

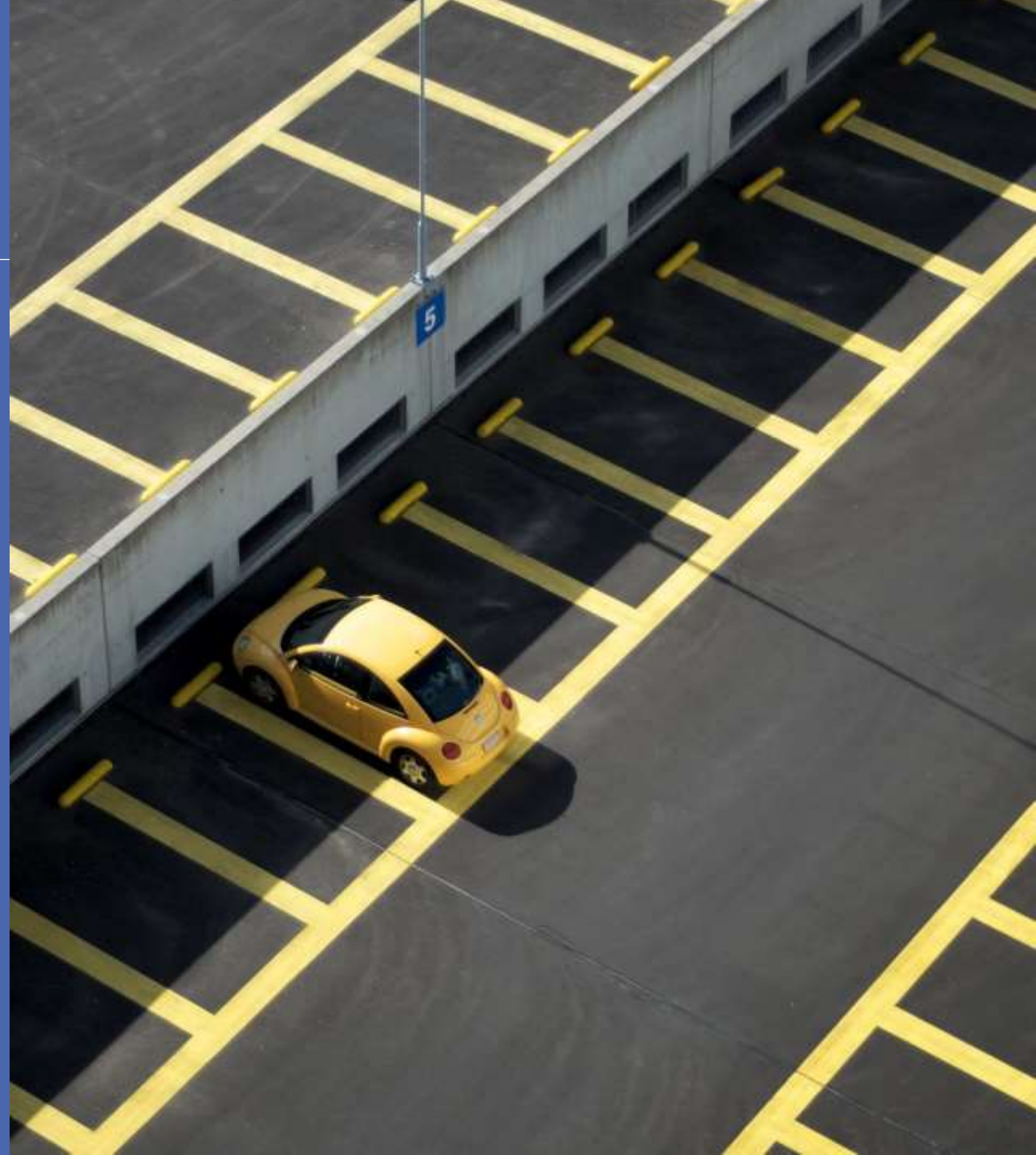
# CHAGAWA

JSP & SERVLET PROJECT

# 목차

A table of contents.

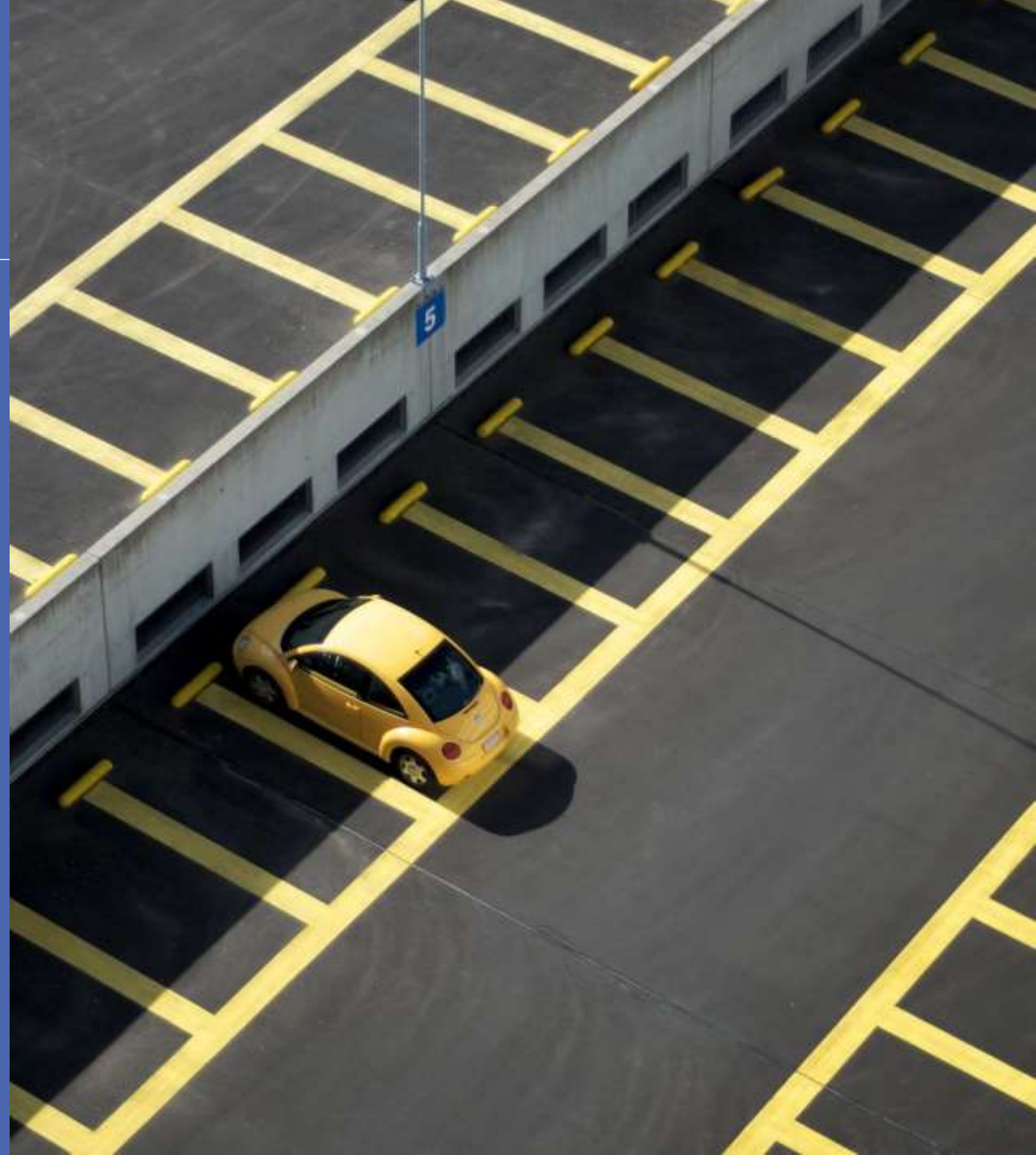
- 01 주제 및 목적
- 02 핵심 요구 사항
- 03 개발 내용 및 담당
- 04 소요 자원
- 05 와이어 프레임



# 목차

A table of contents.

- 06 회원 관리
- 07 공지사항 관리
- 08 포인트 관리
- 09 카풀 관리
- 10 메시지 관리
- 11 이용후기 관리
- 12 개발후기 및 차후개발 내용



# 주제 및 목적

---



주제

---

카풀 서비스  
‘차가와’



목적

---

고객에게 카풀 서비스를 제공해  
수익 창출

# 핵심 요구 사항

차가와 핵심 요구 사항		
번호	테이블명	요구 사항명
1	회원 관리	아이디와 닉네임을 중복될 수 없게 설정해 회원 가입 시 중복 확인
2	공지 사항 관리	비회원과 일반 회원은 현재 공지만, 관리자는 모든 공지를 볼 수 있어야 함
3	포인트 관리	포인트 충전 완료 시 일정 금액이 관리자의 계정에 추가
4	카풀 관리	이미 카풀을 등록 중이거나 동승 중인 경우 새로운 카풀 등록 또는 동승 신청 불가능 동승료보다 보유 포인트가 많은 경우에만 동승 신청 가능
5	메시지 관리	로그인 시 메시지 전송, 메시지 확인, 대화방 구현
6	이용 후기 관리	카풀 이용 후 이용한 카풀에 대한 후기 작성 가능, 본인 글 수정 및 삭제 가능, 글 삭제 후 이용했던 카풀에 대한 후기 재등록 가능, 중복 작성 불가



# 개발 내용 및 담당

주해린

DBA

회원 관리  
시스템

조용우

PM

메시지 관리  
시스템

김태웅

팀장

이용 후기 관리  
시스템

정하영

PL

메인  
카풀 관리  
시스템

김소연

서기

공지 사항  
포인트 관리  
시스템

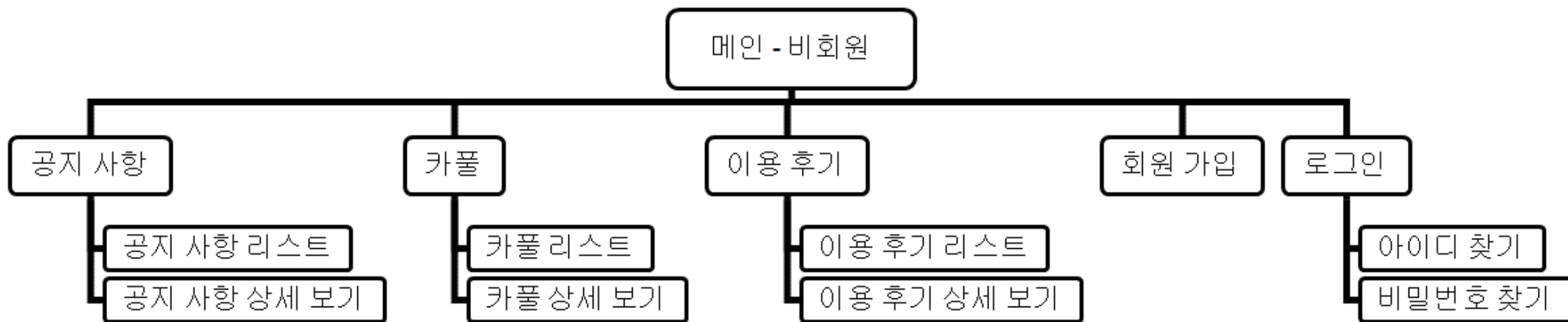
## 개발 내용 및 담당

공통 개발	모듈 개발	담당자	개발 범위
AuthorityFilter default_decorator	메인	정하영	
	회원 관리	주해린	비회원: 회원 가입, 로그인, 아이디 찾기, 비밀번호 찾기 회원: 로그아웃, 내 정보 보기, 내 정보 수정, 비밀번호 변경, 등급 변경, 탈퇴 관리자: 회원 리스트, 회원 상세 보기, 회원 등급 변경, 회원 상태 변경
	공지 사항 관리	김소연	공지 리스트, 공지 보기, 공지 등록, 공지 수정, 공지 삭제
	포인트 관리	김소연	포인트 내역, 수수료 수입 내역, 포인트 충전, 포인트 인출
	카풀 관리	정하영	카풀 리스트, 내 이용 내역, 상세 보기, 카풀 등록, 동승 신청 부터 종료까지 각 단계에 따른 운영 시스템
	메시지 관리	조용우	대화방 리스트, 채팅 로그, 메시지 보내기, 안 읽은 메시지, 방 상태 수정
	이용 후기 관리	김태웅	후기 리스트, 후기 보기, 후기 등록, 후기 수정, 후기 삭제

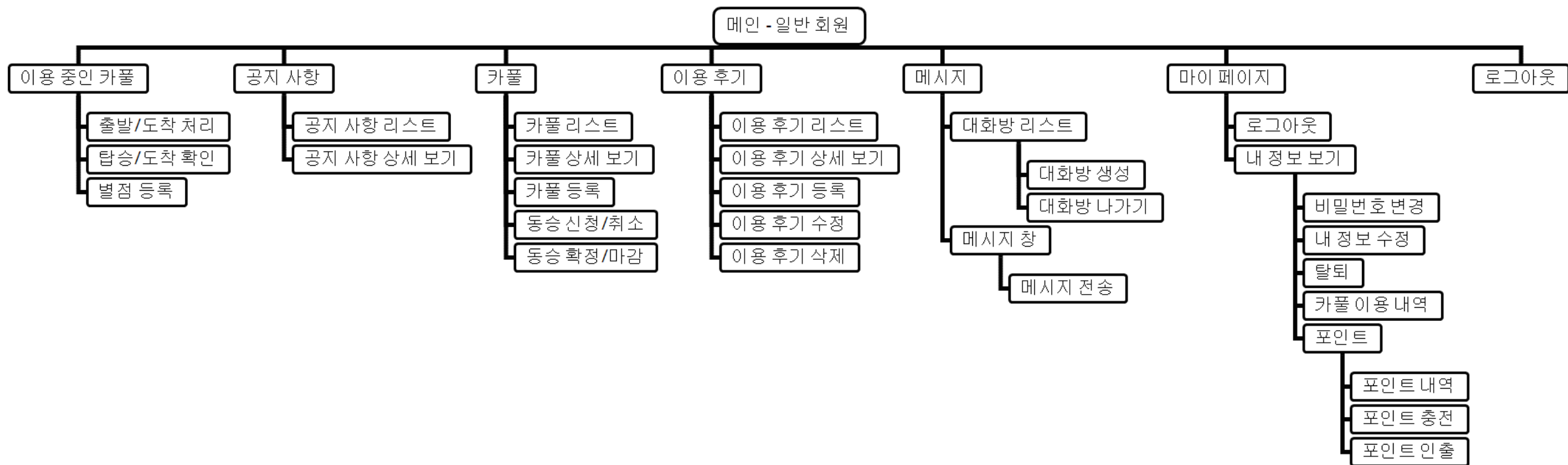
HW	SW	Team
개발 PC SVN PC 서버 PC	Eclipse IDE Oracle SQL Developer Oracle Database Tomcat v9.0	팀장 김태웅 PM 조용우 PL 정하영 DBA 주해린 서기 김소연



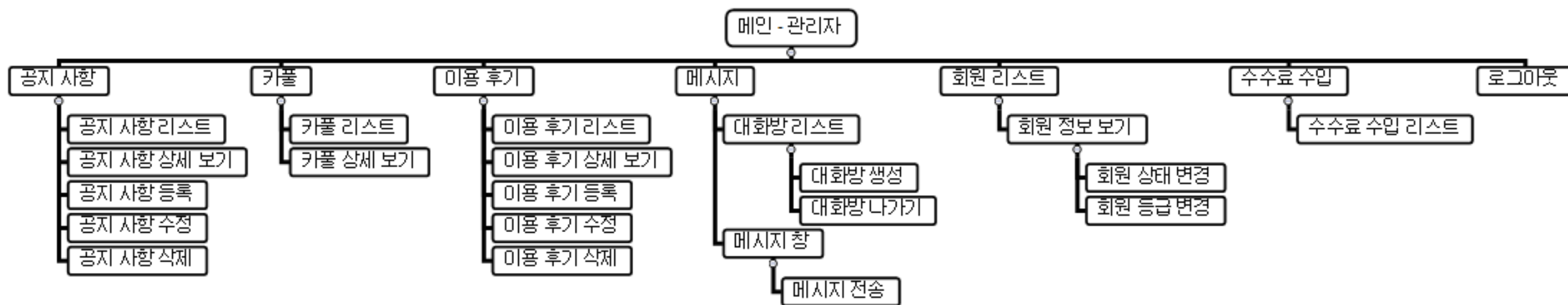
# 와이어 프레임 - 비회원



# 와이어 프레임 - 일반 회원



# 와이어 프레임 - 관리자



Part 6,

회원 관리 - DB 구현

업무 영역		회원 관리 시스템		작성자			주해린	
테이블명		등급		테이블 ID			grade	
테이블 설명		회원 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
등급 번호	gradeNo	number	2		O			
등급명	gradeName	varchar2	30	O				

# 회원 관리 - DB 구현

업무 영역		회원 관리 시스템		작성자			주해린	
테이블명		회원		테이블 ID			member	
테이블 설명		회원 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
아이디	id	varchar2	20		O		영어	
비밀번호	pw	varchar2	20	O				
닉네임	nickname	varchar2	30	O			한글/영어	
이름	name	varchar2	30	O			한글	
프로필 사진	profileImage	varchar2	100					
포인트	myPoint	number	10				default: 200	
성별	gender	varchar2	6	O			남/여	
생년월일	birth	varchar2	10	O				
연락처	tel	varchar2	13	O			01x-xxxx(xxx)-xxxx	
은행	bank	varchar2	15	O				
계좌	account	varchar2	20	O			숫자, '-'	

# 회원 관리 - DB 구현

가입일	regDate	date	-				default: sysdate
최근접속일	conDate	date	-				default: sysdate
상태	status	varchar2	12				정상/이용정지/탈퇴 default: 정상
등급번호	gradeNo	number	2			0	1(동승자)/2(운전자)/9(관리자) default: 1

Part 6,

회원 관리 - DB 구현

업무 영역		회원 관리 시스템		작성자			주해린	
테이블명		운전자		테이블 ID			driver	
테이블 설명		회원 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
아이디	id	varchar2	20		O			
차종	carModel	varchar2	50	O				
차량 번호	carNo	varchar2	11	O			xx(xxx)가xxxx	
면허 번호	licenseNo	varchar2	15	O			xx-xx-xxxxxx-xx	



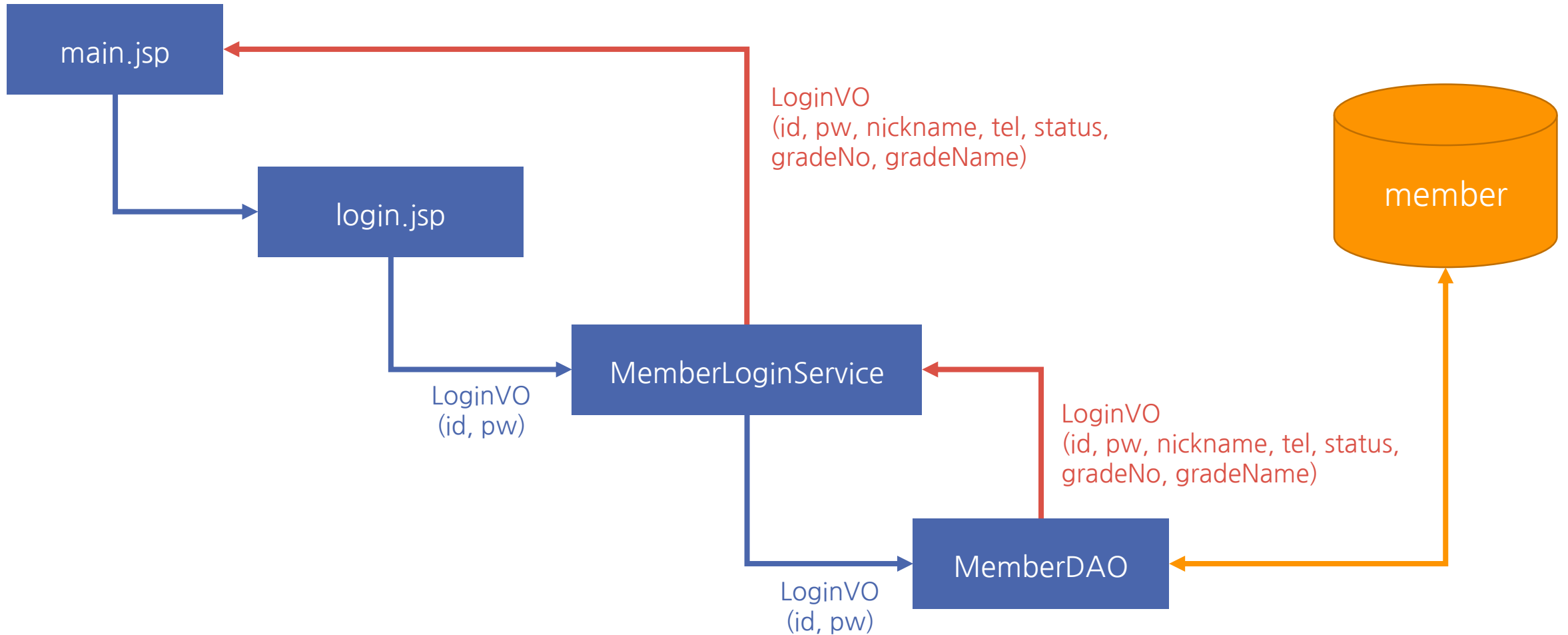
# 회원 관리 - DB 구현

CHAGAWA.MEMBER		
P *	ID	VARCHAR2 (20 BYTE)
*	PW	VARCHAR2 (20 BYTE)
U *	NICKNAME	VARCHAR2 (30 BYTE)
*	NAME	VARCHAR2 (30 BYTE)
	PROFILEIMAGE	VARCHAR2 (100 BYTE)
	MYPOINT	NUMBER (10)
*	GENDER	VARCHAR2 (6 BYTE)
*	BIRTH	VARCHAR2 (10 BYTE)
U *	TEL	VARCHAR2 (13 BYTE)
*	BANK	VARCHAR2 (15 BYTE)
*	ACCOUNT	VARCHAR2 (20 BYTE)
	REGDATE	DATE
	CONDATE	DATE
	STATUS	VARCHAR2 (12 BYTE)
F	GRADENO	NUMBER (2)
MEMBER_PK (ID)		
MEMBER_NICKNAME_UN (NICKNAME)		
MEMBER_TEL_UN (TEL)		
SYS_C009160 (GRADENO)		

	⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ NULLABLE	DATA_DEFAULT	⚡ COLUMN_ID	⚡ COMMENTS
1	ID	VARCHAR2 (20 BYTE)	No	(null)	1	(null)
2	PW	VARCHAR2 (20 BYTE)	No	(null)	2	(null)
3	NICKNAME	VARCHAR2 (30 BYTE)	No	(null)	3	(null)
4	NAME	VARCHAR2 (30 BYTE)	No	(null)	4	(null)
5	PROFILEIMAGE	VARCHAR2 (100 BYTE)	Yes	(null)	5	(null)
6	MYPOINT	NUMBER (10, 0)	Yes	200	6	(null)
7	GENDER	VARCHAR2 (6 BYTE)	No	(null)	7	(null)
8	BIRTH	VARCHAR2 (10 BYTE)	No	(null)	8	(null)
9	TEL	VARCHAR2 (13 BYTE)	No	(null)	9	(null)
10	BANK	VARCHAR2 (15 BYTE)	No	(null)	10	(null)
11	ACCOUNT	VARCHAR2 (20 BYTE)	No	(null)	11	(null)
12	REGDATE	DATE	Yes	sysdate	12	(null)
13	CONDATE	DATE	Yes	sysdate	13	(null)
14	STATUS	VARCHAR2 (12 BYTE)	Yes	'정상'	14	(null)
15	GRADENO	NUMBER (2, 0)	Yes	1	15	(null)

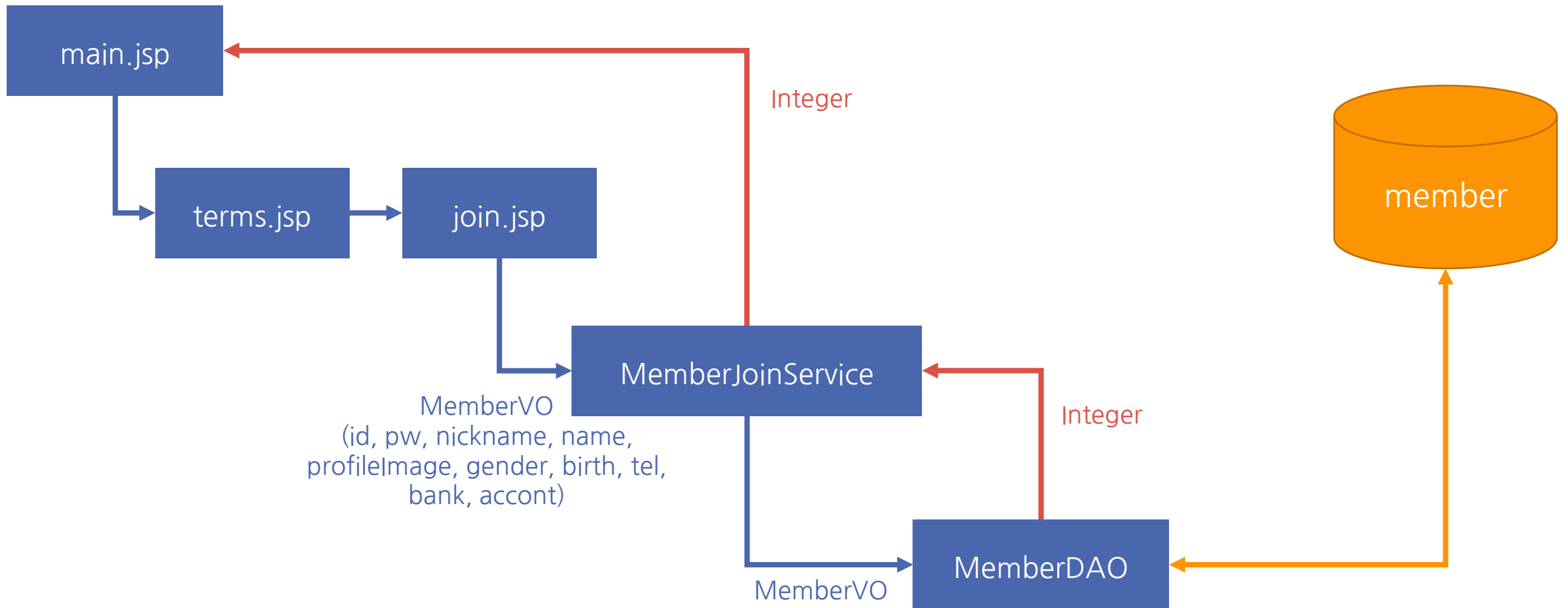
## 회원 관리 - DFD

로그인

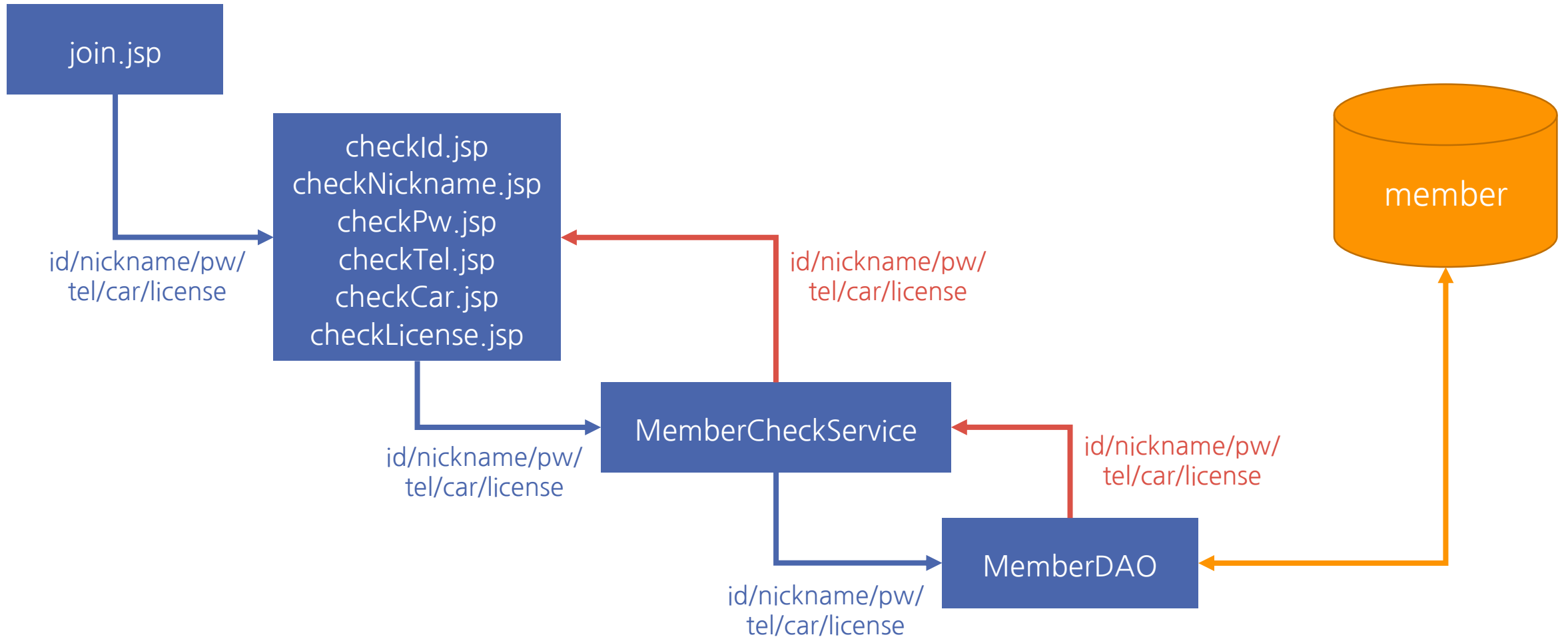


## 회원 관리 - DFD

회원가입

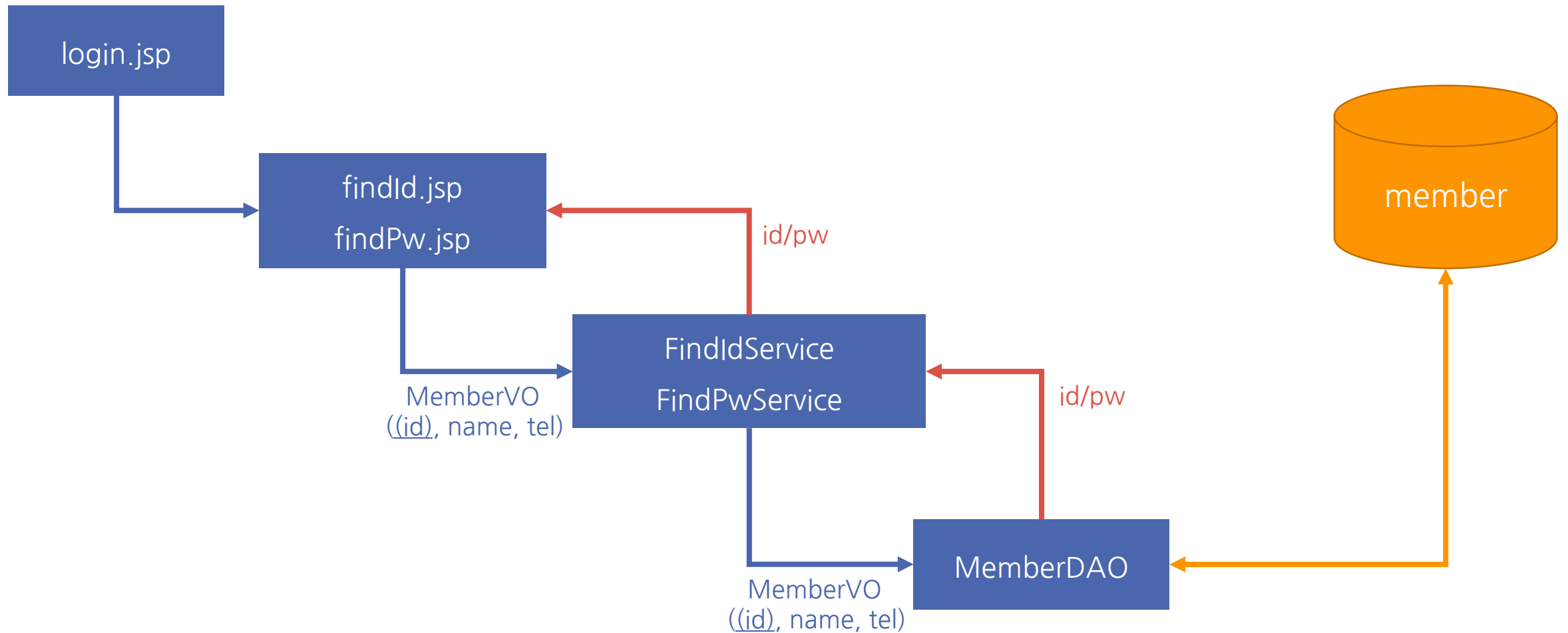


중복확인

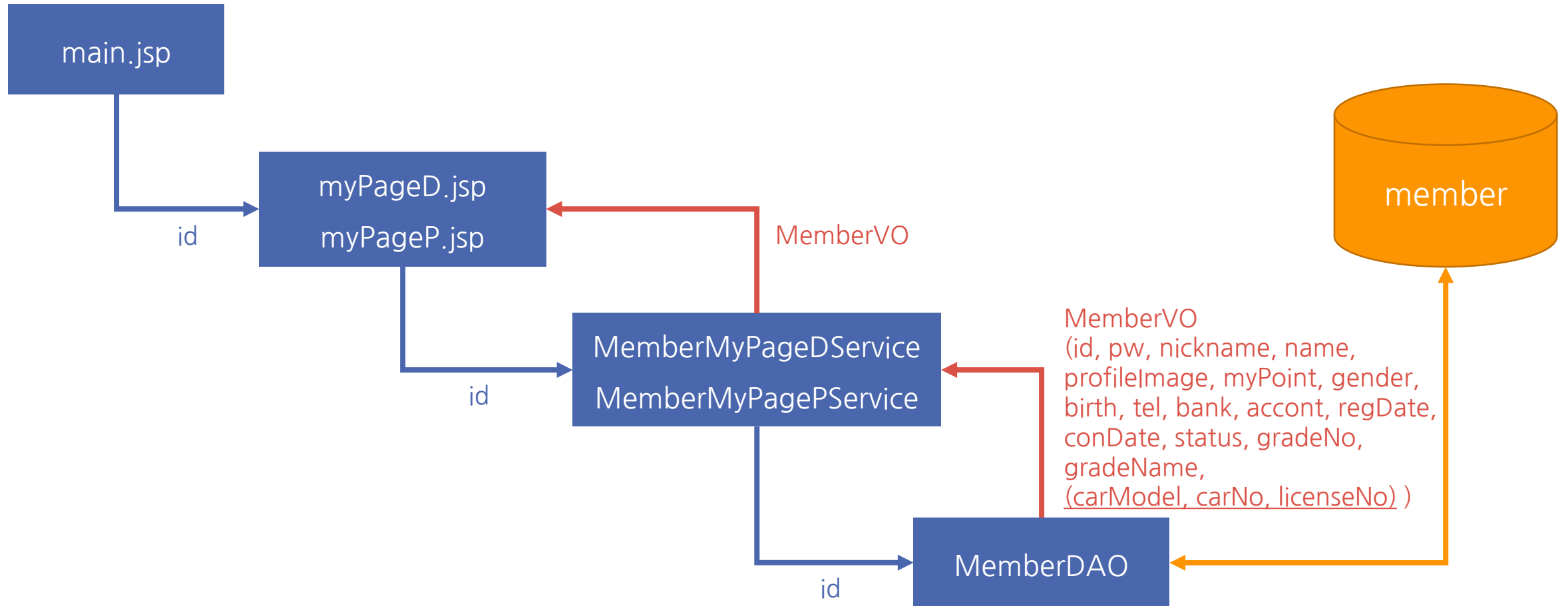


# 회원 관리 - DFD

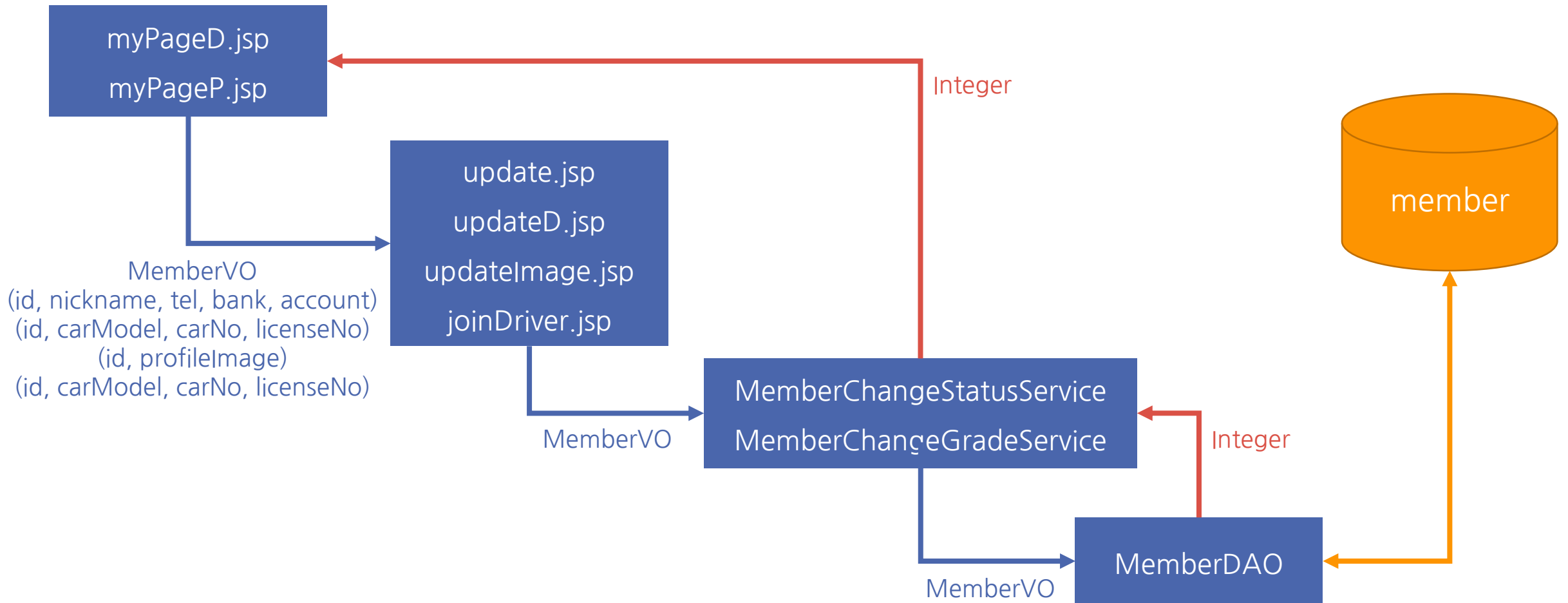
아이디/비밀번호 찾기



내정보보기



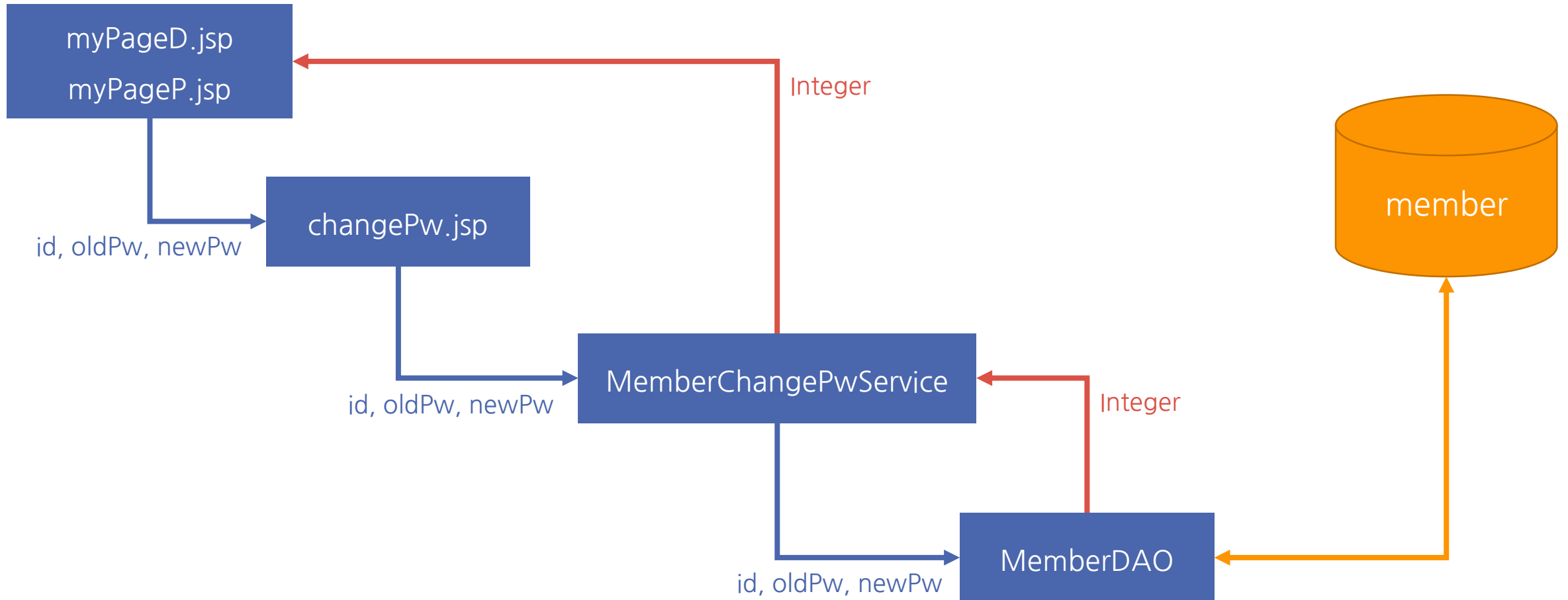
## 내정보수정





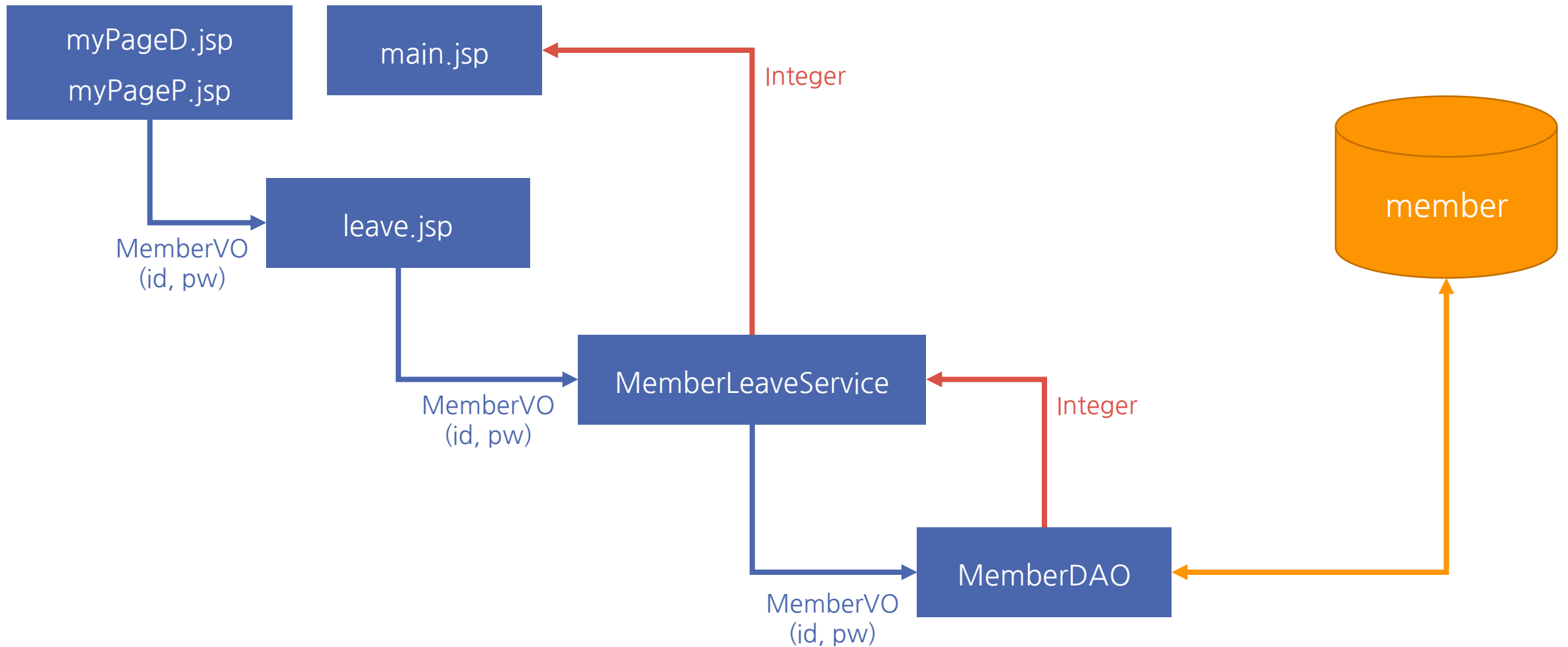
## 회원 관리 - DFD

비밀번호 변경



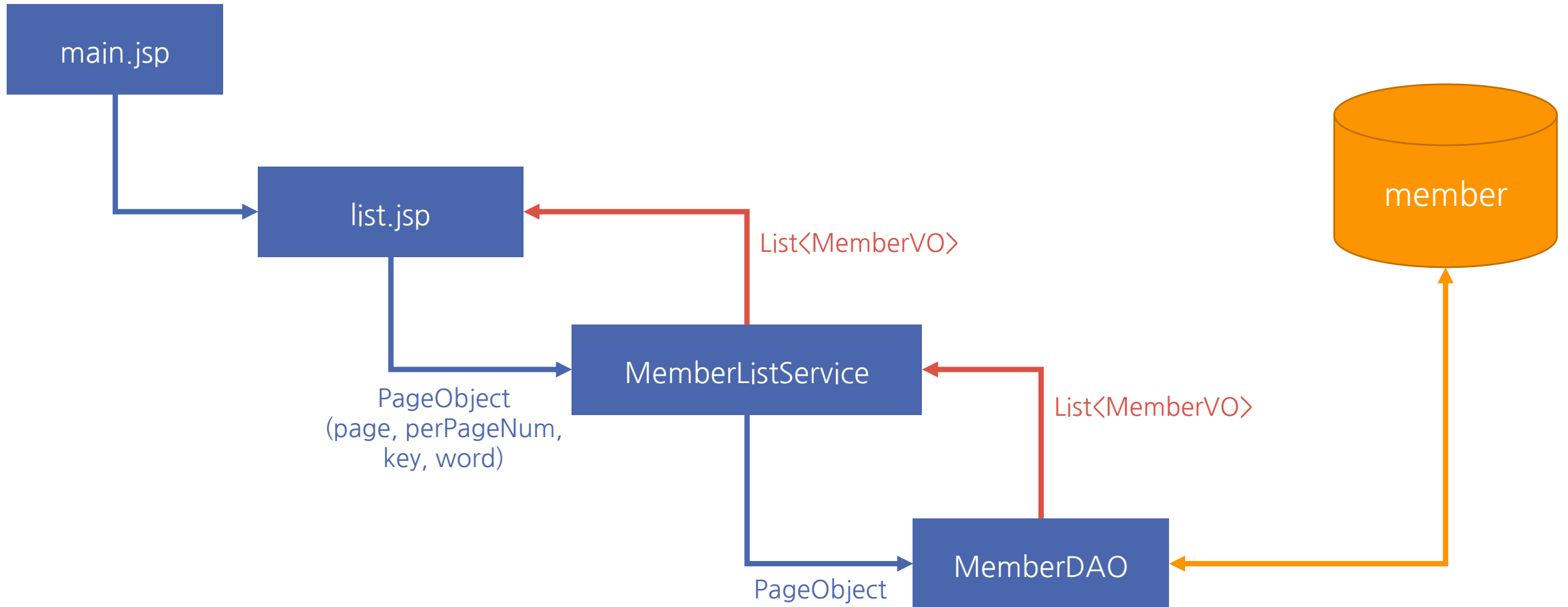
## 회원 관리 - DFD

탈퇴

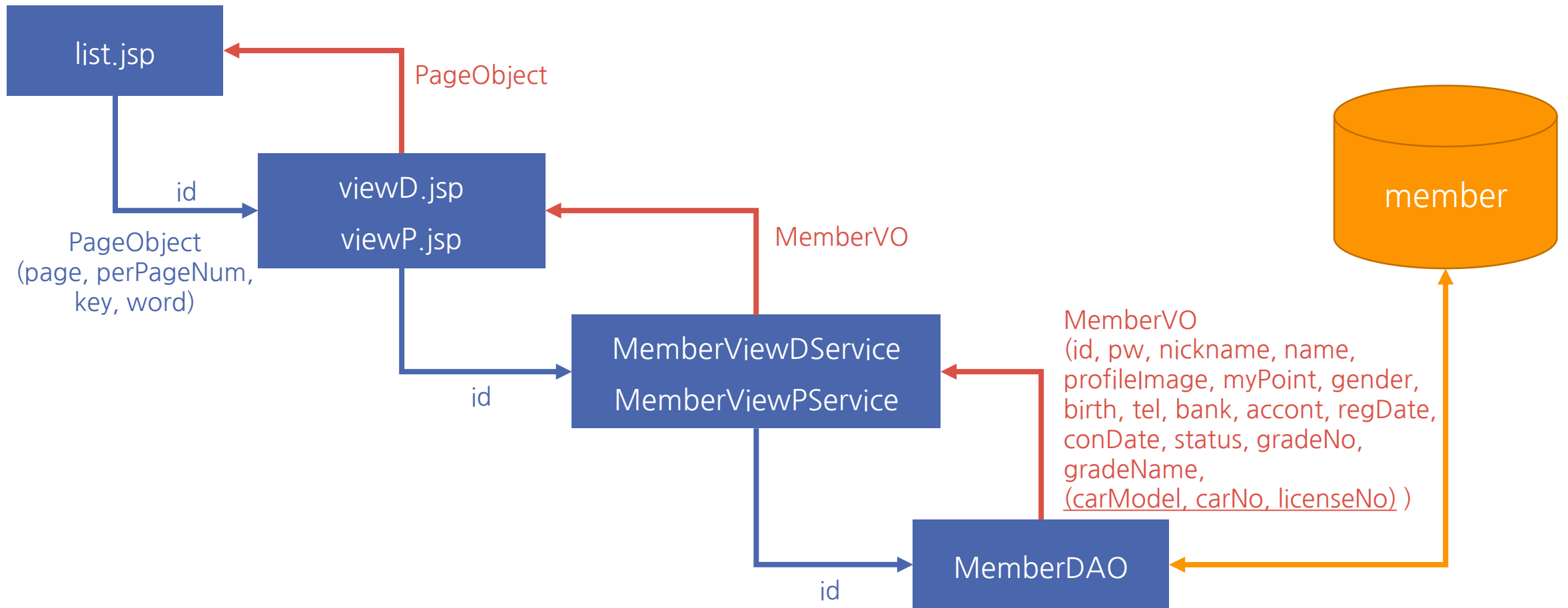


# 회원 관리 - DFD

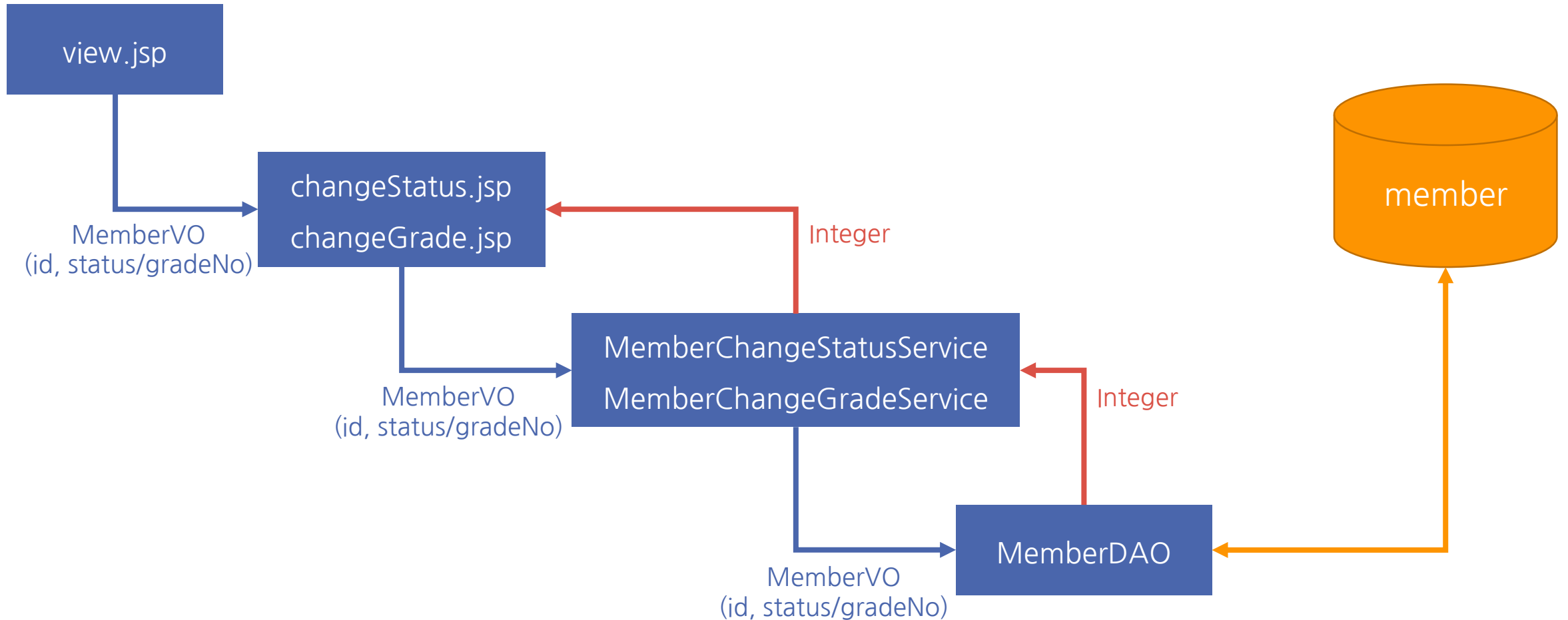
회원리스트



## 회원정보보기



상태/등급 변경



# 회원 관리 - 화면 구현

## 약관동의

### 약관동의

여러분은 본 약관을 읽고 동의하신 후 회원 가입을 신청하실 수 있으며, 차가와는 이에 대한 승낙을 통해 회원 가입 절차를 완료하고 여러분께 차가와 서비스 이용 계정을 부여합니다. 자세히 읽은 후 동의하여 주시기 바랍니다. ✕

여러분을 환영합니다.  
차가와 서비스 및 제품(이하 '서비스')을 이용해 주셔서 감사합니다. 본 약관은 다양한 차가와 서비스의 이용과 관련하여 차가와 서비스를 제공하는 차가와 주식회사(이하 '차가와')와 이를 이용하는 차가와 서비스 회원(이하 '회원') 또는 비회원과의 관계를 설명하며, 아울러 여러분의 차가와 서비스 이용에 도움이 될 수 있는 유익한 정보를 포함하고 있습니다.

차가와 서비스를 이용하시거나 차가와 서비스 회원으로 가입하실 경우 여러분은 본 약관 및 관련 운영 정책을 확인하거나 동의하게 되므로, 잠시 시간을 내시어 주의 깊게 살펴봐 주시기 바랍니다.

다양한 차가와 서비스를 즐겨보세요.  
차가와는 www.chagawa.com을 비롯한 차가와 도메인의 웹사이트 및 응용프로그램(어플리케이션, 앱)을 통해 정보 검색, 다른 이용자와의 커뮤니케이션, 콘텐츠 제공, 상품 쇼핑 등 여러분의 생활에 편리함을 더할 수 있는 다양한 서비스를 제공하고 있습니다.  
여러분은 PC, 휴대폰 등 인터넷 이용이 가능한 각종 단말기를 통해 각양각색의 차가와 서비스를 자유롭게 이용하실 수 있으며, 개별 서비스들의 구체적인 내용은 각 서비스 상의 안내, 공지사항, 차가와 웹고객센터(이하 '고객센터') 도움말 등에서 쉽게 확인하실 수 있습니다.

차가와는 기본적으로 여러분 모두에게 동일한 내용의 서비스를 제공합니다. 다만, '청소년보호법' 등 관련 법령이나 기타 개별 서비스 제공에서의 특별한 필요에 의해서 연령 또는 일정한 등급을 기준으로 이용자를 구분하여 제공하는 서비스의 내용, 이용 시간, 이용 횟수 등을 다르게 하는 등 일부 이용을 제한하는 경우가 있습니다. 자세한 내용은 역시 각 서비스 상의 안내, 공지사항, 고객센터 도움말 등에서 확인하실

☐ 위 약관에 동의합니다.(필수)

확인

취소

localhost 내용:

약관에 동의하지 않은 경우 회원가입이 불가능합니다.

확인

# 회원 관리 - 화면 구현

## 회원가입

### 회원가입

아이디:

비밀번호:

비밀번호 확인:

닉네임:

이름:

프로필사진:    
 ※ 사진을 선택하지 않은 경우, 기본 이미지로 등록됩니다.

성별:

생년월일:

연락처:

계좌:

localhost 내용:

id는 4~20자, 영문자와 숫자로만 작성하고 영문자로 시작해야 합니다.

localhost 내용:

비밀번호는 4~20자로 작성해야 합니다.

localhost 내용:

비밀번호와 비밀번호 확인은 같아야 합니다.

localhost 내용:

이름은 2~10자, 한글로만 작성해야 합니다.

localhost 내용:

성별을 선택해주세요.

localhost 내용:

연도는 네자리 모두 입력하셔야 합니다.

localhost 내용:

월을 선택해주세요.

localhost 내용:

일은 두자리로 입력하셔야 합니다.

localhost 내용:

전화번호는 숫자로만 작성해야 하고 첫번째 공백은 [xxx] 또는 [xxxx] 패턴으로 입력해야 합니다.

localhost 내용:

전화번호는 숫자로만 작성해야 하고 두번째 공백은 [xxxx] 패턴으로 입력해야 합니다.

localhost 내용:

은행을 선택해주세요.

localhost 내용:

계좌번호는 [-]를 포함하여 정확히 입력해 주세요.

확인



# 회원 관리 - 화면 구현

## 회원가입

**회원가입**

아이디: 영문, 숫자 4자 이상 중복확인

비밀번호: 영문, 숫자, 특수기호 4자 이상

비밀번호 확인:

닉네임: 한글, 영문 2자 이상 중복확인

이름: 한글

프로필사진: 파일 선택 선택된 파일 없음  
※ 사진을 선택하지 않은 경우, 기본 이미지로 등록됩니다

성별: -성별-

생년월일: 연도(4자) 월(2자) 일(2자)

연락처: 010 3-4자 4자 중복확인

계좌: 은행- 를 포함하여 정확히 입력해주세요

회원가입 새로입력 취소

**이미 사용중인 아이디입니다.  
다른 아이디를 입력해주세요.**

아이디: admin 중복확인

**이미 사용중인 닉네임입니다.  
다른 닉네임을 입력해 주세요.**

아이디: admin 중복확인

**이미 사용중인 연락처입니다.  
다른 연락처를 입력해주세요.**

연락처: 010 0000 0000 중복확인

**사용가능한 아이디 입니다.**

아이디: test 사용하기

**사용가능한 닉네임 입니다.**

닉네임: test 사용하기

**사용가능한 연락처 입니다.**

연락처: 010 0000 7896 사용하기

# 회원 관리 - 화면 구현

## 회원가입

**회원가입**

아이디:

비밀번호:

비밀번호 확인:

닉네임:

이름:

프로필사진:    
 ※ 사진을 선택하지 않은 경우, 기본 이미지로 등록됩니다.

성별:

생년월일:

연락처:

계좌:

생년월일:

2010

10

▼

01

localhost 내용:

만 14세 미만은 회원가입 하실 수 없습니다.

확인

localhost 내용:

회원가입이 완료되었습니다.

확인

# 회원 관리 - 화면 구현

로그인

로그인

 아이디

 비밀번호

아이디찾기

비밀번호찾기


로그인


localhost 내용:  
driver님 환영합니다.

확인

로그인

로그인 처리되지 않았습니다. 아이디나 비밀번호를 다시 확인해 주세요.

 아이디


 비밀번호

아이디찾기

비밀번호찾기

로그인

탈퇴한 회원은 로그인할 수 없습니다.

 아이디

로그인

이용정지된 회원은 로그인할 수 없습니다.

 아이디

 비밀번호

아이디찾기

비밀번호찾기

로그인

# 회원 관리 - 화면 구현

아이디 찾기

로그인

아이디 찾기

이름:

연락처:

아이디 찾기

취소

아이디 찾기에 실패했습니다. 정보를 다시 입력해주세요.

확인

아이디는 **driver** 입니다.

확인

## 회원 관리 - 화면 구현

비밀번호 찾기

로그인

아이디

비밀번호

아이디찾기 비밀번호찾기 로그인

비밀번호 찾기

아이디:

이름:

연락처: 010

비밀번호 찾기

취소

비밀번호 찾기에 실패했습니다. 정보를 다시 입력해주세요.

확인

비밀번호 찾기에 성공했습니다.  
변경할 비밀번호를 입력해주세요.

새 비밀번호:

비밀번호 확인:

변경하기

# 회원 관리 - 화면 구현

내 정보 보기

내 정보 보기

프로필 사진

회원가입

로그인

나의 포인트

200

카를 이용내역

포인트

아이디	passenger
닉네임	비밀주요
이름	김철수
성별	남자
생년월일	2000 01 01
연락처	010-0000-1234
계좌	3333-1234-0000(카카오)
등급	<div> <div>승용차</div> <div>특수차량</div> </div>
가입일	23 03 03

비밀번호 변경

내정보수정

탈퇴

내 정보 보기

프로필 사진

회원가입

로그인

나의 포인트

5190

카를 이용내역

포인트

아이디	driver
닉네임	driver
이름	김문현
성별	여자
생년월일	1994 01 01
연락처	010-0000-0001
계좌	0000-0000-0000(신한)
등급	<div> <div>승용차</div> <div>특수차량</div> </div>
가입일	23 03 03

비밀번호 변경

내정보수정

탈퇴

차종	미니쿠퍼
차량번호	123가1234
번호번호	12-34-567890-11

비밀번호 변경

내정보수정

탈퇴

# 회원 관리 - 화면 구현

내 정보 수정

내 정보 수정

아이디:	<input type="text" value="passenger"/>		
닉네임:	<input type="text" value="태워주세요"/>	<input type="button" value="중복확인"/>	
이름:	<input type="text" value="김철수"/>		
성별:	<input type="text" value="남자"/>		
생년월일:	<input type="text" value="2000.01.01"/>		
연락처:	<input type="text" value="010"/>	<input type="text" value="0000"/>	<input type="text" value="1234"/> <input type="button" value="중복확인"/>
계좌:	<input type="text" value="카카오"/>	<input type="text" value="3333-1234-0000"/>	
등급:	<input type="text" value="동승자"/>		
가입일:	<input type="text" value="23.03.03"/>		

내 정보 수정

아이디:	<input type="text" value="driver"/>		
닉네임:	<input type="text" value="driver"/>	<input type="button" value="중복확인"/>	
이름:	<input type="text" value="김운전"/>		
성별:	<input type="text" value="여자"/>		
생년월일:	<input type="text" value="1994.01.01"/>		
연락처:	<input type="text" value="010"/>	<input type="text" value="0000"/>	<input type="text" value="0001"/> <input type="button" value="중복확인"/>
계좌:	<input type="text" value="신한"/>	<input type="text" value="0000-0000-0000"/>	
등급:	<input type="text" value="운전자"/>		
가입일:	<input type="text" value="23.03.03"/>		
차종:	<input type="text" value="미니쿠퍼"/>		
차량번호:	<input type="text" value="123가1234"/>	<input type="button" value="중복확인"/>	
면허번호:	<input type="text" value="12-34-567890-11"/>		

수정 새로입력 취소



# 회원 관리 - 화면 구현

이미지 수정



이미지 수정

기본이미지로 변경

이미지 선택: 파일 선택 donggeul3.jpg

바꾸기

취소



이미지 수정

기본이미지로 변경



이미지 수정

기본이미지로 변경

바꾸기

취소



이미지 수정

기본이미지로 변경

# 회원 관리 - 화면 구현

## 비밀번호 변경

기존 비밀번호:

새 비밀번호:

비밀번호 확인:

변경하기

localhost 내용:

비밀번호와 비밀번호 확인은 같아야 합니다.

확인

비밀번호 변경에 실패했습니다.

[\[내정보보기\]로 돌아가기](#)

비밀번호 변경이 완료되었습니다.

[\[내정보보기\]로 돌아가기](#)

# 회원 관리 - 화면 구현

등급 전환

운전자등급으로 전환

차종:

차량번호:

공백없이 정확히 입력해주세요.

중복확인

면허번호:

2자

- 2자

- 6자

- 2자

중복확인

등급 변경

새로입력

취소

localhost 내용:

만 18세 미만은 운전면허를 취득할 수 없으므로 운전자 등급 전환이 불가능합니다.

확인

localhost 내용:

차종은 한글, 영문으로 20자 이내로 입력해 주세요.

확인

localhost 내용:

차량번호는 띄어쓰기 없이 [xx가xxxx] 또는 [xxx가xxxx] 패턴으로 입력해야 합니다.

확인

localhost 내용:

면허번호는 [xx-xx-xxxxxx-xx] 패턴으로 입력해야 합니다.

확인

# 회원 관리 - 화면 구현

등급 전환

운전자등급으로 전환

차종:

차량번호:  중복확인

면허번호:  -  -  -  중복확인

등급변경

새로입력

취소

운전자 등급으로 변경되었습니다.

[내정보보기]로 돌아가기

이미 사용중인 차량번호입니다.  
다른 차량번호를 입력해 주세요.

차량번호:

중복확인

사용가능한 차량번호 입니다.

차량번호:

사용하기

이미 사용중인 면허번호입니다.  
다른 면허번호를 입력해 주세요.

면허번호:  -  -  -

중복확인

사용가능한 면허번호 입니다.

면허번호:  -  -  -

사용하기

# 회원 관리 - 화면 구현

탈퇴

비밀번호 변경

내정보수정

탈퇴

비밀번호 확인:

탈퇴

취소

localhost 내용:

비밀번호가 일치하지 않습니다.

확인

localhost 내용:

탈퇴 처리 되었습니다.

확인

# 회원 관리 - 화면 구현

## 회원 리스트

### 회원리스트

아이디

▼

Search

Q

게시글 수: 10

▼

아이디	닉네임	성별	생년월일	가입일	상태	가입유형
admin	admin	여자	1998.03.18	23.03.03	정상	관리자
driver	driver	여자	1994.01.01	23.03.03	정상	운전자
driver2	전직택시운전사	여자	1988.01.01	23.03.03	정상	운전자
driver3	어제면허땀	여자	1970.01.01	23.03.03	정상	운전자
passenger	태워주세요	남자	2000.01.01	23.03.03	정상	운전자
passenger2	의정부통근러	남자	1990.01.01	23.03.03	정상	동승자
passenger3	seventeen	남자	2005.03.10	23.03.03	정상	동승자
test	test	남자	2000.01.01	23.03.03	탈퇴	동승자

◀◀

◀

1

▶

▶▶

# 회원 관리 - 화면 구현

## 회원 리스트 - 페이징

아이디 ▼

Search

Q

게시글 수: 5 ▼

아이디	닉네임	성별	생년월일	가입일	상태	가입유형
admin	admin	여자	1998.03.18	23.03.03	정상	관리자
driver	driver	여자	1994.01.01	23.03.03	정상	운전자
driver2	전직택시운전사	여자	1988.01.01	23.03.03	정상	운전자
driver3	어제 면허땀	여자	1970.01.01	23.03.03	정상	운전자
passenger	태워주세요	남자	2000.01.01	23.03.03	정상	운전자

# 회원 관리 - 화면 구현

## 회원 리스트 - 검색

닉네임

의정부

Q

게시글 수: 10

아이디	닉네임	성별	생년월일	가입일	상태	가입유형
passenger2	의정부통근러	남자	1990.01.01	23.03.03	정상	등승자

아이디

driver

Q

게시글 수: 10

아이디	닉네임	성별	생년월일	가입일	상태	가입유형
driver	driver	여자	1994.01.01	23.03.03	정상	운전자
driver2	전직택시운전사	여자	1988.01.01	23.03.03	정상	운전자
driver3	어제면허땀	여자	1970.01.01	23.03.03	정상	운전자



# 회원 관리 - 화면 구현

## 회원 상세보기

### [의정부통근러] 님의 정보

프로필사진	
아이디	passenger2
닉네임	의정부통근러
이름	이유리
성별	남자
생년월일	1990.01.01
연락처	010-0000-1111
포인트	200
계좌	3333-1111-000000(카카오)
가입일	23.03.03
회군접속일	23.03.03
상태	정상 <span>상태변경</span>
등급	1등급(통승자) <span>등급변경</span>

[리스트](#)

### [driver] 님의 정보

프로필사진	
아이디	driver
닉네임	driver
이름	김운전
성별	여자
생년월일	1994.01.01
연락처	010-0000-0001
포인트	5190
계좌	0000-0000-0000(신한)
가입일	23.03.03
회군접속일	23.03.03
상태	정상 <span>상태변경</span>
등급	2등급(운전자) <span>등급변경</span>
차종	미니쿠파
차량번호	123가1234
면허번호	12-34-567890-11

[리스트](#)

# 회원 관리 - 화면 구현

## 회원 상태 변경

상태	<div>정상</div> <div>상태변경</div> <div>변경할 상태: <div>정상 ▼</div> <div>변경</div> <div>취소</div></div>
상태	<div>정상</div> <div>상태변경</div> <div>변경할 상태: <div>이용정지 ▼</div> <div>변경</div> <div>취소</div></div> <div><div>정상</div><div>이용정지</div></div>
상태	<div>이용정지</div> <div>상태변경</div>

# 회원 관리 - 화면 구현

## 회원 등급 변경

등급

1등급(동승자)

등급변경

변경할 등급:

1등급(동승자)



변경

취소

등급

1등급(동승자)

등급변경

변경할 등급:

1등급(동승자)



변경

취소

1등급(동승자)

9등급(관리자)

등급

2등급(운전자)

등급변경

변경할 등급:

2등급(운전자)



변경

취소

2등급(운전자)

9등급(관리자)

```
@Override
public String check(String data, String colName) throws Exception {
    System.out.println("MemberDAOImpl-----");
    String result = null;
    try {
        con = DB.getConnection();

        String sql = null;
        if(colName == "carNo" || colName == "licenseNo") {
            sql = "select " + colName + " from driver where " + colName + " = ?";
        } else {
            sql = "select " + colName + " from member where " + colName + " = ?";
        }

        pstmt = con.prepareStatement(sql);
        pstmt.setString(1, data);

        rs = pstmt.executeQuery();

        if(rs != null && rs.next()) {
            result = rs.getString(colName);
        }

    } catch (Exception e) {
        e.printStackTrace();
        throw new Exception("중복체크 DB");
    } finally {
        DB.close(con, pstmt, rs);
    } // end of try~catch~finally

    return result ;
}
```

< 컬럼과 데이터를 각각 입력받아 하나의 서비스로  
아이디, 닉네임, 연락처 등 다양한 항목 중복 확인 가능 >

```
case "/member/checkNickname.sub":
    // 중복확인할 값
    String nickname = request.getParameter("nickname");
    // 확인된 중복된 값
    String checkNickname = (String) Execute.run(memberCheckService, new Object[] {nickname, "nickname"});

    // 중복되고 로그인 상태라면(수정할 때의 경우)
    if(checkNickname != null && (LoginVO)(session.getAttribute("login")) != null) {
        MemberVO checkNicknameVO = (MemberVO) Execute.run(memberViewPService, ((LoginVO)(session.getAttribute("login"))).getId());
        // 중복된 값이 로그인된 아이디의 값과 일치한다면
        if(checkNickname.equals(checkNicknameVO.getNickname())) {
            request.setAttribute("type", 2);
            jsp = "member/checkNickname";
        } else {
            request.setAttribute("type", 1);
            jsp = "member/checkNickname";
        }
    }

    // 중복되고 로그인 안 한 상태라면 (회원가입의 경우)
    } else if(checkNickname != null && (LoginVO)(session.getAttribute("login")) == null) {
        request.setAttribute("type", 1);
        jsp = "member/checkNickname";

    // 중복 안 된 상태
    } else {
        request.setAttribute("type", 2);
        jsp = "member/checkNickname";
    }
    break;
```

< 중복 확인 시 신규 가입과 수정의 경우를 구분해 처리,  
수정 시 중복된 값과 현재 세션의 값이 같을 경우 중복으로 표시되지 않음 >

# 회원 관리 - 오류 수정

```
} else {  
    MemberVO changeVO = new MemberVO();  
    changeVO.setId(changeGradeId);  
    long gradeNo = Long.parseLong(request.getParameter("grade"));  
    changeVO.setGradeNo(gradeNo);  
    Execute.run(memberChangeGradeService, changeVO);  
    jsp = "redirect:view.do?id=" + changeGradeId + "&page=" + request.getPa  
}
```



```
} else {  
    MemberVO changeVO = new MemberVO();  
    changeVO.setId(changeGradeId);  
    long gradeNo = Long.parseLong(request.getParameter("grade").substring(0, 1));  
    changeVO.setGradeNo(gradeNo);  
    Execute.run(memberChangeGradeService, changeVO);  
    jsp = "redirect:view.do?id=" + changeGradeId + "&page=" + request.getParameter("pa  
}
```

등급

2등급(운전자)

등급변경

< myPage.jsp에서 등급이 '2등급(운전자)' 형식으로 표시되어 gradeNo를 받아올 때 substring을 이용해 첫 번째 문자만 추출 >

# 회원 관리 - 오류 수정

```
if(multi.getFilesystemName("profileImage") == null) {  
    updateImageVO.setProfileImage(path + "/" + multi.getFilesystemName("profileImage"));  
} else {  
    updateImageVO.setProfileImage(path + "/" + multi.getFilesystemName("profileImage"));  
}
```

```
<input type="hidden" name="profileImage" value="chagawa.jpg">
```



```
if(multi.getFilesystemName("profileImage") == null) {  
    updateImageVO.setProfileImage(path + "/" + multi.getParameter("profileImage"));  
} else {  
    updateImageVO.setProfileImage(path + "/" + multi.getFilesystemName("profileImage"));  
}
```

< 기본 이미지로 변경 시 'chagawa.jpg' 문자를 받아와야 하므로  
getFilesystemName 대신 getParameter를 사용 >

# Part 6.

## 회원 관리 - 시연



# 회원 관리 - 개발 후기 및 차후 개발 내용

## < DBA 주해린 >

### 개발 후기

처음 프로젝트를 시작할 때  
데이터의 흐름을 잘 이해하지 못해 오류가 많이 났었는데  
값을 일일이 출력해 보고 오류를 해결해 보면서  
내가 잘 이해하지 못한 부분에 대해 알게 되고 이해하게 되었다.  
또한 팀원들의 도움도 많이 받고  
팀원들과 오류를 공유하면서 다양한 오류를 접해보게 되어  
배운 점이 많은 프로젝트였다.

### 차후 개발 내용

- 관리자 리스트에서 특정 상태, 등급의 회원만 볼 수 있는 필터 기능 추가
- 회원가입할 때 연락처 인증 기능 추가

# 공지 사항 관리 - DB 구현

업무 영역		공지 사항 관리 시스템		작성자			김소연	
테이블명		공지 사항		테이블 ID			notice	
테이블 설명		공지 사항 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
번호	no	number			O		notice_seq	
제목	title	varchar2	300	O				
내용	content	varchar2	2000	O				
공지 시작일	startDate	date					sysdate	
공지 종료일	endDate	date					‘9999-12-31’	
공지 작성일	writeDate	date					sysdate	
공지 수정일	updateDate	date					sysdate	
이미지 파일	imgFile	varchar2	100					
일반 파일	gnrFile	varchar2	100					

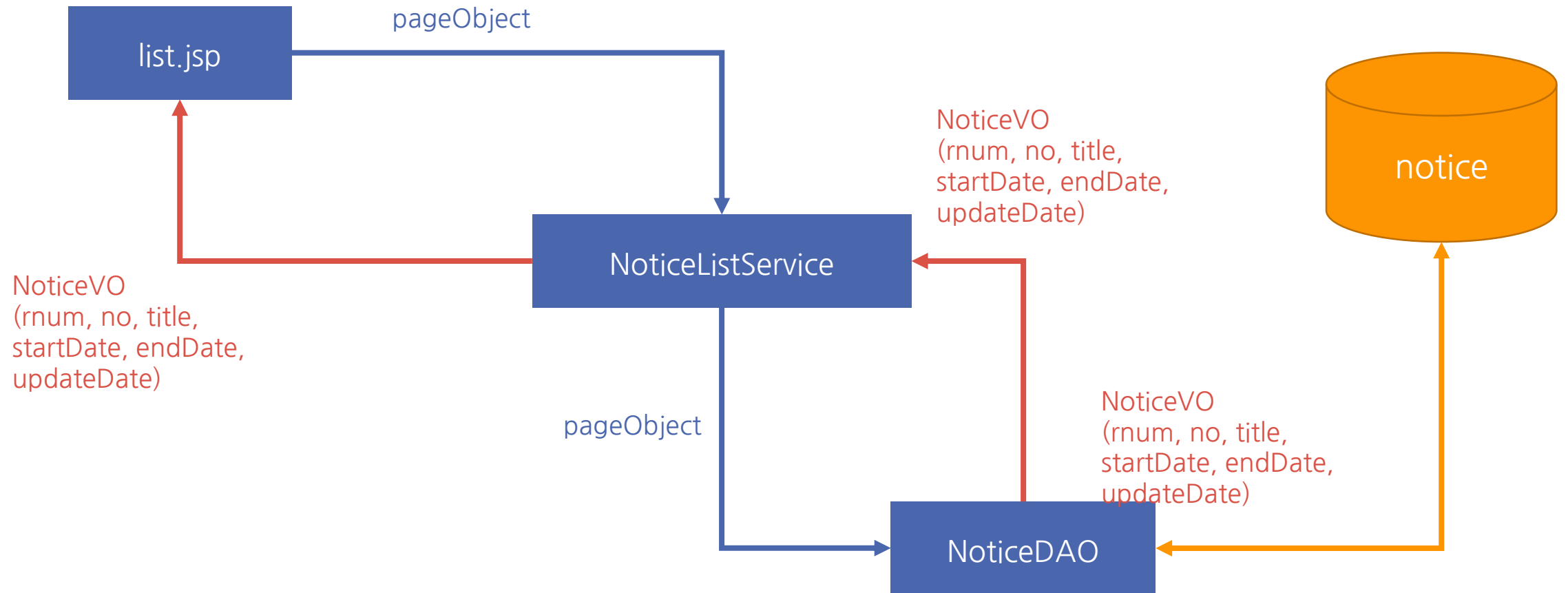
# 공지 사항 관리 - DB 구현

CHAGAWA.NOTICE		
P *	NO	NUMBER
*	TITLE	VARCHAR2 (300 BYTE)
*	CONTENT	VARCHAR2 (2000 BYTE)
	STARTDATE	DATE
	ENDDATE	DATE
	WRITEDATE	DATE
	UPDATEDATE	DATE
	GNRFILE	VARCHAR2 (100 BYTE)
	IMGFILE	VARCHAR2 (100 BYTE)
🔑 NOTICE_PK (NO)		

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	NO	NUMBER	No	(null)	1	(null)
2	TITLE	VARCHAR2 (300 BYTE)	No	(null)	2	(null)
3	CONTENT	VARCHAR2 (2000 BYTE)	No	(null)	3	(null)
4	STARTDATE	DATE	Yes	sysdate	4	(null)
5	ENDDATE	DATE	Yes	'9999-12-31'	5	(null)
6	WRITEDATE	DATE	Yes	sysdate	6	(null)
7	UPDATEDATE	DATE	Yes	sysdate	7	(null)
8	GNRFILE	VARCHAR2 (100 BYTE)	Yes	(null)	8	(null)
9	IMGFILE	VARCHAR2 (100 BYTE)	Yes	(null)	9	(null)

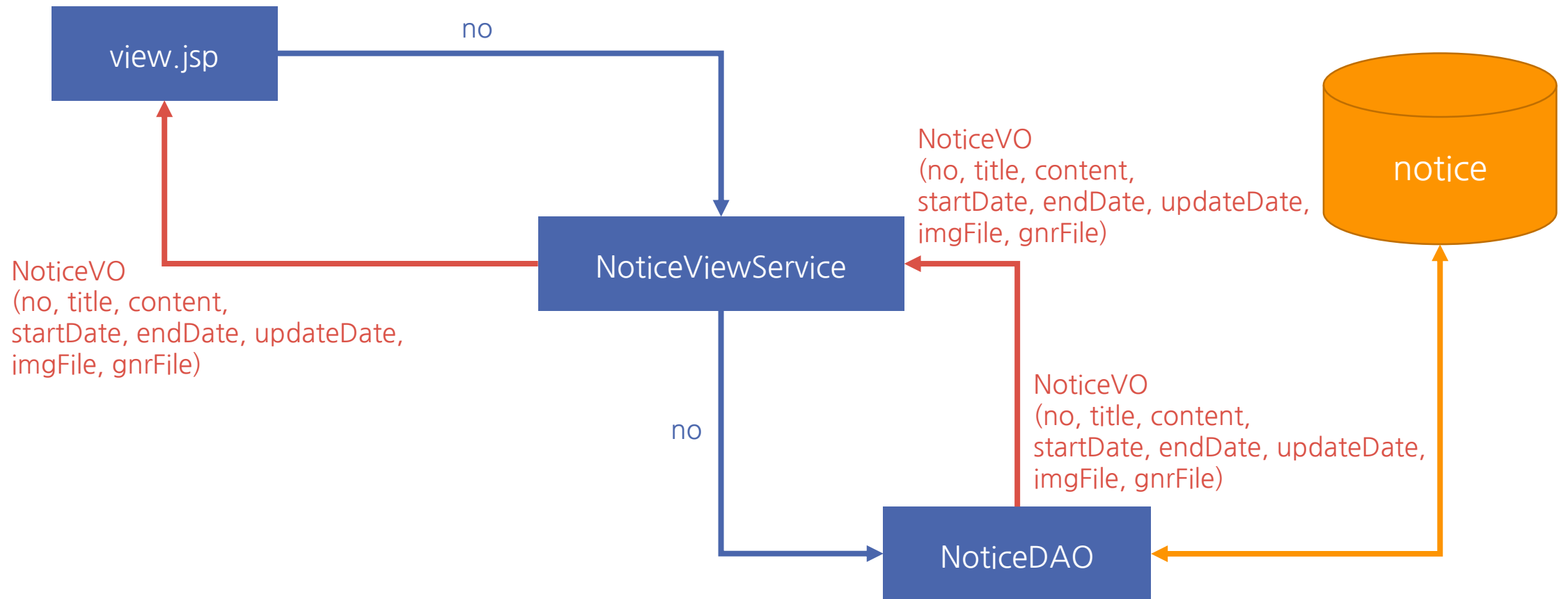
# 공지 사항 관리 - DFD

공지 리스트



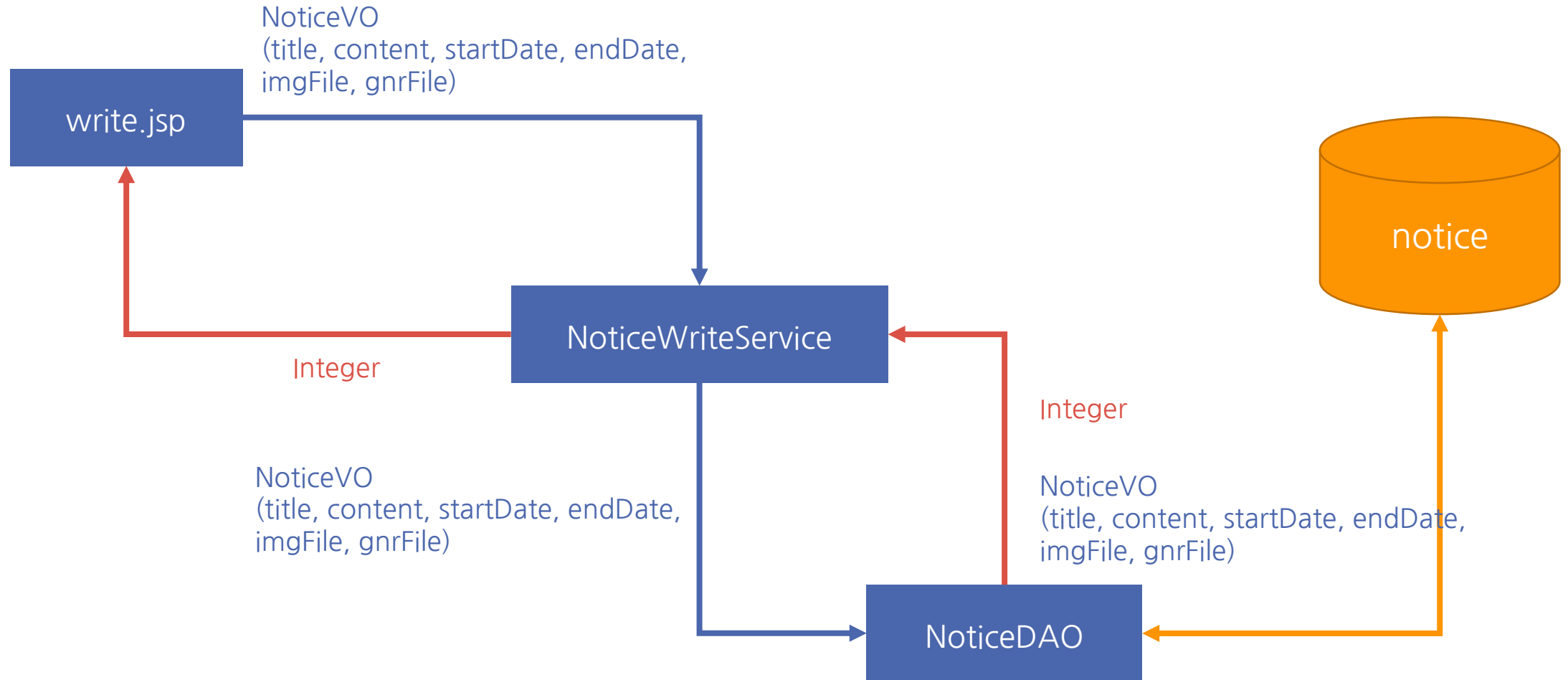
# 공지 사항 관리 - DFD

공지 보기



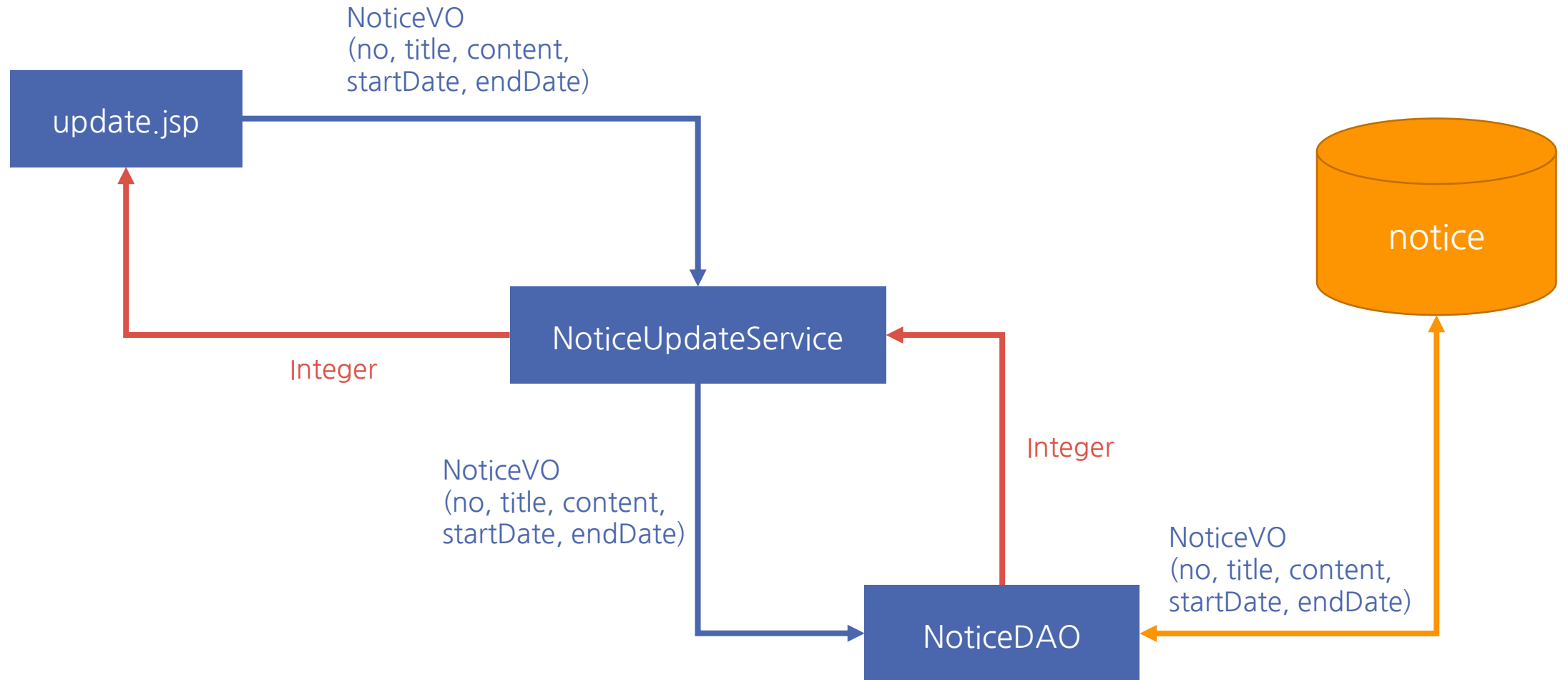
## 공지 사항 관리 - DFD

## 공지 등록

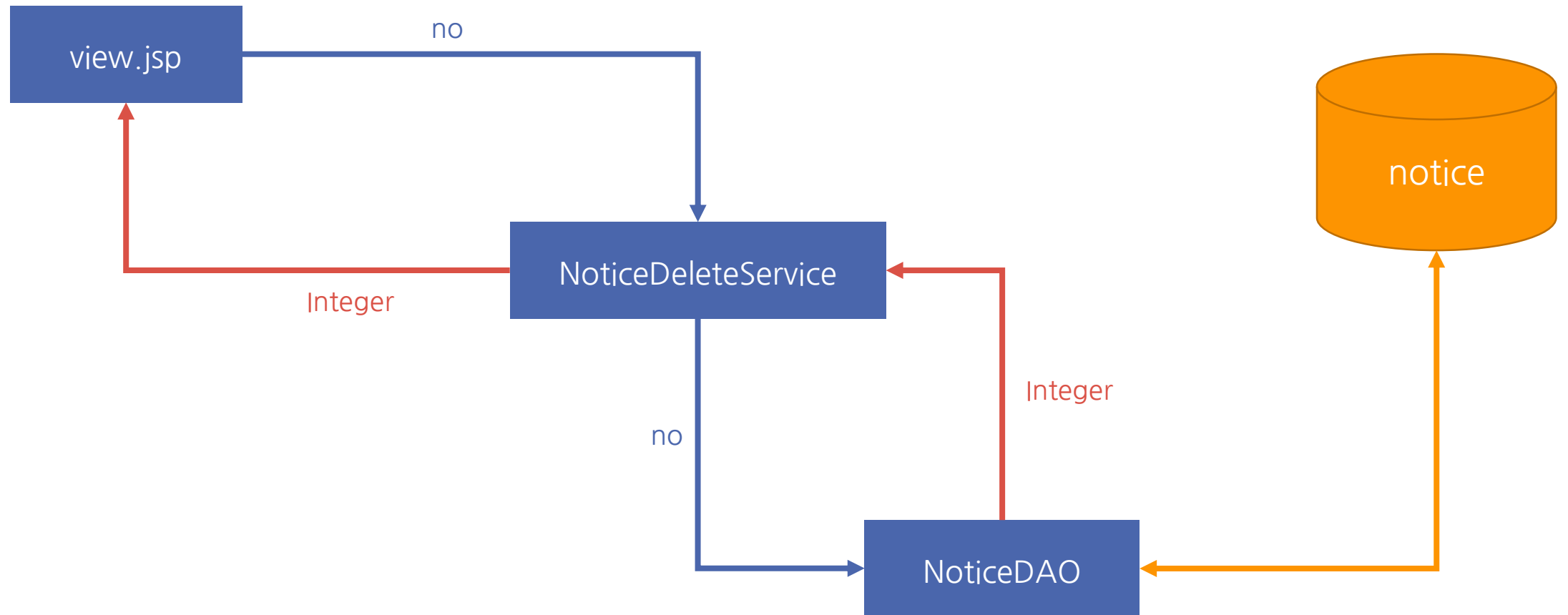


# 공지 사항 관리 - DFD

## 공지 수정



## 공지 삭제





# 공지 사항 관리 - 화면 구현

## 공지 사항

전체

게시글 개수 10

번호	제목	시작일	종료일	최종 수정일
1	현재공지제목	2023-02-15	2024-02-20	2023-03-05
4	현재공지제목	2023-02-15	2024-02-20	2023-03-05
3	현재공지제목	2023-02-15	2024-02-20	2023-03-05
2	현재공지제목	2023-02-15	2024-02-20	2023-03-05
5	시난공시제목	2022-02-15	2022-02-20	2023-03-05
1	현재공지제목	2023-02-15	2024-02-20	2023-03-05
4	현재공지제목	2023-02-15	2024-02-20	2023-03-05
3	현재공지제목	2023-02-15	2024-02-20	2023-03-05
2	현재공지제목	2023-02-15	2024-02-20	2023-03-05
11	예약공지제목	2024-02-15	2024-02-20	2023-03-05

등록

# 공지 사항 관리 - 화면 구현

## 공지 사항

번호	14
제목	당신을 위한 차가 와요.
내용	봄비는 대중교통에 지치셨나요? 차가와에서 비슷한 경로의 차를 찾아, 편안한 운행을 즐기세요.
이미지 파일	등록된 파일이 없습니다. <a href="#">파일 변경</a>
일반 파일	등록된 파일이 없습니다. <a href="#">파일 변경</a>
시작일	2022-01-01
종료일	2023-08-05
최종 수정일	2023-03-05

[수정](#)[삭제](#)[리스트](#)

## Part 7, 공지 사항 관리 - 화면 구현

번호

14

제목

당신을 위한 차가 와요. 수정

내용

봄비는 대중교통에 지치셨나요?  
차가와에서 비슷한 경로의 차를 찾아,  
편안한 운행을 즐기세요. 수정

시작일

2022-02-01



종료일

2023-09-01



번호	14
제목	당신을 위한 차가 와요. 수정
내용	봄비는 대중교통에 지치셨나요? 차가와에서 비슷한 경로의 차를 찾아, 편안한 운행을 즐기세요. 수정
이미지 파일	
일반 파일	
시작일	2022-02-01
종료일	2023-09-01
최종 수정일	2023-03-05

# Part 7, 공지 사항 관리 - 화면 구현

## 공지 사항

전체 ▾

Search

Q

게시글 개수 10 ▾

번호	제목	시작일	종료일	최종 수정일
6	지난공지제목	2022-02-15	2022-02-20	2023-03-05
8	지난공지제목	2022-02-15	2022-02-20	2023-03-05
7	지난공지제목	2022-02-15	2022-02-20	2023-03-05
5	지난공지제목	2022-02-15	2022-02-20	2023-03-05
4	현재공지제목	2023-02-15	2024-02-20	2023-03-05
3	현재공지제목	2023-02-15	2024-02-20	2023-03-05
2	현재공지제목	2023-02-15	2024-02-20	2023-03-05
1	현재공지제목	2023-02-15	2024-02-20	2023-03-05
9	예약공지제목	2024-02-15	2024-02-20	2023-03-05
12	예약공지제목	2024-02-15	2024-02-20	2023-03-05

등록

# 공지 사항 관리 - 화면 구현

 차가와

공지사항

카풀

이용후기

메시지

회원리스트

수수료 수입

로그아웃

## 공지 사항

제목

공지 사항

localhost 내용:

제목은 4~100자 이내로 작성하셔야 합니다.

확인

localhost 내용:

내용은 4~670자 이내로 작성하셔야 합니다.

확인

시작일

이미지 파일

연도-월-일



파일 선택 dan-gold-N

일반 파일

파일 선택 test.txt



이 입력란을 작성하세요.

시작일

2022-01-01



종료일

2023-03-02




초기화

취소


등록

## Part 7, 공지 사항 관리 - 화면 구현

 차가와

공지사항카풀이용후기메시지회원리스트수수료 수입로그아웃

공지 사항

번호	15
제목	공지 사항
내용	공지 사항
이미지 파일	<div></div> <div>파일 변경</div>
일반 파일	<div>test31.txt</div> <div>파일 변경</div>
시작일	2022-01-01
종료일	2023-03-02
최종 수정일	2023-03-05

수정

삭제

리스트

```
// 리스트
case "/notice/list.do":
    PageObject pageObject = PageObject.getInstance(request);
    LoginVO login = (LoginVO) request.getSession().getAttribute("login");
    // 비회원과 일반 회원은 현재 공지 사항만 보임
    if (login == null || login.getGradeNo() == 1 || login.getGradeNo() == 2) {
        request.setAttribute("list", Execute.run(currentNoticeListService, pageObject));
        // 관리자는 전체 공지 사항 보임
    } else if (login.getGradeNo() == 9) {
        request.setAttribute("list", Execute.run(noticeListService, pageObject));
    }
    request.setAttribute("pageObject", pageObject);
    jsp = "notice/list";
    break;
```

< 비회원과 일반 회원은 현재 공지 사항만,  
관리자는 전체 공지 사항을 볼 수 있게 하는 코드 >

# 공지 사항 관리 - 핵심 코드

```
<td><c:choose>
  <c:when test="${vo.gnrFile == '/upload/notice/null'}">
    <p style="font-size: 10px; color: grey;">등록된 파일이 없습니다.</p>
  </c:when>
  <c:otherwise>
    <a href="${vo.gnrFile}" download>${vo.gnrFile.substring(15)}</a>
  </c:otherwise>
</td>
```

< 파일을 업로드하면 파일명의 15번째 글자부터 추출해 다운로드 링크를 생성하고,  
파일을 업로드하지 않으면 '등록된 파일이 없습니다.'라는 메시지를 출력 >



# 공지 사항 관리 - 오류 수정

```
요청한 페이지: /notice/list.do
NoticeController.execute().request - org.apache.catalina.connector.Request
NoticeController.execute().method - GET
PageObject.getInstance() [pageObject = PageObject [page=1, perPageNum=10,
Execute.run() : 실행되는 서비스 - NoticeListServiceImpl
Execute.run() : 전달되는 데이터 - PageObject [page=1, perPageNum=10, startRow=
java.sql.SQLException: ORA-00942: table or view does not exist

at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:440)
at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:396)
```



```
요청한 페이지: /notice/list.do
NoticeController.execute().request - org.apache.catalina.connector.RequestFacade@4183ec
NoticeController.execute().method - GET
PageObject.getInstance() [pageObject = PageObject [page=1, perPageNum=10, startRow=0, en
Execute.run() : 실행되는 서비스 - NoticeListServiceImpl
Execute.run() : 전달되는 데이터 - PageObject [page=1, perPageNum=10, startRow=0, endRow=0,
Execute.run() : 결과 데이터 - PageObject [page=1, perPageNum=10, startRow=1, endRow=10, pe
```

< 오타자로 인한 오류 발생 >

# 포인트 관리 - DB 구현

업무 영역		포인트 관리 시스템		작성자			김소연	
테이블명		포인트		테이블 ID			point	
테이블 설명		포인트 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
번호	no	number			O		point_seq	
아이디	id	varchar2	20	O		O	member.id	
유형	memo	varchar2	300	O				
일자	runDate	date		O			sysdate	
금액	price	number	10	O				

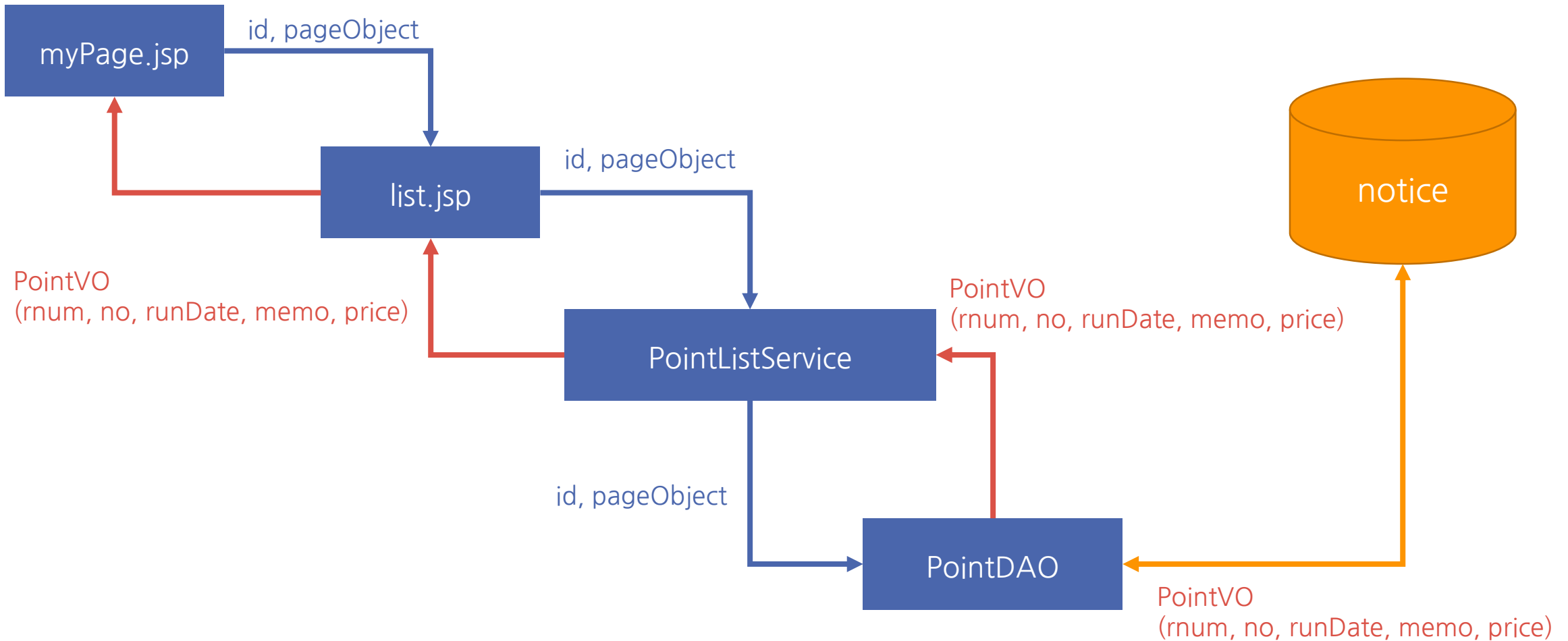
# 포인트 관리 - DB 구현

CHAGAWA.POINT		
P	* NO	NUMBER
	* ID	VARCHAR2 (20 BYTE)
	* MEMO	VARCHAR2 (300 BYTE)
	RUNDATE	DATE
	* PRICE	NUMBER (10)
POINT_PK (NO)		

	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1	NO	NUMBER	No	(null)	1	(null)
2	ID	VARCHAR2 (20 BYTE)	No	(null)	2	(null)
3	MEMO	VARCHAR2 (300 BYTE)	No	(null)	3	(null)
4	RUNDATE	DATE	Yes	sysdate	4	(null)
5	PRICE	NUMBER (10, 0)	No	(null)	5	(null)

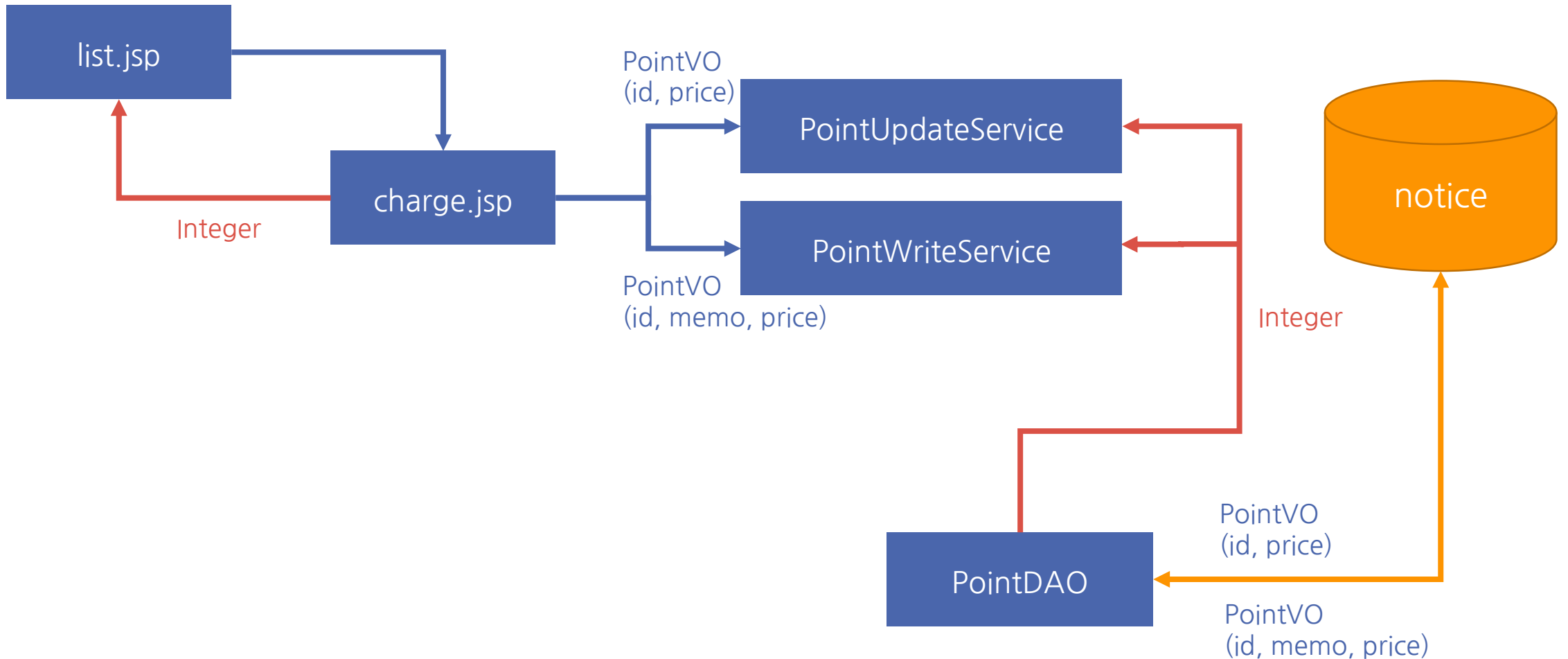
# 포인트 관리 - DFD

포인트/수수료 수입 내역



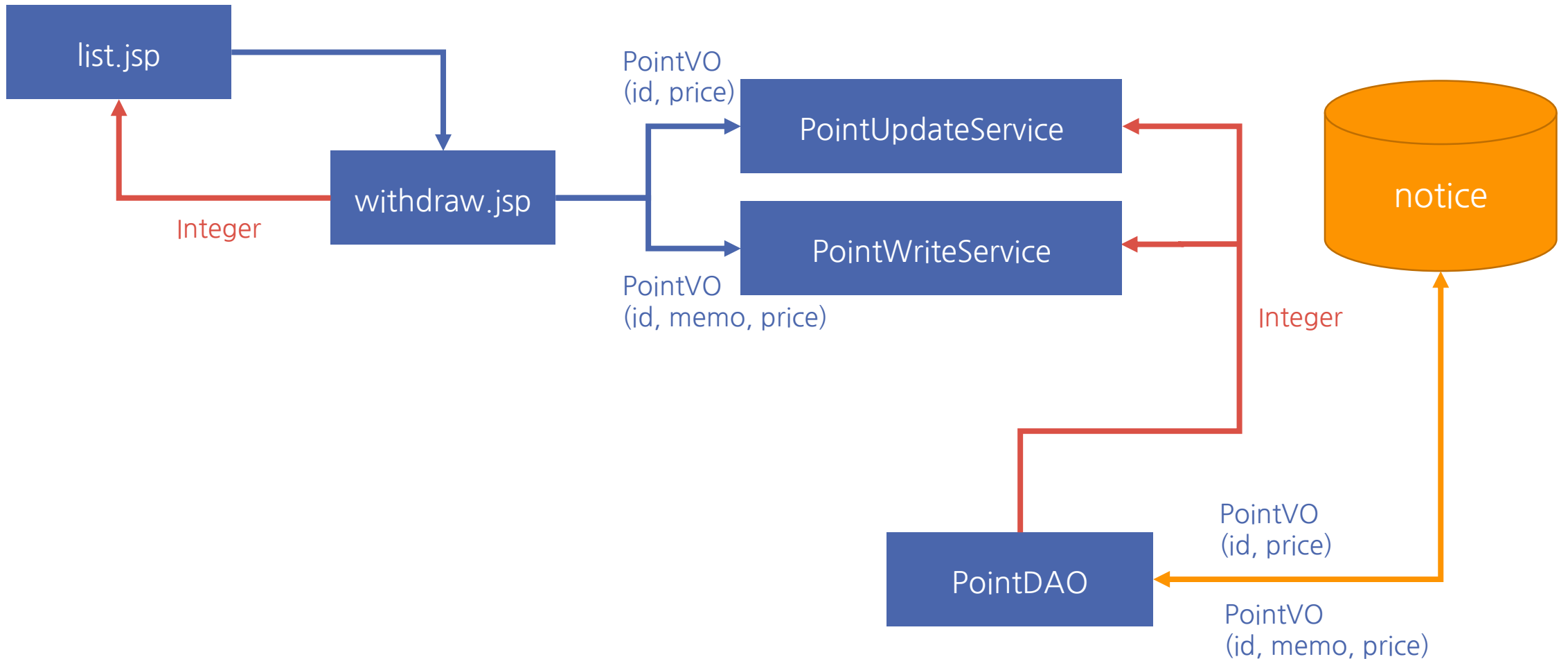
## 포인트 관리 - DFD

포인트 충전



## 포인트 관리 - DFD

포인트 인출



# 포인트 관리 - 화면 구현

차가와

이용중인 카풀공지사항카풀이용후기메시지마이페이지로그아웃

포인트 충전 - Chrome

localhost/point/charge.sub

## 포인트 충전

☒ 5,000p (6,000원)  
☐ 10,000p (12,000원)  
☐ 20,000p (24,000원)

카드 선택: 

===선택===

카드 번호:  -  -  -

충전하기

닫기

# 포인트 관리 - 화면 구현



차가와

공지사항

카풀

이용후기

메시지

회원리스트

수수료 수입

로그아웃

## 수수료 수입 내역

총수입 - 118000원

번호	일자	유형	금액(원)
2	2023-03-06 00:03:38	포인트 충전 수수료	1000





# 포인트 관리 - 화면 구현

차가와

차가와

이용중인 카플    공지사항    카플    이용후기    .메시지    마이페이지    로그아웃    로그아웃

포인

lo

번

인트

내 정보 수정

아이디: driver

닉네임: driver 중복확인

이름: 운전자

성별: 남자

생년월일: 23.01.01

면적차: 010 0000 0001 중복확인

계좌: 국민 0000-000-000001

등급: 운전자

가입일: 23.02.26

차종: 미니쿠퍼

차량번호: 123가1234 중복확인

면허번호: 12-34-567890-11

수정    새로입력    취소

# 포인트 관리 - 화면 구현

 차가와

이용중인 카풀

공지사항

카풀

이용후기

메시지

마이페이지

로그아웃

## 포인트 내역

포인트 충전

포인트 인출

보유 포인트 - 16658p

번호	일자	유형	포인트
3	2023-03-06 00:24:15	포인트 인출	-10000
1	2023-03-06 00:03:38	포인트 충전	5000

```
// 관리자 운영 수익 증가 후 내역 등록
PointVO writeVOadmin = new PointVO();
writeVOadmin.setId("admin");
writeVOadmin.setMemo(request.getParameter("memo") + " 수수료");
writeVOadmin.setPrice(Long.parseLong(request.getParameter("point")) * 2 / 10);
Execute.run(pointWriteService, writeVOadmin);
```

< 충전 금액의 20%를 관리자 계정에 추가해 주는 코드 >

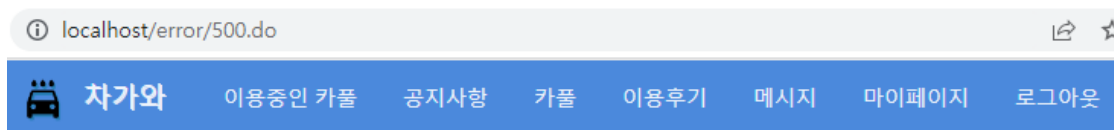
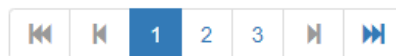
```
} else {  
    // myPoint < withdrawPoint 경우 메시지  
    MemberVO memberVO3 = (MemberVO) Execute.run(memberMyPagePService, id);  
    request.setAttribute("bank", memberVO3.getBank());  
    request.setAttribute("account", memberVO3.getAccount());  
    request.setAttribute("myPoint", memberVO3.getMyPoint());  
    request.setAttribute("fail", 1);  
    jsp = "point/withdraw";  
}
```

```
<c:if test="${fail==1 }">  
    <script type="text/javascript">  
        alert('보유 포인트가 부족합니다.')  
    </script>  
</c:if>
```

< 입력된 인출 포인트보다 보유 포인트가 적을 경우 알림창 >

# 포인트 관리 - 오류 수정

6	2023-03-06 00:28:51	포인트 충전	5000
4	2023-03-06 00:28:47	포인트 충전	5000



## 서버 실행 오류

다시 한번 시도해 보세요.  
계속 오류가 발생되면 전산담당자에게 연락주세요.

< 페이지 이동 시 오류 발생 >

# 포인트 관리 - 오류 수정

```
pageNav.tag ×
23
24 <ul class="pagination">
25   <li data-page=1><c:if test="${pageObject.page > 1 }">
26     <!-- 맨 첫페이지로 이동 -->
27     <a
28       href="${listURI }?page=1&perPageNum=${pageObject.perPageNum}&key=${pageObject.key}&word=${pageObject.word}${query}">
29       <i class="glyphicon glyphicon-fast-backward"></i>
30     </a>
31   </c:if> <c:if test="${pageObject.page == 1 }">

pageNavPoint.tag ×
24 <ul class="pagination">
25   <li data-page=1><c:if test="${pageObject.page > 1 }">
26     <!-- 맨 첫페이지로 이동 -->
27     <a
28       href="${listURI }&page=1&perPageNum=${pageObject.perPageNum}&key=${pageObject.key}&word=${pageObject.word}${query}">
29       <i class="glyphicon glyphicon-fast-backward"></i>
30     </a>
31   </c:if> <c:if test="${pageObject.page == 1 }">

<div class="row text-center">
  <pageNav:pageNavPoint listURI="list.do?id=${login.id }" pageObject="${pageObject }" />
</div>
```

< 페이지네이션 태그 수정 후 아이디를 전달해 오류 수정 >

# Part 8.

공지 사항 / 포인트 관리 - 시연

# 공지 사항/포인트 관리 - 개발 후기 및 차후 개발 내용

< 서기 김소연 >

## 개발 후기

공지 사항 시스템은 이전에 알고 있던 CRUD 기능을 활용하여 비교적 수월하게 개발할 수 있었지만, 포인트 시스템은 형식이 낯설어 감을 잡는 데 많은 시간이 걸렸다.

팀원들에게 자주 질문하면서 이해를 돕고자 노력했는데, 팀원들이 항상 차근차근 알려 주셔서 무사히 마무리할 수 있었던 것 같다.

이번 프로젝트를 통해 처리 흐름에 대한 이해가 부족하다는 것을 깨달았고, 기초적인 지식을 더욱 쌓고 싶다는 생각이 들었다.

## 차후 개발 내용

수정 중

공지 사항을 현재 공지, 지난 공지, 예약 공지, 전체 공지로 분류해 주는 버튼 만들기



# 카풀 관리 - DB 구현

업무 영역		카풀 관리 시스템		작성자			정하영	
테이블명		카풀		테이블 ID			carpool	
테이블 설명		카풀 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
번호	no	number			O		carpool_seq	
등록일시	regTime	date		O			sysdate	
드라이버아이디	driverId	varchar2	30	O		O	member(id)	
출발지역	startRegion	varchar2	100	O				
출발상세주소	startAddress	number		O				
출발위도	startLat	number		O				
출발경도	startlng	number		O				
도착지역	arriveRegion	number		O				
도착상세주소	arriveAddress	number		O				
도착위도	arriveLat	number		O				
도착경도	arriveLng	number		O				

# 카풀 관리 - DB 구현

출발예정시각	startTimePre	date		O			
도착예정시간	arriveTimePre	date		O			
출발시각	startTime	date					
도착시각	arriveTime	date					
상태	status	varchar2	30	O			'모집중','출발대기','운행중','도착','종료','취소'
동승료	price	number	10	O			
최대동승인원	seats	number	2	O			
특이사항	memo	varchar2	500	O			
후기여부	reviewed	number	2	O			default 0
별점여부	star	number	2	O			default 0

# 카풀 관리 - DB 구현

업무 영역		카풀 관리 시스템		작성자			정하영
테이블명		동승자		테이블 ID			passenger
테이블 설명		카풀 동승자 관리에 필요한 데이터					
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고
카풀번호	cpNo	number			O		carpool(no)
동승자아이디	psgId	varchar2	20		O	O	member(id)
신청메시지	msg	varchar2	500				
신청일시	appTime	date		O			sysdate
상태	status	varchar2	30	O			‘신청’, ‘확정’, ‘탑승’, ‘도착’, ‘종료’, ‘미탑승’, ‘취소’
확정여부	fixed	number	2	O			default 0
탑승여부	gotIn	number	2	O			default 0
도착확인여부	arrived	number	2	O			default 0
후기여부	reviewed	number	2	O			default 0
별점여부	star	number	2	O			default 0

## 카풀 관리 - DB 구현

CHAGAWA.PASSENGER		
PF*	CPNO	NUMBER
P*	PSGID	VARCHAR2 (20 BYTE)
	MSG	VARCHAR2 (500 BYTE)
*	APPTIME	DATE
*	STATUS	VARCHAR2 (30 BYTE)
*	FIXED	NUMBER (2)
*	GOTIN	NUMBER (2)
*	ARRIVED	NUMBER (2)
*	REVIEWED	NUMBER (2)
*	STAR	NUMBER (2)
PASSENGER_PK (CPNO, PSGID)		
SYS_C0010853 (CPNO)		

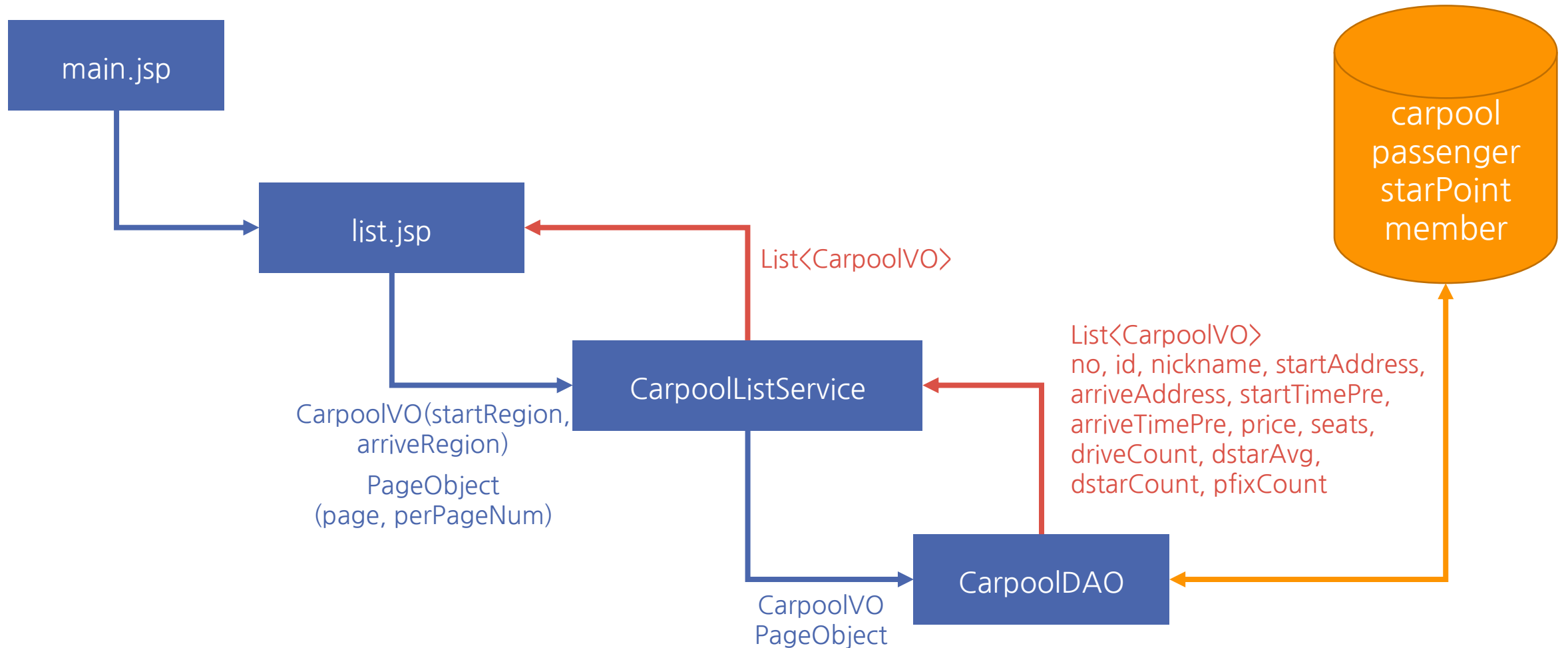
CHAGAWA.CARPOOL		
P*	NO	NUMBER
*	REGTIME	DATE
*	DRIVERID	VARCHAR2 (20 BYTE)
*	STARTREGION	VARCHAR2 (100 BYTE)
*	STARTADDRESS	VARCHAR2 (100 BYTE)
*	STARTLAT	NUMBER
*	STARTLNG	NUMBER
*	ARRIVELAT	NUMBER
*	ARRIVELNG	NUMBER
*	ARRIVEREGION	VARCHAR2 (100 BYTE)
*	ARRIVEADDRESS	VARCHAR2 (100 BYTE)
*	STARTTIMEPRE	DATE
*	ARRIVETIMEPRE	DATE
	STARTTIME	DATE
	ARRIVETIME	DATE
*	STATUS	VARCHAR2 (30 BYTE)
*	PRICE	NUMBER (10)
*	SEATS	NUMBER (2)
	MEMO	VARCHAR2 (500 BYTE)
*	REVIEWED	NUMBER (2)
*	STAR	NUMBER (2)
CARPOOL_PK (NO)		

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CPNO	NUMBER	No	(null)	1	(null)
2	PSGID	VARCHAR2 (20 BYTE)	No	(null)	2	(null)
3	MSG	VARCHAR2 (500 BYTE)	Yes	(null)	3	(null)
4	APPTIME	DATE	No	sysdate	4	(null)
5	STATUS	VARCHAR2 (30 BYTE)	No	'신청'	5	(null)
6	FIXED	NUMBER (2, 0)	No	0	6	(null)
7	GOTIN	NUMBER (2, 0)	No	0	7	(null)
8	ARRIVED	NUMBER (2, 0)	No	0	8	(null)
9	REVIEWED	NUMBER (2, 0)	No	0	9	(null)
10	STAR	NUMBER (2, 0)	No	0	10	(null)

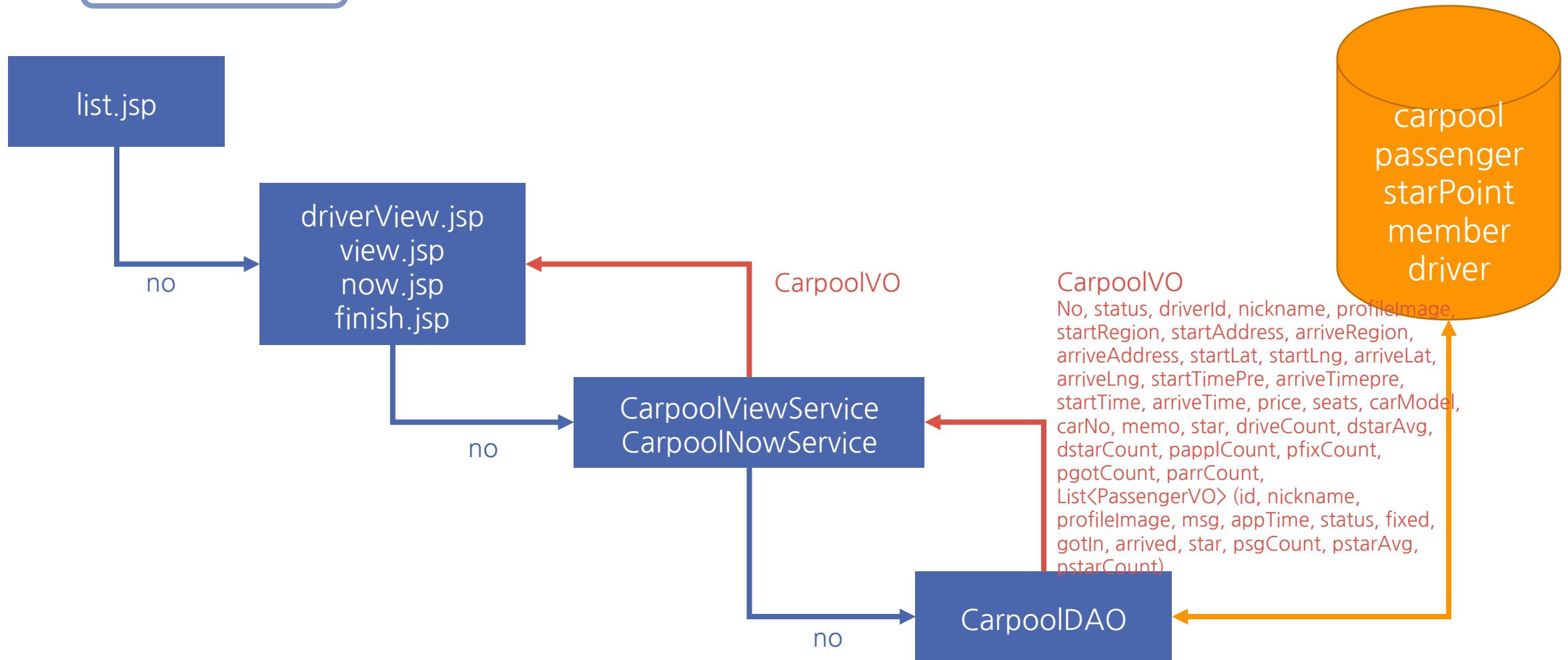
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	NO	NUMBER	No	(null)	1	(null)
2	REGTIME	DATE	No	sysdate	2	(null)
3	DRIVERID	VARCHAR2 (20 BYTE)	No	(null)	3	(null)
4	STARTREGION	VARCHAR2 (100 BYTE)	No	(null)	4	(null)
5	STARTADDRESS	VARCHAR2 (100 BYTE)	No	(null)	5	(null)
6	STARTLAT	NUMBER	No	(null)	6	(null)
7	STARTLNG	NUMBER	No	(null)	7	(null)
8	ARRIVELAT	NUMBER	No	(null)	8	(null)
9	ARRIVELNG	NUMBER	No	(null)	9	(null)
10	ARRIVEREGION	VARCHAR2 (100 BYTE)	No	(null)	10	(null)
11	ARRIVEADDRESS	VARCHAR2 (100 BYTE)	No	(null)	11	(null)
12	STARTTIMEPRE	DATE	No	(null)	12	(null)
13	ARRIVETIMEPRE	DATE	No	(null)	13	(null)
14	STARTTIME	DATE	Yes	(null)	14	(null)
15	ARRIVETIME	DATE	Yes	(null)	15	(null)
16	STATUS	VARCHAR2 (30 BYTE)	No	'모집중'	16	(null)
17	PRICE	NUMBER (10, 0)	No	(null)	17	(null)
18	SEATS	NUMBER (2, 0)	No	(null)	18	(null)
19	MEMO	VARCHAR2 (500 BYTE)	Yes	(null)	19	(null)
20	REVIEWED	NUMBER (2, 0)	No	0	20	(null)
21	STAR	NUMBER (2, 0)	No	0	21	(null)

## 카풀 관리 - DFD

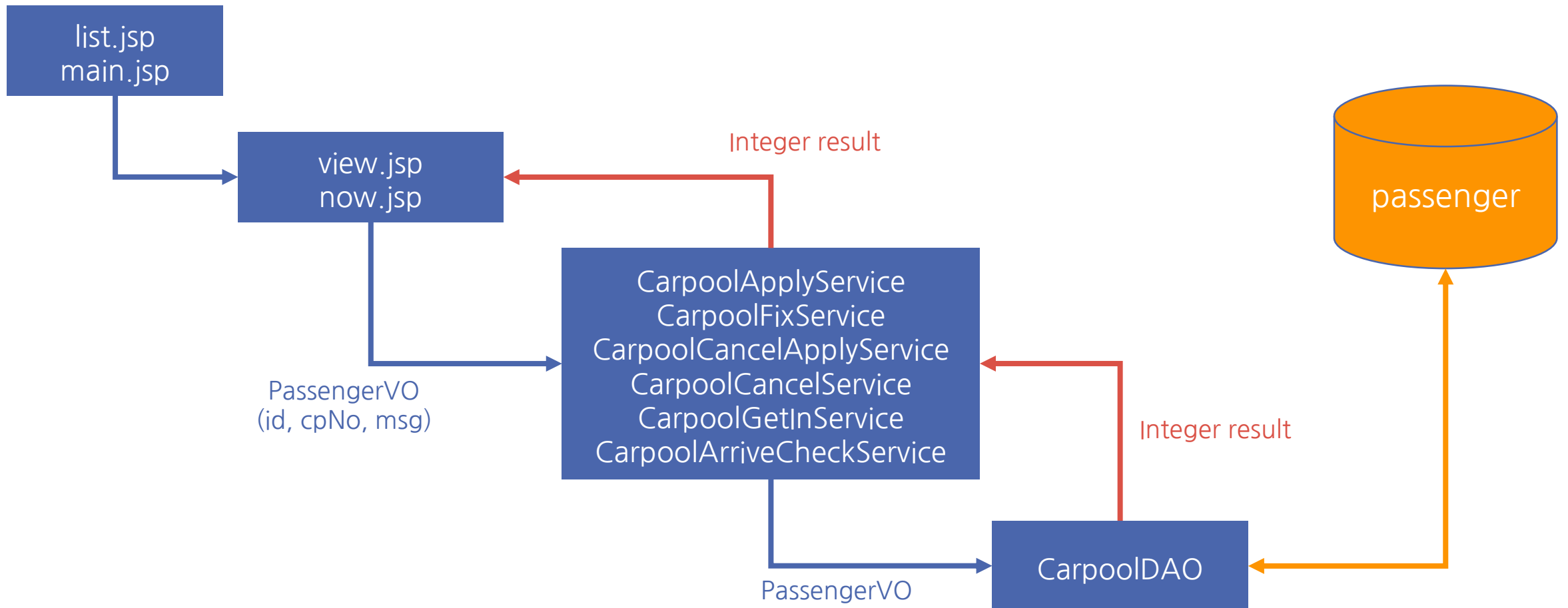
## 카풀리스트



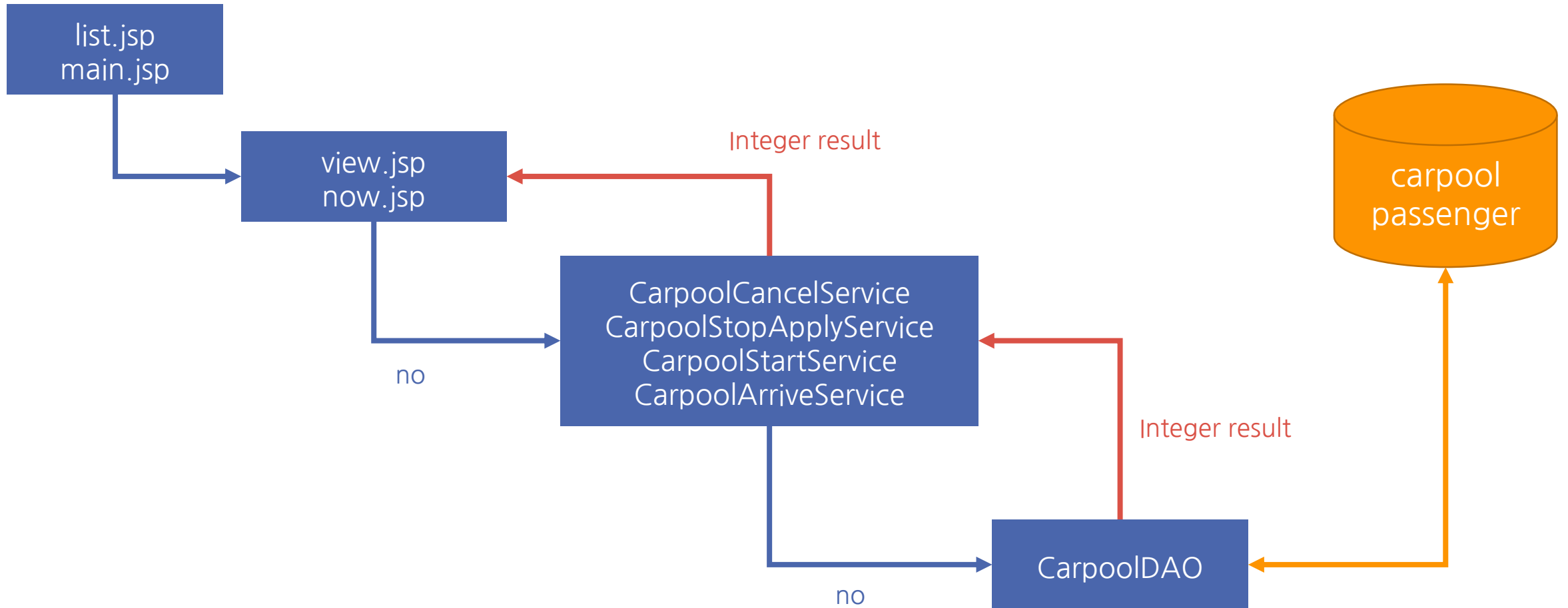
## 카풀 상세보기



## 카풀 운영

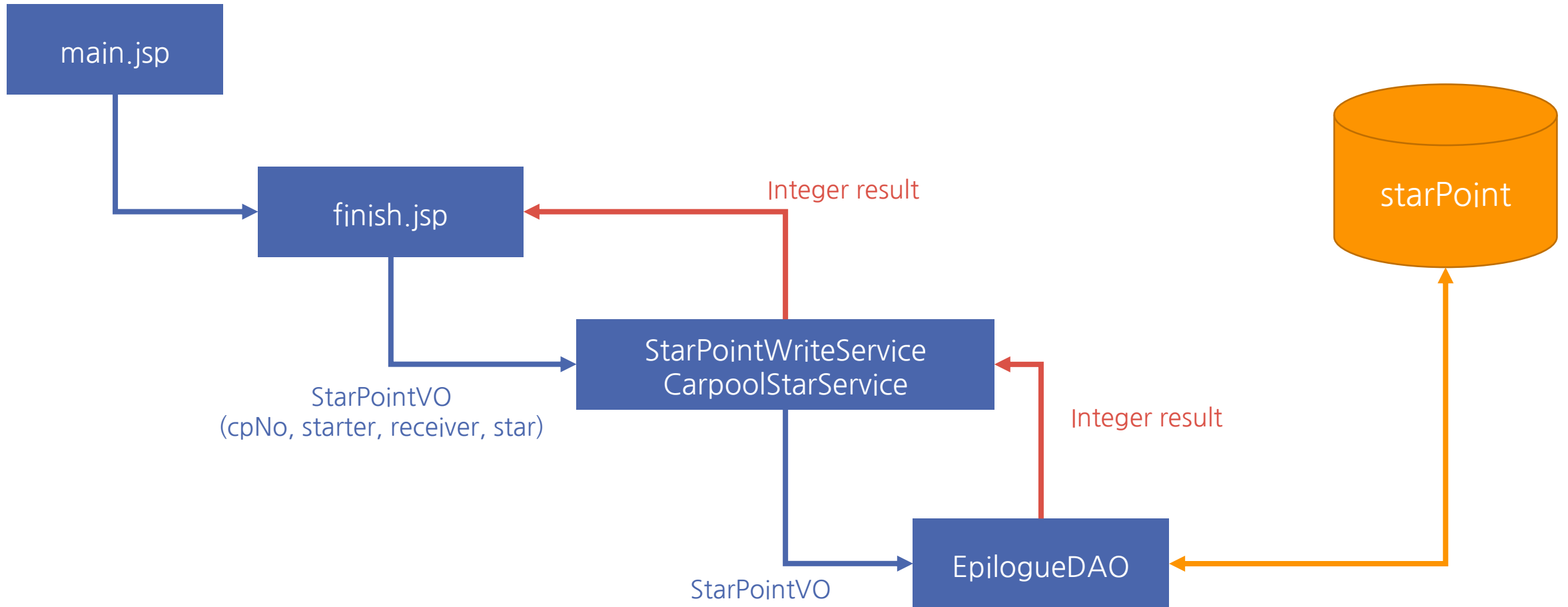


## 카풀 운영

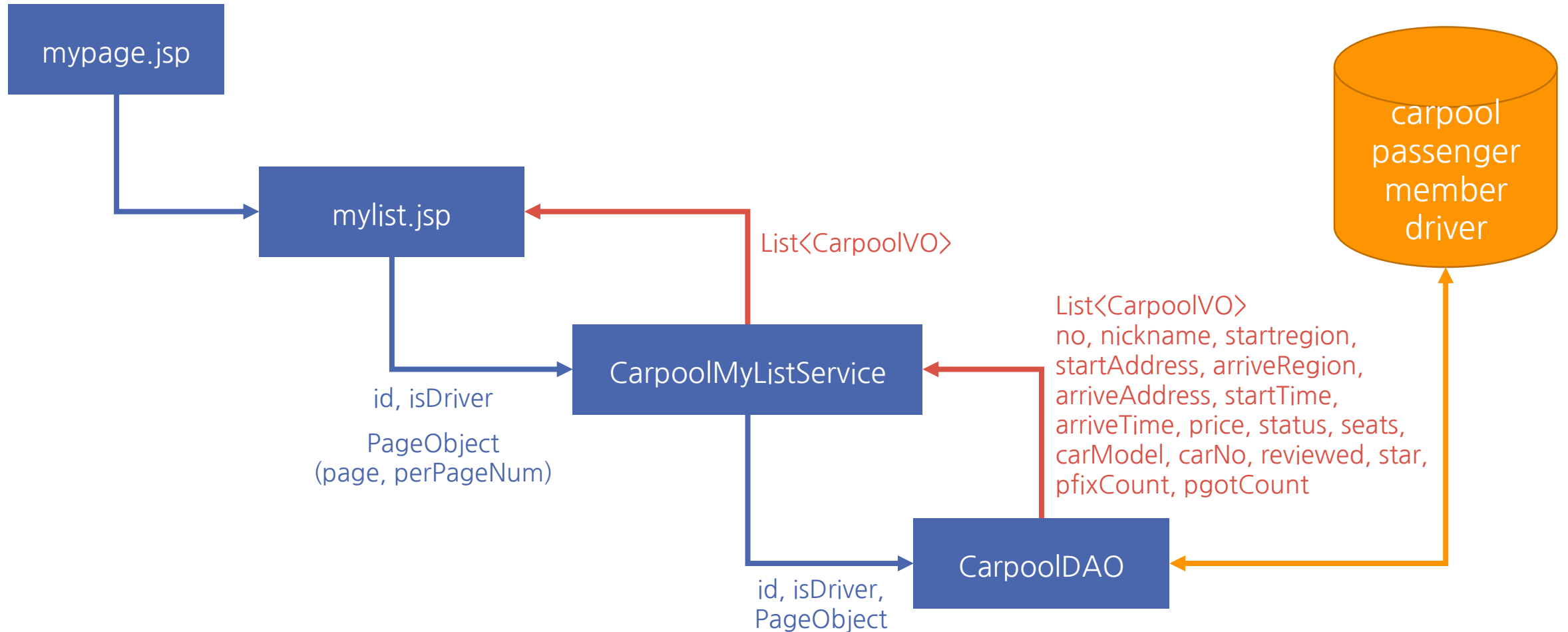




## 카풀운영



내 이용내역



# 카풀 관리 - 화면 구현 [카풀 리스트]

**차가와**

이용중인 카풀

공지사항

카풀

이용후기

메시지 0

마이페이지

로그아웃

## 카풀 리스트

출발

도착

검색

경기 의정부시

경기 의정부시

3번 차

0/4

회룡역 ▶ 의정부율지대병원

2023-03-03 17:50:00 출발 예정 | 2023-03-03 18:10:00 도착 예정

드라이버 어제면허탐 | 0회 운행 | 평균별점 0.0 [0회 참여]

동승료 900 포인트

2번 차

0/2

서초등학교 ▶ 의정부소방서 앞

2023-03-03 15:50:00 출발 예정 | 2023-03-03 15:55:00 도착 예정

드라이버 전직택시운전사 | 0회 운행 | 평균별점 0.0 [0회 참여]

동승료 1000 포인트

1번 차

1/3

의정부시청 ▶ 의정부역

2023-02-17 08:50:00 출발 예정 | 2023-02-17 09:15:00 도착 예정

드라이버 driver | 0회 운행 | 평균별점 0.0 [0회 참여]

동승료 500 포인트

카풀 등록

회원등급 2일 경우 나타남

write.d  
o

⏮

⏪

1

⏩

⏭

비회원인 경우

이 차에 함께 타고 싶다면 로그인하세요

동승을 신청한 경우

이 차에 통승을 신청했어요 cancelapply.do

아래 명단에서 확정여부를 확인하세요.

신청취소

드라이버에게 문의할 내용이 있나요?

## 메시지 보내기

/messenger/write.do

### 포인트가 부족한 경우

localhost 내용:

보유 포인트가 부족합니다. 마이페이지에서 포인트를 충전해 주세요.

확인



# 카풀 관리 - 화면 구현 [카풀 운영-출발대기]

**차가와**

이용중인 카풀    공지사항    카풀    이용후기    메시지 0    마이페이지    로그아웃

## 이용 중인 카풀

### 1번차-운행정보 출발대기

[경기 의정부시] 의정부시청 → [경기 의정부시] 의정부역  
2023-02-17 08:50:00 출발예정 - 2023-02-17 09:15:00 도착예정

**드라이버 driver**  
 0회 운행 | 평균 별점 0.0 (0회 참여)

메시지 보내기

드라이버 메시지  
 ▶ 조용히 가실 분

차량정보 미니쿠퍼 123가1234  
 동승료 500 포인트

출발하기

종료취소

start.do    cancel.do

1/3 동승자 탑승현황

태워주세요
 

탑승완료

탑승하기를 누른 동승자

의정부통곡리
 

탑승하기

동승자 본인일 경우 나타남  
 - 누를 경우 탑승완료 배지로 변경됨

어제면허땀
 

탑승대기

탑승하기를 아직 누르지 않은 동승자

출발하기

종료취소

start.do    cancel.do

드라이버일 경우 나타남



# Part 9, 카풀 관리 - 화면 구현 [카풀 운영-운행중]

 **차가와**

이용중인 카풀   공지사항   카풀   이용후기   메시지 0   마이페이지   로그아웃

## 이용 중인 카풀

### 1번차-운행정보

운행중

[경기 의정부시] 의정부시청 → [경기 의정부시] 의정부역  
2023-02-17 08:50:00 출발예정 - 2023-02-17 09:15:00 도착예정



의정부시청   의정부역

### 2 동승자 목록

 태워주세요

 의정부통근러

 **드라이버 driver**  
0회 운행 | 평균별점 0.0 (0회 참여)

드라이버 메시지

▶ 조용히 가실 분

차량정보 미니쿠퍼 123가1234  
동승료 500 포인트

출발시간 : 2023-03-03 17:14:10

도착하기

드라이버일 경우 나타남

arrive.do

# 카풀 관리 - 화면 구현 [카풀 운영-도착/종료(드라이버)]

**차가와**

이용중인 카풀    공지사항    카풀    이용후기    메시지 0    마이페이지    로그아웃

## 운행이 종료되었어요

### 1번차-운행정보 도착

[경기 의정부시] 의정부시청 → [경기 의정부시] 의정부역  
2023-02-17 08:50:00 출발예정 - 2023-02-17 09:15:00 도착예정

드라이버 **driver**  
 0회 운행 | 평균별점 0.0 (0회 참여)

드라이버 메시지  
 ▶ 조용히 가실 분

차량정보 미니쿠퍼 123가1234  
 동승료 500 포인트  
 예정운행시간 2023-02-17 08:50:00 출발 ▶ 2023-02-17 09:15:00 도착  
 실제운행시간 2023-03-03 17:14:10 출발 ▶ 2023-03-03 17:19:21 도착

### 2 동승자 목록

태워주세요 도착확인대기

★★★★★

의정부통근러 도착확인대기

★★★★★  
 ★★★★★☆  
 ★★★★★☆  
 ★★★★★☆  
 ★★★★★☆  
 ★★★★★☆

별점 등록  
 star.do

arrive.do에서 도착 처리 후  
finish.do로 이동

드라이버일 경우

- 각 동승자에 별점 평가



# 카풀 관리 - 화면 구현 [카풀 운영-도착(동승자)]

**차가와**

이용중인 카풀    공지사항    카풀    이용후기    메시지 0    마이페이지    로그아웃

## 이용 중인 카풀

### 1번차-운행정보 도착

[경기 의정부시] 의정부시청 → [경기 의정부시] 의정부역  
2023-02-17 08:50:00 출발예정 - 2023-02-17 09:15:00 도착예정

**드라이버 driver**  
 0회 운행 | 평균별점 0.0 (0회 참여)

메시지 보내기

**드라이버 메시지**  
 ▶ 조용히 가실 분

차량정보 미니쿠퍼 123가1234  
 동승료 500 포인트

출발시간 : 2023-03-03 17:14:10  
 도착시간 : 2023-03-03 17:19:21

### 3 동승자 목록

태워주세요 도착확인완료

의정부통근러 도착확인 arrive.do

어제면허땀 도착확인대기

도착확인을 누른 동승자

동승자 본인일 경우 나타남

도착확인을 아직 누르지 않은 동승자

# 카풀 관리 - 화면 구현 [카풀 운영-종료(동승자)]

arrive.do에서 도착 처리 후  
finish.do로 이동

차가와

이용중인 카풀    공지사항    카풀    이용후기    메시지 0    마이페이지    로그아웃

## 운행이 종료되었어요

### 1번차-운행정보 도착

[경기 의정부시] 의정부시청 → [경기 의정부시] 의정부역  
2023-02-17 08:50:00 출발예정 - 2023-02-17 09:15:00 도착예정

드라이버 driver  
0회 운행 | 평균별점 0.0 (0회 참여)

★★★★★

드라이버 메시지  
▶ 조용히 가실 분

차량정보 미니쿠퍼 123가1234  
 동승료 500 포인트  
 예정운행시간 2023-02-17 08:50:00 출발 > 2023-02-17 09:15:00 도착  
 실제운행시간 2023-03-03 17:14:10 출발 > 2023-03-03 17:19:21 도착

### 3 동승자 목록

태워주세요 도착확인완료

★★★★★

의정부통근러 도착확인완료

어제먼허맘 도착확인대기

★★★★★

별점 등록  
star.do

동승자일 경우

- 드라이버 및 다른 동승자에 별점 평가

# 카풀 관리 - 화면 구현 [카풀 등록]

출발, 도착지를 지도에 마커로 표시  
드래그로 옮길 수 있음  
▶ 이후 운행정보의 지도에 표시됨

차가와

이용중인 카풀    공지사항    카풀    이용후기    메시지 0    마이페이지    로그아웃

## 카풀 등록하기

출발

경기 의정부시    경민대학교

도착

경기 의정부시    의정부백병원

출발예정시각

2023-03-03 오후 05:40

도착예정시각

2023-03-03 오후 06:00

동승료

1200

최대동승인원

2

드라이버 메시지

드라이버 메시지를 입력해 주세요(예시: 음악 들으면서 같이 가실 분~)


이전으로

카풀 등록하기

write.do

# 카풀 관리 - 화면 구현 [카풀 운영-기타]

이용중인 카풀이 없을 경우


 차가와

이용중인 카풀   공지사항   카풀   이용후기   메시지 0   마이페이지   로그아웃

## 현재 이용 중인 카풀이 없어요

카풀 리스트 보러 가기

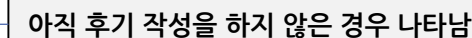
동승 확정 후 탑승하지 않은  
상태로 출발한 경우

 차가와

이용중인 카풀   공지사항   카풀   이용후기   메시지 0   마이페이지   로그아웃

## 차가 가 버렸어요

동승확정 후 출발하기 전까지 탑승하지 않아 미탑승처리 되었어요.  
포인트는 차감되지 않아요.  
해당 차량 운행 종료 후 새로운 카풀을 이용할 수 있어요.



### 후기작성

# 카풀 관리 - 핵심 코드

## ViewService에 사용된 쿼리

```
public class CarpoolViewServiceImpl implements ServiceInterface {

    private CarpoolDAO dao;

    @Override
    public void setDao(Object obj) {
        this.dao = (CarpoolDAO) obj;
    }

    @Override
    public Object service(Object obj) throws Exception {
        CarpoolVO vo = dao.view((Long) obj);
        if (vo != null) {
            vo.setPsgList(dao.psgList(vo.getNo()));
        }
        return vo;
    }
}
```

```
@Override
public CarpoolVO view(Long no) throws Exception {
    System.out.println("CarpoolDAOImpl.view()");
    CarpoolVO vo = null;
    try {
        con = getConnection();
        String sql = " SELECT c.status, c.no, c.driverId, m.nickname, m.profileImage, c.startRegion, c.startAddress, "
            + " c.arriveRegion, c.arriveAddress, c.startLat, c.starting, c.arriveLat, c.arriveIng, "
            + " c.startTimePre, c.arriveTimePre, c.startTime, c.arriveTime, c.price, c.seats, d.carModel, d.carNo, c.memo, c.star, "
            + " (SELECT count(*) FROM carpool WHERE driverId = 'driver' AND status='종료') driveCount, " // 운행횟수
            + " (SELECT trunc(avg(star),2) FROM starpoint WHERE receiver = m.id) dstarAvg, " // 드라이버 평균별점
            + " (SELECT count(*) FROM starpoint WHERE receiver = m.id) dstarCount, " // 드라이버 별점횟수
            + " (SELECT count(*) FROM passenger WHERE fixed = 0 AND cpNo = c.no) papplCount, " // 현재 동승신청자 수
            + " (SELECT count(*) FROM passenger WHERE fixed = 1 AND cpNo = c.no) pfixCount, " // 현재 동승확정자 수
            + " (SELECT count(*) FROM passenger WHERE gotin = 1 AND cpNo = c.no) pgotCount, " // 현재 탑승확정자 수
            + " (SELECT count(*) FROM passenger WHERE arrived = 1 AND cpNo = c.no) parrCount " // 현재 도착확정자 수
            + " FROM carpool c, member m, driver d WHERE c.no = ? AND c.driverId = m.id AND m.id = d.id ";
    }
}
```

- CarpoolVO 내에 동승자 명단 List<PassengerVO> 존재  
 ▶ 카풀 정보와 동승자 리스트를 각각 불러오는 쿼리

- 운행/동승/별점횟수, 평균별점, 동승상태별 동승자 수 등을  
 COUNT, AVG 등을 이용한 서브쿼리로 한 번에 가져옴

```
@Override
public List<PassengerVO> psList(Long cpNo) throws Exception {
    System.out.println("CarpoolDAOImpl.psgList()");
    List<PassengerVO> list = null;
    try {
        con = getConnection();
        String sql = " SELECT m.id, m.nickname, m.profileImage, p.msg, p.appTime, p.status, p.fixed, p.gotIn, p.arrived, p.star, "
            + " (SELECT count(*) FROM passenger WHERE psId = p.psgId AND status='종료') psgCount, " // 동승횟수
            + " (SELECT trunc(avg(star), 2) FROM starpoint WHERE receiver = p.psgId) pstarAvg, " // 평균별점
            + " (SELECT count(*) FROM starpoint WHERE receiver = p.psgId) pstarCount " // 별점횟수
            + " FROM passenger p, member m WHERE cpNo = ? AND p.psgId = m.id ";
    }
}
```



# 카풀 관리 - 핵심 코드

## 카카오 지도 API 활용 - write.jsp

### 카풀 등록하기



```

var container = document.getElementById('map'); // 지도를 표시할 div
var options = {
  center : new kakao.maps.LatLng(37.73931073401507,
    127.04486202736867), // 지도의 중심좌표
  level : 3
// 지도의 확대 레벨
};

// 지도를 표시할 div와 지도 옵션으로 지도를 생성합니다
var map = new kakao.maps.Map(container, options);

// 마커 이미지의 이미지 주소입니다
var imageSrc = "https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/markerStar.png";

// 마커 이미지의 이미지 크기 입니다
var imageSize = new kakao.maps.Size(24, 35);
// 마커 이미지를 생성합니다
var markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize);

// 마커를 생성합니다
var startMarker = new kakao.maps.Marker({
  map : map,
  position : new kakao.maps.LatLng(37.73931073401507,
    127.04486202736867),
  title : '출발',
  image : markerImage
});
startMarker.setDraggable(true);

// 마커를 생성합니다
var arriveMarker = new kakao.maps.Marker({
  map : map,
  position : new kakao.maps.LatLng(37.7393, 127.045),
  title : '도착',
  image : markerImage
});
arriveMarker.setDraggable(true);

$("#submit").click(function() {
  let startLatLng = startMarker.getPosition();
  let arriveLatLng = arriveMarker.getPosition();
  $("#startLat").val(startLatLng.getLat());
  $("#startLng").val(startLatLng.getLng());
  $("#arriveLat").val(arriveLatLng.getLat());
  $("#arriveLng").val(arriveLatLng.getLng());
  return true;
});

```

출발/도착 마커를 드래그 가능하도록 생성

폼 제출 시 마커의 위도/경도를 받아 DB에 저장

# 카풀 관리 - 핵심 코드

카카오 지도 API 활용 - view.jsp, now.jsp 등

이용 중인 카풀

1번차-운행정보

운행중

[경기 의정부시] 의정부시청 → [경기 의정부시] 의정부역  
2023-02-17 08:50:00 출발예정 - 2023-02-17 09:15:00 도착예정



```
var container = document.getElementById('map'); // 지도를 표시할 div

//시작, 도착지점의 좌표
var startLatLng = new kakao.maps.LatLng({vo.startLat}, ${vo.startLng});
var arriveLatLng = new kakao.maps.LatLng({vo.arriveLat}, ${vo.arriveLng});

var options = {
  center : startLatLng, // 지도의 중심좌표
  level : 3
};
// 지도의 확대 레벨

// 지도를 표시할 div와 지도 옵션으로 지도를 생성합니다
var map = new kakao.maps.Map(container, options);

// 마커 이미지의 이미지 주소입니다
var imageSrc = "https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/markerStar.png";

// 마커 이미지의 이미지 크기 입니다
var imageSize = new kakao.maps.Size(24, 35);
// 마커 이미지를 생성합니다
var markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize);

//마커 생성
var startMarker = new kakao.maps.Marker({
  map : map,
  position : startLatLng,
  title : '출발',
  image : markerImage
});
var arriveMarker = new kakao.maps.Marker({
  map : map,
  position : arriveLatLng,
  title : '도착',
  image : markerImage
});

// 지도를 재설정할 범위정보를 가지고 있을 LatLngBounds 객체를 생성합니다
var bounds = new kakao.maps.LatLngBounds();
bounds.extend(startLatLng);
bounds.extend(arriveLatLng);
map.setBounds(bounds);
```

DB에서 출발/도착 위도, 경도를 가져옴

출발/도착 마커를 위에서 가져온 위치에 생성

출발/도착 마커가 보이도록 지도 범위 설정



setDao 메서드를 잘못 작성

```
@Override  
public void setDao(Object obj) {  
    this.dao = (CarpoolDAO) dao;  
    System.out.println("listdao 세팅:"+dao);  
}
```

정보: 적어도 하나의 JAR가 TLD들을 찾기 위해 스캔되었  
listdao 세팅:null  
서버 가동: Init.init() 실행 완료  
2월 21, 2023 10:19:41 오전 org.apache.catali  
정보: 이름이 []인 컨텍스트를 다시 로드하는 것을 완료했

## Part 9, 카풀 관리 - 오류 수정

### 후기를 작성해도 '후기작성' 버튼이 나오는 오류

정보를 가져올 때 reviewed 컬럼을 안 가져와서 발생  
→ 쿼리에 추가

```
sql += " SELECT c.no, m.nickname, c.startRegion, c.startAddress, c.arriveRegion, c.arriveAddress, c.startTime, c.arriveTime, c."
      + " c.status, c.seats, d.carModel, d.carNo, c.star, "
      + " (SELECT count(*) FROM passenger WHERE fixed = 1 AND cpNo = c.no) pfixCount, " // 동승확정자 수;
      + " (SELECT count(*) FROM passenger WHERE gotin = 1 AND cpNo = c.no) pgotCount, " // 탑승자 수;
      + " (SELECT reviewed FROM passenger WHERE cpNo=c.no AND psId = ?) reviewed " // 내 리뷰여부
      + " FROM carpool c, member m, driver d " + condition
      + " AND m.id = d.id AND c.driverId = m.id AND (c.status = '종료' OR c.status = '도착' OR c.status = '취소') ORDER BY c.no
sql += ")); WHERE rnum BETWEEN ? AND ?";
```

3/2/4

27번 차 종료

[경기 의정부시] 이젠컴퓨터학원 앞 ▶ [경기 의정부시] 녹양역 2번출구

2023-02-28 12:58:32 출발 > 2023-02-28 14:35:45 도착  
드라이버 driver2 / 동승료 400 포인트 / 동승자 2명  
차량정보 123가1111(미니쿠팡)

passenger 종료  
passenger2 탑승  
driver3 종료

후기작성

2/1/4

26번 차 종료

[경기 의정부시] 의정부역 ▶ [경기 의정부시] 이젠컴퓨터학원

2023-02-24 16:44:40 출발 > 2023-02-24 16:44:45 도착  
드라이버 driver2 / 동승료 300 포인트 / 동승자 1명  
차량정보 123가1111(미니쿠팡)

passenger 종료  
passenger2 탑승

후기작성

1/1/4

23번 차 종료

[경기 의정부시] 집앞 ▶ [경기 의정부시] 회사앞

2023-02-24 14:07:02 출발 > 2023-02-24 14:07:05 도착  
드라이버 driver / 동승료 400 포인트 / 동승자 1명  
차량정보 123가1234(미니쿠팡)

passenger 종료

후기작성

1/1/3

21번 차 종료

[경기 의정부시] 집앞 ▶ [경기 의정부시] 회사앞

2023-02-24 12:48:21 출발 > 2023-02-24 12:48:25 도착  
드라이버 driver / 동승료 500 포인트 / 동승자 1명  
차량정보 123가1234(미니쿠팡)

passenger 종료

후기작성



3/2/4

27번 차 종료

[경기 의정부시] 이젠컴퓨터학원 앞 ▶ [경기 의정부시] 녹양역 2번출구

2023-02-28 12:58:32 출발 > 2023-02-28 14:35:45 도착  
드라이버 driver2 / 동승료 400 포인트 / 동승자 2명  
차량정보 123가1111(미니쿠팡)

passenger 종료  
passenger2 탑승  
driver3 종료

후기작성

2/1/4

26번 차 종료

[경기 의정부시] 의정부역 ▶ [경기 의정부시] 이젠컴퓨터학원

2023-02-24 16:44:40 출발 > 2023-02-24 16:44:45 도착  
드라이버 driver2 / 동승료 300 포인트 / 동승자 1명  
차량정보 123가1111(미니쿠팡)

passenger 종료  
passenger2 탑승

후기작성

1/1/4

23번 차 종료

[경기 의정부시] 집앞 ▶ [경기 의정부시] 회사앞

2023-02-24 14:07:02 출발 > 2023-02-24 14:07:05 도착  
드라이버 driver / 동승료 400 포인트 / 동승자 1명  
차량정보 123가1234(미니쿠팡)

passenger 종료

후기작성

## Part 9, 카풀 관리 - 오류 수정

포인트가 많은데 부족하다고 뜨는 오류


62

차가와

동승자를 모집하고 있

62번차-운행정보

[경기 의정부시] gg → [경기 의정부시] gg  
2023-03-02 10:36:00 출발 - 2023-03-02 10:38:00 도착

 드라이버 driver  
6회 운행 | 평균별점 5.0 (7회 참여)

드라이버 메시지  
▶ 123w

차량정보 미니쿠퍼 123가1234  
동승료 300 포인트

같이 타고 싶으신가요?

[보유: 19800 포인트]  
※ 확정 후 모집이 마감되면 취소할 수 없어요

같이 타요~ 동승신청

드라이버에게 문의하고 싶은 내용이 있나요? 메시지 보내기

localhost 내용:  
보유 포인트가 부족합니다. 마이페이지에서 포인트를 충전해 주세요.

확인

0 동승신청자

0/1 동승확정자

동승확정자 : 0 / 1

```
script type="text/javascript">
$(function() {
  if ("${param.exist}") { //동승신청 시 이용중인 카풀이 있으면 돌아옴
    alert("이미 이용 중인 카풀이 존재합니다.");
  }
  $("#applyBtn").click(function() {
    if ("${vo.price}" > "${point}") {
      alert("보유 포인트가 부족합니다. 마이페이지에서 포인트를 충전해 주세요.");
      return false;
    }
  });
});
script>
style type="text/css">
eftBox {
  width: 50%;
```

문자열로 가져왔기 때문에 "300" > "19800"로 계산됨  
→ parseInt()로 숫자로 변경하면 정상적으로 비교

```
$(function() {
  if ("${param.exist}") { //동승신청 시 이용중인 카풀이 있으면 돌아옴
    alert("이미 이용 중인 카풀이 존재합니다.");
  }
  $("#applyBtn").click(function() {
    if (parseInt("${vo.price}") > parseInt("${point}")) {
      alert("보유 포인트가 부족합니다. 마이페이지에서 포인트를 충전해 주세요.");
      return false;
    }
  });
});
script>
style type="text/css">
eftBox {
  width: 50%;
  height: 600px;
```

# Part 9.

## 카플 관리 - 시연

# 카풀 관리 - 개발 후기 및 차후 개발 내용

## < PL 정하영 >

### 개발 후기

콘솔이 아니라 웹으로 구현해 보니 더 보람차고 재미있었습니다. 넣고 싶은 기능이 많아지면서 어떻게 만들어낼지 고민하는 과정에서 많이 배울 수 있었고, 초기 설계의 중요성을 더욱 더 깨달았습니다. 지도 API를 가져오는 게 처음에는 막막하고 어려워서 못 할 줄 알았는데 한 번 성공하고 나니 다음에도 API문서를 보고 공부해서 잘 사용할 수 있을 것 같다는 자신감이 생겼습니다. 포인트, 회원, 후기, 메시지 다 조금씩 연결되어 있어서 이미 만들어 주신 기능을 가져다 쓰기도 하고 필요한 기능 개발을 요청하기도 했는데 다 들어 주시고 멋지게 완성해 주셔서 감사합니다.

### 차후 개발 내용

지도 마커를 옮긴 위치에 따라 출발지, 도착지 칸이 자동으로 입력되는 기능

# Part 10, 메시지 관리 - DB 구현

업무 영역		메시지 시스템		작성자			조용우	
테이블명		메시지		테이블 ID			Message	
테이블 설명		메시지 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
번호	no	Number			O			
내용	Content	Varchar2	2000	O				
발신자	Sender	Varchar2	20	O		O	Member(id)	
수신자	Acceptor	Varchar2	20	O		O	Member(id)	
발신 날짜	sendDate	Date		O			Sysdate	
받은 날짜	acceptDate	Date					Default null	
방 번호	Roomno	Number				O	Messageroom(roomno)	

# Part 10, 메시지 관리 - DB 구현

업무 영역		메시지 시스템		작성자			조용우	
테이블명		메시지 대화방		테이블 ID			Messageroom	
테이블 설명		메시지 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
방 번호	no	Number			O			
참여자1	Content	Varchar2	2000	O				
참여자2	Sender	Varchar2	20	O		O	Member(id)	
마지막 메시지	Acceptor	Varchar2	20	O		O	Member(id)	
방 생성 시간	sendDate	Date		O			Sysdate	
참여자1 방 존폐 여부	acceptDate	Date					Default null	
참여자2 방 존폐 여부	Roomno	Number				O	Messageroom(roomno)	
참여자 1 읽지 않은 메시지 개수	Unreadcount1	Number	10				Default 0	
참여자 2 읽지 않은 메시지 개수	Unreadcount2	Number	10				Default 0	

## Part 10, 메시지 관리 - DB 구현

CHAGAWA.MESSAGE		
P	NO	NUMBER
*	CONTENT	VARCHAR2 (2000 BYTE)
*	SENDER	VARCHAR2 (20 BYTE)
*	SENDDATE	DATE
*	ACCEPTER	VARCHAR2 (20 BYTE)
	ACCEPTDATE	DATE
F	ROOMNO	NUMBER
MESSAGE_PK (NO)		
SYS_C0010867 (ROOMNO)		



CHAGAWA.MESSAGEROOM		
P	ROOMNO	NUMBER
*	PAR1	VARCHAR2 (20 BYTE)
*	PAR2	VARCHAR2 (20 BYTE)
	LMCONTENT	VARCHAR2 (2000 BYTE)
	CDATE	TIMESTAMP (0)
	ROOMSTATUS1	VARCHAR2 (10 BYTE)
	ROOMSTATUS2	VARCHAR2 (10 BYTE)
	UNREADCOUNT1	NUMBER (10)
	UNREADCOUNT2	NUMBER (10)
MESSAGEROOM_PK (ROOMNO)		

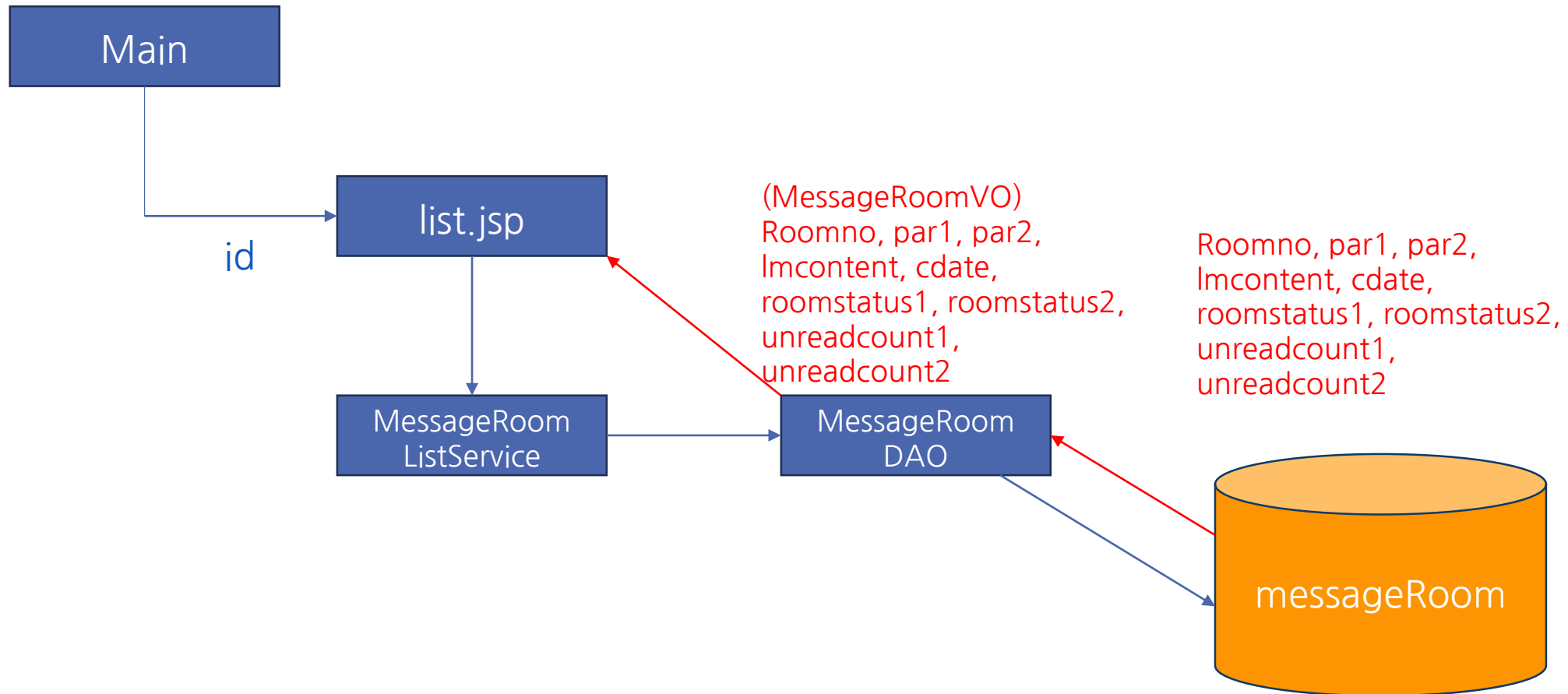
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	NO	NUMBER	No	(null)	1	(null)
2	CONTENT	VARCHAR2 (2000 BYTE)	No	(null)	2	(null)
3	SENDER	VARCHAR2 (20 BYTE)	No	(null)	3	(null)
4	SENDDATE	DATE	No	sysdate	4	(null)
5	ACCEPTER	VARCHAR2 (20 BYTE)	No	(null)	5	(null)
6	ACCEPTDATE	DATE	Yes	null	6	(null)
7	ROOMNO	NUMBER	Yes	(null)	7	(null)

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ROOMNO	NUMBER	No	(null)	1	(null)
2	PAR1	VARCHAR2 (20 BYTE)	No	(null)	2	(null)
3	PAR2	VARCHAR2 (20 BYTE)	No	(null)	3	(null)
4	LMCONTENT	VARCHAR2 (2000 BYTE)	Yes	(null)	4	(null)
5	CDATE	TIMESTAMP (0)	Yes	current_ti...	5	(null)
6	ROOMSTATUS1	VARCHAR2 (10 BYTE)	Yes	(null)	6	(null)
7	ROOMSTATUS2	VARCHAR2 (10 BYTE)	Yes	(null)	7	(null)
8	UNREADCOUNT1	NUMBER (10, 0)	Yes	0	8	(null)
9	UNREADCOUNT2	NUMBER (10, 0)	Yes	0	9	(null)

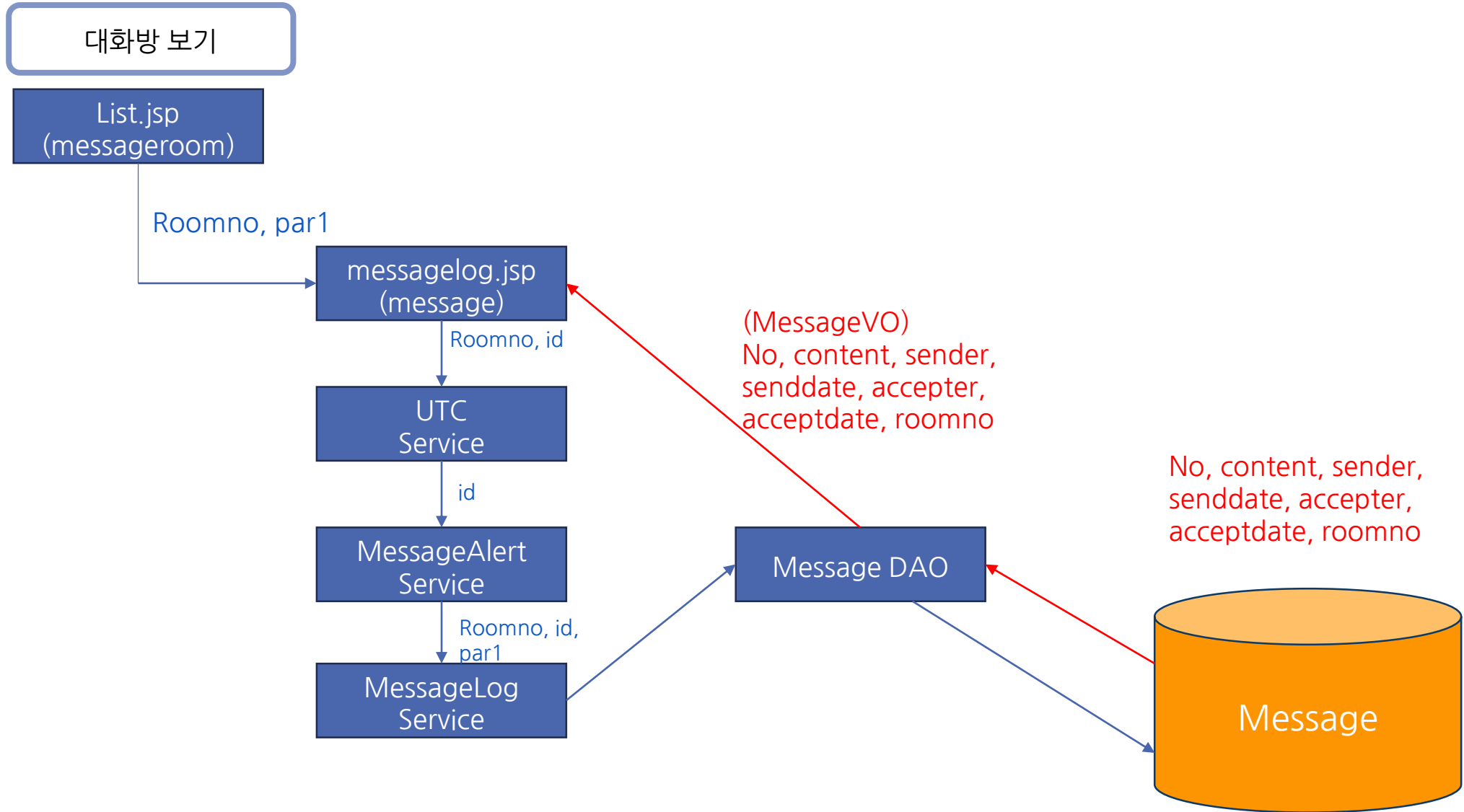


## Part 10, 메시지 관리 - DFD

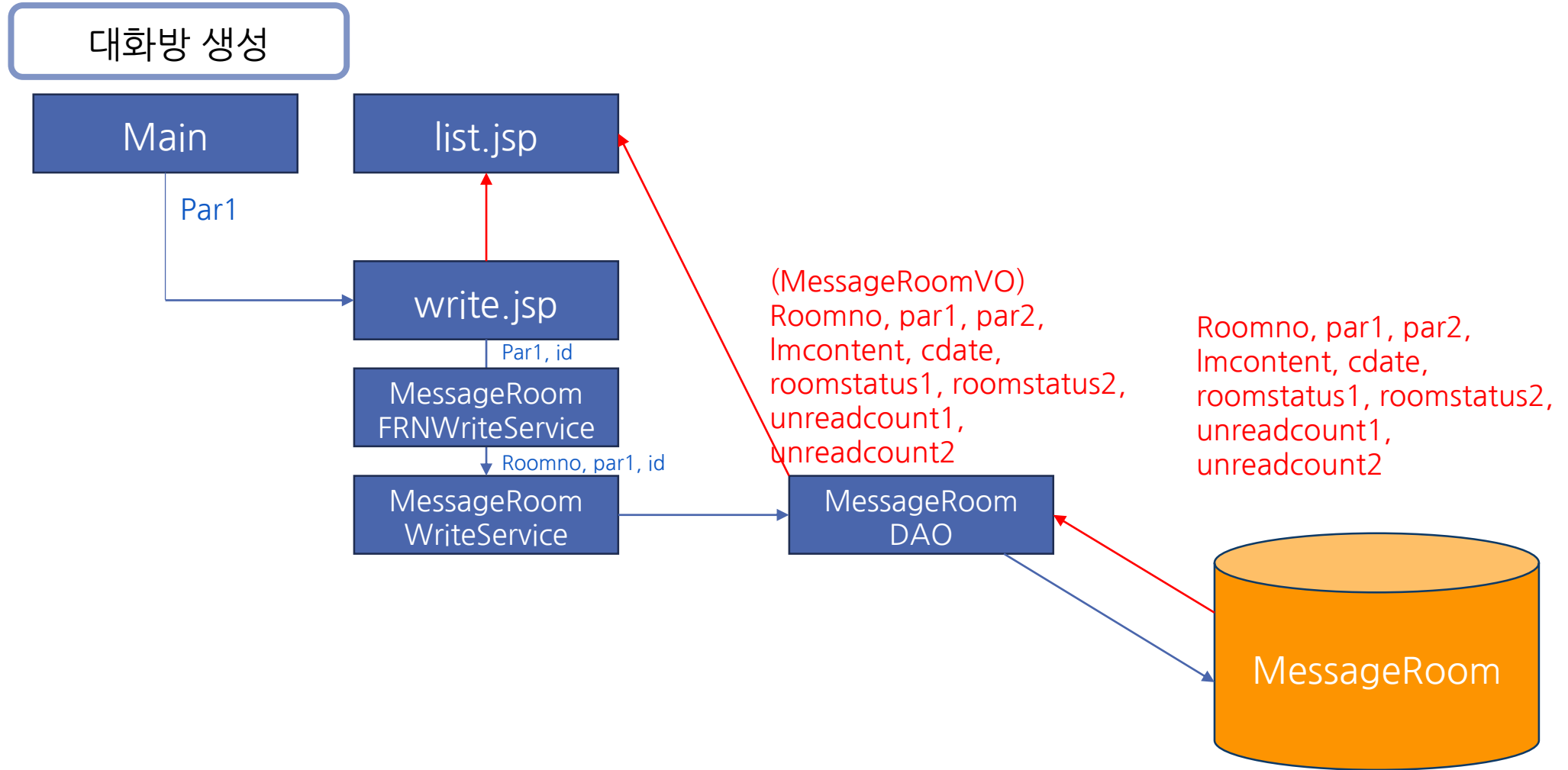
대화방 리스트



## Part 10, 메시지 관리 - DFD

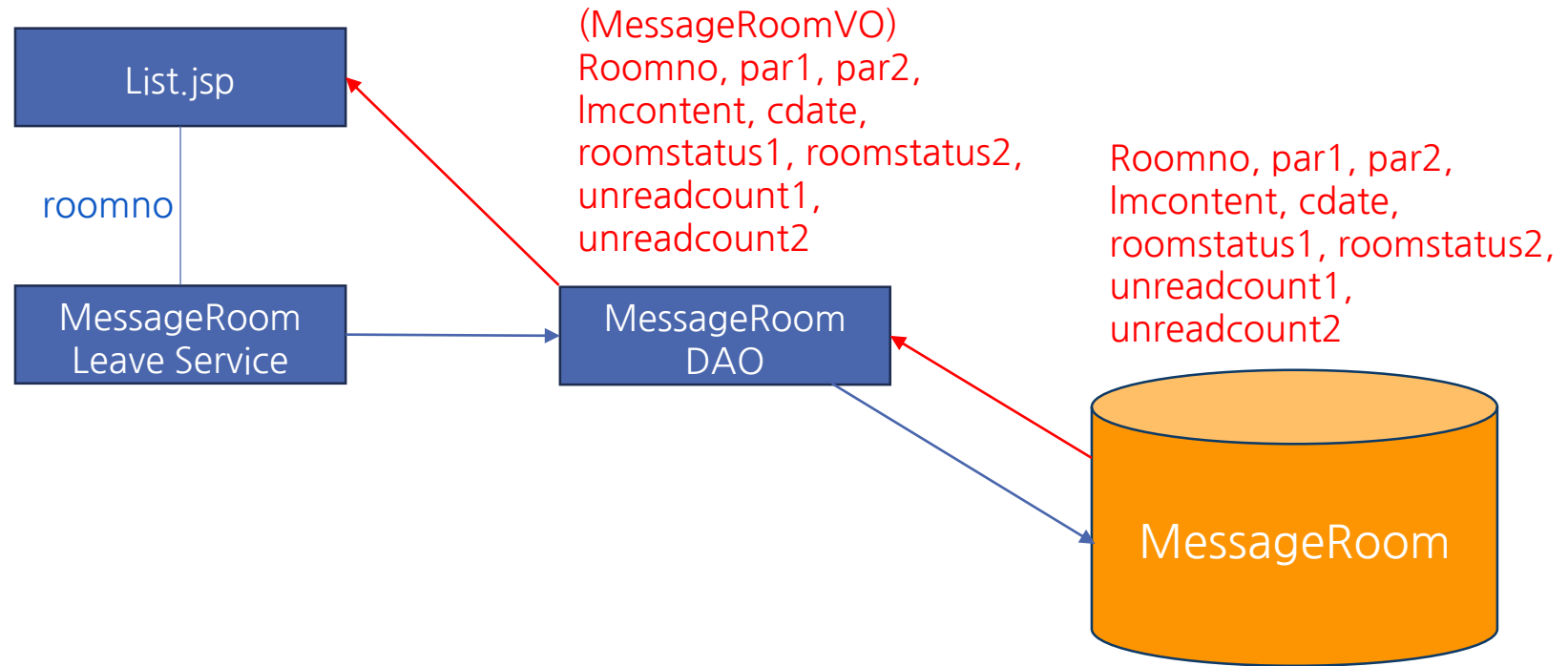


## Part 10, 메시지 관리 - DFD



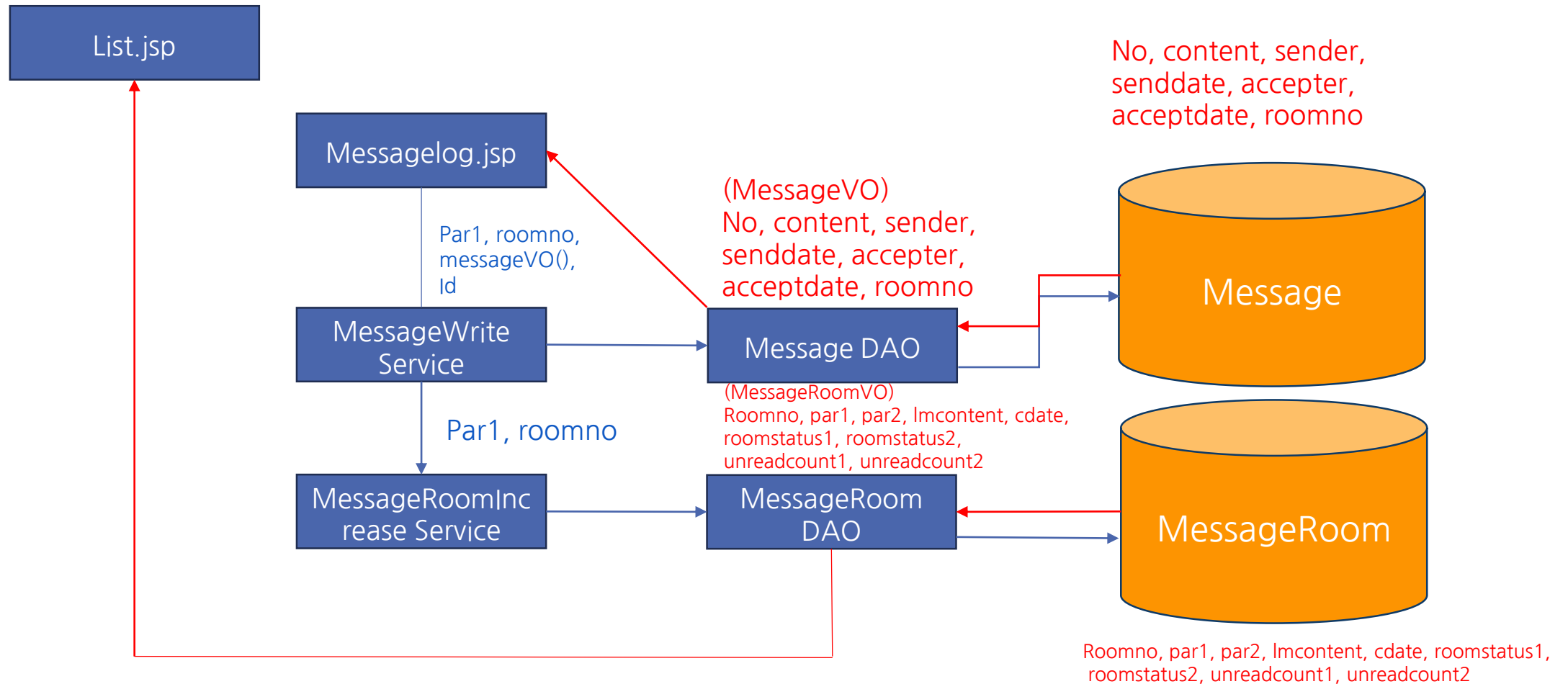
## Part 10, 메시지 관리 - DFD

### 대화방 나가기



## Part 10, 메시지 관리 - DFD

### 메시지 보내기



## Part 10, 메시지 관리 - 화면 구현



### 대화방

프로필	아이디	최근 메시지	방 생성날짜	
	passenger	asd	2023-03-03 15:04:23	나가기
	driver	aaa	2023-03-03 11:49:35	나가기



### 대화방

- 안녕하세요 2023-03-03 15:04:26
- adsfadsfasdf 2023-03-03 16:26:02
- asdf 2023-03-03 16:26:49

asdf

2023-03-03 16:40:04

보내기

## Part 10, 메시지 관리 - 핵심 코드

```
case "/messageroom/write.do":
    String par1 = request.getParameter("par1");
    long roomno = (long) Execute.run(messageRoomFRNService, new Object[] { par1, loginVO.getId() });
    System.out.println(roomno);
    if (roomno == 0) {
        System.out.println(par1);
        System.out.println(loginVO.getId());
        Execute.run(messageRoomWriteService, new Object[] { par1, loginVO.getId() });

        roomno = (long) Execute.run(messageRoomFRNService, new Object[] { par1, loginVO.getId() });
        jsp = "redirect:/message/messagelog.do?roomNo=" + roomno + "&par1=" + par1;
    } else {
        jsp = "redirect:/message/messagelog.do?roomNo=" + roomno + "&par1=" + par1;
    }
    break;
```

```
@Override
public long findroomno(String par, String id) throws Exception{

    long result = 0;
    try {
        con = DB.getConnection();
        String sql = "select roomno from messageroom where (par1 = ? and par2 = ?) or (par1 = ? and par2 = ?)";
        pstmt = con.prepareStatement(sql);
        pstmt.setString(1, par);
        pstmt.setString(2, id);
        pstmt.setString(3, id);
        pstmt.setString(4, par);
        rs = pstmt.executeQuery();

        if(rs != null && rs.next()) {
            result = rs.getLong(1);
        }
    } catch (Exception e) {
        e.printStackTrace(); // 개발자를 위한 코드
        throw new Exception("계시판 글삭제 DB 처리 중 오류 발생");
    } finally {
        DB.close(con, pstmt, rs);
    }

    return result;
}
```

< 대화방이 이미 존재할 경우 FRNService를 실행해 기존 대화방의 roomno를 찾고, roomno가 있으면 기존 대화방으로 redirect 하는 코드 >

## Part 10, 메시지 관리 - 핵심 코드

```
@Override
public int leave(long roomno, String id, int x) throws Exception {
    int result = 0;

    try {
        con = DB.getConnection();

        String sql = "update MessageRoom set roomstatus" + x + " = '없다' "
            + " where par"+ x + " = ? and roomno = ? ";

        pstmt = con.prepareStatement(sql);
        pstmt.setString(1, id);
        pstmt.setLong(2, roomno);
        result = pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace(); // 개발자를 위한 코드
        throw new Exception("게시판 글삭제 DB 처리 중 오류 발생");
    } finally {
        DB.close(con, pstmt);
    }

    return result;
}
```

```
@Override
public int findpar(long roomno, String id) throws Exception{

    try {
        con = DB.getConnection();
        String sql = "select roomno from messageroom where (roomno = ? and par2 = ?)";
        pstmt = con.prepareStatement(sql);
        pstmt.setLong(1, roomno);
        pstmt.setString(2, id);
        rs = pstmt.executeQuery();

        if(rs != null && rs.next()) {
            return 2;
        }

        sql = "select roomno from messageroom where (roomno = ? and par1 = ?)";
        pstmt = con.prepareStatement(sql);
        pstmt.setLong(1, roomno);
        pstmt.setString(2, id);
        rs = pstmt.executeQuery();
        if(rs != null && rs.next()) {
            return 1;
        }
    } catch (Exception e) {
        e.printStackTrace(); // 개발자를 위한 코드
        throw new Exception("게시판 글삭제 DB 처리 중 오류 발생");
    } finally {
        DB.close(con, pstmt, rs);
    }

    return 0;
}
```

```
public class MessageRoomLeaveServiceImpl implements ServiceInterface{
    private MessageRoomDAO dao;

    @Override
    public void setDao(Object obj) {
        this.dao = (MessageRoomDAO)obj;
    }

    public Object service(Object obj) throws Exception {

        Object[] objs = (Object[]) obj;
        long roomno = (long) objs[0];
        System.out.println(objs[0]);
        String id = (String) objs[1];
        System.out.println(objs[1]);
        int x = (int) dao.findpar(roomno, id);
        return dao.leave(roomno, id, x);
    }
}
```

< 사용자가 나가고자 하는 방의 roomno를 찾아주는 메서드를 추가해  
자신이 속한 방의 roomstatus를 '없다'로 update >



## Part 10, 메시지 관리 - 핵심 코드

```
public void doLog() {  
    case "/message/messageLog.do":  
  
        String noStr = request.getParameter("noStr");  
        String par1 = request.getParameter("par1");  
        long no = Long.parseLong(noStr);  
        System.out.println(no);  
  
        Execute.run(UTService, new Object[] { no, loginVO.getId() });  
        Execute.run(messageAlertService, loginVO.getId());  
        request.setAttribute("list", Execute.run(messageLogService, new Object[] { loginVO.getId(), par1, no }));  
        jsp = "message/messageLog";  
        break;  
}
```

```
@Override  
public Object service(Object obj) throws Exception {  
    // TODO Auto-generated method stub  
    Object[] objs = (Object[]) obj;  
    long no = (long) objs[0];  
    String id = (String) objs[1];  
    int x = dao.findpar(no, id);  
    System.out.println("x = " + x);  
    System.out.println("no = " + no);  
    System.out.println("id = " + id);  
    return dao.utc(no, id, x);  
}
```

```
@Override  
public Object alert(String id) throws Exception {  
    // TODO Auto-generated method stub  
    int result = 0;  
    try {  
        // 1. 확인 2. 연결  
        con = DB.getConnection();  
        // 3. sql  
        String sql = "update message set acceptdate = sysdate "  
            + " where accepter = ? and acceptdate is null ";  
        System.out.println("MessageDAOImpl.write().sql : " + sql);  
        // 4. 실행객체 & 데이터 생성  
        pstmt = con.prepareStatement(sql);  
        pstmt.setString(1, id);  
        // 5. 실행 -> sql  
        result = pstmt.executeUpdate();  
        // 6. 데이터 저장 - list  
        // sql이 null 이 아니면 처리하자  
        System.out.println("MessageDAOImpl.write() - 등록 성공");  
    } catch (Exception e) {  
        // TODO: handle exception  
        e.printStackTrace(); // 개발자를 위한 코드  
        throw new Exception("공지사항 등록 DB 처리 중 오류 발생");  
    } finally {  
        // 7. 닫기  
        DB.close(con, pstmt);  
    } // end of try ~ catch ~ finally  
  
    return result;  
}
```

```
@Override  
public Object utc(long no, String id, int x) throws Exception {  
    // TODO Auto-generated method stub  
    int result = 0;  
    try {  
        // 1. 확인 2. 연결  
        con = DB.getConnection();  
        // 3. sql  
        String sql = "update message set unreadcount = x + " + id + " "  
            + " where no = ? and par = ? ";  
        System.out.println("MessageDAOImpl.write().sql : " + sql);  
        System.out.println("no = " + no + " id = " + id);  
        // 4. 실행객체 & 데이터 생성  
        pstmt = con.prepareStatement(sql);  
        pstmt.setLong(1, no);  
        pstmt.setString(2, id);  
        // 5. 실행 -> sql  
        result = pstmt.executeUpdate();  
        // 6. 데이터 저장 - list  
        // sql이 null 이 아니면 처리하자  
        System.out.println("MessageDAOImpl.update() - 등록 성공");  
    } catch (Exception e) {  
        // TODO: handle exception  
        e.printStackTrace(); // 개발자를 위한 코드  
        throw new Exception("공지사항 등록 DB 처리 중 오류 발생");  
    } finally {  
        // 7. 닫기  
        DB.close(con, pstmt);  
    } // end of try ~ catch ~ finally  
  
    return result;  
}
```

< 상대방과의 대화방에 입장하면  
자신의 acceptDate를 null 값에서  
sysdate로 update하는 쿼리와  
읽지 않은 message의 개수를  
0으로 update하는 쿼리를  
실행시키는 메서드 >

## Part 10, 메시지 관리 - 오류 수정

차가와 이용중인 카를 공저사할 카를 이용후기 메시지 0 마이페이지 로그아웃

대화방

안녕하세요

보내기

```
java.lang.NullPointerException
    at com.chagawa.messageRoom.service.MessageRoomIncreaseServiceImpl.service(MessageRoomIncreaseServiceImpl.java:23)
    at com.chagawa.main.Execute.run(Execute.java:8)
    at com.chagawa.message.controller.MessageController.execute(MessageController.java:67)
    at com.chagawa.main.DispatcherServlet.service(DispatcherServlet.java:65)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:779)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:227)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:162)
    at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:189)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:162)
    at com.opensymphony.sitemesh.webapp.SiteMeshFilter.obtainContent(SiteMeshFilter.java:129)
    at com.opensymphony.sitemesh.webapp.SiteMeshFilter.doFilter(SiteMeshFilter.java:77)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:189)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:162)
    at com.chagawa.common.filter.AuthorityFilter.doFilter(AuthorityFilter.java:104)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:189)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:162)
    at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:177)
    at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:97)
    at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:541)
    at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:135)
    at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:92)
    at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:687)
    at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:78)
    at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:368)
```

< init 파일에서 dao를 연결시키지 않아 메시지 전송 시 오류 발생 >

# Part 10.

메시지 관리 - 시연

# 메시지 관리 - 개발 후기 및 차후 개발 내용

## < PM 조용우 >

### 개발 후기

처음 DB설계에서 너무 많이 시간을 뺏겨서 더 많은 것을 구현하지 못했다. 그 점이 제일 아쉬웠고 또, 프로젝트를 진행함에 있어서 최대한 혼자 할 수 있는 건 혼자 하고 싶었지만 역시나 어려움이 많았고 팀원에게 의지 할 수 밖에 없는 상황이 많아서 그런 내 자신이 싫었다. 팀원 분께 너무 감사하기도 하고 죄송스럽기도 했지만 나도 문제에 직면했을 때 스스로 해결하는 역량을 갖춰야 겠다 생각했다. 또한 아직 기본기가 너무 부족해서 막히는 부분이 많았다. 공부를 한참 해야겠다 싶었다

### 차후 개발 내용

- 웹소켓이나 ajax를 사용하여 실시간 채팅 프로그램을 만들어 보고 싶다.

## Part 11, 이용 후기 관리 - DB 구현

업무 영역		이용 후기 관리 시스템		작성자			김태웅	
테이블명		이용 후기		테이블 ID			epilogue	
테이블 설명		이용 후기 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
글 번호	no	number			O		epilogue_seq	
제목	title	varchar2	300	O				
내용	content	varchar2	2000	O				
작성자 ID	writer	varchar2	20	O		O	m.id	
작성일	writeDate	date					sysdate	
카풀 번호	cpNo	number		O		O	carpool.no	
조회수	hit	number					0	

## Part 11, 이용 후기 관리 - DB 구현

업무 영역		별점 관리 시스템		작성자			김태웅	
테이블명		별점		테이블 ID			starPoint	
테이블 설명		별점 관리에 필요한 데이터						
컬럼명	컬럼 ID	타입	길이	NN	PK	FK	비고	
번호	no	number			O		starPoint_seq	
카풀 번호	cpNo	number		O		O	carpool.no	
작성자 ID	starter	varchar2	20	O		O	m.id	
대상자 ID	receiver	varchar2	20	O		O	m.id	
별점	start	number		O				

# Part 11, 이용 후기 관리 - DB 구현

CHAGAWA.EPILOGUE		
P	* NO	NUMBER
	* TITLE	VARCHAR2 (300 BYTE)
	* CONTENT	VARCHAR2 (2000 BYTE)
	* WRITER	VARCHAR2 (20 BYTE)
	WRITEDATE	DATE
F	* CPNO	NUMBER
	HIT	NUMBER
🔑 EPILOGUE_NO_PK (NO)		
🔑 SYS_C0010874 (CPNO)		
🔑 EPILOGUE_NO_PK (NO)		

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	NO	NUMBER	No	(null)	1	(null)
2	TITLE	VARCHAR2 (300 BYTE)	No	(null)	2	(null)
3	CONTENT	VARCHAR2 (2000 BYTE)	No	(null)	3	(null)
4	WRITER	VARCHAR2 (20 BYTE)	No	(null)	4	(null)
5	WRITEDATE	DATE	Yes	sysdate	5	(null)
6	CPNO	NUMBER	No	(null)	6	(null)
7	HIT	NUMBER	Yes	0	7	(null)

## Part 11, 이용 후기 관리 - 화면 구현 [리스트]



차가와

공지사항

카풀

이용후기

회원가입

로그인

### 이용 후기

제목



Search



후기글 개수

10



번호	제목	작성자	작성일	조회수
----	----	-----	-----	-----



후기작성/별점등록





## Part 11, 이용 후기 관리 - 화면 구현 [보기]

[차가와](#)[이용증인 카풀](#)[공지사항](#)[카풀](#)[이용후기](#)[메시지](#) 0[마이페이지](#)[로그아웃](#)

### 후기 글 보기

글번호	21
카풀번호	2
제목	처음 이용해봤습니다.
작성자	passenger
작성일	2023-03-06
내용	부드러운 운전 감사합니다. 앞으로도 가끔 이용 할 거 같네요 ^^
조회수	1

[수정](#)[삭제](#)[글목록](#)

## Part 11, 이용 후기 관리 - 화면 구현 [쓰기]



차가와

이용중인 카풀

공지사항

카풀

이용후기

메시지 0

마이페이지

로그아웃

### 후기 작성

카풀번호

2

2

작성자

passenger

제목

내용

등록

새로입력

취소

## Part 11, 이용 후기 관리 - 화면 구현 [수정]



차가와

이용증인 카풀

공지사항

카풀

이용후기

메시지 0

마이페이지

로그아웃

### 후기 글 수정

번호

21

카풀번호

2

작성자

passenger

제목

처음 이용해봤습니다.(?)

내용

무느러운 운전 감사합니다.  
앞으로도 가끔 이용 할 거 같네요 ^^  
는 구라고 적당히 달리세요 사고나는줄 알았네..

수정

새로입력

취소

## Part 11, 이용 후기 관리 - 화면 구현 [삭제]

 차가와

이용중인 카풀    공지

localhost 내용:

정말 삭제하시겠습니까?

확인

취소

### 후기 글 보기

글번호	21
카풀번호	2
제목	처음 이용해봤습니다.(?)
작성자	passenger
작성일	2023-03-06
내용	부드러운 운전 감사합니다. 앞으로도 가끔 이용 할 거 같네요 ^^ 는 구라고 적당히 달리세요 사고나는줄 알았네..
조회수	1

수정

삭제

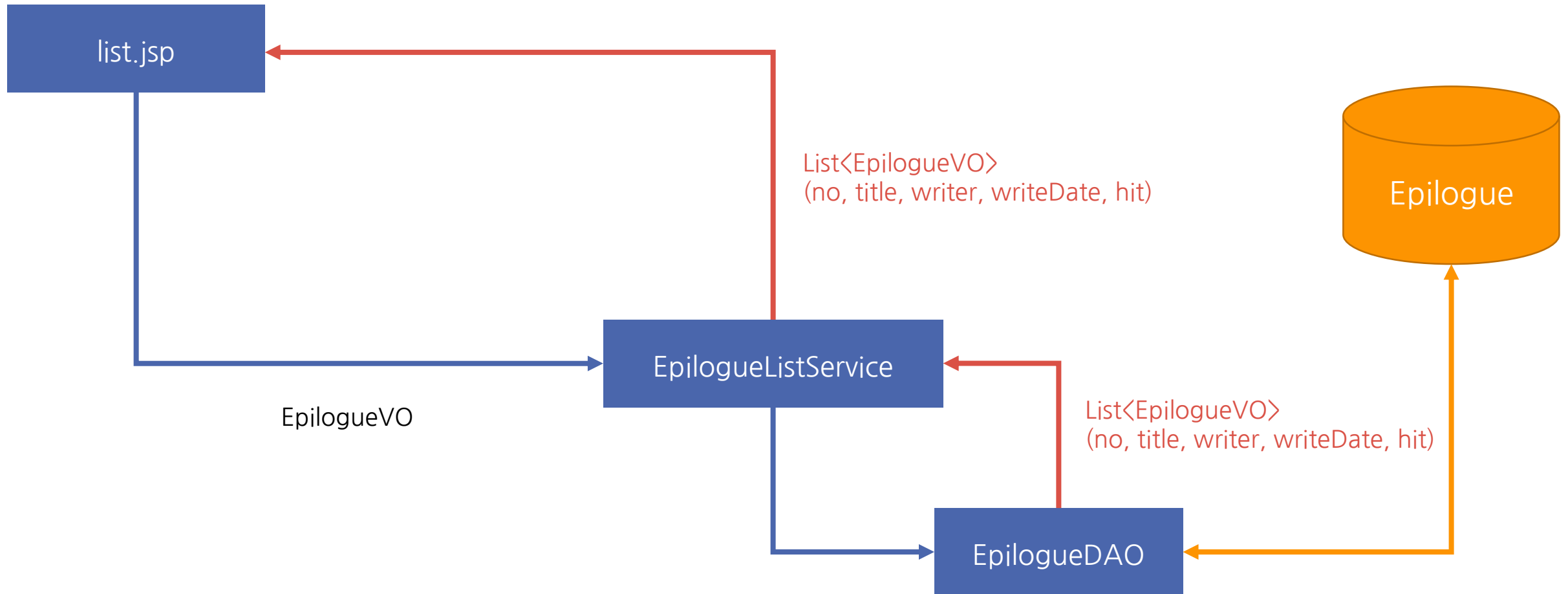
글목록

삭제

취소

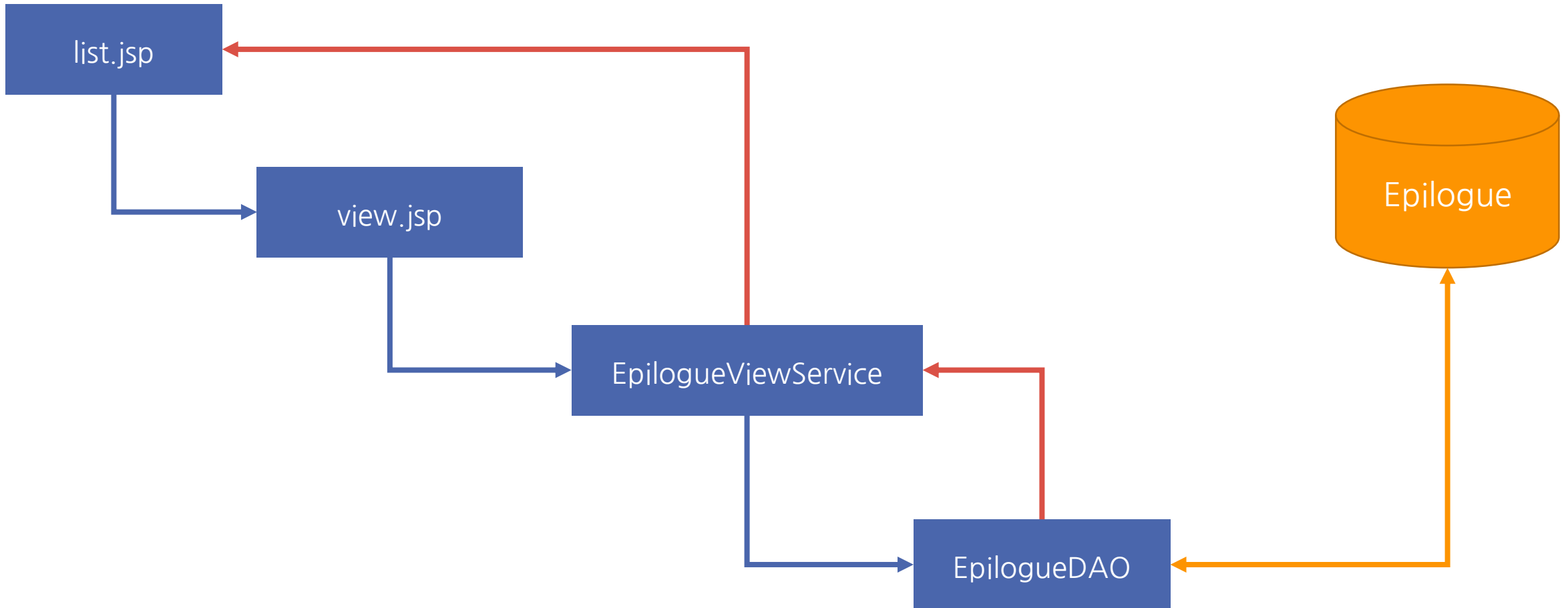
## Part 11, 이용 후기 관리 - DFD

이용 후기 리스트



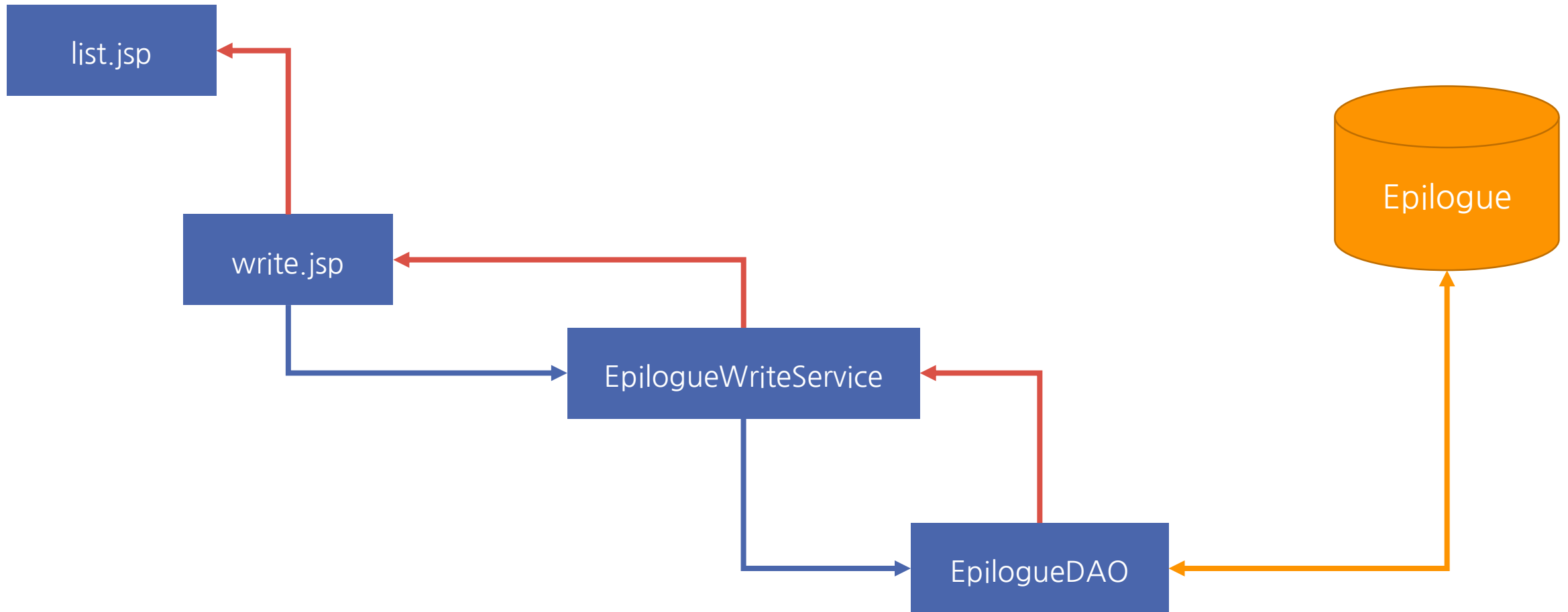
## Part 11, 이용 후기 관리 - DFD

이용 후기 보기



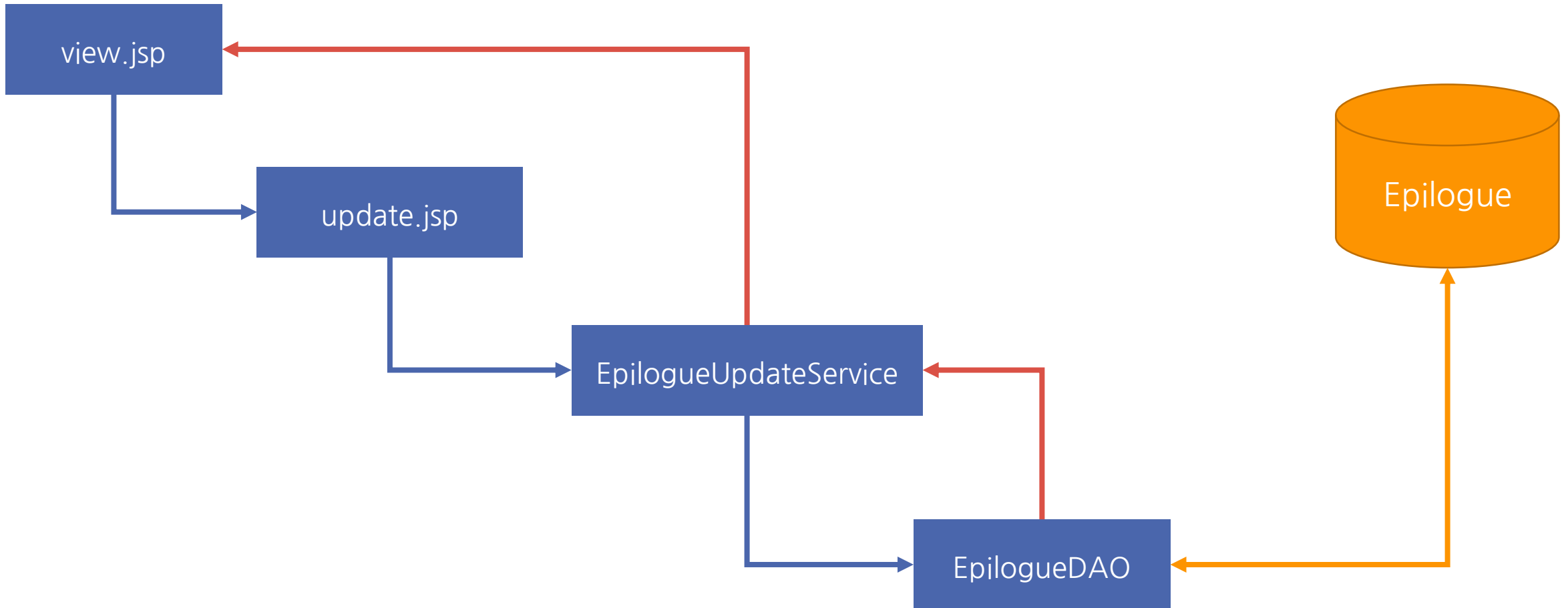
## Part 11, 이용 후기 관리 - DFD

이용 후기 등록



## Part 11, 이용 후기 관리 - DFD

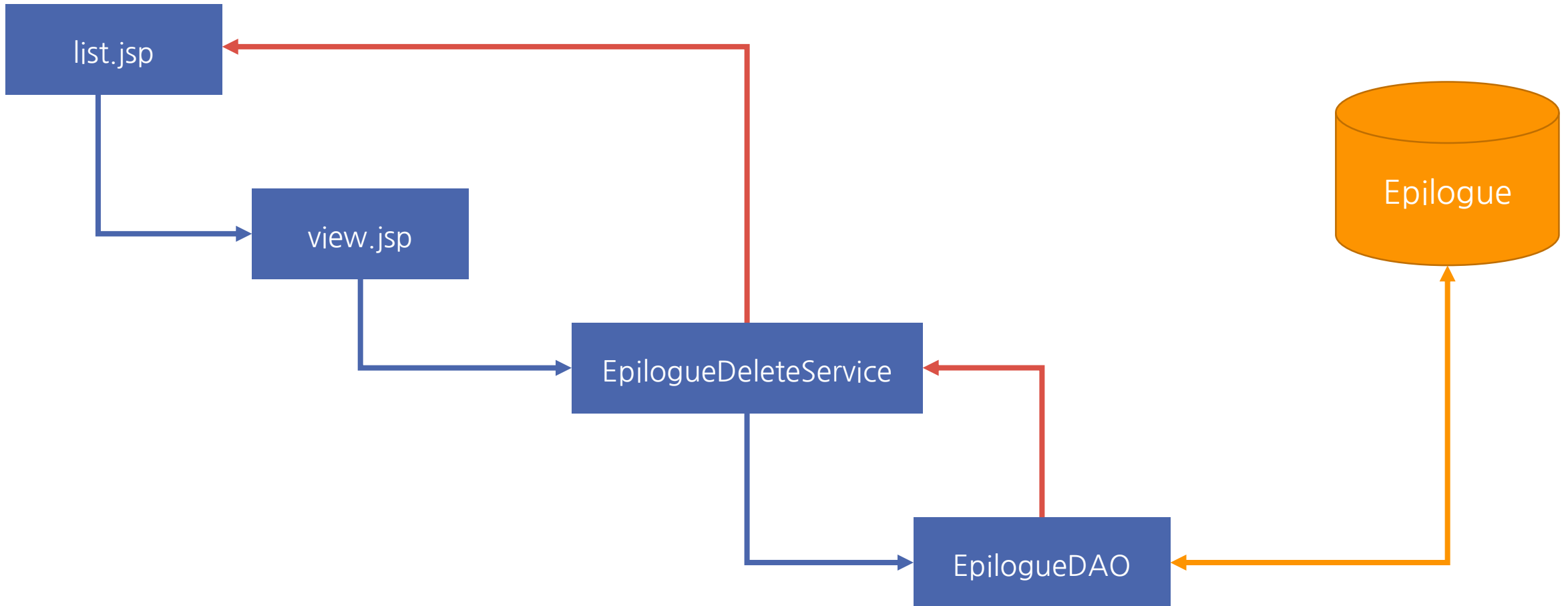
이용 후기 수정





## Part 11, 이용 후기 관리 - DFD

이용 후기 삭제



## Part 11, 이용 후기 관리 - 핵심 코드

```
// Write
case "/epilogue/write.do":

    if(request.getMethod().equals("GET")) {
        // 로그인정보
        LoginVO login = (LoginVO) request.getSession().getAttribute("login");
        // 카풀번호 리스트 CarpoolDAOImpl -> unreviewedCpNo
        List<Long> cplist = (List<Long>) Execute.run(carpoolUnreviewedService, login.getId());
        request.setAttribute("list", cplist);
        jsp = "epilogue/write";
    } else {

        // 페이지 데이터 수집 -> perPageNumStr을 URL 뒤에 붙여서 사용.
        // 데이터 수집
        EpilogueVO writeVO = new EpilogueVO();
        writeVO.setTitle(request.getParameter("title"));
        writeVO.setContent(request.getParameter("content"));
        writeVO.setWriter(request.getParameter("writer"));
        long cpNo = Long.parseLong(request.getParameter("cpNo"));
        writeVO.setReviewed(1);
        writeVO.setCpNo(cpNo);

        // 글등록 처리
        Execute.run(epilogueWriteService, writeVO);
        Execute.run(carpoolReviewedService, writeVO);
        jsp = "redirect:list.do?perPageNum="
            + request.getParameter("perPageNum");
    } // end of if

break;
```

```
<label for="cpNo">카풀번호</label>
<select name="cpNo">
<c:if test="${param.cpNo != null }">

    <option>${param.cpNo }</option>
</c:if>
```



해당 카풀번호에 대한 후기작성 선택 시  
cpNo 자동 세팅



로그인 정보, 카풀번호 가져오기  
해당 카풀번호에 대한 리뷰 1로 세팅  
→ 리뷰작성 완료.

## Part 11, 이용 후기 관리 - 핵심 코드

```
// Delete
case "/epilogue/delete.do":
    // 데이터 수집 - 글번호
    // 로그인정보
    LoginVO login = (LoginVO) request.getSession().getAttribute("login");

    long deleteNo = Long.parseLong(request.getParameter("no"));
    EpilogueVO deleteVO = new EpilogueVO();
    deleteVO.setNo(deleteNo);
    long deleteCpNo = Long.parseLong(request.getParameter("cpNo"));
    deleteVO.setCpNo(deleteCpNo);
    deleteVO.setWriter(login.getId());
    deleteVO.setReviewed(0);
    // 글삭제 처리
    Execute.run(epilogueDeleteService, deleteVO);
    Execute.run(carpoolReviewedService, deleteVO);
    jsp = "redirect:list.do?perPageNum="
        + request.getParameter("perPageNum");
    break;

default:
```

```
<c:if test="${login != null && vo.writer == login.id }">
<a href="update.do?no=${vo.no }&page=${param.page }&perPageNum=${p.
class="btn btn-default">수정</a>
<a href="delete.do" class="btn btn-default" id="deleteBtn">삭제</a>
</c:if>
```



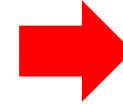
로그인 정보가 작성자와 같으면  
수정, 삭제



로그인 정보, 글 번호, 카풀번호 가져오기  
삭제 시 해당 카풀번호에 대한 리뷰 0으로 세팅  
→ 리뷰 작성 안함  
→ 다시 작성 가능

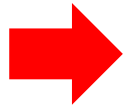
## Part 11, 이용 후기 관리 - 오류 수정

`java.lang.ClassCastException:`  
`com.chagawa.epilogue.dao.EpilogueDAOImpl cannot`  
`be cast to com.chagawa.epilogue.dao.EpilogueDAO`



DAO캐스팅

`invalid identifier`



오타 수정

`ORA-00942: table or view does not exist`



테이블 명 오타 수정

Part 11.

이용 후기 관리 - 시연

## Part 11, 이용 후기 관리 - 개발 후기 및 차후 개발 내용

< 팀장 김태웅 >

### 개발 후기

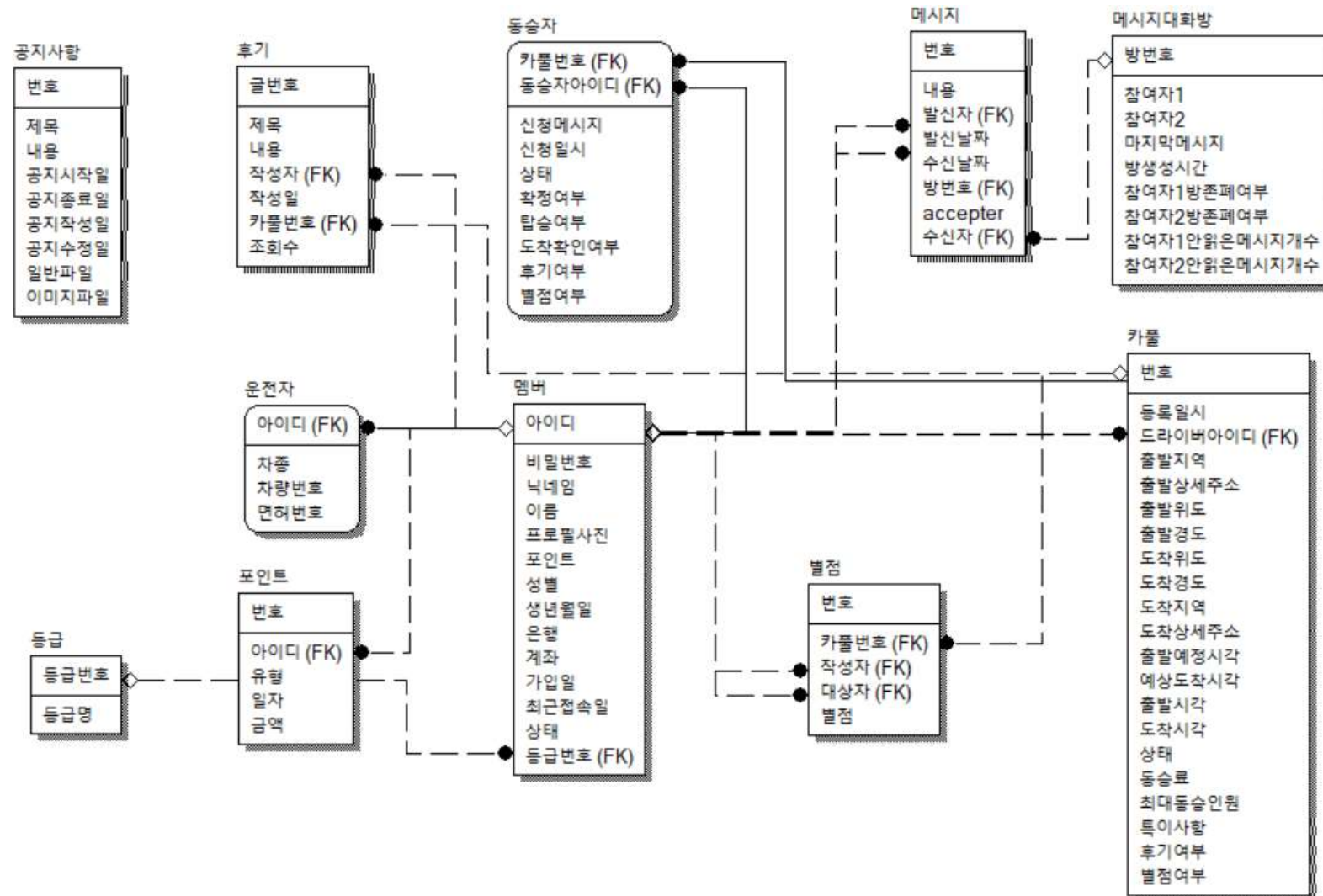
완전 무지의 상태에서 시작해서 뭐부터 시작해야 될지 한숨만 나왔지만, 도움도 받고 수업 자료도 보면서 어찌저찌 마무리는 된 거 같다. 화면이 구현이 됨에 따라 흥미가 조금씩 생기면서 하고싶은 게 생기기 시작 했는데, 아는게 없어서 공부의 필요성을 느끼고는 있는데.. 네 그렇습니다.

다들 고생하셨습니다.

### 차후 개발 내용

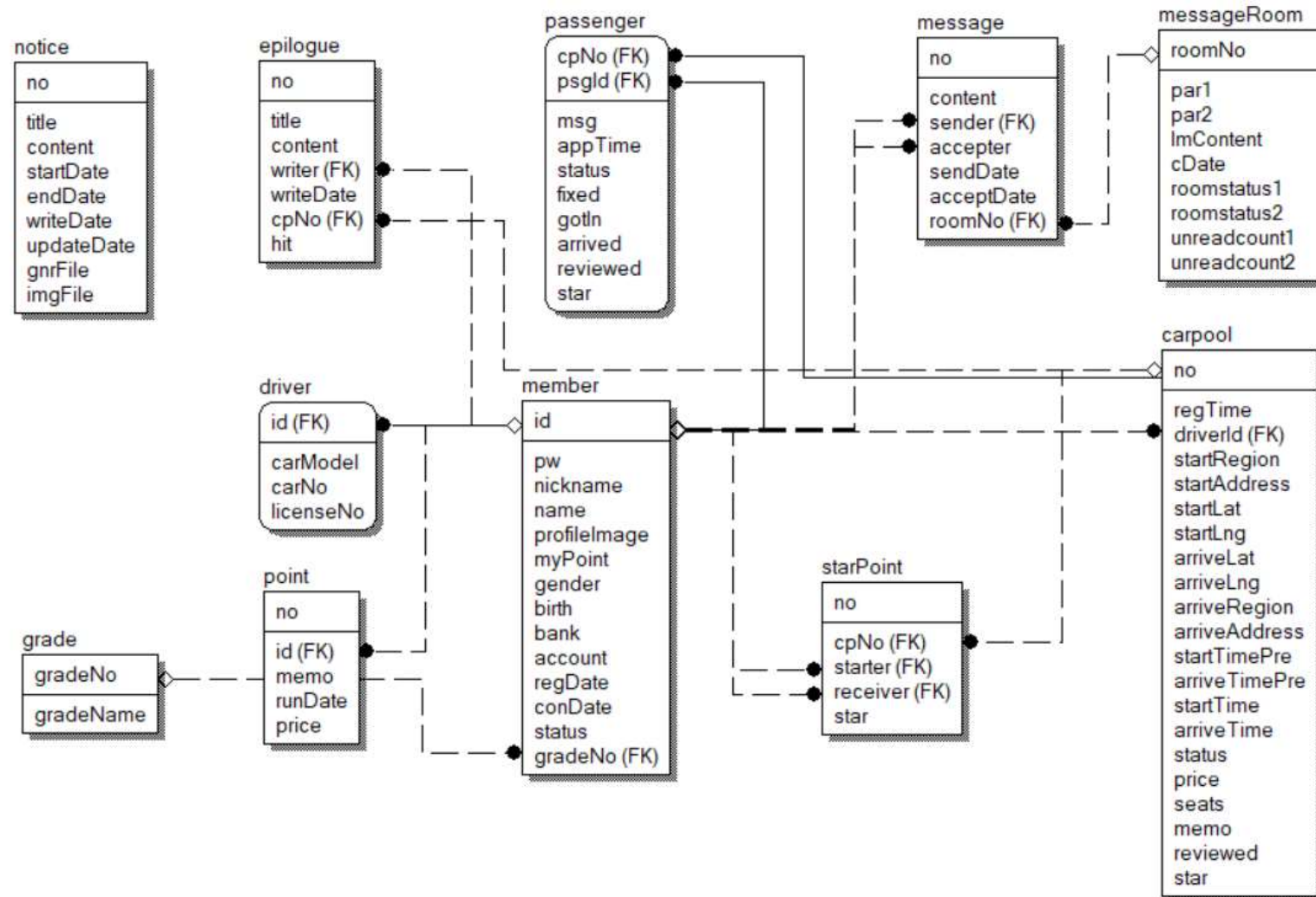
카풀을 이용한 정보가 있어야만 글 등록을 할 수 있어야 할까..? 라는 의문에 대한, 사이트 이용 시 회원이라면 카풀 이용 내역이 없더라도 후기 작성이 가능하게.. (ex) 취소나 기타 등등..

# ERwin - Logical





# ERwin - Physical





감사합니다