



코드로 사람을 만나다.

# 목차

- 간략한 사이트 소개
- 팀원소개 및 역할
- 기획의도
- 개발일정
- DB설계
- 개발 툴, 개발 환경
- 적용기술
- 시연
- 느낀점

코드로 사람을 만난다.



## 간략한 사이트 소개



간략한 사이트 소개

## 팀원 소개 및 역할



# 전성표 (직책 : 팀장)



## 포인트 삽

계정 생성(로그인, 회원 가입),  
암호화(password)  
페이스북API(로그인),  
SMTP 메일 인증

# 한상우 (직책 : 부팀장)



매칭 페이지(개발자 | 디자이너)  
관리자 메인 페이지 통계 제작  
마이 페이지 (개인 정보 수정, 통계, 그래프, 상품 문의)  
포인트 결제 출금 기능  
구매리스트 제작  
footer, header : View 제작

# 박형길 (직책 : 팀원)



## 커뮤니티 페이지

IP 차단기능 , 공지사항

포인트 출석이벤트

질의 응답 페이지

프로젝트.팀원 소개 페이지

## 김용균 (직책 : 팀원)



### 관리자 페이지

회원 정보 관리 및 권한 부여  
게시판 삭제, 수정  
1:1 문의 작성기능(답변 유무체크)

# 주병현 (직책 : 팀원)



## 고객센터

1:1 문의 하기 페이지  
공지사항 페이지  
자주 찾는 질문 페이지 제작  
뷰 메인 제작 및 구현  
관리자 페이지  
회원 정보, 게시판 삭제, 수정  
1:1 문의 작성기능(답변 유무체크)

# 기획 배경



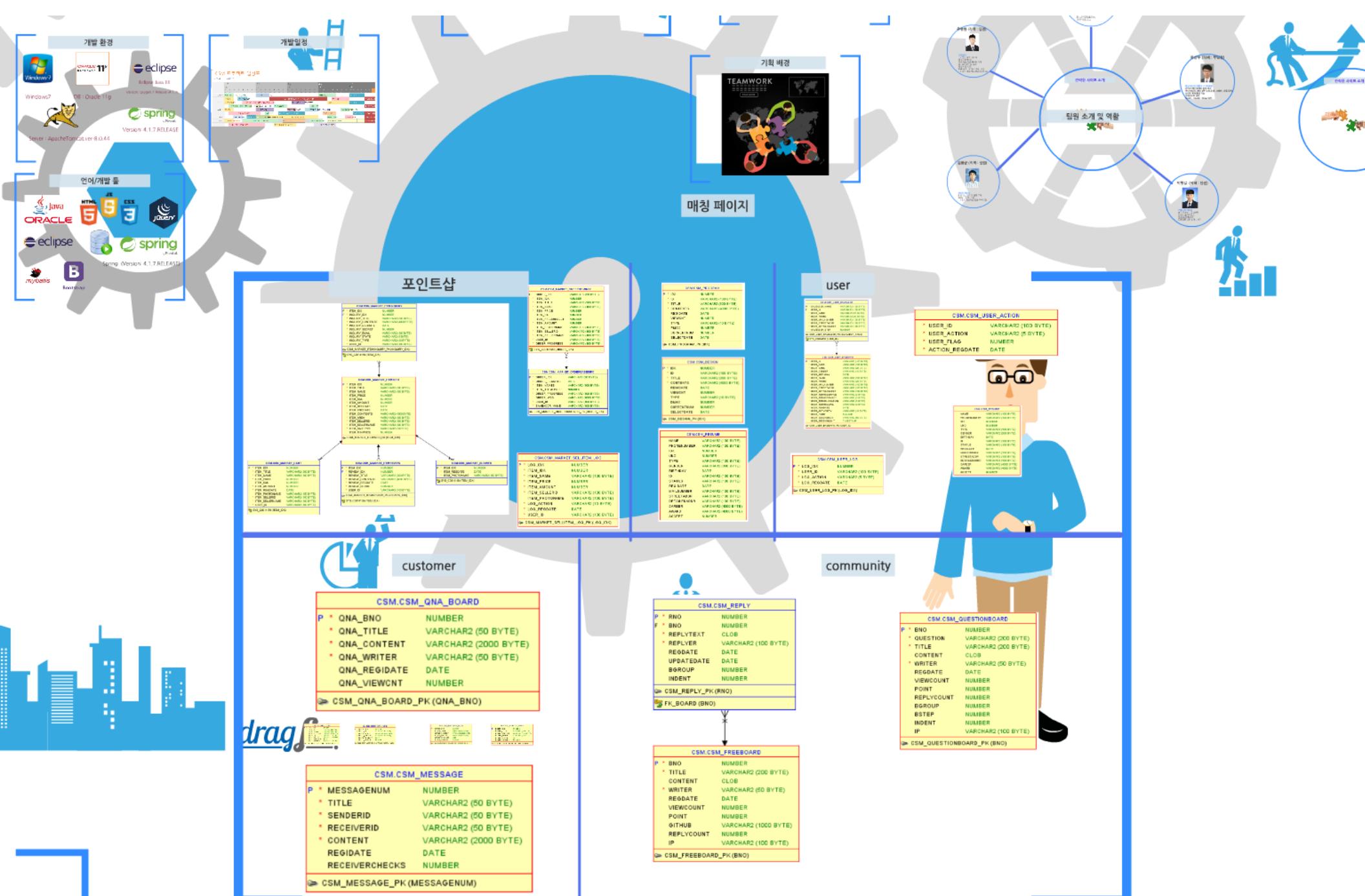
# 기획 배경



# 개발일정

CSM 프로젝트 일정표

시작일: 2018-2-8



모

# 개발 환경



Windows7



Server : ApacheTomcat.ver-8.0.44



DB : Oracle 11g



Eclipse Java EE

Version: Oxygen.2 Release (4.7.2)



Version: 4.1.7.RELEASE

# 언어/개발 툴

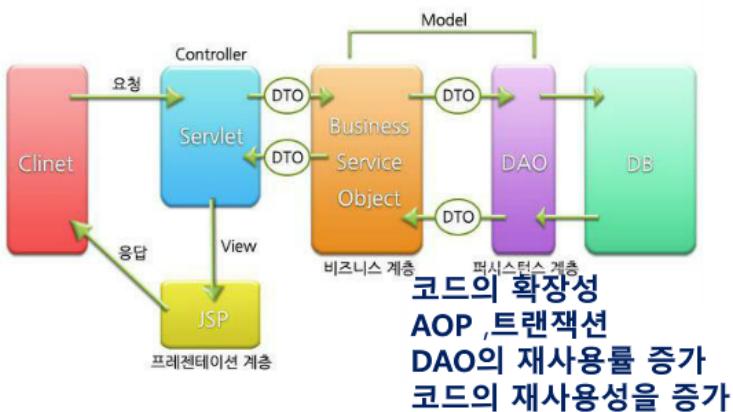


Bootstrap



spring  
by Pivotal™

Spring (Version: 4.1.7.RELEASE)



## 1. 유지 보수

## 2. 코드의 재사용

### 3. 속도 향상



```

<insert into csm_user_baseinfo
  (user_id, user_pass, user_name, user_gender, user_birthday, user_email, user_phone,
   user_mailnumber, user_streetid, user_detailedaddr, user_termservice, user_termsPrivacy
    </if test="user_termsLocation != ''">
      user_termsLocation
    </if>
    <if test="user_termsEmail != ''">
      user_termsEmail
    </if>
    <if test="user_termsName != ''">
      user_termsName
    </if>
    <if test="user_termsGender != ''">
      user_termsGender
    </if>
    <if test="user_termsBirthday != ''">
      user_termsBirthday
    </if>
    <if test="user_termsEmail != ''">
      user_termsEmail
    </if>
    <if test="user_termsName != ''">
      user_termsName
    </if>
    <if test="user_termsGender != ''">
      user_termsGender
    </if>
    <if test="user_termsBirthday != ''">
      user_termsBirthday
    </if>
  </if>
</if>
</insert>

<select id="UserBaseInfo" resultType="resultMap">
  < resultMap type="User" >
    < select * from csm_user_baseinfo where item_area = #area</select>
  </resultMap>
  < resultMap type="Area" >
    < select * from csm_market_info where item_id = #user_id</select>
  </resultMap>
</select>

```

## -DB 선처리 SQL문 작성



1. 유지 보수
2. 코드의 재사용
3. 속도 향상



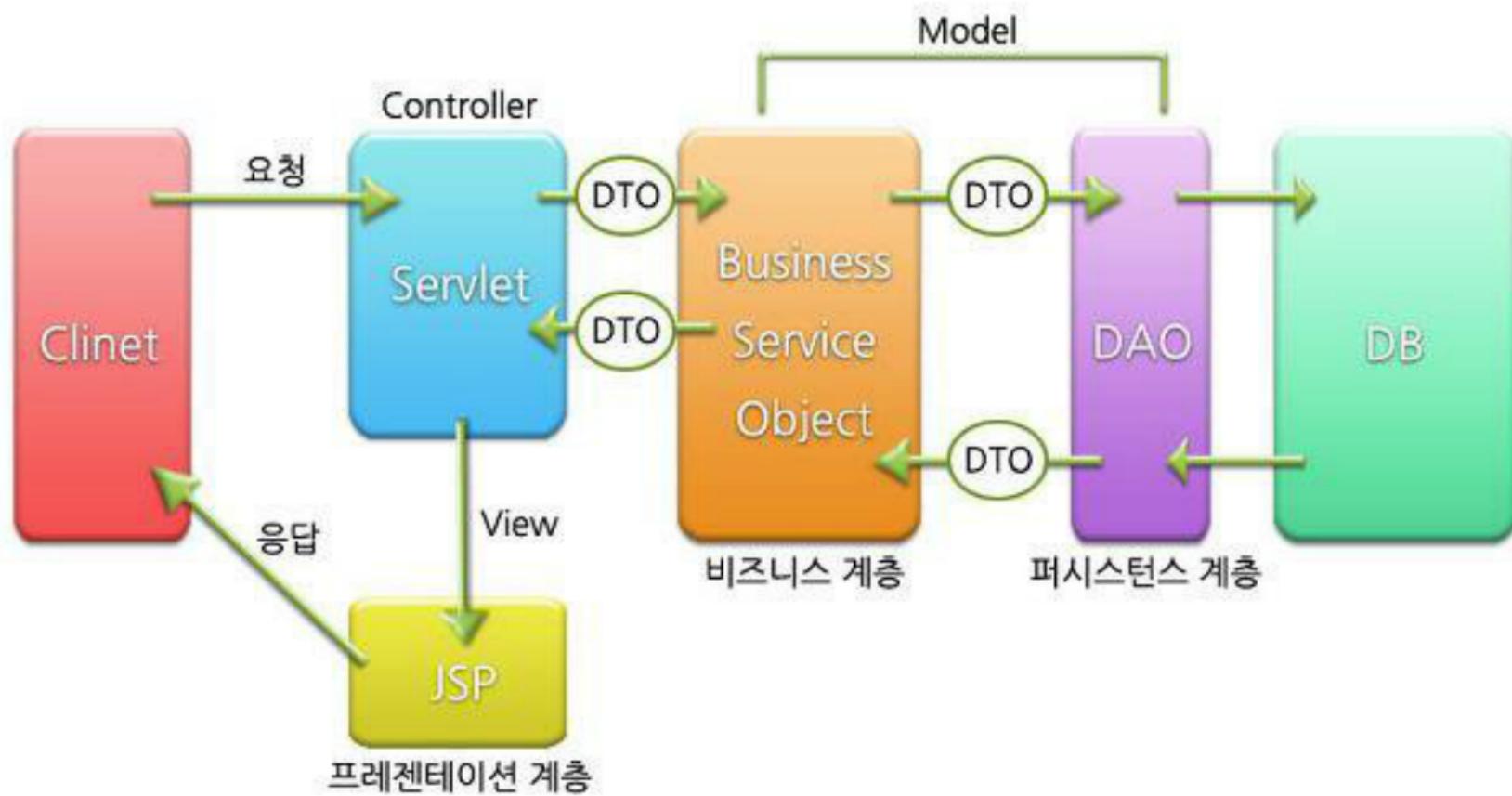
프로

## 1. 유지 보수

- 스프링 5계층 레이어
- x,y,z 식의 3단계 방법 사용
- 서비스와 서비스인터페이스의 적극적인 이용

# 1. 유지 보수

## -스프링 5계층 레이어



# 1. 유지 보수

-x,y,z 식의 3단계

com.samsung.galaxy.user.domain

GROUP ID

ARTIFACT ID

TABLE

TIER & LAYER

com.samsung.galaxy.user.service

GROUP ID

ARTIFACT ID

TABLE

TIER & LAYER

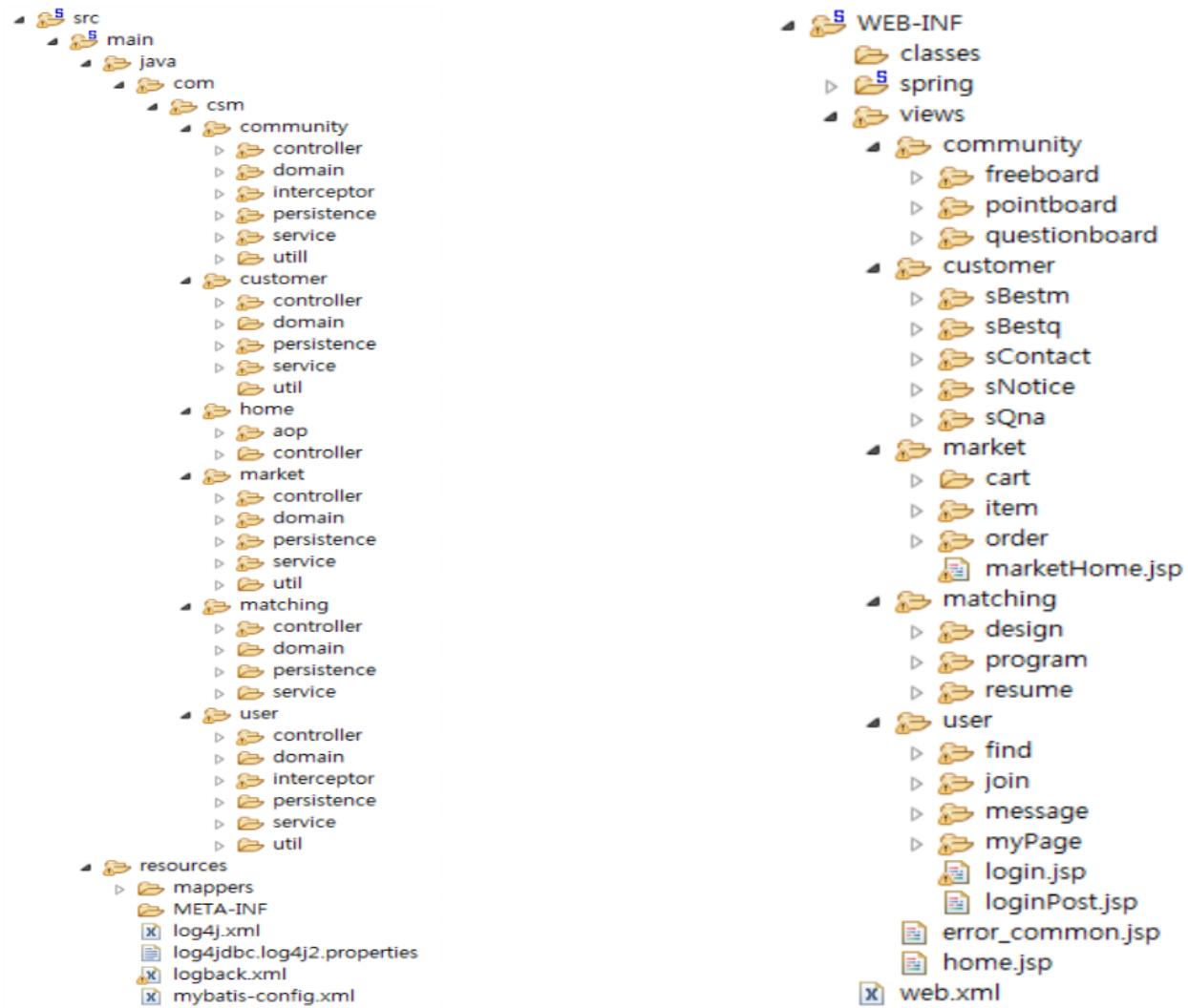
com.samsung.galaxy.user.dao

GROUP ID

ARTIFACT ID

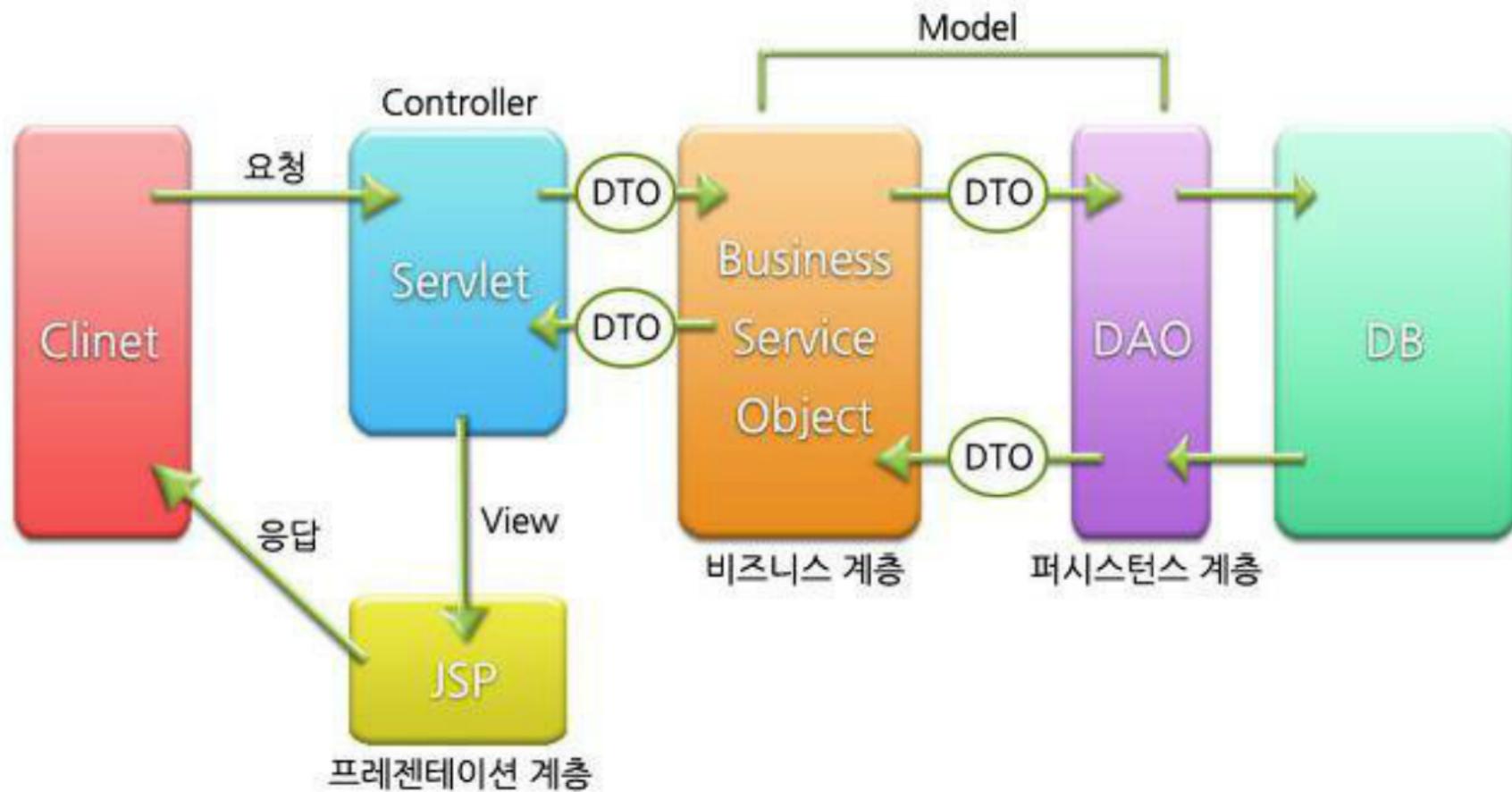
TABLE

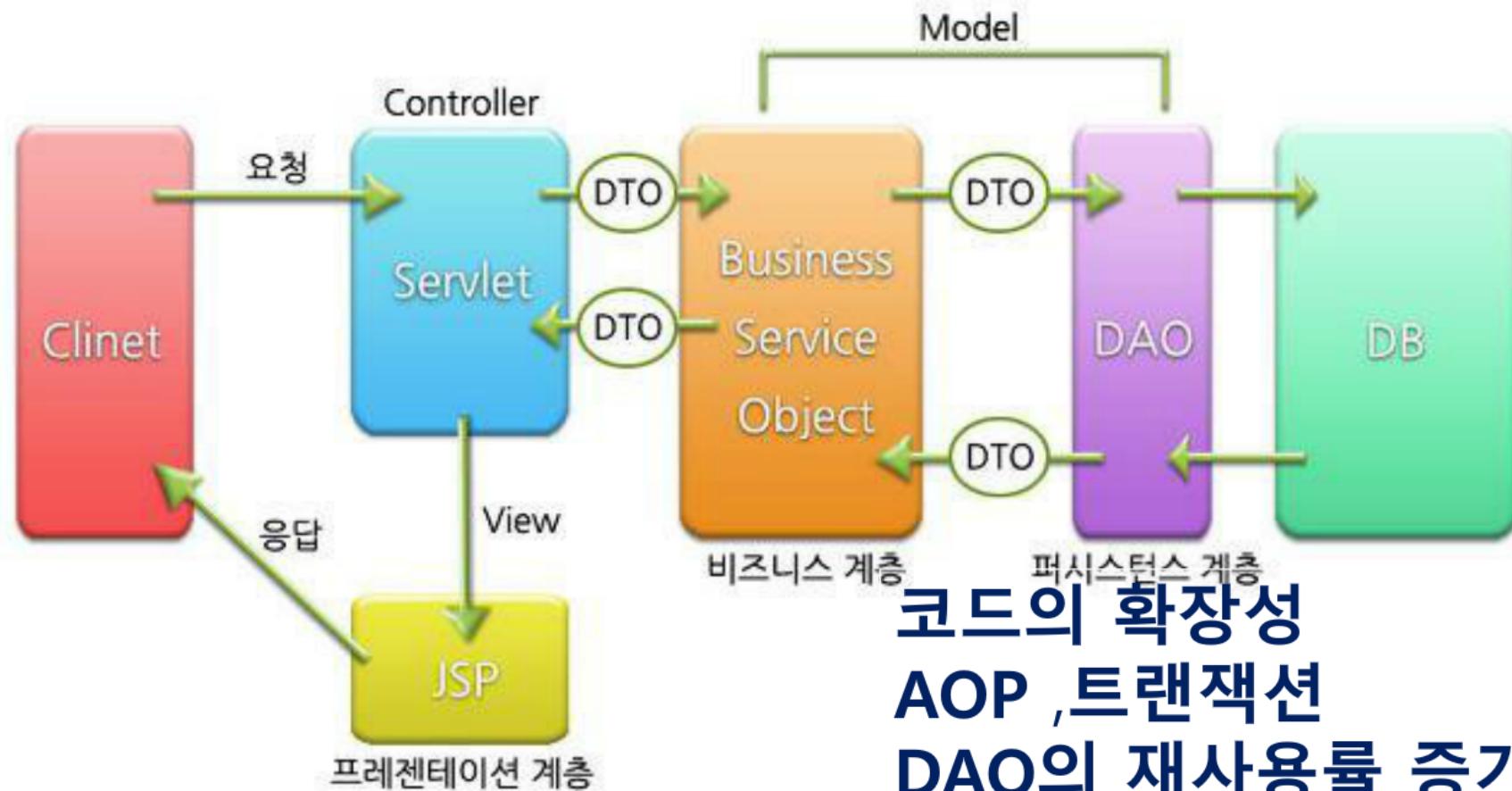
TIER & LAYER



# 1. 유지 보수

- 서비스와 서비스인터페이스의 적극적인 이용





**코드의 확장성  
AOP, 트랜잭션  
DAO의 재사용률 증가  
코드의 재사용성을 증가**

## 2. 코드의 재사용

### -My Batis 동적 SQL

```
<insert id="createBase">
    insert into csm_user_baseinfo
    (user_id, user_pass, user_name, user_gender, user_birthday, user_email, user_phone,
    user_mailNumber, user_streetAddr, user_detailedAddr, user_termsService, user_termsPrivacy
    <if test="user_termsLocation != ''">
        , user_termsLocation
    </if>
    <if test="user_termsEmail != ''">
        , user_termsEmail
    </if>
    ) values (#{{user_id}, #{{user_pass}, #{{user_name}, #{{user_gender}, #{{user_birthday}, #{{user_email},
    #{{user_phone}, #{{user_mailNumber}, #{{user_streetAddr}, #{{user_detailedAddr}, #{{user_termsService},
    #{{user_termsPrivacy
    <if test="user_termsLocation != ''">
        , #{{user_termsLocation}
    </if>
    <if test="user_termsEmail != ''">
        , #{{user_termsEmail}
    </if>
    )
</insert>

<select id="listBaseInfo" resultType="ItemVO">
    <choose>
        <when test="area != ''">
            select * from csm_market_itemInfo where item_area = #{{area}}
        </when>
        <otherwise>
            select * from csm_market_itemInfo where item_sellerId = #{{user_id}}
        </otherwise>
    </choose>
</select>
```

# 3. 속도 향상

## -DB 선처리 SQL문 작성

```

<select id="listAll" resultType="ProgramVO">
<include refid="pagingHeader" />
<![CDATA[
    SELECT
        TO_NUMBER(SUBSTR (DIFF, 2, 9)) || '일 ' || SUBSTR (DIFF, 12, 2) || '시간 ' || SUBSTR (DIFF, 15, 2) || '분 ' || SUBSTR (DIFF, 18, 2) || '초' AS MAX_TIME,
        A.*
    FROM (
        SELECT
            IDX,
            ID,
            TITLE,
            CONTENTS,
            REGIDATE,
            VIEWCNT,
            PMAX,
            CURRENTNUM,
            SELECTDATE,
            ROWNUM,
            TO_CHAR(SYSDATE, 'yyyymmddhh24miss') TIME1,
            TO_CHAR(REGIDATE, 'yyyymmddhh24miss') TIME2,
            (SELECT NUMTODSINTERVAL(SELECTDATE - SYSDATE, 'day')
             FROM DUAL) DIFF,
            CASE WHEN (SELECTDATE - SYSDATE ) <= 0 THEN '마감' ELSE '진행' END AS CHECK_TIME
        FROM
            CSM_PROGRAM
        ORDER BY IDX DESC, REGIDATE DESC) A
    )
]]>
<include refid="pagingFooter" />
</select>

```

## -DB 선처리 SQL문 작성

```

<select id="priceStatistics" resultType="StatisticsVO">
SELECT
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '01' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '01' THEN item_price END) end jan,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '02' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '02' THEN item_price END) end feb,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '03' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '03' THEN item_price END) end mar,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '04' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '04' THEN item_price END) end apr,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '05' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '05' THEN item_price END) end may,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '06' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '06' THEN item_price END) end jun,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '07' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '07' THEN item_price END) end jul,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '08' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '08' THEN item_price END) end aug,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '09' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '09' THEN item_price END) end sep,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '10' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '10' THEN item_price END) end oct,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '11' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '11' THEN item_price END) end nov,
    CASE WHEN SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '12' THEN item_price END) is null then 0 else SUM(CASE WHEN SUBSTR(log_regdate,4,2) = '12' THEN item_price END) end dec
FROM csm_market_sellitem_log
WHERE item_sellerid = #{item_sellerId}
</select>

```

## 그 밖에 사용한 기술들은

- **인터셉터**를 이용한 로그인/아웃 처리 / IP 차단
- **AOP**를 이용 포인트 증감연산처리
- **트랜잭션**을 이용해 단일쿼리 실행 방지
- **AJAX**를 이용한 효율적 페이지 이동, 드래그 앤 드랍 파일업로드
- **SMTP**를 이용한 이메일 인증
- **SHA-256**을 사용해 비밀번호를 단방향 해시 암호화
- **POI**를 이용한 데이터 엑셀 다운로드
- 구글 어스를 이용한 데이터 차트
- CHARTIST.JS 를 이용한 데이터 차트
- 페아로를 이용한 페이스북으로 로그인
- 에디터로는 네이버 스마트에디터 2.0을 사용하였습니다.
- 쿠키를 이용한 자동 로그인 , 오늘 본 상품 생성

# 인터셉터를 이용한 로그인/아웃 처리/IP차단

```
//컨트롤러가 로그인관련처리를 마친후 세션에 로그인 관련데이터를 저장한다.
@Override
public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler,
    ModelAndView modelAndView) throws Exception {
    System.out.println("포스트핸드 호출");
    HttpSession session = request.getSession();
    ModelMap modelMap = modelAndView.getModelMap();
    Object loginDTO = modelMap.get("loginResult");

    if(loginDTO != null) {
        logger.info("new login success : 새로운 로그인 정보를 세션에 담습니다.");
        session.setAttribute(LOGIN, loginDTO);

        //체크박스 선택시 적용..
        if(request.getParameter("useCookie") != null) {
            logger.info("remember me.....");
            Cookie loginCookie = new Cookie("loginCookie", session.getId());
            loginCookie.setPath("/");
            // setMaxAge() : 초단위 시간동안 쿠키를 유지함 (60 * 60 * 24 * 7 ) = 7일
            loginCookie.setMaxAge(60 * 60 * 24 * 7);
            //쿠키를 전송시킨다.
            response.addCookie(loginCookie);
        }
        //response.sendRedirect("/");
        Object dest = session.getAttribute("dest");
        Object referer = session.getAttribute("Referer");
        session.removeAttribute("Referer");
        session.removeAttribute("dest");
        LoginDTO dto = (LoginDTO) session.getAttribute("login");
        if(dto.getUser_authority().equals("ADMIN")) {
            System.out.println("이드민 페이지로 리다이렉트 될 곳 입니다.");
            response.sendRedirect("/admin/adminmain");
        }else {
            response.sendRedirect(dest != null ? (String)dest:(String)referer);
        }
    }
}
```

```
public class FreeboardInterceptor extends HandlerInterceptorAdapter{
    @Inject
    FreeboardService service;

    @Override
    public boolean preHandle(HttpServletRequest req, HttpServletResponse res, Object object)
        throws Exception{
        List<String> list = service.SelectIp();
        for(String li : list) {
            if(req.getRemoteAddr().matches(li)) {
                if(service.IpCount(li)>5) {
                    res.setContentType("text/html; charset=UTF-8");
                    PrintWriter out = res.getWriter();
                    out.println("<script>alert('하루 가능한 글 작성개수는 5개입니다.');//history.go(-1);</script>");
                    return false;
                }
            }
        }
        return true;
    }
}
```

```
        dao.baseInTourderProgressUpdate(vo);  
    }  
}
```

# AOP를 이용 포인트 증감연산처리

```
@After("execution(* com.csm.community.service.FreeboardService.*regist(..))")  
public void freeboardRegistPoint(JoinPoint joinPoint) throws Throwable{  
  
    for ( Object o : joinPoint.getArgs() ) {  
        if ( o instanceof HttpServletRequest ) {  
            HttpSession session = ((HttpServletRequest) o).getSession();  
            LoginDTO dto = (LoginDTO) session.getAttribute("login");  
            UserVO vo = new UserVO();  
            vo.setUser_id(dto.getUser_id());  
            vo.setUser_point(50);  
            dao.pointPayment(vo);  
        }  
    }  
}  
  
@After("execution(* com.csm.community.service.ReplyService.*addReply(..))")  
public void freeboardReplyPoint(JoinPoint joinPoint) throws Throwable{  
  
    for ( Object o : joinPoint.getArgs() ) {  
        if ( o instanceof HttpServletRequest ) {  
            HttpSession session = ((HttpServletRequest) o).getSession();  
            LoginDTO dto = (LoginDTO) session.getAttribute("login");  
            UserVO vo = new UserVO();  
            vo.setUser_id(dto.getUser_id());  
            vo.setUser_point(5);  
            dao.pointPayment(vo);  
        }  
    }  
}
```

```
@After("execution(* com.csm.community.service.QuestionboardService.*regist(..))")  
public void questionboardRegisterPoint(JoinPoint joinPoint) throws Throwable{  
  
    for ( Object o : joinPoint.getArgs() ) {  
        if(o instanceof UserVO) {  
            UserVO vo = (UserVO) o;  
            dao.pointWithdraw(vo);  
        }  
    }  
}  
  
@After("execution(* com.csm.community.service.UserActionService.*insertAction(..))")  
public void userActionPoint(JoinPoint joinPoint) throws Throwable{  
  
    for ( Object o : joinPoint.getArgs() ) {  
        if(o instanceof UserVO) {  
            UserVO vo = (UserVO) o;  
            dao.pointPayment(vo);  
        }  
    }  
}
```

```
});  
};"
```

```
makeDir(uploadPath, yearPath, monthPath, datePath);  
logger.info(datePath);
```

# 트랜잭션을 이용해 단일쿼리 실행 방지

```
@Transactional  
@Override  
public void accept(Integer uno, ResumeVO vo) throws Exception {  
    //insert  
    dao.matching(vo);  
    //update  
    dao.accept(uno);  
}
```

```
@Transactional  
@Override  
public void itemRegister(ItemVO vo) throws Exception {  
    dao.itemRegister(vo);  
  
    String[] files = vo.GetFiles();  
    if(files == null) {return;}  
  
    for(String fileName : files) {  
        dao.addItemImgFile(fileName);  
    }  
}
```

```
@Transactional  
@Override  
public void addReply(ReplyVO vo, HttpServletRequest req) throws Exception{  
    dao.create(vo);  
    freedao.UpdateReplyCount(vo.getBno());  
}
```

```
@Transactional  
@Override  
public void orderProgressChange(OrderVO vo) throws Exception {  
    dao.orderProgressChange(vo);  
    dao.baseInfoOrderProgressUpdate(vo);  
}
```

```
@Transactional  
@Override  
public void orderProgressComplete(OrderDTO dto, String user_id, OrderVO orderVO) throws Exception {  
    //로그삽입  
    for(CartVO vo : dto.getOrderItems()) {  
        vo.setUser_id(user_id);  
        dao.sellItemLogInsert(vo);  
    }  
  
    //배송정보 업데이트  
    orderVO.setUser_id(user_id);  
    dao.orderProgressChange(orderVO);  
    dao.baseInfoOrderProgressUpdate(orderVO);  
}
```

# AJAX를 이용한 효율적 페이지 이동, 드래그 앤 드랍 파일업로드

```
$(".fileDrop").on("drop", function(event) {
    event.preventDefault();

    if(count>=5){
        alert("상품 이미지는 5개까지만 업로드 할 수 있습니다.");
        return;
    }
    count++;

    var files = event.originalEvent.dataTransfer.files;
    var file = files[0];

    var formData = new FormData();

    formData.append("file", file);
    // 파일전송시 processData: false, contentType: false를 반드시 해줘야함
    $.ajax({
        url: '/market/item/itemImgUpload',
        data: formData,
        dataType: 'text',
        processData: false,
        contentType: false,
        type: 'POST',
        success: function(data){

            var fileInfo = getFileInfo(data);

            var html = template(fileInfo);
            $(".uploadedList").append(html);
        }
    });
});
```

```
public static String uploadFile(String uploadPath, String originalName, byte[] fileData) throws Exception{
    UUID uid = UUID.randomUUID();
    String savedName = uid.toString()+"_"+originalName;
    //저장될 경로를 계산한다.
    String savedPath = calcPath(uploadPath);
    File target = new File(uploadPath+savedPath,savedName);

    //원본파일 저장
    FileCopyUtils.copy(fileData, target);

    //원본 파일의 확장자
    String formatName = originalName.substring(originalName.lastIndexOf(".")+1);
    String uploadedFileName = null;
    //이미지 파일이면 썸네일, 아니면 아이콘을 출력
    if(MimeMediaUtil.getMediaType(formatName) != null) {
        uploadedFileName = makeThumbnail(uploadPath, savedPath, savedName);
    }else {
        uploadedFileName = makeIcon(uploadPath, savedPath, savedName);
    }
    return uploadedFileName;
}

//년월일 생성
public static String calcPath(String uploadPath) {
    Calendar cal = Calendar.getInstance();

    String yearPath = "/" + cal.get(Calendar.YEAR);
    String monthPath = yearPath + "/" + new DecimalFormat("00").format(cal.get(Calendar.MONTH)+1);
    String datePath = monthPath + "/" + new DecimalFormat("00").format(cal.get(Calendar.DATE));
    makeDir(uploadPath, yearPath, monthPath, datePath);
    logger.info(datePath);
}
```

# SMTP를 이용한 이메일 인증

```
<!-- 이메일 인증용 빈 생성 -->
<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="sungpyo7014@gmail.com" />
    <property name="password" value="chqnwk121!" />

    <property name="javaMailProperties">
        <props>
            <prop key="mail.transport.protocol">smtp</prop>
            <prop key="mail.smtp.auth">true</prop>
            <prop key="mail.smtp.starttls.enable">true</prop>
            <prop key="mail.debug">true</prop>
        </props>
    </property>
</bean>

public class SignUpMail {
    public String signUp(String email, JavaMailSender mailSender) throws MessagingException, UnsupportedEncodingException {
        //System.out.println(mailSender);
        //System.out.println(email);
        String key = new TempKey().getKey(6, false);

        MailHandler sendMail = new MailHandler(mailSender);
        sendMail.setSubject("[CSM] 이메일 인증");
        sendMail.setText(new StringBuffer().append("<h1>CSM 이메일 인증 메일입니다.</h1>")
            .append("<span>인증번호 : " + key + "</span>")
            .append("<span style='color:red; font-size: 2em;'>" + key + "</span>")
            .append("을 인증번호란에 입력 해주세요."));

        sendMail.setFrom("csm@naver.com", "CSM member");
        sendMail.setTo(email);
        sendMail.send();

        return key;
    }
}
```

# SHA-256을 사용해 비밀번호를 단방향 해시 암호화

```
public class SHA256Util {  
    public String encryptSHA256(String str) {  
        String sha = "";  
        try {  
            MessageDigest sh = MessageDigest.getInstance("SHA-256");  
            sh.update(str.getBytes());  
            byte byteData[] = sh.digest();  
            StringBuffer sb = new StringBuffer();  
            for(int i= 0 ; i < byteData.length; i++) {  
                sb.append(Integer.toHexString((byteData[i]&0xff) + 0x100, 16).substring(1));  
            }  
            sha = sb.toString();  
        }catch (NoSuchAlgorithmException e) {  
            //e.printStackTrace();  
            System.out.println("Encrypt Error - NoSuchAlgorithmException");  
            sha = null;  
        }  
        return sha;  
    }  
}
```

# POI를 이용한 데이터 엑셀 다운로드

```
public class Excel {
    public void sellingItemDown(Map<String, Object> paramMap, HttpServletRequest request, HttpServletResponse response) {
        // Workbook 생성
        Workbook xlsWb = new HSSFWorkbook(); // Excel 2007 이전 버전
        //Workbook xlsWb = new XSSFWorkbook(); // Excel 2007 이상

        // *** Sheet
        // Sheet 생성
        Sheet sheet1 = xlsWb.createSheet("판매기록");

        // 월별 너비 설정
        sheet1.setColumnWidth(0, 5000);
        sheet1.setColumnWidth(1, 5000);
        sheet1.setColumnWidth(2, 5000);
        sheet1.setColumnWidth(3, 5000);
        sheet1.setColumnWidth(4, 5000);
        sheet1.setColumnWidth(5, 5000);
        sheet1.setColumnWidth(6, 5000);
        //____

        // *** Style
        // Cell 스타일 생성
        CellStyle cellStyle = xlsWb.createCellStyle();

        // 줄 바꿈
        cellStyle.setWrapText(true);

        // Cell 색깔, 무늬 채우기
        cellStyle.setFillForegroundColor(HSSFColor.LIME.getIndex());
        cellStyle.setFillPattern(CellStyle.BIG_SPOTS);

        Row row = null;
        Cell cell = null;
        //_____
        int count = 0;

        // 첫 번째 줄
        row = sheet1.createRow(0);
        cell = row.createCell(count++);
        cell.setCellValue("아이템번호");
        cell = row.createCell(count++);
        cell.setCellValue("이름");
        cell = row.createCell(count++);
        cell.setCellValue("점수(기준)");

        try {
            cell = row.createCell(count++);
            cell.setCellValue("구매일");
            cell = row.createCell(count++);
            cell.setCellValue("구매자");
            // 첫 번째 줄에 cell 설정하기-----
            List<SellItemLogVO> list = (List<SellItemLogVO>).paramMap.get("list");
            count=0;
            int i =1;
            for(SellItemLogVO vo : list) {
                row = sheet1.createRow(i++);
                cell = row.createCell(count++);
                cell.setCellValue(vo.getItem_idx());
                cell = row.createCell(count++);
                cell.setCellValue(vo.getItem_name());
                cell = row.createCell(count++);
                cell.setCellValue(vo.getItem_price());
                cell = row.createCell(count++);
                cell.setCellValue(vo.getItem_amount());
                cell = row.createCell(count++);
                cell.setCellValue(vo.getItem_sellerId());
                cell = row.createCell(count++);
                String regdate = vo.getLog_regdate().toString();
                cell.setCellValue(regdate);
                cell = row.createCell(count++);
                cell.setCellValue(vo.getUser_id());
                count = 0;
            }
            // excel 파일 저장
            UUID uid = UUID.randomUUID();
            String savedName = "SellList_"+uid.toString();
            File xlsFile = new File("C:/"+savedName+".xls");
            FileOutputStream fileOut = new FileOutputStream(xlsFile);
            xlsWb.write(fileOut);
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

# 구글 어스를 이용한 데이터 차트

```
<select id="resumeStatistics" resultType="StatisticsVO">
    SELECT
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '01', 1)) jan,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '02', 1)) feb,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '03', 1)) mar,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '04', 1)) apr,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '05', 1)) may,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '06', 1)) jun,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '07', 1)) jul,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '08', 1)) aug,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '09', 1)) sep,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '10', 1)) oct,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '11', 1)) nov,
        COUNT(DECODE(TO_CHAR(regidate, 'MM'), '12', 1)) dec
    FROM csm_resume
    WHERE id = #{id}
</select>
```

# 구글 어스를 이용한 데이터 차트

```
<script type="text/javascript">
google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
  var data = google.visualization.arrayToDataTable([
    ['month', '내가 쓴 이력서', '열람한 이력서'],
    ['1월', parseInt('${resume.jan}'), parseInt('${read.jan}')],
    ['2월', parseInt('${resume.feb}'), parseInt('${read.feb}')],
    ['3월', parseInt('${resume.mar}'), parseInt('${read.mar}')],
    ['4월', parseInt('${resume.apr}'), parseInt('${read.apr}')],
    ['5월', parseInt('${resume.may}'), parseInt('${read.may}')],
    ['6월', parseInt('${resume.jun}'), parseInt('${read.jun}')],
    ['7월', parseInt('${resume.jul}'), parseInt('${read.jul}')],
    ['8월', parseInt('${resume.aug}'), parseInt('${read.aug}')],
    ['9월', parseInt('${resume.sep}'), parseInt('${read.sep}')],
    ['10월', parseInt('${resume.oct}'), parseInt('${read.oct}')],
    ['11월', parseInt('${resume.nov}'), parseInt('${read.nov}')],
    ['12월', parseInt('${resume.dec}'), parseInt('${read.dec}')]
  ]);

  var options = {
    title: '월별 이력서 통계'
  };

  var chart = new google.visualization.ColumnChart(document.getElementById('chart_div'));
  chart.draw(data, options);
}
</script>
```

# CHARTIST.JS 를 이용한 데이터 차트

```
<select id="deleteStatistics" resultType="StatisticsVO">
    SELECT
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '01', 1)) jan,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '02', 1)) feb,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '03', 1)) mar,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '04', 1)) apr,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '05', 1)) may,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '06', 1)) jun,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '07', 1)) jul,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '08', 1)) aug,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '09', 1)) sep,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '10', 1)) oct,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '11', 1)) nov,
        COUNT(DECODE(TO_CHAR(log_regdate, 'MM'), '12', 1)) dec
    FROM csm_user_log
    WHERE log_action = 'D'
</select>
```

# CHARTIST.JS 를 이용한 데이터 차트

```
<select id="contrastDeleteStatistics" resultType="MonthStatisticsVO">
    select b.* , case when b.result &lt; 0 then '감소' else '증가' end as check_status
    from (
        select a.* , round((a.this - a.last) / a.last) * 100, 1) result
        from (
            select
                count(case when to_char(substr(log_regdate, 4, 2), '00') = to_char(substr(sysdate, 4, 2)-1, '00') then log_action end) last,
                count(case when to_char(substr(log_regdate, 4, 2), '00') = to_char(substr(sysdate, 4, 2), '00') then log_action end) this
            from csm_user_log
            where log_action = 'D') a) b
</select>
```

# CHARTIST.JS 를 이용한 데이터 차트

```
var dataEmailsSubscriptionChart3 = {
    labels: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
    series: [
        [parseInt('${resume.jan }'), parseInt('${resume.feb }'), parseInt('${resume.mar }'), parseInt('${resume.apr }'), parseInt('${resume.may }'),
         parseInt('${resume.jun }'), parseInt('${resume.jul }'), parseInt('${resume.aug }'), parseInt('${resume.sep }'), parseInt('${resume.oct }'),
         parseInt('${resume.nov }'), parseInt('${resume.dec }')]
    ]
};
var optionsEmailsSubscriptionChart3 = {
    axisX: {
        showGrid: true
    },
    chartPadding: {
        top: 0,
        right: 5,
        bottom: 0,
        left: 20
    }
};
var responsiveOptions3 = [
    ['screen and (max-width: 640px)', {
        seriesBarDistance: 5,
        axisX: {
            labelInterpolationFnc: function(value) {
                return value[0];
            }
        }
    }]
];
var emailsSubscriptionChart3 = Chartist.Bar('#emailsSubscriptionChart3', dataEmailsSubscriptionChart3, optionsEmailsSubscriptionChart3, responsiveOptions3);

//start animation for the Emails Subscription Chart
md.startAnimationForBarChart(emailsSubscriptionChart3);
```

# 페아로를 이용한 페이스북으로 로그인

```
<!-- 페이스북으로 로그인 시작 -->
<script>
  function statusChangeCallback(response) {
    console.log('statusChangeCallback');
    console.log(response);
    if (response.status === 'connected') {
      if('${logout}' === 'facebook'){
        FB.logout(function(response) {});
        location.href='/user/login';
      }
      testAPI();
    } else {
    }
  }

  function checkLoginState() {
    FB.getLoginStatus(function(response) {
      statusChangeCallback(response);
    });
  }

  window.fbAsyncInit = function() {
    FB.init({
      appId      : '164346997682226',
      cookie     : true, // enable cookies to allow the server to access
                       // the session
      xfbml      : true, // parse social plugins on this page
      version   : 'v2.8' // use graph api version 2.8
    });
    FB.getLoginStatus(function(response) {
      statusChangeCallback(response);
    });
  };
</script>

<fb:login-button scope="public_profile,email" onlogin="checkLoginState();">
</fb:login-button>

<div id="status">
</div>
<!-- 페이스북으로 로그인 끝 -->
```

# 에디터로는 네이버 스마트에디터 2.0을 사용하였습니다.

```
<script type="text/javascript" src="/resources/include/editor/js/HuskyEZCreator.js" charset="utf-8"></script>
```

```
<script>
$(function(){
    //전역변수
    var obj = [];
    //스마트에디터 프레임생성
    nhn.husky.EZCreator.createInIFrame({
        oAppRef: obj,
        elPlaceHolder: "editor",
        sSkinURI: "/resources/include/editor/SmartEditor2Skin.html",
        htParams : {
            // 툴바 사용 여부 (true:사용/ false:사용하지 않음)
            bUseToolbar : true,
            // 입력창 크기 조절바 사용 여부 (true:사용/ false:사용하지 않음)
            bUseVerticalResizer : true,
            // 모드 탭(Editor | HTML | TEXT) 사용 여부 (true:사용/ false:사용하지 않음)
            bUseModeChanger : true,
        }
    });
    //전송버튼
    $("#register").click(function(){
        //id가 smarteditor인 textarea에 에디터에서 대입
        obj.getById["editor"].exec("UPDATE_CONTENTS_FIELD", []);
        //폼 submit
        $("#frm").submit();
    })
})
</script>
```

# 쿠키를 이용한 자동 로그인

```
//체크박스 선택시 적용..
if(request.getParameter("useCookie") != null) {

    logger.info("remember me.....");
    Cookie loginCookie = new Cookie("loginCookie", session.getId());
    loginCookie.setPath("/");
    // setMaxAge() : 초단위 시간동한 쿠키를 유지함 (60 * 60 * 24 * 7 ) = 7일
    loginCookie.setMaxAge(60 * 60 * 24 * 1);
    //쿠키를 전송시킨다.
    response.addCookie(loginCookie);
}

Cookie loginCookie = WebUtils.getCookie(request, "loginCookie");
//쿠키가 존재한다면,
if(loginCookie != null) {

    LoginDTO dto = service.checkLoginBefore(loginCookie.getValue());
    logger.info("기존에 있던 사용자: "+dto);
    //DB에 기록이 있다면..
    if(dto != null) {

        session.setAttribute("login", dto);
        return true;
    }
}

Cookie loginCookie = WebUtils.getCookie(req, "loginCookie");
if ( loginCookie != null ){

    // 쿠키 유효시간을 0으로 설정
    loginCookie.setMaxAge(0);
    // 쿠키 설정을 적용한다.
    response.addCookie(loginCookie);

    //로그아웃시 시간 재설정
    Date date = new Date(System.currentTimeMillis());
    service.keepLogin(dto.getUser_id(), session.getId(), date);
}
```

# 쿠키를 이용한 오늘 본 상품 생성

```
/*
 * @description 특정 key의 쿠키값을 List로 반환한다.
 * @params key: 쿠키 이름
 */
public List<String> getValueList(String key, HttpServletRequest request) throws UnsupportedEncodingException{
    Cookie[] cookies = request.getCookies();
    String[] cookieValues = null;
    String value = "";
    List<String> list = null;

    // 특정 key의 쿠키값을 ","로 구분하여 String 배열에 담아준다.
    // ex) 쿠키의 key/value --> key = "clickItems", value = "211, 223, 303"(상품 번호)
    if(cookies != null){
        for(int i=0;i<cookies.length;i++){
            if(cookies[i].getName().equals(key)){
                value = cookies[i].getValue();
                cookieValues = (URLDecoder.decode(value, encoding)).split(",");
                break;
            }
        }
    }

    // String 배열에 담겼던 값을 List로 다시 담는다.
    if(cookieValues != null){
        list = new ArrayList<String>(Arrays.asList(cookieValues));

        if (list.size() > 3) { // 값이 3개를 초과하면, 최근 것 3개만 담는다.
            int start = list.size() - 3;
            List<String> copyList = new ArrayList<String>();
            for (int i = start ; i < list.size() ; i++) {
                copyList.add(list.get(i));
            }
            list = copyList;
        }
    }
    return list;
}

/**
 * @description 쿠키를 생성 혹은 업데이트한다.
 * @params key: 쿠키 이름, value: 쿠키 이름과 짹을 이루는 값, day: 쿠키의 수명
 */
public void setCookie(String key, String value, int day, HttpServletRequest request, HttpServletResponse response) throws UnsupportedEncodingException{
    List<String> list = getValueList(key, request);
    String sumValue = "";
    int equalsValueCnt = 0;

    if(list != null){
        for(int i=0; i<list.size(); i++){
            sumValue += list.get(i) + ",";
            if(list.get(i).equals(value)){
                equalsValueCnt++;
            }
        }
        if(equalsValueCnt != 0){ // 같은 값을 넣으려고 할 때의 처리
            if(sumValue.substring(sumValue.length()-1).equals(",")){
                sumValue = sumValue.substring(0, sumValue.length()-1);
            }
        }else{
            sumValue += value;
        }
        else{
            sumValue = value;
        }

        if(!sumValue.equals("")){
            Cookie cookie = new Cookie(key, URLEncoder.encode(sumValue, encoding));
            cookie.setMaxAge(60*60*24*day);
            cookie.setPath(path);
            response.addCookie(cookie);
        }
    }
}
```

# 쿠키를 이용한 오늘 본 상품 생성

```
//오늘본상품
@RequestMapping(value="/dayListAjax", method=RequestMethod.GET)
public void dayList(HttpServletRequest request, Model model) throws Exception{
    CookieUtils cookie = new CookieUtils();
    List<String> temp = cookie.getValueList("pageCookie", request);
    if(temp == null) {
        return;
    }
    List<ItemVO> list = new ArrayList<>();
    for(String tmp: temp) {
        Map<String, Object> paramMap = service.itemReadPage(tmp);
        ItemVO listTemp = (ItemVO) paramMap.get("baseInfo");
        if(listTemp.getFiles()!=null) {
            String photoNames[] = listTemp.getFiles();
            for(int i=0; i<photoNames.length ;i++) {
                photoNames[i] = downPath(request)+photoNames[i];
            }
            listTemp.setFiles(photoNames);
        }
        list.add(listTemp);
    }
    model.addAttribute("baseInfo", list);
}
```

# 프로젝트 시연

## 느낀점

1. 합치는 과정에서 많은 시간을 소모
2. 팀프로젝트 경험 부족
3. 생각과 이상에 대한 현실

*End*