



**基建项目管理系统的设计与实现**

**学 院 计算机学院**

**专 业 软件工程**

**年级班别 2015级（4）班**

**指导教师 彭重嘉**

**2017年5月**

**摘 要**

高校基建项目在管理上呈现施工技术复杂，工期要求紧，项目参与单位众多，信息沟通复杂，管理事务多而琐碎等特点，在手工管理过程中，出现信息管理难，沟通难，进度监察难等困难，因此需要应用现代化的信息技术辅助工程项目管理。

本文在了解高校基建项目内容，分析基建项目管理系统需求的基础上，设计并实现了基建项目管理系统。该系统包括项目管理、用户管理、权限管理、系统基础配置管理、站内信等模块。项目组成员可通过系统记录相关进度，管理员可通过系统进行基础的配置。除此之外，系统人员之间可发送信息进行沟通交流。

本系统基于J2EE平台，使用MySQL数据库，采用Spring、FreeMarker、Shiro等框架技术进行设计与开发，利用Maven进行项目jar包依赖的管理，最终实现能够满足高校基建科进行项目进度记录及管理的内容管理系统，使项目中的每一环节可查询，提高沟通效率，减少办公开支。

**关键词：**基建项目，项目管理系统， J2EE，MySQL，Spring

**Abstract**

In aspect of management, infrastructure project in universities has the characteristics of complex construction, tight schedule, numerous participant organization, complex information communication and trivial transaction management. In the process of manual management, there are difficulties in information management, communication, and progress monitoring. Therefore, it’s necessary to use modern information technology to assist project management.

Based on the understanding of the content of infrastructure projects and the analysis of the demand of infrastructure project management system, this article had designed and implemented the infrastructure project management system. The system includes the project management, user management, rights management, system basic configuration management, station letter and other modules. Project team members can record the progress through the system, and the administrator can configure all basic information through the system. In addition, the system personnel can send information to communicate.

This system is based on J2EE platform and the MySQL database, using Spring, FreeMarker, Shiro and other framework technology to design and develop. It also utilizes Maven for project jar package dependency management. Finally the infrastructure project management system is implemented that can meet the requirements of recording project progress, improving the communication efficiency and reducing office expenses.

**Key words：**Infrastructure Project，Project Management System，J2EE，MySQL，Spring

**目 录**

[1 绪论 1](#_Toc483495839)

[1.1 选题背景 1](#_Toc483495840)

[1.1.1 高校基建项目的现状 1](#_Toc483495841)

[1.1.2 高校基建项目管理系统的意义 1](#_Toc483495842)

[1.2 系统开发方法 2](#_Toc483495843)

[1.3 系统开发目标 2](#_Toc483495844)

[1.4 本文的主要工作 3](#_Toc483495845)

[1.5 论文的组织结构 3](#_Toc483495846)

[2 相关技术概述 4](#_Toc483495847)

[2.1 Spring 4](#_Toc483495848)

[2.2 FreeMarker 4](#_Toc483495849)

[2.3 Shiro 5](#_Toc483495850)

[2.4 Maven 5](#_Toc483495851)

[3 系统需求分析 6](#_Toc483495852)

[3.1 编写目的 6](#_Toc483495853)

[3.2 背景 6](#_Toc483495854)

[3.3 定义 6](#_Toc483495855)

[3.4 可行性分析 7](#_Toc483495856)

[3.4.1 引言 7](#_Toc483495857)

[3.4.2 可行性研究的前提 7](#_Toc483495858)

[3.4.3 经济可行性 7](#_Toc483495859)

[3.4.4 技术可行性 8](#_Toc483495860)

[3.5 功能需求 8](#_Toc483495861)

[3.6 性能需求 9](#_Toc483495862)

[3.6.1 数据精确度 9](#_Toc483495863)

[3.6.2 时间特性 9](#_Toc483495864)

[3.7 参与者 10](#_Toc483495865)

[3.8 用例建模 10](#_Toc483495866)

[3.9 用例描述 13](#_Toc483495867)

[4 系统设计 20](#_Toc483495868)

[4.1 引言 20](#_Toc483495869)

[4.2 架构设计 20](#_Toc483495870)

[4.3 界面设计 21](#_Toc483495871)

[4.3.1 系统登录界面 21](#_Toc483495872)

[4.3.2 系统主窗体 22](#_Toc483495873)

[4.3.3 数据列表界面 22](#_Toc483495874)

[4.3.4 项目录入页面 23](#_Toc483495875)

[4.3.5 系统主页页面 23](#_Toc483495876)

[4.4 功能模块设计 24](#_Toc483495877)

[4.4.1 引言 24](#_Toc483495878)

[4.4.2 系统功能模块图 24](#_Toc483495879)

[4.4.3 项目管理模块 25](#_Toc483495880)

[4.4.4 文件管理 27](#_Toc483495881)

[4.4.5 基础配置管理模块 28](#_Toc483495882)

[4.4.6 权限管理模块 31](#_Toc483495883)

[4.4.7 用户管理模块 32](#_Toc483495884)

[4.4.8 站内信模块 33](#_Toc483495885)

[4.5 数据库设计 34](#_Toc483495886)

[4.5.1 引言 34](#_Toc483495887)

[4.5.2 概念结构设计 34](#_Toc483495888)

[4.5.3 逻辑结构设计 38](#_Toc483495889)

[4.5.4 物理结构设计 39](#_Toc483495890)

[4.5.5 完整性约束说明 44](#_Toc483495891)

[4.5.6 索引设计 45](#_Toc483495892)

[4.5.7 安全性设计 46](#_Toc483495893)

[5 系统实现 47](#_Toc483495894)

[5.1 数据库实现 47](#_Toc483495895)

[5.2 界面实现 51](#_Toc483495896)

[5.2.1 ACE框架 51](#_Toc483495897)

[5.2.2 系统登录界面 51](#_Toc483495898)

[5.2.3 系统主页 51](#_Toc483495899)

[5.2.4 录入项目界面 52](#_Toc483495900)

[5.2.5 项目列表界面 53](#_Toc483495901)

[5.2.6 发送信件界面 53](#_Toc483495902)

[5.2.7 上传附件界面 54](#_Toc483495903)

[5.3 功能模块实现 54](#_Toc483495904)

[5.3.1 登录 54](#_Toc483495905)

[5.3.2 录入项目基本信息 55](#_Toc483495906)

[5.3.3 修改项目基本信息 57](#_Toc483495907)

[5.3.4 删除项目 58](#_Toc483495908)

[5.3.5 获取项目列表 58](#_Toc483495909)

[5.3.6 权限控制 60](#_Toc483495910)

[6 总结 62](#_Toc483495911)

[参考文献 63](#_Toc483495912)

[致谢 64](#_Toc483495913)

# 绪论

## 选题背景

### 高校基建项目的现状

如今社会变化十分迅速，而高校顺应社会、顺应教育事业的发展，也在不断地优化自身的办学条件，包括教学楼、实验楼以及创新创业基地等各种基建项目不断建设发展，校园内的许多宿舍楼也在翻修。由此可见高校中的基建项目繁杂，可能出现多个项目同时进行的情况，工作人员需要对多个项目同时进行进度把控。除此之外，基建项目涉及的流程繁多，一般来说一个基建项目完整的需经过立项、设计、预算工程量清单、招标、施工监管、结算六大步骤，而每个步骤包含若干小步骤。

总的来说，高校基建项目在管理上呈现施工技术复杂，工期要求紧，项目参与单位众多，信息沟通复杂，管理事务多而琐碎等特点[1]。如果采用手工进行项目进度的管理，容易出现信息管理难，沟通难，进度监察难等情形，这大大降低了基建科的工作人员的工作效率，大大加大了工作的难度和压力。

因此，目前高校基建项目的管理上需要进一步优化，应用现代化的信息技术辅助工程项目管理。

### 高校基建项目管理系统的意义

高校基建项目管理存在重重困难，任务多、步骤繁杂、沟通不便等等，虽然在某些高校中出现了专业的基建项目管理软件，市场中也有需要各式各样的项目管理软件，而国内高校目前还是很少建立较系统的基建项目管理系统[2]。

作为具有一定特殊性的高校基建科，需要根据高校自身的基建现状以及工作经验积累，量身定做一款具有贴合自身需求的基建项目管理软件。实现这样一个基建项目管理系统，能大大减少手工管理时出现的困难，使得项目的进度信息更新更快捷，相关人员的信息交流更密切，最终将大大提高工作效率以及管理的水平。将项目基本信息以及各流程中的备注信息进行电子化存储，也能更好地使信息传递下去，也更方便地进行公开透明，各个工作人员彼此之间形成更好的监督。

由此可见，高校创建自有的基建项目管理系统具体重大意义，有利于高校基建项目的积极发展及项目管理上的高效协作。

## 系统开发方法

目前软件开发中常见的有两种架构，一种是C/S架构，另一种则是B/S架构。

C/S架构全称为Client/Server，即客户端/服务器结构，这种架构是一种典型的两层架构，客户端接受用户的请求后向数据库服务提出请求，数据库服务将相应的数据返回给到客户端，客户端将数据进行计算（可能涉及到运算、汇总、统计等等）并将最终结果展现给用户。

B/S架构全称为Browser/Server，即浏览器/服务器结构，这种架构呈现三层的结构，客户端接受前端页面用户的请求后向应用服务器提交请求，应用服务器接收到相应请求后从数据库服务中获得所需数据，应用服务器将数据进行计算并将结果提交给客户端，客户端将最终结果展现给用户。

由于B/S架构具有无需安装、易维护、成本低、功能易扩展等优势，所以该系统选择采用B/S架构。

对于Web开发应用平台而言，目前常见的包括ASP.NET、J2EE以及LAMP。三种平台的各部分组合为：ASP.NET:Windows Server+IIS+SQL Server+ASP；J2EE:UNIX+Tomcat+Oracle+JSP；LAMP:Linux+Apache+MySQL+PHP。J2EE平台相比其他两个平台而言，开发具有一定难度，开发速度较慢，但是该平台扩展性较好，且各平台的兼容性较强，具有“编写一次，随处运行”的特性。且J2EE已具有众多成熟优秀的框架技术，数据封装较成熟，能大大提高开发效率。

综上，根据本项目的特点，决定选用J2EE平台进行开发一个B/S架构的基建项目管理系统。

## 系统开发目标

根据项目特点与开发要求，系统必须达到以下目标：

* 1. 用户体验良好，界面简洁明了，操作简单便捷；
  2. 功能完整，运行稳定，信息安全；
  3. 可进行角色配置，对前端页面菜单权限进行灵活控制；
  4. 项目信息存储可靠，项目流程明确，有一定的统计功能。

## 本文的主要工作

本文针对本高校中涉及的基建项目流程、管理上的需求以及存在的问题，结合现有普通项目管理软件的经验，设计与实现了一个适用于本校的基于B/S架构及J2EE平台的基建项目管理系统。主要研究的内容包括以下几个方面：

1. 初步了解：了解国内外基建项目管理系统的研究现状，以及本校的基建项目管理需求和难题。
2. 需求分析：根据高校基建项目管理的需求和开发目标，利用用例建模的方法，明确系统的目的、范围、定义和功能。
3. 系统设计：依据系统的需求分析结果，对系统进行总体设计及详细设计，包括功能模块设计、数据库逻辑设计、前端页面设计等。
4. 编码实现：依据上一步设计的方案，进行系统的具体编码开发工作。

## 论文的组织结构

本文分为七章。

第一章：主要介绍高校基建项目背景，本系统开发方法与开发目标，以及本文的主要工作。

第二章：简单介绍本系统开发过程中所使用到的相关框架技术。

第三章：对本系统进行可行性分析，包括经济可行性、技术可行性等方面。

第四章：依据高校基建背景调查结果和市面已有系统，对本系统进行需求分析。

第五章：依据需求分析结果，对系统的架构、界面、功能模块、数据库模型进行设计。

第六章：依据概要设计和详细设计输出的内容，对系统的功能模块、界面、数据库操作做了详细的实现。

第七章：对本系统的开发及本文工作进行总结。

# 相关技术概述

## Spring

Spring Framework是一个得到了Apache License 2.0许可的开源的企业级应用程序框架，为构建满足企业级需求的应用程序提供了大量的工具集。推出该框架的原因是在使用J2EE进行开发时会提高复杂性。使用EJB(Enterprise Java Bean)创建和部署Bean是沉重的负担，原因是为了创建一个Bean，除了Bean定义之外，还需要创建主程序和组件接口。在当时，由于在EJB世界中没有依赖注入的概念，因此查找方法是找到对象或资源的唯一方法。Spring的1.0版本引入了轻量级应用程序框架以及相关功能，比如JDBC抽象、对象关系映射工具支持、事务管理、调度和邮寄抽象实现以及MVC Web框架[4]。

Spring最引人注目的两个特质就是依赖注入（Dependency Injection）和面向切面编程（Aspect-Oriented Programming）。依赖注入也被称为控制反转，通过Spring将某个对象所依赖的对象的引用进行传递或注入，而不是由使用者主动去创建该依赖。而AOP将一些影响多个类的公共行为封装为一个重用的模块，即所谓的“切面”，以切面的方式控制后台各个处理环节，能以优雅的方式实现日志记录、拦截器等操作。

Spring本身也在持续地演化和提升，它一直致力于将困难的开发任务进行简化，不断地为Java开发人员带来创新性的特性。在Spring最初所挑战的领域，Spring已经突飞猛进，涉及的范围扩展到Java应用开发的各个方面[5]。

## FreeMarker

FreeMarker是一款免费的模板引擎：基于模板和要改变的数据生成文本输出（如HTML网页、电子邮件、配置文件、源代码等）的Java库。模板是使用FreeMarker模板语言（FTL）编写的，这是一种简单的专业语言（不是像PHP这样的全面的编程语言）[6]。

FreeMarker主要由文本、注释、插值和FTL指令四个部分组成，我们可以在表达式中对插值进行特定的运算、判定[7]。除了表达式语言的优点外，FreeMarker相比于常见的JSP而言，在复杂页面上的性能具有少许优势。而且FreeMarker中不能编写Java代码，能很好地实现严格的MVC分离。FreeMarker中也内置了大量的常用功能，例如日期、金额格式化等，大大方便了页面的展示。

## Shiro

Shiro是Java的一个安全框架，类似的框架有Spring Security。Shiro能容易地开发出满足我们权限需求的应用，可以帮助我们完成诸如认证、授权、加密、会话管理、web集成、缓存等功能。而与Spring Security相比，虽然功能上没有那么强大，但对于普通的应用而言，Shiro已经能很好地满足我们的需要，小而简单。

## Maven

Maven这个词可以翻译为“知识的积累”，也可以翻译为“专家”或“内行”。而我们讨论的Maven其实是一个跨平台的项目管理工具。作为Apache组织中的一个颇为成功的开源项目，Maven主要服务于基于Java平台的项目构建、依赖管理和项目信息管理。无论是小型的开源类库项目，还是大型的企业级应用；无论是传统的瀑布式开发，还是流行的敏捷模式，Maven都能大显身手[8]。

而对于本人而言，最关注的莫过于Maven的依赖管理。Maven强大的依赖管理功能，解决了一大堆Jar包的引入问题，不再杂而乱。此外，Maven也有效地防止和解决了依赖冲突、无效依赖等问题。利用好Maven，能减少许多莫名其妙的错误，避免一些不必要的时间消耗。

# 系统需求分析

## 编写目的

需求的编写是为了研究基建项目管理系统的开发途径和应用方法。同时它也是进行项目策划、概要设计和详细设计的基础。是维护人员进行内部维护，信息更新，验收和测试的依据。本需求的预期读者是高校基建项目管理系统软件有关联的决策人，开发组成员，软件验证者。

根据需求调研分析报告，定义系统功能和系统数据流图，通过编写需求分析规格说明书，让开发人员能够根据需求规格说明书进行项目开发。

## 背景

待开发的软件系统名称叫《基建项目管理系统》。本软件适用于本校基建处，可以提供包括项目信息录入、项目流程跟踪、项目文件上传、用户站内沟通等多项服务。基建处人员手工管理基建项目任务繁重、工期紧张、效率较低，为了提高对基建项目的管理效率，更好地保证项目的正常进展，因此开发此系统。

随着计算机技术的不断提高，计算机已深入社会生活的每个角落。若继续采用手工管理的方法，不仅效率低下、流程繁琐、易出错，也耗费了大量人力。为了满足基建处工作人员对基建项目进行高效管理，特开发此基建项目管理系统以提高学校相关部门的管理效率。

## 定义

需求分析是从客户的需求中提取出软件系统能够帮助用户解决问题，通过对用户问题的分析，确定系统的功能需求。这个步骤是对理解需求的升华，直接关系到该系统的质量。分析的根本目的是在开发者和提出需求的人之间建立一种理解和沟通机制，因此，系统的需求分析也应该是开发人员和用户一起完成的。

## 可行性分析

### 引言

应学校要求，为方便学校基建科对校内基建项目进行电子化管理，提高基建项目管理的效率及可靠性，拟为其开发基建项目管理系统。经过对项目进行详细的调查研究，得到项目可行性研究报告，对项目开发中将要面临的问题及其解决方案进行可行性分析，包括项目在经济、技术、操作和校内影响方面的可行性。

### 可行性研究的前提

1.功能要求

该系统以项目管理工作流为核心，实现项目多用户协同工作和移动办公。通过系统，让每个项目的全过程工作流程的细节得以记录，可查询项目管理中每一流程。相关负责人在系统中填写相应的阶段任务工期计划。

此外，系统管理员需通过系统进行基本的用户管理、权限管理等功能操作。

2.性能要求

该系统使用范围是基建科的相关工作人员和和基建项目相关的部门审计，财务，校办等，用户群体数量较小，对于系统的并发压力不大。但是在正常请求量下，需要快速对用户的输入信息及请求进行处理，且需要支撑J2EE平台的正常运行，包括应用服务器及数据库的正常运行。

### 经济可行性

本系统主要为本校基建科服务，根据基建科的相关需求进行开发。由于基建科的经费有限，所以应注重低投入高收益。系统对于机器的硬件配置要求不高，中等性能的服务器就可以支撑起系统的正常运行。而开发人员的费用支出较小，经济负担不大。完成开发这个系统能给基建科的相关人员带来很大的方便，一方面能减少人力方面的费用支出，另一方面能大大提高基建项目管理的计划性和预见性。在项目完成运行后，经济收益将会在较短的几年时间超过投资。

### 技术可行性

该系统对于硬件配置的要求不高，同时并发量不大，用户量基数较小，不需要采用分布式架构。若基于J2EE平台，采用Spring框架、Shiro框架等热门、成熟的技术框架进行Web开发，单台机器已经能支撑相应的请求压力。而开源数据库MySQL以及开源应用服务器Tomcat也已经是十分成熟的组件，在Java的Web开发中属于十分常见的组合。而开发人员对于J2EE开发较为熟悉，对框架技术有初步的使用经验，能在短时间能进行项目框架的搭建。所以，在软硬件方面，在较低的性能要求下，可以进行基建项目管理系统的开发。

## 功能需求

在基建项目管理系统中，要为每个基建处工作人员及其他相关部门人员建立一个账户，该账户中存储用户的个人信息、项目任务信息及站内通讯信息。通过该系统，可录入项目信息，选择相应流程，指定各阶段负责人，录入计划工期。管理员可对系统的基础配置进行修改，对用户及其权限进行管理。系统内用户互相之间可进行通信。

根据上述需求，提出以下功能性需求：

1. 项目管理：系统需要对项目进行管理，包括项目的录入（基本信息、流程信息、阶段信息等），对项目要进行展示且可条件查询。此外，用户还能将项目相关的文件进行上传，指定所归属的项目，其他用户可查询系统中的文件列表并下载查阅。阶段负责人或项目负责人登陆后能看到自己所负责的任务列表，完成一个流程后可提交流程备注进入下一流程，当前流程结束。
2. 用户管理：系统需要进行用户账号的创建，每个账号能查询自己的个人信息，进行密码的修改。管理员能新增系统用户，能看到系统中的所有用户并进行条件查询。还能删除系统中的某一用户。
3. 权限管理：系统中的功能菜单都需要进行权限的分配，每个角色拥有的权限不同，需进行动态的调整。管理员可以查看角色列表、功能权限列表，对角色进行增加、删除、修改信息、修改所拥有权限等操作。
4. 基础配置管理：基建项目中存在许多基础信息需要配置，包括项目每一阶段所包含的流程信息，项目的类别配置信息，以及用户的职称信息配置。管理员需对这些配置信息进行管理，进行增加、删除、修改等操作。
5. 站内信：为了解决沟通难的问题，提高协作效率，用户可对系统中的用户进行一对一发送消息。每个用户可以看到自己的所收到的信息列表以及已发送的列表，查看相应的信件内容。

根据候选需求，提出以下非功能性需求：

1. 功能必须尽可能完善、全面。
2. 系统稳定性、安全性必须有保证。
3. 系统可移植性强，可在不同平台下正常运行。
4. 系统界面美观，操作简单快捷。

## 性能需求

### 数据精确度

在精度需求上，根据使用需求，在各项数据的输入、输出以及传输过程中，可以满足各种精度的需求。如：根据关键字精度的不同，可分为精确查找和泛型查找。精确查找可以精确匹配到相应状态、相应用户的项目信息；泛型查找可以查询到满足与输入关键字相匹配的项目或用户即可输出，可进行项目名查询、用户名查询。查询时应保证查全率，所有相应域包含查询关键字的记录都应该能查到，查到的记录应与给定的单项或组合查询条件完全匹配。

### 时间特性

在软件方面，响应时间、更新处理时间都应比较迅速，完全满足用户要求，具体要求如下：

①响应时间：3~5秒；

②更新处理时间：4~6秒；

③数据转换与传输时间：4~5秒；

④系统启动时间5~10秒

## 参与者

参与者（actor）在建模过程中是处于核心地位的。UML官方文档对参与者的定义为：actor是在系统之外与系统交互的某人或某事物。参与者一定是直接并且主动地向系统发出动作并获得反馈的，否则就不是参与者[9]。

根据需求分析，系统的参与者有：系统管理员、配置管理员、基建处工作人员、项目工作人员。

各参与者的活动如表3.1。

表3.1 参与者活动表

|  |  |
| --- | --- |
| 参与者 | 活动 |
| 系统管理员 | 管理用户、角色，设定权限；项目及项目文件的查询、删除操作；进行系统配置的添加、修改、删除操作；发送站内信息。 |
| 配置管理员 | 对项目流程配置、项目类别配置、用户职称配置进行添加、修改、删除操作；发送站内信息。 |
| 基建处工作人员 | 项目信息录入；项目流程进度更新；项目及项目文件的查询操作；发送站内信息。 |
| 项目工作人员 | 项目流程进度更新；项目及项目文件的查询操作；发送站内信息。 |

## 用例建模

用例建模（Use Case Modeling）是使用用例的方法来描述系统的功能需求的过程，用例模型主要包括用例图和用例描述。根据前文的分析，结合UML的知识点，给出本系统的用例图，用例描述请看本章“用例描述”部分。

系统层用例图描述了系统提供的全部服务。图4.1显示的是本系统的系统层用例，包括五个子用例：项目管理、用户管理、权限管理、基础配置管理、站内信。

为了进一步分析系统，对系统用例进行分解。其中用户管理可以分为查询用户、添加用户、修改用户信息、删除用户、分配角色。权限管理可以分为查询角色、添加角色、修改角色、删除角色、分配功能权限、查询功能权限、修改功能权限名称。基础配置管理可以分为添加项目流程配置、修改项目流程配置、删除项目流程配置、查询项目流程配置、添加项目类型配置、修改项目类型配置、删除项目类型配置、查询项目类型配置、添加用户职称配置、修改用户职称配置、删除用户职称配置、查询用户职称配置。项目管理可以分为添加项目、查询项目、修改项目基本信息、修改项目进度、删除项目。站内信可以分为发送消息、查询接收列表、查询已发送列表。

各个用例图如图3.2、图3.3、图3.4、图3.5、图3.6所示。

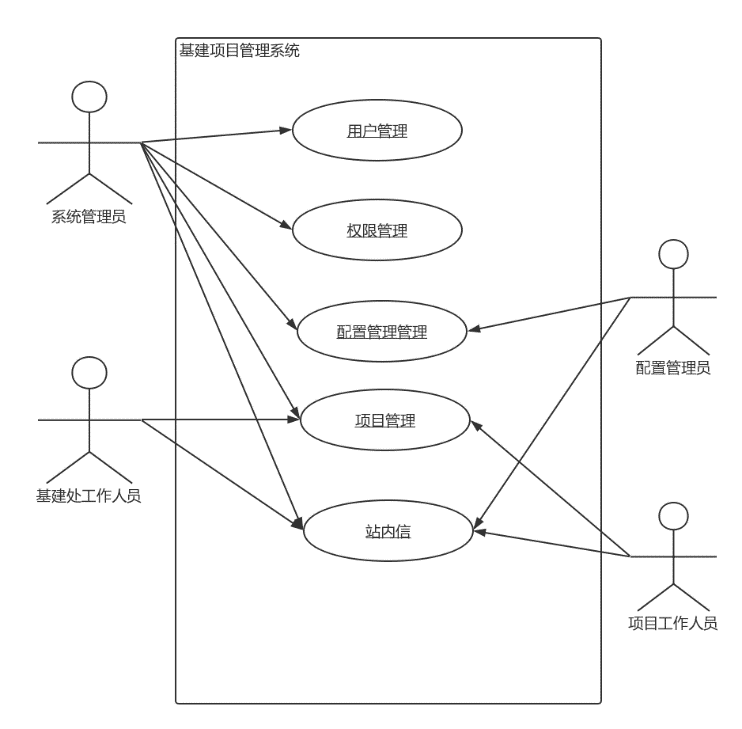


图3.1 系统用例图

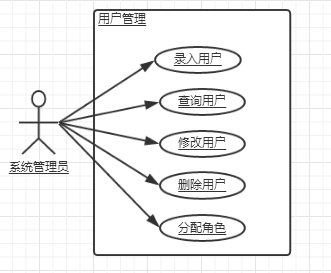


图3.2 用户管理用例图

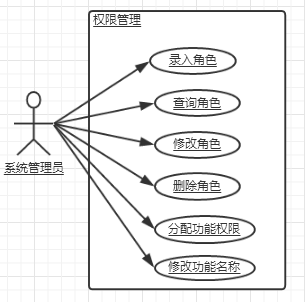


图3.3 权限管理用例图

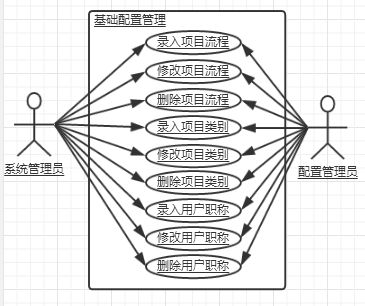


图3.4 基础配置管理用例图

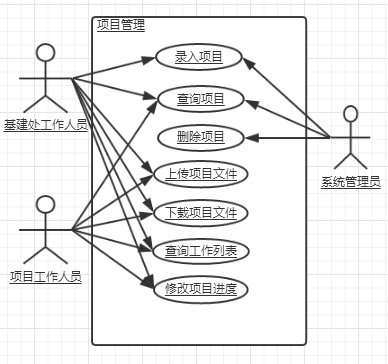


图3.5 项目管理用例图

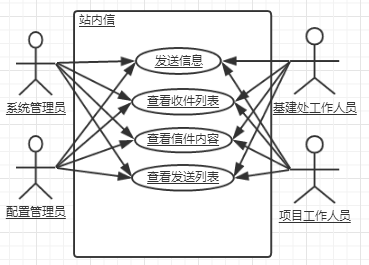


图3.6 站内信用例图

## 用例描述

用例描述一般包括用例名称、用例说明、参与者、前置条件、后置条件、基本事件流、其他事件流、异常事件流。为了更清晰地表示用例，对本系统的主要用例进行详细描述。

1. 录入项目

表3.2 录入项目用例描述

|  |  |
| --- | --- |
| 用例名称 | 添加项目 |
| 参与者 | 系统管理员，基建处工作人员 |
| 用例说明 | 参与者在系统选择添加项目 |
| 前置条件 | 参与者成功登陆系统 |
| 基本事件流 | 1. 参与者用鼠标点击“项目管理”下的“添加项目”菜单项 2. 系统跳转进入“添加项目”页面 3. 参与者填写项目基本信息，点击“提交”，提示成功后点击“下一步” 4. 系统显示流程选择表单，参与者选择本项目应进行的流程，点击“提交”，提示成功后点击“下一步” 5. 系统显示阶段信息表单，参与者选择本项目各阶段负责人及工期，点击“提交” 6. 点击“完成”按钮，完成添加项目的操作 7. 用例终止 |
| 其他事件流 | 三个表单在点击“提交”按钮前，参与者可以随时点击“重置”按钮或点击其他菜单项，表单中的填写的内容不会影响系统 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 系统项目列表新增一条记录，相应负责人的工作列表新增一条或多条记录 |

1. 修改项目基本信息

表3.3 修改项目基本信息用例描述

|  |  |
| --- | --- |
| 用例名称 | 修改项目基本信息 |
| 参与者 | 系统管理员，基建处工作人员 |
| 用例说明 | 参与者对某一项目的基本信息进行修改 |
| 前置条件 | 参与者已登录且项目已存在 |
| 基本事件流 | 1. 参与者点击菜单“项目列表” 2. 鼠标点击列表右侧“查看”按钮 3. 系统进入“项目详情”页面，页面显示项目的基本信息及流程信息 4. 系统管理员或基建处工作人员点击“修改项目基本信息”按钮 5. 参与者填写修改信息的表单 6. 点击“提交”按钮，完成修改项目基本信息的操作 7. 用例终止 |
| 其他事件流 | 在点击“提交”按钮前，参与者可以随时点击“重置”按钮或点击其他菜单，表单中的修改的内容不会生效 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 数据库中该项目信息产生了相应的变化，前端页面显示与修改一致 |

1. 删除项目

表3.4 删除项目用例描述

|  |  |
| --- | --- |
| 用例名称 | 删除项目 |
| 参与者 | 系统管理员 |
| 用例说明 | 参与者在项目列表中选择删除某一项目，前端页面不再展示该项目 |
| 前置条件 | 参与者成功登陆系统且该项目信息已存在 |
| 基本事件流 | 1. 点击“项目列表”菜单进入项目列表页面 2. 参与者用鼠标选择某项目点击“删除”按钮 3. 系统提示“确定要删除吗” 4. 参与者点击“确定”按钮，完成删除操作 5. 用例终止 |
| 其他事件流 | 在点击“确定”按钮前，参与者可以随时点击“取消”按钮，取消本次删除操作 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 数据库中项目状态更改为已删除，前端页面不再展现该项目信息 |

1. 修改项目进度

表3.5 修改项目进度用例描述

|  |  |
| --- | --- |
| 用例名称 | 修改项目进度 |
| 参与者 | 项目工作人员 |
| 用例说明 | 参与者完成项目流程后对该项目的进度进行更新 |
| 前置条件 | 参与者已登录且项目处于该参与者的所负责阶段 |
| 基本事件流 | 1. 参与者点击菜单“我的工作”，进入工作列表页面 2. 鼠标点击列表右侧“查看”按钮 3. 系统进入“工作详情”页面，页面显示项目的基本信息及阶段信息 4. 参与者点击“结束该流程”按钮 5. 参与者填写流程备注信息表单 6. 点击“提交”按钮，完成修改项目进度的操作 7. 用例终止 |
| 其他事件流 | 在点击“提交”按钮前，参与者可以随时点击“重置”按钮或点击其他菜单，表单中的修改的内容不会生效 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 数据库中该项目状态、工作状态及流程状态产生了相应的变化，前端页面显示与修改一致 |

1. 添加角色

表3.6 添加内容用例描述

|  |  |
| --- | --- |
| 用例名称 | 添加角色 |
| 参与者 | 系统管理员 |
| 用例说明 | 参与者新增一个角色，并对角色分配权限 |
| 前置条件 | 参与者成功登陆系统 |
| 基本事件流 | 1. 参与者点击“角色列表”菜单，进入角色列表页面 2. 参与者用鼠标点击“添加角色”按钮 3. 系统弹出添加角色悬浮框 4. 参与者填写表单信息，勾选分配的功能权限 5. 参与者点击“提交”按钮，完成添加角色操作 6. 用例终止 |
| 其他事件流 | 在点击“提交”按钮前，参与者可以随时点击“重置”按钮或点击其他菜单项，表单填写的内容不会生效 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 角色列表出现该角色，且信息与表单填写的一致 |

1. 修改角色

表3.7 修改角色用例描述

|  |  |
| --- | --- |
| 用例名称 | 修改角色 |
| 参与者 | 系统管理员 |
| 用例说明 | 参与者在系统修改角色信息，数据库中数据相应地修改 |
| 前置条件 | 参与者成功登陆系统 |
| 基本事件流 | 1. 参与者点击“角色列表”菜单，进入角色列表页面 2. 参与者用鼠标点击某一角色的“修改”按钮 3. 系统弹出修改角色悬浮窗，表单显示角色的原信息 4. 参与者修改表单中的角色信息 5. 参与者点击“提交”按钮，完成修改角色操作 6. 用例终止 |
| 其他事件流 | 在点击“提交”按钮前，参与者可以随时点击“重置”按钮或其他菜单项，表单修改的内容不会影响系统 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 数据库中角色信息相应地修改，前端页面角色列表展示信息与修改内容一致 |

1. 删除角色

表3.8 删除角色用例描述

|  |  |
| --- | --- |
| 用例名称 | 删除角色 |
| 参与者 | 系统管理员 |
| 用例说明 | 参与者进入系统删除角色，数据库中相应的角色信息被删除 |
| 前置条件 | 参与者成功登陆系统且角色已存在 |
| 基本事件流 | 1. 参与者点击“角色列表”菜单项 2. 系统进入角色列表页面 3. 参与者用鼠标点击某一角色栏右侧“删除”按钮 4. 系统提示“确定要删除吗” 5. 参与者点击“确定”按钮，完成删除操作 6. 用例终止 |
| 其他事件流 | 在点击“确定”按钮前，参与者可以随时点击“取消”按钮，取消本次删除操作 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 数据库中相应角色信息被删除，角色列表中不再出现该条角色信息 |

1. 添加项目流程配置

表3.9 添加项目流程配置用例描述

|  |  |
| --- | --- |
| 用例名称 | 添加项目流程配置 |
| 参与者 | 系统管理员或配置管理员 |
| 用例说明 | 参与者新增项目可选的流程信息 |
| 前置条件 | 参与者成功登陆系统 |
| 基本事件流 | 1. 参与者点击“项目流程配置”菜单 2. 系统进入项目流程配置信息列表页面 3. 参与者用鼠标点击“添加流程”按钮 4. 系统弹出添加流程悬浮框 5. 参与者填写表单信息 6. 参与者点击“提交”按钮，完成添加流程配置信息操作 7. 用例终止 |
| 其他事件流 | 在点击“提交”按钮前，参与者可以随时点击“重置”按钮或点击其他菜单项，表单填写的内容不会生效 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 流程配置列表出现该信息，且流程信息与表单填写的一致 |

1. 修改流程配置信息

表3.10 修改流程配置信息用例描述

|  |  |
| --- | --- |
| 用例名称 | 修改评论 |
| 参与者 | 系统管理员或配置管理员 |
| 用例说明 | 参与者在系统修改流程配置信息，数据库中数据相应地修改 |
| 前置条件 | 参与者成功登陆系统且该流程配置已存在 |
| 基本事件流 | 1. 参与者点击“系统流程配置”菜单，进入流程配置列表页面 2. 参与者用鼠标点击某一流程配置记录右侧的“修改”按钮 3. 系统弹出修改流程配置信息悬浮窗，表单显示流程配置的原信息 4. 参与者修改表单中的流程配置信息 5. 参与者点击“提交”按钮，完成修改流程配置信息操作 6. 用例终止 |
| 其他事件流 | 在点击“确定”按钮前，参与者可以随时点击“重置”按钮或其他菜单项，取消本次修改操作 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 数据库中相应的流程配置信息修改，前端页面流程配置列表展示的信息与修改内容一致 |

1. 发送信息

表3.11 发送信息用例描述

|  |  |
| --- | --- |
| 用例名称 | 发送信息 |
| 参与者 | 系统管理员或配置管理员或基建处工作人员或项目工作人员 |
| 用例说明 | 参与者对系统其他用户发送站内消息 |
| 前置条件 | 参与者成功登陆系统且接收用户已存在 |
| 基本事件流 | 1. 参与者点击“发送消息”菜单 2. 系统进入发送消息页面 3. 参与者填写表单中的相关消息内容 4. 参与者点击“发送”按钮，完成站内信息发送操作 5. 用例终止 |
| 其他事件流 | 在点击“发送”按钮前，参与者可以随时点击“重置”按钮或其他菜单项，取消本次发送操作 |
| 异常事件流 | 提示错误信息，参与者确认 |
| 后置条件 | 参与者可在发送列表中查看该条发送记录，接收方可收到该条消息 |

# 系统设计

## 引言

经过需求分析阶段的工作，已经得出了该系统的基本需求，要对系统进行实现，必须先分析如何实现，系统设计阶段就是对整个系统的开发实现进行设计上的描述。

本系统基于J2EE平台，且采用了基于MVC模式的B/S架构，进行代码组织实现。在系统设计阶段包括了架构设计、功能模块设计、数据库设计以及界面设计四部分。

## 架构设计

B/S结构（Browser/Server，浏览器/服务器模式），是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。[10]

对于Java而言，已经具有了成熟的技术平台进行基于B/S架构的软件实现，也就是J2EE平台。采用B/S架构，能更方便相关工作人员的日常使用，不需要安装，只需要计算机中安装了浏览器就可以登录系统进行工作。对于开发人员而言，也更便于维护和升级。

此外，在B/S架构的基础上，采用了MVC模式的代码组织。MVC即Model View Controller，是模型-视图-控制器的缩写。MVC模式的思想是将业务逻辑、数据、页面展示进行分离的方式进行代码组织，分为表示层、业务逻辑层、数据服务层进行代码的编写。采用MVC模式，能大大降低维护的工作量，在改进页面及用户交互时不需要重新编写业务逻辑。

从用户量及请求并发量上看，该系统对于性能上的压力并不大。用户量较小，且并发量也比较小，所以不需要采取负载均衡、分布式等高并发的处理方式进行架构的设计。从目前的需求来看，一台服务器就足以应对暂时的系统压力。

所以，本系统采取了单机的B/S三层体系进行开发，具体的架构设计如下图：

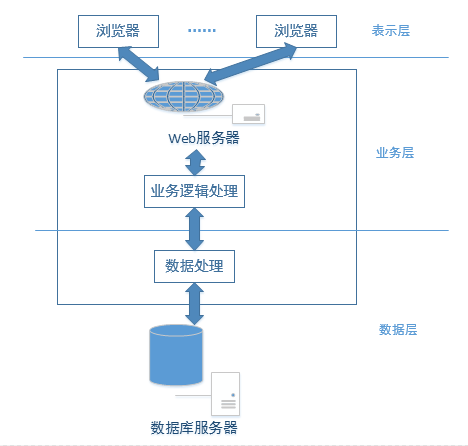


图5.1 系统架构图

## 界面设计

### 系统登录界面

设计草图如图5.2：



图5.2 系统登录界面

### 系统主窗体

设计草图如图5.3：



图5.3 系统主窗体

### 数据列表界面

设计草图如图5.4：



图5.4 数据列表界面

### 项目录入页面

设计草图如图5.5：



图5.5项目录入页面

### 系统主页页面

设计草图如图5.6：

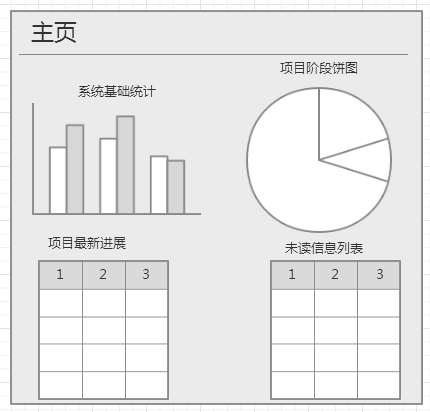


图5.6 系统主页页面

## 功能模块设计

### 引言

系统设计的主要内容之一是系统功能模块结构设计，该部分的任务是合理地对系统进行模块的分解和定义，将每个子系统划分成一个个小的模块，并利用模块结构图将分解后的模块按层次结构联系起来。

在功能模块设计的时候应注意一些原则，包括功能结构的合理性、完备性，系统各功能的独立性。在进行功能模块划分时，要以系统论的设计思想为指导，合理地进行集成和区分，需要了解清楚各个功能特点，理清逻辑。各个功能模块之间应相互独立，完成单一、独立且完整的功能，避免出现重复的功能，一个功能模块内应尽量做到“高内聚”，不同模块间应做到“低耦合”。

### 系统功能模块图

本系统的主要功能有：项目管理、文件管理、基础配置管理、权限管理、用户管理、站内信，各功能下又细分为不同的小功能模块，系统的功能结构图如图5.7所示。

图5.7 系统功能模块图

### 项目管理模块

项目管理模块的功能分为六部分，其中“录入项目”分为三部分，“工作管理”分为三部分，如图5.8所示：

图5.8 项目管理功能模块图

1. 录入基本信息

输入项：项目名称、项目编号、项目描述、项目负责人、项目所属公司、项目开始时间、项目结束时间。

输出项：系统提示操作结果是否成功。

说明：点击“录入项目”，进入添加项目页面，输入所有输入项后，点击“提交”按钮，进行添加项目基本信息操作，系统提示操作的结果。

1. 录入流程信息

输入项：项目id、各阶段所需流程列表。

输出项：系统提示操作结果是否成功。

说明：项目基本信息录入完成后，点击“下一步”，进入流程信息选择页面，输入所有输入项后，点击“提交”按钮，进行添加项目流程信息操作，系统提示操作的结果。

1. 录入阶段信息

输入项：项目id、各阶段开始日期、各阶段结束日期、各阶段负责人。

输出项：系统提示操作结果是否成功。

说明：项目流程信息录入完成后，点击“下一步”，进入阶段信息填写页面，输入所有输入项后，点击“提交”按钮，进行添加项目阶段信息操作，系统提示操作的结果。

1. 查看项目列表

输入项：项目名称、项目编号、项目负责人名称。

输出项：系统提示操作结果是否成功。

说明：选择左侧菜单栏中的“项目列表”，即进入项目列表页面，在此页面可输入搜索条件，点击“查询”，把数据提交到服务器，系统采用分页的方式展现展示相应数据列表， 每页15条记录。

1. 查看项目详情

输入项：项目id 。

输出项：系统提示操作结果是否成功。

说明：选择左侧菜单栏中的“项目列表”，即进入项目列表页面，点击相应项目的“查看”选项，主内容显示相应的项目详情。

1. 查看工作列表

输入项：项目名称、项目编号、阶段名称。

输出项：系统提示操作结果是否成功。

说明：选择左侧菜单栏中的“我的任务”，即进入任务列表页面，在此页面可输入搜索条件，点击“查询”，把数据提交到服务器，系统采用分页的方式展现展示相应数据列表， 每页15条记录。

1. 查看工作详情

输入项：工作id 。

输出项：系统提示操作结果是否成功。

说明：选择左侧菜单栏中的“我的工作”，即进入工作列表页面，点击相应项目的“查看”选项，主内容显示相应的工作详情。

1. 修改项目基本信息

输入项：项目id，项目名称、项目描述、项目负责人、项目所属公司、项目开始时间、项目结束时间。

输出项：系统提示操作结果是否成功。

说明：选择左侧菜单栏中的“项目列表”，点击相应项目的“查看”选项，即进入项目详情页面，修改相应内容后点击“提交”按钮，把数据提交到服务器，系统提示操作的结果。

1. 更新项目进度

输入项：流程id、项目id。

输出项：系统提示操作结果是否成功。

说明：选择左侧菜单栏中的“我的工作”，点击相应工作的“查看”选项，即进入项目阶段工作页面，点击“完成该流程”按钮，输入相应内容后点击“提交”按钮，把数据提交到服务器，系统提示操作的结果。

1. 删除项目

输入项：项目id。

输出项：系统提示操作结果是否成功。

说明：选择左侧栏的“项目列表”菜单项，主内容显示区即显示系统中的项目列表，点击相应项目的“删除”选项，执行“删除”操作。系统提示操作的结果。

### 文件管理

文件管理模块的功能分为四部分，如图5.9所示：

图5.9 文件管理功能模块图

1. 上传文件

输入项：文件名、文件所属项目、待上传文件。

输出项：系统提示操作结果是否成功。

说明：点击“文件列表”菜单项，进入文件列表页面，点击“上传文件”按钮，在表单中输入相应的信息，点击“提交”按钮，执行上传文件操作。系统提示操作结果。

1. 下载文件

输入项：文件编号。

输出项：系统提示操作结果是否成功。

说明：点击“文件列表”菜单项，进入文件列表列表页面，点击记录项右侧“下载”按钮，执行下载文件操作。

1. 删除文件

输入项：文件编号。

输出项：系统提示操作结果是否成功。

说明：在文件列表中点击相应记录项右侧“删除”按钮，弹出确认框点击“确定”按钮，执行删除文件操作。系统提示操作结果。

1. 查看文件列表

输入项：无。

输出项：显示文件列表。

说明：选择左侧“文件列表”菜单项，主内容显示区即显示文件列表，列表进行分页处理，每页显示15条记录。在该页面可进行条件搜索，查询满足条件的列表。

### 基础配置管理模块

基础配置管理模块的功能分为三部分，每部分分为四个小模块，如图5.10所示：

图5.10 基础配置管理功能模块图

1. 添加项目流程配置

输入项：所属阶段、阶段名称、流程名称、流程序号、流程描述。

输出项：系统提示操作结果是否成功。

说明：点击“项目流程配置”菜单项，进入项目流程配置列表页面，点击“添加流程配置”按钮，在表单中输入相应的信息，点击“提交”按钮，执行添加项目流程配置操作。系统提示操作结果。

1. 修改项目流程配置

输入项：所属阶段、阶段名称、流程名称、流程序号、流程描述。

输出项：系统提示操作结果是否成功。

说明：点击“项目流程配置”菜单项，进入项目流程配置列表页面，点击记录项右侧“修改”按钮，在表单中输入相应的信息，点击“提交”按钮，执行修改项目流程配置操作。系统提示操作结果。

1. 删除项目流程配置

输入项：项目流程配置id。

输出项：系统提示操作结果是否成功。

说明：在项目流程配置列表中点击相应记录项右侧“删除”按钮，弹出确认框点击“确定”按钮，执行删除项目流程配置操作。系统提示操作结果。

1. 查看项目流程配置列表

输入项：所属阶段、流程名称。

输出项：显示项目流程配置列表。

说明：选择左侧“项目流程配置”菜单项，主内容显示区即显示属于项目流程配置列表，列表进行分页处理，每页显示15条记录。在该页面可进行条件搜索，查询满足条件的列表。

1. 添加项目类型配置

输入项：类型名称、类型描述。

输出项：系统提示操作结果是否成功。

说明：点击“项目类型配置”菜单项，进入项目类型配置列表页面，点击“添加类型配置”按钮，在表单中输入相应的信息，点击“提交”按钮，执行添加项目类型配置操作。系统提示操作结果。

1. 修改项目类型配置

输入项：类型名称、类型描述。

输出项：系统提示操作结果是否成功。

说明：点击“项目类型配置”菜单项，进入项目类型配置列表页面，点击记录项右侧“修改”按钮，在表单中输入相应的信息，点击“提交”按钮，执行修改项目类型配置操作。系统提示操作结果。

1. 删除项目类型配置

输入项：项目类型配置id。

输出项：系统提示操作结果是否成功。

说明：在项目类型配置列表中点击相应类型记录项右侧“删除”按钮，弹出确认框点击“确定”按钮，执行删除项目类型配置操作。系统提示操作结果。

1. 查看项目类型配置列表

输入项：无。

输出项：显示项目类型配置列表。

说明：选择左侧“项目类型配置”菜单项，主内容显示区即显示属于项目类型配置列表，列表进行分页处理，每页显示15条记录。

1. 添加用户职称配置

输入项：职称名称、职称描述。

输出项：系统提示操作结果是否成功。

说明：点击“用户职称配置”菜单项，进入用户职称配置列表页面，点击“添加用户职称”按钮，在表单中输入相应的信息，点击“提交”按钮，执行添加用户职称配置操作。系统提示操作结果。

1. 修改用户职称配置

输入项：职称名称、职称描述。

输出项：系统提示操作结果是否成功。

说明：点击“用户职称配置”菜单项，进入用户职称配置列表页面，点击记录项右侧“修改”按钮，在表单中输入相应的信息，点击“提交”按钮，执行修改用户职称配置操作。系统提示操作结果。

1. 删除用户职称配置

输入项：用户职称配置id。

输出项：系统提示操作结果是否成功。

说明：在用户职称列表中点击相应记录项右侧“删除”按钮，弹出确认框点击“确定”按钮，执行删除用户职称配置操作。系统提示操作结果。

1. 查看用户职称配置列表

输入项：无。

输出项：显示用户职称配置列表。

说明：选择左侧“用户职称配置”菜单项，主内容显示区即显示属于用户职称配置列表，列表进行分页处理，每页显示15条记录。

### 权限管理模块

评论管理模块的功能分为四个部分，如图5.11所示：

图5.11 权限管理功能模块图

1. 查看角色列表

输入项：无

输出项：显示角色列表

说明：点击左侧“角色列表”菜单项，主内容显示区即显示所有角色列表。列表进行分页处理，每页显示15条记录。

1. 添加角色

输入项：角色名称、角色code、权限。

输出项：系统提示操作结果是否成功。

说明：点击左侧“角色列表”菜单项，主内容显示区即显示所有角色列表，点击上方“添加角色”按钮，输入相应的信息点击“提交”，系统提示操作结果。

1. 修改角色

输入项：角色id、角色名称、角色code。

输出项：系统提示操作结果是否成功。

说明：在角色列表中选中相应角色记录的“修改”按钮，输入相应信息后点击“提交”，执行修改角色操作。系统提示操作结果。

1. 删除角色

输入项：角色id。

输出项：系统提示操作结果是否成功。

说明：在角色列表中选中相应角色记录的“删除”选项，执行删除角色操作。系统提示操作结果。

### 用户管理模块

用户管理模块的功能分为四个部分，如图5.12所示：

图5.12 用户管理功能模块图

1. 查看用户列表

输入项：查询关键字。

输出项：显示用户列表。

说明：点击左侧“用户列表”菜单项，主内容显示区显示所有用户列表。列表进行分页处理，每页显示15条记录。

1. 添加用户

输入项：用户名、密码、姓名、性别、手机、邮箱、职称、角色。

输出项：系统提示操作结果是否成功。

说明：点击左侧“添加用户”菜单项，进入添加用户页面。输入用户名、密码、姓名、性别、手机、邮箱、职称、角色，点击“提交”按钮，执行添加用户操作，系统提示操作结果。

1. 删除用户

输入项：用户id。

输出项：系统提示操作结果是否成功。

说明：在用户列表页面选中相应用户条目的“删除”按钮，执行删除操作。系统提示操作结果。

1. 修改用户

输入项：用户名、密码、姓名、性别、手机、邮箱、职称、角色。

输出项：系统提示操作结果是否成功。

说明：在用户列表页面选中相应用户条目的“修改”按钮，弹出修改用户悬浮框。修改用户名、密码、姓名、性别、手机、邮箱、角色，点击“提交”按钮，执行修改用户操作，系统提示操作结果。

### 站内信模块

站内信模块的功能分为五个部分，如图5.13所示：

图5.13 站内信功能模块图

1. 查看收件列表

输入项：查询关键字。

输出项：显示已收到的信息列表。

说明：点击左侧“收件箱”菜单项，主内容显示区显示所有该用户已收到的信息列表。

1. 查看发件列表

输入项：查询关键字。

输出项：显示已发送的信息列表。

说明：点击左侧“已发送”菜单项，主内容显示区显示所有该用户已发送的信息列表。

1. 发送信息

输入项：收件人账号、信息标题、信息内容。

输出项：系统提示操作结果是否成功。

说明：点击系统左侧“发送信息”按钮，进入发送信息页面。输入收件人账号、信息标题、信息内容，点击“发送”按钮，执行发送信息操作，系统提示操作结果。

1. 查看信息内容

输入项：信息id。

输出项：显示信息内容。

说明：在收件列表页面或发件列表页面选中相应信息条目的“查看”按钮，主内容显示区显示该信息的内容。

1. 删除信息

输入项：信息id。

输出项：系统提示操作结果是否成功。

说明：在收件列表页面或发件列表页面选中相应信息条目的“删除”按钮，点击“确定”按钮，执行删除操作。系统提示操作结果。

## 数据库设计

### 引言

数据库设计是指对一个给定的应用环境，构造（设计）优化的数据库逻辑模式和物理结构，并据此建立数据库及其应用系统，使之能够有效地存储和管理数据，满足各种用户的应用需求，包括信息管理要求和数据操作要求。[11]

数据库设计的合理性对一个系统的影响是举足轻重的，不仅影响到当前系统的性能情况，也会影响到系统后期的需求变更。所以在进行数据库设计时，需要遵循一些原则，包括标准化和规范化、合理使用索引、数据完整性。避免出现数据的冗余，提高系统的查询性能，防止不正确的数据进入数据库。

### 概念结构设计

概念结构设计是整个数据库设计的关键，通过对用户需求进行综合、归纳与抽象，形成一个独立于具体DBMS的概念模型[12]。通常概念设计使用E-R图来进行描述。

由需求分析可以确定本系统必须包含以下实体：项目实体、项目流程实体、项目阶段任务实体、附件实体、站内信实体、用户实体、角色实体、功能实体。各个实体的详细信息如图5.14至5.21所示，实体间的关系如图5.22所示。

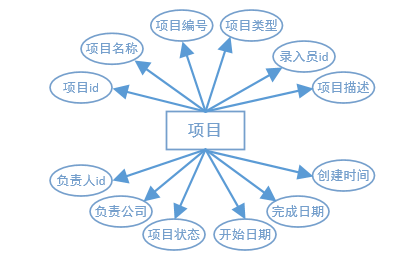


图5.14 项目实体

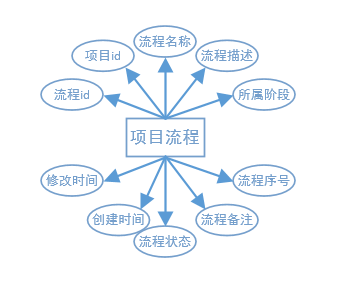


图5.15 项目流程实体

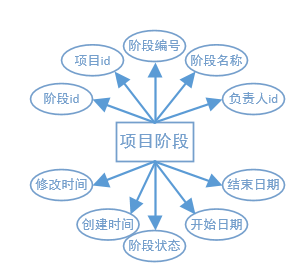


图5.16 项目阶段任务实体

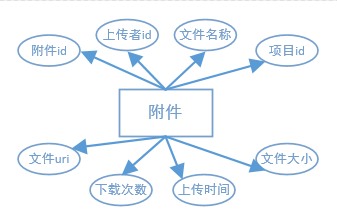


图5.17 附件实体

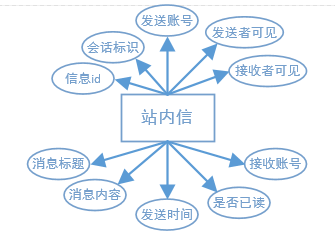


图5.18 站内信实体

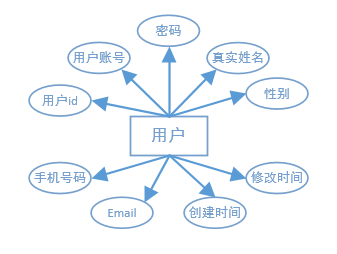


图5.19 用户实体

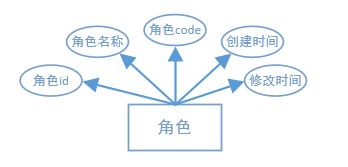


图5.20 角色实体

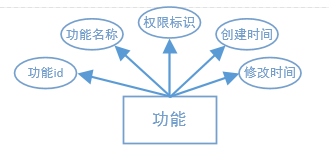


图5.21 功能实体

分析系统各实体之间的关系，可得出系统的实体-关系图，如图5.23所示。

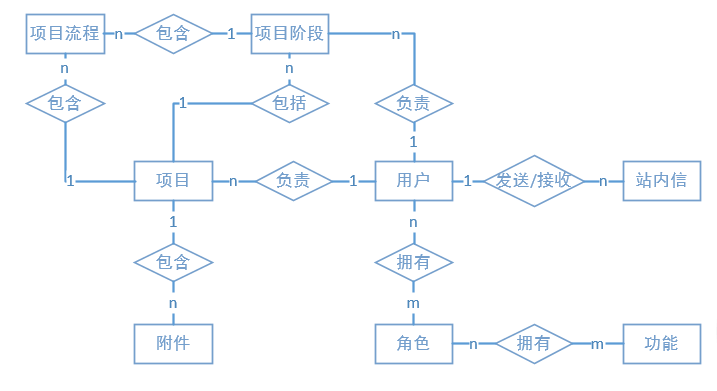


图5.22 数据库E-R图

### 逻辑结构设计

根据上节的数据库概念设计中各个实体的E-R图可以转化为相应的关系模型，具体如下：

**项目基本信息**（自动编号，项目名称，项目编号，类型名称，录入员自动编号，项目描述，负责人自动编号，所属公司名称，项目状态，项目开始日期，项目结束日期）：主码为自动编号；录入员自动编号为对应用户信息的自动编号的外码，项目负责人自动编号为对应用户信息的自动编号的外码；自动编码、项目名称、项目编号、项目状态非空。

**项目流程信息**（自动编号，项目自动编号，项目流程名称，流程描述，所属阶段，流程序号，流程完成时备注，流程状态，创建时间，修改时间）：主码为自动编号；项目自动编号为对应项目信息表的自动编号的外码；自动编号、项目自动编号、项目流程名称、所属阶段、流程序号、流程状态非空。

**项目阶段任务信息**（自动编号，项目自动编号，所属阶段，阶段名称，阶段负责人自动编号，阶段开始日期，阶段结束日期，阶段任务状态，创建时间，结束时间）：主码为自动编号；项目自动编号为对应项目信息表的自动编号的外码，阶段负责人自动编号为对应用户信息的自动编号的外码；自动编号、项目自动编号、所属阶段、阶段名称、阶段负责人自动编号、阶段任务状态非空。

**附件信息**（自动编号，上传者自动编号，附件名称，附件大小，项目自动编号，上传时间，附件地址，下载次数，创建时间，修改时间）：主码为自动编号；上传者自动编号为对应用户信息的自动编号的外码，项目自动编号为对应项目信息的自动编号的外码；自动编号、上传者自动编号、附件名称、附件大小、项目自动编号、上传时间、附件地址均为非空。

**站内信**（自动编号，会话编号，发送者账号，发送者是否可见，信息标题，信息内容，发送时间，接收者账号，接收者是否可见，是否已读，创建时间，修改时间）：主码为自动编号；发送者账号是对应用户信息的用户账号的外码，接收者账号是对应用户信息的用户账号的外码；自动编号、发送者账号、发送者是否可见、信息标题、信息内容、发送时间、接收者账号、接收者是否可见、是否已读非空。

**用户信息表**（自动编号，账号，密码，真实姓名，性别，手机号码，邮箱，职称）：主码为自动编号；自动编号、账号、密码、真实姓名非空。

**角色信息表**（自动编号，角色名称，角色编码）：主码为自动编号；自动编号、角色名称、角色编码非空。

**角色分配**（用户自动编号，角色自动编号）：主码为用户自动编号和角色自动编号；用户自动编号为对应用户信息的自动编号的外码，角色自动编号为对应角色信息的自动编号的外码；用户自动编号和角色自动编号均非空。

**功能权限**（自动编号，功能名称，权限标识）：主码为自动编号；自动编号、功能名称和权限标识均非空。

**权限分配**（角色自动编号，功能自动编号）：主码为角色自动编号和功能自动编号；角色自动编号为对应角色信息的自动编号的外码，功能自动编号为对应功能权限信息的自动编号的外码；角色自动编号、功能自动编号均非空。

### 物理结构设计

该系统采用MySQL作为RDBMS（关系型数据库管理系统），将上述的逻辑结构实施到具体的MySQL环境中，确定数据库中各个表的实体属性（字段）、数据类型、长度等。

1. 项目基本信息表：project\_info

项目基本信息表具有id、项目名称、项目编号、类型名称、录入员id、项目描述、负责人id、所属公司名称、项目状态、项目开始日期、项目结束日期、创建时间、修改时间等字段，具体的逻辑结构如表5.1所示：

表5.1 项目基本信息表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| id | int | 11 | 否 | 是 | 项目id |
| project\_name | varchar | 36 | 否 | 否 | 项目名称 |
| project\_num | varchar | 20 | 否 | 否 | 项目编号 |
| type\_name | varchar | 20 | 是 | 否 | 项目类型名称 |
| entry\_staff\_account | int | 11 | 是 | 否 | 项目录入员id |
| description | varchar | 256 | 是 | 否 | 项目描述 |
| leader\_id | int | 11 | 是 | 否 | 项目负责人id |
| company | varchar | 50 | 是 | 否 | 项目负责公司 |
| status | int | 11 | 否 | 否 | 项目状态 |
| project\_start\_date | date | 0 | 是 | 否 | 项目开始日期 |
| project\_end\_date | date | 0 | 是 | 否 | 项目完成日期 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 修改时间 |

1. 项目流程表：project\_process

项目流程表具有id、所属项目id、项目流程名称、流程描述、所属阶段、流程序号、流程完成时备注、流程状态、创建时间、修改时间等字段，具体的物理结构如表5.2所示：

表5.2 项目流程表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| id | int | 11 | 否 | 是 | 项目流程id |
| project\_id | int | 11 | 否 | 否 | 所属项目id |
| process\_name | varchar | 20 | 否 | 否 | 项目流程名称 |
| description | varchar | 256 | 是 | 否 | 流程描述 |
| period | int | 11 | 否 | 否 | 所属阶段 |
| serial\_num | int | 11 | 否 | 否 | 流程序号 |
| remark | varchar | 256 | 是 | 否 | 流程完成时备注 |
| status | int | 11 | 否 | 否 | 流程状态 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 修改时间 |

1. 项目阶段任务表：project\_period

评论表具有id、项目id、所属阶段、阶段名称、阶段负责人id、阶段开始日期、阶段结束日期、阶段任务状态、创建时间、结束时间等字段，具体的物理结构如表5.3所示：

表5.3 项目阶段任务表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| id | int | 11 | 否 | 是 | 项目阶段任务id |
| project\_id | int | 11 | 否 | 否 | 所属项目id |
| period | int | 11 | 否 | 否 | 所属阶段序号 |
| period\_name | varchar | 36 | 否 | 否 | 所属阶段名称 |
| period\_leader\_id | int | 11 | 是 | 否 | 阶段任务负责人 |
| period\_start\_date | date | 0 | 是 | 否 | 任务开始日期 |
| period\_end\_date | date | 0 | 是 | 否 | 任务完成日期 |
| status | int | 11 | 否 | 否 | 阶段任务状态 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 修改时间 |

1. 附件表：document\_info

附件表具有id、上传者id、附件名称、附件大小、所属项目id、上传时间、附件uri、下载次数、创建时间、修改时间等字段，具体的物理结构如表5.4所示：

表5.4 附件表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| id | int | 11 | 否 | 是 | 附件id |
| uploader\_id | int | 11 | 否 | 否 | 上传者id |
| document\_name | varchar | 36 | 否 | 否 | 附件名称 |
| document\_size | int | 11 | 是 | 否 | 附件大小 |
| project\_id | int | 11 | 否 | 否 | 附件所属项目 |
| upload\_time | datetime | 0 | 否 | 否 | 上传时间 |
| document\_uri | varchar | 256 | 否 | 否 | 附件uri地址 |
| download\_times | int | 10 | 是 | 否 | 下载次数 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 修改时间 |

1. 站内信表：message\_info

站内信表具有id、会话id、发送者账号、发送者是否可见、信息标题、信息内容、发送时间、接收者账号、接收者是否可见、是否已读、创建时间、修改时间等字段，具体的物理结构如表5.5所示。

表5.5 站内信表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| id | int | 11 | 否 | 是 | 站内信递增id |
| trans\_id | int | 16 | 是 | 否 | 会话id |
| sender\_account | varchar | 16 | 否 | 否 | 发送者账号 |
| sender\_visible | int | 11 | 否 | 否 | 发送者是否可见 |
| message\_title | varchar | 36 | 否 | 否 | 消息标题 |
| content | text | 0 | 否 | 否 | 消息内容 |
| send\_time | datetime | 0 | 否 | 否 | 发送时间 |
| receiver\_account | varchar | 16 | 否 | 否 | 接收者账号 |
| receiver\_visible | int | 11 | 否 | 否 | 接收者是否可见 |
| is\_readed | int | 11 | 否 | 否 | 是否已读 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 修改时间 |

1. 用户信息表：portal\_user

用户信息表具有id、账号、密码、真实姓名、性别、手机号码、邮箱、职称、创建时间、修改时间等字段，具体的物理结构如表5.6所示。

表5.6 用户信息表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| id | int | 11 | 否 | 是 | 用户递增id |
| account | varchar | 16 | 否 | 否 | 用户账号 |
| passwd | varchar | 32 | 否 | 否 | 登陆密码 |
| user\_name | varchar | 16 | 否 | 否 | 真实姓名 |
| sex | int | 1 | 是 | 否 | 性别 |
| phone\_num | varchar | 16 | 是 | 否 | 手机号码 |
| email | varchar | 40 | 是 | 否 | 电子邮箱 |
| user\_title | varchar | 16 | 否 | 否 | 用户职称 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 修改时间 |

1. 角色信息表：portal\_role

角色信息表具有id、角色名称、角色code、创建时间、修改时间等字段，具体的物理结构如表5.7所示。

表5.7 角色信息表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| id | int | 11 | 否 | 是 | 用户递增id |
| role\_name | varchar | 16 | 否 | 否 | 角色名称 |
| role\_code | varchar | 16 | 否 | 否 | 角色code |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 修改时间 |

1. 用户角色关系表：portal\_user\_role

用户角色关系表具有用户id、角色id字段，具体的物理结构如表5.8所示。

表5.8 用户角色关系表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| user\_id | int | 11 | 否 | 是 | 用户id |
| role\_id | int | 11 | 否 | 是 | 角色id |

1. 功能权限信息表：portal\_function

功能权限信息表具有id、功能名称、功能标识、创建时间、修改时间等字段，具体的物理结构如表5.9所示。

表5.9 功能权限信息表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| id | int | 11 | 否 | 是 | 功能权限递增id |
| function\_name | varchar | 16 | 否 | 是 | 功能名称 |
| permission | varchar | 64 | 否 | 是 | 权限标识 |
| create\_time | datetime | 0 | 是 | 否 | 创建时间 |
| update\_time | timestamp | 0 | 是 | 否 | 修改时间 |

1. 角色权限关系表：portal\_role\_function

角色权限关系表具有功能权限id、角色id字段，具体的物理结构如表5.8所示。

表5.10 角色权限关系表物理结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 是否为空 | 是否主键 | 描述 |
| function\_id | int | 11 | 否 | 是 | 功能权限id |
| role\_id | int | 11 | 否 | 是 | 角色id |

### 完整性约束说明

主要是对数据库表的三种完整性进行设计。

1、实体完整性

通过设置主键都已实现。

2、参照完整性

①project\_info（项目信息表）的entry\_staff\_account和leader\_id为外键，分别对应portal\_user表的account和id；

②project\_process（项目流程表）的project\_id为外键，对应project\_info表的id，；

③project\_period（项目阶段任务表）的item\_id及claimer\_id为外键，分别对应item\_info表的id及user\_info表的id，；

④document\_info表的uploader\_id和project\_id为外键，分别对应portal\_user的id和project\_info的id；

⑤message\_info表的sender\_account、receiver\_account为外键，对应portal\_user表中的id；

⑥portal\_user\_role表的user\_id和role\_id为外键，分别对应portal\_user表的id和portal\_role表的id；

⑦portal\_role\_function表的role\_id和function\_id为外键，分别对应portal\_role表的id和portal\_function表的id。

【注：外键关联均通过业务逻辑进行实现，而未在数据库层面进行设置，是出于效率考虑，一般不在数据库中设置外键。】

3、用户自定义完整性

因为MySQL并不支持check进行完整性定义，所以只能将自定义完整性在程序中进行设置，在前端以及后台进行双重判断，排除不正确的数据进入数据库[13]。

### 索引设计

在数据库中，索引就是表中数据和相应存储位置的列表。使用索引可以大大减少数据的查询时间。但是创建索引时需要遵循一定的原则，大量创建索引不是合理的做法，这样会占用大量的存储空间，且对索引的动态维护需要一定的成本。我们应该选取经常搜索的列、主关键字以及经常用于连接的列上建立索引。

该系统的数据库索引采用BTree索引，各表中的索引如下：

1、项目信息表（project\_info表）：

以id作为单值索引。

2、项目流程表（project\_process表）：

以id作为单值索引；以project\_id+period作为组合索引。

3、项目阶段任务表（project\_period表）：

以id作为单值索引；以project\_id+leader\_id作为组合索引。

4、附件信息表（document\_info表）：

以id、project\_id、uploader\_id分别作为单值索引。

5、站内信信息表（message\_info表）：

以receiver\_account、sender\_account分别作为单值索引。

6、用户信息表（portal\_user表）：

以id、account分别作为单值索引。

### 安全性设计

数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。

该数据库主要通过用户标识和鉴别以及存取控制保证数据库的安全。首先为该数据库创建两个用户并分配相应的权限，一个是管理员用户，另一个是基建处工作人员用户。管理员用户连接数据库后，拥有所有语句权限、对象权限，可进行数据库的增删改查以及创建用户。而工作人员用户连接数据库后，限制其insert、update、delete等操作的权限，只具有select语句权限，且不具有创建数据库对象的权限。通过不同的权限，限制不同用户的操作，避免出现用户的误操作或恶意篡改，保证数据不会被破坏。

# 系统实现

系统主要采用了MyEclipse、tomcat、MySQL进行开发。前端技术上采用了ACE后台管理框架，且选择freemarker进行视图的渲染。后台技术上选用了Spring、huisadb、Shiro等技术框架，实现了MVC模式开发，能对用户进行权限的控制。数据库上选择了MySQL数据库，通过索引对系统数据的查询进行优化。

## 数据库实现

根据数据库的设计，本节给出系统主要数据表的sql脚本实现。

1. 项目基本信息表

CREATE TABLE `project\_info` (

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '自增id',

`project\_name` varchar(36) DEFAULT NULL COMMENT '项目名称',

`project\_num` varchar(20) DEFAULT NULL COMMENT '项目编号',

`type\_name` varchar(20) DEFAULT NULL COMMENT '项目类别',

`entry\_staff\_account` varchar(11) DEFAULT NULL COMMENT '项目录入员账户',

`description` varchar(256) DEFAULT NULL COMMENT '项目描述',

`leader\_id` int(11) DEFAULT NULL COMMENT '项目负责人id',

`company` varchar(50) DEFAULT NULL COMMENT '项目负责公司名称',

`status` int(11) DEFAULT NULL COMMENT '项目状态【1-7，其中1-6未相关阶段，0为已删除，7为已完成】',

`project\_start\_date` date DEFAULT NULL COMMENT '项目起始日期',

`project\_end\_date` date DEFAULT NULL COMMENT '项目完成日期',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',

PRIMARY KEY (`id`),

KEY `project\_id` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=53 DEFAULT CHARSET=utf8;

1. 项目流程表

CREATE TABLE `project\_process` (

`id` int(11) unsigned NOT NULL AUTO\_INCREMENT COMMENT '自增id',

`project\_id` int(20) DEFAULT NULL COMMENT '项目id',

`process\_name` varchar(20) DEFAULT NULL COMMENT '流程名称',

`description` varchar(256) DEFAULT NULL COMMENT '流程描述',

`period` int(11) DEFAULT NULL COMMENT '流程所属阶段【1-6阶段】',

`serial\_num` int(11) DEFAULT NULL COMMENT '流程序号',

`remark` varchar(256) DEFAULT NULL COMMENT '项目流程完成时备注',

`status` int(11) DEFAULT NULL COMMENT '流程状态【1：已完成；0：未完成】',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',

PRIMARY KEY (`id`),

KEY `project\_id` (`project\_id`),

KEY `project\_id\_period` (`project\_id`,`period`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=94 DEFAULT CHARSET=utf8;

1. 项目阶段任务表

CREATE TABLE `project\_period` (

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '递增id',

`project\_id` int(11) DEFAULT NULL COMMENT '此阶段所属项目id',

`period` int(11) DEFAULT NULL COMMENT '所属阶段【1-6】',

`period\_name` varchar(255) DEFAULT NULL COMMENT '阶段名称',

`period\_leader\_id` int(11) DEFAULT NULL COMMENT '阶段负责人id',

`period\_start\_date` date DEFAULT NULL COMMENT '阶段开始日期',

`period\_end\_date` date DEFAULT NULL COMMENT '阶段完成日期',

`status` int(11) DEFAULT NULL COMMENT '阶段状态【0：未完成；1：已完成】',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',

PRIMARY KEY (`id`),

KEY `id` (`id`) USING BTREE,

KEY `project+leader` (`project\_id`,`period\_leader\_id`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=16 DEFAULT CHARSET=utf8;

1. 附件信息表

CREATE TABLE `document\_info` (

`id` int(11) unsigned NOT NULL AUTO\_INCREMENT COMMENT '自动编号id',

`uploader\_id` int(11) DEFAULT NULL COMMENT '上传者id',

`document\_name` varchar(250) DEFAULT NULL COMMENT '文件名称',

`project\_id` int(11) DEFAULT NULL COMMENT '文件所属项目id',

`document\_size` int(20) DEFAULT NULL COMMENT '文件大小',

`upload\_time` datetime DEFAULT '0000-00-00 00:00:00' COMMENT '上传时间',

`document\_uri` varchar(250) DEFAULT NULL COMMENT '文件uri',

`download\_times` int(10) DEFAULT '0' COMMENT '下载次数',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',

PRIMARY KEY (`id`),

KEY `document\_id` (`id`),

KEY `project` (`project\_id`) USING BTREE,

KEY `uploader` (`uploader\_id`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8;

1. 站内信表

CREATE TABLE `message\_info` (

`id` int(11) unsigned NOT NULL AUTO\_INCREMENT,

`trans\_id` varchar(16) DEFAULT NULL COMMENT '会话id ',

`sender\_account` varchar(11) DEFAULT NULL COMMENT '发送用户账号',

`sender\_visible` int(11) DEFAULT NULL COMMENT '发送者是否可见',

`message\_title` varchar(20) DEFAULT NULL COMMENT '信件标题',

`content` text COMMENT '信件内容',

`send\_time` datetime DEFAULT NULL COMMENT '发送信件时间',

`reply\_message\_id` int(11) DEFAULT NULL COMMENT '回复信件的id',

`receiver\_account` varchar(11) DEFAULT NULL COMMENT '信件回复者账号',

`receiver\_visible` int(11) DEFAULT NULL COMMENT '接收者是否可见',

`is\_readed` int(11) DEFAULT '0' COMMENT '是否已阅读',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',

PRIMARY KEY (`id`),

KEY `receive\_account` (`receiver\_account`) USING BTREE,

KEY `send\_account` (`sender\_account`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

1. 用户信息表

CREATE TABLE `portal\_user` (

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT 'id，自增',

`account` varchar(16) NOT NULL COMMENT '账号',

`passwd` varchar(32) NOT NULL COMMENT '密码，MD5加密',

`user\_name` varchar(16) NOT NULL COMMENT '姓名',

`sex` int(1) DEFAULT NULL COMMENT '性别；值域【1:男，2：女】',

`phone\_num` varchar(16) DEFAULT NULL COMMENT '手机号码',

`email` varchar(40) DEFAULT NULL COMMENT '邮箱',

`user\_portrait` varchar(120) DEFAULT NULL COMMENT '头像uri,

`last\_login\_time` datetime DEFAULT NULL COMMENT '最后一次登录时间',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',

PRIMARY KEY (`id`),

UNIQUE KEY `portal\_user\_account` (`account`) USING BTREE,

KEY `id` (`id`) USING BTREE

) ENGINE=MyISAM AUTO\_INCREMENT=29 DEFAULT CHARSET=utf8;

7、角色信息表

CREATE TABLE `portal\_role` (

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT 'id，自增',

`role\_name` varchar(16) NOT NULL COMMENT '角色名称',

`role\_code` varchar(16) NOT NULL COMMENT '角色code',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,PRIMARY KEY (`id`)

) ENGINE=MyISAM AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

8、功能信息表

CREATE TABLE `portal\_function` (

`id` int(11) NOT NULL AUTO\_INCREMENT COMMENT 'id，自增',

`function\_name` varchar(16) NOT NULL COMMENT '功能名称',

`permission` varchar(200) DEFAULT NULL COMMENT '权限标识',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',

PRIMARY KEY (`id`)

) ENGINE=MyISAM AUTO\_INCREMENT=21 DEFAULT CHARSET=utf8;

## 界面实现

### ACE框架

该基建项目管理系统界面实现主要采用ACE框架，该框架是一个基于Bootstrap的后台管理系统模板框架。Bootstrap是Twitter在几年前开发出的前端框架，该框架不仅界面美观，且兼容了许多浏览器，能大大加快了页面的开发速度。基于Bootstrap的ACE框架，界面也同样简洁美观，还整合了许多实用的插件和组件，能满足大多数管理后台页面的开发。

### 系统登录界面

根据设计草图，登录界面的实现结果如图6.1所示：



图5.1系统登录界面

### 系统主页

系统主页主要展示了项目的基本统计信息，包括项目总数信息、各阶段项目的统计饼图，以及项目最新动态和未读列表。

项目各阶段统计的饼图使用了bootstrap的插件，异步发送请求到后台获取数据后，JS获取数据后填入饼图的数据对象，调用相应方法进行饼图的渲染。

根据设计草图，系统主页的实现结果如图5.2所示：



图5.2系统主页界面

### 录入项目界面

项目信息分三步输入，在同一个页面实现不同表单的切换。

根据设计草图，录入项目界面的实现结果如图5.3所示：

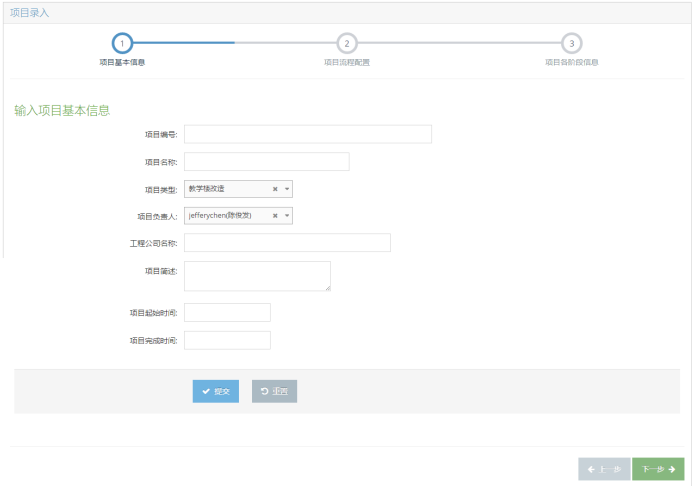


图5.3录入项目界面

### 项目列表界面

列表页主要使用了bootstrap的表格组件，间隔分明。且实现了分页查询功能，在页面下方具有分页组件，能跳到相应的列表页面。页面上方还能进行关键字的查询，查询结果也进行了分页处理。

根据设计草图，项目列表界面如图5.4所示：

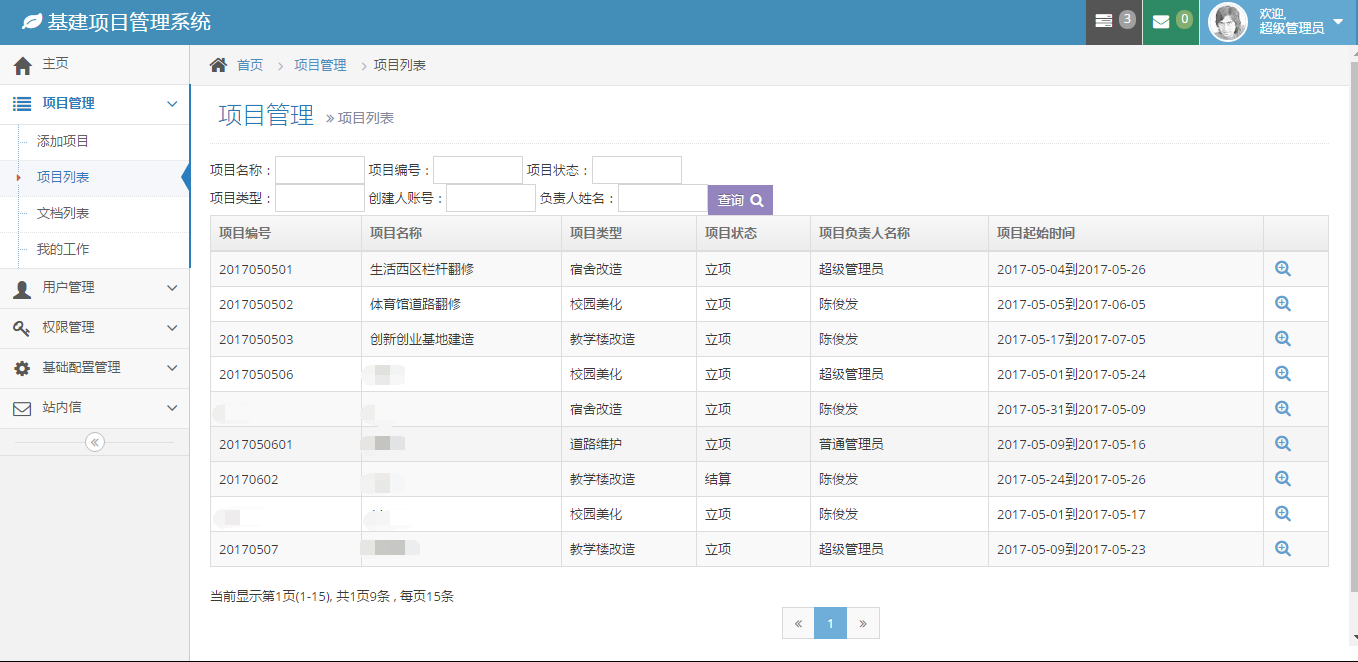


图5.4项目列表界面

### 发送信件界面

选择系统中的某一用户，写信完成后可发送。

根据设计草图，发送信息界面的实现结果如图5.5：

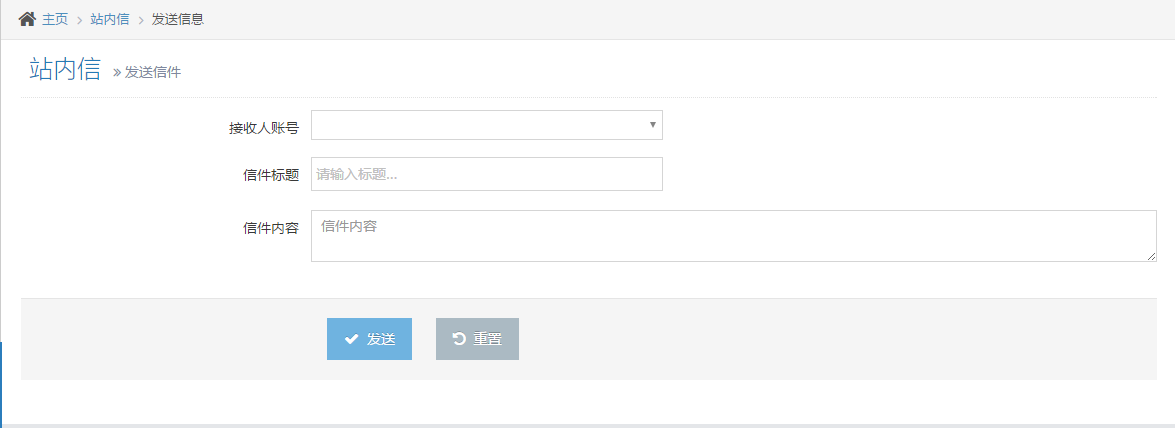


图5.5发送信件界面

### 上传附件界面

上传附件的前端页面主要采用了uploadify插件，用户点击按钮后选择相应文件进行异步上传，上传完成后会返回该文件的uri地址，用户填写其余信息后进行附件信息的添加。

根据设计草图，上传附件界面的实现结果如图5.6：

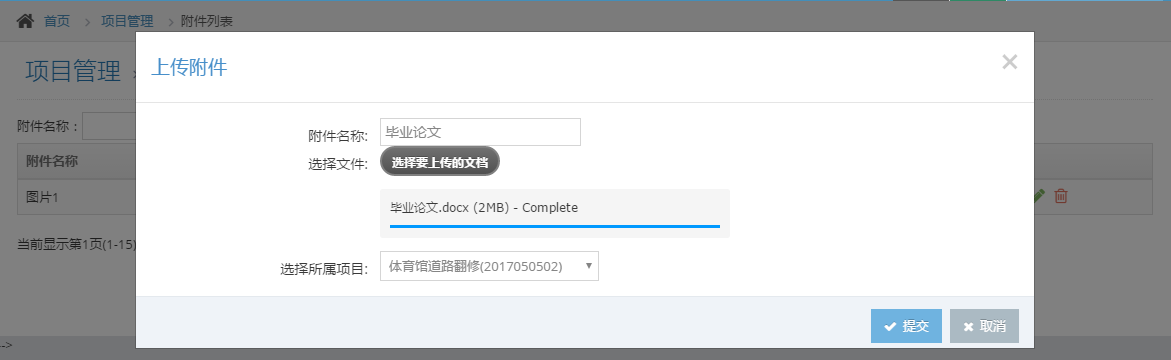


图5.6上传附件界面

## 功能模块实现

根据功能模块的设计，本节对系统的主要功能模块实现进行详细的描述。包括登录、录入项目基本信息、获取项目列表、修改项目基本信息、权限控制。

### 登录

处理逻辑：

1. 获取用户输入的用户名、密码；
2. 判断用户名和密码是否为空，若其中一项为空则返回登录界面，提示“用户名或密码不能为空”；
3. 判断用户名、密码是否正确，若不正确，则返回登录界面，提示出错信息。
4. 登录成功，进入系统首页。

在“登陆”页面填写完表单信息后提交，相应的controller匹配请求并通过SpringMVC框架获取用户对象的相关参数，控制层判断PortalUser对象的合法性，若合法则调用业务处理对象的相关方法进行登陆，将数据写入数据库[14]。在登陆过程中涉及到的权限获取及判断的相关逻辑实现，在后续的权限控制中进行说明讲解。

关键代码：

@RequestMapping(value = "/login", method = RequestMethod.POST)

public String check(Model model) throws ServiceException {

String uname = MVCUtil.getParam("uname");

String passwd = MVCUtil.getParam("passwd");

if (StringUtils.isBlank(uname) || StringUtils.isBlank(passwd)) {

model.addAttribute("msg", "用户名或密码不能为空!");

return "login";

}

passwd = DigestUtils.md5Hex(passwd);//密码进行加密

//得到Subject及创建用户名/密码身份验证Token（即用户身份/凭证）

UsernamePasswordToken token = new UsernamePasswordToken(uname, passwd);

token.setRememberMe(true);

Subject subject = SecurityUtils.getSubject();

try {

subject.login(token);//登陆，即身份验证

return "redirect:/index";

} catch (UnknownAccountException e) {

model.addAttribute("msg", "用户名或密码错误");

return "login";

} catch (IncorrectCredentialsException e) {

model.addAttribute("msg", "系统错误");

return "login";

}

}

### 录入项目基本信息

在“录入栏目”页面填写完表单信息后提交，相应的controller匹配请求并通过SpringMVC框架获取项目基本信息对象的相关参数，控制层判断ProjectInfo对象的合法性，若合法则调用业务处理对象进而调用数据访问对象的addProjectInfo方法，将数据写入数据库。

关键代码：

@RequestMapping(value = "project/addProjectInfo", method = RequestMethod.POST)

@ResponseBody

public void addProjectInfo(ProjectInfo projectInfo,Model model){

AjaxData ajaxdata;

String msg = null;

if (StringUtils.isBlank(projectInfo.getProjectName())) {

msg = "项目名称不能为空！";

} else if (StringUtils.isBlank(projectInfo.getProjectNum())) {

msg = "项目编号不能为空！";

} else if (StringUtils.isBlank(projectInfo.getTypeName())) {

msg = "请选择项目的类型！";

} else if (StringUtils.isBlank(projectInfo.getDescription())) {

msg = "项目描述不能为空！";

} else if (null==projectInfo.getLeaderId()) {

msg = "请选择项目负责人！";

}

//若存在不符合条件的参数，直接返回错误信息

if (StringUtils.isNotBlank(msg)) {

ajaxdata = new AjaxData(false, null, msg);

MVCUtil.ajaxJson(ajaxdata);

return;

}

try {

PortalUser portalUser = (PortalUser) SecurityUtils.getSubject().getPrincipal();

//设置数据对象的录入员账号、创建时间

projectInfo.setEntryStaffAccount(portalUser.getAccount());

projectInfo.setCreateTime(new Date());

int projectId = projectService.addProjectInfo(projectInfo);

JsonObject data = new JsonObject();

data.addProperty("projectId", projectId);

ajaxdata = new AjaxData(true, data, null);

} catch (ServiceException e){

ajaxdata = new AjaxData(false, null, "添加项目失败！");

LOGGER.error(e.getMessage());

}

MVCUtil.ajaxJson(ajaxdata);

}

public int addProjectInfo(Object object) throws ServiceException {

return addReturnGeneratedKey(object);//持久层框架封装的函数，返回自动编号

}

### 修改项目基本信息

在修改项目基本信息的悬浮框中填写修改的信息后提交，controller匹配请求并获取请求参数存入相应的数据对象ProjectInfo中，判断数据对象的合法性后调用业务对象，进而调用数据库操作对象执行modifyProjectInfo函数进行修改操作，将修改项更新到数据库中。

关键代码：

@RequestMapping(value = "project/modifyProjectInfo", method = RequestMethod.POST)

@ResponseBody

public void modifyProjectInfo(ProjectInfo projectInfo,Model model){

AjaxData ajaxdata;

String msg = null;

if (StringUtils.isBlank(projectInfo.getProjectName())) {

msg = "项目名称不能为空！";

} else if (StringUtils.isBlank(projectInfo.getTypeName())) {

msg = "请选择项目的类型！";

} //还有其他判断条件

if (StringUtils.isNotBlank(msg)) {

ajaxdata = new AjaxData(false, null, msg);

MVCUtil.ajaxJson(ajaxdata);

return;

}

try {

projectService.modifyProjectInfo(projectInfo);

ajaxdata = new AjaxData(true, null, null);

} catch (ServiceException e){

ajaxdata = new AjaxData(false, null, "修改项目信息失败！");

LOGGER.error(e.getMessage());

}

MVCUtil.ajaxJson(ajaxdata);

}

public int modifyProjectInfo(Object object) throws ServiceException {

//object为待更新的实体，后面的参数是作为where条件的实体属性名

return update(object, "id");

}

### 删除项目

在列表页面点击相应项目记录的“删除”按钮，handle匹配请求并调用相应的处理器controller，获取项目自动编号，调用业务模型进行业务处理，进而调用数据库操作对象执行deleteProjectById函数，将数据库中的项目信息删除。

关键代码：

@RequestMapping(value = "project/delete", method = RequestMethod.GET)

public String deletePaymentUser(Model model) throws ServiceException {

int id = MVCUtil.getIntParam("id");

if(id>0){

projectService.deleteProjectInfo(id);

}

return "redirect:/project/show\_project\_list";

}

public void deleteProjectById(int projectId) throws ServiceException {

// TODO Auto-generated method stub

delete("DELETE FROM project\_info where id=?", projectId);

}

### 获取项目列表

点击“项目列表”菜单项或在项目列表页面点击“搜索”后，相应处理器controller进行请求处理，调用业务模型查询符合条件的数据对象列表，并返回相应的ModelAndView对象进行渲染然后将视图结果显示到客户端。

@RequestMapping(value=" show\_project\_list",method=RequestMethod.GET)

public String showProjectList(Model model) throws ServiceException {

int pageid = MVCUtil.getIntParam("pageid");

QueryProjectVo queryProjectVo = null;

if (pageid <= 0) {

pageid = 1;

MVCUtil.removeSessionAttribute(PROJECT\_SEARCH);

} else {

Object obj = MVCUtil.getSessionAttribute(PROJECT\_SEARCH);

if (obj != null)

queryProjectVo = (QueryProjectVo) obj;

}

PageBean pagebean = new PageBean(pageid, PAGESIZE);

List<QueryProjectVo> projecctInfoList = projectService

.queryProjectInfoListByPage(queryProjectVo, pagebean);

model.addAttribute("projectInfoList", projecctInfoList);

return "project\_list";

}

public List<QueryProjectVo> queryProjectInfoListByPage(QueryProjectVo queryProjectVo,

PageBean pageBean) throws ServiceException {

StringBuffer querySqlBuffer = new StringBuffer();

List<Object> queryParams = new ArrayList<Object>();

querySqlBuffer.append("SELECT pi.id, pi.project\_name, pi.project\_num, pi.type\_name, pi.entry\_staff\_account, pi.description, pu.user\_name, pi.company, pi.status, pi.project\_start\_date, pi.project\_end\_date, pi.create\_time, pi.update\_time FROM project\_info AS pi JOIN portal\_user AS pu on(pi.leader\_id=pu.id) WHERE 1=1");

if (queryProjectVo != null) {

if (StringUtils.isNotBlank(queryProjectVo.getProjectName())) {

querySqlBuffer.append(" AND pi.project\_name LIKE ?");

queryParams.add("%" + queryProjectVo.getProjectName() + "%");

}

//其他搜索条件暂时忽略

}

List<QueryProjectVo> projectInfoList = projectInfoDao.queryProjectByPage(pageBean, querySqlBuffer.toString(), queryParams);

return projectInfoList;

}

### 权限控制

权限控制主要通过Shiro框架进行控制。在登陆时，对相关密码验证通过的用户进行授权，通过StstemAuthorizingRealm类中的doGetAuthorizationInfo方法进行授权查询及鉴权。一方面通过页面的< @shiro.hasPermission >标签进行权限限制，若不存在该权限则不显示该菜单项。另一方面，通过@ RequiresPermissions标签对各个controller的权限限制，若用户不存在该权限的标识则无法调用到相应的controller函数，避免出现跳过页面限制直接发送请求调用被限制的方法的情况。

关键代码：

protected AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principals) {

PortalUser portalUser = (PortalUser) getAvailablePrincipal(principals);

if (portalUser != null) {

try {

SimpleAuthorizationInfo info = new SimpleAuthorizationInfo();

List<PortalRole> PortalUserRoles = portalUserDao

.listPortalUserRoles(portalUser.getId());

if (PortalUserRoles != null && PortalUserRoles.size() > 0) {

for (PortalRole portalUserRole : PortalUserRoles) {

info.addRole(portalUserRole.getRoleCode());

}

}

List<PortalFunction> portalUserFunctions = portalUserDao

.listPortalUserFunctions(portalUser.getId());

if (portalUserFunctions != null&& portalUserFunctions.size() > 0) {

for (PortalFunction portalUserFunction : portalUserFunctions) {

info.addStringPermission(portalUserFunction.getPermission());

}

}

return info;

} catch (ServiceException e) {

logger.error(

"SystemAuthorizingRealm.doGetAuthorizationInfo faild!",e);

}

return null;

} else

return null;

}

<@shiro.hasPermission name="portal:project:manage">

<li class="">

<a href="#" class="dropdown-toggle">

<i class="menu-icon fa fa-list"></i>

<span class="menu-text"> 项目管理 </span>

<b class="arrow fa fa-angle-down"></b>

</a>

<b class="arrow"></b>

<ul class="submenu">

<@shiro.hasPermission name="portal:project:add">

<li class="" id="level2">

<a href="${ctx}/project/add\_project" target="iframe\_content">

<i class="menu-icon fa fa-caret-right"></i>

添加项目

</a>

<b class="arrow"></b>

</li>

</@shiro.hasPermission>

</ul>

</li>

</@shiro.hasPermission>

@RequiresPermissions("portal:project:add")

@RequestMapping(value = "project/addProjectInfo", method = RequestMethod.POST)

@ResponseBody

public void addProjectInfo(ProjectInfo projectInfo,Model model){

//中间逻辑忽略，重点为RequiresPermissions标签

}

# 总结

本文根据高校基建项目的管理现状，结合国内外基建项目管理情况，对高校基建项目的管理系统需求进行了收集和分析。在此基础上，对系统进行了概要设计与详细设计，包括系统的架构设计、界面设计、功能设计、数据库设计。基于设计的结果进行了系统的数据库实现及编码实现，完成了一个基于J2EE平台的高校基建项目管理系统。在系统功能模块的实现描述上因为篇幅问题未能全部叙述，但功能都已完成。本系统综合地应用了大学期间所学的知识，结合了Java Web目前流行的框架技术及前端流行框架，完整地实现了一个能够满足高校基建处需求的基建项目管理系统。

由于作者时间与精力的限制，本系统还存在一定的不足，今后可以在以下几个方面继续研究：

1. 前端页面实现标签化，避免过多的页面跳转，提高工作效率。
2. 提供邮件或短信提醒功能，提醒负责人完成项目相关工作。
3. 优化数据库的设计，提高数据存储和检索效率。
4. 提供更强大的项目信息统计功能。
5. 优化站内信的设计，实现一对多发送、在线讨论等功能。

# 参考文献

1. SUN Shuyan，QI XiaoJuan，YANG YongJi**．**The Design of Scientific Research Project Management System in Universities[J] **．**Management Science and Engineering，2014，81：57-61**．**
2. 郑彩云**．**高校基建管理模式现状及发展研究[J] **．**新校园·上旬刊，2014,4：59**．**
3. Joshua Bloch**．**Effective Java[M] **．**Addison-Wesley，2008：25-55**．**
4. Mert Caliskan，Kenan Sevindik**．**Spring入门经典[M] **．**北京：清华大学出版社，2015：1-158**．**
5. Craig Walls**．**Spring实战（第四版）[M] **．**北京：人民邮电出版社，2016：19-20**．**
6. Charles Forsythe**．**Instant FreeMarker Starter [M] **．**Packt Publishing，2013：20-90**．**
7. 脚本之家**．**使用Java进行FreeMarker的web模板开发的基础教程[DB/OL] **．**

http://www.jb51.net/article/80361.htm，2016-03-02**．**

1. 许晓斌**．**Maven实战[M] **．**北京：机械工业出版社出版，2011：50-80**．**
2. 谭云杰**．**大象：Thinking in UML[M] **．**北京：中国水利水电出版社，2012：39-42**．**
3. 百度百科**．**B/S结构[DB/OL] **．**http://baike.baidu.com/item/B%2FS结构，2017-03-24**．**
4. 王珊，萨师煊**．**数据库系统概论[M] **．**北京：高等教育出版社，2006：79，198-200**．**
5. 张海藩,牟永敏**．**软件工程导论[M] **．**北京：清华大学出版社，2015：15-188**．**
6. 王飞飞**．**MySQL数据库应用从入门到精通[M] **．**北京：中国铁道出版社，2014：50-128**．**
7. 陈亚辉，缪勇**．**Struts2+Spring+Hibernate框架技术与项目实战[M] **．**北京：清华大学出版社，2012：50-328**．**