

课程设计

个人报告

课程名称： 软件系统课程设计

设计名称： 学生公寓管理系统

专业班级：软件1933班 学号： 2019006330

学生姓名： 杜枝梅

组长姓名： 曹锦荣

指导教师： 王丽娟

2022年 01 月14 日

太原理工大学课程设计任务书

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生姓名 | 杜枝梅 | 专业班级 | 软件1933 | |
| 课程名称 | 软件系统课程设计 | | | |
| 设计名称 | 公寓管理系统 | | 设计周数 | 2 |
| 设计  任务  主要  设计  内容 | 1、学生信息管理需要支持录入、修改、查找、删除等功能；  2、宿舍信息管理可以处理宿舍的基本信息以及宿舍管理员可以对宿舍进修增删改查等功能；   1. 卫生检查管理由宿舍管理员完成宿舍卫生信息的添加和修改； 2. 来访登记主要完成登记来访人员和被访学生信息登记；   5、保修系统主要完成宿舍硬件设施的管理，学生可以查看受理进度等。  6、用户管理主要是系统管理员对本系统的维护，对系统用户的维护以及相应的授权和取缔等操作。 | | | |
| 设计要求 | 课程设计的实验内容包括结构化（生命周期法）的方法学和面向对象的方法学。通过一个模拟项目，要求学生用结构化方法（或面向对象的方法）完成分析、设计、编码实现和测试的整个软件开发过程。应遵循“GB/T 8567 2006 计算机软件文档编制规范”，提高课程设计教学质量。 | | | |
| 主要参考  资 料 | 李爱萍，崔冬华，李东生.软件工程[M].人民邮电出版社.2014  李爱萍.系统分析与设计[M].人民邮电出版社.2015  朱少民.软件测试[M].人民邮电出版社.2016  黑马程序员.JavaWeb程序设计任务教程[M].人民邮电出版社.2017 | | | |
| 学生提交  归档文件 | 1. 项目小组必须提交一份完整的项目技术说明文档（纸质版和电子版），文档内容必须包含《需求分析说明书》、《概要设计说明书》、《详细设计说明书》、《源程序》、《测试计划》、《测试报告》与《项目开发计划》等部分； 2. 每一位同学应提交一份开发日志，详细记录课程设计过程中出现的问题、解决方案及课程设计心得。 | | | |

注：1.课程设计完成后，学生提交的归档文件应按照：封面—任务书—说明书—图纸的顺序进行装订上交（大张图纸不必装订）。

2.可根据实际内容需要续表，但应保持原格式不变。

**指导教师签名**： **日期**：2022.1.4

**目录**

[第一章 概述 1](#_Toc19218)

[第二章 需求分析 2](#_Toc18890)

[2.1. 引言 2](#_Toc28490)

[2.1.1. 目的 2](#_Toc18818)

[2.1.2. 背景 2](#_Toc2980)

[2.1.3. 参考资料 2](#_Toc16)

[2.2. 项目概述 3](#_Toc10946)

[2.2.1. 产品描述 3](#_Toc10617)

[2.3. 产品需求 3](#_Toc11482)

[2.3.1. 卫生检查管理模块 3](#_Toc28759)

[2.3.2. 系统维护模块 5](#_Toc6258)

[第三章 总体设计 7](#_Toc30241)

[3.1. 部署图 7](#_Toc4608)

[3.2. 数据库设计 7](#_Toc15052)

[3.2.1. 卫生检查模块概念结构设计 7](#_Toc14749)

[3.2.2. 系统维护概念结构设计 8](#_Toc17978)

[第四章 详细设计 9](#_Toc26406)

[4.1. 数据库设计 9](#_Toc19023)

[4.1.1. 创建数据库 9](#_Toc16506)

[4.2. 核心架构 10](#_Toc27316)

[第五章 源程序 12](#_Toc17093)

[5.1. 卫生检查模块 12](#_Toc32642)

[5.2. 系统维护模块 14](#_Toc24303)

[第六章 测试报告 16](#_Toc4714)

[第七章 个人开发日志 18](#_Toc2725)

[第八章 个人总结与心得体会 24](#_Toc4640)

# 概述

**一、所选题目：**

太原理工大学公寓管理系统

**二、需求分析：**

用JAVA EE完成实验，宿舍模块在管理员权限下可以列出全部卫生检查记录，用以查看卫生检查相关信息。保修模块在普通用户权限下列出全部维修情况记录。其中：

卫生检查管理由公寓管理员完成宿舍卫生信息的添加和修改；

保修系统主要完成宿舍硬件设施的管理，学生可以查看受理进度和状态等；

此外，在程序中还设计了用户登陆，并设定了密码，并且管理员权限用户对于数据库中数据的存储方式上采取增删改查，以确保本系统应用的安全性。

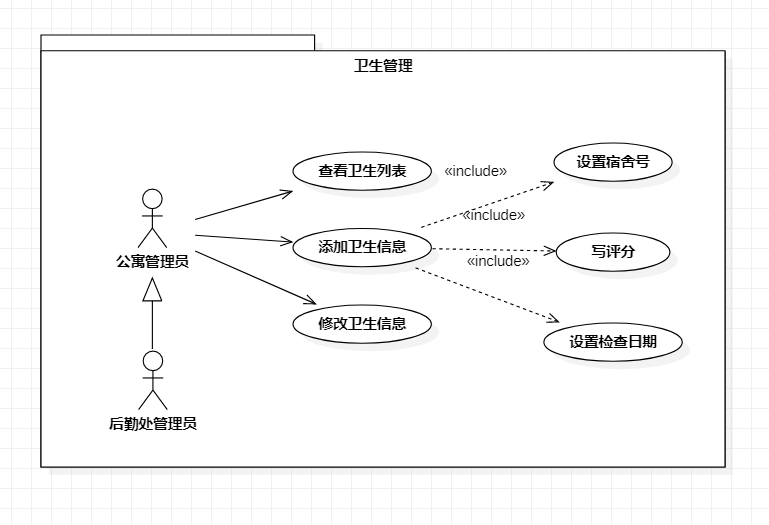
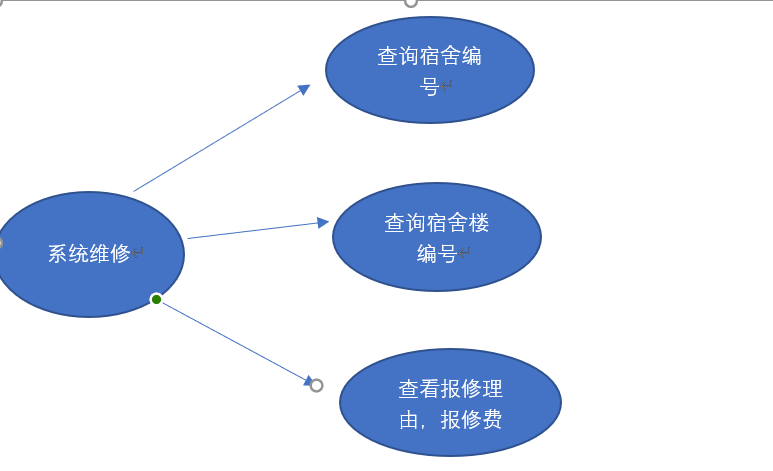
其中卫生管理模块用例图如下：

图2-1 卫生管理模块用例图

系统维修模块用例图如下：

图 2-2 维修系统用例图

**三、小组分工：**

根据所选题目要完成的功能，工作量估计需要10\*4人天，确定开发小组由4人组成分工如表1-1小组工作分配表所示。

表1-1 小组工作分配表

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **姓名** | **角色** | **任务** |
| 01 | 曹锦荣 | 组长 | 需求分析与总体设计 学生及系统管理模块 |
| 02 | 杜枝梅 | 组员 | 寝室卫生检查模块 财产管理模块 |
| 03 | 骈宇静 | 组员 | 寝室管理模块 学生出入及访客管理模块 |
| 04 | 康欣怡 | 组员 | 班级管理模块 集成及系统测试 |

# 需求分析

## 引言

### 目的

此需求规格说明书对学生公寓管理系统软件进行了全面细致的用户分析，详细的陈述了系统的各项功能，其中包括用户的功能性需求和非功能性需求，提供了完整且较详细的系统功能运作蓝图，清楚的表达客户提出的要求。让开发人员与客户就最后的软件做成什么样子达成一致。同时为开发人员提供了一个可靠地设计约束，以便高质量的完成开发任务。

### 背景

为了方便高校对宿舍的管理，就出现了高校电子宿舍管理系统，通过系统来实现宿舍日常管理的和种操作，减少了人力、物力，提高了办事效率，是现代人类不可缺少的资源。宿舍管理是一项琐碎、复杂而又十分细致的工作。如果采用人工管理则需要庞大的人力资料，而且不能避免工作中一些小错误，甚至一旦出现了失误，就会影响整个管理工作的进度。如果采用科学的方法来管理，则可以有效的避免这种错误的产生，管理更方便，快捷，通过该系统，用户可以随时随地的对现有档案进行查阅、添加、修改或删除等操作，具有传统方式所无法比拟的高效率。

本系统主要就太原理工大学宿舍管理设计，主要完成学生信息管理、宿舍信息管理、卫生检查管理、来访登记、保修系统、用户管理等功能。具备一般的宿舍管理要求，方便信息存储管理，方便信息查阅和维护。

### 参考资料

参考书籍及文章：

李爱萍，崔冬华，李东生.软件工程[M].人民邮电出版社.2014

贾铁军，徐方勤.数据库原理及应用[M].机械工业出版社.2017

李爱萍.系统分析与设计[M].人民邮电出版社.2015

朱少民.软件测试[M].人民邮电出版社.2016

黑马程序员.JavaEE程序设计任务教程[M].人民邮电出版社.201

### 产品描述

用JAVA EE构建的学生公寓管理系统，能实现学生信息管理、宿舍信息管理、卫生检查管理、来访登记、保修系统和用户管理。其中学生信息管理需要支持录入、修改、查找、删除等功能；宿舍信息管理应该可以处理宿舍的基本信息以及宿舍管理员可以对宿舍进修增删改查等功能；卫生检查管理由宿舍管理员完成宿舍卫生信息的添加和修改；来访登记主要完成登记来访人员和被访学生信息登记；保修系统主要完成宿舍硬件设施的管理，学生可以查看受理进度和状态等；用户管理主要是系统管理员对本系统的维护，对系统用户的维护以及相应的授权和取缔等操作。

此外，在程序中还设计了用户登陆，并设定了密码，并且分权限设置用户，不同的用户权限可以对系统同进行不同层次的操作与查询，在对于数据库中数据的存储方式上采取加密的程序，以确保本系统应用的安全性。

本人负责卫生检查管理及系统维护两个模块的设计和实现。

### 卫生检查管理模块

#### 需求定义

卫生检查工作添加宿舍信息，删除宿舍信息，修改学生信息，宿舍人员信息查询

是日常公寓管理中必不可少的环节，卫生检查涉及检查时间、评分、宿舍排名等信息。该模块通过权限约束，对学生用户只开放卫生列表的查看权限，对于公寓管理员则可进行宿舍评分等功能。如上图2-1卫生管理用例图所示。

#### 卫生管理模块业务流程图

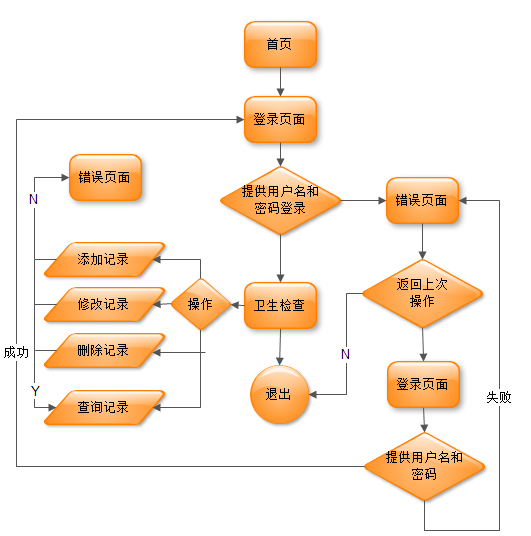


图2-2 卫生管理模块业务流程图

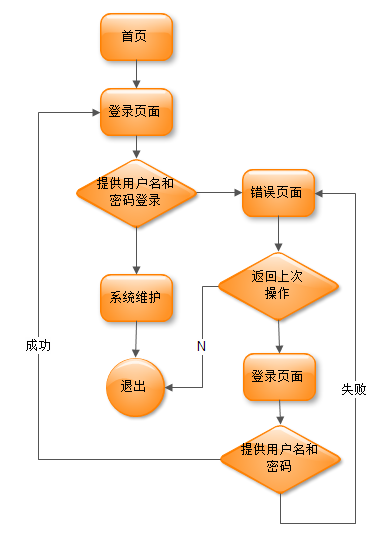
### 系统维护模块

#### 需求定义

本模块分为添加、删除两个界面对管理员进行显示操作，在不同的界面可进行不同的操作。在本界面中添加、删除操作会更新到数据库的相应表中，实现了数据的维护。 当登录用户是普通用户时，无法访问到该页面；当登录用户是管理员时，会自动进入系统维护模块。采用前后台共页面通过权限处理业务逻辑，实现架构重用、代码重用和组件重用。如上图2-2系统用例图所示。

本模块属于系统管理权限分配模块，当登录用户为普通职工时，访问系统管理模块将会出现无权限错误；当登录用户为管理员时，将继续进行授权处理程序。这系统管理模块，可以对管理员和普通进行管理，也可以管理登录用户授权以及取消权限功能。

#### 流程图

图2-4 系统维护模块业务流程图

# 详细设计和实现

## 架构

三层架构(3-tier architecture) 通常意义上的三层架构就是将整个业务应用划分为：表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）。区分层次的目的即为了“高内聚，低耦合”的思想。

在软件体系架构设计中，分层式结构是最常见，也是最重要的一种结构，从下至上分别为：数据访问层、业务逻辑层（又或成为领域层）、表示层，具体描述如下：

表示层位于最外层（最上层），离用户最近。用于显示数据和接收用户输入的数据，为用户提供一种交互式操作的界面。

业务逻辑层在体系架构中的位置很关键，它处于数据访问层与表示层中间，起到了数据交换中承上启下的作用。业务逻辑层的设计对于一个支持可扩展的架构尤为关键，因为它扮演了两个不同的角色。对于数据访问层而言，它是调用者；对于表示层而言，它却是被调用者。

数据层访问层：有时候也称为是持久层，其功能主要是负责数据库的访问，可以访问数据库系统、二进制文件、文本文档或是XML文档。 简单的说法就是实现对数据表的Select，Insert，Update，Delete的操作。如果要加入ORM的元素，那么就会包括对象和数据表之间的mapping，以及对象实体的持久化。

## 流程图

**卫生管理模块流程图如下：**

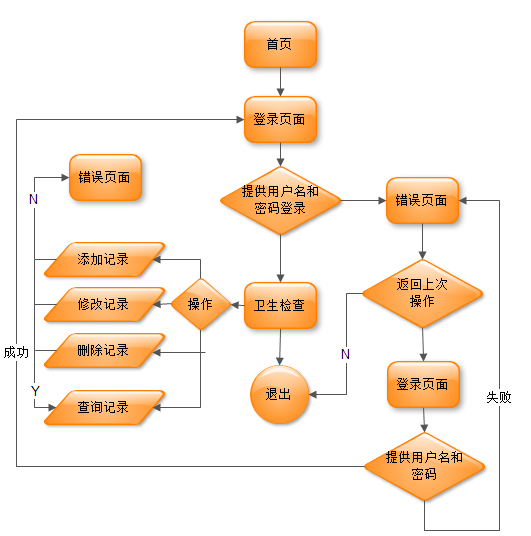


图3-1 卫生管理模块流程图

**维修流程图如下所示：**

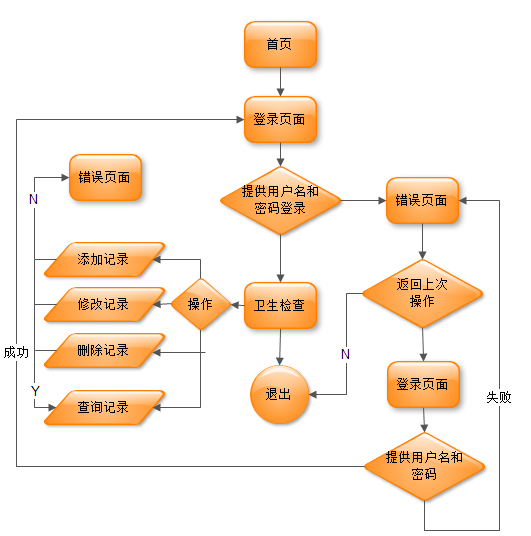


图3-2 维修流程图

## 数据库设计

本系统使用免费开源且轻巧的数据库MySQL。MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL的SQL语言是用于访问数据库的最常用标准化语言。MySQL软件采用了双授权政策（本词条“授权政策”），它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择MySQL作为网站数据库。由于其社区版的性能卓越，搭配PHP和Apache可组成良好的开发环境。

本系统中选用MySql自动集成的命令行工具，也可以使用其他类似工具或图形界面工具。

### 卫生检查模块数据库设计

卫生检查表（HEALTH）如下：

表3-1卫生检查表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 约束 | 默认值 | 备注 |
| HEALTH\_ID | **VARCHAR**(100) | **PK** |  | 系统流水号 |
| SERIAL\_NUMBER | **VARCHAR**(100) |  |  | 宿舍号 |
| SCORE | **VARCHAR**(100) |  |  | 得分 |
| CHECK\_DATE | DATETIME |  |  | 检查时间 |
| MODIFIED\_DATE | DATETIME |  |  | 更新时间 |

参考代码：

**CREATE** **TABLE** HEALTH

(

HEALTH\_ID **VARCHAR**(20),

SERIAL\_NUMBER **VARCHAR**(100),

SCORE **VARCHAR**(100),

CHECK\_DATE DATETIME,

MODIFIED\_DATE DATETIME,

**PRIMARY** **KEY** (HEALTH\_ID)

);

### 维修检查模块数据库设计

维修情况表（REPAIR）如下：

表3-2维修情况表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 约束 | 默认值 | 备注 |
| REPAIR\_ID | **VARCHAR**(20) | **PK** |  | 系统流水号 |
| SERIAL\_NUMBER | **VARCHAR**(100) |  |  | 宿舍号 |
| PHONE | **VARCHAR**(100) |  |  | 联系电话 |
| REPAIR\_TIME | **VARCHAR**(20) |  |  | 维修情况 |
| GOODS | **VARCHAR**(1000) |  |  | 维修物资 |
| REASON | **VARCHAR**(2000) |  |  | 相关人员 |
| STATUS | **VARCHAR**(2000) |  |  | 维修状态 |
| MODIFIED\_DATE | DATETIME |  |  | 来访时间 |

参考代码：

**CREATE** **TABLE** REPAIR

(

REPAIR\_ID **VARCHAR**(20),

SERIAL\_NUMBER **VARCHAR**(100),

PHONE **VARCHAR**(100),

REPAIR\_TIME **VARCHAR**(20),

GOODS **VARCHAR**(1000),

REASON **VARCHAR**(2000),

STATUS **VARCHAR**(2000),

MODIFIED\_DATE DATETIME,

**PRIMARY** **KEY** (REPAIR\_ID)

);

## 核心代码

## 3.4.1卫生检查模块

卫生检查工作添加宿舍信息，删除宿舍信息，修改学生信息，宿舍人员信息查询

是日常公寓管理中必不可少的环节，卫生检查涉及检查时间、评分、宿舍排名等信息。该模块通过权限约束，对学生用户只开放卫生列表的查看权限，对于公寓管理员则可进行宿舍评分等功能。图3-3 宿舍列表图如下。

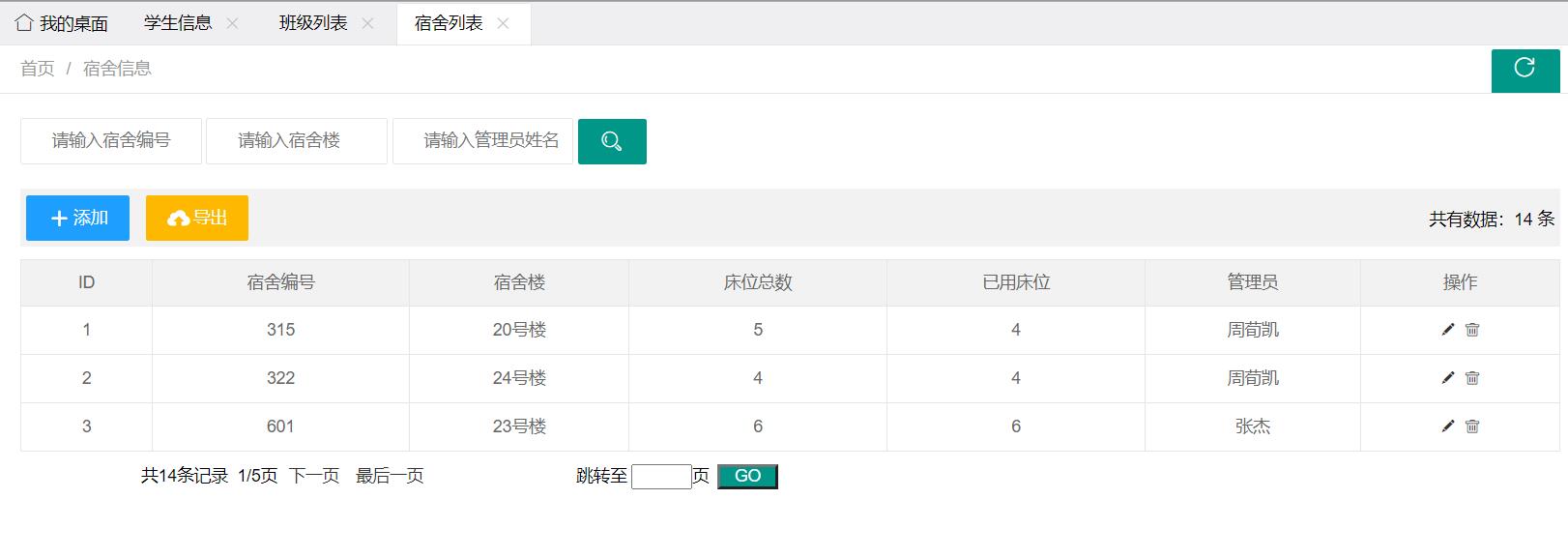


图3-3 宿舍列表图

添加宿舍图如下所示：

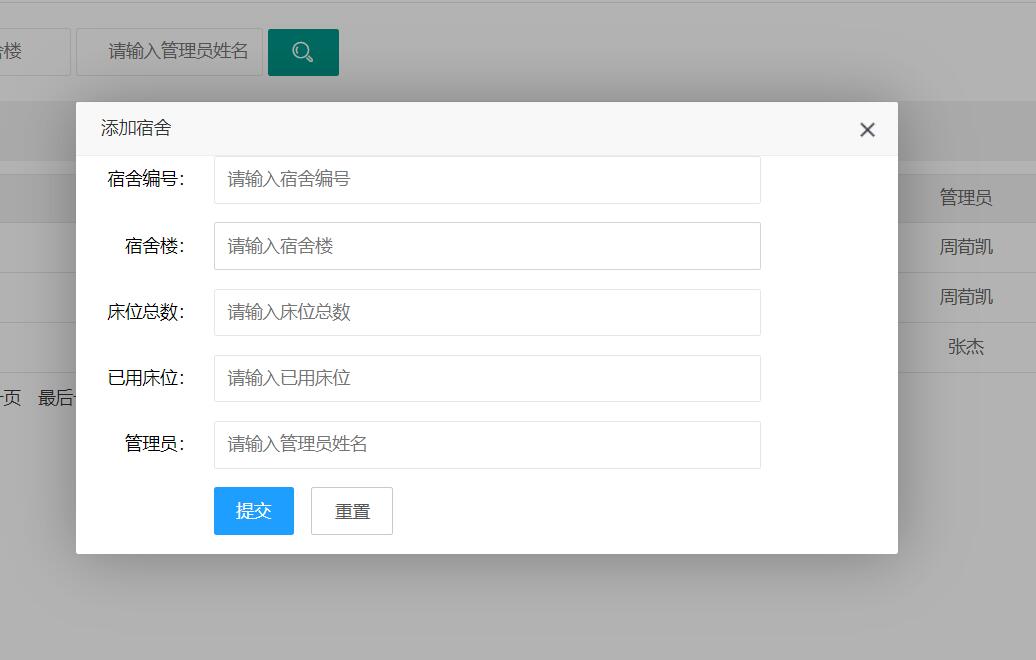
****

图3-4 添加宿舍图

**关键代码：**

@Controller  
**public class** DormCleanController {@Autowired  
 **private** DormCleanService **dormCleanService**;  
@RequestMapping(value = **"/findDormClean"**)  
 **public** String findDormClean(Integer d\_id,String d\_dormbuilding,  
 Integer pageIndex, Integer pageSize, Model model) {  
  
 PageInfo<DormClean> di = **dormCleanService**.findPageInfo(d\_id,d\_dormbuilding,  
 pageIndex,pageSize);  
 model.addAttribute(**"di"**,di);  
 **return "dormclean\_list"**;  
 }@RequestMapping(value = **"/exportdormcleanlist"**, method = RequestMethod.***POST***)  
 @ResponseBody  
 **public** List<DormClean> exportDormclean(){  
 List<DormClean> dormclean = **dormCleanService**.getAll();  
 **return** dormclean;  
 }@RequestMapping(value = **"/addDormClean"** ,method = RequestMethod.***POST***)  
 @ResponseBody  
 **public** String addDormClean( @RequestBody DormClean dormclean) {  
 **int** d = **dormCleanService**.addDormClean(dormclean);  
 **return "dormclean\_list"**;  
 }@RequestMapping( **"/deleteDormClean"**)  
 @ResponseBody  
 **public** String deleteDormClean(Integer g\_id) {  
 **int** d = **dormCleanService**.deleteDormClean(g\_id);  
 **return "dormclean\_list"**;  
 }@RequestMapping( **"/updateDormClean"**)  
 **public** String updateDormClean( DormClean dormclean) {  
 **int** d = **dormCleanService**.updateDormClean(dormclean);  
 **return "redirect:/findDormClean"**;  
 }  
 @RequestMapping( **"/findDormCleanById"**)  
 **public** String findDormCleanById(Integer g\_id, HttpSession session) {  
 DormClean d= **dormCleanService**.findDormCleanById(g\_id);  
 session.setAttribute(**"d"**,d);  
 **return "dormclean\_edit"**;  
 }

## 系统维护模块

本系统的后勤处管理员具备添加管理员的权限，因此普通管理员即公寓管理员登录系统仅可以对系统进行基础的维护，比如添加、修改和删除数据。当后勤处管理员登录系统维护模块，具有该系统一切权限，可以完成授权和移除权限的功能。当点击某个用户时，自动填充到目标授权文本框里，在选择权限设置后，点击立即授权按钮，即可完成授权操作。

**主要的界面截图如下**：



图3-5 维修信息表

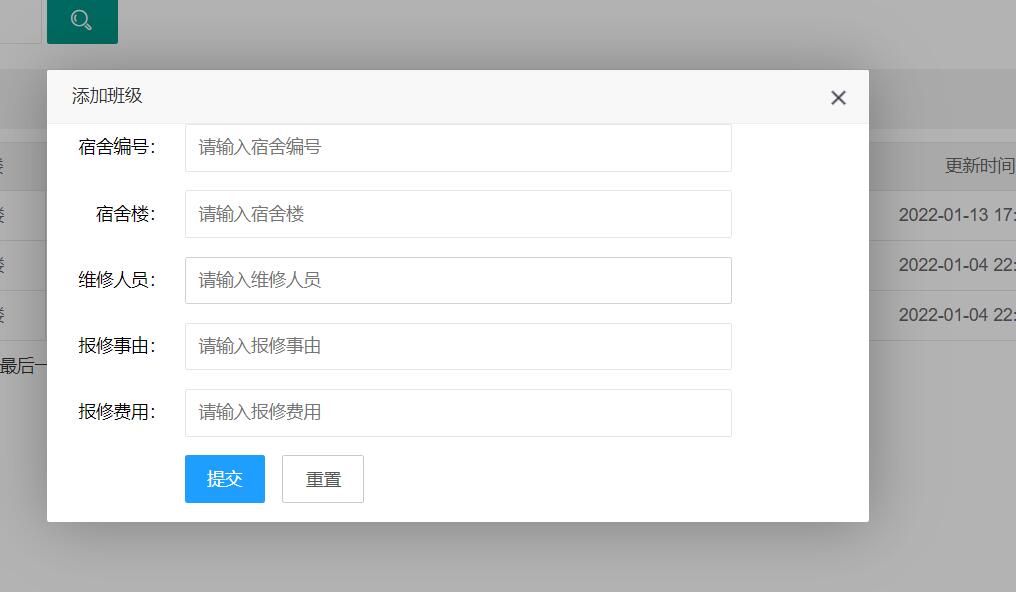


图3-6 维修信息登记表

**主要代码：**

@RequestMapping(value = **"/findDormRepair"**)  
**public** String findDormRepair(Integer d\_id,String d\_dormbuilding,  
 Integer pageIndex, Integer pageSize, Model model) {  
 PageInfo<DormRepair> di = **dormRepairService**.findPageInfo(d\_id,d\_dormbuilding,  
 pageIndex,pageSize);  
 model.addAttribute(**"di"**,di);  
 **return "dormrepair\_list"**;  
}@RequestMapping(value = **"/exportdormrepairlist"**, method = RequestMethod.***POST***)  
@ResponseBody  
**public** List<DormRepair> exportDormrepair(){  
 List<DormRepair> dormRepairList = **dormRepairService**.getAll();  
 **return** dormRepairList;  
}  
@RequestMapping(value = **"/addDormRepair"** ,method = RequestMethod.***POST***)  
@ResponseBody  
**public** String addDormitory( @RequestBody DormRepair dormrepair) {  
 **int** d = **dormRepairService**.addDormRepair(dormrepair);  
 **return "dormrepair\_list"**;  
}

@RequestMapping( **"/deleteDormRepair"**)  
@ResponseBody  
**public** String deleteDormRepair(Integer r\_id) {  
 **int** d = **dormRepairService**.deleteDormRepair(r\_id);  
 **return "dormrepair\_list"**;  
}  
@RequestMapping( **"/updateDormRepair"**)  
**public** String updateDormRepair( DormRepair dormrepair) {  
 **int** d = **dormRepairService**.updateDormRepair(dormrepair);  
 System.***out***.println(dormrepair);  
 **return "redirect:/findDormRepair"**;  
}  
@RequestMapping( **"/findDormRepairById"**)  
**public** String findDormRepairById(Integer r\_id, HttpSession session) {  
  
 DormRepair d= **dormRepairService**.findDormRepairById(r\_id);  
 session.setAttribute(**"d"**,d);  
 **return "dormrepair\_edit"**;  
}

# 测试报告

有两种基本方法可以用来标识测试用例，即功能测试（黑盒测试）和结构测试（白盒测试）。本次测试主要采用了功能测试。功能测试的基本观点是，任何程序都可以看作是将输入定义域取值映射到输出值域的函数。对于每个模块的各个功能，依次操作，如果出现错误就修复之后再测试。对于任何基于数据库操作的都要测试数据库连接是否成功，对于添加和修改数据的操作都要测试合法数据库和非合法数据，对于删除功能测试删除是否成功即可。

本系统经过详细测试分析，系统完整且运行正常，暂无错误出现。

本人主要负责的系统维护即权限管理模块，权限测试均包含在每一功能测试中，现做如下汇报：

表6.1 宿舍情况功能测试

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试结果 | |
| 非管理员权限 | 失败 | |
| 管理员或超级管理员权限 | 添加宿舍情况 | 成功 |
| 修改宿舍情况 | 成功 |
| 查询宿舍情况 | 成功 |

表6.2 来访登记功能测试

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试结果 | |
| 非管理员权限 | 失败 | |
| 管理员或超级管理员权限 | 添加记录 | 成功 |
| 修改记录 | 成功 |
| 查询记录 | 成功 |

表6.3 卫生检查功能测试

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试结果 | |
| 非管理员权限 | 失败 | |
| 管理员或超级管理员权限 | 添加宿舍情况 | 成功 |
| 修改宿舍情况 | 成功 |
| 查询宿舍情况 | 成功 |

表6.4 维修情况功能测试

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试结果 | |
| 非管理员权限 | 失败 | |
| 管理员或超级管理员权限 | 添加维修记录 | 成功 |
| 修改维修记录 | 成功 |
| 查询维修记录 | 成功 |

系统维护功能测试，如表6.6所示：

表6.6 宿舍情况功能测试

|  |  |  |
| --- | --- | --- |
| 测试用例 | 测试结果 | |
| 非管理员权限 | 失败 | |
| 公寓管理员 | 删除用户 | 成功 |
| 授权 | 无权限 |
| 后勤处管理员 | 删除用户 | 成功 |
| 授权 | 成功 |
| 取消权限 | 成功 |

通过对本系统的各个功能的测试，发现并修改了一些编程过程中无法察觉的错误。经重新修改，最终完成了系统的主要功能实施。

# 个人开发日志

|  |
| --- |
| **2022年1月3日 星期一** |
| 随着老师对课设的清晰讲解与指导之后，大三学年上半学期的最后一门课程也拉开了崭新的帷幕。  我们在课设开始之前已经成立了项目小组，小组成员分别为：曹锦荣，杜枝梅，康欣怡，骈宇静，小组组长是曹锦荣。虽然四个女生实战经验没有那么丰富，但是我相信在组长的带领下我们组的成员会顺利并完美的完成此次课设任务。  随后我们便进入了课程的选题阶段，经讨论我们选择了“学生公寓管理系统”作为我们的项目课题。在指导书要求下，结合组员在学生会以及勤工助学的实际生活经验，我们考虑了更加全面的功能，并且得保证小组内的每一位成员都有相对较为均匀的任务和工作量。最后我们讨论出了我们所要做的这个系统预实现的七大功能，分别为：系统维护模块、卫生模块、来访登记模块、卫生检查模块。并且我们一起讨论了各个子功能的大致目的以及实现愿景等。  总之，第一天的工作还是很开心的完成了，虽然并没有进行实际的开发过程，但是我们对项目的认识都有了较为初步的了解。 |
|  |
| **2022年1月4日 星期二** |
| 课设第二天，在昨日的讨论的基础上，我们开始了今天的项目开发，在项目开发之前，我们小组共同讨论决定了本次项目的开发计划，并制定了开发时间表，要求大家在课设的前五天要完成自己实现的子系统功能，计划在课设的后五天进行项目的整体设计、编码和测试。然后我们开始了项目子系统功能的划分。  因为本学期课程对《系统分析与设计》、《软件测试》和《Web程序设计》等课程的学习，故我们小组成员一致决定采用“面向对象的方法”按照软件工程的方法完成本次项目的开发。因为需求分析在一个项目开发过程起着至关重要的作用，需求分析的好坏往往决定了系统开发的成败，故我们组对于需求分析的重视程度格外的高。  于是今天的工作开始了，大家开始进行自己子系统的需求分析阶段，而每个子系统之间的联系程度密切，所以大家在自己完成自己需求分析的文字描述的同时还要和小组成员进行讨论，最终今天虽然完成的工作量挺少的，但是只有准确确认需求，才能更好的开展后续的工作，而且通过今天大家的进一步讨论，我们的关系更加融洽了，而我也更加有信心能和大家共同完成本次的项目开发。 |
|  |
| **2022年1月5日 星期三** |
| 在昨日需求分析的基础上，为了更加完善的完成整个项目的开发，故今天大家的目标是首先完成用例图的设计，确认需求是否完整，是否能完成各个子功能的实现。在进行用例图设计的过程中，大家遇到了一个严重的问题，就是各个用例图中同类参与者名称的统一问题，故经过大家商讨后，修改了需求分析的参与者的统一命名。为了更好的完成用例图的绘制，在画用例图的过程中，我对卫生检查模块的一个用例进行了用例描述，在用例图设计完成之后，我们对需求分析进一步完善。今天的课程设计中，我们明白在分析阶段的时候一定不要过早的陷入细节的设计当中，以防对总体的把握出现偏颇。 |
|  |
| **2022年1月6日 星期四** |
| 开始从分析阶段转向总体设计阶段，总体设计我们主要进行系统架构设计（即软件系统的模块层次结构）、数据库设计和人机界面设计。 |
|  |
| **2022年1月7日 星期五** |
| 今天开始了编码阶段，我们为了集成测试的时候方便，减少测试难度，于是我们全部采用了IDEA为开发平台进行开发，分工明确后大家就开始各自的编码。我对卫生检查管理进行编码。  今天先完成了几个jsp页面与界面设计的统一及接口问题。 |
|  |
| **2022年1月10日 星期一** |
| 在今天的编码阶段，困难较大，同时还要与其他成员进行数据库属性等的统一，然后进行了数据库连接 |
| 因为今天的代码设计到连接数据库，出现了较多页面的乱码，最后经过查资料，了解了Struts架构如何解决乱码问题： |
|  |
|  |
|  |
| **2022年1月11日至2022年1月12日** |
| 两天依然是敲代码+处理bug，但是随着验收日期的到来，我不得不加快了脚步。在一个图书系统的框架基础上进行了置换，虽然有些封装没有完全搞懂，但是在与前端设计组员的配合下，完成了子模块的编码工作。  这几天的学习与交流，让我总结出了Struts框架运行原理： |
|  |
|  |
| **2022年1月13日 星期四** |
| 昨天还完成了集成，我们在集成测试阶段出现了较大的问题，各种页面跳转不成功，不同浏览器对于不同页面的解析不同，各种字体颜色以及页面格式，还有数据库连接查询不出来，页面无反映等的问题一一暴露出来。大家的情绪也都相当低落。但是，还是要努力去克服，今天我们分开查找问题，集成到谁的子系统时，谁进行解释说明，并且通过协调大家对有些文件名进行了重命名，还有数据库里面的各个主键外码的设置，以及通过网上查阅，对css样式的重新修改，最后统一了管理员界面的背景，使整个系统看起来使清晰明了整洁。在集成完以后根据大家各自的测试计划，我们进行了测试，并截图保留，完成了测试报告的绘制。  晚上大家开始整理自己的个人开发文档。交给组长进行汇总。然后期待明天的答辩。 |
|  |

# 个人总结与心得体会

两个礼拜来忙碌紧张而又有条不紊的课程设计，使我有机会对本专业的基本理论、专业知识和基本技术有了更深入的了解和体会，使我在本学期尤其是《系统分析与设计》、《软件测试》和《Web程序设计》等课程的学习中所学到的知识得到了系统和升华，真正达到了学以致用。

对信息管理系统的体会信息管理系统是一门融管理科学、信息科学、系统科学、计算机科学与现代通信技术为一体的一门综合性边缘学科。它是运用系统的方法以计算机和现代通信技术为基本信息处理手段和工具的，能为管理决策提供信息服务的人—机系统.它可以实现数据处理功能、预测功能、计划功能、控制功能和辅助决策功能。管理信息系统的开发是一项复杂的系统工程，必须严格的按照系统规划、系统分析、系统设计、系统实施、系统运行与评价的开发步骤来进行。在系统开发之前，必须了解该系统的特点、适用范围以及使用者需要一个什么样的系统，以此作为基础为开发系统准确定位，然后对使用者所需实现的功能进行分析总结，根据使用者的实际要求来给系统设计一个初步方案。系统的开发不仅是要实现对数据处理的及时与正确，而且要考虑系统是否具有控制功能，及时将数据反馈给操作者，以进行监测和协调，保证系统的正常运行；也要考虑是否具有预测功能，运用一定的数学方法利用历史的数据对未来进行预测的工作。

设计实践上的体会在设计的过程中，我掌握了很多JAVA的编程知识，并对这种成熟并广泛应用的技术进行了深入的学习。设计的过程也是一个再学习的过程，在遇到问题的时候我尽量自己想办法解决，这在很大程度上激发了我们的自学能力；在没有办法解决的情况下，认真的向老师请教，从老师那里我学到了很多的知识，老师对我的指导起到了画龙点睛的作用。

我在设计中经常出现一些问题不知该如何解决，在此时许多同学给予了我们帮助。我首先学到的就是团队合作以及沟通。因为如果团队内部不能很好的进行沟通与合作，就会导致后期集成出现较大问题以及页面整体美观程度；其次我对开发一个完整的项目有了深度的了解，即建立在软件工程的思想上，按照软件工程开发的声明周期来完成整个系统的开发。其中切记不可忽视需求分析，以及按照面向对象还是结构化设计来完成，不可混淆使用，导致项目出现混乱性错误。

在我的程序设计过程中，我充分的体会到了“实践出真知”这一点，书本上的知识是不够的，只有把理论与实践相结合才能够真正的学到知识。一个管理信息系统的设计，不可能一步到位，还需要不断的完善和补充。同时，系统中还存在许多问题，有待在日后的使用中发现和解决。编程前的深思熟虑是减少程序调试工作量的重要方法，只有进行充分考虑，才会减少调试过程中的工作量。虽然在开始写程序之前我们做了多的准备工作，但在真正的写程序时仍然发现许多问题，有些问题是分析时的疏漏，有些则是如果不做无论如何也想不到的。

本次课程设计中，我完成了数据库的链接，前期的分析设计，两个子模块的编码以及测试部分。除此之外，让我认识到了自己的不足，平时相对看书多，编程少，虽然能看懂框架的代码，自己写却总有问题，寒假要把自己动手的能力提高起来。成为真正的软件人才。

本次课程设计涉及到了多本课本，例如《软件工程》《软件测试》《系统分析与设计》《java web课程设计》《算法分析与设计》等，其中部分是大学二年级所学。有些已经遗忘，在课程设计的时候又加深的一次印象，使自己的基础更加牢固。尤其是对《java web课程设计》这门课程理解更加深入，了解到了前端和后台的联系和区别，以及各种类和对象，方法以及框架的使用。

最后，感谢方昀老师，感谢组长和我们搭档们！