

2020.05.22

화면 설계

백한별

[게시판]

회원을 기반으로 한 게시판의 화면설계에 대한 내용입니다.

목차

프로젝트 개요.....	2
프로젝트 설명	
기능정의서	3
사용환경	
개발환경	4
개발기간	4
화면 설계	
UI 스타일 가이드	5
UI 흐름도 / 유즈케이스	6
프로세스 정의	8
화면 설계도	9
데이터 베이스	
회원관리 테이블	20
게시판관리 테이블	21
기능 설계	
MVC 모델	22
파일 구조	23
인터페이스 정의서	
회원관리 기능	25
게시판관리 기능	47
화면 관리 및 파일 다운로드 소스	64
배포환경	68
Contact Us	69

[프로젝트 개요]

회원을 기반으로 한 게시판 시스템을 JSP를 사용하여 Web으로 구현하였다.

- Oracle을 통해 회원과 게시글관련 DB Table 생성
- 페이지간의 이동은 주로 script, form 태그의 action, input 태그의 onclick으로 이루어진다.
- 일반회원은 회원정보보기, 회원정보수정, 회원탈퇴, 글쓰기, 본인글 수정, 답글쓰기, 본인글 삭제 기능들을 사용 가능하다.
- 관리자는 'admin'이라는 별도의 계정을 이용해 회원들의 정보보기, 삭제, 글쓰기, 본인글 수정, 답글쓰기, 글삭제 기능들을 사용 가능하다.
- 회원가입, 회원정보수정, 회원탈퇴, 글쓰기, 글수정, 글삭제 기능을 이용하면 해당 DB Table에 삽입, 수정, 삭제된다.
- 회원정보수정은 회원관련 Table과 게시글관련 Table 모두 수정된다.

[프로그램 설명]

1. 기능정의서

1.1 회원관리		
No	기능	설명
R-M-001	로그인	Id와 password를 입력 받아 회원Table에서 맞는지 확인
R-M-002	회원가입	Id, password, 생년월일, email, address, 전화번호, name을 입력 받아 가입한 날짜와 함께 회원Table에 삽입
R-M-003	정보확인	일반회원 - 로그인한 id를 기반으로 회원Table에서 해당 회원의 정보만을 확인 관리자 - 'admin' 계정으로 로그인하여 회원Table에서 가입된 모든 회원의 정보를 확인
R-M-004	정보수정	비밀번호 확인 후 로그인한 id를 기반으로 id와 가입날짜를 제외한 변경할 정보를 입력해 회원Table과 게시글Table을 수정
R-M-005	탈퇴	일반회원 - 비밀번호 확인 후 id와 정보를 회원Table에서 삭제 관리자 - 'admin' 계정으로 로그인하여 회원을 선택해 회원Table에서 삭제

1.2 게시판 관리		
No	기능	설명
R-B-001	글목록	게시판Table에서 등록된 게시글 10개단위로 불러와 표시
R-B-002	글쓰기	게시판Table의 ref, re_step, re_level 필드를 기반으로 새글과 답글을 구분하여 입력한 제목과 내용, 파일과 로그인한 id의 회원Table에 있는 이름, 이메일, 비밀번호를 게시판 Table에 저장
R-B-003	글수정	글에 등록한 회원 아이디와 비밀번호를 확인하여 제목, 내용, 파일을 게시판Table에서 수정
R-B-004	글삭제	일반회원 - 글을 등록한 회원의 아이디와 비밀번호를 확인하여 게시판Table에서 글을 삭제 관리자 - 아이디와 비밀번호 'admin' 일치 여부 확인하여 글 삭제

2. 사용환경

A. 개발환경

- i. Apache tomcat v8.5
- ii. IDE : Eclipse EE 4.14.0
- iii. DBMS : Oracle Developer 18.4.0.376
Oracle Express Edition 11.2.0.2.0
- iv. JDK : jdk 1.8.0_212
- v. cos.jar (파일 업로드, 다운로드 라이브러리)

3. 개발 기간

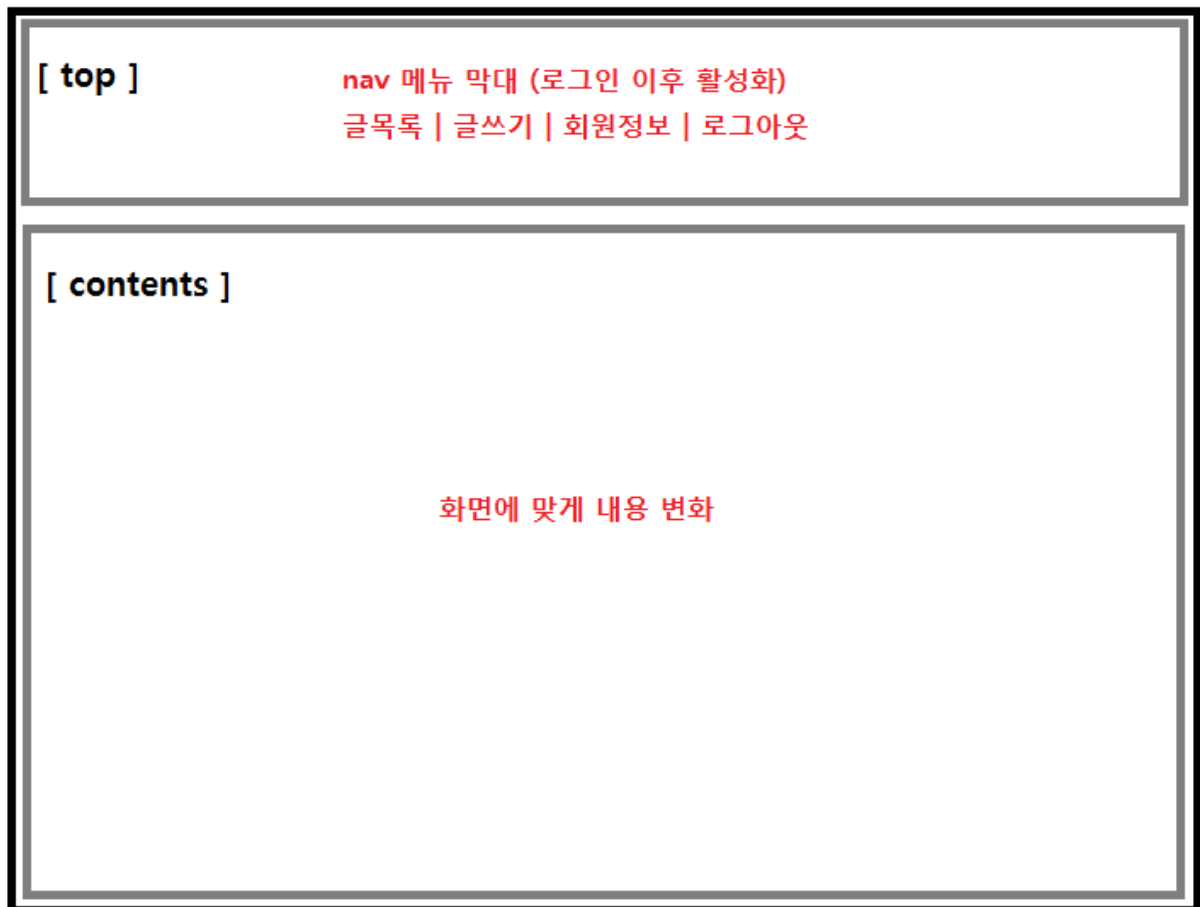
2020.05.15 ~ 2020.05.22

[화면 설계]

1. UI 스타일 가이드

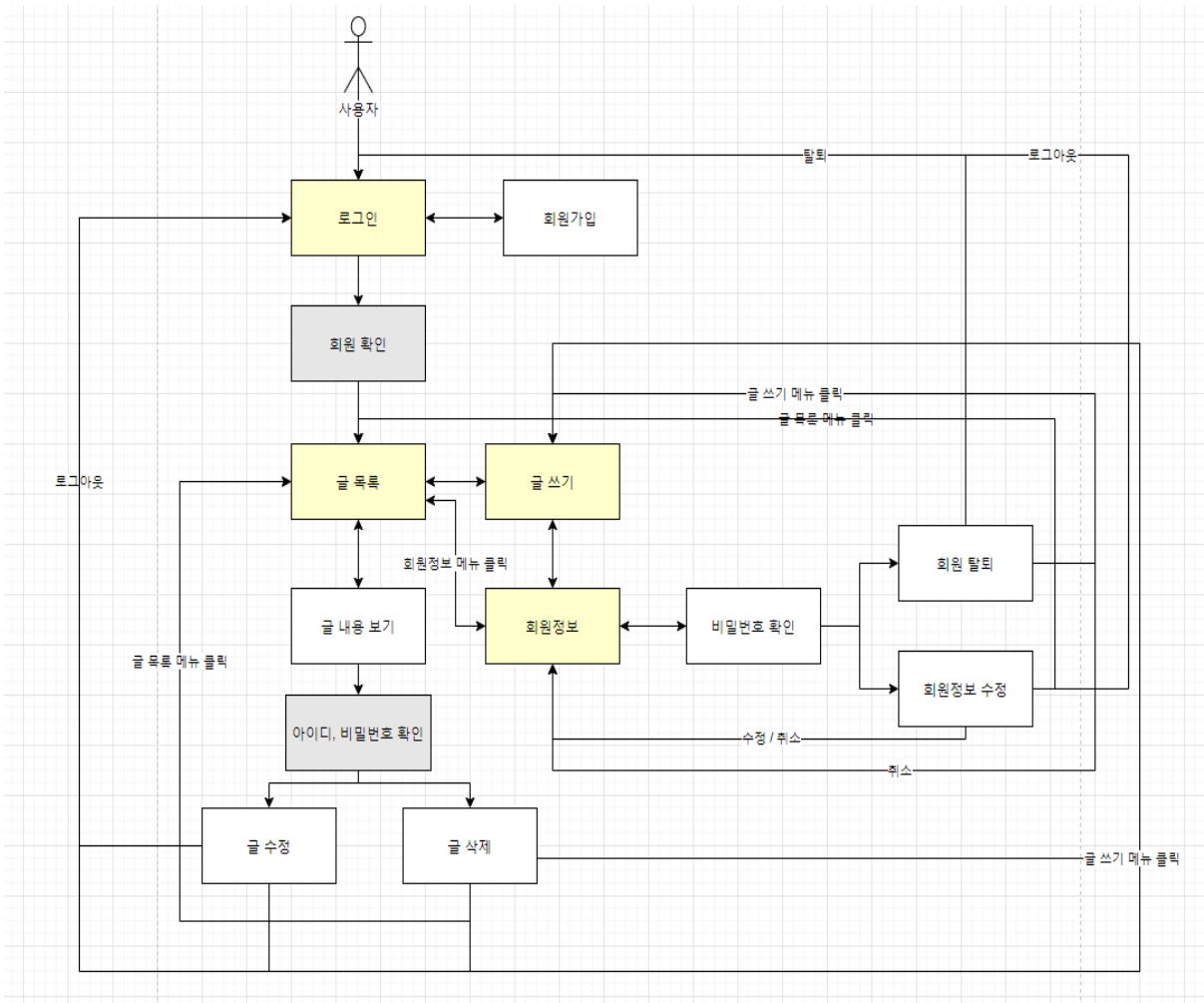
A. 해상도 : 110 x 119 이상 최적화

B. Frame 구분 : top, Contents

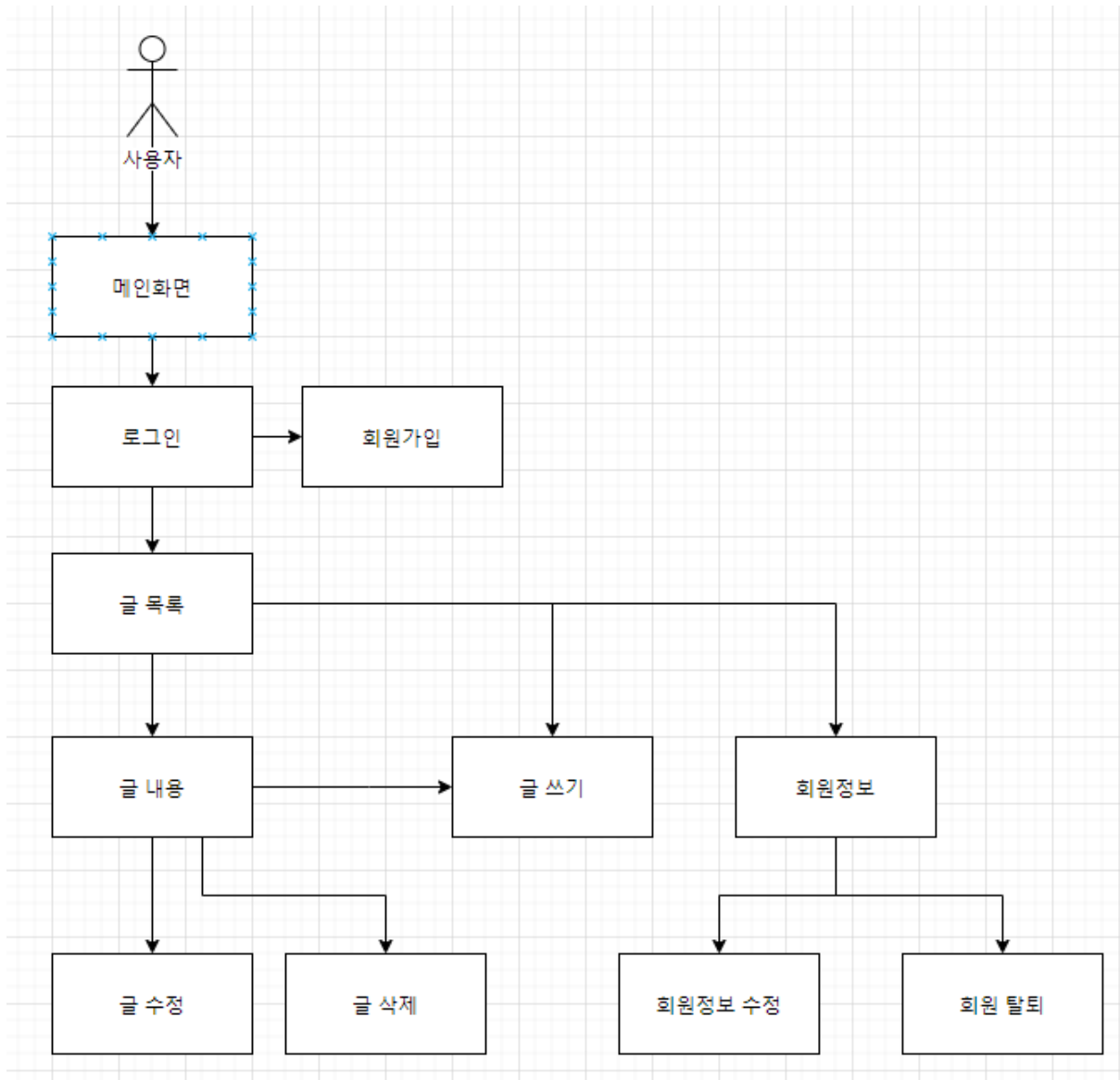


2. UI 흐름도 / 유즈케이스

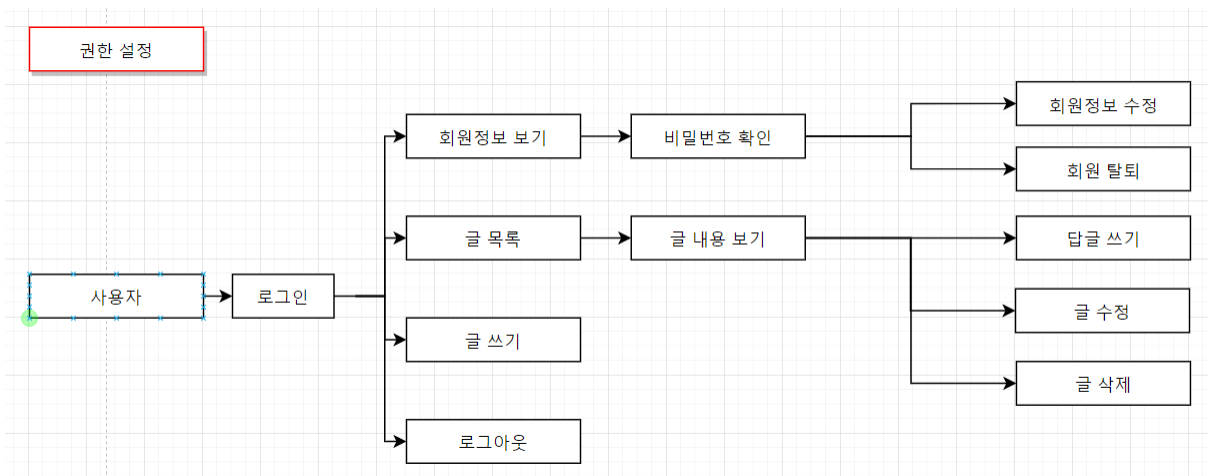
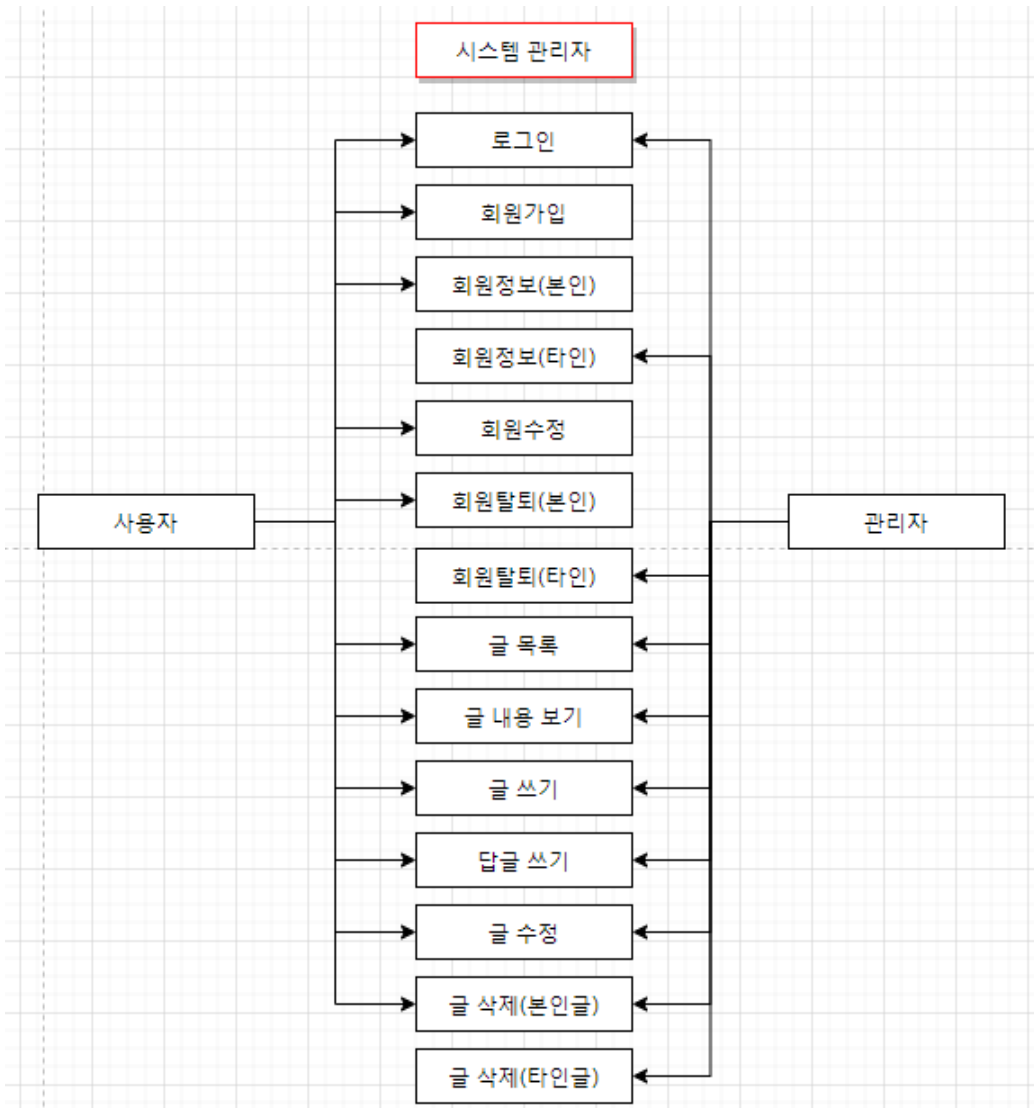
A. UI 흐름도



B. 유즈케이스



3. 프로세스 정의



4. 화면 설계도

A. 로그인

← → ↻ ⓘ localhost:8090/ch13/login.du ☆ BXX | 안병 ⋮

LOGIN

아이디 :

비밀번호 :

1-1 1-2

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Login_Form

Description	
1-1	Post_List
1-2	Sign_Up_Form

B. 회원가입

The screenshot shows a web browser window with the address bar displaying 'localhost:8090/ch13/joinForm.du'. The main content area has a yellow background and contains a 'SIGN UP' form. The form is a vertical stack of input fields with labels in Korean: '아이디' (ID), '비밀번호' (Password), '생년월일' (Date of Birth), '이메일' (Email), '주소' (Address), '전화번호' (Phone Number), and '이름' (Name). The '생년월일' field includes a date picker icon. At the bottom of the form is a red 'sign up' button. Below the button, the text '2-1' is visible.

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Sing_Up_Form

Description	
2-1	Login_Form

C. 글 목록

← → ↻ localhost:8090/ch13/loginPro.du

3-1 3-2 3-3 3-4

글목록 | **글쓰기** | **회원정보** | **로그아웃**

글목록(전체 글:33)

번호	3-5 제목	3-6 작성자	작성일	조회
33	헛드러오	큰집사	2020-05-15 14:48	37
32	비오는 날은	찌질달	2020-05-15 11:41	34
31	아닙니다.	평권	2020-05-15 09:59	10
30	오늘의 날씨는	짱구	2020-05-15 09:46	9
29	오악새가 울기 시작했습니다.	오악새	2020-05-14 16:35	23
28	오악새가 울었습니다.	오악새	2020-05-14 11:39	8
27	주인님	큰집사	2020-05-13 11:22	27
26	... 캔따개는	작은집사	2020-05-14 11:26	7
25	... 사돈남말	큰집사	2020-05-14 11:41	2
24	나는	연근	2020-02-20 11:57	4

[1] [2] [3] [4]

3-7

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Post_List

Description	
3-1	Post_List
3-2	Write_Form
3-3	Member_Main
3-4	Login_Form
3-5	Content
3-6	Microsoft Outlook 연결
3-7	Post_List Page 숫자

D. 글 쓰기

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Wirte_Form

Description	
4-1	Post_List
4-2	Write_Form
4-3	Member_Main
4-4	Login_Form
4-5	파일 선택 창
4-6	Post_List
4-7	Write_Form
4-8	Post_List

E. 회원정보

← → ↻ localhost:8090/ch13/main.du ☆ 1.2.3.4 1.2.3.4

5-1 5-2 5-3 5-4

글목록 **글쓰기** **회원정보** **로그아웃**

aaa님의 정보

아이디	aaa
생년월일	2020-05-02
이메일	aaa@sw.com
주소	지구
전화번호	01012341234
이름	aaa
가입일자	2020-05-21

5-5 5-6

정보수정 **회원탈퇴**

← → ↻ localhost:8090/ch13/main.du ☆ 1.2.3.4 1.2.3.4

글목록 | 글쓰기 | **회원정보** | 로그아웃

관리자님, 어서오세요.

회원목록

5-7

Description	
5-1	Post_List
5-2	Write_Form
5-3	Member_Main
5-4	Login_Form
5-5	Check_Pwd
5-6	Check_Pwd
5-7	Member_List (관리자 계정)

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Member_Main

F. 비밀번호 확인

비밀번호 확인

비밀번호를 입력해 주세요.

비밀번호 :

6-1 6-2

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Check_Pwd

Description	
6-1	정보수정 - Edit_Member 회원탈퇴 - Login_Form
6-2	Member_Main

G. 회원정보 수정

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Edit_Member

Description	
7-1	Post_List
7-2	Write_Form
7-3	Member_Main
7-4	Login_Form
7-5	Member_Main
7-6	Member_Main

H. 글 수정

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Edit_Post_Form

Description	
8-1	Post_List
8-2	Write_Form
8-3	Member_Main
8-4	Login_Form
8-5	파일 선택 창
8-6	Post_List
8-7	Edit_Post_Form
8-8	Post_List

I. 글 내용 보기

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Content

Description	
9-1	Post_List
9-2	Write_Form
9-3	Member_Main
9-4	Login_Form
9-5	파일 다운로드 창
9-6	Edit_Post_Form
9-7	Del_Post_Form
9-8	Write_Form

J. 글 삭제

글 삭제

비밀번호를 입력해 주세요.

비밀번호 :

글 삭제 **글 목록**

10-1 10-2

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Del_Post_Form

Description	
9-1	Post_List
9-2	Post_List

K. 회원 목록

← → ↻ localhost:8090/ch13/list.du ☆ 🔒 안됨

10-1 글목록

10-2 글쓰기

10-3 회원정보

10-4 로그아웃

회원목록(전체 회원 수:7)

번호	아이디	비밀번호	생년월일	이메일	주소	전화번호	이름	가입일자	삭제
7	aaa	aaa	2020-05-22	aaa@sw.com	지구	01012341234	aaa	2020-05-22	삭제
6	admin	admin	null	null	null	null	관리자	2020-05-20	삭제
5	연근	aaa	null	null	null	null	연근	2020-05-19	삭제
4	짱구	aaa	null	null	null	null	짱구	2020-05-19	삭제
3	찌질달	aaa	null	null	null	null	찌질달	2020-05-19	삭제
2	큰집사	aaa	null	null	null	null	큰집사	2020-05-19	삭제
1	랭권	aaa	null	null	null	null	랭권	2020-05-19	삭제

[1]

시스템 정보	
시스템명	회원제 게시판
작성일	2020.05.22
작성자	백한별
화면 ID	Member_List

Description	
10-1	Post_List
10-2	Write_Form
10-3	Member_Main
10-4	Login_Form
10-5	Member_List
10-6	Member_List

[데이터 베이스]

1. 회원관리 테이블

Member	
속성	주요업무
Id	회원의 id
Passwd	회원의 password
Date_number	회원의 생년월일
Email	회원의 email
Address	회원의 주소
Tel	회원의 전화번호
Name	회원의 이름
Reg_date	회원의 가입날짜 및 시간

SCOTT.MEMBER		
P *	ID	VARCHAR2 (15 BYTE)
*	PASSWD	VARCHAR2 (12 BYTE)
	DATE_NUMBER	VARCHAR2 (15 BYTE)
	EMAIL	VARCHAR2 (30 BYTE)
	ADDRESS	VARCHAR2 (25 BYTE)
	TEL	VARCHAR2 (13 BYTE)
*	NAME	VARCHAR2 (15 BYTE)
*	REG_DATE	TIMESTAMP
MEMBER_PK (ID)		

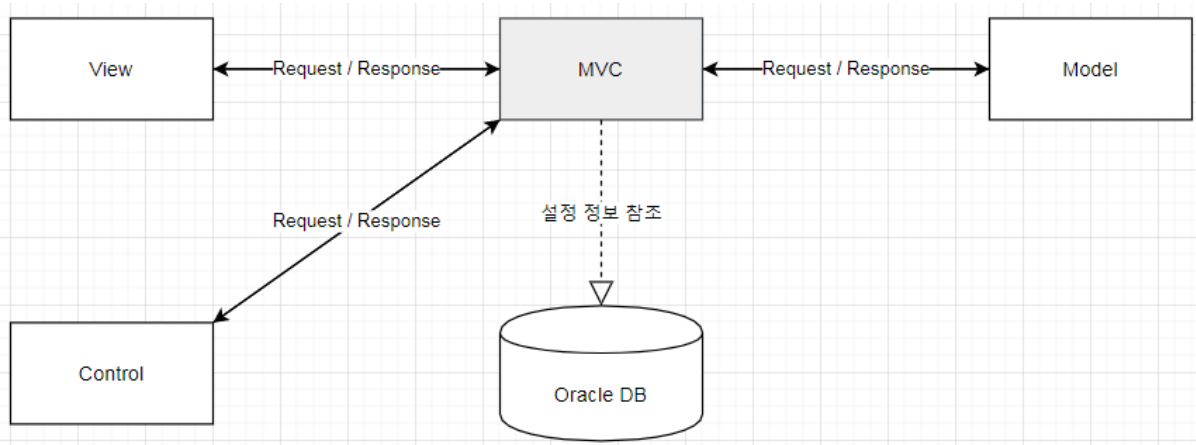
2. 게시판관리 테이블

Boarde	
속성	주요업무
Num	게시글 등록 순서 구분
Writer	'Member' 테이블의 id 값을 받아 게시글 등록한 사람 구분
Email	'Member' 테이블의 Email 값 받아 옴
Subject	게시글의 제목
Passwd	'Member' 테이블의 passwd 값을 받아 게시글을 수정, 삭제할 때 사용
Reg_date	게시글의 등록 날짜 및 시간
Readcount	게시글의 조회수
Ref	새 글의 등록 순서 구분
Re_step	답글의 순서 구분
Re_level	답글의 답글 구분
Content	게시글의 내용
Ip	게시글을 쓴 회원의 ip
Filename	게시글의 첨부파일 이름

SCOTT.BOARDE	
P * NUM	NUMBER (20)
WRITER	VARCHAR2 (15 BYTE)
EMAIL	VARCHAR2 (30 BYTE)
SUBJECT	VARCHAR2 (75 BYTE)
* PASSWD	VARCHAR2 (12 BYTE)
* REG_DATE	TIMESTAMP
READCOUNT	NUMBER
* REF	NUMBER (20)
* RE_STEP	NUMBER (10)
* RE_LEVEL	NUMBER (10)
* CONTENT	VARCHAR2 (3000 BYTE)
* IP	VARCHAR2 (20 BYTE)
FILENAME	VARCHAR2 (500 BYTE)
BOARDE_PK (NUM)	

[기능 설계]

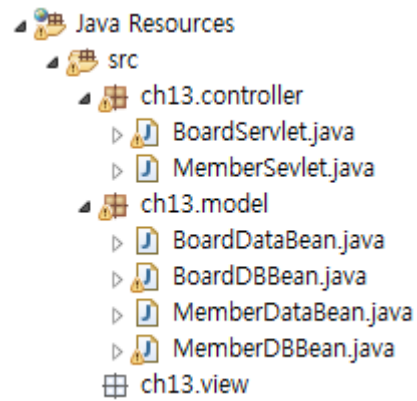
1. MVC 모델



2. 파일 구조

A. Java

i. 소스 코드

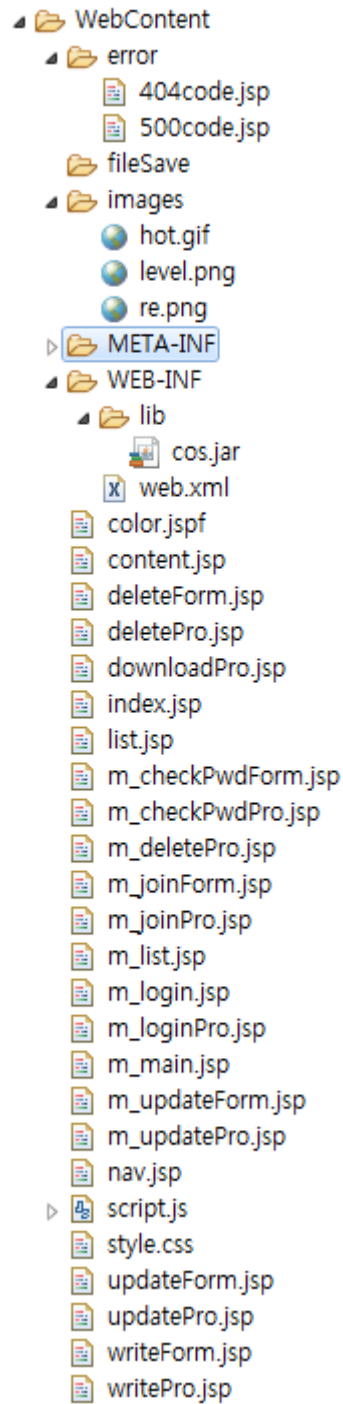


ii. Libraries 참조



B. JSP

i. WebContent'



3. 인터페이스 정의서

A. 회원관리 기능

환경	개발환경	서비스 그룹	화면처리	페이지	25~27
서비스 명	R-M-001				
설명	<p>'m_login.jsp'에서 input 태그로 아이디와 비밀번호를 입력 받아 'ch13.controller'의 'MemberServlet.java' 파일로 request 로 값을 전달, 전달받은 값을 바탕으로 'MemberDBBean.java'를 통해 Oracle DB 의 'Member'테이블에 연결해 등록되어 있는 회원인지, 아이디와 비밀번호가 맞는지 확인 후 session 에 id 값을 넣고 'm_loginPro.jsp'를 거쳐 로그인을 진행한다.</p>				
M_login.jsp 코드설명	<pre> <form name="login" action="LoginPro.do" method="post"> <table border=1 style="margin-left:auto; margin-right:auto;"> <tr> <td colspan="2" align="center">LOGIN</td> </tr> <tr align="center"> <td>아이디 :</td> <td><input type="text" name="id"></td> </tr> <tr align="center"> <td>비밀번호 :</td> <td><input type="password" name="passwd"></td> </tr> <tr> <td colspan="2" align="center"> <input type="submit" value="Login"> <input type="button" onclick="location.href='joinForm.do'" value="sign up"> </td> </tr> </table> </form> </pre> <p>테이블 형태로 만들어진 입력 폼에서 input 태그로 아이디와 비밀번호를 입력받아 'MemberServlet.java'의 'loginPro.do'로 이동한다.</p>				

MemberServlet.java
코드설명

```
if (action.equals("loginPro.do")) {
    String id = request.getParameter("id");
    String passwd = request.getParameter("passwd");
    response.setContentType("text/html;charset=UTF-8");
    try {
        MemberDBBean dbPro = MemberDBBean.getInstance();
        String pwd = dbPro.checkIdPw(id);
        if (pwd == null) {
            // id를 찾을 수 없음
            request.setAttribute("checkId", -1);
            viewPage = "m_login.jsp";
        } else if (passwd.equals(pwd)) {
            // id, passwd 동일, 로그인 성공
            request.setAttribute("checkId", 0);
            request.getSession().setAttribute("id", id);
            viewPage = "list.jsp";
        } else {
            // passwd가 틀림
            request.setAttribute("checkId", 1);
            viewPage = "m_login.jsp";
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

'm_login.jsp'에서 입력한 아이디 값을 request.getParameter 를 통해 String 변수로 받아 'MemberDBBean.java'의 'checkIdPw' 메소드에 값을 넣어 해당 아이디의 비밀번호를 전달받고 입력한 비밀번호와 같은지 확인한다.
로그인 성공 시 입력한 아이디 값은 session 에 값을 등록해 놓고 어디서든 사용 가능하도록 한다.
로그인의 성공여부는 'checkId'라는 변수에 숫자형태로 값을 넘겨준다.

<p>MemberDBBean.jav a 코드설명</p>	<pre> public String checkIdPw(String id) throws Exception { String dbpasswd = null; // 초기화를 null로 해주고 컨트롤러에서 null인 경우가 발생할 수 있게끔한다. Connection conn = null; PreparedStatement pstmt = null; ResultSet rs = null; try { conn = getConnection(); pstmt = conn.prepareStatement("select passwd from member where id = ?"); pstmt.setString(1, id); rs = pstmt.executeQuery(); if (rs.next()) { dbpasswd = rs.getString("passwd"); } } catch (Exception e) { e.printStackTrace(); } finally { closeDBResources(rs, pstmt, conn); } return dbpasswd; } </pre> <p>'m_login.jsp'에서 입력받은 아이디 값에 대한 비밀번호가 Oracle DB 에 존재하는지 확인하는 메소드이다.</p> <p>비밀번호가 존재한다면 String 변수 'dbpass'에 값을 담아 전달한다.</p>
<p>M_loginPro.jsp 코드설명</p>	<pre> <% String id = (String)session.getAttribute("id"); int checkId = (int)request.getAttribute("checkId"); if(checkId == -1){ %> <script> alert("해당 아이디를 찾을 수 없습니다."); </script> <% } else if (checkId == 1) { %> <script> alert("아이디와 패스워드가 일치하지 않습니다."); </script> <% } %> </pre> <p>'MemberServlet.java'에서 비밀번호 일치여부에 대한 값 'checkId'를 int 형 변수로 받아 오류팝업을 출력한다.</p>

환경	개발환경	서비스 그룹	화면처리	페이지	28~30
서비스 명	R-M-002				
설명	<p>'m_joinForm.jsp'에서 Oracle DB 의 'Member'테이블 속성값에 맞춰 input 태그에 입력해 'MemberServlet.java'에 request 로 값을 전달한다.</p> <p>전달받은 값을 'MemberDataBean.java'의 vo 로 값을 넣고 이 값들은 'MemberDBBean.java'를 통해 Oracle DB 의 'Member' 테이블에 insert 후 'm_joinPro.jsp'를 거쳐 회원가입이 완료된다.</p>				
MemberServlet.java 코드설명	<pre> if (action.equals("joinPro.do")) { int check = 0; try { MemberDataBean art = new MemberDataBean(); art.setId(request.getParameter("id")); art.setPasswd(request.getParameter("passwd")); art.setDate_number(request.getParameter("date_num")); art.setEmail(request.getParameter("email")); art.setAddress(request.getParameter("address")); art.setTel(request.getParameter("tel")); art.setName(request.getParameter("name")); art.setReg_date(new Timestamp(System.currentTimeMillis())); MemberDBBean dbPro = MemberDBBean.getInstance(); check = dbPro.joinIdPw(art); } catch (Exception e) { e.printStackTrace(); } request.setAttribute("check", check); viewPage = "m_joinPro.jsp"; } </pre> <p>'m_joinForm.jsp'에서 입력받은 값을 request.getParameter 를 통해 받아와 'MemberDataBean.java'의 vo 로 값을 넣는다.</p> <p>넣은 값들은 'MemberDBBean.java'의 'joinIdPw' 메소드에 넣어 Oracle DB 로 값을 전달하고, 결과값은 int 변수 'check'에 담아 'm_joinPro.jsp'로 'check'값을 전달한다.</p>				

<p>MemberDataBean.java</p> <p>a 코드설명</p>	<pre>public class MemberDataBean { private String id; private String passwd; private String date_number; private String email; private String address; private String tel; private String name; private Timestamp reg_date; }</pre> <p>Oracle DB 의 'Member' 테이블 속성값 자료형에 맞게 입력 후 getter/setter 한다.</p>
<p>MemberDBBean.java</p> <p>코드설명</p>	<pre>public int joinIdPw(MemberDataBean art) throws Exception { Connection conn = null; PreparedStatement pstmt = null; ResultSet rs = null; String sql = ""; int x = -1; try { conn = getConnection(); sql = "select id from member where id=?"; pstmt = conn.prepareStatement(sql); pstmt.setString(1, art.getId()); rs = pstmt.executeQuery(); if (rs.next()) { x = 0; } else { sql = "insert into member(id, passwd, date_number, email, address, tel, name, reg_date) " + " values (?, ?, ?, ?, ?, ?, ?, ?)"; pstmt = conn.prepareStatement(sql); pstmt.setString(1, art.getId()); pstmt.setString(2, art.getPasswd()); pstmt.setString(3, art.getDate_number()); pstmt.setString(4, art.getEmail()); pstmt.setString(5, art.getAddress()); pstmt.setString(6, art.getTel()); pstmt.setString(7, art.getName()); pstmt.setTimestamp(8, art.getReg_date()); pstmt.executeUpdate(); x = 1; } } catch (Exception ex) { ex.printStackTrace(); } finally { closeDBResources(rs, pstmt, conn); } return x; }</pre> <p>'MemberServlet.java'에서 전달받은 MemberDataBean 형태의 변수 'art'의 'id'값을 바탕으로 먼저 'Member'테이블에 값이 있는지 확인한다.</p> <p>해당하는 값이 없을 경우 'Member'테이블에 전달받은 값들을 insert 시킨다.</p> <p>결과값은 int 로 'x'라는 변수에 담아 return 한다.</p>

M_joinPro.jsp
코드설명

```
<%  
    int check = (int)request.getAttribute("check");  
    if(check==1){  
%>  
        <script type="text/javascript">  
            alert("회원가입이 성공적으로 완료되었습니다. 로그인해주세요.");  
            //hbbaek.m for 로그인으로 이동  
            location.href='login.du';  
        </script>  
%>  
    }else{  
%>  
        <script type="text/javascript">  
            alert("이미 존재하는 ID입니다.");  
            history.go(-1);  
        </script>  
%>  
    }  
%>
```

결과값 'check'를 int 형으로 받아 그에 대한 처리를 해준다.

환경	개발환경	서비스 그룹	화면처리	페이지	31~38
서비스 명	R-M-003				
설명	<p>회원정보를 조회할 수 있는 'm_main.jsp'로 진입 전 'MemberServlet.java'의 'main.do'를 호출하게 된다.</p> <p>'main.do'에서 session 에 저장된 'id'를 바탕으로 'MemberDBBean.java'의 'MainIdPw' 메소드를 호출해 메소드의 결과값을 MemberDataBean 타입의 'art' 변수에 담아 'm_main.jsp'로 전달한다.</p> <p>전달받은 'art' 변수의 'id'값을 바탕으로 관리자 계정인 'admin'인지 확인 후 관리자 계정이라면 'm_list.jsp'로 갈 수 있는 input 버튼이 나오고 일반회원이라면 'art' 변수의 값들로 회원정보를 확인한다.</p> <p>'m_list.jsp'에서는 회원 목록이 10 개 단위로 표시될 수 있도록 int 형 변수 'pageSize'를 만들어주고 페이지를 넘길 때 목록의 개수를 정의 할 수 있도록 String 타입의 'pageNum'변수를 request 받아 활용한다.</p> <p>회원의 수는 int 타입의 변수 'count'를 정의해 'MemberDBBean.java'의 'listCount' 메소드의 return 값을 저장하고 이 값이 0 보다 크다면 MemberDataBean 타입의 list 에 'MemberDBBean.java'의 'listId' 메소드의 값을 담아 반복문을 통해 각 회원의 정보를 나타낸다.</p>				
MemberServlet.java 코드설명	<pre> - 일반회원의 main 회원정보 if (action.equals("main.do")) { String id = (String) request.getSession().getAttribute("id"); try { MemberDBBean dbPro = MemberDBBean.getInstance(); MemberDataBean art = dbPro.MainIdPw(id); request.setAttribute("vo", art); } catch (Exception e) { e.printStackTrace(); } viewPage = "m_main.jsp"; } </pre> <p>Session 에 저장된 'id'를 String 타입의 변수로 받아 'MemberDBBean.java'의 'MainIdPw' 메소드 호출하고 그 결과를 MemberDataBean 타입의 변수 'art'로 전달한다.</p>				

<p>M_main.jsp 코드설명</p>	<pre> <% String id = (String)session.getAttribute("id"); MemberDataBean art = (MemberDataBean)request.getAttribute("vo"); SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd"); if (id.equals("admin")) { %> <p>관리자님, 어서오세요.</p> <table style="margin-left:auto; margin-right:auto;"> <tr align="center"> <td> <input type="button" onclick="location.href='list.du'" value="회원목록"> </td> </tr> </table> <% }else{ int check=0; %> <p><%= id %>님의 정보</p> 'MemberServlet.java'에서 전달받은 값들을 request.getAttribute 로 MemberDataBean 타입의 변수 'art'로 받아와 'art'의 'id'로 관리자계정 'admin'과 일치 여부 확인해 각 상황에 맞는 화면을 출력한다. </pre>
<p>MemberDBBean.java a 코드설명</p>	<p>- 일반회원 m_main.jsp 에서 사용</p>

```

public MemberDataBean MainIdPw(String id) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    String sql = "";

    MemberDataBean art = null;

    try {
        conn = getConnection();

        sql = "select * from member where id=?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, id);
        rs = pstmt.executeQuery();

        if (rs.next()) {
            art = new MemberDataBean();
            art.setId(rs.getString("id"));
            art.setPasswd(rs.getString("passwd"));
            art.setDate_number(rs.getString("date_number"));
            art.setEmail(rs.getString("email"));
            art.setAddress(rs.getString("address"));
            art.setTel(rs.getString("tel"));
            art.setName(rs.getString("name"));
            art.setReg_date(rs.getTimestamp("reg_date"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeDBResources(rs, pstmt, conn);
    }
    return art;
}

```

전달받은 String 타입의 변수 'id'를 바탕으로 'Member' 테이블에서 해당하는 데이터만을 select 로 검색한 값들을 'MemberDataBean.java'로 vo 로 값을 넣어 return 한다.

- 관리자 계정 m_list.jsp 의 총 회원의 수

```

public int listCount() {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    int x = 0;
    String sql = "";

    try {
        conn = getConnection();

        sql = "select count(*) from member";
        pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();

        if(rs.next()) {
            x = rs.getInt(1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeDBResources(rs, pstmt, conn);
    }
    return x;
}

```

'Member' 테이블에서 전체 count 를 select 하여 그 값을 int 타입의 변수에 넣어 return 한다.

- 관리자 계정 m_list.jsp 의 회원들의 정보

	<pre> public List<MemberDataBean> listId(int start, int end) { Connection conn = null; PreparedStatement pstmt = null; ResultSet rs = null; List<MemberDataBean> artList = null; String sql = ""; try { conn = getConnection(); sql = "select * from (select rownum rnum, B.* from (select * from member order by id) B)"; sql += " where rnum >=? and rnum <=?"; pstmt = conn.prepareStatement(sql); pstmt.setInt(1, start); pstmt.setInt(2, end); rs = pstmt.executeQuery(); if(rs.next()) { artList = new ArrayList<MemberDataBean>(end); do { MemberDataBean art = new MemberDataBean(); art.setId(rs.getString("id")); art.setPasswd(rs.getString("passwd")); art.setDate_number(rs.getString("date_number")); art.setEmail(rs.getString("email")); art.setAddress(rs.getString("address")); art.setTel(rs.getString("tel")); art.setName(rs.getString("name")); art.setReg_date(rs.getTimestamp("reg_date")); artList.add(art); }while(rs.next()); } }catch (Exception e) { e.printStackTrace(); }finally { closeDBResources(rs, pstmt, conn); } return artList; } </pre> <p>'Member' 테이블의 rnum 을 통해 int 타입의 'start'값과 'end'값으로 불러오하고자 하는 회원의 범위를 지정한다.</p> <p>Rnum 은 테이블이 검색되고 난 이후 적용되는 임시번호이므로 sql 문을 위와같이 분리해서 적어준다.</p> <p>Select 한 내용은 MemberDataBean 타입의 arraylist 에 vo 로 담아 return 한다.</p>
M_list.jsp 코드설명	<pre> <%! int pageSize = 10; SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd"); %> </pre> <p>회원 목록의 표시 개수를 int 타입의 'pageSize' 변수로 선언해 놓는다.</p>

```

<%
String pageNum = request.getParameter("pageNum");

if(pageNum == null){
    pageNum = "1";
}

int currentPage = Integer.parseInt(pageNum);
int startRow = (currentPage-1)*pageSize+1;
int endRow = currentPage*pageSize;
int count=0;
int number=0;
List<MemberDataBean> artList = null;

MemberDBBean dbPro = MemberDBBean.getInstance();
count = dbPro.listCount();

if(count>0){
    artList = dbPro.listId(startRow, endRow);
}

number = count-(currentPage-1)*pageSize;
%>

```

회원 목록의 페이지 값 String 타입의 'pageNum' 값을 전달받지 않고 바로 'm_list.jsp'로 온 경우 String 으로 1 이 들어가도록 if 문을 통해 지정해주고 'pageNum'의 값을 int 타입의 'currentPage' 변수로 타입변환을 통해 넣어준다.

회원 목록을 호출하기 위해 필요한 시작하는 행과 끝나는 행을 int 타입의 'startRow'와 'endRow'에 'currentPage'와 'pageSize'를 통해 구하고 'MemberDBBean.java'의 'listCount'메소드를 통해 총 회원의 수를 int 타입의 'count'로 넣고 'count'가 0 보다 클 경우 'MemberDBBean.java'의 'listId' 메소드를 호출해 회원 목록의 값을 MemberDataBean 타입의 list 에 담는다.

```

<% if(count==0){ %>
<table border=1 style="margin-left: auto; margin-right: auto;">
    <tr>
        <td align="center" width="700">
            등록된 회원이 없습니다.
        </td>
    </tr>
</table>

```

'count'가 0 이라면 등록된 회원이 없다는 문구가 표시되도록 한다.

```

<% }else{ %>
<table border=1 style="margin-left:auto; margin-right:auto;">
  <tr>
    <td align="center">변 호</td>
    <td align="center">아 이 디</td>
    <td align="center">비밀번호</td>
    <td align="center">생년월일</td>
    <td align="center">이 메 일</td>
    <td align="center">주 소</td>
    <td align="center">전화번호</td>
    <td align="center">이 름</td>
    <td align="center">가입일자</td>
    <td align="center">삭 제</td>
  </tr>
  <%
    for(int i=0; i<artList.size(); i++){
      MemberDataBean art = artList.get(i);
  %>
  <tr>
    <td><%=number-- %></td>
    <td align="center"><%=art.getId() %></td>
    <td align="center"><%=art.getPasswd() %></td>
    <td align="center"><%=art.getDate_number() %></td>
    <td align="center"><%=art.getEmail()%></td>
    <td align="center"><%=art.getAddress() %></td>
    <td align="center"><%=art.getTel() %></td>
    <td align="center"><%=art.getName() %></td>
    <td align="center"><%=sdf.format(art.getReg_date()) %></td>
    <td align="center">
      <input type="button" onclick="Location.href='deletePro.do?id=<%=art.getId() %>&pageNum=<%=pageNum %>' "
        value=" 4#"/>
    </td>
  </tr>
  <% } %>
</table>

```

회원의 목록은 반복문을 통해 list 변수 'artList' 의 크기만큼만 돌도록 지정해 테이블형태로 구현한다.

```

<%
}
if(count>0){
    int pageCount = count/pageSize+(count%pageSize==0?0:1);
    int startPage=1;

    if(currentPage%10!=0)
        startPage=(int)(currentPage/10)*10+1;
    else
        startPage=((int)(currentPage/10)-1)*10+1;

    int pageBlock=10;
    int endPage=startPage+pageBlock-1;

    if(endPage>pageCount)
        endPage=pageCount;

    if(startPage>10){
%>
        <a href="list.du?pageNum=<%=startPage-10 %>">[이전]</a>
<%
    }

    for(int i=startPage; i<=endPage; i++){
%>
        <a href="list.du?pageNum=<%=i %>">[<%=i %>]</a>
<%
    }

    if(endPage<pageCount){
%>
        <a href="list.du?pageNum=<%=startPage+10 %>">[다음]</a>
<%
    }
}
%>

```

회원 목록의 페이지가 넘어갈 때 'startRow'의 값을 변경하도록 해주고 페이지를 넘길때마다 'pageNum'값을 전달한다.

환경	개발환경	서비스 그룹	화면처리	페이지	39~43
서비스 명	R-M-004				
설명	<p>'m_main.jsp'에서 정보수정을 하기 전에 비밀번호를 확인하도록 'm_checkPwdForm.jsp'으로 갈 때 int 타입의 'check' 변수에 수정과 삭제를 구분하기 위한 값을 담아간다.</p> <p>'m_checkPwdForm.jsp'에서 비밀번호를 입력해 'MemberServlet.java'의 'checkPwdPro.do'로 넘어가 session 에 저장된 'id'값을 string 변수로 받고 입력받은 비밀번호 값과 'check'변수 값을 request 로 받아와 'id'는 'MemberDBBean.java'의 'checkIdPw' 메소드를 호출해 비밀번호 값을 return 받아와 입력한 비밀번호 값과 일치여부를 확인한다.</p> <p>'check'변수로는 정보수정과 회원탈퇴의 페이지를 구분해서 이동할 때 사용한다.</p> <p>이에 대한 결과값을 int 타입의 'checkId' 변수에 담아 'm_checkPwdPro.jsp'로 전달한다.</p> <p>'checkId' 변수를 request 로 받아와 'm_updateForm.jsp'로 연결해 id 를 제외한 값들을 입력받아 'MemberServlet.java'의 'updatePro.do'로 전달하고 이 값들을 MemberDataBean 타입의 변수에 vo 로 입력해 'MemberDBBean.java'의 'UpdateIdPw' 메소드에 session 의 'id'값과 같이 전달하여 'Member' 테이블을 수정한다.</p> <p>회원정보가 바꿨으므로 이 회원이 쓴 글들의 정보도 'BoardDBBean.java'의 'ChangeBoard' 메소드를 호출해 같이 변경해준다.</p>				
M_ main.jsp 코드설명	<pre> if (id.equals("admin")) { <p>관리자님, 머시오세요.</p> <table style="margin-left: <tr align="center"> <td> <input type="t </td> </tr> </table> }else{ <u>int check=0;</u> </pre> <p>Session 의 'id'값이 'admin'인지 판별여부 진행 후 일반회원일 경우 int 타입의 'check' 변수에 0 을 넣어준다.</p> <pre> <input type="button" onclick="location.href='checkPwd.do?check=<%=check %>' " value="정보수정"> <% check=1; %> <input type="button" onclick="location.href='checkPwd.do?check=<%=check %>' " value="회원탈퇴"> </pre>				

	'check' 변수의 값을 'm_checkPwdForm.jsp'로 넘어갈 때 넘겨준다.
MemberServlet.java 코드설명	<pre> - 비밀번호 확인 checkPwdPro.do if (action.equals("checkPwdPro.do")) { String id = (String) request.getSession().getAttribute("id"); String passwd = request.getParameter("passwd"); int check = Integer.parseInt(request.getParameter("check")); response.setContentType("text/html;charset=UTF-8"); try { MemberDBBean dbPro = MemberDBBean.getInstance(); String pwd = dbPro.checkIdPw(id); MemberDataBean art = dbPro.MainIdPw(id); if (passwd.equals(pwd)) { if (check == 0) { request.setAttribute("checkId", 0); } else { request.setAttribute("checkId", 2); } } else { request.setAttribute("checkId", 1); request.setAttribute("vo", art); } } catch (Exception e) { e.printStackTrace(); } viewPage = "m_checkPwdPro.jsp"; } </pre> <p>Session 에 저장된 id 값 request 를 통해 String 변수로 받아 'MemberDBBean.java'의 'checkId' 메소드를 통해 'Member' 테이블에 등록된 비밀번호를 받아온 값과 'm_checkPwdForm.jsp'에서 입력받은 비밀번호를 request 로 받은 값과 일치여부를 확인한다.</p> <p>'m_main.jsp'에서 넘겨받은 int 타입의 check 변수를 if 문을 통해 구분하고 각 상황에 맞게 'checkId' 변수에 일치여부 결과값을 담아 전달한다.</p> <pre> - 정보수정 updatePro.do </pre>

	<pre> if (action.equals("updatePro.du")) { String id = (String) request.getSession().getAttribute("id"); try { MemberDataBean art = new MemberDataBean(); art.setPasswd(request.getParameter("passwd")); art.setDate_number(request.getParameter("date_num")); art.setEmail(request.getParameter("email")); art.setAddress(request.getParameter("address")); art.setTel(request.getParameter("tel")); art.setName(request.getParameter("name")); MemberDBBean dbPro = MemberDBBean.getInstance(); dbPro.UpdateIdPw(art, id); BoardDBBean bDbPro = BoardDBBean.getInstance(); bDbPro.ChangeBoard(id); } catch (Exception e) { e.printStackTrace(); } viewPage = "m_updatePro.jsp"; } </pre> <p>Session 에 등록된 id 값을 String 변수로 받고 'm_updateForm.jsp'에서 입력한 값들을 request 로 받아 MemberDataBean 타입의 변수 art 에 vo 로 값을 입력한다.</p> <p>이 두 값으로 'MemberDBBean.java'의 'UpdateIdPw' 메소드를 호출한다. 회원정보가 변경되었으므로 게시글의 정보도 변경해주기 위해 'BoardDBBean.java'의 'ChangeBoard' 메소드를 통해 같이 변경해준다.</p>
M_checkPwdPro.jsp 코드설명	<pre> <% int checkId = (int)request.getAttribute("checkId"); if(checkId==1){ %> <script type="text/javascript"> alert("비밀번호가 맞지 않습니다."); history.go(-1); </script> <% }else if(checkId==0){%> <meta http-equiv="Refresh" content="0;url=update.du"> <% }else if(checkId==2){ %> <meta http-equiv="Refresh" content="0;url=deletePro.du"> <% } %> </pre> <p>'MemberServlet.java'에서 넘겨받은 checkId 변수를 int 타입의 변수에 넣어</p>

	상황에 맞게 view 를 이동시킨다.
MemberDBBean.jav a 코드설명	<pre> public void UpdateIdPw(MemberDataBean art, String id) { Connection conn = null; PreparedStatement pstmt = null; ResultSet rs = null; String sql = ""; try { conn = getConnection(); sql = "update member set passwd=?, date_number=?, email=?, address=?, tel=?, name=? where id=?"; pstmt = conn.prepareStatement(sql); pstmt.setString(1, art.getPasswd()); pstmt.setString(2, art.getDate_number()); pstmt.setString(3, art.getEmail()); pstmt.setString(4, art.getAddress()); pstmt.setString(5, art.getTel()); pstmt.setString(6, art.getName()); pstmt.setString(7, id); pstmt.executeUpdate(); } catch (Exception e) { e.printStackTrace(); } finally { closeDBResources(rs, pstmt, conn); } } </pre> <p>전달받은 값을 바탕으로 'Member' 테이블의 값을 update 문을 통해 변경해준다.</p>

BoardDBBean.java
코드설명

```
public void ChangeBoard(String id) {  
    Connection conn = null;  
    PreparedStatement pstmt = null;  
    ResultSet rs = null;  
  
    String sql = "";  
  
    try {  
        conn = getConnection();  
  
        sql = "select * from member where id=?";  
        pstmt = conn.prepareStatement(sql);  
        pstmt.setString(1, id);  
        rs = pstmt.executeQuery();  
  
        if(rs.next()) {  
            sql = "update boarde set email=?, passwd=? where writer=?";  
            pstmt = conn.prepareStatement(sql);  
            pstmt.setString(1, rs.getString("email"));  
            pstmt.setString(2, rs.getString("passwd"));  
            pstmt.setString(3, rs.getString("id"));  
            pstmt.executeUpdate();  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        closeDBResources(rs, pstmt, conn);  
    }  
}
```

'Member' 테이블에서 변경된 값을 'Boarde' 테이블에 연계되는 속성들만 update 문으로 변경해준다.

환경	개발환경	서비스 그룹	화면처리	페이지	44~46
서비스 명	R-M-005				
설명	<p>일반회원일 경우 'm_main.jsp'의 int 타입의 변수 check 값을 가지고 'm_checkPwdForm.jsp'로 넘어가 비밀번호를 입력한다.</p> <p>'MemberServlet.java'의 'checkPwdPro.do'에서 request 로 check 와 입력한 비밀번호를 받아오고, 비밀번호 값과 session 에 등록된 id 를 바탕으로 'MemberDBBean.java'의 'CheckId' 메소드를 호출해 'Member' 테이블의 비밀번호와 일치여부를 확인하고 check 를 통해 회원정보와 회원탈퇴에 대한 구분 값 int 타입의 checkId 변수에 값을 넣어 'm_checkPwdPro.jsp'로 이동한다.</p> <p>checkId 를 바탕으로 'MemberServlet.java'의 'deletePro.do'로 이동한다.</p> <p>관리자계정의 'admin'의 경우 'm_list.jsp'에서 회원의 id 와 pageNum 을 가지고 'MemberServlet.java'의 'deletePro.do'로 이동한다.</p> <p>'MemberServlet.java'의 'deletePro.do'에서 session 의 id 와 pageNum 을 String 변수에 넣고 id 가 'admin'이면 String 변수 mId 에 request 로 받아온 id 를 넣어 'MemberDBBean.java'의 'deleteIdPw' 메소드를 호출하고 결과값으로 int 타입의 deleteId 와 pageNum, mId 를 'm_deletePro.jsp'로 전달한다.</p> <p>일반회원일 경우는 deleteId 만 'm_deletePro.jsp'로 전달한다.</p>				
M_ main.jsp 코드설명	<pre><% check=1; %> <input type="button" onclick="location.href='checkPwd.do?check=<%=check %>'" value="회원탈퇴"></pre> <p>일반회원일 경우 int 타입의 변수 check 를 'm_checkPwdForm.jsp'로 넘긴다.</p>				

<p>MemberSevlet.java 코드설명</p>	<pre> if (action.equals("deletePro.du")) { String id = (String) request.getSession().getAttribute("id"); String pageNum = request.getParameter("pageNum"); try { MemberDBBean dbPro = MemberDBBean.getInstance(); if (id.equals("admin")) { String mId = request.getParameter("id"); dbPro.deleteIdPw(mId); request.setAttribute("deleteId", 0); request.setAttribute("pageNum", pageNum); request.setAttribute("mId", mId); } else { dbPro.deleteIdPw(id); request.setAttribute("deleteId", 1); } } catch (Exception e) { e.printStackTrace(); } viewPage = "m_deletePro.jsp"; } </pre> <p>Session 에 저장된 id 값이 'admin'이면 request 로 회원의 id 값을 String 변수에 저장하고 이 값을 바탕으로 'MemberDBBean.java'의 'deleteIdPw' 메소드를 호출한다.</p> <p>일반회원이라면 session 의 id 값으로 호출한다.</p>
<p>MemberDBBean.java a 코드설명</p>	<pre> public void deleteIdPw(String id) { Connection conn = null; PreparedStatement pstmt = null; ResultSet rs = null; String sql = ""; try { conn = getConnection(); sql = "delete from member where id=?"; pstmt = conn.prepareStatement(sql); pstmt.setString(1, id); rs = pstmt.executeQuery(); } catch (Exception e) { e.printStackTrace(); } finally { closeDBResources(rs, pstmt, conn); } } </pre>

	받은 id 값으로 'Member' 테이블에서 delete 문으로 id 가 해당하는 값들을 삭제한다.
--	---

B. 게시판관리 기능

환경	개발환경	서비스 그룹	화면처리	페이지	47~52
서비스 명	R-B-001				
설명	<p>'list.jsp'에서 페이지에 표시되는 글의 개수를 지정한 int 타입의 pageSize 를 선언하고 현재 페이지의 번호를 저장할 String 타입의 pageNum 변수를 지정한다.</p> <p>pageNum 을 int 타입의 변수 currentPage 로 타입변환을 통해 넣어주고 이 값을 바탕으로 불러올 글을 지정하는 int 타입의 startRow 와 endRow 변수를 설정한다.</p> <p>총 글의 개수를 넣을 int 타입의 count 변수를 선언하고 'BoardDBBean.java'의 'getArticleCount' 메소드를 호출하여 넣어준다.</p> <p>Count 가 0 보다 크다면 BoardDataBean 타입의 list articleList 에 'BoardDBBean.java'의 'getArticles' 메소드에 startRow 와 endRow 를 넣어 호출하여 값을 넣는다.</p> <p>글번호는 int 타입의 변수 number 를 선언하고 count 와 currentPage, pageSize 로 지정해준다.</p> <p>articleList 의 값들을 반복문을 통해 구현해주면 글목록 화면이 구현된다.</p>				
List.jsp 코드설명	<pre> <%! int pageSize = 10; SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm"); %> </pre> <p>글의 개수를 지정한 int 타입의 pageSize 를 선언하고 작성일을 표시할 date 타입을 설정한다.</p>				


```

<%
    request.setCharacterEncoding("UTF-8");

    String pageNum = request.getParameter("pageNum");

    if (pageNum == null) {
        pageNum = "1";
    }

    int currentPage = Integer.parseInt(pageNum); //1, 2
    int startRow = (currentPage - 1) * pageSize + 1; //1, 1
    int endRow = currentPage * pageSize; //10, 20
    int count = 0;
    int number = 0;
    List<BoardDataBean> articleList = null;

    BoardDBBean dbPro = BoardDBBean.getInstance();
    count = dbPro.getArticleCount();

    if (count > 0) {
        /*mysql의 limit (시작하는 인덱스, 몇개씩 읽머라)
        articleList = dbPro.getArticles(startRow, pageSi;
        oracle은 (시작하는 인덱스, 마지막 인덱스)*/
        //hbbaek.m for oracle
        articleList = dbPro.getArticles(startRow, endRow);
    }

    number = count-(currentPage-1)*pageSize;
%>

```

현재 페이지를 request 로 받아올 String 타입의 변수 pageNum 을 설정하고 이 값이 null 일 경우 1 이 되도록 조건문을 만든다.

글의 번호와 불러오고자 하는 시작번호, 끝번호를 지정하기위해 int 타입의 currentPage 에 pageNum 을 형변환을 통해 넣어주고 startRow 와 endRow 를 계산한다.

글의 총 개수 int 타입의 count 를 선언하고 'BoardDBBean.java'의 'getArticleCount' 메소드를 호출해 값을 넣어주고 이를 바탕으로 글번호 number 의 값을 구한다.

Count 가 0 보다 크다면 startRow 와 endRow 값을 넣어 'getArticles' 메소드를 호출해 BoardDataBean 타입의 List 변수 articleList 에 값들을 넣어준다.

	<pre> <% for (int i = 0 ; i < articleList.size() ; i++) { BoardDataBean article = articleList.get(i); } articleList 의 크기만큼 반복문을 통해 테이블에 목록에 표시될 항목과 number 값을 1 씩 차감하며 넣어준다. <% if (count > 0) { int pageCount = count / pageSize + (count % pageSize == 0 ? 0 : 1); int startPage =1; if(currentPage % 10 != 0) startPage = (int)(currentPage/10)*10 + 1; else startPage = ((int)(currentPage/10)-1)*10 + 1; int pageBlock = 10; int endPage = startPage + pageBlock - 1; if (endPage > pageCount) endPage = pageCount; if (startPage > 10) { %> <a href="list.do?pageNum=<%= startPage - 10 %>">[이전] } for (int i = startPage ; i <= endPage ; i++) { %> <a href="list.do?pageNum=<%= i %>">[<%= i %>] } if (endPage < pageCount) { %> <a href="list.do?pageNum=<%= startPage + 10 %>">[다음] } } %> </pre> <p>페이지마다 호출하는 startRow 값은 변화가 필요하므로 위와같이 pageNum 에따라 계산식을 세워준다.</p>
BoardDBBean.java 코드설명	<p>- 글의 총 개수 getArticleCount</p>

```

public int getArticleCount() {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    int x = 0;

    try {
        conn = getConnection();

        pstmt = conn.prepareStatement("select count(*) from board");
        rs = pstmt.executeQuery();

        if(rs.next()) {
            x = rs.getInt(1);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
        closeDBResources(rs, pstmt, conn);
    }
    return x;
}

```

'Board' 테이블의 전체 count 개수를 select 로 구해 int 타입의 변수로 return 한다.

- 글 목록 내용 getArticles

```

public List<BoardDataBean> getArticles(int start, int end) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    List<BoardDataBean> articleList = null;

    try {
        conn = getConnection();

        //BoardDAO - boardList 메서드 페이징 처리
        //pstmt = conn.prepareStatement("select * from boarde order by ref desc, re_step asc");
        String sql = "SELECT * "
            + "FROM (SELECT ROWNUM rnum, B.* FROM "
            + " (SELECT * FROM boarde ORDER BY ref desc, re_step asc ) B ) ";
        sql += " WHERE rnum >= ? and rnum <= ?";
        //rnum 생성하자마자 조건 체크가 안되기 때문에 위와 같이

        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, start);
        pstmt.setInt(2, end);
        rs = pstmt.executeQuery();

        if(rs.next()) {
            articleList = new ArrayList<BoardDataBean>(end);
            do {
                BoardDataBean article = new BoardDataBean();
                article.setNum(rs.getInt("num"));
                article.setWriter(rs.getString("writer"));
                article.setEmail(rs.getString("email"));
                article.setSubject(rs.getString("subject"));
                article.setPasswd(rs.getString("passwd"));
                article.setReg_date(rs.getTimestamp("reg_date"));
                article.setReadcount(rs.getInt("readcount"));
                article.setRef(rs.getInt("ref"));
                article.setRe_step(rs.getInt("re_step"));
                article.setRe_level(rs.getInt("re_level"));
                article.setContent(rs.getString("content"));
                article.setIp(rs.getString("ip"));

                articleList.add(article);
            }while(rs.next());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return articleList;
}

```

‘Boarde’ 테이블의 rnum 값은 테이블이 검색되고 알 수 있으므로 위와같이 sql 문을 나눠서 적어주고 새 글과 답글이 구분되어야 하므로 ref 와 re_step, re_level 로 정렬한 내용을 받아온 start 와 end 값으로 검색해준다.

검색한 값은 BoardDataBean 타입의 list 변수 articleList 에 do while 반복문으로 넣어주고 return 한다.

BoardDataBean.java
a 코드설명

```
public class BoardDataBean {  
    private int num;  
    private String writer;  
    private String subject;  
    private String email;  
    private String content;  
    private String passwd;  
    private Timestamp reg_date;  
    private int readcount;  
    private String ip;  
    private int ref;  
    private int re_step;  
    private int re_level;  
    //hbbaek.m for to add fileName  
    private String fileName;
```

‘Boarde’ 테이블의 속성과 타입을 기반으로 위와 같이 선언해주고 getter / setter 를 만든다.

환경	개발환경	서비스 그룹	화면처리	페이지	53~56
서비스 명	R-B-002				
설명	<p>'writeForm.jsp'에서 글번호, 새 글과 답글의 구분을 위한 int 타입의 변수들 num, ref, re_step, re_level 을 선언하고 값을 지정한다.</p> <p>내용에 입력할 value 값으로 지정할 String 변수 strV 를 공백으로 선언한다.</p> <p>Request 로 받은 num 의 값이 null 이 아닐 경우 선언한 int 형 변수들에 각각의 값들을 형변환을 통해 request 로 넣어준다.</p> <p>Int 형 변수들의 값들은 input 태그 hidden 으로 넘겨주고 나머지 값들은 input 태그로 입력받아 'BoardServlet.java'의 'writePro.do'로 넘겨준다.</p> <p>'writePro.do'에서는 로그인한 계정 바탕으로 게시글의 회원정보를 저장하므로 session 에 등록된 id 를 바탕으로 'MemberDBBean.java'의 'MainIdPw' 메소드를 호출해 MemberDataBean 타입의 변수 mArt 에 값을 저장해놓고 BoardDataBean 타입의 변수 art 에 'writeForm.jsp'에서 받아온 값들과 함께 필요한 값들을 vo 로 넣어준다.</p> <p>Art 변수를 이용해 'BoardDBBean.java'의 'InsertArticle' 메소드를 호출해 'Boarde' 테이블에 글을 등록한다.</p>				
wrtieForm.jsp 코드설명	<pre> <% int num = 0, ref = 1, re_step = 0, re_level = 0; String strV = ""; try{ if(request.getParameter("num")!=null){ num=Integer.parseInt(request.getParameter("num")); ref=Integer.parseInt(request.getParameter("ref")); re_step=Integer.parseInt(request.getParameter("re_step")); re_level=Integer.parseInt(request.getParameter("re_level")); } } /* WritePro.jsp에서 jsp:useBean으로 자료를 넘기게되면 중간과정 거치지 않고 바로 BoardDBBe 넘여가기 때문에 사전에 확인할 수 있도록 WriteForm.jsp에서 위와같이 null 체크해주는 코드가 필요함 */ %> </pre> <p>글번호, 새 글과 답글의 구분을 위한 변수들을 위와 같이 선언해준다.</p> <p>Request 로 받아온 num 값이 null 이 아닐경우 조건문을 통해 해당 값들을 request 로 넣어준다.</p> <pre> <input type="hidden" name="num" value="<%=num%>"> <input type="hidden" name="ref" value="<%=ref%>"> <input type="hidden" name="re_step" value="<%=re_step%>"> <input type="hidden" name="re_level" value="<%=re_level%>"> </pre> <p>Num 과 ref, re_step, re_level 은 직접 입력하는 값이 아니라 새 글과 답글의 구분을 위한 변수들이므로 hidden 으로 자체적으로 넘겨준다.</p>				

BoardServlet.java
코드설명

```
if(action.equals("writePro.do")) {
    String realFolder=""; //웹 어플리케이션상의 절대 경로
    String filename = "";

    //파일이 업로드되는 폴더 지정
    String saveFolder="/fileSave";
    String encType="utf-8"; //인코딩타입
    int maxSize=5*1024*1024; //최대 업로드될 파일크기 5Mb

    ServletContext context = getServletContext();
    //현재 .jsp페이지의 웹 어플리케이션상의 절대 경로 구함
    realFolder = context.getRealPath(saveFolder);
    MultipartRequest multi = null;
    try {
        // 전송을 담당할 콤포넌트를 생성하고 파일을 전송한다.
        // 전송할 파일명을 가지고 있는 객체, 서버상의 절대경로, 최대 업로드될 파일크기, 문자코드, 기본 보안 적용
        multi = new MultipartRequest(request, realFolder, maxSize, encType, new DefaultFileRenamePolicy());

        // Form의 파라미터 목록을 가져온다
        // Enumeration<?> params = multi.getParameterNames();

        // 전송한 파일 정보를 가져와 출력한다
        Enumeration<?> files = multi.getFileNames();

        //파일 정보 있으면
        while(files.hasMoreElements()){
            //input 태그의 속성이 file인 태그의 name 속성값 : 파라미터이름
            String name = (String)files.nextElement();

            //서버에 저장된 파일 이름
            filename = multi.getFileName(name);
        }
    }
```

파일의 업로드, 다운로드를 위해 위와같이 multipart 를 사용하는 코드들을 넣어준다.

	<pre>String id = (String)request.getSession().getAttribute("id"); MemberDBBean mDbPro = MemberDBBean.getInstance(); MemberDataBean mArt = mDbPro.MainIdPw(id); BoardDataBean art = new BoardDataBean(); art.setNum(Integer.parseInt(multi.getParameter("num"))); art.setWriter(mArt.getId()); art.setEmail(mArt.getEmail()); art.setSubject(multi.getParameter("subject")); art.setPasswd(mArt.getPasswd()); art.setContent(multi.getParameter("content")); art.setRef(Integer.parseInt(multi.getParameter("ref"))); art.setRe_step(Integer.parseInt(multi.getParameter("re_step"))); art.setRe_level(Integer.parseInt(multi.getParameter("re_level"))); art.setReg_date(new Timestamp(System.currentTimeMillis())); art.setIp(request.getRemoteAddr()); // hbbaek.a for file upload art.setFileName(filename); BoardDBBean dbPro = BoardDBBean.getInstance(); dbPro.insertArticle(art);</pre> <p>Session 에 저장된 id 값을 바탕으로 게시글에 등록될 회원정보를 'MemberDBBean.java'의 'MainIdPw' 메소드를 호출하여 MemberDataBean 타입의 변수 mArt 에 넣어주고 BoardDataBean 타입의 변수 art 에 request 로 입력받은 값들을 불러와 같이 입력해준 다음 'BoardDBBean'의 'insertArticle' 메소드를 호출한다.</p>
BoardDBBean.java a 코드설명	<pre>int num = article.getNum(); int ref = article.getRef(); int re_step = article.getRe_step(); int re_level = article.getRe_level(); int number = 0;</pre> <p>메소드 호출 시 입력받은 article 값의 num, ref, re_step, re_level 을 vo 를 통해 각각의 int 변수에 넣어주고 새 글의 번호가 될 int 타입의 변수 number 를 선언해준다.</p> <pre>sql = "select max(num) from board"; pstmt = conn.prepareStatement(sql); rs = pstmt.executeQuery(); if(rs.next()) { number = rs.getInt(1) + 1; }else { number = 1; }</pre>

select 문으로 'Boarde' 테이블의 num 값 중 최고값을 구해 값이 있다면 number 에 +1 을 해서 넣어준다.

```
if(num!=0) { //댓글쓰기
    sql = "update boarde set re_step = re_step + 1 where ref = ? and re_step > ?";
    pstmt = conn.prepareStatement(sql);
    pstmt.setInt(1, ref);
    pstmt.setInt(2, re_step);
    pstmt.executeUpdate();
    re_step = re_step + 1;
    re_level = re_level + 1;
}else { //원본글쓰기
    ref = number;
    re_step = 0;
    re_level = 0;
}
```

num 값을 통해 새 글과 답글의 구분을 지어 답글이라면 update 문을 통해 'Boarde' 테이블에서 re_step 을 1 늘려주고 글을 저장할 때 추가 할 수 있도록 받아온 re_step 과 re_level 을 1 씩 늘려준다.

```
sql = "insert into boarde (num, writer, email, subject, passwd, reg_date, ";
sql += "ref, re_step, re_level, content, ip, filename) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, number);
pstmt.setString(2, article.getWriter());
pstmt.setString(3, article.getEmail());
pstmt.setString(4, article.getSubject());
pstmt.setString(5, article.getPasswd());
pstmt.setTimestamp(6, article.getReg_date());
pstmt.setInt(7, ref);
pstmt.setInt(8, re_step);
pstmt.setInt(9, re_level);
pstmt.setString(10, article.getContent());
pstmt.setString(11, article.getIp());
/hbbaek.m for add filename
pstmt.setString(12, article.getFileName());
pstmt.executeUpdate();
```

입력받은 값들과 변경한 값들을 바탕으로 'Boarde' 테이블에 입력해준다.

환경	개발환경	서비스 그룹	화면처리	페이지	57~61
서비스 명	R-B-003				
설명	<p>'BoardServlet.java'의 'content.do'에서 int 타입의 변수 num 에 request 한 num 값을 형변환으로 넣고 이 값을 바탕으로 'BoardDBBean.java'의 'getArticel' 메소드를 호출해 BoardDateBean 타입의 변수 article 에 값을 넣어 'content.jsp'로 전달한다.</p> <p>'content.jsp'에서 전달받은 article 값을 바탕으로 글의 내용을 보고 'updateForm.jsp'로 article 의 num 값과 pageNum 을 전달한다.</p> <p>'updateForm.jsp'에서 기존에 쓰여있던 글의 내용을 바탕으로 수정할 수 있도록 'BoardDBBean.java'의 'updateGetArticle' 메소드를 request 한 num 값을 바탕으로 BoardDataBean 타입의 변수 article 에 넣어서 보여주고 수정된 값은 'BoardServlet.java'의 'updatePro.do'로 넘긴다.</p> <p>'updatePro.jsp'에서 받아온 값들을 바탕으로 BoardDataBean 타입의 변수 art 에 vo 로 값을 넣고 'BoardDBBean.java'의 'updateArticle' 메소드를 호출하고 그 결과를 int 타입의 변수 check 에 담아 'updatePro.jsp'로 전달한다.</p>				
content.jsp 코드설명	<pre> <script type="text/javascript"> function download(filename) { document.downloadFrm.filename.value=filename; document.downloadFrm.submit(); } </script> 첨부파일 다운로드를 위한 함수를 javascript 로 작성하였다. <% int num = (int)request.getAttribute("num"); String pageNum = (String)request.getAttribute("pageNum"); BoardDataBean article = (BoardDataBean)request.getAttribute("vo"); int ref=article.getRef(); int re_step=article.getRe_step(); int re_level=article.getRe_level(); SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm"); %> </pre>				
updateForm.jsp 코드설명	<pre> <% int num = Integer.parseInt(request.getParameter("num")); String pageNum = request.getParameter("pageNum"); try{ BoardDBBean dbPro = BoardDBBean.getInstance(); BoardDataBean article = dbPro.updateGetArticle(num); } %> </pre>				

BoardServlet.java
코드설명

- 글 내용 content.do

```
if (action.equals("content.do")) {  
    int num = Integer.parseInt(request.getParameter("num"));  
    String pageNum = request.getParameter("pageNum");  
    try{  
        BoardDBBean dbPro = BoardDBBean.getInstance();  
        BoardDataBean article = dbPro.getArticle(num);  
        request.setAttribute("vo", article);  
    } catch(Exception e) {  
        e.printStackTrace();  
    }  
    request.setAttribute("num", num);  
    request.setAttribute("pageNum", pageNum);  
    viewPage = "content.jsp";  
}
```

'list.jsp'에서 전달한 num 을 request 받아 'BoardDBBean.java'의 'getArticle' 메소드를 호출하고 이 값을 vo 라는 변수에 담아 num, pageNum 과 함께 'content.jsp'로 전달한다.

- 글 수정 upatePro.do

```
String pageNum = request.getParameter("pageNum");  
int check = 0;
```

글 수정 결과값을 담을 int 타입의 변수 check 를 선언한다.

	<pre> BoardDataBean art = new BoardDataBean(); art.setNum(Integer.parseInt(multi.getParameter("num"))); art.setSubject(multi.getParameter("subject")); art.setContent(multi.getParameter("content")); art.setPasswd(multi.getParameter("passwd")); // hbbaek.a for file upload String cf = art.getFileName(); if(cf!=null && !cf.equals("")) { }else { art.setFileName(filename); } BoardDBBean dbPro = BoardDBBean.getInstance(); check = dbPro.updateArticle(art); }catch(IOException ioe){ System.out.println(ioe); }catch(Exception ex){ System.out.println(ex); } request.setAttribute("pageNum", pageNum); request.setAttribute("check", check); viewPage = "updatePro.jsp"; </pre> <p>BoardDataBean 타입의 변수 art 에 vo 로 multipart request 로 'updateForm.jsp'에서 넘어온 값들을 넣고 'BoardDBBean.java'의 'updateArticle' 메소드를 호출하고 그 결과값을 check 에 넣어서 'updatePro.jsp'로 넘긴다.</p>
BoardDBBean.java a 코드설명	- 글 내용 getArticle

```
pstmt = conn.prepareStatement("update boarde set readcount = readcount + 1 where num = ?");
pstmt.setInt(1, num);
pstmt.executeUpdate();
```

```
pstmt = conn.prepareStatement("select * from boarde where num = ?");
pstmt.setInt(1, num);
rs = pstmt.executeQuery();
```

```
if(rs.next()) {
    article = new BoardDataBean();
    article.setNum(rs.getInt("num"));
    article.setWriter(rs.getString("writer"));
    article.setEmail(rs.getString("email"));
    article.setSubject(rs.getString("subject"));
    article.setPasswd(rs.getString("passwd"));
    article.setReg_date(rs.getTimestamp("reg_date"));
    article.setReadcount(rs.getInt("readcount"));
    article.setRef(rs.getInt("ref"));
    article.setRe_step(rs.getInt("re_step"));
    article.setRe_level(rs.getInt("re_level"));
    article.setContent(rs.getString("content"));
    article.setIp(rs.getString("ip"));
    article.setFileName(rs.getString("filename"));
```

글 내용을 보면 조회수가 올라가도록 'Boarde' 테이블의 readcount 를 +1 해주고 글의 내용을 select 문으로 글 번호 기준으로 들고와 BoardDataBean 타입의 변수에 vo 로 담아서 return 해준다.

- 글 수정 updateGetArticle

```
pstmt = conn.prepareStatement("select * from boarde where num = ?");
pstmt.setInt(1, num);
rs = pstmt.executeQuery();
```

```
if(rs.next()) {
    article = new BoardDataBean();
    article.setNum(rs.getInt("num"));
    article.setWriter(rs.getString("writer"));
    article.setEmail(rs.getString("email"));
    article.setSubject(rs.getString("subject"));
    article.setPasswd(rs.getString("passwd"));
    article.setReg_date(rs.getTimestamp("reg_date"));
    article.setReadcount(rs.getInt("readcount"));
    article.setRef(rs.getInt("ref"));
    article.setRe_step(rs.getInt("re_step"));
    article.setRe_level(rs.getInt("re_level"));
    article.setContent(rs.getString("content"));
    article.setIp(rs.getString("ip"));
    article.setFileName(rs.getString("filename"));
```

기존에 입력된 글 내용을 바탕으로 글 수정이 진행될 수 있도록 글 번호 num 을 기준으로 내용을 select 문으로 가져와 BoardDataBean 타입의 변수에

vo 로 넣어서 return 해준다.

- 글 수정 updateArticle

```
String dbpasswd = "";
//hbbaek.a for 글수정 시 비밀번호 같은 경우 방지
String dbwriter = "";
String sql = "";
int x = -1;

try {
    conn = getConnection();

    //hbbaek.m for 글수정 시 비밀번호 같은 경우 방지
    pstmt = conn.prepareStatement("select writer, passwd from boarde where num = ?");
    pstmt.setInt(1, article.getNum());
    rs = pstmt.executeQuery();

    if(rs.next()) {
        dbpasswd = rs.getString("passwd");
        //hbbaek.a for 글수정 시 비밀번호 같은 경우 방지
        dbwriter = rs.getString("writer");
        //hbbaek.m for 글수정 시 비밀번호 같은 경우 방지
        if((dbwriter.equals(article.getWriter()) && dbpasswd.equals(article.getPasswd()))) {
            sql = "update boarde set writer = ?, email = ?, subject = ?, passwd = ?, "
                + " content = ? where num = ?";
            pstmt = conn.prepareStatement(sql);

            pstmt.setString(1, article.getWriter());
            pstmt.setString(2, article.getEmail());
            pstmt.setString(3, article.getSubject());
            pstmt.setString(4, article.getPasswd());
            pstmt.setString(5, article.getContent());
            pstmt.setInt(6, article.getNum());
            pstmt.executeUpdate();

            x = 1;
        } else {
            x = 0;
        }
    }
}
```

'updateForm.jsp'에서 입력한 비밀번호와 등록된 비밀번호와 session 에 등록된 id 가 일치해야지 글 수정이 되도록 등록된 비밀번호와 아이디를 담을 수 있는 String 변수 dbpass 와 dbwriter 와 return 할 결과값 int 타입의 변수 x 를 선언한다.

글 번호 num 을 기준으로 select 한 내용에서 dbpass 와 dbwriter 를 담아주고 if 조건문으로 입력한 비밀번호와 session 의 아이디가 일치하는지 확인한 후 update 문이 진행될 수 있도록 하고 그 결과값 x 를 return 한다.

환경	개발환경	서비스 그룹	화면처리	페이지	62~63
서비스 명	R-B-004				
설명	<p>'deleteForm.do'에서 비밀번호 passwd 를 입력받아 'BoardServlet.java'의 'deletePro.do'로 전달한다.</p> <p>Request 로 전달받은 num 값과 pageNum, passwd, session 에 저장된 id 값을 바탕으로 'BoardDBBean.java'의 'deleteArticle' 메소드를 호출하고 그 결과값을 check 변수에 담아 deletePro.jsp 로 전달한다.</p>				
BoardServlet.java 코드설명	<pre> if(action.equals("deletePro.do")) { int num = Integer.parseInt(request.getParameter("num")); String pageNum = request.getParameter("pageNum"); String passwd = request.getParameter("passwd"); String id = (String)request.getSession().getAttribute("id"); System.out.println(id); int check = 0; try { BoardDBBean dbPro = BoardDBBean.getInstance(); check = dbPro.deleteArticle(num, passwd, id); } catch (Exception ex) { ex.printStackTrace(); } request.setAttribute("pageNum", pageNum); request.setAttribute("check", check); viewPage = "deletePro.jsp"; } </pre> <p>'Boarde'테이블의 delete 문을 진행하기 위해 필요한 글 번호와 입력한 비밀번호, session 의 id 를 request 로 String 변수에 담아 'BoardDBBean.java'의 'deleteArticle' 메소드를 호출하고 그 결과값을 check 라는 변수에 담아 'deletePro.jsp'로 전달한다.</p>				

BoardDBBean.java
a 코드설명

```
public int deleteArticle(int num, String passwd, String id) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    String dbpasswd = "";
    //hbbaek.a for 삭제 시 비밀번호 같은 경우 방지
    String dbwriter = "";
    int x = -1;

    try {
        conn = getConnection();

        //hbbaek.m for 글삭제 시 비밀번호 같은 경우 방지
        pstmt = conn.prepareStatement("select writer, passwd from boarde where num = ?");
        pstmt.setInt(1, num);
        rs = pstmt.executeQuery();

        if(rs.next()) {
            dbpasswd = rs.getString("passwd");
            //hbbaek.a for 삭제 시 비밀번호 같은 경우 방지
            dbwriter = rs.getString("writer");
            System.out.println(dbwriter);

            //hbbaek.m for 삭제 시 비밀번호 같은 경우 방지
            if((dbwriter.equals(id) && dbpasswd.equals(passwd)) || (id.equals("admin") && passwd.equals("admin"))) {
                pstmt = conn.prepareStatement("delete from boarde where num = ?");
                pstmt.setInt(1, num);
                pstmt.executeUpdate();
                x = 1;
            } else {
                x = 0;
            }
        }
    }
}
```

'deleteForm.jsp'에서 입력한 비밀번호와 session 의 아이디가 'Boarde' 테이블에 등록된 비밀번호와 글쓴이가 일치해야 삭제될 수 있도록한다.
다만 관리자 계정 'admin'이라면 일치하지 않아도 삭제할 수 있도록 if 조건문을 이용해 delete 문을 진행하고 결과값은 int 타입의 변수 x 에 담아 'deletePro.jsp'로 return 한다.

C. 화면 관련 및 파일 다운로드 소스

i. Color.jsp

```
<%  
    String bodyback_c="#FFF0B5";  
    String value_c="#FFE271";  
%>
```

배경과 테이블의 색상을 지정한다.

ii. Index.jsp

```
<script>  
    location.href = 'login.du';  
</script>
```

index 화면 호출 시 바로 login 을 진행할 수 있도록 login 페이지로 넘겨준다.

iii. downloadPro.jsp

```

<%!
public static String con(String s) {
    String str = null;
    try {
        str = new String(s.getBytes("8859_1"), "ksc5601");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return str;
}

%>
<%
try{
    String SAVEFOLDER = request.getSession().getServletContext().getRealPath("/fileSave");
    String filename = request.getParameter("filename");
    File file = new File(con(SAVEFOLDER + File.separator+filename)); //hbbaek.m for 경로
    byte b[] = new byte[(int)file.length()];
    response.setHeader("Accept-Ranges", "bytes");
    String strClient = request.getHeader("User-Agent");
    if(strClient.indexOf("MSIE6.0") != -1){
        response.setContentType("application/smnet;charset=euc-kr");
        response.setHeader("Content-Disposition", "filename="+filename+");");
    }else{
        response.setContentType("application/smnet;charset=euc-kr");
        response.setHeader("Content-Disposition", "attachment;filename="+filename+");");
    }
    out.clear();
    out = pageContext.pushBody();
    if(file.isFile()){
        BufferedInputStream fin = new BufferedInputStream(new FileInputStream(file));
        BufferedOutputStream outs = new BufferedOutputStream(response.getOutputStream());
        int read=0;
        while((read=fin.read(b)) != -1){
            outs.write(b, 0, read);
        }
        outs.close();
        fin.close();
    }
}
}

```

Cos.jar 라이브러리를 이용해 파일 다운로드를 구현하였다.

iv. Nav.jsp

```

<nav>
  <ul>
    <li>
      <a href="list.do">글목록</a>
    </li>
    <li>
      <a href="writeForm.do">글쓰기</a>
    </li>
    <li>
      <a href="main.do">회원정보</a>
    </li>
    <li>
      <a href="login.do">로그아웃</a>
    </li>
  </ul>
</nav>

```

로그인 이후 화면 상단에 기본 메뉴들을 구현하기 위해 별도의 nav 파일을 만들었다.

```

<style type="text/css">
  /* 기본 설정 */
  a{text-decoration:none; color:#000000;}
  a:hover{color:#ff0000;}

  /* nav tag */
  nav ul{padding-top:10px;}
  nav ul li {
    display:inline;
    border-left:1px solid #999;
    font:bold 12px Dotum;
    padding:0 10px;
  }
  nav ul li:first-child{border-left:none;}
</style>

```

Nav 파일의 css 는 위와 같이 설정하였다.

v. Style.css

```
» body {  
    font-size: 9pt;  
    text-align: center;  
}  
  
» table {  
    font-size: 9pt;  
    text-align: center;  
    border: 1pt solid orange;  
}  
  
» select {  
    font-size: 9pt;  
}  
  
» form {  
    font-size: 9pt;  
}  
  
» p{  
    font-weight: 700;  
}
```

기본적으로 많이 사용하는 html 태그에 관하여 css 를 별도의 style 파일로 관리하였다.

[배포환경]

Apache tomcat v8.5 이상

IDE : Eclipse EE 4.0 이상

DBMS : Oracle Developer 18 이상

Oracle Express Edition 11 이상

JDK : jdk 1.8 이상

cos.jar (파일 업로드, 다운로드 라이브러리)

ch18.war

형상관리 : git, github <https://github.com/hanbyeolbaek/notice-board>

[Contact Us]

Name : 백한별 (Baek Han Byeol)

Email : yeongeun0765@gmail.com

Tel : 010-2944-6756