

## 유원규2101151웹프로그래밍\_보고서04

- 개요
- 작업 진행
- 프로젝트 폴더 경로
- 프로젝트 수행 스크린 샷 및 설명
  - index.jsp
  - join.jsp
  - 로그인
  - MyInfoPage.jsp
  - 회원 정보 수정
- 회원 탈퇴 기능
  - 관리자 유저 조회 기능
- 관리자 유저 수정 기능
- 관리자 유저 삭제
- 수정한 소스코드
- MainController
  - User컨트롤러
- UserDAO
- user\_r
- user\_u
- user\_u2
- user\_d
- my\_page
- header
- 수정사항이 없는 파일
- 수행후기

# 유원규2101151웹프로그래밍\_보고서04

## 개요

22년 2학기에 MVC기반 개인 쇼핑몰 제작 결과물을 Spring으로 변경한다.

변경할때 FrontController를 Spring 형식의 컨트롤러로 변경하고 기존의 DAO, DTO, 각종Action은 수정하지 않는 방식으로 변경한다.

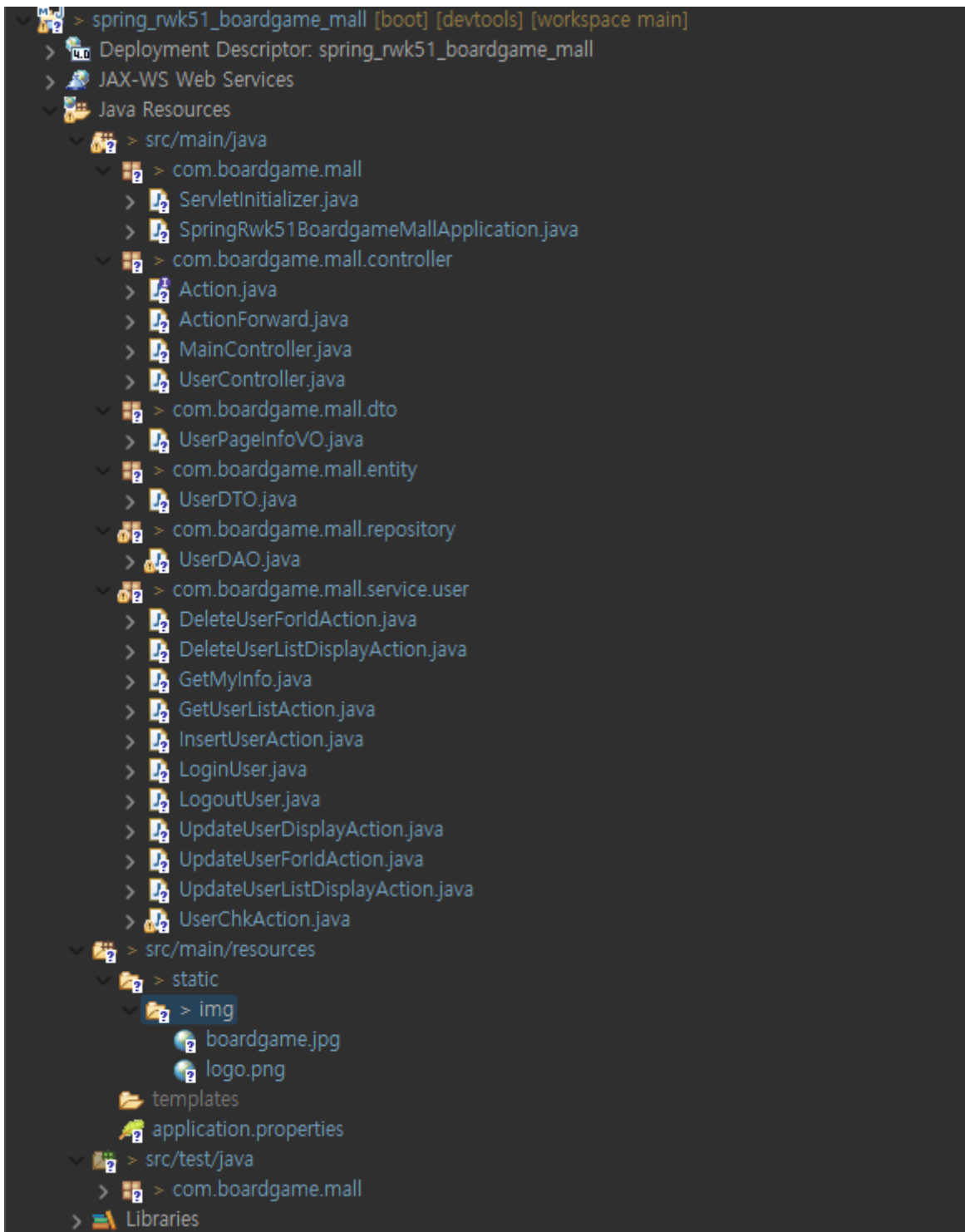
이유는 아직 Service, Repository, Entity, DTO 등에 대해 배우지 않았기 때문이다.

## 작업 진행

1. 상품CRUD
2. 회원CRUD
3. 장바구니

3개의 기능중 회원CRUD 및 로그인 로그아웃 기능 관리자의 회원관리(CRUD)까지 실행함.

## 프로젝트 폴더 경로



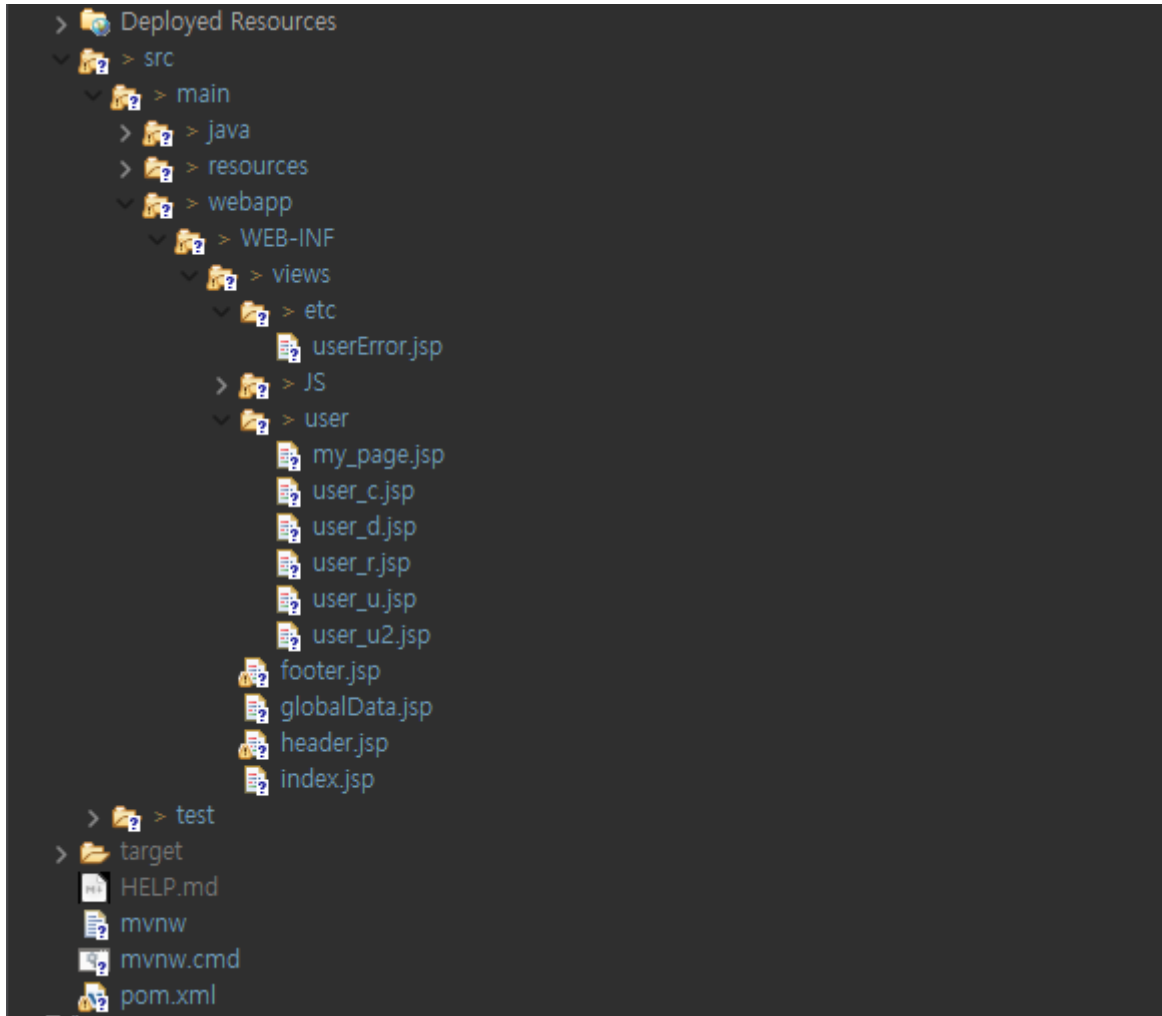
controller -> UserController가 존재함.

dto -> VO삽입

entity -> UserDAO가 들어가 있음.

repository -> UserDTO가 들어가있음

service -> 각종 action파일을 포함함



ppt 내용을 참고하여 view 경로를 설정함.

## 프로젝트 수행 스크린 샷 및 설명

---

**index.jsp**



MainController를 만들고 'localhost:8881'로 접속할 경우 메인 홈페이지로 이동하게함.

해당 페이지에서 로그인,회원가입 상품조회 등이 가능하나 해당 프로젝트의 경우 상품조회 및 상품게시판이  
동은 불가능함.

왼쪽이미지를 클릭하면 메인홈페이지로 이동함.

## join.jsp

## 회원가입

test333

test333

.....

.....

test333@naver.com

00011115555

1234567894567

서울 성동구 서울숲길 17

test333

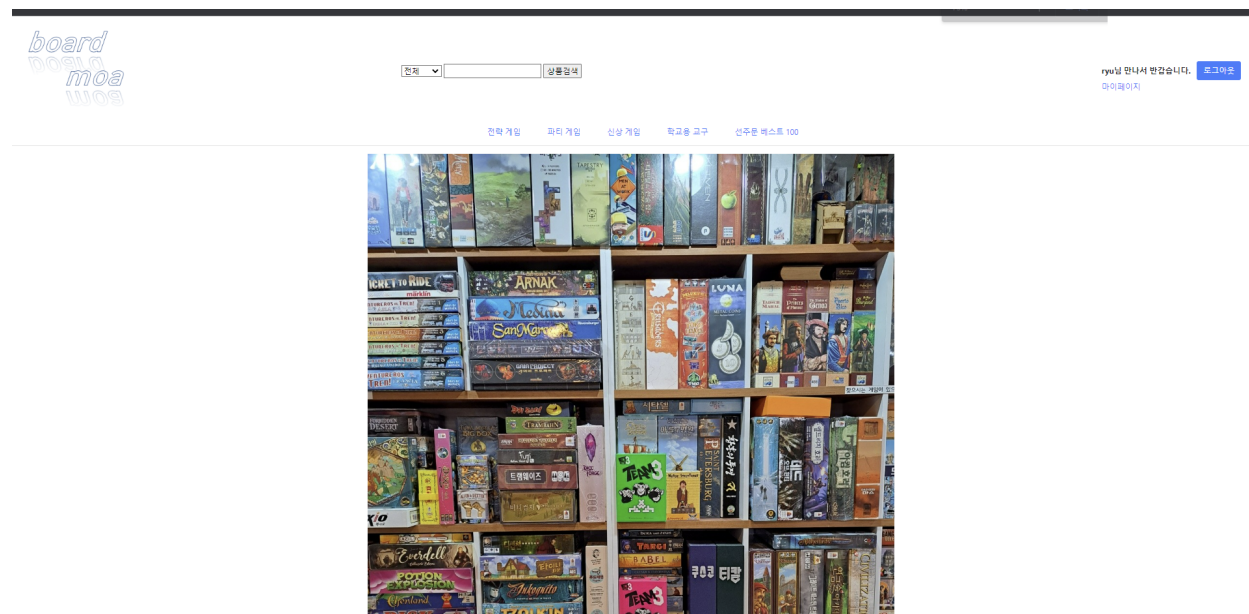
회원가입

메인홈페이지에서 회원가입을 클릭하면 해당페이지로 이동함.

회원가입을 성공하면 자동으로 메인홈페이지로 이동하고 실패할 경우 관리자에게 문의메시지가 뜬다.

추후에 해당 아이디를 통해 관리자가 수정삭제하는 모습을 시현할 예정

## 로그인



로그인을 성공시 메인홈페이지로 이동함.

기존에 존재하던 회원가입,로그인 , idpw 입력칸이 사라짐

사라진 영역에 유저닉네임, 마이페이지, 로그아웃이 나타남.

지금 로그인한계정은 관리자 계정임.

## MyInfoPage.jsp

The screenshot shows a web application interface for a user's profile page. At the top left is a logo with the text 'board', 'nosin', 'moa', and 'wos'. To the right of the logo is a search bar with a dropdown menu labeled '검색' and a button labeled '상품검색'. Further right is a notification area that says 'nyu님 만났서 반갑습니다.' with a '로그아웃' button and a link to '마이페이지'. Below the header is a navigation bar with links: '전학 거임', '과외 거임', '선상 거임', '학교를 그만', and '선주론 베스트 100'. The main content area is divided into three sections. The first section, titled '내 정보', contains user details: '아이디 : nyu', '이름 : nyu', '이메일 : nyu@naver.com', '휴대폰 : 01012341234', '주소 : admin', '상세 주소 : nyu', and '등급 : admin'. There are two green buttons, '회원탈퇴' and '회원수정', at the bottom of this section. The second section, titled '장바구니', is currently empty. The third section, titled '관리자 기능', contains a list of links: '회원정보조회', '회원정보삭제', '회원정보수정', and '상품등록'.

해당 유저는 관리자로 라서 관리자기능이 보이나 일반유저는 관리자 기능이 보이지 않음.

해당페이지에서 회원탈퇴, 회원수정,회원정보조회가 가능함.

장바구니는 임의로 존재하고 기존경로로 설정해뒀기에 제대로 작동안함.

관리자 기능에서 회원관리 CRUD가 가능함.

상품등록은 구현되지 않은 기능.

## 회원 정보 수정

## 정보수정

아이디

ryu

비밀번호

...

비밀번호 재확인

...

이름

ryu

이메일

ryu@naver.com

휴대폰

01012341234

admin

2023.04.06

등급

관리자



수정하기

마이페이지에서 회원수정을 클릭하면 해당 화면으로 이동함.

하얀색 칸이 바꿀 수 있는 것

관리자이므로 등급 변경이 가능함.

수정하기를 누르면 수정이됨.

## 내 정보

아이디 : ryu

이름 : ryu

이메일 : ryu@naver.com

휴대폰 : 01012341234

주소 : admin

상세 주소 : 2023.04.06

등급 : admin

회원탈퇴

회원수정

수정하기 버튼을 누른후 수정이 된 모습

성공적인 수정이 되면 메인홈페이지로 이동함.

##

## 회원 탈퇴 기능

- 회원탈퇴기능은 지금 시현중인 계정이 관리자 이므로 삭제하지않고 관리자 유저 삭제에서 시현 함.

## 관리자 유저 조회 기능



## 관리자 유저 조회

번호	아이디	비밀번호	이름	주민등록번호	이메일	휴대폰	주소	상세 주소	등급
1	ryu	ryu	ryu	123456-1234567	ryu@naver.com	01012341234	admin	2023.04.06	admin
11	won	won	won	123456-1234568	won@naver.com	01012349874	won	won	user
15	kyu	kyu	kyu	1234569874561	kyu@naver.com	01092461256	kyu	kyu	user
17	jjin	jjin	jjin	1	jjin@naver.com	01092875698	jjin	jjin	user
24	키키	aa	전종찬	1234561234563	aa@naver.com	01010101010	대전 동구 판교1길 3	1234	user
27	ddd	dddd	ddd	5001011234565	qwe@qwe	01012334567	서울 성북구 아리랑로 3	testsetset	user
28	wjdt12	wjdt12	김정수	9605041632546	wjdt12@naver.com	01012345678	대구 남구 경상길 1	남구길	user
32	test333	test333	test333	1234567894567	test333@naver.com	00011115555	서울 동대문구 서울시립대로 5	test333	user

관리자가 유저조회를 누르면 뜨는 페이지

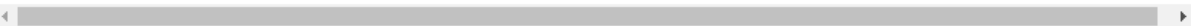
가입한 회원들의 정보를 볼 수 있음.

## 관리자 유저 수정 기능

## 관리자 유저 수정

번호	아이디	비밀번호	이름	주민등록번호	이메일	휴대폰	주소	상세 주소	등급	수정
1	ryu	ryu	ryu	123456-1234567	ryu@naver.com	01012341234	admin	2023.04.06	admin	수정
11	won	won	won	123456-1234568	won@naver.com	01012349874	won	won	user	수정
15	kyu	kyu	kyu	1234569874561	kyu@naver.com	01092461256	kyu	kyu	user	수정
17	jjin	jjin	jjin	1	jjin@naver.com	01092875698	jjin	jjin	user	수정
24	키키	aa	전종찬	1234561234563	aa@naver.com	01010101010	대전 동구 판교1 길 3	1234	user	수정
27	ddd	dddd	ddd	5001011234565	qwe@qwe	01012334567	서울 성북 구 아 리랑 로 3	testsetset	user	수정
28	wjdt12	wjdt12	김정수	9605041632546	wjdt12@naver.com	01012345678	대구 남구 경상 길 1	남구길	user	수정
32	test333	test333	test333	1234567894567	test333@naver.com	00011115555	서울 동대 문구 서울 시립 대로 5	test333	user	수정

[FIRST] [PRE][1][NXT] END



- 페이징 기능 첨부
- 해당 화면에서 원하는 유저를 수정가능함.
- 회원가입에서 보여준 test333 계정을 수정해보겠음.

## 정보수정

아이디

test333

비밀번호

\*\*\*\*\*

비밀번호 재확인

\*\*\*\*\*

이름

test333

이메일

test333@naver.com

휴대폰

00011115555

서울 동대문구 서울시립대로 5

test444

등급

일반사용자

수정하기

- 클릭한 유저의 정보가 자동으로 입력이 됨.
- 주소가 test333에서 test444로 수정해보겠음.

## 관리자 유저 삭제

## 관리자 유저 삭제

번호	아이디	비밀번호	이름	주민등록번호	이메일	휴대폰	주소	상세 주소	등급	삭제
1	ryu	ryu	ryu	123456-1234567	ryu@naver.com	01012341234	admin	2023.04.06	admin	삭제
11	won	won	won	123456-1234568	won@naver.com	01012349874	won	won	user	삭제
15	kyu	kyu	kyu	1234569874561	kyu@naver.com	01092461256	kyu	kyu	user	삭제
17	jjin	jjin	jjin	1	jjin@naver.com	01092875698	jjin	jjin	user	삭제
24	키키	aa	전종찬	1234561234563	aa@naver.com	01010101010	대전 동구 판교1 길 3	1234	user	삭제
27	ddd	dddd	ddd	5001011234565	qwe@qwe	01012334567	서울 성북 구 아 리랑 로 3	testsetset	user	삭제
28	wjdt12	wjdt12	김정수	9605041632546	wjdt12@naver.com	01012345678	대구 남구 경상 길 1	남구길	user	삭제
32	test333	test333	test333	1234567894567	test333@naver.com	00011115555	서울 동대 문구 서울 시립 대로 5	test444	user	삭제

[FIRST] [PRE][1][NXT] END

- 아까전에 수정한 아이디가 test333의 주소가 test444로 변경된게 확인이됨.
- 해당 계정을 삭제해보겠음.
- 삭제가 성공적으로 되면 메인홈페이지로 이동됨.

## 관리자 유저 삭제

번호	아이디	비밀번호	이름	주민등록번호	이메일	휴대폰	주소	상세 주소	등급	삭제
1	ryu	ryu	ryu	123456-1234567	ryu@naver.com	01012341234	admin	2023.04.06	admin	삭제
11	won	won	won	123456-1234568	won@naver.com	01012349874	won	won	user	삭제
15	kyu	kyu	kyu	1234569874561	kyu@naver.com	01092461256	kyu	kyu	user	삭제
17	jin	jin	jin	1	jin@naver.com	01092875698	j	j	user	삭제
24	키키	aa	전종찬	1234561234563	aa@naver.com	01010101010	대전 동구 판교1길 3	1234	user	삭제
27	ddd	dddd	ddd	5001011234565	qwe@qwe	01012334567	서울 성북구 아리랑로 3	testsetset	user	삭제
28	wjdt12	wjdt12	김정수	9605041632546	wjdt12@naver.com	01012345678	대구 남구 경상길 1	남구길	user	삭제

[FIRST] [PRE][1][NXT] END

삭제 된 모습을 확인할 수 있음.

## 수정한 소스코드

- 전체코드를 올리기에 양이 많아서 수정한 소스 부분만 올리게되었습니다.
- 수정한 부분이 없는 소스코드는 첨부하지 않았습니다.

## MainController

```
package com.boardgame.mall.controller;

import javax.servlet.http.HttpSession;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

import com.boardgame.mall.dto.UserPageInfoVO;

@Controller
public class MainController {

    @GetMapping("/")
```

```

public String index(HttpSession session) {
    if( session.isNew()){ //새로운 객체가 만들어졌다면 세션을 저장한다.
        session.setAttribute("loginState", "logout"); //로그인 상태저장
        session.setAttribute("viewState", null); //지금 보는 화면 저장
        session.setAttribute("user_num", null); //유저 번호 저장
        session.setAttribute("user_name", null); //유저 이름 저장
        session.setAttribute("user_id", null); //유저 아이디 저장
        session.setAttribute("user_passwd", null); //유저 비번 저장
        session.setAttribute("user_class", null); //유저 등급 저장

    }
    session.setAttribute("userPageInfoVO",new UserPageInfoVO());
    return "index";
}
@PostMapping("/")
public String index2(HttpSession session) {
    if( session.isNew()){ //새로운 객체가 만들어졌다면 세션을 저장한다.
        session.setAttribute("loginState", "logout"); //로그인 상태저장
        session.setAttribute("viewState", null); //지금 보는 화면 저장
        session.setAttribute("user_num", null); //유저 번호 저장
        session.setAttribute("user_name", null); //유저 이름 저장
        session.setAttribute("user_id", null); //유저 아이디 저장
        session.setAttribute("user_passwd", null); //유저 비번 저장
        session.setAttribute("user_class", null); //유저 등급 저장

    }
    session.setAttribute("userPageInfoVO",new UserPageInfoVO());
    return "index";
}
}

```

- MainController를 통해서 홈페이지 진입을 하도록 설정했습니다.
- session 오류가 자주 발생하여서 값을 생성하도록 만들었습니다.

## User컨트롤러

```

package com.boardgame.mall.controller;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;

```

```

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import com.boardgame.mall.service.user.DeleteUserForIdAction;
import com.boardgame.mall.service.user.DeleteUserListDisplayAction;
import com.boardgame.mall.service.user.GetMyInfo;
import com.boardgame.mall.service.user.GetUserListAction;
import com.boardgame.mall.service.user.InsertUserAction;
import com.boardgame.mall.service.user.LoginUser;
import com.boardgame.mall.service.user.LogoutUser;
import com.boardgame.mall.service.user.UpdateUserDisplayAction;
import com.boardgame.mall.service.user.UpdateUserForIdAction;
import com.boardgame.mall.service.user.UpdateUserListDisplayAction;

@Controller
@RequestMapping("/user")
public class UserController {

    @PostMapping("/join")
    public String userjoin() {
        return "user/user_c";
    }

    @PostMapping("/login")
    public String loginUser(HttpServletRequest request, HttpServletResponse
response ) throws ServletException, IOException {
        System.out.println("로그인 시도");
        Action action = new LoginUser();
        ActionForward forward = null;
        try {
            forward = action.execute(request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
        String s = forward.getPath();
        System.out.println(s);
        return s;
    }

    @PostMapping("/logout")
    public String logoutUser(HttpServletRequest request, HttpServletResponse
response ) throws ServletException, IOException {
        System.out.println("로그아웃 시도");
        Action action = new LogoutUser();
        ActionForward forward = null;
        try {
            forward = action.execute(request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        String s = forward.getPath();
        return s;
    }

    @PostMapping("/UpdateUserDisplay")
    public String UpdateUserDisplay(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        Action action = new UpdateUserDisplayAction();
        ActionForward forward = null;
        try {
            forward = action.execute(request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
        String s = forward.getPath();
        System.out.println(s);
        return s;
    }

    //관리자가 일반유저 수정조회 리스트로 이동
    @GetMapping("/UpdateUserListDisplay/{cpn}")
    public String UpdateUserListDisplay(HttpServletRequest request,
        HttpServletResponse response, @PathVariable int cpn) throws ServletException,
        IOException {

        Action action = new UpdateUserListDisplayAction(cpn);
        ActionForward forward = null;
        try {
            forward = action.execute(request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
        String s = null;
        try {
            s = forward.getPath();
            System.out.println(s);
        } catch (Exception e) {
            e.printStackTrace();
            s = "etc/userError";
        }
        return s;
    }

    //회원 C
    @PostMapping("/CreateUser")
    public String CreateUser(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
        Action action = new InsertUserAction();
        ActionForward forward = null;

```



```

        try {
            forward = action.execute(request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
        String s = forward.getPath();
        System.out.println(s);
        return s;
    }

// 본인 정보 조회
    @GetMapping("/GetMyInfo")
    public String GetUser(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        Action action = new GetMyInfo();
        ActionForward forward = null;
        try {
            forward = action.execute(request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
        String s = forward.getPath();
        System.out.println(s);
        System.out.println("유저 정보조회");
        return s;
    }

//관리자 회원정보 전체조회
    @GetMapping("/GetUserList")
    public String GetUserList(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        Action action = new GetUserListAction();
        ActionForward forward = null;
        try {
            forward = action.execute(request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
        String s = forward.getPath();
        System.out.println(s);
        System.out.println("유저 정보조회");
        return s;
    }

//본인 정보 수정
    @PostMapping("/UpdateUserForId")
    public String PutUser(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        Action action = new UpdateUserForIdAction();
        ActionForward forward = null;
        try {
            forward = action.execute(request, response);
        } catch (Exception e) {

```

```

        e.printStackTrace();
    }
    String s = forward.getPath();
    System.out.println(s);
    System.out.println("유저 업데이트");
    return s;
}
//회원 탈퇴
@PostMapping("/DeleteUserForId")
public String DeleteUser(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    Action action = new DeleteUserForIdAction();
    ActionForward forward = null;
    try {
        forward = action.execute(request, response);
    } catch (Exception e) {
        e.printStackTrace();
    }
    String s = forward.getPath();
    System.out.println(s);
    System.out.println("유저 삭제");
    if(s.equals("logout")) {
        s = logoutUser(request, response);
    } else if(s.equals("index")) {
        s = "index";
    }
    return s;
}

//관리자 회원삭제 목록조회
@GetMapping("/DeleteUserListDisplay/{cpn}")
public String DeleteUserListDisplay(HttpServletRequest request,
HttpServletResponse response, @PathVariable int cpn) throws ServletException,
IOException {
    Action action = new DeleteUserListDisplayAction(cpn);
    ActionForward forward = null;
    try {
        forward = action.execute(request, response);
    } catch (Exception e) {
        e.printStackTrace();
    }

    String s = null;
    try {
        s = forward.getPath();
        System.out.println(s);
    } catch (Exception e) {
        e.printStackTrace();
        s = "etc/userError";
    }
    return s;
}

```

```
}
```

- 기본적으로 ~Action 파일은 대부분 수정하지 않았습니다.

```
##
```

## UserDAO

```
public class UserDAO {

    private PreparedStatement pstmt = null;
    private Connection con = null;
    private ResultSet rs = null;

    public UserDAO() {
        super();
        dbConnect();
    }

    /** Database connection method */
    public void dbConnect() {
        try {
            // Load MariaDB JDBC driver
            Class.forName("org.mariadb.jdbc.Driver");

            // Establish database connection
            String url = "jdbc:mariadb://localhost:3306/rwk51_boardgame_mall";
            String username = "root";
            String password = "maria";
            con = DriverManager.getConnection(url, username, password);

            System.out.println("DB connection successful");
        } catch (Exception e) {
            System.out.println("DB connection failed");
            e.printStackTrace();
        }
    }

    /** Database disconnect method */
    public void disconnect() {
        if (pstmt != null) {
            try {
                pstmt.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        if (con != null) {
```

```

        try {
            con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
/** Add query result set object release part */
if (rs != null) {
    try {
        rs.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

- UserDao의 수정한 부분만 첨부했습니다.
- DBCP를 사용하는데 오류가 존재하여서 JDBC 드라이버 방식으로 DB와 연결을 진행했습니다.
- 기존의 CRUD 코드는 2학년2학기에 작성한 코드와 같습니다.

## user\_r

```

<table class="table table-striped table-hover">
    <thead>
        <tr>
            <th>번호</th>
            <th>아이디</th>
            <th>비밀번호</th>
            <th>이름</th>
            <th>주민등록번호</th>
            <th>이메일</th>
            <th>휴대폰</th>
            <th>주소</th>
            <th>상세 주소</th>
            <th>등급</th>
        </tr>
    </thead>
    <tbody>
        <c:forEach items="${userList}" var="user">
            <tr>
                <td>${user.user_num}</td>
                <td>${user.user_id}</td>
                <td>${user.user_passwd}</td>
                <td>${user.user_name}</td>
                <td>${user.user_RRN}</td>
                <td>${user.user_email}</td>
                <td>${user.user_phone}</td>
                <td>${user.user_address}</td>
            </tr>
        </c:forEach>
    </tbody>
</table>

```

```

        <td>${user.user_addressdetail}</td>
        <td>${user.user_class}</td>
    </tr>
</c:forEach>
</tbody>
</table>

```

## user\_u

```

<c:set var="userList" value="${requestScope.userList}"/>
<c:set var="upiv0" value="${sessionScope.userPageInfoV0}"/>

<c:set var="currentPageNo" value="${upiv0.currentPageNo}"/>

<table class="table table-striped table-hover">
    <thead>
        <tr>
            <th>번호</th>
            <th>아이디</th>
            <th>비밀번호</th>
            <th>이름</th>
            <th>주민등록번호</th>
            <th>이메일</th>
            <th>휴대폰</th>
            <th>주소</th>
            <th>상세 주소</th>
            <th>등급</th>
            <th>수정</th>
        </tr>
    </thead>
    <tbody>
        <c:forEach items="${userList}" var="user">
            <tr>
                <td><c:out value="${user.user_num}"/></td>
                <td><c:out value="${user.user_id}"/></td>
                <td><c:out value="${user.user_passwd}"/></td>
                <td><c:out value="${user.user_name}"/></td>
                <td><c:out value="${user.user_RRN}"/></td>
                <td><c:out value="${user.user_email}"/></td>
                <td><c:out value="${user.user_phone}"/></td>
                <td><c:out value="${user.user_address}"/></td>
                <td><c:out value="${user.user_addressdetail}"/></td>
                <td><c:out value="${user.user_class}"/></td>
                <td>
                    <form id="form" method="post"
action="/user/UpdateUserDisplay">
                        <input class="btn btn-success float-right"
style="margin-left:10px;" type="submit" value="수정"/>

```

```

                                <input type="hidden" name="user_num" value="<c:out
value='${user.user_num}' />" />"/>
                                </form>
                                </td>
                                </tr>
                                </c:forEach>
                                </tbody>
                                </table>
                                <div class="clearfix">
                                <ul class="pagination">
                                <li class="page-item disabled"><a
href="/user/UpdateUserListDisplay/0">[FIRST]</a></li>
                                <c:if test="${currentPageNo > 0}">
                                <li class="page-item">
                                <a href="/user/UpdateUserListDisplay/${currentPageNo - 1}">[PRE]
</a>
                                </li>
                                </c:if>
                                <c:if test="${currentPageNo == 0}">
                                <li class="page-item">[PRE]</li>
                                </c:if>
                                <c:forEach begin="${upivO.startPageNo}" end="${upivO.endPageNo - 1}"
var="i">
                                <c:choose>
                                <c:when test="${i == currentPageNo}">
                                <li class="page-item active">[${i + 1}]</li>
                                </c:when>
                                <c:otherwise>
                                <li class="page-item">
                                <a href="/user/UpdateUserListDisplay${i}">[${i + 1}]</a>
                                </li>
                                </c:otherwise>
                                </c:choose>
                                </c:forEach>
                                <c:if test="${currentPageNo < upivO.pageCnt - 1}">
                                <li class="page-item">
                                <a href="/user/UpdateUserListDisplay${currentPageNo + 1}">[NXT]
</a>
                                </li>
                                </c:if>
                                <c:if test="${currentPageNo == upivO.pageCnt - 1}">
                                <li class="page-item">[NXT]</li>
                                </c:if>
                                <li class="page-item">
                                <a href="/user/UpdateUserListDisplay${upivO.pageCnt - 1}">END</a>
                                </li>
                                </ul>
                                </div>

```

## user\_u2

```
<form method="post" action="/user/UpdateUserForId">
    <!-- 경로 수정 -->
    <div class="form-group">
        <div>
            <span>아이디</span>
            <input class="form-control mb-2" name="user_id"
value="${user.user_id}" size="30" readonly>
        </div>
    </div>
    <!-- 비밀번호 -->
    <div class="form-group">
        <span>비밀번호</span>
        <input class="form-control mb-2" type="password"
name="user_passwd" size="30" value="${user.user_passwd}" required id="user_pw"
oninput="pw_chk()" placeholder="비밀번호">
    </div>

    <div class="form-group has-danger">
        <span>비밀번호 재확인</span>
        <input class="form-control mb-2" type="password"
name="user_passwordchk" size="30" required placeholder="비밀번호 재확인"
id="user_repw" oninput="pw_chk()">
        <span id="pw_chk"></span>
    </div>

    <div class="form-group">
        <span>이름</span>
        <input class="form-control mb-2" type="text"
name="user_name" size="30" value="${user.user_name}" required readonly>
    </div>

    <div class="form-group">
        <span>이메일</span>
        <input class="form-control mb-2" type="email"
name="user_email" size="30" value="${user.user_email}" required readonly>
    </div>

    <div class="form-group">
        <span>휴대폰</span>
        <input class="form-control mb-2" type="number"
name="user_phone" size="30" value="${user.user_phone}" required readonly>
    </div>

    <div class="form-group">
        <input class="form-control mb-2" type="text"
id="user_address" name="user_address" readonly required placeholder="주소 "
value="${user.user_address}">
    </div>
```

```

        <div class="form-group">
            <input class="form-control mb-2" type="text"
name="user_addressdetail" required placeholder="상세주소 ">
        </div>

        <c:if test="${sessionScope.user_class eq 'admin'}">
            <div class="form-group">
                <span>등급</span>
                <select name="user_class" class="form-control mb-2">
                    <option value="user" <c:if test="${user.user_class
eq 'user'}">selected</c:if>>일반사용자</option>
                    <option value="admin" <c:if test="${user.user_class
eq 'admin'}">selected</c:if>>관리자</option>
                </select>
            </div>
        </c:if>

        <c:if test="${sessionScope.user_class ne 'admin'}">
            <div class="form-group">
                <span>등급</span>
                <input class="form-control mb-2" type="text"
name="user_phone" size="30" value="${user.user_class}" required readonly>
            </div>
        </c:if>
        <input type="hidden" name="user_num" value="${user.user_num}" />
        <input type="hidden" name="user_RRN" value="${user.user_RRN}" />
        <input class="btn btn-success float-right" style="margin-
left:10px;" type="submit" value="수정하기" />
    </form>

```

## user\_d

```

<table class="table table-striped table-hover">
    <thead>
        <tr>
            <th>번호</th>
            <th>아이디</th>
            <th>비밀번호</th>
            <th>이름</th>
            <th>주민등록번호</th>
            <th>이메일</th>
            <th>휴대폰</th>
            <th>주소</th>
            <th>상세 주소</th>
            <th>등급</th>
            <th>삭제</th>
        </tr>
    </thead>
    <tbody>

```



```

<c:forEach items="${userList}" var="user">
    <tr>
        <td><c:out value="${user.user_num}"/></td>
        <td><c:out value="${user.user_id}"/></td>
        <td><c:out value="${user.user_passwd}"/></td>
        <td><c:out value="${user.user_name}"/></td>
        <td><c:out value="${user.user_RRN}"/></td>
        <td><c:out value="${user.user_email}"/></td>
        <td><c:out value="${user.user_phone}"/></td>
        <td><c:out value="${user.user_address}"/></td>
        <td><c:out value="${user.user_addressdetail}"/></td>
        <td><c:out value="${user.user_class}"/></td>
        <td>
            <form id="form" method="post"
action="/user/DeleteUserForId">
                <input class="btn btn-success float-right"
style="margin-left:10px;" type="submit" value="삭제"/>
                <input type="hidden" name="user_num" value="<c:out
value='${user.user_num}'"/>"/>
            </form>
        </td>
    </tr>
</c:forEach>
</tbody>
</table>
<div class="clearfix">
    <ul class="pagination">
        <li class="page-item disabled"><a
href="/user/DeleteUserListDisplay/0">[FIRST]</a></li>
        <c:if test="${currentPageNo > 0}">
            <li class="page-item">
                <a href="/user/DeleteUserListDisplay/${currentPageNo - 1}">[PRE]
</a>
            </li>
        </c:if>
        <c:if test="${currentPageNo == 0}">
            <li class="page-item">[PRE]</li>
        </c:if>
        <c:forEach begin="${upivo.startPageNo}" end="${upivo.endPageNo - 1}"
var="i">
            <c:choose>
                <c:when test="${i == currentPageNo}">
                    <li class="page-item active">[${i + 1}]</li>
                </c:when>
                <c:otherwise>
                    <li class="page-item">
                        <a href="/user/DeleteUserListDisplay/${i}">[${i + 1}]</a>
                    </li>
                </c:otherwise>
            </c:choose>
        </c:forEach>
        <c:if test="${currentPageNo < upivo.pageCnt - 1}">

```

```

        <li class="page-item">
            <a href="/user/DeleteUserListDisplay/${currentPageNo + 1}">[NXT]
        </a>

        </li>
    </c:if>
    <c:if test="${currentPageNo == upivo.pageCnt - 1}">
        <li class="page-item">[NXT]</li>
    </c:if>
    <li class="page-item">
        <a href="/user/DeleteUserListDisplay/${upivo.pageCnt - 1}">END</a>
    </li>
</ul>
</div>
</div>

```

## my\_page

```

<c:set var="user" value="${myPage}"/>
<div class="container row">
    <div class="col-lg-7">
        <div class="jumbotron" style="padding-top: 20px;">
            <h3 style="text-align: center;" class="mb-4">내 정보</h3>
            <!-- form -->
            <div class="form-group">
                <div>
                    <span>아이디 : ${user.user_id}</span>
                </div>
            </div>

            <div class="form-group">
                <span>이름 : ${user.user_name}</span>
            </div>

            <div class="form-group">
                <span>이메일 : ${user.user_email}</span>
            </div>

            <div class="form-group has-danger">
                <span>휴대폰 : ${user.user_phone}</span>
            </div>

            <div class="form-group">
                <span>주소 : ${user.user_address}</span>
            </div>

            <div class="form-group">
                <span>상세 주소 : ${user.user_addressdetail}</span>
            </div>

```

```

        <div class="form-group">
            <span >등급 : ${user.user_class}</span>
        </div>
        <form id="form" method="post" action="/user/UpdateUserDisplay">
            <input class="btn btn-success float-right" style="margin-left:
10px;" type="submit" value="회원 수정">
            <input type="hidden" name="user_num" value="${user.user_num}">
        </form>
        <form id="form" method="post" action="/user/DeleteUserForId">
            <input class="btn btn-success float-right" style="margin-left:
10px;" type="submit" value="회원 탈퇴">
            <input type="hidden" name="user_num" value="${user.user_num}">
        </form>
    </div>
</div>

<div class="col-lg-7" >
    <div class="jumbotron" style="padding-top: 20px;">
        <div class="form-group">
            <div><a href="./getCartList.be?user_num=${user.user_num}" >장바구니
</a></div>
        </div>
    </div>
</div>
<c:if test="${sessionScope.user_class eq 'admin'}">
    <div class="col-lg-7" >
        <div class="jumbotron" style="padding-top: 20px;">
            <h3 style="text-align: center;" class="mb-4">관리자 기능</h3>
            <!-- form -->
            <div class="form-group">
                <div><a href="/user/GetUserList">회원 정보조회</a></div>
            </div>

            <div class="form-group">
                <div><a href="/user/DeleteUserListDisplay/0">회원 정보삭제</a>
</div>

            <div class="form-group">
                <div><a href="/user/UpdateUserListDisplay/0">회원 정보수정</a>
</div>

            <div class="form-group has-danger">
                <div><a href ="${boardgameViewDir}/boardgame_c.jsp">상품등록</a>
</div>
            </div>
        </div>
    </div>
</c:if>
</div>

```



```

        </tr>
        <tr>
            <td>
                <a href="/user/GetMyInfo">마이페이지</a>
            </td>
        </tr>
    </table>
</div>
</c:if>

```

## 수정사항이 없는 파일

- user\_c.jsp
- index.jsp
- footer.jsp
- globalData.jsp

## 수행후기

처음에 과제를 할때 어떻게 로직을 수정해야할지에 대해 헛갈렸습니다. 그래서 교수님을 찾아가서 직접 교수님께 물어본 결과

되도록 DAO, DTO, Action 등의 기능을 내버려두고 FrontController를 @Controller를 사용해서 서블릿 방식이 아닌 어노테이션 형식으로 조작하고

jsp 파일을 jstl/el 로 코드를 수정하라는 것이었습니다.

그래서 컨트롤러에서 기존 action을 호출하는 방법을 탐색하게 되었고 스프링에서도 HttpServletRequest, HttpServletResponse를 사용할 수 있어서

메소드에 메게 변수로 해당 객체를 받은 후 해당 객체를 각 action들에게 전달하는 과정으로 코드를 구성했습니다.

또한 기존 action에서 forward.setPath를 통해 경로를 지정했는데

성공시에는 성공한 화면으로 넘어가게 하고 실패시 에러 화면으로 넘어가게 경로를 유지 후 해당 forward.getPath의 값을 return 해주는 방식으로 코드를 구현했습니다.

그 결과 생각보다 코드를 고칠게 없었다 라는 생각이 들게되었습니다.

물론 해당 부분을 구성하는데 어려움이 있었으나 해당부분이 해결되고 난후에 코드 구성에서 어려움은 적었는거 같습니다.

코드를 구성할때 어려웠던거는 페이징 기능을 어떻게 해야할까를 고민을 했습니다.

왜냐하면 자꾸 페이징 기능의 session의 값이 초기화가 되었고, 유저의 session값도 null로 뜨게 되었습니다.

그래서 MainController를 만들어서 메인홈페이지로 서버를 접속할 경우 session을 설정하도록 코드를 구현했습니다.

이렇게 하나하나 순차적으로 코드를 구현하고 구상하고 시행하고 검증하면서 과제를 해결 할 수 있었는데 같습니다.