ACTIVITY 6.2
PROGRAMMING EXERCISES 3

TC4017 Software testing and quality assurance

Student:
Arturo Alfonso Gallardo Jasso A01795510

Professors:
Dr. Gerardo Padilla Zárate
Sajith Delgado Miranda

Activity instructions

1. Generate a repository for the programming exercises in Git.
2. Review the instructions for the programs to be implemented. NOTE: The file with the description mentions using files that are included. This is an error in the instruction.
3. Implement the programs listed at the end of the document using the **Python** language.
4. Follow the PEP-8 coding standard.
5. Verify the correct execution of your programs by generating tests for each exercise using the indicated resources. Document the results.
6. Install the flake8 package using PIP.
7. If you have any doubts about its usage, check the Flake8 tutorial.
8. Verify that your programs do not generate errors or issues using pylint.
9. Flake8 Error Codes:
    a. E*/W***: Errors and warnings from pycodestyle
    b. F*: Detections from PyFlakes
    c. C9: Detections of circular complexity by McCabe script
10. Fix all the issues found by **flake8** and verify that your program continues to function correctly.
11. When finished, upload your programs to your personal repository.
12. The project should be named: StudentID_ActivityNumberA4.2
13. Submit the repository link in the Canvas assignment.
14. Also, upload the source files of the task to Canvas.

Repository

The project folder and a copy of this PDF report can be found in the following link:

https://github.com/AAGJ89/MNA_TC4017_SW.git

Platform details

Python 3.10.11

Visual Studio Code 1.96.4

Problem 1. Reservation System

1.1 Software Requirements

Req 1. Implement a set of classes in Python that implements two abstractions:

1. Hotel
2. Reservation
3. Customers

Req 2. Implement a set of methods to handle the next persistent behaviors (stored in files):

1. Hotels
    a. Create Hotel
    b. Delete Hotel
    c. Display Hotel information
    d. Modify Hotel Information
    e. Reserve a Room
    f. Cancel a Reservation
2. Customer
    a. Create Customer
    b. Delete a Customer
    c. Display Customer Information
    d. Modify Customer Information
3. Reservation
    a. Create a Reservation (Customer, Hotel)
    b. Cancel a Reservation

You are free to decide the attributes within each class that enable the required behavior.

Req 3. Implement unit test cases to exercise the methods in each class. Use the unittest module in Python.

Req 4. The code coverage for all unittests should accumulate at least 85% of line coverage.

Req 5. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue.

Req 6. Be compliant with PEP8.

Req 7. The source code must show no warnings using Fleak and PyLint.Req1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a list of items (presumable numbers).

1.2 Code Highlights

- Two versions of the “ReservationSystem” were generated. First version demonstrated the ability to meet the requirements with nested if under a While loop. The second version was transformed to a class and functions as part of the refactoring process. The second version was modified to meet PEP8 and FLAKE8.
- Two versions of the Test_ReservationSystem were created. With the first meeting the criteria for coverage above 85%, and the second meets PEP8 and FLAKE8.
- Figure 1 shows the first report from the first version of Test Reservation System with a 78%. Then, after ensuring to cover more lines of the code, it finished in 94%.

```
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> coverage run -m unittest TesttoReservationSystem.py
..........
----------------------------------------------------------------------
Ran 10 tests in 0.043s

OK
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> coverage report -m
Name                         Stmts   Miss  Cover   Missing
----------------------------------------------------------
ReservationSystem_v2.py        102     35    66%   16-18, 43, 47-48, 52-67, 77, 96, 99-100, 103-109, 118, 137, 140-141
TesttoReservationSystem.py      58      1    98%   84
----------------------------------------------------------
TOTAL                          160     36    78%
```

Figure 1

```
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> coverage run -m unittest TesttoReservationSystem.py
............Corrupted JSON file detected. Initializing with empty data.
.....
----------------------------------------------------------------------
Ran 17 tests in 0.090s

OK
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> coverage report -m
Name                         Stmts   Miss  Cover   Missing
----------------------------------------------------------
ReservationSystem_v2.py        102     11    89%   67, 99-100, 103-109, 140-141
TesttoReservationSystem.py      92      1    99%   144
----------------------------------------------------------
TOTAL                          194     12    94%
```

Figure 2

- Figure 3 shows compliance of the revision 2 of the code with the criteria at 93% total.

```
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> coverage run -m unittest Test_ReservationSystem_v2.py
..........Corrupted JSON file detected. Starting empty file.
.....
----------------------------------------------------------------------
Ran 15 tests in 0.134s

OK
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> coverage report -m
Name                           Stmts   Miss  Cover   Missing
------------------------------------------------------------
ReservationSystem_v2.py          106     13    88%   60, 89, 129, 133-134, 139-149, 199-200
Test_ReservationSystem_v2.py      87      1    99%   196
------------------------------------------------------------
TOTAL                            193     14    93%
```

Figure 3

- Pylint provided very low scores for “ReservationSystem” at 4.85% (Figure 4) and around 5% for “Test_ReservationSystem”. The same for FLAKE8.

```
ReservationSystem_v2.py:144:4: C0116: Missing function or method docstring (missing-function-docstring)
ReservationSystem_v2.py:144:4: R0917: Too many positional arguments (7/5) (too-many-positional-arguments)
ReservationSystem_v2.py:154:4: C0116: Missing function or method docstring (missing-function-docstring)

--------------------------------------------------------------------
Your code has been rated at 7.77/10 (previous run: 4.85/10, +2.91)
```

Figure 4

- After resolving the errors, both programs were complaint with the standards as the next picture shows.

```
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> pylint ReservationSystem_v2.py

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> flake8 ReservationSystem_v2.py
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> pylint Test_ReservationSystem_v2.py

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity> flake8 Test_ReservationSystem_v2.py
(SWTest_VEnv) PS D:\MNA_SWFiles\TC4017_SW-Testing\6_2-Activity>
```

Figure 5

- Both codes meet the requirements and the objectives of this task.

1.3 Records

- The data for Hotels, Customers and Reservations were stored in the file TestStoredData. Each of these classes were storing very detailed information such as status (Active or Eliminated), creation date or last update date. All data is available in the repository.


```
"hotels": [
    {
        "id": "10",
        "name": "Test Hotel",
        "description": "A test hotel for unit testing",
        "location": "Test Location",
        "telephone": "123-456-7890",
        "amenities": "WiFi, Pool",
        "pet_allowance": "Yes",
        "rate": "150",
        "status": "Active",
        "created_at": "2025-02-16 21:27:36",
        "updated_at": "2025-02-16 21:27:36"
    },
```


Figure 6

Figure 7

Figure 8