



결제, 마이페이지, 최근 본 상품  
(신화원)

# CONTENTS

제주도 숙박예약홈페이지 제작

## I. 개요

## II. 시스템 구성도

## III. 메뉴별 소개

- 1) 데이터베이스 설계
- 2) 메뉴별 순서도
- 3) 구현

## IV. 소감문



## I. 개요

---

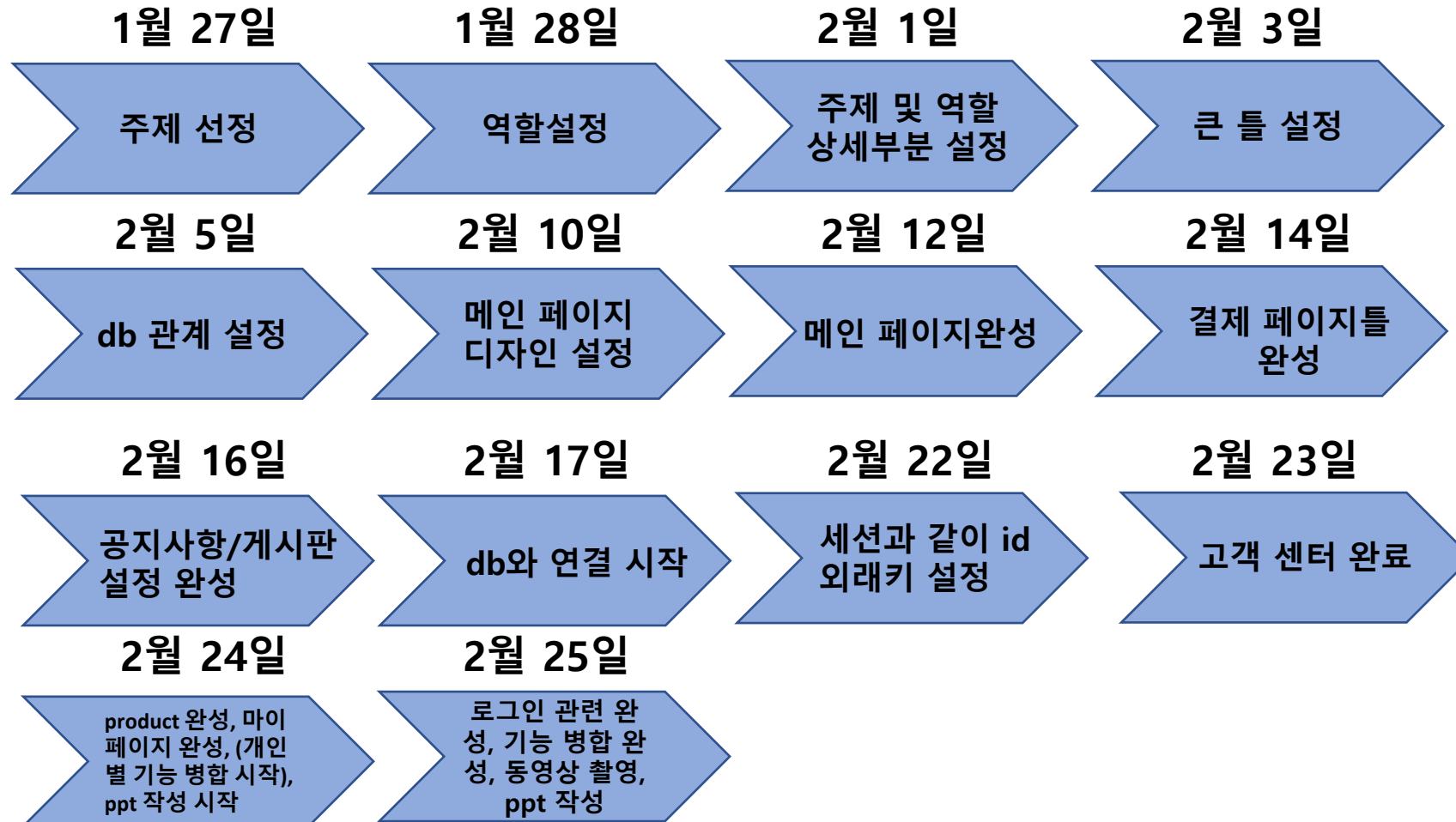
### 프로젝트명

jdbc를 활용하여 client와 server, database를 연결하는  
숙박시설 예약 사이트 구축

### 프로젝트 개요

client(브라우저)와 Server(tomcat, java), DataBase  
server(MySQL)을 연동하여 숙박시설 예약과 결제,  
상품등록 및 관리, 게시판 등록 및 관리할 수 있는 site를  
구축하기

## I. 개발과정 > 개발기간



## I. 개요

---



## I. 개요

### 개발 과정 >> 제약조건 : reservation

컬럼명	DataType	제약조건	내용
res_num	Int(11)	Not null Primary key	예약 번호
res_id	Varchar(100)	Not null Foreign key	예약 아이디
res_name	Varchar(20)	Not null	예약자이름
res_tel	Varchar(20)	Not null	예약자 휴대폰번호
res_date	date	Not null	예약일
res_payment	Varchar(20)	Not null	결제 타입
res_room	Varchar(20)	Not null	예약 상품
res_cost	Int(11)	Not null	예약 금액



## I. 개요

개발 과정 >> 제약조건 : shoppingbag

컬럼명	DataType	제약조건	내용
no	Int(11)	Not null Primary key	장바구니 넘버
id	Varchar(100)	Not null Foreign key	사용자 아이디
product	Varchar(100)	Not null	상품페이지주소
product_name	Varchar(20)	Not null	상품명

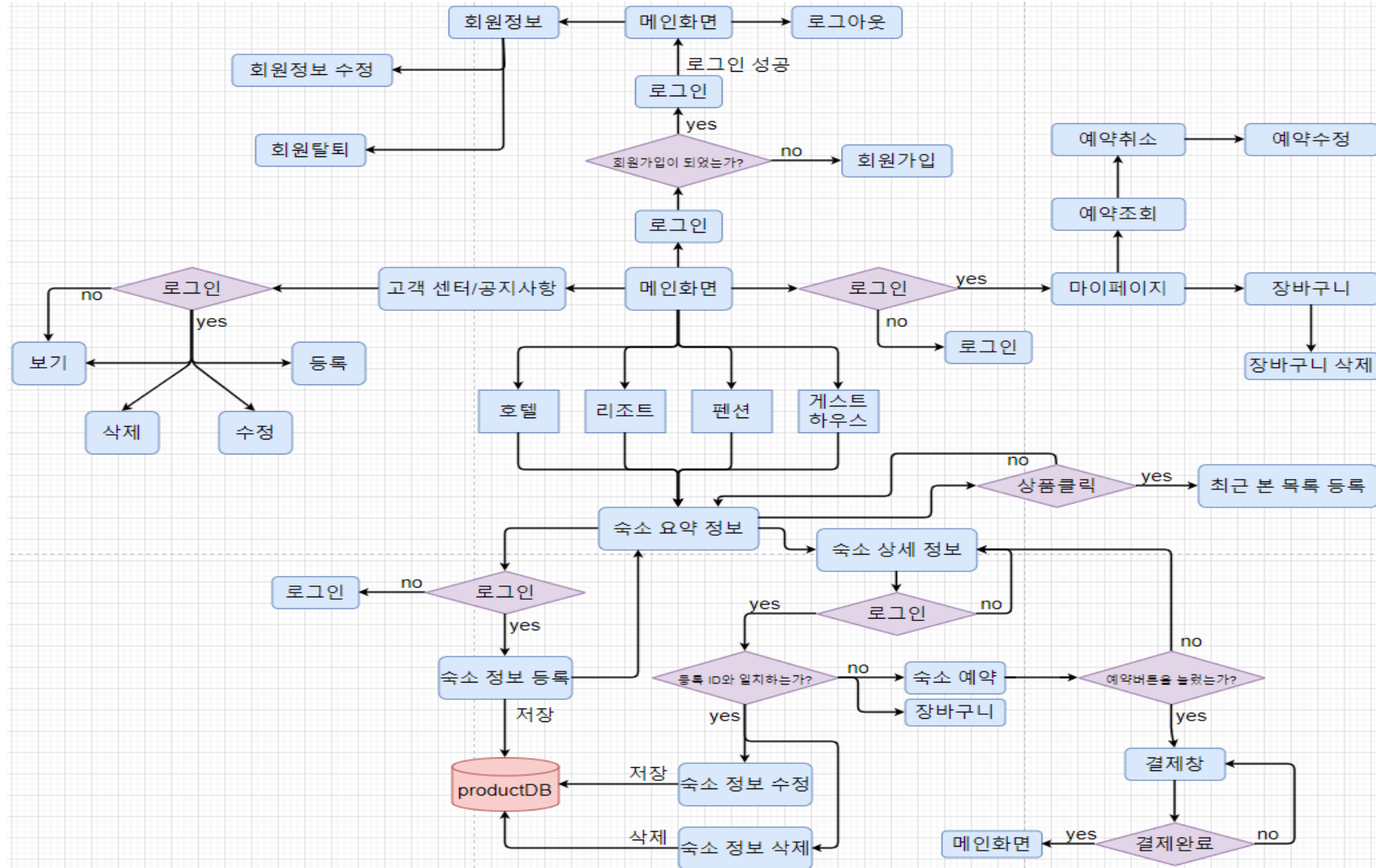
## I. 개요

개발 과정 >> 제약조건 : notice\_board(공지사항)

컬럼 명	Data Type	제약 조건	설명
Notice_no	Int(11)	Not Null Primary Key	등록번호
Notice_title	Varchar(100)	Not Null	제목
Notice_date	Varchar(100)	Not Null	등록 날짜
Notice_writer	Varchar(100)	Not Null Foreign Key	작성자
Notice_file	Varchar(100)	Not Null	첨부파일
Notice_content	Varchar(100)	Not Null	내용

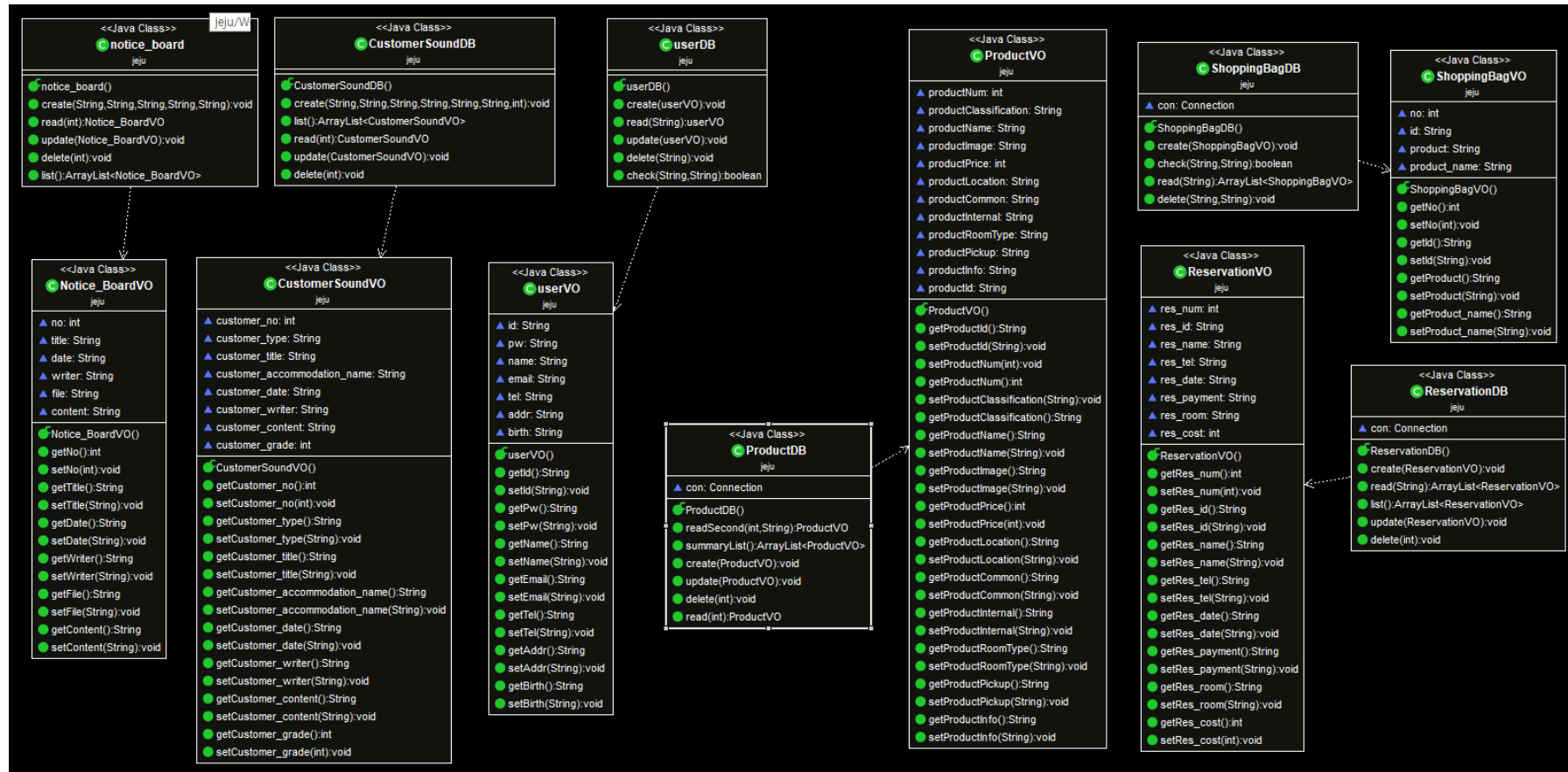


## I. 개요-전체메뉴 순서도

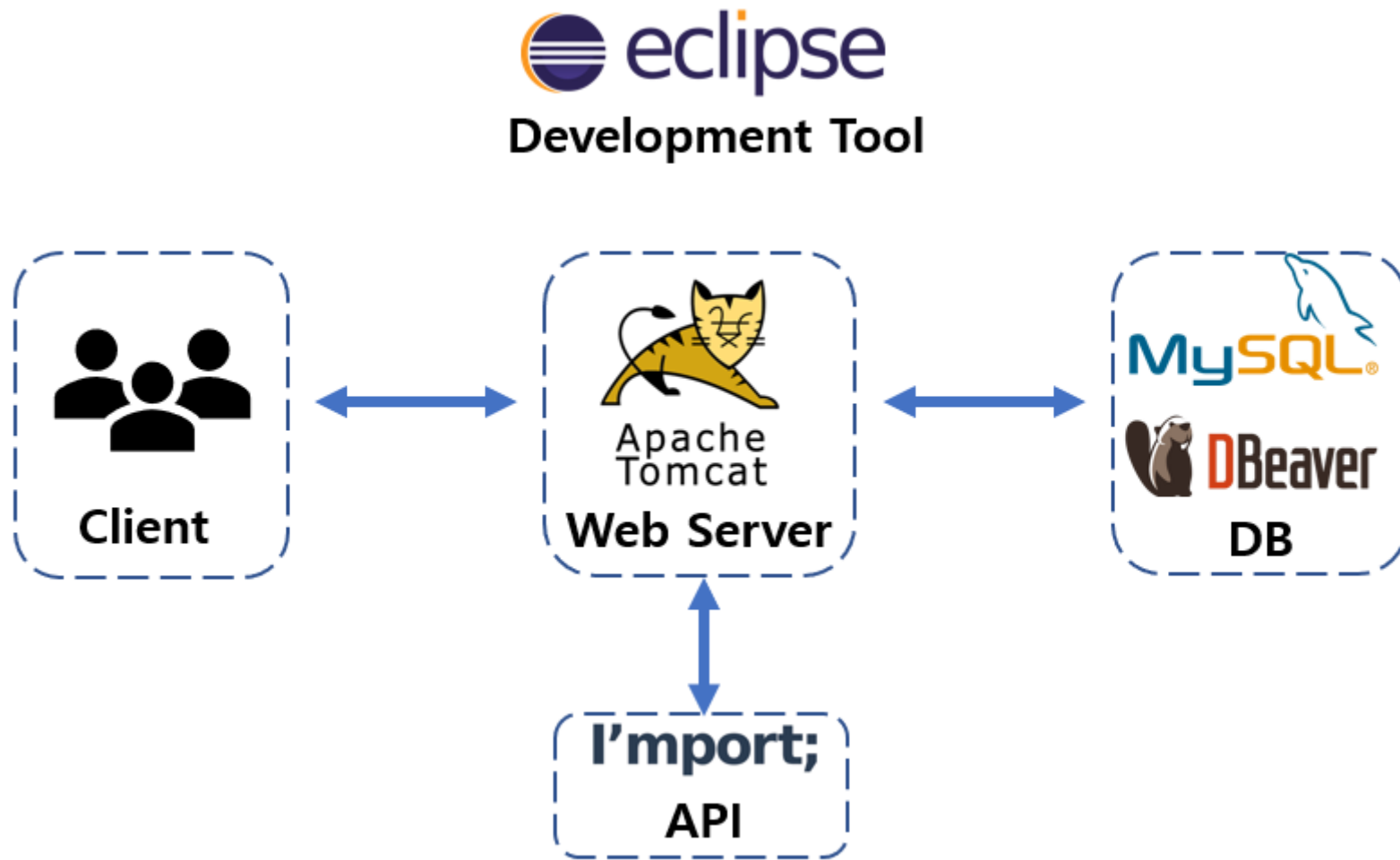


# III. 메뉴별 소개 : 전체메뉴

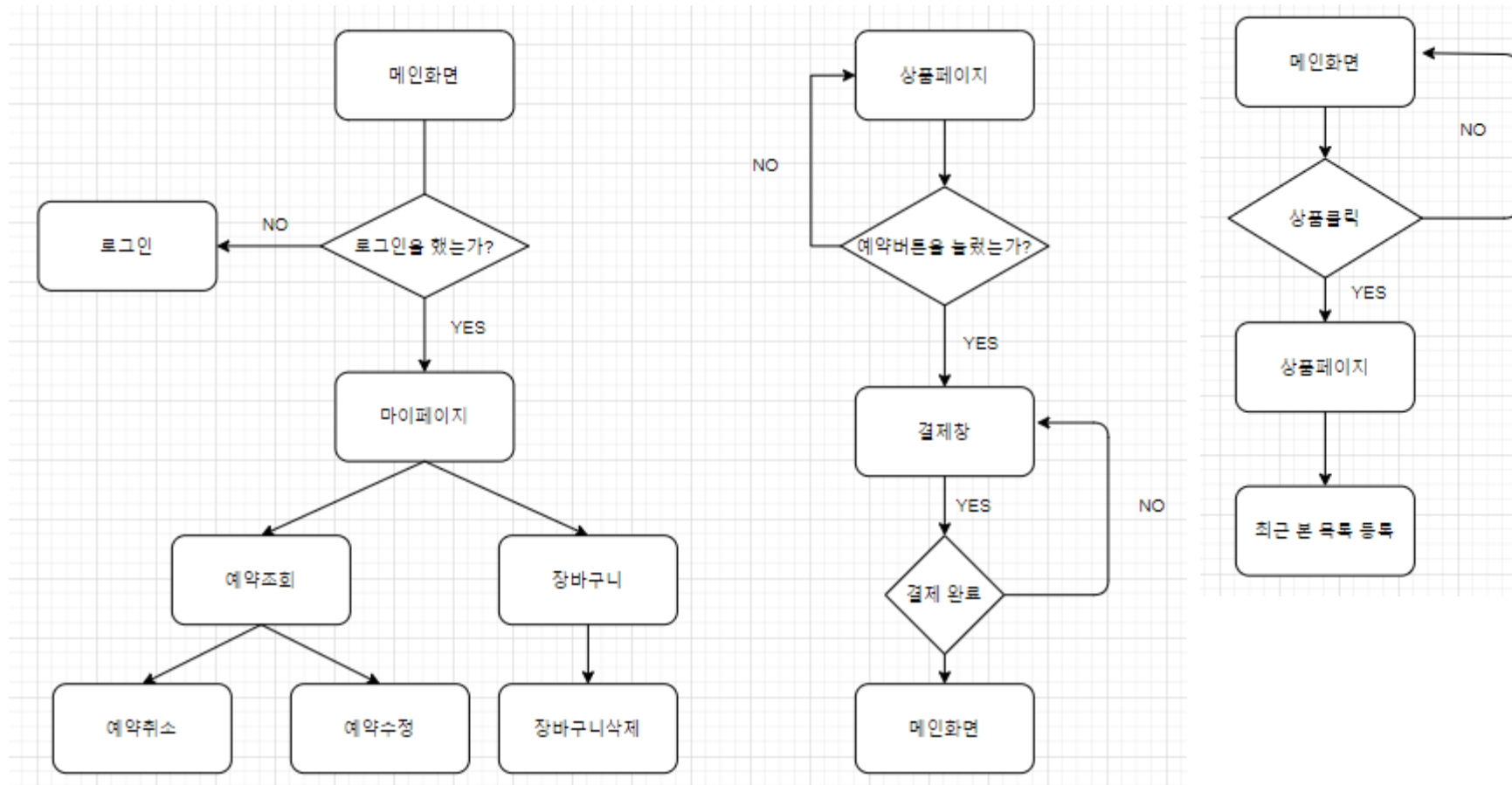
## 전체 UML



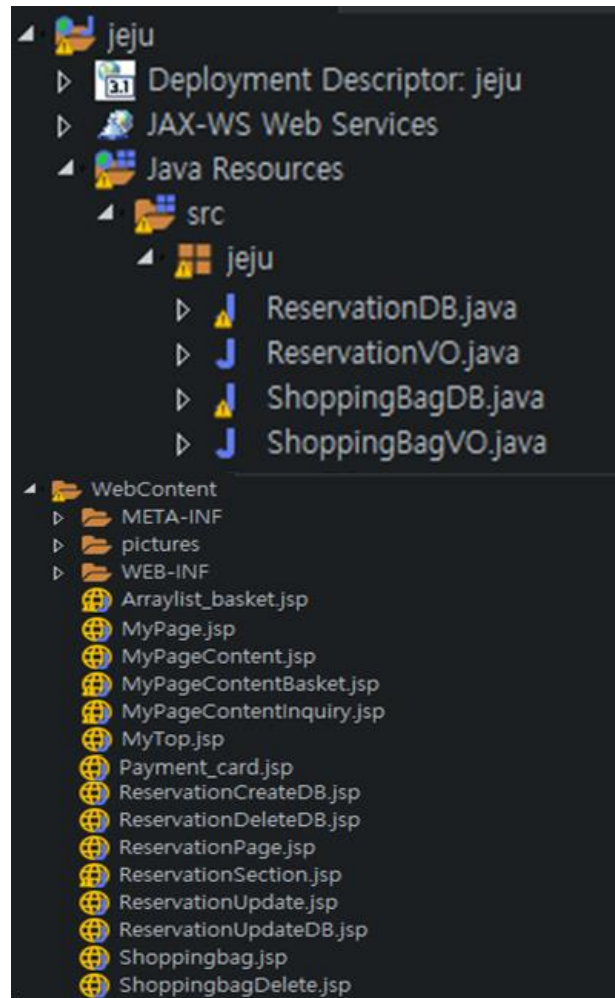
## II. 시스템 구성도



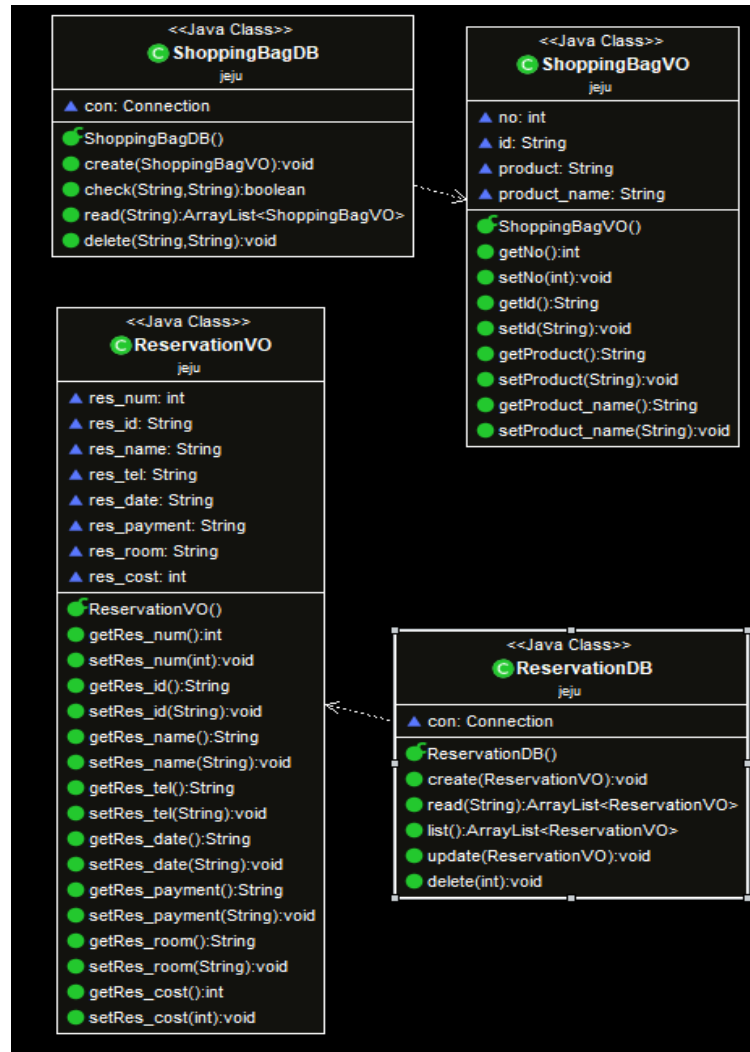
### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 순서도



### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 프로젝트 트리



### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - UML



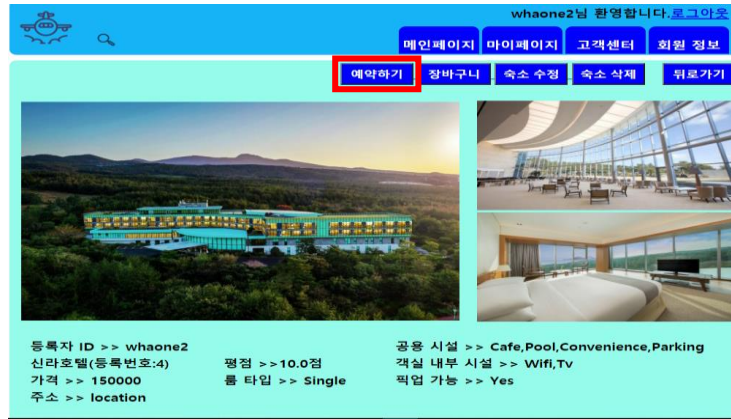
### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

---

## 결제



### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원



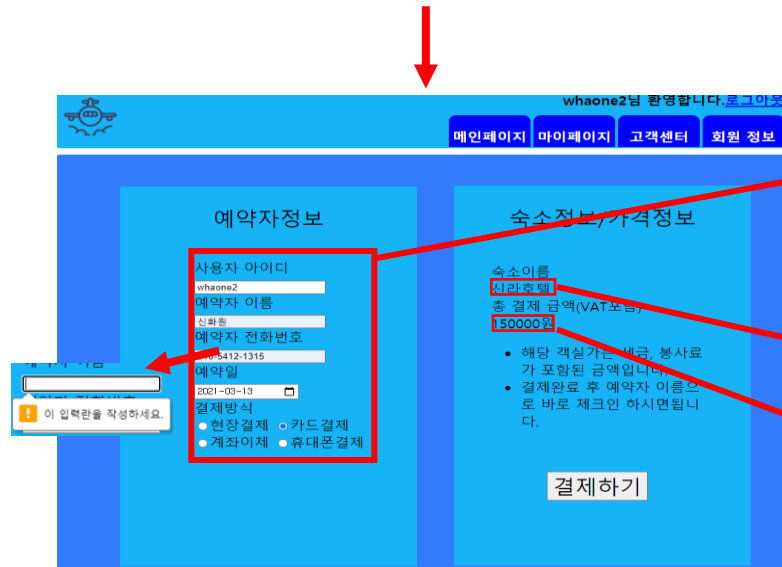
```
<a href="ReservationPage.jsp?res_room=<%=bag.getProductN<%=bag.getProductPrice(%)>">
<button class="button">예약하기</button>
```

예약하기 버튼을 누르면 결제화면(ReservationPage.jsp)으로 방이름(res\_room),방가격(res\_cost)를 넘겨준다.

① localhost:8888/jeju/ReservationPage.jsp?res\_room=신라호텔&res\_cost=150000

```
String res_room = request.getParameter("res_room");
String res_cost = request.getParameter("res_cost");

<!-- 예약정보를 다 작성하고 예약버튼을 눌러 정보를 예약테이블에 insert해주는 db처리 페이지로 넘겨준다 -->
<form action="ReservationCreateDB.jsp" id = "reservationform">
    사용자 아이디<br>
    <input name="res_id" value="<%=session.getAttribute("id")%>" readonly="readonly"><br>
    예약자 이름<br>
    <input name="res_name" required="required"><br><!-- required : 반드시 채워야함 -->
    예약자 전화번호<br>
    <input name="res_tel" required="required"><br>
    예약일 <br>
    <input type="date" name="res_date" required="required"><br>
    결제방식<br>
    <input type="radio" name="res_payment" value="place" required="required">현장결제
    <input type="radio" name="res_payment" value="card" required="required">카드결제<br>
    <input type="radio" name="res_payment" value="transfer" required="required">계좌이체
    <input type="radio" name="res_payment" value="phone" required="required">휴대폰결제<br>
    <br>
    <input type="hidden" name="res_room" value = "<%=res_room%>">
    <%=res_room%>
    <br>
    총 결제 금액 (VAT포함)<br>
    <input type="hidden" name="res_cost" value = "<%=res_cost%>">
    <%=res_cost%>원
```



ReservationPage.jsp

결제에 필요한 정보를 입력하여 db처리 페이지(ReservationCreateDB.jsp)로 넘겨주고 방이름 및 가격은 화면에 보여준다.

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

```

<%
/* 예약장(ReservationSection.jsp)에서 입력한 모든 예약정보를 받아온다. */
String res_id = request.getParameter("res_id");
String res_name = request.getParameter("res_name");
String res_tel = request.getParameter("res_tel");
String res_date = request.getParameter("res_date");
String res_payment = request.getParameter("res_payment");
String res_room = request.getParameter("res_room");
int res_cost = Integer.parseInt(request.getParameter("res_cost"));

ReservationVO bag = new ReservationVO(); // VO생성

/* 위 에서 받아온 모든 값들을 VO에 저장 */
bag.setRes_id(res_id);
bag.setRes_name(res_name);
bag.setRes_tel(res_tel);
bag.setRes_date(res_date);
bag.setRes_payment(res_payment);
bag.setRes_room(res_room);
bag.setRes_cost(res_cost);

/* create를 통해 db에 있는 예약테이블에 담는다. */
ReservationDB db = new ReservationDB();
db.create(bag); // 보낼 데이터가 많기 때문에 가방에 담아서 보내줌
%>

```

#### ReservationCreateDB.jsp

```

public void create(ReservationVO bag) throws Exception {
    String sql = "insert into reservation values (null,?,?,?,?,?,?,?)";
    PreparedStatement ps = con.prepareStatement(sql); //sql문 생성

    ps.setString(1, bag.getRes_id());
    ps.setString(2, bag.getRes_name());
    ps.setString(3, bag.getRes_tel());
    ps.setString(4, bag.getRes_date());
    ps.setString(5, bag.getRes_payment());
    ps.setString(6, bag.getRes_room());
    ps.setInt(7, bag.getRes_cost()); //int형으로 형변환 필요
    ps.executeUpdate(); //4.mysql로 생성한 sql문 보냄 -> 쿼리문 처리

    con.close(); //커넥션 해제
}

```

#### ReservationDB.java

앞서 결제화면에서 보내준 입력 값을 받아 DB와 웹서버간 데이터 교환을 원활하게 하기 위해 VO 메소드를 사용해 DAO로 보내준다.

Create 메소드에서 Insert쿼리문을 작성하여 예약정보를 예약테이블에 저장해준다.

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

```
<!-- if문 통해서 결제방식에 따라 카드결제시 카드결제 페이지로 넘겨준다. -->
<%if(bag.getRes_payment().equals("card")){ %>
    <script type="text/javascript">
        /* url을 통해 자바스크립트로 상품명과 상품가격에 대한 파라미터값 보냄 */
        /* location.href : 페이지이동메서드 */
        /* 카드결제서버로 이동 */
        /* 파라미터값으로 상품명과 상품가격을 보내준다 */
        location.href="Payment_card.jsp?room=<%=request.getParameter("res_room")%>&cost=<%=request.getParameter("res_cost")%>";
    </script>
<%
    } else { /* 카드결제 제외 예약등록 */
    %>
    <script type="text/javascript">
        alert("예약이 완료되었습니다. 결제관련해서 따로 연락드립니다.")
        location.href="MainPage.jsp"
    </script>
<%
    }
    %>
```

ReservationCreateDB.jsp

예약테이블에 입력 값이 저장된 후(db처리 후)  
결제방식에 따라 추가 처리해준다.

위와 같이 처리 후 결제 방식에 따라 조건을 주어  
카드결제일 경우 따로 i am port에서 제공하는  
API를 사용하는 서버로 넘겨준다.

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

```
<%
String room = request.getParameter("room");
String cost = request.getParameter("cost");
%>
```

넘겨준 방이름과 방가격을 받아준다.

```
15 <!-- https://www.iampor.kr/getstarted 에서 참조하기 -->
16 <!-- Line16~17 -->
17 <script type="text/javascript" src="https://code.jquery.com/jquery-1.12.4.min.js" </script>
18 <script type="text/javascript" src="https://cdn.iampor.kr/js/iampor.payment-1.1.5.js"></script>
```

lampor 사이트에서 제공하는 결제API를 사용하여 연결

```
/* getParameterByName함수란?
html파일에서 url을 통해 파라미터값을 보내고 javascript에서 값을 받기 위해 사용한 함수
*/
function getParameterByName(name) {
    name = name.replace(/\[/, "\[").replace(/\]/, "\]");
    var regex = new RegExp("[\?&]" + name + "=(^&#*)");
    results = regex.exec(location.search);
    return results === null ? "" : decodeURIComponent(results[1].replace(/\+/g, " "));
}
</script>
```

자바소스에서 받아준 방이름과 가격을 자바스크립트에서 사용  
하기 위해 getParameterByName함수를 선언하여 사용

```
if (rsp.success) { /* 결제성공시 */
    var msg = '결제가 완료되었습니다.';
    msg += '고유ID : ' + rsp.imp_uid;
    msg += '상점 거래ID : ' + rsp.merchant_uid;
    msg += '결제 금액 : ' + rsp.paid_amount;
    msg += '카드 승인번호 : ' + rsp.apply_num;
    location.href="MainPage.jsp"; //결제 성공시 메인화면으로 넘어가기
} else { /* 결제실패시 */
    var msg = '결제에 실패하였습니다.';
    msg += '에러내용 : ' + rsp.error_msg;
    location.href="ReservationPage.jsp"; //결제 실패시 다시 결제화면으로 넘어가기
}
alert(msg);
```

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

---

## 마이페이지 (예약조회)

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

#### 마이페이지(예약조회)



```

40 <%
41 // 로그인 세션
42 String id = (String)session.getAttribute("id");
43 %>
44
45 <nav id="myMain">
46   <%if( id == null) {%>
47     <!-- 로그인하지 않은 경우 마이페이지 사용 불가 -> 로그인페이지로 넘겨준다 -->
48     <script type="text/javascript">
49       location.href = "Login.jsp"
50     </script>
51   <%else{%>
52     <!-- 로그인한 경우 -->
53     <nav id="MyTop">
54       <jsp:include page="MyTop.jsp"></jsp:include>
55     </nav>
56   <%>
57   <nav id="MyPageContent">
58     <jsp:include page="MyPageContent.jsp"></jsp:include>
59   </nav>
60 </nav>

```

마이페이지의 경우 메인페이지(MainPage.jsp)에서 버튼을 누른 경우 로그인 여부를 통해 비 로그인시 로그인 화면으로 넘어가고 로그인 시 마이페이지(MyPage.jsp)로 넘어간다.

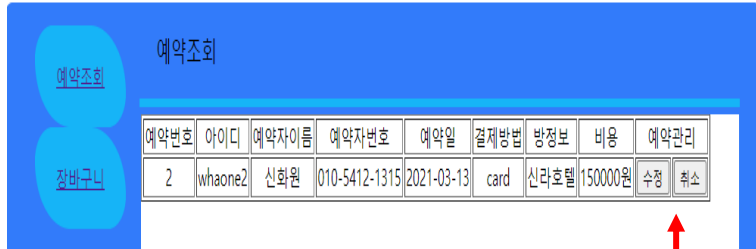
```

15 <!-- 마이페이지 디폴트 화면 -->
16 <div id="MyPageright">
17   <p class = "a">MyPage</p>
18   <hr id = "MyPagehr" color = "#16b4f7" size = 6>
19   <div id = "MyPagecenter"><!-- right라는 영역안에 center라는 영역을 만들기 위해 설정 -->
20     환영합니다<br>
21     저희 사이트를 사용해 주셔서 감사합니다.
22   </div>
23 </div>

```

마이페이지로 넘어오면 디폴트화면을 보여준다.

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원



#### MyPageContentInquiry.jsp

```

<!-- 위에서 디버거를 통해 read해온 예약정보리스트를 불러오기 -->
<!-- 예약정보가 들어있는 리스트 활용 -->
<% for( int i = 0; i<list.size(); i++){
    ReservationVO bag = list.get(i); %>
<!-- 예약정보에 대한 것과 값을 넣을 테이블 생성 -->
<table border="1">
<tr> <!-- tr: table row 한 행에 해당하는 테이블 -->
<td align = "center"><%= bag.getRes_num()%></td>
<td align = "center"><%= bag.getRes_id()%></td>
<td align = "center"><%= bag.getRes_name()%></td>
<td align = "center"><%= bag.getRes_tel()%></td>
<td align = "center"><%= bag.getRes_date()%></td>
<td align = "center"><%= bag.getRes_payment()%></td>
<td align = "center"><%= bag.getRes_room()%></td>
<td align = "center"><%= bag.getRes_cost()%></td>
</tr>
<tr><!-- 예약테이블에 저장된 정보를 출력해준다. -->
<td align = "center"><%= bag.getRes_num()%></td>
<td align = "center"><%= bag.getRes_id()%></td>
<td align = "center"><%= bag.getRes_name()%></td>
<td align = "center"><%= bag.getRes_tel()%></td>
<td align = "center"><%= bag.getRes_date()%></td>
<td align = "center"><%= bag.getRes_payment()%></td>
<td align = "center"><%= bag.getRes_room()%></td>
<td align = "center"><%= bag.getRes_cost()%></td>

```

위에서 받아온 예약정보를 화면에 출력해준다

```

<%
// 로그인 세션
String id = (String)session.getAttribute("id");
ReservationDB db = new ReservationDB();
ArrayList<ReservationVO> list = db.read(id); //id 정보를 주면 그 id에 해당하는 예약정보를 받아온다.
%>

```

자바 부분에서 로그인 된 아이디에 해당하는 예약테이블을 불러온다.

```

public ArrayList<ReservationVO> read(String res_id) throws Exception {
    String sql = "select * from reservation where res_id = ?";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setString(1, res_id);
    ResultSet rs = ps.executeQuery(); //ResultSet => db에서 테이블셋을 뜻함(ResultSet)

    ArrayList<ReservationVO> list = new ArrayList<ReservationVO>(); //예약정보를 담을 리스트 생성
    while(rs.next()) { //rs.next() => 테이블셋의 첫번째~n번째 행에 값이 있으면 true
        // Record pointer 이동 : Record Pointer가 있는 지점 자료만 참조
        // Record pointer의 처음 위치는 0번째이고 next()를 통해 다음으로 이동
        int res_num = rs.getInt(1);
        String res_id1 = rs.getString(2);
        String res_name = rs.getString(3);
        String res_tel = rs.getString(4);
        String res_date = rs.getString(5);
        String res_payment = rs.getString(6);
        String res_room = rs.getString(7);
        int res_cost = rs.getInt(8);

        ReservationVO bag = new ReservationVO();
        bag.setRes_num(res_num);
        bag.setRes_id(res_id1);
        bag.setRes_name(res_name);
        bag.setRes_tel(res_tel);
        bag.setRes_date(res_date);
        bag.setRes_payment(res_payment);
        bag.setRes_room(res_room);
        bag.setRes_cost(res_cost);

        list.add(bag); //VO를 통해 만든 bag을 list에 쌓는다
    }

    return list; //여러개의 레코드를 가져오기 위해 리스트에 담
}

```



### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

예약조회

예약조회

장바구니

예약번호	아이디	예약자이름	예약자번호	예약일	결제방법	방정보	비용	예약관리
2	whaone2	홍길동	010-1234-4321	2021-03-13	card	신라호텔	150000원	수정 취소

#### MyPageContentInquiry.jsp

예약조회

삭제됨

```
<form>
<td><button name = "upd" value = <%= bag.getRes_num()%> formaction="ReservationUpdate.jsp">수정</button></td>
<td><button name = "del" value = <%= bag.getRes_num()%> formaction="ReservationDeleteDB.jsp">취소</button> </td>
</form>
```

MyPageContentInquiry.jsp 에서 버튼을 생성하여 취소버튼을 누를 경우 "del"이라는 이름으로 예약번호를 DB 삭제처리 페이지(ReservationDeleteDB.jsp)로 넘겨준다.

```
<%
//delete : 예약번호를 통해 그에 해당하는 예약지를것
//입력받은 예약번호는 문자열이기 때문에 정수형으로 변경해야함

String res_num1 = request.getParameter("del");
int res_num = Integer.parseInt(res_num1); //정수형으로 변환
System.out.println(res_num);
ReservationDB db = new ReservationDB();
db.delete(res_num);

response.sendRedirect("MyPageContentInquiry.jsp"); //delete를 수행하고 예약조회 페이지를 다시보여주기 위해 사용
%>
```

ReservationDeleteDB.jsp에서 넘겨받은 예약번호(res\_num)을 통해 예약테이블에서 해당하는 부분을 삭제해준다.  
DB 처리 후 예약조회 페이지로 화면을 넘겨준다.

```
public void delete(int res_num, throws Exception {
//예약번호를 통해서 삭제
String sql = "delete from reservation where res_num = ?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setInt(1, res_num);

ps.executeUpdate(); //return = void
con.close(); //커넥션 해제
}
```

예약번호를 통해 삭제 쿼리문 실행

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

예약조회

예약번호	아이디	예약자이름	예약자번호	예약일	결제방법	방정보	비용	예약관리
2	whaone2	신화원	010-5412-1315	2021-03-13	card	신라호텔	150000원	수정 취소

#### MyPageContentInquiry.jsp

예약정보수정

변경할 예약자이름:

변경할 예약자번호:

수정하기

변경할 예약자이름:

변경할 예약자번호:

수정하기

```
<form>
<td><button name = "upd" value = <%= bag.getRes_num()%> formaction="ReservationUpdate.jsp">수정</button></td>
<td><button name = "del" value = <%= bag.getRes_num()%> formaction="ReservationDeleteDB.jsp">취소</button> </td>
</form>
```

취소와 같은 방식으로 버튼을 누를 시 "upd"라는 이름으로 예약번호(res\_num)을 넘겨준다.

```
<p class="a">예약정보수정</p>
<hr id="MyPagehr" color="#16b4f7" size=6>
<div id="MyPagecenter">
<form action="ReservationUpdateDB.jsp" id = "upform"><!-- 업데이트 기능이 있는 페이지로 아래 값들을 넘겨준다 -->
<input type="hidden" name = upd1 value = <%=res_num%>>
변경할 예약자이름: <input name = "upd_name"><br>
변경할 예약자번호: <input name = "upd_tel"><br>
<button>수정하기</button>
</form>
```

ReservationUpdate.jsp 페이지로 넘어와 예약번호(res\_num), 변경할 이름(upd\_name) 변경할 번호(upd\_tel)을 받아서 업데이트 DB 처리 페이지로 넘겨준다.

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

```

<%
/* 예약정보 수정하기 */
/* 예약자이름 및 전화번호 수정하기 */
/* 변경사항 작성하는 페이지(reservationUpdate.jsp)에서 무엇을 변경할지 결정하기 위해 예약번호(res
String res_num1 = request.getParameter("upd1"); //입력은 문자열
int res_num = Integer.parseInt(res_num1); //정수값으로 변경
String res_name = request.getParameter("upd_name");
String res_tel = request.getParameter("upd_tel");

ReservationDB db = new ReservationDB();
ReservationVO bag = new ReservationVO();
bag.setRes_num(res_num);
bag.setRes_name(res_name);
bag.setRes_tel(res_tel);
db.update(bag);

response.sendRedirect("MyPageContentInquiry.jsp"); //예약조회페이지로 넘어감
%>

public void update(ReservationVO bag) throws Exception {
String sql = "update reservation set res_name = ?, res_tel = ? where res_num = ?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, bag.getRes_name());
ps.setString(2, bag.getRes_tel());
ps.setInt(3, bag.getRes_num());

ps.executeUpdate();
con.close();
}

```

ReservationUpdateDB.jsp에서 수정할 상품에 해당하는 예약번호(res\_num)와 수정을 위해 입력 받은 이름, 번호를 VO에 담아 처리 시켜준다.

수정이 끝난 후 예약조회페이지로 화면을 넘겨준다

업데이트 메소드

예약번호	아이디	예약자이름	예약자번호	예약일	결제방법	방정보	비용	예약관리
2	whaone2	신화원	010-5412-1313	2021-03-13	card	신라호텔	150000원	수정 취소

예약번호	아이디	예약자이름	예약자번호	예약일	결제방법	방정보	비용	예약관리
2	whaone2	홍길동	010-1234-4321	2021-03-13	card	신라호텔	150000원	수정 취소

예약정보가 바뀐 걸 확인 할 수 있다

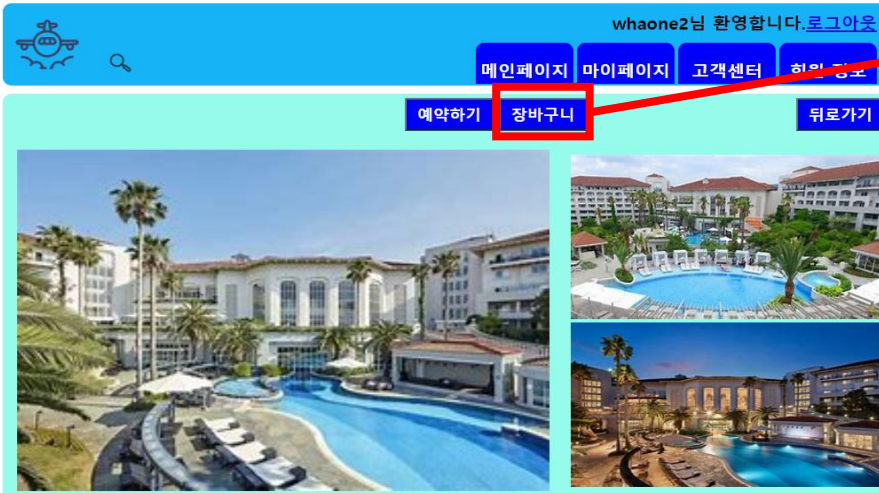
### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

---

## 마이페이지 (장바구니)

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

#### 마이페이지(장바구니)



```
<a href="Shoppingbag.jsp?name=<%=bag.getProductNum()%>&page=<%=bag.getProductNum()%>">
<button class="button">장바구니</button>
```

오른쪽 사진(ProductDetail.jsp)에서 장바구니 버튼을 누르면 상품 이름과 상품번호(주소결정)을 장바구니 입력처리페이지(Shoppingbag.jsp)로 넘겨준다.

```
<%
/* 상품페이지에서 장바구니담기 버튼을 누를시 shoppingbag.jsp파일로 상품주소 및 상품이름을 파라미터값으로 받아온다. */
// 로그인 세션
String id = (String)session.getAttribute("id");
String product = request.getParameter("page"); //장바구니에 담을 상품주소에 해당하는 productNum값을 받아오기
String product1 = "ProductDetail.jsp?productNum="+product; //product1에 상품페이지 주소를 넣어준다
String product_name = request.getParameter("name");//장바구니에 담을 상품이름 받아오기

ShoppingBagDB db = new ShoppingBagDB();
ShoppingBagVO bag = new ShoppingBagVO();

bag.setId(id);
bag.setProduct(product1);
bag.setProduct_name(product_name);

/* 로그인된 상태에서 같은아이디가 같은 상품을 계속해서 담는것(중복)을 막기위한 설정 */
//사용자 아이디와 상품이름이 중복되는건 입력하지x
if(db.check(id, product_name) == false ){
    db.create(bag); //장바구니 정보를 db에 담기
}

response.sendRedirect(product1);//장바구니 담기 버튼을 누른 그 페이지(현재있는페이지로)로 다시 돌아간다
%>
```

앞선 페이지에서 받아온 상품번호를 통해 장바구니에 담을 상품 주소를 만들고 이름을 받아온다.

사용자당 하나의 상품에 대해 한 번만 장바구니에 담을 수 있도록 조건문을 통해 DB 처리하도록 설정했다.

장바구니테이블에 입력이 완료되면 상품상세페이지로 화면을 전환해 준다.

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

```
// 로그인된 아이디 사용자가 장바구니에 담은 제품이 있는지 파악을 위해 boolean형태의 메소드 생성 -> 만약 있으면 true로 반환 없으면 false로 반환
public boolean check(String id, String product_name) throws Exception {
    String sql = "select * from shoppingbag where id = ? and product_name = ?";
    PreparedStatement ps = con.prepareStatement(sql); // sql 구문 작성
    ps.setString(1, id);
    ps.setString(2, product_name);
    ResultSet rs = ps.executeQuery(); // 4.mysql로 생성한 sql문 보냄 -> 쿼리문 처리
    boolean result = false;
    if(rs.next()) { // 있는지 없는지 파악해주는 것 -> 있으면 들어가고 없으면 안들어감
        result = true;
    }
    return result;
}
```

장바구니에 넣기 전 중복유무 처리 해주는 메소드를 통해 불린 값을 출력해줘서 판단 해준다.

#### 중복체크를 위한 메소드

```
// 장바구니 버튼을 눌렀을 때 해당 아이디와 상품정보를 테이블에 넣어주는 쿼리문
public void create(ShoppingBagVO bag) throws Exception {
    String sql = "insert into shoppingbag values (null,?, ?, ?)";
    PreparedStatement ps = con.prepareStatement(sql); // sql 구문 작성
    ps.setString(1, bag.getId());
    ps.setString(2, bag.getProduct());
    ps.setString(3, bag.getProduct_name());
    ps.executeUpdate(); // 4.mysql로 생성한 sql문 보냄 -> 쿼리문 처리

    con.close(); // 커넥션 해제
}
```

위의 중복유무 처리가 완료되면 입력 메소드를 통해 장바구니 테이블에 넣어준다.

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

장바구니

장바구니삭제

삭제됨

```

/* 장바구니에 담은 상품을 보여주기 위해 db처리해주는 코드 */
ShoppingBagDB db = new ShoppingBagDB();
ArrayList<ShoppingBagVO> list = db.read(id);
%>

<td>관심상품</td>
<!-- 장바구니 내에 상품이름을 클릭하면 상품페이지로 이동하기 위해 a태그 활용 -->
<td><a href=<%=bag.getProduct()%><%=bag.getProduct_name()%></a></td>
<!-- Line 146~147 삭제버튼을 눌렀을 때 삭제기능저리가 있는 페이지로 넘어간다 -->
<input type = "hidden" name = "id" value =<%=bag.getId()%>
<input type = "hidden" name = "product" value =<%=bag.getProduct()%>
<td><button formaction="ShoppingbagDelete.jsp">장바구니삭제</button></td>

<%
/* MyPageContentBasket.jsp에서 삭제를 위해 필요한 아이디와 상품관련페이지를 받아온다*/
String id = request.getParameter("id");
String product = request.getParameter("product");

/* db를 생성하여 삭제기능 */
ShoppingBagDB db = new ShoppingBagDB();
db.delete(id, product);

/* 삭제 후 삭제버튼을 눌렀던 페이지인 장바구니페이지로 다시 돌아간다 */
response.sendRedirect("MyPageContentBasket.jsp");
%>

```

마이페이지에 있는 장바구니 페이지를 확인하면 장바구니에 담은 상품을 확인 할 수 있다.

상품이름을 클릭하면 상품페이지로 넘어간다.

장바구니삭제 버튼을 누르면 ShoppingbagDelete.jsp 페이지로 아이디와 상품페이지를 받아와 삭제해준다.

```

//장바구니 삭제
public void delete(String id , String product_name) throws Exception {
    String sql = "delete from shoppingbag where id = ? and product = ?";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setString(1, id);
    ps.setString(2, product_name);
    ps.executeUpdate();
    con.close();
}

```

장바구니 삭제 메소드



### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

---

## 최근 본 상품

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원

#### 최근 본 상품



```
<a class="cellBlock" href="Arraylist_basket.jsp?productNum=<%=bag2.getProductNum() %>
&productName=<%=bag2.getProductName() %>">
```

사진을 클릭하면 바로 상품상세페이지가 아닌 방문결과를 저장하기 위해 만들어 둔 Arraylist\_basket.jsp 페이지로 상품번호(productNum)과 상품명(productName)을 넘겨준다.

```
<!-- 메인페이지 오른쪽 최근방문기록을 위해 방문기록을 ArrayList에 담아 세션으로 설정 -->
<!-- 방문한 곳 페이지 주소 받아오기 -->
<!-- 상품페이지에서 예매버튼이 있는 상세한 상품페이지로 넘어갈 때 이 페이지(arraylist_basket.jsp)로 상품페이지의 주소와 상품명들을 파라미터로 보내준다 -->
<%
String productNum = request.getParameter("productNum"); //방문한 상품 상세페이지 주소를 갖고있음 ex) A호텔에 대한 페이지주소
String real_productpage = "ProductDetail.jsp?productNum="+productNum; //상품넘버를 통해 상품페이지 생성
session.setAttribute("productsession", real_productpage); //productsession : 상품페이지 주소(세션)

/* 조건문을 통해 상품의 상세페이지를 담은 ArrayList가 비어있는 경우 배열리스트를 생성해서 값을 담는다. */
if(session.getAttribute("productlist")==null){ //
    ArrayList<String> productlist = new ArrayList<>();
    productlist.add((String)session.getAttribute("productsession"));
    session.setAttribute("productlist", productlist); //productlist : 배열리스트로 상품페이지 주소가 들어있음
}else{ /* 상품의 상세페이지를 담은 배열리스트에 값이 들어 있는 경우 덧붙여 사용한다. */
    ArrayList<String> productlist1 = (ArrayList)session.getAttribute("productlist");
    productlist1.add((String)session.getAttribute("productsession"));
    session.setAttribute("productlist", productlist1); //productlist : 배열리스트로 상품페이지 주소가 들어있음
}
System.out.println(session.getAttribute("productlist"));
response.sendRedirect(real_productpage+"&productNum="+productNum); //현재 방문하기 위해 눌렀던 상품페이지로 다시 돌아감
%>
```

이전 페이지에서 받아온 상품번호를 통해 상품주소를 생성하여 real\_productpage에 담아준다.

그 후 ArrayList를 생성하여 배열리스트가 빈 경우, 차있는 경우를 나눠 주소를 쌓아준다.

그리고 나서 방문주소가 담긴 리스트를 세션 설정한다.

모든 설정이 끝나면 위에서 사진을 눌렀을 때, 넘어가는 상품상세페이지로 이동시켜준다.

### III. 메뉴별 소개 : 결제, 마이페이지, 최근 본 상품 - 신화원



상품 주소 세션을 보낸 페이지에서 현재 상품상세페이지 (ProductDetail)로 전환되고 세션으로 처리된 주소 리스트를 받아서 최근 본 상품 부분에 찍어 준다.

이 부분은 하이퍼링크 처리가 되어 누르면 최근 본 상품 페이지로 화면이 전환된다.

```
<% if (session.getAttribute("productlist")!=null){
    /* arraylist_basket.jsp에서 세션으로 설정한 productlist를 담은 새로운 리스트를 생성 */
    ArrayList<String> productlist1 = (ArrayList)session.getAttribute("productlist"); /* object -> arraylist */
    if(productlist1.size()<5){ /* 최근방문페이지가 5개 보다 작은경우 */
        switch(productlist1.size()) { /* switch-case문을 통해 방문페이지가 1개 2개 3개 4개 인경우 나눈 */
            case 1: %>
                <tr>
                    <th>최근 본 상품</th>
                </tr>
                <tr><!-- 방문한페이지주소 (링크)+ 이름으로 표현-->
                    <td><a href=<%=productlist1.get(productlist1.size()-1) %>>1. 최근본상품</a></td>
                </tr>
                <tr>
                    <td>호텔이름</td>
                </tr>
                <tr>
                    <td>호텔이름</td>
                </tr>
                <tr>
                    <td>호텔이름</td>
                </tr>
            </tr>
        }
    }
}
```

위 사진처럼 찍기 위해서 최근 본 상품페이지 리스트를 받아와 조건처리 하여 출력하도록 설정했다.

1. 페이지리스트 < 5

- switch~case문을 통해 확인

2. 페이지리스트 >=5

- 가장 최근 상품페이지를 맨 상단에 찍어야 하기 위해 마지막 인덱스 부터 5개만 사용해서 찍는다.  
ex) list.size()-1 ~ list.size()-5 까지 출력

최근 본 상품은 메인페이지와 상품페이지에서  
확인가능 + 세션처리로 인해 창