学号：2015261020203

200222200X2XX40XXX



**本 科 生 毕 业 论 文**

论 文 题 目： 基于JavaWeb的奖助学金管理系统设计

作 者： 陈 帅

院 系： 计算机学院

专 业： 计算机科学与技术

班 级： 201302

指 导 教 师： 关 玉 蓉

2017 年 4 月 10 日

NO.： 2015261020203

2008200X2XX40XXX

200X2XX40XXX

**Huanggang Normal University**

**Thesis Graduates**

Topic ： Design of Scholarship Management Baseon JavaWeb

Author ： Chen Shuai

College ： College of Computer

Specialty ： Computer Science and Technology

Class ： 201302

Tutor ： Guan Yurong

4 10, 2017

**郑重声明**

本人所呈交的毕业论文（设计）是本人在指导教师 关玉蓉 的指导下独立研究并完成的。除了文中特别加以标注引用的内容外，没有剽窃、抄袭、造假等违反学术道德、学术规范和侵权行为，本人完全意识到本声明的法律后果由本人承担。

特此郑重声明！

指导老师（签名）：

论文作者（签名）：

2017年4月10日

**摘 要**

奖助学金设立，是为了激励勤奋学习、努力进取的学生或对家庭困难的学生进行资助。为简化奖助学金的评比工作，加快评比效率，采用B/S结构，搭建网站系统是一种优秀的解决方案。本设计是基于Java Web的奖助学金管理系统，参考传统奖助学金评比流程，结合学生及老师等多种角色的需求，运用本人所学的软件工程及编程知识，并借鉴多个类似项目经验来进行开发，运用当前行业流行的SSM(spring struts2 mybatis)框架及MVC思想，前端使用以jQuery为主的JS构建，后台数据库采用标准的轻量化数据库MySQL，运用多种经典的设计模式，并选择Github来作为项目代码及分支管理工具。项目具有良好的代码风格，高度的可维护扩展性，优秀的页面及操作逻辑。各种角色用户只需要通过浏览器便可以实现登录系统进行操作。服务器的选择也保证了系统的流畅运行。

本系统包含六大模块：学院、班级、角色、账户、申请、审批。同时包含日志功能和安全防御模块。学生可以登录系统填写信息及查询评比进度或结果。教师可以通过系统直观对比学生各项信息然后分配奖助学金种类等级，同时可以导出各种数据表格。

论文主要介绍了本课题的开发背景及开发过程。详尽的阐述了系统的需求分析、技术说明、设计思想、功能实现、测试及维护各项过程。系统开发过程中的重点在论文中都有体现，

**关键词：**Scholarship ,B/S, SSM, MySQL

**Abstract**

**目 录**

# 绪论

## 1.1 研究背景和意义

奖助学金设立，是为了激励勤奋学习、努力进取的学生或对家庭困难的学生进行资助。奖助学金的评比力求做到公正公平公开。我校长期以来奖助学金的评比工作十分繁琐复杂，并没有使用一些工具，因此容易出现纰漏错误，以及评比效率不尽人意。我校历来奖助学金评比，首先是学校发送通知及材料到各班班主任，班主任自行联系本班同学，然后同学准备材料提交申请，教师在验证材料后选择评比小组进行评比，然后公示，待全班同学同意后方可向上提交申请，同时还会有各项表格需要同学们再次填写确认。教师还需要完成其它的表格以进行其它方面的统计。在这个过程中有几个问题：

1.学生每年都要提交材料，即使材料与去年相似还是要重新审核。

2.每个学生提交材料时间不一定，容易出现意外拖延，造成时间浪费。

3.评比过程繁琐，学生不能很好的了解审批过程，不够透明。

4.学生信息需要确认多次，人为处理费时费力

5.没有较好的历年数据进行直观的对比，同学老师经常问起以前获奖之人。

这些问题综合起来，导致整个奖助学金申请审批任务繁重。为此，借助计算机的高性能及相关技术，来实现解决此问题，成为教师学生的一致想法。

随着科学和社会的发展，信息化的潮流已经深入渗透到各行各业的方方面。近年来，网络技术的发展迅速，各种网站架构平台得到了越来越多的应用，如PHP,ASP.NET，J2EE等。前端网页的技术也是百花齐放，各种基于JS的框架如雨后春笋般林立，如Bootstrap,React,Angular JS等。数据库的发展也日渐完善，关系型数据库如mysql、Oracle等，非关系型数据库MongoDB等，数据库的完善性不断加强，效率也越来越高。这些技术的成熟发展，高效开发能力，以及各自的特点，为Web系统的开发提供了强大的基础。

MVC模式是施乐帕罗奥多研究中心（Xerox PARC）在20世纪80年代为程序语言Smalltalk发明的一种软件架构。MVC模式的目的是实现一种动态的程序设计，使后续对程序的修改和扩展简化，并且使程序某一部分的重复利用成为可能。现MVC思想已经深入渗透到程序设计当中。

Spring于2003 年兴起的一个轻量级的Java 开发框架。它的特点包含控制反转、面向切面、容器、MVC。Spring的源码设计精妙、结构清晰、匠心独运，处处体现着大师对Java设计模式灵活运用以及对Java技术的高深造诣。Spring 中间层可以轻易地结合于任何基于 MVC 框架的网页层，例如 Struts。

Struts2是一个基于MVC设计模式的Web应用框架，它本质上相当于一个servlet，在MVC设计模式中，Struts2作为控制器(Controller)来建立模型与视图的数据交互。

MyBatis 本是apache的一个开源项目iBatis。它一个基于Java的持久层框架。它最强大的特性之一就是支持动态SQL。Mybatis是一个轻量级框架，对于小项目来说非常合适。

SSM(Struts+Spring+MyBatis)是一个典型的基于MVC思想的集成框架。实现了视图、控制器与模型的彻底分离，而且还实现了业务逻辑层与持久层的分离。这样无论前端如何变化，模型层只需很少的改动，并且数据库的变化也不会对前端有所影响，大大提高了系统的可复用性。而且由于不同层之间耦合度小，有利于团队成员并行工作，大大提高了开发效率。

本文将描述如何采用SSM框架开发奖助学金管理系统，以解决上述人为评比过程中容易出现的错误。

## 研究内容

系统将以MVC为思想分层设计系统结构，同时功能采取模块化。系统的功能模块如下：

1．学院模块 包含学院的一些基本信息

2．班级模块 包含班级的一些基本信息，比如入学年份、学历、专业名称、所属学院、学年制等。

3．角色模块 固定角色为 管理员 和 学生。可自定义角色，并关联此角色对应的多个学院或多个班级。

4．账户模块 此模块即学生或教师的信息，以学号或工号作为登录账号，密码自己修改。账户关联的角色为学生则填写所属学院班级。账户关联的角色为自定义角色，则为教师或辅导员。所管理的学院班级即对应角色所关联的学院班级。

5．申请模块 此模块为学生填写自己详细的信息，比如身份证 银行卡及联系信息，以及家庭基本信息和学校成绩信息等。填写完成后，提交申请，以供教师进行评比。

6．审批模块 此模块为教师用来审批学生申请。教师所管理的班级学生提交的未审批的申请都会在这里显示出来。此模块会将个学生的详细信息以直观的方式显示出来，方便老师进行筛选。最终进行奖助学金的分配。同时这个模块还包含了以往所有的申请提交结果。教师可随时查看。另外，教师也能在这里导出各种表格，如 国家励志奖学金初审表等。

同时系统还有日志审计模块和安全防御功能。

# 关键技术说明

## 2.1 Java EE说明

Java EE,Java平台企业版，它由2005年SUN公司改名自j2ee而来。Java EE是由一系列的技术标准所组成的平台，这其中包含Applet、JDBC、EJB、JSP、Servlet、JNDI、JSTL等广为人知的技术。目前版本已经达到Java EE 7 (2013年5月28日)。随着Java技术的发展，J2EE平台得到了迅速的发展，成为Java语言中最活跃的体系之一。现如今，J2EE不仅仅是指一种标准平台，它更多的表达着一种软件架构和设计思想。

Java EE和传统应用开发的技术架构相比，它包含许多组件，主要可简化且规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。其核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循J2EE架构的不同平台之间，存在良好的兼容性。

## 2.2 Spring说明

Spring 是一个开源的Java／Java EE全功能栈（full-stack）的应用程序框架，以Apache许可证形式发布，也有.NET平台上的移植版本。

Spring的一个特点是控制反转（Inversion of Control，Ioc）。控制反转是面向对象编程中的一种设计原则，可以用来减低计算机代码之间的耦合度。最常见的实现方式是依赖注入（Dependency Injection，DI）。程序中将对象的控制权交予Spring容器，在运行时候才由容器决定将具体实现的动态的”注入”到被调用类的对象中。

Spring的另一个特点是面向切面（aspect-oriented programming，AOP）。面向切面是面向对象的一种补充。面向切面是一种可以通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一添加功能的一种技术。比如我们可以将多种功能的日志代码提取出来，这样降低了代码的重复性。并且便于管理。

Spring 框架是一个分层框架，由 7 个定义良好的模块组成。Spring 模块构建在核心容器之上，核心容器定义了创建、配置和管理 bean 的方式。如图2.2。Spring 框架的功能可以用在任何 J2EE 服务器中。 Spring 的核心要点是：支持不绑定到特定 J2EE 服务的可重用业务和数据访问对象。因此，这样的对象可以在不同 J2EE 环境 （Web 或 EJB）、独立应用程序、测试环境之间重用。

## 2.3 Struts2说明

Struts2是采用MVC思想设计的一种Web应用框架。它本质上等同于一个Servelet，对Servlet进行了封装。Struts2 框架如图2.3所示，处理流程可以看出，客户端首先发送一个请求，这个请求经过一系列的过滤器(Filter)后，核心控制器FilterDispatcher动态调用Action,执行完毕后，会根据Struts.xml返回浏览器，可以是HTML，图像，文档，此时支持的视图非常多。



图2.2 Spring框架



图2.3 Struts2 框架

Struts 2 是一优秀的web框架，如今越来越多的项目开始采用Struts2进行开发。同时需要注意的是，Struts曾经出现过安全漏洞，因此，采用新版本的Struts2是一种良好的策略。

## 2.4 Mybatis 说明

MyBatis是一个开源的Java持久化框架，它通过XML描述符或注解把对象与存储过程或SQL语句关联起来。它支持自定义SQL、存储过程和高级映射。它几乎消除了所有JDBC代码，也基本不需要手工去设置参数和结果。Mybatis和Hibernate相比，其轻量化的特点得到了诸多开发人士的青睐。

Mybatis的该能架构分为三层，如图2.4所示。



图2.4 Mybatis 框架

API接口层：提供给外部使用的接口API，开发人员通过这些本地API来操纵数据库。接口层一接收到调用请求就会调用数据处理层来完成具体的数据处理。

数据处理层：负责具体的SQL查找、SQL解析、SQL执行和执行结果映射处理等。它主要的目的是根据调用的请求完成一次数据库操作。

基础支撑层：负责最基础的功能支撑，包括连接管理、事务管理、配置加载和缓存处理，这些都是共用的东西，将他们抽取出来作为最基础的组件。为上层的数据处理层提供最基础的支撑。

# 系统分析

黄冈师范学院前身是创建于1905年的“黄州府师范学堂”。历经百余年耕耘，学校现已发展成为一所多科性大学，涉及理、工、农、经、文、法、教、管、艺等九大学科门类，形成硕士研究生、本科、专科三种办学层次。全日制在校学生16729人。奖助学金名额数千次人计，包含原鄂东职业技术学院。奖助学金管理系统就是辅助学校，简化教师学生的操作，在公正公平公开的原则下，提高评比效率，同时保存数据。

## 3.1 可行性分析

黄冈师范学院奖助学金多年来评比公平透明，在设立国家奖助学金类别同时也设立了如“明珠学子”奖助学金等其它类别。这些评比过程都已经实现规范化，因此，在这基础上实现系统功能设计分析实现是完全可行的。黄冈师范学院奖助学金评比是以校学生资助管理中心为中心的，同时结合同学的学科成绩排名及家庭基本信息。系统信息流程图如下：



图3.1 系统信息流程图

#### 3.1.1 奖助学金说明

国家励志奖学金与国家助学金

一、资助对象

我校家庭经济困难的全日制本专科学生；

原鄂东职业技术学院家庭经济困难的全日制专科学生。

二、申请条件

（一）国家励志奖学金的申请条件

国家励志奖学金用于奖励资助高校全日制普通本专科学生中，品学兼优的大二以上（含大二）家庭经济困难学生，不得和国家奖学金兼得；且符合以下条件：

1、热爱社会主义祖国，拥护中国共产党的领导；

2、遵守宪法和法律，遵守学校规章制度；

3、诚实守信，道德品质优良；

4、在校期间学习成绩优秀；

5、家庭经济困难，生活俭朴。

（二）国家助学金的申请条件

国家助学金用于资助高校全日制普通本专科在校生中的家庭经济困难学生，且符合以下条件：

1、热爱社会主义祖国，拥护中国共产党的领导；

2、遵守宪法和法律，遵守学校规章制度；

3、诚实守信，道德品质优良；

4、勤奋学习，积极上进；

5、家庭经济困难，生活俭朴。

三、资助金额和名额

黄冈师范学院名额：

（一）国家励志奖学金额度为每人5000元/年，按比例确定每年具体名额；

（二）国家一等助学金额度为每人2000元/半年，按比例确定每年具体名额；

（三）国家二等助学金额度为每人1500元/半年，按比例确定每年具体名额；

（四）国家三等助学金额度为每人1000元/半年，按比例确定每年具体名额。

各学院要按分配的名额等额进行评审推荐。

原鄂东职业技术学院名额：

（一）国家励志奖学金额度为每人5000元/年，按比例确定每年具体名额；

（二）国家一等助学金额度为每人2000元/半年，按比例确定每年具体名额；

（三）国家二等助学金额度为每人1500元/半年，按比例确定每年具体名额；

（四）国家三等助学金额度为每人1000元/半年，按比例确定每年具体名额。

各学院要按分配的名额等额进行评审推荐。

四、评定程序

符合条件的学生可向各学院提出书面申请，由各学院初审、公示、确定初审名单，并分别填写《湖北省国家励志奖学金申请审批表》和附表2《湖北省国家助学金申请表》（两表由各学院填写完整，统一收齐送校学生资助管理中心盖章后，留在本学院存档备查）。最后由我校家庭经济困难学生资助工作领导小组集体审核、公示、确定。评比阶段分为各学院宣传、摸底、初审阶段、各学院公示、上报获奖者和资助者名单阶段、学校家庭经济困难学生资助工作领导小组汇总审定、公示和确定名单阶段及学校领导终审名单阶段，同时向省教育厅上报我校名单。

明珠学子”励志奖学金

一、奖励对象

我校全日制本专科学生中品学兼优的学生。

二、基本条件

申请“明珠学子”励志奖学金的基本条件

二年级以上（含二年级）的全日制普通本专科在校生，符合以下条件：

（一）热爱社会主义祖国，拥护中国共产党的领导；

（二）遵守宪法和法律，遵守学校各项规章制度，学年内未受到任何处分；

（三）团结同学，诚实守信，道德品质优良；

（四）勤奋学习，刻苦钻研，积极追求上进。

（五）学业综合成绩排名上一学年度前1/3的成绩优秀学生或学业综合成绩排名上一学年度前1/2且在社会实践、创新能力、综合素质、个人特长等方面有突出表现的学生

三、奖励标准和名额分配

“明珠学子”奖学金的等级、标准及评选人数：

一等奖学金每人每年3000元，按比例确定每年具体名额；

二等奖学金每人每年2000元，按比例确定每年具体名额；

三等奖学金每人每年1000元，按比例确定每年具体名额。

各学院要按分配的名额等额进行评审推荐，名额分配见附表3。

四、评定程序

评定程序：符合条件的学生可向本学院提出书面申请;由各学院初审、公示、确定初审名单，并填写《（黄冈师范学院“明珠学子”励志奖学金申请审批表》一式两份（一份学院留存备查，一份装入学生档案）。同时填写附件2《黄冈师范学院XXXX学年“明珠学子”励志奖学金获奖学生名单登记表》上报学生资助管理中心审核；最后由学校“明珠学子”励志奖学金评审推荐工作领导小组审查、公示、确定。

## 3.2 需求分析

#### 3.2.1 系统流程分析

奖助学金的流程如图3.2.1所示，在奖助学金通知开始之后，学生提交申请，学院接受申请，各个班级根据自己情况进行意见交流，然后提交进行一次审批，然后公示以及提交各项表格，然后二次审批，再次公示，最终完成。



图3.2.1 评比流程图

#### 3.2.2 用例分析

UML提供的用例图是进行需求分析和功能建模的强有力工具。用例是对一个活动者(Actor)使用系统的异乡功能时所进行的交互过程的一个文字描述序列。使用用例图可以将我们的系统功能模型更加清晰，使分析和开发人员不再受限于实现细节，从整体上来把握系统的设计。

⑴管理员管理申请提交用例

如图3.2.2.1，管理员在每年的奖助学金申请工作开始时，开启系统的提交申请按钮，结束后可以关闭提交申请。这样保证了申请提交时间的规范性。各个奖助学金的评比时间不尽相同，因此，此功能配合公告可以使多种奖助学金的评比工作更清晰。



图 3.2.2.1 管理员管理申请提交用例

用户：管理员

前提条件：用户已登录

主事件流：

①用户登入系统，选择系统设置。

②查看当前申请提交状态(开启/关闭)。

③选择更改状态按钮。

④修改成功并提示。

⑤事件流结束。

⑵学生提交申请用例

学生登入系统后，核对自己基本信息，填写其它必要信息(如银行卡号),如果提交的是助学金，还需要填写家庭信息，也方便评审时进行比较。提交申请后。可以查看审批的进度和状态。



图 3.2.2.2 学生提交申请用例

用户：学生

前提条件：用户已登录

主事件流：

①用户登录系统，选择奖/助学金申请模块。

②用户进入该模块，并提示相关信息。

③用户填写必要信息并通过格式验证。

④用户选择提交申请，未曾提交申请则提示“提交成功”。否则执行事件流A1。

⑤事件流结束。

其它事件流A1：

①更新用户填写申请资料。

②更新用户申请审批状态。

③转到主事件流5.

⑶审批申请用例。

教师登录系统后，能够查看自己所管理学生中提交的所有申请信息，并且系统把学生提交助学金申请时家庭信息对比显示出来，同时包含学生以往获奖助金情况、成绩排名情况等。教师可以查看学生具体完整的申请信息。当教师管理学生过多时候，可以进行一些条件筛选。最后教师可以分配给学生奖助学金等级。申请需要教师级别和院负责人级别俩次审批。



图 3.2.2.3 审批申请用例

用户：教师/辅导员等用户

前提条件：用户已登录，学生已提交申请。

主事件流：

①用户登录系统，并选择审批申请模块。

②进入模块，显示所有待审批学生申请。

③选择筛选条件，显示特定学生申请。

④单击学生，查看学生基本信息。

⑤单击申请，查看申请的具体完整信息。

⑥选择奖助金类别，审批通过申请。

⑦选择申请，驳回申请。

⑧自动修改审批状态，教师审批完成需要院负责人再次审批

⑨事件流结束。

⑷审批管理用例

当奖助学金申请完全审批完毕后，这些审批信息也会保存。包含学生提交的申请，审批结果，审批日期，审批人等。同时我们还能筛选这些审批申请信息以及导出各项数据表格。如2017年计算机学院的国家助学金名单表。这些不同的表格类型能满足多种需求，如名单表，卡号核对表等。



图 3.2.2.4 管理审批申请用例

用户：非学生管理角色用户

前提条件：用户已登录，申请审批已完成。

主事件流：

①用户登录系统，并选择审批申请模块所有审批申请功能。

②查看所有审批申请信息。

③查看某一审批的学生详细申请信息。

④选择多项审批信息。

⑤修改奖助金类别。

⑥选择通过/废弃操作并执行。

⑦主事件流结束。

用户：非学生管理角色用户

前提条件：用户已登录，申请审批已完成。

主事件流：

①用户登录系统，并选择审批申请模块所有审批申请功能。

②查看所有审批申请信息。

③选择导出年份。

④选择导出数据表格类型。

⑤选择导出表格，然后下载。

⑥主事件流结束。

⑸学院管理用例

管理员可以添加删除学院，以及修改学院信息。

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择学院管理模块。

②选择添加学院。

③填写学院信息。

④选择保存。

⑤主事件流结束。



图 3.2.2.5 学院管理用例

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择学院管理模块。

②选择学院。

③修改学院信息。

④选择保存。

⑤主事件流结束。

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择学院管理模块。

②选择学院。

③选择删除选项。

④选择执行。

⑤主事件流结束。

⑹班级管理用例

管理员及院长可以添加删除班级，以及修改班级信息。

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择班级管理模块。

②选择添加班级。

③填写班级信息。

④选择保存。

⑤主事件流结束。



图 3.2.2.6 班级管理用例

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择班级管理模块。

②选择班级。

③修改班级信息。

④选择保存。

⑤主事件流结束。

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择班级管理模块。

②选择班级。

③选择删除选项。

④选择执行。

⑤主事件流结束。

⑺角色管理用例

角色可以把账户和学院班级绑定起来，一个教师可以管理一个、多个班级或一个、多个学院。这时候灵活的班级和学院对应很复杂，所以将二者设计成可以自由搭配。这样满足了很多角色的管理需求。

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择角色管理模块。

②选择添加角色。

③填写角色信息，选择关联学院/班级。

④选择保存。

⑤主事件流结束。

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择角色管理模块。

②选择角色。

③修改角色信息。

④选择保存。

⑤主事件流结束。

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，选择角色管理模块。

②选择角色。

③选择删除选项。

④选择执行。

⑤主事件流结束。



图 3.2.2.7 角色管理用例

⑻账户管理用例

非学生的管理类角色账户，可以管理其下级账户的基本信息。如修改学生的学院班级，基本联系方式，以及账号密码等。同时，辅导员可以在这里导入学生账户，或者批量重置学生账户密码。

用户：辅导员、管理员等

前提条件：用户已登录

主事件流：

①用户登录系统，并选择账户管理模块。

②选择 添加账户 。

③填写账户基本信息，并通过校验。

④点击保存

⑤主事件流结束。

用户：辅导员、管理员等

前提条件：用户已登录

主事件流：

①用户登录系统，并选择账户管理模块。

②选择待修改账户 。

③修改账户基本信息，并通过校验。

④点击保存

⑤主事件流结束。



图 3.2.2.8 账户管理用例

用户：辅导员、管理员等

前提条件：用户已登录

主事件流：

①用户登录系统，并选择账户管理模块。

②勾选待删除账户 。

③选择删除操作。

④点击执行。

⑤主事件流结束。

用户：辅导员、管理员等

前提条件：用户已登录

主事件流：

①用户登录系统，并选择账户管理模块。

②选择导入账户。

③进入导入账户页面，选择正确的EXCEL文件。

④点击执行导入

⑤若导入成功，显示新导入账户列表，否则执行其它事件流A2。

⑥主事件流结束。

其它事件流A2：

①勾选需要导入覆盖重复账户及显示非法账户信息。

②执行导入。

③显示导入账户列表。

④事件流结束。

⑼审计用例

系统的各项操作，如各账户的登录成功失败，申请的提交审批结果，审批的修改驳回，以及对系统设置的修改等，这些操作都需要通过审计模块来记性记录，以方便系统出现错误后进行分析查看操作。



图 3.2.2.9 审计用例

用户：管理员

前提条件：用户已登录

主事件流：

①用户登录系统，并选择审计模块。

②查看操作列表。

③选择筛选条件，并执行。

④查看审计信息。

⑤主事件流结束。

⑽修改系统设置选项用例

系统设置中包含一下选项：开始/关闭登录验证码、奖/助学金类别管理，填写申请提示信息设置，申请说明文件。这些操作都是管理员的权限。管理员登录系统后，选择系统设置模块就能对其进行修改。

# 系统设计

## 4.1 整体设计

系统采用三层架构，以MVC中心思想。系统的整体架构如图：



图4.1 体系结构图

Action层：只处理页面发送的请求。

Service层：处理各项功能的业务逻辑。

DAO层：提供数据库相关的操作。

每个层次只处理自己的核心问题，将业务逻辑与数据持久化分开，实现功能的解耦。这样，系统的代码逻辑更加清晰，分析问题也方便。

## 4.2 页面设计

系统整体页面布局采用上中下三个区域。上层显示系统LOGO及各项功能模块。中层分左右两个层次，左侧为二级功能模块菜单，右侧为功能模块主操作区域。下层显示系统版本及版权信息。

## 4.3 功能模块

系统分为申请模块，审批模块，账户管理模块，角色管理模块，学院管理模块，班级管理模块，审计模块及系统设置模块。这些模块的逻辑由以下Service来进行业务逻辑的处理：

AccountService:处理账户业务逻辑。

LoginService:处理登录业务逻辑。

ApplyService:处理审批申请业务逻辑。

AuditService:处理审计业务逻辑。

CollegeService:处理学院业务逻辑。

DatasService:处理申请信息业务逻辑。

GradeService:处理班级业务逻辑。

RoleService:处理角色业务逻辑。

ScholarshipService:处理奖助学金类别业务逻辑。

SettingService:处理系统设置业务逻辑。

这10个Service类对数据库的操作，是通过DAO类来实现的,DAO类中封装了对表的基本操作如增删改查，还有一些特性的数据操作。DAO类列表如下：

AccountDao:处理对账户表的数据操作。

ApplyDao:处理对审批申请表的数据操作。

AuditDao:处理对审计表的数据操作。

CollegeDao:处理对学院表的数据操作。

DatasDao:处理对申请信息表的数据操作。

GradeDao:处理对班级表的数据操作。

RoleDao:处理对角色表的数据操作。

ScholarshipDao:处理对奖助学金类别表的数据操作。

SettingDao:处理对系统设置表的数据操作。

## 4.4 数据库设计

系统采用MySQL数据库来实现维护与管理。Mysql数据库的优势这里不在累述。

系统设立的Entity实体类及数据表对照如下：

|  |  |  |
| --- | --- | --- |
| 表 | 实体类 | 说明 |
| Account | Account | 账户 |
| Apply | Apply | 审批申请 |
| Audit | Audit | 审计 |
| College | College | 学院 |
| Datas | Datas | 申请信息 |
| Grade | Grade | 班级 |
| Role | Role | 角色 |
| Relation | 无 | 角色关联 |
| Scholarship | Scholarship | 奖/助学金类别 |
| Setting | BaseConfig | 系统设置 |
| 无 | AppConfig | 系统级配置变量 |

数据表节选：



图4.2.1.1 账户表

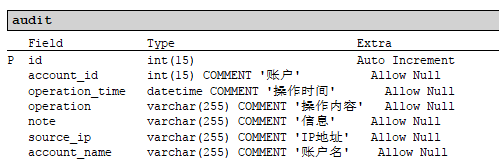


图4.2.1.2 审计表



图4.2.1.3 班级表

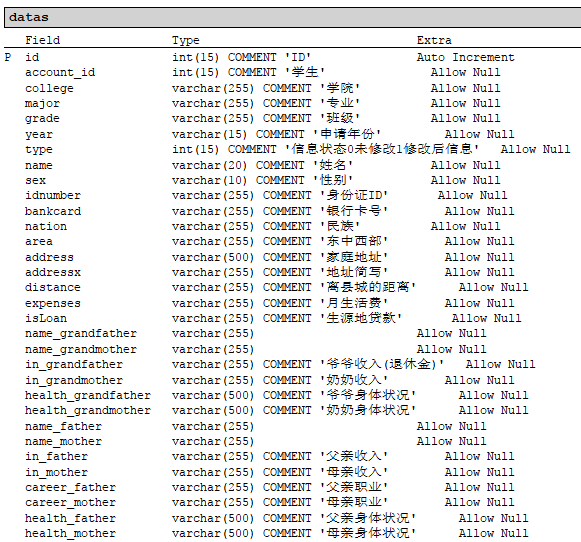


图4.2.1.4 申请信息表(部分)

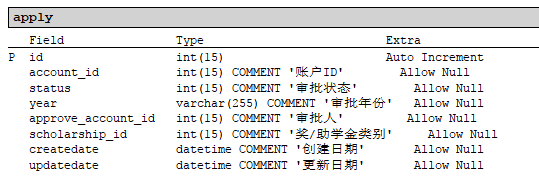


图4.2.1.5 审批申请表

# 系统实现

## 5.1 项目包层次图

本Java Web项目名称为Scholarship。包含俩大根路径，src为后端文件，其中有java类文件及配置文件等，Webroot为前端文件，其中有JSP，JS及图片文件等。

src结构如下：



图5.1.1 src包结构图

com.scholarship.dao：处理数据操作的Dao接口类文件。

com.scholarship.dao.impl：Dao接口具体实现类文件。

com.scholarship.module：JavaBean类(即实体类)文件。

com.scholarship.service：处理业务逻辑的Service接口类文件。

com.scholarship.service.impl：Service接口具体实现类文件。

com.scholarship.webapp.action：处理请求的Action类文件。

com.scholarship.webapp.filter：Http请求过滤器文件。

com.scholarship.webapp.interceptor：Http请求拦截器文件。

com.scholarship.webapp.listener：监听器文件。

prop：JDBC参数文件。

spring：spring配置文件，包含dao,database,service,struts等。

sqlmap：Mybatis sql配置文件。

struts：Action请求配置文件。

WebRoot结构如下：



cvsTemplate：各项数据表格的标准模板，提供下载。

images：页面图片文件。

pages：网页文件。

scripts：JS文件。

styles：CSS文件

## 5.2 代码实现

#### 5.2.1 JavaBean类

JavaBean类即是Entity实体类。同时JavaBean类对数据做了封装，对属性进行private控制，再通过set get方法来实现对其修改访问。另外，与数据库表字段不同的是，JavaBean对其它对象的引用是直接创建其对象，而不是数据库表中通过外键关联实现。我们在Mybatis的配置中可以实现外键转换成对象，这就十分的方便了。

#### 5.2.2 DAO类

DAO类基本的操作为增删改查，此外，每个DAO都有自己不同的逻辑，所以也会有其他的功能方法。下面以AccountDaoImpl类为例，其功能包含，统计账户人数、多种格式的查询，如按学号查询，按名字查询，按条件查询，账户的添加、修改、删除。 具体代码如下图所示：



在SSM中，通过Spring来进行Mybatis的管理。配置项有DataSource(数据源)，sqlSessionFactory、transactionManager。采用双数据库设计，所以需配置两个数据源，通过transactionManager能实现事务的管理，通过sqlSessionFactory我们能够得到sqlSession，进而执行sql。这些的对象通过Spring注入，只需要编写功能代码。其中count方法对应的Mybatis sql实现代码如下：



#### 5.2.3 Service类

Service类主要处理业务逻辑。其需要DAO对象的支持，而和DAO类一样，我们将将Service中DAO类的创建交予Spring的注入实现。以账户的统计功能及登录功能为例，代码如下：





#### 5.2.4 Action 类

Action类通过页面请求来执行，通过struts的配置文件来使请求与Action类绑定，通知配置返回页面或其它返回类型。同样，我们通过Spring来管理Action的各项配置。Action类中对service的引用也是通过Spring注入实现的。以登录时查询用户是在在线功能为例，代码如下：



## 5.3 系统运行

系统采用Tomcat7为服务器发布。

系统登录功能图示如下：



图 5.3.1 登录页面

当学生登录系统后可以查询申请状态及填写申请信息，提交申请。功能图示如下：



图 5.3.2 申请页面

辅导员登录系统后，可以对学生的提交申请进行查看并进行审批。



图 5.3.3 审批图

同时，对于已经审批完成的申请，可以导出各项数据表格：

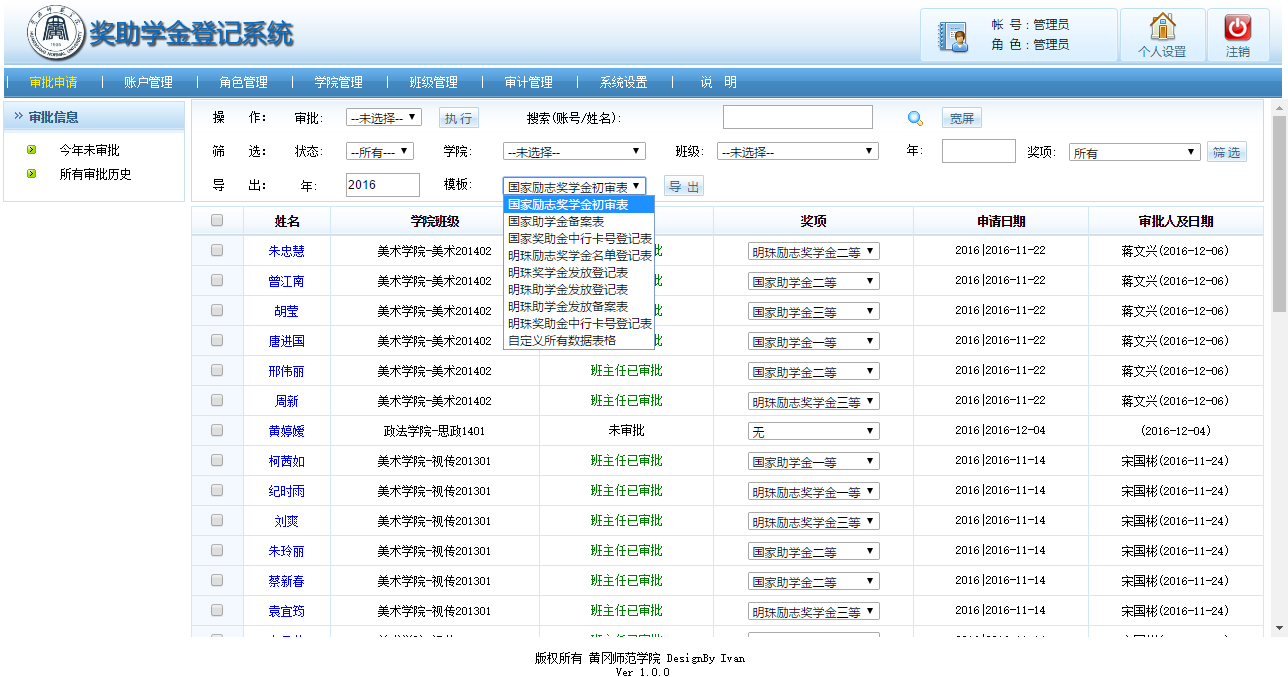


图 5.3.4 导出表格图

管理员及教师，可以导入学生信息，可在页面下载表格模板，然后填写数据，执行导入：

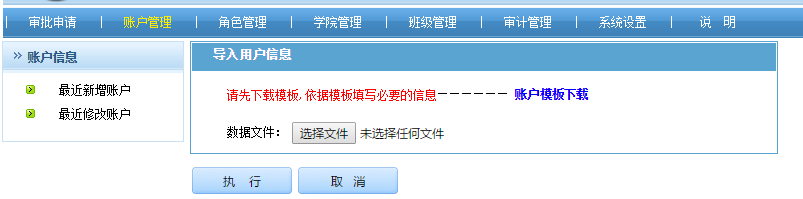


图5.3.5 导入用户图

管理员同时能够对账户，角色，学院，班级等进行管理。



图5.3.6 账户图

另外，对于不懂操作流程的用户，可以在说明里查看使用手册：



图 5.3.7 说明图

# 第6章 系统测试