《Web应用开发技术》课程设计

经济与管理学院教师绩效管理信息系统

学生姓名1：林 华 栋

学生姓名2：高玉林

学 号1：091401323

学 号2：091402323

专 业：信息管理与信息系统

班 级：0914201班

指导老师：罗正军

2016年12月13日星期二

目录

1. [**需求背景** 3](#_Toc469500538)
2. [**系统概述** 4](#_Toc469500539)
3. [**用户需求分析** 4](#_Toc469500540)
4. [**数据库设计** 4](#_Toc469500541)
5. [**功能模块设计** 4](#_Toc469500542)

[(一) 用户角色分类 4](#_Toc469500543)

[(二) 教师绩效管理 5](#_Toc469500544)

[1. 科研绩效管理 5](#_Toc469500545)

[2. 教学绩效管理 5](#_Toc469500546)

[3. 公益绩效管理 6](#_Toc469500547)

[(三) 教师绩效审核 6](#_Toc469500548)

[1. 一级审核（指定管理员审核） 6](#_Toc469500549)

[2. 二级审核 6](#_Toc469500550)

[(四) 系统设置 7](#_Toc469500551)

[1. 基础设置 7](#_Toc469500552)

[2. 科研设置 7](#_Toc469500553)

[3. 教学设置 7](#_Toc469500554)

[4. 用户管理 7](#_Toc469500555)

[(五) 数据导出 7](#_Toc469500556)

[1． 科研数据导出 7](#_Toc469500557)

[2． 教学数据导出 7](#_Toc469500558)

[3． 公益数据导出 7](#_Toc469500559)

1. [**核心代码** 7](#_Toc469500560)
2. **需求背景**

为了推动经济与管理学院的办公信息化的发展进程，更好地对学院的教师在科学研究、基础教学以及公益活动等方面绩效进行有效地信息化管理，以及提高办公效率，方便教师的绩效录入工作，摆脱繁重的Excel汇总工作，因此决定开发一个基于B/S架构的绩效管理系统。该系统拟定于2016年年末完成。

**二. 系统概述**

综合考虑各方面的需求因素以及技术因素，本系统是基于Java语言并且在B/S架构上开发的一个web应用软件，采用Apache公司的 Tomcat web服务器，数据库采用的是Microsoft SQLServer 2008R2数据库系统软件，开发过程中所用的IDE（集成开发环境）是Genuitec开发商的MyEclipse，前端框架采用的是Apache的一个开源web框架—Struts2，其版本是2.3，后台的持久层的开发采用JBoss的开源ORM框架—Hibernate，其对应的版本是4.1.4，项目开发过程中采用GitHub作为版本控制工具。由于Java语言本身的性质，以及hibernate框架本身对于数据库的查询在某些情况下缓存会比较大，所以对服务器系统的内存的大小会有相应的容量需求。本系统开发完成后初定在经济与管理学院的网站上开辟一个链接，重定向到本系统的主界面，完成系统的上线工作。

**三．用户需求分析**

用户需求主要有以下几个方面：

1. 基础操作与用户管理:基础操作包括登录与退出系统，用户在登录时系统识别身份为一般教师、子系统管理员或系统管理员，根据不同教师的身份显示不尽相同的界面；用户管理包括一般教师用户信息修改、密码找回，系统管理员的教师管理：增加教师，删除教师，更改教师权限等。
2. 系统设置：分为科研设置、教学设置和基础设置。科研分为八个团队项目设置，每个团队项目对应项目类别、级别和评分设置，可以由系统管理员进行增加、修改和删除；教学分为课堂教学、教学研究和创新项目三个部分，每个部分下属多个模块分别对应相关设置，总计22个模块，系统管理员可对每个模块的级别分数、类别分数等进行设置；基础设置包括学期设置、国籍设置、系设置、研究所设置和公益管理员设置，其中系（研究所）设置可对系（研究所）基础属性设置，也可以同时设置系主任（研究所所长）。
3. 绩效添加与审核：绩效添加方面，普通教师用户团队或者个人手动添加科研、教学相关模块绩效，公益绩效则为个人手动添加。其中，团队方面由团队负责人添加项目，项目参与人之后手动选择项目加入；绩效审核方面，系主任（研究所所长/公益管理员）首先审核所属系（研究所/个人）的教学（科研/公益）方面的团队项目绩效，大管理员则审核所有系和研究所方面的个人绩效和团队成员绩效。
4. 数据导出：用户要求大管理员可以根据需要（指定时间范围、学期、系别或者研究所）从数据库中导出所需要的科研、教学或公益部分或者全部绩效数据成为Excel表格。

**四．数据库设计**

**五．功能模块设计**

1. **用户角色分类**

根据需求分析，本系统用户类型在宏观上分为两类用户，分别是：普通用户和管理员用户，然后细化下去，普通用户中细化出四类用户，分别是：普通教师用户，系主任，研究所所长以及公益管理员。其中系主任、研究所所长以及公益管理员在身份上也都属于教师用户。其具体的角色分类关系如下图所示：

|  |
| --- |
| 用户角色  管理员  普通用户  系主任  普通教师用户  研究所所长  公益管理员  图：3.1用户角色结构图 |

1. **教师绩效管理**
2. **科研绩效管理**

科研项目绩效功能使用的用户分为两类，一类是项目负责人，另外一类是项目参与人，对于项目负责人而言，可以添加项目信息，编辑项目信息，查看项目成员信息，更新项目信息，删除项目信息，查看项目的审核状态；对于项目参与人而言，可以选择加入项目。

1. **教学绩效管理**

教学绩效绩效功能的使用与科研绩效管理模块功能的使用大致相似，区别在于，科研整体全都是项目制，而教学绩效这一个模块一共18个子模块，其中8个是项目制子模块，每个项目都有一个负责人，另外10个子模块是个人制，也就是教师既是负责人也是唯一参与人。

对于项目制子模块而言，每个子模块绩效的录入规则如下，由每个项目的负责人将项目信息录入系统，然后相应的项目里的成员教师登录系统进行确认加入，项目负责人可以新增项目信息，修改项目信息，删除项目信息，查看本项目的成员信息以及审核状态；对于项目成员，可以确认加入本项目。

1. **公益绩效管理**

公益绩效功能使用的用户为全体教师，属于个人绩效，公益绩效管理模块共4个子模块，分别是活动参与，活动新增，活动发布和活动缺席。活动新增需填写活动说明，活动发布即为活动负责人，教师个人根据实际情况准确选择已参与/缺席的活动进行填写，确认加入/缺席活动。准确选择公益活动过程提供了日期对已开展活动进行检索，活动缺席需填写活动缺席原因，活动发布需填写活动简介。

1. **教师绩效审核**
2. **一级审核（指定管理员审核）**
   * 1. **科研绩效审核**

对于从属于项目体制的绩效而言，由对应的**指定管理员**对项目进行审核，然后由系统管理员针对项目里的参与成员进行审核。科研项目绩效从属于项目体制类绩效，所以也同样满足上述业务逻辑，完成此项审核工作的指定管理员是研究所所长。

对于尚未被审核的项目的状态，本系统设置成4种状态：待完善、待审核、审核通过、未通过审核。其中，只有项目处于“待审核”状态的时候，研究所所长才能对其进行审核，如果项目处于“待完善”的状态，此时研究所所长不能对其进行审核，除此以外，研究所所长可以查看已经审核通过的项目和未通过审核的项目。另外：只有项目审核通过，对应项目的成员才能进行二级审核（项目审核由指定管理员审核，而项目成员的审核由系统管理员进行审核），如果项目没有通过一级审核，那么该项目的所有成员**没有**机会进行二级审核，也就是说，项目审核不通过，默认对应项目的全部参与教师全**不**通过。

* + 1. **教学绩效审核**

教学绩效的审核与科研审核大致相同，区别在于教学绩效分为从属于项目体制类绩效和从属于个人类绩效，对于项目体制的绩效的相关操作的说明请参照科研绩效审核。对于个人类的绩效，只需要系统管理员进行审核，不需要经过指定管理员进行审核。个人类的绩效分为3种状态：待审核、未通过审核、管理员审核通过；系统管理员只能在“待审核”状态下进行个人绩效审核，决定通过与否。同时提供给系统管理员系别选择、学期选择等记录筛选条件。

* + 1. **公益绩效审核**

公益绩效审核不同于科研教学审核，公益管理员直接负责所有用户的公益绩效审核，没有上一级的系统管理员审核。用户在公益绩效中添加绩效，即到达审核部分。审核状态分为3种状态：待审核、审核通过和审核不通过。公益管理员可以查看所有已审核、未审核和审核不通过的个人绩效。记录筛选条件有系别选择、活动时间范围选择等筛选条件供公益管理员对特定活动进行选择，进行统一审核。

1. **二级审核**

二级审核主要针对科研项目以及教学绩效审核，公益绩效的审核全权由公益管理员进行审核，对于团队项目（科研和教学都是如此），如果项目审核通过，那么系统管理员这边相应的项目里的成员对应的记录此时是“所长/系主任审核通过”的状态，说明项目已经通过，如果项目没有通过，对应的项目成员在系统管理员审核界面直接显示“未通过审核”，如果显示待审核，说明此时系主任或者研究所所长尚未审核对应的项目。系统管理员要进行审核的时候，只能选择下拉框中的“所长/系主任审核通过”状态才能进行审核。

1. **系统设置**
2. **基础设置**

**系设置**：设置系的编号，系名称，系管理员的编号，系管理员姓名；修改系的设置信息；删除系的设置信息；增加系的设置信息。

**研究所设置**：主要设置研究所的名称以及研究所所长。另外，可以增加研究所，修改研究所信息，删除研究所信息。

**国籍设置**：设置国籍有关信息，主要是针对外国专家的国籍进行设置。

**学期设置**：对学期信息进行动态更新，以适应每年的绩效的录入工作。

**公益管理员设置**：增加、删除、修改公益管理员的有关信息。

1. **科研设置**
   * 1. **期刊论文设置**

期刊论文设置主要针对期刊论文以下几个方面的基础信息进行设置：期刊类别设置、期刊表设置、以及期刊评分设置。

* + 1. **学术著作设置**

学术著作设置主要针对学术著作一下几个方面的基础信息进行设置：字数类别设置、出版社类别设置、出版社设置、学术著作评分设置。

* + 1. **科研项目设置**

科研项目设置主要针对以下几个方面基础信息进行设置：科研项目类别设置、科研项目评分设置。

* + 1. **科研项目奖励设置**

科研项目奖励设置主要针对以下几个方面基础信息进行设置：科研项目奖励类别设置、科研项目奖励级别设置、科研项目奖励评分设置。

* + 1. **参加学术会议设置**

参加学术会议设置主要针对以下几个方面的基础信息进行设置：学术会议类别设置、论文检索类别设置、学术会议地点设置、学术会议评分设置。

* + 1. **邀请专家讲学设置**

邀请专家讲学设置主要针对以下几个方面的基础信息进行设置：专家类别设置、国籍设置、邀请专家讲学评分设置。

* + 1. **主承办学术会议设置**

主承办学术会议设置主要针对以下几个方面的基础信息进行设置：学术会议类别设置、学术会议地点设置、学术会议评分设置。

* + 1. **入选人才项目工程设置**

入选人才项目工程设置主要针对以下几个方面的基础信息进行设置：人才工程设置、人才工程评分设置。

* + 1. **本人承担任务设置**

该基础信息设置子模块没有响应的子模块。

1. **教学设置**
2. **课堂教学**

课堂教学主要针对以下几个方面的基础信息进行设置：总站堂时系数设置、综合教学评估设置、论文指导系数设置、教学竞赛系数设置、教学能力提升设置、优秀教师团队称号系数设置。

1. **教学研究**

教学研究主要针对以下几个方面的基础信息进行设置：教学研究成果设置、论文发表等级设置、教学成果等级设置、教材编写等级设置、精品课程等级设置、品牌专业等级设置、企业培养基地设置、暑期课程设置。

1. **创新项目**

创新项目主要针对以下几个方面的基础信息进行设置：创新项目等级设置、学生竞赛指导等级设置、学生竞赛等级设置、竞赛获奖等级设置、竞赛级别设置、本科生导师等级设置、校外实践等级设置。

1. **用户管理**

此设置可以添加、删除、更新用户信息。

1. **数据导出**

数据导出主要是针对科研模块的八个子模块，教学模块的十八个子模块以及公益管理数据的Excel数据表的导出。

1. **科研模块的数据导出**

科研模块的数据导出按照研究所类别、项目中相关日期范围以及八个子模块的分类来进行数据的分类导出对应模块的全部数据。

1. **教学模块的数据导出**

教学模块的数据导出按照系别、项目或个人的绩效对应的学期以及教学统共的18个子模块的分类进行对应模块的数据导出。

1. **公益模块的数据导出**

公益模块的数据导出包括导出指定日期范围内的活动列表，根据时间范围、系（管工、经贸、工商）、活动类别（选择性活动、指定活动、其他活动）以及

活动日期找到单个活动并对单个活动的参与信息和缺席信息同时进行导出

**六．核心代码**

1. **hibernate核心配置文件以及数据源的配置。**

|  |
| --- |
| <?xml version=*'1.0'* encoding=*'UTF-8'*?>  <!DOCTYPE hibernate-configuration PUBLIC  "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  <!-- Generated by MyEclipse Hibernate Tools. -->  <hibernate-configuration>  <!-- HQL-Querylanguage http://www.tutorialspoint.com/hibernate/hibernate\_query\_language.htm -->  <session-factory>  <property name=*"dialect"*>  org.hibernate.dialect.SQLServerDialect  </property>  <property name=*"connection.url"*>  jdbc:sqlserver://localhost:1433;  DatabaseName=ScientificResearch  </property>  <property name=*"connection.username"*>sa</property>  <property name=*"connection.password"*>123456</property>  <property name=*"connection.driver\_class"*>  com.microsoft.sqlserver.jdbc.SQLServerDriver  </property>  <property name=*"myeclipse.connection.profile"*>  SqlServerDriver  </property>  <property name=*"show\_sql"*>true</property>  <property name=*"format\_sql"*>true</property>  <property name=*"hbm2ddl.auto"*>update</property>  <property name=*"hibernate.connection.provider\_class"*>org.hibernate.service.jdbc.connections.internal.C3P0ConnectionProvider</property>  <property name=*"hibernate.c3p0.acquire\_increment"*>2</property>  <property name=*"hibernate.c3p0.idle\_test\_period"*>2000</property>  <property name=*"hibernate.c3p0.max\_size"*>25</property>  <property name=*"hibernate.c3p0.min\_size"*>5</property>  <property name=*"c3p0.max\_statements"*>10</property>  <property name=*"hibernate.c3p0.timeout"*>2000</property>  ……………此处省略hibernate model的配置  <session-factory> |

1. **登录拦截器业务代码**

|  |
| --- |
| **public** String intercept(ActionInvocation ai) **throws** Exception {  // **TODO** Auto-generated method stub  String actionName = ai.getProxy().getActionName();  Map<String, Object>session = ai.getInvocationContext().getSession();  Teacher teacher = (Teacher)session.get("teacher");  String teacherLevel = (String)session.get("teacherLevel");  String str = "relogin";  **if**("login".equals(actionName)||(teacher!=**null**&&actionName.substring(0, 2).equals(teacherLevel))){  str = ai.invoke();  }  **else**{  HttpServletRequest req = ServletActionContext.*getRequest*();  HttpSession sessions = req.getSession();  sessions.invalidate();  req.setAttribute("loginstatus","抱歉，登陆已失效，请重新登陆！");  }  **return** str;  } |

1. **项目的持久层示例代码（以科研项目为例）**

|  |
| --- |
| **public** **void** save(ScientificResearchProject transientInstance) {  ***log***.debug("saving ScientificResearchProject instance");  **try** {  getSession().save(transientInstance);  ***log***.debug("save successful");  } **catch** (RuntimeException re) {  ***log***.error("save failed", re);  **throw** re;  }  }  **public** **void** delete(ScientificResearchProject persistentInstance) {  ***log***.debug("deleting ScientificResearchProject instance");  **try** {  getSession().delete(persistentInstance);  ***log***.debug("delete successful");  } **catch** (RuntimeException re) {  ***log***.error("delete failed", re);  **throw** re;  }  } |

1. **绩效的CRUD操作的示例代码（以科研项目为例）**

|  |
| --- |
| **public** **void** save(TeacherAndscientificResearchProject transientInstance) {  ***log***.debug("saving TeacherAndscientificResearchProject instance");  **try** {  getSession().save(transientInstance);  ***log***.debug("save successful");  } **catch** (RuntimeException re) {  ***log***.error("save failed", re);  **throw** re;  }  }  **public** **void** delete(TeacherAndscientificResearchProject persistentInstance) {  ***log***.debug("deleting TeacherAndscientificResearchProject instance");  **try** {  getSession().delete(persistentInstance);  ***log***.debug("delete successful");  } **catch** (RuntimeException re) {  ***log***.error("delete failed", re);  **throw** re;  }  }  **public** TeacherAndscientificResearchProject merge(  TeacherAndscientificResearchProject detachedInstance) {  ***log***.debug("merging TeacherAndscientificResearchProject instance");  **try** {  TeacherAndscientificResearchProject result = (TeacherAndscientificResearchProject) getSession()  .merge(detachedInstance);  ***log***.debug("merge successful");  **return** result;  } **catch** (RuntimeException re) {  ***log***.error("merge failed", re);  **throw** re;  }  } |

1. **绩效审核示例代码（以科研项目为例）**

|  |
| --- |
| **public** **boolean** updateCheckoutStatus(  List<TeacherAndscientificResearchProject> TARProList) {  Session session = **this**.getSession();  Transaction tx = **null**;  **boolean** updateFlag = **false**;  **try** {  **for** (**int** i = 0; i < TARProList.size(); i++) {  session.update(TARProList.get(i));  }  tx = session.beginTransaction();  tx.commit();  updateFlag = **true**;  } **catch** (Exception ex) {  ex.printStackTrace();  }  **return** updateFlag;  } |

1. **数据导出部分的示例代码（以科研项目为例）**

|  |
| --- |
| **public** **class** ClassTeachExcel {  /\*\*  \* function：导出课堂教学数据  \* **@param** ths 导出数据的字段  \* **@param** tfClassTeachPfomsList Excel表对应的数据实体类  \* **@param** departmentName 研究所名称  \* **@param** foredate 下限日期  \* **@param** afterdate 上限日期  \* **@return** HSSFWorkbook  \*/  @SuppressWarnings("deprecation")  **public** **static** HSSFWorkbook generateExcel(String[] ths,List<TfclassTeachPefromanceUnionTfterm> tfClassTeachPfomsList,String researchLabName,String foredate,String afterdate){  HSSFWorkbook wb = **new** HSSFWorkbook();  TftermDAO tftermDAO=**new** TftermDAO();  HSSFSheet sheet = wb.createSheet(tftermDAO.findById(foredate).getTerm()+"-"+tftermDAO.findById(afterdate).getTerm());  //设置列宽  **for**(**int** i=0;i<ths.length;i++){  sheet.setColumnWidth((**short**) i, 5000);  }  // 合并单元格 org.apache.poi.hssf.util.Region(short rowFrom,short colFrom,short rowTo,short colTo)  org.apache.poi.hssf.util.Region reg = **new** org.apache.poi.hssf.util.Region((**short**)0,(**short**)0,(**short**)1,(**short**)(ths.length-1));  sheet.~~addMergedRegion~~(reg);  //创建预备居中格式  HSSFCellStyle cellStyle = wb.createCellStyle();  //创建预备居中大字体格式  HSSFCellStyle cellStyleforFonts = wb.createCellStyle();  //设置格式居中  cellStyle.setAlignment(CellStyle.***ALIGN\_CENTER***);  cellStyleforFonts.setAlignment(CellStyle.***ALIGN\_CENTER***);  //创建字体  HSSFFont font = wb.createFont();  //设置字体大小  font.setFontHeightInPoints((**short**)14);  //合并字体到 居中大字体格式  cellStyleforFonts.setFont(font);  **int** rownum = 0;  HSSFRow row = sheet.createRow((**short**) rownum++);  HSSFCell cells[] = **new** HSSFCell[1];  cells[0] = row.createCell(0);  cells[0].setCellValue("课堂教学["+researchLabName+"]");  //应用格式  cells[0].setCellStyle(cellStyleforFonts);  rownum++;  row = sheet.createRow((**short**) rownum++);  HSSFCell cell[] = **new** HSSFCell[ths.length];  **for**(**int** i=0;i<ths.length;i++){  cell[i] = row.createCell(i);  cell[i].setCellValue(ths[i]);  cell[i].setCellStyle(cellStyle);  }  **if**(tfClassTeachPfomsList!=**null**){  **for**(**int** i=0;i<tfClassTeachPfomsList.size();i++){  row = sheet.createRow(rownum++);  cell = **new** HSSFCell[ths.length];  **for**(**int** p=0;p<ths.length;p++){  cell[p] = row.createCell(p);  }  /\*\*  \* 导出字段：  \* {"课堂绩效编号","总课时","综合教学评估结果","当前教师工号","当前教师姓名","绩效成绩","学期"};  \*/  cell[0].setCellValue(tfClassTeachPfomsList.get(i).getClassTeachPerformance().getClassPefromanceId());  cell[0].setCellStyle(cellStyle);  cell[1].setCellValue(tfClassTeachPfomsList.get(i).getClassTeachPerformance().getSumtime());  cell[1].setCellStyle(cellStyle);  cell[2].setCellValue(tfClassTeachPfomsList.get(i).getClassTeachPerformance().getTfclassTeachEvaluation().getEvaluResult());  cell[2].setCellStyle(cellStyle);  cell[3].setCellValue(tfClassTeachPfomsList.get(i).getClassTeachPerformance().getTeacher().getTeacherId());  cell[3].setCellStyle(cellStyle);  cell[4].setCellValue(tfClassTeachPfomsList.get(i).getClassTeachPerformance().getTeacher().getTeacherName());  cell[4].setCellStyle(cellStyle);  cell[5].setCellValue(tfClassTeachPfomsList.get(i).getClassTeachPerformance().getFinalScore());  cell[5].setCellStyle(cellStyle);  cell[6].setCellValue(tfClassTeachPfomsList.get(i).getCurrentTerm().getTerm());  cell[6].setCellStyle(cellStyle);  }  }  **return** wb;  }  } |

1. **系统设置示例代码（以科研项目为例）**

|  |
| --- |
| /\*\*  \* 删除一条科研项目设置的记录  \*  \* **@throws** Exception  \*/  **public** **void** delProjectType() **throws** Exception {  **this**.getProType().setSpareTire("0");  Transaction tx = **null**;  **try** {  **this**.proTypeDAO.merge(**this**.proType);  tx = **this**.proTypeDAO.getSession().beginTransaction();  tx.commit();  ServletActionContext.*getResponse*().getWriter().write("succ");  } **catch** (Exception e) {  // **TODO**: handle exception  tx.rollback();  **throw** e;  }  }  /\*\*  \* 更新一条科研项目名称  \*/  **public** **void** updateProjectType() **throws** Exception {  proType.setSpareTire("1");  Transaction tx = **null**;  **try** {  proTypeDAO.merge(proType);  tx = proTypeDAO.getSession().beginTransaction();  tx.commit();  ServletActionContext.*getResponse*().getWriter().write("succ");  } **catch** (Exception e) {  // **TODO**: handle exception  tx.rollback();  **throw** e;  }  }  /\*\*  \* 添加一条科研项目设置信息  \*/  **public** String addProjectType() **throws** Exception {  proType.setProjectTypeId(pkmk.mkpk("ProjectTypeID", "ProjectType", "srpt"));  proType.setSpareTire("1");  Transaction tx = **null**;  **try** {  proTypeDAO.save(proType);  tx = proTypeDAO.getSession().beginTransaction();  tx.commit();  **this**.setOperstatus(1);  **this**.viewScienResearchProjectType();  } **catch** (Exception e) {  // **TODO**: handle exception  tx.rollback();  **this**.setOperstatus(-1);  **throw** e;  }  **return** "success";  } |