|  |
| --- |
| **浙江大学城市学院** |
| **详细设计（用户认证）** |
| **H5套圈小游戏** |

|  |
| --- |
| 项目名称 基于html5的休闲小游戏  组长 陈帆  组员 张荣阳、赵伟宏  小组号 G07  专业班级 软件工程1601  指导老师 杨枨 |



版 本 历 史

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 说明 | 作者 |
| 2018.05.18 | 0.1 | 初稿 | 张荣阳 |
| 2018.06.26 | 1.0 | 定稿 | 张荣阳 |

HIPO

文档编号：详设-2-用户登录认证系统

定稿日期：2018/6/26

子系统：用户登录认证系统

子系统概述：识别用户，获取用户信息

负责人：张荣阳

实现本子系统需要调动的资源：前端登录，开发方服务器暂存的服务器信息

子系统设计原则：（应参照系统设计原则稍作变动）

**注：H+IPO图 ——**

***开始工作前请先阅读概要设计文档-附录I-1 体系结构设计报告，尤其是系统总体架构。***

*H代表Hierarchy 层次*

*IPO代表 Input Process Output 输入输出处理*

*所以首先要有这个模块的层次图，再对这个层次图中的各个最小单元进行IPO分析，IPO表中的各项内容解释参考放于同文件夹下的详细设计文稿。*

*简而言之，就是一张结构层次图加上若干张IPO表。*

*下面将附上一份根据这份模板改造的一张IPO图，若不会使用Visio（当然最好是学会，还是挺简单的），可以画一张手绘的层次图然后发给我，由我来代画。*

*HIPO根据我的理解主要还是面向过程的一种设计方法，如果不适应可以自行研究面向对象的设计方法提交相应的文档即可。*

*详细设计的方法，详见书本。*

*六个子系统：*

*1、资源子系统 -YaoT*

*2、账户子系统 -Czf*

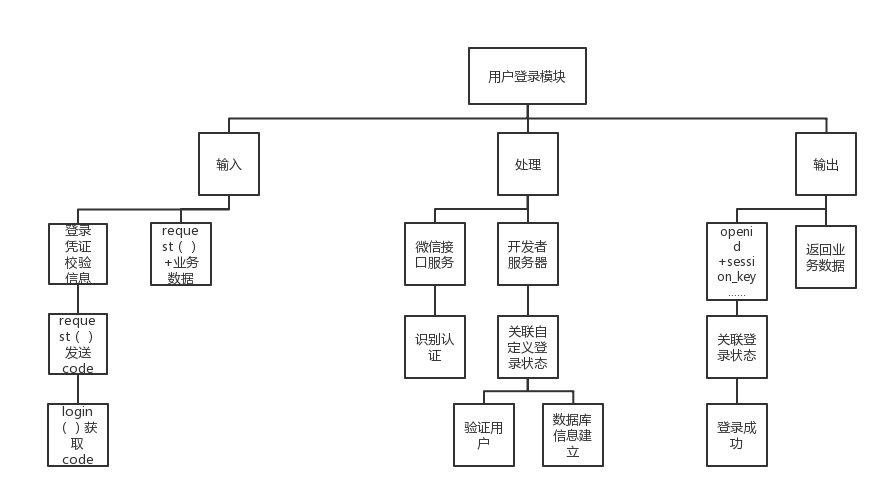
*3、评论子系统 -Wsq*

*4、~~前端整合子系统~~*

*5、~~框架自带全权后台管理子系统~~*

*6、~~日常维护子系统~~*

# H图



# IPO表-1-小游戏部分

（可做缩进以适应层次图结构）

|  |  |
| --- | --- |
| **小游戏部分** | |
| 1. 程序描述：   位于服务器端，关于游戏结束后从数据库将用户上次游戏的积分调出 | |
| 1. 功能：   为当前用户显示曾经游戏的积分记录（仅仅是最高分记录） | 1. 性能（可选） |
| 1. 输入：   用户相关信息认证（有游戏记录） | 1. 输出：   用户曾经游戏的最高记录（仅仅是针对积分） |
| 1. 流程逻辑   C:\Users\Administrator\AppData\Roaming\Tencent\Users\905600387\QQ\WinTemp\RichOle\EH@LF)]M36VVFJWW1YSPG%X.png | |
| 1. 注释设计 | 1. 限制条件：用户有过游戏记录 |
| 1. 接口：   C:\Users\Administrator\AppData\Roaming\Tencent\Users\905600387\QQ\WinTemp\RichOle\NWNKO2~~AM~[9ZQH}{$T(2A.png | |
| 1. 测试计划：待定 | |
| 1. 尚未解决的问题：暂无 | |
| 1. 设计说明：…… | |
| 编号：01  设计者：张荣阳 | |

# IPO表-2-开发者服务器

（可做缩进以适应层次图结构）

|  |  |
| --- | --- |
| **开发者服务器** | |
| 1. 程序描述：   位于服务器端，关于游戏结束后从数据库将用户最新游戏的积分和皮肤信息更新 | |
| 1. 功能：   每次用户游戏结束后自动将相应信息更新（积分，皮肤） | 1. 性能（可选） |
| 1. 输入：游戏端传入相应数据 | 1. 输出：将需要更新的信息导入 数据库 |
| 1. 流程逻辑：   C:\Users\Administrator\AppData\Roaming\Tencent\Users\905600387\QQ\WinTemp\RichOle\M[6[$2C67QQBB}RTSK%VSX4.png | |
| 1. 注释设计：无 | 1. 限制条件：用户登录 |
| 1. 接口：   C:\Users\Administrator\AppData\Roaming\Tencent\Users\905600387\QQ\WinTemp\RichOle\NWNKO2~~AM~[9ZQH}{$T(2A.png | |
| 1. 测试计划：手动 | |
| 1. 尚未解决的问题：服务器有上限，无法解决 | |
| 1. 设计说明：…… | |
| 编号：02  设计者：张荣阳 | |

# IPO表-3-微信服务接口

（可做缩进以适应层次图结构）

|  |  |
| --- | --- |
| **微信服务接口** | |
| 1. 程序描述：   位于服务器端，关于游戏结束后从数据库将用户最新游戏的积分和皮肤信息更新 | |
| 1. 功能：   每次用户游戏结束后自动将相应信息更新（积分，皮肤） | 1. 性能（可选） |
| 1. 输入：游戏端传入相应数据 | 1. 输出：将需要更新的信息导入 数据库 |
| 1. 流程逻辑：   C:\Users\Administrator\AppData\Roaming\Tencent\Users\905600387\QQ\WinTemp\RichOle\QCT~S$QOZE~PNHP@SDNP2O1.png | |
| 1. 注释设计 | 1. 限制条件：用户登录 |
| 1. 接口：   C:\Users\Administrator\AppData\Roaming\Tencent\Users\905600387\QQ\WinTemp\RichOle\NWNKO2~~AM~[9ZQH}{$T(2A.png | |
| 1. 测试计划：待定 | |
| 1. 尚未解决的问题：暂无 | |
| 1. 设计说明：此模块直接调用微信现有接口。 | |
| 编号：03  设计者：张荣阳 | |

#小游戏

var openId = (wx.getStorageSync('openId'))

if (openId) {

wx.getUserInfo({

success: function (res) {

that.setData({

nickName: res.userInfo.nickName,

avatarUrl: res.userInfo.avatarUrl,

})

},

fail: function () {

// fail

console.log("获取失败！")

},

complete: function () {

// complete

console.log("获取用户信息完成！")

}

})

} else {

wx.login({

success: function (res) {

console.log(res.code)

if (res.code) {

wx.getUserInfo({

withCredentials: true,

success: function (res\_user) {

wx.request({

//后台接口地址

url: 'https://....com/wx/login',

data: {

code: res.code,

encryptedData: res\_user.encryptedData,

iv: res\_user.iv

},

method: 'GET',

header: {

'content-type': 'application/json'

},

success: function (res) {

// this.globalData.userInfo = JSON.parse(res.data);

that.setData({

nickName: res.data.nickName,

avatarUrl: res.data.avatarUrl,

})

wx.setStorageSync('openId', res.data.openId);

}

})

}, fail: function () {

wx.showModal({

title: '警告通知',

content: '您点击了拒绝授权,将无法正常显示个人信息,点击确定重新获取授权。',

success: function (res) {

if (res.confirm) {

wx.openSetting({

success: (res) => {

if (res.authSetting["scope.userInfo"]) {////如果用户重新同意了授权登录

wx.login({

success: function (res\_login) {

if (res\_login.code) {

wx.getUserInfo({

withCredentials: true,

success: function (res\_user) {

wx.request({

url: 'https://....com/wx/login',

data: {

code: res\_login.code,

encryptedData: res\_user.encryptedData,

iv: res\_user.iv

},

method: 'GET',

header: {

'content-type': 'application/json'

},

success: function (res) {

that.setData({

nickName: res.data.nickName,

avatarUrl: res.data.avatarUrl,

})

wx.setStorageSync('openId', res.data.openId);

}

})

}

})

}

}

});

}

}, fail: function (res) {

}

})

}

}

})

}, complete: function (res) {

}

})

}

}

})

}

},

globalData: {

userInfo: null

}

PDL

#服务器端

procedure <处理来自/remark/delete?id=的请求> interface<http请求对象> is

begin

if <从request对象的session中得出没有登录> then

重定向到‘非法操作’页面

else if <请求不是get请求> then

raise 404异常

else #请求是get请求

if <请求用户不存在> then

返回用户未找到出错页面

begin <整合信息>

将code和服务器小程序相关id等等准备发送至微信服务接口

end

微信服务接口接收信息成功，产生相应反馈

用户登录成功

end

修改说明：服务器端处理前端发送过来的登录注册请求并且响应，不再获取微信接口。