**目 录**

[**1 绪 论** 1](#_Toc6243693)

[1.1 研究的背景及意义 1](#_Toc6243694)

[1.1.1 选题的背景 1](#_Toc6243695)

[1.1.2 国内外研究现状 1](#_Toc6243696)

[1.1.3 研究的意义 2](#_Toc6243697)

[1.2 系统目标 2](#_Toc6243698)

[**2 需求分析** 3](#_Toc6243699)

[2.1 业务需求 3](#_Toc6243700)

[2.2 功能需求 9](#_Toc6243702)

[2.2.1 角色分析 9](#_Toc6243703)

[2.2.2 业务功能 10](#_Toc6243704)

[2.3 非功能需求 18](#_Toc6243705)

[2.3.1环境需求 18](#_Toc6243706)

[2.3.2 性能需求 18](#_Toc6243707)

[2.3.3 安全需求 19](#_Toc6243708)

[**3 总体设计** 20](#_Toc6243709)

[3.1系统设计的原则 20](#_Toc6243710)

[3.2 系统体系结构设计 20](#_Toc6243711)

[3.3 系统功能结构设计 21](#_Toc6243712)

[**4 数据库设计** 23](#_Toc6243713)

[4.1 概念结构设计 23](#_Toc6243714)

[4.1.1 设计思路 23](#_Toc6243715)

[4.1.2 E-R图 23](#_Toc6243716)

[4.2 逻辑结构设计 26](#_Toc6243717)

[4.2.1 设计思路 26](#_Toc6243718)

[4.2.2 逻辑模型 26](#_Toc6243719)

[4.3 物理结构设计 29](#_Toc6243720)

[4.3.1 存取方式 29](#_Toc6243721)

[4.3.2 存储结构 29](#_Toc6243722)

[**5 界面设计** 30](#_Toc6243723)

[5.1 界面关系图或工作流图 30](#_Toc6243724)

[5.2 界面设计成果 30](#_Toc6243725)

[5.2.1 主界面 30](#_Toc6243726)

[5.2.2 子界面 32](#_Toc6243727)

[**6 详细设计** 34](#_Toc6243728)

[6.1 系统主要功能模块介绍 34](#_Toc6243729)

[6.2功能模块设计 34](#_Toc6243730)

[6.2.1用户管理 34](#_Toc6243731)

[6.2.2实验室管理 37](#_Toc6243732)

[6.2.3实验室预约管理 40](#_Toc6243731)

[6.2.4信息统计报表管理 44](#_Toc6243732)

[7**编码** 51](#_Toc6243736)

[7.1 代码实现与核心算法 51](#_Toc6243734)

[7.2 代码优化分析 54](#_Toc6243735)

[**8 测试** 5](#_Toc6243736)4

[8.1 测试方案设计 5](#_Toc6243737)4

[8.1.1测试策略 5](#_Toc6243731)5

[8.1.2测试进度安排 5](#_Toc6243731)5

[8.1.3测试资源 5](#_Toc6243731)5

[8.1.4关键测试点 5](#_Toc6243731)5

[8.2 测试用例构建 5](#_Toc6243737)4

[8.2.1测试用例编写约定 5](#_Toc6243731)5

[8.2.2测试用例设计 5](#_Toc6243731)5

[8.2.3关键测试用例 5](#_Toc6243731)5

[8.2.4测试用例维护 5](#_Toc6243731)5

[**9 总结与展望** 5](#_Toc6243747)6

[9.1 设计工作总结 5](#_Toc6243748)6

[9.2 未来工作展望 5](#_Toc6243748)6

[**谢 辞** 5](#_Toc6243750)7

[**参考文献** 5](#_Toc6243751)8

[附录A 外文翻译—原文部分 6](#_Toc6243753)0

[附录B 外文翻译—译文部分 6](#_Toc6243753)2

[附录C 软件使用说明书 6](#_Toc6243754)4

[附录D 主要源代码 6](#_Toc6243755)8

**1 绪论**

## 1.1 研究的背景及意义

### 1.1.1选题的背景

国内各高校在管理实验室上都开始信息化建设并逐步推进改革，师生用户对于实验室也有了更高目标的需求，实验室工作管理人员的管理事务从实验室、设备、学生、实验课程等方面在不断变多。师生要求越来越高，管理事务越来越繁重，高校就迫切需要一套基于线上平台的具有高效快捷、数据充分共享、功能完备的综合管理系统，来解决实验室的管理乱象，以提高管理人员的工作效率，增强校园师生用户的使用体验。

之前高校普遍采用人工管理的方式来管理校园实验室，但这种管理方式不但严重消耗人力物力，对资源有着较大的浪费，而且容易造成实验室管理系统的混乱，比如说学生预约的实验室可能已被占用。因此，我们需要研究符合当下的高校实验室管理系统。

### **1.1.2 国内外研究现状**

国内因计算机技术起步较国外晚，且对实验室的投入建设不足，因此国内各高校实验室管理系统发展较为缓慢。目前国内各高校结合自己学校的实际情况，有些是购买现成的实验室管理软件，但费用较高且软件公司不提供源代码，维护不方便，常出现有些功能用不上，有些想用的功能却没有；而有的高校则采取和软件公司合作，由高校提出具体需求，然后由软件公司开发，这种情况同样开发费用较高；有的高校则由有网络程序开发经验的老师带上几位学生组成开发团队，负责整套系统的开发。通过期刊网查询，发现有以下几所高校的研究生开发过相类似的系统：电子科技大学开发基于校园网WEB的实验室管理系统;湖南中医药大学基于ASP.NET2.0开发了开放式的管理系统;湖南师范大学开发基于SSH 框架的管理系统。

而在国外，从20世纪70年代末至今，经历了科技技术的爆炸性发展，对于高校实验室管理系统的研究也较国内更快，国际上采用最为广泛的ASP.NET管理系统已经发展到了第四代，这一管理系统能极大提升用户的体验，且能够大大降低运行该管理系统所需的成本费用。

### **1.1.3 研究的意义**

国内各高校管理系统具有一定的局限性，不能完全满足学生远程预约课程的需求，与国外所普遍使用的管理系统有较大差距以及国内整个系统的继承性较差，且相对功能都不是很完备，难以满足高校新型实验的师生教学需求，而各高校又要求系统有较为完备功能。由此，当前高校的实验室管理系统仍有很大的探索空间，进行对于高校实验室管理系统的研究十分有意义。

## 1.2 系统目标

本课题想通过设计一套功能较完备的实验室管理系统，师生能通过系统能预约实验室及实验设备，能上传下载实验资料，管理员可以轻松地管理实验室的事务，如能清楚了解各实验室实验设备预约使用情况，对预约等一系列操作进行审批管理，并能够对预约使用情况做出统计报表等功能。该系统应具有一定的可用性及维护性，并能让学校根据自身需求选择适合的实验室管理系统，且能方便师生用户的日常使用和学校管理人员的维护工作。

该系统采用面向对象方法，利用UML建模技术来实现系统的需求分析以及详细设计，确定系统需求，设计系统相对应的结构、UI、数据库等。编码阶段利用SSM框架，采用Java编程语言，根据需求分析内容和详细设计内容编写系统代码，实现系统功能并且提高系统的性能。在实现系统功能用到相应的web技术，例如HTML、CSS、Ajax等相关技术。

本实验室管理系统被分为六个模块，其中包括用户登录注册管理模块、用户功能管理模块、话题管理模块、设备管理模块、预约管理模块、数据统计管理模块等。登录注册管理模块包括用户信息的登录注册等。用户功能管理模块，包括用户个人信息显示、修改、删除、增加以及用户预约实验室和实验设备的功能。话题管理模块包括话题发布和浏览各话题的功能。设备管理模块包括设备信息的增加、修改、删除、查询功能以及设备报修、设备申请用、设备维修、设备采购功能等。预约管理模块包括显示实验室及设备使用情况功能、预约实验室设备功能、审核预约申请功能等。数据统计管理模块包括设备采购金额统计、实验室实验设备预约申请数目统计功能等。

# 2 需求分析

## 2.1 业务需求

(1)能够实现用户信息的管理、实验室及设备信息的管理、预约选择、发布公告、课件发布下载等功能，对高校实验室的全面管理。

(2)实现对实验室预约过程中产生的所有业务数据的管理，如预约统计信息的管理与维护等。

(3)具有配套的系统后台管理维护功能，能够对物品信息、用户信息、系统日志等信息进行管理与维护，不同类别的用户具有不同的权限。

(4)响应速度合理，安全性较高。

(5)系统运行稳定，并且应易于维护。

(6)方便用户使用，可用性较高。

2.1.1主要业务流程

（1）用户登录注册活动图

用户信息登录注册功能如图2-1所示：

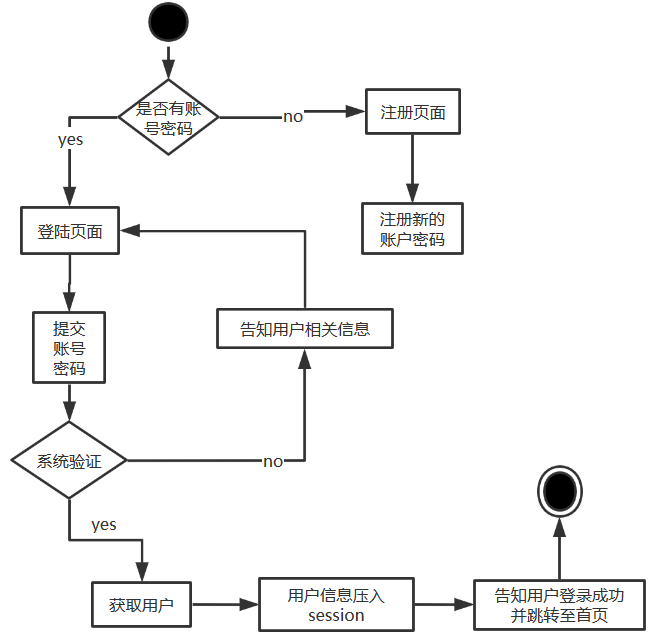


图2-1用户登录注册活动图

（2）预约实验室、实验器材活动图

用户预约实验室、实验器材功能如图2-2所示：

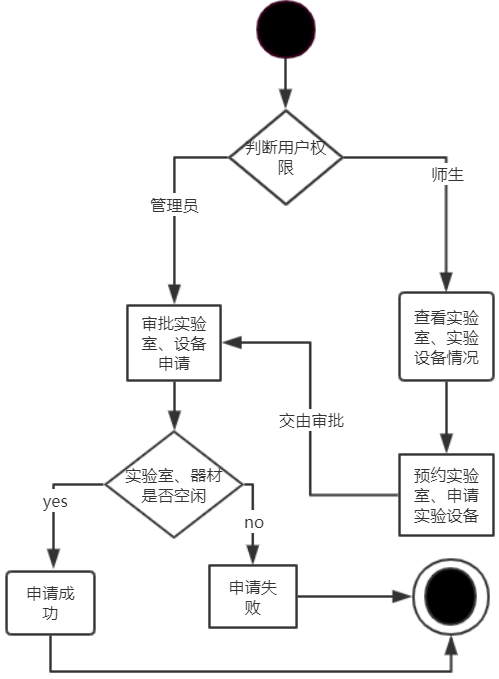


图2-2预约实验室实验器材活动图

（3）实验室课程管理活动图

实验课程管理如图2-3所示：

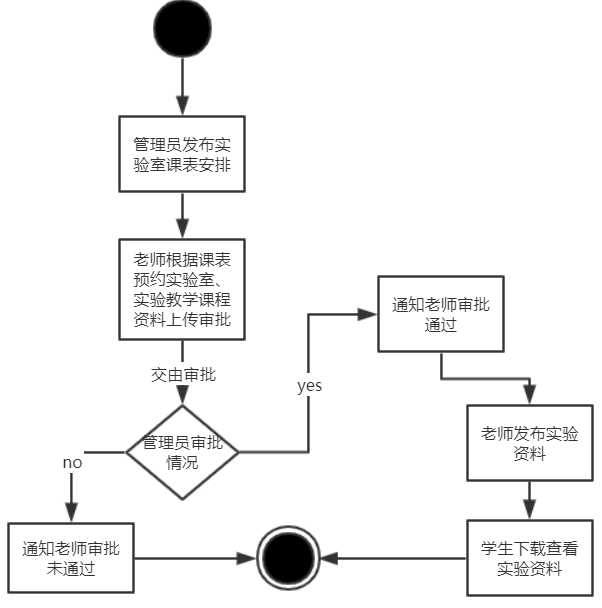


图2-3实验室课程管理活动图

（4）实验室管理活动图

实验室功能管理如图2-4所示：

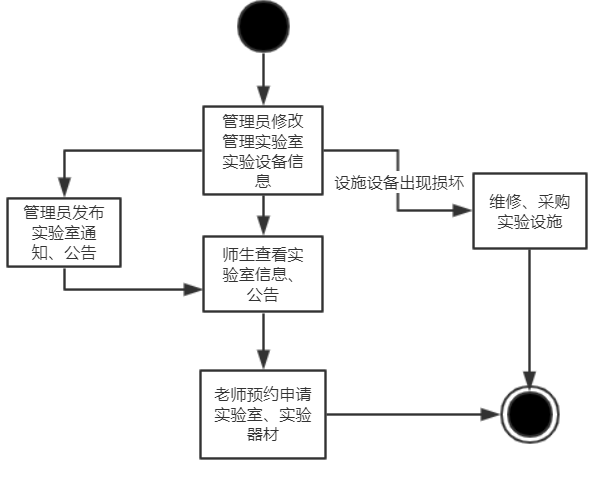


图2-4实验室管理活动图

（5）实验室设备管理活动图

实验室设备信息管理功能如图2-5所示：

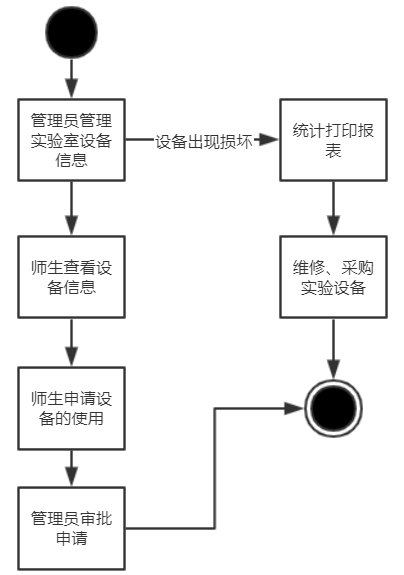


图2-5实验室设备信息管理活动图

（6）实验室信息统计报表活动图

实验室信息统计报表功能如图2-6所示：

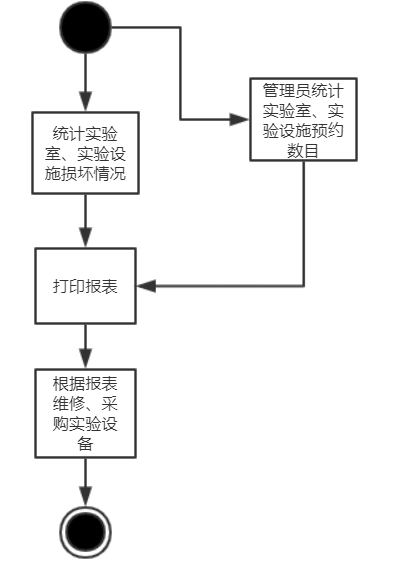


图2-6实验室信息统计报表活动图

## 2.2 功能需求

### 2.2.1 角色分析

系统各角色职责功能如表2-1所示：

表2-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 用户（老师） | 管理自己的用户信息、预约实验室及设备、上传实验课所需资料等功能 |
| 用户（学生） | 管理个人用户信息、预约实验室及设备、下载实验所需教材等功能 |
| 系统管理员 | 维护整个系统的用户组织结构，负责对系统所有合法用户信息和实验室及设备信息的增删改查及信息的统计报表操作、审核师生预约处理、统计预约数目及设备报修数目等功能 |

### 2.2.2 业务功能

以下从业务角度出发，给出了系统的总体用例图，包含登录注册模块、用户信息管理、实验室及设备信息管理、课程管理、报表统计管理、预约审核管理和课件上传下载管理等用例，系统总业务功能如图2-7所示：

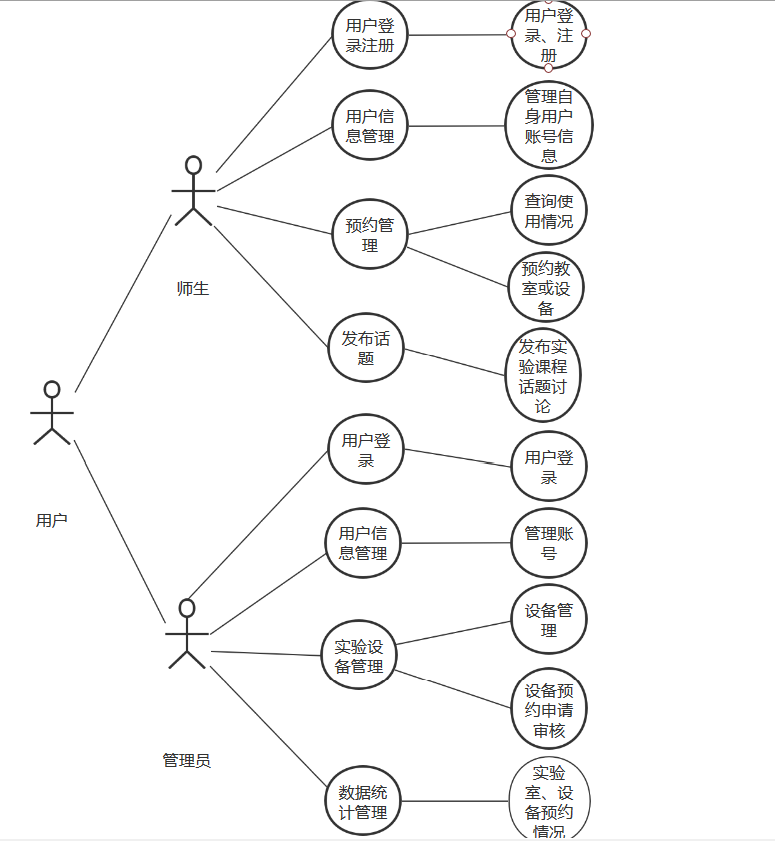


图2-7系统总体用例图

（1）用户登录注册

用户登录注册用例如图2-8所示：

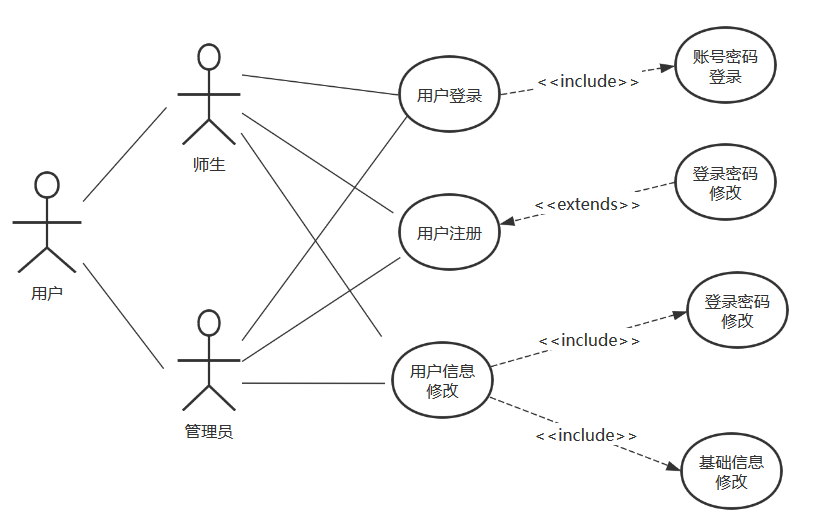


图2-8用户登录注册用例图

用户登录注册用例描述如表2-2所示：

表2-2用户登录注册用例描述

| **描述项** | **说明** |
| --- | --- |
| **用例名称** | 登录注册 |
| **标识符\*** | YL01 |
| **用例描述** | 登录注册是系统对用户身份识别的验证模块，当用户在提供有效的账号和密码后，并向系统后台进行身份验证，返回验证后的信息，根据返回信息判断身份验证是否通过。若用户暂时还没有在该系统完成注册或想要在这个平台注册一个新的用户号码，可以注册合法的账户和密码，再返回登录页面。 |
| **参与者表** | 所有用户 |
| **优先级** | 1 |

续表2-2

| **状态\*** | 进行中 |
| --- | --- |
| **前置条件** | 用户成功进入登录系统页面后输入对应账户名和密码 |
| **后置条件** | 系统提示用户登录成功或告之登录失败并再次跳转至登录页面 |
| **基本操作流** | 1. 填写账号、密码  2. 系统验证账号、密码是否填写完整  3. 如果验证不完整，提示用户按要求填写必要的信息  4. 将有效的用户账号、密码提交到服务器验证  5. 系统根据服务器验证结果进行相应操作  6. 如果是账号或密码操作，直接提示账号或密码有误，不可以直接提示账号错误或密码错误  7. 如果验证通过，直接登录到系统主界面。 |
| **可选操作流** | 1a. 登录信息不充分：    1a1. 系统提示登录信息不充分  1a2. 用例结束  2a. 用户取消登录：  2a1. 用户取消登录  2a2. 用例结束  3a. 用户注册信息已被占用  3a1. 用户注册信息被后台提示已存在  3a2. 系统提醒用户注册信息已经存在，要求用户重新输入新的注册信息  3a3. 用例结束 |
| **被泛化用例表** | 该用例的特化用例列表 |

（2）用户信息管理

用户信息管理如图2-9所示：

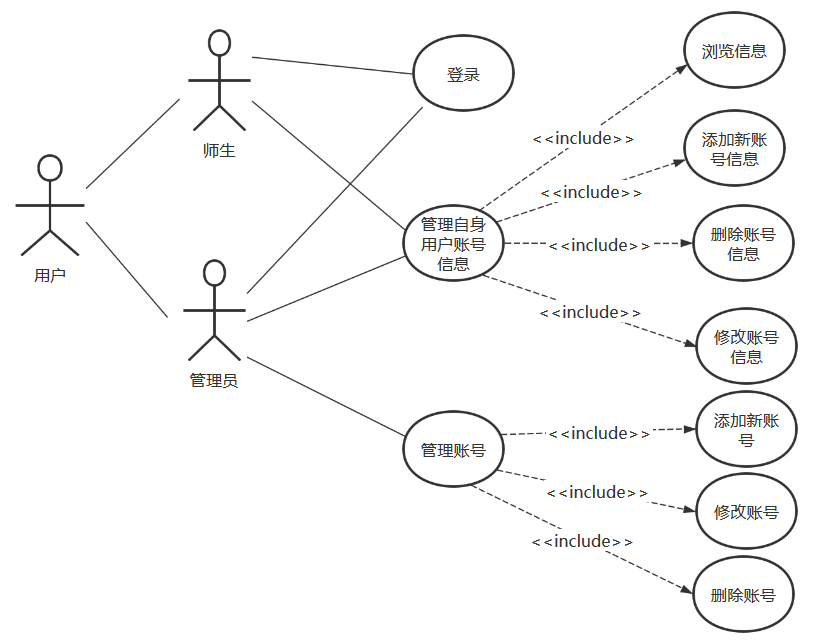


图2-9 用户信息管理用例图

用户信息用例描述如表2-3所示：

表2-3用户信息管理用例描述

|  |  |
| --- | --- |
| **描述项** | **用户信息管理用例描述** |
| **用例名称** | 用户信息管理 |
| **标识符\*** | YL02 |
| **用例描述** | 描述了不同权限身份用户管理储存在后台信息的过程 |
| **参与者表** | 所有用户 |
| **优先级** | 2 |
| **状态\*** | 进行中 |
| **前置条件** | 用户登录系统成功，进入系统主页面 |

续表2-3

|  |  |
| --- | --- |
| **后置条件** | 系统显示用户注册时所填写的所有相关个人信息 |
| **基本操作流** | 1.用户在系统登录界面上输入对应账号密码，进入主界面。  2.在主界面的选项框内，点击用户信息框进入用户相关界面查看个人信息。 |

1. 预约管理

预约管理如图2-10所示：

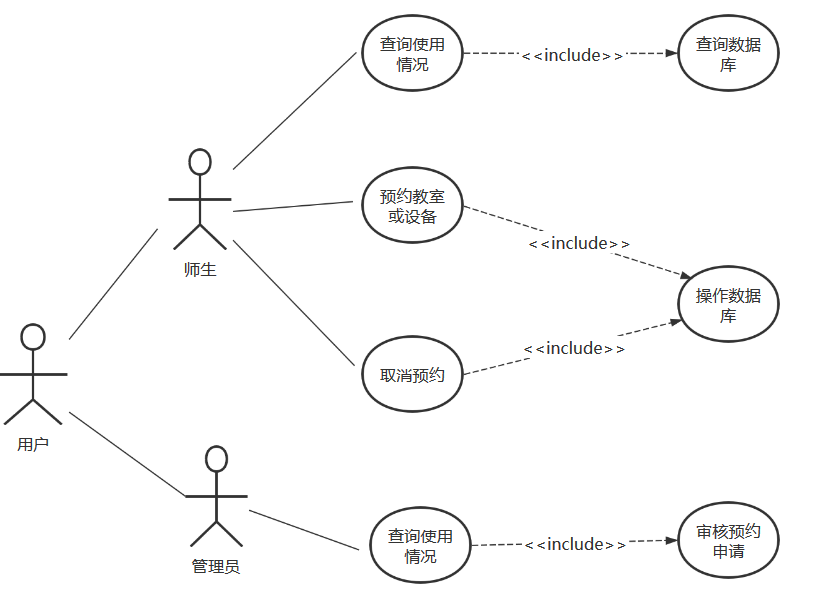


图2-10 预约管理用例图

用户预约管理用例描述如表2-4所示：

表2-4预约管理用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 预约管理 |
| **标识符\*** | YL03 |
| **用例描述** | 描述了用户使用本系统进行预约实验室及设备的整个过程 |
| **参与者表** | 老师学生用户 |
| **优先级** | 2 |
| **状态\*** | 进行中 |
| **前置条件** | 用户登录系统成功，并对实验室或试验设备有预约需求 |
| **后置条件** | 预约申请获批，用户能在预约时间内使用实验室或设备；申请失败，无法使用实验室实验设备 |
| **基本操作流** | 1. 用户进入预约界面  2. 用户浏览实验室实验设备使用情况  3. 根据自身需求、使用情况进行预约  4. 管理员审核预约  5. 如果预约未通过，直接提示预约失败  6. 如果预约通过，显示预约成功 |
| **可选操作流** | 1a. 用户想预约的实验室或实验设备已被占用：    1a1. 系统提示不能预约  1a2. 用例结束  2a. 管理员在一定期限内未对预约申请完成审核：    2a1. 系统提示错误  2a2. 用例结束 |

（4）发布话题

发布话题流程如图2-11所示：

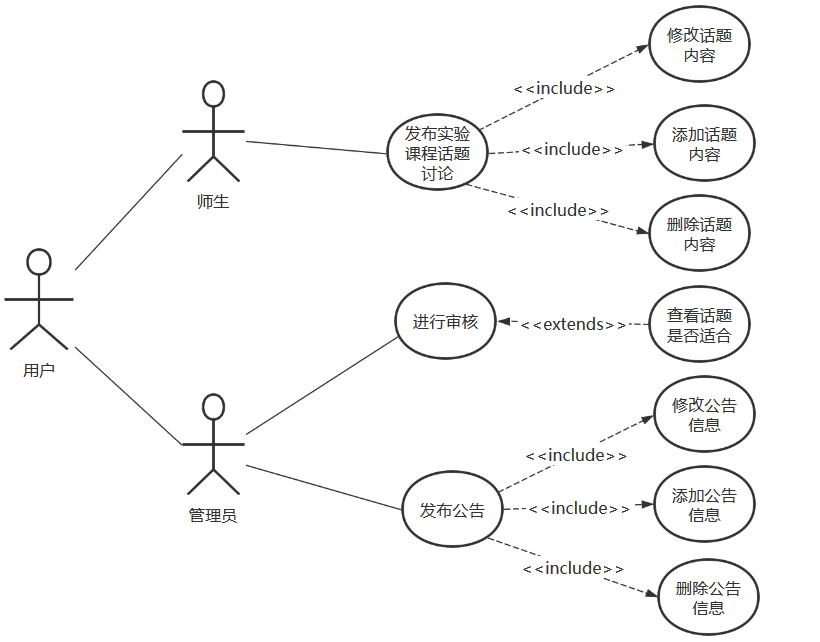


图2-11 发布话题用例图

用户发布话题用例描述如表2-5所示：

表2-5发布话题用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 话题发布 |
| **标识符\*** | YL04 |
| **用例描述** | 描述了管理员发布话题公告，师生讨论话题过程 |
| **参与者表** | 所有系统用户 |
| **优先级** | 3 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已成功登录系统 |
| **后置条件** | 系统给出发送成功提示信息 |

续表2-5

|  |  |
| --- | --- |
| **基本操作流** | 1. 用户进入话题区界面  2. 用户浏览话题区帖子  3. 根据自身需求发布通知或帖子  4. 管理员进行审核帖子内容  5. 审核通过与他人讨论并做出评论，审核未通过发布失败 |
| **可选操作流** | 1a. 用户发布的话题经审核不符合规定：    1a1. 用户发布的话题不符合规定  1a2. 话题被删除  1a3. 用例结束  2a. 用户取消信息编辑及发布：    2a1. 发布取消  2a2. 用例结束 |

（5）实验设备管理

实验设备管理流程如图2-12所示：

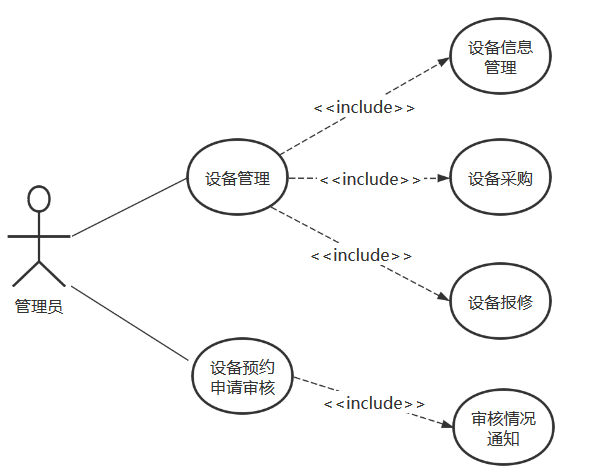


图2-12 实验设备管理用例图

实验设备管理用例描述如表2-6所示：

表2-6实验设备管理用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 实验设备用例 |
| **标识符\*** | YL05 |
| **用例描述** | 管理员对设备信息的修改等操作管理 |
| **参与者表** | 管理员用户 |
| **优先级** | 3 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已成功登录系统且设备使用情况发送变化 |
| **后置条件** | 系统给出数据修改成功信息提示，后台同步更新变化数据 |
| **基本操作流** | 1. 管理员点击实验设备框，进入设备管理页面  2. 管理员修改实验设备信息  3. 管理员统计实验设备损坏情况  4. 管理员根据损坏情况采购、维修实验设备  5. 管理员对实验设备的申请使用作出审批 |
| **可选操作流** | 1a. 用户预约的实验设备已被占用：    1a1. 预约的实验设备已被占用  1a2. 申请被取消  1a3. 用例结束  2a. 师生取消实验设备的申请：    2a1. 申请取消  2a2. 用例结束 |

（6）数据统计管理

数据统计管理流程如图2-13所示：

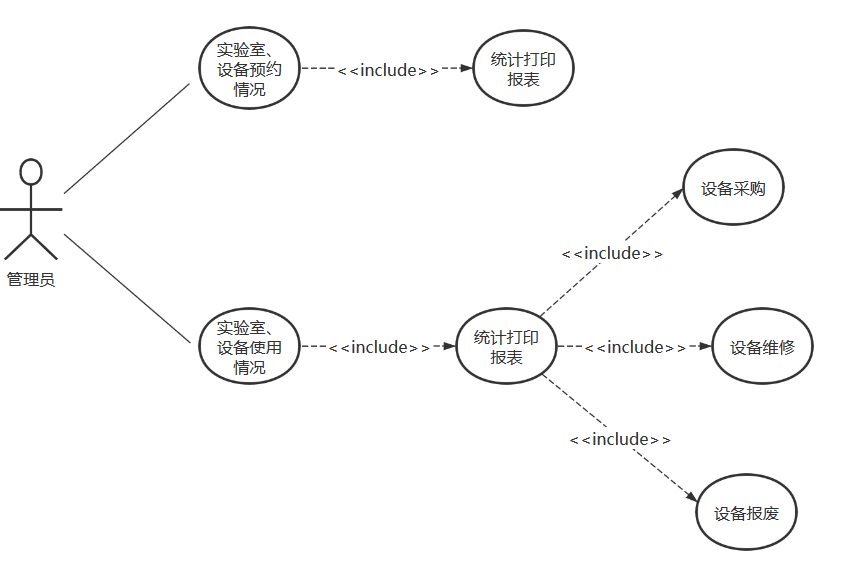


图2-13 数据统计管理用例图

数据统计管理用例描述如表2-7所示：

表2-7实验室、设备数据使用统计管理用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 实验室及设备数据使用统计管理 |
| **标识符\*** | YL06 |
| **用例描述** | 描述了用户对实验设施损坏数目、金额以及实验室预约情况的统计模块整个过程 |
| **参与者表** | 限管理员用户 |
| **优先级** | 3 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员凭管理员权限登录系统 |
| **后置条件** | 系统根据使用信息打印统计 |

续表2-7

|  |  |
| --- | --- |
| **基本操作流** | 1. 管理员点击数据统计框，进入数据统计管理页面  2. 管理员浏览实验室及设备的使用情况、预约情况  3. 管理员统计实验设备损坏情况  4. 管理员根据使用清况、预约情况打印采购维修实验设备的报表  5. 管理员根据报表作出采购、维修设备 |

## 2.3 非功能需求

### 2.3.1环境需求

### 服务器端环境如表2-8所示：

表 2-8 服务器端环境

|  |  |
| --- | --- |
| 系统所使用数据库 | Microsoft SQL Server 2008 SP1 |
| 操作系统 | Win 10 |
| Web服务器 | IIS 7.5 |
| 数据库 | SQL Server 2008 |

客户端环境如表2-9所示：

表 2-9 客户端环境

|  |  |
| --- | --- |
| 操作系统 | Windows 7 及其以上版本的操作系统 |
| 浏览器 | IE 6.0版本以及其以上版本、Firefox、Chrome |
| 分辨率 | 最佳效果为1024×768像素 |

### 2.3.2 性能需求

（1）可维护性：系统可以较方便地找回后台储存的北方数据，实现对丢失数据的恢复工作。能较方便快捷地对系统某些单一独立功能模块进行适当修改的功能。

（2）易操作性：系统应提供美观实用和立体直观的图形化用户管理界面，充分考虑用户的习惯，简单明了，拥有一些菜单的处理快捷键，方便用户操作和管理。

（3）可扩展性：系统从系统结构、功能设计、管理对象等多方面的功能扩展来考虑，以满足不同用户对于系统扩充功能的要求。系统还应采取有一定扩充能力的硬件和软件平台，并充分考虑软件和硬件的可扩展性。系统应具备灵活的可扩充能力，来适应软件和硬件的更新换代。

### 2.3.3 安全需求

（1）容错性：当用户在输入或误操作导致产生非法数据时，系统应具有一定的容错机制及审核机制。在这种情况发生时，系统应对用户做出提示，提示用户重新输入或者进行其他操作。

（2）系统的外在环境安全：系统自身的安全系统应该包括对于系统信息的备份，不同用户权限的识别和其他措施，以确保数据和系统信息不被泄露。

（3）系统内部安全：除确保外部系统安全，系统还应该保证授权用户的合法使用。

（4）系统运行安全：系统能够在用户误操作导致的数据损坏或数据丢失的问题下找回丢失数据，实现一定程度的数据复用。

3 总体设计

## 3.1系统设计的原则

从本系统的总体结构出发，本系统的总体设计应遵循以下三条设计原则：

1. 实用性原则。实验室管理系统的好坏将主观影响到学校师生的学习情况，因此系统应满足师生正常学习体验。该系统要求能够满足不同类别用户的实际需求，以求解决以往高校实验室管理系统功能不够完备的缺点。
2. 安全性原则。系统所提供的安全机制能防止因用户的非法操作而导致发生错误，并对不同类别的用户有不同的权限划分。
3. 可扩展性原则。系统主要由SSM框架构成，而SSM框架通常应预留接口以备后续系统功能升级。

## 3.2 系统体系结构设计

本系统采用使用较广的B/S架构模式，B/S结构如图3-1所示：

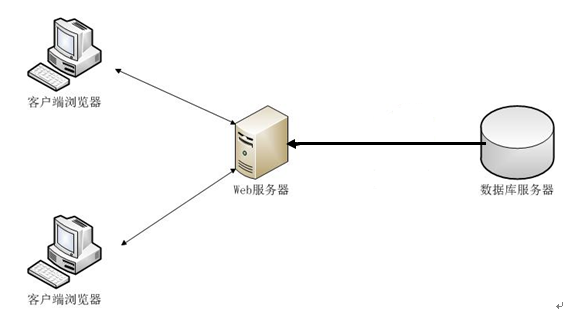


图3-1 B/S体系结构

B/S结构，即浏览器与服务器交互的架构模式。浏览器方面是客户端在浏览器运行程序，由此客户不需要了解程序具体工作流程，只需在浏览器上即可运行程序。服务器端是指完成了的应用程序要部署在服务器上才可以运行，客户在浏览器输入相应的IP地址，即可运行应用程序。

Spring MVC框架，其中MVC字母分别代表模型、视图与控制器。

Struct2的工作原理是客户端先发送请求，请求经过过滤器，FilterDispatcher被调用，ActionMapper决定调用对应action，之后请求交给ActionProxy，后向Struct.xml配置文件处理找到调用的action类，生成实例。

Spring的核心是IOC，主要包括依赖查找和依赖注入两种方式，IOC通过配置文件让Spring管理所有Bean类。ApplicationContext和BeanFactory是Spring的核心接口。其中ApplicationContext是高级容器，它继承了多个接口；BeanFactory也被叫做低级容器，是生成Bean实例的工厂。IOC过程是加载Bean到低级、高级容器中，然后调入getBean，获取Bean，注入实例。

MyBatis内部对JDBC进行封装处理,因此可以直接编写SQL语句，MyBatis通过xml文件或注解配置statement然后映射动态参数生成能执行的SQL语句，最后MyBatis执行SQL语句并映射为java对象。

系统结构如图3-2所示：

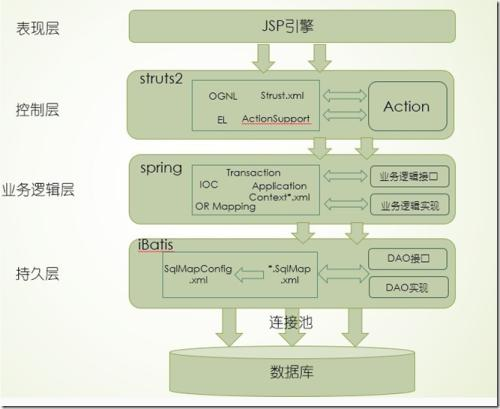


图3-2 系统结构图

系统主要由四层组成。表现层是主要用来管理用户的请求以及响应，并用来显示数据和接收传输用户数据等功能来保证用户体验；控制层是调用下层的逻辑层，其职能包括负责读取来自于表现层传输的数据，控制用户的输入；业务逻辑层可以调用最底层持久层的相对应的方法来实现与数据库信息的交互，或直接把执行的结果送回上一层处理；持久层是负责向一或多个存储器存储数据的一组类，它的功能包括与数据库直接进行交互处理。

## 3.3 系统功能结构设计

实验室系统管理总包图如图3-3所示：

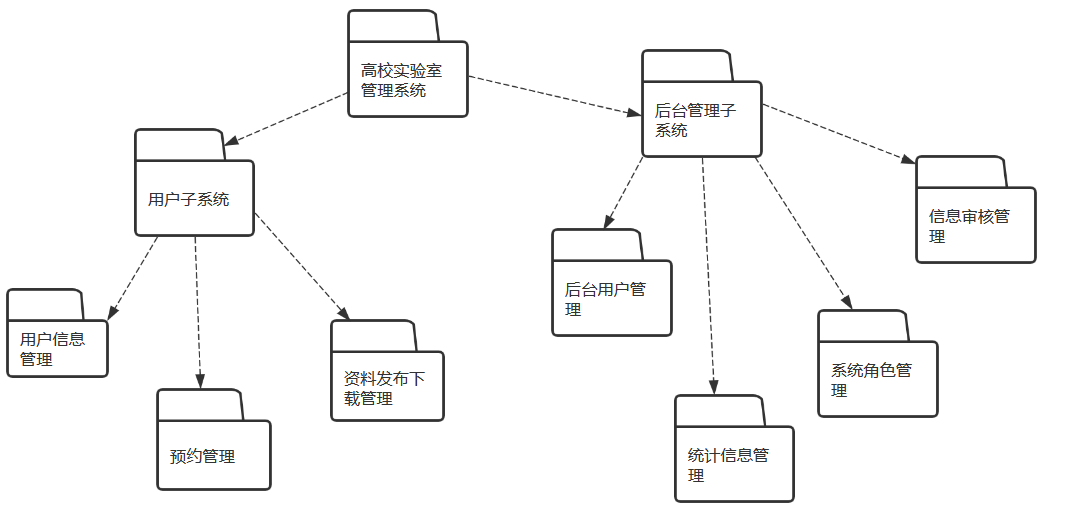


图3-3高校实验室管理系统总包图

如图3-3所示，系统包含用户子系统模块和后台管理子系统。其中，用户子系统模块由用户信息管理、预约管理、资料发布下载管理3个模块组成。后台管理子系统分为后台用户管理、统计信息管理、系统角色管理和信息审核管理4个模块。

1. 用户信息管理：管理员用户有权限对所有用户的信息进行修改查询操作，师生用户仅能查询个人注册时填写信息。
2. 预约管理：师生用户都可以查看实验室、设备使用情况，并根据信息进行相应预约操作。
3. 资料发布下载管理：老师上传教学所用资料以便学生下载。
4. 后台用户管理：后台储存信息应与页面显示保持一致，且实时更新。
5. 统计信息管理：管理员有权限对实验室和设备使用情况分类统计，并能用不同表格形式统计。
6. 系统角色管理：系统用户分为两大类，一类是师生用户，一类是管理员用户，不同类别用户被赋予权限不同，登录界面不同，能点击按钮也不同。
7. 信息审核管理：管理员能对用户注册、预约及上传资料等操作进行审核处理。

# 4 数据库设计

## 4.1 概念结构设计

### **4.1.1 设计思路**

实验室管理系统的数据库概念结构设计，首先从E-R图开始设计，有了E-R模型，就可以清楚地看出各实体之间存在的关系。该系统的主要实体有用户（用户包括学生、教师和管理员，这三者在同一张表结构中，但可以用角色的不同权限字段来区分）、角色权限、实验室、设备、实验室日志、设备日志和通知等多个实体。师生与设备是一对多的关系，管理员和设备形成一对多；学生/教师与管理员都与角色权限有一对一的关系；师生管理员与通知都是一对多；学生/教师与实验室有一对多的关系，而实验室日志与实验室是多对一的关系。

### **4.1.2 E-R图**

系统总E-R图如图4-1所示：

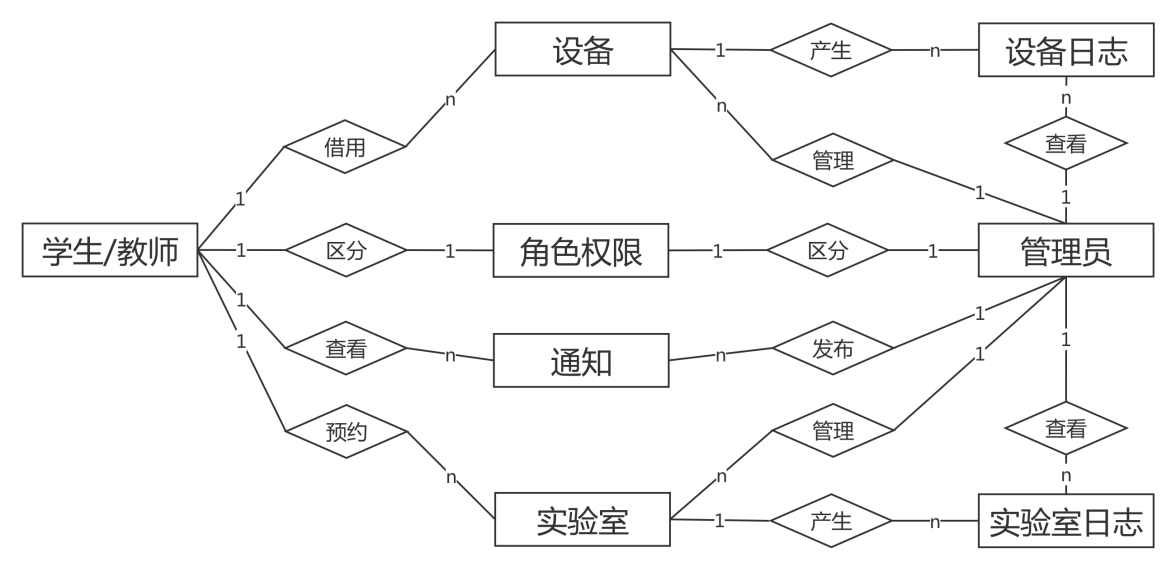


图4-1 总E-R图

（1）用户实体

用户实体图如图4-2所示：

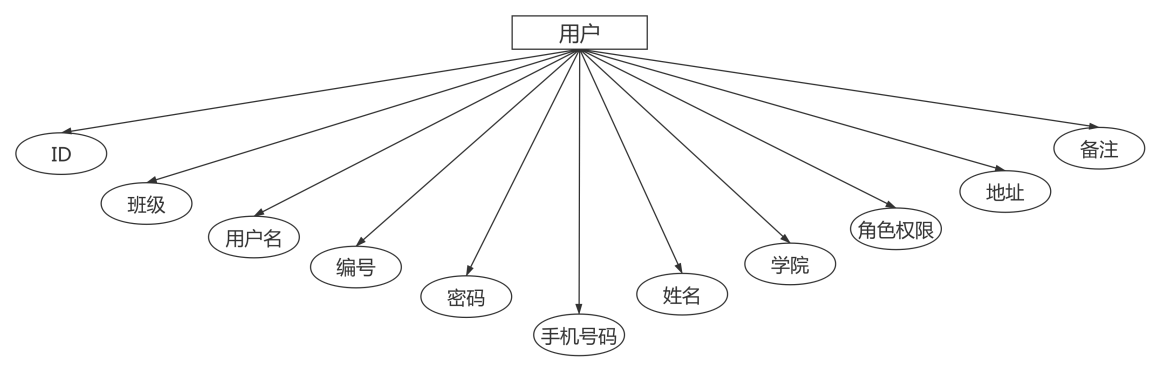


图4-2用户实体图

（2）角色权限实体

角色的不同权限实体图如图4-3所示：

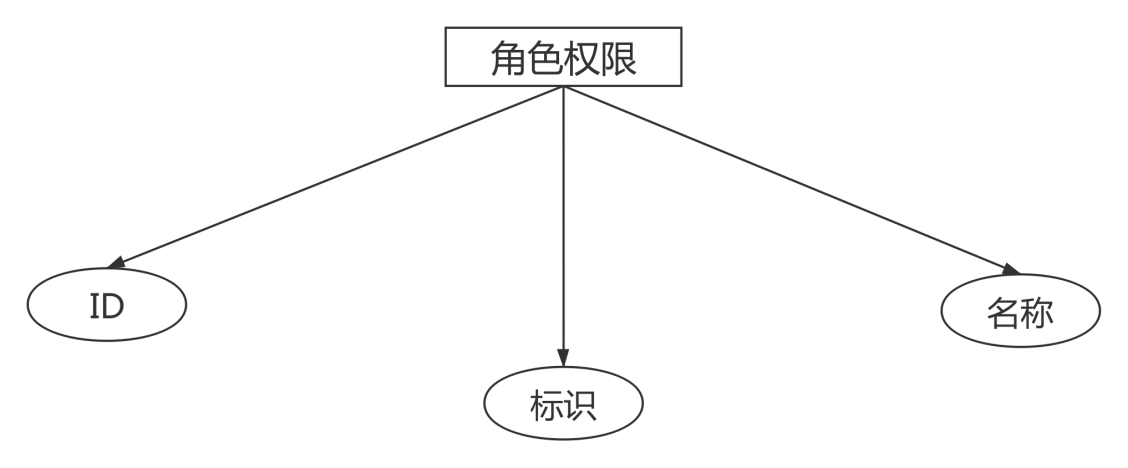


图4-3 角色权限实体图

（3）实验室实体

实验室相关信息实体图如图4-4所示：

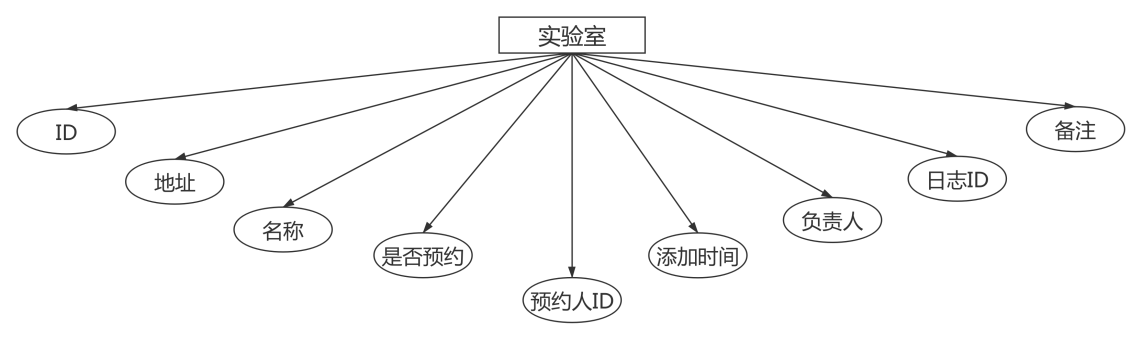


图4-4 实验室实体图

（4）设备实体

设备情况实体图如图4-5所示：

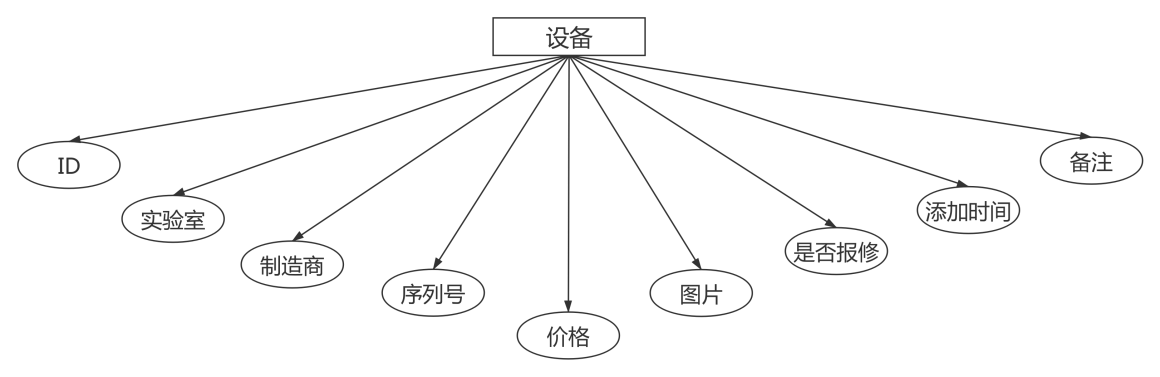


图4-5 设备实体图

（5）实验室日志实体

实验室日志信息实体图如图4-6所示：

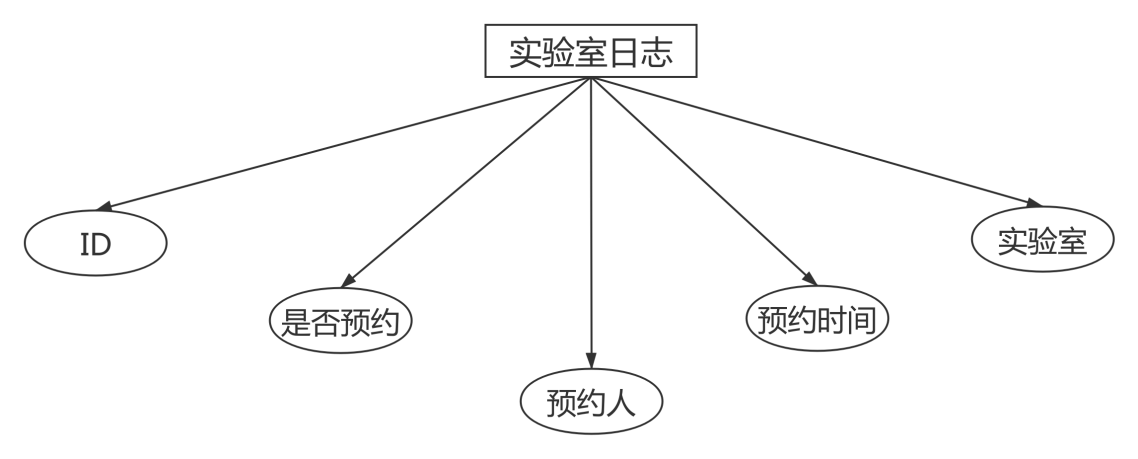


图4-6 实验室日志实体图

（6）设备日志实体

设备日志实体情况如图4-7所示：

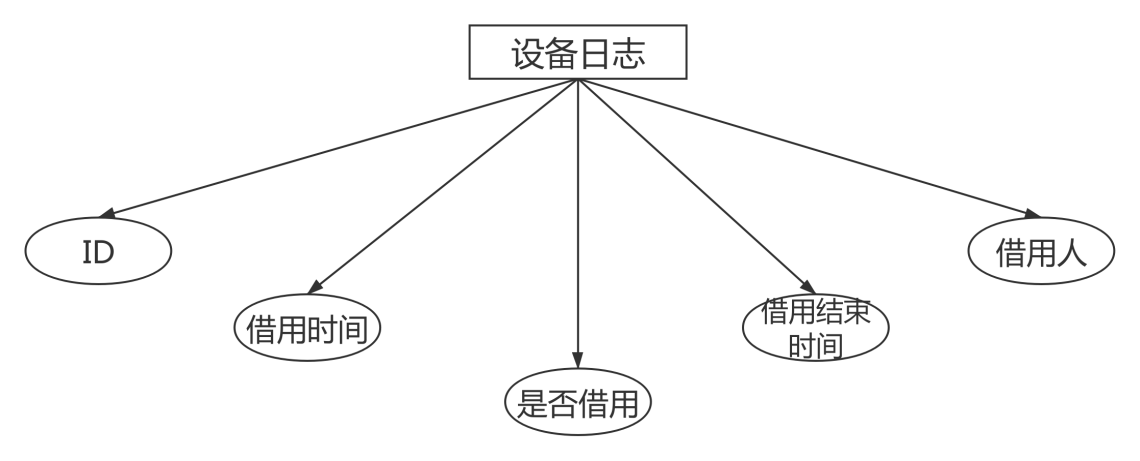


图4-7 设备日志实体图

（7）通知实体

通知处理实体图如图4-8所示：

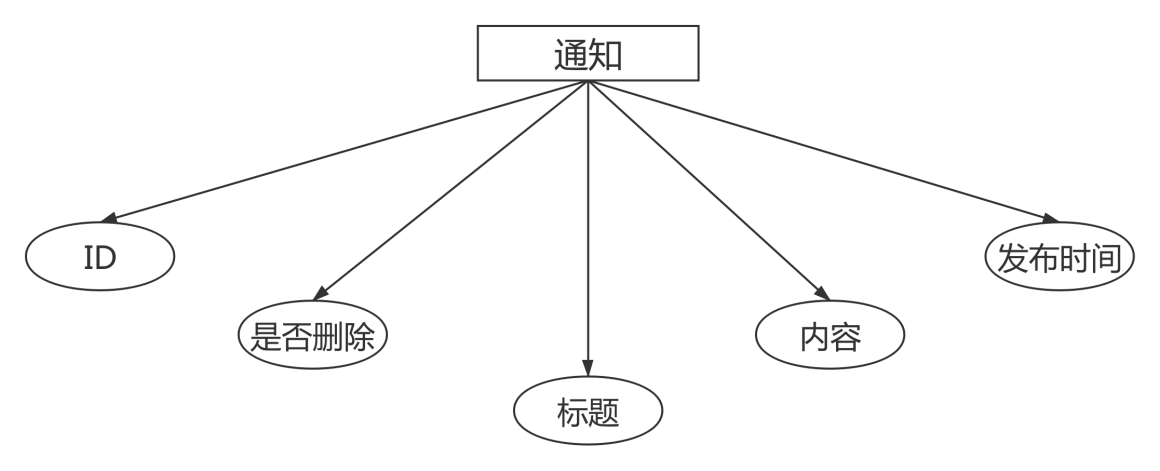


图4-8 通知实体图

## 4.2 逻辑结构设计

### **4.2.1 设计思路**

在E-R图完成实体模型的图形化后进入设计阶段，就可以通过概念模型的设计的研究成果根据实体模型图形化建立关系模型。关系模型的意义在于将E-R模型转化为表格的形式，使得关系更直观立体，便于理解。二维表格则包含了基础的数据如字段名等、记录以及记录之间所存在的联系、安全性和说明等。因此得出的逻辑结构模型是否能与概念结构设计的E-R图模型一一对应，以及从数目和用途上是否能够确保用户的使用需求，都需要进行核查处理。

本系统的数据库名称为db\_lab\_manager，数据库表有：用户表（t\_user）、角色权限表（t\_role）、 实验室表（t\_laboratory）、设备表（t\_equipment）、实验室日志表（t\_laboratorylog）、设备日志表（t\_equipmentlog）、通知表（t\_notice）、固定资产表（t\_purchase）、设备维修日志表（t\_repairlog）。

### **4.2.2 逻辑模型**

（1）用户表(t\_user)用于存储用户信息，如表4-1所示：

表4-1 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 用户ID |
| bj | varchar(255) | N | Y | 班级 |
| isDelete | int(11) | N | Y | 是否删除 |
| isSh | int(11) | N | Y | 是否审核 |
| name | varchar(255) | N | Y | 用户名 |
| number | varchar(2) | N | Y | 编号 |
| pass | varchar(255) | N | Y | 密码 |
| phone | varchar(255) | N | Y | 电话 |
| realName | varchar(255) | N | Y | 真实姓名 |
| xy | varchar(255) | N | Y | 学院 |
| role\_id | int(11) | N | Y | 权限角色ID |
| address | varchar(255) | N | N | 地址 |
| time | datetime | N | Y | 注册时间 |
| bz | varchar(255) | N | N | 备注 |

（2）角色权限表（t\_role）用于存储角色权限信息，如表4-2所示：

表4-2 用户权限表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 角色权限ID |
| enName | varchar(255) | N | Y | 标识 |
| name | varchar(255) | N | Y | 名称 |

（3）实验室表（t\_laboratory）用于存储实验室信息，如表4-3所示：

表4-3 实验室表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 实验室ID |
| address | varchar(255) | N | Y | 地址 |
| bz | varchar(255 | N | Y | 备注 |
| fzr | varchar(255) | N | Y | 负责人 |
| isDelete | int(11) | Y | N | 是否删除 |
| isYy | int(11) | N | Y | 是否预约 |
| name | varchar(255) | N | Y | 名称 |
| time | datetime | N | Y | 添加时间 |
| user\_id | int(11) | N | Y | 用户ID |
| laboratoryLogId | int(11) | N | Y | 实验室日志ID |

（4）设备表（t\_equipment）用于存储设备信息，如表4-4所示：

表4-4 设备表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 设备ID |
| bz | varchar(255) | N | Y | 备注 |
| isBx | int(11) | N | Y | 是否报修 |
| isDelete | int(11) | N | Y | 是否删除 |
| jg | double | N | Y | 价格 |
| sbxlh | varchar(255) | N | Y | 设备序列号 |
| time | datetime | N | Y | 添加时间 |
| xh | varchar(255) | N | Y | 型号 |
| zzs | varchar(255) | N | Y | 制造商 |

续表4-4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| photo | varchar(255) | N | Y | 图片 |
| isJy | int(11) | N | Y | 是否借用 |
| JyUser\_id | int(11) | N | Y | 借用人ID |

（5）实验室日志表（t\_laboratorylog）用于存储实验室日志信息，如表4-5所示：

表4-5 实验室日志表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 实验室日志ID |
| isYy | varchar(30) | N | Y | 是否预约 |
| laboratory\_id | int(11) | N | N | 实验室ID |
| user\_id | int(11) | N | N | 预约人ID |
| time | datetime | N | N | 预约时间 |
| endTime | datetime | N | N | 预约结束时间 |

（6）设备日志表（t\_equipmentlog）用于存储设备日志信息，如表4-6所示：

表4-6 存储设备日志信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 设备日志ID |
| isYy | int(8) | N | Y | 是否借用 |
| user\_id | int(11) | N | N | 借用人ID |
| time | datetime | N | N | 借用时间 |
| endTime | datetime | N | Y | 借用结束时间 |

（7）通知表（t\_notice）用于存储通知信息，如表4-7所示：

表4-7 存储通知信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 通知ID |
| isDelete | int(11) | N | Y | 是否删除 |
| name | varchar(255) | N | Y | 标题 |
| nr | varchar(255) | N | Y | 内容 |
| time | datetime | N | Y | 发布时间 |

（8）固定资产表（t\_purchase）用于存储固定资产信息，如表4-8所示：

表4-8 存储固定资产信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 固定资产ID |
| count | int(11) | N | Y | 数量 |
| isDelete | int(11) | N | Y | 是否删除 |
| money | double | N | Y | 总价 |
| price | double | N | Y | 单价 |
| time | datetime | N | Y | 添加时间 |
| user\_id | int(11) | N | Y | 使用者ID |

（9）设备维修日志表（t\_repairlog）用于存储设备维修日志信息，如表4-9所示：

表4-9 存储设备维修日志信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 是否空 | 说明 |
| id | int(11) | Y | N | 设备维修日志ID |
| bxTime | datetime | N | Y | 报修时间 |
| bz | varchar(255) | N | Y | 备注 |
| endTime | datetime | N | Y | 报修结束时间 |
| title | varchar(255) | N | Y | 标题 |
| Equipment\_id | int(11) | N | Y | 设备ID |

## 4.3 物理结构设计

### **4.3.1 存取方式**

数据库的物理结构设计，重点是指数据的存取方式及不同存储结构类别的选择。其中选择哪种存取方式要根据所使用的关系型数据库系统其所支持的存取方式来确定。而数据库管理系统的存取方式包括的是索引方式和聚簇方式。索引方式又包括B+树索引以及hash索引；聚簇存取方法既能满足单个关系的连结，还可以进行连接多个关系，因此聚簇存取能大大提高查询数据的效率，但在连接时要确定是否进行单一连接或多个连接则会降低数据的性能。

**4.3.2 存储结构**

数据库实则是由一或多个表空间所构成的，而一个独立的表空间又是由一或多个数据文件所组成的，数据块又构成了数据文件，因此数据库中有无数个数据块。数据在逻辑上被存储在表空间中，在物理上又可以说是放在数据文件中。在设计数据库物理结构，是指根据逻辑结构选择数据库系统，再设计存储结构以及存取方式。

5 界面设计

## 5.1 界面关系图

高校实验室管理系统界面关系图如图5-1所示：

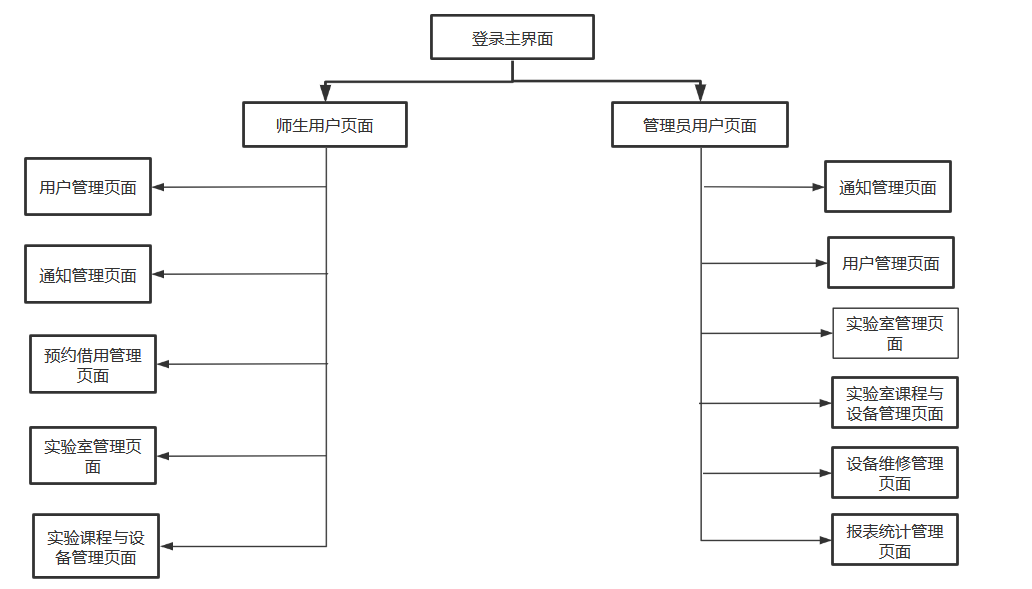


图5-1高校实验室管理系统界面关系图

## 5.2 界面设计成果

### **5.2.1 主界面**

（1）系统登录注册界面如图5-2所示：



图5-2系统登录注册界面

（2）管理员用户登录显示的主界面如图5-3所示：



图5-3 管理员用户登录显示主界面

（3）师生用户登录显示的主界面如图5-4所示：



图5-4 师生用户登录显示主界面

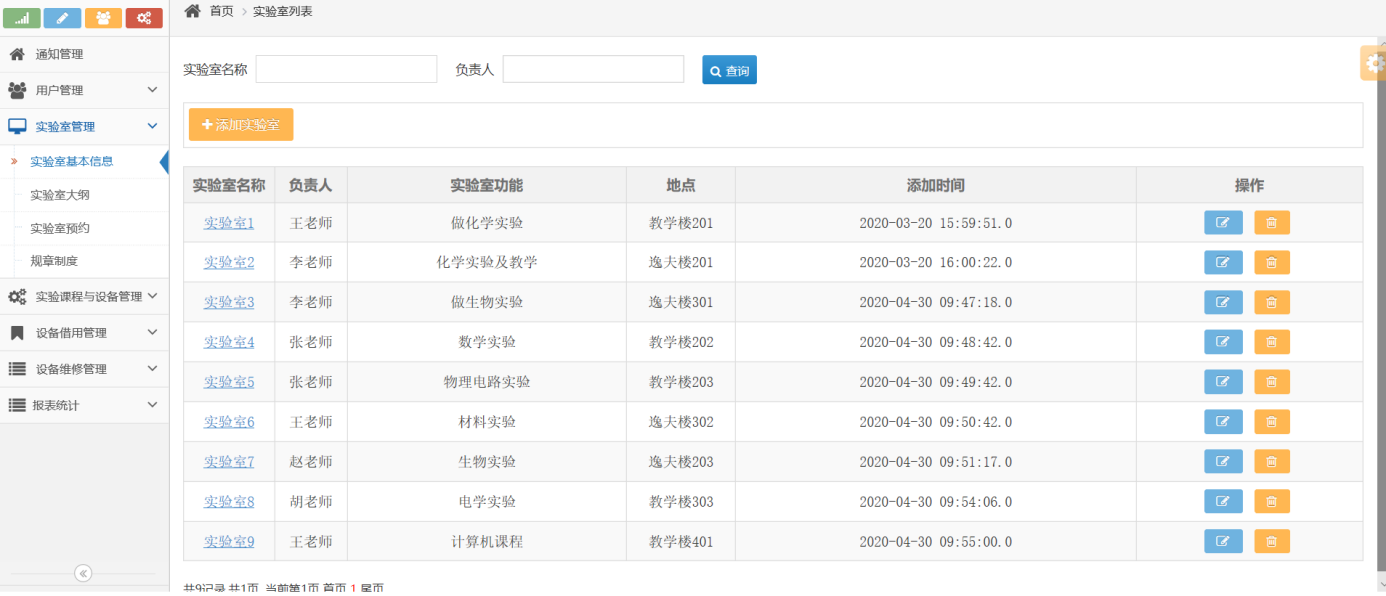
### **5.2.2 子界面**

（1）修改处理用户的个人信息页面如图5-5所示：



图5-5 修改用户信息页面

（2）实验室基本信息查询页面如图5-6所示：

图5-6实验室基本信息页面

（3）实验室预约管理页面如图5-7所示：

图5-7实验室预约页面

（4）设备申请维修处理页面如图5-8所示：



图5-8设备维修管理页面

（5）设备借用情况统计页面如图5-9所示：

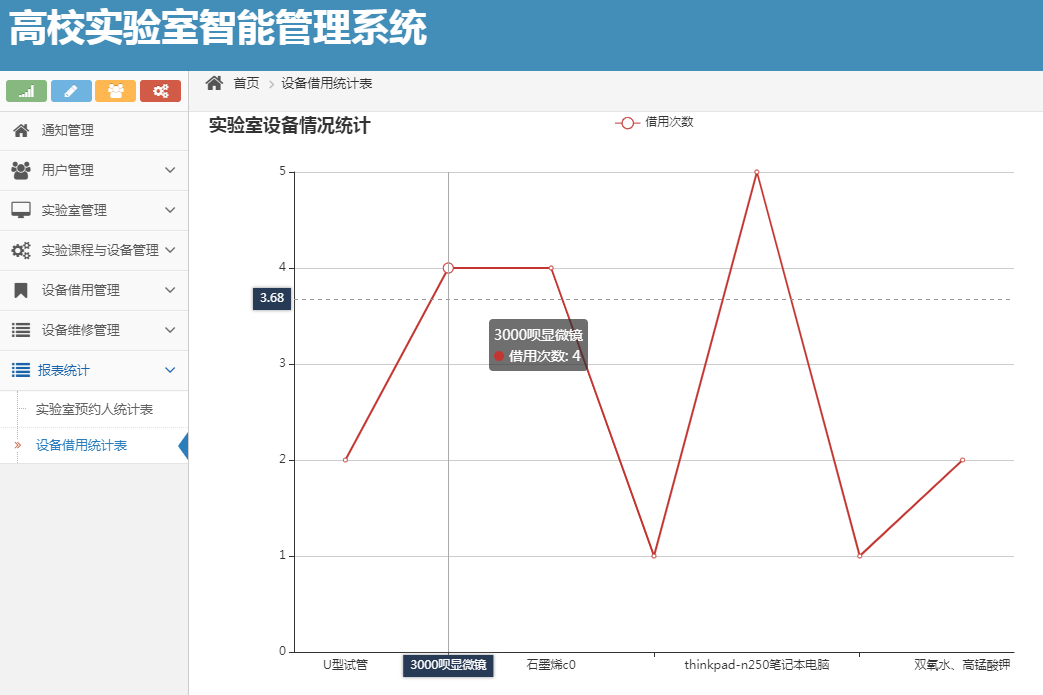


图5-9设备借用情况统计页面

6 详细设计

## 6.1 系统主要功能模块介绍

该高校实验室管理系统由登录注册模块、用户模块、话题模块、设备管理模块、预约模块、统计模块共六个模块组成。其中登录注册模块主要包括游客的注册以及师生用户和管理员用户的登录；用户模块的功能包括的是由不同权限来查看修改个人信息以及查看实验室及设备使用情况；话题模块的功能是发布浏览话题以及上传下载实验教学资料等；设备管理模块主要是包括实验设备的租借、报修、采购、审批处理等功能；预约模块由师生预约申请实验室及设备和管理员审核处理功能组成；统计模块包括设备采购金额数额统计和预约数量统计。

系统确定的总体功能结构如图6-1所示：

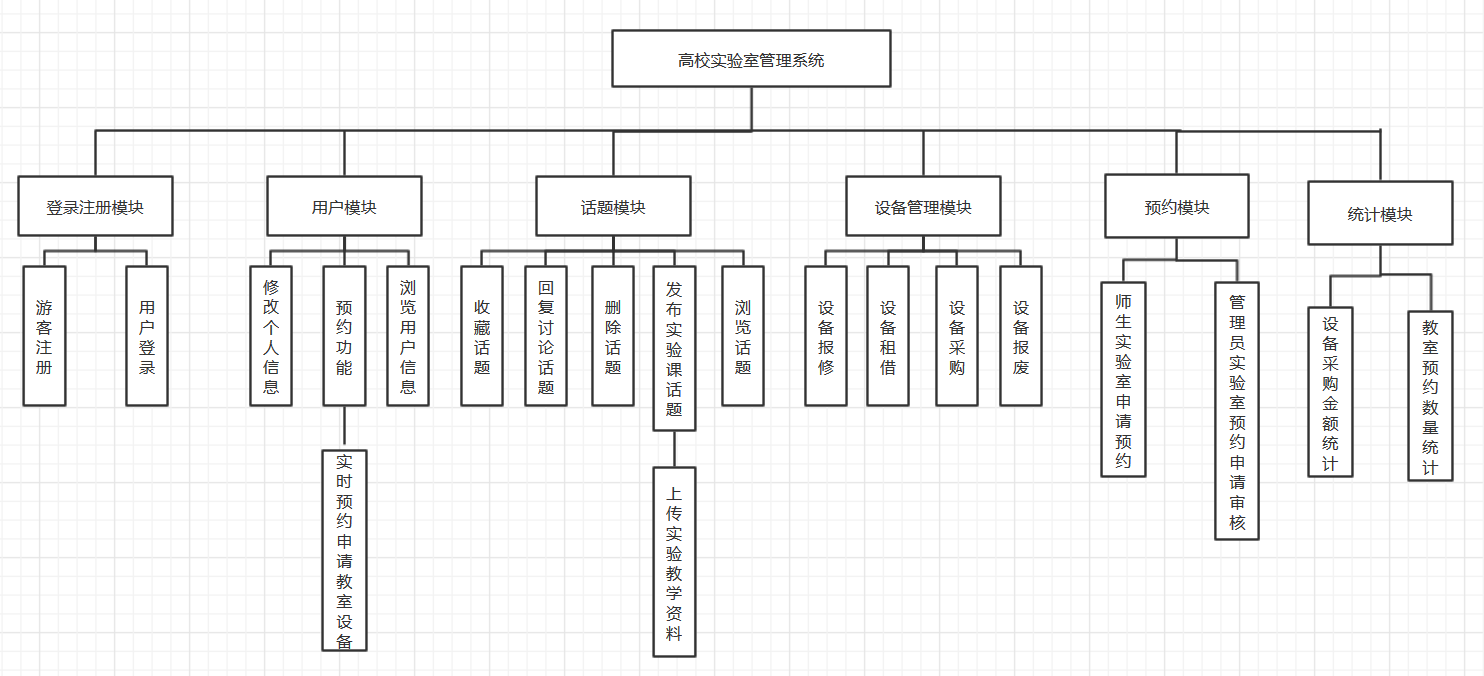


图6-1 高校实验室管理系统功能结构图

## 6.2功能模块设计

### 6.2.1用户管理

用户管理：用户新建账户后，能查看所有系统已注册用户的个人信息，并能对用户信息做增删改查等操作。

（1）功能结构设计

用户个人设置信息管理的包图如图6-2所示：

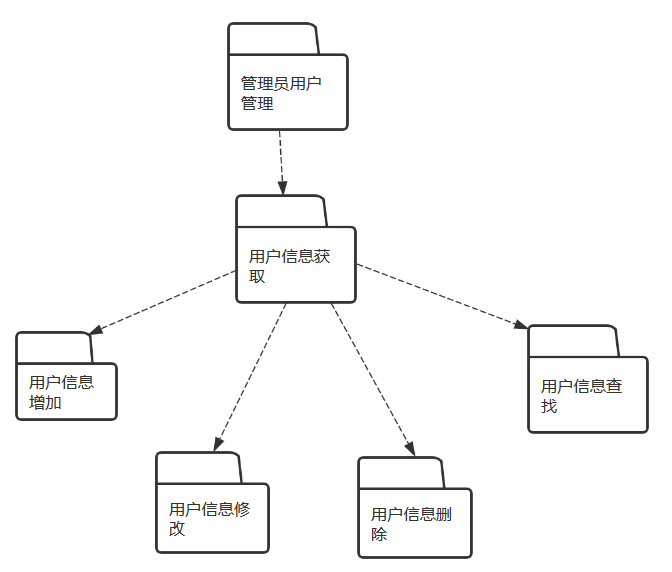


图6-2用户个人信息设置管理包图

（2） 类图设计

用户个人设置信息管理类图如图6-3所示：

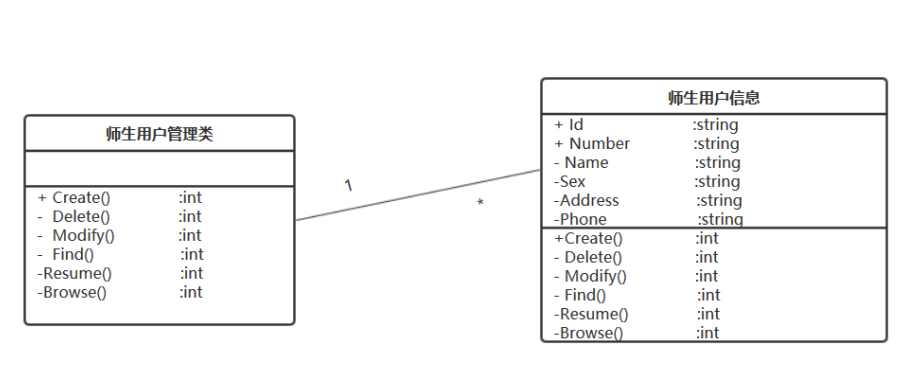


图6-3用户个人信息设置管理类图

用户个人设置信息管理类描述如表6-1所示：

表6-1用户个人信息设置管理类描述

|  |  |  |
| --- | --- | --- |
| 用户个人所设置信息类 | | |
| 类的描述：负责响应页面发出的用户信息获取请求，负责用户信息的修改 | | |
| 操作描述 | GetUsersList () | 获取用户信息 |
| AddUserCart() | 添加用户信息 |
| DeleteUserCart () | 删除用户信息 |
| SerachUser() | 查找用户信息 |
| AlterUserCart() | 修改用户信息信息 |

（3） 顺序图设计

用户个人信息设置管理的顺序图如图6-4所示：

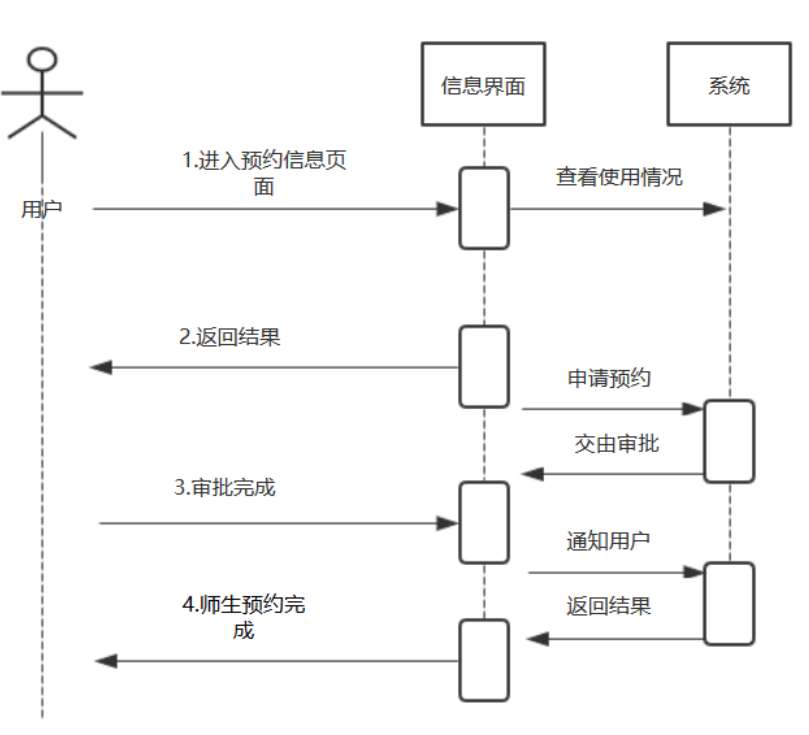


图6-4用户个人信息设置管理顺序图

（4） 核心处理流程设计

用户个人信息设置的管理活动图如图6-5所示：

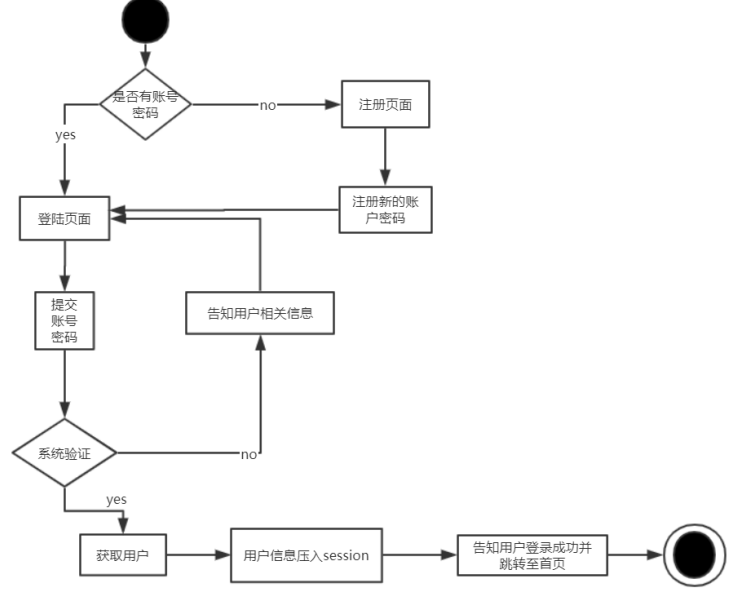


图6-5用户个人信息设置管理活动图

### 6.2.2实验室具体工作管理

实验室具体工作管理：主要分为实验室基本信息管理、实验室大纲管理、实验室预约管理、实验室设备管理四个部分。实验室基本信息和实验室大纲由管理员发布和操作，师生能预约实验室及实验设备，管理员管理实验设备信息等。

（1） 功能结构设计

实验室具体工作管理包图如图6-6所示：

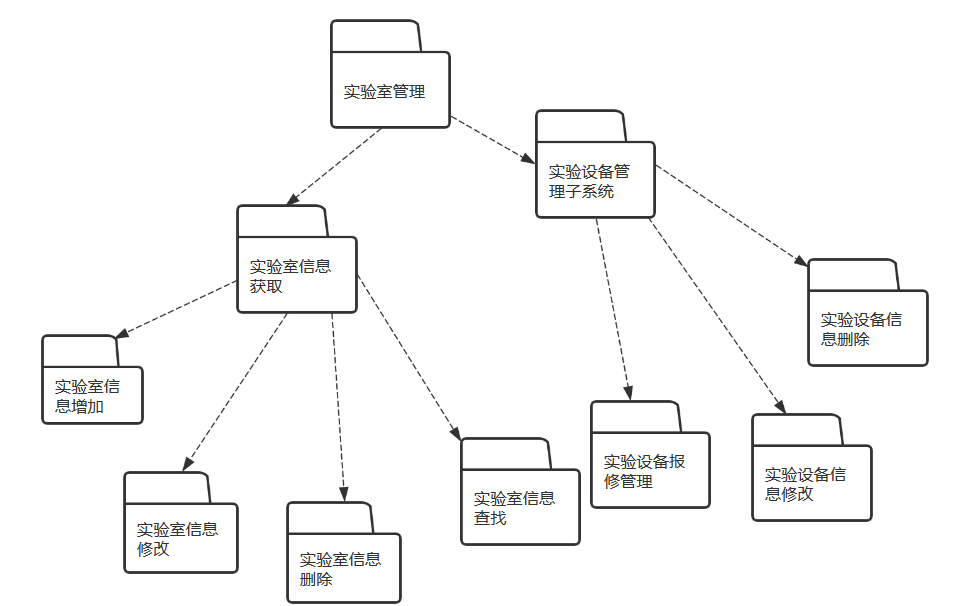


图6-6实验室具体工作管理包图

（2） 类图设计

实验室具体工作管理类图如图6-7所示：

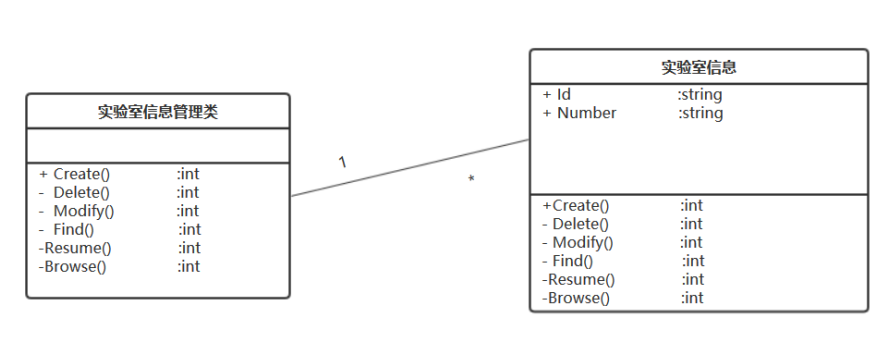


图6-7实验室具体工作管理类图

实验室具体工作管理类描述如表6-2所示：

表6-2实验室具体工作管理类描述

|  |
| --- |
| 实验室具体工作管理类 |
| 类的描述：负责管理实验室各项事务及实验设备信息 |

续表6-2

|  |  |  |
| --- | --- | --- |
| 操作描述 | GetLabList () | 获取实验室信息 |
| AddLabCart() | 添加实验室信息 |
| DeleteLabCart () | 删除实验室信息 |
| SerachLab() | 查找实验室信息 |
| AlterLab.eqCart() | 修改实验室信息 |
|  | GetLab.eqList () | 获取实验室信息 |
|  | AddLab.eqCart() | 添加实验室信息 |
|  | DeleteLab.eqCart () | 删除实验室信息 |

（3） 顺序图设计

实验室信息管理顺序图如图6-8所示：

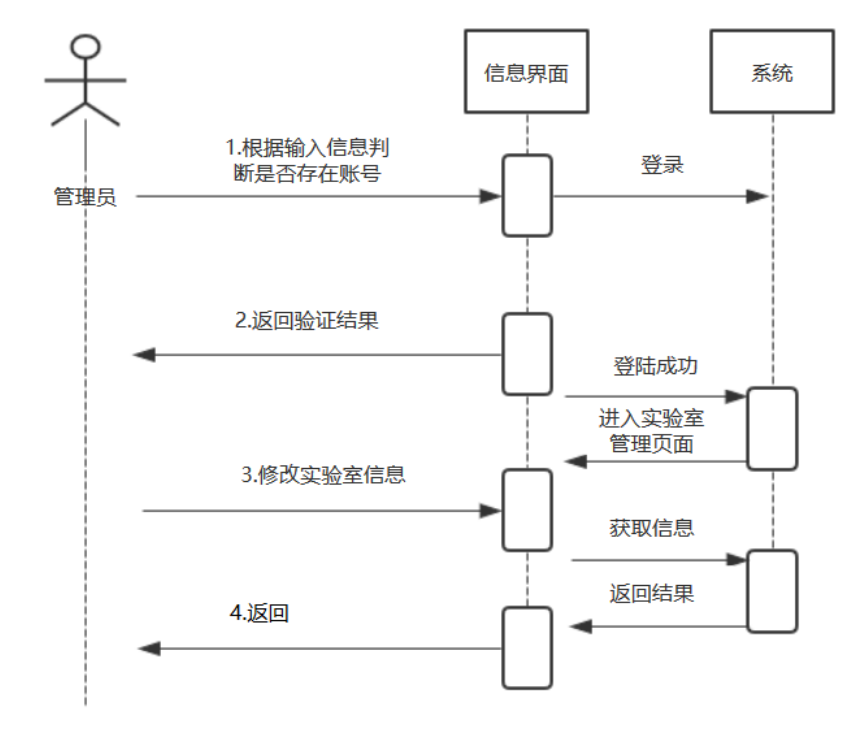


图6-8实验室信息管理顺序图

（4） 核心处理流程设计

实验室信息管理活动图如图6-9所示：

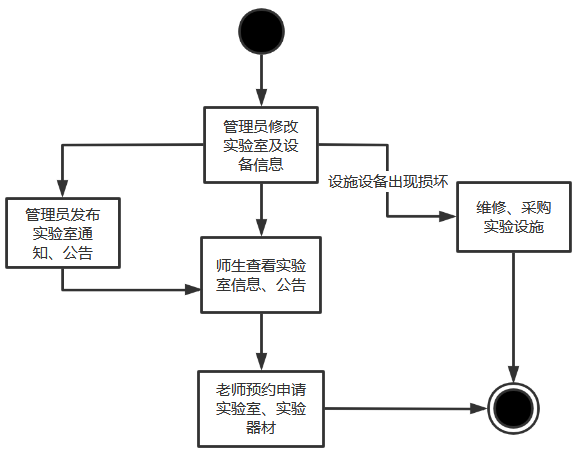


图6-9实验室信息管理活动图

### 6.2.3实验室预约管理

（1） 功能结构设计

实验室预约管理包图如图6-10所示：

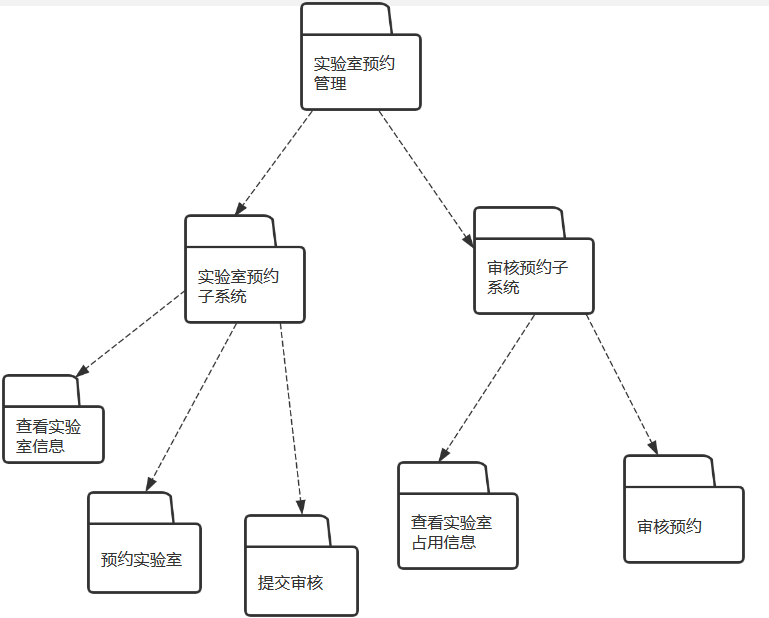


图6-10实验室预约管理包图

（2） 类图设计

实验室预约管理类图如图6-11所示：

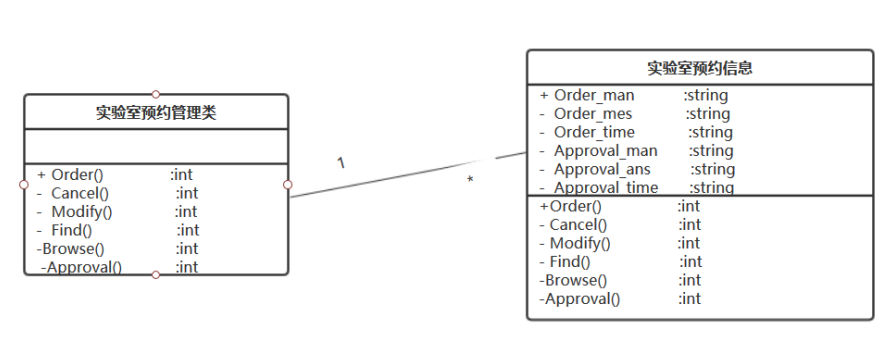


图6-11实验室预约管理类图

实验室预约管理类描述如表6-3所示：

表6-3实验室预约管理类描述

|  |  |  |
| --- | --- | --- |
| 实验室预约管理类 | | |
| 类的描述：负责用户发出预约实验室请求以及负责管理员审批预约请求 | | |
| 操作描述 | Order () | 预约申请 |
| Cancel() | 取消预约申请 |
| Modify () | 修改预约申请 |
| Examine | 审核预约信息 |
| Browse() | 浏览实验室使用信息 |

（3） 顺序图设计

实验室预约管理顺序图如图6-12所示：

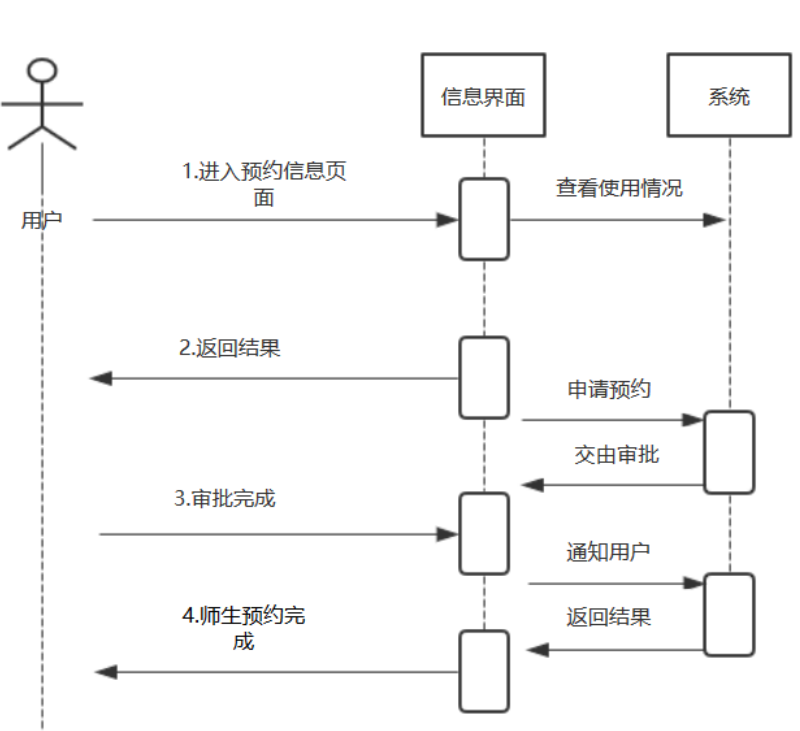


图6-12实验室预约管理顺序图

（4） 核心处理流程设计

预约实验室、实验器材流程图如图6-13所示：

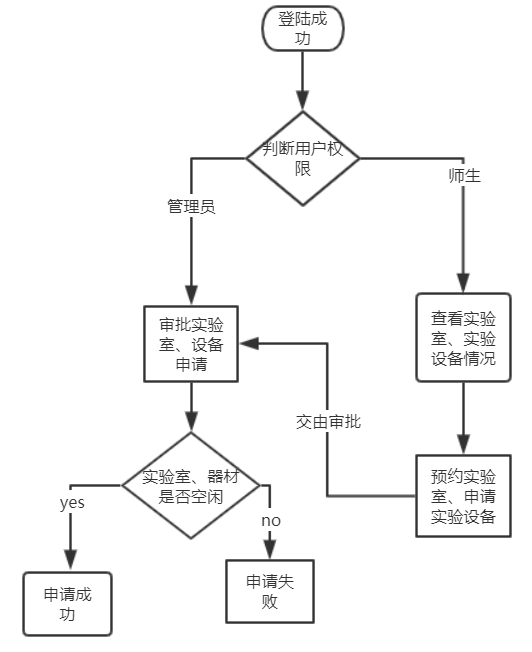


图6-13预约实验室、实验器材流程图

### 6.2.4信息统计报表管理

（1） 功能结构设计

信息统计报表管理包图如图6-14所示：

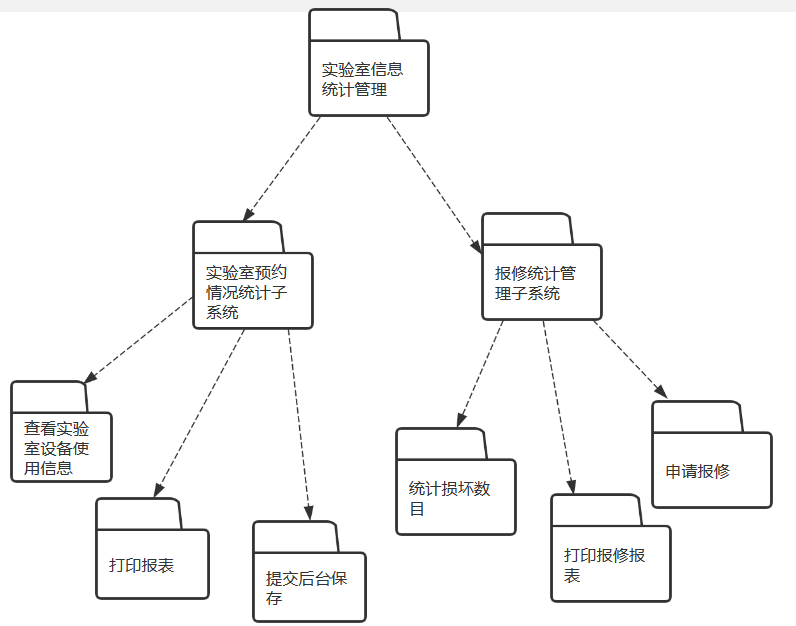


图6-14信息统计报表管理包图

（2）类图设计

信息统计报表管理类图如图6-15所示：

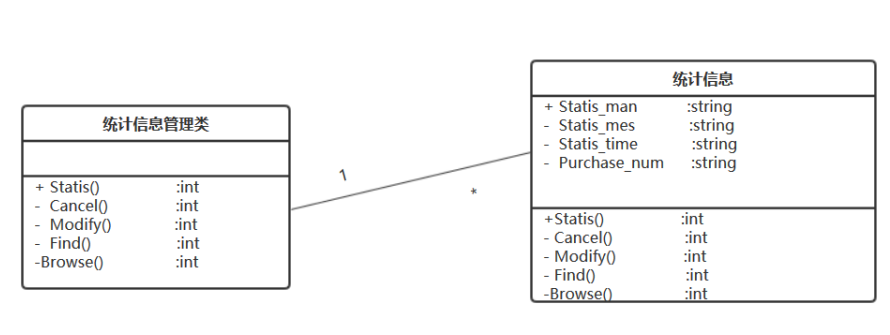


图6-15信息统计报表管理类图

信息统计报表管理类描述如表6-4所示：

表6-4信息统计报表管理类描述

|  |  |  |
| --- | --- | --- |
| 信息统计报表管理类 | | |
| 类的描述：负责统计实验室及设备使用情况，并维修、采购损坏设备 | | |
| 操作描述 | Statis () | 统计信息 |
| Browse() | 浏览使用情况 |
| Analysis () | 分析统计信息 |
| Save() | 保存统计信息 |
| Print() | 打印报表 |

（3） 顺序图设计

信息统计报表管理顺序图如图6-16所示：

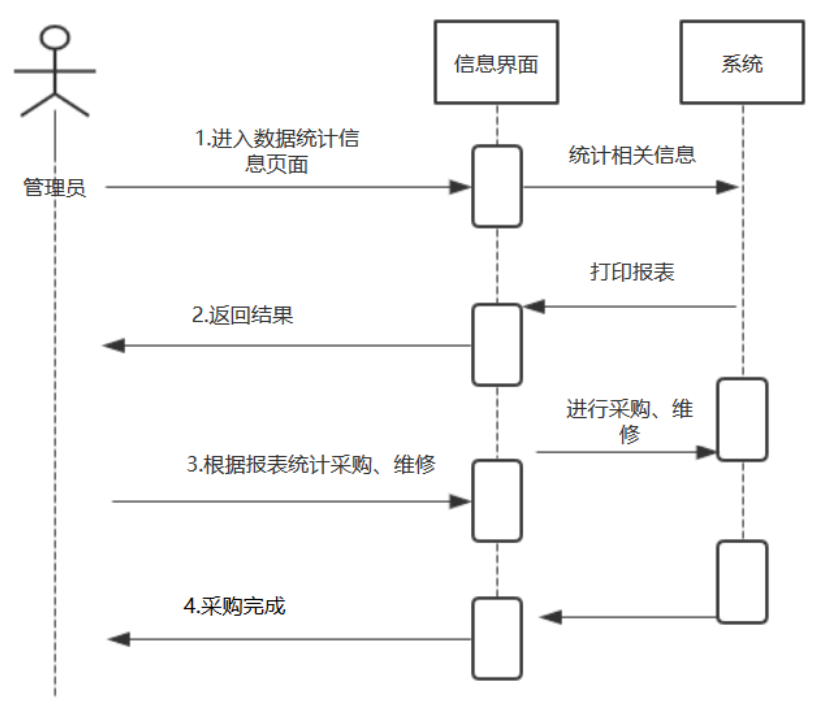


图6-16信息统计报表管理顺序图

（4） 核心处理流程设计

信息统计报表管理流程图如图6-17所示：

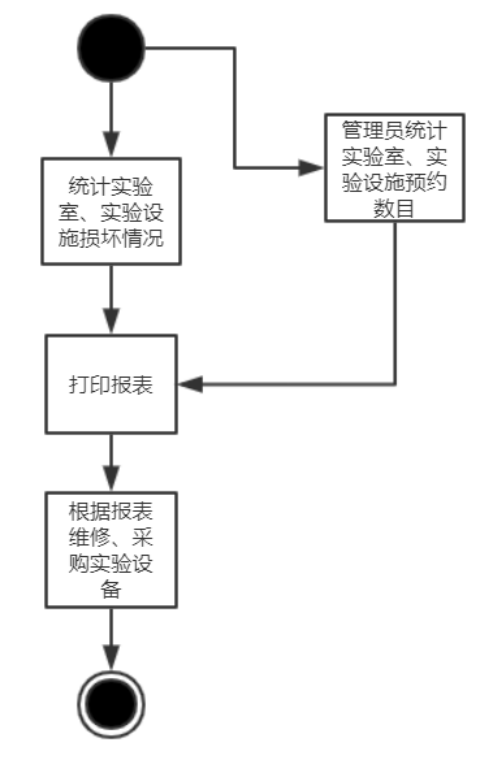


图6-17信息统计报表管理流程图

7 编码

## 7.1 代码实现与核心算法

（1）实验室model，对实验室进行定义

public class Laboratory implements Serializable {

private Integer id;

//实验室名称

private String name;

//负责人

private String fzr;

//负责人电话

private String fzrDh;

//具体地点

private String address;

//实验室备注

private String bz;

//0未删除，1已删除

private Integer isDelete;

//是否审核（0未预约 1预约）

private Integer isYy;

//预约人

private User user;

//预约时间

private Date time;

}

（2）实验室dao，与数据库完成交互代码

public class LaboratoryDaoImpl extends BaseDaoImpl<Laboratory> implements LaboratoryDao {

@Override

public Pager<Laboratory> getList(Laboratory bean) {

Map<String, Object> alias = new HashMap<String, Object>();

StringBuffer sb = new StringBuffer();

sb.append("from Laboratory where isDelete = 0");

if (bean != null) {

if (bean.getName() != null && !"".equals(bean.getName())) {

sb.append(" and name like :name");

alias.put("name", "%" + bean.getName().trim() + "%");

}

if (bean.getFzr() != null && !"".equals(bean.getFzr())) {

sb.append(" and fzr like :fzr");

alias.put("fzr", "%" + bean.getFzr().trim() + "%");

}

if (bean.getIsYy() != null && !"".equals(bean.getIsYy())) {

sb.append(" and isYy = :isYy");

alias.put("isYy", bean.getIsYy());

}

if (bean.getUser() != null && bean.getUser().getName() != null && !"".equals(bean.getUser().getName())) {

sb.append(" and user.name like :username");

alias.put("username", "%" + bean.getUser().getName() + "%");

}

}

return findByAlias(sb.toString(), alias);

}

@Override

public Laboratory findById(Integer id) {

StringBuilder sb = new StringBuilder();

sb.append("from Laboratory where isDelete = 0 and id = :id");

Query query = getSession().createQuery(sb.toString());

query.setParameter("id", id);

return (Laboratory) query.uniqueResult();

}

}

（3）实验室action：处理实验室具体业务逻辑的代码，如新增用户、查询、修改等功能

public class LaboratoryAction extends ActionSupport implements ModelDriven<Laboratory> {

private LaboratoryLogService laboratoryLogService;

private User user;

private Laboratory laboratory;

private Integer laboratoryId;

private Map<String, Object> map = new HashMap();

/\*\*

\* list

\*

\* @return

\*/

public String list(){

User user1 = UserUtils.getUser();

if (user1 == null || user1.getId() == null){

ActionContext.getContext().put("login", 1);

return SUCCESS;

}

Pager<Laboratory> pagers = null;

Role role = user1.getRole();

if (role.getEnName().equals("admin")||role.getEnName().equals("js")||role.getEnName().equals("xs")) {

pagers = laboratoryService.getList(laboratory);

ActionContext.getContext().put("pagers", pagers);

ActionContext.getContext().put("user", user1);

ActionContext.getContext().put("laboratory", laboratory);

}

return SUCCESS;

}

1. 实验室预约、取消预约代码

/\*\*

\* 预约

\*

\* @return

\*/

public void yy() throws IOException {

laboratory.setIsYy(1);

laboratory.setUser(UserUtils.getUser());

LaboratoryLog laboratoryLog = new LaboratoryLog();

laboratoryLog.setIsYy(1);

laboratoryLog.setLaboratory(laboratory);

laboratoryLog.setTime(new Date());

laboratoryLog.setUser(UserUtils.getUser());

laboratoryLogService.save(laboratoryLog);

laboratory.setLaboratoryLogId(laboratoryLog.getId());

laboratoryService.updates(laboratory);

map.put("flag", true);

map.put("url", "laboratory\_list2.do");

JsonUtils.toJson(map);

}

/\*\*

\* 取消预约

\*

\* @return

\*/

public void qx() throws IOException {

Laboratory laboratory1 = laboratoryService.findById(laboratory.getId());

LaboratoryLog laboratoryLog = laboratoryLogService.findById(laboratory1.getLaboratoryLogId());

if (laboratoryLog.getUser().getId() == UserUtils.getUser().getId()){

laboratory1.setIsYy(0);

laboratory1.setUser(null);

laboratoryService.update(laboratory1);

laboratoryLog.setIsYy(0);

laboratoryLog.setEndTime(new Date());

laboratoryLog.setQx(UserUtils.getUser());

laboratoryLogService.updates(laboratoryLog);

map.put("flag", true);

map.put("url", "laboratory\_list2.do");

JsonUtils.toJson(map);

} else {

map.put("flag", false);

map.put("url", "laboratory\_list2.do");

JsonUtils.toJson(map);

}

}

## 7.2 代码优化分析

使用GET方法替换POST方法来完成请求

浏览器通过XMLHttpRequest对象来进行通信，浏览器使用的POST请求会发送两个TCP数据包传送数据，浏览器先发送header,服务器响应后，浏览器再发送数据，而使用GET方式的请求时，浏览器会将header和数据一起发送，因此用GET替换POST能使效率提高。

$.ajax({

type:'GET',

url:'proinfo',

data:data,

success:function(result){

//返回成功后进行响应操作

document.getElementById('photo\_id').innerText = result['photo\_id'];

document.getElementById('user\_id').innerText = result['user\_id'];

document.getElementById('bx\_time').innerText = result['bx\_time'];

document.getElementById('end\_time').innerText = result['end\_time];

document.getElementById('laboratory\_id').innerText = result['laboratory\_id'];

}

});

8 测试

## 8.1 测试方案设计

### 8.1.1 测试策略

在对该系统的功能进行初步实现之后，应开始进行对系统的测试工作，找出系统中因系统开发人员疏忽而导致存在的Bug，通过测试环节，用提交的Bug报告来为之后的软件改进提供参考标准，并能够在以后系统改进之中找到凭据。

### 8.1.2 测试进度安排

测试进度安排如表8-1所示：

表8-1 测试进度安排表

|  |  |
| --- | --- |
| **测试进度** | **测试任务** |
| 单元测试阶段 | 对最小的程序模块进行测试 |
| 集成测试阶段 | 将软件单元组合为单元模块或系统，再进行测试 |
| 系统测试阶段 | 对照需求说明文档，检测软件产品与系统设计的差异 |
| 验收测试阶段 | 以交付的用户为主体进行的测试 |

### 8.1.3 测试资源

（1）Windows10操作系统

（2）Eclipse开发环境

（3）JDK1.7

（4）Tomcat8.0服务器

（5）谷歌浏览器

8.1.4 关键测试点

（1）管理员能否对所有用户信息进行修改操作。

（2）师生是否只能对自己的用户信息进行修改操作。

（3）登录系统时若用户输错账号或密码，系统是否提示账号或密码错误。

（4）文件是否能正确下载。

## 8.2 测试用例构建

### 8.2.1 测试用例编写约定

在测试用例编写过程中，测试员应尽可能将各测试用例包含各个业务点，并对多个

不同类型的数据进行并发测试，测试的数据要具有一定的代表性。测试用例编写时要简洁明了，方便客户和测试人员阅读。

### **8.2.2** 测试用例设计

本系统主要测试用例设计如表8-2所示：

表8-2系统测试用例设计

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 测试项 | 描述/输入/操作 | 期望结果 |
| 01 | 用户注册 | 若该用户注册信息与已注册用户保持一致 | 提示该用户已存在 |
| 02 | 用户登录 | 输入相对应的用户账号信息 | 成功跳转至登录成功的界面，失败则提示用户账号或密码错误 |
| 03 | 管理员用户删除其他用户信息 | 后台查看用户信息，可点击删除对用户信息进行删除操作 | 该用户信息被删除，无法凭该信息登录系统 |
| 04 | 管理员审核用户注册 | 管理员对其他用户提交的注册信息进行审核 | 审核通过，则用户可以登录，审核未通过，则用户无法登陆 |
| 05 | 管理员管理实验室信息 | 管理员对实验室信息进行修改等操作 | 信息修改后的结果能在实验室信息界面显示 |
| 06 | 师生用户管理用户信息 | 师生用户对自己的用户信息进行修改等操作 | 后台信息同步修改 |
| 07 | 用户查询实验室信息 | 后台根据关键字对实验室信息进行查询 | 输出对应的查询结果 |

### 8.2.3 关键测试用例

（1）用户的登录测试用例如表8-3所示：

表8-3 用户登录模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试目的** | | 测试用户登录模块的功能 | |
| **预置条件** | | 用户输入正确的信息登录系统 | |
| **序号** | **测试步骤** | | **预期结果** |
| 1 | 用户输入对应的用户名和密码之后点击登录按钮 | | 跳转系统主页面 |
| 2 | 用户没有输入对应的用户名和密码之后点击登录按钮 | | 系统提示用户账号或密码错误 |

续表8-3

|  |  |
| --- | --- |
| **测试状态** | 上述操作均达到“测试”要求 |
| **测试结果** | 经过测试，登录模块能够对用户的操作进行正确处理，也可以对错误的操作进行提示，达到了预期目标。 |

（2）管理员审核用户注册信息测试用例如下表8-4所示：

表8-4 审核注册信息模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试目的** | | 测试管理员审核用户在注册信息模块中各项功能是否完善 | |
| **预置条件** | | 游客或用户填写相关注册信息，待管理员进行审核 | |
| **序号** | **测试步骤** | | **预期结果** |
| 1 | 管理员对用户注册信息进行审核，且审核通过 | | 用户信息在后台保存，注册成功 |
| 2 | 管理员对不合理的注册信息进行删除 | | 用户信息在后台同步删除，注册失败 |
| **测试状态** | | 上述操作均达到“测试”要求 | |
| **测试结果** | | 经过测试，该模块能够对管理员的操作进行正确处理，达到了预期目标。 | |

### 8.2.4 测试用例维护

1. 后期根据用户的反馈，及时完善系统功能。
2. 根据用户后期新增的功能需求，可以增加新的测试用例。
3. 多余的测试用例后期可以删除或修改。

9 总结与展望

## 9.1 设计工作总结

本系统是高校实验室管理系统，在拿到论文的题目时，我就想到了校园里大学生经常使用的图书馆预约系统。在开发系统之前，先要去做项目的需求分析。虽然在大学期间也使用过学校的类似自习室预约系统，但不是很了解实验室管理系统，指导老师实地带我去企业调研，查询企业里的相关资料，我就对这个项目有了大概的了解，系统用户应该分为两大类，一类是师生用户，另一类是管理员用户，同时系统还应该具有用户信息填写注册、预约实验室及设备、用户讨论交流、设施使用统计报表、文件浏览下载等主要功能。通过前期的实地调研和之后在网上查询的相关资料，对项目各部分应该具有哪些功能点有所了解，有了系统整体的雏形，整理了类图、顺序图、流程图等，完成了需求分析说明书和系统设计说明书的设计。在代码编写阶段，通过网上资料的查询和指导老师的帮助解决了在一些小的功能模块上出现的问题。在测试阶段，先列出关键测试点，再根据测试点来设计测试用例，同时将不同类型的大量数据都放进主要功能点来测试是否符合预期要求。

## 9.2 未来工作展望

虽然本系统的开发暂时告一段落，但是系统距离一个成熟优秀的实验室管理系统还有较大的差距，还需要后续的一些优化处理，如在用户使用系统，并不断提出问题时，要对一些测试点进行修改，增加新的测试用例来完善系统功能。系统具体优化工作如下所示：

1. 优化界面，使界面更美观：在设计系统之初，只是设计了一个简单的登录注册界面，之后可以在前台创建一个更富有平台特色的，能对高校实验室及设备做一个详细介绍的页面，这样就能突出系统的功能，吸引用户的使用。
2. 添加系统功能：系统面向师生的功能主要是预约实验室及设备，查询设备信息等功能，可以在后期添加一个意见收集界面，其功能是能够帮助师生及时反馈信息，如实验设备损坏等情况。
3. 完善系统功能：在预约界面的设计中，不够人性化，例如师生在点击某个实验室时不能看到实验室的具体信息，要点击实验室列表，才能看到相关信息。之后在完善系统功能时，用户在点击某个实验室时，可以显示其相关信息。

谢 辞

参考文献

[1] 杨俊杰.新时期高校实验室仪器设备管理问题与策略[M].中国教育技术装备，2019.

[2] 真国诚.基于计算机软件工程的现代化技术[J].电子技术与软件工程，2017,（22）：152-153.

[3] 王小琼.软件工程点团队项目驱动教学研究[J].科技视界，2017,（25）：50+85.

[4] 伍春香，刘斌，李小红.基于能力培养的软件工程教学实践[J].计算机教育，2017,（08）：38-42.

[5] 郑大鹏，王琳，路良刚.软件工程综合实训课程融合初探[J].计算机教育，2017,（07）：135-139.

[6] 解新星.软件工程数据挖掘研究进展[J].通讯世界，2017,（14）：71-72.

[7] 韩寅聪.关于高校实验室设计管理的思考[M]. 基层设计，2019.

[8] 刘琴.软件工程中程序设计方法的比较[J].计算机时代，2017,（07）：53-55.

[9] 段翰林.计算机软件工程的自动化管理[J].科技资讯，2017,15（22）：12-13.

[10] 崔萌萌，安强.项目管理对软件工程开发的必要性研究[J].科技风，2017,（17）：65.

[11] 黄伟国.基于CDIO的软件工程课程教学改革[J].信息与电脑（理论版），2017,（13）：225-226+229.

[12] 张乾燕，陈林，秦川.翻转课堂引领软件工程专业课程教学改革--以宿迁学院为例[J].大学教育，2017,（08）：50-52.

[13] 刘鑫.大数据时代独立学院软件工程专业人才培养模式改革研究[J].通讯世界，2017,（18）：263-264.

[14] 徐完平.大数据背景下软件工程专业教学模式的思考[J].福建电脑，2017,33（09）：169+41.

[15] 张平.互联网+背景下地方本科院校软件工程专业教学改革研究[J].经营管理者，2017,（24）：440.

[16] 刘爱萍.高校实验室安全管理存在的问题及解决方法[M]. 2019,（14）：60-61。

[17] 黄一磊.电子信息软件工程的发展[J].中国新通信，2017,19（17）：35.

[18] 畅鹏.形式概念分析在软件工程中的应用[J].通讯世界，2017,（19）：273-274.

[19] 尚竞成.计算机软件工程的技术及管理措施分析[J].通讯世界，2017,（19）：96.

[20] 刘春爽，王志海.计算机软件工程管理与应用解析[J].通讯世界，2017,（19）：26-27.

[21] 王润芳.浅析计算机软件工程项目自动化管理分析[J].电子世界，2017,（16）：53.

[22] 田钟晓，虞翔.面向对象的软件工程中软件需求分析方法[J].电子技术与软件工程，2017,（16）：48.

[23] 许琴.论软件工程化过程中的质量管理[J].电子技术与软件工程，2017,（16）：59-60.

[24] 宋杉.计算机软件工程项目的自动化管理[J].电子技术与软件工程，2017,（16）：69.

[25] Enck, William, Ongtang, Machigar, McDaniel, Patrick. Understanding Android Security[J]. IEEE security & privacy, 2009, 7(1): 50-57.

[26] Hohzaki R, Maehara H. A single-shot game of multi-period inspection[J]. European Journal of Operational Research, 2010, 207(3): 1410-1418.

[27]Marcelo M. Eler,Andre T. Endo,Vinicius H.S. Durelli. An Empirical Study to Quantify the Characteristics of Java Programs that May Influence Symbolic Execution from a Unit Testing Perspective[J]. The Journal of Systems &amp; Software,2016.

[28]Kebo Zhang,Hailing Xiong. A new version of code Java for 3D simulation of the CCA model[J]. Computer Physics Communications,2016.

[29]Zeinab Iranmanesh,Mehran S. Fallah. Specification and Static Enforcement of Scheduler-Independent Noninterference in a Middleweight Java[J]. Computer Languages, Systems &amp; Structures,2016.

[30]Kebo Zhang,Junsen Zuo,Yifeng Dou,Chao Li,Hailing Xiong. Version 3.0 of code Java for 3D simulation of the CCA model[J]. Computer Physics Communications,2016.

附录A 外文翻译—原文部分

Data Persistence on Curriculum Management System based on MyBatis

原文出处-2nd International Conference on Economics, Social Science, Arts, Education and Management Engineering (ESSAEME 2016)

**(1)Abstract**

Course of information management and development of curriculum management systemis to reduce the educational task complexity and the effective ways to improve the efficiency of the educational administration work. Data persistence is an important work of software development, this article is based on MyBatis technology research and provide technical support for the system development. First of all, research by the interface layer, data processing layer and foundation support layer of MyBatis function architecture; Then, based on the Sybase database management system, curriculum information table, for example, for data storage logic structure design; Finally, to design the persistence code, including total configuration files, object mapping files and entity class object. The actual application, attention should be paid to give full play to the advantage of the MyBatis, improve the overall performance of the system and the development efficiency.

**(2)Introduction**

Course management is an important educational administration work, involves the teaching plans, course arrangement and performance management aspects of work, is one of the most complex work of educational administration system. Was carried out on the course information management, development of curriculum management system, is to conduct standardized management, to reduce the educational task complexity and the effective ways to improve the efficiency of the educational administration work.

Data persistence is a process to convert data in memory model for storage model and converts the storage model in memory, to achieve data in memory and the exchange of data in the database, is an important part of the software system development. Java development the persistent technology is the most commonly used Hibernate and iBatis. IBatis is an Apache open source project, in 2010 by the "Apache software foundation" migrated to Google code ", and changed its name to "MyBatis" . MyBatis is to support the common SQL queries, stored procedures and advanced mapping of excellent persistence layer framework; Eliminates almost all JDBC code, parameter of manual setup and result set retrieval; Using a simple XML or annotations for configuration and the original mapping, the interface and Java POJOs mapping database records. The programmer can follow one's indiscriminately use object to manipulate the database programming thinking, improve software development efficiency and maintainability. In this paper, based on the study of MyBatis, provide technical support to the development of the software.

1. **Logical Structure Design on Data Store**

Logical Structure Design on Data Store associated with the database management system of concrete structure, and choosing the Sybase as course management system. Sybase database is launched in high-performance products of Sybase company, and it is based on client/server architecture, the application is run on the multiple machines. Advantage is to support sharing resources and balance load between multiple devices, allowing for multiple host environment, make full use of the enterprise has some various systems.

Sybase database is mainly composed of three parts: one is that for database management and maintenance of "Sybase SQL Server", is a programmable database management system, is the core of the Sybase product software, to play the data management, high-speed buffer management, transaction management role; Second, support the establishment of the database application system and the development of "Sybase SQL Toolset, ISQL interact with SQL Server is a SQL syntax analyzer, DWB is convenient and practical database management tools, APT is mainly engaged in the development of the practical application environment; Three is that other manufacturers under heterogeneous environment application software with any type of data connection "the Sybase Open Client/Open Server".

1. **Functional Architecture on MyBatis**

Mybatis function architecture is divided into three layers, namely, interface layer, data processing layer and foundation support layer, described the following :

The Interface layer: provide for the use of external Interface API, developers through the API to manipulate the database. The interface layer receives the call request is called data processing layer to complete specific data processing. SqlSessionFactory and SqlSession is the core of the MyBatis interface layer, SqlSessionFactory object by the SqlSessionFactoryBuilder is created, the main function is to create the SqlSession objects; SqlSession object's main function is to complete a database access and mapping as a result, because is not thread safe, so the SqlSession objects need to limit the scope of the method.

The data processing layer: It is responsible for the specific SQL search, parse SQL, SQL execution and the execution result map processing, and the main purpose is according to the call request to complete a database operation., MyBatis.XML analyses SqlMapConfig.xml and SqlMap.xml Two configuration files when application startup. The SqlMapConfig.xml to complete the analysis in the XMLConfigBuilder class; SqlMapConfig.xml complete parsing in the XMLConfigBuilder class, the involved PreparedMapping, ResultMapping, LanguageDriver, Discriminator, caching, and automatic mapping and so on a series of object structure. Executor is the core in MyBatis with the full process of database operations, need MappedStatement, parameter and result Handler instance objects. Transaction commit/rollback, entrusted to the Transaction object to complete; Cache, create CacheKey ()/is Cached (); Lazy loading, defer load (); Close, close (), mainly is the transaction rollback/closed.

The Foundation support layer: It is responsible for the most basic functions, including the

connection management, transaction management, configuration, load and cache handling, the Shared component is extracted as the most basic components, provide the Foundation of support for the data processing layer. MyBatis provides a simple data source/connection pool under the org. apache. ibatis.datasourse, implementation class is PooledDataSource, contains the maximum number of connections, the largest number of idle connections, and connection is not enough to take out the longest time waiting time, etc.; MyBatis for transaction processing is relatively simple, TransactionIsolationLevel defines several isolation level, doesn't support nested transaction, application development, entrust the Spring to handle affairs; MyBatis includes two kinds of cache. Session and the Statement is scope level, default is Session, BaseExecutor according to MappedStatement Id, SQL, the value and CacheKey rowBound to construction, and maintenance by using the localCache BaseExccutor cache. Global level 2 cache could be come true through CacheExecutor, and entrust TransactionalCacheManager save or access to the cache.

1. **Code Design on Data Persistence**

MyBatis persistence design is mainly the following three configuration file:

Total MyBatis configuration file: the configuration.xml. The configuration file defines the database connection information and global configuration information. Using the configuration file has powerful function, convenient operation, good format check mechanism, automatic file loading, etc.

The object mapping file: courseMapper.xml. Entity class mapping file is used to finish to the operation of the database objects, including to perform a variety of SQL statements, stored procedure, the input parameter mapping, mapping, caching mechanisms, and can through several relatively complex configuration make the relationship between objects and lazy loading, realized the data storage logic from the upper logic code.

Entity class object file: Course.Java. Entity class object is used to complete the operation of

the database, in between the business logic and database resources, improve the independence of the class. Database is required in the operation of each table needs to write the DAO implementation class. The DAO and the business logic layer transfer is an entity object, is a mapping of database tables in the program. For each attribute the get and set methods as access interface, improves the independence of the class. Due to the many properties CourseComprehensiveInformationTable, previously only 10 attributes.

**（6）Conclusion**

MyBatis has the following advantages: one is that simple, through the document and source code, can be more fully master the design ideas and implementation; Second, practical, and provide data mapping function, access to the underlying data encapsulation and DAO framework, it is easier to develop and deploy DAL layer; The third is, flexible, basic can be achieved through SQL, do not use the data access framework to implement all functions; The fourth is, fully functional, through configuration object mapping relationship between the data access layer to solve the

problem, such as DAO support, and encapsulates the Ado.NET in the DAO plane, I got and DataMapper; The fifth is, maintainability, by providing DAL layer, business logic and data access logic separation, the system design more clearly, easier maintenance, easier to unit test. SQL improves the maintainability and code separation. This paper research the actual application, should pay attention to and the combination of these advantages, improve the overall performance of the system and the development efficiency.

附录B 外文翻译—译文部分

**基于 MyBatis 的课程管理系统中的数据持久性**

1. **内容摘要**

课程信息管理管理和课程管理系统开发是为了降低教育任务的复杂性和提高教务工作效率的有效途径。数据持久化是软件开发的一项重要工作，本文基于MyBatis技术研究，为系统开发提供技术支持。首先对MyBatis功能架构的接口层，数据处理层和基础支持层进行研究，然后，基于Sybase数据库管理系统，课程信息表，例如，进行数据存储逻辑结构设计；最后设计持久化代码，包括总配置文件，对象映射文件和实体类对象，在应用中充分发挥MyBatis的优势，提高系统的整体性能和开发效率。

1. **导言**

课程管理是一项重要的教务管理工作，涉及到教学计划、课程安排和绩效管理等方面的工作，是教务系统最复杂的工作之一。是在课程信息管理上进行的，制定课程管理制度进行规范化管理，降低教育任务复杂性和提高教务工作效率的有效途径。

数据持久化是将内存模型中的数据转换为存储模型并转换为内存中的存储模型，实现内存中的数据和数据库中的数据交互的过程，是软件系统开发的重要组成部分。Java开发最常用的持久化技术是Hibernate和iBatis.IBatis 是一个 Apache 开源项目，2010 年由“Apache 软件基金会”迁移到谷歌，并改名为“MyBatis”。MyBatis是支持优秀持久层框架的常见 SQL 查询，存储过程和高级映射；消除几乎所有的 JDBC 代码，手动设置的参数和结果集检索；使用简单的 XML 或注释进行配置和实体映射，接口和 Java POJO 映射数据库记录, 程序员可以跟随一个人直接地使用对象来操纵数据库编程思维，提高软件开发效率和可维护性。本文在对 MyBatis 的基础上进行研究，为软件的开发提供技术支持。

**（3）数据存储的逻辑结构设计**

具体结构数据库管理系统关联的数据库逻辑结构设计，选择 Sybase 作为课程管理系统。 sybase 数据库以高 p 方式启动 Sybase 公司的性能产品，基于客户端/服务器架构，应用程序在多台机器上运行。 优势在于支持资源共享和平衡负载 多个设备，允许多个主机环境，充分利用企业有一些不同的系统。

Sybase 数据库主要由三部分组成：一是用于数据库管理和维护的“Sybase SQLServer”，是一个可编程的数据库管理系统，是 Sybase 产品软件的核心，发挥数据管理、高速缓冲区管理、事务管理的作用；二是支持建立数据库应用系统和开发 “Sybase SQL 工具集，ISQL 与 SQLServer 交互是一个 SQL 语法分析器，DWB 是方便实用的数据库管理工具，APT 主要从事实际应用环境的开发；三是其他厂商在异构环境下应用软件与任何类型的数据连接“Sybase 开放客户端/开放服务器”。

**（4）MyBatis上的功能架构**

将Mybatis 功能架构分为三层，即接口层，数据处理层和基础支撑层，说明如下：

接口层：提供使用外部接口 API，开发人员通过API操作数据库，接口层接收调用请求称为数据处理层来完成具体的数据处理。SqlSessionFactory和SqlSessionFactory是MyBatis接口层的核心，由SqlSessionFactoryBuilder创建SqlSessionFactory对象，主要功能是创建SqlSession对象；SqlSession对象的主要功能是完成一个数据库访问和映射

的结果，因为不是线程安全的，所以 SqlSession 对象需要限制方法的范围。

数据处理层：负责具体的 SQL 搜索、解析、SQL 执行和执行结果映射处理，主要目的是根据调用请求来完成数据库操作，当应用程序启动时，MyBatis.xml 分析 SqlMapConfig.xml 和 SqlMap.xml 两个配置文件。在 XMLConfigBuild 中完成 SqlMapConfig.xml 的分析。SqlMapConfig.xml 在 XMLConfigBuilder 类中完成解析，所涉及的 PreparedMapping，ResultMapping，语言驱动程序，判别器，缓存和自动映射等一系列物体结构中，执行程序是 MyBatis 中的核心，具有数据库操作的全过程，需要映射语句、参数和结果 Handler 实例对象。 事务提交或回滚托付事务对象来完成；缓存，创建 CacheKey()/is Cached()；懒加载, 定义 load ()和 close ()等方法，最主要的是事务的回滚和关闭。

基础支持层：负责最基本的功能，包括连接管理、事务管理、配置、负载和缓存处理，提取共享组件作为最基本的组件，为数据处理层提供支持的基础。 Mybatis 下提供了一个简单的数据源/连接池 org.apache.ibatis.datasourse，实现类为 PooledDataSource，包括最大连接数，最大空闲连接数，连接不够取出最长等待时间等。MyBatis 用于事务处理比较简单，事务隔离级别定义了几个隔离级别，不支持嵌套事务，应用开发，委托 Spring 处理事务；MyBatis 包括两种缓存、会话和语句是作用域级，默认是会话，根据 Mapped 语句 ID、SQL 执行器，值和缓存键行绑定到构造，并使用本地缓存基 Excutor 缓存进行维护。 全局二级缓存可以通过缓存执行器实现，并委托事务缓存管理器保存或访问缓存。

**（5）关于数据持久性的代码设计**

MyBatis 持久化设计主要是以下三个配置文件：

MyBatis 总配置文件：configuration.xml。配置文件定义数据库连接信息和全局配置信息，使用配置文件功能强大，操作方便，有良好的格式检查机制，文件自动加载等。

对象映射文件：courseMapper.xml。实体类映射文件用于完成对数据库对象的操作，包括执行各种 SQL 语句、存储过程、输入参数映射、映射、缓存机制，并可以通过几种相对复杂的配置使对象与懒加载之间的关系，实现了从上层逻辑代码的数据存储逻辑。

实体类对象文件：Course.Java。实体类对象用于完成数据库在业务逻辑和数据库资源之间的操作，提高了类的独立性。 数据库是需要在每个表的操作中需要编写DAO 实现类，而 DAO 和业务逻辑层的传输是一个实体对象，是程序中数据库表的映射，对于每个属性，get 和 set 方法作为访问接口，提高了类的独立性。 由于许多属性，课程综合信息表，以前只有 10 个属性。

1. **总结**

MyBatis 有以下优点：一是简单，通过文档和源代码就可以更全面地掌握设计思想和实现；二是实用，它提供了数据映射功能、访问底层数据封装和 DAO 框架，更容易开发和部署 DAL 层；三是灵活，基本可以通过 SQL 实现，不使用数据访问框架实现所有功能；四是功能齐全，通过数据访问层之间的配置对象映射关系来解决 DAO 支持等问题，封装 Ado.NET 在 DAO 平面上，得到了数据管理；第五是可维护性，通过提供的 DAL 层、业务逻辑和数据访问逻辑分离，系统设计更清晰、维护更容易、单元测试更容易。 SQL 提高了可维护性和代码分离性. 本文研究了实际应用，应注意并结合这些优点，提高系统的整体性能。

附录C 软件使用说明书

用户进入系统，成功登陆页面，用户分为两类，一类是管理员用户，一类是师生用户。

管理员登录后左侧管理区域有通知管理、用户管理、实验室管理、实验室课程与设备管理、设备借用管理、设备维修管理、报表统计等功能。

管理员用户登录界面如图10-1所示：



图10-1 管理员用户登录界面

师生用户登录后有通知管理、用户管理、设备借用管理、实验室课程与设备管理等功能。

师生用户登录界面如图10-2所示：



图10-2 师生用户登录界面

管理员用户在登陆后可根据自己的不同需求对系统进行操作，例如：在用户管理页面能查看所有注册用户列表，显示其权限名称、登录账号、真实姓名、学工号等相关信息。

管理员操作界面如图10-3所示：



图10-3 管理员操作界面

附录D 主要源代码

实验室信息：

public class Laboratory implements Serializable {

private Integer id;

//实验室名称

private String name;

//负责人

private String fzr;

//负责人电话

private String fzrDh;

//具体地点

private String address;

//实验室备注

private String bz;

//0未删除，1已删除

private Integer isDelete;

//是否审核（0未预约 1预约）

private Integer isYy;

//预约人

private User user;

//预约时间

private Date time;

private Integer laboratoryLogId;

@Id

@GeneratedValue

public Integer getId() {

return id;

}

@Column(name="isDelete", nullable=false, columnDefinition="int default 0", updatable = true, insertable = false)

public Integer getIsDelete() {

return isDelete;

}

public void setIsDelete(Integer isDelete) {

this.isDelete = isDelete;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

实验室dao，实现与数据库的信息交互代码：LaboratoryDaoIpml.java

public class LaboratoryDaoImpl extends BaseDaoImpl<Laboratory> implements LaboratoryDao {

@Override

public Pager<Laboratory> getList(Laboratory bean) {

Map<String, Object> alias = new HashMap<String, Object>();

StringBuffer sb = new StringBuffer();

sb.append("from Laboratory where isDelete = 0");

if (bean != null) {

if (bean.getName() != null && !"".equals(bean.getName())) {

sb.append(" and name like :name");

alias.put("name", "%" + bean.getName().trim() + "%");

}

if (bean.getFzr() != null && !"".equals(bean.getFzr())) {

sb.append(" and fzr like :fzr");

alias.put("fzr", "%" + bean.getFzr().trim() + "%");

}

if (bean.getIsYy() != null && !"".equals(bean.getIsYy())) {

sb.append(" and isYy = :isYy");

alias.put("isYy", bean.getIsYy());

}

if (bean.getUser() != null && bean.getUser().getName() != null && !"".equals(bean.getUser().getName())) {

sb.append(" and user.name like :username");

alias.put("username", "%" + bean.getUser().getName() + "%");

}

}

return findByAlias(sb.toString(), alias);

}

@Override

public Laboratory findById(Integer id) {

StringBuilder sb = new StringBuilder();

sb.append("from Laboratory where isDelete = 0 and id = :id");

Query query = getSession().createQuery(sb.toString());

query.setParameter("id", id);

return (Laboratory) query.uniqueResult();

}

}

实验室action：处理具体业务逻辑的代码LaboratoryAction.java

public class LaboratoryAction extends ActionSupport implements ModelDriven<Laboratory> {

@Autowired

private UserService userService;

@Autowired

private RoleService roleService;

@Autowired

private LaboratoryService laboratoryService;

@Autowired

private LaboratoryLogService laboratoryLogService;

private User user;

private Laboratory laboratory;

private Integer laboratoryId;

private Map<String, Object> map = new HashMap();

/\*\*

\* list

\*

\* @return

\*/

/\*\*

\* 预约处理

\*

\* @return

\*/

public void yy() throws IOException {

laboratory.setIsYy(1);

laboratory.setUser(UserUtils.getUser());

LaboratoryLog laboratoryLog = new LaboratoryLog();

laboratoryLog.setIsYy(1);

laboratoryLog.setLaboratory(laboratory);

laboratoryLog.setTime(new Date());

laboratoryLog.setUser(UserUtils.getUser());

laboratoryLogService.save(laboratoryLog);

laboratory.setLaboratoryLogId(laboratoryLog.getId());

laboratoryService.updates(laboratory);

map.put("flag", true);

map.put("url", "laboratory\_list2.do");

JsonUtils.toJson(map);

}

/\*\*

\* 取消预约

\*

\* @return

\*/

public void qx() throws IOException {

Laboratory laboratory1 = laboratoryService.findById(laboratory.getId());

LaboratoryLog laboratoryLog = laboratoryLogService.findById(laboratory1.getLaboratoryLogId());

if (laboratoryLog.getUser().getId() == UserUtils.getUser().getId()){

laboratory1.setIsYy(0);

laboratory1.setUser(null);

laboratoryService.update(laboratory1);

laboratoryLog.setIsYy(0);

laboratoryLog.setEndTime(new Date());

laboratoryLog.setQx(UserUtils.getUser());

laboratoryLogService.updates(laboratoryLog);

map.put("flag", true);

map.put("url", "laboratory\_list2.do");

JsonUtils.toJson(map);

} else {

map.put("flag", false);

map.put("url", "laboratory\_list2.do");

JsonUtils.toJson(map);

}

}

/\*\*

\* 跳转add

\*

\* @return

\*/

public String add(){

User user1 = UserUtils.getUser();

ActionContext.getContext().put("bean", user1);

return SUCCESS;

}

/\*\*

\* 查询修改

\*

\* @return

\*/

public String edit() {

if (laboratoryId != null && !laboratoryId.equals("")){

Laboratory bean = laboratoryService.findById(laboratoryId);

ActionContext.getContext().put("bean", bean);

}

return SUCCESS;

}

/\*\*

\* 审核

\*

\* @return

\*/

public void updateSh() throws IOException {

user.setIsSh(1);

userService.updates(user);

map.put("flag", true);

map.put("url", "user\_list.do");

JsonUtils.toJson(map);

}