****

**软 件 学 院**

**实践成果四报告**

**课题名称：基于微信小程序的中工软考研帮（V1.0）**

**专业： 软件工程**

**班级： 卓越161**

**学号： 201619140220**

**学生姓名： 崔烁豪**

**指导老师： 韩玉民**

2019年7月4日

**中原工学院软件学院**

**实践成果四课题任务书**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名** | 崔烁豪 | **班级** | 卓越161班 |
| **题 目** | 基于微信小程序的中工软考研帮 | | |
| **设**  **计**  **任**  **务** | 为了帮助我校打算考研的同学能对学长学姐的考研院校专业选择结果和初试成绩有一个直观的了解，研发一个基于微信小程序的历届考研成绩等信息展示平台−中工软考研帮。主要功能包括：  1.报考信息展示  主要包括报考院校、报考专业、研究方向、考研年份等。  2.学生基本信息展示  展示报考学生的学号、姓名、性别、所在院校、所在学院、所在班级等。  3.考研成绩展示  主要包括初试总分、我院考生排名、单科成绩等。。  4.学生详细信息展示  主要包括学生的电话、QQ、微信等联系方式和博客、GitHub等技术博客网址。  开发工具：微信开发者工具、手机微信。 | | |
| **时**  **间**  **进**  **度** | 第4周-第5周（2019-03-18~2019-03-31）：对需求进行深入分析，并设计出适合微信小程序端的页面与数据库，仔细理解需求，规划功能。  第6周-第15周（2019-04-01~2019-06-09）：学习并完成每一步的开发，在过程中发现问题并改进问题，学习使用微信小程序的云开发，并通过云函数在JS中操作云数据库中的数据，并进行下一步的功能改进，并优化界面。  第16周-第18周（2019-06-10~2019-06-30）：搜集并添加真实数据，调试测试，通过实际用户的操作，提出建议并修改，撰写报告。 | | |
| **原 主始 要资 参料 考与 文 献** | [01] 徐飞，苗凤君.Oracle数据库基础与案例开发详解.清华大学出版社，2014  [02] 卢潇.软件工程[M].北京：清华大学出版社；北京交通大学出版社，2005  [03] 黑马程序员.Java EE企业级应用开发教程.北京：人民邮电出版社，2017  [04] 张海藩.软件工程导论(第四版)[M].北京:清华大学出版社.2003. | | |

# 摘要

为了方便中工软考研的同学，中工软考研帮把中原工学院软件学院历年学长学姐的报考、分数、详细信息整合到一块，在微信小程序中展示出来，帮助有考研意愿的学生在选择考研目标和考研专业的时候，能参考一下历年学长学姐的报考信息。同学们可以根据学长学姐的报考情况，更好的明确自己的择校、报考等事宜。另一方面，当考研同学确定了院校专业后，也可以从这个网站上查找历年考上这个院校专业的学长学姐，通过网站提供的联系方式咨询学长学姐。

本系统用微信小程序开发平台开发，参考微信开发文档（小程序版）实现所需的各项功能，数据存放在微信小程序提供的云开发数据库里，在JS中通过云函数操作数据库。并且录有2019年考研学子报考院校专业的真实信息。

本文共5章，第1章主要介绍了中工软考研帮开发的背景，研究意义及系统功能，第2章是对中工软考研帮的系统分析，第3章是中工软考研帮的数据库详细设计，第4章是对中工软考研帮的具体实现进行阐述，第5章介绍在实践成果完成过程中的一些心得体会，以及对此次实践成果的评价，进行总结。

关键字：微信小程序；中工软；考研帮；报考；信息

至少3个关键字，要能反映题目

# 目录

[摘要 II](#_Toc13323593)

[目录 1](#_Toc13323594)

[第1章 概述 1](#_Toc13323595)

[1.1 课题背景 1](#_Toc13323596)

[1.2 中工软考研帮的功能 1](#_Toc13323597)

[1.3 中工软考研帮的开发意义 2](#_Toc13323598)

[第2章 系统分析 3](#_Toc13323599)

[2.1技术分析 3](#_Toc13323600)

[2.2需求分析 3](#_Toc13323601)

[第3章 数据库分析与设计 4](#_Toc13323602)

[3.1 实体分析 4](#_Toc13323603)

[3.2关系设计 6](#_Toc13323604)

[3.3数据表设计 7](#_Toc13323605)

[第4章 系统实现 9](#_Toc13323606)

[4.1修改学生联系方式流程 9](#_Toc13323607)

[4.2功能实现 11](#_Toc13323608)

[4.2.1学生基本信息的添加、修改和展示 11](#_Toc13323609)

[4.2.2学生联系方式的添加、修改和查看 13](#_Toc13323610)

[4.2.3学生成绩信息的添加、修改和查看 15](#_Toc13323611)

[第5章 结束语 17](#_Toc13323612)

[指导老师评语 19](#_Toc13323613)

[附录 20](#_Toc13323614)

[1. 管理员数据操作的主要代码： 20](#_Toc13323615)

[1.1 连接数据库的云函数初始化的代码： 20](#_Toc13323616)

[1.2 添加学生基本信息（不包括报考信息）的主要代码 20](#_Toc13323617)

[1.3 添加学生报考信息的主要代码 22](#_Toc13323618)

[1.4 修改学生基本信息的主要代码 22](#_Toc13323619)

[1.5 添加学生详细信息的主要代码 23](#_Toc13323620)

[1.6 修改学生详细信息的主要代码 25](#_Toc13323621)

[1.7 添加学生初试考试科目的主要代码 26](#_Toc13323622)

[1.8 添加学生初试单科成绩的主要代码 28](#_Toc13323623)

[1.9 修改学生初试成绩的主要代码 30](#_Toc13323624)

[2. 学生数据库操作主要代码： 31](#_Toc13323625)

[2.1 连接数据库的云函数初始化的代码： 31](#_Toc13323626)

[2.2 根据年份查询考研学生信息的主要代码： 32](#_Toc13323627)

[2.3 获取学生基本信息的主要代码 34](#_Toc13323628)

[2.4 获取学生详细信息的主要代码 35](#_Toc13323629)

[2.5 获取学生成绩信息的主要代码 36](#_Toc13323630)

# 第1章 概述

本章主要阐述中工软考研帮研发的开发背景、主要功能及开发意义。以下中工软考研帮简称为此程序。

## 1.1 课题背景

有研究调查表明，有七成以上的在校大学生有明确的考研意愿。调查结果显示，75.3%的同学有较为明确的考研意向，仍在考虑的占20.2%。而根据参与调查的学生的各种维度进行分类发现，这种近乎盲目的考研热几乎是不分年级、不分性别、不分学校层次的。而考研是为了什么呢？提高就业竞争力，给自己的学历加码是主要原因。大多数考研人害怕就业受歧视，非全日制研究生接受率不高；考研择校更关注学校整体知名度而非学术水平。

近年来，我校越来越鼓励中工学子考研，而同学们在选择考研后面临的第一个问题就是择校择专业。大多数同学在确定院校专业的时候对所选择的院校专业缺乏准确的、整体的认识。我们知道，学长学姐也曾经经历过这样的抉择。为了帮助中工学子考研选择院校和选择专业，在韩老师的引导下，我开发了基于微信小程序的中工软考研帮。

此程序把历年考生的考研报考信息、初试成绩、考生的联系方式等详细信息整合到一起，供有考研意向的在校学生在选择院校专业的时候，能参考一下历年学长学姐的报考信息，从而能有一个一个更好的选择。同时，在校学生也能通过此程序上提供的详细信息联系到学长学姐，进一步咨询更为详细的考研信息。

## 1.2 中工软考研帮的功能

中工软考研帮的功能为：历年考研学子的院校专业报考信息、初试成绩信息的展示，历年学子联系方式等详细信息的展示。中工软考研帮的使用群体为中工软有考研意愿的在校学生。

## 1.3 中工软考研帮的开发意义

近年来，我校越来越鼓励中工学子考研，而同学们在选择考研后面临的第一个问题就是择校择专业。大多数同学在确定院校专业的时候对所选择的院校专业缺乏准确的、整体的认识。我们知道，学长学姐曾经也经历过这样的抉择。而很久以来，很少有专门的平台是用来展示同学校同专业已经考过研的学长学姐在考研时所选择的院校专业和考试的结果。而此程序的功能，就是展示这些信息。帮助选择考研的在校学生在参考历年学长学姐的报考信息、咨询了学长关于考研的准备过程后，能对考研这件事有一个更清楚的认识，从而更好的考上研究生。

# 第2章 系统分析

本章主要是对中工软考研帮进行合理的分析，通过技术分析、需求分析这两个方面描述这个项目。

## 2.1技术分析

此程序的开发是在微信开发者工具上完成的。

页面设计用到的技术有：WXML、WXSS。

数据库存储用到的技术：微信小程序云开发数据库。

数据库操作用到的技术：微信小程序云函数。

用到的开发工具有：微信开发者工具、手机微信等。

## 2.2需求分析

中工软考研帮所需实现的功能模块图如图2.1所示.

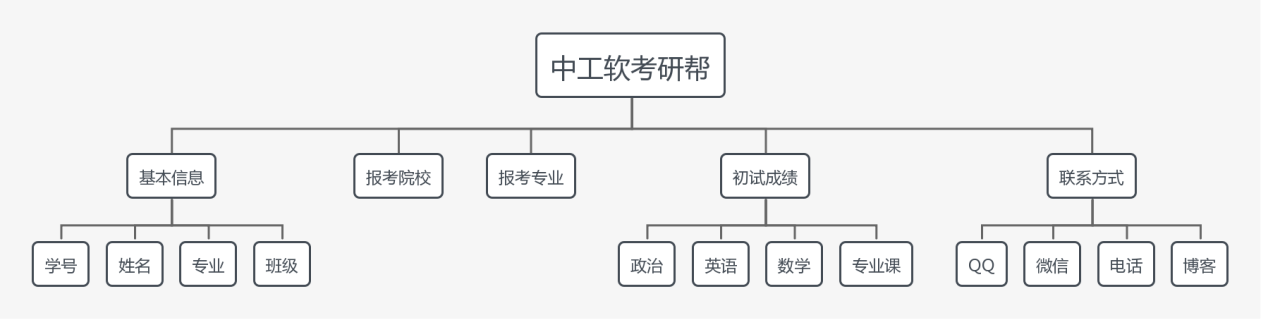


图2.1 系统功能模块图

用户需求如下：

信息展示。首页展示历年考研学子的姓名和初试成绩及在学院的排名。点击首页的姓名可以查看学生的基本信息和详细信息，包括学号、姓名、性别、学校、学院、班级；报考院校、报考专业、研究方向、考研年份；手机号、QQ、微信、博客、GitHub等。点击首页的分数可以查看学生的成绩信息，包括四门考试的科目、单科成绩等，首页主要展示历年考研学子的院校专业报考信息、初试成绩信息的展示，详细成绩分布的展示，历年学子联系方式等详细信息的展示。

# 第3章 数据库分析与设计

数据库拥有很强的功能，可以建立一个数据库把我们所需要的数据给存储起来，并可以进行增删改查的操作，还可以对数据库进行导入导出等操作。我们采用微信小程序云开发中提供的数据库来保存中工软考研帮的各项信息。

## 3.1 实体分析

中工软考研帮涉及的实体包括学生、联系方式、成绩、管理员等。经过分析，学生实体E-R图如图3.1所示，联系方式实体E-R图如图3.2所示，成绩实体E-R图如图3.3所示，管理员实体E-R图如图3.4所示。



图3.1 学生实体E-R图



图3.2 联系方式实体E-R图



图3.3 成绩实体E-R图



图3.4 管理员实体E-R图

## 3.2关系设计

这里有四个实体，分别是：学生、联系方式、成绩、管理员。它们之间的关系是：成绩属于学生，一对一。联系方式属于学生，一对一。管理员管理学生，一对多。管理员管理联系方式，一对多。管理员管理成绩，一对多。中工软考研帮的系统E-R图如图3.5所示。



图3.5 系统总体E-R图

## 3.3数据表设计

中工软考研帮的数据库设计共有四个表：学生表、联系方式表、成绩表、管理员表。学生表如表3.1所示，联系方式表如表3.2所示、成绩表如表3.3所示、管理员表如表3.4所示。

1.学生表。记录学生的基本信息，包括学生的学号、姓名、性别、学校、学院、专业、班级、报考院校、报考专业、初试总分、考研年份等。

表3.1 student数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **可否为空** | **约束条件** | **说明** |
| \_id | string | NOT NULL | 主键 | 学号 |
| name | string | NOT NULL | 无 | 姓名 |
| sex | string | NOT NULL | 无 | 性别 |
| sch | string | NOT NULL | 无 | 学校 |
| college | string | NOT NULL | 无 | 学院 |
| major | string | NOT NULL | 无 | 专业 |
| clazz | string | NOT NULL | 无 | 班级 |
| register\_sch | string | NOT NULL | 无 | 报考院校 |
| register\_major | string | NOT NULL | 无 | 报考专业 |
| register\_direction | string | NOT NULL | 无 | 研究方向 |
| score\_total | string | NOT NULL | 无 | 总分 |
| score\_year | string | NOT NULL | 无 | 考研年份 |

2.联系方式表。记录学生的联系方式，包括学号、姓名、性别、电话、QQ、微信、博客地址、GitHub地址等。

表3.2 info数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **可否为空** | **约束条件** | **说明** |
| \_id | string | NOT NULL | 主键 | 学号 |
| name | string | NOT NULL | 无 | 姓名 |
| sex | string | NOT NULL | 无 | 性别 |
| tel | string | NULL | 无 | 手机号 |
| qq | string | NULL | 无 | QQ号 |
| wechat | string | NULL | 无 | 微信 |
| blog | string | NULL | 无 | 博客 |
| github | string | NULL | 无 | GitHub |

3.成绩表，记录了学生的考研初试成绩信息。包括学号、姓名、单科考试科目、单科成绩、考研年份等。

表3.3 score数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **可否为空** | **约束条件** | **说明** |
| \_id | string | NOT NULL | 主键 | 学号 |
| name | string | NOT NULL | 否 | 姓名 |
| politics\_score | string | NOT NULL | 否 | 政治成绩 |
| english\_subject | string | NOT NULL | 否 | 英语科目 |
| english\_score | string | NOT NULL | 否 | 英语成绩 |
| math\_subject | string | NOT NULL | 否 | 数学科目 |
| math\_score | string | NOT NULL | 否 | 数学成绩 |
| major\_subject | string | NOT NULL | 否 | 专业课科目 |
| major\_score | string | NOT NULL | 否 | 专业课成绩 |
| score\_year | string | NOT NULL | 否 | 总分 |

4.管理员表。记录管理员信息。包括管理员的编号、姓名、密码、邮箱等。

表3.4 manager数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **可否为空** | **约束条件** | **说明** |
| \_id | string | NOT NULL | 主键 | 管理员编号 |
| name | string | NOT NULL | 无 | 姓名 |
| pwd | string | NOT NULL | 无 | 密码 |
| email | string | NULL | 无 | 邮箱 |

# 第4章 系统实现

软件设计是一个把软件需求变换成软件表示的过程。最初这种表示知识描绘出软件的总的框架,然后进一步细化，把它加工成在程序细节上非常接近于源程序的软件表示。从工程管理的角度来看，软件设计分两步完成。首先做概要设计，将软件需求转化成数据结构和软件的系统结构。然后是详细设计，即过程设计。通过对结构表示进行细化，得到软件的详细的数据结构和算法。

中工软考研帮设计阶段主要目的是将系统分析阶段所提出的反映了用户信息需求的系统逻辑方案转换成可以实施的基于计算机与通信系统的技术方案。

## 4.1修改学生联系方式流程

修改学生联系方式流程如图4.1所示。



图4.1 修改学生联系方式流程

## 4.2功能实现

系统的用户界面作为人机接口起着越来越重要的作用。它的好坏直接影响到系统的使用性。具有友好界面的系统，对于用户来说，无疑是一种享受。

本系统的各模块功能实现过程如下所示。

### 4.2.1学生基本信息的添加、修改和展示

管理员添加的学生基本信息存储到云数据库的学生表。管理员修改学生信息时，长按主页的学生姓名，在弹出的操作菜单中选择修改学生基本，如何在修改学生基本信息页面进行学生基本信息的数据修改。修改成功后，数据库中学生表对应的信息也会修改。学生基本信息的添加修改只有管理员能操作。

由于学生数据较多，学生基本信息的展示采用分页和不同年份分别展示的方法展示数据。每页展示最多8条学生基本信息数据，剩余的数据在下一页展示。用户可以根据考研年份查看历年的考研报考数据。普通用户和管理员都可以查看学生基本信息。

学生基本信息添加页面如图4.2所示，学生基本信息修改页面如图4.3所示，学生基本信息展示页面如图4.4所示。



图4.2 学生基本信息添加页面

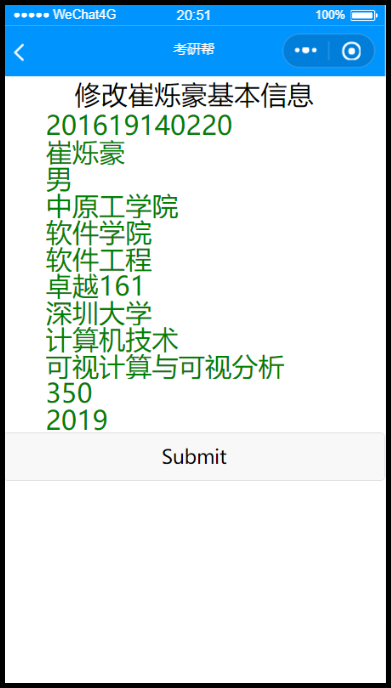


图4.3 学生基本信息修改页面

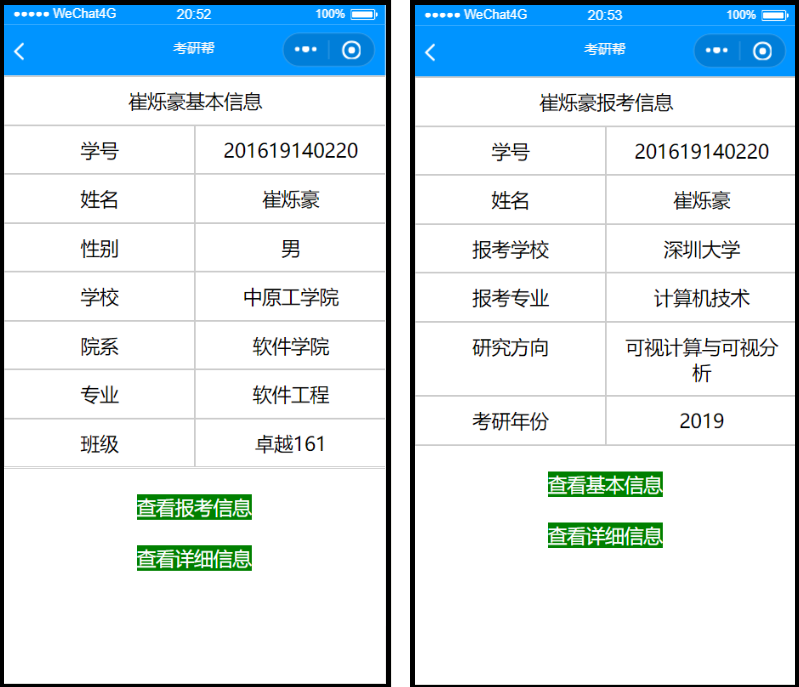


图4.4 学生基本信息展示页面

### 4.2.2学生联系方式的添加、修改和查看

在学生基本信息已经存在的基础上，管理员可以添加和修改学生的联系方式信息。如果学生没有联系方式数据，管理员点击查看详细信息，会弹出是否添加学生详细信息的对话框，点击确认进入学生联系方式添加页面。如果学生有联系方式数据，管理员长按首页的学生姓名即，在弹出的菜单栏中选择修改学生详细信息，进入学生联系方式修改页面。学生联系方式的添加修改只有管理员能操作。

点击学生基本信息展示页面下的查看详细信息可以查看学生的联系方式信息。普通用户和管理员都可以查看学生的联系方式信息。

学生联系方式的添加页面如图4.5所示。学生联系方式的修改页面如图4.6所示。学生联系方式的查看页面如图4.7所示。

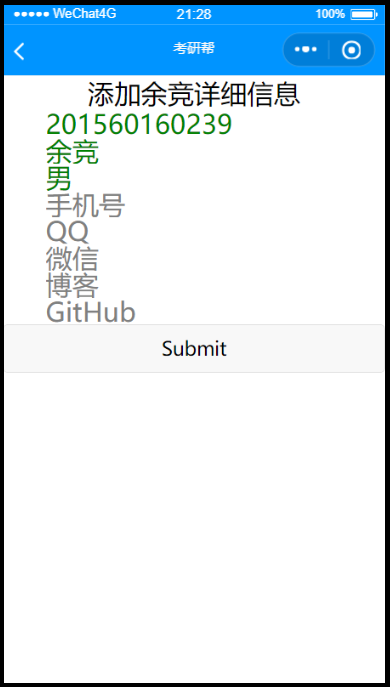


图4.5 学生联系方式添加页面

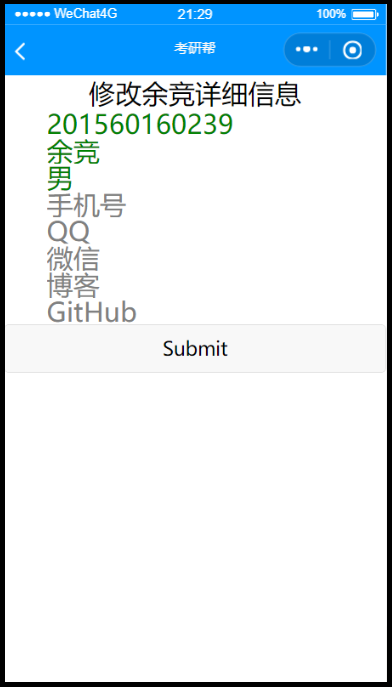


图4.6 学生联系方式修改页面



图4.7 学生联系方式查看页面

### 4.2.3学生成绩信息的添加、修改和查看

在学生基本信息已经存在的基础上，管理员可以添加和修改学生的成绩信息。如果学生没有成绩信息数据，管理员点击首页中的初试分数即进入学生成绩信息添加页面。如果学生有成绩信息数据，管理员长按首页中的初试成绩即进入学生成绩信息修改页面。学生成绩信息的添加修改只有管理员能操作。

点击首页中的初试成绩可以查看学生的成绩信息。普通用户和管理员都可以查看学生的成绩信息。

学生成绩信息的添加页面如图4.8所示。学生成绩信息的修改页面如图4.9所示。学生成绩信息的查看页面如图4.10所示。

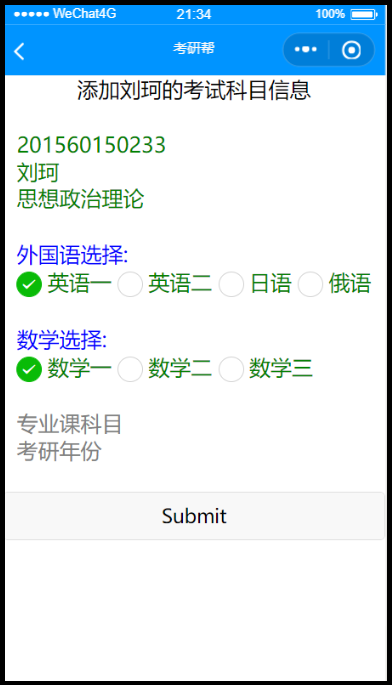


图4.8 学生成绩信息添加页面

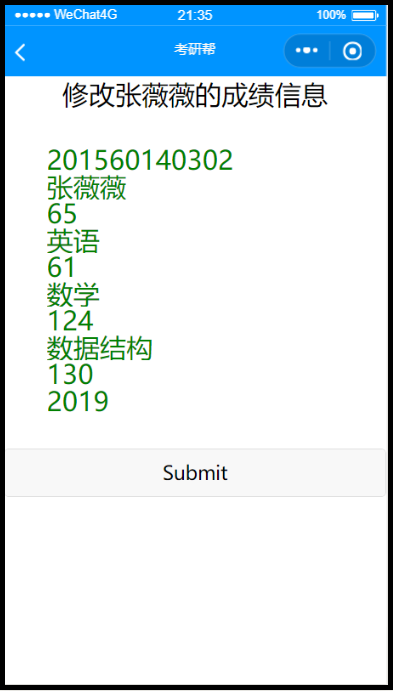


图4.9 学生成绩信息修改页面



图4.10 学生成绩信息查看页面

# 第5章 结束语

20世纪最重要的资源是石油，而21世纪这一状况已经改变了。Google 、脸书、亚马逊掌握着每一个人的信息、行踪、喜好，海量的数据带来巨大的财富和权利。信息时代的竞争已经完全改变，反垄断需要新的标准和紧跟时代的行动。数据经济中谁掌握了数据谁就掌握了市场。

在使用微信开发者工具进行中工软考研帮的开发设计中，我遇到过不少功能实现上的困难，主要是因为对微信小程序的开发还不熟悉。以往用JAVA设计后端，而微信开发者工具推出了云开发，淡化了后台的概念。开发微信小程序的重点主要集中在前端的设计和数据库的操作上。前端不再是以往的HTML和CSS，而是微信自己推出的程序设计语言WXML和WXSS，虽然大体上和HTML、CSS很相似，但是在一些标签的使用上还是有一些差异。在数据库方面，微信小程序的数据库和以往的开发过程中使用的MYSQL有很大的不同，MYSQL是一种关系型数据库，微信小程序的数据库是一种非关系数据库。微信小程序中的集合相当于MYSQL中的表，微信小程序中的记录相当于MYSQL中的一行数据，微信小程序中的字段相当于MYSQL中的列。在数据库的操作方面，微信小程序的开发文档上对数据操作进行了详细的介绍，我根据上面的介绍，以及博客上其他人的经验，在JS中设计了对应的数据库操作代码。在数据获取上，学长学姐的初试成绩从学院公布的2019年考研光荣榜上采集，学长学姐的报考院校专业从2019年考研报考确认缴费名单上采集，学长学姐的学号信息从学院公布的学期学业进展表上采集。虽然数据并不全面，但是绝对真实可靠。

这次的实践也让我认识到，每一次的程序开发都会遇到很多不懂得问题，包括技术实现、系统优化、页面美化等方面上的问题。遇到技术的的问题不能怕，先自己尝试着解决；自己无法解决的，通过百度、谷歌等搜索引擎查阅并学习前辈的经验教训；搜索引擎上也解决不了的问题，问一问同学老师；如果以上方法都解决不了，把问题详细描述一下，记在笔记里，留给以后解决，同时换个思路实现功能。解决的问题，也要记在笔记里，以后再遇到相似的问题，方便查阅参考。遇到逻辑上、设计上的问题时，要善于利用画图工具，画下系统功能逻辑图、技术实现的过程图。整个项目的逻辑要梳理清楚，多做日志记录。然后在后期实际开发中，根据前期设计按部就班的推进开发进度。这样，不仅开发更高效，也能避免在后期走一些不必要的弯路。

非常感谢韩老师的指导帮助。

# 指导老师评语

成绩：

指导老师签字：

年 月 日

# 附录

## 管理员数据操作的主要代码：

### 1.1 连接数据库的云函数初始化的代码：

wx.cloud.init({

env: 'wx-tsui-e1xd2'

})

const db = wx.cloud.database({

env: 'wx-tsui-e1xd2'

})

### 1.2 添加学生基本信息（不包括报考信息）的主要代码

Page({

// 添加学生基本信息（不包括报考信息）

data: {

error:"",

success:"",

sch: "中原工学院",

college: "软件学院",

major: "软件工程",

score\_year: "2019",

},

formSubmit: function (e) {

//如果id未输入，不能插入学生信息，并提示

if (e.detail.value.id == "" ){

this.setData({

success: "",

error:"请输入学号"

})

} else if (e.detail.value.name == ""){

this.setData({

success: "",

error: "请输入姓名"

})

}else{

console.log('form发生了submit事件，携带数据为：', e.detail.value),

db.collection('KaoYan\_student').add({

data: {

\_id: e.detail.value.id,

name: e.detail.value.name,

sex: e.detail.value.sex,

sch: e.detail.value.sch,

college: e.detail.value.college,

major: e.detail.value.major,

clazz: e.detail.value.clazz,

register\_sch: e.detail.value.register\_sch,

register\_major: e.detail.value.register\_major,

register\_direction: e.detail.value.register\_direction,

score\_total: e.detail.value.score\_total,

score\_year: e.detail.value.score\_year,

}

})

this.setData({

success: wx.showToast({

title: "插入成功",

})

})

}

},

formReset: function () {

console.log('form发生了reset事件')

this.setData({

success: "",

error: "",

sch: "中原工学院",

college: "软件学院",

major: "软件工程",

score\_year: "2019",

})

},

formClean:function(){

console.log('form发生了clean事件')

this.setData({

success: "",

error: "",

sch: "",

college: "",

major: "",

score\_year: "",

})

},

})

### 1.3 添加学生报考信息的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 添加学生报考信息

\*/

data: {

student: {}

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是register")

let that = this

db.collection('KaoYan\_student').doc(options.id).get({

success: function (res) {

that.setData({

student: res.data

})

}

})

},

})

### 1.4 修改学生基本信息的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 修改学生基本信息（包括报考信息）

\*/

data: {

student:{}

},

// 提交按钮，修改数据

formSubmit:function(e){

console.log('form发生了submit事件，携带数据为：', e.detail.value),

db.collection('KaoYan\_student').doc(e.detail.value.id).update({

data: {

name: e.detail.value.name,

sex: e.detail.value.sex,

sch: e.detail.value.sch,

college: e.detail.value.college,

major: e.detail.value.major,

clazz: e.detail.value.clazz,

register\_sch: e.detail.value.register\_sch,

register\_major: e.detail.value.register\_major,

register\_direction: e.detail.value.register\_direction,

score\_total: e.detail.value.score\_total,

score\_year: e.detail.value.score\_year,

},

success:function(){

setTimeout(function () {

wx.showToast({

title: '数据修改成功',

})

}, 2000)

//返回上一页

wx.navigateBack({changed: true});

},

fail:function(){

console.log("基本信息数据修改失败")

}

})

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是updateStudent")

console.log("学号为"+options.id)

let that = this

db.collection('KaoYan\_student').doc(options.id).get({

success: function (res) {

that.setData({

student: res.data

})

}

})

},

})

### 1.5 添加学生详细信息的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 添加详细信息

\*/

data: {

info:{},

student:{}

},

formSubmit: function (e) {

console.log('form发生了submit事件，携带数据为：', e.detail.value),

db.collection('KaoYan\_info').add({

data: {

\_id: e.detail.value.id,

name: e.detail.value.name,

sex: e.detail.value.sex,

tel: e.detail.value.tel,

qq: e.detail.value.qq,

wechat: e.detail.value.wechat,

blog: e.detail.value.blog,

github: e.detail.value.github

},

success: function(){

setTimeout(function () {

wx.showToast({

title: '插入成功',

})

}, 2000)

//返回上一页

wx.navigateBack({ changed: true });

}

})

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是addInfo")

console.log(options.id)

let that = this

db.collection("KaoYan\_student").doc(options.id).get({

success: function (res) {

console.log(res.data)

that.setData({

student: res.data

})

},

})

},

})

### 1.6 修改学生详细信息的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 修改详细信息

\*/

data: {

info:{}

},

// 提交按钮，修改数据

formSubmit: function (e) {

console.log('form发生了submit事件，携带数据为：', e.detail.value),

db.collection('KaoYan\_info').doc(e.detail.value.id).update({

data: {

name: e.detail.value.name,

sex: e.detail.value.sex,

tel: e.detail.value.tel,

qq: e.detail.value.qq,

wechat: e.detail.value.wechat,

blog: e.detail.value.blog,

github: e.detail.value.github,

},

success: function () {

setTimeout(function () {

wx.showToast({

title: '数据修改成功',

})

}, 2000)

//返回上一页

wx.navigateBack({ changed: true });

},

fail: function () {

console.log("详细信息数据修改失败")

}

})

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是updateInfo")

console.log("学号为"+options.id)

let that = this

db.collection('KaoYan\_info').doc(options.id).get({

success: function (res) {

that.setData({

info: res.data

})

},

})

},

})

### 1.7 添加学生初试考试科目的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 添加初试考试科目信息

\*/

data: {

englishItems:[

{ name: '英语一', checked: 'true' },

{ name: '英语二' },

{ name: '日语' },

{ name: '俄语' }

],

mathItems: [

{name: '数学一',checked:'true'},

{name: '数学二'},

{name: '数学三'}

],

score:{}

},

//英语考试科目，单选按钮

englishRadioChange: function(e) {

console.log('英语科目：', e.detail.value)

},

// 数学考试科目，单选按钮

mathRadioChange: function (e) {

console.log('数学科目：', e.detail.value)

},

// 提交按钮

formSubmit: function (e) {

console.log('form发生了submit事件，携带数据为：', e.detail.value)

if (e.detail.value.major\_subject==""){

wx.showToast({

title: "请输入专业课科目",

})

// 如果信息输入正确，则插入数据库

}else{

db.collection('KaoYan\_score').add({

data: {

\_id: e.detail.value.id,

name: e.detail.value.name,

english\_subject:e.detail.value.english\_subject,

math\_subject:e.detail.value.math\_subject,

major\_subject:e.detail.value.major\_subject,

score\_year: e.detail.value.score\_year

},

success: function () {

wx.showToast({

title: "插入成功"

}),

wx.showModal({

title: '添加成绩信息',

content: '继续添加成绩信息？',

success(res) {

if (res.confirm) {

wx.navigateTo({

url: './addScore?id=' + e.detail.value.id,

})

} else if (res.cancel) {

//返回上一页

wx.navigateBack({ changed: true });

}

}

})

}

})

}

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是addScoreSubject，页面加载")

console.log(options.id)

let that = this

db.collection('KaoYan\_student').doc(options.id).get({

success: function (res) {

that.setData({

score: {

\_id: res.data.\_id,

name:res.data.name,

}

})

}

})

},

})

### 1.8 添加学生初试单科成绩的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 添加初试单科成绩

\*/

data: {

score:{}

},

// 提交按钮

formSubmit: function (e) {

console.log('form发生了submit事件，携带数据为：', e.detail.value)

db.collection('KaoYan\_score').doc(e.detail.value.id).update({

data: {

politics\_score: e.detail.value.politics\_score,

english\_score: e.detail.value.english\_score,

math\_score: e.detail.value.math\_score,

major\_score: e.detail.value.major\_score,

score\_total: parseInt(e.detail.value.politics\_score)+

parseInt(e.detail.value.english\_score)+

parseInt(e.detail.value.math\_score)+

parseInt(e.detail.value.major\_score)

},

success: function () {

setTimeout(function () {

wx.showToast({

title: '成绩数据添加成功',

})

}, 2000)

db.collection('KaoYan\_student').doc(e.detail.value.id).update({

data: {

score\_total: parseInt(e.detail.value.politics\_score) +

parseInt(e.detail.value.english\_score) +

parseInt(e.detail.value.math\_score) +

parseInt(e.detail.value.major\_score)

}

})

//返回上一页

wx.navigateBack({

delta:3

});

},

fail: function () {

console.log("成绩信息数据修改失败")

}

})

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是addScore，页面加载")

console.log(options.id)

let that = this

db.collection('KaoYan\_score').doc(options.id).get({

success: function (res) {

console.log(res.data.english\_subject + "成绩：" + res.data.english\_score)

// 如果没有成绩信息

if (typeof res.data.english\_score =="undefined"){

that.setData({

score: res.data

})

// 如果已经有成绩信息，提示并返回上一页

}else{

setTimeout(function () {

wx.showToast({

title: '已有成绩信息',

})

}, 2000)

wx.navigateBack({

delta:1

})

}

}

})

db.collection('KaoYan\_score').doc(options.id).get({

success: function (res) {

}

})

},

})

### 1.9 修改学生初试成绩的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 修改初试成绩

\*/

data: {

score:{}

},

// 提交按钮，修改数据

formSubmit: function (e) {

console.log('form发生了submit事件，携带数据为：', e.detail.value),

db.collection('KaoYan\_score').doc(e.detail.value.id).update({

data: {

politics\_score: e.detail.value.politics\_score,

english\_subject: e.detail.value.english\_subject,

english\_score: e.detail.value.english\_score,

math\_subject: e.detail.value.math\_subject,

math\_score: e.detail.value.math\_score,

major\_subject: e.detail.value.major\_subject,

major\_score: e.detail.value.major\_score,

score\_total: parseInt(e.detail.value.politics\_score) +

parseInt(e.detail.value.english\_score) +

parseInt(e.detail.value.math\_score) +

parseInt(e.detail.value.major\_score),

score\_year: e.detail.value.score\_year

},

success: function () {

setTimeout(function () {

wx.showToast({

title: '数据修改成功',

})

}, 2000)

db.collection('KaoYan\_student').doc(e.detail.value.id).update({

data: {

score\_total: parseInt(e.detail.value.politics\_score) +

parseInt(e.detail.value.english\_score) +

parseInt(e.detail.value.math\_score) +

parseInt(e.detail.value.major\_score)

}

})

//返回上一页

wx.navigateBack({ changed: true });

},

fail: function () {

console.log("详细信息数据修改失败")

}

})

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是updateScore，页面加载")

let that = this

db.collection('KaoYan\_score').doc(options.id).get({

success: function (res) {

that.setData({

score: res.data

})

}

})

},

})

## 学生数据库操作主要代码：

### 2.1 连接数据库的云函数初始化的代码：

wx.cloud.init({

env: 'wx-tsui-e1xd2'

})

const db = wx.cloud.database({

env: 'wx-tsui-e1xd2'

})

### 2.2 根据年份查询考研学生信息的主要代码：

Page({

/\*\*

\* 页面的初始数据

studentList：根据year查询到的学生基本信息数据

\*/

data: {

// KaoYan\_student: '未获取',

year: '2019',

studentList: [],

},

// 点击姓名，查看学生基本信息

NamePress: function (e) {

wx.navigateTo({

url: '/pages/student/student?id=' + e.currentTarget.dataset.variable,

})

},

// 点击分数，查看学生成绩信息

ScorePress: function (e) {

wx.navigateTo({

url: '/pages/score/score?id=' + e.currentTarget.dataset.variable,

})

},

// js中实现绑定的长按点击事件

// 长按姓名，可以进行修改添加操作

// 需要管理员权限

NamelongPress: function (e) {

console.log(e.currentTarget.dataset.variable)

wx.showActionSheet({

itemList: ['修改基本信息', '修改详细信息'],

success(res) {

console.log(res.tapIndex)

if(res.tapIndex==0){

console.log("点击的是修改基本信息")

wx.navigateTo({

url: '../student/updateStudent?id=' + e.currentTarget.dataset.variable,

})

} else if (res.tapIndex == 1){

console.log("点击的是修改详细信息")

wx.navigateTo({

url: '../info/updateInfo?id=' + e.currentTarget.dataset.variable,

})

}

},

fail(res) {

console.log(res.errMsg)

}

})

},

// 长按主页中的分数，修改成绩信息

ScorelongPress: function (e) {

console.log(e.currentTarget.dataset.variable)

wx.showModal({

title: '修改成绩信息',

content: '修改成绩信息？',

success(res) {

if (res.confirm) {

wx.navigateTo({

url: '../score/updateScore?id=' + e.currentTarget.dataset.variable,

})

}

}

})

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

var that = this

console.log("页面已加载")

db.collection('KaoYan\_student').where({

"score\_year": that.data.year,

}).orderBy('score\_total', 'desc').get({

success: function (res) {

console.log(res.data)

// let test = res.data.sort(res.data.)

that.setData({

studentList: res.data

})

}

})

},

/\*\*

\* 生命周期函数--监听页面显示

\*/

onShow: function () {

console.log("student页面显示")

this.onLoad()

},

/\*\*

\* 页面相关事件处理函数--监听用户下拉动作

\*/

onPullDownRefresh: function () {

console.log("用户下拉")

var that = this

db.collection('KaoYan\_student').where({

"score\_year": that.data.year,

}).orderBy('score\_total', 'desc').get({

success: function (res) {

console.log(res.data)

// let test = res.data.sort(res.data.)

that.setData({

studentList: res.data

})

wx.showToast({

title: '已刷新',

})

}

})

setTimeout(function () {

// 这里写刷新要调用的函数，比如：

wx.stopPullDownRefresh();

}, 500);

},

})

### 2.3 获取学生基本信息的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 获取学生基本信息

\*/

data: {

student:{}

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是student")

let that = this

db.collection('KaoYan\_student').doc(options.id).get({

success: function (res) {

console.log(typeof res.data.\_id)

that.setData({

student:res.data

})

}

})

},

})

### 2.4 获取学生详细信息的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 根据学号获取学生详细信息

\*/

data: {

info:{}

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是info，页面加载")

console.log(options.id)

let that = this

db.collection('KaoYan\_info').doc(options.id).get({

success: function (res) {

that.setData({

info: res.data

})

},

fail:function(){

console.log("无详细数据")

wx.showModal({

title: '无学生详细信息',

content: '添加学生详细信息？',

success(res) {

if (res.confirm) {

wx.navigateTo({

url: './addInfo?id=' + options.id,

})

} else if (res.cancel) {

//返回上一页

wx.navigateBack({changed:true});

}

}

})

}

})

},

})

### 2.5 获取学生成绩信息的主要代码

Page({

/\*\*

\* 页面的初始数据

\* 获取学生成绩信息

\*/

data: {

score:{

\_id:"",

name:"",

politics\_score:-1,

english\_subject:"英语",

english\_score:-1,

math\_subject:"数学",

math\_score:-1,

major\_subject:"专业课",

major\_score:-1,

score\_total: -1,

score\_year:2019

}

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

console.log("这里是score，页面加载")

console.log(options.id)

let that = this

db.collection('KaoYan\_score').doc(options.id).get({

success: function (res) {

that.setData({

score: res.data

})

},

fail: function () {

console.log("无成绩数据")

wx.showModal({

title: '无学生成绩信息',

content: '添加学生成绩信息？',

success(res) {

if (res.confirm) {

wx.navigateTo({

url: './addScoreSubject?id=' + options.id,

})

} else if (res.cancel) {

//返回上一页

wx.navigateBack({ changed: true });

}

}

})

}

})

},

})