# 用户会员消费码设计

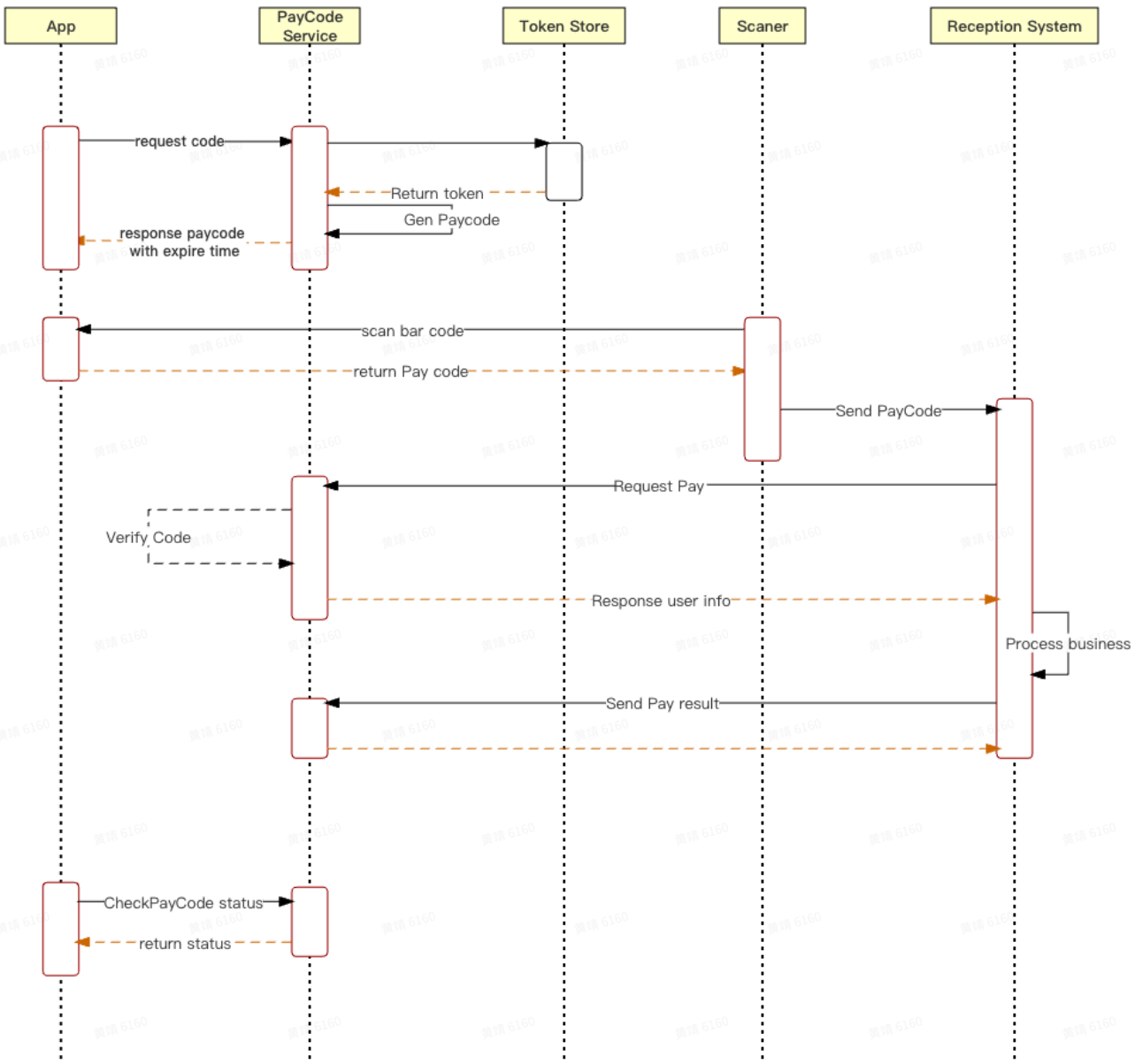
## 需求产品PRD

## 目标

用户在住店过程中出示用户消费码，消费码实际是一个20位数字，数字算法后面介绍。

通过这个20位数字生成二维码和条形码（Code128C），前台用扫描枪扫描后，调用消费码服务获取用户信息，从而后续实现业务处理。

## 总体流程

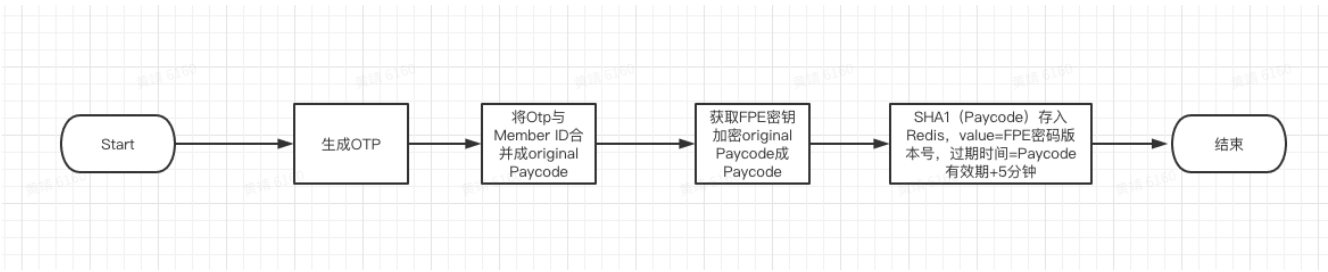


# PayCode生成算法以及流程

PayCode由2位前缀数字+18-22位业务码组成

前2位为场景码，按照请求生成Paycode时传入指定的场景。

生成的PayCode做SHA1处理后当Key存入Redis中，value为业务码的密钥版本号。




## 业务码生成规则

业务码由 memberid（为了兼容memberid中最后一位是字母的用户，memberid最后2位为字母码，a-z按01-26来编码，00代表memberid中没有字母，不区分大小写）+8位动态口令码（以下简称OTP），先通过康托尔配对算法将2个数字组合成一个，再通过FPE FF1加密算法加密后生成。

康托尔配对Java实现 https://github.com/majehuang/cantor_pairing/blob/main/Cantor.java

## FPE FF1密钥版本管理

提供生成密钥接口，每次生成后，存入数据库，版本号为时间戳。

加密时获取版本号最大的密钥，进行加密。

## OTP生成规则

参考 https://github.com/jchambers/java-otp

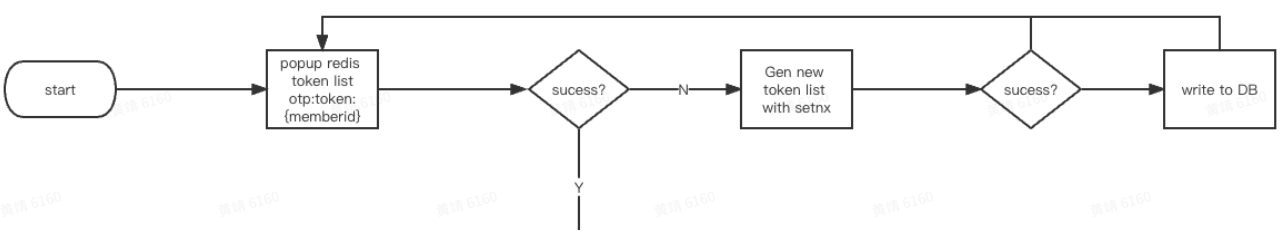
https://zhuanlan.zhihu.com/p/484991482

OTP生成算法参照RFC6238标准，包含如下要素

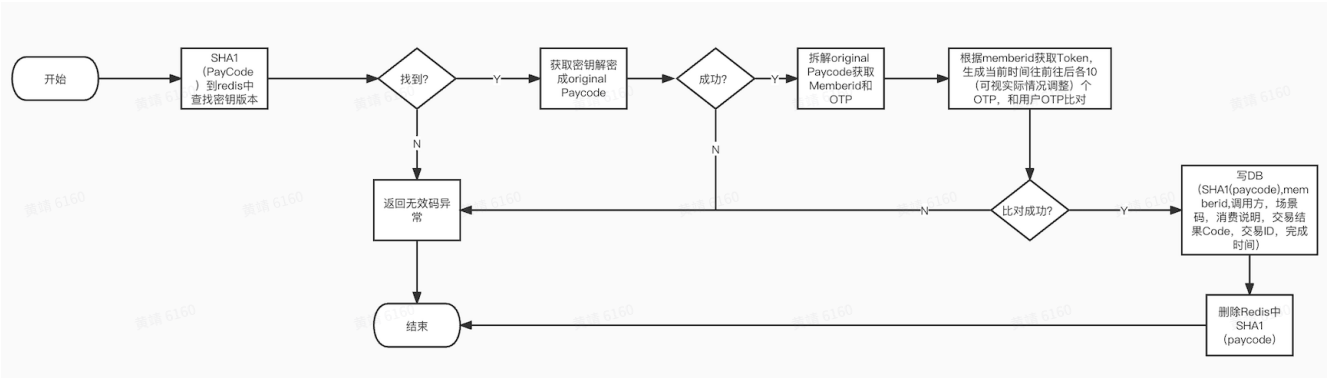
Token：服务端为每个用户生成的密钥，每个密钥只能生成20次动态口令，超过20次需要更换。

Time step: 默认为 60秒

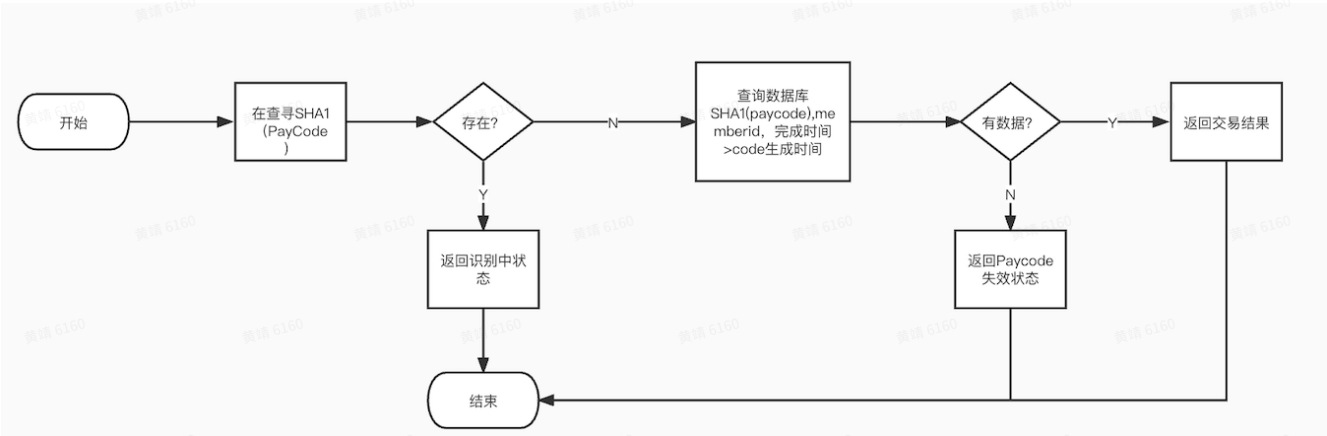
## Token获取流程



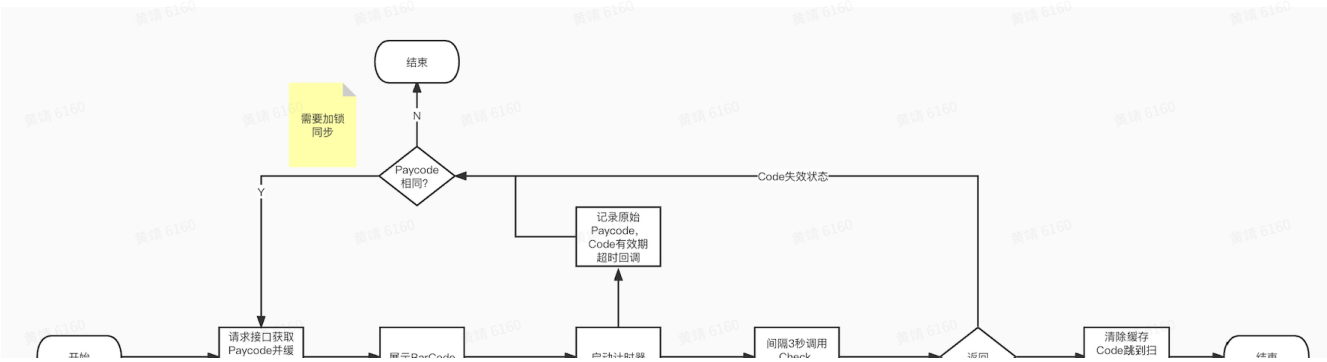

## Verify PayCode流程

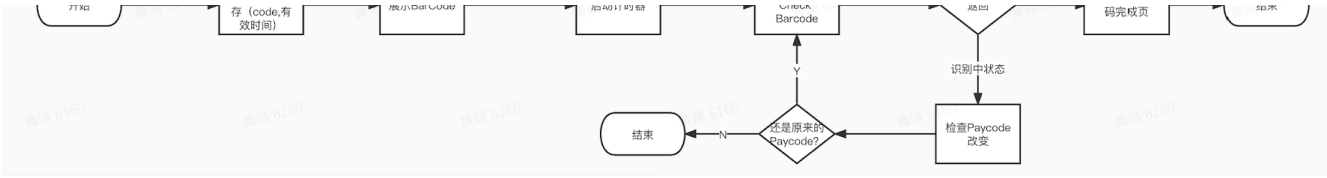



## 服务端CheckPayCode流程




## 客户端展示Paycode流程






## 接口整理

### 获取消费码

@needLogin

//场景码 sceneCode

PaycodeContent{ String paycode,long expiredTime //unixtimestamp}

PaycodeContent getPayCode(String sceneCode );

### 扫码获取用户信息(内网接口)

UserInfo consumeByScanPassivity(String payCode, int consumeAmount //, String businessId,int businessType,String tradeRemark) ;

### 检查Paycode状态

@needLogin

Response getPaycodeStatus(String paycode);

### 生成FPE密钥(内网接口)

Response genSecretKey();