**长沙理工大学**

**软件工程《专业实训》报告**

**学生管理系统的设计与实现**

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **学 院** | **计通学院** | **专 业** | **软件工程** |
| **班 级** | **软件19-4** | **学 号** | **201908130410** |
| **学生姓名** | **刘淼** | **指导教师** | **谢中科** |
| **课程成绩** |  | **完成日期** | **2022年10月26日** |

软件工程《专业实训》成绩评定标准

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 支撑毕业要求指标点 | 考核与评价方式及成绩比例（%） | | | | 成绩比例（%） |
| 平时表现 | 实训日志 | 项目验收 | 实训报告 |
| 课程目标1 | 2.4 | 5 | \ | 15 | 10 | 30 |
| 课程目标2 | 3.3 | 5 | \ | 15 | 8 | 28 |
| 课程目标3 | 4.1 | 2 | 3 | 10 | 10 | 25 |
| 课程目标4 | 7.1 | \ | 5 | 5 | 7 | 17 |
| 合计 | | 12 | 8 | 45 | 35 | 100 |

**软件工程《专业实训》成绩评定表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 支撑毕业要求指标点 | 考核与评价方式及成绩比例（%） | | | | 成绩比例（%） |
| 平时表现 | 实训日志 | 项目验收 | 实训报告 |
| 课程目标1 | 2.4 |  | \ |  |  |  |
| 课程目标2 | 3.3 |  | \ |  |  |  |
| 课程目标3 | 4.1 |  |  |  |  |  |
| 课程目标4 | 7.1 | \ |  |  |  |  |
| 合计 | |  |  |  |  |  |

综合成绩 指导教师签字 年 月 日

**软件工程《专业实训》任务书**

计通学院 软件工程专业

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | 软件系统开发实训 | 时间 | 2021～2022年第2学期17～19周 | |
| 学生姓名 | 刘淼 | 指导老 师 | | 谢中科 |
| 题 目 | 家校通系统设计与实现 | | | |
| 主要内容：  本实训任务要求完成学生家-校联系系统的设计与实现，本系统完成三端通信：学校端、家庭端、管理端；每个角色需要完成的工作：  一、 家庭端需要完成的工作：  1 、与学生责任关系管理。  2 、学生家庭任务管理。  3、老师发布学校任务的查看。  4，对老师发布的公告的查看  二、 学校端需要完成的工作：  1 、与学生责任关系管理。  2 、学生学校任务管理。  3、对家长发布的家庭任务的查看  4，对公告的管理  三、管理端需要完成的工作：  1 、对所有学生信息的管理。  2 、对所有老师信息的管理。  3 、对所有家长信息的管理。 | | | | |

|  |
| --- |
| 要求：  (1) 能独立地运用程序设计、数据结构与算法、软件工程等方面专业知识，学 习和运用软件系统开发工具，编制一个规模适当的应用软件。  (2) 按要求编写实训报告书，能正确编写分析、设计、编码、测试等技术文 档。  (3) 通过实训培养学生严谨的科学态度，在软件系统开发的实践中了解和熟悉 有关软件知识产权、法律法规对软件产业的影响。  (4) 通过实训，掌握软件系统开发的各个阶段，以及不同阶段的项目管理要求 和方法。  (5) 在老师的指导下， 根据项目需求，学习软件系统的开发技术，并运用到项 目的开发过程中，提高项目适应能力和学习能力。 |
| 应当提交的文件：  (1)实训日志、实训报告。  (2)实训附件(源程序、各类图纸、实验数据、运行截图等)。 |

家校通系统设计与实现

学生姓名： 刘淼 指导老师： 谢中科

摘 要 该系统具备家长与老师交流的功能外增，增强了网络教学系统的交互性、实时性，使得家长和老师 能够更好地协同交流，拓展沟通交流渠道，架起家校沟通的桥梁，有利于家校教育形成合力，提 高了教育的时效性，促进了学生健康成长。通过该系统， 老师可以给学生家长进行发布公告，与一对一发布任务功能。学生家长可以查看教师发布的公告，任 务等。 管理员可以管理所有的人员信息。

关键词: 学生管理，图形化， Web ，数据库

目录

[1 引 言](#_bookmark1) [1](#_bookmark2)

[1.1 本文主要内容](#_bookmark3) [1](#_bookmark4)

[1.2 设计平台](#_bookmark5) [1](#_bookmark6)

[2 设计原理](#_bookmark7) [2](#_bookmark8)

[2.1 Java 编程](#_bookmark9) [2](#_bookmark10)

[2.2 JDBC 数据库连接](#_bookmark11) [2](#_bookmark12)

[2.3 MySQL 数据库](#_bookmark13) [2](#_bookmark14)

[3 家校通系统的需求分析](#_bookmark15) [4](#_bookmark16)

[3.1 可行性分析](#_bookmark17) [4](#_bookmark18)

[3.1.1 技术可行性](#_bookmark19) [4](#_bookmark20)

[3.1.2 经济可行性](#_bookmark21) [4](#_bookmark22)

[3.1.3 操作可行性](#_bookmark23) [4](#_bookmark24)

[3.2 需求分析](#_bookmark25) [4](#_bookmark26)

[3.2.1 需求分析概述](#_bookmark27) [4](#_bookmark28)

[3.2.2 业务逻辑图](#_bookmark29) [5](#_bookmark30)

[3.2.4 数据流图](#_bookmark31) [5](#_bookmark32)

[4 家校通系统的总体设计](#_bookmark33) [7](#_bookmark34)

[4.1 总体设计](#_bookmark35) [7](#_bookmark36)

[4.1.1 总体设计思想](#_bookmark37) [7](#_bookmark38)

[4.1.2 总体结构图](#_bookmark39) [7](#_bookmark40)

[4.1.3 模块设计](#_bookmark41) [7](#_bookmark42)

[5 家校通系统数据库设计](#_bookmark43) [8](#_bookmark44)

[5.1 实体分析](#_bookmark45) [8](#_bookmark46)

[5.2 数据表设计](#_bookmark47) [8](#_bookmark48)

[5.2.1 数据库定义](#_bookmark49) [8](#_bookmark50)

[5.2.2 数据库表](#_bookmark51) [9](#_bookmark52)

[6 家校通系统的详细设计](#_bookmark53) [11](#_bookmark54)

[6.1 结构设计](#_bookmark55) [11](#_bookmark56)

[6.2 登录部分设计](#_bookmark57) [11](#_bookmark58)

[6.2.1 登录界面设计](#_bookmark59) [11](#_bookmark60)

[6.2.2 逻辑设计](#_bookmark61) [11](#_bookmark62)

[6.3 家庭端功能部分设计](#_bookmark63) [12](#_bookmark64)

[6.3.1 家长功能界面设计](#_bookmark65) [12](#_bookmark66)

[6.3.2 逻辑功能设计](#_bookmark67) [12](#_bookmark68)

[6.4 管理端功能部分设计](#_bookmark69) [12](#_bookmark70)

[6.4.1 管理端功能界面设计](#_bookmark71) [12](#_bookmark72)

[6.4.2 管理端功能逻辑设计](#_bookmark73) [13](#_bookmark74)

[7 学生管理系统的实现](#_bookmark75) [13](#_bookmark76)

[7.1 系统具体功能的实现](#_bookmark77) [13](#_bookmark78)

[7.1.1 权限登录功能的实现](#_bookmark79) [13](#_bookmark80)

[7.1.2 任务发布功能实现](#_bookmark81) [14](#_bookmark82)

[7.1.3 添加功能实现](#_bookmark83) [15](#_bookmark84)

[7.1.4 删除功能实现](#_bookmark85) [16](#_bookmark86)

[7.1.5 更新功能的实现](#_bookmark87) [16](#_bookmark88)

[7.2 系统界面实现](#_bookmark89) [17](#_bookmark90)

[7.2.1 登陆界面实现](#_bookmark91) [17](#_bookmark92)

[7.2.2 家庭端界面实现](#_bookmark93) [19](#_bookmark94)

[7.2.3 管理端界面实现](#_bookmark95) [20](#_bookmark96)

[7.2.4 学校端界面实现](#_bookmark97) [22](#_bookmark98)

[8 家校通系统的测试用例](#_bookmark99) [25](#_bookmark100)

[9 结束语 26](#_bookmark101)

[参考文献](#_bookmark102) [27](#_bookmark103)

[附录](#_bookmark104) [28](#_bookmark105)

1 引 言

1.1 本文主要内容

本文设计实现了一个学生管理系统，前端选择了 web 端， 使用 vue 框架与 Element-UI 组件，后端采用了 ssm 框架， springBoot，Mybatis-Plus,Spring-MVC 进行开发。 主要包括公告模块，任务模块以及人员管理模块。

第二章介绍了该系统的需求分析，综合市场需要给出分析，第三章介绍了该系统的 总体设计，结合整个系统给出分析。第四章介绍了数据库的设计以及其中包含的数据 表。第五章介绍了数据库的设计。第六章给出了该系统的详细设计。第七章介绍了如 何实现该系统，第八章做了该系统的测试样例，第九章是结束语。

1.2 设计平台

(1) 开发环境：Windows10

(2) 开发工具： IntelliJ IDEA

(3) 编程语言：Java，JavaScript

2 设计原理

2.1 Java 编程

Java 是一门面向对象编程语言，不仅吸收了 C++语言的各种优点，还摒弃了 C++里难以理解的多继承、指针等概念，因此 Java 语言具有功能强大和简单易用两 个特征。Java 语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论， 允许程序员以优雅的思维方式进行复杂的编程 [1] 。

Java 具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、 多线程、动态性等特点 [2] 。Java 可以编写桌面应用程序、Web 应用程序、分布 式系统和嵌入式系统应用程序等

2.2 JDBC 数据库连接

JDBC 是由 Java 编程语言编写的类及接口组成，同时它为程序开发人员提供了 一组用于实现对数据库访问的 JDBC API，并支持 SQL 语言。利用 JDBC 可以将 Java 代码连接到 oracle、DB2、SQLServer、MYSQL 等数据库，从而实现对数据库中数据 操作的目的。

JDBC 全称为 Java Data Base Connectivity (Java 数据库连接)，可以为多种 数据库提供统一的访问。JDBC 是 sun 开发的一套数据库访问编程接口，是一种 SQL 级的 API。它是由 Java 语言编写完成，所以具有很好的跨平台特性，使用 JDBC 编 写的数据库应用程序可以在任何支持 Java 的平台上运行，而不必在不同的平台上编 写不同的程序。

2.3 MySQL 数据库

MySQL 是一个关系型数据库管理系统， 由瑞典 MySQL AB 公 司开发 ，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方 面， MySQL 是最好的 RDBMS (Relational Database Management System ，关系数 据库管理系统) 应用软件之一。

MySQL 是一种关系型数据库管理系统，关系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 所使用的 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，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。

3 家校通系统的需求分析

3.1 可行性分析

可行性分析是指在现有的技术、经济、法律和社会效益等方面的条件下分析每一 项需求实施的可行性，通过可行性分析，可以对系统的技术、操作和经济方面提出相 关要求和设计方案。可行性分析就是解决一个项目是都有可行解以及是否值得去解的

问题。

3.1.1 技术可行性

由于该系统是基于 web 实现开发的，所以满足大部分电脑支持， 采用的语言是 Java 语言，具有可移植性，功能易于实现，界面比较美观，操作简单，能够移植到 其他计算机上运行，与平台无关，能够满足更多的用户，使得该系统能被很多家庭所

接收。所以该系统在技术上是可行的。

3.1.2 经济可行性

现代化信息管理和电脑普及的今天，该系统如果开发成功，成本比较低，工作效 率较高，维护费用比较低，能给一个学校带来很多的方便，该系统采用比较容易学习 的 Java 语言编写，由于 Java 平台的自由和开放性使得开发周期比较短，开发人员成 本低。该系统成本主要是开发和数据库的维护上，所设计的成本都比较低，能给一个

学校带来很大的利润。因此在经济上是可行的。

3.1.3 操作可行性

该系统在操作上是可行的，系统是采用 Windows 平台开发的，是学校与系统之间 的操作，窗口和界面简单直观，易于操作和管理，业务逻辑清晰，各模块容易扩充。 因此在操作上是可行的。

3.2 需求分析

3.2.1 需求分析概述

为契合数字化经济发展趋势，助力教育行业实现从传统教育到“互联网+”教育 模式的转型， 班主任要改变过去传统的沟通交流方式本项目采用图文识别等手段，构 建家校互联平台， 该系统除了具备现有家长与老师交流的功能外,还支持图文识别功

能，从而丰富了教学方式的多样性，增加了网络教学系统的交互性、实时性，使得家

长和老师能够更好地协同交流，拓展沟通交流渠道，架起家校沟通的桥梁，有利于家 校教育形成合力，提 高了教育的时效性， 促进了学生健康成长， 开辟了家校教育发

展新空间。

3.2.2 业务逻辑图

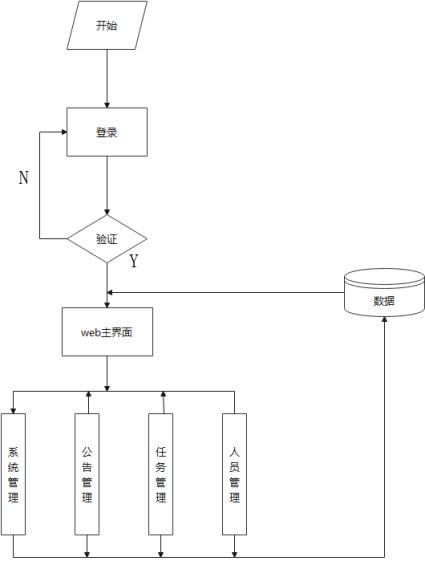


图 3.1 业务逻辑图

3.2.4 数据流图

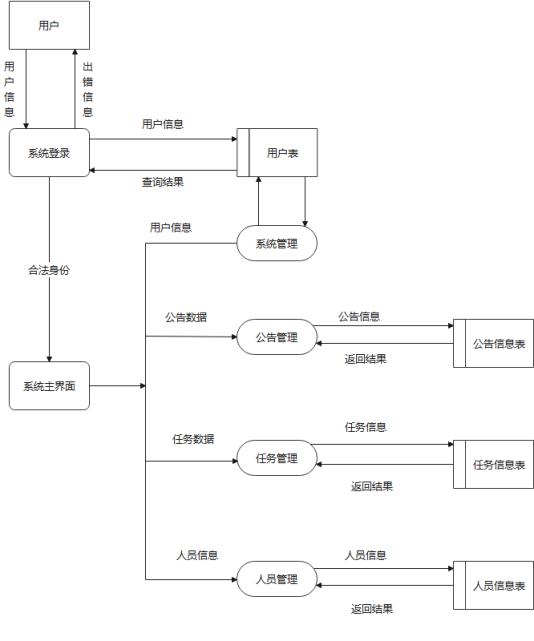


图 3.2 数据流图

4 家校通系统的总体设计

4.1 总体设计

4.1.1 总体设计思想

本系统设计采用 Spring-MVC，主要将代码放入 6 个包中， 分别是：

(1) entity 包： 存放数据库表的对象类

(2) mapper 包： 存放 mapper 层的接口

(3) service 包： 存放 service 层的类

(4) controller 包： 存放控制层代码

(5) config 包： 存放配置文件

4.1.2 总体结构图



图 4.1 数据结构图

4.1.3 模块设计

根据需求分为三个模块，公告模块，任务模块，人员管理模块。

公告模块包括教师与家长对公告的操作，老师可以给所有的学生家长发布公告。任 务模块包括教师可以对学校任务进行增删查， 对家庭任务进行查，家长可以对家庭任务 进行增删查， 对学校任务进行查。人员管理模块包括管理员可以对家长， 老师，学生进 行增删改查。

5 家校通系统数据库设计

5.1 实体分析

通常，采用实体-联系图(E-R 图) 来建立数据模型，包含实体、关系和属性三种成 分[4] ，他是描述现实世界概念结构模型的有效方法。是一种数据库设计的概念模型，是 描述数据实体与其他数据实体的关联的一种设计模式。实体内部的联系通常是指组成实 体的各种属性之间的联系，实体之间的联系通常是指不同实体集之间的联系。

ER 图如下：

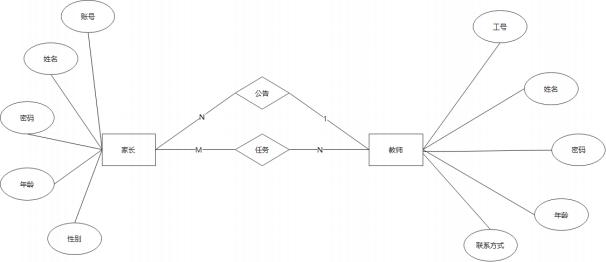


图 5.1 学生家-校联系系统 ER 图

5.2 数据表设计

5.2.1 数据库定义

数据库名为： hometoschool，登录用户名为 root，口令默认为 liumiao。该登录 用户的服务器角色是“database creators”，具有数据库 hometoschool 的“public”、

“db\_owner”等权限。

5.2.2 数据库表

表 5-1 管理员表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(5) not null | Id |
| username | varchar(255)not null | 用户名 |
| password | varchar(255)not null | 密码 |

表 5-2 家庭任务表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(5) not null | Id |
| studentid | varchar(255)not null | 学生 id |
| taskname | varchar(255)not null | 任务名称 |
| taskcontent | varchar(255)not null | 任务内容 |
| taskbehave | varchar(255)not null | 任务表现 |
| Time | Timestamp null | 发布时间 |
| img | varchar(255) null | 图片 |
| username | varchar(255)not null | 家长账号 |

表 5-3 公告表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(5) not null | id |
| notice\_title | varchar(255)not null | 公告名字 |
| notice\_content | varchar(255)not null | 公告内容 |
| teacherid | varchar(255)not null | 老师 id |
| time | Timestamp null | 发布时间 |

表 5-4 家长表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(5) not null | 家长 id |
| name | varchar(255)not null | 家长姓名 |
| sex | varchar(2)not null | 性别 |
| age | int(3) not null | 年龄 |
| phone\_number | varchar(11)not null | 手机号 |
| address | varchar(255)not null | 住址 |
| img | varchar(255)not null | 照片 |
| username | varchar(255)not null | 用户名 |
| password | varchar(255)not null | 密码 |

表 5-5 关系表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(3) not null | 关系 id |
| studentid | varchar(255)not null | 学生 id |
| parented | varchar(255)not null | 父母 id |
| relation | varchar(255)not null | 亲情关系 |

表 5-6 学校任务表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(5) not null | id |
| studentid | varchar(255)not null | 学生 id |
| taskname | varchar(255)not null | 任务名称 |
| taskcontent | varchar(255)not null | 任务内容 |

|  |  |  |
| --- | --- | --- |
| taskbehave | varchar(255)not null | 任务表现 |
| time | timestamp null | 发布时间 |
| img | varchar(255) null | 图片 |
| teacherid | varchar(255)not null | 老师工号 |

表 5-7 学生表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(5) not null | 学生 id |
| studentid | varchar(255)not null | 学生学号 |
| Name | varchar(255)not null | 学生姓名 |
| sex | varchar(2)not null | 性别 |
| age | int(3) not null | 年龄 |
| grade | varchar(10)not null | 年纪 |
| phone\_number | varchar(11)not null | 手机号 |
| address | varchar(255)not null | 住址 |
| img | varchar(255)not null | 照片 |
| teacherid | varchar(255)not null | 班主任工号 |

表 5-8 教师表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(5) not null | 老师 id |
| teacherid | varchar(255)not null | 老师工号 |
| Name | varchar(255)not null | 老师姓名 |
| sex | varchar(2)not null | 性别 |
| age | int(3) not null | 年龄 |
| phone\_number | varchar(11)not null | 手机号 |
| address | varchar(255)not null | 住址 |
| img | varchar(255)not null | 照片 |
| password | varchar(255)not null | 密码 |

6 家校通系统的详细设计

6.1 结构设计

通过对学生管理系统进行详细设计后，学生管理系统由公告管理 与通知、任务管理与通知、人员管理三个模块组成。其中公告模块包括发布公告，查看 公告，删除公告功能。任务管理与通知模块包括发布任务，查看任务，删除任务。人员 管理模块包括对所有人员的增删改查功能。

6.2 登录部分设计

6.2.1 登录界面设计

登录界面包括用户名，密码输入框， 身份选择框以及登录按钮。如图 6.1 所示

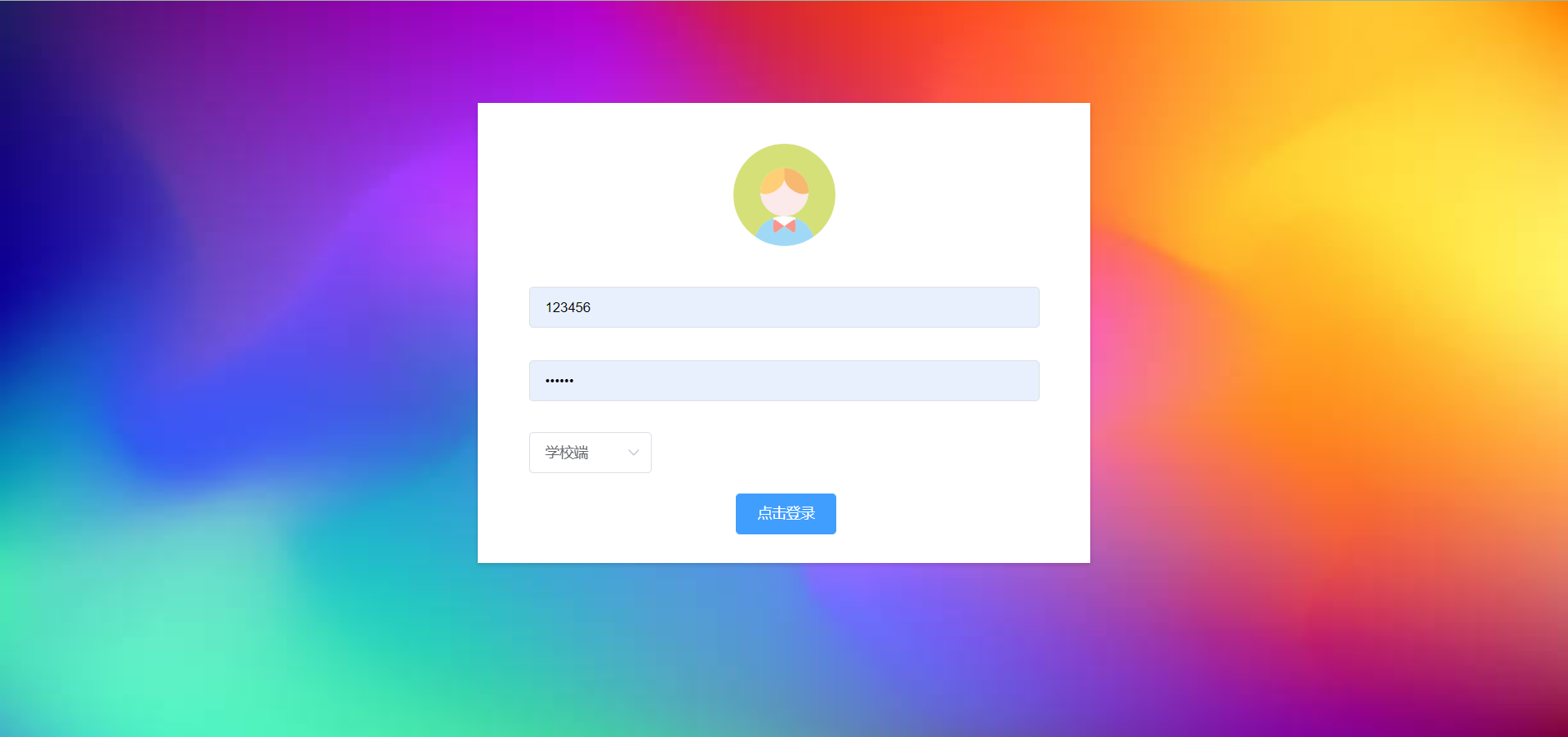


图 6.1 登录界面

6.2.2 逻辑设计

如果用户名与密码无法在身份对应的数据库里找到，则会提示账号，密码信息填写错误。如果 没输入用户名或者密码就点击登录则会提示输入信息不得为空。

6.3 家庭端功能部分设计

6.3.1 家庭端功能界面设计

该界面包括侧边栏与主界面，侧边栏包括学生信息，家庭任务，查看公告部分。其中学生信息 包括负责学生， 添加学生，家庭任务包括查看已发布，发布新任务功能。



图 6.2 .1家庭端模块界面图



图 6.2 .2家庭端模块发布任务



图 6.2 .3家庭端模块查看公告



图 6.2 .4家庭端模块管理已发布任务



图 6.2 .5家庭端模块管理添加学生

6.3.2 逻辑功能设计

点击负责学生按钮主界面可以跳转到学生信息模块，可以查看学生的详细信息。添加学生按钮 可以跳转到添加学生模块。点击查看已发布任务按钮可以查看已经发布过的任务，点击发布新任务 可以继续发布新任务。点击查看公告可以查看学生老师发布的公告。

6.4 管理端功能部分设计

6.4.1 管理端功能界面设计

该模块包括侧边栏与主界面，侧边栏包括学生信息，教师信息，家长信息。其中学生信息包括查看 信息与添加信息，教师信息包括查看信息与添加信息，家长模块包括查看信息与添加信息。



图 6.3.1 管理端模块界面

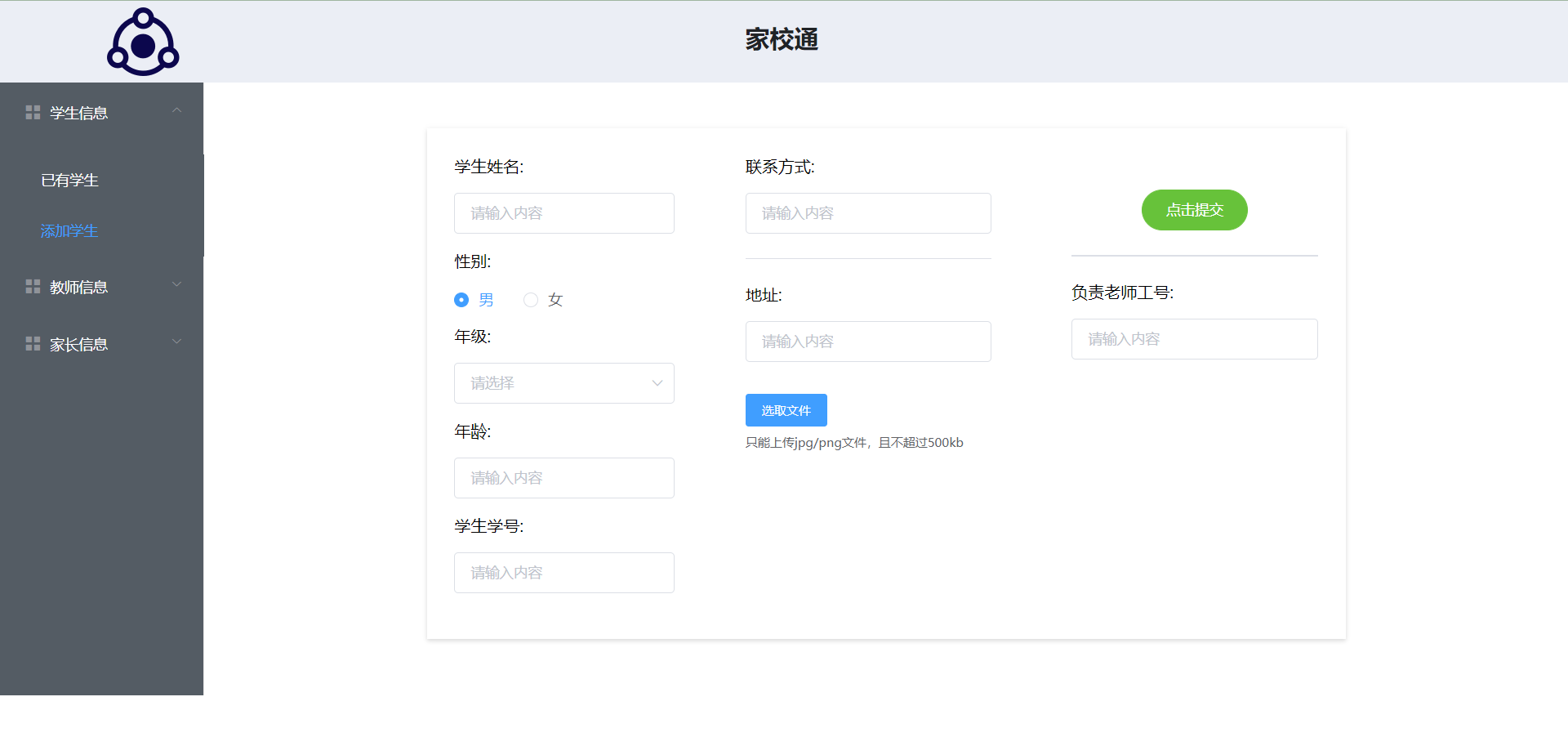


图 6.3.2 管理端模块添加学生



图 6.3.23管理端模块管理教师信息



图 6.3.24管理端模块管理家长信息

6.4.2 管理端功能逻辑设计

点击查看学生信息按钮可以查看学生的详细信息， 其中还包括删除，修改按钮，点击添加学生 信息按纽，可以添加新的学生信息。点击查看教师信息按钮可以查看教师的详细信息，其中还包括 删除，修改按钮，点击添加教师信息按纽，可以添加新的教师信息。点击查看家长信息按钮可以查 看家长的详细信息，其中还包括删除，修改按钮， 点击添加家长信息按纽， 可以添加新的家长信息。

7 家校通系统的实现

7.1 系统具体功能的实现

7.1.1 权限登录功能的实现

在 web 界面输入账号， 密码后将账号，密码传给后端，去数据库搜索，账号存在的话，

返回一个 token 给前端，前端储存。前端每次请求都在请求头里带上 token 后端：

package com.example.demo.config;

import com.example.demo.util.JWT;

import io.jsonwebtoken.\*;

import org.springframework.transaction.reactive.TransactionalOperatorExtensionsKt; import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.handler.HandlerInterceptorAdapter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class JWTInterceptor extends HandlerInterceptorAdapter {

@Override

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {

String myuri=request.getRequestURI();

if (RequestMethod.OPTIONS.toString().equals(request.getMethod())){

System.out.println("排除生成 token 的路径");

return true;

}

String token=request.getHeader("jwt");

if(token==null| |"".equals(token.trim())){

throw new ServletException("token 获取失败");

}

try{

Claims claims=JWT.parseJwt(token);

request.setAttribute("ID",String.valueOf(claims.getSubject())); return true;

}catch (ExpiredJwtException e){

throw new ServletException("身份过期，请重新登录");

}catch (UnsupportedJwtException | MalformedJwtException | SignatureException |IllegalArgumentException e){

throw new ServletException("jwt 解析失败");

}

}

}

7.1.2 任务发布功能实现

后端：

//家长添加任务

@Insert("INSERT INTO

homebehave(studentid,taskname,taskbehave,img,username)VALUES(#{studentid},#{taskname}, #{taskbehave},#{img},#{username})")

Boolean inserttask(String studentid,String taskname,String taskbehave,String img,String username);

//增添任务

public Integer inserttask(homebehave giao){

return homebehaveMapper.insert(giao);

}

@PostMapping(value = "/addtask")

@ApiOperation(value = "增添家庭任务")

public ResultModel addtask(@RequestParam(required = true)String studentid, @RequestParam(required = true)String taskname, @RequestParam(required = true)String taskcontent, @RequestParam(required = true)String taskbehave){

try {

String username=(String)request.getAttribute("ID");

String img=null;

homebehave behave=new

homebehave(studentid,taskname,taskcontent,taskbehave,img,username);

Integer flag=homebehaveService.inserttask(behave);

if(flag<=0){

return new ResultFailure("添加失败");

}else {

return new ResultSucess("添加成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

7.1.3 添加功能实现

//增添学生

@Insert("INSERT INTO students(studentid,NAME,sex,age,grade,phone\_number,address,img,teacherid)VALUES(#{stud entid},#{name},#{sex},#{age},#{grade},#{phonenumber},#{address},#{img},#{teacherid})") Boolean addstudent(String studentid,String name,String sex,Integer age,String grade,String phonenmber,String address,String img,String teacherid);

//添加学生

public Integer insertstudent(students giao){

return studentsMapper.insert(giao);

}

//对学生的操作------------------------------------------------------------------------

--

@PostMapping("/addstudent")

@ApiOperation("增添学生")

public ResultModel addstudent(@RequestParam(required = true)String studentid, @RequestParam(required = true)String name, @RequestParam(required = true)String sex, @RequestParam(required = true)Integer age, @RequestParam(required = true)String grade, @RequestParam(required = true)String phoneNumber, @RequestParam(required = true)String address, @RequestParam(required = true)String teacherid){

try {

String img=null;

students student=new

students(studentid,name,sex,age,grade,phoneNumber,address,img,teacherid);

Integer flag=studentService.insertstudent(student);

if(flag<=0){

return new ResultFailure("添加失败");

}else {

return new ResultSucess("添加成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

7.1.4 删除功能实现

//根据学号删除学生

@Delete("DELETE FROM students WHERE studentid=#{studentid}")

Boolean deletebystudentid(String studentid);

//根据学号删除学生

public Boolean deletebystudentid(String studentid){

return studentsMapper.deletebystudentid(studentid);

}

@GetMapping("/deletestu")

@ApiOperation("删除学生")

public ResultModel deletestu(String studentid){

try {

Boolean flag=studentService.deletebystudentid(studentid);

if(flag){

return new ResultSucess("删除成功");

}else {

return new ResultFailure("删除失败");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

7.1.5 更新功能的实现

@Update("UPDATE students SET img=#{img} WHERE id=@@identity")

Boolean addimg(String img);

//修改学生信息

public Integer updatestu(students giao){

return studentsMapper.updateById(giao);

}

@GetMapping("/updatestu")

@ApiOperation("修改学生信息")

public ResultModel update(@RequestParam(required = true)Integer id, @RequestParam(required = true)String studentid, @RequestParam(required = true)String name, @RequestParam(required = true)String sex, @RequestParam(required = true)Integer age, @RequestParam(required = true)String grade, @RequestParam(required = true)String phoneNumber, @RequestParam(required = true)String address, @RequestParam(required = true)String teacherid, @RequestParam(required = false)MultipartFile file){

try {

String img=null;

if(file!=null){

Filedo fi=new Filedo(file);

img=fi.load();

}

students student=new

students(studentid,name,sex,age,grade,phoneNumber,address,img,teacherid);

student.setId(id);

Integer flag=studentService.updatestu(student);

if(flag<=0){

return new ResultFailure("修改失败");

}else {

return new ResultSucess("修改成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

7.2 系统界面实现

7.2.1 登陆界面实现

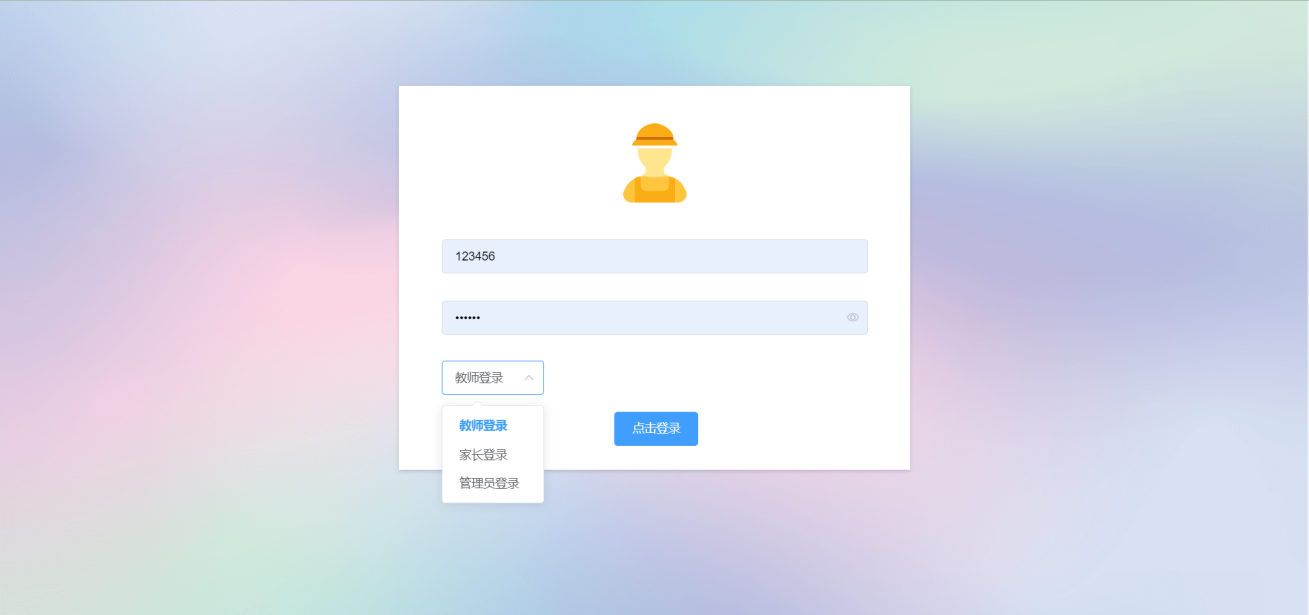


图 7.1 登录界面

<template>

<div>

<div id="box">

<div id="img">

<img src="../../../../public/picture/ren1.png" alt="" style="height:

100px;width: 100px;margin-top: 40px">

</div>

<form>

<div class="input" style="margin-top: 80px;">

<el-input v-model="username" placeholder="请输入账号"></el-input> </div>

<div class="input" style="margin-top: 12px">

<el-input placeholder=" 请输入密码 " v-model="password" show-password></el-

input>

</div>

<div id="select">

<el-select v-model="value" placeholder="请选择">

<el-option

v-for="item in options"

:key="item.value"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</div>

<div style="height: 20px;width: 94px;margin-left: auto;margin-right:

auto;margin-top: 10px">

<el-button type="primary" @click="login">点击登录</el-button>

</div>

</form>

</div>

</div>

</template>

7.2.2 家长界面实现

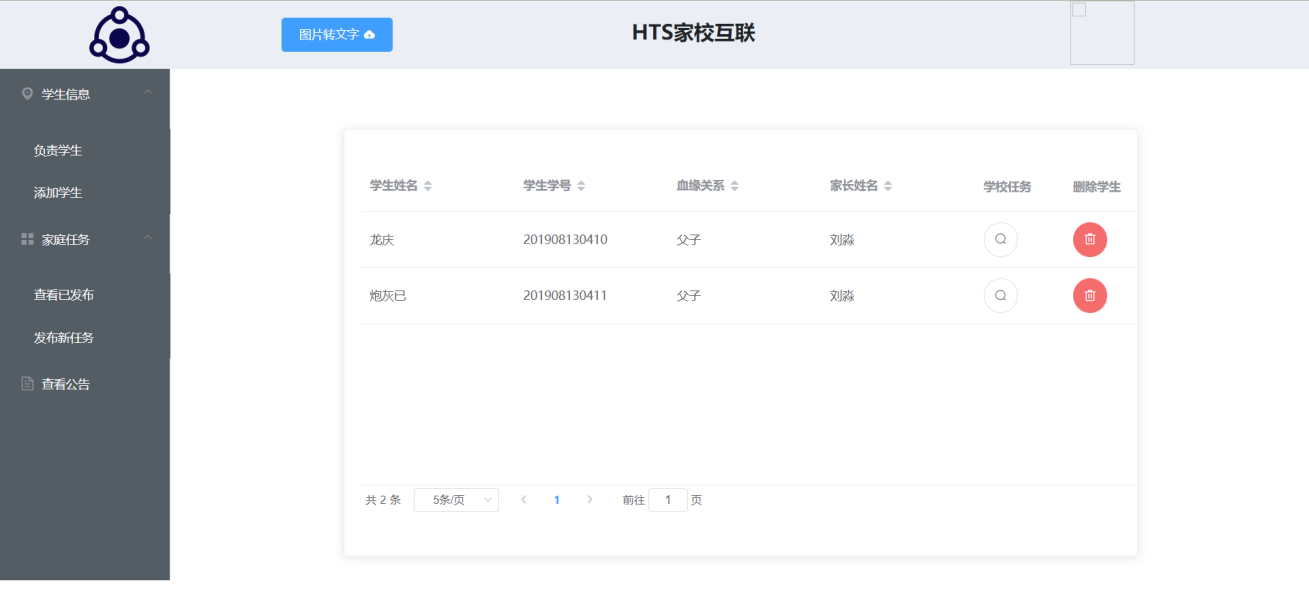


图 7.2 家庭端模块主界面

<template>

<div>

<div id="top">

<top></top>

</div>

<div style="height: 600px">

<div style="height: 100%;width: 100%">

<el-container style="height: 100%;width: 100%">

<el-aside width="200px" style="height: 100%">

<el-menu

default-active="this.$route.path" router

class="el-menu-vertical-demo"

background-color="#545c64"

text-color="#fff"

style="height: 100%">

<el-submenu index="1">

<template slot="title">

<i class="el-icon-location"></i>

<span>学生信息</span>

</template>

<el-menu-item-group>

<el-menu-item index="/relationship">负责学生</el-menu-item> <el-menu-item index="/addstudent">添加学生</el-menu-item>

</el-menu-item-group>

</el-submenu>

<el-submenu index="2">

<template slot="title">

<i class="el-icon-menu"></i>

<span>家庭任务</span>

</template>

<el-menu-item-group>

<el-menu-item index="/mytask">查看已发布</el-menu-item> <el-menu-item index="/addhometask">发布新任务</el-menu-item>

</el-menu-item-group>

</el-submenu>

<el-menu-item index="/parnotice" >

<i class="el-icon-document"></i>

<span slot="title">查看公告</span>

</el-menu-item>

</el-menu>

</el-aside>

<el-main>

<router-view></router-view>

</el-main>

</el-container>

</div>

</div>

</div>

</template>

7.2.3 管理员界面实现

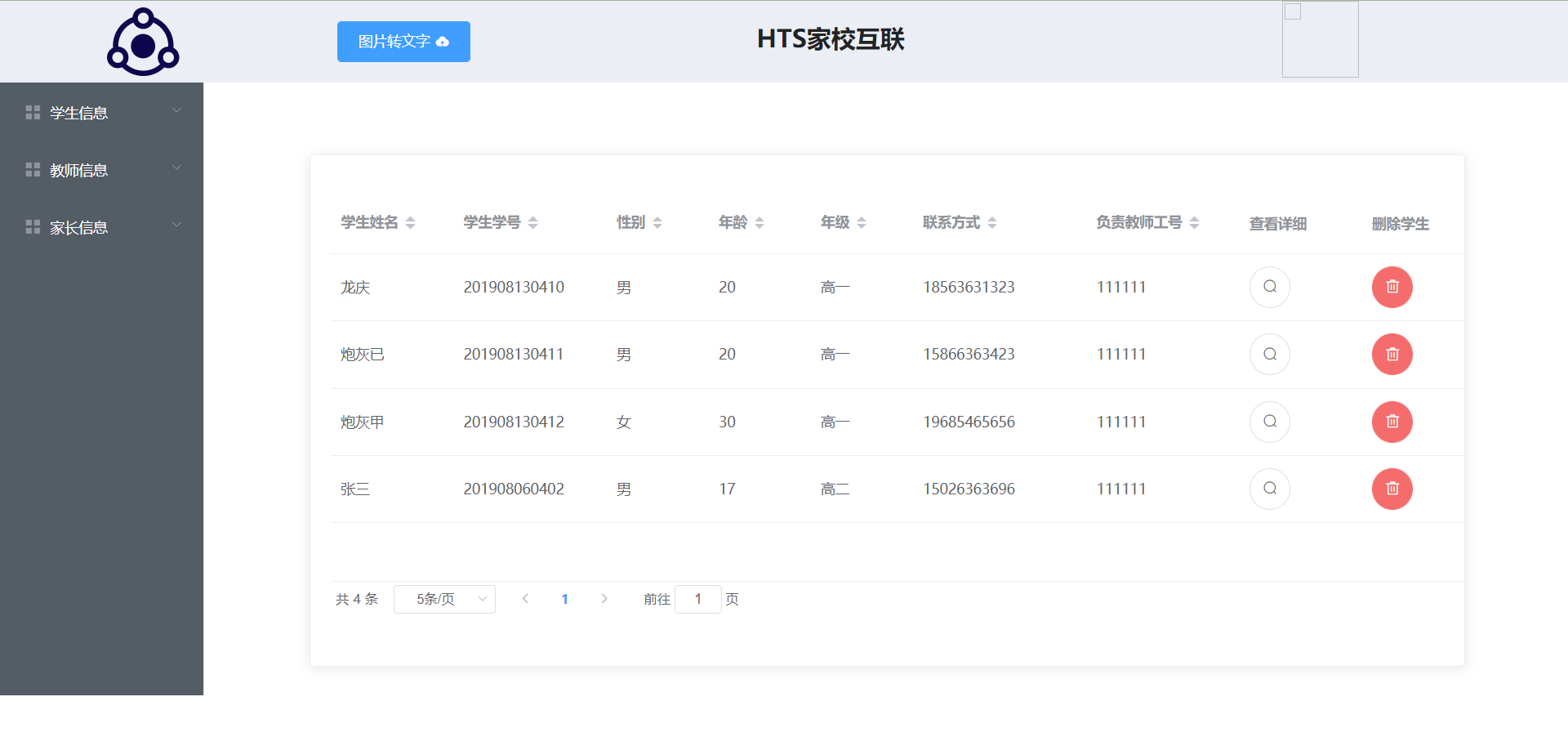


图 7.3 管理端模块主界面

<template>

<div>

<div id="top">

<top></top>

</div>

<div style="height: 600px">

<div style="height: 100%;width: 100%">

<el-container style="height: 100%;width: 100%">

<el-aside width="200px" style="height: 100%">

<el-menu

default-active="this.$route.path" router

class="el-menu-vertical-demo"

background-color="#545c64"

text-color="#fff"

style="height: 100%">

<el-submenu index="1">

<template slot="title">

<i class="el-icon-menu"></i>

<span>学生信息</span>

</template>

<el-menu-item-group>

<el-menu-item index="/astudent">已有学生</el-menu-item> <el-menu-item index="/addstudenta">添加学生</el-menu-item>

</el-menu-item-group>

</el-submenu>

<el-submenu index="2">

<template slot="title">

<i class="el-icon-menu"></i>

<span>教师信息</span>

</template>

<el-menu-item-group>

<el-menu-item index="/ateacher">已有教师</el-menu-item> <el-menu-item index="/addteachera">添加教师</el-menu-item>

</el-menu-item-group>

</el-submenu>

<el-submenu index="3">

<template slot="title">

<i class="el-icon-menu"></i>

<span>家长信息</span>

</template>

<el-menu-item-group>

<el-menu-item index="/aparent">已有家长</el-menu-item>

<el-menu-item index="/addparenta">添加家长</el-menu-item> </el-menu-item-group>

</el-submenu>

</el-menu>

</el-aside>

<el-main>

<router-view></router-view>

</el-main>

</el-container>

</div>

</div>

</div>

</template>

<script>

import top from "@/views/common/top";

export default {

name: "admin",

components:{top}

}

</script>

<style scoped>

</style>

7.2.4 教师界面实现

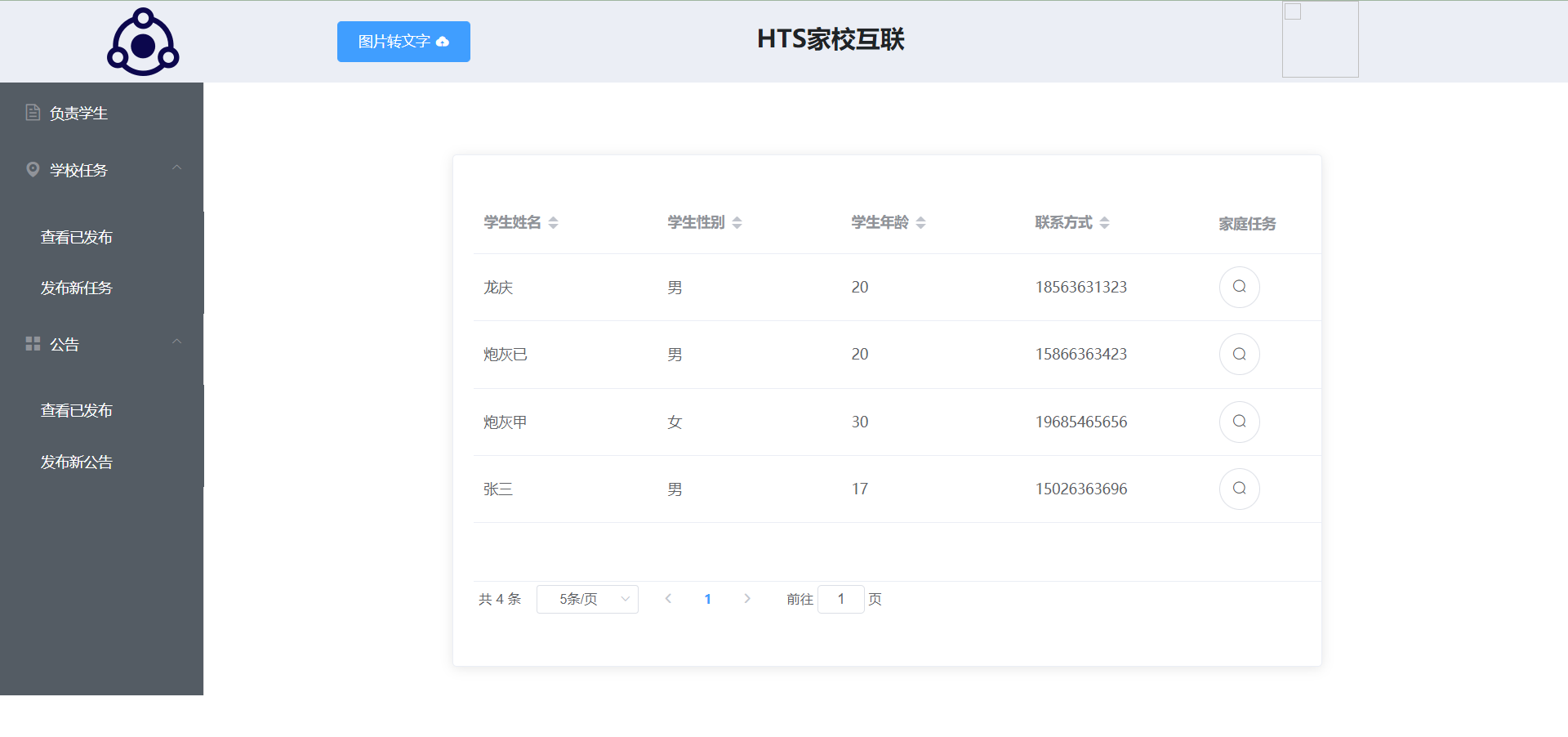


图 7.4 教师端模块主界面

<template>

<div>

<div id="top">

<top></top>

</div>

<div style="height: 600px">

<div style="height: 100%;width: 100%">

<el-container style="height: 100%;width: 100%">

<el-aside width="200px" style="height: 100%">

<el-menu

default-active="this.$route.path" router

class="el-menu-vertical-demo"

background-color="#545c64"

text-color="#fff"

style="height: 100%">

<el-menu-item index="/teastudent" >

<i class="el-icon-document"></i>

<span slot="title">负责学生</span>

</el-menu-item>

<el-submenu index="1">

<template slot="title">

<i class="el-icon-location"></i>

<span>学校任务</span>

</template>

<el-menu-item-group>

<el-menu-item index="/schtask">查看已发布</el-menu-item>

<el-menu-item index="/addschooltask">发布新任务</el-menu-item> </el-menu-item-group>

</el-submenu>

<el-submenu index="2">

<template slot="title">

<i class="el-icon-menu"></i>

<span>公告</span>

</template>

<el-menu-item-group>

<el-menu-item index="/teanotice">查看已发布</el-menu-item> <el-menu-item index="/addnotice">发布新公告</el-menu-item>

</el-menu-item-group>

</el-submenu>

</el-menu>

</el-aside>

<el-main>

<router-view></router-view>

</el-main>

</el-container>

</div>

</div>

</div>

</template>

<script>

import top from '@/views/common/top'

export default {

name: "teachers",

components:{top}

}

</script>

<style scoped>

</style>

8 学生家-校联系系统的测试用例

表 8-1 登录页面测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试编号 | 用例描述 | 预期结果 | 结论 |
| 1 | 输入错误的账号 | 提示账号不存在 | 与预期一致 |
| 2 | 不输入，直接点击登录按钮 | 提示为空 | 与预期一致 |
| 3 | 输入合理的账号、密码 | 登录成功 | 与预期一致 |

表 8-2 主页面测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试编号 | 用例描述 | 预期结果 | 结论 |
| 1 | 管理员账号密码登陆 | 主界面功能全部可用 | 与预期一致 |
| 2 | 普通用户账号密码登录 | 主界面功能全部可用 | 与预期一致 |
| 3 | 点击任意功能按钮 | 跳转下一界面 | 与预期一致 |

表 8-3 用户管理页面测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试编号 | 用例描述 | 预期结果 | 结论 |
| 1 | 未输入正确信息添加用户 | 提示信息不能为空 | 与预期一致 |
| 2 | 未输入正确信息删除用户 | 提示输入正确用户编  号删除用户 | 与预期一致 |

表 8-4 信息修改页面测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试编号 | 用例描述 | 预期结果 | 结论 |
| 1 | 未输入正确信息进行修改 | 提示信息为空 | 与预期一致 |
| 2 | 未输入正确用户编号进行修改 | 修改功能不可用 | 与预期一致 |

9 结束语

通过这次课程设计，应用自己所学习的知识并且查找资料书籍，大致完成了一个 家庭财务管理系统，在系统完成期间，遇到了各种各样的困难，一些通过查找书籍资 料得到了解决，也有一些通过百度搜索、视频学习得到了解决。本次课程设计让我将 平时学习的理论知识更好的应用到了实践中，在此次实训中我更加了解到只有自己多 去做、多去写，才能更快的提高自己的专业能力和技术水平。

通过本次系统的实现， 我进一步加深了自己对 web 全栈开发的理解，也为我日后 的学习和工作打好了基础。除此之外，在完成系统设计期间，我同时进一步提高了自 己的自学能力，作为一个未来的 IT 工作者，此次实训也为日后学习和就业都提供了非 常大的帮助。

参考文献

[1] .陈强. 精通 Java 开发技术[M].北京：清华大学出版社， 2014

[2] .Adam Freeman HTML5 权威指南.北京：人民邮电出版社， 2014

[3] .姚永一. SQL Server 数据库实用教程[M].北京：电子工业出版社， 2010 [4] .萨师煊，王珊．数据库系统概论[M]．北京： 高等教育出版社， 2005

[5] .里斯.JDBC TM 与 Java TM 数据库编程(第 2 版) [M].北京：中国电力出版社.2002

附录

部分核心代码展示

package com.example.demo.controller;

import com.example.demo.entity.parents;

import com.example.demo.entity.students;

import com.example.demo.entity.teachers;

import com.example.demo.result.ResultFailure;

import com.example.demo.result.ResultModel;

import com.example.demo.result.ResultNull;

import com.example.demo.result.ResultSucess;

import com.example.demo.service.AdminService;

import com.example.demo.service.ParentsService;

import com.example.demo.service.StudentService;

import com.example.demo.service.TeachersService;

import com.example.demo.util.Filedo;

import com.example.demo.util.JWT;

import com.sun.org.apache.xpath.internal.operations.Bool;

import io.swagger.annotations.Api;

import io.swagger.annotations.ApiOperation;

import lombok.extern.slf4j.Slf4j;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.validation.annotation.Validated;

import org.springframework.web.HttpRequestHandler;

import org.springframework.web.bind.annotation.\*;

import org.springframework.web.multipart.MultipartFile;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.validation.Valid;

import java.util.ArrayList;

import java.util.HashMap;

@RestController

@Slf4j

@Validated

@RequestMapping(value = "/admin")

@Api(tags = "管理员")

public class AdminController {

@Resource

private AdminService adminService;

@Resource

private StudentService studentService;

@Resource

private TeachersService teachersService;

@Resource

private ParentsService parentsService;

@Autowired

HttpServletRequest request;

@GetMapping("/login")

@ApiOperation("管理员登录接口")

public ResultModel login(@RequestParam(required = true) String username,

@RequestParam(required = true) String password){

try {

Integer flag=adminService.login(username,password);

if(flag<=0){

return new ResultFailure("登录失败， 查无此用户");

}else {

HashMap<String,Object> map=new HashMap<>();

map.put("jwt", JWT.createJwt(username,username,10800000));

return new ResultSucess("登录成功",map);

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

//对学生的操作--------------------------------------------------------------------------

@PostMapping("/addstudent")

@ApiOperation("增添学生")

public ResultModel addstudent(@RequestParam(required = true)String studentid, @RequestParam(required = true)String name, @RequestParam(required = true)String sex, @RequestParam(required = true)Integer age, @RequestParam(required = true)String grade, @RequestParam(required = true)String phoneNumber, @RequestParam(required = true)String address, @RequestParam(required = true)String teacherid){

try {

String img=null;

students student=new

students(studentid,name,sex,age,grade,phoneNumber,address,img,teacherid);

Integer flag=studentService.insertstudent(student);

if(flag<=0){

return new ResultFailure("添加失败");

}else {

return new ResultSucess("添加成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@PostMapping("/addstuimg")

@ApiOperation(value = "添加图片")

public ResultModel addimg(MultipartFile file){

try {

Filedo fi=new Filedo(file);

String path=fi.load();

Boolean flag=studentService.addimg(path);

if(flag){

return new ResultSucess("添加成功");

}else {

return new ResultFailure("添加失败");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@GetMapping("/deletestu")

@ApiOperation("删除学生")

public ResultModel deletestu(String studentid){

try {

Boolean flag=studentService.deletebystudentid(studentid);

if(flag){

return new ResultSucess("删除成功");

}else {

return new ResultFailure("删除失败");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@GetMapping("/findstu")

@ApiOperation("查询所有学生")

public ResultModel findstu(){

try {

ArrayList<students> stus=studentService.findall();

if(stus.size()<=0){

return new ResultNull("暂无学生");

}else {

HashMap<String,Object> map=new HashMap<>();

map.put("student",stus);

return new ResultSucess("查询成功",map);

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@GetMapping("/updatestu")

@ApiOperation("修改学生信息")

public ResultModel update(@RequestParam(required = true)Integer id,

@RequestParam(required = true)String studentid,

@RequestParam(required = true)String name,

@RequestParam(required = true)String sex,

@RequestParam(required = true)Integer age,

@RequestParam(required = true)String grade,

@RequestParam(required = true)String phoneNumber,

@RequestParam(required = true)String address,

@RequestParam(required = true)String teacherid,

@RequestParam(required = false)MultipartFile file){

try {

String img=null;

if(file!=null){

Filedo fi=new Filedo(file);

img=fi.load();

}

students student=new

students(studentid,name,sex,age,grade,phoneNumber,address,img,teacherid);

student.setId(id);

Integer flag=studentService.updatestu(student);

if(flag<=0){

return new ResultFailure("修改失败");

}else {

return new ResultSucess("修改成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

//-----------------------------------------------------------对老师操作

@PostMapping("/addtea")

@ApiOperation("增添新教师")

public ResultModel addtea(@RequestParam(required = true)String teacherid,

@RequestParam(required = true)String name,

@RequestParam(required = true)String sex,

@RequestParam(required = true)Integer age,

@RequestParam(required = true)String phoneNumber,

@RequestParam(required = true)String address,

@RequestParam(required = true)String password){

try {

String img=null;

teachers teacher=new

teachers(teacherid,name,sex,age,phoneNumber,address,img,password);

Integer flag=teachersService.insertteacher(teacher);

if(flag<=0){

return new ResultFailure("添加失败");

}else {

return new ResultSucess("添加成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@PostMapping("/addteaimg")

@ApiOperation(value = "添加图片")

public ResultModel addteaimg( MultipartFile file){

try {

Filedo fi=new Filedo(file);

String path=fi.load();

Boolean flag=teachersService.addimg(path);

if(flag){

return new ResultSucess("添加成功");

}else {

return new ResultFailure("添加失败");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@GetMapping("/deletetea")

@ApiOperation("删除教师")

public ResultModel deletetea(@RequestParam(required = true)String teacherid){ try {

Boolean flag=teachersService.deletebyteaid(teacherid);

if(flag){

return new ResultSucess("删除成功");

}else {

return new ResultFailure("删除失败");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@GetMapping("findtea")

@ApiOperation("查询所有教师")

public ResultModel findtea(){

try {

ArrayList<teachers> stus=teachersService.findall();

if(stus.size()<=0){

return new ResultNull("暂无老师");

}else {

HashMap<String,Object> map=new HashMap<>();

map.put("teacher",stus);

return new ResultSucess("查询成功",map);

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@PostMapping("/updatetea")

@ApiOperation("修改教师信息")

public ResultModel updatetea(@RequestParam(required = true)Integer id, @RequestParam(required = true)String teacherid,

@RequestParam(required = true)String name, @RequestParam(required = true)String sex, @RequestParam(required = true)Integer age, @RequestParam(required = true)String phoneNumber, @RequestParam(required = true)String address, @RequestParam(required = true)String password, @RequestParam(required = false)MultipartFile file){

try {

String img=null;

if(file!=null){

Filedo fi=new Filedo(file);

img=fi.load();

}

teachers

teachers(teacherid,name,sex,age,phoneNumber,address,img,password); teacher.setId(id);

Integer flag=teachersService.updateteacher(teacher);

if(flag<=0){

return new ResultFailure("修改失败");

}else {

return new ResultSucess("修改成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

//----------------------------------------------------------------对父母的操作

@PostMapping("/addpar")

@ApiOperation("增添父母")

public ResultModel addpar(@RequestParam(required = true)String name, @RequestParam(required = true)String sex, @RequestParam(required = true)Integer age, @RequestParam(required = true)String phoneNumber, @RequestParam(required = true)String address, @RequestParam(required = true)String username, @RequestParam(required = true)String password){

try {

String img=null;

parents

parents(name,sex,age,phoneNumber,address,img,username,password); Integer flag=parentsService.insterpar(parent);

if(flag<=0){teacher=new

parent=new

return new ResultFailure("添加失败");

}else {

return new ResultSucess("添加成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@PostMapping("/addparimg")

@ApiOperation(value = "添加图片")

public ResultModel addparimg(MultipartFile file){

try {

Filedo fi=new Filedo(file);

String path=fi.load();

Boolean flag=parentsService.addimg(path);

if(flag){

return new ResultSucess("发布成功");

}else {

return new ResultFailure("发布失败");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@GetMapping("/deletepar")

@ApiOperation("删除父母")

public ResultModel deletepar(@RequestParam(required = true) String username){ try {

Boolean flag=parentsService.deletepar(username);

if(flag){

return new ResultSucess("删除成功");

}else {

return new ResultFailure("删除失败");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@GetMapping("/findpar")

@ApiOperation("查询所有父母")

public ResultModel finpar(){

try {

ArrayList<parents> stus=parentsService.findall();

if(stus.size()<=0){

return new ResultNull("暂无家长");

}else {

HashMap<String,Object> map=new HashMap<>();

map.put("parent",stus);

return new ResultSucess("查询成功",map);

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

@PostMapping("/updatepar")

@ApiOperation("修改父母信息")

public ResultModel updatepar(@RequestParam(required = true)Integer id,

@RequestParam(required = true)String name,

@RequestParam(required = true)String sex,

@RequestParam(required = true)Integer age,

@RequestParam(required = true)String phoneNumber,

@RequestParam(required = true)String address,

@RequestParam(required = true)String username,

@RequestParam(required = true)String password,

@RequestParam(required = false)MultipartFile file){

try {

String img=null;

if(file!=null){

Filedo fi=new Filedo(file);

img=fi.load();

}

parents parent=new

parents(name,sex,age,phoneNumber,address,img,username,password);

parent.setId(id);

Integer flag=parentsService.updatepar(parent);

if(flag<=0){

return new ResultFailure("修改失败");

}else {

return new ResultSucess("修改成功");

}

} catch (Exception e) {

e.printStackTrace();

return new ResultFailure("操作失败，系统故障");

}

}

}