Refactoring - Report and tests

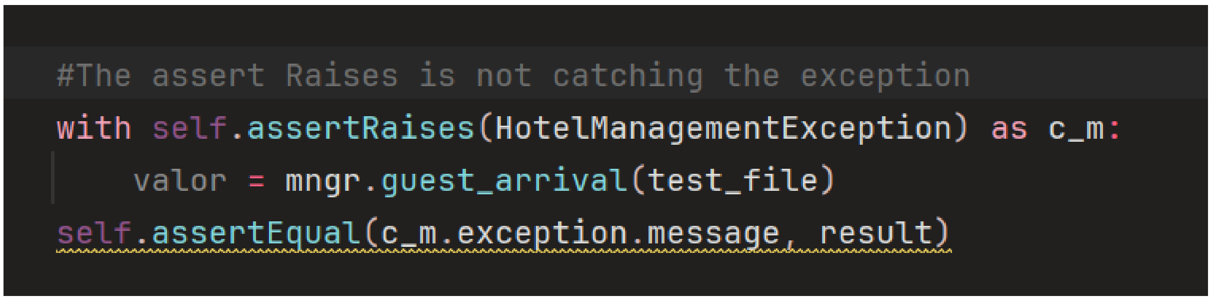
Alejandro Sanchez Vega - 100495744
Iciar García Izquierdo - 100495789
Isabel Hernanz García - 100495778

Introduction:

This document contains the tests developed and the reporting information about the refactoring done. The code that we refactored most was the one in HotelManager since it is where most of the code has been developed.
First, we looked for mysterious names: All class names were self explanatory, as well as all the methods' names. Then, we moved onto looking for names in variables. The first variable we changed names was "res", inside the validation methods as "res" did not explain what this variable was doing. As it was checking if the regular expression was compiled, we named it "regex_check". There are more variables that may be confusing, but since most of them are one-lettered variables that start with the first letter of what they mean (like f as file, e as exception…), we did not change them.
After removing them we extracted all the validation methods and created a new package to use inheritance. We created a superclass containing the information that matched all the validation methods, and then subclasses for each of the different parameters in order to check the different code. The same was done for the json files, and furthermore we refactored data from json management in functions 2 and 3.

The first commits did indeed have errors, but were done in order to be able to discuss how to solve them in group. Most of the errors were due to circular imports but were solved successfully. Then, we took some screenshots of the tests each time big changes were made and committed.

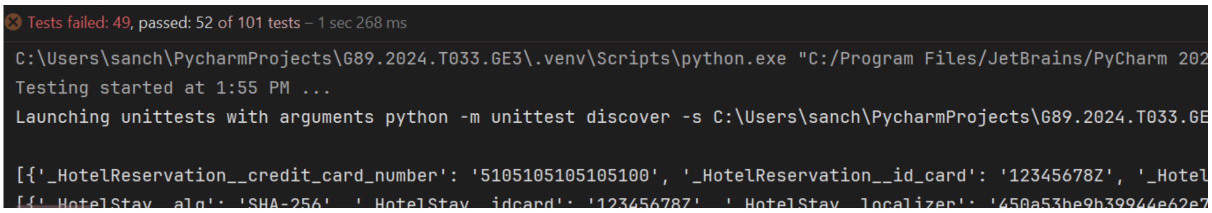
The test for the CheckIn function gives error in the case where the JSON format is invalid. Even Though this happens we committed as if it they were right because the program works as intended, but the test is failing to catch the exception.



```
#The assert Raises is not catching the exception
with self.assertRaises(HotelManagementException) as c_m:
    valor = mngr.guest_arrival(test_file)
self.assertEqual(c_m.exception.message, result)
```

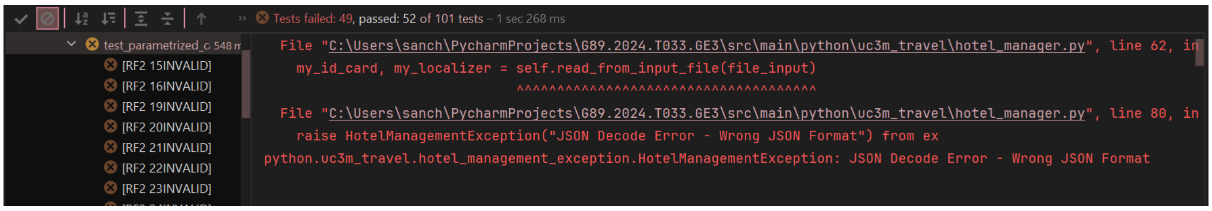
line 124

Specifically, the code part in charge of catching the error is this one, which being compared with the other test error handling code is identical. So we couldn't find and fix the reason for this error.
But we checked that at the end, the code works completely as intended.



```
Tests failed: 49, passed: 52 of 101 tests – 1 sec 268 ms
C:\Users\sanch\PycharmProjects\G89.2024.T033.GE3\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 202
Testing started at 1:55 PM ...
Launching unittests with arguments python -m unittest discover -s C:\Users\sanch\PycharmProjects\G89.2024.T033.GE

[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_Hotel
[{'_HotelStay__alg': 'SHA-256', '_HotelStay__idcard': '12345678Z', '_HotelStay__localizer': '450a53be9b39944e62e7
```

at the end all the test pass expect those for the CheckIn specifically the JSON format Error.

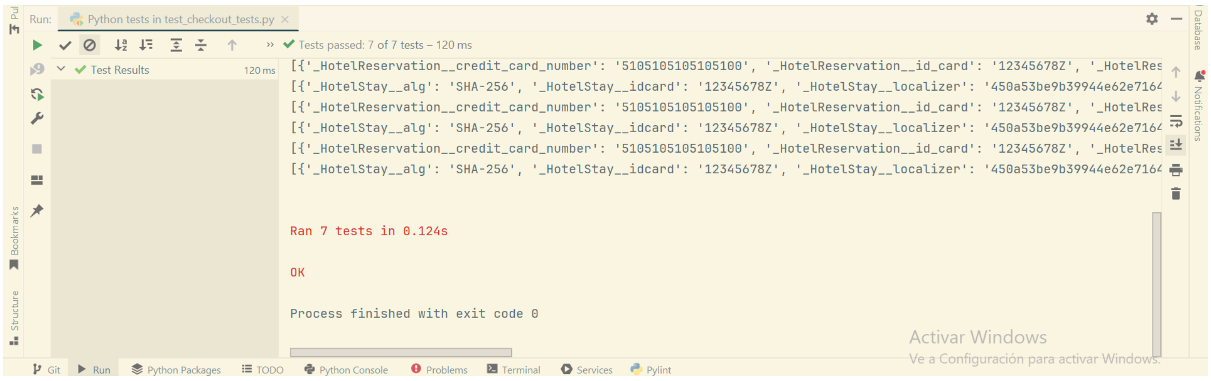


```
Tests failed: 49, passed: 52 of 101 tests – 1 sec 268 ms

test_parametrized_c 548 m
  [RF2 15INVALID]
  [RF2 16INVALID]
  [RF2 19INVALID]
  [RF2 20INVALID]
  [RF2 21INVALID]
  [RF2 22INVALID]
  [RF2 23INVALID]

  File "C:\Users\sanch\PycharmProjects\G89.2024.T033.GE3\src\main\python\uc3m_travel\hotel_manager.py", line 62, in
    my_id_card, my_localizer = self.read_from_input_file(file_input)
                               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "C:\Users\sanch\PycharmProjects\G89.2024.T033.GE3\src\main\python\uc3m_travel\hotel_manager.py", line 80, in
    raise HotelManagementException("JSON Decode Error - Wrong JSON Format") from ex
python.uc3m_travel.hotel_management_exception.HotelManagementException: JSON Decode Error - Wrong JSON Format
```

In this image we can see that all the errors are of the same type, and are caused because of the above code not catching the HotelManagementException exception.

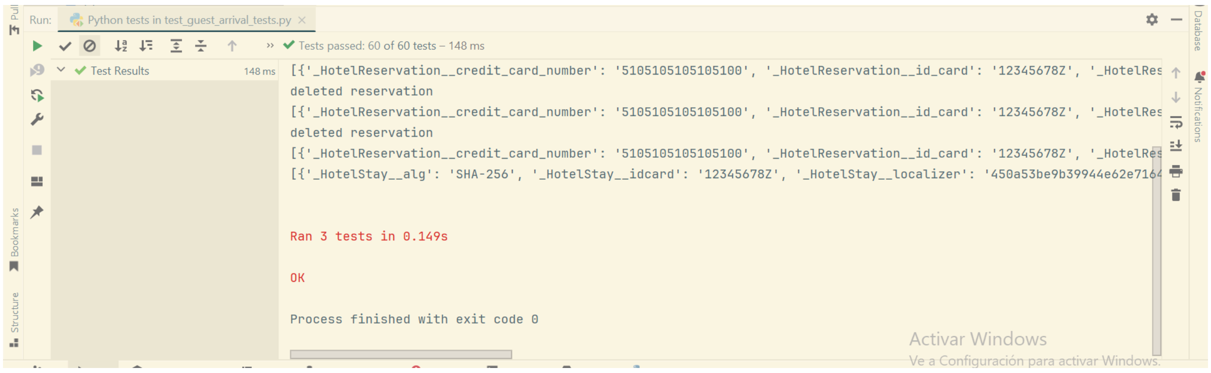
After finishing the refactoring of the code, we did the singleton and its test cases. In this last part of the assignment, it was said to be done in a few files (hotel_manager, and the storage files) but we only developed the singleton for hotel_manager because in the implementation for the storage files we were not able to make it work correctly.

Lastly, here is the results of the tests cases in the final commit:
Tests for the CheckOut function:



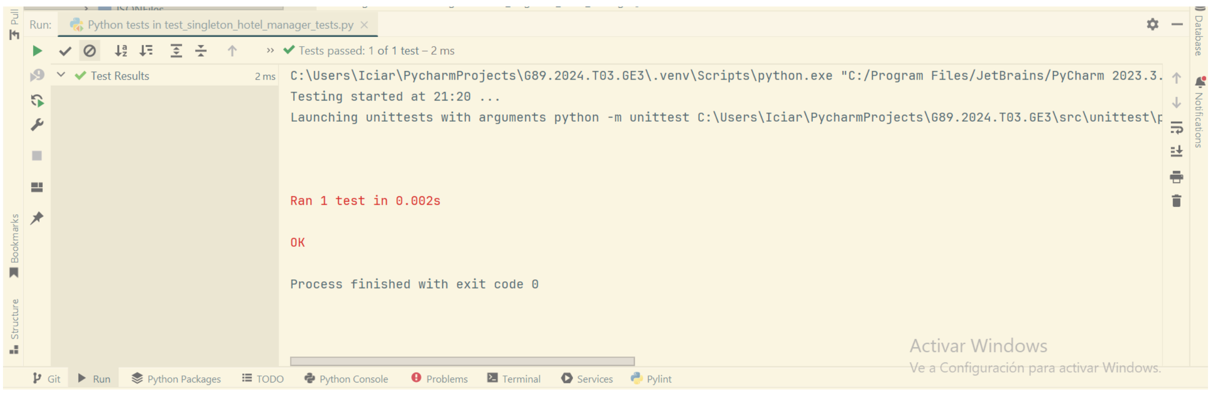
```
Run:  Python tests in test_checkout_tests.py
Tests passed: 7 of 7 tests – 120 ms
Test Results  120 ms

[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelStay__alg': 'SHA-256', '_HotelStay__idcard': '12345678Z', '_HotelStay__localizer': '450a53be9b39944e62e7164
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelStay__alg': 'SHA-256', '_HotelStay__idcard': '12345678Z', '_HotelStay__localizer': '450a53be9b39944e62e7164
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelStay__alg': 'SHA-256', '_HotelStay__idcard': '12345678Z', '_HotelStay__localizer': '450a53be9b39944e62e7164


Ran 7 tests in 0.124s

OK

Process finished with exit code 0
```

Tests for the GuestArrival function:



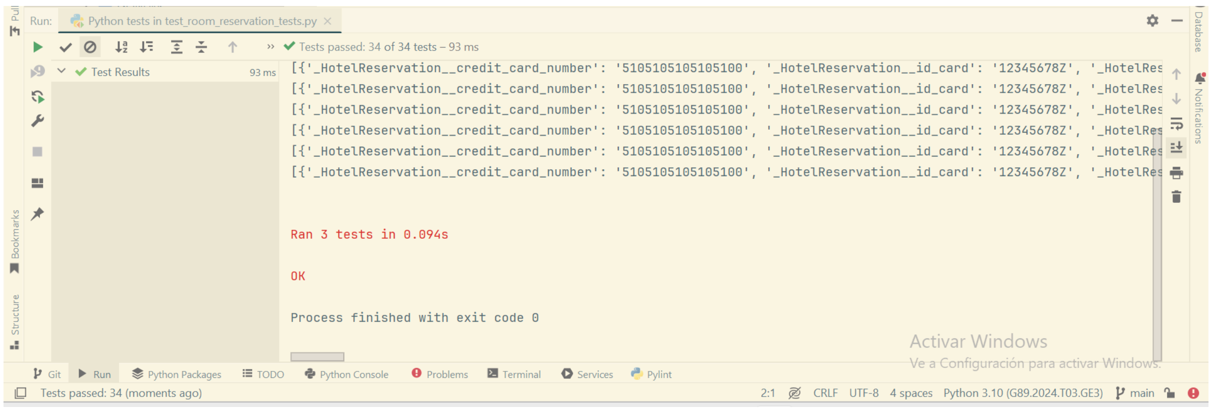
```
Run:  Python tests in test_guest_arrival_tests.py
Tests passed: 60 of 60 tests – 148 ms
Test Results  148 ms

[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
deleted reservation
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
deleted reservation
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelStay__alg': 'SHA-256', '_HotelStay__idcard': '12345678Z', '_HotelStay__localizer': '450a53be9b39944e62e7164


Ran 3 tests in 0.149s

OK

Process finished with exit code 0
```

Tests for the singleton:


Run: Python tests in test_singleton_hotel_manager_tests.py
Tests passed: 1 of 1 test – 2 ms
Test Results 2 ms
Git Run Python Packages TODO Python Console Problems Terminal Services Pylint
Activar Windows
Ve a Configuración para activar Windows.

```
C:\Users\Iciar\PycharmProjects\G89.2024.T03.GE3\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.
Testing started at 21:20 ...
Launching unittests with arguments python -m unittest C:\Users\Iciar\PycharmProjects\G89.2024.T03.GE3\src\unittest\p



Ran 1 test in 0.002s

OK

Process finished with exit code 0
```

Tests for RoomReservation function:


Run: Python tests in test_room_reservation_tests.py
Tests passed: 34 of 34 tests – 93 ms
Test Results 93 ms
Git Run Python Packages TODO Python Console Problems Terminal Services Pylint
Activar Windows
Ve a Configuración para activar Windows.
Tests passed: 34 (moments ago)
2:1 CRLF UTF-8 4 spaces Python 3.10 (G89.2024.T03.GE3) main

```
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes
[{'_HotelReservation__credit_card_number': '5105105105105100', '_HotelReservation__id_card': '12345678Z', '_HotelRes


Ran 3 tests in 0.094s

OK

Process finished with exit code 0
```