

**FINAL PROJECT**

# PMS

*Project Management System*



이준석

이연학

유재인

김하나

김진수

김형규

전현규

# CONTENTS

- 001.** Topic Selection
  - 002.** Requirements Documentation
  - 003.** Development Environment
  - 004.** Function Flow
  - 005.** DB Construction
  - 006.** Screen Design
  - 007.** Review
  - 008.** Software Quality Assurance
- 



## *Topic Selection*

# **Topic Selection**

- 자본주의 사회에서 기업의 목표는 최대의 이윤을 남기는 것이다. 그러나 결과에만 맞추어진 초점으로 인하여 그 공정(process)이 무시되었을 때의 문제점은 이천 물류창고 참사 등 각종 인재의 형태로 우리 곁에 다가오고 있다.
- **효율적인 목표 달성과 체계적인 공정 관리**라는 두 마리 토끼를 잡기 위한 선택인 PMS 시스템을 개발하여 프로젝트의 진행상황, 소요시간, 비용을 체계적으로 관리함으로써 프로젝트 진행 효율을 향상시키고 한다.



# [ **PMS?** ]

(Project Management System)

기업이 운영하는 다양한 프로젝트의 정보개요, 진척현황, 인력배정, 원가분석 등의 정보를 한 눈에 조회할 수 있어 기업의 전략추진과 경영혁신, 운영관리를 체계적으로 관리하여 업무의 생산성과 효율성을 향상시키는 솔루션

# PMBOK

(Project  
Management  
Body  
of  
Knowledge)

1. BOK(Body of Knowledge)란 특정 직무 또는 전문 분야에서 활용되는 개념, 원리, 규칙, 표준, 용어, 기술, 도구, 기법 등을 체계적으로 정리한 결과물을 가리킨다.
2. 19세기 중반부터 프로젝트 관리 전문가들이 프로젝트의 관리와 관련한 BOK를 확립하려는 노력을 기울이기 시작했는데 이를 PMBOK라고 한다.
3. PMI에서 발간한 PMBOK Guide는
  - 5가지 프로세스 그룹
  - 10가지 프로젝트 관리의 지식영역을 정의한다.



# PMBoK KNOWLEDGE AREAS



Integration  
Management



Scope  
Management



Schedule  
Management



Cost  
Management



Procurement  
Management



Quality  
Management



Stakeholder  
Management



Resource  
Management



Communications  
Management



Risk  
Management



# *Requirements Documentation*

# Requirements Documentation



대분류	주요기능	세부사항	웹/앱	담당자	우선순위	진행률	비고
사원정보	사원등록	개인 정보 입력(사원번호/이름/비밀번호/희망연락처)	공통	김진수	5	100%	-
	로그인	사원번호/비밀번호 입력 -> 로그인	공통		1	100%	-
	사원번호찾기	이름/이메일 -> 일치하는 사원번호 출력	공통		2	100%	-
	비밀번호찾기	사원번호/이름/이메일(인증번호) -> 비밀번호 재설정(변경)	공통		3	100%	-
	로그아웃	로그아웃 -> 메인화면 이동 / 로그인 세션 정보 삭제	공통		4	100%	-
	개인정보	내 프로젝트 팀원 확인	공통		7	100%	-
		개인정보 -> 입력 정보 확인 및 수정	공통		6	100%	-
프로젝트총괄관리	CTO 선정	CEO - 직원리스트 - CTO 설정 및 변경	공통	김형준	1	100%	-
	프로젝트 추가	프로젝트 추가(팀원이 아닌 사원중 한명 팀장 지정)	월	전현규	1	100%	-
	프로젝트 총괄	CEO / CTO 대시보드 - 전체 프로젝트 진행률 등	월	유재인	1	100%	-
프로젝트통합관리	대시보드	PM 대시보드 - 프로젝트 진행률(개인별) 등	월		2	100%	-
		팀원 대시보드 - 해당 프로젝트 전체 진행률 / 일정 등	월		3	100%	-
	PM 선정	CEO / CTO - 직원리스트	공통	김형준	2	100%	-
		CEO / CTO - 직원리스트 - 프로젝트 팀장 설정 및 변경	공통		3	100%	-
	업무관리	PM - 업무 리스트	공통	이연학	1	100%	-
		PM - 업무 리스트 - 새 업무 추가 (시작일/종료일/담당자 지정 등)	공통		2	100%	-
		PM - 업무 리스트 상세 - 반려사유, 결재/반려 처리(결재 신청된 사항만), 삭제	공통		3	100%	파일업로드 추가
		팀원 - 업무 리스트(내업무)	공통		4	100%	-
		팀원 - 업무 리스트 상세 - 진행률 갱신 / 결재 신청	공통		5	100%	-
프로젝트일정관리	일정관리	CEO / CTO - 프로젝트 간트챠트	월	전현규	2	100%	-
		PM - 프로젝트 간트챠트	월	전현규	3	100%	-
		팀원 - 프로젝트 간트챠트	월	전현규	4	100%	-

# Requirements Documentation



대분류	주요기능	세부사항	웹/앱	담당자	우선 순위	진행률	비고
프로젝트 자원관리	인적자원관리	사원리스트	공통	김형준	4	100%	-
		팀장권한 프로젝트 팀원 설정 및 변경	공통		5	100%	-
프로젝트 의사소통관리	공지사항	CEO / CTO / PM / 팀원 - 공지사항 게시판 리스트	공통	김하나	1	100%	-
		CEO / CTO / PM - 공지사항 작성 페이지	공통		3	100%	-
		CEO / CTO / PM - 공지사항 상세페이지 (수정/삭제 가능)	공통		7	100%	-
		팀원 - 공지사항 상세페이지 (조회만 가능)	공통		4	100%	-
	회의록	CEO / CTO / PM / 팀원 - 회의록 게시판 리스트	공통		2	100%	-
		PM / 팀원 - 회의록 게시판 상세 (수정/파일 업로드 가능)	공통		6	100%	-
		CEO / CTO - 회의록 게시판 상세 (조회만 가능)	공통		8	100%	-
		PM / 팀원 - 회의록 작성 페이지 (파일 업로드 가능)	공통		4	100%	-
프로젝트 리스크관리	이슈리스트	CEO / CTO / PM / 팀원 - 리스크관리 - 이슈리스트 리스트	공통	이준석	1	100%	-
		CEO / CTO / PM / 팀원 - 리스크관리 - 이슈리스트 상세 (댓글 가능)	공통		3	100%	파일업로드 추가
		PM / 팀원 - 리스크관리 - 이슈리스트 추가	공통		2	100%	-
		PM - 리스크관리 - 이슈리스트 상세 (해결방안/댓글 가능)	공통		4	100%	-



## 전체진행일정

항목	05/08	05/15	05/22	05/29	06/05
화면 설계	80 %	100 %			
데이터 베이스 설계	60 %	100 %			
데이터베이스 생성		80 %	100 %		
화면 구현 웹		20 %	70 %	100 %	
화면 구현 앱			40 %	100 %	
통합테스트	-	-	-	-	100%





## 권한별 접근 기능

권한	권한별 접근 기능
CEO/CTO	대시보드 - 대시보드를 통한 각 프로젝트 간트차트(PM), 각 게시판 접근 인적자원관리(CTO선정, PM선정)
PM	대시보드/간트차트/업무리스트/각 게시판/인적자원관리(팀원 추가/삭제)
팀원	대시보드/간트차트/업무관리/각 게시판
인사관리자	인적자원관리
Admin	모든 게시판 및 권한





## *Development Environment*

# **Development Environment**

## **Language**

1. JAVA
2. JSP
3. JAVASCRIPT
4. CSS
5. HTML
6. Ajax
7. JSON
8. JSTL

## **Development Tools**

1. Oracle eclipse
2. eXERD
3. Android Studio

## **WAS**

1. Apache Tomcat 9.0

## **DBMS**

1. Oracle SQL 11g

## **Javascript Library**

1. jQuery

## **Framework**

1. Spring

## **Bootstrap**

1. Bootswatch - Minty
2. Dashio - admin



# Technology Used

## 기반기술

- Spring MVC
- Mybatis
- Ajax & Jquery
- Vue.js
- Android Studio

## 핵심기술

- Gantt chart - Highcharts

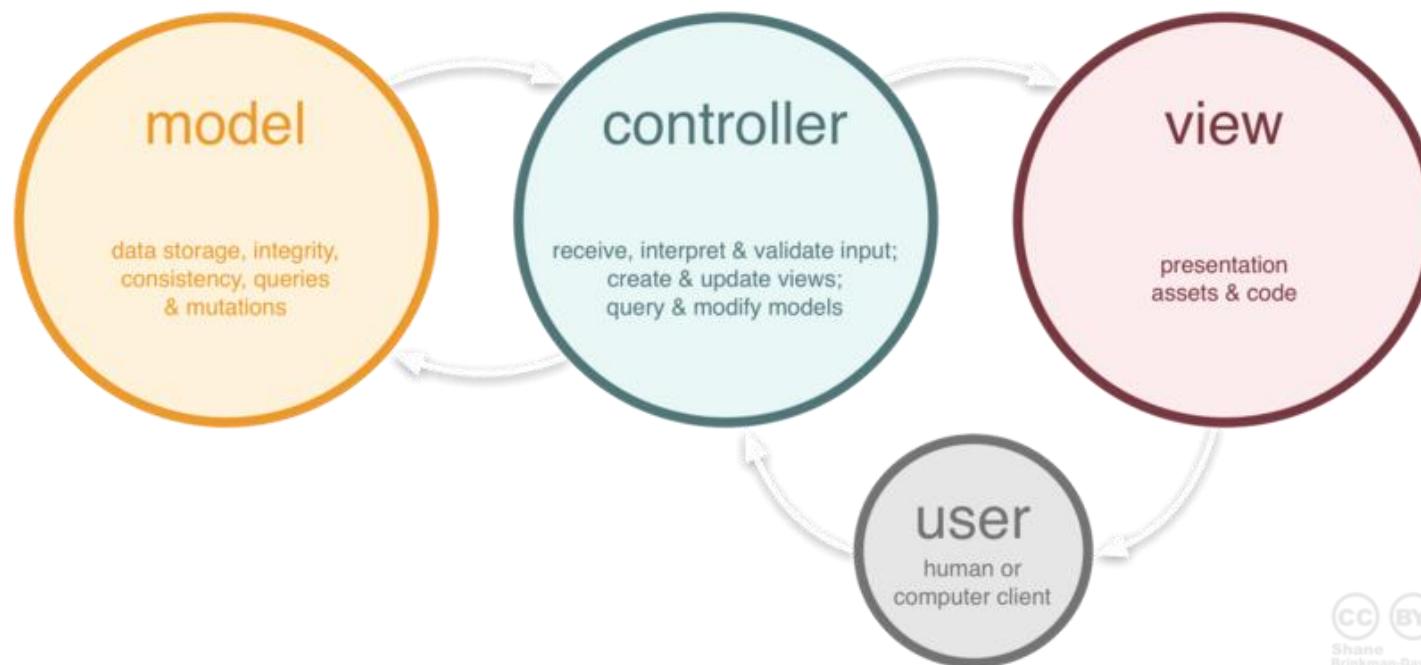
## 구현기술

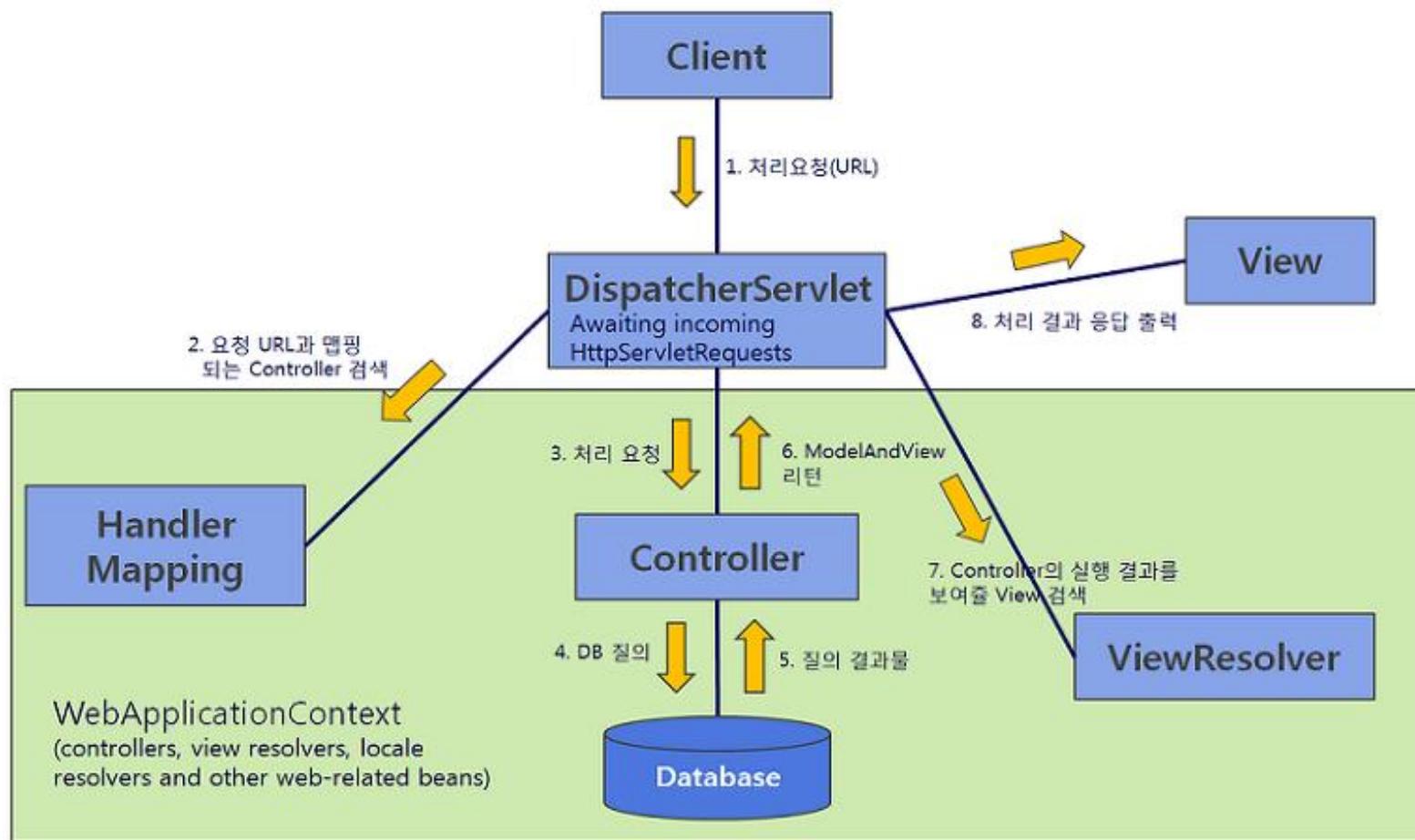
- EasyPieChart
- sweetAlert2
- Multipart Resolver(File Upload)
- mail Sender
- chatting

### Spring MVC

MVC패턴 = 디자인패턴 중 하나

디자인 패턴이란, 프로그램이나 어떤 특정한 것을 개발하는 중에 발생했던 문제점들을 정리해서 상황에 따라 간편하게 적용해서 쓸 수 있는 것을 정리하여 특정한 "규약"을 통해 쉽게 쓸 수 있는 형태로 만든 것





번호	제목	작성일	조회수
11	회의 공지	2020. 5. 11	10
10	이슈 발생시 바로 이슈리스트 등록 바랍니다	2020. 5. 8	0
9	주말 과제 공지	2020. 5. 8	15
8	업무 분담 숙지 바랍니다	2020. 5. 7	0
7	1주차 발표에 대한 피드백 회의 공지	2020. 5. 7	10

Previous 1 2 3 Next

## dispatcher

```
<context:component-scan base-package="project06">
    <context:include-filter type="annotation"
        expression="org.springframework.stereotype.Controller"/>
    <context:include-filter type="annotation"
        expression="org.springframework.stereotype.Service"/>
    <context:include-filter type="annotation"
        expression="org.springframework.stereotype.Repository"/>
</context:component-scan>
```

## Ctrl

```
@RequestMapping(params="method=list")
public String list(@ModelAttribute("nsch") NoticeSch sch,
    Model d, HttpServletRequest request) {
    HttpSession session = request.getSession();
    if(session.getAttribute("mno")==null) {
        return "WEB-INF\views\main\login.jsp";
    }
    d.addAttribute("nlist", service.list(sch));
    return "WEB-INF\views\main\noticeList.jsp";
}
```

## model

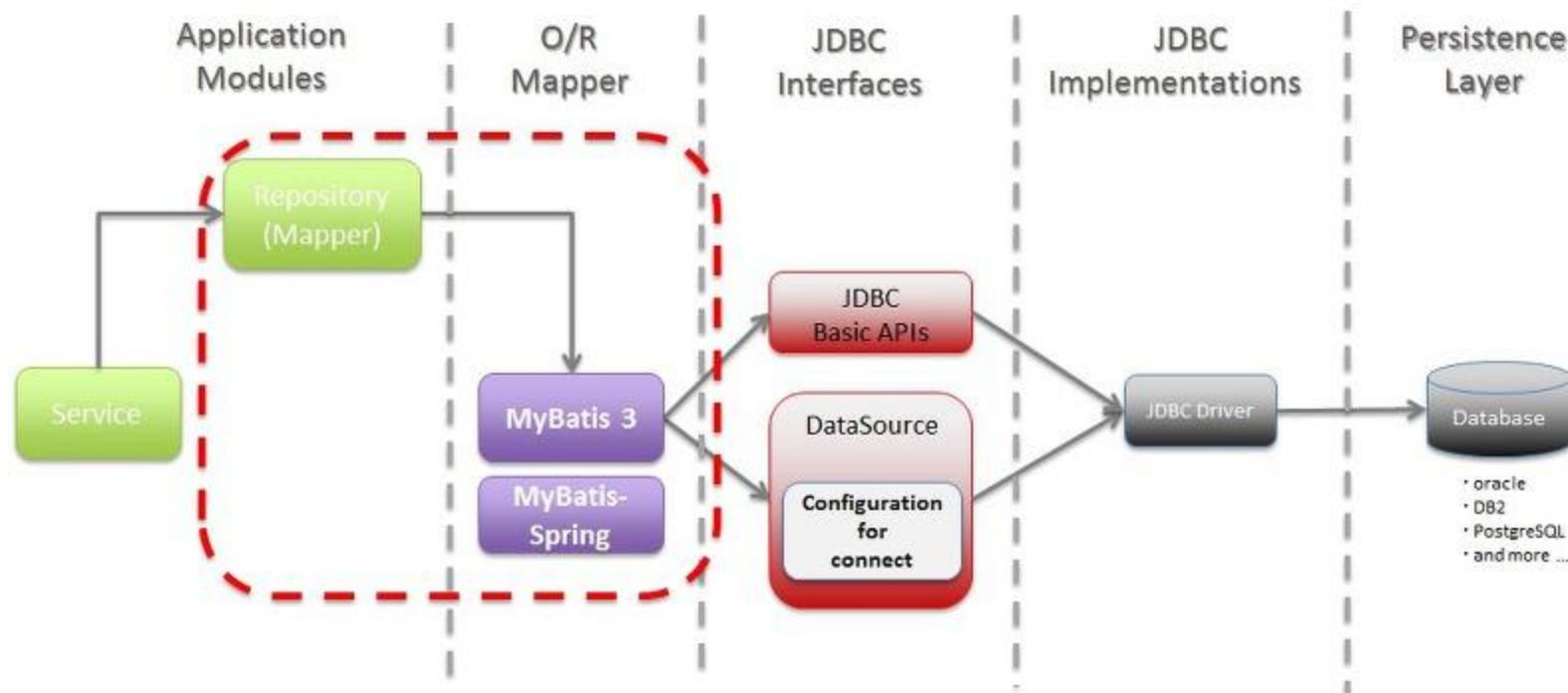
```
public interface NoticeDao{
    public int totCnt(NoticeSch sch);
    public ArrayList<Notice> list(NoticeSch sch);
```

## jsp

```
<tbody>
<c:forEach var="notice" items="${nlist}">
    <tr onclick="javascript:go(${notice.nno})">
        <td style="text-align: center;">${notice.nno}</td>
        <td style="padding-left:70px;">${notice.title}</td>
        <td style="text-align: center;"><fmt:formatDate
            value="${notice.wdate}" type="date"/></td>
        <td style="text-align: right;padding-right:70px;">
            ${notice.views}</td>
    </tr>
</c:forEach>
</tbody>
```

## MyBatis

- 자바 퍼시스턴스 프레임워크의 하나 – xml 서술자나 annotation을 사용
- 복잡한 쿼리나 다이나믹한 쿼리에 강하며 / 프로그램 코드와 SQL 쿼리의 분리로 코드의 간결성 및 유지보수성이 향상된다 는 장점이 있음



번호	제목	작성일	조회수
11	회의 공지	2020. 5. 11	10
10	이슈 발생시 바로 이슈리스트 등록 바랍니다	2020. 5. 8	0
9	주말 과제 공지	2020. 5. 8	15
8	업무 분담 숙지 바랍니다	2020. 5. 7	0
7	1주차 발표에 대한 피드백 회의 공지	2020. 5. 7	10

[Previous](#) [1](#) [2](#) [3](#) [Next](#)

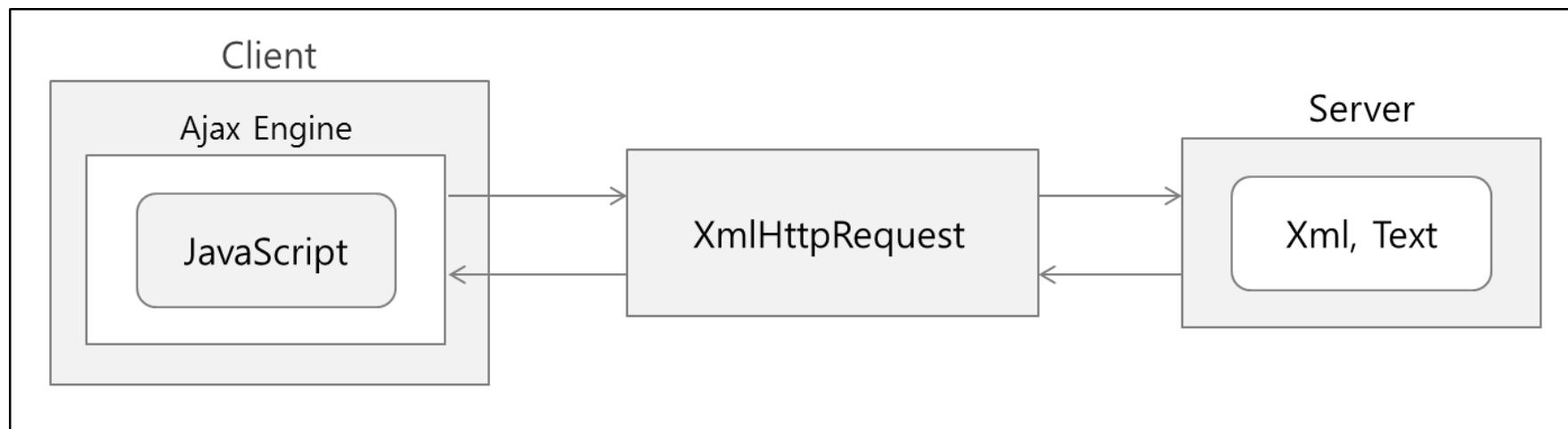
## Mapper

```
<select id="list" resultMap="noticeResult" parameterType="nsch" >
    SELECT *
    FROM(
        select ROW_NUMBER() OVER(ORDER BY nno desc) cnt, p.*
        FROM pmsnotice p
    WHERE 1=1
    <if test="title!=null and title!="">
        and title like '%'||#{title}||'%'
    </if>
    ORDER BY nno DESC )
    WHERE cnt BETWEEN #{start} AND #{end}
</select>
```

### Ajax Jquery

Ajax = Asynchronous JavaScript and XML의 약어

- Web에서 화면을 갱신하지 않고 Server로부터 Data를 가져오는 방법. Ajax의 동작원리는 Browser에서 서버로 보낼 Data를 Ajax Engine을 통해 Server로 전송한다. 이 때 Ajax Engine에서는 JavaScript를 통해 DOM을 사용하여 XMLHttpRequest(XHR) 객체로 Data를 전달, 이 XHR을 이용해서 Server에서 비동기 방식으로 자료를 조회해 올 수 있다. Server에서 Data를 전달 할 때 화면전체의 HTML을 전달하지 않고 Text 또는 Xml형식으로 Browser에 전달한다.



### Ajax Jquery

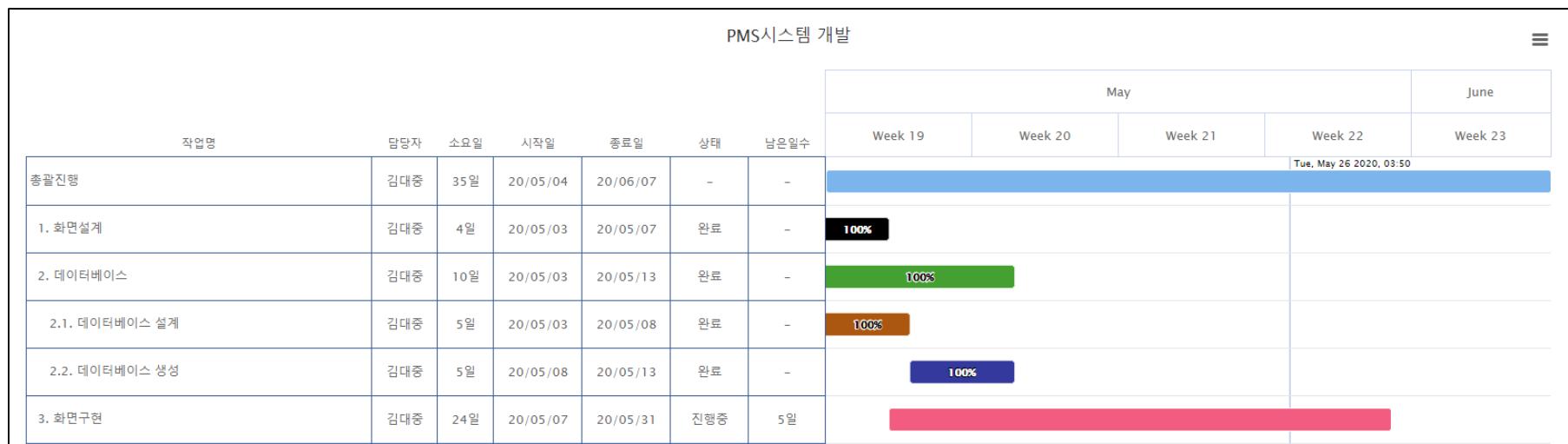
- JQuery = 오픈 소스 기반의 자바스크립트 라이브러리로, 자바스크립트를 손쉽게 활용할 수 있게 도와준다. JQuery를 이용하면 짧고 단순한 코드로도 웹페이지에 다양한 효과나 연출을 적용할 수 있다.
- Ajax의 기본 Method를 이용해서 Server와 통신을 하면 상당히 복잡하다. 이는 XMLHttpRequest를 직접 사용하기 때문인데, jQuery를 이용하면 100줄 정도의 Source를 몇 줄 만으로 간단하게 Server와 Data를 주고받을 수 있다. 사용방법은 JavaScript에서 jQuery 문법을 사용해서 간단하게 구현할 수 있다.

```
$.ajax({
    type:"post", // Http 요청방식(GET, POST)
    // 요청이 전송되는 URL이 포함된 문자열
    url:"${path}/chart.do?method=ajaxdata",
    data: $("form").serialize(), // Http요청 후 return하는 값
    dataType:"json", // return받은 데이터의 Type을 지정
    // 성공적으로 값을 서버로 보냈을 경우 처리하는 코드
    success:function(data){
        console.log("데이터 불러오기 성공")
    },
    // 값을 제대로 받아오지 못했을 때 처리하는 코드
    error:function(err){
        console.log("ajax처리 에러");
        console.log(err);
    }
});
```

## Ajax Jquery

Jquery문법으로 Ajax를 사용해서 데이터를 받아와서 차트  
데이터로 출력하는 예제

```
$.ajax({
    type:"post",
    url:"${path}/chart.do?method=ajaxdata",
    data: $("form").serialize(),
    dataType:"json",
    success:function(data){
        var clist = data.chartlist;
        var dataInfo = [];
        $.each(clist,function(idx, chart){
            dataInfo.push({
                name: chart.tname, id: ""+chart.tno+"",
                parent: ""+chart.refno+"",
                start: today+((chart.sdate+1) * day),
                end: today+((chart.edate+1) * day),
                completed : chart.prog,
                owner : ""+chart.name+"", y : idx+1,
                collapsed : true, description : chart.level,
            });
        });
        ...
        Highcharts.ganttChart('chartpm', cInfo);
    }
});
```



### Vue.js

- 사용자 인터페이스 생성 Progressive Framework
- Vue는 점진적인 채택이 가능하게 설계되어 단일형 Framework와 다르다
- 핵심 라이브러리는 뷰 레이어만 초점을 맞춰 기존 프로젝트와 통합이 쉬움
- 정교한 단일 페이지 응용프로그램을 완벽하게 지원 가능

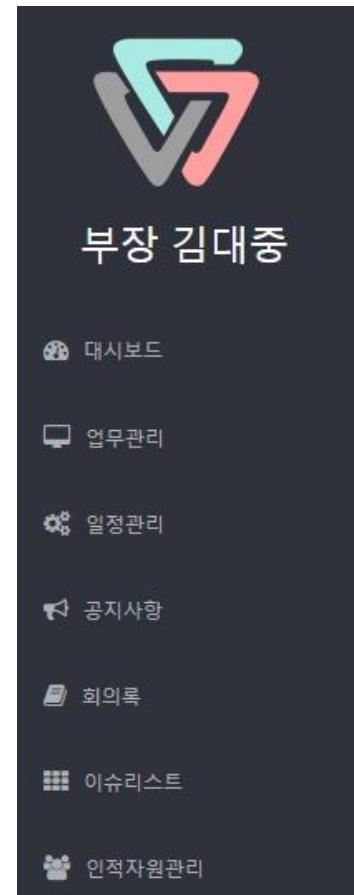


부장 김대중

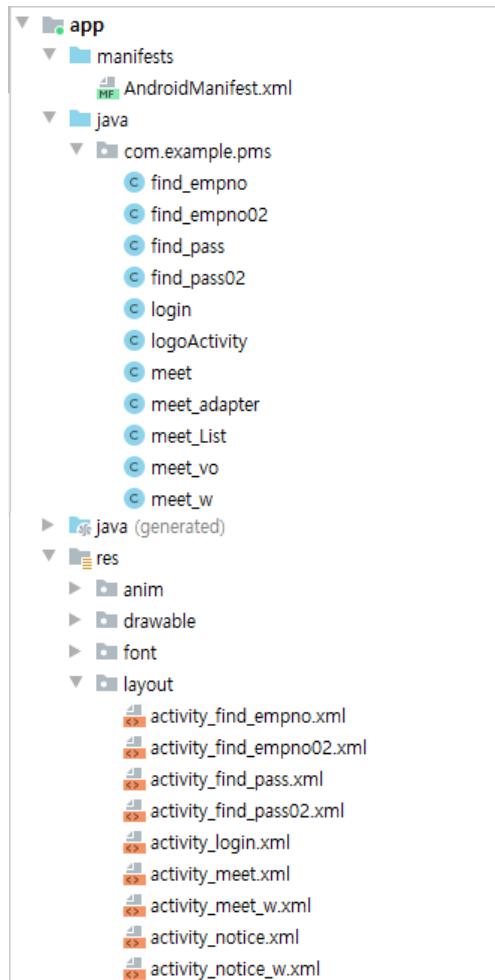
### ▼ 적용 부분

- 접근 권한에 따른 사이드 메뉴 표시 변경

```
v-if="mdiv==6 || mdiv==4 || mdiv==5"  
  
var vm = new Vue({  
  el: "#show",  
  data:{mdiv:${infor_M.mdiv}}}  
});
```



## Android studio



- Android 앱 개발을 위한 공식 통합 개발 환경(IDE)
- Java와 코틀린을 기반으로 안드로이드 애플리케이션을 제작한다

### 프로젝트 구조

Android 스튜디오의 각 프로젝트에는 소스 코드 파일 및 리소스 파일과 함께 하나 이상의 모듈이 포함됩니다. 모듈에는 다음 폴더가 포함됩니다.

**Manifests** : AndroidManifest.xml 파일이 포함됩니다.

**Java** : JUnit 테스트 코드를 비롯한 자바 소스 코드 파일을 포함합니다.

**Res** : 코드가 아닌 모든 리소스를 포함합니다(XML 레이아웃, UI 문자열 및 비트맵 이미지 등).

## Manifests

```
<uses-permission android:name="android.permission.INTERNET"/>
<application
    android:networkSecurityConfig="@xml/network_security_config"
```

안드로이드 어플리케이션을 구동하는데 필요한 설정값을 관리해주는 곳이다.

## Java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.risk_main);
```

자바언어를 이용해 안드로이드 프로그래밍을 한다

## Res

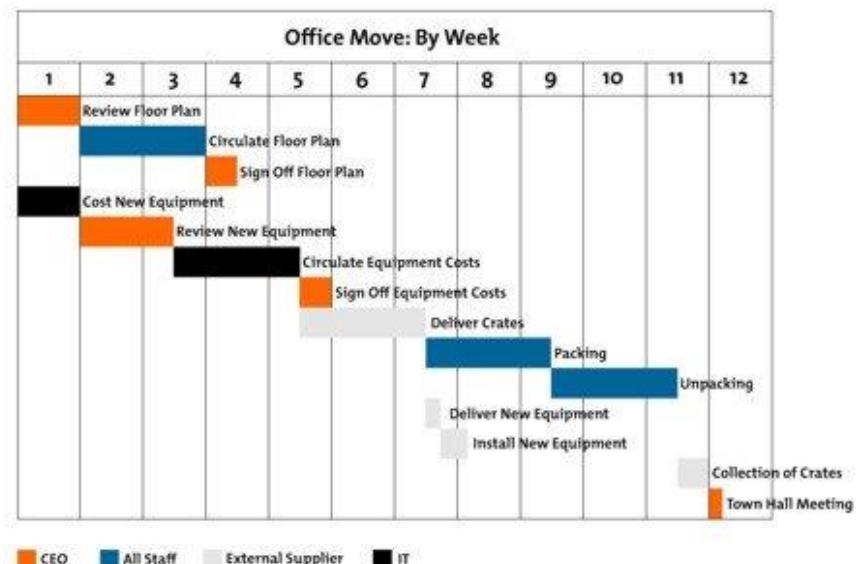
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
```

우리가 눈에 보이는 ui 를 구현하는 파일

### PMS와 Gantt Chart

#### 팀 프로젝트 계획 및 예약

- 프로젝트를 성공적으로 완료하려면 많은 업무(Task)를 제어하고 일정대로 완료되었는지 확인해야 한다. 마감일을 놓치거나 순서대로 작업을 완료하지 않으면 나머지 프로젝트에 영향을 줄 수 있고 결과적으로 프로젝트의 진행이 늦어질 수 있으며 이로 인해서 더 많은 비용이 들어간다. 따라서 수행해야 할 모든 업무를 보고 각 업무를 완료해야하는 시점을 한 눈에 알 수 있는 수단이 필요하다.
- 간트차트(Gantt Chart)는 이러한 정보를 시각적으로 전달하고 프로젝트와 관련된 모든 작업과 시간 순서에 따라 표시되는 정보를 간략하게 표시한다. 이를 통해 프로젝트의 관련 업무 및 완료 시기를 즉각적으로 확인할 수 있다.



## Highcharts

- Javascript를 이용해 웹의 통계적인 정보 시각화를 위해 사용되는 차트 라이브러리
- JSON 형식의 옵션을 사용하여 데이터 입력, 옵션 설정이 가능하고 다양한 차트 유형을 제공
- 장점 : 데이터 시각화 라이브러리인 D3.js 보다 사용하기 쉽다.

## ▼ 적용 부분

- 일정관리에서 PM과 팀원의 일정을 파악할 수 있는 Gantt Chart 사용

### >간트차트(PM)

PMS시스템 개발



## Highcharts Example

>간트차트(PM)

### PMS시스템 개발



## Highcharts - 데이터입력

- HighCharts에 데이터를 입력시키기 위해 DB에서 받아온 ArrayList 객체를 JSON 데이터로 변환시켜 Ajax를 이용해 불러올 필요가 있다.
- Spring에서 JSON처리 Viewer를 호출하고 컨트롤러에서 사용, 해당 데이터를 페이지로 불러온다.

### Controller

```
@RequestMapping(params="method=ajaxdata")
public String chtData(HttpServletRequest request,
    Model d) {
    HttpSession session = request.getSession();
    PmsMember emp =
        (PmsMember)session.getAttribute("infor_M");

    d.addAttribute("chartlist",
        cservice.chartList(emp.getPno()));
    return "pageJsonReport";
}
```

### dispatcher - servlet.xml

```
<!-- json데이터 처리 Viewer ArrayList ==> json 데이터 변환 처리 -->
<bean id="pageJsonReport"
    class="org.springframework.web.servlet.view.json.MappingJacksonJsonView">
    <property name="contentType" value="text/html;charset=utf-8"/>
</bean>
```

### jsp Ajax

```
$ajax({
    type: "post",
    url: "${path}/chart.do?method=ajaxdata",
    data: $("form").serialize(),
    dataType: "json",
    success: function(data){
        var clist = data.chartlist;
        var dataInfo = [];
        $.each(clist, function(idx, chart){
            dataInfo.push({
                name: chart.tname, id: ""+chart.tno+"",
                parent: ""+chart.refno+"",
                start: today+((chart.sdate+1) * day),
                end: today+((chart.edate+1) * day),
                completed : chart.prog,
                owner : ""+chart.name+"", y : idx+1,
                collapsed : true, description : chart.level,
            });
        ...
        Highcharts.ganttChart('chartpm', cInfo);
    }
});
```

## Highcharts – 차트 선언 및 기본 입력정보

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script src="https://code.highcharts.com/gantt/highcharts-gantt.js"></script>
<script src="https://code.highcharts.com/gantt/modules/exporting.js"></script>
```

```
var cInfo = {
    series : seriesInfo,
    tooltip : tooltipInfo,
    title: titleInfo,
    xAxis: xAxisInfo,
    yAxis: yAxisInfo,
};

dateFormat = Highcharts.dateFormat,
defined = Highcharts.defined,
isObject = HighchartsisObject,
reduce = Highcharts.reduce;

Highcharts.ganttChart('chartpm', cInfo);

<div id="chartpm"></div>
```

- Highcharts를 사용할 웹페이지에 script 태그로 위의 세 가지 js 추가한다.
- script단에서 아래와 같은 형태로 선언한다.  
Highcharts.사용할 chart종류(차트가 입력될 div, 차트 입력정보(JSON))
- 차트 입력정보에는 JSON데이터를 통째로 넣을 수 있다. 가독성을 위해서 입력정보를 분리해서 입력하는 것을 추천
- 차트 입력 정보, 하위 항목에도 JSON형식으로 데이터를 입력할 수 있다.
  - series : 차트 데이터
  - tooltip : 툴팁 데이터
  - title : 차트명 데이터
  - xAxis : X축 데이터
  - yAxis : Y축 데이터

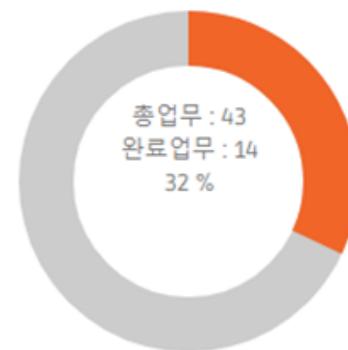
## EasyPieChart

- jQuery plugin
- 다른 plugin들과 달리 하나의 function을 실행하여 여러 개의 pie chart 구현 가능
- 장점 : <c:forEach>를 활용하여 구현해야 하는 pie chart의 개수가 확장성이 있을 때 활용하기 좋음

### ▼ 적용 부분

- CEO / CTO 대시보드에서 프로젝트 별 상세 내용을 표현하기 위해 사용

```
<div class="chart" data-percent="${ceo.cp }"  
style="float: left; width: 30%; height: 230px;  
text-align: center; margin-left: 50px;"><br>  
<span class="title">총업무 : ${ceo.total }</span>  
<span class="title">완료업무 : ${ceo.ccnt }</span>  
<span class="title">${ceo.cp } %</span></div>
```



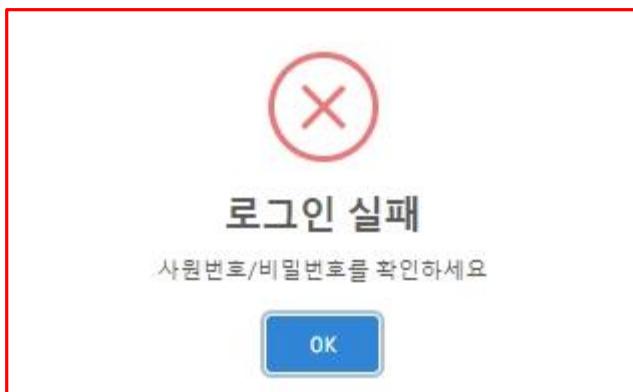
## Sweet Alert

- jQuery Plug-In
- 브라우저 IE 11+, Edge, Chrome, Firefox, Safari, Opera 적용 가능
- JS에서 기본으로 제공하는 Alert와 Confirm을 Model UI를 이용해 더 세련된 웹 페이지를 만들 수 있다.

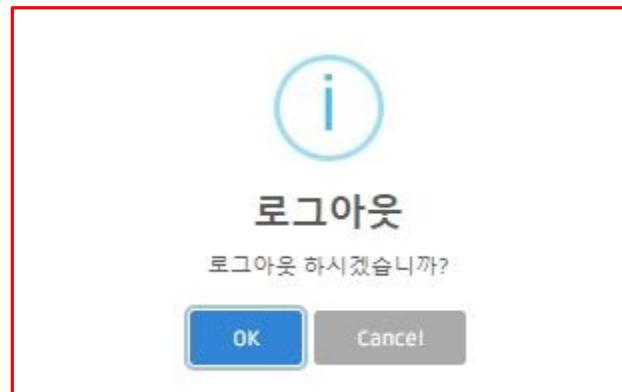
### ▼ 적용 부분

- Alert, Confirm 과 같은 경고 창을 띄워야 할 상황에 적용

Alert



Confirm



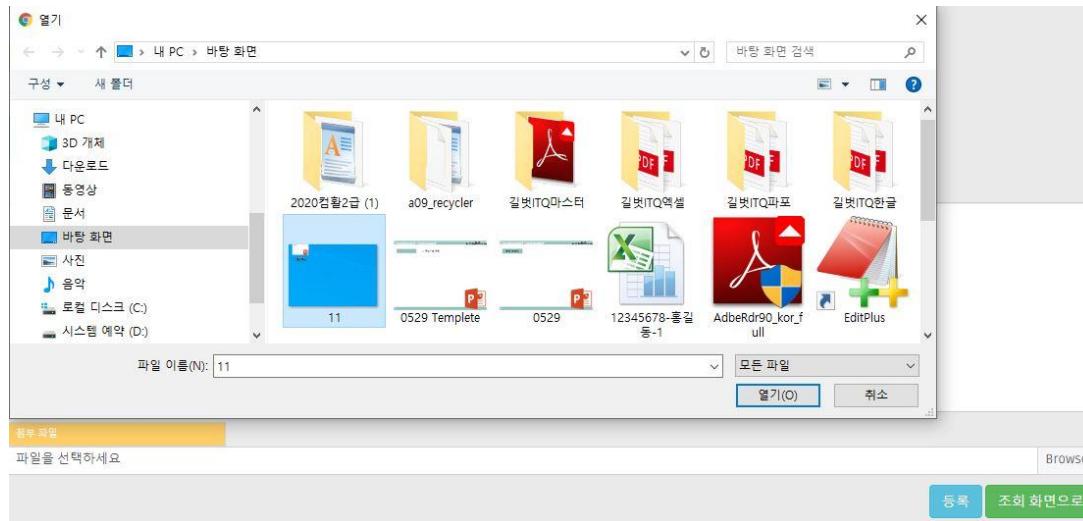
### 파일 업로드 처리

- 파일 업로드를 위해 쓰는 Multipart Resolver는 파일 업로드와 같은 멀티파트 포맷의 요청정보를 처리하는 핸들러이다.
- Multipart Resolver를 사용하기 위해서 Dispatcher Servlet에 아래와 같이 빈을 설정 해줘야 한다.

```
<bean id="multipartResolver"
      class="org.springframework.web.multipart.commons.CommonsMultipartResolver"/>
<!-- 다운로드 viewer -->
<bean id="download" class="project06.util.DownloadViewer" />
```

### 개발 순서

1. 화면 단에서 파일 처리를 위해 form enctype="multipart/form-data"로 설정
2. VO 단에서 Multi parte File 파일을 전송하는 객체 생성
3. Service 단에서 Multipart File을 파일 객체로 변환
4. Dao를 통해 파일 업로드 정보를 입력한다.



### view

```
<form method="post" enctype="multipart/form-data" id="fileForm">
<input type="file" name="report">
```

### dao

#### 파일 정보 입력처리

```
<insert id="insertRepo" parameterType="String">
    insert into PMSTFILE values(pmstfile_seq.nextval,pmstask_seq.currval,#{fname}, sysdate)
</insert>
```

```
public void insertRepo(String fname);
```

### SendMail

1. 스프링에서는 메일을 수신 발신 처리하는 모듈을 포함하고 있다
2. 개발 순서
  - 1) 프레임워크 확인 및 등록.(lib)  
javaMailSenderImpl
  - 2) 메일 서버에 host 사용 등록 처리.
  - 3) 컨테이너에서 bean으로 등록  
SMTP(발송 서버에 등록) : HOST(gmail), PORT, 계정(id, pass)

```
<Dispatcher-servlet.xml>
<!-- SMTP 서버 설정 -->
<bean id="mailSender"
      class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host" value="smtp.gmail.com"/>
    <property name="port" value="587"/>
    <!-- gmail에 등록한 계정과 비밀번호 입력 -->
    <property name="username" value="" />
    <property name="password" value="" />
    <!-- 보안 관련 설정. -->
    <property name="javaMailProperties">
      <props>
        <prop key="mail.smtp.auth">true</prop>
        <prop key="mail.smtp.starttls.enable">true</prop>
      </props>
    </property>
  </bean>
```

FIND PASSWORD

사원번호  
사원번호를 입력하세요

이름  
이름을 입력하세요

이메일  
이메일을 입력하세요

인증번호  
전송

FIND

### chatting

#### - web socket 통신을 통한 채팅 처리

: 기본적으로 웹 서버 기반으로 socket 통신을 이용하여,  
여러가지 채팅 클라이언트의 요청값으로 실시간 처리하는 것을 말한다.

#### - 통신 방식

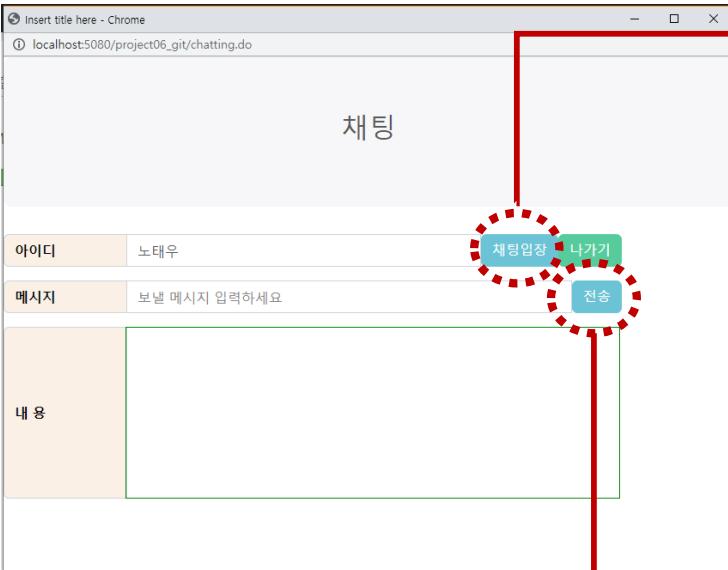
- 1) 한명의 클라이언트가 메시지를 소켓통신을 통해 서버에 전달하면
- 2) 서버는 push방식으로 접속한 클라이언트에게 메시지를 전달한다.
- 3) js의 socket통신과 java의 handler와 연동처리
- 4) 프로세스

##### **front end(js,html)**

서버에 WEB SOCKET통신으로 접속  
지원되는 메서드를 통해서 서버에 접속,  
메시지 전송, 서버와 접속종료 처리

##### **back end(java)**

handler를 통해서 client에서 오는 정보를 가지고,  
접속, 메시지 처리, 종료, 예외처리 등을 한다.



## 웹 소켓을 통한 서버와의 연결 처리

### jsp

```
if(confirm("채팅창 접속합니다!")){
    wsocket = new WebSocket("ws://192.168.4.112:5080/${path}/chat-ws.do");
    wsocket.onopen=function(evt){
        console.log(evt);
        wsocket.send("msg:"+$("#id").val()+" :연결 접속했습니다.");
    }
}
```

### Handler

```
@Override
public void afterConnectionEstablished(WebSocketSession session) throws Exception {
    users.put(session.getId(), session);
}
```

## 메시지를 주고 받을 때 처리 내용

### jsp

```
wsocket.onmessage = function(evt){
    var data = evt.data;
    if(data.substring(0,4)=="msg:"){
        revMsg=data.substring(4);
        recieveMsg(revMsg);
    }
};
```

### Handler

```
protected void handleTextMessage(WebSocketSession session, TextMessage message) throws Exception {
    for(WebSocketSession ws : users.values() ){
        // 1. 각 client에게 메시지 전달.
        ws.sendMessage(message);
    }
}
```

## Business logic

1. Admin과 프로젝트에 참여중인, PM, 팀원만 사용 가능
2. 채팅입장 버튼클릭 시 서버에 WEB SOCKET통신으로 접속이 가능
3. 메시지 전송 시 접속중인 인원에게 내용이 보여진다.
4. 나가기 버튼으로 접속 종료



# *Function Flow*

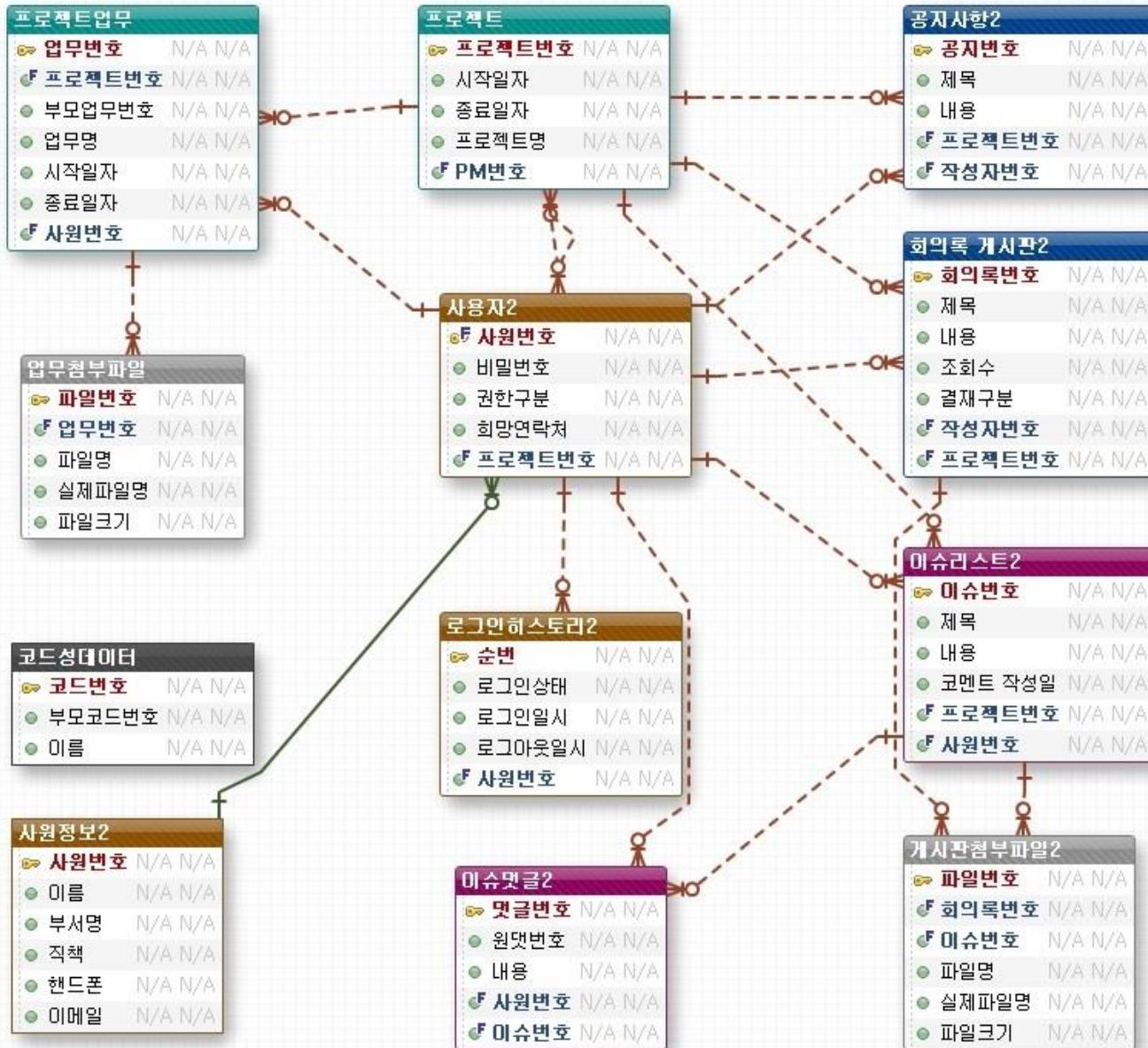
# Function Flow



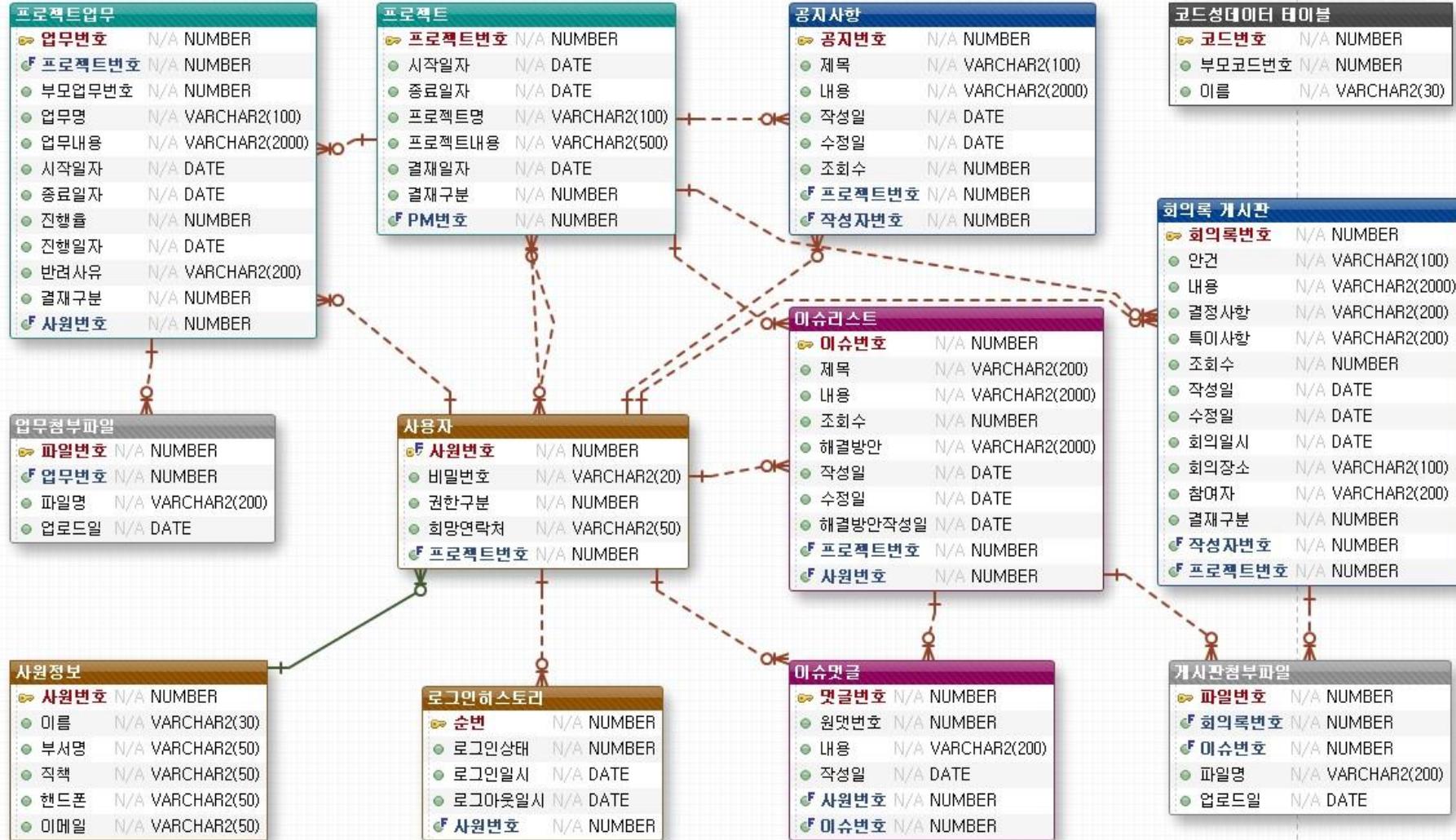


*DB Construction*

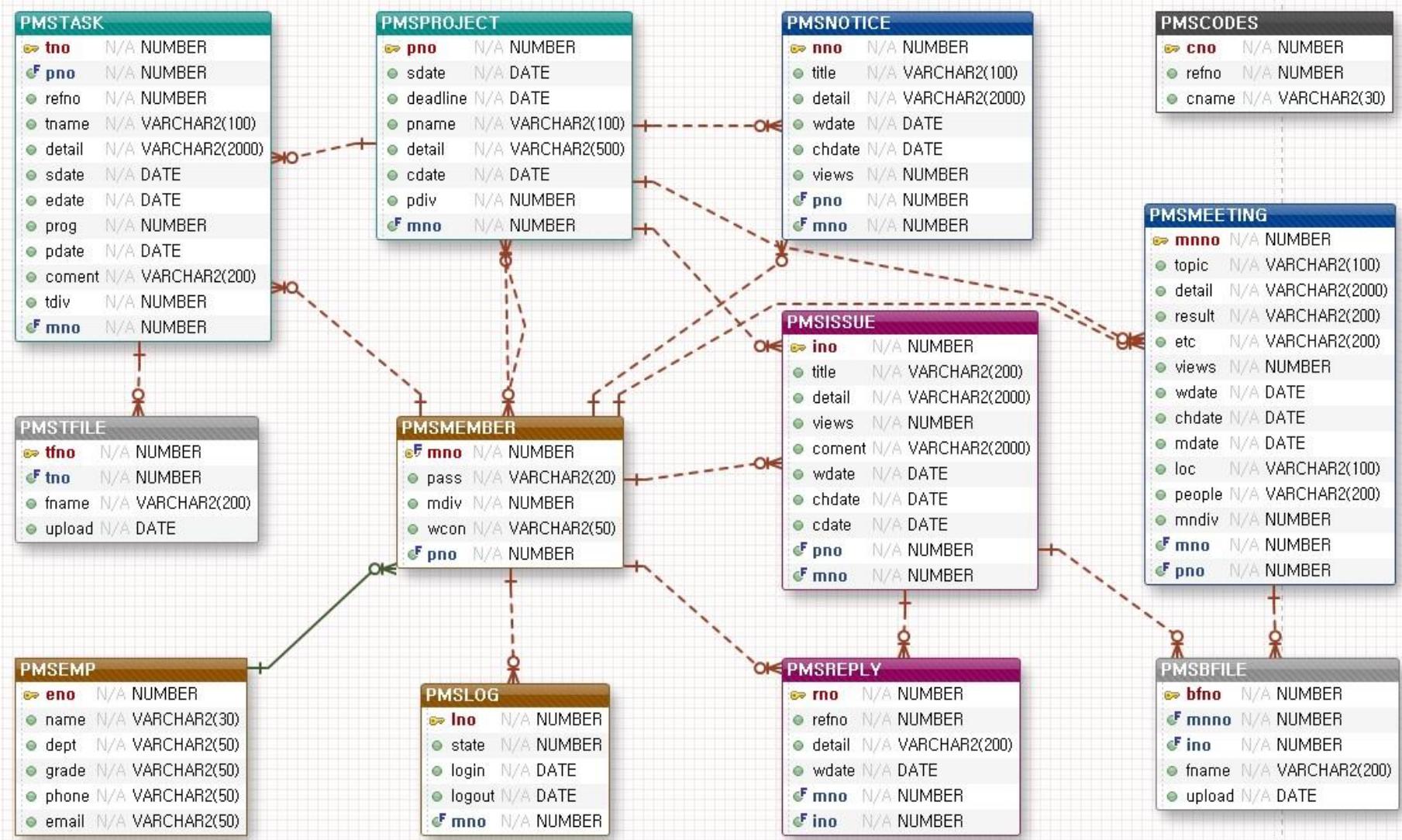
# ERD 개요



# ERD 논리 설계



# ERD 를 리 설계





***Screen Design***



**화면설계**

**WEB**

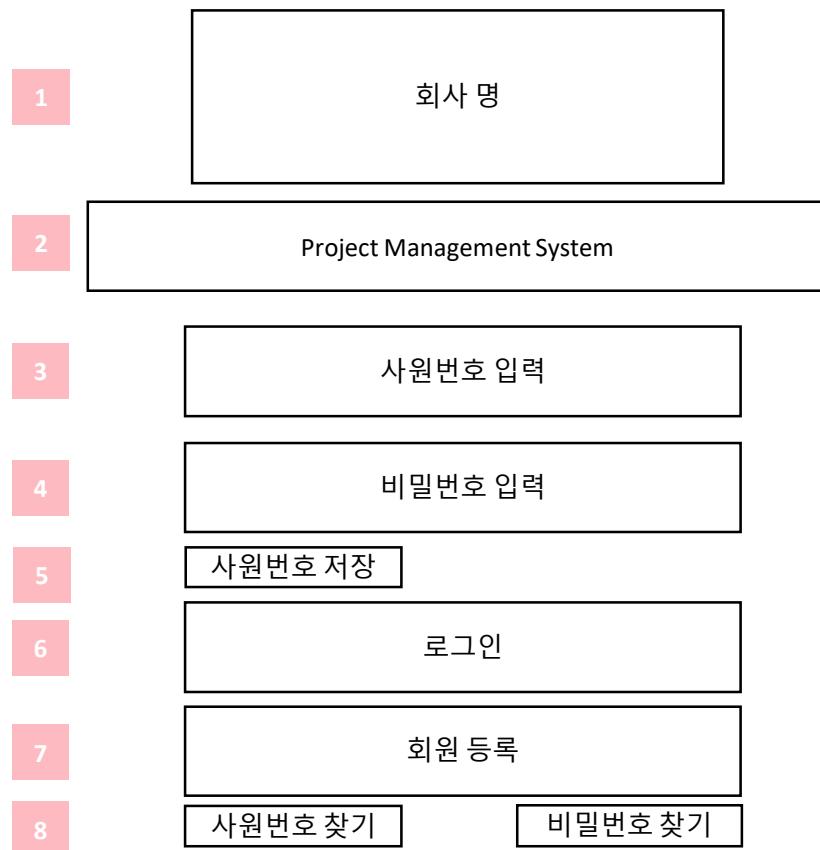


화면설계

# 사원정보

김진수

- 사원번호-비밀번호 로그인 & 사원번호-비밀번호 찾기 & 회원등록



- 1 회사명  
- 텍스트 내용
- 2 P.M.S  
- 텍스트 내용
- 3 사원번호 입력  
- 텍스트 입력 받기
- 4 비밀번호 입력  
- 텍스트 입력 받기(정규식)
- 5 사원번호 저장  
- 체크박스  
- 쿠키를 이용해 사원번호 저장 및 가져오기
- 6 로그인  
- submit 가능 버튼
- 7 회원 등록  
- 페이지 이동
- 8 사원번호/비밀번호 찾기  
- 페이지 이동

- 사원정보의 이름 & 이메일로 사원번호 찾기

1 사원번호 찾기

2 이름  
이름 입력

3 이메일  
이메일 입력

4 확인

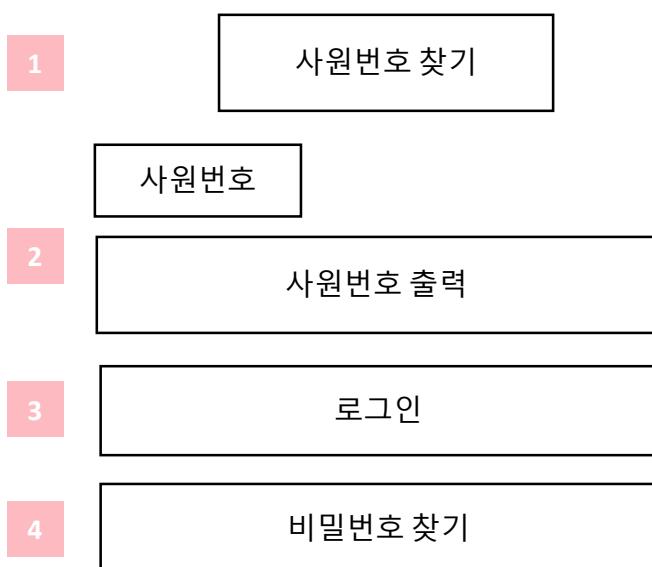
1 사원번호 찾기  
- 텍스트 내용

2 이름  
- 텍스트 내용  
이름 입력  
- 텍스트 입력 받기

3 이메일  
- 텍스트 내용  
이메일 입력  
- 텍스트 입력 받기

4 확인  
- 이름, 이메일 일치하는  
사원번호 출력

- 이름 & 이메일이 일치하는 사원번호 출력



1 사원번호 찾기  
- 텍스트 내용

2 사원번호  
- 텍스트 내용  
사원번호 출력  
- 데이터 Select  
사원번호 출력

3 로그인  
- 로그인(메인화면) 이동

4 비밀번호 찾기  
- 페이지 이동

- 사원번호 & 이름 & 이메일이 일치하면 비밀번호 재설정

1 비밀번호 찾기

2 사원번호

3 이름

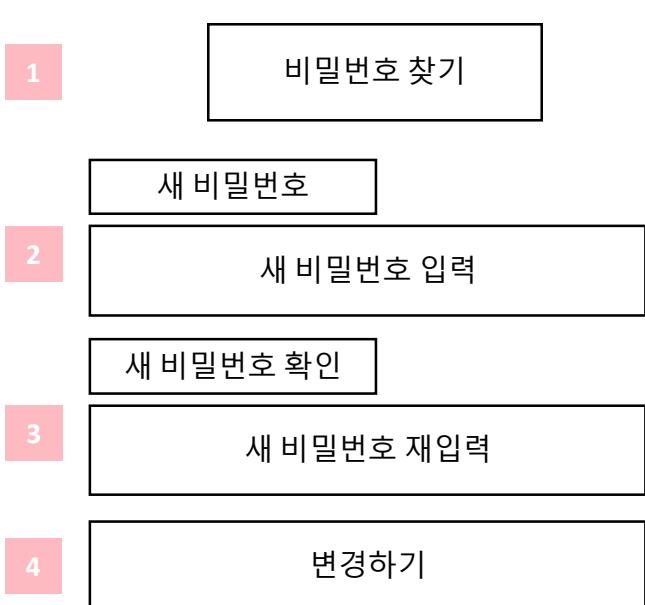
4 이메일

5 인증번호 입력

6 변경하기

- 1 비밀번호 찾기
  - 텍스트 내용
- 2 사원번호
  - 텍스트 내용
  - 사원번호 입력
  - 텍스트 입력 받기
- 3 이름
  - 텍스트 내용
  - 이름 입력
  - 텍스트 입력 받기
- 4 이메일
  - 텍스트 내용
  - 이메일 입력
  - 텍스트 입력 받기
- 5 인증번호 입력
  - 텍스트 입력 받기
  - 전송/인증
  - 인증번호 전송/인증 버튼
- 6 변경하기
  - 비밀번호 변경 페이지 이동

● 비밀번호 변경하기



- 1 비밀번호 찾기
  - 텍스트 내용
- 2 새 비밀번호
  - 텍스트 내용
  - 새 비밀번호 입력
  - 텍스트 입력 받기(정규식)
- 3 새 비밀번호 확인
  - 텍스트 내용
  - 새 비밀번호 재입력
  - 텍스트 입력 받기(정규식)
- 4 변경하기
  - 비밀번호 일치 여부 확인
  - 데이터 Update 수정
  - 로그인 페이지 이동

- 사원번호 & 이름으로 회원 등록

- 1 회원 등록
- 2 사원번호
- 3 사원번호 입력
- 4 이름
- 5 이름 입력
- 6 회원 등록

- 1 회원 등록
  - 텍스트 내용
- 2 사원번호
  - 텍스트 내용
  - 사원번호 입력
  - 텍스트 입력 받기
- 3 이름
  - 텍스트 내용
  - 이름 입력
  - 텍스트 입력 받기
- 4 회원 등록
  - 사원번호, 이름이 일치하면
  - 회원 등록 페이지로 이동

- 회원 등록 정보 입력

1 회원 등록

2 사원번호

3 사원번호 출력

4 비밀번호

5 비밀번호 입력

6 비밀번호 재입력

7 일치 확인

8 희망 연락처

9 희망 연락처 입력

10 핸드폰

11 이메일

12 회원 등록

- 1 회원 등록  
- 텍스트 내용
- 2 사원번호  
- 텍스트 내용  
사원번호 출력  
- 입력된 사원번호 출력
- 3 비밀번호  
- 텍스트 내용  
비밀번호 입력  
- 텍스트 입력 받기(정규식)
- 4 희망연락처  
- 텍스트 내용  
희망연락처  
- 체크 : 사원정보 핸드폰 번호  
사원정보 이메일  
- 다른 번호/메일 입력 시 저장
- 5 회원 등록  
- 데이터 insert

- 프로젝트명 & 개인정보 & 로그아웃 & 참여인원

1

2

프로젝트 명

3

개인정보/로그아웃

4

참여인원 @@명

5

각 화면 내용

1

상단 메뉴

- Include 사용

2

프로젝트 명

- 참여하고 있는 프로젝트 명  
출력(팀원/팀장)

3

개인정보

- 개인정보 페이지 이동
- 사원번호, 비밀번호, 이름,  
부서명, 직책, 희망연락처
- 로그아웃
- 로그인 세션 정보 제거
- 로그인 화면으로 이동

4

참여 인원

- 프로젝트에 참여한 인원 중  
로그인 된 인원 수
- 클릭하면 참여 인원의  
정보 출력(팝업)  
(이름, 직책, 희망연락처)

5

각 화면 내용

- 각 화면 내용 출력

- 개인 정보 확인 팝업 화면(비밀번호 입력)



1 개인 정보 팝업 화면  
- 팝업 화면

2 개인정보  
- 텍스트 내용

3 사원번호  
- 텍스트 내용  
사원번호 출력  
- 로그인 된 사원번호 출력

4 비밀번호  
- 텍스트 내용  
비밀번호 입력  
- 텍스트 입력 받기(정규식)

5 정보 확인  
- 비밀번호 일치 여부 확인  
- 정보 확인/수정 팝업

● 개인정보 확인 및 수정

개인 정보	
1 사원번호	2 전화번호
3 사원번호 출력	4 전화번호 출력
5 이름	6 이메일
7 이름 출력	8 이메일 출력
9 직책	10 비밀번호
11 직책 출력	12 비밀번호 * 출력
13 부서	14 비밀번호 재입력
15 부서 출력	16 일치 확인
17 희망 연락처	
18 희망 연락처 출력/수정	
19 닫기	
20 정보 수정	

- 1 개인정보 팝업 화면
  - 정보 확인/수정 화면
- 2 정보 출력
  - 사원번호/이름/직책/부서
- 3 정보 출력
  - 전화번호/이메일
- 4 비밀번호
  - 비밀번호 \* 출력
  - 일치 확인
  - 비밀번호 일치 확인(정규식)
- 5 희망 연락처
  - 입력된 정보 출력
  - 희망 연락처 수정
  - 수정은 새로운 내용 입력
- 6 닫기
  - 팝업 화면 닫기
- 7 정보 수정
  - 비밀번호/희망연락처 수정

- 참여인원의 기본정보 및 로그인 여부

1

2 이름	직책	연락처	로그인 상태

3

이름 입력

검색

4			

5

페이지 처리

## 1 참여 인원 팝업 화면

- 팝업 창 출력

2

표 컬럼명

- 텍스트 내용

3

이름 입력

- 텍스트 입력 받기

검색

- 일치하는 정보 출력

4

리스트

- 프로젝트 참여 인원  
리스트 출력

5

페이지 처리

- 리스트에 5~6줄 출력  
페이지 처리



화면설계

# 프로젝트 의사소통관리

김하나

● 공지사항 게시판 리스트

1

2

no	제목	작성일	조회수

3

4

5 글쓰기

1 화면 사이즈

- 한 화면에 보이는 글의 개수를 조정
- 3, 5, 10, 20, 30 사이즈 정도

2 검색창

- 제목으로 내용을 검색

3 공지사항 리스트

- 텍스트로 출력

4 pagination

- 글의 개수에 따라 페이지 증가

5 글쓰기 버튼

CEO/ 팀장 → 有  
팀원 → 無

- 공지사항 게시판 상세페이지

1

제목

작성자

작성일

조회수

내용

2

수정

삭제

1 상세페이지

2 수정/삭제 버튼

CEO/ 팀장 → 有  
팀원 → 無

- 공지사항 게시판 작성 페이지 - CEO/ 팀장 만 접근 가능

1

제목

내용을 입력해주세요

2

글쓰기

1

상세페이지

2

글쓰기 버튼

● 회의록 게시판 리스트

1

2

no	제목	작성일	조회수

3

4

5 글쓰기

1 화면 사이즈

- 한 화면에 보이는 글의 개수를 조정
- 3, 5, 10, 20, 30 사이즈 정도

2 검색창

- 제목으로 내용을 검색

3 회의록 리스트

- 텍스트로 출력

4 pagination

- 글의 개수에 따라 페이지 증가

5 글쓰기 버튼

CEO/ 팀장 → 有  
팀원 → 無

● 회의록 상세페이지

1

제목

작성자

작성일

조회수

내용

2

첨부파일 다운로드

3

수정

1 상세페이지

2 첨부파일 다운로드

- 파일을 클릭하면 다운로드 처리

3 수정 버튼

팀장 /팀원 → 有

CEO → 無

● 회의록 게시판 작성 페이지 – 팀장/팀원만 접근 가능

1

제목

내용을 입력해주세요

2

파일탐색

등록

3

글쓰기

1 상세페이지

2 글쓰기 버튼

3 파일탐색/등록 버튼

팀장 /팀원 → 有

CEO→ 無



화면설계

# 프로젝트 리스크관리

이준석

## 상단 메뉴

## 이슈리스트

번호	제목	작성자	작성일	조회수
4 3				
3				
2				
1				

4 < 1 2 3 > 5 글쓰기

## 사이드 메뉴

- 1 페이지 크기 설정
  - 초기값 5
  - 10, 20 으로 변경 가능
- 2 프로젝트 번호 설정
  - 초기값 1001
  - 현재 진행중인 프로젝트 번호로 선택 가능
  - Admin만 볼 수 있다.
- 3 상세페이지 이동
  - 클릭 시 해당하는 리스트의 상세페이지로 이동
- 4 블록 설정
  - 보여줄 블록 크기 5
  - 리스트의 양에 따라 1~5, 6~10 ... 5씩 증가해서 보여준다.
- 5 글쓰기
  - 팀원, PM, Admin만 작성 가능
  - 클릭 시 작성 페이지로 이동
- 6 검색 창
  - 제목, 작성자로 검색 가능

## 상단 메뉴

제목	
내용	
첨부 파일	<b>1</b>

첨부

등록

조회화면으로

1

## 파일 첨부 기능

- 클릭 시 팝업창이 뜨며 파일을 선택하면 파일 이름이 출력되며 등록 시 저장된다.

## 상단 메뉴

작성자		작성일	
제목		수정일	
내용			
<b>1</b> 첨부			
해결방안 작성일			
<b>2</b> 해결방안			

3 수정
4 해결방안 저장
5 뒤로가기

**4** 입력 댓글작성

**5** 댓글1 작성자 | 작성일 | 삭제 수정 댓글

**6** 댓글1의 댓글 작성자 | 작성일 | 삭제 수정 댓글

댓글2 작성자 | 작성일 | 삭제 수정 댓글

- 1** 파일 첨부 확인
  - 작성 페이지에서 저장한 파일을 확인하고 다운로드, 수정 할 수 있다.
- 2** 해결방안 작성
  - 팀장, Admin만 작성 가능
- 3** 수정
  - 작성자, Admin 만 수정 가능
- 4** 댓글 입력
  - 댓글을 입력하는 입력란
- 5** 댓글
  - 입력된 댓글의 작성자와 내용 등을 확인 할 수 있다.
  - 수정, 삭제를 할 수 있고 대댓글을 달 수 있다.
- 6** 대댓글
  - 선택한 댓글의 바로 아래에 생기며 내용의 왼쪽에 여백으로 댓글과 구별할 수 있다.



화면설계

# 프로젝트 일정관리

전현규

### 프로젝트 추가

프로젝트명

시작일   **1**

종료일

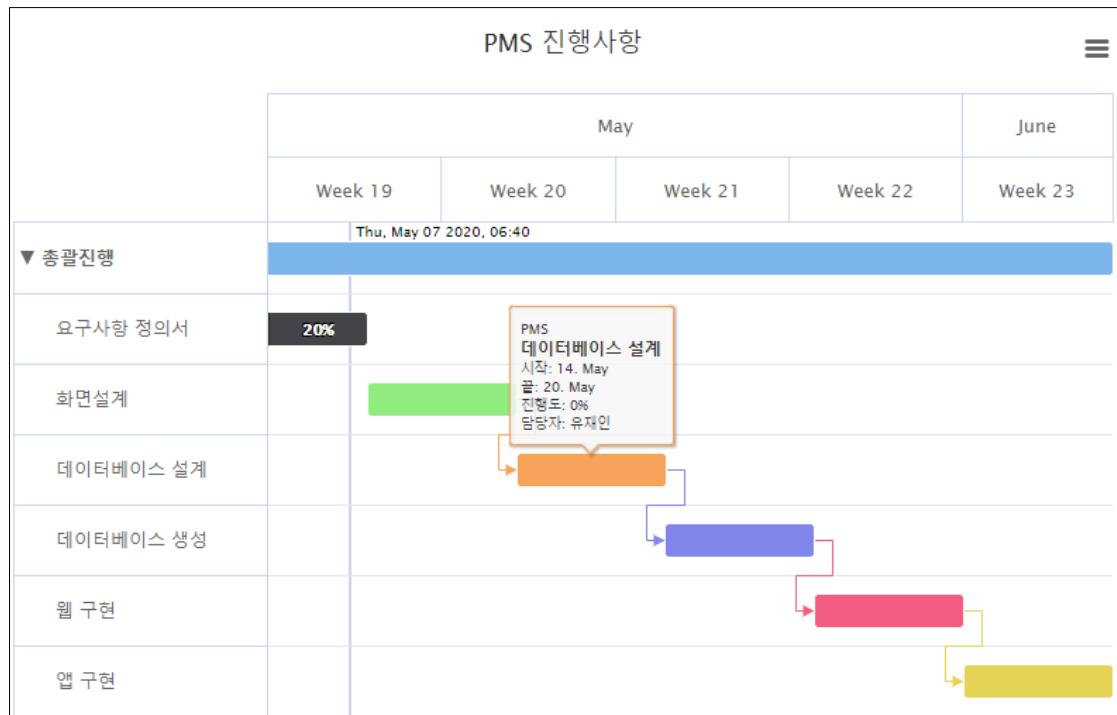
PM **2**

프로젝트 개요

**3**

- 1** 클릭시 캘린더 형식으로 선택
- 2** 현재 팀에 참여하지 않은 인원만 선택
- 3** 검증절차를 거쳐서 완료되면 등록
  - 프로젝트명, 시작일, 종료일, PM, 프로젝트개요가 입력되었는가?
  - 시작일과 종료일은 오늘 날짜 이후로 설정되었는가?
  - 종료일이 시작일 이후로 설정되었는가?

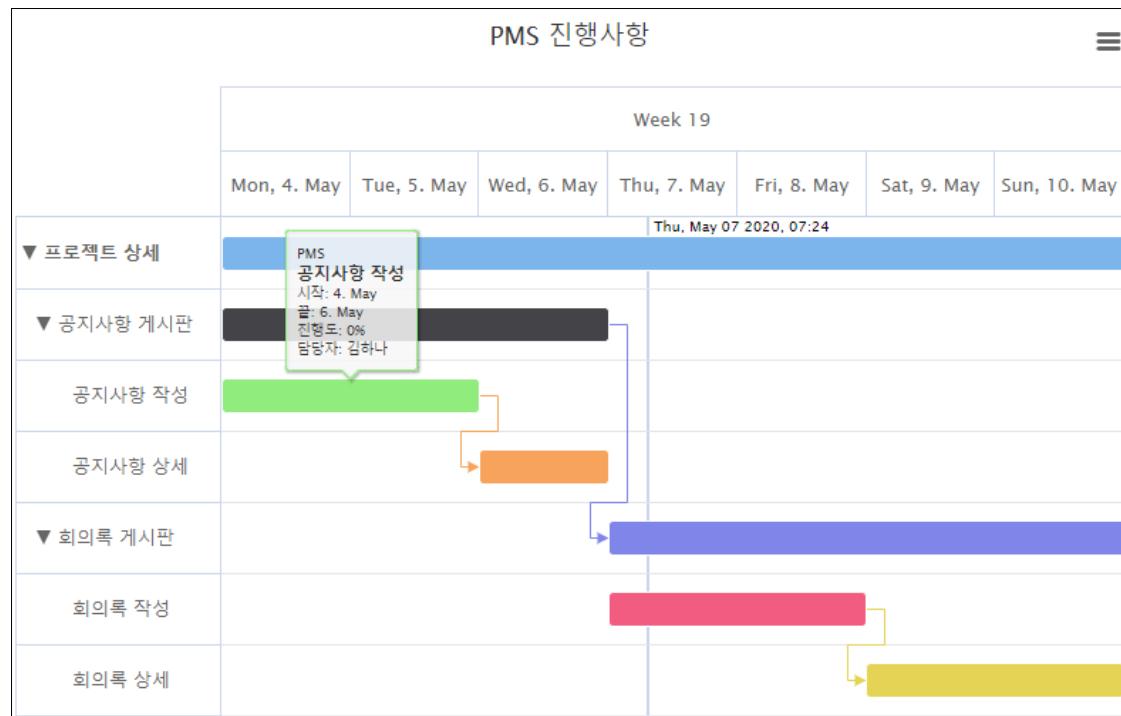
1



1

본인이 담당하고 있는 프로젝트의 전체 업무와 일정이 표시된 간트차트 출력

1



1

본인이 소속되어 있는 프로젝트에서 담당하고 있는 업무와 해당 업무가 포함된 상위 업무, 각 업무이 대한 일정이 표시된 간트차트 출력



화면설계

# 프로젝트 통합관리

이연학

● 업무 리스트

5

### 상단메뉴

1      검색

번호	업무 명	2 종료일	3 결재 여부

6      4      업무등록

1      2      3

- 1 업무명 검색  
- 업무 명 입력해서 검색
- 2 업무 종료일 표시
- 3 결재 여부 표시  
- 업무 중 / 결재 / 반려처리
- 4 업무 등록  
- PM / 관리자 권한
- 5 메뉴  
- 리스트 이동 / 채팅 / 개인정보
- 6 페이징 처리  
- 업무 개수에 따른 페이징 블록 생성

● 업무 상세

글번호	
담당자	
업무명	

시작일 ~ 마감일
-----------

업무 내용
-------

1 업무 진행도	저장
----------	----

2 결재여부		신청	반려	결재
--------	--	----	----	----

3 진행률		수정
-------	--	----

4 반려사유	등록
--------	----

5 수정	삭제	목록
------	----	----

**1 업무 진행도 저장**

- 업무 진행도 기준에 따라 담당자가 직접 수정

**2 결재 여부 처리**

- 진행률이 100이 되면 담당자가 결재 신청, 신청 후 PM이 반려, 결재완료 처리

**3 진행률 수정**

- 업무 담당자가 업무 기준에 따라 진행률 수정

**4 반려 사유 등록**

- 반려 처리시에만 PM이 반려 사유 등록

**5 업무 수정 삭제**

- 해당 업무 업무명 / 담당자 / 업무 내용 수정이나 업무삭제

● 업무 작성

업무명

상위업무번호

1

시작일

마감일

담당자

업무 상세

2

첨부 파일

선택

등록

목록

1

시작일 마감일 설정

- Datepicker 이용한 날짜 선택

2

첨부 파일

- 파일 선택 후 등록



화면설계

# 프로젝트 총괄/통합관리

김형준

- CEO, CTO, HR, PM 권한 설정 직원 리스트

사원번호	이름	직책	부서명	이메일	핸드폰	권한
2						

- 1 직원 리스트
  - 등록된 데이터 받기
- 2 더블클릭 시 상세 이동
  - 페이지 이동
- 3 페이징 처리
  - 한 페이지에 볼 데이터
- 4 검색
  - select box(구분)
  - input(텍스트입력)
  - submit 기능 버튼

- CEO, CTO, HR, PM 권한 설정 상세 페이지

1

&gt; 권한설정

이름

부서명

직책

핸드폰

이메일

권한

프로젝트

3

프로젝트번호 선택

2

CEO 설정

CTO 설정

HR 설정

PM 설정

4

이전

1

직원 상세정보

- 등록된 데이터 받기

2

권한 설정버튼

- submit기능 버튼

3

PM설정 시 프로젝트 할당

- select box

- DB에서 데이터 불러오기

4

이전 페이지 이동

- 페이지 이동

## ● PM 팀원 추가

**1**

&gt;사원목록

구분	<input type="text"/>	Search
----	----------------------	--------

사원번호	이름	직책	부서명	이메일	핸드폰	권한	선택
------	----	----	-----	-----	-----	----	----

**2**

<input type="radio"/>
<input type="radio"/>
<input type="radio"/>
<input type="radio"/>

<	1	2	3	>
---	---	---	---	---

**4**

팀원삭제이동

**3**

추가

**1**

직원 리스트

- 등록된 데이터 받기

**2**

직원 선택

- radio 사원 선택

**3**

선택 직원 팀원 추가 버튼

- submit 기능 버튼

**4**

팀원 삭제 페이지 이동

- 페이지 이동

## ● PM 팀원 삭제

**1**

&gt;사원목록

구분	<input type="text"/>	Search
----	----------------------	--------

사원번호	이름	직책	부서명	이메일	핸드폰	권한	선택
------	----	----	-----	-----	-----	----	----

**2**

<input type="radio"/>
<input type="radio"/>
<input type="radio"/>
<input type="radio"/>

<	1	2	3	>
---	---	---	---	---

**4**

팀원추가이동

**3**

삭제

**1**

직원 리스트

- 등록된 데이터 받기

**2**

팀원 선택

- radio 팀원 선택

**3**

선택 직원 팀원 삭제 버튼

- submit 기능 버튼

**4**

팀원 삭제 페이지 이동

- 페이지 이동

- 인사관리자 사원등록/수정/퇴사 처리 리스트

1

&gt;사원목록

구분		Search
----	--	--------

사원번호

이름

직책

부서명

이메일

핸드폰

2

&lt; | 1 | 2 | 3 | &gt;

3

사원등록

1

직원 리스트

- 등록된 데이터 받기

2

더블클릭 시 상세 이동

- 페이지 이동

3

사원등록 페이지 이동

- 페이지 이동

- 인사관리자 사원 등록

1

&gt; 사원정보 입력

이름

부서명

직책

핸드폰

이메일

2

등록

3

취소

1

사원정보 입력

- input 텍스트 입력

2

등록 버튼

- submit 기능 버튼

3

이전 페이지 이동

- 페이지 이동

- 인사관리자 사원정보 수정 및 퇴사처리

1

&gt;사원정보 입력

이름

5

부서선택

부서명

6

직책선택

직책

핸드폰

이메일

2

퇴사

3

수정

4

이전

1

사원정보

- 등록된 데이터 받기

2

퇴사 처리 버튼

- submit 기능 버튼

3

수정 처리 버튼

- submit 기능 버튼

4

이전 페이지 이동

- 페이지 이동

5

수정 할 부서 선택

- select box

- DB에서 데이터 불러오기

6

수정 할 직책 선택

- select box

- DB에서 데이터 불러오기



화면설계

## 프로젝트 총괄/통합관리

유재인

1 Project 명

Project 시작일  
2

PM 직책 이름

종료일

프로젝트 상세 표시 위한  
각종 차트 영역

Project 명

PM 직책 이름

Project 시작일

종료일

프로젝트 상세 표시 위한  
각종 차트 영역

1 프로젝트 게이지바  
- 진행 정도에 따른 게이지 변경

2 각종 차트 영역  
- 프로젝트의 상세 내용을 한 페이지  
에 확인할 수 있도록 각종 차트를 구  
현

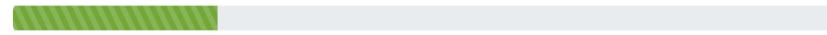
## 1 Project 전체 진행률



Project 시작일

종료일

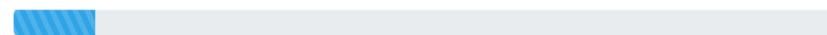
## 2 팀원 명



팀원 명



팀원 명



## 1 프로젝트 게이지바

- 프로젝트의 진행 정도 표시

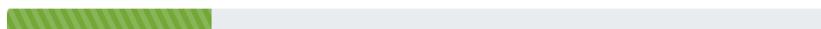
## 2 팀원 별 게이지바

- 각 팀원 별 진행 정도 표시

**1** Project 전체 진행률

Project 시작일

종료일

**2** 업무 명

시작일

종료일

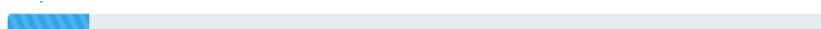
## 업무 명



시작일

종료일

## 업무 명



시작일

종료일

**1** 프로젝트 게이지바

- 프로젝트의 진행 정도 표시

**2** 업무 별 게이지바

- 각 업무 별 진행 정도 표시



**화면설계**

**APP**

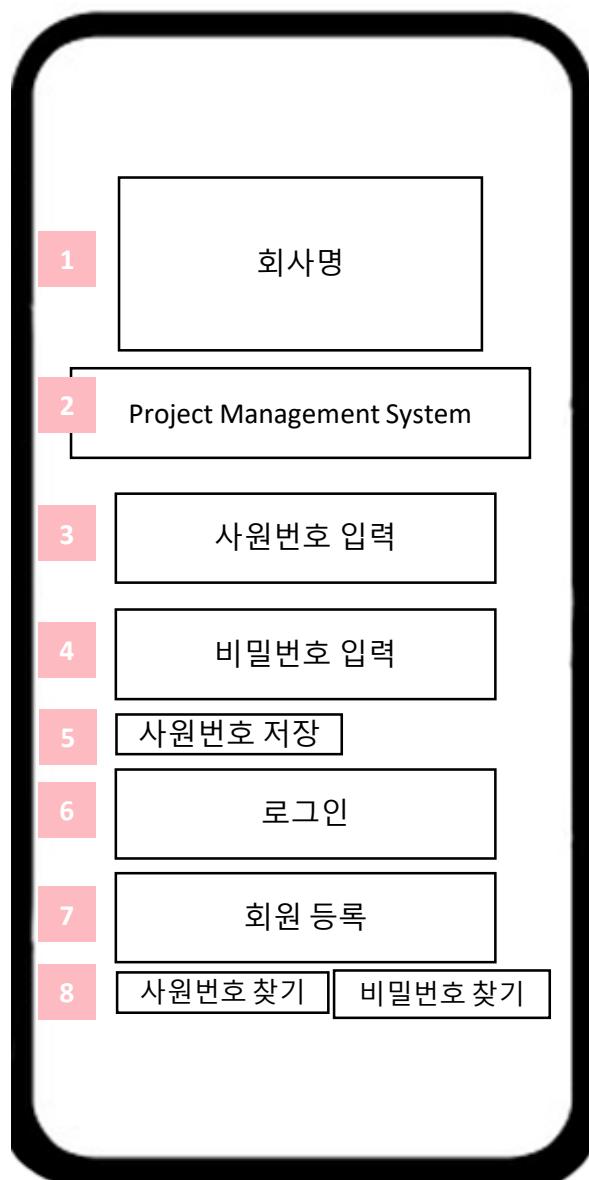


화면설계

# 사원정보

김진수

- 로그인 & 메인화면



- 1 회사명  
- 텍스트 내용
- 2 P.M.S  
- 텍스트 내용
- 3 사원번호 입력  
- 텍스트 입력 받기
- 4 비밀번호 입력  
- 텍스트 입력 받기
- 5 사원번호 저장  
- 체크박스
- 6 로그인  
- 페이지 이동 버튼
- 7 회원 등록  
- 페이지 이동
- 8 사원번호/비밀번호 찾기  
- 페이지 이동

- 사원정보의 이름 & 이메일로 사원번호 찾기



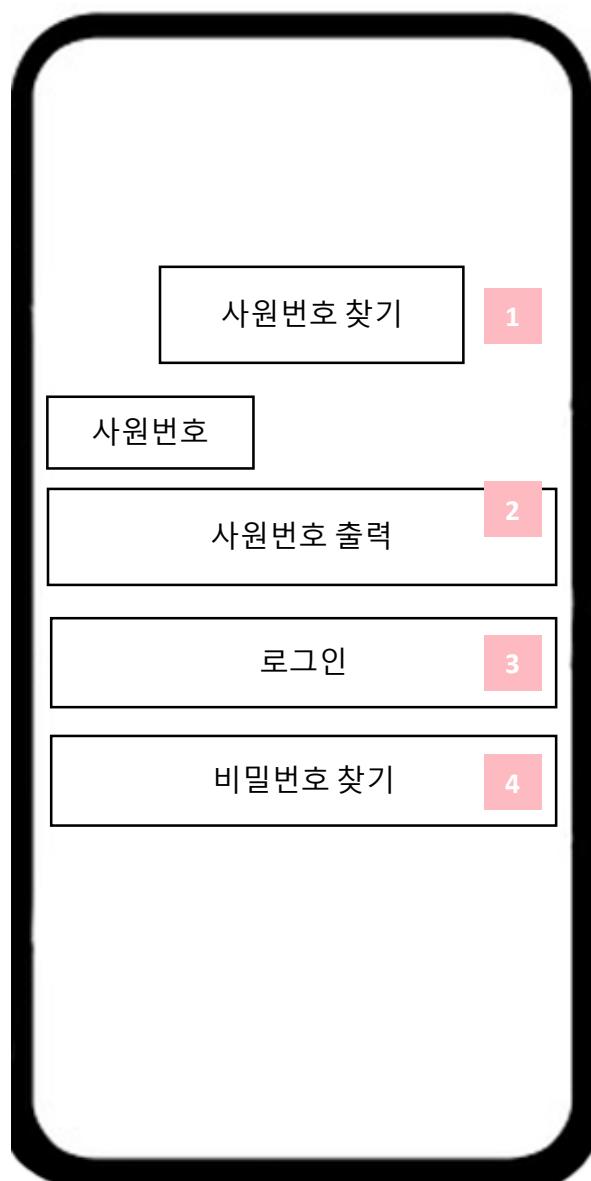
1 사원번호 찾기  
- 텍스트 내용

2 이름  
- 텍스트 내용  
이름 입력  
- 텍스트 입력 받기

3 이메일  
- 텍스트 내용  
이메일 입력  
- 텍스트 입력 받기

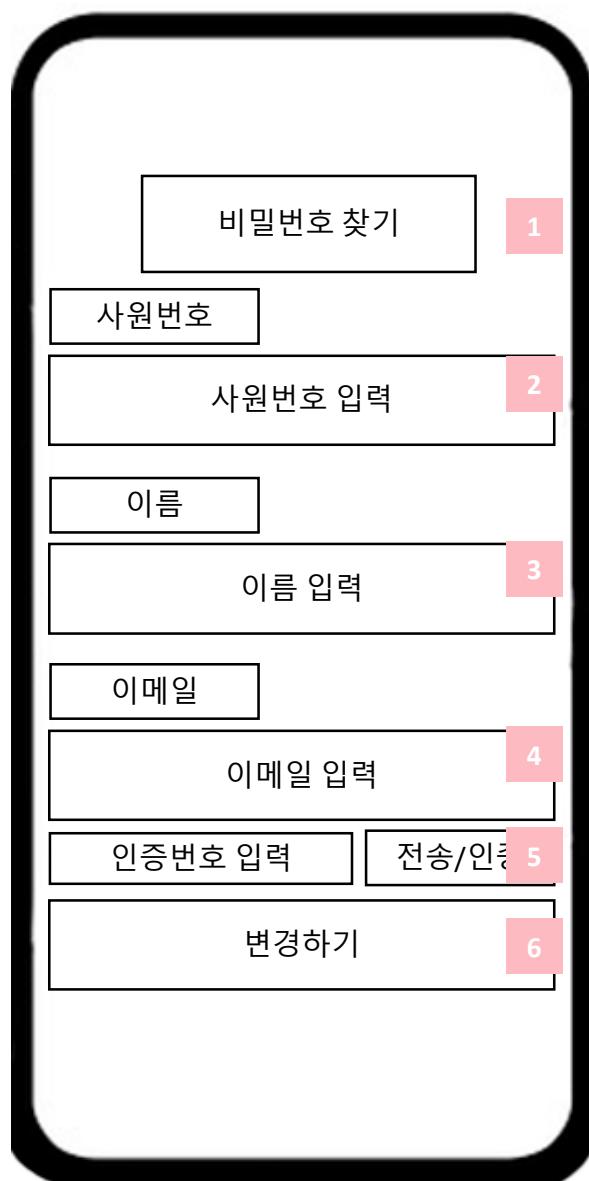
4 확인  
- 이름, 이메일 일치하는  
사원번호 출력

- 이름 & 이메일이 일치하는 사원번호 출력



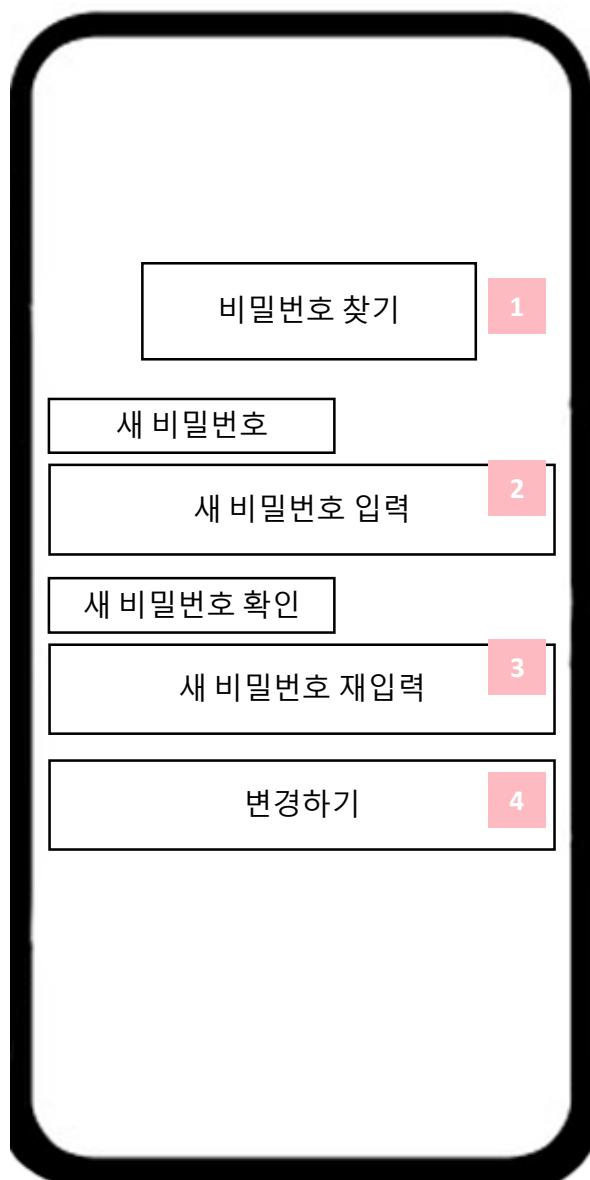
- 1 사원번호 찾기  
- 텍스트 내용
- 2 사원번호  
- 텍스트 내용  
사원번호 출력  
- 일치하는 사원번호 출력
- 3 로그인  
- 로그인(메인화면) 이동
- 4 비밀번호 찾기  
- 페이지 이동

- 사원번호 & 이름 & 이메일이 일치하면 비밀번호 재설정



- 1 비밀번호 찾기  
- 텍스트 내용
- 2 사원번호  
- 텍스트 내용  
사원번호 입력  
- 텍스트 입력 받기
- 3 이름  
- 텍스트 내용  
이름 입력  
- 텍스트 입력 받기
- 4 이메일  
- 텍스트 내용  
이메일 입력  
- 텍스트 입력 받기
- 5 인증번호 입력  
- 텍스트 입력 받기  
전송/인증  
- 인증번호 전송/인증 버튼
- 6 변경하기  
- 비밀번호 변경 페이지 이동

- 비밀번호 변경하기



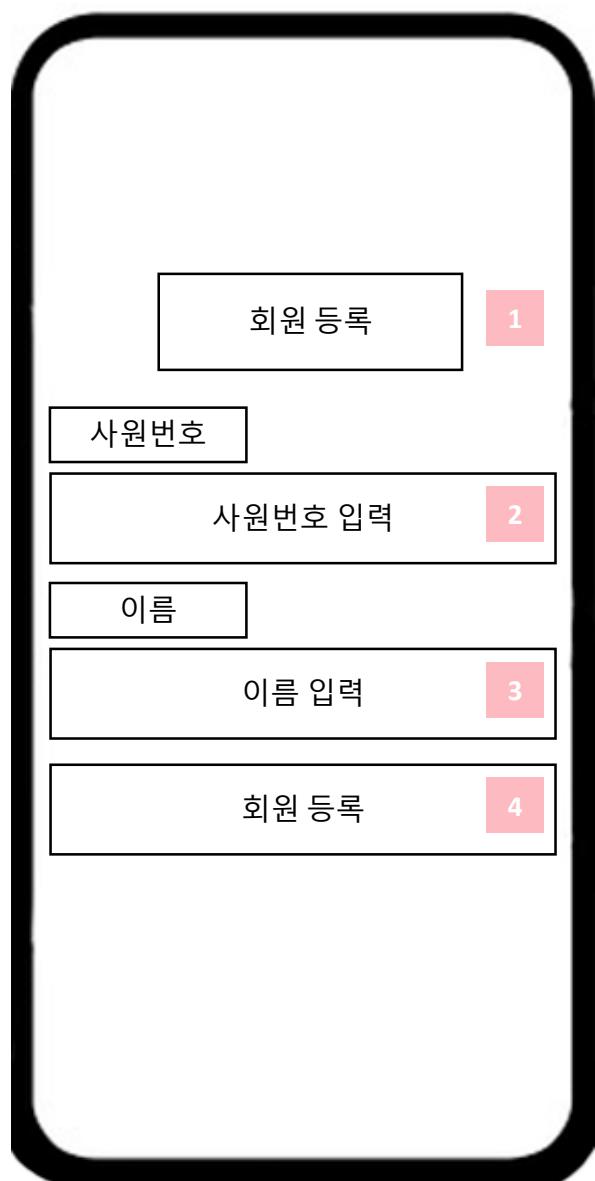
1 비밀번호 찾기  
- 텍스트 내용

2 새 비밀번호  
- 텍스트 내용  
새 비밀번호 입력  
- 텍스트 입력 받기(정규식)

3 새 비밀번호 확인  
- 텍스트 내용  
새 비밀번호 재입력  
- 텍스트 입력 받기(정규식)

4 변경하기  
- 비밀번호 일치 여부 확인  
- 로그인 페이지 이동

- 사원번호 & 이름으로 회원 등록



● 회원 등록 정보 입력

회원 등록

1

사원번호

2

사원번호 출력

3

비밀번호

4

비밀번호 입력

5

비밀번호 재입력

6

일치 확인

희망 연락처

7

희망 연락처 입력

8

핸드폰

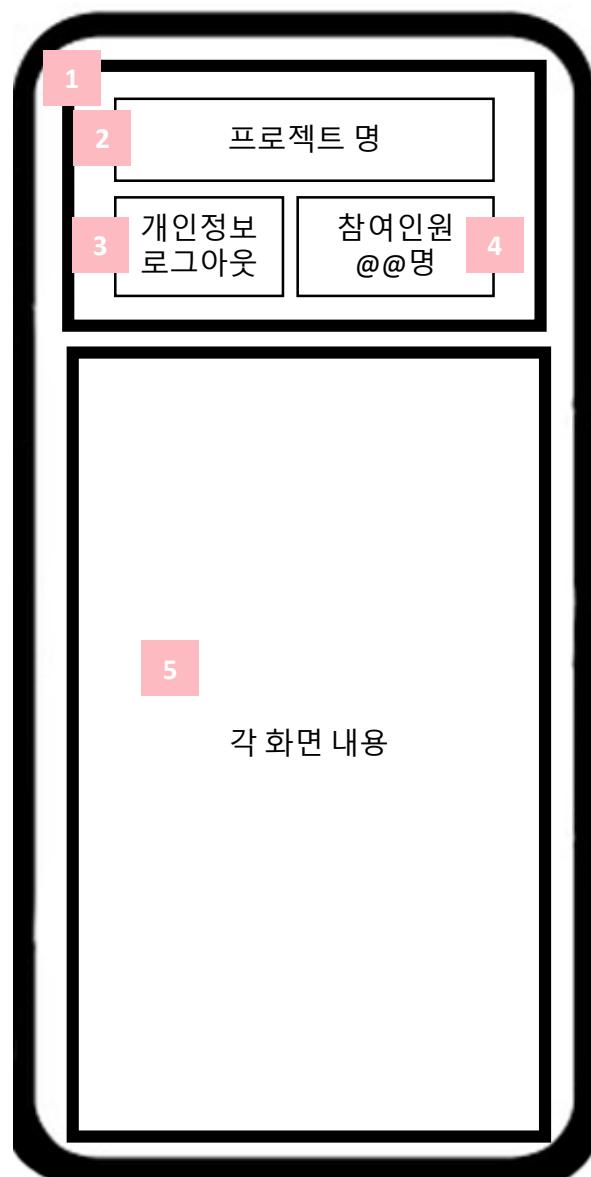
이메일

회원 등록

9

- 1** 비밀번호 찾기
  - 텍스트 내용
- 2** 사원번호
  - 텍스트 내용
  - 사원번호 입력
  - 텍스트 입력 받기
- 3** 이름
  - 텍스트 내용
  - 이름 입력
  - 텍스트 입력 받기
- 4** 이메일
  - 텍스트 내용
  - 이메일 입력
  - 텍스트 입력 받기
- 5** 인증 번호 입력
  - 텍스트 입력 받기
  - 전송/인증
  - 인증 번호 전송/인증 버튼
- 6** 변경하기
  - 비밀번호 변경 페이지 이동

- 프로젝트명 & 개인정보 & 로그아웃 & 참여인원



- 1 상단 메뉴
  - Include 사용
- 2 프로젝트 명
  - 참여하고 있는 프로젝트 명 출력(팀원/팀장)
- 3 개인정보
  - 개인정보 페이지 이동
  - 사원번호, 비밀번호, 이름, 부서명, 직책, 희망연락처
  - 로그아웃
  - 로그인 세션 정보 제거
  - 로그인 화면으로 이동
- 4 참여 인원
  - 프로젝트에 참여한 인원 중 로그인 된 인원 수
  - 클릭 하면 참여 인원의 정보 출력(팝업)  
(이름, 직책, 희망연락처)
- 5 각 화면 내용
  - 각 화면 내용 출력

- 개인 정보 확인 팝업 화면(비밀번호 입력)



1 개인 정보  
- 텍스트 내용

2 사원번호  
- 텍스트 내용  
사원번호 출력  
- 로그인 된 사원번호 출력

3 비밀번호  
- 텍스트 내용  
비밀번호 입력  
- 텍스트 입력 받기(정규식)

4 정보 확인  
- 비밀번호 일치 여부 확인  
- 정보 확인/수정 팝업

● 개인정보 확인 및 수정

개인 정보 1

사원번호 2

사원번호 출력

이름

이름 출력

직책

직책 출력

부서

부서 출력

전화번호

전화번호 출력

이메일

이메일 출력

비밀번호 3

비밀번호 \* 출력

비밀번호 재입력

일치 확인

희망 연락처 4

희망 연락처 출력/수정

정보 수정 5

1 개인정보 팝업 화면  
- 정보 확인/수정 화면

2 정보 출력  
- 사원번호/이름/직책/부서  
- 전화번호/이메일

3 비밀번호  
- 비밀번호 \* 출력  
일치 확인  
- 비밀번호 일치 확인(정규식)

4 희망 연락처  
- 입력된 정보 출력  
희망 연락처 수정  
- 수정은 새로운 내용 입력

5 정보 수정  
- 비밀번호/희망연락처 수정

- 참여인원의 기본정보 및 로그인 여부

The image shows a smartphone screen with a black border. At the top, there is a header bar with three sections: '프로젝트 명' (Project Name), '개인정보 로그아웃' (Personal Information Logout), and '참여인원 @@명' (Participants @@). Below this is a table with four columns: '이름' (Name), '직책' (Position), '연락처' (Contact), and '로그인 상태' (Login Status). The first column ('이름') is highlighted with a pink box and labeled '1'. Below the table is a search bar with two buttons: '이름 입력' (Name Input) and '검색' (Search). The second button is also highlighted with a pink box and labeled '2'. Below the search bar is a table with five rows, each consisting of four empty fields. The first row has a pink box around its first field and is labeled '3'.

1 표 컬럼명  
- 텍스트 내용

2 이름 입력  
- 텍스트 입력 받기  
검색  
- 일치하는 정보 출력

3 리스트  
- 프로젝트 참여 인원  
리스트 출력

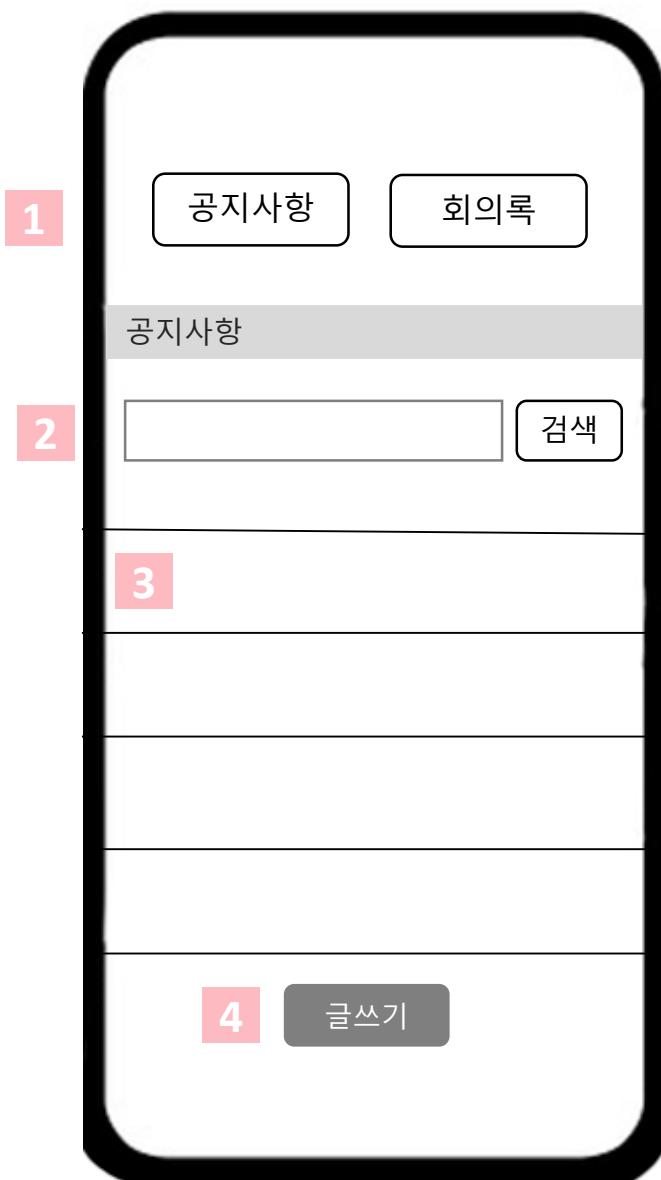


화면설계

# 프로젝트 의사소통관리

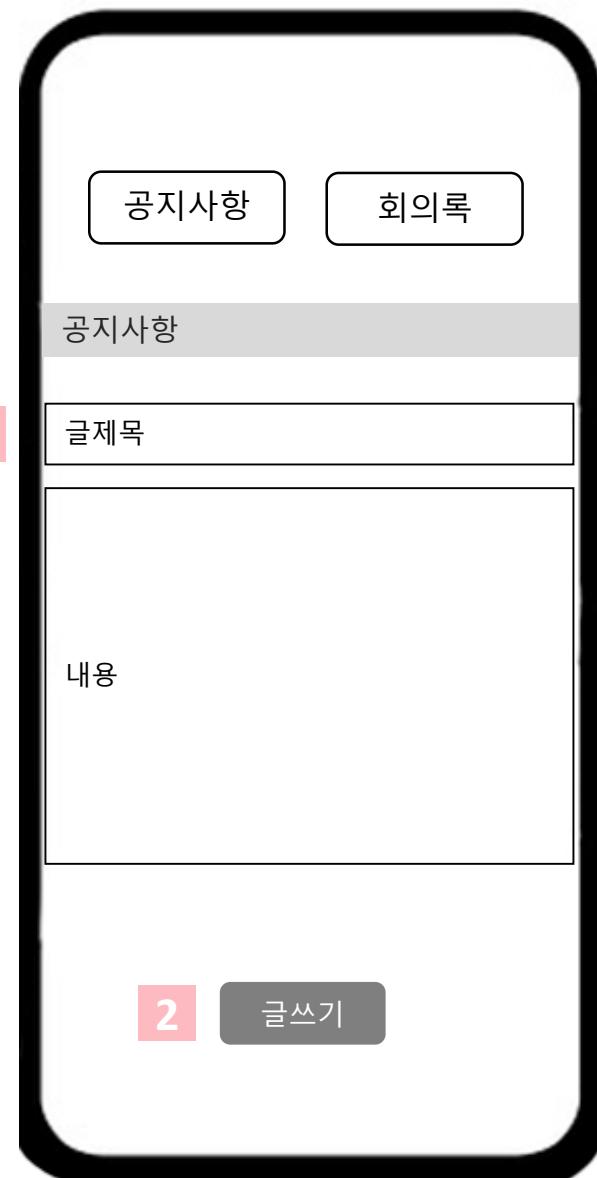
김하나

- 공지사항 리스트



- 1 공지사항/회의록 간 이동 버튼  
- 클릭시 각 페이지로 이동
- 2 검색 기능  
- 제목으로 리스트 검색
- 3 공지사항 리스트
- 4 글작성버튼

- 공지사항 글쓰기



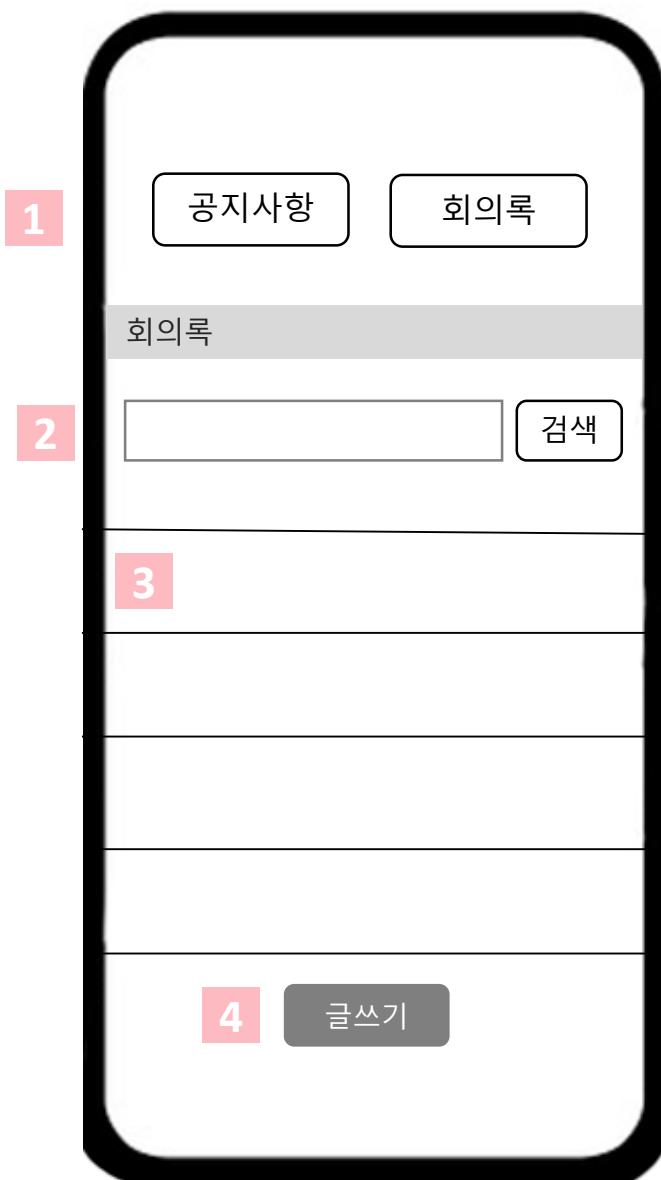
**1** 글작성

- 글제목과 글 내용을 입력

**2** 글쓰기 버튼

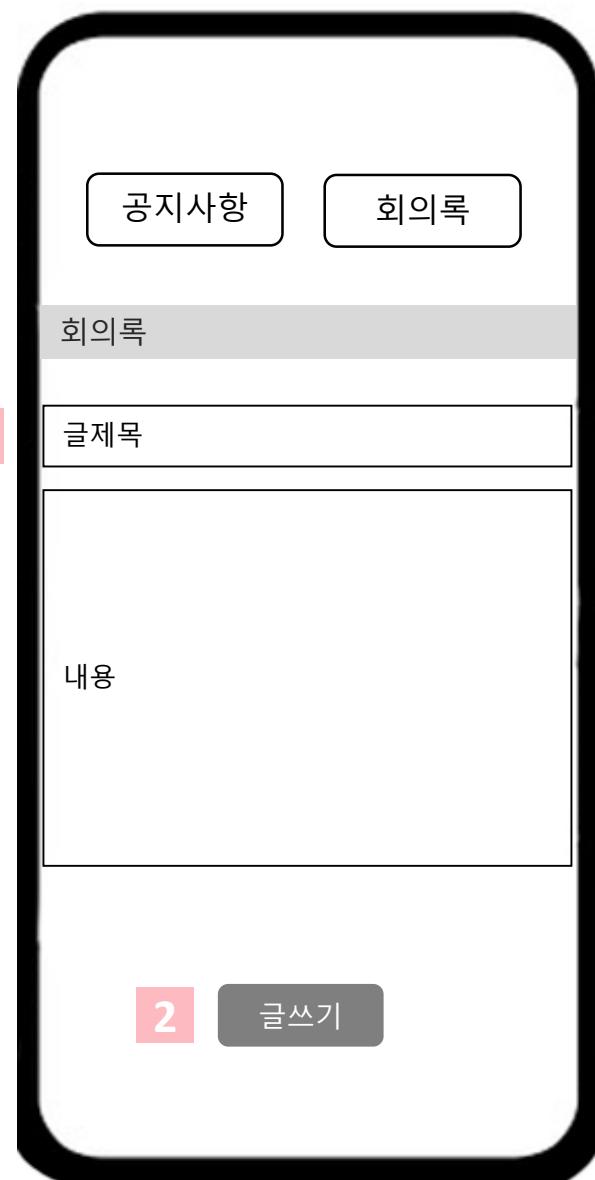
- 클릭시 등록

- 공지사항 리스트



- 1 공지사항/회의록 간 이동 버튼  
- 클릭시 각 페이지로 이동
- 2 검색 기능  
- 제목으로 리스트 검색
- 3 회의록 리스트
- 4 글작성버튼

- 회의록 글쓰기



**1** 글작성

- 글제목과 글 내용을 입력

**2** 글쓰기 버튼

- 클릭시 등록



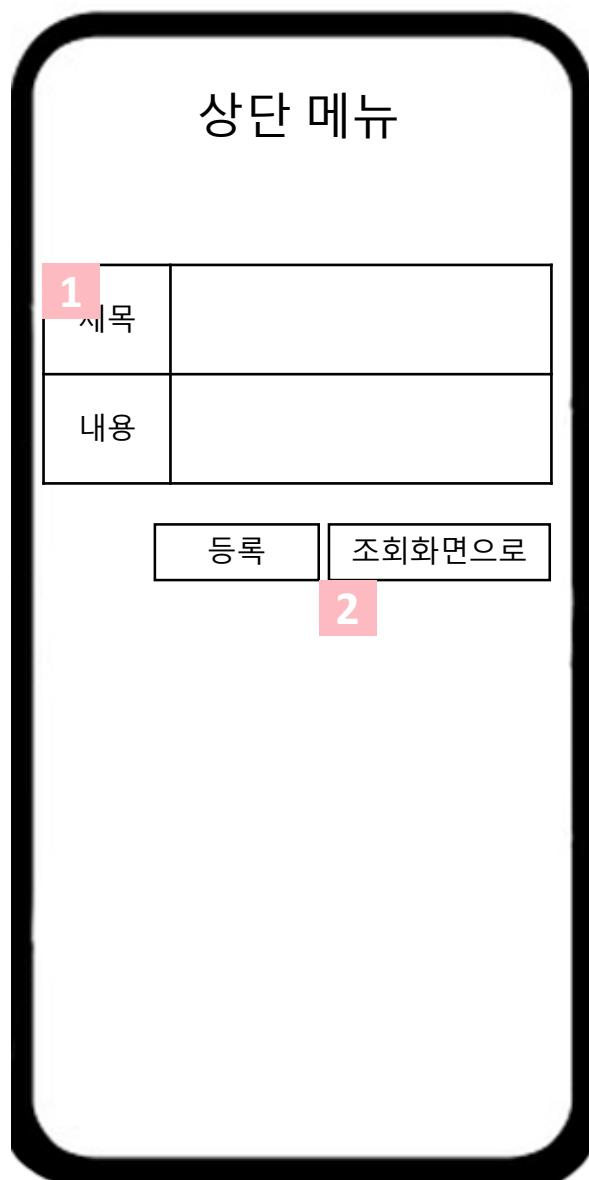
화면설계

# 프로젝트 리스크관리

이준석



- 1 검색 창
  - 제목으로 검색 가능
- 2 이슈리스트
  - 데이터 베이스에 등록된 이슈리스트 내용을 출력
- 3 글쓰기
  - 클릭 시 작성 페이지로 이동



1

입력 란

- 제목과 내용을 입력할 수 있다.

2

조회화면으로

- 이슈리스트 페이지로 이동한다.

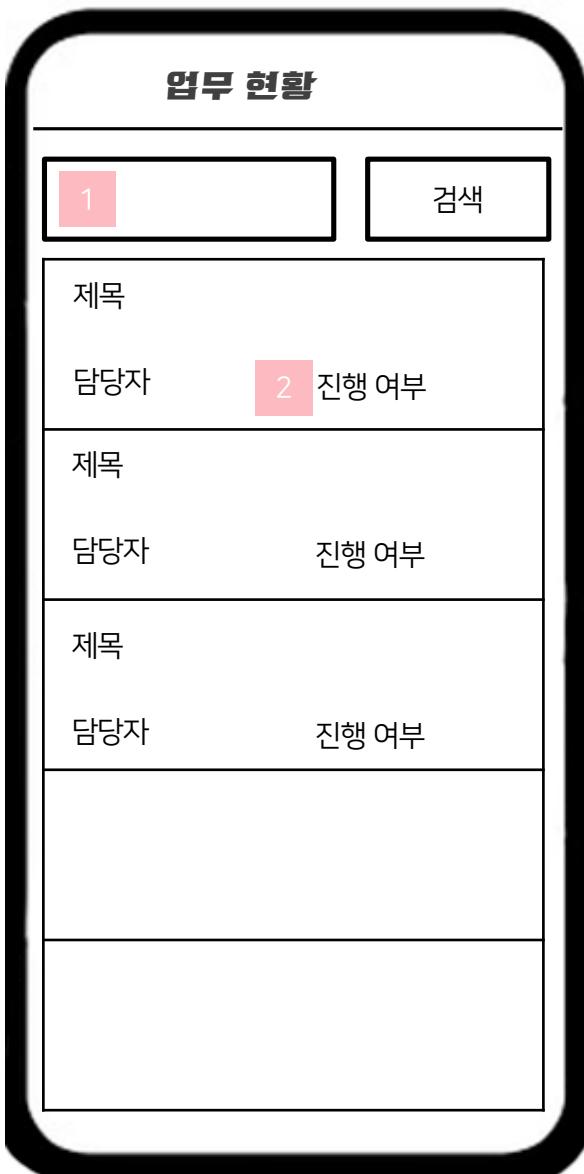


화면설계

# 프로젝트 통합관리

이연학

- 안드로이드 업무 리스트



- 1 검색 처리
  - 업무 명으로 검색
- 2 결재 여부
  - 업무 중 / 결재 / 반려처리



**화면구현**

**WEB**



화면구현

# 사원정보

김진수

### LOGIN

사원번호를 입력하세요.

비밀번호를 입력하세요.

사원번호 저장

[사원번호 찾기](#)      [비밀번호 찾기](#)

**LOGIN**

## Business logic

1. 사원번호 입력
2. 비밀번호 입력
3. 사원번호 저장(쿠키 이용)
4. 사원번호/비밀번호 찾기
  - 사원번호/비밀번호 찾기 페이지 이동
5. 로그인

```

$( "#ckSave" ).change(function(){
  // 사원번호 저장 체크
  if($("#ckSave").is(":checked")){
    setCookie("key", $("#mno").val(), 1);
  }else{ // 사원번호 저장 체크 해제
    deleteCookie("key");
  }
});
$("#mno").keyup(function(){
  if($("#ckSave").is(":checked")){
    setCookie("key", $("#mno").val(), 1);
  }
});

```

**// 쿠키 설정**

```
function setCookie(cookieName, value, exdays){
  var exdate = new Date();
  exdate.setDate(exdate.getDate() + exdays);
  var cookieValue = escape(value) + ((exdays==null)
    ? "" : "; expires=" + exdate.toGMTString());
  document.cookie = cookieName + "=" + cookieValue;
}
```

**// 쿠키 삭제**

```
function deleteCookie(cookieName){
  var expireDate = new Date();
  expireDate.setDate(expireDate.getDate() - 1);
  document.cookie = cookieName + "=" + "; expires=" +
    expireDate.toGMTString();
}
```

**LOGIN**

[사원번호 찾기](#)

▶ LOGIN

**mapper Select**

```
SELECT * FROM pmsmember
 WHERE mno=#{mno} AND pass=#{pass}
SELECT * FROM pmsemp WHERE eno=#{mno}
```

**mapper Insert**

```
INSERT INTO pmslog
values(pmslog_seq.nextval,1,sysdate,null,#{mno})
로그인 정보 저장
```

## controller

```
HttpSession session = request.getSession();

PmsMember infor_M = service.memberInforAll(member);
pmsemp emp = service.emplInfor(infor_M.getMno());

if(infor_M.getMno()!=0) {
    if(infor_M.getMdiv() == 8 || infor_M.getMdiv() == 9) {
        return "WEB-INF\www\views\wwwmain\login.jsp";
    } else if(infor_M.getMdiv() == 7) {
        session.setAttribute("infor_M", infor_M);
        return "forward:/PMSEmp.do?method=empList";
    } else {
        session.setAttribute("infor_M", infor_M);
        session.setAttribute("emp", emp);
        return "forward:/DashTeam.do?method=list";
    }
} else {
    return "WEB-INF\www\views\wwwmain\login.jsp";
}
```

권한(mdiv)별 로그인 여부  
PmsMember / PmsEmp 테이블 정보를 세션 값으로 생성

FIND PASSWORD

사원번호  
사원번호를 입력하세요

이름  
이름을 입력하세요

이메일  
이메일을 입력하세요

인증번호  
 전송



**Business logic**

1. 사원번호/이름 입력
2. 이메일 입력 -> 전송
  - 입력한 이메일로 메일 전송
3. 전송 -> 인증 -(클릭)-> 인증번호 체크
  - 전송된 인증번호와 입력 값 비교
4. 모두 일치하면 비밀번호  
재설정(변경) 페이지 이동

**service****인증번호 생성**

```
for(int i=0; i<8; i++) {
    int randomNum = (int)(Math.random()*3);
    if(randomNum == 0) {
        sendNum += (char)((Math.random()*26)+97);
    }else if(randomNum == 1) {
        sendNum += (char)((Math.random()*26)+65);
    }else if(randomNum == 2) {
        sendNum += (int)((Math.random()*10));
    }
}
```

아스키코드 이용 : (char)정수 -> 문자 변환

**Dispatcher-Servlet****SMTP 서버 설정**

```
<bean id="mailSender"
      class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host" value="smtp.gmail.com"/>
    <property name="port" value="587"/>
    <property name="username" value=" id"/> <!-- 구글 아이디 -->
    <property name="password" value="pass"/> <!-- 구글 비번 -->
    <property name="JavaMailProperties">
      <props>
        <prop key="mail.smtp.auth">true</prop>
        <prop key="mail.smtp.starttls.enable">true</prop>
      </props>
    </property>
</bean>
```

### FIND PASSWORD

※ 8~20자리, 영문, 숫자, 특수문자 조합

새 비밀번호  
새 비밀번호를 입력하세요  
8~20자리로 적어주세요

새 비밀번호 확인  
새 비밀번호 재입력

**CHANGE**

**mapper**

**Update**

```
UPDATE pmsmember SET pass=#{pass}
WHERE mno=#{mno}
```

**Business logic**

- 새 비밀번호 입력
  - 8~20자리, 영문, 숫자, 특수문자 조합
  - 정규식 이용 판별
- CHANGE
  - DataBase 비밀번호 Update

**view**

```
var num = pass01.search(/[0-9]/g);
var eng = pass01.search(/[a-z]/ig);
var spe = pass01.search(/\`~!@#$%^&*::₩/?]/gi);

if(pass01.length < 8 || pass01.length > 20) {
  return 1;
} else if(pass01.search(/\Ws/) != -1) {
  return 2;
} else if(num < 0 || eng < 0 || spe < 0) {
  return 3;
} else{
  return 0;
}
```

8~20자리, 영문, 숫자, 특수문자 조합 조건 판별

**view**

```
var ck = "${ck}"; // 구분 번호
if(ck == 1){
  Swal.fire({
    펑업내용
  }).then((result) => {
    if (result.value) { // OK 클릭 시
      $(location).attr("href","${path}/main.do");
    }
  })
}
```

비밀번호 변경 완료 팝업 출력 후 메인으로 이동

**PROJECT PMS 시스템 개발**

채팅 개인정보 참여인원 LogOut

**INFORMATION**

사원번호: 10000005  
이름: 김대중  
직책: 부장  
부서: 개발1팀  
전화번호: 010-2175-9239  
이메일: DJ1998@daum.net  
새 비밀번호: 새 비밀번호를 입력하세요  
비밀번호를 다시 입력하세요  
희망 연락처: 희망 연락처 입력하세요  
○전화번호 ○이메일

**view** Session 값 활용  
value="\${infor\_M.mno}",  
value="\${emp.name }",  
value="\${emp.grade }",...

**mapper**

```
UPDATE pmsmember SET pass=#{pass}
<if test="wcon!=null and wcon!=''>
    , wcon=#{wcon}
</if> WHERE mno=#{mno}
```

**view**

```
$("#[name=ck]").click(function(){
    var ckPhone = $("#ckPhone").is(":checked");
    var ckMail = $("#ckMail").is(":checked");
    if(ckPhone){
        var txt = $("#phone").val();
        $("#con").val(txt);
    }else if(ckMail){
        var txt = $("#mail").val();
        $("#con").val(txt);
    }
});
```

전화번호&이메일 체크 -> 희망연락처 자동 입력

**INFORMATION**

사원번호: 10000005  
비밀번호: 비밀번호를 입력하세요  
**CHECK**

**Business logic**

- 비밀번호 입력
  - Check 개인정보 확인
- 비밀번호 변경
- 희망연락처 등록
  - 전화번호/이메일 체크
  - 등록된 정보 이용


**PROJECT PMS시스템 개발**

채팅

개인정보

참여인원

LogOut

**service**

```

sch.setCount(dao.empLogCount(sch));
sch.setPageSize(5);
sch.setPageCount((int)(Math.ceil(sch.getCount())/(double)sch.getPa

if(sch.getCurPage()==0) {
sch.setCurPage(1);

sch.setStart((sch.getCurPage()-1)*sch.getPageSize()+1);
sch.setEnd(sch.getCurPage()*sch.getPageSize());

sch.setBlocksize(5);

int blocknum = (int)Math.ceil(sch.getCurPage()/(double)sch.getBlocksize());

int endBlock = blocknum*sch.getBlocksize();
sch.setEndBlock(endBlock>sch.getPageCount()
    ?sch.getPageCount():endBlock);

sch.setStartBlock((blocknum-1)*
    sch.getBlocksize()+1);

return dao.logList(sch);
}

```

**PARTICIPANTS**

PARTICIPANTS			
사원 이름	Search		
이름	직책	희망 연락처	상태
김대중	부장	010-2175-9239	로그인
추미애	차장	010-2415-2920	로그아웃
오세훈	과장	010-6002-7753	로그아웃
안희정	대리	010-9589-3836	로그인
박영선	대리	010-1543-2532	로그아웃

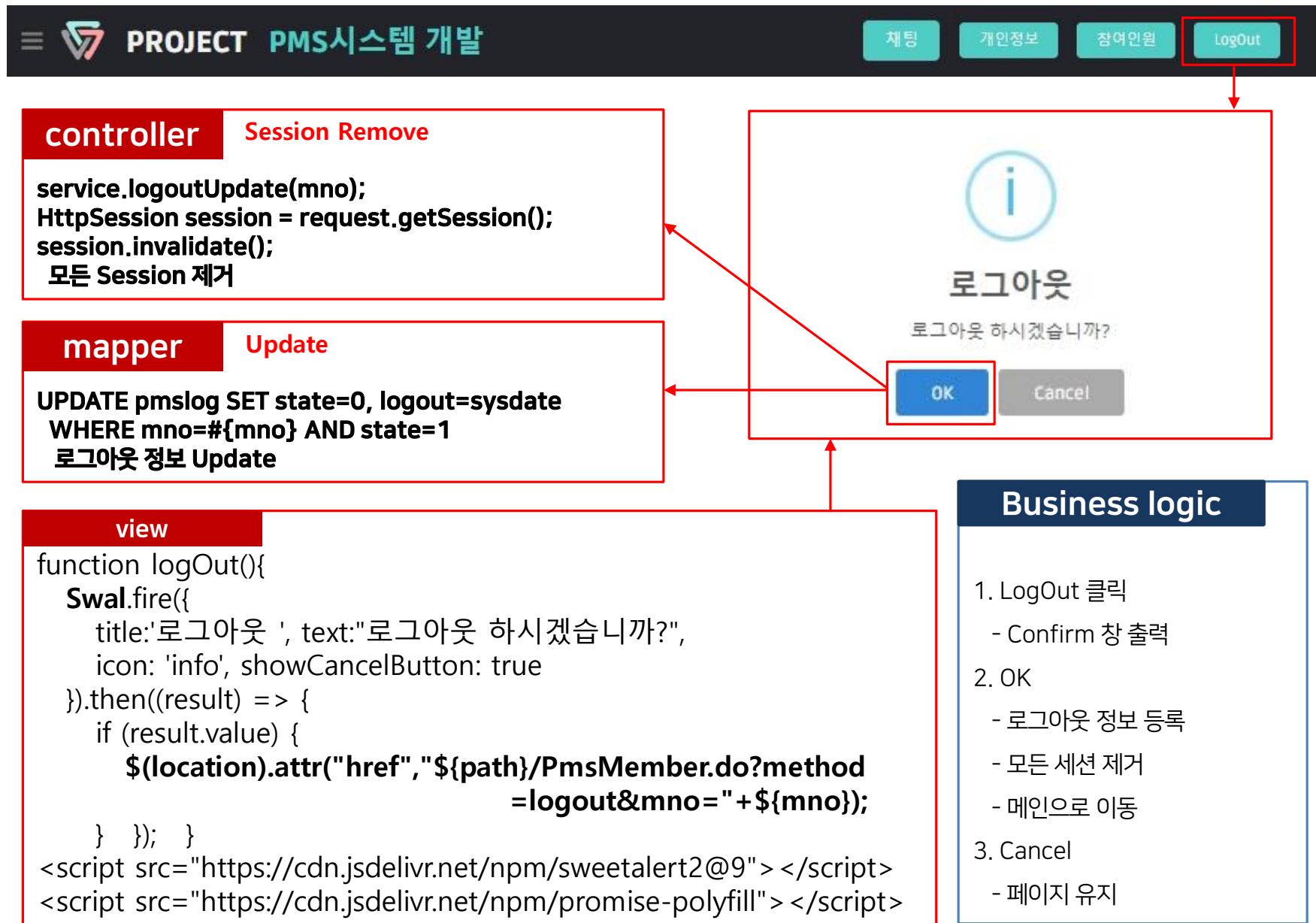
Previous 1 2 Next

**mapper**

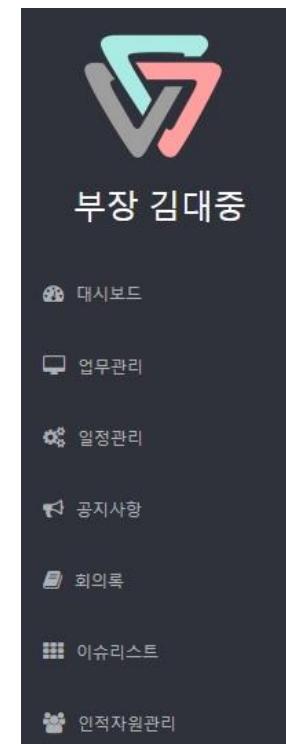
```

SELECT * FROM
(
  SELECT rownum cnt, e.name, e.grade, m.wcon,
  CASE WHEN l.state = 0 THEN '로그아웃' ELSE '로그인' END AS state
  FROM PMSEMP e, PMSMEMBER m, PMSLOG l
  WHERE e.eno=m.mno AND m.mno=l.mno
  AND login=(SELECT max(login) FROM pmslog WHERE mno=m.mno)
  <if test="pno!=null and pno!="">
    AND m.pno=#{pno} </if>
  <if test="name!=null and name!="">
    AND e.name like '%'||#{name}||'%'
  </if> ORDER BY m.mno
) WHERE cnt BETWEEN #{start} AND #{end}

```



## 사이드 메뉴



## view

```
var vm = new Vue({
    el:"#show",
    data:{mdiv:$infor_M.mdiv}
});
```

```
<li v-if="mdiv==2||mdiv==3||mdiv==6||mdiv==4||mdiv==5">메뉴 내용01</li>
<li v-if="mdiv==6||mdiv==4||mdiv==5">메뉴 내용02</li>
```

## Business logic

## 1. 로그인

- 회원정보의 권한 비교
- 권한에 따라 v-if로 메뉴 내용 보이기

## 2. 메뉴 클릭

- 메뉴에 연결된 페이지 이동



화면구현

# 프로젝트 의사소통관리

김하나



부장 김대중

 대시보드 업무관리 일정관리 공지사항 회의록 이슈리스트 인적자원관리 공지사항페이지 크기: 

제목

Search

번호	제목	작성일	조회수
11	회의 공지	2020. 5. 11	15
10	이슈 발생시 바로 이슈리스트 등록 바랍니다	2020. 5. 8	0
9	주말 과제 공지	2020. 5. 8	15
8	업무 분담 숙지 바랍니다	2020. 5. 7	0
7	1주차 발표에 대한 피드백 회의 공지	2020. 5. 7	10

Previous 1 2 3 Next

✓ 글쓰기

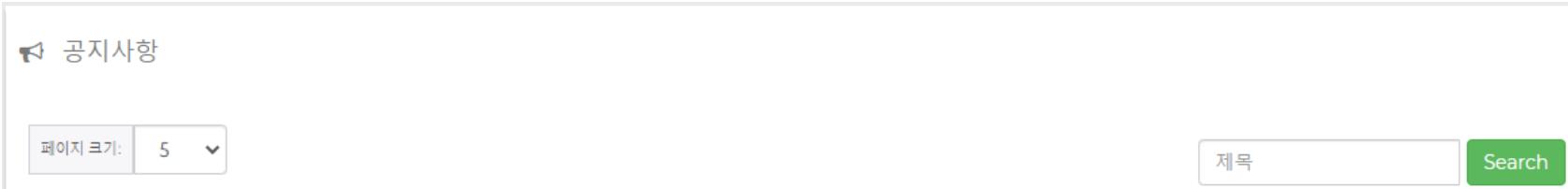
번호	제목	작성일	조회수
11	회의 공지	2020. 5. 11	15
10	이슈 발생시 바로 이슈리스트 등록 바랍니다	2020. 5. 8	7
9	주말 과제 공지	2020. 5. 8	15
8	업무 분담 숙지 바랍니다	2020. 5. 7	5
7	1주차 발표에 대한 피드백 회의 공지	2020. 5. 7	10
6	기한 내 작업 완수 목표를 꼭 달성합시다	2020. 5. 4	1
5	5월5일 휴일 과제 공지	2020. 5. 4	20
4	회의 공지	2020. 5. 4	25
3	프로젝트에 활용할 bootstrap template 투표	2020. 5. 4	20
2	프로젝트 역할 분담에 대한 회의 일정 공지	2020. 5. 4	17

## mapper

```
<select id="list" resultMap="noticeResult" parameterType="nsch" >
    SELECT *
    FROM(
        select ROW_NUMBER() OVER(ORDER BY nno desc) cnt, p.*
        FROM pmsnotice p
    WHERE 1=1
    <if test="title!=null and title!=''">
        and title like '%'||#{title}||'%'
    </if>
    ORDER BY nno DESC )
    WHERE cnt BETWEEN #{start} AND #{end}
</select>
```

## Business logic

1. Controller에 list 요청
2. DB작업을 위해 Service단에서 Dao 호출
3. Mapper에서 list 요청 쿼리를 찾아 데이터를 저장해서 Service에 넘김
4. Service → Controller → jsp에 리스트 내용을 보여줌



## controller

```
@RequestMapping(params="method=list")
public String list(@ModelAttribute("nsch") NoticeSch sch, Model d, HttpServletRequest request) {
    d.addAttribute("nlist", service.list(sch));
    return "WEB-INF\\views\\main\\noticeList.jsp";
}
@RequestMapping(params="method=Ajaxlist")
public String ajaxlist(NoticeSch sch, Model d) {
    d.addAttribute("nlist", service.list(sch));
    return "jsonReport";
}
```

## service

```
@Service
public class NoticeService {
    @Autowired(required=false)
    private NoticeDao dao;
    public ArrayList<Notice> list(NoticeSch sch){

        if(sch.getPageSize()==0) {
            sch.setPageSize(5); // 화면에 한번에 보여줄 데이터 총 건수 초기값 = 5
        }
        sch.setPageCount((int)(Math.ceil(sch.getCount())/(double)sch.getPageSize())));
    }
}
```

## Business logic

1. jsp에서 form태그와 script 작성
2. 페이지 크기 변화, 검색기능을 Ajax로 처리
3. 서비스 단에서 페이지 크기와 크기 설정 시, 한페이지에 보이는 게시글의 개수를 설정
4. 페이지 크기를 선택한만큼 화면에 게시글 개수가 표시됨

8	업무 분담 숙지 바랍니다	2020. 5. 7	5
7	1주차 발표에 대한 피드백 회의 공지	2020. 5. 7	10

Previous 1 2 3 Next

## view

```
function goPage(no){
    $("#curPage").val(no);
    $("form").attr("action", "${path}/notice.do?method=list");
    $("form").submit();
}

<c:forEach var="cnt" begin="${nsch.startBlock }" end="${nsch.endBlock}">
    <li class="page-item ${nsch.curPage==cnt?'active':''}">
        <a class="page-link" href="javascript:goPage(${cnt})">${cnt}</a></li>
</c:forEach>
```

## service

```
sch.setBlocksize(5);

int blocknum = (int)Math.ceil(sch.getCurPage()/(double)sch.getBlocksize());
int endBlock = blocknum*sch.getBlocksize();
sch.setEndBlock(endBlock>sch.getPageCount()?
                sch.getPageCount():endBlock);
sch.setStartBlock((blocknum-1)*sch.getBlocksize()+1);

return dao.list(sch);
```

## Business logic

1. jsp에서 form 태그와 script작성
2. pagination을 Ajax로 처리
3. 서비스 단에서 block 사이즈 설정
4. 페이지 크기에 따라 pagination 범위 자동 조절

≡ V PROJECT PMS시스템 개발

제팅 개인정보 참여인원 LogOut

부장 김대중

대시보드 업무관리 일정관리 공지사항 회의록 이슈리스트 인적자원관리

## ▶ 공지사항

제목  
제목입력하세요

내용  
내용입력하세요

등록 조회 화면으로

제목

내용

등록 조회 화면으로

## controller

```
@RequestMapping(params="method	insert")
public String insert(Notice insert) {
    service.insert(insert);

    insert.setTitle("");
    insert.setDetail("");
    return "WEB-INF\\views\\main\\noticeInsert.jsp";
}
```

## mapper

```
<insert id="insert" parameterType="notice">
    insert into pmsnotice values(pmsnotice_seq.nextval,#{title},
                                    #{detail},sysdate,null,0,1001,10000005)
</insert>
```

## Business logic

- 내용을 입력하고 등록 버튼을 누르면
- controller 단에서 insert Service를 호출
- Service에서 Dao 호출
- Dao에서 mapper를 연결하여 DB에 새 글을 추가

 PROJECT PMS시스템 개발

채팅

개인정보

참여인원

LogOut



부장 김대중

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

인적자원관리

## 공지사항

## 제 목

회의 공지

## 글번호

11

## 조회수

27

## 등록일

2020. 5. 11 오전 12:00:00

## 수정일

## 내 용

5월 11일 오후 1시 30분, 507호 강의실에서 ERD 설계에 대한 회의가 있을 예정입니다.

[조회 화면으로](#)[수정](#)[삭제](#)

조회수

27

**dao**

```
public void uptReadCnt(int no);
```

**mapper**

```
<update id="uptReadCnt" parameterType="int">
    UPDATE pmsnotice
        SET views = views+1
    WHERE nno = #{no}
</update>
```

**service**

```
public Notice getNotice(int no) {
    dao.uptReadCnt(no);
    Notice n = dao.getNotice(no);

    return n;
}
```

**Business logic****조회수 증가**

1. dao에 조회수를 1 증가시키는 메소드 작성
2. mapper에 글번호를 가지고 조회수 1 증가시키는 sql 작성
3. servic에서 조회수 1 증가시키는 메소드를 호출하여 하나의 데이터를 읽고 리턴한다

case1.

CEO/PM으로 로그인



case2.

사원으로 로그인

## view

```

<script>
var vm = new Vue({
  el:"#reg",
  data:{mdiv:${infor_M.mdiv}}
});
</script>

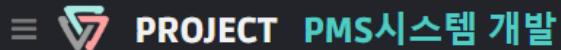
<div id="reg">
  <button id="regBtn" type="button" class="btn btn-theme" v-if="mdiv!=5"
    style="margin-left: 90%; "><i class="fa fa-check"></i> 글쓰기 </button>
</div>

```

## Business logic

### 권한에 따라 버튼 보이기

- 로그인 시 infor\_M 세션 생성
- 스크립트에서 'Vue' 사용. 'reg' 라고 아이디를 선언하고, infor\_M.mdiv의 데이터 가져오기
- mdiv가 적용될 버튼에 div로 세션을 주고 뷰에서 지정한 아이디 선언.
- 버튼 속성에서 V-if로 사원을 제외한 모든 사람이 버튼을 볼 수 있게 설정

 PROJECT PMS시스템 개발

채팅

개인정보

참여인원

LogOut



부장 김대중

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

인적자원관리

## 회의록

페이지 크기: 

제목

Search

번호	제목	작성일	작성자	조회수
6	ERD 설계	2020. 5. 12	안희정	9
5	1주차 발표 피드백 개선방안논의	2020. 5. 9	추미애	18
4	ERD 설계	2020. 5. 6	오세훈	20
3	bootstrap template 결정	2020. 5. 5	안희정	15
2	프로젝트 역할 분담	2020. 5. 4	추미애	20

Previous

1

2

Next

✓ 글쓰기

부장 김대중

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

회의록

페이지 크기: 

## PMS 시스템 개발 회의록

번호

6	ERD 설계
5	1주차 발표 피드백 개선방안
4	ERD 설계
3	bootstrap template 결정

## 사원 문희상

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

회의록

페이지 크기: 

## 소모고 회의록

번호

제

5	mvc2 모델 적용 방법
4	동행 게시판 추가여부
3	포인트 테이블 작성여부
2	프로젝트 명 선정

## controller

```
PmsMember m = (PmsMember) session.getAttribute("infor_M");
sch.setPno(m.getPno());
```

## mapper

```
<select id="list" resultMap="meetResult" parameterType="msch" >
    SELECT *
    FROM(
        SELECT ROW_NUMBER() OVER(ORDER BY mnno desc) cnt, mnno,topic,detail,decision,
        etc,views,wdate,chdate,mdate,loc,people,mndiv,m.name writer,pno
        FROM pmsmeeting p, pmsemp m
        WHERE 1=1 AND p.mno=m.eno and pno=#{pno}
        <if test="topic!=null and topic!=''">
        and topic like '%'||#{topic}||'%'
        </if>
        ORDER BY mnno desc)
        WHERE cnt BETWEEN #{start} AND #{end}
</select>
```

## Business logic

1. 세션에서 pno를 받아
2. 담당 프로젝트별로 보이는 게시글이 다르게 설정

 PROJECT 쇼모고 개발

채팅

개인정보

참여인원

Logout



사원 문희상

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

## 회의록

안건

안건을 입력하세요

회의 참여자

참여자를 입력하세요(,를 사용하여 구분)

회의 장소

회의 장소를 입력하세요

내용

내용을 입력하세요

결정 사항

내용을 입력하세요

특이 사항

입력하세요

첨부 파일

파일 선택

Browse

등록

조회 화면으로



## service

```

@Value("${upload3}")
private String upload3; // 업로드할 위치..
@Value("${tmpUpload3}")
private String tmpUpload3;// 임시업로드 위치.
private void upload(MultipartFile mtf) {
    String fileName=mtf.getOriginalFilename();
    if(fileName!=null&&!fileName.equals("")) {
        File tmpFile = new File(tmpUpload3+fileName);
        if(tmpFile.exists()) tmpFile.delete();
        try {
            mtf.transferTo(tmpFile);
            File orgFile = new File(upload3+fileName);
            Files.copy(tmpFile.toPath(), orgFile.toPath(),
                StandardCopyOption.REPLACE_EXISTING);

        } catch (IllegalStateException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        dao.insertRepo(fileName);
    }
}
  
```

## Business logic

1. 파일을 선택하고 등록 버튼을 누르면
2. 서버에 파일이 등록되고 DB에는 파일 이름이 등록됨



부장 김대중

 대시보드 업무관리 일정관리 공지사항 회의록 이슈리스트 인적자원관리 회의록

안건

ERD 설계

작성자

안희정

회의참석자

김대중, 추미애, 박영선, 정은경 xffd

회의장소

507호

급변호

10

조회수

30

중략

첨부 파일

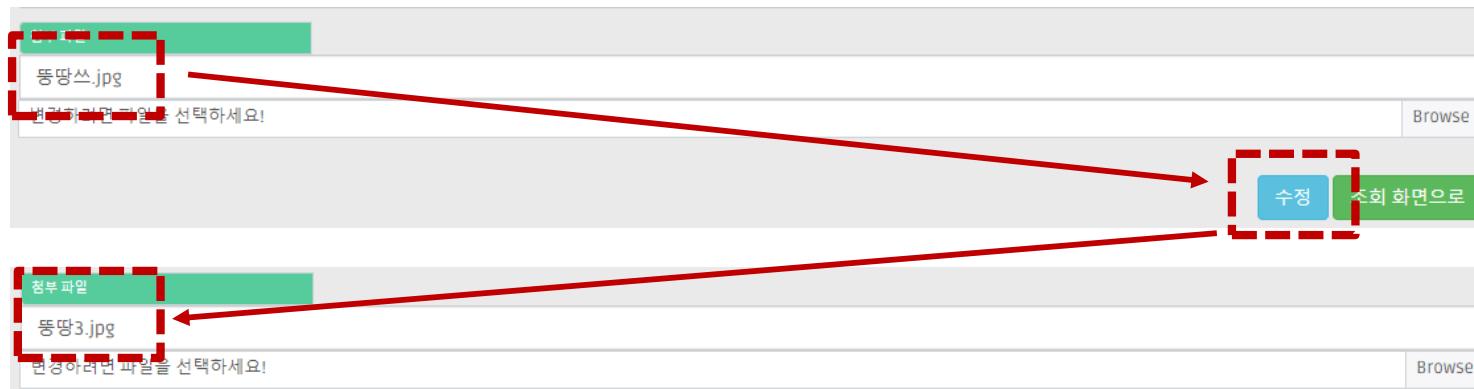
뚱땅쓰.jpg

변경하려면 파일을 선택하세요!

Browse

수정

조회 화면으로



## service

```

private void upload3(Meet upt) {

    MultipartFile[] files = upt.getReport();
    for(int idx=0;idx<files.length;idx++) {
        String fileName=files[idx].getOriginalFilename();
        if(fileName!=null&&!fileName.equals("")) {
            File tmpFile = new File(tmpUpload3+fileName);
            // 해당 폴더에 동일한 파일이 있으면 삭제 처리
            if(tmpFile.exists()) tmpFile.delete();
            try {
                // Stream으로 온 MultipartFile을 실제 파일로 변경처리.
                files[idx].transferTo(tmpFile);
                File orgFile = new File(upload3+fileName);

                // tmp위치에 있는 파일을 현재 웹서버에 특정할 폴더로 이동.
                // StandardCopyOption.REPLACE_EXISTING : 동일 파일명이 있을 때,
                // 마지막에 올린 파일로 변경
                Files.copy(tmpFile.toPath(), orgFile.toPath(),
                        StandardCopyOption.REPLACE_EXISTING);

            } catch (IllegalStateException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}

```

## Business logic

1. Browse 버튼을 눌러 변경할 파일을 선택
2. 수정버튼을 누르면 변경된 파일이 웹서버에 저장됨 (DB에도 변경된 파일로 이름 변경)



화면구현

# 프로젝트 리스크관리

이준석



부장 노태우

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

인적자원관리

## ■■■ 이슈리스트

페이지 크기: 5      프로젝트 번호: 1001      선택

번호

제목

작성자

제목

작성자

검색

18	결제화면	홍준표	2020. 5. 27	2
17	쿠폰 테이블	정은경	2020. 5. 25	2
16	로그인 문제	김무성	2020. 5. 24	0
15	insert	주미애	2020. 5. 24	0
14	update	오세훈	2020. 5. 22	0

« 1 2 3 4 »

✓글쓰기

## ■ 이슈리스트

페이지 크기: 5 프로젝트 번호: 1001 선택

번호	제목	작성자	작성일	조회수
18	결제화면	홍준표	2020. 5. 27	2
17	쿠폰 테이블	정은경	2020. 5. 25	2
16	로그인 문제	김무성	2020. 5. 24	0
15	insert	추미애	2020. 5. 24	0
14	update	오세훈	2020. 5. 22	0

« 1 2 3 4 »

✓글쓰기

## mapper

```
<select id="list" resultMap="riskResult" parameterType="rsch" >
select *
from(
SELECT rownum cnt,pa.*
FROM (SELECT p.* ,pm.NAME FROM PMSEMP pm,PMSISSUE p WHERE pm.ENO =p.MNO
WHERE 1=1
<if test="title != null and title != ''">
and title like '%'||#{title}||'%'
</if>
<if test="name != null and name != ''">
and name like '%'||#{name}||'%'
</if>
<if test="pno != null and pno != ''">
and pno like '%'||#{pno}||'%'
</if>
ORDER BY ino DESC)
where cnt between #{start} and #{end}
</select>
```

## Business logic

- 화면에 최근 작성된 글이 위로 오게 정렬
- 초기값으로 5개씩 보이며 페이지 크기에서 10개, 20개로 변경 가능
- 제목, 작성자로 검색이 가능하며
- Admin은 프로젝트 번호로도 검색이 가능하다.
- 프로젝트 번호 선택은 admin만 가능
- 팀원, PM은 본인이 등록된 프로젝트만 보기 때문에 프로젝트 번호 선택창이 보이지 않는다.
- 글쓰기는 팀원, PM, Admin만 가능하며
- 그 외에는 보이지 않는다.

16 로그인 문제

김무성

2020. 5. 24

0

15 insert

추미애

2020. 5. 24

0

14 update

오세훈

2020. 5. 22

0

« 1 2 3 4 »

✓글쓰기

## service

```

sch.setBlocksize(5);
int blocknum = (int) Math.ceil(sch.getCurPage()
    /(double)sch.getBlocksize());
int endBlock = blocknum*sch.getBlocksize();
sch.setEndBlock(endBlock>sch.getPageCount()?
    sch.getPageCount():endBlock);
// 시작번호는 현재블럭의 번호 ( blocknum - 1)*현재블럭크기 + 1
sch.setStartBlock((blocknum-1)*sch.getBlocksize()+1);

```

## Business logic

### 페이지 처리

1. 페이지 초기값 5로 설정
2. 첫번째 페이지 부터 보여지게 설정
3. 페이지를 넘길때마다 시작번호와 마지막 번호가 페이지 사이즈 만큼 증가

## ≡ PROJECT



부장 노태우

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

인적자원관리

## ■■■ 게시판 등록

프로젝트 번호: 1001

제목

제목입력하세요

내용

내용입력하세요

첨부 파일

파일을 선택하세요

제작

개인정보

참여인원

Logout

등록

조회화면으로

프로젝트 번호: 1001

제 목  
제목 입력하세요

내 용  
내용 입력하세요

첨부 파일  
파일을 선택하세요

Browse

등록      조회화면으로

## mapper

```
<insert id="insert" parameterType="risk">
    INSERT INTO PMSISSUE VALUES(pmsissue_seq.nextval,NVL(${title},'내용없음'),
    NVL(${detail},'내용없음'),0,null,sysdate,null,null,#{pno},#{mno})
</insert>
```

## view

```
<c:choose >
    <c:when test="${infor_M.pno ne 0}">
        <input name="pno" type="hidden" value="${infor_M.pno}" />
    </c:when>
    <c:otherwise>
        <div id="pm">
            <div style="float:left;">
                <span class="input-group-text" style="height:34px">프로젝트 번호:</span>
            </div>
            <div >
                <select class="form-control" style="width:100px;float:left; " name="pno">
                    <c:forEach items="${pnolist}" var="pno" varStatus="sts" >
                        <option value="${pno }" ${pno}></option>
                    </c:forEach>
                </select>
            </div>
        </div>
    </c:otherwise>
</c:choose>
```

## Business logic

- 제목과 내용을 입력하고 등록 버튼을 누르면 본인이 속한 프로젝트 게시판에 내용이 등록된다.
- Admin일 경우 프로젝트 번호를 선택하여 등록할 수 있다.

### - View

Admin일 경우에만 프로젝트 번호 선택이 가능하도록 처리한 모습이다.

프로젝트 번호: 1001

제 목

제목 입력하세요

내 용

내용 입력하세요

첨부 파일

파일을 선택하세요

Browse

등록 조회화면으로

## service

```
private void upload(MultipartFile mtf) {
    String fileName=mtf.getOriginalFilename();
    if(fileName!=null&&!fileName.equals("")) {
        File tmpFile = new File(tmpUpload+fileName);
        if(tmpFile.exists()) tmpFile.delete();
        try {
            mtf.transferTo(tmpFile);
            File orgFile = new File(upload+fileName);
            Files.copy(tmpFile.toPath(), orgFile.toPath(),
                StandardCopyOption.REPLACE_EXISTING);
        }
    }
}
```

## Business logic

- 파일을 선택한 후 등록을 하면
- 지정한 폴더에 파일이 저장되며
- 데이터베이스에 파일이름이 저장된다.

### - Service

첨부 파일의 유무에 따라 중복처리, 변경처리를 하는 부분이다.

 PROJECT

부장 노태우

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

인적자원관리

작성자

홍준표

제 품

결제화면

작성 일

2020. 5. 27 오전 12:00:00

수정일

내 용

결제 화면에서 쿠폰 사용 시 총 결제금액 산출의 업데이트 문제 팝업창과 부모창 간의 실시간 업데이트시 부모창 업데이트 후 자식창 업데이트 하면 변경값 반영이 안됨

첨부 파일

변경할려면 파일을 선택하세요!

해결방안 작성일

해결방안

내용입력하세요

댓글창 열기

작성자	
홍준표	
제 목	
결제화면	
작성일	2020. 5. 27 오전 12:00:00
수정일	
내 용	결제 화면에서 쿠폰 사용 시 총 결제금액 산출의 업데이트 문제 팝업창과 부모창 간의 실시간 업데이트시 부모창 업데이트 후 자식창 업데이트 하면 변경값 반영이 안됨

## mapper

```
<select id="getRisk"
       resultType="risk" parameterType="int">
    SELECT *
    FROM PMISSUE p, PMSEMP pm
    where 1=1
    and p.mno = pm.ENO
    and ino = #{ino}
</select>
```

## view

```
<div class="input-group mb-3">
    <div class="input-group-prepend">
        <span class="bg-success text-white input-group-text">작성자</span>
    </div>
    <input class="form-control" readonly="readonly" style="background-color: #ffffff;" value="${risk.name }" placeholder="작성자입력하세요"/>
</div>
<div class="input-group mb-3">
    <div class="input-group-prepend">
        <span class="bg-success text-white input-group-text">제 목</span>
    </div>
    <input name="title" class="form-control" id="title" value="${risk.title}" placeholder="제목입력하세요" />
```

## Business logic

- 이슈리스트의 상세내용 확인
- 본인이 작성자라면 제목, 내용 수정가능
- 수정 시 수정일 현재날짜로 업데이트
- Admin일 경우에만 본인이 작성하지 않은 글 수정 가능

첨부 파일

pie.png

변경할려면 파일을 선택하세요!

Browse

## controller

```
@RequestMapping(params="method=download")
public String download(@RequestParam("fname") String fname, Model d) {
    // 탑재할 모델명은 파일명으로 설정..
    System.out.println("다운로드할 파일명: "+fname);
    d.addAttribute("downloadFile", fname);
    return "download"; // 컨테이너에서 선언한 viewer명..
}
```

## DownloadViewer

```
@Value("${upload}")
private String upload;

@Override
protected void renderMergedOutputModel(Map<String, Object> model,
    HttpServletRequest request, HttpServletResponse response)
throws Exception {

    String fname = (String)model.get("downloadFile");
    File file = new File(upload+fname);
    response.setContentType("application/download; charset=UTF-8");
    response.setContentLength((int)file.length());
    fname = URLEncoder.encode(fname, "utf-8").replaceAll("\\+", "");
}
```

## Business logic

- 등록 페이지에서 첨부한 파일 이름 확인 가능
- 파일명 클릭 시 다운로드 가능
- 변경할 파일을 선택하여 첨부파일 변경 가능
- 파일변경은 작성자와 Admin만 가능

### - DownloadViewer

Controller에서 전달되어온 파일명을 객체로 할당하여 다운로드 처리를 한다.

해결방안 작성일

해결방안

내용입력하세요

댓글창 열기

수정

해결방안 저장

뒤로가기

## controller

```
HttpSession session = request.getSession();
PmsMember emp =(PmsMember)session.getAttribute("infor_M");
if(emp.getMdiv()==4) {
    d.addAttribute("ri", 3);
} else if(emp.getMdiv()==6) {
    d.addAttribute("ri", 4);
} else {
    d.addAttribute("ri", 5);
}
```

## Business logic

1. 해당 프로젝트의 PM과 Admin만 작성 가능
2. 수정, 해결방안 저장 버튼은 권한이 있을 경우에만 보인다.

- 권한처리

Controller에서 session으로 각각의 권한 값 (mdiv)을 받아 View단으로 넘겨주었다.

PROJECT

Browse

해결방안 작성일

내용입력하세요

댓글

내용입력하세요

수정

해결방안 저장

뒤로가기

입력

총준표	쿠폰 코드의번호를 획득 한뒤 새로운 쿠폰을 등록해야 할거 같습니다.	수정	댓글	2020. 6. 1 오전 11:18:38
총준표	↳ 그뒤 제약조건을 삭제하고 다시 생성한 뒤 일반회원 리스트 작성해야 할것 같습니다.	수정	댓글	2020. 6. 1 오전 11:18:38
정은경	↳ 정말 감사합니다	수정	댓글	2020. 6. 1 오전 11:18:38
김무성	쿠폰부여의 DAO 메서드를 각각 작성하여 Service에서 일괄처리해보세요	수정	댓글	2020. 6. 1 오전 11:18:38
정은경	해결되었습니다. 도와주셔서 감사합니다.	수정	댓글	2020. 6. 1 오전 11:18:38

© Copyrights Dashio. All Rights Reserved  
Created with Dashio template by TemplateMag

댓글

내용입력하세요

입력

홍준표 쿠폰 코드의번호를 획득 한뒤 새로운 쿠폰을 등록해야 할거 같습니다.

수정 댓글 2020. 6. 1 오전 11:18:38

**controller**

```
// 댓글입력
@RequestMapping(params="method=cinsert")
public String cinsert(Comment insert, Model d) {
    service.cinsert(insert);
    d.addAttribute("chkreply", 1);

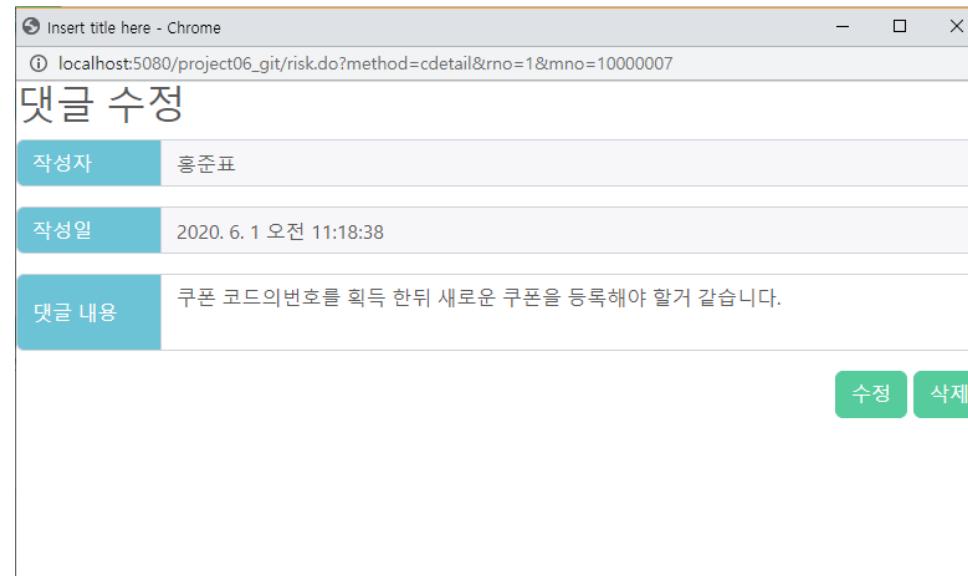
    return "forward:/risk.do?method=comment";
}
```

**view**

```
var method = "${chkreply}";
if(method=="1"){
    $(location).attr("href", "${path}/risk.do?method=comment&ino="+ino);
}
```

**Business logic**

- 상세 페이지에서 댓글창 열기 버튼을 클릭 시 확인할 수 있다.
- 댓글을 입력하면 입력된 순서대로 나열된다.
- 작성자와 내용 작성시간이 표시되며
- 수정과 삭제를 할 수 있는 수정버튼과 대댓글을 달 수 있는 댓글 버튼이 있다
  - 문제해결
    - 댓글 내용 입력 후 페이지가 새로고침 될 때마다 입력했던 내용이 추가로 입력되는 문제 발생
    - Controller에서 forward를 리턴하여 요청정보가 그대로 유지되는 것이 원인
    - 댓글입력 후 method=comment(댓글 리스트)로 가게하여 추가 입력을 방지



## view

```
var mno = "${param.mno}"; // 로그인한 사용번호  
var cmno = "${comment.mno}"; // 작성자 사용번호  
var md = "${infor_M.mdv}"  
if(md!=6){  
    if(mno!=cmno){  
        alert("수정권한이 없습니다.");  
        window.close();  
    }  
}
```

## Business logic

- 수정 버튼을 클릭 시 나오는 팝업창이다.
- 글의 작성자와 Admin만 사용할 수 있다.
- 댓글의 수정과 삭제를 할 수 있다.
- 수정, 삭제 시 팝업창이 자동으로 닫히며 수정, 삭제한 내용이 댓글에 적용이 된다.

홍준표	쿠폰 코드의 번호를 획득 한뒤 새로운 쿠폰을 등록해야 할거 같습니다.	<button>수정</button>	<button>댓글</button>	2020. 6. 1 오전 11:18:38
홍준표	↳ 그뒤 제약조건을 삭제하고 다시 생성한 뒤 일반회원 리스트 작성해야 할것 같습니다.	<button>수정</button>	<button>댓글</button>	2020. 6. 1 오전 11:18:38
정은경	↳ 정말 감사합니다	<button>수정</button>	<button>댓글</button>	2020. 6. 1 오전 11:18:38
김무성	쿠폰부여의 DAO 메서드를 각각 작성하여 Service에서 일괄처리해보세요	<button>수정</button>	<button>댓글</button>	2020. 6. 1 오전 11:18:38
정은경	해결되었습니다. 도와주셔서 감사합니다.	<button>수정</button>	<button>댓글</button>	2020. 6. 1 오전 11:18:38

## mapper

```
<select id="clist" resultMap="commentResult" parameterType="rsch" >
SELECT level,re.* ,e.*
FROM PMSREPLY re,PMSEMP e
WHERE re.MNO = e.ENO
and ino = #{ino}
START WITH refno =0
CONNECT BY PRIOR rno = refno
ORDER siblings BY rno
</select>
```

## view

```
<!-- 댓글 들여쓰기 -->
<td class="text-left" >
    <c:forEach varStatus="sts"
        begin="2" end="${comment.level}">&ampnbsp&ampnbsp&ampnbsp&ampnbsp
        <c:if test="${comment.level>1 and sts.last}">↳</c:if>
    </c:forEach>${comment.detail}
</td>
```

## Business logic

- 댓글 버튼을 클릭 시 나오는 팝업창이다.
- 대댓글을 달수 있다.
- 내용을 입력하면 팝업창이 자동으로 닫힌다.
- 입력한 내용은 선택한 댓글 아래 추가되며 화살표와 일정간격 오른쪽으로 이동한 상태로 출력된다.

### - 들여쓰기

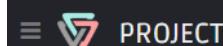
Mapper에 level을 추가하여 View에서 Level가 높을수록 왼쪽에 여백이 생기도록 하였다.



화면구현

# 프로젝트 일정관리

전현규



대표이사 전현규



대시보드



인적자원관리

## &gt; 프로젝트 추가

프로젝트명

시작일

Select date

종료일

Select date

PM

선택

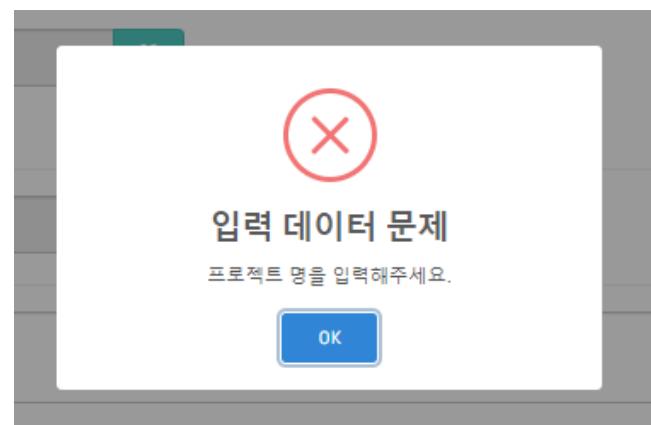
프로젝트 개요

등록

뒤로가기

프로젝트명	<input type="text"/>
시작일	<input type="date"/>  Select date
종료일	<input type="date"/>  Select date
PM	<input type="button" value="선택"/>
프로젝트 개요	<input type="text"/>

[등록](#) [뒤로가기](#)



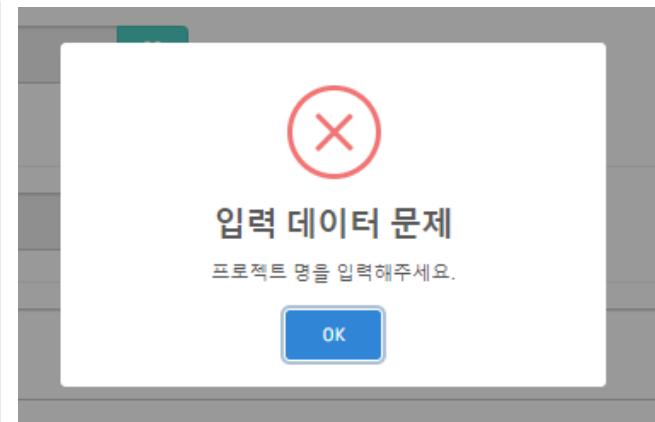
## view

```
if($(".name=pro_pname").val()=='' ||
   $(".name=pro_pname").val()==null){
  Swal.fire(
    '입력 데이터 문제',
    '프로젝트 명을 입력해주세요.',
    'error'
  )
}
```

## Business logic

1. 프로젝트 추가 입력란에 정보를 입력하여 등록
  - 프로젝트명, 개요는 텍스트 형식
  - 시작일, 종료일은 Datepicker를 이용한 Calendar 형식
  - PM은 별도의 창을 통해서 입력받는 형식
2. 작성 조건에 어긋날 경우 등록 불가능
  - 모든 입력부분에 공란이 있어서는 안됨
  - 시작일, 종료일은 오늘 날짜보다 이후여야 함
  - 종료일은 시작일보다 이후여야 함

프로젝트명	<input type="text"/>
시작일	<input type="text"/> Select date
종료일	<input type="text"/> Select date
PM	<input type="text"/> 선택
프로젝트 개요	<input type="text"/>



## controller

```
@RequestMapping(params="method=proins")
Project insp = new Project();
pmsemp inspe = new pmsemp();
Project insertedp = new Project();
inspe = peservice.getPmsemp(mno);
inspe.setPno(0);
peservice.updatePm(inspe);
prservice.proIns(insp);
insertedp = prservice.getProjectformno(mno);
inspe.setPno(insertedp.getPno());
peservice.updatePm(inspe);

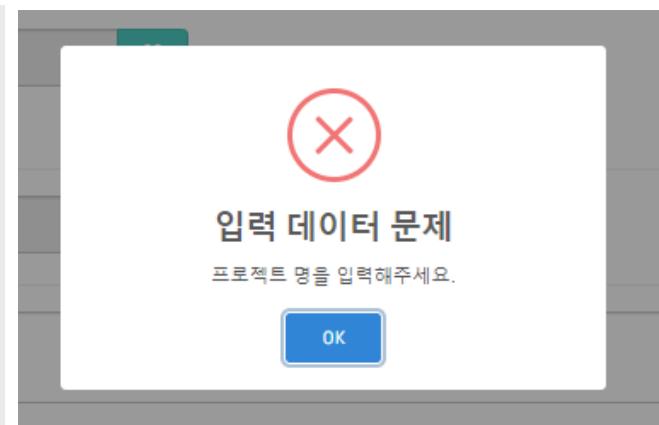
return "forward:/DashTeam.do?method=list";
```

## Business logic

- 등록버튼을 누르면 method=proins가 되어 등록 수행
- insp : 입력한 프로젝트 정보가 저장되어 DB에 추가할 객체  
inspe : 프로젝트에 선택한 PM의 사원정보를 불러올 객체  
insertedp : 새롭게 추가된 프로젝트의 정보를 불러올 객체
- PM으로 선택된 사원의 권한 변경(제약조건 위반 방지)
- 새로 추가된 프로젝트 정보 DB에 추가(INSERT)
- 추가된 프로젝트의 넘버를 가져와 PM으로 선택된 사원에게 배정
- 1~5의 과정을 수행 후 DashTeam.do?method=list로 Forward

프로젝트명	<input type="text"/>
시작일	<input type="text"/> Select date
종료일	<input type="text"/> Select date
PM	<input type="text"/> 선택
프로젝트 개요	<input type="text"/>

[등록](#) [뒤로가기](#)



## service

```

public void updatePm(pmsemp updatePm) {
    int isMem = rep.memCheck(updatePm);
    if(isMem==0) {
        if(updatePm.getPno()==0) rep.insertMem3(updatePm);
    }else{
        if(updatePm.getPno()==0) rep.updatePm1_1(updatePm);
        else rep.updatePm1(updatePm);
    }
    rep.updatePm4(updatePm);
}

public void proIns(Project ins) {
    dao.proIns(ins);
}

```

## Business logic

- 첫번째 updatePm을 통해 PM으로 선택한 사원에게 권한을 부여
- memCheck를 통해 선택한 사원이 PMS시스템에 가입되어 있는지를 확인
  - 가입 되어있지 않다면 insertMem3을 통해 PMS시스템에 가입
  - 가입 되어있다면 updatePm1\_1을 통해 PM권한만을 부여
- proIns를 통해 새로운 프로젝트 정보를 등록
- 두번째 updatePm을 통해 PM으로 선택한 사원에 프로젝트 배정
  - updatePm1를 통해 PM에게 새로운 프로젝트를 배정
  - updatePm4를 통해 새로운 프로젝트의 PM을 선택한 사원으로 설정



SELECT PM

Search

이름	직책	희망 연락처	상태
박근혜	전무	010-6634-1032	<span style="background-color: #00aaff; color: white; border: 1px solid #00aaff; border-radius: 5px; padding: 2px 10px; font-weight: bold;">선택</span>
전두환	차장	010-7714-8058	<span style="background-color: #00aaff; color: white; border: 1px solid #00aaff; border-radius: 5px; padding: 2px 10px; font-weight: bold;">선택</span>
최규하	과장	010-6078-9450	<span style="background-color: #00aaff; color: white; border: 1px solid #00aaff; border-radius: 5px; padding: 2px 10px; font-weight: bold;">선택</span>
박정희	대리	010-9665-6200	<span style="background-color: #00aaff; color: white; border: 1px solid #00aaff; border-radius: 5px; padding: 2px 10px; font-weight: bold;">선택</span>
김문수	대리	010-5206-7975	<span style="background-color: #00aaff; color: white; border: 1px solid #00aaff; border-radius: 5px; padding: 2px 10px; font-weight: bold;">선택</span>

Previous
1
2
Next

## view

```

function selectpm() {
    window.open("${path}/project.do?method=selectpm&mdiv='"+mdiv,
               "SelectPM", "width=780, height=520, left=300, top=100");
}

function selpm(no){
    var eno = $('#pm_eno'+no).val();
    var ename = $('#pmname_grade'+no).val();
    $('[name=pro_mno]',opener.document).val(eno);
    $('[name=pro_mname]',opener.document).val(ename);
    self.close();
}

```

## Business logic

- 프로젝트추가 페이지의 PM부분 '선택'버튼을 누르면 팝업창으로 PM 선택창(project.do?method=selectpm)을 호출
- PM 선택창에서 선택을 누르면 selpm(no)를 통해 내가 선택한 사원정보가 프로젝트추가 폐에지의 PM부분으로 입력

### SELECT PM

Search

이름	직책	희망 연락처	상태
박근혜	전무	010-6634-1032	<span style="background-color: #00aaff; color: white; border-radius: 5px; padding: 2px 10px; border: none; font-weight: bold;">선택</span>
전두환	차장	010-7714-8058	<span style="background-color: #00aaff; color: white; border-radius: 5px; padding: 2px 10px; border: none; font-weight: bold;">선택</span>
최규하	과장	010-6078-9450	<span style="background-color: #00aaff; color: white; border-radius: 5px; padding: 2px 10px; border: none; font-weight: bold;">선택</span>
박정희	대리	010-9665-6200	<span style="background-color: #00aaff; color: white; border-radius: 5px; padding: 2px 10px; border: none; font-weight: bold;">선택</span>
김문수	대리	010-5206-7975	<span style="background-color: #00aaff; color: white; border-radius: 5px; padding: 2px 10px; border: none; font-weight: bold;">선택</span>

Previous
1
2
Next

## controller

```

@RequestMapping(params="method=selectpm")
public String logList(@ModelAttribute("pmsempSch") pmsempSch sch,
    Model m, @RequestParam("mdiv") int mdiv) {
    sch.setPno(mdiv);
    //프로젝트에 참여하지 않은 대리급 이상 사원 호출
    m.addAttribute("loglist", peservice.selectpm(sch));
    return "WEB-INF\\views\\main\\selectpm.jsp";
}

```

## Business logic

- 팝업창이 열리면 프로젝트에 참여하지 않은 사원의 리스트를 peservice.selectpm(sch)로 불러와 selectpm.jsp 페이지에 뿌려준다.

PROJECT PMS시스템 개발

제팅 개인정보 참여인원 Logout

부장 김대중

간트차트

PMS시스템 개발

작업명	담당자	소요일	시작일	종료일	상태	남은일수	May				June	
							Week 19	Week 20	Week 21	Week 22	Week 23	
총괄진행	김대중	36일	20/05/04	20/06/08	-	-	100%					
1. 화면설계	김대중	4일	20/05/04	20/05/08	완료	-	100%					
2. 데이터베이스	김대중	10일	20/05/04	20/05/14	완료	-	100%					
2.1. 데이터베이스 설계	김대중	5일	20/05/04	20/05/09	완료	-	100%					
2.2. 데이터베이스 생성	김대중	5일	20/05/09	20/05/14	완료	-	100%					
3. 화면구현	김대중	24일	20/05/08	20/06/01	진행 중	3일 초과	100%					
3.1. 사설정보	김대중	24일	20/05/08	20/06/01	진행 중	3일 초과	50%					
3.1.1. 로그인/로그아웃(웹)	추미애	4일	20/05/08	20/05/12	완료	-	100%					
3.1.2. 번호 찾기(웹)	추미애	4일	20/05/12	20/05/16	완료	-	100%					
3.1.3. 개인정보(웹)	추미애	4일	20/05/16	20/05/20	완료	-	100%					
3.1.4. 로그인/로그아웃(앱)	추미애	4일	20/05/20	20/05/24	진행 중	11일 초과	10%					
3.1.5. 번호 찾기(앱)	추미애	4일	20/05/24	20/05/28	진행 중	7일 초과	10%					
3.1.6. 개인정보(앱)	추미애	4일	20/05/28	20/06/01	진행 중	3일 초과	100%					
3.2. 프로젝트 총괄관리	김대중	24일	20/05/08	20/06/01	진행 중	3일 초과	80%				20%	
3.3. GTO 담당(앱)	김대중	3일	20/05/08	20/05/11	진행 중	-	100%					

PROJECT PMS시스템 개발
팀
개인정보
참여인원
Logout



### 과장 오세훈

- 대시보드
- 업무관리
- 일정관리
- 공지사항
- 회의록
- 이슈리스트

#### 간트차트

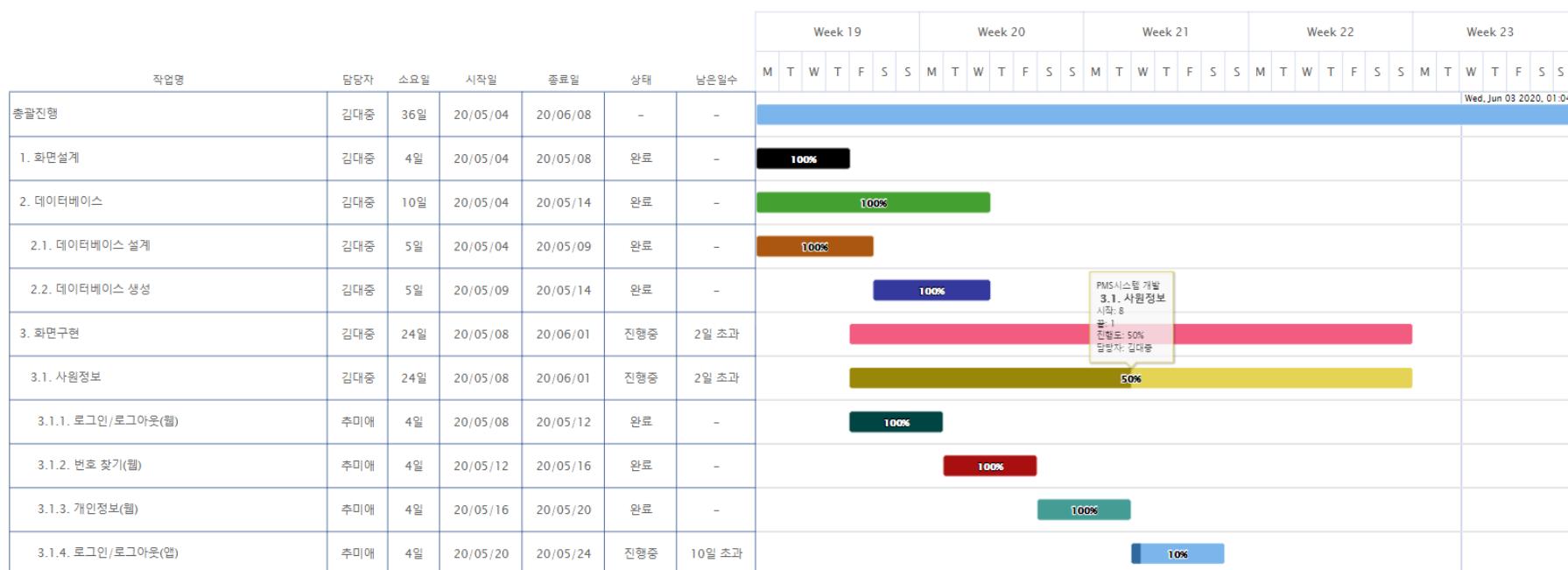
PMS시스템 개발

작업명	담당자	소요일	시작일	종료일	상태	남은일수	Week 19							Week 20							Week 21							Week 22							Week 23						
							M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
총괄진행	김대중	36일	20/05/04	20/06/08	-	-	Thu, Jun 04 2020, 03																																		
3.2. 프로젝트 총괄관리	김대중	24일	20/05/08	20/06/01	진행중	3일 초과								80%																											
3.2.1. CTO 선정(웹)	오세훈	7일	20/05/08	20/05/15	완료	-								100%																											
3.2.5. CTO 선정(앱)	오세훈	7일	20/05/08	20/05/15	완료	-								100%																											
3.3. 프로젝트 통합관리	김대중	24일	20/05/08	20/06/01	진행중	3일 초과																																			
3.3.2. PM 선정(웹)	오세훈	8일	20/05/15	20/05/23	결제대기	12일 초과																																			
3.3.5. PM 선정(앱)	오세훈	8일	20/05/15	20/05/23	결제대기	12일 초과																																			
3.5. 프로젝트 자원관리	김대중	24일	20/05/08	20/06/01	진행중	3일 초과																																			
3.5.1. 인적자원관리 (웹)	오세훈	8일	20/05/24	20/06/01	진행중	3일 초과																																			
3.5.2. 인적자원관리 (앱)	오세훈	8일	20/05/24	20/06/01	진행중	3일 초과																																			

PMS시스템 개발  
3.5. 프로젝트 자원관리  
시작: 8  
종료: 1  
진행도: 0%  
담당자: 김대중

Highcharts.com

## PMS시스템 개발



## view

```

$.ajax({
    type: "post",
    url: "${path}/chart.do?method=ajaxdata",
    data: $( "form" ).serialize(),
    dataType: "json",
    success: function(data){
        Highcharts.ganttChart('chart', cInfo);
    }
    error: function(err){
        console.log("ajax처리 에러");
        console.log(err);
    }
});

```

## Business logic

- 일정관리 메뉴를 통해 차트 페이지로 이동하면 본인의 회원정보를 조회한다.
- 회원등급(관리자, PM, 팀원)과 참여프로젝트를 조회해 Ajax로 JSON 데이터를 받아와 차트 API에 입력하여 간트차트를 출력한다.

## PMS시스템 개발



## controller

```

@RequestMapping(params="method=chart")
public String chartpm(HttpServletRequest request, Model d) {
    HttpSession session = request.getSession();
    if(session.getAttribute("mno")==null) {
        return "WEB-INF\\views\\main\\login.jsp";
    }
    PmsMember emp =(PmsMember)session.getAttribute("infor_M");
    d.addAttribute("project", pservice.getProject(emp.getPno()));

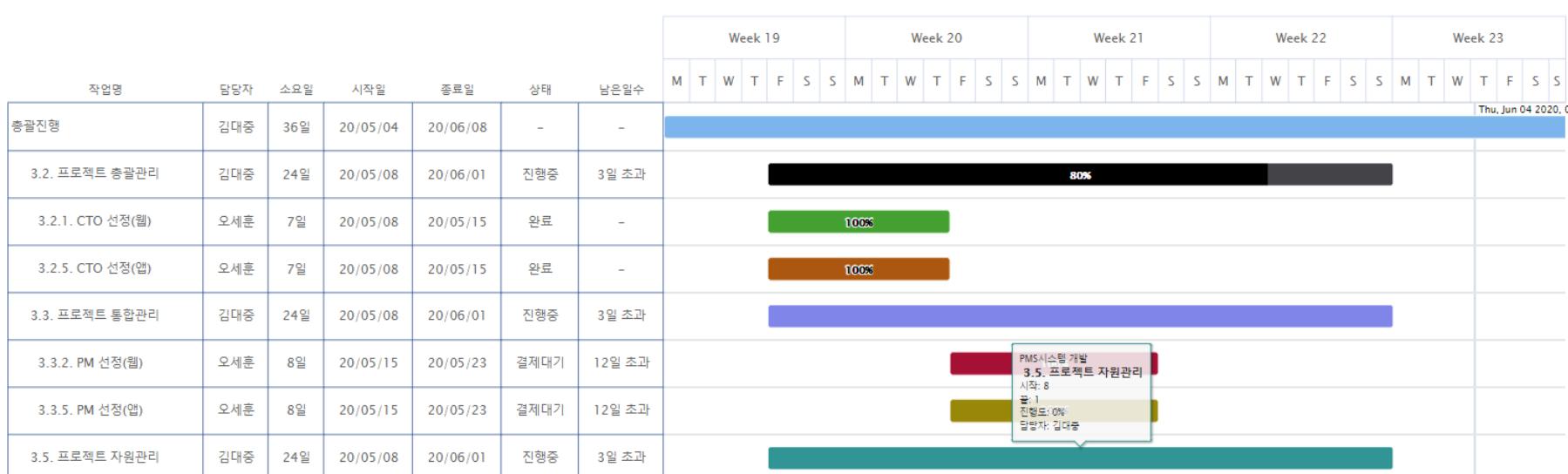
    return "WEB-INF\\views\\main\\chart.jsp";
}
@RequestMapping(params="method=ajaxdata")
public String chtData(HttpServletRequest request,
    Model d) {
    HttpSession session = request.getSession();
    PmsMember emp =
    (PmsMember)session.getAttribute("infor_M");
    if(emp.getMdiv()==5)
        d.addAttribute("chartlist",
            cservice.chartListTm(emp.getPno(), emp.getMno()));
    else
        d.addAttribute("chartlist",
            cservice.chartList(emp.getPno()));
    return "pageJsonReport";
}

```

## Business logic

- 일정관리 메뉴를 클릭하면 method=chart가 실행되어 로그인 정보를 바탕으로 자신이 참여중인 프로젝트 정보를 모델에 추가 후 chart.jsp로 이동한다.
- chart.jsp의 Ajax를 통해 method=ajaxdata가 실행되어 로그인 정보를 바탕으로 PM의 경우 프로젝트의 전체 업무 정보를 모델에 추가한다.(PM의 mdiv = 4)
- pageJsonReport를 이용하여 모델에 추가한 객체정보를 JSON 형식으로 변환한다.

## PMS시스템 개발



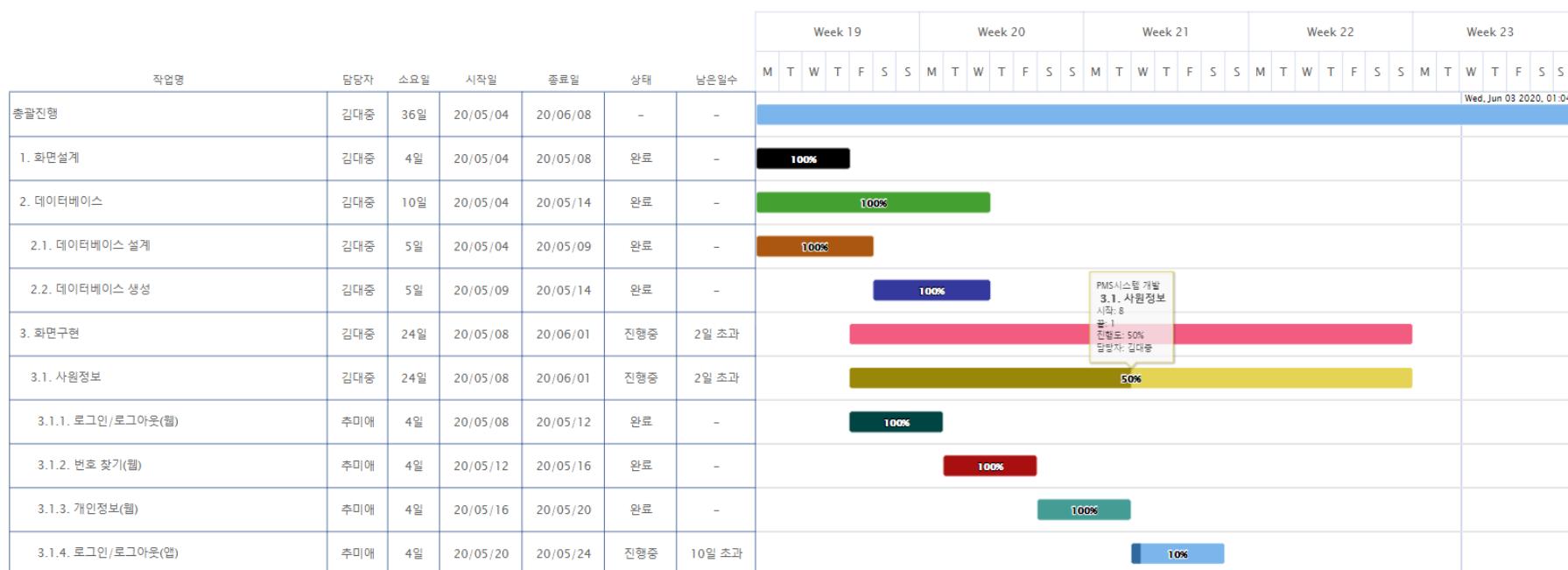
## controller

```
@RequestMapping(params="method=ajaxdata")
public String chtData(HttpServletRequest request,
    Model d) {
    HttpSession session = request.getSession();
    PmsMember emp =
    (PmsMember)session.getAttribute("infor_M");
    if(emp.getMdiv()==5)
        d.addAttribute("chartlist",
            cservice.chartListTm(emp.getPno(), emp.getMno()));
    else
        d.addAttribute("chartlist",
            cservice.chartList(emp.getPno()));
    return "pageJsonReport";
}
```

## Business logic

- 팀원의 경우 프로젝트에서 자신이 담당하고 있는 업무 정보를 모델에 추가한다.(팀원의 mdiv = 5)
- pageJsonReport를 이용하여 모델에 추가한 객체정보를 JSON 형식으로 변환한다.

## PMS시스템 개발



## service

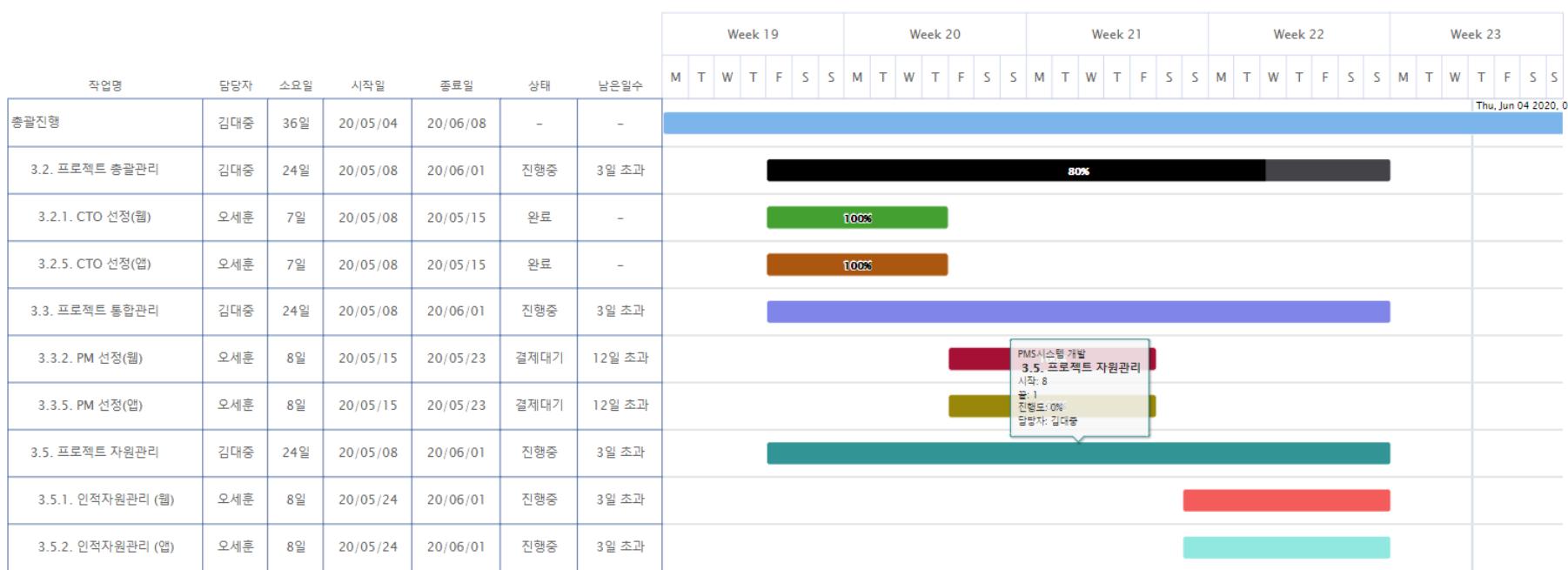
```
public Project getProject(int pno){
    return dao.getProject(pno);
}
```

```
public ArrayList<Chart> chartList(int pno){
    return dao.chartList(pno);
}
```

## Business logic

1. getProject를 통해 현재 참여중인 프로젝트 정보를 호출한다.
2. chartList를 통해 참여중인 프로젝트의 전체 업무를 호출한다.

## PMS시스템 개발



## service

```

public ArrayList<Chart> chartListTm(int pno, int mno){
    ArrayList<Chart> clisttmfinal = new ArrayList<Chart>();
    ArrayList<Chart> clisttm = new ArrayList<Chart>();
    ArrayList<Integer> tmRefno = dao.getRefno(mno);
    for(int refno : tmRefno) {
        clisttm = dao.chartListttm(pno, mno, refno);
        for(Chart chttm : clisttm) {
            clisttmfinal.add(chttm);
        }
    }
    return clisttmfinal;
}

```

## Business logic

- clisttmfinal : 화면에 출력될 팀원 업무가 저장될 Chart객체  
clisttm : clisttmfinal에 순차적으로 데이터를 전달할 Chart객체  
tmRefno : 각 팀원 업무의 상위업무 리스트가 저장될 객체
- getRefno를 통해 자신의 상위업무 리스트를 호출하고 상위업무, 프로젝트번호, 사원번호를 통해 호출한 업무를 clisttm에 담아 clisttmfinal에 전달하는 과정을 for문으로 반복한다.
- 일련의 과정으로 모든 업무를 저장한 clisttmfinal을 반환한다.

작업명	담당자	소요일	시작일	종료일	상태	남은일수
총괄진행	김대중	36일	20/05/04	20/06/08	-	-
1. 화면설계	김대중	4일	20/05/04	20/05/08	완료	-
2. 데이터베이스	김대중	10일	20/05/04	20/05/14	완료	-
2.1. 데이터베이스 설계	김대중	5일	20/05/04	20/05/09	완료	-
2.2. 데이터베이스 생성	김대중	5일	20/05/09	20/05/14	완료	-
3. 화면구현	김대중	24일	20/05/08	20/06/01	진행중	2일 초과
3.1. 사원정보	김대중	24일	20/05/08	20/06/01	진행중	2일 초과
3.1.1. 로그인/로그아웃(웹)	추미애	4일	20/05/08	20/05/12	완료	-
3.1.2. 번호 찾기(웹)	추미애	4일	20/05/12	20/05/16	완료	-
3.1.3. 개인정보(웹)	추미애	4일	20/05/16	20/05/20	완료	-
3.1.4. 로그인/로그아웃(앱)	추미애	4일	20/05/20	20/05/24	진행중	10일 초과

	ABC TNAME	▼	123 TNO	▼
1	1. 화면설계		1,001	
2	2. 데이터베이스		1,002	
3	2.1. 데이터베이스 설계		1,005	
4	2.2. 데이터베이스 생성		1,006	
5	3. 화면구현		1,003	
6	3.1. 사원정보		1,007	
7	3.1.1. 로그인/로그아웃(웹)		1,014	
8	3.1.2. 번호 찾기(웹)		1,015	
9	3.1.3. 개인정보(웹)		1,016	
10	3.1.4. 로그인/로그아웃(앱)		1,017	

## mapper

```
(SELECT LTRIM(SYS_CONNECT_BY_PATH(r, '.'),'.')||'.' || tname tname, tno
FROM (SELECT rank() over (partition BY refno ORDER BY tno) r,
tno, tname, refno FROM pmstask WHERE pno = #{pno})
CONNECT BY PRIOR tno = refno
START WITH refno =0) prt
```

## Business logic

- rank() over (partition BY 상위업무번호 ORDER BY 업무번호) 를 이용해서 상위업무번호 내의 업무순서에 따른 RANK를 매긴다.
- LTRIM(SYS\_CONNECT\_BY\_PATH(r,'.'),'.')||'.'을 이용해서 각 업무에 상위업무(1).하위업무(2).하위업무(3)순으로 번호를 매긴다.



화면구현

# 프로젝트 통합관리

이연학



부장 노태우

 대시보드 업무관리 일정관리 공지사항 회의록 이슈리스트 인적자원관리 업무현황

## 업무 진행률 기준

- 요구사항분석 10%
- 화면구현 20%
- 기능구현 30%
- 데이터베이스연동 30%
- 테스트 10%

검색

페이지 크기: 5

프로젝트번호	번호	업무 명	종료일	결재여부
1001	1005	데이터베이스 설계	2020-05-09 00:00:00	결재완료
1001	1004	테스트	2020-06-08 00:00:00	결재완료
1001	1003	화면구현	2020-06-01 00:00:00	결재완료
1001	1002	데이터베이스	2020-05-14 00:00:00	결재완료
1001	1001	화면설계	2020-05-08 00:00:00	결재완료

Previous 1 2 3 4 5 Next

✓ 업무등록

업무현황

검색

페이지 크기: 5

업무 진행률 기준

- 요구사항분석 10%
- 화면구현 20%
- 기능구현 30%
- 데이터베이스연동 30%
- 테스트 10%

프로젝트번호	번호	업무 명	종료일	결재여부
1001	1005	데이터베이스 설계	2020-05-09 00:00:00	결재완료
1001	1004	테스트	2020-06-08 00:00:00	결재완료
1001	1003	화면구현	2020-06-01 00:00:00	결재완료
1001	1002	데이터베이스	2020-05-14 00:00:00	결재완료
1001	1001	화면설계	2020-05-08 00:00:00	결재완료

Previous 1 2 3 4 5 Next

✓ 업무등록

## view

### 5번 logic

```
if(mdiv != 4 && mdiv != 6){
    alert("권한이 없습니다.");
} else{
    $(location).attr("href","${path}/task.do?method=insForm");
}
```

## mapper

### 1번 logic

```
AND pm.pno = #{pno}
<if test="mdiv!=2 and mdiv!=3 and mdiv!=4 and mdiv!=6">
AND pm.mno = #{mno}
</if>
```

## Business logic

- 로그인한 계정이 담당한 업무만 출력
- 업무 개수에 따라 페이지 블록 증감
- 업무 명으로 검색
- 업무 클릭 시 해당 업무 상세화면으로 이동
- 업무 등록은 PM / 관리자만 가능

≡ V PROJECT

부장 노태우

업무현황

업무 진행률을 기준

- 요구사항분석 10%
- 화면구현 20%
- 기능구현 30%
- 데이터베이스연동 30%
- 테스트 10%

프로젝트번호  
1002

글번호  
1054

담당자  
김영삼 부장

담당자 변경 ▾

상위업무명  
화면구현

업무명  
관리자 시스템

시작일  
2020-05-08 00:00:00

마감일  
2020-06-01 00:00:00

업무 내용  
관리자 시스템 개발

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

인적자원관리

채팅

개인정보

참여인원

LogOut

≡ V PROJECT

부장 노태우

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

인적자원관리

2020-05-08 00:00:00

마감일

2020-06-01 00:00:00

업무 내용

관리자 시스템 개발

업무 진행도

내용 입력하세요

저장하기

결재 여부

결재완료

결재신청

반려처리

결재완료

진행률: 100

진행률수정

반려 사용

사유를 입력하세요

사유 등록

수정

삭제

조회 화면으로

**업무현황**

업무 진행률 기준  
 - 요구사항분석 10%  
 - 화면구현 20%  
 - 기능구현 30%  
 - 데이터베이스연동 30%  
 - 테스트 10%

글번호	1054
담당자	김영삼 부장
상위업무명	화면구현
업무명	관리자 시스템
시작일	2020-05-08 00:00:00
마감일	2020-06-01 00:00:00
업무내용	관리자 시스템 개발

**업무 진행도**

내용 입력하세요.

**결재 여부**

결재완료      결재신청      반려처리      결재완료

전통수 100      진행률수정

반려 사유  
사유를 입력하세요.

**저장하기**

**사유 등록**

수정      삭제      조회 화면으로

## Business logic

- 권한 별로 출력되는 화면이 다름 (사진은 관리자 기준)
- 담당자, 업무명, 업무내용 수정가능 (PM)
- 업무 진행도, 업무 진행률 상단의 진행률 기준에 맞춰 저장 / 수정 (담당자)
- 하위 업무는 진행률이 100이면 결재신청가능 (담당자)
- 결재 신청된 업무만 반려, 결재완료 처리가능 (PM)

### view

### 5번 logic

```

$("#divBtn3").click(function(){
  if(tdiv==22){ // 결재 신청 되었을 때
    if(confirm("결재완료 하시겠습니까?")){
      alert("결재완료 되었습니다.");
      $("#deForm").attr("action","${path}/task.do?method=upTdiv3");
      $("#deForm").submit();
    }
  }else if (emno == tmno && tprog == 100 && refno != tno && refno != 0 && mdiv == 4 || emno == tmno && tprog == 100 && refno != tno && refno != 0 && mdiv == 6){
    if(confirm("PM업무 "+tno+" 번 결재 하시겠습니까?")){
      // PM 업무(상위업무)의 진행률이 100%가 됐을 때
      $("#deForm").attr("action","${path}/task.do?method=upTdiv4");
      $("#deForm").submit();
    }
  }else{
    alert("신청 여부 혹은 진행률을 확인 해주세요");
  });
});
  
```

**업무현황**

업무 진행률 기준  
 - 요구사항분석 10%  
 - 화면구현 20%  
 - 기능구현 30%  
 - 데이터베이스연동 30%  
 - 테스트 10%

글번호	1054
담당자	김영삼 부장
상위업무명	화면구현
업무명	관리자 시스템
시작일	2020-05-08 00:00:00
마감일	2020-06-01 00:00:00
업무내용	관리자 시스템 개발
업무진행도	내용 입력하세요.

**결제 여부**

결제완료      결제신청      반려처리      결제완료

전체수 100      진행률수정

반려 사유

사유를 입력하세요.

**저장하기**

**사유 등록**

수정      삭제      조회 화면으로

## Business logic

- 최상위 업무, 상위업무, 하위 업무로 나누어져 있는 업무 프로세스에서 상위 업무들은 진행률 수정이나 결재 신청 불가 / 하위 업무가 결재 완료 처리되면 하위 업무 개수에 따라 상위 업무 진행률이 상승함 하위 업무 5개 있는 상위업무는 하위업무 1개가 결재 완료 처리되면 진행률 20%로 상승 100%가 되면 상위업무도 PM이 결재 완료처리 가능
- 반려 처리 시에만 반려사유 등록가능 (PM)
- 삭제 버튼 클릭 시 해당 업무 삭제 (PM)
- 조회화면으로 버튼 클릭 시 업무리스트로 이동
- 관리자 계정은 모든 기능 사용가능

## mapper logic 1번

```

UPDATE pmstask
SET prog = ROUND((SELECT count(*)
                   FROM PMTASK
                  WHERE refno = #{refno}
                  AND tdiv = 24) * 100 /
                   (SELECT count(*)
                   FROM PMTASK
                  WHERE refno = #{refno}))
WHERE tno = #{tno}
  
```

PROJECT

부장 노태우

대시보드

업무관리

일정관리

공지사항

회의록

이슈리스트

안적자율관리

프로젝트번호  
1001

상위업무번호  
최상위 업무 0

업무명  
입력하세요

시작일

마감일

담당자  
윤보선 차장

업무 내용  
내 맘은 뭄더러뮤더러럼

첨부 파일  
파일을 선택하세요

## Business logic

1. 프로젝트 번호는 로그인한 계정이 맡은 프로젝트로 고정 변경 불가능
2. 상위 업무 / 담당자는 그 프로젝트에 있는 상위업무 / 팀원들만 출력 / 선택
3. 관리자는 모든 프로젝트, 상위 업무 / 팀원 출력 / 선택 가능

### mapper logic 2번 상위업무

```
SELECT pe.eno, pe.name, pe.GRADE, pm.pno, pm.mdiv
FROM pmsmember pm, pmsemp pe
WHERE pm.mdiv = 5
<if test="mdiv!=6">
    AND pm.pno = #{pno}
</if>
AND pe.eno = pm.mNO
```

### view logic 2번 상위업무

```
<select class="form-control" name="refno">
    <option value="0">최상위 업무 0</option>
    <c:forEach items="${refnolist}" var="ref" varStatus="sts" >
        <option value="${ref.refno}">${ref.tname} ${ref.tno}</option>
    </c:forEach>
</select>
```

프로젝트번호  
1002

상위업무번호  
최상위 업무 0

업무명  
입력하세요

시작일

마감일

담당자  
윤보선 차장

업무 내용  
내 맘은 땀더러땀더러땀

첨부파일  
파일을 선택하세요

Browse

등록 조회 화면으로



화면구현

# 프로젝트 총괄/통합관리

김형준

## &gt; 사원 목록

이름

검색

Search

사원번호	이름	직책	부서명	이메일	핸드폰	권한
10000002	박근혜	전무		Park2013@gmail.com	010-6634-1032	PMS등록안됨
10000003	이명박	기술이사		MB2008@naver.com	010-0224-1025	CTO
10000004	노무현	과장	인사과	MH2003@daum.net	010-8586-8894	인사관리자
10000005	김대중	부장	개발1팀	DJ1998@daum.net	010-2175-9239	PM
10000006	김영삼	부장	개발2팀	YS1993@naver.com	010-5762-1972	PM

« 1 2 3 »

**mapper**

```
1. SELECT a.eno, a.name, a.GRADE, a.DEPT, a.EMAIL,
a.PHONE,
(select c.CNAME from pmstmember b, pmstcodes c
where a.eno=b.mno and b.mdiv=c.cno) cname
FROM pmsemp a
WHERE 1=1
ORDER BY a.eno asc)
```

```
2. SELECT a.eno, a.name, a.GRADE, a.DEPT, a.EMAIL,
a.PHONE,
c.CNAME,b.pno
FROM pmsemp a, pmstmember b, pmstcodes c
WHERE b.mdiv=c.cno(+)
AND a.eno = b.mno(+)
AND a.eno = #{eno}
```

**Business logic**

1. PMSEMP, PMSTMEMBER, PMSTCODES 3개의 테이블을 조인하여 사원의 정보와 권한정보를 불러온다.
2. Eno사원번호를 통해 해당 사원을 더블클릭 시 상세 페이지로 이동한다.

## &gt; 사원 목록

이름

검색

Search

사원번호	이름	직책	부서명	이메일	핸드폰	권한
10000002	박근혜	전무		Park2013@gmail.com	010-6634-1032	PMS등록안됨
10000003	이명박	기술이사		MB2008@naver.com	010-0224-1025	CTO
10000004	노무현	과장	인사과	MH2003@daum.net	010-8586-8894	인사관리자
10000005	김대중	부장	개발1팀	DJ1998@daum.net	010-2175-9239	PM
10000006	김영삼	부장	개발2팀	YS1993@naver.com	010-5762-1972	PM

« 1 2 3 »

## view

```
<select class="schdiv" id="selsch">
<option value="name">이름
<option value="grade">직급
<option value="dept">부서
```

## script

```
$("#selsch").change(function(){
  $(".search").attr({"id":this.value,"name":this.value});
```

## view

```
<form><input class="sch-bar search" path="name">
```

<!-- 검색창 -->

```
<input id="name" name="name"
<input id="grade" name="grade"
<input id="dept" name="dept"
```

## Business logic

Select box의 이름, 직급, 부서를 선택하게 되면  
Script jquery로 input검색창의 id와 name의 값이  
Select box의 option의 value값으로 변경된다.  
변경된 value값에 따라 검색이 가능하다.

## &gt; 사원 목록

이름

검색

Search

사원번호	이름	직책	부서명	이메일	핸드폰	권한
10000002	박근혜	전무		Park2013@gmail.com	010-6634-1032	PMS등록안됨
10000003	이명박	기술이사		MB2008@naver.com	010-0224-1025	CTO
10000004	노무현	과장	인사과	MH2003@daum.net	010-8586-8894	인사관리자
10000005	김대중	부장	개발1팀	DJ1998@daum.net	010-2175-9239	PM
10000006	김영삼	부장	개발2팀	YS1993@naver.com	010-5762-1972	PM

« 1 2 3 »

## service

```

sch.setCount(rep.totCnt(sch)); 총 데이터 건수
If(sch.getPageSize()==0) {sch.setPageSize(5); 한번에 보일 데이터}
Sch.setPageCount((int)(Math.ceil
(sch.getCount()/(double)sch.getPageSize()) 총 페이지 수
if(sch.getCurPage()==0)
{sch.setCurPage(1); 현재 클릭한 페이지 초기값
sch.setStart((sch.getCurPage()-1)*sch.getPageSize()+1)
sch.setEnd(sch.getCurPage()*sch.getPageSize());

```

```

Sch.setBlocksize(3); 한번에 보여질 블럭
int blocknum = (int) Math.ceil(sch.getCurPage()
/(double)sch.getBlocksize()); 블럭번호
int endBlock = blocknum*sch.getBlocksize();
sch.setEndBlock(endBlock>sch.getPageCount()
?sch.getPageCount():endBlock); 블럭끝
sch.setStartBlock((blocknum-1)*sch.getBlocksize()+1); 블럭시작

```

## Business logic

- 총 데이터 건수 선언
- 화면에 한번에 보여줄 데이터 건수 초기값 설정
- 총 페이지 수 : 총건수/한번에 보여줄 데이터  
(총건수/페이지크기를 실수로 처리하여 나머지가 있게 한 후 반  
올림 처리 한다.)
- 페이지 블록을 위한 속성값 설정
- 초기 block size 설정(한번에 보일 block크기)
- 시작과 끝나는 블록을 설정

>사원정보 수정

이름	박근혜
부서명	
직책	전무
핸드폰	010-6634-1032
이메일	Park2013@gmail.com
권한	PMS등록 안됨
프로젝트	0 <input type="button" value="1001 ▾"/>

## service

```

if(isMem==0) { //2. PMS등록 확인
//3. 기존 CEO권한 변경
rep.updatepmssCeo2();
rep.updatepmssCeo3();
//4. 새로운 CEO Member등록
rep.updatepmssCeo4(updateCeo);
// 비밀번호 난수 설정
ranNum = makePass("");updateCeo.setPass(ranNum);
rep.insertCeo1(updateCeo);
//5. 메일전송
sendMail(updateCeo.getEno(), ranNum);
}else{
//3.기존 CEO권한 변경
rep.updatepmssCeo2();rep.updatepmssCeo3();
//6.새로운 CEO 권한 변경
rep.updatepmssCeo1(updateCeo)
rep.updatepmssCeo4(updateCeo)
}

```

## Business logic

1. CEO설정 클릭
2. 권한을 설정하는 직원이 PMS에 등록되었는지 확인
3. 기존 CEO권한 해제
4. PMS에 등록되어 있지 않다면 insert로 PMS등록하고 직급 변경
5. 해당 직원 email로 PMS등록 안내 이메일 전송  
사원번호, 비밀번호
6. PMS에 등록되어 있다면 update로 권한과 직급 변경

## &gt;사원정보 수정

이름 박근혜

## 부서명

직책 전무

핸드폰 010-6634-1032

이메일 Park2013@gmail.com

권한 PMS등록안됨

프로젝트 0 1001 ▼

CEO 설정

HR 설정

CTO 설정

PM 설정

이전

**mapper**기존 PM의 프로젝트 정보를 받아옴

&lt;update&gt;

UPDATE pmstask

SET mno=#{eno}WHERE mno=(select p.mno  
from pmsproject p  
where p.pno=#{pno})

&lt;/update&gt;

&lt;update&gt;

UPDATE pmsprojectSET mno=#{eno}WHERE pno=#{pno}

&lt;/update&gt;

**Business logic**

1. PM설정 클릭
2. 권한을 설정하는 직원이 PMS에 등록되었는지 확인
3. 기존 PM권한 해제
4. PMS에 등록되어 있지 않다면 insert로 PMS등록
5. 해당 직원 email로 PMS등록 안내 이메일 전송  
    사원번호, 비밀번호
6. PMS에 등록되어 있다면 update로 권한 변경
7. 기존 PM의 프로젝트 정보를 받아옴

> 팀원 추가							
사원번호	이름	직책	부서명	이메일	핸드폰	권한	선택
10000002	박근혜	전무		Park2013@gmail.com	010-6634-1032	구분없음	<input type="radio"/>
10000008	전두환	차장	개발3팀	Tank1980@daum.net	010-7714-8058	구분없음	<input type="radio"/>
10000009	최규하	과장	개발3팀	Choi1979@naver.com	010-6078-9450	구분없음	<input type="radio"/>
10000010	박정희	대리	개발3팀	Yusin1963@gmail.com	010-9665-6200	구분없음	<input type="radio"/>
10000013	김문수	대리	개발3팀	lmoonsu@daum.net	010-5206-7975	구분없음	<input type="radio"/>
10000014	유승민	사원	개발3팀	seongmin@gmail.com	010-3570-8717	구분없음	<input type="radio"/>

« 1 2 3 »

팀원삭제이동

추가

## mapper

```

SELECT *FROM(
SELECT pmlist.* FROM (
SELECT * FROM (
// PMS등록된 사원 리스트
SELECT a.eno, a.name, a.GRADE, a.DEPT, a.EMAIL, a.PHONE, c.CNAME,b.pno
FROM pmsemp a, pmsmember b, pmcodes c
WHERE a.eno = b.mno and b.mdiv=c.cno AND c.cno =9 AND b.pno is NULL
UNION ALL
// PMS등록 안된 사원 리스트
select a.eno, a.name, a.GRADE, a.DEPT, a.EMAIL, a.PHONE, c.CNAME,b.pno
from (pmsemp a left outer join pmsMember b on a.eno=b.mno), pmcodes c
where b.mno is NULL AND c.cno=9)
ORDER BY eno ASC) pmlist

```

## Business logic

1. PMS에 등록되어 있거나 등록안된 직원 중에 서 권한이 구분없음 이거나 프로젝트번호가 null인 사원을 불러온다.
2. 원하는 직원을 선택하고 추가 버튼 클릭
3. PMS에 등록 여부를 확인
4. PMS에 등록되어 있지 않다면 PMS에 등록
5. PMS에 등록되어 있다면 update로 설정하는 PM과 동일한 PNO를 부여

## &gt; 팀원 삭제

이름

검색

Search

사원번호	이름	직책	부서명	이메일	핸드폰	권한	프로젝트	선택
10000015	추미애	차장	개발1팀	miae58@naver.com	010-2415-2920	팀원	1001	<input type="radio"/>
10000016	오세훈	과장	개발1팀	sehu61@daum.net	010-6002-7753	팀원	1001	<input type="radio"/>
10000017	안희정	대리	개발1팀	heejung65@naver.com	010-9589-3836	팀원	1001	<input type="radio"/>

« 1 2 3 »

팀원추가이동

삭제

## mapper

```
<!-- 팀원삭제 pno==>null로 변경 -->
<update>
UPDATE pmsmember
SET pno=null, mdiv=9
WHERE mno =#{eno}
</update>
```

## controller

로그인한 PM의 프로젝트번호와 동일한 팀원 리스트 불러옴  
PmsMember pmsm=  
(PmsMember)session.getAttribute("infor\_M");  
sch.setPno(pmsm.getpNo());

## Business logic

- 로그인한 PM의 프로젝트번호와 동일한 팀원을 리스트로 불러옴
- 삭제 할 팀원을 선택 후 삭제 버튼 클릭
- 해당 팀원의 프로젝트 번호가 null로 변경
- 해당 팀원의 권한이 구분없음으로 변경

## &gt;사원정보 입력

이름

부서명

직책

핸드폰

이메일

등록

취소

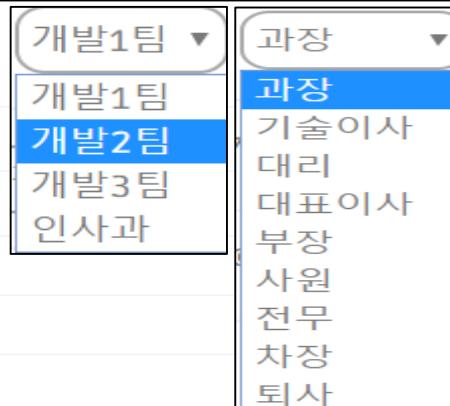
**mapper**

```
<!-- 사원등록 -->
<insert>
insert into pmsemp
values(pmsemp_seq.nextval,#{name},#{dept},
#{grade},#{phone},#{email})
</insert>
```

**Business logic**

1. 사원정보 입력
2. 등록 버튼 클릭
3. 입력한 사원정보 DB에 저장

>사원정보 수정	
이름	김대중
부서명	개발1팀 <input type="button" value="개발1팀"/>
직책	부장 <input type="button" value="과장"/>
핸드폰	010-2175-9239
이메일	DJ1998@daum.net



퇴사

수정

이전

**mapper**

```

<!-- 직급, 부서 수정-->
<update>
UPDATE pmsemp
SET dept =#{dept}, grade =#{grade}
WHERE eno =#{eno}
<!-- 직급 리스트 -->
<select>
select DISTINCT grade
from pmsemp
order by grade asc
<!-- 부서 리스트 -->
<select>
select DISTINCT dept
from pmsemp
WHERE dept IS NOT null
order by dept ASC

```

**Business logic**

1. 부서 또는 직책 선택
2. 수정 버튼 클릭
3. 해당 직원의 사원번호로 update로 수정 처리
4. 부서와 직급 select box는 DB에 등록된 데이터 리스트로 불러옴

## &gt;사원정보 수정

이름 김대중

부서명 개발1팀 직책 부장 

핸드폰 010-2175-9239

이메일 DJ1998@daum.net

**mapper**

```
<!-- 퇴사처리 -->
<!-- pms등록된 직원 -->
<update>
UPDATE pmsemp
SET grade = '퇴사', dept = null
WHERE eno =#{eno}
<update>
UPDATE pmsmember
SET mdiv = 8 , pno = null
WHERE mno =#{eno}
<!-- pms등록 안된 직원 -->
<delete>
delete from pmsemp
WHERE eno =#{eno}
```

**service**

```
// 퇴사 처리
public void deleteG(pmsemp deleteG) {
int isMem = rep.memCheck(deleteG);
// PMS등록 확인
if(isMem==0) {
// PMS등록 안된직원 delete
rep.delGrade3(deleteG);
} else {
// PMS등록된 직원 update
rep.delGrade1(deleteG);
rep.delGrade2(deleteG);
```

**Business logic**

1. 퇴사 버튼 클릭
2. 해당 직원이 PMS에 등록되어 있는지 확인
3. 등록이 안되어 있다면 delete로 DB에서 데이터를 삭제
4. PMS에 등록되어 있다면 해당 직원의 PMS업무 정보를 남기기위해 직급을 퇴사, 부서를 null, 권한을 퇴사자로 변경, 프로젝트 번호로 null로 변경



화면구현

## 프로젝트 총괄/통합관리

유재인

≡ V PROJECT

대표이사 문재인

PM 대시보드

인적자원관리

### 프로젝트 현황

PMS시스템 개발 PM 김대중

31 %

간트차트 보기

공지사항 보기

회의록 보기

이슈리스트 보기

총업무 : 43  
완료업무 : 14  
32 %

미완료업무 : 29  
진행중업무 : 12  
41 %

완료업무 : 14  
기한초과업무 : 5  
35 %

쇼모꼬 개발 PM 김영삼

31 %

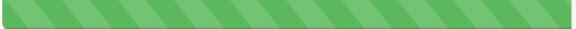
Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다.

2020. 6. 8 마감

PMS시스템 개발

PM 김대중

31 %

2020. 6. 8 마감

## service

```
if(plist.get(i).getGrade()<0 &&
(plist.get(i).getProg() != 100 ||  
!plist.get(i).getTdiv().equals("결재완료")) {  
    p.setGrade(2);  
    System.out.println(p.getGrade());  
}  
  
if(prog+10<dprog&&p.getGrade()==0)  
    p.setGrade(1);
```

## Business logic

1. 최상위 업무들 기준
2. 기한 내 완료 못한 업무가 있는 경우 빨간색
3. 진행률이 예상 진행률보다 낮은 경우 주황색
4. 그 이외의 경우 초록색

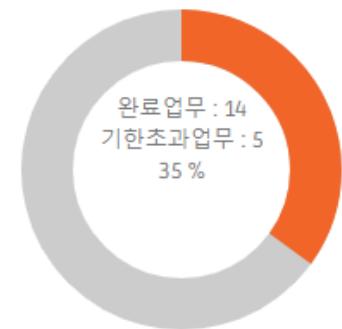
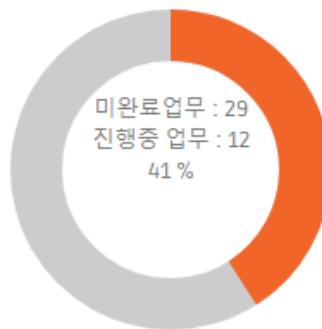
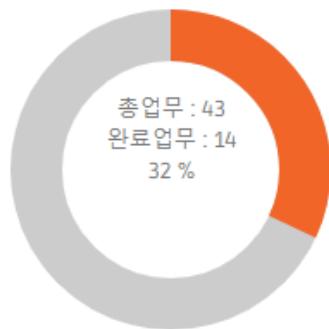
[간트차트 보기](#)[공지사항 보기](#)[회의록 보기](#)[이슈리스트 보기](#)

## controller

```
HttpSession session = request.getSession();
PmsMember emp =(PmsMember)session.getAttribute("infor_M");
int pno = Integer.parseInt(request.getParameter("pno"));
int proc = Integer.parseInt(request.getParameter("proc"));
emp.setPno(pno);
session.setAttribute("infor_M", emp);
if(proc==1) {
    return "forward:/chart.do?method=pm";
} else if(proc==2) {
    return "forward:/risk.do?method=list";
} else if(proc==3) {
    return "forward:/meet.do?method=list";
} else {
    return "forward:/notice.do?method=list";
}
```

## Business logic

- 로그인시 infor\_M 세션이 생성
- infor\_M 객체 중 프로젝트 번호 항목 존재
- CEO/CTO의 경우 맡은 프로젝트가 없음
- 클릭시 Ctrl단에서 infor\_M에 해당프로젝트 번호를 입력  
후 세션 변경
- Ctrl단에서 해당 Ctrl로 forward



## view

```
<div class="chart" data-percent="${ceo.cp }"  
style="float: left; width: 30%; height: 230px;  
text-align: center; margin-left: 50px;"><br>  
<span class="title">총업무 : ${ceo.total }</span>  
<span class="title">완료업무 : ${ceo.ccnt }</span>  
<span class="title">${ceo.cp } %</span></div>
```

## Business logic

1. 총업무 - 프로젝트 내 총 업무
2. 완료업무 - 결재완료된 업무
3. 진행중업무 - 진행률이 0이 아닌 업무
4. 기한초과업무 - 결재일이 완료일보다 늦은 업무

**내 프로젝트 현황**

PMS시스템 개발



31%

PM 김대중

2020. 6. 8 마감

---

**팀원별 업무 현황**

팀원	업무	진행률
추미애	ERD 설계	51% 완료
오세훈	1주차 발표 피드백 개선방안 논의	50% 완료
안희정	ERD 설계	66% 완료
박영선	미해결	0%

**> 공지사항 최근글**

#	제목	작성자	작성일
11	회의 공지	김대중	21일전
9	주말 과제 공지	김대중	24일전
7	1주차 발표에 대한 피드백 회의 공지	김대중	25일전

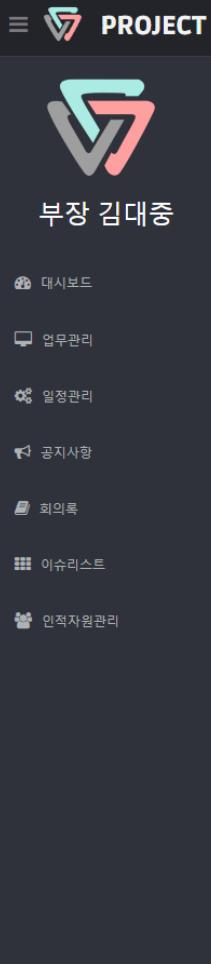
**> 회의록 최근글**

#	제목	작성자	작성일
10	ERD 설계	안희정	20일전
8	1주차 발표 피드백 개선방안 논의	추미애	23일전
5	ERD 설계	오세훈	26일전

**> 이슈리스트 최근글**

#	해결여부	작성자	작성일
18	미해결	홍준표	5일전
17	미해결	김무성	8일전
16	미해결	김무성	8일전



추미애



51 % 완료

오세훈



50 % 완료

## service

```
if(tasks.get(j).getGrade()<0 &&
   (tasks.get(j).getProg() != 100 ||
    !tasks.get(j).getTdiv().equals("결재완료")) ) {
    team.setGrade(2);
    t=2;
} else if(tasks.get(j).getProg()<(dprog>100?100:dprog)) {
    t=1;
}
```

## Business logic

1. 해당 팀원의 전체 업무에 대하여
2. 기한 내 완료 못한 업무가 있는 경우 빨간색
3. 진행률이 예상 진행률보다 낮은 경우 주황색
4. 그 이외의 경우 초록색

## &gt; 공지사항 최근글

#	제목	작성자	작성일
11	회의 공지	김대중	21일전
9	주말 과제 공지	김대중	24일전
7	1주차 발표에 대한 피드백 회의 공지	김대중	25일전

## &gt; 회의록 최근글

#	제목	작성자	작성일
10	ERD 설계	안희정	20일전
8	1주차 발표 피드백 개선방 안논의	추미애	23일전
5	ERD 설계	오세훈	26일전

## view

```

<h4 onclick="goNot()"><i class="fa fa-angle-right">
</i> 공지사항 최근글</h4>
<hr>
<table class="table">
  <thead>
    <tr>
      <th>#</th>
      <th>제목</th>
      <th style="width:70px;">작성자</th>
      <th style="width:70px;">작성일</th>
    </tr>
  </thead>
  <tbody>
    <c:forEach var="notice" items="${nlist}">
      <tr onclick="goNoti(${notice.nno})">
        <td>${notice.nno }</td>
        <td>${notice.title }</td>
        <td>${notice.writer }</td>
        <td>${notice.wdate }일전</td>
      </tr>
    </c:forEach>
  </tbody>
</table><br><br><br>
```

## Business logic

1. 프로젝트에 해당하는 게시판 연결
2. 최근 게시글 3개씩 표시
3. 게시판 이름 클릭 시 리스트로 이동
4. 해당 글 선택 시 상세화면으로 이동

채팅
개인정보
참여인원
Logout

### 내 프로젝트 현황

PMS시스템 개발

31%

PM 김대중

2020. 6. 8 마감

---

### 세부 업무 현황

업무명	시작일
로그인/로그아웃(웹)	2020. 5. 8 시작
번호 찾기(웹)	2020. 5. 12 시작
개인정보(웹)	2020. 5. 16 시작
로그인/로그아웃(앱)	2020. 5. 20 시작

> 공지사항 최근글

#	제목	작성자	작성일
11	회의 공지	김대중	21일전
9	주말 과제 공지	김대중	24일전
7	1주차 발표에 대한 피드백 회의 공지	김대중	25일전

> 회의록 최근글

#	제목	작성자	작성일
10	ERD 설계	안희정	20일전
8	1주차 발표 피드백 개선방안논의	추미애	23일전
5	ERD 설계	오세훈	26일전

> 이슈리스트 최근글

#	해결여부	작성자	작성일
18	미해결	홍준표	5일전
17	미해결	김무성	7일전
16	미해결	김무성	8일전

개인정보(웹)

2020. 5. 16 시작

100 % 완료

2020. 5. 20 마감

로그인/로그아웃(앱)

2020. 5. 20 시작

10 % 완료

2020. 5. 24 마감

## service

```
if(tasks.get(i).getGrade()<0 &&
   (tasks.get(i).getProg() != 100 ||
   !tasks.get(i).getTdiv().equals("결재완료")) ) {
    t.setGrade(2);
} else if(tasks.get(i).getProg()<(dprog>100?100:dprog)) {
    t.setGrade(1);
} else {
    t.setGrade(0);
}
```

## Business logic

1. 해당 팀원의 개별 업무에 대하여
2. 기한 내 완료 못한 업무가 있는 경우 빨간색
3. 진행률이 예상 진행률보다 낮은 경우 주황색
4. 그 이외의 경우 초록색



**화면구현**

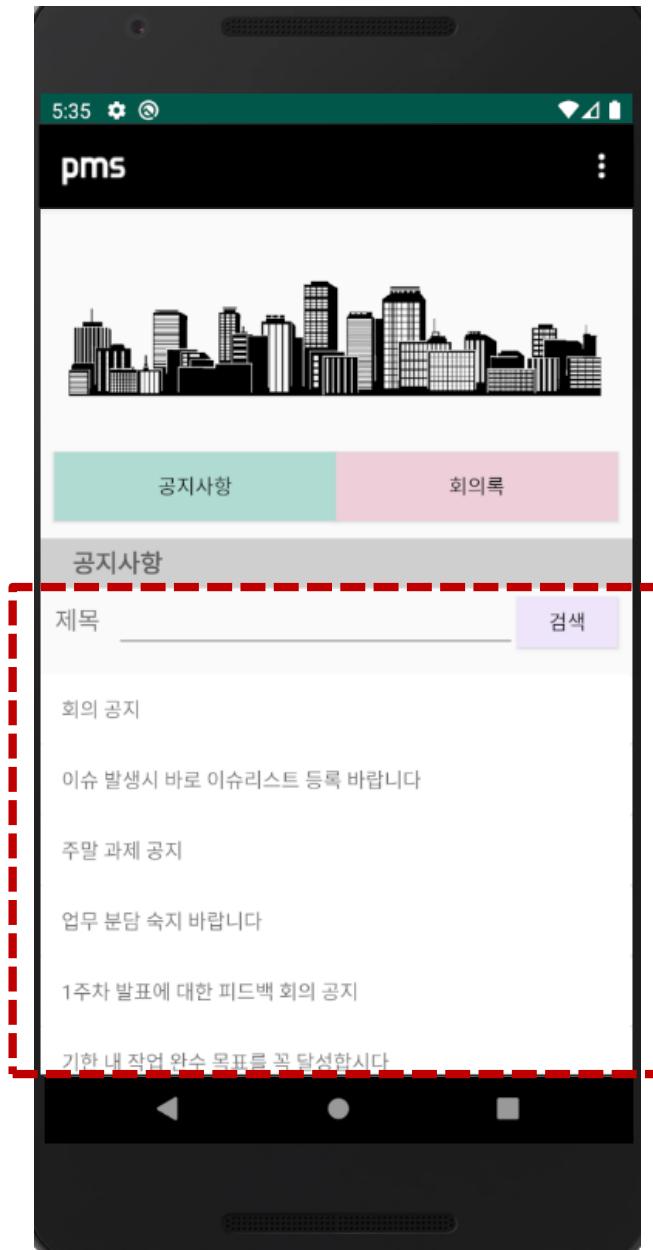
**APP**



화면구현

# 프로젝트 의사소통관리

김하나

**service**

```
// 안드로이드 리스트 출력
public ArrayList<Notice> andlist(Notice sch) {
    return dao.andlist(sch);
}
```

**DAO**

```
public ArrayList<Notice> andlist(Notice sch);
```

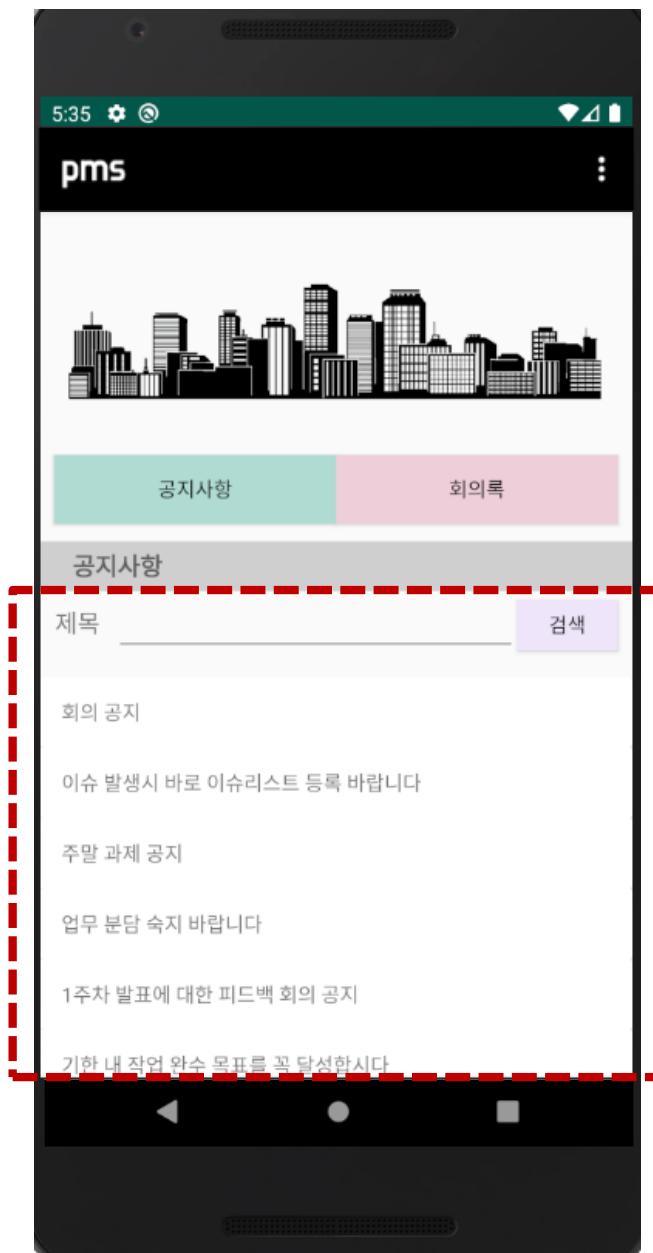
**Ctrl**

```
//안드로이드 db 연결
//http://localhost:5080/project06_git/meet.do?method=applist
@RequestMapping(params="method=applist")
public String andlist(Meet sch, Model d) {
    d.addAttribute("andlist", service.andlist(sch));
    return "WEB-INF\\views\\main\\andMeet.jsp";
}
```

**mapper**

```
<select id="andlist" resultMap="noticeResult" parameterType="nsch">
    SELECT title
    FROM pmsnotice
    WHERE 1=1
    <if test="title!=null and title!=''">
        and title like '%'||#{title}||'%'
    </if>
    ORDER BY nno DESC
</select>
```

\* Vo= 그대로 사용



## layout

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/abc"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

## Java - list

```
public class notice_List {
    private ArrayList<notice_vo> andlist;
    public ArrayList<notice_vo> getAndlist() { return andlist; }
    public void setAndlist(ArrayList<notice_vo> andlist) { this.andlist = andlist; }
}
```

## Java - adapter

```
@Override
public int getItemCount() { return items.size(); }
// 추가되는 단위(VO)를 ArrayList에 할당 처리 메서드..
public void addNewItem(notice_vo item) { items.add(item); }
// 전체 데이터 객체를 저장..
public void setItems(ArrayList<notice_vo> items) { this.items = items; }
// 각각의 단위객체 index로 return 처리..
public notice_vo getItem(int position) { return items.get(position); }
// 특정 단위를 객체를 지정한 위치에 저장
public void setItem(int position, notice_vo item) { items.set(position, item); }
// RecyclerView.ViewHolder를 상속받은 ViewHolder 내부 객체가
// 단위 화면의 요소객체와 VO 데이터를 mapping처리한다.
static class ViewHolder extends RecyclerView.ViewHolder {
    TextView textView;
```



## layout

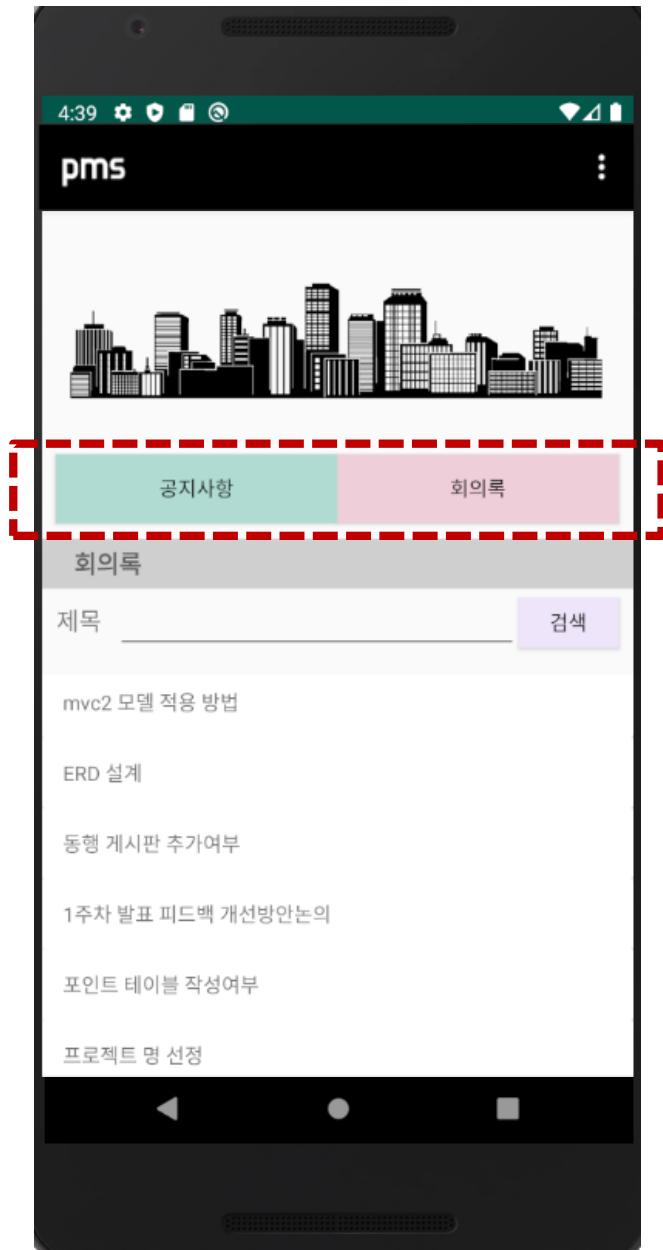
```
LinearLayout(vertical)
    imageView2
    LinearLayout(horizontal)
        button2_6- "공지사항"
        button2_10- "회의록"
    ScrollView
        LinearLayout(vertical)
            textView4- "공지사항"
        LinearLayout(vertical)
            textView15- "TextView"
            editText4- "내용을 입력해주세요"
            textView16- "TextView"
            buttonnn- "글쓰기"
```



## layout

```
<Button  
    android:id="@+id/button2_6"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="10dp"  
    android:layout_marginTop="10dp"  
    android:layout_weight="1"  
    android:background="#AFDBD3"  
    android:clickable="true"  
    android:text="공지사항" />
```

```
<Button  
    android:id="@+id/button2_10"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp"  
    android:layout_marginRight="10dp"  
    android:layout_weight="1"  
    android:background="#EECED9"  
    android:clickable="true"  
    android:text="회의록" />
```



## java

```
Button bt6 = (Button) findViewById(R.id.button2_6);
bt6.setOnClickListener(noti);

View.OnClickListener noti = new View.OnClickListener() {
    Intent int01;

    public void onClick(View v) {
        int01 = new Intent(getApplicationContext(), notice.class);
        startActivity(int01);
    }
};

Button btp = (Button) findViewById(R.id.button2_10);
btp.setOnClickListener(pers);

View.OnClickListener pers = new View.OnClickListener(){
    Intent int01;
    public void onClick(View v){
        int01 = new Intent(getApplicationContext(), meet.class);
        startActivity(int01);
    }
};
```



화면구현

# 프로젝트 리스크관리

이준석



## Mapper

```
<select id="andlist" resultMap="riskResult" parameterType="rsch" >
    SELECT p.ino,p.title,pm.NAME,p.VIEWS ,p.pno
    FROM PMSEMP pm,PMSISSUE p
    WHERE pm.ENO =p.MNO
    ORDER BY ino DESC
</select>
```

## View

```
{"andlist":
  [<c:forEach var="risk" items="${andlist}" varStatus="sts">
    {"ino":${risk.ino},"title":${risk.title}),"name":${risk.name}>,
    "views":${risk.views}, "pno":${risk.pno}}
    <c:if test="${!sts.last}">,</c:if></c:forEach>
  ]}
```

## Main.java

```
adapter = new riskAdapter();
recyclerView.setAdapter(adapter);
makeRequest();
```

```
String url = "http://192.168.4.112:5080/project06\_git/risk.do?method=applist";
```

## ArrayList.java

```
private ArrayList<risk_list> andlist;
public ArrayList<risk_list> getAndlist() {
    return andlist;
}
public void setAndlist(ArrayList<risk_list> andlist) {
    this.andlist = andlist;
}
```



## Mapper

```
<select id="andlist" resultMap="riskResult" parameterType="rsch" >
    SELECT p.ino,p.title,pm.NAME,p.VIEWS ,p.pno
    FROM PMSEMP pm,PMSISSUE p
    WHERE pm.ENO =p.MNO
    <if test="title != null and title != ''"> 추가
        and title like '%'||#{title}||'%'
    </if>
    ORDER BY ino DESC
</select>
```

## Main.java

```
"http://192.168.4.112:5080/project06_git/risk.do?method=applist"+ "&" + "title=" +ename; 추가
editText = findViewById(R.id.editText4);
String ename = editText.getText().toString();
EditText에 입력된 값 ename에 등록
button = findViewById(R.id.button8);
button.setOnClickListener(v) {
    adapter = new riskAdapter();
    recyclerView.setAdapter(adapter);
    makeRequest();
});
버튼 클릭 시 검색된 내용 출력
```



화면구현

# 프로젝트 통합관리

이연학

**Json****pfSearch.jsp**

```
{
  "wlist": [
    [
      <c:forEach var="work" items="${wlist}" varStatus="sts">
        {
          "tno":${work.tno}, "tname":${work.tname}, "tdiv":${work.tdiv}, "name":${work.name}
          <c:if test="${!sts.last}">,</c:if></c:forEach>
        }
    ]
  ]
}
```

**Ctrl**

```
@RequestMapping(params="method=applist")
public String perforlist(Task sch, Model d) {
    d.addAttribute("wlist", service.workList(sch));
    return "WEB-INF\\views\\main\\pfSearch.jsp";
}
```

**java**

```
String url = "http://192.168.4.235:5080/project06_git/task.do?method=applist"
        +"&"+"tname="+tname;
```



# *Process Management*

# 회의록

일자	주제	세부내용	결론
2020-05-04	프로젝트 주제 선정	pms와 다른 주제들 중 이번 프로젝트로 진행할 주제 선정	나이키 등 의류 쇼핑몰&지점관리 시스템
2020-05-04	프로젝트 역할 분담	프로젝트에서 각자 담당할 part 역할 분담 및 백업/협업 계획 수립	역할분담 완료
2020-05-04	bootstrap template 결정	프로젝트에 활용할 bootstrap template 비교/논의 및 결정	선정 완료
2020-05-06	ERD 설계	요구사항정의서와 화면 설계를 바탕으로 ERD 설계	러프한 ERD 개요 설계
2020-05-08	1주차 발표 피드백 개선방안논의	발표에 대한 피드백 확인 및 자체 회의를 통한 프로젝트 개선방향 및 추가로 프로젝트에 필요한 기능 논의	주말을 활용하여 결정된 사안에 대해 각자 업무 처리 후 월요일에 다시 논의
2020-05-11	ERD 설계	테이블간 식별/비식별 관계 및 세부 컬럼 논의	DB 생성 후 추후 수정
2020-05-15	2주차 발표 피드백 개선방안논의	게시판 디자인 통일/검색기능 추가/개발환경 권한추가/소스통합/간트차트개선	19일까지 수정/추가
2020-05-18	3주차 진행 예정 논의	이번주까지 웨일온	힘내자
2020-05-19	피드백 개선 내용 점검	게시판 디자인 통일, 검색기능 추가 완료 include top에서 권한추가 완료	파이팅
2020-05-21	3주차 발표 내용 점검	게시판 정렬 문제 / 상세페이지 디자인 통일	게시판 정렬문제 - sql 수정으로 해결 디자인통일은 다음주 월요일까지
2020-05-22	3주차 피드백 내용 점검	각자 파트에서 개선되야 하는 부분 논의	차주까지 개선 완료
2020-05-25	4주차 진행 예정 논의	이번 주 각자 파트 논의 및 분배	28일 목요일 까지 완료



# ISSUE List

일자	담당자	이슈내용	해결여부	해결방안
2020-05-06	유재인 전현규	ERD설계에 있어 계시판 통합의 문제 사용자 번호와 사원 번호 중복의 문제 테이블간 연결시 식별/비식별 구분 문제 등	X	DB생성과 ERD수정의 시행착오 수정법으로 접근하여 해결해 볼 예정
2020-05-11	유재인	bootstrap 양식에서 footer가 하단에 고정되지 않는 문제 발생	O	css 파일에서 footer 하단 고정을 설정하고 main-content div의 min-height를 부여함 / include되는 top.jsp와 각 세부 페이지간의 충돌을 방지함
2020-05-11	김하나	css를 수정하여 footer 하단 고정을 처리하였으나 화면단에 반영이 되지 않고 수정 전 css 파일이 지속적으로 적용됨	O	브라우저 환경설정에서 캐시된 이미지 및 파일에 대하여 인터넷 사용 기록 삭제를 처리하여 해결
2020-05-14	전현규	데이터를 JSON형식으로 받아오는데 성공했으나 차트 API에 적용시키는 중에 요구하는 데이터 사이즈에 따른 문제 발생	O	자바스크립트 상에서 JSON형식으로 받아 API에 적용
2020-05-18	전현규	사용하는 간트차트에서 y축 정보를 추가하면 hierarchy(계층구조) 기능이 작동하지 않음을 확인	O	Y축 정보를 사용하지 않는 방안(1안), hierarchy를 사용하지 않는 방안(2안) 이 있었고 이중 2안을 사용하는 것으로 결정
2020-05-19	전현규	팀원 간트차트에서 상위 업무 정보를 못받아와서 표현상 문제 발생	O	SQL문의 UNION기능을 사용해 불러오는데 성공
2020-05-19	유재인	대시보드 개발을 진행할 시간이 부족함... 여기저기 불려다님....	X	살려주세요....
2020-05-19	김형준	권한수정 상세페이지에서 권한수정 시 오류 발생	O	컨트롤러에서 사원번호를 eno로 받는 걸 peno로 수정하면서 상세페이지에 eno와 peno 두 가지로 받아서 해결
2020-05-21	전현규	프로젝트 추가에 있어서 로그인 세션과 동일한 필드명 사용으로 인한 400에러 발생	O	HttpServletRequest방식을 사용해 파라미터에 해당하는 값을 따로 받아와 입력처리
2020-05-25	전현규	차트의 계층구조를 표현할 때 DB에서 불러온 공백(space)이 적용되지 않는 문제 발생	O	차트 내부 useHTML기능을 활성화하고 Oracle의 LEVEL기능을 이용해서 &nbsp;를 활용해 공백 입력 처리
2020-05-27	유재인	chart plugin을 사용하는 과정 중 body에 include된 top.jsp에서 jQuery가 재선언되어 plugin이 사용할 수 없는 문제 발생	O	plugin 선언 및 관련 함수 부분을 head가 아닌 body 내부 include 및에 위치시켜 해결





# *Software Quality Assurance*

## QA란?

- 품질 보증을 뜻하는 Quality Assurance의 약어
- 제품을 시장에 내보내도 되는지를 보장해주는 업무로 테스트와 이슈 등록 및 처리 나아가 Risk 예방까지가 QA의 업무임

## QA 설정

1. 테스트 시나리오 작성
2. 단위 테스트 수행 및 오류 수정
3. 교차 테스트 수행 및 오류 수정
4. 통합 테스트 수행 및 오류 수정

# 소프트웨어 QA

## Test 케이스리포트

테스트코드	테스트내용	예상 결과	테스트 결과
3001 - CEO 로그인			
3001-01	10000001 / 1234qwer!로 로그인	로그인 성공	
3001-02	사이드바 확인	사이드바에 대시보드와 인적자원관리 메뉴 2가지만 보임	
3001-03	대시보드 확인	대시보드에 등록된 2개의 프로젝트가 보이고 맨 아래 프로젝트 추가 버튼 보임	
3001-04	프로젝트 추가 버튼	프로젝트추가 버튼 클릭 시 해당 화면으로 전환됨	
3001-05	사이드바 대시보드 메뉴 클릭	프로젝트 추가X 사이드바의 대시보드 클릭하면 대시보드 정상적으로 보임	
3001-06	프로젝트 별 메뉴 확인	프로젝트별 게이지바 1개 / 파이차트3개 정상적으로 보임	
3001-07	간트차트 버튼 클릭	간트차트로 이동 확인	
3001-08	공지사항 버튼 클릭	공지사항 이동 확인	
3001-09	회의록 버튼 클릭	회의록 이동 확인	
3001-10	이슈리스트 버튼 클릭	이슈리스트 이동 확인	
3002 - CTO 로그인			
3002-01	10000003 / qwer1234!로 로그인	로그인 성공	
3002-02	사이드바 확인	사이드바에 대시보드와 인적자원관리 메뉴 2가지만 보임	
3002-03	대시보드 확인	대시보드에 등록된 2개의 프로젝트가 보이고 맨 아래 프로젝트 추가 버튼 보임	
3002-04	프로젝트 추가 버튼	프로젝트추가 버튼 클릭 시 해당 화면으로 전환됨	
3002-05	사이드바 대시보드 메뉴 클릭	프로젝트 추가X 사이드바의 대시보드 클릭하면 대시보드 정상적으로 보임	
3002-06	프로젝트 별 메뉴 확인	프로젝트별 게이지바 1개 / 파이차트3개 정상적으로 보임	
3002-07	간트차트 버튼 클릭	간트차트로 이동 확인	
3002-08	공지사항 버튼 클릭	공지사항 이동 확인	
3002-09	회의록 버튼 클릭	회의록 이동 확인	
3002-10	이슈리스트 버튼 클릭	이슈리스트 이동 확인	
3003 - PM로그인1			
3003-01	10000005 / 1234qwer! 로 로그인	로그인 성공	
3003-02	사이드바 확인	사이드바에 7가지 메뉴 보임	
3003-03	대시보드 확인	대시보드 정상적으로 보임	
3003-04	대시보드 팀원 확인	대시보드에 팀원 7명 각각 이름+게이지바 보임	
3003-05	되돌아오기 체크	사이드바 메뉴 클릭 후 해당페이지에서 대시보드 메뉴 클릭시 대시보드 정상	

# 소프트웨어 QA

## 통합테스트

테스트 코드	테스트 내용	예상 결과	테스트 결과	담당자	테스터 1	테스터 2	처리현황
2006-01	제목/내용 input 활성화	제목/내용 수정 가능	O	김하나	임진혁	김현태	
2006-02	수정 버튼 클릭	2006-1의 수정 사항 반영 / 수정일 파트에 sysdate 출력	O	김하나	임진혁	김현태	
2006-03	삭제 버튼 클릭	글 내용 삭제	O	김하나	임진혁	김현태	
2007 - 공지사항 글쓰기(CEO로그인)				김하나			
2007-01	제목/내용 입력창	제목/내용 입력 가능	O	김하나	임진혁	김현태	
2007-02	등록버튼	2007-1 등록	O	김하나	임진혁	김현태	
2007-03	조회화면으로 버튼	공지사항 리스트로 이동	O	김하나	임진혁	김현태	
2008 - 회의록 리스트(PM/팀원 로그인)				김하나			
2008-01	PM ID: 10000005 / PW: 1234qw	로그인 성공	PM 10000005로 로그인 시 500 에러 발생 PM 10000006 정상 로그인	김하나	임진혁	김현태	06-02 처리완료
2008-02	리스트 클릭	상세화면으로 이동/ 조회수 +1	O	김하나	임진혁	김현태	
2008-03	페이지 크기 클릭	한 화면에 보일 리스트 개수 조절 (3,5,10,20,30개)	O 3, 30개 표시되지 않음	김하나	임진혁	김현태	06-02 처리완료
2008-04	검색창	제목으로 리스트 검색 가능	O	김하나	임진혁	김현태	
2008-05	pagination	숫자를 누르면 해당 페이지로 이동	O	김하나	임진혁	김현태	
2008-06	글쓰기 버튼	클릭시 글작성 페이지로 이동	O	김하나	임진혁	김현태	
2009 - 회의록 상세(PM/팀원 로그인)				김하나			
2009-01	안건/회의참석자/내용/결정사항/	안건/회의참석자/내용/결정사항/특이사항 수정 가능	O	김하나	임진혁	김현태	
2009-02	수정 버튼 클릭	2009-1의 수정 사항 반영 / 수정일 파트에 sysdate 출력	O	김하나	임진혁	김현태	
2009-03	파일첨부	첨부파일 출력	첨부파일 출력X	김하나	임진혁	김현태	06-03 처리완료
2009-04	첨부파일 변경	첨부파일 변경 가능	첨부파일 변경X	김하나	임진혁	김현태	06-03 처리완료
2009-05	조회화면으로 버튼	회의록 리스트로 이동	O	김하나	임진혁	김현태	
2010 - 회의록 글쓰기(PM/팀원 로그인)				김하나			
2010-01	안건 등 입력창	입력 가능	O	김하나	임진혁	김현태	
2010-02	첨부파일	등록 가능	첨부파일 등록X	김하나	임진혁	김현태	06-03 처리완료
2010-03	등록버튼	2010-1, 2010-2 등록	O	김하나	임진혁	김현태	
2010-04	조회화면으로 버튼	회의록 리스트로 이동	O	김하나	임진혁	김현태	
3001-09 회의록 리스트(CEO)				김하나			
3001-09	회의록 버튼 클릭	회의록 이동 확인	O	김하나	임진혁	한송우	
3001-09-001	리스트 클릭	상세화면으로 이동/ 조회수 +1	O	김하나	임진혁	한송우	06-02 처리완료
3001-09-002	페이지 크기 클릭	한 화면에 보일 리스트 개수 조절 (3,5,10,20,30개)	1. 10,20,30 선택 시 숫자 짤림 1. 검색없이 검색창에 키워드 입력 후 페이지 이동시 자동검색 처리됨	김하나	임진혁	한송우	
3001-09-003	검색창	제목으로 리스트 검색 가능	O	김하나	임진혁	한송우	06-02 처리완료
3001-09-004	pagination	숫자를 누르면 해당 페이지로 이동	O	김하나	임진혁	한송우	
3001-09-005	글쓰기 버튼	비활성화(보이지 않음)	O	김하나	임진혁	한송우	
2011 - 회의록 상세(CEO)				김하나			
2011-01	수정/조회화면으로 버튼	비활성화(보이지 않음)	O	김하나	임진혁	한송우	
3001 - CEO 로그인				유재인			
3001-01	10000001 / 1234qwer로 로그인	로그인 성공	로그아웃 안됨	유재인	박현주	한송우	0602 처리완료
3001-02	사이드바 확인	사이드바에 대시보드와 인적자원관리 메뉴 2가지만 보임	O	유재인	박현주	한송우	
3001-03	대시보드 확인	대시보드에 등록된 2개의 프로젝트가 보이고 맨 아래 프로젝트 추가 버튼 보임	O	유재인	박현주	한송우	
3001-04	프로젝트 추가 버튼	프로젝트추가 버튼 클릭 시 해당 화면으로 전환됨	O	유재인	박현주	한송우	
3001-05	사이드바 대시보드 메뉴 클릭	프로젝트 추가X 사이드바의 대시보드 클릭하면 대시보드 정상적으로 보임	O	유재인	박현주	한송우	
3001-06	프로젝트 별 메뉴 확인	프로젝트별 게이지바 1개 / 파일자트3개 정상적으로 보임	O	유재인	박현주	한송우	

## 오류보고

일자	작성자	시나리오 번호	이슈내용
2020-06-02	이윤배	4001-05	URL의 쿼리스트링에선 pno가 1002로 출력되나 업무 등록화면에서 프로젝트번호가 1001로 입력되어있음 정보 입력후 등록 시 + 상세화면에서 정보 수정시에도 같은 에러 확인 Cause: java.sql.SQLIntegrityConstraintViolationException: ORA-02291: integrity constraint (SCOTT.FK_PMSPROJECT_TO_PMSTASK) violated - parent key not found 500 에러 확인
2020-06-02	이윤배	6002-08	PM을 '최규하' 이외에 다른 사람으로 지정 했을 때, 정보 입력 후 등록시 외래키관련 에러 발생 SQL []: ORA-02291: integrity constraint (SCOTT.FK_PMSPROJECT_TO_PMSMEMBER)
2020-06-02	임성빈	5003-05, 5007-05, 5014-05	첨부파일 다운로드 시 페이지 오류 (사이트에 연결할 수 없음)
2020-06-02	임성빈	5004-05~5004-16, 5008-04~5008-15, 5012-05~5012-15, 5015-04~5015-14	댓글창 수정버튼 클릭 시 팝업창은 호출되나 404에러( 메시지 //project06_git/risk.do)
2020-06-02	임진혁	2009-03-01, 2009-04-01	파일을 올리거나 수정할시 파일명이 출력 X
2020-06-02	이윤배	전체 업무관리 '상태'컬럼	통합테스트 시나리오에 '결재대기'로 표기된 부분 모두 '결재신청'으로 되어있음

## 오류수정

- 각 part 담당자에게 오류 내용 전달
- 오류 내용 확인 및 오류 수정
- 수정 후 해당 사항 단위 테스트 및 교차 테스트 수행
- 3번 수행 후 이상 없을 시 통과

**THANK YOU** ❤

---

