

Reservation System in Python

Documentation

Author: Michał Deć

Nowy Sącz, 2024

1. Program Scope

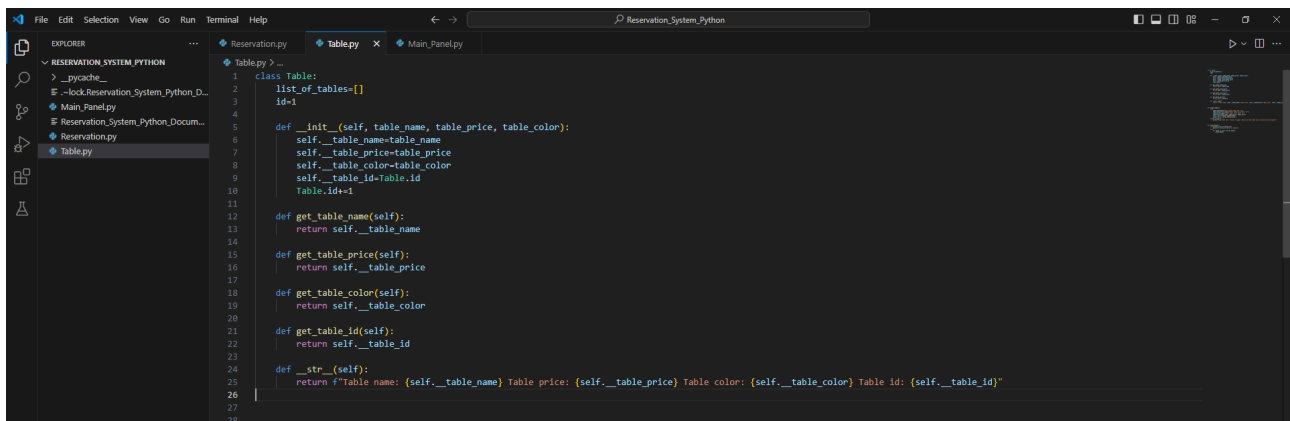
We will code simple application in Python. The goal of application is to map reservation system. We will have three modules. In module named „Main_Panel” we will have control panel to create table, reserve table, view table, and view reservations. In module „Table” there will be class named „Table” to create new table. In module „Reservation” there will be class named „Reservation” to order a table. Written application will also let to control basic exceptions that may rise during program execution. Application has been written with the use of Python 3.12.3 version in Visual Studio Code IDE.

2. Source Code

2.1 Table Module

2.1.1 Class Table

We have class „Table” inside „Table” module. This class allows us to create new table. Every table has his name, price, and color. Inside class we have also defined Accessors to return object attribute. What is more, inside class we have defined table id for every table. On the beginning it is set to 1 and with every new created table it is increased by one. Every table is saved to list defined on the class scope.

A screenshot of a code editor window titled 'Reservation_System_Python'. The editor shows a Python file named 'table.py' with the following code:

```
1 class Table:
2     list_of_tables=[]
3     id=1
4
5     def __init__(self, table_name, table_price, table_color):
6         self.__table_name=table_name
7         self.__table_price=table_price
8         self.__table_color=table_color
9         self.__table_id=Table.id
10        Table.id+=1
11
12    def get_table_name(self):
13        return self.__table_name
14
15    def get_table_price(self):
16        return self.__table_price
17
18    def get_table_color(self):
19        return self.__table_color
20
21    def get_table_id(self):
22        return self.__table_id
23
24    def __str__(self):
25        return f"Table name: {self.__table_name} Table price: {self.__table_price} Table color: {self.__table_color} Table id: {self.__table_id}"
26
27
28
```

The Explorer panel on the left shows the project structure with files like 'Reservation_System_Python.py', 'Main_Panel.py', and 'Table.py'.

2.1.2 Function create_table()

Function „create_table()” written inside „Table” module allows for creating new table. We enter table price, name, and color. If data is correct, object constructor from „Table” class is initiated and table object is created, and added to table list stored inside „Table” class.

```
28
29 def create_table():
30     try:
31         table_name=input("Enter table name here: \n")
32         table_price=float(input("Enter table price here: \n"))
33         table_color=input("Enter table color here: \n")
34         table=Table(table_name, table_price, table_color)
35         table.list_of_tables.append(table)
36         print("Table created successfully")
37     except ValueError:
38         print("Wrong table data. Please try again. Make sure that table price consists only of digits")
39
40
```

2.1.3 Function view_tables()

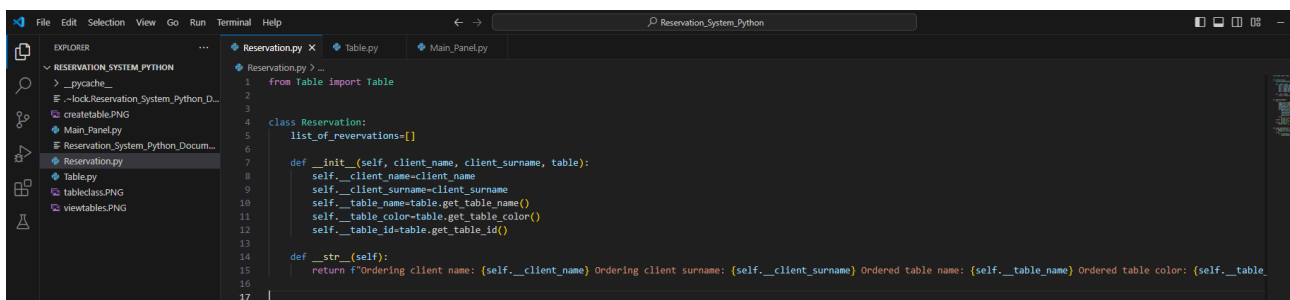
Function „view_tables()” written inside „Table” module allows for listing all available tables from table list stored inside „Table” class. If length of table list is zero, so there are no available tables at the moment, we are notified that there are no tables at the moment. Otherwise, this function lists all table objects from the list.

```
40
41
42 def view_tables():
43     if len(table.list_of_tables)==0:
44         print("No available tables at moment")
45     else:
46         for table in table.list_of_tables:
47             print(table)
48
49
50
```

2.2 Reservation Module

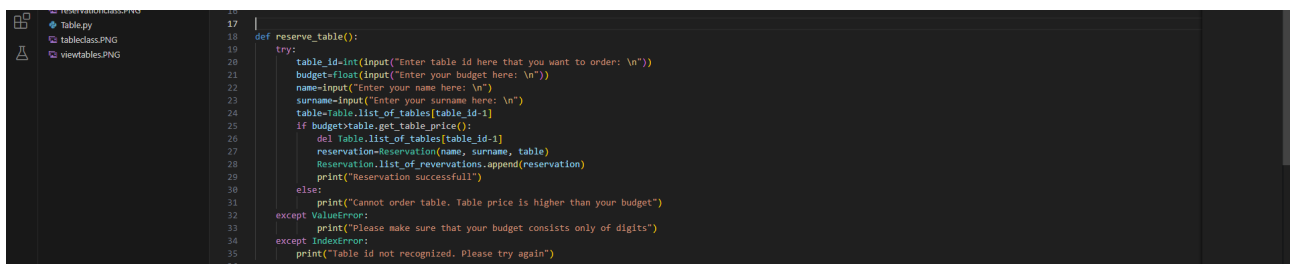
2.2.1 Class Reservation

Inside „Reservation” module we have defined „Reservation” class. This class allows for creating new objects as reservations. What is more, we store reservation list at the class scope. Inside this class we also override string method to print all object’s attributes.



2.2.2 Function reserve_table()

Function „reserve_table()” inside „Reservation” module allows us to reserve a table. We reserve table with the use of table id. Every table has his id, which is auto incremented every time we create table object. We reserve table by his id. To do it we appeal to table list index. For example table one is stored at position 0 inside list. To reserve this table we use [id-1] because table one has id 1, so we need to get access to position 0. Moreover, to reserve a table client’s budget has to be higher than table price. If reservation is successful, table is deleted from table list and transferred to reservation list.



2.2.3 Function view_reservations()

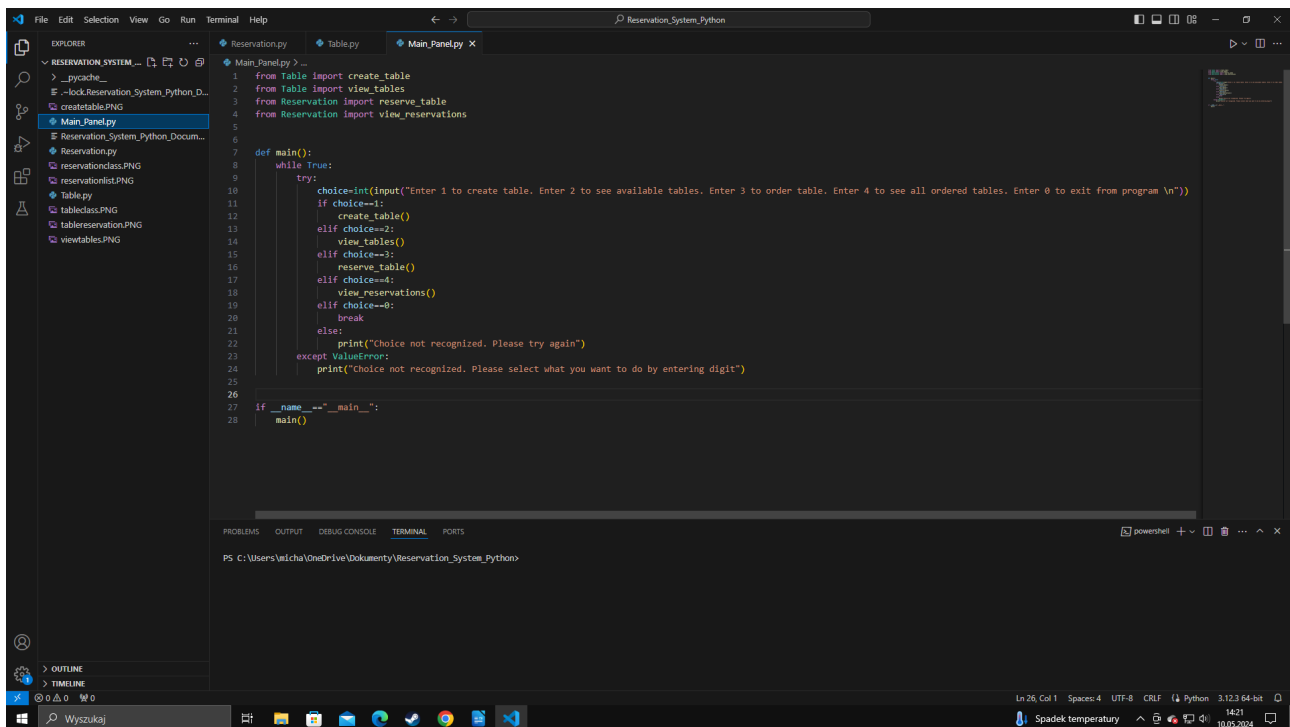
Function „view_reservations()” written inside „Reservation” module allows us to view all reserved tables. If length of reservation list is 0, it means that there are no reservations at the time. Otherwise, it lists all reservations from reservation list stored inside „Reservation” class.

```
37
38 def view_reservations():
39     if len(Reservation.list_of_reservations)==0:
40         print("No reservations at current time")
41     else:
42         for reservation in Reservation.list_of_reservations:
43             print(reservation)
```

2.3 Main_Panel module

2.3.1 Function main()

Function „main()” written inside „Main_Panel” module maps control panel. It allows us to either create new table, view all available tables, create new reservation, and view all reservations. We achieve that by implementing basic if and elif conditions.



```
File Edit Selection View Go Run Terminal Help
Reservation_System_Python
Main_Panel.py
1 from Table import create_table
2 from Table import view_tables
3 from Reservation import reserve_table
4 from Reservation import view_reservations
5
6
7 def main():
8     while True:
9         try:
10             choice = int(input("Enter 1 to create table. Enter 2 to see available tables. Enter 3 to order table. Enter 4 to see all ordered tables. Enter 0 to exit from program \n"))
11             if choice == 1:
12                 create_table()
13             elif choice == 2:
14                 view_tables()
15             elif choice == 3:
16                 reserve_table()
17             elif choice == 4:
18                 view_reservations()
19             elif choice == 0:
20                 break
21             else:
22                 print("Choice not recognized. Please try again")
23         except ValueError:
24             print("Choice not recognized. Please select what you want to do by entering digit")
25
26
27 if __name__ == "__main__":
28     main()
29
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\wlcha\OneDrive\Documents\Reservation_System_Python>

Ln 26, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit

Spidek temperature 14:21 10.05.2024

3. Resources used for program implementation

- 1) Python: Python version 3.12.3 - <https://www.python.org/downloads/>
- 2) IDE: Visual Studio Code - <https://code.visualstudio.com/>

List of contents

1. Program Scope.....	2
2. Source Code.....	3
2.1 Table Module.....	3
2.1.1 Class Table.....	3
2.1.2 Function create_table().....	4
2.1.3 Function view_tables().....	4
2.2 Reservation Module.....	5
2.2.1 Class Reservation.....	5
2.2.2 Function reserve_table().....	5
2.2.3 Function view_reservations().....	6
2.3 Main_Panel module.....	7
2.3.1 Function main().....	7
3. Resources used for program implementation.....	8