

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：张子豪 学号：15352427)

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
2.1.1 读写块.....	1
2.1.2 文件管理.....	1
2.1.3 数据库连接.....	2
2.2 创新功能设计 .....	3
2.2.1 数据库在程序中的应用 .....	3
2.2.2 定时器.....	4
2.2.3 界面美化.....	4
2.2.4 Web Browser .....	4
第三章 功能测试 .....	5
3.1 基础功能测试 .....	5
3.1.1 读写块.....	5
3.1.2 文件管理.....	5
3.1.3 数据库连接.....	6
3.2 创新功能测试 .....	6
3.2.1 登录、注册界面.....	6
3.2.2 加载用户全部交易记录.....	6
3.2.3 Web Browser .....	7
第四章 项目总结 .....	8

# 第一章 项目介绍

RFID 技术可通过无线电信号识别特定目标并读写相关数据，当下应用广泛。本项目是基于 RFID 读写器的上位机软件，可通过计算机发出读写卡片等一系列操控命令，并与远程数据库连接来记录信息。此外，本项目借助了 IE 的内核，在项目中嵌入浏览器来方便使用。

## 第二章 项目设计

### 2.1 基础功能设计

#### 2.1.1 读写块

首先判断密钥类型，并将用户输入的密钥转为 char 数组，为空则给出相应提示。然后根据选中的扇区和块号调用函数读写块即可。过程中较为复杂的只有 CString 类型转 char 数组，这个过程封装为函数方便调用。

```
void CDebugger::Transform_CString_to_UnsignedChar(CString str,unsigned char* res){
    unsigned char buffer[50];
    for(int i=0;i<str.GetLength();i++){
        char ch=str[i];
        if(ch>0x40) buffer[i]=(unsigned char)ch-0x37;
        else buffer[i]=(unsigned char)atoi(&ch);
    }
    int j=0;
    int leng=str.GetLength();
    if(leng%2==1) {
        res[j++]='0';
        leng++;
    }
    for(i=0;j<leng/2;){
        res[j]=(unsigned char)(buffer[i++]<<4);
        res[j++] |= buffer[i++];
    }
}
```

#### 2.1.2 文件管理

##### 1. 文件写入

使用 CFile 的 Open 函数打开文件，然后将指针移到文件末尾，写入内容即可。

```
CFile file;
file.Open(FileName,OpenFlags);
file.SeekToEnd();
file.Write(content,strlen(content));
```

## 2. 文件读取

首先使用 **CFileFind** 寻找当前路径下是否有指定文件存在，若存在则使用 **CStdioFile** 打开文件，然后使用 **ReadString** 函数循环读取文件中的内容直至读取完毕。最后将读取到的内容显示在编辑框中。

```
CFileFind finder;
if( finder.FindFile(FileName) ){
    CStdioFile file; file.Open(FileName,OpenFlags,&CFileException);
    while( file.ReadString(tmp) ) res+=tmp;
    file.Close();
}
```

## 3. 文件删除

同样先使用 **CFileFind** 寻找当前路径下是否有指定文件存在，若存在则使用 **CFile** 打开文件，然后使用 **Remove** 函数删除文件即可。

### 2.1.3 数据库连接

ADO 与 ODBC 一起工作来访问数据源，使用方便，但缺点是只能访问本地的数据源，一旦程序被移植到其他设备上将无法正常工作。考虑到程序的可移植性，本次实验使用 **mysql-connector-c** 来连接位于公网的远程数据库。

## 1. 引入 mysql-connector-c

从官网下载 32 位 **mysql-connector-c** 的压缩包，解压后将 **.lib\libmysql.lib** 与 **.lib\libmysql.dll** 两个库文件引入工程，然后在代码中包含 **.include\mysql.h** 头文件，即可调用相关函数。

```
#pragma comment(lib,".libmysql.lib") //引入 lib

#include ".include\mysql.h" //包含头文件
```

## 2. 连接到数据库

初始化 **mysql** 库，设置编码，然后调用函数连接服务器。

```
bool ConnectDB(){
    mysql_library_init(0,NULL,NULL); //初始化 mysql 库
    if((m_pSQL=mysql_init(NULL))==NULL) return false;
    mysql_options(m_pSQL,MYSQL_SET_CHARSET_NAME,"gb2312"); //编码设置
    if(mysql_real_connect(m_pSQL,"a.aniviakid.cn","user1","user1","RFID",3306,NULL,0)==NULL){
        AfxMessageBox(mysql_error(m_pSQL)); //报错提示
        return false;
    }
    return true;
}
```

## 3. 执行 sql 语句

使用 `mysql_query` 函数执行语句，然后将执行结果(如 `select` 结果)存储在 `MYSQL_RES` 中。

```
Bool ExcuteQuery(char* strSQL){
    If( mysql_query(m_pSQL,strSQL) ){
        AfxMessageBox(mysql_error(m_pSQL));
        return false;
    }
    m_pSQLResultSet=mysql_store_result(m_pSQL); //存储 sql 语句执行结果
    return true;
}
```

## 2.2 创新功能设计

### 2.2.1 数据库在程序中的应用

#### 1. 登录、注册功能

进入主程序前先将所有界面隐藏，然后弹出登录对话框

```
m_MainMenu.ShowWindow(SW_HIDE);
CLogin login;
login.DoModal();
```

在登录对话框中点击登录按钮后捕获用户输入的用户名与密码，然后根据用户名在数据库中取出对应密码进行比对，比对成功则将隐藏的界面重新显示，并关闭登录对话框。

```
CAdoMySQLHelper sql;
sql.ConnectDB();
bool res=sql. ExecuteQuery(
    "select password from user where username='"+username+"'");
if(res) {
    MYSQL_ROW pSQLRow=NULL;
    CString select_res;
    while(pSQLRow=mysql_fetch_row(sql.Get_m_pSQLResultSet())) select_res+=pSQLRow[0];
    //比对 select_res 与用户输入的密码 下略
}
```

新用户使用程序可点击注册，然后弹出注册对话框，用户输入账户信息后使用 `insert` 语句将数据插入数据库。

```
CREGISTER registerd;
registerd.DoModal();

CAdoMySQLHelper sql;
sql.ConnectDB();
bool res=sql. ExecuteQuery(
    "insert into user(username,password) values('"+username+"','"+password+"');"
);
if(res==true){
    MessageBox("注册成功");
}
```

```

        CDialog::OnCancel();
    }
    else MessageBox(mysql_error(sql.Get_m_pSQL()));

```

## 2. 查询登入用户所有交易记录

数据库中另设一张表来存储交易记录，以 user 表中的 username 作为外键，存储交易项和交易金额。连接到数据库后根据当前登入用户的用户名来查询所有交易记录即可。代码基本与上一部分相同，连接到数据库执行 select 语句，然后从 MYSQL\_RES 中取出查询结果，不再赘述。

### 2.2.2 定时器

程序记录并实时显示用户登入后的使用时间。首先创建一个定时器，设置响应时间为 1000 毫秒，即每秒执行一次响应函数，然后在响应函数中更新时间并显示到对话框中即可。

```

class CWeb : public CDialog { //.h 头文件中增加成员变量
public:
    UINT_PTR m_ActiveTimer;
}

m_ActiveTimer = SetTimer(0, 1000, NULL); //开启定时器

void CWeb::OnTimer(UINT nIDEvent) { //响应函数
    if(nIDEvent==0) { /*更新时间并显示，逻辑较简单，略*/ }
}

```

### 2.2.3 界面美化

#### 1. 设置控件颜色

在每个控件开始绘制内容在窗口的时候，都会向父窗口发送 WM\_CTLCOLOR 消息，因此只需要重载相应的父窗口消息响应函数 OnCtlColor()即可，程序会运行完响应函数的代码再开始绘制内容。因此在响应函数中判断并改变刷子颜色即可。

#### 2. 增加背景图片

在项目中插入 Bitmap 资源，在父窗口绘制界面时载入 bitmap 作为背景图，重载父窗口的 OnPaint 函数即可。

### 2.2.4 Web Browser

MFC 中提供了许多 ActiveX 插件，而其中就包括了 Microsoft Web Browser 来满足在 MFC 中嵌入浏览器的需求。Microsoft Web Browser 基于 IE 浏览器内核，微软提供了接口可以方便的在 MFC 中调用。

在对话框中插入 Microsoft Web Browser 控件，然后使用 CWebBrowser2 的库函数 Navigate 导航到指定服务器地址即可。

```

m_web_browser.Navigate(strURL,NULL,NULL,NULL,NULL);

```

导航完成后在当前页面打开新的页面，程序默认会使用系统的浏览器而不是 MFC 当前的对话框打开链接，因此需要修改消息响应。在 `StatusTextChange` 函数中，当属性值发生改变时，存下新的值。然后重载 `NewWindow2` 函数，在打开新窗口时使用 `Navigate` 导航到上一步中存下的 url，然后取消新窗口的加载。

```
void CWeb::OnStatusTextChangeExplorer(LPCTSTR Text) {
    m_strURL=Text;
}

void CWeb::OnNewWindow2Explorer(LPDISPATCH FAR* ppDisp, BOOL FAR* Cancel) {
    m_web_browser.Navigate(m_strURL,NULL,NULL,NULL,NULL);
    *Cancel=TRUE;
}
```

Microsoft Web Browser 一个缺陷是不支持 HTML5 和 CSS3，这是由于 Microsoft Web Browser 的本质是 IE 浏览器的内核，旧版本 IE 不支持 HTML5 与 CSS3，而微软新开发的 Microsoft Edge 浏览器也没有对 MFC 中提供相应接口。

## 第三章 功能测试

### 3.1 基础功能测试

#### 3.1.1 读写块

该部分内容已在课上验收测试，因此不在报告中占篇幅。

#### 3.1.2 文件管理

模拟一次交易，初始化钱包金额为 100，在工程路径下可以看到写入的文件。

📄 history.txt      2018/5/15 21:39      文本文档      1 KB

打开如下



点击“查询本地记录”按钮，在程序中读取文件，效果如下



### 3.1.3 数据库连接

此部分不做单独测试，在后续数据库应用中一并测试。

## 3.2 创新功能测试

### 3.2.1 登录、注册界面

进入程序后首先弹出登录界面，未注册用户可点击注册按钮进入用户注册。



图 登录界面



图 注册界面

### 3.2.2 加载用户全部交易记录

承接 **3.1.2 文件管理** 内容，点击“查询历史记录”按钮，可以从数据库中加载出当前用户的全部交易记录





### 3.2.3 Web Browser

进入 Web 选项卡，点击开始按钮，Web Browser 开始导航，同时定时器开始记录用户上机时间



图 首页如上

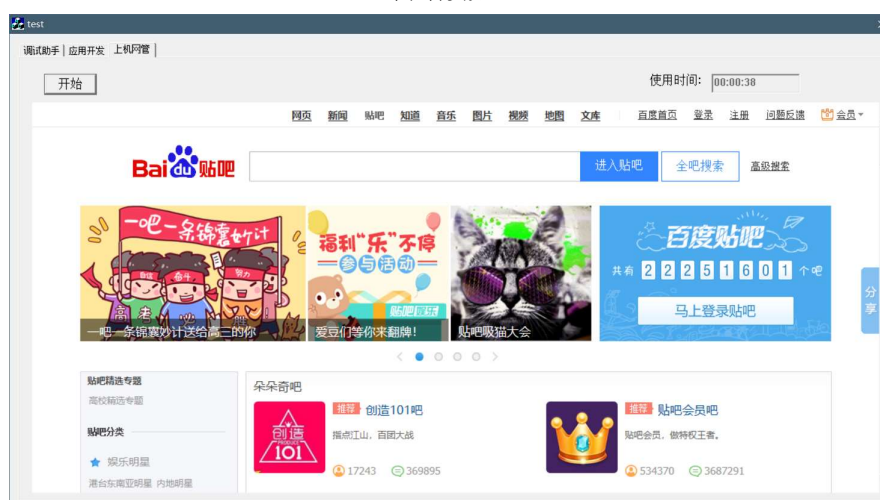
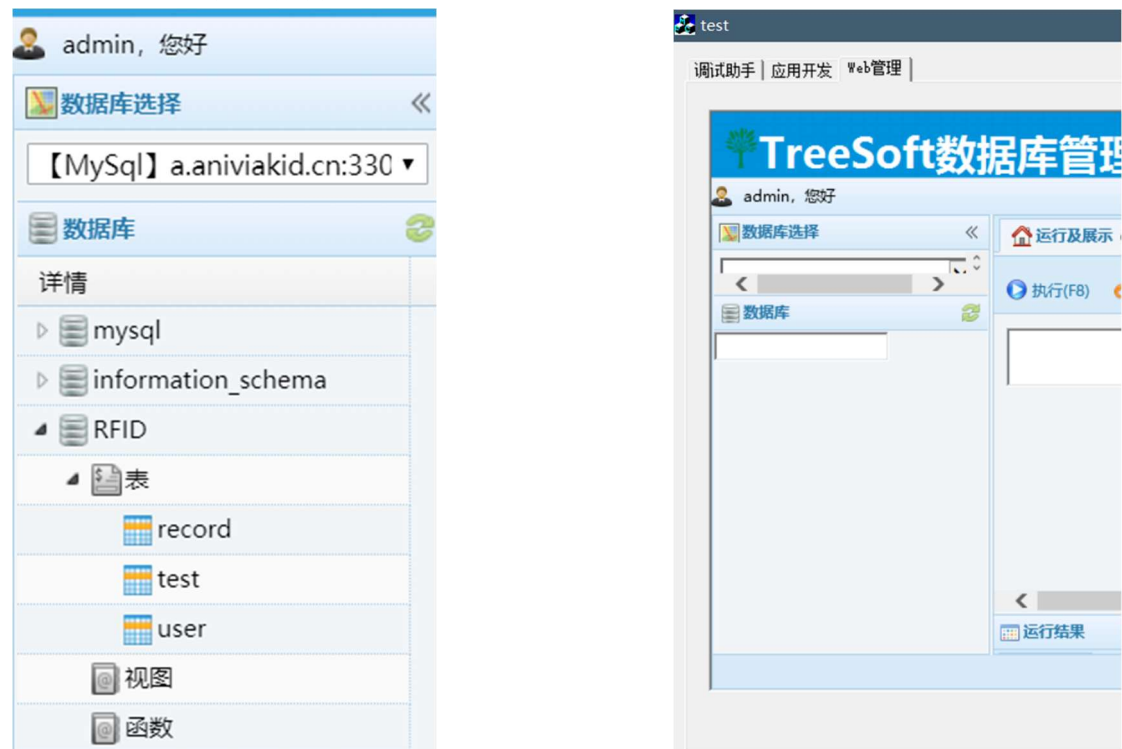


图 跳转至其他页面如上

## 第四章 项目总结

通过这次项目，我对 MFC 有了一个大致的认知。了解了 MFC 中的类向导，并自学了如何使用 mysql-connector-c 来连接数据库。在实现登录、注册页面的过程中，对 MFC 中父对话框与子对话框之间的关系也有了更清晰的认识。

在实现 Web Browser 的过程中，原本是打算在 MFC 中嵌入浏览器通过校园网访问搭载在本地的 Tomcat 服务器（通过路由器在 WAN 口设置端口转发），然后通过网页管理数据库。最后实现的时候出现了一点小波折，MFC 中无法显示出连接的数据库，下左图为 chrome 浏览器中加载网页管理数据库，下右图为 MFC 中加载网页。



由于 Windows10 上各种浏览器都很完善，因此测试的时候都很正常，最后直接调用 IE 内核才重现了这个 bug。翻阅网页源码后发现这部分是用 CSS3 来实现的，而 IE 内核并不支持 HTML5 和 CSS3，因此这个想法只能作罢。