
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 系统需求分析	3
2.1 引言.....	3
2.2 需求分析.....	3
2.3 功能需求分析	3
2.4 非功能需求分析.....	6
2.5 环境需求分析	6
第三章 系统结构分析与设计.....	7
3.1 引言.....	7
3.2 系统模块结构图.....	8
3.3 模块一.....	8
3.4 模块二.....	9
3.5 模块三.....	9
3.5 模块四.....	9
3.6 模块五.....	9
3.7 工作流程.....	9
3.8 数据库设计.....	10
3.8.1 数据库概念设计	10
3.8.2 数据库逻辑设计	14
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.....	18
4.2.1 关键技术一：Javascript	18
4.2.2 关键技术二：JDBC	18
4.2.3 关键技术三：Servlet.....	19
4.2.4 关键技术四：JSP.....	19
4.2.5 关键技术五：MVC 设计模式.....	20
第五章 系统实现.....	21
5.1 系统实现介绍.....	21
5.2 系统实现的不足.....	28

第一章 绪论

1.1 系统的目的和意义

通过对当前宿舍管理系统的分析和了解，得知当前宿舍管理系统有些采用 C/S，使得客户端必须要特定的软件支持，并且针对不同的操作系统必须开发不同版本的软件。即使有些系统采用了 B/S 模式，但是系统的视图和控制不分离，对数据库的操作代码和页面显示代码在一个页面，容易导致系统运行时不稳定，并且使得系统后期维护和升级困难，提高了产品的总成本。

本课题就是在这样背景下提出的，学生宿舍管理系统的成功开发将满足我校学生管理发展的需要，改变原有宿舍管理工作的常规模式，为提高我校学生管理水平有着重要的作用。目的是通过对已知的学生宿舍管理系统的研究，以 Jsp+Servlet+MySQL 技术设计并实现一个学生宿舍管理系统，以期改变学校现有的宿舍管理工作方式，充分利用计算机技术和校园网的优势，为学校实现优质高效的宿舍管理工作提供信息化技术支持。通过将开发实现的学生宿舍管理系统，投入到学校宿舍管理工作中，提高工作效率的同时，可以节省人力物力财力，不仅获得学生的拥护和肯定，而且可以为学校的稳定发展发挥应有的作用，具有较好的社会和经济效益。

学生宿舍管理系统的研究意义：随着招生规模的扩大和学生人数的增多，传统管理操作方式的弊端日益明显，与当今信息化时代背景下的校园信息化的矛盾日益突出。比如面对大量的信息，如果只依靠人工记载，最后形成纸质版本的文件，不仅工作量大，还容易出错，统计信息的可靠性不是很高，缺乏规范性，使得工作质量难以提高；纸质版本的信息查询、更新比较繁琐，文件保存起来也有诸多不便，相关文件信息一旦丢失，将无法找回。宿舍管理工作往往涉及多个部门之间的协作，传统的人工操作方式在各个部门的沟通、衔接方面也需要诸多环节，占据了大量的时间，导致管理效率低下，信息传达、反馈不及时。鉴于此，这种传统的管理操作方式急需被取代，现代化的管理信息系统呼之欲出，本文设计的系统模式是针对宿舍管理工作的实际需要专门开发设计的，具有较强的现实意义。

第二章 需求分析

2.1 引言

需求分析是软件计划阶段的重要活动，也是软件生存周期中的一个重要环节，该阶段是分析系统在功能上需要“实现什么”，而不是考虑如何去“实现”。需求分析的目标是把用户对待开发软件提出的“要求”或“需要”进行分析与整理，确认后形成描述完整、清晰与规范的文档，确定软件需要实现哪些功能，完成哪些工作。此外，软件的一些非功能性需求(如软件性能、可靠性、响应时间、可扩展性等)，软件设计的约束条件，运行时与其他软件的关系等也是软件需求分析的目标。

目前，软件需求的分析与设计方法较多，一些大同小异，而有的则基本思路相差很大。从开发过程及特点出发，软件开发一般采用软件生存周期的开发方法，有时采用开发原型以帮助了解用户需求。在软件分析与设计时，自上而下由全局出发全面规划分析，然后逐步设计实现。从系统分析出发，可将需求分析方法大致分为功能分解方法、结构化分析方法、信息建模法和面向对象的分析方法。

2.2 需求分析

学生宿舍管理系统是学校管理工作的一个重要组成部分，学生宿舍管理系统专门为宿舍管理人员研发的，在结合现代先进的宿舍管理模式和方法进行研发。通过网络技术把宿舍信息和学生信息等进行统一管理，同时该系统还必须满足实际工作需要。在实际运行中应具有可操作性强、为用户提供方便、提高工作效率等特点，这样可以把管理人员的繁重劳动简化，从而提高效率和准确率。此外，系统设计还应该考虑在现行条件下，充分保证其安全性及稳定性。

学生宿舍管理系统主要是对计算机数据库进行有效的管理。为了行之有效的进行宿舍管理，经过与宿舍管理科人员的反复交流沟通，确定本课题研究的学生宿舍管理系统首先应具备 5 项基本操作：

- (1) 用户必须输入正确地用户名和密码才能进入系统进行操作；
- (2) 系统可完整学生考勤记录信息；
- (3) 系统可提完整的宿舍楼信息情况；
- (4) 系统可提供查询功能，迅速准确的获取各种信息；
- (5) 系统可提供增加删除修改用户的功能；

2.3 功能需求分析

学生宿舍管理系统开发的总目标是实现学生宿舍管理的系统化、规范化和自动化。该系统的主要用户为系统管理员、宿舍管理员和学生，其中，系统管理员的权限最大，拥有的功能最多，宿舍管理员次之，学生最后。

因此可以将该系统划分为三个功能模块：一个是系统管理员模块，可以管理

宿舍管理员、管理学生、管理宿舍楼、管理寝室和管理考勤；一个是宿舍管理员模块，可以管理学生考勤记录和查看学生信息；最后是学生模块，可以查看自己的考勤记录。系统设计统一登录方式，以用户名和密码来区分登录系统的角色。

综上，学生宿舍管理系统既要实现对于宿舍的管理，又要实现对于用户的管理，还要兼顾到各个功能模块之间的协调合作，共同完成在学生宿舍管理系统的业务需求。下面采用 UML 用例模型对系统功能需求进行分析。

构建 UML 用例模型的第一步是确定模型中的使用者有哪些，确定使用者的原则有：谁是系统的维护者、谁是系统的参与者等。一般维护者处于系统内部，对系统有绝对的控制权；而参与者一般都位于系统的外部，处于系统的控制之外。

经过对学生宿舍管理系统的分析，可以确定本系统用例模型有三种，系统管理员、宿舍管理员和学生。下面分别对这三个角色的功能进行描述：

1) 系统管理员

系统管理员可以登录系统，进入系统后，可以添加、查看、修改和删除宿舍管理员信息；可以添加、查看、修改和删除学生信息；可以添加、查看、修改和删除宿舍楼信息；可以查看和删除学生考勤记录。宿舍管理员主要功能如下（图 2-1 为系统管理员用例图）：

- a. 登录
- b. 管理宿舍管理员
- c. 管理学生
- d. 管理宿舍楼
- e. 管理考勤
- f. 修改密码

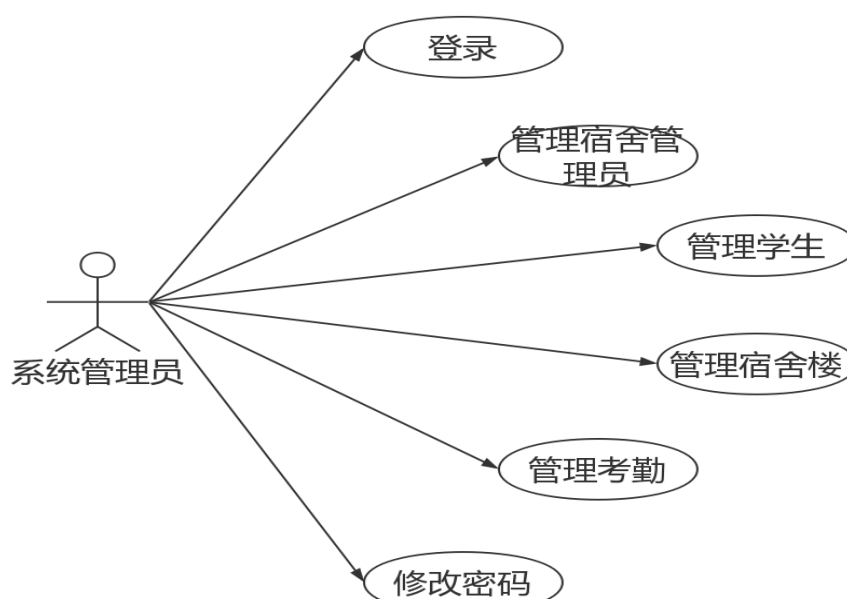


图 2-1 系统管理员用例图

2) 宿舍管理员

宿舍管理员可以登录系统，进入系统后，可以查看学生信息；可以添加、查看、修改和删除学生的考勤记录。宿舍管理员主要功能如下（图 2-2 为宿舍管理员用例图）：

- a. 登录
- b. 查看学生
- c. 管理考勤
- e. 修改密码

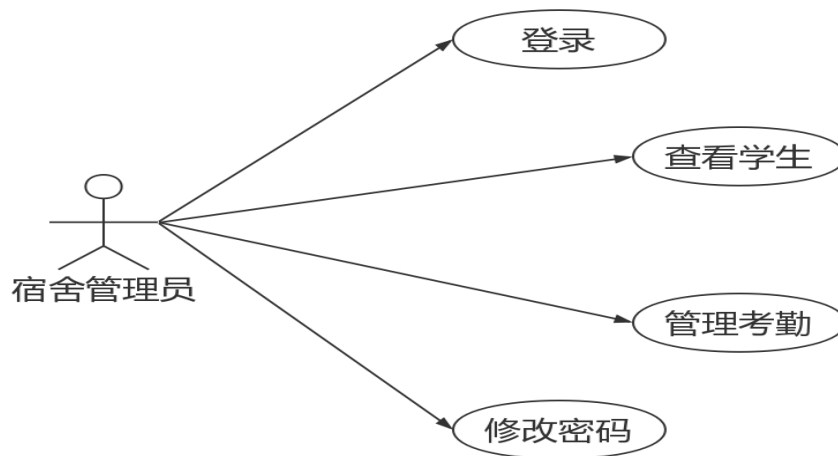


图 2-2 宿舍管理员用例图

3) 学生

学生可以登录系统，进入系统后，可以查看自己的考勤记录。学生主要功能如下（图 2-3 为学生用例图）：

- a. 登录
- b. 查看考勤记录
- e. 修改密码；

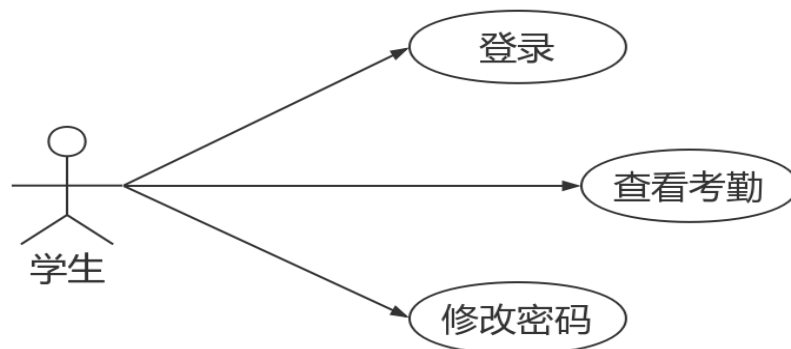


图 2-3 学生用例图

2.4 非功能需求分析

(1) 系统的实用性

系统的设计应该充分了解宿舍管理的实际情况，更好的满足实际宿舍管理工作中遇到的各种状况和要求。

(2) 系统的安全性

学生宿舍管理系统中存储了大量的学生信息，一旦信息遭到泄露会给学生的学习和生活带了较大的影响，因此开发中的管理系统需具备防火墙，并对其中的学生信息进行加密处理，从而保证系统数据的安全；此外应强调管理系统的规范操作，对系统数据库进行定期备份和检查。

(3) 系统的稳定性

学生宿舍与学生的生活和学习息息相关，学生宿舍管理系统的稳定运行能够在最大程度上降低对学生的不良影响，最大限度地发挥其宿舍管理的功能。

(4) 系统的易用性

系统设计过程中应对系统的不同界面进行统一设计和美化，尽量使系统中的帮助和提示功能趋于丰富和智能化，简化和降低系统对操作者的操作要求。

(5) 系统的可扩展性

系统要求采用纯 B/S 架构，接口开放，可在平台基础上进行二次开发或是定制，能适应不断发展的需求和实际情况。

2.5 环境需求分析

1. 系统开发平台

(1) MyEclipse

MyEclipse 企业级工作平台（MyEclipse Enterprise Workbench，简称 MyEclipse）是对 Eclipse IDE 的扩展，利用它我们可以在数据库和 JavaEE 的开发、发布，以及应用程序服务器的整合方面极大的提高工作效率。它是功能丰富的 JavaEE 集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持 HTML, Struts, JSF, CSS, Javascript, SQL, Hibernate。

对于以上每一种功能上的类别，在 Eclipse 中都有相应的功能部件，并通过一系列的插件来实现它们。MyEclipse 结构上的这种模块化，可以让我们在不影响其他模块的情况下，对任一模块进行单独的扩展和升级。简单而言，MyEclipse 是 Eclipse 的插件，也是一款功能强大的 JavaEE 集成开发环境，支持代码编写、配置、测试以及除错，MyEclipse6.0 以前版本需先安装 Eclipse。MyEclipse6.0 以后版本安装时不需安装 Eclipse。

总之，MyEclipse，是一个十分优秀的用于开发 Java, J2EE 的 Eclipse 插件集合，MyEclipse 的功能非常强大，支持也十分广泛，尤其是对各种开元产品的

支持十分不错。

(2) Tomcat 服务器

Tomcat 是一个小型的轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好 Apache 服务器，可利用它响应对 HTML 页面的访问请求。实际上 Tomcat 部分是 Apache 服务器的扩展，但它是独立运行的，所以当你运行 Tomcat 时，它实际上作为一个与 Apache 独立的进程单独运行的。

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现。因为 Tomcat 技术先进、性能稳定，而且免费，因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的 Web 应用服务器。

Tomcat 的环境主要有以下几方面技术优势：

1) Tomcat 中的应用程序是一个 WAR (Web Archive) 文件。WAR 是 Sun 提出的一种 Web 应用程序格式，与 JAR 类似，也是许多文件的一个压缩包。

2) 在 Tomcat 中，应用程序的部署很简单，你只需将你的 WAR 放到 Tomcat 的 webapp 目录下，Tomcat 会自动检测到这个文件，并将其解压。

3) Tomcat 不仅仅是一个 Servlet 容器，它也具有传统的 Web 服务器的功能：处理 html 页面。

4) Tomcat 也可以与其它一些软件集成起来实现更多的功能。

2. 运行环境

操作系统：Windows XP 以上版本。

服务器软件：Tomcat7.0 以上版本。

浏览器：IE、Fire Fox、Google Chrome。

数据库管理系统软件：MySQL5.7

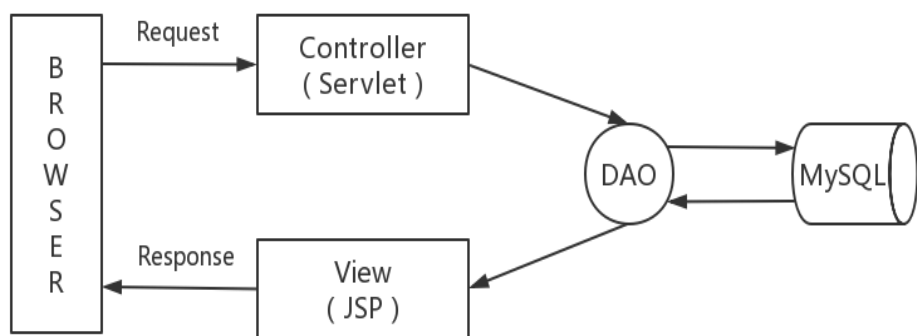
Java 开发包：JDK7.0 以上

第三章 系统结构分析与设计

3.1 引言

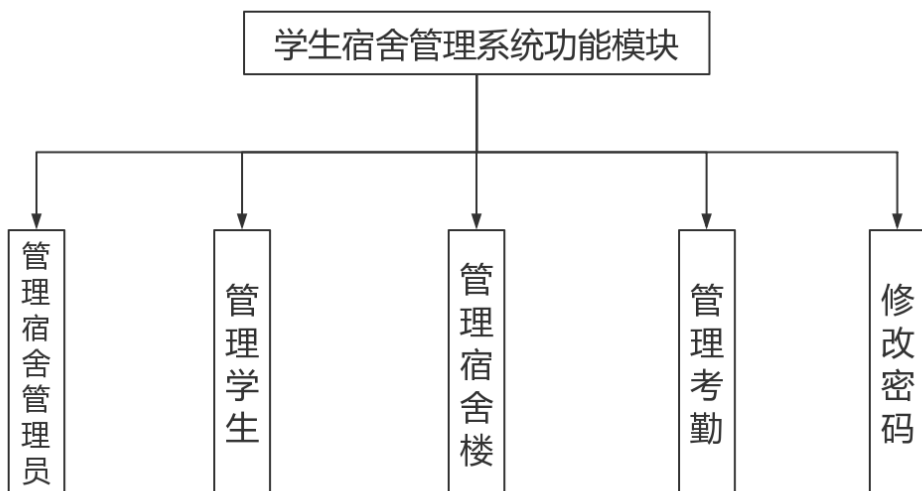
系统设计阶段的任务是设计软件系统的模块层次结构，设计数据库的结构以及设计模块的控制流程，其目的是明确软件系统“如何做”。在本章节中，主要是在系统分析的基础上，配合图例，对系统架构、系统功能模块和系统数据库进行了设计。

学生宿舍管理系统采用三层架构模式：表示层 (JSP 页面)，业务逻辑层 (DAO 模式)，控制层 (SERVLET)，程序中业务的控制由控制器来负责，通过 web.xml 配置文件可以由系统负责判断某个 JSP 页面该调用哪个控制类中的哪个方法来处理请求，控制器再将请求的相关参数传递给实例化好的 DAO 组件，进行页面跳转。图 4-1 为系统架构模式图：



3.2 系统模块结构图

从系统分析可知，本系统主要有管理宿舍管理员功能、管理学生功能、管理宿舍楼功能、管理考勤功能和修改密码功能。该系统的功能结构图如图 4-2 所示：



3.3 模块一

管理宿舍管理员：只有系统管理员才拥有此功能，系统管理员对宿舍管理员信息进行管理，可以添加宿舍管理员信息、修改宿舍管理员信息、查看宿舍管理员信息和删除宿舍管理员信息，其中添加宿舍管理员信息需要填写用户名、密码、姓名、性别和联系电话等信息。

3.4 模块二

管理学生：系统管理员拥有管理学生的全部功能，系统管理员对学生信息进行管理，可以添加学生信息、修改学生信息、查看学生信息和删除学生信息，其中添加学生信息需要填写学号、密码、姓名、性别、宿舍楼、寝室和联系电话等信息。宿舍管理员仅拥有查看学生信息功能，能够查看学生学号、姓名、性别、宿舍楼、寝室和电话等相关信息。

3.5 模块三

管理宿舍楼：只有系统管理员才拥有此功能，系统管理员对宿舍楼信息进行管理，可以添加宿舍楼信息、修改宿舍楼信息、查看宿舍楼信息和删除宿舍楼信息，其中添加宿舍楼信息需要填写宿舍楼名称和宿舍楼简介等信息。

3.5 模块四

管理考勤：系统管理员拥有管理考勤部分功能，只能查看和删除考勤记录。宿舍管理员拥有考勤管理的全部功能，可以添加考勤记录、修改考勤记录、查看考勤记录和删除考勤记录，其中，添加考勤记录需要填写学生的学号、日期和备注等信息。

3.6 模块五

修改密码：系统管理员、宿舍管理员和学生都拥有此功能，可以修改登录密码。

3.7 工作流程

根据系统分析，系统工作流程主要涉及学生工作流程、宿舍管理员工作流程和系统管理员工作流程。

学生工作流程：学生进入系统，输入学号和密码，经过系统验证成功后进入学生主页，学生只能查看自己的考勤记录，显示考勤日期、学号、姓名、宿舍楼和寝室等相关信息。

宿舍管理员工作流程：宿舍管理员进入系统后，输入用户名和密码，经过系统验证成功后方可进入宿舍管理员主页，然后宿舍管理员可以查看学生信息和管

理考勤。

系统管理员工作流程：系统管理员进入系统后，输入用户名和密码，经过系统验证成功后方可进入系统管理员主页，然后系统管理员可以管理宿舍管理员信息、管理学生信息、管理宿舍楼信息和管理考勤记录。

系统工作总体流程如图 3-3 所示：

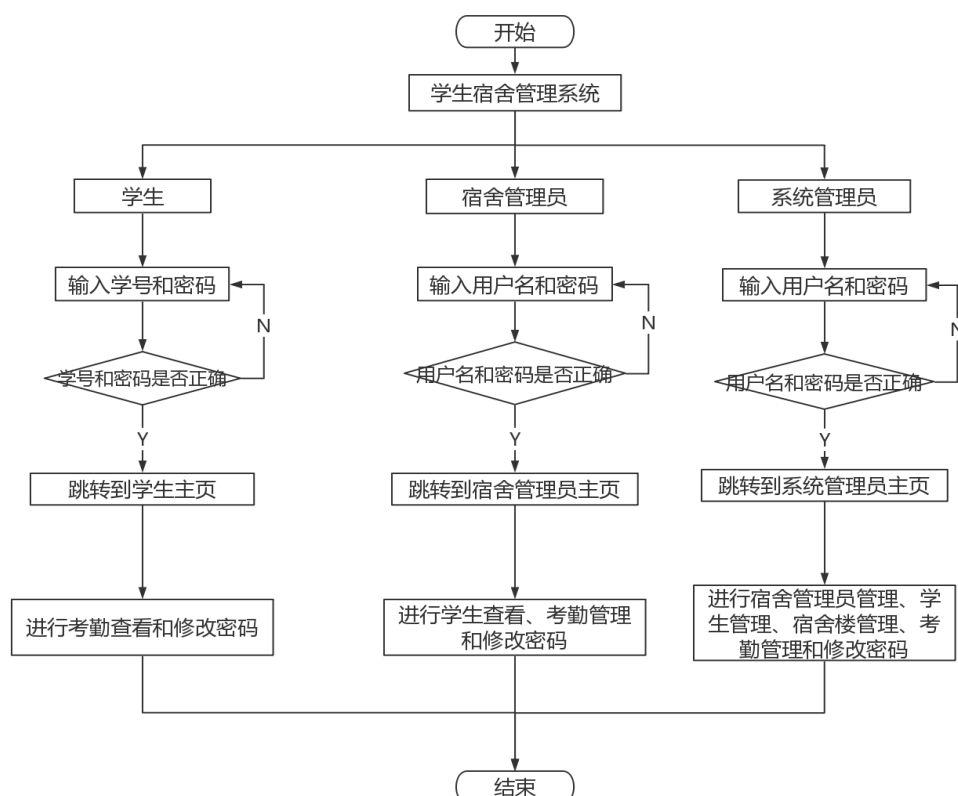


图 3-3 系统工作总体流程图

3.8 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^[9]。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。学生宿舍管理系统采用 MySQL 作为后台数据库开发工具。

3.8.1 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

信息世界中包含的基本概念有实体和联系。

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。

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)。

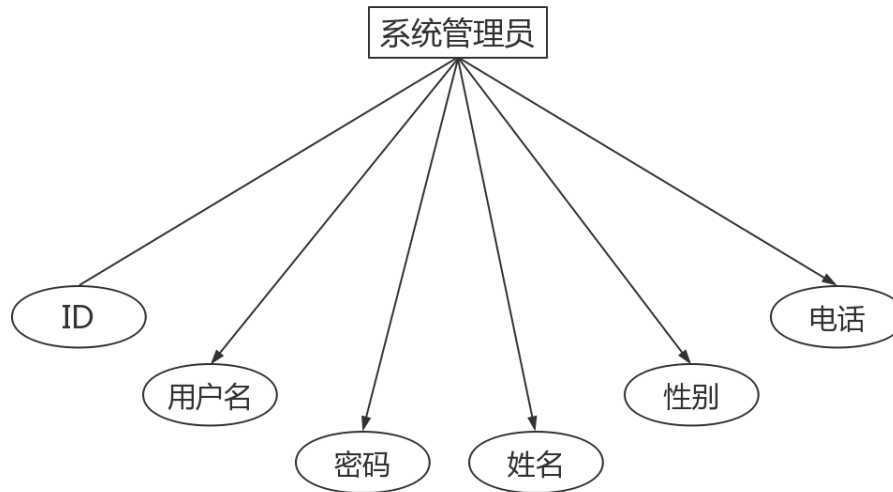
```

    erDiagram
        SYSTEM_ADMIN ||--o{ DORM_BUILDING : 管理
        DORM_ADMIN ||--o{ STUDENT : 管理
        DORM_ADMIN ||--o{ ATTENDANCE_RECORD : 记录
        STUDENT ||--o{ ATTENDANCE_RECORD : 查看
        DORM_BUILDING ||--o{ DORM_ROOM : 包含
  
```

根据系统分析的主要实体有：系统管理员、宿舍管理员、学生、宿舍楼、寝

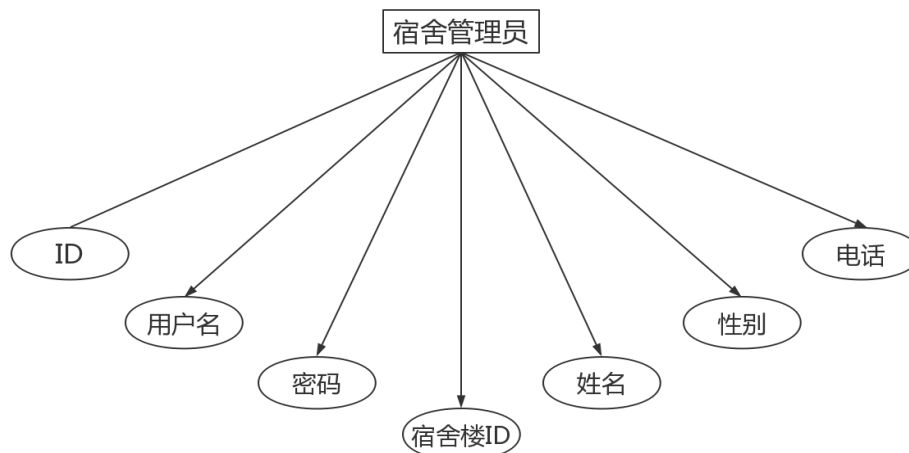
室和考勤记录。各个实体具体的描述属性图如下（实体属性在下图中并没有全部给出，因为属性过多的原因）：

1. 系统管理员实体



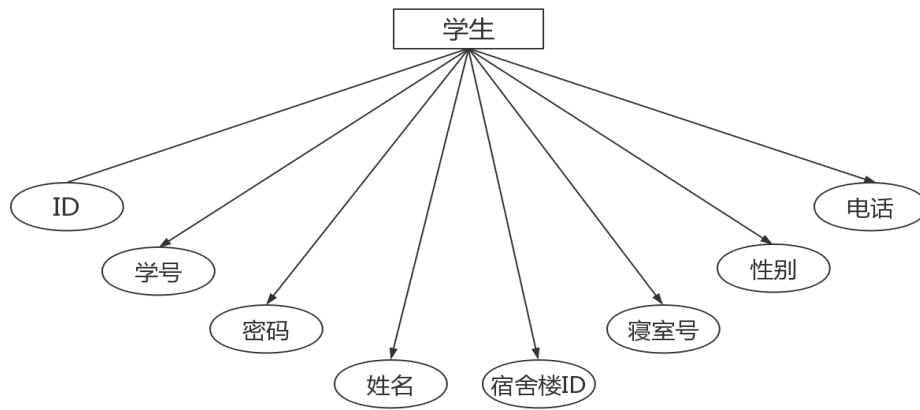
3-5 系统管理员实体图

2. 宿舍管理员实体



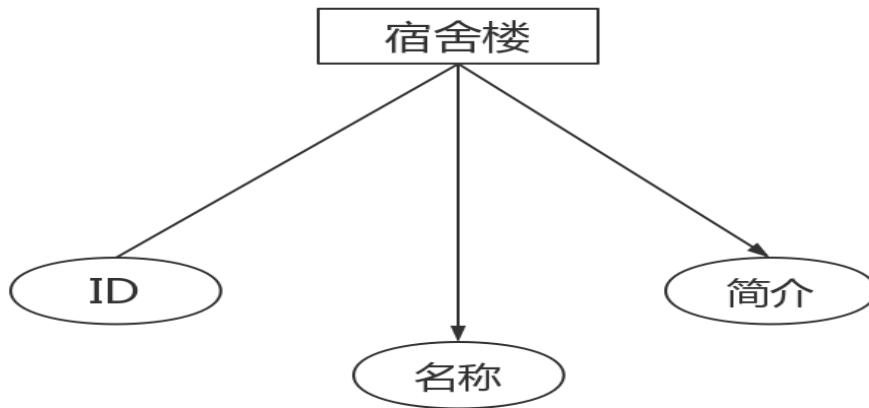
3-6 宿舍管理员实体图

3. 学生实体



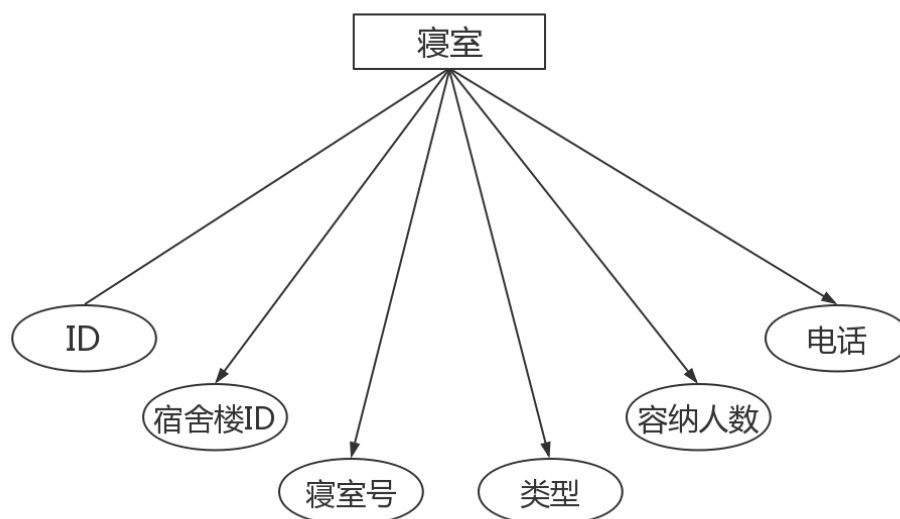
3-7 学生实体图

4. 宿舍楼实体



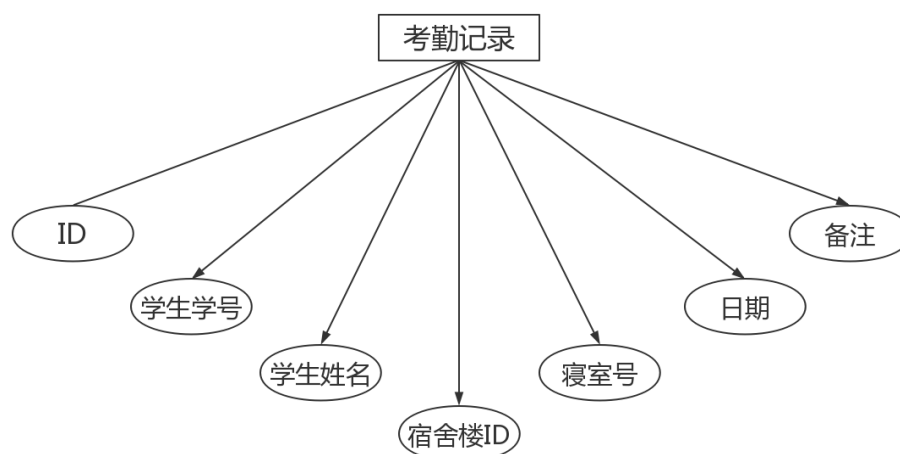
3-8 宿舍楼实体图

5. 寝室实体



3-9 寝室实体图

6. 考勤记录实体



3-10 考勤记录实体图

3.8.2 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 db_dorm，数据库包括：①系统管理员表 (t_admin) ②宿舍管理表 (t_dormmanager) ③学生表 (t_student) ④宿舍楼表 (t_dormbuild) ⑤寝室表 (t_dorm) ⑥考勤记录表 (t_record)。各表数据结构如下：

(1) 系统管理员表(t_admin), 存储系统管理员详细信息

字段名称	数据类型	主键	是否空	说明
adminId	int(11)	Y	N	系统管理员 ID
userName	varchar(20)	N	Y	用户名
password	varchar(20)	N	Y	密码
name	varchar(20)	N	Y	姓名
sex	varchar(10)	N	Y	性别
tel	varchar(20)	N	Y	电话

(2) 宿舍管理表(t_dormmanager), 存储宿舍管理员详细信息

字段名称	数据类型	主键	是否空	说明
dormManId	int(11)	Y	N	宿舍管理员 ID
userName	varchar(20)	N	Y	用户名
password	varchar(20)	N	Y	密码
dormBuildId	int(11)	N	Y	宿舍楼 ID
name	varchar(20)	N	Y	性别
sex	varchar(20)	N	Y	性别
tel	varchar(20)	N	Y	电话

(3) 学生表(t_student), 存储学生详细信息

字段名称	数据类型	主键	是否空	说明
studentId	int(11)	Y	N	学生 ID
stuNum	varchar(20)	N	Y	学号
password	varchar(20)	N	Y	密码
name	varchar(20)	N	Y	姓名
dormBuildId	int(11)	N	Y	宿舍楼 ID
dormName	varchar(11)	N	Y	寝室号
sex	varchar(10)	N	Y	性别
tel	varchar(15)	N	Y	电话

(4) 宿舍楼表(t_dormbuild), 存储宿舍楼详细信息

字段名称	数据类型	主键	是否空	说明
dormBuildId	int(11)	Y	N	宿舍楼 ID
dormBuildName	varchar(20)	N	Y	宿舍楼名称
dormBuildDetail	varchar(50)	N	Y	简介

(5) 寝室表 (t_dorm), 存储寝室详细信息

字段名称	数据类型	主键	是否空	说明
dormId	int(11)	Y	N	寝室 ID
dormBuildId	int(11)	N	Y	宿舍楼 ID
dormName	varchar(20)	N	Y	寝室号
dormType	varchar(20)	N	Y	类型
dormNumber	int(11)	N	Y	容纳人数
dormTel	varchar(20)	N	Y	电话

(6) 考勤记录表 (t_record), 存储考勤记录详细信息

字段名称	数据类型	主键	是否空	说明
recordId	int(11)	Y	N	考勤记录 ID
studentNumber	varchar(20)	N	Y	学号
studentName	varchar(30)	N	Y	学生姓名
dormBuildId	int(11)	N	Y	宿舍楼 ID
dormName	varchar(11)	N	Y	寝室号
date	date	N	Y	日期
detail	varchar(50)	N	Y	备注

第四章 关键技术研究

4.1 实现技术路线

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。

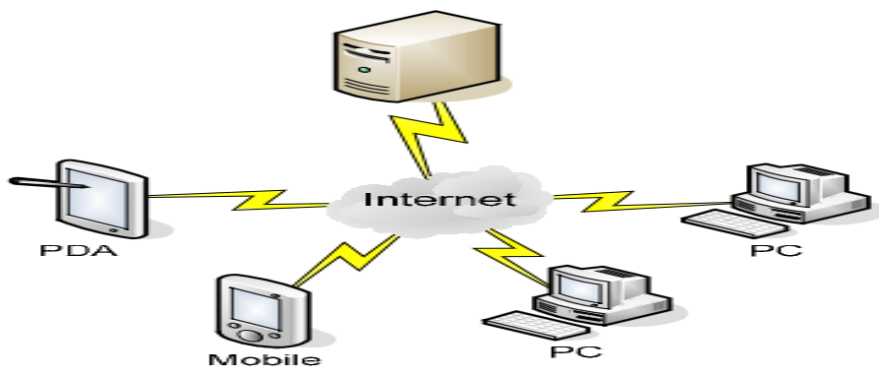


图 4-1 B/S 模式应用系统网络结构图

各个管理模块的实现基本相似,使用的技术路线见下图:

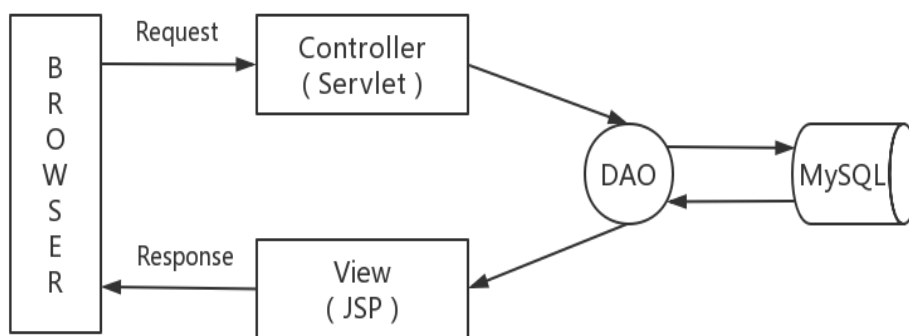


图 4-2 管理模块技术路线

4.2 关键技术研究

系统采用三层架构模式：表示层（JSP 页面），业务逻辑层（DAO 模式），控制层（SERVLET），程序中业务的控制由控制器来负责，通过 web.xml 配置文件可以由系统负责判断某个 JSP 页面该调用哪个控制类中的哪个方法来处理请求，控制器再将请求的相关参数传递给实例化好的 DAO 组件，进行页面跳转。

4.2.1 关键技术一：JavaScript

JavaScript 是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言。同时也是一种广泛用于客户端 Web 开发的脚本语言，常用来给 HTML 网页添加动态功能，比如响应用户的各种操作。它最初由网景公司(Netscape)的Brendan Eich 设计，是一种动态、弱类型、基于原型的语言，内置支持类。

Javascript 语言与 Java 语言在语法上比较相似，但随着对 Javascript 的深入了解后会发现，它们说到底是两种语言。

4.2.2 关键技术二：JDBC

JDBC 的全称是 Java Database Connectivity，即 Java 数据库连接，它是一种可以执行 SQL 语句的 Java API。程序可通过 JDBC API 连接到关系数据库，并使用结构化查询语言（SQL，数据库标准的查询语言）来完成对数据库的查询、更新。与其他数据库编程环境相比，JDBC 为数据库开发提供了标准的 API，所以使用 JDBC 开发的数据库应用可以跨平台运行，而且可以跨数据库（如果全部使用标准的 SQL）。也就是说，如果使用 JDBC 开发一个数据库应用，则应该应用既可以在 Windows 平台上运行，也可以在 Unix 等其他平台上运行，既可以使用 MySQL 数据库，也可以使用 Oracle 等数据库，而程序无需进行任何修改。

通过使用 JDBC，就可以使用同一种 API 访问不同的数据库。换言之，有了

JDBC API, 就不必为访问 Sybase 数据库学习一组 API, 为访问 Oracle 数据库又学习一组 API, 软件开发人员可以使用标准的 API 编写需要的应用程序, 随后根据数据库的不同, 加入相应的不同的数据库驱动程序就可以正常工作。

对于 Java 语言的各种跨平台性, 其实它们都采用相似的结构, 因为它们都需要让相同的程序在不同的平台上运行, 所以需要中间的转换程序(为了实现 Java 程序的跨平台性, Java 为不同操作系统提供了不同虚拟机)。

4.2.3 关键技术三: Servlet

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。

最早支持 Servlet 标准的是 JavaSoft 的 Java Web Server, 此后, 一些其它的基于 Java 的 Web 服务器开始支持标准的 Servlet。

最早支持 Servlet 技术的是 JavaSoft 的 Java Web Server。此后, 一些其它的基于 Java 的 Web Server 开始支持标准的 Servlet API。Servlet 的主要功能在于交互式地浏览和修改数据, 生成动态 Web 内容。这个过程为:

1. 客户端发送请求至服务器端;
2. 服务器将请求信息发送至 Servlet;
3. 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;
4. 服务器将响应返回给客户端。

Servlet 看起来像是通常的 Java 程序。Servlet 导入特定的属于 Java Servlet API 的包。因为是对象字节码, 可动态地从网络加载, 可以说 Servlet 对 Server 就如同 Applet 对 Client 一样, 但是, 由于 Servlet 运行于 Server 中, 它们并不需要一个图形用户界面。从这个角度讲, Servlet 也被称为 FacelessObject。

一个 Servlet 就是 Java 编程语言中的一个类, 它被用来扩展服务器的性能, 服务器上驻留着可以通过“请求-响应”编程模型来访问的应用程序。虽然 Servlet 可以对任何类型的请求产生响应, 但通常只用来扩展 Web 服务器的应用程序。

4.2.4 关键技术四: JSP

JSP 本是一个简化的 Servlet 设计, 它是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准。JSP 技术有点类似 ASP 技术, 它是在传统的网页 HTML (标准通用标记语言的子集) 文件(*.htm, *.html)中插入 Java 程序段(Scriptlet)和 JSP 标记(tag), 从而形成 JSP 文件, 后缀名为(*.jsp)。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。

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。网页还能通过 tags 和 scriptlets 访问存在于服务端的资源的应用逻辑。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(JavaServer Pages)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。

4.2.5 关键技术五：MVC 设计模式

MVC 是一种使用 MVC (Model View Controller 模型-视图-控制器) 设计创建 Web 应用程序的模式，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中：

Model (模型) 表示应用程序核心 (比如数据库记录列表)。

View (视图) 显示数据 (数据库记录)。

Controller (控制器) 处理输入 (写入数据库记录)。

MVC 模式同时提供了对 HTML、CSS 和 JavaScript 的完全控制。

Model (模型) 是应用程序中用于处理应用程序数据逻辑的部分。

通常模型对象负责在数据库中存取数据。

View (视图) 是应用程序中处理数据显示的部分。

通常视图是依据模型数据创建的。

Controller (控制器) 是应用程序中处理用户交互的部分。

通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

MVC 分层有助于管理复杂的应用程序，因为我们可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。MVC 分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。本系统使用的是最典型的 JSP + servlet + javabean 的模式。

第五章 系统实现

5.1 系统实现介绍

1. 首页：

系统登录页面如下，可供系统管理、宿舍管理员和学生登录系统，需要输入用户名和密码。



图 5-1 系统登录页面

使用说明：

系统管理员

用户名：admin

密码：111

宿舍管理员

用户名：manager1(manager1-7)

密码：123

学生

用户名：(001-010)

密码：233

2. 系统管理员：

系统管理员主页如下，显示宿舍管理员管理、学生管理、宿舍楼管理、缺勤记录、修改密码和退出系统等功能栏。



图 5-2 系统管理员主页

系统管理员管理宿舍管理员，可以修改、添加、查看和删除宿舍管理员，以下为查看宿舍管理员信息，包括宿舍管理员的编号、姓名、性别、电话、所属宿舍楼、用户名等信息。



图 5-3 管理宿舍管理员

系统管理员添加宿舍管理员，需要填写用户名、密码、姓名、性别和联系电话等信息。



The image shows a web form for adding a dormitory administrator. It contains six input fields: username, password, repeat password, name, gender, and contact phone. The 'username' field is highlighted with a blue border. Below the fields are two buttons: '保存' (Save) and '返回' (Return).

*用户名：	<input type="text"/>
*密码：	<input type="password"/>
*重复密码：	<input type="password"/>
*姓名：	<input type="text"/>
*性别：	<input type="text" value="男"/>
*联系电话：	<input type="text"/>

图 5-4 添加宿舍管理员

系统管理员管理学生，可以修改、添加、查看和删除学生，以下为查看学生信息，包括学生的学号、姓名、性别、电话、所属宿舍楼、寝室等信息。

学生管理

添加

全部宿舍楼

▼

姓名

▼

搜索

学号 ▲	姓名 ◆	性别 ◆	宿舍楼 ◆	寝室 ◆	电话	操作
001	A	女	橘园三舍	102	111	<div>修改</div> <div>删除</div>
002	B	男	橘园五舍	203	123	<div>修改</div> <div>删除</div>
003	C	男	橘园六舍	304	123	<div>修改</div> <div>删除</div>
004	D	男	橘园四舍	405	123	<div>修改</div> <div>删除</div>
005	E	男	橘园四舍	506	123	<div>修改</div> <div>删除</div>
006	F	女	橘园一舍	607	111	<div>修改</div> <div>删除</div>
007	G	女	橘园一舍	809	111	<div>修改</div> <div>删除</div>
008	H	男	橘园六舍	910	123	<div>修改</div> <div>删除</div>
009	I	男	橘园五舍	110	123	<div>修改</div> <div>删除</div>
010	J	男	橘园七舍	111	123	<div>修改</div> <div>删除</div>

图 5-5 管理学生

系统管理员添加学生，需要填写学号、密码、姓名、性别、宿舍楼、寝室和联系电话等信息。

*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

图 5-6 添加学生

系统管理员管理宿舍楼，可以修改、添加、查看和删除宿舍楼，以下为查看宿舍楼信息，包括宿舍楼的编号、名称和简介等信息。

宿舍楼管理

添加

名称:搜索

编号	名称	简介	操作
1	橘园一舍	1栋	<div>管理员修改删除</div>
2	橘园二舍	2栋	<div>管理员修改删除</div>
3	橘园三舍	3栋	<div>管理员修改删除</div>
4	橘园四舍	4栋	<div>管理员修改删除</div>
5	橘园五舍	5栋	<div>管理员修改删除</div>

图 5-7 管理宿舍楼

系统管理员添加宿舍楼，需要填写宿舍楼名称和简介等信息。

*名称：

简介：

保存

返回

图 5-8 添加宿舍楼

系统管理员管理考勤记录，可以查看和删除考勤记录，以下为查看考勤记录信息，包括日期、学生学号、姓名、宿舍楼、寝室和备注等信息。

缺勤记录

起始日期	✕	终止日期	✕	全部宿舍楼	姓名	搜索
日期	学号	姓名	宿舍楼	寝室	备注	操作
2019-06-01	001	A	橘园三舍	102	缺勤	删除
2019-06-03	014	B	橘园五舍	203	缺勤	删除
2019-06-03	013	C	橘园六舍	304	缺勤	删除
2019-06-04	006	D	橘园四舍	405	请假	删除
2019-06-05	004	E	橘园四舍	506	缺勤	删除
2019-06-05	004	F	橘园一舍	607	缺勤	删除

图 5-9 管理考勤记录

3. 宿舍管理员：

宿舍管理员主页如下，显示学生查看、缺勤记录、修改密码和退出系统等功能栏。



图 5-10 宿舍管理员主页

宿舍管理员管理学生，可以查看学生信息和给学生添加考勤记录，以下为查看学生信息，包括学生的学号、姓名、性别、电话、所属宿舍楼、寝室等信息。

学生管理

橘园一舍 姓名 搜索

学号	姓名	性别	宿舍楼	寝室	电话	操作
006	F	女	橘园一舍	607	111	添加缺勤记录
007	G	女	橘园一舍	809	111	添加缺勤记录

图 5-11 管理学生

宿舍管理员考勤记录，可以添加、修改、查看和删除考勤记录，以下为查看考勤记录，包括日期、学生的学号、姓名、所属宿舍楼、寝室和备注等信息。

缺勤记录

[添加](#) 起始日期 终止日期 橘园一舍 姓名 搜索

日期	学号	姓名	宿舍楼	寝室	备注	操作
2019-06-05	004	F	橘园一舍	607	缺勤	修改 删除

图 5-12 管理考勤记录

宿舍管理员添加考勤记录，需要填写学号、日期和备注等信息。

添加缺勤记录

*学号:

*日期:

*备注:

[保存](#) [返回](#)

图 5-13 添加考勤记录

4. 学生：

学生主页如下，显示缺勤记录、修改密码和退出系统等功能栏。



图 5-14 学生主页

学生只能查看考勤记录，以下为查看考勤记录，包括日期、学生的学号、姓名、所属宿舍楼、寝室和备注等信息。

缺勤记录

起始日期

✕

⋮

终止日期

✕

⋮

搜索

日期	学号	姓名	宿舍楼	寝室	备注
2019-06-01	001	A	橘园三舍	102	缺勤

图 5-15 查看考勤记录

5.2 系统实现的不足

1. 系统实现的功能不足：

登陆界面未给出验证码功能。

因为是内部系统，又没有想到验证学生身份的方式，所以没有给出学生账号的注册功能。

未实现在线人数统计和文件图片上传功能。

2. 技术与设计上的不足：

未能提供 SQL 脚本，实现对系统数据表和示例数据的初始化工作。

未能通过 HttpSession 对象实现用户会话管理。

未能提供可通过 ant 进行编译、打包和生成 war 包的 build.xml 文件。