A picture containing text

Description automatically generated

成绩：

**2022－2023第二学期**

**《Python语言程序设计》大作业**

|  |  |
| --- | --- |
| 大作业名称： | 基于Python的食堂管理系统的设计 |
| 姓 名： |  |
| 学 号： |  |
| 专 业： | 电子信息工程 |
| 班 级： | 电信21 - 班 |
| 任课教师签名： |  |
| 批阅日期： |  |

测控技术与通信工程学院

2023 年 6 月

# 一、大作业目的

1、掌握Python开发的基础知识和程序设计方法，培养学生的设计思维和设计能力；

2、能够应用Python基础知识对系统进行分析和合理设计。

# 二、大作业内容

假设你现在为某食堂开发一套管理系统，要求编程实现如下功能。

程序功能包含：

1. 用户身份设定。比如：用户身份可为食堂管理员、打饭的阿姨、吃饭的小白等。

(1) 食堂管理员：管理的账户和密码是唯一的

(2) 吃饭的小白：首次登录需要注册，设置登录账户和密码

2、用户登录系统后，显示食堂的基本信息，比如：名称(西一食堂)、地点(西区)、食堂或菜品的特色介绍等。

3、食堂管理员的权限可包括：当日菜品的种类和数量的设定、菜品的成本、销售价格的制定、计算当日食堂的盈利等；

4、打饭的阿姨的权限可包括：跟吃饭的小白确定所选择的饭菜、计算总价格并扣除饭费，饭卡资金不足的时候有提示等；

1. 吃饭的小白的权限可包括：选择今天想要吃的饭菜、确定饭卡内的资金是否足够、

对菜品的评价等。

注：以上5点基本功能要实现，程序在满足基本功能的基础上可以自行扩展。

测控技术与通信工程学院 电子信息工程系

Python语言程序设计 课程组

2023/06

# 基于Python的食堂管理系统的设计

# 1、系统功能需求分析

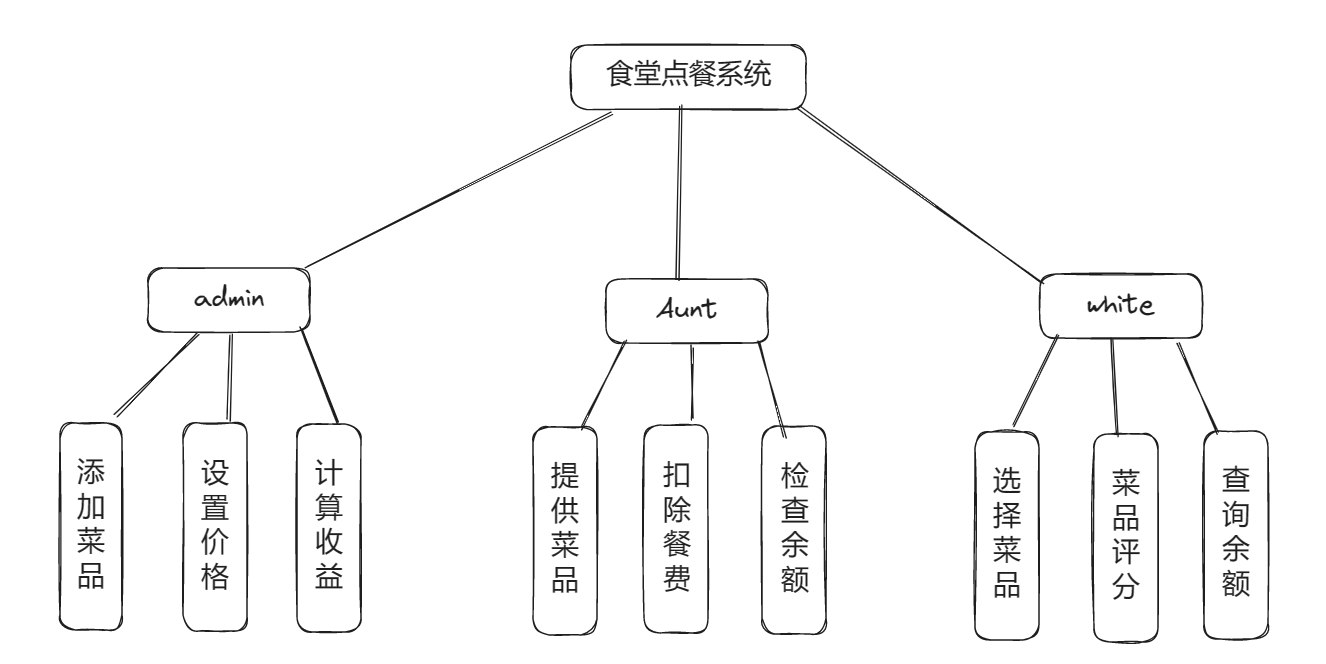
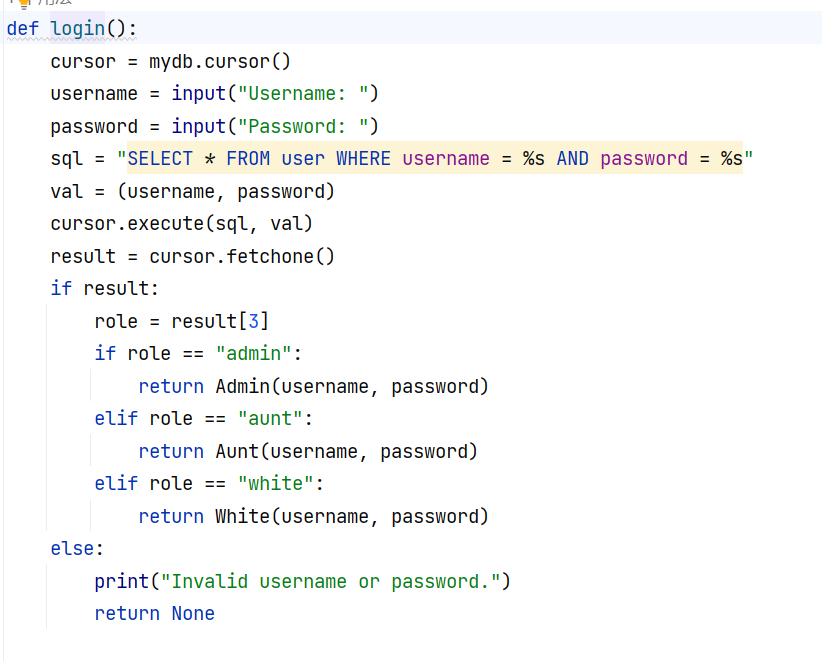


图1：点餐系统功能图

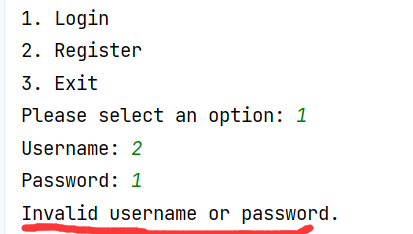
# 2、具体实践内容以及程序涉及到的知识点

1. 程序功能包含的不同用户的注册，各个用户不同功能和权限，实现对菜品的添加，实现对菜品的选择和评分等功能
2. 使用了python的类继承，mysql数据库等知识来实现各种功能。
3. 用户登入函数：

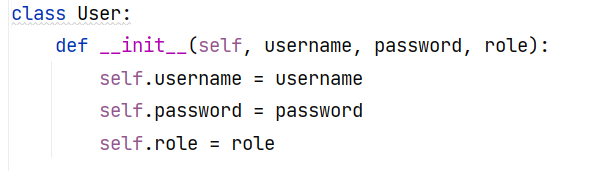
用户从控制台输入用户名和密码，程序从数据库读取数据并检查用户信息，不存在会提示



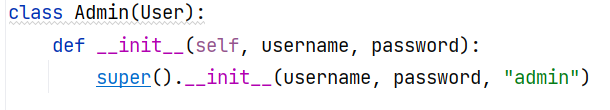
错误提示：



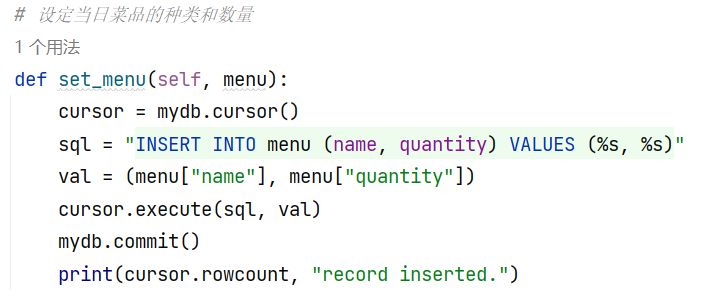
1. 用户类：初始化用户属性



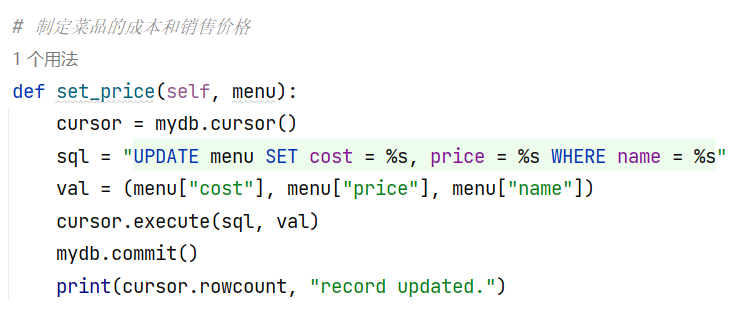
1. 管理员类，具体化管理员的属性和功能函数



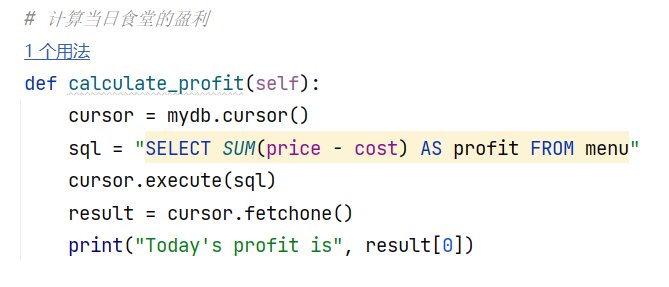
设置菜单：



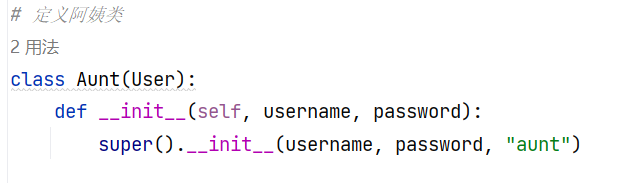
设置价格：



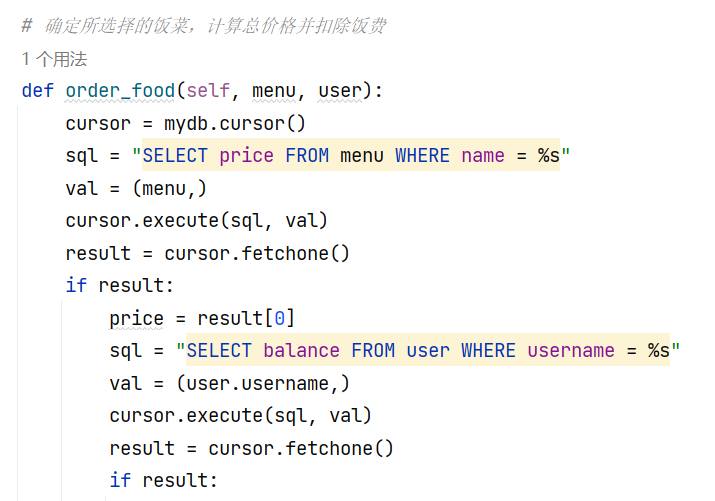
计算收益;



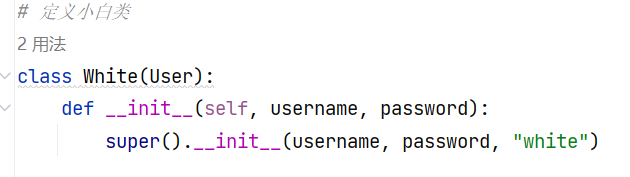
1. 定义打菜阿姨类



从数据库中检查用户余额是否充足来决定提供菜品



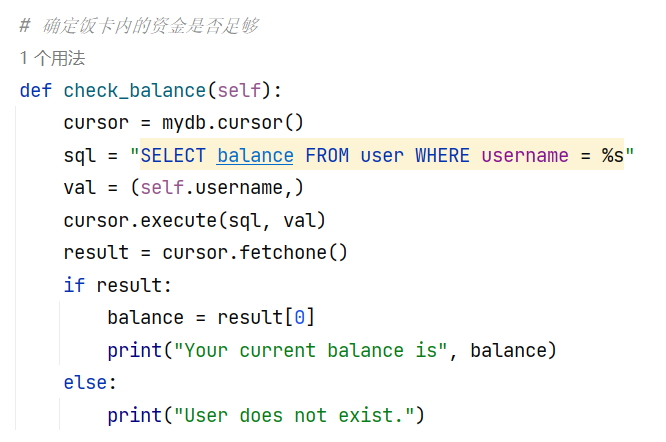
1. 定义小白类



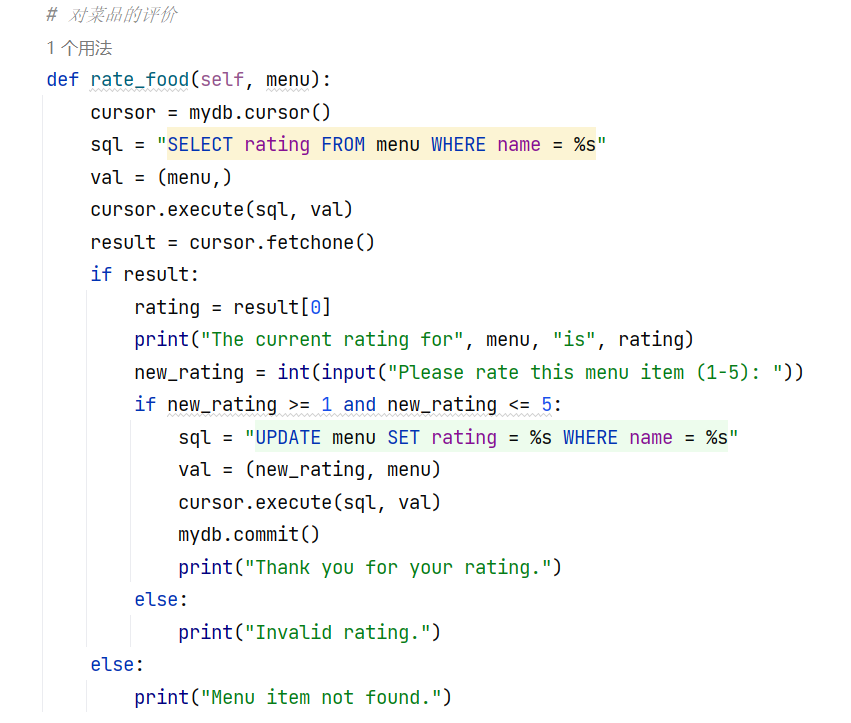
选择菜品：



查询饭卡余额：

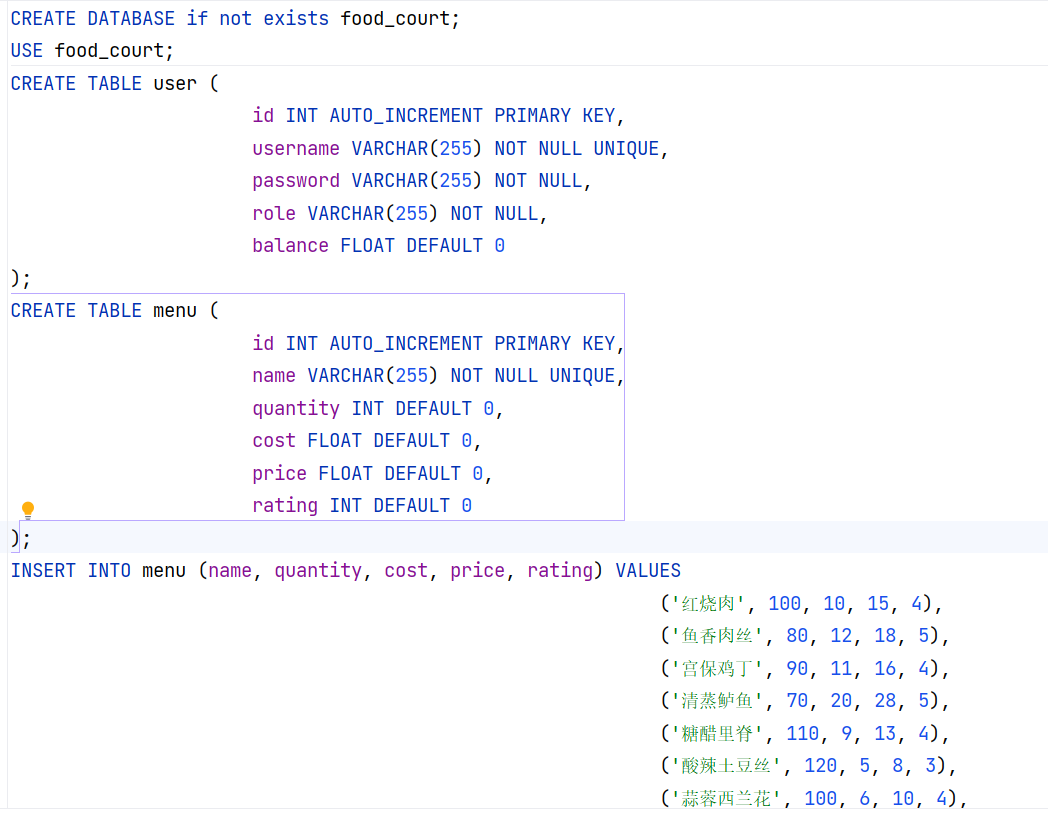


生成菜品评价并保存在数据库

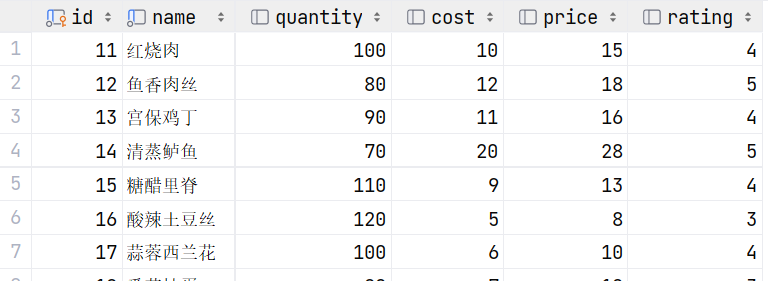


1. 数据库语句：

建表语句

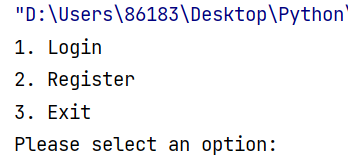


1. 表结构

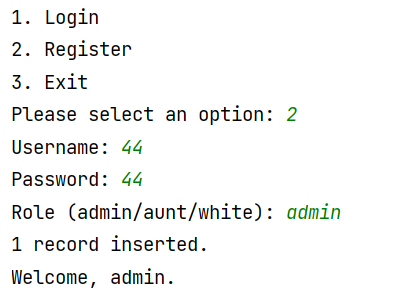


# 3、程序调试过程及运行结果

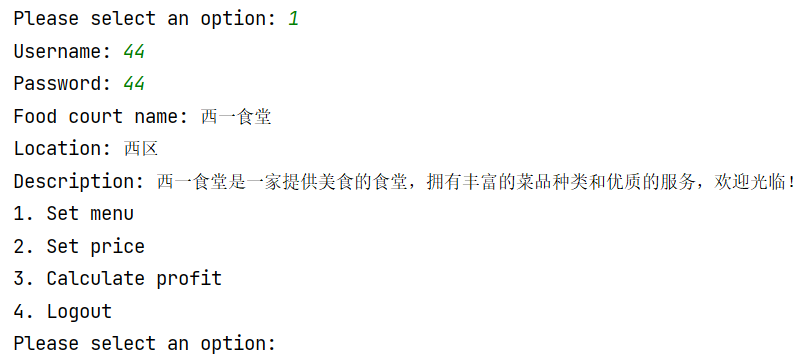
1. 登入界面：



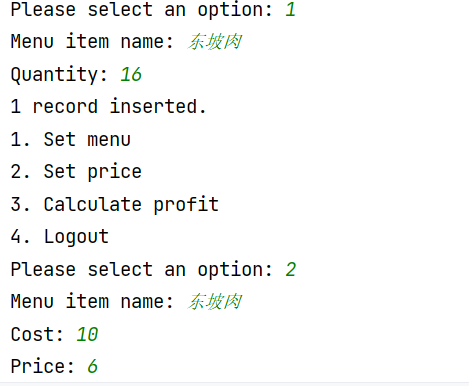
1. 选择注册;



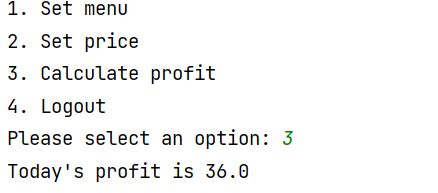
1. 实现登入：



1. 设置菜品和价格：



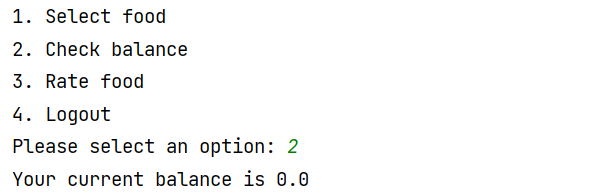
1. 计算收益：



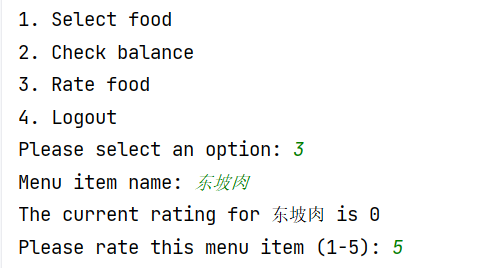
1. 给用户提供商品



1. 查询余额



1. 菜品评分



# 4、附录：原程序

#coding=utf-8

import mysql.connector

# 连接数据库

mydb = mysql.connector.connect(

host="localhost",

user="root",

password="123456",

database="food\_court"

)

# 食堂基本信息

food\_court\_info = {

"name": "西一食堂",

"location": "西区",

"description": "西一食堂是一家提供美食的食堂，拥有丰富的菜品种类和优质的服务，欢迎光临！"

}

# 定义用户类

class User:

def \_\_init\_\_(self, username, password, role):

self.username = username

self.password = password

self.role = role

# 定义管理员类

class Admin(User):

def \_\_init\_\_(self, username, password):

super().\_\_init\_\_(username, password, "admin")

# 设定当日菜品的种类和数量

def set\_menu(self, menu):

cursor = mydb.cursor()

sql = "INSERT INTO menu (name, quantity) VALUES (%s, %s)"

val = (menu["name"], menu["quantity"])

cursor.execute(sql, val)

mydb.commit()

print(cursor.rowcount, "record inserted.")

# 制定菜品的成本和销售价格

def set\_price(self, menu):

cursor = mydb.cursor()

sql = "UPDATE menu SET cost = %s, price = %s WHERE name = %s"

val = (menu["cost"], menu["price"], menu["name"])

cursor.execute(sql, val)

mydb.commit()

print(cursor.rowcount, "record updated.")

# 计算当日食堂的盈利

def calculate\_profit(self):

cursor = mydb.cursor()

sql = "SELECT SUM(price - cost) AS profit FROM menu"

cursor.execute(sql)

result = cursor.fetchone()

print("Today's profit is", result[0])

# 定义阿姨类

class Aunt(User):

def \_\_init\_\_(self, username, password):

super().\_\_init\_\_(username, password, "aunt")

# 确定所选择的饭菜，计算总价格并扣除饭费

def order\_food(self, menu, user):

cursor = mydb.cursor()

sql = "SELECT price FROM menu WHERE name = %s"

val = (menu,)

cursor.execute(sql, val)

result = cursor.fetchone()

if result:

price = result[0]

sql = "SELECT balance FROM user WHERE username = %s"

val = (user.username,)

cursor.execute(sql, val)

result = cursor.fetchone()

if result:

balance = result[0]

if balance >= price:

balance -= price

sql = "UPDATE user SET balance = %s WHERE username = %s"

val = (balance, user.username)

cursor.execute(sql, val)

mydb.commit()

print("Order placed successfully. Your balance is", balance)

else:

print("Insufficient balance. Please recharge your card.")

else:

print("User does not exist.")

else:

print("Menu item not found.")

# 定义小白类

class White(User):

def \_\_init\_\_(self, username, password):

super().\_\_init\_\_(username, password, "white")

# 选择今天想要吃的饭菜

def select\_food(self):

cursor = mydb.cursor()

sql = "SELECT name FROM menu"

cursor.execute(sql)

result = cursor.fetchall()

print("Today's menu:")

for menu in result:

print(menu[0])

food = input("Please select a menu item: ")

return food

# 确定饭卡内的资金是否足够

def check\_balance(self):

cursor = mydb.cursor()

sql = "SELECT balance FROM user WHERE username = %s"

val = (self.username,)

cursor.execute(sql, val)

result = cursor.fetchone()

if result:

balance = result[0]

print("Your current balance is", balance)

else:

print("User does not exist.")

# 对菜品的评价

def rate\_food(self, menu):

cursor = mydb.cursor()

sql = "SELECT rating FROM menu WHERE name = %s"

val = (menu,)

cursor.execute(sql, val)

result = cursor.fetchone()

if result:

rating = result[0]

print("The current rating for", menu, "is", rating)

new\_rating = int(input("Please rate this menu item (1-5): "))

if new\_rating >= 1 and new\_rating <= 5:

sql = "UPDATE menu SET rating = %s WHERE name = %s"

val = (new\_rating, menu)

cursor.execute(sql, val)

mydb.commit()

print("Thank you for your rating.")

else:

print("Invalid rating.")

else:

print("Menu item not found.")

# 用户登录函数

def login():

cursor = mydb.cursor()

username = input("Username: ")

password = input("Password: ")

sql = "SELECT \* FROM user WHERE username = %s AND password = %s"

val = (username, password)

cursor.execute(sql, val)

result = cursor.fetchone()

if result:

role = result[3]

if role == "admin":

return Admin(username, password)

elif role == "aunt":

return Aunt(username, password)

elif role == "white":

return White(username, password)

else:

print("Invalid username or password.")

return None

# 用户注册函数

def register():

cursor = mydb.cursor()

username = input("Username: ")

password = input("Password: ")

role = input("Role (admin/aunt/white): ")

sql = "INSERT INTO user (username, password, role) VALUES (%s, %s, %s)"

val = (username, password, role)

cursor.execute(sql, val)

mydb.commit()

print(cursor.rowcount, "record inserted.")

if role == "admin":

return Admin(username, password)

elif role == "aunt":

return Aunt(username, password)

elif role == "white":

return White(username, password)

# 主函数

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

while True:

print("1. Login")

print("2. Register")

print("3. Exit")

choice = input("Please select an option: ")

if choice == "1":

user = login()

if user:

# 显示食堂基本信息

print("Food court name:", food\_court\_info["name"])

print("Location:", food\_court\_info["location"])

print("Description:", food\_court\_info["description"])

if user.role == "admin":

while True:

print("1. Set menu")

print("2. Set price")

print("3. Calculate profit")

print("4. Logout")

choice = input("Please select an option: ")

if choice == "1":

name = input("Menu item name: ")

quantity = int(input("Quantity: "))

menu = {"name": name, "quantity": quantity}

user.set\_menu(menu)

elif choice == "2":

name = input("Menu item name: ")

cost = float(input("Cost: "))

price = float(input("Price: "))

menu = {"name": name, "cost": cost, "price": price}

user.set\_price(menu)

elif choice == "3":

user.calculate\_profit()

elif choice == "4":

break

else:

print("Invalid choice.")

elif user.role == "aunt":

while True:

print("1. Order food")

print("2. Logout")

choice = input("Please select an option: ")

if choice == "1":

menu = input("Menu item name: ")

user.order\_food(menu, user)

elif choice == "2":

break

else:

print("Invalid choice.")

elif user.role == "white":

while True:

print("1. Select food")

print("2. Check balance")

print("3. Rate food")

print("4. Logout")

choice = input("Please select an option: ")

if choice == "1":

food = user.select\_food()

print("You have selected", food)

elif choice == "2":

user.check\_balance()

elif choice == "3":

menu = input("Menu item name: ")

user.rate\_food(menu)

elif choice == "4":

break

else:

print("Invalid choice.")

elif choice == "2":

user = register()

if user:

if user.role == "admin":

print("Welcome, admin.")

elif user.role == "aunt":

print("Welcome, aunt.")

elif user.role == "white":

print("Welcome, white.")

elif choice == "3":

break

else:

print("Invalid choice.")