TESTING:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Display menu | Display menu correctly displayed |  | Pass |
|  |  |  |  |
| (Rooms Initialised correctly) After program starts, Press ‘V’ | Displays ‘empty’ for all rooms |  | Pass |
| Validation | It should be a loop till an integer is entered |  | Pass |
| (Add customer “Bob Smith, 3 people, card number 3556622887 ” to room 5) Select A, enter “Bob” | Press ‘v’ Displays “Bob Smith, Number of people : 3 ” for room 5 |  | Pass |
| (Delete customer from room number 2) | Press ‘v’ Displays room 2 as empty |  | Pass |
| View empty rooms | Press ‘e’, Displays all empty rooms |  | Pass |
| Find customer | Press ‘f’ to find customer BOB in room 2 |  |  |
| (Store Data in a file) | Print statement: file has been stored |  | Pass |
| (Load Data) | Shows data from the file stored |  | Pass |
| (Order) | Shows the names of the customers in alphabetical order |  | Pass |
| Quit | Press ‘q’ to quite from the code |  |  |

The test cases were chosen on the basis of everything I tested, while making the code. I made sure there were test cases for validation are present as well, since that was added later on. As I was making the code and testing, I started filling out the test tables initially there were a lot of errors but they kept getting solved and I kept updating the test cases. I tried to cover all parts of the display menu in my test cases to show that they all work and function well.

I chose to show how the add and delete functions work, because customers are constantly being changed and they need to be added and removes continuously so these functions have to work flawlessly. All other methods were shown as well since they r required on a regular basis in a hotel and test cases show that they do work.