# 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**

|  |  |  |
| --- | --- | --- |
| **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、数据结构说明

本章说明本程序系统中使用的全局数据常量、变量和数据结构。

2.1、常量

无

2.2、变量

全局变量：

appData: {

userinfo: null,//储存用户个人信息，包括头像，名称等

userip:null//用户个人的openid

},

2.3、数据结构

数据内容如下表：

|  |  |
| --- | --- |
| 界面 | 数据 |
| 登录界面 | 学号、密码 |
| 注册界面 | 昵称、学号、密码、邮箱、手机、qq |
| 索引界面 | 用户昵称、用户专业、辅导科目 |
| 被索引用户详细信息界面 | 昵称、专业、辅导科目、联系方式、辅导时间、自我介绍 |
| 用户个人信息界面 | 订单界面、修改资料界面、成为大佬界面、我的课堂界面 |
| 修改资料界面 | 昵称、学号、密码、邮箱、手机、qq |
| 订单界面 | 订单信息（大佬信息、辅导科目、时间） |
| 成为大佬界面 | 昵称、专业、辅导科目、联系方式、辅导时间、自我介绍 |
| 我的课堂 | 被辅导人信息、授课时间 |

**数据库**

表一 个人信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 昵称 | 密码 | 昵称 | QQ | 手机 |
|  |  |  |  |  |  |

表二 大佬信息

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学号 | 昵称 | 专业 | 授课科目 | 联系方式 | 空闲时间 | 自我介绍 |
|  |  |  |  |  |  |  |

表三 订单信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生学号 | 大佬学号 | 大佬联系方式 | 学生联系方式 | 授课时间 | 授课科目 |
|  |  |  |  |  |  |

订单信息：ddxx: [sj,kc,xc,qq]([时间、课程、薪酬、QQ]）

课程信息：kcxx:[kc,zy,nj,sj,bc，ms]([课程、专业、年级、时间、报酬、个人介绍])

开启课程 :kqkc[kc，zy，nj，sj,xc，ms,qq] (课程、专业、年纪、时间、薪酬、个人介绍、QQ)

# 3、模块设计

索引模块

浏览及预约模块

**3.1、软件结构**

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

**3.2、功能设计说明**

外部接口数据采集模块将对用户的输入进行收集分析，并传输给后三个模块；注册及登录模块为用户提供登录和注册服务，成功登录或注册完成后用户可选择进入用户信息界面及预约界面，同时，注册的信息会保存至数据存储模块；用户信息模块将调用数据存储模块内保存的用户信息返还给用户，用于用户的查看及修改；浏览及预约模块内置索引模块，用户可根据自己需求搜索不同专业不同学科的大佬，该模块会将存储模块内保存的大佬信息返还给用户；数据存储模块将会保存上述模块中的数据，并在需要时返还给相应模块。

**3.3、模块1：注册和登录模块**

3.3.1、功能描述

主要实现用户的登录注册功能

3.3.2、输入数据

登录：学号、密码

注册：昵称、学号、密码、邮箱、手机、QQ

学号限制为北邮10位学号、密码为8~16位、含英文及数字

详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。

3.3.3、输出数据

用户的昵称、学号、密码、邮箱、手机、QQ等信息将上传至本地或云端保存

3.3.4、数据设计

给出本程序中的局部数据结构说明，包括数据结构名称，功能说明，具体数据结构说明（定义、注释设计、取值）等。相关数据库表，数据存储设计（具体说明需要以文件方式保存的数据文件名、数据存储格式、数据项及属性等。）

3.3.5、算法和流程

登陆

输入密码

用户识别

注册

3.3.6、函数说明

登陆按钮

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().then(res => {

console.log(res.data);

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

console.log("登录成功");

this.setData({ xh: res.data[0].xh })

wx.switchTab({

url: '../wd/wd',

})

app.appData.userinfo = { username: this.data.username, password: this.data.password }

}

else {

wx.showToast({

title: '用户名或密码错误',

icon: 'none',

duration: 2000

})

}

})

}

}

注册按钮

djzcan: function () {

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

wx.showToast({

title: '请点击头像识别用户',

icon: 'none',

duration: 2000

})

} else {

wx.navigateTo({ url: '../zc/zc?username=' + this.data.username + '&avatarUrl=' + this.data.avatarUrl })

}

}

3.3.7 全局数据结构与该模块的关系

云函数获得的 userinfo是全局变量

**3.4、模块2：用户信息模块**

3.4.1、功能描述

为用户提供各种操作 如修改密码等功能，同时包含对订单的管理

3.4.2、输入数据

用户可在此模块中修改之前保存的信息，如用户名、密码、联系方式等

3.4.3、输出数据

用户修改后的信息将输出至数据存储模块，并替换原有数据

onLoad: function (options) {

this.setData({

tcid:options.tcid

})

const db = wx.cloud.database({

env: 'sjk-666'

})

db.collection('ddxq').where({

tcid: this.data.tcid

}).get(

{

success: res => {

this.setData({

ddxq: res.data

})

console.log(this.data.ddxq[0])

},

})

if(this.data.ddxq == [])

{

wx.showModal({

title: '错误',

content: '数据拉取错误',

success: function (res) {

if (res.confirm) {

console.log('用户点击确定')

wx.navigateBack({

delta: 1

});

}

}

});

}

},

onShareAppMessage: function () {

},

cancelbtn:function(){

var that=this;

const db = wx.cloud.database({

env: 'sjk-666'

})

wx.showModal({

title: '删除',

content: '确认删除订单',

showCancel: true,

success(res) {

if (res.confirm) {

console.log('确认取消订单')

wx.cloud.callFunction({

name:'deletedd',

data:{

crid:that.data.tcid//作用域，this在闭包里面，需要用that传递一下

},

success(res){

console.log("传入的", that.data.tcid)

console.log("返回", res.result.tcid)

console.log('调用成功')

console.log('取消成功')

wx.navigateBack({

delta:1

})

}

**3.5、模块3：浏览及预约模块**

3.5.1、功能描述

主要实现用户浏览信息及预约课程功能

3.5.2、输入数据

用户按需求搜索大佬的专业、科目

3.5.3、输出数据

向用户返回搜索结果，包含大佬的详细信息：昵称、专业、所授科目、自我介绍、

联系方式、空闲时间

onShow: function () {

const db = wx.cloud.database({

env: 'sjk-666'

})

db.collection('kcxx').where({

\_id: this.data.id

}).get(

{

success: res => {

this.setData({

kcxx: res.data

})

console.log(this.data.kcxx[0])

},

})

if (this.data.kcxx == []) {

wx.showModal({

title: '错误',

content: '数据拉取错误',

success: function (res) {

if (res.confirm) {

console.log('用户点击确定')

wx.navigateBack({

delta: 1

});

}

}

});

}

addbtn:function(){

var that = this;

const db = wx.cloud.database({

env: 'sjk-666'

})

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

data:{

\_id:app.appData.userip,

tcid:this.data.id,

kc:this.data.kcxx[0].kc,

sj:this.data.kcxx[0].sj,

qq:this.data.kcxx[0].qq,

xc:this.data.kcxx[0].xc

},

success:res=>{

console.log("下单成功",res)

that.setData({

shanchu:1

})

fail: err => {

that.setData({

shanchu:0

})

wx.showToast({

icon: 'none', title: '新增记录失败' })

console.error('[数据库] [新增记录] 失败：', err)

console.log(that.data.shanchu)

if(this.data.shanchu == 1){

wx.cloud.callFunction({

name: 'delete',

data: {

crid: this.data.id

},

success(res) {

wx.navigateBack({

delta: 1})

console.log("shuju", res.result.id)

console.log('调用成功')

},

fail() {

console.log('shibai')

}

})

}

},

cancel:function(){

wx.navigateBack({

delta:1

})

}

})

**3.6、模块4：数据存储模块**

3.6.1、功能描述

主要实现对数据的存储

3.6.2、输入数据

用户在上述模块输入及修改的所有信息都将输入存储模块

3.6.3、输出数据

在用户信息模块和浏览及预约模块中，存储模块内的特定信息将显示给用户用于增删查改

3.6.4、函数说明

调用API

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

avatarUrl: this.data.avatarUrl,

id: this.data.username,

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)

}

})

**4、 接口设计**

**4.1、外部接口**

云端和前端的连接：

const db = wx.cloud.database({

env: 'sjk-666'

})

**4.2、内部接口**

利用wx.navigateTo进行页面间的跳转

**4.3、接口说明**

1.注册和登录模块获取用户的输入信息，并将信息保存至云端

2.navigateto中url可以携带参数，进行页面间变量传递

# 5、数据库设计

**数据库**

表一 个人信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 昵称 | 密码 | 昵称 | QQ | 手机 |
|  |  |  |  |  |  |

表二 大佬信息

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学号 | 昵称 | 专业 | 授课科目 | 联系方式 | 空闲时间 | 自我介绍 |
|  |  |  |  |  |  |  |

表三 订单信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生学号 | 大佬学号 | 大佬联系方式 | 学生联系方式 | 授课时间 | 授课科目 |
|  |  |  |  |  |  |

# 6、系统出错处理

调用云平台api时，异步回调失败时调用fail，在控制台输出错误原因

例如：

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)

}

# 7、其他设计

安全设计：与学校校园网关账号和密码相连接，确保注册人员为校内人员。

此外，利用msgSecCheck可以检查一段文本是否含有违法违规内容

利用ImgSecCheck可检验一张图片是否含有违法违规内容

同时考虑为小程序添加举报功能可以充当一部分监管效果。