Software Testing and QA

Activity 6.2 Hotel Reservation System

Luis Sebastián Gan Cadena

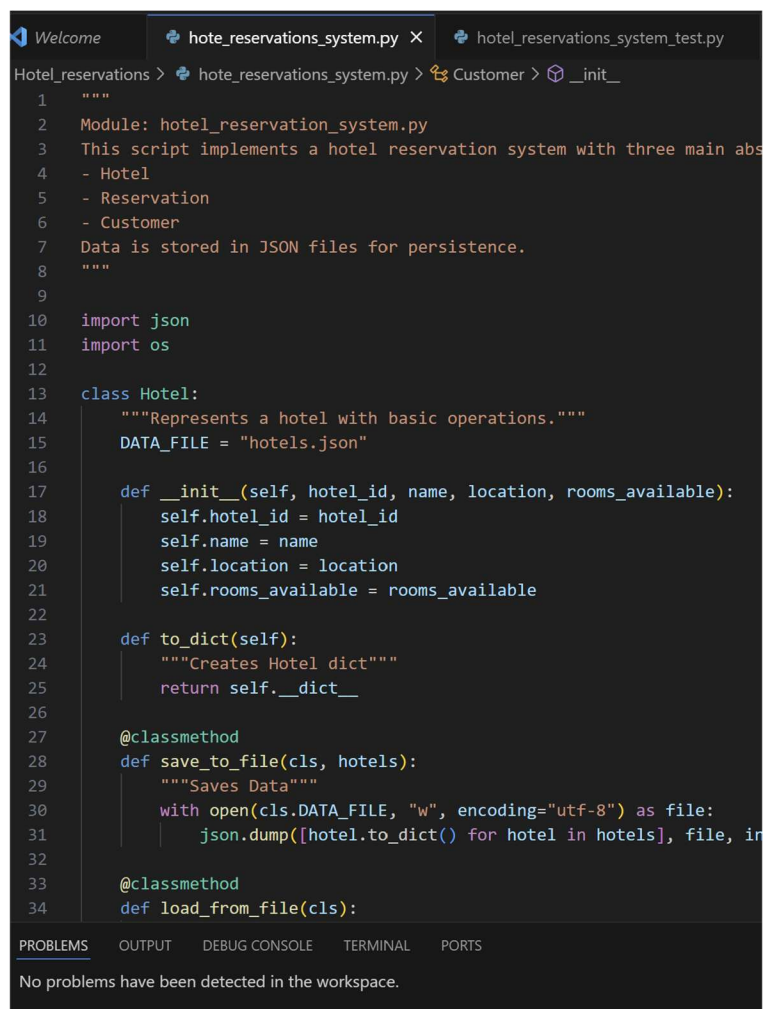A01186635

Building the Reservation System

First it was created the python file with the required objects. Ensuring PEP-8 alignment with pylint extension in VSCode.

The Hotel object was created with basic properties such as name, id, location, available rooms, and some basic methods such as save and load data (in json format), Create, delete and display.

The Customer object was created with the properties such as name, id and email, and some basic methods such as save and load data (in json format), create a dictionary with its data, and create.

For Reservation Object we have the same methods as customer and the properties are id, hotel id and customer id, being the object that brings together customer and hotel.

Here we can see a snapshot of the code where is being compliant with pylint extension.

```python
"""
Module: hotel_reservation_system.py
This script implements a hotel reservation system with three main abs
- Hotel
- Reservation
- Customer
Data is stored in JSON files for persistence.
"""

import json
import os

class Hotel:
    """Represents a hotel with basic operations."""
    DATA_FILE = "hotels.json"

    def __init__(self, hotel_id, name, location, rooms_available):
        self.hotel_id = hotel_id
        self.name = name
        self.location = location
        self.rooms_available = rooms_available

    def to_dict(self):
        """Creates Hotel dict"""
        return self.__dict__

    @classmethod
    def save_to_file(cls, hotels):
        """Saves Data"""
        with open(cls.DATA_FILE, "w", encoding="utf-8") as file:
            json.dump([hotel.to_dict() for hotel in hotels], file, in

    @classmethod
    def load_from_file(cls):
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

No problems have been detected in the workspace.

Unit Test

For the unit test the hotel_reservation_system_test.py file was created.
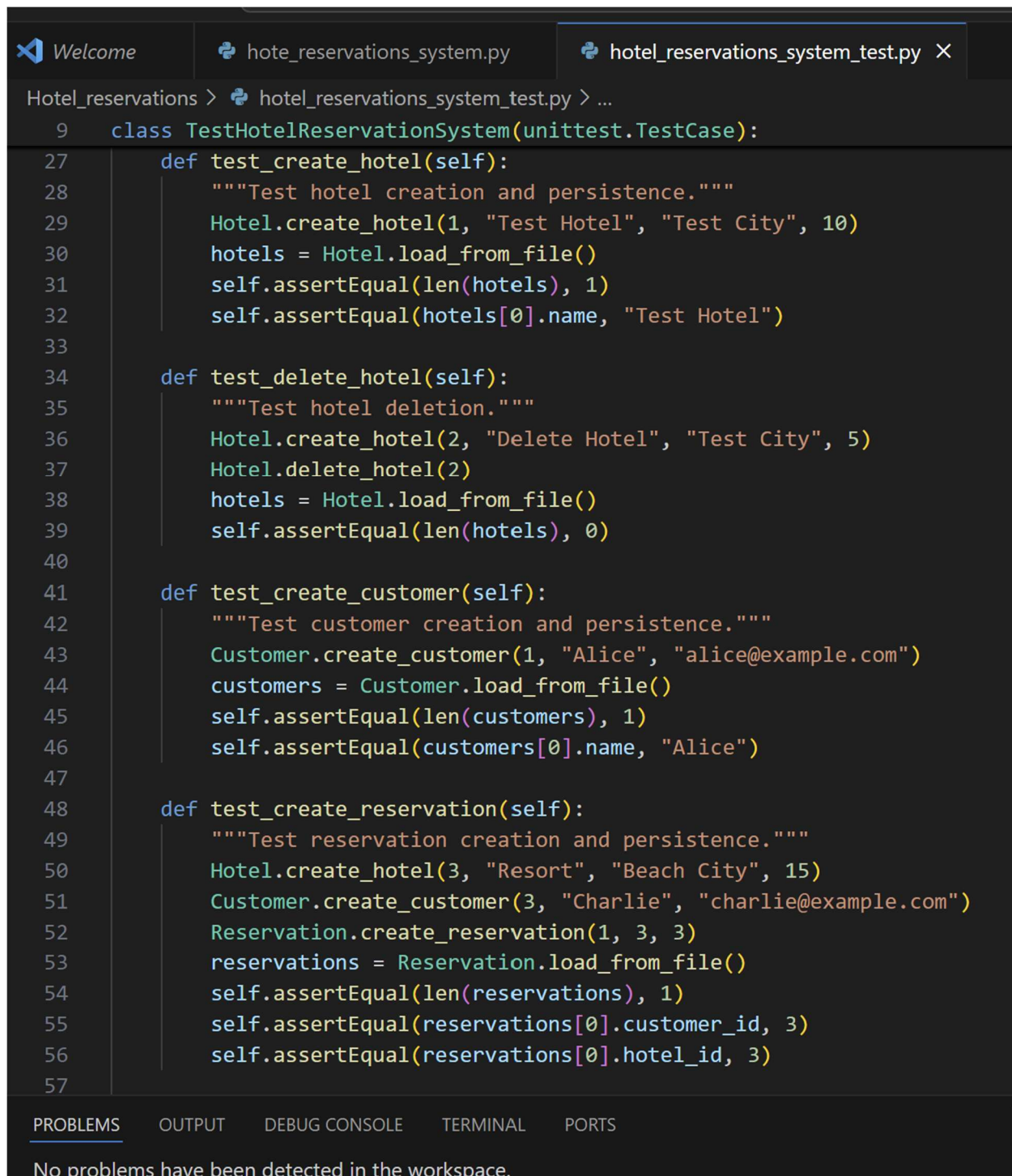
Tests where created for the create, delete and display Hotel methods, and for the create Customer and Reservation methods.

As a good practice, at the beginning of the test Object declaration it was setup configured and clean up of the data files.

```python
class TestHotelReservationSystem(unittest.TestCase):
    """Test object for the Hotel Reservation Systems application"""
    @classmethod
    def setUpClass(cls):
        """Set up test environment before any test runs."""
        cls.test_hotel_file = "test_hotels.json"
        cls.test_customer_file = "test_customers.json"
        cls.test_reservation_file = "test_reservations.json"
        Hotel.DATA_FILE = cls.test_hotel_file
        Customer.DATA_FILE = cls.test_customer_file
        Reservation.DATA_FILE = cls.test_reservation_file

    def tearDown(self):
        """Clean up files after each test."""
        for file in [self.test_hotel_file, self.test_customer_file, self.test_reservation_file]:
            if os.path.exists(file):
                os.remove(file)
```

Here is another snapshot of the code being compliant with pylint.

```python
 9      class TestHotelReservationSystem(unittest.TestCase):
27          def test_create_hotel(self):
28              """Test hotel creation and persistence."""
29              Hotel.create_hotel(1, "Test Hotel", "Test City", 10)
30              hotels = Hotel.load_from_file()
31              self.assertEqual(len(hotels), 1)
32              self.assertEqual(hotels[0].name, "Test Hotel")
33
34          def test_delete_hotel(self):
35              """Test hotel deletion."""
36              Hotel.create_hotel(2, "Delete Hotel", "Test City", 5)
37              Hotel.delete_hotel(2)
38              hotels = Hotel.load_from_file()
39              self.assertEqual(len(hotels), 0)
40
41          def test_create_customer(self):
42              """Test customer creation and persistence."""
43              Customer.create_customer(1, "Alice", "alice@example.com")
44              customers = Customer.load_from_file()
45              self.assertEqual(len(customers), 1)
46              self.assertEqual(customers[0].name, "Alice")
47
48          def test_create_reservation(self):
49              """Test reservation creation and persistence."""
50              Hotel.create_hotel(3, "Resort", "Beach City", 15)
51              Customer.create_customer(3, "Charlie", "charlie@example.com")
52              Reservation.create_reservation(1, 3, 3)
53              reservations = Reservation.load_from_file()
54              self.assertEqual(len(reservations), 1)
55              self.assertEqual(reservations[0].customer_id, 3)
56              self.assertEqual(reservations[0].hotel_id, 3)
57
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

No problems have been detected in the workspace.

Testing

Here is a snapshot of the successful ran of tests in the terminal.

```
PS C:\Users\sito-\Desktop\MNA\04-Software Quality\Tarea 6\Hotel_reservations> python .\hotel_reservations_system_test.py
Customer 'Alice' added.
.Hotel 'Test Hotel' created successfully.
.Hotel 'Resort' created successfully.
Customer 'Charlie' added.
Reservation ID 1 created successfully.
.Hotel 'Delete Hotel' created successfully.
Hotel ID 2 deleted.
.Hotel 'Display Hotel' created successfully.
ID: 4, Name: Display Hotel
Location: Display City, Rooms: 20
.Hotel 'Display Hotel' created successfully.
ID: 4, Name: Display Hotel
Location: Display City, Rooms: 20
.
----------------------------------------------------------------
Ran 5 tests in 0.145s

OK
PS C:\Users\sito-\Desktop\MNA\04-Software Quality\Tarea 6\Hotel_reservations>

.Hotel 'Display Hotel' created successfully.
ID: 4, Name: Display Hotel
Location: Display City, Rooms: 20
.
----------------------------------------------------------------
Ran 5 tests in 0.145s

OK
PS C:\Users\sito-\Desktop\MNA\04-Software Quality\Tarea 6\Hotel_reservations>
.Hotel 'Display Hotel' created successfully.
ID: 4, Name: Display Hotel
Location: Display City, Rooms: 20
.
----------------------------------------------------------------
Ran 5 tests in 0.145s

.Hotel 'Display Hotel' created successfully.
ID: 4, Name: Display Hotel
```

```
Ran 5 tests in 0.145s

.Hotel 'Display Hotel' created successfully.
ID: 4, Name: Display Hotel
Location: Display City, Rooms: 20

.
-----------------------------------------------------
.Hotel 'Display Hotel' created successfully.
ID: 4, Name: Display Hotel
Location: Display City, Rooms: 20

.Hotel 'Display Hotel' created successfully.
ID: 4, Name: Display Hotel
Location: Display City, Rooms: 20
ID: 4, Name: Display Hotel
Location: Display City, Rooms: 20
Location: Display City, Rooms: 20

.
-----------------------------------------------------
Ran 5 tests in 0.145s


-----------------------------------------------------
Ran 5 tests in 0.145s


Ran 5 tests in 0.145s


OK
PS C:\Users\sito-\Desktop\MNA\04-Software Quality\T
```