摘 要

随着电脑的普及与使用，现在的管理也提升了一个档次，渐渐实现了无纸化办公，即从原来的人工记录管理模式转变为电脑一体化管理。高校是科研的阵地，后勤的公寓管理也应该一改传统的人工管理，更加信息化，时代化，节省人力物力，提高效率。基于这一点，开发此学生公寓管理软件。

学生宿舍管理软件，是以高校的管理方式而设计的一种实用型管理系统。本系统最大的特点是通用性、简单操作性，适用于同行业以及一些同类型的企业管理。随着学校寄宿人员的增多，公寓管理人员的负担越来越重，为了让所有公寓管理人员能从繁重的工作中解脱出来，实现无纸化办公；使工作更有条理，更方便，更有效率而开发出这套公寓管理软件。

软件从对学生的分房开始到最后的公寓管理全部电脑化。它主要包括：宿舍、寝室的配置、住宿人员登记、纪律检查、以及对学生住宿以及学生晚归情况查询、修改、删除，可以更快地了解到每个学生的住宿情况,使宿舍的管理效率更高，做的更好！

**目 录**

**1. 学生信息管理系统概述**

1.1 研究背景

1.2 开发意义

**2. 系统需求分析**

2.1 开发环境和软件

2.2 可行性分析

2.3 系统需求分析

2.4 系统设计与功能分析

**3 登陆流程**

3.1 宿管员登陆

3.2 管理员登陆

3.3 系统逻辑结构设计

3.3.1 SQL语句实现

**4. 系统模块详细设计**

4.1 用户登录模块

4.2 各用户操作模块

4.2.1 宿管员操作模块

4.2.2 管理员操作模块

**5. 系统运行与测试**

5.1 宿管员登录

5.2 管理员登录

**6．课程设计总结**

**7. 附录（代码）**

**1. 学生宿舍管理系统概述**

**1.1 研究背景**

学生宿舍管理系统是学校管理的重要工具，是学校不可或缺的部分。随着各大高校的扩招，学生宿舍的住宿管理也日益繁多，宿管员工作繁琐，人工管理的难度加大，显然是不能满足实际的需要，效率也是很低的。并且这种传统的方式存在着很多的弊端，如：保密性差、查询不便、效率低，很难维护和更新等。然而，本系统针对以上缺点能够极大地提高学生宿舍管理的效率,也是科学化、正规化的管理；本系统的开发是为了方便宿管员的管理以及相关修理信息的更新，为学校领导及老师提供直观、及时方便的学生信息，包括晚归、回家登记、寝室报修，实现各类寝室信息管理工作流程的系统化、规范化和自动化，提高迅速反应能力，提供有效的决策支持。

随着时代的发展，计算机技术越来越深入各行各业，为广大的用户提供了更为周到和便捷的服务。目前各行各业广泛使用专用系统，其内容范围跨越了教育科研、文化事业、金融、商业、新闻出版、娱乐、体育等各个领域，其用户群十分庞大，因此，设计开发好一个专用系统对一个机构(或部门)的发展十分重要。近年来，随着用户要求的不断提高及计算机科学的迅速发展，特别是数据库技术的广泛应用，向用户提供的服务将越来越丰富，越来越人性化。  
对于学校而言，学生宿舍管理是必不可少的组成部分。目前仍然存在有学校停留在宿舍管理部门人员手工记录数据的阶段，手工记录对于规模小的学校来说勉强可以接受；但对于学生信息量比较庞大，需要记录存档的数据比较多的高校来说，人工记录是相当麻烦的,而且当查找某条记录时，由于数据量庞大，只能依靠人工去一条一条地查找，这样不但麻烦而且浪费许多时间，工作效率比较低。采用计算机来管理宿舍和学生的信息，大大提高了查询的速度，节约了人力和物力资源，达到了预期的要求。今天，计算机的价格已经十分低廉，性能却有了长足的进步。它已经被应用于许多领域，计算机之所以如此流行的原因主要有以下几个方面：

第一，计算机可以代替人工进行许多繁杂的劳动；

第二，计算机可以节省许多资源；

第三，计算机可以大大的提高人们的工作效率；

第四，计算机可以使敏感文档更加安全;

### **1.2 开发意义**

随着这些年电脑计算机的速度质的提高，成本的下降，IT互联网大众趋势的发展。我们使用电脑的高效率才处理数据信息成为可能。 学生宿舍管理系统的出现，正是管理人员与信息数据，计算机的进入互动时代的体现。友好的人机交互模式，清晰简明的图形界面，高效安全的操作使得我们对成千上万的信息的管理得心应手。通过这个系统,可以做到信息的规范管理,科学统计和快速的查询,从而减少管理方面的工作量｡毋庸置疑,切实有效地把计算机管理引入学校管理中,对于促进学校管理制度,提高学校教学质量与办学水平有着显著意义。

#### **2. 系统需求分析**

## **2.1 开发环境和软件**

（1） 操作系统：Windows 7

（2） 数据库软件：mysql 5.5

（3） Java开发工具：myeclipse10.7

**2.2 可行性分析**

**一、经济的可行性**

主要从对项目的经济上进行分析评价，一方面是支出的费用，包括设备购置费、管理和维护费用、人员工资和培训费等，另一个是取得的收益。这是个超小型的管理系统，从投入的人力，财力与物力来讲是非常之小的，只要一台电脑，一台打印机，这个系统就可以搞起来，考虑到学校里有电脑，现只要购置一台打印机就可以了。从节省人力方面，可以让管理人员从繁与复杂的工作中解脱出来，做更多的工作，可以给宿舍里的管理提高一个层次。

1. **技术上的可行性**

技术上的可行性分析要考虑将来要采用的硬件和软件技术能否满足用户提出的要求（如计算机的容量、速度等）。此外，还要考虑我们的技术水平，数据库设计方面刚开始接触，但也想通过做这个系统使得自己更加了解以及使用数据库方面的知识，同时这个学期也在学习java面向程序设计，所以选择用javaweb做这个系统能结合数据库以及java方面的知识，算是对这个学期学习的两门的课程做了一个系统的加固和应用。再者，在宿舍里生活了两年，对这个管理模试应该比较的熟。相对来说，宿舍管理系统的编写可能比较难，因为没有试过用java程序结合sql语句来使用存储在数据库的数据，但是我相信只要有心去做，就可以达到自己想要的结果。

**三、总结**

通过以上两点的分析，编写或开发这个宿舍管理系统是可行的，只要这个系统能实现对学生、管理员数据的输入、修改、删除、与查询就可以了。如果技术方面可行，也可以按专业、寝室号、籍贯等分类查询，以便学校方面管理人员可以很方便的查询想要的数据，以便进行统一管理，统一安排。

**2.3系统需求分析**

第一部分：调查用户需求

本系统的最终用户为宿舍楼管理员，本宿舍楼的学生。根据我们日常生活中的经验，结合对自己学校宿舍楼管理老师的咨询和对同宿舍楼同学的调查，得出用户的下列实际要求：

一、宿舍楼的基本情况

学生住在宿舍楼中，每栋宿舍楼都会有若干名老师负责本宿舍楼的日常管理。

1、学生的基本信息：

入校时，每位同学都有唯一的学号，并被分配到指定的宿舍楼和指定的宿舍，也会有一个宿舍号，其入校时间就是他的入住时间。另外，为了管理上的方便，同一院系的学生的宿舍一般在一起，相应地会有其所在的院系名称。

# **2.4 系统设计与功能分析**

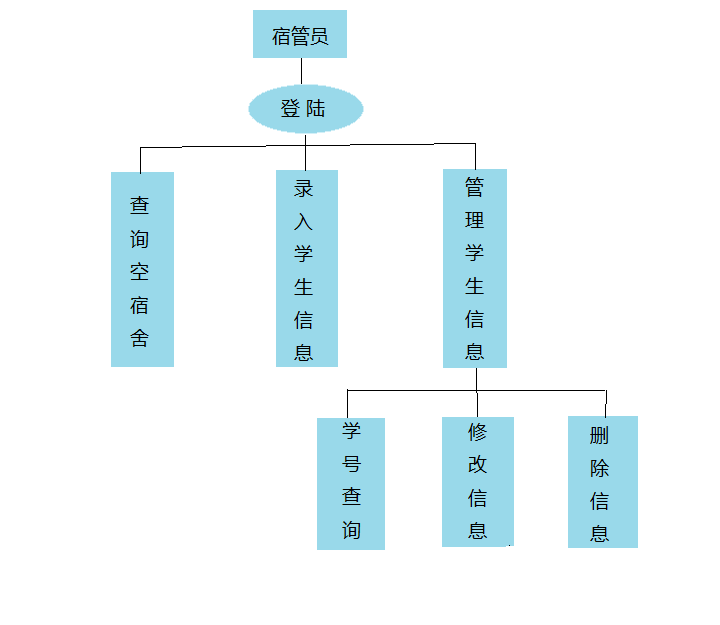
学生宿舍管理系统，主要用于宿管员对学生信息管理，查询，更新与维护，使用方便，易用性强，图形界面清晰明了。该软件用java语言编写，用mysql数据库作为后台的数据库进行信息的存储，用SQL语句完成学生入住宿舍信息的添加，查询，修改，删除的操作以及宿管员的录入，修改，删除等。用jdbc驱动实现前台Java与后台SQL数据库的连接。Java语言跨平台性强，可以在windows，linux等系统下使用，方便简单，安全性好。mysql数据库高效安全，两者结合可相互利用各自的优势。

**该系统实现的大致功能：**

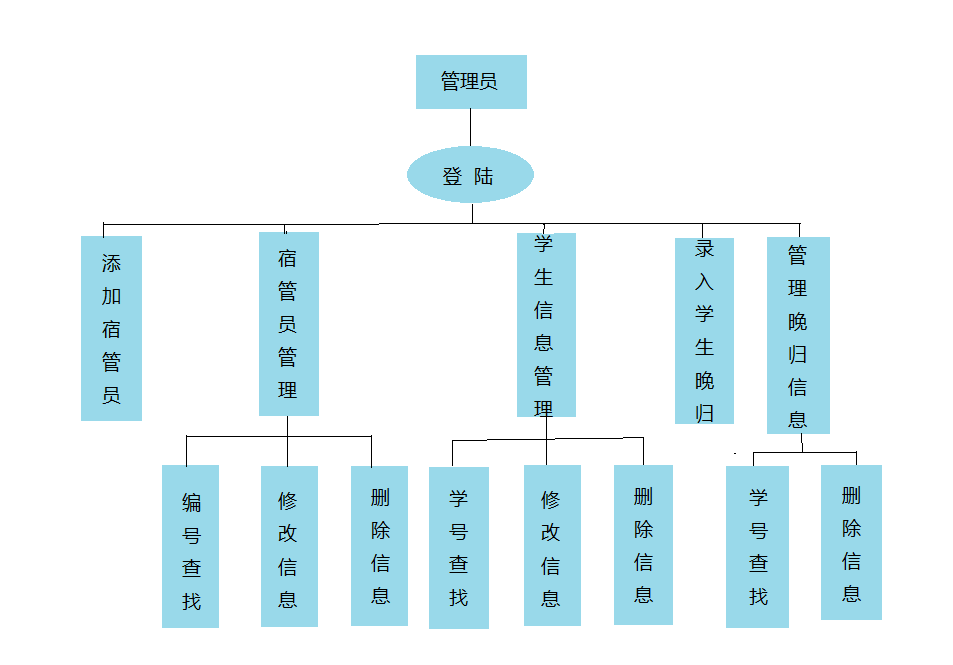
1. 用户登陆界面。该界面可以选择使用者的身份，“宿管员，管理员”。不同的身份有不同的操作界面和功能权限。ID号和密码输入正确即可登录。
2. 宿管员管理界面。提供了对学生入住情况信息的查询，添加，修改，删除；学生信息的录入，修改，删除，查看个宿舍记录以及剩余空床位宿舍以便于添加学生入住宿舍等功能。
3. 管理员管理界面。拥有最高的权限。允许添加和修改、查询、删除宿管员信息等功能。
4. 登录的用户信息分别存储在SQL数据库的“宿管员信息表”，“管理员信息表”中，如果用户信息不存在则三张表中，将会无权利登录本管理系统，保证了本学生管理系统的安全性。

## **3.** **登陆流程**

## **3.1 宿管员登陆：**



**3.2 管理员登陆：**



**实体属性表：**

|  |  |
| --- | --- |
| **实体** | **属性** |
| **学生** | **序号，姓名，学号，性别，学生院系，年级，专业，电话号码，入住时间，离开时间，宿舍楼号， 宿舍房号** |
| **宿舍** | **序号，宿舍楼号 ，宿舍号，收费** |
| **宿管员** | **序号，宿舍楼号，编号，姓名，密码** |
| **管理员** | **管理员编号，姓名，密码 ，职称** |
| **晚归记录** | **晚归日期，宿舍楼号，宿舍房号，晚归姓名** |

## **3.3 系统逻辑结构设计**

**学生表：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型** | **空值** | **约束条件** |
| **序号** | Int | **Not null** | **主键** |
| **姓名** | varchar(30) | **null** |  |
| **学号** | varchar(30) | **null** |  |
| **性别** | varchar(30) | **null** |  |
| **学生院系** | varchar(30) | **null** |  |
| **年级** | varchar(30) | **null** |  |
| **专业** | varchar(30) | **null** |  |
| **电话号码** | varchar(30) | **null** |  |
| **入住时间** | varchar(30) | **null** |  |
| **离开时间** | varchar(30) | **null** |  |
| **宿舍楼号** | varchar(30) | **null** |  |
| **宿舍房号** | varchar(30) | **null** |  |

**宿舍表：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型** | **空值** | **约束条件** |
| **序号** | Int | **Not null** | **主键** |
| **宿舍楼号** | varchar(30) | **null** |  |
| **宿舍号** | varchar(30) | **null** |  |
| **收费** | varchar(30) | **null** |  |

**宿管员表：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型** | **空值** | **约束条件** |
| **序号** | Int | **Not null** | **主键** |
| **宿舍楼号** | varchar(30) | **null** |  |
| **编号** | varchar(30) | **null** |  |
| **姓名** | varchar(30) | **null** |  |
| **密码** | varchar(30) | **null** |  |

**管理员：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型** | **空值** | **约束条件** |
| **编号** | varchar(30) | **Not null** | **主键** |
| **姓名** | varchar(30) | **null** |  |
| **密码** | varchar(30) | **null** |  |
| **职称** | varchar(30) | **null** |  |

**晚归记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型** | **空值** | **约束条件** |
| **学号** | varchar(30) | **Not null** | **主键** |
| **姓名** | varchar(30) | **null** |  |
| **宿舍楼号** | varchar(30) | **null** |  |
| **宿舍房号** | varchar(30) | **null** |  |
| **晚归日期** | varchar(30) | **null** |  |

## **SQL语句实现**

学生信息表：

create table student(

id int primary key auto\_increment,student\_no varchar(30),

student\_name varchar(30),student\_sex varchar(2),

department varchar(30),

grade varchar(30),

professional varchar(30),

phonenumber varchar(30),

int\_time varchar(30),

out\_time varchar(30),b

uilding\_no varchar(30),

dormitory\_no varchar(30));

宿舍表：

create table dormitory(

id int primary key auto\_increment,

building\_no varchar(30),

dormitory\_no varchar(30),

cost varchar(30));

宿管员表：

create table employee(

id int primary key auto\_increment,

building\_no varchar(30) ,

employee\_no varchar(30) ,

employee\_name varchar(30),

employee\_ps varchar(30));

管理员表：

create table admin(

admin\_id varchar(30) primary key,

admin\_name varchar(30) ,

admin\_ps varchar(30),

title varchar(30));

晚归信息表：

create table laterecord(

student\_no varchar(30) primary key,

student\_name varchar(30),

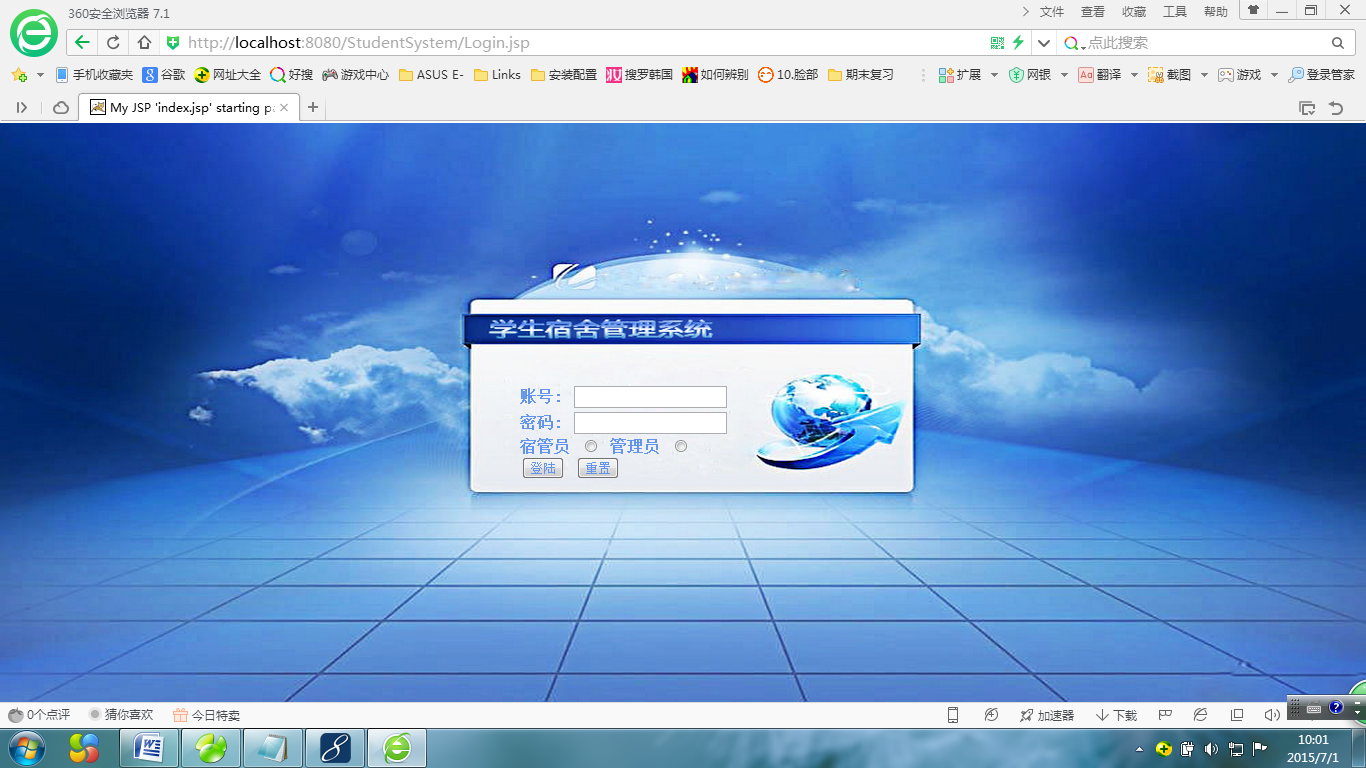
building\_no varchar(30),

dormitory\_no varchar(30),

date varchar(30));

## **4. 系统模块详细设计**

## **用户登录模块**



## **4.2 各用户操作模块**

## **4.2.1 宿管员操作模块（图1）**

## **4.2.2 管理员操作模块（图2）**

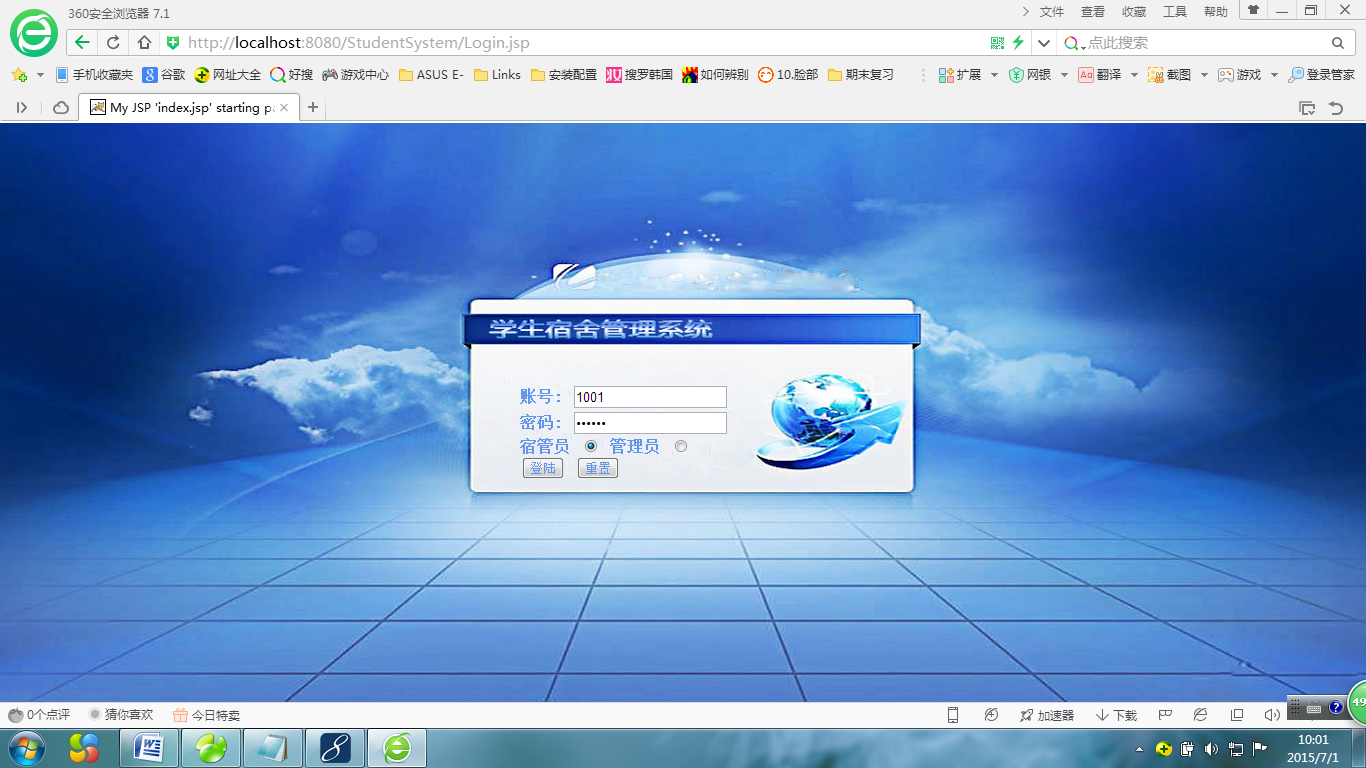
**（图1）：**

**（图2）：**



## **5. 系统运行与测试**

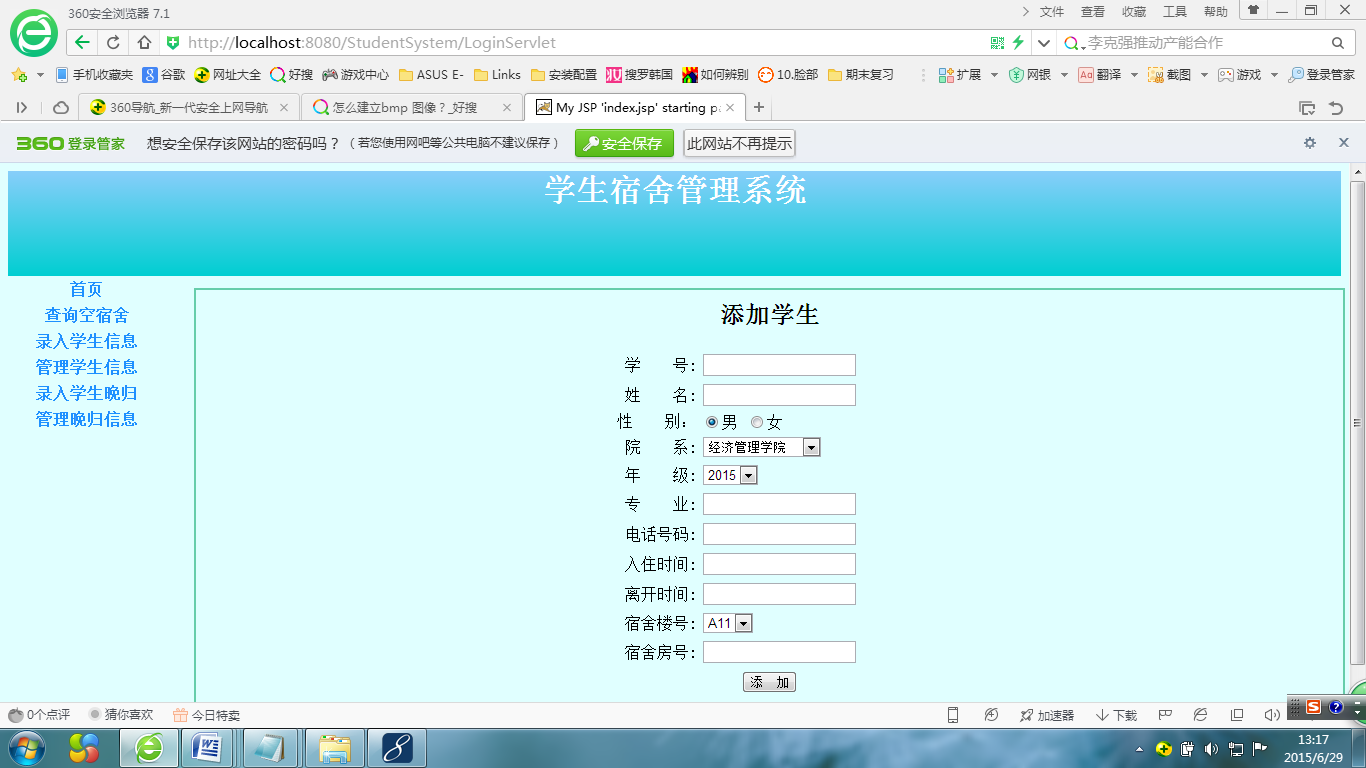
## **5.1 宿管员登录：**



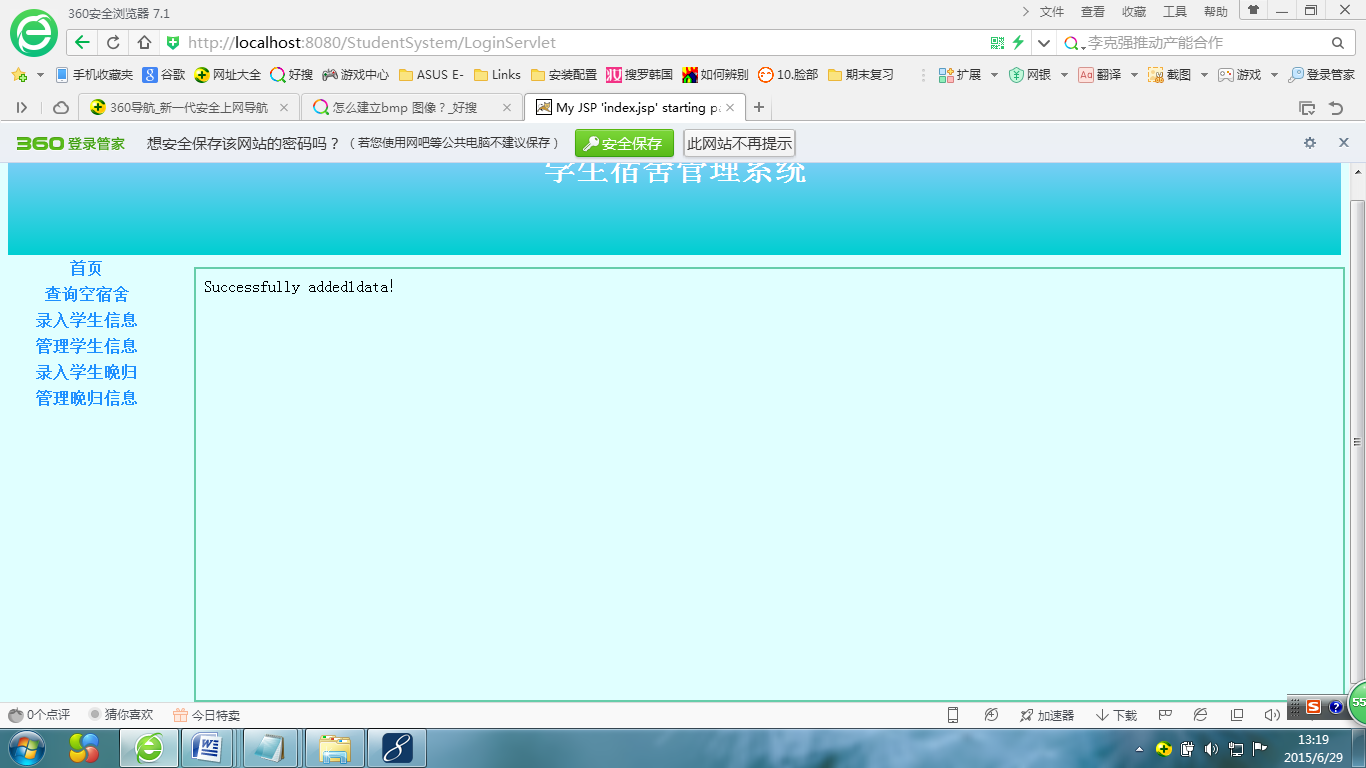
点击“宿管员”按钮。输入正确的ID和密码。验证成功则可进入管理员管理界面。

*宿管员ID号和登录密码存在数据库中的宿管员信息表。表中存在的宿管员才允许登录*。

1. 点击“录入学生信息”。在页面的输入栏中输入正确的数据



点击添加按钮若页面“Successfully added1data！”出现如下图则添加成功，否着页面“Add data failure！”则添加失败；



1. 点击“管理学生信息”，会出现全部学生信息，此时可通过学号准确查找学生信息对学生信息进行修改、删除操作，通过修改学生信息可实现对学生的调宿舍管理；



点击单个学生行后的“修改”，页面自动接收某一学生具体基本信息，在页面的输入栏中输入正确的数据；



1. 点击“录入晚归信息”。在页面的输入栏中输入正确的数据



点击添加按钮若页面“Successfully added1data！”出现如下图则添加成功，否着页面“Add data failure！”则添加失败；

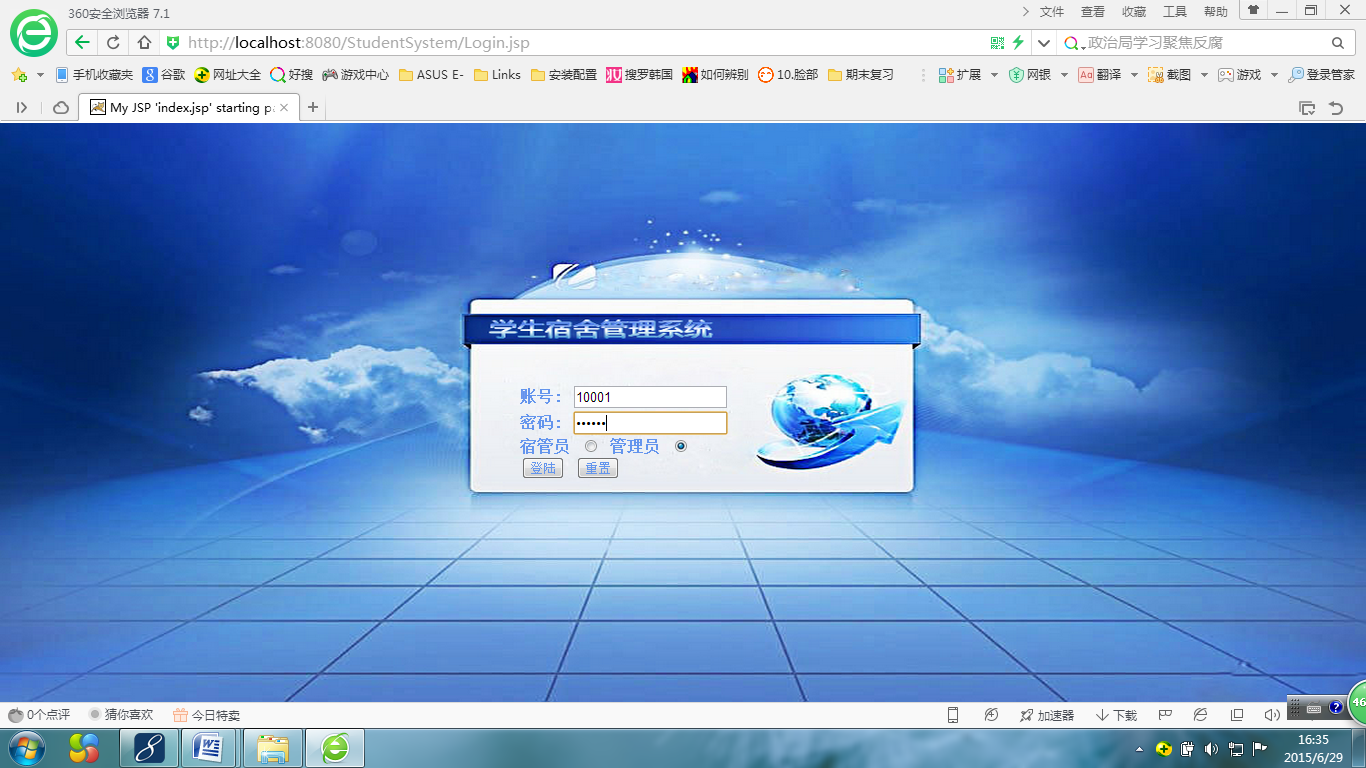
1. 点击“管理晚归信息”，会出现全部学生晚归信息，此时可通过学号准确查找学生信息对学生信息进行删除操作；



## **5.2 管理员登录：**

在登录界面选择“管理员”按钮，输入正确的ID和密码。验证成功则可进入管理员管理界面。

管理员*ID号和登录密码存在数据库中的*管理员*信息表。表中存在的宿管员才允许登录*



登录成功！进入教师管理的操作界面：



点击“添加宿管员”。在页面的输入栏中输入正确的数据；



点击添加按钮若页面“Successfully added1data！”出现如下图则添加成功，否着失败；

点击“宿管员管理”，会出现全部宿管员信息，此时可通过编号准确查找学生信息对管理员信息进行修改、删除操作；



点击管理员行后的“修改”，页面自动接收某一管理员的具体基本信息，在页面的输入栏中输入需要修改的数据；



点击“确定”即可修改成功，修改成功后自动返回管理员管理的界面；管理员的学生信息管理部分功能与宿管员的学生宿舍管理部分相同。

**6．课程设计总结**

课程设计是培养学生综合运用所学知识,发现,提出,分析和解决实际问题,锻炼实践能力的重要环节,是对学生实际工作能力的具体训练和考察过程. 本次课程设计虽然很辛苦，实在是受益匪浅。本来这门课的知识学的就不够扎实，本次课程设计，在设计过程中碰到了很多问题，刚开始要设计的时候，还真不知道从哪里下手，但最终通过去图书馆借书、请教同学以及查找相关视频等途径使得疑问都得到了解决，让我学会了好多书本上没有的东西，通过本次课程设计我也能将课本上的知识融会贯通，起到了很好的辅助学习的效果，而且我发现我学到的知识比整整一个学期学到的都多。理论和实践的相结合是学习最有效的方法。在设计的过程中发现了自己的不足之处，对以前所学过的知识理解得不够深刻，掌握得不够牢固，通过这次课程设计之后，一定把以前所学过的知识重新温故。

通过这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从理论中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。

#### **7. 附录（代码）**

package com.stu.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.stu.dao.EmployeeDao;

import com.stu.entity.Employee;

public class AddEmpServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* Constructor of the object.

\*/

public AddEmpServlet() {

super();

}

/\*\*

\* Destruction of the servlet. <br>

\*/

public void destroy() {

super.destroy(); // Just puts "destroy" string in log

// Put your code here

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("UTF-8");

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

// building\_no employee\_no employee\_name employee\_ps

String building\_no = request.getParameter("building\_no");

String employee\_no = request.getParameter("employee\_no");

String employee\_name = request.getParameter("employee\_name");

String employee\_ps = request.getParameter("employee\_ps");

// 实例化employee

Employee employee=new Employee(building\_no,employee\_no,employee\_name,employee\_ps);

// 实例化employeeDao

EmployeeDao employeedao=new EmployeeDao();

// 添加学生信息

int row = employeedao.saveEmployee(employee);

if(row > 0){

// 更新成输出信息

out.print("Successfully added employee" + row + "data！");

}else{

out.print("Add employee data failure！");

}

out.flush();

out.close();

}

public void init() throws ServletException {

// Put your code here

}

}