小程序微服务应用系统

**系统由五部分组成**

*公共服务管理平台(后台管理端)、商家服务平台(用户端)、微信小程序综合平台、微信公众号服务平台(暂时不做)、文件服务平台*。

1. 公共服务管理平台（commPtatform）

项目地址：https://github.com/tanbow1/commpt.git

「公共服务管理平台」（commPtatform）是所有应用的核心，包含管理类代码、业务类代码、小程序后端服务代码、微信公众号后端服务代码。不对外开放，仅供管理员使用。

管理类代码，针对所有系统进行统一管理维护。

1. 代码包含管理、系统自身相关JAVA后台及页面，此处即**系统设置、代码表维护、系统分类**中相关功能代码；

含以下菜单项：

系统设置（该平台下的一些个性化配置）

代码表维护

1. 国家地区（系统中国家地区代码表）
2. 文件管理（系统中所有已存储的文件一览）
3. 账户类型（系统中账户类型代码表，即邮箱、qq、支付宝等）
4. 日志类型（系统中存储日志时使用的类别代码表）
5. 商品分类（微信小程序中呈现的商品分类代码表）
6. 证件类型（系统中证件类型代码表，可配置多系统代码表不一致）
7. 支付类型（系统中支付类型代码表维护）

业务分类

1. 商品管理（系统中需要呈现商品的地方，配置其商品显示属性）
2. 单位管理（接入本系统中的小程序所属单位审核，管理）
3. 在接入／进驻本系统时，需审核应用所有者；
4. 应用所有者管理
5. 应用管理（接入／进驻本系统中的应用审核、管理）

本系统以应用为单位。

**入驻**：

1. 对于只是简单的在本系统中呈现部分信息的，入口布置在「微信小程序综合平台」统一的应用入口；
2. 对于只需要一个入口，本身已经有完整的小程序，入口与上述一样。

**接入：**该功能需要针对性的进行开发，除与**入驻**类应用呈现在一起以外，在「微信小程序综合平台」也会有专门的入口部署，以便更加快速的找到。

系统分类

1. 菜单管理（系统中菜单代码表，包含本系统自身菜单树和其他地方使用到的菜单树维护）
2. 用户管理（用户维护，系统中需要登录的地方采用统一用户，权限管理，即绑定菜单项；用户地址、账号管理）
3. 角色管理（用户角色、角色菜单树、用户角色表维护）
4. 系统日志（系统中业务类操作日志记录表）
5. 参数配置（系统中需要灵活配置参数的参数管理表）
6. 表管理（数据库表结构一览）

业务类代码

1.代码包含业务分类操作相关的JAVA后台及页面，此处即**业务分类**中相关功能代码；

2.「商家服务平台」的后端服务JAVA代码；

3.「微信小程序综合平台」的后端服务JAVA代码；

4.「微信公众号服务平台」的后端服务JAVA代码；

1. 用户端商家服务平台（commPtatformWeb）

项目地址：https://github.com/tanbow1/commptWeb.git

该平台对外开放，提供给需要入驻的商家和普通用户使用。与「微信小程序综合平台」配合使用。未登录下，展示网站首页，注册登录后，根据用户权限开放功能。

用户分类：

1、商家：具有普通用户的功能，另外也包含应用拥有者的管理功能；

2、普通用户：输入基本信息注册成功后即可使用。

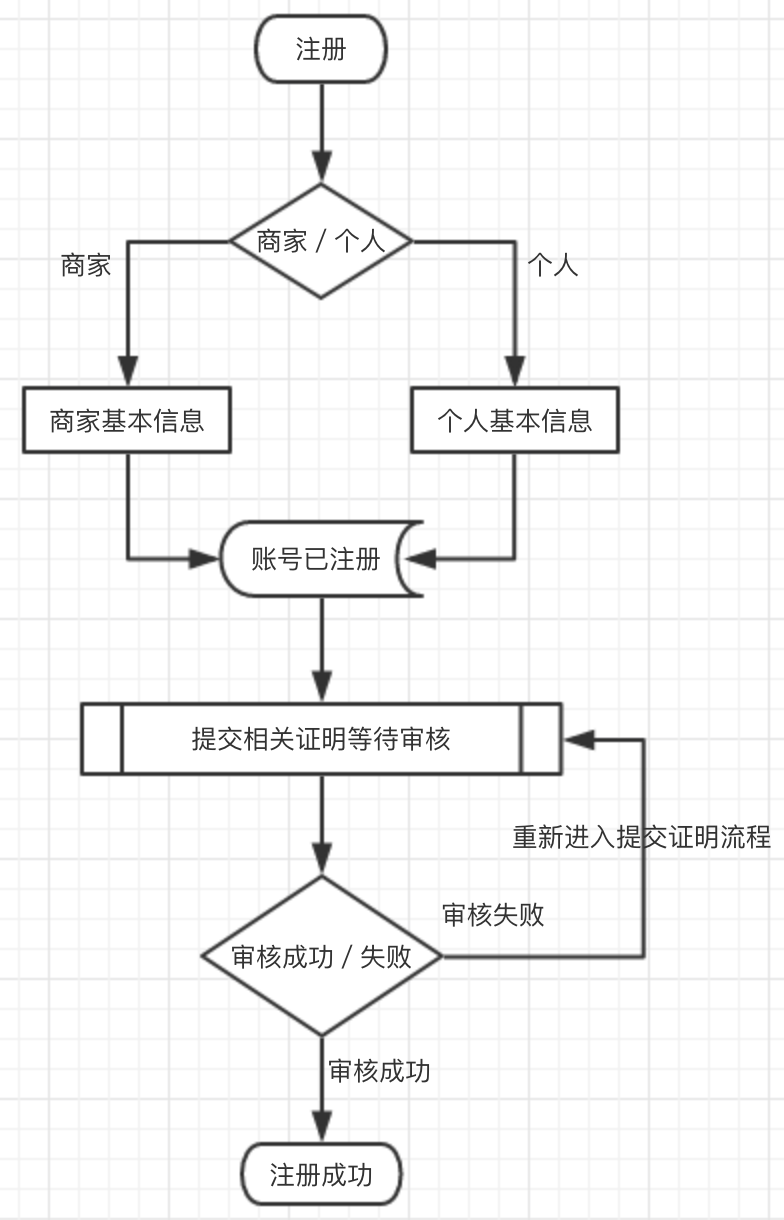
以下为一个完整的注册过程，不合理的待完善：

用户注册时，选择 个人／商家 后需提交相应资料，提交审核，审核通过方可进行应用入驻／接入。发布应用后，用户获得应用相关管理功能。

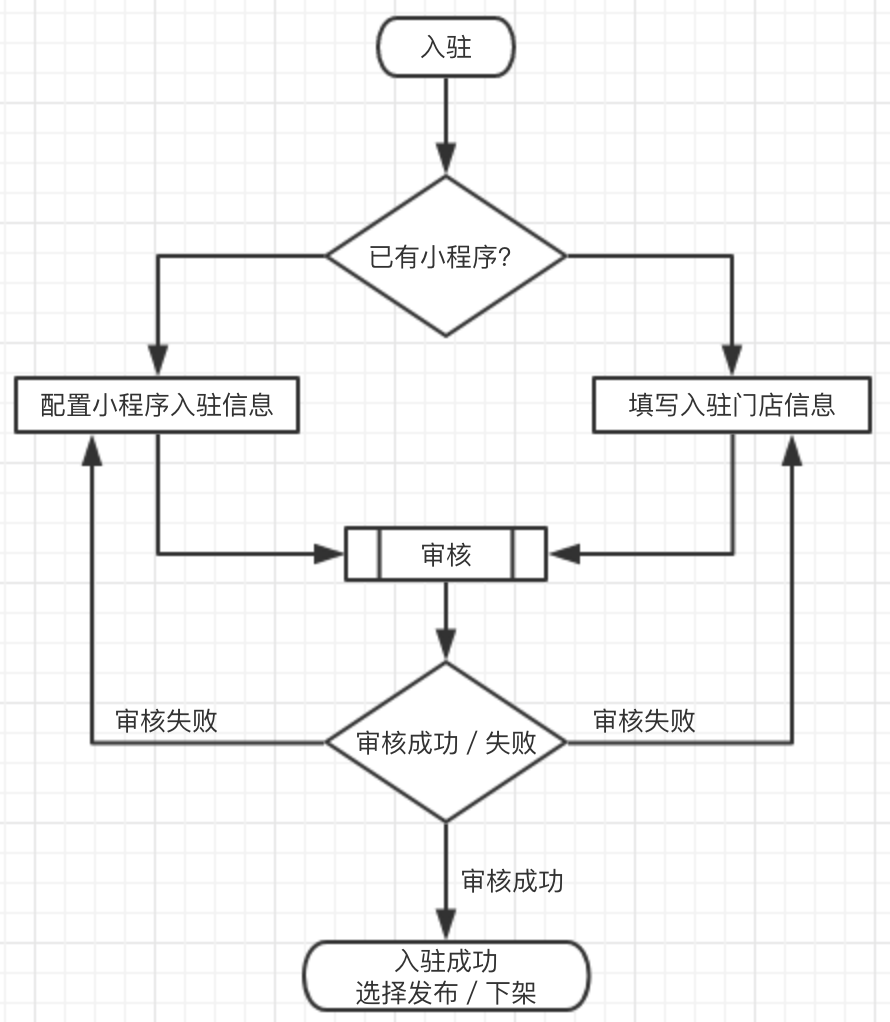
注：此处身份选择商家／个人 字段留作备用，暂时无区别。

A）相关流程：

**1.注册流程**



**2.入驻流程**



**3.接入流程**

联系客服人员，确认开发需求或提交诉求，留下联系方式。

B）用户菜单项：

系统设置

1. 通用设置
2. 用户信息（注册用户信息维护）
3. 登录密码
4. 社交账号绑定

绑定后支持该账号／密码登录和社交APP扫码登录

1. 安全中心

a)最近登录（登录时间 地点）

b)操作日志

c)登录授权（可以配置多个登录账号进行授权）

1. 帮助

应用管理（对应用进行发布／下架／编辑／删除／数据统计等操作）

1. 我的发布

a)已发布应用

b)待发布应用

1. 访问量（统计每个应用访问数据，所有应用访问总数据）
2. 交易量（统计每个应用交易数据，所有应用交易总数据）

我要发布（提交入驻小程序／门店申请，提交应用接入申请）

订单管理（此处订单包含接入的小程序应用内所有订单相关的管理及发布过程中产生的费用，所以有时候需明确指定某个应用后进入订单管理比较合适 ）

1）全部

2）待付款

3）待发货

4）退款／售后

5）待处理订单

6）已完结订单

我的消息

首先生成的订单会在此处弹出信息提示，此处提供所有信息展示。

**C）网站基本功能（对外核心业务）**

通过查看搜索分类门店等信息获得自己想要的信息去消费。

包含商品展示 选购 微信支付 等线上功能。

具体功能和页面 待定。

1. 微信小程序综合平台

项目地址：https://github.com/tanbow1/commptMiniapp.git

本平台只存在于微信小程序端，定位为所有小程序及其门店信息展示入口的集合。提供给消费者浏览本平台商家发布的所有小程序及门店信息。此处根据应用分类来进行消费者展示。同时商家／个人有需要也可在此提交**入驻／接入**申请。入驻流程略有区别，整合了pc端的注册和入驻流程，接入流程维持不变。该平台使用微信自动登录。

小程序包含功能：

屏1:同城

定位 搜索框 展示分类 附近门店列表（由近到远排序）

1. 通过关键词搜索
2. ~~图标展示时下使用量较高的9类+最后一个全部~~
3. ~~附近综合列表分2拦（最新、最近），默认综合（即：系统默认存在一套展示规则，付费优先）~~

2）展示常用5个分类 + 12个主分类

3）列表展示分3个排序（综合、最近、最新）

浮动框展示「入驻」入口。

屏2:~~门店~~ =>发现

~~关键词搜索~~

~~所有门店分类展示，类似同城中‘全部’~~

~~浮动框展示「入驻」入口~~

展示分类：

关注

我关注的内容,可以看着是收藏功能，可编辑删除。

关注细分：商家／商品／内容 ，目前不区分，列表放在一起。

推荐

爱美食

天天特价

限时好礼

每日精选

…

小程序

此处展示小程序使用记录和所有与本平台有关联的小程序

此处小小程序不仅仅指代微信小程序，也可以是一个跳转的链接。

此外，主体页面默认展示平台的推送信息+我关注的最新动态，根据上述3类点击后跳转至具体页面。

屏3:~~推荐（精选）~~=>订单

~~此处展示开发接入的和付费入驻的门店应用列表，支持通过商品分类等条件定位门店搜索框。此处展示的每一条应该是完整的小程序应用入口，对于只需简单展示的入驻门店不支持在此显示。~~

全部／待付款／处理中／已成功

屏4:我的=>剔除关于订单部分，转至屏3

顶部：微信头像／昵称 ，可点击进入明细：

头像

姓名

昵称

性别

生日

手机号码

证件类型

证件号

国籍（地区）

所在地区

详细地址

证件资料照片

~~1）我的收藏~~

~~2）我的订单~~

~~a)全部订单~~

~~与用户关联的所有订单信息展示~~

~~b)待处理订单~~

~~与用户关联的所有未完结订单信息展示~~

~~3)我的发布~~

~~我发布的所有应用 和~~ **~~入驻／接入~~**~~入口~~

~~4）账户与安全~~

~~会员名~~

~~绑定手机号~~

~~修改登录密码~~

~~5）收货地址~~

~~收获人-联系电话-所在地区（选择）-详细地址-邮编 列表~~

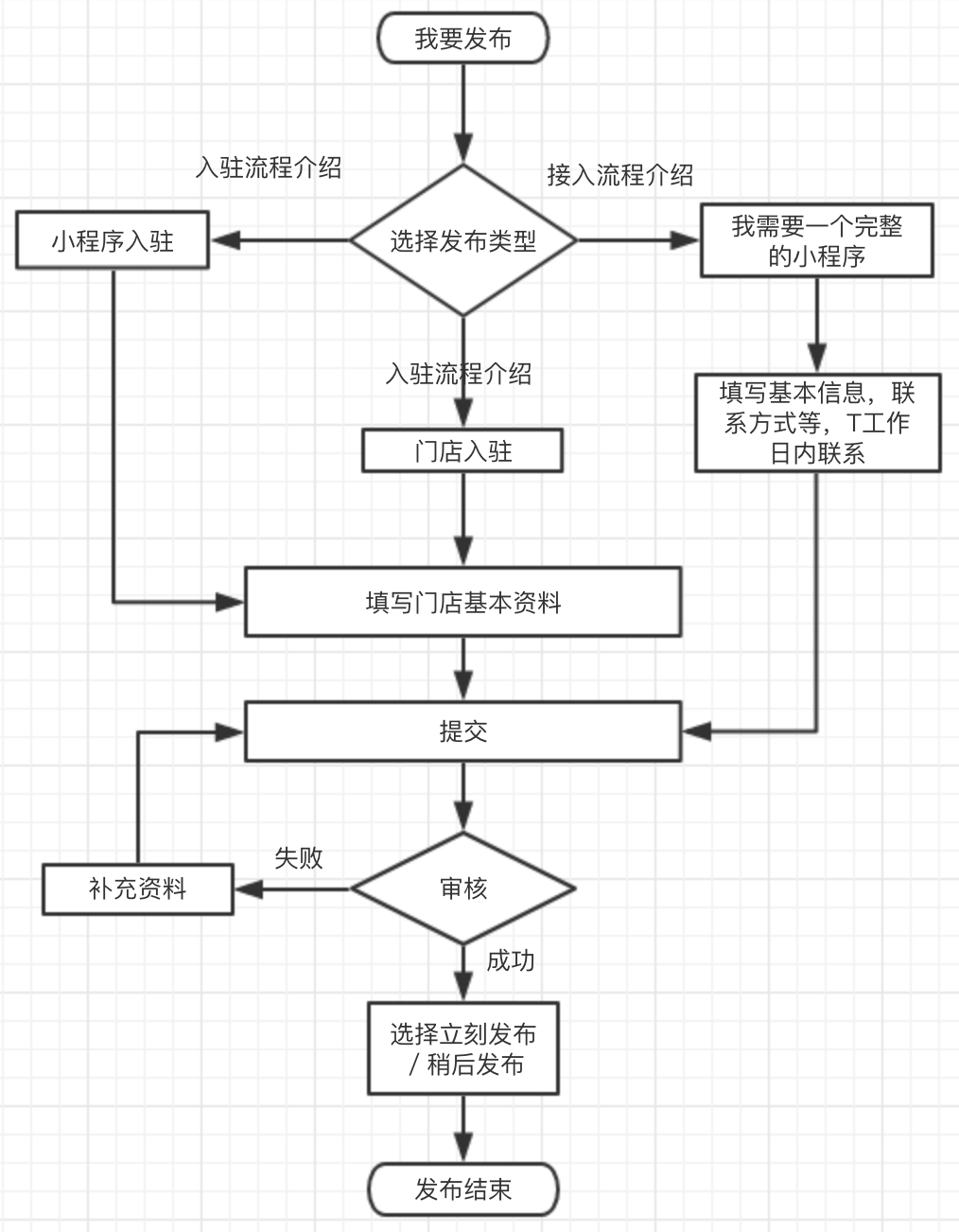
~~//5）清除缓存~~

我的账单（账单统计+记账功能）／我的收藏／我要发布（发布应用／店铺／单一消息）

我的发布／我的店铺／我的应用

常见问题／联系客服

发布流程：



1. 微信公众号服务平台

平台目前暂时用作推广，后续有时间再开发。

1. 文件服务平台

与上述「文件管理」关联，此处用于提供不同方式的文件存储服务。

如果有可能的话，可以做成一个类似云存储。

**详细设计**

1. 开发

「公共服务管理平台」采用springMVC+mybatis／easyui bootstrap等开发，包含所有后台代码和**系统设置、代码表维护、系统分类、业务分类**四部分管理代码。

「**商家服务平台」采用vueJs bootstrap等开发，纯前端项目。**

**「微信小程序综合平台」采用微信小程序开发。**

「微信公众号服务平台」采用微信公众号开发。

「文件服务平台」采用FTP存储大尺寸、下载资源文件；nginx存放小图片，小文件等。

数据存储：Oracle／redis

「公共服务管理平台」部署Weblogic上，

「商家服务平台」部署在nginx上。

1. 页面设计

主要是针对「商家服务平台」和「微信小程序综合平台」相关页面设计。

小程序：



商家服务平台：

**其他**

小程序进入逻辑：

wx.checkSession({

success:**function**(){

/\*session有效，继续后续操作

如调用本地(commpt)业务需从wx小程序storage获取3rd\_session此处即accessToken，本地系统登录态完全失效则需要在wx小程序中重新登录以获取最新accessToken。

如未获取到accessToken则需要重新登录。

\*/

},

fail:**function**(){

//session失效 用户登录态失效， 发起登录

wx.login({  
 success: function(res) {  
 if (res.code) {

*/\*获取code后发起网络请求*wx.request将code发送给commpt本地平台，本地平台*请求url：*https://api.weixin.qq.com/sns/jscode2session?appid=APPID&secret=SECRET&js\_code=JSCODE&grant\_type=authorization\_code\*/

*/\*成功则会获取到*openid和session\_key, unionid根据具体情况看是否返回

失败则会获取到errcode和errmsg

成功获取到openid、session\_key后，公共服务平台生成随机session作为第三方3rd\_session，维护小程序在本系统中的登录状态。此处即为该用户唯一标识openid绑定的用户userId下jwt生成的accessToken。accessToken本地生成或刷新后发送到wx小程序端存入storage，调用本地(commpt)业务发起wx.request时带上参accessToken。

注：session\_key不回传到wx小程序端，如有需要则保存在本地平台中用于后期用户数据解密key。

\*/

} else {  
 //获取用户登录态失败，退出小程序重新进入。  
 }  
 }  
});

}

})

小程序授权逻辑：

获取某个权限：

wx.getSetting({  
 success:**function** (res) {  
 **if** (!res.authSetting[**'scope.userInfo'**]) {  
 wx.authorize({  
 **scope**:**'scope.userInfo'**,  
 success:**function** () {  
 **console**.log(**'同意'**);  
 },  
 fail:**function** () {  
 **console**.log(**'不同意'**);  
 }  
 });  
 }**else** {  
 **console**.log(**'已授权过了'**);  
 }  
 }  
});

|  |  |
| --- | --- |
| wx.getUserInfo | 获取用户信息 |
| avatarUrl | 头像链接 |
| city | Changzhou |
| country | China |
| province | Jiangsu |
| gender | 1 |
| language | zh\_CN |
| nickName | 懂事长 |

|  |  |
| --- | --- |
| wx.login | 登录 |
| code | js\_code |