



jdbc를 활용하여 client와 server, database를 연결하는 숙박시설 예약 사이트 구축

제주도 숙박예약홈페이지 제작

공지사항 및 고객센터
이동형

CONTENTS 제주도 숙박예약홈페이지 제작

I. 개요

II. 시스템 구성도

III. 메뉴별 소개

- 1) 데이터베이스 설계
- 2) 메뉴별 순서도
- 3) 구현

IV. 소감문



I. 개요

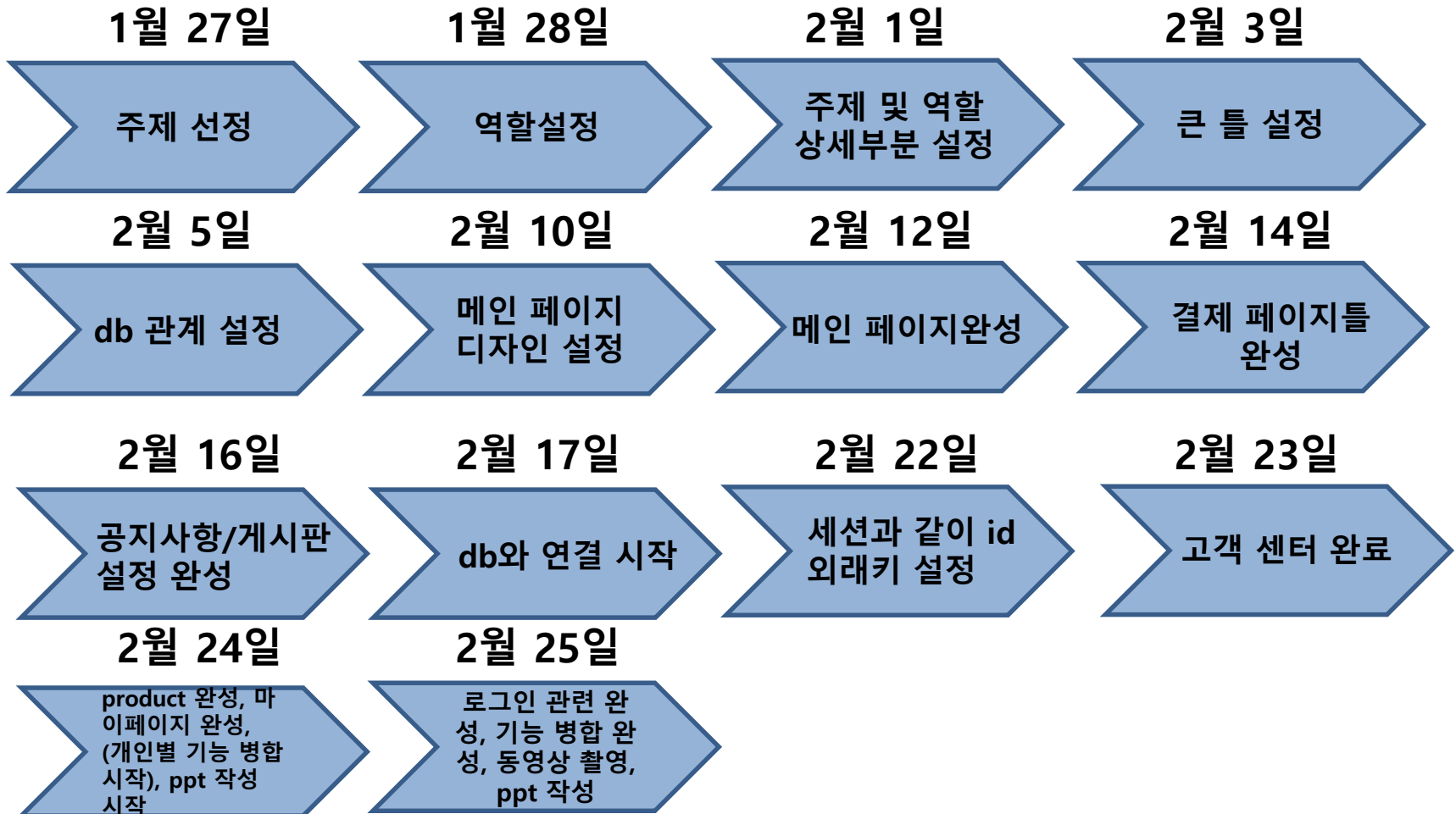
프로젝트명

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 : customer_sound(고객의 소리)

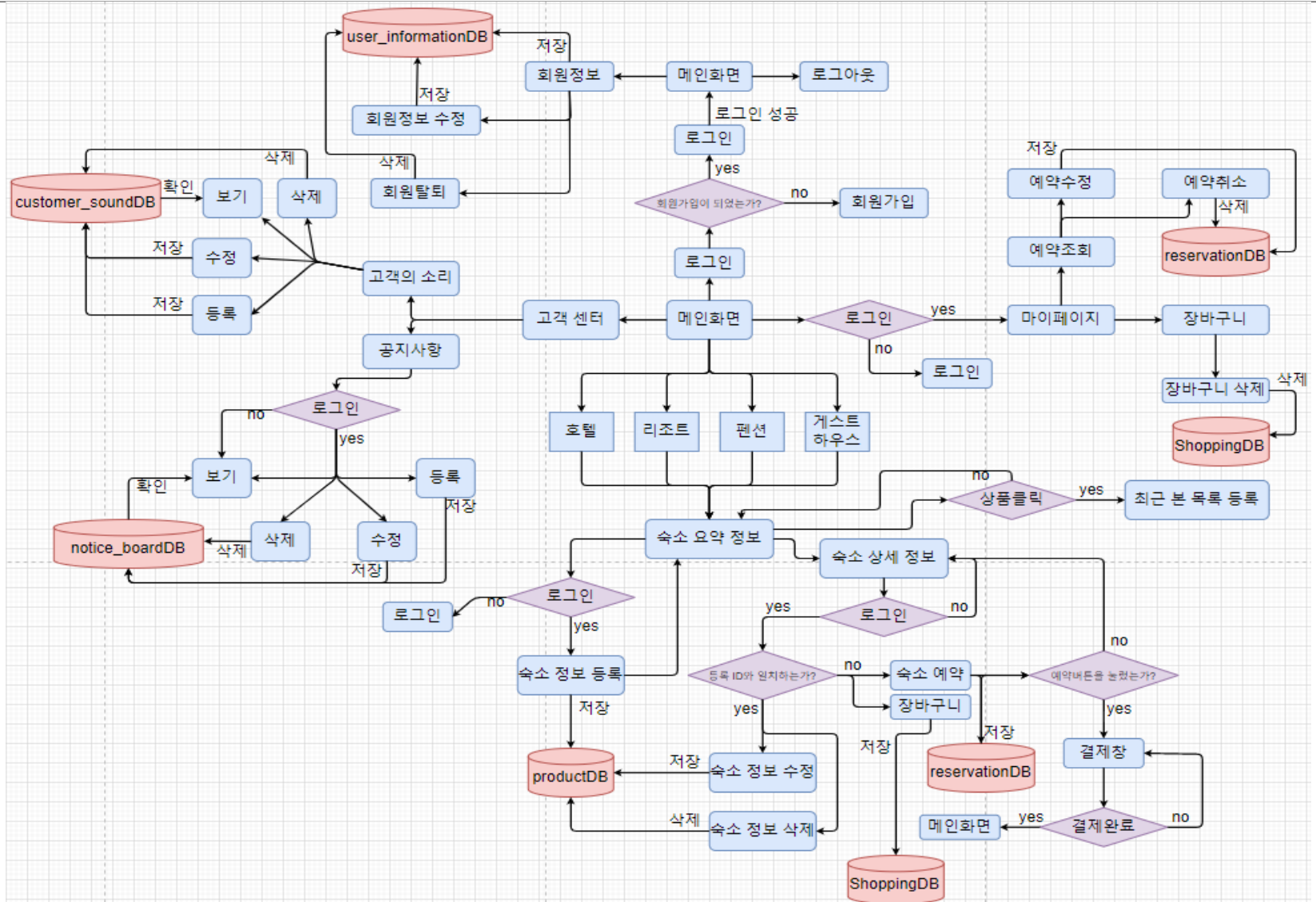
컬럼 명	Data Type	제약 조건	설명
Customer_no	Int(11)	Not Null Primary Key	등록 번호
Customer_type	Varchar(100)	Not Null	좋아요/나빠요
Customer_title	Varchar(100)	Not Null	제목
Customer_acoomodation_name	Varchar(100)	Not Null	숙소 이름
Customer_date	Varchar(100)	Not Null	등록 날짜
Customer_writer	Varchar(100)	Not Null Foreign Key	작성자
Customer_content	Varchar(100)	Not Null	내용
Customer_grade	Int(11)	Not Null	평점

I. 개요

개발 과정 >> 제약조건 : notice_board(공지사항)

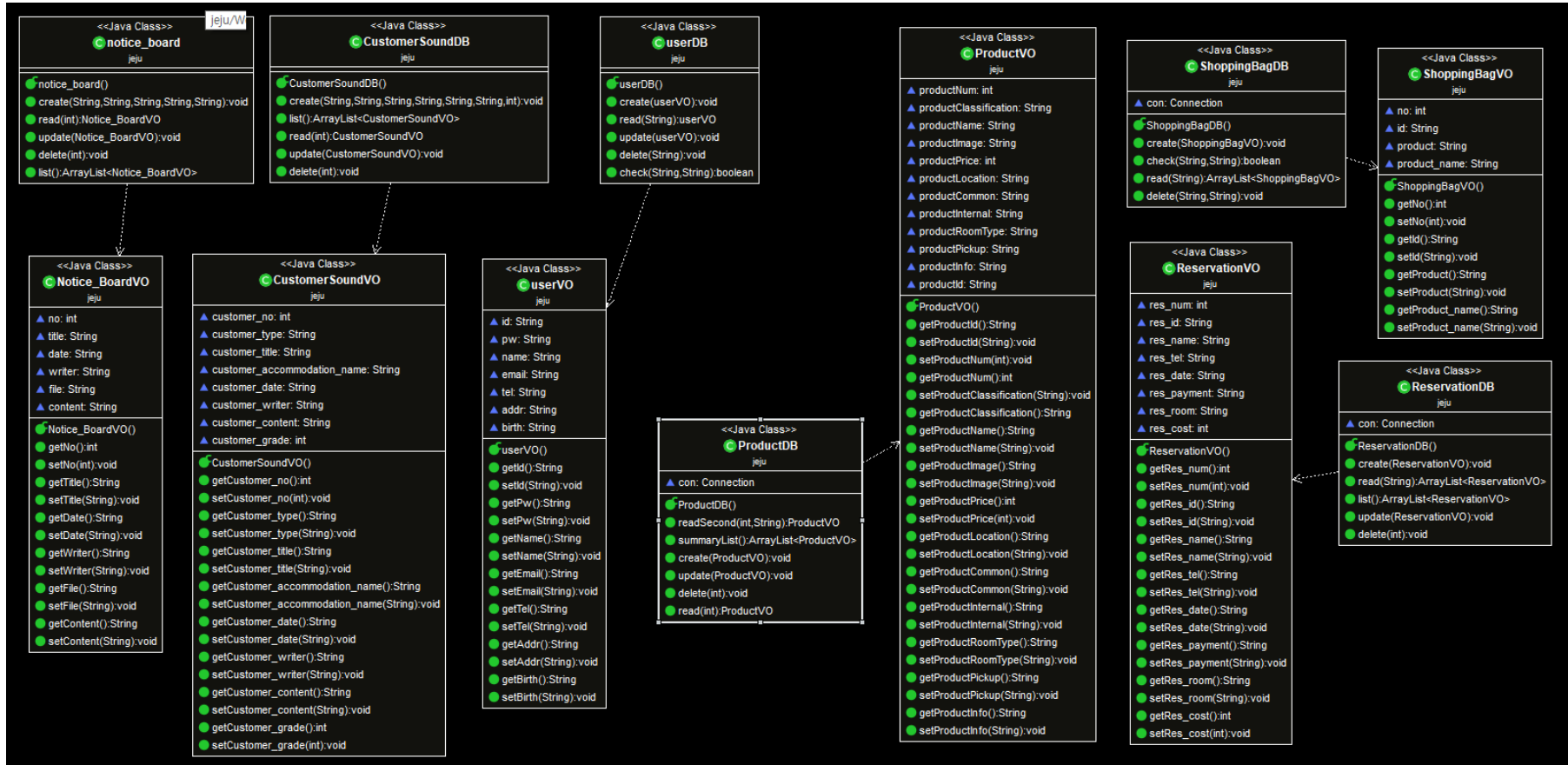
컬럼 명	Data Type	제약 조건	설명
Notice_no	Int(11)	Not Null Primary Key	등록번호
Notice_title	Varchar(100)	Not Null	제목
Notice_date	Varchar(100)	Not Null	등록 날짜
Notice_writer	Varchar(100)	Not Null Foreign Key	작성자
Notice_file	Varchar(100)	Not Null	첨부파일
Notice_content	Varchar(100)	Not Null	내용

I. 개요-전체메뉴 순서도



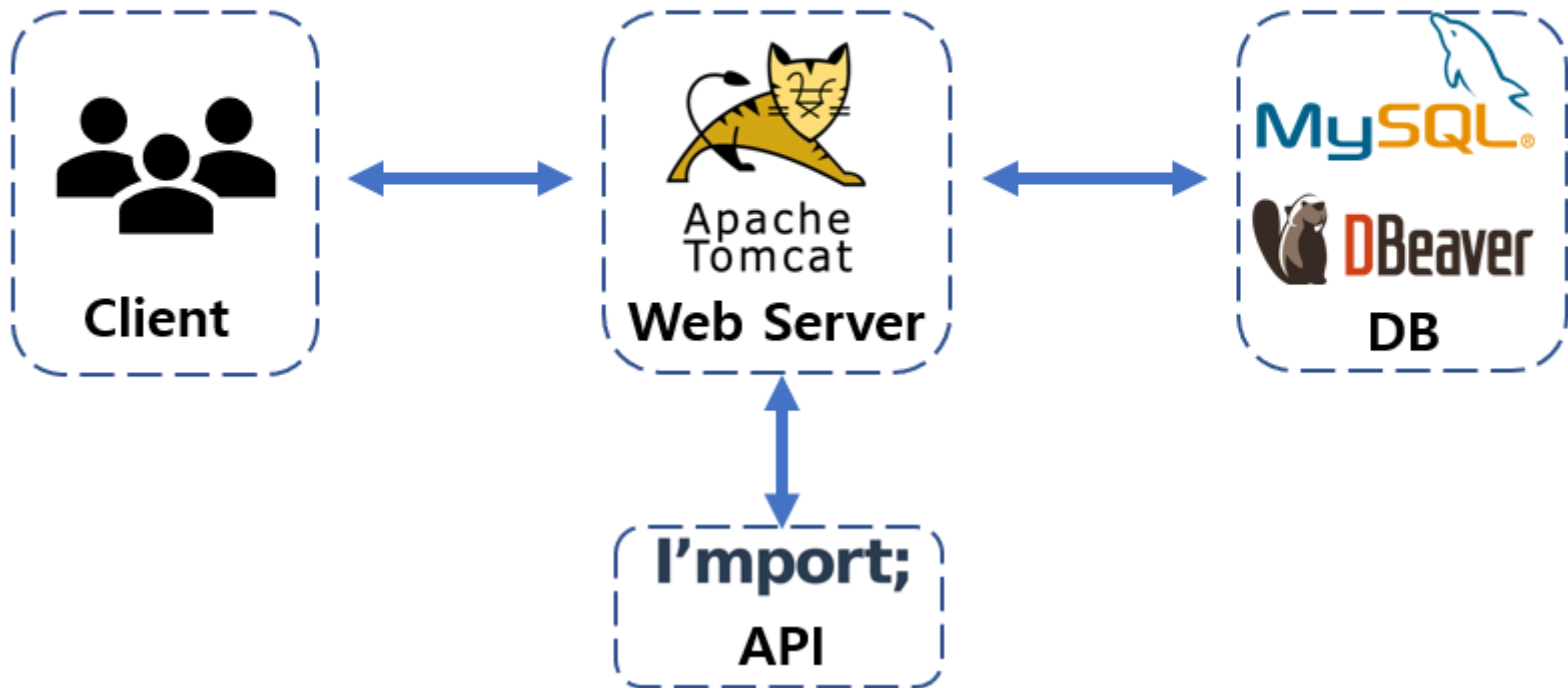
III. 메뉴별 소개 : 전체메뉴

전체 UML

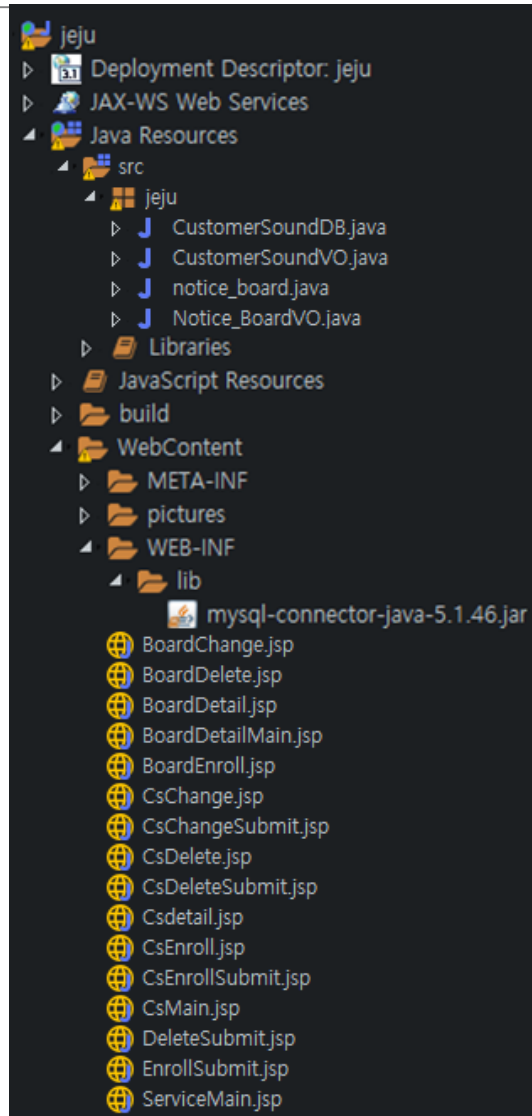


II. 시스템 구성도

 **eclipse**
Development Tool



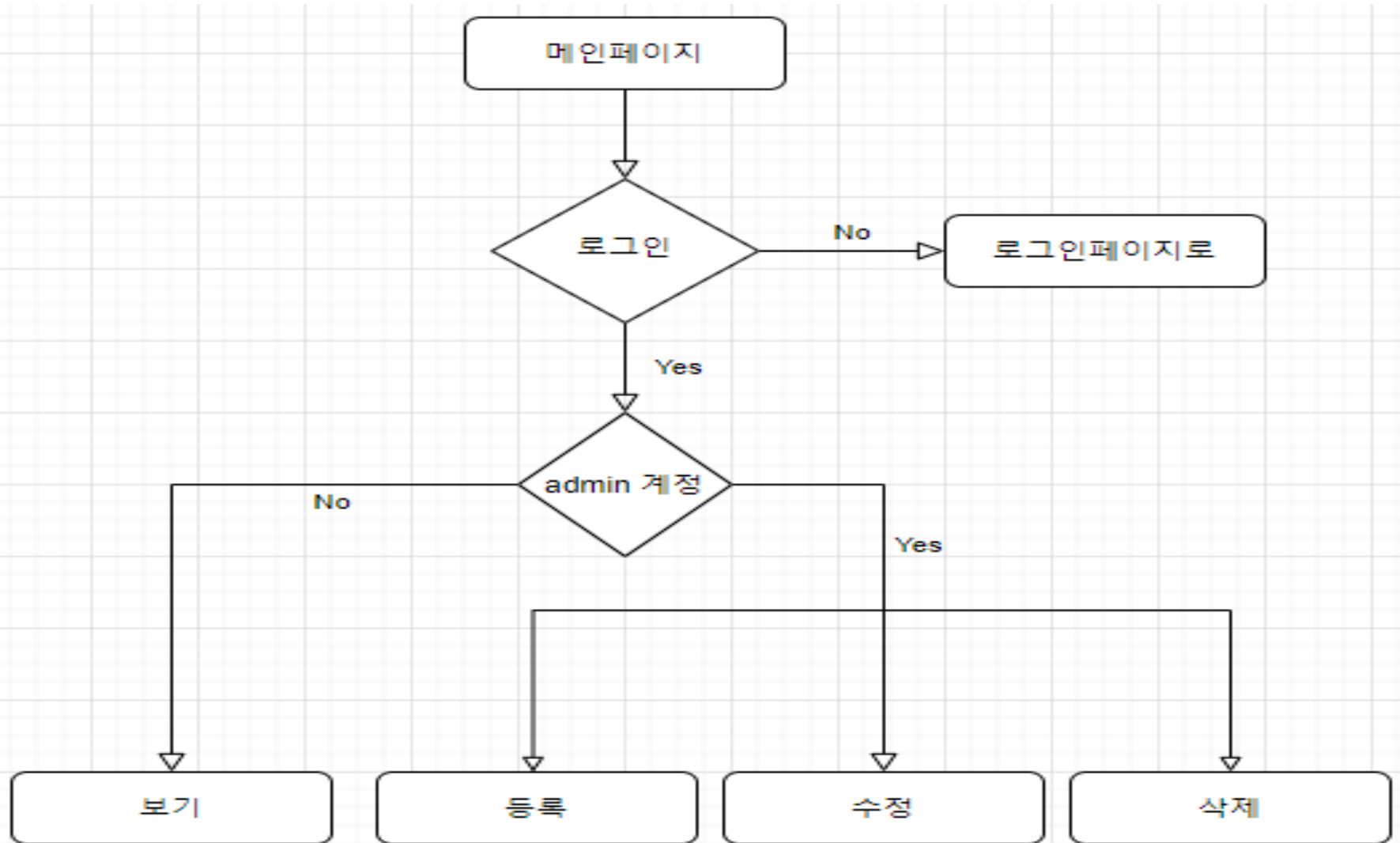
III. 메뉴별 소개 : 고객센터 공지사항 프로젝트 트리



III. 메뉴별 소개 : 고객센터 공지사항 UML



III. 메뉴별 소개 : 고객센터 공지사항 설계도



III. 메뉴별 소개 : 게시판(연결)

```
44      로깅을 해주세요.
45      <% } else { %> <!-- 로그인에 성공했을 경우 아래 구문 수행 -->
46      <%= id %>님 환영합니다.<a href="Logout.jsp">로그아웃</a> <!-- 로그인된 사용자의 아이디를 출력 -->
47      <% } %>
48      </th><!-- 테이블 헤더 완성 -->
49      </tr><!-- 테이블의 한줄을 완성 -->
50      <tr>
51      <td><!-- 테이블 데이터 셀을 생성 -->
52      <a class="menuLink" href="MainPage.jsp">메인페이지</a> <!-- 세션을 이용해 메인페이지 셀을 눌렀을때 로그인된 아이디 값을 넘겨주면서 메인페이지로 연결되는 데이터를 생성 -->
53      </td><!-- 테이블 데이터 완성 -->
54      <td>
55      <a class="menuLink" href="MyPage.jsp">마이페이지</a>
56      </td>
57      <td>
58      <a class="menuLink" href="ServiceMain.jsp">고객센터</a>
59      </td>
60      <td>
61      <% if( id == null) { %> <!-- 세션을 이용해 로그인이 되지 않았을때 로그인을 하기위해서 로그인페이지로 연결되는 데이터를 생성 -->
62      <a class="menuLink" href="Login.jsp">로그인</a>
63      <% } else { %> <!-- 로그인에 성공 했을경우 로그아웃을 할 수 있게하는 로그아웃으로 연결되는 셀을 생성 -->
64      <a class="menuLink" href="UserUpdate.jsp">회원 정보</a>
65      <% } %>
66      </td>
67      </tr>
68      </table> <!-- 테이블 완성 -->
69      </nav><!-- menu 레이아웃 완료 -->
70      </header> <!-- 헤더 레이아웃 완료 -->
71
72
73
74      <nav id="FixedBasket"><!-- 오른쪽에 고정될 장바구니 디스플레이를 만들기위해서 FixedBasket 아이디를 가진 nav 레이아웃 생성 -->
75      <table id="BasketContent" style="width:100%" > <!-- 장바구니는 아래로 꼭 나와야 하기 때문에 하나의 열을 가진 테이블을 생성 -->
76      <tr> <!-- 각 테이블의 행을 생성하는 태그 -->
77      <th>장바구니</th> <!-- 테이블의 헤더를 생성 -->
78      </tr>
79      <tr>
80      <td>호텔이름</td> <!-- 테이블의 데이터를 생성 -->
81      </tr>
82      <tr>
83      <td>호텔이름</td>
84      </tr>
85      <tr>
86      <td>호텔이름</td>
87      </tr>
```

MainPage.jsp 에서 고객센터 클릭시 ServiceMain.jsp로 연결

III. 메뉴별 소개 : 게시판(공지사항_등록)



MainPage.jsp 에서 공지사항 클릭시 ServiceMain.jsp로 연결

III. 메뉴별 소개 : 게시판(공지사항_등록)



로그인을 해주세요.

[메인페이지](#)[마이페이지](#)[고객센터](#)[로그인](#)

[공지사항](#)[고객소리](#)

공지사항

번호	제목	작성일	작성자	첨부파일	비고
5	부대시설 및 식사 이용 결제	2021.02.24	admin	없음	
4	비회원 예약 가능 여부	2021.02.24	admin	없음	
3	잔액지변으로 예약취소 규정	2021.02.24	admin	없음	

로그인하기

```
if (session.getAttribute("id") == null) {
%>
<!-- 세션이 잡혀 있지 않으면 로그인하기! -->
<form action="Login.jsp">
  <button>로그인하기</button>
</form>
<%
  } else if (session.getAttribute("id").equals("admin")) {
%>
<!-- admin 관리자 계정으로 로그인시 등록하기 활성화 -->
<form action="BoardEnroll.jsp">
  <button>등록하기</button>
</form>
<%
  } else {
%>
<!-- 관리자와 다른 세션 설정시 등록권한이 없습니다! -->
등록권한이 없습니다.
<%
  }
%>

<%if (id == null || !(id.equals("admin"))){ %>
<!-- 로그인한 id에 따라서 버튼 활성화 시키기 admin만 삭제 가능-->
<%} else if(id.equals("admin")) {%>
<td><button onclick="location.href='BoardDelete.jsp'">삭제</button></td>
<%>
```

세션이 잡혀있지 않으면 로그인하기 활성화
세션이 admin id로 잡혀 있으면 등록하기 및 삭제 활성화
세션이 admin외 다른 경로 잡혀 있으면 등록권한이 없습니다.

III. 메뉴별 소개 : 게시판(공지사항_등록)

로그인을 해주세요.

메인페이지 마이페이지 고객센터 로그인

공지사항

공지사항

고객소리

번호	제목	작성일	작성자	첨부파일	비고
5	부대시설 및 식사 이용 결제	2021.02.24	admin	없음	
4	비회원 예약 가능 여부	2021.02.24	admin	없음	
3	전세전화번호로 예약취소 규정	2021.02.24	admin	없음	

로그인하기

admin님 환영합니다.로그아웃

메인페이지 마이페이지 고객센터 회원 정보

공지사항

공지사항

고객소리

번호	제목	작성일	작성자	첨부파일	비고
5	부대시설 및 식사 이용 결제	2021.02.24	admin	없음	삭제
4	비회원 예약 가능 여부	2021.02.24	admin	없음	삭제
3	전세전화번호로 예약취소 규정	2021.02.24	admin	없음	삭제

등록하기

lee님 환영합니다.로그아웃

메인페이지 마이페이지 고객센터 회원 정보

공지사항

공지사항

고객소리

번호	제목	작성일	작성자	첨부파일	비고
5	부대시설 및 식사 이용 결제	2021.02.24	admin	없음	
4	비회원 예약 가능 여부	2021.02.24	admin	없음	
3	전세전화번호로 예약취소 규정	2021.02.24	admin	없음	

등록권한이 없습니다.

각 세션 id 값에 맞는 기능만 활성화

공지사항 등록 및 삭제 기능은 admin 세션일 경우만 활성화

III. 메뉴별 소개 : 게시판(공지사항_등록)

The image shows two screenshots of a web application interface. The left screenshot displays the '공지사항' (Notice) menu with a list of notices. The right screenshot shows the '공지사항 등록 페이지' (Notice Registration Page) with a form for registering a new notice. A green arrow points from the '등록하기' (Register) button in the notice list to the registration form.

공지사항

번호	제목	작성일	작성자	첨부파일	비고
5	부대시설 및 식사 이용 결제	2021.02.24	admin	없음	삭제
4	비회원 예약 가능 여부	2021.02.24	admin	없음	삭제
3	원래지번으로 예약취소 규정	2021.02.24	admin	없음	삭제


[공지사항](#)
[고객소리](#)
[등록하기](#)

공지사항 등록 페이지

제목 :
날짜 :
작성자 :
첨부파일 :
내용 :
[등록하기](#)

공지사항 등록하기 버튼 클릭 -> 공지사항 등록 페이지(BoardEnroll.jsp)로 이동

III. 메뉴별 소개 : 게시판(공지사항_등록 create 메서드)

admin님 환영합니다.로그아웃

메인페이지마이페이지고객센터회원 정보

공지사항 등록 페이지

제목 :

날짜 :

작성자 :

첨부파일 :

내용 :

최소수수료는 할인쿠폰 적용 전 객실 판매금액 기준으로 계산되며, 실 결제금액에서 최소수수료를 차감한 금액으로 환불 처리됩니다.

만약, 결제 시 포인트를 사용한 경우 포인트 -> 현금 순으로 차감됩니다.

환불 예정금액은 조회 시점 및 실제 환불 진행 시점에 따라 달라질 수 있으므로 예약/결제 진행 당시 안내된 취소/환불 규정을 꼭 숙지하여 이용해 주시길 바랍니다.

등록하기

```
<%  
//BoardEnroll.jsp에서 입력값 받아오기!  
  
String title = request.getParameter("title");  
String date = request.getParameter("date");  
String writer = request.getParameter("writer");  
String file = request.getParameter("file");  
String content = request.getParameter("content");  
  
String id = (String)session.getAttribute("id");  
  
notice_board nb = new notice_board(); // create 메서드 사용을 위한 new(틀) 만들기!  
nb.create(title, date, writer, file, content); // 메서드 사용!  
%>
```

입력된 값을 EnrollSubmit.jsp 에서 받아주고 notice_board의 create 메서드를 사용하여 Database로 입력된 값을 저장해 준다.

III. 메뉴별 소개 : 게시판(공지사항_등록 create 메서드)

```
public class notice_board {  
    public void create(String title, String date, String writer, String file, String content)  
  
    // 3. SQL문으로 만들기!  
    String sql = "insert into notice_board values (null, ?, ?, ?, ?, ?)";  
    // 변수 sql에 SQL문 중 create에 해당하는 insert문을 넣어준다. : table명을 지정!  
    // 입력하지 않아도 되는 값이 있을때는 해당 부분에 null이라고 넣어주거나  
    //  
    PreparedStatement ps = con.prepareStatement(sql);  
    // 변수 con으로 메서드 prepareStatement를 사용해서 변수 sql에 있는 문자열을 SQL문으로 변경!  
    ps.setString(1, title); // sql 변수에 입력되는 값을 각가 타입에 맞게 set메서드로 변경  
    ps.setString(2, date);  
    ps.setString(3, writer);  
    ps.setString(4, file);  
    ps.setString(5, content);  
    System.out.println("SQL문으로 변경완료!"); // SQL문으로 변경위의 내용이 잘 처리되는지 확인  
  
    // 4. SQL문을 mySQL 서버로 전송!  
    ps.executeUpdate(); // ps의 메서드인 executeUpdate로 mySQL로 전송!  
    System.out.println("전송완료!"); // 전송과정 문제 확인 출력!
```

Create 메서드는 각 입력값을 sql문으로 변경하여 MySQL database로 저장시켜준다.

III. 메뉴별 소개 : 게시판(공지사항 list 메서드)

admin님 환영합니다.로그아웃

메인페이지 마이페이지 고객센터 회원 정보

공지사항

공지사항

고객소리

번호	제목	작성일	작성자	첨부파일	비고
6	취소수수료 기준	2021.02.24	admin	없음	<button>삭제</button>
5	부대시설 및 식사 이용 결제	2021.02.24	admin	없음	<button>삭제</button>
4	비회원 예약 가능 여부	2021.02.24	admin	없음	<button>삭제</button>
3	천재지변으로 예약취소 규정	2021.02.24	admin	없음	<button>삭제</button>

등록하기

```
<%
for (int i = 0; i < list.size(); i++) { //최근등록사항이 상위노출(SQL문 order by 활동)
//가방에 넣기
Notice_BoardVO bag = list.get(i);
%>

<tr> <!-- 브라우져 노출 부분 -->
<td><%=bag.getNo() %></td>
<td><a href="BoardDetailMain.jsp?notice_no=<%=bag.getNo()%>"><%=bag.getTitle() %></a></td>
<td><%=bag.getDate()%></td>
<td><%=bag.getWriter() %></td>
<td><%=bag.getFile() %></td>
<%if (id == null || !id.equals("admin")) { %>
<!-- 로그인한 id에 따라서 버튼 활성화 시키기 admin만 삭제 가능-->
<%} else if(id.equals("admin")) {%>
<td><button onclick="location.href='BoardDelete.jsp'">삭제</button></td>
<%} %>

</tr>

<%} %>
```

List 메서드를 사용하여 database row 값들을 가져와서
ServiceMain.jsp 에 각 row 의 data들을 노출한다.


III. 메뉴별 소개 : 게시판(공지사항 list 메서드)



```
public ArrayList<Notice_BoardVO> list() throws Exception {  
    ArrayList<Notice_BoardVO> list = new ArrayList<>();  
  
    //3. SQL문으로 가져오기  
    String sql = "select notice_no, notice_title, notice_date, notice_writer, notice_file "  
        + "from notice board order by notice no desc";  
  
    //4. 가져온 SQL문을 실행  
    PreparedStatement ps = con.prepareStatement(sql); // SQL문으로 변경!  
    System.out.println("SQL문 만들기 성공");  
  
    //4. 서버로 보내기!  
    ResultSet rs = ps.executeQuery(); //DB에 있는 테이블 쿼리 값을 rs에 넣는다!  
    System.out.println("SQL문 서버로 전송 성공!");  
  
    //검색결과가 잘 있는지 확인!  
    // System.out.println(rs.next()); // 값이 들어있는지 확인 있으면 true 없으면 false  
  
    while (rs.next()) {  
        //while반복문으로 테이블 row값만큼 값을 저장!  
        //가져올 값을 먼저 변수로 설정  
        int no = rs.getInt(1);  
        String title = rs.getString(2);  
        String date = rs.getString(3);  
        String writer = rs.getString(4);  
        String file = rs.getString(5);  
  
        //가져올 값을 가방에 넣어줘  
        Notice_BoardVO bag = new Notice_BoardVO();  
        bag.setNo(no);  
        bag.setTitle(title);  
        bag.setDate(date);  
        bag.setWriter(writer);  
        bag.setFile(file);  
  
        //가방에 넣은 값을 더 큰 가방으로 옮겨(컨테이너)  
        list.add(bag);  
    }  
}
```

최신글 순으로 노출하기 위해 sql문에 order by 구문 사용
여러 개의 data를 가져오기 위해 list 로 data를 묶어서 리턴을 받는다

III. 메뉴별 소개 : 게시판(공지사항_수정)


admin님 환영합니다.로그아웃

메인페이지
마이페이지
고객센터
회원 정보


lee님 환영합니다.로그아웃

메인페이지
마이페이지
고객센터
회원 정보

공지사항 세부내용

번호	제목	내용	작성일	작성자	첨부파일	비고
6	취소수수료 기준	취소수수료는 할인쿠폰 적용 전 객실 판매금액 기준으로 계산되며, 실 결제금액에서 취소수수료를 차감한 금액으로 환불 처리됩니다. 만약, 결제 시 포인트를 사용한 경우 포인트 -> 현금 순으로 차감됩니다. 환불 예정금액은 조회 시점 및 실제 환불 진행 시점에 따라 달라질 수 있으므로 예약/결제 진행 당시 안내된 취소/환불 규정을 꼭 숙지하여 이용해 주시길 바랍니다.	2021.02.24	admin	없음	수정하기

돌아가기

공지사항 세부내용

번호	제목	내용	작성일	작성자	첨부파일	비고
6	취소수수료 기준	취소수수료는 할인쿠폰 적용 전 객실 판매금액 기준으로 계산되며, 실 결제금액에서 취소수수료를 차감한 금액으로 환불 처리됩니다. 만약, 결제 시 포인트를 사용한 경우 포인트 -> 현금 순으로 차감됩니다. 환불 예정금액은 조회 시점 및 실제 환불 진행 시점에 따라 달라질 수 있으므로 예약/결제 진행 당시 안내된 취소/환불 규정을 꼭 숙지하여 이용해 주시길 바랍니다.	2021.02.24	admin	없음	

돌아가기

공지사항의 제목을 클릭하여 세부내용 페이지로 연결

잡혀있는 세션값의 계정에 따라서 수정하기 기능 활성화 여부 변동
Admin 계정만 공지사항을 수정할 수 있음.

III. 메뉴별 소개 : 게시판(공지사항_수정)

admin님 환영합니다.로그아웃

메인페이지 마이페이지 고객센터 회원 정보

공지사항 세부내용

제목 : 취소 후 환불 기간

첨부파일 : 없음

내용 :
환불은 예약취소 후 PG사 승인 취소 기간에 따라 주말, 공휴일을 제외한 영업일 기준 최대 3-5일 소요되오니
시간 양해 부탁드립니다.

번호 : 1

수정하기

공지사항으로

```
<hr color = "blue">
<form action="BoardChange.jsp">
  제목 : <input name = "title"><br>
  첨부파일 : <input name = "file"><br>
  내용 : <br>
  <textarea rows="10" cols="100" name = "content"></textarea><br>
  번호 : <input name = "no"><br>
  <button>수정하기</button>
</form>

<!-- <form action="bbs삭제.jsp">
  <button>삭제하기</button>
</form> 나중에 javascript로 하기!-->

<a href="ServiceMain.jsp">
  <button>공지사항으로</button> <!-- 수정하지 않으면 다시 돌아가기 -->
```

수정값을 입력하는 페이지! 입력된 수정값은 수정하기 버튼 클릭시 submit BoardChange.jsp에서 입력값을 받아 처리한다.

III. 메뉴별 소개 : 게시판(공지사항_수정 update 메서드)

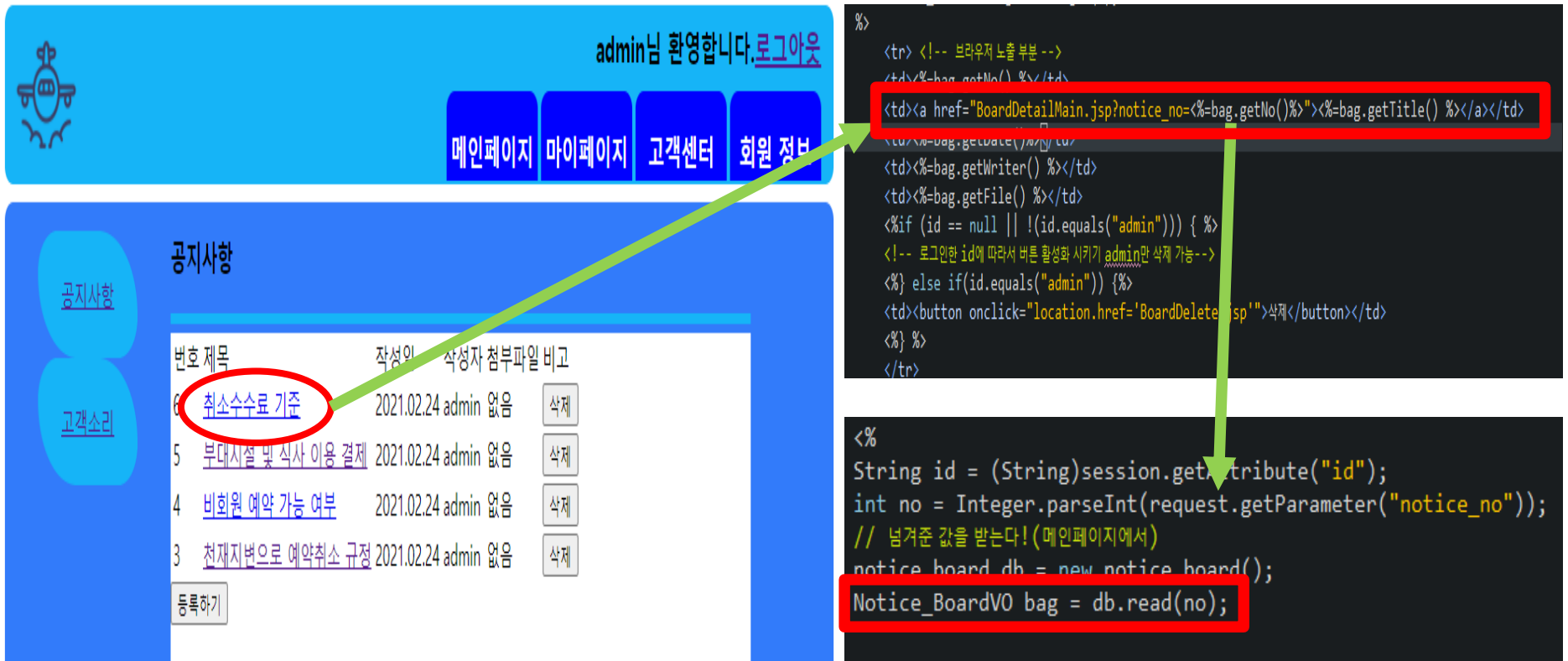
```
6<%  
7    String id = (String)session.getAttribute("id");  
8    //BoardDetail.jsp에서 수정된 입력값을 받아줌  
9    String title = request.getParameter("title");  
10   String file = request.getParameter("file");  
11   String content = request.getParameter("content");  
12   String s1 = request.getParameter("no");  
13   int no = Integer.parseInt(s1); // 유니크값 no 스트림에서 인트로 캐스팅!  
14  
15   Notice_BoardVO bag = new Notice_BoardVO() // bag을 사용하기 위한 vo new(를) 생성!  
16   bag.setTitle(title);  
17   bag.setFile(file);  
18   bag.setContent(content);  
19   bag.setNo(no);  
20   // 각 값을 bag에 넣어줌  
21  
22   notice_board nb = new notice_board(); // update 메서드를 사용하기 위해 new(를) 생성  
23   nb.update(bag); // update 메서드 사용!  
24 %>
```

```
public void update(Notice_BoardVO bag) throws Exception {  
    // 매개변수 전부 넣어줘야한다!
```

```
// 3. SQL문으로 만들기!  
String sql = "update notice_board set notice_title = ?, notice_file = ?, notice_content = ?"  
            + "where notice_no = ?";  
// 변수 sql에 SQL문 중 create에 해당하는 insert문을 넣어준다.!  
PreparedStatement ps = con.prepareStatement(sql);  
// 변수 con으로 메서드 preparedStatement를 사용해서 변수 sql에 있는 문자열을 SQL문으로 변경!  
ps.setString(1, bag.getTitle()); // sql 변수에 입력되는 값을 각각 타입에 맞게 set메서드로 변경  
ps.setString(2, bag.getFile());  
ps.setString(3, bag.getContent());  
ps.setInt(4, bag.getNo());  
System.out.println("SQL문으로 변경완료!"); // SQL문으로 변경위의 내용이 잘 처리되는지 확인  
  
// 4. SQL문을 MySQL 서버로 전송!  
ps.executeUpdate(); // ps의 메서드인 executeUpdate로 MySQL로 전송!  
System.out.println("전송완료!"); // 전송과정 문제 확인 출력!
```

입력값을 받아서 update 메서드를 처리 Update 메서드를 처리
할때
notice_no의 유니크한 값을 가지고 해당 row값의 data 수정

III. 메뉴별 소개 : 게시판(공지사항 세부사항)



admin님 환영합니다.로그아웃

메인페이지 마이페이지 고객센터 회원 정보

공지사항

공지사항

고객소리

번호	제목	작성일	작성자	첨부파일	비고
6	<u>최소수수료 기준</u>	2021.02.24	admin	없음	<input type="button" value="삭제"/>
5	<u>분대시설 및 식사 이용 결제</u>	2021.02.24	admin	없음	<input type="button" value="삭제"/>
4	<u>비회원 예약 가능 여부</u>	2021.02.24	admin	없음	<input type="button" value="삭제"/>
3	<u>천재지변으로 예약취소 규정</u>	2021.02.24	admin	없음	<input type="button" value="삭제"/>

```
%>
<tr> <!-- 브라우져 노출 부분 -->
<td><a href="BoardDetailMain.jsp?notice_no=<%=bag.getNo()%>"><%=bag.getTitle() %></a></td>
</td><%=bag.getDate() %></td>
<td><%=bag.getWriter() %></td>
<td><%=bag.getFile() %></td>
<%=if (id == null || !(id.equals("admin"))) { %>
<!-- 로그인한 id에 따라서 버튼 활성화 시키기 admin만 삭제 가능-->
<%= else if(id.equals("admin")) { %>
<td><button onclick="location.href='BoardDelete.jsp'">삭제</button></td>
<%= %>
</tr>

<%
String id = (String)session.getAttribute("id");
int no = Integer.parseInt(request.getParameter("notice_no"));
// 넘겨준 값을 받는다! (메인페이지에서)
notice_board db = new notice_board();
Notice_BoardVO bag = db.read(no);
```

공지사항의 제목을 클릭하여 세부내용 페이지로 연결 처리과정

입력받은 notice_no 컬럼을 유니크 값으로 해서 read 메서드를 처리

III. 메뉴별 소개 : 게시판(공지사항 세부사항_read 메서드)

```
public Notice_BoardVO read(int no) throws Exception { //입력하는 매개변수가 no이다

//3. SQL문으로 바꾸기
String sql = "select * from notice_board where notice_no = ?";
//내가 필요한 SQL문 만들기
PreparedStatement ps = con.prepareStatement(sql); // SQL문으로 변경!
ps.setInt(1, no);
System.out.println("SQL문 만들기 성공");

//4. 서버로 보내기!
ResultSet rs = ps.executeQuery(); //DB에 있는 테이블 쿼리 값을 rs에 넣는다!
System.out.println("SQL문 서버로 전송 성공!");
//VO를 만들고!
Notice_BoardVO bag = new Notice_BoardVO();
// System.out.println(rs.next()); 확인해주기 위한 출력
if(rs.next()) { //rs.next가 true 라면
    //가져올 값을 먼저 변수로 설정 꺼내주는 방법은 2가지 인덱스 or 항목명
    int no1 = rs.getInt(1);
    String title = rs.getString(2);
    String content = rs.getString(6);
    String date = rs.getString(3);
    String writer = rs.getString(4);
    String file = rs.getString(5);

    bag.setNo(no1);
    bag.setTitle(title);
    bag.setContent(content);
    bag.setDate(date);
    bag.setWriter(writer);
    bag.setFile(file);
}
return bag;
```

Read 메서드 처리과정

유니크 값인 notice_no를 조건값으로 하고 모든 data를 가져오는 sql 문을 사용
data 값이 많기 때문에 VO를 사용하요 한번에 data값을 넣고 빼고 가능

III. 메뉴별 소개 : 게시판(공지사항_삭제)

The diagram illustrates the deletion process of a notice. It shows three screenshots of a web application interface.

Left Screenshot (Notice List): Displays a list of notices. The '삭제' (Delete) button for the notice with number 6 is circled in red. A green arrow points from this button to the deletion form.

번호	제목	작성일	작성자	첨부파일	비고
6	취소 후 환불 기간	2021.02.24	admin	없음	삭제
5	부대시설 및 식사 이용 결제	2021.02.24	admin	없음	삭제
4	비회원 예약 가능 여부	2021.02.24	admin	없음	삭제
3	천재지변으로 예약취소 규정	2021.02.24	admin	없음	삭제

Middle Screenshot (Deletion Form): Shows the '공지사항 삭제' (Notice Deletion) form. The '번호' (Number) field is filled with '6'. The '삭제하기' (Delete) button is highlighted. A green arrow points from this button to the right screenshot.

Right Screenshot (Updated Notice List): Shows the '공지사항' (Notice) section after deletion. The notice with number 6 has been removed from the list.

번호	제목	작성일	작성자	첨부파일	비고
5	부대시설 및 식사 이용 결제	2021.02.24	admin	없음	삭제
4	비회원 예약 가능 여부	2021.02.24	admin	없음	삭제
3	천재지변으로 예약취소 규정	2021.02.24	admin	없음	삭제

삭제 처리 과정

III. 메뉴별 소개 : 게시판(공지사항_삭제 delete 메서드)

```
<tr><!-- 브라우저 노출 부분 -->
<td><%=bag.getNo() %></td>
<td><a href="BoardDetailMain.jsp?notice_no=<%=bag.getNo()%>"><%=bag.getTitle() %></a></td>
<td><%=bag.getDate()%></td>
<td><%=bag.getWriter() %></td>
<td><%=bag.getFile() %></td>

<%if (id == null || !id.equals("admin")) { %>
<!-- 로그인한 id에 따라서 버튼 활성화 시키기 admin만 삭제 가능-->
<%} else if(id.equals("admin")) {%>
<td><button onclick="location.href='BoardDelete.jsp'">삭제</button></td>

<%} %>
</tr>
```

```
<h3>공지사항 삭제</h3>
<hr color = "blue">
<form action="DeleteSubmit.jsp"> <!-- 삭제 처리 -->
    번호 : <input name = "no"><br>
    <button>삭제하기</button>
</form>
```

```
<%  
//BoardDelete.jsp에서 입력된 값 가져오기!  
String s1 = request.getParameter("no"); // 입력값은 모두 스트링이다!  
int no = Integer.parseInt(s1); // 스트링으로 입력 받은 값을 int로 캐스팅!  
  
String id = (String)session.getAttribute("id");  
  
notice board nb = new notice_board(); // delete메서드 사용을 위한 new(들) 생성  
nb.delete(no); // 메서드 사용!
```

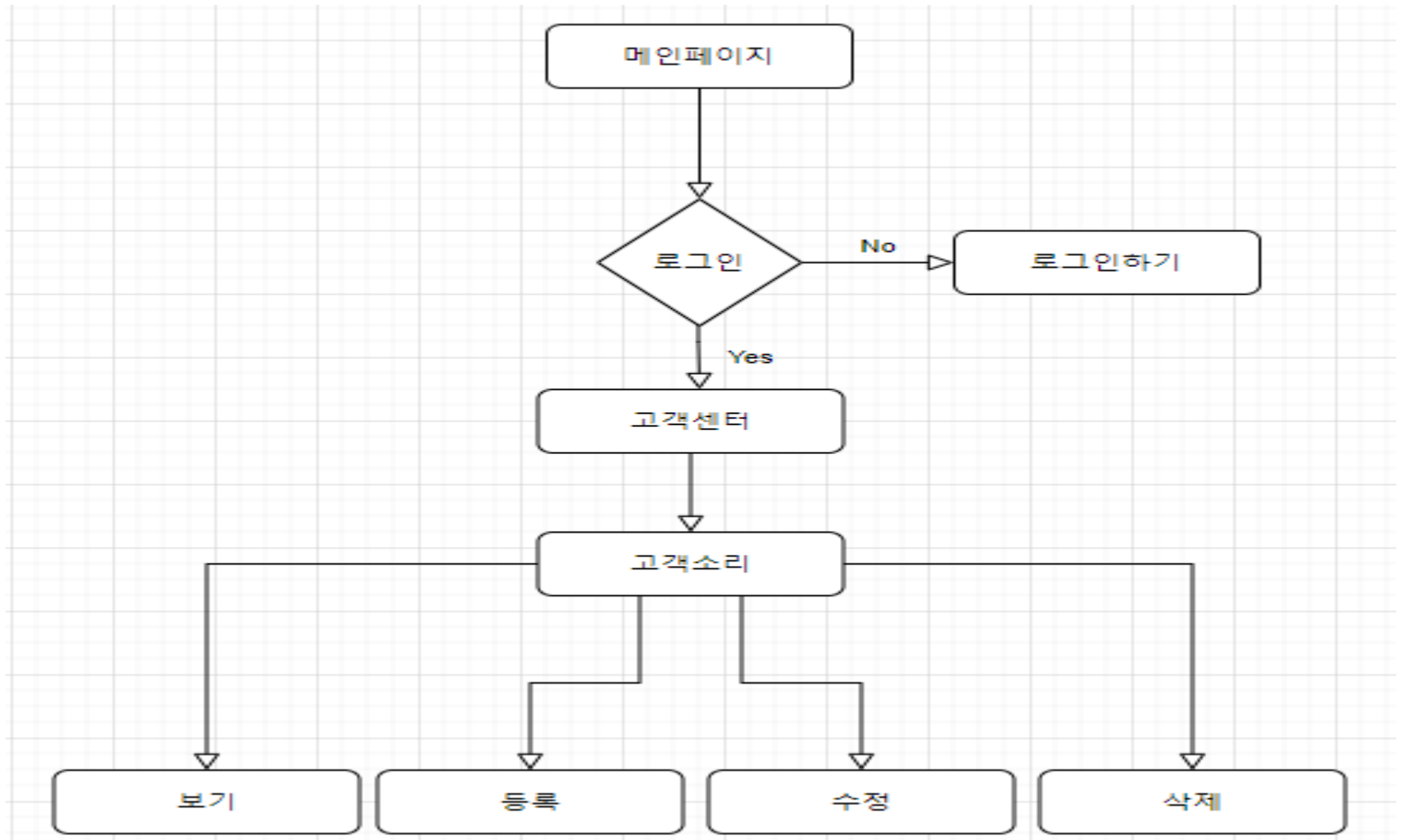
```
// 3. SQL문으로 만들기
String sql = "delete from notice_board where notice_no = ?";
// 변수 sql에 SQL문 중 create에 해당하는 insert문을 넣어준다.
PreparedStatement ps = con.prepareStatement(sql);
// 변수 con으로 메서드 preparedStatement를 사용해서 변수 sql에 있는 문자열을 SQL문으로 변경!
ps.setInt(1, no); // sql 변수에 입력되는 값을 각각 타입에 맞게 set메서드로 변경
System.out.println("SQL문으로 변경완료!"); // SQL문으로 변경위의 내용이 잘 처리되는지 확인

// 4. SQL문을 mySQL 서버로 전송!
ps.executeUpdate(); // ps의 메서드인 executeUpdate로 mySQL로 전송!
System.out.println("전송완료!"); // 전송과정 문제 확인 출력!
```

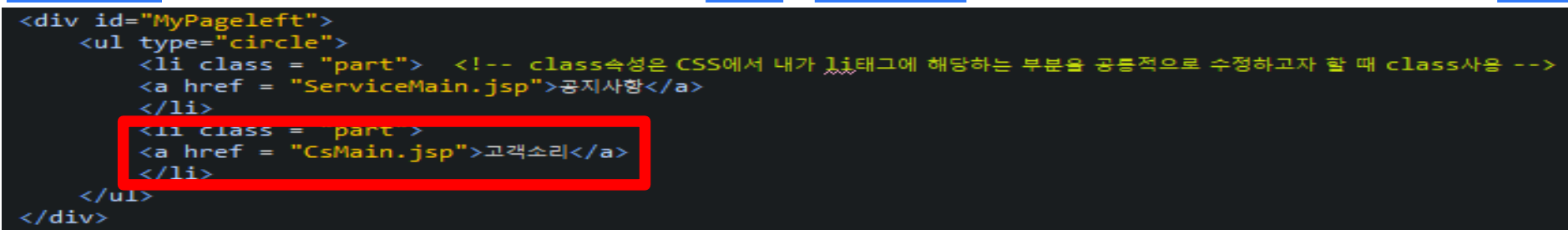
1. Admin 세션의 경우 삭제 버튼 활성화
2. 유니크값으로 사용하기 위한 notice no 를 입력해서 submit

3. 입력값을 받아 delete 처리
4. notice_no를 유니크 값으로 받고 delete sql문 실행

III. 메뉴별 소개 : 고객센터 고객센터 설계도



III. 메뉴별 소개 : 게시판(고객소리 list 메서드)



고객센터 페이지에서 왼쪽 고객소리 클릭시 고객소리(CsMaing.jsp) 페이지로 이동
<a href> 사용

III. 메뉴별 소개 : 게시판(고객소리 list 메서드)

lee님 환영합니다.로그아웃

메인페이지마이페이지고객센터회원 정보

공지사항

고객소리

고객의 소리

고객의 소리(좋아요)

번호	제목	업체명	작성일	작성자	평점	비고
2	여기라면 다음에 또 이용할 거예요.	라마다 서울 호텔	2021.02.24	won	4	<input type="button" value="삭제"/>
1	여기만한 곳은 어디에도 없을 거예요.	오크우드 프리미어 코엑스센터	2021.02.24	lee	5	<input type="button" value="삭제"/>

고객의 소리(나빠요)

번호 제목 업체명 작성일 작성자 평점 비고

```
<!-- 노출되는 값이 여러 row값이기 때문에 for문을 돌려준다! -->
%
for (int i = 0; i < list.size(); i++) {
    CustomerSoundVO bag = list.get(i);
    if (bag.getCustomer_type().equals("좋아요")) {
%<!-- list에서 가져온 bag에서 값을 빼야한다! -->
<tr>
    <td><%=bag.getCustomer_no()%></td>
    <td><a href="Csdetail.jsp?customer_no=<%=bag.getCustomer_no()%>"><%=bag.getCustomer_title()%></a></td>
    <td><%=bag.getCustomer_accommodation_name()%></td>
    <td><%=bag.getCustomer_date()%></td>
    <td><%=bag.getCustomer_writer()%></td>
    <td><%=bag.getCustomer_grade()%></td>
    <td><button onclick="location.href='CsDelete.jsp'">삭제</button></td>
</tr>
```

List 메서드를 사용해서 게시판의 내용을 노출! 좋아요와 나빠요를 구별
For문들 통해서 list.size 만큼 테이블 값을 가져온다.

III. 메뉴별 소개 : 게시판(고객소리 list 메서드)

```
public ArrayList<CustomerSoundVO> list() throws Exception {
    // 1. 커넥트 연결
    Class.forName("com.mysql.jdbc.Driver"); // 경로지정
    System.out.println("연결완료!");
    // 2. Db연결하기
    String url = "jdbc:mysql://localhost:3306/jeju?useUnicode=true&characterEncoding=utf8";
    // 연결할 url 입력!
    Connection con = DriverManager.getConnection(url, "root", "1234");
    // 커넥션 메서드 db연결 drivermanager.getConnection(3번째꺼 전부 입력하느거)
    System.out.println("db연결 완료!");
    // 3. SQL문으로 변경하기!
    String sql = "select * from customer_sound order by customer_no desc";
    // 먼저 sql문 만들기! order by 컬럼명 desc 내림차순 정리!
    PreparedStatement ps = con.prepareStatement(sql);
    // sql문으로 변환하기!

    ResultSet rs = ps.executeQuery();
    // 리드 리스트 반환값이 있는 애들은 쿼리문을 사용!
    System.out.println("서버전송 성공!");
    // 쿼리문은 true와 false로 나온다!

    // 반복문을 통해 다수의 데이터를 가져온다(게시판 같은 한번에 많은 데이터를 가져와야할때 반복문 사용!
    while (rs.next()) { // rs문이 참일때를 의미 loop 루프문으로 사용!
        int customer_no = rs.getInt(1);
        String customer_type = rs.getString(2);
        String customer_title = rs.getString(3);
        String customer_accommodation_name = rs.getString(4);
        String customer_date = rs.getString(5);
        String customer_writer = rs.getString(6);
        int customer_grade = rs.getInt(8);
        // 가져올 값에 대한 변수를 지정!

        // 가방 만들어주기!
        CustomerSoundVO bag = new CustomerSoundVO();
        bag.setCustomer_no(customer_no);
        bag.setCustomer_type(customer_type);
        bag.setCustomer_title(customer_title);
        bag.setCustomer_accommodation_name(customer_accommodation_name);
        bag.setCustomer_date(customer_date);
        bag.setCustomer_writer(customer_writer);
        bag.setCustomer_grade(customer_grade);
        // 가방에 각 값 넣어주기!

        // list만들어서 가방 전부 넣어주기! 가장위에 반복문 전에 변수로 선언해주기!
        list.add(bag); // bag으로 받은 값들을 전부 list에 넣어주기!
    }
    System.out.println(list.size()); // 리스트 안에 개수가 잘 들어가 있는지 확인 하는 프린트!
    return list; // 리턴값은 list의 타입인 ArrayList<입력받은 변수(bag)의 타입>이다!
}
```

1. 커넥트 연결하기
- Class 부품 사용

2. Db연결하기
- Connection 부품사용

3. SQL문으로 변경하기
- PreparedStatement 부품사용

4. Db서버에서 테이블 값 저장하기
- executeQuery 부품사용
- resultSet 의 값은 true/false
- resultSet 값이 true인 경우
- while문 처리

III. 메뉴별 소개 : 게시판(고객소리_등록)

공지사항

고객소리

고객의 소리

고객의 소리(좋아요)

번호	제목	업체명	작성일	작성자	평점	비고
2	여기라면 다음에 또 이용할 거예요.	라마다 서울 호텔	2021.02.24	won	4	삭제
1	여기만한 곳은 어디에도 없을 거예요.	오크우드 프리미어 코엑스센터	2021.02.24	lee	5	삭제

고객의 소리(나빠요)


번호 제목 업체명 작성일 작성자 평점 비고

[등록하기](#)

```
<%  
    if (session.getAttribute("id") == null) {  
    %>  
    <!-- 세션이 잡혀 있지 않으면 로그인하기! -->  
    <form action="Login.jsp">  
        <button>로그인하기</button>  
    </form>  
    <%  
        } else {  
    %>  
    <!-- 세션이 잡혀 있으면 등록하기 활성화 -->  
    <form action="CsEnroll.jsp">  
        <button>등록하기</button>  
    </form>  
    <%>  
    %>
```

세션이 잡혀 있지 않으면 등록하기 기능 비활성화
세션이 있는 경우에만 등록하기 활성화

III. 메뉴별 소개 : 게시판(고객소리_등록)

lee님 환영합니다.[로그아웃](#)

[메인페이지](#)[마이페이지](#)[고객센터](#)[회원 정보](#)

고객의 소리 등록페이지

제목 :

종류 :

업체명 :

날짜 :

작성자 :

내용 :
다 좋았는데 사진에 비해 방이 조금 좁고 추웠어요 ㅠ 날씨가 추운 것도 한 몫 한 것 같습니다.

평점 :

```
<h3>고객의 소리 등록페이지</h3>
<!-- 고객의 소리 등록시 입력값 받는 곳 -->
<hr color="black">
<form action="CsEnrollSubmit.jsp">
제목 : <input name = "customer_title"><br>
종류 :
<select name = "customer_type">
    <option value="좋아요">좋아요</option>
    <option value="나빠요">나빠요</option>
</select><br>
업체명 : <input name = "customer_accommdation_name"><br>
날짜 : <input name = "customer_date"><br>
작성자 : <input name = "customer_writer"><br>
내용 : <br>
<textarea rows = "10" cols = "100" name = "customer_content"></textarea><br>
평점 :
<select name = "customer_grade">
    <option value = 5>5</option>
    <option value = 4>4</option>
    <option value = 3>3</option>
    <option value = 2>2</option>
    <option value = 1>1</option>
</select>
<button>등록하기</button>
</form>
```

고객의소리 등록할 data 값을 입력을 받는다.

III. 메뉴별 소개 : 게시판(고객소리_등록 create 메서드)

```
6.0.0.0
7 // 각각의 변수 타입에 맞게 입력된 값을 받아준다!
8 String customer_title = request.getParameter("customer_title");
9 String customer_type = request.getParameter("customer_type");
10 String customer_accommodation_name = request.getParameter("customer_accommodation_name");
11 String customer_date = request.getParameter("customer_date");
12 String customer_writer = request.getParameter("customer_writer");
13 String customer_content = request.getParameter("customer_content");
14 int customer_grade = Integer.parseInt(request.getParameter("customer_grade"));

6 // 로그인 세션
7 String id = (String)session.getAttribute("id");
8
9 // 입력된 값을 메서드를 사용하여 SQL문으로 변경하여 서버로 전송해준다(create메서드 사용)
10 // 새로운 메서드 사용하기 위해서 new 객체 만들기
11 CustomerSoundDB db = new CustomerSoundDB();
12 db.create(customer_type, customer_title, customer_accommodation_name, customer_date, customer_writer, customer_content, customer_grade);
13 // SQL문으로 보내기!
14 %>
```

등록페이지에서 입력한 값을 받아서
Create메서드 처리

```
public void create(String customer_type, String customer_title, String customer_accommodation_name,
String customer_date, String customer_writer, String customer_content, int customer_grade)
throws Exception {
```

```
// 3. SQL문으로 변경
String sql = "insert into customer_sound values (null, ?, ?, ?, ?, ?, ?, ?)";
// SQL문에 맞게 쿼리문 생성!
PreparedStatement ps = con.prepareStatement(sql);
// sql 문으로 변경!
ps.setString(1, customer_type);
ps.setString(2, customer_title);
ps.setString(3, customer_accommodation_name);
ps.setString(4, customer_date);
ps.setString(5, customer_writer);
ps.setString(6, customer_content);
ps.setInt(7, customer_grade);
// 각 요소에 맞는 값을 넣어준다.
System.out.println("SQL문으로 변경완료");

// 4. MySQL db로 전송!
ps.executeUpdate(); // MySQL 서버로 전송하는 메서드!
System.out.println("MySQL 서버전송 성공!");
```

입력값을 받지 않는 컬럼에는 null을 입력
customer_no는 오토 인크리먼트
처리된 입력값을 MySQL db에 저장

III. 메뉴별 소개 : 게시판(고객소리 read 메서드)



lee님 환영합니다.로그아웃

메인페이지

마이페이지

고객센터

회원 정보

고객의 소리 세부내용

번호	제목	업소명	내용	날짜	작성자	수정
3	여기라면 다음에 또 이용할 거예요.	호텔 캘리년 개포	다 좋았는데 사진에 비해 방이 조금 좁고 추웠어요ㅠ 날씨가 추운 것도 한 몫 한 것 같습니다.	2021.02.21	lee3	수정하기

```
<%
String id = (String)session.getAttribute("id");
int customer_no = Integer.parseInt(request.getParameter("customer_no"));
// 입력값을 받아주기: 입력받은 값이 없기 때문에 int 형에 null을 넣어야 합니다!
CustomerSoundDB db = new CustomerSoundDB(); // read 메서드를 쓰기위한 설정!
CustomerSoundVO bag = db.read(customer_no) // read 메서드를 호출할때 bag에 있는 값을 꺼내기 위한 설정!
%>

<tr>
<td><%=bag.getCustomer_no() %></td>
<td><%=bag.getCustomer_type() %></td>
<td><%=bag.getCustomer_title() %></td>
<td><%=bag.getCustomer_accommodation_name() %></td>
<td><%=bag.getCustomer_content() %></td>
<td><%=bag.getCustomer_date() %></td>
<td><%=bag.getCustomer_writer() %></td>
<td><%=bag.getCustomer_grade() %></td>
<td><button onclick="location.href='CsChange.jsp'">수정하기</button></td>
</tr>
</table>
```

입력된 customer_no 값을 받아서 해당 row의 테이블만 호출시킨다.
Read 메서드 사용

III. 메뉴별 소개 : 게시판(고객소리 read 메서드)

```
public CustomerSoundVO read(int customer_no) throws Exception { //리턴받는 bag의 타입을 void 위치에 넣기!
```

```
//3. SQL문 만들기!  
String sql = "select * from customer_sound where customer_no = ?"  
// sql문 변수로 만들어 놓기!  
PreparedStatement ps = con.prepareStatement(sql);
```

```
//4. DB서버로 전송!  
ResultSet rs = ps.executeQuery(); // 가져오는 값은 resultset -> 엑스큐트쿼리문 사용!  
System.out.println("DB서버 전송 성공!");
```

```
CustomerSoundVO bag = new CustomerSoundVO();  
if (rs.next()) { // rs.next()는 true, false 값이고 해당 조건은 true일때이다!  
    // 가져와야하는 값을 전부 bag에 넣어준다!  
    int customer_no1 = rs.getInt(1);  
    String customer_type = rs.getString(2);  
    String customer_title = rs.getString(3);  
    String customer_accommodation_name = rs.getString(4);  
    String customer_date = rs.getString(5);  
    String customer_writer = rs.getString(6);  
    String customer_content = rs.getString(7);  
    int customer_grade = rs.getInt(8);  
    // 변수에 먼저 담아둔 후 bag에 담기!  
    bag.setCustomer_no(customer_no1);  
    bag.setCustomer_type(customer_type);  
    bag.setCustomer_title(customer_title);  
    bag.setCustomer_accommodation_name(customer_accommodation_name);  
    bag.setCustomer_date(customer_date);  
    bag.setCustomer_writer(customer_writer);  
    bag.setCustomer_content(customer_content);  
    bag.setCustomer_grade(customer_grade);  
    // 각각의 가방에 담기!
```

```
return bag;
```

read 메서드 처리과정

Sql문의 조건값은 유니크 값인
customer_no로 지정

가져올 data 값이 많아서 VO를
활용해서 한번에 처리함.

III. 메뉴별 소개 : 게시판(고객소리_수정)

lee님 환영합니다.로그아웃

메인페이지 마이페이지 고객센터 회원 정보

게시글 번호 : 3

종류 : 나빠요 ▾

제목 : 다음에는 이용 안해요!

업소명 : 호텔 컬리번 개포

내용 :
상태가 너무 안 좋았고 청소도 안되어 있었어요! 다음에는 이용 안해요!

평점 : 5 수정하기

```
61 form action="CsChangeSubmit.jsp">
62 <!-- 수정 내용을 입력하기! -->
63 게시글 번호 : <input name = "customer_no"><br>
64 종류 : <select name = "customer_type">
65 <option value = "좋아요">좋아요</option>
66 <option value = "나빠요">나빠요</option>
67 </select><br>
68 제목 : <input name = "customer_title"><br>
69 업소명 : <input name = "customer_accommodation_name"><br>
70 내용 : <br>
71 <textarea rows="10" cols="100" name="customer_content"></textarea><br>
72 평점 : <select name = "customer_grade">
73 <option value = "5">5</option>
74 <option value = "4">4</option>
75 <option value = "3">3</option>
76 <option value = "2">2</option>
77 <option value = "1">1</option>
78 </select>
79
80 <button>수정하기</button> <!-- 클릭으로 제출 디폴트값이 submit -->
81 </form>
```

고객의 소리 수정사항 입력하는 과정
수정하기 클릭시 CsChangeSubmit.jsp전송

III. 메뉴별 소개 : 게시판(고객소리_수정 update 메서드)

```
<%
//로그인 세션
String id = (String)session.getAttribute("id");
// CsChange에서 입력된 값을 받아와준다
String customer_type = request.getParameter("customer_type");
String customer_title = request.getParameter("customer_title");
String customer_accommodation_name = request.getParameter("customer_accommodation_name");
String customer_content = request.getParameter("customer_content");
String s1 = request.getParameter("customer_grade"); // 입력값은 항상 스트링! 파싱필요!
String s2 = request.getParameter("customer_no");
int customer_grade = Integer.parseInt(s1);
int customer_no = Integer.parseInt(s2);
// 입력된값 받아주기!

//가방 만들기! 가방은 vo에서 만들기!
CustomerSoundVO bag = new CustomerSoundVO();
bag.setCustomer_type(customer_type);
bag.setCustomer_title(customer_title);
bag.setCustomer_accommodation_name(customer_accommodation_name);
bag.setCustomer_content(customer_content);
bag.setCustomer_grade(customer_grade);
bag.setCustomer_no(customer_no);
// 가방에 넣었으면 update 메서드를 사용해야지!
CustomerSoundDB db = new CustomerSoundDB();
db.update(bag);
%>
```


CsChange.jsp에서 입력된 값을 받아서
read 메서드 처리

```
public void update(CustomerSoundVO bag) throws Exception { // cud는 가져오는 값이 없기 때문에 void사용!

//3. SQL문 변경하기!
String sql = "update customer_sound set customer_type = ?, customer_title = ?, "
+ "customer_accommodation_name = ?, customer_content = ?, customer_grade = ? "
+ "where customer no = ?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, bag.getCustomer_type());
ps.setString(2, bag.getCustomer_title());
ps.setString(3, bag.getCustomer_accommodation_name());
ps.setString(4, bag.getCustomer_content());
ps.setInt(5, bag.getCustomer_grade());
ps.setInt(6, bag.getCustomer_no());
System.out.println("SQL문 변경 및 입력 완료!");
//4. 서버전송하기!
ps.executeUpdate();
System.out.println("서버전송완료!");
```

유니크한 값 customer_no조건으로
변경된 data를 각 database에 저장

III. 메뉴별 소개 : 게시판(고객소리_삭제)

lee님 환영합니다.로그아웃

메인페이지마이페이지고객센터회원 정보

공지사항

고객소리

고객의 소리


고객의 소리(좋아요)

번호	제목	업체명	작성일	작성자	평점	비고
2	여기라면 다음에 또 이용할 거예요.	라마다 서울 호텔	2021.02.24	won	4	삭제
1	여기만한 곳은 어디에도 없을 거예요.	오크우드 프리미어 코엑스센터	2021.02.24	lee	5	삭제

고객의 소리(나빠요)

번호	제목	업체명	작성일	작성자	평점	비고
4	너무 별로였어요	호텔 캘리년 개포	2021.02.24	lee	2	삭제
3	다음에는 이용 안해요!	호텔 캘리년 개포	2021.02.21	lee	1	삭제

[등록하기](#)

lee님 환영합니다.로그아웃

메인페이지마이페이지고객센터회원 정보

고객의 소리 삭제하기

게시글 번호 :

[삭제하기](#)

삭제 버튼 클릭시 유니크 값인
게시글 번호(customer_no) 입력 페이지

III. 메뉴별 소개 : 게시판(고객소리_삭제 delete 메서드)

```
<h3>고객의 소리 삭제하기</h3>
<hr color = "green">
<form action="CsDeleteSubmit.jsp"
  게시글 번호 : <input name = "customer_no"><br>
  <!-- 유니크한 값인 게시글 번호로 삭제하기! -->
  <button>삭제하기</button>
</form>
```

```
<!-- 삭제파트에서 넘어온 입력값(게시글번호) 받기 -->
<%
// CsDelete에서 입력한 값을 받아 줘야 한다!
String s1 = request.getParameter("customer_no");
int customer_no = Integer.parseInt(s1);
//입력값은 스트링이기 때문에 인트로 캐스팅

String id = (String)session.getAttribute("id");

CustomerSoundDB db = new CustomerSoundDB(); // delete 메서드 사용 new
db.delete(customer_no); // 메서드 사용
```

```
public void delete(int customer_no) throws Exception {
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("연결성공!");

    String url = "jdbc:mysql://localhost:3306/jeju"; // 입력값이 숫자라서 인코딩 불필요!
    Connection con = DriverManager.getConnection(url, "root", "1234");
    System.out.println("db연결 성공!");

    String sql = "delete from customer sound where customer no = ?";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setInt(1, customer_no);
    System.out.println("SQL문 변경 완료!");

    ps.executeUpdate();
    System.out.println("서버전송 완료!");
}
```

유니크한 값인 customer_no 값으로 해당 테이블을 전체 delete 하는 메서드 사용

III. 메뉴별 소개 : 게시판(고객소리_삭제)



lee님 환영합니다.로그아웃

메인페이지마이페이지고객센터회원 정보

공지사항
고객소리

고객의 소리

고객의 소리(좋아요)

번호	제목	업체명	작성일	작성자	평점	비고
2	여기라면 다음에 또 이용할 거예요.	라마다 서울 호텔	2021.02.24	won	4	<div>삭제</div>
1	여기만한 곳은 어디에도 없을 거예요.	오크우드 프리미어 코엑스센터	2021.02.24	lee	5	<div>삭제</div>

고객의 소리(나빠요)

번호	제목	업체명	작성일	작성자	평점	비고
3	다음에는 이용 안해요!	호텔 컬리년 개포	2021.02.21	lee	1	<div>삭제</div>

등록하기

Database 에서 삭제된 data는 웹브라우저에서도 삭제

IV. 소감문 : 이동형

첫 프로젝트에서 게시판 파트를 담당하였습니다. 게시판 구축에 있어서 가장 중점적으로 사용한 것은 CRUD 부분입니다. 게시판의 특성상 등록, 보기, 수정, 삭제 모두를 한번에 적용할 수 있었던 프로젝트였습니다.

프로젝트를 진행하면서 가장 중요하게 생각된 부분은 처음 기획 부분입니다. 정확하고 구체적인 기획이 진행 되어야 개발을 하는 부분에서 기획을 다시 하는 일이 없을 것 같습니다.

게시판 구축 부분에서 힘들었던 부분은 공지사항 파트에서 모든 id 가 로그인시 공지사항을 전부 등록, 보기, 수정, 삭제가 가능한 것을 admin 계정만 등록 삭제 수정이 가능하게 만드는 것이였습니다.. 해당 부분을 해결하기 위해서 로그인 후 세션값을 끌어와서 세션값이 admin 이면 등록, 보기, 삭제가 가능하게 if 조건문을 활용했습니다.

Jdbc를 활용해서 브라우저, server, database까지 연결하여 처음 작업을 해보았는데 일상적으로 많이 사용하는 인터넷 기능들이 어떻게 연결되고 처리되고 있는지 알게 되는 프로젝트였습니다.