北京工业大学

2018-2019学年 第1学期

信息学部实验学院

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称： | 软件工程导论 | | |
| 报告性质： | □作业报告 █实验报告 | | |
| 学号： | 17080304 | 姓名： | 张文婷 |
| 任课教师： | 于学军 | 课程性质： | 学科基础必修课 |
| 学分： | 2 | 学时： | 48 |
| 班级： | 170803 | 成绩： |  |
| 小组成员： | 17080219 吴优17080318金戈慧17080315 张潇 | | |
| 教师评语： |  | | |

2019 年 6 月 29 日

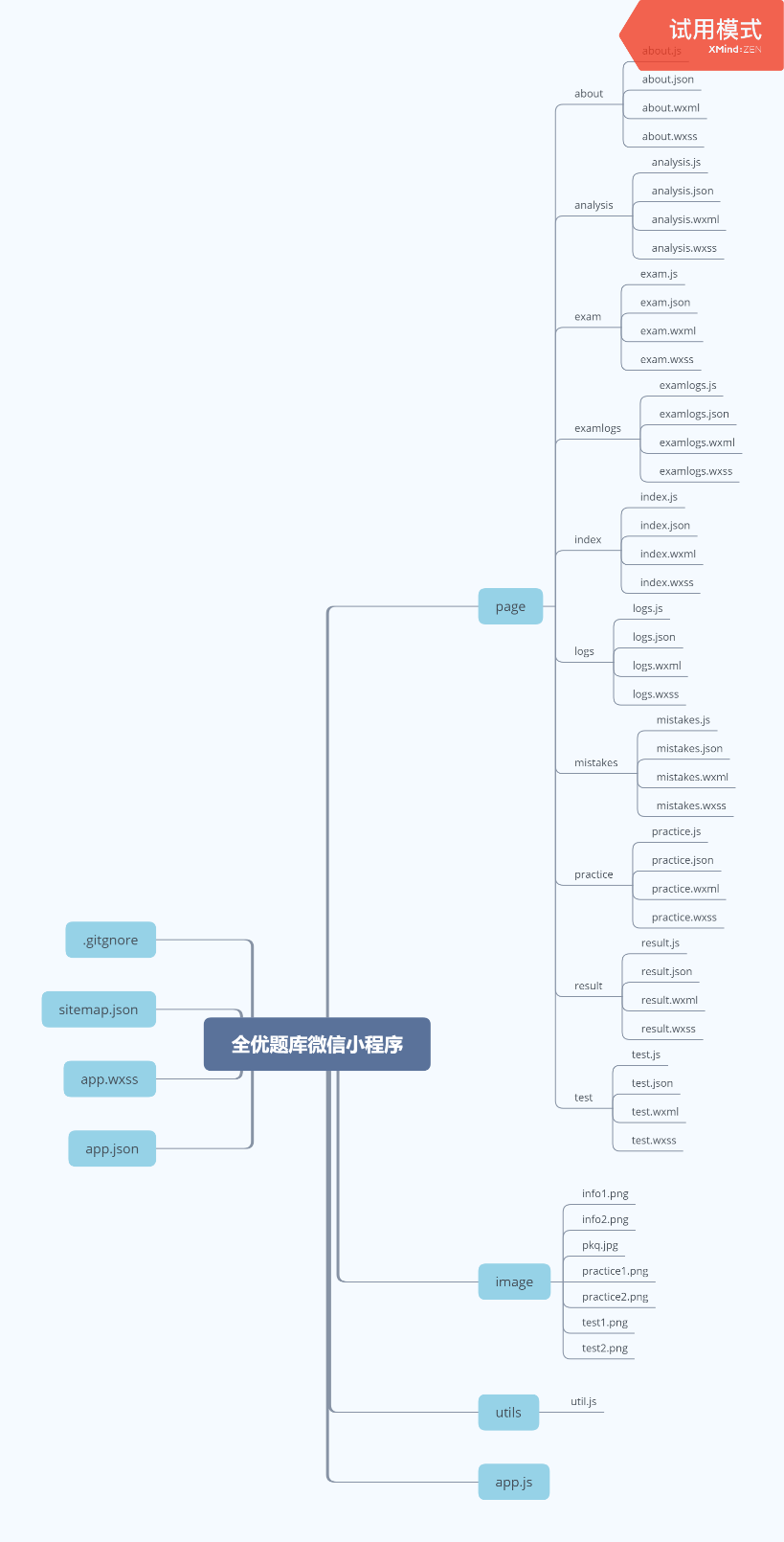
# 实验内容要求

通过综合性课程设计熟悉软件工程、程序设计开发、敏捷实践，软件测试，熟悉敏捷迭代开发方式。

# 全优题库微信小程序需求定义总结

全优题库需满足用户的各种刷题相关需求：练习模式下用户可根据自身情况选择科目、难度、题型，及时判定用户答案并查看解析；测验模式下用户选择科目后系统自动组卷，题型占比合理，答题时具有倒计时功能，规定时间内可手动交卷，超过规定时间则自动交卷，答题结束系统自动判分，显示正确率等具体成绩情况，亦可查看解析，每次测试都将被记录，可由用户查看或手动清除；任何模式下，任何题目皆可加入错题本或移除。全优题库的界面美观， 具有背景音乐，人机交互良好，每道题都带有区别于背景色的题目框与答题框，易于用户作答。

# 全优题库微信小程序架构定义及设计总结



全优题库总体分为三级页面：

一级页面：

练习模式 pages/index/index 功能：选择科目，题目的类型和难度并进入练习模式。

测试模式 pages/test/test 功能：选择科目，进入测试模式

我的信息 pages/info/info 功能：进入关于页面，登录记录页面，错题本页面，测试记录页面的页面。

二级页面：

练习页面 pages/practice/practice 功能：做题练习，进入查看解析的页面

测试页面 pages/exam/exam 功能：测试练习，倒计时，提交答卷

关于页面 pages/about/about 功能：展示软件版本

错题本页面 pages/mistakes/mistakes 功能：查看用户加入的错题

测试记录页面 pages/examlogs/examlogs 功能：查看用户以往测试记录

三级页面：

结果页面 pages/result/result 功能：在测试结束之后显示用户的答题结果

解析页面 pages/practice/practice 功能：练习模式下用户查看道题的正确答案和解析

# 全优题库微信小程序部分代码实现总结

## 传值部分

问题：涉及页面跳转时需要用到上一页的数据，不知道如何获取?

解决方法：使用url跳转传值方法：

wx.navigateTo({

url: '/pages/result/result?ans\_tester=' + that.data.ans\_tester + '&answer=' + that.data.ans + '&correct=' + that.data.correct + '&error=' + that.data.error + '&ration=' + that.data.ration + '&analysis=' + that.data.analysis,

})

在下一页使用带参数的onLoad函数加载:

onLoad: function (options) {

this.setData({

correct: options.correct - 0,

error: options.error - 0,

ration: options.ration - 0,

})

}

问题：当传值对象为一维数组时，url传值方法会默认将类似{c,c,c,4096}的一维数组转化为一个元素‘c,c,c4096’赋值给新页面的元素

解决方法：使用setStorage异步缓存方法，如果异步不可以，可以使用同步缓存方法

wx.setStorage({

key: ‘IndexS’，

data: this.data.IndexS

})

需要使用元素的界面用getStorageSync方法获取，用异步获取的方法有时会出问题

Var value= getStroageSync（‘IndexS’）

问题：当希望传入动态变化的数组作为缓存传递对象时，setdata不能动态的增加数组的长度

解决方法：使用unshift方法在缓存时就改变数组的长度

var tmp = wx.getStorageSync('indexS') || []

tmp.unshift(that.data.indexS)

wx.setStorageSync('indexS', tmp)

在新页面实时获取长度改变的数组：

var value1=wx.getStorageSync('indexS');

问题：当WXML中使用for循环，for中可迭代的变量长度是变化时，且每一个迭代变量都对应一个按钮，js中不可能动态的改变按钮对应的函数个数。

解决方法：使用按钮传参的方法，在按钮中设置属性，当被触发时，使用带参数的触发函数，根据条件解决状态。

addToMis: function(options){

var t = options.currentTarget.id;

if (t < 3){

…

} else if (t >= 3 && t < 6) {

…

} else {

…

}

## 赋值部分

问题：对于数组setdata内部赋值会不识别数组类似a[1]的形式

解决方法：设计一个新的变量间接赋值

var id\_t = "id[" + 0 + "]";

var title\_t = "title[" + 0 + "]";

var options\_tA = "optionAs[" + 0 + "]"; //设置字符串变量

var options\_tB = "optionBs[" + 0 + "]";

var options\_tC = "optionCs[" + 0 + "]";

var options\_tD = "optionDs[" + 0 + "]";

var ans\_t = "ans[" + 0 + "]";

var analysis\_t = "analysis[" + 0 + "]";

that.setData({

[id\_t]: that.data.items[0].\_id,

[title\_t]: that.data.items[0].title,

[options\_tA]: that.data.items[0].A, //用中括号找字符串所代表的的变量

[options\_tB]: that.data.items[0].B,

[options\_tC]: that.data.items[0].C,

[options\_tD]: that.data.items[0].D,

[ans\_t]: that.data.items[0].answer,

[analysis\_t]: that.data.items[0].analysis

})

问题：在设计删除用户测试记录时，将examlogs数组设为空，只在瞬间有效，下一次测试时之前的记录会重新出现，使用removeStorage方法是同样的结果

解决方法：使用wx.clearStroage方法，配合app.launch()方法，清空本地内存，如果不调用app。Launch，则无法向缓存中写入值。

问题：数字无法做加减法，在变量运算中自动将数字识别为字符串

解决方法：单纯赋值中，可以使用变量-0的方法。加减运算中可以调用Number方法。

this.setData({

correct: options.correct - 0,

error: options.error - 0,

ration: options.ration - 0,

})

var afterCu = Number(9 \* that.data.index\_S) + 1;

问题：小程序的input输入框，在页面刷新后框中内容并不清空

解决方法：对input的属性value进行数据绑定，在每次激活按钮刷新时，动态对input框刷新

<input class='input-style' placeholder="输入你的答案" focus="true" value = "{{value}}"/>

goNext: function () {

var that = this;

this.setData({

value: ''

})

}

问题：当onLoad函数带参数时，刷新页面是不一定会触发对应的点击事件，意味着onLoad的参数不一定存在，setdata函数发生错误

解决方案：使用this.onShow()函数代替this.onLoad()函数

## 数据库

问题：在向数据库获取内容的get函数内赋值，小程序会报错xxx is undefined

解决方法：进入函数内部this指针的指向会发生变化，因此在函数外部定义一根指针指向this，函数内部直接引用自定义指针代替this指针

var that = this;

问题：数据库每次只能同时返回20条记录，在测试中需要一次性显示很多条信息

解决方法：使用limit方法，将同时返回的数据上限调高

db.collection('mistakes').limit(1000).get({

success: function(res){

that.setData({

items: res.data,

items\_length: res.data.length

})

}

})

问题：使用后端将数据写入数据库时没有upsert的方法，同样的数据会反复插入

解决方法：每一次将\_id保存，插入时不使用默认\_id,而是已有\_id插入，同样的数据不会被反复插入了。

db.collection('mistakes').add({

// data 字段表示需新增的 JSON 数据

data: {

\_id: that.data.id,

A: that.data.A,

B: that.data.B,

C: that.data.C,

D: that.data.D,

title: that.data.title,

analysis: that.data.analysis,

answer: that.data.answer,

type: 0

},

})

问题：错题本中可以查看解析，做题的时候也可以查看解析，但是在错题本中不能再次显示添加错题本的功能

解决方法：在数据库集合中添加一条属性，区分错题本的错题和错题时的错题type，用wx：if方法或hidden的属性根据type值对按钮进行隐藏

<button bindtap='addToMis' class='weui-btn' wx:if='{{type==0}}'>加入错题本</button>

<button bindtap='goBack' class='weui-btn' wx:if='{{type==1}}'>返回</button>

## 页面跳转：

问题：由于逻辑原因，不允许用户在访问下一页之后仍然可以回转这一页

解决方法：使用redirectTo方法代替navigateTo方法。

wx.redirectTo({

url: '/pages/result/result?ans\_tester=' + that.data.ans\_tester + '&answer=' + that.data.ans + '&correct=' + that.data.correct + '&error=' + that.data.error + '&ration=' + that.data.ration + '&analysis=' + that.data.analysis,

})

问题：由于跳转页面被加入tabBar，无法通过navigateTo的方法进行跳转

解决方法：通过SwitchTab或者navigateBack的方法解决

goBack: function () {

wx.navigateBack({

})

}

问题：由于新加页面，对于一个只需要返回一次的页面，返回两次

解决方法：使用delta属性使得navigateTo的页面回跳俩次

reset: function(){

wx.navigateBack({

delta: 2

})

},

问题：在测验模式中设置计时器，点了提交按钮或者返回仍然计时，时间到了之后仍然执行步骤

解决方法：在onUnLoad的函数以及提交点击事件中加入clearInterval()函数停止计时器的计时

## 引用部分：

问题：使用全局变量的时候，发现没有权限

解决方法：加入const app = getApp()

问题：引用某一个页面的js函数，发现没有权限

解决方法：通过const util = require(' ');引入所需要文件

问题：引用路径是发现绝对路径无效

解决方法：使用相对路径通过../的方法实现向上层跳转

## app.wxss

/\*设置容器格式\*/

.container {

height: 100vh;

width: 100%;

flex-direction: column;

align-items: center;

padding: 20rpx 0;

box-sizing: border-box;

}

/\*设置存放背景图片的容器\*/

.background{

width: 100%;

height: 100%;

position: absolute;

left: 0;

top: 0;

z-index: -1;

position: fixed;

}

/\*设置背景图片的显示格式\*/

.backimg{

width: 100%;

height: 100%;

position: absolute;

z-index: -1;

}

/\*重新设置按钮样式\*/

.weui-btn{

background-color: rgb(112,128,144);

color: rgba(241, 251, 255, 0.808);

}

.weui-btn:after{

border: none;

}

/\*设置输入栏格式\*/

.input-style{

background-color: rgba(255,255,255,0.8);

border-bottom: 20rpx;

border-top: 20rpx;

width: 100%;

}

/\*设置显示测验结果的容器样式\*/

.showResult{

display: block;

justify-content: center;

align-items: center;

text-align: center;

background-color: rgb(62,67,89);

color: white;

margin-top: 0;

margin-bottom:20rpx;

}

/\*设置存放题目及答案的容器样式\*/

.model{

margin: 20rpx;

background-color: rgba(255,255,255,0.6);

}

/\*导入微信组件样式文档\*/

@import "style/weui.wxss"

## info.wxml

<view class="container" >

/\*设置背景图片\*/

<view class='background'>

<image src='/images/background5.jpg' class='backimg'></image>

</view>

……

/\*设置背景音乐\*/

<audio id="myAudio" src="/music/Ukulele.mp3" id="myAudio" controls style="width:0;height:0" class="hide" loop="true"></audio>

## info.wxss

/\*设置用户信息的格式\*/

.userinfo {

text-align: center;

top: 0rpx;

position: relative;

margin-bottom: 15vh;

}

/\*设置头像显示格式\*/

.userinfo-avatar {

width: 128rpx;

height: 128rpx;

margin-top: 20vh;

border-radius: 50%;

border: 5px Solid rgba(216, 233, 240, 0.808);

}

/\*设置昵称显示格式\*/

.userinfo-nickname {

color: rgba(216, 233, 240, 0.808);

display: block;

}

/\*重新定义容器样式\*/

.container {

display: flex;

}

## exam.wxss

/\*定义页面样式\*/

page {

width: 100%;

height: 100%;

display: block;

}

.radio-indiv{

display: inline;

margin: 10rpx;

}

.radio{

display: block;

}

/\*设置输入栏的边框\*/

.input-style{

border: 5rpx solid rgb(71, 107, 141);

}

/\*设置计时器位置固定\*/

.countDown{

position:fixed;

top:0;

}

/\*设置滚动容器的高度及位置\*/

.show-scroll{

position:absolute;

bottom: 0;

height: 94vh;

}

# 全优题库测试用例及测试结果总结

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 01 | | |
| 测试用例名称 | 练习模式做题 | | |
| 产品名称 | 全优题库 | 产品版本 | 3.0 |
| 测试类型 | 黑盒测试 | 测试日期 | 2019.7.4 |
| 测试用例描述 | 选择科目、难度、题型并填入答案 | | |
| 测试步骤 | 1.在科目框中选择“计算机”，难度框中选择“简单”，题型框中选择“选择”，点击“开始练习”；  2.点击A选项、C、A、B、D；  3.点击返回；  3.题型框中选择“填空”，点击“开始练习”；  4.答案框中输入“1m/s”、“1”、“5”  5.点击返回；  3.题型框中选择“简答”，点击“开始练习”；  4.答案框中输入“33”、“7”、“59”  5.点击返回；  6.点击“开始练习”；  7.点击返回 | | |
| 前置条件 | 用户已授权登录 | | |
| 后置条件 | 查看解析 | | |
| 输入数据 | 科目：计算机、难度：简单、题型：选择、开始练习、A、C、A、B、D、返回、题型：填空、开始练习、1m/s、1、5、返回、题型：简答、开始练习、33/7/59、返回、开始练习、返回 | | |
| 预期结果 | 1.显示根据选择科目、难度、题型随机生成的题目以及可选答案  2.用户选择答案  2.1单选题  2.1.1用户选择的选项前做出标记  2.1.2用户点击已选择答案，答案变回未选中状态  2.2填空题  2.2.1用户在答题框输入答案  2.3简答题  2.3.1用户在答题框输入答案  3.用户选择“下一题”查看题目  4.答题完毕  用户点击“返回”，返回首页。  5.用户中途点击返回  返回首页面。 | | |
| 实际结果 | 1.显示根据选择科目、难度、题型随机生成的题目以及可选答案  2.用户选择答案  2.1单选题  2.1.1用户选择的选项前做出标记  2.1.2用户点击已选择答案，答案变回未选中状态  2.2填空题  2.2.1用户在答题框输入答案  2.3简答题  2.3.1用户在答题框输入答案  3.用户选择“下一题”查看题目  4.答题完毕  用户点击“返回”，返回首页。  5.用户中途点击返回  返回首页面。 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 02 | | |
| 测试用例名称 | 练习模式解析 | | |
| 产品名称 | 全优题库 | 产品版本 | 3.0 |
| 测试类型 | 黑盒测试 | 测试日期 | 2019.7.4 |
| 测试用例描述 | 查看解析 | | |
| 测试步骤 | 1.用户在每道题的页面中点击“查看解析”，查看每道题的解析  2.用户点击“返回”，回到做题页面。 | | |
| 前置条件 | 用户进入做题界面 | | |
| 后置条件 | 加入错题本 | | |
| 输入数据 | 查看解析、返回 | | |
| 预期结果 | 1. 查看解析   1.1用户在每道题的页面中点击“查看解析”，显示每道题的解析  1.2用户点击“返回”，回到做题页面 | | |
| 实际结果 | 1. 查看解析   1.1用户在每道题的页面中点击“查看解析”，显示每道题的解析  1.2用户点击“返回”，回到做题页面 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 03 | | |
| 测试用例名称 | 测试模式做题 | | |
| 产品名称 | 全优题库 | 产品版本 | 3.0 |
| 测试类型 | 黑盒测试 | 测试日期 | 2019.7.4 |
| 测试用例描述 | 选择科目、填入答案并提交 | | |
| 测试步骤 | 1.在科目框中选择“计算机”，点击“开始测试”；  2.点击A选项、C、B、1、2、3、4、5、6、提交；  3.点击返回； | | |
| 前置条件 | 用户已授权登录 | | |
| 后置条件 | 查看解析 | | |
| 输入数据 | 科目：计算机、开始测试、A、C、B、1、2、3、4、5、6、提交、返回 | | |
| 预期结果 | 1. 选择科目    1. 显示根据选择科目随机生成的题目以及可选答案   2.用户选择答案   * 1. 单选题   2.1.1用户选择的选项前做出标记  2.1.2用户点击已选择答案，答案变回未选中状态  2.2填空题  2.2.1用户在答题框输入答案  2.3简答题  2.3.1用户在答题框输入答案  3.答题完毕  用户点击“提交”，出现答题结束页面。 | | |
| 实际结果 | 1. 选择科目    1. 显示根据选择科目随机生成的题目以及可选答案   2.用户选择答案   * 1. 单选题   2.1.1用户选择的选项前做出标记  2.1.2用户点击已选择答案，答案变回未选中状态  2.2填空题  2.2.1用户在答题框输入答案  2.3简答题  2.3.1用户在答题框输入答案  3.答题完毕  用户点击“提交”，出现答题结束页面。 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 04 | | |
| 测试用例名称 | 计时 | | |
| 产品名称 | 全优题库 | 产品版本 | 3.0 |
| 测试类型 | 黑盒测试 | 测试日期 | 2019.7.4 |
| 测试用例描述 | 为测试模式中做题设置倒计时 | | |
| 测试步骤 | 1. 用户在倒计时为0s前点击提交； 2. 用户点击“回到首页”，“开始测试”； 3. 用户在倒计时为0s前点击返回； 4. 用户点击 “开始测试”； 5. 等待60s，倒计时为0s时系统自动提交。 | | |
| 前置条件 | 用户进入测试模式做题界面 | | |
| 后置条件 | 自动交卷 | | |
| 输入数据 | 提交、回到首页、开始测试、返回、开始测试 | | |
| 预期结果 | 1. 做题页面显示倒计时 2. 用户在倒计时结束前选择返回，倒计时停止 3. 倒计时结束时自动交卷，跳转到答题结束页面 | | |
| 实际结果 | 1. 做题页面显示倒计时 2. 用户在倒计时结束前选择返回，倒计时停止 3. 倒计时结束时自动交卷，跳转到答题结束页面 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 05 | | |
| 测试用例名称 | 查看解析和正确率 | | |
| 产品名称 | 全优题库 | 产品版本 | 3.0 |
| 测试类型 | 黑盒测试 | 测试日期 | 2019.7.4 |
| 测试用例描述 | 查看测试模式中答题后的解析和正确率 | | |
| 测试步骤 | 1. 用户点击 “查看解析” 2. 用户点击“退出” | | |
| 前置条件 | 用户进入测试模式并自动交卷或点击提交 | | |
| 后置条件 | 加入错题本 | | |
| 输入数据 | 查看解析、退出 | | |
| 预期结果 | 1.查看正确率  1.1显示用户所做试卷正确率  2.查看解析  2.1显示用户所选题目的解析  2.2显示用户所选答案是否正确  2.3显示选项：“加入错题本”  3. 用户选择“退出”：  返回首页。 | | |
| 实际结果 | 1.查看正确率  1.1显示用户所做试卷正确率  2.查看解析  2.1显示用户所选题目的解析  2.2显示用户所选答案是否正确  2.3显示选项：“加入错题本”  3. 用户选择“退出”：  返回首页。 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 06 | | |
| 测试用例名称 | 查看测试记录 | | |
| 产品名称 | 全优题库 | 产品版本 | 3.0 |
| 测试类型 | 黑盒测试 | 测试日期 | 2019.7.4 |
| 测试用例描述 | 查看或清除测试记录 | | |
| 测试步骤 | 1. 用户点击 “查看测试记录” 2. 用户点击“清除测试记录” | | |
| 前置条件 | 用户已授权登录 | | |
| 后置条件 | 无 | | |
| 输入数据 | 查看测试记录 | | |
| 预期结果 | 1. 用户点击“查看测试记录”   1.1显示每次测试的日期、科目和正确率  2.用户点击“清除测试记录”  2.1测试记录清除  3.用户点击取消  3.1返回首页面 | | |
| 实际结果 | 1. 用户点击“查看测试记录”   1.1显示每次测试的日期、科目和正确率  2.用户点击“清除测试记录”  2.1测试记录清除  3.用户点击取消  3.1返回首页面 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 07 | | |
| 测试用例名称 | 查看错题本 | | |
| 产品名称 | 全优题库 | 产品版本 | 3.0 |
| 测试类型 | 黑盒测试 | 测试日期 | 2019.7.4 |
| 测试用例描述 | 查看或移除错题 | | |
| 测试步骤 | 1. 用户点击 “查看错题本” 2. 用户点击“查看解析” 3. 用户点击“返回” 4. 用户点击“移除错题” 5. 用户点击取消 | | |
| 前置条件 | 用户已授权登录 | | |
| 后置条件 | 无 | | |
| 输入数据 | 查看错题本、查看解析、返回、移除错题、取消 | | |
| 预期结果 | 1. 显示已加入错题本的错题 2. 点击查看解析，显示每道题的解析 3. 点击移除错题，错题被移出错题本 4. 点击取消，回到首页 | | |
| 实际结果 | 1. 显示已加入错题本的错题 2. 点击查看解析，显示每道题的解析 3. 点击移除错题，错题被移出错题本 4. 点击取消，回到首页 | | |

测试结果总结：

最终所有功能都达到预期效果。

过程中遇到的问题

|  |  |
| --- | --- |
| 问题 | 解决办法 |
| 本地背景图在手机上不能显示 | 将背景图作为<image>并设置z-index：-1 |
| 背景图和计时器会向上滚动 | 设置position：fixed使得背景图和计时器固定 |
| 测试题向上滑动时与计时器重叠 | 将测试题的容器设置为<scroll-view>，并设置scroll-y='true'、位置及高度 |
| 客户意识不到科目、难度和题型是可选的 | 为科目、难度和题型设计样式，为了方便引入微信官方设计的样式weui-cells weui-cells\_after-title |
| 背景图颜色较深导致题目信息看不清 | 给题目设置半透明背景 |
| 练习模式输入框刷新时输入内容没有清空 | 设置数据绑定，动态清空 |
| 查看测试记录的功能的传参问题 | 解决缓存数据问题，正确使用wx.getStorageSync函数wx.setStorageSync函数 |

1. **主要问题及建议**

建议：

1. 升级题库，无需手动录题，设计算法使管理员可上传从网上下载的套卷
2. 开发者在需求分析过程中加强需求跟踪