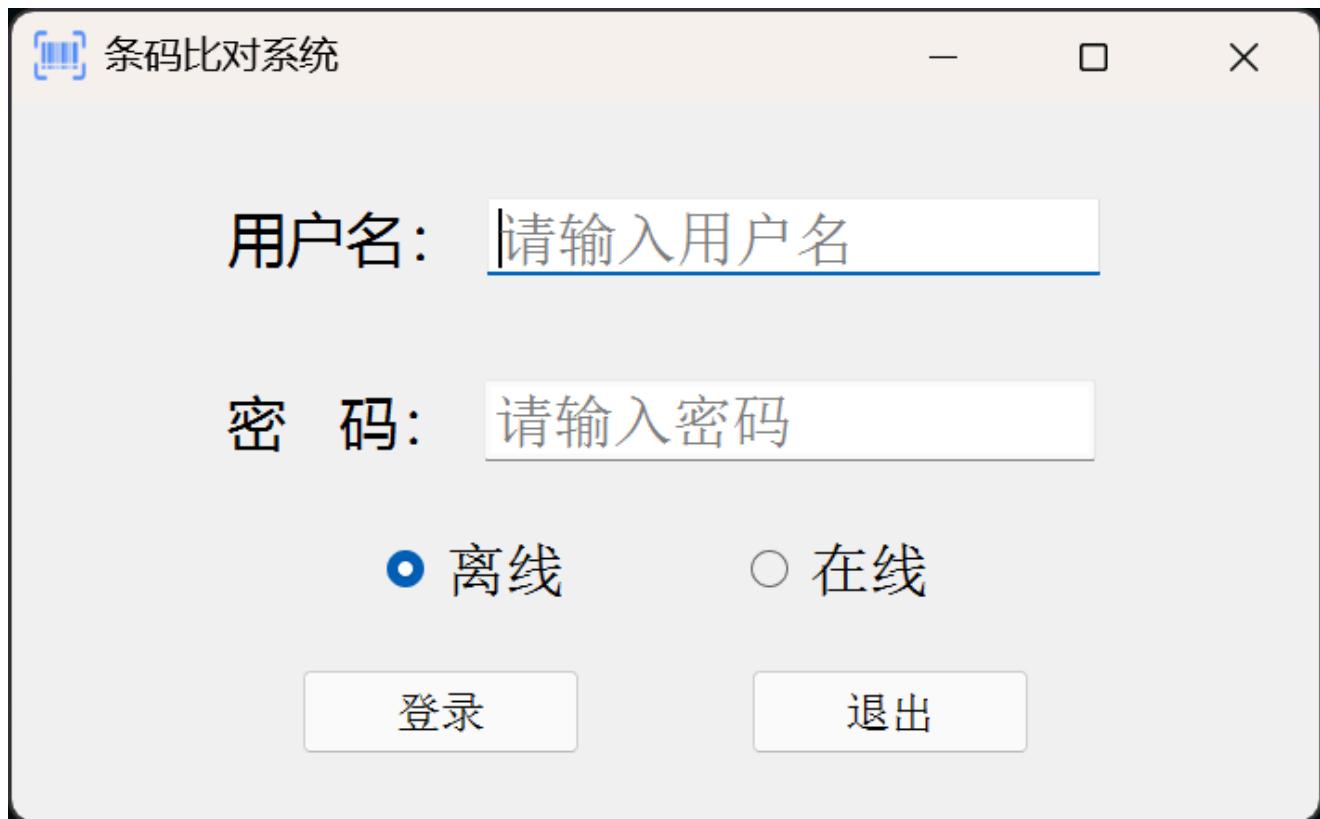


01 BarcodeCompare

1. 登录



- 登录模式分为在线模式和离线模式
 - 在线模式：使用MySQL数据库，需要在程序根目录的config.ini文件中配置数据库信息，目前程序数据库密码为明文。
 - 离线模式：使用SQLite3数据库，无需配置。
- 默认用户名及密码
 - 用户名: admin
 - 密码：iflogic2025

2. 用户管理

条码比对系统

用户 语言 配置

内盒比对 外箱比对 卡片比对 订单管理 用户管理

用户信息

	用户名	权限
1	admin	超级管理员
2	zel	管理员

用户配置

用户名:

密码:

权限:

添加 修改

删除 清空

2.1 权限

- 超级管理员：仅能创建用户，不可使用软件功能。
- 管理员：可以使用软件全部功能。（领班使用）。
- 普通用户：仅能使用内盒比对、外箱比对及卡片比对，不可以对订单配置进行操作。（操作员使用）

2.2 数据库表

users表位于barcode_compare数据库中

barcode_compare

main

表

formats

modes

orders

roles

users

视图

A-Z 索引

触发器

查询

备份

users表存放用户相关信息

对象 users @main (barcode_compare) - 表

表配置文件 开始事务 单元格编辑器 筛选 & 排序

id # integer	name TEXT	password TEXT	role_id # integer
1	admin	iflogic2025	1
2	zel	1	2

2.3 界面按钮功能

- 添加：users表中插入用户记录
- 修改：选中左侧用户列表的用户后，users表中修改用户记录
- 删除：选中左侧用户列表的用户后，users表中删除用户记录
- 清空：清空users表中所有记录

3. 订单管理

条码比对系统

用户 语言 配置

内盒比对 外箱比对 卡片比对 订单管理 用户管理

订单号	校验格式	条码模式	创建时间
1 IMO000118	卡片: 3 位 - 18 位 内盒: 3 位 - 18 位 外箱: 1 位 - 18 位	起始与结束条码	

订单信息

订单配置

订单号	1 位	—	19 位
卡片:	1 位	—	19 位
内盒:	1 位	—	19 位
外箱:	1 位	—	19 位
内盒标签文件:	<input type="button" value="浏览"/>		
外箱标签文件:	<input type="button" value="浏览"/>		
单卡标签文件:	可选项	<input type="button" value="浏览"/>	

添加 修改
删除 清空

3.1 校验格式

用于卡片ICCID(或序列号)与内盒或外箱ICCID(或序列号)位数不一致情况。

例如：

卡片ICCID为：89000000000000000001

内盒ICCID为：S833X89000000000000000001F

外箱ICCID为：S833X89000000000000000001F

则，校验格式：

卡片：1 - 19 位

内盒：6 - 24 位

外箱：6 - 24 位

3.2 标签文件

- 标签文件仅支持xlsx和csv文件格式，xls文件格式需要在WPS软件中转换一下。

标签文件支持的格式在barcode_compare数据库的formats表中。

Barcode Comparison Data									
对象	barcode_compare				formats				
	表配置文件	开始事务	单元格编辑器	筛选 & 排序	列	数据分析	导入	导出	
	# INTEGER	name # TEXT	type # INTEGER	filename # TEXT	box_number # TEXT	start_number # TEXT	end_number # TEXT	quantity # TEXT	barcode # TEXT
1	CQ01	1		盒号	J	K	L	二维码	
2	CQ01	2		箱号	J	K	L	二维码	
3	CQ02	1		盒号	J	K	L		
4	CQ02	2		箱号	J	K	L		
5	CQ02	3		数量	条形码ICCID	条形码IMSI	卡总数量		
6	CQ03	1		盒号	开始	结束	数量		
7	CQ03	2		箱号	开始	结束	数量		
8	Surabaya	1	No.WO/OF	Inner Box No.	First SN	Last SN	Quantity	Barcode	
9	Surabaya	2	No.WO/OF	Outer Box No.	First SN	Last SN	Quantity	Barcode	

字段名	值	功能
type	1:内盒， 2: 外箱	
filename	可为空	prd文件名，大部分订单不使用
box_number	对应xlsx内盒或外箱文件中关于盒号或箱号的列名	盒号或箱号
start_number	对应xlsx内盒或外箱文件中关于起始ICCID或起始序列号的列名	起始ICCID
end_number	对应xlsx内盒或外箱文件中与结束ICCID或起始序列号的列名	结束ICCID
quantity	对应xlsx内盒或外箱文件中关于数量的列名	每盒或每箱的数量
barcode	对应xlsx内盒或外箱文件中关于打印的条形码的列名	标签打印的条形码

- 目前支持印尼和重庆的三种格式，如需自定义增加格式，可以修改数据库，或修改代码中
src/database/format/sqlite/format_sqlite_dao.cpp

```
void FormatSqliteDao::init() {  
  
    // check if table exists  
    std::string      sql = "SELECT name FROM sqlite_master WHERE type = 'table'"
```

```

AND name = 'formats';

SQLite::Statement query(*db_, sql);
if (!query.executeStep()) {
    // create table if not exists
    sql = "CREATE TABLE IF NOT EXISTS formats (" +
        "id INTEGER PRIMARY key AUTOINCREMENT," +
        "name TEXT," +
        "type INTEGER," +
        "filename TEXT," +
        "box_number TEXT," +
        "start_number TEXT," +
        "end_number TEXT," +
        "quantity TEXT," +
        "barcode TEXT);";
    db_->exec(sql);

    // initialize format data
    std::shared_ptr<Format> cq01_box = std::make_shared<Format>(Format{0,
"CQ01", 1, "", "盒号", "J", "K", "L", "二维码"});
    std::shared_ptr<Format> cq01_carton = std::make_shared<Format>(Format{0,
"CQ01", 2, "", "箱号", "J", "K", "L", "二维码"});
    std::shared_ptr<Format> cq02_card = std::make_shared<Format>(Format{0,
"CQ02", 3, "", "数量", "条形码ICCID", "条形码IMSI", "卡总数量", "二维码"});
    std::shared_ptr<Format> cq03_box = std::make_shared<Format>(Format{0,
"CQ03", 1, "", "盒号", "开始", "结束", "数量", "二维码"});
    std::shared_ptr<Format> cq03_carton = std::make_shared<Format>(Format{0,
"CQ03", 2, "", "箱号", "开始", "结束", "数量", "二维码"});
    std::shared_ptr<Format> surabaya_box =
        std::make_shared<Format>(Format{0, "Surabaya", 1, "No.WO/OF", "Inner
Box No.", "First SN", "Last SN", "Quantity", "Barcode"});
    std::shared_ptr<Format> surabaya_carton =
        std::make_shared<Format>(Format{0, "Surabaya", 2, "No.WO/OF", "Outer
Box No.", "First SN", "Last SN", "Quantity", "Barcode"});

    add(cq01_box);
    add(cq01_carton);
    add(cq02_card);
    add(cq03_box);
    add(cq03_carton);
    add(surabaya_box);
    add(surabaya_carton);
}
}

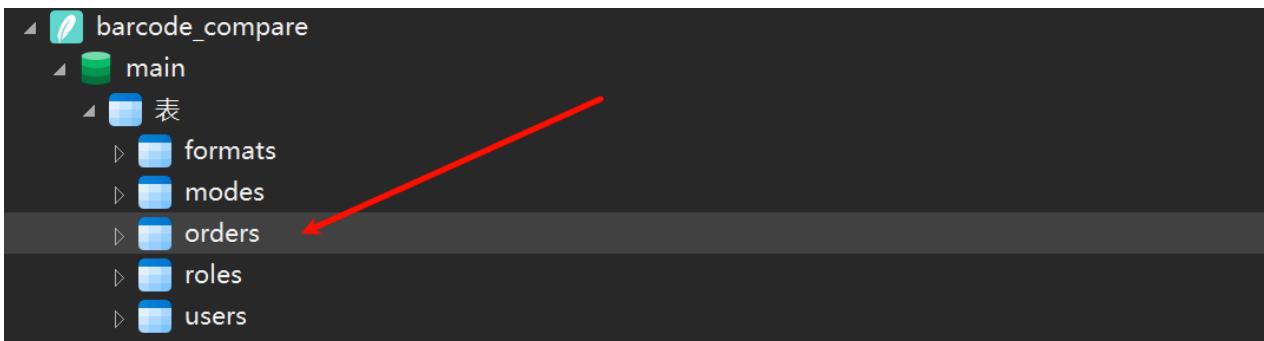
```

- 对于单卡标签数据，如果需要使用卡片比对功能，需要提供该文件，否则可以不提供直接创建订单

3.3 数据库表

内盒、外箱和卡片的数据在创建订单后会在box_data, carton_data, card_data数据库中创建对应订单号的数据表，并在barcode_compare主数据库的orders表中创建订单信息。

- 订单表



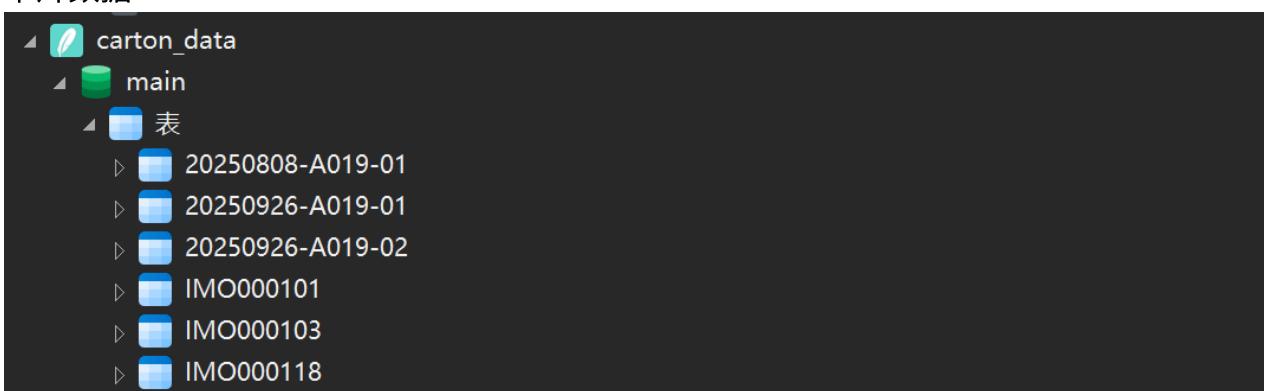
- 内盒数据



- 外箱数据



- 卡片数据



3.3 界面按钮功能

- 添加：orders表中插入订单记录，并导入数据到相应数据表中
- 修改：选中左侧订单列表的订单后，users表中修改订单记录
- 删除：选中左侧订单列表的订单后，users表中删除订单记录
- 清空：清空orders表中所有记录

4. 内盒比对

- 注：不可使用admin使用该功能。



4.1 使用步骤

- 选择订单，需先设置订单，导入标签数据。

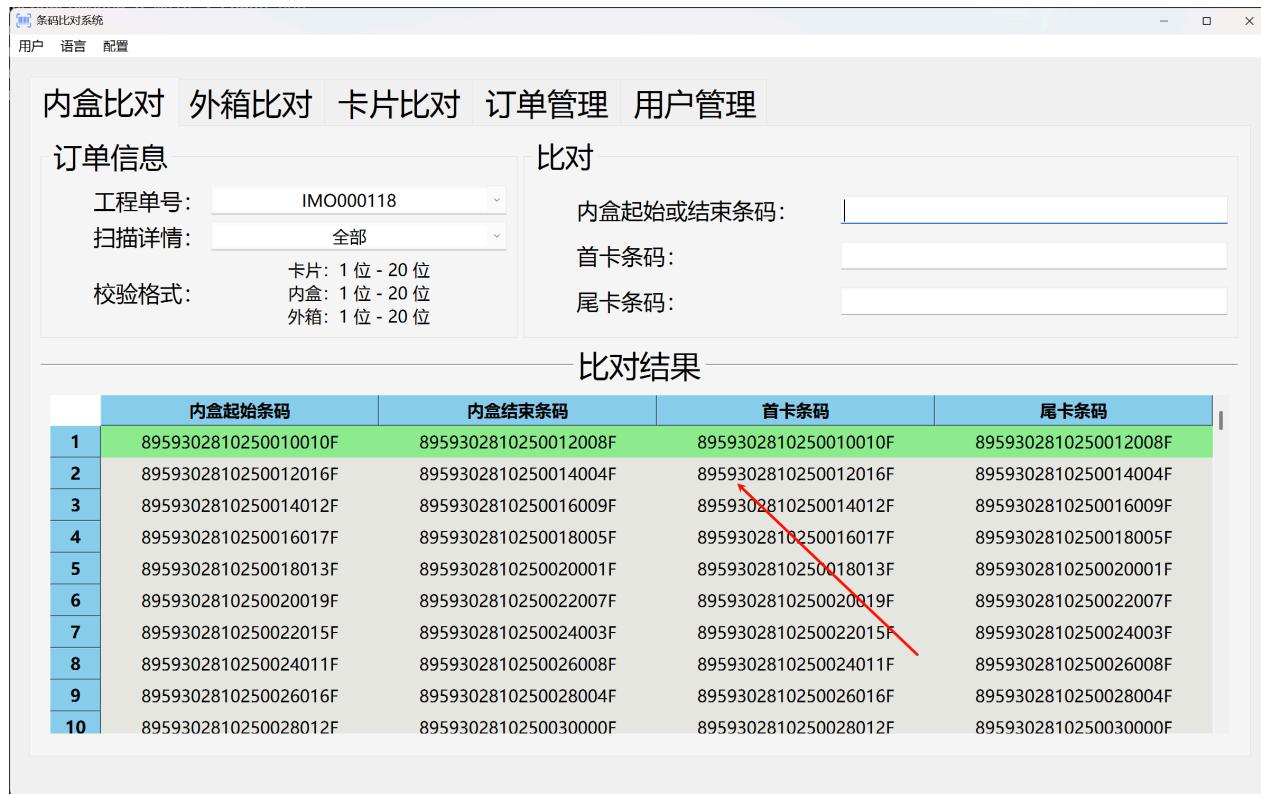


- 首先扫描内盒的起始或者结束条码，只需扫描其中之一即可
- 扫描首卡条码
- 再扫描尾卡条码

- 不匹配会弹窗提醒



- 成功后，该条数据会标记为绿色



5. 外箱比对



5.1 使用步骤

- 选择订单，需先设置订单，导入标签数据。



- 扫描外箱起始或结束条码，可以弹出对应该箱的内盒信息



- 其中0001为内盒号，黄色为待扫描（必须顺序扫描），成功扫描后变为绿色，当所有内盒扫描成功后，该条外箱数据变为绿色。

6. 卡片比对

用于OPP包装使用



6.1 使用步骤

- 选择订单，使用该功能需要提供单卡标签文件。



- 扫描内盒起始或结束条码后将弹窗内盒中都有卡片的信息



- 然后先将一个内盒的标签条码扫描完毕，这个过程中卡片会变为黄色，当内盒中所有标签条码扫完后，自动切换到卡片条码栏，再扫一遍卡片上的条码，当扫完两遍后，内盒数据变为绿色。