**详细设计报告**

1. 引言

1.1编写目的

随着计算机的发展，随之而产生的许多计算机软件业在一步一步改变着人们的日常生活，包括衣、食、住、行等各个方面，但是在此发展过程中，人们对计算机行业产品的需求也在不断提高，人们对计算机产品需求的日益提高与计算机生产发展速度已经逐渐形成了矛盾，这迫切需要新一代、现代化的计算机产品的出现来满足人们的需求。因此我们以此开发项目作为实例来阐释一下如何才能以最小的代价（包括时间和金钱）来开发出高质量的软件来，以此来满足人们日益增长的需求。

问卷调查是社会学常用的研究方法,可用于描述性、解释性或探索性的研究。随着计算机技术的迅速发展, 计算机辅助问卷调查开始兴起。它的出现并不是取代传统调查方式,而是与之相辅相成,以达成最佳调查效果。但问卷如何能更加广泛的传播以及不同类别的问卷如何能精准的发布到相关人群的手中一直是困扰问卷发行的主要问题。随着移动互联网的发展，移动社交软件已经成为人们日常生活不可或缺的一部分。所以通过社交软件平台发布可以有效的解决上述问题。

本项目小组通过使用在祖国大陆最流行的社交软件-微信中的微信小程序来生成调查问卷以方便微信用户登录填写调查问卷。本项目小组基于此方案撰写软件系统详细设计以供天津理工大学企划与项目开发指导老师以及指导学长、学姐阅读。

1.2项目背景

本项目名称：基于微信小程序的调查问卷生成系统。

本项目为天津理工大学实训项目，是一个基于微信小程序的调查问卷生成器。

使用用户：在校大学生，有需要了解商品潜在客户的商家；

实现软件单位：天津理工大学基于微信小程序的调查问卷生成器课题小组。

其他关联软件或其他系统：MATLAB Excel，用来处理数据，进行数据分析。

腾讯云，用来存储数据。

微信开发者工具，用来编写微信小程序代码。

PhpMyAdmin，用来管理Mysql数据库。

微信公众平台，用来进行微信小程序开发管理、分析统计进入微信小程序用户数据，涵盖数据概况、实时统计、访问分析、来源分析以及生成用户画像。

1.3术语定义

1.微信小程序：英文名Mini Program，是一种不需要下载安装即可使用的应用，它实现了应用“触手可及”的梦想，用户扫一扫或搜索即可打开应用。

2.MATLAB：MATLAB是美国MathWorks公司出品的商业数学软件，用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境，主要包括MATLAB和Simulink两大部分。MATLAB是matrix&laboratory两个词的组合，意为矩阵工厂（矩阵实验室）。是由美国mathworks公司发布的主要面对科学计算、可视化以及交互式程序设计的高科技计算环境。它讲数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等诸多强大功能集成在一个易于使用的视窗环境中，为科学研究、工程设计以及必须进行有效数值计算的众多科学领域提供了一种全面的解决方案，并在很大程度上摆脱了传统非交互式程序设计语言（如C、Fortran）的编辑模式，代表了当今国际科学计算软件的先进水平。

3.Microsoft Excel是Microsoft为使用Windows和Apple Macintosh操作系统的电脑编写的一款电子表格软件。直观的界面、出色的计算功能和图表工具，再加上成功的市场营销，使Excel成为最流行的个人计算机数据处理软件。在1993年，作为Microsoft Office的组件发布了5.0版之后，Excel就开始成为所适用操作平台上的电子制表软件的霸主。

4.腾讯云：腾讯云有着深厚的基础架构，并且有着多年对海量互联网服务的经验，不管是社交、游戏还是其他领域，都有多年的成熟产品来提供产品服务。腾讯在云端完成重要部署，为开发者及企业提供云服务、云数据、云运营等整体一站式服务方案。具体包括云服务器、云存储、云数据库和弹性web引擎等基础云服务；腾讯云分析（MTA）、腾讯云推送（信鸽）等腾讯整体大数据能力；以及QQ互联、QQ空间、微云、微社区等云端链接社交体系。这些正是腾讯云可以提供给这个行业的差异化优势，造就了可支持各种互联网使用场景的高品质的腾讯云技术平台。

5.用户画像，用户画像又称用户角色，作为一种勾画目标用户、联系用户诉求与设计方向的有效用户，用户画像在各领域得到了广泛的应用。我们在实际操作的过程中往往会以最为浅显和贴近生活的话语将用户的属性、行为与期待联结起来。作为实际用户的虚拟代表，用户画像所形成的用户角色并不是脱离产品和市场之外所构建出来的，形成的用户角色需要有代表性能代表产品的主要受众和目标群体。

6.phpMyAdmin：phpMyAdmin是一个以PHP为基础，以Web-Base方式架构在网站主机上的MySQL的数据库管理工具，让管理者可用Web接口管理MySQL数据库。借由此Web接口可以成为一个简易方式输入繁杂SQL语法的较佳途径，尤其要处理大量资料的汇入及汇出更为方便。其中一个更大的优势在于由于phpMyAdmin跟其他PHP程式一样在网页服务器上执行，但是您可以在任何地方使用这些程式产生的HTML页面，也就是于远端管理MySQL数据库，方便的建立、修改、删除数据库以及资料表。也可借由phpMyAdmin建立常用的php语法，方便编写网页时所需要的sql语法正确性。

7. MySQL：关系型数据库库管理系统。MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于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。MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表种，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。

8.数据聚类：所谓数据聚类是指根据数据的内在性质将数据分成一些聚合类，每一聚合类中的元素尽可能具有相同的特性，不同聚合类之间的特性差别尽可能大。聚类分析的目的是分析数据是否属于各个独立的分组，使一组中的成员彼此相似，而与其他组中的成员不同。它对一个数据对象的集合进行分析，但与分类分析不同的是，所划分的类是未知的，因此，聚类分析也称为无指导或无监督的（Unsupervised）学习。聚类分析的一般方法是将数据对象分组为多个类或簇（Cluster），在同一簇中的对象之间具有较高的相似度，而不同簇中的对象差异较大。由于聚类分析的上述特征，在许多应用中，对数据集进行了聚类分析后，可将一个簇中的各数据对象作为一个整体对待。数据聚类（Cluster analysis）是对于静态数据分析的一门技术，在许多领域受到广泛应用，包括机器学习，数据挖掘，模式识别，图像分析以及生物信息。

9．DBA：数据库管理员（Database Administrator，简称DBA），是从事管理和维护数据库管理系统（DBMS）的相关工作人员的统称，属于运维工程师的一个分支，主要负责业务数据库从设计、测试到部署交付的全生命周期管理。DBA的核心目标是保证数据库管理系统的稳定性、安全性、完整性和高性能。DBA的主要职责是运维和管理数据库管理系统，侧重于运维管理。

10．Session：在计算机中，尤其是在网络应用中，称为“会话控制”。Session对象存储特定用户会话所需的属性及配置信息。这样，当用户在应用程序的Web页之间跳转时，存储在Session对象中的变量将不会丢失，而是在整个用户会话中一直存在下去。当用户请求来自应用程序的Web页时，如果该用户还没有会话，则Web服务器将自动创建一个Session对象。当会话过期或被放弃后，服务器将终止该会话。Session对象最常见的一个用法就是存储用户的首选项。例如，如果用户指明不喜欢查看图形，就可以将该信息存储在Session对象中。

11．php：PHP（外文名Hypertext Preprocessor，中文名：“超文本预处理器“）是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点，利于学习，使用广泛，主要适用于Web开发领域。PHP独特的语法混合了C、Java、Perl以及PHP自创的语法。它可以比CGI或者Perl更快速地执行动态网页。用PHP做出的动态页面与其他的编程语言相比，PHP是将程序嵌入到HTML（标准通用标记语言下的一个应用）文档中去执行，执行效率比完全生成HTML标记的CGI要高许多；PHP还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快。

12．Js：Javascript是一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为JavaScript引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在HTML（标准通用标记语言下的一个应用）网页上使用，用来给HTML网页增加动态功能。在此是为微信小程序增加动态与交互功能。

13．wxml：wxml（WeiXin Markup Language）是微信框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构。

14．wxss：wxss（WeiXin Style Sheets）是微信的一套样式语言，用于描述WXML的组件样式。WXSS用来决定WXML的组件应该怎么显示。为了适应广大的前端开发者，WXSS具有CSS大部分特性。同时为了更适合开发微信小程序，WXSS对CSS进行了扩充以及修改。

15. Python：Python是一种面向对象的解释型计算机程序设计语言，Python是纯粹的自由软件，源代码和解释器CPython遵循GPL(GNU General Public License)协议。Python语法简洁清晰，特色之一是强制用空白符(white space)作为语句缩进。Python具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起。常见的一种应用情形是，使用Python快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如3D游戏中的图形渲染模块，性能要求特别高，就可以用C/C++重写，而后封装为Python可以调用的扩展类库。

1.4参考资料，参考文献

1. phpMaAdmin定义-百度百科：

<https://baike.baidu.com/item/PhpMyAdmin/9624049?fr=aladdin>

2. MySQL定义-百度百科：

<https://baike.baidu.com/item/mySQL/471251?fr=aladdin>

3.《微信小程序开发入门与实践》清华大学出版社出版 雷磊

4．《软件工程导论》清华大学出版社 朱少民

5．微信小程序-百度百科：

<https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1%E5%B0%8F%E7%A8%8B%E5%BA%8F/20171697?fr=aladdin>

6．MATLAB-百度百科：

<https://baike.baidu.com/item/MATLAB/263035?fr=aladdin>

7．Microsoft Office Excel百度百科：

<https://baike.baidu.com/item/Microsoft%20Office%20Excel/1448703?fr=aladdin>

8．腾讯云-百度百科：

<https://baike.baidu.com/item/%E8%85%BE%E8%AE%AF%E4%BA%91/9905046?fr=aladdin>

9．用户画像-百度百科：

<https://baike.baidu.com/item/%E7%94%A8%E6%88%B7%E7%94%BB%E5%83%8F/22085710?fr=aladdin>

10．概要设计说明书范例及模板-百度文库-By 最爱lxxy

<https://wenku.baidu.com/view/06b1e3ab85868762caaedd3383c4bb4cf7ecb7bd.html>

11．概要设计-百度百科：

<https://baike.baidu.com/item/%E6%A6%82%E8%A6%81%E8%AE%BE%E8%AE%A1/9827718>

12．数据聚类-百度百科：

<https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E8%81%9A%E7%B1%BB/2335124>

13．微信对手机配置要求：

<https://zhidao.baidu.com/question/2009992975776493508.html>

14．数据库管理员定义-百度百科：

<https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E5%91%98/1216449>

15．Session定义-百度百科：

<https://baike.baidu.com/item/session/479100?fr=aladdin>

16．Php定义-百度百科：

<https://baike.baidu.com/item/PHP/9337?fr=aladdin>

17．Js定义-百度百科：

<https://baike.baidu.com/item/javascript/321142?fromtitle=js&fromid=10687961&fr=aladdin>

18．Wxml基本解释-简书

<https://www.jianshu.com/p/015d0e998196>

19．Wxss-微信公众平台

<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxss.html>

20．Python-百度百科：

<https://baike.baidu.com/item/Python/407313?fr=aladdin>

2.总体设计

2.1需求概述

（1）生成问卷功能

前台显示的问题由后台数据库中的指定的问题表动态生成

（2）用户填写问卷功能

用户可查看问卷并对问卷进行填写

（3）收集问卷功能

用户提交后的问卷结果可以上传到腾讯云的云端数据库中

（4）表单验证功能

用户必须将全部问题答完才能提交问卷

（5）管理员登录验证功能

具有管理员权限的用户可登录至小程序后台修改问卷

（6）管理员修改问卷功能

管理员有权限对问卷中的问题进行增删改查等基本操作

（7）管理员发布问卷功能

不同管理员可以发布不同的问卷并记录在数据库中

（8）导入或导出

数据库中的数据均可导入或导出到其他设备中

（9）对于用户的推荐功能

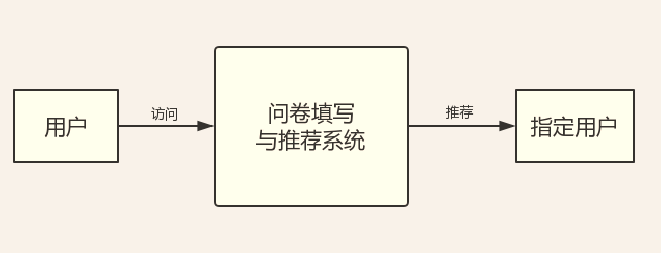
数据库管理员可对数据库中导出的填写问卷结果进行分析，对一个指定的用户分别进行基于用户的协同推荐算法和基于物品的协同推荐算法分析，算出与该用户相关度较高的五个用户并根据这五个用户所选择的物品计算每个物品的权重，选择权重最高的前五个物品对该用户进行推荐

（10） 推送功能

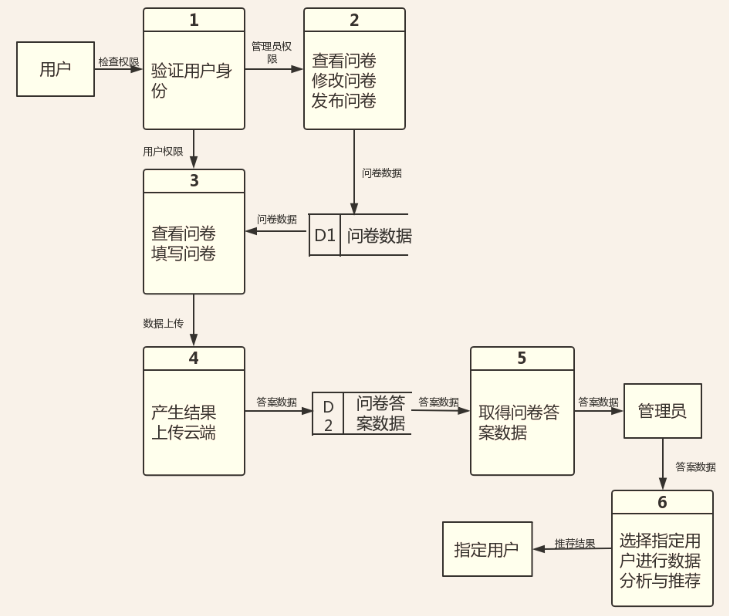
管理员可对数据分析结果以发送邮件的方式进行推送

2.2软件结构

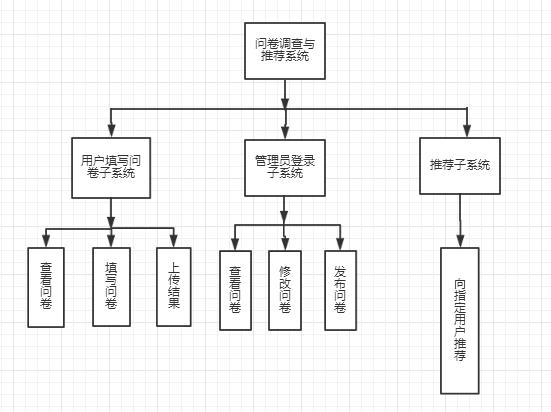
(1)顶层数据流图



（2）二层数据流图



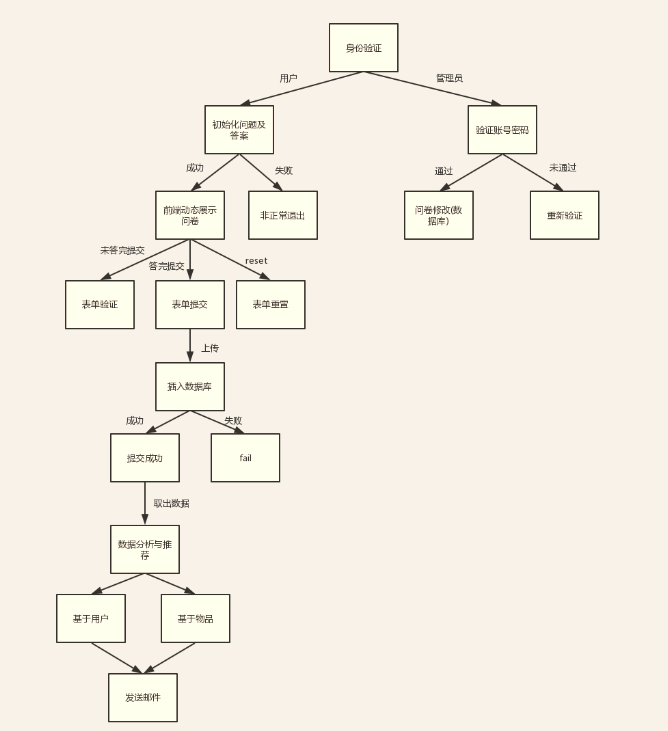
（3）层次方框图



3程序设计说明

1）模块描述

本软件的详细功能模块图如图所示：



2）身份选择模块

功能：用户进行身份选择

输入项目：点击事件

输出项目：用户界面或管理员界面

储存分配：程序运行需要占用内存至多1MB

模块界面：(尚未美化)



软件设计方法(算法)：

<view class="btn-area">

<button bindtap='jinru'>进入调查问卷</button>

<button bindtap='denglu'>管理员登录</button>

</view>

接口：内部接口：微信小程序内部wxml与js的接口bindtap

外部接口：通过扫描小程序码进入该模块

控件名称及其功能：

|  |  |  |
| --- | --- | --- |
| 控件 | 控件名 | 功能 |
| Button | 进入调查问卷 | 连接调查问卷页面 |
| Button | 管理员登录 | 连接管理员登录界面 |

3）初始化问题及答案模块

功能：初始化问卷，显示问题以及备选项，并使系统进入正常工作状态

性能：2秒内完成

输入项目：配置文件、数据库文件

输出项目：产生全局变量，包括问题数组、备选项数组

储存分配：程序运行需要占用内存约1MB

软件设计方法(算法)：

该算法的目的是从数据库中取出管理员已经修改好的问题以及备选项，用于前端渲染。

数据库内容：



1. 取出问题算法：

Js：

wx.request({//取出问题

url: 'https://vnuyheuv.qcloud.la/extractquestion.php',//开发环境

data: {},

header: {

'content-type': 'application/json'

},

success: function (res) {

console.log(res.data)

question = res.data.split("|");//这里将返回的数据按“|”分开存入数组question中

var j=0;

for (var i = 0; i < question.length-1; i++) {//注意这里i的范围

var temp = 'titles[' + j + ']'//这里使array的下标可以动态改变，titles代表问题题目，用来对前台渲染

j++;

that.setData({

[temp]: question[i] //注意：这里temp必须要加[]，至于为什么我也不明白

})

}

}

})

Extractquestion.php：

**<?php**

header("content-type:text/html;charset=utf-8");

$link = mysqli\_connect('localhost', 'root', 'xxx','testresult');

mysqli\_set\_charset($link, "utf8");//!!

if (!$link) {

die('Could not connect: ' . mysql\_error());

echo "fail";

}else {

  $result = mysqli\_query($link,"SELECT \* FROM questions");

  while($row = mysqli\_fetch\_array($result))

{

echo $row['questionname'];

echo "|";

}

mysqli\_close($link);

}

**?>**

1. 取出答案算法：

Js：

wx.request({//取出答案

url: 'https://vnuyheuv.qcloud.la/extractanswer.php',//开发环境

data: {},

header: {

'content-type': 'application/json'

},

success: function (res) {

console.log(res.data)

answer1 = res.data.split("|");//将每个问题的答案存入answer1中(此时选项还未区分)

for(var i=0;i<answer1.length-1;i++){//注意i的范围

answer2 = answer1[i].split(" ");//将每个问题的答案按空格区分，并存入answer2(存每个问题的选项)数组中

for(j=0;j<answer2.length-1;j++){//对某一问题的选项进行遍历

var temp = 'as[' + i + '][' + j + ']'//这里使array的下标可以动态改变，as代表answer，存问题答案，对前台进行渲染

that.setData({

[temp]: answer2[j] //注意：这里item必须要加[]，至于为什么我也不明白

})

}

}

}

})

Extractanswer.php：

**<?php**

header("content-type:text/html;charset=utf-8");

$link = mysqli\_connect('localhost', 'root', 'xxx','testresult');

mysqli\_set\_charset($link, "utf8");//!!

if (!$link) {

die('Could not connect: ' . mysql\_error());

echo "fail";

}else {

   $result = mysqli\_query($link,"SELECT A,B,C,D,E,F,G,H FROM questions");

   while($row = mysqli\_fetch\_array($result))

{

  if($row['A']!=null){

 echo $row['A'];

  echo " ";

}

if($row['A']==null){

echo "1 ";

}

if($row['B']!=null){

echo $row['B'];

echo " ";

}

if($row['C']!=null){

echo $row['C'];

echo " ";

}

if($row['D']!=null){

echo $row['D'];

echo " ";

}

if($row['E']!=null){

echo $row['E'];

echo " ";

}

if($row['F']!=null){

echo $row['F'];

echo " ";

}

if($row['G']!=null){

echo $row['G'];

echo " ";

}

if($row['H']!=null){

echo $row['H'];

echo " ";

}

echo "|";

}

mysqli\_close($link);

}

**?>**

接口：通过wx.request访问腾讯云生产环境中的php脚本文件

4）动态展示问卷模块

功能：将初始化模块执行后所获得的输出项目(全局问题、答案变量)在前端动态渲染

性能：1秒内完成

输入项目：全局变量，包括问题数组、备选项数组

输出项目：前台信息

储存分配：程序运行需要占用内存约1MB

模块界面：



软件设计方法(算法)：

<!-- 展示 -->

<!-- <view class="section section\_gap"> -->

<view wx:for="{{titles}}" wx:for-index="xxx" wx:for-item="timu">

<view class="section\_\_title">{{xxx+1}}.{{titles[xxx]}}</view>

<!-- 输入类题型 -->

<block wx:if="{{titles[xxx][1]=='输'&&titles[xxx][2]=='入'}}">

<input name="{{timu}}" style="background-color: #FFFFFF" placeholder="请在这里输入" />

</block>

<!-- 多选类题型 -->

<block wx:elif="{{titles[xxx][1]=='多'&&titles[xxx][2]=='选'}}">

<checkbox-group name="{{timu}}">

<label wx:for="{{as[xxx]}}">

<checkbox value="{{item}}"/>{{item}}

</label>

</checkbox-group>

</block>

<!-- 单选类题型 -->

<block wx:else>

<radio-group bindchange="bindRadioChange" name="{{timu}}">

<label wx:for="{{as[xxx]}}">

<!-- for只能放在label里，否则会出现单选变为多选..-->

<radio value="{{item}}"/>{{item}}

</label>

</radio-group>

</block>

流程逻辑：

本模块利用两重循环实现前台动态渲染；

首先，第一重循环循环全局变量{{titles}}，对于每个标题titles[xxx]，判断其题型，如果为输入题型，则进入显示输入类题型模块，如果为多选题型，则进入显示多选类题型模块，如果为单选类题型，则进入显示单选类题型模块。以此类推。

其次，选定题目类型后进行第二重循环，循环全局变量{{as[xxx]}}，（这里as[xxx]对应的所有值即为该题的备选项）显示该问题备选项。

伪码：

START

Wx:for(titles){

Wx:if(titles[xxx] == ‘输入’){

Display : <input>

}

Wx:eIif(titles[xxx] == ‘多选’){

Wx:for(as[xxx]){

Display : <checkbox value=”{{item}}”>

}

}

Wx:else(titles[xxx] == ‘单选’){

Wx:for(as[xxx]){

Display : <radio value=”{{item}}”>

}

}

}

STOP

控件名称及其功能：

|  |  |  |
| --- | --- | --- |
| 控件 | 控件名 | 功能 |
| View | 循环问题 | 动态循环展示数据库中的问题 |
| Block | 判断输入 | 判断当前问题是否为输入类问题 |
| Input | 输入框 | 为输入类题型提供输入框 |
| Block | 判断多选 | 判断当前问题是否为多选类问题 |
| Checkbox-group | 多选项目组 | 提供多个多选框 |
| Label | 循环多选类项目答案 | 将该多选类题型的答案动态输出 |
| Checkbox | 多选框 | 提供一个多选框 |
| Block | 判断单选 | 判断当前问题是否为单选类问题 |
| radio-group | 单选项目组 | 提供多个单选框 |
| label | 循环单选类项目答案 | 将该单选类题型的答案动态输出 |
| radio | 单选框 | 提供一个单选框 |

接口：用到了wxml与js间的内部接口，以数据传递方式实现

5）初始化失败非正常退出模块

功能：关闭并退出系统

输入项目：无

输出项目：无

流程逻辑：退出

接口：微信小程序内部提供

6）表单验证模块

功能：当用户提交未答完的问卷时，拒绝提交

性能：用户点击提交后在毫秒内完成

输入项目：点击事件

输出项目：提示框Modal

模块界面：



流程逻辑：

START

If(‘用户提交了完整问卷’){

Wx.request;

Wx.navigateTo;

…

}

Else{

Wx.showModel;

}

STOP

接口：Wx.showModel;

7）表单提交模块

功能：将用户提交的数据上传至云端服务器中对应的php脚本文件

性能：1秒内完成

输入项目：点击事件

输出项目：e.detail.value(用户提交的数据)、提交成功页面

储存分配：程序运行需要占用内存至多1MB

模块界面：



软件设计方法(算法)：

前台：

<view class="btn-area">

<button form-type="submit" name="submit">Submit提交</button>

<button form-type="reset">Reset重置</button>

</view>

Js：

formSubmit: function (e) {

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

var formData = e.detail.value;//存前台传来的数据，这里formData为object类型

for (k in formData) {//遍历前台提交的数据并存入数组，遍历object类型时，只能用这种遍历方法

console.log(k + "：" + formData[k]);

mysubmit[i] = "'"+formData[k]+"'";//将每个问题答案存入mysubmit数组中，以实现将数据动态传入后台

i++;

}

this.setData({

mysubmit: mysubmit,

})

wx.request({

url: 'https://924010491.joesgod.club/newinsert.php',//生产环境

data:

{

mysubmit,

},

method: 'POST', // OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT

header: {

'content-type': 'application/x-www-form-urlencoded; charset=UTF-8'

},

success: function (res) {

var err = res.data.error

if (err) {

\_this.setData({

error: err

})

}

else {

console.log("返回成功的数据:" + res.data) //返回的会是对象，可以用JSON转字符串打印出来方便查看数据

console.log("返回成功的数据:" + JSON.stringify(res.data)) //这样就可以愉快的看到后台的数据啦

}

},

fail: function () {

// fail

},

complete: function () {

// complete

}

})

wx.navigateTo({

url: '/pages/tijiaohou/tijiaohou',

})

控件名称及其功能：

|  |  |  |
| --- | --- | --- |
| 控件 | 控件名 | 功能 |
| Button | 提交 | 连接Js中提交函数 |

接口：wx.request，wx.navigateTo

8）表单重置模块

功能：重置所有已填的问题答案

性能：1秒内完成

输入项目：点击事件

输出项目：初始界面

模块界面：



软件设计方法(算法)：

//表单重置按钮

formReset: function (e) {

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

this.setData({

allValue: ''

})

wx.pageScrollTo({

scrollTop: 0,

})

},

控件名称及其功能：

|  |  |  |
| --- | --- | --- |
| 控件 | 控件名 | 功能 |
| Button | 重置 | 连接Js中重置函数 |

接口：wx.pageScrollTo

9）接收并插入数据至数据库模块

功能：接收并插入数据至数据库

输入项目：mysubmit(用户上传的数据)

输出项目：数据库中的一条数据

储存分配：至多占用内存1M

软件设计方法(算法)：

**<?php**

header("content-type:text/html;charset=utf-8");

//echo "enter";

$link = mysqli\_connect('localhost', 'root', 'xxx','testresult');

mysqli\_set\_charset($link, "utf8");//编码转换至utf-8

if (!$link) {

die('Could not connect: ' . mysql\_error());

echo "fail";

}else {

foreach($\_POST as $key=>$val) {//事先不知道传过来的数值有几个采用的方法

echo $key.": ".$val."<br>\n";

}

$query = "insert into newinsert values(".$val.")";//拼接

$result=mysqli\_query($link, $query);

echo "$result";

}

**?>**

流程逻辑：

START

$link = mysqli\_connect;

foreach($\_POST as $key=>$val){//接收数据

echo $key.": ".$val."<br>\n";

}

$query = "insert into newinsert values(".$val.")";

$result=mysqli\_query($link, $query);//插入数据

STOP

接口：wx.request

10）提交成功模块

功能：提示用户数据提交成功

性能：1秒内完成

输入项目：无

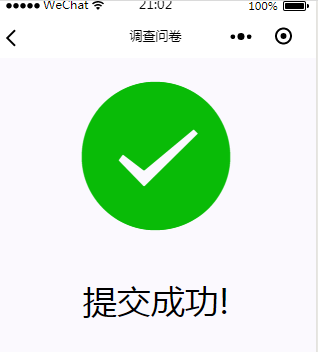
输出项目：提交成功页面

模块界面：

点击前：



点击并成功提交后：



软件设计方法(算法)：

Index.js:

wx.navigateTo({

url: '/pages/tijiaohou/tijiaohou',

})

Tijiaohou.wxml:

<!--pages/tijiaohou/tijiaohou.wxml-->

<!--成功图标-->

<view class='imageCenter'>

<icon type="success" size="150"/>

</view>

<view class="viewTitle">

<text class="view-Name" style='font-size:80rpx'>提交成功!</text>

</view>

控件名称及其功能：

|  |  |  |
| --- | --- | --- |
| 控件 | 控件名 | 功能 |
| Icon | 图标 | 提供一个成功图标 |
| Text | 文本 | 展示“提交成功！”文本 |

接口：wx.navigateTo

11）提交失败模块

功能：提示插入数据库失败

输入项目：点击事件

输出项目：提示页面

模块界面：

点击前：



点击并提交失败：



软件设计方法(算法)：

Index.js:

wx.navigateTo({

url: '/pages/tijiaohoufail/tijiaohoufail',

})

Tijiaohou.wxml:

<!--pages/tijiaohoufail/tijiaohoufail.wxml-->

<!--成功图标-->

<view class='imageCenter'>

<icon type="cancel" size="150"/>

</view>

<view class="viewTitle">

<text class="view-Name" style='font-size:80rpx'>提交失败!</text>

</view>

控件名称及其功能：

|  |  |  |
| --- | --- | --- |
| 控件 | 控件名 | 功能 |
| Icon | 图标 | 提供一个失败图标 |
| Text | 文本 | 展示“提交失败！”文本 |

接口：wx.navigateTo

12）基于用户的协同推荐模块

功能：用基于用户的相关度分析算法向用户推荐

输入项目：指定用户id

输出项目：对该用户推荐的货物id

软件设计方法(算法)：

import xlrd

import math

data = xlrd.open\_workbook("D:\\企划与项目开发 微信小程序\\推荐\\123.xls") #将excel表格导入

sheets = data.sheets()

sheet\_1\_by\_name = data.sheet\_by\_name(u'Sheet1')

userID = list(map(int, sheet\_1\_by\_name.col\_values(2))) #将excel中用户编号列导入

ProductID = list(map(int, sheet\_1\_by\_name.col\_values(6))) #将excel中产品编号导入

item = userID[0]

i = 0

length = len(userID)

relation = {}

lists = []

j = 0

while i < length:

if item == userID[i]:

lists.append(ProductID[i])

i += 1

else:

relation[userID[i-1]] = lists

item = userID[i]

lists = []

continue

relation[userID[i-1]] = lists #以上将该用户ID所购买的所有产品种类对应关系放入relation字典中

#print(relation[userID[0]])

print("请输入想推荐的用户id(1-302)")

uid = input("请输入想推荐的用户id(1-302)");

goods\_list = list(relation[userID[int(uid)-1]])

print(goods\_list)

z = 0

while z < len(goods\_list):

goods\_list[z] = int(goods\_list[z])

z += 1

similarityTwo = {}

for key3 in relation.keys():

similarityTwo[key3] = len(set(goods\_list) & set(relation[key3])) \

/ math.sqrt((len(set(goods\_list))) \* len(set(relation[key3])))

#以上为计算两用户间的相似度确定邻居

fff = zip(similarityTwo.values(), similarityTwo.keys())

ggg = sorted(fff)

#按照相似度排列，从低到高

qqqqTwo = 0

tagtagTwo = 1

kkTwo = 5

#print(ggg.\_\_len\_\_())

recommend\_set\_Two = set()

neighbors = {}

#print(ggg)

while qqqqTwo < kkTwo and tagtagTwo <= len(ggg):

# print(ggg.\_\_len\_\_()-tagtagTwo)

if ggg[ggg.\_\_len\_\_()-tagtagTwo][0] == 1.0:

# print("1")

tagtagTwo += 1

continue

elif len(set(relation[int(ggg[ggg.\_\_len\_\_() - tagtagTwo][1])]) -

(set(relation[int(ggg[ggg.\_\_len\_\_() - tagtagTwo][1])]) & set(goods\_list))) == 0:

tagtagTwo += 1

# print("2")

continue

else:

# print("3")

recommend\_set\_Two = recommend\_set\_Two | (set(relation[int(ggg[ggg.\_\_len\_\_() - tagtagTwo][1])]) -

(set(relation[int(ggg[ggg.\_\_len\_\_() - tagtagTwo][1])]) &

set(goods\_list)))

neighbors[int(ggg[ggg.\_\_len\_\_() - tagtagTwo][1])] = ggg[ggg.\_\_len\_\_() - tagtagTwo][0]

qqqqTwo += 1

tagtagTwo += 1

#取出相似度不为1的用户中相似度最高的kkTwo个用户

#print(recommend\_set\_Two)

recommend\_set\_Two = list(recommend\_set\_Two)

#print(recommend\_set\_Two)

if len(recommend\_set\_Two) == 0:

print("您买的物品基本覆盖了全部商品，没有为您推荐的物品")

#print("邻居", neighbors)

index = 0

dict\_Three = {}

while index < len(recommend\_set\_Two):

dict\_Three[recommend\_set\_Two[index]] = 0.0

index += 1

for indexFour in relation:

relation[indexFour] = list(set(relation[indexFour]))

indexTwo = 0

for indexOne in neighbors:

while indexTwo < len(relation[int(indexOne)]):

for indexThree in dict\_Three:

if relation[int(indexOne)][indexTwo] == indexThree:

dict\_Three[indexThree] += neighbors[indexOne]

break

indexTwo += 1

indexTwo = 0

#计算这五个用户中所有的购买的商品的权重

#print(dict\_Three)

ffffff = zip(dict\_Three.values(), dict\_Three.keys())

gggggg = sorted(ffffff)

#print(gggggg)

wer = 0

list\_range = []

while wer < len(gggggg):

list\_range.append(gggggg[wer][1])

wer += 1

#print(list\_range)

list\_range.reverse()

#print(list\_range)

kkk = 5

if len(list\_range) > kkk:

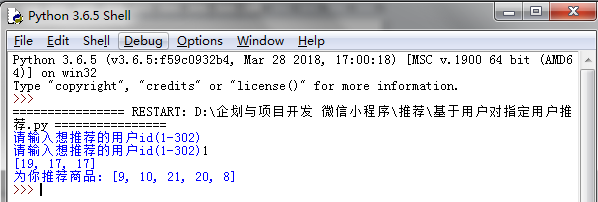
list\_range = list\_range[0:kkk]

#取出权重最大的五个商品进行推荐

print("为你推荐商品:", list\_range)

#print(str(list\_range))

运行界面：



流程逻辑：

START

#导入数据

#以上将每个用户ID所购买的所有产品种类对应关系放入relation字典中,例如：顾客1购买了L,J,J 则对应relation[1]={L,J,J}

#输入想要推荐用户的id，通过relation字典找出其买过的商品

#通过余弦相似度公式计算该用户与其他每个用户间的相似度

#排序，找出和该用户最高相似度的五个用户

#找出五个用户与需要推荐的用户所购买物品的差集

#计算上一步所求差集中每个物品的权重

#(note：权重计算方式：例 1：A，B 2：A，B，C 3：A，B，C，D；向1用户进行推荐，则C的权重为2和3用户的相似度相加、D的权重为3的相似度)

#取出权重最大的五个商品进行推荐

STOP

例：

用户 1：A,B

2：A,B,C

3：A,B,C,D

根据原矩阵计算每个用户及其购买的商品，并存入relation字典

输入一个用户id，例如 1，向其推荐一个商品

利用余弦公式计算1用户与其他用户的相关度，例：

1->2 : 0.2

1->3 : 0.3

因为3的相关度更高，所以选择3用户的商品对1用户进行推荐

计算3用户商品与1用户商品的差集，为 C,D

计算C,D权重为 0.3 0.3

选择权重最大的商品进行推荐，因为权重相同，所以即可推荐C也可D

13）基于物品的协同推荐模块

功能：用基于物品的相关度分析算法向用户推荐

输入项目：指定用户id

输出项目：推荐的商品id

软件设计方法(算法)：

import math

import xlrd

data = xlrd.open\_workbook("D:\\企划与项目开发 微信小程序\\推荐\\123.xls") #将excel表格导入

sheets = data.sheets()

sheet\_1\_by\_name = data.sheet\_by\_name(u'Sheet1')

userID = list(map(int, sheet\_1\_by\_name.col\_values(2))) #将excel中用户编号列导入

#print(userID)

ProductID = list(map(int, sheet\_1\_by\_name.col\_values(6))) #将excel中产品编号导入

#print(ProductID)

#print(str(userID[0]))

item = userID[0]

i = 0

length = len(userID)

relation = {}

lists = []

j = 0

while i < length:

if item == userID[i]:

lists.append(ProductID[i])

i += 1

else:

relation[userID[i-1]] = lists

item = userID[i]

lists = []

continue

relation[userID[i-1]] = lists #以上将该用户ID所购买的所有产品种类对应关系放入relation字典中

i = 0

dict\_prod\_popul = {}

while i < len(userID): #计算买各个产品的用户数量及次数,有重复

if(ProductID[i] not in dict\_prod\_popul.keys()):

dict\_prod\_popul[ProductID[i]] = []

dict\_prod\_popul[ProductID[i]].append(userID[i])

else:

dict\_prod\_popul[ProductID[i]].append(userID[i])

i += 1

#print(dict\_prod\_popul)

for index in dict\_prod\_popul: #计算买各个产品的用户数量，去处重复

dict\_prod\_popul[index] = list(set(dict\_prod\_popul[index]))

#print(dict\_prod\_popul)

#print(len(set(ProductID)))

similarity = {} #计算两两物品间的相似度，例(1,2)->value

for index\_one in dict\_prod\_popul:

for index\_two in dict\_prod\_popul:

if index\_one == index\_two:

continue

similarity[index\_one, index\_two] = len(set(dict\_prod\_popul[index\_one]) & set(dict\_prod\_popul[index\_two])) / \

math.sqrt(len(dict\_prod\_popul[index\_one]) \* len(dict\_prod\_popul[index\_two]))

#print(similarity)

print("请输入想推荐的用户")

uid = input("请输入用户编号(1-302):")

goods\_list = list(relation[userID[int(uid)-1]])

print(goods\_list)

#goods\_list = list(goods\_list.split(','))

i = 0

while i < len(goods\_list):

goods\_list[i] = int(goods\_list[i])

i += 1

#print(goods\_list)

#goods\_list = [1, 2, 3, 4, 5, 6]

#print(goods\_list)

i = 0

k = 5

#inverse\_similarity = {}

#for index\_three in similarity:

# inverse\_similarity[similarity[index\_three]] = list(index\_three)

#print(inverse\_similarity)

find = {} #将这些物品所有的所有有关的相似度相加，找出与这些物品最相似的商品

while i < len(goods\_list):

for index\_four in similarity:

if goods\_list[i] == index\_four[0]:

if index\_four[1] not in find:

find[index\_four[1]] = similarity[index\_four]

else:

find[index\_four[1]] += similarity[index\_four]

i += 1

#print(find)

#print(len(find))

f = zip(find.values(), find.keys()) #将权重从小到大排序

g = sorted(f)

#print(g)

#print(len(g))

#取出排序最高的k个

recommend = []

i = 1

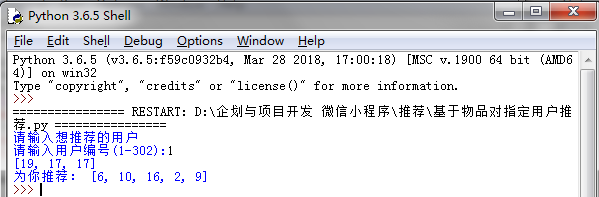
while i <= k:

recommend.append(g[len(g)-i][1])

i += 1

print("为你推荐：", recommend)

界面截图：



流程逻辑：

START

#导入数据

#以上将每个用户ID所购买的所有产品种类对应关系放入relation字典中,例如：顾客1购买了L,J,J 则对应relation[1]={L,J,J}

#找出购买各个产品的用户并存放在数组中

#利用上面得到的数组以及余弦相似度公式，将数组里面的元素两两计算相似度，存入字典similarity中

#输入想要推荐用户的id，通过relation字典找出其买过的商品

#将该用户购买的商品所有有关的商品相似度相加，找出与这些商品相似度最高的五个商品

#进行推荐

STOP

举例：

用户 1：A,B

2：A,B,C

3：A,B,C,D

求逆：

A：1,2,3

B：1,2,3

C：2,3

D：3

根据原矩阵计算每个用户及其购买的商品，并存入relation字典

根据逆矩阵计算A,B,C,D两两之间的相似度。

(A,B)->0.1;

(A,C)->0.2;

(A,D)->0.3;

(B,C)->0.4;

(B,D)->0.5;

(C,D)->0.6;

假如求向用户1推荐一个商品

计算 除A,B商品外，其他和A,B商品有关的商品的累积相似度，例：

C->0.2+0.4

D->0.3+0.5

选择一个最大相似度的商品进行推荐，即推荐D商品

14）发送邮件模块

功能：向推荐对象发送邮件

输入项目：推荐内容

输出项目：一封邮件

软件设计方法(算法)：

import smtplib

from email.mime.text import MIMEText

from email.header import Header

sender = 'from@runoob.com'

receivers = ['xxxxxxxxx@qq.com']

# 接收邮件，可设置为你的QQ邮箱或者其他邮箱 # 三个参数：第一个为文本内容，第二个 plain 设置文本格式，第三个 utf-8 设置编码

message = MIMEText('Python 邮件发送测试...', 'plain', 'utf-8')

message['From'] = Header("菜鸟教程", 'utf-8') # 发送者

message['To'] = Header("测试", 'utf-8') # 接收者

subject = 'Python SMTP 邮件测试'

message['Subject'] = Header(subject, 'utf-8')

try:

smtpObj = smtplib.SMTP('localhost')

smtpObj.sendmail(sender, receivers, message.as\_string())

print "邮件发送成功"

except smtplib.SMTPException:

print "Error: 无法发送邮件"

接口：SMTP邮件协议

15）验证账号密码模块

功能：验证管理员账号与密码

输入项目：输入用户名和密码

输出项目：管理员权限页面

储存分配：程序运行需要占用内存至多1MB

模块界面：

提交前：



正确提交后：



软件设计方法(算法)：

前台：

<view>

<text>管理员登录</text>

</view>

<form bindsubmit="formSubmit" bindreset="formReset">

<block>

<text>管理员账号：</text>

<input name="zhanghao" placeholder="请在这里输入账号" />

<text>管理员密码：</text>

<input name="mima" placeholder="请在这里输入密码" />

</block>

<view>

<button form-type="submit">登录</button>

<button form-type="reset">重置</button>

<button bindtap='return'>返回上层</button>

</view>

<view>

<label>{{tishi}}</label>

</view>

</form>

Js：

formSubmit: function (e) {

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

var formData = e.detail.value;//存前台传来的数据，这里formData为object类型

var mysubmit = new Array();//mysubmit用来存放传入后台的数据

var i,k;

i = 0;

for (k in formData) {//遍历前台提交的数据并存入数组，遍历object类型时，只能用这种遍历方法

console.log(k + "：" + formData[k]);

mysubmit[i] = "'" + formData[k] + "'";//将账号密码传入mysubmit数组中

i++;

}

var account;//账号

var password;//密码

var item;

var \_this=this;

account=mysubmit[0];

password=mysubmit[1];

wx.request({

url:'https://vnuyheuv.qcloud.la/verifyAdminAccount.php',

data:{

account:account

},

method: 'POST',

header: {

'content-type': 'application/x-www-form-urlencoded; charset=UTF-8'

},

success: function (res) {

console.log(res.data);

item=res.data;

if (item== password) {

console.log("登录成功");

getApp().globalData.AdminAccount=account;

error="登录成功";

wx.navigateTo({

url: '/pages/control/control',

})

}

})

},

formReset:function(e){

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

this.setData({

allValue: ''

})

},

return:function(e){

console.log('返回首页事件发生')

wx.navigateBack({

delta: 1

})

}

后台：

**<?php**

header("content-type:text/html;charset=utf-8");

$account=$\_POST["account"];

$servername = "localhost";

$username = "root";

$password = "xxx";

$dbname = "testresult";

// 创建连接

$conn = new mysqli($servername, $username, $password,$dbname);

// 检测连接

if ($conn->connect\_error) {

die("连接失败: " . $conn->connect\_error);

}

$sql = "SELECT password FROM administrator WHERE account=".$account.";";

$result = $conn->query($sql);

if (mysqli\_num\_rows($result) > 0) {

// 输出数据

while($row = mysqli\_fetch\_assoc($result)) {

echo "'".$row["password"]."'";

}

} else {

echo "账号错误";

}

**?>**

流程逻辑：

START

Input account;

Input password;

If(Submit)

Wx.request({

Upload account;

Return item;

})

If(item == password){

wx.navigateTo;

}

Else if(reset){

setData : allValue: ''

}

Else{

wx.navigateBack;

}

STOP

接口：

wx.request 、wx. navigateTo、wx.navigateBack、前台和后台的内部函数接口

16）重新验证模块

功能：对未通过验证的管理员重新验证用户名和密码

输入项目：输入用户名和密码

输出项目：无

储存分配：程序运行需要占用内存至多1MB

软件设计方法(算法)：

success: function (res) {

console.log(res.data);

item=res.data;

var error;

if (item == "账号错误") {

console.log("账号不存在");

error="账号不存在";

}

else {

if (item== password) {

console.log("登录成功");

getApp().globalData.AdminAccount=account;

error="登录成功";

wx.navigateTo({

url: '/pages/control/control',

})

}

else{

console.log("密码错误");

error = "密码错误";

}

}

\_this.setData({

tishi: error

})

}

流程逻辑：

START

Input account;

Input password;

If(Submit)

Wx.request({

Upload account;

Return item;

})

If(item == ‘账号错误’){

error="账号不存在";

}

Else{

if (item== password) {

wx.navigateTo;

}

else{

error = "密码错误";

}

}

Else if(reset){

setData : allValue: ''

}

Else{

wx.navigateBack;

}

STOP

接口：

wx.navigateTo、wx.navigateBack、前台和后台的内部函数接口

17）问卷修改模块

功能：管理员对问卷进行管理

输入项目：修改内容

输出项目：新问卷

软件设计方法(算法)：

现已经可以在PhpMyAdmin中修改问题。

管理员在前台可视化修改问题正在玩命开发中…

流程逻辑：

1.通过wx.request接口提取最新问卷的问题以及备选项在前台以变量显示

2.管理员通过键盘输入对显示的问题进行增删改操作

3.管理员提交新问卷

接口：wx.request等