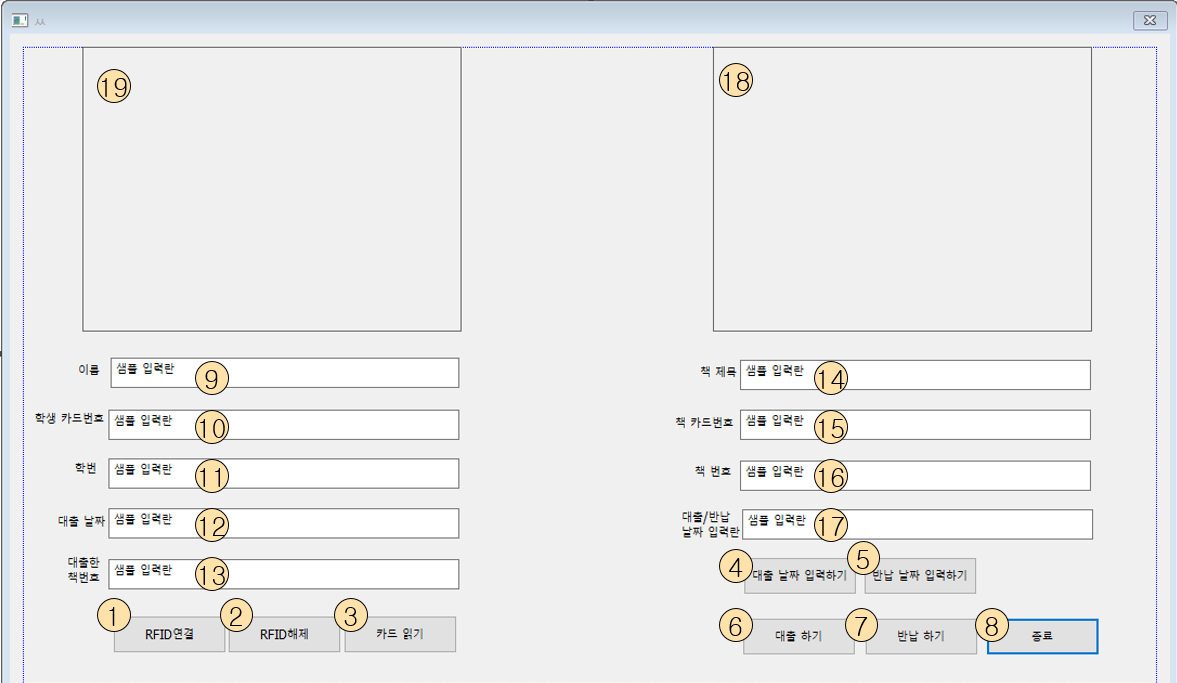
RFID 프로젝트

김혁구



1. RFID연결 – 카드를 읽기 위한 장치를 컴퓨터와 연결한다.

2. RFID해제 – 장치 연결을 해제한다.

3. 카드 읽기 – 붙어있는 카드를 읽는다

4. 대출 날짜 입력하기 – 17번에 날짜를 입력한 뒤 날짜 입력하기 버튼을 통해서 날짜를 입력 받는다. 대출하기 버튼을 누르기 전에 사용한다.

5. 반납 날짜 입력하기 - 17번에 날짜를 입력한 뒤 날짜 입력하기 버튼을 통해서 날짜를 입력 받는다. 반납하기 버튼을 누르기 전에 사용한다.

6. 대출하기 - 4번에서 입력한 날짜와 대출했다는 데이터를 데이터베이스 속에 집어넣는다.

7. 반납하기 – 데이터 베이스 속에서 대출했다는 데이터를 삭제하고, 5번에서 입력한 날짜를 데이터베이스에 집어넣는다.

8. 종료 – 조욜

9. 학생 이름 - 3번에서 학생카드(14443A/B)를 읽었을 때, 이름이 출력된다.

10. 학생 카드번호 - 3번에서 학생카드(14443A/B)를 읽었을 때, 학생 카드의 고유번호가 출력된다.

11. 학생 학번 - 3번에서 학생카드(14443A/B)를 읽었을 때, 학생의 학번 출력된다.

12. 책 대출했던 날짜 - 3번에서 학생카드(14443A/B)를 읽었을 때, 이전에 대출했던 책 날짜가 출력된다.

13. 대출한 책 번호 - 3번에서 학생카드(14443A/B)를 읽었을 때, 대출중인 책 번호가 출력된다. 대출한 책이 없으면 빌린 책 없음이라 출력된다.

14. 책 제목 – 책 제목이 출력된다.

15. 책 카드번호 – 책 카드의 고유번호가 출력된다.

16. 책 번호 – 책의 번호가 출력된다.

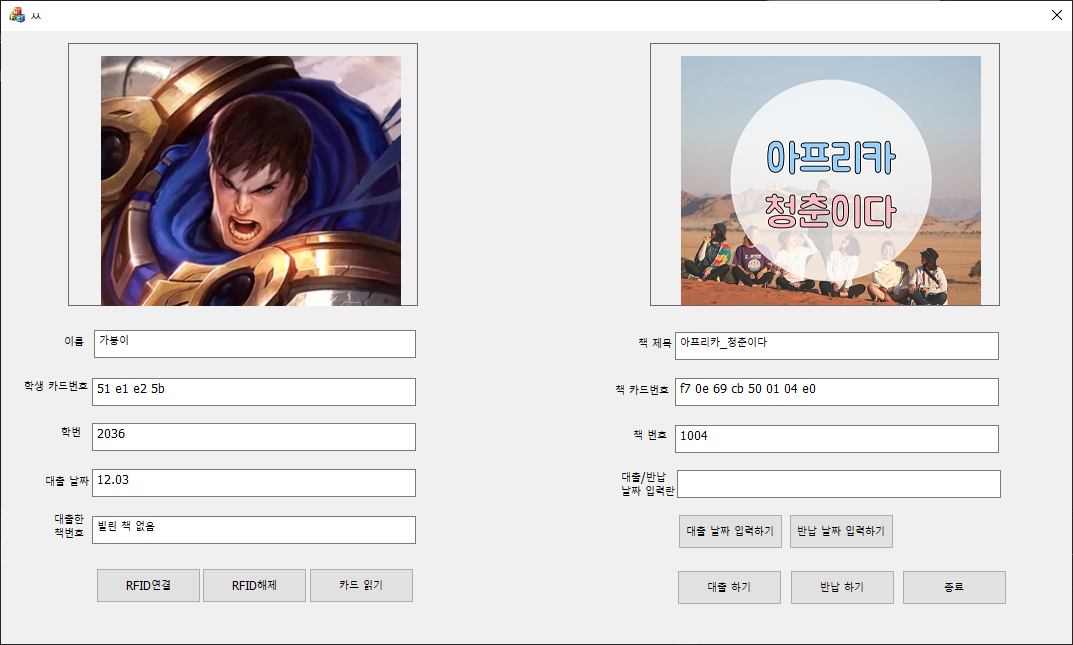
17. 대출/반납 날짜 입력하는 칸 – 대출/반납하기 전 날짜를 입력하는 칸이다.

18. 책 사진이 등장하는 곳 - 책 카드를 찍었을 때 데이터(csv)에서 사진 경로를 불러와서 데이터에 저장되어있는 사진을 불러온다.

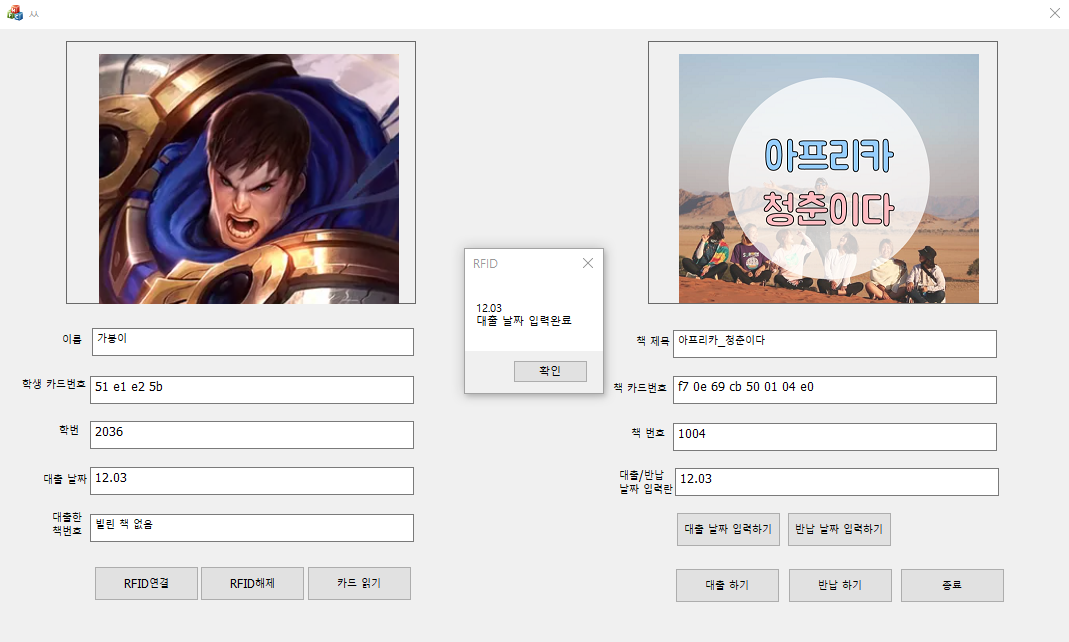
19. 학생 사진 나오는 곳 – 학생 카드를 찍었을 때 데이터(csv)에서 사진 경로를 불러와서 데이터에 저장되어있는 사진을 불러온다.



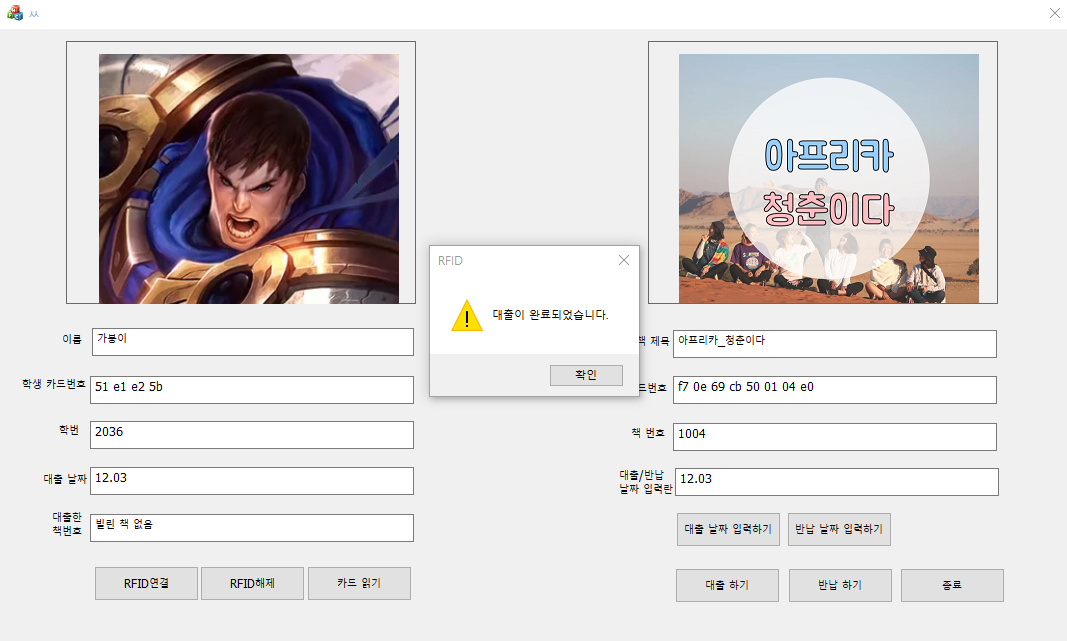
RFID 연결 버튼을 눌렀을 때. 아무것도 출력되지 않는다.



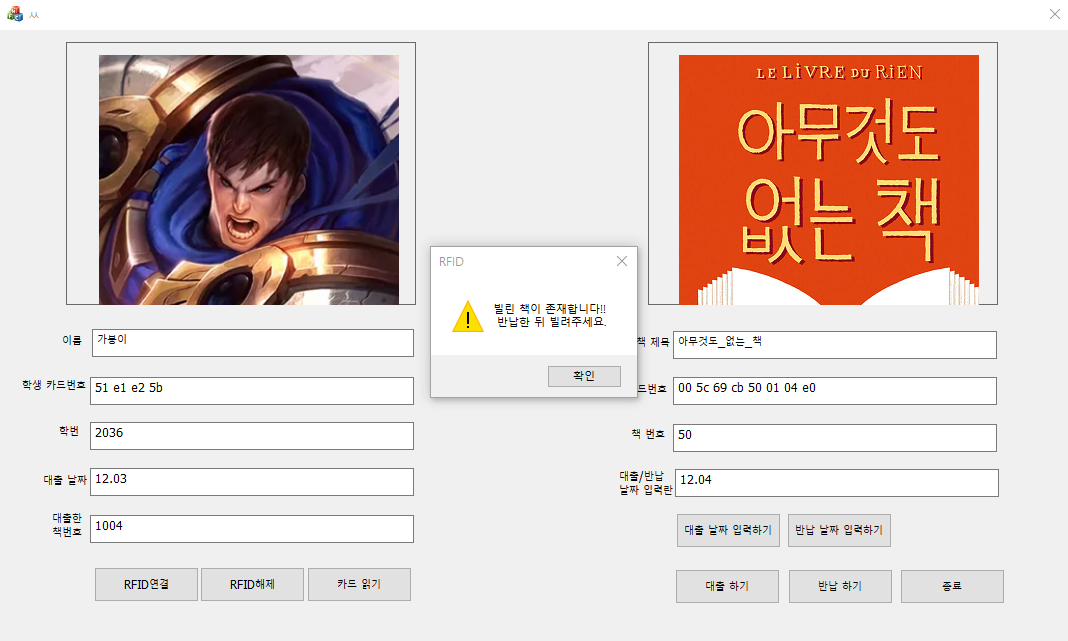
학생 카드와 책 카드를 카드 읽기 버튼을 눌러서 불러온다.



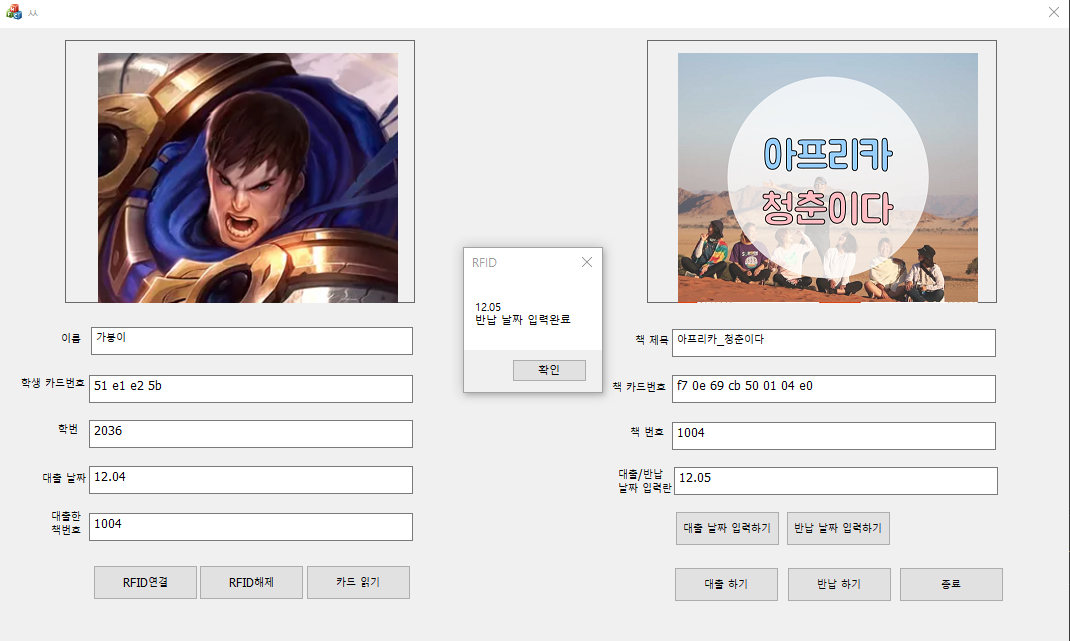
대출 날짜를 입력하고 대출 날짜를 입력하기 버튼을 눌렀을 때



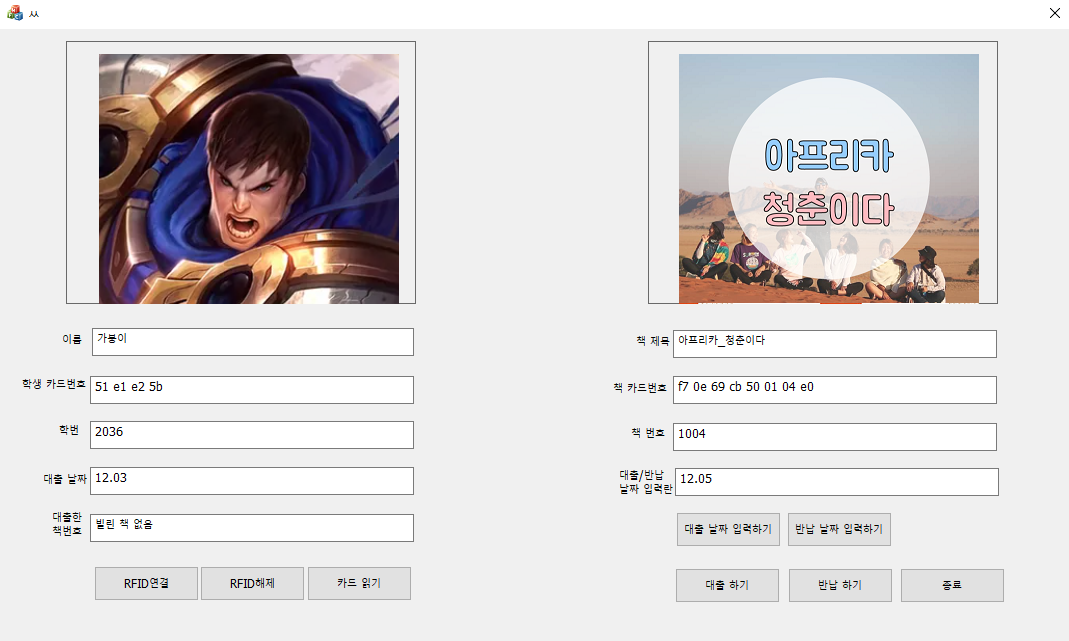
대출하기 버튼을 누르면 이렇게 된다.



만약 대출한 책 번호가 아닌 다른 책 번호를 입력하고 반납하기 버튼을 누른다면 반납할 수 없다고 뜬다.



제대로 책 번호와 반납 날짜를 적으면 반납이 완료된다.



다시 학생카드를 찍어보면 대출한 책이 없다고 나오는 것을 볼 수 있다.

// RFIDDlg.cpp: 구현 파일

//

#include "is\_d2xx.h"

#include "pch.h"

#include "framework.h"

#include "RFID.h"

#include "RFIDDlg.h"

#include "afxdialogex.h"

#include <string.h>

#include <iostream>

#include <fstream>

#include <cstring>

#include <atlstr.h>

using namespace std;

// sound 출력용

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

#include "mmsystem.h"

#pragma comment(lib,"winmm.lib") // 라이브러리 불러오기

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

#define MAX\_DATA 100

string st\_string[10];

CString st\_data\_number;

CString st\_card\_number;

CString st\_name;

CString st\_number;

CString st\_picture;

CString st\_borrow\_flag;

CString st\_borrow\_date;

CString st\_return\_date;

CString st\_book\_number;

int st\_flag;

CString bo\_data\_number;

CString bo\_card\_number;

CString bo\_name;

CString bo\_number;

CString bo\_picture;

string book\_number;

void find\_student(CString \_m\_strRfid);

void find\_book(CString \_m\_strRfid);

// 응용 프로그램 정보에 사용되는 CAboutDlg 대화 상자입니다.

class CAboutDlg : public CDialogEx

{

public:

CAboutDlg();

// 대화 상자 데이터입니다.

#ifdef AFX\_DESIGN\_TIME

enum { IDD = IDD\_ABOUTBOX };

#endif

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 지원입니다.

// 구현입니다.

protected:

DECLARE\_MESSAGE\_MAP()

};

CAboutDlg::CAboutDlg() : CDialogEx(IDD\_ABOUTBOX)

{

}

void CAboutDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

}

BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialogEx)

END\_MESSAGE\_MAP()

// CRFIDDlg 대화 상자

CRFIDDlg::CRFIDDlg(CWnd\* pParent /\*=nullptr\*/)

: CDialogEx(IDD\_RFID\_DIALOG, pParent)

, m\_strRfid(\_T(""))

, m\_strRfid2(\_T(""))

, st\_m\_strRfid(\_T(""))

, bo\_m\_strRfid(\_T(""))

, m\_strDate(\_T(""))

{

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

CRFIDDlg::~CRFIDDlg()

{

OnDisconnect();

}

void CRFIDDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Text(pDX, IDC\_EDIT2, m\_strRfid);

DDX\_Text(pDX, IDC\_EDIT1, st\_name);

DDX\_Text(pDX, IDC\_EDIT3, st\_number);

DDX\_Text(pDX, IDC\_EDIT4, st\_borrow\_date);

DDX\_Text(pDX, IDC\_EDIT5, st\_book\_number);

DDX\_Text(pDX, IDC\_EDIT7, m\_strRfid2);

DDX\_Text(pDX, IDC\_EDIT6, bo\_name);

DDX\_Text(pDX, IDC\_EDIT8, bo\_number);

DDX\_Text(pDX, IDC\_DATE, m\_strDate);

DDV\_MaxChars(pDX, m\_strDate, 20);

}

BEGIN\_MESSAGE\_MAP(CRFIDDlg, CDialogEx)

ON\_WM\_SYSCOMMAND()

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

ON\_BN\_CLICKED(IDC\_BUTTON1, &CRFIDDlg::OnConnect)

ON\_BN\_CLICKED(IDC\_BUTTON3, &CRFIDDlg::OnReadOnce)

ON\_BN\_CLICKED(IDC\_BUTTON2, &CRFIDDlg::OnDisconnect)

ON\_BN\_CLICKED(IDC\_BUTTON4, &CRFIDDlg::OnBnClickedButton4)

ON\_BN\_CLICKED(IDC\_BUTTON5, &CRFIDDlg::OnBnClickedButton5)

ON\_BN\_CLICKED(IDC\_BUTTON6, &CRFIDDlg::OnBnClickedButton6)

ON\_BN\_CLICKED(IDC\_BUTTON7, &CRFIDDlg::OnBnClickedButton7)

ON\_BN\_CLICKED(IDOK, &CRFIDDlg::OnBnClickedOk)

ON\_COMMAND(IDD\_RFID\_DIALOG, &CRFIDDlg::OnIddRfidDialog)

END\_MESSAGE\_MAP()

// CRFIDDlg 메시지 처리기

BOOL CRFIDDlg::OnInitDialog()

{

CDialogEx::OnInitDialog();

// 시스템 메뉴에 "정보..." 메뉴 항목을 추가합니다.

// IDM\_ABOUTBOX는 시스템 명령 범위에 있어야 합니다.

ASSERT((IDM\_ABOUTBOX & 0xFFF0) == IDM\_ABOUTBOX);

ASSERT(IDM\_ABOUTBOX < 0xF000);

CMenu\* pSysMenu = GetSystemMenu(FALSE);

if (pSysMenu != nullptr)

{

BOOL bNameValid;

CString strAboutMenu;

bNameValid = strAboutMenu.LoadString(IDS\_ABOUTBOX);

ASSERT(bNameValid);

if (!strAboutMenu.IsEmpty())

{

pSysMenu->AppendMenu(MF\_SEPARATOR);

pSysMenu->AppendMenu(MF\_STRING, IDM\_ABOUTBOX, strAboutMenu);

}

}

// 이 대화 상자의 아이콘을 설정합니다. 응용 프로그램의 주 창이 대화 상자가 아닐 경우에는

// 프레임워크가 이 작업을 자동으로 수행합니다.

SetIcon(m\_hIcon, TRUE); // 큰 아이콘을 설정합니다.

//SetIcon(m\_hIcon, FALSE); // 작은 아이콘을 설정합니다.

// TODO: 여기에 추가 초기화 작업을 추가합니다.

return TRUE; // 포커스를 컨트롤에 설정하지 않으면 TRUE를 반환합니다.

}

void CRFIDDlg::OnSysCommand(UINT nID, LPARAM lParam)

{

if ((nID & 0xFFF0) == IDM\_ABOUTBOX)

{

CAboutDlg dlgAbout;

dlgAbout.DoModal();

}

else

{

CDialogEx::OnSysCommand(nID, lParam);

}

}

// 대화 상자에 최소화 단추를 추가할 경우 아이콘을 그리려면

// 아래 코드가 필요합니다. 문서/뷰 모델을 사용하는 MFC 애플리케이션의 경우에는

// 프레임워크에서 이 작업을 자동으로 수행합니다.

void CRFIDDlg::OnPaint()

{

if (IsIconic())

{

CPaintDC dc(this); // 그리기를 위한 디바이스 컨텍스트입니다.

SendMessage(WM\_ICONERASEBKGND, reinterpret\_cast<WPARAM>(dc.GetSafeHdc()), 0);

// 클라이언트 사각형에서 아이콘을 가운데에 맞춥니다.

int cxIcon = GetSystemMetrics(SM\_CXICON);

int cyIcon = GetSystemMetrics(SM\_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

// 아이콘을 그립니다.

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

CDialogEx::OnPaint();

}

}

// 사용자가 최소화된 창을 끄는 동안에 커서가 표시되도록 시스템에서

// 이 함수를 호출합니다.

HCURSOR CRFIDDlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

void CRFIDDlg::OnConnect()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

//열린 포트번호 찾기

if (is\_GetDeviceNumber(&usbnumber) == IS\_OK)

{

printf("FTDI USB To Serial 연결된 개수 : %d\n", usbnumber);

if (is\_GetSerialNumber(0, readSerialNumber) == IS\_OK)

{

if (memcmp(changeSerialNumber, readSerialNumber, sizeof(changeSerialNumber) != 0))

{

if (is\_SetSerialNumber(0, changeSerialNumber) == IS\_OK)

{

printf(" USB To Serial Number 를 변경 하였습니다.\n");

printf(" FTDI SerialNumber : %s \n", changeSerialNumber);

}

}

else

printf(" FTDI SerialNumber : %s \n", changeSerialNumber);

}

}

//열린 포트번호로 연결

unsigned long portNumber;

if (is\_GetCOMPort\_NoConnect(0, &portNumber) == IS\_OK)

{

printf("COM Port : %d\n", portNumber);

}

if (is\_OpenSerialNumber(&ftHandle, readSerialNumber, 115200) != IS\_OK)

{

printf("USB To Serial과 통신 연결 실패\n");

//return -1;

}

else {

printf("Serial포트 %d와 통신 연결성공!! \n", portNumber);

}

Sleep(100);

flag\_r = 0;

}

void CRFIDDlg::OnReadOnce()

{

CString temp, temp1 = \_T("");

CString temp2, temp3 = \_T("");

CString st\_temp, st\_temp1 = \_T("");

CString st\_temp2, st\_temp3 = \_T("");

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

//ISO15693모드로 읽기( 싱글모드 읽기 )

if (flag\_r == 0)

{

if (is\_WriteReadCommand(ftHandle, CM1\_ISO15693, CM2\_ISO15693\_ACTIVE + BUZZER\_ON,

writeLength, wirteData, &readLength, readData) == IS\_OK)

{

int i;

printf("ISO 15693 UID : ");

for (i = 0; i < readLength; i++)

{

temp.Format(\_T("%02x "), readData[i]);

st\_temp.Format(\_T("%02x"), readData[i]);

temp1 += temp;

st\_temp1 += st\_temp;

printf("%02x ", readData[i]);

}

m\_strRfid2 = temp1;

bo\_m\_strRfid = st\_temp1;

find\_book(bo\_m\_strRfid);

show\_bo\_picture();

PlaySoundW(\_T("bo\_alarm.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

UpdateData(FALSE);

printf("\n");

}

}

else

{

// read continue mode

}

//ISO14443A모드로 읽기

if (is\_WriteReadCommand(ftHandle, CM1\_ISO14443AB, CM2\_ISO14443A\_ACTIVE + BUZZER\_ON,

writeLength, wirteData, &readLength, readData) == IS\_OK)

{

int i;

printf("ISO 14443AB UID : ");

for (i = 0; i < readLength; i++)

{

temp2.Format(\_T("%02x "), readData[i]);

st\_temp2.Format(\_T("%02x"), readData[i]);

temp3 += temp2;

st\_temp3 += st\_temp2;

printf("%02x ", readData[i]);

}

m\_strRfid = temp3;

st\_m\_strRfid = st\_temp3;

find\_student(st\_m\_strRfid);

show\_st\_picture();

PlaySoundW(\_T("st\_alarm.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

UpdateData(FALSE);

printf("\n");

}

}

void CRFIDDlg::OnDisconnect()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

// 무선파워를 끊어요.

is\_RfOff(ftHandle);

//USB 포트를 Close

if (is\_Close(ftHandle) == IS\_OK)

{

st\_data\_number.Empty();

st\_card\_number.Empty();

st\_name.Empty();

st\_number.Empty();

st\_picture.Empty();

st\_borrow\_flag.Empty();

st\_borrow\_date.Empty();

st\_return\_date.Empty();

st\_book\_number.Empty();

st\_flag = 0;

bo\_data\_number.Empty();

bo\_card\_number.Empty();

bo\_name.Empty();

bo\_number.Empty();

bo\_picture.Empty();

m\_strRfid.Empty();

m\_strRfid2.Empty();

book\_number="";

printf("연결을 닫습니다. \n");

}

}

void CRFIDDlg::OnBnClickedButton4()

{

CString text;

UpdateData(TRUE);

st\_borrow\_date = m\_strDate;

text = st\_borrow\_date + \_T("\n대출 날짜 입력완료");

MessageBox(text);

return;

}

void CRFIDDlg::OnBnClickedButton5()

{

CString text;

UpdateData(TRUE);

st\_return\_date = m\_strDate;

text = st\_return\_date + \_T("\n반납 날짜 입력완료");

MessageBox(text);

return;

}

void CRFIDDlg::OnBnClickedButton6()

{

if (book\_number == "")

{

AfxMessageBox(\_T("책 카드를 찍어주세요."));

}

else if (st\_number == "")

{

AfxMessageBox(\_T("학생 카드를 찍어주세요."));

}

else

{

if (st\_borrow\_flag == \_T("0"))

{

ofstream fiout("c:\\work\\RFID\\RFID\\RFID\\data\\student\_data.csv");

if (!fiout) { // 열기 실패 검사

cout << "student\_data.csv 파일을 열 수 없다\n";

}

for (int i = 0; i < st\_flag; i++)

{

fiout << st\_string[i] << "\n";

}

fiout << string(CT2CA(st\_data\_number)) << "," << string(CT2CA(st\_card\_number)) << "," << string(CT2CA(st\_name))

<< "," << string(CT2CA(st\_number)) << "," << string(CT2CA(st\_picture)) << ",1," << string(CT2CA(st\_borrow\_date))

<< ",-," << book\_number << endl;

for (int i = st\_flag+1; i < 5; i++)

{

fiout << st\_string[i] << "\n";

}

//fiout << name << "," << school\_number << "," << name << ",안녕하세요, 이거 띄어쓰기도 되냐?, ㅏㅁㄴ\n"; // name 쓰기

//fiout << school\_number << "," << dept << "\n"; // sid 쓰기

//fiout << dept << "\n"; // dept 쓰기

fiout.close(); // 파일 닫기

AfxMessageBox(\_T("대출이 완료되었습니다."));

}

else

{

AfxMessageBox(\_T("빌린 책이 존재합니다!!\n 반납한 뒤 빌려주세요."));

}

}

}

void CRFIDDlg::OnBnClickedButton7()

{

if (book\_number == "")

{

AfxMessageBox(\_T("책 카드를 찍어주세요."));

}

else if (st\_number == "")

{

AfxMessageBox(\_T("학생 카드를 찍어주세요."));

}

else

{

if (st\_book\_number == bo\_number)

{

if (st\_borrow\_flag == \_T("1"))

{

ofstream fiout("c:\\work\\RFID\\RFID\\RFID\\data\\student\_data.csv");

if (!fiout) { // 열기 실패 검사

AfxMessageBox(\_T("student\_data.csv 파일을 열 수 없다"));

cout << "student\_data.csv 파일을 열 수 없다\n";

}

for (int i = 0; i < st\_flag; i++)

{

fiout << st\_string[i] << "\n";

}

fiout << string(CT2CA(st\_data\_number)) << "," << string(CT2CA(st\_card\_number)) << "," << string(CT2CA(st\_name))

<< "," << string(CT2CA(st\_number)) << "," << string(CT2CA(st\_picture)) << ",0," << string(CT2CA(st\_borrow\_date))

<< "," << string(CT2CA(st\_return\_date)) << "," << book\_number << endl;

for (int i = st\_flag + 1; i < 5; i++)

{

fiout << st\_string[i] << "\n";

}

fiout.close(); // 파일 닫기

AfxMessageBox(\_T("반납이 완료되었습니다."));

}

else

{

AfxMessageBox(\_T("반납할 책이 없습니다."));

}

}

else

{

AfxMessageBox(\_T("반납할 책을 찍어주세요!"));

}

}

}

void CRFIDDlg::OnBnClickedOk()

{

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

CDialogEx::OnOK();

}

void find\_student(CString \_m\_strRfid)

{

int i;

char temp[100], temp0[200], \* context=NULL;

string temp2;

ifstream fin;

fin.open("c:\\work\\RFID\\RFID\\RFID\\data\\student\_data.csv");

if (!fin) { // 파일 열기 실패 확인

cout << "student\_data.csv 파일을 열 수 없다\n";

}

fin >> temp; // 맨 윗줄 없얘는 용도

st\_string[0] = temp;

for (i = 1; i < 5; i++)

{

fin >> temp0;

st\_string[i] = temp0;

temp2 = strtok\_s(temp0, ",", &context);

st\_data\_number=temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

st\_card\_number = temp2.c\_str(); //wcout << (const wchar\_t\*)st\_name << endl; 잘 나오는지 확인용 코드

temp2 = strtok\_s(NULL, ",", &context);

st\_name = temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

st\_number = temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

st\_picture = temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

st\_borrow\_flag = temp2.c\_str();

if (st\_borrow\_flag == \_T("0"))

{

temp2 = strtok\_s(NULL, ",", &context);

st\_borrow\_date = temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

st\_return\_date = temp2.c\_str();

st\_book\_number.Format(\_T("빌린 책 없음"));

}

else

{

temp2 = strtok\_s(NULL, ",", &context);

st\_borrow\_date = temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

st\_return\_date = temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

st\_book\_number = temp2.c\_str();

}

if (st\_card\_number == \_m\_strRfid)

{

st\_flag = i;

for (int j = i + 1; j < 5; j++)

{

fin >> temp0;

st\_string[j] = temp0;

}

break;

}

}

fin.close();

}

void find\_book(CString \_m\_strRfid)

{

int i;

char temp[100], temp0[200], \* context = NULL;

string temp2;

ifstream bk\_fin;

bk\_fin.open("c:\\work\\RFID\\RFID\\RFID\\data\\book\_data.csv");

if (!bk\_fin) { // 파일 열기 실패 확인

cout << "book\_data파일을 열 수 없다";

}

bk\_fin >> temp; // 맨 윗줄 없얘는 용도

for (i = 1; i < 3; i++)

{

bk\_fin >> temp0;

cout << temp0 << endl;

temp2 = strtok\_s(temp0, ",", &context);

bo\_data\_number = temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

bo\_card\_number = temp2.c\_str(); //wcout << (const wchar\_t\*)st\_name << endl; 잘 나오는지 확인용 코드

temp2 = strtok\_s(NULL, ",", &context);

bo\_name = temp2.c\_str();

temp2 = strtok\_s(NULL, ",", &context);

bo\_number = temp2.c\_str();

book\_number = temp2;

temp2 = strtok\_s(NULL, ",", &context);

bo\_picture = temp2.c\_str();

if (bo\_card\_number == \_m\_strRfid) break;

}

bk\_fin.close();

}

void CRFIDDlg::show\_st\_picture()

{

///////////////////////////////// 이미지 출력 /////////////////////////////////

CRect m\_rect;

CBitmap m\_bitmap;

//CString Path\_img = "img\\null.bmp";

CString Path\_img = st\_picture; //\_T("img\\beer.bmp");

m\_bitmap.DeleteObject();

m\_bitmap.Attach((HBITMAP)LoadImage(NULL, Path\_img, IMAGE\_BITMAP, 0, 0, LR\_LOADFROMFILE));

CDC\* pDC = GetDC();

CDC memDC;

memDC.CreateCompatibleDC(pDC);

memDC.SelectObject(m\_bitmap);

((CStatic\*)GetDlgItem(IDC\_STATIC02))->GetWindowRect(m\_rect);

ScreenToClient(&m\_rect);

CClientDC dc(this);

dc.BitBlt(100, 25, 300, 250, &memDC, 0, 0, SRCCOPY);

ReleaseDC(pDC);

DeleteDC(memDC);

}

void CRFIDDlg::show\_bo\_picture()

{

CRect m\_rect;

CBitmap m\_bitmap;

//CString Path\_img = "img\\null.bmp";

CString Path\_img = bo\_picture; //\_T("img\\beer.bmp");

m\_bitmap.DeleteObject();

m\_bitmap.Attach((HBITMAP)LoadImage(NULL, Path\_img, IMAGE\_BITMAP, 0, 0, LR\_LOADFROMFILE));

CDC\* pDC = GetDC();

CDC memDC;

memDC.CreateCompatibleDC(pDC);

memDC.SelectObject(m\_bitmap);

((CStatic\*)GetDlgItem(IDC\_STATIC03))->GetWindowRect(m\_rect);

ScreenToClient(&m\_rect);

CClientDC dc(this);

dc.BitBlt(680, 25, 300, 250, &memDC, 0, 0, SRCCOPY);

ReleaseDC(pDC);

DeleteDC(memDC);

}

// RFIDDlg.h: 헤더 파일

//

#pragma once

#include "is\_d2xx.h"

// CRFIDDlg 대화 상자

class CRFIDDlg : public CDialogEx

{

// 생성입니다.

public:

CRFIDDlg(CWnd\* pParent = nullptr); // 표준 생성자입니다.

~CRFIDDlg();

IS\_HANDLE ftHandle = 0;

char readSerialNumber[100] = "COM07";

char changeSerialNumber[100] = "RFID01";

short usbnumber;

unsigned char wirteData[1024];

unsigned short writeLength = 0;

unsigned char readData[1024];

unsigned short readLength = 0;

BOOL flag\_r=0;

// 대화 상자 데이터입니다.

#ifdef AFX\_DESIGN\_TIME

enum { IDD = IDD\_RFID\_DIALOG };

#endif

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 지원입니다.

// 구현입니다.

protected:

HICON m\_hIcon;

// 생성된 메시지 맵 함수

virtual BOOL OnInitDialog();

afx\_msg void OnSysCommand(UINT nID, LPARAM lParam);

afx\_msg void OnPaint();

afx\_msg HCURSOR OnQueryDragIcon();

DECLARE\_MESSAGE\_MAP()

public:

afx\_msg void OnConnect();

afx\_msg void OnDisconnect();

afx\_msg void OnReadOnce();

CString m\_strRfid;

CString m\_strRfid2;

CString st\_m\_strRfid;

CString bo\_m\_strRfid;

afx\_msg void OnBnClickedButton4();

afx\_msg void OnBnClickedButton5();

afx\_msg void OnBnClickedButton6();

afx\_msg void OnBnClickedOk();

afx\_msg void OnBnClickedButton7();

void show\_st\_picture();

void show\_bo\_picture();

CString m\_strDate;

afx\_msg void OnIddRfidDialog();

};