

中山大学数据科学与计算机学院

# 射频识别实验 报告

上位机软件的功能设计与实现

小组成员：

(姓名：           学号：    )

(姓名：           学号：    )

(姓名：           学号：    )

(姓名：           学号：    )

指导教师：

胡建国

# 目 录

第一章 项目介绍 .....	1
第二章 项目设计 .....	1
2.1 基础功能设计 .....	1
2.2 创新功能设计 .....	5
第三章 功能测试 .....	7
3.1 基础功能测试 .....	7
3.2 创新功能测试 .....	9
第四章 项目总结 .....	11

# 第一章 项目介绍

近些年智能卡的日渐普及已经使得智能卡成为我们每个人生活中不可或缺的一部分，在金融（银行卡）、饮食（饭卡）、医疗（医疗卡）、社会保障（社保卡）、公共交通（公交卡）、物业（业主卡）等各个领域里，我们都能看到智能卡扮演的重要角色。而包括卡片在内外，整个智能卡工作系统实际需要三大组件：智能卡、读卡器、后台系统。

而我们此次实现的项目就是后台系统的角色，是一款面向智能卡管理员的 Windows 程序，界面清爽接口清晰，提供了诸多功能：录入卡片持卡人信息，初始化电子钱包，电子钱包的充值、消费，充值消费的文件记录，对卡片挂失禁用其它功能，补卡操作转移卡片信息，对密码进行哈希以提高安全性，以及对卡片进行读块写块的调试功能等。

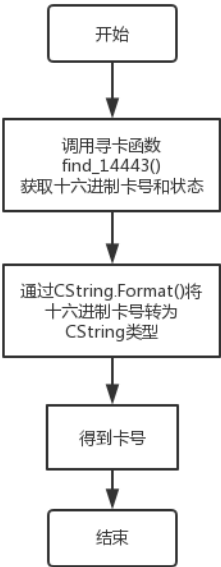
# 第二章 项目设计

## 2.1 基础功能设计

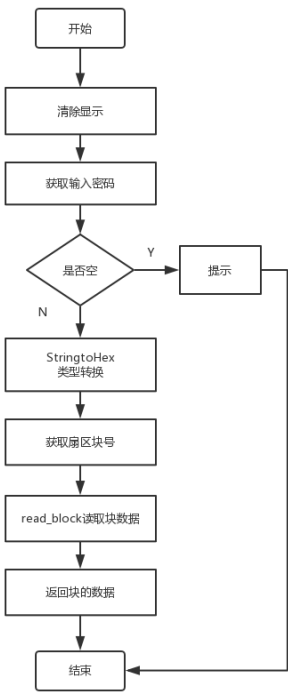
### 1. 寻卡读块读扇区写块

寻卡主要通过调用 find\_14443()函数即可获得卡号，再对该十六进制的 unsigned char 卡号转换为 CString 类型便是理想的结果；

读块模块主要调用 read\_block 函数来读取数据，但由于其输入参数要求十六进制数，故而需要一个 StringToHex 模块；

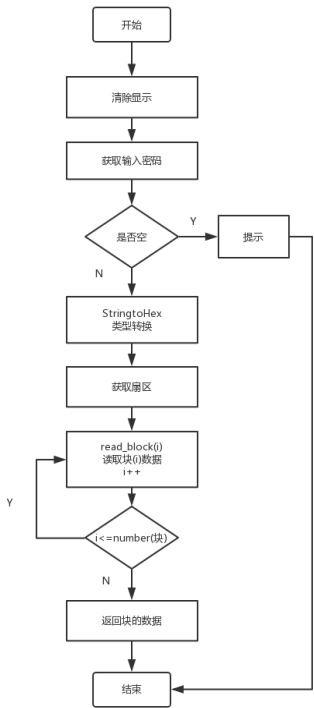


寻卡流程图

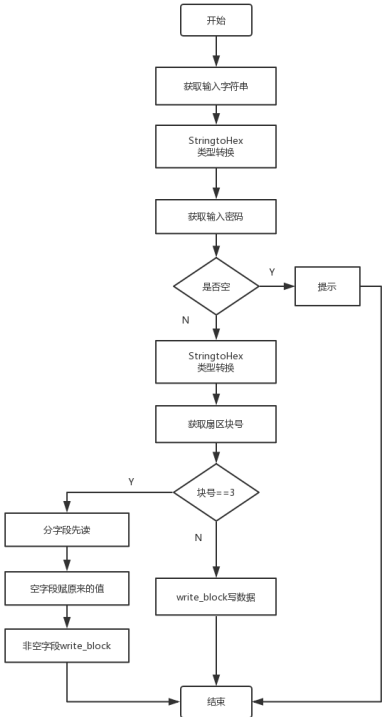


读块流程图

读扇区实则是确定一个扇区后遍历其所有的块，进行读数据，故而与读块模块极其类似；写块也与读块模块类似，只是调用的是 `write_block` 函数，同样由于输入参数的要求，`StringtoHex` 模块能起到作用，特别地，当写块 3 数据时，由于分段，可能某一段输入而其他段没有进行输入，故而要先读出旧数据，将没有输入的段赋予原先的旧数据；



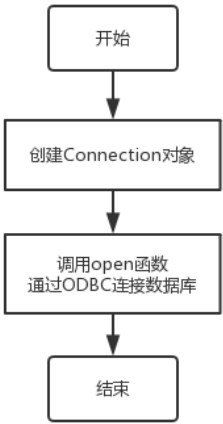
读扇区流程图



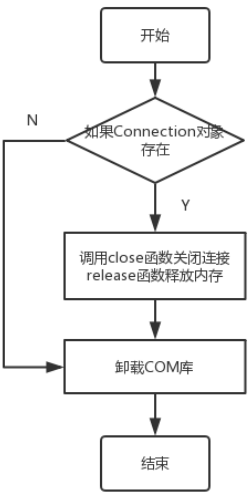
写块流程图

## 2. 数据库连接/关闭

数据库的连接、关闭主要通过 `Connection` 连接对象来实现，连接对应着 `Connection` 对象的 `open` 函数，关闭则对应着 `Connection` 对象的 `close` 函数：



连接流程图



断开流程图

### 3. 数据库更新/查询

数据库的更新操作包括增加、修改、删除，都是通过 **Command** 命令对象调用相关的 **sql** 语句来实现的，当然也需要在执行前与 **Connection** 对象进行绑定，再调用 **Excute** 函数执行；数据库的查询与 **Recordset** 记录集对象息息相关，**Recordset** 对象通过调用 **open** 函数执行相关 **select** 的 **sql** 命令，将返回的结果保存在 **Recordset** 记录集对象中，需要通过类型转换获得最终想要的数据库中的数据：



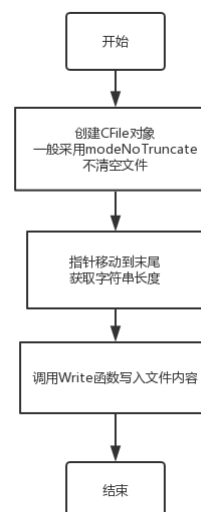
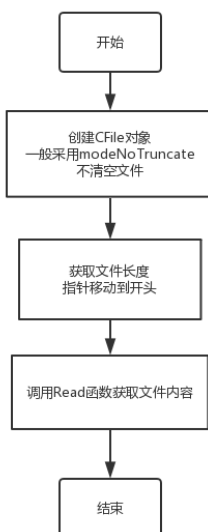
数据库更新操作



数据库查询操作

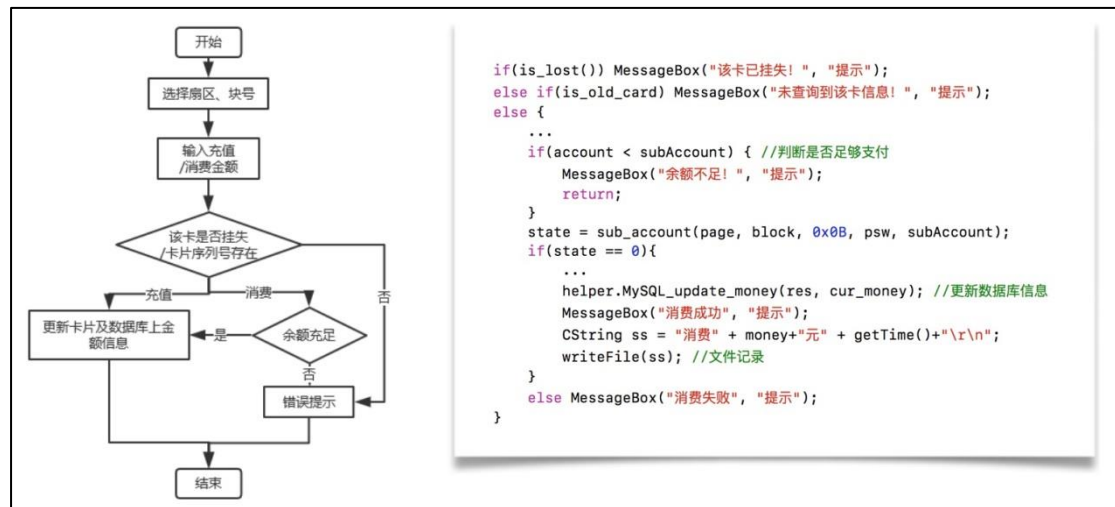
### 4. 文件读写操作

文件的读写通过 **CFile** 对象来实现，实现较为简单，均是先创建 **CFile** 对象，然后根据需求将指针移动到开头/末尾，调用相应的 **Read** 或 **Write** 函数即可实现读写文件功能：



## 5. 充值消费

流程图与消费伪代码：（充值类似）



## 6. 创建/删除卡片

创建卡片流程图伪代码：



删除卡片流程图伪代码：



## 2.2 创新功能设计

### 1. 界面美化:

根据功能需求,我们设计了五项选项卡,分别为读写卡片,创建卡片,交易记录,查询信息,转移信息。功能分类后方便使用,运用我们实验课学习的知识让上位机的界面简洁,大方。

在此基础上,我们运用了 SkinMagic 插件对上位机的界面进行了整体美化,MFC 本身的设计呆板单调,在使用了 SkinMagic 插件后对各个部件的颜色,形状等都进行了优化,让感官体验得到了巨大提升。

下面来介绍一下具体的试验方法,首先下载 SkinMagic 压缩包,其中文件包括 lib 部分和 skin 部分,lib 中包括 SkinMagic.dll, SkinMagic.lib, SkinMagicLib.h, SkinMagicLibMD6.lib, SkinMagicLibMT6.lib, 其中, SkinMagicLib.h 是必须的头文件,添加到工程中。SkinMagic.lib 和 SkinMagic.dll 动态链接库配套使用的, SkinMagicLibMD6.lib 和 SkinMagicLibMT6.lib 静态链接库,分别用于 VC6 的 Release 和 Debug 版编译。Skin 中包括可以加载的皮肤资源,把这些文件全部放在项目资源目录下。

接下来,在 stdafx.h 中添加代码

```
#include "SkinMagicLib.h"
#pragma comment(lib,"SkinMagic.lib")
#pragma comment(lib,"SkinMagicLibMT6.lib")
#pragma comment(linker, "/FORCE:MULTIPLE")
#pragma comment(linker, "/OPT:NOREF")
```

在主程序.cpp 文件的 InitInstance()函数开始处添加初始化 SkinMagic 资源的代码。

```
VERIFY(1 == InitSkinMagicLib(AfxGetInstanceHandle(), NULL, NULL, NULL));
```

这样配置和搭建后就可以轻松简便地调用皮肤库中的皮肤了:

在 InitInstance()函数中加载皮肤文件并设置程序皮肤。

```
VERIFY(1 == LoadSkinFile("Devior.smf"));
VERIFY(1 == SetWindowSkin(m_pMainWnd->m_hWnd, "MainFrame"));
VERIFY(1 == SetDialogSkin("Dialog"));
```

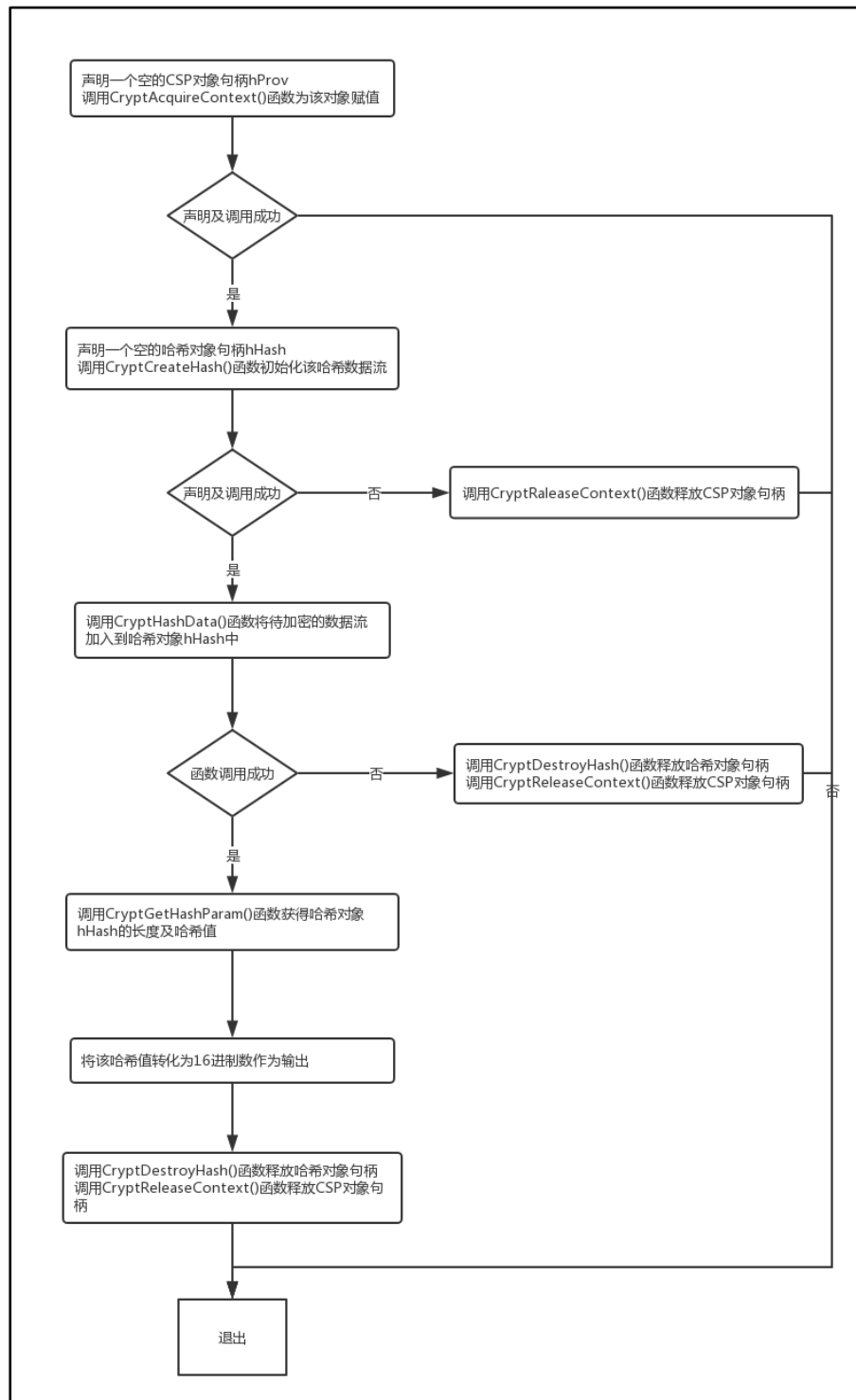
效果如下:



## 2. 对密码进行哈希加密

学生密码并不直接存放在服务器端的数据库中，因为这样可能产生安全和隐私问题，而是对密码经过 MD5 加密后将哈希值存储在数据库内。

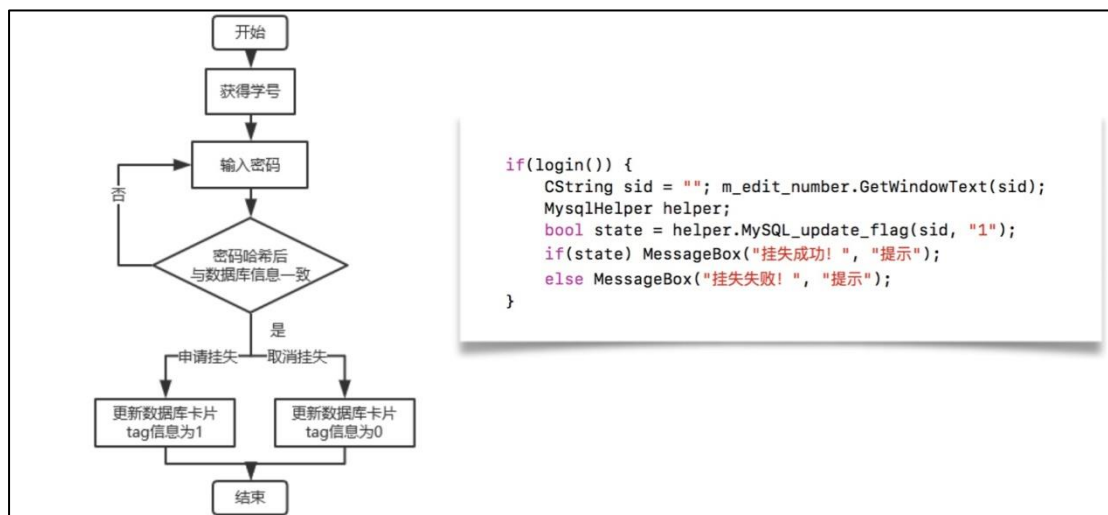
MD5 加密过程实现流程图如下：



## 3. 挂失的申请/取消

伪代码与流程图如下：





#### 4. 补卡操作

伪代码与流程图如下：



## 第三章 功能测试

### 3.1 基础功能测试

- 开场寻卡：



- 读块：



- 读扇区：



- 写块



对比上下两图，将块 1 信息修改后点击写块，再次读扇区得到结果：



可见写块成功，块 1 信息发生了改变

- 录入/查询持卡人信息

放置卡片于读卡器上，输入持卡人信息：



通过后台查看数据库可见：数据库更新操作成功：

14352110	003F5D60	12306	斯嘉丽	女	16	法学院	50.6	0
15252256	AA4F12FF	123456	星盛	男	16	数据科学与计算机学院	2000	0
15352222	3058d70e	80117C58A42F70625894696B4896FD1D	林	男	15	移动	250	1

再于另一接口通过学号密码查看持卡人信息可见：数据库查询操作成功：

The interface shows a '卡片登录' (Card Login) section with fields for '学号' (Student ID: 15352222), '密码' (Password: hh13), and '挂失' (Lost: 0). There are buttons for '申请挂失' (Apply for loss) and '解除挂失' (Cancel loss). A red box highlights the '信息一览' (Information Overview) section, which displays card details: '序列号' (Serial Number: 3058d70e), '学院' (College: 移动), '姓名' (Name: 林), '年级' (Grade: 15), '性别' (Gender: 男), and '余额' (Balance: 250). A '查询卡片信息' (Query Card Information) button is also present.

- 充值消费：

电子钱包初始化 250 元并充值 300 元：

The interface shows the '电子钱包' (Electronic Wallet) section. The '余额' (Balance) is 250. A red box highlights the '余额' field. The '钱包初始化' (Wallet Initialization) button is visible. The '充值' (Recharge) button is also visible, with a '金额' (Amount) of 300. The '历史交易记录' (Transaction History) section is empty.

再消费 100 元并查看电子钱包记录：

The interface shows the '电子钱包' (Electronic Wallet) section. The '余额' (Balance) is 450. A red box highlights the '余额' field. The '钱包初始化' (Wallet Initialization) button is visible. The '扣费' (Deduction) button is visible, with a '金额' (Amount) of 100. The '历史交易记录' (Transaction History) section shows two transactions: '初始化250元 2018年5月17日 22:22' and '充值300元 2018年5月17日 22:22'.

可见充值消费、文件读写、数据库同步余额测试成功。

## 3.2 创新功能测试

- 挂失功能

学号 15352222 的同学申请挂失：



那么当其对应的卡片放置于读卡器上进行充值消费时，其功能均被禁用：



#### • 补卡功能



点击转移信息后，将该学号对应的持卡人的信息转移到新卡上去（由于这里实验中只有一张卡，故我们采用随机分到另一个卡号上）：

stunum	cardnum	pwd	stuname	stusex	grade	college	money	flag
14352110	003F5D60	12306	斯嘉丽	女	16	法学院	50.6	0
15252256	AA4F12FF	123456	星露	男	16	数据科学与计算机学院	2000	0
15352222	3058d70f	80117C58A42F70625894696B4896FD1D	林	男	15	移动	250	1

#### • 哈希加密

对于哈希算法，我们的测试模块测试其以下几点特征：

- ①相同密码字符串哈希值相同；
- ②不同密码字符串哈希值不同；
- ③长度任意的密码字符串哈希后均 32 位哈希值：

由 1、2 字符串知①特征满足；1、3、4 字符串知②特征满足；1、4、5 字符串知③满足。

时间精力，而且和我们的课程 RFID 没有太大关系。在寻找了大量资料后，我决定采用 SkinMagic 库来进行美化。在配置这个库的时候也遇到了很多问题，因为网上的使用说明大多针对 VS 系列，VC++6.0 的资料不太好找，但在进行了大量尝试后终于成功引入了皮肤。

也感谢其他组员完成的工作和对我的信任，将直接决定 TA 直观印象的界面设计交给我，我也很努力地让我们的作品和其他组有感官上的不同。

耿悦：

主要负责的模块：学生密码通过 MD5 加密后代替原先的明文密码保存在数据库中。

具体的实现步骤：C 原生的库中就有专门负责加密的库 “Wincrypt.h”，并且这个库的功能比较强大，定义了大量的加解密算法的 API，可以十分方便地实现不同的加解密功能。主要是通过学习网上的资料，了解不同接口的使用方法，最后整合在一起实现最终的功能。按照步骤来，具体的实现并不算很复杂。整个代码的编写过程中学习到了很多新的知识，主要了解到了通过调用原生库实现简单的加密过程的全部流程和一些基础的密码学的知识。