



게시판 제작 포트폴리오

쉬동민

목차

1. 자기소개

2. 게시판 설계

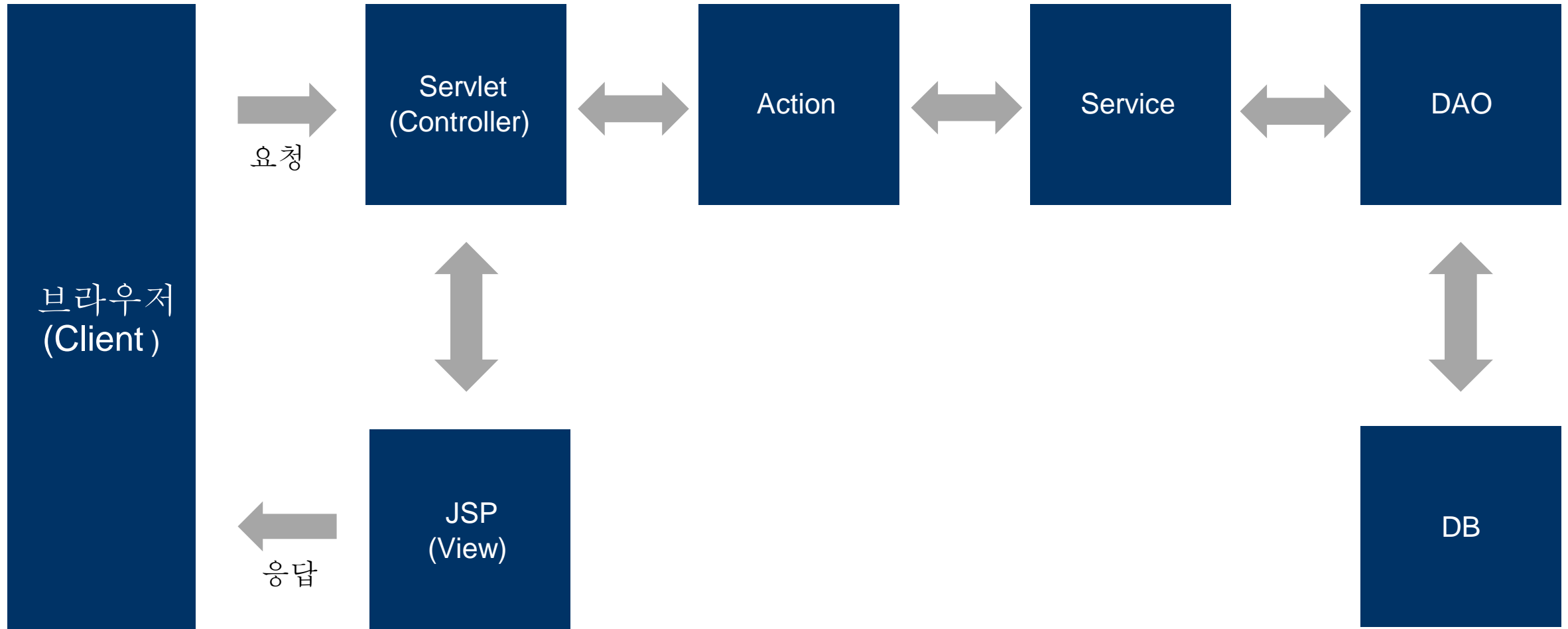
3. 주요기능

4. 테스트케이스

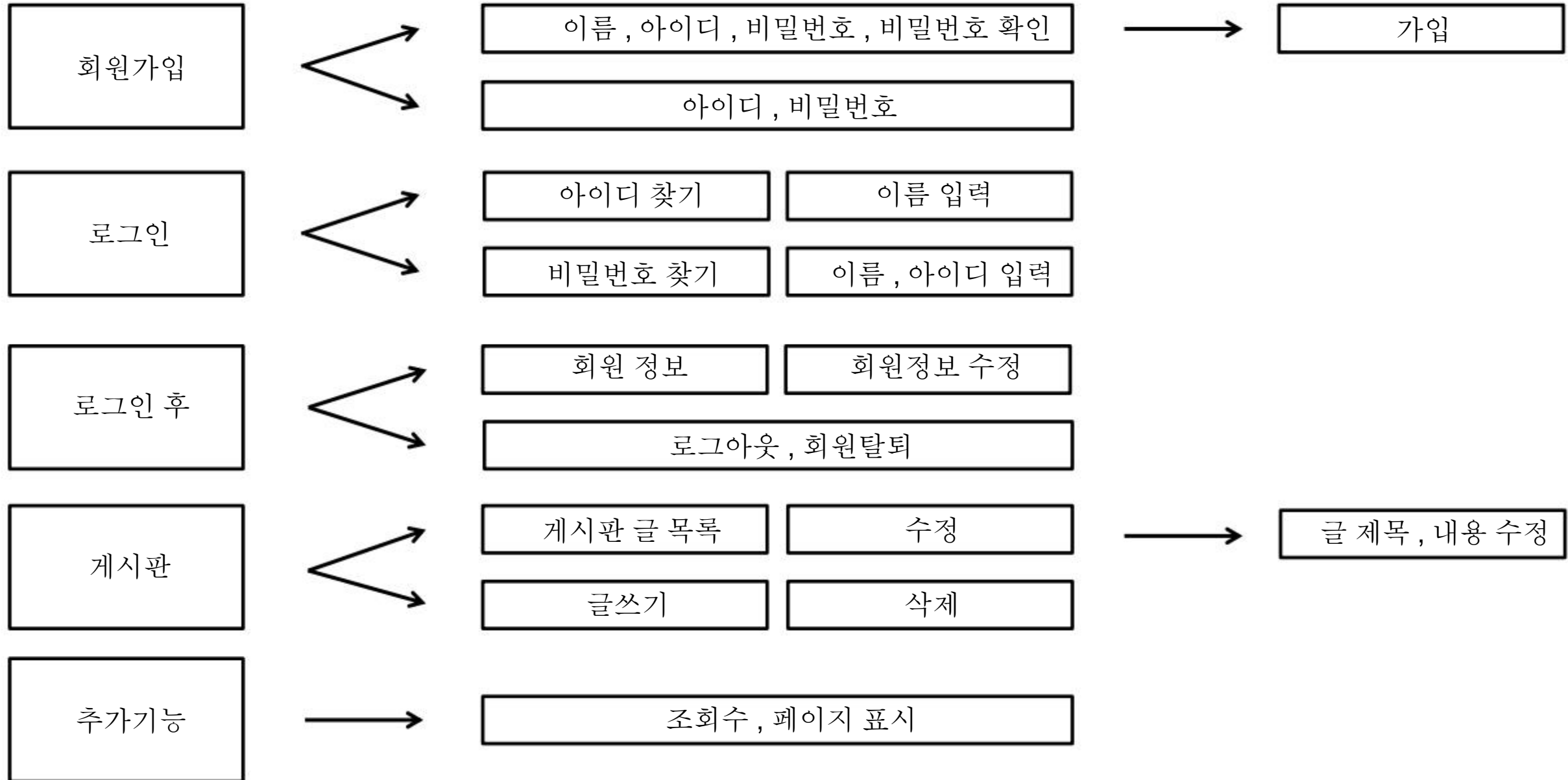
자기 소개

이름	
생일	
연락처	
메일	
학력	
취미	
특기	

게시판 설계 (MVC2 패턴)



게시판 설계 (세부 기능)



게시판 설계 (세부 기능 : 회원 가입)

← → ↻ ⓘ localhost:8181/memberJoin.do

이름

아이디

비밀번호

비밀번호확인

```
if (name == null || name.equals("") || !reg.isValid(MEMBER_NAME, name) ||
    id == null || id.equals("") || !reg.isValid(MEMBER_ID, id) ||
    pwd == null || pwd.equals("") || !reg.isValid(MEMBER_PASSWORD, pwd) ||
    confirmPwd == null || confirmPwd.equals("")) {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('잘못된 접근입니다.');
```

▶ 이름 , 아이디 , 비밀번호가 잘 들어갔는지 정규표현식으로 확인

게시판 설계 (세부 기능 : 로그인)

← → ↻ ⓘ localhost:8181/login.do

아이디 :

비밀번호 :

```
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.');
```

▶ DB 에 있는 회원정보와 같은지 확인 후 로그인

게시판 설계 (세부 기능 : 게시판 및 게시판 수정)

← → ↻ ⓘ localhost:8181/list.do?pn=1

번호	제목	작성자	작성일시	조회수
61	test9	admin	2020-11-06 09:41:34	1
60	test8	admin	2020-11-06 09:41:29	0
59	test7	admin	2020-11-06 09:41:23	1
58	test6	admin	2020-11-06 09:41:18	0
57	test5	admin	2020-11-06 09:41:13	0
56	test4	admin	2020-11-06 09:41:09	0
55	test3	admin	2020-11-06 09:41:05	4
54	test2	admin	2020-11-06 09:40:58	1
53	test1	admin	2020-11-06 09:40:50	1
51	111111	admin1	2020-11-05 09:25:22	2

≤ 1 2 3 4 5 > 글쓰기

localhost:8181 내용:

로그인이 필요한 서비스 입니다.

확인

▶ 글쓰기는 로그인 후 가능

← → ↻ ⓘ localhost:8181/modify.do?pn=1&sq=55

제목 test3

테스트 글

내용

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('잘못된 접근입니다. ');location.href='/login.do';</sc");
    out.close();
    return null;
}

String num = request.getParameter("sq");
RegExp regexp = new RegExp();
if (num == null || num.equals("") || !regexp.isValid(SEQUENCE_NUMBER, num)) {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>alert('잘못된 접근입니다. ');history.back();</script>");
    out.close();
    return null;
}
```

▶ 게시글 작성자와 아이디가 같아야 수정 및 삭제 가능

게시판 설계 (세부 기능 : 로그인 매니저)

```
7 public class LoginManager implements HttpSessionBindingListener {
8     private static Hashtable LoginUsers = new Hashtable();
9
10    private LoginManager() {
11        super();
12    }
13
14    public static LoginManager getInstance() {
15        return LoginManager.LazyHolder.INSTANCE;
16    }
17
18    private static class LazyHolder {
19        private static final LoginManager INSTANCE = new LoginManager();
20    }
21
22    @Override
23    public void valueBound(HttpSessionBindingEvent event) {
24        LoginUsers.put(event.getSession(), event.getName());
25    }
26
27    @Override
28    public void valueUnbound(HttpSessionBindingEvent event) {
29        // Action action = new MemberLogoutAction();
30        // ((MemberLogoutAction) action).logoutProc(getMemberId(event.getSession()));
31        LoginUsers.remove(event.getSession());
32    }
33
34    public void removeSession(String id) {
35        Enumeration e = LoginUsers.keys();
36        HttpSession session = null;
```

```
        Enumeration e = LoginUsers.keys();
        HttpSession session = null;
        while(e.hasMoreElements()) {
            session = (HttpSession) e.nextElement();
            if (LoginUsers.get(session).equals(id)) {
                session.invalidate();
            }
        }
    }

    public boolean isLogin(String sessionId) {
        boolean isLogin = false;
        Enumeration e = LoginUsers.keys();
        String key = "";
        while(e.hasMoreElements()) {
            key = (String) e.nextElement();
            if (sessionId.equals(key)) {
                isLogin = true;
            }
        }
        return isLogin;
    }

    public void setSession(HttpSession session, String id) {
        session.setAttribute(id, this);
    }

    public String getMemberId(HttpSession session) {
        return (String) LoginUsers.get(session);
    }
```

게시판 설계 (세부 기능 : 정규 표현식)

```
5 public class RegExp {
6     public static final int MEMBER_NAME = 0;
7     public static final int MEMBER_ID = 1;
8     public static final int MEMBER_PASSWORD = 2;
9     public static final int SEQUENCE_NUMBER = 3;
10
11     private final String EXP_MEMBER_NAME = "[가-힣a-z]{1,10}$";
12     private final String EXP_MEMBER_ID = "[a-z0-9]{4,20}$";
13     private final String EXP_MEMBER_PASSWORD = "^[.]{4,20}$";
14     private final String EXP_IS_NUMBER = "[1-9][0-9]*$";
15
16     public boolean isValid(int type, String data) {
17         boolean result = false;
18         switch (type) {
19             case MEMBER_NAME :
20                 result = Pattern.matches(EXP_MEMBER_NAME, data);
21                 break;
22             case MEMBER_ID :
23                 result = Pattern.matches(EXP_MEMBER_ID, data);
24                 break;
25             case MEMBER_PASSWORD :
26                 result = Pattern.matches(EXP_MEMBER_PASSWORD, data);
27                 break;
28             case SEQUENCE_NUMBER :
29                 result = Pattern.matches(EXP_IS_NUMBER, data);
30                 break;
31         }
32
33         return result;
34     }
35 }
```

게시판 테스트 케이스

분류	조건	결과
회원가입	회원가입후 DB 에저장되었습니까	○
로그인	로그인후회원정보,로그아웃,회원탈퇴버튼이있는페이지로이동합니까	○
로그아웃	로그아웃후첫페이지(로그인화면)로돌아갑니까	○
회원탈퇴	회원탈퇴하면 DB 에회원정보가삭제됩니다	○
게시판	게시판클릭하면게시판글목록으로이동합니까	○
글쓰기	글쓰기를하면게시판글목록에글이생성됩니다	○
글수정및삭제	글하나를클릭후수정및삭제가됩니다	○
	글작성자로그인확아이디가같아야수정및삭제가됩니다	○

END