

Project

• • •
Web_POS

이름 : 이상록

목차

1. 개발환경
2. 요구사항 정의서
3. UI설계
4. DATABASE
5. 주요기술상세
6. 개발자 테스트

소프트웨어

JAVA 1.8, Apache Tomcat v8.5, MySQL(Workbench 8.0),
GIT, Spring Framework, Ecilpse

하드웨어

Server Computer(LP 완료, 한 대)

Windows 버전

Windows 10 Enterprise LTSC
Copyright © 2018 Microsoft
Corporation. All rights reserved.

시스템

프로세서 : Intel(R) Core(TM) i3-
6100 CPU @ 3.70GHz 3.70GHz
설치된 메모리(RAM) : 8.00GB
시스템 종류 : 64비트 운영 체제

Server Computer(LP 완료, 한 대)

Windows 버전

Windows 10 Pro
Copyright © 2019 Microsoft
Corporation. All rights reserved.

시스템

프로세서 : Intel(R) Core(TM) i7-
3517U CPU @ 1.90GHz 2.40GHz
설치된 메모리(RAM) : 4.00GB
시스템 종류 : 64비트 운영 체제

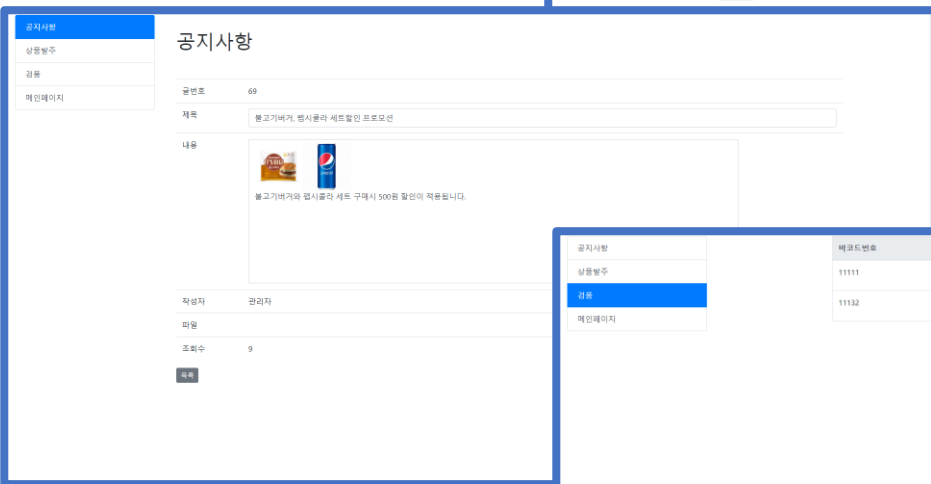
ID	이름	내용	세부항목	중요도	요구사항
A1	메인	로그인	로그인	상	로그인 페이지
			회원가입	상	회원가입 리스트 -이름(필수) -아이디(필수) -비밀번호(필수) -이메일(필수) -성별(필수) -생년월일(필수)
상품판매		상품등록	상	바코드번호를 입력하여 원하는 상품을 등록	
		평선키	하	원하는 상품을 평선키에 등록하여 버튼 등록만으로 상품 등록	
		계산기	중	리스트에 등록된 상품들의 합계 등을 출력	
		가상패드	상	바코드번호를 입력하거나 원하는 기능선택	
		판매내역	상	거래내역을 날짜별로 조회하여 환불	
A3		상품발주	공지사항	중	관리자가 등록한 공지사항 확인
			상품발주	상	원하는 상품에 해당하는 발주 수량을 입력하여 저장 후 관리자가 승인하여 발주
B1	관리자 페이지	공지사항	공지사항	중	관자가 원하는 공지사항을 등록, 수정, 삭제
상품등록		상품등록	상	관리자가 새로운 상품을 등록 - 상품번호 - 바코드번호 - 타입 - 세부타입 - 제조사 - 상품명 - 유통기한 - 사진	
					발주승인
개별발주 승인		상	사용자가 발주 신청한 매장번호를 선택하여 세부 상품들을 승인		



1. 로그인 – WEB_POS를 사용하기 위한 필수적인 기능이며
Interceptor를 사용하여 로그인을 하지 않고 상품판매, 상품발주 버튼을 클릭하면 자동으로 로그인 페이지로 이동, 추가적으로 admin 계정으로 로그인을 하면 관리자 페이지로 이동
2. 회원가입 – LoginController를 이용하여 member_info 테이블에 member_no, id, pw, email, gender, age에 대한 정보를 Post방식으로 저장

1. 상품등록 - Ajax(getJSON)를 이용해 상품등록, 상품등록 시 promotion테이블에 행사정보를 조회해 자동으로 할인적용
2. 결제 - 등록된 상품들의 member_no, barcode_no, md_ea, sale_price, type(현금, 카드), list_no에 대한 정보를 Post방식으로 DB(pos_board)에 저장
3. 판매내역 - model.addAttribute를 이용해 posPaymentList.jsp에 오늘 날짜에 해당하는 판매내역 정보를 출력, 날짜를 선택하면 Ajax(getJSON)을 이용해 과거 거래내역 출력가능
4. 판매내역 상세보기 - pos_board 테이블에 저장된 정보 중 사용자의 member_no, list_no, sale_date와 일치하는 정보를 출력





배코드번호	상품명	원가	정가	입고수량	검통완료
11111	고형만두	1500	4000	2	승인
11132	서울우유1L	400	2200	4	승인

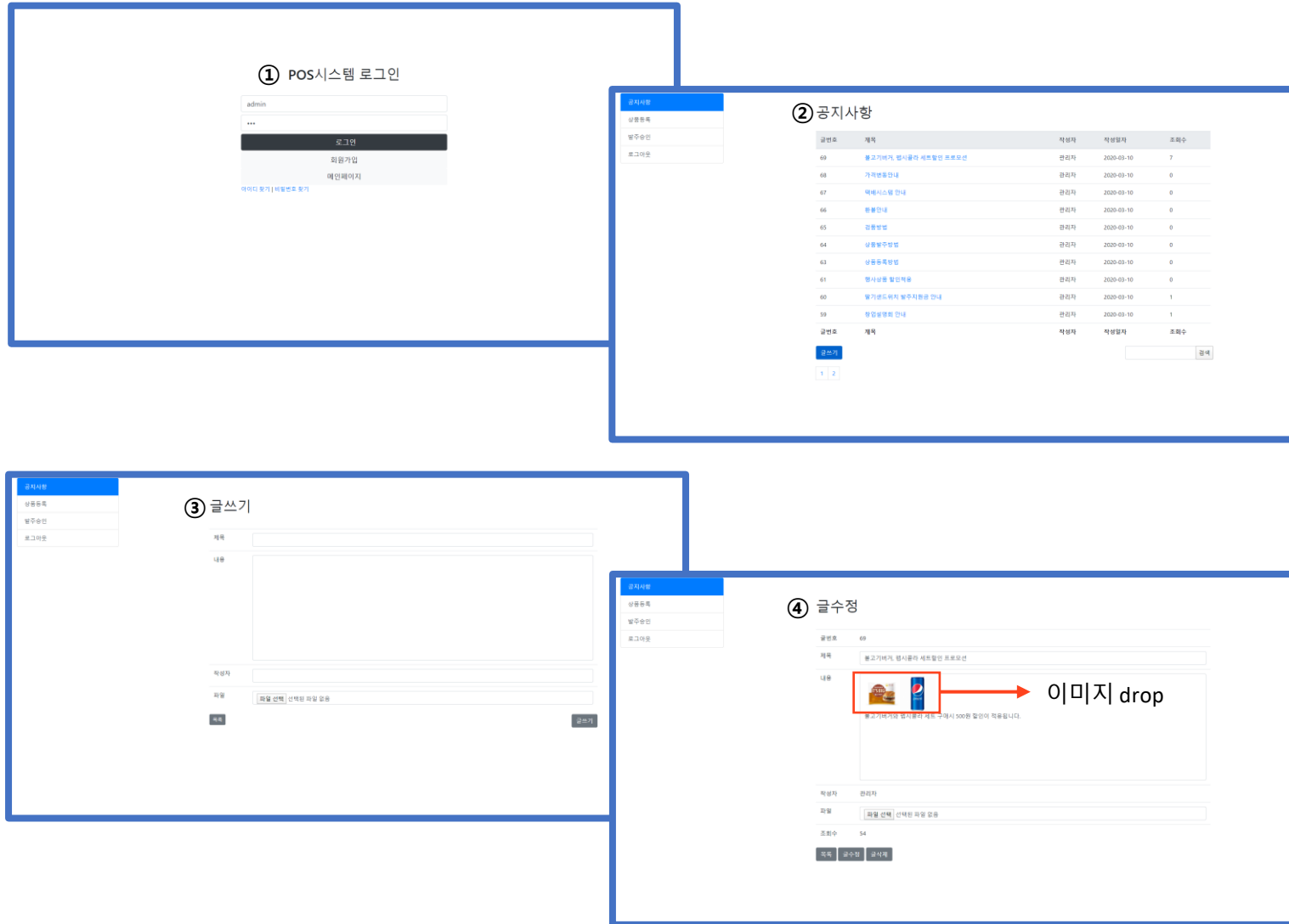
배코드번호	사진	상품명	제조사	상품명	원가	정가	유통기한	재고 수량	발주 수량
11125		김치찌개	삼양	분할용김치	300	1400	180	0	2
11144		김치찌개	밀도	용두명	300	1100	90	0	0
11146		김치찌개	농심	육개장사발면	200	850	90	0	0

```

$.getJSON("orderList?member_no="+member_no+"&type="+type+"&detail_type="+detail_type+"&md_name="+md_name, function (data) {
    console.log(data);
    str += "<table class='table table-bordered text-center' id='detailtable' style='width: 100%;>";
    str += "<thead class='thead-light'>";
    str += "<tr>";
    str += "<th width=10%>배코드번호</th>";
    str += "<th width=10%>사진</th>";
    str += "<th width=10%>상품명</th>";
    str += "<th width=10%>제조사</th>";
    str += "<th width=10%>상품명</th>";
    str += "<th width=10%>원가</th>";
    str += "<th width=10%>정가</th>";
    str += "<th width=10%>유통기한</th>";
    str += "<th width=10%>재고 수량</th>";
    str += "<th width=10%>발주 수량</th>";
    str += "</tr>";
    cnt = 0;
    $.each(data, function() {
        console.log("order = "+this.order_ea);
        str += "<tr id='detail'+cnt+'>";
        str += "<td id='barcode_no'+cnt+'>"+this.barcode_no + "</td>";
        str += "<td id='img'+cnt+'>";
        str += "<img src='"+this.md_img + "'/;></td>";
        str += "<td>"+this.detail_type + "</td>";
        str += "<td>"+this.company + "</td>";
        str += "<td>"+this.md_name + "</td>";
        str += "<td>"+this.cost + "</td>";
        str += "<td>"+this.price + "</td>";
        str += "<td>"+this.shelf_life + "</td>";
        str += "<td id='approval_ea'+cnt+'>"+this.approval_ea+"</td>";
        str += "<td id='input_type'+cnt+'>";
        str += "<input type='text' value='"+this.order_ea+"' id='order_ea'+cnt+' size='7px'></td>";
        str += "<input id='hidden_ea'+cnt+' type='hidden' value='"+this.order_ea+"'>";
        str += "</td>";
        cnt++;
    });
    str += "</thead>";
    str += "</table>";
    $("detailtable").html(str);
    str = "";
});

```

1. 공지사항 – 상품발주 버튼을 클릭하면 OrderMain.jsp로 이동하여 공지사항을 기본값으로 지정, 일반 사용자는 게시글 확인, 검색, 페이지 이동 권한만 부여됨
2. 게시글확인 – 등록된 공지사항 확인 시 조회수가 자동으로 증가하고 업로드한 파일을 다운가능
3. 상품발주 – 상품발주 탭 클릭 후 Ajax를 이용하여 화면전환, 이후 모든 화면전환을 예시와 같이 Ajax를 이용해 화면전환
4. 검통 – 상품발주 적용 후 관리자가 요청을 승인하면 order_board 테이블의 approval_ea 컬럼을 SELECT하여 출력, “승인”버튼 클릭 시 해당 tr삭제



1. 로그인 – Interceptor를 이용해 관리자 계정으로 로그인 시 관리자 페이지로 이동
2. 공지사항 – 사용자가 조회 가능한 게시물을 등록, 수정, 삭제 할 수 있으며, 페이지마다 10개의 게시물이 등록되면 페이징 처리되어 페이지가 1개씩 증가 10개 이상의 페이지가 등록되면 “<, >”버튼이 생성되어 10페이지 단위로 이동가능
3. 글쓰기 – 게시글 등록 시 <div>에 contentEditable 기능을 사용하여 게시글 내용에 text를 입력하거나 이미지 파일을 drop하면 자동으로 이미지가 업로드
4. 글수정 – 기존에 등록된 게시글의 제목, 내용, 파일을 수정할 수 있으며, 글수정, 글삭제가 가능

공지사항

상품등록

발주승인

로그아웃

① 상품등록

바코드번호: 11111

타입: 냉동식품

세부타입: 만두

제조사: 해태

상품명: 고향만두

원가: 1500

정가: 4000

유통기한: 365

사진: 파일 선택 고향만두.jpg →

상품등록



1. 상품등록 - 사용자가 판매, 발주가 가능하도록 상품의 정보를 입력 후 등록하면 md_info, md_price 테이블에 저장, 사진업로드 시 우측<div>에 이미지 업로드

공지사항

②

상품등록

발주승인

로그아웃

매장번호	수량	원가 합계	정가 합계	전체 승인
1	5	1500	6000	승인
2	9	3400	11500	승인

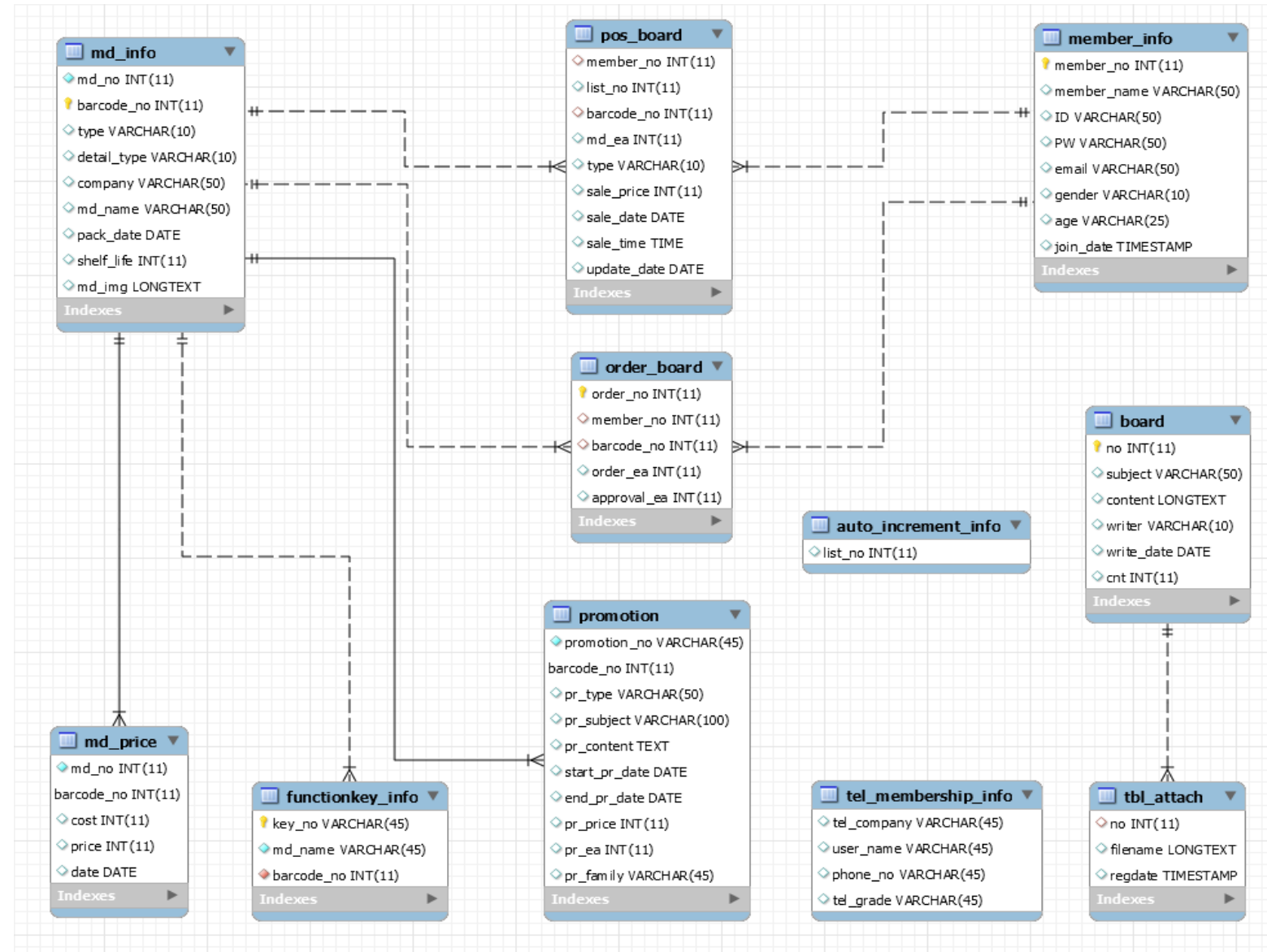
바코드번호

바코드번호	상품명	원가	정가	수량	발주승인
11125	불닭볶음면	300	1400	2	승인
11139	신라면	400	1300	3	승인
11149	진라면매운맛	400	1200	4	승인

2. 발주승인 - 사용자가 원하는 상품을 발주하면 사용자의 회원번호에 해당하는 발주리스트가 출력되고 원하는 사용자의 리스트를 클릭하면 상세 발주내역이 출력, 상세 테이블에서 개별 승인이나 전체 승인으로 발주승인이 가능

CVS_DATABASE

md_info, member_info 테이블을
메인으로 로그인, 발주, 상품판매
등의 기능을 구현



member_info
member_no INT(11)
member_name VARCHAR(50)
ID VARCHAR(50)
PW VARCHAR(50)
email VARCHAR(50)
gender VARCHAR(10)
age VARCHAR(25)
join_date TIMESTAMP
Indexes

데이터베이스명	CVS					
테이블명	member_info					
테이블 설명	회원가입한 회원들에 대한 정보를 저장하는 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
회원번호	member_no	int(11)	NO	PRI		auto_increment
이름	member_name	varchar(50)	YES			
아이디	ID	varchar(50)	YES	UNI		
비밀번호	PW	varchar(50)	YES			
전자메일	email	varchar(50)	YES	UNI		
성별	gender	varchar(10)	YES			
나이	age	varchar(25)	YES			
가입일	join_date	timestamp	YES			CURRENT_TIMESTAMP/ DEFAULT_GENERATED

member_info

- 회원가입 시 member_info 테이블에 정보를 저장하고 결제,발주 시 member_info를 외래키로 테이블들을 조인하여 정보를 SELECT

auto_increment_info
list_no INT(11)

데이터베이스명	CVS					
테이블명	auto_increment_info					
테이블 설명	데이터 저장 후 거래번호 자동증가를 위한 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
거래번호	list_no	int(11)	NO			

auto_increment_info

- 상품구매 시 list_no에 사용자의 member_no를 저장하여 count(list_no)로 거래번호를 증가

pos_board
member_no INT(11)
list_no INT(11)
barcode_no INT(11)
md_ea INT(11)
type VARCHAR(10)
sale_price INT(11)
sale_date DATE
sale_time TIME
update_date DATE
Indexes

데이터베이스명	CVS					
테이블명	pos_board					
테이블 설명	판매내역에 관한 정보를 저장하는 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
회원번호	member_no	int(11)	YES	member_no_fkpb (member_info)		
거래번호	list_no	int(11)	YES			
바코드번호	barcode_no	int(11)	YES	fk_pb(md_info)		
상품수량	md_ea	int(11)	YES			
거래 타입	type	varchar(10)	YES			
할인가격	sale_price	int(11)	YES			
판매일	sale_date	date	YES			
판매시간	sale_time	time	YES			

pos_board

- 사용자가 상품 판매 후 사용자의 정보, 판매한 상품에 관한 정보 등을 저장하여 추후 판매내역 조회, 환불 등의 기능을 제공

order_board
order_no INT(11)
member_no INT(11)
barcode_no INT(11)
order_ea INT(11)
approval_ea INT(11)
Indexes

데이터베이스명	CVS					
테이블명	order_board					
테이블 설명	발주한 상품에 관한 정보를 저장하는 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
주문번호	order_no	int(11)	NO	PRI		auto_increment
회원번호	member_no	int(11)	YES	member_no_fk (member_info)		
바코드번호	barcode_no	int(11)	YES	fk_ob(md_info)		
주문수량	order_ea	int(11)	YES			0
입고예정수량	approval_ea	int(11)	YES			

order_board

- 사용자가 발주한 상품에 대한 정보를 저장한 테이블

md_info
md_no INT(11)
barcode_no INT(11)
type VARCHAR(10)
detail_type VARCHAR(10)
company VARCHAR(50)
md_name VARCHAR(50)
pack_date DATE
shelf_life INT(11)
md_img LONGTEXT
Indexes

데이터베이스명	cvs					
테이블명	md_info					
테이블 설명	상품에 대한 정보를 저장한 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
상품번호	md_no	int(11)	NO	UNI		auto_increment
바코드번호	barcode_no	int(11)	NO	PRI		
상품타입	type	varchar(10)	YES			
상품상세타입	detail_type	varchar(10)	YES			
제조사	company	varchar(50)	YES			

md_info

- 사용자가 판매, 발주 가능한 상품들을 저장하는 테이블
상품을 등록 할때마다 각각의 md_no를 제공하고, barcode_no를 외래키로 다른 테이블과 연결

md_price
md_no INT(11)
barcode_no INT(11)
cost INT(11)
price INT(11)
date DATE
Indexes

데이터베이스명	cvs					
테이블명	md_price					
테이블 설명	상품의 가격에 대한 정보를 저장한 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
상품번호	md_no	int(11)	NO	UNI		auto_increment
바코드번호	barcode_no	int(11)	NO	PRI	addprice(md_info)	
원가	cost	int(11)	YES			
정가	price	int(11)	YES			
가격등록일	date	date	YES			

md_price

- 매년 변동되는 물가를 고려하여 상품 가격변경이 용이하도록 md_info테이블에 JOIN

promotion
promotion_no VARCHAR(45)
barcode_no INT(11)
pr_type VARCHAR(50)
pr_subject VARCHAR(100)
pr_content TEXT
start_pr_date DATE
end_pr_date DATE
pr_price INT(11)
pr_ea INT(11)
pr_family VARCHAR(45)
Indexes

데이터베이스명	CVS					
테이블명	promotion					
테이블 설명	상품 프로모션에 관한 정보를 저장하는 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
행사번호	promotion_no	varchar(45)	NO			
바코드번호	barcode_no	int(11)	NO	PRI	fk_pr(md_info)	
행스타입	pr_type	varchar(50)	YES			
행사제목	pr_subject	varchar(100)	YES			
행사내용	pr_content	text	YES			
행사시작일	start_pr_date	date	YES			
행사종료일	end_pr_date	date	YES			
할인가	pr_price	int(11)	YES			

promotion

- n+1, 묶음행사, 세트할인 등 상품 할인 프로모션과 관련된 정보를 저장

functionkey_info
key_no VARCHAR(45)
md_name VARCHAR(45)
barcode_no INT(11)
Indexes

데이터베이스명	CVS					
테이블명	functionkey_info					
테이블 설명	단축키 버튼에 관한 정보를 저장하는 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
키번호	key_no	varchar(45)	NO	PRI		
상품명	md_name	varchar(45)	NO			
바코드번호	barcode_no	int(11)	NO	fk_fi(md_info)		

functionkey_info

- 상품판매의 단축키에 입력한 상품정보를 저장하는 테이블

tel_membership_info ▼
tel_company VARCHAR(45)
user_name VARCHAR(45)
phone_no VARCHAR(45)
tel_grade VARCHAR(45)

데이터베이스명	CVS					
테이블명	tel_membership_info					
테이블 설명	통신사 할인적용을 위한 통신사 회원에 관한 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
통신사이름	tel_company	varchar(45)	YES			
회원이름	user_name	varchar(45)	YES			
핸드폰번호	phone_no	varchar(45)	YES			
통신사등급	tel_grade	varchar(45)	YES			

tel_membership_info

- 상품판매 페이지에서 회원이름, 핸드폰번호를 등록하여 통신사 등급에 따라 gold는 10%, silver는 5%할인 적용(행사상품 제외)

board
no INT(11)
subject VARCHAR(50)
content LONGTEXT
writer VARCHAR(10)
write_date DATE
cnt INT(11)
Indexes

데이터베이스명	CVS					
테이블명	board					
테이블 설명	공지사항 게시판 정보를 저장하는 테이블					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
게시물 번호	no	int(11)	NO	PRI		auto_increment
글제목	subject	varchar(50)	YES			
글내용	content	longtext	YES			
작성자	writer	varchar(10)	YES			
작성일	write_date	date	YES			
조회수	cnt	int(11)	YES			0

board

- 공지사항과 관련된 게시판 정보를 저장하는 테이블
사용자는 게시글 확인만 가능하고, 관리자만 글쓰기, 글수정, 삭제가 가능

tbl_attach
no INT(11)
filename LONGTEXT
regdate TIMESTAMP
Indexes

데이터베이스명	CVS					
테이블명	tbl_attach					
테이블 설명	공지사항 게시판 board 테이블에 업로드한 파일을 외래키로 연결					
컬럼명	컬럼ID	타입 및 길이	Null	PK	FK	기본 값/ EXTRA
게시물 번호	no	int(11)	YES		tbl_attach_no(board)	
파일명	filename	longtext	YES			
등록일	regdate	timestamp	YES			CURRENT_TIMESTAMP

tbl_attach

- 공지사항 게시판에 업로드한 파일을 no를 외래키로 연결


```
<button class="btn btn-dark btn-lg btn-block" type="submit" style="padding-right: 1%;">로그인</button>
<button class="btn btn-light btn-lg btn-block" id="join">회원가입</button>
```

LoginpageGet.jsp

```
<input class="btn btn-secondary btn-lg btn-block my-1" type="submit" value="회원가입" id="join">
```

Memberjoin.jsp

```
@RequestMapping(value = "/LoginpageGet", method = RequestMethod.GET)//로그인 화면
public void LoginpageGet() throws Exception {

    Logger.info("login get...");

}

@RequestMapping(value = "/LoginpagePost", method = RequestMethod.POST)//로그인 화면
public void LoginpagePost(LoginVO lvo, Model model) throws Exception {

}

@RequestMapping(value = "/Memberjoin", method = RequestMethod.GET)//회원가입 화면
public void MemberjoinGet() throws Exception {

    Logger.info("mj get...");

}

@RequestMapping(value = "/Memberjoin", method = RequestMethod.POST)//회원가입 화면
public String MemberjoinPost(Member_infoVO mivo) throws Exception {

    Logger.info("mj post...");
    Logger.info("mivo = " + mivo);

    lservice.Memberjoin(mivo);

    return "login/LoginpageGet";
}
```

LoginInterceptor.java

```
<!-- 로그인 -->
<select id="Loginpage" resultType="com.cvs.model.Member_infoVO">

    select * from member_info where id = #{id} and pw = #{pw}

</select>

<!-- 회원가입 -->
<insert id="Memberjoin">
    insert into member_info(member_name , id, pw, email,gender,age )
    values(#{member_name}, #{id}, #{pw}, #{email}, #{gender}, #{age} )
</insert>
```

LoginMapper.xml

```
public Member_infoVO Loginpage(LoginVO lvo)throws Exception;

public void Memberjoin(Member_infoVO mivo)throws Exception ;
```

LoginController.java

LoginService.java

```
@Override
public Member_infoVO Loginpage(LoginVO lvo) throws Exception {

    Member_infoVO mivo = session.selectOne(namespace+".Loginpage",lvo);

    System.out.println("mapper = "+mivo);

    return mivo;
}

@Override
public void Memberjoin(Member_infoVO mivo) throws Exception {
    System.out.println("mapper");
    session.insert(namespace + ".Memberjoin", mivo);
}
```

LoginMapperImpl.java

```
public Member_infoVO Loginpage(LoginVO lvo)throws Exception;

public void Memberjoin(Member_infoVO mivo)throws Exception ;
```

LoginMapper.java

```
@Autowired
private LoginMapper lm;

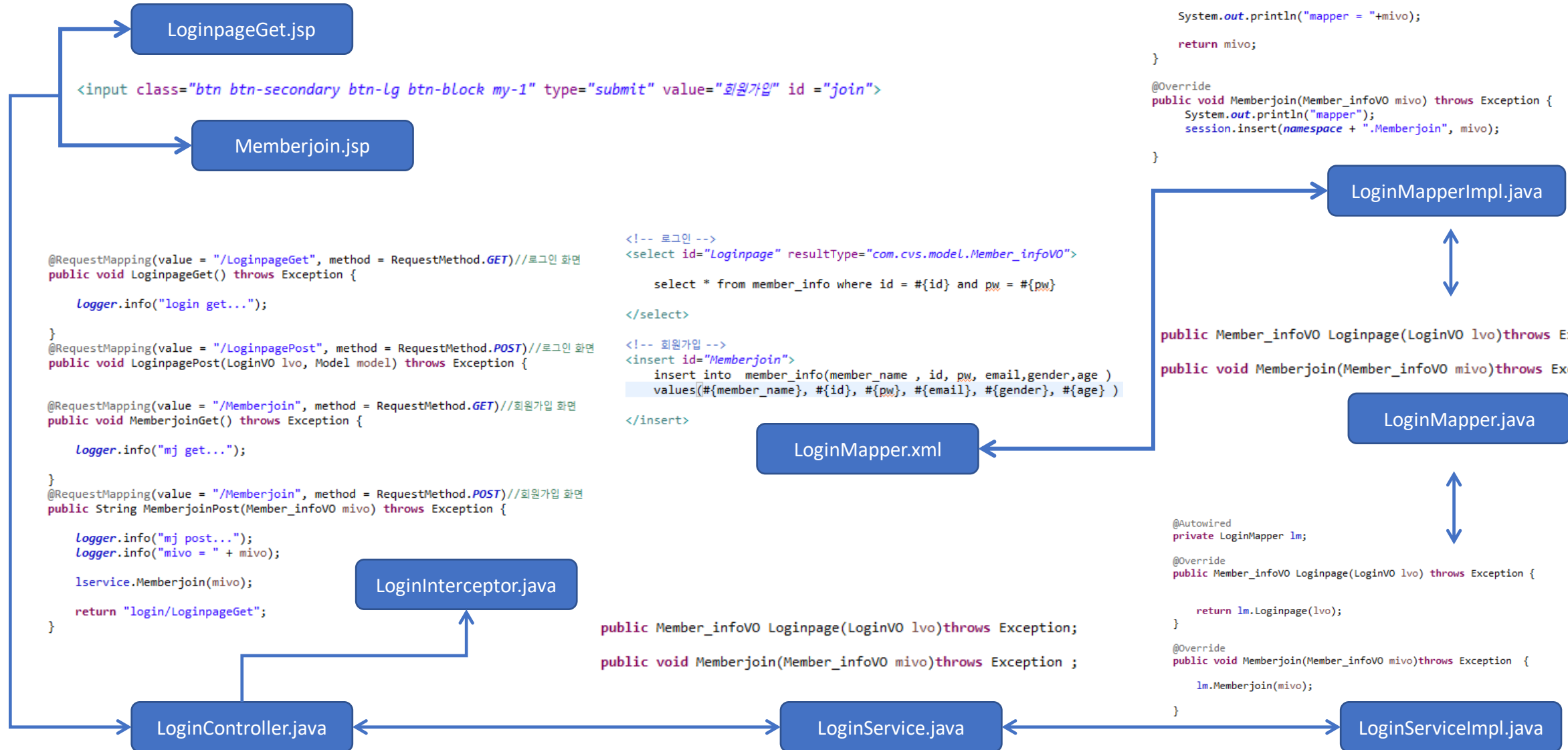
@Override
public Member_infoVO Loginpage(LoginVO lvo) throws Exception {

    return lm.Loginpage(lvo);
}

@Override
public void Memberjoin(Member_infoVO mivo)throws Exception {

    lm.Memberjoin(mivo);
}
```

LoginServiceImpl.java



```
<button id=md_update>상품등록</button> <tr class="payment">
<td class="paymentList">판매내역</td> <td colspan="2">현금</td>
<td colspan="2">카드</td>
</tr>
```

```
<!-- 상품 바코드 등록 -->
<select id="inputMdlist" resultType="com.cvs.model.Md_infoVO">

    select md_info.barcode_no,md_name, price,
    promotion_no, pr_type, pr_ea, pr_subject, pr_content, pr_family, pr_price

    from md_info join md_price on md_info.md_no = md_price.md_no
    left join promotion on md_info.barcode_no = promotion.barcode_no
    where md_info.barcode_no=#{barcode_no};

</select>

<!-- 결제 -->
<insert id="Payment">
    <![CDATA[
        insert into pos_board(list_no,member_no, barcode_no, md_ea, sale_price, type,
        values(

            (select count(*) as cnt from auto_increment_info
            list_no where list_no = member_no),

            #{member_no},#{barcode_no},#{md_ea},#{sale_price}, #{type}, sysdate(),cu

        ]]]>
</insert>

<!-- 판매 리스트 top -->
<select id="PaymentList" resultType="com.cvs.model.Pos_boardVO">
    <![CDATA[
        SELECT pos_board.list_no, list_md_ea, total_price, pos_board.type, pos_board.sale_date,
        pos_board.sale_time
        FROM md_info JOIN md_price
        ON md_info.md_no = md_price.md_no
        JOIN pos_board
        ON md_info.barcode_no = pos_board.barcode_no
        join (SELECT list_no, sum((price-sale_price)*md_ea) total_price, sum(md_ea)
        FROM md_info
        JOIN md_price
        ON md_info.md_no = md_price.md_no
        JOIN pos_board
        ON md_info.barcode_no = pos_board.barcode_no
        group by list_no
        ) total
        on total.list_no = pos_board.list_no
        where sale_date = #{sale_date} and member_no = #{member_no}
        group by pos_board.list_no, pos_board.sale_date, list_md_ea,
        pos_board.type, total_price, sale_date, sale_time
        order by pos_board.list_no desc;

    ]]]>
</select>
```

```
@Override
public Md_infoVO inputMdlist(int barcode_no) throws Exception {
    Md_infoVO mdlist = session.selectOne(namespace + ".inputMdlist", barcode_no);
    return mdlist;
}

@Override
public void Payment(Pos_boardVO pvo) throws Exception {
    session.insert(namespace + ".Payment", pvo);
}

@Override
public List<Pos_boardVO> PaymentList(String sale_date, int member_no) throws Exception {
    Map<String, Object> paramMap = new HashMap<String, Object>();
    paramMap.put("sale_date", sale_date);
    paramMap.put("member_no", member_no);

    List<Pos_boardVO> PaymentList = session.selectList(namespace + ".PaymentList", paramMap);
    return PaymentList;
}

public Md_infoVO inputMdlist(int barcode_no) throws Exception;
public void Payment(Pos_boardVO pvo) throws Exception;
public List<Pos_boardVO> PaymentList(String sale_date, int member_no) throws Exception;

@Override
public Md_infoVO inputMdlist(int barcode_no) throws Exception {
    Md_infoVO mdlist = pm.inputMdlist(barcode_no);
    return mdlist;
}

@Override
public void Payment(Pos_boardVO pvo) throws Exception {
    pm.Payment(pvo);
}

@Override
public List<Pos_boardVO> PaymentList(String sale_date, int member_no) throws Exception {
    return pm.PaymentList(sale_date, member_no);
}
```

```
@RequestMapping(value = "/posPayment", method = RequestMethod.GET)//pos화면
public void posGet(Functionkey_infoVO fivo,Buttonpage bp, Model model) throws Exception {
    logger.info("pos get...");
    model.addAttribute("functionkeyList", pservice.functionButton(bp));
}

@ResponseBody
@RequestMapping(value = "/posList", method = RequestMethod.GET)//상품리스트
public Md_infoVO posList(@RequestParam int barcode_no,Md_infoVO mdvo) throws Exception {
    mdvo = pservice.inputMdlist(barcode_no);
    logger.info("mdvo="+mdvo);
    logger.info("barcode_no = "+barcode_no);

    return mdvo;
}

@ResponseBody
@RequestMapping(value = "/Payment", method = RequestMethod.POST)//결제
public void Payment(@RequestBody Pos_boardVO pvo) throws Exception {
    pservice.Payment(pvo);
    logger.info("Payment post...");
    logger.info("getBarcode_no"+pvo.getBarcode_no());
    logger.info("getMd_ea"+ pvo.getMd_ea());
}

@RequestMapping(value = "/posPaymentList", method = RequestMethod.GET)//전체 매출 리스트 화면
public void posPaymentListGet(@RequestParam String sale_date, int member_no, Pos_boardVO pvo) throws Exception {
    logger.info("posPaymentList get...");
    logger.info("sale_date = "+sale_date);

    model.addAttribute("PaymentList", pservice.PaymentList(sale_date, member_no));
    model.addAttribute("selectdate", pservice.PaymentList(sale_date, member_no));
}
```

PosController.java

PosService.java

PosServiceImpl.java

PosMapperImpl.java

PosMapper.java

PosMapper.xml

```
<div class="tab-pane fade" id="select_type" role="tabpanel" aria-labelledby="select_type_page">
  <jsp:include page="select_MD_Type.jsp"/>
</div>
```

OrderMain.jsp

typePage.jsp

typePage.js

```
@RequestMapping(value = "/OrderMain", method = RequestMethod.GET)//order main화면
public void mdOrderGet() throws Exception {

    Logger.info("mdOrder get...");

}

@ResponseBody
@RequestMapping(value = "/orderList", method = RequestMethod.GET)//발주 리스트 화면
public ResponseEntity<List<Md_infoVO>> orderListGet(@RequestParam String member_no, String type,

ResponseEntity<List<Md_infoVO>> entity = null;

Logger.info("orderList get...");
List<Md_infoVO> orderList = oservice.orderList(member_no, type, detail_type, md_name);

Logger.info("type = "+type);
Logger.info("detail_type = "+detail_type);
Logger.info("md_name = "+md_name);

entity = new ResponseEntity<List<Md_infoVO>>(orderList, HttpStatus.OK);
Logger.info("entity = "+entity.toString());

return entity;

}

@ResponseBody
@RequestMapping(value = "/MDorder", method = RequestMethod.POST)//상품발주
public void MDorder(@RequestBody Order_boardVO obvo) throws Exception {

    Logger.info("MDorder post...");

    oservice.MDorder(obvo);
    Logger.info("mivo = " + obvo);

}
```

OrderController.java

```
<select id="orderList" resultType="com.cvs.model.Md_infoVO">

SELECT md_info.barcode_no, md_img, type, detail_type, company, md_name, member_t.order_ea ,
member_t.approval_ea, cost, price, shelf_life, md_img FROM
md_info left join order_board on md_info.barcode_no = order_board.barcode_no
left join md_price on md_info.barcode_no = md_price.barcode_no
left join (select barcode_no, order_ea, approval_ea from order_board where member_no = #{me
on md_info.barcode_no = member_t.barcode_no
where type = #{type} and
detail_type like concat('%',{detail_type},%') and
md_name like concat('%',{md_name},%') group by
md_info.barcode_no, md_img, type, detail_type, company, md_name, member_t.order_ea,
member_t.approval_ea, cost, price, shelf_life, md_img
order by md_name asc

</select>

<insert id="MDorder">

<choose>
<when test="order_ea != 0">
<choose>
<when test="count != 0">
UPDATE order_board
SET order_ea = #{order_ea} where member_no=#{member_no} and barcode_no=#{barcode_no};
</when>
<otherwise>
insert into order_board (member_no, barcode_no, order_ea)
values (#{member_no},#{barcode_no}, #{order_ea})
</otherwise>
</choose>
</when>
<when test="order_ea == 0">
DELETE FROM order_board
WHERE barcode_no=#{barcode_no} and member_no = #{member_no};
</when>
</choose>
</insert>
```

OrderMapper.xml

```
public List<Md_infoVO> orderList(String member_no, String type,

public void MDorder(Order_boardVO obvo) throws Exception;
```

OrderService.java

```
@Override
public List<Md_infoVO> orderList(String member_no, String type, String detail_type, String r

Map<String, Object> paramMap = new HashMap<String, Object>();
paramMap.put("member_no", member_no);
paramMap.put("type", type);
paramMap.put("detail_type", detail_type);
paramMap.put("md_name", md_name);

List<Md_infoVO> orderList = session.selectList(namespace + ".orderList", paramMap);

return orderList;
}

@Override
public void MDorder(Order_boardVO obvo) throws Exception {

    session.insert(namespace + ".MDorder", obvo);

}
```

OrderMapperImpl.java

```
public List<Md_infoVO> orderList(String member_no, String type,

public void MDorder(Order_boardVO obvo) throws Exception;
```

OrderMapper.java

```
@Override
public List<Md_infoVO> orderList(String member_no, String type, String detail_type,

return om.orderList(member_no, type, detail_type, md_name);
}

@Override
public void MDorder(Order_boardVO obvo) throws Exception {

    om.MDorder(obvo);

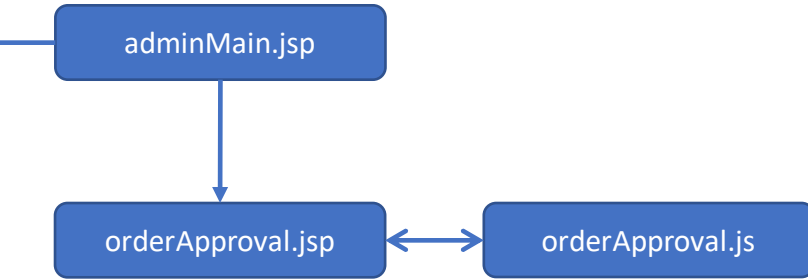
}
```

OrderServiceImpl.java

```
<div class="tab-pane fade" id="List-orderApproval" role="tabpanel" aria-labelledby="List-orderApproval-list">
  <jsp:include page="orderApproval.jsp"/>
</div>
```

```
@Override
public List<Md_infoVO> approval_list(Md_infoVO mdvo) throws Exception {

    List<Md_infoVO> approval_list = session.selectList(namespace + ".approval_list", mdvo);
    return approval_list;
}
```



```
@RequestMapping(value = "/adminMain", method = RequestMethod.GET) // 로그인 화면
public void adminMainGet() throws Exception {

    logger.info("adminMain get...");

}

@ResponseBody
@RequestMapping(value = "/approval_list", method = RequestMethod.GET) // 발주승인
public ResponseEntity<List<Md_infoVO>> approval_list(Md_infoVO mdvo) throws Exce

    ResponseEntity<List<Md_infoVO>> entity = null;
    logger.info("approval_list GET...");
    List<Md_infoVO> approval_list = bservice.approval_list(mdvo);

    entity = new ResponseEntity<List<Md_infoVO>>(approval_list, HttpStatus.OK);
    logger.info("approval_list entity = " + entity.toString());
    return entity;
}
```

```
<select id="approval_list" resultType="com.cvs.model.Md_infoVO">
  select order_board.member_no, total_order_ea.total_order_ea, sum(cost) total_cost, sum(
  order_board left join (select member_no, sum(order_ea) total_order_ea from order_bo
  total_order_ea on total_order_ea.member_no = order_board.member_no
  left join md_info on order_board.barcode_no = md_info.barcode_no
  left join md_price on order_board.barcode_no = md_price.barcode_no
  where total_order_ea.total_order_ea > 0
  group by order_board.member_no;
</select>
```

AdminMapper.xml

```
public List<Md_infoVO> approval_list(Md_infoVO mdvo) throws Exception;
```

AdminMapper.java

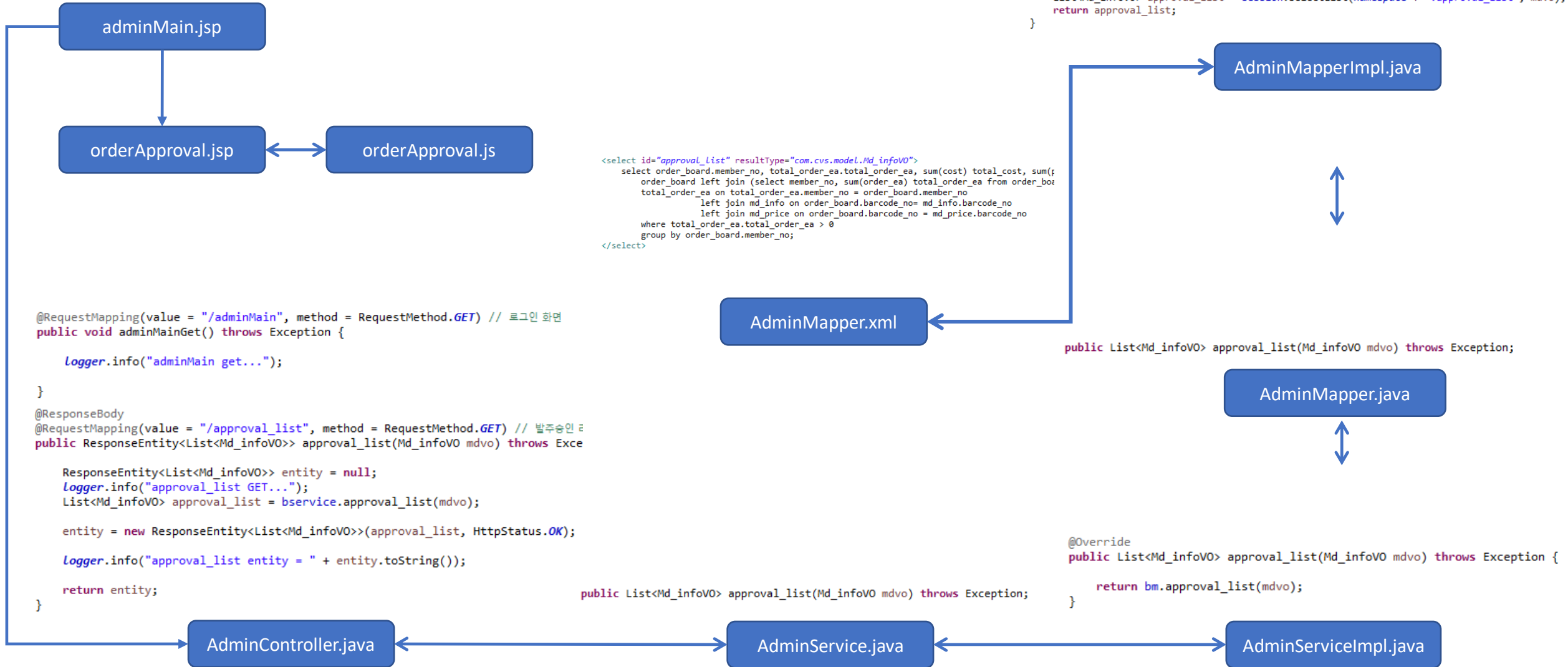
```
@Override
public List<Md_infoVO> approval_list(Md_infoVO mdvo) throws Exception {

    return bm.approval_list(mdvo);
}
```

AdminController.java

AdminService.java

AdminServiceImpl.java



테스트 케이스 시나리오								
테스트 시나리오ID	테스트 시나리오	테스트 케이스 ID	테스트 케이스 상세 설명	사전 조건	테스트 데이터	예상 결과	결과 확인 방법	PASS / FAIL
pos	pos 회원가입	pos_회원가입_01	pos시스템 회원 생성		ID : test PW : 123	pos시스템 회원 가입	로그인	PASS
	pos 로그인	pos_로그인_01	회원가입이 제대로 되었는지 로그인을 통해 검증	회원가입	ID : test PW : 123	pos시스템 로그인	ID,PW 입력 후 로그인 버튼 클릭	PASS
	pos 상품판매	pos_상품리스트등록_01	키보드로 바코드 입력 후 상품리스트 등록		바코드번호 : 88010430	상품 리스트에 쌀새우깡 등록	바코드 번호 입력 후 상품등록 버튼 클릭	PASS
		pos_상품리스트등록_02	평선키 등록		콜	평선키에 펍시콜라 등록	1)키설정 클릭 -> 2)등록을 원하는 위치 클릭 -> 3)상품 검색 -> 4)원하는 상품 클릭	PASS
		pos_상품리스트등록_03	평선키로 상품리스트에 등록	평선키 등록	월드콘	상품 리스트에 월드콘 등록	월드콘으로 입력된 평선키 클릭	PASS
		pos_상품리스트등록_04	숫자패드로 바코드 입력 후 상품리스트 등록		바코드번호 : 88010430	상품 리스트에 쌀새우깡 등록	숫자패드에서 해당하는 숫자 클릭 후 상품등록 버튼 클릭	PASS
		pos_상품 수량 변경_01	수량 버튼으로 리스트에 등록된 상품의 수량변경	상품 등록	스크류바	리스트에 등록된 스크류바 수량 1증가	1)리스트에서 해당 상품 클릭 -> 2)오른쪽 상단의 수량 버튼 클릭	PASS
		pos_상품 수량 변경_02	숫자패드의 수량변경 버튼으로 리스트에 등록된 상품의 수량변경	상품 등록	스크류바	리스트에 등록된 스크류바 수량 8로 증가	1)리스트에서 해당 상품 클릭 -> 2)오른쪽 하단숫자패드에서 원하는 숫자 클릭(8) -> 3)수량변경 버튼 클릭	PASS
		pos_상품 삭제_01	삭제 버튼으로 리스트에 등록된 상품 삭제	상품 등록	스크류바	리스트에 등록된 스크류바 삭제	1)리스트에서 해당 상품 클릭 -> 2)오른쪽 상단의 삭제 버튼 클릭	PASS
		pos_상품 할인_01	행사상품의 경우 개수만큼 등록하여 가격 할인	상품 등록	칭따오, 호가든, 기네스, 아사히	정가에서 가격이 할인되어 상품의 가격이 각각 2500원으로 변경	해당 행사 개수만큼 상품을 입력(칭따오, 호가든, 기네스, 아사히)	PASS
		pos_상품 할인_02	통신사 할인 버튼으로 상품할인	상품 등록	쌀새우깡 2개, 코카콜라	행사가 적용 된 쌀새우깡 2개의 가격을 제외하고, 코카콜라 1개 통신사 할인적용	1)상품등록 -> 2)통신사할인 버튼 클릭 -> 3)통신사와 전화번호 입력 후 검색-> 4)해당 회원정보 클릭 -> 5)합계 테이블에서 할인금액 확인	PASS
		pos_상품 할인_03	통신사 할인 적용 취소	통신사 할인	할인 취소	통신사 할인으로 할인된 금액 취소	할인 취소 버튼 클릭	PASS
		pos_상품 결제	등록된 상품 결제	상품 등록	월드콘, 빠빠꼬, 스크류바	등록된 상품에 대한 정보다 DB에 저장되어 화면 초기화	오른쪽 하단 숫자패드의 현금 or 카드 버튼 클릭	PASS

테스트 케이스 시나리오								
테스트 시나리오 ID	테스트 시나리오	테스트 케이스 ID	테스트 케이스 상세 설명	사전 조건	테스트 데이터	예상 결과	결과 확인 방법	PASS / FAIL
	pos 판매내역	pos_판매내역_01	오늘 판매한 상품의 리스트 출력	상품 결제	판매내역	오늘 판매한 거래내역(거래번호, 거래시간, 상품 수, 결제 타입, 매출)출력	오른쪽 하단 숫자패드의 판매내역 버튼 클릭	PASS
		pos_판매내역_02	날짜별 판매내역 출력	상품 결제	2020-02-14	해당 날짜에 판매한 거래내역(거래번호, 거래시간, 총 상품 수, 결제 타입, 총 매출)출력	왼쪽 상단 달력을 이용하여 원하는 날짜(2020-02-14) 클릭	PASS
		pos_판매내역_03	판매내역 상세보기	판매내역 출력	거래번호 98	해당 거래번호의 상세 정보 (상품명, 수량, 상품 수, 금액, 할인, 합계) 출력	출력된 판매내역 중 원하는 거래 리스트 클릭	PASS
		pos_환불	결제가 완료된 상품 환불	상품 결제	거래번호 91	91번 거래내역 삭제	1)판매내역 리스트 중 원하는 거래 리스트 클릭 -> 2)오른쪽 상단 환불 버튼 클릭	PASS
	pos 상품발주	pos_공지사항_01	공지사항 게시판 출력	상품발주 페이지로 접속		공지사항 테이블(글번호, 제목, 작성자, 작성일자,조회수) 출력	메인페이지에서 상품발주 버튼 클릭 후 첫번째 페이지인 공지사항 확인 or 왼쪽 탭에서 공지사항 탭 클릭하여 공지사항 확인	PASS
		pos_공지사항_02	공지사항 상세 내용 출력	공지사항 페이지로 접속	글번호8	글번호 8에 관한 상세정보 (글번호, 제목, 내용, 작성자,파일, 조회수) 출력	공지사항 테이블 중 원하는 게시글의 제목을 클릭	PASS
		pos_상품발주 리스트_01	상품발주 리스트 출력	상품발주 페이지로 접속	상온식품	해당 상품타입에 해당하는 상품 리스트 출력(바코드번호, 상품타입, 제조사, 상품명, 원가, 정가, 유통기한, 예정 수량, 발주 수량)	1)왼쪽 탭에서 상품발주 탭 클릭 -> 2)상단의 원하는 네이비바(상온식품)클릭	PASS
		pos_상품발주 리스트_02	셀렉트 박스를 이용한 상품 세부타입 출력	상품발주 탭 클릭	신선식품 -> 샌드위치	흑당샌드위치와 관련된 정보리스트 출력 (88010440, 샌드위치, 롯데제과, 흑당샌드위치, 600, 1300, 7, 0)	셀렉트 박스에서 원하는 상품 세부타입 선택	PASS
		pos_상품발주 리스트_03	검색창을 이용한 상품 세부타입 검색	상품발주 탭 클릭	신선식품 -> 전체(셀렉트 박스)-> 버(검색창)	불고기버거와 관련된 정보리스트 출력 (88010443, 햄버거, 롯데푸드, 불고기버거, 700, 1500, 7, 0, 0)	텍스트 박스에서 특정 단어 입력 후 검색 버튼 클릭	PASS
		pos_상품발주	원하는 상품 발주	상품발주 탭 클릭	햄버거 -> 발주수량 3	햄버거 3개 발주 등록 후 DB에 저장	원하는 상품의 발주수량 수정 후 오른쪽 하단 적용 버튼 클릭	PASS
		pos_검품	상품을 전달받고 해당 상품 검품	검품 탭 클릭	햄버거 3개	검품 게시판 목록에 있는 햄버거 상품제거	원하는 상품의 검품완료 승인버튼 클릭	PASS
	pos관리자	pos_관리자 페이지	Interceptor기능을 이용해 관리자 페이지 접속	관리자 계정 생성	ID : admin PW : 123	기존의 메인화면이 아닌 관리자 페이지로 접속	로그인 페이지에서 관리자 계정으로 로그인	PASS
		pos_관리자 공지사항_01	공지사항 게시판 글쓰기	관리자 페이지로 접속	제목 : 세계맥주 묶음할인 프로모션	게시판에 "세계맥주 묶음할인 프로모션" 게시글이 56번으로 등록	글쓰기 버튼 클릭 후 제목, 내용, 작성자, 첨부파일 등록 후 글쓰기 버튼 클릭	PASS
		pos_관리자 공지사항_02	공지사항 게시판 글수정	게시물 등록	56번 게시물 제목 -> 아이스크림 2+1 프로모션	56번 게시물의 제목이 아이스크림 2+1 프로모션으로 변경	게시물 제목 클릭 후 제목, 내용, 첨부파일 수정 후 글수정 버튼 클릭	PASS
		pos_관리자 공지사항_03	공지사항 게시판 글삭제	게시물 등록	56번 게시물 -> 삭제	56번 게시물 삭제	게시물 제목 클릭 후 글삭제 버튼 클릭	PASS
		pos_상품등록	상품등록		육개장 사발면 등록	상품 판매가 가능한 새우깡 상품 등록	상품등록 탭 클릭 후 바코드번호, 타입, 세부타입, 제조사, 상품명, 원가, 정가, 유통기한, 사진 등록 후 상품등록 버튼 클릭	PASS
		pos_발주승인	사용자가 원하는 상품 발주 후 발주승인	사용자의 상품발주	2번 매장 새우깡 2개	발주승인 게시물의 새우깡이 삭제되고, 사용자가 발주한 상품의 개수는 0개, 예정수량은 2개로 변경	발주승인 탭에서 발주승인 버튼 클릭 후 사용자 ID로 로그인 후 발주 승인 확인	PASS

감사합니다.

GitHub: <https://github.com/roksang-code/cvsProject>

GitHub의 WEB_POS 사이트 주소 및 사용설명을 확인해주세요