

## 目 录

目 录	I
第一章 绪论	2
1.1 系统的目的和意义	2
第二章 宿舍管理系统需求分析	3
2.1 引言	3
2.2 需求分析	3
2.3 界面需求分析	4
2.4 功能需求分析	4
2.5 数据库需求分析	4
第三章 宿舍管理系统结构分析与设计	5
3.1 引言	5
3.2 系统模块结构图	6
3.3 学生注册模块	7
3.4 系统管理员模块	7
3.5 宿舍管理员模块	13
3.6 学生模块	15
第四章 关键技术研究	17
4.1 实现技术路线	17
4.2 关键技术研究	19
4.2.1 数据库技术	19
4.2.1 JSP 技术	23
4.2.1 Servlet 技术	24
4.2.1 JDBC 驱动技术	25
第五章 系统实现	27
5.1 系统实现介绍	27
5.2 系统实现的不足	29

# 第一章 绪论

## 1.1 系统的目的和意义

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。对于目前人数众多的学校,手工记录数据未免过于繁琐,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。宿舍管理系统就是以这个目的出发,在充分节省资源的情况下,做到对在校学生做一个记录。

此系统可以实现的功能有:对所有在校学生进行信息登记并可以做到随时修改和更新,在特定条件下对学生信息进行检索查询,可以对学生的违规进行记录,可以上传和下载文件,方便对学生进行通知,此系统面对不同人群,对各个人群进行分开登录,防止系统被攻击添加验证码功能。

我们使用日趋成熟的计算机技术来代替传统的人工模式,来实现学生信息的现代化管理,这一技术具有手工管理所无法比拟的优点,如:检索迅速,查找方便,可靠性高,存储量大,保密性好,寿命长,成本低等。这些优点能够极大地提高学生信息管理的效率,使学校在市场经济的发展过程中具有一定的竞争优势,也是学校实现科学化,正规化,现代化的重要条件。实现了科学、统一管理,避免了人工操作产生的错误,大大提高了工作效率。

为了实现提高高校宿舍管理部门的工作效率;充分利用资源;减少不必要的人力,物力和财力的支出;方便宿舍管理部门的工作人员全面地掌握学生住宿情况等目的;为宿舍管理部门,开发设计专用系统学生宿舍信息管理系统来进行管理学生宿舍信息,使学生宿舍信息实现标准化的管理和规范化的制度十分重要。

## 第二章 宿舍管理系统需求分析

### 2.1 引言

【简要说明需求分析的目的，内容，手段等】

此系统旨在对在校学生的信息进行记录，方便在校师生查询和修改，为学校宿舍管理提供一个快捷便利的管理手段，提高高校宿舍管理部门的工作效率，减少不必要的人力，物力和财力的支出，方便宿舍管理部门的工作人员全面地掌握学生住宿情况而做出的管理工具。目的是减少对学校在校学生管理的成本，使学校管理科学化，正规化，现代化。

### 2.2 需求分析

【通过需求分析，主要详细说明系统要实现的主要内容和功能有那些】

这个宿舍管理系统主要面向于教育机构。

在宿舍中居住的人群主要是学生，剩下的是管理学生的宿舍管理员，同时，为了方便系统的维护与管理，还需要专门设置一个系统管理员的权限账号。

在宿舍的管理系统中，需要做到以下功能：

1. 需要对所有宿舍相关人员的信息登记。
2. 能够随时更新宿舍相关人员的信息。
3. 能够通过特定的条件来查询指定宿舍相关人员的信息。
4. 拥有考勤机制来对宿舍学生进行管理。
5. 可以上传与下载文件，方便宿舍消息的通知与公告。
6. 系统同时面向系统管理员，宿舍管理员与学生，不同人群通过不同的账号密码进行登录访问，需要对不同人群的权限进行划分。
7. 为了防止系统被恶意攻击，在登录时需要输入验证码。
8. 同时为了方便用户账号密码的记忆，拥有修改密码功能。

在满足上述功能的同时，尽量做到界面的美观，功能界面清晰简洁易懂。

## 2.3 界面需求分析

由于宿舍管理系统的使用对象（大多情况是没有什么计算机操作经历的宿舍管理员），使得界面需求应该变得以“简”为重。

在本宿舍管理系统中：

1. 登录界面只有输入账号、密码、验证码的输入栏，账号类型的选择，记住账号密码的勾选栏，和登录与注册的按钮。
2. 系统界面中主要分为功能菜单界面和信息界面，使用时，信息界面与菜单界面互不干扰，同时，系统界面需要有一个欢迎界面作为首页。
3. 功能菜单界面中，所有的功能类型位于系统的一侧纵向菜单中排列。
4. 信息界面用表格呈现，对于信息查询和添加的按钮位于表格的上方，修改与删除的按钮位于表格中每条信息的最后一格，分页按钮位于表格下方。

## 2.4 功能需求分析

由于宿舍管理系统主要是基于宿舍相关人员信息的管理系统，因此系统需要连接数据库，数据库中各表记录不同的信息，满足：

1. 各表之间相互独立。
2. 账户不能直接访问数据库，只能看到数据库中表的返回的查询信息。
3. 功能按钮（添加，修改，查询，删除等）的操作与其修改表的相应操作相对应。
4. 不同账户对数据库中表的修改权限不同。
5. 学生以自己学号为账户。

同时为了通知消息的公布，所有用户拥有文件的上传与下载功能，上传的文件需经过审核。

另外，为了防止系统被恶意攻击，登录时，需要验证码功能，验证码需要在每次刷新访问界面的时候进行更新。

## 2.5 数据库需求分析

连接的数据库包含系统管理员表、宿舍楼信息表、宿舍管理员信息表、考勤

违规记录表和学生信息表，各表的信息为：

1. 系统管理员表：记录系统管理员的账户密码，以及姓名性别联系方式（主要用于系统管理员的登录）
2. 宿舍楼信息表：记录宿舍楼楼号
3. 宿舍管理员信息表：记录宿舍管理员的账户密码，以及管理的宿舍楼号、姓名、性别和联系方式
4. 考勤违规记录表：记录违规学生的学号、姓名、宿舍楼号、寝室号以及违规日期
5. 学生信息表：记录学生的学号（账户）、密码，以及姓名、宿舍楼号，寝室号、性别和联系方式

## 第三章 宿舍管理系统结构分析与设计

### 3.1 引言

【简要说明系统分析的目的，作用，系统分析的策略，系统分析的方法，把软件工程那套东西写进来，包括用例图，时序图，类结构图等】

学生宿舍管理是当代学校中不可或缺的组成部分，在信息化飞速发展的今天，它的出现已经成了一种必然，本文将从宿舍管理系统的历史和现状进行分析，以求证本系统的设计具有实际的意义，并得到它在未来的发展，从而跟上时代的脚步，做出一个高质量的、安全性好的宿舍管理系统，使其不仅能满足学校现在的需要，也能满足将来一段时间的需要。

学生宿舍管理系统的开发主要包括前台用户界面的开发和后台数据库的开发，对于后台数据库的建立和维护要求建立起数据一致性和完整性强、数据安全性的数据库。而对于前端应用程序的开发则要求应用程序能提供强大的数据操纵功能，界面友好、使用简单方便等特点。本系统采用 java 作为前台用户界面的开发语言，java 语言开源而且稳定，后台的数据库采用 MySql。

整个系统可以分为以下四个模块：

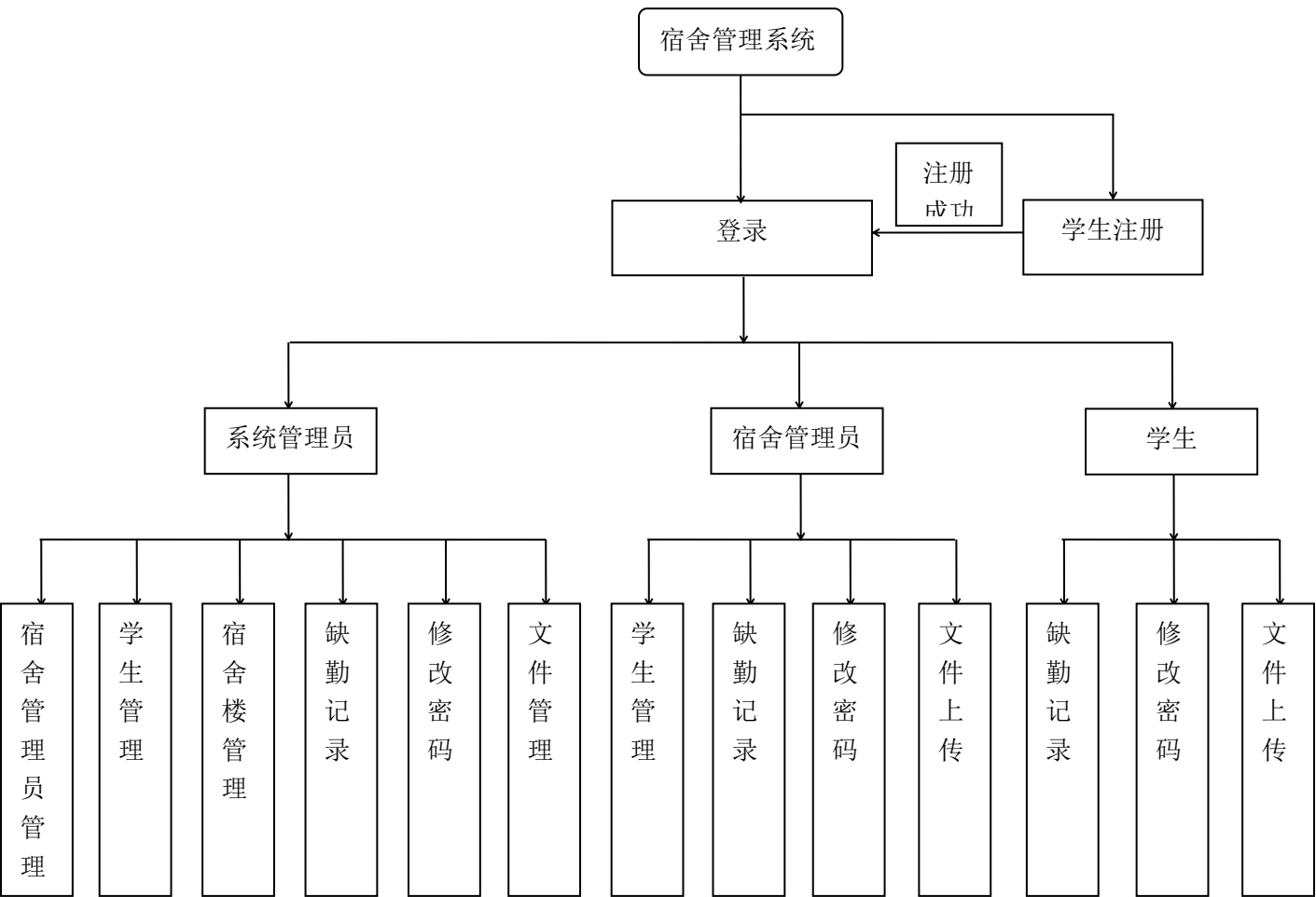
- 学生注册模块
- 系统管理员模块
- 宿舍管理员模块
- 学生模块

下面对各个模块做详细说明。

### 3.2 系统模块结构图

【给出一个系统模块的组合起来的结构框图】

系统总功能图：



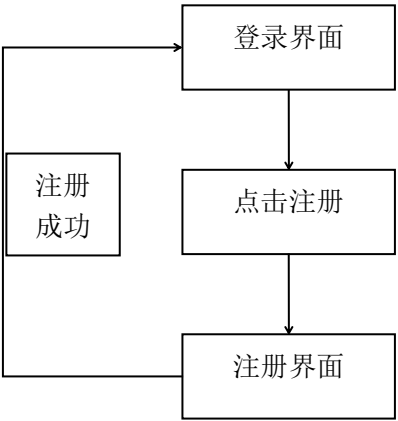
### 3.3 学生注册模块

【根据功能分析得出的结论，划分模块，分别设计各个模块结构，理清各个模块之间的交互关系】

学生注册模块主要用于学生第一次登陆系统，使学生将自己的个人信息记录到数据库中，这样可以减少统一录入信息的工作量。这个注册主要是针对学生的，没有增加宿舍管理员注册，宿舍管理员注册由系统管理员直接设定即可。

在登录界面点击注册，跳转到注册页面，输入各项信息后，点击确定便注册成功跳转到登录界面。这样学生注册完成。

注册流程如下：



### 3.4 系统管理员模块

系统管理员作为系统拥有最高权限的用户，可以对所有用户信息进行操作，同时也拥有系统的所有功能。系统管理员登录时需要输入系统管理员的用户名和密码，同时要在下方身份选择系统管理员，这样登录时服务器便会在系统管理员表中获取用户名和密码进行对比，验证通过则登录成功，否则提示错误无法通过。

登录成功后就可以对系统管理员的功能进行操作。以下对各个功能逐个说明

- 宿舍管理员管理：可以对现有的宿舍管理员信息包括登录用户名和密码进行查询、增加、修改和删除操作。查询可选择按姓名查找或者按用户名查找，

同时也实现模糊查询操作。点击添加管理员时页面跳转到输入管理员信息界面，输入管理员的全部信息便添加成功。

姓名

搜索

姓名

用户名

\*用户名:

\*密码:

\*重复密码:

\*姓名:

\*性别:

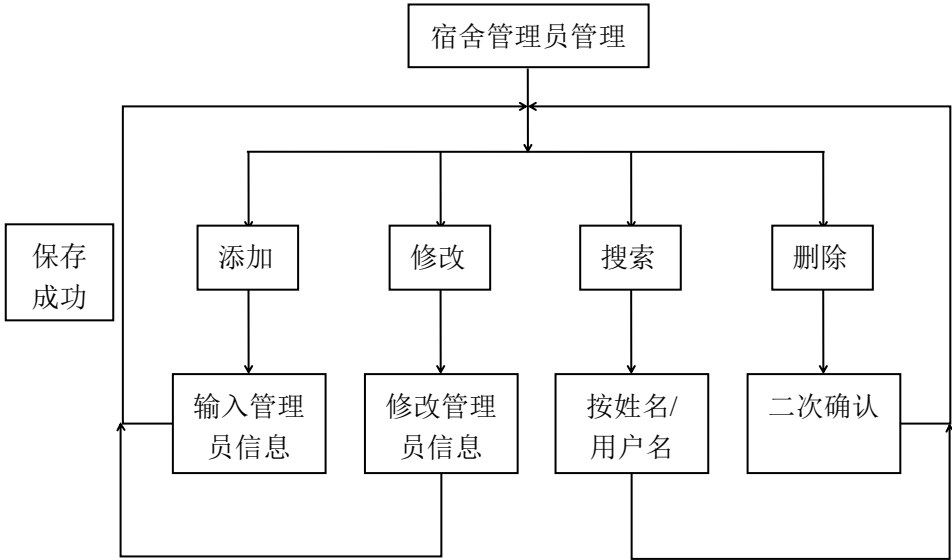
请选择...

\*联系电话:

保存

返回

宿舍管理员管理层次图：



- 学生管理：对数据库中现有的学生信息进行增删改查操作，其中查询时可限定查找的学生位于哪栋宿舍楼中，然后选择按姓名、学号或者寝室号查找，



或者直接进行模糊查询操作。添加学生信息时需要输入学生的全部信息，然后点击确定即可。

全部宿舍楼

姓名

搜索

姓名

学号

寝室

\*学号:

\*密码:

\*重复密码:

\*姓名:

\*性别:

请选择...

\*宿舍楼:

1栋

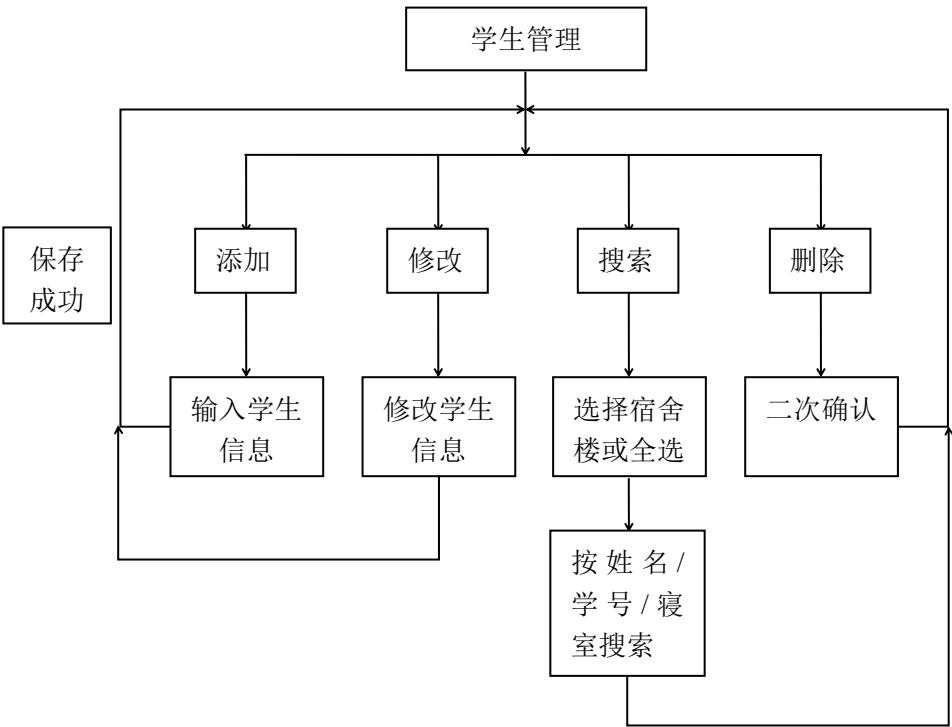
\*寝室:

\*联系电话:

保存

返回

学生管理层次图



- 宿舍楼管理：对现有的宿舍楼信息进行增删改查操作。查询时可以进行模糊查询操作。同时可以在此界面查看每栋宿舍楼的宿舍管理员信息，也可以为某个宿舍楼增加或者减少管理员，系统限定每个管理员最多负责一个宿舍楼的管理工作，因此添加管理员时会在所有没有负责任何宿舍楼的管理员中进行选择。例如：此管理员没有负责任何宿舍楼，

5	sdf	女	123	无	manager6	<span>修改</span>	<span>删除</span>
---	-----	---	-----	---	----------	-----------------	-----------------

则添加管理员时，添加列表只能选择她。

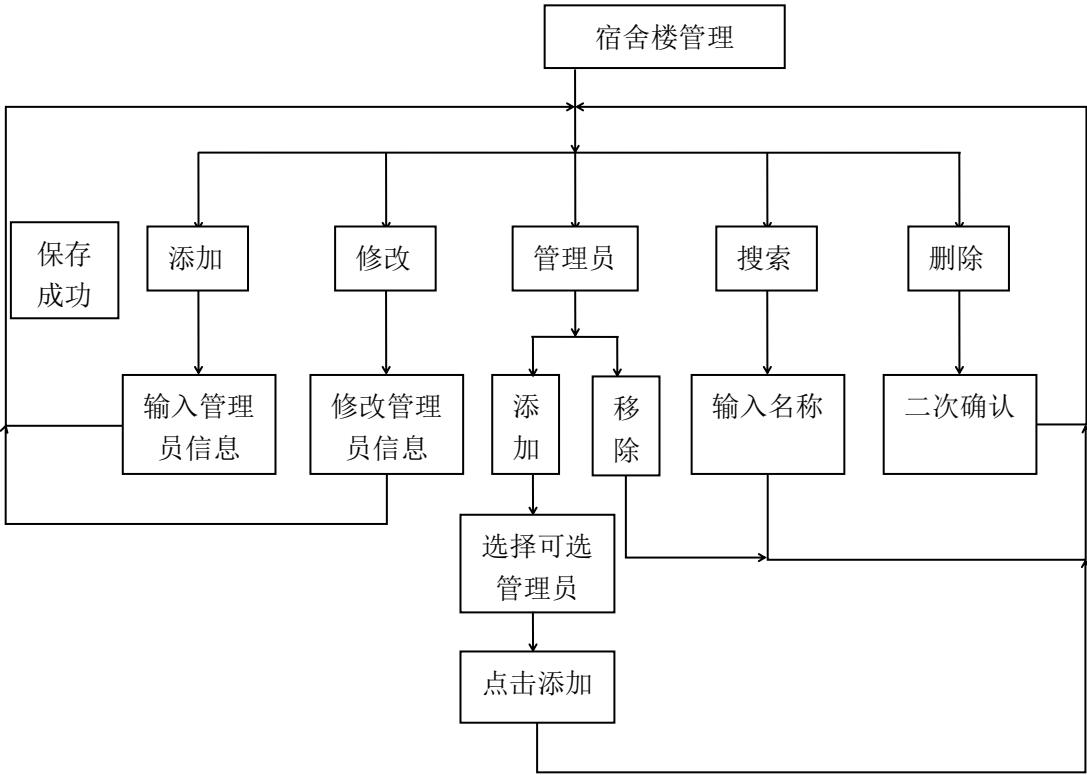
请选择管理员...

添加

请选择管理员...

sdf

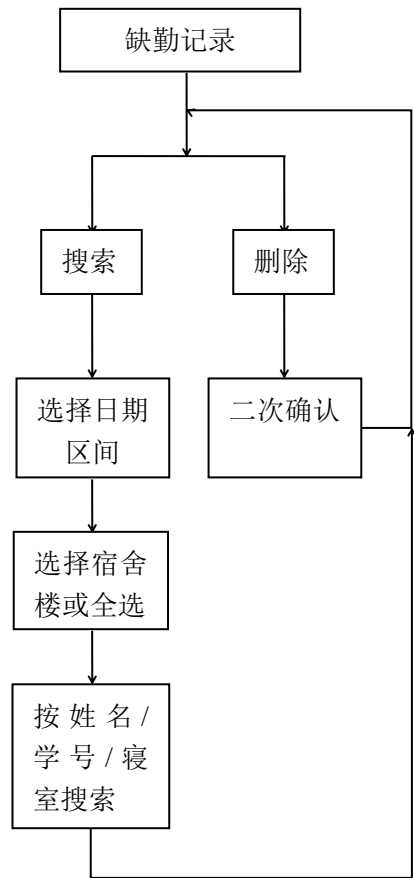
宿舍楼管理层次图：



- 缺勤记录：此功能记录着所有学生的缺勤记录。查询记录使可选择按照宿舍楼和姓名、学号、寝室进行查询，或者对学生缺勤记录进行删除操作

1栋	▼	学号	▼		搜索
	备注	姓名		操作	
		学号			
	--	寝室		删除	

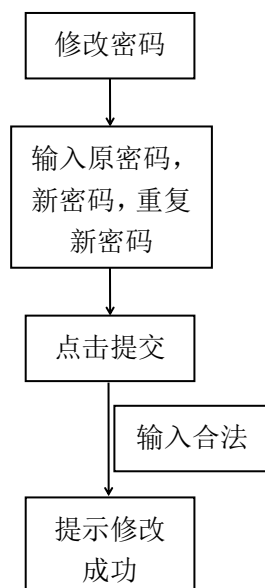
缺勤记录层次图：



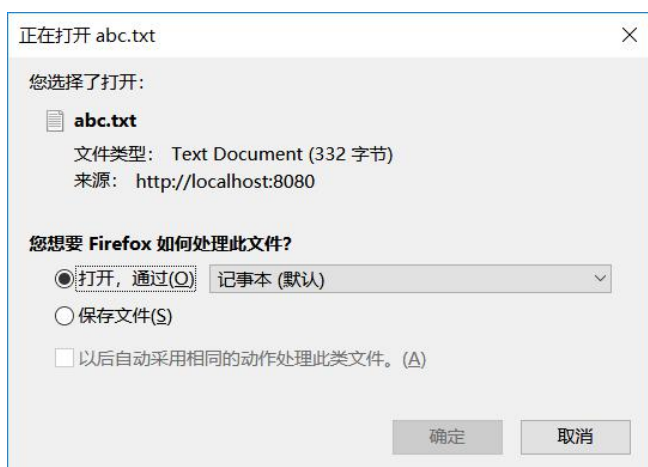
- 修改密码：可对自己的密码进行修改，修改时需要输入以下内容，输入合法后提交便修改成功。

*原密码：	<input type="password"/>
*新密码：	<input type="password"/>
*重复密码：	<input type="password"/>
<input type="button" value="提交"/>	

修改密码层次图：



- 文件管理：可以上传和下载文件，列表中的文件后面点击下载便显示如图所示内容，点击确定便开始进行下载操作。



点击上传按钮，系统跳转到上传文件界面，可选择需要上传文件的路径、上传文件用户的身份和备注信息，点击提交后便开始上传。

请选择需要上传的文件: 浏览... 未选择文件。

请选择您的身份 (可选) : ☐ 本科生 ☐ 研究生 ☐ 教师 ☐ 宿舍管理员

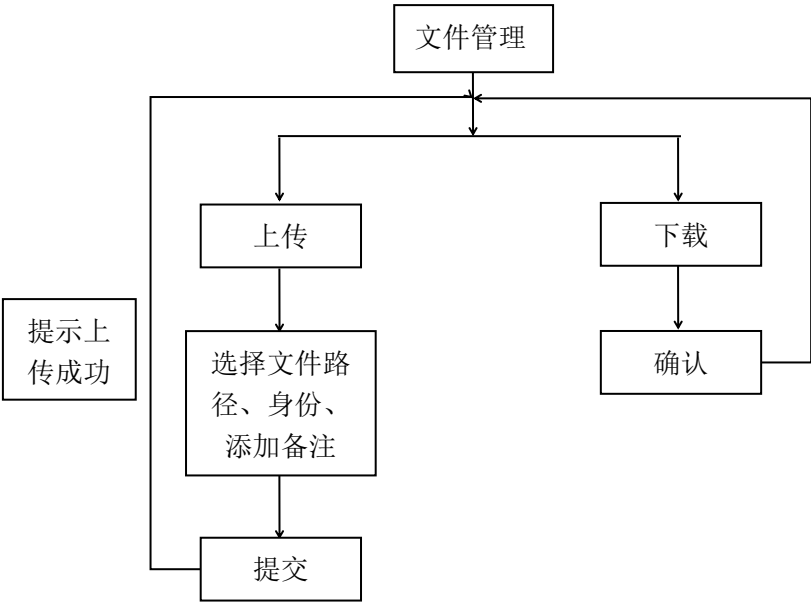
请输入备注 (可选)

[提交](#) [返回](#)

**上传成功，正在等待管理员审核**

[继续上传](#) [返回](#)

文件管理层次图：



### 3.5 宿舍管理员模块

宿舍管理员作为权限位于系统管理员之下的用户，不能修改宿舍管理员信息和宿舍楼信息，其他功能和系统管理员类似，但是还是有所不同。宿舍管理员进行登录时需要输入宿舍管理员的用户名和密码，同时要在下方身份选择系统管理

员，这样登录时服务器便会在宿舍管理员表中获取用户名和密码进行对比，验证通过则登录成功，否则提示错误无法通过。

宿舍管理员拥有的功能为：查看学生信息、添加学生缺勤记录、修改密码、文件管理。下面对各个功能逐个说明：

- 学生查看：可以查看学生的详细信息，但是只能查看到自己所管理的宿舍楼中的学生的信息。可以选择按照姓名、学号或者寝室进行搜索查询。

姓名

姓名

学号

寝室

搜索

也可以在某个学生信息后边点击添加缺勤记录，添加记录时只需要输入备注即可，日期会自动读取今天的日期，点击保存后，便可在缺勤记录中查看到改缺勤记录。

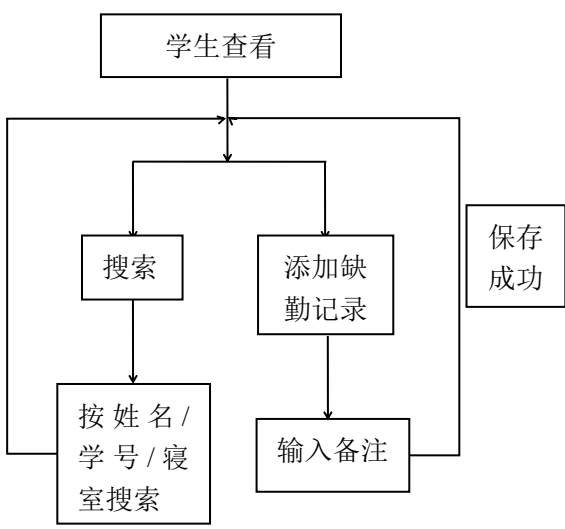
\*学号: 002

\*日期: 2019-06-23

\*备注:

保存 返回

学生查看层次图：



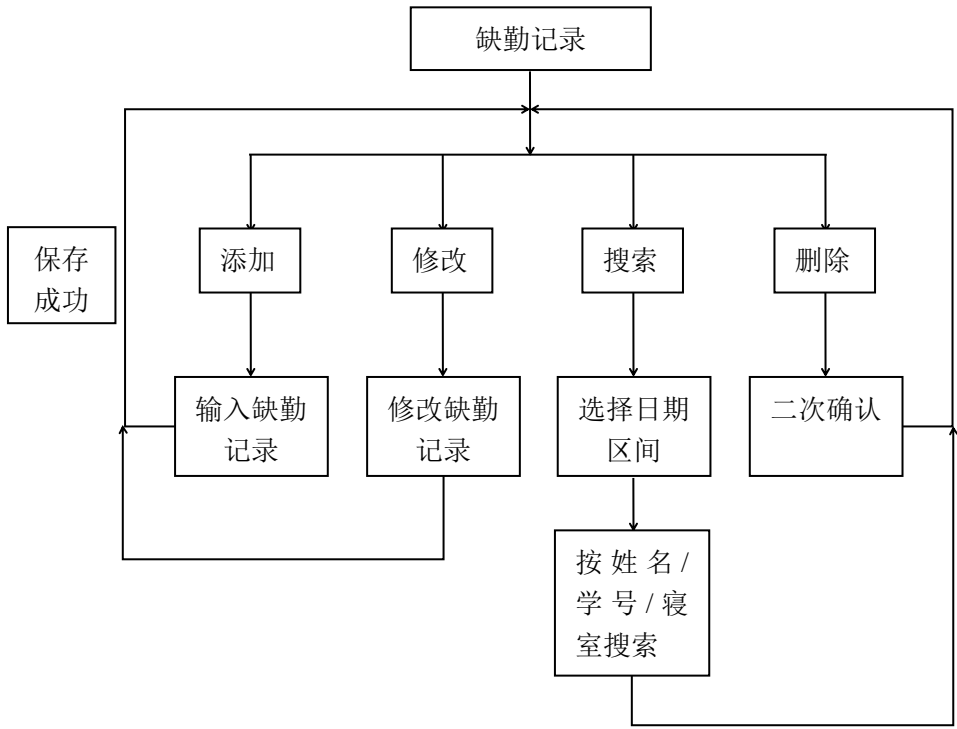
- 缺勤记录：可以对现有的且位于本宿舍楼中的学生的缺勤记录进行查询、修改和删除操作，查询时可选择按学号、姓名或者寝室进行查询。点击修改时可以修改学生的学号和备注信息。

\*学号:

\*日期:

\*备注:

缺勤记录层次图：



- 修改密码：此功能和系统管理员修改密码时功能相同，这里不再说明。
- 文件管理：此功能和系统管理员文件管理时功能相同，这里不再说明。

### 3.6 学生模块

学生作为系统中拥有权限最低的用户，仅可以查看自己的违规记录、修改密码和文件管理功能。学生登录时需要输入学生的用户名和密码，同时要在下方身

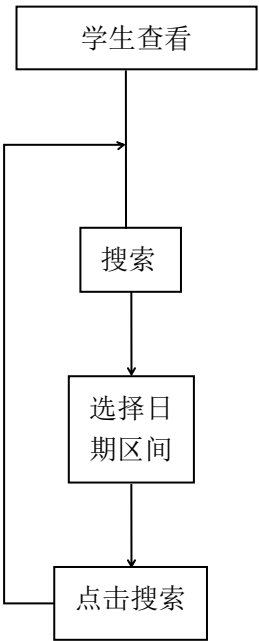
份选择学生，这样登录时服务器便会在学生表中获取用户名和密码进行对比，验证通过则登录成功，否则提示错误无法通过。

以下对各个功能逐个说明：

- 查看缺勤记录：每个用户仅能查看自己的缺勤记录，以及按日期进行搜索功能，不能进行其他操作。



查看缺勤记录层次图：



- 修改密码：此功能和系统管理员修改密码时功能相同，这里不再说明。
- 文件管理：此功能和系统管理员文件管理时功能相同，这里不再说明。



## 第四章 关键技术研究

### 4.1 实现技术路线

【根据上面设计的系统功能模块结构，说明各个模块实现的总体技术路线。比如采用什么平台，使用什么语言，选择什么数据库，服务器端怎么做，客户端怎么做等等。以及为什么采用这条技术路线，和技术路线的优势】

#### 1. 系统软件环境

(1) 服务端 操作系统: Ubuntu 5.4.0

数据库系统: MySQL5.7

服务器: Tomcat8.5.39

(2) 客户端 操作系统: Microsoft Windows10

开发工具: MyEclipse,

开发语言: Java

(3) 开发模式: MVC 模式, 数据层、控制层、界面分开, 其中一个模块逻辑业务的变动不影响到其他层, 通过控制层, 将页面的信息保存到数据库, 也可从 数据库查询到的结果反馈到页面, 从而使系统稳定、高效的运行。

#### 2. Jsp + Servlet + MySQL 的技术路线

在与其他同类编程语言思想相比较之下, Java 具有开发速度快、可移植、面向对象、可扩展性强、开源自由等特点, 逐渐成为 web 开发领域的主流技术。就本设计而言, 用户通过浏览器访问本系统, 高效的执行效率是提高用户体验的重要因素。

选择使用 CSS+HTML+JSP 设计界面, 一些特别的显示效果使用 JavaScript 技术实现了一个很好的用户界面。

本系统是基于 Jsp+Servlet+Mysql 的架构, 适合我们这种初学者上手。

这里分开介绍一下三者。

(1) JSP:

JSP 的特点与优势：一次编写，到处运行；系统的多平台支持；内容和显示分离；生成可重用组件；健壮的存储管理和安全性。

JSP 的工作原理：当用户访问一个 JSP 页面时，JSP 页面将被编译成 Servlet 文件 JSP 引擎调用 JAVA 编译器，编译 Servlet 文件为可执行的代码文件用 JAVA 虚拟机执行.class 文件，并将执行结果返回给服务器，服务器将执行结果以 HTML 格式发送给客户端的浏览器

JSP 的两种工作模式：JSPMODEL1，JSP 页面独自响应请求并将处理结果返回客户，缺点是将导致角色定义不清和职责分配不明，给项目带来不必要的麻烦。JSP MODEL2，是一种把 JSP 和 Servlet 以及 JQuery 联合使用来实现动态内容服务的方法，项目越复杂其功能越突出

JSP 开 发 WEB 站 点 的 主 要 方 式 ： 直 接 使 用 JSPJSP+JAVABEANJSP+JAVABEAN 框架整合应用

(2) MySQL 是一个小型关系型数据库管理系统,开发者为瑞典 MySQL AB

公司,现在已经被 Sun 公司收购,支持 FreeBSD、Linux、MAC、Windows 等多种操作系统与其他的大型数据库例如 Oracle、DB2、SQL Server 等相比功能稍弱一些。优点如下:

1. 它使用的核心线程是完全多线程，支持多处理器。
2. 有多种列类型：1、2、3、4、和 8 字节长度自有符号 / 无符号整数、FLOAT、DOUBLE、CHAR、VARCHAR、TEXT、BLOB、DATE、TIME、DATETIME、TIMESTAMP、YEAR、和 ENUM 类型。

3. 它通过一个高度优化的类库实现 SQL 函数库并像他们能达到的一样快速，通常在查询初始化后不该有任何内存分配。没有内存漏洞。
4. 全面支持 SQL 的 GROUP BY 和 ORDER BY 子句，支持聚合函数 (COUNT()、COUNT(DISTINCT)、AVG()、STD()、SUM()、MAX() 和 MIN())。你可以在同一查询中混来自不同数据库的表。
5. 支持 ANSI SQL 的 LEFT OUTER JOIN 和 ODBC。
6. 所有列都有缺省值。你可以用 INSERT 插入一个表列的子集，那些没用明确给定值的列设置为他们的缺省值。

MySQL 可以工作在不同的平台上。支持 C、C++、Java、Perl、PHP、Python 和 TCL API。

## 4.2 关键技术研究

### 4.2.1 数据库技术

从需求分析可以看出，本系统存在以下三个实体：

学生实体：用于显示学生的属性，包括学生学号、姓名、所在宿舍楼、性别、宿舍号、联系方式、违规记录、用户名、密码。

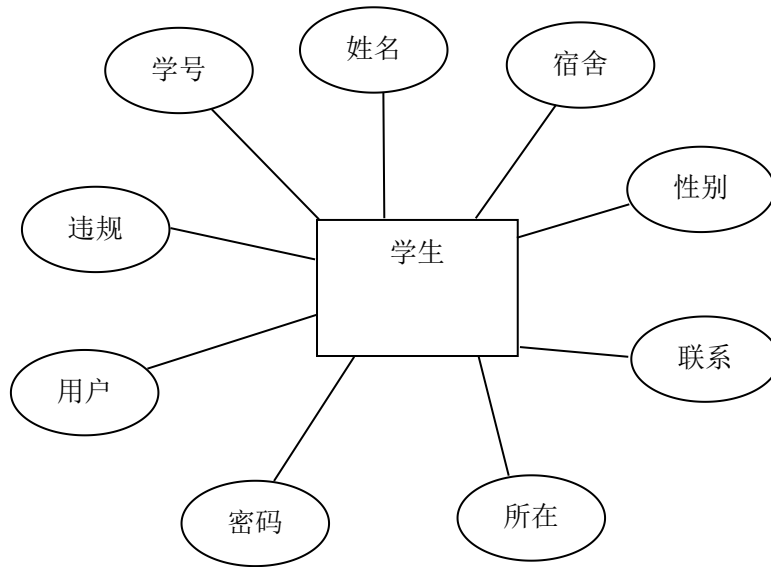
宿舍管理员实体：用于显示宿管员的属性，包括姓名、性别、联系方式、所负责宿舍楼、用户名、密码。

系统管理员实体：用于显示系统管理员的属性，包括姓名、性别、联系方式、用户名、密码。

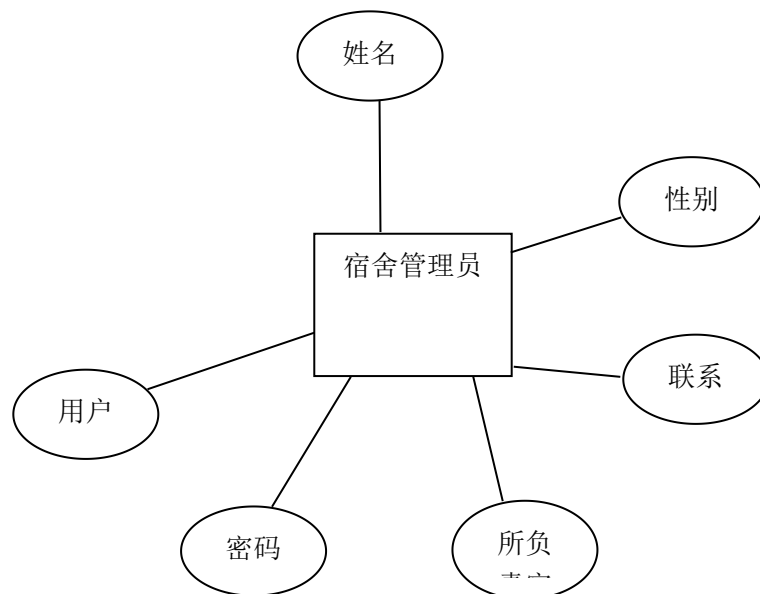
E-R 模型设计：

根据实体设计，可以得到各个实体 E-R 图和部分子系统 E-R 图。

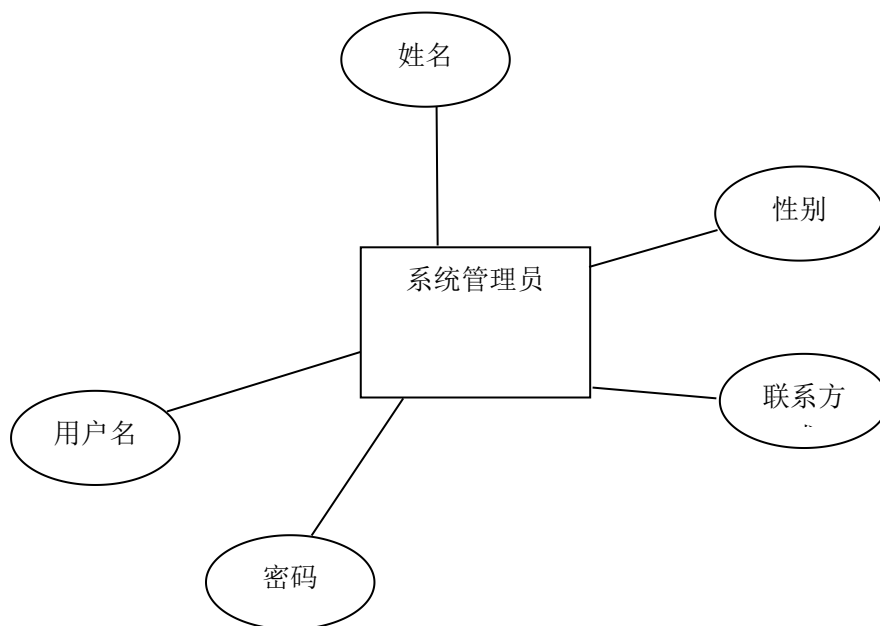
(1) 学生实体 E-R 图如图所示



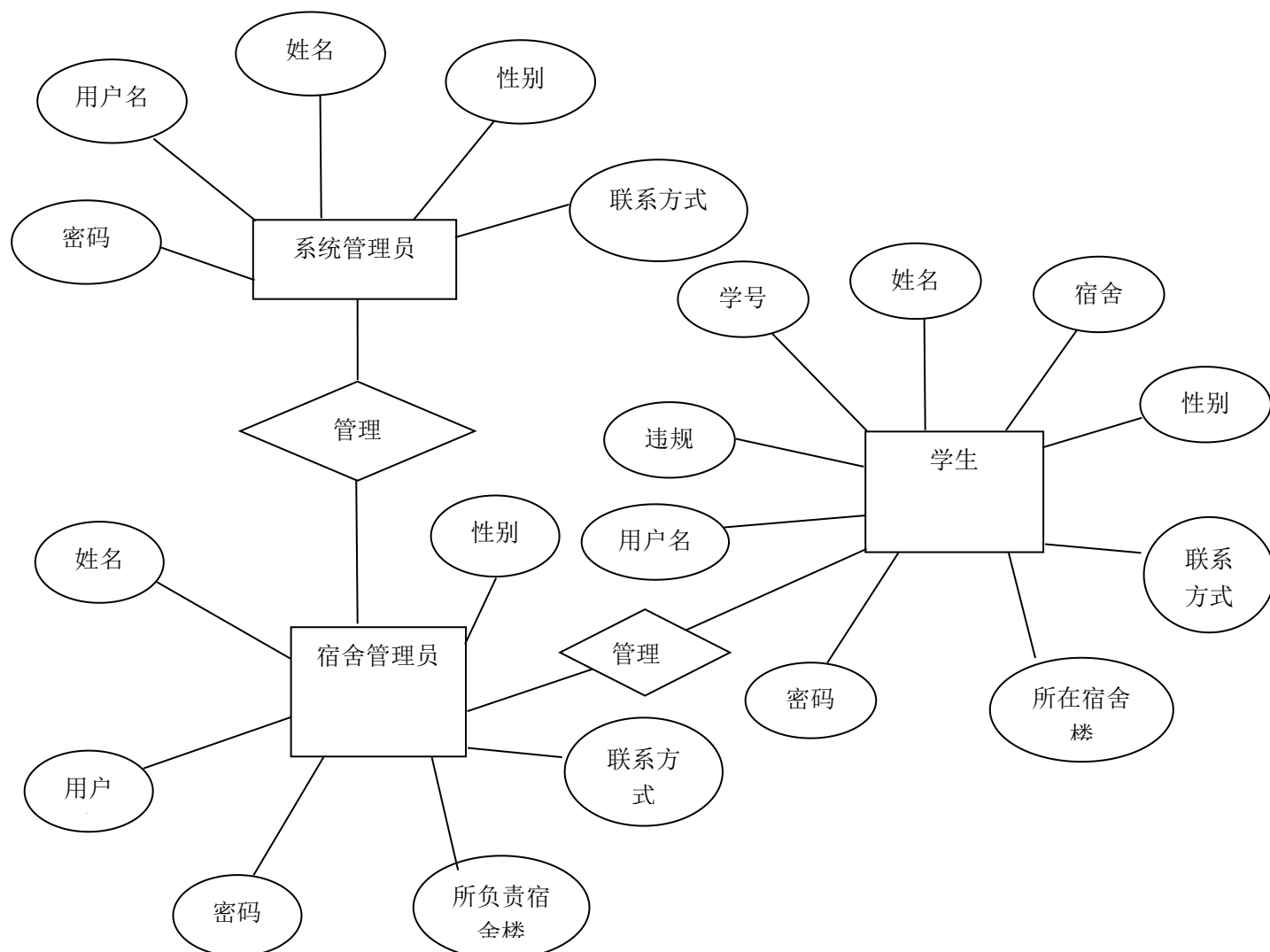
(2) 宿舍管理员实体 E-R 图如图所示



(3) 系统管理员实体 E-R 图如图所示



(4) 汇总之后的 E-R 图如图所示



## (5) 数据库逻辑结构设计

## 1. 学生信息表

存储了学生的基本信息和登录信息。

名	类型	长度	小数点	允许空值 (	
▶ studentId	int	11	0	<input type="checkbox"/>	 1
stuNum	varchar	20	0	<input checked="" type="checkbox"/>	
password	varchar	20	0	<input checked="" type="checkbox"/>	
name	varchar	20	0	<input checked="" type="checkbox"/>	
dormBuildId	int	11	0	<input checked="" type="checkbox"/>	
dormName	varchar	11	0	<input checked="" type="checkbox"/>	
sex	varchar	10	0	<input checked="" type="checkbox"/>	
tel	varchar	15	0	<input checked="" type="checkbox"/>	

## 2. 宿舍管理员信息表

存储了宿舍管理员的基本信息和登录信息。

名	类型	长度	小数点	允许空值 (	
▶ dormManId	int	11	0	<input type="checkbox"/>	 1
userName	varchar	20	0	<input checked="" type="checkbox"/>	
password	varchar	20	0	<input checked="" type="checkbox"/>	
dormBuildId	int	11	0	<input checked="" type="checkbox"/>	
name	varchar	20	0	<input checked="" type="checkbox"/>	
sex	varchar	20	0	<input checked="" type="checkbox"/>	
tel	varchar	20	0	<input checked="" type="checkbox"/>	


## 3. 系统管理员信息表

存储了系统管理员的基本信息和登录信息。

名	类型	长度	小数点	允许空值 (	
▶ adminId	int	11	0	<input type="checkbox"/>	 1
userName	varchar	20	0	<input checked="" type="checkbox"/>	
password	varchar	20	0	<input checked="" type="checkbox"/>	
name	varchar	20	0	<input checked="" type="checkbox"/>	
sex	varchar	10	0	<input checked="" type="checkbox"/>	
tel	varchar	20	0	<input checked="" type="checkbox"/>	

## 4. 宿舍楼信息表

存储了宿舍楼的基本信息。

名	类型	长度	小数点	允许空值 (	
▶ dormBuildId	int	11	0	<input type="checkbox"/>	 1
dormBuildName	varchar	20	0	<input checked="" type="checkbox"/>	
dormBuildDetail	varchar	50	0	<input checked="" type="checkbox"/>	

## 5. 缺勤记录表

存储了学生的缺勤记录信息。

名	类型	长度	小数点	允许空值 (	
▶ recordId	int	11	0	<input type="checkbox"/>	 1
studentNumber	varchar	20	0	<input checked="" type="checkbox"/>	
studentName	varchar	30	0	<input checked="" type="checkbox"/>	
dormBuildId	int	11	0	<input checked="" type="checkbox"/>	
dormName	varchar	11	0	<input checked="" type="checkbox"/>	
date	date	0	0	<input checked="" type="checkbox"/>	
detail	varchar	50	0	<input checked="" type="checkbox"/>	

### 4.2.1 JSP 技术

JSP 全名为 Java Server Pages，中文名叫 java 服务器页面，其根本是一个简化的 Servlet 设计。用 JSP 开发的 Web 应用是跨平台的，既能在 Linux 下运行，也能在其他操作系统上运行。

它实现了 html 语法中的 java 扩展（以 <%, %>形式）。JSP 与 Servlet 一样，是在服务器端执行的。通常返回给客户端的就是一个 HTML 文本，因此客户端只要有浏览器就能浏览。

JSP 技术使用 Java 编程语言编写类 XML 的 tags 和 scriptlets，来封装产生动态网页的处理逻辑。JSP 将网页逻辑与网页设计的显示分离，支持可重用的基于组件的设计，使基于 Web 的应用程序的开发变得迅速和容易。 JSP 是一种动态页面技术，它的主要目的是将表示逻辑从 Servlet 中分离出来。

在宿舍管理系统中，JSP 用于接收 Servlet 传的值，并显示在页面上，如下所示：

```

<%@page import="java.util.Date"%>
<%@ 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=UTF-8">
<title>Insert title here</title>
</head>
<body>

    <%
        //1. 通知客户端浏览器：这是一个需要下载的文件，不能再按普通的 html 的方式打开。
        //即设置一个响应的类型：application/x-msdownload
        response.setContentType("application/x-msdownload");

        //2. 通知客户端浏览器：不再有浏览器来处理该文件，而是交由用户自行处理
        //设置用户处理的方式：响应头：Content-Disposition
        response.setHeader("Content-Disposition", "attachment;filename=def.txt");
    %>

    <h4>Test Page</h4>

    Time: <%= new Date() %>

</body>
</html>

```

### 4.2.1 Servlet 技术

Servlet (Server Applet) 是 Java Servlet 的简称，称为小服务程序或服务连接器，用 Java 编写的服务器端程序，具有独立于平台和协议的特性，主要功能在于交互式地浏览和生成数据，生成动态 Web 内容。

狭义的 Servlet 是指 Java 语言实现的一个接口，广义的 Servlet 是指任何实现了这个 Servlet 接口的类，一般情况下，人们将 Servlet 理解为后者。Servlet 运行于支持 Java 的应用服务器中。从原理上讲，Servlet 可以响应任何类型的请求，但绝大多数情况下 Servlet 只用来扩展基于 HTTP 协议的 Web 服务器。

这个过程为：

客户端发送请求至服务器端；

服务器将请求信息发送至 Servlet；

Servlet 生成响应内容并将其传给服务器。响应内容动态生成，通常取决于客户端的请求；

在宿舍管理系统中，Servlet 用于接收服务器发送的请求并进行处理，将处理结果返回给服务器。例如：



```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    request.setCharacterEncoding("utf-8");
    HttpSession session = request.getSession();
    String userName = request.getParameter("userName");
    String password = request.getParameter("password");
    String remember = request.getParameter("remember");
    String userType = request.getParameter("userType");

    Connection con = null;
    try {
        con=dbUtil.getCon();
        Admin currentAdmin = null;
        DormManager currentDormManager = null;
        Student currentStudent = null;
        String sessionCode = (String)request.getSession().getAttribute("CHECK_CODE_KEY");
        String paramCode = request.getParameter("CHECK_CODE_PARAM_NAME");

        if(paramCode != null && paramCode.equals(sessionCode)){
            if("admin".equals(userType)) {
                Admin admin = new Admin(userName, password);
                currentAdmin = userDao.Login(con, admin);
                if(currentAdmin == null) {
                    request.setAttribute("admin", admin);
                    request.setAttribute("error", "用户名或密码错误! ");
                    request.getRequestDispatcher("login.jsp").forward(request, response);
                } else {
                    if("remember-me".equals(remember)) {
                        rememberMe(userName, password, userType,response);
                    }
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if(con != null) {
            con.close();
        }
    }
}

```

此为 LoginServlet 中的部分代码，其作用是处理用户登录的请求，当用户的个人信息输入错误时，返回 error，并显示在页面上。

同时需要在 web.xml 中配置响应的参数，如图：

```

<servlet>
    <servlet-name>loginServlet</servlet-name>
    <servlet-class>com.lero.web.LoginServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>loginServlet</servlet-name>
    <url-pattern>/login</url-pattern>
</servlet-mapping>

```

### 4.2.1 JDBC 驱动技术

宿舍管理系统是采用 JDBC 驱动直接访问 MySQL 数据库，驱动程序完全由 Java 语言编写，运行速度快，而且具备了跨平台的特点。使用纯 Java 驱动方式进行数据库连接，首先需要下载数据库厂商提供的驱动程序 JAR 包，并将 JAR 包引入工程中。如图所示：

```
▼ lib
  jstl.jar
  mysql-connector-java-3.1.12-bin.jar
  standard.jar
```

连接数据库的部分代码为：

```
package com.lero.util;

import java.io.IOException;

public class PropertiesUtil {

    public static String getValue(String key) {
        Properties prop = new Properties();
        InputStream in = new PropertiesUtil().getClass().getResourceAsStream("/dorm.properties");
        try {
            prop.load(in);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return (String)prop.get(key);
    }
}
```

连接数据库的参数配置在 `dorm.properties` 中

```
dbUrl=jdbc:mysql://localhost:3306/db_dorm?useUnicode=true&characterEncoding=UTF8
dbUserName=root
dbPassword=
jdbcName=com.mysql.jdbc.Driver
pageSize=5
```

## 第五章 系统实现

### 5.1 系统实现介绍

【本章只是对系统实现内容，实现程度，实现效果的一个介绍，不是满篇贴代码，主要针对系统的关键功能，关键技术，技术难点实现结果的说明，可以粘贴部分系统程序的界面做说明和展示】

登录主界面如图：



用系统管理员的身份进行登录，登录后的主界面如下



用户进行登录时，JSP 将用户输入的用户名、密码、身份信息、验证码发送给 Servlet 进行处理，Servlet 根据用户所选择的身份信息，在指定的表中查找是否有此用户的记录，然后进行验证，将结果返回给服务器，登录成功后便进行跳转。由于设置了 session，用户无法直接访问登陆之后的界面，而会被重定向到

登录界面。

分页功能的实现：实现代码如下所示

```
private String genPagation(int totalNum, int currentPage, int pageSize){
    int totalPage = totalNum%pageSize==0?totalNum/pageSize:totalNum/pageSize+1;
    StringBuffer pageCode = new StringBuffer();
    pageCode.append("<li><a href='dormBuild?page=1'>首页</a></li>");
    if(currentPage==1) {
        pageCode.append("<li class='disabled'><a href='#'>上一页</a></li>");
    }else {
        pageCode.append("<li><a href='dormBuild?page="+currentPage-1+"'>上一页</a></li>");
    }
    for(int i=currentPage-2;i<=currentPage+2;i++) {
        if(i<1||i>totalPage) {
            continue;
        }
        if(i==currentPage) {
            pageCode.append("<li class='active'><a href='#'>"+i+"</a></li>");
        } else {
            pageCode.append("<li><a href='dormBuild?page="+i+"'>"+i+"</a></li>");
        }
    }
    if(currentPage==totalPage) {
        pageCode.append("<li class='disabled'><a href='#'>下一页</a></li>");
    } else {
        pageCode.append("<li><a href='dormBuild?page="+currentPage+1+"'>下一页</a></li>");
    }
    pageCode.append("<li><a href='dormBuild?page="+totalPage+"'>尾页</a></li>");
    return pageCode.toString();
}
```

在系统中的效果是



点击下一页会显示下一页的内容



---

## 5.2 系统实现的不足

此宿舍管理系统大体上实现了宿舍管理中一些简单的功能，但是还是存在很多不足之处：

1. 由于设计时设计的一些路径没有规律性，因此在添加 **filter** 时遇到了一些难以解决的困难，导致在设计时遇到了很多的 **bug**。
2. 系统没有实现统计在线人数的功能。
3. 一些界面只是简单的实现了功能，但是界面不够美观
4. 作为一个宿舍管理系统，目前实现的功能还远远不够，一些功能还需要进行完善。
5. 数据库中各个表之间没有关联，会出现学生信息和缺勤记录学号不同的情况。
6. 由于是第一次设计这样一个完整的系统，在一些方面没有经验，不能完全从客户的角度出发去思考问题，有待进步。