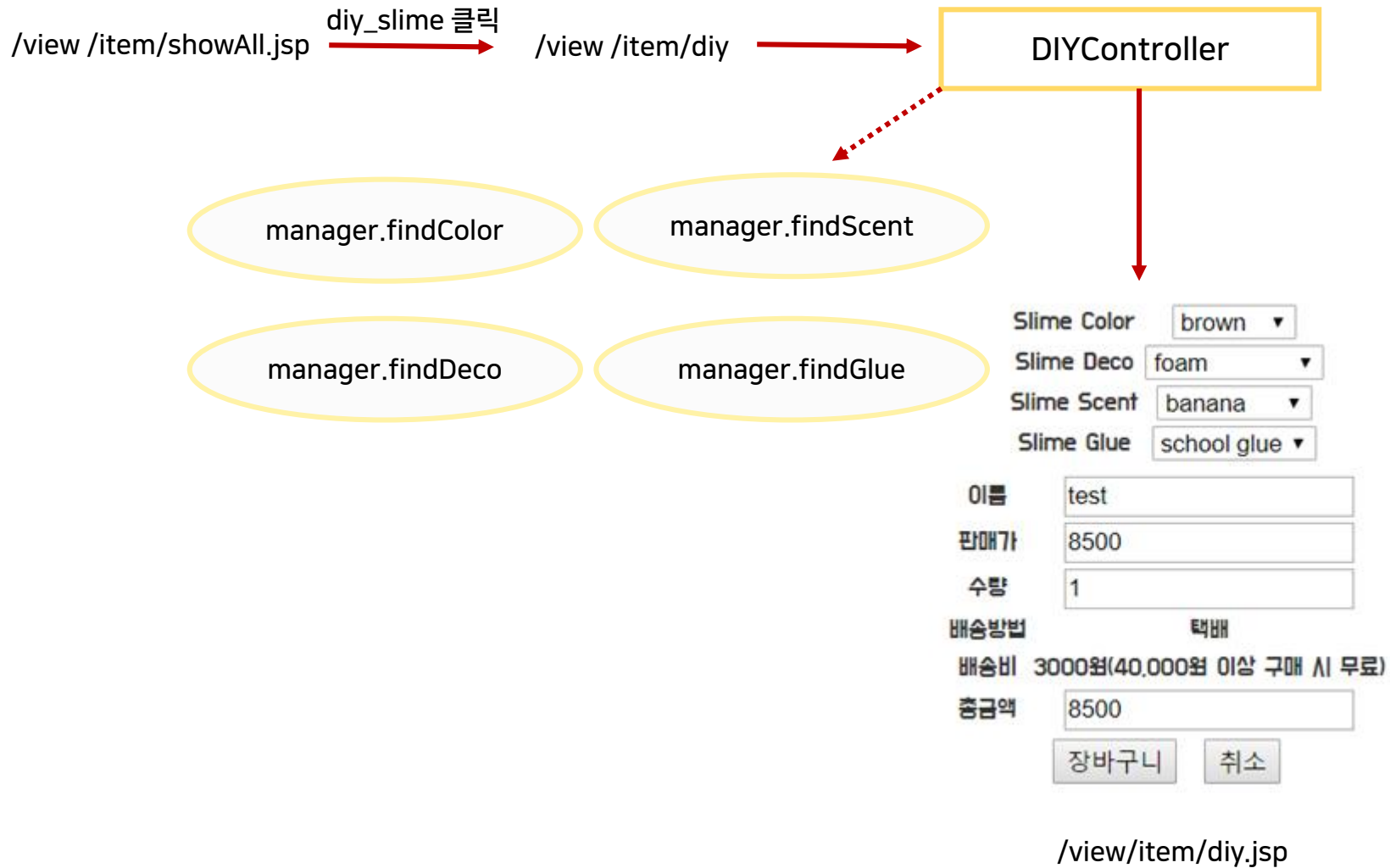


\* 편의상 시작지점을 `/view/item/ShowAll.jsp`로 하지만 헤더에 메뉴가 있기 때문에 어느 페이지에서나 접근 가능합니다.

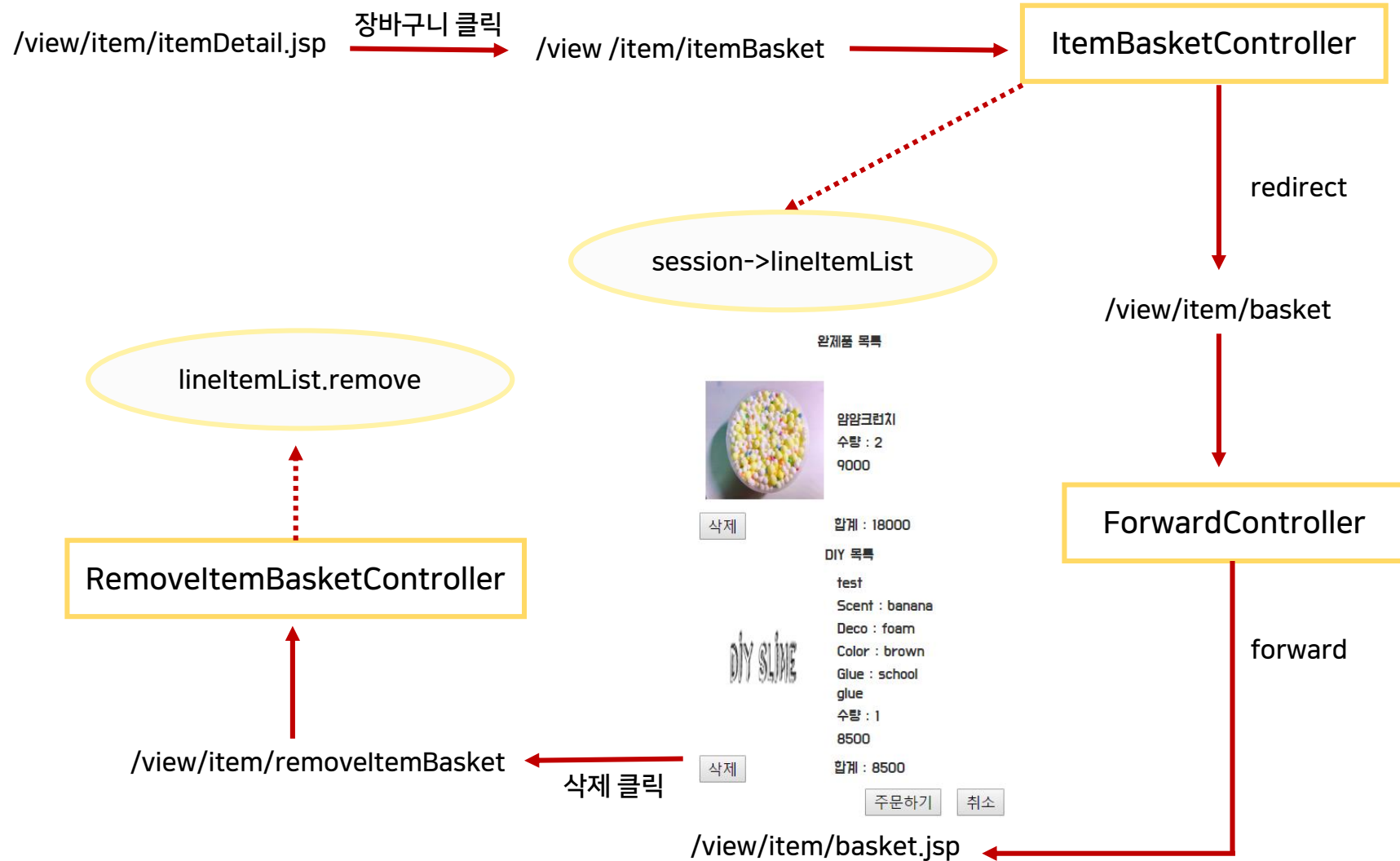
# 08 Request URI



# 08 Request URI

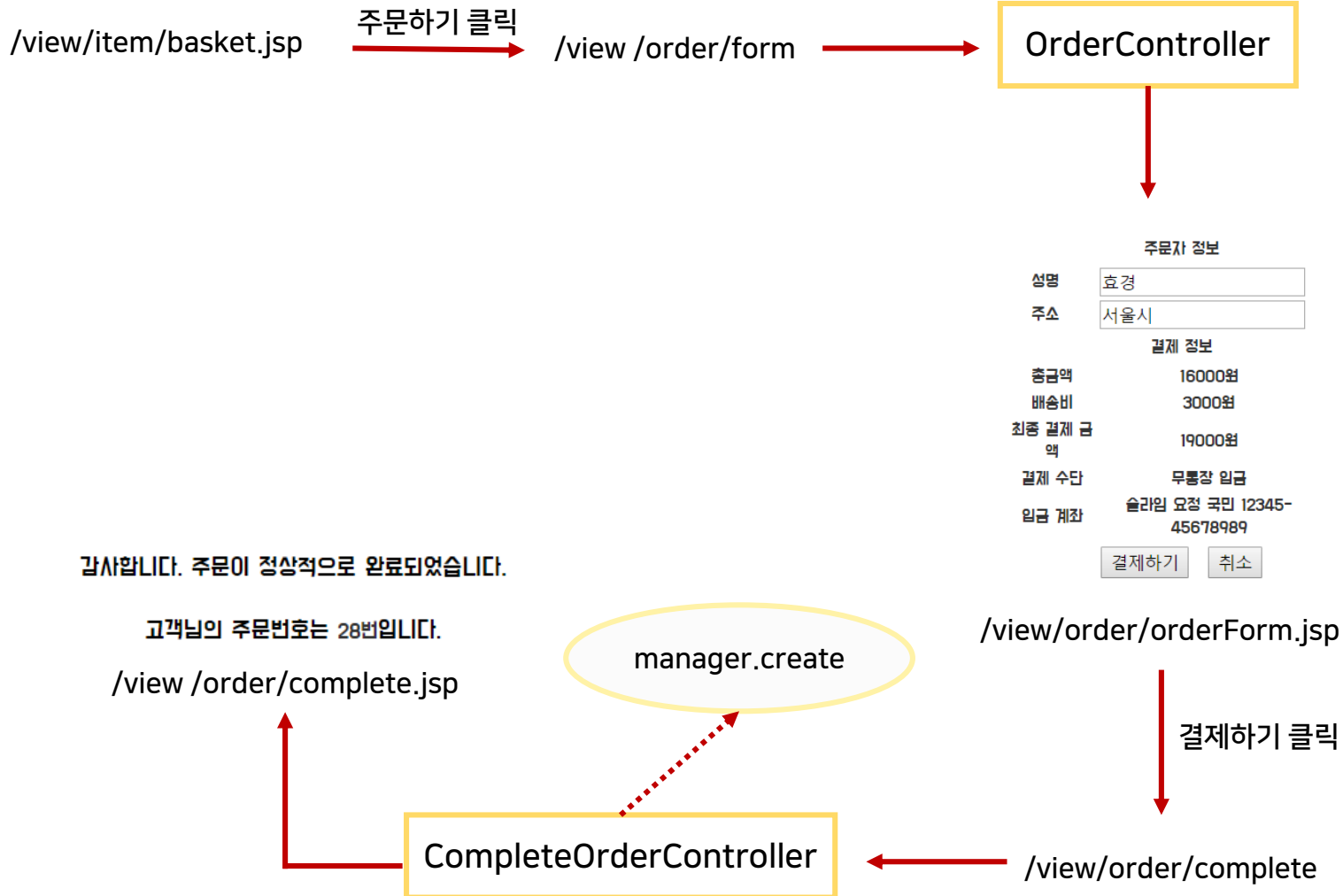


# 08 Request URI

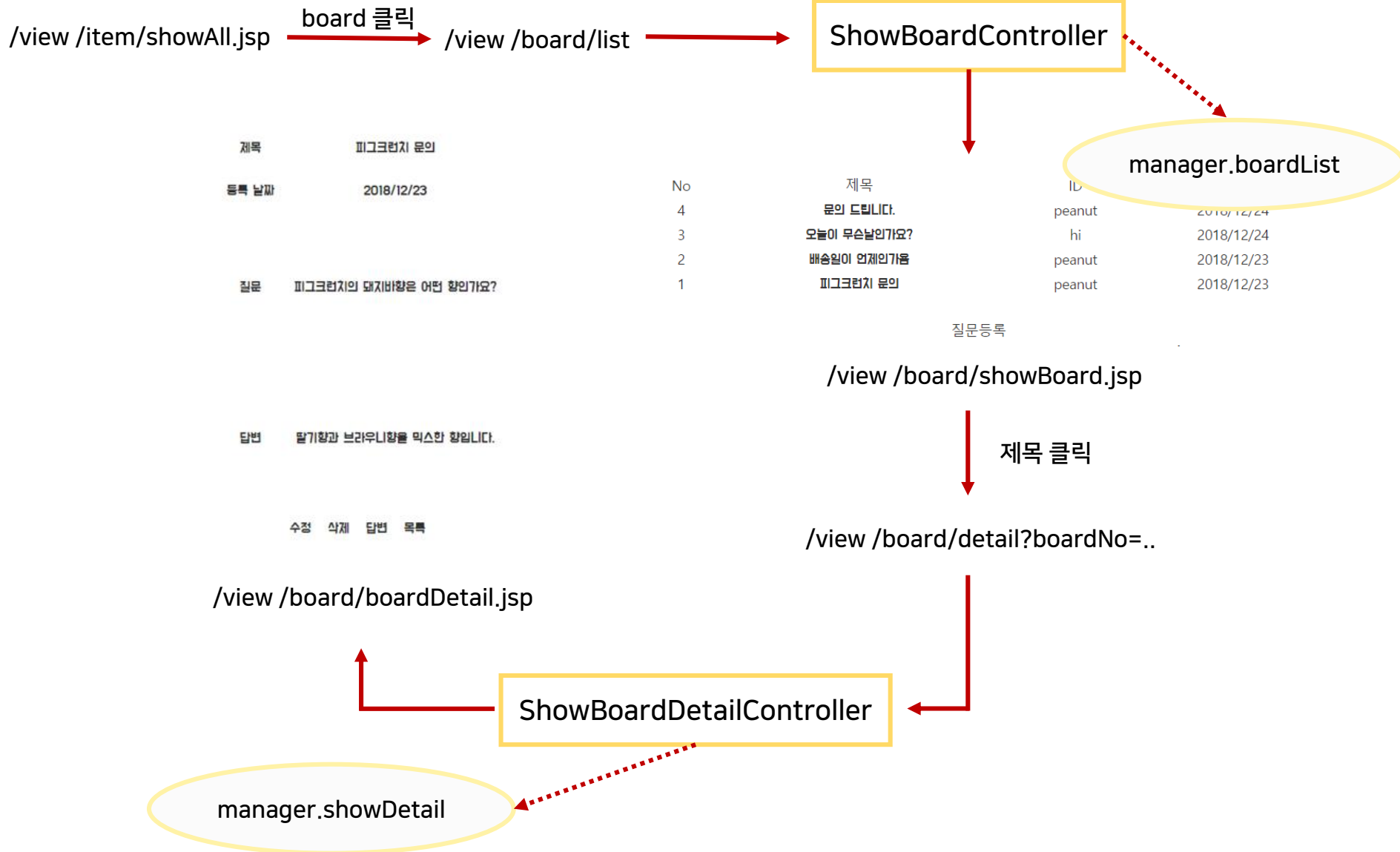


\* Diy가 장바구니에 저장되는 과정은 Controller와 url만 다를뿐 동일하여 생략합니다.

# 08 Request URI

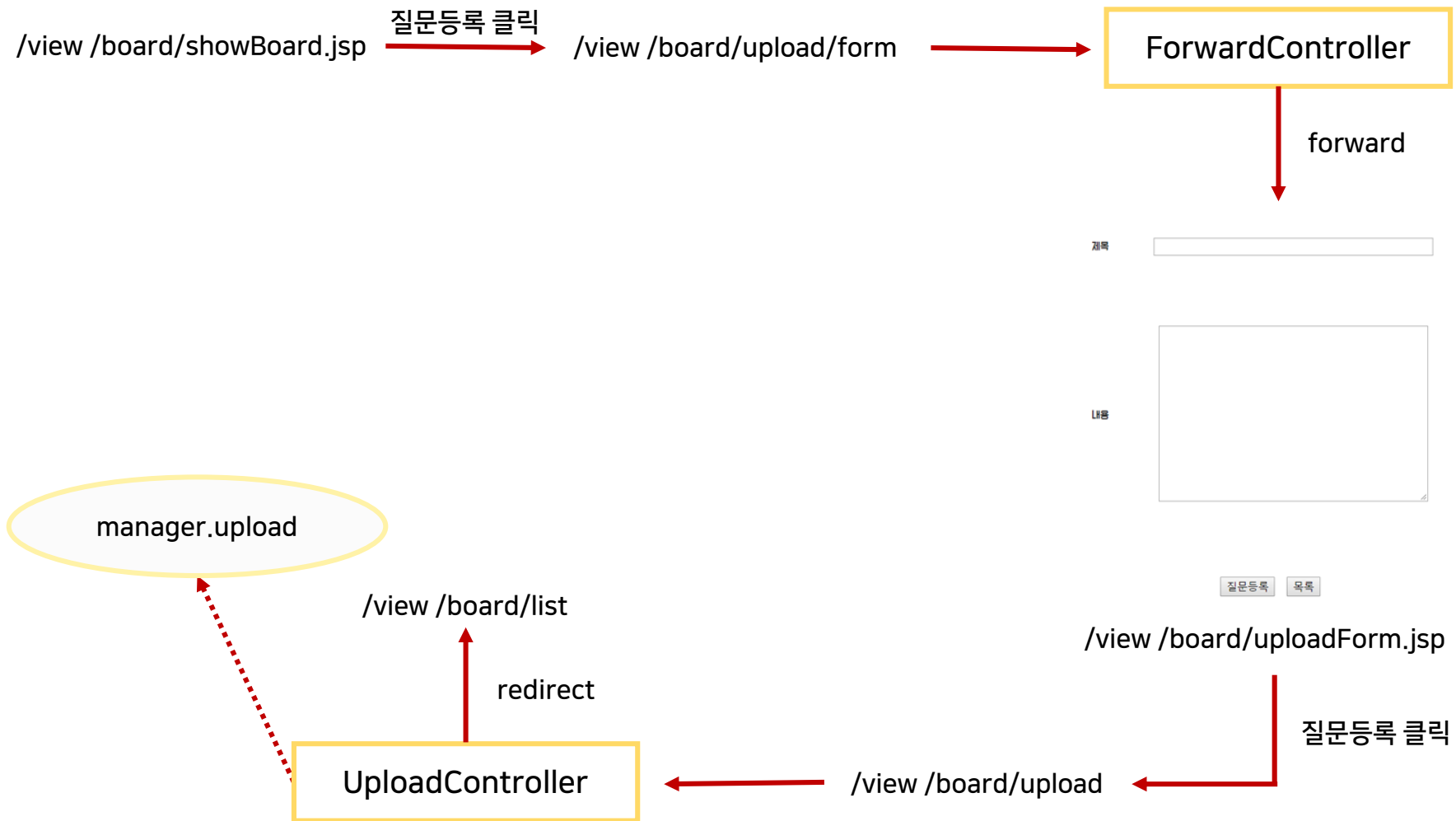


# 08 Request URI



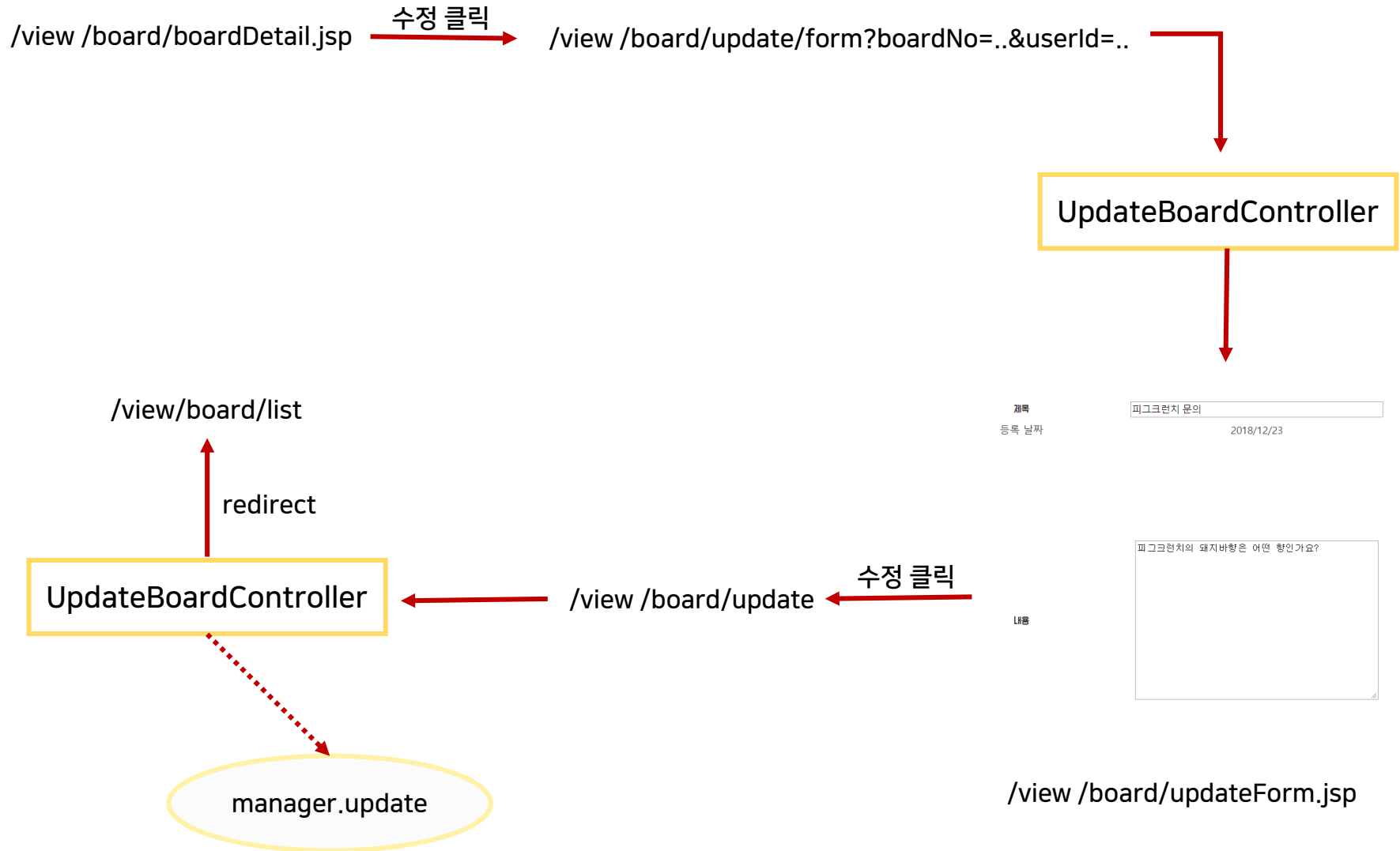
\* 편의상 시작지점을 /view/item/ShowAll.jsp로 하지만 헤더에 메뉴가 있기 때문에 어느 페이지에서나 접근 가능합니다.

# 08 Request URI

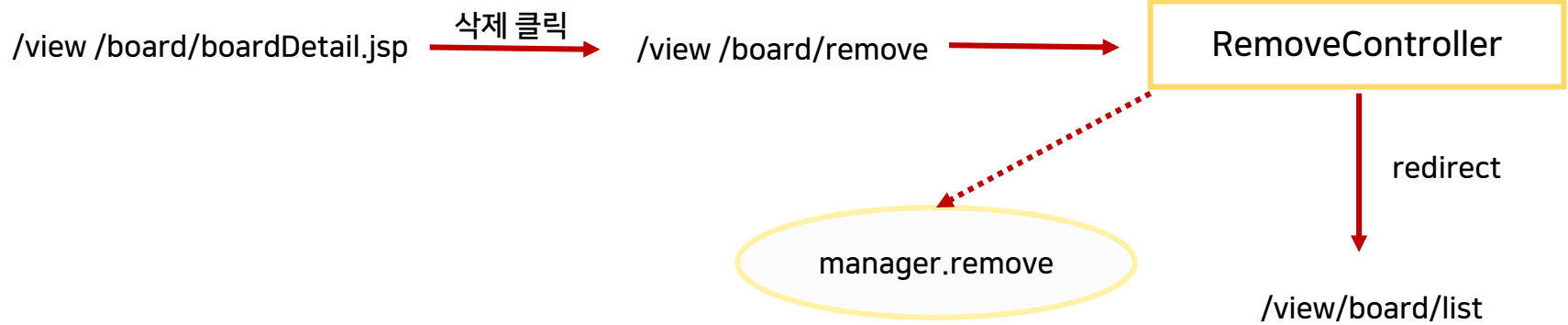




# 08 Request URI

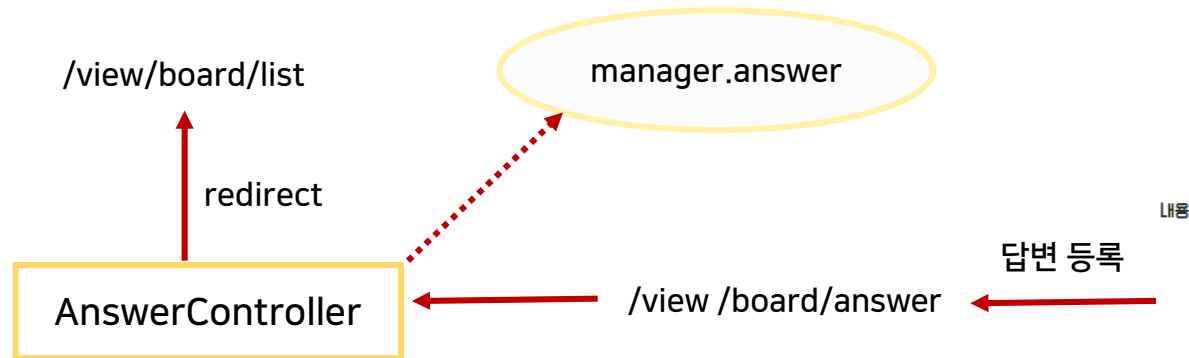
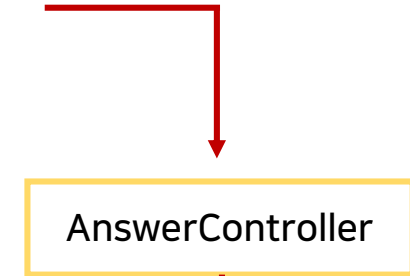


# 08 Request URI



# 08 Request URI

/view /board/boardDetail.jsp → 답변 클릭 → /view /board/update/form?boardNo=..



답변 등록 목록

/view /board/answerForm.jsp

# 09 구현 방법

## ① 데이터 삽입 중 생긴 오류

`java.sql.SQLIntegrityConstraintViolationException: ORA-02291: 무결성 제약조건(DBP0107.R_7)이 위배되었습니다- 부모 키가 없습니다`

→ FK로 선언된 필드에 부모 테이블에 없던 데이터를 삽입하려고 했을 때 생긴 오류

## ② button을 클릭하지 않고도 textArea 內 글자를 변할 수 있게 하기 위해

```
<select id="color" name="color" onChange="DiyPrice();getCount();">
```

→ onClick 대신 onChange를 사용

## ③ value에 name이 저장되어 있지만, name을 클릭할 경우 클릭한 값의 price 알기 위해

```
function DiyPrice(){
    resultPrice = 0;
    var color = document.getElementById("color");
    if (form.color.value == "")
        resultPrice += 0;
    else{
        for(i=0; i<form.color.options.length; i++) {
            if(form.color.options[i].selected == true) {
                switch (i){
```

→ 클릭한 name의 index를 읽어 index에 해당하는 price를 가져옴

# 09 구현 방법

## ① 사용자가 입력하지 않은 부분은 알림창을 띄워 입력을 받게 함

```
function answerBoard() {  
    if (form.answer.value == "") {  
        alert("답변을 입력하십시오.");  
        form.title.focus();  
        return false;  
    }  
    form.submit();  
}
```

```
function boardUpdate() {  
    if (form.title.value == "") {  
        alert("제목을 입력하십시오.");  
        form.title.focus();  
        return false;  
    }  
}
```

## ② 목록 클릭시 해당 uri로 이동하게 함

```
function boardList(targetUri) {  
    form.action = targetUri;  
    form.submit();  
}
```

## ③ 게시판 글 삭제 시 팝업 창 생성

```
function boardRemove() {  
    return confirm("정말 삭제하시겠습니까?");  
}
```

# 09 구현 방법

- ④ DIY에서 사용자가 선택지 클릭시마다 해당 가격을 더해줌.  
glue타입 뿐 아니라 다른 선택지들도 동일하게 작동

```
var glue = document.getElementById("glue");
if (form.glue.value == "")
    resultPrice += 0;
else{
    for(i=0; i<form.glue.options.length; i++) {
        if(form.glue.options[i].selected == true) {
            switch (i){
                case 1 : // amos
                    resultPrice += 5000;
                    break;
                case 2 : // clear glue
                    resultPrice += 5400;
                    break;
                case 3 :// glue-all
                    resultPrice += 6300;
                    break;
                case 4 : // school glue
                    resultPrice += 6000;
                    break;
                case 5 : // wood glue
                    resultPrice += 6600;
                    break;
            }
            break;
        }
    }
}
document.getElementById("Price").value = resultPrice;
```

# 09 구현 방법

⑤ 게시판 내용 보기에서 판매자가 아직 답변하지 않았을 경우 아래의 문구를 띄워줌

```
<tr>
  <td width = "150" align = "center" height = "150">답변</td>
  <td width = "400" align = "center" height = "150">${board.answer}
    <c:if test="${empty board.answer}">
      판매자가 아직 답변을 등록하지 않았습니다.
    </c:if>
  </td>
</tr>
```

⑥ 넘어온 error 를 받아서 해당 메시지를 띄워줌

```
<c:if test="${updateFailed || deleteFailed || answerFailed}">
  <font color="red"><c:out value="${exception.getMessage()}" /></font>
</c:if>
<br><br>
```

⑦ header 부분에서 logout 상태일 때는 login이라는 문구가 보이게 하고, 반대의 경우  
logout( userId ) 문구가 보이게 함

```
<c:if test="${empty sessionScope.user}">
  <li><a href=" <c:url value='/view/user/login/form' />" class="list">login</a></li></c:if>
<c:if test="${!empty sessionScope.user}"><li>
  <a href="<c:url value='/user/logout' />" class = "list" class="list">logout(&nbsp;${sessionScope.user}&nbsp;)</a></li></c:if>
```

# 09 구현 방법

⑧ 금액 계산 중 diy와 완제품 가격의 합계가 40000원 이상이면 무료배송, 미만이면 배송비가 3000원 붙게 함

```
<c:set var="dIndex" value="0" />
<c:set var="dTotalPrice" value="0"/>
<c:forEach var="diy" items="${sessionScope.DIYList}">

    <c:set var="dIndex" value="${dIndex+1}" />
    <c:set var="tTotalPrice" value="${tTotalPrice + diy.diy_total_price}"/>
</c:forEach>

<c:if test="${(tTotalPrice + dTotalPrice) >= 40000}"><c:set var="dlPrice" value="0"/></c:if>
<c:if test="${(tTotalPrice + dTotalPrice) < 40000}"><c:set var="dlPrice" value="3000"/></c:if>
```

⑨ diy에서 select option을 받아오는 과정으로, db에 저장된 option들을 불러오게 했음

```
<td>Slime Color</td>
<td>
    <select id="color" name="color" onChange="DiyPrice();getCount();">
        <option value="">--선택--</option>
        <c:forEach var="color" items="${colorList}">
            <option value="${color.colorType}">${color.colorType}</option>
        </c:forEach>
    </select>
</td>
```



# 09 구현 방법

- ⑩ 저장된 아이템을 한 줄에 3개씩 출력하기 위해 i라는 변수를 설정해 i가 3의 배수일 때 마다 줄바꿈(</tr><tr>)을 하도록 구현

```
<c:set var="i" value="1" />
<c:forEach var="item" items="${itemList}">
  <td>
    <table align = "center">
      <tr><td width="120" height="120"><img src = "<c:url value = '/images/' />${i}.jpg" width = "90%" height = "90%"></td></tr>
      <tr><td width="120" height = "50">
        <a class = "list" href="<c:url value='/view/item/detail'">
          <c:param name="item_name" value='${item.slimename}' />
          </c:url>">
            ${item.slimename}</a>
        </td></tr>
    </table>
  </td>
  <c:if test="${i%3==0}">
    </tr><tr>
  </c:if>
  <c:set var="i" value="${i+1}" />
</c:forEach>
```

# 09 구현 방법

- ① 세션에 저장된 id가 admin이면 게시글에 답변할 수 있으며 아닌 경우 해당 게시글로 redirect 하며 answerFailed라는 에러를 발생시켜 관리자만 답변할 수 있다는 메시지를 띄움

```
if (userId.equals("admin")) {  
    Board board = new Board(boardNo, userId);  
    request.setAttribute("board", board);  
    return "/view/board/answerForm.jsp";  
}  
  
return "redirect:/view/board/detail?boardNo=" + boardNo + "&answerFailed=true";
```

- ② 세션에 저장된 id가 admin이거나 해당 게시글의 작성자인 경우에만 게시글을 지울 수 있게 하고, 아닐 경우 해당 게시글로 redirect 후 deleteFailed라는 에러를 발생시켜 다른 사용자의 글은 지울 수 없다는 메시지를 띄움

```
if (UserSessionUtils.getUserFromSession(request.getSession()).equals("admin")) {  
    BoardManager manager = BoardManager.getInstance();  
    manager.remove(boardNo);  
    return "redirect:/view/board/list";  
}  
  
if (UserSessionUtils.isLoginUser(userId, session)) {  
    BoardManager manager = BoardManager.getInstance();  
    manager.remove(boardNo);  
    return "redirect:/view/board/list";  
}  
  
return "redirect:/view/board/detail?boardNo=" + boardNo + "&deleteFailed=true";
```

# 09 구현 방법

③ board에서 수정,삭제,답변에 대해 넘어온 error를 확인하고 메시지를 넘겨줌

```
if (request.getParameter("updateFailed") != null) {  
    // 수정 시도 실패  
    request.setAttribute("exception",  
        new IllegalStateException("본인이 쓰지 않은 글은 수정할 수 없습니다."));  
    request.setAttribute("updateFailed", true);  
}  
else if (request.getParameter("deleteFailed") != null) {  
    // 삭제 시도 실패  
    request.setAttribute("exception", new IllegalStateException("다른 사용자의 글은 삭제할 수 없습니다."));  
    request.setAttribute("deleteFailed", true);  
}  
  
else if (request.getParameter("answerFailed") != null) {  
    request.setAttribute("exception", new IllegalStateException("관리자만 답변할 수 있습니다."));  
    request.setAttribute("answerFailed", true);  
}
```

# 09 구현 방법

④ 세션에 저장된 장바구니에 담은 완제품 목록과 DIY 목록 가져오고 주문을 생성함

```
if(session.getAttribute("lineItemList") != null) {
    lineItemList = (ArrayList<Lineitem>)session.getAttribute("lineItemList");
    LineItemManager lineItemMan = LineItemManager.getInstance();
    for(int i = 0; i < lineItemList.size(); i++) {
        lineItemList.get(i).setOrderno(orderNo);
        lineItemMan.create(lineItemList.get(i));
    }
}

if(session.getAttribute("DIYList") != null) {
    DIYList = (ArrayList<DIY>)session.getAttribute("DIYList");
    DIYManager diyMan = DIYManager.getInstance();
    for(int i = 0; i < DIYList.size(); i++) {
        DIYList.get(i).setOrderno(orderNo);
        diyMan.create(DIYList.get(i));
    }
}
```

# 09 구현 방법

⑤ 세션에 로그인 정보가 있는 경우에만 주문 가능하도록 설정

```
if (!UserSessionUtils.isLogged(request.getSession())) {  
    return "redirect:/view/user/login/form";  
}
```

⑥ 리스트 사이즈로 삭제 가능 여부 판단

```
if(DIYList == null || DIYList.size() <= index || index < 0) {  
    return "redirect:/view/item/basket";  
}  
  
if(lineItemList == null || lineItemList.size() <= index || index < 0) {  
    return "redirect:/view/item/basket";  
}
```

⑦ 로그아웃 시, 세션에 저장된 장바구니 정보 모두 삭제

```
session.removeAttribute("id");  
session.removeAttribute("lineItemList");  
session.removeAttribute("DIYList");  
session.invalidate();
```

# 09 구현 방법

- ⑧ 사용자가 diy 옵션을 모두 클릭해서 장바구니 버튼을 누르면 해당 옵션들을 불러와 ArrayList와 session에 저장한다. 처음 장바구니에 저장된 게 없으면 ArrayList를 새로 생성 후 session에 저장하며 아닐 경우 session에 저장된 정보를 불러오고 새로운 diy 정보를 list에 add함

```
ArrayList<DIY> DIYList = null;
if(session.getAttribute("DIYList") == null )
    DIYList = new ArrayList<DIY>();
else
    DIYList = (ArrayList<DIY>)session.getAttribute("DIYList");

int index = DIYList.size();

//name id 誤
DIY diy = new DIY();
diy.setDiy_itemno(index);
diy.setDiy_name((String)request.getParameter("diy_name"));
diy.setDiy_amount(Integer.parseInt(request.getParameter("count")));
diy.setScenttype((String)request.getParameter("scent"));
diy.setDecotype((String)request.getParameter("deco"));
diy.setColortype((String)request.getParameter("color"));
diy.setGluetype((String)request.getParameter("glue"));
diy.setDiy_price(Integer.parseInt(request.getParameter("Price")));
diy.setDiy_total_price(Integer.parseInt(request.getParameter("totalPrice")));

DIYList.add(diy);

session.setAttribute("DIYList", DIYList);
```

# 09 구현 방법

⑨ 사용자가 myPage에서 자신의 주문 내역을 받아올 때 orderno를 사용하여 diy와 완제품 주문 정보를 받아옴

```
int orderno = Integer.parseInt(request.getParameter("orderno"));

try{
    LineItemManager lineItemMan = LineItemManager.getInstance();
    DIYManager DIYMan = DIYManager.getInstance();

    List<Lineitem> lineItemList = lineItemMan.findLineItmeList(orderno);
    List<DIY> DIYList = DIYMan.findDIYList(orderno);

    request.setAttribute("lineItemList", lineItemList);
    request.setAttribute("DIYList", DIYList);
}catch(Exception e) {
    e.printStackTrace();
    return "redirect:/view/user/myPage";
}
```

# 09 구현 방법

- ⑩ myPage에 접근시 로그인 되어있지 않으면 loginForm으로 이동, 로그인 되어 있다면 해당 id 정보를 이용해 사용자 정보와 주문 정보를 불러옴

```
if (!UserSessionUtils.isLogined(request.getSession())) {  
    return "redirect:/view/user/login/form"; // login form 占쏙옙정占쏙옙占쏙옙 redirect  
}  
  
try{  
    UserManager userManager = UserManager.getInstance();  
    User user = userManager.findUser(UserSessionUtils.getUserFromSession(request.getSession()));  
    request.setAttribute("user", user);  
  
    OrderManager orderMan = OrderManager.getInstance();  
    List<Order> orderList = orderMan.findOrderList(user.getId());  
    request.setAttribute("orderList", orderList);  
  
}catch(Exception e) {  
    e.printStackTrace();  
    return "redirect:/view/item/showAll";  
}  
  
return "/view/user/myPage.jsp";  
}
```



# 09 구현 방법

```
create sequence board_seq start with 1 increment by 1 maxvalue 9999;
```

- ① DIY의 item\_no는 시퀀스로 부여해줌  
(board의 boardno, order의 orderno도 마찬가지로 sequence 사용)

```
String sql = "INSERT INTO DIY (diy_itemno, diy_amount, diy_price, orderno, gluetype, colortype, scenttype, decotype) " +  
"VALUES (diy_seq.nextval, ?, ?, ?, ?, ?, ?, ?)";
```

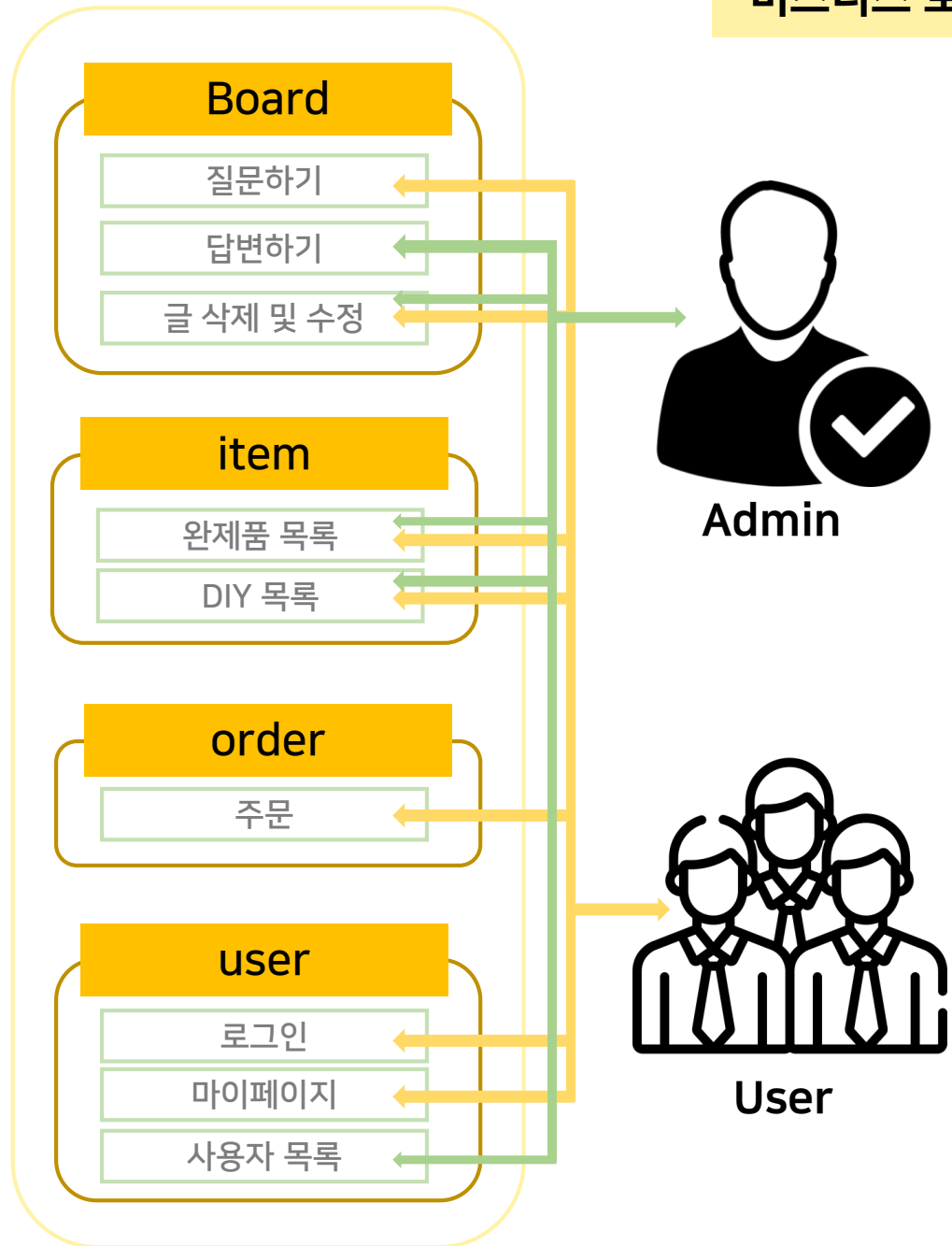
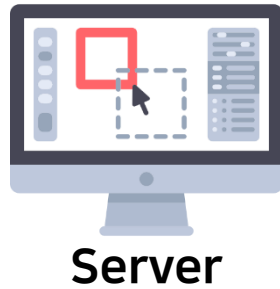
sequence로 생성된 PK를 알아 내기 위해 다음과 같은 방법 사용

```
try {  
    jdbcUtil.executeUpdate(key);  
    ResultSet rs = jdbcUtil.getGeneratedKeys();  
    if(rs.next()) {  
        int generatedKey = rs.getInt(1);  
        diy.setDiy_itemno(generatedKey);  
    }  
    return diy;  
}
```

- ② DB에 저장된 date 타입의 변수를 String형으로 변환시키는 과정  
(board의 upload\_time을 처리하는 것도 동일한 과정)

```
try {  
    ResultSet rs = jdbcUtil.executeQuery();  
    if (rs.next()) {  
        java.text.DateFormat df = new java.text.SimpleDateFormat("yyyy/MM/dd");  
        java.util.Date utilDate = new java.util.Date(rs.getDate("ordertime").getTime());  
        String date = df.format(utilDate);  
        Order order = new Order(orderNo, rs.getString("order_address"), rs.getInt("totalprice"), date);  
        return order;  
    }  
}
```

# 09 구현 방법



# 10 결론, 소감 및 업무분장

김효경 - 따로따로 배웠었던 jsp, jstl과 DB를 연결해서 프로젝트를 만든 것이 신기했고 처음엔 힘들었지만 전반적으로 흐름을 이해하고 나서 부터는 코드를 짜는 것이 수월했던 것 같다. 프로젝트를 완성해서 뿌듯하다.

박나현 - 이 홈페이지를 구성하면서 MVC 구조를 경험하는 좋은 기회가 되었고, 팀원들과 역할 분담을 하고 또한 오류를 서로 같이 잡는 부분이 많은 도움이 되었다. 조금 더 역량이 된다면 진짜 홈페이지처럼 만들어 보고 싶고, 다른 페이지는 만들어질 때 어떻게 구현되는지 보고 싶다. 힘들었지만 나에겐 많은 것이 남은 좋은 프로젝트였다.

성연주 - 팀플로 진행되는 프로젝트는 이번이 처음이었는데, 꽤 색다른 경험이었던 것 같다. 힘들기도 했지만 프로젝트를 통해 많은 것을 배우고 얻어간 것 같아서 값진 경험이었다고 생각한다.

# 10 결론, 소감 및 업무분장

김효경 - model 정의, 게시판 기능 전체, User 관련 controller와 DAO, Item 관련 controller와 DAO, PPT 목차[ 1, 2, 3, 8, 9 ] 및 전반적인 오류 수정

박나현 - css, order관련 controller와 DAO, DIYDAO와 controller, lineItem 관련 DAO와 controller, 장바구니 구현, PPT 목차[ 5, 6 ] 및 전반적인 오류 수정

성연주 - user, order, 장바구니 관련 jsp, PPT 목차[ 4, 7, 9 ]

---

**감사합니다**

THANK YOU :)

---