

代 号 10701
分类号 TP31

学 号 1104121991
密 级 公开

西安电子科技大学

硕士学位论文



题 (中、英文) 目 基于 SSH 框架的学生信息管理系统的研究与实现

Research and implementation of student information
management system based on SSH framework

作 者 姓 名 汤健哲 指导教师姓名、职务 赵明英 副教授

学 科 门 类 工 学 学科、专业 电机与电器

提交论文日期 二〇一四年五月

西安电子科技大学

学位论文独创性(或创新性)声明

秉承学校严谨的学风和优良的科学道德,本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知,除了文中特别加以标注和致谢中所罗列的内容以外,论文中不包含其他人已经发表或撰写过的研究成果;也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处,本人承担一切的法律责任。

本人签名: 汤健哲

日期 2014.6.23

西安电子科技大学

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定,即:研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件,允许查阅和借阅论文;学校可以公布论文的全部或部分内容,可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证,毕业后结合学位论文研究课题再撰写的文章一律署名单位为西安电子科技大学。

(保密的论文在解密后遵守此规定)

本学位论文属于保密,在___年解密后适用本授权书。

本人签名: 汤健哲

日期 2014.6.23

导师签名: 赵明英

日期 2014.6.23

摘要

随着社会经济的发展，学生人数不断增长，如何有效的对学生信息进行管理越来越受到重视。相比老式的管理方式，以互联网络作为传播媒介的信息化管理系统，具有安全可靠、更新方便、存储量大、成本低等特点，从而得到了很好的发展。

本文的目标就是研究并实现一个基于SSH框架的学生信息管理系统。首先对J2EE的多层架构及组件技术进行了深入的分析，同时对基于J2EE的轻量级框架Struts、Spring和Hibernate框架进行了研究，结合Struts、Spring和Hibernate的各自特点，将这三种框架整合成一个SSH多层框架。系统设计则由于考虑到了扩展性和可维护性，采用了模块化编程的方式，构建了表示层、业务逻辑层、数据持久层等几个方面。在此基础上论述了系统主要模块的设计过程和设计结果，并展示了系统各部分的详细实现。

关键词： 信息管理 J2EE Struts框架 Spring框架 Hibernate框架

Abstract

With the social and economic development, the number of students is constantly increasing, how to effectively manage the information of students has been paid more and more attention. Compared with the old way of management, to the Internet as information management system of the media, is safe and reliable, convenient update, large storage capacity, low cost, and thus obtained the very good development.

The goal of this paper is to research and implement a student information management system based on SSH framework. Firstly, multi-layer architecture of J2EE and component technology are analyzed, at the same time, the Struts lightweight framework J2EE and Hibernate framework based on Spring are studied, combined with their respective characteristics of Struts, Spring and Hibernate, these three frameworks are integrated into a SSH multilayer frame. System design, considering the expansibility and maintainability, using modular programming way, constructs the said several layer, business logic layer, data persistence layer. In this paper based on the design process and results of the main modules of the system, and presents the detailed implementation of each part of the system.

Keywords: **Information management** **J2EE**
 Struts framework **Spring framework** **Hibernate framework**

目录

第一章 绪论	1
1.1 研究背景与意义	1
1.2 国内外现状	1
1.3 论文主要研究内容	3
1.4 本文的组织结构	3
第二章 系统相关技术研究	5
2.1 J2EE 技术架构	5
2.1.1 J2EE 架构分析	5
2.1.2 J2EE 架构的组成技术	5
2.2 STRUTS 框架	6
2.2.1 MVC 框架	6
2.2.2 Struts 框架原理	7
2.3 HIBERNATE 框架	8
2.3.1 Hibernate 框架特性	8
2.3.2 Hibernate 框架运行流程	9
2.3.3 Hibernate 框架缓存机制	11
2.4 SPRING 框架	11
2.4.1 Spring 框架特性	11
2.4.2 Spring 框架的反射与动态代理	12
2.5 SSH 框架整合	13
2.5.1 表示层	13
2.5.2 业务层	13
2.5.3 数据层	13
2.5.4 SSH 框架工作流程	13
2.5.5 SSH 框架特性	14
第三章 系统总体结构分析与设计	15
3.1 软件体系结构分析	15
3.1.1 C/S 结构分析	15
3.1.2 B/S 结构分析	15
3.2 系统结构分析与设计	15
3.2.1 系统总体结构设计	15
3.2.2 系统管理模块结构分析	16
3.2.3 管理员模块分析	16
3.2.4 教师模块结构分析	16
3.2.5 学生模块结构分析	17
第四章 系统软件详细设计	19
4.1 系统数据库详细设计	19
4.2 加密算法设计	25
4.2.1 数据加密算法分析	25

4.2.2AES 算法设计	25
4.3 系统软件详细设计	28
4.3.1 管理员模块详细设计	28
4.3.2 教师模块详细设计	29
4.3.3 学生模块详细设计	31
第五章 信息管理系统的具体实现	33
5.1 系统分层架构的实现	33
5.1.1 表示层的实现	34
5.1.2 业务逻辑层的实现	36
5.1.3 数据持久层的实现	37
5.2 系统具体业务实现	39
5.2.1 登陆页面	39
5.2.2 管理员页面	39
5.2.3 教师页面	41
5.2.4 学生页面	43
5.3 安全模块实现	45
第六章 总结	47
致谢	49
参考文献	51

第一章 绪论

1.1 研究背景与意义

随着社会经济的发展以及科技水平的不断进步,接受高等教育的学生人数也在不断增长。如此一来,就导致了各个高等院校的学生规模不断扩大,如何有效的管理学生信息也就成为了学校管理工作的重要组成部分。由于信息管理在当下的学校管理中占据了比较大的比例,各项管理工作都有一部分工作需要依附于信息管理,所以信息管理工作已经成为了一项判断学校教育质量好坏的一个重要指标。

以往的管理方式是采用纸质作为信息载体的,它存在着工作效率低、流程复杂、易出错等不利因素。而且现在的信息管理需要对大量的数据进行统计和分析,继续沿用以前的管理方式已经明显满足不了当前的教学管理需求了,因此,借助于计算机网络应用的信息化管理方式应运而生。

学生信息管理系统是实现信息化管理的一项具体应用,是为了应对大量学生信息的处理工作而开发的管理软件,一般借助于互联网络用以实现学生信息管理中的各种具体需求,主要功能室对各种学生信息进行常规的管理,如添加、删除、修改、查询等。具有查询快捷、安全可靠、更新方便、存储量大、成本低等特点,而这些特点都是过去的管理方式所无法企及的。所以越来越多的学校已经采用了此类系统,用以处理高校繁杂的管理工作这使得建立全面的具有通用性,实用性,灵活性和安全性的学生管理信息系统,具有一定的现实意义。

1.2 国内外现状

国内外对于学生信息管理系统一直都在的研究不断发展之中。从计算机在校内出现之日起,学生信息管理系统就开始发挥它的作用。可以这样说,学生信息管理系统是伴随着计算机技术的快速发展而不断进步的。现今绝大部分高校都已经实现了对学生信息的计算机网络化管理。

上个世纪 90 年代初,学生信息管理系统普遍的采用计算机集中处理数据的方式,基本上都是使用一个简单的数据库用以实现一些数据存储功能。随着技术的革新,学生信息管理系统中出现了一种基于计算机分布式处理来管理数据的方式。这种方式对软件、硬件的要求都比较高,系统的可移植性也不好,所以并没有得到大规模推广。而随着网络成本的不断下降,紧接着又出现了基于客户/服务器结构与浏览器/服务器结构的学生信息管理系统,这种系统一般运行在校园局域网内,功能也不是很丰富,只提供一些查询信息的基础功能。

进入新世纪以来,随着互联网技术的进一步发展,基于网络开发平台的学生信息管理系统开始得到了广泛的应用。当前主流的网络开发应用系统主要基于两种技术平台: SUN 公司推出的 J2EE 和 Microsoft 公司推出的 .NET 平台。

J2EE 6 自 2009 年 12 月推出以来,已经有 6 个主要的 IT 厂商经过认证、开源和商业实施后将其进行推广,使 J2EE 6 成为了到现在为止最为快速的获得采用的平台产品。作为下一代 J2EE 产品——J2EE 7 的开发进展也十分顺利。其中,有超过 20 个不同的企业和数百名工程师已经参与了它的开发,通过 JCP (Java 社区)对数十个活跃的 Java 规范要求进行了开发处理。

J2EE 7 的规范要求包括: J2 EE 7 平台、Java Persistence API 2.1、JAX-RS 2.0、用于 RESTful 网络服务的 Java API、Servlet 3.1、表达语言 3.0、Java 信息服务 2.0、JavaServer Faces 2.2、Enterprise JavaBeans 3.2、面向 Java EE 1.1 的 Contexts and Dependency Injection、Bean Validation 1.1 等。

由于 J2EE 平台的特点,很适合借助框架的开发,为应用程序提供了一个代码可重用的设计。让应用程序代码先访问类库然后再执行任务,与此同时框架则根据程序的流程,调用相应的程序代码。目前主流的开源框架首推 SSH 框架,它是由 struts 框架、spring 框架、hibernate 框架组成的一个集成框架。

Struts 框架出现于 2000 年,利用服务器端生成的 HTML 和客户端验证的 Javascript 的完美结合,使开发和维护变得更加容易,并且在大多数的标准下都运行良好。Struts 框架的最新版本是由 struts 和 webwork 衍生出 struts2,它的框架结构更为清晰,有更好的安全性,同时还拥有强大的拦截器,插件机制更易于扩展,框架结构有利于测试。

Spring 框架是 2003 年开始崭露头角的轻量级 J2EE 开源框架,是从 Rod Johnson 在其著作中描述的一些概念和思想引申出来的,用以解决复杂的企业应用开发。最新发布的 Spring4 相比旧版本在功能上更进了一步,拥有了许多新特性,如:泛型限定式依赖注入、核心容器的其他改进、Web 开发的增强、集成 Bean Validation 1.1 到 SpringMVC、Groovy Bean 定义 DSL、更好的 Java 泛型操作 API、JSR310 日期 API 的支持、注解、脚本、任务、MVC 等其他的特性改进。

Hibernate 是一个开源的轻量级框架,对 JDBC 进行了轻量级的对象封装,主要功能是对对象关系映射,让程序员对数据库操作时,能够便利的进行面向对象的编程。最近,JBoss 发布了流行的对象/关系映射框架 Hibernate 框架的最新版本——Hibernate 4。Hibernate 4 同样也拥有旧版本所没有的新特性,如:多租户架构支持、引入了 ServicesAPI、提供了更优秀的日志,支持了 i18n 与消息编码(通过 JBoss Logging 而非 slf4j)、为 OSGi 支持做好了准备。

1.3 论文主要研究内容

目前对学生信息管理系统研究不少，然而经过查阅大量的资料，却发现基于 SSH 框架的学生信息管理系统方面的研究不是很多，SSH 框架作为一个优秀的开源框架，已经应用到了很多管理系统的开发中。SSH 框架是典型 J2EE 三层结构，分为表现层、业务逻辑层和数据层，即三层体系结构。本课题研究的目标是在于研究与实现一个基于 SSH 框架学生管理系统，通过该框架开发出将学生的大量数据信息准确的分类、整理、查看的管理系统，并且根据不同的角色赋予不同的权限，管理学生的信息。

本文的研究重点在于：

(1) 需要一个具有足够稳定性的系统，使管理系统能够处理大量的并发用户，从而不会在访问人数过多时轻易地崩溃。系统采用模块化编程，各模块间实现低耦合高内聚的特点。这样一来，不但可以方便的加入新的功能模块，删除不需要的功能模块，而且利于系统后期的功能升级与维护。

(2) 现存的信息管理系统对于数据安全的问题都不是很重视，然而信息系统里面包含很多研究生或者老师的个人信息，例如联系方式、家庭住址、学生成绩等等。如果这些信息被不法分子盗取，将会带来很严重的问题。所以如何对系统进行设计，使它能够具有足够的安全性，避免可能会出现安全漏洞及数据泄漏，这也是本系统重点研究的部分。

(3) 对 J2EE 框架整合的研究，依照 Struts、Spring 及 Hibernate 的优点，将这三个框架进行相应的整合，并对其进行改进和优化扩展和优化，使 SSH 框架能够发挥更优异的性能。

1.4 本文的组织结构

第一章：提出了本文所研究课题的提出背景和研究意义，并介绍了本课题的主要研究内容及本文的组织结构。

第二章：详细地介绍了建立网络数据库应用系统的相关技术，即J2EE架构多层结构及组件技术，并对其不足指出进行了深入的分析；介绍了框架的基本概念同时对Struts、Spring和Hibernate等框架进行了研究与剖析。

第三章：对系统的总体结构进行了分析，介绍了系统软件模块的设计方案，并对各功能模块进行了简要的说明。

第四章：详细分析了系统软件部分的设计，包括数据库的设计；数据加密算法的设计；软件模块的功能结构及流程图设计。

第五章：详细讲述了系统的实现，展示了系统的界面布局并且介绍了系统不

同权限用户的操作流程。

第六章：总结论文完成的主要工作及存在的问题。

第二章 系统相关技术研究

本章主要介绍建立信息管理系统的相关技术，包括 J2EE 技术以及基于其体系结构的 SSH 框架的研究与分析。

2.1 J2EE 技术架构

2.1.1 J2EE 架构分析

J2E 即 Java2 平台企业版 (Java 2 Platform Enterprise Edition)。J2EE 的本质是技术规范与技术指南，其中囊括了拥有同样标准及规格的各类组件、服务架构与技术层次，这样使用 J2EE 架构搭建的各个平台之间能够做到很好的兼容。这样一来，企业的后台产品信息之间兼容性较差的问题就得以解决，而且让企业内、外部之间也不在出现难以联通的窘迫情景。

J2EE 组件是被 J2EE 规范这样定义的：

(1) 客户层 (Client tier)：又称客户代理，普遍地支持 HTTP 协议，在客户端机器上运行。

(2) 网络应用层 (Internet tier)：包括 JSP，SERVLET 等网络部件，由网络容器运行，在 J2EE 中它运行在 J2EE 服务器上。

(3) 企业组件层 (EJB tier)：企业组件层由 EJB 容器运行，支持 EJB、JMS、JTA 等服务和技术，同样运行在 J2EE 服务器上。

(4) 企业信息系统 (EIS tier)：运行在 EIS 服务器上，企业信息系统包括像财务，CRM 这样的企业内常用信息系统，需要数据库系统的支持是它的特点。

J2EE 四层架构示意图如图 2.1 所示：

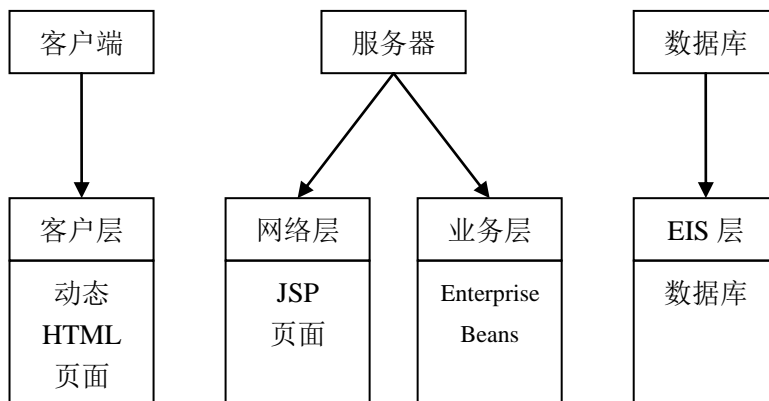


图 2.1 J2EE 四层架构示意图

2.1.2 J2EE 架构的组成技术

EJB 框架是 J2EE 架构的核心技术，强调组成的高可伸缩性，适用于研发大型

的企业应用。在 EJB 体系结构中，应用程序容器或服务器处理所有的底层分配方面的问题与基础结构服务方面的问题，而且 EJB 容器为了能够完美的解决系统性能的问题，通过削减数据库访问次数同时进行分布式处理的方法，提出了专门的系统性能解决方案。

然而 EJB 是为分布式应用而服务的重量级框架，分布式处理方式又是针对大型应用构造的跨平台协作处理方式，但是小型应用不需要采用分布式的解决方案就能够完成所有的需求，如果使用 EJB 就显得太臃肿了，所以我们有时需要采用轻量级框架进行系统开发。

轻量级框架的偏重点在于减小开发的复杂度，当然相应的它的处理能力就会有所削弱（如事务功能弱、不具备分布式处理能力），对于开发中小型企业应用有很高的契合度。使用轻量级框架开发时，一方面应该尽可能的采用基于 POJOs 的方法进行开发，让应用不对任何容器产生依赖，从而提高开发调试的效率；另一方面要利用轻量级框架多数是开源项目这一特点，参考开源社区提供的良好的设计、大量的快速构建工具以及许多现成可供参考的开源代码，加快项目的开发进度。比如目前 Struts+Spring+Hibernate 已经成为许多开发者开发 J2EE 中小型企业应用所偏爱的一种架构^[6]。

2.2 Struts 框架

2.2.1 MVC 框架

MVC 是 Model View Controller 的缩写，代表着模型—视图—控制器这种设计方式，是一种软件设计的典范。主要是采用一种业务逻辑和数据显示分离的方法来进行代码组织，将业务逻辑聚集到一个部件里面，数据的交互围绕在界面和用户之间，能在改进代码和进行个性化定制的同时保留原有的业务逻辑，从而减少工作量。

（1）视图：

视图是用户交互界面的统称。对以往的 WEB 应用程序来说，由 HTML 元素组成的界面就是视图。而对 MVC 来说，视图的含义更为丰富，而且能为应用程序处理很多不同的视图。作为视图来讲，其自身并不具备处理业务的能力，不论所传输的数据是何种类型，制单纯作为一种允许用户操纵并输出数据的方式^[10]。

（2）模型：

与业务逻辑有关的数据以及处理方法的封装就是模型。在 MVC 的三个部件中，模型主要的作用就是处理业务逻辑。在一般情况下会用相应的组件对象来操作数据库，从模型得到的数据是独立的，即模型和数据的格式并没有关系。因此模型

可以提供数据给复数个视图，同时由于模型使用的代码只要编写完成就能被多次使用，这样也可以降低了程序的重复性^[10]。

（3）控制器：

控制器的功能在于根据收到的指令来调用模型和视图从而完成该指令。当点击页面中的按钮进行 HTML 表单传送时就会使用到控制器的功能，控制器自身并不具备处理数据的能力，它只是通过收到的请求来判断调用什么模型来响应请求，之后再判断使用什么视图显示接收到的数据^[10]。

2.2.2 Struts 框架原理

Struts 在 MVC 的基础上更进一步，实现 MVC 时加入了 Servlet 和 JSP 标记（属于 J2EE 规范）。可以说 Struts 在继承了 MVC 的各项特性的同时，又发挥了 J2EE 的优势，使自己相对于 MVC 来有了更优异的性能。

Struts 的工作原理：

模型：模型是多个组件构成的结合体。组件主要分为三种：表单组件、控制器组件以及一般组件。表单组件的功能是将客户端发来的用户请求信息封装起来。控制器组件的功能则是获取从核心控制器程序传递而来的表单信息，随后再调用相关的业务组件对表单信息进行对应的处理。一般组件提供相应业务逻辑方法^[15]。

视图：主要由 JSP 编写的页面组成，为用户提供交互用的界面。包括 HTML 文档、JSP 客户化标签、脚本程序、及多媒体文件等^[15]。

控制器：在 Struts 框架中，作为控制器使用的组件是 ActionServlet。控制组件的主要功能是提供接口用以处理请求信息，比如过滤及转发请求至相应的动作类（这些动作类都是 Action 类的子类）。另外，控制组件也要把相应的请求参数填充到表单组件中，并发送给相应的动作类，之后，动作类再去进行业务逻辑的实现。处理完业务逻辑后，控制权会传给 JSP 页面，由 JSP 页面完成界面显示。控制器的控制逻辑通过 Struts.xml 配置文件进行配置^[15]。

Struts 的工作流程：

（1）服务器启动后，根据部署描述文件（web.xml）初始化 Struts 的核心控制器 ActionServlet，ActionServlet 会根据请求和配置信息找到相应的 Action 和 ActionForm，读取 Struts-config.xml 文件内容到内存；

（2）在确定要调用的 Action 和 ActionForm 后，就将请求中包含的值填充到 FormBean 中，同时 ActionServlet 还需要准备好 ActionMapping 参数；

（3）Struts 将请求分发至相应的 Action 去处理，ActionMapping 参数、Request 参数和 Response 参数都作为参数传递给 execute()方法去执行；

（4）Action 调用业务逻辑方法，得到返回值，并返回 ActionForward 对象；

(5) 此时，控制权重新回到 `ActionServlet`，`ActionServlet` 会根据 `ActionForward` 返回的对象转发到相应的 JSP 页面；

(6) 处理结果返回给浏览器。

Struts 工作流程图如图 2.2 所示：

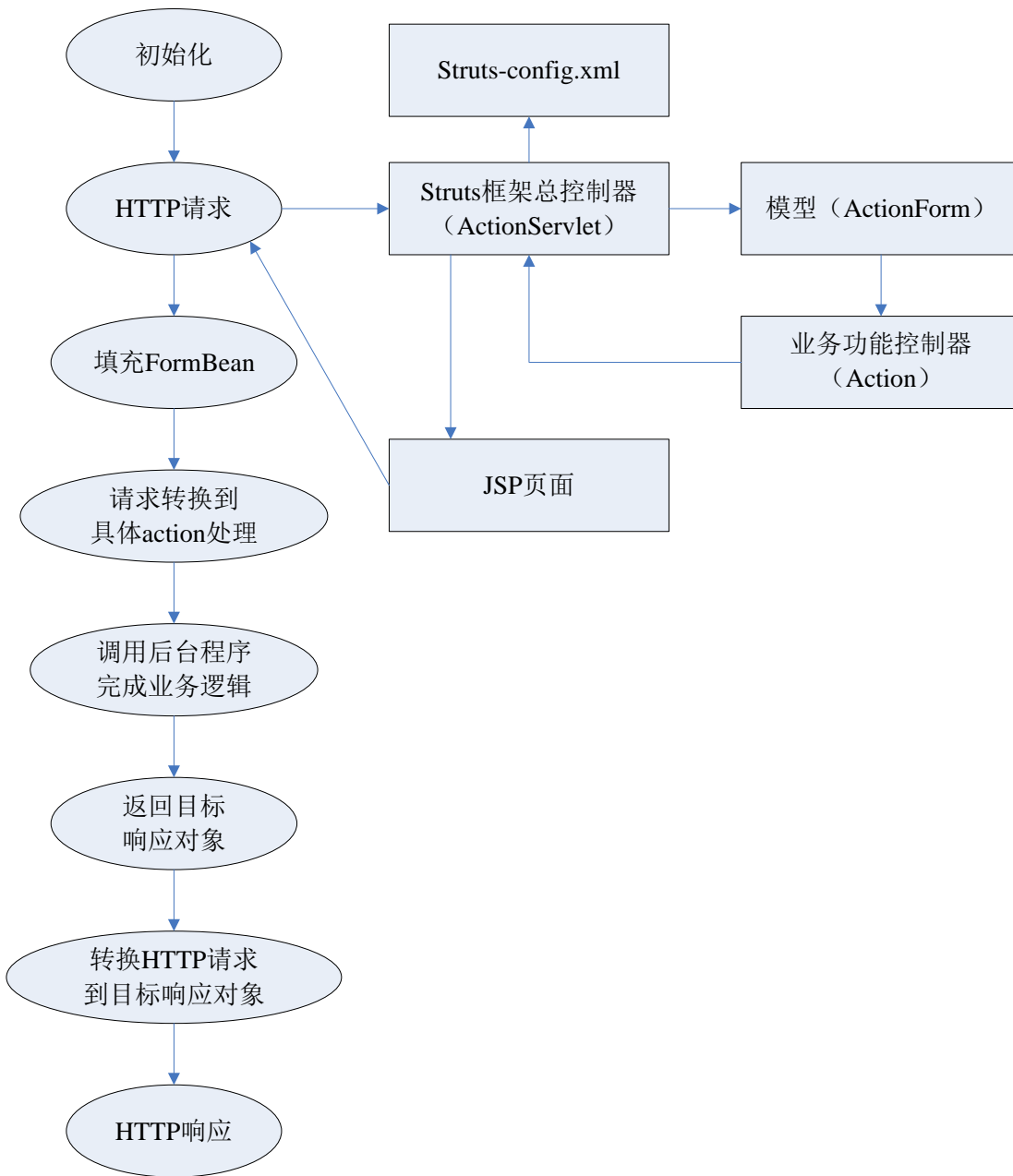


图 2.2 Struts 工作流程图

2.3 Hibernate 框架

2.3.1 Hibernate 框架特性

Hibernate 框架是一种对象关系映射框架，通过对 java 数据库连接进行轻量级

的封装，从而利用面向的对象方式来与数据库交互数据。不但给出了在数据表和类之间的映射关系，还拥有查询和恢复的功能。与通过 SQL 语句来操纵数据库相比，Hibernate 框架能够有效地减少操纵数据库的工作时间。此外 Hibernate 框架能够使用代理方式精简类的载入过程，不但减少了使用 HQL 从数据库读取数据的程序编写量，而且节省了开发的成本。另外 Hibernate 框架还能够与多种服务器集成使用^[18]。

2.3.2 Hibernate 框架运行流程

(1) 连接数据库：

通过 configuration 类读取并解析配置文件 hibernate.cfg.xml，配置文件中定义了连接数据库的相应信息，包括 java 数据库连接驱动、数据库用户名、数据库密码等。获取所需信息后利用 sessionFactory 类生成 session 从而连接上数据库。

(2) 数据库写操作：

添加、修改、删除操作即为数据库的写操，在添加简单的 JAVA 持久对象时，Hibernate 框架通过的对应的事件监听器进行处理。首先，Hibernate 框架读取映射文件得到对象相应的数据表名，以及对象属性相应的列名，然后，借助反射机制获取相应对象的参数信息，最后，将这些信息组成相应 SQL 语句完成数据库写操作。

(3) 从数据库中载入对象：

利用 session.load()方法可以向数据库中载入对象，通过设置对应参数可以决定是否使用延迟加载来减少内存使用，若使用延迟加载，那么返回的对象就会是一种属性都是空（NULL）的代理类（除主键）。要使用返回的类的信息时，Hibernate 框架先会在缓存里查找，如果在缓存中找不到需要的数据，则会再向数据库提出请求，读取到所需的数据。

(4) 数据库查询操作：

Hibernate 框架拥有多种查询方法，其中使用最多方法是 HQL 查询方法。利用 session.createQuery()方法可以创建 Query 查询对象，其中通过 HQL 语句配置相应参数，Hibernate 框架则解析 HQL 语句，使得其等够被 java 数据库连接识别，若提供了缓存查询功能，则执行查询方法时，Hibernate 框架将首先查询缓存中的数据，若未查询到，再对数据库进行查询操作^[18]。

Hibernate 工作流程图如图 2.3 所示：

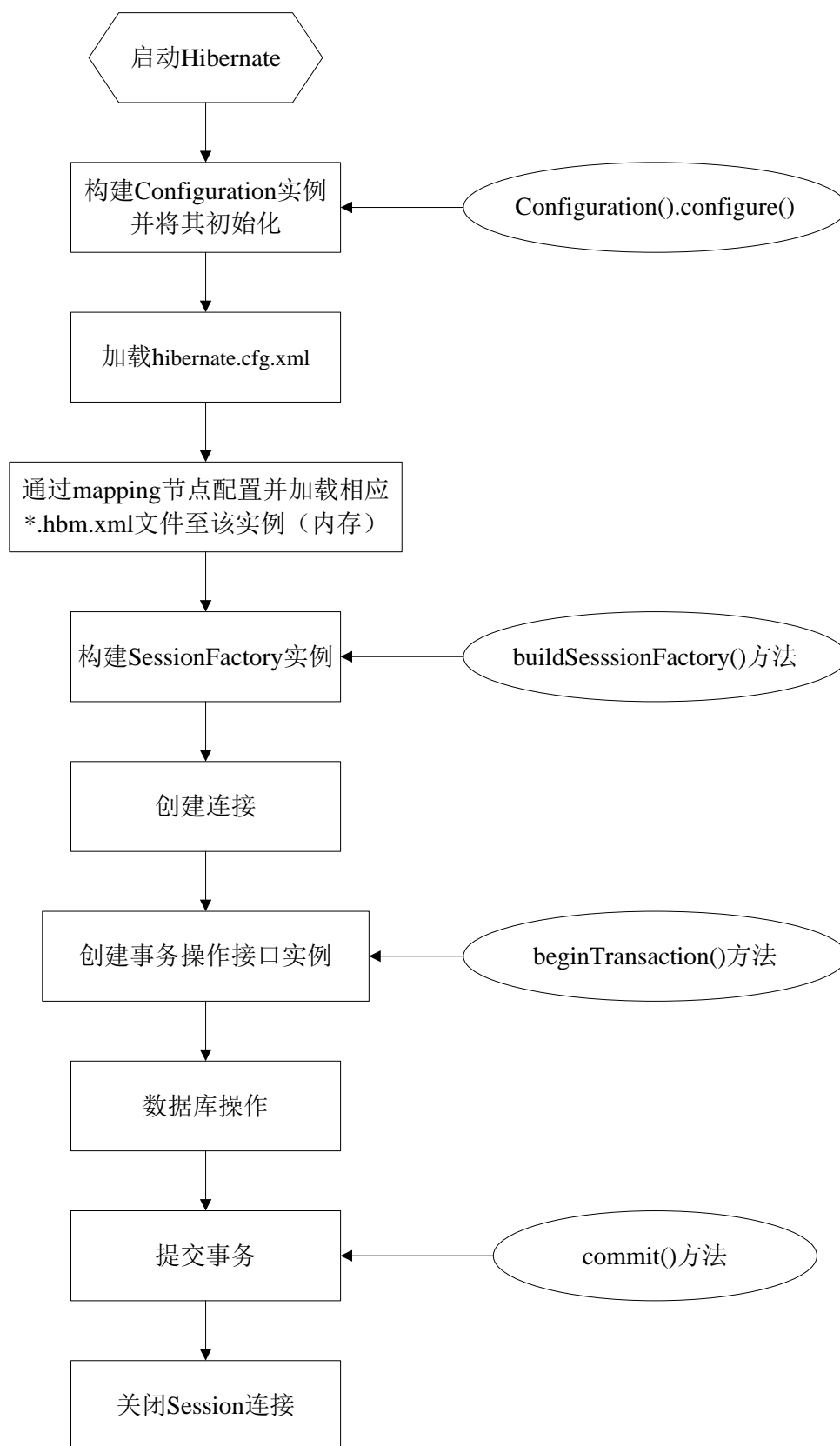


图 2.3Hibernate 工作流程图

2.3.3 Hibernate 框架缓存机制

Hibernate 框架拥有两级缓存，第一级缓存是 Session 对象的缓存。二级缓存则是 SessionFactory 对象的外置缓存。两级缓存都处在数据持久层，主要功能是存放数据库数据的备份。SessionFactory 对象的缓存作用是存储元数据，而 SessionFactory 对象的缓存作用是读取数据。

2.4 Spring 框架

2.4.1 Spring 框架特性

Spring 框架也是一个开源得框架，可以进行复杂的企业级别的应用开发工作，Spring 框架利用一般的 Java 类就可以达到企业级 Java 类的效果。而且 Spring 框架还有更多的用途，不但可以开发服务器端，而且开发其它应用时，也可以达到简单性、松耦合和可测试性^[19]。

Spring 框架特性如下：

(1) 轻量级：Spring 框架所占用的内存空间很少，全部 Spring 框架所形成的 JAR 文件的大小只有一兆多，而且 Spring 框架运行时所耗费的系统资源也同样不多，另外，Spring 框架不是注入式的框架，所构造的对象不依赖于 Spring 框架所拥有的特殊类。

(2) 控制反转：Spring 框架使用控制反转技术来完成程序的松耦合。当程序中使用了控制反转技术时，程序中一个对象需要用到其它对象时，不需要创建或者查找该对象，而是经过被动的方法让其它对象把信息自己传递进来。

(3) 面向切面：Spring 框架支持面向切面编写程序从而达到程序的内聚性。开发程序时，将业务逻辑与系统级服务分割开来，使对象只进行业务逻辑的处理而并不处理系统级的服务，例如日志、事务及安全等。

(4) 容器：Spring 框架作为一个容器，可以管理相应对象的配置信息与持续周期。配置组件信息后，可以用这些配置信息来创建对应的组件实例。所有的组件都能够创建属于自己的实例，而且在使用时可以生成多个实例，同时使他们关联起来。

(5) 框架：Spring 框架能够把简易的应用配置、组合成为复杂的应用。应用对象的组合一般都被配置到 XML 配置文件中。同时，Spring 框架还拥有许多基本的功能，如：事务管理、持久化集成等^[19]。

Spring 工作流程图如图 2.4 所示：

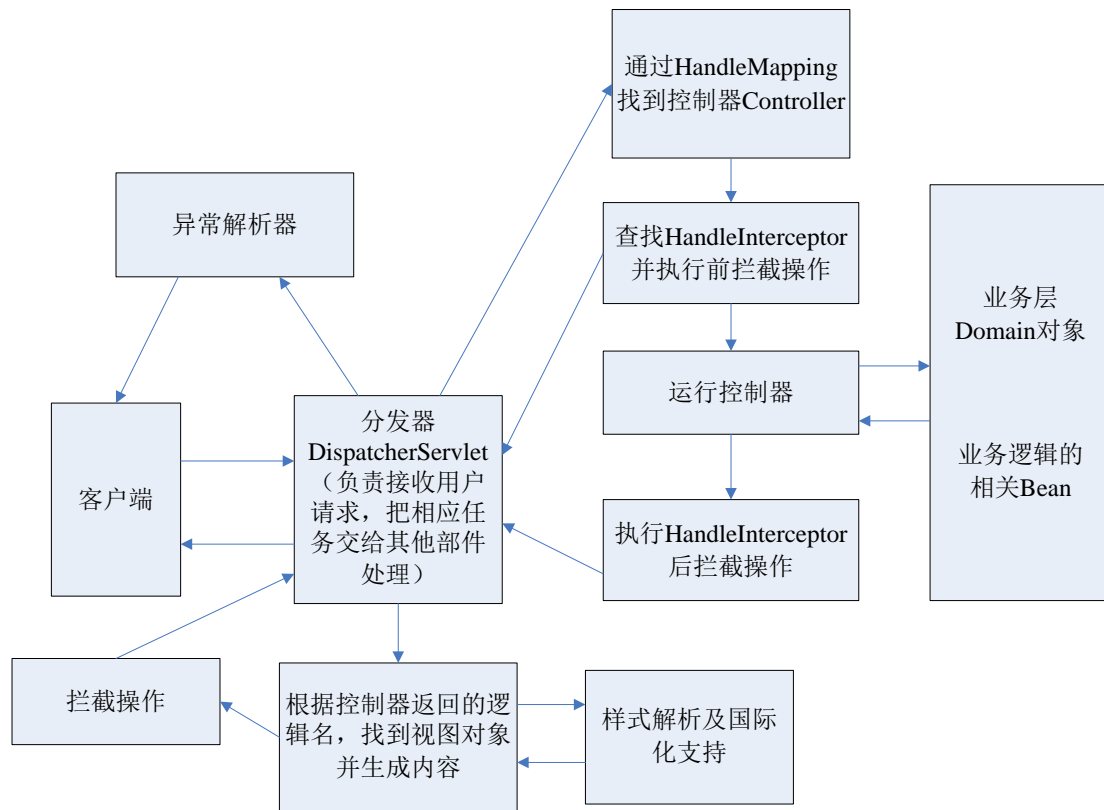


图 2.4 Spring 工作流程图

2.4.2 Spring 框架的反射与动态代理

(1) 控制反转(IOC):

将类与类的依赖关系写在配置文件中，程序在运行时根据配置文件动态加载依赖的类，降低的类与类之间的耦合度。它的原理是在 `applicationContext.xml` 加入 Bean 标记，在 Bean 标记中通过 `class` 属性说明具体类名、通过 `property` 标签说明该类的属性名、通过 `constructor-args` 说明构造的参数。

经由反射，通过类的说明可以得到类的父类、实现的接口、内部类、构造函数、方法、属性，并可以根据构造器实例化一个对象，唤起一个方法。比如取得属性值，修改属性值^[36]。

(2) 面向切面(AOP):

该机制来自于代理模式，代理模式有三个重要的组成部分，分别是通用接口、代理、真实对象。代理与真实对象实现的是同一接口，将真实对象作为代理的一个属性。代理的作用是向客户端公开，当客户端调用代理的方法时，代理找到真实对象，调用真实对象的方法，在调用真是对象前后提供相关的服务，如事务、安全、日志^[36]。

2.5 SSH 框架整合

在 J2EE 体系结构包括表示层、业务逻辑层与数据层, Struts、Spring、Hibernate 三个框架相应的负责完成不同的层的功能。

2.5.1 表示层

表示层由 Struts 负责, 通过表单控制类获取页面中提交的表单数据, 随后经过控制器的操作, 跳转到所对应的页面。所需参数在配置文件中定义, 由核心控制器进行加载。

2.5.2 业务层

业务层由 Spring 负责管理, 为控制器提供调用的程序接口, 并且提供数据持久层的数据访问接口, 起到了整个程序框架 Java 组件的管理作用, 还可以进行事务管理关联数据库。

2.5.3 数据层

Hibernate 负责数据持久化层, 进行数据库的添加、删除、修改、查询操作而且拥有数据映射功能。利用配置文件将 Java 组件和数据库中对应的表关联起来。之后给出数据访问接口。

2.5.4 SSH 框架工作流程

在 SSH 系统中, 对象的调用过程主要表现为: 页面——控制器——业务逻辑——数据访问接口——数据库。

运行流程: 表单组件获取提交的数据信息, 控制器从表单组件中得到数据之后将其转化成一般的 Java 对象, 再调用业务逻辑进行具体业务处理。处理完成后根据配置文件跳转至相应页面。另一方面, 这些一般的 Java 对象会通过调用数据访问接口的方式, 与数据库交互。

Spring 方面, AOP 管理事务控制, IOC 管理各个组件的耦合, DaoTemplate 作为常规持久层的快速开发模板。

Struts 方面, 提供控制层 Action, 页面标签和 Model 数据, 调用业务层。

Hibernate 方面, 负责数据库和对象的映射, 负责 DAO 层(Data Access Object: 数据访问)。

Spring 整合 Hibernate 和 Struts, 只要在配置好 applicationContext.xml 后, 在 Struts 的 action 中直接调用就可以了。Hibernate 访问数据库的操作都在 Spring 中实现了, Spring 的调用又在 Struts 的 action 中实现了, 这样 SSH 框架就完成了整合。

2.5.5 SSH 框架特性

SSH 框架是由三个框架整合而成的，是现今主流的 WEB 开源框架，用于构建灵活、易于扩展的多层网络应用程序。

Struts 提供了系统的大部分功能，构建 MVC 架构，提供视图及控制器的相关程序，进行页面跳转。Hibernate 框架与数据库交互数据。Spring 一方面作为一个轻量级的 IOC 容器，负责查找、定位、创建和管理对象及对象之间的依赖关系，另一方面能使 Struts 和 Hibernate 更好地工作。

SSH 框架结构图如图 2.5 所示。

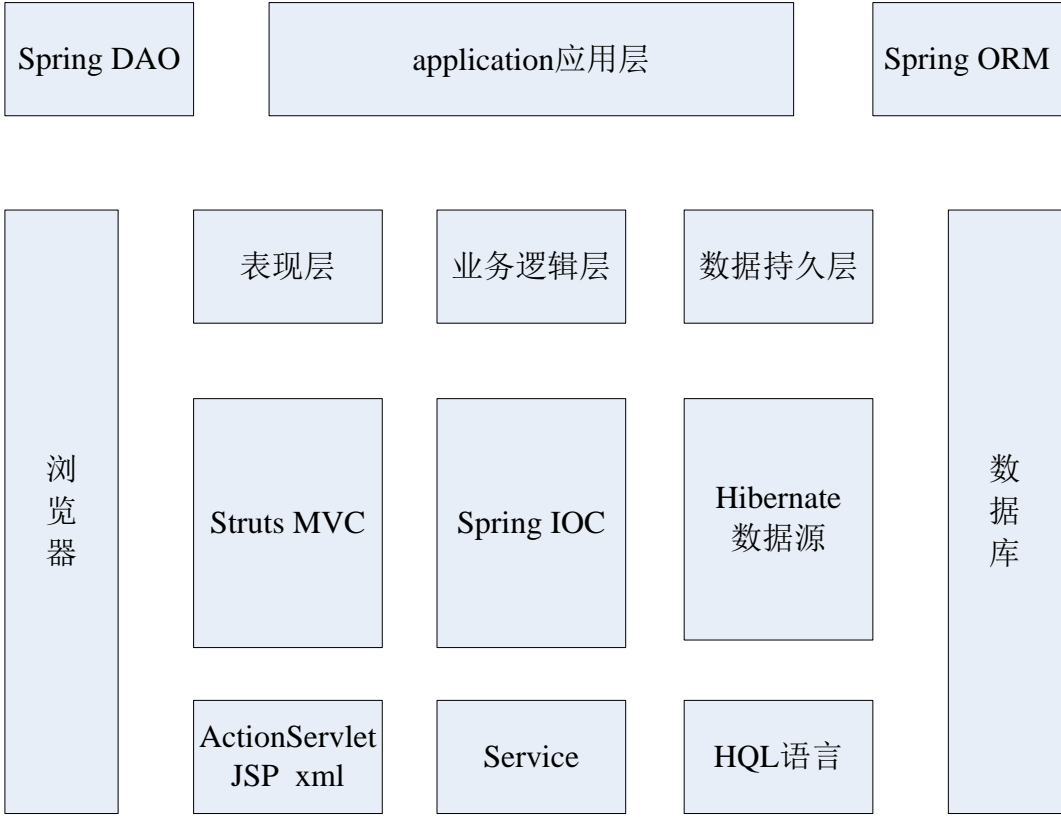


图 2.5 SSH 框架结构图

第三章 系统总体结构分析与设计

本章对系统的总体结构进行了分析,介绍了系统软件模块的设计方案,并对各功能模块进行了大致的说明。

3.1 软件体系结构分析

3.1.1 C/S 结构分析

即 Client/Server (客户机/服务器) 结构,是大家熟知的软件系统体系结构,通过将任务合理分配到 Client 端和 Server 端,降低了系统的通讯开销,需要安装客户端才可进行管理操作。客户端和服务端程序不同,用户的程序主要在客户端,服务器端主要提供数据管理、数据共享、数据及系统维护和并发控制等,客户端程序主要完成用户的具体业务^[24]。

3.1.2 B/S 结构分析

即 Browser/Server (浏览器/服务器) 结构,是随着 Internet 技术的兴起,对 C/S 结构的一种变化或者改进的结构。在这种结构下,用户界面完全通过 WWW 浏览器实现。客户端基本上没有专门的应用程序,应用程序基本上都在服务器端^[24]。

3.2 系统结构分析与设计

3.2.1 系统总体结构设计

系统的主要功能由系统管理模块实现,而管理员模块、教师模块与学生模块作为用户功能模块对主模块进行功能继承扩展。系统结构图如图 3.1 所示:

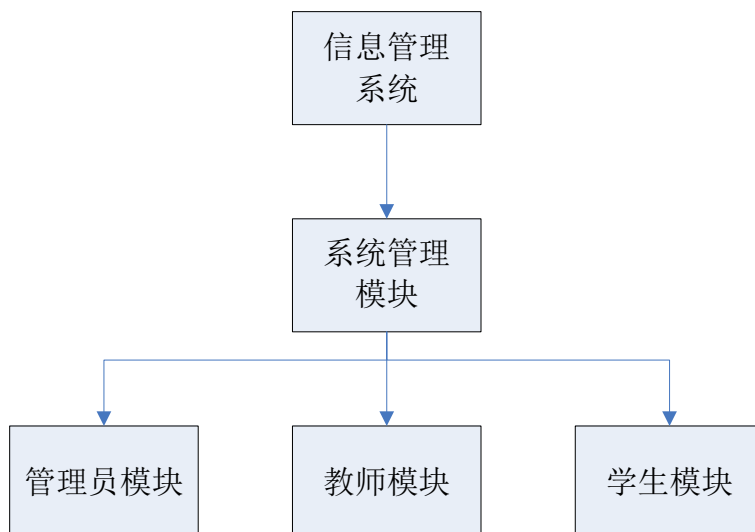


图 3.1 系统结构图

3.2.2 系统管理模块结构分析

系统管理模块由用户管理子模块，信息管理子模块，系统安全子模块，系统维护子模块四个子模块组成。管理模块结构图如图 3.2 所示：

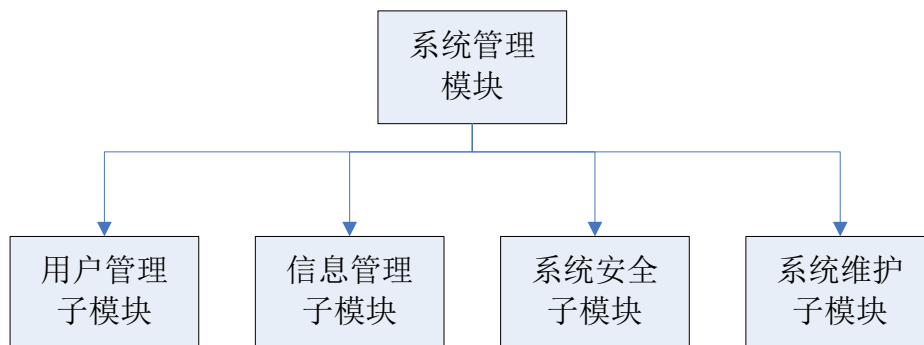


图 3.2 管理模块结构图

(1) 用户管理子模块：负责对系统用户的信息进行增添、删除、修改的操作。拥有管理员权限的用户可对所有用户的信息进行操作，而其它权限的用户，即教师与学生，则基本只有修改本用户密码的功能。

(2) 信息处理子模块：负责系统信息的显示，存储并且对信息进行处理，使各用户功能模块的信息产生必要的关联，实质上就是对数据库进行操作。如：课程信息就需要在教师模块与学生模块之间产生对应关系。

(3) 系统安全子模块：负责保证系统中数据的安全，系统启动后就会在后台开始运行。主要采用的安全策略是：用户权限等级控制（使不同用户依据权限使用不同的功能）、IP 地址访问限制（只有备案过的 IP 地址能够访问系统，减少不明 IP 访问系统带来的风险）、数据库数据加密处理（使数据通过加密后再由网络传输，防止数据遭到修改和窃取）。

(4) 系统维护子模块：保证系统运行稳定，主要是进行系统数据的维护。包括系统中主要数据的定期正常更新外，还有一些数据需要进行不定期的更新、调整，以及数据内容的增加、数据结构的调整。此外，还有数据的备份与恢复等。

3.2.3 管理员模块分析

管理员模块主要功能都是由系统管理模块提供，换言之该模块其实是起到一个系统管理模块的借口的作用，实现系统的具体管理。

3.2.4 教师模块结构分析

教师模块除了继承系统管理模块中户管理子模块，信息管理子模块的部分功能外，还加入了公告发布子模块、课程管理子模块，成绩管理子模块。教师模块

结构图如图 3.3 所示:

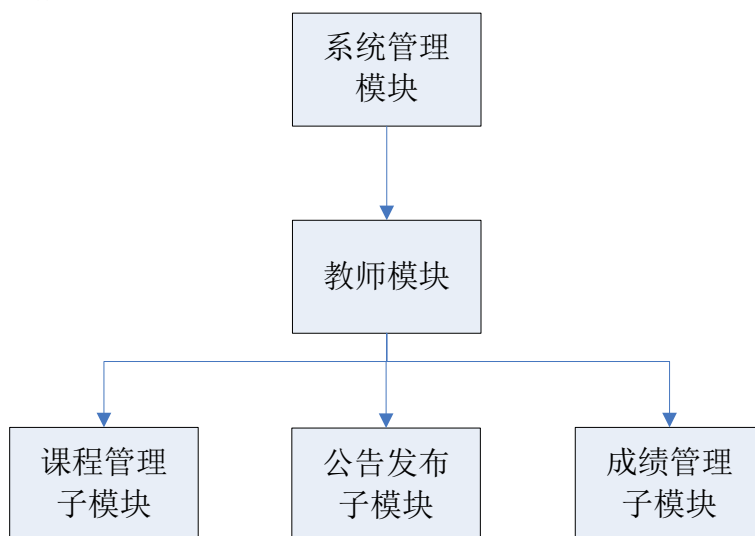


图 3.3 教师模块结构图

(1) 课程管理子模块: 主要负责查看课程信息, 调整该课程的学生人数、教授地点, 处理该课程的作业等。

(2) 公告发布子模块: 向数据库中添加公告信息, 并且可以上传文件, 用文件数据校验保证传输的可靠性。

(3) 成绩管理子模块: 管理学生的课业成绩, 比如成绩的录入, 更新。

3.2.5 学生模块结构分析

学生模块除了继承系统管理模块中户管理子模块, 信息管理子模块的部分功能外, 还加入了选课子模块, 作业提交子模块。学生模块结构图如图 3.4 所示:

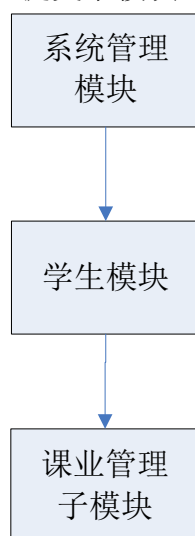


图 3.4 学生模块结构图

课业管理子模块: 显示对应于当前用户的可选课程信息, 并且拥有搜索过滤的功能, 只显示定制课程。在选课时间内, 可对选课信息进行修改、更新。上传

文件的功能模块，加入了校验功能，保证文件传输的可靠性。

第四章 系统软件详细设计

本章详细分析了系统软件部分的设计，包括数据库的设计；数据加密算法的设计；软件模块的功能结构及流程图设计。

4.1 系统数据库详细设计

数据库是信息管理的核心，决定数据存取的效率，因而数据库中的数据存储空间决定了系统的数据维护、检索的易用性。本系统采用在 WEB 应用方面是最好的关系数据库管理系统应用软件——MySQL。它具有体积小、速度快、灵活性高、总体拥有成本低的优点。

通过分析系统的信息构成，将系统分为了用户信息表、教师信息表、学生信息表、公告信息表、课程信息表、学生课业成绩信息表、作业信息表。

(1) 用户信息表

用户信息表包括账号、用户名、密码（6~30 位）、权限等级（0：管理员、1：教师、2：学生）。用户实体 E-R 图如图 4.1 所示，用户信息表如表 4.1 所示：

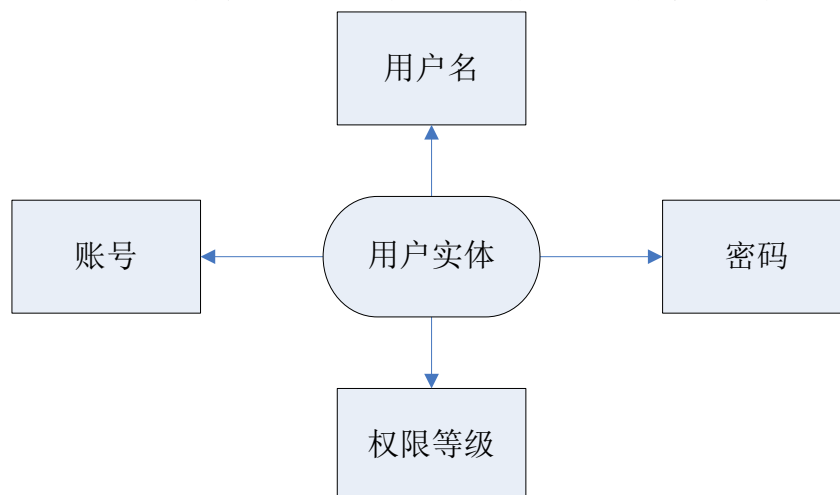


图 4.1 用户实体 E-R 图

表 4.1 用户信息表

字段名	数据类型	是否为空	是否主键	默认值	描述
ID	int(10)	NO	YES	0	账号
name	varchar(30)	YES		NULL	用户名
password	varchar(30)	YES		NULL	密码
grade	int(10)	YES		0	权限等级

(2) 教师信息表

教师信息表包括教师编号、教师头像、姓名、性别、职称、导师类别、学院、专业、简介、电话、邮箱。教师实体 E-R 图如图 4.2 所示，教师信息表如表 4.2 所示：

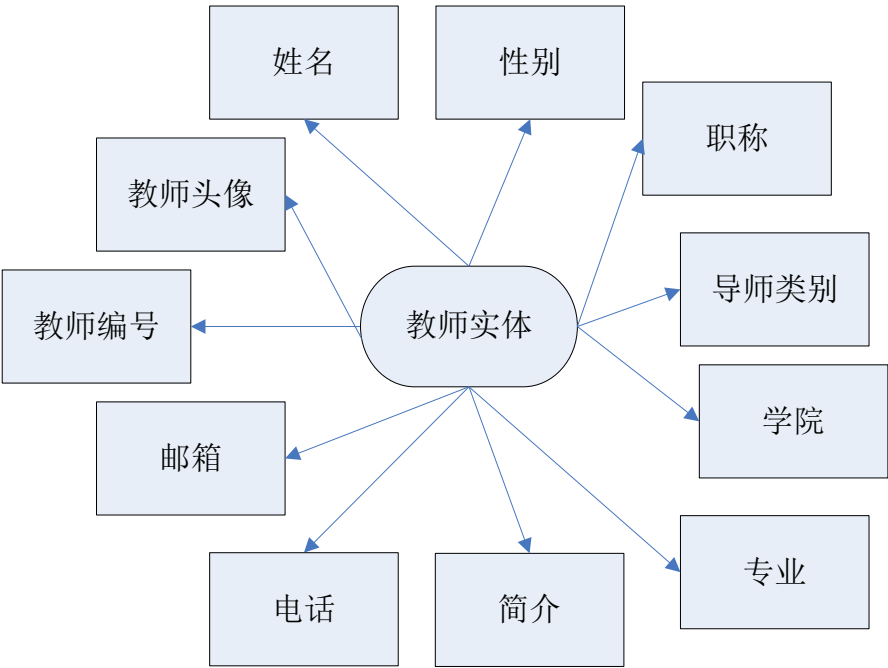


图 4.2 教师实体 E-R 图

表 4.1 教师信息表

字段名	数据类型	是否为空	是否主键	默认值	描述
teacherID	int(11)	NO	YES	0	教师编号
picture	varchar(30)	YES		NULL	教师头像
name	varchar(30)	YES		NULL	姓名
sex	varchar(4)	YES		NULL	性别
pro_ranks	varchar(30)	YES		NULL	职称
type	varchar(30)	YES		NULL	导师类别
academy	varchar(30)	YES		NULL	学院
major	varchar(30)	YES		NULL	专业
introduction	varchar(300)	YES		NULL	简介
phone	int(13)	YES		0	电话
Email	varchar(30)	YES		NULL	邮箱

(3) 学生信息表

学生信息表包括学号、学生头像、姓名、性别、生日、年龄、学院、专业、电话、邮箱。学生实体 E-R 图如图 4.3 所示，学生信息表如表 4.3 所示：

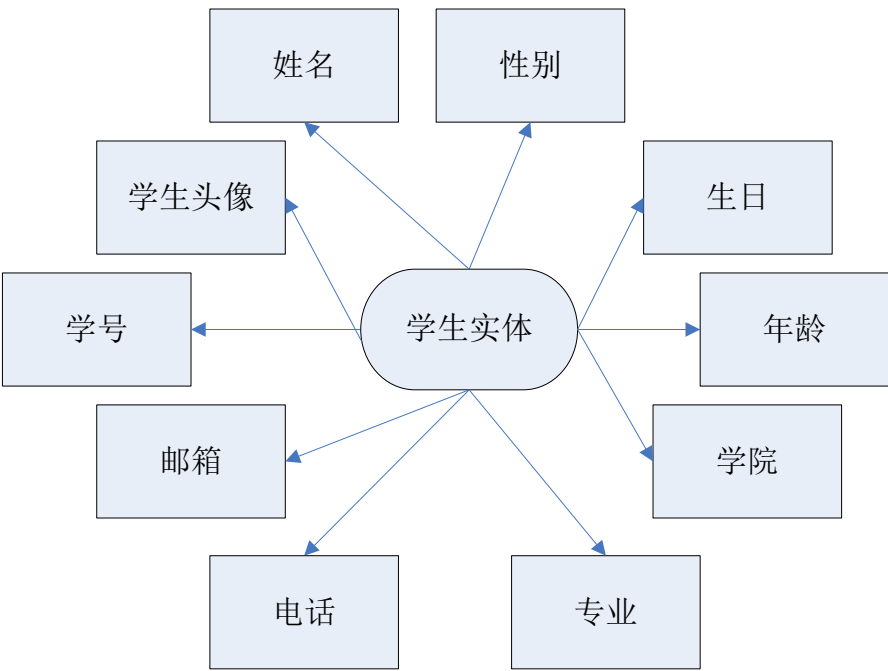


图 4.3 学生实体 E-R 图

表 4.3 学生信息表

字段名	数据类型	是否为空	是否主键	默认值	描述
studentID	int(11)	NO	YES	0	学号
picture	varchar(30)	YES		NULL	学生头像
name	varchar(30)	YES		NULL	姓名
sex	varchar(4)	YES		NULL	性别
birthday	date	YES		NULL	生日
age	int(30)	YES		NULL	年龄
academy	varchar(30)	YES		NULL	学院
major	varchar(30)	YES		NULL	专业
phone	int(13)	YES		0	电话
Email	varchar(30)	YES		NULL	邮箱

(4) 公告信息表

公告信息表包括 ID、公告标题、公告内容、附件、发布时间、发布人。公告实体 E-R 图如图 4.4 所示，公告信息表如表 4.4 所示：

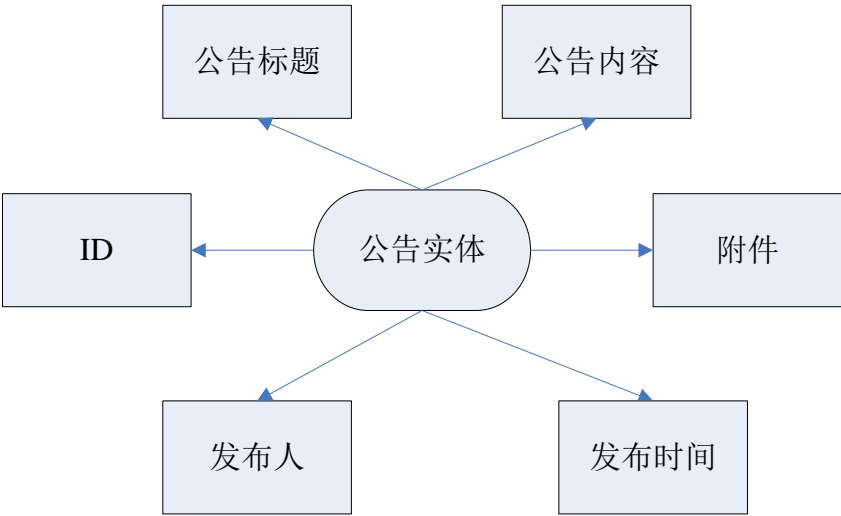


图 4.4 公告实体 E-R 图

表 4.4 公告信息表

字段名	数据类型	是否为空	是否主键	默认值	描述
ID	int(10)	NO	YES	0	ID(自动编号)
title	varchar(30)	YES		NULL	公告标题
context	text	YES		NULL	公告内容
attachment	varchar(30)	YES		NULL	附件
time	date	YES		NULL	发布日期
name	varchar(30)	YES		NULL	发布人

(5) 课程信息表

课程信息表包括课程编号、课程名称、上课教室、上课时间、上课人数、教师编号。课程实体 E-R 图如图 4.5 所示，课程信息表如表 4.5 所示：

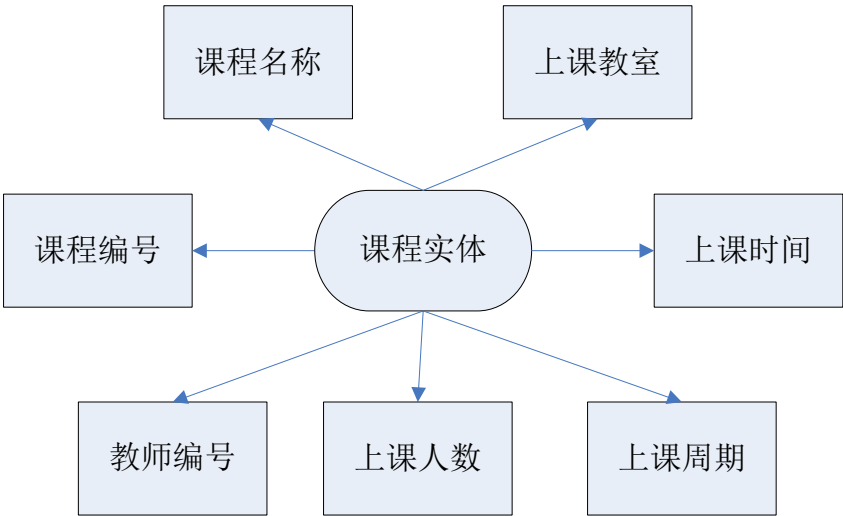


图 4.5 课程实体 E-R 图

表 4.5 课程信息表

字段名	数据类型	是否为空	是否主键	默认值	描述
lessonID	int(10)	NO	YES	0	课程编号
lessonname	varchar(30)	YES		NULL	课程名称
classroom	varchar(30)	YES		NULL	上课教室
number	int(30)	YES		NULL	上课人数
time	date	YES		NULL	上课时间
period	varchar(30)	YES		NULL	课程周期
teacherID	int(10)	NO		0	教师编号

(6) 学生课业成绩信息表

学生课成绩信息表包括 ID、成绩、课程编号、学号。学生课业成绩实体 E-R 图如图 4.6 所示，学生课业成绩信息表如表 4.6 所示：

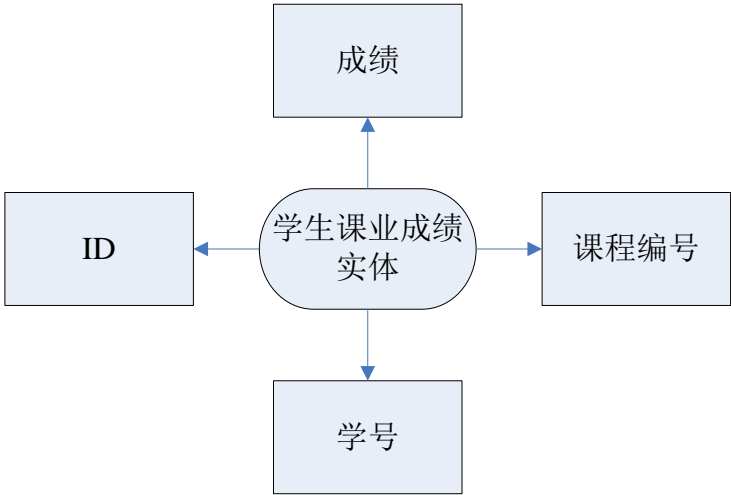


图 4.6 学生课业成绩实体 E-R 图

表 4.6 学生课业成绩信息表

字段名	数据类型	是否为空	是否主键	默认值	描述
ID	int(10)	NO	YES	0	ID(自动编号)
mark	int(30)	YES		NULL	成绩
lessonID	int(10)	NO	YES	0	课程编号
studentID	int(10)	NO		0	学号

(7) 作业信息表

作业信息表包括 ID、作业名、存储路径、课程编号、学号。作业实体 E-R 图如图 4.7 所示，作业信息表如表 4.7 所示：

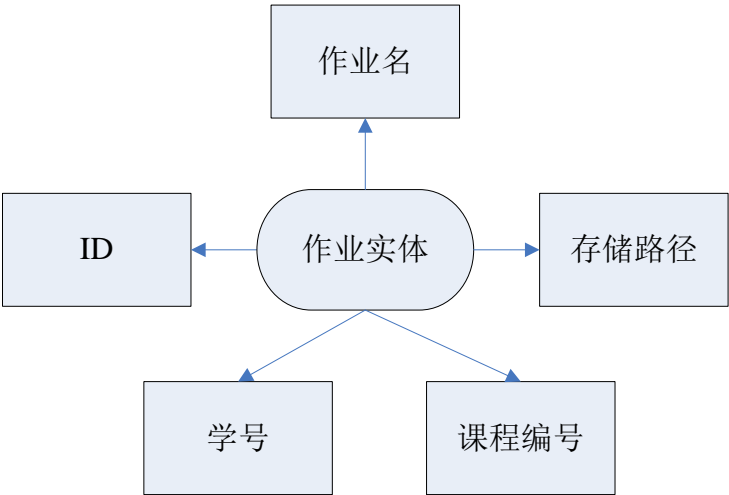


图 4.7 作业实体 E-R 图

表 4.7 作业信息表

字段名	数据类型	是否为空	是否主键	默认值	描述
ID	int(10)	NO	YES	0	ID(自动编号)
assignment	varchar (30)	YES		NULL	作业名
path	varchar(30)	YES		NULL	存储路径
lessonID	int(10)	NO	YES	0	课程编号
studentID	int(10)	NO		0	学号

将所有表录入 MySQL 数据库中，如图 4.8 所示：

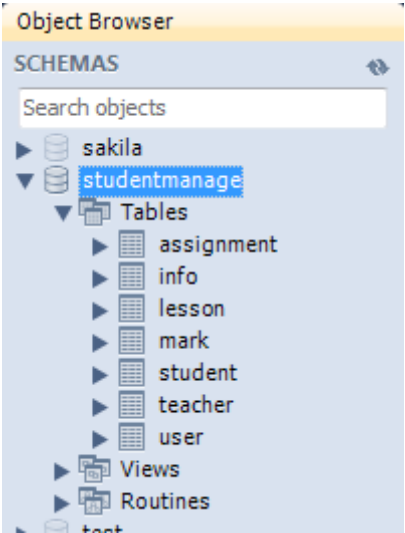


图 4.8 MySQL 数据库数据表分布图

4.2 加密算法设计

4.2.1 数据加密算法分析

对称加密算法拥有算法透明开放、运算容易、消耗系统资源少等特点。发出的数据使用一个密钥通过级加密算法进行加密后，将数据通过网络传送出去。收到数据后，再使用同样的密钥经过算法解密，从而得到真正的数据。其中，密钥采用随机数的方式，从而减少被破解的可能性^[31]。流程图如图 4.9 所示：

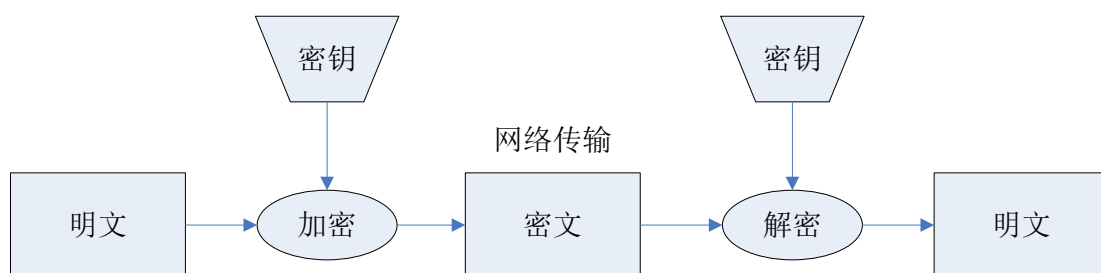


图 4.9 对称加密算法流程图

主流的对称加密算法有：

数据加密算法：英文缩写为 DES（Data Encryption Standard），特点是加密快速，一般用在需加密的数据较多的地方。

三重数据加密算法：英文缩写为 3DES（Triple DES），特点是使用三种不一样的密钥对数据进行加密使得安全性更好。

高级加密算法：英文缩写为 AES（Advanced Encryption Standard），目前最新的加密算法，拥有更快的加密速度与更好的安全性。

因此，根据系统的具体需求，最终采用了 AES 算法对数据进行加密处理。

4.2.2 AES 算法设计

AES 算法的基本步骤是：轮变化、圈变化与密钥扩展。

AES 算法使用的密钥构成是分组密钥，如算法中给出一百二十八位数据信息，那么所需密钥的长度也应该是一百二十八位数据。对数据进行分组加密的轮数用字母 An 来表示。因为用户提供的密钥不会是长度非常长的数据，加密运算过程中就需要使用扩展密钥把提供的密钥长度加长，这个扩展密钥一般与获取数据拥有一样长度，在加密的轮次中加入加密运算^[32]。

a) 圈变化

(1) 字节转换是指在对每个字节进行非线性的变换，也就是能够经过运算得出的二元域的向量空间的映射。

(2) 行移位变换是指将数据分为多行，每行都以一定的偏移量进行数据换位，偏移量一般是依据参数 Aof 而定的。

(3) 列混淆变换是把状态中的每一列看作有限域 GF(2⁸)上的乘法，公式如式 4-1 所示，矩阵表示如式 4-2 所示：

$$b(x) = (03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02) \cdot a(x) \bmod (x^4 + 1) \quad (\text{式 4-1})$$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (\text{式 4-2})$$

(4) 轮密钥加变换就是逐字节相加，有限域 GF(2⁸)上的加法是 mod2 加法，即应将处于圈密钥状态中的字节按位“异或”。

b) 轮变化：

轮变化次数关联于分组长度，如表 4.8 所列。

表 4.8 轮变化次数表

AES 类型	密钥长度/字	分组大小/字	轮变化数/次
一百二十八位	四	四	十
一百九十二位	六	四	十二
二百五十六位	八	四	十四

c) 密钥扩展：

AES 算法首先将用户提供的密钥 “AESKey” (密钥长度为 An)，经过扩展密钥可获取长度为 4*(An+1)的扩展密钥。

AES 的加密与解密流程如图 4.10 所示：

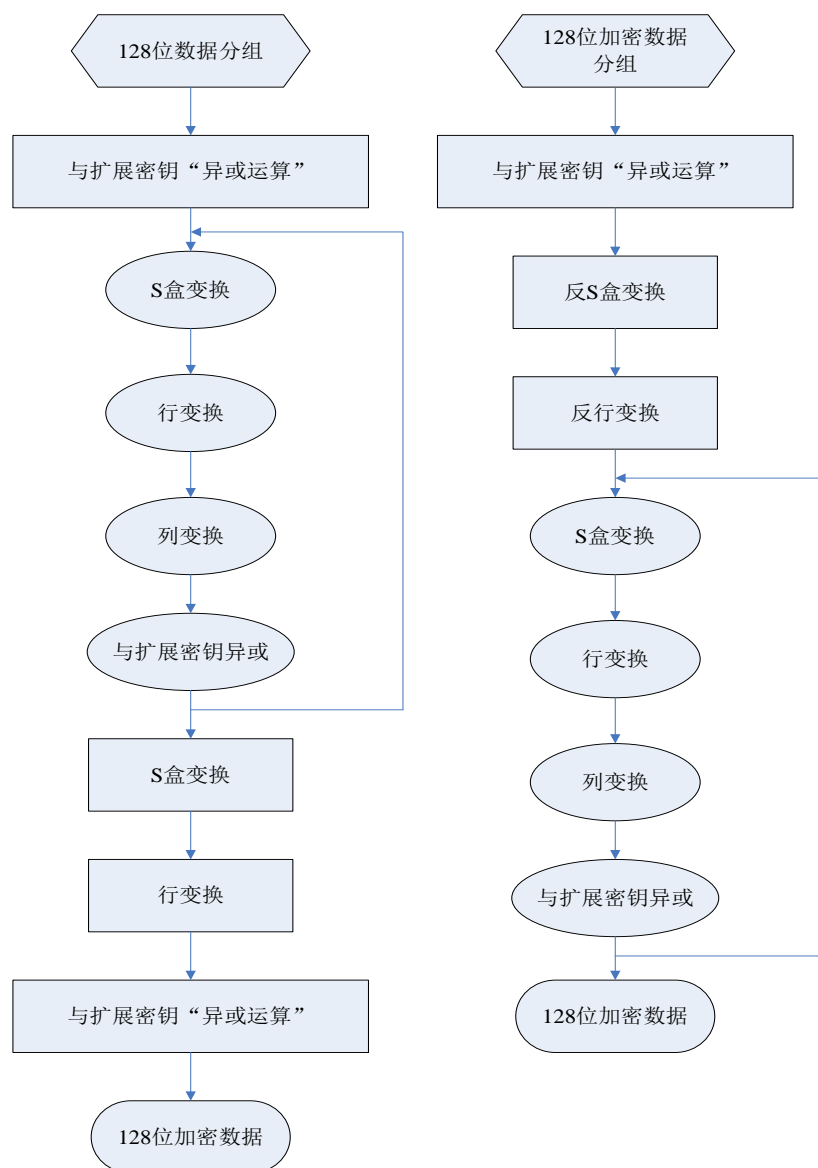


图 4.10 AES 算法流程图

4.3 系统软件详细设计

4.3.1 管理员模块详细设计

管理员模块的功能包括用户信息处理（查询用户信息、添加用户信息、修改用户信息）；系统维护（数据备份、数据恢复、冗余数据清理）。模块流程图如图 4.11 所示：

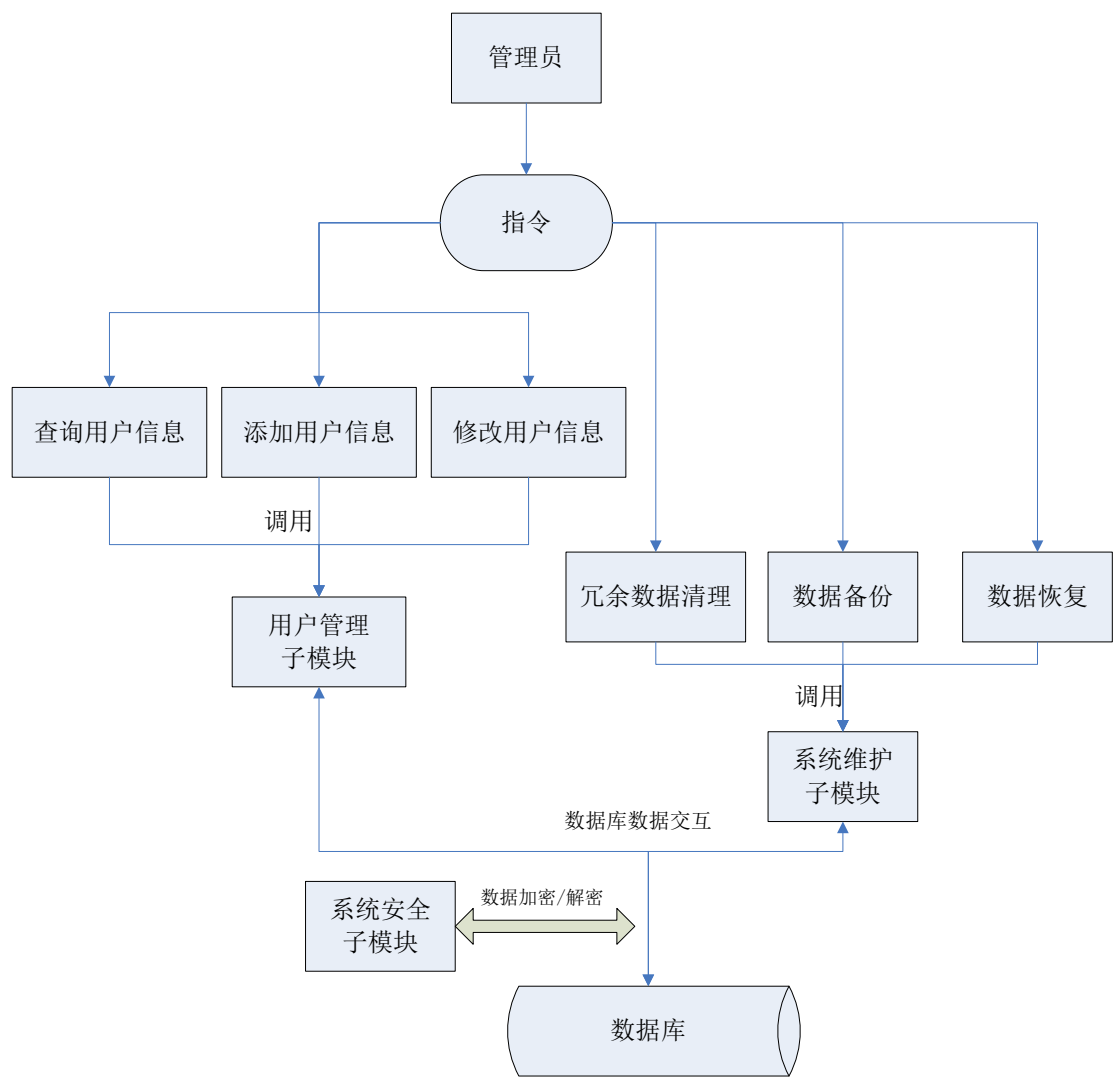


图 4.11 管理员模块流程图

添加用户信息：管理员可以增加一个新用户。添加时新用户的信息会与数据库中现有的用户信息进行比对，管理员首先将要添加的用户信息与现有的数据库中所存储的用户信息进行比对，如果不存在相同的信息则新用户会成功添加进数据库；如果已经存在相同的信息则新用户会添加失败，需要重新输入。

查询用户信息：可以查看数据库中所存储的用户信息，管理员在查看信息时可以通过设置信息过滤条件，去除那些不需要的信息，用以显示所需的信息，避

免盲目的查找。如果不设置过滤条件，系统默认显示全部用户信息，并以用户 ID 升序排列。

修改用户信息：管理员可以修改所选用户的信息。同样可以设置信息过滤条件找到需要修改的用户条目，可以修改的项目包括用户名与密码，修改的内容需要符合数据库的设定规则，同时修改后的内容不能与现有的信息重复，否则系统会取消这次修改操作。

数据备份：管理员需要设置数据库备份计划完成数据库数据的备份。需要设置有备份类型与备份计划。备份类型包括完全备份、差异备份、事务日志备份、文件备份。备份计划则是设置具体的时间来进行数据库备份。

数据恢复：管理员需要在数据库出现故障时，通过此模块将保存在数据库中丢失的电子数据进行抢救和恢复。模块采用的是回滚恢复这一技术手段，使用完整的数据库备份和日志相结合，可以使一个数据库或者被选择的表空间恢复到某个特定时间点，需要与数据备份模块配合使用。

冗余数据清理：管理员需要在数据库产生冗余数据时手动将其清除。可以使用模块中定制的 SQL 语句查询数据库，以找出冗余数据。

4.3.2 教师模块详细设计

教师模块的功能包括用户信息处理（查看教师用户信息、修改教师用户信息）；公告发布（发布、修改公告信息；上传附件）；课程管理（查看课程信息、查看选课学生信息、查看学生作业）；成绩管理（查询成绩信息、提交成绩信息）。模块流程图如图 4.12 所示：

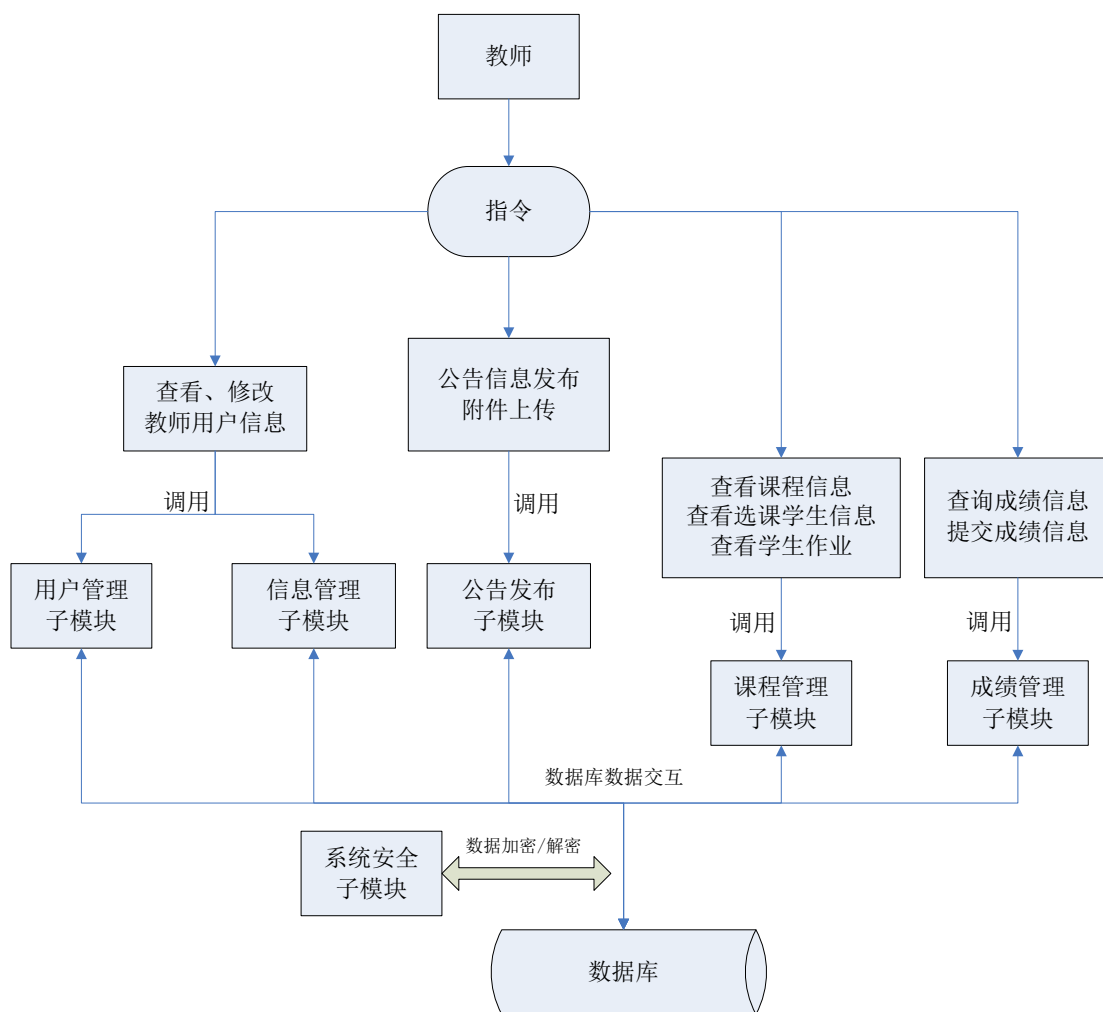


图 4.12 教师模块流程图

查看、修改当前教师用户的信息：教师用户可以查看相应的用户信息。修改相应信息时，修改的内容需要符合数据库的设定规则，否则系统会取消这次修改操作。

发布、修改公告信息：教师用户可以添加新的公告信息，并且可以上传附件（表格、文档等），附件上传完成后，系统将对文件进行校验，确保其在传输过程中未出现异常。如果校验未通过，则系统会提示重新上传。

查看课程信息：教师用户可以查看课程信息，包括学生人数、上课时间及地点等。

查看选课学生信息：教师用户可以查看所有选择自己课程的学生信息，查找时可通过学号精确查找特定学生。

查看学生作业：教师用户可以一次性下载已上传的所有作业文件。如有学生未上传作业，则系统会给出提示信息。

查询成绩信息：教师用户通过此模块查询到所有学生的平均分数、最高分、最低分、以及成绩的总体分布。

提交成绩信息：在学期末为选择自己教授课程的学生打分。系统默认情况下，所有的学生分数均为 0。

4.3.3 学生模块详细设计

学生模块的功能包括用户信息处理（查看、修改当前学生用户的信息；查看教师信息；查看成绩信息；查看公告信息）；课业管理（选课、提交作业）。模块流程图如图 4.13 所示：

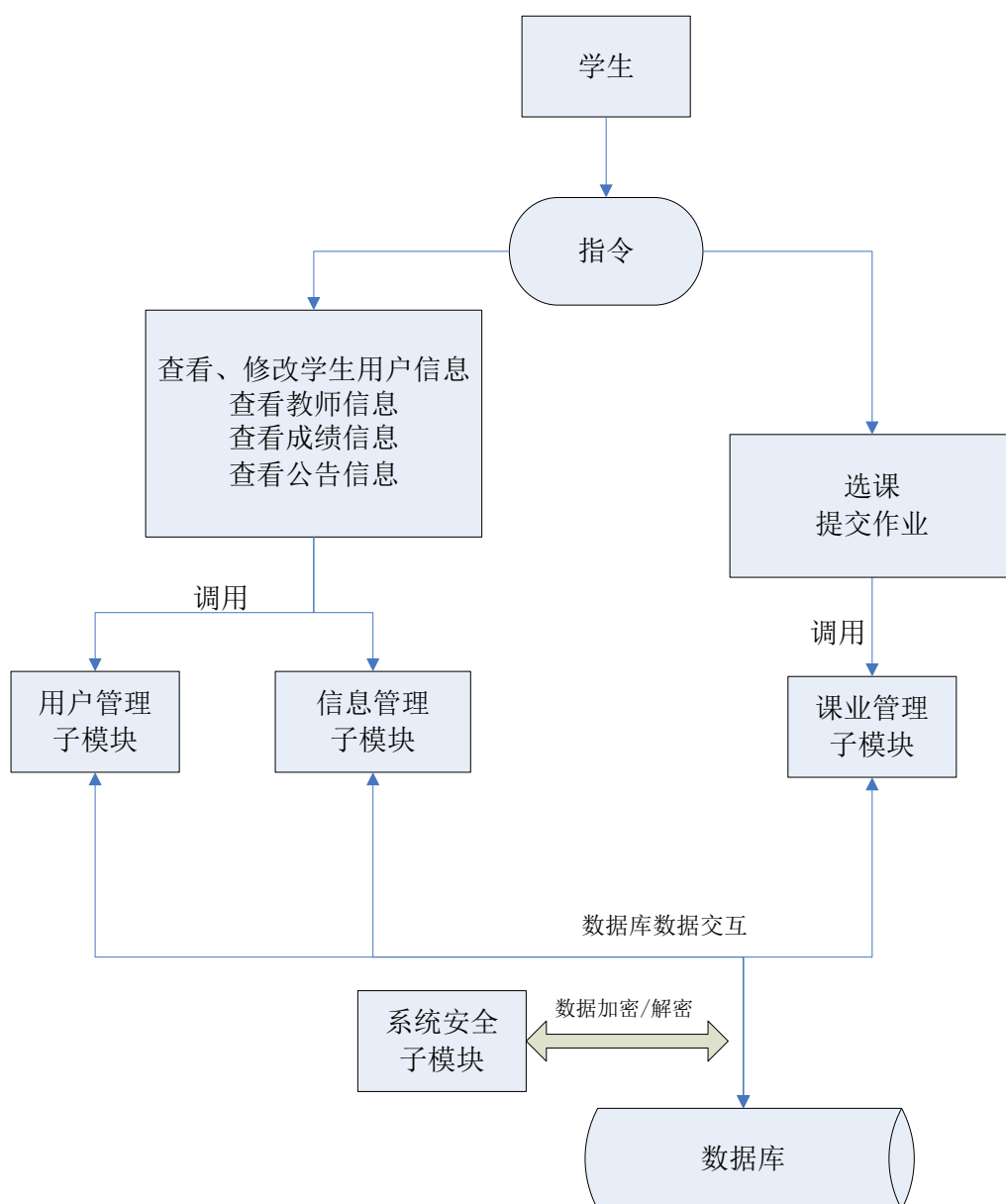


图 4.13 学生模块流程图

查看、修改当前学生用户的信息：与教师模块功能一致。

查看教师信息：学生用户可以查看到所有存在的教师信息，查找时可通过设置过滤条件（教师编号、所教课程等）精确查找。

查看成绩信息：学生用户可以查看到自己的成绩，及排名情况。如有需要补考的科目，系统将会显示提示信息。

查看公告信息：学生用户可以查看到所有发布的公告信息并且下载附件。

选课：学生用户可以查看到所有可选的课程信息，并选择想要修习的课程。查找课程信息时时可通过设置过滤条件（课程编号、上课时间等）精确查找。

提交作业：上传作业压缩文件，文件信息要与课程相对应。上传完成时，系统将对文件进行校验，确保其在传输过程中未出现异常。如果校验未通过，则系统会提示重新上传。

第五章 信息管理系统的具体实现

本章首先介绍了系统分层架构的实现及各层次具体的配置方案，之后介绍了各个页面的实现与功能及其层次结构分析。

5.1 系统分层架构的实现

分层是为了隔离各个模块，使其完成各自的功能，达到松耦合的目的。其特点主要表现在：下层为上层提供其所需的服务，但实现的过程，上层无法知晓，从而实现下层的隐藏；层与层之间的依赖性降低，每个模块完成各自的功能，不涉及其他的业务逻辑，从而使得各模块之间得以解耦，依赖性降低；各层之间的依赖性降低，各模块就可以很容易的被替换，如果替换的下层模块提供了上层所需的方法，上层模块将不做任何改动，达到轻松替换每个模块的目的；各层就可以进行各自的映射。以登录页面为例的SSH框架分层结构的运行流程如图5.1所示：

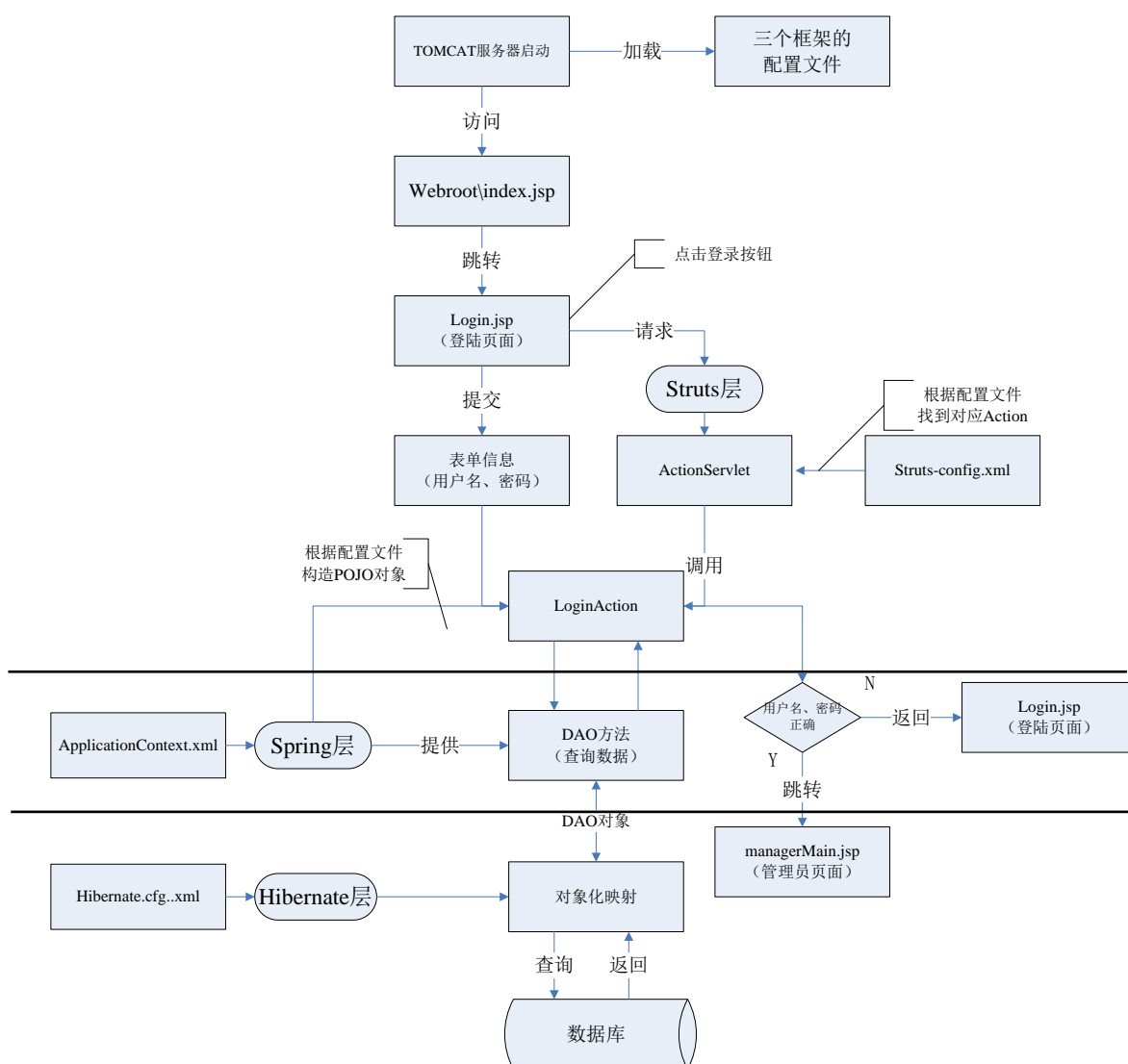


图 5.1 SSH 框架流程图

5.1.1 表示层的实现

表示层由 Struts 框架负责。为系统提供了中央控制器（ActionServlet）和应用控制器(Action)。ActionServlet 是 Struts 框架的核心,采用的是前端控制模式; Action 则负责完成对服务的调用,对 ActionServlet 接受并分发下来请求进行处理。在运用中,我们可以重写 ActionServlet,也可以直接使用,然后继承 Action,使其完成我们要求的功能^[27]。

有了 Struts 提供的控制器,再运用 JSTL 标签、Struts 标签库中的标签配以 JSP 或 HTML 文件,就是 Struts 中的视图。另外,用于视图与控制器之间表单数据传输的 ActionFormBean,也被归于到了视图中。Struts 并没有定义模型层的具体实现。一般情况下,Action 中所使用的服务是由业务层提供的。Struts 中还有不可缺少的配置文件。该配置文件描述了用户请求路径和 Action 映射关系的信息。ActionServlet 通过解析此配置文件得知需把用户的请求发往哪一个 Action^[27]。

Struts 框架配置文件实现如下:

(1) web.xml

只有将需要的 Servlet 参数写入该配置文件,在框架运行时才能够加载所需的 Servlet。Struts2 框架开始运行时,首先就要运行核心控制器,之后核心控制器再统一安排之后的程序运行,而这一操作是通过读取解析该配置文件来实现的。

在配置 Struts2 的核心控制器的参数时,能够先设置控制器的初始化参数,在众多初始化参数中,config 参数、actionPackages 参数、以及 configProviders 参数是至关重要的。

Config 参数: 主要包括一系列配置文件的地址参数,Struts 框架通过解析该参数来获取需要的配置文件。

actionPackages 参数: 主要包括程序所使用的 Jar 包的内容,Struts2 框架通过解析该参数可以获取 Jar 包中的控制器类。

configProviders 参数: 主要包括所需类的参数,Struts2 框架通过解析该参数可以建立相应的类。

配置过滤器元素的时候,可以在过滤器映射项目中设置该过滤器能够过滤的内容。一般情况下将会把所有内容都过滤掉。

(2) struts.xml

Struts2 框架的关键在于配置 struts.xml 文件。在通常情况下,Struts2 框架运行时首先查找该文件,并且将其读取解析其中的参数。该配置文件还能够进行分解,生成多个子文件,这样,Struts2 框架就可以通过读取主文件来调用子文件的内容。可以有效的防止只使用单个配置文件导致的配置文件容量过于巨大而引起的一些不必要的问题^[27]。

Struts2 框架通过读取配置文件中的包属性，管理控制器与过滤器。包中包含有复数个控制器、过滤器以及他们的结合体。通过包属性能够关联多种不同的组件，使其拥有逻辑联系。包属性与 Java 中的对象属性比较相近，不但能够使用继承功能，而且能够被继承，甚至也拥有抽象这一定义。

Struts2 框架通过命名空间的方法进行控制器的操作，在相同的空间中不可以出现名字相同的控制器，反之，在不同的空间中则可以出现名字相同的控制器。Struts2 框架不提供只拥有一个控制器的命名空间，只会提供包属性，通过它来设置控制器所应处的空间。

命名空间把控制器区分成了在逻辑意义上不一样的组件，每个组件都会拥有自己专属的前缀信息。利用命名空间能够消除控制器可能会出现的重名的问题。Struts2 框架的信息参数中具有命名空间的项目，能够依据命名空间的不同而向服务器发出不一样的控制器要求。

Struts2 框架通过读取配置文件参数，可以建立相应的控制器，同时通过调用控制器来进行用户请求的处理工作。即控制器实际上是作为 Struts2 框架的构成基础而存在的，控制器真正负责底层工作的运行。

每一个控制器都会处理与自己相对应的请求信息，在控制器需要处理相应请求信息时，Struts2 框架就将调用该控制器。当然，控制器不可能直接将处理后的信息像是给提请求的用户，而是需要视图组件配合，将信息展示在页面中像是给用户。所以，配置控制器的同时，也要把视图组件与相应控制器的联系信息写入配置文件中。

Struts2 框架利用包属性来关联控制器，所以，控制器的配置信息存放于包信息中，配置控制器信息时由包中的对应子元素来进行的。需要确定的控制器信息主要有控制器名字属性及控制器类属性。该 name 属性既是该 Action 的名字，也是该 Action 需要处理的 URL 的前半部分。

配置文件代码如下：

```
<struts>
<constant name="struts.i18n.encoding" value="utf-8"/>
<constant name="struts.ui.theme" value="simple"/>
<constant name="struts.custom.i18n.resources" value="application"></constant>
<package >
    <action >
        <!-- 配置 action-->
    </action>
</package>
```

</struts>

5.1.2 业务逻辑层的实现

业务逻辑层由 Spring 框架负责,该框架在程序结构中有非常重要的作用,链接了表示层与数据持久层,在程序运行时起到了粘合剂的用途。因为层与层之间的架构是弱耦合的,所以层与层之间的关系是单向链接的,上层的程序无法获知下层的程序,这样就算是上层的程序发生了变化,对于下层程序也是没有任何影响的。在系统设计之初,使用了抽象函数构造程序主体,那么程序间的联系就可以叫做弱依赖联系。

Spring 框架的核心机制是依赖注入,传统情况下当某个实例需要调用另一个实例时,通过调用者来创建被调用者的实例,通常通过 new 方法来创建。而在依赖注入的模式下创建被调用者的工作不再由调用者来完成,因此称之为"控制反转";创建被调用者实例的工作通常由 Spring 来完成,然后注入调用者,所以也称之为"依赖注入"。依赖注入一般有两种方式,分别为设置注入与配置构造注入。设置注入通过 IoC 容器使用属性的 setter 方式注入被依赖的实例,而构造注入通过 IoC 容器使用构造器来注入被依赖的实例。

Spring 框架有两个核心接口: BeanFactory 接口和 ApplicationContext 接口,其中 ApplicationContext 是 BeanFactory 的子接口。这两个接口都可以代表 Spring 容器,作为生成 Bean 实例的工厂,并管理 Spring 中的 bean。bean 是 Spring 中的基本单位,在基于 Spring 的 java EE 工程,所有的组件都被当成 bean 处理^[45]。

Spring 框架配置文件实现如下:

applicationContext.xml:

该文件是 Spring 框架的默认配置文件,主要功能包括:数据源配置、事务管理器配置、aop 注解支持、缓存配置等。

配置文件代码如下:

```
<bean id="dataSource"
    class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName"
        value="com.mysql.jdbc.Driver">
    <property>
        <!--数据库属性配置-->
    </property>
</bean>
<bean id="TransactionManager"
```

```
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="dataSource" />
</bean>
<!-- 集成 hibernate -->
<bean id="sessionFactory"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
  <property name="dataSource" ref="dataSource"/>
  <property name="mappingResources">
    <list>
      <!--hibernate 映射配置文件属性 -->
    </list>
  </property>
  <property name="hibernateProperties">
    <props>
      <prop key="hibernate.dialect">
    </props>
  </property>
</bean>
<!-- 配置 Hibernate 事务策略 -->
  <bean id="txManager"
class="org.springframework.orm.hibernate3.HibernateTransactionManager">
  <property name="sessionFactory" ref="sessionFactory">
  </property>
</bean>
```

5.1.3 数据持久层的实现

数据持久层由 **Hibernate** 框架处理，管理对象到数据库的映射，并提供数据查询和获取数据的方法。在持久层中，我们不仅可以使**用 Hibernate** 进行映射，还可以使**用 iBATIS** 进行 **SQL** 语句与对象之间的映射，此外，也可以直接使用 **JDBC** 对数据库进行操作^[22]。

Hibernate 框架在使用前需要将配置文件的信息参数配置完毕。主要的配置信息包括与 **MySQL** 数据库链接所需的参数，如用户名、密码等。**Hibernate** 框架运行后将会解析配置文件，将这些信息参数读取到程序中，之后由此建立 **Session** 对象，再通过调用该方法实现数据库操作。另外，数据映射参数则会在程序中的

类与数据库交互时，起到数据映射的作用^[22]。

(1) hibernate.cfg.xml:

配置数据库的主要参数包括：数据库方言、数据库连接字符串、数据用户名、数据库密码、数据库驱动。同时导入对象与表的映射文件。

配置文件代码如下：

```
<hibernate-configuration>
<session-factory>
<!--数据库方言（是一个类的全名）与数据库连接信息 -->
<property name="dialect">
org.hibernate.dialect.MySQL5Dialect</property>
<property name="connection.url">
jdbc:mysql:///studentmanage
</property>
<property name="connection.driver_class">
com.mysql.jdbc.Driver</property>
    <property name="connection.username">root</property>
    <property name="connection.password">root</property>
    <!--导入映射配置文件 -->
    <mapping resource=" /user.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

(2) user.hbm.xml（以 user 表为例）：

配置映射文件，用来获取数据库的表名与字段。

配置文件代码如下：

```
<hibernate-mapping>
<class name="com.studentmanage.staff.entity.StaffInfo" table="user">
<id name="ID" type="java.lang.Long">
<column name="ID" length="10"/>
<generator class="native"/>
</id>
<property name="username" type="java.lang.String">
<column name="name" length="30" />
</property>
<property name="password" type="java.lang.String">
<column name="password" length="30"/>
</property>
<property name="grade" type="java.lang.Long">
```

```
<column name="grade" length="10"/>
</property>
</class>
</hibernate-mapping>
```

5.2 系统具体业务实现

5.2.1 登陆页面

首先经由登录页面输入用户名和密码并点击登录按钮后，系统将进行验证操作，判断返回信息并进行相应的页面跳转。若通过验证，则由权限等级的不同而跳转至不同的页面进行后续操作。若没有通过验证，则会弹出错误提示信息。错误提示包括用户名不存在、密码错误、用户重复登录等。页面实现使用 Struts 中的 LoginAction 调用了 CheckUser 函数，Spring 构造了对应的 user 对象并提供 DAO 接口，Hibernate 将对象映射到了数据库的 user 表。登录页面如图 5.2 所示：

图 5.2 登录页面实现

5.2.2 管理员页面

(1) 管理员主页面

管理员登录系统之后，管理员主页面将会显示。该页面作为实现管理员功能的中转页面，页面左侧边框中有“用户管理”与“系统维护”这两个按钮，只要点击相应的按钮，页面右侧就会显示该功能的页面从而进行具体操作。页面实现使用 Struts 中的 LoginAction。管理员主页面如图 5.3 所示：



图 5.3 管理员主页面实现

(2) 用户管理页面

点击“用户管理”按钮，页面右侧会显示出一个表格，表格中的数据是系统所有用户的信息。表格上方可以勾选不显示的项目，便于查询。点击“添加新用户”按钮会弹出设置新用户属性的对话框，设置完毕后，若新用户没有出现与数据库中原有数据重复的情况，则新用户添加成功。点击表格对应行中的按钮则可对用户信息进行修改与删除操作。页面实现使用 Struts 中的 ManagerAction 用以处理用户管理功能，Spring 构造了对应的 user 对象，Hibernate 将对象映射到了数据库的 user 表。用户管理页面如图 5.4 所示：



图 5.4 用户管理页面实现

(3) 系统维护页面

进入系统维护页面后，会显示当前数据备份的状态，点击“设置备份计划”按钮可以更改计划。数据恢复功能会使数据库数据变为最新的备份数据，所以应谨慎使用，在点击确定按钮后需要输入验证码，验证通过系统才会执行恢复功能，以此来防止管理员误操作。页面实现使用 Struts 中的 ManagerAction 调用 DBmanageService，Spring 提供 DAO 接口，Hibernate 执行对应的 MySQL 函数。系

统维护页面如图 5.5 所示：



图 5.5 系统维护页面实现

5.2.3 教师页面

(1) 教师主页面

教师用户登录系统之后，教师主页面将会显示。该页面作为实现教师模块功能的中转页面，页面左侧边框中有“用户管理”、“公告发布”、“课程管理”这三个按钮，只要点击相应的按钮，页面右侧就会显示该功能的页面从而进行具体操作。页面实现如图 5.6 所示：



图 5.6 教师主页面实现

(2) 用户管理页面

点击“用户管理”按钮，页面右侧会显示当前教师用户的基本信息，信息栏右侧有两个按钮，点击“修改用户信息按钮”会跳转至信息修改页面，在该页面中，所有的可修改项目都会变为可编辑的状态，还可以重新上传头像文件。页面具体布局与学生模块相近可见下文学生的信息管理页面部分。“修改密码”按钮

的功能则是修改用户登录密码。页面实现使用 Struts 中的 TeacherAction 调用了 getTeacherInfo 函数, Spring 构造了对应的 teacher 对象并提供 DAO 接口, Hibernate 将对象映射到了数据库的 teacher 表。页面实现如图 5.7 所示:



图 5.7 用户管理页面实现

(3) 课程管理页面

进入课程管理页面后, 会以表格形式显示当前教师用户所教授的所有课程。表格每一行课程名称之后按钮的功能为“课程信息查看”: 显示当前行中课程的具体信息, 如课程编号、课程名称、上课教室等; “学生信息查看”: 查看选择该课程的所有学生信息, 并且可以给该学生的这门课程打分; “作业收取”: 下载学生上传的作业文件; “成绩统计”: 以直方图显示该门课程的成绩分布, 便于教师了解学生的学习情况, 点击“返回”按钮可以跳转回课程管理页面。页面实现使用 Struts 中的 teacherAction 调用了 getLessonInfo、getMarkInfo 等函数, Spring 构造了对应的 lesson 对象与 mark 对象并提供 DAO 接口, Hibernate 将对象映射到了数据库的 lesson 表与 mark 表。页面实现如图 5.8、5.9 所示:



图 5.8 课程管理页面实现

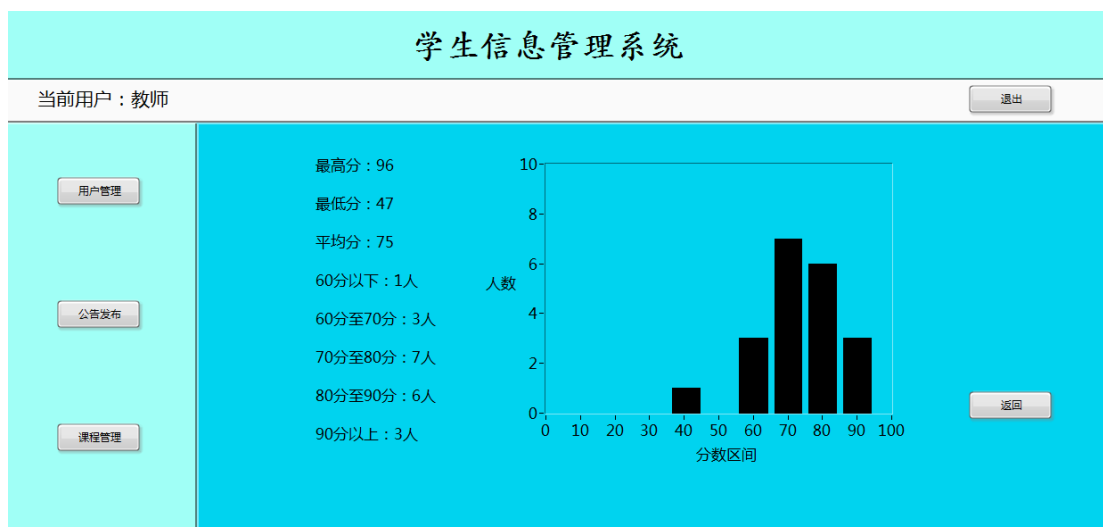


图 5.9 成绩统计页面实现

(4) 公告发布页面

进入公告发布页面后，用户可以在编辑框内输入公告信息，有需要的情况下可以添加附件如表格、文档等。公告编辑完成后，点击“发送”按钮完成公告发布。页面实现使用 Struts 中的 `teacherAction` 调用了 `setNotice`, `addAttachment` 等函数，Spring 构造了对应的 `notice` 对象并提供 DAO 接口，Hibernate 将对象映射到了数据库的 `notice` 表。页面实现如图 5.10 所示：

公告标题:

公告内容:

添加附件: 发布:

图 5.10 公告发布页面实现

5.2.4 学生页面

(1) 学生主页面

学生用户登录系统之后，学生主页面将会显示。该页面作为实现学生模块功能的中转页面，页面左侧边框中有“用户管理”与“课业管理”这两个按钮，只要点击相应的按钮，页面右侧就会显示该功能的页面从而进行具体操作。页面实

现如图 5.11 所示:



图 5.11 学生主页面实现

(2) 用户管理页面

学生用户的用户管理页面与教师用户大体相同, 页面右侧会显示当前学生用户的基本信息。点击“修改用户信息按钮”会跳转至信息修改页面, 在该页面中, 所有的可修改项目都会变为可编辑的状态, 像学号这一类的项目是不可修改的, 点击“浏览”按钮可以浏览本地的图片文件, 可以选择合适的文件作为头像上传至系统。。“返回”按钮的功能是跳转回用户管理页面。页面实现使用 Struts 中的 studentAction 调用了 setStudentInfo、addPicture 等函数, Spring 构造了对应的 student 对象并提供 DAO 接口, Hibernate 将对象映射到了数据库的 student 表。页面实现如图 5.12 所示:

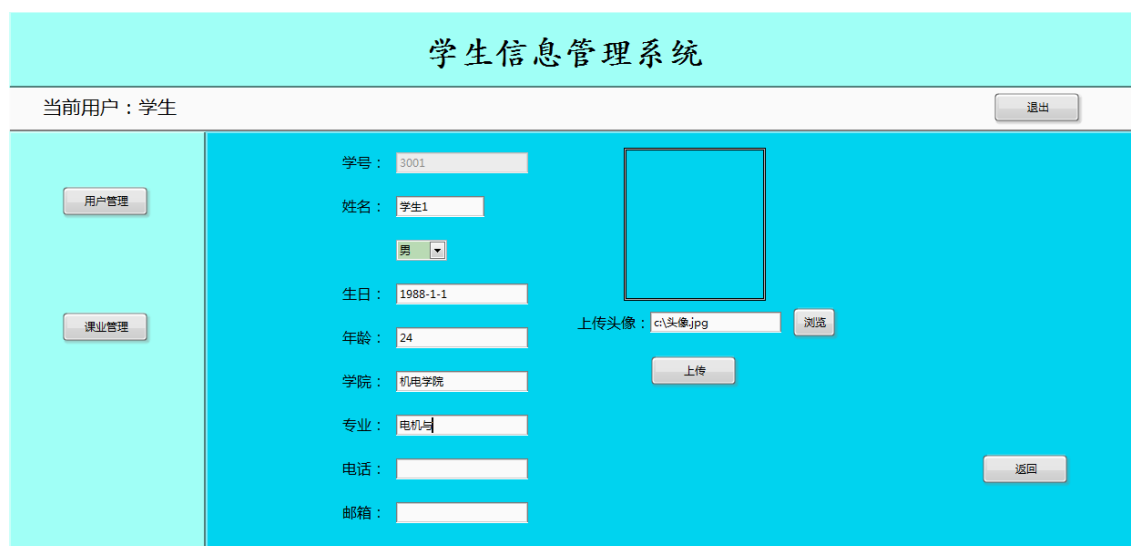


图 5.12 用户管理页面实现

(3) 课业管理页面

进入课业管理页面后, 会以表格形式显示当前学生用户所选修的所有课程。因为学生用户的课程较多, 所以与教师用户不同, 学生用户拥有课程搜索功能, 可以用开设学院及教师姓名这两项来过滤其他不需要显示的课程。其他功能中,

“课程信息查看”与教师用户相同；“任课教师信息查看”可以查看教授该门课程教师的基本信息；“作业提交”可以上传作业文件；“成绩查看”可以查看当前用户该门课程的成绩。页面实现使用 Struts 中的 studentAction 调用了 getLessonInfo、getMark、uploadAssignment 等函数，Spring 构造了对应的 lesson 对象并提供 DAO 接口，Hibernate 将对象映射到了数据库的 lesson 表、assignment 表、mark 表。页面实现如图 5.13 所示：



图 5.13 课业管理页面实现

5.3 安全模块实现

本系统的安全策略主要是 IP 地址访问限制与数据库数据加密传输。如果访问者的 IP 地址是非法的，那么他将无法正常进入系统，浏览器会显示 404 错误页面，如图所示。以“AESKEY”这一字符串作为密钥为例，某一学生的姓名“学生 1”这一信息将会被密钥加密后向网络中传输，如果发生数据泄露，至会得到一长串密文，无法获取有效的数据。如图 5.14、5.15 所示。



图 5.14 错误页面

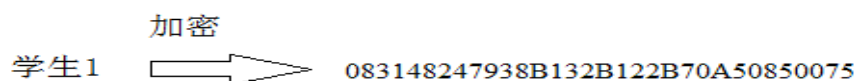


图 5.15 加密算法的实现

第六章 总结

本文的主要工作总结如下：

学生信息管理系统是实现信息化管理的一项具体应用，是为了应对大量学生信息的处理工作而开发的管理软件，一般借助于互联网络用以实现学生信息管理中的各种具体需求，主要功能室对各种学生信息进行常规的管理，如添加、删除、修改、查询等。具有查询快捷、安全可靠、更新方便、存储量大、成本低等特点，而这些特点都是过去的管理方式所无法企及的。

根据学校教学管理模式的具体需要，拟定了系统的设计方案，使所实现的系统具有良好的性能并且具有简化编程、便于用户使用、高智能化的优点。设计了一个整合Struts、Spring与Hibernate的SSH组合框架，实现了信息管理系统。并且使系统具有多种优势，比如：数据安全性好，运行速度快，运行稳定等。

SSH组合框架在系统搭建时，显示了足够强大的优势：第一，Struts框架在实现MVC架构的同时，还加入了 Servlet和 JSP 标记，充分发挥了J2EE的优势，使得构建的MVC架构有了更优异的性能；第二，Hibernate框架能够有效地减少操纵数据库的工作时间，此外还能够使用代理方式精简类的载入过程，不但减少了使用HQL从数据库读取数据的程序编写量，而且节省了开发的成本。另外Hibernate框架还能够与多种服务器集成使用；第三，Spring框架为控制器提供调用的程序接口，并且提供数据持久层的数据访问接口，起到了整个程序框架Java组件的管理作用，还可以进行事务管理关联数据库。

在分析了网络数据传送中的诸多不安全因素的基础上，本文采用对称加密算法，充分发挥了对称加密算法的加密速度快的优势，保证了数据通过公共网络得到安全的传送。

本文所设计的学生信息管理系统虽然能满足一般使用的需要，但是因为欠缺研发经验，而且也缺少开发时间，因此设计中有些方面存在缺陷，接下来要更深入的学习相关的知识，以期更进一步

致谢

衷心感谢我的导师赵明英老师在我攻读硕士学位期间在学习和生活上对我的关怀和帮助。在我读研间，赵老师以身作则，教给了我做科研的基本思路和方法，培养了我分析问题、解决问题的能力。在我做毕业设计的过程中，赵老师时刻关注着课题的进展情况，并且在课题进行的过程中给了我很多前瞻性和富有成效的指导。赵老师勤勉严谨、一丝不苟的治学态度，谦恭热情、坦诚豁达的处事态度，是我一生学习的榜样。同时，赵老师深厚的学术造诣、丰富的实践经验和开阔的科研视野不断熏陶着我，感染着我，并成为我日后从事科学研究的专业准则。

另外我还要感谢 307 实验室的赵启、杨思源、刘振洗、付伟、李春林、薛冲同学，在实验室的日子里，他们给了我很多帮助，我也从他们身上得到了不少启发，学到了很多東西。

特别感谢我的家人，多年来，他们给予了我伟大而无私的爱，一直是我坚强的后盾，激励着我前进。

感谢所有帮助和支持我的人，你们对我的关心和鼓励让我在前进的道路上一步一个脚印，走向人生新的起点。

参考文献

- [1]卿斯汉. 密码学与计算机网络安全. 北京: 清华大学出版社, 2001: 92-150
- [2]罗莉, 温锦生. 网络信息安全技术. 太原科技. 2000, (3): 22-23
- [3]毕晓玲. 网络安全技术的现状和发展. 山西师范大学学报(自然科学版). 2002, 16(2): 24-25
- [4]宋晓雯. 我国的网络信息、安全. 现代图书情报技术, 2002, (1): 53-54
- [5]李晓莉. 计算机网络安全的研究. 郑州大学工学硕士学位论文. 2002: 63-64
- [6]rod johnson. Expert One-on-One: J2EE Design and Developmen. Wrox ,2002
- [7]欧群雍. 基于SOA的教务管理系统的设计与实现: [硕士学位论文]. 南京: 南京理工大学, 2010
- [8]武春红, 王晓明. 基于C/S和B/S模式的教务管理信息系统的设计与实现. 电化教育研究, 2005, (3): 50-52
- [9]陈少云. 基于网络的高职学院教务管理系统的设计与实现: [硕士学位论文]. 成都: 四川大学, 2005
- [10] 冯鹏.基于SSH的教务管理系统的设计与实现[D].电子科技大学 2012
- [11]高在伟. 面向服务的体系结构及其应用研究: [硕士学位论文]. 西安: 西北大学, 2007
- [12]张海藩. 软件工程导论[M]. 北京: 清华大学出版社, 2003: 145-163
- [13]张建党. 基于网络 过程化教学系统设计与实现[D]. 电子科技大学, 2011-09-01
- [14]张小敏. 研究生招生管理信息系统的设计与实现[D]. 山东大学, 2010-04-10
- [15]戚欣. 基于J2EE 架构的层次结构研究及应用[j]. 吉林大学学报, 2008, 62(3): 295-297
- [16]张桂元, 贾燕枫. Eclipse 开发入门与项目实践[M]. 北京: 人民邮电出版社, 2006
- [17]洪明. 欧美国家教育信息化的现状与趋势[J]. 比较教育研究, 2002(7)
- [18]侯建梅. 基于校园网的高校教务信息管理系统的设计与研究[J]. 科技资讯, 2007, (11)
- [19] 史胜辉. Struts+Spring+hibernate框架技术在Web开发中的研究与应用[J]. 电脑知识与技术. 2009(06)
- [20]李兴华, 王月清. Java 网络开发实战经典[M]. 清华大学出版社, 2010: 445-480
- [21]孙卫琴. 精通Struts: 基于MVC的Java 网络设计与开发[M]. 电子工业出版社, 2004: 235-241
- [22]孙卫琴. 精通Hibernate: Java对象持久化技术详解[M]. 电子工业出版社, 2010: 126-138

- [23]RSA Laboratories. PKCS#1: RSA Cryptography Standard, Version 2.1, Mar 1998.
- [24] 徐亚. 基于B/S架构的课堂教学质量评价系统研究[J]. 电脑知识与技术. 2009(24)
- [25]王卫平, 王松涛, 王名茗. 一种基于J2EE、Spring和Hibernate的轻量级EAI 构架. 计算机系统应用. 2005, 11: 38-41
- [26]Wenbo Mao, 王继林等译. 现代密码学理论与实践. 电子工业出版社, 2004: 156-157
- [27] 王健合. 基于SSH框架的员工管理系统[D]. 南昌大学. 2012
- [28]赖溪松, 韩亮, 张真诚等著. 计算机密码学及其应用. 国防工业出版社, 2001: 122-123
- [29]Hibernate: Object/Relational Mapping and Object Persistence for Java. [EB/OL]: 24-27
- [30]杨开英, 刘榭. MVC设计模式在J2EE平台上的研究与实现. 微机发展. 2004, 14(11): 114-116
- [31]赵雪梅 .AES加密算法的实现及应用.常熟理工学院学报 2010
- [32] 师军,张福泰,王耀燕,张永铂. 高级加密标准Rijndael算法中的S盒及其实现[33] 孟庆强, 宗平. J2EE 与三层网络 应用开发. 计算机与现代化, 2004 (3) : 34-38
- [34] Deepak Alur. Core J2EE Paterns-Best Parctices and Design Strategates. China Machine Press, 2002: 89-96
- [35] Nadir Gulzar. Practical J2EE Application Architecture. The McGraw-Hill Companies, Inc., 2003: 36-240
- [36] 于龄涛. J2EE 体系下的返利网系统的设计与实现[D]. 华中科技大学, 2007
- [37] 罗玉玲. J2EE 应用开发详解. 北京: 电子工业出版社, 2009. 89-128
- [38] 蔡剑, 景楠. Java 网络 应用开发: J2EE 和Tomcat. 第二版. 北京: 清华大学出版社, 2005. 236-326
- [39] 蔡明, 陈永运. J2EE 架构的研究与应用. 计算机应用于软件, 2004, 2(6): 41-45
- [40] Vlada Matena, Beth Stearns. Applying Enterprise JavaBeans: Component-BasedDevelopment for the J2EE Platform.北京: 机械工业出版社, 2003: 18-21
- [41] 孙鑫. Struts2 深入详解. 北京: 电子工业出版社, 2008. 279-398
- [42] 张文建, 王建, 范孝良. Struts 在J2EE 网络 层中的应用.微型机与应用, 2003, 8: 17-20

- [43] Budi Kurniawan. Struts 2 Design and Programming: A Tutorial. BRAINY SOFTWARE, 2008: 257-265
- [44] 王肖锋, 曹作良. 基于Struts 体系结构的企业级应用系统. 天津理工学院学报, 2003, 19 (4): 62-66
- [45] 记文柯. Spring 技术内幕-深入解析Spring 架构与设计原理. 北京: 机械工业出版社, 2010: 102-238



西安电子科技大学

地址：西安市太白南路2号

邮编：710071

网址：www.xidian.edu.cn