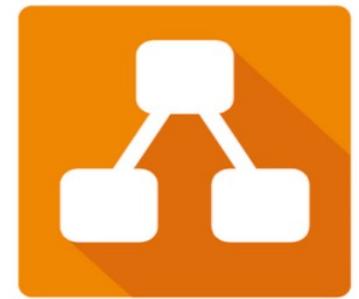
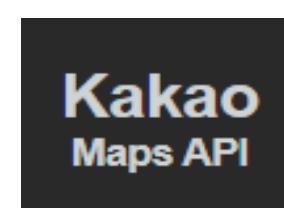


# Jsp와 MySQL을 활용한 제주도 호텔 예약 웹사이트 구현

이준규

# 사용 프로그램



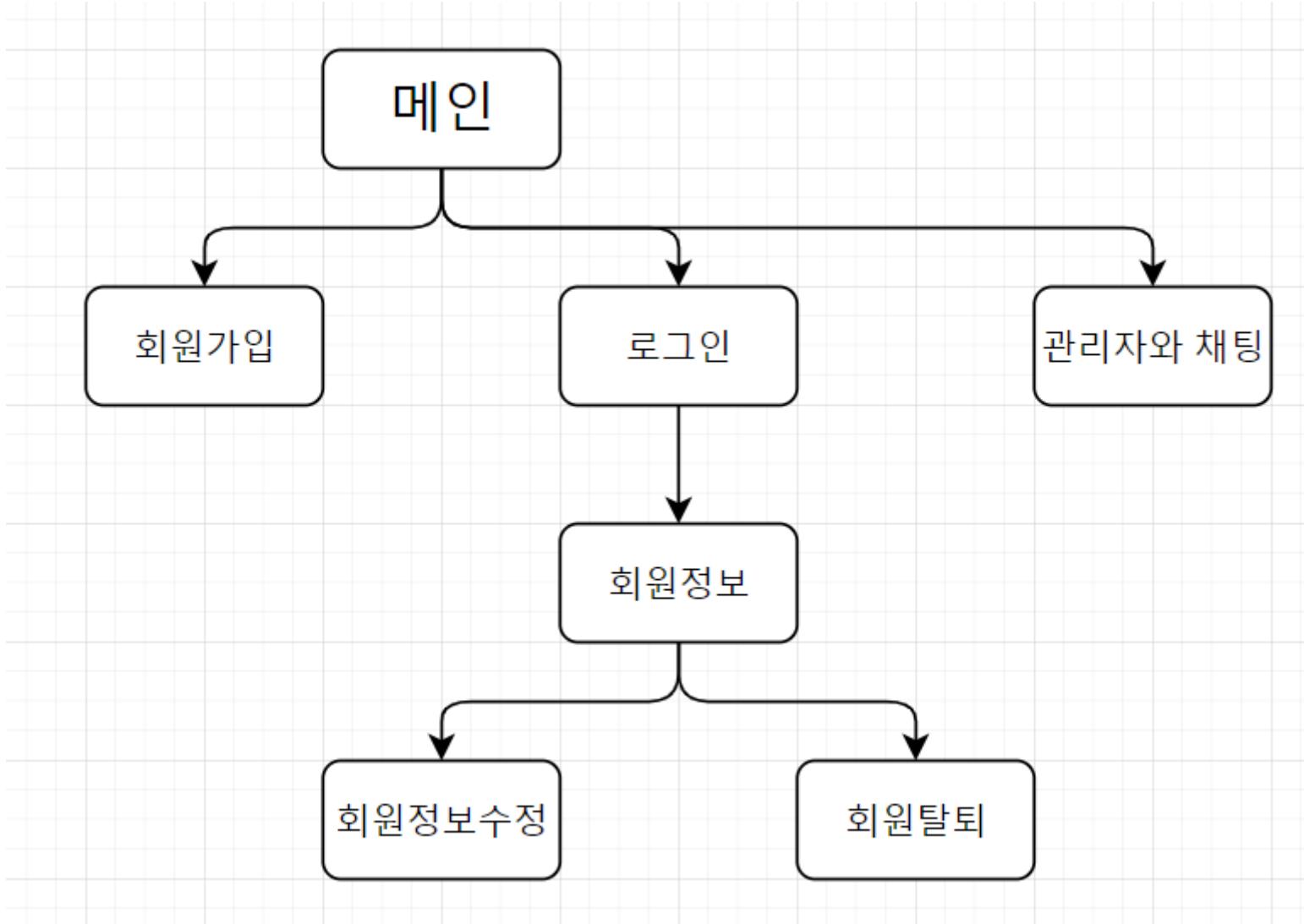
draw.io



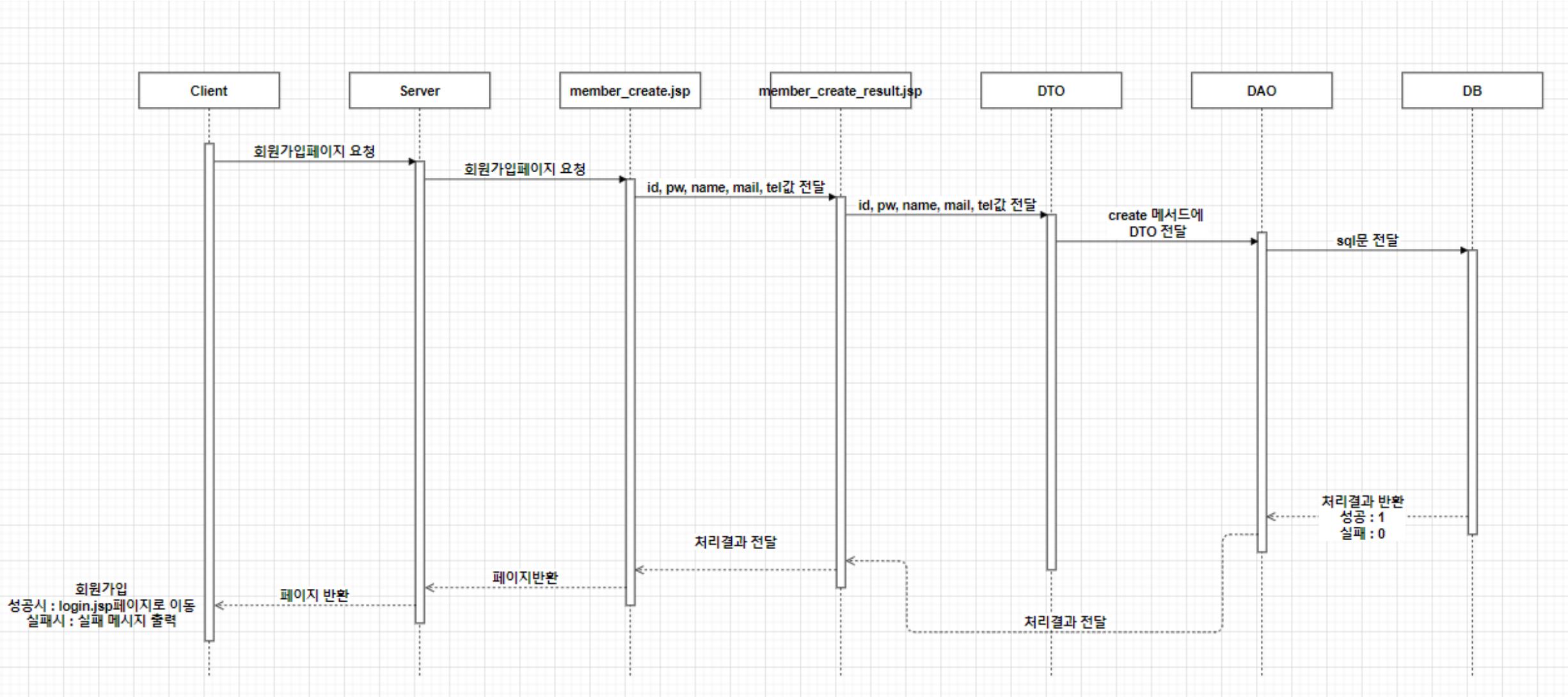
## 프로젝트 개요

프로젝트명	jsp와 mySQL을 이용한 제주도 호텔 예약 웹사이트 구현
담당업무	메인페이지, 회원가입, 로그인, 회원정보, 채팅
참여인원	5명
목표	<ul style="list-style-type: none"> <li>■ 회원가입, 로그인, 회원정보수정, 탈퇴기능</li> <li>■ 관리자와 유저의 1:1 대화 기능</li> </ul>
개발 환경	Windows10
사용 도구	Eclipse, mySQL, JDBC, Dbeaver, Maven, Slack, Draw.io, Apache Tomcat, Object aid, Github, Figma, iimport, kakao map api
사용 기술	Java, JavaScript, Jquery, JSP, HTML, CSS, SQL, Json, Websocket, open api, ajax

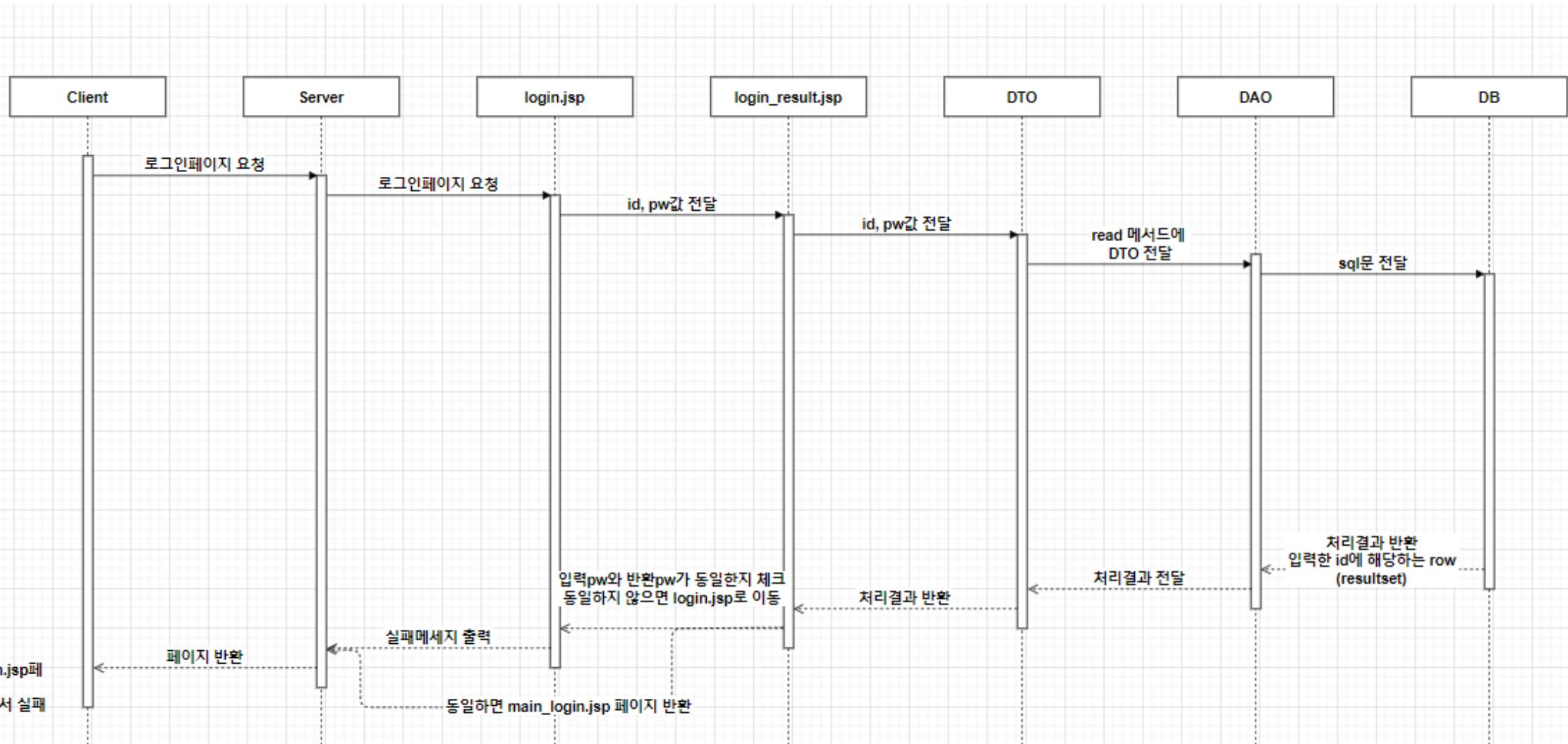
## 순서도



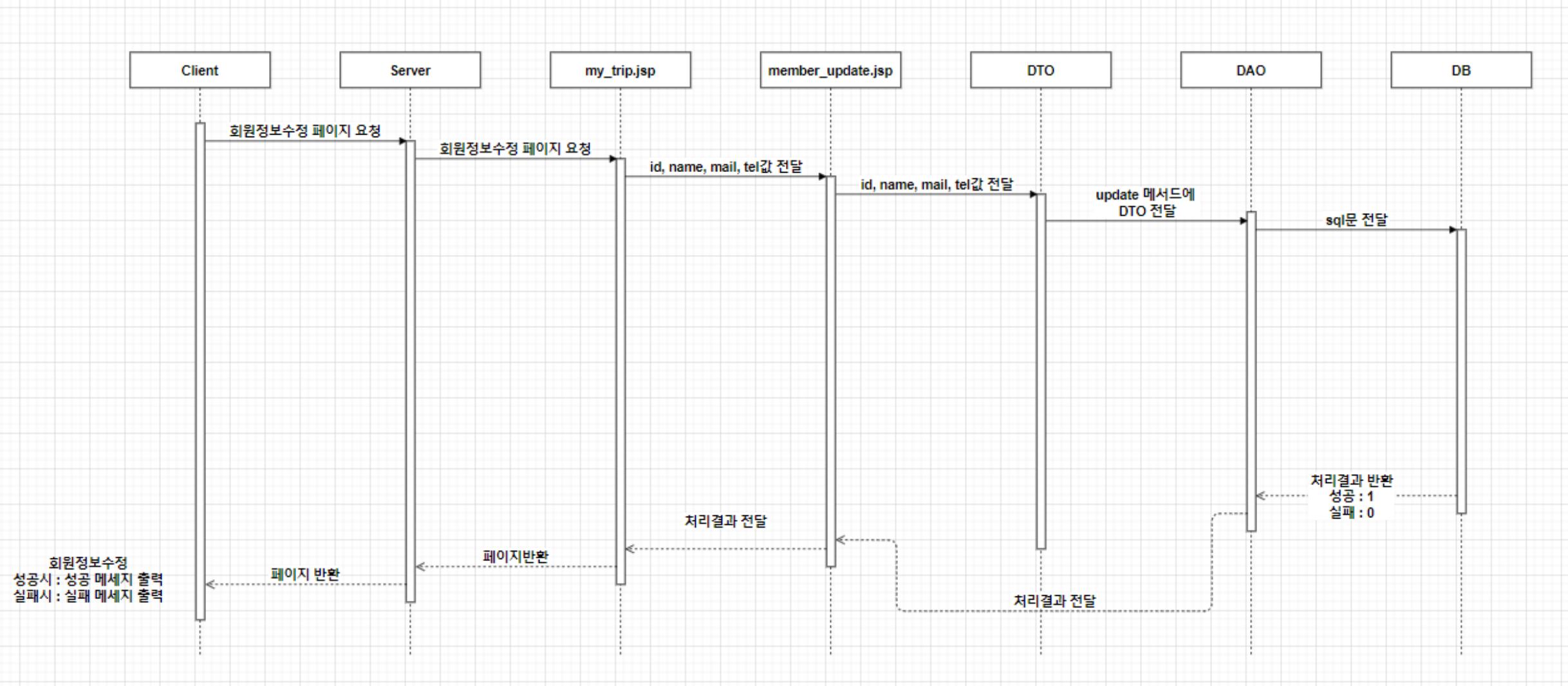
# 시퀀스 다이어그램 - 회원가입



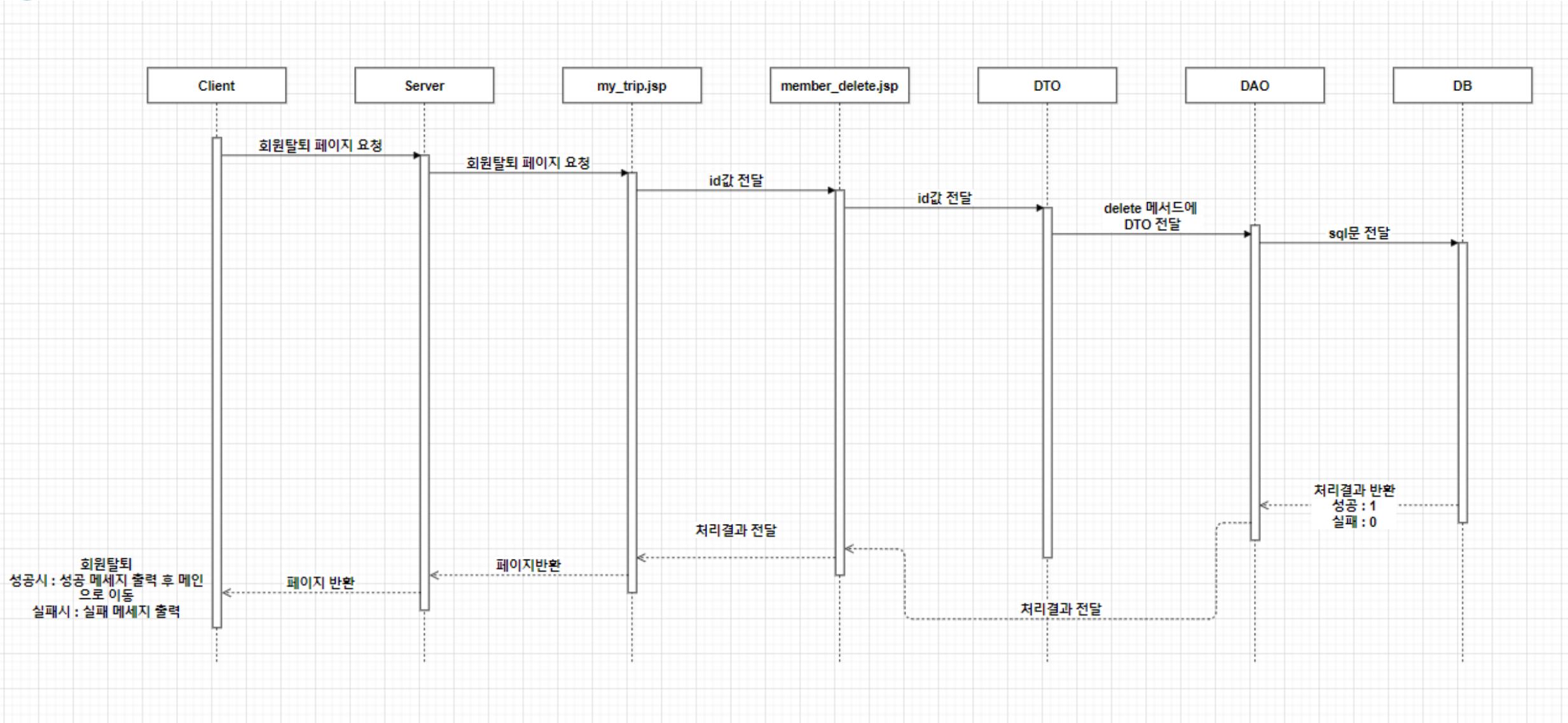
# 시퀀스 다이어그램 - 로그인



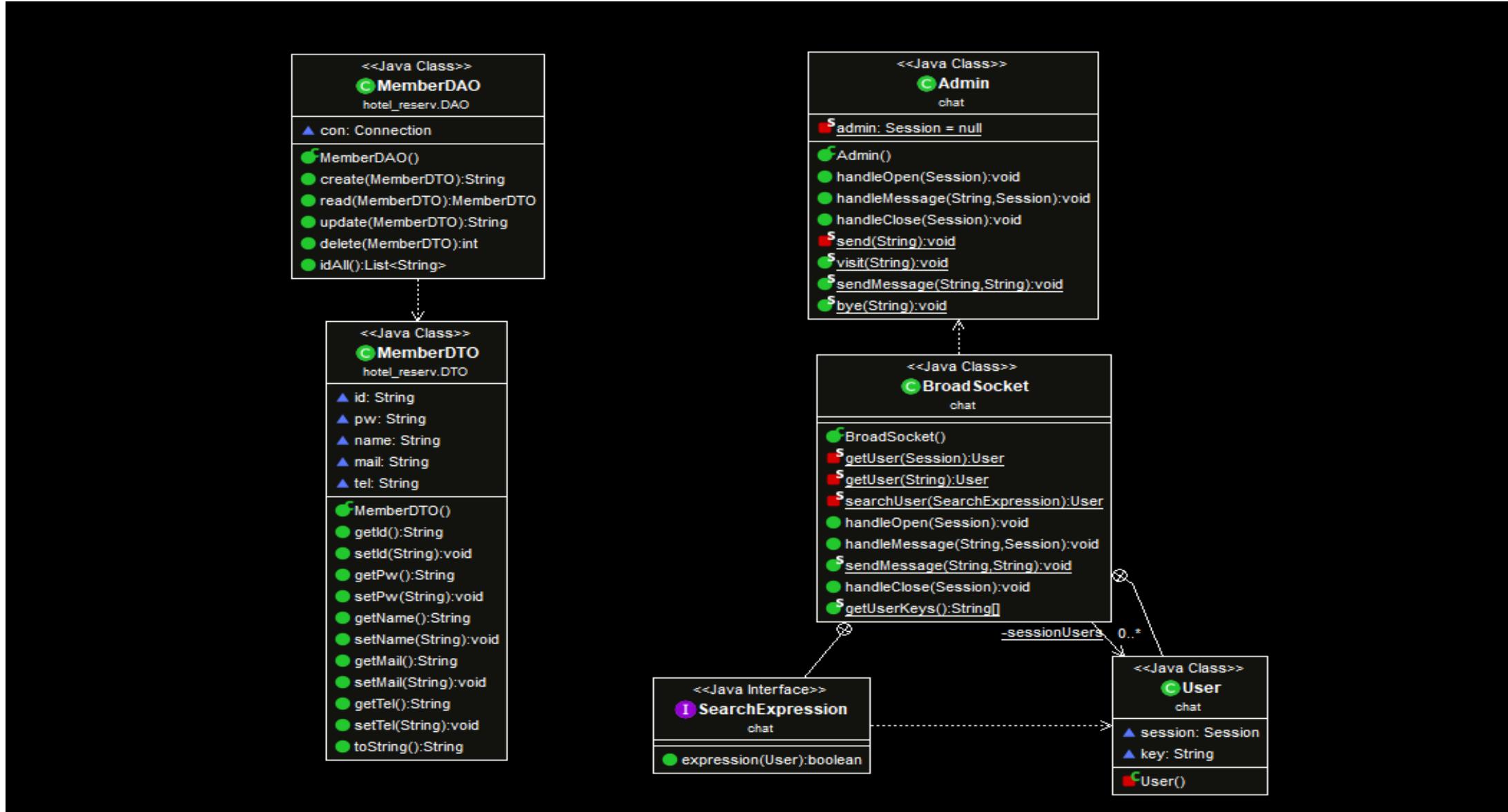
# 시퀀스 다이어그램 - 회원정보수정



# 시퀀스 다이어그램 - 회원탈퇴



# 시스템구성도



# Member table 제약조건

DB	table	coulum	type	NN	PK	comment
hotel_reserv	member	id	varchar(20)	O	O	아이디
		pw	varchar(20)	O		비밀번호
		name	varchar(20)	O		이름
		mail	varchar(100)	O		메일
		tel	varchar(20)	O		전화번호

ABC m_id	1	varchar(20)	[v]	[ ]
ABC pw	2	varchar(20)	[v]	[ ]
ABC name	3	varchar(20)	[v]	[ ]
ABC mail	4	varchar(100)	[v]	[ ]
ABC tel	5	varchar(20)	[v]	[ ]

# 메인 페이지(1)

```

<body>
<div id="total">
    <div id="menu_frame">
        <jsp:include page="menu.jsp"></jsp:include>
    </div>
    <hr>
    <div id="notice_frame">
        notice. 고객님의 여정이 코로나19(COVID-19)발생 상황에 영향을 받을 수 있으니 예약시 코로나 현황을 꼭 확인해주세요.
    </div>
    <div id="main_frame">
        각자 작업
    </div>
</div>
</body>

```



```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<ul>
    <!-- 상단 로고 및 메뉴 -->
    <li class="menu" style="margin: 30px 0 0 -50px;"><a href="main.jsp" id="logo">JEJU Hotel <span style="color:#C29D79;">Booking</span></a></li>
    <li class="menu" style="margin: 50px 30px 0 100px;"><a href="login.jsp">Hotel</a></li>
    <li class="menu" style="margin: 50px 30px 0 100px;"><a href="login.jsp">Review</a></li>
    <li class="menu" style="margin: 10px 20px 20px 330px; font-size: 15px;"><a href="member_create.jsp">sign up</a></li>
    <li class="menu" style="margin: 10px 0 20px 0; font-size: 15px;"><a href="login.jsp">login</a></li>
</ul>

```

상단 로고 및 메뉴를 만들어 전체 프로젝트 기본 틀로 사용

# 회원가입 페이지(1)

회원가입

ID  중복확인

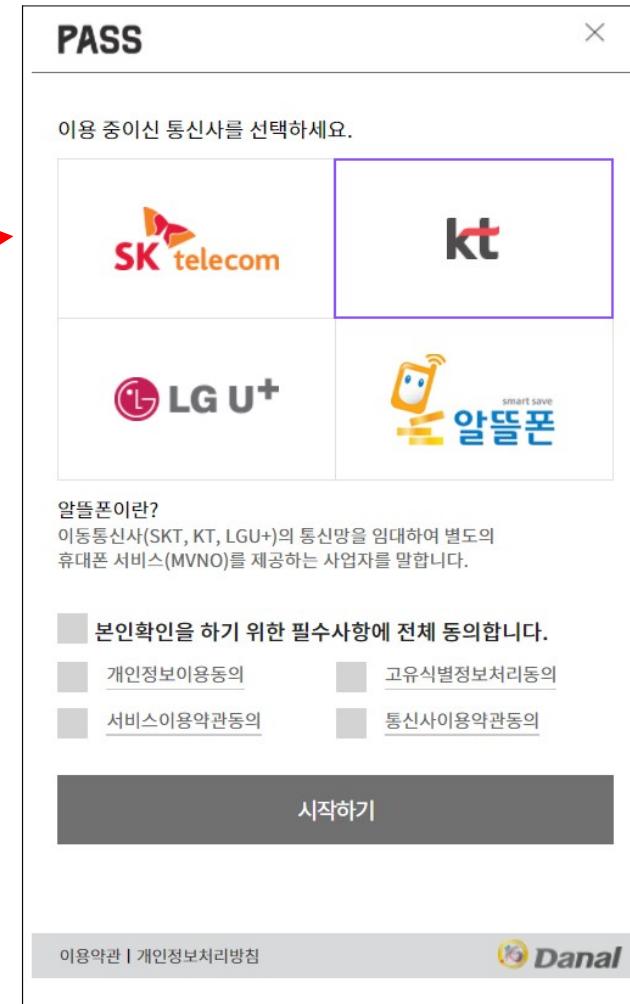
PW

이름

mail

tel

버튼 클릭시 페이지 이동



본인인증 후 성공여부 메세지  
출력 후 버튼 아래에 결과 표시



회원가입

본인인증 성공

ID  중복확인

PW

이름

mail

tel

## 회원가입 페이지(2)

&lt;%

```
String cerMsg = request.getParameter("cerMsg"); //본인인증 결과를 받아옴
if(cerMsg == null) cerMsg = ""; //본인인증을 하기 전 공백
```

%&gt;

3. 본인인증 페이지에서  
받아온 데이터 출력

```
$('#cer').click(function() {
    location.href='cer.jsp' /* 본인인증 버튼 클릭시 */
    /* cer.jsp페이지로 이동 */
})
```

1. 버튼 클릭 시  
본인인증 페이지로 이동

```
<button id='cer'>본인인증</button><br> <!-- 본인인증 버튼 --&gt;
&lt;div id="cerMsg" class="msg"&gt;&lt;%= cerMsg %&gt;&lt;/div&gt;&lt;br&gt;&lt;!-- 본인인증 버튼 클릭 후 리턴된 메세지를 출력 --&gt;</pre>

```

```
<script type="text/javascript" src="https://code.jquery.com/jquery-1.12.4.min.js" ></script>
<script type="text/javascript" src="https://cdn.iamport.kr/js/iamport.payment-1.1.5.js"></script>
<script>
document.cookie = "crossCookie=bar; SameSite=None; Secure";
let IMP = window.IMP;
IMP.init("iamport");
IMP.certification({
    merchant_uid: "ORD20180131-0000011"
}, function (rsp) {
    if (rsp.success) { /* 본인인증 성공시 */
        alert("인증에 성공하였습니다."); /* 성공메세지 출력 */
        location.href="member_create.jsp?cerMsg=본인인증 성공"/* cerMsg를 가지고 회원가입 화면으로 이동 */
    } else { /* 본인인증 실패시 */
        alert("인증에 실패하였습니다. 에러 내용: " + rsp.error_msg); /* 실패+에러 메세지 출력 */
        location.href="member_create.jsp?cerMsg='본인인증 실패'"/* cerMsg를 가지고 회원가입 화면으로 이동 */
    }
});
```

2. 본인인증 성공 여부에 따른 메세지 출력 후  
cerMsg값을 가지고 회원가입 페이지로 다시 이동

# 본인인증 페이지(1)

- ☒ ① Indicate whether a cookie is intended to be set in a cross-site context by specifying its SameSite attribute

Because a cookie's SameSite attribute was not set or is invalid, it defaults to SameSite=Lax which prevents the cookie from being set in a cross-site context. This behavior protects user data from accidentally leaking to third parties and cross-site request forgery.

Resolve this issue by updating the attributes of the cookie:

- Specify SameSite=None and Secure if the cookie is intended to be set in cross-site contexts. Note that only cookies sent over HTTPS may use the Secure attribute.
- Specify SameSite=Strict or SameSite=Lax if the cookie should not be set by cross-site requests.

## AFFECTED RESOURCES

▼ 1 cookie

Name	Domain & Path
AWSELB	service.iampoint.kr/

⊕ [Learn more: SameSite cookies explained](#)

## 크롬 쿠키보안 정책 - SameSite 속성의 기본값이 Lax

None : 동일 사이트와 크로스 사이트에 모두 쿠키 전송이 가능

Strict : 서로 다른 도메인에서는 아예 전송이 불가능

Lax : Strict 설정에 일부 예외 (HTTP get method / a href / link href)를 두어 적용되는 설정

## 본인인증 페이지(2)

### 1. 쿠키의 SameSite속성을 None으로 바꿈

```
document.cookie = "crossCookie=bar; SameSite=None; Secure";
```

본인인증 페이지 코드의 맨위에 해당 코드를 추가

### 2. Tomcat의 설정 변경

```
<CookieProcessor sameSiteCookies="none"/>
```

Tomcat폴더의 context.xml파일을 열어 마지막줄에 해당 코드를 추가

# 회원가입 페이지(3)

YouTube BIGDATA - Daum...

U Hotel Booking

localhost:8889 내용:  
사용 가능한 아이디입니다.

Hotel

확인

notice. 고객님의 여정이 코로나19(COVID-19)발생 상황에 영향을 받을 수 있으니 예약시 코로나19 예방수칙을 철저히 준수해주시기 바랍니다.

입  
본인인증

ID test999 중복확인

```

ID <input id='inputId' name = "id"> <!-- 아이디 입력칸 -->
<button id="idCheck">중복확인</button> <!-- 중복확인 버튼 -->
<div id="idMsg" class="msg"></div><br> <!-- 아이디 글자수 체크 후 메세지 출력 -->
```

```

$( '#idCheck' ).click( function() { /* 중복확인 버튼 클릭시 */
    idValue = $( '#inputId' ).val();
    $.ajax({
        url: "login_check2.jsp",
        data:{ id: idValue },
        success: function(result){
            alert(result);
        }
    })
})
```

버튼 클릭시  
중복확인  
페이지로  
넘겨서 결과  
반환

```

<% String id = request.getParameter("id");
MemberDTO dto = new MemberDTO();
dto.setId(id);
MemberDAO dao = new MemberDAO();
String result = dao.read2(dto);
%>
<%= result%>
```

DB에 입력받은  
아이디와 같은  
아이디가 있는지 확인

결과 반환받아  
출력

# 회원가입 페이지(4)

아이디 입력후 커서가 벗어났을때 글자수에 따라 메세지를 출력하는 함수 사용

```
ID <input id='inputId' name = "id"> <!-- 아이디 입력칸 -->
<button id="idCheck">중복확인</button> <!-- 중복확인 버튼 -->
<div id="idMsg" class="msg"></div><br> <!-- 아이디 글자수 체크 후 메세지 출력 -->
```

```
$('#inputId').blur(function() { /* 입력칸에서 커서가 벗어났을때 */
    idValue = $('#inputId').val(); /* 입력칸에 입력된 값을 변수에 저장 */
    idMsgTag = $('#idMsg');
    idMsgTag2 = $('#idMsg2');
    if(idValue.length < 2 || idValue.length > 20){ /* 입력값이 2글자 미만이거나 20글자 초과시 */
        idMsgTag.html('<font color=red>아이디는 2~20글자 이내로 입력해주세요.</font>'); /* 해당 메세지 출력 */
    } else{
        idMsgTag.html('<font color=green>입력가능한 아이디입니다.</font>');
    }
}

$('#inputPw').blur(function() {
    pwValue = $('#inputPw').val();
    pwMsgTag = $('#pwMsg');
    if(pwValue.length < 2 || pwValue.length > 20){
        pwMsgTag.html('<font color=red>비밀번호는 2~20글자 이내로 입력해주세요.</font>');
    } else{
        pwMsgTag.html('<font color=green>사용가능한 비밀번호입니다.</font>');
    }
}

$('#inputName').blur(function() {
    nameValue = $('#inputName').val();
    nameMsgTag = $('#nameMsg');
    if(nameValue.length > 20){
        nameMsgTag.html('<font color=red>이름은 20글자 이내로 입력해주세요.</font>');
    } else{
        nameMsgTag.html('');
    }
}

$('#inputMail').blur(function() {
    mailValue = $('#inputMail').val();
    mailMsgTag = $('#mailMsg');
    if(mailValue.length > 100){
        mailMsgTag.html('<font color=red>이메일은 100글자 이내로 입력해주세요.</font>');
    } else{
        mailMsgTag.html('');
    }
}

$('#inputTel').blur(function() {
    telValue = $('#inputTel').val();
    telMsgTag = $('#telMsg');
    if(telValue.length > 20){
        telMsgTag.html('<font color=red>전화번호는 20글자 이내로 입력해주세요.</font>');
    } else{
        telMsgTag.html('');
    }
})
```

PW <input id='inputPw' name = "pw"><br> <!-- 비밀번호 입력칸 -->
<div id="pwMsg" class="msg"></div><br> <!-- 비밀번호 글자수 체크 후 메세지 출력 -->

이름 <input id='inputName' name = "name"><br> <!-- 이름 입력칸 -->
<div id="nameMsg" class="msg"></div><br> <!-- 이름 글자수 체크 후 메세지 출력 -->

mail <input id='inputMail' name = "mail"><br> <!-- 메일 입력칸 -->
<div id="mailMsg" class="msg"></div><br> <!-- 메일 글자수 체크 후 메세지 출력 -->

tel <input id='inputTel' name = "tel"><br> <!-- 전화번호 입력칸 -->
<div id="telMsg" class="msg"></div><br> <!-- 전화번호 글자수 체크 후 메세지 출력 -->
<button id="create" >회원가입</button> <!-- 회원가입 버튼 -->

같은 방식으로 비밀번호, 이름, 메일, 전화번호의 글자수 체크 메세지 출력

ID	test999
입력가능한 아이디입니다.	
ID	test99999999999999999999
중복확인	
아이디는 2~20글자 이내로 입력해주세요.	

PW	1234
사용가능한 비밀번호입니다.	
이름	test999
mail	test999@naver.com
tel	010-1234-5678

PW	1234ddddddddd
비밀번호는 2~20글자 이내로 입력해주세요.	
이름	test999ddddddddd
이름은 20글자 이내로 입력해주세요.	
mail	test999@naver.comddd
메일은 100글자 이내로 입력해주세요.	
tel	010-1234-5678dddddd
전화번호는 20글자 이내로 입력해주세요.	

# 회원가입 페이지(5)

## 회원가입 버튼 클릭

모든 체크메세지들이 입력 가능일 경우  
 입력받은 값들을 ajax를 사용하여 회원가입처리  
 페이지로 넘겨 DB에 입력 후 성공시 메세지 출력 후  
 로그인 페이지로 이동

체크 메세지중 하나라도 입력불가가 있을 경우  
 경고메세지 출력

```
<%
    String id = request.getParameter("id");      //입력값 받아서 변수에 저장
    String pw = request.getParameter("pw");
    String name = request.getParameter("name");
    String mail = request.getParameter("mail");
    String tel = request.getParameter("tel");

    MemberDTO dto = new MemberDTO();           //DTO에 입력값 저장
    dto.setId(id);
    dto.setPw(pw);
    dto.setName(name);
    dto.setMail(mail);
    dto.setTel(tel);

    MemberDAO dao = new MemberDAO();
    String result = dao.create(dto);           //DAO의 create메서드를 사용하여 DB에 입력
%>
<%= result%> |
```

```
$('#create').click(function() {      /* 회원가입 버튼 클릭시 */
    idValue = $('#inputId').val();    /* 입력칸에 입력받은 값을 변수에 저장 */
    pwValue = $('#inputPw').val();
    nameValue = $('#inputName').val();
    mailValue = $('#inputMail').val();
    telValue = $('#inputTel').val();

    cerMsg = $('#cerMsg').text();     /* 모든 체크메세지를 변수에 저장 */
    idMsg = $('#idMsg').text();
    idMsg2 = $('#idMsg2').text();
    pwMsg = $('#pwMsg').text();
    nameMsg = $('#nameMsg').text();
    mailMsg = $('#mailMsg').text();
    telMsg = $('#telMsg').text();

    if(idMsg == '입력가능한 아이디입니다.' && /* 모든 체크메세지들이 입력 가능일 경우 */
        cerMsg == '본인인증 성공' &&
        pwMsg == '사용가능한 비밀번호입니다.' &&
        nameMsg == "" &&
        mailMsg == "" &&
        telMsg == "") {
        $.ajax({                                /* 입력받은 데이터들을 ajax를 사용하여 넘긴다 */
            url: "member_create_result2.jsp",
            data: {
                id: idValue,
                pw: pwValue,
                name: nameValue,
                mail: mailValue,
                tel: telValue
            },
            success: function(result){
                alert(result)                      /* 처리 결과 출력 */
                if (result == "회원가입성공"){ /* 성공시 로그인 페이지로 이동 */
                    location.href="login.jsp";
                }
            }
        })
    } else{                                /* 체크메세지중 입력불가한 항목이 하나라도 있을경우 메세지 출력 */
        alert('가입조건이 맞지 않습니다.');
    }
})
```

# 로그인 페이지

```
<h3>로그인</h3>
<div style="margin: 50px;">
```

```
<!-- 품액션을 사용해 버튼 클릭시 입력칸에 입력한 정보들을 로그인 결과처리 페이지로 넘긴다. -->
<form action="login_result.jsp">
<table>
<tr>
    <td>아이디</td>
    <td><input name='id'></td>
</tr>
<tr>
    <td>비밀번호</td>
    <td><input name='pw'></td>
</tr>
</table>
<!-- 로그인 실패시 메세지 띄우는 부분 -->
<div id="errMsg" style="color:red"><%= errMsg %></div><br>
<button >로그인</button>
</form>
</div>
```

```
<%
//회원가입 실패시 세션에 저장된 에러메세지를 변수에 저장
String errMsg = (String)session.getAttribute("errMsg_login");

if(errMsg == null) errMsg = ""; //에러메세지가 없을 경우 빈칸

session.invalidate(); //사용 후 세션을 닫아준다.
%>
```

로그인



로그인 정보가 올바르지 않습니다.

로그인




입력된 값을 품액션으로 로그인처리  
페이지로 넘긴 후  
로그인 성공시 세션에 데이터 저장 후  
메인으로 이동

```
<%
String id = request.getParameter("id");      /* 넘겨준 입력값들을 변수에 저장 */
String pw = request.getParameter("pw");

MemberDTO dto = new MemberDTO();      /* DTO에 입력값들 저장 */
dto.setId(id);
dto.setPw(pw);

MemberDAO dao = new MemberDAO();
/* DAO의 read메서드를 사용해 입력받은 아이디에 해당하는 row를 검색하여 DTO에 넣어 가져온다*/
MemberDTO dto2 = dao.read(dto);

if(pw.equals(dto2.getPw())){      /* DB에서 가져온 패스워드와 입력받은 패스워드가 동일할 경우 */

    /* 호텔 예약에 필요한 데이터들을 로그인이 되어있는 동안 사용할 수 있도록 세션에 저장한다. */
    session.setAttribute("id",dto2.getId());
    session.setAttribute("name",dto2.getName());
    session.setAttribute("mail",dto2.getMail());
    session.setAttribute("tel",dto2.getTel());
    response.sendRedirect("main_login.jsp");      /* 로그인된 메인페이지로 이동 */

} else{      /* DB에서 가져온 패스워드와 입력받은 패스워드가 동일하지 않을 경우 */
    /* 에러메세지를 세션에 저장하고 로그인페이지로 다시 이동 */
    session.setAttribute("errMsg_login", "로그인 정보가 올바르지 않습니다.");
    response.sendRedirect("login.jsp");
}
%>
```

로그인 실패시 세션에 에러메세지 저장  
후 로그인 화면으로 다시 이동

# 로그아웃

The diagram illustrates the login and logout process for a JEJU Hotel Booking application.

**로그인 시 (Login Process):**

- 세션에 회원정보가 저장된 상태
- 로그인 전용 메뉴
- 모든 기능 사용 가능

**로그아웃 시 (Logout Process):**

세션을 닫고 메인으로 이동

```
<%>
session.invalidate();          /* 세션닫고 */
response.sendRedirect("main.jsp"); /* 메인으로 이동 */
%>
```

# 회원정보 페이지(1)

```
<%
String id = (String)session.getAttribute("id"); //로그인 되어있는 아이디

MemberDTO dto = new MemberDTO();
dto.setId(id);

MemberDAO dao = new MemberDAO();
MemberDTO dto2 = dao.read(dto); //DB에서 해당 아이디의 데이터 read

String msg = request.getParameter("msg");
%>
```

```
<button id='updateB' class='info_button'>수정하기</button>
<button id='deleteB' class='info_button'>회원탈퇴</button>
```

```
$( '#updateB' ).click(function() { /* 버튼 클릭시 */
    pwTag = $('#inputPw') /* 입력값 변수에 저장 */
    mailTag = $('#inputMail')
    telTag = $('#inputTel')

    pwValue = pwTag.val();
    mailValue = mailTag.val();
    telValue = telTag.val();
    check = confirm("수정하시겠습니까?") /* 제크메세지 띄워서 체크 */
    if(check){ /* 확인시 ajax로 수정처리페이지에 값을 넘김 */
        $.ajax({
            url: "member_update2.jsp",
            data:{
                pw: pwValue,
                mail: mailValue,
                tel: telValue
            },
            success: function(result){
                if(result == "수정성공"){ /* 결과에 따른 메세지 출력 */
                    alert("수정되었습니다.")
                }else{
                    alert("수정실패하였습니다.")
                }
            }
        })
    }
})
```

페이지 접속시 세션에  
저장되어있는 아이디로 DB  
에서 회원정보 read

수정하기 버튼 클릭시  
처리 페이지로 데이터 전달  
반환된 결과에 따라 메세지 출력

```
<%
/* 입력받지 않는 id, name은 세션에서 받아와서 변수에 저장 */
String id = (String)session.getAttribute("id");
String name = (String)session.getAttribute("name");

/* 입력받은 데이터들은 변수에 저장 */
String pw = request.getParameter("pw");
String mail = request.getParameter("mail");
String tel = request.getParameter("tel");

/* DTO에 데이터 저장 */
MemberDTO dto = new MemberDTO();
dto.setId(id);
dto.setPw(pw);
dto.setMail(mail);
dto.setTel(tel);

/* DAO의 update메서드를 사용해 DB에 저장된 값을 수정 */
MemberDAO dao = new MemberDAO();
String result = dao.update(dto);
%>
```

# 회원정보 페이지(2)

```
<button id='updateB' class='info_button'>수정하기</button>
<button id='deleteB' class='info_button'>회원탈퇴</button>
```

```
$('#deleteB').click(function() { /* 탈퇴 버튼 클릭시 */
    check = confirm("정말 탈퇴하시겠습니까?") /* 메세지 띄워서 확인 */
    if(check){ /* 탈퇴처리 페이지로 이동 */
        location.href="member_delete.jsp"
    }
})
```

```
<%
/* 로그인되어있는 아이디 변수에 저장*/
String id = (String)session.getAttribute("id");

MemberDTO dto = new MemberDTO();
dto.setId(id);
/* DAO의 delete메서드를 사용해 DB데이터 삭제 */
MemberDAO dao = new MemberDAO();
int result = dao.delete(dto);
/* 삭제 결과를 세션에 저장 후 페이지 이동 */
if (result == 1){
    response.sendRedirect("main.jsp?msg=true");
} else{
    response.sendRedirect("my_trip.jsp?msg=false");
}
%>
```

탈퇴하기 버튼 클릭시  
처리 페이지로 데이터 전달

처리결과에 따라 msg값을 갖고 페이지이동

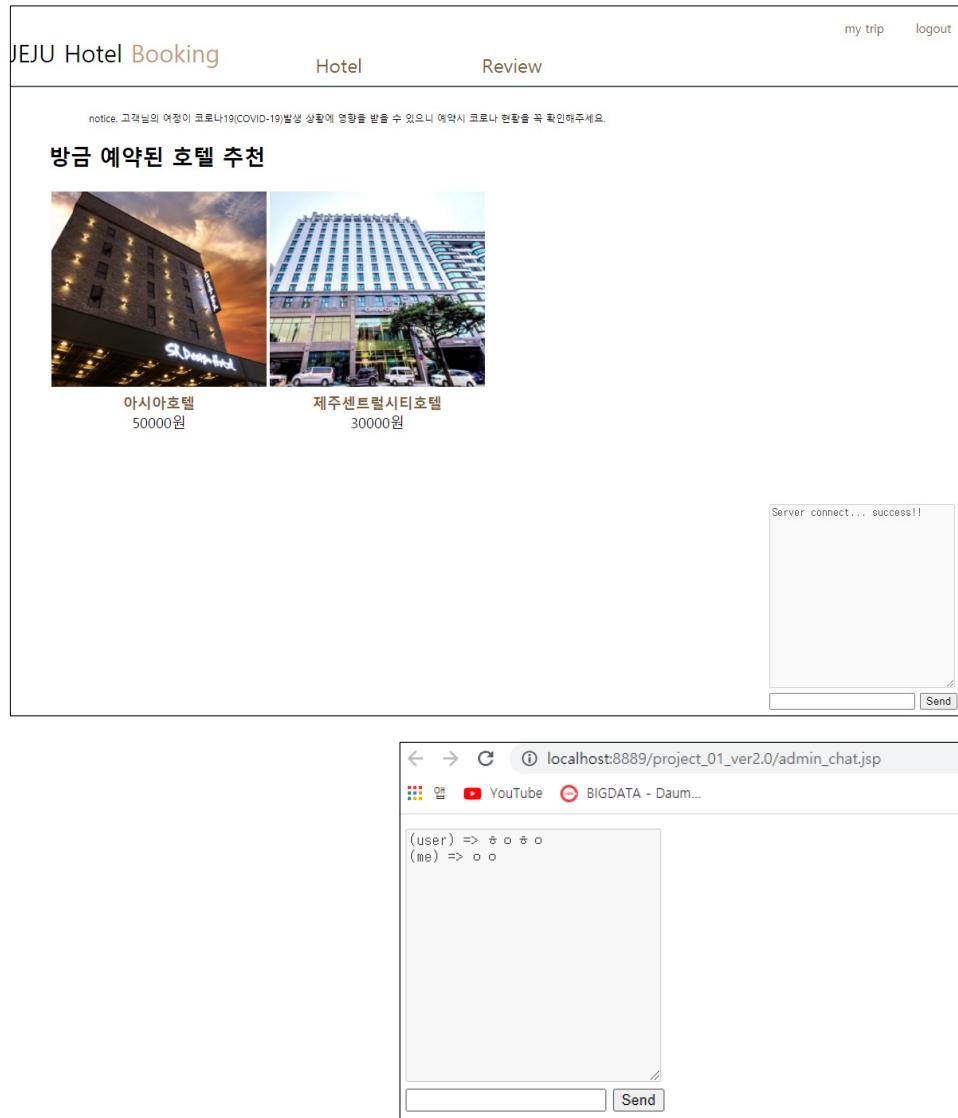
```
<%
String msg = request.getParameter("msg");
%>
<script type="text/javascript">
<% if (msg != null){ %>
alert('탈퇴 성공!');
location.href="main.jsp"
<% } %>
</script>
```

메세지 출력 후 msg값을 빼고 페이지이동

localhost:8889 내용:

탈퇴 성공!

확인



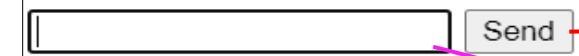
## 새로운 로그인이 있을때 관리자페이지에서 채팅창 자동 추가



```
Server connect... success!!
(test14) => ㅎㅇ ㅎㅇ
(admin) => ㅇㅇ
```

입력칸에서 엔터를 치거나  
send버튼을 누를 경우

채팅창에 입력값 출력 후  
서버로 전송



```
<div id="chat">
    <!-- 서버와 주고받는 메시지가 보이는 부분 -->
    <textarea id="messageTextArea" rows="15" cols="30" disabled="disabled"></textarea>
    <form>
        <!-- 송신 메시지를 작성하는 텍스트 박스 -->
        <input id="textMessage" type="text" onkeydown="return enter()">
        <!-- 메시지 송신을 하는 버튼 -->
        <input onclick="sendMessage()" value="Send" type="button">
    </form><br />
</div>
```

```
/* 서버의 broadsocket의 서블릿으로 웹 소켓을 한다. */
var webSocket = new WebSocket("ws://localhost:8889/chat/broadsocket");

var messageTextArea = document.getElementById("messageTextArea");

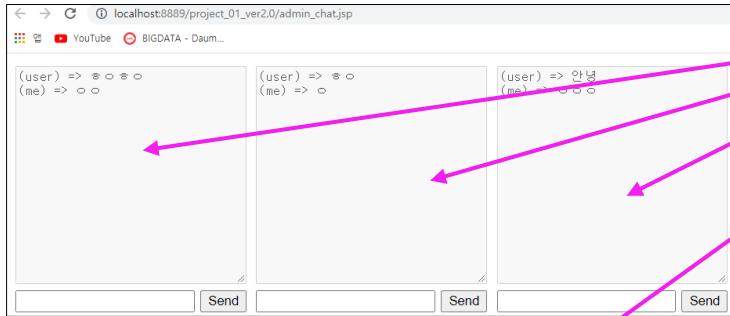
/* 접속시 메시지가 보이는 부분에 문구 출력 */
webSocket.onopen = function(message) {
    messageTextArea.value += "Server connect... success!!\n";
};

/* 에러가 발생했을때 문구 출력 */
webSocket.onerror = function(message) {
    messageTextArea.value += "error...\n";
};

/* 서버로부터 메시지가 도착하면 출력 */
webSocket.onmessage = function(message) {
    messageTextArea.value += "(admin) => " + message.data + "\n";
};

/* 서버로 메시지를 발송하는 함수
Send 버튼을 누르거나 텍스트 박스에서 엔터를 치면 실행 */
function sendMessage() {
    let message = document.getElementById("textMessage");
    /* 입력한 값을 채팅창에 출력 */
    messageTextArea.value += "<%= id%> => " + message.value + "\n";
    /* 입력한 값을 소켓으로 전송 */
    webSocket.send(message.value);
    /* 입력칸 초기화 */
    message.value = "";
}

function enter() {
    /* 엔터를 누르면 (keyCode 13 == 엔터)*/
    if(event.keyCode === 13) {
        /* 함수 실행 */
        sendMessage();
        /* form에 의해 자동 submit을 막는다. */
        return false;
    }
    return true;
}
```



```
//서버의 admin의 서블릿으로 웹 소켓을 한다.
var webSocket = new WebSocket("ws://localhost:8889/chat/admin");
webSocket.onopen = function(message) { };
webSocket.onclose = function(message) { };
webSocket.onerror = function(message) { };
//서버로부터 메시지가 오면
webSocket.onmessage = function(message) {
    //제이슨 형태의 메시지
    let node = JSON.parse(message.data);

    //유저가 접속하면
    if(node.status === "visit") {
        let form = $(".template").html();
        //속성 data-key에 unique키를 넣고 body에 template을 추가
        form = $("<div class='float-left'></div>").attr("data-key",node.key).append(form);
        $("body").append(form);

        //유저가 메시지를 보내면
    } else if(node.status === "message") {
        //key로 해당 div영역을 찾고
        let $div = $("[data-key='"+node.key+"']");
        //console영역을 찾아서
        let log = $div.find(".console").val();
        //메시지를 추가
        $div.find(".console").val(log + "(user) => " +node.message + "\n");

        //유저가 접속을 끊으면
    } else if(node.status === "bye") {
        //해당 키로 div를 찾아서 제거
        $("[data-key='"+node.key+"']").remove();
    }
};
```

```
<div class="template" style="display:none">
    <!-- 유저와 주고받는 메시지가 보이는 부분 -->
    <textarea rows="15" cols="30" class="console" disabled="disabled"></textarea>
    <form>
        <!-- 메시지를 작성하는 텍스트 박스 -->
        <input type="text" class="message" onkeydown="if(event.keyCode === 13) return false;">
        <!-- 메시지 송신을 하는 버튼 -->
        <input value="Send" type="button" class="sendBtn">
    </form><br />
</div>
```

새로운 브라우저에서 유저 로그인 시 채팅창 생성

유저가 메인페이지를 떠나면 채팅창 닫힘

```
//전송 버튼을 클릭하면
$(document).on("click", ".sendBtn", function(){
    let $div = $(this).closest(".float-left");
    let message = $div.find(".message").val();
    let key = $div.data("key");
    let log = $div.find(".console").val();
    //채팅창에 메시지 추가
    $div.find(".console").val(log + "(me) => " + message + "\n");
    $div.find(".message").val("");

    //웹소켓으로 메시지전송
    webSocket.send(key+"#####"+message);
});

//엔터키를 누르면 버튼을 클릭하는 트리거를 발생
$(document).on("keydown", ".message", function(){
    if(event.keyCode === 13) {
        $(this).closest(".float-left").find(".sendBtn").trigger("click");
        // form에 의해 자동 submit을 막는다.
        return false;
    }
    return true;
});
```