**G07组代码清单**



G07小组：

成员：汤扬 孙昭 张璇

目录

[1概述 3](#_Toc501896278)

[1.1需求概述 3](#_Toc501896279)

[1.2参考资料 3](#_Toc501896280)

[2详细设计描述 3](#_Toc501896281)

[2.1功能结构 3](#_Toc501896282)

[3数据 5](#_Toc501896283)

[3.1E-R图 5](#_Toc501896284)

[3.2数据表结构 5](#_Toc501896285)

[4命名规范 6](#_Toc501896286)

[5编码规则 6](#_Toc501896287)

[5.1javascript 书写位置 6](#_Toc501896288)

[1.引入外部 js 文件 6](#_Toc501896289)

[5.2javascript 命名规范 7](#_Toc501896290)

[5.3注释 9](#_Toc501896291)

[5.4严格模式 10](#_Toc501896292)

[5.5模式 10](#_Toc501896293)

[5.6编写规范 11](#_Toc501896294)

[5.7变量 12](#_Toc501896295)

[6用户模块代码 13](#_Toc501896296)

# 1概述

## 1.1需求概述

小程序将分为以下四大模块：前端信息聚集和检索模块，用户模块，管理员模块，信息库模块。前端信息聚集和检索模块是程序的核心模块，实现了相应的功能与服务之间的跳转；用户模块和管理员模块对信息拥有不同的权限；信息库模块更新和存放程序将会用到的数据。

## 1.2参考资料

·《软件工程导论》 清华大学出版社 作者：张海藩等

·《软件工程 实践者的研究方法》 机械工业出版社 作者：罗杰 S.普莱斯曼等

·SE2017秋-G07-项目计划 作者：汤扬、孙昭、张璇

·SE2017秋-G07-可行性计划 作者：汤扬、孙昭、张璇

·SE2017秋-G07-需求分析报告 作者：汤扬、孙昭、张璇

·SE2017秋-G07-总体设计 作者：汤扬、孙昭、张璇

·SE2017秋-G07-详细设计 作者：汤扬、孙昭、张璇

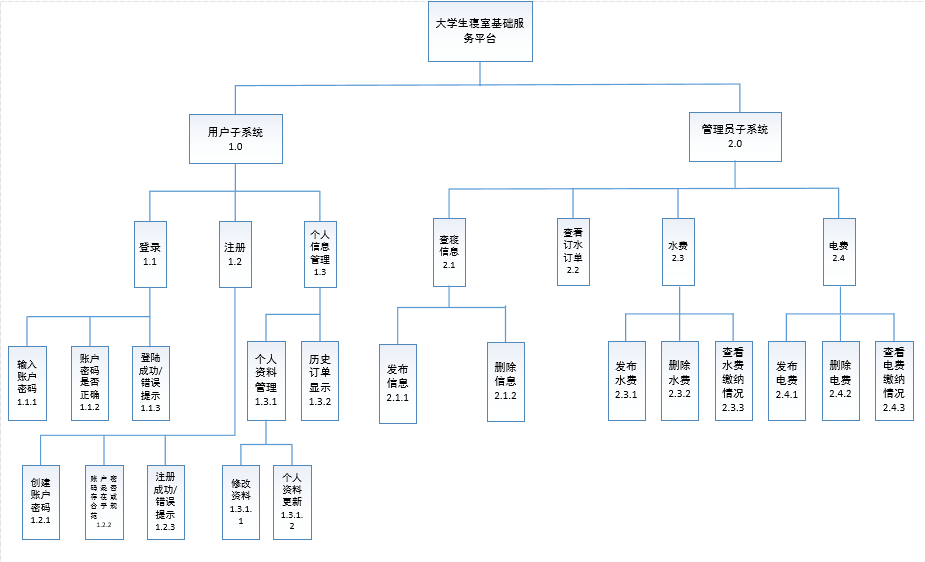
·http://blog.csdn.net/baidu\_32262373/article/details/52801121 js编码规范

# 2详细设计描述

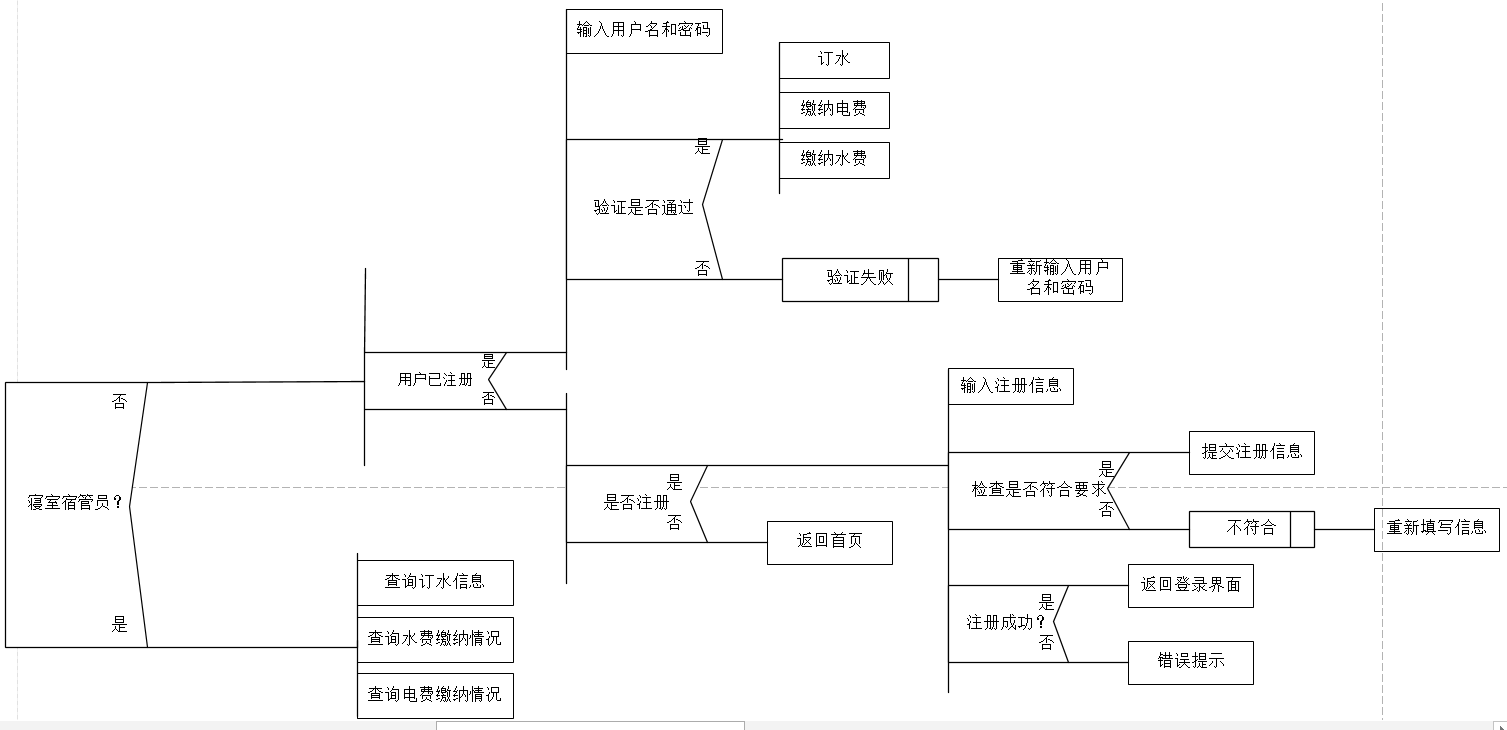
## 2.1功能结构

小程序将供用户注册并登录，填写必须的个人信息，可以进行水电费查询、订购桶装水等服务。小程序管理者会定期上传水电费等信息，供用户查询。同时小程序还提供支付功能，可以通过小程序订购桶装水。

HIPO图

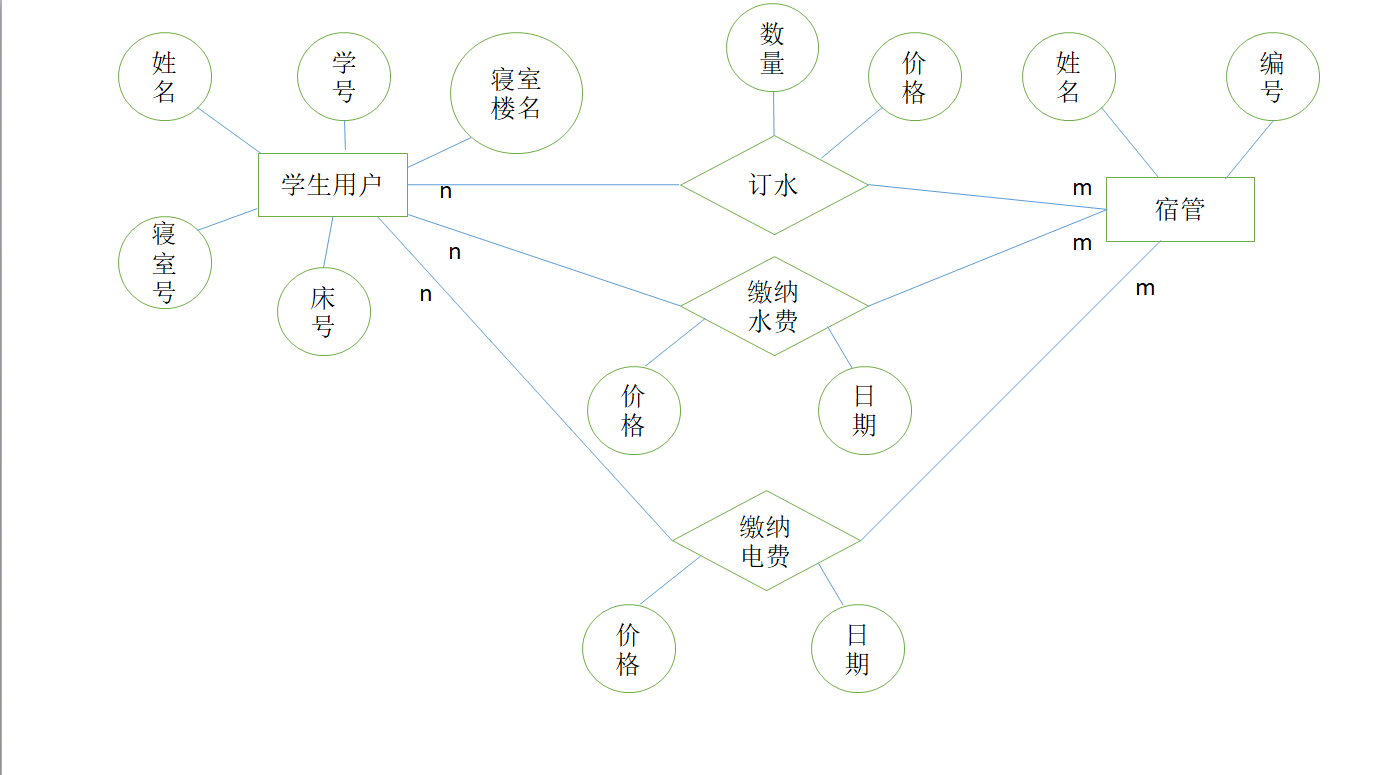


PAD图



# 3数据

## 3.1E-R图



## 3.2数据表结构

本系统涉及5张表，如下：

**学生用户表（表名：StudentUser）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 别名 | 字段类型 | 是否为空 | 描述 | 备注 |
| SNo | 学号 | int | 否 | 用户学号 | 主码 |
| Sname | 姓名 | varchar(20) | 否 | 用户姓名 |  |
| SDormName | 寝室楼名 | varchar(15) | 否 | 用户所在寝室楼名 |  |
| SDormNo | 寝室号 | int | 否 | 用户所在寝室编号 |  |
| SBedNo | 床号姓名 | int | 否 | 用户的床号 |  |

**宿管用户表（表名：DormManager）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 别名 | 字段类型 | 是否为空 | 描述 | 备注 |
| DManagerNo | 编号 | varchar(15) | 否 | 宿管编号 | 主码 |
| DManagerName | 姓名 | varchar(20) | 否 | 宿管姓名 |  |

**订水订单表（表名：BoardingOrder）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 别名 | 字段类型 | 是否为空 | 描述 | 备注 |
| SNo | 学号 | int | 否 | 用户学号 | 主码、外码 |
| DManagerNo | 编号 | varchar(15) | 否 | 宿管编号 | 主码、外码 |
| Account | 数量 | int | 否 | 订水数量 |  |
| Date | 日期 | Date | 否 | 订水日期 |  |

**水费表（表名：WaterFee）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 别名 | 字段类型 | 是否为空 | 描述 | 备注 |
| SNo | 学号 | int | 否 | 用户学号 | 主码、外码 |
| DManagerNo | 编号 | varchar(15) | 否 | 宿管编号 | 主码、外码 |
| Price | 价格 | Float | 否 | 应付水费 |  |
| Date | 日期 | Date | 否 | 付费日期 |  |

**电费表（表名：ElectricityFee）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 别名 | 字段类型 | 是否为空 | 描述 | 备注 |
| SNo | 学号 | int | 否 | 用户学号 | 主码、外码 |
| DManagerNo | 编号 | varchar(15) | 否 | 宿管编号 | 主码、外码 |
| Price | 价格 | Float | 否 | 应付电费 |  |
| Date | 日期 | Date | 否 | 付费日期 |  |

# 4命名规范

数据库命名使用匈牙利命名法

匈牙利命名法关键是：标识符的名字以一个或者多个小写字母开头作为前缀；前缀之后的是首字母大写的一个单词或多个单词组合，该单词要指明变量的用途。

编码阶段使用驼峰命名法

驼峰命名法，正如它的名称所表示的那样，指的是混合使用大小写字母来构成标识符的名字。其中第一个单词首字母小写，余下的单词首字母大写。

# 5编码规则

## 5.1javascript 书写位置

## 引入外部 js 文件

格式：<script src=”aaa.js”></script>

注：

* html和jsp页面中不允许将后面那个标签去掉
* ECMA第五版规定 type 属性不写时默认为 type=”text/javascript”，因此省略这个属性。

#### 2.自己写js 代码

<script>

// 自定义的 js 代码

</script>

#### 3.标签的位置

如果将 <script> 标签放到 <head> 标签中，等待外部js 文件全部下载、解析并执行完之后才能解析 <body> 中的元素，这会造成打开页面时会出现短暂的吃顿现象。

因此我们要求将 <script> 标签放到 <body> 元素中，并且置于 <body> 元素中的 html 代码的最下面。

#### 4.推荐

将 html 代码和 js 代码分离，也就是将 js 代码全部写到一个独立的以 \*.js 的文件中，再通过引入外部js 的方式引入到页面中。可以增加代码的美观性和方便后期代码维护。

### 5.2javascript 命名规范

#### 1.区分大小写

Javascript 中的一切（如：变量、函数名和操作符）都区分大小写。

如：

* 变量名 test 和变量名 Test 就是两个变量。
* typeof 作为关键字不能作为变量名，但typeOf就可以作为变量名了，因为它是一个新的变量。

#### 2.标识符

标识符：变量、函数、属性和函数参数的名字。

推荐命名规则：

* 第一个字符必须是字母或下划线（\_）或美元符号（$）；
* 其它字符可以是字母或下划线或美元符号或数字；
* 用全英文命名的，采用驼峰大小写格式。即第一个字母小写，后面每个有意义的首字母大写。   
  如：firstSecond myCar
* 标识符不能是关键字、保留字、true、false、null。

#### 3.关键字和保留字

在使用中有不熟悉的关键字和保留字可查询下表：

关键字有：（javascript 中有意义的单词）ECMA第五版

| 关 | 键 | 字 |  |
| --- | --- | --- | --- |
| break | do | instanceof | typeof |
| case | else | new | var |
| catch | finally | return | void |
| continue | for | switch | while |
| debugger | function | this | with |
| default | if | throw | try |
| delete | in |  |  |

保留字有：（javascript 中暂时没有意义，以后可能被用作关键字的单词）ECMA第三版

| 保 | 留 | 字 |  |
| --- | --- | --- | --- |
| abstract | enum（枚举） | int | short |
| boolean | export | interface | static |
| byte | extends | long | super |
| char | final | native | synchronized |
| class | float | package | throws |
| const | goto | private | transient |
| double | implements | protected | volatile |
| yield | import | public | let |

### 5.3注释

#### 1.单行注释

// 单行注释

#### 2.多行注释

/\*

\* 这是一个多行注释

\* 多行注释的第二行

\*/

#### 3.方法注释

方法的注释一般应注明方法是干什么用的，注意事项是什么，参数含义，返回数据信息。

/\*\*

\* 主要功能：生成随机数

\* 注意事项：此方法只能生成11-15位的随机数

\*

\* @param size 生成随机数的长度，范围在数字11~14之间

\* @return number 随机数

\*/

function getRandom (size) {

return Math.random().toString(16).substr(2, size);

}

### 5.4严格模式

'use strict';

严格模式 可以将js 代码中的一些不确定行为得到处理，对某些不安全的操作也会抛出错误，强制性的将js 代码变得更加规范和不容易出错。

上述看似字符串的xx 实际上是一个编译指示，用于告诉Javascript 引擎切换到严格模式。

### 5.5模式

#### 1.分号的使用

每一句代码的结尾推荐加上”;”，尽管在 javascript 中代码结束不加分号也不会报错。

加分号”;”的好处有：

* 避免错误，诸如不完整的输入；
* 可以通过删除多余的空格来压缩js 代码，代码结尾没有分号会导致压缩错误；
* 增进代码性能。解析器不必花时间推测应该从哪里插入分号来结束语句。

#### 2.代码块的使用

如：if条件语句中，如果代码块中只有一条语句

if (test)

alert(test);

上述代码不会导致错误，但不推荐使用。推荐使用的方式如下：

if (test) {

alert(test);

}

### 5.6编写规范

#### 1.对于条件语句

if (条件) {

// 条件语句

}

注：if 与 () 和 {} 之间的空格

#### 2.对于循环语句

for (var i = 0; i < 10; i++) {

// 循环语句

}

注：for的括弧内每个分号后面要空一格

#### 3.对于方法

function test (param1, param2) {

// 方法体

}

注意：参数中逗号后面有个空格

#### 4.js 代码中尽量使用单引号。

### 5.7变量

#### 1.使用var 定义变量

javascript 里的变量是松散类型的，也就是即可以用 var 定义字符串类型的，也可以用var 定义整数类型的。

#### 2. var message;

像这样只定义没有初始化值的，默认会保存一个特殊的值 —— undefined

#### 3.同一个变量不推荐接收多种数据类型的值

var hhh = 'hello';

hhh = 111;

像这种首先用 hhh 变量接收字符串类型的值，再用 hhh 变量接收整型的值，语法上没有错误，但我们不推荐这样使用。

#### 4.认识局部变量

function test () {

var message = 'hello';

}

test();

alert(message);

这里的message 是用 var 定义的局部变量，在调用方法test() 时被创建出来，在方法结束的时候也同时销毁了，所以alert(message) 会报错。

#### 5.认识全局变量

function test () {

message = 'hello';

}

test();

alert(message);

这里的message 没有用var 定义，在调用方法 test() 时被创建出来，没有用var 定义的变量属于全局变量，因此就算方法执行完毕也不会被回收器回收，所有 alert(message) 时就会弹出 ‘hello’。

#### 6.用 var 可以同时定义多个变量，中间用逗号”,”隔开。

如：

var aaa = 'hi',

bbb = 12,

ccc = true;

#### 7.严格模式下注意事项

严格模式下变量名不能为 eval 和 arguments ，尽管这两个词既不是关键字也不是保留字，但就是不能使用，否则报语法错误

# 6用户模块代码

用户登录

var Bmob = require('../../utils/bmob.js');

var common = require('../../utils/common.js');

var app = getApp();

// var User = Bmob.Object.extend("user");

// var query = new Bmob.Query(User);

Page({

onLoad: function () {

},

data: {

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

},

/\*\*

\* 生命周期函数--监听页面初次渲染完成

\*/

onReady: function () {

},

/\*\*

\* 生命周期函数--监听页面显示

\*/

onShow: function () {

},

/\*\*

\* 生命周期函数--监听页面隐藏

\*/

onHide: function () {

},

/\*\*

\* 生命周期函数--监听页面卸载

\*/

onUnload: function () {

},

radioChange:function(e) {

// console.log(e.detail.value);

},

tap\_register: function (e) {

wx.navigateTo({

url: '../register/register',

})

},

tap\_cancel: function (e) {

wx.navigateBack({

})

},

formSubmit: function (event) {

Bmob.User.logIn(event.detail.value.id, event.detail.value.passwd,{

success: function (user) {

// var currentUser = Bmob.User.current();

if (user.get("registered")!=true) {

common.showTip("对不起,您输入的用户不存在或密码错误", "loading");

Bmob.User.logOut();

}

else if (user.get("type") != event.detail.value.type) {

common.showTip("对不起,您输入的用户不存在或密码错误", "loading");

Bmob.User.logOut();

}

else {

wx.getStorage({

key: Bmob.\_getBmobPath(Bmob.User.\_CURRENT\_USER\_KEY),

success: function (res) {

var Data = JSON.parse(res.data);

common.showTip("登录成功,正在跳转", "success", function () {

if (event.detail.value.type == 1) {

wx.reLaunch({

url: '../indexuser/indexuser'

})

}

else if (event.detail.value.type == 2) {

wx.reLaunch({

url: '../indexadmin/indexadmin'

})

}

});

}

})

}

},

error: function (user, error) {

// The login failed. Check error to see why.

console.log(error)

common.showTip("对不起,您输入的用户不存在或密码错误", "loading");

}

});

}

用户注册

var Bmob = require('../../utils/bmob.js');

var common = require('../../utils/common.js');

var that;

// 获取全局应用程序实例对象

var app = getApp();

// 创建页面实例对象

Page({

data: {

userInfo: {},

showView: false,

loading: true,

array1: ['慕贤1', '慕贤2', '慕贤3', '弘毅1', '弘毅2', '精诚1', '精诚2', '致远1', '致远2', '尚雅1', '尚雅2', '思睿1', '思睿2', '求真1', '求真2', '惟学1', '明德2', '明德3', '问源1', '问源2'],

array2: ['慕贤', '弘毅', '精诚', '致远', '尚雅', '思睿', '求真', '惟学', '明德', '问源'],

index: 0,

},

onLoad: function (options) {

that = this;

//调用应用实例的方法获取全局数据

app.getUserInfo(function (userInfo) {

console.log(userInfo)

//更新数据

that.setData({

userInfo: userInfo

})

})

showView: (options.showView == "true" ? true : false)

},

radioChange: function () {

that = this;

that.setData({

showView: (!that.data.showView)

})

},

pickerSelected1: function (e) {

//改变index值，通过setData()方法重绘界面

this.setData({

index: e.detail.value

});

},

pickerSelected2: function (e) {

//改变index值，通过setData()方法重绘界面

this.setData({

index: e.detail.value

});

},

tap\_cancel: function (e) {

wx.navigateBack({

})

},

formSubmit: function (event) {

Bmob.User.logOut();

that.setData({

loading: false,

tips: "用户注册中，请稍后！"

})

var accouttype = event.detail.value.type;

var accoutName = event.detail.value.accoutName;

var realname = event.detail.value.realname;

var dormname = event.detail.value.picker1;

var dormnameadmin = event.detail.value.picker2;

var dormnum = event.detail.value.dormnum;

var bednum = event.detail.value.bednum;

var accoutPswd = event.detail.value.accoutPswd;

var accoutPswd1 = event.detail.value.accoutPswd1;

var User = Bmob.Object.extend("\_User");

var query = new Bmob.Query(User);

var user = new Bmob.User();

query.equalTo("username", accoutName);

query.first({

success: function (object) {

user = object;// 查询成功

if (user == undefined) {

common.showTip("该学号或工号不存在！", "loading");

}

else {

var dbtype = user.get("type");

var dbrealname = user.get("userRealName");

var dbdormname = user.get("dormName");

var dbdormnameadmin = user.get("dormNameAdmin");

var dbdormnum = user.get("dormNo");

var dbbednum = user.get("bedNo");

if (user.get("registered") == true) common.showTip("该用户已注册！", "loading");

else if (accouttype == 1) {

if (accoutPswd == accoutPswd1 && accoutPswd != "" && accoutName != "" && accouttype==dbtype && realname==dbrealname && dormname==dbdormname && dormnum==dbdormnum && bednum==dbbednum) {

Bmob.User.logIn(accoutName, accoutName, {

success: function (user) {

wx.getStorage({

key: Bmob.\_getBmobPath(Bmob.User.\_CURRENT\_USER\_KEY),

success: function (res) {

var Data = JSON.parse(res.data);

var currentUser = Bmob.User.current();

console.log(currentUser);

currentUser.set("password", accoutPswd);

currentUser.set("registered", true);

currentUser.set("userPic", that.data.userInfo.avatarUrl);

currentUser.save();

that.setData({

loading: true,

})

common.showTip("注册成功,正在跳转", "success", function () {

wx.reLaunch({

url: '../indexuser/indexuser'

})

});

}

})

},

error: function (user, error) {

// The login failed. Check error to see why.

console.log(error)

}

});

// user.signUp(null, {

// success: function (res) {

// that.setData({

// loading: true,

// })

// common.showTip("注册成功,正在跳转", "success", function () {

// wx.redirectTo({

// url: '../login/login'

// })

// });

// console.log(res)

// // wx.getStorage({

// // key: Bmob.\_getBmobPath(Bmob.User.\_CURRENT\_USER\_KEY),

// // success: function(res) {

// // var Data =JSON.parse(res.data) ;

// // if(Data.code=="202"){

// // common.showTip("该用户名已经存在！","loading");

// // }

// // else{

// // common.showTip("注册成功请登录","success",function(){

// // wx.redirectTo({

// // url: '../login/login'

// // })

// // });

// // }

// // }

// // })

// },

// error: function (userData, error) {

// that.setData({

// loading: true,

// })

// console.log(userData)

// console.log("userData")

// console.log(error)

// },

// complete: function (userData) {

// that.setData({

// loading: true,

// })

// }

// });

}

else {

console.log("注册信息不全或有误");

common.showTip("注册信息不全或有误", "loading");

}

}

else if (accouttype == 2) {

if (accoutPswd == accoutPswd1 && accoutPswd != "" && accoutName != "" && accouttype == dbtype && realname == dbrealname && dormnameadmin == dbdormnameadmin) {

Bmob.User.logIn(accoutName, accoutName, {

success: function (user) {

wx.getStorage({

key: Bmob.\_getBmobPath(Bmob.User.\_CURRENT\_USER\_KEY),

success: function (res) {

var Data = JSON.parse(res.data);

var currentUser = Bmob.User.current();

console.log(currentUser);

currentUser.set("password", accoutPswd);

currentUser.set("registered", true);

currentUser.set("userPic", that.data.userInfo.avatarUrl);

currentUser.save();

that.setData({

loading: true,

})

common.showTip("注册成功,正在跳转", "success", function () {

wx.redirectTo({

url: '../indexadmin/indexadmin'

})

});

}

})

},

error: function (user, error) {

// The login failed. Check error to see why.

console.log(error)

}

});

}

else {

console.log("注册信息不全或有误");

common.showTip("注册信息不全或有误", "loading");

}

}

}

},

error: function (error) {

console.log("查询失败: " + error.code + " " + error.message);

}

});

}

// /\*\*

// \* 生命周期函数--监听页面加载

// \*/

// onLoad () {

// },

// /\*\*

// \* 生命周期函数--监听页面初次渲染完成

// \*/

// onReady () {

// },

// /\*\*

// \* 生命周期函数--监听页面显示

// \*/

// onShow () {

// },

// /\*\*

// \* 生命周期函数--监听页面隐藏

// \*/

// onHide () {

// },

// /\*\*

// \* 生命周期函数--监听页面卸载

// \*/

// onUnload () {

// },

// /\*\*

// \* 页面相关事件处理函数--监听用户下拉动作

// \*/

// onPullDownRefresh () {

// },

// })

})

用户订水

var Bmob = require('../../utils/bmob.js');

// 获取全局应用程序实例对象

var app = getApp();

var that;

// 创建页面实例对象

Page({

/\*\*

\* 页面名称

\*/

name: "barreledwateruser",

/\*\*

\* 页面的初始数据

\*/

data: {

array1: ['慕贤1', '慕贤2', '慕贤3', '弘毅1', '弘毅2', '精诚1', '精诚2', '致远1', '致远2', '尚雅1', '尚雅2', '思睿1', '思睿2', '求真1', '求真2', '惟学1', '明德2', '明德3', '问源1', '问源2'],

index: 0,

money: 0,

loading: true,

openId: '',

dataInfo: [],

PayResult: ''

},

pickerSelected1: function (e) {

//改变index值，通过setData()方法重绘界面

this.setData({

index: e.detail.value

});

},

tap\_cancel: function (e) {

wx.navigateBack({

})

},

inputEvent: function (e) {

this.setData({

money: e.detail.value\*10

})

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad() {

that = this;

// 页面初始化 options为页面跳转所带来的参数

that.setData({

user: Bmob.User.current(),

})

that.setData({

index: that.data.array1.indexOf(Bmob.User.current().get("dormName"))

})

//获取open id，请在官网填写微信小程序key

wx.login({

success: function (res) {

if (res.code) {

//发起网络请求

console.log(res.code)

Bmob.User.requestOpenId(res.code, {

success: function (result) {

that.setData({

loading: true,

openId: result.openid

})

console.log(result)

},

error: function (error) {

// Show the error message somewhere

console.log("Error: " + error.code + " " + error.message);

}

});

} else {

console.log('获取用户登录态失败！' + res.errMsg)

common.showTip('获取用户登录态失败！', 'loading');

}

}

});

},

onReady: function () {

// 页面渲染完成

},

onShow: function () {

// 页面显示

},

onHide: function () {

// 页面隐藏

},

onUnload: function () {

// 页面关闭

},

formSubmit: function (event) {

var dormname = event.detail.value.picker1;

var dormnum = event.detail.value.dormnum;

var quantity = event.detail.value.quantity;

if (quantity > 0 && quantity < 6 && dormnum > 100 && dormnum < 800) {

var openId = this.data.openId;

if (!openId) {

console.log('未获取到openId请刷新重试');

}

//传参数金额，名称，描述,openid

Bmob.Pay.wechatPay(that.data.money, '桶装水', '桶装水', openId).then(function (resp) {

console.log('resp');

console.log(resp);

that.setData({

loading: true,

dataInfo: resp

})

//服务端返回成功

var timeStamp = resp.timestamp,

nonceStr = resp.noncestr,

packages = resp.package,

orderId = resp.out\_trade\_no,//订单号，如需保存请建表保存。

sign = resp.sign;

//打印订单号

console.log(orderId);

//发起支付

wx.requestPayment({

'timeStamp': timeStamp,

'nonceStr': nonceStr,

'package': packages,

'signType': 'MD5',

'paySign': sign,

'success': function (res) {

//付款成功,这里可以写你的业务代码

var BoardOrder = Bmob.Object.extend("BoardOrder");

var boardOrder = new BoardOrder();

var currentUser = Bmob.User.current();

console.log(currentUser.get("username"), dormname, Number(dormnum), Number(quantity), orderId)

// 添加数据，第一个入口参数是Json数据

boardOrder.save({

username: currentUser.get("username"),

dormName: dormname,

dormNo: Number(dormnum),

amount: Number(quantity),

orderId: orderId,

}, {

success: function (result) {

// 添加成功

console.log("订水记录创建成功, objectId:" + result.id);

wx.redirectTo({

url: '../paysuccessful/paysuccessful',

})

},

error: function (result, error) {

// 添加失败

console.log('创建订水记录失败');

}

});

console.log(res);

},

'fail': function (res) {

//付款失败

console.log('付款失败');

console.log(res);

// wx.redirectTo({

// url: '../payfailed/payfailed',

// })

}

})

}, function (err) {

console.log('服务端返回失败');

common.showTip(err.message, 'loading', {}, 6000);

console.log(err);

});

}

else {

console.log("信息填写有误")

common.showTip("订单信息填写有误", "loading");

}

},

})

寝室报修

var Bmob = require('../../utils/bmob.js');

var common = require('../../utils/common.js');

// 获取全局应用程序实例对象

var app = getApp();

var that;

// 创建页面实例对象

Page({

/\*\*

\* 页面名称

\*/

name: "fixuser",

/\*\*

\* 页面的初始数据

\*/

data: {

array1: ['慕贤1', '慕贤2', '慕贤3', '弘毅1', '弘毅2', '精诚1', '精诚2', '致远1', '致远2', '尚雅1', '尚雅2', '思睿1', '思睿2', '求真1', '求真2', '惟学1', '明德2', '明德3', '问源1', '问源2'],

index: 0,

},

pickerSelected1: function (e) {

//改变index值，通过setData()方法重绘界面

this.setData({

index: e.detail.value

});

},

tap\_cancel: function (e) {

wx.navigateBack({

})

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad () {

that = this;

// 页面初始化 options为页面跳转所带来的参数

that.setData({

user: Bmob.User.current(),

})

that.setData({

index: that.data.array1.indexOf(Bmob.User.current().get("dormName"))

})

},

/\*\*

\* 生命周期函数--监听页面初次渲染完成

\*/

onReady () {

},

/\*\*

\* 生命周期函数--监听页面显示

\*/

onShow () {

},

/\*\*

\* 生命周期函数--监听页面隐藏

\*/

onHide () {

},

/\*\*

\* 生命周期函数--监听页面卸载

\*/

onUnload () {

},

/\*\*

\* 页面相关事件处理函数--监听用户下拉动作

\*/

onPullDownRefresh () {

},

formSubmit: function (event) {

var dormname = event.detail.value.picker1;

var dormnum = event.detail.value.dormnum;

var bednum = event.detail.value.bednum;

var textarea = event.detail.value.textarea;

if (dormnum > 100 && dormnum < 800 && bednum >= 1 && bednum <= 4 && textarea !="") {

var FixOrder = Bmob.Object.extend("FixOrder");

var fixOrder = new FixOrder();

var currentUser = Bmob.User.current();

// 添加数据，第一个入口参数是Json数据

fixOrder.save({

username: currentUser.get("username"),

dormName: dormname,

dormNo: Number(dormnum),

bedNo: Number(bednum),

content: textarea,

}, {

success: function (result) {

// 添加成功

console.log("报修记录创建成功, objectId:" + result.id);

wx.redirectTo({

url: '../successful/successful',

})

},

error: function (result, error) {

// 添加失败

console.log('创建报修记录失败');

}

});

}

else {

console.log("信息填写有误")

common.showTip("报修信息填写有误", "loading");

}

},

})