目录

1. 需求分析 1
1.1 可行性分析 1
1.2 功能需求 1
2. 系统设计说明书 1
2.1 系统开发环境与软件 1
2.2 系统概念结构设计 2
2.3 系统逻辑结构设计 3
2.4 实体属性表及信息 3
2.5 SQL 语句实现 4
2.6 系统结构功能图 8
3. 数据库设计与使用 9
3.1 用户登录与验证 9
3.2 商品管理 9
3.3 商品添加与删除 9
3.4 订单管理 9
3.5 订单更新 9
4. 系统测试 10
4.1 GUI 界面设计 10
4.2 登录与验证 10
4.3 商品查询与订单查询 12
4.4 商品增删 13
4.5 订单修改 16
4.6 退出系统 18
4.7 版本②GUI 18
5. 附录（代码） 27
5.1 数据库板块 27
5.2 GUI 板块 28
5.3 系统功能板块 29

For reference only
Copyright chimeC All rights reserved

1.需求分析

1.1 可行性分析

1.1.1 经济可行性

网上商城系统的开发将使用 MySQL 数据库作为数据存储解决方案，MySQL 是一个开源的关系型数据库管理系统，具有免费、稳定、高效等特点。因此，从经济角度来看，采用 MySQL 作为数据库系统是可行的，无需额外购买昂贵的数据库软件。

1.1.2 管理可行性

网上商城系统的管理只需通过计算机和互联网即可实现。管理员可以通过 Web 界面进行商品管理、订单处理、用户管理等操作，无需复杂的硬件或软件环境配置。同时，由于系统基于 Web 开发，用户可以通过浏览器随时随地进行访问和操作，方便快捷。因此，从管理角度来看，系统的开发也是可行的。

1.2 功能需求

（1）用户登录及验证：在登录界面中，用户输入用户名和密码后,系统通过查询数据库验证是否存在该用户，如果验证成功就显示商品,管理界面，否则提示“无效的账号或密码”，并返回登录界面进行注册

（2）商品管理：用户登录系统后，进入商品管理界面，用户可以查看所有商品，系统会查询数据库显示商品记录。

（3）删除商品：在商品管理界面，用户选择“删除商品”后，系统会从数据库中删除此条商品记录，并提示删除成功，返回到商品查询界面进行更新即可看到删除商品后的全部商品信息。

（4）添加商品：用户登录系统后，可以选择“添加商品”，进入增加商品界面，输入商品的基本信息、品牌、价格等基本信息之后，系统会向数据库新增一条商品记录。可以回到查询页面进行查看。

（5）订单管理：用户登录系统后，可以选择“订单管理”，进入订单管理界面，查看所有订单，系统会查询数据库显示订单记录。

（6）修改订单状态：在订单管理界面，用户选择“修改状态”后,进入订单状态修改界面，用户选择订单状态，进行提交，系统会更新数据库中该条记录的订单状态。

For reference only
Copyright chinaxiv

2. 系统设计说明书

2.1 系统开发环境与软件

本系统基于 Python 和 C++版本分别进行开发，主要使用到了 Vscode、visual studio2022、Navicat、SQL server 等软件。

2.2 系统概念结构设计

2.2.1 局部 ER 图

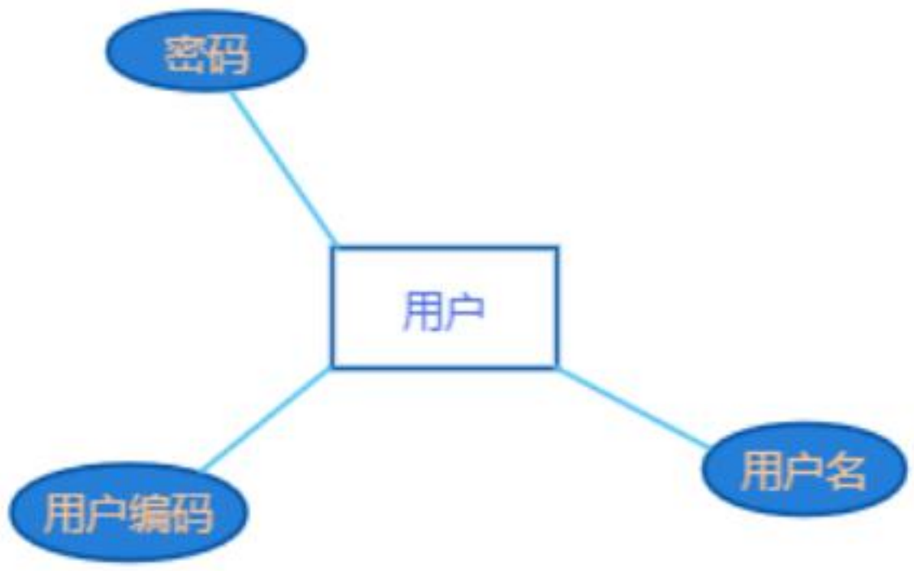


图 2.2.1 用户实体 ER 图

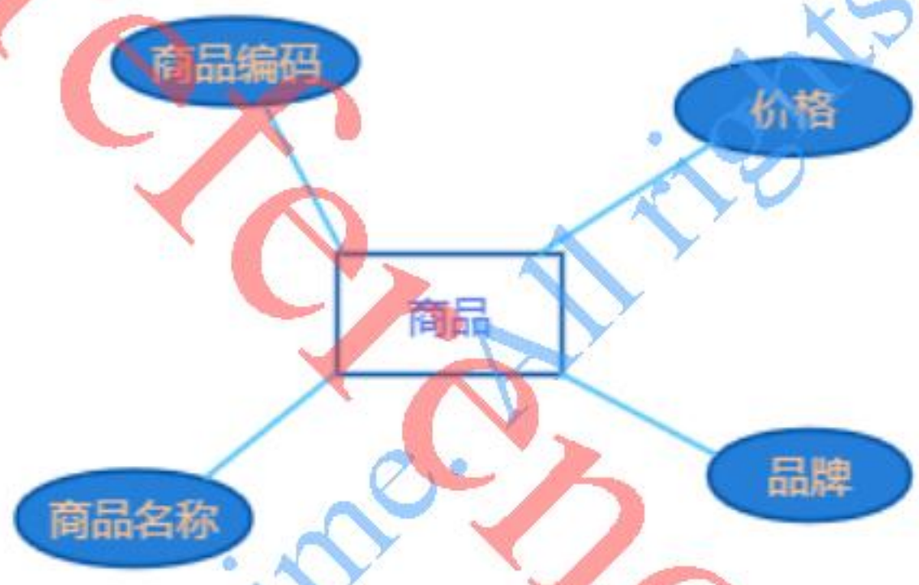


图 2.2.2 商品实体 ER 图

图 2.2.3 订单实体 ER 图

For reference only
Copyright chimeras rights reserved

2.2.2 整体 ER 图

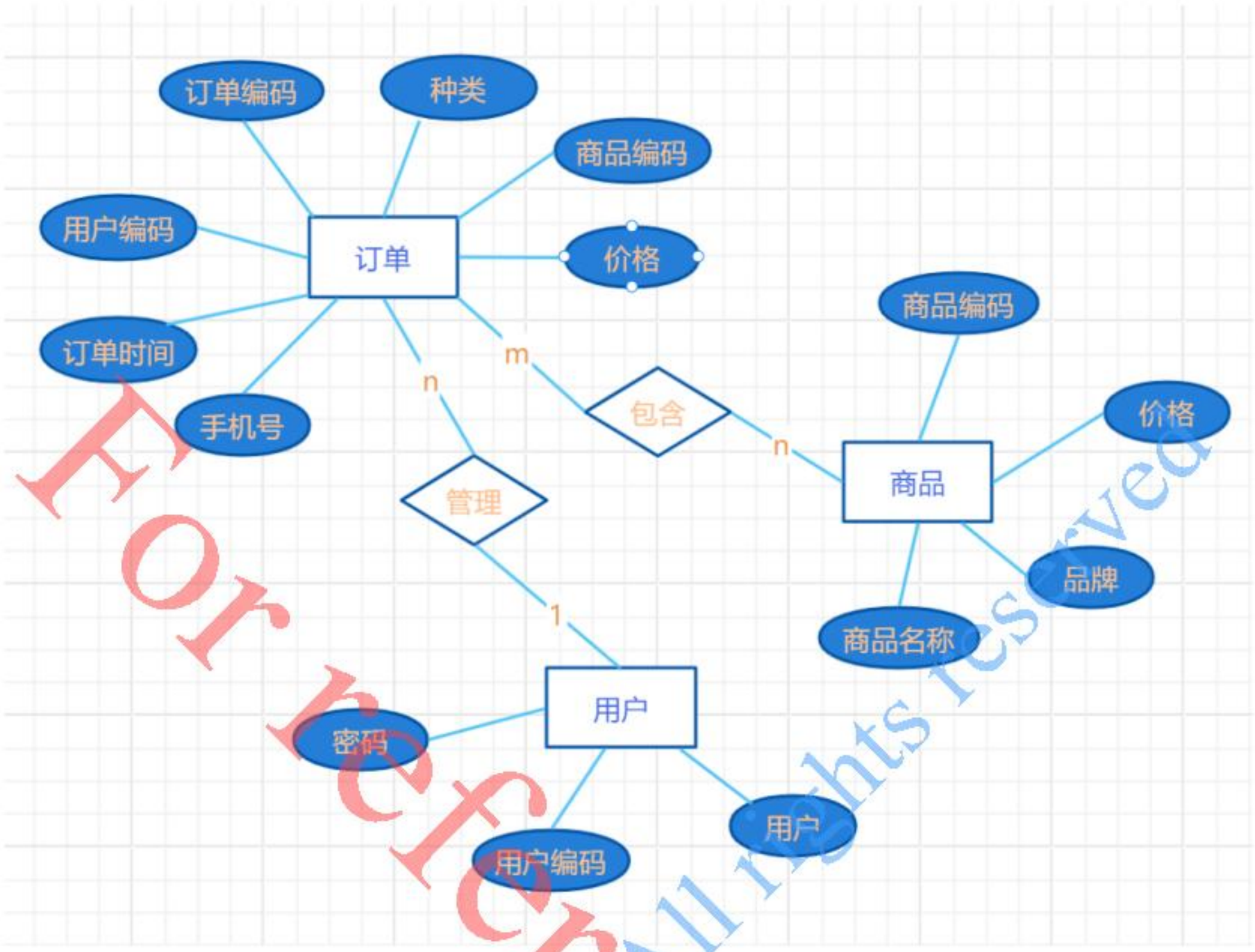


图 2.2.4 系统整体 ER 图

2.3 系统逻辑结构设计

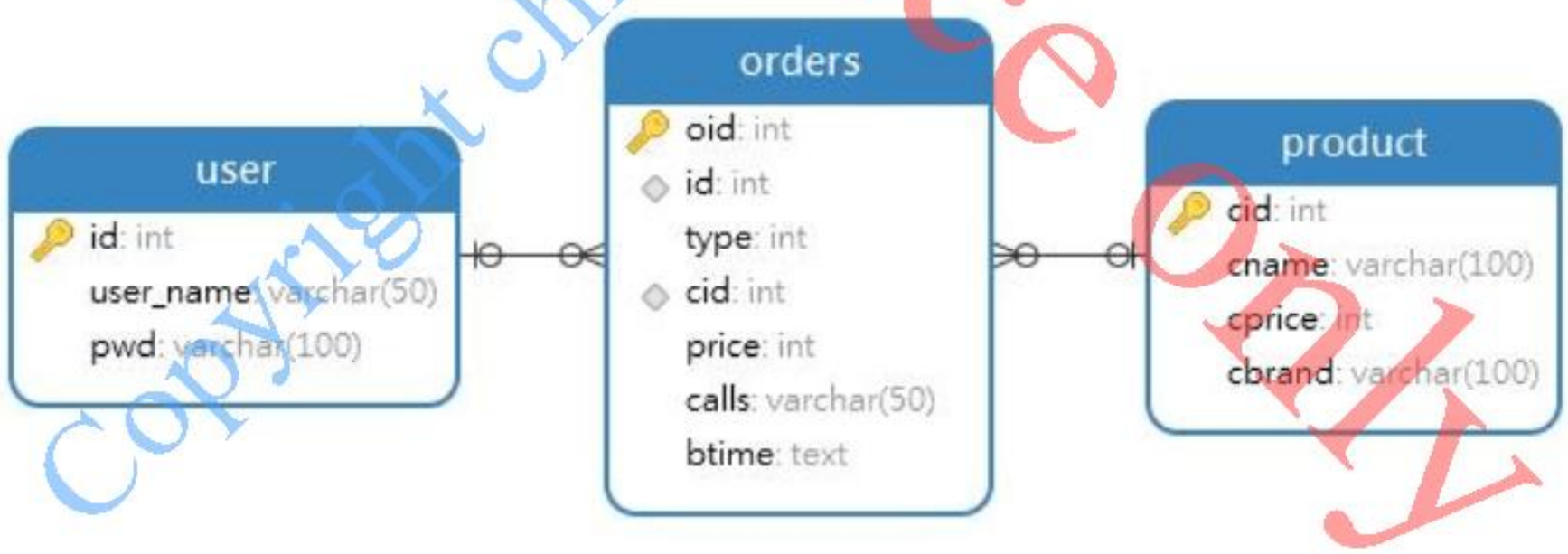


图 2.3.1 系统逻辑结构图

2.4 实体属性表及信息

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			✓		🔑 1	用户编码
user_name	varchar	50		✓			用户名
pwd	varchar	100		✓			密码

图 2.4.1 用户表 User

For reference only
Copyright chimecence All rights reserved

名	类型	长度	小数点	不是 null	虚拟	键	注释
cid	int			✓		🔑1	商品编码
cname	varchar	100					商品名称
cprice	int						价格
cbrand	varchar	100					品牌

图 2.4.2 商品表 product

名	类型	长度	小数点	不是 null	虚拟	键	注释
oid	int			✓		🔑1	订单编码
id	int						用户编码,外键
type	int						种类
cid	int						商品编码,外键
price	int						价钱
calls	varchar	50					电话
btime	text						下单时间

图 2.4.3 订单表 Orders

2.5 SQL 语句实现

数据库的实现部分包括三个板块：数据库、表的创建；数据的插入；数据库的查询、更新部分。

2.5.1 数据库、表的创建：

```
-- 创建网上商城数据库
create database if not exists shopping;
-- 创建表
-- 创建USER 表
create table if not exists user(
 id INT(10) NOT NULL,
 user_name VARCHAR(50) NOT NULL,
 pwd VARCHAR(100) NOT NULL,
 PRIMARY KEY(id)
);
-- 创建 product 表
CREATE TABLE IF NOT EXISTS product (
    cid INTEGER PRIMARY KEY,
    cname VARCHAR(100),
    cprice INTEGER,
    cbrand VARCHAR(100)
);
-- 创建 orders 表，设置外键约束
CREATE TABLE IF NOT EXISTS orders (
    oid INTEGER PRIMARY KEY,
    id INTEGER,
```

For reference only
Copyright chimeCe All rights reserved

```
    type INTEGER,
    cid INTEGER,
    price INTEGER,
    calls VARCHAR(50),
    btime TEXT,
    FOREIGN KEY (cid) REFERENCES product(cid) ON DELETE CASCADE,
    FOREIGN KEY (id) REFERENCES user(id) ON DELETE CASCADE
);
```

2.5.2 数据的插入

这里只演示插入部分数据（各 10 条）

```
-- 插入数据到 user 表
INSERT INTO user (id,user_name,pwd) VALUES
(157,1,1),
(117,2,2),
(153,3,3),
(194,4,4),
(121,5,5),
(132,6,6),
(108,7,7),
(116,9,9),
(138,10,10);
-- 插入数据到 product 表
INSERT INTO product (cid, cname, cprice, cbrand) VALUES
(1001, '华为手机', 1599, '华为'),
(1002, '华为笔记本', 8699, '华为'),
(1003, '智能音箱', 699, '小米'),
(1004, '三星单反相机', 2199, '三星'),
(1005, '小米智能电视机', 1999, '小米'),
(1006, '苹果手机', 5899, 'Apple'),
(1007, '联想笔记本', 4999, '联想'),
(1008, '索尼耳机', 699, '索尼'),
(1009, '华硕显示器', 1799, '华硕'),
(1010, '戴尔台式电脑', 3999, '戴尔');
-- 插入数据到 order 表
INSERT INTO orders (oid, id, type, cid, price, calls, btime) VALUES
(1, 157, 1, 1003, 933, '831970', '2024-04-28 22:25:59'),
(2, 117, 1, 1002, 558, '372575', '2023-11-29 22:25:59'),
(3, 153, 1, 1002, 677, '805932', '2023-10-02 22:25:59'),
(4, 194, 1, 1004, 361, '049694', '2023-11-13 22:25:59'),
(5, 121, 1, 1003, 249, '959695', '2023-09-26 22:25:59'),
(6, 132, 1, 1003, 318, '845742', '2024-04-29 22:25:59'),
```

For reference only
Copyright chinese All rights reserved

```
(7, 108, 1, 1004, 295, '251196', '2024-01-19 22:25:59'),
(8, 194, 1, 1002, 571, '697854', '2024-06-04 22:25:59'),
(9, 116, 1, 1003, 220, '026070', '2023-09-01 22:25:59'),
(10, 138, 1, 1001, 172, '913627', '2024-03-14 22:25:59');
```

id	user_name	pwd
108	7	7
116	9	9
117	2	2
121	5	5
132	6	6
138	10	10
153	3	3
157	1	1
194	4	4

图 2.5.1 创建 User 表

cid	cname	cprice	cbrand
1001	华为手机	1599	华为
1002	华为笔记本	8699	华为
1003	智能音箱	699	小米
1004	三星单反相机	2199	三星
1005	小米智能电视机	1999	小米
1006	苹果手机	5899	Apple
1007	联想笔记本	4999	联想
1008	索尼耳机	699	索尼
1009	华硕显示器	1799	华硕
1010	戴尔台式电脑	3999	戴尔

图 2.5.2 创建 product 表

For reference only
Copyright.chinecan rights reserved

oid	id	type	cid	price	calls	btime
1	157	1	1003	933	831970	2024-04-28 22:25:59
2	117	1	1002	558	372575	2023-11-29 22:25:59
3	153	1	1002	677	805932	2023-10-02 22:25:59
4	194	1	1004	361	049694	2023-11-13 22:25:59
5	121	1	1003	249	959695	2023-09-26 22:25:59
6	132	1	1003	318	845742	2024-04-29 22:25:59
7	108	1	1004	295	251196	2024-01-19 22:25:59
8	194	1	1002	571	697854	2024-06-04 22:25:59
9	116	1	1003	220	026070	2023-09-01 22:25:59
10	138	1	1001	172	913627	2024-03-14 22:25:59

图 2.5.3 创建 Order 表

2.5.3 数据库的查询、更新与删除

```sql
-- 查询商品表
SELECT * from product WHERE cbrand = '华为';
-- 更新订单数据
UPDATE orders
set calls = '123456'
WHERE oid = 4;
select * from orders;
-- 删除商品信息
DELETE FROM product
WHERE cid = 1009;
```

cid	cname	cprice	cbrand
1001	华为手机	1599	华为
1002	华为笔记本	8699	华为

图 2.5.4 查询商品信息

oid	id	type	cid	price	calls	btime
1	157	1	1003	933	831970	2024-04-28 22:25:59
2	117	1	1002	558	372575	2023-11-29 22:25:59
3	153	1	1002	677	805932	2023-10-02 22:25:59
4	194	1	1004	361	123456	2023-11-13 22:25:59
5	121	1	1003	249	959695	2023-09-26 22:25:59
6	132	1	1003	318	845742	2024-04-29 22:25:59
7	108	1	1004	295	251196	2024-01-19 22:25:59
8	194	1	1002	571	697854	2024-06-04 22:25:59
9	116	1	1003	220	026070	2023-09-01 22:25:59
10	138	1	1001	172	913627	2024-03-14 22:25:59

图 2.5.5 更新订单信息（订单编码为 4 的手机号已被更新）

For reference only
Copyright chinese All rights reserved

cid	cname	cprice	cbrand
1001	华为手机	1599	华为
1002	华为笔记本	8699	华为
1003	智能音箱	699	小米
1004	三星单反相机	2199	三星
1005	小米智能电视机	1999	小米
1006	苹果手机	5899	Apple
1007	联想笔记本	4999	联想
1008	索尼耳机	699	索尼
1010	戴尔台式电脑	3999	戴尔

图 2.5.6 商品信息删除（商品编码为 1009 已被删除）

2.6 系统结构功能图

图 2.6.1 系统结构图

For reference only
Copyright Chimes All rights reserved

3.数据库设计与使用

3.1 用户登录与验证

结合 1.2 小结，我们需要实现用户的登录、注册与验证功能，而这一部分内容将嵌入到我们的 GUI 程序中。

3.2 商品管理

商品管理主要是进行商品的查询操作，因此需要将数据库的查询模块嵌入到 GUI 程序中。当我们点击商品管理板块时，可以得到所有商品的信息，为了后续商品信息的方便以及商品信息过多的情况，我们添加了一个数据更新按钮在窗口最底部以避免死机。

3.3 商品添加与删除

商品添加与删除是一个独立的板块，需要将数据库的添加与删除内容嵌入GUI程序中，由于这里我们有着外接约束，而且在创建表的时候就设计了级联删除与更新，因此当输入任意存在的商品信息时，如果订单中也存在该商品，那么订单中的这条记录与商品中的这条记录都会被删除。当我们添加商品时需要输入全部的商品信息，否则会提示重新输入。

3.4 订单管理

在这个板块中，其基本内容与商品管理板块一致。

3.5 订单更新

在订单更新板块，我们设置了多个空白框可以填写，但是填写的信息至少一个，否则不能进行订单的信息更新。其次需要注意的是，在订单表中，我们预设了两个外键约束，即用户 id，与商品 cid，当我们在更新数据时需要确保更新的这两个信息必须存在于用户表与商品表中，否则会提示重新输入。

For reference only
Copyright chinaeast all rights reserved

4.系统测试

4.1 GUI 界面设计

图 4.1 ①界面

注：这里的功能说明将采用功能较为完整的版本（队长所作）进行，在第 4 部分末尾将给出②版本（组员 1 所作）的示意图。

4.2 登录与验证

我们在 user 表中预设了一位用户名为‘123456’的用户，如果继续执行注册这个账号（图 4.2.1）的话会出现图 4.2.2 的提示信息。若逆序注册会成功，如图 4.2.3 所示。

图 4.2.1 注册

copyright chinese only
All rights reserved

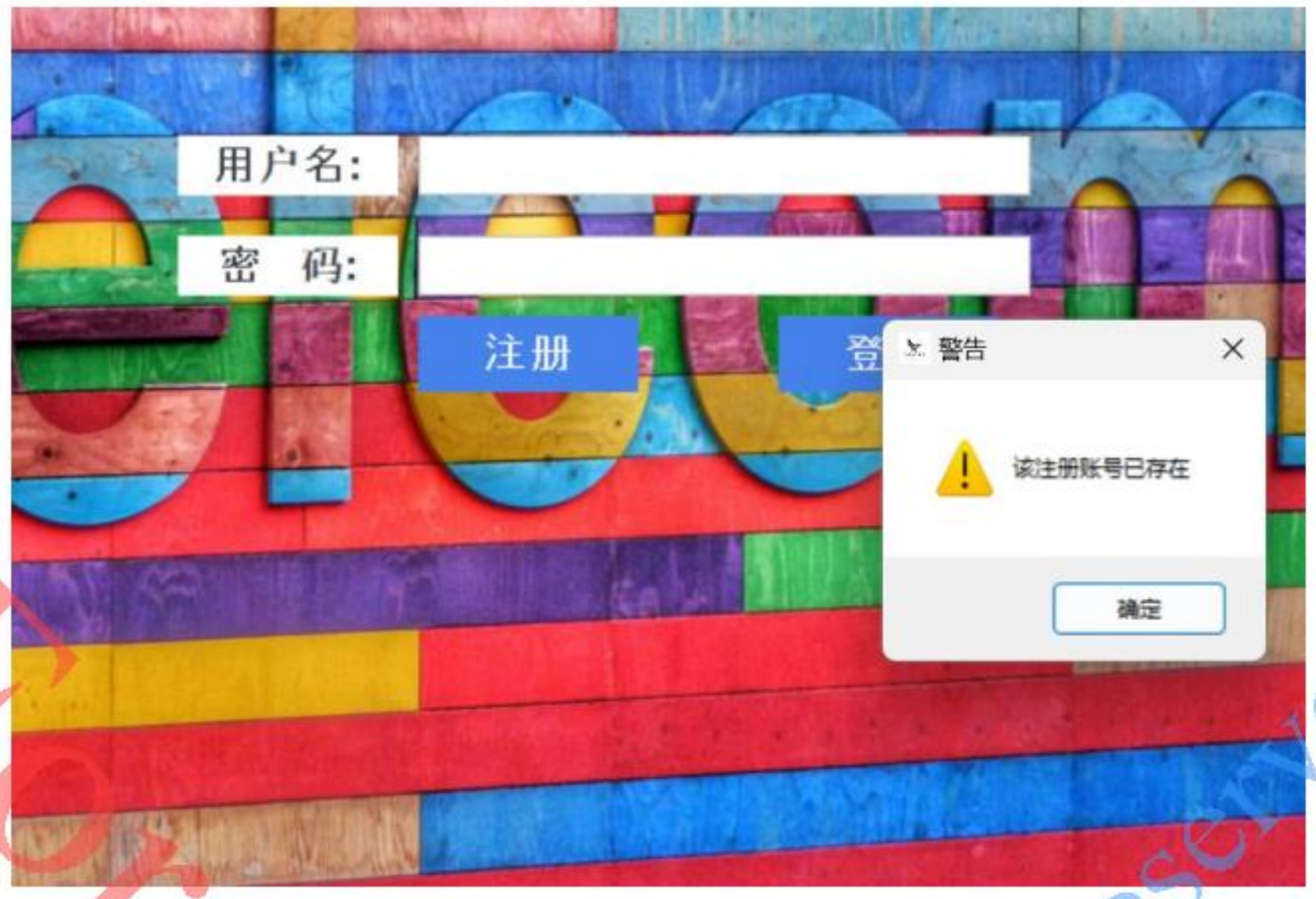


图 4.2.2 注册失败

图 4.2.3 注册成功

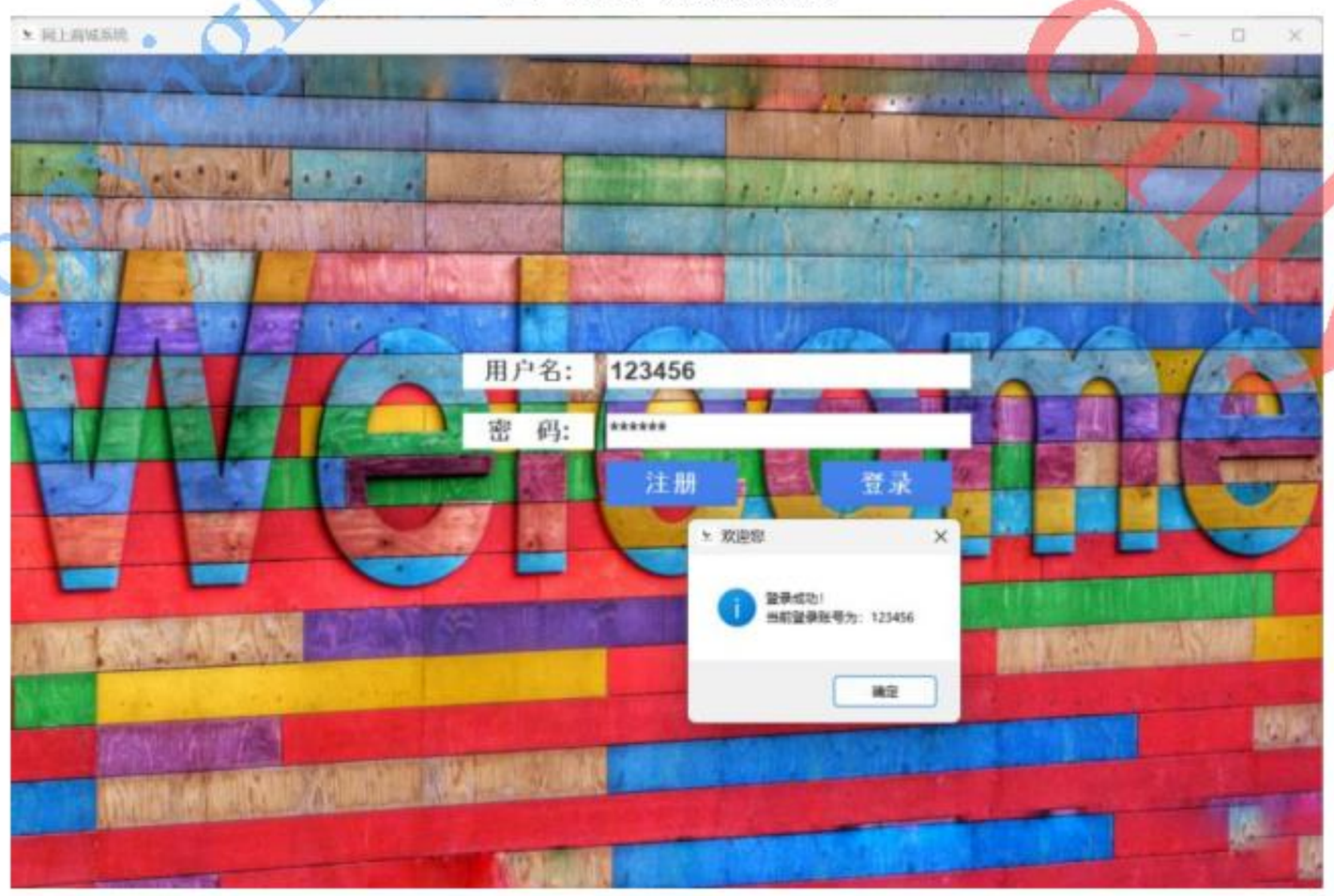


图 4.2.4 登录界面

For reference only
Copyright chinese all rights reserved

当我们点击确定后会自动进行操作界面如图 4.2.5 所示，而登录界面会自动关闭。

商品编码	商品名称	商品价格	商品品牌
1001	iphone15pro	20000	苹果
1003	智能音箱	699	小米
1004	三星单反相机	2199	三星
1005	小米智能电视机	1999	小米
1006	苹果手机	5899	Apple
1007	联想笔记本	4999	联想
1008	索尼耳机	699	索尼
1009	华硕显示器	1799	华硕
1010	戴尔台式电脑	3999	戴尔
1011	飞利浦电吹风	199	飞利浦
1012	惠普打印机	799	惠普
1013	海尔冰箱	2999	海尔
1014	夏普空调	3599	夏普
1015	美的洗衣机	2399	美的
1016	华为平板电脑	2999	华为
1017	小米智能手表	899	小米
1018	苹果笔记本	8999	Apple
1019	索尼电视	4399	索尼
1020	戴尔游戏笔记本	6999	戴尔
1021	飞利浦剃须刀	299	飞利浦
1022	惠普一体机	2999	惠普
1023	海尔洗碗机	3499	海尔
1024	夏普电视	2999	夏普
1025	美的空气净化器	1499	美的
1026	华为路由器	199	华为
1027	小米电动滑板车	1999	小米
1028	苹果智能手表	2599	Apple

图 4.2.5 主界面（操作界面）

4.3 商品查询与订单查询

如图 4.2.5 和图 4.3.1 所示分别为商品查询界面和订单查询界面，我们已经提前往数据库中预设了约 100 多条数据方便后续的使用。（当我们滑动鼠标时数据是可以动的）

订单编码	用户ID	商品种类	商品编码	价格	电话号码	订单时间
1	13	2	1004	333	777777	2024-11-11 11:11
4	194	1	1004	361	049694	2023-11-13 22:25:59
5	121	1	1003	249	959695	2023-09-26 22:25:59
6	132	1	1003	318	845742	2024-04-29 22:25:59
7	108	1	1004	295	251196	2024-01-19 22:25:59
9	116	1	1003	220	026070	2023-09-01 22:25:59
15	178	1	1003	62	050787	2024-01-16 22:25:59
16	157	1	1003	252	741854	2024-03-02 22:25:59
21	171	1	1003	843	618367	2023-12-09 22:25:59
23	125	1	1003	240	185249	2023-11-16 22:25:59
27	166	1	1004	893	630323	2024-01-29 22:25:59
28	187	1	1004	797	408698	2024-02-25 22:25:59
29	115	1	1003	572	669152	2023-07-20 22:25:59
30	149	1	1004	179	785190	2024-03-10 22:25:59
31	108	1	1004	538	267640	2023-12-13 22:25:59
32	186	1	1003	274	487505	2024-04-05 22:25:59
33	155	1	1004	80	143624	2023-08-22 22:25:59
35	193	1	1004	997	535498	2023-11-02 22:25:59
38	164	1	1004	256	433189	2023-11-04 22:25:59
41	189	1	1003	961	448339	2024-01-16 22:25:59
42	190	1	1003	632	151200	2024-03-30 22:25:59
43	194	1	1003	923	235308	2023-12-03 22:25:59
45	164	1	1003	524	338205	2024-01-25 22:25:59
46	111	1	1004	785	379987	2024-03-16 22:25:59
48	126	1	1004	756	253361	2024-02-09 22:25:59
49	113	1	1004	541	519529	2024-04-16 22:25:59
50	153	1	1004	178	146178	2024-01-24 22:25:59

图 4.3.1 订单查询界面

For reference only
Copyright chinese All rights reserved

4.4 商品增删

图 4.4.1 商品增删界面

比如我们想要删除商品编码为 1004 的商品，如图 4.4.2 进行操作后点击确定，可以删除，如图 4.4.3 中，原图 4.2.5 中商品编号为 1004 的商品在更新后已经消失。再进入订单查询页面（图 4.4.4）可以看到原图 4.3.1 中的商品编号为 1004 的订单在点击更新后也已经消失。

For reference only
Copyright chinese all rights reserved

图 4.4.2 删除示例

商品查询 | 商品增删 | 订单查询 | 订单修改

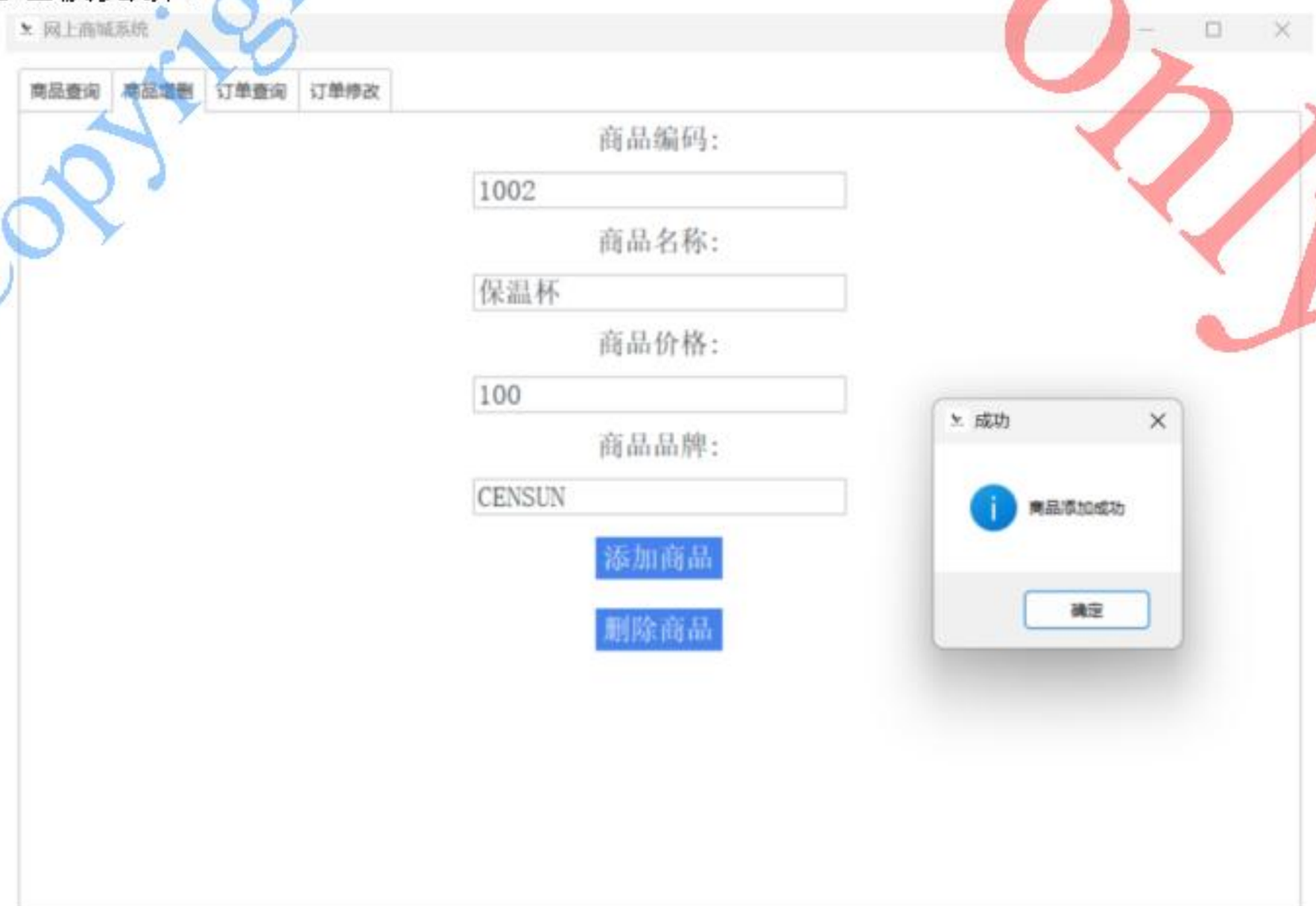
商品编码	商品名称	商品价格	商品品牌
1001	iphone15pro	20000	苹果
1003	智能音箱	699	小米
1005	小米智能电视机	1999	小米
1006	苹果手机	5899	Apple
1007	联想笔记本	4999	联想
1008	索尼耳机	699	索尼
1009	华硕显示器	1799	华硕
1010	戴尔台式电脑	3999	戴尔
1011	飞利浦电吹风	199	飞利浦
1012	惠普打印机	799	惠普
1013	海尔冰箱	2999	海尔
1014	夏普空调	3599	夏普
1015	美的洗衣机	2399	美的
1016	华为平板电脑	2999	华为
1017	小米智能手表	899	小米
1018	苹果笔记本	8999	Apple
1019	索尼电视	4399	索尼
1020	戴尔游戏笔记本	6999	戴尔
1021	飞利浦剃须刀	299	飞利浦
1022	惠普一体机	2999	惠普
1023	海尔洗碗机	3499	海尔

图 4.4.3 删除结果 1

订单编码	用户ID	商品种类	商品编码	价格	电话号码	订单时间
5	121	1	1003	249	959695	2023-09-26 22:25:59
6	132	1	1003	318	845742	2024-04-29 22:25:59
9	116	1	1003	220	026070	2023-09-01 22:25:59
15	178	1	1003	62	050787	2024-01-16 22:25:59
16	157	1	1003	252	741854	2024-03-02 22:25:59
21	171	1	1003	843	618367	2023-12-09 22:25:59
23	125	1	1003	240	185249	2023-11-16 22:25:59
29	115	1	1003	572	669152	2023-07-20 22:25:59
32	186	1	1003	274	487505	2024-04-05 22:25:59
41	189	1	1003	961	448339	2024-01-16 22:25:59
42	190	1	1003	632	151200	2024-03-30 22:25:59
43	194	1	1003	923	235308	2023-12-03 22:25:59
45	164	1	1003	524	338205	2024-01-25 22:25:59
51	165	1	1005	999	123456	2024-06-20 22:25:59
52	166	1	1006	5899	654321	2024-06-21 22:25:59
53	167	1	1007	4999	987654	2024-06-21 22:25:59
54	168	1	1008	699	456789	2024-06-21 22:25:59

图 4.4.4 删除结果 2

商品添加也可以高效的进行，如图 4.4.5 所示，可以查询商品信息（图 4.4.6），保温杯的信息已经被更新。



Copyright chinecall rights reserved
For reference only

图 4.4.5 商品添加

图 4.4.6 商品添加结果

但是如果只填写了任一个信息如图 4.4.7 所示，那么会报出警告信息。

图 4.4.7 错误填写数据

For reference only
Copyright chimence all rights reserved

4.5 订单修改

图 4.5.1 订单修改主页面

下面我们更新订单编码为 21 的订单数据（图 4.5.2），再次进入订单查询界面可以看到更新的订单内容（图 4.5.3）。

图 4.5.2 订单更新

For reference only
Copyright chimes All rights reserved

商品查询 商品增删 订单查询 订单修改

订单编码	用户ID	商品种类	商品编码	价格	电话号码	订单时间
5	121	1	1003	249	959695	2023-09-26 22:25:59
6	132	1	1003	318	845742	2024-04-29 22:25:59
9	116	1	1003	220	026070	2023-09-01 22:25:59
15	178	1	1003	62	050787	2024-01-16 22:25:59
16	157	1	1003	252	741854	2024-03-02 22:25:59
21	123	3	1002	100	100100	2024-06-28 21:13:00
23	125	1	1003	240	185249	2023-11-16 22:25:59
29	115	1	1003	572	669152	2023-07-20 22:25:59
32	186	1	1003	274	487505	2024-04-05 22:25:59
41	189	1	1003	961	448339	2024-01-16 22:25:59
42	190	1	1003	632	151200	2024-03-30 22:25:59
43	194	1	1003	923	235308	2023-12-03 22:25:59
45	164	1	1003	524	338205	2024-01-25 22:25:59
51	165	1	1005	999	123456	2024-06-20 22:25:59
52	166	1	1006	5899	654321	2024-06-21 22:25:59
53	167	1	1007	4999	987654	2024-06-21 22:25:59
54	168	1	1008	699	456789	2024-06-21 22:25:59
55	169	1	1009	1799	987123	2024-06-21 22:25:59
56	170	1	1010	3999	321654	2024-06-21 22:25:59
57	171	1	1011	199	159753	2024-06-21 22:25:59
58	172	1	1012	799	654987	2024-06-21 22:25:59

图 4.5.3 更新的订单数据

值得注意的是，所有字段都是必填项，否则会产生图 4.5.4 的错误信息。

图 4.5.4 订单更新错误

For reference only
Copyright chimeras all rights reserved

4.6 退出系统

图 4.6.1 系统关闭

图 4.6.1 所示，当点击“是”按钮后将自动关闭系统。

4.7 版本②GUI

现对版本②的优点做一个说明：由于采用 C++的 MFC 框架,这个框架相较于①版本 python 的 tkinter 框架更为成熟和稳定，对于大型应用也较为友好，因此开发难度较大。它的布局界面较 python 更为美观，且页面简单，功能也可以多样，如果开发内容开发完善的话（这里只实现了商品管理的功能）是一个优秀的系统。

For reference only
Copyright claimed. All rights reserved

图 4.7.1 注册登录界面

图 4.7.2 登录警告界面

For reference only
Copyright chimes all rights reserved

图 4.7.3 登录成功

图 4.7.4 功能主界面

For reference only
Copyright chimes all rights reserved

图 4.7.5 商品管理界面

商品序号	商品名称	商品种类	商品价格(美元)
1	NVIDIA RTX 4090 Ti	显卡	1999.00
2	NVIDIA RTX 4090	显卡	1599.00
3	NVIDIA RTX 4080 Ti	显卡	1399.00
4	Lenovo Legion Y7000 2024	笔记本电脑	1005.34
5	Lenovo YOGA Pro 16s AI元启	笔记本电脑	2479.49
6	vivo S19	手机	344.04
7	OPPO Find X7 Ultra	手机	826.58
8	Redmi K70	手机	356.45
9	Ronshen BCD-501WD18FP	冰箱	943.38
10	Sony FE 24-70mm f/2.8 GM II	镜头	1824.28
11	NVIDIA RTX 4080	显卡	1199.00

图 4.7.6 导入商品数据

For reference only
Copyright chimeance. All rights reserved

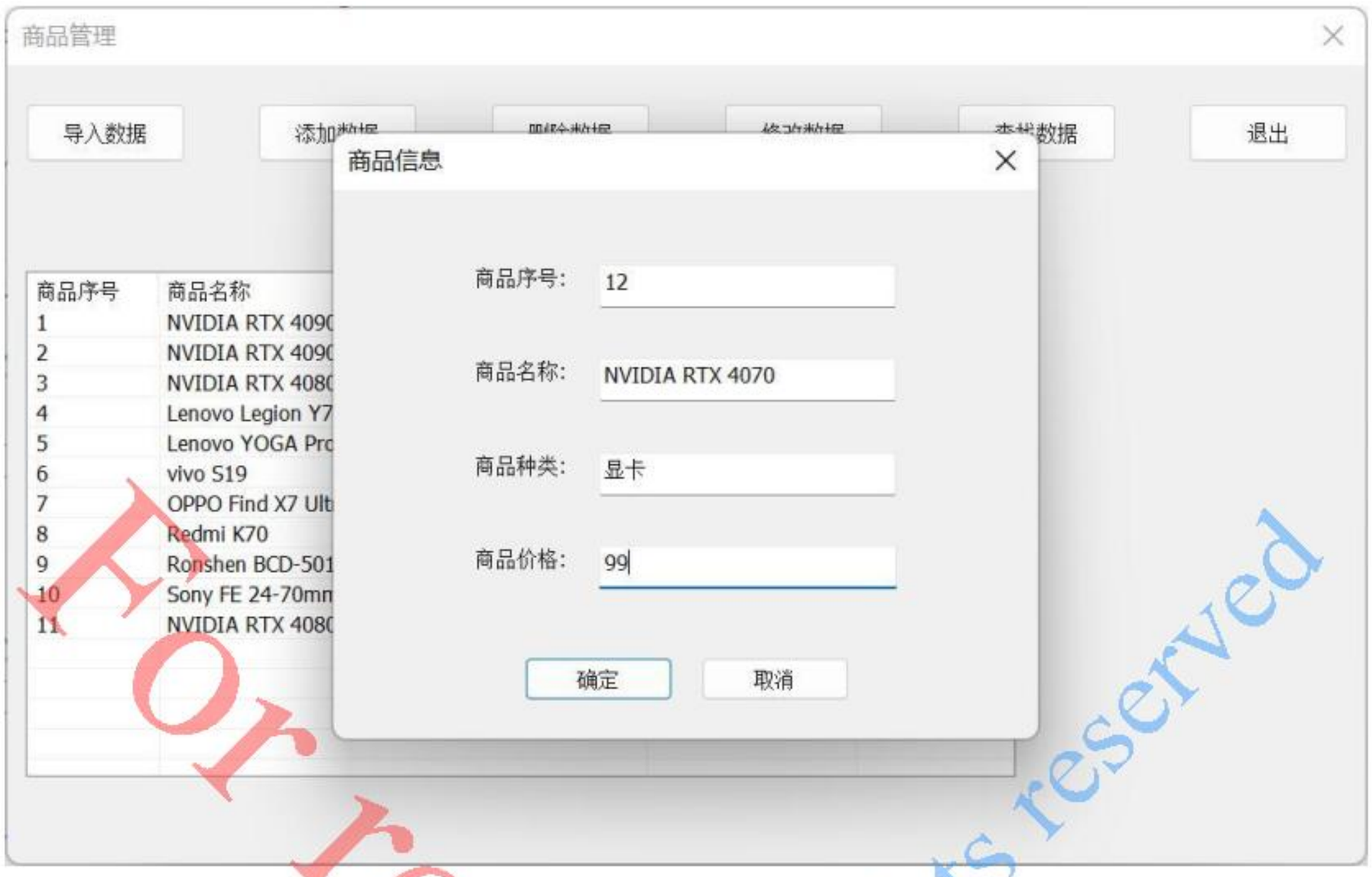


图 4.7.7 增加商品数据

图 4.7.8 增加成功

For reference only
Copyright chimeng All rights reserved

图 4.7.9 商品删除（选中商品即可）

图 4.7.10 商品删除提示

For reference only
Copyright chinese All rights reserved

图 4.7.11 删除成功

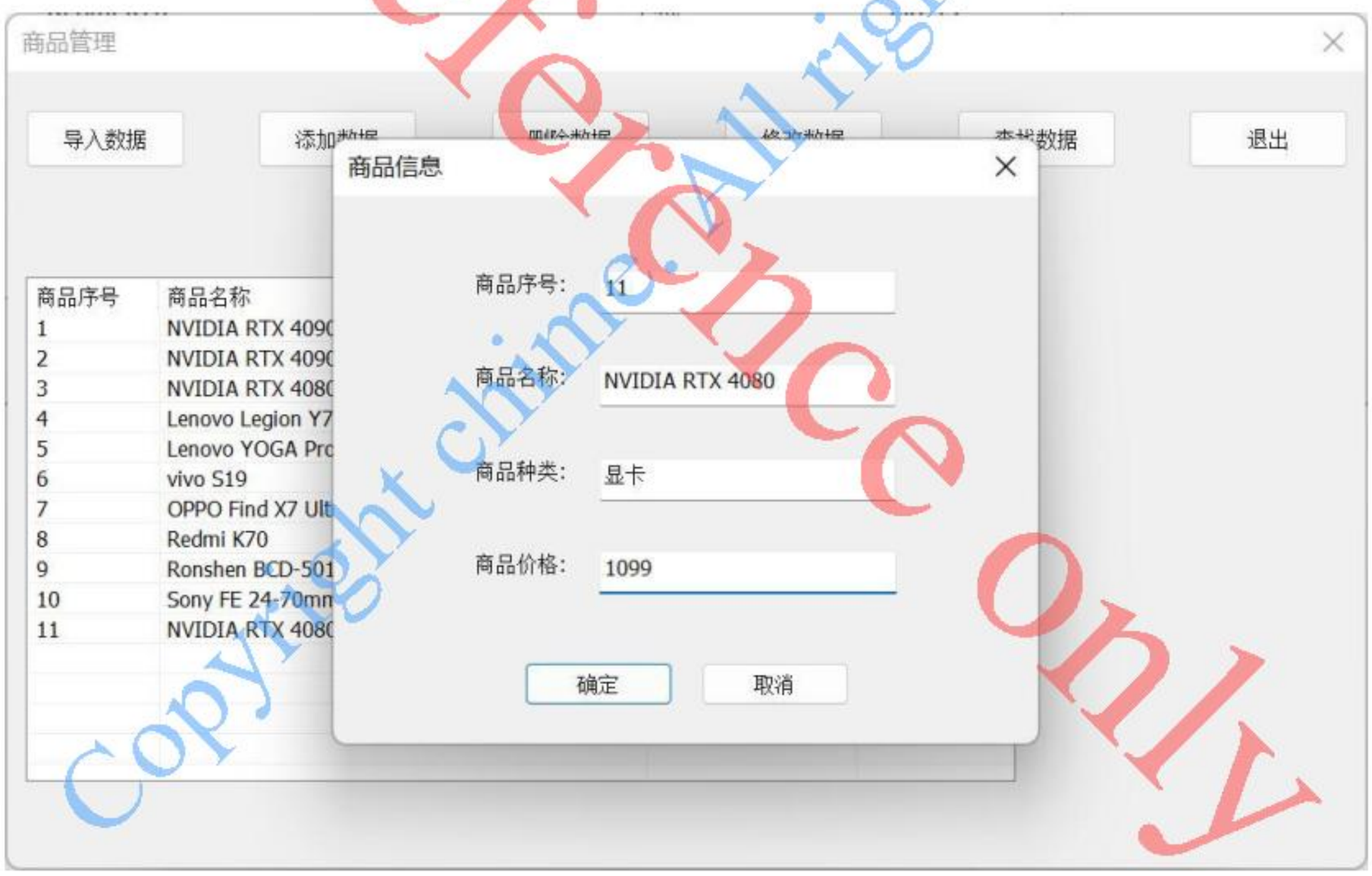


图 4.7.12 商品信息修改（修改序号为 11 的商品信息）

For reference only
Copyright chimeance All rights reserved

图 4.7.13 商品信息修改成功（序号为 11 的商品信息被修改）

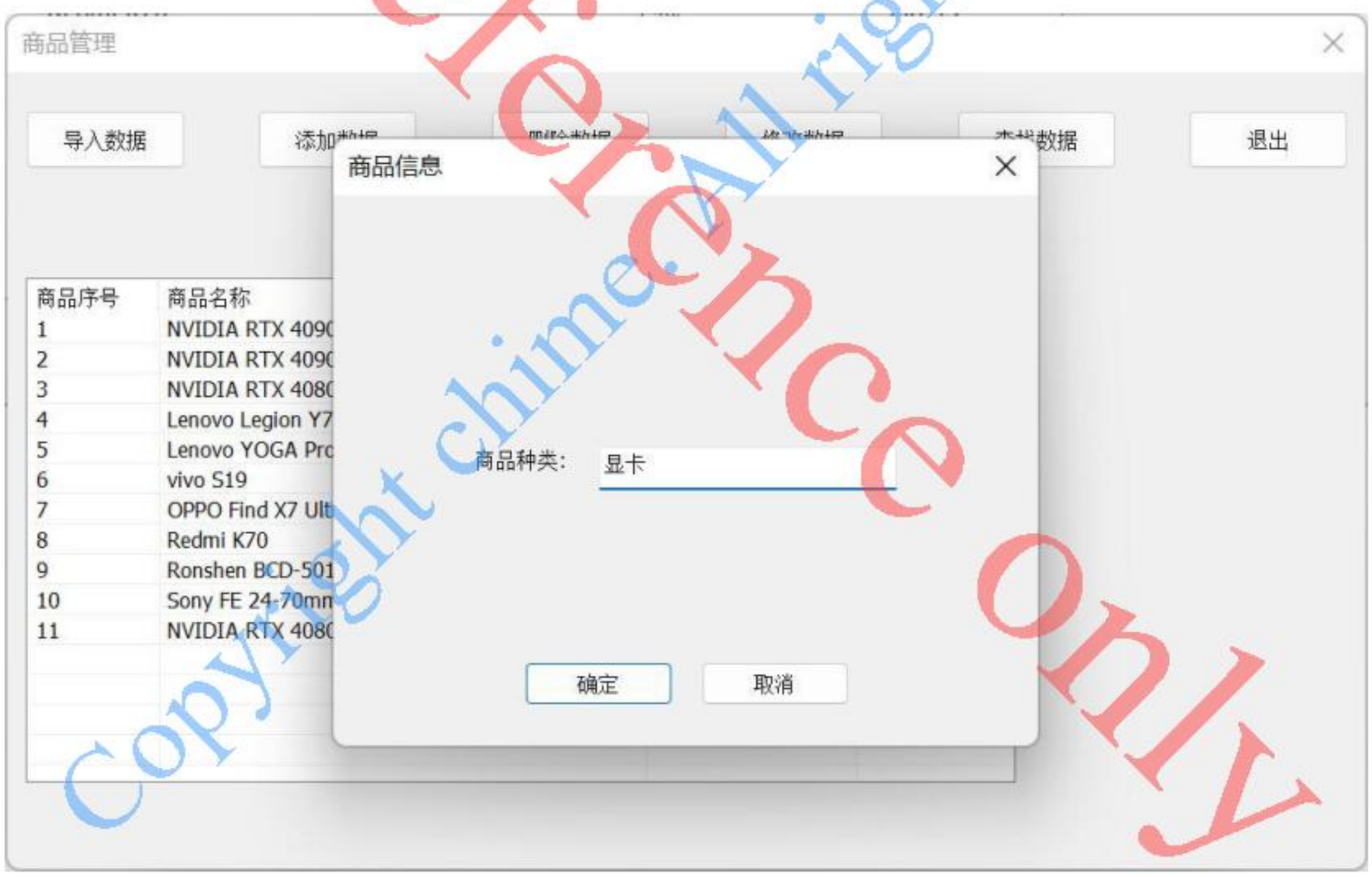


图 4.7.14 查找商品

For reference only
Copyright chimeence All rights reserved

图 4.7.15 查询到的商品信息

For reference only
Copyright chimeec All rights reserved

5.附录（代码）

5.1 数据库板块

```python
-- 数据库功能编写（增删查改）
class DatabaseManager:
    def __init__(self, host="localhost", user="root", password="123456", database="shopping"):
        self.host = host
        self.user = user
        self.password = password
        self.database = database

    def connect(self):
        return pymysql.connect(host=self.host, user=self.user, password=self.password, database=self.database, charset="utf8")

    def execute_query(self, query, params=None):
        try:
            conn = self.connect()
            with conn.cursor() as cursor:
                cursor.execute(query, params)
                result = cursor.fetchall()
                conn.commit()
            return result
        except pymysql.Error as e:
            messagebox.showerror("Database Error", f"Database error: {e}")
            return None
        finally:
            conn.close()

    def insert_user(self, name, password):
        query = "INSERT INTO user (name, password) VALUES (%s, %s)"
        self.execute_query(query, (name, password))

    def get_user(self, name):
        query = "SELECT name, password FROM user WHERE name = %s"
        return self.execute_query(query, (name,))

    def get_all_users(self):
        query = "SELECT name, password FROM user"
```

For Reference only
Copyright chimeng All rights reserved

```python
        return self.execute_query(query)

    def get_all_products(self):
        query = "SELECT cid, cname, cprice, cbrand FROM product"
        return self.execute_query(query)

    def get_all_orders(self):
        query = "SELECT oid, id, type, cid, price, calls, btime FROM orders"
        return self.execute_query(query)

    def update_order(self, oid, new_id, new_type, new_cid, new_price, new_calls, new_btime):
        query = """
        UPDATE orders
        SET id = %s, type = %s, cid = %s, price = %s, calls = %s, btime = %s
        WHERE oid = %s
        """
        self.execute_query(query, (new_id, new_type, new_cid, new_price, new_calls, new_btime, oid))
```

5.2 GUI 板块

```python
# GUI 布局
class GUI:
    def __init__(self, root):
        self.root = root
        self.root = style.master
        self.root.title('网上商城系统')
        self.root.geometry('1100x700')
        self.root.iconbitmap('logo.ico')

        # Background image
        self.canvas = tk.Canvas(self.root, height=700, width=1100)
        self.login_background = ImageTk.PhotoImage(Image.open('welcome.jpg').resize((1100, 700)))
        self.login_image = self.canvas.create_image(0, 0, anchor='nw', image=self.login_background)
        self.canvas.pack(side='top')

        # Fonts
        self.font_style = tkFont.Font(size=14)
```

Reference only
Copyright chimes All rights reserved

```
        self.font_style2 = tkFont.Font(size=12)
        self.font_style_m = ('Helvetica', 16, 'bold')

        # Username and Password Labels
        tk.Label(self.root, text='用户名:', bg='white', font=self.font_style_m, width=8, height=1).place(x=380, y=250)
        tk.Label(self.root, text='密　码:', bg='white', font=self.font_style_m, width=8, height=1).place(x=380, y=300)

        # Username Entry
        self.var_user_name = tk.StringVar()
        self.entry_user_name = tk.Entry(self.root, textvariable=self.var_user_name, width=25, font=self.font_style_m)
        self.entry_user_name.place(x=500, y=250)

        # Password Entry
        self.var_user_pwd = tk.StringVar()
        self.entry_user_pwd = tk.Entry(self.root, textvariable=self.var_user_pwd, show='*', width=25, font=self.font_style_m)
        self.entry_user_pwd.place(x=500, y=300)

        # Register and Login Buttons
        tk.Button(self.root, bg='white', text='注册', command=self.user_register, width=8, height=1, font=self.font_style_m).place(x=500, y=340)
        tk.Button(self.root, bg='white', text='登录', command=self.user_login, width=8, height=1, font=self.font_style_m).place(x=680, y=340)

        # Database Manager
        self.db_manager = DatabaseManager()
```

5.3 系统功能板块

```
def function1(self, frame_goods_select):
        tree = ttk.Treeview(frame_goods_select, columns=("cid", "cname", "cprice", "cbrand"), show="headings")
        tree.heading("cid", text="商品编码")
        tree.heading("cname", text="商品名称")
        tree.heading("cprice", text="商品价格")
        tree.heading("cbrand", text="商品品牌")
        tree.pack(fill=tk.BOTH, expand=True)
```

For reference only
Copyright chinaAI all rights reserved

```python

        tree.column("cid", width=225, anchor='center')
        tree.column("cname", width=225, anchor='center')
        tree.column("cprice", width=225, anchor='center')
        tree.column("cbrand", width=225, anchor='center')

         # 定义两种不同的标签
        tree.tag_configure('evenrow', background='#BA9BC9')  # 偶数行颜色
        tree.tag_configure('oddrow', background='#e6e6fa')   # 奇数行颜色

        self.load_product_data(tree)
        # 添加更新按钮
        update_button = tk.Button(frame_goods_select, text="更新商品", command=lambda: self.load_product_data(tree))
        update_button.pack(side=tk.BOTTOM, pady=10)


        rows = self.db_manager.get_all_products()
        for i, row in enumerate(rows):
            if i % 2 == 0:
                tree.insert("", tk.END, values=row, tags=('evenrow',))
            else:
                tree.insert("", tk.END, values=row, tags=('oddrow',))

    def function2(self, frame_goods_add_delete):

        tk.Label(frame_goods_add_delete, text="商品编码:",font=self.font_style).pack(pady=5)
        entry_cid = tk.Entry(frame_goods_add_delete, width=25, font=self.font_style)
        entry_cid.pack(pady=5)

        tk.Label(frame_goods_add_delete,text="商品名称:",font=self.font_style).pack(pady=5)
        entry_cname = tk.Entry(frame_goods_add_delete, width=25, font=self.font_style)
        entry_cname.pack(pady=5)

        tk.Label(frame_goods_add_delete, text="商品价格:",font=self.font_style).pack(pady=5)
```

For reference only
Copyright chinecence All rights reserved

```
42.         entry_cprice = tk.Entry(frame_goods_add_delete, width=25, font=self.font_style)
43.         entry_cprice.pack(pady=5)
44.
45.         tk.Label(frame_goods_add_delete,text="商品品牌:",font=self.font_style).pack(pady=5)
46.         entry_cbrand = tk.Entry(frame_goods_add_delete, width=25, font=self.font_style)
47.         entry_cbrand.pack(pady=5)
48.
49.         def add_product():
50.             cid = entry_cid.get()
51.             cname = entry_cname.get()
52.             cprice = entry_cprice.get()
53.             cbrand = entry_cbrand.get()
54.
55.             if not (cid and cname and cprice and cbrand):
56.                 messagebox.showwarning(title='警告', message='所有字段均为必填项')
57.                 return
58.             try:
59.                 query = "INSERT INTO product (cid, cname, cprice, cbrand) VALUES (%s, %s, %s, %s)"
60.                 self.db_manager.execute_query(query, (cid, cname, cprice, cbrand))
61.                 messagebox.showinfo(title='成功', message='商品添加成功')
62.
63.                 entry_cid.delete(0, tk.END)
64.                 entry_cname.delete(0, tk.END)
65.                 entry_cprice.delete(0, tk.END)
66.                 entry_cbrand.delete(0, tk.END)
67.             except pymysql.Error as e:
68.                 messagebox.showerror(title='错误', message=f'数据库错误: {e}')
69.     def function3(self, frame_order_select):
70.         tree = ttk.Treeview(frame_order_select, columns=("oid", "id", "type", "cid", "price", "calls", "btime"), show="headings")
71.         tree.heading("oid", text="订单编码")
72.         tree.heading("id", text="用户 ID")
73.         tree.heading("type", text="商品种类")
74.         tree.heading("cid", text="商品编码")
75.         tree.heading("price", text="价格")
76.         tree.heading("calls", text="电话号码")
```

For reference only
Copyright chimeA all rights reserved

```
        tree.heading("btime", text="订单时间")

        tree.column("oid", width=100, anchor='center')
        tree.column("id", width=100, anchor='center')
        tree.column("type", width=100, anchor='center')
        tree.column("cid", width=100, anchor='center')
        tree.column("price", width=100, anchor='center')
        tree.column("calls", width=200, anchor='center')
        tree.column("btime", width=200, anchor='center')

        tree.pack(fill=tk.BOTH, expand=True)

        # 定义两种不同的标签
        tree.tag_configure('evenrow', background='#BA9BC9')
        tree.tag_configure('oddrow', background='#e6e6fa')

        self.load_order_data(tree)

        update_button = tk.Button(frame_order_select, text="更新订单", command=lambda: self.load_order_data(tree))
        update_button.pack(side=tk.BOTTOM, pady=10)

        return tree

    def function4(self, frame_order_revise):
        tk.Label(frame_order_revise, text="更新的订单编码",font=self.font_style2).pack(pady=5)
        global entry_oid
        entry_oid = tk.Entry(frame_order_revise, width=30,font=self.font_style2)
        entry_oid.pack(pady=5)

        tk.Label(frame_order_revise, text="更新用户ID",font=self.font_style2).pack(pady=5)
        global entry_id
        entry_id = tk.Entry(frame_order_revise, width=30,font=self.font_style2)
        entry_id.pack(pady=5)

        tk.Label(frame_order_revise, text="更新商品种类",font=self.font_style2).pack(pady=5)
        global entry_type
        entry_type = tk.Entry(frame_order_revise, width=30,font=self.font_style2)
```

Copyright chinese All rights reserved
For reference only

```
        entry_type.pack(pady=5)

        tk.Label(frame_order_revise, text="更新商品编码",font=self.font_style2).pack(pady=5)
        global entry_cid
        entry_cid = tk.Entry(frame_order_revise, width=30,font=self.font_style2)
        entry_cid.pack(pady=5)

        tk.Label(frame_order_revise, text="更新价钱",font=self.font_style2).pack(pady=5)
        global entry_price
        entry_price = tk.Entry(frame_order_revise, width=30,font=self.font_style2)
        entry_price.pack(pady=5)

        tk.Label(frame_order_revise, text="更新电话号码",font=self.font_style2).pack(pady=5)
        global entry_calls
        entry_calls = tk.Entry(frame_order_revise, width=30,font=self.font_style2)
        entry_calls.pack(pady=5)

        tk.Label(frame_order_revise, text="更新订单时间",font=self.font_style2).pack(pady=5)
        global entry_btime
        entry_btime = tk.Entry(frame_order_revise, width=30,font=self.font_style2)
        entry_btime.pack(pady=5)

        update_button = tk.Button(frame_order_revise, text="更新订单",font=self.font_style2, command=self.perform_update)
        update_button.pack(pady=20)
```

Copyright chinese AI Lab. All rights reserved
For Reference only

5.4 版本②代码

```
#include "pch.h"
#include "MySQLinterface.h"

// 连接数据库
bool MySQLinterface::Connect_MySQL()
{
    mysql_init(&mysqlCon);

```

```
    if (!mysql_real_connect(&mysqlCon, "localhost", "root", "123456", "shopping", 3306, NULL, 0))
    {
        AfxMessageBox(_T("访问数据库失败！"));
        return false;
    }
    else
    {
        AfxMessageBox(_T("访问数据库成功！"));
        mysql_query(&mysqlCon, "SET NAMES 'GB2312'");
    }
    return true;
}

// 获取全部信息
vector<CommodityInfo> MySQLinterface::Get_All_Info()
{
    // vector 容器存储商品信息
    vector<CommodityInfo> Info;

    // C 语言组合字符串
    snprintf(sql, SQL_MAX, "SELECT * FROM shopping.commodity;");

    // 查询数据
    int ret = mysql_real_query(&mysqlCon, sql, (unsigned long)strlen(sql));

    // 判断查询结果
    if (ret)
    {
        AfxMessageBox(_T("查询失败！"));
        return Info;
    }

    // 获取结果集
    res = mysql_store_result(&mysqlCon);

    // 获取查询到的一行数据
    // 给 row 赋值，判断 row 是否为空，不为空就打印数据。
    while (row = mysql_fetch_row(res))
    {
        CommodityInfo commodityinfo = CommodityInfo(atoi(row[0]), row[1], row[2], stod(row[3]));
        Info.push_back(commodityinfo);
```

For reference only
Copyright chinecat all rights reserved

```
50.     }
51.
52.     // 释放结果集
53.     mysql_free_result(res);
54.
55.     return Info;
56.}
57.
58.// 添加商品信息
59.bool MySQLinterface::Add_Info(CommodityInfo Info)
60.{
61.     // C 语言组合字符串
62.     snprintf(sql, SQL_MAX, "INSERT INTO shopping.commodity VALUES
   (%d, '%s', '%s', %.2f);", Info.m_c_id, Info.m_c_name.c_str(), Inf
   o.m_c_kind.c_str(), Info.m_c_price);
63.
64.     // 查询数据
65.     int ret = mysql_real_query(&mysqlCon, sql, (unsigned long)str
   len(sql));
66.
67.     // 判断添加结果
68.     if (ret)
69.     {
70.         AfxMessageBox(_T("添加失败！"));
71.         return false;
72.     }
73.     return true;
74.}
75.
76.// 删除商品信息
77.bool MySQLinterface::Delete_Info(CommodityInfo Info)
78.{
79.     // C 语言组合字符串
80.     snprintf(sql, SQL_MAX, "DELETE FROM shopping.commodity WHERE
   (c_id = %d);", Info.m_c_id);
81.
82.     // 查询数据
83.     int ret = mysql_real_query(&mysqlCon, sql, (unsigned long)str
   len(sql));
84.
85.     // 判断删除结果
86.     if (ret)
87.     {
88.         AfxMessageBox(_T("删除失败！"));
```

For reference only
Copyright chimeAll rights reserved

```
        return false;
    }
    return true;
}

// 修改商品信息
bool MySQLinterface::Update_Info(CommodityInfo oldInfo, CommodityInfo newInfo)
{
    // C 语言组合字符串
    snprintf(sql, SQL_MAX, "UPDATE shopping.commodity SET c_id = %d, c_name = '%s', c_kind = '%s', c_price = %.2f WHERE (c_id = %d);", newInfo.m_c_id, newInfo.m_c_name.c_str(), newInfo.m_c_kind.c_str(), newInfo.m_c_price, oldInfo.m_c_id);

    // 查询数据
    int ret = mysql_real_query(&mysqlCon, sql, (unsigned long)strlen(sql));

    // 判断修改结果
    if (ret)
    {
        AfxMessageBox(_T("修改失败！"));
        return false;
    }
    return true;
}

// 查找商品信息
vector<CommodityInfo> MySQLinterface::Search_Info(CString kind)
{
    vector <CommodityInfo> result;

    // C 语言组合字符串
    snprintf(sql, SQL_MAX, "SELECT * FROM shopping.commodity WHERE (c_kind = '%s');", kind);

    // 查询数据
    int ret = mysql_real_query(&mysqlCon, sql, (unsigned long)strlen(sql));

    // 判断查询结果
    if (ret)
    {
```

For reference only
Copyright chimed. All rights reserved

```
        AfxMessageBox(_T("查询失败！"));
        return result;
    }

    // 获取结果集
    res = mysql_store_result(&mysqlCon);

    // 获取查询到的一行数据
    // 给 row 赋值，判断 row 是否为空，不为空就打印数据。
    while (row = mysql_fetch_row(res))
    {
        CommodityInfo commodityinfo = CommodityInfo(atoi(row[0]), row[1], row[2], stod(row[3]));
        result.push_back(commodityinfo);
    }

    // 释放结果集
    mysql_free_result(res);

    return result;
}
```

注：由于代码量较大，这里只展示①与②的核心代码。

For reference only
Copyright chimeCAll rights reserved