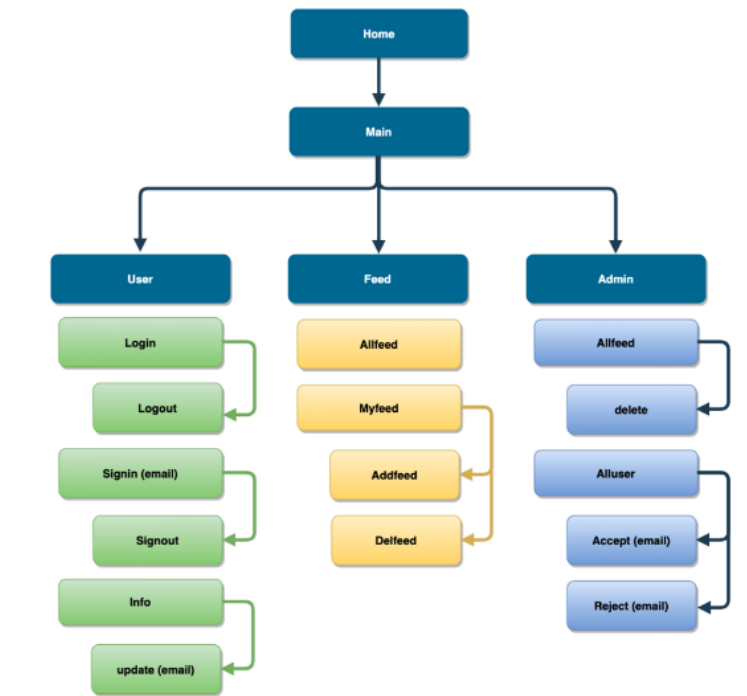


SNS Project 4 230209

Java - Project

- viewport
- 예외처리

IA Information Architecture



회원 경로	회원명	작성사	작성일	페이지	s
Description					
1	패스워드 입력 오차 확인 JS				
2	회원 가입 신청 후 담당자의 승인 후 사용가능				
3	회원 클릭하면 회원 가입 승인 후 디비 이동				
4	회원 클릭하면 회원 가입 거부 후 디비 삭제				
5					

Group Talk

가입 신청서

아이디 (이메일 주소)

패스워드

패스워드 (확인) 1

이름

회원 가입 신청 2

주소

Group Talk

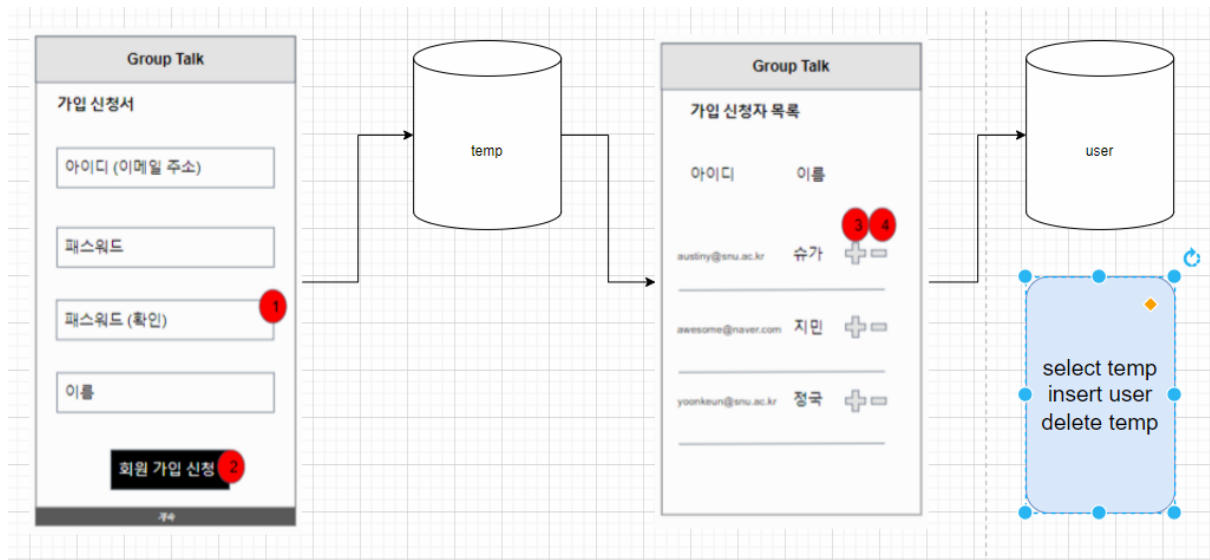
가입 신청자 목록

아이디 이름

austiny@snu.ac.kr 슈가 3 4

awesome@naver.com 지민 + -

yoonkeun@snu.ac.kr 정국 + -



temp 테이블 생성 : 신청자용 임시 테이블

HeidiSQL 12.3.0.6589

파일 편집 검색 쿼리 도구 이동 도움말

호스트: yoonkeunsoo.cafe24.com 데이터베이스: yoonkeunsoo 테이블: temp

yooneunsoo.temp: 1 행 (중) (대략적)

id	password	name	ts
test@test.com	0000	0000	2023-02-09 08:47:43

Left sidebar: Cafe24 > yoonk... 48.0 KiB > feed 16.0 KiB > temp 16.0 KiB > user 16.0 KiB

userDAO.java

```
package jdbc;

import java.sql.*;
import java.util.ArrayList;

import javax.naming.NamingException;

import org.json.simple.JSONArray;
import org.json.simple.JSONObject;

import util.ConnectionPool;

public class userDAO {

    // 관리자가 임시 테이블 자료를 유저테이블로 가입 승인 절차
    public static int insertadmin(String id) throws NamingException, SQLException {

        Connection conn = null;
        PreparedStatement pstmt = null;
        ResultSet rs = null;

        try {
```

```

// temp 에서 id 값으로 데이터 조회해오기 select

String sql = "SELECT * FROM temp WHERE id = ?";

conn = ConnectionPool.get();
pstmt = conn.prepareStatement(sql);
pstmt.setString(1, id);

rs = pstmt.executeQuery();

rs.next();

String tid = rs.getString(1);
String tpassword = rs.getString(2);
String tname = rs.getString(3);

// 조회한 데이터를 user 에 넣기 insert
String sql2 = "INSERT INTO user(id,password,name) VALUES(?,?,?)";

pstmt = conn.prepareStatement(sql2);
pstmt.setString(1, tid);
pstmt.setString(2, tpassword);
pstmt.setString(3, tname);

pstmt.executeUpdate(); //성공 1, 실패 0 을 가지고 나간다.

// temp 에서 해당 데이터 삭제하기 delete

String sql3 = "DELETE FROM temp WHERE id = ?";

pstmt = conn.prepareStatement(sql3);
pstmt.setString(1, id);

return pstmt.executeUpdate();

}finally {
    if(rs != null) rs.close();
    if(pstmt != null) pstmt.close();
    if(conn != null) conn.close();
}
}

// 회원 가입 신청
public static int inserttemp(String id, String password, String name) throws NamingException, SQLException {

    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        String sql = "INSERT INTO temp(id,password,name) VALUES(?,?,?)";

        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, id);
        pstmt.setString(2, password);
        pstmt.setString(3, name);

        return pstmt.executeUpdate(); //성공 1, 실패 0 을 가지고 나간다.

    }finally {
        if(pstmt != null) pstmt.close();
        if(conn != null) conn.close();
    }
}

//회원 가입
public static int insert(String id, String password, String name) throws NamingException, SQLException {

    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        String sql = "INSERT INTO user(id,password,name) VALUES(?,?,?)";

        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, id);

```

```

        pstmt.setString(2, password);
        pstmt.setString(3, name);

        return pstmt.executeUpdate(); //성공 1, 실패 0 을 가지고 나간다.

    }finally {
        if(pstmt != null) pstmt.close();
        if(conn != null) conn.close();
    }
}

//회원 정보 수정
public static int edit(String id, String password, String name) throws NamingException, SQLException {

    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        String sql = "UPDATE user SET password = ?, name = ? WHERE id = ?";

        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, password);
        pstmt.setString(2, name);
        pstmt.setString(3, id);

        return pstmt.executeUpdate(); //성공 1, 실패 0 을 가지고 나간다.

    }finally {
        if(pstmt != null) pstmt.close();
        if(conn != null) conn.close();
    }
}

//회원 가입시 아이디가 이미 존재하는 지 여부 확인
public static boolean exist(String id) throws NamingException, SQLException {

    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    try {
        String sql = "SELECT id FROM user WHERE id = ?";

        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, id);

        rs = pstmt.executeQuery();

        return rs.next(); // 조회한 아이디가 DB에 존재하면 true 없으면 false

    }finally {
        if(rs != null) rs.close();
        if(pstmt != null) pstmt.close();
        if(conn != null) conn.close();
    }
}

// 회원 탈퇴
public static int delete(String id) throws NamingException, SQLException {

    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        String sql = "DELETE FROM user WHERE id = ?";

        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, id);

        return pstmt.executeUpdate(); //성공 1, 실패 0 을 가지고 나간다.

    }finally {
        if(pstmt != null) pstmt.close();
        if(conn != null) conn.close();
    }
}

//회원 로그인
public static int login(String id, String upassword) throws NamingException, SQLException {

    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

```

```

try {
    String sql = "SELECT id, password FROM user WHERE id = ?";

    conn = ConnectionPool.get();
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, id);

    rs = pstmt.executeQuery();

    if(!rs.next()) return 1; // 1 아이디가 존재하지 않는 경우
    if(!upassword.equals(rs.getString("password"))) return 2; //아이디는 존재하지만 비밀번호 일치하지 않는 경우
    return 0; // 로그인 성공

}finally {
    if(rs != null) rs.close();
    if(pstmt != null) pstmt.close();
    if(conn != null) conn.close();
}
}

//회원 목록, 꺼내기 연습용
public static ArrayList<userDTO> list() throws NamingException, SQLException {

    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    try {
        String sql = "SELECT * FROM user ORDER BY ts DESC";

        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql);

        rs = pstmt.executeQuery();

        ArrayList<userDTO> users = new ArrayList<userDTO>();

        while(rs.next()) {
            users.add(new userDTO(rs.getString(1),
                rs.getString(2),
                rs.getString(3),
                rs.getString(4)));
        }
        return users;
    }finally {
        if(rs != null) rs.close();
        if(pstmt != null) pstmt.close();
        if(conn != null) conn.close();
    }
}

//임시 테이블 목록 보기
public static String gettemp() throws NamingException, SQLException {

    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    try {
        String sql = "SELECT * FROM temp ORDER BY ts DESC";

        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql);

        rs = pstmt.executeQuery();

        JSONArray users = new JSONArray();

        while(rs.next()) {
            JSONObject obj = new JSONObject();
            obj.put("id", rs.getString(1));
            obj.put("password", rs.getString(2));
            obj.put("name", rs.getString(3));
            obj.put("ts", rs.getString(4));

            users.add(obj);
        }

        return users.toJSONString();
    }
}

```

```

    }finally {
        if(rs != null) rs.close();
        if(pstmt != null) pstmt.close();
        if(conn != null) conn.close();
    }
}

}
}

```

temp.jsp

```

<%@page import="jdbc.*"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page errorPage = "page_error_page.jsp"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Group Talk</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>
<body>
<%@ include file="/header.jsp" %>

<%
    sid = (String) session.getAttribute("id");
    if (!sid.equals("admin")){
%>
    <!-- Modal -->
    <div class="modal fade" id="exampleModal2" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-labelledby="staticModal2Label" data-bs="static">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h1 class="modal-title fs-5" id="exampleModalLabel">관리자 전용 메뉴</h1>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    관리자만 사용 가능한 메뉴입니다.<br>
                    로그인 화면으로 이동합니다.
                </div>
                <div class="modal-footer">
                    <button onclick="location.href='/user/login.jsp'" class="btn btn-primary">로그인 이동</button>
                </div>
            </div>
        </div>
    </div>

    <script>
        $(function() {
            $("#exampleModal2").modal("show");
        });
    </script>

<%} else {
    session.setAttribute("id", sid);
}
%>

<style>
@font-face {
    font-family: 'GangwonEdu_OTFBoldA';
    src: url('https://cdn.jsdelivr.net/gh/projectnoonnu/noonfonts_2201-2@1.0/GangwonEdu_OTFBoldA.woff') format('woff');
    font-weight: normal;
    font-style: normal;
}

body {
    display: flex;
    align-items: center;
    padding-top: 30px;
    padding-bottom: 30px;
    background-color: #f5f5f5;
    font-family: 'GangwonEdu_OTFBoldA';
}
</style>

```

```

<div class="container bg-warning shadow mx-auto p-5 w-75">

<script>
function addItem(id) {
$.ajax({
type:"post",
url: "useradd.jsp",
data : {id:id},

dataType:"text",

success:function(data) {
alert("관리자 회원등록 성공");
searchFunction();

}
});
}

function delItem(no) {
$.ajax({
type:"post",
url: "/feed/feeddel.jsp",
data : {no:no

},
dataType:"text",

success:function(data) {
searchFunction();
}
});
}

function searchFunction() {

$.ajax({
type:'post',
url:'tempall.jsp',
success:function(data){
var users = JSON.parse(data.trim());

var str = "";
for(var i = 0; i < users.length; i++){

str += "<tr><td>" + users[i].id + "</td>";
str += "<td><small>&nbsp;" + users[i].name + "</small></td>";
str += "<td><div onclick='delItem(\"" + users[i].id + "\"" + "\")'><span class='btn btn-danger'>Reject</span></div></td>";
str += "<td><div onclick='addItem(\"" + users[i].id + "\"" + "\")'><span class='btn btn-primary'>Accept</span></div></td>";
str += "</tr>";
str += "<tr><td colspan=4 height=40><hr></td></tr>"
} $("#ajaxTable").html(str);
}
});
}

window.onload = function() {
searchFunction();
}

</script>

<h1>가입 신청자 목록</h1>
<hr>
<div class = "input-group justify-content-center">

<table align=center width="400">
<thead>
</thead>
<tbody id="ajaxTable">

</tbody>
</table>
</div>

<%% include file="/footer.jsp" %>
</body>
</html>

```

Python - Inheritance, Exception

Google Colaboratory

 <https://colab.research.google.com/drive/12QgQAEUqqGDCSz9DwlB7LX8hdwLwGdJj?usp=sharing>

