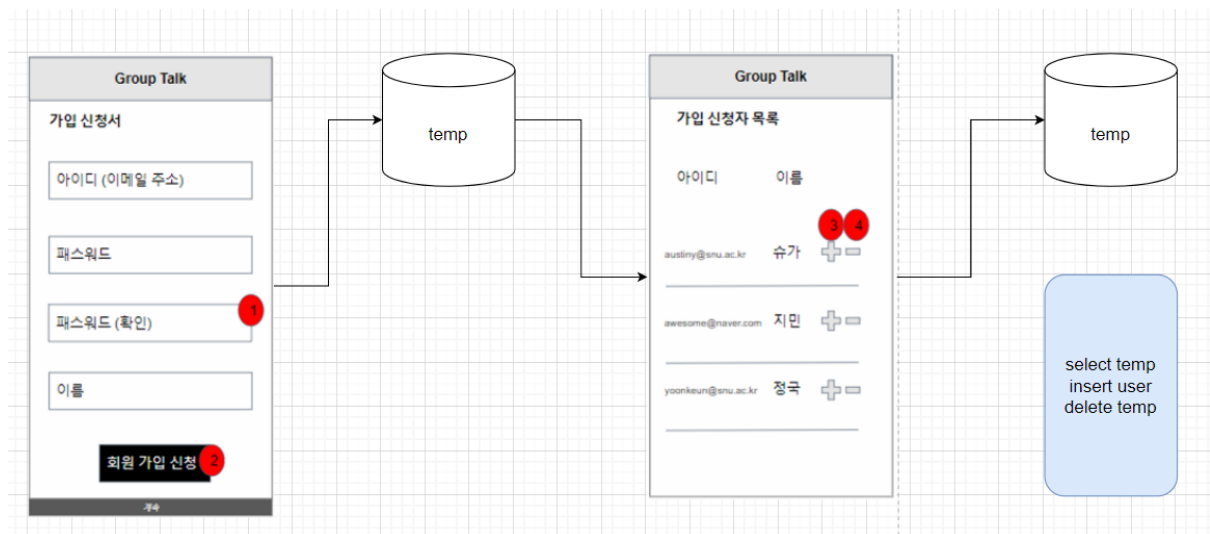
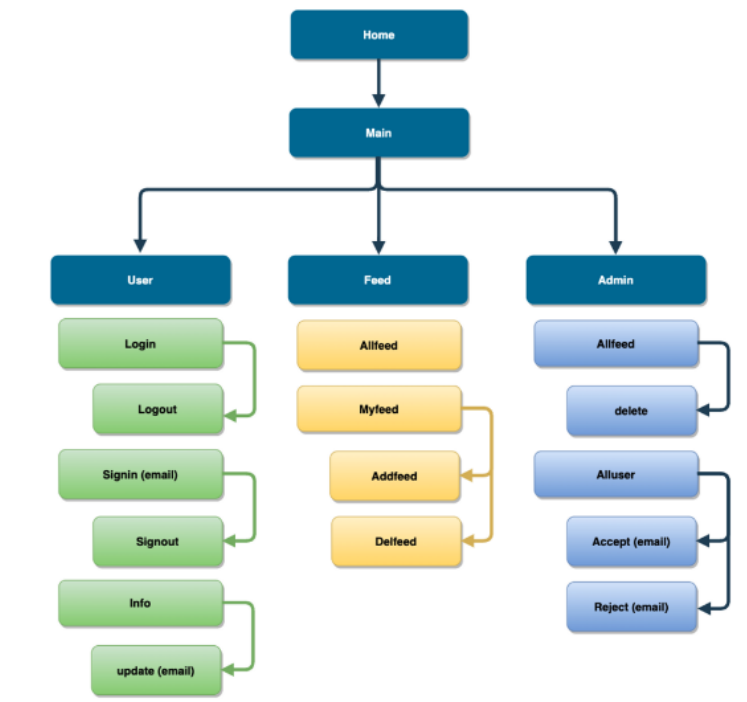
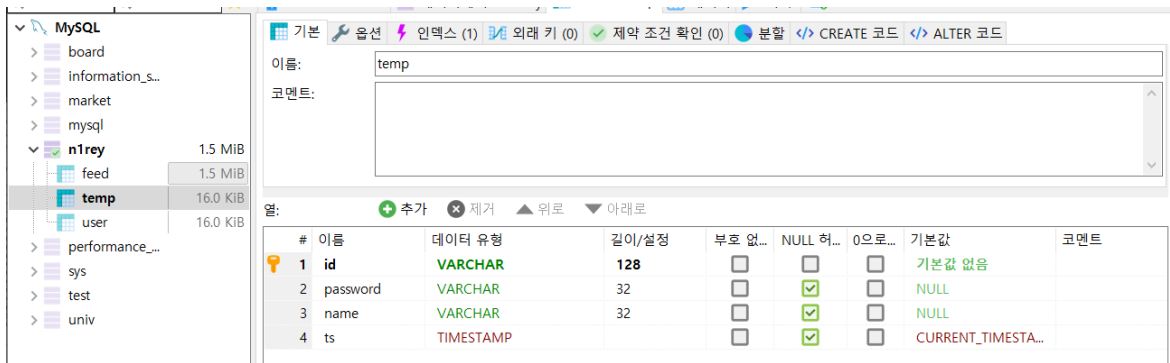


SNS Project 4

IA Information Architecture



▼ temp 테이블 생성 : 신청자 용 임시 테이블

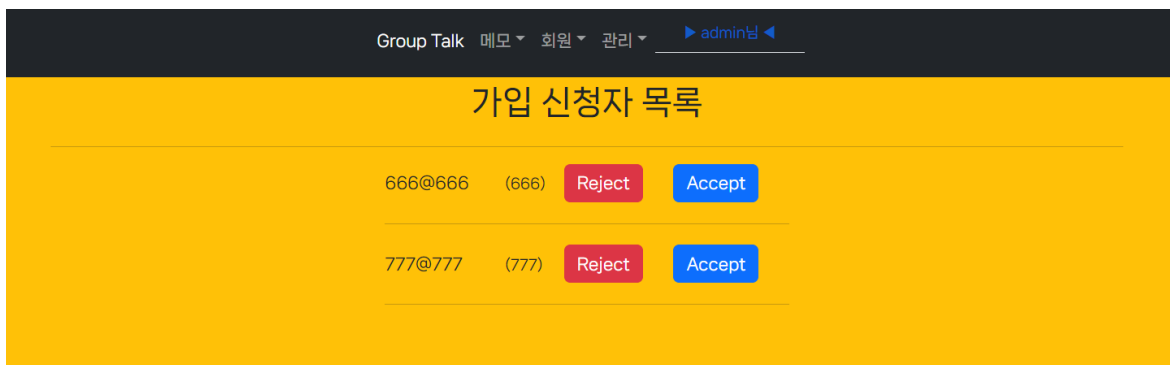


▼ 모달 창이 아닌 곳 클릭 금지 방법

```
<div class="modal fade" id="exampleModal2" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">
```

관리자 전용 페이지

▼ 회원 관리



▼ 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>
</head>
<body>
<%@ include file = "/header.jsp" %>
<%
    sid = (String) session.getAttribute("id");
    if(!sid.equals("admin")){
        %>
<!-- Modal -->
<div class="modal fade" id="exampleModal1" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-labelledby="exampleModal1Label" data-bs-static="true">
<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" onclick="location.href='/user/login.jsp'" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
관리자만 사용 가능한 메뉴입니다. <br>
로그인 페이지로 이동합니다.
</div>
<div class="modal-footer">
<button type="button" onclick="location.href='/user/login.jsp'" class="btn btn-primary">이동</button>
</div>
</div>
</div>
</div>
```

```

        </div>
    </div>
</div>
</div>

<script>
    $(function() {
        $("#exampleModal1").modal("show");
    });
</script>

<%
    } else {
        session.setAttribute("id", sid);
    }
%>
<style>
@font-face {
    font-family: 'NanumSquareNeo-Variable';
    src: url('https://cdn.jsdelivr.net/gh/projectnoonnu/noonfonts_11-01@1.0/NanumSquareNeo-Variable.woff2') format('woff2');
    font-weight: normal;
    font-style: normal;
}

body {
    display: flex;
    align-items: center;
    padding-top: 30px;
    padding-bottom: 30px;
    background-color: #f5f5f5;
    font-family: 'NanumSquareNeo-Variable';
}
</style>
<div class="container bg-warning shadow mx-auto p-5 w-75">
<!-- container 화면 전체를 차지하지 않는 고정 크기 컨테이너,
        브라우저의 크기에 반응하여 자동으로 크기가 변동됨
        bg-warning bg는 배경 warning은 노란색
        shadow 입체감 있게 그림자 효과 추가
        mx-auto m 바깥여백 x 가로로 auto 화면 가운데 정렬
        p padding 안쪽 여백
        w 전체 너비
-->
<script>
function addUser(id) {
    $.ajax({
        type:"post",
        url: "useradd.jsp",
        data : {id:id

    },
    dataType:"text",

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

function delUser(id) {
    $.ajax({
        type:"post",
        url: "userdel.jsp",
        data : {id:id

    },
    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='delUser(\"" + users[i].id + "\"" + "&#39;><span class='btn btn-danger'>Reject</span></div></td>";
                str += "<td><div onclick='addUser(\"" + users[i].id + "\"" + "&#39;><span class='btn btn-primary'>Accept</span></div></td>";
            }
            str += "</tr>";
            document.getElementById("tempall").innerHTML += str;
        }
    });
}

```

```

        str += "</tr>";
        str += "<tr><td colspan=4 height=40><hr></td></tr>";
    } $("#ajaxTable").html(str);

    }
    });
}

window.onload = function(){
    searchFunction();
}
</script>
<center><h2>가입 신청자 목록</h2></center>
<div>
    <div class="input-group justify-content-center" >

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

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

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

```

▼ tempall.jsp

```

<%@page import="jdbc.*"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    out.print(userDAO.gettemp());
%>

```

▼ useradd.jsp

```

<%@page import="jdbc.*"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    String id = request.getParameter("id");

    out.print(userDAO.insertadmin(id));
%>

```

▼ userdel.jsp

```

<%@page import="jdbc.*"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    String id = request.getParameter("id");

    out.print(userDAO.deletetemp(id));
%>

```

▼ userDAO.java

```

package jdbc;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
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 유저 테이블로 가입 승인 절차
    public static int insertadmin(String id) throws NamingException, SQLException {
        Connection conn = null;
        PreparedStatement pstmt = null;
        ResultSet rs = null;

        try {

            // temp에서 id 값으로 데이터 조회하기
            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);

            rs.next();
            // 조회한 데이터를 user에 넣기
            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();

            // temp에서 해당 데이터 삭제하기
            String sql3 = "DELETE FROM temp WHERE id = ?";
            pstmt = conn.prepareStatement(sql3);
            pstmt.setString(1, id);

            return pstmt.executeUpdate(); //성공하면 1, 실패하면 0을 가지고 나감
        } finally {
            if(rs != null) rs.close();
            if(pstmt != null) pstmt.close();
            if(conn != null) conn.close();
        }
    }

    //관리자가 임시 테이블 자료를 가입 승인 거절 후 삭제
    public static int deletetemp(String id) throws NamingException, SQLException {
        Connection conn = null;
        PreparedStatement pstmt = null;

        try {
            String sql = "DELETE FROM temp WHERE id = ?";
            conn = ConnectionPool.get();
            pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, id);

            return pstmt.executeUpdate(); //성공하면 1, 실패하면 0을 가지고 나감
        } finally {
            if(pstmt != null) pstmt.close();
            if(conn != null) conn.close();
        }
    }

    //임시 회원 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 ASC";

            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();
    }
}
}
}

```

▼ 내가 한 insertadmin

```

//관리자가 임시 테이블 자료를 유저 테이블로 가입 승인 절차
public static int myinsertadmin(String id) throws NamingException, SQLException {
    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        // temp에서 id 값으로 데이터 조회하기
        String sql1 = "SELECT * FROM temp WHERE id = ?";
        // 조회한 데이터를 user에 넣기
        String sql1 = "INSERT INTO user SELECT * FROM temp WHERE id = ?";
        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql1);
        pstmt.setString(1, id);

        pstmt.executeUpdate();
        // temp에서 해당 데이터 삭제하기
        String sql2 = "DELETE FROM temp WHERE id = ?";
        pstmt = conn.prepareStatement(sql2);
        pstmt.setString(1, id);

        return pstmt.executeUpdate(); //성공하면 1, 실패하면 0을 가지고 나감
    } finally {
        if(pstmt != null) pstmt.close();
        if(conn != null) conn.close();
    }
}
}

```