



重慶工程學院  
Chongqing Institute of Engineering

# 毕业设计（论文）

题 目 基于 Web 的家校助学联动平台

的设计与实现

学 院 软件学院

专 业 软件工程

学生姓名 陈垚樨 学 号 169001427

指导教师 孟高原 职 称 讲师

2020 年 5 月 25 日



## 学生毕业设计（论文）原创性声明

本人以信誉声明：所呈交的毕业设计（论文）是在导师的指导下进行的设计（研究）工作及取得的成果，设计（论文）中引用他（她）人的文献、数据、图件、资料均已明确标注出，论文中的结论和结果为本人独立完成，不包含他人成果及为获得重庆工程学院或其它教育机构的学位或证书而使用其材料。与我一同工作的同志对本设计（研究）所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

毕业设计（论文）作者（签字）：

年      月      日



## 摘要

在通信技术日益发展的今天，因为太多社交平台和通讯工具占据了人们的生活，导致平台选择的不正确性，让交流出现一定障碍，尤其是在学校与家长之间的沟通存在很大问题。在此问题上，经过对家校之间沟通问题等相关需求的调查与研究，便产生了基于 Web 的家校助学联动平台的设计和开发。

该平台系统，是在目前国内最流行的社交平台——微信的基础上进行开发的，通过微信交流沟通，以及利用微信小程序对家校之间沟通信息进行优化处理，将有效的信息输出给双方，提升沟通效率和质量。

本文将在基于 Web 的家校助学联动平台的基础上，针对学校教师与家长的需求进行调研和分析，然后进行功能模块分析以及系统设计等。具体内容如下：

- 1) 对家校助学联动平台的背景与国内外现状进行研究。
- 2) 结合调研的结果对家校助学联动平台进行进一步的需求分析。
- 3) 根据需求分析对家校助学联动平台进行详细的系统分析与设计，落实到每个功能模块。
- 4) 针对设计好的系统设计对系统进行开发与实现，以及后期对系统的相关测试。

**关键词：**家校助学平台 沟通 媒介 微信小程序



## ABSTRACT

Nowadays, with the development of communication technology, too many social platforms and communication tools occupy people's lives, which leads to the incorrect choice of platforms and some obstacles in communication, especially in the communication between schools and parents. On this issue, through the investigation and Research on the communication between home and school, the design and development of home school student interaction platform based on web is produced.

The platform system is developed on the basis of wechat, the most popular social platform in China. Through wechat communication, as well as using wechat small programs to optimize the communication information between home and school, it can output effective information to both sides and improve the communication efficiency and quality.

Based on the web-based home school student interaction platform, this paper will investigate and analyze the needs of school teachers and parents, and then carry out functional module analysis and system design. The details are as follows:

- 1) This paper studies the background and current situation at home and abroad.
- 2) Combined with the results of the research, this paper further analyzes the needs of the home school student linkage platform.
- 3) According to the demand analysis, the paper makes a detailed system analysis and design of the home school student interaction platform, which is implemented to each functional module.
- 4) Reference system design to develop and implement the system, as well as the related tests of the system in the later stage.

**Keywords:** home school student interaction platform; mini-program; Communication; medium





## 目 录

|                            |     |
|----------------------------|-----|
| 摘 要 .....                  | I   |
| ABSTRACT .....             | III |
| 1 绪 论 .....                | 1   |
| 1.1 研究背景 .....             | 1   |
| 1.2 国内外研究现状 .....          | 1   |
| 1.3 论文的组织结构 .....          | 2   |
| 1.4 本章小结 .....             | 3   |
| 2 关键技术分析 .....             | 5   |
| 2.1 技术框架 .....             | 5   |
| 2.1.1 Vue.js .....         | 5   |
| 2.1.2 TypeScript .....     | 5   |
| 2.1.3 koa .....            | 6   |
| 2.1.4 uni-app(微信小程序) ..... | 6   |
| 2.2 开发工具 .....             | 6   |
| 2.3 本章小结 .....             | 7   |
| 3 需求分析 .....               | 9   |
| 3.1 需求描述 .....             | 9   |
| 3.2 系统角色分析 .....           | 10  |
| 3.3 系统流程分析 .....           | 10  |
| 3.4 功能需求分析 .....           | 11  |
| 3.4.1 后台管理系统功能分析 .....     | 11  |
| 3.4.2 微信小程序家长功能分析 .....    | 12  |
| 3.4.3 微信小程序教师功能分析 .....    | 13  |
| 3.5 用例图 .....              | 14  |
| 3.6 用例描述 .....             | 16  |

|                     |    |
|---------------------|----|
| 3.7 本章小结 .....      | 19 |
| 4 系统分析与设计 .....     | 21 |
| 4.1 技术架构设计 .....    | 21 |
| 4.2 系统架构设计 .....    | 22 |
| 4.3 功能设计 .....      | 23 |
| 4.4 功能模块设计 .....    | 24 |
| 4.4.1 成绩导入 .....    | 24 |
| 4.4.2 发布并查看公告 ..... | 26 |
| 4.4.3 家长绑定 .....    | 27 |
| 4.5 数据库设计 .....     | 28 |
| 4.5.1 概念结构设计 .....  | 29 |
| 4.5.2 逻辑结构设计 .....  | 29 |
| 4.5.3 物理结构设计 .....  | 30 |
| 4.6 界面设计 .....      | 36 |
| 4.7 本章小结 .....      | 37 |
| 5 系统实现 .....        | 39 |
| 5.1 开发环境 .....      | 39 |
| 5.2 成绩导入编码实现 .....  | 39 |
| 5.2.1 界面原型 .....    | 39 |
| 5.2.2 实现原理 .....    | 39 |
| 5.2.3 关键代码 .....    | 41 |
| 5.3 发布公告编码实现 .....  | 41 |
| 5.3.1 界面原型 .....    | 41 |
| 5.3.2 实现原理 .....    | 42 |
| 5.3.3 关键代码 .....    | 43 |
| 5.4 家长绑定编码实现 .....  | 43 |
| 5.4.1 界面原型 .....    | 43 |

|                          |    |
|--------------------------|----|
| 5.4.2 实现原理 .....         | 44 |
| 5.4.3 关键代码 .....         | 44 |
| 5.5 教师查看反馈家长反馈编码实现 ..... | 45 |
| 5.5.1 界面原型 .....         | 45 |
| 5.5.2 实现原理 .....         | 45 |
| 5.5.3 关键代码 .....         | 46 |
| 5.6 管理员添加班级编码实现 .....    | 47 |
| 5.6.1 界面原型 .....         | 47 |
| 5.6.2 实现原理 .....         | 47 |
| 5.6.3 关键代码 .....         | 48 |
| 5.7 本章小结 .....           | 49 |
| 6 系统测试 .....             | 51 |
| 6.1 接口测试 .....           | 51 |
| 6.2 主要功能测试 .....         | 53 |
| 6.3 本章小结 .....           | 54 |
| 工作总结与展望 .....            | 55 |
| 总结 .....                 | 55 |
| 展望 .....                 | 55 |
| 致 谢 .....                | 57 |
| 参考文献 .....               | 59 |



## 1 绪 论

### 1.1 研究背景

前苏联教育学家苏霍姆林斯基指出，家庭和学校是学生生存时间最长、人际关系最丰富的两种教育环境，对学生的全面发展有着不可替代的影响；并且根据相关报告显示，几乎 90%的家长都迫切的想了解自己子女在学校的学习以及各方面的表现情况<sup>[1]</sup>。

与此同时，社交平台的应用是家校沟通中的一种新的形式，而在这个通讯发达的时代中，学校老师与学生家长在沟通中，微信有着一个至关重要的地位。微信，毫无疑问已经成为每个智能手机必不可少的一个软件了。并且，老师或者班主任也会选择在微信群中频繁发布通知和布置相关的课题作业。

但在发布重要信息时，问题就开始逐渐出现。由于家长人数众多，导致重要信息在发布后，被群聊中不重要的日常聊天消息淹没和覆盖。这导致学校与家长沟通的有效性被降低。另一方面，在老师与家长私下沟通交流中，有时家长希望与老师能有更多的沟通，但问题就出现在，他们总是有顾虑，无法知晓老师的作息，害怕打扰到老师生活。更进一步，会发现老师也没有足够的精力去重复的回复相似问题。

以上的情况，如果一直持续发展下去，学生的学习情况、家长的问题反馈，以及教师和学校的相关通知等一系列的信息传递，都没有得到有效的完善和解决。这也导致新形式下的沟通，转变为传统的单向的沟通。

综上所述，以上诸多的问题，产生了一个平台来有效地解决学校与家长之间沟通问题的需求，使学校与家长之间的沟通能够更加的有效，有质量。

### 1.2 国内外研究现状

在国外，大量研究资料表明，美、英、日等发达国家在理论和实践上都比我国更早地开展了家庭学校合作研究，积累了丰富的经验<sup>[6]</sup>。20 世纪 90 年代至 21 世纪初，美国的家庭与学校合作进入了一个成熟阶段。

美国，在 20 世纪末，历任的总统都始终遵循着“要运用信息技术促进教育改革与发展”的战略方针。20 世纪 90 年代时，美国的高校教师就开始通过电子邮件与学

生家长进行交流；美国的加州幼儿园采用 Android 技术开发的家校通平台，家长可以通过它了解自己子女在校各种情况，还能发表在教育子女过程中的一些问题，参与学校相关讨论；在 2011 年，一款叫 Remind 的产品，已经成为美国最大的家校通产品，有超过 150 万教师和 2350 万的学生和家长通过它进行沟通；教师可以通过它群发消息给学生和家长，让家长更多的参与学校教学。有效地解决了教师与家长，学生沟通中的问题，实现有针对性的沟通<sup>[2]</sup>

而我国，相对国外发展较晚。2002 年春，“家校通”在北京电化教育研究会年会上正式露面，人们开始对它有了初步的认识和了解。2002 年 7 月，国家教育部把“基于网络条件下的家校互联、家校互动的应用研究”和《家校通的研究与应用》作为一项重点研究课题，列入国家教育部“十五规划”<sup>[11]</sup>。到 2007 年，几乎全国所有的省市都有了家校通，用户也再创新高。媒体更是对各地家校通的开通和使用情况进行了大面积的报道<sup>[11]</sup>。

相关“家校通”平台在发展中逐渐产生。我国最早期的“家校通”系统为电话家校通系统。主要为三大通信运营商——中国移动，中国联通，中国电信推出的各企业分公司与地区合作的家校通管理系统<sup>[2]</sup>。2007 年 1 月 15 日，教育部和中国移动通信有限公司在京举办了“寓教于动，移动与教育携手共进研讨会”，双方签署了《教育部与中国移动合作框架协议》，并以规模推广“家校通”，加速教育信息化的进程为目的。广东电信于 2010 年推出“翼校通”平台<sup>[2]</sup>。截止到 2011 年，陕西、河北、四川、广东、吉林、浙江、黑龙江、甘肃、江苏、内蒙古、广西、山东、河南、天津、北京、新疆、西藏、贵州等地都相继开通了家校通业务<sup>[2]</sup>。

因此，在关于学生教育的持续发展下，家校通已然会成为将来学校与家长沟通的一个重要趋势。

### 1.3 论文的组织结构

第一章 绪论，本章提出家校沟通的问题，引出家校助学联动平台的背景，再根据背景调查出国内外研究现状作对比，引出该平台在现在情景下的趋势。

第二章 核心使用技术分析，本章内容是描述开发本平台所使用的相关技术简介，如微信小程序开发的相关技术解释和优点等。

第三章 需求分析，本章会根据研究背景和研究现状对家校助学联动平台进行合理的需求与业务分析。

第四章 系统分析分析与设计，本章描述了对系统的需求分析的前提下，通过用例描述等方法，对该系统进行详细的系统与功能的分析与描述，在此基础上，完成相关模块设计等。

第五章 系统实现，本章描述针对详细设计得到的系统分析，对系统部分主要功能的编码实现，并完成系统的架构设计、数据库设计、界面设计以及功能设计。

第六章 系统测试，本章对已经完成的系统，进行相关的单元测试和系统测试的描述。

## 1.4 本章小结

本章结合研究背景，对家校助学进行相关调研，并在此基础上，分析了国内外研究现状，发现国外的针对家校助学相关研究远远早于国内，但在互联网发达的今天，家校助学的研究也在快速的成熟起来。





## 2 关键技术分析

开发一个系统，选择一个好的技术，是一个至关重要的环节，如果技术选择不正确，则会影响后期开发效率和质量，因此要根据平台需求和当下现状合理选择技术框架。

### 2.1 技术框架

#### 2.1.1 Vue.js

网页开发快速发展，这使得网页变得更加动态化，因此 Javascript 在网页开发中，也就起到了一个必不可少的作用。在曾经硕大的服务端代码让人烦恼的时候，Javascript 也就开始将一些服务端处理的代码带到了前端。但是在前端杂乱的 HTML 文件和 CSS 文件下，缺乏一个组织和管理框架。于是，各个开发者为了提高开发效率，Angular、React、Vue 等框架就逐渐出现在开发者面前，并且他们得到了广泛应用和青睐。在众多框架中，Vue 是一个开发友好、用途广泛，且高性能的渐进式框架。

官方介绍，Vue 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

当然，值得注意的是，Vue 在一定程度上，就相当于一个现成的服务端应用，可以将它作为该应用的一部分嵌入其中，带来更加丰富的交互体验。例如你希望将更多业务逻辑，转移到前端来实现，就可以使用 vue 基础的核心库，以及它的生态系统，如常见的 Vuex（状态管理）、Vue-Router（路由管理）等。

除此之外，vue 还可以将网页分割成多个可以复用的组件形式，即封装为一个组件文件，而每个组件文件会包含属于自己的 HTML、CSS、Javascript，通过组件之间的拼接，类似于拼图，将各个组件的块儿，渲染到网页中相应的地方。

#### 2.1.2 TypeScript

TypeScript 是一种由微软开发的开源、跨平台的编程语言。它是 JavaScript 的

超集，TypeScript 的代码最终会通过编译器，直接编译为 JavaScript 代码。

由于 ECMAScript 标准严格的审核机制，有时候很多新提出的新特性无法及时的发布，并应用到新版本的 JavaScript 中。因此 TypeScript 为了满足一些开发者的需求，就开发出了这门语言，它添加了可选的静态类型系统，以及很多尚未正式发布的 ECMAScript 新特性等。

值得注意的是，Node.js 或者是 JavaScript 语言，它本身是一个弱类型的脚本语言，对于值的数据类型具有不确定，这往往就导致了很多隐患的问题，因为值类型出错而无法被察觉，导致后期维护成本增加。而对于 Typescript 语言而言，它是一门强类型的语言，每一个变量的定义，都需要对数据类型定义进行严格的声明，因此，这种特性，就很好的解决了 JavaScript 数据类型不确定性的隐患。它也与很多服务端语言相似，如 Java 语言。

### 2.1.3 koa

Koa 是一个由 Express 幕后的原班人马打造的 web 框架，它通过利用 async 函数，让开发者丢弃回调函数，并有力地增强错误处理。当然，Koa 也是一个渐进式框架，若要更好的处理服务端业务，可以使用它的生态系统中的相关中间件，如 koa-router（路由处理），koa2-cor（跨域处理）等。

### 2.1.4 uni-app(微信小程序)

uni-app 是一个使用 Vue.js 开发所有前端应用的框架，开发者编写一套代码，可发布到 iOS、Android、H5、以及各种小程序（微信/支付宝/百度/头条/QQ/钉钉）等多个平台。

本次开发只针对微信小程序这一个单独的平台进行编码开发。该框架对会 Vue 的开发者十分友好，减少了学习微信小程序的成本，无需额外学习微信编码的内容，只需要了解 uni-app 官方 API 即可。

## 2.2 开发工具

本平台的开发工具大致如下：

- (1) MockingBot（墨刀）：该工具为原型设计软件；
- (2) HBuilderX：该工具为微信小程序开发编辑器；

(3) WebStorm: 该工具为后台管理系统前端开发编辑器;

(4) Visual Studio Code: 该工具为后端开发编辑器;

(5) PostMan: 该工具为接口调试工具;

(6) Navicat Premium 12: 该工具为数据库可视化操作工具, 以及数据库物理模型设计工具。

### 2.3 本章小结

本章主要讲解了开发使用的相关框架进行介绍, 总结出对应框架在开发中较传统开发的一些突出优势。最后, 列举了针对不同系统所需要使用的工具。



## 3 需求分析

了解到一个需求后，就需要将需求这些想法落实到具体的描述上，让需求不再是概念和抽象的东西，而是一个能够具体落实的点和实施的业务流程。因此，需要对前期系统设计的构思和想法进行需求分析，并将它进行具体描述，以呈现给用户们和开发者们。

需求分析，不仅仅只是简单分析，还需要根据实际情况，划分具体的范围，防止和避免模棱两可的概念，导致后期开发因理解不到返工，而影响开发效率。

### 3.1 需求描述

目前，学校和家庭交流和沟通的方式，基本上都是通过 QQ 群或者微信群。老师发布一则通知时，群里的信息，多数都是家长都一些大量重复的话，比如“收到”，“谢谢老师”等无价值信息，以上这种刷屏现象，就会导致后续看消息的人需要在很多无价值信息中去寻找有价值信息，这时候，往往会不经意间遗漏了一些关键的有用信息。另外，老师要求学生购买一本书或者其他用品时，部分家长希望能够统一购买，但目前基本通过人工根据微信消息进行统一，费时同时容易出错。更重要的是，因为不同的同学各门课程的学习情况不同，再家长和老师都希望能够通过及时获得互动，来实时掌握学生的学习情况以及学生存在的问题，及时纠正和弥补不足的地方。

针对以上困扰，总结出以下需求描述：

学校管理员或者老师可以通过后台管理系统将学生、教师，学校课程以及学生成绩等相关的信息进行处理。通过后台的数据处理完善之后，老师和家长就可以通过微信小程序绑定和查看班级或者学生的学习情况。学校或者老师有需要通知的公告、作业或者统计需要发布的时，可以登录微信小程序，进行发布。发布之后，家长可以登录微信小程序，对通知进行查看，然后结合通知再对学生进行辅导和监督。与此同时，家长要是在学生这边发现和相关的问题，也可以通过微信小程序，针对老师发布相关反馈。家长发布反馈后，老师也可以登录微信小程序，对家长的反馈进行查看，并进行接收确认，然后针对问题着手去解决。

### 3.2 系统角色分析

通过对需求的了解和相关调研，得出家校助学联动平台需要实现两个系统，一个是针对学校对老师家长等相关信息管理的后台管理系统；另一个则是解决老师与家长之间沟通问题、处理相关信息的收集，以及展示学生学习情况的平台——微信小程序。

根据不同系统的职能，一共分配出了 4 个角色，分别是管理员、班主任、任课老师和家长。而不同的角色有着自己不同的业务流程。根据业务需求，得到如下分析：

(1) 管理员，是管理学校老师、学生和学生家长相关信息的角色。他基本的业务流程大致为登录后台管理系统后，对教师、家长、学生、班级、课程、学期、成绩、公告和反馈这些信息进行处理；

(2) 班主任，是沟通学生家长，管理班级信息，和处理公告和反馈等工作的角色。他的业务流程分为两部分，一部分是登录后台管理系统，对学生、家长、成绩、公告和反馈进行信息处理；另一部分则是登录微信小程序，查看班级学生及其家长情况，和发布公告、处理反馈信息等操作。

(3) 任课老师，是沟通家长，处理公告和反馈信息的角色。他也和班主任角色相同，业务流程分为后台管理系统和微信小程序两部分业务流程，微信小程序和班主任业务流程相同，后台管理系统则只有公告反馈和反馈查询的操作；

(4) 家长，是了解学生信息动态，和老师反馈和沟通的角色。他的业务流程在微信小程序上，通过登录查看学生学习情况、了解孩子课程老师信息、浏览教师公告动态，以及重要的发布问题反馈。

### 3.3 系统流程分析

综上分析，将各个角色业务流程整合，得出对需求描述进行业务和角色分析后的业务流程图（图 3.1）。

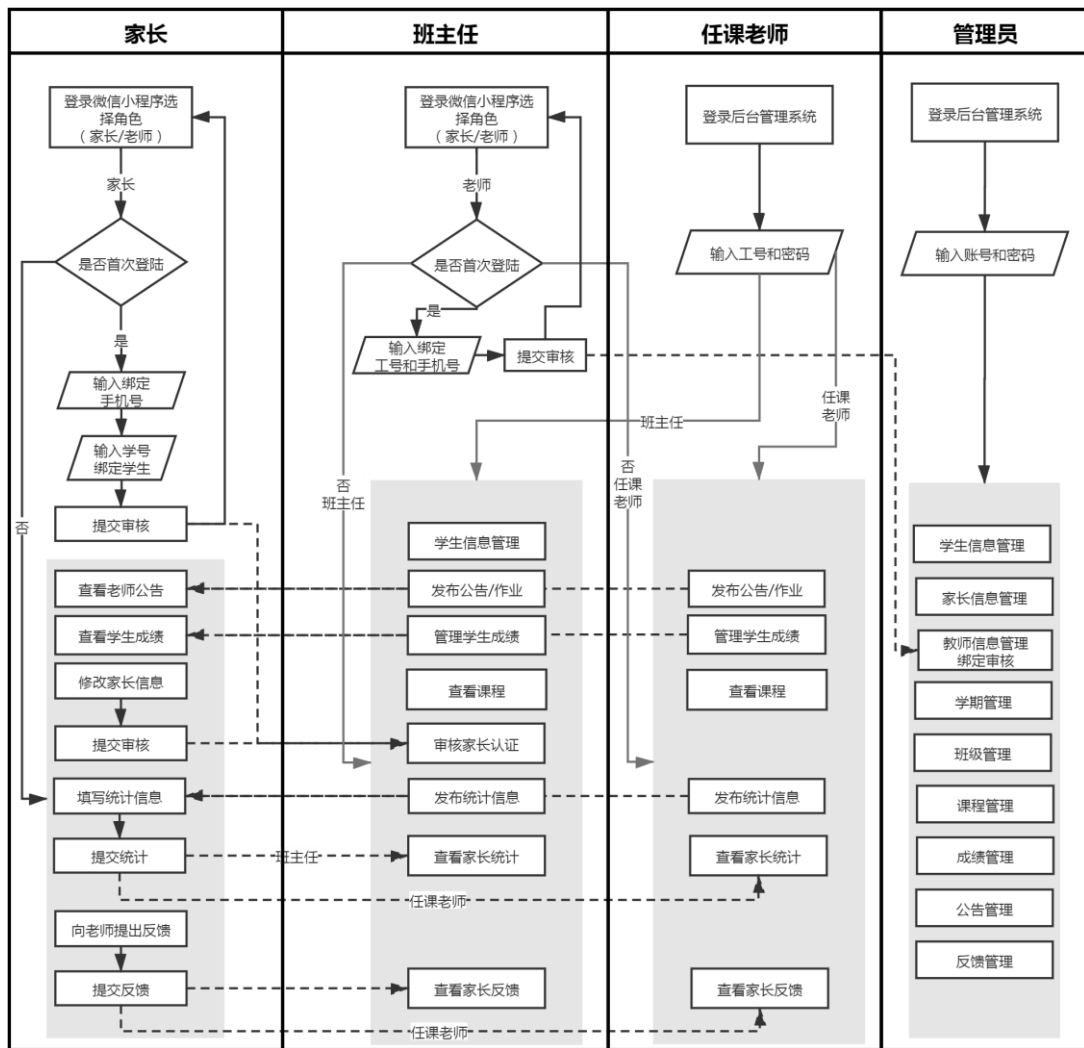


图 3.1 业务流程图

### 3.4 功能需求分析

将需求分析中得到的业务流程图进行分析，将各个功能点细化。然后，分别把后台管理系统和微信小程序的具体功能点统计起来，进行详细的描述。

#### 3.4.1 后台管理系统功能分析

在后台管理系统中，主要的功能是对学校基础信息的管理，如学生、老师、课程、班级等基础信息的处理，当然也有对家长相关信息处理和审核功能。

后台管理系统里，有两种角色，一个管理员角色，一个老师角色，管理员可以使用所有管理功能，而老师也需要分情况，如果是班主任，能处理学生管理、家长管理、

成绩管理、公告管理和反馈管理；如果是任课教师则只有公告管理和反馈管理。具体功能需求分析如下表 3.1 所示：

表 3.1 后台管理系统功能需求分析

| 序号 | 功能   | 描述   |
|----|------|--|
| 1  | 学生管理 | 对学生基本信息的增加、修改、删除、查询操作                      |
| 2  | 家长管理 | 对家长基本信息的增加、删除、修改、查询操作                      |
| 3  | 教师管理 | 对家长基本信息的增加、删除、修改、查询操作，以及对教师班级、课程和教师职务的分配操作 |
| 4  | 学期管理 | 对学期信息的增加、修改、删除、查询操作                        |
| 5  | 班级管理 | 对家长基本信息的增加、修改、删除、查询操作，以及课程、教师和班主任的分配       |
| 6  | 课程管理 | 对课程信息的增加、修改、删除、查询操作                        |
| 7  | 成绩管理 | 对学生成绩的增加、修改、删除、查询操作                        |
| 8  | 公告管理 | 对公告的发布、修改、删除、查询操作                          |
| 9  | 反馈管理 | 对反馈的查看和删除操作                                |

### 3.4.2 微信小程序家长功能分析

家长在本平台只有微信小程序的需求，微信小程序可以很方便的通过老师的分享或者搜索进入。

而家长主要可以通过微信小程序的操作有：对公告、作业等老师发布的信息进行查询、了解学生课程老师的部分信息与学生各个科目的成绩情况，以及针对老师发布学生、课程、教师、作业等问题的反馈信息，家长具体的功能需求分析，如下表 3.2 所示：



表 3.2 关于家长的微信小程序功能需求分析

| 序号 | 功能     | 描述                                      |
|----|--------|---|
| 1  | 公告列表   | 查看老师发布的关于孩子的公告、作业或者统计                   |
| 2  | 发布反馈   | 根据家长自己需求，想老师提出关于学生、课程、作业、老师和统计不同方面的反馈需求 |
| 3  | 学生信息列表 | 展示家长孩子的基础信息                             |
| 4  | 学生成绩列表 | 展示家长孩子的学习成绩情况                           |
| 5  | 反馈管理   | 改变和查看家长自己发布的反馈信息及其状态，以及删除反馈信息           |
| 6  | 课程信息列表 | 查看家长孩子的课程以及对应课程老师的联系方式                  |

### 3.4.3 微信小程序教师功能分析

老师可以使用两个系统，一个即上面提到的后台管理系统，另一个则就是和家长进行对应沟通的微信小程序。

老师在微信小程序的功能需求大致分为班级学生基本信息和其家长部分信息查看、发布对应类型公告，以及处理家长反馈三个主要功能。具体功能需求分析如下表 3.3 所示：

表 3.3 关于教师的微信小程序功能需求分析

| 序号 | 功能       | 描述                      |
|----|----------|-------------------------|
| 1  | 公告列表     | 查看老师发布的关于老师自己的公告、作业或者统计 |
| 2  | 发布公告     | 发布公告、作业或者统计             |
| 3  | 班级课程列表   | 展示在不同班授课的课程基本信息         |
| 4  | 反馈管理     | 处理修改反馈信息及其状态            |
| 5  | 班级成绩列表   | 展示班级学生成绩情况              |
| 6  | 公告和作业管理  | 查看公告、作业和统计，以及删除操作       |
| 7  | 学生家长信息展示 | 展示各个学生家长的姓名和联系方式        |

### 3.5 用例图

通过总结和设计，系统的功能被逐一的分类罗列出来，因此，现在就可以通过这些设计，进一步的分析和设计出用例图，来寻找各个参与者与各个功能用例之间的关系。

一般情况下，用例图是由参与者、用例、箭头、系统边界组成，而用例图的作用在于获取需求和指导测试。当然，也少不了指导在整个过程中其他工作流的作用。而在用例之间的关系中也存在着包含，泛化等关系，根据实际情况会有不同的关系。本次系统设计相对简单，设计用例中只涉及了包含关系，因此未标注包含关系的说明。

根据前面的角色分析，可以了解到参与者为家长、任课老师、班主任和管理员。各个角色之间也存在这相同的功能用例和不同的功能用例，因此需将所有功能罗列出来，进行角色与用例，用例与用例之间的关联，经过整理得到以下用例图（如图 3.2）：

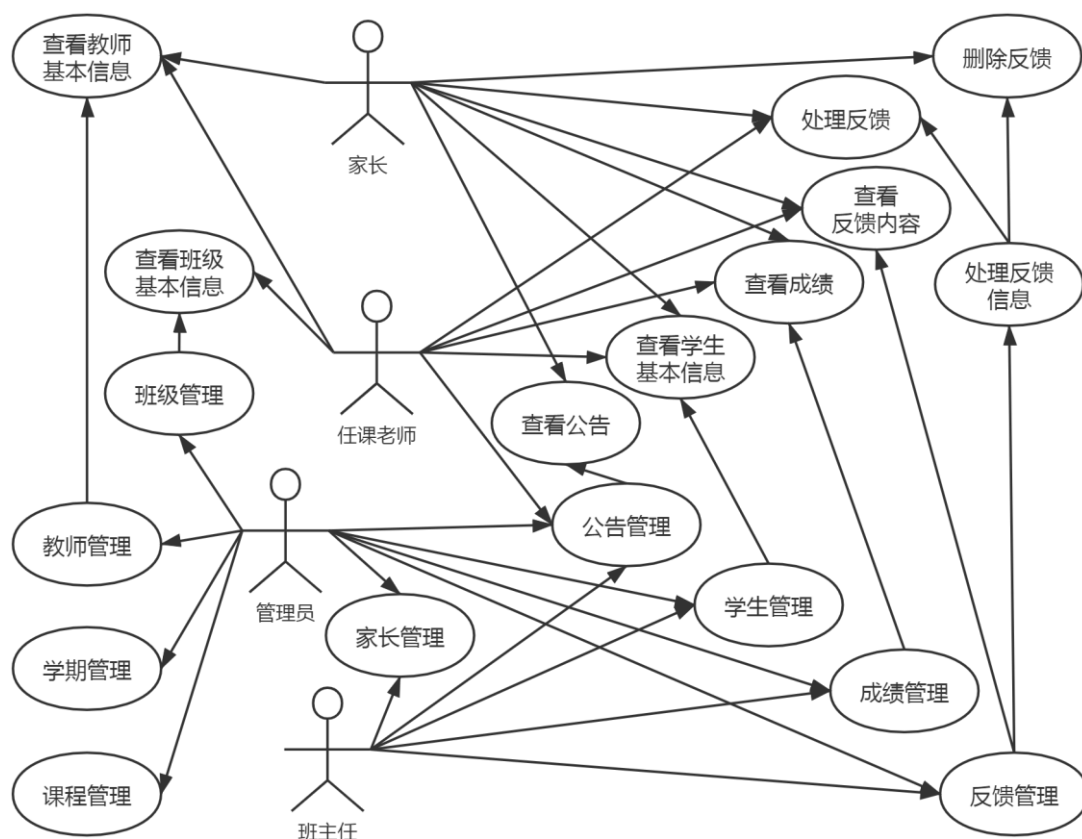


图 3.2 用例图

通过图 3.2 的用例图，我们可以通过子系统的角度，对各个用例进行统计，并对

用例进行描述，具体内容如下表 3.4，表 3.5 所示：

表 3.4 后台管理系统用例表

| 角色                 | 子系统  | 用例名称   | 用例功能描述        |
|--------------------|------|--------|---------------|
| 管理员<br>班主任<br>任课教师 | 登录   | 登录     | 后台管理系统登录      |
| 管理员                | 教师管理 | 教师信息查询 | 教师信息列表展示      |
|                    |      | 教师信息处理 | 教师信息的增加、删除、修改 |
| 管理员<br>班主任         | 学生管理 | 学生信息查询 | 学生信息列表展示      |
|                    |      | 学生信息处理 | 学生信息的增加、删除、修改 |
| 管理员<br>班主任         | 家长管理 | 家长信息查询 | 家长信息列表展示      |
|                    |      | 家长信息处理 | 家长信息的增加、删除、修改 |
| 管理员<br>班主任         | 学期管理 | 学期信息查询 | 学期信息列表展示      |
|                    |      | 学期信息处理 | 学期信息的增加、删除、修改 |
| 管理员<br>班主任         | 班级管理 | 班级信息查询 | 班级信息列表展示      |
|                    |      | 班级信息处理 | 班级信息的增加、删除、修改 |
| 管理员<br>班主任         | 课程管理 | 课程信息查询 | 课程信息列表展示      |
|                    |      | 课程信息处理 | 课程信息的增加、删除、修改 |
| 管理员<br>班主任         | 成绩管理 | 成绩信息查询 | 成绩信息列表展示      |
|                    |      | 成绩信息处理 | 成绩信息的增加、删除、修改 |
|                    |      | 成绩导入   | 成绩信息的导入       |
| 管理员<br>班主任         | 公告管理 | 公告信息查询 | 公告信息列表展示      |
|                    |      | 公告信息处理 | 公告信息的增加、删除、修改 |
| 管理员<br>班主任         | 反馈管理 | 反馈信息查询 | 反馈信息列表展示      |
|                    |      | 反馈信息处理 | 反馈信息的删除       |

表 3.5 微信小程序用例表

| 角色    | 子系统 | 用例名称     | 用例功能描述                            |
|-------|-----|----------|-----------------------------------|
| 教师/家长 | 绑定  | 绑定微信     | 家长/教师填写基本信息和绑定微信号                 |
| 教师/家长 | 公告  | 查看首页公告信息 | 查看发布的公告信息                         |
| 教师    | 发布  | 发布公告     | 发布公告/作业/统计                        |
| 教师    |     | 发布反馈     | 发布课程/作业/学生/教师/统计反馈                |
| 教师/家长 | 我的  | 反馈管理     | 处理和查看反馈信息                         |
| 教师/家长 |     | 切换账号     | 退出账号或者切换身份                        |
| 教师/家长 |     | 成绩管理     | 家长查看孩子各科成绩信息；<br>老师查看各个班级学生课程成绩信息 |
| 教师    |     | 公告管理     | 公告/统计删除和查看                        |
| 教师    |     | 作业管理     | 作业删除和查看                           |
| 教师    |     | 班级管理     | 查看学生课程老师/拨打电话                     |
| 教师    |     | 课程管理     | 查看学生课程老师/拨打电话                     |
| 家长    |     |          |                                   |

### 3.6 用例描述

得到用例图之后，为了进一步明确用例之间的关系，我们就需要编写出详细的用例描述，来将各个系统功能点具体化。有了用例描述，就起到为后期系统测试提供参考和指导的作用。无需开发者或者测试人员，再去做相应的功能点梳理。也让开发者在编码时有了一个明确的需求和清晰的工作流，大大提升了编码的效率。

由于系统功能点相对较多，在此只列举几个相对重要的用例来阐述，其他用例描述做此类比即可，具体内容，参考表 3.6、表 3.7、表 3.8、表 3.9、表 3.10：

表 3.6 管理员成绩导入-用例描述

|          |  |
|----------|--|
| 用例名称     | 管理员成绩导入  |
| 用例标识号    | 001  |
| 参与者      | 管理员  |
| 简要说明     | 管理员通过后台管理系统批量导入学生成绩  |
| 前置条件     | 管理员已经登录后台管理系统  |
| 基本事件流    | <ol style="list-style-type: none"> <li>1. 登录后台管理系统，点击“成绩管理”侧边菜单栏选项；</li> <li>2. 系统出现成绩管理页面，点击“导入”按钮；</li> <li>3. 选择导入成绩的表格。</li> </ol> |
| 其他事件流 A1 | 无  |
| 异常事件流    | <ol style="list-style-type: none"> <li>1. 表格内容错误；</li> <li>2. 表格导入失败；</li> <li>3. 网络异常，无法提交。</li> </ol>                                |
| 后置条件     | 成绩管理页面显示新导入的成绩列表   |
| 注释       | 无  |

表 3.7 教师发布公告-用例描述

|          |  |
|----------|--|
| 用例名称     | 教师发布公告并查看  |
| 用例标识号    | 002  |
| 参与者      | 班主任/任课教师   |
| 简要说明     | 班主任/任课教师向学生发布公告、作业、统计，发布之后并查看内容  |
| 前置条件     | 班主任/任课教师已经登录微信小程序  |
| 基本事件流    | <ol style="list-style-type: none"> <li>1. 点击导航栏“发布”图标；</li> <li>2. 系统出现公告表单，填写表单信息；</li> <li>3. 点击“发布”按钮发布公告。</li> <li>4. 发布成功之后，点击导航栏“首页”图标；</li> <li>5. 系统出现首页的公告列表，点击相关公告查看。</li> </ol> |
| 其他事件流 A1 | 无  |
| 异常事件流    | <ol style="list-style-type: none"> <li>1. 表单信息未填写完成；</li> <li>2. 网络异常，无法提交。</li> </ol>   |
| 后置条件     | 小程序首页和公告管理页面新增一条公告信息   |
| 注释       | 无  |

表 3.8 家长绑定-用例描述

|          |  |
|----------|--|
| 用例名称     | 家长绑定   |
| 用例标识号    | 003  |
| 参与者      | 家长   |
| 简要说明     | 家长通过微信小程序绑定自己的基本信息   |
| 前置条件     | 家长未绑定个人信息  |
| 基本事件流    | <ol style="list-style-type: none"> <li>1. 打开微信小程序，进入绑定页面，选择绑定角色，点击“家长”按钮；</li> <li>2. 系统页面出现家长基本信息表单页面，填写表单；</li> <li>3. 点“下一步”按钮，系统出现绑定孩子信息表单页面，填写孩子信息；</li> <li>4. 点击“绑定”按钮，提交表单。</li> </ol> |
| 其他事件流 A1 | 无  |
| 异常事件流    | <ol style="list-style-type: none"> <li>1. 表单信息未填写完成；</li> <li>2. 网络异常，无法提交。</li> </ol>   |
| 后置条件     | 后台管理系统家长管理列表中对应家长信息更新  |
| 注释       | 无  |

表 3.9 教师查看家长反馈-用例描述

|          |   |
|----------|---|
| 用例名称     | 教师查看家长反馈  |
| 用例标识号    | 004   |
| 参与者      | 教师  |
| 简要说明     | 教师通过微信小程序查看家长反馈内容   |
| 前置条件     | 教师已经登录微信小程序   |
| 基本事件流    | <ol style="list-style-type: none"> <li>1. 点击底部导航栏“我的”图标；</li> <li>2. 系统出现教师个人信息和相关菜单；</li> <li>3. 点击“反馈管理”按钮，系统出现家长反馈列表；</li> <li>4. 点击反馈列表要查看的项，系统出现反馈详细信息。</li> </ol> |
| 其他事件流 A1 | 无   |
| 异常事件流    | <ol style="list-style-type: none"> <li>1. 网络异常，无法获取。</li> </ol>   |
| 后置条件     | 无   |
| 注释       | 无   |

表 3.10 管理员添加班级-用例描述

|          |   |
|----------|---|
| 用例名称     | 管理员添加班级   |
| 用例标识号    | 005   |
| 参与者      | 管理员   |
| 简要说明     | 管理员通过后台管理系统新增新的班级及其信息   |
| 前置条件     | 管理员已经登录后台管理系统   |
| 基本事件流    | <ol style="list-style-type: none"> <li>1. 点击菜单侧边栏“班级管理”选项；</li> <li>2. 系统出现班级信息列表；</li> <li>3. 点击页面“新增”按钮，系统弹出新增班级需要填写的信息的表单；</li> <li>4. 填写表单信息，点击“提交按钮”。</li> </ol> |
| 其他事件流 A1 | 无   |
| 异常事件流    | <ol style="list-style-type: none"> <li>1. 表单信息未填写完成；</li> <li>2. 表单信息填写错误；</li> <li>3. 网络异常，无法提交。</li> </ol>  |
| 后置条件     | 后台管理系统班级管理列表中新增一条班级信息   |
| 注释       | 无   |

### 3.7 本章小结

本章描述了家校助学联动平台的需求，并通过需求对平台业务和角色进行了分析，得出对应的业务流程图，最后通过业务流程图分析个人业务中的具体功能点，并对其进行分析和描述，为后期系统设计提供参考。





## 4 系统分析与设计

需求分析之后的信息，都是一个个零散的点，即系统的不同需求功能点。有了这些点，我们就需要寻找这些点的关系，把他们联系起来，形成关联，让各个功能能有对应的逻辑关系，让后续开发有准确的参考。

系统的分析与设计，直接影响到后期的开发，如果设计不当，就会为后期带来不必要的工作量，甚至严重的系统逻辑漏洞无法弥补。

### 4.1 技术架构设计

本平台开发使用的技术，整体分为两部分：一个是前端开发技术，另一个则是后端开发技术。开发模式，则是前后端分离开发，前端的数据通过 axios 异步请求接收，数据则通过相应 Vue.js 进行渲染，后台则通过 koa 框架以及相应路由处理进行请求识别；接受到请求后，在 mysql2 框架的处理下，将数据库查询出来，然后将查询数据进行相应处理形成接口所要求的的数据结构相应给前端接口。详细技术框架结构如下图 4.1 所示：

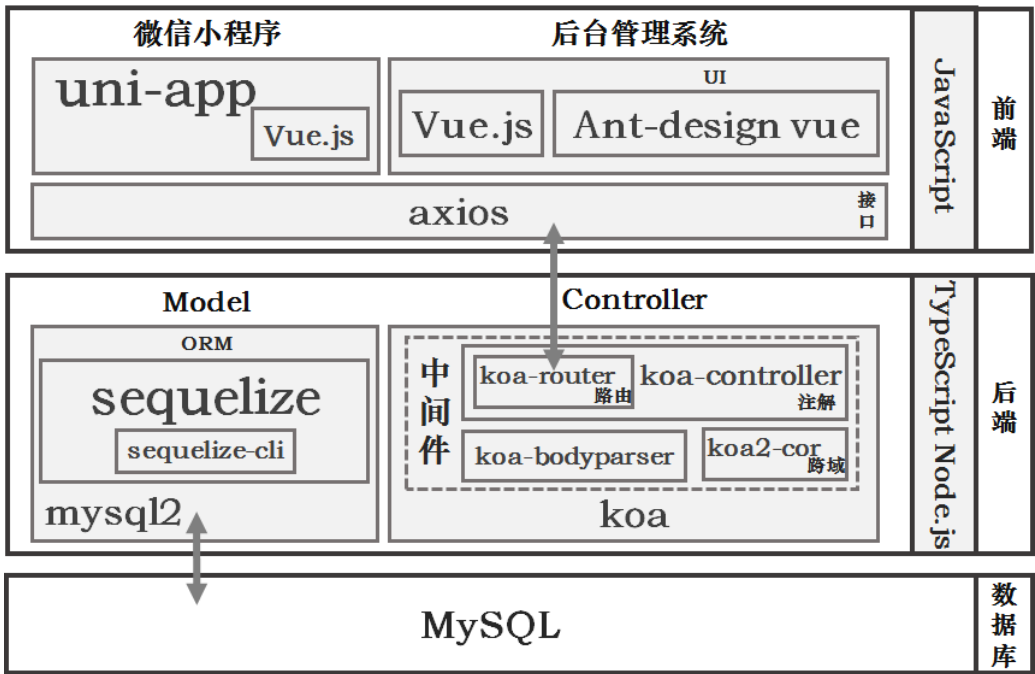


图 4.1 技术框架结构图

如图，前端开发分为微信小程序开发和后台管理系统开发。

(1) 微信小程序的开发是基于 Vue.js 的 uni-app 前端应用框架，它可以实现多个平台统一开发，减少了学习成本。本系统的开发则只针对微信小程序这一个平台。

(2) 后台管理系统的开发则是以 Vue.js 为主导开发，其中后台管理系统 UI 框架使用的是 Ant-design vue 框架进行页面快速搭建。极大的提高了，静态页面开发的效率。

由于开发模式为前后端分离，则微信小程序和后台管理系统的服务端统一为一个，只需要有对应的接口请求。

后端开发是使用 TypeScript 语言和 Node.js 开发的。本系统相对简单，TypeScript 可以将系统做到轻量级，有别于其他语言开发，让系统空间占用庞大。而 TypeScript 语言是一个强类型语言，可以在变量上对其类型进行制约，这样可以极大的减少后端开发中变量类型不匹配而出错，且不易察觉的问题。

本平台开发的后端开发模式，分为 Model 和 Controller 两部分。

(1) Model 部分使用的是 sequelize 框架，sequelize 是基于 mysql2 数据库查询框架的 ORM 框架。mysql2 是通过 sql 语句对数据库进行查询，而 sequelize 则是将 JDBC 转换为面向对象的概念，使每一个操作的表变为一个实体类对象，因此，数据查询则被转化为对象关系映射，即 ORM(Object Relational Mapping)。而 ORM 的好处就在于，可以将开发的重心不放在对 sql 语句查询上，去专心处理业务逻辑。

(2) Controller 部分是使用的 koa 框架进行开发，koa 框架是在 Express 的基础上再出开发的框架，它是一个渐进式的框架，框架本身有许多中间件对该框架进行支撑，而本系统则采用了 koa-controller、koa-bodyparse 等中间件进行开发。通过 koa-controller 中注解，可以简化对路由处理的操作，让开发重心不放在路由处理上，而专注与业务处理上。

## 4.2 系统架构设计

通过系统需求设计与分析、系统功能设计与分析以及数据的设计与分析，再结合上一章节选择的技术框架。我们就可以为系统实现前做系统架构的设计，为后期编码做好技术准备。结合本章节设计的结果，可设计出如下图 4.2 的系统架构设计：

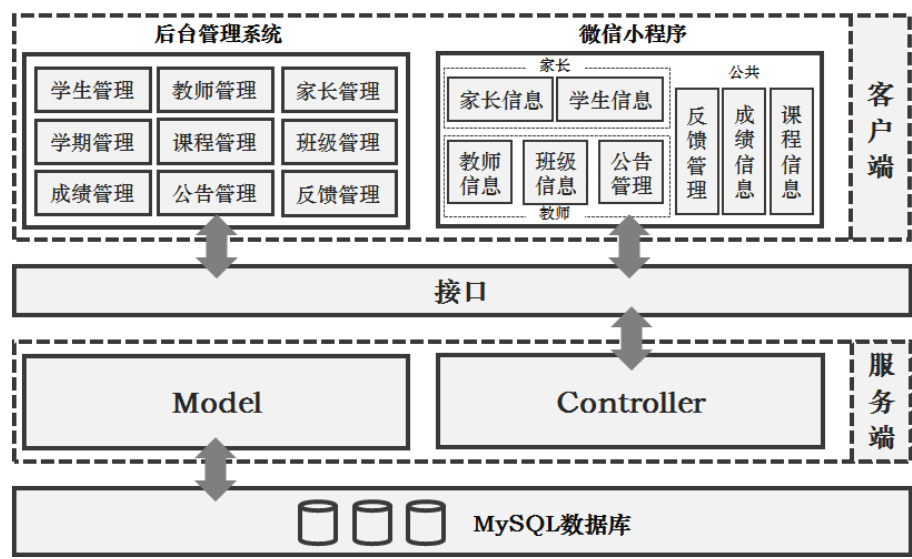


图 4.2 系统架构图

### 4.3 功能设计

上一章节，我们得到需求分析、业务和角色分析的描述，因此现在就可以将它们进行整合得到系统对应的系统功能模块图（如图 4.3、4.4）。

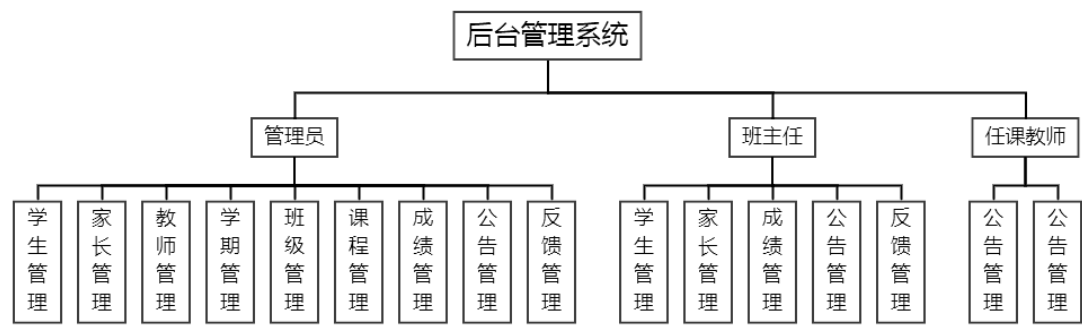


图 4.3 后台管理系统-系统功能模块图

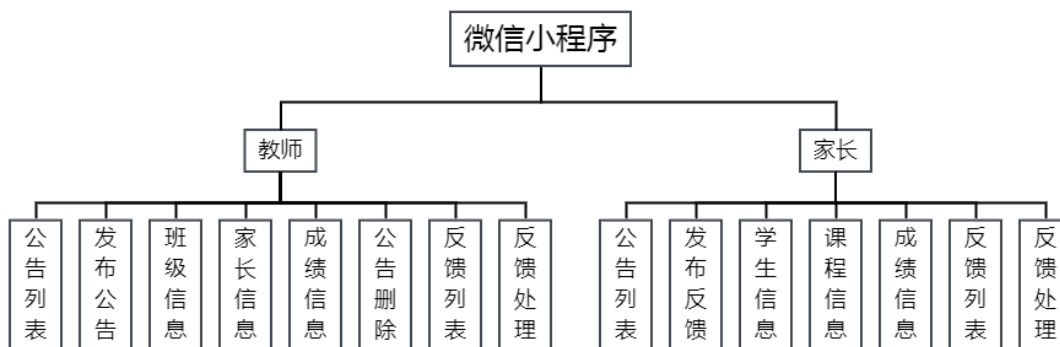


图 4.4 微信小程序-系统功能模块图

有了以上的系统功能模块图，我们就能更加明确的得到家校助学联动平台内部组织关系，也更加清晰的理清各个角色和各个功能的内部逻辑关系，也规范了各自功能部分，让它更加有条理。

## 4.4 功能模块设计

得到系统功能模块图之后，我们就可以更加细化，针对各个功能模块进行详细设计，这里将详细设计三个主要功能模块。

### 4.4.1 成绩导入

成绩导入是后台管理系统成绩管理里的一个功能模块，通过分析，我们可以得到导入成绩时序图（图 4.5）。

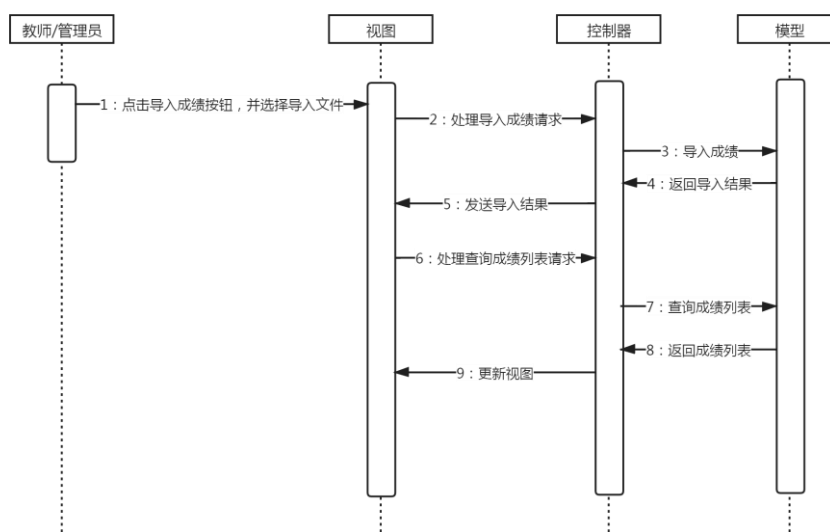


图 4.5 导入成绩时序图

教师或者管理员点击后台管理系统导入成绩按钮，就会选择有成绩的文件，让后经过处理后，将处理后的数据通过请求发送给控制层，控制层则处理之后调用模型层方法，导入成绩到数据库，导入之后把导入的结果返回给控制层，控制层则将返回结果返回响应给视图，即页面。

页面接受到响应之后，需要更新成绩列表，即视图发送成绩列表请求，控制层接受处理，模型层查询，然后返回结果，通过控制层再返回给视图，视图渲染新的成绩列表。

在成绩导入中，主要涉及的对象有 Grade（成绩）、Class（班级）、Term（学期）、Course（课程）、Student（学生），其主要包含的属性和方法如下图 4.6 所示：

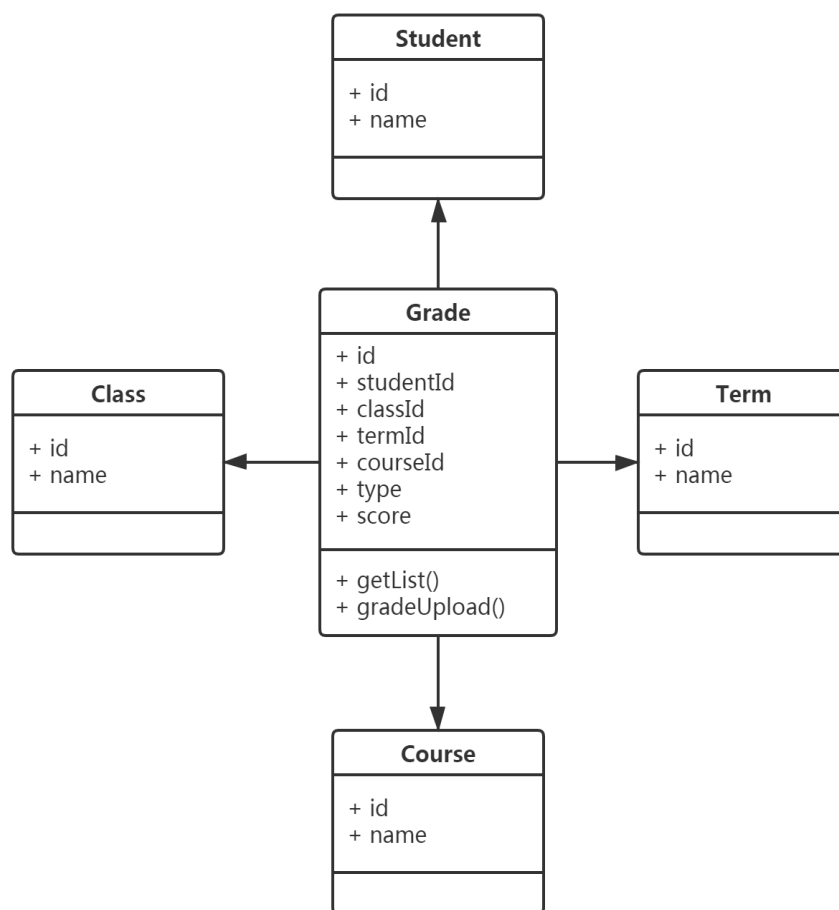


图 4.6 导入成绩类图

#### 4.4.2 发布并查看公告

发布并查看公告是老师在微信小程序里的一个功能模块，通过分析，我们得到如下图 4.7 所示时序图：

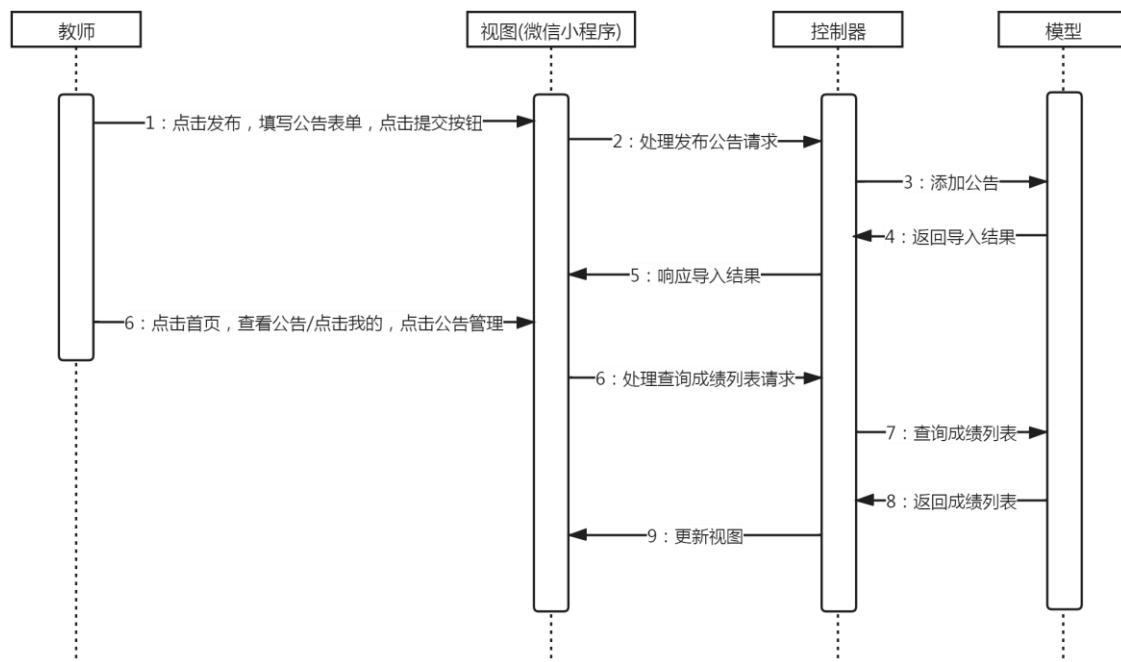


图 4.7 发布并查看公告时序图

如时序图所描述，教师通过登录小程序，点击“发布”，填写发布公告表单，并按下提交按钮，微信小程序则发送发布公告请求给控制层，控制层调用方法添加公告到数据库，并返回结果给控制层，控制层返回结果给微信小程序。

微信小程序得到结果反馈给教师，教师点击“首页”或点击“我的”里的公告管理，查看公告，此时微信小程序发送公告列表请求，控制层接受处理，模型层查询，然后返回结果，通过控制层再返回给视图，视图渲染新的公告列表。

在发布并查看公告中，主要涉及的对象有 Release（公告）、Course（课程）、Class（课程）、Teacher（教师），其中包含的属性和方法如下图 4.8 所示：

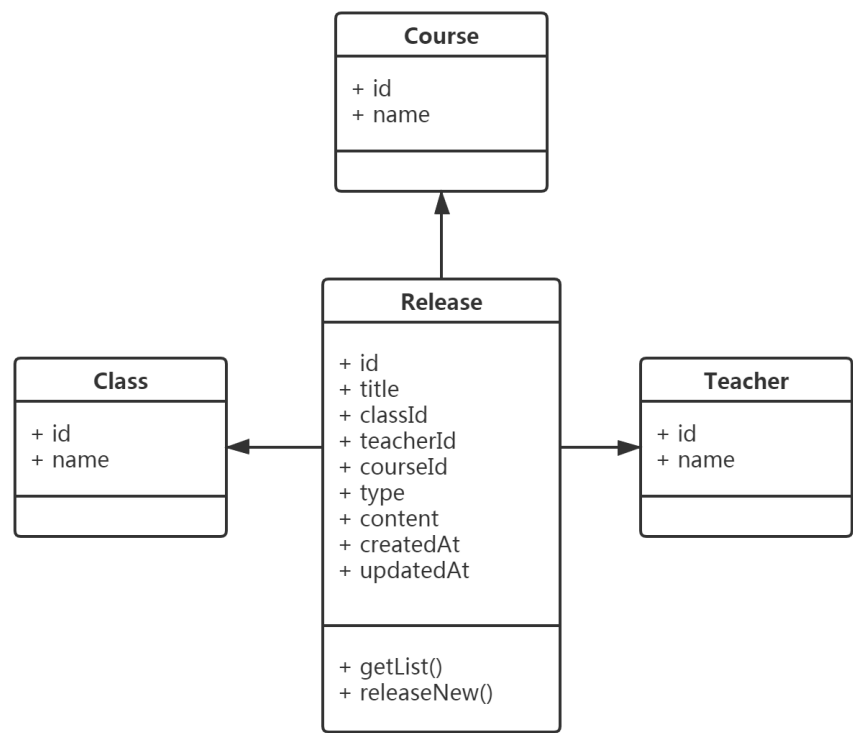


图 4.8 发布并查看公告类图

4.4.3 家长绑定

家长绑定是通过微信小程序填写家长和孩子基本信息，老师登录后台管理系统审核的功能模块，通过分析，我们可以得到如图 4.9 的时序图：

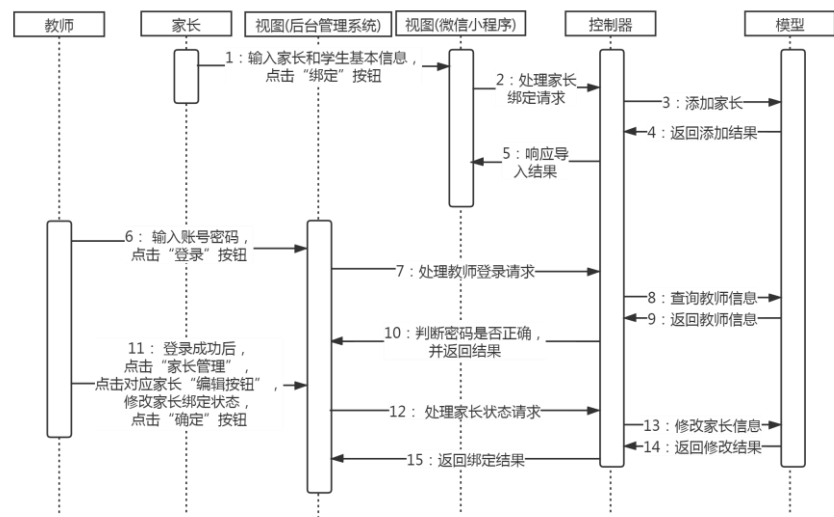


图 4.9 家长绑定时序图

家长在微信小程序中，输入家长和学生基本信息，点击绑定之后，视图发送处理请求，控制器接收处理，模型进行家长添加，最后返回结果。

然后，教室登录后台管理系统，修改家长绑定状态，经过一系列处理，修改了家长数据，完成家长绑定。

在绑定家长中，主要涉及的对象有 Parent（家长）、Student（学生）、Main（登录）、Teacher（教师），其中包含的属性和方法如下图 4.10 所示：

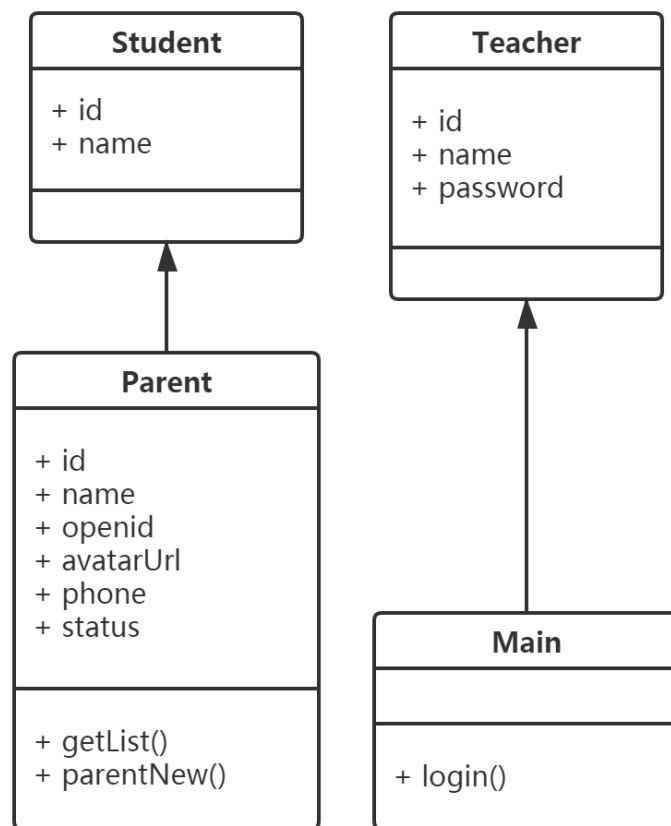


图 4.10 家长绑定类图

## 4.5 数据库设计

经过详细的分析与设计之后，到了系统设计至关重要的环节——数据库设计。数据存储的结构在很大程度上会影响后期编码的效率和质量，数据库设计如果关系混乱，就会影响数据查询速度；如果字段重复冗余，就会影响存储空间，增加服务器负担。



4.5.1 概念结构设计

根据前面所分析的功能和用例关系，将各个功能需要存储的数据分类封装为一个实体，通过相互存在的关系，设计为互相关联的实体，并对所有数据信息进行建模。

建立概念模型通常需要使用 E-R 图来对数据信息进行描述。有功能分析可以分析出学生、教师、家长、课程、班级、成绩、公告、反馈这几个实体，实体与实体之间的具体描述如下图 4.11 所示：

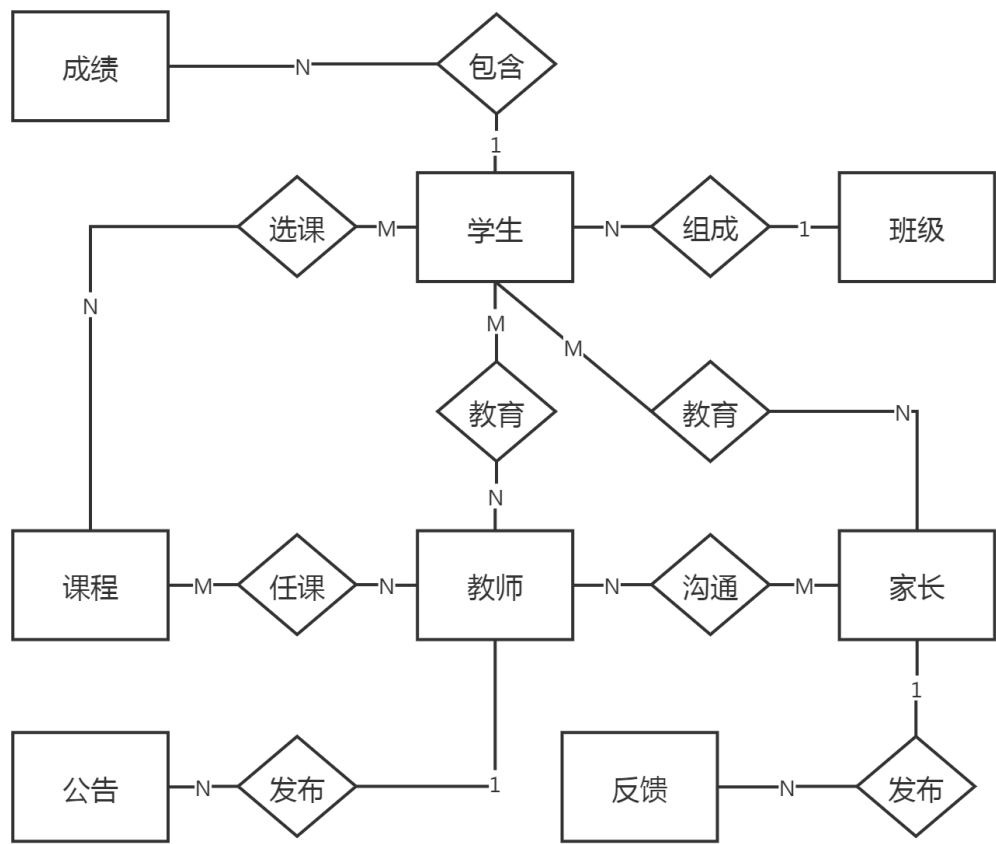


图 4.11 E-R 图

4.5.2 逻辑结构设计

根据 E-R 图，我们可以看到，实体与实体之间存在很多多对多关系，为了开发的时候，能够更好的处理数据，减少开发复杂性，我们需要针对多对多关系，设计中间表。因此在以下的逻辑结构设计中，我们需要把中间表作为一个实体，放入到各个实体的关系中。

因此，可以通过 PDM（物理数据模型）的设计，来对数据库中存在的关系进行展示。当然我们需要借助相关的工具去完成它。本系统数据库物理模型设计是使用的 Navicat premium 工具设计的物理数据模型，如果模型设计完整，可以通过物理数据模型直接导出数据库结构。

结合 E-R 图的描述，总结出如下图 4.12 物理模型图：

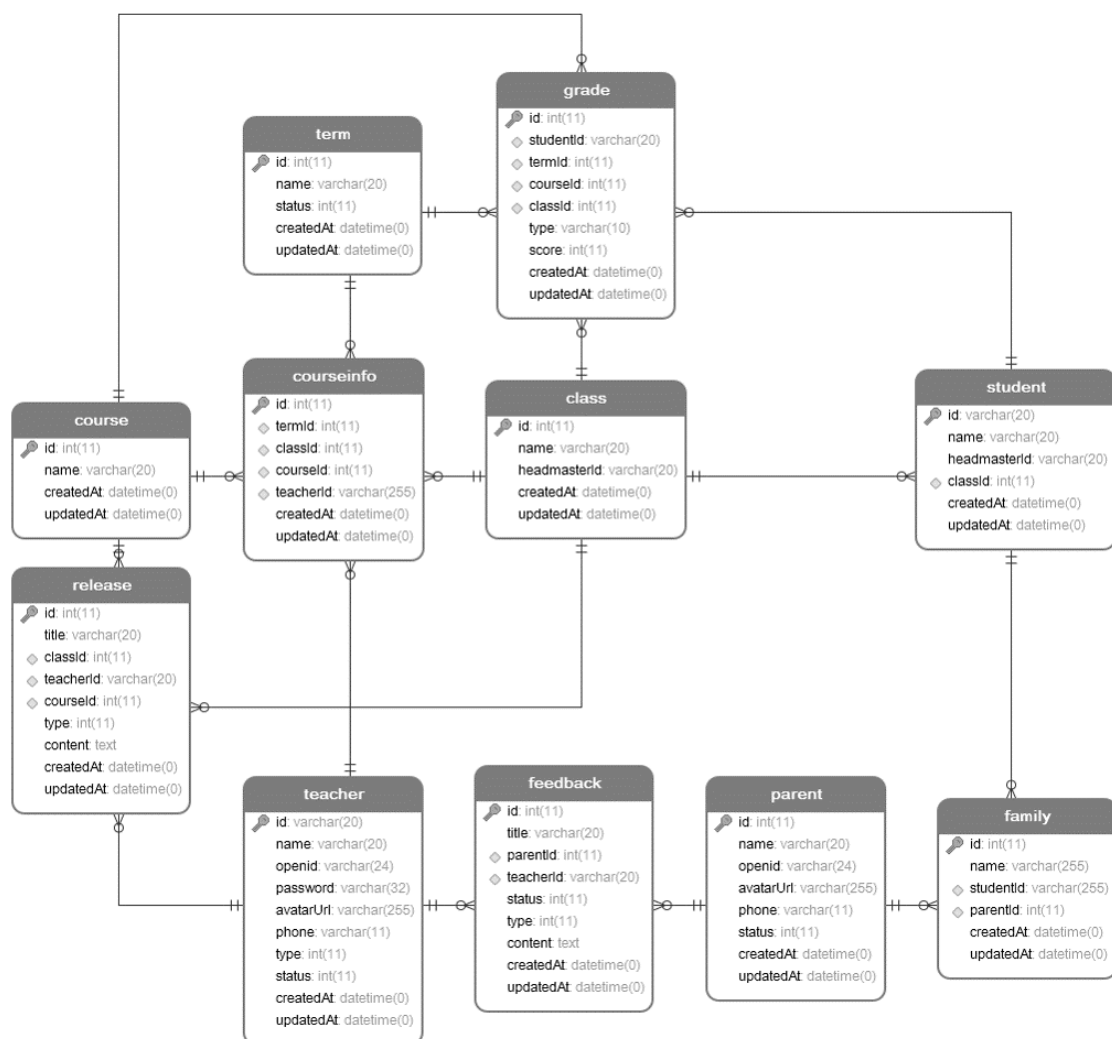


图 4.12 物理模型图

### 4.5.3 物理结构设计

物理模型图的设计设计完成，就代表着可以将各个数据表完整的设计和表示出来。

虽然物理模型图已经将表和关系表示出来，但是表中的每一字段的约束条件并没

有定义，如字段是否为空，字段是否唯一，字段是否含有默认值等等，这些更具体的描述，还需要通过表格一一列举出来。

通过物理模型图，经过整理，一共设计出如下 11 张表。

班级表用于存取学生班级相关信息，具体如表 4.1 所示：

表 4.1 班级表

| 表名           | class（班级表） |             |     |    |
|--------------|------------|-------------|-----|----|
| 列名（字段）       | 字段含义       | 数据类型        | 不为空 | 备注 |
| id           | ID         | int(11)     | ✓   | 主键 |
| name         | 班级名称       | varchar(20) | ✓   |    |
| headmasterId | 班主任 ID     | varchar(20) | ✓   |    |
| createdAt    | 创建时间       | datetime    | ✓   |    |
| updatedAt    | 更新时间       | datetime    | ✓   |    |

课程表用于存取学生每学期要写的课程和老师每学期教师要教的课程的信息，具体如表 4.2 所示：

表 4.2 课程表

| 表名        | course（课程表） |             |     |    |
|-----------|-------------|-------------|-----|----|
| 列名（字段）    | 字段含义        | 数据类型        | 不为空 | 备注 |
| id        | ID          | int(11)     | ✓   | 主键 |
| name      | 课程名称        | varchar(20) | ✓   |    |
| createdAt | 创建时间        | datetime    | ✓   |    |
| updatedAt | 更新时间        | datetime    | ✓   |    |

课程信息表是用于解决课程、班级和教师之间多对多关系的中间表，具体如表 4.3 所示：

表 4.3 课程信息表

| 表名        | courseInfo（课程信息表） |             |     |             |
|-----------|-------------------|-------------|-----|-------------|
| 列名（字段）    | 字段含义              | 数据类型        | 不为空 | 备注          |
| id        | ID                | int(11)     | ✓   | 主键          |
| termId    | 学期 ID             | int(11)     | ✓   | 外键->学期表中 id |
| classId   | 班级 ID             | int(11)     | ✓   | 外键->班级表中 id |
| courseId  | 课程 ID             | int(11)     | ✓   | 外键->课程表中 id |
| teacherId | 教师 ID             | varchar(20) | ✓   | 外键->教师表中 id |
| createdAt | 创建时间              | datetime    | ✓   |             |
| updatedAt | 更新时间              | datetime    | ✓   |             |

家庭表是由于学生和家長多对多关系，而创建的中间表，具体内容如表 4.4 所示：

表 4.4 家庭表

| 表名        | family（家庭表） |          |     |             |
|-----------|-------------|----------|-----|-------------|
| 列名（字段）    | 字段含义        | 数据类型     | 不为空 | 备注          |
| id        | ID          | int(11)  | ✓   | 主键          |
| studentId | 学生 ID       | int(11)  | ✓   | 外键->学生表中 id |
| parentId  | 家长 ID       | int(11)  | ✓   | 外键->家长表中 id |
| createdAt | 创建时间        | datetime | ✓   |             |
| updatedAt | 更新时间        | datetime | ✓   |             |

学期表是用于存储学校每学期的信息的表，具体如表 4.5 所示：

表 4.5 学期表

| 表名        | term（学期表） |             |     |                             |
|-----------|-----------|-------------|-----|-----------------------------|
| 列名（字段）    | 字段含义      | 数据类型        | 不为空 | 备注                          |
| id        | ID        | int(11)     | ✓   | 主键                          |
| name      | 学期名称      | varchar(20) | ✓   |                             |
| status    | 学期状态      | int(11)     |     | 默认值为 0<br>1 当前学期<br>0 非当前学期 |
| createdAt | 创建时间      | datetime    | ✓   |                             |
| updatedAt | 更新时间      | datetime    | ✓   |                             |

反馈表是用于存储家长对教师进行反馈的信息，具体如表 4.6 所示：

表 4.6 反馈表

| 表名        | feedback（反馈表） |             |     |                                      |
|-----------|---------------|-------------|-----|--------------------------------------|
| 列名（字段）    | 字段含义          | 数据类型        | 不为空 | 备注                                   |
| id        | ID            | int(11)     | ✓   | 主键                                   |
| title     | 反馈标题          | varchar(20) | ✓   |                                      |
| teacherId | 教师 ID         | varchar(20) | ✓   | 外键->教师表中 id                          |
| parentId  | 家长 ID         | int(11)     | ✓   | 外键->家长表中 id                          |
| status    | 反馈状态          | int(11)     | ✓   | 默认值 2<br>0 已解决<br>1 已读<br>2 未读       |
| type      | 反馈类型          | int(11)     | ✓   | 0 学生<br>1 课程<br>2 教程<br>3 作业<br>4 统计 |
| content   | 反馈内容          | text        | ✓   |                                      |
| createdAt | 创建时间          | datetime    | ✓   |                                      |
| updatedAt | 更新时间          | datetime    | ✓   |                                      |

分数表是用于存储学生成绩信息的表，具体如表 4.7 所示：

表 4.7 分数表

| 表名        | grade（分数表） |             |     |            |
|-----------|------------|-------------|-----|------------|
| 列名（字段）    | 字段含义       | 数据类型        | 不为空 | 备注         |
| id        | ID         | int(11)     | ✓   | 主键         |
| studentId | 学生 ID      | varchar(20) | ✓   | 外键→学生表中 id |
| termId    | 学期 ID      | int(11)     | ✓   | 外键→学期表中 id |
| courseId  | 课程 ID      | int(11)     | ✓   | 外键→课程表中 id |
| classId   | 班级 ID      | int(11)     | ✓   | 外键→班级表中 id |
| type      | 考试类型/场次    | varchar(10) | ✓   |            |
| score     | 分数         | int(11)     | ✓   |            |
| createdAt | 创建时间       | datetime    | ✓   |            |
| updatedAt | 更新时间       | datetime    | ✓   |            |

家长表是用于存取学生家长信息的表，具体如表 4.8 所示：

表 4.8 家长表

| 表名        | parent（家长表） |              |     |                        |
|-----------|-------------|--------------|-----|------------------------|
| 列名（字段）    | 字段含义        | 数据类型         | 不为空 | 备注                     |
| id        | ID          | int(11)      | ✓   | 主键                     |
| name      | 家长姓名        | varchar(20)  | ✓   |                        |
| openId    | 微信 openID   | varchar(24)  |     |                        |
| avatarUrl | 微信头像地址      | varchar(255) |     |                        |
| phone     | 手机号码        | varchar(11)  |     |                        |
| status    | 绑定状态        | int(11)      | ✓   | 默认值 0<br>1 绑定<br>0 未绑定 |
| createdAt | 创建时间        | datetime     | ✓   |                        |
| updatedAt | 更新时间        | datetime     | ✓   |                        |

公告表是用于存取教师向家长发布相关类型公告信息的表，具体如表 4.9 所示：

表 4.9 公告表

| 表名        | release（公告表） |             |     |                      |
|-----------|--------------|-------------|-----|----------------------|
| 列名（字段）    | 字段含义         | 数据类型        | 不为空 | 备注                   |
| id        | ID           | int(11)     | ✓   | 主键                   |
| title     | 公告标题         | varchar(20) | ✓   |                      |
| classId   | 班级 ID        | int(11)     | ✓   | 外键->班级表中 id          |
| teacherId | 教师 ID        | varchar(20) | ✓   | 外键->教师表中 id          |
| courseId  | 课程 ID        | int(11)     | ✓   | 外键->课程表中 id          |
| type      | 公告类型         | int(11)     | ✓   | 0 公告<br>1 作业<br>2 统计 |
| content   | 公告内容         | text        | ✓   |                      |
| createdAt | 创建时间         | datetime    | ✓   |                      |
| updatedAt | 更新时间         | datetime    | ✓   |                      |

学生表是用于存取学生基本信息的表，具体如表 4.10 所示：

表 4.10 学生表

| 表名           | student（学生表） |             |     |             |
|--------------|--------------|-------------|-----|-------------|
| 列名（字段）       | 字段含义         | 数据类型        | 不为空 | 备注          |
| id           | 学号           | varchar(20) | ✓   | 主键          |
| name         | 学生姓名         | varchar(20) | ✓   |             |
| classId      | 班级 ID        | int(11)     | ✓   | 外键->班级表中 id |
| headmasterId | 班主任 ID       | varchar(20) | ✓   |             |
| createdAt    | 创建时间         | datetime    | ✓   |             |
| updatedAt    | 更新时间         | datetime    | ✓   |             |

教师表是用于存取教师相关基本信息的表，具体如表 4.11 所示：

表 4.11 教师表

| 表名        | teacher（教师表） |              |     |                               |
|-----------|--------------|--------------|-----|-------------------------------|
| 列名（字段）    | 字段含义         | 数据类型         | 不为空 | 备注                            |
| id        | ID           | varchar(20)  | ✓   | 主键                            |
| name      | 教师姓名         | varchar(20)  | ✓   |                               |
| password  | 登录密码         | varchar(32)  | ✓   | 默认为 123456，通过 md5 加密          |
| openid    | 微信 openID    | varchar(24)  |     |                               |
| avatarUrl | 微信头像地址       | varchar(255) |     |                               |
| phone     | 手机号码         | varchar(11)  |     |                               |
| type      | 教师类型         | int(11)      | ✓   | 0 任课教师<br>1 班主任<br>2 任课老师兼班主任 |
| status    | 绑定状态         | int(11)      | ✓   | 默认值 0<br>1 绑定<br>0 未绑定        |
| createdAt | 创建时间         | datetime     | ✓   |                               |
| updatedAt | 更新时间         | datetime     | ✓   |                               |

综上所述的各个表的字段内容，我们得到数据库每个表的详细信息，但这仅仅只是一个分散的表信息。因此，我们还需要把各个表之间的关系梳理清楚，每个约束条件依赖那些东西，每个外键对应哪一个表。

## 4.6 界面设计

本系统中由于后台管理系统结构简单，以及现有的成熟 UI 框架，界面设计只有简单的结构草图，而重点的界面设计则放在了微信小程序的 UI 上。

根据需求设计和系统相关设计，通过 MockingBot（墨刀）原型设计工具，为微信



小程序设计了一套交互式的 UI 原型。

关于微信小程序的设计中，主要以深绿色和橙黄色为主导颜色，把简约的风格作为设计的理念，部分原型效果如下图所示：



图 4.13 微信小程序部分原型图

### 4.7 本章小结

通过上章节的需求分析，本章将需求进行了更加详细的设计，通过用例图、系统功能概要图、物理数据模型图等，将系统的细节更加清晰，更加具体的呈现给用户们和开发者们。



## 5 系统实现

根据系统的相关分析与设计，我们就可以搭建开发环境，并针对具体的功能进行系统开发了。

### 5.1 开发环境

本平台开发使用 HbuilderX、Visual Studio Code、WebStorm 三个编辑器共同开发的，环境版本是 node-v12, 数据库使用 MySQL 8 版本，服务器使用本机 node 环境。

### 5.2 成绩导入编码实现

#### 5.2.1 界面原型

成绩导入主要是通过填写好的学生成绩表格文件，将它快速导入后台管理系统的成绩管理列表中，需要填入的信息有学号、学期、课程、班级、场次（类型）和分数。成绩导入的原型如图 5.1 所示：

| <input type="checkbox"/> | 学号   | 姓名  | 班级    | 学期           | 课程 | 场次   | 分数  | 操作                                    |
|--------------------------|------|-----|-------|--------------|----|------|-----|---------------------------------------|
| <input type="checkbox"/> | 1001 | 陈芳洁 | 1000班 | 2019-2020下学期 | 语文 | 期末测试 | 100 | <a href="#">编辑</a> <a href="#">删除</a> |
| <input type="checkbox"/> | 1001 | 陈芳洁 | 1000班 | 2019-2020下学期 | 数学 | 期末测试 | 100 | <a href="#">编辑</a> <a href="#">删除</a> |
| <input type="checkbox"/> | 1001 | 陈芳洁 | 1000班 | 2019-2020下学期 | 英语 | 期末测试 | 100 | <a href="#">编辑</a> <a href="#">删除</a> |
| <input type="checkbox"/> | 1002 | 陈祺然 | 1000班 | 2019-2020下学期 | 语文 | 期末测试 | 100 | <a href="#">编辑</a> <a href="#">删除</a> |

图 5.1 成绩导入原型图

#### 5.2.2 实现原理

当用户将学生成绩表格文件完成后，点击“导入”按钮导入表格文件后，前端解析表格内容，将它转换和格式化为 js 对象。再发送批量添加成绩请求，服务端接受请求，进行相关处理，将数据添加到数据库。实现后，上传成绩和上传成绩完成效果

如下图 5.2 和图 5.3 所示：

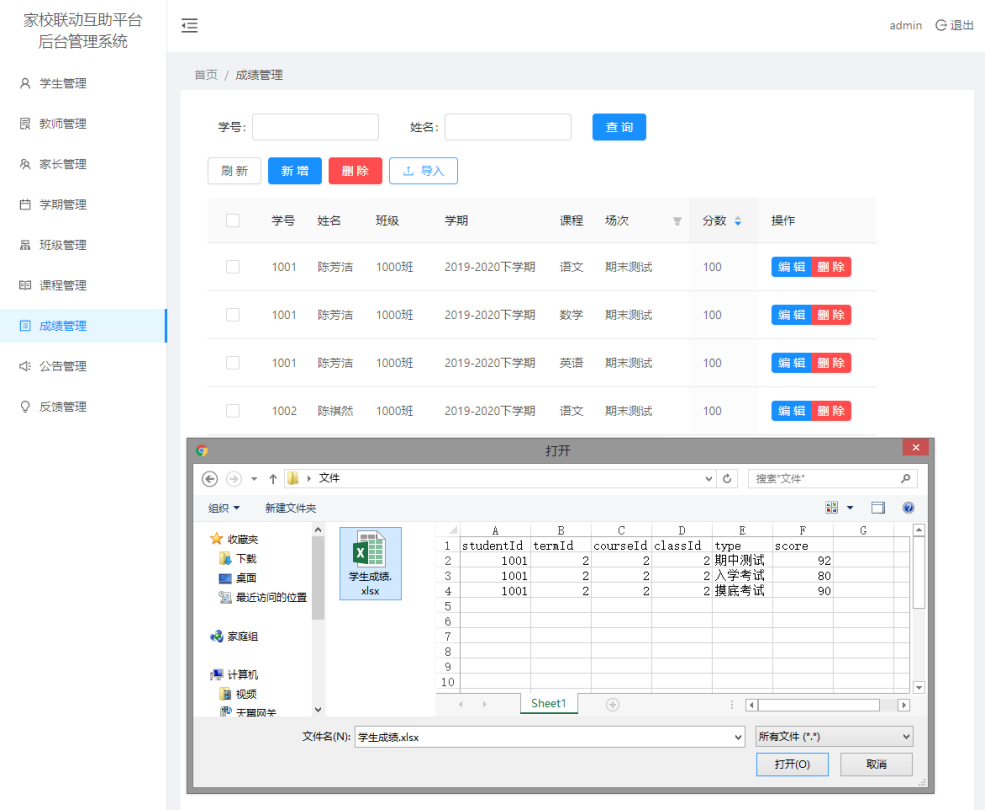


图 5.2 成绩导入选择文件效果图

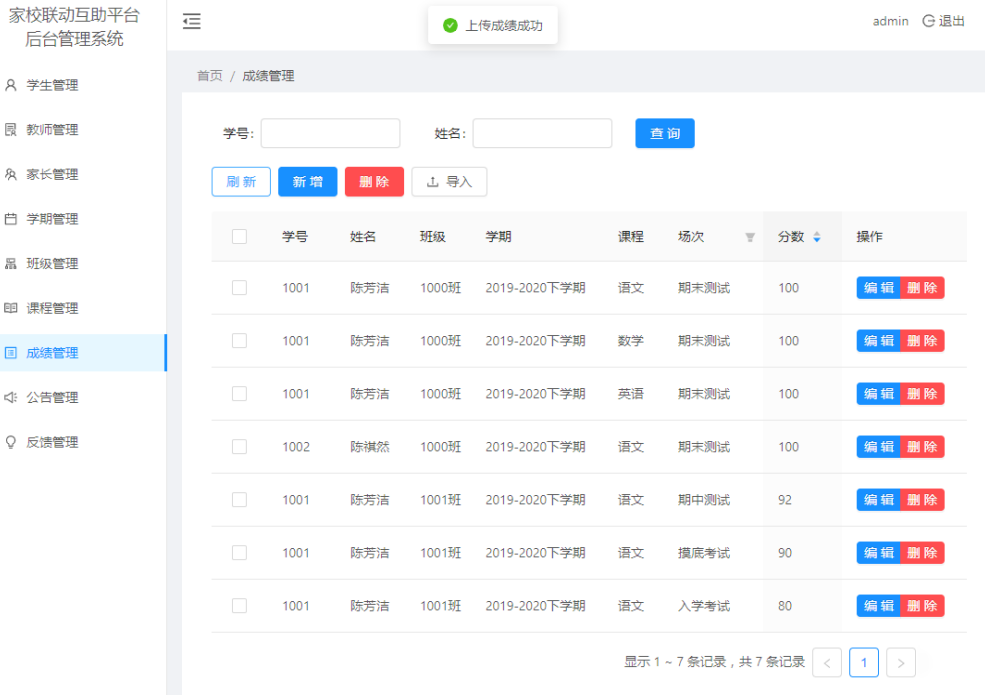


图 5.3 成绩导入完成效果图

### 5.2.3 关键代码

前端处理成绩导入代码：

```
// 成绩上传文件处理

import readExcel from 'read-excel-file'

async handleUpload (file) {
  await readExcel(file).then(rows => {
    rows = rows.slice(1)
    rows = rows.map(g => { return { studentId: g[0], termId: g[1], courseId:
g[2], classId: g[3], type: g[4], score: g[5] } })
    this.uploadData = rows
  })
}
```

后端处理批量添加成绩代码：

```
// 上传成绩

@Post('/gradeUpload')

public async upload(@Ctx ctx: Context) {
  const grades = ctx.request.body.rows
  let res = (await ctx.state.db['grade'].bulkCreate(grades))
  ctx.body = { code: 0, data: res, message: res ? '成功' : '出错' }
}
```

## 5.3 发布公告编码实现

### 5.3.1 界面原型

当教师有公告、作业或者统计需要告知家长时，教师就要登录微信小程序发布相应类型的公告，教师需要提交公告标题、发布类型、班级、课程和公告内容相关信息。发布之后通过首页查看发布的公告。原型如图 5.4 所示：

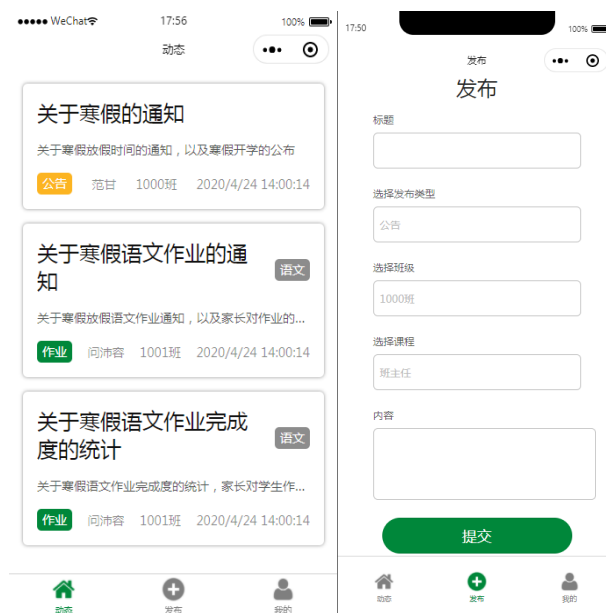


图 5.4 首页公告和发布公告原型图

### 5.3.2 实现原理

当教师有新的公告要发布时，填写公告表单提交之后，向后台发送请求，后台向数据库添加公告，点击首页向后台请求最新的公告列表。效果图如图 5.5 所示：



图 5.5 填写发布公告表单（左）和首页公告更新后列表（右）效果图

### 5.3.3 关键代码

发布公告后端代码：

// 增加或更新

```
@Post('/releaseNew')
```

```
public async upsert(@Ctx ctx: Context) {
    const release = ctx.request.body
    delete release.name
    let res = (await ctx.state.db['release'].upsert({ ...release }))
    ctx.body = { code: 0, data: res, message: res ? '成功' : '出错' }
}
```

## 5.4 家长绑定编码实现

### 5.4.1 界面原型

当家长第一次使用微信小程序查看学生学习情况的时候，需要填写姓名、手机号和要绑定的孩子的学号，然后通过老师审核之后，就可以正常使用了。原型图如 5.6 所示：



图 5.6 家长绑定流程原型图

### 5.4.2 实现原理

家长进入微信小程序，获得表单，填写个人基本信息，点击“下一步”后，向后台发送添加家长请求，保存家长的微信 openId 等重要信息。成功之后输入绑定孩子的学号，点击“绑定”之后，向后台发送添加孩子请求，完成绑定。效果图如图 5.7 所示：

图 5.7 家长绑定填写表单效果图

### 5.4.3 关键代码

添加家长信息代码：

// 家长信息增加或更新

@Post('/parentNew')

```
public async upsert(@Ctx ctx: Context) {
    const par = ctx.request.body
    let res = (await ctx.state.db['parent'].upsert({ ...par }))
    ctx.body = { code: 0, data: res, message: res ? '成功' : '出错' }
}
```

添加和绑定孩子代码：



```
// 家长孩子增加或更新
@Post('/familyNew')
public async addChild(@Ctx ctx: Context) {
    const family = ctx.request.body.rows
    let res = (await ctx.state.db['family'].bulkCreate(family))
    ctx.body = { code: 0, data: res, message: res ? '成功' : '出错' }
}
```

## 5.5 教师查看反馈家长反馈编码实现

### 5.5.1 界面原型

当教师抽出自己空余的时间时，可以登录微信小程序，通过反馈管理查看家长对他的相关反馈。接受反馈信息后，针对反馈的内容和家长做相关沟通和交流。原型如图 5.8 所示：



图 5.8 微信小程序教师反馈管理家长反馈列表

### 5.5.2 实现原理

教师通过底部导航栏，选择“我的”菜单项，再点击“反馈管理”按钮，进入家长反馈列表。进入页面前，微信端会发生请求获取家长反馈列表，然后返回给微信端进行渲染。效果图如如 5.9 所示：



图 5.9 微信小程序“我的”页面效果图

### 5.5.3 关键代码

获取家长反馈列表代码：

```
@Get('/feedback')
public async getList(@Ctx ctx: Context) {
    const { like } = sequelize.Op; const title = ctx.query.title
    const teacherId = ctx.query.teacherId;
    const parentId = ctx.query.parentId
    let feedbackList = (await ctx.state.db['feedback'].findAll({
        include: [
            { model: ctx.state.db['parent'], attributes: ['name'] },
            { model: ctx.state.db['teacher'], attributes: ['name'] }],
        attributes: {exclude: ['createdAt'] },
        where: {title: { [like]: '%' + (title ? title : '') + '%'} },
```

```

        teacherId: { [like]: '%' + (teacherId ? teacherId : '') + '%' },
        parentId: { [like]: '%' + (parentId ? parentId : '') + '%' }}
    })).map((entry: any) => entry.toJSON())

feedbackList = feedbackList.reduce((memo: any[], cur: any, index:
    number) => { memo[index] = { ...cur, parent: cur.parent.name, teacher:
        cur.teacher.name }; return memo }, [])

ctx.body = { code: 0, data: feedbackList, message: '获取成功' }
}

```

## 5.6 管理员添加班级编码实现

### 5.6.1 界面原型

当学校新增加了班级，管理员就需要通过后台管理系统，将新的班级信息添加到系统，好让学生、老师和课程在后续操作中能够操作完整。管理员需要填写班级名称、班级课程、班级教师和班主任相关信息。原型如图 5.10 所示：

图 5.10 新增班级原型图

### 5.6.2 实现原理

管理员通过登录后台管理系统，选择菜单“班级管理”选项进入班级列表，进入之前，会发送获取班级列表请求。获取到班级列表之后，点击“新增”按钮，弹出添加班级表单页，填写完整信息后，发送添加班级请求。添加成功之后，班级列表会更新出新增班级的信息。效果图如图 5.11 和图 5.12 所示：

新增

\*

名称:

1004班

\*

课程:

语文 × 数学 × 英语 ×

\*

教师:

问沛容 × 范甘 × 白文翰 ×

\*

班主任:

范甘

取消

确认

图 5.11 填写新增班级表单信息效果图

成功

首页 / 班级管理

名称:

查询

刷新

新增

删除

| <input type="checkbox"/> | 名称    | 课程       | 教师                 |
|--------------------------|-------|----------|--------------------|
| <input type="checkbox"/> | 1000班 | 语文 数学 英语 | 问沛容 范甘 白文翰         |
| <input type="checkbox"/> | 1001班 | 语文 数学 英语 | 范甘 问沛容 白文翰         |
| <input type="checkbox"/> | 1002班 | 语文 数学 英语 | 白文翰 范甘 喻皓昱 赖忆雪 周康安 |
| <input type="checkbox"/> | 1003班 | 语文 数学 英语 | 喻皓昱 赖忆雪 周康安        |
| <input type="checkbox"/> | 1004班 | 语文 数学 英语 | 问沛容 范甘             |

图 5.12 新增班级成功效果图

5.6.3 关键代码

获取班级列表代码:

```
@Get('/classList')
public async classList(@Ctx ctx: Context) {
    let classList = (await ctx.state.db['class'].findA11({
        attributes: ['id', 'name'] })).map((entry: any) => entry.toJSON())
```

```

    ctx.body = { code: 0, data: classList, message: '获取成功' }
  }
}

新增班级代码:

@Post('/classNew')
public async upsert(@Ctx ctx: Context) {
  const { eq } = sequelize.Op; const classes = ctx.request.body
  let termId = (await ctx.state.db['term'].findOne({ attributes: ['id'],
    where: { status: { [eq]: 1 } } })).toJSON().id; let res = false;
  if (await ctx.state.db['class'].upsert({ id: classes.id, name:
    classes.name, headmasterId: classes.headmasterId })) {
    try { let classId = (await ctx.state.db['class'].findOne({
      attributes: ['id'], where: { name: { [eq]: classes.name },
      headmasterId: { [eq]: classes.headmasterId } } })).toJSON().id
      (classes.teacherIds + '').split(',').forEach((cls: any) => {
        (classes.courseIds + '').split(',').forEach(async (c: any) => {
          await ctx.state.db['courseInfo'].upsert({ teacherId: cls,
            courseId: c, termId, classId }) }) }); res = true
        } catch (err) { console.log(err)
          ctx.state.db['class'].destroy({
            where: {name: { [eq]: classes.name }, headmasterId: {
              [eq]: classes.headmasterId } } }); res = false } }
          ctx.body = { code: 0, data: res, message: res ? '成功' : '出错' }
        }
      }
    }
  }
}

```

## 5.7 本章小结

本章展示了系统实现的开发环境，以及管理员成绩导入、教室发布公告并查看和家长绑定等相关的界面原型、实现原理以及关键代码的描述。



## 6 系统测试

测试在软件工程中具有一个重要的地位，在开发中，无时无刻都有测试，见到一段代码的测试，大到一个功能甚至系统的测试。测试的目的就在于在有限的资源里，找到影响用户需求的错误 bug，从而修复它，提高系统的质量。

按照测试的过程，通常测试会分为单元测试、集成测试、系统测试和验收测试。不同的测试都在不同的环节中，由于本系统相对简单，以及单元测试在开发中，也做了简单的测试，本章就只针对系统的能够进行对应的测试。

### 6.1 接口测试

由于系统开发采用前后端分离的概念，因此前后端分离中，接口就是连接客户端和服务端的桥梁，所以至关重要不可忽略。

经过时间，本次接口测试采用的工具是 Postman。Postman 这个工具功能比较强大，可以发送几乎所有 HTTP 请求，并可直接使用谷歌浏览器插件无需下载。这一工具，可以大大提高接口测试效率，就不再需要，写请求代码请求来测试接口问题。并且还可以保存你请求过的请求数据以及 mock 测试数据，方便下次测试，无需再输入相应信息，减少无效劳动。下面表 6.1 和表 6.2 则是通过 postman 对接口测试情况的统计：

表 6.1 后台管理系统接口测试

| 子系统    | 接口说明          | 预期结果 | 测试结果 | 备注 |
|--------|---------------|------|------|----|
| 后台管理系统 | 学生列表/查询接口     | 符合要求 | 符合要求 |    |
|        | 学生新增/编辑接口     | 符合要求 | 符合要求 |    |
|        | 学生批量删除/单个删除接口 | 符合要求 | 符合要求 |    |
|        | 教师列表/查询接口     | 符合要求 | 符合要求 |    |
|        | 教师新增/编辑接口     | 符合要求 | 符合要求 |    |
|        | 教师批量删除/单个删除接口 | 符合要求 | 符合要求 |    |
|        | 家长列表/查询接口     | 符合要求 | 符合要求 |    |

表 6.1 后台管理系统接口测试（续上表）

| 子系统    | 接口说明          | 预期结果 | 测试结果 | 备注 |
|--------|---------------|------|------|----|
| 后台管理系统 | 家长新增/编辑接口     | 符合要求 | 符合要求 |    |
|        | 家长批量删除/单个删除接口 | 符合要求 | 符合要求 |    |
|        | 学期列表/查询接口     | 符合要求 | 符合要求 |    |
|        | 学期新增/编辑接口     | 符合要求 | 符合要求 |    |
|        | 学期批量删除/单个删除接口 | 符合要求 | 符合要求 |    |
|        | 班级列表/查询接口     | 符合要求 | 符合要求 |    |
|        | 班级新增/编辑接口     | 符合要求 | 符合要求 |    |
|        | 班级批量删除/单个删除接口 | 符合要求 | 符合要求 |    |
|        | 课程列表/查询接口     | 符合要求 | 符合要求 |    |
|        | 课程新增/编辑接口     | 符合要求 | 符合要求 |    |
|        | 课程批量删除/单个删除接口 | 符合要求 | 符合要求 |    |
|        | 成绩列表/查询接口     | 符合要求 | 符合要求 |    |
|        | 成绩新增/编辑接口     | 符合要求 | 符合要求 |    |
|        | 成绩批量删除/单个删除接口 | 符合要求 | 符合要求 |    |
|        | 公告列表/查询接口     | 符合要求 | 符合要求 |    |
|        | 公告新增/编辑接口     | 符合要求 | 符合要求 |    |
|        | 公告批量删除/单个删除接口 | 符合要求 | 符合要求 |    |
|        | 反馈列表/查询接口     | 符合要求 | 符合要求 |    |
|        | 反馈批量删除/单个删除接口 | 符合要求 | 符合要求 |    |



表 6.2 微信小程序接口测试

| 子系统   | 接口说明           | 预期结果 | 测试结果 | 备注                         |
|-------|----------------|------|------|----------------------------|
| 微信小程序 | 公告查询           | 符合要求 | 符合要求 | 老师/家长，同后台公告接口              |
|       | 发布反馈/反馈信息状态修改  | 符合要求 | 符合要求 | 家长/教师（状态修改）                |
|       | 反馈的删除/反馈信息列表   | 符合要求 | 符合要求 | 家长，同后台管理反馈接口               |
|       | 家长基本信息         | 符合要求 | 符合要求 | 家长                         |
|       | 家长孩子信息列表       | 符合要求 | 符合要求 | 家长                         |
|       | 家长孩子成绩列表       | 符合要求 | 符合要求 | 家长                         |
|       | 孩子课程信息列表       | 符合要求 | 符合要求 | 家长                         |
|       | 发布公告/公告列表/公告删除 | 符合要求 | 符合要求 | 教师，同后台管理公告接口（包含小程序公告、作业管理） |
|       | 课程信息列表         | 符合要求 | 符合要求 | 教师                         |
|       | 班级成绩列表         | 符合要求 | 符合要求 | 教师                         |
|       | 学生家长列表         | 符合要求 | 符合要求 | 教师                         |

## 6.2 主要功能测试

一个系统开发完成之后，最重要的就是每一个功能能够真正可用，真正可以在实际环境中运行，功能测试就是为系统的每一个功能能够真正运行，能够完成业务流程而产生的。

一定程度上将，功能测试可分为基本功能验证、数据类型测试和探索性测试。通俗来说，基本功能验证，就是验证每一个功能是否可用；数据类型测试，数据处理是否正确，例如，手机号码，密码，以及个数的极值边界等；最后，探索性测试就是通过一些特殊场景下功能是否能够正常使用，请求突然中断，请求快速重复等极端场景或突发场景。

本次功能测试，只针对相对数据处理比较多的功能进行了功能测试，测试的情况则如表 6.3 所示：

表 6.3 主要功能测试

| 子系统    | 用例          | 功能说明                | 测试结果 | 备注              |
|--------|-------------|---------------------|------|-----------------|
| 后台管理系统 | 教师的添加/修改    | 教师基本信息的处理           | 符合要求 |                 |
|        | 家长的添加/修改    | 家长基本信息的处理           | 符合要求 |                 |
|        | 班级的添加/修改    | 班级中老师、课程、班主任分配的处理   | 符合要求 |                 |
|        | 成绩的添加/导入/修改 | 成绩信息的处理             | 符合要求 |                 |
| 微信小程序  | 公告发布        | 作业、公告、统计的发布         | 符合要求 | 教师              |
|        | 公告管理        | 查看和删除公告             | 符合要求 | 教师              |
|        | 发布反馈        | 发布教师、课程、学生、作业、统计的反馈 | 符合要求 | 家长              |
|        | 反馈管理        | 反馈的状态修改/删除          | 符合要求 | 家长/教师<br>(状态修改) |

### 6.3 本章小结

本章对家校助学联动平台系统进行了接口测试和系统测试，并对其进行了统计和描述。但在大批量的数据处理还是有一定的缺陷，需要去完善和处理。从整体情况上来看，各项测试都符合用户需求。

## 工作总结与展望

### 总结

通过本次毕业设计的研究，系统完成了需求大部分功能：管理员的学生管理、教师管理、家长管理、学期管理、班级管理、课程管理、成绩管理以及公告和反馈管理；但本次系统设计的核心在于提供给家长和老师的微信小程序设置，家长可以通过小程序查看老师信息和学生信息，以及向老师反馈课程、老师、统计等相关的问题，老师则通过反馈去解决问题，极大减少了在班级群中接收反馈的问题。其中，老师也可以通过微信小程序查看学生和它班级的学习情况，查看和处理学生家长的反馈，以及根据需求发布作业、公告和统计。

当然，本次毕业设计的创新点就是微信小程序，他是一个成熟的系统，所以安全性也会很高，获取用户唯一标识 `openid` 需要通过对称解密算法 `AES-128-CBC` 来对获取的用户信息中敏感数据 `encryptedData` 和 `iv` 加密算法初始向量结合来进行获取。后台管理系统的登录密码也采用 `md5` 加密算法，使密码相对不可解密来增加安全性。

### 展望

本系统设计相对简单，还需要分阶段完善更多的问题。由于研究时长限制和系统角色关系相对简单的原因，本系统未采用安全性更高的 `JWT` 认证和权限管理。因此安全性还有待提高。

在前后端分离设计中，由于框架研究不够深入的原因，接口请求只简单采用了传统的 `GET` 和 `POST` 请求，并未采用 `RESTful` 设计风格，来对接口进行请求进行设计，这会加大后期重构或者维护的时候的难度。

除了以上问题以外，还有一些很多细节问题需要完善，任何系统都不是一次就能设计的完美，只有在不断的完善中变得更好。没有最好，只有更好。技术在不断革新，思想在不断进步。软件工程的每一个环节都是一个产品、一个系统必不可少的部分，而每一个部分都会有它不同的优化方法，在本系统中，测试其实也是一个在开发中重要的环节，但本次毕业设计，并未采用相关框架，或者自己编写代码进行单元测试，

这也是一个需要完善的地方。

“君子生非异也，善假于物也”，让系统更完美，就需要学会借助工具，学会站在巨人的肩膀上去不断完善系统的问题。很多东西不需要深入去研究，要学会把中心放在真正需要用到的地方。每一个框架的目的就是增加效率，把重心放在重点上，让一个核心的东西更加友好，更加通俗易懂。

本次毕业设计收获颇多，受益匪浅，希望在以后的发展中能够百尺竿头更进一步。

## 致 谢

时光荏苒，论文也在悄然无声中来到了尾声。这也预示着，论文的完成也就是大学生活即将结束的标志。风风雨雨中，所有的成果都少不了这一路上陪伴着我一起走过的那些人。感恩过去的你们，感恩现在的成果，未来我也将不负众望。

当然，在无数日的探索与研究中，要非常感谢我毕业设计指导老师——孟高原老师，一直以来对我研究的关心、支持和帮助。在他精心的时间安排，研究资料的提供，以及问题的解惑等等一系列指导里，让我在论文设计探索的道路上，避免了很多绊脚石，同时也让我在各个方面的研究能够更加顺利的进行。

再者，也不能忘记感激这些关于本论文的相关文献的作者们，对家庭与学校之间的关系和沟通上的研究做出的贡献，以及本平台使用的技术框架的开发者们。这些东西，毫无疑问的给了我更多的探索灵感和研究思路。让我对本论文里，家校助学联动平台有了更深刻的认知和了解。也希望我的研究能够在家校助学联动的发展中做出一点微小的贡献。

最后，也衷心感谢能够在忙碌的工作当中，抽出宝贵时间的答辩评阅老师们！



## 参考文献

- [1] Yafei Wang, Yumeng Zhao. A Survey on the Application of WeChat Platform on Home-school Cooperation[J]. School of Humanities and Law North China University of Technology Beijing, China. Advances in Social Science, Education and Humanities Research, volume 357, 2nd International Conference on Education, Economics and Social Science (ICEESS 2019), 2019:314.
- [2] 邱睿, 孟琦. 基于微信小程序的家校通系统开发与实现[D]. 上海: 上海师范大学, 2019:5.
- [3] 杨青. 社交平台在家校沟通中应发挥更积极作用 [J]. 中国教育学刊, 2018(1):102.
- [4] 傅钟, 雷笑天, 王强, 张健. 家校微信小程序的设计和开发 [J]. 微处理机, 2018(6):32
- [5] 徐永慧. 实用型小学家校通平台的开发与应用 [J]. 信息与电脑, 2017(14):87-88
- [6] Vasyl Sukhomlynsky, Du Diankun translate. Suggestions for teachers. Educational Science Publishing House [M], 1984.
- [7] Ren Xuemei. A study on home-school cooperation of elementary schools based on WeChat platform[D]. Sichuan Normal University. 2015.
- [8] Li Na. “On the enlightenment of US’ s home-school cooperation to China’ s education. Asian-Pacific Education. 2016(13):290.
- [9] Wang Yanling. “A new form of home-school cooperation—parents as teaching assistants”, international and comparative education. 2001(7):56.
- [10] Lin Xiaofen. A study on home-school cooperation in primary schools based on network communicative tools [D]. Shanghai Normal University. 2017.
- [11] 梁云真, 李俊. “家校通” 相关研究述评[J]. 河南: 软件导刊(教育技术), 2011:34

