Catering Services Application

By Yoshihiro Nakagawasai
Coder Academy 2023

Introduction

“Catering Services for your party” This is an application that is a food catering planning guide for users.

The final quote changes depending on the user’s selection, and the user can confirm their selection on the final screen.

It is a requirement to accept user input and produce printed output and interact with file system.

Features

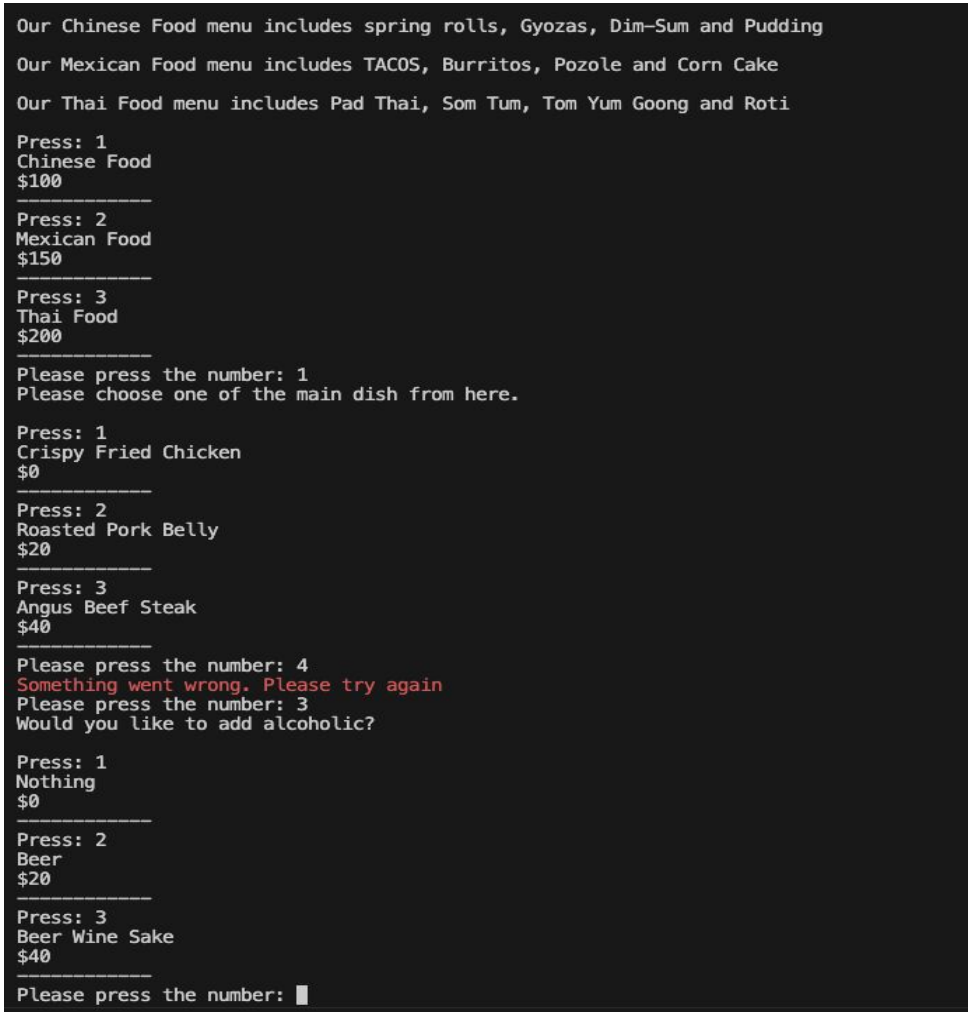
- Display menu and price list
- Menu Selection
- Exit or Retry
- Main Function

Feature 1: Display menu and price list

The title will be displayed as soon as the application starts. A price list of the plan will be displayed along with a menu explanation of the catering service.

Feature 2: Menu Selection

Here the user selects the desired menu plan by pressing the number. After that, the user keep selecting a main dish and drinks, and the price is calculated according to the user chosen menu.
If the user makes a wrong selection, the system is designed to display "Something wrong. Please try again" and then retry.



Feature 3: Exit or Retry

The final screen of the application will display a final selection. The user can retry from the beginning by selecting "y". Conversely, exit the application by selecting "n".

Feature 4: Main Function

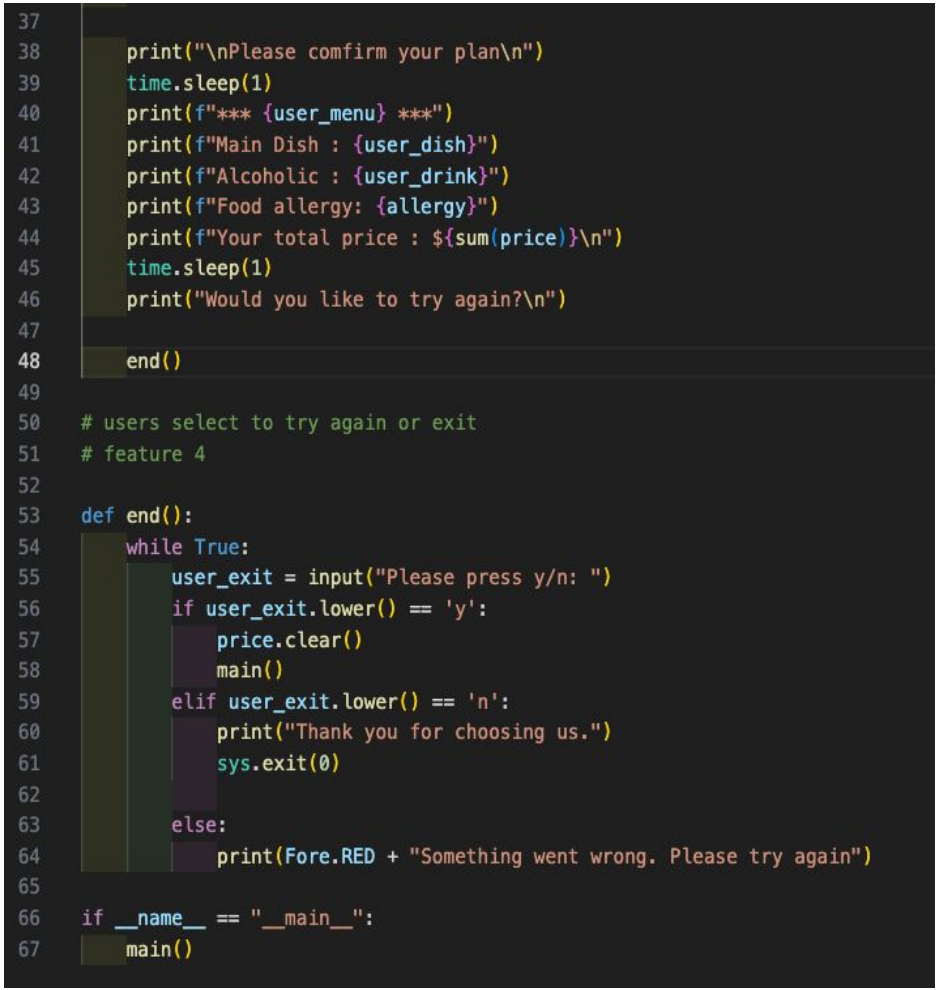
This feature is the core of this application.
Most functions are called from this Main Function, and the output and input screens are displayed alternately, the use view the menus and select menus comfortably.
A walk-through of my application code from here.

src > main.py > main

```python
import sys
import colorama
from colorama import Fore
from functions import (price, welcome, show_plan, show_dish,
                       show_drink, select_plan, select_dish, select_drink)
import time
colorama.init(autoreset=True)


# Main

def main():
    welcome()
    show_plan()

    # Choose a plan

    user_menu = select_plan()
    print("Please choose one of the main dish from here.\n")

    show_dish()

    # Choose a main dish

    user_dish = select_dish()


    print("Would you like to add alcoholic?\n")
    show_drink()

    # Choose a drink

    user_drink = select_drink()
```

A walk-through of application code

The code is designed with emphasis on readability and understandability. Finalliy, this main function displays the menu the user has selected and at the same time displays the total price.



```
37
38      print("\nPlease comfirm your plan\n")
39      time.sleep(1)
40      print(f"*** {user_menu} ***")
41      print(f"Main Dish : {user_dish}")
42      print(f"Alcoholic : {user_drink}")
43      print(f"Food allergy: {allergy}")
44      print(f"Your total price : ${sum(price)}\n")
45      time.sleep(1)
46      print("Would you like to try again?\n")
47
48      end()
49
50  # users select to try again or exit
51  # feature 4
52
53  def end():
54      while True:
55          user_exit = input("Please press y/n: ")
56          if user_exit.lower() == 'y':
57              price.clear()
58              main()
59          elif user_exit.lower() == 'n':
60              print("Thank you for choosing us.")
61              sys.exit(0)
62
63          else:
64              print(Fore.RED + "Something went wrong. Please try again")
65
66  if __name__ == "__main__":
67      main()
```

A walk-through of functions code

The menu and price list are designed using a dictionary, and the functions below are arranged to make it easy to understand what is displayed and how.

An empty list of price is defined.

functions.py

```python
# import main
import pyfiglet
import time
from colorama import Fore

# Calculate total price

price = []

# Plans
# There are 3 types of menu which include id, menu and price
# It is easy to see, add and remove items
# These prices and name of menus are linked other functions
# So just changing here will be reflected in other functions
# And also there are main dish menu and drink menu as well

plan1 = {"id": 1, "menu": "Chinese Food", "price": 100}
plan2 = {"id": 2, "menu": "Mexican Food", "price": 150}
plan3 = {"id": 3, "menu": "Thai Food", "price": 200}


def plan(id, menu, price):
    print(f"Press: {id}")
    print(menu)
    print(f"${price}")
    print("----------")

# Main dishes

dish1 = {"id": 1, "menu": "Crispy Fried Chicken", "price": 0}
dish2 = {"id": 2, "menu": "Roasted Pork Belly", "price": 20}
dish3 = {"id": 3, "menu": "Angus Beef Steak", "price": 40}


def dish(id, menu, price):
    print(f"Press: {id}")
    print(menu)
    print(f"${price}")
    print("----------")


```

A walk-through of functions code

- import pyfiglet
- import colorma
- import time

functions.py

```python
drink1 = {"id": 1, "menu": "Nothing", "price": 0}
drink2 = {"id": 2, "menu": "Beer", "price": 20}
drink3 = {"id": 3, "menu": "Beer Wine Sake", "price": 40}


def drink(id, menu, price):
    print(f"Press: {id}")
    print(menu)
    print(f"${price}")
    print("------------")


# This function expresses the title of this application
# And also it shows the 3 types of details of the menus
# feature 1

def welcome():
    print(pyfiglet.figlet_format("Catering services for your party"))
    print(Fore.LIGHTGREEN_EX +"Please check our catering menus and price list\n"
        "After selecting menus, you can check the total price\n")
    print(f"Our {plan1['menu']} menu includes spring rolls, Gyozas, Dim-Sum and Pudding\n")
    print(f"Our {plan2['menu']} menu includes TACOS, Burritos, Pozole and Corn Cake\n")
    print(f"Our {plan3['menu']} menu includes Pad Thai, Som Tum, Tom Yum Goong and Roti\n")

# feature 1

def show_plan():
    time.sleep(.5)
    plan(**plan1)
    plan(**plan2)
    plan(**plan3)

def show_dish():
    time.sleep(.5)
    dish(**dish1)
    dish(**dish2)
    dish(**dish3)

def show_drink():
    time.sleep(.5)
    drink(**drink1)
```

A walk-through of functions code

Loops also have a simple design and easy-to-understand structure, allowing users to retry after a red message warns them of any input errors.

functions.py

functions.py > ...

```python
    drink(**drink3)

# Plans
# Users select one of the menu by pressing the number [1 or 2 or 3]
# feature 2

def select_plan():

    while True:
        user_num = input("Please press the number: ")
        if user_num == "1":
            price.append(plan1["price"])
            return plan1["menu"]
        elif user_num == "2":
            price.append(plan2["price"])
            return plan2["menu"]
        elif user_num == "3":
            price.append(plan3["price"])
            return plan3["menu"]
        else:
            print(Fore.RED + "Something went wrong. Please try again")

# Main dish
# Users select one of the main dish by pressing the number [1 or 2 or 3]

def select_dish():

    while True:
        user_num = input("Please press the number: ")
        if user_num == "1":
            price.append(dish1["price"])
            return dish1["menu"]
        elif user_num == "2":
            price.append(dish2["price"])
            return dish2["menu"]
        elif user_num == "3":
            price.append(dish3["price"])
            return dish3["menu"]
        else:
            print(Fore.RED + "Something went wrong. Please try again")
```

A walk-through of functions code

It receives the user's allergy input and finally displays it along with the user's selections.

main.py > ...

```
30      show_drink()
31
32      # Choose a drink
33
34      user_drink = select_drink()
35
36      allergy = input("Please fill out your food allergies if you have:\n")
37
38      print("\nPlease comfirm your plan\n")
39      time.sleep(1)
40      print(f"*** {user_menu} ***")
41      print(f"Main Dish : {user_dish}")
42      print(f"Alcoholic : {user_drink}")
43      print(f"Food allergy: {allergy}")
44      print(f"Your total price : ${sum(price)}\n")
45      time.sleep(1)
46      print("Would you like to try again?\n")
47
48      end()
49
50  # users select to try again or exit
51  # feature 4
52
53  def end():
54      while True:
55          user_exit = input("Please press y/n: ")
56          if user_exit.lower() == 'y':
57              price.clear()
58              main()
59          elif user_exit.lower() == 'n':
60              print("Thank you for choosing us.")
61              sys.exit(0)
62
63          else:
64              print(Fore.RED + "Something went wrong. Please try again")
65
66  if __name__ == "__main__":
67      main()
```

A review of my development

A particularly challenging part of building the application was testing. Initially, I wrote the main function and other functions in the same file, but I faced many errors during testing. Based on that experience, I recoded the main function and other functions separately to deal with any errors. This experience will help you focus on the importance of thinking about functions separately the next time you start coding, and write code that is easy to read and understand.