**项目报告：线上交易系统**

## **项目地址：**

[Online-trading-system (github.com)](https://github.com/wanqian-windrunner/Online-trading-system)

**1.分组情况**

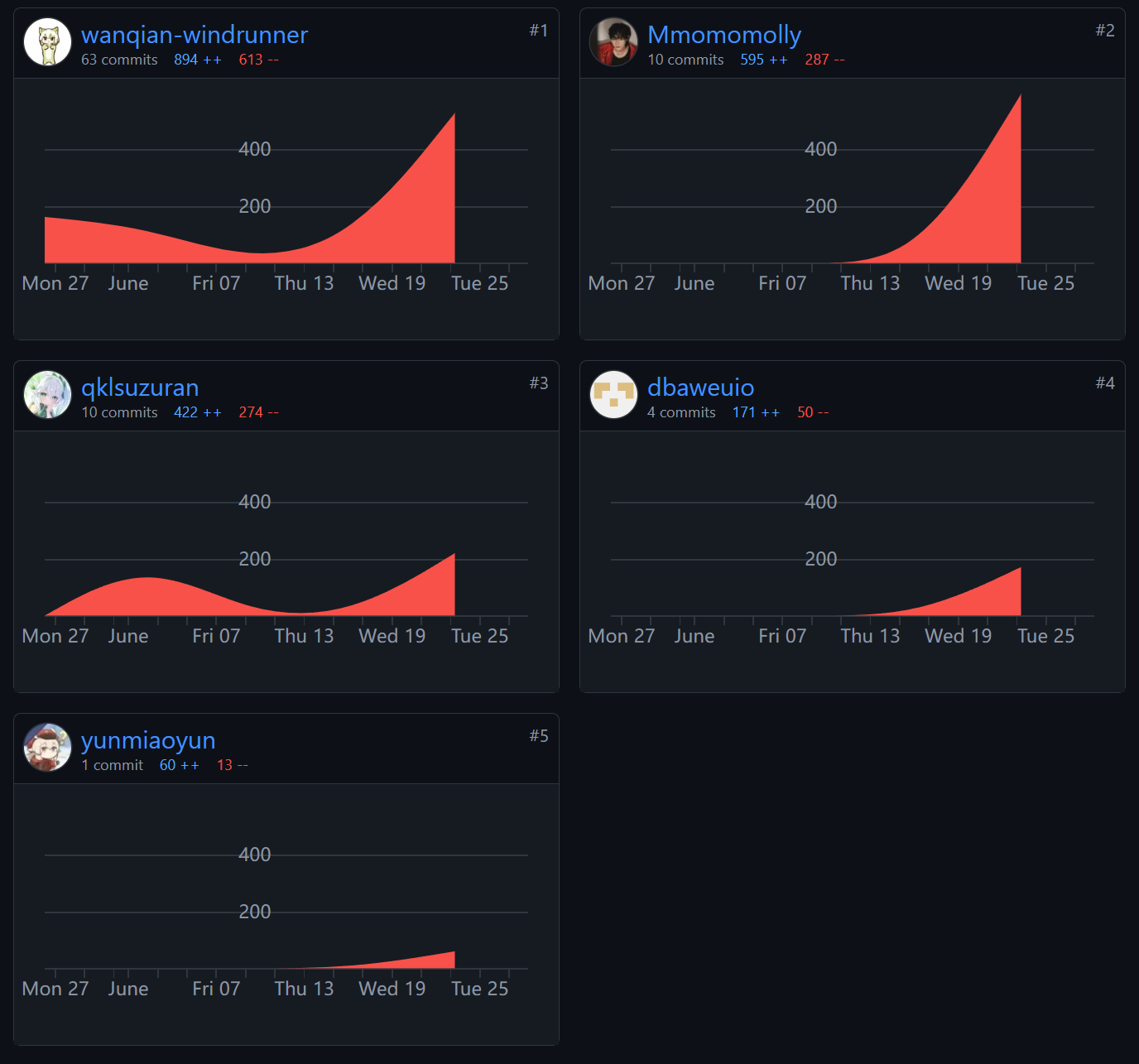
**组员名单及分工**

* **组长**：张腾
  + 主要负责项目总体架构设计、注册登录实现、后端数据处理、功能封装、bug维护。
* **组员1**: 侯景尧
  + 主要负责用户界面设计，购买售卖实现，登录验证实现。
* **组员2**: 孙羽墨
  + 负责购物车添加商品、结算功能实现。
* **组员3**: 石正宇
  + 负责系统评价功能实现。
* **组员4**: 董智嘉
  + 负责完善商品，项目测试。

**2.每名同学的参与度及其在组内的贡献**

[Contributors to wanqian-windrunner/Online-trading-system (github.com)](https://github.com/wanqian-windrunner/Online-trading-system/graphs/contributors?from=2024-05-26&to=2024-06-28&type=a)

依次为张腾、孙羽墨、侯景尧、石正宇、董智嘉



**3.项目提出过程**

项目的提出基于以下背景：

* 学习使用github进行多人协作
* 学习和掌握开发技能的需要。
* 实践软件工程中团队协作和项目管理的能力。

在讨论后，我们决定开发一个简单但功能完善的线上交易系统，功能包括用户注册登录、商品浏览、购物车、订单处理等。

**4.设计过程**

**系统架构设计**

我们采用了经典的MVC架构，将系统分为前端、后端和数据库三部分。

**功能模块设计**

1. **用户模块**：用户注册、登录、信息管理。
2. **商品模块**：商品浏览、详情展示。
3. **购物车模块**：添加商品到购物车、查看购物车、结算购物车。
4. **系统评价模块**：评价功能，评价浏览功能。

**数据库设计**

 **用户数据 (users)**:

* 包含多个用户的信息，每个用户有唯一的用户名和密码哈希值。
* 每个用户还有一个余额 (balance)，一个背包 (bag) 用于存储每种商品的数量，以及一个购物车 (cart) 用于临时存放要购买的商品数量。

 **商品数据 (goods)**:

* 包含多种商品的信息，每种商品有唯一的编号作为键。
* 每种商品有名称 (name)、价格 (price) 和库存数量 (count)。

**5.程序代码（部分模块略，详见**[Online-trading-system(github.com)](https://github.com/wanqian-windrunner/Online-trading-system)**）**

以下是主要功能模块的代码：

evaluatate.py

import utils\_else

# 在用户选择退出系统时执行此程序

class Evaluate\_system:

    def \_\_init\_\_(self,username):

        self.username = username

        # 读取系统原有数据

        self.dict = utils\_else.Data\_process.read()

    def score(self):

        # 如果用户输入不符合要求，重新执行此模块

        while True:

            #让用户输入对本系统的评星

            self.score=int(input('''

                               评星系统

                    你可以选择输入数字 1/2/3/4/5.

                  数字越大说明你对本系统的评价越高\n'''))

            #判断输入是否符合要求

            if  self.score == 1 or  self.score == 2 or  self.score == 3 or  \

                self.score == 4 or  self.score == 5:

                break

            #如果不符合要求就让用户重新输入

            else:

                print('''你输入的不是1/2/3/4/5,请重新输入''')

    def comment(self):

        #让用户输入对本系统的评价

        self.comment=input('''

                                评价系统

                      如果你有其他的建议，请在这里输入：

                      (如果你需要跳过，请直接按回车键)\n''')

    def submit(self):

         # 在程序中更改

        self.dict["score"]["count"] += 1

        self.dict["score"]["sum"] += self.score

        self.dict["score"]["average value"] = (self.dict["score"]["sum"])/(self.dict["score"]["count"])

        self.dict["comment"][self.username] = self.comment

        # 将数据保存在系统中

        utils\_else.Data\_process.write(self.dict)

    def show\_stars(self):

        self.average\_data = self.dict["score"]["average value"]

        #返回平均分的值

        print(f'该系统目前总分数为：{self.average\_data}，感谢您的评价！')

    def show\_comment(self):

        #返回用户的评价

        for i in self.dict["comment"]:

            print(i,":",self.dict["comment"][i])

if \_\_name\_\_ == '\_\_main\_\_':

    a = Evaluate\_system("user3")

    a.score()

    a.comment()

    a.submit()

    a.show\_stars()

    a.show\_comment()

login.py

import json

from utils import Data\_process

import hashlib

import random

class LoginSystem:

    def \_\_init\_\_(self):

        self.login\_status = False

        # 读取所有数据

        self.totaldata = Data\_process.read()

        # 提取用户数据

        self.user\_data = self.totaldata['users']

    def login(self):

        # 输入用户名和密码

        self.username = input("请输入您的用户名: ")

        self.password = input("请输入您的密码: ")

        # 打印输入的用户名和密码 (用于调试)

        # print(self.username, self.password)

        # 判断用户名和密码是否正确，正确则存储数据，否则提示错误，为了保证密码安全，我们使用hashlib库对密码进行加密

        if self.username in self.user\_data and self.user\_data[self.username]['password'] == hashlib.sha256(self.password.encode()).hexdigest():

            print("\033[32m登录成功！欢迎, \033[0m" + self.username + "\033[32m!\033[0m")

            self.login\_status = True

        else:

            print("\033[31m无效的用户名或密码。\033[0m")

    def register(self):

        # 输入新用户名

        self.username = input("请输入您的用户名: ")

        # 检查用户名是否已存在

        if self.username in self.user\_data:

            print("\033[34m用户名已存在！请尝试一个新名字或登录。\033[0m")

            return 0

        # 输入密码

        self.password = input("请输入您的密码: ")

        # 确认密码是否一致

        if input("请再次输入您的密码: ") != self.password:

            print('\033[31m抱歉，再见~\033[0m')

            return 0

        # 输入余额

        # self.balance = input("请输入您的余额: ")

        # 创建用户数据字典

        self.userdata = {

            'name': self.username,

            'password': hashlib.sha256(self.password.encode()).hexdigest(),

            'balance': 1000,

            'bag': {},

            'cart':{}

        }

        # 将新用户数据添加到总数据中

        self.totaldata['users'][self.username] = self.userdata

        # 将总数据写回文件

        Data\_process.write(self.totaldata)

        self.login\_status = True

    def logout(self):

        # 退出程序

        exit(0)

if \_\_name\_\_ == '\_\_main\_\_':

    # 创建登录系统实例

    login = LoginSystem()

    # 执行登录

    login.login()

    # 打印用户数据 (用于调试)

    print(login.userdata)

main.py

from menu import Menu

from login import LoginSystem

from utils import Data\_process

from evaluatate import Evaluate\_system

loginsystem=LoginSystem()

# 登陆账户，然后存储用户数据

while not loginsystem.login\_status:

    option = input("Choose your option:\n\t1.register\n\t2.login\n")

    if option == '1':

        loginsystem.register()

    elif option == '2':

        loginsystem.login()

Data\_process.data\_init()

userdata = Data\_process.read()['users'][loginsystem.username]

# print(userdata)

menusystem = Menu(loginsystem.login\_status,userdata)

menusystem.function\_menu()

evalutatesystem = Evaluate\_system(userdata['name'])

evalutatesystem.score()

evalutatesystem.comment()

evalutatesystem.submit()

evalutatesystem.show\_stars()

Shopping\_cart

import json

from utils import Data\_process, Showshowway

from trade import Trade

class Shopping\_cart:

    choice\_number = None

    def \_\_init\_\_(self, userdata):

        self.goods\_data = Data\_process.read()['goods']

        self.shopping\_cart = userdata['cart']

        self.goods\_count = userdata.read()['count']

    def add\_time(userdata):  # 加入个数

        goods = Data\_process.read()['goods']

        Showshowway.show\_goods()

        print()

        Shopping\_cart.choice\_number = input("Input which you want to add in?(q to quit):")

        if Shopping\_cart.choice\_number == 'q':

            return 0

        try:

            Shopping\_cart.choice\_number = int(Shopping\_cart.choice\_number)

        except :

            print('\nInvalid choice. Please try again.\n')

            Shopping\_cart.add\_time(userdata)

        if Shopping\_cart.choice\_number not in range(1,len(goods)+1):

            print('\nInvalid choice. Please try again.\n')

            Shopping\_cart.add\_time(userdata)

        Showshowway.show\_detail(str(Shopping\_cart.choice\_number))

        print()

        while True:

            add\_num = input("Please input how many you want to add (q to quit): ")

            print()

            if add\_num == "q":

                return 0

            try:

                if int(add\_num) <= int(goods[str(Shopping\_cart.choice\_number)]['count']):

                    print('\033[32mAdd Shopping Cart Successfully!\033[0m\n')

                    trade = Trade(userdata)

                    trade.cart(str(Shopping\_cart.choice\_number),int(add\_num))

                    break

                else:

                    print('\033[33mThe quantity you selected is greater than the quantity in stock.\033[0m\n'

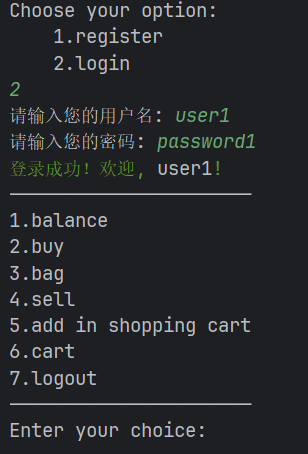
                        'Please try again.')

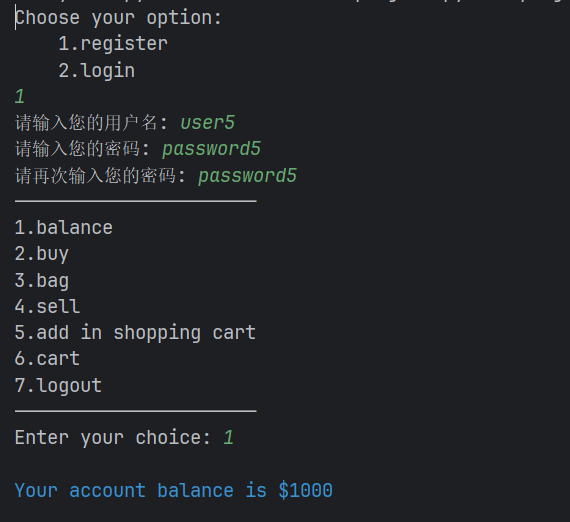
            except:

                print('\nInvalid choice. Please try again.\n')

                continue

**6.代码运行截图**





文本

描述已自动生成

文本

描述已自动生成

文本

描述已自动生成

文本

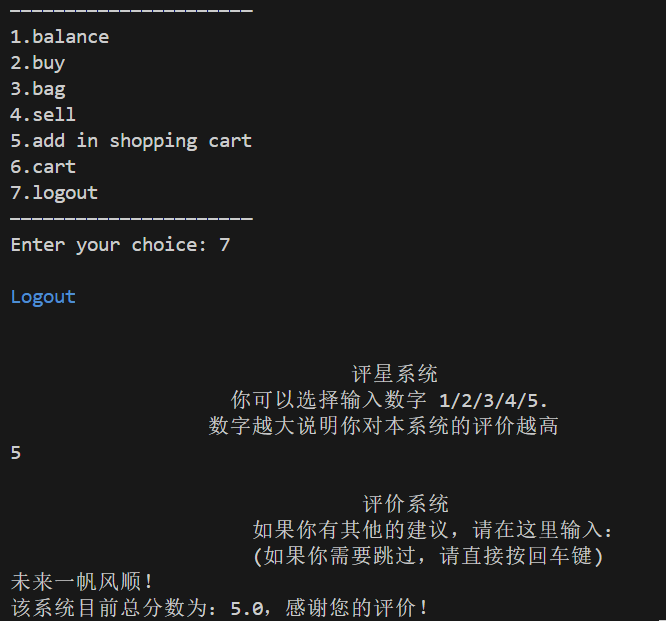
描述已自动生成

文本

描述已自动生成

文本

描述已自动生成



**7.总结**

在项目确定和落实的过程中，组内成员积极提出想法与建议，并在组长的带领下研究探讨模块化开发及git工具在项目开发中的应用方式。但在目标实现过程中也发现了许多问题，包括没有事先明确整体架构，导致为实现相应模块的功能，大面积更改其他模块内容，对效率以及成员心态产生了极大的影响。在未来我们将继续学习，继续进步，探寻实现目标、解决问题的高效率、高质量方式。

**8.项目未来改进**

1. 可以使用sql数据库来管理数据

2. 可以使用第三方库与css来实现更好看的前端

3. 增加单个商品的评论功能

4. 增加更多商品，并实现商品的检索功能

5.