

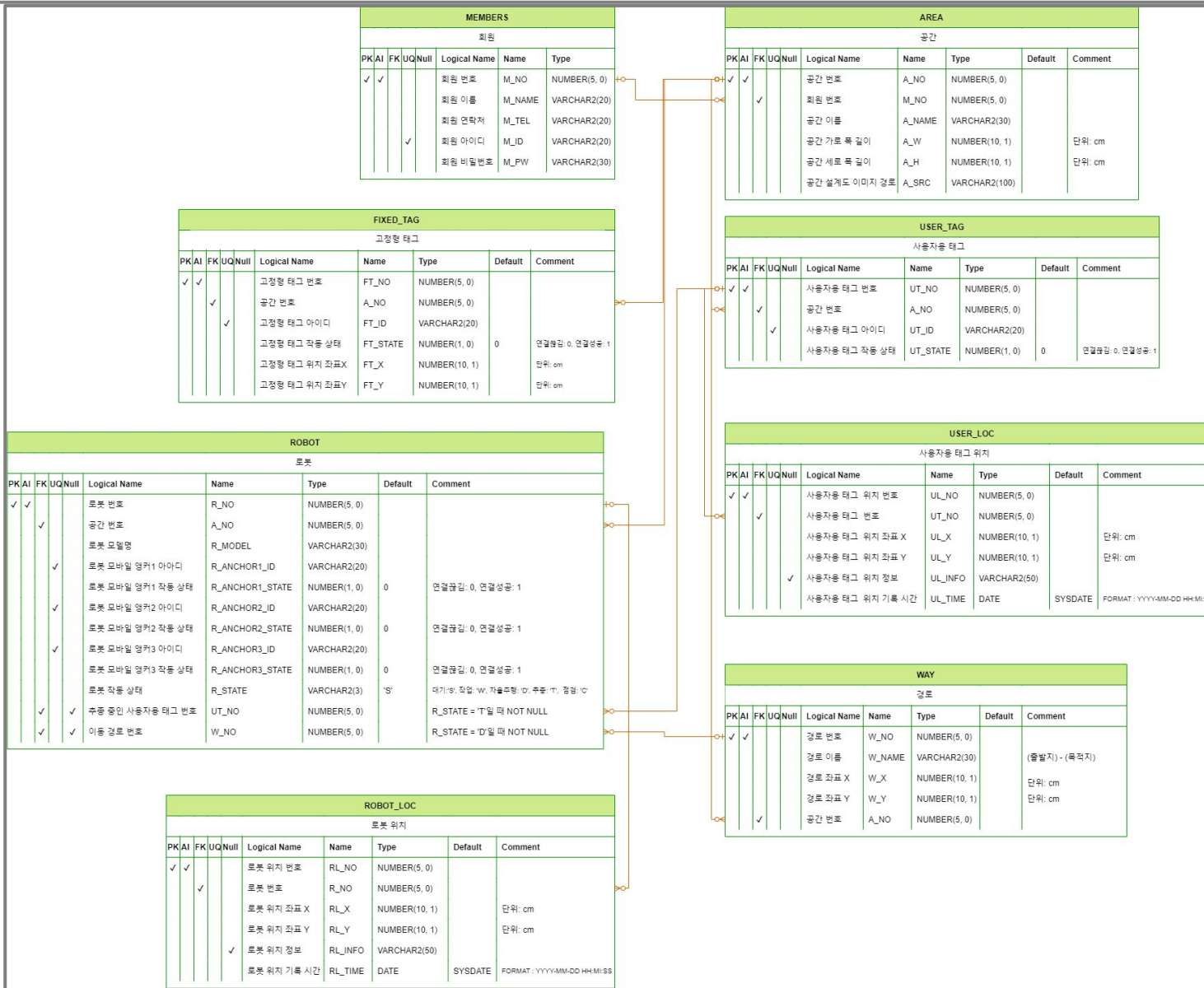
DCT

부제 : 로봇 통합 관리 시스템

2022. 12. 16

팀명 : 디셉티콘
팀원 : 유정협(팀장), 강민주, 문상건, 이호열, 정충원

OracleDB E-R Diagram



JinJa2 템플릿

```
<div id="header">
  <nav>
    <div class="navbar navbar-expand-lg navbar-dark bg-dark">
      <div class="container">
        <a class="navbar-brand" href="/home">DCT</a>
        <button aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation" class="navbar-toggler" data-bs-target="#navbarResponsive">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link active" href="/home">HOME</a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" href="/set">관리 및 설정</a>
            </li>
          </ul>
          <ul class="navbar-nav ms-md-auto">
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle mypage" data-bs-toggle="dropdown">
                <div class="dropdown-menu">
                  <a class="dropdown-item my" href="/mypage">내 정보</a>
                </div>
              </li>
            <li>
              <a class="nav-link" href="javascript:logout();"><span class="ba">
            </li>
          </ul>
        </div>
      </div>
    </div>
  </nav>
</div>
```

frm_body.html

frm_header.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <{% block head %}>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> DCT | {% block title %}{{ title }}{% endblock %}</title>
    <link rel="stylesheet" type="text/css" href="../static/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="../static/bootstrap.css">
    <link rel="stylesheet" type="text/css" href="../static/footer.css">
    <link rel="stylesheet" type="text/css" href="../static/common.css">
    <link rel="shortcut icon" type="image/x-icon" href="../static/img/favicon.ico">
    <script src="../static/common.js"></script>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="../static/bootstrap.bundle.min.js"></script>
    <script src="../static/prism.js"></script>
    <script src="../static/sweetalert.min.js"></script>
    <script src="../static/jquery.canvasResize.js"></script>
    <script src="../static/canvasResize.js"></script>
    <script src="../static/jquery.exif.js"></script>
  <{% block src %}>{{ endblock %}}
  <{% endblock %}>
</head>
<body>
  <div class="wrap bg-dark">
    <{% block wrap %}>{{ endblock %}}
  </div>
  <{% include 'frm_footer.html' ignore missing %}>
  <script>
    $(document).ready(() => {
      $(".nav-link").removeClass('active');
      let pageSrc = document.location.href.split('/');
      let now = ""
      if (pageSrc.includes('home')){
        now = "/home";
      }else{
        now = "/set";
      }
    });
  </script>
</body>
</html>
```

frm_footer.html

```
<div class="footer_wrap">
  <div class="footer-distributed">
    <div class="footer-left">
      <h3>Diception</h3>
      <div class="footer-center">
        <p class="footer-links">
          <a href="/home">홈</a>
          <a href="/set">관리 및 설정</a>
        </p>
      </div>
      <p class="footer-company-name">© PROJECT DCT Integrated Management System</p>
    </div>
    <div class="footer-center">
      <div>
        
        <p><span>경상북도 상주시 사벌면 엄암리 13-25</span>경상북도 스마트팜 혁신밸리<br>10시 ~ 17시 (월~금)</p>
      </div>
      <div>
        
        <p>000-000-0000</p>
      </div>
      <div>
        
        <p><a href="mailto:BT@company.com">BT@company.com</a></p>
      </div>
    </div>
    <div class="footer-right">
      <p class="footer-company-about">
        <span>About the company</span>
        <span>관주인공지능사관학교 AI모델링Track 사물지능을 배우고 있는학생들로 구성된 2차 프로젝트 팀으로 자율주행 스마트팜 로봇 통합 관리 시스템을 제작하고 있습니다.</span>
      </p>
    </div>
  </div>
</div>
```

프로젝트명

DCT

화면이름

로그인 화면

화면코드

LOGIN

유의사항

- 비 로그인 시, 서비스 이용 불가
- 뒤로가기를 막기 위해
`window.close();`
`window.open();`
으로 경로 이동
- 아이디 혹은 비밀번호 미입력으로 로그인 시,
미입력창 아래에 빨간글씨로
입력 안내 문구 출력
- 비회원은 오른쪽 아래
회원가입 버튼으로 회원가입
필요
- DB 미등록된 아이디-
비밀번호로 로그인시, 로그인
실패 alert창 open & 입력값
모두 지워짐

DCT

로그인

①

② 비밀번호 찾기 ③ 아이디 찾기 ④ 회원가입


화면설명

- ① 로그인
 - 아이디 입력
 - 비밀번호 입력
- ② 비밀번호 찾기
- ③ 아이디 찾기
- ④ 회원가입
 - 회원가입 페이지로 이동

아이디를 입력해 주세요.

비밀번호를 입력해 주세요.

로그인



로그인 실패

로그인 정보를 다시 입력해 주십시오.

JavaScript 코드

로그인 폼 체크

AJAX 통신

로그인 결과

- (성공) 페이지 링크
- (실패) alert창 호출

```
function checkLoginForm() {  
    let check = true;  
    if($('#input[name="M_ID"]').val() == ''){  
        $('#M_ID > .feedback').attr('class', "feedback invalid-feedback");  
        $('#M_ID > .feedback').text("아이디를 입력해 주세요.");  
        $('#input[name="M_ID"]').focus();  
        check = false;  
    }else{  
        $('#M_ID > .feedback').attr('class', "feedback valid-feedback");  
        $('#M_ID > .feedback').text("");  
    }  
    if($('#input[name="M_PW"]').val() == ''){  
        $('#M_PW > .feedback').attr('class', "feedback invalid-feedback");  
        $('#M_PW > .feedback').text("비밀번호를 입력해 주세요.");  
        $('#input[name="M_PW"]').focus();  
        check = false;  
    }else{  
        $('#M_PW > .feedback').attr('class', "feedback valid-feedback");  
        $('#M_PW > .feedback').text("");  
    }  
    if (!check) {return check;};  
    let formData = $('#form[name="login"]').serialize();  
  
    $.ajax({  
        url : "/login/result",  
        type : 'POST',  
        data : formData,  
    }).then((result)=>{  
        console.log(result);  
        if (result["result"]){  
            window.close();  
            window.open('/home');  
        }else{  
            swal("로그인 실패", "로그인 정보를 다시 입력해 주십시오.", "error").then(result =>{  
                if (result){  
                    window.close();  
                    window.open('/');  
                }  
            })  
        }  
    })  
}
```


Flask

Oracle DB 접속 및 해제 함수

```
import os

# DB 접속/
def conn_db() :
    os.putenv('NLS_LANG', '.UTF8') # 한글 지원하기
    conn = cx_Oracle.connect('dct', '1234', 'project-db-stu.ddns.net:1524/xes')
    cursor = conn.cursor()
    return conn, cursor

# DB 접속 해제
def close_db(conn, cursor) :
    cursor.close()
    conn.close()
```

DB 회원 데이터 ID, PW 조회 함수

```
# 로그인
def login_member(M_ID, M_PW) :
    conn, cursor = conn_db() # DB 접속

    query = f"""SELECT M_NO, M_NAME, M_ID FROM MEMBERS WHERE M_ID = :1 AND M_PW = :2"""
    cursor.execute(query, [M_ID, M_PW]) # 쿼리문 실행
    data = cursor.fetchone() # 조회 결과(한 튜플) 가져오기
    close_db(conn, cursor) # DB 접속 해제

    if data != None:
        session['user_no'] = data[0]
        session['user_name'] = data[1]
        session['user_id'] = data[2]
        return jsonify({'message': 'success', 'result': True})
    else :
        return jsonify({'message': 'success', 'result': False})
```

로그인 페이지 HTML 템플릿 뿌려주기

```
##### 로그인 #####
@app.route('/') # 로그인
def login() :
    if len(session) :
        return redirect(url_for('home'))
    else :
        return render_template('login.html', title="로그인")

@app.route('/login/result', methods=['POST']) # 로그인 + 입력값 체크
def login_result() :
    if request.method == 'POST' :
        M_ID = request.form['M_ID']
        M_PW = request.form['M_PW']
        return login_member(M_ID, M_PW)
```

프로젝트명

DCT

화면이름

회원가입

화면코드

SIGN UP

유의사항

- 각 입력 칸의 값이 주어진 조건을 만족하지 않을 시, 빨간 글씨로 안내 문구 출력
- 각 입력 칸의 값이 주어진 조건을 만족할 시, 초록색 글씨로 안내 문구 출력
- 아이디 중복 검사 사용가능한 검사에서 확인 버튼 클릭 시, 아이디 입력 값이 최종 아이디로 바뀜
- 아이디 중복 검사에서 취소 버튼 클릭 시, 기존에 아이디 입력 칸에 입력된 값은 지워짐

로그인 / 회원가입

회원 가입

- 1 이름
가
지
사용 가능한 이름입니다.
- 2 아이디
asdfqwe
중복확인
사용 가능한 아이디입니다.
- 3 비밀번호

사용 가능한 비밀번호입니다.
- 4 비밀번호 확인

비밀번호가 일치합니다.
- 5 휴대폰 번호
휴대폰번호 양식에 맞지 않습니다.(000-0000-0000)
- 6 회원 가입

아이디 유효성 검사

아이디를 다시 입력해주세요.
(아이디는 영문 대·소문자 6~15자 이내로 사용 가능합니다.)

취소 확인


아이디 중복 검사

사용 가능한 아이디
[asdfqwe]
이 아이디를 사용하시겠습니까?

취소 확인

회원 가입

이름
가
지



회원가입 완료

DCT의 회원이 되신 것을 축하드립니다!

휴대폰 번호
010-9646-4565
휴대폰번호가 입력되었습니다.

회원 가입

화면설명

- ① 이름 입력 칸
 - 2~6자 한글만 입력 가능
- ② 아이디 입력 칸
 - 6~15자 영문 대, 소문자만 입력 가능
- ③ 비밀번호 입력 칸
 - 영문 대,소문자, 특수문자, 숫자 적어도 한개 포함하여 8~16자로만 입력 가능
- ④ 비밀번호 확인 입력 칸
 - 비밀번호와 같은 값을 입력해야 함
- ⑤ 휴대폰 번호 입력 칸
 - 000-0000-0000형식으로 0은 숫자로만 입력 가능
 - 맨 앞 두 숫자는 01이어야 함
- ⑥ 회원 가입 버튼
 - 모든 양식 충족 시, 회원가입 완료

JavaScript 코드 (아이디 중복성 검사 & AJAX통신)

```
// 아이디 중복 검사
function confirm_id(){
  let ID = $('input[name="M_ID"]')
  ck_id1(ID);
}
```

아이디 중복 검사

```
// 아이디 정규식 검사
function ck_id1(ID){
  let regid = /^[A-Za-z0-9_-]{6,15}$/;
  if (!regid.test(ID)){
    swal({
      title: '아이디 유효성 검사',
      text: '아이디를 다시 입력해주세요.\n(아이디는 영문 대·소문자 6~15자 이',
      content: "input",
      buttons: {
        cancel: "취소",
        ok: "확인"
      },
    }).then((result) =>{
      switch(result){
        case "ok" :
          ID = $('.swal-content__input').val();
          ck_id1(ID);
        case "cancel" :
          $('input[name="M_ID"]').val("");
        default:
          $('input[name="M_ID"]').val("");
      }
    })
  }else{
    ck_id2(ID);
  }
}
```

아이디 정규식 검사
DB조회 후, 중복성 검사

//DB에서 ID 조회 >> 아이디 중복성 검사

```
function ck_id2(ID){
  let M_ID = {"M_ID" : ID};
  let set = ""

  $.ajax({
    type: "POST",
    url: "/signup/idck",
    contentType: 'application/json',
    dataType: 'JSON',
    data: JSON.stringify(M_ID),
  }).then((result)=>{
    if (result){
      swal({
        title: '아이디 중복성 검사',
        text: '아이디가 이미 사용 중인 아이디입니다.',
        buttons: {
          cancel: "취소",
          ok: "확인"
        },
      }).then((result)=>{
        switch(result){
          case "ok" :
            ck_id1("");
          case "cancel" :
            $('input[name="M_ID"]').val("");
          default:
            $('input[name="M_ID"]').val("");
        }
      })
    }else if (result == false && result != ""){
      swal({
        title: '아이디 중복성 검사',
        text: '아이디가 이미 사용 중인 아이디입니다.',
        buttons: {
          cancel: "취소",
          ok: "확인"
        },
      }).then((result)=>{
        switch(result){
          case "ok" :
            ck_id1("");
          case "cancel" :
            $('input[name="M_ID"]').val("");
          default:
            $('input[name="M_ID"]').val("");
        }
      })
    }
  })
}
```

아이디 중복 검사(DB 조회)를 위한
AJAX 통신

JavaScript 코드(회원가입 폼 검사 & AJAX 통신)

회원가입 폼
정규식 검사

```
function checkSignupForm () { //회원가입 form 확인
  let k_name = /^[가-힣]{2,6}$/;
  let e_name = /^[a-zA-Z]{2,30}$/;
  let regpw = /^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$%^&*()\-_+=\|\\|\[\]\{\};:~\`".,<>\|/?]).{8,15}$/;
  let regtel1 = /^01[179]{3}-[0-9]{3}-[0-9]{4}$/;
  let regtel2 = /^010-[0-9]{4}-[0-9]{4}$/;
```

```
  if(k_name.test($('input[name="M_NAME"]').val()) || e_name.test($('input[name="M_NAME"]').val())){
    $('#M_NAME > .feedback').attr('class', "feedback valid-feedback");
    $('#M_NAME > .feedback').text("사용 가능한 이름입니다.");
  }else{
    $('#M_NAME > .feedback').attr('class', "feedback invalid-feedback");
    $('#M_NAME > .feedback').text("이름은 한글 2~6자로 입력해 주세요");
    $('input[name="M_NAME"]').focus();
    return false;
  }
```

```
  if($('input[name=M_ID]').val() == $('input[name=id_confirm]').val()){
    if($('input[name="mpw_confirm"]').val() == $('input[name="mpw_confirm"]').val()){
      $('#mpw_confirm > .feedback').attr('class', "feedback valid-feedback");
      $('#mpw_confirm > .feedback').text("비밀번호가 일치합니다.");
    }else{
      $('#mpw_confirm > .feedback').attr('class', "feedback invalid-feedback");
      $('#mpw_confirm > .feedback').text("비밀번호가 일치하지 않습니다.");
      $('input[name="mpw_confirm"]').focus();
      return false;
    }
  }
```

```
  if(regtel1.test($('input[name="M_TEL"]').val()) || regtel2.test($('input[name="M_TEL"]').val())){
    $('#M_TEL > .feedback').attr('class', "feedback valid-feedback");
    $('#M_TEL > .feedback').text("휴대폰번호가 입력되었습니다.");
  }else{
    $('#M_TEL > .feedback').attr('class', "feedback invalid-feedback");
    $('#M_TEL > .feedback').text("휴대폰번호 양식에 맞지 않습니다.");
    $('input[name="M_TEL"]').focus();
    return false;
  }
```

```
  $.ajax({
    url : "/signup/result",
    type : 'POST',
    data : formData,
  }).then((result)=>{
    if (result["result"]){
      swal("회원가입 완료", "DCT의 회원이 되신 것을 축하드립니다!", "success").then(result=>{
        switch (result){
          case true :
            window.close();
            window.open('/');
          case false :
            default:
            window.close();
            window.open('/');
        }
      });
    }else{
      swal("회원가입 실패", "회원 가입 정보를 다시 입력해 주십시오. \n만약 이 문제가 계속 발생한다면 고객센터로 문의하십시오.");
      if (result){
        window.close();
        window.open('/signup');
      }
    }
  });
}
```

회원가입 정보 AJAX 통신

Flask

```
# 아이디 중복 체크
def check_id(M_ID) :
    conn, cursor = conn_db() # DB 접속

    query = f""""SELECT COUNT(*) FROM MEMBERS WHERE M_ID = :1""""
    cursor.execute(query, [M_ID]) # 쿼리문 실행
    data = cursor.fetchone() # 조회 결과(한 튜플) 가져오기
    close_db(conn, cursor) # DB 접속 해제

    if data[0] == 0:
        return jsonify({'message':'success', 'result': True})
    else :
        return jsonify({'message':'success', 'result': False})

# 회원가입
def signup_member(M_NAME, M_ID, M_PW, M_TEL) :
    try :
        conn, cursor = conn_db() # DB 접속

        query = f""""INSERT INTO MEMBERS (M_NAME, M_ID, M_PW, M_TEL) VALUE
        cursor.execute(query, [M_NAME, M_ID, M_PW, M_TEL]) # 쿼리문 실행
        conn.commit()

        return jsonify({'message':'success', 'result': True})
    except :
        return False
```

아이디 중복 체크 함수
(DB에서 입력한 아이디와 동일한
아이디 개수 조회)

회원가입 함수
(회원 정보 DB에 등록)

```
##### 회원 가입 #####
@app.route('/signup') # 회원 가입
def signup() :
    if len(session) :
        return redirect(url_for('home'))
    else :
        return render_template('signup.html', title="회원 가입")

@app.route('/signup/idck', methods=["POST"]) # 아이디 중복 체크
def ck_id() :
    if request.method == 'POST' :

        M_ID = request.get_json()['M_ID']
        return check_id(M_ID)

@app.route('/signup/result', methods=["POST"]) # 회원 가입 결과
def signup_result() :
    if request.method == 'POST' :
        M_NAME = request.form['M_NAME']
        M_ID = request.form['M_ID']
        M_PW = request.form['M_PW']
        M_TEL = request.form['M_TEL']
        return signup_member(M_NAME, M_ID, M_PW, M_TEL)
```

회원가입 템플릿 뿌리기
AJAX 통신

- 아이디 중복체크
- 회원가입 데이터 등록

프로젝트명

DCT

화면이름

홈

화면코드

HOME

유의사항

- 반응형 웹페이지로 사용자의 화면이 작아지면 메뉴바의 메뉴들이 사라지고 클릭하면 메뉴를 볼 수 있는 아이콘으로 바뀜
- 홈화면은 회원이 등록한 제일 첫번째 공간 설계도면 이미지가 출력됨
- 등록된 설계도면이 없을 때는 이미지 대신 안내문구가 출력됨
- 공간리스트 위에 마우스 hover시, 투명한 흰색으로 해당 행의 스타일이 바뀜



화면설명

- ① 메뉴 바
 - Home
 - 관리 및 설정
- ② 마이페이지 버튼
 - (회원 이름)님 반갑습니다.
- ③ 로그아웃
- ④ 공간 번호. 공간 이름
- ⑤ 공간 이름
- ⑥ 공간 설계도면
 - 로봇, 태그 실시간 위치 확인
- ⑦ 공간 리스트
 - 회원이 등록한 공간 리스트

HTML

```
{% extends "frm_body.html" %}
{% block head %}
{% block src %}
<link rel="stylesheet" href="https://cdn.jsdelivr.net/bxslider/4.2.12/jquery.bxslider.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/bxslider/4.2.12/jquery.bxslider.min.js"></script>
{% endblock %}
{{ super() }}
{% endblock %}
{% block wrap %}
{% include 'frm_header.html' ignore missing%}
{% if set_list is defined %}
    {% if a_info is defined %}
        {% set a_no = a_info[0] %}
        {% set a_name = a_info[1] %}
        {% set a_src = a_info[2] %}
    {% else %}
        {% set a_no = areas[0][0] %}
        {% set a_name = areas[0][1] %}
        {% set a_src = areas[0][2] %}
    {% endif %}
{% else %}
    {% set a_no = "" %}
    {% set a_name = "" %}
    {% set a_src = "" %}
{% endif %}
<div>
    <ol class="breadcrumb page_loc">
        <li class="breadcrumb-item"><a href="/home">공간</a>
        <li class="breadcrumb-item"><a href="#">{{a_no}}.</a>
    </ol>
</div>
<div class="content" style="padding:0.5rem;">
    <div class="flex-row">
        <div class="loc_info_wrap flex-auto flex-col">
            {% if a_info is defined %}
                <h3>{{ a_name }}</h3>
            {% endif %}
            <div class="area map flex-auto">
```

Jinja 템플릿 활용

- frm_body.html
- frm_header.html
- 공간 리스트 반복문으로 출력
- frm_footer.html

```
<div class="area map flex-auto">
    {% if set_list is defined %}
        
    {% else %}
        <h5> 설계도 사진이 존재하지 않습니다.<br>관리 및 설정 페이지에서 공간 설계도 사진을 등록해 주십시오. </h5>
    {% endif %}
</div>
</div>
<div class="area_list flex-auto">
    <h3>공간 리스트</h3>
    <table class="table table-hover">
        <thead>
            <tr class="table-secondary">
                <th>번호</th>
                <th>이름</th>
            </tr>
        </thead>
        <tbody>
            {% if set_list is defined %}
                {% for no, name, src in set_list['area'] %}
                    {% if a_no == no %}
                        <tr onclick="location.href='/home/{{ no }}'" class="row-active">
                            <td>{{ no }}</td>
                            <td>{{ name }}</td>
                        </tr>
                    {% else %}
                        <tr onclick="location.href='/home/{{ no }}'" class="table-dark">
                            <td>{{ no }}</td>
                            <td>{{ name }}</td>
                        </tr>
                    {% endif %}
                {% endfor %}
            {% else %}
                <tr>
                    <td colspan="2" class="center">등록된 공간이 없습니다.</td>
                </tr>
            {% endif %}
        </tbody>
    </table>
</div>
</div>
```

JavaScript 코드

설계도면 미등록 시, 화면에 안내 문구 출력해주는 코드

```
function imgError(page){
  if (page == 'home'){
    $('.area_map > img').replaceWith('<h5><br> 이미지가 존재하지 않습니다. <br> 관리 및 설정 페이지에서 공간 설계도 사진을 등록해 주십시오. </h5>');
    $('.area_map > h5').css('padding', '1rem');
    $('.area_map > h5').css('color', 'lime');
  }else if(page == 'setting'){
    $('.A_IMG > img').replaceWith('<h5><br> 이미지가 존재하지 않습니다. <br> 공간 설계도 사진을 등록해 주십시오. </h5>');
    $('input[name=A_SRC]').attr("value", "");
    $('.A_IMG > h5').css('color', 'lime');
  }
}
```

로그아웃 함수

```
//////////로그아웃//////////
function logout(){
  swal({
    title: "로그아웃 확인",
    text: "정말로 로그아웃 하시겠습니까?",
    buttons: {
      cancel: "취소",
      ok: "확인"
    }
  }).then((value)=>{
    switch(value){
      case "cancel" :
        window.location.reload();
      case "ok" :
        window.close();
        window.open('/logout');
      default :
        window.location.reload();
    }
  })
}
```


Flask

```
##### 홈 #####
@app.route('/home') # 홈
def home():
    if len(session):
        allAreas(session['user_no'])

        if set_list['area'] != [] :
            return redirect(url_for('showA_NO', a_no=set_list['area'][0][0]))
        else :
            return render_template('home.html', title="홈")
    else :
        return redirect(url_for('login'))
```

```
@app.route('/home/<a_no>') # 홈 when 공간번호 = a_no
def showA_NO(a_no):
    if len(session):
        a_info = selectArea(set_list['area'], a_no)

        if a_info != []:
            return render_template('home.html', title=a_info[0][1])
        else :
            return redirect(url_for('page_not_found'))
    else :
        return redirect(url_for('login'))
```

로그아웃

```
@app.route('/logout') # 로그아웃
def logout():
    if len(session):
        session.pop('user_no', None)
        session.pop('user_name', None)
        session.pop('user_id', None)
        return redirect(url_for('login'))
    else :
        return redirect(url_for('/'))
```

html 템플릿 뿌리기

- home 화면
- 공간 번호로
설계도면 불러오기

모든 공간 조회 by 회원 번호

```
def allAreas(m_no):
    conn, cursor = conn_db() # DB 접속

    query = f"""SELECT A_NO, A_NAME, A_SRC FROM AREA WHERE M_NO = :1 ORDER BY A_NO"""
    cursor.execute(query, [m_no]) # 쿼리문 실행
    area = cursor.fetchall() # 조회 결과(한 튜플) 가져오기
    set_list['area'] = area
    close_db(conn, cursor) #
```

전체 공간 정보 리스트에서 원

```
def selectArea(areas, a_no):
    a_no = int(a_no)
    for area in areas:
        if area[0] == a_no:
            return area
    return []
```

현재 Session의 회원번호로
전체 공간 정보 조회 함수

전체 공간 정보에서 공간 번호로
특정 공간 데이터 불러오는 함수

로그아웃
session 삭제

프로젝트명

DCT

화면이름

관리 및 설정

화면코드

SETTING

유의사항

- 화면 위쪽 회색 네비게이션 바의 텍스트마다 각 section id로 링크를 걸어 놓아 스크롤 다운 없이 빠른 이동 가능
- 전체 공간 리스트에서 원하는 공간 클릭 시, 그 공간에 대한 정보, 고정형 태그, 로봇, 사용자 태그, 경로가 출력
- 공간 수정, 등록 버튼 클릭 시, 왼쪽 아래 사진처럼 화면 출력
- 각 section별 등록 버튼 클릭 시, 오른쪽 아래 사진처럼 출력

DCT HOME 관리 및 설정 농장주님 반갑습니다. 로그아웃

공간 / 고정형 태그 / 로봇 / 사용자 태그 / 경로 1

28. 토마토

3 수정 4 삭제

5 공간 리스트 (총 26개) 6 추가

번호	이름
28	토마토
29	파프리카
30	딸기
41	김보디아
42	김보디아
43	tomato
44	tomato
45	potato
46	토마토
47	dct
48	토마토
49	robot
50	복숭아

고정형 태그 (총 0개)

번호	아이디	상태	설치 좌표1	설치 좌표2	7 추가
등록된 고정형 태그 없음					

로봇 설정(총 0개)

번호	모델명	상태	병커1 ID	병커1 상태	병커2 ID	병커2 상태	병커3 ID	병커3 상태	7 추가
등록된 로봇 없음									

공간 이름

토마토

설계도면 사진

파일 선택 선택된 파일 없음

사진 미리보기 사진 90도 회전 수정

상태 7 추가

설치 좌표1 설치 좌표2 취소

좌표 기록 등록

등록된 고정형 태그 없음

상태 병커2 ID 병커2 상태 병커3 ID 병커3 상태 취소

가 [] 대기 [] 등록

등록된 로봇 없음

상태 취소

대기 사용자 태그 스캔 등록

등록된 사용자 태그 없음

좌표 취소

[] 경로 그리기 등록

등록된 경로가 없습니다.

화면설명

- ① section 이동 링크
 - 공간
 - 고정형 태그
 - 로봇
 - 사용자 태그
 - 경로
- ② 현재 페이지 공간 정보
 - 공간번호. 공간 이름
- ③ 현재 공간 수정 버튼
- ④ 현재 공간 삭제
- ⑤ 공간 리스트
 - 현재 로그인 회원의 전체 공간 목록
- ⑥ 공간 등록 버튼
- ⑦ 태그 및 경로 등록 버튼

JavaScript 코드

```
// 이미지 미리보기 기능
function previewIMG(rotation){
  let file = $('input[name=file]')[0].files[0];

  if ($('#input[name=file]').val() == ""){
    $('#A_SRC > .feedback').attr('class', "feedback invalid-feedback");
    $('#A_SRC > .feedback').text("설계도면 사진을 등록해 주세요.");
  }else{
    $('#A_SRC > .feedback').attr('class', "feedback");
    $('#A_SRC > .feedback').text("");

    // 이미지 크기 조정
    function resizeIMG(file,
      $.canvasResize(file,
        width: 300,
        height: 0,
        crop: false,
        quality: 100,
        rotate: angle,
        callback: function(data, width, height) {
          let img = ``
          $(img).appendTo('#previewDiv')
        }
      )
    )
  }
  let angle = 0;
  if (rotation) {
    angle = rotation;
  }
}
```

공간 설계도면
미리보기 기능 함수

```
function addData(gubun) {
  if (gubun == 'AREA'){
    location.href="/set/new";
  }else if (gubun == 'FIXED_TAG'){
    $('#FixedTag_form').css('display', 'table-row');
    $('#set_FixedTag > div > table > tthead > tr > th:last-child > button').text('취소');
    $('#set_FixedTag > div > table > tthead > tr > th:last-child > button').attr('onclick', "closeADD('FIXED_TAG');");
    $('#set_FixedTag > div > table > tbody > tr.FixedTag_form > td:nth-child(2) > input').focus();
  }else if (gubun == 'ROBOT'){
    $('#Robot_form').css('display', 'table-row');
    $('#set_Robot > div > table > tthead > tr > th:last-child > button').text('취소');
    $('#set_Robot > div > table > tthead > tr > th:last-child > button').attr('onclick', "closeADD('ROBOT');");
    $('#set_Robot > div > table > tbody > tr.Robot_form > td:nth-child(2) > input').focus();
  }else if (gubun == 'USER_TAG'){
    $('#UserTag_form').css('display', 'table-row');
    $('#set_UserTag > div > table > tthead > tr > th:last-child > button').text('취소');
    $('#set_UserTag > div > table > tthead > tr > th:last-child > button').attr('onclick', "closeADD('USER_TAG');");
    $('#set_UserTag > div > table > tbody > tr.UserTag_form > td:nth-child(2) > input').focus();
  }else if (gubun == 'WAY'){
    $('#Way_form').css('display', 'table-row');
    $('#set_Way > div > table > tthead > tr > th:last-child > button').text('취소');
    $('#set_Way > div > table > tthead > tr > th:last-child > button').attr('onclick', "closeADD('WAY');");
    $('#set_Way > div > table > tbody > tr.Way_form > td:nth-child(2) > input').focus();
  }
}
```

취소 버튼 기능 함수

```
function editIMG(no){
  let check = true;
  let formData = new FormData();

  if ($('#input[name=A_NAME]').val() == ""){
    $('#A_NAME > .feedback').css('invalidCSS');
    $('#A_NAME > .feedback').text('공간 이름을 입력해 주십시오.');
```

공간 수정 및 등록 함수

```
    check = false;
  }

  $.ajax({
    url: `/set/${no}/edit_resu`,
    type: 'POST',
    data: formData,
    contentType: false,
    processData: false
  }).done((result)=>{
    console.log(result);
    if (number(result["result"])){
      $('#.show').css('display', 'block');
      $('#.edit').css('display', 'none');
      $('#previewDiv').empty();
      if (type == "update"){
        swal("공간 수정 성공", "공간이 업데이트 되었습니다.", "success").then(()=>{
          location.href="/set/${no}/edit_resu";
        })
      }
    }
  })
}
```

등록 버튼 기능 함수

Flask

```
from datetime import datetime
```

```
UPLOAD_FOLDER = './static/upload'
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}
```

```
# setting html에 출력할 정보 딕셔너리
```

```
set_list = {
    'area': [],
    'fixed_tag': [],
    'robot': [],
    'user_tag': [],
    'way': []
}
```

```
# 업로드 파일 확장자 확인
```

```
def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1]
```

```
# 업로드 파일명 설정
```

```
def set_filename(filename):
    ext = filename.rsplit('.', 1)[1]
    now = datetime.now().strftime('%Y%m%d_%H%M%S')
    return f"{now}.{ext}"
```

```
# 공간 등록
```

```
def insertArea(M_NO, A_NAME, filename, ori_file):
    result = {}
    result['type'] = "수정"
```

```
try :
```

```
    conn, cursor = conn_db() # DB 접속
```

```
    A_SRC = set_filename(filename)
```

```
    query = f"INSERT INTO AREA (M_NO, A_NAME, A_SRC) VALUES (:1, :2, :3)"
    cursor.execute(query, [M_NO, A_NAME, A_SRC]) # 쿼리문 실행
```

```
def updateArea(A_NO, A_NAME, filename="") :
```

```
    result = {}
    result['type'] = "수정"
```

```
try :
```

```
    conn, cursor = conn_db() # DB 접속
```

```
    if filename != "" :
```

```
        A_SRC = set_filename(filename)
```

```
        query = f"UPDATE AREA SET A_NAME=:1, A_SRC=:2 WHERE A_NO=:3"
```

```
        cursor.execute(query, [A_NAME, A_SRC, A_NO]) # 쿼리문 실행
```

```
    else :
```

```
        query = f"UPDATE AREA SET A_NAME=:1 WHERE A_NO=:2"
```

```
        cursor.execute(query, [A_NAME, A_NO]) # 쿼리문 실행
```

```
    ) # DB 접속 해제
```

```
    return
```

```
    C, A_NO]
```

```
    else
```

위) DB에 공간 등록 함수

아래) DB에서 공간 정보
수정 함수

위) - 이미지 파일 저장 경로 설정

- 이미지 파일 확장자 지정
- 공간 정보 담을 딕타입 변수 선언
- 이미지 파일 확장자 확인 함수
- 업로드 파일명 설정 함수

Flask

```
def selectByANO(A_NO) :  
    A_NO = int(A_NO)  
  
    conn, cursor = conn_db() # DB 접속  
  
    query = f"""SELECT A_NO, A_NAME, A_SRC FROM AREA WHERE A_NO = {A_NO}"""  
    cursor.execute(query, [A_NO]) # 쿼리문 실행  
    area = cursor.fetchall() # 조회 결과(한 튜플) 가져오기  
    set_list['area'] = area  
  
    if area != [] :  
        query = f"""SELECT FT_NO, FT_ID, FT_STATE, FT_NAME FROM FIXED_TAG WHERE A_NO = {A_NO}"""  
        cursor.execute(query, [A_NO]) # 쿼리문 실행  
        fixed_tag = cursor.fetchall() # 조회 결과(한 튜플) 가져오기  
        set_list['fixed_tag'] = fixed_tag  
  
        query = f"""SELECT R_NO, R_MODEL, R_STATE, R_NAME FROM ROBOT WHERE A_NO = {A_NO}"""  
        cursor.execute(query, [A_NO]) # 쿼리문 실행  
        robot = cursor.fetchall() # 조회 결과(한 튜플) 가져오기  
        set_list['robot'] = robot  
  
        query = f"""SELECT UT_NO, UT_ID, UT_STATE FROM USER_TAG WHERE A_NO = {A_NO}"""  
        cursor.execute(query, [A_NO]) # 쿼리문 실행  
        user_tag = cursor.fetchall() # 조회 결과(한 튜플) 가져오기  
        set_list['user_tag'] = user_tag  
  
        query = f"""SELECT W_NO, W_NAME, W_X, W_Y FROM WAY WHERE A_NO = {A_NO}"""  
        cursor.execute(query, [A_NO]) # 쿼리문 실행  
        way = cursor.fetchall() # 조회 결과(한 튜플) 가져오기
```

```
@app.route('/set/<a_no>/edit_result', methods=['POST']) # 설정도면 이미지 업로드  
def upload_file(a_no):  
    if len(session) : # 로그인 상태이면,  
        if request.method == 'POST': # Post방식으로 데이터 전송 받았을 때,  
            result = [] # 데이터 업데이트 및 등록 결과 반환 리스트  
  
            # 받은 데이터 변수에 넣기  
            a_name = request.form['A_NAME']  
  
            try :  
                file = request.files['file']  
                print(file)  
                angle = request.form['angle']  
                print(angle)  
  
                if file and allowed_file(file.filename): # 파일이 선택되어 있고, 이미지 파일일 때,  
                    filename = secure_filename(set_filename(file.filename))  
                    file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))  
  
                    if a_no == "new" : # 공간 등록  
                        result = insertArea(session['user_no'], a_name, filename)  
                    else : # 공간 업데이트  
                        os.remove(os.path.join(app.config['UPLOAD_FOLDER'], selectArea(areas, a_no)[2]))  
                        result = updateArea(a_no, a_name, filename)  
  
                    if result != False: # 등록 및 업데이트 성공했을 때,  
                        allAreas(session['user_no']) # set_list['area'] 업데이트  
                        return jsonify({'message': f'공간 설계도가 {result[0]}되었습니다.', 'result': True, 'fname': result[1], 'a_no': a_no})  
                    else :  
                        return jsonify({'message': f'공간 설계도 {result[0]} 실패', 'result': False})  
  
                else : # file이 이미지 파일이 아닐 때,  
                    return jsonify({'message': '이미지 파일을 선택해 주십시오.(확장자: jpg, png, jpeg)', 'result': False})  
  
            except :  
                print('there is no file')  
                result = updateArea(a_no, a_name)  
  
            if result != False:  
                allAreas(session['user_no'])  
                a_info = selectArea(set_list['area'], a_no)  
                return jsonify({'message': f'공간 설계도가 {result[0]}되었습니다.', 'result': True, 'fname': a_info[2], 'a_no': a_no})  
            else :  
                return jsonify({'message': f'공간 설계도 {result[0]} 실패', 'result': False})  
  
        else :  
            return jsonify({'message': '로그인한 사용자가 아닙니다.', 'result': False})
```

공간 정보 업데이트

특정 공간에 대한
공간, 고정형 태그, 로봇,
사용자 태그, 경로 정보 조회 함수