毕业论文

**基于SSM学生宿舍管理系统**的实现

专业：**软件工程**

学号：**1804230913**

姓名： **高铭阳**

武汉纺织大学数学与计算机学院

目录

[1．1 系统开发的目的 5](#_Toc74574748)

[2 系统技术概括 5](#_Toc74574749)

[2．1 系统前端技术概括 5](#_Toc74574750)

[2.1.1 Jquery 及 Ajax 5](#_Toc74574751)

[2．2 系统后端技术概括 6](#_Toc74574752)

[2.2.1 MySQL 数据库 6](#_Toc74574753)

[2.2.2 Tomcat 服务器 8](#_Toc74574754)

[2.2.3 MVC 设计模式 8](#_Toc74574755)

[2.2.4 Spring 框架 9](#_Toc74574756)

[2.2.5 SpringMVC 框架 10](#_Toc74574757)

[2.2.6 MyBatis 框架 10](#_Toc74574758)

[3 系统设计 12](#_Toc74574759)

[3．1 系统功能介绍 12](#_Toc74574760)

[3．2 系统设计 12](#_Toc74574761)

[3.2.1 数据库设计 12](#_Toc74574762)

[3.2.2 数据表关系 15](#_Toc74574763)

[4 系统实现 15](#_Toc74574764)

[4.1.1 学生管理 15](#_Toc74574765)

[4.1.2 班级管理 17](#_Toc74574766)

[4.1.3 宿舍管理 17](#_Toc74574767)

[4.1.4 卫生管理 18](#_Toc74574768)

[4.1.5 访客管理 19](#_Toc74574769)

[4.1.6 管理员管理 19](#_Toc74574770)

[5 总结与展望 19](#_Toc74574771)

[参考文献 20](#_Toc74574772)

1 引言

## 1．1 系统开发的目的

学生宿舍管理系统主要对学校宿舍相关事务进行管理，管理内容包括学员信息管理、班级管理、宿舍管理、卫生管理、访客管理、管理员管理等。通过系统管理更好的帮助学校和宿舍完成数据信息整合管理工作.

# 系统技术概括

## 系统前端技术概括

### Jquery 及 Ajax

Jquery 是一个快速、简洁的 JavaScript 框架，其倡导写更少的代码，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[4]。

Jquery 的核心特性可以总结为：具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的 CSS 选择器，并且可对 CSS 选择器进行扩展；拥有便捷的插件扩展机制和丰富的插件。

一般而言，选择使用 Jquery 是因为它的以下特点：

1. 更方便的选择器

Jquery 的选择机制构建于 CSS 的选择器，它提供了快速查询 DOM 文档中元素的能力，而且大大强化了 JavaScript 中获取页面元素的方式。相比于原本的选择器，Jquery 使用更少的语句即能达到相同的效果，同时选择的精度也要更高。

1. 提供页面动态效果

Jquery 中内置了一系列的动画效果，比如淡入淡出、元素移除等动态特效。这些特效增强了用户在浏览网页时的体验感，许多的网站都有使用。

1. 创建 Ajax 无刷新网页

Ajax 是异步的 JavaScript 和 XML 的简称，可以协助开发出非常灵敏无刷新的网页

1. 对 JavaScript 语言的增强

Jquery 提供了对基本 JavaScript 结构的增强，比如元素迭代和数组处理等操作。

1. 增强的事件处理

Jquery 提供了各种页面事件，它可以避免程序员在 HTML 中添加太多事件

处理代码，同时它的事件处理器消除了各种浏览器兼容性问题。

Ajax 是 Jquery 框架的一部分功能简称，它在网页制作中应用广泛，其具体功能在上面叙述 Jquery 的特点中也有提到，即可以使我们发送请求时进行异步操作，在不刷新页面的情况下对后台的处理产生响应。

不借助 Ajax 的话，我们在页面逻辑操作时，必须要进行从当前页面到控制视图层再返回当前页面的多次跳转，很影响用户的操作体验。而借助 Ajax，后端在处理请求时用户依然可以在前端页面继续进行浏览，直到响应返回前端用户再进行操作，整个过程更加连贯，用户的操作体验自然也会更好。

## 系统后端技术概括

### MySQL 数据库

数据库分为关系型数据库和非关系型数据库。

关系型数据库指的是采用了关系模型来组织数据的数据库，拥有关系，元组，属性，域，关键字，关系模式等概念。关系型数据库比较直观，更容易被理解，使用起来较为方便，易于维护。但关系型数据库性能欠佳，在处理大量关系复杂的数据时，效率很低。我们常用的关系型数据库有 [Oracle](https://www.oracle.com/database/index.html)[，Microsoft SQL Server，](https://www.microsoft.com/en-us/sql-server/)

MySQL[，PostgreSQL](https://www.postgresql.org/)[，DB2](https://www.ibm.com/analytics/us/en/db2/)[，Microsoft Access](https://products.office.com/zh-cn/access)[，SQLite](https://www.sqlite.org/) [等](https://www.sqlite.org/)[5]。

非关系数据库指的是指与关系型数据库相对的非关系型的，分布式的数据库。

它以键值对存储，且结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，不局限于固定的结构。非关系型的数据库结构简单，处理数据时可以节省一点的时间和开销。但它仅限处理一些简单的没有复杂关联关系的数据库，不适合持久存储海量数据。由于非关系型数据库都是针对某些特定的应用需求出现的，因此，不同非关系数据库的特长也有所不同，主要分为以下几类：面向高性能并发读写的 key-value 数据库，主流代表为 [Redis，](https://redis.io/) [Amazon DynamoDB；](https://aws.amazon.com/cn/dynamodb/)面向海量数据访问的面向文档数据库，主流代表为 [MongoDB；](https://www.mongodb.com/)面向搜索数据内容的搜索引擎，主流代表为 [Splunk](https://www.splunk.com/zh-hans_cn)[，Solr；](https://lucene.apache.org/solr/)面向可扩展性的分布式数据库，主流代表为 [Cassandra](https://cassandra.apache.org/)[，HBase[](https://hbase.apache.org/)6]。

MySQL 是典型的关系型数据库，它由瑞典 MySQL AB 公司开发，在被收

购之后属于 [Oracle](https://baike.baidu.com/item/Oracle) 旗下产品。与其他关系型数据库相同，MySQL 将数据保存在不同的表中，确保效率并提高了灵活性。MySQL 使用 C 和 [C++编](https://baike.baidu.com/item/C++)写，并使用了多种[编译器进](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8)行测试，保证了[源代码的](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81)可移植性，使得它能够支持大多数的操作系统。

MySQL 所使用的操作语言 是用于访问[数据库的](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)最常用标准化语言SQL 语

言。由于其体积小、速度快、总体拥有成本低，尤其是[开放源码这](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E7%A0%81/7176422)一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。虽然 MySQL 与其他的大型数据库比有所不足，但对于个人开发者，MySQL 数据库包含了个人开发所需的全部功能，并且它是免费开放的，降低了学习成本。也正是出于上述原因，我在房屋租赁系统中选择了 MySQL 数据库为项目的数据库。

### Tomcat 服务器

搭建 Web 项目，服务器是必不可少的。而 Tomcat 技术先进、性能稳定，而且免费，是个人 JavaWeb 开发的首选。

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，由 [Apache、](https://baike.baidu.com/item/Apache/6265) Sun 和其他一些公司及个人共同开发而成，属于轻量级应用[服务器。](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)它更多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。在 Tomcat 配置完成后，开发者就可以在服务器上运行 JSP 页面和 Servlet。它具有处理 HTML 页面请求的功能，同时还是一个 Servlet 和 JSP 容器。

我在项目中搭建的是 Tomcat 7.0 的版本，个人使用时经过一些简单的配置

配合 Idea、Eclipse 等 Java 开发软件就能较为方便地使用。

### MVC 设计模式

MVC 是软件工程中的一种架构模式，它的全称是 Model View Controller，其实即是 Model（模型），View（视图），Controller（控制器）的组合。它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。

在 Web 应用程序中，MVC 设计模式的三个部分有着各自的业务逻辑：

Model（模型层）：Model（模型）是应用程序中用于处理应用程序数据逻辑的单位对象，通常模型对象负责在数据库中存取数据。举例来说，数据库中每一个数据表都可以对应一个模型，该模型与数据表拥有相同的属性。在我们需要在该数据表中获取数据时，该模型就会被使用。

View（视图层）：View（视图）是应用程序处理数据后进行显示的部分，它提供一个可视化的部分，来较为直观的在用户数据处理完成后产生响应。

Controller（控制层）：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，因为您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。

我在项目中使用的 SSM 框架也是基于 MVC 设计模式的。SSM 是 Spring、

SpringMVC、MyBatis 的首字母缩写，因为 SpringMVC 是 Spring 中的部分内容，所以 SSM 框架可以说由 Spring、MyBatis 两个开源框架整合而成。SSM 框架经常作为数据源较简单的 Web 项目的框架，在下面我也将对 SSM 框架进行一个大概的描述。

### Spring 框架

Spring 框架用于简化软件开发而创建，是 JavaWeb 开发的常用框架。它的主要功能是使用 JavaBean 代替 EJB（Enterprise Java Beans 的技术简称，是之前的应用开发组件）。

在 Spring 框架之前，开发过程中编写的代码经常出现类似或者重复的情

况，这导致开发过程复杂，开发效率和性能都不太理想。Spring 的出现就是为了解决早期这种应用开发的情况，它致力于提供一个以统一的、高效的方式构造整个应用，并且可以将单层框架以最佳的组合揉和在一起建立一个连贯的体系。

Spring 作为一个容器框架，其最大的两个特性是控制反转（IOC）和面向切面（AOP）。

控制反转：控制反转是 Spring 框架作为容器框架的主要特性。原本我们在

Java 开发中，我们要使用对象时都需要自身创建对象再进行操作。而使用

Spring 框架后，所需对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象，方便开发者使用对象的同时节省空间，促进了松耦合。

面向切面：面向切面[是函数式编程的](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0%E5%BC%8F%E7%BC%96%E7%A8%8B/4035031)一种衍生范型。利用面向切面可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的[耦合度降](https://baike.baidu.com/item/%E8%80%A6%E5%90%88%E5%BA%A6/2603938)低，提高程序的可重用性，同时提高了开发的效率。

在项目中使用 Spring 框架需要在项目中配置部分的配置文件，一般在项目中会配置文件搭配注解使用，简化配置并方便进行管理。

### SpringMVC 框架

SpringMVC 是 Spring 框架的后续产品，它在 SSM 框架中的主要作用作为 Controller（控制器）部分处理 View（视图层）。SpringMVC 分离了控制器、模型对象、过滤器以及处理程序对象的角色，这种分离让它们更容易进行定制。

在实际的使用中，它的作用类似于 Servlet。配置文件配置完成后，创建对应的文件，并一一添加注解，通过这些注解的引用，前端的请求就能转到对应的方法进行处理。但不同的是，SpringMVC 使用注解，一个文件可以处理多个请求，在处理整个项目的请求时，更加轻松，管理起来也更加实用。

### MyBatis 框架

MyBatis 框架是基于 Java 的持久层框架，用于定制化 sql、存储过程以及高级映射。

原本在 JavaWeb 开发中需要借助 JDBC 工具实现对数据库的 sql 语句操作，

但这样设置 sql 的参数并不方便。MyBatis 消除了几乎所有的 [JDBC](https://baike.baidu.com/item/JDBC) [代](https://baike.baidu.com/item/JDBC)码和参数的手工设置以及[结果集的](https://baike.baidu.com/item/%E7%BB%93%E6%9E%9C%E9%9B%86)检索， 使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的模型映射成数据库中的记录。

相比传统的持久层操作，MyBatis 框架下的持久层操作更为简单易学。只需要几个 jar 包和几个配置文件将可以成功搭建出所需的环境。搭建完环境后，开发者就可以开始 sql 映射文件的编写。由于 MyBatis 把 sql 语句统一在这些 sql 映射文件中，实际操作中的编写和管理起来也更为方便。

MyBatis 框架的框架架构流程如下：

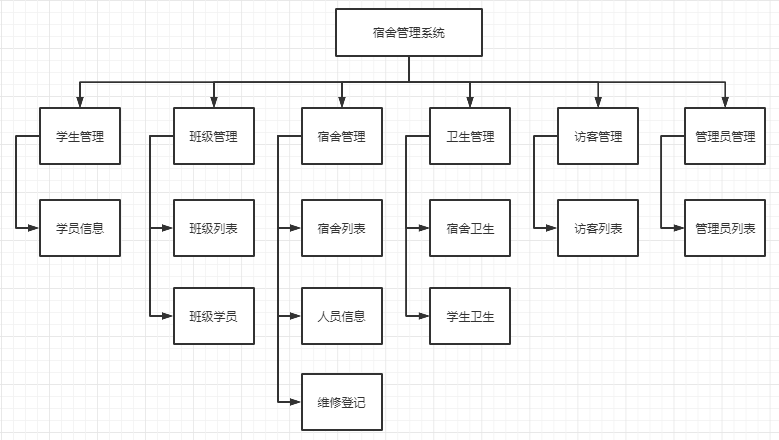
1. 加载配置：在开启项目后，服务器会读取我们实现配置的文件以便在后面进行启用。
2. sql 解析：当前台发送请求调用了事前设定的持久层对象的方法时，会根据方法名称在 sql 映射文件中找出 ID 与之相同的部分，之后传入参数，进行解析，得到最终要执行的 sql 语句和参数。
3. sql 执行：将之前 sql 解析得到的参数整合到 sql 语句中，然后执行 sql

语句

1. 结果映射：将操作数据库的结果按照映射的配置进行转换，可以转换成 HashMap、JavaBean 或者基本数据类型，并将最终结果返回。

# 系统设计

## 系统功能介绍



学生管理：展示学生基本信息，并可以完成学员信息添加、检索导出等操作

班级管理：班级列表展示，班级信息添加、导出、检索等

宿舍管理：宿舍列表展示、宿舍人员分配、维修等级

卫生管理：宿舍卫生和学员卫生的检查情况

访客管理：外来访客登记，功能包括信息添加、检索、导出等

管理员管理：宿舍管理员列表展示，人员添加、导出、检索等

## 系统设计

### 数据库设计

t\_admin（管理员表）

|  |  |  |
| --- | --- | --- |
| a\_id | int(11) | 是 |
| a\_username | varchar(20) | 是 |
| a\_password | varchar(50) | 是 |
| a\_name | varchar(20) | 否 |
| a\_phone | bigint(20) | 否 |
| a\_power | varchar(20) | 否 |
| a\_describe | varchar(40) | 否 |

t\_class（班级表）

|  |  |  |
| --- | --- | --- |
| c\_id | int(11) | 是 |
| c\_classid | int(11) | 是 |
| c\_classname | varchar(30) | 否 |
| c\_counsellor | varchar(30) | 否 |

d\_dormgrade（宿舍卫生表）

|  |  |  |
| --- | --- | --- |
| g\_id | int(11) | 是 |
| d\_id | int(11) | 是 |
| d\_dormbuilding | varchar(20) | 否 |
| d\_grade | int(11) | 否 |
| create\_time | datetime | 否 |
| update\_time | datetime | 否 |

d\_dormitoryinfo（宿舍表）

|  |  |  |
| --- | --- | --- |
| d\_id | int(11) | 是 |
| s\_dormitoryid | int(11) | 是 |
| d\_dormbuilding | varchar(20) | 否 |
| d\_bedtotal | varchar(20) | 否 |
| d\_bed | varchar(20) | 否 |
| a\_name | varchar(20) | 否 |

d\_dormrepair（维修表）

|  |  |  |
| --- | --- | --- |
| r\_id | int(11) | 是 |
| d\_id | int(11) | 是 |
| d\_dormbuilding | varchar(20) | 是 |
| r\_name | varchar(20) | 否 |
| reason | varchar(50) | 否 |
| create\_time | datetime | 否 |
| update\_time | datetime | 否 |

d\_stgrade（卫生表）

|  |  |  |
| --- | --- | --- |
| g\_id | int(11) | 是 |
| s\_studentid | int(11) | 是 |
| s\_name | varchar(20) | 否 |
| s\_grade | int(11) | 否 |
| s\_classid | int(11) | 否 |
| s\_dormitoryid | int(11) | 否 |
| create\_time | datetime | 否 |
| update\_time | datetime | 否 |

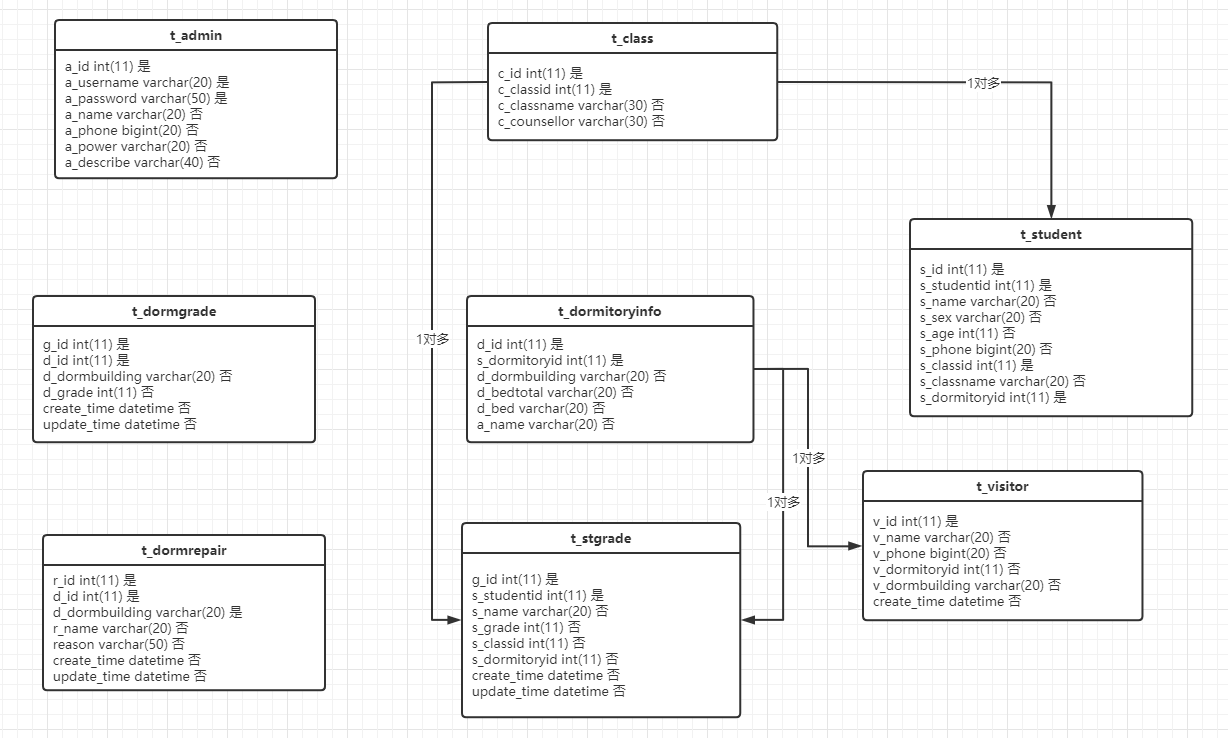
d\_student（学生表）

|  |  |  |
| --- | --- | --- |
| s\_id | int(11) | 是 |
| s\_studentid | int(11) | 是 |
| s\_name | varchar(20) | 否 |
| s\_sex | varchar(20) | 否 |
| s\_age | int(11) | 否 |
| s\_phone | bigint(20) | 否 |
| s\_classid | int(11) | 是 |
| s\_classname | varchar(20) | 否 |
| s\_dormitoryid | int(11) | 是 |

d\_visitor（访客表）

|  |  |  |
| --- | --- | --- |
| v\_id | int(11) | 是 |
| v\_name | varchar(20) | 否 |
| v\_phone | bigint(20) | 否 |
| v\_dormitoryid | int(11) | 否 |
| v\_dormbuilding | varchar(20) | 否 |
| create\_time | datetime | 否 |

### 数据表关系



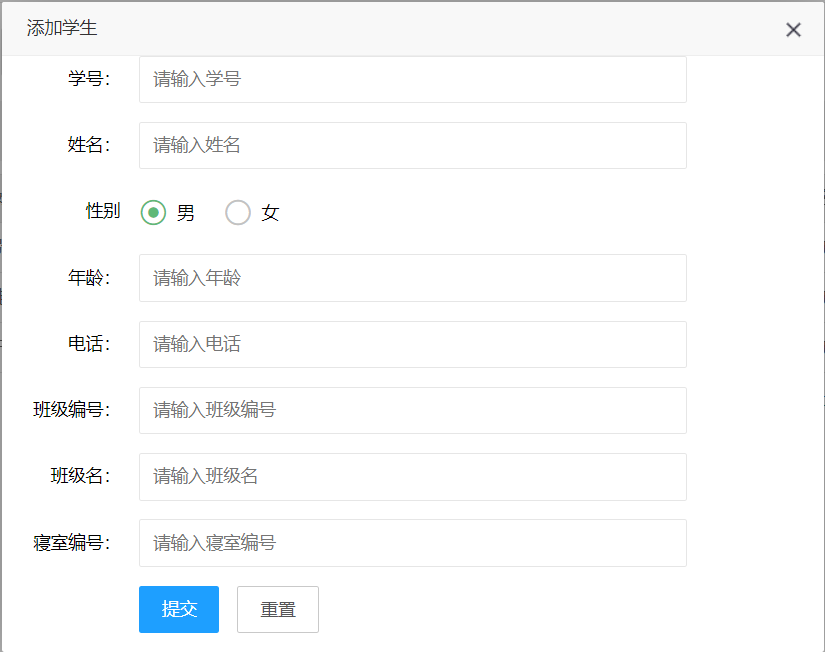
# 系统实现

### 学生管理

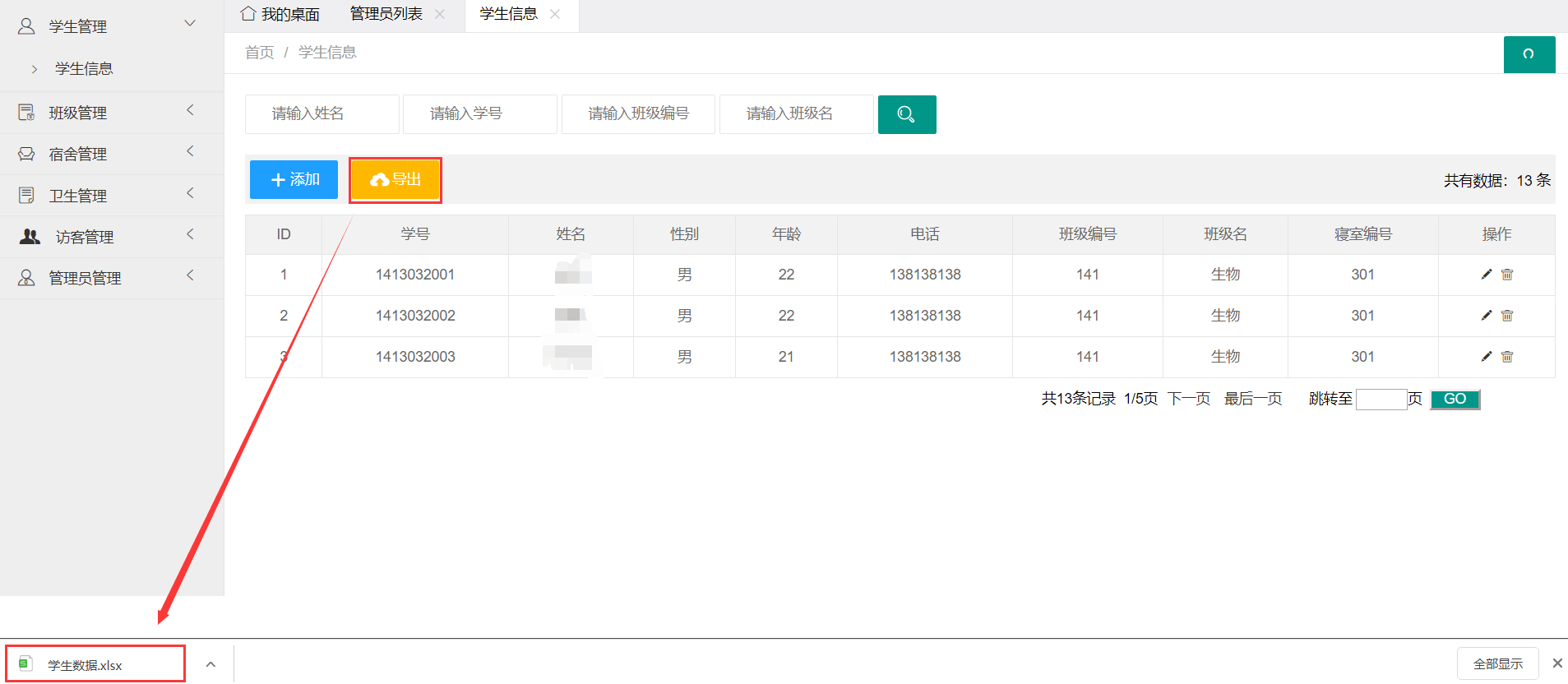
1 学生信息



2 添加学生



3 导出学生



4 学生检索



### 班级管理

1 班级列表



2 班级学生



### 宿舍管理

1 宿舍列表



2 人员信息



3 维修登记



### 卫生管理

1 宿舍卫生



2 学生卫生



### 访客管理

1 访客列表



### 管理员管理

1 管理员列表



# 总结与展望

。本平台后端采用的是Spring MVC搭配MyBatis框架，前端使用了JavaScript、Jsp、CSS等技术。在此之前，对于后台框架以及前端技术这些知识，我仅仅是了解一二、还不太熟悉原理。因此，这些知识对我来说还是十分陌生。但是如果不使用框架开发的话，虽然也可以开发出来，但是开发起来会很麻烦，项目也没有条理性，不适合对其进行管理，于是我一边学习相应知识一边开发该平台。在实践中检验理论，不断加深我的理解。前端页面大多找的模板，然后在自己进行修改，主要是样式方面的调整，花的时间不是很多，后端方面由于功能众多，逻辑结构相对复杂，所以我在此花的时间较多。

开发中碰到困难时，我需要到网上去查看相关资料、各种技术文档以及视频教程。好在经过我的不懈努力，总算完成了它，每完成一个功能点时，心中都会十分开心，这也是支持我继续下去的原因。通过整个平台从无到有的开发，我对MVC的开发模式有了深刻的了解，对Spring MVC和MyBatis框架也有着跟深刻的了解。

这个平台应用还有着不足之处。功能上还有些毛病，用起来不是特别方便，需要后期完善。界面也不是特别优美，还需要继续优化。还有一些逻辑问题都需要我改进，优化它们。

# 参考文献

[] 李兴华.Java开发实战经典[M].清华大学出版社.2009.

[2] 高宏.JavaScript从入门到精通[M].机械工业出版社.2012.

[3] David Flanagan .JavaScript：The.Definitive.Guide(6th Edition)[M].机械工业出版社.2011.

[4] 疯狂软件.Spring+MyBatis企业应用实战[M].电子工业出版社.2017.

[5] 杨开振.深入浅出MyBatis技术原理与实战[M].电子工业出版社.2016.