

南昌大学

软件学院大作业任务书

题 目： 软院宿舍财产管理系统

课程名称： 信息系统基础

专 业： 软件工程

班 级： 2018 级 9 班

学 号： 8002118240

学生姓名： 杨孟衡

起讫日期： 2020. 11. 18— 2020. 1.10

任课教师： 朱小刚 职称： 教授

系 主 任： 徐健锋

完成时间： 2020 年 1 月 10 日

大作业的要求和内容：（包括题目选择范围、技术要求、递交时间、考核方法等）

综合运用所学知识，个人独立完成一个信息系统的业务流程图设计，设计内容和功能不做具体限定，但不能过于简单，不得抄袭或者网上直接下载充当。最终提交的作业，是系统的业务流程图设计文档，可以手绘，也可以用设计工具导出后打印。

大作业的评分点涵盖大作业从选题、分析、设计三个过程。具体评分点及各评分点的比重如下：

- 选题 20%
 评分依据：选题的难度、创新度、工作量等
- 分析 20%
 评分依据：分析是否充分、表述是否明确、功能的实用价值等
- 设计 40%
 评分依据：结构完整性、内容充实度、格式符合度、图表规范程度等
- 创新 20%
 评分依据：附有部分业务代码，设计或算法是否有创新等

教师小结：

成绩： _____

教 师 签 名： _____

系 负 责 人： _____

学 生 姓 名： 杨 孟 衡

南昌大学

NANCHANG UNIVERSITY

信息系统基础设计大作业



题 目: 软院宿舍财产管理系统

学 院: 软件学院

专 业: 软件工程

班 级: 软件工程 1809

学 号: 8002118240

学生姓名: 杨孟衡

完成人数: 1 人

起讫日期: 2020-11-28—2021-01-10

任课教师: 朱小刚 职称: 教授

完成时间: 2021 年 01 月 10 日

填表日期: 2021 年 1 月 10 日

目 录

1 开发背景	1
1.1 系统开发背景与意义	1
1.2 研究内容	1
1.3 文档的组织结构	2
2 系统需求分析	3
2.1 系统需求分析	3
2.2 可行性分析	3
2.2.1 经济可行性	3
2.2.2 技术可行性	3
2.3 功能分析	4
2.4 关键技术	4
2.4.1 Ajax	4
2.4.2 MYSQL 数据库	4
2.4.3 JQuery	4
2.4.4 B/S 模式	5
2.4.5 Java 编程语言	5
3 系统设计	7
3.1 系统整体设计	7
3.2 系统功能	8
3.3 数据库设计	9
3.3.1 数据库概念设计	9
3.3.2 数据库逻辑设计	13
4 系统实现	18
4.1 功能实现及代码	18
4.1.1 首页	18

4.1.2 登录	19
4.1.3 盘点页面	22
4.1.4 报修页面	26
5 系统测试与分析	30
5.1 系统测试的目的与目标	30
5.2 系统测试意义	30
5.3 系统测试的方法	30
5.4 系统测试环境	31
5.5 软件测试	31
5.6 功能测试	32
5.7 测试总结	33
6 总结与展望	34
参 考 文 献	35

1 开发背景

1.1 系统开发背景与意义

宿舍管理是高校后勤工作中的一个重要组成部分，是高校“服务育人”中的一项内容。学生宿舍是学生们的家，学生在校学习期间一半以上的时间是在宿舍中渡过的。因此宿舍环境的好坏、生活设施的完善与否都将影响到学生们身心的发育和学校的稳定，与大学生的学习与成长有着紧密的联系。宿舍管理的内容十分丰富，包括宿舍的公共卫生、安全、纪律、水电及公共财产的管理，精神文明宿舍建设等诸方面，这些都与大学生们的日常生活息息相关。

学生宿舍的公共财产是为满足学生的生活学习需要而购置的，充分发挥公共财产的作用，保持它的完好程度和耐用程度，既可避免资金的浪费，又可以满足同学们的需要。而公共财产的完好程度与同学们对它的爱护有着密切的联系。在学生宿舍中发生的一些现象很令人深思：各方面条件相同的宿舍，但其中设施的完好程度却截然不同；入学时同时配备的桌凳、书架等物品，有的宿舍里保护的洁净完好；有些宿舍里却肢断腿残，伤痕累累；不少学生在放东西的空床位上洗脸、洗衣服，使洗浴用水经常溅湿床板致使霉烂。宿舍的门锁损坏现象亦很严重。上述现象不仅反映出人为的原因造成数量可观的公共财产的损坏，也暴露了部分学生爱护公共财产观念淡薄，如不引起重视，不仅仅是可见的公共财产的物质损失，而且对正在学习和成长的学生们的思想和公德意识会造成不良影响，并产生十分严重的后果。

因此，对高校系统进行分析研究并作出是一套是适应于高校的宿舍公共财产管理系统将有重要的实用价值。为了能正确适应各大高校学生宿舍公共财产管理的需求进程，本系统提供了宿舍的公共财产进行登记、遗失、保修等数据的管理，方便及满足了宿管的管理需求，以及学生维护宿舍公共财产安全的反馈需求。

1.2 研究内容

软院宿舍财产管理系统通过简洁、一目了然的方法进行界面设计，主界面上可以方便的查看各模块的详细明细等信息，研究的内容主要包括以下方面：

1. 公共财产基本信息登记模块：由宿管人员逐一对宿舍楼内每个房间的公共财产进行登记，包括购买时间、价格、使用年限、品类等，同时需要明确具体的使用者数据，

2. 公共财产盘点模块:由宿管定期清点各个宿舍的公共财产是否与分配的情况一致,盘点后的信息需要记录到系统。
3. 公共财产报废模块:对使用年限已到或损坏程度过于严重没有修复可能的公共财产需要进行报废处理。
4. 公共财产报修模块:公共财产如果出现毁损,需要维修的。

1.3 文档的组织结构

文档的组织结果安排如下:

第一章绪论介绍了系统开发的背景与意义,对主要研究内容进行了说明,对文档的组织结构进行了阐述。

第二章主要讨论了系统的需求以及实现相关功能的关键技术和方法。

第三章对系统进行了业务建设与设计,对系统的用户角色和功能需求作了初步设计。

第四章主要对系统相关实现作了更详细的阐述,阐述了系统开发过程中所设计的代码、接口等。

第五章系统测试与分析。

第六章主要阐述了个人总结和展望。

最后为参考文献。

2 系统需求分析

2.1 系统需求分析

需求分析就是详细分析要解决的问题，弄明白问题的要求，包括需要输入哪些数据，将要得到何种结果，最后期望输出什么等。总之，“需求分析”在软件工程当中就是确定要计算机“做什么”。需求分析通过对收集到的需求资料进行分析，深层次的说明需求所表达的含义，介绍软件需要开发的功能和特性，描述软件要提供的接口或者和其它软件交互的方式，确定其它的非功能性需求。所以，需求分析是确定软件需要完成的内容，而不是如何完成这些功能要求。

2.2 可行性分析

2.2.1 经济可行性

开发本系统的相关资料均可通过网上搜索、线下调查获得，在开发过程中所需的软件系统都可从相关站点免费获得。信息的获取都是通过自己去各大地区长询问人民群众得到，不需要任何其他费用。

所以从经济角度开发工程监管信息系统是完全可行的。

2.2.2 技术可行性

开发工具支持：Java 语言，目前很多公司都是用此语言，有不少专用的开发工具和插件还提交了可视化设计，这是非常方便的。

数据库管理系统采用 MySQL，它能够处理大量数据，同时保持数据的完整性、安全性。因此本系统的开发平台已成熟可行。

硬件方面，在科技飞速发展的今天，硬件更新速度越来越快，容量越来越大，可靠性越来越高，价格越来越便宜，因此硬件平台也能够满足本系统所需。

本系统采用的技术均属当前流行的开发技术，具有技术成熟、效率高、稳定、安全等优点，并且自行配置的兼容性能优越，能够支持以上软件的运行。所以软院宿舍财产管理系统在技术上是完全可行的。

2.3 功能分析

1. 登录功能

用户想进软院宿舍财产管理系统的话必须先登录才能查询。

2. 公共财产登记功能

管理员可以对财产进行登记管理, 可以添加、查看、修改、删除财产登记信息。

3. 公共财产盘点功能

管理员可以对财产进行盘点管理, 可以添加、查看、修改、删除财产盘点信息。

4. 公共财产报废功能

用户可以对财产进行废物处理, 可以添加、查看、修改、删除财产报废信息。

5. 公共财产报修功能

管理员可以对财产进行报修处理, 可以添加、查询、修改、删除财产报修信息。

2.4 关键技术

2.4.1 Ajax

Ajax 全称为“Asynchronous JavaScript and XML” (异步 JavaScript 和 XML), 是指一种创建交互式网页应用的网页开发技术。Ajax 技术可以提供高度交互的 Web 应用, 给予用户更丰富的页面浏览体验, Ajax 功能的实现主要依靠 XMLHttpRequest 对象及其属性和方法、发送请求和处理响应。

2.4.2 MYSQL 数据库

软院宿舍财产管理系统使用 MYSQL 平台, MYSQL 相比其它数据库软件有独特的优越性, 可视化的数据库创建平台, 自动生成数据库关系图, 视图的可视化编辑, 这些特性都为程序员带来了很大便捷。MYSQL 数据库和 JAVA 配合较为紧密, 选用 MYSQL 数据库, 无须再编写驱动或者下载驱动。ADO.NET 提供了所有操作 MYSQL 数据库的方法。而 MYSQL 数据库是相对比较成熟和完善的, 适合软院宿舍财产管理系统的要求。

2.4.3 JQuery

jQuery 是一个快速、简洁的 JavaScript 框架, 是继 Prototype 之后又一个优秀的 JavaScript 代码库 (或 JavaScript 框架)。jQuery 设计的宗旨是“write Less, Do More”, 即倡导写更少的代码, 做更多的事情。它封装 JavaScript 常用的功能代码, 提供一种简便的 JavaScript 设计模式, 优化 HTML 文档操作、事件处理、动画设计和

Ajax 交互。

2.4.4 B/S 模式

B/S (Browser/Server) 结构即浏览器和服务器结构。它是随着 Internet 技术的兴起, 对 C/S 结构的一种变化或者改进的结构。在这种结构下, 用户工作界面是通过 WWW 浏览器来实现, 极少部分事务逻辑在前端 (Browser) 实现, 但是主要事务逻辑在服务器端 (Server) 实现, 形成所谓三层结构。这样就大大简化了客户端电脑载荷, 减轻了系统维护与升级的成本和工作量, 降低了用户的总体成本 (TCO)。以目前的技术看, 局域网建立 B/S 结构的网络应用, 并通过 Internet/Intranet 模式下数据库应用, 相对易于把握、成本也是较低的。它是一次性到位的开发, 能实现不同的人员, 以不同的接入方式访问和操作共同的数据库; 它能有效地保护数据平台和管理访问权限, 服务器数据库也很安全。

B/S 模式最大的优点就是可以在任何地方进行操作而不需客户端安装任何专门的软件。只要有一台能上网的电脑就能使用, 客户端零维护, 系统的扩展非常容易。它具有分布性特点, 可以随时随地进行业务处理。业务扩展简单方便, 通过增加网页即可增加服务器功能。在维护方面, 只需要改变网页, 即可实现所有用户的同步更新, 共享性较强。它实现了跨平台的系统集成服务, 提供了异种机、异种网、异种应用服务的联机、联网、统一服务的开放性基础。而 C/S 模式开发是具有针对性的, 对客户端有一定的要求, 因此缺少通用性, 移植性, 业务变更不灵活, 在维护、系统升级方面都有很大的麻烦而且兼容性较差, 对于不同的开发工具, 相互之间很难兼容, 具有较大的局限性, 新技术不便于使用。另外, 它的开发成本较高, 需要具有一定专业水准的技术人员才能完成。

2.4.5 Java 编程语言

Java 是由 Sun Microsystems 公司推出的 Java 面向对象程序设计语言 (以下简称 Java 语言) 和 Java 平台总称。它具有三大特点如下:

第一、封装: 面向对象编程的核心思想之一就是数据和对数据的操作封装在一起。抽象即从具体的实例中抽取共同的性质形成一般的概念。封装就是将抽象得到的数据和行为 (或功能) 相结合, 形成一个有机的整体, 也就是将数据与操作数据的源代码进行有机的结合, 形成 “类”, 其中数据和函数都是类的成员。

封装的目的是增强安全性和简化编程, 使用者不必了解具体的实现细节, 而只是要通过外部接口, 以特定的访问权限来使用类的成员。

继承: 继承体现了一种先进的编程思想模式。子类可以继承父类的属性和方法, 即集成父类的数据和数据上的操作, 又可以增加子类特有的属性和方法, 提高了实例的拓

展性。本质上，父类和子类还是一个实体。

多态：多态就是通过传递给父类对象引用不同的子类对象从而表现出不同的行为。

同时还有其他的一些特性：多线程、安全性、健壮性、可移植性还面向对象等特性。

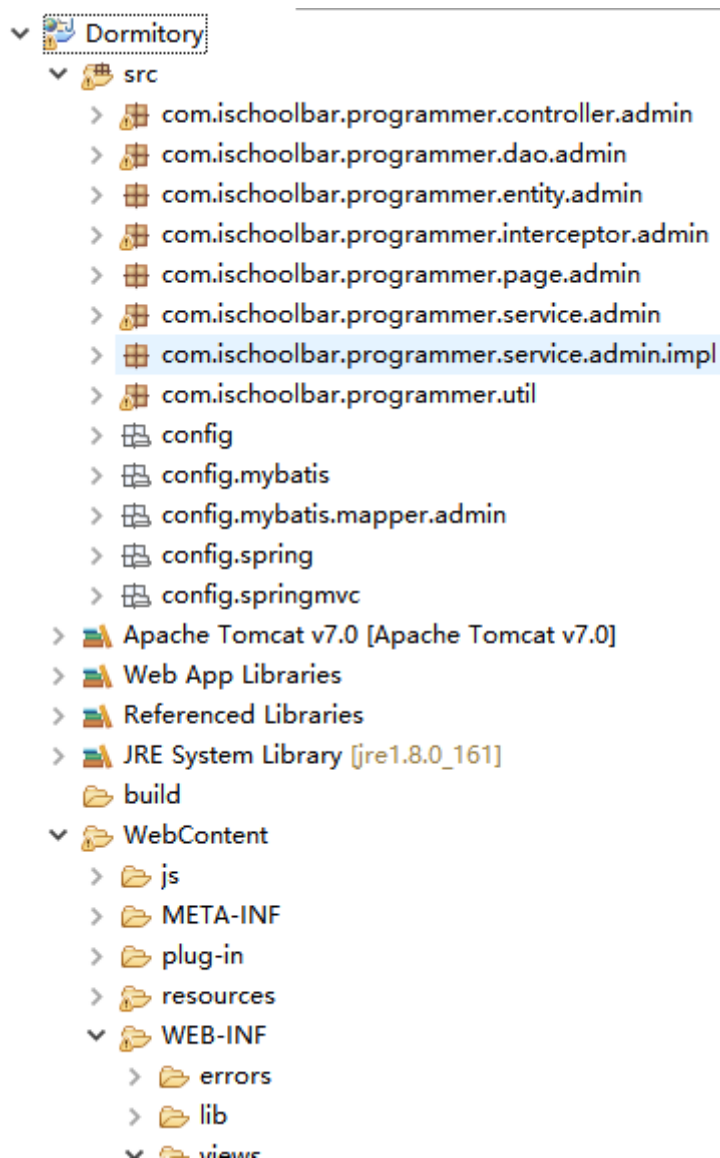
Sun 公司对 Java 语言有如下解释：Java 编程语言是个简单的、平台无关性、面向对象、多线程、安全、动态的语言。对此，本系统采用的编程语言就是 Java.

3 系统设计

3.1 系统整体设计

软院宿舍财产管理系统是基于 J2EE 搭建 B/S 系统具有典型的三层架构模式。

在 eclipse 中建立 Dormitory 项目工程，导入框架的 Jar 包，再进行配置文件的书写。然后再加入自己需要的插件，在完成这些工作之后，整个系统的框架就在 eclipse 环境中完成了搭建。具体的结构如图：



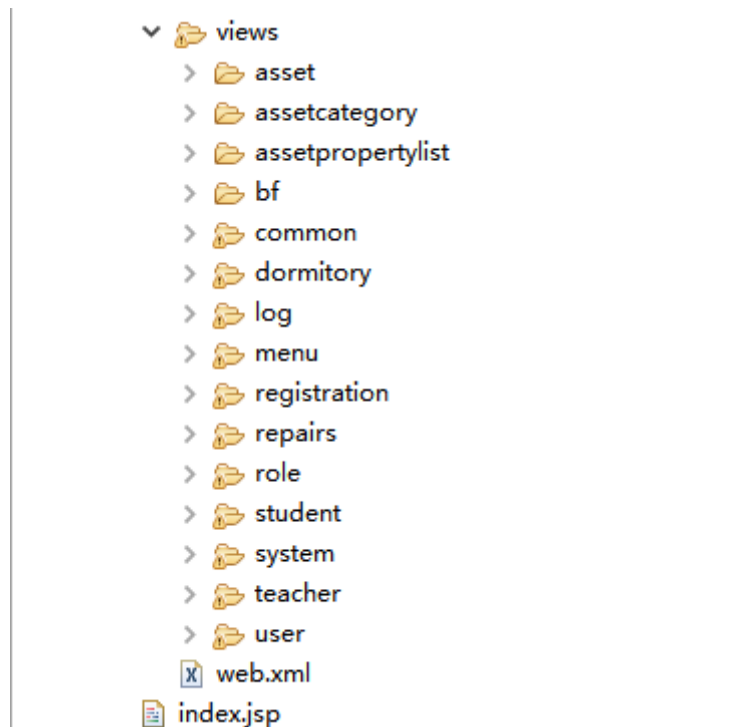


图 3.1 项目框架结构

3.2 系统功能

- 系统功能模块

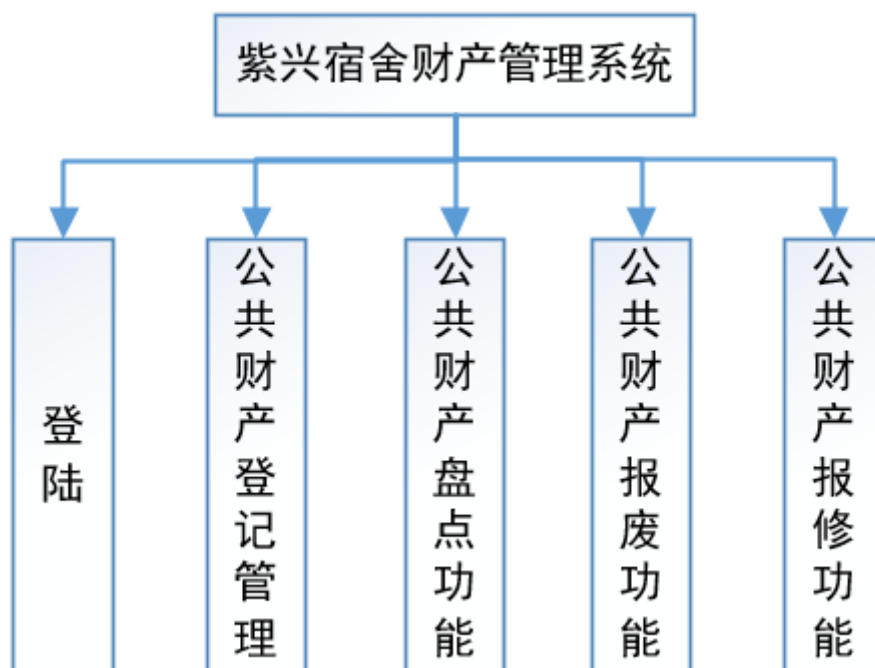


图 3.2 系统功能模块

● 管理员用例图

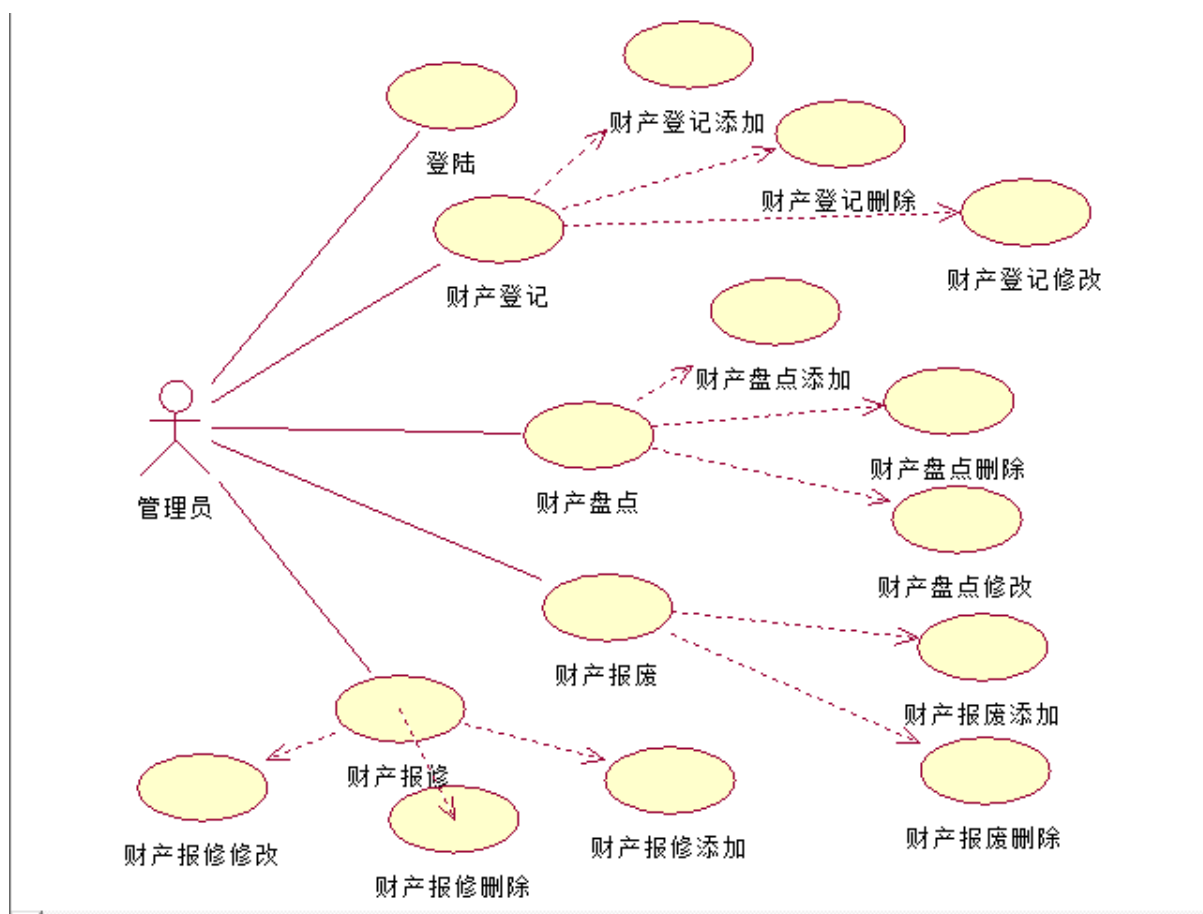


图 3.3 管理员用例图

3.3 数据库设计

数据库的设计关系到整个应用系统的运行效率，数据库设计得好，不仅有利于日常数据的维护更新，而且可以提高系统的运行效率，缩短数据查询响应周期，增加网站的流量。合理的数据库设计可以使围绕它支持的 Web 页面的 Java 代码简单化，易于实现，并且可以提高数据存储的效率，保证数据的完整一致。酒店管理系统采用 MySQL 作为后台数据库开发工具。

3.3.1 数据库概念设计

概念模型用于信息世界的建模，与具体的 DBMS 无关。为了把现实世界中的具体事

物抽象、组织为某一 DBMS 支持的数据模型。人们常常首先将现实世界抽象为信息世界，然后再将信息世界转换为机器世界。也就是说，首先把现实世界中的客观对象抽象为某一种信息结构，这种信息结构并不依赖于具体的计算机系统和具体的 DBMS，而是概念级的模型，然后再把模型转换为计算机上某一个 DBMS 支持的数据模型。实际上，概念模型是现实世界到机器世界的一个中间层次。

信息世界中包含的基本概念有实体和联系。

(1) 实体 (entity)

客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物，也可以是抽象的概念或联系。例如，一个学生、一门课、一个供应商、一个部门、一本书、一位读者等都是实体。

(2) 联系 (relationship)

在现实世界中，事物内部以及事物之间是有联系的，这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是组成实体的各属性之间的联系。两个实体型之间的联系可以分为 3 类，一对一联系，(1:1)；一对多联系(1 : n)；多对多联系(m : n)。

概念模型是对信息世界建模，所以概念模型应该能够方便、准确地表示信息世界中的常用概念。概念模型的表示方法很多，其中最为常用的是 P. P. S. Chen 于 1976 年提出的实体，联系方法 (Entity-Relationship Approach) 简记为 E-R 表示法)。该方法用 E-R 图来描述现实世界的概念模型，称为实体-联系模型，简称 E-R 模型。绘制软院宿舍财产管理系统的全局 E-R 模型如图 3-4 所示。

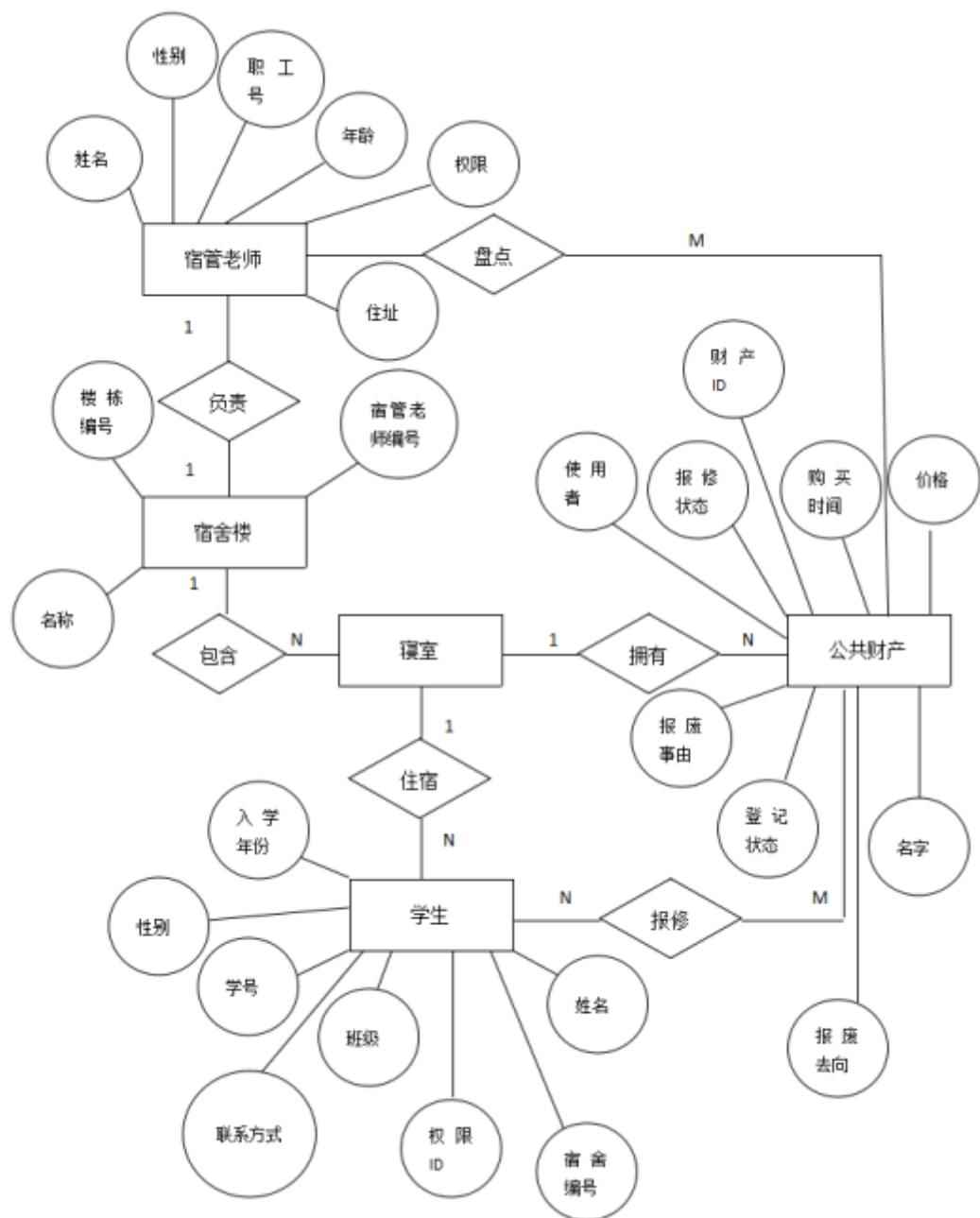


图 3.4 系统全局 E-R 图

根据系统分析的主要实体有：管理员、学生、宿管、公共财产。各个实体具体的描述属性图如下（实体属性在下图中并没有全部给出，因为属性过多的原因）：

1. 管理员实体(如图 3-5 所示)

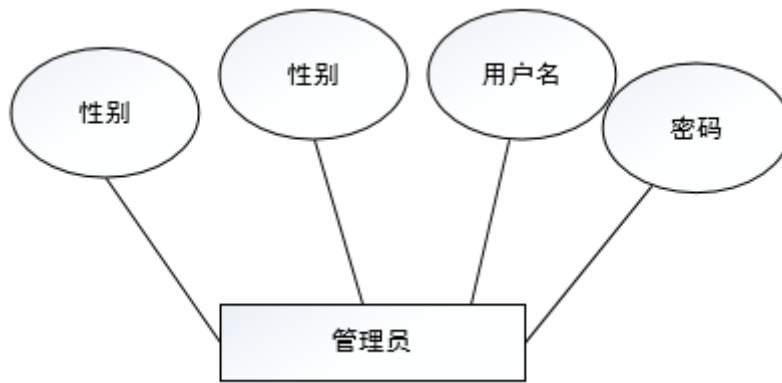


图 3-5 管理员实体

2. 学生实体(如图 3-6 所示)

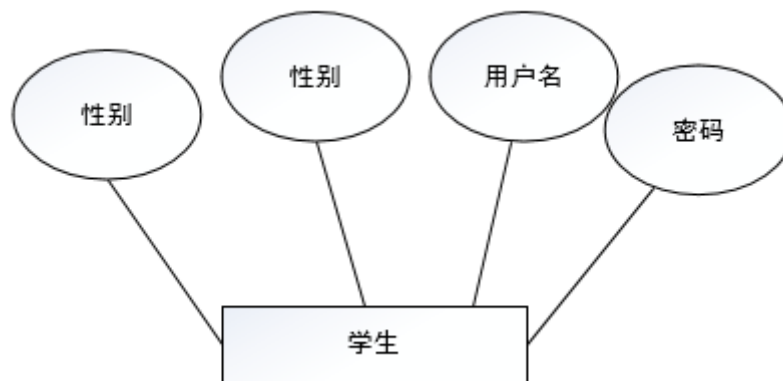


图 3-6 学生实体

3. 宿管实体(如图 3-7 所示)

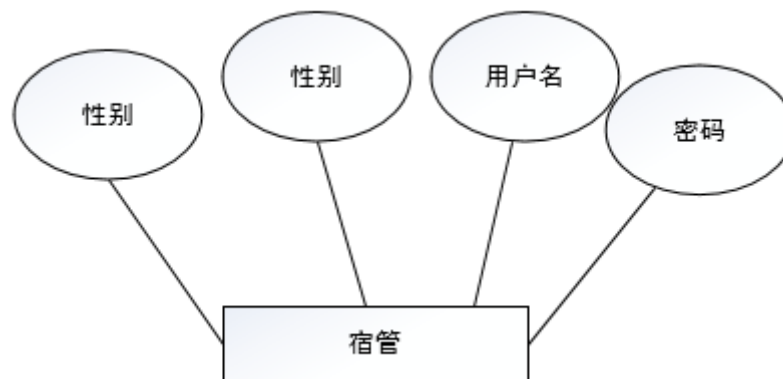


图 3-7 宿管实体

4. 财产实体(如图 3-8 所示)

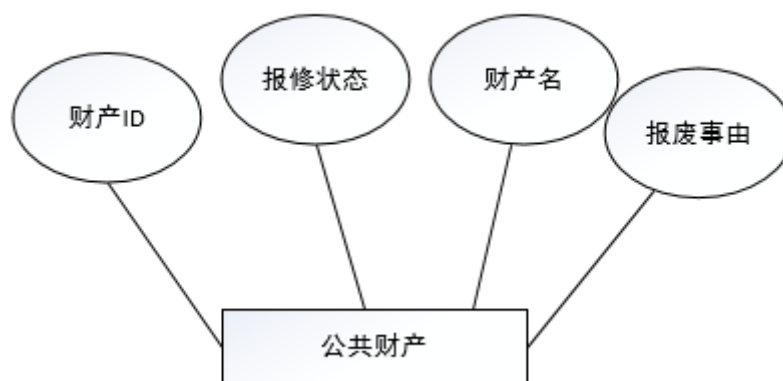


图 3-8 财产实体

3.3.2 数据库逻辑设计

数据库逻辑设计主要是把数据库概念设计时设计好的基本 E-R 图转换为与选用 DBMS 产品所支持的数据模型相符合的逻辑结构。它包括数据项、记录及记录间的联系、安全性和一致性约束等等。导出的逻辑结构是否与概念模式一致，从功能和性能上是否满足用户的要求，要进行模式评价。

本系统数据库名称为 dormitory，一共有十二张数据表，分别为财产表(如表 3-9 所示)、权限表(如表 3-10 所示)、财产类别表(如表 3-11 所示)、宿舍表(如表 3-12 所示)、日志表(如表 3-13 所示)、菜单表(如图 3-14 所示)、用户角色表(如图 3-15 所示)、学生表(如图 3-16 所示)、宿管表(如图 3-17 所示)、管理员表(如图 3-18 所示)、盘点表(如图 3-19 所示)、报修表(如图 3-20 所示)

表 3-9 财产表(tb_asset)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
assetid	int	主键	否	无	财产 ID (自增)
bedchamberid	int	无	是	无	宿舍 ID
createtime	date	无	是	无	创建时间
cateoryid	int	无	是	无	类别 ID
repairstatus	int	无	是	无	报修状态
scriptime	date	无	是	无	创建时间
scrapreason	varchar(128)	无	是	无	修改时间

registration status	varchar(128)	无	是	无	报废状态
assetname	varchar	无	是	无	财产名
uses	varchar	外键	是	无	使用者
price	float	无	是	无	价格

表 3-10 权限表(tb_authority)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
id	int	主键	否	无	权限 ID
roleId	int	无	否	无	角色 ID
menuId	int	无	否	无	菜单 ID

表 3-11 财产类别表(tb_assetcategory)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
id	int	主键	否	无	类别 ID（自增）
cname	varchar(10)	无	否	无	类别名
cyear	int	无	否	无	使用年限

表 3-12 宿舍表(tb_dormitory)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
dormitory Id	int	主键	否	无	宿舍 ID
name	varchar(100)	无	是	无	宿舍名
address	varchar(128)	无	是	无	地址
teacherId	int	无	是	无	老师 ID

表 3-13 日志表(tb_log)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
id	int	主键	否	无	日志 ID

content	varchar(10)	无	否	无	内容
createTime	datetime	无	是	无	创建时间

表 3-14 菜单表(tb_menu)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
id	int	主键	否	无	菜单 ID
parentId	int	无	否	无	菜单父 ID
name	varchar(32)	无	是	无	创建时间
url	varchar(128)	无	是	无	路劲
icon	varchar(32)	无	是	无	图标路劲

表 3-15 用户角色表(tb_role)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
id	int	主键	否	1	角色 ID
name	varchar (32)	无	否	无	名字
remark	varchar(128)	无	是	无	备注

表 3-16 学生表(tb_student)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
username	varchar(40)	主键	否	无	用户名
id	varchar (40)	无	否	无	学生 ID
class	varchar(255)	无	是	无	班级
year	date(0)	无	是	无	入学年份
password	varchar(32)	无	是	无	密码
sex	int(11)	无	是	无	性别
age	int(11)	无	是	无	年龄
roleId	int(11)	无	是	无	角色 ID
photo	varchar(128)	无	是	无	联系电话

dormitory _id	int(11)	无	是	无	宿舍 ID
------------------	---------	---	---	---	-------

表 3-17 宿管表(tb_teacher)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
id	varchar (32)	主键	否	1	宿管老师 ID
username	varchar (100)	无	否	无	用户名
Password	varchar(32)	无	是	无	密码
sex	int	无	是	无	性别
age	int	无	是	无	年龄
address	varchar(128)	无	是	无	地址
roleId	int	无	是	无	角色 ID
photo	varchar(128)	无	是	无	联系电话

表 3-18 管理员表(tb_user)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
id	int	主键	否	无	管理员 ID
username	varchar(32)	无	否	无	用户名
password	varchar(32)	无	否	无	密码
roleId	int	无	否	无	角色 ID
photo	varchar(128)	无	是	无	联系电话
sex	int	无	否	无	性别
age	int	无	是	无	年龄
address	varchar(128)	无	否	无	地址

表 3-19 盘点表(tb_registration)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
teaid	varchar(10)	主键	否	1	老师 ID
assetid	int	无	是	无	财产 ID

rstatus	int	无	是	无	状态
rcreateti me	date	无	是	无	创建时间
rid	int	无	否	无	财产 ID

表 3-20 报修表(tb_repairs)表结构

字段名称	数据类型	约束条件	可否空	默认值	注释
id	int	主键	否	1	报修 ID
stuid	varchar(10)	无	是	无	学生 ID
assetid	int	无	是	无	财产 ID
createtim e	date(0)	无	是	无	创建时间
status	int	无	是	无	状态
brank	varchar(100)	无	是	无	备注

4 系统实现

4.1 功能实现及代码

4.1.1 首页

本界面是软院宿舍财产管理系统的首页设计，其功能主要具体如图 4-1 所示

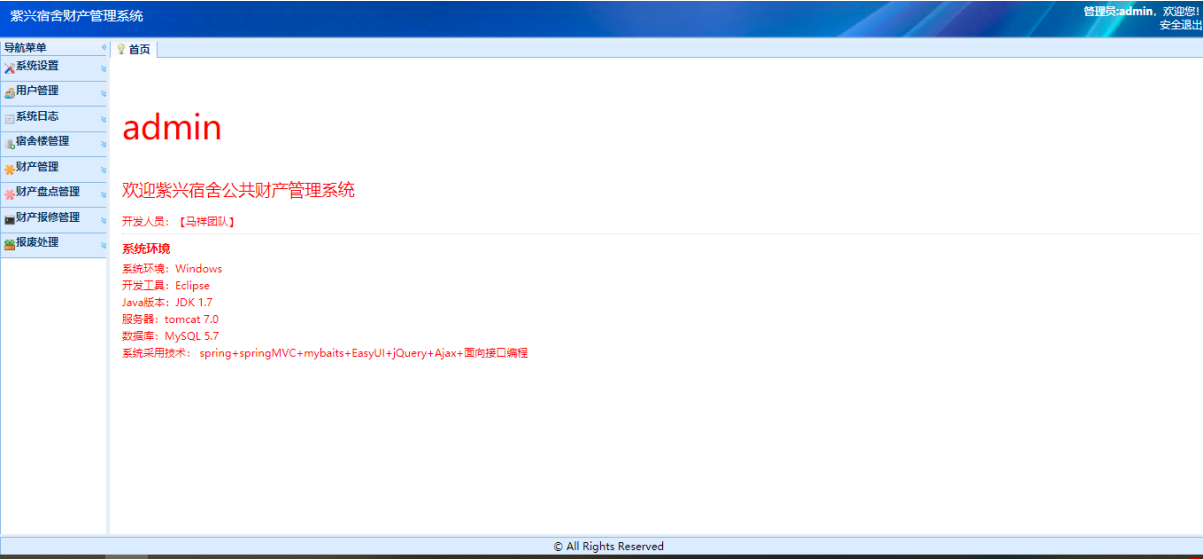


图 4-1

如图 4-2 所示, 为 web 前端关键代码

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>欢迎页面</title>
</head>
<body>
<div title="欢迎使用" style="padding:20px;overflow:hidden; color:red; " >
<p style="font-size: 50px; line-height: 60px; height: 60px;">
${admin.username}</p>
<p style="font-size: 25px; line-height: 30px; height: 30px;">欢迎紫兴宿舍公共财产管理系统</p>
<p>开发人员: 【马祥团队】</p>
<hr />
<h2>系统环境</h2>
<p>系统环境: Windows</p>
<p>开发工具: Eclipse</p>
<p>Java版本: JDK 1.7</p>
<p>服务器: tomcat 7.0</p>
<p>数据库: MySQL 5.7</p>
<p>系统采用技术: spring+springMVC+mybatis+EasyUI+jQuery+Ajax+面向接口编程</p>
</div>
</body>
</html>
```

图 4-2 首页前端代码

其后台关键代码为

```
5. @RequestMapping(value="/welcome",method=RequestMethod.GET)
```

```

6. public ModelAndView welcome(ModelAndView model){
7.     model.setViewName("system/welcome");
8.     return model;
9. }
10. /**
11.  * 跳转到首页页面
12.  * @param model
13.  * @return
14.  */
15. @RequestMapping(value="/login",method=RequestMethod.GET)
16. public ModelAndView login(ModelAndView model){
17.     model.setViewName("system/login");
18.     return model;
19. }

```

4.1.2 登录

本界面是软院宿舍财产管理系统的登录设计，其功能主要具体如图 4-3 所示



图 4-3 登陆页面

如图 4-4 所示, 为 web 前端关键代码


```

<div class="login-center clearfix">
<div class="login-center-img"></div>
<div class="login-center-input">
<input type="text" name="username" id="username" value="" placeholder="请输入您的用户名" onfocus="this.placeholder=&#39;&#39;"
onblur="this.placeholder=&#39;&#39;;请输入您的用户名&#39;&#39;">
<div class="login-center-input-text">用户名</div>
</div>
</div>
<div class="login-center clearfix">
<div class="login-center-img"></div>
<div class="login-center-input">
<input type="password" name="password" id="password" value="" placeholder="请输入您的密码" onfocus="this.placeholder=&#39;&#39;"
onblur="this.placeholder=&#39;&#39;;请输入您的密码&#39;&#39;">
<div class="login-center-input-text">密码</div>
</div>
</div>
<div class="login-center clearfix">
<div class="login-center-img"></div>
<div class="login-center-input">
<input style="width:50%;" type="text" name="cpacha" id="cpacha" value="" placeholder="请输入验证码" onfocus="this.placeholder=&#39;&#39;"
onblur="this.placeholder=&#39;&#39;;请输入验证码&#39;&#39;">
<div class="login-center-input-text">验证码</div>

</div>
</div>

```

图 4-4 登陆前端代码

其后台关键代码为：

```

1. public Map<String, String> loginAct(User user,String cpacha,Integer selected,HttpServletRequest request){
2.     Map<String, String> ret = new HashMap<String, String>();
3.     if(selected==1){
4.         ret.put("type", "error");
5.         ret.put("msg", "请选择角色！");
6.         return ret;
7.     }
8.     if(user == null){
9.         ret.put("type", "error");
10.        ret.put("msg", "请填写用户信息！");
11.        return ret;
12.    }
13.    if(StringUtils.isEmpty(cpacha)){
14.        ret.put("type", "error");
15.        ret.put("msg", "请填写验证码！");
16.        return ret;
17.    }
18.    if(StringUtils.isEmpty(user.getUsername())){
19.        ret.put("type", "error");
20.        ret.put("msg", "请填写用户名！");
21.        return ret;
22.    }
23.    if(StringUtils.isEmpty(user.getPassword())){
24.        ret.put("type", "error");
25.        ret.put("msg", "请填写密码！");
26.        return ret;
27.    }
28.    Object loginCpacha = request.getSession().getAttribute("loginCpacha");
29.    if(loginCpacha == null){//Session 没数据代表超时
30.        ret.put("type", "error");

```

```

31.         ret.put("msg", "会话超时，请刷新页面！"); //获取验证码
32.         return ret;
33.     }
34.     if(!cpacha.toUpperCase().equals(loginCpacha.toString().toUpperCase())){
35.         ret.put("type", "error");
36.         ret.put("msg", "验证码错误！");
37.         logService.add("用户名为"+user.getUsername()+"的用户登录时输入验证码错误!");
38.         return ret;
39.     }
40.     switch(selected){
41.         case 1://管理员   user 表
42.             User findByUsername = userService.findByUsername(user.getUsername()); //去数据库查询
43.             if(findByUsername == null){
44.                 ret.put("type", "error");
45.                 ret.put("msg", "该用户名不存在！");
46.                 logService.add("管理员登录时，用户名为"+user.getUsername()+"的用户不存在!");
47.                 return ret;
48.             }
49.             if(!user.getPassword().equals(findByUsername.getPassword())){
50.                 ret.put("type", "error");
51.                 ret.put("msg", "密码错误！");
52.                 logService.add("管理员登录时,用户名为"+user.getUsername()+"的用户登录时输入密码错误!");
53.                 return ret;
54.             }
55.             request.getSession().setAttribute("admin", findByUsername);
56.             request.getSession().setAttribute("myrole", 1);
57.             break;
58.         case 2://老师用户   teacher 表
59.             TeacherEntity findByTeachername = teacherService.findByTeachername(user.getUsername());
60.             if(findByTeachername == null){
61.                 ret.put("type", "error");
62.                 ret.put("msg", "该用户名不存在！");
63.                 logService.add("老师登录时，用户名为"+user.getUsername()+"的用户不存在!");
64.                 return ret;
65.             }
66.             if(!user.getPassword().equals(findByTeachername.getPassWord())){
67.                 ret.put("type", "error");
68.                 ret.put("msg", "密码错误！");
69.                 logService.add("老师登录时,职工号为"+user.getUsername()+"的用户登录时输入密码错误!");
70.                 return ret;
71.             }
72.             request.getSession().setAttribute("admin", findByTeachername);
73.             request.getSession().setAttribute("myrole", 2);

```

```

74.         break;
75.     case 4://学生用户    student 表
76.         StudentEntity findByStudentname = studentService.findByStudentname(user.getUsername());
77.         if(findByStudentname == null){
78.             ret.put("type", "error");
79.             ret.put("msg", "该用户名不存在! ");
80.             logService.add("学生登录时, 学号为"+user.getUsername()+"的用户不存在!");
81.             return ret;
82.         }
83.         if(!user.getPassword().equals(findByStudentname.getPassword())){
84.             ret.put("type", "error");
85.             ret.put("msg", "密码错误! ");
86.             logService.add("学生登录时,学号为"+user.getUsername()+"的用户登录时输入密码错误!");
87.             return ret;
88.         }
89.         request.getSession().setAttribute("admin", findByStudentname);
90.         request.getSession().setAttribute("myrole", 4);
91.         break;
92.     }
93.     Role role = roleService.find(Long.valueOf(selected));
94.     List<Authority> authorityList = authorityService.findListByRoleId(role.getId());//根据角色获取权限列表
95.     String menuIds = "";
96.     for(Authority authority:authorityList){
97.         menuIds += authority.getMenuId() + ",";
98.     }
99.     if(!StringUtils.isEmpty(menuIds)){
100.         menuIds = menuIds.substring(0,menuIds.length()-1);
101.     }
102.     List<Menu> userMenus = menuService.findListByIds(menuIds);
103.     request.getSession().setAttribute("role", role);
104.     request.getSession().setAttribute("userMenus", userMenus);
105.     ret.put("type", "success");
106.     ret.put("msg", "登录成功! ");
107.     logService.add("用户名为{" +user.getUsername()+"}, 角色为{" +role.getName()+"}的用户登录成功!");
108.     return ret;
109. }

```

4.1.3 盘点页面

本界面是软院宿舍财产管理系统的盘点页面设计，其功能主要具体如图 4-5 所示

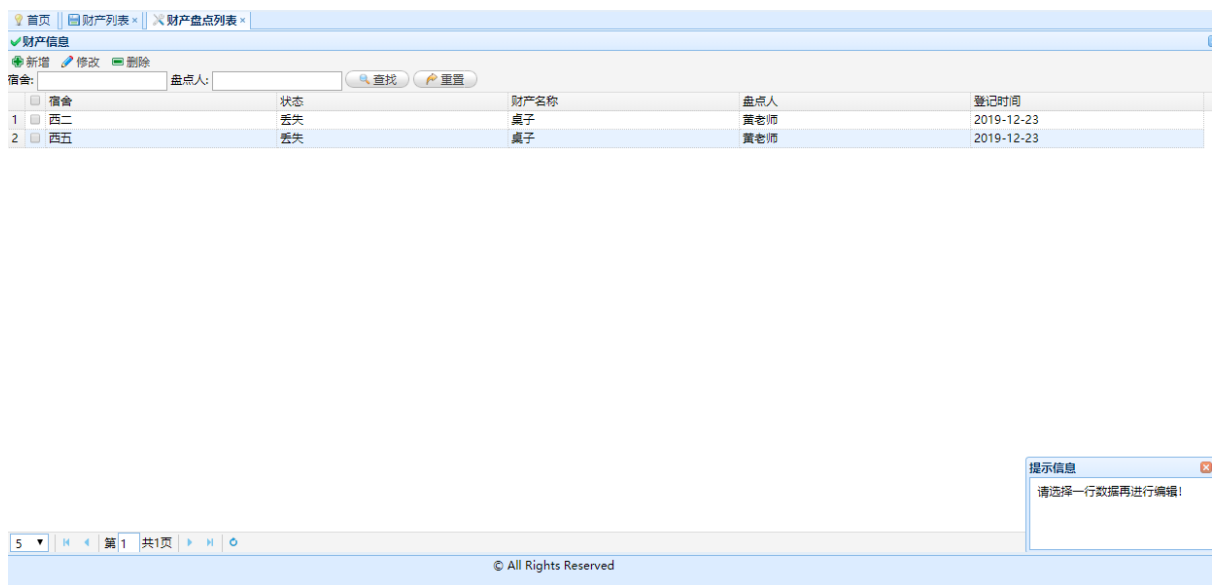


图 4-5 盘点页面

如图 4-6 所示, 为 web 前端关键代码

```
<html>
<head>
  <meta charset="UTF-8">
  <title>盘点</title>
</head>
<body>
  <table id="dg" class="" title="列表" style="width: 700px; height: 300px">
    <thead>
      <tr>
        <th data-options="field:'ck',checkbox:true"></th>
        <th data-options="field:'name',width:80">宿舍</th>
        <th data-options="field:'rstatus',width:80" formatter="myTest">状态</th>
        <th data-options="field:'assetname',width:80">财产名称</th>
        <th data-options="field:'username',width:80">盘点人</th>
        <th data-options="field:'recreatetime',width:80" formatter="changeDate">登记时间</th>
      </tr>
    </thead>
  </table>

  <div id="toolbar">
    <a href="#" class="easyui-linkbutton add" iconCls="icon-add" onclick="create()" plain="true">新增</a>
    <a href="#" class="easyui-linkbutton edit" iconCls="icon-edit" onclick="edit()" plain="true">修改</a>
    <a href="#" class="easyui-linkbutton remove" iconCls="icon-remove" onclick="del()" plain="true">删除</a>
    <div>
      宿舍: <input class="easyui-textbox" id="paramname">
      盘点人: <input class="easyui-textbox" id="paramusername">
      <a href="#" class="easyui-linkbutton" onclick="doSearch()" iconCls="icon-search">查找</a>
      <a href="#" class="easyui-linkbutton" onclick="doReset()" iconCls="icon-redo">重置</a>
    </div>
  </div>
  <div id="dlg" class="easyui-dialog" title="数据参数" data-options="modal:true" style="width: 400px; height: 350px;" closed="true" buttons="#dlg-butto.
    <form method="post" id="fm">
```

图 4-6 盘点前端代码

其后台关键代码为:

1. @RequestMapping(value = "/delete", method = RequestMethod.POST)
2. @ResponseBody
3. public AjaxJson delete(String ids) {
4. AjaxJson j = new AjaxJson();
5. if (StringUtils.isEmpty(ids)) {
6. j.setMsg("删除失败!");
7. j.setSuccess(false);
8. return j;
9. }
10. if (ids.contains(",")) {
11. ids = ids.substring(0, ids.length() - 1);
12. }

```

13. if (registrationService.delete(ids) <= 0) {
14. j.setMsg("盘点信息删除失败，请联系管理员！");
15. j.setSuccess(false);
16. return j;
17. }
18. j.setMsg("盘点信息删除成功！");
19. j.setSuccess(true);
20. return j;
21. }
22. @RequestMapping(value = "/edit", method = RequestMethod.POST)
23. @ResponseBody
24. public AjaxJson edit(RegistrationEntity registrationEntity) {
25. AjaxJson j = new AjaxJson();
26. if (registrationEntity == null) {
27. j.setMsg("请填写正确的盘点信息！");
28. j.setSuccess(false);
29. return j;
30. }
31. if (StringUtils.isEmpty(registrationEntity.getRcreatetime().toLocaleString())) {
32. j.setMsg("请填写盘点时间");
33. j.setSuccess(false);
34. return j;
35. }
36. if (StringUtils.isEmpty(String.valueOf(registrationEntity.getRstatus()))) {
37. j.setMsg("请填写设施状态");
38. j.setSuccess(false);
39. return j;
40. }
41. j.setMsg("保存成功！");
42. j.setSuccess(true);
43. try {
44. registrationService.updateByExampleSelective(registrationEntity);
45. } catch (Exception e) {
46. j.setMsg("保存失败！");
47. j.setSuccess(false);
48. }
49. return j;
50. }

51. @RequestMapping(value = "/list", method = RequestMethod.POST)
52. @ResponseBody
53. public Map<String, Object> getList(Page page,
54. @RequestParam(name = "username", required = false, defaultValue = "") String username,
55. @RequestParam(name = "name", required = false, defaultValue = "") String name, HttpServletRequest request) {
56. Map<String, Object> ret = new HashMap<String, Object>();

```

```

57. Map<String, Object> queryMap = new HashMap<String, Object>();
58. Integer myrole = (Integer) request.getSession().getAttribute("myrole");
59. switch (myrole) { // 1 管理员 2 老师 4 学生
60. case 2:
61. TeacherEntity teacherEntity = (TeacherEntity) request.getSession().getAttribute("admin");// 获取老师
62. queryMap.put("teaid", teacherEntity.getId());
63. break;
64. }
65. queryMap.put("username", username);
66. queryMap.put("name", name);
67. queryMap.put("offset", page.getOffset());
68. queryMap.put("pageSize", page.getRows());
69. ret.put("rows", registrationService.findList(queryMap));
70. ret.put("total", registrationService.getTotal(queryMap));
71. return ret;
72. }
73. @RequestMapping(value = "/add", method = { RequestMethod.POST, RequestMethod.GET })
74. @ResponseBody
75. public AjaxJson add(RegistrationEntity registrationEntity, HttpServletRequest request) {
76. AjaxJson j = new AjaxJson();
77. if (registrationEntity == null) {
78. j.setMsg("请填写正确的盘点信息!");
79. j.setSuccess(false);
80. return j;
81. }
82. if (StringUtils.isEmpty(String.valueOf(registrationEntity.getAssetid()))) {
83. j.setMsg("请填写财产名");
84. j.setSuccess(false);
85. return j;
86. }
87. if (StringUtils.isEmpty(registrationEntity.getRcreatetime().toGMTString())) {
88. j.setMsg("请填写登记时间");
89. j.setSuccess(false);
90. return j;
91. }
92. Integer myrole = (Integer) request.getSession().getAttribute("myrole");
93. switch (myrole) { // 1 管理员 2 老师 4 学生
94. case 2: // 老师
95. TeacherEntity teacherEntity = (TeacherEntity) request.getSession().getAttribute("admin");// 获取老师
96. registrationEntity.setTeaid(teacherEntity.getId());
97. break;
98. case 1:
99. j.setMsg("无权限");
100. j.setSuccess(false);
101. return j;

```

```

102. }
103. j.setMsg("保存成功! ");
104. j.setSuccess(true);
105. try {
106.     registrationService.insertSelective(registrationEntity);
107. } catch (Exception e) {
108.     j.setMsg("保存失败! ");
109.     j.setSuccess(false);
110. }
111.     return j;
112. }

```

4.1.4 报修页面

本界面是软院宿舍财产管理系统的报修设计，其功能主要具体如图 4-7 所示

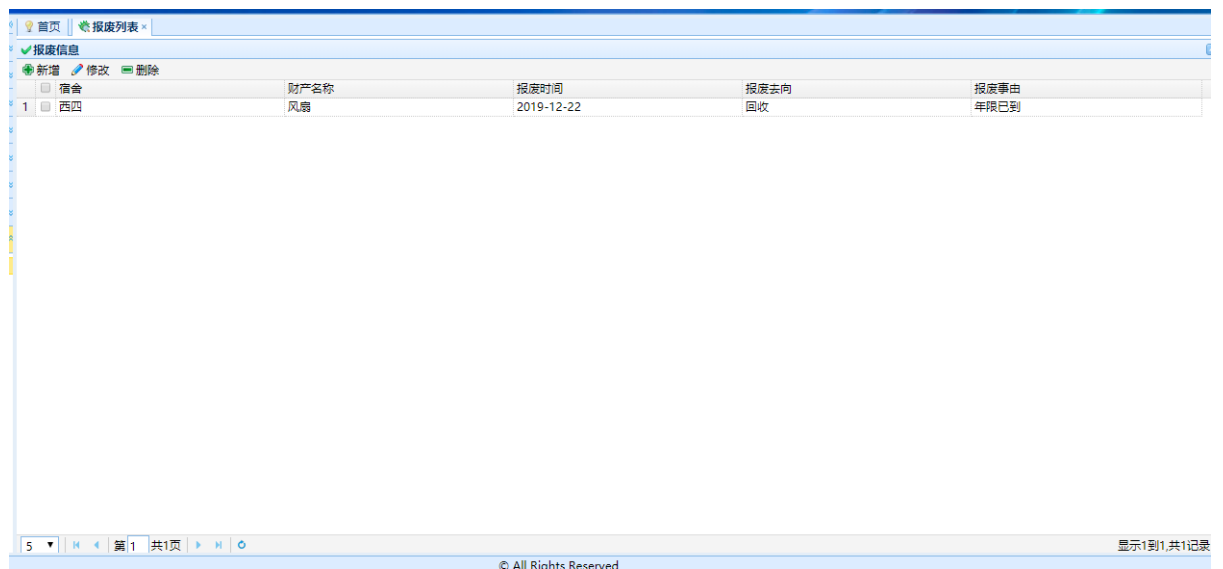


图 4-7 报修页面

如图 4-8 所示, 为 web 前端关键代码

```

function upload() {
    if ($("#photo-file").val() == '')
        return;
    var formData = new FormData();
    formData
        .append('photo', document.getElementById('photo-file').files[0]);
    $("#process-dialog").dialog('open');
    var interval = setInterval(start, 200);
    $.ajax({
        url : 'upload_photo',
        type : 'post',
        data : formData,
        contentType : false,
        processData : false,
        success : function(data) {
            clearInterval(interval);
            $("#process-dialog").dialog('close');
            if (data.type == 'success') {
                $("#preview-photo").attr('src', data.filepath);
                $("#add-photo").val(data.filepath);
                $("#edit-preview-photo").attr('src', data.filepath);
                $("#edit-photo").val(data.filepath);
            } else {
                $.messager.alert("消息提醒", data.msg, "warning");
            }
        },
        error : function(data) {
            clearInterval(interval);
            $("#process-dialog").dialog('close');
            $.messager.alert("消息提醒", "上传失败!", "warning");
        }
    });
};

```

图 4-8 报修前端代码

其后台关键代码为：

1. @RequestMapping(value = "/list", method = RequestMethod.**GET**)
2. **public** ModelAndView list(ModelAndView model) {
3. model.setViewName("repairs/list");
4. **return** model;
5. }
6. @RequestMapping(value = "/edit", method = RequestMethod.**POST**)
7. @ResponseBody
8. **public** Map<String, String> edit(RepairsEntity repairsEntity) {
9. Map<String, String> ret = **new** HashMap<String, String>();
10. **if** (repairsEntity == **null**) {
11. ret.put("type", "error");
12. ret.put("msg", "请填写正确的报修信息！");
13. **return** ret;
14. }
15. **if** (repairsService.edit(repairsEntity) <= 0) {
16. ret.put("type", "error");
17. ret.put("msg", "报修信息修改失败，请联系管理员！");
18. **return** ret;
19. }
20. Asset as = **new** Asset();
21. as.setAssetid(repairsEntity.getAssetid());
22. as.setRepairstatus(0);
23. assetService.edit(as);
24. ret.put("type", "success");


```

25. ret.put("msg", "报修信息修改成功! ");
26. return ret;
27. }
28. @RequestMapping(value = "/add", method = RequestMethod.POST)
29. @ResponseBody
30. public Map<String, String> add(RepairsEntity repairsEntity, HttpServletRequest request) {
31. Map<String, String> ret = new HashMap<String, String>();
32. if (repairsEntity == null) {
33. ret.put("type", "error");
34. ret.put("msg", "请填写正确的报修信息! ");
35. return ret;
36. }
37. if (StringUtils.isEmpty(repairsEntity.getBrank())) {
38. ret.put("type", "error");
39. ret.put("msg", "请填写报修详情! ");
40. return ret;
41. }
42. if (StringUtils.isEmpty(String.valueOf(repairsEntity.getAssetid()))) {
43. ret.put("type", "error");
44. ret.put("msg", "请填写报修财产! ");
45. return ret;
46. }
47. Integer myrole = (Integer) request.getSession().getAttribute("myrole");
48. if (myrole != 4) { // 学生
49. ret.put("type", "error");
50. ret.put("msg", "无权限");
51. return ret;
52. }
53. StudentEntity stu = (StudentEntity) request.getSession().getAttribute("admin"); // 获取学生
54. repairsEntity.setCreatetime(new Date());
55. repairsEntity.setStatus(1); // 1 正在维护
56. repairsEntity.setStuid(stu.getId());
57. Asset as = new Asset();
58. as.setAssetid(repairsEntity.getAssetid());
59. as.setRepairstatus(1);
60. assetService.edit(as);
61. if (repairsService.add(repairsEntity) <= 0) {
62. ret.put("type", "error");
63. ret.put("msg", "报修信息添加失败, 请联系管理员! ");
64. return ret;
65. }
66. ret.put("type", "success");
67. ret.put("msg", "报修信息添加成功! ");
68. return ret;
69. }

```


5 系统测试与分析

测试是开发时期最后一个阶段，是保证软件质量的重要手段。软件测试就是在受控制的条件下对系统或应用程序进行操作并评价操作结果的过程，所谓控制条件应包括正常条件与非正常条件。软件测试过程中应该故意地去促使错误的发生，也就是事情在不该出现的时候出现或者在应该出现的时候没有出现。从本质上说，软件测试是“探测”，在“探测”中发现软件的毛病。

5.1 系统测试的目的与目标

在此系统进行初步实现之后，开始进行对系统进行测试，找出系统中存在的 Bug，通过测试，用提交的 Bug 报告来为以后软件的改进提供标准和参考，能够在以后的系统改进中找到依据。

测试后的软件各模块基本功能可以顺利进行，尽可能的提高软件的健壮性。

5.2 系统测试意义

软件危机曾经是软件界甚至整个计算机界最热门的话题。为了解决这场危机，软件从业人员、专家和学者做出了大量的努力。现在人们已经逐步认识到所谓的软件危机实际上仅是一种状况，那就是软件中有错误，正是这些错误导致了软件开发在成本、进度和质量上的失控。有错是软件的属性，而且是无法改变的，因为软件是由人来完成的，所有由人做的工作都不会是完美无缺的。问题在于我们如何去避免错误的产生和消除已经产生的错误，使程序中的错误密度达到尽可能低的程度。

5.3 系统测试的方法

- 从是否关心软件内部结构和具体实现的角度划分：黑盒测试和白盒测试；
- 从是否执行程序的角度：静态测试和动态测试；
- 从软件开发的过程按阶段划分有：单元测试、集成测试、确认测试、系统测试、验收测试、回归测试、Alpha 测试、Beta 测试；

单元测试又称模块测试，是针对软件设计的最小单位——程序模块（这里所说的程序模块在 Java 中一个模块就是一个方法），进行正确性检验的测试工作。其目的在于发现各模块内部可能存在的各种差错。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

集成测试（组装测试、联合测试），通常在单元测试的基础上，需要将所有模块按

照设计要求组装成为系统。这时需要考虑的问题是：

- 在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；
- 一个模块的功能是否会对另一个模块的功能产生不利的影响；
- 各个子功能组合起来，能否达到预期要求的父功能；
- 全局数据结构是否有问题；
- 单个模块的误差累积起来，是否会放大，从而达到不能接受的程度。

确认测试（Validation Testing），确认测试又称有效性测试。任务是验证软件的功能和性能及其它特性是否与用户的要求一致。对软件的功能和性能要求在软件需求规格说明书中已经明确规定。它包含的信息就是软件确认测试的基础。

系统测试（System Testing），是将通过确认测试的软件，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其它系统元素结合在一起，在实际运行环境下，对计算机系统进行一系列的组装测试和确认测试。系统测试的目的在于通过与系统的需求定义作比较，发现软件与系统的定义不符合或与之矛盾的地方。

验收测试（Acceptance Testing），在通过了系统的有效性测试及软件配置审查之后，就应开始系统的验收测试。验收测试是以用户为主的测试。软件开发人员和 QA（质量保证）人员也应参加。由用户参加设计测试用例，使用生产中的实际数据进行测试。在测试过程中，除了考虑软件的功能和性能外，还应对软件的可移植性、兼容性、可维护性、错误的恢复功能等进行确认。

5.4 系统测试环境

处理器：Inter(R) Core (TM)2 Duo T5750

内存：6GB

硬盘：500G

操作系统：Windows 10

数据库：mysql

5.5 软件测试

通过对软院宿舍财产管理系统的分析、设计和实现，完成了系统功能，下面对软院宿舍财产管理系统测试，完成软件开发最后步骤。

在测试过程中，可以使用测试工具进行系统测试、功能测试，测试过程中，遇到的 Bug，需要及时修改，并形成测试文档，说明测试过程、修改情况。

软件测试包括了黑盒测试和白盒测试，不同的测试方法，测试目的不同，通过对软

院宿舍财产管理系统的详细测试，完成了测试目的。

5.6 功能测试

软院宿舍财产管理系统包括了多个功能，下面对各主要功能进行测试用例说明。由于测试过程具有一定的相似性，所以本系统选取一部分功能进行测试，其余相似功能模块不再一一展示。

表 5.1 登陆测试用例

功能特性	用户登录验证				
测试目的	验证是否输入合法的信息				
测试数据	用户名称：1111 密码：1111				
测试内容	操作描述	数据	期望结果	实际结果	测试状态
1	输入用户姓名，按“登陆”按钮。	用户姓名：1111， 密码为空	显示警告信息“用户名或密码误!”	显示警告信息“用户名或密码误!”	与期望结果相同
2	输入密码，按“登陆”按钮。	用户姓名为空，密码：1111	显示警告信息“用户名或密码误!”	显示警告信息“用户名或密码误!”	与期望结果相同
3	输入用户姓名和密码，按“登陆”按钮。	用户姓名：1， 密码：1	显示警告信息“用户名或密码误!”	显示警告信息“用户名或密码误!”	与期望结果相同
4	输入用户姓名和密码，按“登陆”按钮。	用户名：1111， 密码：1111	正确登入到会员操作界面	正确登入到会员操作界面	与期望结果相同

表 5.2 登录测试用例

功能特性	用户注册验证				
测试目的	验证是否能注册成功				
测试数据	用户名称：1111 密码：1111				
测试内容	操作描述	数据	期望结果	实际结果	测试状态
1	输入用户姓名，按“注册”按钮。	用户姓名：1111， 密码为空	显示警告信息“密码不能为空!”	显示警告信息“密码不能为空!”	与期望结果相同

2	输入密码，按“注册”按钮。	用户姓名为空，密码：1111	显示警告信息“用户名不能为空!”	显示警告信息“用户名不能为空!”	与期望结果相同
3	输入用户姓名和密码，按“注册”按钮。	用户姓名：1，密码：1	显示警告信息“注册成功!”	显示警告信息“注册成功”	与期望结果相同
4	输入用户姓名和密码，按“登陆”按钮。	用户名：1111，密码：1111	正确登入到会员操作界面	正确登入到会员操作界面	与期望结果相同

5.7 测试总结

把开始的代码写得越好，它出现的错误也就越少，你也就越能相信所做过的测试是彻底的。系统化测试以一种有序方式设法探测潜在的麻烦位置。同样，毛病最可能出现在边界，这可以通过手工的或者程序的方式检查。自动进行测试是最理想的，用得越多越好，因为机器不会犯错误、不会疲劳、不会用臆想某此实际无法工作的东西能行来欺骗自己。回归测试检查一个程序是否能产生与它们过去相同的输出。在做了小改变之后就测试是一种好技术，能帮助我们出现问题的范围局部化，因为新问题一般就出现在新代码里面。

测试和排错常常被说成是一个阶段，实际上它们根本不是同一件事。简单地说，排错是在你已经知道程序有问题时要做的事情。而测试则是在你在认为程序能工作的情况下，排错是在你已经知道程序有问题时要做的事情。而测试则是在你在认为程序能工作的情况下，为设法打败它而进行的一整套确定的系统化的试验。

Edsger Dijkstra 有一个非常有名的说法：测试能够说明程序中有错误，但却不能说明其中没有错误。他的希望是，程序可以通过某种构造过程正确地做出来，这样就不再会有错误了，因此测试也就不必要了。这确实是个美好生活的目标，但是，对今天的实际程序而言，这仍然还只是一个理想。所以应该集中精力讨论如何测试，如何才能更快地发现程序错误，如何才能使得工作更有成效、效率更高。

6 总结与展望

软院宿舍财产管理系统，实现了该网站的后台管理模块、前台管理模块、会员管理模块、订单管理模块、搜索管理模块、单点登录管理模块。本文档介绍了软院宿舍财产管理系统的设计与实现过程，主要经过需求分析，相关技术研究，系统架构设计，后台数据库设计，模块设计和数据库设计，最后进行了编码，最终实现了一个 B/S 模式的软院宿舍财产管理系统。在系统的设计和开发过程中，主要完成了以下工作：

1. 对软院宿舍财产管理系统进行用户需求调研，将现有业务与传统技术进行对比总结，完成本系统的用户需求分析。通过对用户需求分析和技术背景研究，分析了系统的业务流程，并采用 UML 工具对用户需求进行统一建模，导出系统的软件需求分析说明。
2. 软院宿舍财产管理系统使用 ECLIPSE 和 MYSQL 数据库开发。
3. 根据系统业务需求对系统数据库进行详细设计，主要采用了数据库 3NF 标准对数据库内部各种关系进行抽象分析，建立数据库 E-R 模型图，最大程度上避免数据冗余，保证数据的一致性。
4. 实现了软院宿舍财产管理系统的架构设计，在系统的设计与实现的过程中，系统采用经典的三层结构，面向对象的 JAVA 语言和 MYSQL 数据库存储数据。三层框架包括了界面层、逻辑层和数据层。

此外，我还觉得，我个人在这次设计中走了很多弯路。主要是因为平时很少接触软件开发工作，在应用方面缺乏经验，以后还需要更多的努力。

对我来说，这次设计的本身所产生的影响，还远远没有结束，我从本次课程设计中学到了许多课本上没有的知识。通过自己的学习和努力；通过老师的指导和教育，使我不仅仅在知识水平和解决实际问题的能力上有了很大的提高。还从思想的深处体会到，要把自己的所学变成现实时所面对的种种难题。

系统不免有错误和待改进之处，请各位老师提出宝贵意见。

参 考 文 献

- [1] 刘京华等. Java Web 整合开发王者归来+Java Web 整合开发实战. 北京:清华大学出版社, 2017. 1
- [2] 罗永权. HTML 语言的网页制作技巧与方法研究 [J]. 计算机产品与流通, 2019(10):151
- [3] 老虎工作室, JSP 动态网页制作基础培训教材[M] (第 2 版), 北京, 清华大学出版社, 2005. 1
- [4] 杨开振等. Java EE 互联网轻量级框架整合开发一. 北京. 电子工业出版社, 2017. 7