

JSP게시판 프로젝트

Java Server Pages

제작자: 김동민

JSP Communication Bulletin Board

CONTENTS

01

Intro

요구 사항

설계 단계

02

개발환경

개발 일정

개발 환경

03

DB 설계

Member DB

Article DB

Member
History DB

04

페이지 설계

메인 페이지

로그인 페이지

회원가입 페이지

정보수정 페이지

게시판 페이지

글 쓰기 페이지

글 보기 페이지

05

화면 구현 및 테스트

공통 클래스

테스트 케이스

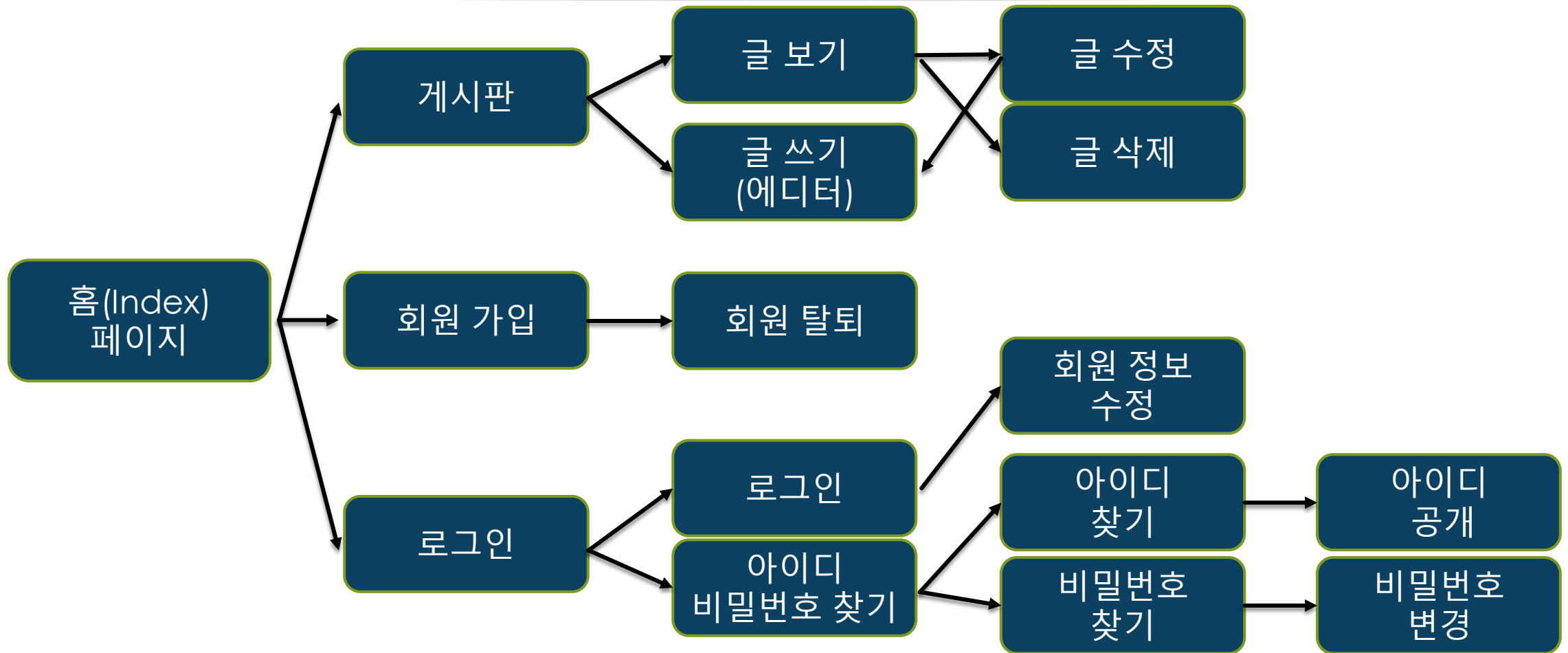
1-1(1) 요구 사항 - 회원

- 로그인 : 아이디 비밀번호를 이용한 로그인 기능
- 로그아웃 : 30분이 지나면 자동 로그아웃 및 버튼클릭한번에 로그아웃되는 기능
- 회원가입 : 이름, 아이디, 비밀번호, 나이, 휴대폰번호, 이메일 등을 기입하는 회원가입 기능
- 아이디 찾기 : 이름으로 본인의 아이디를 찾는 기능
- 비밀번호 찾기 : 이름과 아이디로 비밀번호를 변경 가능한 기능
- 회원정보수정 : 본인의 아이디 외에 모든 정보를 수정 가능한 기능
- 회원탈퇴 : 회원을 탈퇴 할 수 있는 기능

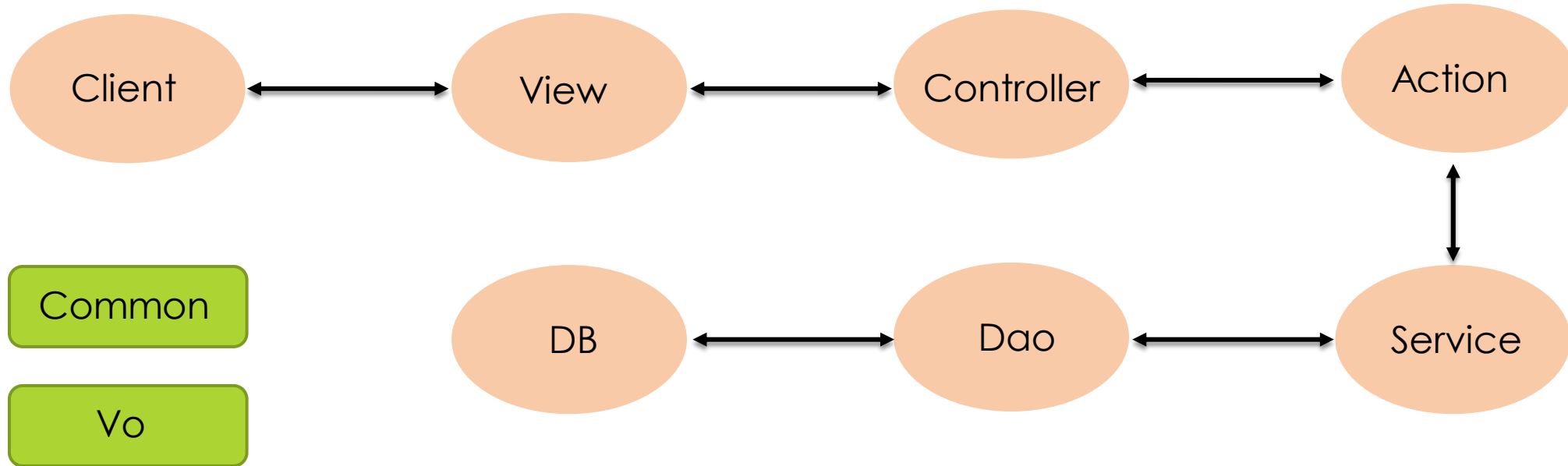
1-1(2) 요구 사항 - 게시판

- 게시판 : 글 제목과 작성일 작성자를 볼 수 있는 화면
- Pagination : 10개씩 끊어서 최신 순으로 앞뒤로 이동 가능한 기능
- 글 쓰기 : 에디터를 이용해 글씨 크기 폰트 사진첨부등의 기능
- 글 수정 : 글을 쓴 아이디로 로그인시 글을 수정 가능한 기능
- 글 삭제 : 글을 쓴 아이디로 로그인시 글을 삭제 가능한 기능
- 글 검색 : 글의 제목, 내용 등의 특정한 단어로 글을 찾을 수 있는 기능

1-2(1) 뷰 설계 단계



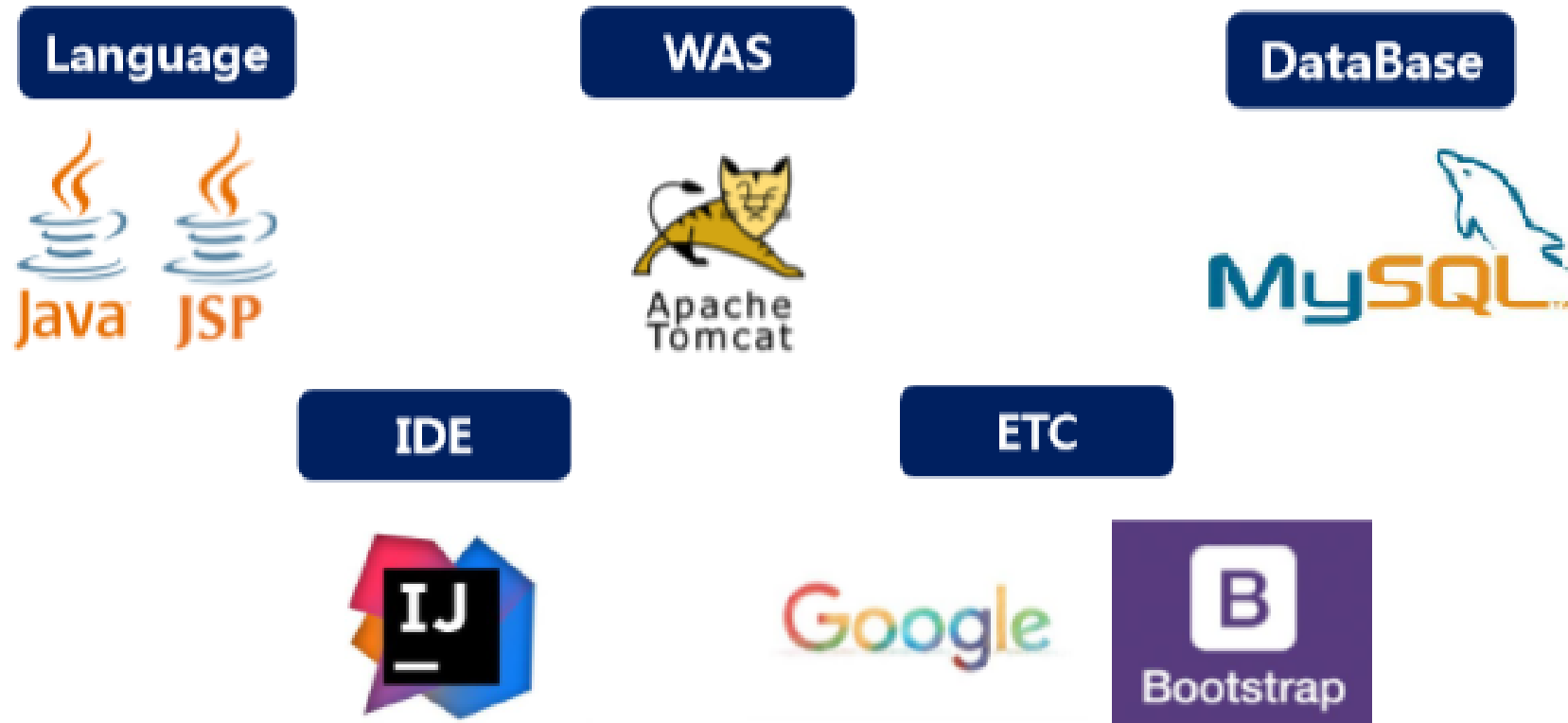
1-2(2) 구조 설계 단계



2-1 개발 일정

10월	9일	10일	11일	12일	12일	14일	15일	16일	17일	18일	19일	20일	21일	22일
요구사항 분석														
		DB 설계												
			화면 설계											
					화면 구현									
												통합 테스트		

2-2 개발 환경



3-1 회원 DB

The screenshot displays the MySQL Enterprise Workbench interface. On the left, the 'SCHEMAS' tree shows the 'kb' database selected, with the 'member' table highlighted under the 'Tables' folder. The main window shows the 'member' table structure for the 'kb' schema. The table has the following columns:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
sq	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
id	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
pwd	VARCHAR(200)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
ign_fl	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
leave_fl	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
email	VARCHAR(200)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
phone	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
name	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
age	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Below the table structure, there are fields for 'Column Name', 'Data Type', 'Charset/Collation', 'Default', and 'Comments'. The 'Storage' section includes options for 'Virtual', 'Stored', 'Primary Key', 'Not Null', 'Unique', 'Binary', 'Unsigned', 'Zero Fill', 'Auto Increment', and 'Generated'.

3-2 게시판 DB

The screenshot displays a database management interface. On the left, a 'Navigator' pane shows a tree of schemas, with 'kb1' selected. The main area shows the 'board' table structure. The table has the following columns:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
mb_sq	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
num	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
subject	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
content	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
hit	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
wdate	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP
udate	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
ddate	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Below the table structure, there are fields for 'Column Name', 'Data Type', 'Charset/Collation', 'Default', and 'Storage' options. The 'Storage' options include 'Virtual', 'Stored', 'Primary Key', 'Not Null', 'Unique', 'Binary', 'Unsigned', 'Zero Fill', 'Auto Increment', and 'Generated'.

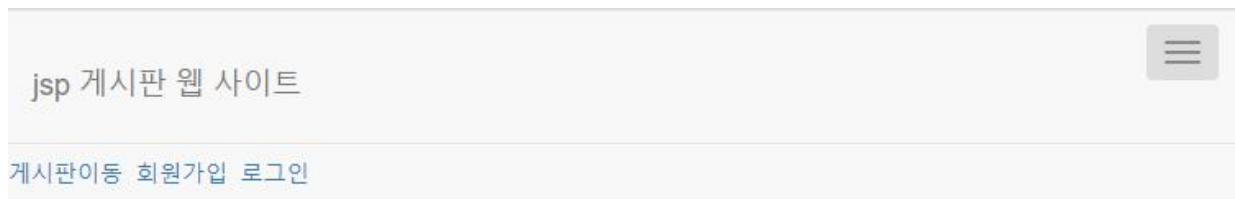
3-3 회원 History DB

The screenshot displays the MySQL Enterprise Workbench interface. On the left, the 'Navigator' pane shows the 'SCHEMAS' tree with 'kb' selected, and 'member_history' under 'member'. The main area shows the 'Table Name: member_history' in the 'Schema: kb' with 'Charset/Collation: utf8mb4' and 'Engine: InnoDB'. Below this is a table of columns:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
sq	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
mb_sq	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
evt_type	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dtm	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP

At the bottom, there are fields for 'Column Name:', 'Data Type:', 'Charset/Collation:', 'Default:', 'Storage:' (Virtual/Stored), 'Primary Key', 'Not Null', 'Unique', 'Binary', 'Unsigned', 'Zero Fill', 'Auto Increment', and 'Generated'. The 'Columns' tab is selected at the bottom of the main area.

4-1 메인 페이지



```
@WebServlet("/index.jsp")
public class HomeController extends HttpServlet {
    protected void doProcess
        (HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        ActionForward forward = null;
        Action action = new HomeAction();
        try {
            forward = action.execute(request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
        if (forward != null) {
            if (forward.isRedirect()) {
                response.sendRedirect(forward.getPath());
            } else {
                RequestDispatcher dispatcher
                    = request.getRequestDispatcher(forward.getPath());
                dispatcher.forward(request, response);
            }
        }
    }
}
```

4-2 로그인 페이지

jsp 게시판 웹 사이트

로그인화면

아이디

비밀번호

로그인

```
String id = request.getParameter( s: "id");
String pwd = request.getParameter( s: "pwd");
if (id == null || id.equals(""))
    || pwd == null || pwd.equals("")) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('잘못된 접근입니다.');
```

4-3(1) 회원가입 페이지

jsp 게시판 웹 사이트

회원가입

아이디

중복확인

비밀번호

비밀번호확인

이름

나이

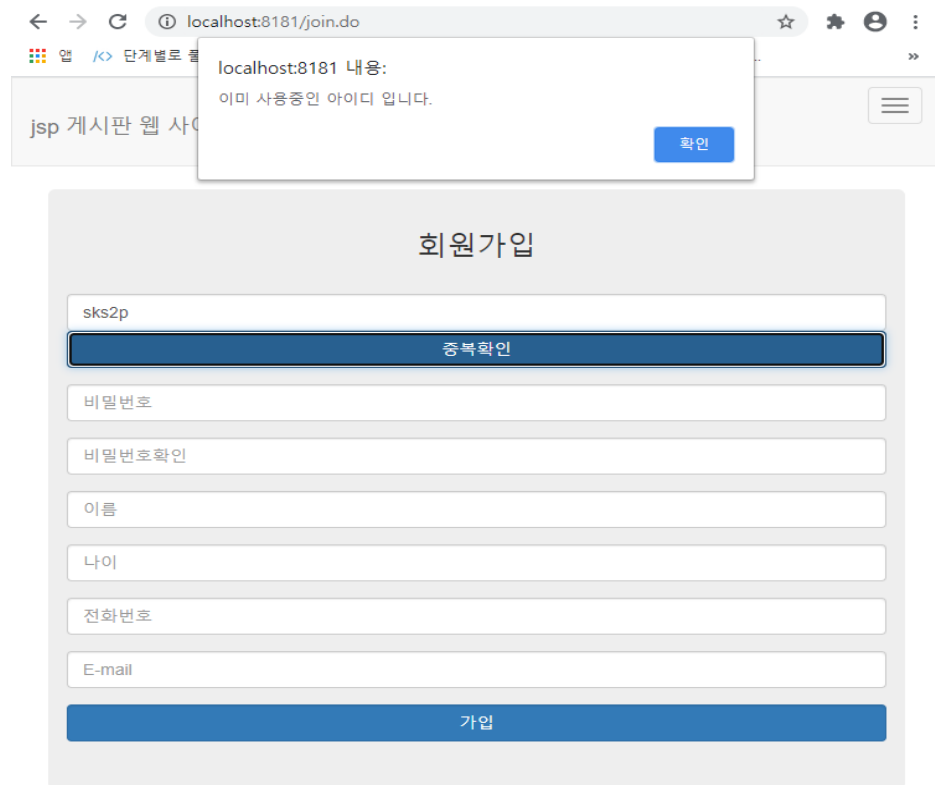
전화번호

E-mail

가입

```
String id = request.getParameter( s: "id");
String name = request.getParameter( s: "name");
String age = request.getParameter( s: "age");
String phone = request.getParameter( s: "phone");
String email = request.getParameter( s: "email");
String pwd = request.getParameter( s: "pwd");
String pwd_confirm = request.getParameter( s: "pwd_confirm");
if (id == null || id.equals(""))
    || !RegExp.checkString(MEMBER_ID, id)
    || pwd == null || pwd.equals("")
    || !RegExp.checkString(MEMBER_PWD, pwd)) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('잘못된 접근입니다.');
```

4-3(2) 회원가입 페이지



The screenshot shows a web browser at `localhost:8181/join.do`. A modal alert box is displayed with the text: "localhost:8181 내용: 이미 사용중인 아이디 입니다." (localhost:8181 content: The ID is already in use). Below the alert is a registration form titled "회원가입" (Sign Up). The form contains the following fields: a text input with "sks2p", a "중복확인" (Check Duplicate) button, a "비밀번호" (Password) field, a "비밀번호확인" (Confirm Password) field, "이름" (Name), "나이" (Age), "전화번호" (Phone Number), and "E-mail" fields. A "가입" (Sign Up) button is at the bottom.

```
function checkId() {  
    var id = $('#id').val();  
    $.ajax({  
        url: "/checkid.ajax"  
        , type: "post"  
        , data: {  
            id: id  
        }  
        , dataType: "json"  
        , error: function () {  
            alert("통신 실패");  
        }  
        , success: function (data) {  
            if (data.id == "true") {  
                // 중복된 아이디 없음  
                isChecked = true;  
                alert("사용가능한 아이디 입니다.");  
            } else {  
                // 중복된 아이디 있음  
                alert("이미 사용중인 아이디 입니다.");  
            }  
        }  
    });  
}
```

Ajax를 이용한 아이디 중복 확인

4-4 회원 정보 수정

jsp 게시판 웹 사이트

회원 정보 수정

sks2p

비밀번호

비밀번호확인

이름

나이

전화번호

E-mail

수정

```
LoginManager lm = LoginManager.getInstance();
String id = lm.getMemberId(request.getSession());
if(id == null) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('로그인이 필요한 서비스 입니다.');" + location.href = '/login.do'; "</script>");
    out.close();
    return null;
}
String pwd = request.getParameter("s:pwd");
String pwd_confirm = request.getParameter("s:pwd_confirm");
if (pwd == null || pwd.equals("")
    || !RegExp.checkString(MEMBER_PWD, pwd)) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('잘못된 접근입니다.');" + location.href = '/'; "</script>");
    out.close();
    return null;
}
if (!pwd.equals(pwd_confirm)) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('잘못된 접근입니다.');" + location.href = '/'; "</script>");
    out.close();
    return null;
}
```


4-5 게시판 페이지

jsp 게시판 웹 사이트

번호	제목	조회수	작성자	작성일
50	asdasd	32	sks2p	2020-10-14 16:28:54
49	a	32	sks2p	2020-10-12 15:14:02
48	a	9	sks2p	2020-10-12 15:13:59
47	a	4	sks2p	2020-10-12 15:13:56
46	a	4	sks2p	2020-10-12 15:13:50
45	a	2	sks2p	2020-10-12 15:13:47
44	a	0	sks2p	2020-10-12 15:13:44
43	a	0	sks2p	2020-10-12 15:13:41
42	a	1	sks2p	2020-10-12 15:13:38
41	a	0	sks2p	2020-10-12 15:13:35

< 1 2 3 4 5 > 전체 검색

글쓰기

```
String pageNum = request.getParameter("pn");
if (pageNum == null)
    !ReqExp.checkString(IS_NUMBER, pageNum) {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<script>alert('잘못된 접근입니다.');" + location.href + "/" + "</script>");
        out.close();
        return null;
    }
int page = Integer.parseInt(pageNum);
if (page < 1) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('처음 페이지입니다.');" + location.href + "/list.do" + "</script>");
    out.close();
    return null;
}
String filter = request.getParameter("filter");
String keyword = request.getParameter("keyword");
String query = "";
if (filter == null || filter.equals("")) {
    filter = "all";
}
if (keyword != null && !keyword.equals("")) {
    query = makeSearchQuery(filter, keyword);
}
BoardService service = new BoardService();
Pagination pagination = new Pagination(page, service.getArticleCount(query));
if (page > pagination.getTotalPageCount()) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>location.href='/list.do?pn=" + pagination.getTotalPageCount() + "';</script>");
    out.close();
    return null;
}
```

4-6 글 쓰기 페이지

jsp 게시판 웹 사이트

게시판 글쓰기

글 제목

사진 외부연결주소

HTML

맑은고딕 10pt

가 간 가 과 다 라

URL

등록

```
<div class="container">
  <div class="row">
    <form id="editorForm" action="/register.do" method="post">
      <table class="table table-striped" style="...">
        <thead>
          <tr>
            <th colspan="2" style="...">게시판 글쓰기</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>
              <input type="text" class="form-control" placeholder="글 제목" name="subject"
                id="subject" maxlength="100">
            </td>
            <td>
              <div style="height: 350px;">
                <jsp:include page="/editor/editorSkinForRegister.jsp" flush="false"/>
              </div>
            </td>
          </tr>
        </tbody>
      </table>
    </form>
    <input type="button" class="btn btn-primary pull-right" onclick="checkData()" value="등록">
  </div>
</div>
```

4-7 글 보기 페이지

jsp 게시판 웹 사이트 

게시판 글보기	
제목	누구나 그렇게 서른이 된다
작성자	sks2p
작성일시	2020-11-02 15:16:26
내용	<p>세상에 당연한건 없었다. 고된 시간을 견뎌낸 꽃이라 해서 모두가 제때 열매를 맺는 것은 아니었으니까.</p> <p>조금 느리고 서투르면 어때 우리의 서른은, 아직 피어나는 중인걸.</p> <p>편채원/ 누구나 그렇게 서른이 된다.</p>

목록

수정 삭제

```
String num = request.getParameter("s:num");
if (num == null || num.equals("")) {
    if (!RegExp.checkString(ARTICLE_NUM, num)) {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<script>alert('잘못된 접근입니다.');" + location.href + "/" + "</script>");
        out.close();
        return null;
    }
}
int buff = Integer.parseInt(num);
if (buff <= 0) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('잘못된 접근입니다.');" + location.href + "/" + "</script>");
    out.close();
    return null;
}
BoardService service = new BoardService();
ArticleVo vo = service.getArticle(buff);
if (vo == null) {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('잘못된 접근입니다.');" + location.href + "/" + "</script>");
    out.close();
    return null;
}
```

5-1(1) 공통 클래스

```
public class BCrypt {
    // BCrypt parameters
    private static final int GENSALT_DEFAULT_LOG2_ROUNDS = 10;
    private static final int BCRYPT_SALT_LEN = 16;

    // Blowfish parameters
    private static final int BLOWFISH_NUM_ROUNDS = 16;

    // Initial contents of key schedule
    private static final int P_orig[] = {
        0x243f6a88, 0x85a308d3, 0x13198a2e, 0x03707344,
        0xa4093822, 0x299f31d0, 0x082efa98, 0xec4e6c89,
        0x452821e6, 0x38d01377, 0xbe5466cf, 0x34e90c6c,
        0xc0ac29b7, 0xc97c50dd, 0x3f84d5b5, 0xb5470917,
        0x9216d5d9, 0x8979fb1b
    };

    private static final int S_orig[] = {
        0xd1310ba6, 0x98dfb5ac, 0x2ffdf72db, 0xd01adfb7,
        0xb8e1afed, 0x6a267e96, 0xba7c9045, 0xf12c7f99,
        0x24a19947, 0xb3916cf7, 0x0801f2e2, 0x858efc16,
        0x636920d8, 0x71574e69, 0xa458fea3, 0xf4933d7e,
    };
}
```

암호화 클래스

```
public class RegExp {
    public static final int ARTICLE_NUM = 0;
    public static final int ARTICLE_SUBJECT = 1;
    public static final int ARTICLE_CONTENT = 2;
    public static final int MEMBER_ID = 3;
    public static final int MEMBER_PWD = 4;
    public static final int IS_NUMBER = 5;

    public static final String EXP_ARTICLE_NUM = "[0-9]*$";
    public static final String EXP_ARTICLE_SUBJECT = "^.{1,100}$";
    public static final String EXP_ARTICLE_CONTENT = "^.{1,65535}$";
    public static final String EXP_MEMBER_ID = "^([a-z0-9]){4,20}$";
    public static final String EXP_MEMBER_PWD = "^.{4,30}$";
    public static final String EXP_IS_NUMBER = "[0-9]*$";

    public static boolean checkString(int type, String data) {
        boolean result = false;
        switch (type) {
            case ARTICLE_NUM:
                result = Pattern.matches(EXP_ARTICLE_NUM, data);
                break;
            case ARTICLE_SUBJECT:
                result = Pattern.matches(EXP_ARTICLE_SUBJECT, data);
                break;
            case ARTICLE_CONTENT:
                result = Pattern.matches(EXP_ARTICLE_CONTENT, data);
                break;
            case MEMBER_ID:
                result = Pattern.matches(EXP_MEMBER_ID, data);
                break;
            case MEMBER_PWD:
                result = Pattern.matches(EXP_MEMBER_PWD, data);
                break;
            case IS_NUMBER:
                result = Pattern.matches(EXP_IS_NUMBER, data);
                break;
        }
        return result;
    }
}
```

정규표현식 클래스

5-1(2) 공통 클래스

```
public class Pagenation {
    // 한페이지에 보여줄 글 개수
    private final int SHOW_ARTICLE_COUNT = 10;
    // 페이지 번호 묶음
    private final int PAGE_GROUP = 10;
    // 현재 페이지 번호
    private int nowPageNumber;
    // 글 총 개수
    private int totalArticleCount;
    // 시작 글 번호
    private int startArticleNumber;
    // 총 페이지 개수
    private int totalPageCount;
    // 시작 페이지, 끝 페이지
    private int startPage;
    private int endPage;

    public Pagenation(int nowPageNumber, int totalArticleCount) {
        this.nowPageNumber = nowPageNumber;
        this.startArticleNumber = (nowPageNumber - 1) * SHOW_ARTICLE_COUNT;
        this.totalArticleCount = totalArticleCount;
        this.totalPageCount = (int) Math.ceil(((double) totalArticleCount / SHOW_ARTICLE_COUNT));
        if (this.totalPageCount < 1) {
            this.totalPageCount = 1;
        }
        this.startPage
            = ((int) (((double) nowPageNumber / PAGE_GROUP + 0.9) - 1)) * PAGE_GROUP + 1;
        this.endPage = this.startPage + (PAGE_GROUP - 1);
        if (this.endPage > this.totalPageCount) {
            this.endPage = this.totalPageCount;
        }
    }
}
```

페이지네이션 클래스

```
public class LoginManager implements HttpSessionBindingListener {
    private static Hashtable loginUsers = new Hashtable();
    private LoginManager() { super(); }
    public static LoginManager getInstance() { return LoginManager.LazyHolder.INSTANCE; }
    private static class LazyHolder {
        private static final LoginManager INSTANCE = new LoginManager();
    }

    @Override
    public void valueBound(HttpSessionBindingEvent event) { loginUsers.put(event.getSession(), event.getName()); }

    @Override
    public void valueUnbound(HttpSessionBindingEvent event) {
        Action action = new MemberLogoutAction();
        ((MemberLogoutAction) action).logoutProc(getMemberId(event.getSession()));
        loginUsers.remove(event.getSession());
    }

    public void removeSession(String id) {
        Enumeration e = loginUsers.keys();
        HttpSession session = null;
        while(e.hasMoreElements()) {
            session = (HttpSession) e.nextElement();
            if (loginUsers.get(session).equals(id)) {
                session.invalidate();
            }
        }
    }
}
```

로그인매니저 클래스

5-2 테스트 케이스

	테스트	결과
회원가입	로그인이 되었을시 회원가입 버튼이 비활성화 되는가	로그인시 버튼이 비활성화 됨
	아이디 중복 확인이 올바르게 작동하는가	중복 확인이 올바르게 작동함
	아이디, 패스워드에 정규표현식이 올바르게 작동하는가	정규표현식이 작동함
	회원가입이 성공적으로 이루어 지는가	회원가입이 이루어짐
로그인	로그인이 성공적으로 이루어 지는가	로그인이 이루어짐
	아이디 패스워드 오류 시 알림이 작동하는가	오류 출력이 이루어짐
	이름으로 아이디 찾기를 할 수 있는가	아이디 찾기가 가능함
	이름과 아이디로 비밀번호 변경을 할 수 있는가	비밀번호 변경이 가능함
	회원정보 수정이 제대로 이루어지는가	회원정보 수정이 이루어짐

5-2 테스트 케이스

	테스트	결과
게시판	글이 최신 글부터 보이는가	최신 글부터 10개씩 화면에 출력됨
	게시판에 글번호 제목 조회수 작성자 작성일이 보이는가	게시판에서 확인 가능함
	글을 클릭하면 상세보기가 이루어지는가	상세보기가 이루어짐
	페이지네이션을 클릭하면 해당페이지로 이동하는가	해당페이지로 이동함
	찾는 글의 제목을 적고 검색을 클릭하면 검색이 되는가	검색이 이루어짐
	로그인이 안 된 상태로 글쓰기 클릭시 로그인안내가 되는가	로그인안내가 이루어짐
	본인의 글만 수정, 삭제가 가능한가	본인의 글만 수정, 삭제가능함