



西安电子科技大学
XIDIAN UNIVERSITY

面向对象程序设计

Object Oriented Programming

实验报告

Experimental Report

学号	*****	姓名	***
班级	****	任课教师	***
实验名称	购物车模拟系统		
实验学期	2023 – 2024 学年第 2 学期		
实验日期	2024.6.8	实验地点	G346
报告成绩			

西安电子科技大学计算机科学与技术学院

一、 实验环境

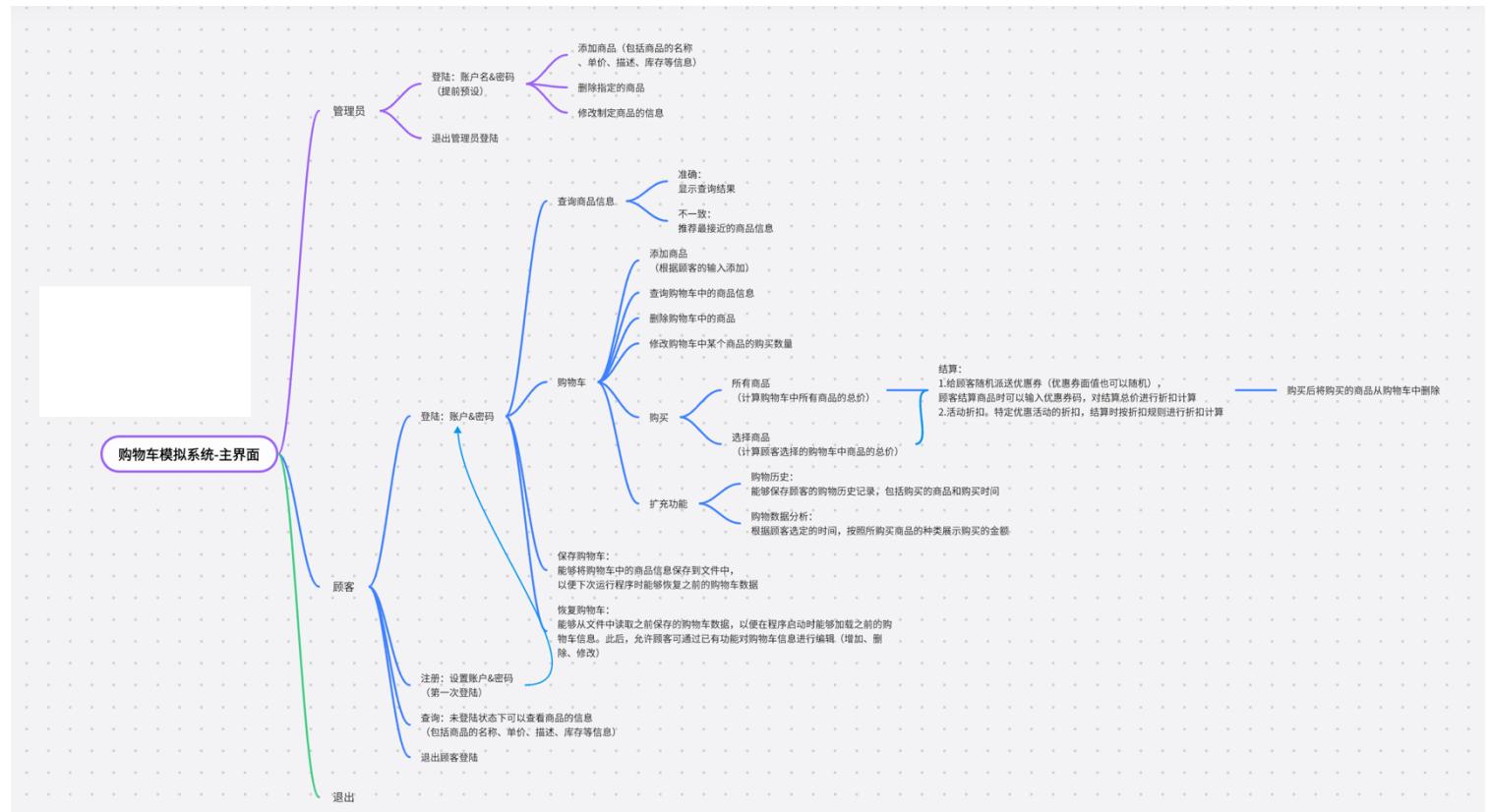
写出本次实验的软件和硬件环境。

软件环境：Xcode Version 15.4

硬件环境：macOS Sonoma 14.5，处理器为 2.3 GHz 八核 Intel Core i9

二、 实验内容

本次上机实验的完整需求（网上购物模拟系统）。



三、 实验步骤

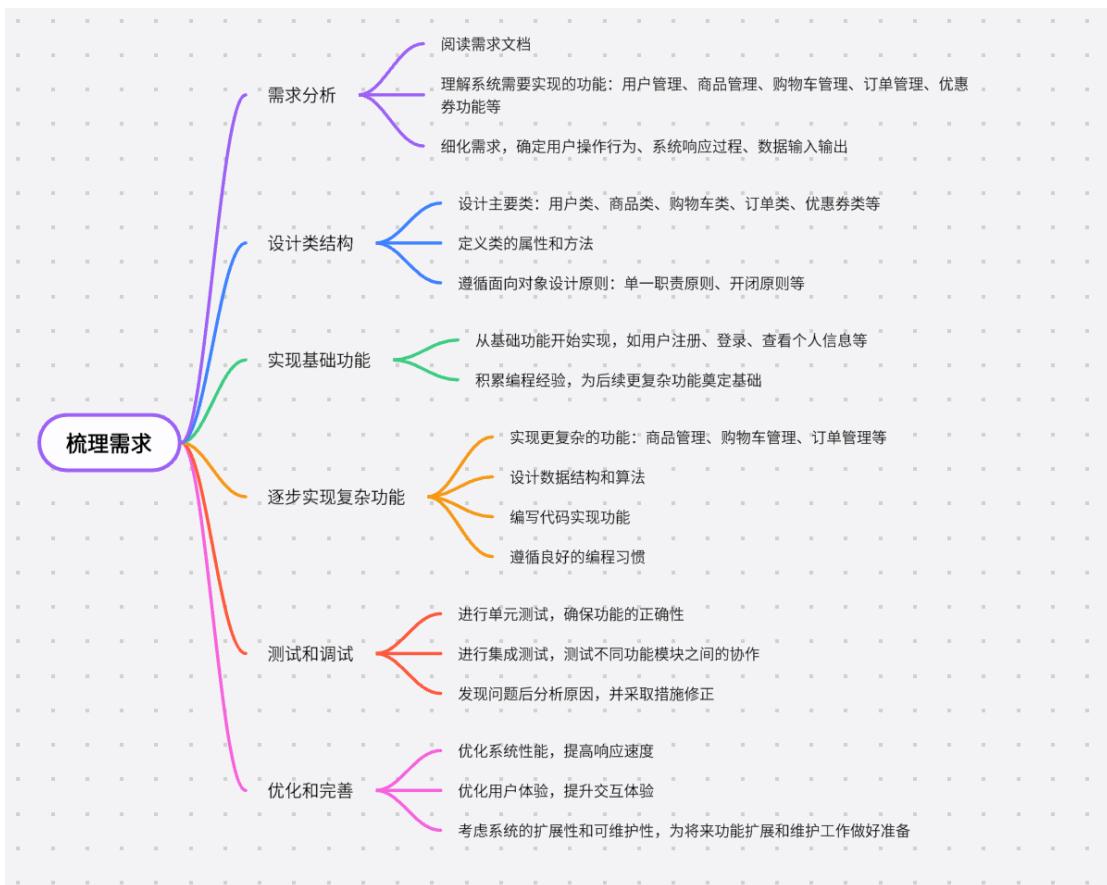
写出实验思路、关键数据结构定义和类的接口定义、主要算法流程以及遇到的问题和解决方案。

首先面对题目了解到大概要实现的各种具体功能包括用户管理、商品管理、购物车管理、订单管理以及优惠券功能等。根据需求分析,我设计出系统中需要的主要类,如用户类、商品类、购物车类、订单类、优惠券类等。

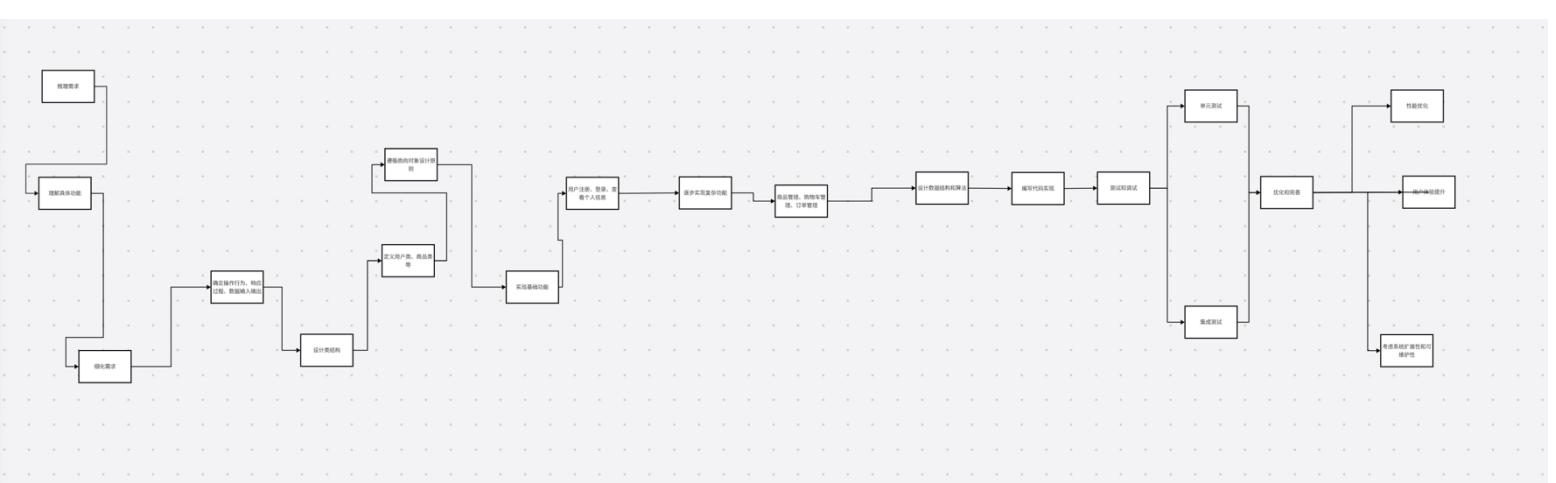
在类设计完成后,我就从最基础的功能开始实现,即从第一次上机要求中的用户注册、登录、查看个人信息功能开始实现。思考在 main 函数中要进行什么操作。完成基础功能后,逐步实现更复杂的功能,如商品管理、购物车管理。对于每个功能模块,我会先设计出相应的数据结构和算法,然后编写代码进行实现,一点一点测试调 bug, 如果太过于自信, 写完很多之后再测试发现有 bug 会很难找到错误 (血的教训), 而且还要进行集成测试, 测试不同模块之间的协作是否正常。

最后再进行一些界面上的优化或者进行购物车操作逻辑的优化，例如登陆界面的展示优化，比原来的只是几行提示更加直观。

所有的流程大致如下（思维导图）：



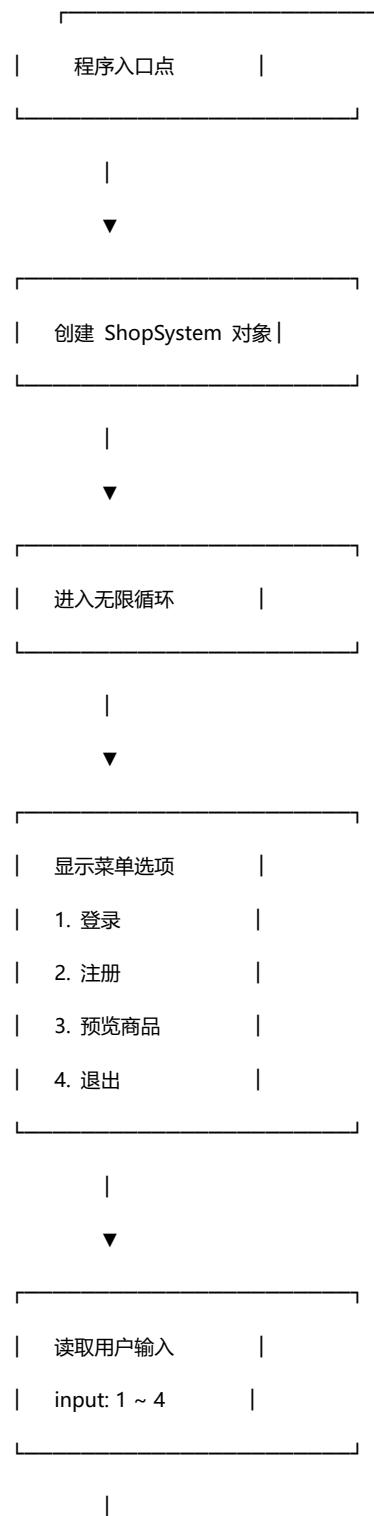
整体的工作流程大致如下（流程图）：

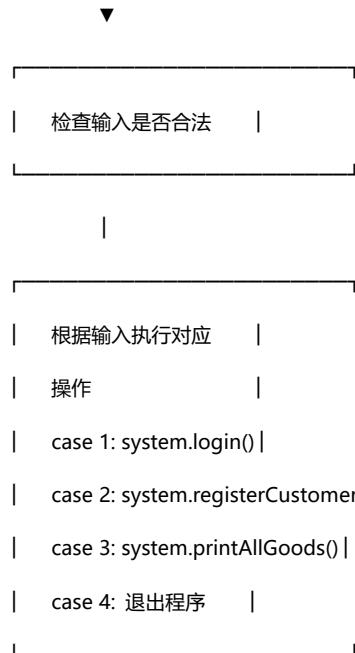


比较重要的主要架构以及整个调用接口逻辑

Main.cpp

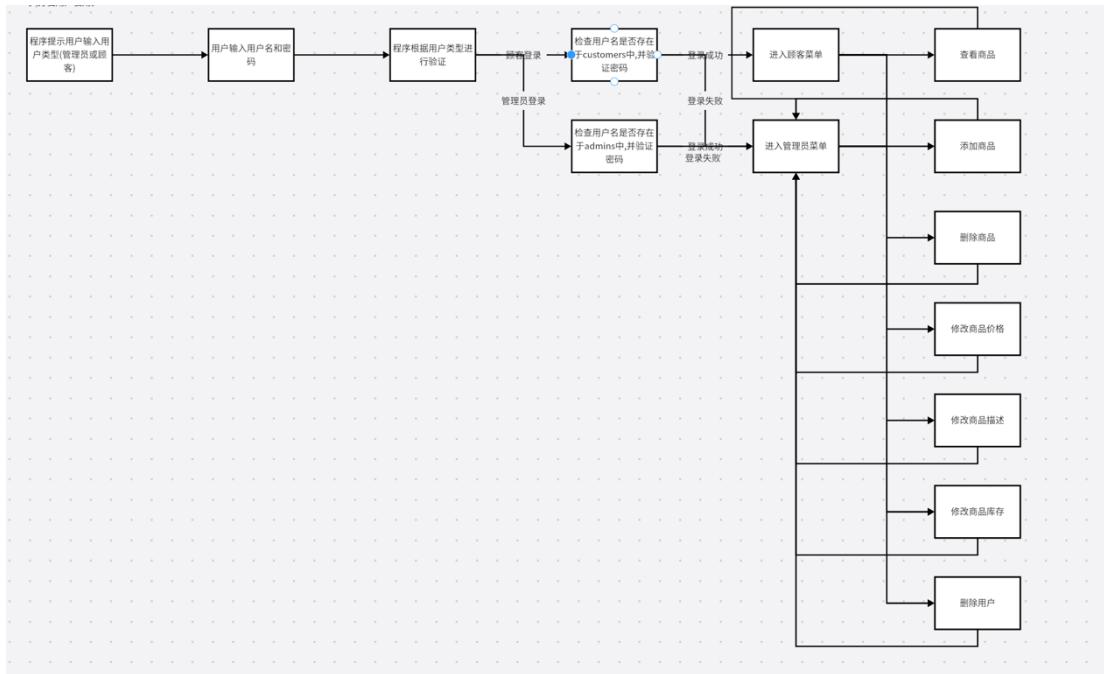
这当然是最重要的部分，是我们整个系统的进入界面，通过用户的选择接口，系统进入不同的功能当中。



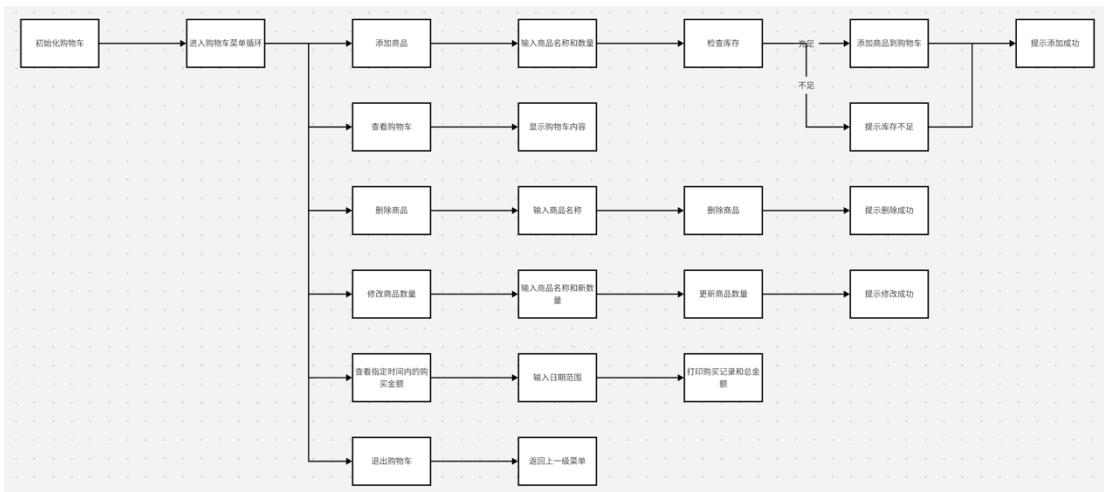


用户登录功能函数十分重要，以及进入顾客购物车后的各种接口设计十分重要。

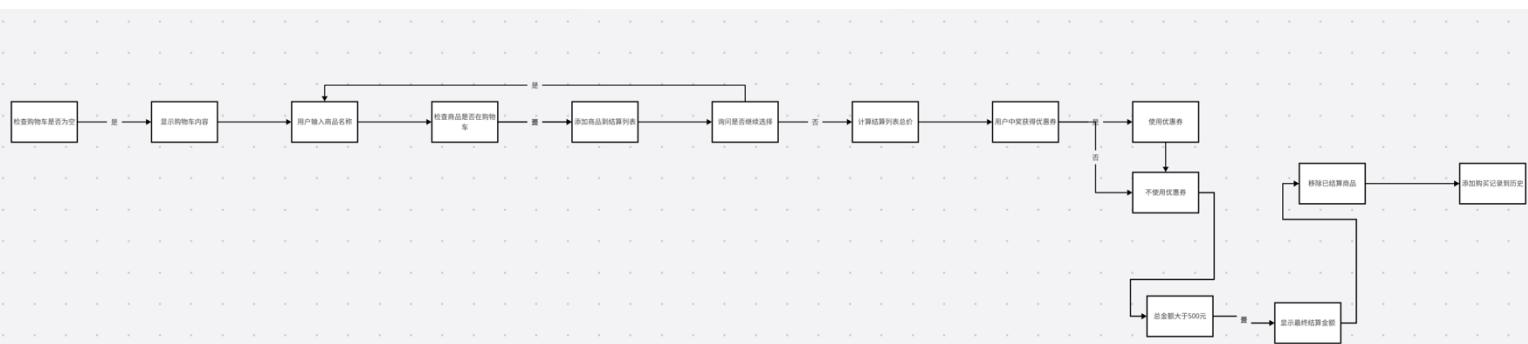
用户登录 login 的操作逻辑以及流程：



以下是顾客进入购物车系统之后的接口以及操作逻辑:



用户选择购买购物车内的商品，这个函数实现也很重要，如果顾客选择购买购物车内的物品，用户可以选择一次性购买所有的商品，也可以选择一个一个添加自己想买的商品最终进行购买，以一次一次添加商品购买的实现函数为例进行大体流程概述：



其中比较重要的算法：

名字不匹配--推荐商品算法

void shopSystem::recommendGoods(const string& keyword)

我这个算法实现的比较低级，大小写敏感，实际上没有实现传统意义上的推荐，内部使用的就是进行子串的匹配。具体的流程如下：

首先，它定义了一个“推荐列表”，里面存放着商品的名字和商品对象。这个列表本来是空的。

然后，它会去检查商店里所有的商品。对于每一个商品，它会看看这个商品的名字里是不是包

含了用户输入的那个关键词。如果一个商品的名字里真的有这个关键词,那算法就会把这个商品添加到"推荐列表"里。等把所有商品都检查完了以后,算法会看看"推荐列表"里有没有东西。如果"推荐列表"里有商品,那算法就会把这些商品的详细信息都显示出来,让用户看看。但如果"推荐列表"里什么都没有,那算法就会告诉用户,没有找到和关键词相关的商品。

用于从用户获取日期范围的算法

```
pair<chrono::system_clock::time_point, chrono::system_clock::time_point>
shopSystem::getDateRangeFromUser()
```

首先,程序提示用户输入起始日期和结束日期,格式为 "YYYY-MM-DD"。接下来,程序使用 istringstream 将输入的字符串转换为 tm 结构体。tm 结构体包含年、月、日等字段,可以用来表示日期。程序调用 get_time() 函数将字符串解析为 tm 结构体。这个函数使用指定的格式化模式(在这里是 "%Y-%m-%d")来解析输入的字符串。然后,程序使用 mktime() 函数将 tm 结构体转换为 time_t 类型。time_t 是一个整数值,表示自某个固定时间点(通常是 1970 年 1 月 1 日 00:00:00 UTC)经过的秒数。

最后,程序使用 chrono::system_clock::from_time_t() 将 time_t 值转换为 chrono::system_clock::time_point 类型。这个类型表示一个特定的时间点,可以用于计算两个时间点之间的时间差等操作。为了确保起始日期在结束日期之前或相等,程序会比较两个时间点,如果起始日期大于结束日期,就交换它们的位置。

遇到的问题及解决:

遇到的问题以及解决其实已经在每次的上机设计文件中已经说明中,以下内容将从其中进行摘取。

1.初期我在设置 User 的构造函数以及析构函数是放在 private 中的,后面我在定义 Customer 的构造函数以及析构函数时就会出现报错提醒 Base class 'User' has private default constructor,原因是在 Customer 类中的构造函数使用了 User 类的默认构造函数,但是 User 类中的默认构造函数是私有的,无法在子类 Customer 中直接调用。我将 User 的这些相关函数放入 protected 之中就解决了问题,加深了 private, protected, public 的三个用法,并在 stack overflow 中查询了网友一语中的的描述:

private hides from other classes within the package. public exposes to classes outside the package. protected is a version of public restricted only to subclasses.

2.关于 Xcode 使用特性的问题

本次在 shop system 中实现了构造函数,登陆函数,注册函数,保存顾客数据函数,输出商品信息函数,实现了本次上机的所有要求,其中顾客注册后的保存顾客数据部分让我调试了很久,原因是 Xcode 这个 Ide 需要进行另外的设置才能把数据 output 到文件中,中文

互联网上对于 Xcode 的使用教程并不多，我没有找到讲的很明白的，后续再 YouTube 才找到正确的使用方式。

3. 关于实现优惠券随机发放机制的实现

我对随机进行发送优惠券的功能设想是在结算之前可以通过 rand 函数生成数字，然后通过其除以 2 得到的余数来判断是否发送优惠券

```
if (rand() % 2 == 0) { sendRandomCoupon(customerName,totalCost); }
```

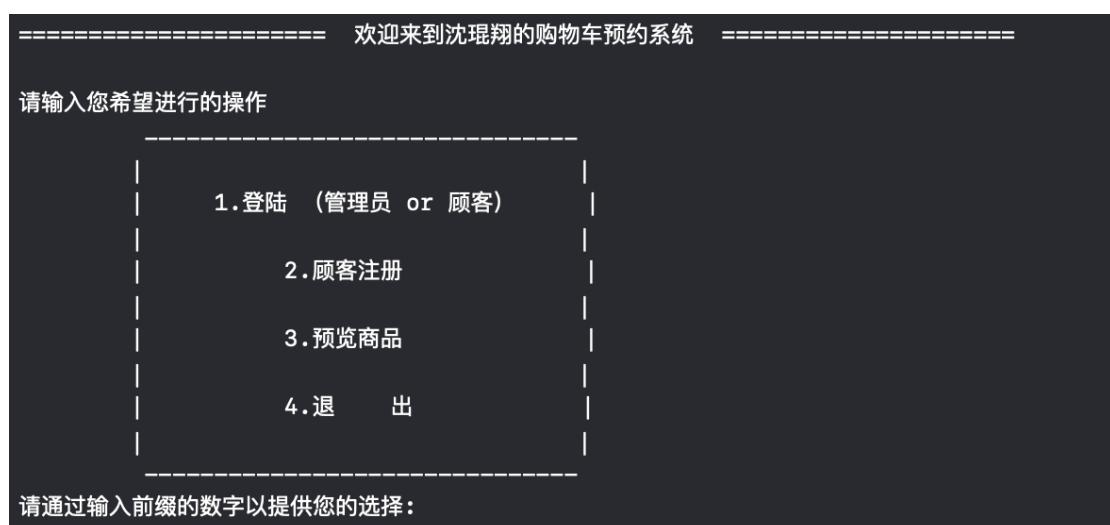
但是通过多次测试之后我发现几乎无法得到优惠券，经过搜索得知在使用 rand 函数之前需要初始化随机数生成器，否则，每次运行程序可能都会得到相同的随机数序列，导致 rand%2==0 条件始终不成立，于是我在 main 函数的开始部分添加了 srand(static_cast<unsigned int>(time(0))); // 初始化随机数生成器 之后再可以实现

四、实验结果

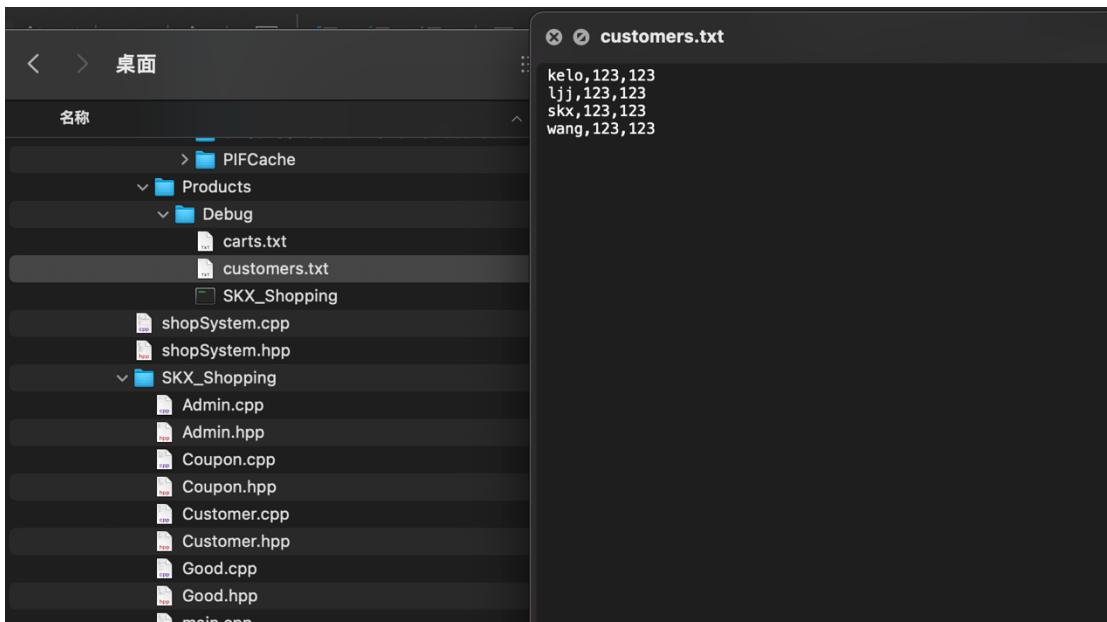
针对不同输入，该程序都存在哪几类可能出现的情况，你的测试数据要完全覆盖了你所想到的这些情况，并给出测试结果。

注：因为本次实验报告不是在同一天写完的，电脑系统会根据时间进行亮色模式和暗黑模式的转换，所以截图中有的是白底有的是黑底，见谅。

1. 编译并运行程序，显示主界面：



2. 输入 “1”，进行登陆



这是原先在 customers 记录的文件中记录的用户-密码-手机号。

2.1 输入 2 进入顾客登陆界面，登陆用户 skx

```
请通过输入前缀的数字以提供您的选择:
1
输入用户类型(1.管理员 2.顾客): 2
输入用户名: skx
输入密码: 123
***顾客登录成功***
您要进行:
1.查看商品
2.查找商品
3.打开购物车
4.购买购物车内商品
5.修改密码
6.退出
7.注销账号
请选择操作 (1-7) :
```

2.1.1 输入 1 查看商品

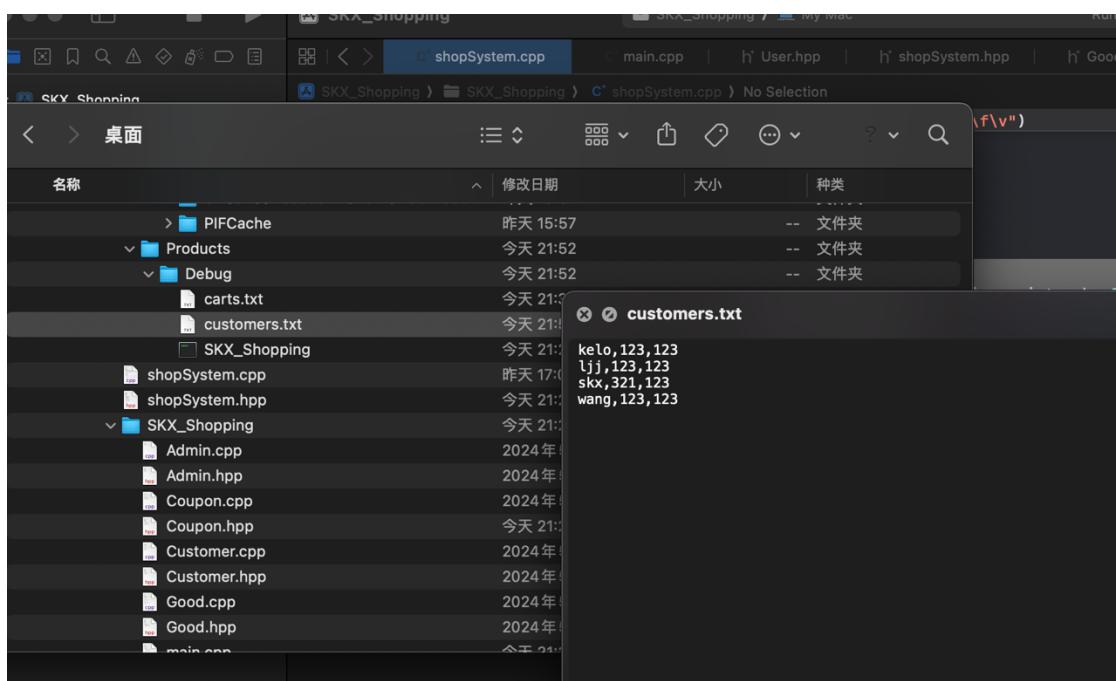
```
请选择操作 (1-7) :1
商品名称: AirPods_2
价格:1799.99¥
描述: 苹果无线耳机 (库克的挤牙膏产品)
库存: 99件
商品名称: Anker_Power_Bank
价格:109¥
描述: 专业做电子产品的Anker, 充电宝质感一流
库存: 9件
商品名称: Macbook
价格:12999¥
描述: M2芯片, 你还在等什么!
库存: 2000件
商品名称: iPhone_14
价格:7899¥
描述: 拥有此产品你可以登岛!
库存: 100件
```

2.1.2 输入 2 进行模糊查找商品

```
请选择操作 (1-7) :2
输入商品名称: Anker
未找到商品: Anker
推荐商品:
商品名称: Anker_Power_Bank
价格:109¥
描述: 专业做电子产品的Anker, 充电宝质感一流
库存: 9件
```

2.1.3 输入 5 修改用户的密码

```
请选择操作 (1-7) :5
输入新密码: 321
***密码修改成功***
```



2.1.4 输入 3 进入顾客的购物车系统

请选择操作 (1-7) :3

您已打开购物车，您需要：

1. 添加商品
2. 查看购物车
3. 删除商品
4. 修改商品数量
5. 查看指定时间内商品购买金额
6. 退出购物车

请选择 (1-6) :

2.1.4.1 输入 1 在购物车添加商品

您已打开购物车，您需要：

1. 添加商品
 2. 查看购物车
 3. 删除商品
 4. 修改商品数量
 5. 查看指定时间内商品购买金额
 6. 退出购物车
- 请选择 (1-6) : 1
查询后输入正确要添加商品名称: Macbook
所需数量为: 5
5件Macbook添加成功

2.1.4.2 输入 2 查看购物车

请选择 (1-6) : 2
商品名称: Macbook, 数量: 5
单价: 12999￥, 描述: M2芯片, 你还在等什么!
总价: 64995￥

2.1.4.3 输入 4 修改商品数量

请选择 (1-6) : 4
查询后输入正确要修改数量商品名称: Macbook
数量更改为: 10
顾客 skx 的购物车中商品 Macbook 的数量已修改为 10
Macbook数量已更改
您已打开购物车，您需要：
1. 添加商品
2. 查看购物车
3. 删除商品
4. 修改商品数量
5. 查看指定时间内商品购买金额
6. 退出购物车
请选择 (1-6) : 2
商品名称: Macbook, 数量: 10
单价: 12999￥, 描述: M2芯片, 你还在等什么!
总价: 129990￥

2.1.4.4 输入 3 选择删除商品

```

请选择 (1-6) : 3
查询后输入正确要删除商品名称: Macbook
顾客 skx 的购物车中已删除商品 Macbook
***Macbook删除成功***
您已打开购物车, 您需要:
1.添加商品
2.查看购物车
3.删除商品
4.修改商品数量
5.查看指定时间内商品购买金额
6.退出购物车
请选择 (1-6) : 2
总价: 0￥

```

2.1.4.5 重新加入商品, 然后退出购物车再购买, 再查看购买记录

```

请选择操作 (1-7) :4
您已进入购买界面, 请选择 (1.从购物车中选择商品进行结算 2.结算所有商品 3.退出) : 2
结算前, 总价为: 64995 ￥
恭喜您已获得优惠券码: COUPON191374, 面值为: 3249元
请选择是否使用优惠券 (1.是 2.否) : 2
***开业大酬宾!!! 有活动啦~~~消费总额大于500元有优惠, 已为您自动打9折!!!! ***
最终结算金额为: 58495.5 ￥。购买成功, 已从购物车中移除购买的商品。
您已进入购买界面, 请选择 (1.从购物车中选择商品进行结算 2.结算所有商品 3.退出) : 3
您要进行:
1.查看商品
2.查找商品
3.打开购物车
4.购买购物车内商品
5.修改密码
6.退出
7.注销账号
请选择操作 (1-7) :3
您已打开购物车, 您需要:
1.添加商品
2.查看购物车
3.删除商品
4.修改商品数量
5.查看指定时间内商品购买金额
6.退出购物车
请选择 (1-6) : 5
请输入起始日期 (YYYY-MM-DD): 2022-09-09
请输入结束日期 (YYYY-MM-DD): 2025-09-09
商品名称: Macbook, 购买数量: 5, 购买金额: 64995, 购买时间: 2024-06-07 22:38:38

```

2.2.进入管理员界面

The screenshot shows a terminal window with the following details:

- Terminal Title:** SKX_Shopping
- Current File:** shopSystem.cpp
- Code Snippet:**

```

12
13
14 //购物系统的构建，包括初始化管理员账号以及商品
15 shopSystem::shopSystem()
16 {
17     string admin_name,admin_key;
18     admin_name="shenkunxiang";
19     admin_key="123";
20     admins[admin_name]=Admin(admin_name,admin_key);
21     goods["AirPods_2"]=Good("AirPods_2",1799.99,"苹果无线耳机（库克的挤牙膏产品）",99);
22     goods["iPhone_14"]=Good("iPhone_14",7899,"拥有此产品你可以登岛！",100);
23     goods["Anker_Power_Bank"]=Good("Anker_Power_Bank",109,"专业做电子产品的Anker，充电宝质感一流",99);
24     goods["Macbook"]=Good("Macbook",12999,"M2芯片，你还在等什么！",2000);
25 }
```

- Application Output:**

```

===== 欢迎来到沈琨翔的购物车预约系统 =====

请输入您希望进行的操作

-----|-----|-----|-----|-----|-----|-----|-----|
| 1. 登陆 (管理员 or 顾客) |-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|
| 2. 顾客注册 |-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|
| 3. 预览商品 |-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|
| 4. 退    出 |-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|

```

请通过输入前缀的数字以提供您的选择:

 - 1

输入用户类型(1.管理员 2.顾客): 1
输入用户名: shenkunxiang
输入密码: 123
管理员登录成功
您要进行:
1.添加商品
2.删除商品
3.修改商品价格
4.修改商品描述
5.修改商品库存
6.删除用户
7.退出

2.2.1 管理员添加商品,然后进行修改价格、描述、库存

```

请选择操作 (1-7) :1
输入商品名称: Watch
输入商品价格: 2000
输入商品描述: iWatch
输入商品库存: 100
***商品添加成功***

```

请通过输入前缀的数字以提供您的选择：

3

商品名称: AirPods_2

价格:1799.99¥

描述: 苹果无线耳机 (库克的挤牙膏产品)

库存: 99件

商品名称: Anker_Power_Bank

价格:109¥

描述: 专业做电子产品的Anker, 充电宝质感一流

库存: 9件

商品名称: Macbook

价格:12999¥

描述: M2芯片, 你还在等什么!

库存: 1995件

商品名称: Watch

价格:2000¥

描述: iWatch

库存: 100件

请选择操作 (1-7) :4

输入要修改描述商品名称: Watch

输入新描述: hahaha

Watch 的描述已更新为: hahaha

您要进行:

- 1.添加商品
- 2.删除商品
- 3.修改商品价格
- 4.修改商品描述
- 5.修改商品库存
- 6.删除用户
- 7.退出

请选择操作 (1-7) :5

输入要修改库存商品名称: Watch

输入新库存: 200

Watch 的库存已更新为: 200

3.在主界面输入 3 进入预览商品界面

请通过输入前缀的数字以提供您的选择：

3

商品名称: AirPods_2

价格:1799.99¥

描述: 苹果无线耳机 (库克的挤牙膏产品)

库存: 99件

商品名称: Anker_Power_Bank

价格:109¥

描述: 专业做电子产品的Anker, 充电宝质感一流

库存: 9件

商品名称: Macbook

价格:12999¥

描述: M2芯片, 你还在等什么!

库存: 2000件

商品名称: Watch

价格:2000¥

描述: hahaha

库存: 200件

商品名称: iPhone_14

价格:7899¥

描述: 拥有此产品你可以登岛!

库存: 100件

4. 测试购物车功能中将数据录入文件功能、从文件中读取功能、加入购物车产品数量溢出状况反应

库存中有如下产品，注意库存的状态：

请选择操作 (1-7) :1

商品名称: AirPods_2

价格:1799.99¥

描述: 苹果无线耳机 (库克的挤牙膏产品)

库存: 99件

商品名称: Anker_Power_Bank

价格:109¥

描述: 专业做电子产品的Anker, 充电宝质感一流

库存: 9件

商品名称: Macbook

价格:12999¥

描述: M2芯片, 你还在等什么!

库存: 2000件

商品名称: iPhone_14

价格:7899¥

描述: 拥有此产品你可以登岛!

库存: 100件

我在 skx 用户的购物车中尝试加入商品：

1.正常

```
查询后输入正确要添加商品名称: Macbook  
所需数量为: 10  
***10件Macbook添加成功***
```

2.正常

```
请选择 (1-6) : 1  
查询后输入正确要添加商品名称: iPhone_14  
所需数量为: 10  
***10件iPhone_14添加成功***
```

3.正常

```
请选择 (1-6) : 1  
查询后输入正确要添加商品名称: AirPods_2  
所需数量为: 10  
***10件AirPods_2添加成功***
```

4.溢出检测

```
请选择 (1-6) : 1  
查询后输入正确要添加商品名称: Anker_Power_Bank  
所需数量为: 10000  
库存不足
```

检查购物车中的数据是否录入文件中：

如图可见，在我退出 skx 这一顾客用户之后，文件中加载了这一用户的购物车信息



将程序终止运行，重新编译运行然后登录 skx 这一顾客用户，可以发现 skx 这一顾客的购物车数据已经从文件中录入到系统中，如下图所示



5. 测试购物功能中单独选择物品进行结算的功能、选择所有直接结算的功能、结算的时候是否实现随机发放优惠券并使用功能、购物之后是否进行了库存减少的功能

单独选择物品进行结算：

购买了购物车中的一件商品，可以观察到没有收到优惠券，如果收到了优惠券系统会发送优惠券码，然后用户选择是否使用，后面再测试全体购买的时候会不会触发优惠券发送。

```
您已进入购买界面，请选择（1.从购物车中选择商品进行结算 2.结算所有商品 3.退出）：1
```

```
商品名称：AirPods_2，数量：10
单价：1799.99¥，描述：苹果无线耳机（库克的挤牙膏产品）
商品名称：iPhone_14，数量：10
单价：7899¥，描述：拥有此产品你可以登岛！
商品名称：Macbook，数量：10
单价：12999¥，描述：M2芯片，你还在等什么！
总价：226980¥
请输入要购买的商品名称：AirPods—2
购物车中没有 AirPods—2 这个商品
是否继续选择其他商品？（1.是 2.否）：1
请输入要购买的商品名称：AirPods_2
AirPods_2 已添加到结算列表。
是否继续选择其他商品？（1.是 2.否）：2
顾客 skx 的购物车中已删除商品 AirPods_2
结算前，总价为：17999.9 ¥
请选择是否使用优惠券（1.是 2.否）：2
***开业大酬宾！！！有活动啦～～消费总额大于500元有优惠，已为您自动打9折！！！***
最终结算金额为：16199.9 ¥
```

选择结算购物车所有的商品：

可以观察到触发到了优惠券发送，此时应该所有的物品都已经结算情况，现在我们看一下库存有没有更新

```
您已进入购买界面，请选择（1.从购物车中选择商品进行结算 2.结算所有商品 3.退出）：2
结算前，总价为：226980 ¥
恭喜您已获得优惠券码：COUPON117423，面值为：11351元
请选择是否使用优惠券（1.是 2.否）：1
请输入优惠券码：COUPON117423
优惠券码有效！已为您进行优惠
***开业大酬宾！！！有活动啦～～消费总额大于500元有优惠，已为您自动打9折！！！***
最终结算金额为：194066 ¥。购买成功，已从购物车中移除购买的商品。
```

库存更新：

===== 欢迎来到沈琨翔的购物车预约系统 =====

请输入您希望进行的操作

- | |
|-------------------|
| 1. 登陆 (管理员 or 顾客) |
| 2. 顾客注册 |
| 3. 预览商品 |
| 4. 退出 |

请通过输入前缀的数字以提供您的选择:

3

商品名称: AirPods_2

价格: 1799.99 元

描述: 苹果无线耳机 (库克的挤牙膏产品)

库存: 89 件

商品名称: Anker_Power_Bank

价格: 109 元

描述: 专业做电子产品的Anker, 充电宝质感一流

库存: 9 件

商品名称: Macbook

价格: 12999 元

描述: M2芯片, 你还在等什么!

库存: 1990 件

商品名称: iPhone_14

价格: 7899 元

描述: 拥有此产品你可以登岛!

库存: 90 件

结算之后是否从购物车中清空:

通过重新登录 skx 用户后观察其购物车以及购物车文件记录可以看到结算之后会进行数据

更新。

```

445 void voidSystem::checkoutAllItemsOfKind(const string& custom
501     auto itemIt = it->second.rfind(selectedItem);
502     int quantity = itemIt->second;
503
504     auto goodIt = goods.find(selectedItem);
505     if (goodIt == goods.end()) {
506         cout << "找不到商品 " << selectedItem << " 的价格信息";
507         continue;
508     }
509     Good& theGoods = goodIt->second;
510     if (!theGoods.checkStock(quantity)) {
511         cout << "库存不足, 无法结算 " << quantity << " 个 ";
512     }
513 }
```

请输入前面的数字以提供您的选择:

1. 输入用户类型(1. 管理员 2. 顾客): 2
输入用户名: skx
输入密码: 321
用户名登录成功
您要进行:
1. 查看商品
2. 查找商品
3. 打开购物车
4. 购买购物车内商品
5. 修改密码
6. 退出
7. 放弃操作
请选择操作 (1~7) : 3
您已打开购物车, 您需要:
1. 添加商品
2. 查看购物车
3. 删减商品
4. 修改商品数量
5. 查看指定时间段内商品购买金额
6. 退出购物车
请选择 (1~6) : 2
总价: 1799.99 元
您已打开购物车, 您需要:
1. 添加商品
2. 查看购物车
3. 删减商品
4. 修改商品数量
5. 查看指定时间段内商品购买金额
6. 退出购物车
请选择 (1~6) :

五、 实验总结

面向对象编程的应用

在这个部分，我学到了如何将现实世界中的实体抽象成类，然后根据它们的属性和行为来设计程序。比如，用户类、商品类、购物车类等，每个类都有自己的特性和行为，通过类的封装、继承和多态性，使得程序结构更加清晰和灵活。这种面向对象的编程方式让我更好地组织和管理程序代码，提高了代码的可读性和可维护性。

数据结构的选择与使用

在这个部分，我学会了选择合适的数据结构来存储和管理数据。比如，使用 map 来快速查找用户、商品和购物车信息，以及使用 unordered_map 来存储顾客的购物车数据。这些数据结构的选择使得系统操作更加高效和方便，同时也提高了程序的性能和稳定性。

文件操作

在这个部分，我掌握了如何在程序中进行文件读写操作，将数据持久化地保存到文件中，实现数据的长期存储。通过文件操作，我学会了如何格式化数据的保存和解析，保证了数据的准确性和完整性。文件操作让系统具备了存储和恢复数据的能力，提高了系统的可靠性和持久性。

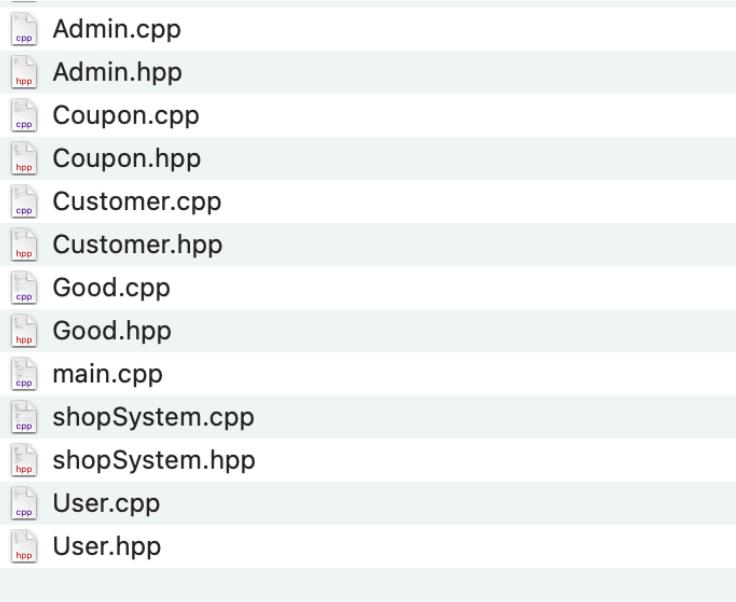
时间管理

在这个部分，我学到了如何使用 C++ 的时间库（chrono）来处理时间数据，实现时间的记录和查询。通过时间管理，我能够准确记录购买时间，并根据时间范围查询购买记录。时间的管理让系统具备了时间相关功能，增强了系统的实用性和功能性。

锻炼良好的大语言模型的使用习惯

大语言模型的出现确实改变了学生的学习习惯以及企业中的工作习惯，正确使用大语言模型可以提高学习效率、工作效率。

附录 源程序清单（要求注释不少于源代码行数的三分之一）



main.cpp: 是整个项目的入口文件,包含主函数(main function)。它负责初始化系统,调用其他类的方法,并控制整个程序的流程。

shopSystem.cpp 和 shopSystem.hpp: 包含购物系统(shopSystem)类的实现和声明。购物系统类是整个系统的核心,负责协调其他类的工作,实现整个购物流程的逻辑。

User.cpp 和 User.hpp: 包含用户(User)类的实现和声明。用户类是客户类和管理员类的基类,包含一些共享的属性和方法。

Admin.cpp 和 Admin.hpp: 包含管理员(Admin)类的实现和声明。管理员类负责管理系统的一些高级功能,如管理商品、用户等。

Customer.cpp 和 Customer.hpp: 包含客户(Customer)类的实现和声明。客户类用于管理客户的信息和操作,如客户的注册、登录、购物等。

Coupon.cpp 和 Coupon.hpp: 包含优惠券(Coupon)类的实现和声明。优惠券类用于处理系统中的优惠券功能,如优惠券的生成、使用等。

Good.cpp 和 Good.hpp: 包含商品(Good)类的实现和声明。商品类用于管理商品的信息,如商品的名称、价格、库存等。