二维码支付系统

课程名称：综合训练

专业班级：1615204

学 号：161520327

姓 名：兰帅

同 组 人：朱文勇 覃燕婷

指导老师：许峰

成 绩：



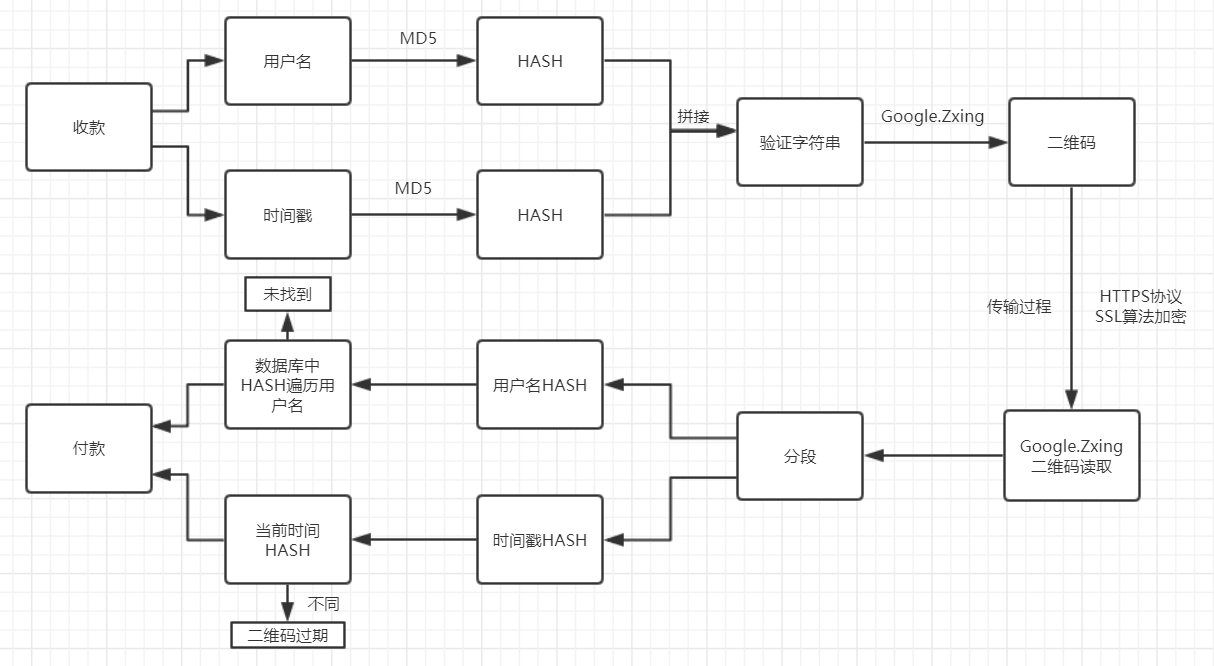
南京航空航天大学

2018年12月28日

1. 产品设计思想

模拟基于加密的二维码实现电子交易的系统，实现交易的迅速便捷，可以在web浏览器和移动端同时使用。采用Google.zxing接口实现二维码的生成和解析，采用加密协议传送，产品类似webapp形式呈现。开发利用DAO的设计模式，由javabean的VO，预先设定各项操作的接口类，真正实现操作的代理类和集成各项代理的工厂类，该设计模式使工程构造简明清晰，并且减少了耦合度。

1. 采用的安全技术
2. 采用Google的zxing二维码生成与读取接口对信息进行加密，可以将支付信息保存入二维码中。
3. 采用MD5加密算法，获取用户标识的不可逆散列值。
4. 使用时间戳，限制二维码使用周期。
5. 传输过程中使用SSL加密传输，保证传输过程安全。
6. 验证过程



如图所示，收款人提出收款后，根据其当前的用户名和时间戳分别产生HASH散列值，将散列值相拼接，然后使用Google的Zxing工具转化为二维码；付款人拍摄到二维码后，根据固定位数分离出收款人用户名的散列值和二维码生成时间的散列值，在数据库中遍历所有用户的的用户名的MD5散列值与目标对比，若匹配，则找到收款人，否则没有收款目标，拒绝收款，并打印警告；再使用当前时间段的时间戳生成MD5散列值，与目标MD5散列值比较，若相等，则说明支付时间与收款码生成时间在同一个时间间隔中，否则二维码过期，不进行交易。只有在找到收款人且二维码未过期的前提下才进行交易。

后来答辩时根据老师的想法增加了Tomcat的传输过程安全保护机制，又在本地使用JDK 自带的 keytool工具生成证书，将证书应用于Tomcat，在Tomcat服务器中配置启动了HTTPS协议替代HTTP协议，使用8443端口替代了8080端口，传输过程中的数据是用SSL协议加密后的密文，这样就增强了传输过程中的安全性。

1. 主要数据结构

1.用户类：用来存储用户信息

**public** **class** user {

**private** String username;

**private** String password;

**private** **double** balance;

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

}

2.历史支付记录类：用来存储

**public** **class** payhistory {

**private** String payer;

**private** String accepter;

**private** **int** money;

**private** Timestamp time;

**public** String getPayer() {

**return** payer;

}

**public** **void** setPayer(String payer) {

**this**.payer = payer;

}

**public** String getAccepter() {

**return** accepter;

}

**public** **void** setAccepter(String accepter) {

**this**.accepter = accepter;

}

**public** **int** getMoney() {

**return** money;

}

**public** **void** setMoney(**int** money) {

**this**.money = money;

}

**public** Timestamp getTime() {

**return** time;

}

**public** **void** setTime(Timestamp time) {

**this**.time = time;

}

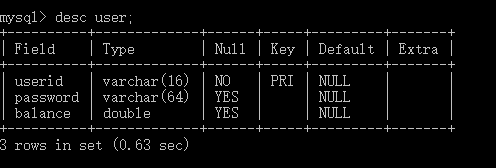
}

1. 主要数据库表
2. 用户表

Userid:用户标识；

Password:用户密码；

Balance:用户账号余额；



1. 支付历史记录表

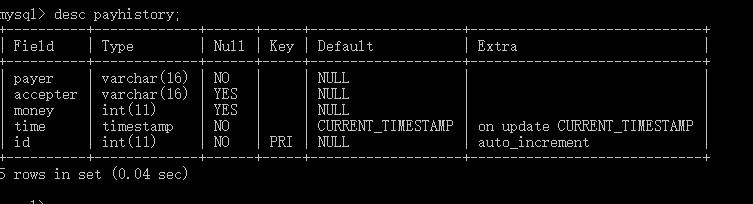
Id:主键，唯一标识一次记录，无实际含义，自增；

Payer:付款人用户标识；

Accepter:收款人用户标识；

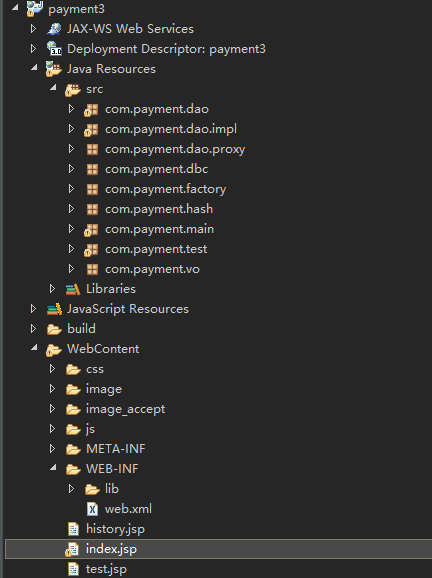
Money：交易金额；

Time：交易完成时间



1. 主要代码解析

本项目使用DAO设计模式，代码框架如下



Vo中保存映射数据表的简单java类；dao定义需要实现操作的接口；Impl中是接口的真实实现类，完成的是数据库的具体操作，但是不负责数据库的打开与关闭；Proxy，代理类，主要是完成数据库的打开、关闭，并且调用真实实现类对象操作；factory，工厂类，通过工厂类取得一个DAO的实例化对象；main，实现具体的服务。

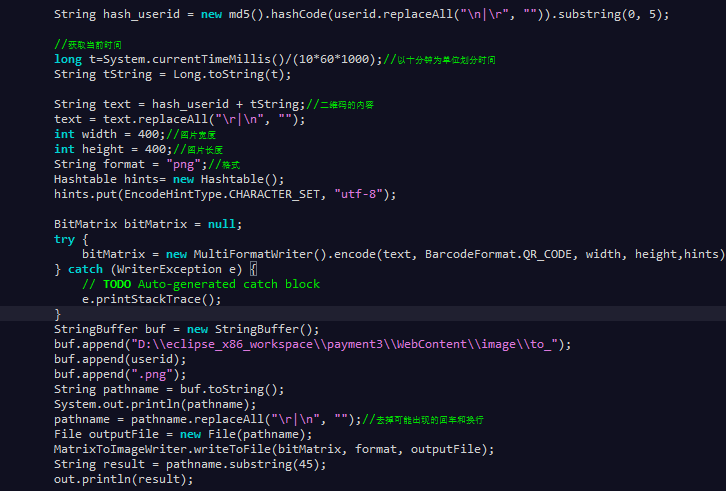
Css保存样式文件，js保存JavaScript文件，image保存生成的二维码图片文件，image\_accept保存需要解析的收款二维码的照片/图片，web.xml是servlet映射文件，lib中保存引入的jar包。

实现的主要功能如下：

1. 收款：

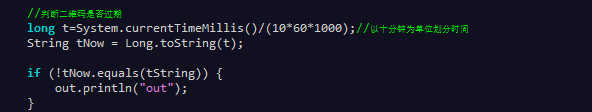
收款功能通过生成与收款人和交易时间相匹配的二维码来实现。用户登录之后使用用户名进行MD5加密，获取对应的散列值，再加上当前时间（使用currentTimeMillis方法获取，其得到与1970年一月一日之间的时间差，进而得到当前所在的十分钟的时间标识）。以字符串的形式与用户散列值进行拼接得到唯一表示此次收款交易的信息。使用Google.zxing将其转化为二维码形式，并在视图中显示出来。

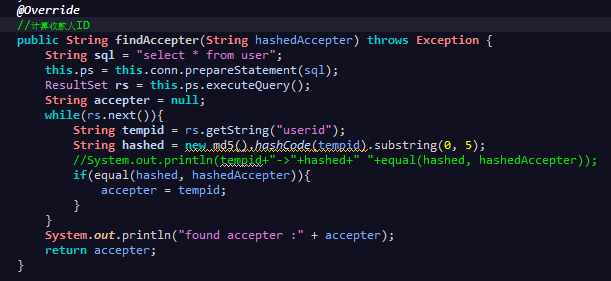
Gather.java

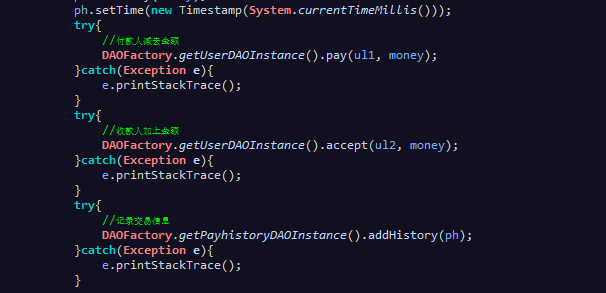


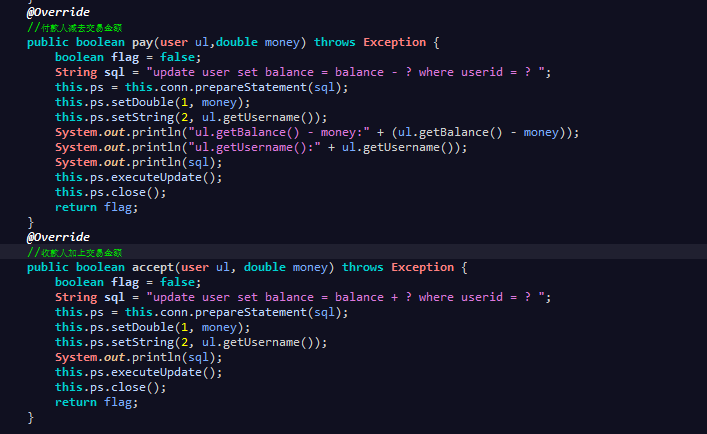
1. 付款

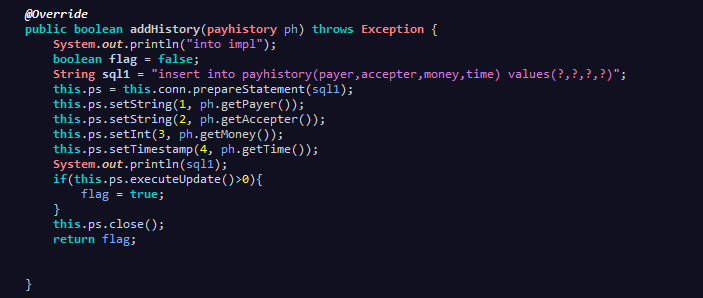
付款时先扫描二维码，将扫描的图片传回服务器后进行读取。获取二维码中字符串，其前5位为用户HASH标识，之后部分为二维码生成的时间表示。先匹配时间，若当前时间与二维码中时间标识不同（相隔十分钟以上），则输出二维码已过期，停止交易；二维码未过期，再去数据库中匹配用户标识，即select所有用户，分别对其用户名进行MD5加密，取前五位（由于样本数量较少，考虑到二维码识别难度，暂取五位），若与二维码中用户标识相同，则找到收款人，进行交易，给收款人加上交易金额，给付款人减去交易金额，并将交易活动记录入数据库。







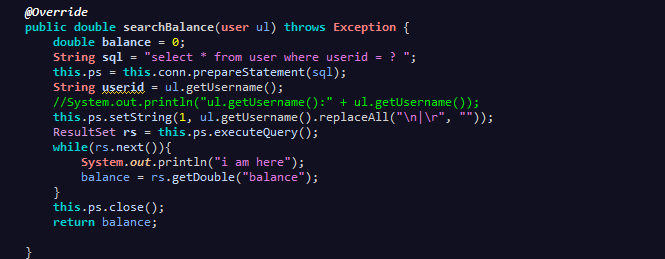




1. 余额查询

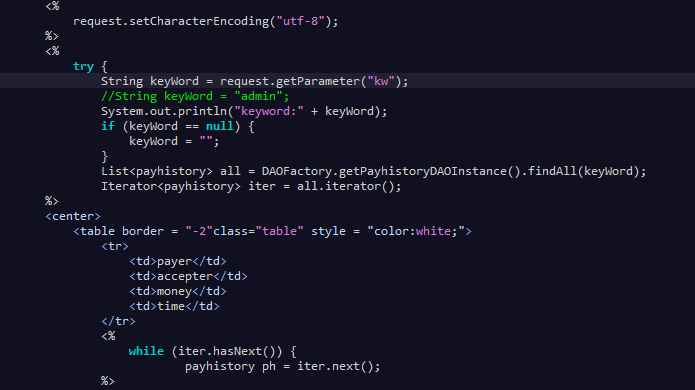
先在DAO中定义活动的接口，然后在Impl中具体实现与表的交互，Proxy提供数据库代理，Factory提供活动入口，查询user表中相关元组即可。

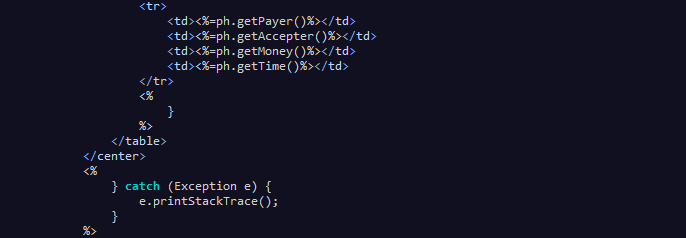
UserDAOImpl.java



1. 查询历史记录

用户登录之后，使用用户名遍历payhistory表即可。使用JSP打印出历史记录的表格（由于ajax的局部刷新的视觉效果不好，所以显示到另一页面，页面之间使用jQuery-params传值）。





1. 测试数据和结果

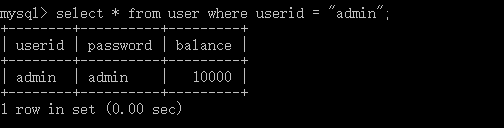
首先，将项目使用Tomcat服务器在本地跑起来，打开WiFi，使用移动设备连入，则设备与服务器处于同一局域网中。

在终端使用ipconfig指令获取WiFi局域网的IP地址，例如172.60.2.1，则在连入设备中输入https://172.60.2.1:8080/payment3即可进入项目界面。

注册一个名为admin用户，默认新注册用户账户金额都为10000元。



查看数据库，确实存在此用户



使用admin账号登录



收款，生成收款二维码



注册另一账户，root，打开root账号的付款界面。（选择文件就是调用摄像头）



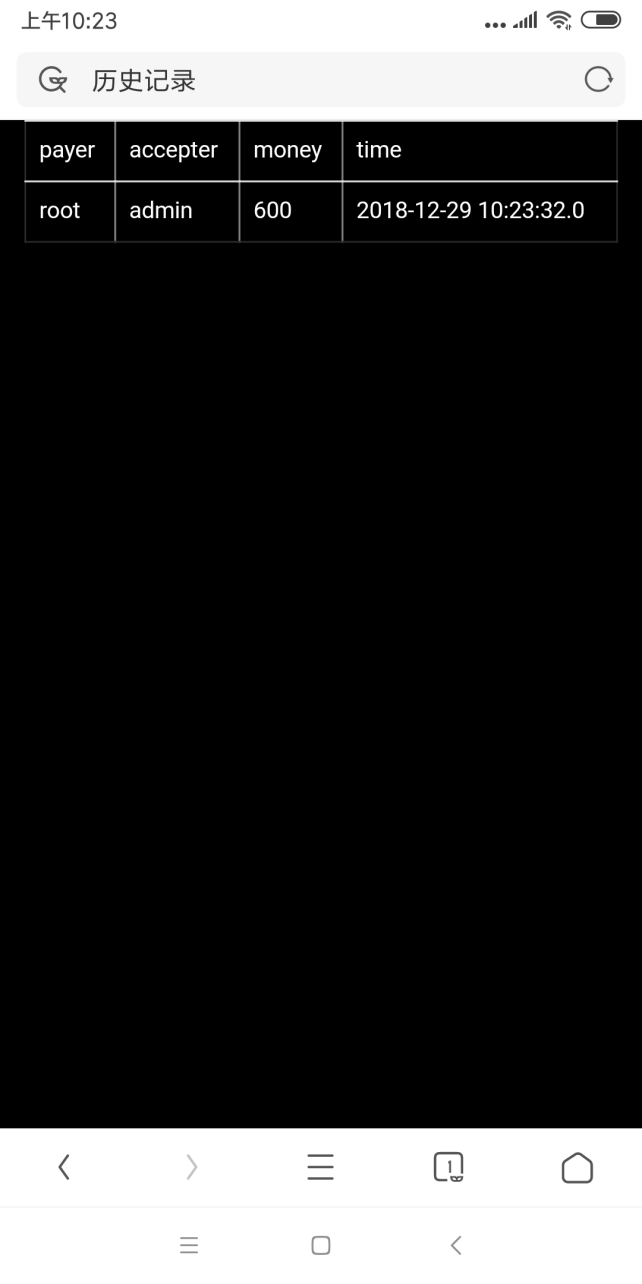
拍摄二维码



输入金额，确定支付



查看历史记录



查看余额



超过十分钟后再付款，则提示二维码失效



1. 心得体会

这个项目整体做了两个星期左右，主攻了大约一个星期，我负责后端，和二维码生成与读取以及服务器的SSL加密配置。

项目开始时需要选择平台，原本打算使用Android写一个APP，不过由于笔记本电脑无法运行Android studio，又考虑到自己以前还是做过一些javaweb项目的，所以选择使用webAPP的方式呈现。

使用bootstrap框架可以很好地支持移动端的响应式布局，使用ajax进行前后端之间的传值，用DAO设计模式以避开后端代码之间的耦合度太高问题，使用servlet处理请求，轻车熟路，这一部分做地很爽。

二维码生成与破解是调用别人的接口来实现的。开始时选择的是一个日本公司的接口，只有两个jar包，而且方法也不算复杂，甚得我心，但是实现效果是在不够理想，只要光线稍微变化一些就识别不出来，屏幕上有一些暗影也不行。后来改成了Google的zxing，虽然效果也不算太好，同一二维码只有从特定的角度拍摄才能识别出来，但是至少不太复杂的二维码是没有问题的（后来在stackflow里看到有人使用hints，可以提高二维码的识别成功率，试用了一下，几乎没什么区别）。

开始认为二维码算是一种加密算法，但是随着项目的推进和原型的成型，意识到二维码只是一种可逆的变化，加密程度太低。后来想到调用哈希算法取用户名的散列值，这样只有服务器端可以遍历数据表，确定二维码对应的收款人，而别人即使得到了二维码也没什么用。由于MD5为128位，SHA1位160位，为了处理二维码方便起见，选择了MD5算法。

MD5算法调用java的security包内的方法稍加修改即可，这部分实现起来还是比较容易的，时间戳可以调用currentTimeMiles方法可以获取自1970年1月1日以来的毫秒数，将其除以十分钟（10\*60\*1000），即可得到当前所在十分钟时间区间的标识（即在十分钟内付款为有效）以此作为判断二维码是否过期的标准。

项目已经结束了，但是它带给我的意义却不止于此。最后时刻看着越来越多的人什么都没做却直接从网上抄下代码甚至直接将整个工程改个名字拿来用，我越发怀疑自己这么长时间以来宵衣旰食，每天通宵和各种奇葩的bug作斗争的意义何在。我写这份报告的之后已经是29号上午了，在这工程和一年的拼搏即将迎来尾声之时，我感到一种释然。曾经听到过一个故事，有一名医生为战场上的伤员治病，但他们病好了之后就再次上战场，再次受伤，医生心力憔悴，深感自己的努力付诸东流，离开了军营，但是后来他又回来了，因为他明白了自己本就是一名医生。而我本就是一名计算机行业工作者，我希望在计算机行业中一展宏图，就必须认真做好自己眼前的每一件事情。选择了行业之后，就必须用一生的时间来磨练自己的技艺，这也是一种获得别人尊重的方法。

1. 其他数据

代码运行环境：

IDE：Eclipse Mars.1 Release (4.5.1)

Jdk: 1.8

数据库：mysql 5.7

前端外部插件：bootstrap，jquery，

后端外部插件：Google.zxing

编辑工具：

eclipse，webstorm，Dreamweaver

编写代码行数：

大约1900

GitHub仓库地址：https://github.com/Liais/QRCode\_Payment