## 酒店管理系统操作手册

## 背景

《酒店管理系统V1.0》是一款专业的酒店管理软件，软件主要功 能有：登记入住、预定信息、快捷制卡、房间设置、制卡记录、房间 状态、开锁记录、系统设置和系统日志。并集成SQM门锁接口。系 统满足对酒店布局、房间设置进行相关配置，生成图形化页面，用户 操作简单易懂，支持对酒店房态进行实时管理，并支持 制作：客人 卡、使能卡、禁用卡、服务区区卡、区域卡、楼层卡、主管卡、应急 卡、通讯卡、时间窗卡、初始化卡、授权卡、快速修复卡、解码卡、 一次性卡 15种卡类型，满足酒店门锁日常使用要求。

## 系统操作说明

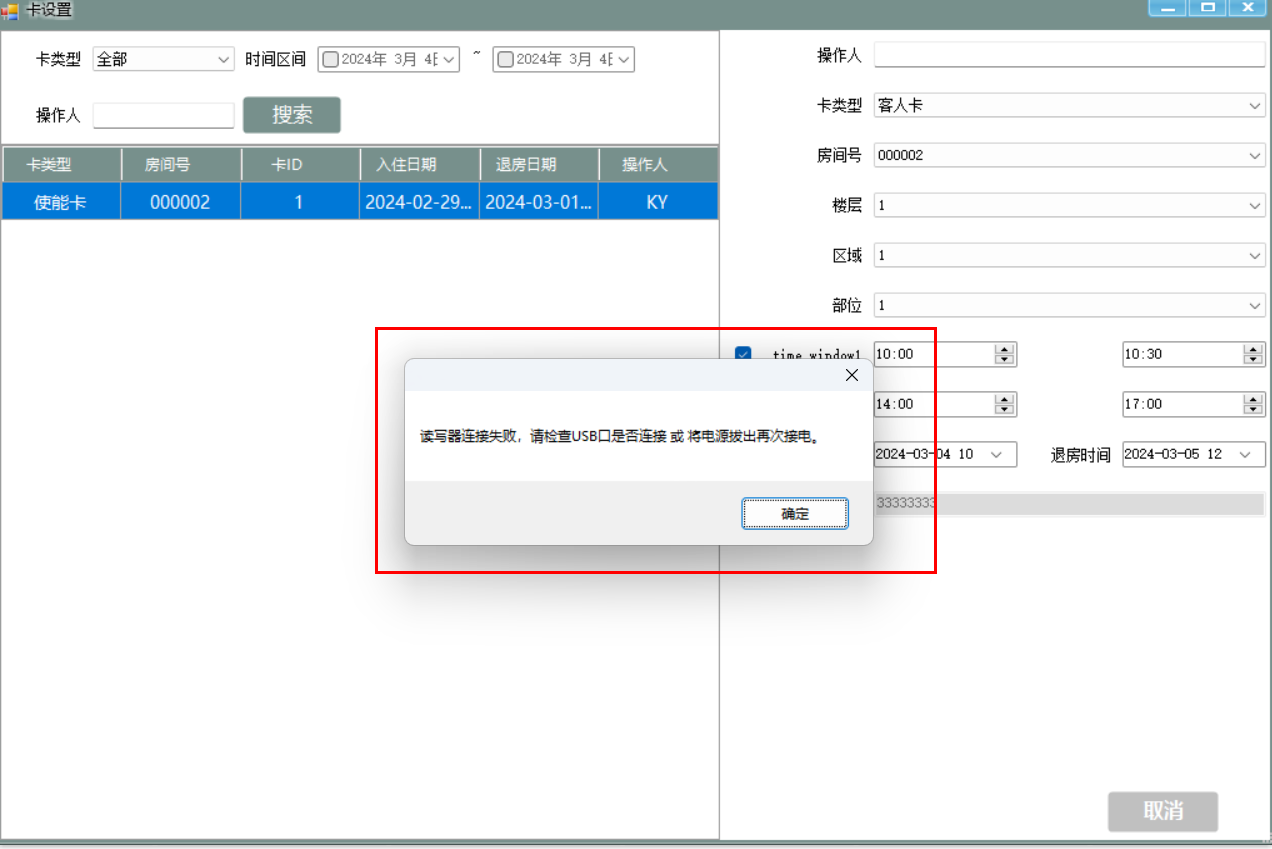
**2.1 制卡记录**

登录酒店后台系统，进入主界面，如下图：

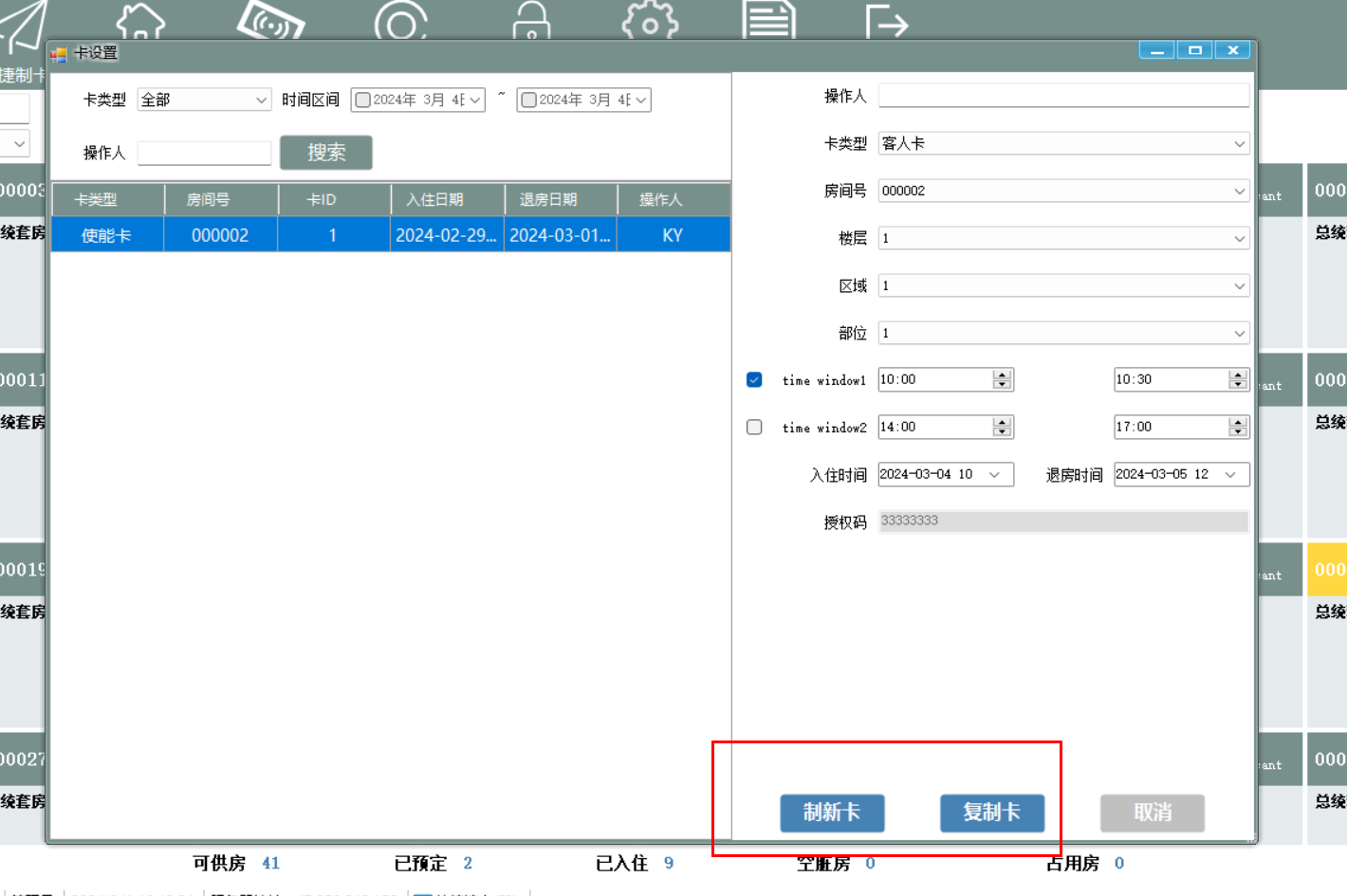


点击【制卡记录】

如没有插上制卡记录的读写器，则会弹出提示【器连接失败，请检查USB口是否连接 或 将电源拔出再次接电。】，如下图：



反之，插上对应的读写器，再次点击【制卡记录】，如下图

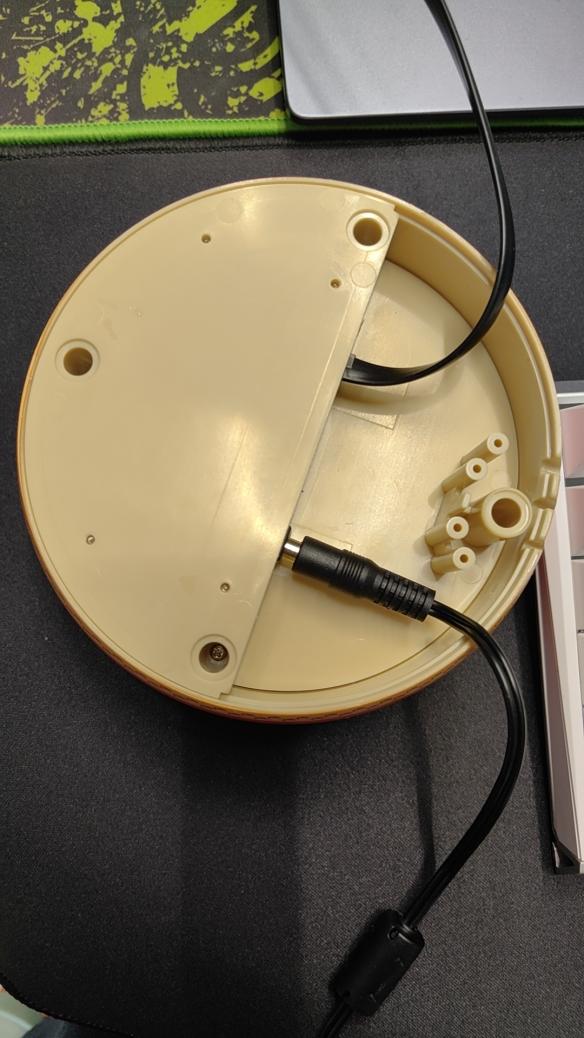


系统对应的卡类型有：客人 卡、使能卡、禁用卡、服务区区卡、区域卡、楼层卡、主管卡、应急 卡、通讯卡、时间窗卡、初始化卡、授权卡、快速修复卡、解码卡、 一次性卡 15种卡类型

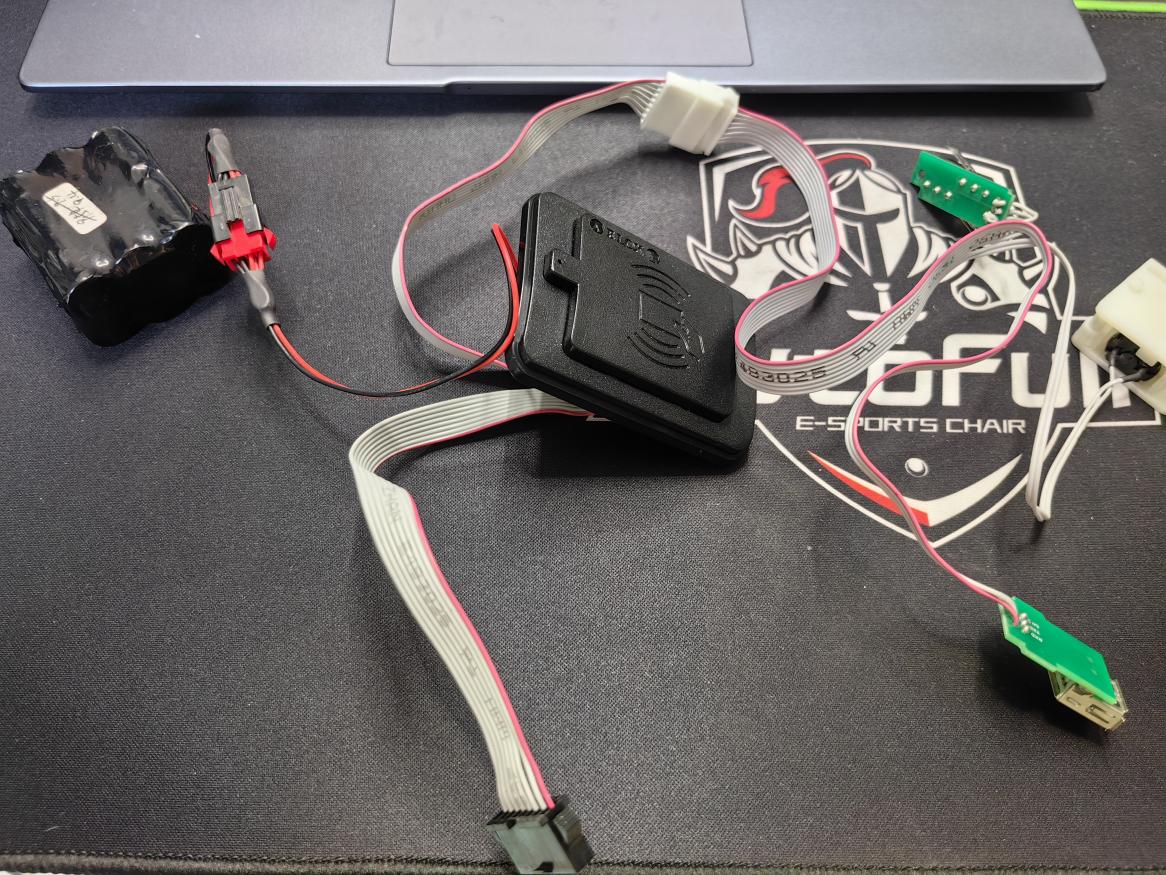
操作对应卡制作步骤如下：

1. 检查读写器



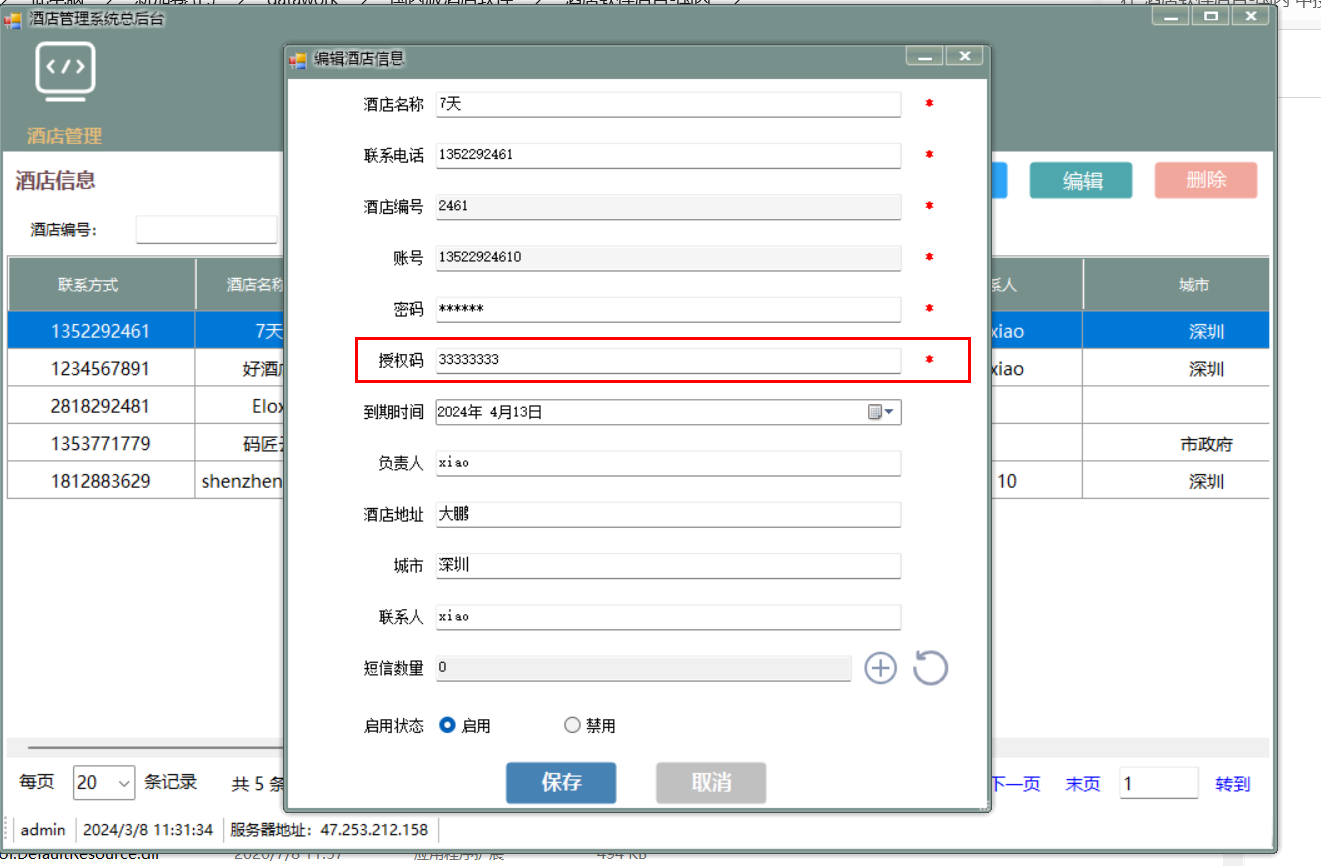


1. 门锁同步器

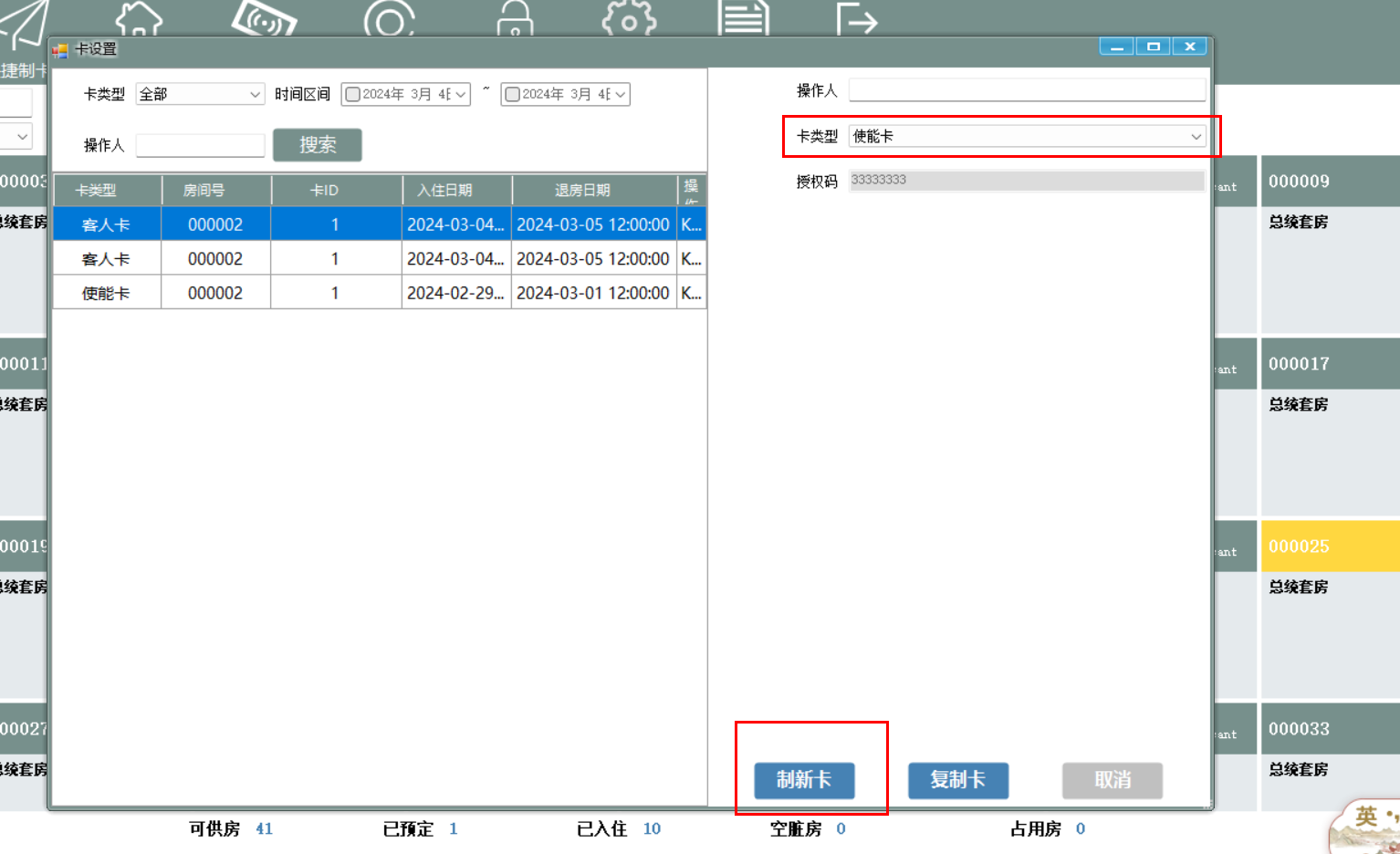


**2.1.1 制作【授权卡】**

制作卡的准入条件，初始化的时候会在系统按照授权码进行初始化



**2.1.2 制作【恢复卡】**



重新拿个卡放在读写器上，进行【制新卡】

制作成功之后，会生成对应的卡记录



#### 制卡示例代码（C#）：

##### 【btnSubmit\_Click】点击事件代码

源代码路径：hotelmanager\HotelManager\卡设置\Frm卡设置.cs

对应的方法：**btnSubmit\_Click**

private void btnSubmit\_Click(object sender, EventArgs e)

{

settingcard = true;

if (Verification())

{

return;

}

if (cboxCardCategory.Text == "guest card" || cboxCardCategory.Text == "客人卡")

{

hotel\_room model1 = new DAL\_SQLite\_hotel\_room().GetSinglehotel\_room(Convert.ToInt32(cboxRoom.SelectedValue));

if (model1.room\_status == 2)

{

if (MessageBox.Show("The room has been checked in. Do you need to make a card again", "message", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

string str16 = GetStr16();

try

{

//string returnmsg = string.Empty;

//if (true)

if (PortHelper.MakeCard(str16.ToString(), out string returnmsg))

{

hotel\_room hr = cboxRoom.SelectedItem as hotel\_room;

hotel\_makecard\_log model = new hotel\_makecard\_log

{

hotel\_id = Program.user.hotel\_id,

opertorid = Program.user.id,

opertor = Program.user.realname,

guest\_name = "",

guest\_phone = "",

serial\_number = "1",

room\_number = cboxRoom.Text,

room\_id = Convert.ToInt32(cboxRoom.SelectedValue),

building = cboxBuilding.Text,

section = cboxSection.Text,

zone = cboxZone.Text,

operated\_time = dtpOperatedDate.Value.ToString("yyyy-MM-dd HH:mm:ss"),

period\_time = dtpPeriodDate.Value.ToString("yyyy-MM-dd HH:mm:ss"),

is\_new = 1

};

switch (cboxCardCategory.Text)

{

case "guest card":

case "客人卡":

model.card\_type = "00";

break;

case "enable card":

case "使能卡":

model.card\_type = "02";

break;

case "disable card":

case "禁用卡":

model.card\_type = "03";

break;

case "section card":

case "服务区区卡":

model.card\_type = "04";

break;

case "zone card":

case "区域卡":

model.card\_type = "05";

break;

case "building card":

case "楼层卡":

model.card\_type = "06";

break;

case "master card":

case "主管卡":

model.card\_type = "07";

break;

case "emergency card":

case "紧急卡":

model.card\_type = "08";

break;

case "communication card":

case "通信卡":

model.card\_type = "09";

break;

case "time window card":

case "时间窗口卡":

model.card\_type = "12";

break;

case "initialization card":

case "初始化卡":

model.card\_type = "13";

break;

case "authorization card":

case "授权卡":

model.card\_type = "15";

break;

case "disposable card":

case "一次性卡":

model.card\_type = "17";

break;

case "decoder card":

case "解码卡":

model.card\_type = "21";

break;

case "quick fix card":

case "快速修复卡":

model.card\_type = "23";

break;

}

ResultMsg msg = new ResultMsg();

if (MyCommon.IsConnectedToInternet())

{

model.is\_new = 0;

msg = new DAL\_SQLite\_hotel\_makecard\_log().Insert(model);

new DALhotel\_makecard\_log().Insert(model);

}

else

{

msg = new DAL\_SQLite\_hotel\_makecard\_log().Insert(model);

}

if (msg.Success)

{

if (cboxCardCategory.Text == "guest card" || cboxCardCategory.Text == "客人卡")

{

new DAL\_SQLite\_hotel\_room().UpdateRoomStatus(new hotel\_room { id = model.room\_id, room\_status = 2 });

new DALhotel\_room().UpdateRoomStatus(new hotel\_room { id = model.room\_id, room\_status = 2 });

BindData();

}

else

{

BindData();

}

MessageBox.Show("Business card printing success");

}

else

{

MessageBox.Show("Business card printing fail");

}

}

else

{

try

{

Program.mkPort.Close();

Program.mkPort.Open();

if (!PortHelper.HandShake())

{

if (Program.ci == "en-US")

{

MessageBox.Show("The reader failed to connect. Please check whether the USB port is connected or unplug the power supply and connect it again.");

}

else

{

MessageBox.Show("读写器连接失败，请检查USB口是否连接 或 将电源拔出再次接电。");

}

btnSubmit.Visible = false;

btnCopy.Visible = false;

Program.mkPort.Close();

}

}

catch (Exception)

{

if (Program.ci == "en-US")

{

MessageBox.Show("The reader failed to connect. Please check whether the USB port is connected or unplug the power supply and connect it again.");

}

else

{

MessageBox.Show("读写器连接失败，请检查USB口是否连接 或 将电源拔出再次接电。");

}

btnSubmit.Visible = false;

btnCopy.Visible = false;

Program.mkPort.Close();

}

if (returnmsg == "A0AA")

{

MessageBox.Show("Business card printing fail,Please check whether the card is put in");

}

else if (string.IsNullOrEmpty(returnmsg))

{

MessageBox.Show("Business card printing fail,Please check whether the power supply is connected normally");

}

}

}

catch (FormatException)

{

MessageBox.Show("Business card printing fail");

}

}

}

else

{

string str16 = GetStr16();

try

{

//string returnmsg = "";

//if (true)

if (PortHelper.MakeCard(str16.ToString(), out string returnmsg))

{

hotel\_room hr = cboxRoom.SelectedItem as hotel\_room;

hotel\_makecard\_log model = new hotel\_makecard\_log

{

hotel\_id = Program.user.hotel\_id,

opertorid = Program.user.id,

opertor = Program.user.realname,

guest\_name = "",

guest\_phone = "",

serial\_number = "1",

room\_number = cboxRoom.Text,

room\_id = Convert.ToInt32(cboxRoom.SelectedValue),

building = cboxBuilding.Text,

section = cboxSection.Text,

zone = cboxZone.Text,

operated\_time = dtpOperatedDate.Value.ToString("yyyy-MM-dd HH:mm:ss"),

period\_time = dtpPeriodDate.Value.ToString("yyyy-MM-dd HH:mm:ss"),

is\_new = 1

};

switch (cboxCardCategory.Text)

{

case "guest card":

case "客人卡":

model.card\_type = "00";

break;

case "enable card":

case "使能卡":

model.card\_type = "02";

break;

case "disable card":

case "禁用卡":

model.card\_type = "03";

break;

case "section card":

case "服务区区卡":

model.card\_type = "04";

break;

case "zone card":

case "区域卡":

model.card\_type = "05";

break;

case "building card":

case "楼层卡":

model.card\_type = "06";

break;

case "master card":

case "主管卡":

model.card\_type = "07";

break;

case "emergency card":

case "紧急卡":

model.card\_type = "08";

break;

case "communication card":

case "通信卡":

model.card\_type = "09";

break;

case "time window card":

case "时间窗口卡":

model.card\_type = "12";

break;

case "initialization card":

case "初始化卡":

model.card\_type = "13";

break;

case "authorization card":

case "授权卡":

model.card\_type = "15";

break;

case "disposable card":

case "一次性卡":

model.card\_type = "17";

break;

case "decoder card":

case "解码卡":

model.card\_type = "21";

break;

case "quick fix card":

case "快速修复卡":

model.card\_type = "23";

break;

}

ResultMsg msg = new ResultMsg();

if (MyCommon.IsConnectedToInternet())

{

model.is\_new = 0;

msg = new DAL\_SQLite\_hotel\_makecard\_log().Insert(model);

new DALhotel\_makecard\_log().Insert(model);

}

else

{

msg = new DAL\_SQLite\_hotel\_makecard\_log().Insert(model);

}

if (msg.Success)

{

if (cboxCardCategory.Text == "guest card" || cboxCardCategory.Text == "客人卡")

{

string update\_time = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");

if (MyCommon.IsConnectedToInternet())

{

new DAL\_SQLite\_hotel\_room().UpdateRoomStatus(new hotel\_room { id = model.room\_id, room\_status = 2, is\_new = 0, update\_time = update\_time });

new DALhotel\_room().UpdateRoomStatus(new hotel\_room { id = model.room\_id, room\_status = 2, is\_new = 0, update\_time = update\_time });

}

else

{

new DAL\_SQLite\_hotel\_room().UpdateRoomStatus(new hotel\_room { id = model.room\_id, room\_status = 2, is\_new = 2, update\_time = update\_time });

}

BindData();

}

else

{

BindData();

}

MessageBox.Show("Business card printing success");

}

else

{

MessageBox.Show("Business card printing fail");

}

}

else

{

try

{

Program.mkPort.Close();

Program.mkPort.Open();

if (!PortHelper.HandShake())

{

if (Program.ci == "en-US")

{

MessageBox.Show("The reader failed to connect. Please check whether the USB port is connected or unplug the power supply and connect it again.");

}

else

{

MessageBox.Show("读写器连接失败，请检查USB口是否连接 或 将电源拔出再次接电。");

}

btnSubmit.Visible = false;

btnCopy.Visible = false;

Program.mkPort.Close();

}

}

catch (Exception)

{

if (Program.ci == "en-US")

{

MessageBox.Show("The reader failed to connect. Please check whether the USB port is connected or unplug the power supply and connect it again.");

}

else

{

MessageBox.Show("读写器连接失败，请检查USB口是否连接 或 将电源拔出再次接电。");

}

btnSubmit.Visible = false;

btnCopy.Visible = false;

Program.mkPort.Close();

}

if (returnmsg == "A0AA")

{

MessageBox.Show("Business card printing fail,Please check whether the card is put in");

}

else if (string.IsNullOrEmpty(returnmsg))

{

MessageBox.Show("Business card printing fail,Please check whether the power supply is connected normally");

}

}

}

catch (FormatException)

{

MessageBox.Show("Business card printing fail");

}

}

}

else

{

string str16 = GetStr16();

try

{

//string returnmsg = "";

//if (true)

if (PortHelper.MakeCard(str16.ToString(), out string returnmsg))

{

hotel\_room hr = cboxRoom.SelectedItem as hotel\_room;

hotel\_makecard\_log model = new hotel\_makecard\_log

{

hotel\_id = Program.user.hotel\_id,

opertorid = Program.user.id,

opertor = Program.user.realname,

guest\_name = "",

guest\_phone = "",

serial\_number = "1",

room\_number = cboxRoom.Text,

room\_id = Convert.ToInt32(cboxRoom.SelectedValue),

building = cboxBuilding.Text,

section = cboxSection.Text,

zone = cboxZone.Text,

operated\_time = dtpOperatedDate.Value.ToString("yyyy-MM-dd HH:mm:ss"),

period\_time = dtpPeriodDate.Value.ToString("yyyy-MM-dd HH:mm:ss"),

is\_new = 1

};

switch (cboxCardCategory.Text)

{

case "guest card":

case "客人卡":

model.card\_type = "00";

break;

case "enable card":

case "使能卡":

model.card\_type = "02";

break;

case "disable card":

case "禁用卡":

model.card\_type = "03";

break;

case "section card":

case "服务区区卡":

model.card\_type = "04";

break;

case "zone card":

case "区域卡":

model.card\_type = "05";

break;

case "building card":

case "楼层卡":

model.card\_type = "06";

break;

case "master card":

case "主管卡":

model.card\_type = "07";

break;

case "emergency card":

case "紧急卡":

model.card\_type = "08";

break;

case "communication card":

case "通信卡":

model.card\_type = "09";

break;

case "time window card":

case "时间窗口卡":

model.card\_type = "12";

break;

case "initialization card":

case "初始化卡":

model.card\_type = "13";

break;

case "authorization card":

case "授权卡":

model.card\_type = "15";

break;

case "disposable card":

case "一次性卡":

model.card\_type = "17";

break;

case "decoder card":

case "解码卡":

model.card\_type = "21";

break;

case "quick fix card":

case "快速修复卡":

model.card\_type = "23";

break;

}

ResultMsg msg = new ResultMsg();

if (MyCommon.IsConnectedToInternet())

{

model.is\_new = 0;

msg = new DAL\_SQLite\_hotel\_makecard\_log().Insert(model);

new DALhotel\_makecard\_log().Insert(model);

}

else

{

msg = new DAL\_SQLite\_hotel\_makecard\_log().Insert(model);

}

if (msg.Success)

{

if (cboxCardCategory.Text == "guest card" || cboxCardCategory.Text == "客人卡")

{

new DAL\_SQLite\_hotel\_room().UpdateRoomStatus(new hotel\_room { id = model.room\_id, room\_status = 2 });

new DALhotel\_room().UpdateRoomStatus(new hotel\_room { id = model.room\_id, room\_status = 2 });

BindData();

}

else

{

BindData();

}

MessageBox.Show("Business card printing success");

}

else

{

MessageBox.Show("Business card printing fail");

}

}

else

{

try

{

Program.mkPort.Close();

Program.mkPort.Open();

if (!PortHelper.HandShake())

{

if (Program.ci == "en-US")

{

MessageBox.Show("The reader failed to connect. Please check whether the USB port is connected or unplug the power supply and connect it again.");

}

else

{

MessageBox.Show("读写器连接失败，请检查USB口是否连接 或 将电源拔出再次接电。");

}

btnSubmit.Visible = false;

btnCopy.Visible = false;

Program.mkPort.Close();

}

}

catch (Exception)

{

if (Program.ci == "en-US")

{

MessageBox.Show("The reader failed to connect. Please check whether the USB port is connected or unplug the power supply and connect it again.");

}

else

{

MessageBox.Show("读写器连接失败，请检查USB口是否连接 或 将电源拔出再次接电。");

}

btnSubmit.Visible = false;

btnCopy.Visible = false;

Program.mkPort.Close();

}

if (returnmsg == "A0AA")

{

MessageBox.Show("Business card printing fail,Please check whether the card is put in");

}

else if (string.IsNullOrEmpty(returnmsg))

{

MessageBox.Show("Business card printing fail,Please check whether the power supply is connected normally");

}

}

}

catch (FormatException)

{

MessageBox.Show("Business card printing fail");

}

}

}

##### **【MakeCard】方法代码**

源代码路径：hotelmanager\HotelManager\PostHelper.cs

对应的方法：**MakeCard**

public static bool MakeCard(string str16,out string returnmsg)

{

LogManage.WriteLog("开始制卡了");

LogManage.WriteLog("先关闭制卡器COM口连接");

Program.mkPort.Close();

LogManage.WriteLog("先打开制卡器COM口连接");

Program.mkPort.Open();

LogManage.WriteLog("发送A0 握手");

//握手

byte[] data = MyCommon.HexStringToByteArray("A0");

// Send the binary data out the port

Program.mkPort.Write(data, 0, data.Length);

Thread.Sleep(250);

LogManage.WriteLog("休眠了250毫秒");

//读取字节个数

int n = Program.mkPort.BytesToRead;

byte[] buf = new byte[n];

Program.mkPort.Read(buf, 0, n);

returnmsg = MyCommon.ByteArrayToHexString(buf).Replace(" ", "");

LogManage.WriteLog("返回值：" + returnmsg);

if (returnmsg == "A0")

{

byte[] data1 = MyCommon.HexStringToByteArray(str16);

//Thread.Sleep(1000);

LogManage.WriteLog("开始输入制卡指令：" + str16);

// Send the binary data out the port

Program.mkPort.Write(data1, 0, data1.Length);

Thread.Sleep(250);

//读取字节个数

int n1 = Program.mkPort.BytesToRead;

byte[] buf1 = new byte[n1];

Program.mkPort.Read(buf1, 0, n1);

returnmsg = MyCommon.ByteArrayToHexString(buf1).Replace(" ", "");

LogManage.WriteLog("接收制卡回应：" + returnmsg);

LogManage.WriteLog("关闭制卡器COM口连接");

Program.mkPort.Close();

if (returnmsg != "AA"&& returnmsg != "A0AA")

{

return true;

}

else

{

return false;

}

}

else

{

return false;

}

}

##### **【DALhotel\_makecard\_log.Insert】方法**

源代码路径： \hotelmanager\HotelManager\卡设置\Frm卡设置.cs

对应的方法：实例化hotel\_makecard\_log 表数据，插入数据

if (MyCommon.IsConnectedToInternet())

{

model.is\_new = 0;

msg = new DAL\_SQLite\_hotel\_makecard\_log().Insert(model);

new DALhotel\_makecard\_log().Insert(model);

}

else

{

msg = new DAL\_SQLite\_hotel\_makecard\_log().Insert(model);

}

##### 【BindData】方法

源代码路径：\hotelmanager\HotelManager\卡设置\Frm卡设置.cs

对应的方法：**BindData**

数据绑定成功之后，弹出对应的提示，对应的卡信息制作完成

private void BindData()

{

string cardtype = string.Empty;

switch (cboxCardType.Text)

{

case "guest card":

case "客人卡":

cardtype = "00";

break;

case "enable card":

case "使能卡":

cardtype = "02";

break;

case "disable card":

case "禁用卡":

cardtype = "03";

break;

case "section card":

case "服务区区卡":

cardtype = "04";

break;

case "zone card":

case "区域卡":

cardtype = "05";

break;

case "building card":

case "楼层卡":

cardtype = "06";

break;

case "master card":

case "主管卡":

cardtype = "07";

break;

case "emergency card":

case "紧急卡":

cardtype = "08";

break;

case "communication card":

case "通信卡":

cardtype = "09";

break;

case "time window card":

case "时间窗口卡":

cardtype = "12";

break;

case "initialization card":

case "初始化卡":

cardtype = "13";

break;

case "authorization card":

case "授权卡":

cardtype = "15";

break;

case "disposable card":

case "一次性卡":

cardtype = "17";

break;

case "decoder card":

case "解码卡":

cardtype = "21";

break;

case "quick fix card":

case "快速修复卡":

cardtype = "23";

break;

}

dgv.DataSource = new DAL\_SQLite\_hotel\_makecard\_log().QueryList(new hotel\_makecard\_log

{

opertor = txtSearchOperator.Text,

start\_date = dtpStart.Checked ? dtpStart.Value.ToString("yyyy-MM-dd HH:mm:ss") : "",

end\_date = dtpEnd.Checked ? dtpEnd.Value.ToString("yyyy-MM-dd HH:mm:ss") : "",

card\_type = cardtype

});

}

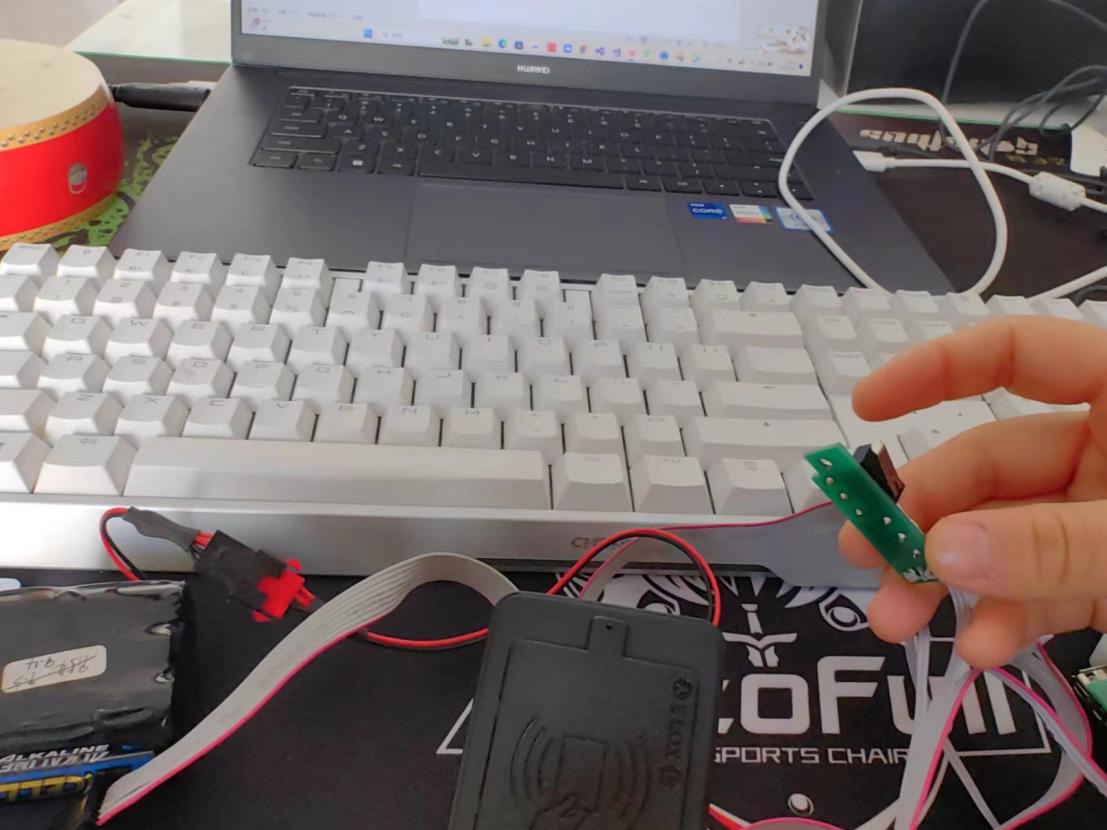
把制作好的使能卡放在门锁同步器进行写锁：



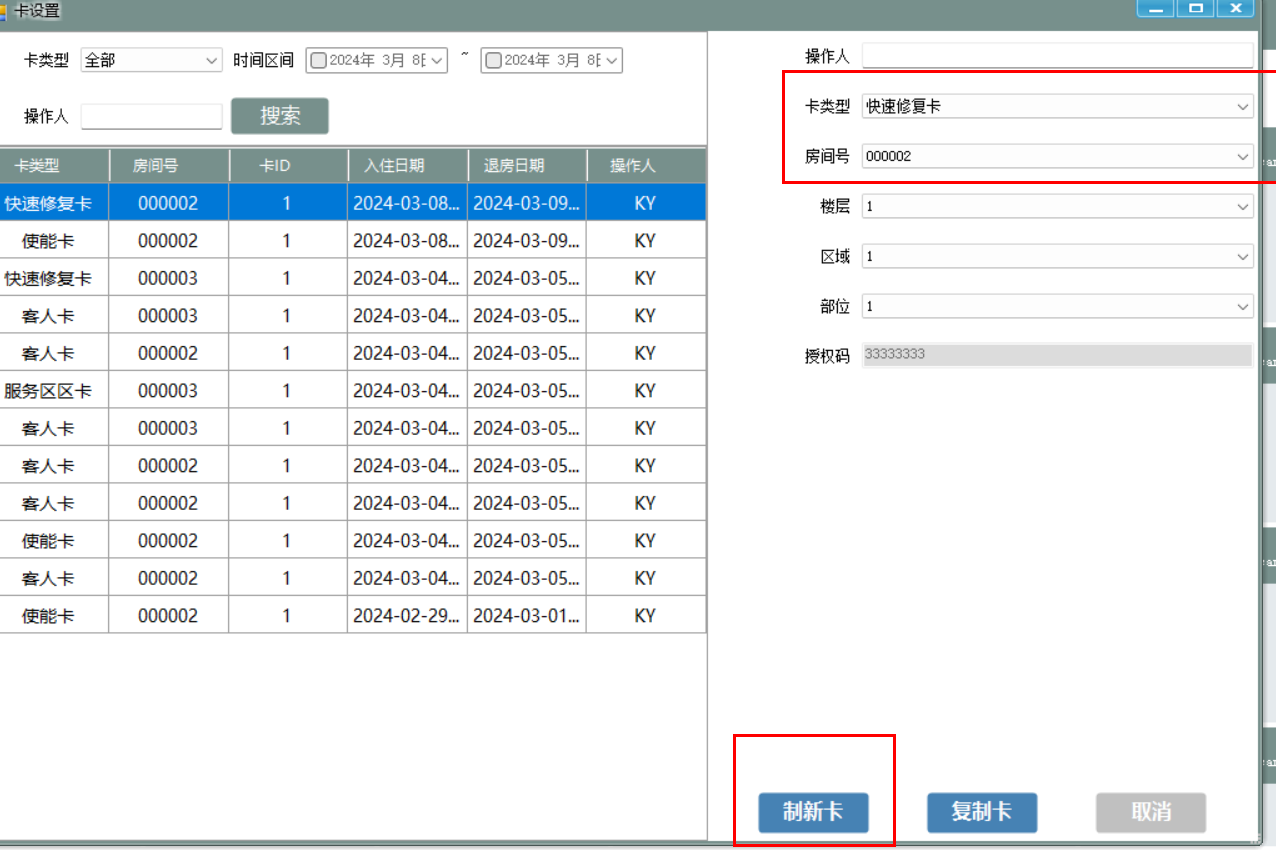
绿灯一直亮着

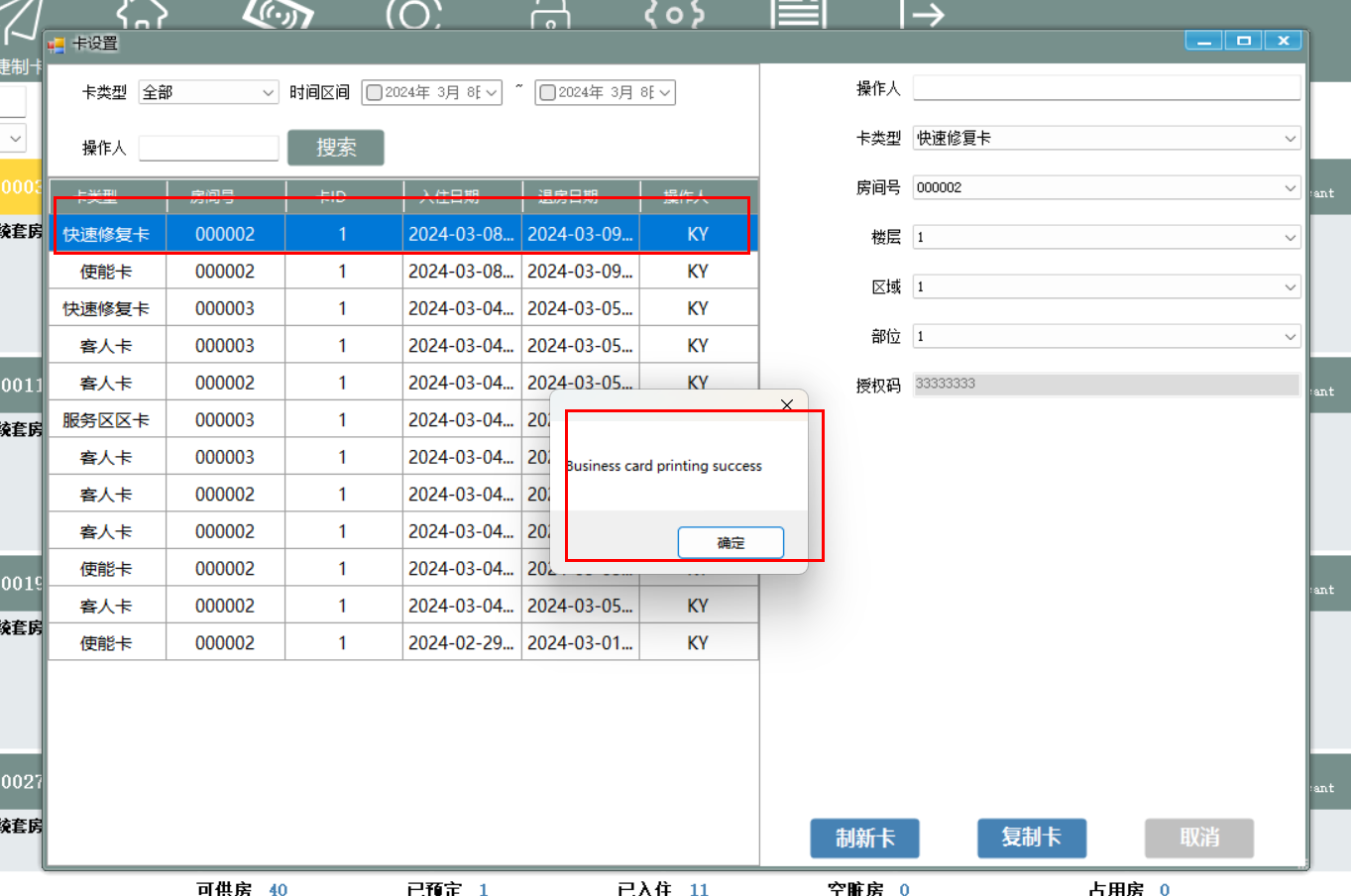


把对应的电机线的开关按一下，绿灯熄灭，说明对应的卡写到锁里，验证成功



**2.1.3 制作【快修卡】**





制卡成功

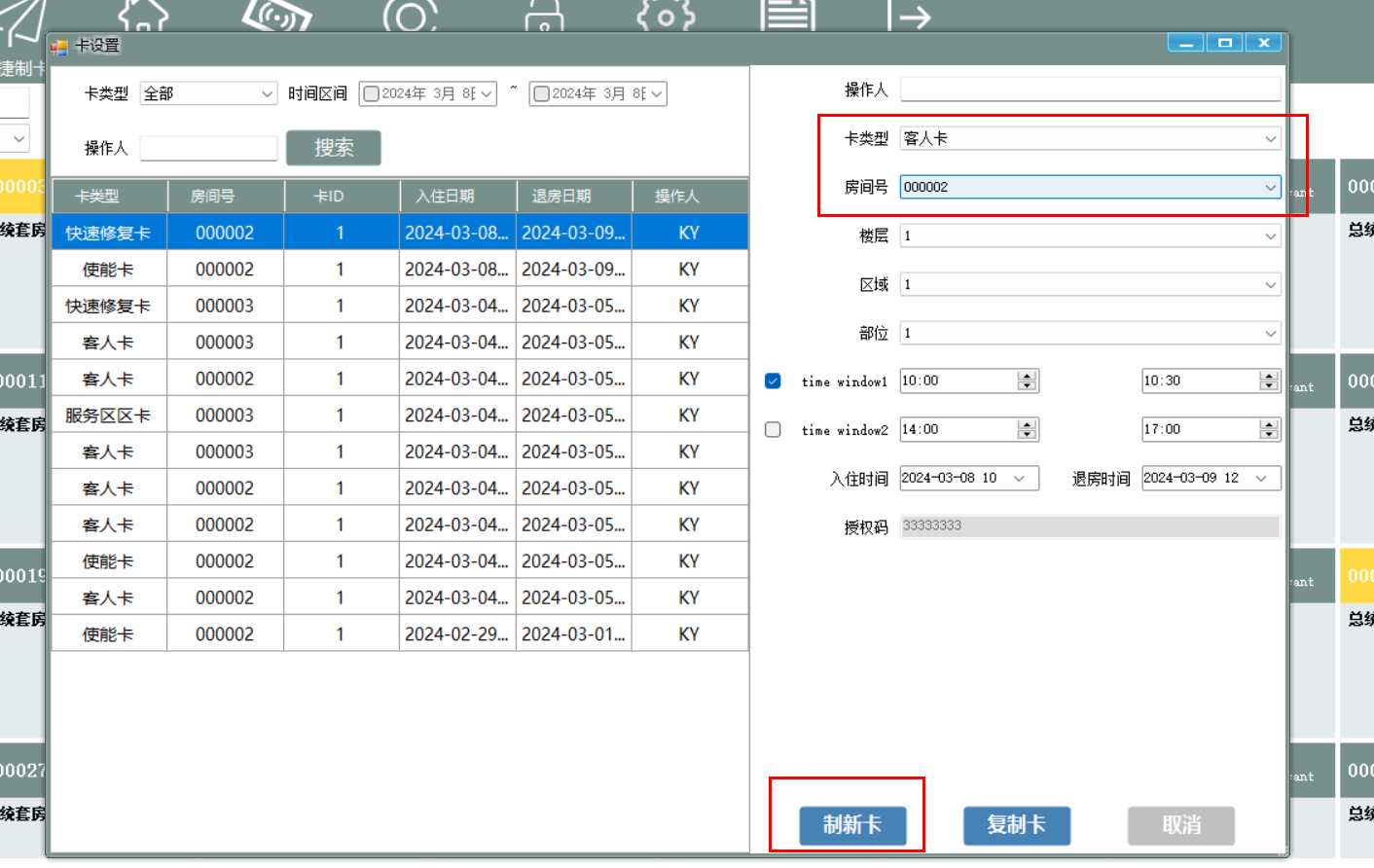
把制作好的恢复卡放在门锁同步器进行写锁：

绿灯闪烁，等待几秒绿灯熄灭，说明对应的卡写到锁里，验证成功

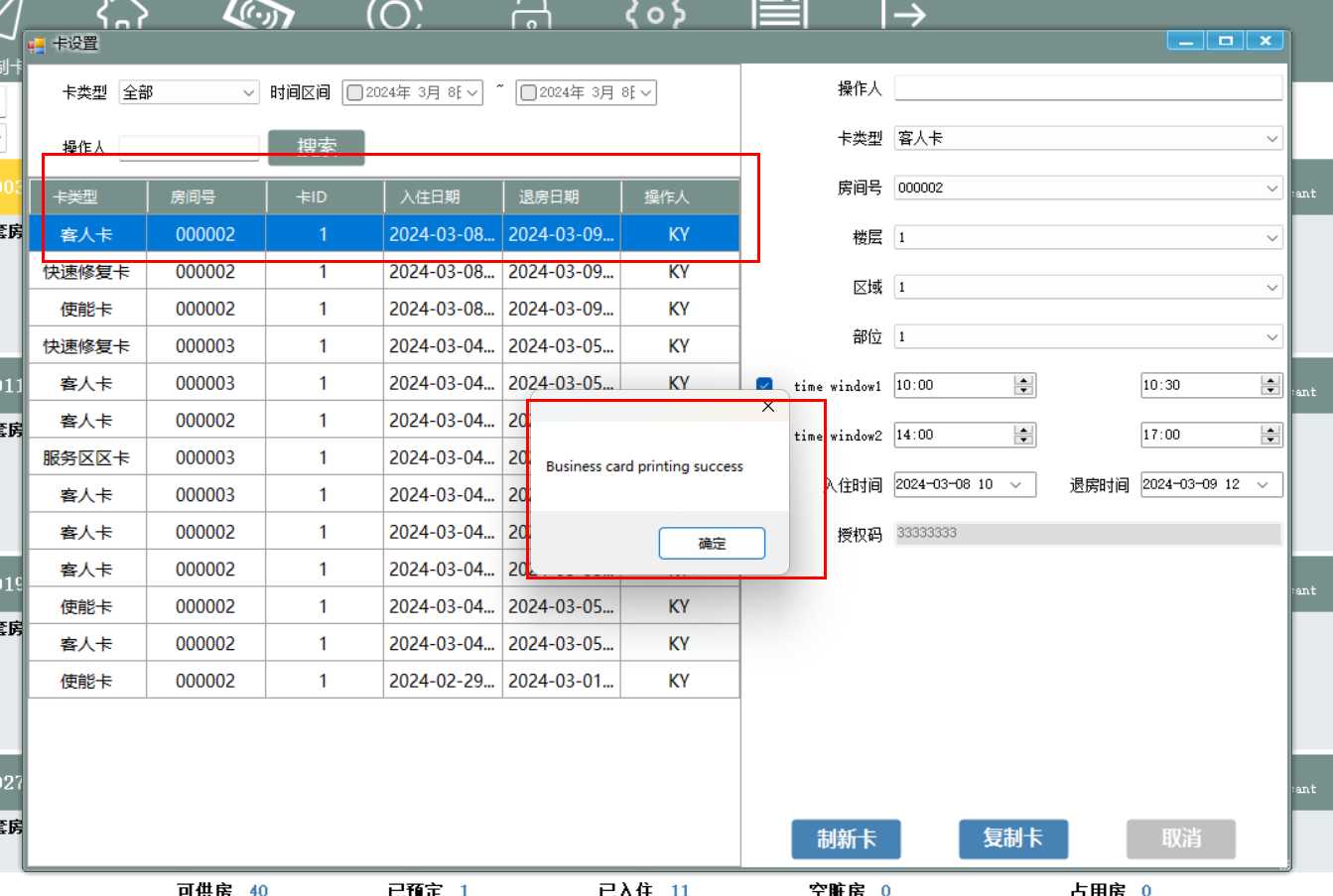


**2.1.4 制作【宾客卡】**

把对应的卡片放到读写器上，前台界面操作如下图：



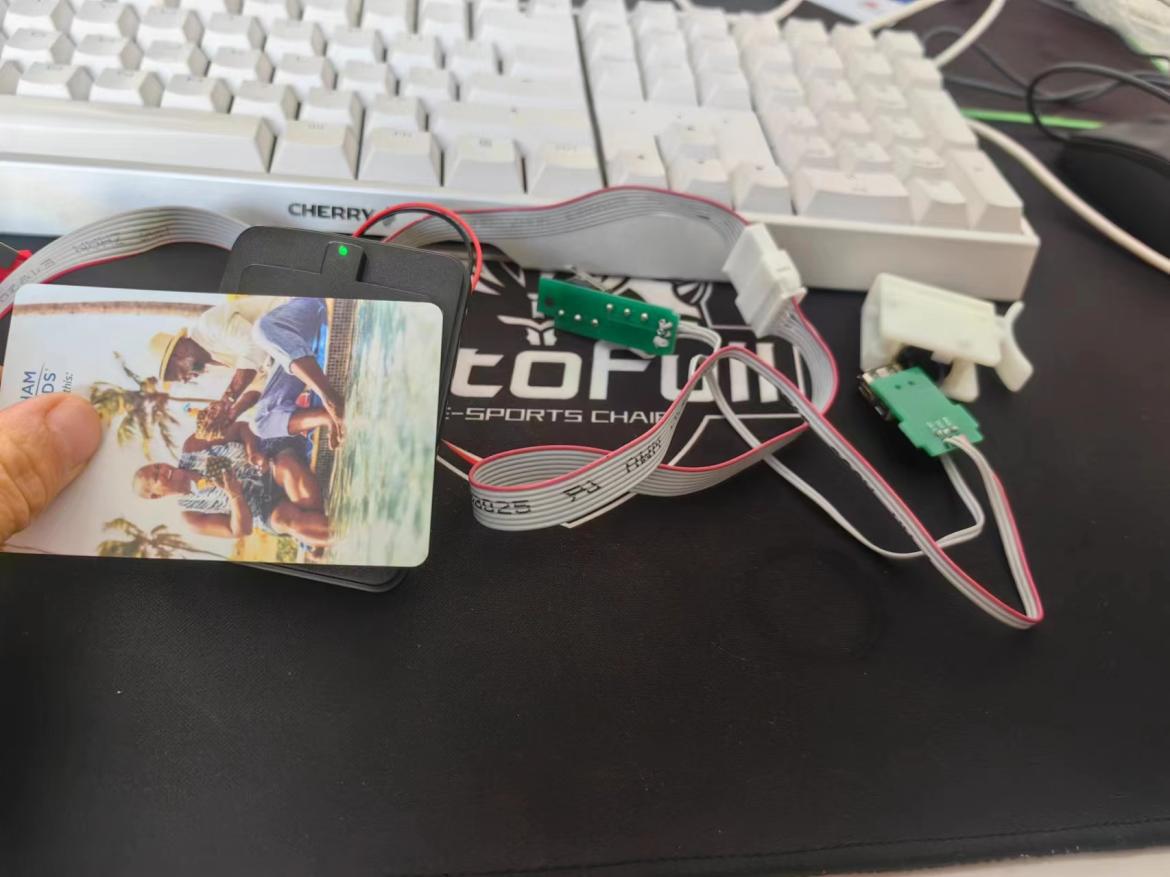
填写界面对应的信息，点击【制新卡】按钮



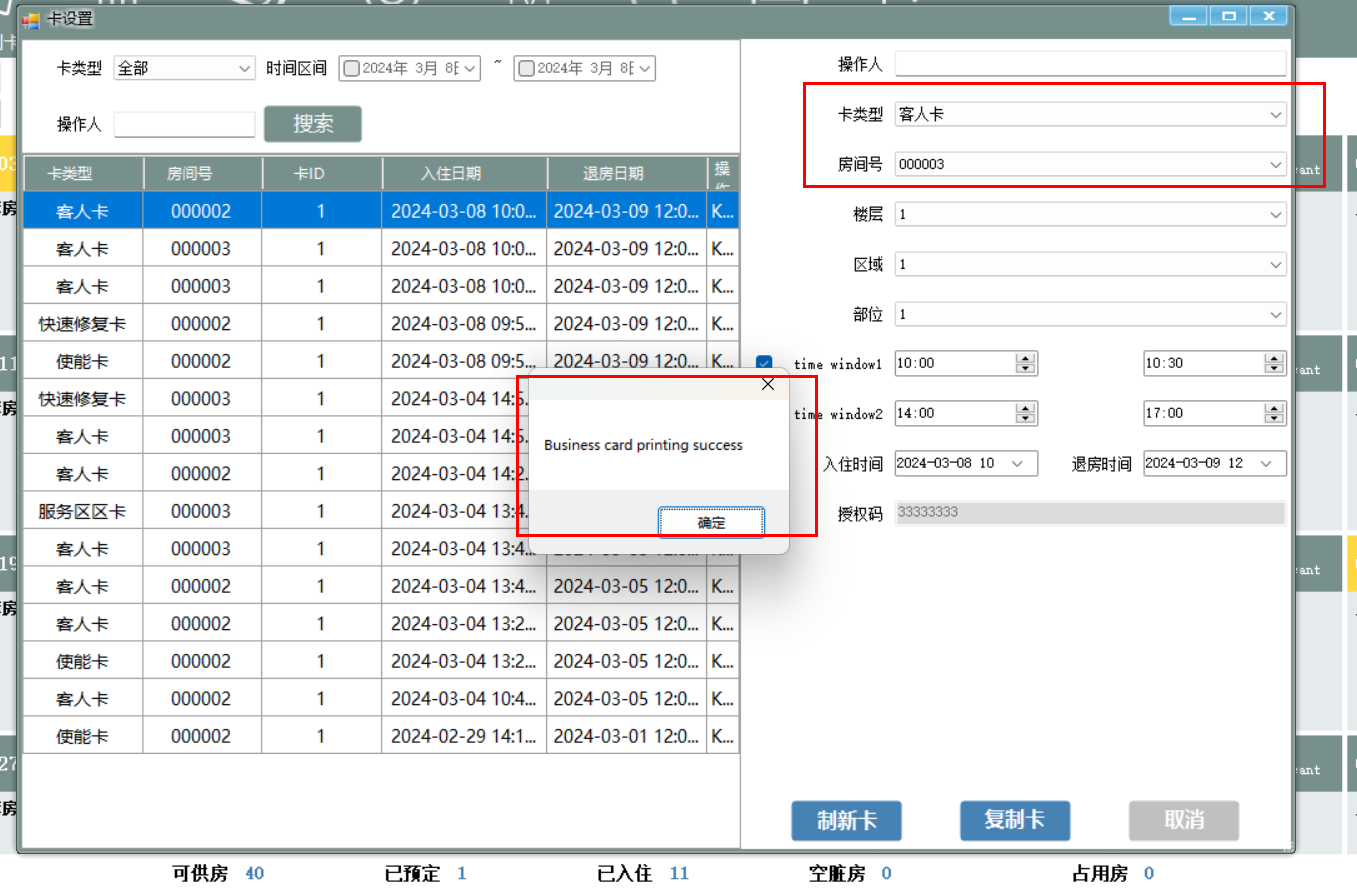
弹出提示【Business card printing success】,且生成一条制卡记录，则说明制卡成功

对门锁同步器上进行验证

放在门锁同步器上，绿灯闪烁，并且对应的电机锁打开，说明可以开锁，验证成功

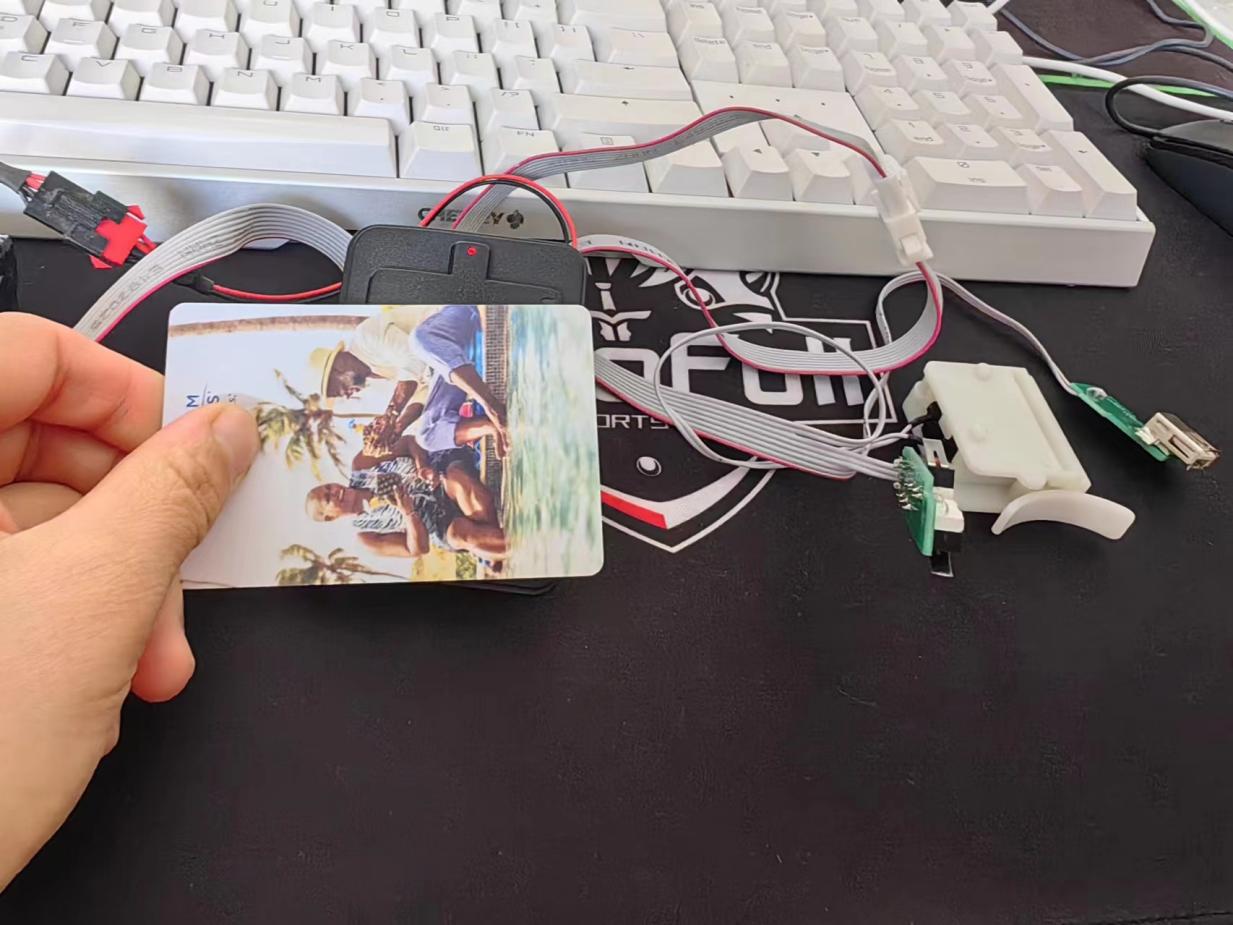


反之换个房间进行制卡



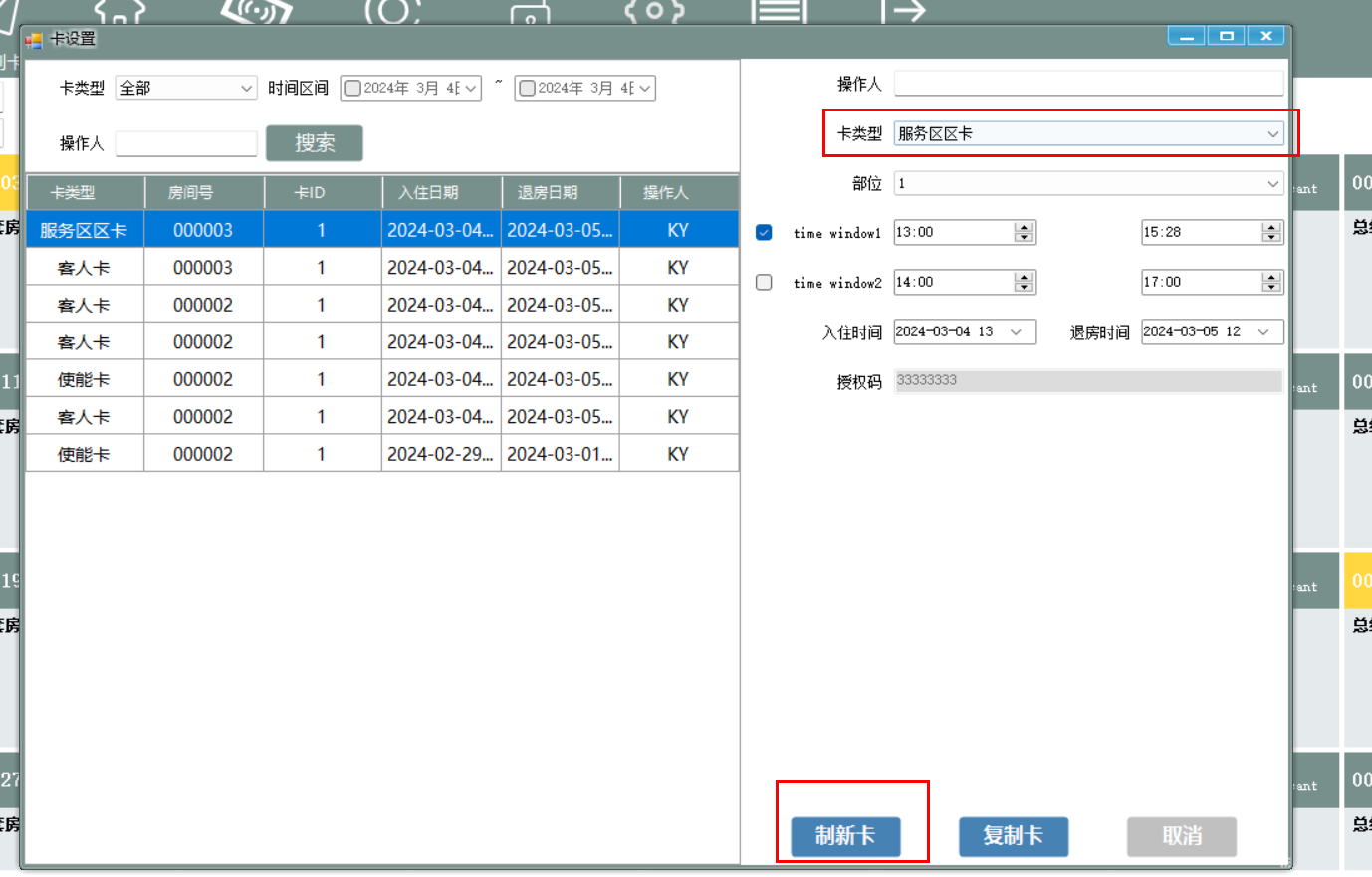
制卡成功

门锁同步器进行验证时,红灯闪烁，说明不能开锁



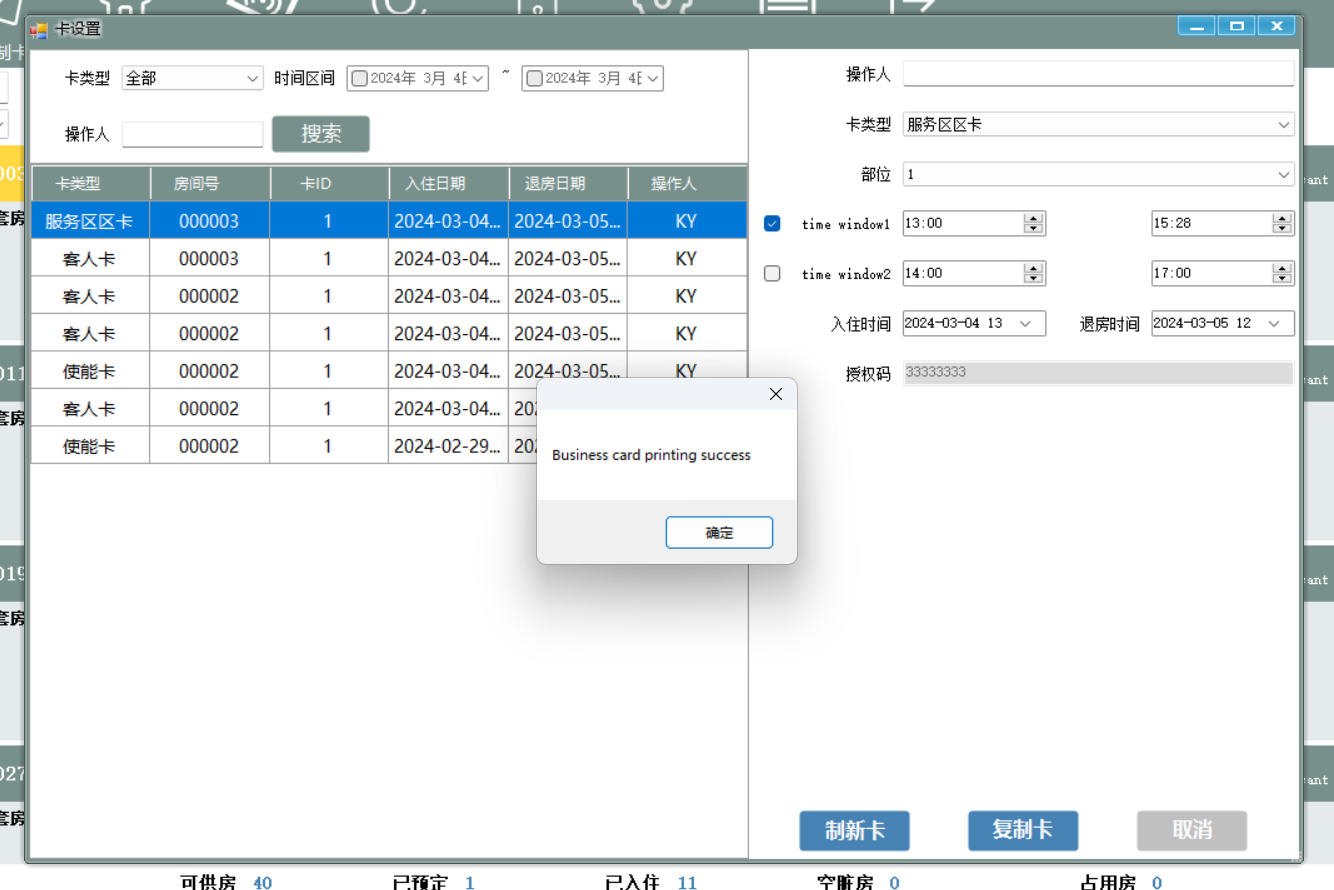
注：连接器锁反锁之后贵宾卡也是无法打开

**2.1.5 制作【服务卡】**



重新拿个卡放在读写器上，进行【制新卡】

制作成功之后，会生成对应的卡记录



把制作好的服务卡放在门锁同步器进行验证：



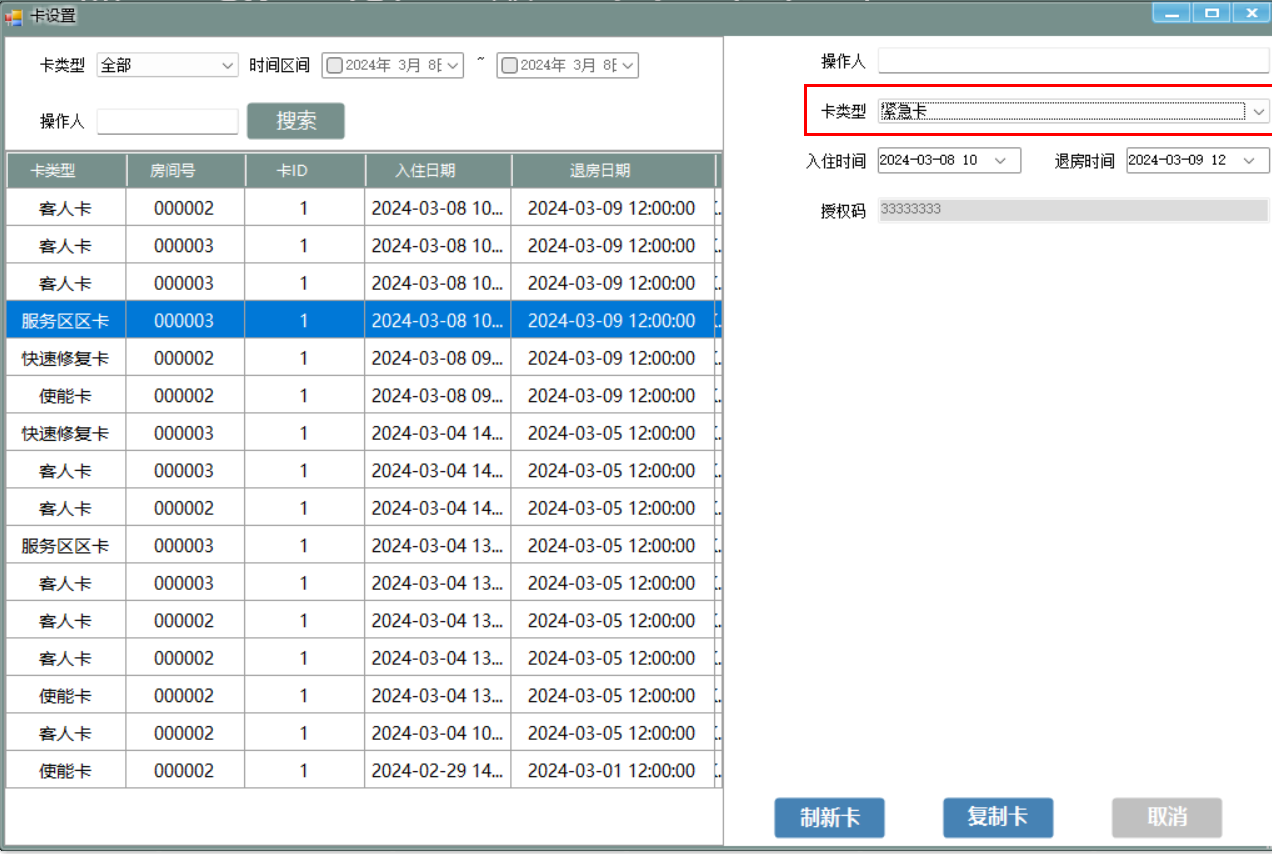
进行服务卡验证时不仅绿灯闪烁，电机线连接锁开关也会打开

但是如果门反锁就不会打开锁(按住上开关)，红灯闪烁

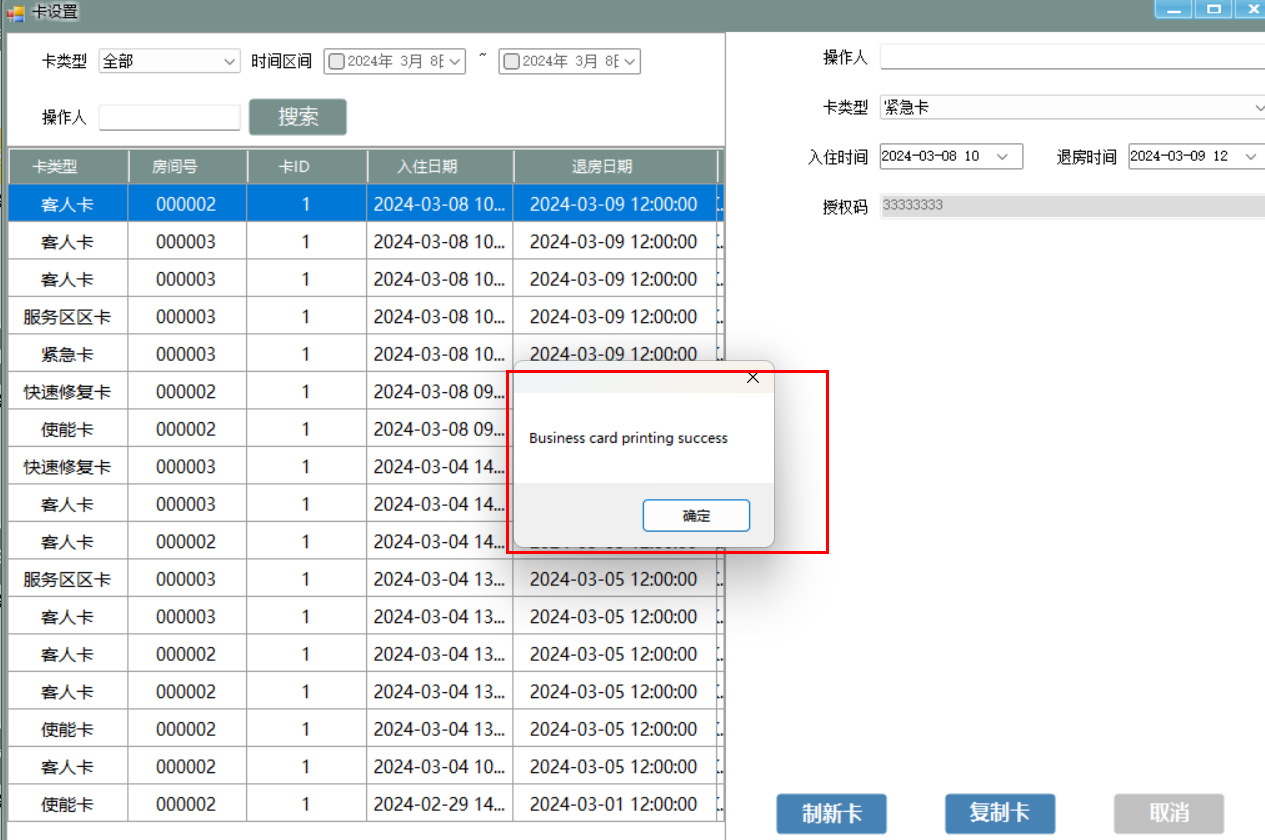


**注：时间窗只针对服务卡，并且门锁反锁服务卡是无法打开**

**2.1.6 制作【应急卡】**



放需要制作的卡在读写器上，点击**【制新卡】**



制作成功

门锁同步器进行验证：

绿灯闪烁，电机线的锁也会打开，即使门反锁也会打开





注：视频中应急卡动作是分解的（每操作一次状态变一次），就是刷一下锁打开，然后再刷一下锁关闭，这个动作分解在本次验证中没有成功，也就是本次刷应急卡的时候，锁打开，过几秒锁就自动关闭了

等待一个小时进行验证动作分解，验证成功