# 1、系统概述

**1.1、系统简介**

本项目旨在实现一个同学之间互相帮助的平台，同学们可以在其

中展示自己学习成果，帮助他人，结交朋友，亦可得到他人的学习

指导，学习经验。

**1.2、术语表**

定义系统或产品中涉及的重要术语，为读者在阅读文档时提供必要的参考信息。

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | Userip | 全局变量，储存用户的云平台的openid |
| 2 | Userinfo | 用户的头像昵称等个人信息 |
| 3 | Ddxx | 订单信息 |
| 4 | Kcxx | 课程信息 |
| 5 | Yhxx | 用户信息 |
| 6 | Kqkc | 开启课程 |
| 7 | Ktxq | 课堂详情 |
| 8 | Shouye | 首页 |
| 9 | Wd | 我的 |
| 10 | Wddd | 我的订单 |
| 11 | Wdkt | 我的课堂 |
| 12 | Zc | 注册 |

**1.3、系统运行环境**

硬件平台：电脑、手机

操作系统：安卓，windows，IOS，MAC

数据库系统：既可在小程序前端操作，也能在云函数中读写的JSON 数据库

编程平台：微信web开发者工具

**1.4、开发环境**

微信web开发者工具。

wxml，wxss，js语言

# 2、系统说明

**2.1、整体说明**

在该部分简要介绍系统的整体功能，并给出系统对于的流程图。对于一些模块间合作完成或是系统整体合作完成的功能，在该部分应向用户解释该如何输入，又应得到什么输出。

# 

索引模块

浏览及预约模块

数据采集模块

注册及登录模块

用户输入

数据存储模块

用户信息模块

**2.2、模块1 登陆模块**

2.2.1、功能描述

为用户提供登陆功能

2.2.2、使用说明

需要用户点击头像供程序识别，并输入密码进行登陆

2.2.3、重要代码分析

（获取用户信息）

onGetUserInfo: function (e) {

if (!this.logged && e.detail.userInfo) {

this.setData({

logged: true,

avatarUrl: e.detail.userInfo.avatarUrl,

userInfo: e.detail.userInfo

})

}

wx.cloud.callFunction({

name: 'login',

data: {},

success: res => {

console.log('[云函数] [login] user openid: ', res.result.openid)

app.globalData.openid = res.result.openid

this.setData({username:res.result.openid})

console.log(this.data.username)

// this.data.username = res.result.openid

wx.showToast({

title: '用户识别成功！',

icon: 'none',

duration: 2000

})

app.appData.userip=res.result.openid

},

fail: err => {

wx.showToast({

title: '用户识别失败！',

icon: 'none',

duration: 2000

})

console.error('[云函数] [login] 调用失败', err)

}

})

},

（登陆函数，可以判读密码正确性并进行页面跳转）

loginBtnClick: function () {

if (this.data.username == null) {

wx.showToast({

title: '请点击头像识别用户',

icon: 'none',

duration: 2000

})

} else {

wx.cloud.init({

env: 'sjk-666',

traceUser: true

});

const db = wx.cloud.database();

db.collection('yhxx').where({\_id:this.data.username}).get({

success:res =>{

console.log(res.data);

if (res.data[0].mm == this.data.password) {

console.log("登录成功")

wx.switchTab({

url: '/pages/zhuye/zhuye'

})

app.appData.userinfo = {

username: this.data.username, password: this.data.password ,touxiang:this.data.avatarUrl}

}

else{

wx.showToast({

title: '用户名或密码错误',

icon: 'none',

duration: 2000

})

}

},

})

**2.3、模块2 注册模块**

2.2.1、功能描述

为用户提供注册功能

2.2.2、使用说明

需要用户输入个人信息

2.2.3、重要代码分析

addbtn: function () {

if (this.data.inputValue5 == '') {

wx.showModal({

title: 'QQ号为空',

content: '请输入QQ号',

success: function (res) {

if (res.confirm) {

console.log('用户点击确定')

}

}

});

}

if (this.data.inputValue4 == '') {

wx.showModal({

title: '邮箱为空',

content: '请输入邮箱',

success: function (res) {

if (res.confirm) {

console.log('用户点击确定')

}

}

});

}

if (this.data.inputValue3 == '') {

wx.showModal({

title: '昵称为空',

content: '请输入昵称',

success: function (res) {

if (res.confirm) {

console.log('用户点击确定')

}

}

})

}

if (this.data.inputValue2 == '') {

wx.showModal({

title: '密码为空',

content: '请输入密码',

success: function (res) {

if (res.confirm) {

console.log('用户点击确定')

}

}

})

}

if (this.data.inputValue1 == '') {

wx.showModal({

title: '学号为空',

content: '请输入学号',

success: function (res) {

if (res.confirm) {

console.log('用户点击确定')

}

}

});

}

if (this.data.inputValue1 != '' &&

this.data.inputValue2 != '' &&

this.data.inputValue3 != '' &&

this.data.inputValue4 != ''&&

this.data.inputValue5 != '') {

const db = wx.cloud.database({

env: 'sjk-666'

})

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

// data 字段表示需新增的 JSON 数据

data: {

// \_id: 'todo-identifiant-aleatoire', // 可选自定义 \_id，在此处场景下用数据库自动分配的就可以了

\_id: this.data.username,

avatarUrl: this.data.avatarUrl,

xh: this.data.inputValue1,

mm: this.data.inputValue2,

nc: this.data.inputValue3,

yx: this.data.inputValue4,

qq: this.data.inputValue5,

},

success(res) {

// res 是一个对象，其中有 \_id 字段标记刚创建的记录的 id

console.log(res)

}

})

wx.navigateTo({

url: '../shouye/shouye',

})

}

},

**在此仅展示两个模块，其他更多模块的功能及代码详见GitHub**

**<https://github.com/zzyskywalker/dalaobang.git>**

**2.4、模块3--浏览及预约模块**

2.4.1、功能描述

显示可以选择的课程并且点击后显示详细信息，可以点击预约模块确认订单并进行预约

2.4.2、使用说明

用户进行点击即可

**2.5 模块4--云存储模块**

2.5.1、功能描述

包括云函数和云api，可对数据库进行增删查改等操作

2.5.2、使用说明

封装在程序之内，无需用户操作

# 3、功能测试

该部分针对前一部分的使用说明中提到的各功能进行测试。

**3.1、模块1--登陆模块**

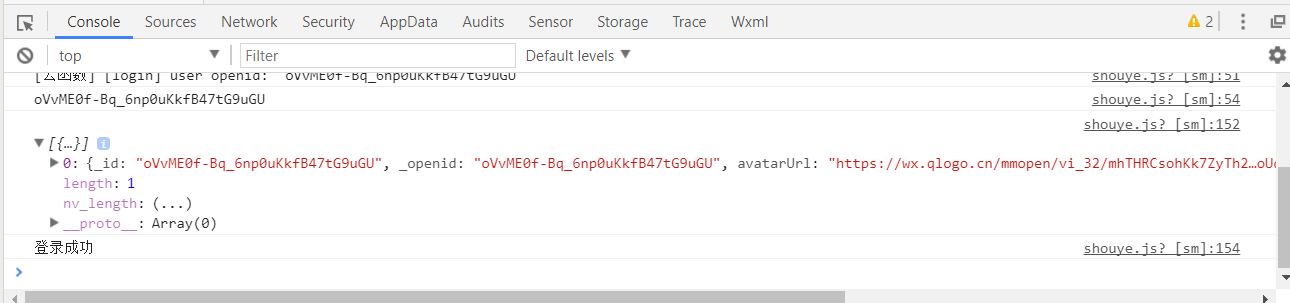
3.1.1、测试用例

利用自己注册的账号进行程序功能的测试

3.1.2、测试结果

根据测试样例，运行软件，得到测试结果。

实现的页面以及登陆时控制台的输出显示



3.1.3、结果分析

可以完整地实现目标功能

**3.2、模块2--注册模块**

3.1.1、测试用例

利用自己注册的账号进行程序功能的测试

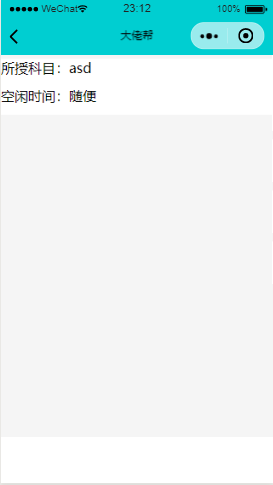
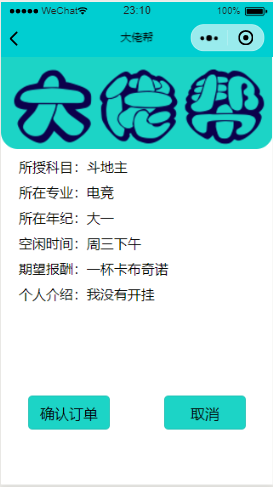
3.1.2、测试结果

根据测试样例，运行软件，得到测试结果。

实现的页面



其他模块仅展示效果



# 4、性能测试

4.1、精度分析

输入数据可以保证准确地保存在云平台，同时从云平台向用户返回展示相关数据时也可保证准确。

4.2、时间特性分析

页面可在毫秒级时间内跳转，并且可以在1s内获取云端数据显示在页面上

4.3、灵活性分析

在安卓及苹果机都可以使用，云平台权限严谨，有一定防止攻击的效果，可以保持鲁棒性。

# 4、测试结论和建议

基本功能经测试均可实现，但是还有一些提高目标没有实现。

比如评分功能，学生可以根据授课人的辅导水平等满意程度对授课人进行评分，综合分数越高的大佬的课程会被优先展示在课程页面上。

以及举报功能，一方出现违约的情况或者平台里出现不良信息等，用户可进行监督并举报，以保证平台稳定性。

另外附加模块——问答模块，用户有不会的习题可以上传到平台，可以使用积分进行悬赏，其他用户可以进行回答并且赚取积分，参考案例有作业帮、百度知道。

将云开发平台转为自己搭建服务器，因为这样更加的自由，可以设置更多的内容。