



# 시작하기

## 파이널 발표 대본

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c0b2b97e-9b61-4e68-93e6-02aa3b8474b9/%EC%9B%90%EB%B3%B8\\_%EB%B3%B5%EC%82%AC%EB%B3%B8.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c0b2b97e-9b61-4e68-93e6-02aa3b8474b9/%EC%9B%90%EB%B3%B8_%EB%B3%B5%EC%82%AC%EB%B3%B8.pdf)

## ppt 보여주기

1. 안녕하세요 ConnectionPool의 발표를 시작하겠습니다. ppt는 딱 5장으로 기획의도와 전체적인 기능을 소개드리고 바로 connectionPool을 보여드리도록 하겠습니다.
2. 저희 조는 수영 활성화를 위한 커뮤니티 사이트 ConnectionPool을 만들었습니다.
3. 사용자 기능은 회원가입, 로그인, 마이페이지 등의 회원서비스가 있습니다. 출석체크는 캘린더를 잘 만들어주셔서 후에 보여드리도록 하겠습니다.  
저희 ConnectionPool은 유료서비스나 판매상품이 없어 사이트 활동을 촉진하게 하는 쿠폰이나 적립금을 활용할 수 없어 자체적으로 포인트와 인기도 그리고 등급 제도를 만들었고요  
수영에 대한 정보나 사람을 나누는 자유게시판, 자신의 수영에 대한 이야기를 하는 수영 일기, 게시글이나 댓글에 대한 신고 기능과 공지사항을 게시하는 공지사항 게시판을 만들었습니다.
4. 또 사람뿐만 아니라 오프라인에서 번개 수영 모임을 가질 수 있는 소모임 메뉴와, 실제 같은 지역의 사람들끼리 정기적인 모임을 가질 수 있도록 하는 팀, 그리고 팀끼리 시합을 할 수 있는 배틀 기능을 구현했습니다. 소모임, 팀, 배틀 모두 참여자끼리 채팅을 할 수 있도록 했습니다.
5. 관리자 기능은 회원 관리와 메인 페이지의 배너 관리, 신고 내역들과 신고 당한 회원들에게 정지 및 탈퇴 패널티를 주는 신고관리, 마지막으로 공지관리가 있습니다. 그럼 바로 사이트를 보여드리겠습니다.

## 화면 보여주기 - 사용자화면

1. 메인페이지를 보시면 아까 팀을 이루어 배틀을 할 수 있다 말씀드렸잖아요 그 배틀 결과에 따른 순위로 승부욕을 자극합니다. 또한 클린한 홈페이지 운영을 위해 사용자들끼리 서로의 인기를 내리고 올리고 이를 랭킹으로 보여줍니다.
2. 마이페이지에서는 회원 정보를 변경할 수 있고 출석체크를 잘 만들어주셨는데 지금 보시면 포인트가 100인데 출석체크를 누르면 오늘 날짜에 체크표시가 되고 포인트가 올라갑니다. 그 외에 나의 팀 활동들을 조회할 수 있고 포인트 내역도 확인할 수 있습니다.
3. 다음으로 수영에 대한 정보와 잡담을 나누는 자유게시판인데 카테고리나 첨부파일을 첨부해 글을 올릴 수 있고 게시글이나 댓글을 신고 할 수 있습니다. 게시글이 신고될 경우 글은 바로 가려집니다.
4. 다음으로는 소그룹인데 원하는 조건에 따른 소그룹을 검색할 수 있고 신청하고 채팅참여까지 가능합니다.
5. 팀은 본인이 팀을 생성할 수 있고요 50포인트가 소요되며 파워등록할 경우 팀 모집화면에서 상단에 노출되며 같은 지역의 사람들이 팀에 신청할 수 있고 마찬가지로 채팅 참여가 가능합니다.
6. 배틀은 팀의 리더들이 배틀을 모집하거나 신청할 수 있고 당일날 오프라인으로 팀들이 만나 대결을 하고, 그 후 한팀이 승부 결과를 적고 상대 팀이 이를 승인하거나 이의제기를 하여 결과를 확정합니다.

---

## 화면 보여주기 - 관리자 화면

1. 관리자는 회원 정보를 분류에 따라 조회할 수 있고 이를 excel로 반출할 수 있습니다.
2. 배너는 status 컬럼을 이용해서 Y일 경우 홈페이지에 노출되게 H일 경우 숨김 메뉴로, 숨김메뉴에서 아예 삭제 시 N으로 바꾸어 숨김메뉴에서도 보이지 않게 했습니다.
3. 신고내역에서는 신고가 들어온 목록들이 나오고 이를 클릭해서 신고를 확정할지 철회할지 결정할 수 있으며 신고가 여러번 쌓인 회원들은 패널티를 받아 정지되거나 탈퇴됩니다.
4. 또한 공지사항은 관리자화면에서 따로 화면을 두번 만들지 않고 공지메뉴에서 바로 글 작성하기 버튼을 눌러 공지를 작성할 수 있도록 했습니다.

---

## 소그룹 메뉴만 코드리뷰 할 예정

1. 코드리뷰는 조건에 따라 소그룹을 조회해온 기능과 groupDetailView에 모집을 한 사람, 참여 신청한 사람, 참여 신청 완료한 사람, 그냥 아예 외부인 정말 많은 조건에 따라 화면이 바뀌는데요 그거를 어떻게 처리했는지 간단히 말씀드리고 끝내겠습니다.

2. 먼저 소그룹 조회 화면을 보시면 url에 groupList.mo로만 적혀있는데 requestParam의 defaultValue를 사용해서 그렇습니다. 페이지는 없을 시 1페이지로, groupArea는 모든 지역을 뜻하는 all로, 성별은 무관을 뜻하는 A로, 레벨은 무관을 뜻하는 E로, 모집중은 처음에 아무 생각 없이 정원이 모두 차지 않으면 모집중이지! 하고 groupMember로 담아왔는데요 소모임 시작 시간이 지나도 모집 완료였습니다,,, 두 가지를 한번에 groupMember 필드에 담아와 처리했습니다.

```
/**
 * selectGroupList : 아래 조건에 맞추어 group 리스트를 select하는 메소드
 * @param currentPage : 요청하는 페이지
 * @param groupArea : 소그룹 지역
 * @param gender : 소그룹 성별 조건
 * @param level : 소그룹 레벨 조건
 * @param groupMember : 소그룹 모집인원
 */
@RequestMapping("groupList.mo")
public ModelAndView selectGroupList(ModelAndView mv,
    @RequestParam(value="cpage", defaultValue="1") int currentPage,
    @RequestParam(value="groupArea", defaultValue="all") String groupArea,
    @RequestParam(value="gender", defaultValue="A") String gender,
    @RequestParam(value="level", defaultValue="E") String level,
    @RequestParam(value="groupMember", defaultValue="A") String groupMember) {

    Group group = new Group();
    group.setGroupArea(groupArea);
    group.setGender(gender);
    group.setLevel(level);
    group.setGroupMember(groupMember);

    PageInfo pi = Pagination.getPageInfo(moimService.selectGroupCountList(group), currentPage, 10, 9);
    //System.out.println(pi);
    mv.addObject("pi", pi).addObject("groupList", moimService.selectGroupList(pi, group)).addObject("group", group).setViewName("moim/groupListView");

    return mv;
}
```

해당 조건들을 Group Vo에 담아서 먼저 조건에 맞는 소그룹의 갯수를 구하는 selectGroupCountList 메소드를 먼저 실행하고 날린 sql문은 아래와 같습니다.

```
<select id="selectGroupCountList" parameterType="group" resultType="_int">
    SELECT
        COUNT(*)
    FROM
        "GROUP"
    WHERE
        GROUP_STATUS = 'Y'
    <if test='groupArea != "all"'>
        AND
            GROUP_AREA = #{ groupArea }
    </if>
    <if test='gender != "A"'>
        AND
            GENDER = #{ gender }
    </if>
    <if test='level != "E"'>
        AND
            "LEVEL" = #{ level }
    </if>
    <if test='groupMember != "A"'>
        AND
            (GROUP_MEMBER - 1) <![CDATA[>]]>(SELECT
                COUNT(*)
            FROM
                APPLY "A"
            WHERE
                A.MOIM_NO = GROUP_NO
            AND
                ACCEPT_YN = 'Y'
            )
    AND
        TO_CHAR(START_TIME, 'YYYYMMDDHH24MISS') <![CDATA[>= TO_CHAR(SYSDATE, 'YYYYMMDDHH24MISS')]]>
    </if>
</select>
```

조건들이 전체거나 무관일 경우 아예 조건이 걸리지 않아 모든 소그룹 리스트가 조회되고요, 조건이 있을 경우는 조건에 맞는 것만 select되도록 했습니다. 설명이 필요할 것 같은 조건은 마지막에 groupMember에 대한 것인 거 같은데요, GROUP\_MEMBER는 참여 인원입니다. 본인을 제외하기 위해 -1을 해줬고요, 이것이 뒤에 서브쿼리문보다 클 때만 모집중이다. 이런 조건식인데요 서브쿼리문은 해당 소그룹의 식별값을 조건으로 또 수락된 신청 내역만 조회해서 참여자의 수를 구한 것입니다. 모집하는 인원이 신청한 인원보다 많을 때만 모집중인 것인 것이고 또 아래 And로 이어지는 조건문은 소그룹의 시작시간이 오늘날보다 작을 때 즉 아직 시작되지 않은 모임만 모집중으로 따져서 수를 구해왔습니다.

```

<select id="selectGroupList" parameterType="group" resultMap="groupResultMap">
    SELECT
        G.GROUP_NO
        ,M.NICKNAME
        ,G.MEM_NO
        ,G.GROUP_AREA
        ,A.KOR_AREA
        ,G.CHANGE_NAME
        ,G.GROUP_TITLE
        ,CASE WHEN G.GENDER = 'A' THEN '무관'
              WHEN G.GENDER = 'F' THEN '여자만'
              ELSE '남자만' END "GENDER"
        ,CASE WHEN "LEVEL" = 'E' THEN '무관'
              WHEN "LEVEL" = 'B' THEN '초급'
              WHEN "LEVEL" = 'M' THEN '중급'
              ELSE '고급' END "LEVEL"
        ,CASE WHEN TO_CHAR(G.POWER_DURATION, 'YYYYMMDD') <![CDATA[>= TO_CHAR(SYSDATE, 'YYYYMMDD')]]> THEN '파워'
              ELSE '일반' END "POWER_DURATION"
        ,TO_CHAR(G.START_TIME, 'YYYY/MM/DD hh24:mi') "START_TIME"
        ,TO_CHAR(G.END_TIME, 'YYYY/MM/DD hh24:mi') "END_TIME"
        ,CASE WHEN GROUP_MEMBER - 1 <![CDATA[<= (SELECT
                                                    COUNT(*)
                                                    FROM
                                                    APPLY "A"
                                                    WHERE
                                                    A.MOIM_NO = G.GROUP_NO
                                                    AND
                                                    ACCEPT_YN = 'Y'
                                                    )]]> THEN '모집마감'

        ELSE '(((SELECT
                    COUNT(*)
                    FROM
                    APPLY "A"
                    WHERE
                    A.MOIM_NO = G.GROUP_NO
                    AND
                    ACCEPT_YN = 'Y') + 1)|| '/' || G.GROUP_MEMBER || ')' END "GROUP_MEMBER"

    FROM
        "GROUP" G
    JOIN
        MEMBER "M" ON(G.MEM_NO = M.MEM_NO)
    JOIN
        AREA "A" ON(G.GROUP_AREA = A.ENG_AREA)
    WHERE
        GROUP_STATUS = 'Y'
    <if test='groupArea != "all"'>
        AND
        GROUP_AREA = #{ groupArea }
    </if>
    <if test='gender != "A"'>
        AND
        G.GENDER = #{ gender }
    </if>
    <if test='level != "E"'>
        AND
        "LEVEL" = #{ level }
    </if>
    <if test='groupMember != "A"'>
        AND
        (GROUP_MEMBER - 1) <![CDATA[>]]>(SELECT
                                                    COUNT(*)
                                                    FROM
                                                    APPLY "A"
                                                    WHERE
                                                    A.MOIM_NO = GROUP_NO
                                                    AND
                                                    ACCEPT_YN = 'Y'
                                                    )
        AND
        TO_CHAR(START_TIME, 'YYYYMMDDHH24MISS') <![CDATA[>= TO_CHAR(SYSDATE, 'YYYYMMDDHH24MISS')]]>
    </if>
    ORDER
        BY G.POWER_DURATION, G.CREATE_DATE DESC
</select>

```

수를 구해왔으니 실제 List도 select해오는 메소드를 사용했습니다. from 전인 select절은 화면에 바로 뿌려질 수 있도록 값을 가공해서 조금 길고요 조건은 selectCountList와 같아서

다시 controller로 돌아오면 해당 조건이 담긴 group Vo를 또 화면에 보내주는데요 페이징 바를 조건에 따라 만들어주기 위함입니다.

보시면 조회 group에서 필드값을 빼서 서브쿼리를 완성하고 있는데요 조건 없이 검색해서 빈 group이어도 아까 보셨듯이 controller에서 defaultValue가 있으니 잘 처리됩니다.

1. 로그인 된 사람이 이 글을 올린 사람일 때

- ## 2. 로그인 된 사람이 이 글을 올린 사람이 아닐 때

- i. 모집 끝났을 때

- ii. 모집종일 때

- iv. 신청 내역이 있을 때(로그인이 되어 있고 이 글을 작성한 사람이 아니면 무조건 신청 내역을 조회해온다.)

1. 신청을 했고 수락됐을 때 → 참여 취소하기 / 채팅하기
2. 신청을 했고 대기중일 때 → 신청 취소하기