信用卡系统介绍

我们的项目分为前台和后台两个子系统。

前台系统是客户端系统

在该系统用户可以提出信用卡办理申请和额度提升办理。

用户还可以在该客户端查看自己的基本信息，管理自己的信用卡及与其绑定的银行卡。可以实现信用卡的还款和流水查询。

后台系统使用人员为银行柜台操作人员和大堂经理级别的人员。

在该部分的系统中后台管理人员可以查看并处理前台用户的请求信息。后台管理人员还可以查看单个用户的基本信息，处理或修改单个用户的基本信息，如：用户到柜台申请信用卡的激活，信用卡密码的修改等等。后台管理员还可以查看所有用户的分期情况和逾期情况。并做出相应的操作，如：发送短信提醒，或者查询该用户的基本信息看是否符合拉入黑名单的条件。

后台系统还有一种属于超级管理员的模式。当使用超级管理员的账号登入时，系统的功能将会在普通后台管理员的操作界面基础上增加各表的打印报表功能（如：所有用户的分期情况及总额报表，银行总流水报表）和数据字典管理功能。

其中：

办卡方式有用户在客户端申请办卡，用户直接到柜台请求办理信用卡两种方式。但是信用卡的激活能且仅能在柜台上面激活。后台会根据用户填写的详细信息进行额度打分，然后进行额度等级判定来设置最高透支额度。

额度提升：我们有两种额度提升方式。其一，永久额度提升，其中包括用户可以在办理信用卡后的三个月后提出信用卡额度提升申请，也可以让系统在每三个月对用户的信用及用卡情况进行相应的额度提升操作。该种额度提升不会降低用户的额度（逾期用户及黑名单用户除外）。其二，临时额度提升，其中包括发卡行为促销信用卡，提升用户质量及增加用户量，银行可以在某段时间内提升用户的额度，待时间截止后额度自动回退到原来的额度。

分期还款：分期还款我们有多个等级的还款方式，如（分3期，分6期）用户可以选择分期期数进行还款。但是前提要还足这个月的最低还款金额（最低还款=逾期滞纳金+逾期日息+逾期本金+上月未还+上月未还利息+提现+提现手续费+提现利息+使用的临时额度+当期分期应还+当前分期应还手续费+(本期额度内消费金额)\*10%）

逾期：我们暂定信用卡还款，每个月3号为出账单日（即显示你这个月该还多少钱），23号为最后还款日。本系统支持全额还款、最低还款以及分期还款。如果前面所诉还款均未满足则23号后该用户将属于逾期还款，后台将会将该用户的进本信息存入逾期表，待后继处理。

此次项目中用到的一些技术点有：

1. 工具使用的是支持Springboot开发的STS软件
2. 服务器使用的是tomcat9.0.16，并且使用了Nginx反向代理实现了分布式集群，加大了系统服务器的可维护性和服务器整体性能；还通过Nginx将项目文件进行了动静分离，静态文件放在Nginx指定的目录下，减轻了服务器的压力；最后通过Nginx实现了数据共享，解决了分布式系统下session不一致的问题。初次之外，我们还运用了Redis作为二级缓存和实现session功能。Redis搭建了主从机制，主Redis能进行读写操作，而从Redis只能进行读操作。当主Redis进行写操作后，从Redis会自动同步数据。同时我们还配置了哨兵机制，提高了Redis的可用性和稳定性。对于会经常用到的数据，如数据字典里的数据，我们会提前存到Redis里面，当用户访问时，就不需要访问数据库，直接从Redis里获取，大幅度减轻了服务器的压力。
3. 数据库我们使用的是Oracle 11g，关于数据库方面，我们对数据量巨大的表格如：用户流水，建立了数据库时间分区以及数据库索引优化，大幅度加大了数据读取方面的速度与性能。为了杜绝在代码实现中途出现了异常但是修改了数据库数据的情况，我们在代码中加入了Springboot+mybatis提供的@Transactional注解来实现事务管理，使得事务部分的所有sql语句成为一个整体。从而使得数据库数据的准确性和完整性。
4. 代码管理中心，我们是用的是svn
5. 技术方面，我们使用了SpringBoot 2.13+SpringCloud G来搭建的项目。
6. 依赖：我们使用了redis实现session共享的两个依赖，导出excel报表的poi依赖，分页插件依赖和一些其他的基本的依赖。
7. 接口方面，我们的项目使用了信用卡四要素验证api，百度地图地理定位api，短信验证码api和短信模板发送api等